Data cleansing and wrangling with Diabetes.csv data set

Shiloh Bradley
Webster University St. Louis

Executive Summary

Through data wrangling, data is prepared for further analysis techniques. While highly important for analysis and data integrity, wrangling and cleansing can take anywhere from 50% to 80% of the a data scientist's time. There are numerous techniques that can be applied, but the focus was on using the dplyr and tidyr packages in R Studio. There are dozens of techniques in both of the packages that are used to tidy data, but a few were specifically used and one was used to run the experiment. dplyr techniques that were applied include distinct(), sample_frac(), and group_by(). The tidyr packages that were explored were not applicable to this data set and therefore not used in the experiment.

Table 1: Summary of Characterization before tests

```
> summary(d)
  Pregnancies      PG.Concentration  Diastolic.BP     Tri.Fold.Thick    Serum.Ins
 Min.   : 0.000   Min.   :  0.0    Min.   :  0.00   Min.   : 0.00   Min.   :  0.0
 1st Qu.: 1.000   1st Qu.: 99.0    1st Qu.: 62.00   1st Qu.: 0.00   1st Qu.:  0.0
 Median : 3.000   Median :117.0    Median : 72.00   Median :23.00   Median : 30.5
 Mean   : 3.845   Mean   :120.9    Mean   : 69.11   Mean   :20.54   Mean   : 79.8
 3rd Qu.: 6.000   3rd Qu.:140.2    3rd Qu.: 80.00   3rd Qu.:32.00   3rd Qu.:127.2
 Max.   :17.000   Max.   :199.0    Max.   :122.00   Max.   :99.00   Max.   :846.0
 NA's   :3        NA's   :3        NA's   :3        NA's   :3       NA's   :3
      BMI         DP.Function          Age            Diabetes
 Min.   : 0.00   Min.   :0.0780   Min.   :21.00          :  3
 1st Qu.:27.30   1st Qu.:0.2437   1st Qu.:24.00   Healthy:500
 Median :32.00   Median :0.3725   Median :29.00   Sick   :268
 Mean   :31.99   Mean   :0.4719   Mean   :33.24
 3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00
 Max.   :67.10   Max.   :2.4200   Max.   :81.00
 NA's   :3       NA's   :3        NA's   :3
```

```
> summary(sample_frac(d, 0.5, replace = TRUE))
  Pregnancies      PG.Concentration  Diastolic.BP    Tri.Fold.Thick   Serum.Ins          BMI        DP.Function         Age             Diabetes
 Min.   : 0.000   Min.   :  0.0    Min.   :  0.00   Min.   : 0.0   Min.   :  0.00   Min.   : 0.0   Min.   :0.0850   Min.   :21.00          :  1
 1st Qu.: 1.000   1st Qu.: 97.0    1st Qu.: 64.00   1st Qu.: 0.0   1st Qu.:  0.00   1st Qu.:26.5   1st Qu.:0.2350   1st Qu.:24.00   Healthy:257
 Median : 3.000   Median :114.0    Median : 70.00   Median :22.0   Median : 23.00   Median :31.2   Median :0.3830   Median :29.00   Sick   :128
 Mean   : 3.948   Mean   :117.5    Mean   : 68.29   Mean   :19.7   Mean   : 77.92   Mean   :31.4   Mean   :0.4415   Mean   :33.51
 3rd Qu.: 6.000   3rd Qu.:135.0    3rd Qu.: 80.00   3rd Qu.:32.0   3rd Qu.:126.00   3rd Qu.:36.5   3rd Qu.:0.6010   3rd Qu.:41.00
 Max.   :14.000   Max.   :199.0    Max.   :110.00   Max.   :99.0   Max.   :543.00   Max.   :57.3   Max.   :1.6980   Max.   :72.00
 NA's   :1        NA's   :1        NA's   :1        NA's   :1      NA's   :1        NA's   :1      NA's   :1        NA's   :1
>
```

Table 2: sample_frac() test 1

Table 3: sample_frac() test 2

```
> summary(sample_frac(d, 0.5, replace = TRUE))
  Pregnancies      PG.Concentration  Diastolic.BP    Tri.Fold.Thick   Serum.Ins          BMI        DP.Function         Age             Diabetes
 Min.   : 0.000   Min.   :  0.0    Min.   :  0.00   Min.   : 0.00   Min.   :  0.00   Min.   : 0.00   Min.   :0.0780   Min.   :21.00          :  2
 1st Qu.: 1.000   1st Qu.: 97.0    1st Qu.: 62.00   1st Qu.: 0.00   1st Qu.:  0.00   1st Qu.:26.40   1st Qu.:0.2370   1st Qu.:24.00   Healthy:231
 Median : 3.000   Median :115.5    Median : 70.00   Median :22.00   Median :  0.00   Median :31.20   Median :0.3450   Median :29.00   Sick   :153
 Mean   : 3.826   Mean   :120.2    Mean   : 68.06   Mean   :19.59   Mean   : 79.17   Mean   :31.54   Mean   :0.4709   Mean   :33.49
 3rd Qu.: 6.000   3rd Qu.:141.0    3rd Qu.: 80.00   3rd Qu.:32.00   3rd Qu.:125.25   3rd Qu.:36.30   3rd Qu.:0.6400   3rd Qu.:41.00
 Max.   :15.000   Max.   :199.0    Max.   :122.00   Max.   :52.00   Max.   :600.00   Max.   :57.30   Max.   :2.1370   Max.   :81.00
 NA's   :2        NA's   :2        NA's   :2        NA's   :2       NA's   :2        NA's   :2       NA's   :2        NA's   :2
>
```

```
> summary(sample_frac(d, 0.5, replace = TRUE))
  Pregnancies      PG.Concentration  Diastolic.BP    Tri.Fold.Thick   Serum.Ins          BMI        DP.Function         Age             Diabetes
 Min.   : 0.000   Min.   :  0.0    Min.   :  0.00   Min.   : 0.00   Min.   :  0.00   Min.   : 0.00   Min.   :0.0780   Min.   :21.00          :  2
 1st Qu.: 1.000   1st Qu.: 97.0    1st Qu.: 62.00   1st Qu.: 0.00   1st Qu.:  0.00   1st Qu.:26.40   1st Qu.:0.2370   1st Qu.:24.00   Healthy:231
 Median : 3.000   Median :115.5    Median : 70.00   Median :22.00   Median :  0.00   Median :31.20   Median :0.3450   Median :29.00   Sick   :153
 Mean   : 3.826   Mean   :120.2    Mean   : 68.06   Mean   :19.59   Mean   : 79.17   Mean   :31.54   Mean   :0.4709   Mean   :33.49
 3rd Qu.: 6.000   3rd Qu.:141.0    3rd Qu.: 80.00   3rd Qu.:32.00   3rd Qu.:125.25   3rd Qu.:36.30   3rd Qu.:0.6400   3rd Qu.:41.00
 Max.   :15.000   Max.   :199.0    Max.   :122.00   Max.   :52.00   Max.   :600.00   Max.   :57.30   Max.   :2.1370   Max.   :81.00
 NA's   :2        NA's   :2        NA's   :2        NA's   :2       NA's   :2        NA's   :2       NA's   :2        NA's   :2
>
```

Table 4: sample_frac() test 3

Problem Description

The data set that was tested was the Diabetes.csv data set which was collected from the Pima Indians Diabetes Database. There are 768 instances classified under nine different attributes which included the following: pregnancies (number of pregnancies), PG Concentration (plasma glucose at 2 hours in an oral glucose tolerance test), Diastolic BP (Diastolic Blood Pressure (mm Hg)), Tri Fold Thick (Triceps Skin Fold Thickness (mm)), Serum Ins (2-Hour Serum Insulin (mu U/ml)), BMI (Body

Mass Index: (weight in kg/ (height in m)^2)), DP Function (Diabetes Pedigree Function), Age (years), and Diabetes (whether or not the person has diabetes).

This data set is considered “messy”, meaning that it has missing and inaccurate data, where inaccurate data is defined, such as blatant biological levels that are not possible for a human being to sustain life at (i.e. too low/high of heart rate or blood pressure). Because of this, the primary focus was on data wrangling and cleaning and how that affects the character (statistical deviation) of the data. Statistical deviation meaning how the mean, median, mode, average, or upper and lower quartiles changed from the original, unaltered data set.

An additional aspect considered in the experiment was the element of “cost”, including time and potential financial cost for a company. “It is often said that 80% of data analysis is spent on the process of cleaning and preparing the data” (Dasu, T., Johnson, T.).

Analysis Technique

My hypothesis was that the character of the data[1] would not be changed much by removing extraneous or unnecessary data. If anything, the result of any tests ran on the cleansed data should be more concise and accurate. Any information that does not need to be included, would not be and therefore the results of any tests (that could be ran after cleansing techniques are applied) should find more consistent values closer and be to the curve and not have so many random outliers. By eliminating outliers, I

[1] Character of the data: Summary of the statistics, example: mean, median, mode, average, etc.

would be able to tell true trends and not have the data skewed. This should help with data integrity. I also hypothesize that this would be financially costly.

The focus of the experiment was on extracting certain points/ lines of data and seeing if the removal affects the character of the data, meaning that if I remove or change X many lines of data, will the character of the data be the same or is it even affected? Additionally, the two main goals of data cleansing/ wrangling are: “1 Make data suitable to use with 1 a particular piece of software 2 Reveal information” (Grolemund, G.).

This was accomplished through R Studio by the tidyr and dplyr packages and using data cleansing, editing, and pre-processing techniques. These are all reasonable steps that were used to prepare data for computation and analysis and are noted as fundamental functions for cleaning, processing, and manipulating data. Each one was ran individually, like in a lab experiment, where each test would be the independent variable (Boehmke, B.). Analyzing each isolated technique demonstrates the effect on the character of the data. This analysis also further emphasizes the no free lunch theorem, in that every technique will not have the same effect or even work with the given data set.

To include some examples of each, cleansing can be done through Tidy Data in tidyr which is a foundation to data wrangling. “In a tidy data set: Each variable is saved in its own column, each observation is saved in its own row and each "type" of observation stored 3 in a single table ” (Grolemund). This is one way to organize data and make it visually easier. It essentially condenses the data down into compartments,

similar to clustering where similar data is grouped together, however it differs in that Tidy Data is not based on k-distance. The four fundamental functions are as follows: gather() takes multiple columns, and gathers them into key-value pairs: it makes “wide” data longer, spread() takes two columns (key & value) and spreads into multiple columns, it makes “long” data wider, separate() splits a single column into multiple columns, unite() combines multiple columns into a single column (Boehmke).

Another manipulation technique is through the dplyr package in R that allows the analyst to use seven fundamental functions to transform the data. The analyst can mold the data set into something workable and less abstract and messy. The seven fundamental functions in dplyr include the following: select() for selecting variables, filter() which provides basic filtering capabilities, group_by() which groups data by categorical levels, summarise() to summarise data by functions of choice, arrange() for ordering data, join() for joining separate data frames, and mutate() to create new variables (Boehmke).

There are several other functions that can be used, but I focused on three from dplyr and none from tidyr because the ones from tidyr did not apply. (No free lunch.) The three from dplyr that I used were distinct(), sample_frac(), and group_by(). distinct() is used to remove duplicate rows. The result was that it did not change the character of the data. sample_frac() was used to randomly select a fraction of the rows and it did change the character of the data each time that it was ran. group_by() was used to group data by the same variable and it also did not change the character of the data.

For comparison, **Table 1** contains the summary of characterization before any tests have been ran on the Diabetes.csv data set. In **Table 2**, **Table 3**, and **Table 4**, they summarize three randomized tests using the sample_frac() function. Additionally, no tables regarding distinct() and group_by() were included since they did not change the character. When looking at the sample_frac() results, we can see that the character is different each time. This is due to half of the data being randomly selected each time and being used to represent the set. This technique is not one that I would recommend for data cleansing in the medical field considering that the data is randomized and can cause discrepancies and inconsistencies.

Results

Table 1: Summary of Characterization before tests

```
> summary(d)
  Pregnancies      PG.Concentration  Diastolic.BP     Tri.Fold.Thick    Serum.Ins
 Min.   : 0.000   Min.   :  0.0    Min.   :  0.00   Min.   : 0.00   Min.   :  0.0
 1st Qu.: 1.000   1st Qu.: 99.0    1st Qu.: 62.00   1st Qu.: 0.00   1st Qu.:  0.0
 Median : 3.000   Median :117.0    Median : 72.00   Median :23.00   Median : 30.5
 Mean   : 3.845   Mean   :120.9    Mean   : 69.11   Mean   :20.54   Mean   : 79.8
 3rd Qu.: 6.000   3rd Qu.:140.2    3rd Qu.: 80.00   3rd Qu.:32.00   3rd Qu.:127.2
 Max.   :17.000   Max.   :199.0    Max.   :122.00   Max.   :99.00   Max.   :846.0
 NA's   :3        NA's   :3        NA's   :3        NA's   :3       NA's   :3
      BMI         DP.Function          Age           Diabetes
 Min.   : 0.00   Min.   :0.0780   Min.   :21.00          :  3
 1st Qu.:27.30   1st Qu.:0.2437   1st Qu.:24.00   Healthy:500
 Median :32.00   Median :0.3725   Median :29.00   Sick   :268
 Mean   :31.99   Mean   :0.4719   Mean   :33.24
 3rd Qu.:36.60   3rd Qu.:0.6262   3rd Qu.:41.00
 Max.   :67.10   Max.   :2.4200   Max.   :81.00
 NA's   :3       NA's   :3        NA's   :3
```

```
> summary(sample_frac(d, 0.5, replace = TRUE))
  Pregnancies      PG.Concentration  Diastolic.BP    Tri.Fold.Thick   Serum.Ins          BMI        DP.Function         Age             Diabetes
 Min.   : 0.000   Min.   :  0.0    Min.   :  0.00   Min.   : 0.0   Min.   :  0.00   Min.   : 0.0   Min.   :0.0850   Min.   :21.00          :  1
 1st Qu.: 1.000   1st Qu.: 97.0    1st Qu.: 64.00   1st Qu.: 0.0   1st Qu.:  0.00   1st Qu.:26.5   1st Qu.:0.2350   1st Qu.:24.00   Healthy:257
 Median : 3.000   Median :114.0    Median : 70.00   Median :22.0   Median : 23.00   Median :31.2   Median :0.3830   Median :29.00   Sick   :128
 Mean   : 3.948   Mean   :117.5    Mean   : 68.29   Mean   :19.7   Mean   : 77.92   Mean   :31.4   Mean   :0.4415   Mean   :33.51
 3rd Qu.: 6.000   3rd Qu.:135.0    3rd Qu.: 80.00   3rd Qu.:32.0   3rd Qu.:126.00   3rd Qu.:36.5   3rd Qu.:0.6010   3rd Qu.:41.00
 Max.   :14.000   Max.   :199.0    Max.   :110.00   Max.   :99.0   Max.   :543.00   Max.   :57.3   Max.   :1.6980   Max.   :72.00
 NA's   :1        NA's   :1        NA's   :1        NA's   :1      NA's   :1        NA's   :1      NA's   :1        NA's   :1
>
```

Table 2: sample_frac() test 1

Table 3: sample_frac() test 2

```
> summary(sample_frac(d, 0.5, replace = TRUE))
  Pregnancies      PG.Concentration  Diastolic.BP    Tri.Fold.Thick   Serum.Ins          BMI        DP.Function         Age             Diabetes
 Min.   : 0.000   Min.   :  0.0    Min.   :  0.00   Min.   : 0.00   Min.   :  0.00   Min.   : 0.00   Min.   :0.0780   Min.   :21.00          :  2
 1st Qu.: 1.000   1st Qu.: 97.0    1st Qu.: 62.00   1st Qu.: 0.00   1st Qu.:  0.00   1st Qu.:26.40   1st Qu.:0.2370   1st Qu.:24.00   Healthy:231
 Median : 3.000   Median :115.5    Median : 70.00   Median :22.00   Median :  0.00   Median :31.20   Median :0.3450   Median :29.00   Sick   :153
 Mean   : 3.826   Mean   :120.2    Mean   : 68.06   Mean   :19.59   Mean   : 79.17   Mean   :31.54   Mean   :0.4709   Mean   :33.49
 3rd Qu.: 6.000   3rd Qu.:141.0    3rd Qu.: 80.00   3rd Qu.:32.00   3rd Qu.:125.25   3rd Qu.:36.30   3rd Qu.:0.6400   3rd Qu.:41.00
 Max.   :15.000   Max.   :199.0    Max.   :122.00   Max.   :52.00   Max.   :600.00   Max.   :57.30   Max.   :2.1370   Max.   :81.00
 NA's   :2        NA's   :2        NA's   :2        NA's   :2       NA's   :2        NA's   :2       NA's   :2        NA's   :2
>
```

Table 4: sample_frac() test 3

Issues

One issue that came into play, which I expect I'll continually see, is that the tidyr and dplyr packages were not compatible with the version of R Studio that was available to me. In order to run the packages, I had to update the software. Additionally, this could work in reverse that now that I have updated the software, there may be packages that

could be obsolete now. Essentially, technology is changing quickly and there are consistently new data analysis techniques that can be applied.

Appendix

Code

```
# Final research report

# MATH 3210

# Data Wrangling with R


# Retrieve Diabetes.csv data set
d = read.csv('C:/Users/WUStudent/Downloads/Diabetes.csv')


# Load dplyr
# Load tidyr


# Inspect Data Set
head(d)


# Characterization of the data
summary(d)


# dplyr tests

# Remove duplicate rows
distinct(d)
# view characteristics after each test
summary(distinct(d))

#Randomly select fraction of the rows
sample_frac(d, 0.5, replace = TRUE)
summary(sample_frac(d, 0.5, replace = TRUE))

sample_frac(d, 0.5, replace = TRUE)
summary(sample_frac(d, 0.5, replace = TRUE))

sample_frac(d, 0.5, replace = TRUE)
summary(sample_frac(d, 0.5, replace = TRUE))

group_by(d, Diastolic.BP)
summary(group_by(d, Diastolic.BP))

group_by(d, Pregnancies)
summary(group_by(d, Pregnancies))

between(40, 126)
```

```
filter_(Diastolic.BP, 40)

Diabetes %>% filter(Diastolic.BP >= 40)

# tidyr tests

spread(0, Diastolic.BP, PG.Concentration)

# view characteristics after each test
summary(d)
```

References

Boehmke, B. (2014). *Data Processing with dplyr & tidyr.* Retrieved from RPub.com.

Dasu, T., Johnson, T. (2003). *Exploratory Data Mining and Data Cleaning.* Wiley.

Data Wrangling with dplyr and tidyr. Retrieved from RStudio.com

Grolemund, G. (2015). *Data Wrangling with R: How to work with the structures of your data.* Retrieved from RStudio.com.

Vaidyanathan, R. *Data Wrangling I.* Retrieved from ramnathv.github.io.