Having mastered the predictive models required by the exam syllabus, in the last component of this study manual we will get you fully ready for the PA exam. We will first walk you through how the PA exam overall is like. Then we will introduce a simple framework that can help you write an effective and substantive executive summary, the very last part of an exam project. We will end with a detailed discussion on past and sample PA projects released by the SOA.

General Structure of a PA Project

The current task-based exam format has only been in place since the June 2019 PA exam and is radically different from the open-ended format of the December 2018 PA exam. According to page 8 of the exam syllabus,[xi]

> "the December 2019 exam will be structured in a manner similar to this sample project [Hospital Readmissions] and to the June 2019 exam."

In the new exam format, you are presented with a series of well-defined tasks intended to be done in order, with results from one task informing work in later tasks. Specific guidance is provided in each task to ensure that you stay on track. Almost always, the tasks center on the same business problem with the goal of *identifying and interpreting the key factors* affecting something (the target variable) the client you work for is interested in. These tasks can be grouped as follows (tasks of types 2 and 3 may be intertwined):

Type 1: Data exploration, transformations, and cleaning. In the warm-up part of an exam project, you are often asked to perform exploratory data analysis to familiarize yourself with the dataset. With the aid of graphical displays and summary statistics, you may do univariate data exploration and look at each variable in turn, making variable transformations if necessary, or do bivariate data exploration and have a feel for the relationship between the target variable and other variables in the dataset. You may also be asked to use advanced data exploration tools like PCA and cluster analysis to generate new features (the PCs or the group assignments). If the project centers on GLMs, then a common task is to use graphical aids to look for variables that are likely to interact and create appropriate interaction variables. Overall, the objective of the first part of the project is to let you gain a good understanding of the variables in the dataset, form preliminary conclusions regarding which variables are predictive of the target variable, and create potentially useful features for prediction.

Type 2: Model construction, evaluation, selection, and validation. Drawing upon your understanding of the data in the warm-up part, you will be asked to do some "real stuff" in the main body of the project, concerned with constructing, evaluating, selecting, and validating predictive models (GLMs and/or decision trees). Besides fitting models in R, you are also expected to describe what you are doing, at a conceptual level, and show your thought process clearly in the Word report template. This is the reason why each of Chapters 3 to 6 of this manual begins with a conceptual foundation section. Remember:

This exam is much more about *communication* than about coding.

[xi] The June 2020 exam syllabus is not yet available when this edition of the manual goes in press.

In this part, it is fine to include technical terms as the intended audience is the examination grading team, not the client.

In addition to the report template, the exam will provide you with an Rmd file that contains some code supplied by your virtual "assistant." The code can help you tackle the tasks in both Parts 1 and 2 of the project tremendously; there is no need to (and you shouldn't!) start from scratch. In most cases, the supplied code is technically correct, meaning that it can be run in R without errors, but you are not expected to leave the code intact. Instead, adapt the code to suit your purposes (that is how the SOA tests you!). For example:

- If you have done any transformations (e.g., log) on your variables, be sure to modify the code to reflect the transformations.
- If you have created any new features (e.g., interaction variables), they should be inserted into the model formula (mainly for GLMs).
- Some variables may be treated technically in R as a numeric variable, but it may be more desirable to convert them to a factor.
- If you have to fit a GLM, you may need to modify the `family` argument so that the right target distribution and link function are used.
- If you have to fit a decision tree, you may need to modify the control parameters so that the right level of tree complexity is reached.
- Make sure that your models are fitted to the training set (check the `data` argument of the `glm()`, `rpart()`, and `train()` functions) and predictions are generated on the test set (check the `newdata` and `newx` arguments of the `predict()` function). Changes along this line are required for the December 2018 PA exam and the Student Success sample project.

Type 3: Considering alternatives. An exam project cannot test all of the predictive analytic techniques covered in the exam syllabus, so to make the project more comprehensive in scope there is often a descriptive task (e.g., the second-to-last task in the June 2019 exam) that requires you to explain, at a conceptual level, the pros and cons of alternative modeling techniques that could be used. No additional fitting is required. This is a task which you can prepare for well in advance using the conceptual foundation sections of this study manual, demonstrate your knowledge of the exam material to the grader, and get some easy points.

Type 4: Executive summary (20 points). The last task is almost always writing an executive summary intended to be read by your client, who is someone not familiar with predictive analytics and has nothing else to refer to. We will defer a detailed discussion of this part to the next section.

At the end of the exam, you will submit both the completed Word report template and the Rmd file that contains your code for grading. If you feel the need, you may also submit Excel files (.xlsx or .csv), but there is no expectation that you do so.

Structure of Executive Summary

Up to this point, you should be sufficiently well prepared to handle exam tasks of types 1, 2, and 3 based on what you learned in preceding chapters of this manual. The only part that we have not

directly discussed is the executive summary, which carries the greatest number of points among all the tasks and is notoriously difficult to write. The difficulty comes from the need to summarize the technical analysis you have performed using concise, non-technical language easily comprehensible to the client as well as to decide how much information to include. The key is to present only the essentials of your analysis which a high-level audience would need to know. In short, show the big picture without getting bogged down in unnecessary details.

To make your writing as comfortable as possible, in what follows we will provide a four-part framework for you to put together the executive summary systematically. While there is no page limit, in general a good executive summary should be about 2 to 2.5 pages long. One that is too short means that you are missing important points and one that is excessively long is overwhelming clients with unimportant details.

Part 1: Introduction. The introductory paragraph of the executive summary should provide the background of the business problem the client would like you to solve using predictive analytics and describe the objective of the data-driven analysis you have performed. To this end, feel free to paraphrase the information given on the first page of the project statement.

Example from June 2019 PA exam

"This report provides a preliminary investigation into the **factors that contribute to greater or lesser severity of automobile accidents**. We have used the 2014-2019 Cary, NC data that you supplied. The severity of an individual accident is represented by Crash Score, a variable you have created that combines information from each accident such as number of injuries and number of vehicles involved. You also supplied variables that you believe relate to Crash Score. As requested, we have **applied analytic techniques to determine which of those variables relate to Crash Score** and the **magnitude of the effect** for those that do."

Part 2: Data description. The second part of the executive summary should give a high-level description of the data underlying your analysis, including the data source, its reliability, limitations (if any), the key characteristics of the variables, and the major data cleaning and alteration steps you performed. These items are explained as follows.

1. *Data source:* We can begin by explicitly stating where the dataset you used is from, including the time period or geographical areas over which the observations were collected. In many cases, the dataset is provided directly by the client you work for.

Example from June 2019 PA exam

"The data you supplied came from **23,127** accidents in the town of Cary, NC, **from 2014 through mid-2019**."

2. *Reliability and limitations:* Comment on the reliability of the dataset (usually, it is *largely* reliable, or else there is no point in using it to do predictive analytics!). Are there any missing values or obvious recording errors? If you have taken any data cleaning steps such as removing outliers or missing values, be sure to make a mention. If possible, do point out any potential limitations in the data. If the dataset was collected only in a restricted time period

or geographical location, then the findings of your analysis may be rather local and may not generalize well to other cases.

Example from June 2019 PA exam

"The supplied data had no **missing values** or **obvious errors**. We have therefore assumed that it is **reliable** for the intended purpose. Because the data is from a **single locality**, the findings of this analysis may **not generalize well** to crash severity for the whole state."

3. *Key variables:* Give a general description of what types of variables (e.g., demographic factors of patients/customers, characteristics of XYZ) are in the data. Then point out the characteristics of the key variables, with particular attention paid to the target variable. Your description can revolve around the following aspects:

Example from June 2019 PA exam

"The supplied variables appear to be a **good cross-section of prevailing conditions at the time of the accident**."

Example from Dec 2018 PA exam

"The models were built using a dataset from the U.S. Mine Safety and Health Administration (MSHA) from 2013 to 2016, **including variables for year, state, mine characteristics, employee count, employee hours, employee activities, and injuries reported**."

- *Statistical nature:* You can describe the statistical characteristics (e.g., are they continuous with a prominent skew or categorical with various levels?) of the distribution of the key variables. If it helps, feel free to include tables and graphics to visualize the shape of the distribution. The June 2019 PA exam, for example, includes a histogram of the target variable to exhibit its right skewness.
- *Are they appropriate to serve as predictors?* If some of the variables capture sensitive information such as gender, ethnicity, age, and religion, do make a note that they may be inappropriate predictors that can raise discriminatory or legal concerns, an issue that should be discussed with the client before the final predictive model is put to use.

Example from the Hospital Readmissions Sample Project

"The supplied data includes **race** and **age**. Given that this model might be used to decide which patients receive additional support, it is possible that these variables would be considered **inappropriate**. Age appears in the recommended model and this should be **reviewed** prior to implementation."

- *Are the predictor variables within the client's control?* In some projects, the client is interested in using our predictive analytic results to drive actions. It will be useful to point out which predictors are variables that the client can actively change and which are not.

Example from June 2019 PA exam

"The supplied variables appear to be a good cross-section of prevailing conditions at the time of the accident. Some variables may prove relevant but are **not within NC DOT's control**. These are..."

4. *Data alterations:* Since the dataset is supplied by the client, they have the right (and they should care) to know if any alterations were made to the variables, for otherwise the client may misinterpret the results of the model. The June 2019 PA exam model solution says that the "executive summary need *not list every alternation* made, but some indication of the nature of the alternations or an example should be provided." Examples of material alterations that should be disclosed include:

   - Combinations of factor levels to bring out the effects of the categorical variables on the target variable and reduce their dimensions.
   - The use of PCA to replace original variables with one or more PCs.
   - The use of cluster analysis to replace original variables with the variable of group assignments.
   - Any important actions as a result of the data exploration tasks in the early part of the project that have a major impact on the final predictive model.

Part 3: Model evaluation, selection, validation, and interpretation. This is the most important part of the executive summary where you describe how you perform model construction, feature generation (you can also describe feature generation in Part 2 if it fits better), model/feature selection, model validation, and interpretation, all in *non-technical, laymen* terms. As the client is likely not someone familiar with predictive analytics, the choice of the language is important, making this part of the executive summary especially difficult to write. The challenge lies in presenting the essentials without making overly crude statements or swamping the client with unnecessary technical details.

- *Type of predictive model used and why:* In the main body of the executive summary, we can begin by saying which type of predictive model (GLMs and/or decision trees) is used to address the business problem stated in the introduction. Provide a high-level description of the model, its deliverable, and its merits (and demerits), using terms that can be understood by the client. For this purpose, the conceptual foundation sections in Chapters 4 and 5 are useful.

  ▷ *GLMs:* As we learned in Chapter 4, a GLM relates a function of the mean of the target variable linearly to a set of predictors. The target variable itself is not transformed.

  From a predictive point of view, a GLM excels in accommodating a wide variety of distributions for the target variable being predicted, whether it be a continuous variable with a symmetric distribution (e.g., normal) or with a prominent skew (e.g., gamma), or a discrete variable taking positive integer values (e.g., Poisson) or only two possible values (e.g., binomial); here you should make reference to the nature of the target variable in the exam project. With such flexibility, we are at liberty to choose a distribution that best aligns with the characteristics of the target variable. Moreover, the function relating the target mean and the predictors can be readily selected to ensure that the predictions

generated are always appropriate (again, refer to the target variable). The deliverable is an analytic equation that clearly shows how the model predictions depend on the features, with the coefficients of the equation providing a readily interpretable measure of the directional effect each feature exerts on the target variable.

On the downside, GLMs are unable to capture nonlinear relationships unless they are manually factored in the model in advance. More generally, they highly depend on the form the features enter the model and are prone to model mis-specification issues. (It is a bit debatable as to whether to mention the drawbacks of the predictive model you use as this may undermine the value of the analysis you perform to the client. My suggestion is to briefly mention their downsides, but don't overdo it.)

EXAM NOTE

In the description above, we have deliberately avoided the use of technical terms such as "exponential family" or "link function" to maximize its readability to the audience. It may be fine to include some technical terms if their inclusion really adds value, but they must be explained to the client.

Example from June 2019 PA exam

"With the data ready for analysis, we have selected a generalized linear model (GLM) for drawing inferences. A GLM has the advantages of being able to **accommodate a variety of distributions** for the quantity being predicted (hence can account for the skewness seen in the graph presented earlier), can ensure that **predictions are always appropriate** (positive numbers in this case), provides a systematic method for deciding which factors have predictive power, and, for those that do, provides an **easily interpretable measure** of the effect each factor has on Crash Score."

▷ *Decision trees:* Decision trees, which are covered in Chapter 5, are predictive models that provide an alternative method for linking a target variable to predictors. They work by partitioning the set of values of the predictor variables into a set of mutually exclusive and exhaustive regions where observations are more homogeneous and thus more amenable to prediction. The deliverable is a series of if/then classification rules which are based on the values or levels of the features and clearly indicate the key predictors and their interactions. To predict an observation, use the classification rules to assign the observation to the appropriate region, where observations share the same predicted value or class.

Decision trees have the notable advantages of being easy to interpret due to the transparent nature of the if/then rules. Looking at a decision tree, we can easily tell which predictors are the most important, i.e., those that appear in top splits. Interactions between variables are automatically handled as well.

By far the greatest drawback of decision trees is their instability in the sense that a different set of data can give rise to a rather different fitted tree and a different set of predictions. This problem, though, can be alleviated by the use of ensemble methods.

EXAM NOTE

In the executive summary, it is a good idea not to say "decision trees are easy to explain to *non-technical audience*." Are you implying that the client is a layman? ☻ Also note that we deliberately avoid using technical terms like "overfitting" or "high-variance predictions," but explain their meaning in non-technical language.

Example from December 2018 PA exam

"Decision trees are models with a simple, easy-to-interpret structure based on a set of if/then rules that clearly highlight key factors and interactions."

- *Feature selection process:* Every PA project will involve feature selection in some way, whether it is for a GLM (using stepwise selection or regularization) or a decision tree (using pruning). When describing the feature selection process, again refrain from technical terms such as bias/variance/overfitting/AIC/BIC, but provide a general explanation of why features should be removed:

  > To simplify the model and retain only variables with predictive power. This way, we make the predictive model easier to interpret and improve prediction accuracy.

  If you perform feature selection for a GLM using the BIC, for instance, you can justify your selection criterion as follows:

Example from June 2019 PA exam

"After selecting a GLM that aligns with the data, the next step is to use a statistical method to **simplify the model** by **selecting only those factors that have predictive power**. Commonly used statistical methods allow some leeway in making this selection. We have selected **a method that tends to remove more variables**. This will allow NC DOT to **concentrate** on the ones that have the **greatest relationship** to Crash Score."

Notice that the SOA did not spell out the selection criterion (BIC in this case), but skillfully described the reasons for using this criterion in plain language, which are more of interest to the client.

If you reduce the complexity of a decision tree by pruning, you can say:

> After fitting a large, complex decision tree to the data, the next step is to control its complexity and prune back branches that add little predictive power so that the tree is of reasonable size for interpretation and the client can concentrate on the predictors with the strongest relationship to the target variable. The pruning starts at the bottom of the tree to avoid missing out on good splits that are preceded by mediocre splits.

Regardless of how you select the features, do point out which features are retained at the end of the selection process as the client is definitely interested in knowing the key factors affecting the target variable.

Example from June 2019 PA exam

"As a result, the following variables were retained.

- ▷ Road Class
- ▷ Road Feature
- ▷ Time of Day
- ▷ Traffic Control"

- *Model validation:* This is where we justify the model we select and convince the client that it is a "good" model capable of addressing the client's needs. To explain the model validation process, we can start by mentioning that the selected model is evaluated by applying it to a set of unseen (test) data and assessing its predictive performance there. It is a good idea to explain the need for using an independent set of data: Evaluating the model on the set of data where it is fitted (trained) will give an overly optimistic picture of its true predictive performance. If there is an existing model which your model is expected to beat, point out how your model outperforms the existing model with respect to the performance metric(s) you use.

  If you performed any model diagnostics and generated any diagnostic plots in an earlier task (for GLMs only), describe whether the diagnostic analysis supports the model assumptions (hopefully it does!) without mentioning specific terms like "`Residuals vs Fitted`" or "`Normal Q-Q`" plots.

Example from June 2019 PA exam

"Various tests, both **numerical** and **graphical**, were conducted to ensure that the model selected was **appropriate**. In particular, we verified that removing the other variables did not reduce predictive power and that the **various assumptions underlying use of a GLM were reasonably satisfied**."

- *Interpretation:* We can end Part 3 of the executive summary by interpreting the output of the model in plain language. For GLMs, the three-part structure introduced on page 251 is useful:

  1. Translate the coefficient estimates into precise statements on the effect of each feature on the target variable.
  2. Explain, using common sense, whether the statements agree with our prior knowledge.
  3. Describe what we can learn from these statements taken collectively considering the business context. In other words, go back to the "big picture."

Example from June 2019 PA exam

The GLM coefficients tell us the **effect** of various levels of the selected factors on Crash Score. Due to the nature of the model, the coefficients can be transformed to **percentage changes**.

Feature	Interpretation	Commentary
Road Class = OTHER	8% decrease in Crash Score compared to State Highway	Drivers may go faster on U.S. highways versus State highways, leading to more severe crashes. Similarly, drivers on other roads may drive slower.
Road Class = US HWY	2% increase in Crash Score compared to State Highway	
⋮	⋮	⋮

For base decision trees, you can address the following items:

1. Which variables are the most informative? They are those that appear in the first few splits.
2. Which variables appear to have interactions?
3. Do the findings make sense?

When making statements based on the coefficient estimates, we should refer to the features using their real name (e.g., operational time), not the coded name in R (e.g., `op_time`).

There is no need to copy any R output here as the client, likely someone not familiar with predictive analytics, is probably unable to make sense of the output without additional explanations. However, using visual aids like a table to present your results will help.

Part 4: Concluding paragraph. The last part of the executive summary is a short paragraph summarizing the key findings of your data-driven analysis and restating their significance in a positive light. You can take the opportunity to discuss the business implications of your work and how your analysis supports the original broad goal of the client, e.g.,

> Our analysis has indicated the key drivers of crash severity. The results have practical implications for NC DOT for designing safer roads in the future to reduce crash severity.

It is also a good idea to suggest possible ways the analysis can be improved (e.g., possibly useful variables that are currently missing in the dataset), possible next steps, and areas for future analysis.

Example from June 2019 PA exam

"Our investigation into crashes in Cary, NC has **indicated factors** that lead to changes in the expected severity. We note that the changes are not particularly large but may help guide NC DOT in making changes to road configurations. Should you find this report useful, we would be pleased to **analyze a set of state-wide data**. We would also like to more closely analyze the relationship between Road Feature and Traffic Control as both are related to intersections. Perhaps there is additional data that could help us better understand this relationship.."

Past/Sample Project	Data Cleaning/Exploration Issues	Predictive Analytic Techniques
June 2019 Exam PA	Bivariate exploration, combining factor levels, interaction	GLMs (OLS, gamma regression), regularization, PCA
Hospital Readmissions	Univariate exploration, combining factor variables, interaction	GLMs (logistic/probit regression), cluster analysis
Dec 2018 Exam PA	Univariate exploration, bivariate exploration, removing problematic observations, missing data, collinearity	GLMs (Poisson regression with offset), decision trees (base trees only)
Student Success	Univariate exploration, bivariate exploration, removing problematic observations	GLMs (logistic regression), decision trees (base trees and random forests)

Table 6.2: Summary of the four released PA projects.

You may be tempted to acknowledge alternative predictive models that could be tried in an attempt to demonstrate your predictive analytics expertise. While discussing alternative methods is technically fine, it may sound unprofessional to the client. The June 2019 model solution makes it clear that:

> "It is *not appropriate* to discuss *alternative modeling approaches* that might be tried *given more time*, such as random forest. The client expected you to build the best possible model with the available time and data *(so bossy!)*."

If you feel the need to acknowledge modeling alternatives, you can do so towards the end of Part 3 of the executive summary, but do accompany your description by a brief explanation of why these methods are not attempted.

Comments on Released PA Projects

The SOA has released a total of four full-length projects. Table 6.2 summarizes what each project tests. Note that only the June 2019 PA exam and the Hospital Readmissions sample project follow the current exam format. To access these four projects:

- *June 2019 and Dec 2018 exams:* `www.soa.org/education/exam-req/syllabus-study-materials/edu-multiple-choice-exam/`.
- *The two sample projects:* See page 8 of the exam syllabus, `www.soa.org/globalassets/assets/files/edu/2019/2019-12-exam-pa-syllabus.pdf`.

June 2019 Exam PA

This is the first PA exam project following the new exam format. There are a total of 11 tasks. The client is the North Carolina Department of Transportation (NC DOT) and your actuarial

consulting firm is tasked with identifying key factors affecting `Crash_Score`, a positive, continuous, right-skewed variable measuring crash severity, based on a collection of variables that relate to road conditions. The main predictive models tested are GLMs and regularized regression. In the commentary below, both content-related and coding-related comments are given.

The June exam was given over a two-day testing window (June 13 and June 14). The datasets were different, but the tasks were the same. For simplicity, we will consider the June 13 exam.

Task 1 – Explore the relationship of each variable to `Crash_Score` *(5 points)*

- *Content:* In this warm-up task, you are asked to perform bivariate data exploration (covered in Section 2.2 of this manual), investigating the relationship between each variable and the target variable `Crash_Score` as a way to identify potentially useful predictors. Task 1 of Section 3.4 and Task 3 of Section 4.3 are in a similar vein. You are specifically required to use *both* graphical displays (split box plots, because `Crash_Score` is numeric and the variables are factors) and summary statistics (the mean and median of `Crash_Score` grouped by different levels of the factor variables), so for full credits be sure to apply both kinds of tools.

  Although not explicitly requested by the task statement, exploring the distribution of `Crash_Score` is expected by the SOA, presumably because the target variable is the most important variable in the data, but many candidates neglected to do so; the same thing happened in the December 2018 PA exam. You can look at the summary statistics of `Crash_Score` like the mean and median, which are already produced when you run the chunk loading the data, and observe that the mean is greater than the median and the maximum is far greater, suggesting the presence of a right skew. The right skew is easily confirmed by a histogram and can be dealt with by a log transformation. The log-transformed `Crash_Score` is then used throughout the task.

  EXAM NOTE

  In general, you are encouraged to copy R output or graphs from RStudio to your report template as long as they are well connected with your write-up. The model solution for Task 1, for example, includes a histogram for `Crash_Score` and the split box plot for one of the variables (`Rd_Feature`) for illustration.

  When you look at the box plots for `Crash_Score` or its log-transformed version, you can see that there is not much of a difference in its median across different levels of the factor variables. Fortunately, looking at the precise values of its means and medians can help a bit. Because the variables are factors with multiple levels, we will not say whether they increase or decrease with the target variable. Rather, in your write-up describe whether `Crash_Score` is particularly high or low in certain levels of the factors and spell out those levels. Of course, whether a value is high or low can be somewhat subjective, so different candidates will come up with slightly different factor levels that appear to impact on `Crash_Score` noticeably. As the model solution says, "[b]ecause there is no clear boundary between appearing to be predictive and not, it was not necessary for candidates to reproduce the same list as presented here."

- *Coding:* In the Rmd template, you are directly provided with code using a `for` loop to produce a box plot for `Crash_Score` split by different levels of each factor variable and using the `dplyr`

package to calculate the group means and medians. All you have to do is change `Crash_Score` to `log(Crash_Score)` and run the code. You will need to write some simple code using the `geom_histogram()` function to generate a histogram for `Crash_Score`, which shouldn't be a difficult task (see Section 2.2).

Task 2 – Reduce the number of factor levels where appropriate *(5 points)*

- *Content:* This data exploration task requests that you combine some of the sparse factor levels into factor levels with more observations. Task 4 in Section 4.3 provides further practice.

  The model solution goes straight to describing the combinations for five factors, `Time_of_Day`, `Rd_Feature`, `Rd_Character`, `Rd_Surface`, and `Traffic_Control`. To add more substance to your written response, you can explain the rationale for combining levels prior to making the combinations, namely, to strike a balance between the following conflicting goals:

  1. To ensure that each level has a sufficient number of observations to improve the reliability of the model fitting procedure.
  2. To preserve the differences in the mean/median of the target variable among different factor levels. Such differences may be useful for prediction.

  As the model solution says, "[i]t was not sufficient to only combine those levels with extremely low counts." For example, although there are only 808 observations with `Time_of_Day = 1`, the mean (or median) of `log(Crash_Score)` there is sufficiently different that it is better for us to keep the level as is. You may also notice that some factors such as `year` and `Rd_Configuration` have a small number of observations in some of their levels (e.g., `2019` and `UNKNOWN`), yet the SOA did not combine those levels. This is because the mean of `log(Crash_Score)` is more or less the same among different levels of such factors, which are probably not useful predictors for `log(Crash_Score)`. That is why the task specifically asks that you "[c]onsider using knowledge of the factor levels as well as *evidence from Task 1*," where you identified potentially useful predictors, to make the combinations.

  Because different students have different perceptions of what a high or low mean level is and what number of observations is small (e.g., is 200 small? What about 1,000?), this task will attract a wide range of answers. According to the model solution, "[t]he number of variables for which factors were combined was less important than the *quality of the combinations* made and the *supporting evidence*." Nevertheless, for full credits you are expected to find "*multiple* cases where factor levels could be combined." To play safe, you should describe at least three to four variables whose levels will be grouped.

- *Coding:* The Rmd template provides code for making bar charts showing the number of observations at each factor level. In my opinion, it is much more convenient to look at the output in Task 1 that shows the mean, median, and the number of observations for `log(Crash_Score)` split by different factor levels. As discussed above, both the number of observations and the similarity of the means/medians should be taken into account when combining the factor levels.

  You are also given code for combining factor levels and you will have to do a fair amount of manual changes. For each variable you choose to work on, assign it to the `var` variable and define the new levels in the third argument of the `mapvalues()` function. It is a good idea to use informative names for the new levels such as `"OVERNIGHT"`, `"LATE-EARLY"`, `"DAYTIME"`

rather than "1", "2", "3" (this is where the knowledge of the factor levels may help). To make it easy for the grader to check your work, it is suggested that you make one chunk for each factor variable. After running each chunk, you should check that the desired changes have been correctly made.

Before you run the code, you should make sure that the variable you are dealing with is indeed a factor, as warned by the Rmd template. For example, `Time_of_Day` is originally an integer variable, so it needs to be converted to a factor via the `as.factor()` function. Failure to make the factor conversion will result in an error.

Task 3 – Use observations from principal components analysis (PCA) to generate a new feature ***(9 points)***

- *Content:* In this task you are asked to run a PCA on the three weather-related variables `Rd_Conditions`, `Light`, and `Weather`, interpret the output, and use the results to generate a new feature. These are all illustrated in the case study in Subsection 6.1.2.

  Before starting to interpret the output from the PCA, you could have briefly described, in one or two sentences, what PCA is. A succinct description like the one below can be useful (feel free to refine it):

  > Principal components analysis (PCA) is an advanced data-exploration technique that serves to summarize and reduce the dimension of a dataset by means of a set of linear combinations of the original features, known as principal components. These new features are generated to capture as much information from the dataset (with respect to the variance) as possible.

  Then you can proceed to describe the (cumulative) proportion of variance explained by the first few PCs (which is rather low) and the properties of the PC loadings, paying attention to their sign and magnitude. Then the new feature is created from the largest (in absolute value) loadings of the first PC, as discussed in Method 2 on page 356.

- *Coding:* Because `Rd_Conditions`, `Light`, and `Weather` are factor variables and the `prcomp()` function cannot handle factor variables, code is provided to explicitly binarize the three variables using the `dummyVars()` function in the `caret` package (as we learned in Section 3.4, the conversion of the factors to characters is not really necessary). Unlike the case for GLMs, the `fullRank` argument is set to `FALSE` so that all factor levels receive a loading and no baseline levels are left out. The binarization creates a new data frame, called `datPCAbin`, of $4 + 6 + 5 = 15$ dummy variables, only 12 of which are linearly independent. PCA is then run on this data frame and the first 12 PCs together are able to explain all of the variation in the data.

  When you create the new feature based on the dominant loadings of the first PC, make sure that the variables have been scaled, in case the given chunk has not performed standardization. Instead of manually typing the four loadings -0.51, 0.5, -0.46, and 0.43, you can replace them by the more accurate values in the `rotation` matrix. That is, you can use the following command:

```
phi <- PCAweather$rotation
```

```
dat2$WETorDRY <- phi["Rd_ConditionsDRY", 1] * datPCAbin.std$Rd_ConditionsDRY +
  phi["Rd_ConditionsWET", 1] * datPCAbin.std$Rd_ConditionsWET +
  phi["WeatherCLEAR", 1] * datPCAbin.std$WeatherCLEAR +
  phi["WeatherRAIN", 1] * datPCAbin.std$WeatherRAIN
```

More importantly, the two factors, `Rd_Conditions` and `Weather`, that play a role in the new feature should be dropped from the `dat` data frame at the end. If the new feature is added but the original two variables are retained, then collinearity will arise and a rank-deficient model will be produced when we fit a GLM later.

Task 4 – Select an interaction *(7 points)*

- *Content:* In the last data exploration task, you are asked to generate an interaction variable that can be used when constructing a GLM in subsequent tasks. The detection of interaction and creation of interaction variables have been covered in the case studies in Sections 3.4 and 4.3. As discussed there, the following structure is useful for tackling this kind of task related to interactions:

  ▷ Describe the meaning of an interaction between variables to signal to the grader what you are looking for.

  ▷ Propose variables that intuitively are likely to interact based on common sense.

  ▷ Use graphical displays to confirm or dispel the conjectured interaction.

  In this exam, the interaction effect between variables appears rather inconspicuous. It turns out that the interaction variable will be eliminated when we perform stepwise selection in a later task (Task 6).

- *Coding:* Code is given to produce a box plot for `Crash_Score` split by the levels of one factor and faceted by another factor. All we have to do is replace `Crash_Score` by `log(Crash_Score)` and the two factors by the factors we are interested in.

Task 5 – Select a distribution and link function *(10 points)*

- *Content:* Tasks 5 to 8 are the meat of this exam project, where you construct, select, and validate a GLM for `Crash_Score`. In Task 5, you are asked to specify the two components of a GLM: The distribution of the target variable and the link function. These two components are well covered in Section 4.1. Here we should select a target distribution that aligns with the characteristics of `Crash_Score`, which is positive, continuous with a right skew, as we have seen in Task 1. The gamma and inverse Gaussian distributions both serve this purpose well. The link function should ensure that the predictions are appropriate (positive in this case) and make the model output easily interpretable. The log link is arguably the best choice.

  You are specifically required to explore two pairs of target distribution and link function. Your response can be structured as follows:

  ▷ Begin by explaining the two pairs of target distribution and link function you choose with reference to `Crash_Score`.

▷ Point out that you will evaluate a candidate GLM using a training/test split. Describe the properties of the training and test sets (e.g., the proportion of observations that go to the two sets) and explicitly check, using summary statistics like the mean, that the built-in stratification works well. The model solution does not say so, but feel free to briefly explain why the use of a training/test split is necessary.

▷ Justify the model you select based on its performance on the test set. Because `Crash_Score` is a numeric variable, a metric like the RMSE is applicable. A general model selection criterion like AIC can also be used.

For the June 13 exam, it turns out that the test RMSE of the gamma GLM with a log link is even slightly higher than that of the OLS linear model. For a highly skewed variable like `Crash_Score`, the RMSE is less effective as a metric because the presence of a few extremely large observations in the test set can disproportionately distort its value. A likelihood-based metric like AIC is more reliable.

There is a subtlety in this question: SOA expects that you consider the two time-related variables `year` and `Month` no later than this task because model fitting starts here. For modeling purposes, `year` can be retained as a numeric (integer) variable to account for potential trend effects. `Month`, however, should better be converted into a factor because any monthly effect is probably seasonal in nature (see also Task 1 in Section 4.2). We have seen why a factor conversion makes sense when we dealt with the variables `inj`, `age_cat`, and `veh_age` in Sections 4.2 and 4.3. If you decide to drop `year` and `Month` (not preferred), you should justify your decision with reference to the graphs and summary statistics in Task 1, which show that `year` and `Month` are non-predictive variables.

- *Coding:* You are directly provided with a code chunk to do the training/test split using the usual `createDataPartition()` function from the `caret` package; no modifications to this chunk are needed. Code is also given to fit an OLS linear model for `Crash_Score` on all other variables using the `glm()` function and to calculate its test RMSE. Here are some points to watch out:

  ▷ Before you fit any model, make sure that you have converted `Month` to a factor with the command `dat$Month <- as.factor(dat$Month)`.

  ▷ This part involves fitting several GLMs. It is a good idea to use a descriptive model name for each new model created (e.g., `GLMols`, `GLMgamma`, `GLMig`) to improve readability and reduce the chance of using incorrect summary statistics.

  ▷ When you modify the code for fitting the gamma GLM and inverse Gaussian GLM, you should not only change the `family` argument of the `glm()` function, but also insert the interaction variable identified in Task 4 in the `formula` argument.

  In this project, the model fitting algorithm for the inverse Gaussian GLM fails to converge. You should still include the code for fitting the model so that the grader knows that you have done the real work, but comment it out so that the entire Rmd file can be run without error. Remember that "[g]raders expect that your Rmd code can be run from beginning to end," as stated on the first page of the project statement.

Task 6 – Select features using AIC or BIC ***(12 points)***

- *Content:* This task requires that you perform stepwise selection, and describe and justify the selection criterion (AIC or BIC) and selection process (forward or backward selection) you use. Here what you learned in Subsections 3.2.4 and 4.1.2 is very useful.

  According to the model solution, "[a]ny combination of criterion and selection process is valid *if justified appropriately.*" The solution selects the BIC and forward selection since they tend to produce a final model with fewer features so that the client (NC DOT) can concentrate on the key factors affecting crash severity. There is also an important exam-based motivation for choosing a criterion and selection process that tend to favor simpler models: You will have an easier time interpreting the final model in a later task. Of course, you will not spell out this reason to the grader! ☻

  Note that the four retained features, `Rd_Class`, `Rd_Feature`, `Time_of_Day`, and `Traffic_Control`, were identified in Task 1 as variables that may be predictive of `Crash_Score`. This shows that the bivariate data exploration performed there has some value.

- *Coding:* You are given code to run a forward selection based on the BIC for the OLS linear model using the `stepAIC()` function. Two changes are needed:

  1. To change the distribution and link function from normal and the identity link respectively to gamma and the log link, which we recommended in Task 5.
  2. Change the `upper` and `lower` models in the `scope` argument of the `stepAIC()` function respectively to the full model fitted in Task 5 and the null model (i.e., intercept-only model) with the gamma distribution and the log link. This will automatically insert the interaction variable into the selection process.

  The model solution includes a separate chunk for manually fitting the final model based on `Rd_Class`, `Rd_Feature`, `Time_of_Day`, and `Traffic_Control`; this is the list of retained variables taken from the output of the `stepAIC()` function. To save effort and avoid making mistakes, we can simply assign the final model returned by the `stepAIC()` function to an object. That is, you can use the following command to save the final model easily:

```
GLMgammaR <- stepAIC(GLMgamma1, direction = "forward", k = log(nrow(train)),
                     scope = list(upper = GLMgamma, lower = GLMgamma1))
```

Task 7 – Validate the model *(9 points)*

- *Content:* In this 6-point, short task you are asked to validate the model selected in Task 6 and show that it outperforms the assistant's OLS model (as well as the full model) with respect to the AIC and the test RMSE despite using far fewer features and thus using much less information. You are also required to diagnose the final model and see if the model assumptions are largely satisfied, judging by the behavior of the *standardized deviance residuals*. If the fitted GLM is appropriate, then the points in the "`Residuals vs Fitted`" graph should scatter around zero with more or less the same amount of variation and no special patterns, and those in the "`Normal Q-Q`" plot should lie on the reference line closely. The normality is mostly satisfied, except for the two tails of the distribution, which is not uncommon for a GLM.

  The model solution says that

> "[s]ome candidates did not understand that the Q-Q plot checks the normality of the *standardized deviance residuals*. If an appropriate model is being used, this should be the case regardless of the distribution used in the model."

Subsection 4.1.2 could have helped these candidates dispel their misconceptions.

- *Coding:* No code is provided for this task, because you can "recycle" the code in Task 5 for calculating the AIC and test RMSE of the final model. Diagnostic plots can also be easily generated by applying the `plot()` function to the final model.

Task 8 – Interpret the model *(9 points)*

- *Content:* Now that the final model is selected, it is time to run it on the full dataset to make it more robust and interpret its output. This is illustrated in the last task of both Sections 4.2 and 4.3.

  Because it helps with your description, it is a good idea to paste the coefficient estimates table of the model summary to your report in this part. The model solution looks at the *percentage* changes (i.e., $e^{\hat{\beta}_j} - 1$) in `Crash_Score` associated with different classes of different factors. It is perfectly fine if you state the findings equivalently in terms of *multiplicative* changes (i.e., the target variable is multiplied by $e^{\hat{\beta}_j}$). You can conclude by saying that overall your findings can inform NC DOT when it comes to designing roads in the future to reduce crash severity. As always, try to tie your results to the business problem, something that truly can impress graders.

- *Coding:* Again, no code is provided for this task. You merely have to fit the final model to the full dataset by changing the `data` argument of the `glm()` function from `train` (training set) to `dat`. To get the percentage change $e^{\hat{\beta}_j} - 1$ in one go, you can use the command `exp(coef(GLMgammaRdat)) - 1`.

Task 9 – Investigate ridge and LASSO regressions *(12 points)*

- *Content:* This 12-point task explores whether the use of regularization produces a more predictive model. To begin with, you are required to describe how regularization works and differs from stepwise selection you performed in Task 6 as an alternative to reducing model complexity and overfitting (keyword!). Details like the objective of regularization, how the penalty term is defined for ridge regression and lasso, how ridge regression and lasso compare, and how the hyperparameter λ is selected (by cross-validation) are expected. For more details, please refer to Subsection 3.2.5. You will then evaluate the test RMSE of the ridge regression model and the lasso model and recommend one model among the two penalized regression models and the GLM selected in Task 6.

  The model solution is not very explicit about why the gamma GLM in Task 6 is recommended. Because you are specifically asked not to "base your recommendation solely on the mean squared errors from each model," you can support your recommendation by saying that the gamma GLM not only is the most predictive in the sense that it has the lowest test RMSE, but also has the least number of features, which is appealing to the NC DOT. Keep in mind that identifying the key factors affecting crash severity is one of the most important objectives of the whole analysis.

- *Coding:* The Rmd template has code for fitting a lasso model to the training set, determining the value of λ that minimizes the cross-validation error, and calculating the test RMSE for the best model. The only change required is to add the interaction variable when setting up the design matrix in the training set as well as in the test set. To repeat the whole process for ridge regression, simply set `alpha` to `0` instead of `1`. Unlike Task 5, there is no need to change the family argument from `gaussian` to `Gamma`, which is not allowed by the `glmnet()` function. The fact that `glmnet()` has restrictions on the model form is a downside to regularization.

Task 10 – Consider a decision tree ***(5 points)*** In this task you are asked to describe the pros and cons of using a regression decision tree, which is an alternative model for predicting `Crash_Score`. No fitting is needed. You will find the advantages and disadvantages discussed towards the end of Subsection 5.1.1 useful. Before describing the pros and cons, it is also a good idea to give a precise and concise description of decision trees as in the model solution:

> "A regression decision tree is an alternative method of linking predictors to a target variable. A tree divides the feature space into a finite, non-overlapping set of buckets. All observations in a given bucket have the same predicted value."

Note that because almost all of the variables in this exam are factors, advantages and disadvantages that are specific to continuous variables score lower.

It is expected that future exams will have one task like this to assess your conceptual understanding of an alternative predictive model not tested in the project. An exam project may, for example, test GLMs and base decision trees in the main body of the project and ask you to consider the merits and demerits of using ensemble trees like random forests and boosting. At a minimum, you should list and briefly describe *at least two to three* advantages and disadvantages. The more, the better, so long as your points are well justified and not conflicting with one another.

Task 11 – Executive summary ***(20 points)*** At long last, you are asked to summarize your analysis using an executive summary. Please refer to the "Structure of Executive Summary" section earlier on how to write a good executive summary. You can see that the model solution indeed follows the suggested 4-part structure very closely. Also notice that the solution makes good use of the information given on the first page of the project statement, e.g., "If this investigation looks promising, they will provide statewide data for further analysis" is rephrased in the concluding paragraph as a possible next step.

Note that the executive summary should only cover your work in Tasks 1-8, but not Task 9. In other words, there is no need to (and we shouldn't) discuss what we did using the ridge regression and lasso models.

Hospital Readmissions Sample Project

This sample project was released in May 2019, one month before the June 2019 PA exam (yes, the SOA announced the change of exam format just one month before the June exam!!). Similar to the June exam, this sample project has a total of 11 miscellaneous tasks covering data exploration, data cleaning, model construction, feature selection, and model interpretation. Here we are hired by a group of hospitals to construct a GLM for identifying patients who are most likely to be readmitted based on the given patient level data. The target variable is `Readmission.Status`, a binary variable that reflects whether a patient is readmitted or not. The GLM constructed is required to outperform