

 **ACTEX Learning**

**Exam PA**  
**Study Guide**

**11<sup>th</sup> Edition**

**Ambrose Lo, PhD, FSA, CERA**



**An SOA Exam**



 **ACTEX Learning**

**Exam PA**

**Study Guide**

**11<sup>th</sup> Edition**

**Ambrose Lo, PhD, FSA, CERA**



*Actuarial & Financial Risk Resource Materials*  
Since 1972

Copyright © 2024, ACTEX Learning, a division of ArchiMedia Advantage Inc.

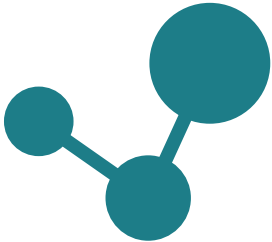
Printed in the United States of America.

No portion of this ACTEX Study Guide  
may be reproduced or transmitted in any part or by  
any means without the permission of the publisher.



# Welcome to Actuarial University

Actuarial University is a reimagined platform built around a more simplified way to study. It combines all the products you use to study into one interactive learning center.



You can find integrated topics using this network icon.


When this icon appears, it will be next to an important topic in the manual. Click the **link** in your digital manual, or search the underlined topic in your print manual.

1. Login to: [www.actuarialuniversity.com](http://www.actuarialuniversity.com)

2. Locate the **Topic Search** on your exam dashboard and enter the word or phrase into the search field, selecting the best match.

3. A topic “**Hub**” will display a list of integrated products that offer more ways to study the material.

4. Here is an example of the topic **Pareto Distribution**:

 Pareto Distribution ×

The (Type II) **Pareto distribution** with parameters  $\alpha, \beta > 0$  has pdf

$$f(x) = \frac{\alpha\beta^\alpha}{(x + \beta)^{\alpha+1}}, \quad x > 0$$

and cdf

$$F_P(x) = 1 - \left(\frac{\beta}{x + \beta}\right)^\alpha, \quad x > 0.$$

If  $X$  is Type II Pareto with parameters  $\alpha, \beta$ , then

$$E[X] = \frac{\beta}{\alpha - 1} \text{ if } \alpha > 1,$$

and

$$Var[X] = \frac{\alpha\beta^2}{\alpha - 2} - \left(\frac{\alpha\beta}{\alpha - 1}\right)^2 \text{ if } \alpha > 2.$$

- ACTEX Manual for P →
- Probability for Risk Management, 3rd Edition 🔒
- GOAL for SRM 🔒
- ASM Manual for IFM 🔒
- Exam FAM-S Video Library 🔒

Related Topics ▾

Within the **Hub** there will be unlocked and locked products.

**Unlocked Products** are the products that you own.

ACTEX Manual for P →

**Locked Products** are products that you do not own, and are available for purchase.

Probability for Risk Management, 3rd Edition 🔒

Many of Actuarial University's features are already unlocked with your study program, including:

<b>Instructional Videos*</b>	<b>Planner</b>
<b>Topic Search</b>	<b>Formula &amp; Review Sheet</b>

**Make your study session more efficient with our Planner!**

✓	7/1/2023 - 7/16/2023	Interest Rates and the Time Value of Money		→
✓	7/16/2023 - 8/12/2023	Annuities		→
✓	8/12/2023 - 8/27/2023	Loan Repayment		→
✓	8/27/2023 - 9/15/2023	Bonds		→
✓	9/15/2023 - 9/22/2023	Yield Rate of an Investment		→
✓	9/22/2023 - 10/11/2023	The Term Structure of Interest Rates		→
✓	10/11/2023 - 10/30/2023	Asset-Liability Management		→

*\*Available standalone, or included with the Study Manual Program Video Bundle*

# Contents

<b>Preface</b>	<b>xiii</b>
P.1 About Exam PA . . . . .	xiv
P.2 About this Study Manual . . . . .	xxii
<b>I A Crash Course in R</b>	<b>1</b>
<b>Chapter 1 Basics of R Programming</b>	<b>3</b>
1.1 Getting Started in R . . . . .	5
1.1.1 Basic Infrastructure . . . . .	5
1.1.2 Data Types . . . . .	12
1.2 Data Structures . . . . .	16
1.2.1 Vectors . . . . .	16
1.2.2 Matrices . . . . .	22
1.2.3 Data Frames . . . . .	26
1.2.4 Lists . . . . .	32
1.2.5 Sidebar: Functions . . . . .	34
1.3 Basic Data Management . . . . .	39
1.4 for Loops . . . . .	53
1.5 End-of-Chapter Practice Problems . . . . .	60
<b>Chapter 2 Data Exploration and Visualization</b>	<b>69</b>
2.1 Making ggplots . . . . .	70
2.1.1 Basic Features . . . . .	70
2.1.2 Customizing Your Plots . . . . .	84
2.2 Data Exploration . . . . .	86
2.2.1 Univariate Data Exploration . . . . .	87
2.2.2 Bivariate Data Exploration . . . . .	102
2.3 End-of-Chapter Practice Problems . . . . .	114
<b>II Theory of and Case Studies in Predictive Analytics</b>	<b>123</b>
<b>Chapter 3 Linear Models</b>	<b>125</b>
3.1 A Primer on Predictive Analytics . . . . .	127
3.1.1 Basic Terminology . . . . .	128

3.1.2	The Model Building Process . . . . .	134
3.1.3	Bias-Variance Trade-off . . . . .	162
3.1.4	Feature Generation and Selection . . . . .	177
3.2	Linear Models: Conceptual Foundations . . . . .	188
3.2.1	Model Formulation . . . . .	188
3.2.2	Model Evaluation and Validation . . . . .	191
3.2.3	Feature Generation . . . . .	204
3.2.4	Feature Selection . . . . .	229
3.2.5	Regularization . . . . .	237
3.3	Case Study 1: Fitting Linear Models in R . . . . .	246
3.3.1	Exploratory Data Analysis . . . . .	248
3.3.2	Simple Linear Regression . . . . .	254
3.3.3	Multiple Linear Regression . . . . .	261
3.3.4	Evaluation of Linear Models . . . . .	276
3.4	Feature Selection and Regularization . . . . .	280
3.4.1	Preparatory Work . . . . .	280
3.4.2	Model Construction and Feature Selection . . . . .	295
3.4.3	Model Validation . . . . .	314
3.4.4	Regularization . . . . .	318
	Conceptual Review Questions for Chapter 3 . . . . .	330
<b>Chapter 4</b>	<b>Generalized Linear Models</b>	<b>337</b>
4.1	Conceptual Foundations of GLMs . . . . .	338
4.1.1	Selection of Target Distributions and Link Functions . . . . .	341
4.1.2	Weights and Offsets . . . . .	352
4.1.3	Fitting and Assessing the Performance of a GLM . . . . .	357
4.1.4	Performance Metrics for Classifiers . . . . .	372
4.2	Case Study 1: GLMs for Continuous Target Variables . . . . .	388
4.2.1	Data Preparation . . . . .	388
4.2.2	Model Construction and Evaluation . . . . .	390
4.2.3	Model Validation and Interpretation . . . . .	398
4.3	Case Study 2: GLMs for Binary Target Variables . . . . .	402
4.3.1	Data Exploration and Preparation . . . . .	403
4.3.2	Model Construction and Selection . . . . .	416
4.3.3	Interpretation of Model Results . . . . .	432
4.4	Case Study 3: GLMs for Count and Aggregate Loss Variables . . . . .	437
4.4.1	Data Exploration and Preparation . . . . .	437
4.4.2	Model Construction and Evaluation . . . . .	447
4.4.3	Predictions . . . . .	459
	Conceptual Review Questions for Chapter 4 . . . . .	464
<b>Chapter 5</b>	<b>Tree-Based Models</b>	<b>469</b>
5.1	Conceptual Foundations of Decision Trees . . . . .	470
5.1.1	Single Decision Trees . . . . .	470
5.1.2	Ensemble Tree Model I: Random Forests . . . . .	504
5.1.3	Ensemble Tree Model II: Boosting . . . . .	510



5.2	Mini-Case Study: A Toy Decision Tree . . . . .	519
5.2.1	Basic Functions and Arguments . . . . .	520
5.2.2	Pruning a Decision Tree . . . . .	526
5.3	Extended Case Study: Classification Trees . . . . .	533
5.3.1	Problem Set-up and Preparatory Steps . . . . .	533
5.3.2	Construction and Evaluation of Single Classification Trees . . . . .	546
5.3.3	Construction and Evaluation of Ensemble Trees . . . . .	568
	Conceptual Review Questions for Chapter 5 . . . . .	590
<b>Chapter 6</b>	<b>Unsupervised Learning Techniques</b>	<b>595</b>
6.1	Principal Components Analysis . . . . .	597
6.1.1	Conceptual Foundations . . . . .	597
6.1.2	Additional PCA Issues . . . . .	603
6.1.3	A Simple Case Study . . . . .	611
6.2	Cluster Analysis . . . . .	635
6.2.1	$K$ -means Clustering . . . . .	638
6.2.2	Hierarchical Clustering . . . . .	647
6.2.3	Practical Issues in Clustering . . . . .	657
6.2.4	A Simple Case Study . . . . .	659
	Conceptual Review Questions for Chapter 6 . . . . .	676
<b>III</b>	<b>Final Preparation</b>	<b>681</b>
<b>Chapter 7</b>	<b>Discussions on Past PA Exams</b>	<b>683</b>
7.1	April 2024 Exam . . . . .	686
7.2	October 2023 Exam . . . . .	687
7.3	April 2023 Exam . . . . .	700
7.4	October 2022 Exam . . . . .	718
7.4.1	October 11 Exam . . . . .	720
7.4.2	October 12 Exam . . . . .	737
7.5	April 2022 Exam . . . . .	745
7.5.1	April 12 Exam . . . . .	745
7.5.2	April 14 Exam . . . . .	760
7.6	December 2021 Exam . . . . .	770
7.6.1	December 13 Exam . . . . .	771
7.6.2	December 14 Exam . . . . .	786
7.7	June 2021 Exam . . . . .	798
7.7.1	June 21 Exam . . . . .	798
7.7.2	June 22 Exam . . . . .	813
7.8	December 2020 Exam . . . . .	825
7.8.1	December 7 Exam . . . . .	825
7.8.2	December 8 Exam . . . . .	836
7.9	June 2020 Exam . . . . .	851
7.9.1	June 16 and 19 Exams . . . . .	851
7.9.2	June 17 and 18 Exams . . . . .	861

7.10	December 2019 Exam . . . . .	872
7.11	June 2019 Exam . . . . .	884
<b>Chapter 8</b>	<b>Practice Exams</b>	<b>893</b>
	Practice Exam 1 Project Statement . . . . .	897
	Practice Exam 1 Suggested Solutions . . . . .	912
	Practice Exam 2 Project Statement . . . . .	930
	Practice Exam 2 Suggested Solutions . . . . .	947
<b>Chapter A</b>	<b>Analysis of Past PA Exam Questions by Theme</b>	<b>965</b>
A.1	Problem Definition . . . . .	966
	A.1.1 General Matters . . . . .	966
	A.1.2 Choice of Target Variables . . . . .	966
A.2	Data . . . . .	967
	A.2.1 General Use . . . . .	967
	A.2.2 Data Exploration . . . . .	969
	A.2.2.1 Univariate . . . . .	969
	A.2.2.2 Bivariate . . . . .	970
	A.2.2.3 Interaction . . . . .	973
	A.2.3 Data Cleaning/Preparation . . . . .	974
A.3	General Model Construction, Evaluation, and Selection . . . . .	978
	A.3.1 Model Complexity and Bias-Variance Trade-off . . . . .	978
	A.3.2 Cross-Validation . . . . .	978
	A.3.3 Model Assessment and Comparison . . . . .	979
	A.3.3.1 Regression Case . . . . .	979
	A.3.3.2 Classification Case . . . . .	981
A.4	GLMs . . . . .	984
	A.4.1 Target Distributions . . . . .	984
	A.4.2 Link Functions . . . . .	985
	A.4.3 Predictions . . . . .	985
	A.4.4 Interpretation of Summary Output . . . . .	986
	A.4.5 Categorical Predictors . . . . .	986
	A.4.6 Feature Generation . . . . .	987
	A.4.7 Interpretation of Coefficients . . . . .	988
	A.4.8 Predictors as Numeric vs. Categorical . . . . .	990
	A.4.9 Offsets and Weights . . . . .	991
	A.4.10 Stepwise Selection . . . . .	991
	A.4.11 Regularization . . . . .	993
	A.4.12 Diagnostics . . . . .	995
A.5	Decision Trees: Single Trees . . . . .	995
	A.5.1 Interpretations of Tree Output . . . . .	995
	A.5.2 Transformations of Target Variables . . . . .	997
	A.5.3 Classification Trees . . . . .	997
	A.5.4 Pruning . . . . .	998
	A.5.5 Categorical Predictors . . . . .	999
	A.5.6 Feature Generation . . . . .	1000

---

	A.5.7	Trees vs. GLMs . . . . .	1000
A.6		Decision Trees: Ensemble Trees . . . . .	1001
	A.6.1	General Ensemble Trees . . . . .	1001
	A.6.2	Random Forests . . . . .	1001
	A.6.3	Boosting . . . . .	1002
	A.6.4	Interpretational Tools . . . . .	1003
A.7		PCA . . . . .	1004
	A.7.1	Mechanics/Uses . . . . .	1004
	A.7.2	Interpretation . . . . .	1004
	A.7.3	Additional PCA Issues . . . . .	1005
A.8		Cluster Analysis . . . . .	1006
	A.8.1	<i>K</i> -means Clustering . . . . .	1006
	A.8.2	Hierarchical Clustering . . . . .	1007



# Preface

## ⚠ NOTE TO STUDENTS ⚠

Please read this preface carefully 📖, even if it looks long. It contains **VERY** important information that will help you navigate this manual and Exam PA smoothly! 👍

## Why this Study Manual?

*“The PA modules are so difficult to follow.”*

*“The PA modules make things unnecessarily complicated and are riddled with errors.”*

*“I feel that the PA modules don’t cover enough ground for me to handle the exam well. I have to supplement my learning with external resources.”*


*“I hate having to alternate among the PA modules, the R Markdown files, the required textbooks, and online readings.”*

*“There is a lack of useful study resources for Exam PA in the market.”*

These are some of the most common comments PA exam candidates who studied for the exam solely using the Society of Actuaries (SOA)’s e-learning modules have voiced on Internet forums, e.g., the old Actuarial Outpost, Reddit 🗨, Discord 🗨. These “complaints” and the importance of passing this exam to earn the Associateship of the Society of Actuaries (ASA) designation in today’s exam curriculum have motivated me to develop a completely new Exam PA study manual with the goal of streamlining, synthesizing, and augmenting the materials in the PA e-learning modules in a coherent and exam-oriented format. With this manual, you will have in your possession a reliable learning resource that hosts all of the useful materials in a single place and shows you how to prepare for this exam effectively and efficiently. There is no longer a need to alternate among the e-learning modules, the suggested textbooks in the syllabus, R markdown files, and additional online readings. Starting from the very basics and adopting a case study approach, we will learn fundamental concepts in predictive analytics, make some fancy and informative graphs 📊 in R (a powerful programming language), implement predictive models step by step in concrete settings, understand what the output in R means, and write your responses to the liking of PA exam graders. No prior knowledge in R or the SRM exam material is assumed.

## P.1 About Exam PA


### Exam Administrations

Exam PA (Predictive Analytics) is a 3.5-hour computer-based exam offered for the first time in December 2018 by the SOA. There are two sittings each year, one in April and one in October,<sup>1</sup> and each testing window lasts for four days. In October 2024, the exam will be delivered via computer-based testing (CBT)  in a Prometric exam center on **October 15-18**. The registration deadline is September 10. You can check out the exam’s official homepage for more information:

<https://www.soa.org/education/exam-req/edu-exam-pa-detail/>.

After you register for the exam online (and pay the exorbitant **\$\$\$** \$1,170 exam fee!) at

<https://www.soa.org/education/exam-req/registration/edu-registration/>,

you will receive an email confirmation letter  from the SOA containing your candidate ID, which will allow you to schedule an appointment at Prometric (<https://www.prometric.com/soa>). You will also receive access to the SOA’s PA e-learning modules until the end of the month in which the exam is administered (April 30 for the April sitting, October 31 for the October sitting). According to the exam homepage and syllabus, these modules

“provide support designed to enhance candidates’ knowledge from the SRM Exam learning objectives and readings”

and

“guidance regarding knowledge and approaches that will be expected in the exam.”

There are a total of 5 modules, plus an additional module that provides an introduction to R. (Well, as the previous page says, the modules are not easy to read, and with this study manual, it is not really necessary to go over the modules. 😊)

### What is Exam PA Like and How to Study for It?

Typically one of the last exams students take before attaining their ASA designation, Exam PA is the first of its kind in the history of actuarial exams that heavily integrates predictive modeling, R programming, and written communication in a fully proctored setting, and this new exam style calls for a completely different approach to assessment as well as learning.

**Exam format.** In Exam PA, you will be asked to perform a data-driven analysis of a business problem,<sup>2</sup> using a combination of general tools for constructing and evaluating predictive models (e.g., training/test set split, cross-validation), and specific types of models and techniques (e.g., generalized linear models, decision trees, principal components analysis, and clustering). Such an analysis does not lend itself to the multiple-choice format of many other preliminary exams you have taken, which can only elicit a simple response. Instead, Exam PA is a computer-based

<sup>1</sup>From 2018 to 2022, Exam PA was held in June and December.

<sup>2</sup>The business problem is not necessarily (and usually not) actuarial in focus. Even if it is actuarially related, there is no expectation that candidates have specific product or practice-area knowledge.

**written-answer** 🗨️ exam consisting of a set of well-defined and independent tasks (usually 9 to 12 tasks), most of which are further broken down into one or more subtasks that require reasonably short answers (you need not write long essays or reports!). The whole exam carries a total of **70 points**, with the points for each task and subtask shown at the beginning of the (sub)task *in italics*. As the exam lasts for 3.5 hours, or 210 minutes, on average you should spend  $210/70 = 3$  minutes per exam point. A 10-point task, for example, should translate into approximately 30 minutes of work. If you have worked on that task for 50 minutes, then you know that it is time to move on.

Wondering where to put your written answers? The exam paper, available in Microsoft Word 📄 format, includes designated spaces labelled “**ANSWER:**” for you to type 🖋️ your written responses to different exam subtasks. At the end of the exam, you will upload 📁 the entire Word file for grading. You will be assessed on both the technical accuracy of your answers as well as the clarity of your thought process. Unlike other ASA exams, questions in Exam PA tend to be more open-ended and often there is not a unique best answer, as is true of predictive modeling in practice. To score high, you are expected to justify your answers carefully and adequately, based on the business problem and your prior knowledge of predictive analytics. ⚠️ Credit is awarded depending on how *good* 👍 or *bad* 👎 your answers are, not (only) whether they are *right* ✓ or *wrong* ✗.

**Typical exam questions.** As I said, PA consists of written-answer (rather than multiple-choice) questions. What are these questions like? To give you a first taste of the exam, here are some representative tasks taken from the latest released exams.

- **Type 1: Conceptual questions**

Each exam has quite a number of subtasks that require you to *describe* or *explain* the predictive analytic concepts covered in the syllabus. Here are some good examples:

▷ April 2024 exam, Task 1 (b):

(2 points) Describe the steps to calculate the within cluster sum of squares using latitude and longitude [two of the predictors].

▷ April 2024 exam, Task 9 (c):

(2 points) You are fitting a GLM model by using LASSO regression.

Identify a hyperparameter that can be tuned and describe how you would tune it using cross validation.


▷ October 2023 exam, Task 2 (a):


(4 points) Describe two similarities and two differences between K-means clustering and hierarchical clustering.

▷ October 2023 exam, Task 5 (c):

(2 points) Describe a bivariate visualization that can be applied to understand the relationship between a numeric variable and a categorical variable.

- ▷ April 2023 exam, Task 5 (a):
  - (3 points) Compare and contrast single decision tree and tree-based ensemble models.
- ▷ April 2023 exam, Task 8 (c):
  - (2 points) Describe the process of searching for the optimal value of the hyperparameter lambda in a lasso regression.

These descriptive subtasks are good ways for the SOA to test your conceptual understanding of predictive analytics. You can secure these easy exam points simply by studying this manual (in particular, the conceptual foundations sections) carefully and practicing explaining different concepts. These subtasks also mean that there are definitions and descriptions you have to memorize  in advance as part of your exam preparation.

- **Type 2: Analytical questions (examining graphs  and output)**

In the majority of the exam tasks, you will examine some externally generated graphs and output, and provide explanations (e.g., why the model behaves in the way shown), interpretations (e.g., what does the output mean or imply?), or recommendations (e.g., which model is the best, in what sense?). Here are some examples:

- ▷ April 2024 exam, Task 6 (a)-(b):
  - (a) (1 point) Interpret the intercept and the coefficient for **fare**.
  - (b) (1 point) Recommend and justify which model is better based upon the output above.
- ▷ October 2023 exam, Task 2 (b):
  - (3 points) Explain the tradeoff between selecting a value of  $K = 2$  and  $K = 4$ . Recommend a value for  $K$  and justify your recommendation.
- ▷ October 2023 exam, Task 4 (b):
  - (2 points) Interpret the Complexity Parameter table. Recommend and justify a CP value to use for the model.
- ▷ April 2023 exam, Task 1 (b):
  - (2 points) Explain, using the graph above, why the **Daytype** variable is statistically significant while the **DayofWeek** variable is not.
- ▷ April 2023 exam, Task 5 (c):
  - (2 points) Determine if this tree shows an interaction between month and year. If there is an interaction, describe it. If not, explain why there is no interaction.

These analytical tasks are more demanding (and interesting!) than tasks of type 1 above, which mainly test the ability to recall, because you are required to formulate your responses based on the given output coupled with your prior knowledge in predictive analytics. It is



not enough to memorize; you will have to reason and apply.

• **Type 3: Simple calculation questions** 📊

There are also some subtasks where you are asked to use the given output to calculate certain model quantities by hand. Examples include:

- ▷ April 2024 exam, Task 5 (a):  
(3 points) Determine the information gain of this split using the entropy measure.
- ▷ April 2024 exam, Task 11 (a):  
(1 point) Calculate how many observations your total sample will contain. Assume you can find 10 observations for each pair of regions.
- ▷ October 2023 exam, Task 4 (a):  
(4 points) Calculate the change in the **Absolute Error**, using the testing data row, between the first decision tree model and building a bagged model using both decision trees. State which of these two approaches yields a better result for this observation. Show all work.
- ▷ October 2023 exam, Task 7 (b):  
(3 points) Calculate the model's predicted 7-year loan repayment rate for each scenario below and show your work:
- ▷ October 2023 exam, Task 10 (c):  
(3 points) Calculate the RMSE and MAE for the test data above using the tree model. Show your work.
- ▷ April 2023 exam, Task 8 (f):  
(2 points) You are provided with the confusion matrix produced by the lasso model with a positive response cutoff threshold of 0.5.  
Calculate sensitivity and specificity. Show all work.

As you can see, the shift of exam focus from working out multiple-choice problems efficiently to crafting computer-aided written responses makes Exam PA a completely different (and hopefully more enjoyable and practical!) learning experience compared with all other ASA exams you have taken. To study for this exam effectively: ⚠️

It is very important to spend time *understanding* the subject, at least at a conceptual level, and learning how to *communicate* your thoughts precisely and concisely. (Having taught in a [CAE university](#) for about 10 years and graded hundreds of mock exams submitted by past PA students, I can say written communication is an area in which actuarial science students leave much to be desired. 🙄) Unlike other ASA exams, you can't expect to do well just by drilling mechanical practice problems again and again mindlessly. Instead, make an effort to *understand*, *describe*, and *explain* things. You will find that translating your thoughts into words is harder than you imagined.

## New Exam Format Effective from April 2023

Ever since Exam PA was introduced in December 2018, its format and style have undergone significant changes. The latest revamp took place in the April 2023 sitting, effective from which the exam time has been reduced remarkably from 5.25 hours to 3.5 hours. Perhaps the more striking change is:

Starting with the April 2023 administration, **R and RStudio (a convenient platform to implement R) will not be available on the exam.**

How will this “big” change affect the exam and our preparation? My answer, which is confirmed by all of the released exams following the new format (the April 2024, October 2023, and April 2023 exams), is:

You will learn the material and prepare for the exam in essentially the same way, perhaps paying less attention to R code syntax. `</>`

Even when R and RStudio were available on the exam from December 2018 to October 2022, Exam PA was never designed as a coding exam. Candidates did have to know *some* R, but only to the extent that they understood what the code (contained in a separate R markdown file generously provided by the SOA) was doing and knew how to make minor adjustments if necessary. The focus of the exam has always been on *understanding* and *interpretation*, reflected by the abundance of past exam questions belonging to Types 1 and 2 above. With R and RStudio no longer available, the only major difference is that the code and output relevant to the exam tasks will be provided directly in the exam paper; you need not take the trouble to run the code in RStudio or see the R output. The emphasis on conceptual understanding and interpretation is likely to remain (or will even be greater).

**⚠** There is one change I do expect to see in the new exam format:

There may be more tasks of Type 3, where you have to do some simple manual calculations based on the R code and R output given.

With all the useful code and output given in the exam paper, you may be asked to explain what a certain number in the output means or how it is calculated, or to use the output to do some simple arithmetic calculations. This is a good way for the SOA to test your deeper understanding. You can't just rely on R to do everything!

## Historical Pass Rates %

The following table shows the number of sitting candidates, number of passing candidates, and pass rates for Exam PA since it was offered in December 2018:

(For written-answer exams, including PA, the SOA does not announce pass marks, i.e., the actual score you have to get to pass the exam, nor does it release the grading rubric. Yes, the grading is very much a black-box process. 📦)

Sitting	# Candidates	# Passing Candidates	Pass Rate
April 2024	2188	1413	64.6%
October 2023	2315	1468	63.4%
April 2023	2193	1552	70.8% (highest ever!)
October 2022	1554	1005	64.7%
April 2022	1171	773	66.0%
December 2021	1922	1321	68.7%
June 2021	1691	1055	62.4%
December 2020	1954	1228	62.8%
June 2020	1389	812	58.5%
December 2019	2048	1098	53.6% (I took this exam! 😊)
June 2019	1282	642	50.1%
December 2018	1042	524	50.3%

The pass rates, which are usually in the **60-70% range**,<sup>3</sup> are higher than those of other ASA-level exams, which are typically 40-50%. Meanwhile, about 40% of the candidates failed every time, even among those who have reached this far in the ASA journey. I have heard of candidates who have failed PA twice or thrice (😞), so the exam is neither a beast nor a breeze! Worst of all, the exam is offered only twice a year, so in the unfortunate event that you fail, you will need to wait another six months, which adds a lot to your travel time to ASA.

<sup>3</sup>They became noticeably higher after COVID-19 broke out possibly due to the [temporary refund policy](#), which allowed students to withdraw 14 days before the exam started. This policy is likely to end soon.

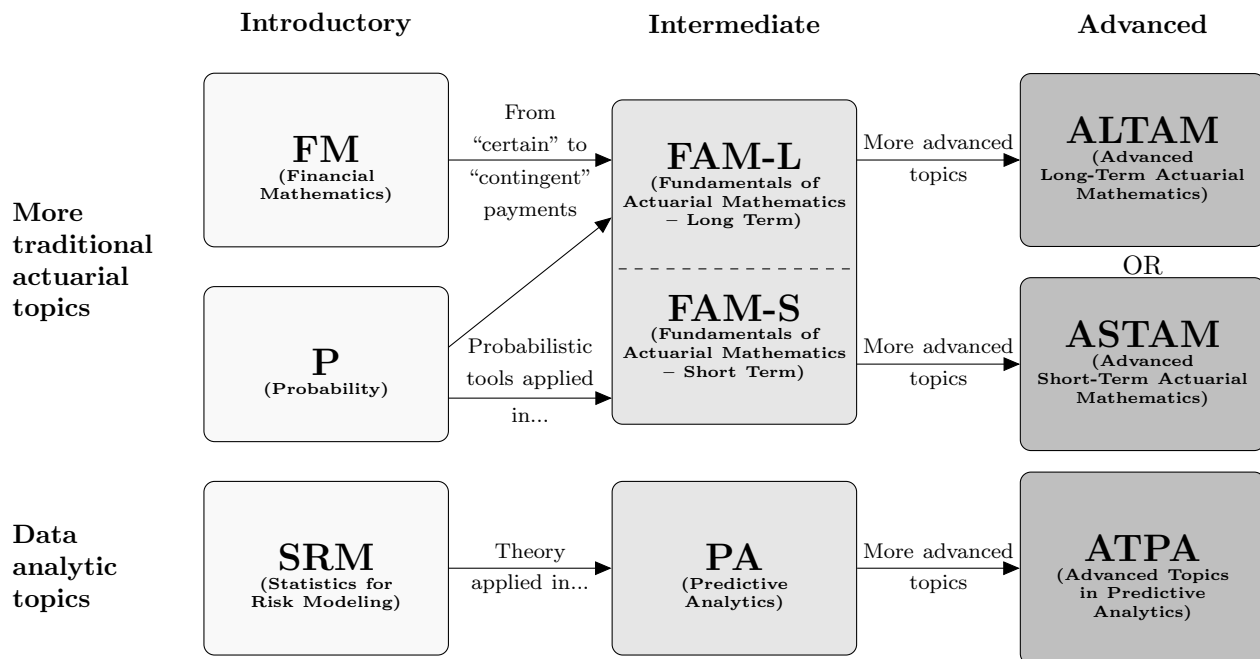
## Predictive Analytics Trio 🌀: SRM, PA, and ATPA

Since 2018, the SOA has redesigned the ASA curriculum to reflect more contemporary and powerful predictive analytic methods that have proved useful in actuarial practice. In the current curriculum, there are a total of 3 exams (or assessments) with a heavy focus on predictive analytics:

- SRM (Statistics for Risk Modeling)
- PA
- ATPA (Advanced Topics in Predictive Analytics)

The flowchart below shows how these 3 exams (and other ASA exams for your information) are related. While there is no set order in which the exams should be taken, students typically attempt exams from left to right, or from introductory, intermediate, to advanced. In the case of the predictive analytics trio, that means taking SRM, PA, and ATPA, in this order.

### Flowchart of ASA Exams Effective from 2022



**SRM vs. PA.** From December 2018 to June 2021, Exam SRM was a formal prerequisite for Exam PA. Although this prerequisite is no longer in place, knowledge of the SRM materials is still assumed. As the PA exam syllabus says,

“Exam PA assumes knowledge of probability, mathematical statistics, and selected analytical techniques as covered in Exam P (Probability), VEE Mathematical Statistics, and Exam SRM (Statistics for Risk Modeling),”

so it is reasonable to prepare for PA at the same time as or shortly after taking SRM, e.g., taking SRM in early September and PA in mid-October, or SRM in early January and PA in mid-April.

In essence, Exams SRM and PA share the same theme of working with *models*, but test it differently. As a precursor, Exam SRM is a traditional multiple-choice exam that serves to provide you with the foundational knowledge behind the modeling process. The emphasis is on the underlying theory, including the uses, motivations, mechanics, pros and cons, do's and don'ts of, and similarities and differences between different predictive analytic techniques. As a natural continuation, Exam PA will have you apply the theory you learned in Exam SRM to a business problem and see first hand how things play out. Although SRM is an important stepping stone to PA and the two exams have a rather big overlap, I would still recommend spending **at least 2 months** 📅 studying for PA intensively, even if you have taken SRM. Here are the reasons:

- (*Different skills tested*) Even though you will not apply mathematical formulas or do calculations by hand as often as in SRM, you will need time to gain hands-on experience with fitting and interpreting predictive models in R, and need practice on communicating your thoughts in writing. The written-answer format of Exam PA means that the SOA can test the material of SRM in greater breadth and depth, and assess your higher-level thinking, e.g., can you describe a certain concept or explain why something is true? You have to know how things work, at least at a conceptual level, and organize your thoughts in words.
- (*Scope*) There are some additional concepts (e.g., exploratory data analysis, elastic nets, performance metrics for classifiers, the elbow method for  $K$ -means clustering) and practical considerations that are tested in PA, but not seen in SRM. You do have to hit the books (or this manual)! 📖

**PA vs. ATPA.** Introduced in January 2022, the ATPA Assessment is a 96-hour take-home computer-based assessment (rather than a proctored exam) that tests additional data and modeling concepts on the basis of those in Exams SRM and PA, and consists of more involved and open-ended tasks than those in PA. As a result, ATPA is preferably taken after passing SRM and PA.

Although ATPA is a take-home assessment and 4 days seem a lot of time, you would be wise not to underestimate the amount of time and effort necessary to master the topics that can be tested, and the workload and pressure that the assessment can create. Unlike PA, which only requires some basic knowledge of R programming, proficiency with R is critical to success in ATPA. During the 96-hour window, you will spend most of your time dealing with various data issues, constructing and evaluating more advanced predictive models than those covered in PA, and finally turning your results into a written report. Make sure that you have set aside enough free time in your schedule 📅 for the next 4 days before you start the assessment. In my experience, you may need more than a day just to clean the data and get it in good shape in R before building any models. You will be busy doing coding 🖥️ and writing! 📝

Note that I have written a separate study manual for ATPA. To learn more, please check out:

<https://www.actexlearning.com/exams/atpa/exam-atpa-study-manual>.

## P.2 About this Study Manual

### What is Special about This Study Manual?

I fully understand that you have an acutely limited amount of study time and that Exam PA, as a written-answer exam with a new format effective from April 2023, is not easy to prepare for. With this in mind, the overriding objective<sup>4</sup> of this study manual is to help you develop a conceptual understanding of and hands-on experience with the materials of Exam PA as effectively and efficiently as possible, so that you will pass the exam on your first try easily, go on to ATPA confidently, and get your ASA ASAP. Here are some unique features of this manual to make this possible.

#### Feature 1: The Coach DID Play!

Usually coaches don't play 😊, but as a study manual author, I took the initiative to write the **December 2019 Exam PA** and the **February-April 2023 ATPA Assessment** to experience first-hand what the real exams were like, despite having been an FSA since 2013 (and technically free from exams thereafter!). I made this decision in the belief that *teaching* an exam and *taking* an exam are rather different activities, and braving the exam myself is the best way to ensure that this manual is indeed useful for exam preparation. If the manual is useful, then at the minimum the author himself can do well, right? I am thrilled that with the help of my own manual...



[Shopping Cart](#) [Section](#) [My Account](#)

You are here: [My Account](#) » [My Transcripts](#) » [Grade Slip](#)

Grade Slip

The scale of grades runs from 0 to 10, passing grades are 6 through 10. A grade of 0 does not mean that the candidate received no credit but that he/she had a very poor paper. Similarly, a grade of 10 indicates a very fine paper but not necessarily a perfect one.

Today's Date: 1/24/2022

**Dec 2019 Predictive Analytics**

Course	Grade
EXAMPA	10

ID: XXXXXXXXXX Candidate ID: 67666

Ambrose Lo FSA,CERA  
Associate Professor  
University of Iowa  
241 Schaeffer Hall  
Iowa City, IA 52242-1409

[Printer Friendly Version](#)

<sup>4</sup>A secondary but still important objective is to let you have some fun along the way. 😊



# Online Services

[Shopping Cart](#)   [Section](#)   [My Account](#)

You are here: [My Account](#) » [My Transcripts](#) » [Grade Slip](#)

## Grade Slip

The scale of grades runs from 0 to 10. passing grades are 6 through 10. A grade of 0 does not mean that the candidate received no credit but that he/she had a very poor paper. Similarly, a grade of 10 indicates a very fine paper but not necessarily a perfect one.

Today's Date: 7/1/2023

**Jun 2023 Advanced Topics in Predictive Analytics**

ID:

Course	Grade
ATPA	11

Ambrose Lo FSA,CERA  
Associate Professor  
University of Iowa  
241 Schaeffer Hall  
Iowa City, IA 52242-1409

If you use this PA study manual, you can rest assured that it is written from an exam taker’s perspective by a professional instructor who has experienced the “pain” of (AT)PA candidates and truly understands their needs. Drawing upon his “real battle experience” and firm grasp of the exam topics, the author will go to great lengths to help you prepare for this challenging exam in the best possible way. You are in capable hands. 👍

### Feature 2: Three-part Structure

To maximize your learning effectiveness and efficiency, I have divided this study manual into three parts:

- **Part I: A Crash Course in R**

The first part of the manual is a crash course in R covering the elements of R programming that are particularly germane to Exam PA to get you up to speed. They include the basics of R programming and data visualization using the `ggplot2` package, covered respectively in Chapters 1 and 2 of the manual. At the completion of this part, you will be equipped with the fundamental R programming skills necessary for constructing predictive models and making some simple but informative graphs in the rest of this manual.





- **Part II: Theory of and Case Studies in Predictive Analytics**

Armed with R basics, you will learn the theory of different types of predictive analytic techniques illustrated by a series of case studies in the second part (Chapters 3 to 6), also the linchpin, of this manual. Each chapter in this part follows the same arrangement:

- ▷ *Theory*: Each chapter begins with a conceptual foundations section describing the mechanics of various predictive analytic techniques, including linear models (Chapter 3), generalized linear models (Chapter 4), decision trees (Chapter 5), and principal components and cluster analyses (Chapter 6). The explanations in these sections are thorough, but *exam-focused* and *learning-oriented*. Instead of showing unnecessary technicalities that add little value to your preparation for PA (predictive analytics can be a very mathematical subject!), I strive to follow the SOA’s PA modules very closely and cover enough (but just enough) ground for you to understand predictive analytics at the level required by Exam PA.
- ▷ *Practice*: After learning the ins and outs, pros and cons, and do’s and don’ts of these techniques, we will turn to their practical implementations and gain some hands-on experience through a number of task-based case studies using R. Do read these case studies carefully as they illustrate a wide range of skills necessary for tackling various types of tasks in Exam PA, ranging from data pre-processing, data exploration, model construction, model evaluation, and model selection.

### • Part III: Final Preparation

Last but not least, the third part concludes this manual with the following resources:

- ▷ *Chapter 7*: This chapter includes my commentary on the SOA’s past PA exams, which reflect the SOA’s expectations of PA candidates and are quite indicative of future exams
- ▷ *Chapter 8*: This chapter presents two original full-length practice exams updated for the new exam format and designed to mimic the real PA exam in terms of style and difficulty, with detailed illustrative solutions provided
- ▷ *Appendix A*: This appendix categorizes all relevant exam tasks since June 2019 (i.e., all past exams following a task-based format) by topic. A cursory glance  at this appendix can reveal the themes that consistently emerge in past exams, and you may take advantage of it to identify relevant exam questions on a certain topic and make your learning more focused.
- ▷ A downloadable  and printable  cheat sheet, available as a separate file, provides a “helicopter”  view of the entire PA exam, and is useful for both regular review and last-minute exam preparation. The cheat sheet can be accessed from

<https://www.actexlearning.com/formula-and-review-sheets>.

After completing Part III, you will be ready to take (and pass!) the October 2024 PA exam.



## Other Features

This manual throughout is also characterized by the following features that make your learning as smooth as possible:

- Each chapter in Parts I and II starts by explicitly stating which learning objectives and outcomes of the PA exam syllabus we are going to cover, to assure you that we are on track and hitting the right target.
- Objects in R are shown in `typewriter` font and code chunks with output in gray boxes for aesthetic reasons. (PA exam questions may show R objects in `typewriter` font or in **bold**.)

```
...LOTS OF R CODE HERE...
...LOTS OF R CODE HERE...
...LOTS OF R CODE HERE...
```

Formulas, functions, and commands that are of great importance are boxed to aid identification and retention.


- Important exam items and common mistakes committed by students are highlighted by boxes that look like:

**⚠ EXAM NOTE ⚠**

Be sure to pay special attention to boxes like this!

- The main text of this manual is interspersed with more than 110 exercises, all with complete solutions, to assess your understanding regularly. Some of these exercises are based on recent SOA and CAS exams, but many are original. (If you have used the *ACTEX Study Manual for Exam SRM*, you may have seen some of these past exam questions in some form, but I have rewritten many of them in the language and style of Exam PA. There is also no harm in giving them a second look!) These examples are instrumental in illustrating a number of conceptual items that can be tested in Exam PA.
- Each chapter in Part II concludes with a number of conceptual review questions designed to help you look back on the most important conceptual issues in that chapter. Solutions to these questions can be found in the main text, indicated by marginal labels such as the **Q3.1** one on the right.

## Supplementary Files

This study manual comes with a number of supplementary files (e.g., R Markdown files with completely reproducible R code, datasets, and files to be released) that can be downloaded from [Actuarial University](#).  All users of the manual (whether it is the printed or digital version) will receive by email a keycode that provides electronic access to all supplementary files shortly after their order is placed. If you can't retrieve that email (be sure to check your junk/spam folders), please reach out to [support@actexlearning.com](mailto:support@actexlearning.com) for assistance.

It is a good idea (but not absolutely essential, given the new exam format) to run the R Markdown

files as you work through this manual, making sure that your output agrees with what is shown here. This is especially important if you have ordered a printed copy of this study manual—run the code to see the beautiful colors! ☺

**⚠ NOTE ⚠**

Commentary on the April 2024 PA exam will be available on [Actuarial University](#) shortly after the SOA posts the exam with solutions [online](#).



## What is new in the 11th edition of the Manual?

Albeit relatively established, this manual is periodically “re-trained” taking the latest PA exams into account to improve its “predictive power.” There have been updates throughout the manual, now in its 11th edition, in terms of clarity, substance, and exam focus (some in response to student questions), but the following have seen the most significant improvements:

- Chapter 1: This chapter, which provides a minimal introduction to R programming, has been trimmed. Some of the more advanced and less essential R functions and knowledge are removed and deferred to ATPA.
- Subsection 2.2.1: The skewness of a distribution on page 93 (now accompanied by some illustrative graphs)
- Subsection 3.1.2: Problem definition on page 136, stratified sampling on page 140, and target leakage on pages 146-147
- Subsection 3.2.5: Standardization of predictors on page 239
- Subsection 3.3.3: Task 6 (a)-(b) on pages 272-274
- Subsection 6.2.2: How to cut the dendrogram on page 652 (the ambiguous situation when the cut exactly coincides with an inter-cluster dissimilarity is now addressed) and recommending the number of clusters to use on pages 672-673
- New in-text exercises: 3.2.18 (rewritten), 5.1.14, 5.1.16 (b), and 6.2.6 (the right dendrogram has been fixed)
- New end-of-chapter conceptual review questions: 3.5, 3.21, 3.33, 5.7, and 6.19
- Section 7.1: Commentary on the April 2024 exam (to be released shortly)
- Appendix A: Questions of the April 2024 exam have been added and categorized.

## Two Add-ons

If you have purchased this manual and are interested in upgrading your manual to include any of the following add-ons, please email Customer Service at [support@actexlearning.com](mailto:support@actexlearning.com).

**Instructional videos.** 🎥 Instructional videos (<https://www.actexlearning.com/exams/pa/exam-pa-study-manual>) accompanying the core of this manual (Parts I and II, or Chapters 1 to 6) are available for purchase as an add-on. In these videos, I (Ambrose) will walk you 🚶 through the fundamental concepts in predictive analytics and the construction of predictive models in R step by step, with a strong emphasis on key test items in Exam PA. With the aid of visuals, these videos aim to make the materials in the manual as accessible as possible and will add substantial value to your learning.

When it comes to learning strategies, some students find it useful to watch the videos to get the “big picture,” then read the manual to learn the details. Alternatively, you may first read the manual, then watch the videos to consolidate your understanding. Both modes of learning are fine and which one is better depends entirely on your preferences.

**Graded mock exam.** 📄 In addition to the two practice exams in Chapter 8 of this manual, we offer a separate mock exam (<https://www.actexlearning.com/exams/pa/exam-pa-mock-exam>), with completely different questions, and an optional 1:1 live feedback session.

A common “complaint” against Exam PA is that its written and somewhat open-ended exam format makes it difficult for students to evaluate their work even after reading the SOA’s model solutions to past exams, e.g., if you write this, how many points can you expect to get? How to improve your answers? This is precisely why we create this mock exam with grading service, which provides a valuable opportunity for you to assess your overall understanding of the PA exam syllabus and, more importantly, have your work graded from a **critical eye** 👁, and receive **personalized feedback** 🗨 (not generated by AI in any way!). You will work on the mock exam under simulated exam conditions and submit your solutions to us. Having taken PA in the past and now teaching for PA, we (Ambrose and his team) will then grade your work from start to finish, with a score out of 70, and offer specific feedback that will help you enhance the quality of your write-up and improve your performance on the real exam.

### ⚠ NOTE ⚠

Based on the SOA’s publicly posted [pass lists](#), **86%** of the students who attempted the October 2023 and April 2023 editions of the graded mock exam passed the real PA exam. 👍 In comparison, the pass rate for the October 2023 Exam PA was 63%.

For the October 2024 sitting, the graded mock exam (currently available for pre-order) is expected to be released on *Actuarial University* in mid-August and the last day of submission is September 30 (Monday). Within 2 weeks of your submission, you will receive by email: ✉

- (1) Your graded mock exam with personalized (and possibly critical!) feedback

- (2) Detailed illustrative solutions to the mock exam along with grading rubric

## Announcements




As time goes by, I may post news and announcements (e.g., new files becoming available) about this study manual and Exam PA on my personal web page:

<https://sites.google.com/site/ambroseloy/publications/PA>

A list of errata (if any) will also be maintained. I would greatly appreciate it if you could bring any potential errors, typographical or otherwise, to my attention via email ([amblo201011@gmail.com](mailto:amblo201011@gmail.com)) so that they can be fixed in a future edition of the manual.

## Contact Us

If you encounter problems with your learning, we always stand ready to help.

- ✉ For **technical issues**  (e.g., not able to access, download, or print supplementary files from *Actuarial University*, extending your digital license, upgrading your product, exercising the Pass Guarantee), please email ACTEX Learning's Customer Service at [support@actexlearning.com](mailto:support@actexlearning.com). The list of FAQs available on <https://www.actuarialuniversity.com/help/faq> may also be useful.
-  For questions related to **specific contents** of this manual and Exam PA, including potential errors, please feel free to raise them in the PA community on ACTEX's Discord channel, which provides a convenient platform for you to network with other PA students, and I will strive to respond to your questions ASAP.  Please note:
  - ▷ Remember to check out the errata list on my [personal web page](#). It may happen that the errors you discover have already been addressed.
  - ▷ Instead of saying
 

“You mention (somewhere) in your manual that...”

 it would be great to quote the specific page(s) of the manual your questions are about. This will provide a concrete context and make our discussion much more fruitful.
  - ▷ (*Less important in the new exam format*) If you experience issues with R, e.g., your code can't run and you keep seeing weird error messages, please provide the version of R (not RStudio!) you are using and a screenshot of the error messages.

## Acknowledgments

I am grateful to Mr. Tony Pistilli for proofreading an early version of this study manual and many past students for taking the time to send me comments and suggestions, which have improved the quality of the manual in no small measure. All errors that remain are solely mine.

## About the Author

**Ambrose Lo**, PhD, FSA, CERA, is the author of several study manuals for professional actuarial examinations and an Adjunct Associate Professor at the Department of Statistics and Actuarial Science, the University of Hong Kong (HKU). He earned his BSc in Actuarial Science (first class honors) and PhD in Actuarial Science from HKU in 2010 and 2014, respectively, and attained his Fellowship of the Society of Actuaries (FSA) in 2013. He joined the Department of Statistics and Actuarial Science, the University of Iowa (UI) as Assistant Professor of Actuarial Science in August 2014, and was promoted to Associate Professor with tenure in July 2019. His research interests lie in dependence structures, quantitative risk management as well as optimal (re)insurance. His research papers have been published in top-tier actuarial journals, such as *ASTIN Bulletin: The Journal of the International Actuarial Association*, *Insurance: Mathematics and Economics*, and *Scandinavian Actuarial Journal*. He left the UI and returned to Hong Kong in July 2023.

Besides dedicating himself to actuarial research, Ambrose attaches equal (if not more!) importance to teaching and education, through which he nurtures the next generation of actuaries and serves the actuarial profession. He has taught courses on a wide range of actuarial science topics, such as financial derivatives, mathematics of finance, life contingencies, and statistics for risk modeling. He is also the (co)author of the *ACTEX Study Manuals for Exams ATPA, MAS-I, MAS-II, PA, and SRM*, a *Study Manual for Exam FAM*, and the textbook *Derivative Pricing: A Problem-Based Primer* (2018) published by Chapman & Hall/CRC Press. Although helping students pass actuarial exams is an important goal of his teaching, inculcating students with a thorough understanding of the subject and logical reasoning is always his top priority. In recognition of his outstanding teaching, Ambrose has received a number of awards and honors ever since he was a graduate student, including the [2012 Excellent Teaching Assistant Award](#) from the Faculty of Science, HKU, public recognition in the *Daily Iowan* as a faculty member “making a positive difference in students’ lives during their time at UI” for nine years in a row (2016 to 2024), and the 2019-2020 Collegiate Teaching Award from the UI College of Liberal Arts and Sciences.



**Part I**

**A Crash Course in R**





# Chapter 2

## Data Exploration and Visualization

**\*\*\*FROM THE PA EXAM SYLLABUS\*\*\***

### **2. Topic: Data Exploration and Visualization (20-30%)**

#### **Learning Objectives**

The Candidate will be able to work with various data types, understand principles of data design, and construct a variety of common visualizations for exploring data.

#### **Learning Outcomes**

The Candidate will be able to:

- d) Apply the key principles of constructing graphs.
- e) Apply univariate data exploration techniques.
- f) Apply bivariate data exploration techniques.

*Chapter overview:* An integral part of any predictive analytic exercise is the use of graphical displays to investigate the characteristics of the variables of interest, on their own and in relation to one another, and to visualize the results of the predictive models constructed. In this regard, one of the key strengths of R as a programming language is that it offers versatile graphing capabilities, both in the base installation and with add-on packages. With a minimal amount of code, we can produce a wide variety of high-quality graphs. In Exam PA, you will be asked to take advantage of R's graphing capabilities and make sense of different types of graphical displays [\[1\]](#). Instead of using R's base graphical platform, you will make graphs using the `ggplot2` package,<sup>1</sup> which may be new to you even if you have used R before. Compared to R's base graphics system, `ggplot2` involves vastly different syntax based on the so-called "grammar of graphics" (in fact, "gg" stands for "grammar of graphics") and lends itself to producing sophisticated graphs that

---

<sup>1</sup>The `ggplot2` package is developed by Hadley Wickham, who is also a core developer of RStudio. Earlier the package was called `ggplot`, but substantial changes were made later, so the name of the package was upgraded to `ggplot2`.

would be cumbersome to create using base R graphics.

Synthesizing the material in the first four chapters of the book *Data Visualization: A Practical Introduction* (which is listed in the exam syllabus), this chapter presents some of the important graphical functions in the `ggplot2` package most relevant to Exam PA. These functions can be used to construct different types of graphs such as scatterplots, histograms, boxplots, and bar charts, which will all be illustrated in the context of a real insurance dataset. In Section 2.1, we will learn the basic structure of a ggplot, make some simple but informative plots, and learn how to tweak the appearance of a ggplot. Section 2.2 draws upon the data visualization techniques covered in Section 2.1 to perform exploratory data analysis, which is the use of graphs and summary statistics to uncover patterns and relationships in a set of data, and generate hypotheses which can be answered quantitatively in a predictive model at a later stage.

## 2.1 Making ggplots

### 2.1.1 Basic Features

Let's begin by installing (make sure to install a package the first time you use it!) and loading the `ggplot2` package.

```
# CHUNK 1
# Uncomment the next line the first time you use ggplot2
# install.packages("ggplot2")
library(ggplot2)
```

With the last command, we can use all the functions in the `ggplot2` package until the end of the current R session.


**Skeleton.** In its simplest form, a ggplot consists of two parts: The core `ggplot()` function (not `ggplot2()`!) and a chain of additional functions pasted together using the plus (+) sign defining the exact type of plot to be made.

- (1) *ggplot() function:* The `ggplot()` function initializes the plot, defines the source of data using the `data` argument (almost always a **data frame** in Exam PA), and, most importantly, specifies what variables in the data are “mapped” to visual elements in the plot by the `mapping` argument. Mappings in a ggplot are specified using the `aes()` function, with `aes` standing for “aesthetics.” They determine the role different variables play in the plot. The variables may, for instance, correspond to visual elements such as the x- or y-variables, color, size, and shape, specified by the `x`, `y`, `color`, `size`, and `shape` aesthetics, respectively.
- (2) *Geom functions:* Subsequent to the `ggplot()` function, we put in *geometric objects*, or *geoms* for short, which include points, lines, bars, histograms, boxplots, and many other possibilities, by means of one or more *geom functions*. Placed layer by layer, these geoms determine what kind of plot is to be drawn and modify its visual characteristics, taking the data and aesthetic mappings specified in the `ggplot()` function as inputs.

Here is the generic structure of a ggplot: (The uppercase letters are placeholders.)

```
ggplot(data = DATA, mapping = aes(AESTHETIC_1 = VARIABLE_1,
                                   AESTHETIC_2 = VARIABLE_2,
                                   ...)) +
  geom_TYPE(...) +
  geom_TYPE(...) +
  OTHER_FUNCTIONS +
  ...
```

Don't worry if the ideas above seem puzzling at this stage. It is commonly acknowledged that the learning curve of ggplots is steep, much more so than R's base graphics system, but taking some time to learn how to make ggplots will pay dividends not only in Exam PA, but also in your real work. You will gain a much better understanding of how a ggplot works after going through the example plots in this chapter and in the rest of this study manual.

**Case study: Personal injury insurance dataset.** To illustrate data visualization and exploration techniques, in this chapter we will look at a personal injury insurance dataset.<sup>2</sup>  This dataset contains the information of 22,036 settled personal injury insurance claims. These claims were reported during the period from July 1989 to the end of 1999, with claims settled with zero payment excluded. The variables in the dataset are described in Table 2.1.



Variable	Description
amt	settled claim amount (continuous numeric variable)
inj	injury code, with seven levels: 1 (no injury), 2, 3, 4, 5, 6 (fatal  ) , 9 (not recorded)
legrep	legal representation (0 = no, 1 = yes)
op_time	operational time (a standardized amount of time elapsed between the time when the injury was reported and the time when the claim was settled)

Table 2.1: Data dictionary for the personal injury (`persinj`) insurance claims dataset.

In Section 4.2, we will build a model to predict the size of personal injury insurance claims using other variables in the dataset. For now, we will perform data exploration of the variables in the dataset. The insights we gain here will go a long way towards constructing a good predictive model.

<sup>2</sup>This dataset is a pre-processed version of the `ausautoBI8999` data in the `CASdatasets` package. This dataset also accompanies the textbook *Generalized Linear Models for Insurance Data* (2008), by de Jong and Heller.

- To get started, let's run CHUNK 2 to load the external CSV file containing the dataset into R as a data frame called `persinj` (meaning “personal injury”) using the `read.csv()` function, which takes the name of the CSV file  supplied as a character string as an argument. In this preliminary section, we will take out a subset of 50 observations from the `persinj` data, called `persinj50`, and explain the main characteristics of a ggplot using these 50 observations. This will help us appreciate the different types of visual effects that can be produced on a ggplot more easily. In Section 2.2, we will return to the full dataset and learn why we use a certain plot for a certain purpose.

### ⚠ ONE MORE REMINDER! ⚠

Please read page xxv of the preface of this manual about how to access the Rmd files as well as datasets that go with this manual. There is no need to transfer the R code in the manual to your RStudio line by line!

```
# CHUNK 2
persinj <- read.csv("persinj.csv")
# Take out a subset of 50 observations from the full data
persinj50 <- persinj[seq(1, nrow(persinj), length = 50), ]
```

- First encounter with ggplots.** As our first example, let's make a *scatterplot* for the two numeric variables in the `persinj50` data, `amt` and `op_time`. The plot, produced by the code in CHUNK 3, is given in Figure 2.1.1. The code obeys the two-part structure discussed earlier:

- *ggplot() function:* The first line of the code makes it clear that we are using the `persinj50` data, where the variables `op_time` and `amt` are mapped to the variables on the x-axis and y-axis through the `x` and `y` aesthetics, respectively. There is no need to name the variables as `persinj50$op_time` or `persinj50$amt` as the data source is already specified in the data argument.
- *Geom:* Given these mappings, we use `geom_point()` to make a scatterplot of `amt` (the y-variable) against `op_time` (the x-variable). The plot comprises 50 points ● (hence the name of the function) corresponding to the 50 paired values of the two variables and allows us to see the two variables in comparison with each other.

Later, we will fine-tune this plot in different ways to capture different sorts of information.

```
# CHUNK 3
ggplot(data = persin50, mapping = aes(x = op_time, y = amt)) +
  geom_point()
```

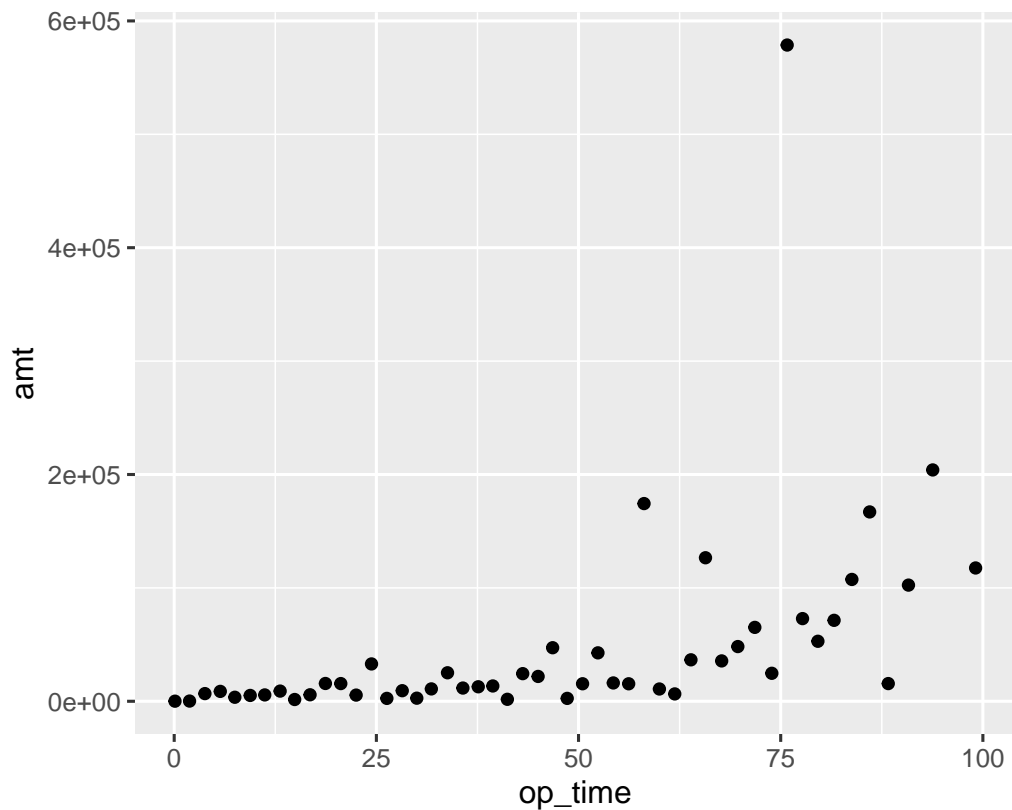


Figure 2.1.1: A basic scatterplot of `amt` against `op_time` in the `persin50` dataset.

**Using aesthetics the right way: The essence of aesthetic mappings.** One of the most common ways to modify the appearance of a plot is to color the observations in order to produce a more impressive visual effect. To color the data points say, in `blue`, you may be tempted to make use of the `color`<sup>3</sup> aesthetic and simply insert `color = "blue"` as an additional argument to the `aes()` function, as in CHUNK 4. Doing so will produce unexpected and undesirable results as shown in Figure 2.1.2. To your astonishment, all of the data points are colored in `red` instead of `blue` and there is a legend saying “blue.” What has gone awry here?

<sup>3</sup>Both the American spelling (`color`) and British spelling (`colour`) are accepted.

```
# CHUNK 4
# It is OK to suppress the names of the data and mapping arguments
# so long as they are supplied in order
ggplot(persinj50, aes(x = op_time, y = amt, color = "blue")) +
  geom_point()
```

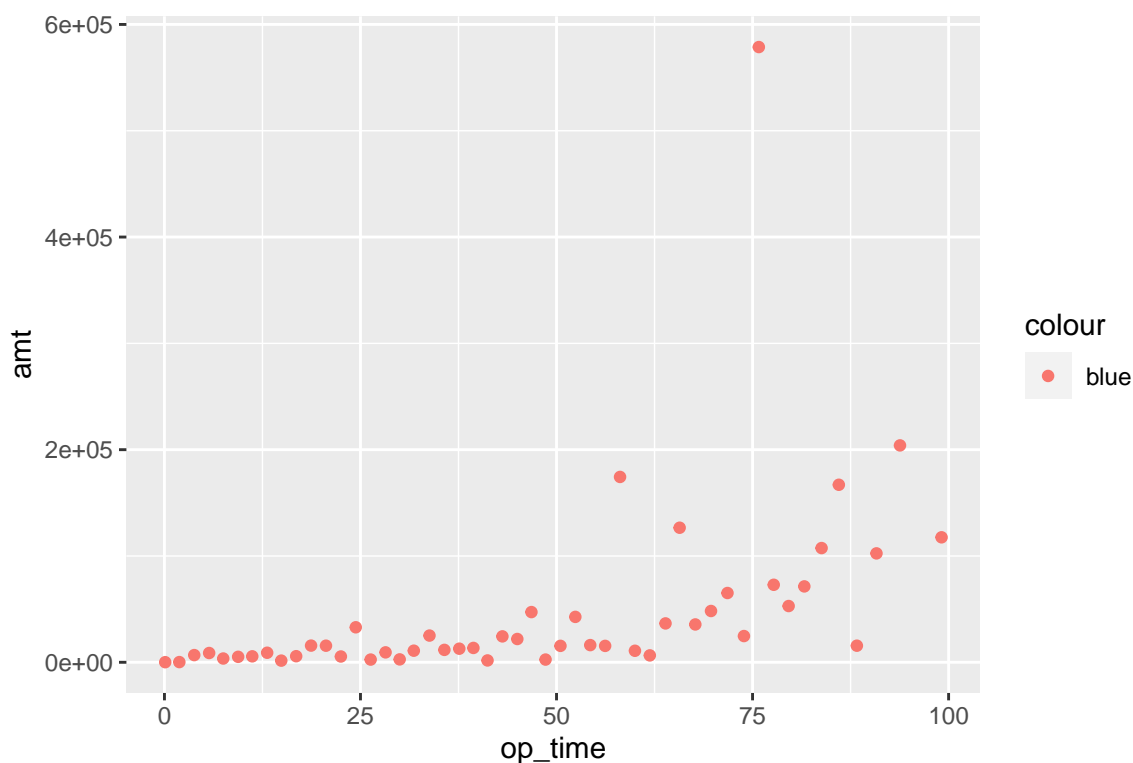


Figure 2.1.2: A version of Figure 2.1.1 with all of the points inadvertently colored in red.

Bear in mind that an aesthetic is a mapping between variables in our data and visual properties of the graph. The use of `color = "blue"` instructs the `aes()` function to map the `color` aesthetic to a variable named "blue" in the `persinj50` dataset. There is no such variable in our data, but the `aes()` function will do its best by treating "blue" as if it were a variable. The effect is the creation of a new character variable behind the scenes taking one and only one value, "blue". As all observations in the data share the same "blue" value, all of them will be mapped to the same color. In `ggplot2`, the default first-category hue is red (not blue!). This explains why every point in the scatterplot becomes red in color.

To do the coloring the right way, we should realize that making all the points blue in color does *not involve any mapping* between variables in our data and the `color` aesthetic. After all, all the observations are colored in blue and are not distinguished on the basis of color. As a result, we should not put `color = "blue"` inside the `aes()` function. It should instead be placed inside the `geom_point()` function to modify the color of the plotted points. Figure 2.1.3 shows the desired scatterplot using the code in CHUNK 5. To our liking, all of the points are colored in blue.

```
# CHUNK 5
ggplot(persinj50, aes(x = op_time, y = amt)) +
  geom_point(color = "blue")
```

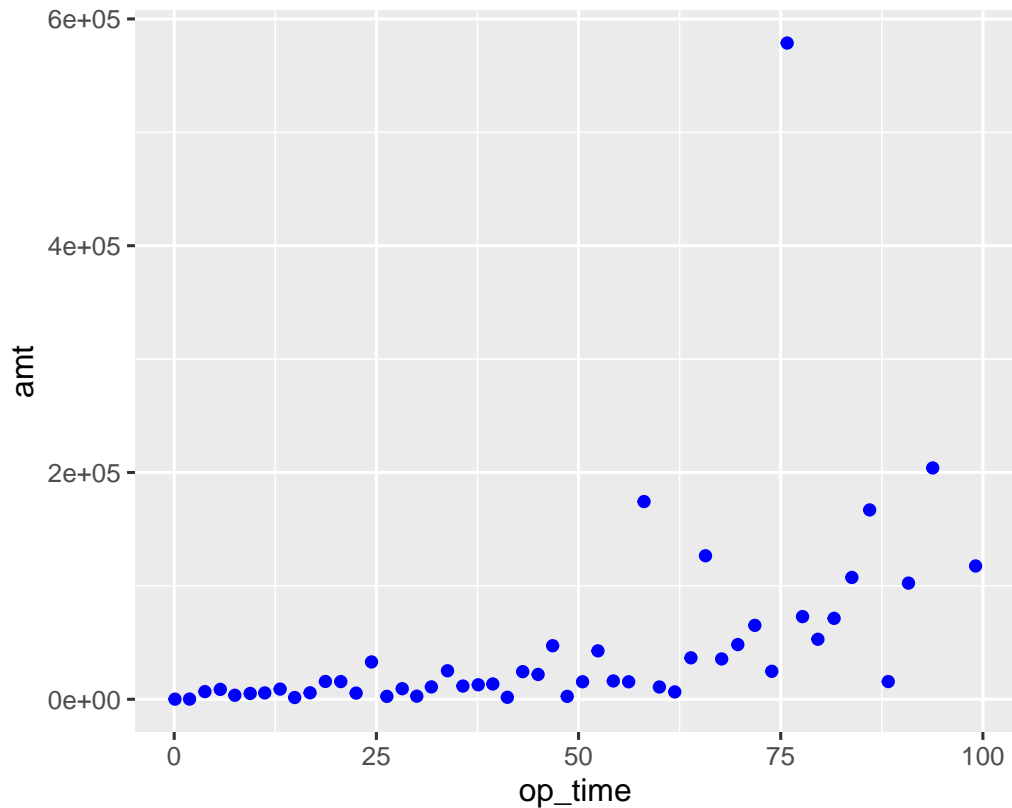


Figure 2.1.3: A version of Figure 2.1.1 with all of the points correctly colored in blue.

Figures 2.1.2 and 2.1.3 highlight a subtle but extremely important mentality when making ggplots:

The `aes()` function is reserved for mappings between aesthetics and *variables*.

To set a property that affects how a plot looks but does not involve mapping variables to aesthetic elements, we should do it outside the `aes()` function—in the geom functions. To put it another way, the aesthetics determine *what* relationships we want to see in the plot whereas the geoms determine *how* we want to see the relationships.

Now let's see an example of using the `color` aesthetic correctly. In the `persinj50` data, the `legrep` variable is a binary variable equal to 1 for injuries with legal representation and 0 for those without. To color the different injuries according to the presence of legal representation, we map the `color` aesthetic to `legrep` treated as a factor (recall that factors are discussed on page 21). The resulting scatterplot, generated by the code in CHUNK 6, is given in Figure 2.1.4, where injuries without legal representation (`legrep = 0`) are displayed in red whereas those with legal representation (`legrep = 1`) are displayed in teal. A legend is produced accordingly. Notice that there is a genuine mapping between the `legrep` variable and `color`, with `legrep`

= 0 mapped to the **red** color and **legrep** = 1 mapped to the **teal** color. In other words, the observations are differentiated on the basis of the **legrep** variable by color. (The **color** aesthetic does not say what colors are used to discriminate injuries on the basis of **legrep**, though.)

```
# CHUNK 6
ggplot(persinj50, aes(x = op_time, y = amt, color = factor(legrep))) +
  geom_point()
```

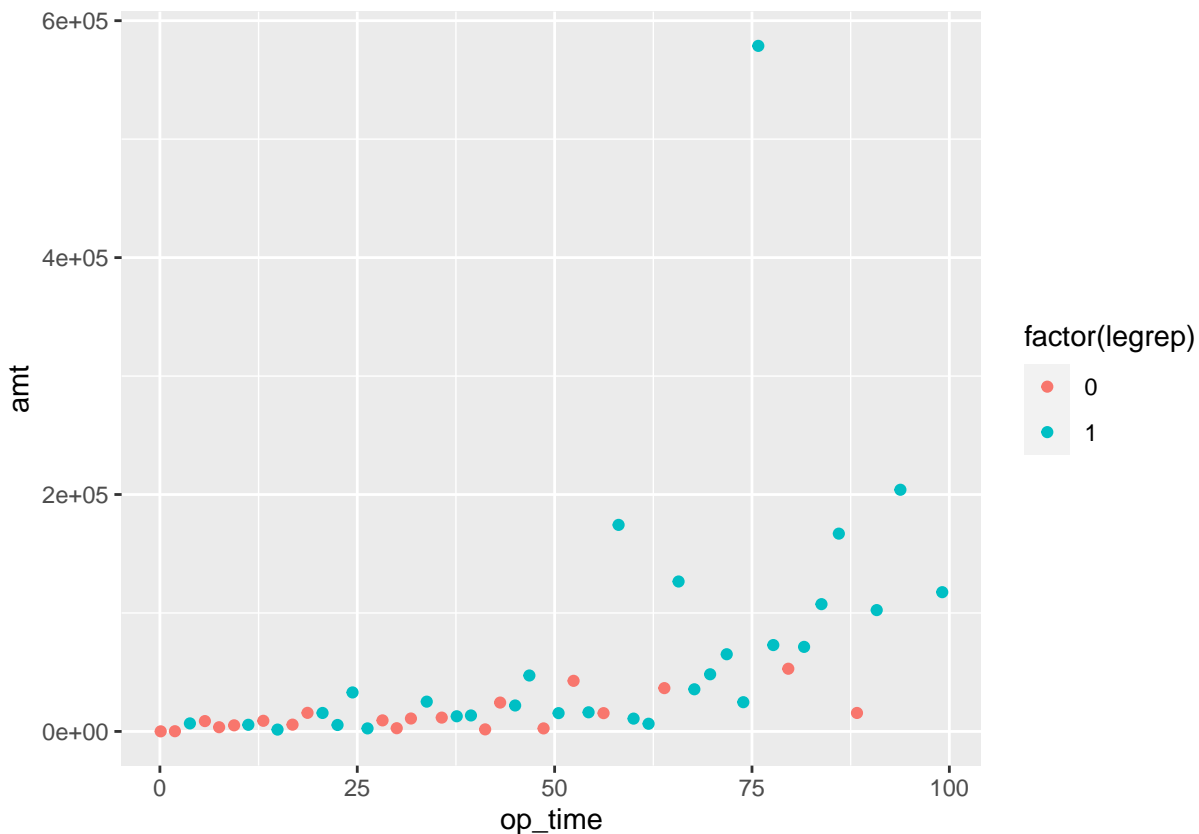


Figure 2.1.4: A version of Figure 2.1.1 with the observations distinguished by **legrep**.

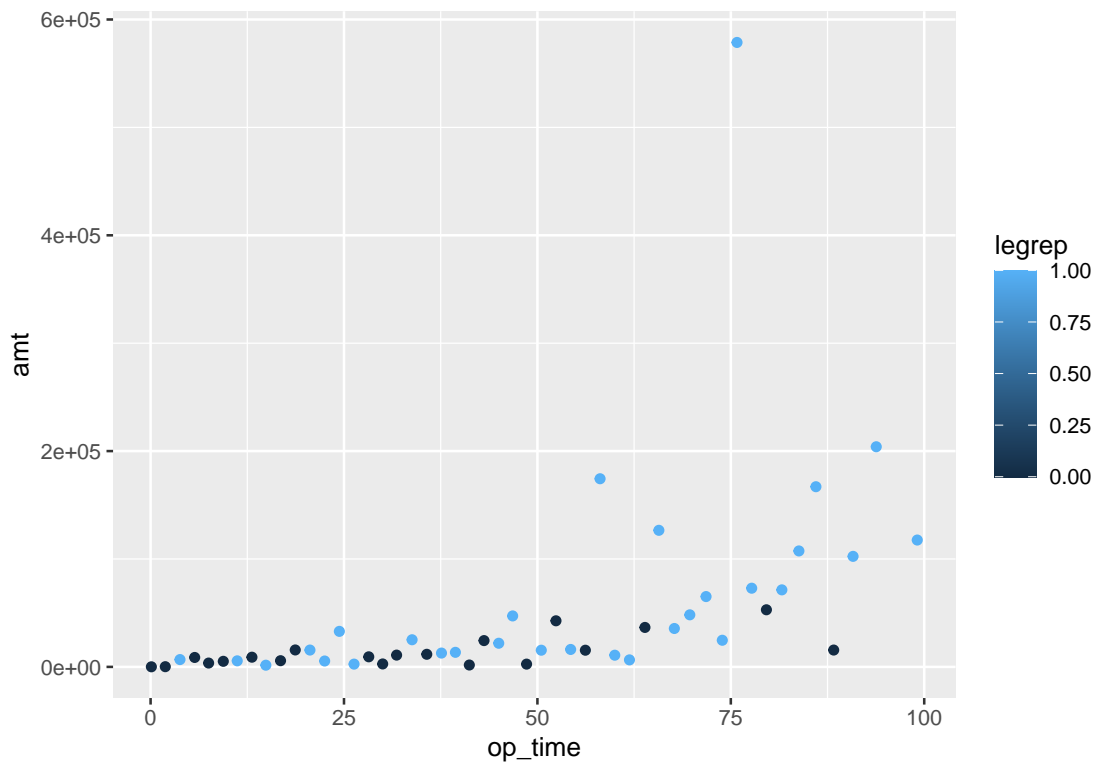
**Exercise 2.1.1.** 🧠 (Why do we need to convert **legrep** to a factor?) To see why the conversion of **legrep** to a factor variable is needed, run the following code in CHUNK 7:

```
# CHUNK 7
ggplot(persinj50, aes(x = op_time, y = amt, color = legrep)) +
  geom_point()
```

What do you notice? Why is the coloring done in the way you observe?

*Solution.* Running the code produces the following scatterplot with a color gradient from 0 to 1:





This is because the `legrep` variable in the `persinj50` dataset is treated by design as a continuous numeric variable even though the two levels, 0 and 1, are merely class labels that do not convey any sense of numeric order. R implicitly allows for values between 0 and 1 for the `legrep` variable and therefore uses the color gradient to differentiate the observations by color (though you can observe that there are only two colors in the plot, corresponding to the two extremes in the color gradient). This explains the point made in Subsection 1.2.1 that whether to treat a categorical variable as a factor can affect how the resulting graphical output looks, sometimes materially.  $\square$

**Some important geoms for Exam PA.** Besides `geom_point()`, there are a number of geoms that are important for Exam PA. They are listed below along with their commonly used arguments which affect how the plot looks. (No need to memorize the table entries!)

Geom	Type of Object Produced	Frequently Used Arguments
<code>geom_bar()</code>	Bar chart	<code>fill</code> , <code>alpha</code>
<code>geom_boxplot()</code>	Boxplot	<code>fill</code> , <code>alpha</code>
<code>geom_histogram()</code>	Histogram	<code>fill</code> , <code>alpha</code> , <code>bins</code>
<code>geom_point()</code>	Scatterplot	<code>color</code> , <code>alpha</code> , <code>shape</code> , <code>size</code>
<code>geom_smooth()</code>	Smoothed curve	<code>color</code> , <code>fill</code> , <code>method</code> , <code>se</code>

The names of these geoms are pretty self-explanatory. For example, `geom_smooth()`, as its name suggests, produces a “smoothed” curve and, by default, produces a ribbon around the curve showing the standard error bands. It is typically used in conjunction with `geom_point()`. The smoothed curve can be generated by different statistical methods, such as a linear fit (by setting `method = "lm"`). The default is the use of nonparametric smoothing methods (`method = "gam"` or `method = "loess"`), which are beyond the syllabus of Exam PA. To switch off the standard error bands, you can set `se = FALSE`.

We will illustrate the use of `geom_bar()`, `geom_boxplot()`, and `geom_histogram()` in Section 2.2. For now, let’s continue with the scatterplots we have just produced and make them more fancy and informative. In Figure 2.1.5, we plot the 50 observations in the `persinj50` dataset using large points (`size = 2`) and a small amount of transparency (`alpha = 0.5`), classify them according to whether they have legal representation or not, and fit a separate smoothed curve to each kind of injuries via the `geom_smooth()` function. The commands are collected in CHUNK 8.

Note that:

- For each of the two types of injuries, the smoothed curve and the standard error ribbon are indicated by the same color (red for those without legal representation and teal for those with legal representation), which is appealing from an aesthetic perspective. The consistent coloring is achieved by mapping both the `color` aesthetic and `fill` aesthetic (which controls filled areas of bars, polygons and, in this case, the interior of standard error bands) to the `legrep` variable treated as a factor variable.
- If you omit `fill = factor(legrep)` (try this in R!), then the two smoothed curves will still be colored according to the presence of legal representation due to the `color` aesthetic, but without the `fill` aesthetic, the `geom_smooth()` function will shade the two standard error ribbons by its default color, which is gray.
- The `alpha` argument controls the transparency of the plotted objects on a scale from 0 (fully transparent) to 1 (opaque); the default value is 1. The transparency of the objects increases by decreasing `alpha`; the lower the value of `alpha`, the more transparent the points. In the limit when `alpha` is exactly zero, the objects become completely invisible. The `alpha` argument is particularly useful when there is a lot of overlapping among the data points. By setting `alpha` to an intermediate value such as 0.5, we make it easy to see where most of the observations cluster.

```
# CHUNK 8
ggplot(persinj50, aes(x = op_time, y = amt,
                     color = factor(legrep), fill = factor(legrep))) +
  geom_point(size = 2, alpha = 0.5) +
  geom_smooth()
```

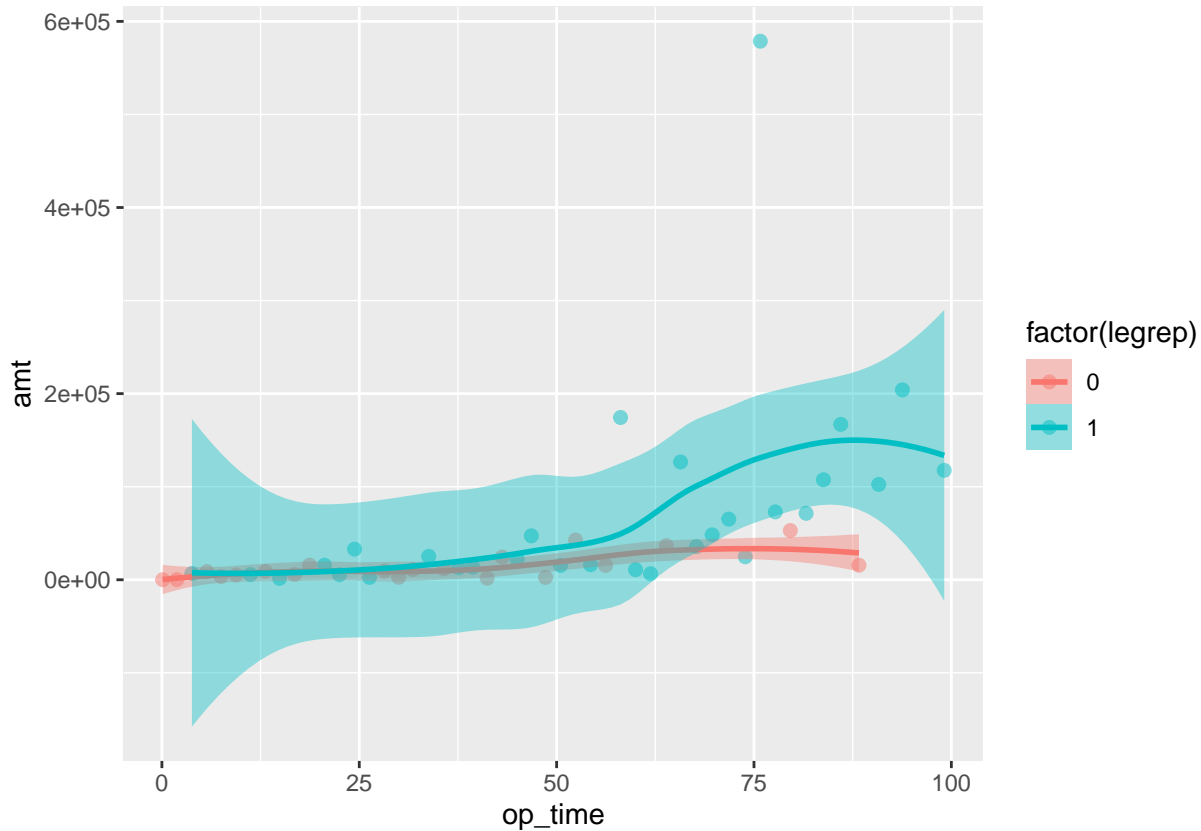


Figure 2.1.5: A version of Figure 2.1.4 with the data points enlarged and the standard error bands added.

Thanks to Figure 2.1.5, we can see that the claim amount of the two types of injuries behaves quite differently with respect to `op_time`, with those for legal representation being much more sensitive to changes in `op_time`. In Section 4.2, we will use a predictive model to quantify the difference between the two forms of behavior formally. Figure 2.1.5, produced by `ggplot2`, allows us to discover such a phenomenon in the first place and is a very useful starting point for such an investigation.

**Geom-specific aesthetics.** What if you want to make just one smoothed curve applied to all 50 observations in the `persinj50` dataset while still having them colored according to the presence of legal representation? We can do so by specifying different aesthetic mappings for different geoms as in CHUNK 9.

```
# CHUNK 9
ggplot(persinj50, aes(x = op_time, y = amt)) +
  geom_point(aes(color = factor(legrep)), size = 2, alpha = 0.5) +
  geom_smooth()
```

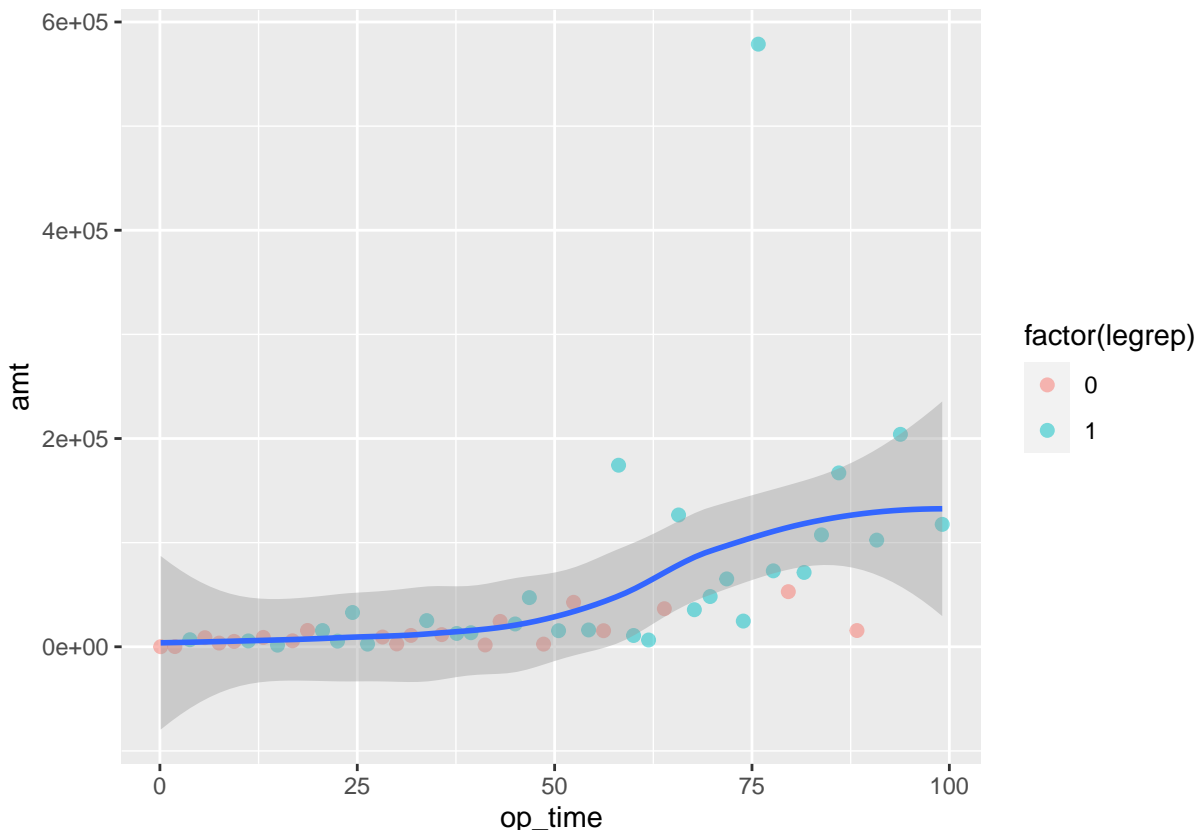



Figure 2.1.6: A variation of Figure 2.1.5 with a single standard error band.

Notice that the `aes()` function in the `ggplot()` call only has the `x` and `y` aesthetics; the `color` aesthetic is moved to the `geom_point()` function. As a result, the 50 points will be distinguished in color by the `legrep` variable. However, there is no such mapping in the `geom_smooth()` function, so a single smoothed curve (colored in blue by default) fitted to all of the 50 observations surrounded by two standard error bands (colored in gray by default) will be produced as shown in Figure 2.1.6. In general, aesthetic mappings common to most, if not all, geoms can be specified in the initial `ggplot()` call. These mappings will be inherited by all geoms. If needed, you can then put in additional aesthetics that apply only to a particular geom to override the default aesthetics.

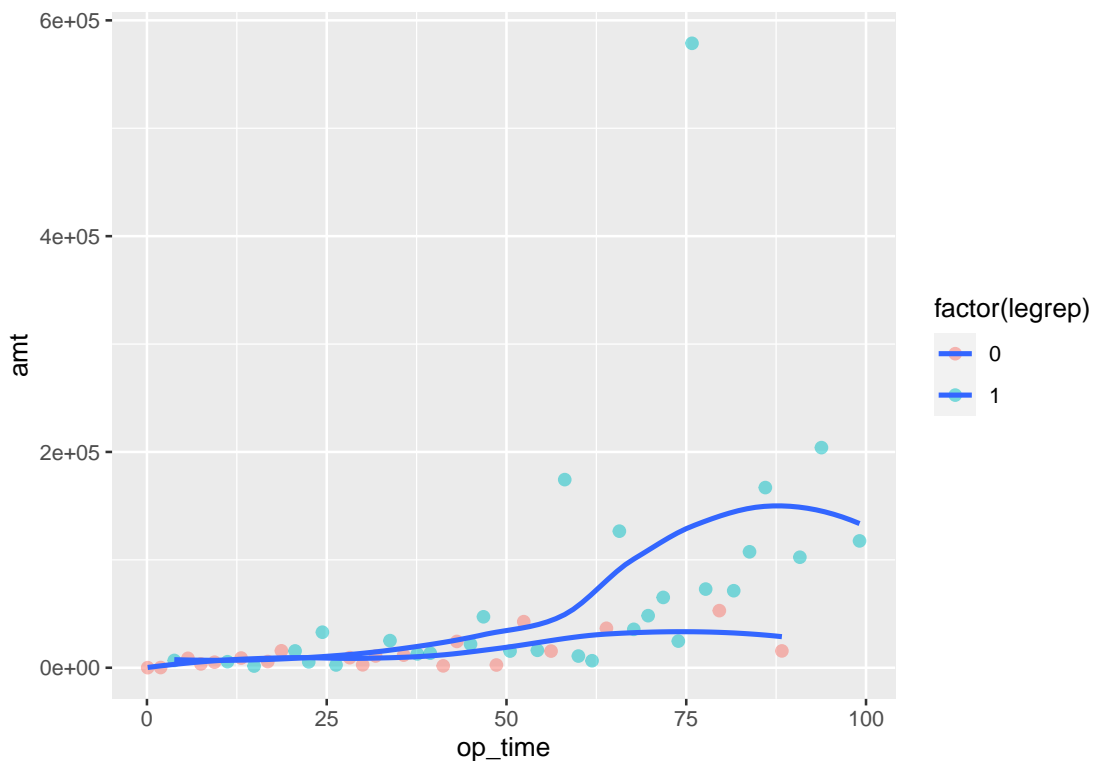
**Exercise 2.1.2.**  (**Variations of CHUNK 9**) Consider the following variations of CHUNK 9. Think about what kind of plots will be produced. Then run the code and see what happens.

```
# CHUNK 10
ggplot(persinj50, aes(x = op_time, y = amt, fill = factor(legrep))) +
  geom_point(aes(color = factor(legrep)), size = 2, alpha = 0.5) +
  geom_smooth(se = FALSE)

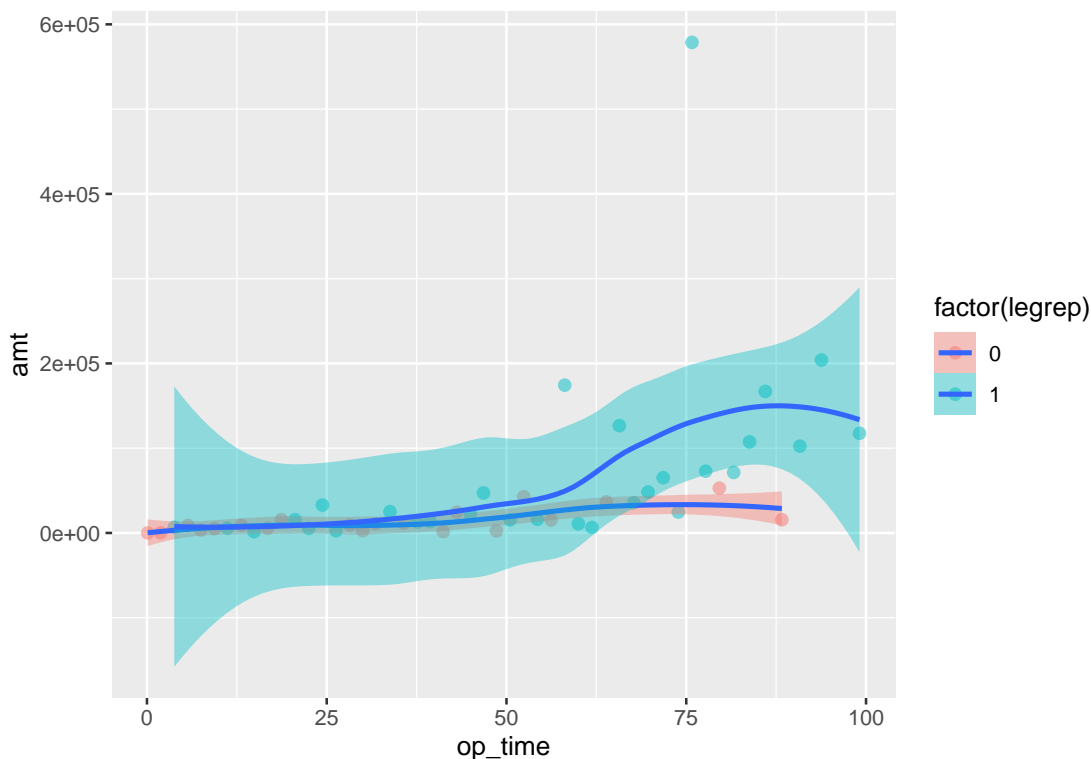
# CHUNK 11
ggplot(persinj50, aes(x = op_time, y = amt)) +
  geom_point(aes(color = factor(legrep)), size = 2, alpha = 0.5) +
  geom_smooth(aes(fill = factor(legrep)))
```

*Solution.* Let's look at the two chunks separately.

- *CHUNK 10:* Compared to CHUNK 9, the code in CHUNK 10 has the `fill` aesthetic added to the initial `ggplot()` call and the option `se = FALSE` added to the `geom_smooth()` function. The `fill` aesthetic has no effect on the `geom_point()` function, but it does affect the `geom_smooth()` function, which, by default, produces the standard error bands that are filled in color according to the `fill` aesthetic. Even though these bands are switched off due to the option `se = FALSE`, separate smoothed curves are still fitted to the two groups of injuries. Without the `color` aesthetic, however, the two curves share the same color, which is blue.



- *CHUNK 11*: Compared to *CHUNK 9*, the code in *CHUNK 11* has the `fill` aesthetic added to the `geom_smooth()` function. As a result, separate smoothed curves are fitted to the two groups of injuries with the standard error bands filled in color according to `legrep`. As the `color` aesthetic is absent in `geom_smooth()`, the two smoothed curves still share the same color (blue).



*Remark.* This example once again illustrates the subtlety of ggplots. A seemingly minor change in your code can lead to substantially different output. □

**Faceting.** In Figures 2.1.4 to 2.1.6, we grouped the 50 observations in the `persinj50` data according to the presence of legal representation. In `ggplot2`, grouping is achieved by mapping one or more categorical variables in the data to visual elements like `color`, `shape`, `fill`, `size`, and `linetype`, as we did earlier.

- We now consider *faceting*, which is another useful way to categorize our data into distinct groups. While grouping showcases two or more groups of observations in a single plot, faceting displays the observations in *separate* plots (known as a “small multiple” plot) produced for each value of the faceting variable placed side-by-side, usually on the same scale, to facilitate comparison. In `ggplot2`, faceting is accomplished by the `facet_wrap()` function or the `facet_grid()` function, depending on how many faceting variables there are. The `facet_wrap()` function is often used when there is only one faceting variable. Its generic syntax is

```
facet_wrap(~ FACET_VAR, ncol = N),
```

where the first argument specifies, following the tilde character (`~`), the faceting variable by means of R’s formula syntax (more details on formulas in R will be given in Chapter 3). The second argument, which is optional, determines the number of columns used to display the facets. The code in CHUNK 12, for example, produces the faceted scatterplot in Figure 2.1.7.

```
# CHUNK 12
```

```
ggplot(persinj50, aes(x = op_time, y = amt)) +
  geom_point(size = 2, alpha = 0.5) +
  geom_smooth() +
  facet_wrap(~ legrep) # Try to add scales = "free" to see what happens
```

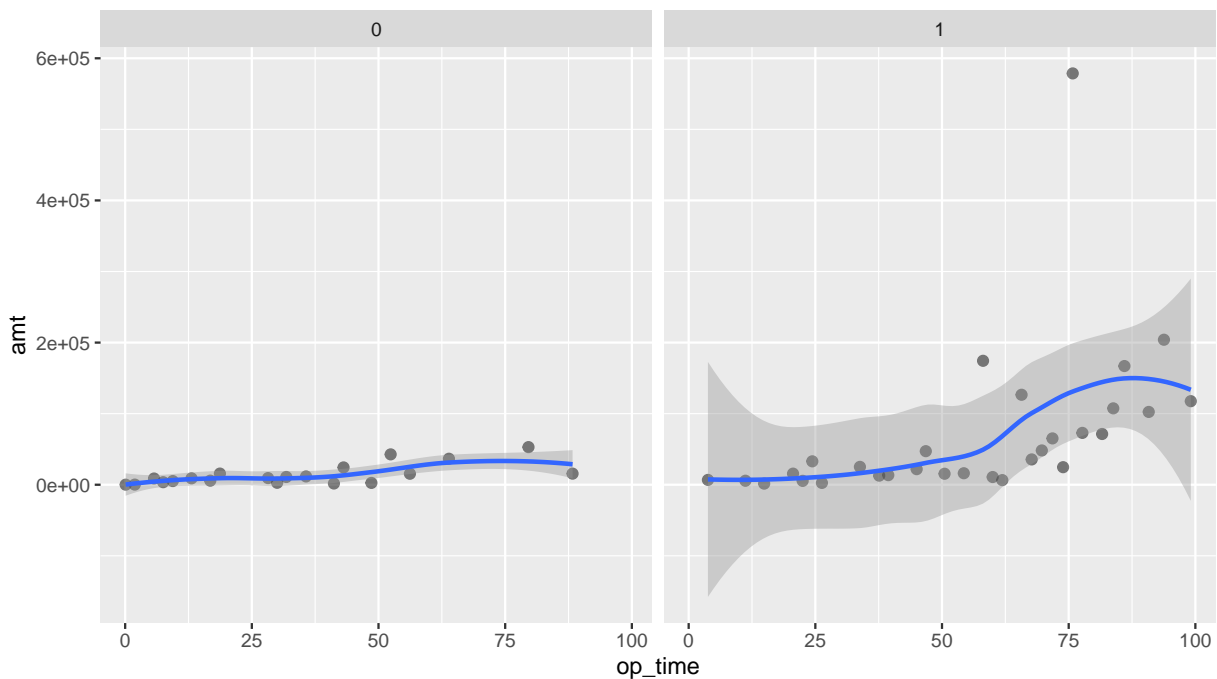


Figure 2.1.7: Scatterplots of `amt` against `op_time` faceted by `legrep` in the `persinj50` dataset.

The two scatterplots are laid out in order according to the two values of `legrep`. A label is displayed at the top of each facet (0 and 1) for easy identification. By default, the two plots are designed to share the same scale of the vertical axis. Adding the option `scales = "free"`, which is used in some past PA exams (e.g., December 2020 exam), to the `facet_wrap()` function will relax this constraint and “free” the scales (try it!). Regardless, Figure 2.1.7 is once again a manifestation that the claim amount of injuries with legal representation behaves differently as a function of `op_time` from those without.

It is possible to do faceting when there are two faceting variables. In this case, we can do a cross-classification using the `facet_grid()` function, which produces a two-dimensional “grid” of plots. Its syntax is similar to that of `facet_wrap()`:

```
facet_grid(FACET_VAR_1 ~ FACET_VAR_2, ncol = N)
```

## 2.1.2 Customizing Your Plots

We learned the basic structure of a ggplot in the previous subsection. We now look at how to customize a ggplot in terms of the appearance and range of coordinate axes, and how to add and modify cosmetic enhancements such as legends, titles, and subtitles. You will see that it is easy to fine-tune a ggplot to suit your needs.

**Axes.** Many datasets have outlier values which, when included in the plots, may distort the scaling of the axes in such a way to obscure the big picture in the data. In other cases, you may want to zoom in and focus on observations lying in a certain range of values, which you can achieve by tweaking the coordinate axes. In `ggplot2`, you can adjust the range of values of the coordinate axes by using the `xlim` and `ylim` arguments of the `coord_cartesian()` function. The two arguments are set to a two-element numeric vector indicating the desired lower and upper limits of the x-axis and y-axis. Data points outside the limits are thrown away.

Another way to display the data points more effectively is to adjust the scale, for example, from a linear scale to a log scale. This is especially useful when dealing with highly skewed variables, as we will see in the next section. The function `scale_x_log10()` (do not miss the parentheses!) converts the scale of the x-axis of a ggplot to a log 10 basis and re-positions the data points accordingly. When this function is applied, the points 10, 100, 1000, 10000 will be shown as consecutive numbers on the x-axis because their log 10 counterparts,  $\log_{10} 10 = 1$ ,  $\log_{10} 100 = 2$ ,  $\log_{10} 1000 = 3$ , and  $\log_{10} 10000 = 4$  are consecutive. As you can expect, the function `scale_y_log10()` performs the same operation on the y-axis.

**Titles, subtitles, and captions.** In some cases, you want to give more fancy names for the axis labels. The `labs()` function allows us to set the label for the x-axis, y-axis, and the text for the title, subtitle, and caption of a ggplot using the `x`, `y`, `title`, `subtitle`, and `caption` arguments, respectively, with the desired label or text supplied as a character string.<sup>4</sup>

### ⚠ EXAM NOTE ⚠

In quite a few past PA exams, you are asked to comment on the strengths and weaknesses of a given plot. The lack of (useful) axis labels and titles, while minor in most cases, is often a weakness considered by the SOA. Try to keep this in mind for future exams.

To illustrate the use of the cosmetic enhancements above, run CHUNK 13 to produce an enhanced version of the scatterplot in CHUNK 8 (Figure 2.1.5) with labels for the x-axis, y-axis, and the title added, and with the y-axis restricted to the range between  $-200,000$  and  $300,000$ . The resulting scatterplot is shown in Figure 2.1.8. The restriction of the y-axis has the effect

<sup>4</sup>If you want to set just the label for the x-axis, y-axis, or the text for the title, you can use the `xlab()`, `ylab()`, and `ggtitle()` functions, respectively.



of excluding the outlier whose claim amount is close to 600,000, way more than other claim amounts, and helping us focus on the main trend in the data much more easily.

```
# CHUNK 13
ggplot(persinj50, aes(x = op_time,
                      y = amt,
                      color = factor(legrep),
                      fill = factor(legrep))) +
  geom_point(size = 2, alpha = 0.5) +
  geom_smooth() +
  labs(title = "Personal Injury Dataset",
       x = "Operational Time",
       y = "Claim Amount") +
  coord_cartesian(ylim = c(-200000, 300000))
```

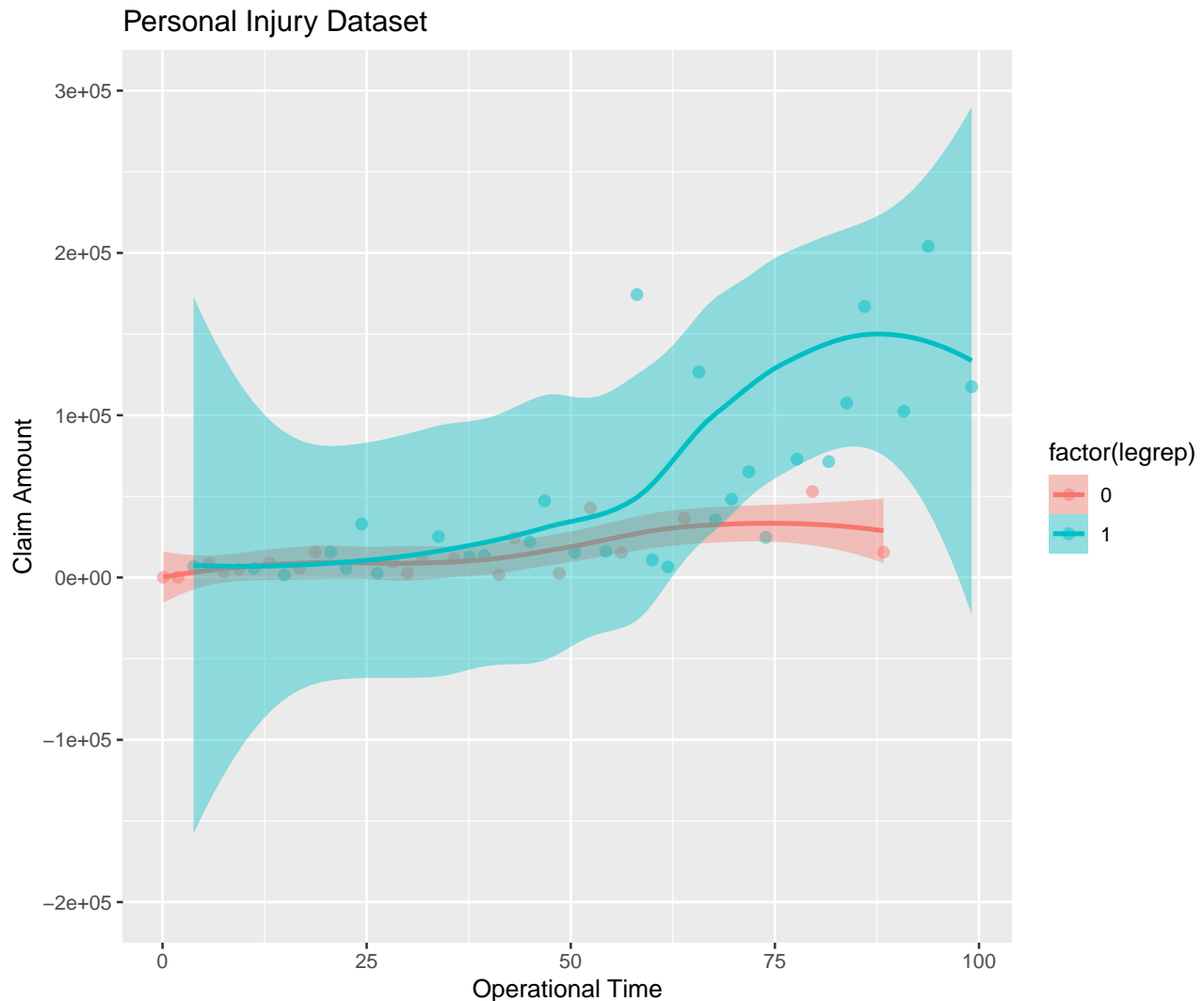


Figure 2.1.8: A version of Figure 2.1.5 with labels for the x-axis, y-axis, and the title added, and with the y-axis restricted to the range between  $-200,000$  and  $300,000$ .

## 2.2 Data Exploration

- Now that we have learned how to make some simple ggplots, in this section we will apply our data visualization techniques to perform *exploratory data analysis* (EDA), which is an indispensable part of any predictive modeling exercise and is tested in almost every PA exam. The following tasks from the three most recent PA exams will convince you of how important EDA is.

### April 2024 PA Exam: Task 4

Your manager is interested in the relationship between the market share held by the airline with the highest market share between two cities (**large\_ms**) and the average fare between those cities (**fare**). Your assistant produces the graphs below.

- (a) (3 points) Describe one pro and one con of each visualization in explaining the relationship between **large\_ms** and **fare**.
- (b) (2 points) Recommend a visualization for your assistant to create to understand the modeling implications of the graphic above.

### October 2023 PA Exam: Task 5

Your client is interested in obtaining a deeper understanding of how tuition prices and the size of the universities are reflected in the dataset.

- (a) (3 points) Suggest two numerical variables from the Data Dictionary for this analysis and describe two univariate techniques that can be used to explore them.
- (b) (2 points) Suggest a categorical variable from the Data Dictionary for this analysis and describe a univariate technique to explore the variable.
- (c) (2 points) Describe a bivariate visualization that can be applied to understand the relationship between a numeric variable and a categorical variable.
- (d) (2 points) Interpret the plot above.

### April 2023 PA Exam: Task 1

Your assistant creates two graphs and wants to choose the graph that provides the more easily understood visualization of the relative number of boardings at each stop.

- (a) (2 points) State which graph your assistant should use and explain why this graph is better than the alternative.

EDA is an integral part of predictive modeling because it serves two important purposes:

- *Data validation*: It allows us to perform commonsense checks and identify nonsensical data values (e.g., a negative value for age, which is impossible), which are potential data errors that may lead to unreasonable model results and should be fixed. It also reveals the possible existence of outliers that merit further considerations. After anomalous and inappropriate data values have been removed, the data becomes ready for analysis.
- *Characteristics of variables*: It also helps us understand the key characteristics of the variables in the data. Such an understanding may suggest useful ways to pre-process the variables to improve the prediction performance and interpretability of the models we will construct, and, most importantly, decide on an appropriate type of predictive model that is likely to meet our business needs.

Typically, EDA is accomplished by a *combination* of two kinds of tools:

- (1) *Descriptive statistics* (a.k.a. summary statistics) that quickly summarize different distributional properties of the variable(s) of interest

Examples: Mean, variance, mode, correlation, table of frequency counts

- (2) *Graphical displays* (a.k.a. visual displays) that allow us to get a quick impression of the overall distribution of the variable(s) of interest

Graphs are often more informative than a table of summary statistics and sometimes can reveal information that would be missed otherwise, e.g., the presence of outliers.

Examples: Histograms, boxplots, bar charts, and their variants

In this section, we will return to the full `persin` data (with 22,036 observations) and use it to illustrate the creation and interpretation of descriptive statistics and graphical displays.

## 2.2.1 Univariate Data Exploration

Let's begin with *univariate* data exploration—exploration that sheds light on the distribution of only one variable at a time. The specific statistics and graphical tools will depend on whether the variables you are analyzing are numeric or categorical (precise definitions of numeric and categorical variables will be given on page 131). Both types of variable are part of the dataset of a typical PA exam.

### Numeric Variables

**Descriptive statistics.** Statistical summaries are mainly used to reveal two aspects of the distribution of a **numeric variable**:

- *Central tendency*: The central tendency of a numeric variable, whether it be continuous or discrete, is often quantified by its **mean** and **median**. These two metrics capture, in a loose sense, the typical “size” of the variable and can be readily produced in R by applying the `summary()` function to the variable of interest.

- *Dispersion:* Common measures of dispersion include **variance**, **standard deviation**, and **inter-quartile range** (defined as the difference between the 75% quantile and the 25% quantile of a variable), all of which measure in a way how spread out the values of the numeric variable are over its range.

**Graphical displays.** To visualize the distribution of numeric variables, histograms and boxplots are convenient graphical aids.

- *Histograms:* **Histograms** divide the observations into several equally spaced bins (or buckets) and provide a visual summary of the count or relative frequency in each bin. Looking at a histogram, we can learn about the overall shape of the distribution of a numeric variable and where most observations lie. Figure 2.2.1 (a) shows a prototypical histogram.

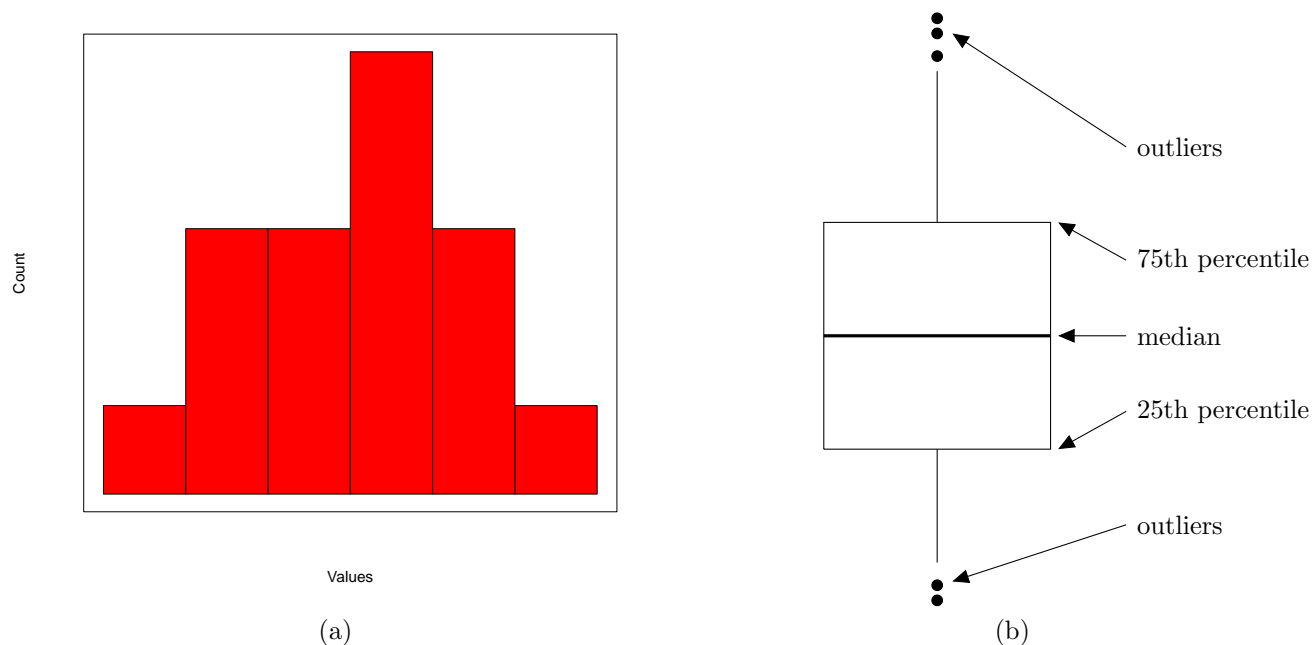



Figure 2.2.1: Prototypical histogram (left) and boxplot (right).

- *Boxplots:* **Boxplots**, a.k.a. *box plots* and *box-and-whiskers plots*, visualize the distribution of a numeric variable by placing its 25% quantile, the median, the 75% quantile in a “box,” with the rest of the data points constituting the “**whiskers**.” The amount of spacing between different parts of a boxplot reflects the degree of dispersion and skewness of the variable’s distribution. “**Outliers**,” defined here as data points that are above or below 1.5 times the inter-quartile range from either edge of the box, are shown as large dotted points. See Figure 2.2.1 (b) for a prototypical boxplot.

Although boxplots do not directly show the actual shape of the variable’s distribution, they offer a useful graphical summary of the key numeric statistics and allow for a visual

comparison of the distributions of different numeric variables (in particular, the relative magnitude of their median and dispersion) or the distribution of the same numeric variable across different levels of another categorical variable. We will see how this works in Subsection 2.2.2.

**Exercise 2.2.1.**  (What can we get from a boxplot?) Determine whether each of the following distributional quantities can be obtained (at least approximately) from a boxplot.

- (a) Mean
- (b) Median
- (c) Mode
- (d) Inter-quartile range
- (e) Standard deviation

*Solution.* Among the five quantities above, only the median (b) and inter-quartile range (d) can be read off a boxplot. □

**Summary statistics.** In the `persinj` data, there are two numeric variables, `amt` and `op_time`. In CHUNK 1, we focus on the `amt` variable for the purposes of illustration and apply the `summary()` function to the `amt` variable.


```
# CHUNK 1
# Reload the data
persinj <- read.csv("persinj.csv")

summary(persinj$amt)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       10    6297   13854   38367   35123 4485797
```

When the `summary()` function is applied to a numeric variable, a six-number “summary” is produced. We can see that the mean of claim amount (38,367) is way higher than its median (13,854) and the 75th percentile is much further away from the median than the 25th percentile, indicating that the distribution of claim amount is highly skewed to the right. The right skew suggests that the values to the right of the mean of claim amount tend to be further away from the mean than those to the left, so there is a heavy tail that extends to the far right. In fact, the largest claim amount, 4,485,797, is almost an astronomical figure compared with the mean or median.

In the following exercise, you will calculate the summary statistics for the two groups of injuries classified by legal representation, which is a binary categorical variable. Doing so helps us understand the effect of legal representation on claim amount.

**Exercise 2.2.2.**  (Calculating the summary statistics for two groups of observations) Write R code to calculate the summary statistics for claim amount separately for injuries with legal representation and those without legal representation. Comment on the central tendency and dispersion of claim amount for these two groups of injuries.

*Solution.* To extract the two groups of injuries, we can use the method of logical subsetting introduced in Chapter 1 to split the dataset into two subsets:

- (1) A subset called `persinj.0` corresponding to injuries without legal representation (`legrep = 0`)
- (2) A subset called `persinj.1` corresponding to injuries with legal representation (`legrep = 1`)

Then we look at the summary statistics of the claim amount variable within the two subsets. This is done in CHUNK 2.

```
# CHUNK 2
persinj.0 <- persinj[persinj$legrep == 0, ]
persinj.1 <- persinj[persinj$legrep == 1, ]
summary(persinj.0$amt)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      10   4061   11164   32398   29641 2798362

summary(persinj.1$amt)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      20   7305   15309   41775   38761 4485797

sd(persinj.0$amt)

## [1] 77820.33

sd(persinj.1$amt)

## [1] 97541.38
```

The comparison is apparent: Claims with legal representation not only are larger on average (perhaps with legal advice the claimant is able to fight for larger settled claims), but also are more spread out. The relative variability is confirmed when the standard deviations of the two groups of claim amounts are computed by the `sd()` function.  $\square$

**Histograms.** Now we turn to visual representations. In `ggplot2`, a histogram is constructed by the `geom_histogram()` function, which only requires the `x` aesthetic (but not the `y` aesthetic) capturing the numeric variable of interest. In CHUNK 3, we make four histograms colored in blue for the claim amount variable (Figure 2.2.2) with different choices of the `bins` parameter, which controls the number of bins in a histogram. Note that:

- To produce better visual effects, we have restricted the range of the horizontal axis to be

between 0 and 100,000.<sup>5</sup> (Try to see how the histograms look if the command `xlim(0, 100000)` is lifted.)

- Recall that the `fill` argument is for coloring the *interior* of shapes. Had we used the `color` argument and typed `color = "blue"` inside the `geom_histogram()` function, only the border lines of the vertical bars would be in blue and the interior would remain gray. (Try it!)
- One thing that differentiates a ggplot from a plot created from the base R installation is that a ggplot can be saved as an object in R and manipulated further. Using the `grid.arrange()` function in the `gridExtra` package, we can place several ggplots (four histograms, `p1`, `p2`, `p3`, and `p4`, in this case) in a single figure for ease of comparison.

Note that this is not the same as faceting, where each figure is for a sub-sample of the data corresponding to certain values of the faceting variable(s). Here, we would like to arrange completely independent ggplots, all of which correspond to the entire sample, side-by-side.

By default, `geom_histogram()` chooses the number of bins based on a rule of thumb (which is 30 here). The higher the value of `bins`, the closer the resulting plotted function to a smooth curve. When plotting histograms, it is a good idea to experiment with a few values of `bins` as different values of the parameter can reveal quite different patterns. In Figure 2.2.2, 30 (default) and 40 are both fine values for `bins` while 10 makes the histogram too crude and 80 may have described the data too finely. Regardless of the value of `bins`, the histograms corroborate the earlier findings in CHUNK 1 that the distribution of claim amount is heavily lopsided to the right with a pronounced tail.

```
# CHUNK 3
library(ggplot2)
p1 <- ggplot(persin的角度, aes(x = amt)) +
  geom_histogram(bins = 10, fill = "blue") +
  xlim(0, 100000) +
  ggtitle("Bins = 10")
p2 <- ggplot(persin的角度, aes(x = amt)) +
  geom_histogram(fill = "blue") +
  xlim(0, 100000) +
  ggtitle("Default value")
p3 <- ggplot(persin的角度, aes(x = amt)) +
  geom_histogram(bins = 40, fill = "blue") +
  xlim(0, 100000) +
  ggtitle("Bins = 40")
p4 <- ggplot(persin的角度, aes(x = amt)) +
  geom_histogram(bins = 80, fill = "blue") +
  xlim(0, 100000) +
  ggtitle("Bins = 80")
```

<sup>5</sup>In fact, if you use the `xlim` and `ylim` functions directly without using the `coord_cartesian()` function, you are in effect looking at the *conditional* distribution of the data given that the observations lie within the ranges specified.

```
# CHUNK 3 (Cont.)  
library(gridExtra)  
grid.arrange(p1, p2, p3, p4, ncol = 2)
```

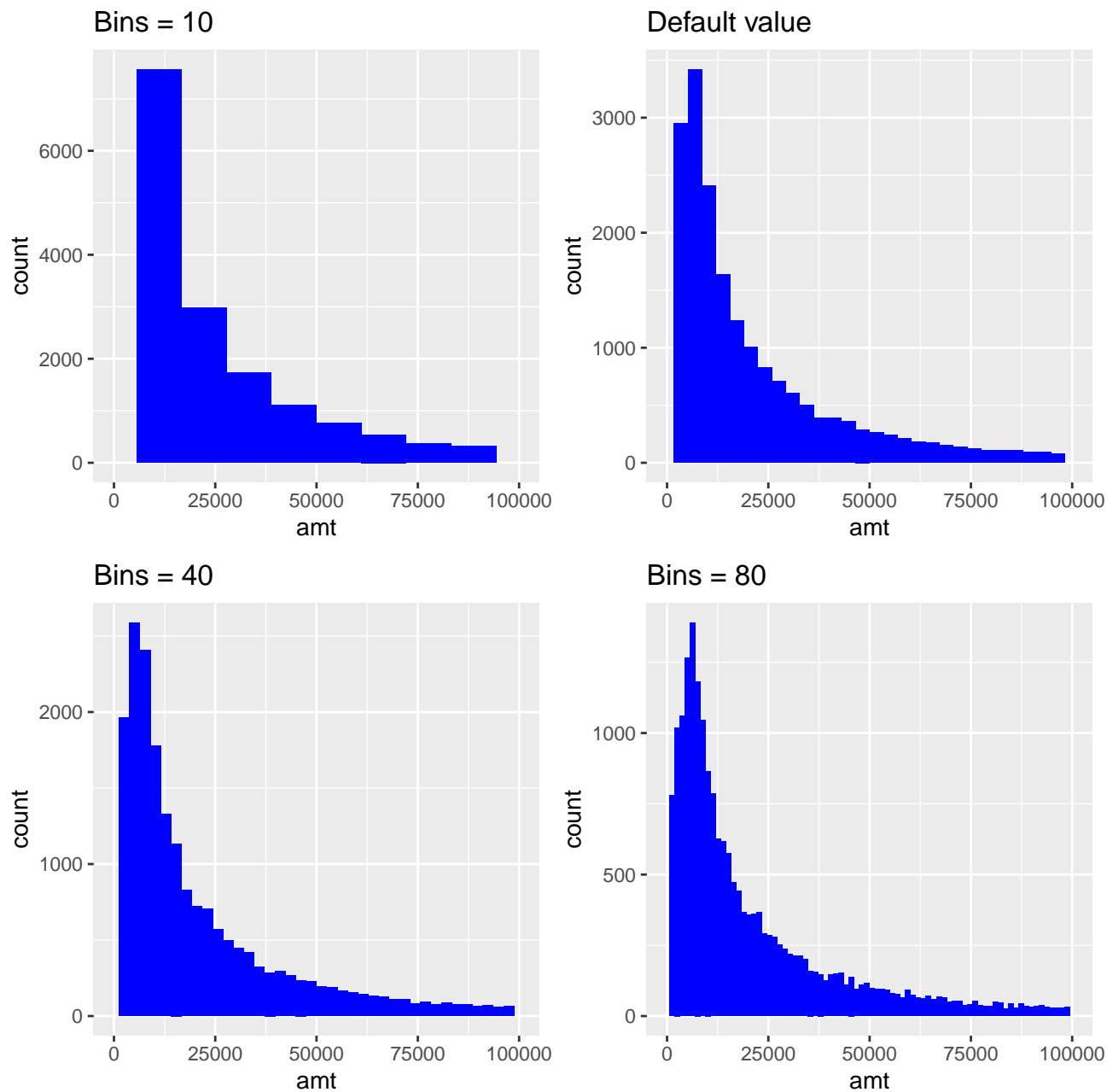


Figure 2.2.2: Four histograms of `amt` in the `persin` dataset with different values of `bins`.



**Problems with skewed data and possible solutions.** Earlier in this chapter, we briefly mentioned the concept of the skewness of a distribution. You may recall from Exam P/FAM/STAM that skewness is a measure of the asymmetry of a distribution. It may help to distinguish the three cases below:

- Case 1.* For a symmetric distribution, i.e., a distribution whose histogram or boxplot is symmetric about the mean  $\mu$ , values above  $\mu$  are, on average, of the same distance from  $\mu$  as those below (see Figure 2.2.3 (a)).
- Case 2.* A *right-skewed* distribution is one for which values above the mean tend to be further away from the mean than those below, leading to a histogram with a “long tail” that extends to the right (see Figure 2.2.3 (b)).
- Case 3.* A *left-skewed* distribution is the opposite: Values below the mean tend to be further away from the mean than those above, creating a histogram with a long left tail (see Figure 2.2.3 (c)).

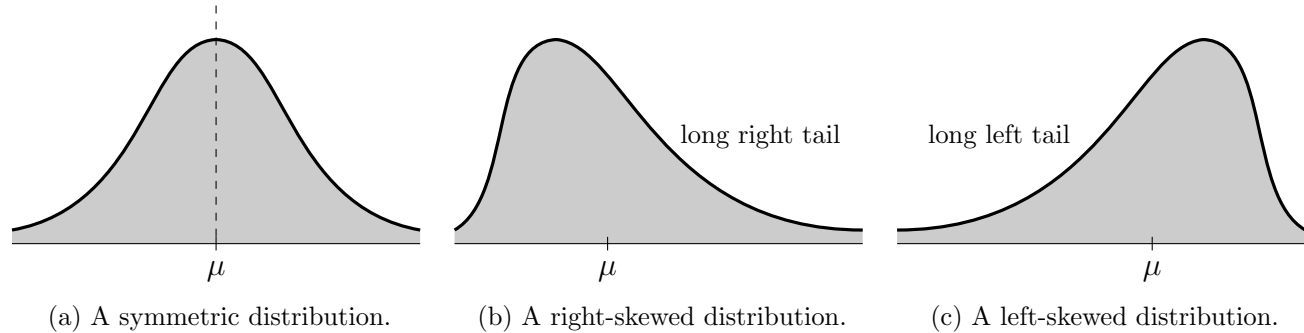


Figure 2.2.3: Graphical illustrations of the density functions (or histograms) of symmetric, right-skewed, and left-skewed distributions.

In real modeling work, it is right-skewed distributions (to different degrees) that arise most frequently, especially in insurance and financial applications. Many variables such as income and loss amount are, by design, non-negative, so there is a limit to how long the left tail can be, but the right tail may be extraordinarily thick—the largest values can be astronomically large compared to the mean. These variables can be problematic for two reasons:

- (*Predictive power*) After all, our objective in predictive analytics is to study the association between the target variable and predictors in the data over a wide range of variable values. If most of the observations of the target variable cluster narrowly in the small-value range, this will make it difficult to investigate the effect of the predictors on the target variable globally—we simply don’t know enough about the target variable in the right tail. The same idea applies to a right-skewed predictor. If a predictor exhibits a heavy right skew, we are unable to differentiate the observations of the target variable effectively on the basis of the values of the predictor, most of which are concentrated in the small-value range.

- (*Model fitting*) A number of predictive models (e.g., linear models, decision trees) are fitted by minimizing the sum of the squared discrepancies between the observed values and predicted values of the target variable. If the target variable is right-skewed, then the outliers, or extreme values, will contribute substantially to the sum and have a disproportionate effect (or “leverage”) on the model. This is undesirable unless the right tail is where our main concern lies.
- To correct for the **skewness**, we can apply a monotone concave function to shrink the outliers relative to the smaller values and symmetrize the overall distribution, while preserving the ranks of the observed values of the variable. This will dampen the effects of the extreme values on the model and tend to improve its overall goodness of fit than using the original right-skewed variable. Two commonly used transformations for dealing with right-skewed variables are:
  - **Log transformation:** This transformation (typically with respect to base  $e$ ) can be applied as long as the variable of interest is *strictly* positive. None of the variable values should be zero or negative. (Remember that  $\log x$  is not well-defined if  $x \leq 0$ !)

**⚠ EXAM NOTE ⚠**

The log transformation is one of the (if not, the) most commonly used variable transformations in Exam PA and in applied modeling in general.

- **Square root transformation:** Although not discussed in the PA modules, the square root transformation  $\sqrt{x}$  is in a similar spirit to the log transformation, but is applicable even to non-negative variables, some of whose values can be zero.

To see the effects of the log and square root transformations in action, CHUNK 4 produces a histogram for the log of claim amount and the square root of claim amount. It is evident that the log transformation has effectively removed the right skewness of claim amount and made the resulting distribution much more symmetric. In contrast, the square root of claim amount remains highly right-skewed. In general, the log transformation does a better job of remedying the right skewness of a variable than the square root transformation, but it may overdo things and make the transformed variable left-skewed (fortunately, not the case in Figure 2.2.4).

Because of the extreme skewness of claim amount in the `persin` data, the log transformation is the more appropriate transformation to use and we will adopt it in the rest of this section. We will see that the log transformation makes it much easier to discover relationships between variables.

```
# CHUNK 4
p1 <- ggplot(persin, aes(x = log(amt))) +
  geom_histogram()
p2 <- ggplot(persin, aes(x = sqrt(amt))) +
  geom_histogram()
grid.arrange(p1, p2, ncol = 2)
```

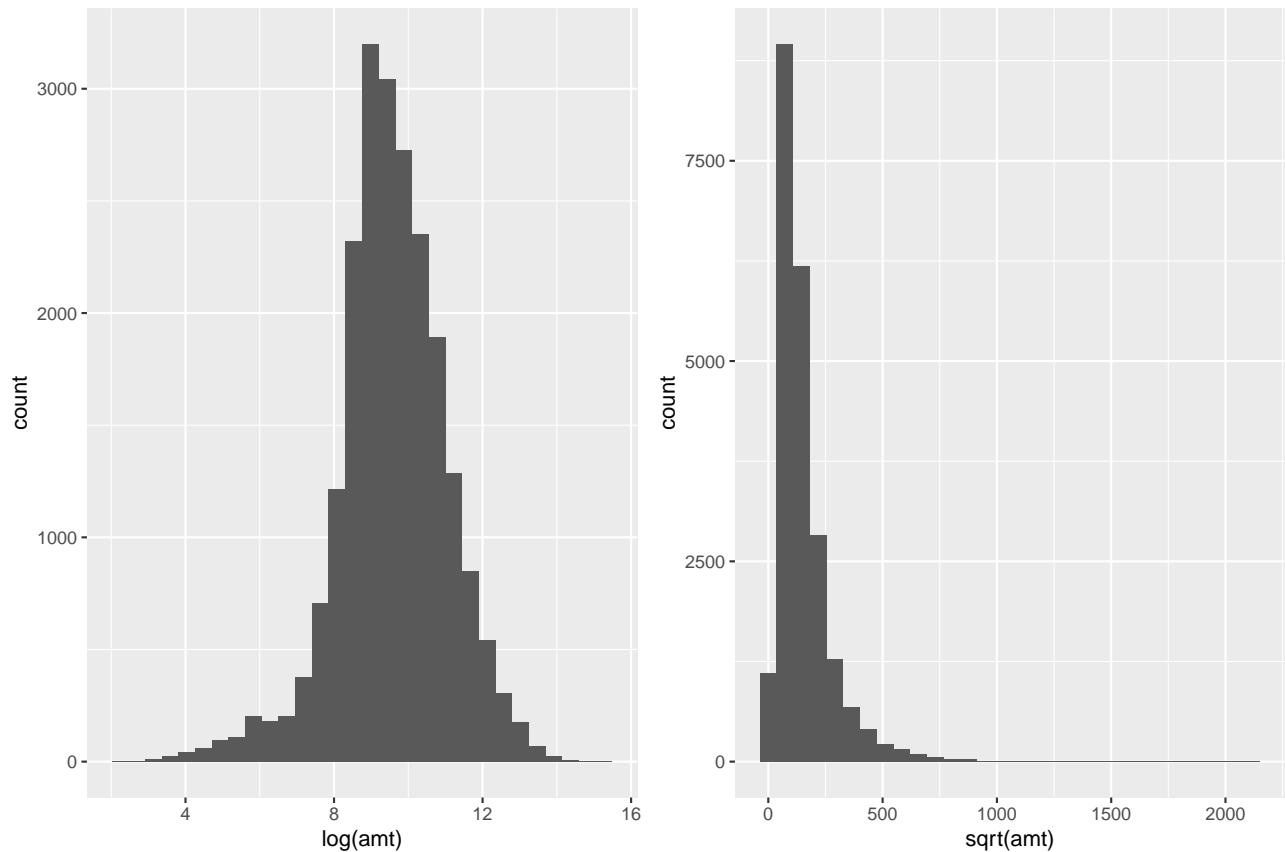


Figure 2.2.4: Histograms of the log of claim amount (left) and square root of claim amount (right) in the `persin` dataset.

**Outliers.** We mentioned outliers in our discussion of skewed data above. Let's digress slightly and look at outliers in some greater depth, which were tested in some recent PA exams.

**⚠ EXAM NOTE ⚠**

"Outliers" have been removed from the learning outcomes of the PA exam syllabus effective from April 2023, so the importance of outliers is likely to drop in future exams.

• There is no universal quantitative definition of **outliers**. We generally think of outliers as anomalous data points that substantially differ from the overall pattern of the data and appear to be strange (for categorical variables, observations in sparse factor levels can be considered outliers). They typically arise in two ways:

- *Errors:* Outliers can arise due to errors in the data collection process, such as data entry errors. They are observations whose values are so ridiculous that they can be safely dismissed as erroneous, such as a negative age and a current customer who was born in 1600. If an outlier is erroneous, then it is reasonable to correct the error or remove it entirely.
- *Natural:* Outliers can also arise naturally. Not necessarily errors, they are simply observations whose values are far away from the rest of the data, but are possible in theory. Examples include a policyholder currently aged 140 and an actuary earning \$10 million a year (well, some actuaries are insanely rich!).

Natural outliers are harder to deal with than erroneous outliers. Here are some possible options:

- *Remove:* If we (somehow!) know that a natural outlier is not likely to have a material effect on the final model, then it is fine to remove it.
- *Keep:* If the outliers make up only an insignificant proportion of the data and are unlikely to create bias, then it is sensible to leave them in the data.
- *Modify:* We can also modify the outliers to make them more reasonable, like censoring the policyholder age of 140 at 100.
- *Using robust model forms:* Instead of minimizing the squared error between the predicted values and the observed values, we could replace the squared error by the *absolute error*, which places much less relative weight on the large errors and reduces the impact of outliers on the fitted model.

(This is related to the concept of a loss function, to be discussed on page 152.)

**A variation of histograms: Density plots.** Before moving on to the next type of graphical displays for numeric variables, let’s also mention *density plots*, which are smoothed and scaled versions of histograms displaying “density” rather than counts on the vertical axis. Although there is hardly any mention of density plots in the PA modules and they function in more or less the same way as a histogram, they made an appearance in the December 7, 2020 and December 8, 2020 exams, so it is beneficial to get some exposure to these plots.

In CHUNK 5, we use the `geom_density()` function to make the density plots for the log of claim amount and the square root of claim amount (see Figure 2.2.5).

```
# CHUNK 5
p1 <- ggplot(persinj, aes(x = log(amt))) +
  geom_density()
p2 <- ggplot(persinj, aes(x = sqrt(amt))) +
  geom_density()
grid.arrange(p1, p2, ncol = 2)
```

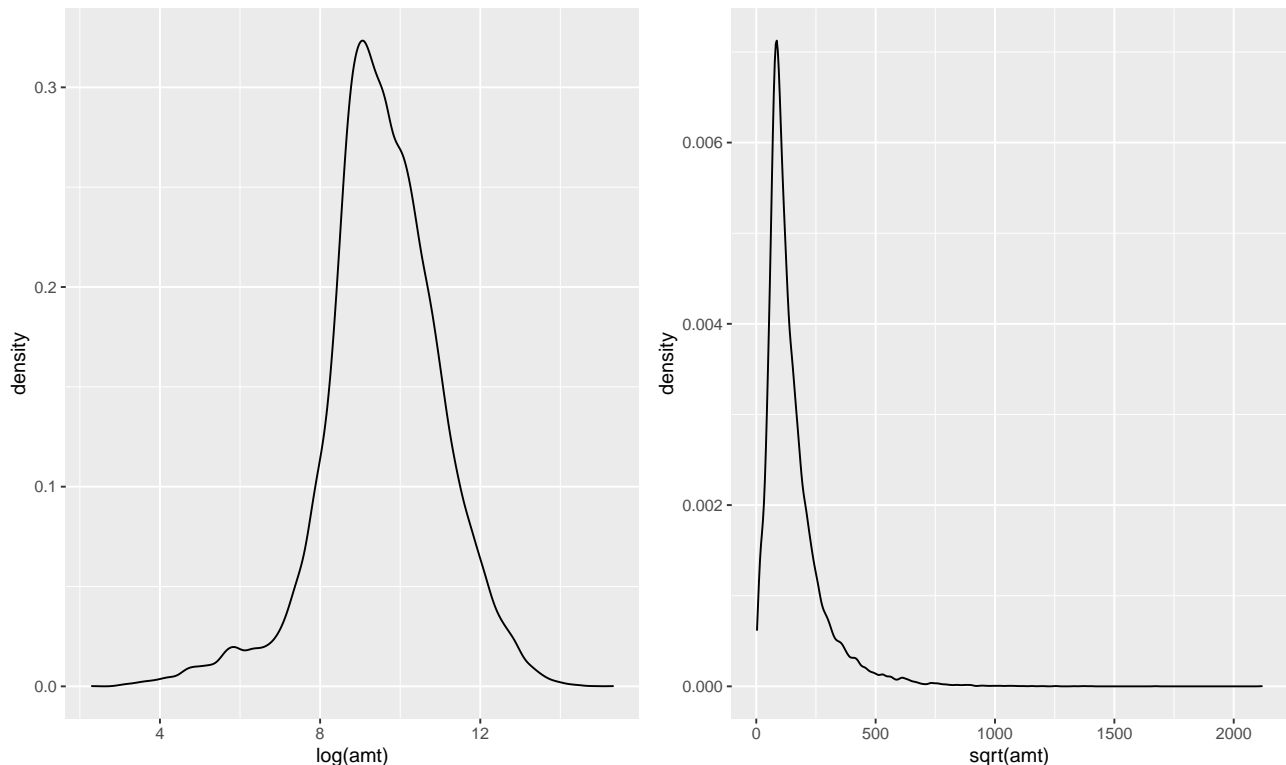


Figure 2.2.5: Density plots of the log of claim amount (left) and square root of claim amount (right) in the `persinj` dataset.

Comparing Figures 2.2.4 and 2.2.5, we can see that the density plots have the same shape as the corresponding histograms and can be interpreted in the same way. Do note that the vertical axis of the density plots shows “density” rather than “count” and the area under a density curve is always 1.

**Boxplots.** Besides histograms, boxplots are also convenient graphical aids to visualize the distribution of a numeric variable. They are constructed in `ggplot2` by the `geom_boxplot()` function, which takes the `y` aesthetic representing the numeric variable of interest (the `x` aesthetic is optional, but can be added to achieve splitting, as we will see later in this section). Use CHUNK 6 to draw a boxplot for each of claim amount and the log of claim amount (Figure 2.2.6).

```
# CHUNK 6
p1 <- ggplot(persinj, aes(y = amt)) +
  geom_boxplot()
p2 <- ggplot(persinj, aes(y = log(amt))) +
  geom_boxplot()
grid.arrange(p1, p2, ncol = 2)
```

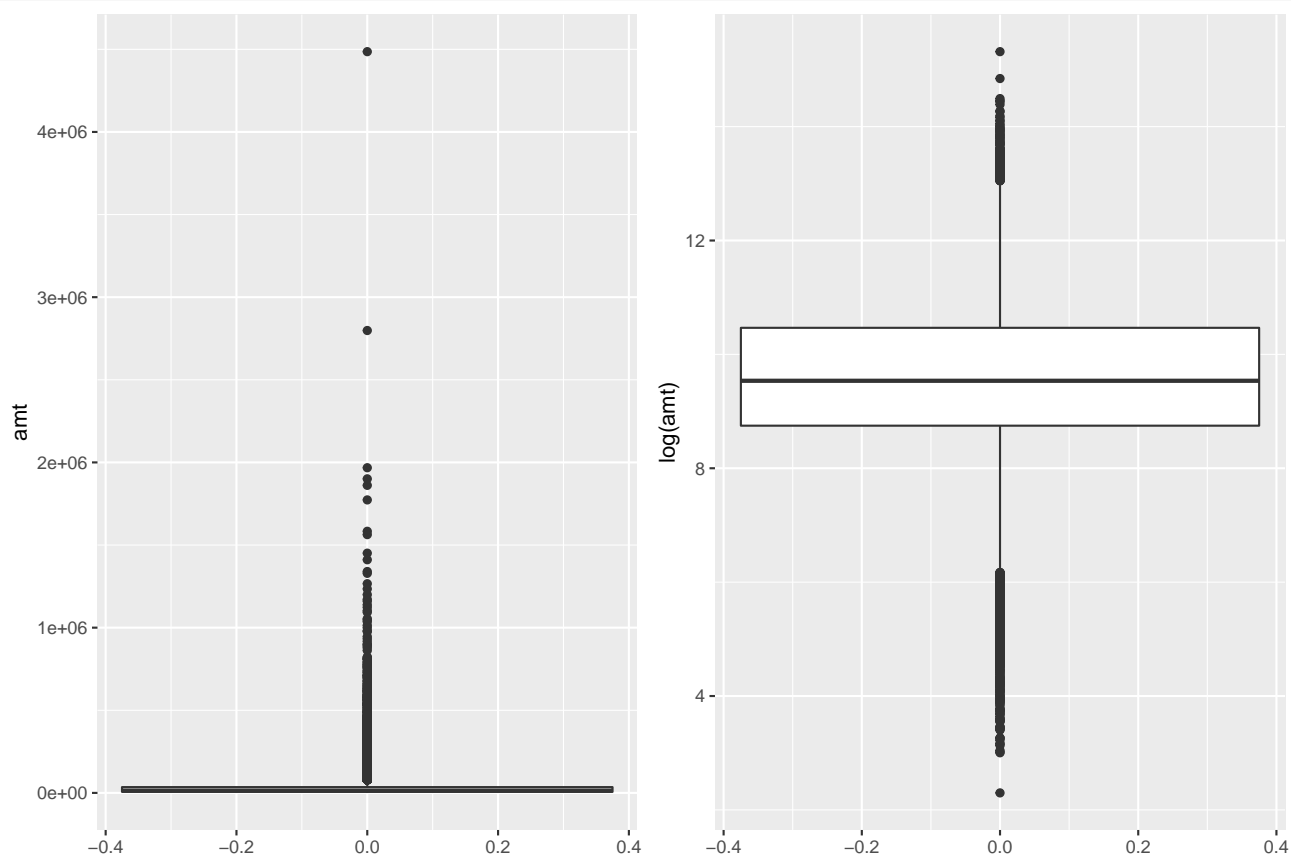


Figure 2.2.6: Boxplots of claim amount (left) and the log of claim amount (right) in the `persinj` dataset.

While most of the raw claim amounts are so close that the 25% percentile, median, and 75% percentile all degenerate to the same line, the log transformation corrects for the skewness and re-positions the data points for much easier visual inspection. Still there are quite a number of “outliers,” which are shown as large dotted points.

## Categorical Variables

### Case 1: Given raw data (more common in Exam PA)

- *Descriptive statistics—Frequency tables:* **Categorical variables**, even when coded as numbers, do not always have a natural order, so statistical summaries like the mean and median may not make sense. To understand the distribution of a categorical variable, we can look at the relative frequency of each of its levels through a frequency table, constructed by the `table()` function in R.
- *Graphical displays—bar charts:* When the number of levels of a categorical variable increases, a frequency table becomes more and more difficult to read. In most cases, the frequencies *per se* are not that important; what truly matters is their relative magnitude. In this regard, **bar charts** extract the information in a frequency table and present the numeric counts visually, highlighting the relative frequency of each level in the variable. Looking at a bar chart, we can easily tell which levels are the most popular and which ones have minimal observations.

The `persinj` dataset has two categorical variables, injury code (`inj`) and legal representation (`legrep`). Recall from Table 2.1 that `inj` has seven levels while `legrep` is binary. In CHUNK 7, we make two frequency tables for `inj`, one showing the raw counts and one showing the percentage counts of the seven levels of `inj`.

```
# CHUNK 7
table(persinj$inj)

##
##      1      2      3      4      5      6      9
## 15638  3376  1133   189   188   256  1256

table(persinj$inj)/nrow(persinj)

##
##           1           2           3           4           5           6
## 0.709656925 0.153203848 0.051415865 0.008576874 0.008531494 0.011617353
##           9
## 0.056997640
```

We can see that the predominant group of injuries is those of injury code 1, followed by codes 2, 9 and 3. The other three groups have minimal observations.

The numbers in a frequency table can be depicted in a bar chart created in `ggplot2` by the `geom_bar()` function, which takes the `x` aesthetic representing the categorical variable of interest and produces a bar proportional to the number of observations for each level of the variable. Run CHUNK 8 to produce two bar charts for injury code corresponding to the two frequency tables in CHUNK 7 (see Figure 2.2.7).

**Note:** The command `y = ..prop..`, `group = 1` in the bar chart in CHUNK 8, as its name indicates, computes proportions rather than raw counts and relies on the so-called stat function associated with a geom function. This concept is rather involved, but is of limited use in Exam PA. If you are interested in what it is, please read pages 80 and 81 of *Data Visualization: A Practical Introduction*.

```
# CHUNK 8
# first convert inj and legrep to factors (original data type is integer)
persinj$inj <- as.factor(persinj$inj)
persinj$legrep <- as.factor(persinj$legrep)

p1 <- ggplot(persinj, aes(x = inj)) +
  geom_bar(fill = "blue")
p2 <- ggplot(persinj, aes(x = inj)) +
  geom_bar(fill = "blue", aes(y = ..prop.., group = 1))
grid.arrange(p1, p2, ncol = 2)
```

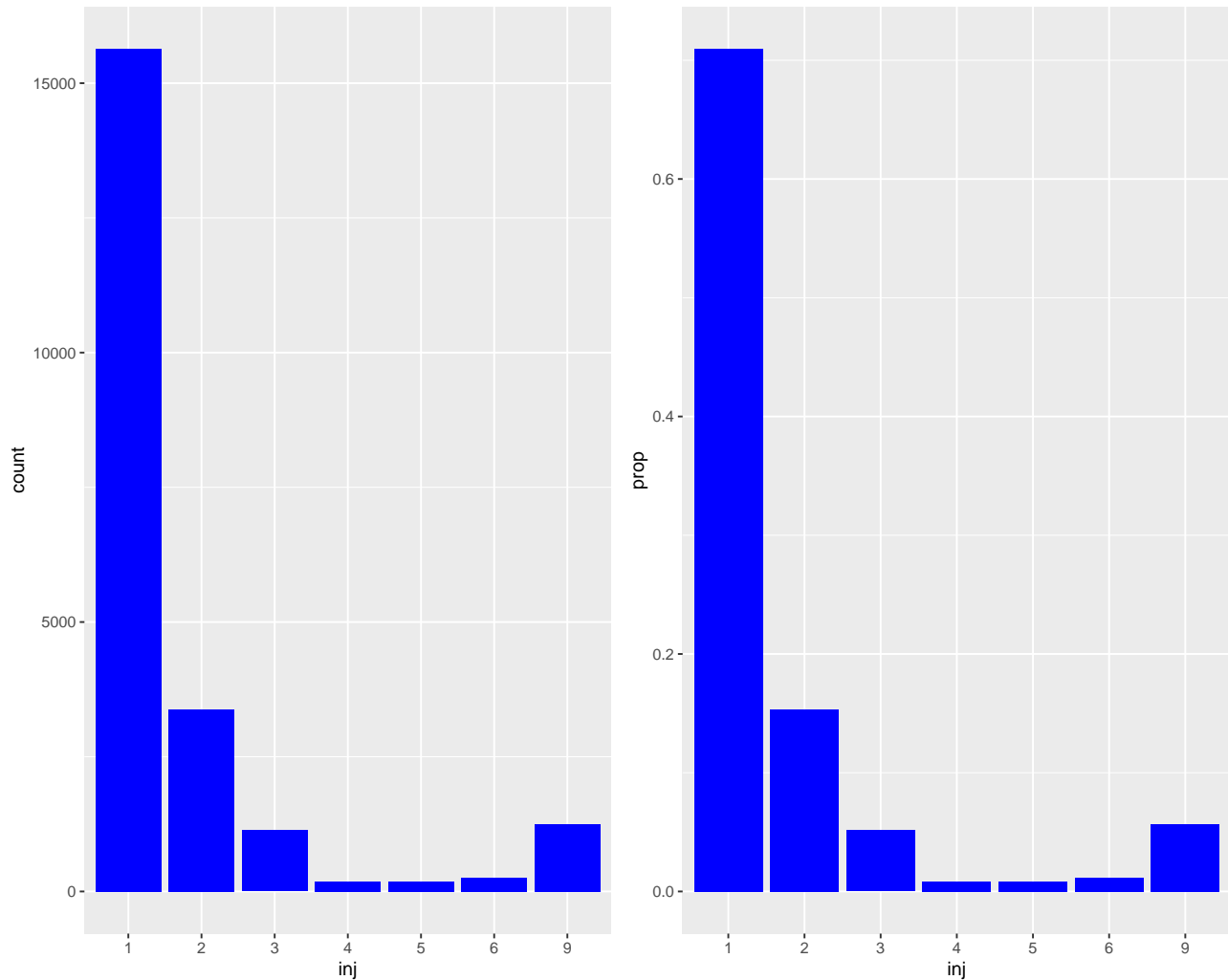


Figure 2.2.7: Bar charts of injury code in the `persinj` dataset.



## Case 2: Given summarized data

In real applications, it is not uncommon that the data we have has been grouped, or summarized, in some form in advance, to make it more manageable. In the case of the `persinj` dataset, instead of having the information for each individual claim, the data may have already been grouped by certain categorical variables, such as `inj`. CHUNK 9 below produces a version of `persinj` dataset, called `persinj_by_inj`, that shows the number of observations for each level of `inj`. The code involves the `tidyverse` package, which was featured in some past PA exams. Instead of worrying about the somewhat convoluted code syntax (which will be covered in Exam ATPA), try to pay attention to the output, which is far more important in the new exam format.

```
# CHUNK 9
# Uncomment the next line the first time you use the tidyverse package
# install.packages("tidyverse")

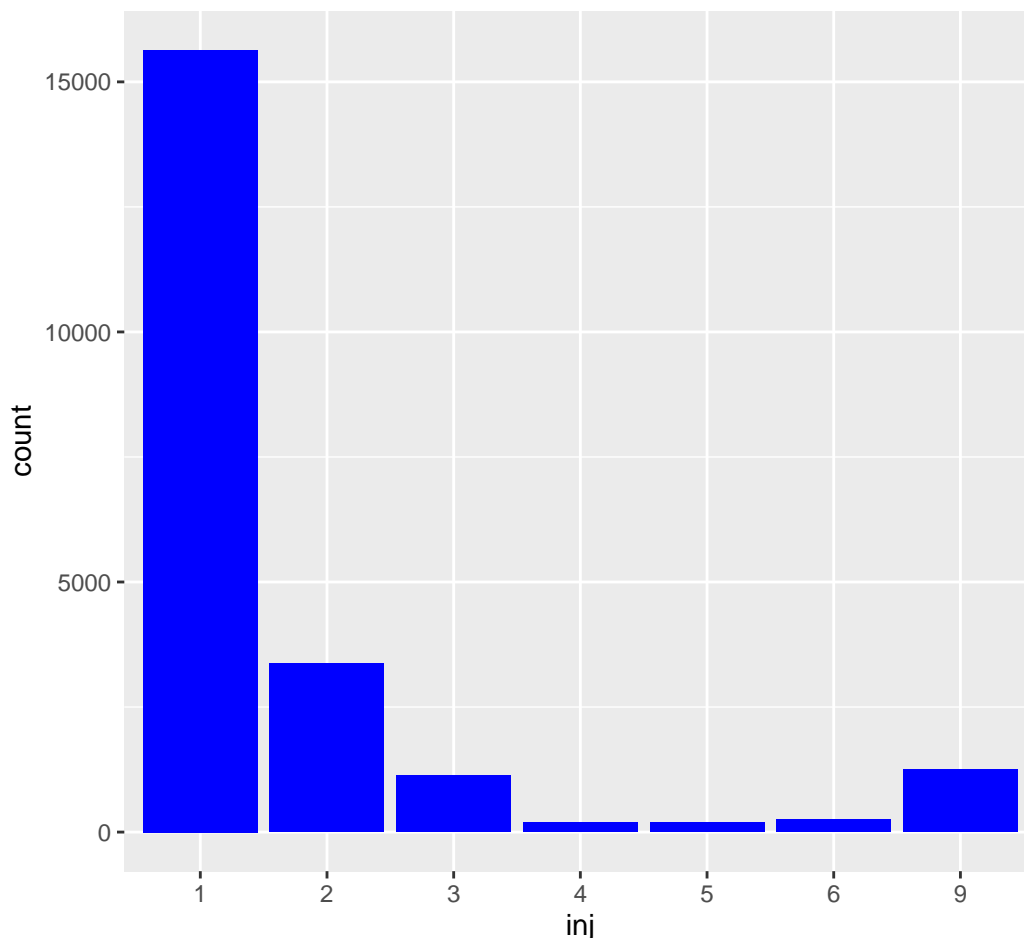
library(tidyverse)
persinj_by_inj <- persinj %>%
  group_by(inj) %>%      # grouped by inj
  summarize(count = n()) # count the no. of observations for each level of inj
persinj_by_inj

## # A tibble: 7 x 2
##   inj   count
##   <fct> <int>
## 1 1     15638
## 2 2      3376
## 3 3      1133
## 4 4       189
## 5 5       188
## 6 6       256
## 7 9      1256
```

The output shows, for example, that there are 15,638 observations with an injury code of 1, consistent with the frequency table in CHUNK 7.

If all we have is the `persinj_by_inj` dataset (we no longer have access to the original `persinj` dataset), how can we display the counts for each injury code? The `geom_bar()` function will not work well. If we map the `inj` variable to the `x` aesthetic of the `geom_bar()` function, then the function will keep track of how many times each distinct value of `inj` has occurred. It will (faithfully!) treat `inj` as a variable with 7 distinct values, 1, 2, 3, 4, 5, 6, and 9, each of which appears once and only once, which is definitely not what we want. Instead, the `geom_col()` function will suit our purpose. When `inj` is mapped to the `x` aesthetic and `count` to the `y` aesthetic, the function will display the counts for each value of `inj`, as CHUNK 10 shows.

```
# CHUNK 10
ggplot(persinj_by_inj, aes(x = inj, y = count)) +
  geom_col(fill = "blue")
```



This bar chart is identical to the one in the left panel of Figure 2.2.7 based on the original version of the `persinj` dataset.

In summary:

The `geom_bar()` function is for visualizing the distribution of a categorical variable given individual (raw) data, while the `geom_col()` function is for the same purpose, given grouped (summarized) data.

## 2.2.2 Bivariate Data Exploration

Data exploration becomes even more intriguing and challenging when two or more variables are analyzed together rather than in isolation. This has the important advantage of revealing relationships, patterns, and outliers which become apparent only when variables are considered in combination with one another. This subsection therefore focuses on *bivariate data exploration*, where pairs of variables are investigated either numerically or graphically to identify potentially interesting relationships that can provide useful input for a predictive model. Of particular interest is the relationship between the target variable and each predictor variable in a given setting.

There are three types of bivariate combinations, depending on the type of variables.

### Combination 1: Numeric vs. numeric

- *Descriptive statistics:* An easy way to summarize the *linear* relationship between two numeric variables is through the *correlation coefficient*, or simply *correlation*, which is a unit-free metric on a scale from  $-1$  to  $+1$ , as we learned from Exam P. (To be precise, in Exam PA we are looking at *sample* correlations computed from data rather than population correlations.)

*Case 1.* If the correlation is  $+1$ , then the two variables are perfectly positively correlated.

*Case 2.* If the correlation is  $0$ , then the two variables are uncorrelated.

*Case 3.* If the correlation is  $-1$ , then the two variables are perfectly negatively correlated.

These extreme correlation values almost never arise in real datasets, but they provide useful benchmarks for judging the size of a typical correlation. The larger the correlation in magnitude (i.e., the closer they are to  $+1$  or  $-1$ ), the stronger the degree of linear association between the two variables.

In CHUNK 11, we use the `cor()` function to compute the correlation between the claim amount and operational time, and between the log-transformed claim amount and operational time in the `persinj` data.

```
# CHUNK 11
cor(persinj$amt, persinj$op_time)
## [1] 0.3466114
cor(log(persinj$amt), persinj$op_time)
## [1] 0.6070667
```

The two variables are moderately positively correlated on the original scale and the correlation becomes stronger when the claim amount is on the log scale.

As much as correlation is a compact summary of the extent to which two numeric variables move in tandem, it can only capture *linear* relationships. A zero correlation only means that two variables are not linearly related, but they may be related in more subtle ways. More complex relationships (e.g., quadratic) can be revealed more effectively by graphical displays.

**Exercise 2.2.3.** (Motivated from Exam P Sample Question 320: What can you say about two variables with a zero correlation?) Consider the following dataset with two variables:

```
# CHUNK 12
X <- c(-1, 0, 1)
Y <- c(1, 0, 1)
```

Determine which of the following statements about the two variables is true.

- (A) The correlation between  $X$  and  $Y$  is positive; they are dependent.

- (B) The correlation between  $X$  and  $Y$  is positive; they are unrelated.
- (C) The correlation between  $X$  and  $Y$  is zero; they are dependent.
- (D) The correlation between  $X$  and  $Y$  is zero; they are unrelated.
- (E) The correlation between  $X$  and  $Y$  is negative; they are dependent.

*Solution.* In the rest of CHUNK 12, we compute the (sample) correlation between  $X$  and  $Y$ :

```
# CHUNK 12 (Cont.)
cor(X, Y)

## [1] 0
```

The zero correlation suggests that  $X$  and  $Y$  are *linearly* unrelated. However, the two variables are perfectly dependent via the quadratic relationship  $Y = X^2$ . **(Answer: (C))**  $\square$

*Remark.* This toy example illustrates the pitfall of using the correlation to detect useful predictors. If  $Y$  is the target variable and  $X$  is a potential predictor, then  $X$  may seem to be a predictor with limited predictive power based on its zero correlation with  $Y$ . To appreciate the relationship between the two variables more closely, a graphical tool like a scatterplot, which we discuss next, is useful.

- *Graphical displays:* The relationship between two numeric variables (one of which is usually the target variable) is typically visualized by a **scatterplot**, where values of the two variables are graphed on a two-dimensional plane, as we saw in Section 2.1. Such a plot often gives us a good sense of the nature of the relationship (e.g., increasing, decreasing, polynomial, periodic) between the two numeric variables and sometimes yields insights that correlations alone cannot provide. This explains why scatterplots are one of the most commonly used graphical displays in data exploration.

Run CHUNK 13 to make two scatterplots, one for claim amount and one for the log of claim amount, both against operational time (see Figure 2.2.8). We have set `alpha` to a very small value due to the large number of overlapping observations.

Both plots exhibit an increasing relationship, but the scatterplot for the log of claim amount displays a much more conspicuous upward sloping trend, indicating that the log of claim amount is approximately positively linear in operational time. This is a further manifestation of the merits of the log transformation in uncovering relationships that would otherwise be obscure.

```
# CHUNK 13
p1 <- ggplot(persinj, aes(x = op_time, y = amt)) +
  geom_point(alpha = 0.05) +
  geom_smooth(method = "lm", se = FALSE)
p2 <- ggplot(persinj, aes(x = op_time, y = log(amt))) +
  geom_point(alpha = 0.05) +
  geom_smooth(method = "lm", se = FALSE)
grid.arrange(p1, p2, ncol = 2)
```

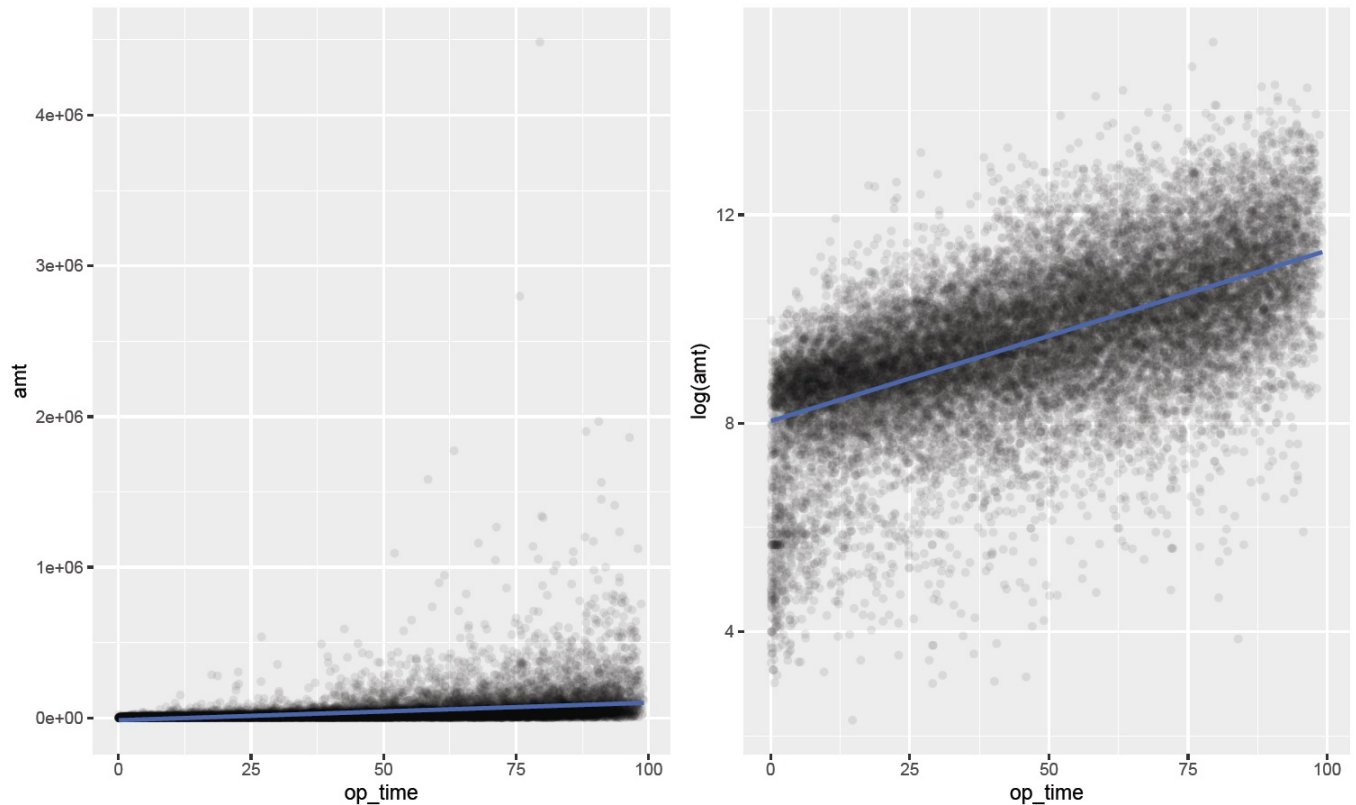


Figure 2.2.8: Scatterplots of claim amount (left) and the log of claim amount (right) against operational time in the `persinj` dataset.

Although a scatterplot itself is confined to depicting the relationship between only two numeric variables, the effect of a third, categorical variable can be incorporated and investigated by decorating the observations by color, shape, or other visual elements according to the levels assumed by this third variable. This way, we can visually inspect whether the relationship between the two numeric variables varies with the levels of the third, categorical variable. In statistical language, this phenomenon is known as *interaction*, which we will study in Section 3.2, and is an important modeling issue to keep in mind when constructing an effective predictive model.

Now run CHUNK 14 to make a scatterplot for the log of claim amount against operational time, with the observations color-distinguished by legal representation (note that the `color`

aesthetic is mapped to `legrep`); see Figure 2.2.9. The scatterplot shows that the two smoothed lines corresponding to the two levels of `legrep` have markedly different slopes and intercepts (keep in mind that we are on the log scale, so a small change in the intercept and slope can matter a lot on the original scale). In other words, the linear relationship between the log of claim amount and operational time depends materially on whether legal representation is present or not. We can also roughly tell the effect of legal representation on the (log of) claim amount:

Injuries with legal representation (i.e., those with `legrep = 1`) tend to produce higher claim amounts, except when operational time is extraordinarily large (90 or higher), in which case legal representation does not seem to have a noticeable impact on claim amount.

In Section 4.2, we will formally assess the extent of interaction and construct a model that properly takes the interaction effect into account.

```
# CHUNK 14
```

```
ggplot(persinj, aes(x = op_time, y = log(amt), color = legrep)) +  
  geom_point(alpha = 0.25) + geom_smooth(method = "lm", se = FALSE)
```

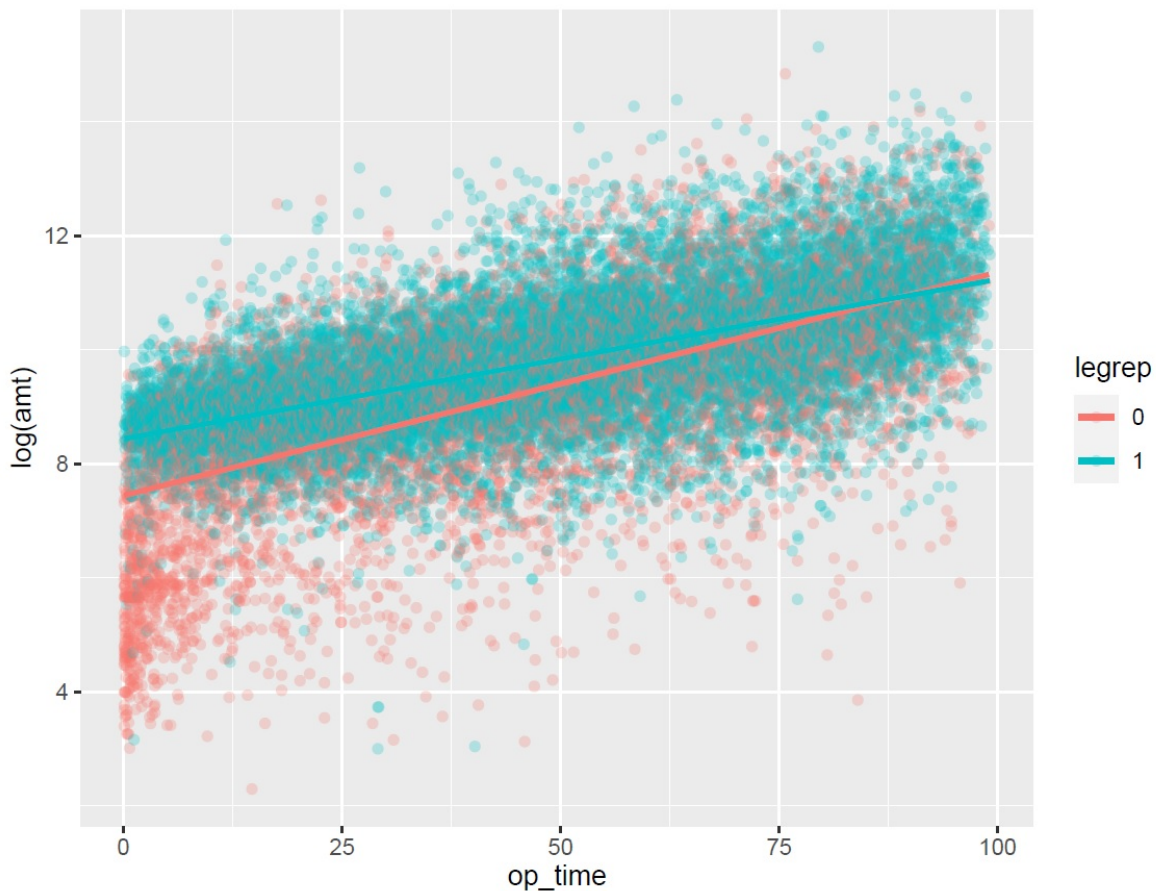


Figure 2.2.9: Scatterplot of the log of claim amount against operational time colored by legal representation in the `persinj` dataset.

## Combination 2: Numeric vs. categorical

To understand the interplay between a numeric variable and a categorical variable, it is best to investigate the distribution of the former indexed (or “split”) by each possible level of the latter. In effect, we are looking at the conditional distribution of the numeric variable given different levels of the categorical variable.

- *Descriptive statistics:* To summarize the association between a numeric variable and a categorical variable, we can partition the data into different subsets, one subset for each level of the categorical variable, and compute the mean of the numeric variable there. These conditional means varying substantially may suggest a strong relationship between the two variables.

In CHUNK 15 below, we produce a table of the mean of the log-transformed claim amount (it is also fine to use the untransformed claim amount variable) split by different levels of `inj` and `legrep`, which are categorical.

```
# CHUNK 15
library(tidyverse)
persinj %>%
  group_by(inj) %>%
  summarize(
    mean = mean(log(amt)),
    median = median(log(amt)),
    n = n()
  )

## # A tibble: 7 x 4
##   inj    mean median     n
##   <fct> <dbl> <dbl> <int>
## 1 1      9.37  9.36 15638
## 2 2     10.3  10.3  3376
## 3 3     10.7  10.9  1133
## 4 4     11.0  11.2   189
## 5 5     10.8  10.6   188
## 6 6      9.68  9.07   256
## 7 9      8.35  8.57  1256
```

```
# CHUNK 15 (Cont.)
persinj %>%
  group_by(legrep) %>%
  summarize(
    mean = mean(log(amt)),
    median = median(log(amt)),
    n = n()
  )

## # A tibble: 2 x 4
##   legrep mean median    n
##   <fct> <dbl> <dbl> <int>
## 1 0      9.18  9.32  8008
## 2 1      9.77  9.64 14028
```

It is clear that the claim amount on average increases from injury code 1 to injury code 4, then decreases down to injury code 9. The two means split by `legrep` are in agreement with what we observed in Figure 2.2.9, which shows that larger claim amounts are associated with the use of legal representation.



- *Graphical displays:* The conditional distribution of a numeric variable given a second, categorical variable is best visualized by *split boxplots*, where a series of boxplots of the numeric variable split by the categorical variable are made.

Run CHUNK 16 to construct two split boxplots for the log of claim amount, one split by injury code and one split by legal representation (Figure 2.2.10). The categorical variable that is used to split the numeric variable simply enters the `x` aesthetic and a collection of boxplots of the numeric variable for each level of the categorical variable will be shown. The two boxplots further support the findings based on the summary statistics above, but turn them more powerfully into diagrams.



```
# CHUNK 16
```

```
p1 <- ggplot(persinj, aes(x = inj, y = log(amt))) + geom_boxplot()
p2 <- ggplot(persinj, aes(x = legrep, y = log(amt))) + geom_boxplot()
grid.arrange(p1, p2, ncol = 2)
```

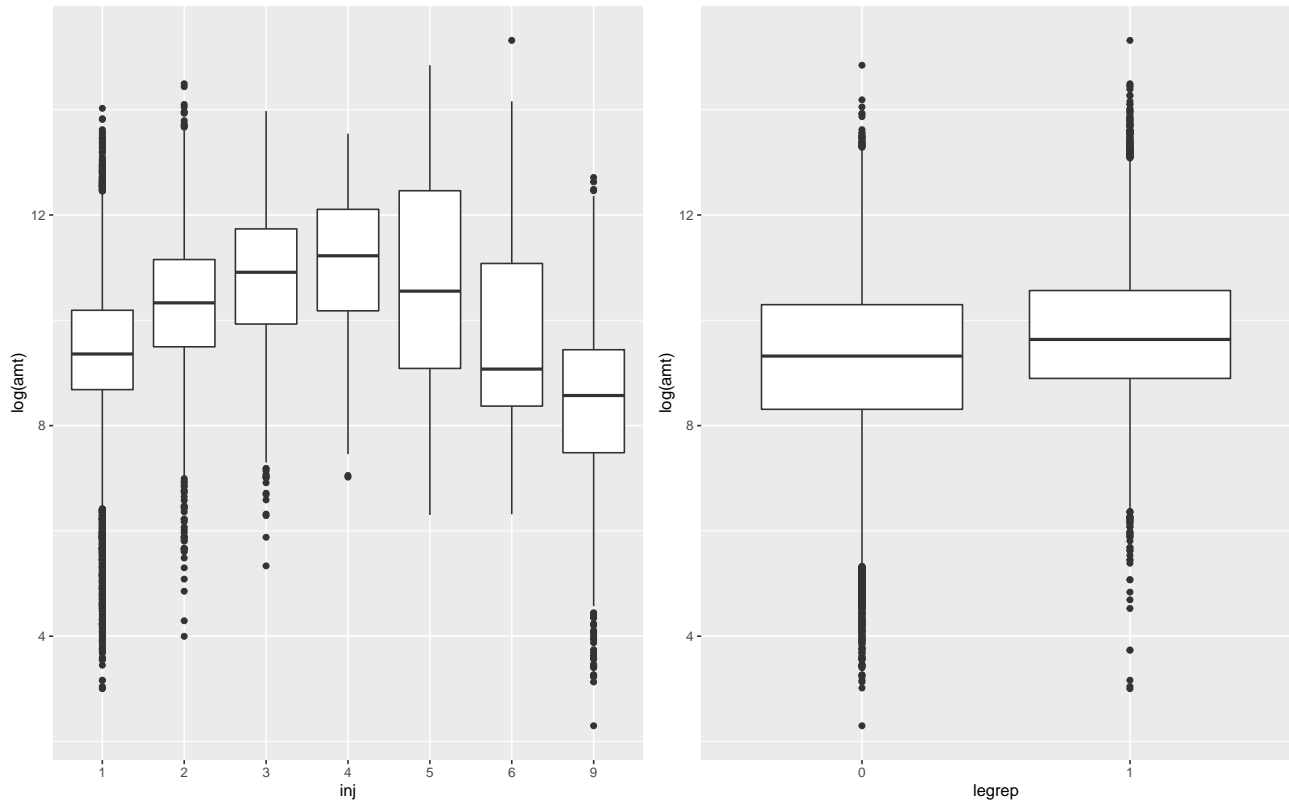


Figure 2.2.10: Two split boxplots for the log of claim amount, one split by injury code (left) and one split by legal representation (right), in the `persinj` data.

In CHUNK 17, we split the log of claim amount by injury code (the `x` aesthetic), followed by legal representation (the `fill` aesthetic) within each injury code, to view a *three-way relationship* (Figure 2.2.11). Now the effect of legal representation is even more pronounced: Regardless of the injury code, larger claim sizes tend to be injuries with legal representation, with the effect being the most prominent for injuries of code 5 and code 9.

```
# CHUNK 17
ggplot(persinj, aes(x = inj, y = log(amt), fill = legrep)) +
  geom_boxplot()
```

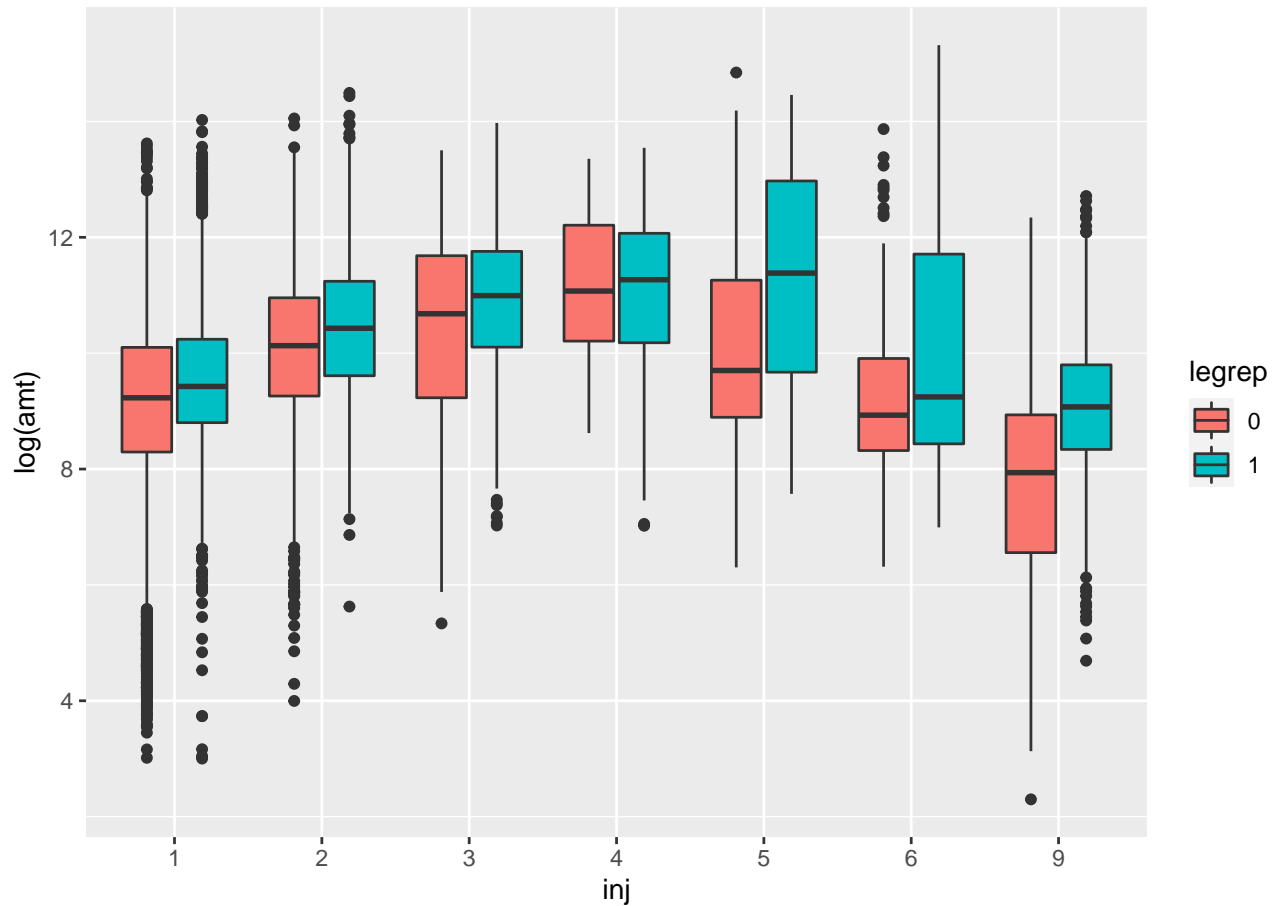


Figure 2.2.11: Boxplots of the log of claim amount split by injury code and legal representation in the `persinj` data.

Although not as effective as split boxplots (in my opinion), histograms can also be adapted to visualize the distribution of a numeric variable split by a categorical variable. They can either be histograms **stacked** on top of one another (using the `fill` aesthetic) to highlight the contribution of each categorical level to the overall distribution of the numeric variable, or **dodged** histograms with each bin placed side by side for comparison (note the argument `position = "dodge"`).

Run CHUNK 18 to produce both types of histograms for the log of claim amount split by legal representation (Figure 2.2.12). For the dodged histogram, it is necessary to specify `y = ..density..` so that the histogram shows the density rather than raw counts; counts are misleading because there are a lot more injuries with legal representation than those without. Both histograms suggest that larger claims (e.g., those with `log(amt)` greater than 9) tend to be those with legal representation, as expected.

```
# CHUNK 18
```

```
p1 <- ggplot(persinj, aes(x = log(amt), fill = legrep)) +  
  geom_histogram()  
p2 <- ggplot(persinj, aes(x = log(amt), y = ..density.., fill = legrep)) +  
  geom_histogram(position = "dodge")  
grid.arrange(p1, p2)
```

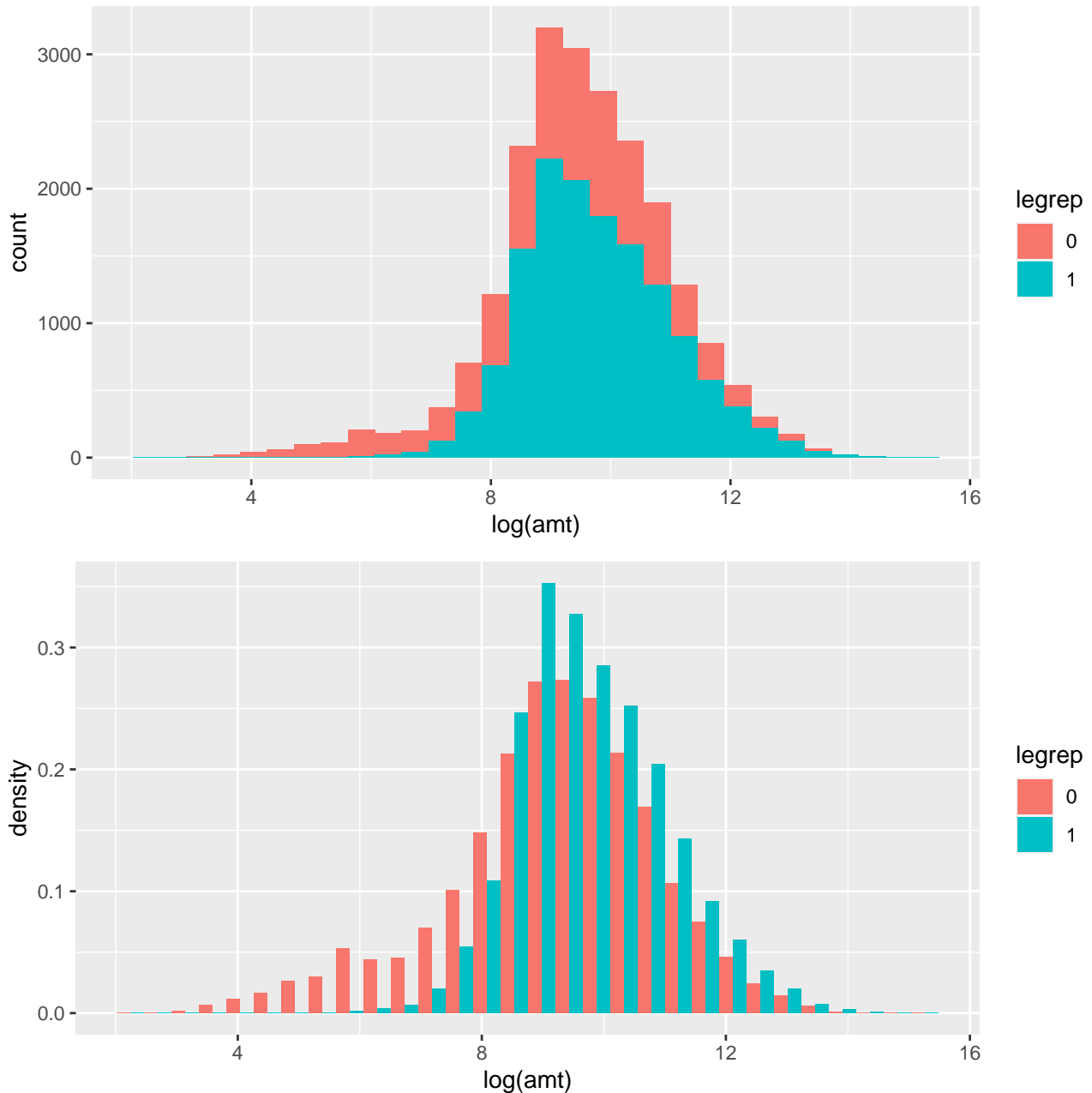


Figure 2.2.12: Stacked (top) and dodged (bottom) histograms of the log of claim amount in the persinj data.

### Combination 3: Categorical vs. categorical


- *Descriptive statistics:* When examining a pair of categorical variables, it is often useful to construct a two-way frequency table showing the number of observations for every combination of the levels of the two variables using the `table()` function. When two arguments are supplied to the `table()` function, the first argument will correspond to the rows of the two-way frequency table while the second argument will correspond to its columns.

Run CHUNK 19 to make a two-way frequency table for legal representation crossed with injury code.

```
# CHUNK 19
table(persinj$legrep, persinj$inj)

##
##      1      2      3      4      5      6      9
## 0 5571 1152  374   56   85  121  649
## 1 10067 2224  759  133  103  135  607
```

- *Graphical displays:* The proportions in a frequency table are useful statistics, but a visual picture often expresses these statistics much more powerfully and makes for easy interpretation, especially when the two categorical variables have a lot of levels. To visualize the distribution of a categorical variable split by another categorical variable effectively, *split bar charts* can be of use. These charts come in different versions, each one being useful for a certain purpose. Run CHUNK 20 to produce three bar charts for injury code split by legal representation (see Figure 2.2.13):
  - ▷ *Stacked:* The first bar chart has counts within each injury code colored by legal representation because the `fill` aesthetic is set to `legrep`. In other words, each bar is broken down proportionally into injuries with legal representation (colored in teal) and those without legal representation (colored in red).
  - ▷ *Dodged:* The second bar chart has counts within each injury code separated according to legal representation and placed side by side for comparison due to the option `position = "dodge"` in the `geom_bar()` function.
  - ▷ *Filled:* In the third bar chart, the relative proportions (not counts) of injuries with and without legal representation within each injury code are shown due to the option `position = "fill"` (not to be confused with the `fill` aesthetic). This makes it easy to compare proportions across different injury codes, although we lose the ability to see the number of injuries in each code. In predictive modeling, factor levels with more observations are generally considered more reliable than sparse levels.

Although not discussed in the PA modules, filled bar charts are usually the most useful for depicting the interplay between two categorical variables. A cursory glance  at the filled bar chart in Figure 2.2.13 shows that there is a higher proportion of injuries with legal representation for codes 1 to 4 than for codes 5, 6, and 9.

```
# CHUNK 20
p1 <- ggplot(persinj, aes(x = inj, fill = legrep)) +
  geom_bar()
p2 <- ggplot(persinj, aes(x = inj, fill = legrep)) +
  geom_bar(position = "dodge")
p3 <- ggplot(persinj, aes(x = inj, fill = legrep)) +
  geom_bar(position = "fill") +
  ylab("Proportion")
grid.arrange(p1, p2, p3, ncol = 2)
```

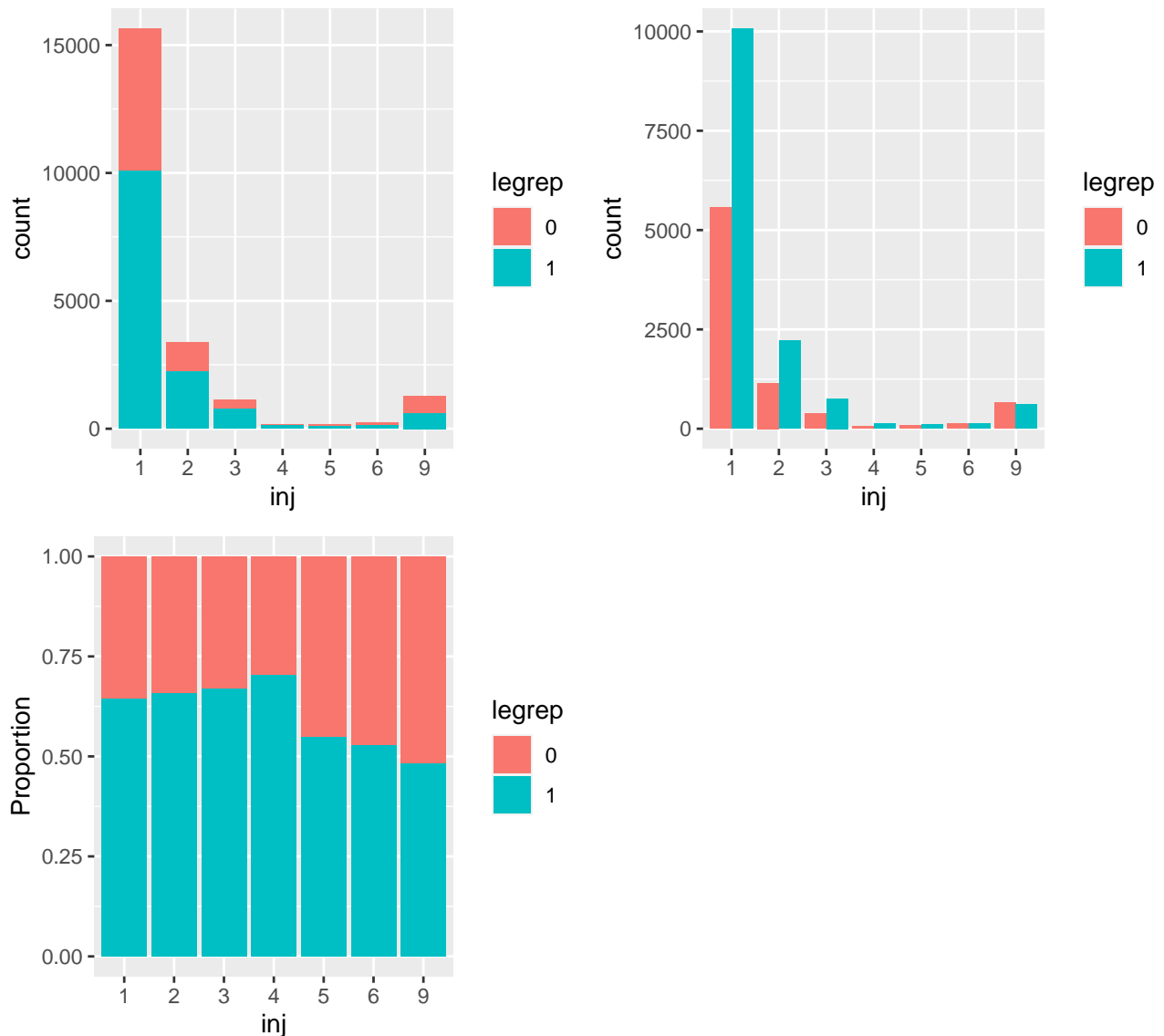


Figure 2.2.13: Stacked (top left), dodged (top right), and filled (bottom left) bar charts for injury code split by legal representation in the `persinj` data.

## 2.3 End-of-Chapter Practice Problems

**Problem 2.3.1.** 🧩 (Small differences in code, large differences in output!) Consider the personal injury insurance dataset again and the following chunks of R commands (which look so similar!):

```
persinj <- read.csv("persinj.csv")
persinj$inj <- as.factor(persinj$inj)
persinj$legrep <- as.factor(persinj$legrep)
library(ggplot2)

# CHUNK 1
ggplot(persinj, aes(x = inj, color = legrep)) +
  geom_bar()

# CHUNK 2
ggplot(persinj, aes(x = inj, fill = legrep)) +
  geom_bar()

# CHUNK 3
ggplot(persinj, aes(x = inj)) +
  geom_bar(fill = legrep)


# CHUNK 4
ggplot(persinj, aes(x = inj)) +
  geom_bar(aes(fill = legrep))
```

Make a guess of what each chunk of code does. Then run the code in R and see the output.

*Solution.* Although the four chunks of code look similar, they produce drastically different output.

- *CHUNK 1:* Here we are making a bar chart for injury code with the *boundary* (not the interior) of the vertical bars colored according to legal representation. As you can see, the colors are hardly perceptible.
- *CHUNK 2:* This is similar to CHUNK 1, except that this time it is the *interior* of the vertical bars that is color-coded according to legal representation. The output is the same as the top left panel of Figure 2.2.13.
- *CHUNK 3:* This chunk of code does not work (try to run it in R and you will get an error!). The reason is that the `fill` argument (not `fill` aesthetic) of the `geom_bar()` function is mapped to a variable (`legrep` here) instead of a constant (e.g., "blue"). This is the opposite of the mistake discussed in CHUNK 4 of Section 2.1 (Figure 2.1.2).
- *CHUNK 4:* This chunk of code generates the same output as CHUNK 2. Instead of

putting the `fill` aesthetic in the `ggplot()` call, it is placed inside the `geom_bar()` function. □

**Problem 2.3.2.**  (Data exploration: Univariate and bivariate) The `ggplot2` package comes with a dataset named `diamonds` that contains the prices and other attributes of approximately 54,000 diamonds. To load the dataset, use the following commands:

```
library(ggplot2)
data(diamonds)
```

- (a) Determine the number of observations and variables in the `diamonds` dataset.

With the aid of appropriate graphical displays and/or summary statistics, complete the following subtasks.

- (b) Perform univariate exploration of the price of diamonds (`price`) and the quality of the cut (`cut`). Determine if any of these two variables should be transformed and, if so, what transformation should be made. Do your recommended transformation(s), if any, and delete the original variable(s).
- (c) Explore the relationship between the price of diamonds and the weight of diamonds (`carat`).
- (d) Explore the relationship between the price of diamonds and the quality of the cut.
- (e) Reconcile the apparent contradiction between what you get in parts (c) and (d).

*Solution.* (a) The number of rows of the `diamonds` dataset can be obtained by the `nrow()` function:

```
nrow(diamonds)
## [1] 53940
```

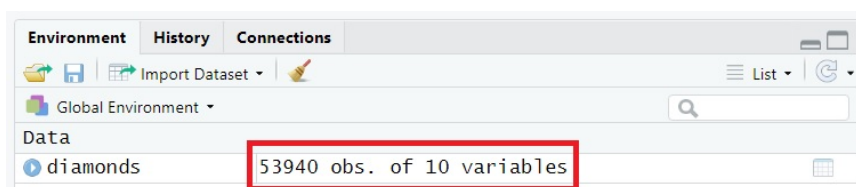
To get the number of variables, we can first extract the column names of the dataset via the `colnames()` function, then apply the `length()` function to calculate its length:

```
length(colnames(diamonds))
## [1] 10
```

More efficiently, we can apply the `dim()` function to `diamonds` to get both of its row and column dimensions.

```
dim(diamonds)
## [1] 53940    10
```

*Remark.* You can also see the number of observations and variables of the `diamonds` dataset directly from the environment pane:



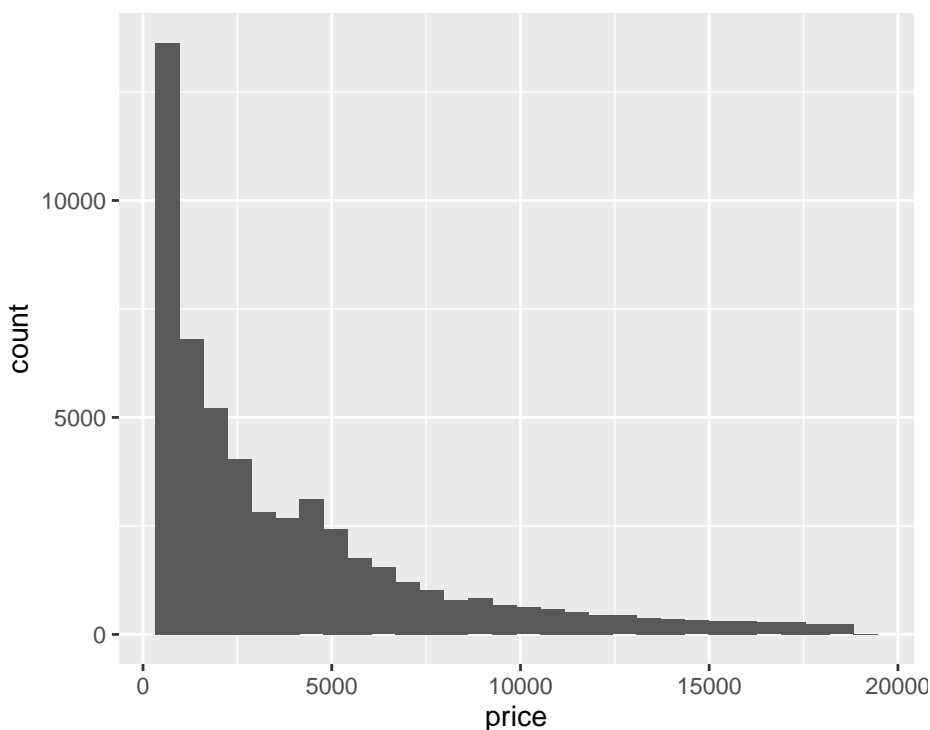
However, it is desirable to know how to write code to extract these attributes of a dataset.

- (b) • The `price` variable is a (positive and technically continuous) numeric variable. Let's use the `summary()` function to learn about its numeric statistics.

```
summary(diamonds$price)
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      326    950    2401   3933   5324   18823
```

The variable ranges from 326 to 18,823 and its mean is much higher than its median, an indication of its pronounced right skewness. This is confirmed by the histogram below.

```
ggplot(diamonds, aes(x = price)) +
  geom_histogram()
```



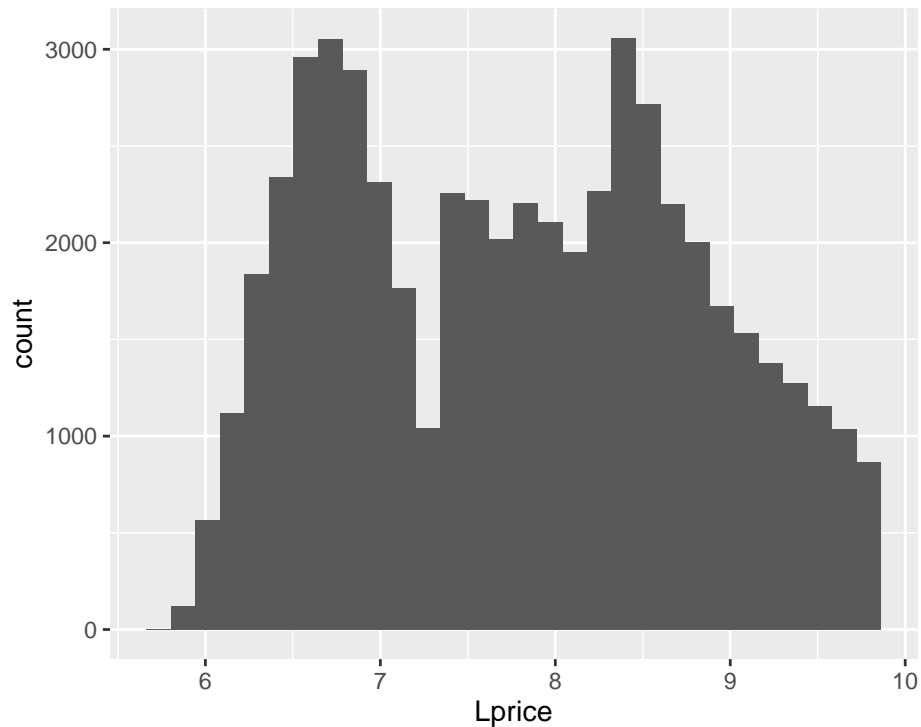
To deal with the right skewness of `price`, we can use a log transformation. The following commands create the log-transformed `price` and delete the original `price` variable (recall what you learned in Section 1.3).



```
diamonds$Lprice <- log(diamonds$price)
diamonds$price <- NULL
```

The resulting histogram shows that the distribution of the log-transformed price is closer to symmetric, although it is not bell-shaped.

```
ggplot(diamonds, aes(x = Lprice)) +
  geom_histogram()
```



*Remark.* For both histograms, you can experiment with different values of the `bins` parameter.

- The `cut` variable is a 5-level categorical variable; its levels are "Fair", "Good", "Very Good", "Premium", and "Ideal".

```
levels(diamonds$cut)
## [1] "Fair"      "Good"      "Very Good" "Premium"   "Ideal"
```

The following table shows the counts and percentage for each level:

```
table(diamonds$cut)
##
##      Fair      Good Very Good      Premium      Ideal
##      1610     4906     12082     13791     21551
```

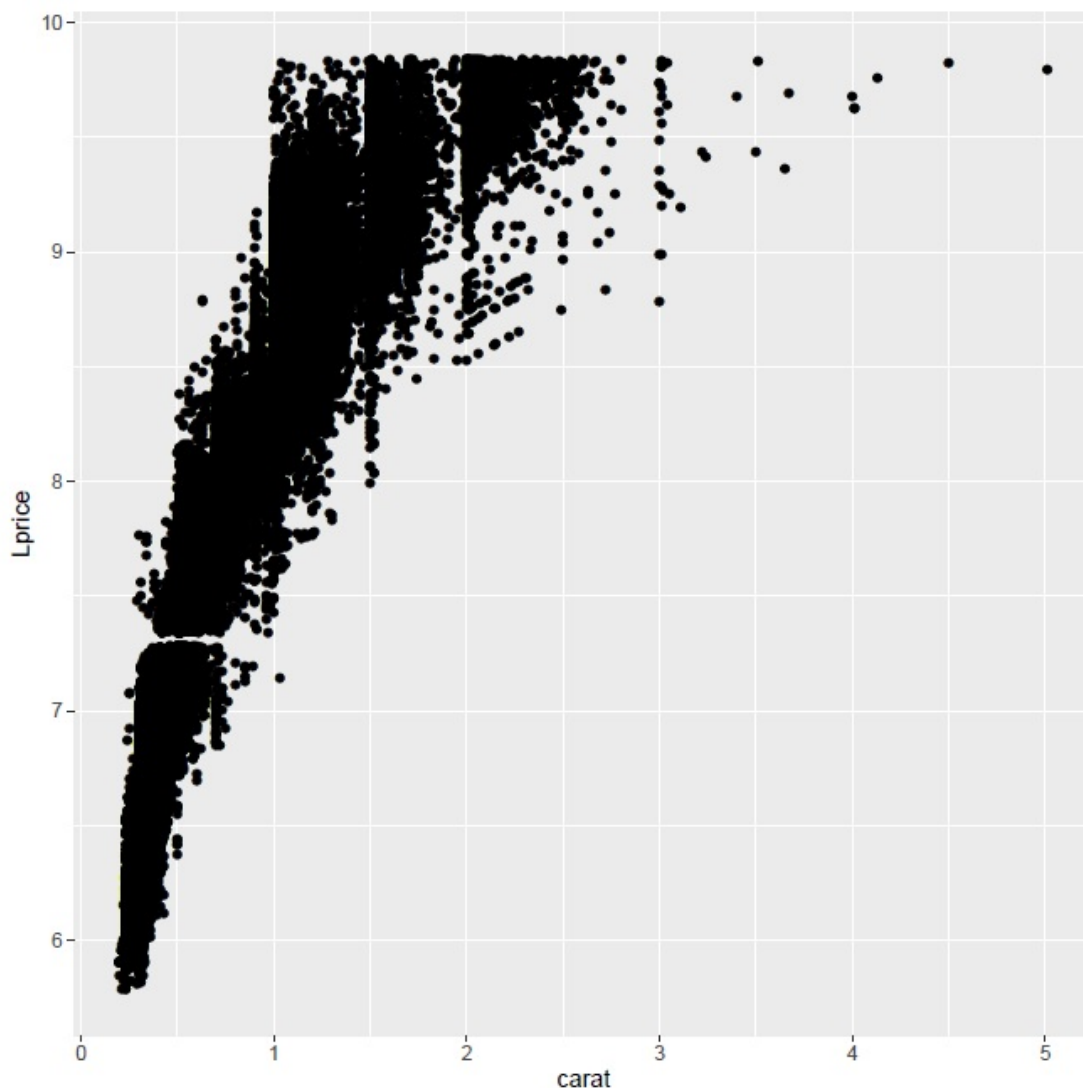
```
table(diamonds$cut)/nrow(diamonds)
```

```
##  
##      Fair      Good Very Good   Premium     Ideal  
## 0.02984798 0.09095291 0.22398962 0.25567297 0.39953652
```

The number of observations increases from "Fair" to "Ideal". About 40% of the diamonds are of "Ideal" quality.

Since cut is a categorical variable, numeric transformations such as the log transformation are not applicable and so we would prefer to leave it as it is.

- (c) Because `Lprice` and `carat` are both numeric variables, a scatterplot for the two variables is appropriate for exploring their relationship. The scatterplot shows that the two variables are strongly positively related; the heavier the diamond, the more expensive it is, conforming to our intuition.

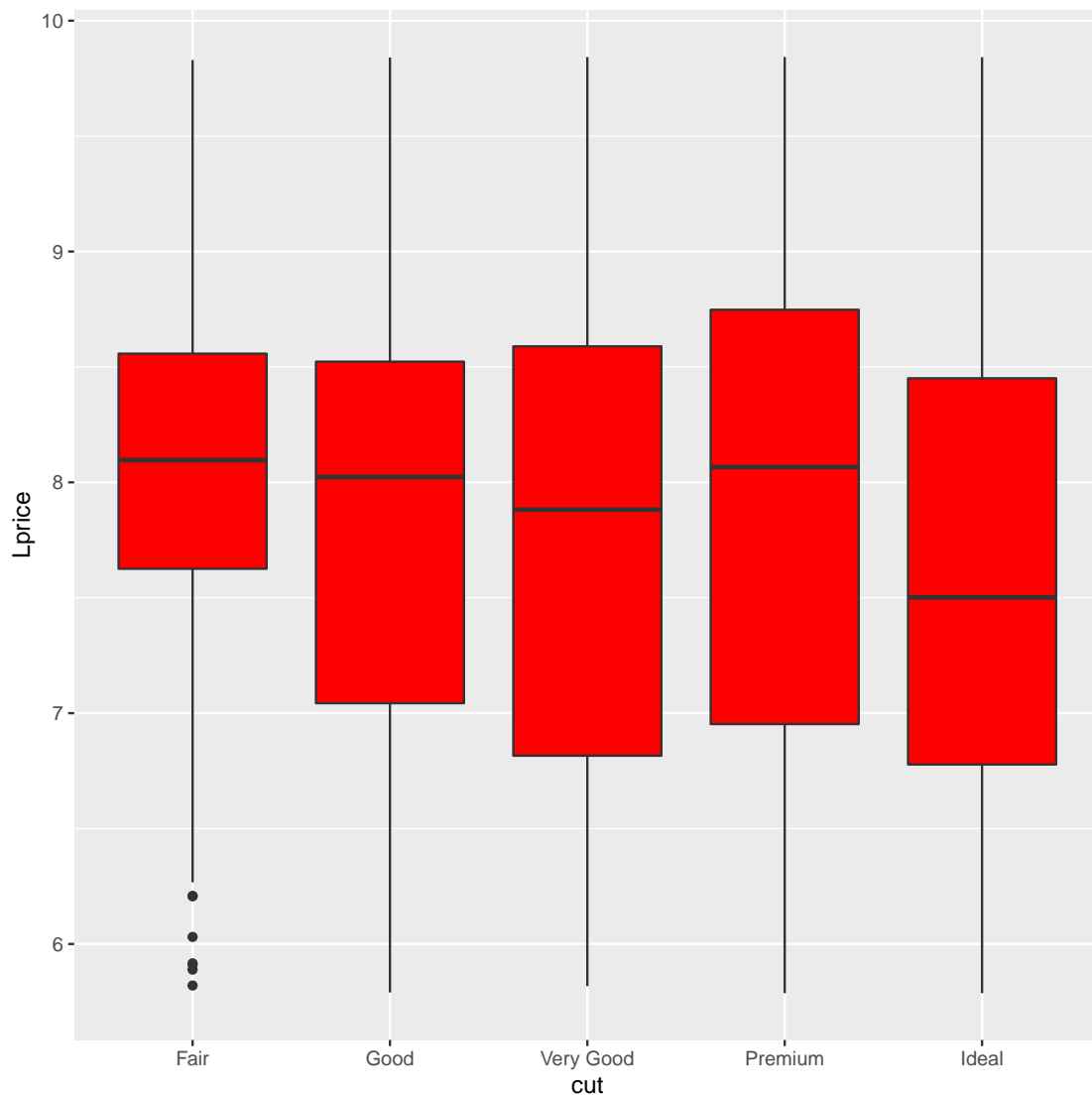


*Remark.* (i) You can add the `alpha` argument to the `geom_point()` function to reduce the amount of overlapping.

(ii) If you plot `Lprice` against the log of `carat`, the resulting relationship is very close to linear.

(d) As `cut` is a categorical variable, a split boxplot for `Lprice` broken by the levels of `cut` is appropriate for exploring their relationship. The split boxplot, however, suggests the counter-intuitive idea that diamonds of a higher quality tend to be cheaper (though only slightly). How can this be the case?

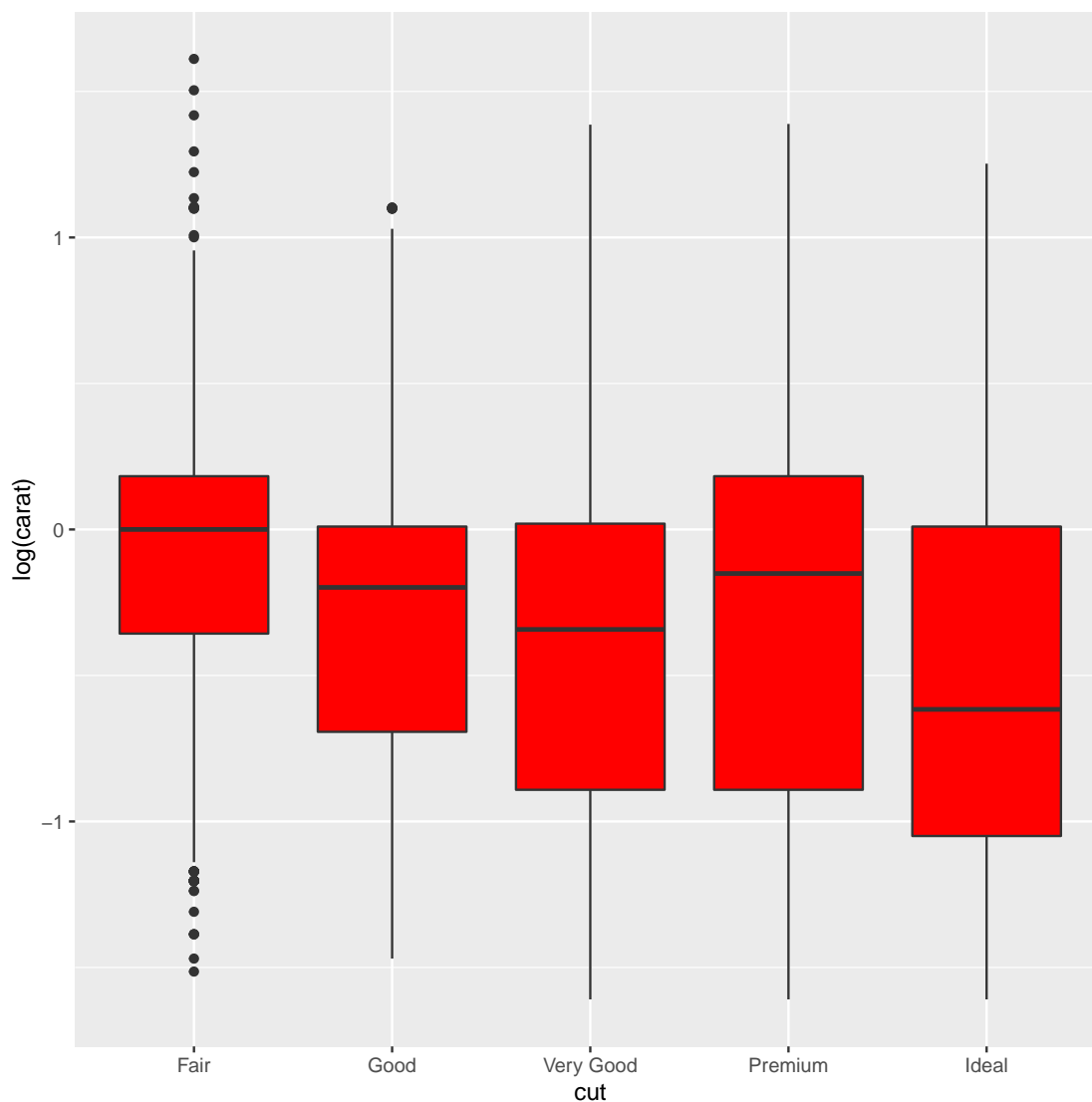
```
ggplot(diamonds, aes(x = cut, y = Lprice)) +  
  geom_boxplot(fill = "red")
```



(e) To reconcile the contradiction between parts (c) and (d), one can look at the relationship

between `carat` and `cut`. Again, as `carat` is numeric and `cut` is categorical, a split boxplot for `carat` split by `cut` will serve our purpose.

```
ggplot(diamonds, aes(x = cut, y = log(carat))) +  
  geom_boxplot(fill = "red")
```



The split boxplot shows that the weight of a diamond tends to drop as the quality of the cut becomes higher ("Premium" is an exception). According to part (c), the `carat` is an important predictor of `Lprice`, so the findings in part (d) may be a result of the negative relationship between `carat` and `cut`—higher quality diamonds may be less pricey because they weigh less. □

# Chapter 8

## Practice Exams

### Introduction

Having been well *trained* on the core of this study manual (Chapters 1 to 6) and past PA exam projects (Chapter 7), you need to be exposed to unseen *tests* to avoid overfitting and identify areas in which you need more *training*. To this end, please make good use of the two substantially updated practice exams in this chapter, one on classification and one on regression. Each exam comes with the following resources:

- (1) A project statement describing a business problem, a data dictionary, and a series of tasks you have to complete

According to the PA exam syllabus,

“A hardcopy of the problem statements will *not* be available at Prometric testing centers. The statements will be available for the entirety of the exam on-screen.”

This is rather unfortunate because when I took the exam in December 2019, having a printed statement to look at helped quite a lot.

- (2) (*Available for download on Actuarial University as a separate file*) A Microsoft Word document with spaces labeled as “**ANSWER:**” for you to write your responses to each specific subtask when you practice

On the real exam, this Word document and the project statement are the same file. In other words, you will enter your responses directly in the project statement, similar to FSA written-answer exams. This is the only file you will submit for grading.

- (3) Detailed illustrative solutions with sample responses and related learning outcomes from the PA exam syllabus identified<sup>1</sup>

---

<sup>1</sup>Not all subtasks conveniently fit into the learning outcomes in the syllabus, but they still lie within the general scope of Exam PA.

**⚠ NOTE ⚠**

In addition to Practice Exams 1 and 2, we have introduced a graded mock exam product with completely different questions, which is available for separate purchase. Please refer to page xxvii of the preface or check out <https://www.actexlearning.com/exams/pa/exam-pa-mock-exam> for more details.

**What are these two practice exams like?**

Designed taking the new exam format effective from April 2023 and the style of recent PA exams into account, these two practice exams give you a holistic review of the entire PA exam syllabus and have the following characteristics:

- They consist of 7 to 9 tasks,<sup>2</sup> with a total of **70 points**. Some tasks are longer and some shorter. Almost all tasks are further broken down into a few subtasks. Exam points are provided for each subtask, so you have some idea of how much you should write. As I mentioned in the preface of this manual, you should spend about 3 minutes per exam point.
- Following the exam format effective from the December 2021 sitting, different tasks are mutually independent and can be answered in any order (unless you have a special preference, you may simply start with Task 1). Even if you struggle in a certain task, you can proceed to the next task and start anew. You will not make data preparation or modeling decisions that affect the rest of the project. There are also no tasks about comparing the performance of models constructed in different tasks. (You may have to rank models and select the best model *within the same task*, however.)
- Like recent exams, there are a large number of conceptual or descriptive tasks testing your prior understanding of predictive analytic concepts (look for the verbs “Describe” and “Explain” in the question prompt). You can complete these tasks without looking at any R code or output, or referring to the business problem.
- (*New!*) Starting from the April 2023 sitting, R and RStudio will not be available on the exam, but as the PA exam syllabus says,

“all code and output relevant to the tasks will be provided as part of the exam materials.”

The two practice exams embrace this new format. In quite a few tasks (e.g., Tasks 1, 2, 3, 6, and 8 of Practice Exam 1), you are given some R output and asked to use the output to answer the questions. Sometimes R code is also provided (e.g., Task 5 of Practice Exam 1). You are expected to know what the code does to address some subtasks adequately. This is what Chapters 1 and 2, and the R-based case studies in the manual are for.

---







<sup>2</sup>Although recent exams seem to have more and more tasks (e.g., the April 2024 exam has 12 in total) of a silo nature, the two practice exams remain valuable and perfectly applicable resources as some tasks can be easily broken into shorter tasks. The total points are still fixed at 70.

- They strike a good balance between easy items testing topics regularly featured in past exams and harder, more unfamiliar items. As comprehensive as this study manual is, each PA exam will likely have a small number of unfamiliar tasks designed to identify the candidates that thrive on new, unseen exam tasks and are not overfitted to past exams. The harder items in the practice exams are in a similar vein.

(In fact, I am not surprised if members of the PA exam committee have access to this manual and deliberately test obscure things I did not discuss at length! 🤪)

### How to use these practice exams?

To make the most of these practice exams, here are my recommendations:

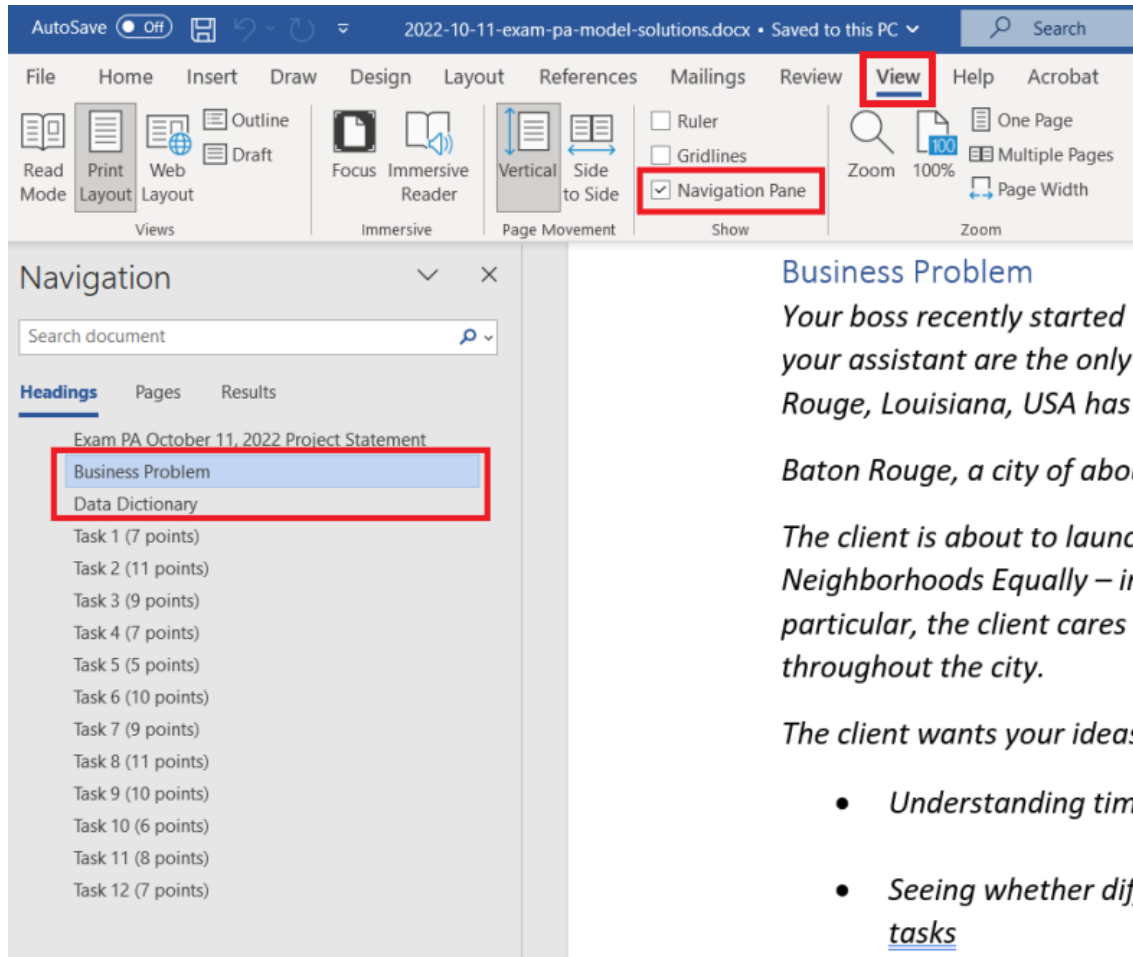
- Attempt them only when you have finished reading the study manual and studied recent PA exams (at least the April 2024, October 2023, and April 2023 exams). Working on the practice exams when you are not fully ready defeats their purpose.
- Set aside exactly 3 hours 30 minutes and work on each exam in a simulated exam environment detached from distractions. Put away your manual, notes, and phone—no Facebook , Instagram , Twitter , or Snapchat . You can only have your calculator  with you (you may also use Excel , which is available on the exam, to do calculations if you prefer). When you are finished, compare your responses with my suggested solutions and see how well you have done.
- Be sure to read the task statement and the Business Problem section carefully. Almost always, the Business Problem section has something useful for answering a few subtasks, and a seemingly minor point mentioned there can make a huge difference.
- Budget your time wisely. Don't spend a disproportionate amount of time on a single subtask, no matter how difficult it seems. As I mentioned in the preface, you should spend 3 minutes per exam point on average.

#### NOTE

Don't feel too frustrated if you find these two exams hard and long—they are probably (a bit) harder than the real exam! It is better to see something more difficult when you practice than to be defeated on the real exam, right? 😊

## A Note on Navigation

During the exam, you may want to scroll back to the Business Problem and Data Dictionary at the beginning of the Microsoft Word file, then continue to work on different tasks. To navigate back and forth efficiently, press **Ctrl+F**, or click **View > Navigation Pane**.



This may save you some time and trouble on the exam, where every second counts!



## ACTEX PA Manual Practice Exam 1 Project Statement

**IMPORTANT NOTICE – THIS IS THE PROJECT STATEMENT OF THE FIRST PRACTICE EXAM. IF YOU ARE NOT READY FOR IT YET, LEAVE IMMEDIATELY AND RETURN LATER.**

### General Information for Candidates

This examination has 9 tasks numbered 1 through 9 with a total of 70 points. The points for each task are indicated at the beginning of the task, and the points for subtasks are shown with each subtask.

Each task pertains to the business problem and data dictionary described below. Additional information on the business problem may be included in specific tasks—where additional information is provided, including variations in the target variable, it applies only to that task and not to other tasks. For this exam there is no data file or .Rmd file provided. Neither R nor RStudio are available or required.

The responses to each specific subtask should be written after the subtask and the answer label, which is typically ANSWER, in the separate Word document.<sup>3</sup> Each subtask will be graded individually, so be sure any work that addresses a given subtask is done in the space provided for that subtask. Some subtasks have multiple labels for answers where multiple items are asked for—each answer label should have an answer after it.

Each task will be graded on the quality of your thought process (as documented in your submission), conclusions, and quality of the presentation. The answer should be confined to the question as set. No response to any task needs to be written as a formal report. Unless a subtask specifies otherwise, the audience for the responses is the examination grading team and technical language can be used.

Prior to uploading your Word file, it should be saved and renamed with your five-digit candidate number in the file name. If any part of your exam was answered in French, also include “French” in the file name. Please keep the exam date as part of the file name.

The Word file that contains your answers must be uploaded before the five-minute upload period time expires.

---

<sup>3</sup>As mentioned before, you will write your responses directly in the project statement on the real exam.

## Business Problem

You work at ABC, a large actuarial consulting firm, and have been asked to assist School Wiz, a group dedicated to providing remedial education to troubled students. School Wiz has heard about you the legend, because you are among the very few who got Grade 10 in Exam PA, and wants to explore using your services to advance their business goals. They have collected preliminary data<sup>4</sup> of past high school students. They would like to be able to identify which of the incoming high school students have a high tendency to fail, before they enter their high school year. These students will receive remedial services in time.

School Wiz has determined that out of the three grade variables in the data, G1, G2, and G3, they would like you to just focus on building predictive models based on G3. A student who receives a grade of 10 or more will pass. Your goal is to use the available data to construct two models that will predict if a student will pass (rather than the overall grade). One model should be GLM-based and one should be tree-based.

School Wiz has provided the following data dictionary.

---

<sup>4</sup>This practice exam is based on the setting of the Student Success sample project (available from pages 8 and 9 of the [June 2021 PA exam syllabus](#)) and turns it into a much more useful task-based project consistent with the current exam format. The dataset for this sample project in turn is adapted from the Student Performance Data Set contributed by Paulo Cortez to the UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science.

**Data Dictionary**

<b>Name</b>	<b>Description</b>	<b>Variable Values</b>
sex	Student's sex	Binary: F (female) or M (male)
age	Student's age	Integer from 15 to 22
Medu	Mother's education	Integer from 0 (none) to 4 (higher education)
Fedu	Father's education	Integer from 0 (none) to 4 (higher education)
Mjob	Mother's job	Factor: at_home, health (health care related), other, services (civil services, administrative or police), teacher
Fjob	Father's job	Same levels as Mjob
studytime	Weekly study time	Integer from 1 (very short) to 4 (very long)
failures	Number of past class failures	Integer from 0 to 3
schoolsup	Extra educational support	Binary: yes or no
famsup	Extra family supplement	Binary: yes or no
paid	Extra paid classes	Binary: yes or no
activities	Extra-curricular activities	Binary: yes or no
internet	Internet access at home	Binary: yes or no
romantic	Has a romantic relationship	Binary: yes or no
famrel	Quality of family relationships	Integer from 1 (very bad) to 5 (excellent)
freetime	Free time after school	Integer from 1 (very low) to 5 (very high)
goout	Going out with friends	Integer from 1 (very low) to 5 (very high)
Dalc	Weekday alcohol consumption	Integer from 1 (very low) to 5 (very high)
Walc	Weekend alcohol consumption	Integer from 1 (very low) to 5 (very high)
absences	Number of absences in high school year	Integer from 0 to 75
G1	First trimester grade in high school year	Integer from 0 to 20
G2	Second trimester grade in high school year	Integer from 0 to 20
G3	Third trimester grade in high school year	Integer from 0 to 20
pass	Pass indicator	0 if a student fails and 1 if a student passes

**Task 1 (7 points)**

The following is the correlation matrix for **G1**, **G2**, and **G3**:

	G1	G2	G3
G1	1.0000000	0.8821056	0.8301591
G2	0.8821056	1.0000000	0.9151279
G3	0.8301591	0.9151279	1.0000000

- (a) (2 points) Describe one strength and one weakness of a correlation matrix as a bivariate data exploration tool.

**ANSWER:**

---

- (b) (2 points) Based on the correlation matrix, explain why basing pass or fail entirely on **G3**, as requested by School Wiz, may be a sensible decision.

**ANSWER:**

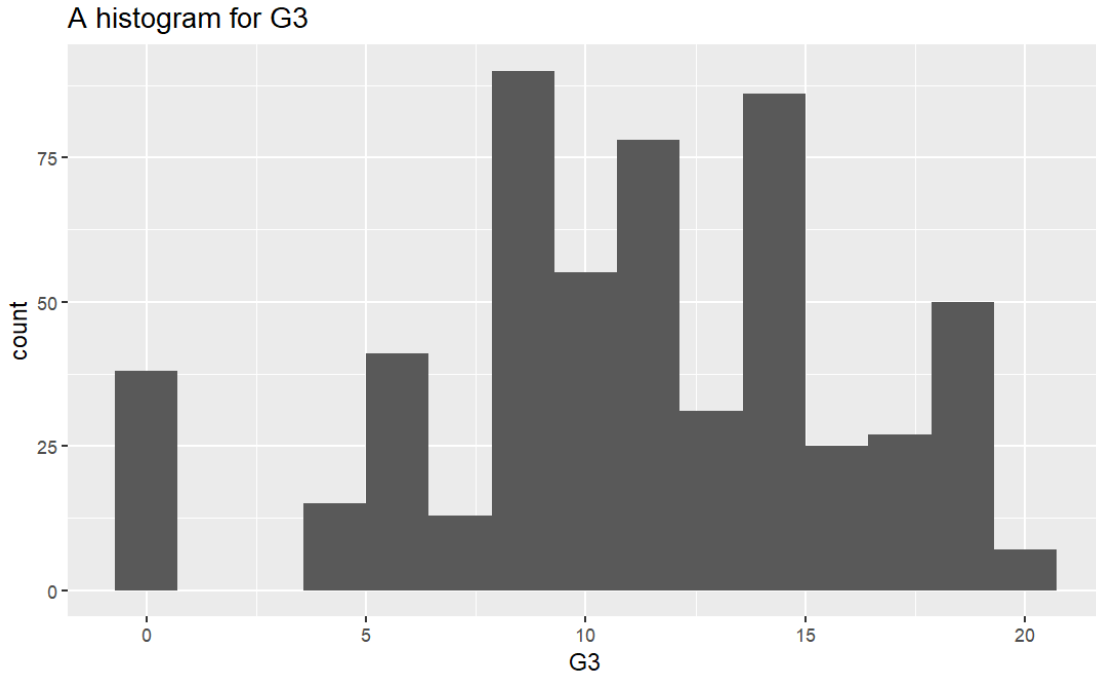
---

- (c) (3 points) Describe how principal components analysis can provide an alternative method for determining whether a student will pass or not based on all of **G1**, **G2**, and **G3**.

**ANSWER:**

**Task 2 (5 points)**

An alternative to modeling **pass** is to treat **G3** as the target variable and model it directly to determine pass or fail. To explore this alternative, your assistant has produced the following histogram for **G3**:



(a) (2 points) Describe the distributional characteristics of **G3**.

**ANSWER:**

---

(b) (3 points) Discuss one advantage and one disadvantage of modeling **G3** as the target variable over modeling **pass** for School Wiz from a GLM perspective.

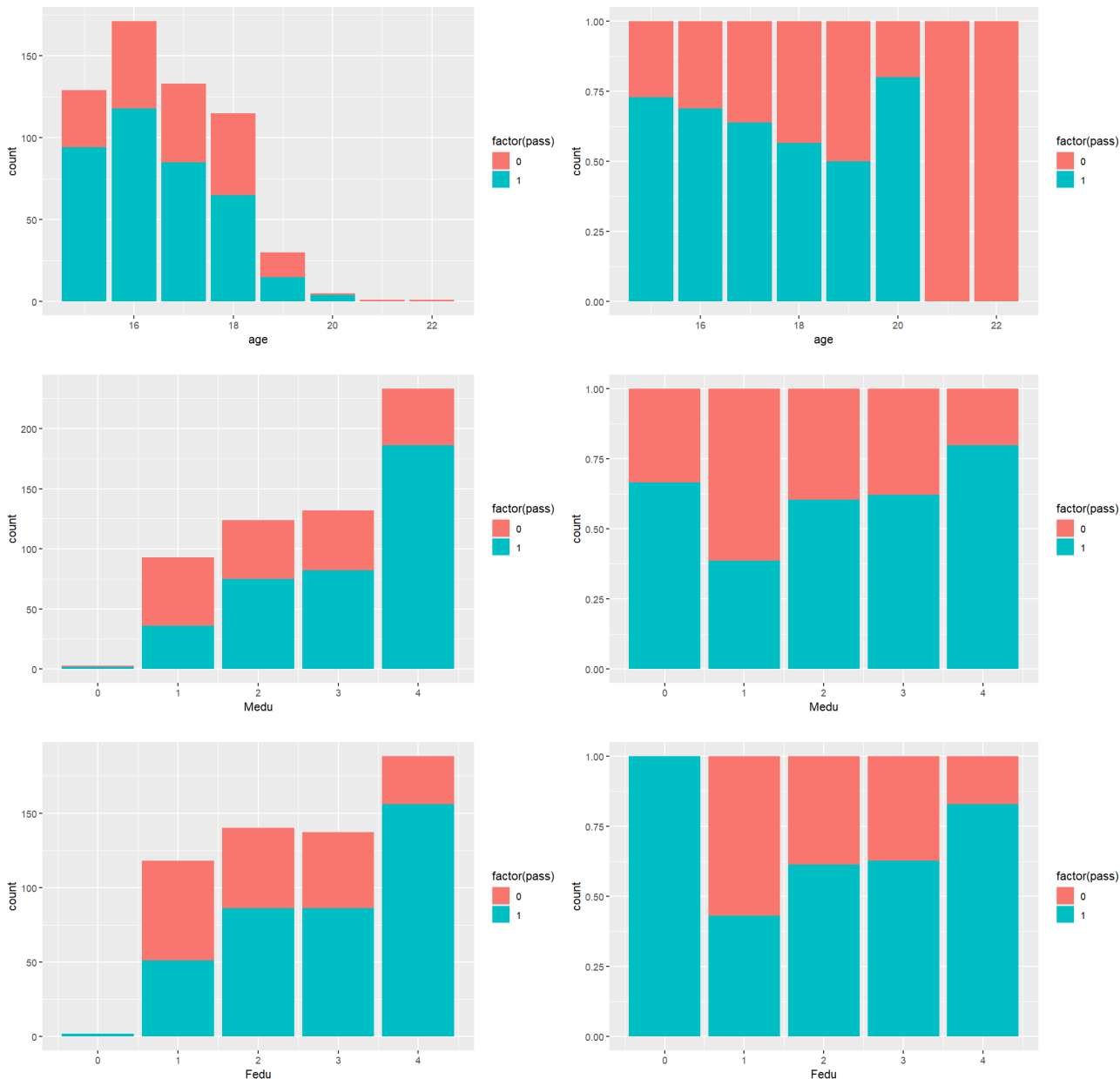
**ANSWER:**

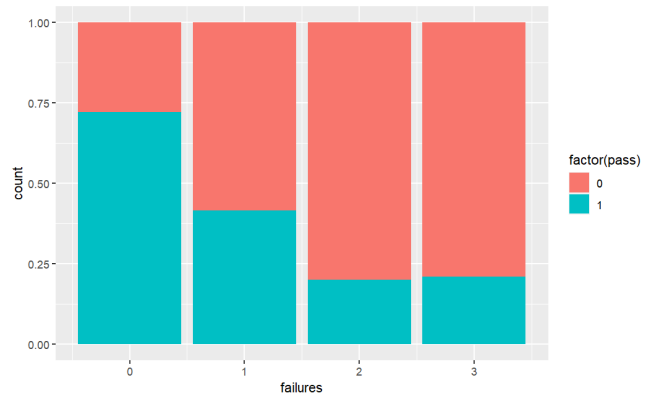
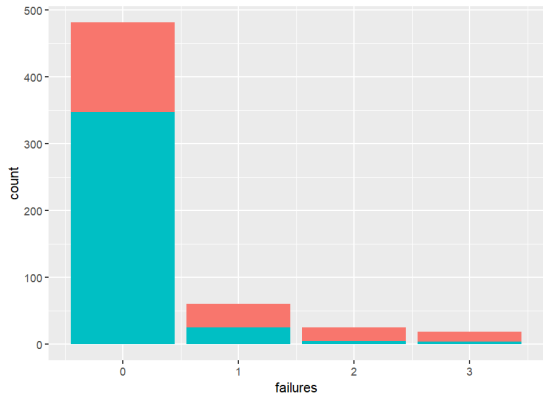
**Task 3 (7 points)**

You have asked your assistant to investigate the variables in the data.

(a) (2 points) Explain the problem with using **absences** for predicting **pass**.

Your assistant has conducted exploratory data analysis for **pass**. Here is part of the output:





(b) (3 points) Describe two anomalies of the data revealed by the output above.

**ANSWER:**

(c) (2 points) Identify and explain which variable above appears to be the most important predictor of **pass**.

**ANSWER:**

**Task 4 (5 points)**

Your assistant suggested creating a new variable that flags any previous class failures and including this flag variable in your models, in addition to variables that already exist in the data. The value of the variable is 1 if **failures** is higher than or equal to 1, and 0 if **failures** is 0. Your assistant thinks that this variable may be a useful feature for predicting **pass**.

- (a) (3 points) Explain the modeling impacts, if any, of adding the new flag variable when running a GLM.

**ANSWER:**

---

- (b) (2 points) Explain the modeling impacts, if any, of adding the new flag variable when running a decision tree.

**ANSWER:**



**Task 5 (9 points)**

Your assistant has provided the following R code to perform a certain cluster analysis.

```
data.hc <- data.all[, c("Medu", "Fedu")]
```

```
data.hc$Medu <- scale(data.hc$Medu)
```

```
data.hc$Fedu <- scale(data.hc$Fedu)
```

```
hc <- hclust(dist(data.hc))
```

(a) (2 points) Explain how cluster analysis can be used to develop features for a predictive model.

**ANSWER:**

---

(b) (3 points) Explain what kind of cluster analysis is performed by your assistant.

**ANSWER:**

---

(c) (2 points) Explain what the **scale()** function in your assistant's code does and why it is important.

**ANSWER:**

---

In retrospect, your assistant thinks that the code should have included a random seed so that the same output will be obtained every time the code is run. He apologizes for this omission.

(d) (2 points) Critique your assistant's statement.

**ANSWER:**

**Task 6 (12 points)**

Having read the *ACTEX Study Manual for Exam PA*, your assistant knows that accuracy, sensitivity, specificity, and AUC are commonly used performance metrics for a classifier.

(a) (3 points) Define accuracy, sensitivity, specificity, and AUC for a general classifier.

**ANSWER:**

**Accuracy:**

**Sensitivity:**

**Specificity:**

**AUC:**

---

(b) (4 points) Describe how accuracy, sensitivity, specificity, and AUC vary with the cutoff of a classifier.

**ANSWER:**

**Accuracy:**

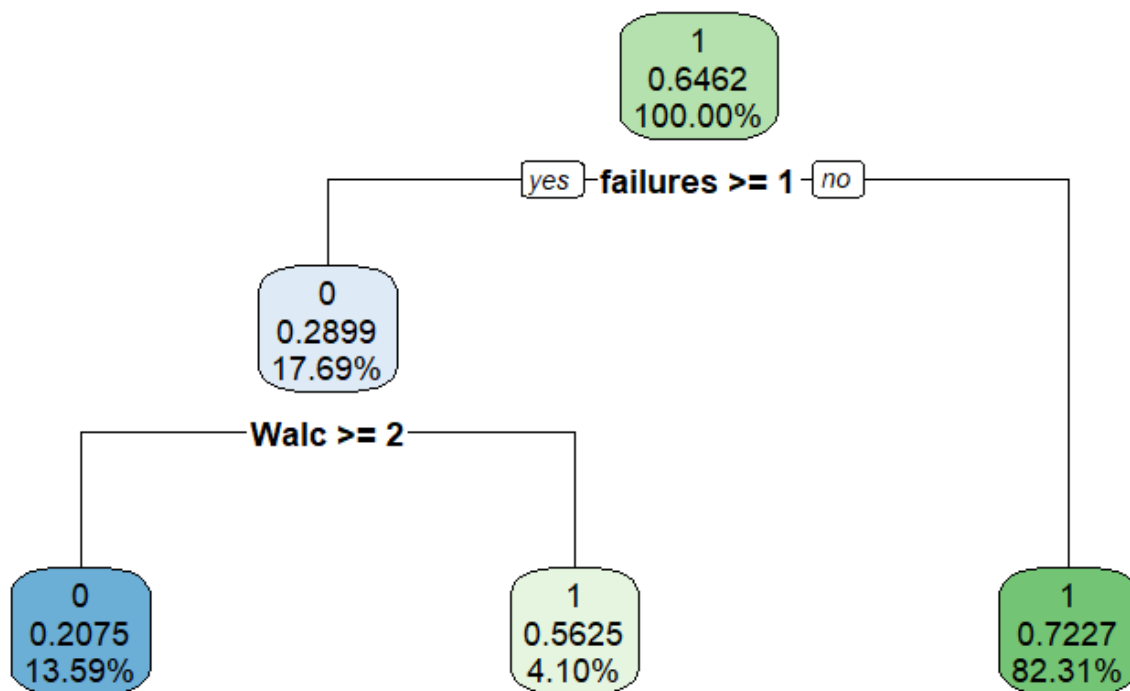
**Sensitivity:**

**Specificity:**

**AUC:**

---

Your assistant has fitted a classification tree for **pass** on a subset of the data containing 390 observations, resulting in the following tree.



(c) (5 points) Fill in the following confusion matrix for the classification tree based on a cutoff of 0.5. Show your work.

Prediction	Reference	
	0	1
0	?	?
1	?	?

**Task 7 (11 points)**

Your assistant has provided code to split the data into the training (70%) and test sets (30%).

- (a) (2 points) Describe the trade-off involved when selecting the percentages of data in the training and test sets.

**ANSWER:**

---

Your assistant has also set up code for fitting a regularized logistic regression model for **pass** on the training set.

- (b) (3 points) Explain why **lambda** and **alpha** in an elastic net are hyperparameters and how these two parameters affect an elastic net.

**ANSWER:**

**Why lambda and alpha are hyperparameters:**

**Lambda:**

**Alpha:**

---

- (c) (2 points) Describe the significance of using alpha equal to 1 in this business problem.

**ANSWER:**

---

The following shows the coefficient estimates of the variables selected in the resulting model:

	s0
(Intercept)	0.49996348
Medu	0.23399344
Fedu	0.09891187
Mjobother	-0.14728207
failures	-0.68078016
famsupyes	-0.51504749
goout	-0.07769020
Walc	-0.02494307

(d) (4 points) Interpret the estimates of the intercept, and the coefficients for the categorical variable and the numeric variable with the most significant impact on **pass**.

**ANSWER:**

**Intercept:**

**Coefficient for categorical variable:**

**Coefficient for numeric variable:**

**Task 8 (7 points)**

Your assistant has run a boosted classification tree for **pass** on the training set.

- (a) (3 points) Explain the role played by the **eta** and **nrounds** parameters in a boosted tree, and the considerations for selecting these two parameters.

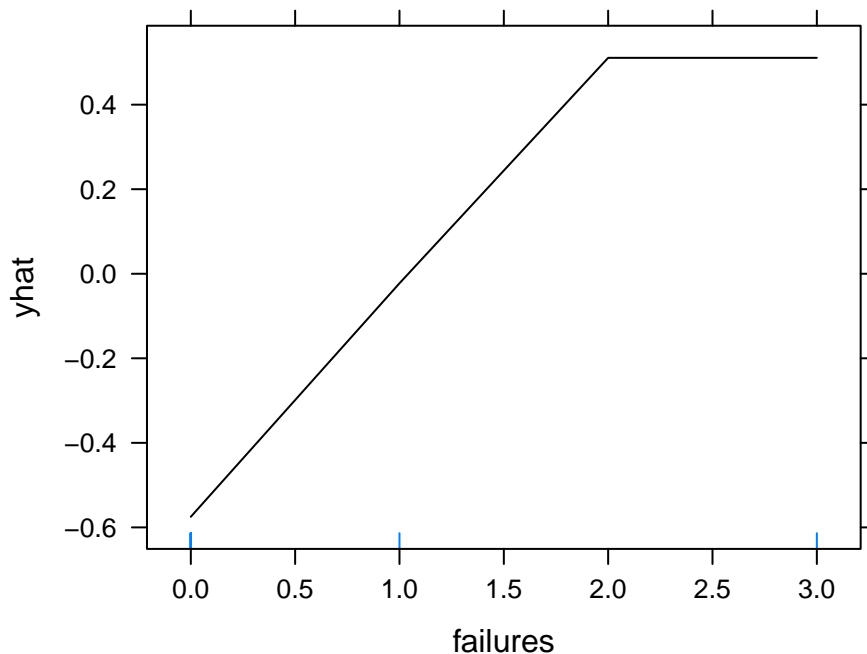
**ANSWER:**

**eta:**

**nrounds:**

---

The partial dependence plot for **failures** is provided below.



- (b) (2 points) Provide an interpretation of the plot above.

**ANSWER:**

---

- (c) (2 points) Describe the limitation of a partial dependence plot with respect to the interaction between variables.

**ANSWER:**

**Task 9 (7 points)**

To help School Wiz put the prediction performance of the models you have constructed or will construct in perspective, your assistant suggests fitting an intercept-only GLM for **pass** on the training set.

- (a) (2 points) Explain how the intercept-only GLM can be used to assess the prediction performance of other models.

**ANSWER:**

---

- (b) (3 points) Describe the characteristics of the prediction produced by this model and its ROC curve.

**ANSWER:**

---

- (c) (2 points) Determine the test AUC of this model.

**ANSWER:**

**\*\*END OF PRACTICE EXAM 1\*\***

## Practice Exam 1 Suggested Solutions

### ▲ NOTE ▲

The following apply to both Practice Exams 1 and 2:

- Each practice exam has a fairly comprehensive coverage of the topics in the PA exam syllabus, ranging from the business problem, data exploration, data preparation, to modeling issues concerning GLMs and decision trees. Apart from conceptual and descriptive items, I made a deliberate attempt to include some subtasks that test your understanding of basic R code and hand calculation based on some R output. These subtasks may figure more prominently in the new exam format.
- The following “suggested” solutions are mainly for illustration purposes. Even though they are likely to be more detailed than what you can write in 3.5 hours, they are by no means perfect. Feel free to augment and refine my responses as you see fit. In many cases, there is a range of fully satisfactory approaches and there may be valid alternatives not discussed.
- Particularly important points in the solutions are underlined for easy identification. While these points (or phrases with similar meaning) can definitely enrich your responses, there is **no expectation that you cover all of them**. As the [Guide to SOA Exams](#) says,  
*“...candidates do not always need to cover every possible aspect of the solution to receive full points...”*
- Some commentary and exam-taking strategies for specific subtasks are shown in *italics*. They are not part of the solutions.

### Task 1 – Justify using G3 to determine pass or fail (7 points)

**Ambrose’s comments:** This is an unseen, but not-so-demanding task specific to this business problem. It is about why using only one grade variable to determine pass or fail makes sense (though it may not be the optimal decision) with reference to the strong correlations among the three grade variables. A closely related topic is PCA.

#### Relevant PA exam learning outcomes:

- 2b) Identify the types of variables and terminology used in predictive modeling.
- 2f) Apply bivariate data exploration techniques.
- 3b) Apply principal components analysis to transform data.



- (a) (2 points) Describe one strength and one weakness of a correlation matrix as a bivariate data exploration tool.

**ANSWER:**

**Strength.** A correlation matrix provides a convenient way to summarize the strength of the linear relationship between numeric variables, one pair at a time, by a set of metrics (the correlations), ranging from  $-1$  to  $+1$ . Entries of the matrix that are close to  $+1$  or  $-1$  indicate strongly linearly related variables.

**Weakness.** Any of the following:

- A correlation matrix only captures linear relationships. Two numeric variables that have a nearly zero correlation can be related in many other regular ways (e.g., quadratic).
- A correlation matrix only captures the (linear) relationship between two numeric variables at a time. It may fail to represent relationships that exist among a group of numeric variables.
- A correlation matrix only works for numeric variables, not categorical (factor) variables.

- (b) (2 points) Based on the correlation matrix, explain why basing pass or fail entirely on **G3**, as requested by School Wiz, may be a sensible decision.

**ANSWER:**

The correlation matrix indicates that **G1**, **G2**, and **G3** are strongly positively correlated with each other, with all pairwise correlations greater than 0.8, meaning that they tend to move in the same direction. This is perhaps not surprising given that they capture very similar information (grades in consecutive trimesters) about students, so simply using one of the three grade variables, say **G3**, to build a pass/fail classifier will not result in too much loss of information compared to using all of the three grade variables. The two modeling approaches (using **G3** only versus using all of **G1**, **G2**, and **G3**) will likely produce similar classifications for students.

- (c) (3 points) Describe how principal components analysis can provide an alternative method for determining whether a student will pass or not based on all of **G1**, **G2**, and **G3**.

**ANSWER:**

Principal components analysis (PCA) is an analytic technique for summarizing high-dimensional data. It relies on the use of composite variables known as the principal components