ACTEX Learning

Study Manual for Advanced Topics in Predictive Analytics Assessment

2nd Edition

Ambrose Lo, PhD, FSA, CERA

An SOA Exam

ACTEX Learning

Study Manual for Advanced Topics in Predictive Analytics Assessment

2nd Edition

Ambrose Lo, PhD, FSA, CERA

Actuarial & Financial Risk Resource Materials
Since 1972

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.

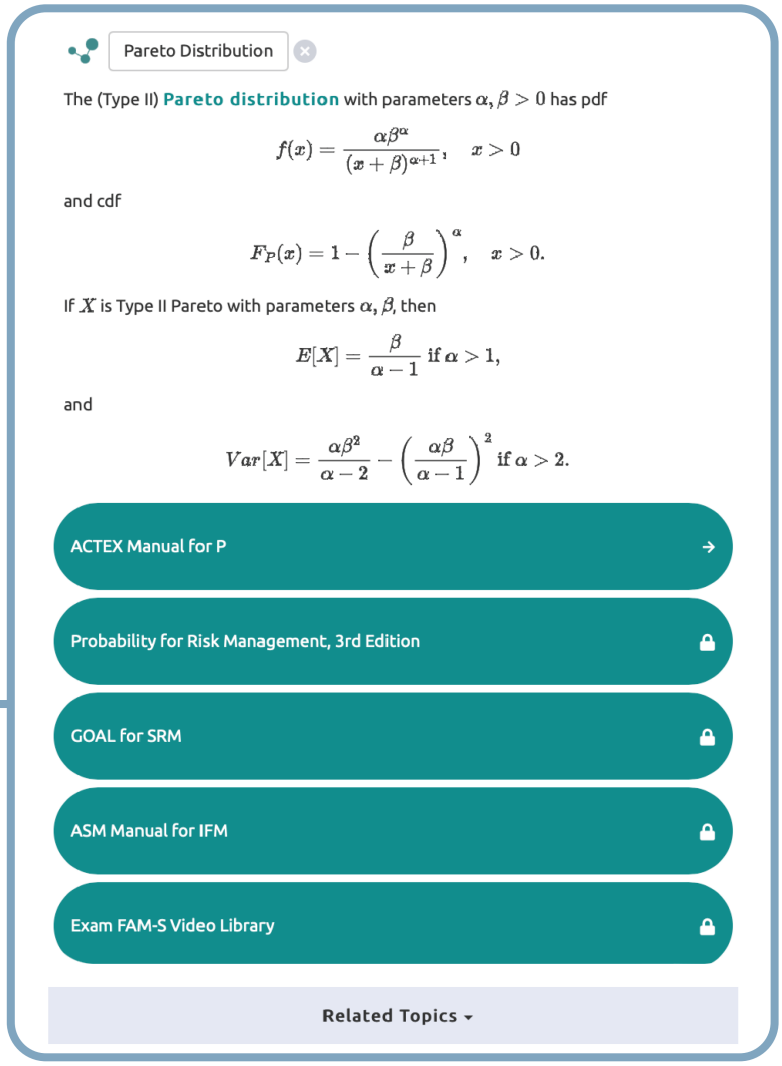
When this icon appears, it will be next to an important topic in the manual. Click the **link** in your digital manual, or search the underlined topic in your print manual.

1. Login to: www.actuarialuniversity.com

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.

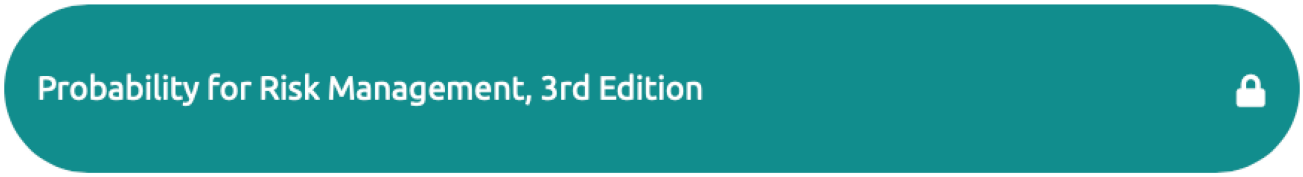
4. Here is an example of the topic **Pareto Distribution:**



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

Unlocked Products are the products that you own.

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

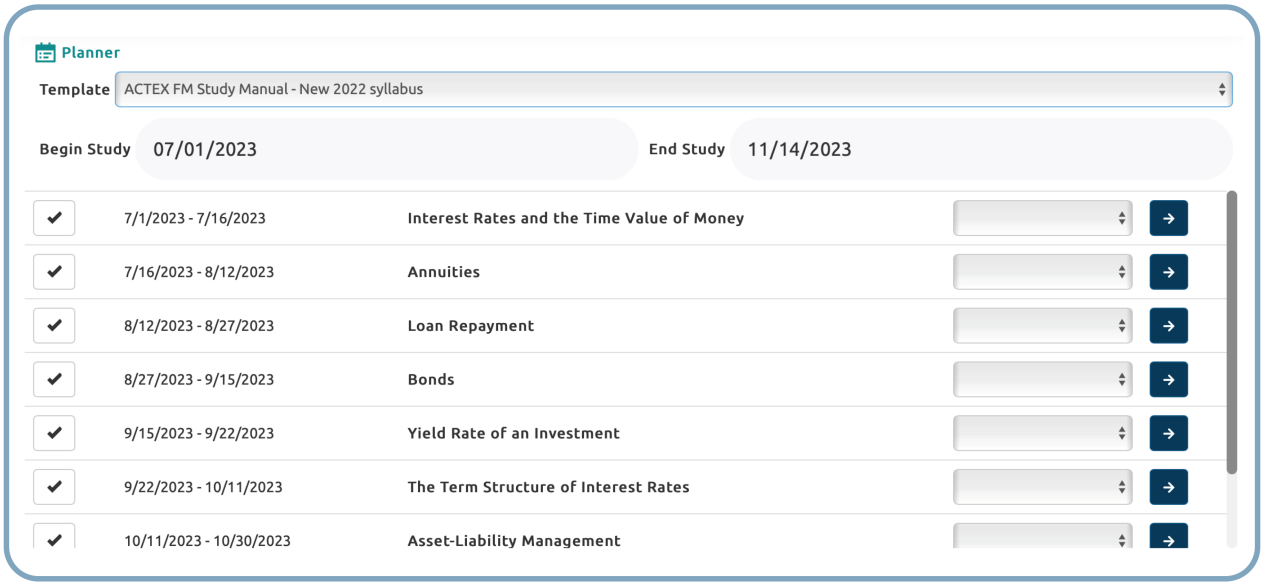


Many of Actuarial University's features are already unlocked with your study program, including:

ATPA Video Course* Planner

Topic Search and Interactivity with the ATPA Video Course

Make your study session more efficient with our Planner!



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

Contents

Preface

⚠ NOTE TO STUDENTS ⚠

Please read this preface carefully 📖, even if it looks long. It contains **VERY IMPORTANT** information that will help you make the most of this study manual and ease your learning.

Thank you very much for choosing to use this study manual, which is designed to provide comprehensive coverage of Exam ATPA (*Advanced Topics in Predictive Analytics*) and prepare you (more than!) adequately for this challenging exam.

P.1 About Exam ATPA

Exam Administrations

Exam ATPA is a 96-hour take-home computer-based 💻 assessment that has been part of the SOA's Associateship curriculum since January 2022. Because it is not a proctored exam, it should, more precisely, be called the ATPA "Assessment," although the SOA sometimes refers to it as "Exam" ATPA too. As will be discussed below, the preparation this assessment calls for is also broadly comparable to that of a closed-book exam that requires several hundred hours of study. There are three assessment windows per year, each lasting for three months, according to the schedule below:

https://www.soa.org/education/exam-req/exam-day-info/atpa-submission-schedule/ 🔗

For example, for the February-April 2026 window:

- Registration is open from December 1, 2025 to March 31, 2026. (In other words, you cannot register in April 2026 and complete the assessment in the same month.)
- The Assessment is available for download 📥 from February 2, 2026 to April 30, 2026. Once downloaded, the assessment must be completed within 96 hours (or 4 days). To have the full 96 hours to work on the assessment, you should start no later than April 26, 2026, 11:59 p.m. CT.
- **(IMPORTANT ⚠)** Your registration is tied to a specific assessment window. If you register between December 1, 2025 and March 31, 2026, then you will be eligible to take the ATPA Assessment in February-April 2026 window (and only this window). If you do not start or finish the assessment on or before April 30, 2026, then you will have to register for a future window (and pay the exorbitant $1,230 fee! **$$$**).

- Shortly after your registration, you will receive access to the ATPA e-learning modules until the end of the assessment window (April 30, 2026 in this case). According to the exam homepage:

  > "The ATPA e-Learning modules provide support designed to enhance candidates' knowledge from the Statistics for Risk Modeling (SRM) Exam and Predictive Analytics (PA) Exam learning objectives and readings. The modules will also clarify the SOA's expectations regarding a successful ATPA Assessment submission."

  In theory, it is possible to pass ATPA solely using the SOA's modules (that's how I passed it before this study manual came into existence!), which cover enough ground—well, they define the scope of this exam. In practice, this is by no means the best strategy to prepare for the assessment. If you have the stamina to go through the modules after completing this manual, you will find that:

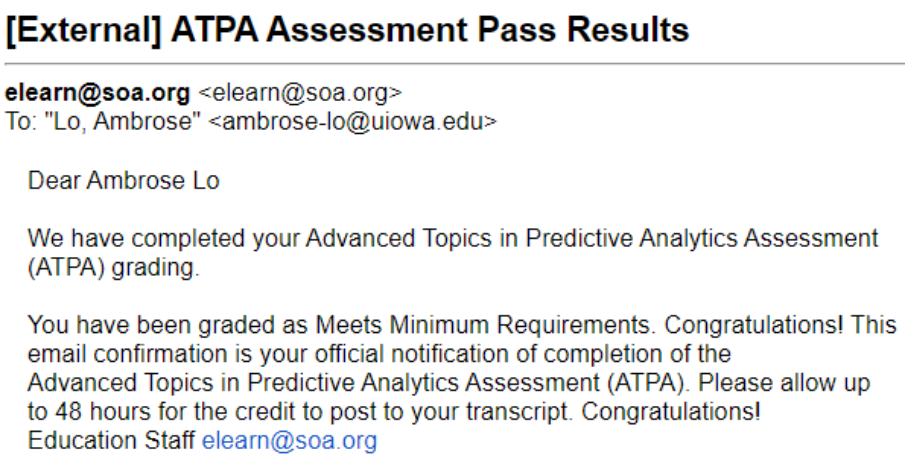
  ▷ It is a hassle to alternate between the SOA's PowerPoint slides and the accompanying R Markdown (Rmd) files. (The fact that SOA heavily uses PowerPoint slides is incongruent with communication in data science, where the norm is to use Rmd.)

  ▷ The modules are rather wordy and poorly organized. Not only are the explanations unnecessarily convoluted at times, the logical connections between different topics are not always clear.

  ▷ The modules present a number of case studies mostly of a silo nature. More often than not, you will fit some predictive models and look at the output, but the results have hardly any connections to the wider business problem or interesting insights.

  This manual is designed to avoid these problems and strives to help you prepare for the ATPA Assessment effectively, efficiently, and relatively enjoyably (more details in Section P.2).

Your assessment will be graded on a pass/fail basis (not on a scale from 0 to 10, which is for a proctored exam) and results will be emailed ✉ to you about two months following the end of an assessment window, e.g., late June 2026 for the February-April 2026 window. (You can expect "late June 2026" to be the last or second-to-last business day in June 2026.) In my experience, the email should be from `elearn@soa.org` with the subject "ATPA Assessment Pass Results."



[External] ATPA Assessment Pass Results

elearn@soa.org <elearn@soa.org>
To: "Lo, Ambrose" <ambrose-lo@uiowa.edu>

Dear Ambrose Lo

We have completed your Advanced Topics in Predictive Analytics Assessment (ATPA) grading.

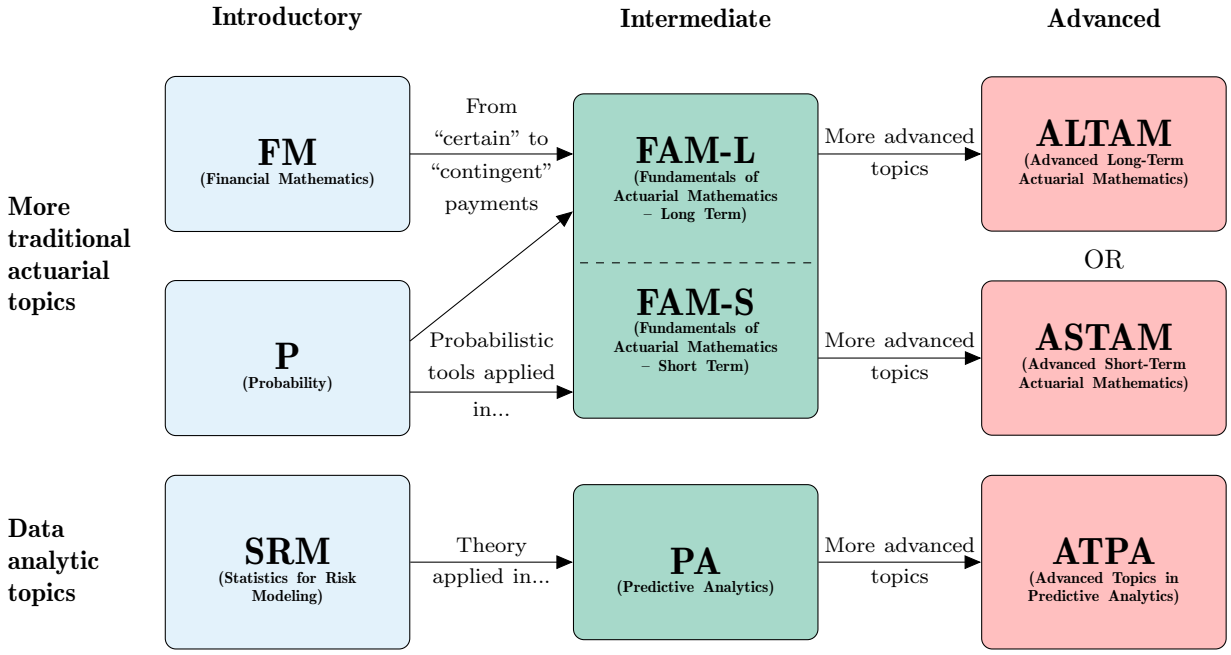
You have been graded as Meets Minimum Requirements. Congratulations! This email confirmation is your official notification of completion of the Advanced Topics in Predictive Analytics Assessment (ATPA). Please allow up to 48 hours for the credit to post to your transcript. Congratulations!
Education Staff elearn@soa.org

(The word "Pass" in the email subject may be replaced by another word if a student doesn't pass. 🤞)

What is the ATPA Assessment Like and How to Prepare for It?

In the current ASA curriculum, there are a total of 3 exams (or assessments) with a heavy focus on predictive analytics: SRM, PA, and ATPA, as the flowchart below shows. (Together, they form the recently introduced *Data Science for Actuaries Micro-credential*; see https://www.soa.org/programs/soa-ready/micro-credentials/, if anyone really cares about it...)

Flowchart of ASA Exams Effective from 2022



As the last component of the data analytics strand of the ASA curriculum, ATPA builds on the foundation planted in Exams SRM-PA and introduces "advanced" data and modeling concepts in two directions: (Remember that the first letter A in "ATPA" stands for "Advanced.")

- **Advanced predictive analytics models (ATPA Modules 3 and 4)**

  You will learn even more advanced predictive models than those covered in Exams SRM-PA. While these models share the same goal of improving prediction performance in different situations and issues from Exams SRM-PA like hyperparameter tuning and the bias-variance trade-off still apply, each of them has some subtleties (conceptual and programmatic) that you will appreciate when you get to specific sections of the ATPA syllabus.

- **Advanced data issues and management (ATPA Modules 1 and 2)**

  Effective from the April 2023 sitting of Exam PA, R and RStudio were no longer required or available on the exam, and many students seem to pay hardly any attention to R programming when they prepare for PA. For ATPA, however, proficiency with R is critical to

success. The datasets (notice the letter "s"...you are often provided with multiple datasets for your ATPA Assessment!) are almost always complicated by various data issues that you need to resolve before you can construct your advanced predictive models and carry out your interesting analysis. That's how the SOA tests your knowledge!

The ATPA syllabus has a note on the programming language you can use:

> "For your assessment you are free to use any programming language or statistical software."

On page 10 of Module 3, however, the SOA says that:

> "For the assessment, you can still use your language of choice, but we recommend that R be used for these models as the R code provided in this module will enable you to implement these models."

Accordingly, this study manual will follow the ATPA modules and solely adopt R as the programming language.

(**Note:** Some students may prefer to use Excel, which actuaries are often more familiar with, to clean and prepare data, which is perfectly fine.)

On the exam homepage, you can find a Sample ATPA Assessment with solution, which represents the scope of a typical assessment and the types of tasks that may be tested. The Sample Assessment suggests that a typical ATPA Assessment should have the following characteristics:

- There are 6-8 tasks, some with multiple items, with a total of 40 points.

  (⚠ **Note:** Mark allocations may not be given in the real assessment.)

- Unlike Exam PA, each task in ATPA builds upon the work and conclusions from prior tasks. As a result, the tasks should be done in order with results from one task informing work in later tasks, very much like a predictive modeling project in practice.

- Most of the tasks fall under the following themes:

  ▷ Manipulating and exploring some large datasets (with MANY variables) to prepare for subsequent analysis, e.g., Task 1 in the Sample Assessment

  ▷ Conceptual issues that test whether you have digested the material in the ATPA modules, e.g., Task 2

  ▷ (Majority) Constructing and tuning predictive models, e.g., Tasks 3-6

  ▷ Communicating your findings in writing, e.g., Task 7

  Like in Exam PA, it is highly unlikely that you need to write mathematical formulas or do theoretical derivations.

- There is an Rmd file that provides only little code in support of some initial data work.

  (⚠ **Note:** The real assessment may not provide such an Rmd file.)

- You will write the responses to each task in the provided Word file, which is the only file you will submit for grading. (No Rmd files can be or need to be submitted.)

Here are two tips that will help you succeed in your ATPA Assessment:

- **Prepare in advance**

  Although ATPA is a take-home assessment (which means that you are at liberty to refer to the ATPA modules and consult external resources such as textbooks and the Internet anytime)[1] 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. The SOA is no philanthropy—they give you 4 days with the expectation that you need at least 2 to 3 full days to finish the whole assessment. 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 (or Python) before building any predictive models. Then you will spend another 2 to 3 days performing your analysis and turning it into words. You will be busy doing tons of coding and writing!

  To make the 4 days slightly easier to get by:

  ▷ Study the advanced predictive models in the syllabus carefully, paying attention to their conceptual aspects (e.g, their mechanics, intended use, pros and cons) as well as practical implementations in R.

  ▷ Familiarize yourself with the R code </> in this study manual, which in turn closely follows that in the ATPA modules, to the extent that you know what each chunk of code does. If you are asked to fit a certain model or manage data in a certain way, then simply copy and modify the relevant R code. To save time and reduce errors, avoid writing R code from scratch.

- **Show your thought process and work clearly**

  As mentioned earlier, the only deliverable for your ATPA Assessment will be the Word document containing your written responses. Because the grader will not have access to your R code or see your R output, you should grasp the chance to document your thought process. Provide concrete evidence of what you have done (e.g., what models you have fitted), explain the rationale (e.g, refitting some models because you detect overfitting or diagnostic issues), and always justify the choices you make (e.g., what performance metric you are using). The grader can only grade based on what you have included in your Word document, so don't be afraid to state the obvious—obvious things may help you score! If that helps with your written explanations, try to copy and paste the R output (e.g., summary output of models, informative graphs) from RStudio into the Word document; see Section A.3 for more details.

[1] However, you may not discuss the assessment with other individuals.

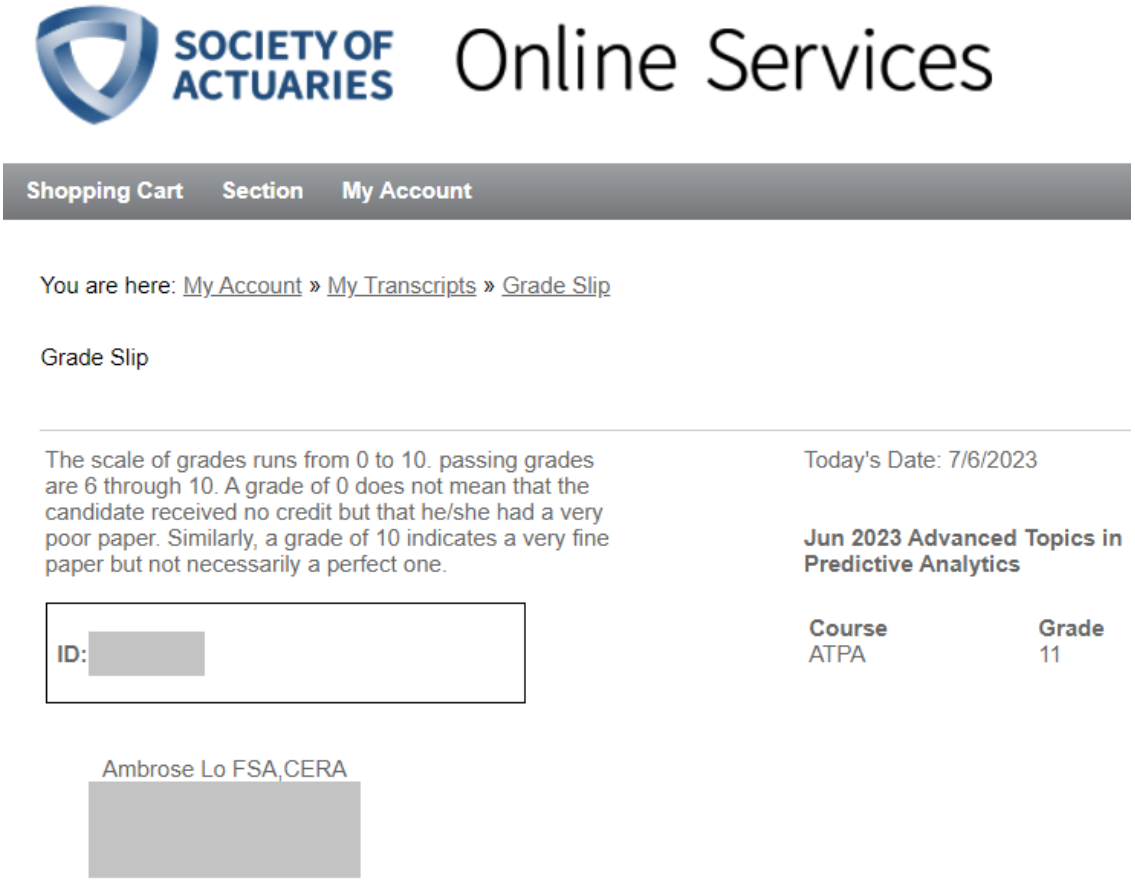
P.2 About this Study Manual

What is Special about This Study Manual?

Having been an actuarial student before and now an actuarial teacher, I understand too well that you have an acutely limited amount of study time outside of work/schoolwork, and that ATPA, as a relatively new component of the ASA curriculum, may seem intimidating. With this in mind, the overriding objective[2] of this study manual is to help you develop a conceptual understanding of and hands-on experience with the ATPA materials as effectively and efficiently as possible, so that you will pass the assessment on your first try easily and get your ASA ASAP (it's worth saying "ASA" twice). 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 **February-April 2023 ATPA Assessment** (besides Exams SRM-PA) to experience first-hand what the real assessment was like, despite having been an FSA since 2013 (and technically free from SOA exams thereafter!). I made this decision in the belief that teaching for an exam and taking an exam are rather different activities, and braving the ATPA Assessment myself is the best way to ensure that this manual is indeed useful for exam preparation. If the manual is effective, then at the minimum the author himself can pass, right? Fortunately...


SOCIETY OF ACTUARIES 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: 7/6/2023

Jun 2023 Advanced Topics in Predictive Analytics

ID:

Course	Grade
ATPA	11

Ambrose Lo FSA,CERA

[2]The secondary but still important objective is to let you have some fun along the way. 😜

⚠ **Note:** According to the SOA's Guide to Exams:

> "For some exams a grade of 11 will be recorded. In all cases that grade indicates the exam was passed, but does not indicate how well the candidate performed. There are three situations that can lead to this: ... The exam is graded on a pass-fail basis, ..."

So grade 11 simply means a pass. It doesn't mean my raw score in the assessment is so high that it breaks the scale! 💣

If you use this ATPA 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 ATPA 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 assessment in the best possible way. You are in good hands. 👍

Feature 2: Exam-focused Content

The advanced predictive models and data issues covered in ATPA can be very technically challenging. It is easy to get bogged down in unnecessary technicalities that may be useful for learning data science in general, but add little value to the ATPA Assessment. In this regard, this study manual is specifically geared towards helping you pass the assessment. It follows the ATPA modules very closely in terms of coverage and R-based implementations, but streamlines and augments the module materials in a coherent and exam-oriented format. With a nice blend of theory and practice, the manual:

- Presents the mechanics of all advanced predictive models and data issues in the syllabus at a suitable level.
- Illustrates these models and issues by a set of task-based case studies using R that provide you with some valuable hands-on experience.

You will get to manipulate some complex data, learn how these models work, and implement them step by step in R, all of which are crucial to success in the ATPA Assessment. I will also share with you my insights into what it takes to frame your written responses to the liking of ATPA exam graders.

Other Features

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

- Each chapter or section starts by explicitly stating which learning objectives and outcomes of the ATPA 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.

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

Formulas, R functions, and R commands that are of great importance are boxed to make them stand out.

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

> **⚠ EXAM NOTE ⚠**
>
> Be sure to pay special attention to boxes like this!

What is New in the Second Edition of the Manual?

About 100 pages longer than the first edition, the second edition of this manual has seen substantial updates in terms of clarity, substance, and exam focus throughout, but here are the most significant improvements:

- The following are entirely new:
  - ▷ Section 3.1 (Data Pipeline)
  - ▷ Subsection 3.5.3 (SOA's Case Studies)
  - ▷ Section 4.3 (Algorithmic Fairness)
  - ▷ Appendix A: Where to Go Next?
  - ▷ Appendix R: A Crash Course in R for ATPA
    (This new appendix is added in response to various student requests.)
- The mini-case study in Section 3.4 (Combining Data from Multiple Sources) has been substantially updated and expanded.
- All typos and errors brought to my attention have been fixed.

Supplementary Files

This study manual comes with a number of supplementary files, e.g., Rmd files with completely reproducible R code, datasets, that can all 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` for assistance.

Contact Us

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

- For **technical issues** (e.g., not able to access or download supplementary files from *Actuarial University*, extending your digital license), please email ACTEX Learning's Customer Service at

  `support@actexlearning.com`.

  The list of FAQs on https://www.actuarialuniversity.com/help/faq may also be useful.

- For questions related to **specific contents** of this manual and Exam ATPA, including potential errors (typographical or otherwise), please feel free to raise them in the ATPA forum on ACTEX's Discord channel, which provides a convenient platform for you to network with other ATPA students (and me!), and I will strive to respond to your questions ASAP.

  As far as possible, please keep your questions short and specific. Instead of saying

  "You mention in your manual that...,"

  please quote the specific page(s) of the manual your questions are about, or, even better, take a screenshot of the relevant pages. This will provide a concrete context and make our discussion much more fruitful.

Acknowledgments

I am grateful to students who used the first edition of this manual and took 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 in 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 study manuals for various actuarial exams, including INV 201, ATPA, FAM, MAS-I, MAS-II, PA, SRM, 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

Advanced Predictive Analytics Models and Related Issues

Chapter 2

Model Explainability and Communication

FROM THE ATPA EXAM SYLLABUS

4. Topic: Model Explainability and Communication (25-35%)

Learning Objectives

The Candidate will be able to effectively communicate the results of applying predictive analytics, including the relationship between model input and output, to solve a business problem.

Learning Outcomes

The Candidate will be able to:

a) Understand aspects of explainability, in particular:
   - The connection between ethics and explainability
   - Suitability, decomposability, algorithmic transparency, and post-hoc interpretability
   - The difference between explainability and interpretability
   - When a lack of explainability may be acceptable.

b) Communicate and justify a recommended analytics solution, including use as appropriate of:
   - Variable importance plots
   - Partial dependence plots
   - Individual conditional expectation plots
   - Shapley values
   - Lift and gain charts.

(Continued on the next page...)

*****FROM THE ATPA EXAM SYLLABUS*****
(Continued)

c) Explain why a model is predicting certain values for certain records.

d) Perform data and model governance and develop model documentation in an ethical context.

e) Communicate in a clear and straightforward manner using common language that is appropriate for the intended audience.

f) Structure a report in an effective manner while following standards of practice for actuarial communication.

Chapter overview: The previous chapter covered the ins and outs of several advanced predictive analytics models which are the centerpiece of the ATPA syllabus. Having decided to use some of these models and spent hours building and tuning them to your liking, you will want to communicate the results of your predictive modeling work to other audiences such as your peers, supervisors, and clients. Based on ATPA Module 4, this chapter aims to establish good forms of communication, especially written communication, which is the main (if not the only) form of communication tested in your ATPA Assessment.

2.1 Techniques for Interpreting Opaque Models

⚠ EXAM NOTE ⚠

The advanced models in Chapter 1 are the focus of ATPA, and some (or many) of them must be tested in the ATPA Assessment, but the techniques covered in this section may or may not. If they are indeed tested, then they are likely to show up in a relatively short task after the modeling stage of the assessment, e.g., Task 6 of the Sample Assessment, which only carries 4 points (out of 40).

Explanation vs. interpretation. ATPA Module 4 begins by making a distinction between two ways of making sense of a predictive model:

- *Explanation*

  According to the SOA, an *explanation* refers to a technical breakdown of the steps a predictive model goes through to turn inputs (predictors) into outputs (final predictions). At a broader level, an explanation is about "explaining" the reasoning behind a model's decision-making process.

  The models we have learned in Exams PA-ATPA differ widely in terms of explainability—the degree to which they can be explained. The more complex a model, the less explainable it tends to become.

Example 1. GLMs, which provide an analytic equation relating the target mean to the predictors, are inherently explainable. A single glance at the model equation

$$g(\mu) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

shows that the predictors contribute to the output via a monotonic function (the link) of a linear combination. The same goes for GAMs and (to a smaller extent) GLMMs we studied in Sections 1.1 and 1.2.

Example 2. At the other extreme, ensemble trees and neural networks are notoriously opaque and difficult to explain because of the non-linear and complex relationship between the input and output variables. Ensemble trees are of low explainability due to the presence of multiple base trees obfuscating the input-output relationship. In the same vein, recall from Section 1.3 that a neural network involves repeated non-linear transformations of the input variables, so many that it is virtually impossible to trace how the input variables make up the neurons in the output layer.

In general, a more explainable model produces decisions that are easier for humans to comprehend 🧠 and is more likely to earn trust from its users.

- *Interpretation*

  While literally similar to an explanation in meaning, the SOA defines an *interpretation* as a statement that discusses the implications of the model output in the context of a given business problem. In Exams PA-ATPA, here are the most common forms of interpretation:

  ▷ Which variables are significant predictors of the target variable?

  ▷ For the significant predictors, what are their relationships with the target variable, e.g., positive, negative, non-monotonic? What are the implications of these relationships for the business problem?

  In general, a more interpretable model has a tendency to produce insights that are valuable for solving the business problem at hand.

⚠ EXAM NOTE ⚠

Despite stressing the difference between explanations and interpretations at the outset, ATPA Module 4 later uses the two terms almost interchangeably and focuses on techniques to interpret (in the sense above) a model. My advice when you take your ATPA Assessment is:

- If you are asked to "explain" a model, describe the mechanics of the model as well as the relationships between the key predictors and the target with a technical flavor. There is no need to discuss the implications of these relationships for the wider business problem.
- If you are asked to "interpret" a model, describe the relationships between the key predictors and the target, and try your best to relate these findings to the business problem.

Global vs. local interpretability. A model that is inherently explainable is generally more interpretable. After all, even a layman has an easy time unraveling the inner workings of the model and seeing how each predictor makes its way to the final output.

What about models that are not as explainable such as **neural networks**? Fortunately, there are techniques to make approximately correct interpretations that shed light on the relationships between the target variable and predictors, without the users having to delve into the mechanics of the models. Due to the intrinsic complexity of opaque models, it would be a thankless task to produce completely accurate explanations, so these techniques inevitably simplify the model structure somewhat in an attempt to produce comprehensible but hopefully insightful statements, and we should be aware of the limitations of these techniques.

In ATPA, we will learn and apply a few interpretational techniques (or methods). They can be categorized as follows.

- *Global vs. local*

  Global methods take a holistic view on how a predictive model produces predictions for all observations in the data. In other words, they investigate the general (hence "global") behavior of the model.

  Local methods, in contrast, study how a predictive model makes predictions only for some observations of interest. In other words, they investigate the "local" behavior of the model on specific observations.

  In theory, it is possible to aggregate the results of local methods on a sufficiently large number of observations to come up with an approximately global interpretation.

- *Model-specific vs. model-agnostic*

  As their name suggests, interpretational techniques that are *model-specific* are "specific" to certain types of predictive model and have limited applicability.

  The focus of ATPA is therefore on *model-agnostic* techniques, which can be used on basically any predictive models and are widely applicable. Applied after a model has been trained, these techniques do not rely on the inner workings of the model and are usually concerned with analyzing input-output relationships.

2.1.1 Global Method 1: Variable Importance

How does variable importance work? ***Variable importance*** is a simple interpretational tool that assigns a score to each variable in a predictive model measuring its "importance." The larger the importance score of a predictor, the more it explains the target variable, and the more "important" it appears. To facilitate visual comparison, we can use a variable importance plot (or table) to display and rank the predictors in descending order of variable importance, e.g.:

Variable	Importance Score	
1	100	(most important)
2	76	(second most important)
3	34	
⋮		

Looking at a variable importance plot, we can easily tell which predictors are the most important, e.g., Variable 1 in the hypothetical table above, followed by Variables 2 and 3, in this order.

Here is a mnemonic:

A variable importance plot indeed shows the VIP, or Very Important Predictors!

How does variable importance fit the classifications of interpretational techniques we introduced above? To begin with, variable importance is computed for each variable across all observations (rather than specific observations) in the data, so it is a global method. However, it is a model-specific method because the precise definition of the variable importance score varies with the type of model you use:

- For GLMs, the variable importance score of a predictor is defined as the absolute value of the t-value (or z-value for linear models) of the predictor. As we learned in Exam SRM, the larger the t-value in magnitude, the more significant the variable.
- For decision trees, including single and ensemble ones, the variable importance score of a predictor is the average drop in node impurity (which is **RSS** for regression trees and **Gini index** for classification trees) due to splits over that predictor over all the base trees:

$$\text{Variable importance score} = \frac{1}{B} \times \sum_{\substack{\text{all splits over} \\ \text{that predictor}}} \text{Impurity reduction}.$$

This is the definition of variable importance you saw in Exam PA. (Remember? 😊)

Strangely, the ATPA modules do not discuss how to define variable importance scores for more advanced predictive models like GAMs, GLMMs, or neural networks, although they are the focus of the ATPA Assessment.

Pros and cons of variable importance.

⊕ By design, this method reduces a predictive model, however complex, to a set of scores, one per variable. We can easily compare these scores and understand, at a high level, which variables have the greatest impact on the target variable.

⊖
- *(Limited applicability)* As a model-specific method, variable importance is only applicable to specific model types.
- *(Nothing about relationships)* Although variable importance tells us which predictors are most influential, the importance scores themselves do not shed light on the relationship between the predictors and the target variable. In other words, we know that a variable with a large importance score contributes significantly to the target variable, but whether that contribution is positive, negative, or follows a more complex relationship remains unknown.

  (This deficiency is filled by the tool in Subsection 2.1.2.)

- *(Susceptibility to strongly dependent predictors)* When there are two highly related predictors, the importance score of one predictor can be adversely skewed by the presence of the other predictor.

  As an extreme example, consider a decision tree and two numeric predictors, X_1 and $X_2 = X_1 + 0.000001$, with X_2 being essentially a duplicate of X_1. Even if X_1 is a strong predictor, the tree may mistakenly use X_2 as the split variable, which leads to a dilution of the importance of X_1 relative to other predictors in the data.

R demonstration. Let's end this subsection by looking at some real variable importance table and plot based on the `Bikeshare` data we first studied in Subsections 1.1.2 and 1.3.2. In CHUNK 1, we load and prepare the `Bikeshare` data, following the same adjustments we performed earlier.

```
# CHUNK 1
rm(list = ls()) # Start with a clean environment
library(ISLR2)
library(caret)
data(Bikeshare)

# Repeat data adjustments from Subsection 1.1.2
Bikeshare <- Bikeshare[, !names(Bikeshare) %in%
                         c("season", "day", "weekday",
                             "atemp", "casual", "registered")]
Bikeshare$hr <- as.numeric(Bikeshare$hr)
levels(Bikeshare$weathersit)[3:4] <- "rain/snow"

# Repeat the same training/test set split in Subsections 1.1.2 and 1.3.2
set.seed(0)
train_ind <- createDataPartition(Bikeshare$bikers, p = 0.7, list = FALSE)
dat_train <- Bikeshare[train_ind, ]
dat_test <- Bikeshare[-train_ind, ]
```

Then in CHUNK 2, we fit a random forest to `bikers` using all other variables as predictors. Given the focus of this chapter, we are not interested in tuning the random forest to optimal performance; we only need a decently and efficiently trained random forest for illustration purposes, so we simply set the number of variables randomly sampled at each split equal to 5.

```
# CHUNK 2
# install.packages("randomForest")
set.seed(1)

# set the number of variables randomly sampled at each split = 5
rfGrid <- expand.grid(mtry = 5)

# set the validation method as 5-fold cross-validation
# needed only when mtry has two or more possible values
```

```
ctrl <- trainControl(method = "cv", number = 5)

RF <- train(bikers ~ .,
            data = dat_train,
            method = "rf",
            trControl = ctrl,
            tuneGrid = rfGrid,
            ntree = 50, # reduce no. of base trees from 500 (default)
                        # to 50 to save run time
            importance = TRUE)
```

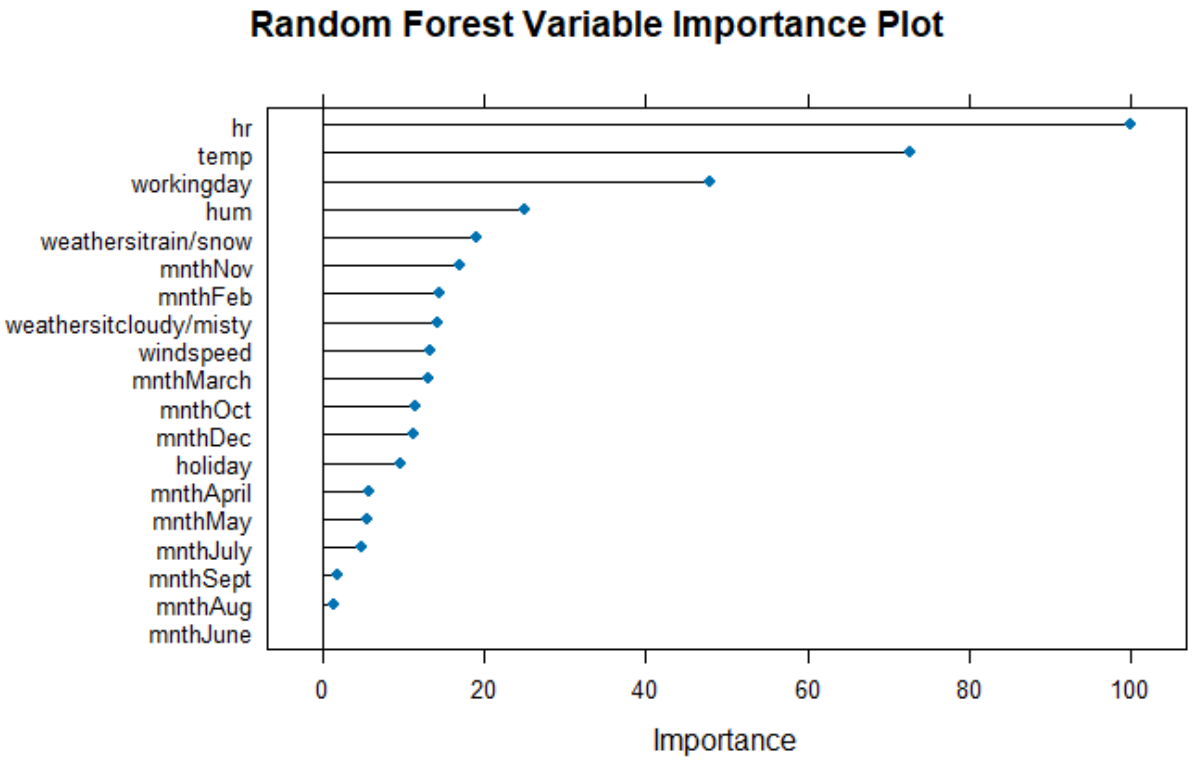
Having fitted the random forest, in CHUNK 3 we use the aptly named `varImp()` function in the `caret` package to make a variable importance table and apply the `plot()` function to make a variable importance plot.

```
# CHUNK 3
varImp(RF)

rf variable importance

                        Overall
hr                      100.000
temp                     72.796
workingday               47.874
hum                      25.092
weathersitrain/snow      19.066
mnthNov                  17.064
mnthFeb                  14.487
weathersitcloudy/misty   14.095
windspeed                13.369
mnthMarch                12.943
mnthOct                  11.394
mnthDec                  11.282
holiday                   9.538
mnthApril                 5.789
mnthMay                   5.590
mnthJuly                  4.776
mnthSept                  1.909
mnthAug                   1.380
mnthJune                  0.000

plot(varImp(RF, type = 1), main = "Random Forest Variable Importance Plot")
# Type = 1 uses the method described in the ATPA modules
```



In the variable importance tables and plots, the variables have been sorted in descending order of importance and the importance scores have been scaled so that the most important predictor has a score of 100. We can see that `hr` is by far the most important variable, followed in order by `temp`, `workingday`, and `hum`. These findings align well with the exploratory data analysis we performed in Task 1 of Subsection 1.1.2, but with the variable importance scores, we are able to assert quantitatively how important the predictors are.

2.1.2 Global Method 2: Partial Dependence

We have just identified `hr` and `temp` as the most important predictors of `bikers`, but how does `bikers` vary with these two variables? The variable importance scores say nothing about relationships, but this is where partial dependence can fill the gap. (The treatment of partial dependence here is similar to that in Exam PA, so this subsection is mostly a review!)

How does partial dependence work? ***Partial dependence plots***, or PDPs, attempt to visualize the *average marginal effect* of a given predictor of interest on the target variable, i.e.,

> the association between the target and predictor after averaging out the values or levels of other predictors not of interest.

Looking at these plots 👁, we can gain some insights into how the target variable "depends" on each predictor on a "partial" basis.[1] Because these plots concern relationships across all observations in the data, they are a global method.

[1] In statistics, the qualifier "partial" usually means that the quantity concerned is computed after accounting for the effects of other variables.

Intuitively, partial dependence uses averaging to tease out the relationships between the target and the predictors. To understand exactly how this works, let's consider a target variable Y and p predictors $X_1, \ldots, X_p$, and we are interested in how one of the predictors, say X_1, drives Y. Mathematically, the *partial dependence* of Y on X_1 based on a predictive model is defined as

$$\boxed{\text{PD}(x_1) := \frac{1}{n}\sum_{i=1}^{n} \hat{f}(\underbrace{x_1}_{\text{fixed}}, \underbrace{x_{i2}, \ldots, x_{ip}}_{\text{averaged}}),} \tag{2.1.1}$$

where:

- $\hat{f}$ is the fitted signal function, i.e., the prediction produced by the model under investigation.
- x_1 is a fixed value or level of X_1 (depending on whether X_1 is numeric or categorical).
- $\{(x_{i2}, \ldots, x_{ip})\}_{i=1}^{n}$ is the set of observed values of $X_2, \ldots, X_p$ in the training set, and n is the size of the training set.

By definition, $\text{PD}(x_1)$ simply equals the average of the model predictions over all the observed values of $X_2, \ldots, X_p$ (the variables not of interest) in the training set while keeping the value or level of X_1 (the variable of interest) fixed at x_1 for all training observations. The following diagram visualizes the whole procedure and emphasizes that $\text{PD}(x_1)$ is a function of x_1 (boxed):

X_1	X_2	$\cdots$	X_p		Model Prediction		Partial Dependence
$\boxed{x_1}$	x_{12}	$\cdots$	x_{1p}	(apply the model) $\to$	$\hat{f}(\boxed{x_1}, x_{12}, \ldots, x_{1p})$		
$\boxed{x_1}$	x_{22}	$\cdots$	x_{2p}	(apply the model) $\to$	$\hat{f}(\boxed{x_1}, x_{22}, \ldots, x_{2p})$		
$\vdots$	$\vdots$	$\cdots$	$\vdots$		$\vdots$	(average) $\to$	$\text{PD}(\boxed{x_1})$
$\vdots$	$\vdots$	$\ddots$	$\vdots$		$\vdots$		
$\boxed{x_1}$	x_{n2}	$\cdots$	x_{np}	(apply the model) $\to$	$\hat{f}(\boxed{x_1}, x_{n2}, \ldots, x_{np})$		

Repeating the calculations at various other choices of x_1, we can then produce a PDP, which is a plot of $\text{PD}(x_1)$ (on the y-axis) against x_1 (on the x-axis), and examine its behavior with a view to understanding how X_1 affects the target variable. If, for example, the PDP shows that $\text{PD}(x_1)$ tends to increase with x_1, then we may deduce that X_1 has a positive marginal effect on the target.

To give you some idea what a PDP really looks like, the following exercise demonstrates the geometric form of $\text{PD}(x_1)$ for GLMs, which are a simple type of model we are familiar with. Even though you will do hardly any mathematical derivations in Exam ATPA, the exercise may give you some useful insights into the workings of PDPs.

Exercise 2.1.1. **(Motivated from pages 36 and 37 of ATPA Module 4: Partial dependence for a (G)LM)** Describe the form of the PDP for X_1 in each of the following cases: (Assume that X_1 is a numeric variable.)

(a) A linear regression model $\mu_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$ for $i = 1, \ldots, n$.

(b) A log-link GLM $\ln \mu_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip}$ for $i = 1, \ldots, n$.

Solution.

(a) For a linear regression model, the model prediction takes the linear form

$$\hat{f}(X_1, X_2, \ldots, X_p) = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 + \cdots + \hat{\beta}_p X_p.$$

Then by (2.1.1),

$$\begin{aligned} \text{PD}(x_1) &= \frac{1}{n}\sum_{i=1}^{n} \hat{f}(x_1, x_{i2}, \ldots, x_{ip}) \\ &= \frac{1}{n}\sum_{i=1}^{n} (\hat{\beta}_0 + \underbrace{\hat{\beta}_1 x_1}_{\text{(free of } i)} + \hat{\beta}_2 x_{i2} + \cdots + \hat{\beta}_p x_{ip}) \\ &= \hat{\beta}_1 x_1 + c, \end{aligned}$$

where $c := \frac{1}{n}\sum_{i=1}^{n}(\hat{\beta}_0 + \hat{\beta}_2 x_{i2} + \cdots + \hat{\beta}_p x_{ip}) = \hat{\beta}_0 + \hat{\beta}_2 \bar{x}_2 + \cdots + \hat{\beta}_p \bar{x}_p$ is a constant that does not depend on x_1. In other words, the PDP is a straight line in x_1, with an intercept of c and a slope of $\hat{\beta}_1$, which is the OLS estimate of the coefficient of X_1. In this simple case, the PDP is an exact representation of the marginal effect of X_1 on the target.

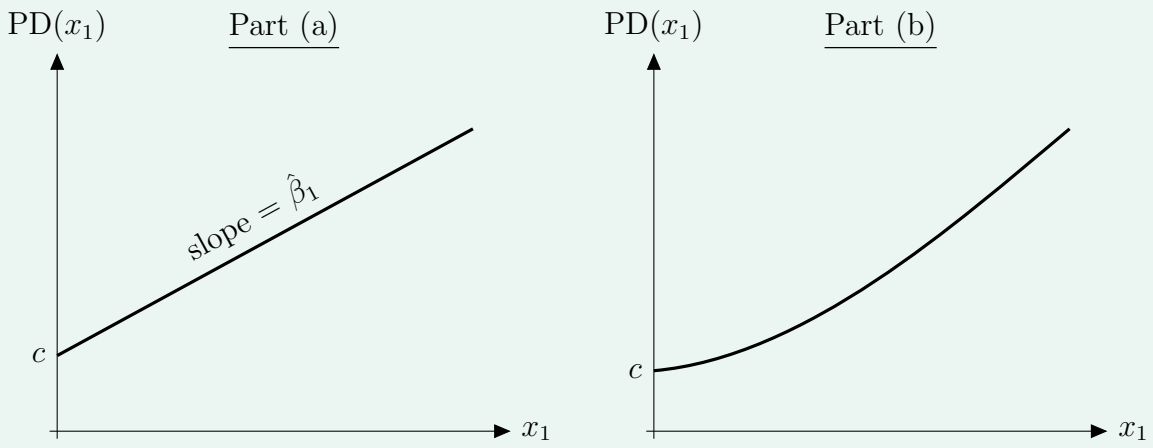
(b) For a log-link GLM,

$$\begin{aligned} \text{PD}(x_1) &= \frac{1}{n}\sum_{i=1}^{n} \hat{f}(x_1, x_{i2}, \ldots, x_{ip}) \\ &= \frac{1}{n}\sum_{i=1}^{n} \mathrm{e}^{\hat{\beta}_0 + \hat{\beta}_1 x_1 + \cdots + \hat{\beta}_p x_{ip}} \\ &= \left(\frac{1}{n}\sum_{i=1}^{n} \mathrm{e}^{\hat{\beta}_0 + \hat{\beta}_2 x_{i2} + \cdots + \hat{\beta}_p x_{ip}}\right) \mathrm{e}^{\hat{\beta}_1 x_1} \\ &= c\mathrm{e}^{\hat{\beta}_1 x_1} \end{aligned}$$

where $c := \frac{1}{n}\sum_{i=1}^{n} \mathrm{e}^{\hat{\beta}_0 + \hat{\beta}_2 x_{i2} + \cdots + \hat{\beta}_p x_{ip}}$ does not depend on x_1. In this case, the PDP is an exponential curve in x_1, as we would expect from the exponential model equation. The curve goes up or down, depending on the sign of $\hat{\beta}_1$. □

Remark.

(i) The graphs below visualize the PDP for each of the two cases: (Assume that $\hat{\beta}_1 > 0$)



(ii) If X_1 is categorical (rather than numeric), then the PDP will be a discrete set of points (rather than a straight line or curve) computed according to (2.1.1) and indexed by the possible levels of X_1.

(iii) For models other than GLMs, $\hat{f}$ is generally such a complex function that it is virtually impossible to determine their PDP in closed form, in which case we can only examine PD(x_1) numerically (i.e., using a computer to calculate PD(x_1) for various x_1 and make a PDP) rather than analytically.

Exercise 2.1.2. **(A common misconception about PDPs)** Critique the following statement made by your assistant (back from Exam PA!):

> "A partial dependence plot shows the model predictions at the average values of the variables that are not of interest in the training set, while keeping the variable of interest at a fixed value/level."

Solution. The statement looks reasonable on the surface, but it mixes up the order in which the averaging should take place and is conceptually incorrect. Instead of taking a single model prediction evaluated at the average values of the variables that are not of interest, a partial dependence score is actually an average of many model predictions, each evaluated at the observed values of the variables not of interest, while keeping the variable of interest at a fixed value/level. □

Remark. Task 9 (c) of the April 2023 PA exam is:

> "(*2 points*) Describe how values for a partial dependence plot are calculated for a specific variable [in a random forest model]."

As noted in the SOA's comments, the most common reason for students to lose points was to state that:

> "...the value of the predictor variable [presumably the predictors not of interest] is fixed at the average of that variable across all observations,"
>
> which is exactly the misconception tested in this exercise.
>
> Mathematically,
>
> $$\underbrace{\text{PD}(x_1) := \frac{1}{n_{\text{tr}}} \sum_{i=1}^{n_{\text{tr}}} \hat{f}(x_1, x_{i2}, \ldots, x_{ip})}_{\text{(correct definition)}} \neq \underbrace{\hat{f}\left(x_1, \frac{1}{n}\sum_{i=1}^{n} x_{i2}, \ldots, \frac{1}{n_{\text{tr}}}\sum_{i=1}^{n} x_{ip}\right)}_{\text{(misconception shown in the statement)}}.$$
>
> Simply put, the averaging should be outside the $\hat{f}$ function, not inside.

Pros and cons of partial dependence.

⊕

- *(It is model-agnostic)* The computation of partial dependence does not in any way depend on the structure or properties of $\hat{f}$. Given $\hat{f}$ and the training data, we can use (2.1.1) to construct the PDP without having to know what type of predictive model it is. In other words, PDPs are model-agnostic.
- *(Ease of interpretation)* As an interpretational technique, a PDP is an intuitive visualization which is easy to interpret. It clearly shows (or tries to show!) how the target variable, approximated by the average prediction, depends on each predictor.
- *(Ease of implementation)* A PDP is also computationally easy to produce. The function $\hat{f}$ is already available as part of the model training process, and all we have to do is apply it to the adjusted training observations (the adjustment being that all values of X_1 are set to x_1). No refitting of models is needed.

⊖ Unfortunately, PDPs suffer from some non-trivial drawbacks.

- *They assume that the variable of interest is independent of other variables.* The calculation of $\text{PD}(x_1)$ is based on the modified training set above, where all values/levels of X_1 are forced to be x_1, an arbitrary value/level of interest (rather than its original values/levels in the training set, $x_{11}, x_{21}, \ldots, x_{n1}$). Doing so destroys the relationships between X_1 and other predictors in the data, and implicitly assumes that they are independent of each other. This assumption is questionable in many cases in real life and can create combinations of predictor values/levels that are previously unseen and practically unreasonable.

  Example. Consider, for instance:

  ▷ X_1 = age, ranging from 5 to 65 in the training set

  ▷ X_2 = income, ranging from \$1,000 to \$1,000,000 in the training set

  It is generally true that X_1 and X_2 are positively correlated (income tends to increase with age). If we ignore this correlation and compute the partial dependence of the target variable on X_1 at, say $x_1 = 10$, then in the process we would be unwittingly including the model prediction for a 10-year-old millionaire, which is not an entirely impossible, but extremely unrealistic scenario. (A super rich kid!! 🧒 $$$$$)

- (This point is harder to grasp!) *They may miss heterogeneous relationships in the data.*

  Example. To see what this means, consider a variable for which:

  ▷ Half of the observations in the data have a positive relationship with the target variable (the larger the variable value, the larger the prediction).

  ▷ The other half has a negative relationship of similar strength.

  By design, the partial dependence on this variable would average the predictions over all of the observations and cancel out the different (or heterogeneous) effects of both halves of the data, making the PDP a roughly flat line. If we rely on the PDP, then we may mistakenly believe that this variable is unimportant in predicting the target variable.

  There are two ways to remedy this drawback:

  (1) As briefly discussed on page 39 of ATPA Module 4, one remedy is to make a PDP for two variables of interest, rather than a single variable at a time. Although such a PDP may reveal interactions, it takes a three-dimensional plot or a heat map to construct, both of which are much harder to decipher and take longer to produce.

  (2) Another remedy is a local version of PDPs called *individual conditional expectation plots*, which will be covered in Subsection 2.1.4.

In short, a PDP produces (or tries to produce) potentially useful insights by simplifying a predictive model, but it runs the risk of oversimplification and should not be trusted blindly.

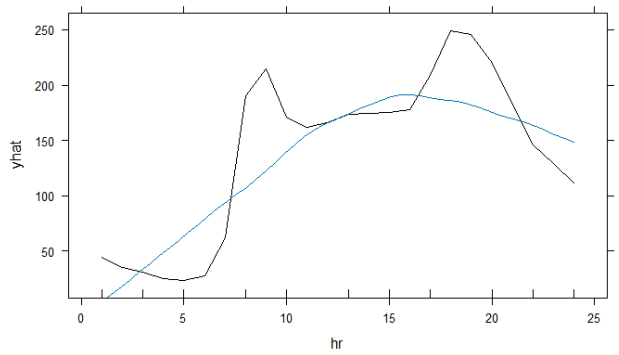
R demonstration. In R, we can generate PDPs using the `partial()` function in the `pdp` package and specify the variable of interest as a character string in the `pred.var` argument of the function. There are options for customizing the appearance of the plot.

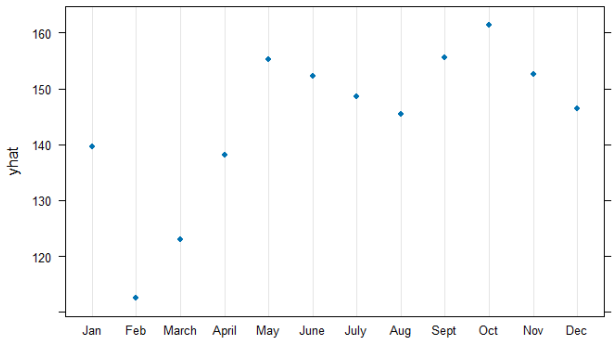
Let's run CHUNK 4 to make PDPs for `hr` (the most important numeric variable) and `mnth` (the most important categorical variable) using the random forest fitted above. Do take a look at what the different options of the `partial()` function do.

```
# CHUNK 4
library(pdp)

partial(
  RF,
  train = dat_train, # the original training data
  pred.var = "hr",
  plot = TRUE, # generates a plot of partial dependence values;
  # the default is FALSE, which generates a table of partial dependence values
  smooth = TRUE, # adds a blue smoothed curve
  rug = TRUE # produces eleven tick marks above the horizontal axis;
  # these are the min, max, and deciles of the variable
)
```

```
partial(
  RF,
  train = dat_train,
  pred.var = "mnth",
  plot = TRUE # smooth and rug are irrelevant to categorical variables
)
```





The first PDP reproduces the bimodal wave-like relationship between `bikers` and `hr` we saw in Task 1 of Subsection 1.1.2, with one peak at 8-9 a.m. and another peak at 6-7 p.m. The PDP for `mnth` has a somewhat similar shape (although the 12 levels of `mnth` are not treated as ordered) and shows that the number of bikers is the highest in May, June, September, and October, and becomes much lower in January, February, and March (too cold to bike in the winter!).

Exercise 2.1.3. **(Similar to Task 9 (e) of the April 2023 Exam PA: What is bad about the smoothed curve?)** Discuss the danger of using the blue smoothed curve in the PDP for `hr` above.

Solution. A danger of including a smoothed curve in a PDP is that it may "over-smooth" the partial dependence curve and obscure some subtle but useful patterns.

In the PDP for `hr` above, the blue smoothed curve hides the two modes of the partial dependence curve and produces a uni-modal curve that peaks at about 3 p.m. This can lead to a potentially huge loss of information.

In any case, one would be wise not to rely entirely on the smoothed curve. □

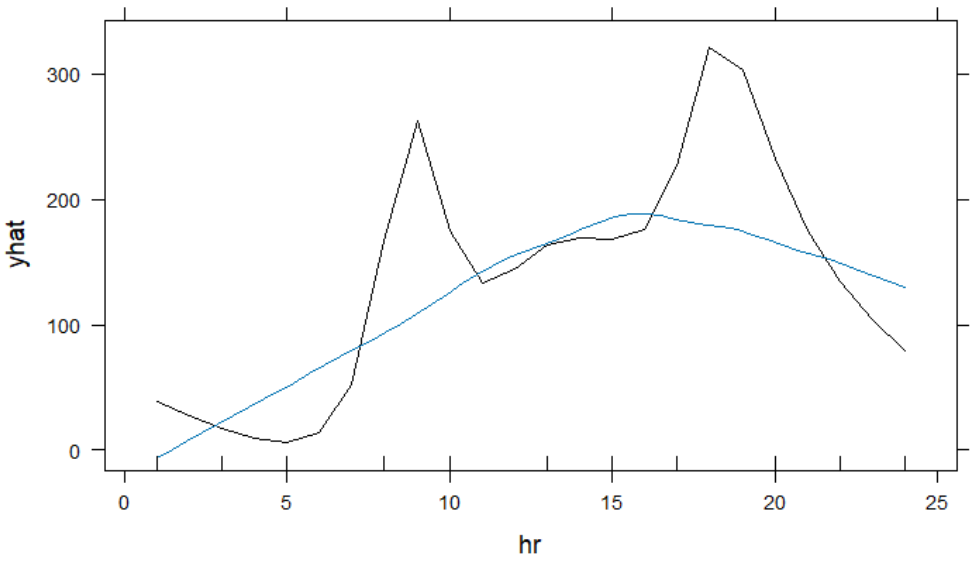
The ATPA modules only illustrate PDPs for PA-level models such as linear models, GLMs, and ensemble trees, but not the advanced predictive models covered in Chapter 1. Just out of curiosity, CHUNK 5 refits the final neural network in Subsection 1.3.2 and uses it to produce the PDP for `hr`. (If you are interested, the extra options inserted to the `partial()` function are needed because the `predict()` function applied to an `ANN2` object is a list, not a vector.)

```
# CHUNK 5
# Repeat OHE from Subsection 1.3.2
binarizer <- dummyVars(~ mnth + weathersit, data = Bikeshare)
Bikeshare <- cbind(Bikeshare, data.frame(predict(binarizer, Bikeshare)))

# Repeat the creation of X matrices and y vectors for building neural networks
X_train <- Bikeshare[train_ind, !names(Bikeshare) %in% c("mnth", "weathersit",
                                                         "bikers")]
y_train <- Bikeshare[train_ind, "bikers"]

library(ANN2) # for fitting neural networks
nn <- neuralnetwork(
  X = X_train,
  y = y_train,
  hidden.layers = c(20, 15),
  regression = TRUE,
  standardize = TRUE,
  loss.type = "squared",
  activ.functions = "tanh",
  learn.rates = 1e-03,
  n.epochs = 500,
  batch.size = 32,
  val.prop = 0.1,
  random.seed = 1
)

partial(
  nn,
  train = X_train,
  pred.var = "hr",
  plot = TRUE,
  smooth = TRUE,
  rug = TRUE,
  type = "regression",
  pred.fun = function(object, newdata){
    mean(ANN2:::predict.ANN(object, newdata = newdata)$predictions)
  }
)
```



The shape of the PDP resembles the one produced by the random forest, but the trough between hours 10 and 15 is deeper. Overall, this PDP is more similar to the split boxplots on page 16.

2.1.3 Global Method 3: Global Surrogate Models

Idea. Another possible way to explain a complex model is to approximate the complex model by an interpretable model, such as a linear regression model or a decision tree. The interpretable model is fitted to the predictions of the complex model on the training set as the target variable and serves as a "surrogate" for the latter model. (The original predictors continue to be predictors in the surrogate model.) The surrogate model is unable to pick up all the nuances of the complex model, but we are able to explain the predictions easily due to the surrogate's inherent interpretability. This method concerns all the observations in the training data, so it is a global method. It also works for all types of predictive model, so it is model-agnostic.

R demonstration. In CHUNK 6, we fit a linear regression model to the predicted values of the tuned neural network on the training set (called `prediction`) and output the model summary.

```
# CHUNK 6
# Create a new data frame containing the neural network predictions
dat_train_surrogate <- dat_train
dat_train_surrogate$prediction <- predict(nn, newdata = X_train)$predictions

# Fit the surrogate LM
lm_surrogate <- lm(prediction ~ . - bikers, data = dat_train_surrogate)
summary(lm_surrogate)


Call:
```

```
lm(formula = prediction ~ . - bikers, data = dat_train_surrogate)

Residuals:
    Min      1Q  Median      3Q     Max
-245.74  -67.26  -20.42   45.19  340.93

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)              -1.4792     8.3703  -0.177 0.859729
mnthFeb                  -5.9325     6.5114  -0.911 0.362286
mnthMarch                -3.3680     6.5389  -0.515 0.606528
mnthApril                18.6604     7.5107   2.485 0.012999 *
mnthMay                  43.9610     8.5298   5.154 2.63e-07 ***
mnthJune                 11.7250     9.6789   1.211 0.225787
mnthJuly                -14.5051    10.3998  -1.395 0.163142
mnthAug                  -0.4845     9.7913  -0.049 0.960537
mnthSept                 41.3935     9.0886   4.554 5.36e-06 ***
mnthOct                  54.3948     7.6031   7.154 9.41e-13 ***
mnthNov                  50.5866     6.9905   7.236 5.18e-13 ***
mnthDec                  39.5437     6.5494   6.038 1.66e-09 ***
hr                        6.2031     0.1946  31.878  < 2e-16 ***
holiday                 -18.9153     7.9406  -2.382 0.017245 *
workingday               -2.4931     2.8110  -0.887 0.375179
weathersitcloudy/misty    5.8669     3.1347   1.872 0.061311 .
weathersitrain/snow     -17.1621     4.9471  -3.469 0.000526 ***
temp                    298.1754    14.6537  20.348  < 2e-16 ***
hum                    -155.4050     8.1553 -19.056  < 2e-16 ***
windspeed                21.8303    11.0602   1.974 0.048454 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 97.75 on 6034 degrees of freedom
Multiple R-squared:  0.4514,    Adjusted R-squared:  0.4497
F-statistic: 261.3 on 19 and 6034 DF,  p-value: < 2.2e-16
```

The linear surrogate model confirms that `hr` and `temp` are the most significant predictors of the neural network's predictions and, by extension, `bikers`, as evidenced by their large t-values in absolute value (31.878 and 20.348). However, the surrogate model is unable to capture the non-linear relationships between `bikers` and each of `hr` and `temp`.

2.1.4 Local Method 1: ICE Plots

The next two interpretational techniques are local in nature.

Idea. ***Individual conditional expectation*** (ICE) plots are local versions of **PDPs** and display the marginal effect of a predictor on the target variable for each observation separately.

Mathematically, the ICE for X_1 (a predictor of interest) and the ith observation in the training set is

$$\text{ICE}_i(x_1) = \hat{f}(x_1, x_{i2}, \ldots, x_{ip}),$$

where:

- $\hat{f}$ is the fitted predictive model.
- x_1 is a fixed value or level of X_1.
- $x_{i2}, \ldots, x_{ip}$ are the values or levels of $X_2, \ldots, X_p$ (predictors not of interest) for the ith observation.

If we plot $\text{ICE}_i(x_1)$ as a function of x_1 for each $i = 1, \ldots, n$, then we will get a collection of curves, one corresponding to each training observation. These curves together make up an ICE plot.

Note that unlike $\text{PD}(x_1)$ in (2.1.1), no averaging is taken over the training set to get $\text{ICE}_i(x_1)$. This is because ICE plots are a local interpretability method, aiming to show how the model prediction behaves for each individual observation. In fact, averaging the individual ICE curves over the entire training set retrieves the (global) partial dependence curve:

$$\frac{1}{n}\sum_{i=1}^{n} \text{ICE}_i(x_1) \overset{(2.1.1)}{=} \text{PD}(x_1).$$

Pros and cons of ICE plots.

⊕

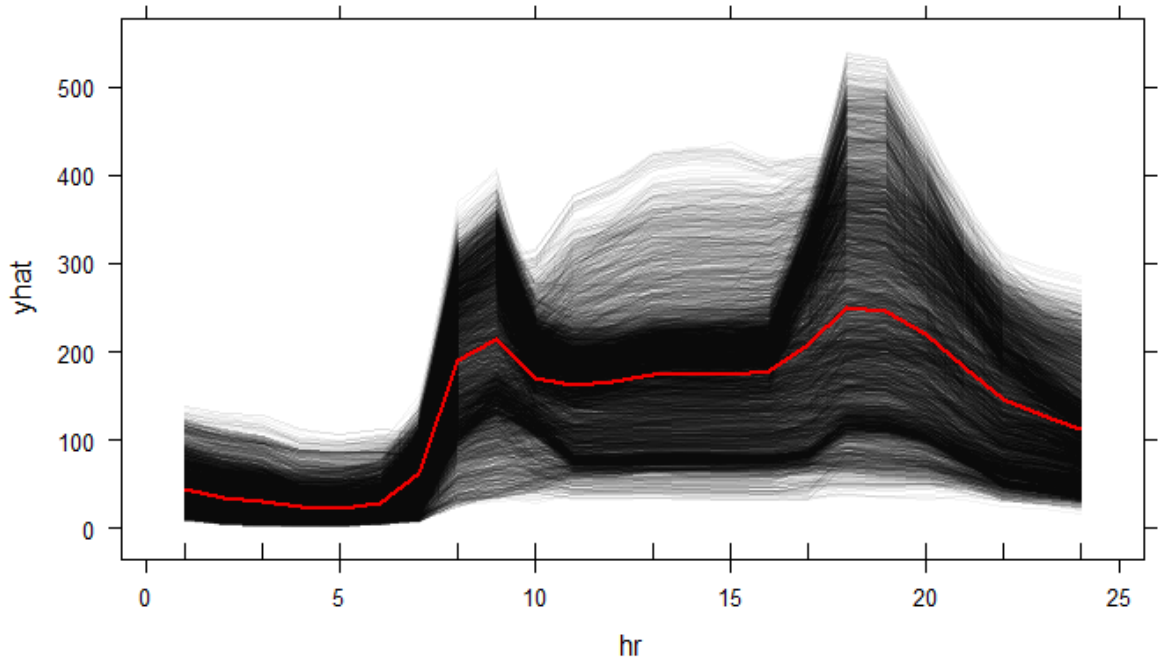
- *(Ability to capture heterogeneous relationships)* By design, ICE plots overcome one of the main problems with PDPs we discussed in Subsection 2.1.2 with regard to the heterogeneous effects of interactions. With every observation displayed separately in an ICE plot, we can visually inspect if the relationships between the model predictions (as a proxy of the target variable) and the predictor of interest are different for different observations. If the relationships do vary substantially, that speaks to the interactions that may exist in the data.
- *(A weaker point: Simplicity)* To some, ICE plots may be more intuitive and easier to understand than PDPs because there is no need to average the model predictions over the training set.

⊖

- *(Independence)* ICE plots suffer from the same independence problem as PDPs in the sense that the way the ICE curves are generated still assumes that the predictor of interest is independent of other predictors. The curves may be evaluated at previously unseen and practically unreasonable combinations of predictor values.
- *(Ease of visual interpretation)* Even for moderately sized training data, an ICE plot can easily become overcrowded. There are so many curves that you can see hardly anything. Some potential solutions include adding transparency to the ICE curves and drawing only a random sample of the curves (at the expense of a loss of information).

R demonstration. In R, ICE plots can be produced by the same `partial()` function for making PDPs, but with the option `ice = TRUE` inserted. As an example, run CHUNK 7 to make an ICE plot for the `hr` variable using the random forest fitted in Subsections 2.1.1 and 2.1.2.

```
# CHUNK 7
partial(
  RF,
  train = dat_train,
  pred.var = "hr",
  plot = TRUE,
  rug = TRUE,
  ice = TRUE, # make an ICE plot rather than a PDP
  alpha = 0.05 # make the lines more transparent
)
```



The ICE curves follow more or less the same wave-like course (although some take unusually large values between 10 a.m. and 5 p.m.), meaning that the hour-bikers relationship is quite consistent over the observations. With no obvious interactions, the PDP we made in CHUNK 4 appears to be a good summary of the marginal relationship between `hr` and `bikers`.

2.1.5 Local Method 2: SHAP

Idea. *Shapley values* provide a way to interpret a model using concepts from coalitional game theory, which is a discipline at the intersection of economics and mathematics. When applied to explaining models, this technique is often called ***Shapley Additive Explanations*** (SHAP).

The technical details of Shapley values are beyond the scope of ATPA.[2] Loosely speaking, we think of each predictor value of a given observation in the data as a "player," and these players are playing a "game" where they collaborate with each other to produce the model prediction of the given observation (more precisely, the model prediction in excess of the average value of the target variable) as the "payout." Shapley values then provide a quantitative method for distributing the game "payout" among the team "players" in a fair manner. From the Shapley values, we can gain insights into how each predictor moves the observation away from the average value of the target variable.

Example: As a simple example, suppose that the average of the (numeric) target variable on the training set is 500, and there are $p = 3$ predictors, whose Shapley values for a particular observation are 50, -20, and 30, respectively. These values account for the deviation of the model prediction for this observation from 500 as the baseline, and the model prediction equals $500 + 50 + (-20) + 30 = 560$.

R demonstration.

(The SOA's code in CHUNKs 11-14 of the Rmd file for Section 4.3 does not seem to work on relatively new versions of R or the `shapr` package. If a workaround other than downgrading R or `shapr` to lower versions is available, we will make an announcement on the ATPA forum in the Discord channel.)

2.1.6 Digression: Lift and Gain Charts

The interpretational methods thus far all serve to connect the inputs to the output of the model. For some reason, the ATPA modules conclude the discussion of model interpretation with two model-agnostic methods that are not exactly about explanations or interpretations. They are graphical methods that attempt to demonstrate the quality of a binary classifier, i.e., the target variable is a categorical variable that takes only two levels, which we code as "positive" and "negative." Although these visuals can be constructed on any set of data, we typically do so on the test (or held-out) set to assess the performance of a classifier on previously unseen data and prevent overfitting.

Graphical Method 1: Lift Charts

To construct lift and gain charts for a binary classifier on a set of data, we need the following two ingredients for each observation (which are readily available):

- The predicted probability of a positive outcome produced by the classifier
- The true class label of the target variable (positive or negative)

[2] If you are interested, you may read the supplementary notes (https://cdn-files.soa.org/e-learning/atpa/4.3_jobaid_shapley_values.pdf) prepared by the SOA or Sections 9.5 and 9.6 of *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*.

Idea. Here is the construction procedure for a **lift chart**:

Step 1. Sort the observations in *descending order* of the predicted probability of a positive outcome. In other words, the first observation has the highest predicted probability and the last observation has the lowest.

Step 2. Starting with the first observation, compute the following ratio for each observation:

$$\frac{\text{Cumulative number of } \oplus \text{ based on } \textbf{sorted} \text{ data}}{\text{Cumulative number of } \oplus \text{ based on } \textbf{random} \text{ data}}. \tag{2.1.2}$$

The meaning of the numerator of this ratio should be clear. We simply count how many positive outcomes we have seen as we traverse the whole set of sorted data.

The denominator looks more intricate and is based on the hypothetical, randomly shuffled data where each observation has the same probability of being a positive outcome. If, for example, 200 out of 1,000 observations are positive responses, then each observation has a probability of $200/1{,}000 = 0.2$ to be positive, and the cumulative numbers of positives are $0.2, 0.4, 0.6, \ldots, 199.8, 200$, rising by increments of 0.2, as we go from the first observation to the last observation.

Step 3. Plot the ratios in Step 2 in order of the observations.

By design, a lift chart provides a visual assessment of the quality of a classifier relative to purely random classifications. If the classifier is successful in detecting positive outcomes, then the numerator of (2.1.2) (based on the classifier and the associated predictors) should increase more rapidly than the denominator of (2.1.2) (based on random chance), leading to lift chart values that are consistently larger than 1. The more the points on the lift chart stay above 1, the better the classifier. Regardless of how good the classifier is, the values will always converge to 1 as we hit the last observation—the total numbers of positives must be the same whether it is the sorted data or the random data.

⚠ EXAM NOTE ⚠

If lift and gain charts are tested in your ATPA Assessment, then very likely you will be asked to produce and examine the charts for one or more classifiers, and say something about the (relative) quality of model fit, so let's look at some concrete lift and gain charts.

Illustrative example. It is much easier to see how things work in the context of a simple example. Let's consider the following toy dataset with five observations:

Observation	Target Variable	Predicted Probability of Positive Class
1	+	0.8
2	−	0.2
3	+	0.4
4	+	0.9
5	−	0.6

To begin with, we sort the five observations in descending order of the predicted probability, which leads to the following sorted data:

Observation	Target Variable	Predicted Probability of Positive Class
1	+	0.9
2	+	0.8
3	−	0.6
4	+	0.4
5	−	0.2

For convenience, we drop the original observation numbers, which play no role, and relabel the observations in descending order of the predicted probability, e.g., Observation 1 in the unsorted data becomes Observation 2 in the sorted data.

Now let's work on (2.1.2).

- *(Numerator)* From the true class labels of the observations in the second column of the table, we can get the cumulative number of positive responses by direct summation:

Observation	Target Variable	Cumulative Number of + Responses
1	+	1
2	+	2
3	−	2
4	+	3
5	−	3

  For example, both observations 1 and 2 are positive, so the cumulative number as of observation 2 is $1 + 1 = 2$. Observation 3 is negative, so the cumulative number as of observation 3 remains 2.

- *(Denominator)* If the observations were randomly sorted, then with 3 positives out of 5 observations, we would expect each observation to be positive with a probability of $3/5 = 0.6$. As we go past each observation, the cumulative number of positive responses would increase by 0.6, leading to the following table:

Observation	Cumulative Number of + Responses
1	0.6
2	1.2
3	1.8
4	2.4
5	3.0

Taking the ratio of the values in the two tables above, we get:

Observation	Value of (2.1.2)
1	$1/0.6 = 1.6667$
2	$2/1.2 = 1.6667$
3	$2/1.8 = 1.1111$
4	$3/2.4 = 1.25$
5	$3/3 = 1$

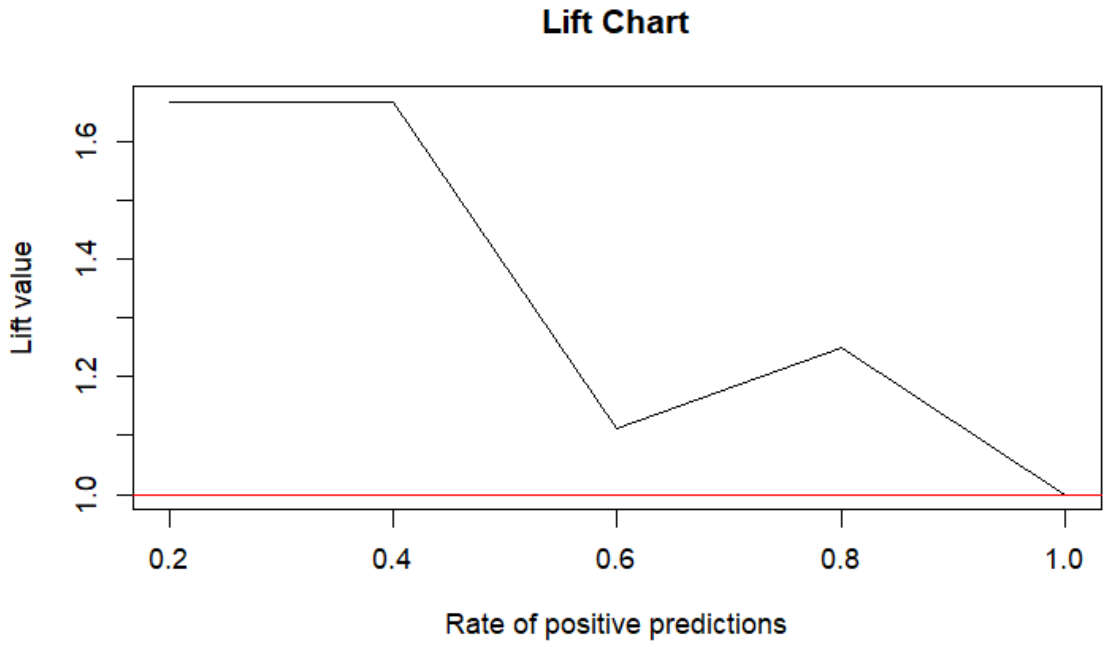
In CHUNK 8, we follow the ATPA modules and use functions in the `ROCR` package to make the lift chart for the dataset above.

```
# CHUNK 8
# Uncomment the next line the first time you use ROCR
# install.packages("ROCR")
library(ROCR)

# Set up the toy data
pred <- c(0.8, 0.2, 0.4, 0.9, 0.6)
truth <- c("+", "-", "+", "+", "-")

# Generate an object that compares predictions against truth
compare_pred <- prediction(pred, truth)

# Construct a lift chart
lift <- performance(compare_pred, measure = "lift", x.measure = "rpp")
plot(lift, main = "Lift Chart")
abline(h = 1, col = "red") # reference line
```



The ATPA modules and the associated Rmd files don't bother to explain what the R functions above are doing. If you are interested, we first use the `prediction()` function (not the usual `predict()`) in the `ROCR` package to transform the input data (including the predicted probabilities and true class labels) into a single R object, which is then passed to the `performance()` function to produce a wide variety of performance evaluations, depending on how the `measure` (for the performance metric on the y-axis) and `x.measure` (for the performance metric on the x-axis) arguments are specified. If `measure = "lift"` and `x.measure = "rpp"`, as in CHUNK 8, then we are plotting the lift value defined in (2.1.2) against the rate of positive predictions, which is the (cumulative) proportion of positive predictions in the data as we go from the first observation ($20\% = 0.2$) to the last observation ($100\% = 1.0$).

The lift chart above starts with relatively high values at the first two observations, which are indeed positive, then suffers a drop as we arrive at the third observation, which is actually a negative outcome despite the relatively high predicted probability. The chart rises again at the fourth observation, which is positive, then converges to 1 at the fifth and last observation, as it should. Because all of the chart values (except the last one) are moderately above 1, the classifier has a modest amount of predictive power on the toy dataset.

Graphical Method 2: Gain Charts

Idea. Similar in spirit to a lift chart, a **gain chart** is another graphical method (actually an equivalent method, as will be discussed below) for assessing the quality of fit of a binary classifier. As with a lift chart, we first rank the observations in descending order of the predicted probability of a positive outcome, but instead of plotting the ratio of cumulative numbers of positive outcomes for the sorted data relative to the random data, we plot the following pair of cumulative proportions of positive outcomes for each observation:

$$\left(\begin{array}{c} \text{cumulative proportion of + outcomes} \\ \text{based on } \textbf{random} \text{ data} \end{array} , \begin{array}{c} \text{cumulative proportion of + outcomes} \\ \text{based on } \textbf{sorted} \text{ data} \end{array} \right). \quad (2.1.3)$$

Intuitively, if a binary classifier predicts positive outcomes accurately, then the cumulative proportions based on the sorted data should increase much faster than the cumulative proportions based on the random data. This will be reflected in a gain chart where the points lie well above the straight line connecting $(0, 0)$ and $(1, 1)$. (In this connection, a gain chart is akin to an ROC curve.)

Illustrative example. Let's try to construct the gain chart for the toy dataset above, reproduced below for your convenience:

Observation	Target Variable	Predicted Probability of Positive Class
1	+	0.9
2	+	0.8
3	−	0.6
4	+	0.4
5	−	0.2

From this table, we can easily compute the cumulative proportions of positive responses:

Observation	Target Variable	Cumulative Proportion of ➕ Responses Based on Random Data	Cumulative Proportion of ➕ Responses Based on Sorted Data
1	+	0.2	1/3
2	+	0.4	2/3
3	−	0.6	2/3
4	+	0.8	3/3 = 1
5	−	1.0	1

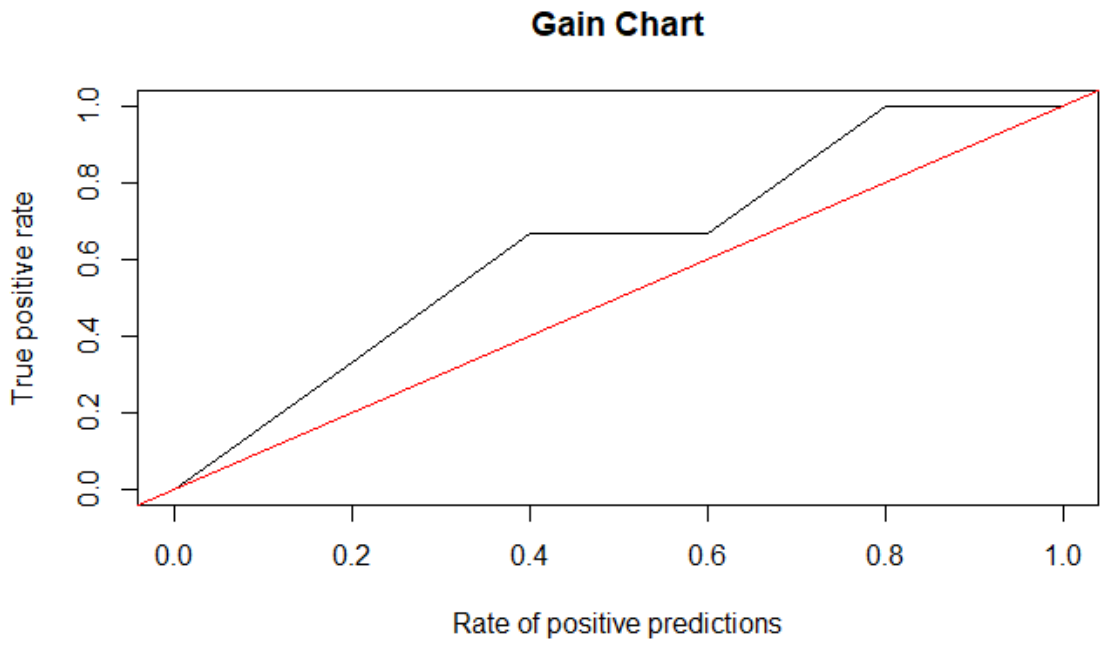
Let's take the first observation as an example.

- Based on the random data, each of the five observations has the same chance to be positive, so the cumulative proportion of positive outcomes is simply $1/5 = 0.2$.
- Based on the sorted data, the first observation is indeed a positive outcome. With a total of 3 positive outcomes in the data, the cumulative proportion of positive outcomes is $1/3$.

Therefore, the first point on the gain chart is $(0.2, 1/3)$.

In CHUNK 9, we make the gain chart for the dataset above.

```
# CHUNK 9
# Construct a gain chart
gain <- performance(compare_pred, measure = "tpr", x.measure = "rpp")
plot(gain, main = "Gain Chart")
abline(a = 0, b = 1, col = "red") # reference line
```



The `measure` argument of the `performance()` function is set to `"tpr"`, meaning "true positive rate," and the last line of the chunk produces the reference line passing through $(0, 0)$ and $(1, 1)$ for comparison. You can see that the points on the gain chart are slightly above the reference line, which suggests a modest amount of predictive power. This is in agreement with the lift chart we saw earlier.

A closing remark. The ATPA modules introduce lift charts and gain charts as separate and unrelated graphical methods. This is somewhat unfortunate because there is actually a 1-to-1 correspondence between the two charts. The correspondence lies in the fact that:

> If (x, y) is a point on a gain chart, then $(x, y/x)$ must also be a point on the corresponding lift chart.
>
> (Equivalently, if (x, y) is a point on a lift chart, then (x, xy) must also be a point on the corresponding gain chart.)

This follows immediately by definition when you compare (2.1.2) and (2.1.3), and divide the numerator and denominator of (2.1.3) by the total number of positive outcomes to change "number" to "proportion":

$$\begin{aligned} &\frac{\text{Cumulative } \textbf{number} \text{ of } \oplus \text{ based on ranked data}}{\text{Cumulative } \textbf{number} \text{ of } \oplus \text{ based on random data}} \\ = &\frac{\text{Cumulative number of } \oplus \text{ based on ranked data} \,/\, \text{total number of } \oplus}{\text{Cumulative number of } \oplus \text{ based on random data} \,/\, \text{total number of } \oplus} \\ = &\frac{\text{Cumulative } \textbf{proportion} \text{ of } \oplus \text{ based on ranked data}}{\text{Cumulative } \textbf{proportion} \text{ of } \oplus \text{ based on random data}}. \end{aligned}$$

Taking the fourth point on the gain chart, $(0.8, 1)$, as an example, we can easily check that $(0.8, 1/0.8) = (0.8, 1.25)$ is the corresponding point on the lift chart. Although not noted in the ATPA modules, this correspondence between a lift chart and a gain chart means that they are essentially equivalent graphical tools for demonstrating the performance of a classifier. The sad news is:

> In the presence of one chart, the other chart does not really add much value!

Appendix A

Where to Go Next?

At this point, you have almost reached the end of your preparation for the ATPA Assessment. Before you set aside 96 hours and formally begin the assessment, you may want to work on the three things below (so long as time permits).

A.1 The Sample Assessment

The SOA has designed a sample ATPA Assessment. All relevant files, including the model solution, data files, and Rmd files, can be downloaded from

https://www.soa.org/499563/globalassets/assets/files/edu/2022/atpa-sample-assessment.zip

While I suggest studying the sample assessment in some depth, I will not recommend treating it too seriously or getting bogged down in all of the tricky details in the SOA's "super comprehensive" solution, which could hardly be done in 96 hours. Instead of trying to match everything in the model solution, pay attention to the rationale behind the various actions that the solution takes, and if applicable to your real assessment, incorporate those actions into your answers. When I took the assessment back in April 2023, my solution was far simpler and more direct than that of the SOA's solution, and...

I was able to pass! 😜

A.2 Advance Preparation

As you know, the real ATPA Assessment is open-book, but quite a lot of students found that 96 hours are barely enough to complete the assessment, and advance preparation will save you plenty of time and make your life during the 96-hour window. Before starting the assessment, you may want to:

(1) Prepare a Microsoft Word file containing a short description, preferably in your own words, of each of the advanced predictive models covered in Chapter 1.

If you are asked to fit those models in the real assessment, then start your response with the description you have prepared beforehand. To maximize the chance of a pass, grasp

every chance to signal your understanding to the grader and show your thought process 🧠, even at the risk of stating the obvious.

(2) Save all Rmd files that accompany this study manual in a single folder in your computer so that they are readily accessible, or even better, prepare a master Rmd file containing the essential R code from the various Rmd files.

A.3 Practicing Copy and Paste from RStudio to Word

In the real assessment, there will be quite a few instances in which you may want to transfer your output in RStudio to the Word file that you will upload in order to back up your answers.

- ⚠ For text-based output (e.g., model summaries), you may try to copy and paste the R output to the Word file, then change the font to a monospaced one, such as Consolas, Courier New, and Lucida Console. The output should align properly and closely resemble what you see in RStudio:

- For graphical output, "copy and paste" works even without any adjustments.

R code adds marginal value and need not be pasted, however.

Some students may prefer to take screenshots, but the instructions in the assessment may or may not allow screenshots, e.g., in recent sittings, pasting an Excel table as a picture will result in an automatic disqualification of your submission. In any case, it is crucial to read and fully comply with the SOA's instructions.

At the end of the day, the format of presentation of the output is of secondary importance at best. The much more important thing is the quality of your write-up (in particular, the communication of your predictive modeling work) in the Word report.

. . . . . .

Although not exactly a proctored exam that requires several hundred hours of study, the ATPA Assessment is by no means a breeze, but hopefully this study manual and the last-minute tips above can reduce some of the stress and make your experience less painful.

ATPA is the last member in the SRM-PA-ATPA Trio in the ASA curriculum. I am truly grateful to all of you 🤝 for choosing to use this study manual (as well as my SRM and PA manuals), and it is my honor to keep you company in your exam-taking journey. Last but not least...

I Wish You The Best of Luck in Your ATPA Assessment

and, More Importantly, a Rewarding Actuarial Career! 😊

Appendix R

A Crash Course in R for ATPA

Appendix overview: In the ATPA Assessment, you will manipulate data in R, create graphs that turn the raw data into useful insights, and construct various predictive models. Proficiency with R programming is therefore critical to success in the assessment.

To get you up to speed, this appendix (which is a version of the crash course in the *ACTEX Study Manual for Exam PA*) provides a minimal introduction to R and covers some simple R commands and functions that are used in the main text of this study manual. We begin in Section R.1, where we set up our R toolkit and learn about the three most commonly used types of data in R. Section R.2 expands the discussion in Section R.1 and presents four important data structures that R uses to store information. The subtle differences between these data structures are highlighted. Section R.3 discusses `for` loops, which are important programming tools for reducing repetition in our code. Section R.4 concludes this appendix with a minimal introduction to the `ggplot2` package for data visualization.

R.1 Getting Started in R

Developed in the early 1990s, R[1] is an extremely versatile open-source programming language for statistical computing, graphics, and, in recent years, document preparation. For ATPA, the most conspicuous merits of R are three-fold:

- *(It's free!)* Unlike many commercial statistical software platforms, R is free to use. This point is important as it ensures that every ATPA candidate can access and practice using the software platform required for the assessment for free.

  (Well, we still have to pay $1,230 for the ATPA Assessment!! 😠)

- *(Making graphs)* R has superior graphics capabilities. It allows you to create elegant, informative, and customizable graphs which would be difficult, if not impossible, to create in other programming languages.

[1] The name of R comes from the shared first letter of the first names of the two inventors, Professors Robert Gentleman and Ross Ihaka.

- *(Numerous packages)* R outperforms many other software platforms due to the wide collection of add-on packages that perform a wide range of tasks, general and specialized, and significantly enhance the functionality of R. More impressively, these packages can be downloaded from online repositories for free.

In this introductory section, I will walk you through the installation of **R** and **RStudio** (a good companion to R), the basic features of RStudio, and the common data types in R.

R.1.1 Basic Infrastructure

First things first, let's install R and RStudio on your computer (if you have not already done so).

⚠ NOTE ⚠

If your computer already has R installed and its version is 4.0 or above, then for the purposes of this manual and the ATPA Assessment there is no need to reinstall R.

Installing R. R can be freely downloaded and installed from *The Comprehensive R Archive Network* (CRAN) at https://cran.rstudio.com. The top of the web page shows three links for downloading R:

- Download R for Linux
- Download R for macOS
- Download R for Windows

Choose the link that works for the operating system of your computer. To install R on Windows, for example, click "Download R for Windows," then the "base" link, and finally "Download R 4.5.2 for Windows" (4.5.2 will soon be replaced by a more updated version of R). An installer program will be downloaded. Running the program and stepping through the installation wizard (the default choices will do) will install R into your program files and place a shortcut in your Start menu. When you open R (by clicking its icon on your desktop, for example), you should see a window like Figure R.1.1. If so, you have successfully installed R.

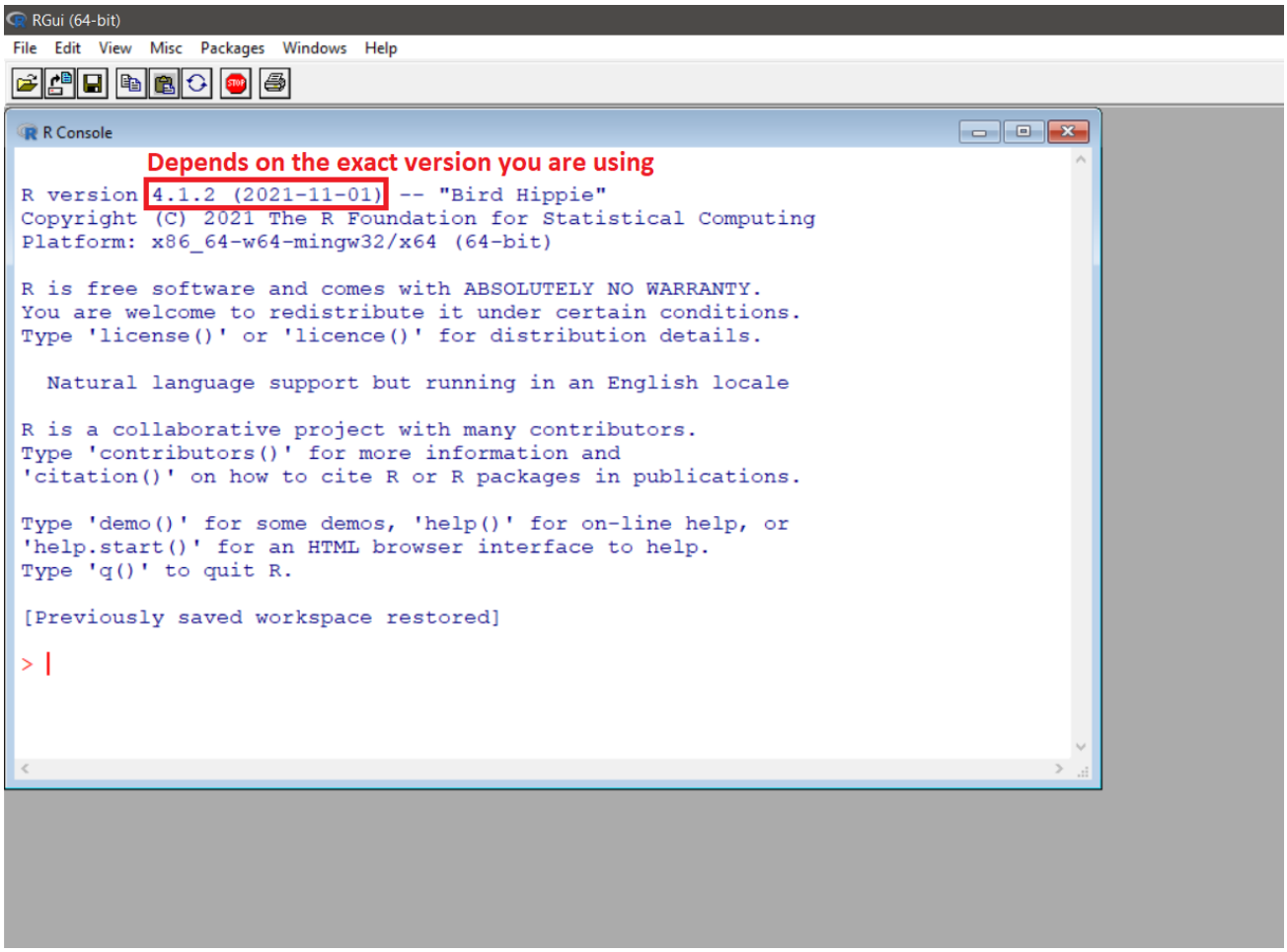


Figure R.1.1: The basic command line interface of R.

Installing RStudio. Instead of using R's primitive command line interface in Figure R.1.1, it will be much easier to work with R within *RStudio*, which is the most popular *integrated development environment* (IDE) for R and is available for free. An IDE is a software application that provides comprehensive facilities to programmers. Typically, it has a graphical user interface (GUI) and consists of at least a code editor, a compiler or interpreter, and a debugger. The RStudio IDE in particular offers the following valuable features:

- Code highlighting and prediction as well as syntax error detection, helping you write correct code much more efficiently than in R's command line interface.
- The ability to generate dynamic reports where R code and output are integrated into a text document.
- Management of installed packages, environments, and command history all within one workspace.

We can succinctly put the relationship between R and RStudio this way:

> R is the language to speak in and RStudio provides a convenient platform for you to talk to your computer in this language.

RStudio can be downloaded and installed from

https://posit.co/download/rstudio-desktop/.

Look for "RStudio Desktop," select the version corresponding to the operating system of your computer, and follow the instructions there to get RStudio set up.

Layout of RStudio. When you first launch RStudio (by clicking its icon on your desktop, for example), you will see a screen that looks like Figure R.1.2 with four distinct parts:

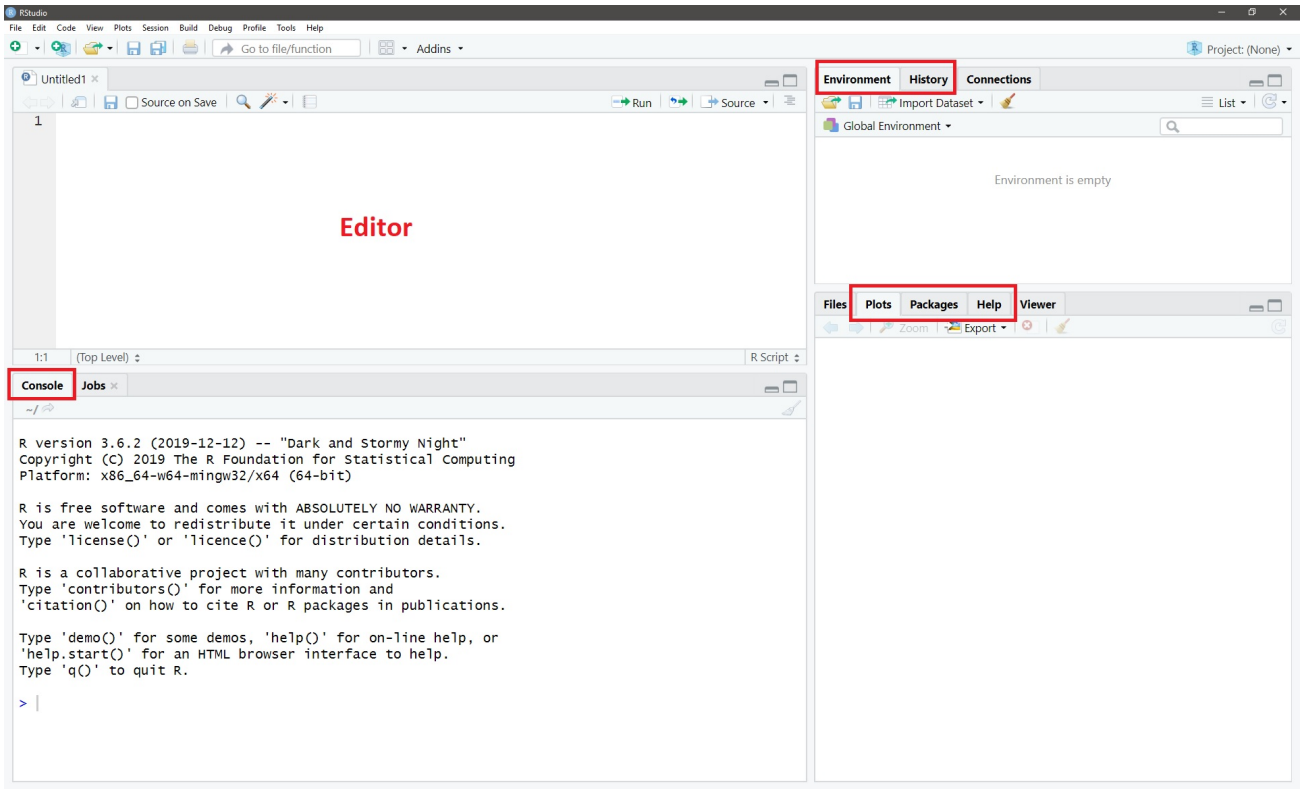


Figure R.1.2: The basic interface of RStudio.

- *Lower left pane—console:* This is the direct command line interface to R. All output from running commands and scripts will be shown here. To directly run a command here, place the cursor on the command prompt indicated by the right arrow `>`, type the command, and press `Enter`. Unless your command is extremely short and simple, you would be better off typing and saving your commands in the editor pane.

- *Upper left pane—editor:* The editor provides different kinds of files where you can compose and save commands for further manipulation. If your editor is closed for whatever reason, you can make it appear by going to the menu bar and selecting `File > New File >`, followed by the type of file where you want to do the editing. The simplest choice is an R script, a plain text file where you can write, edit, and run all or part of your R code. This creates a reproducible record of your work for future use.

- *Lower right pane:* This is where you can find information about plots, packages, and help files.
  - ▷ The `Plots` tab displays all of the plots you make if you run commands in the R console or in an R script in the editor pane. The plots can be saved and exported to external files. This tab is not particularly important for ATPA as the plots generated by running commands in R Markdown files will be shown directly underneath your commands.
  - ▷ When you click the `Packages` tab, you can see a list of installed R packages (we will further discuss packages on page 289). Notice that installed packages are not automatically loaded into your current R session because it would be unmanageable and slow to load them all at once. To load an installed package, simply check the box for that package and it will be available for use. To install a new, external package, click `Install` and type the name of the package. Note that you only have to install a new package once.
  - ▷ The `Help` tab is where you can find help documentations of R functions and installed packages. To get help on a function, type the command (either in the R console or in the editor pane)

    `help("FUNCTION_NAME")` or simply `?FUNCTION_NAME`,

    where `FUNCTION_NAME` is a placeholder for the name of the function of interest. To access the documentation of an installed package, use the command

    `help(package = "PACKAGE_NAME")`,

    where `PACKAGE_NAME` is the name of the package of interest.
- *Upper right pane:* This pane contains information about the workspace and command history. In the `Environment` tab, you can see a list of objects that you have imported or created in the current environment. If you have created a data object, then clicking on the name of the object will open a new tab in the editor showing how the object looks. This can ensure that the data object is set up properly. The `History` tab records all of the commands you have run and allows you to reuse or modify old commands.

You will get fully comfortable with these four panes as you work through this study manual.

R Markdown files. This study manual (and the ATPA modules) comes with ***R Markdown*** (Rmd) files with clearly numbered R chunks. An Rmd file is simply a plain text document in which text is interspersed with chunks of R code. When the file is compiled, a new document with the code chunks turned into their output is generated. In this day and age, most data scientists turn to R Markdown as the preferred means of communicating results due to its capability to generate dynamic reports in different formats (e.g., PDF, HTML, slideshows), where R code and output are embedded with texts for expository convenience. The results produced are internally tied to the code, so changing the data input automatically updates the output. In contrast, if you typeset your report in an external word-processing document like Word and your data or

code changes, you will have to take the trouble to rerun your code in R, and copy and paste the output from R to your Word file manually, not to mention that Word files are no match for reports generated by R Markdown in terms of typesetting quality.[2] (Ironically and very sadly, Microsoft Word is the channel through which you communicate to the SOA in the ATPA Assessment.)

When you open an Rmd file (try `File > New File > R Markdown...`), you will see texts with a white background together with texts highlighted with a gray background, called ***R chunks***, as in Figure R.1.3.

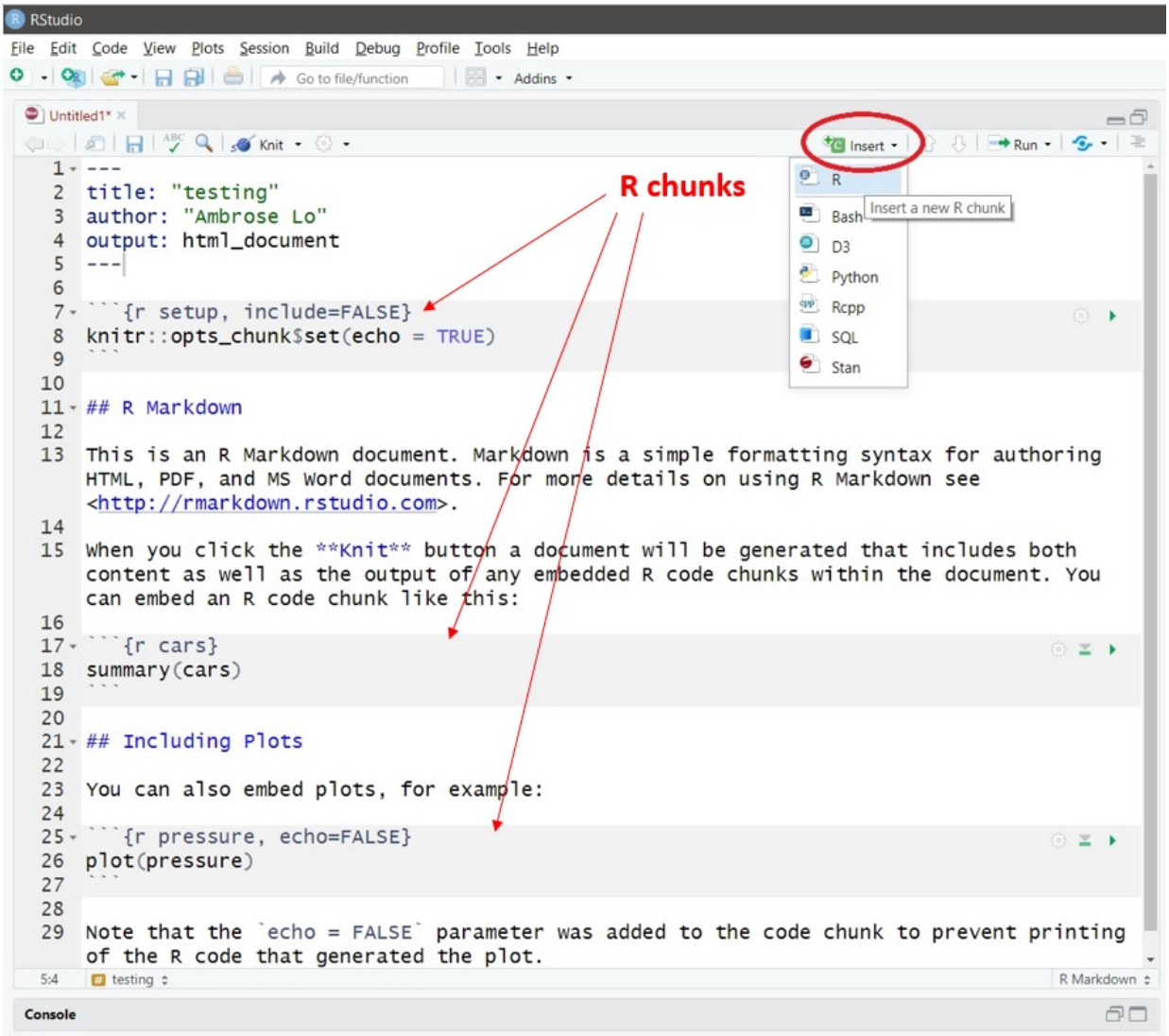


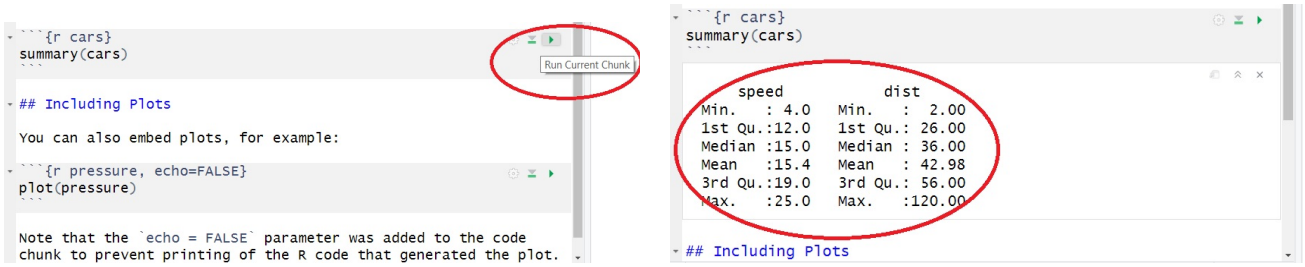
Figure R.1.3: The typical structure of an R Markdown file.

R chunks are where R commands are stored, executed, and separated from other texts. To insert a new R chunk, select `Insert a new code chunk > R` in the top right corner of the file. This will produce a new chunk with three backticks signifying the beginning and end of the chunk. To run all of the code in an R chunk, click the green triangle at the top right corner of

[2] This study manual is typeset in LaTeX, not in Word!

the chunk (or press `Ctrl + Shift + Enter`) and the output, if any, is shown right below each chunk; see Figure R.1.4. If you want to run only part of the code in an R chunk, highlight the commands you want to run and click `Run > Run Selected Line(s)` at the top right corner of the Rmd file, or simply press `Ctrl + Enter`.

When you click the Rmd file you want to work with, this will automatically open RStudio and load the file at the same time.



(a) Clicking the right triangle runs the code in an R chunk. (b) Output produced by running an R chunk.

Figure R.1.4: How to run an R chunk and where the output is produced in an Rmd file.

R packages. One of the most important reasons for R's skyrocketing popularity in recent years is the abundance of user-contributed packages available for download and installation. These packages make R highly extensible. In R, a *package* is simply a collection of R functions, datasets, and pre-written code designed for a particular purpose. If you think of a statistical technique, the chances are that someone has written a package for it and contributed it to CRAN.[3] Rather than reinvent the wheel, we can build on the efforts of others.

We have covered how to use the GUI provided by RStudio to install and load packages. An alternative that is preferred from a programming point of view is to include the following commands in your Rmd file: (`PACKAGE_NAME` is the name of the package you want to install or load.)

- To install a new package:

```
install.packages("PACKAGE_NAME")
```

- To load an installed package:

```
library(PACKAGE_NAME)
```

  Quotation marks are optional.

  (**Note:** Many people refer to R packages as "libraries," because you will use the R function `library()` to load a package. Using the terms "packages" and "libraries" interchangeably is innocuous in most instances. Functions in R are discussed further in Subsection R.2.5.)

[3] See https://cran.r-project.org/web/packages/index.html.

Who We Are

A Benefit Corporation Experienced at Teaching Actuaries!

Experienced

More than 50 years experience helping students prepare and pass actuarial exams! We are an eLearning technology and education company leveraging experts in the field to constantly update our learning content in a format that works for you.

Trustworthy

ACTEX Learning is a leading US based provider of study materials for actuarial exams. Our authors and content contributors are renowned academics and Actuaries that are proud to have their names on the cover of our manuals and textbooks!

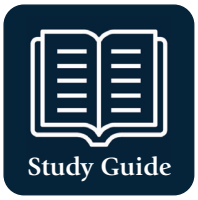
Mission Focused

We are a Benefit Corporation focusing on the mission of accessible high quality actuarial education. We're dedicated to empowering actuarial students by offering test prep materials that are not just effective and efficient but also tailored to suit every type of student.

Become an ACTEX Champion

Join our Global ACTEX Champion Program and bring the benefits to your Actuarial Club. To learn more about the program, scan the QR code on the right. If you have any questions or would like to speak to a Champion Coordinator, please do not hesitate to reach out to us at champions@actexlearning.com.

ACTEX Has the Solutions to Help You with Exam Prep



Our study guides offer the most recommended actuarial prep program. Search our interactive manuals for different topics and toggle easily between concepts and study materials.

Available for P, FM, FAM, ALTAM, ASTAM, PA. ATPA, MAS-I, MAS-II, CAS 5, CAS 6 US & CAN, CAS 7, CAS 8, CAS 9

Want to know you're prepared for your exam? Practice efficiently with our robust database of questions and solutions and predict your success through GOAL's innovative scoring system. GOAL also features dedicated instructor support so you can get help where you need it and pass the exam with confidence!

Available for P, FM, FAM, ALTAM, ASTAM, MAS-I, MAS-II, CAS 5, CAS 6 US & CAN

Master key topics and formulas with our flashcards, which allow you to filter by topic. To help you direct your focus, each card is rated to indicate its importance on the exam.

Available for P, FM, FAM, ALTAM, ASTAM, PA. ATPA, MAS-I, MAS-II, CAS 5, CAS 6 US & CAN, CAS 7, CAS 8, CAS 9

Studies have shown video learning can lead to better retention. We offer hours of video instruction to aid you in your studies. They're a great way to deepen your learning on challenging topics and gain a variety of perspectives from our expert instructors.

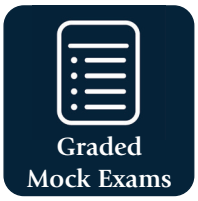
Available for P, FM, FAM, ALTAM, ASTAM, PA. ATPA, MAS-I, MAS-II, CAS 5, CAS 6 US & CAN, CAS 7, CAS 8, CAS 9

ACTEX offers convenient online courses approved for CAS VEE credits. All courses are available on-demand. Students complete the courses at their own pace and take the final exams on the date of their choosing.

Available for Accounting & Finance, Mathematical Statistics, and Economics

Study Materials are Available for the Following:

SOA: P, FM, FAM, ALTAM, ASTAM, SRM, PA, ATPA, CFE, GI, GH, ILA, RET
CAS: MAS-I, MAS-II, CAS 5, CAS 6C, CAS 6US, CAS 7, CAS 8, CAS 9



The ACTEX Graded Mock Exam is a great way to predict your exam outcome! Before you take the official exam - take the new ACTEX Graded Mock Exam and get feedback from an expert. The ACTEX Graded Mock Exam has all the typical elements your SOA exam will have. This can help you evaluate your progress. The questions and format are set up just like the SOA exam.

Available for ALTAM, ASTAM, PA

ACTEX Bootcamps provide a more individualized approach, allow you to ask questions in real time, and boost your last-minute learning. You'll review the harder topics on the exam, as well as common errors and exam strategy. All classes are recorded for future on-demand viewing.

Available for P, FM, FAM, SRM

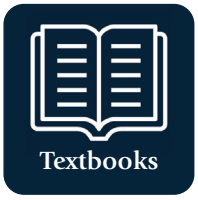
Alongside our P & FM study guide, this course is comparable to a one-semester college class. This course offers SOA Exam practice problems, video solutions, timed practice tests, sample questions, and more. You'll also have 1:1 email support from an instructor for 180 days after purchase.

The Advanced topics in Predictive Analytics video course is designed to help you more easily climb the steep ATPA learning curve. This module-focused video course for Topic 3 in the syllabus, includes videos, end of module assessments and lecture slides. This video course is a deep dive into the three modules. Access to an instructor during the duration of the course as well as participation in a discussion forum.

Available for P, FM, and ATPA

This at-a-glance tool helps you memorize and recall key formulas and information. It covers important formulas needed to prepare your exam. Also, it's an easy-to-print format you can study with, no matter where you are.

Available for P, FM, FAM, ALTAM, ASTAM, PA. ATPA, MAS-I, MAS-II, CAS 5



Looking for Extra Preparation?

Explore our range of textbooks designed to support your studies. From recommended to required readings, ACTEX offers exceptional materials to help you succeed.

ACTEX Learning

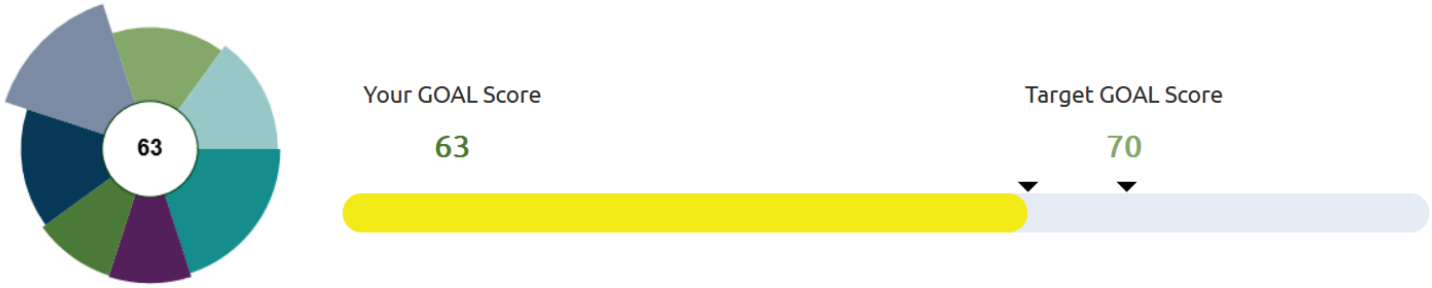
Use GOAL to Practice What You've Learned

- Over 22,000 exam-style problems with detailed solutions
- 3 learning modes (Practice, Quiz, Simulated Exams)
- 3 levels of difficulty (Core, Advanced and Mastery)
- You control your topics and sub-topics
- Dedicated instructor support

GOAL is currently available for the following SOA & CAS Exams:

Exam P 1,050+ Questions	Exam FM 1,500+ Questions	Exam FAM-L 1,300+ Questions	Exam FAM-S 900+ Questions	Exam FAM 2,300+ Questions
Exam ALTAM 1,400+ Questions	Exam ASTAM 1,300+ Questions	Exam SRM 1,150+ Questions	Exam MAS-I 1,050+ Questions	
Exam MAS-II 850+ Questions	Exam CAS 5C 500+ Questions	Exam CAS 6US 550+ Questions	Exam CAS 6CA 650+ Questions	

Use GOAL Score to Gauge Your Exam Readiness



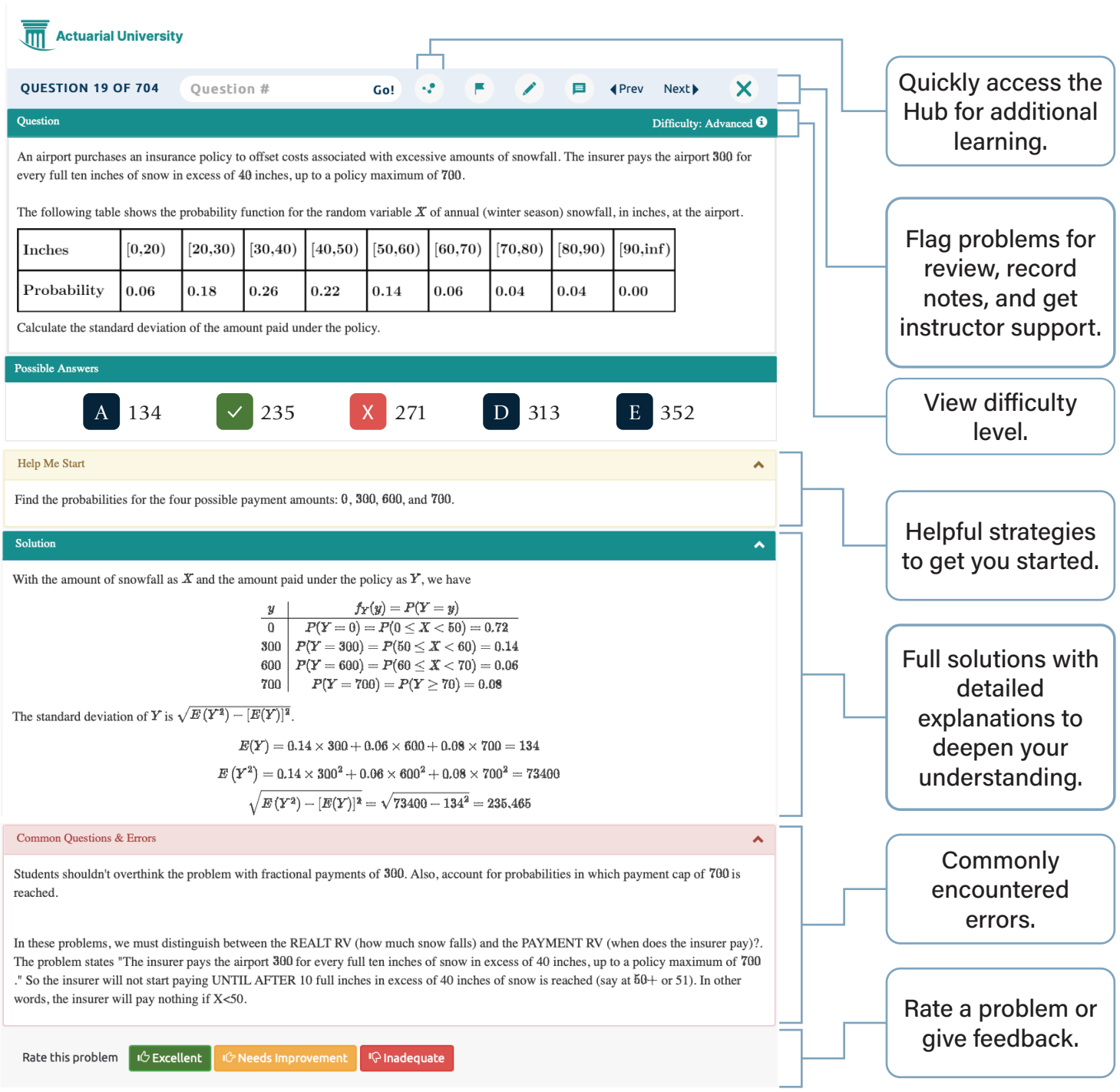
Measure how prepared you are to pass your exam with a tool that suits any study approach. A GOAL Score of 70 or above indicates readiness.

Your score is broken into categories, allowing you to study efficiently by concentrating on problem areas. GOAL Score quantifies your exam readiness by measuring both your performance and the consistency of your performance. Your GOAL Score also analyzes your strengths and weaknesses by category, topic, and level of difficulty.

Scan to Learn More

GOAL Improves Your Studies

How you can prepare for your exam confidently with GOAL custom Practice Sessions, Quizzes, & Simulated Exams:



Actuarial University

QUESTION 19 OF 704 | Question # | Go! | Prev | Next

Question — Difficulty: Advanced

An airport purchases an insurance policy to offset costs associated with excessive amounts of snowfall. The insurer pays the airport 300 for every full ten inches of snow in excess of 40 inches, up to a policy maximum of 700.

The following table shows the probability function for the random variable X of annual (winter season) snowfall, in inches, at the airport.

Inches	[0,20)	[20,30)	[30,40)	[40,50)	[50,60)	[60,70)	[70,80)	[80,90)	[90,inf)
Probability	0.06	0.18	0.26	0.22	0.14	0.06	0.04	0.04	0.00

Calculate the standard deviation of the amount paid under the policy.

Possible Answers

A 134 ✓ 235 X 271 D 313 E 352

Help Me Start

Find the probabilities for the four possible payment amounts: 0, 300, 600, and 700.

Solution

With the amount of snowfall as X and the amount paid under the policy as Y, we have

y	$f_Y(y) = P(Y = y)$
0	$P(Y = 0) = P(0 \le X < 50) = 0.72$
300	$P(Y = 300) = P(50 \le X < 60) = 0.14$
600	$P(Y = 600) = P(60 \le X < 70) = 0.06$
700	$P(Y = 700) = P(Y \ge 70) = 0.08$

The standard deviation of Y is $\sqrt{E(Y^2) - [E(Y)]^2}$.

$$E(Y) = 0.14 \times 300 + 0.06 \times 600 + 0.08 \times 700 = 134$$

$$E(Y^2) = 0.14 \times 300^2 + 0.06 \times 600^2 + 0.08 \times 700^2 = 73400$$

$$\sqrt{E(Y^2) - [E(Y)]^2} = \sqrt{73400 - 134^2} = 235.465$$

Common Questions & Errors

Students shouldn't overthink the problem with fractional payments of 300. Also, account for probabilities in which payment cap of 700 is reached.

In these problems, we must distinguish between the REALT RV (how much snow falls) and the PAYMENT RV (when does the insurer pay)?. The problem states "The insurer pays the airport 300 for every full ten inches of snow in excess of 40 inches, up to a policy maximum of 700 ." So the insurer will not start paying UNTIL AFTER 10 full inches in excess of 40 inches of snow is reached (say at 50+ or 51). In other words, the insurer will pay nothing if X<50.

Rate this problem | Excellent | Needs Improvement | Inadequate

Quickly access the Hub for additional learning.

Flag problems for review, record notes, and get instructor support.

View difficulty level.

Helpful strategies to get you started.

Full solutions with detailed explanations to deepen your understanding.

Commonly encountered errors.

Rate a problem or give feedback.

Thank You for Choosing ACTEX Learning!

We're committed to helping you succeed on your actuarial journey.

For the latest study guides, textbooks, free Formula Sheets, and more resources for SOA, CAS, IFoA, and IAI exams, visit:

ACTEX Learning https://actexlearning.com/

Your destination for comprehensive actuarial exam preparation and professional development.

Looking for additional study material or other actuarial books?

THE ACTUARIAL BOOKSTORE https://www.actuarialbookstore.com/

The #1 online source for actuarial books and study guides.

Scan to Learn More