DAEN 500- DL2 – Data Analytics Fundamentals

Fall 2020 Final Examination Exercise Package

10/7 – 10/13/2020

Final Submission Deadline: NLT 11:59PM (EST). Tuesday, 13 Oct 2020

*Failure to submit will result in DAEN COURSE FAIL/NoCredit*

Name:\_\_\_\_Brad Dyer\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ GMU G#\_\_\_01296321\_\_\_\_\_\_\_\_

Student Signature (Honor Certification): \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

This exam is **OPEN BOOK/OPEN NOTES**. You may consult any of the course texts, and the various reference materials recommended in the syllabus. ***The exam of course IS NOT “Open Web”,*** especially in that you may NOT utilize expert “help” sites such as Stack Overflow, or other programming help or collaboration sites.

HONOR CODE CERTIFICATION

**Your signature above declares that you have followed the conditions of this exam, and that the work is yours alone**. **Specifically:**

This must be your own work, authored and completed by you. As stated earlier, this is an “open source exam” – allowing books, notes or courseware, as well as *general* expert advice gained PRIOR to exam. YOU MAY NOT, HOWEVER, SEED OR USE ANY ADVICE ON HOW TO SOLVE THE QUESTION OR ANY CODE WRITTEN BY ANY OTHER INDIVIDUAL. *Any violation will result in an immediate failure in the exam and for the course, as well as referral to the GMU Honor Committee for determination of any other appropriate disciplinary consequences.*

*NOTE: Your* ***submission*** *of any responses, files, programs, etc. in response to the DAEN500 final exam instructions, will also be your personal certification of your full compliance with the spirit and letter of the* ***GMU Honor Code*** *standards for take home and/or in-class exams.*

Additionally, you are restricted from discussing the substance of the questions on this exam with any other individual, until after you have submitted your final response for grading. The completed exam -- with your answers embedded in this docx document (add extra pages as necessary) should be submitted following instructions contained in the Final Exam Instructions BB site. If you have any trouble submitting and have extra parts of the answers you have trouble appending to this document, you may simply submit additional pages separately (***the exam submission site is set for multiple submissions, just in case***). Make certain all are submitted PRIOR TO THE DEADLINE!

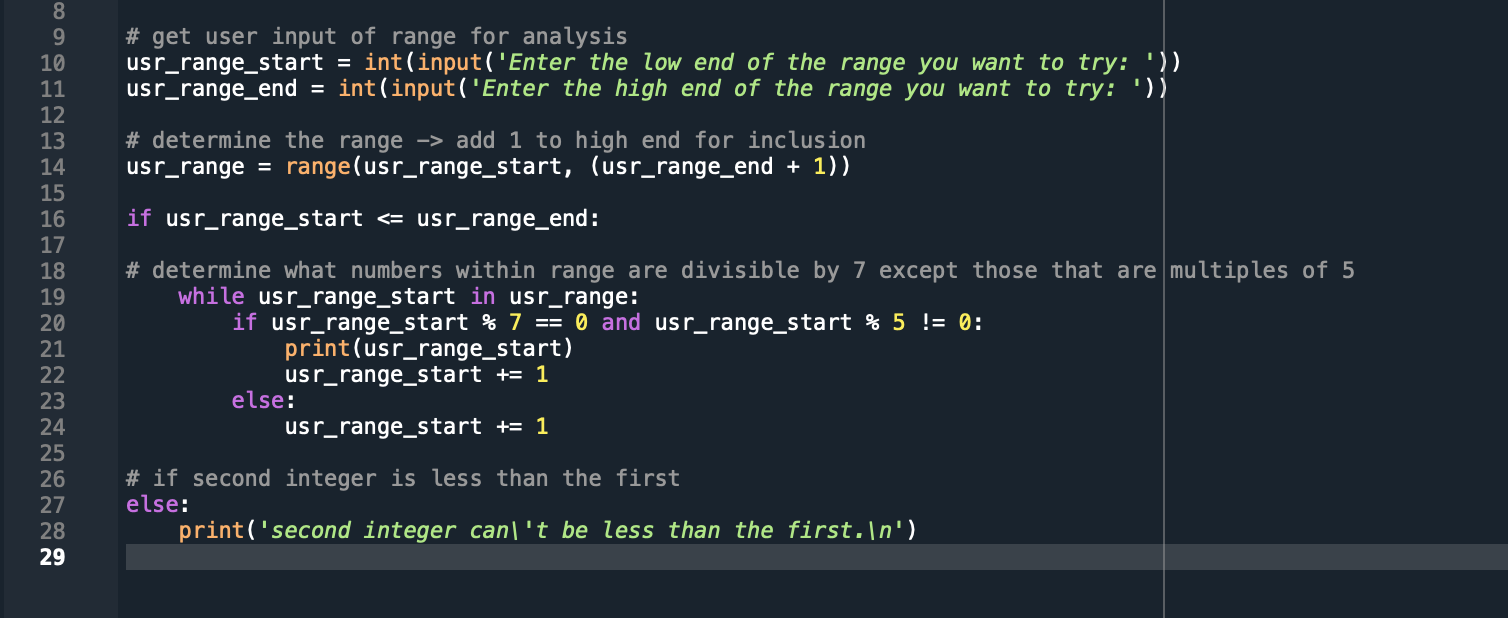
 FINAL EXAM PROBLEMS

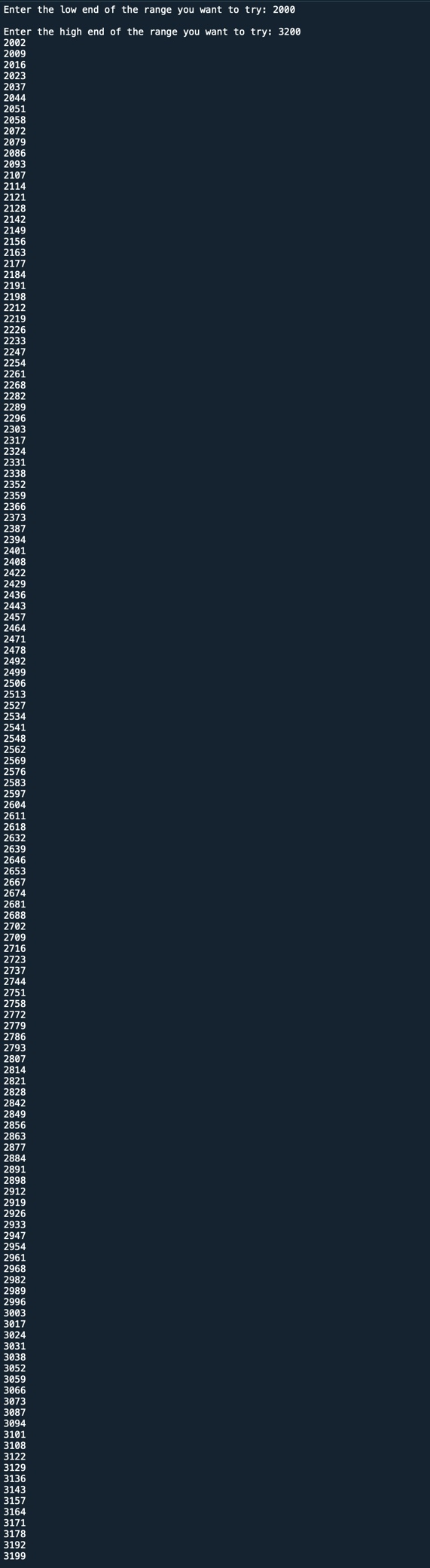
COMPLETE ALL & INSERT ANSWERS BELOW QUESTIONS

# Problem 1: Python Programming Problem (15 Points Total)

* **Design and implement a Python program that is based on the following requirements: a) program will *find* *all numbers -- within a specified range -- which are divisible by 7 but are not a multiple of 5*; and b) demonstrate the program works by running the program for the range: *numbers between 2000 and 3200*.**
* **INSERT (cut&paste) your Python code in space below and *then insert a screen shot in space below, showing your successful run and output.***

NOTE of alternative for help: To help test your code, you also may use a Python “programming window” found in the. **Zybooks Section 35 Additional Material**.



****

# Problem 2: Python Programming Problem

# (15 Points Total)

* **Design and implement a Python program that is based on the following requirements:**

**a) define a class which has *at least two* methods**

* + **Method 1 – getString: to get a string from console input; and,**
  + **Method 2 - printString: to print the string in upper case.**

**b) *demonstrate code works using three different test input strings***

* ***INSERT* *code below* and *INSERT* a screen shot of the program and successfully run output that *includes test input for input strings (test strings must include (a) all upper case, (b) all lower case, and (c) mix of upper and lower case).***

﻿class Input\_and\_shout:

def \_\_init\_\_(self):

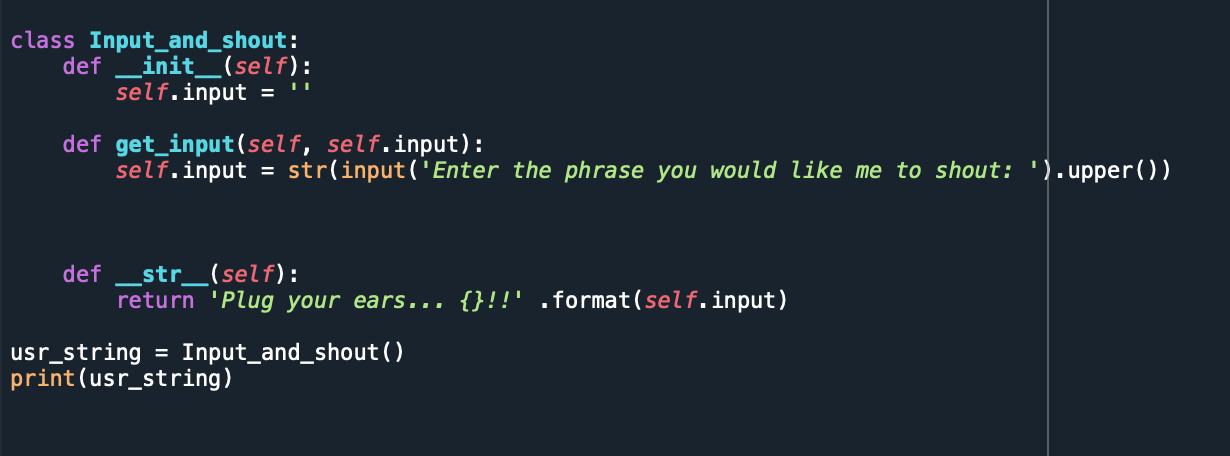
self.input = str(input('Enter the phrase you would like me to shout: ').upper())

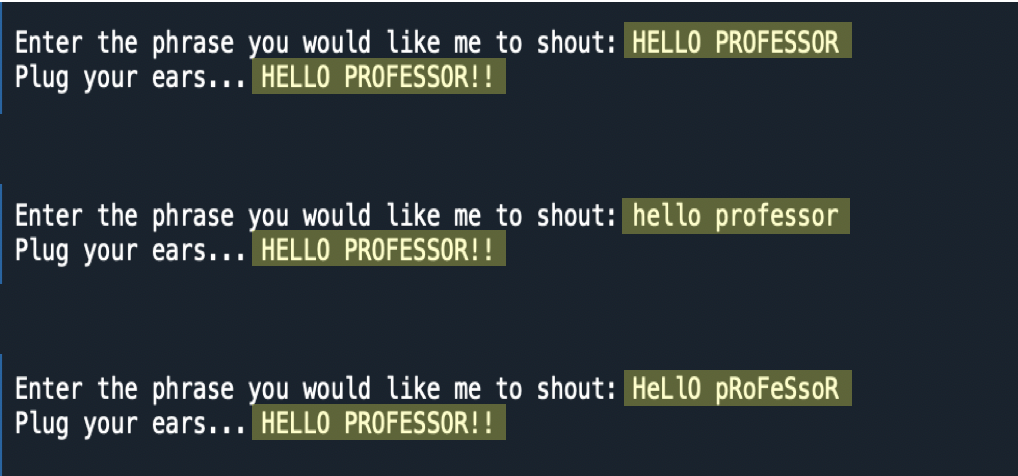
def \_\_str\_\_(self):

return 'Plug your ears... {}!!' .format(self.input)

usr\_string = Input\_and\_shout()

print(usr\_string)





As a side note, I do realize that the first method converts the text to upper case versus the second method (as requested in the instructions). Obviously, it would have been easier to do it in the second method but my intent was to make the output a little more “fun” by giving the process a purpose – to “shout” back (in text) the phrase the user gave as an input – and I did not want to capitalize the whole phrase… in order to give the user a warning so that they had time to “plug their ears” before the shouting.



# Problem 3: R Programming Problem

# (20 Points Total)

* **Perform the following problems using R:**
  + Create a vector of courses (e.g., MATH 101) you have taken previously. Make sure you have at least 8 courses. Name the vector myCourses
  + Get the length of the vector myCourses
  + Get the first two courses from myCourses
  + Get the 3rd and 4th courses from myCourses
  + Sort myCourses using a method
  + Sort myCourse in the reverse direction
* *INSERT* *code below* and *INSERT* a screen shot of the program and successfully run output.

**# Create the vector myCourses**

**myCourses <- c('Chem 111', 'Comp Sci 110', 'Spanish 141', 'Physics 215', 'Stats 220', 'Mgt 486', 'Aviation 251', 'Philos 310', 'Phys Ed 486', 'Econ 310')**

**# Get the length**

**length(myCourses)**

**# Get the first two courses**

**myCourses[1:2]**

**# Get the 3rd and 4th courses**

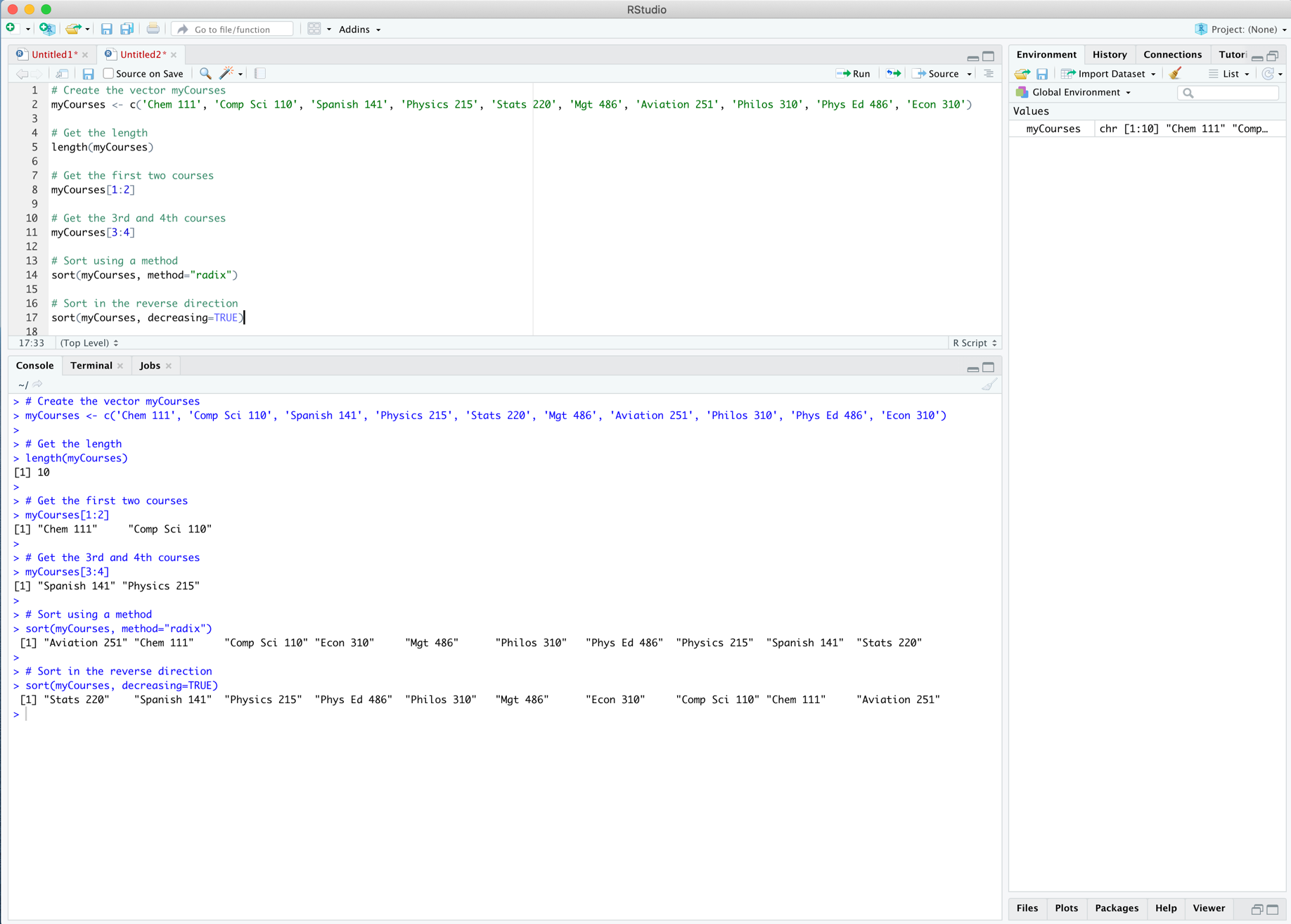
**myCourses[3:4]**

**# Sort using a method**

**sort(myCourses, method="radix")**

**# Sort in the reverse direction**

**sort(myCourses, decreasing=TRUE)**





# Problem 4: Principal Component Analysis

# (25 points)

**Provide a description of the following:**

1. What is a component – Provide a description (5 points)
2. Principal Component Analysis – Provide a description (5 points)
3. Provide and explain a specific example of a Principal Component Analysis (15 points)

**ANSWER:**

1. A component in the context of Principle Component Analysis (PCA) is the term used to describe the new variable that is created when interrelated predictor variables are combined during PCA to simplify analysis. They are typically labeled as Z1, Z2, etc..
2. Principal Component Analysis simplifies the analysis of predictor variables by combining original predictor variables that may be interdependent. This process aids in determining which predictors best explain variability in the data and makes it simpler by reducing the number of parameters in the analysis. The goal is to use as few components as possible while still maximizing data variability.
3. One specific example (without using the internet to search for a specific example because, as instructed, the test is not “Open Web”) would be a behavioral analysis predicting what variables determine whether someone will become a smoker. In this example there could be many interrelated variables such as any type of “euphoric feeling” a smoker may or may not have had from his or her first cigarette. This feeling’s original variables could be described as heightened awareness, a spike of energy, dizziness, a feeling of mental focus, and others. Testing each of these predictor variables individually could make the analysis unnecessarily complicated when they could be combined through principle component analysis into a single component of “euphoria” when determined that they are interrelated.

# Problem 5: Multiple vs. Logistic

# (30 points)

# Describe: What is difference between Multiple Regression and Logistic Regression? What circumstances might determine which to use? (10 points)

# Demonstrate: Using any data, and any tool set you’ve learned about, show differences (20 points)

# SUGGESTION: may be solved using RapidMiner, or other toolsets, BOTH TO ANALYZE AND TO VISUALIZE REGRESSION DIFFERENCES…

Step 1: Perform a quick search of the [UCIS public data archive](https://archive.ics.uci.edu/), a well-curated site which you already have seen as part of your introductory RapidMiner training.

Step 2: Pick a dataset you find interesting, input dataset into regression tools you’ve chosen.

Step 3: Run the dataset (*may be a significant subset, if the dataset is very large*) first. a Multiple Regression and then a Logistic Regression, and use visualizations to demonstrate the conceptual answers you provided for 5.(a).

ANSWER 5(a):

Various regression analyses on data can be useful in validating results, but one type of analysis may be better than another. For example, one can use both multiple linear regression and logistic regression to study a dataset. Both methods model relationships between a quantitative response variable and one or more predictor variables. However, there are differences between the two models including circumstances when each should be used depending on the type of data.

One main difference between the two models relates to the outcome variable. Multiple linear regression output normally has a continuous relationship to the covariates. In other words, as the covariate changes along the x-axis, the output on the y-axis would change incrementally; the type and amount of change being dependent on the relationship between the outcome and the covariate(s). Comparatively, a logistic regression output is most appropriate for a binomial outcome – one in which there are only two possible outcomes. Therefore, the output variable is either “one” or “the other”. For example, in one data set I found, demographics such as race, location, marital status, etc. were used to determine whether someone’s salary was either <= 50K or > 50K, and that was it, either one or the other with no other options. In this case a logistic regression would likely be most appropriate because this is a binomial outcome. This is true in most circumstances. When choosing whether to use a multiple linear regression model or a logistic model, if the outcome type is binomial then one might start with a logistic regression model. The abalone data set I used had an outcome variable of age which ranged from 1 to 29 and therefore fit best into a multiple linear regression model. However, in order to fit it into a logistic regression model, I changed the age outcome for the data to binomial by dividing the age range into two categories – 1 to 10 years and greater than 10 years – in order to demonstrate the difference in analysis using a logistic regression using the same data (see [logistic regression analyis](#logregression) for results). The results demonstrate that between the two options, a multiple linear regression model fits better with a continuous outcome while a logistic regression fits better with a binomial outcome.

Another difference between the two models relates to the population linear regression function. Both models use this formula to mathematically describe the relationship within each model which results in an expected value. Because of the binomial relationship in a logistic regression model as described above, the expected value within the population linear regression function of a multiple linear model differs from that of a logistic regression. In a multiple linear regression, the expected value is the expected outcome given a set of values for the predictor variables. In the abalone data set, one can look at the scatterplots with the trend lines to determine the predicted outcome of age based on specific predictor variables (see [screenshot of Tableau dashboard](#TableauDash) below). In a logistic regression, the outcome instead represents the “proportion of 1’s”. In other words, it shows the likelihood of the outcome related to 1 occurring compared to outcome related to 0. For example, if the outcome related to 1 occurs 60 times out of 100, then the expected value, or outcome, is 60%. The logistic regression on the abalone data predicted the age would be greater than 10 years 98 times out of 366 so therefore the expected outcome would be 27%.

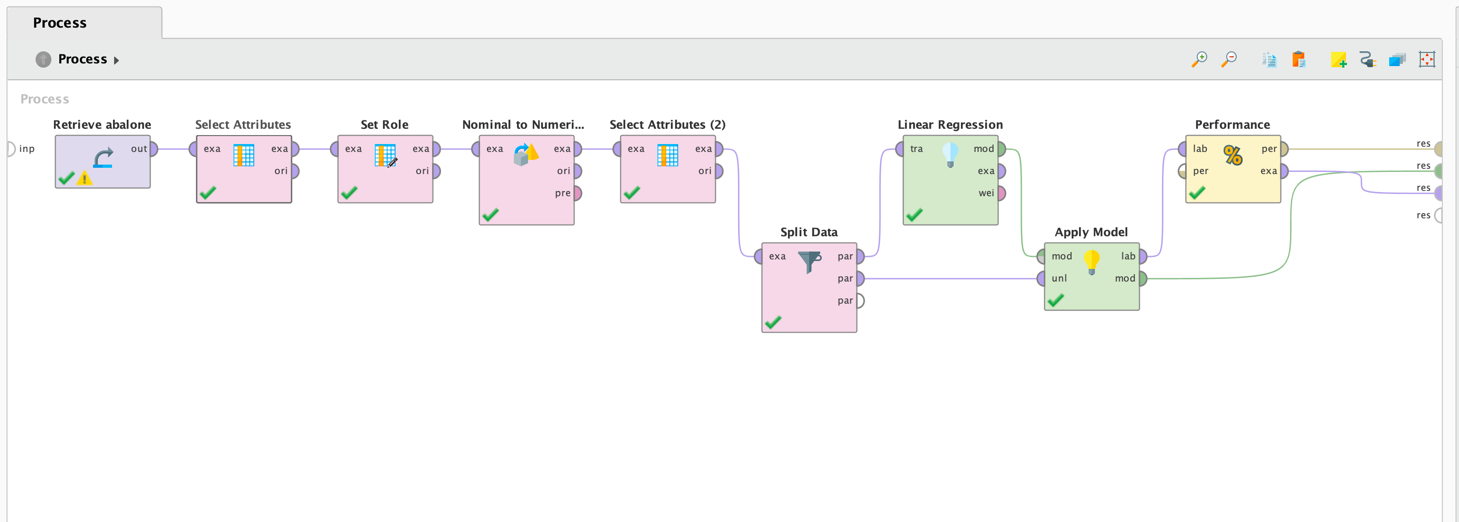
In addition, when estimating the population regression parameters from sample data in a linear regression model, one uses functions such as the sum of squared errors and the root mean square error to measure the error of the model. In the case of the abalone data, the root mean squared error was 2.297 indicating that this model, on average, predicts the age of the abalone within 2.297 years. However, in a logistic regression analysis, this method is not appropriate. Instead, one should use the maximum likelihood method to determine the best S-shaped curve for the data. In order to calculate the maximum likelihood for the abalone data, in addition to the calculation I did in RapidMiner, I also did a logistic regression model on the data in R because I was able to better manipulate the plots to get the b0 and b1 for the maximum likelihood. The coefficients section within the summary output lists the estimate for the “(Intercept)” and the given predictive variable which indicates the estimate b0 and b1 for the population logistic regression function – the intercept and slope respectively (see [R results](#Rresults) in 5(b) for more).

One final difference is related to the visualization that results from plotting a covariate to an outcome. Specifically, because the response must be either 0 and 1 in a logistic regression, all symbols on the plot representing the outcome will be along two separate lines paralleling the x-axis. One line will be at 0 and the other will be at 1. Then, if a trend line was to be plotted the curve would have an “S-like” shape. This makes sense conceptually when you realize that the line cannot proceed past the range of 0 to 1 on a binomial outcome and therefore the trend line must merge with data lines. On the other hand, the numerical predictive variable modeled by a linear regression is continuous along a much larger range. Therefore, comparatively, outcomes plotted in a multiple regression visualization, such as a scatterplot, should trend continuously and the trend line is also a continuous straight line with some constant slope (unless, of course, a non-linear relationship is introduced through transformation). This is where I admit my limitations however. I spent many hours trying to get both RapidMiner and Tableau to plot a trend line within the binomial logistic regression graph and had no success. I have come to the conclusion that RapidMiner will not plot a trend line for a binomial results – please tell me if I am wrong. As I mentioned in the previous paragraph however, I did, finally transition to using R’s visualization tools which eventually got me the graphs you see in the last section of 5(b) below.

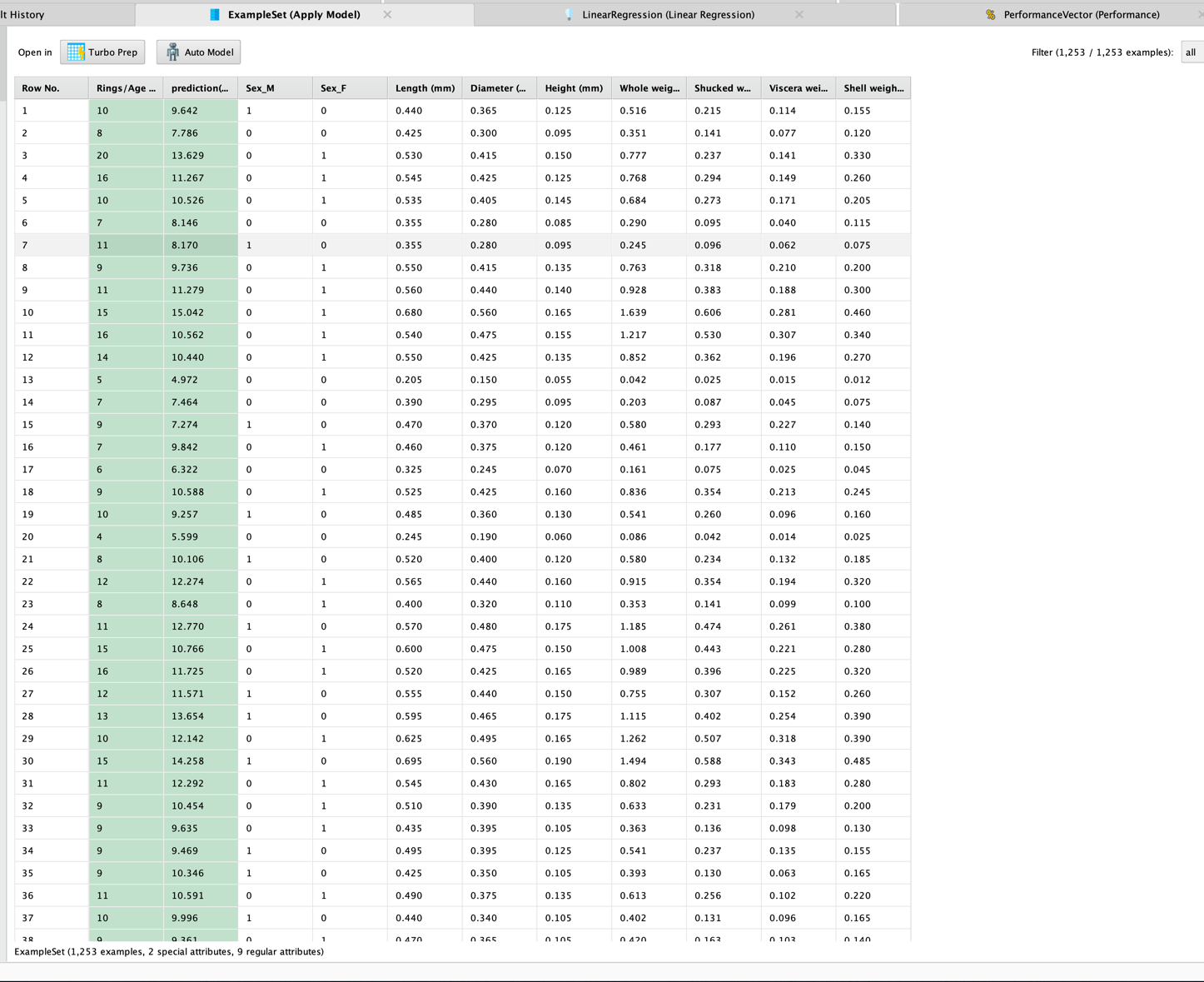
ANSWER 5(b)

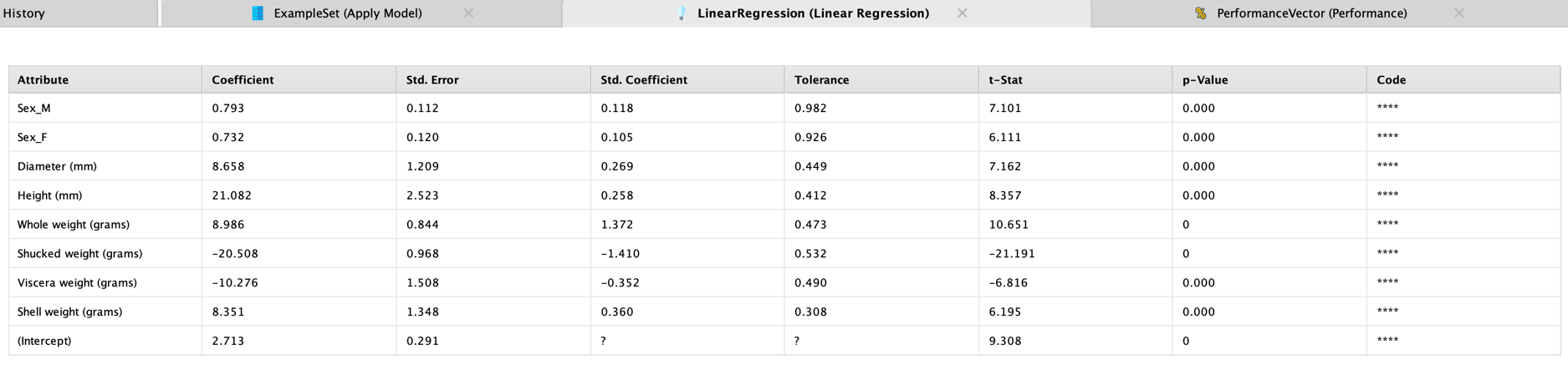
Linear Regression Model

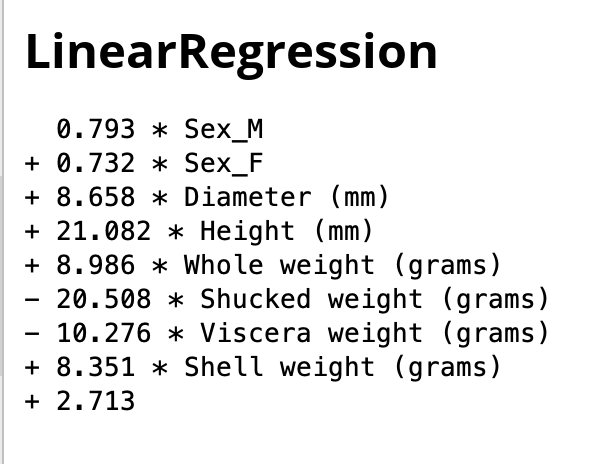
RapidMiner Multiple Linear Regression Analysis Design



Below is a sample output for the abalone data using the multiple linear regression model.







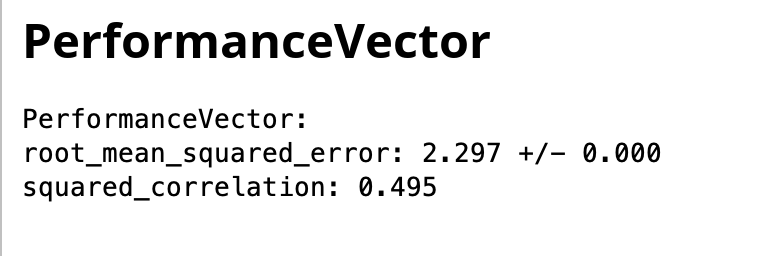
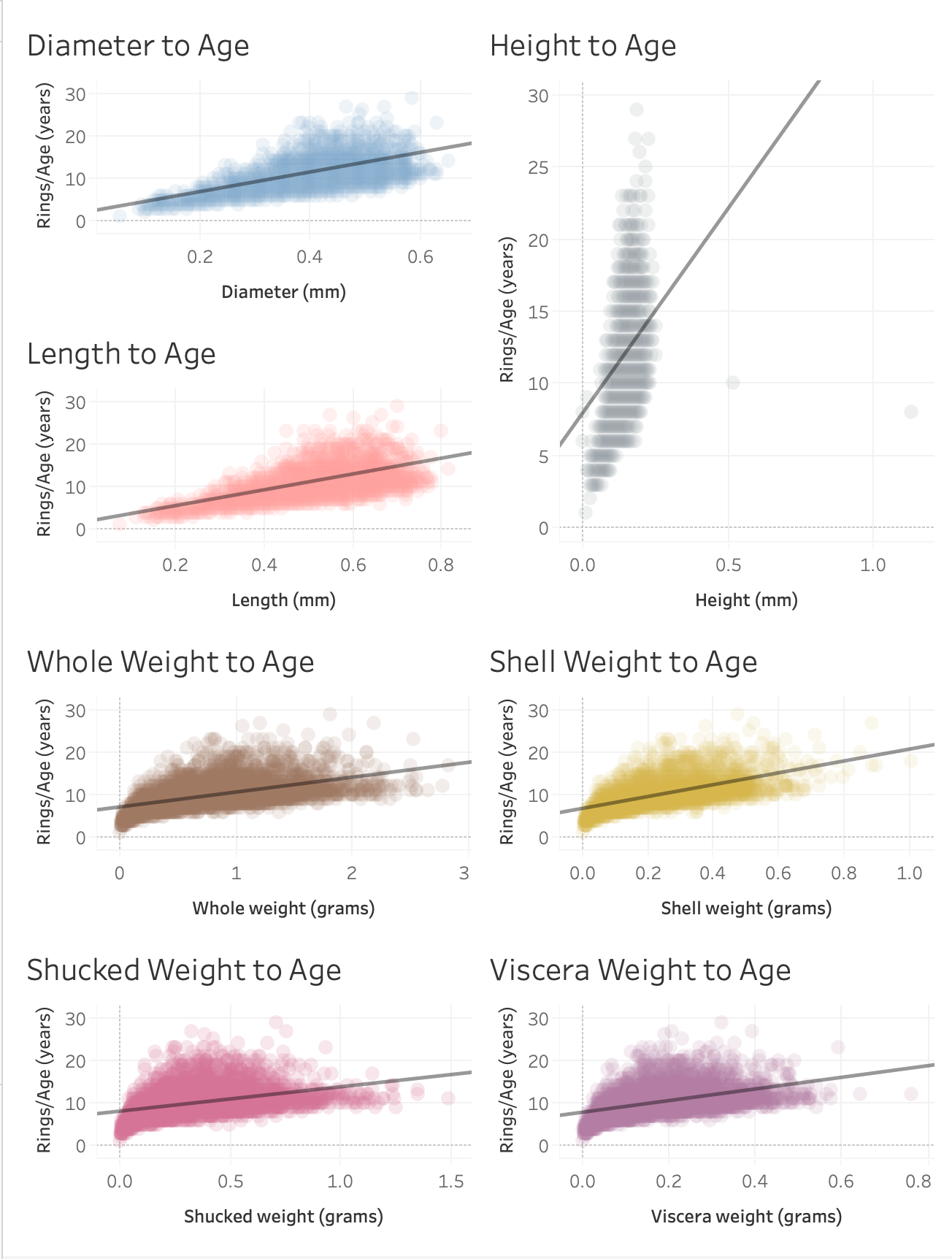
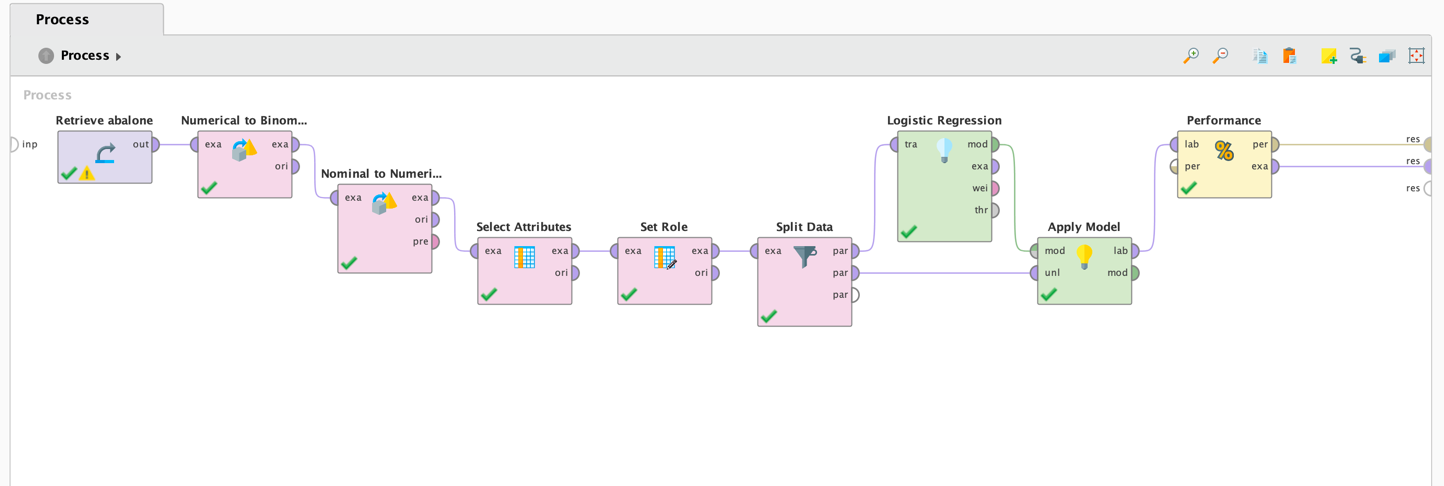


Tableau Dashboard graphical relationships of abalone inputs to age ([back to 5a](#Para_TableauDash))

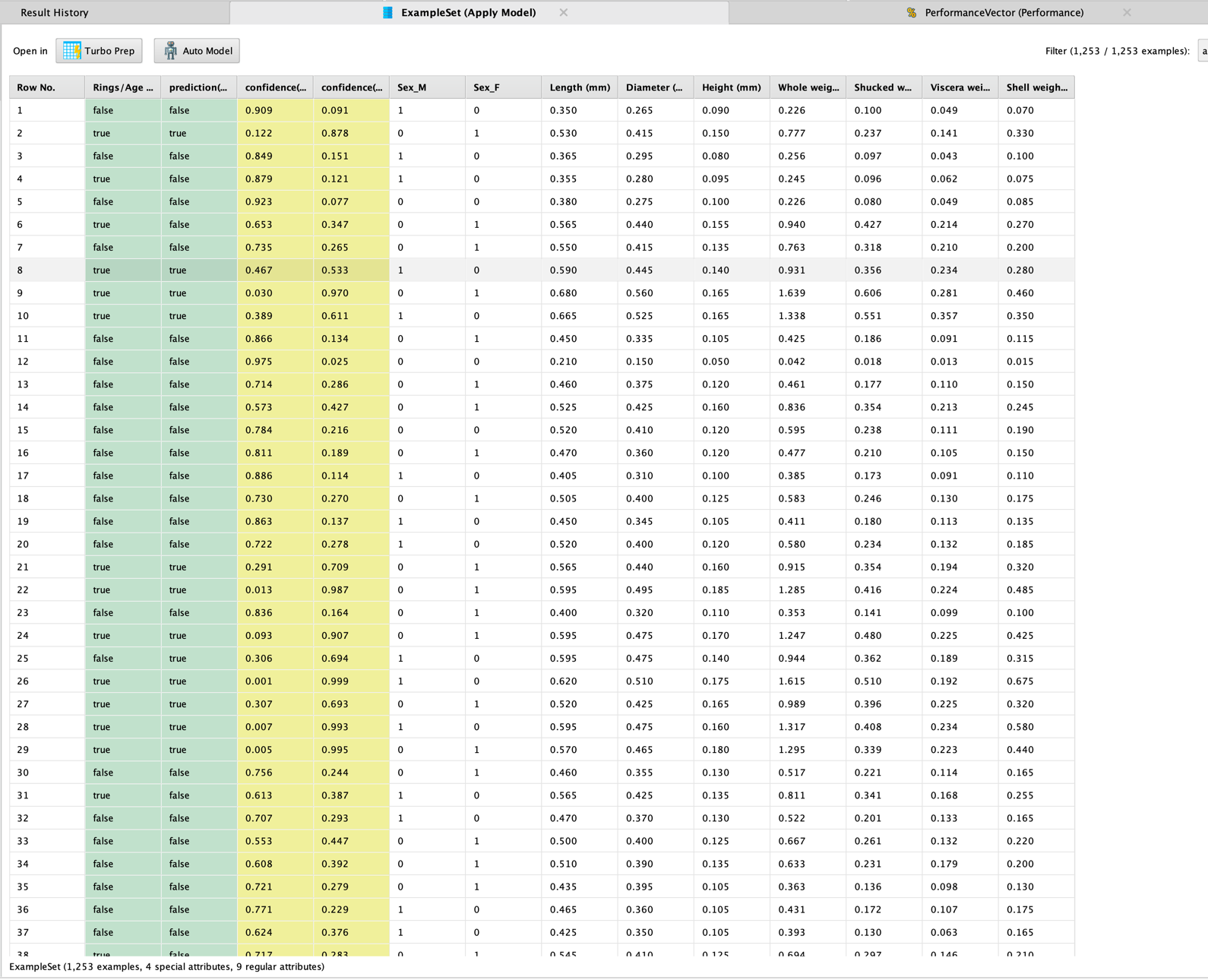


Logistic Regression Model ([back to 5a](#Para_logregression))

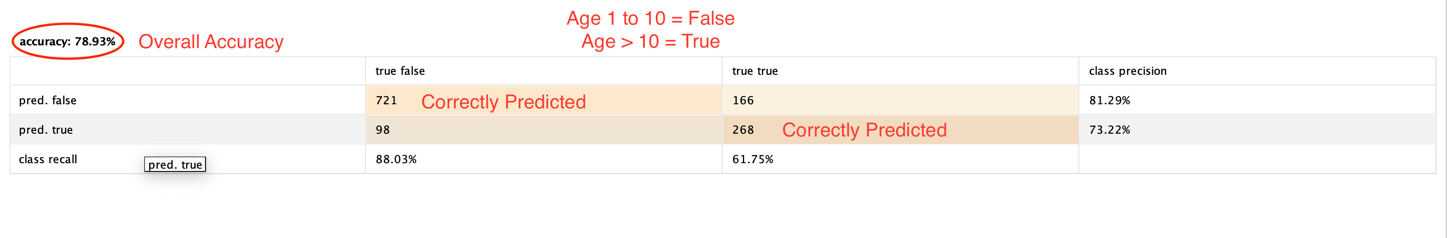
RapidMiner Logistic Regression Analysis Design



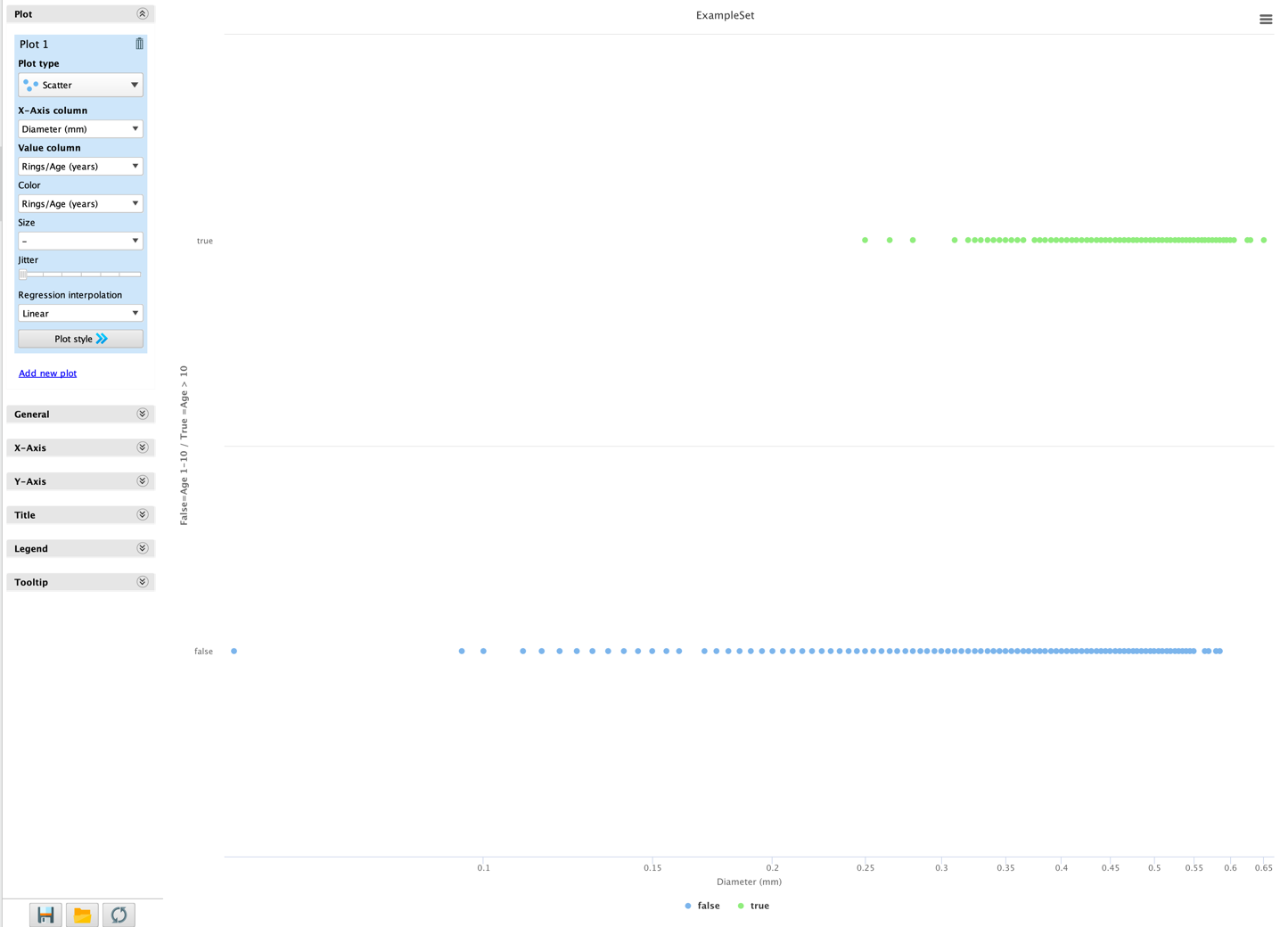
Below is a sample output for the abalone data using the multiple linear regression model.



Performance Data from RapidMiner using a logistic regression analysis on a binary abalone age

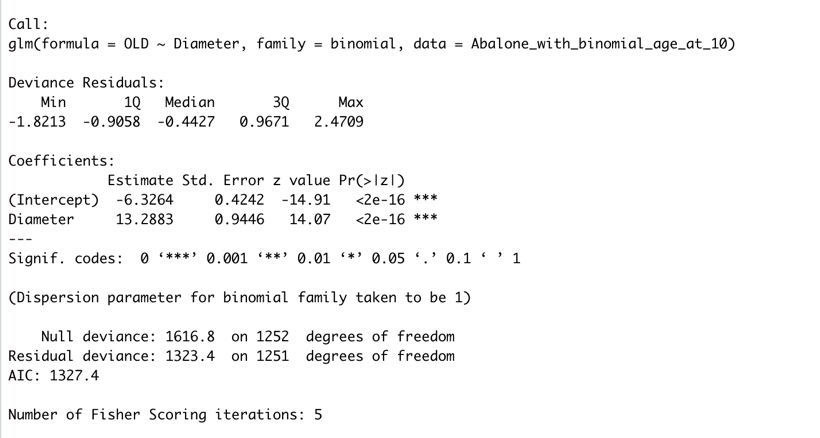
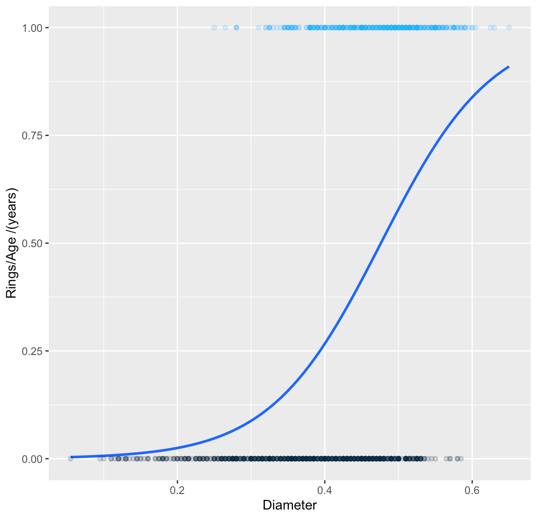


Logistic Regression scatterplot from RapidMiner

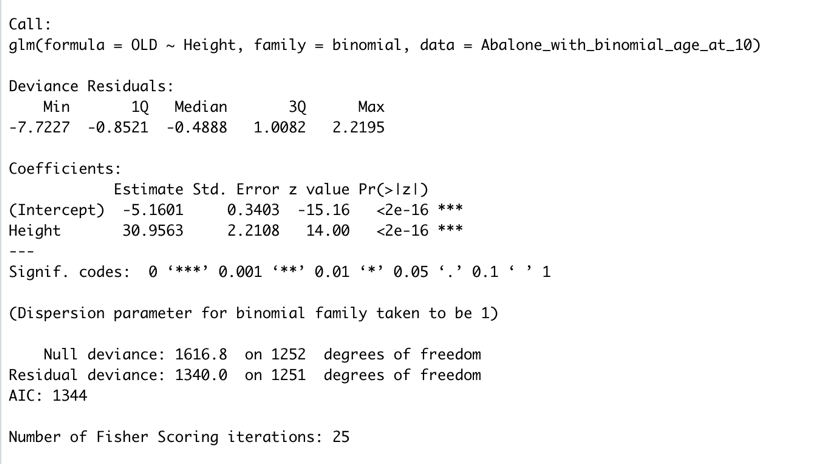
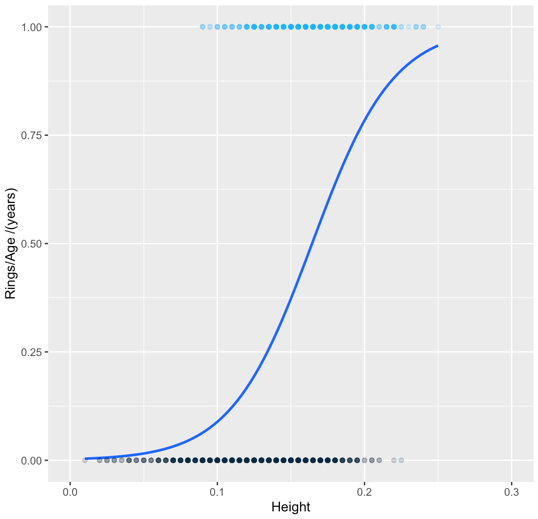


R Logistic Regression Analysis Script, Summary and Graphs ([back to 5a](#Para_TRresults))

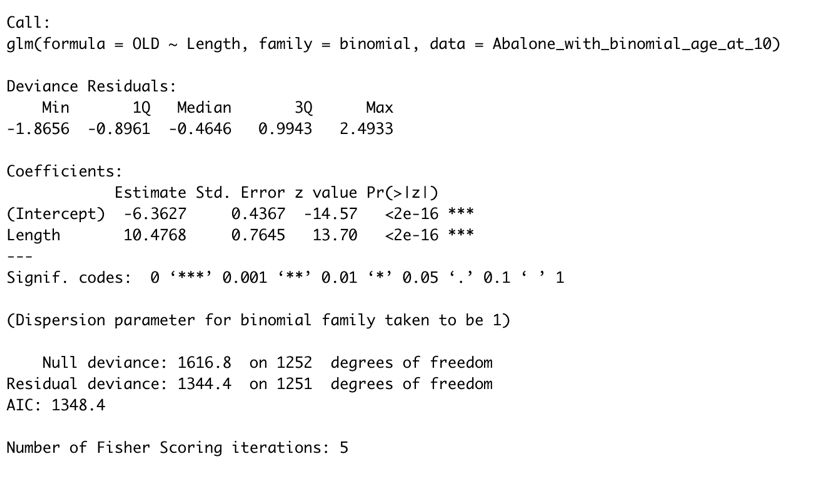
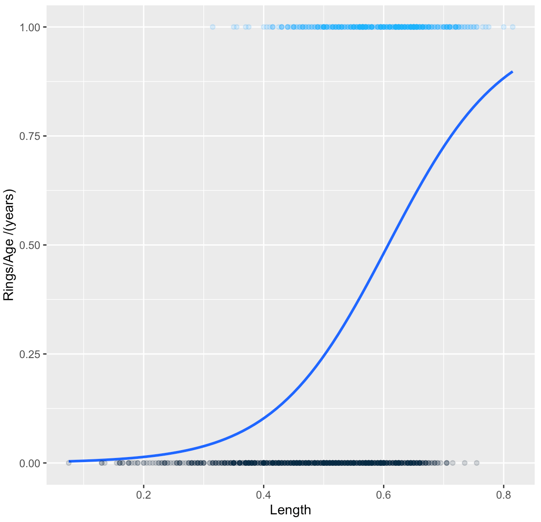
Diameter to Age



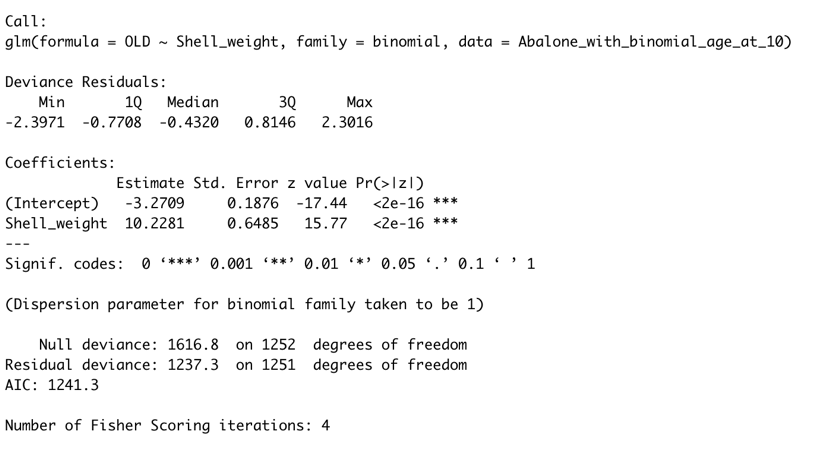
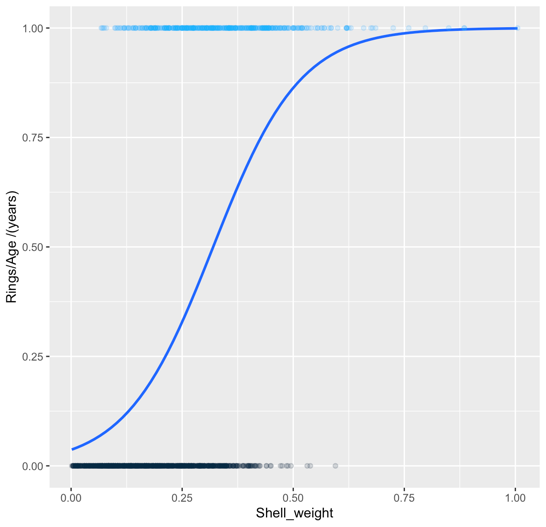
Height to Age



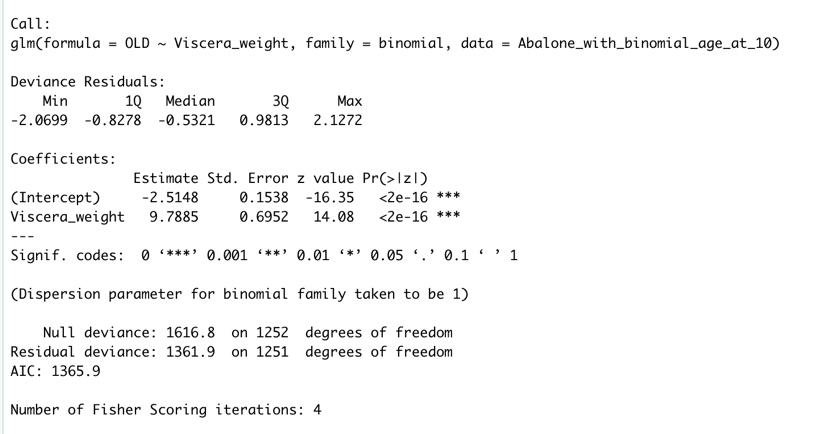
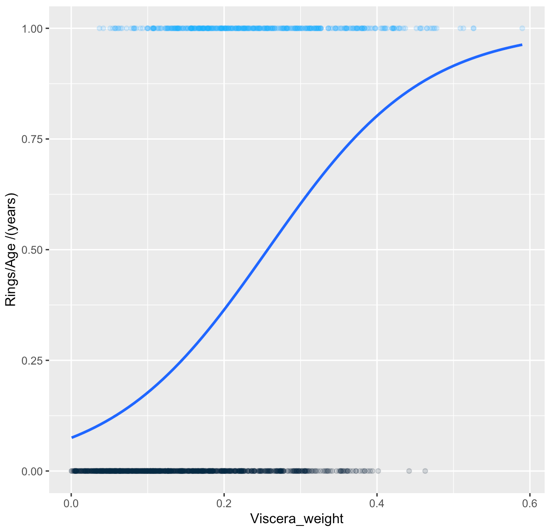
Length to Age



Shell Weight to Age



Viscera Weight to Age



Whole Weight to Age

