Frida Muedsam
Introduction to Data Science
September 7th, 2020

Introducing Data Science:

According to Alivia Smith from Data Iku, there are seven universal steps to successfully conducting a data science project. Step 1: Understand the Business, Step 2: Get Your Data, Step 3: Explore and Clean Your Data, Step 4: Enrich Your Dataset, Step 5: Build Helpful Visualizations, Step 6: Get Predictive, Step 7: Iterate, Iterate, Iterate. These past two week I have mainly explored the first two steps- understanding the topic and retrieving data. ‘

Step 1- Data is only as good as the questions you ask, so it is important to begin with figuring out what you really want to know. A very specific question would provide a more valuable answer. By effectively asking the right questions, it is much easier to define your next strategy in data analysis (Durcevic, 2020). This step can also include conceptualization and operationalization. Conceptualization includes giving concepts explicit definitions and further elaborating on what we really mean by them, and operationalization is actually measuring the concept.

Step 2- Getting Data: There are many different ways to retrieve data for your project. Common sources include databases, logs, web servers, API's, and online repositories. During this step, it is important to consider the level of measurement in your variables. I have created and attached a table which defines the different types of variables and plan on referring back to it when exploring data sources (Durcevic, 2020).

Variable Type	Nominal	Ordinal	Discrete	Continuous
Definition	Categorical data that cannot be ordered or ranked.	Categorical data that can be ordered.	Numerical data with definite number of possibilities.	Numerical data with infinite numerical possibilities.
Example	Language, race, hair color.	Level of education.	Number of students in a class, how many days of rain last year.	A person's height, temperature, weight of a new born baby.

Utilizing descriptive statistics in Data Science:

Source: Khan Academy, "Statistics: Descriptive Statistics."

Three common tools used when describing or commenting on graphs or data include mean, median, and mode.

Mean:

The average value in a set of numbers. The mean follows the skew of the graph (seen below).

Equation given by: $\overline{X} = \frac{\sum_{i=1}^{n} x_i}{n}$

This equation is representing the sum of all the variables divided by the total number of variables (n).

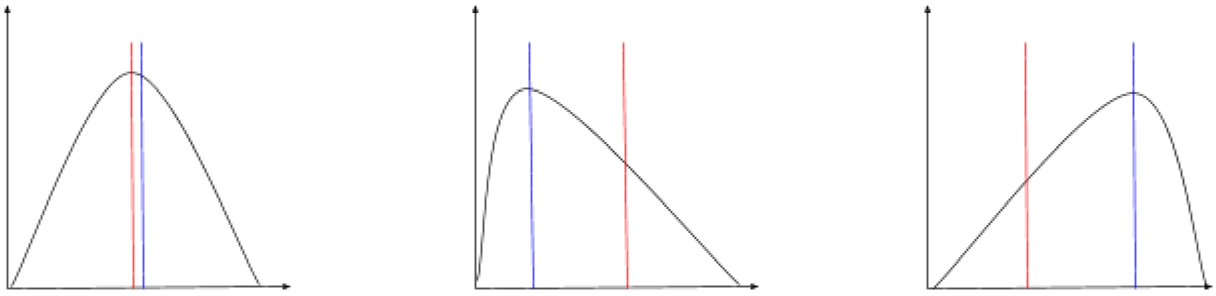
Median:

The median is the value that separates the bottom half of the data from the top. This is a preferred method of measurement because it is resistant to outliers.

Mode:

The number that appears most often.

*mean (blue line) and median (red line) are displayed on the graphs demonstrating that the mean follows skew and the resistance of median



Variance:

Variance describes how dispersed a set of data is.

Variance is given by the equation: $S^2 = \frac{\sum(x_i - \bar{x})^2}{n-1}$

Similarly, standard deviation is calculated by taking the square root of the variance.

Practicing with R (Factors):

Factors are used in R to store variables composed of categorical data. Some examples of good categorical variables to use include gender, race, and hometown.

*The code below is from a practice session I did using the website *DataCamp,* I referenced my notes and https://www.gastonsanchez.com/r4strings/chars.html to complete the code.

Example 1: Create a vector that contains all the observations that belong the survey of gender of a set of 5 different individuals.

1) Assign to variable theory the value "factors for categorical variables"

```
theory <- "factors for categorical variables"
```

2) Convert the character vector gender_vector to a factor with factor() and assign the result to factor_gender_vector. Print out factor_gender_vector and assert that R prints out the factor levels below the actual values.

```
# gender vector
gender_vector <- c("Male", "Female", "Female", "Male", "Male")

# Convert gender_vector to a factor
factor_gender_vector <-factor(gender_vector)
```

```
# Print out factor_gender_vector
print(factor_gender_vector)
```

3) Change the factor levels of factor_survey_vector to c("Female", "Male"). Mind the order of the vector elements here.

```
# Code to build factor_survey_vector
survey_vector <- c("M", "F", "F", "M", "M")
factor_survey_vector <- factor(survey_vector)

# Specify the levels of factor_survey_vector
levels(factor_survey_vector) <- c("Female","Male")
```

4) Ask a **summary()** of the survey_vector and factor_survey_vector. Interpret the results of both vectors. Are they both equally useful in this case?

```
# Build factor_survey_vector with clean levels
survey_vector <- c("M", "F", "F", "M", "M")
factor_survey_vector <- factor(survey_vector)
levels(factor_survey_vector) <- c("Female", "Male")
factor_survey_vector

# Generate summary for survey_vector
summary(survey_vector)
# Generate summary for factor_survey_vector
summary(factor_survey_vector)
```

***identifying "Male" and "Female" as factor levels in factor_survey_vector enables R to show the number of elements for each category.*

5) Read the code in the editor to test if male is greater than (>) female.

Since "Male" and "Female" are nominal factor levels, R returns a warning message, therefore R is not able to compare the two variables

Example 2: Let us say that you are leading a research team of five data analysts and that you want to evaluate their performance. To do this, you track their speed, evaluate each analyst as "slow", "medium" or "fast."

1) Assign speed_vector a vector with 5 entries, one for each analyst. Each entry should be either "slow", "medium", or "fast"

```
# Create speed_vector

speed_vector <- c("medium", "slow", "slow", "medium", "fast")
```

2) From speed_vector, create an ordered factor vector: factor_speed_vector. Set ordered to TRUE, and set levels to c("slow", "medium", "fast").

```
# Create speed_vector

speed_vector <- c("medium", "slow", "slow", "medium", "fast")

# Convert speed_vector to ordered factor vector

factor_speed_vector <-
factor(speed_vector,ordered=TRUE,levels=c("slow", "medium",
"fast"))

levels(factor_speed_vector)

# Print factor_speed_vector

factor_speed_vector

summary(factor_speed_vector)
```

3) Use [2] to select from factor_speed_vector the factor value for the second data analyst. Store it as da2. Use [5] to select the factor_speed_vector factor value for the fifth data analyst. Store it as da5. Compare values.

```
# Create factor_speed_vector

speed_vector <- c("medium", "slow", "slow", "medium", "fast")
```

```
factor_speed_vector <- factor(speed_vector, ordered = TRUE,
levels = c("slow", "medium", "fast"))

# Factor value for second data analyst

da2 <- factor_speed_vector[2]

# Factor value for fifth data analyst

da5 <-factor_speed_vector[5]

# Is data analyst 2 faster than data analyst 5?

da2>da5
```

Works Cited:

10 Steps For Asking The Right Data Analysis Questions. (2020, August 17). Retrieved September 07, 2020, from https://www.datapine.com/blog/data-analysis-questions/

Khan, S. (Director). (2009). *Statistics: The average | Descriptive statistics | Probability and Statistics | Khan Academy*[Video file]. Retrieved 2020, from https://www.youtube.com/watch?v=uhxtUt_-GyM

Smith, A. (n.d.). 7 Fundamental Steps to Complete a Data Analytics Project. Retrieved September 07, 2020, from https://blog.dataiku.com/2019/07/04/fundamental-steps-data-project-success

Gaston Sanchez with contributions from Chitra Venkatesh. (n.d.). Handling Strings with R. Retrieved September 07, 2020, from https://www.gastonsanchez.com/r4strings/chars.html

Wickham, H., & Grolemund, G. (2017). *R for data science: Import, tidy, transform, visualize and model data*. Beijing: O'Reilly.

Data Camp- Factors. (2020). Retrieved September 07, 2020, from https://learn.datacamp.com/