Dr. Straw's Tips for Success in D208

June 1, 2022

A new updated document will be posted toward the beginning of each month. Other updates may be posted during the month to fix a link or correct some other problem.

Please send all questions and suggestions with the subject: "D208 tips suggestion" to eric.straw@wgu.edu.

My tips provide answers to the most common student questions. These tips are sorted in alphabetical order by title. You may want to search this document for key words.

Bivariate Visualizations

Donatello and Roualdes (2020) have a great section on bivariate visualizations that provides several options for each combination of variable types (i.e. categorical and continuous). This book provides examples in R and these examples can be adapted to Python.

Donatello, R. & Roualdes, E. (2020). *Applied Statistics*. Section 2.4 Bivariate Visualizations. Available at https://norcalbiostat.github.io/AppliedStatistics_notes/bivariate-visualizations.html

Confusion Matrix in R

You will receive an error on the R confusionMatrix() function if the variables are not factors and if the variables have different levels. A factor is the name for a categorical variable in R. You can use either the as.factor() or factor() function to ensure your variables are treated as factors. You must also ensure your variables have the same levels. Factor variables have categories (e.g. hot or cold; 1, 2, or 3; etc.). In the language of R, these are called levels. Thus, a factor variable has levels. You can use either the factor() or levels() function to ensure your variables in the confusionMatrix() function have the same number of levels.

Here is an example confusionMatrix() statement:

```
confusionMatrix(as.factor(MyLogisticModel$MyDependentVariable), as.factor(MyPredictedVariable))
```

Data and Data Dictionary

Do not use the data from a previous class (e.g. D206, D207). Download the D208 data and data dictionary for D208.

1. Go to the D208 course page
2. Select View Task under Assessments at the bottom center of the page
3. Select D208 Definitions and Data Files under Scenario on the Task Overview page
4. Select the link for the data set you will be using
5. Unzip the downloaded folder
6. The data file is in CSV format
7. The data dictionary is in PDF format. Ignore the Scenario on page 1 of the PDF. The Scenario has nothing to do with your work in this class.

Data: Hard Choices

The data (both churn and medical) is designed to force hard decisions. There are no outstanding models hidden in this data. There are only hard choices. The goal of the tasks is not to produce a beautiful model of which you can be proud. The goal of the tasks is to follow a decision-making process that you can be proud of given the hard choices you must make. And, you need to justify the decisions you make in your narrative. Making the decisions and justifying your decisions is an essential element of the tasks.

Data: PDF Scenarios

The scenarios described on page 1 of both the churn dataset and the medical dataset are just examples. Even though these say, "You have been asked to..." it does not mean that you should use the example scenario as the basis for your analysis in D208. In fact, both example scenarios use a categorical dependent variable, which is appropriate for logistic regression but not for linear regression because linear regression requires a continuous dependent variable. Thus, you will need to select a continuous variable for your dependent variable for the linear regression.

DataCamp: Data Files

Do the following to access the data files for the D208 resources in DataCamp.

(1) From the D208 custom track in DataCamp (i.e. the landing page), select a course title.

(2) You will find the data files for that course at the bottom right corner of the page.

Python data files are in CSV format. R data files are in FST (fast storage) format. These FST files require the fst package and use of read_fst().

DataCamp: PDF of Slides

You can download a PDF of the slides for a DataCamp chapter by selecting the page icon in the upper right corner of any of the chapter's videos. Having these slides available will make your studies more efficient because you will not need to search online for syntax help as you complete the demonstration portion after each video.

You can also view the slides on the Slides tab next to the R Console in the exercises. However, this view is quite small and challenging to view.

Dummy Variables

We create dummy variables (also called indicator variables) to represent categorical variables. Dummy variables always contain values of 1 or 0. For example, a categorical variable may contain one of three categories (i.e. k=3): Yes, No, or Maybe. For linear and logistic regression, we would encode this as two dummy variables (i.e. k-1=2), perhaps labeled as Yes and No. If Yes=0 and No=0 then we know the response was Maybe. Thus, for linear and logistic regression we always have one fewer dummy variables (k-1) than the number of categories in our categorical variable (k). Adding a Maybe dummy variable to our example would provide no new information and would create a multicollinearity problem in linear and logistic regression.

This rule does not apply to machine learning algorithms such as KNN (D209 Task 1) and regression trees (D209 Task 2). See Shmueli (2015) for more details at http://www.bzst.com/2015/08/categorical-predictors-how-many-dummies.html

Failure to Converge

R may return a warning of "glm.fit: fitted probabilities numerically 0 or 1 occurred" because the log likelihood optimization failed to converge. This is a problem with complete or nearly-complete separation. The Institute for Digital Research & Education has a good article on separation and includes examples in R as well as a section focused on techniques for dealing with separation.

For those using Python, scikit-learn's LogisticRegression method should prevent the separation problem through the C parameter as described by Pananos (2018).

Features – too many

Adding more independent variables to your regression model will reduce the Root Mean Squared Error (RMSE) and increase the R-squared. But more independent variables does not necessarily make a better model. Thus, you should select the model with the highest adjusted R-squared and lowest Akaike's Information Criteria (AIC) because both penalize the model based on the number of independent variables.

Feature Selection

Regression analysis requires careful feature selection -- choosing which independent variables to retain in your model. Filter-based statistics and Recursive Feature Elimination (RFE) are valuable techniques. However, Principal Component Analysis (PCA) and Multiple Correspondence Analysis (MCA) are not because PCA and MCA are feature elimination techniques, not feature selection techniques.

Smith(2018) discusses why RFE is a better option than stepwise.

RFE with Python: Brownlee (2020) illustrates RFE in Python. Additionally, the caret package book includes a chapter with more details on using the RFE function. And, this Data Camp course includes RFE.

RFE with R: Perez-Riverol (2018) has a section illustrating RFE in R.

Independent Variable Selection

You can use either pairwise correlation via a heatmap or analysis of variance (ANOVA) to help you identify independent variables for your initial model because you are required to statistically justify your exclusion of variables from your initial list of independent variables. For pairwise correlation, use Pearson correlation for continuous-to-continuous relationships and point-biserial correlation for categorical-to-continuous relationships.

You should include a good number of independent variables in your model so that you can demonstrate the model reduction required in D2 of each task. Somewhere between 6 and 12 independent variables will allow you to demonstrate this process and control the workload caused by adding more independent variables.

Logistic Regression Equation

Knowing how to interpret the logistic regression model (i.e. equation) is important. The UCLA Institute for Digital Research & Education has a great article that explains the interpretation in detail. The key to this explanation is found in the section entitled, *Logistic regression with a single continuous predictor variable*. The regression output shows a *math* independent variable with a coefficient of 0.1563404 and then explains, "for a one-unit increase in the math score, the expected change in [natural] log odds [of being in an honors class] is .1563404."

Every coefficient in the logistic regression model can be interpreted this way: (1) For continuous variables, a one-unit change in an independent variable causes the natural log of the probability of the dependent variable being 1 to change by the value of the coefficient of that independent variable. (2) For categorical variables, which can be either 0 or 1, the natural log of the probability of the dependent variable being 1 is not changed when the categorical variable is 0 (0 x coefficient=0), but changes by the value of the coefficient when the value of the categorical variable is 1 (1 x coefficient=coefficient).

Here is the form of the logistic regression equation, where p is the probability that your dependent variable is equal to 1.

$$\text{logit}(p) = B_0 + B_1x_1 +....+B_kx_k$$

Sometimes you will see the equation written like this.

$$\ln(p/(1-p)) = B_0 + B_1x_1 +....+B_kx_k \text{ because } \text{logit}(p) = \ln(p/(1-p))$$

You can read the full article for all the details or scan the article to help you understand concepts like natural log odds. The author uses the phrase "log odds" to mean natural log of the probability (i.e. log base e of the probability).

Multicollinearity

Multicollinearity can be challenging to understand, identify, and avoid. Frost (2017) has a great discussion and video on multicollinearity. One of the keys is deciding how much correlation of independent variables you are willing to tolerate. See the Testing for multicollinearity with Variance Inflation Factors (VIF) section in Frost for guidance.

Frost, J. (2017). Multicollinearity in regression analysis: Problems, detection, and solutions. Statistics by Jim.

Order of Tasks

The data for D208 indicates that students should start with Task 1 first.

Average days to complete the course is 10 days shorter for those who start with Task 1

51 days for starting with Task 1 vs 61 days for starting with Task 2

A higher percentage of students who start with Task 1 pass their first submission

42% for Task 1 when starting with Task 1 vs 36% for Task 2 when starting with Task 2

40% for Task 2 when starting with Task 1 vs 27% for Task 1 when starting with Task 2

Panopto

You must narrate and explain your code in your Panopto videos in D208, and you will need to create a Panopto video for each task. Your Panopto videos are one way you will demonstrate your solution to each task.

There are two links at the bottom of each task overview page: (1) Panopto Access; and (2) Panopto how-to videos. I encourage you to ensure you have Panopto access as soon as possible. This access allows you to place your completed video in the D208 course folder.

Process Overview

This high-level overview of the process flow for D208 may help you avoid problems and save time.

1. Explore and prepare the data.
2. Select your research question and dependent variable.

3. Cast a wide net to identify a good number of independent variables for your initial model. Use wisdom and a statistical method to select these initial independent variables. Using correlation coefficient is a great statistical method to select your initial independent variables. Ensure your cutoff is such that you select a good number of independent variables. There is no correct number, but 6 to 12 independent variables is a good range and gives you the opportunity to demonstrate the model reduction requirement in D2.
4. Reduce your model by eliminating independent variables until you arrive at your final reduced model, which will be your best model. Using p-values is the minimum method for model reduction. In addition, you should use something like stepwise (which relies on p-values) or RFE (Recursive Feature Elimination) or one of the many other methods you can learn about online.
5. Compare your initial model and final reduced model via metrics.

Pseudo R-Squared

Pseudo R-squared in logistic regression is not the same as R-squared in linear regression.

- The range of pseudo R-squared is NOT limited to 0-1
- Pseudo R-squared does NOT represent the amount of variation in the dependent variable that is explained by the independent variables

"While pseudo R-squareds cannot be interpreted independently or compared across datasets, they are valid and useful in evaluating multiple models predicting the same outcome on the same dataset. In other words, a pseudo R-squared statistic without context has little meaning. A pseudo R-squared only has meaning when compared to another pseudo R-squared of the same type, on the same data, predicting the same outcome. In this situation, the higher pseudo R-squared indicates which model better predicts the outcome." (UCLA Advanced Research Computing Statistical Methods and Data Analytics FAQ, next to last paragraph, available at https://stats.oarc.ucla.edu/other/mult-pkg/faq/general/faq-what-are-pseudo-r-squareds/)

P-Value

The p-value is the probability of getting the result if the null hypothesis is true (i.e. rejecting the null hypothesis when it is true).

For regression, we look at the p-value for each independent variable and remove independent variables that have a p-value above our selected significance level.

The null hypothesis for each independent variable is: This independent variable has no effect on the dependent variable.

A p-value < 0.05 (also called the 0.05 significance level) means there is only a 5% probability that we could get the results we did if this independent variable has no effect on the dependent variable. The common significance levels are 0.01, 0.05, and 0.10.

Hannes has a detailed walk-thru on p-values and related statistics if you need a refresher. Hannes uses R, but the concepts apply to Python as well.

Research Questions

The research questions you select for the tasks in this course must be broad enough to include a good number of independent variables. For example, "Does A cause B?" is too narrow. A much better approach is to ask, "What causes B?"

You can use industry knowledge, wisdom, and pairwise correlation to identify the appropriate variables to include in your initial model. You are required to statistically justify your exclusion of variables from your initial list of independent variables. Pairwise correlation gives you this statistical justification. You need to cast a wide net (i.e. include a good number of independent variables). There is no magic number, but 6 to 12 independent variables is a good range and gives you the opportunity to demonstrate the model reduction requirement in D2.

You will reduce your initial model by removing independent variables that have little or no influence on the dependent variable. P-values and Recursive feature elimination (RFE) are great strategies for reducing your model. Your final model will include only a handful of independent variables and these will be the independent variables that influence the dependent variable. Do not write your research question to attempt to capture this final model. Write a broad research question and then work toward discovering the answer.

Residual Errors

The requirement description in D208 Task 1, E2 uses the phrase, "...including the model's residual error." You can use residual errors to evaluate the fit of a linear regression model by evaluating both the residual standard error (RSE) and plots of the residuals. You will need to do both to meet the requirements for Task 1.

The RSE is a standard deviation of the residuals and is in the units of the dependent variable. A smaller RSE means a better fitting model. R provides the RSE with the summary() function. You need to calculate the RSE in Python.

Artificial Intelligence and Machine Learning website has examples of calculating RSE and other metrics in Python.

Residuals are the difference between actual values and predicted values for the dependent variable and are calculated on each observation. As a formula, this looks like,

Residual = Actual - Predicted

You should produce both a residual plot (aka density plot) and a Q-Q plot (aka normal probability plot) to evaluate the residuals. The residual plot places the predicted values on the x-axis and the residuals on the y-axis. The Q-Q plot places theoretical quantiles on the x-axis and the residuals on the y-axis. Qualtrics has a good article illustrating visual evaluation of the residual plot. The Ecological Modelling group at UCD has a good article illustrating visual evaluation of the Q-Q plot.

Start with Task 1 first

The data for D208 indicates that students should start with Task 1 first.

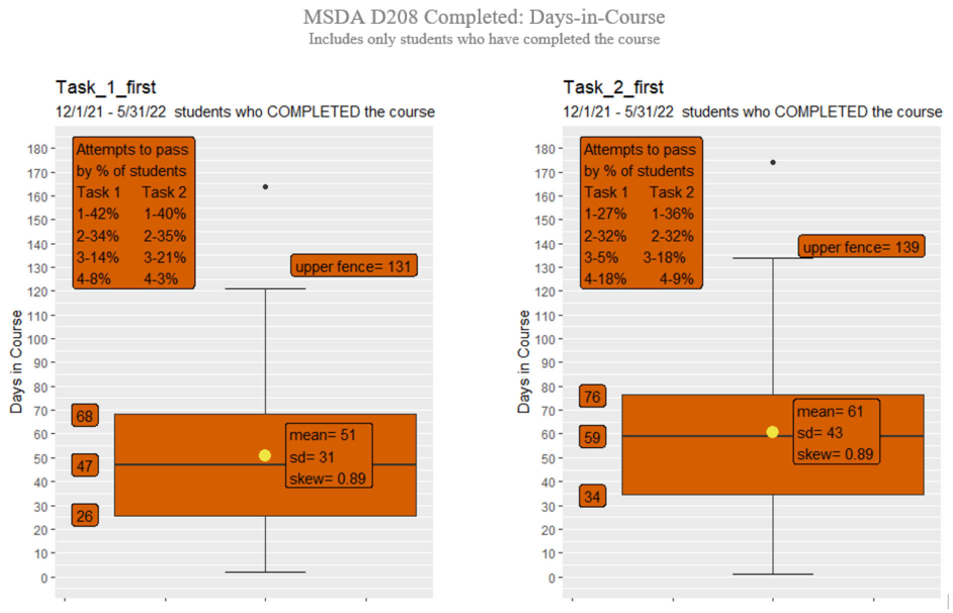
Average days to complete the course is 10 days shorter for those who start with Task 1

51 days for starting with Task 1 vs 61 days for starting with Task 2

A higher percentage of students who start with Task 1 pass their first submission

42% for Task 1 when starting with Task 1 vs 36% for Task 2 when starting with Task 2

40% for Task 2 when starting with Task 1 vs 27% for Task 1 when starting with Task 2



Task Requirement C2

Requirement C2 requires you to include the predictor variables that you think are needed to answer your research question. This requirement uses the word "all" and sometimes causes confusion for students. It does not mean "all predictor variables in the dataset". It means the predictor variables that you think are needed to answer your research question.

Task Requirement D3

Requirement D3 for Task 1 (linear regression) states that you must include at least one continuous independent variable and at least one categorical independent variable. This is an academic requirement, not a fundamental requirement when doing linear regression.

Requirement D3 for Task 2 (logistic regression) does not have this same academic requirement. Thus, you can use any combination of types of variables as independent variables in Task 2.

There is a fundamental requirement for the dependent variable that has restrictions on what type variable you can use – continuous dependent variable for linear regression and categorical dependent variable for logistic regression. These restrictions are not simply academic, they are fundamental to model building through linear regression and logistic regression.

Textbooks

Data Science Using Python and R , which was the textbook in D206, provides good supplemental content for this course. Relevant material includes the following chapters and sections. You can find additional helpful content by searching this book.

- Chapter 11 Regression Modeling (all)
- Chapter 12 Dimension Reduction (sections 12.3 and 12.4)
- Chapter 13 Generalized Linear Models (section 13.4)

Practical Statistics for Data Scientists is also a great supplemental resource in this course. Relevant material includes the following chapters and sections. You can find additional helpful content by searching this book.

- Chapter 4: Regression and Prediction (Simple Linear Regression; Multiple Linear Regression; Factor Variables in Regression; and Interpreting the Regression Equation)
- Chapter 5: Classification (Logistic Regression and Evaluating Classification Models)

Visualizations (graphs, charts, etc.)

The Python Graph Gallery and R Graph Gallery are each the single best resource for visualizations for each tool. Both of these sites are beautifully designed with an easy visual search home page and navigation bar, and provide detailed examples and suggestions for coding and improving your visualizations.

y-Intercept

You need to include a y-intercept in your regression model. Some methods force the model to pass through zero, which results in a model without a y-intercept. This is possible but not desirable.

Using Python

If you use statsmodels, you should use statsmodels.formula.api to include the y-intercept. Using statsmodels.api removes the y-intercept unless you add a column of 1s to your X. See the statsmodels documentation for more details: https://www.statsmodels.org/stable/index.html

Using R

Do not include an offset in your lm() or glm() function call. An offset can be seen in some online examples as a +0 or -1 in the lm() or glm() function call. Adding an offset will remove the y-intercept. See the latest version of stats documentation for more detail: https://www.rdocumentation.org/packages/stats/versions/3.6.2