Data Cleansing

As many of you have heard the expression before, “put good in, get good out,” this phrase holds true for data science. Data collection and cleansing is extremely important for having accurate, accountable data that will explain trends or variances in the real world. For instance, if you are collecting data for many companies for many years and are missing data, are you going to keep that data that is missing variables or are you going to delete it?

Data cleansing is one of the most important pieces of data science. According to Forbes, data cleansing and organizing consists of 60% of the time spent by data scientists. Another 19% of the total time spent by data scientists consists of collecting data sets.In addition these two tasks were voted as the least liked tasks by data scientists at 57% and 21% respectively. (<https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/#690b6416f637>).

Data Cleansing Steps

WHO data on obesity rates

<http://apps.who.int/gho/data/view.main.CTRY2450A?lang=en>

PPP International $

Explain why data is good or not

GDP does not account for some industries

Obesity could be due to other factors(race, region)

Datathon example with pharmaceutical companies

-Show different types of downloads similarity

-Download as csv

-Delete unnecessary rows from the top (2 and 3)

-Find & Select under home tab

-Find All “male” (this will include female as well)

-Close

-Delete sheet columns

-Delete both sexes row (now a constant)

-Open new sheet

=LEFT function

=LEFT(‘xmart’!B3,4)

-Insert new row at top

-copy and paste in years

-insert new row at side

-copy and paste countries

-Alternatively can do the previous two steps first or add insert a step down

-Show how find and replace does not select ‘No d’

-Save file as excel workbook

-Make sure to move the workbook you want displayed to the front

-Right click on workbook you want to move to end

-Do the same for “No d values but change the look function so it looks for values

GGPLOT and Testing

<https://www.kaggle.com/uciml/forest-cover-type-dataset/data>

graphname <- ggplot(data, aes(x variable, y variable)) +

labs(title=”title”) + geom type()

Standard histogram

graph1<- ggplot(cover\_type\_forest, aes(Elevation)) + labs(title = "Elevation") + geom\_histogram()

Graph1

graphtest1 <- ggplot(cover\_type\_forest, aes((Slope))

graphtest1 <- graphtest1 +geom\_histogram()

graphtest1

#Trying log on aes(slope) makes weird value

graph2 <- ggplot(cover\_type\_forest, aes(Elevation, Slope)) + labs(title="Elevation and Slope") + geom\_point()

graph2

(Above sucks)

install.packages

library(ggthemes)

#Heat map

graphheat <- ggplot(cover\_type\_forest, aes( Elevation, Slope)) + geom\_bin2d(bins=4.5) + scale\_fill\_gradient(low = "white", high = "steelblue") + labs(title = "Heatmap Elevation versus Slope", subtitle = "1 = low, 5 = high") + theme\_tufte() + labs(x = "Elevation")+ labs(y = "Slope" )

graphheat

#alpha alters transparency of the data

scatter1<-ggplot(assignment4, aes(gdp\_log , PHYSINT))

scatter1<-scatter1 + geom\_point(alpha=0.3) + geom\_smooth(method = "loess", )

scatter1

#binhex command

#install hexbin package

#formula is for trend line

#hexagonal heat map

scatter2<-ggplot(assignment4, aes(gdp\_log , democ))

scatter2<-scatter2 + stat\_bin\_hex() + stat\_smooth(method = "lm", formula = y ~ poly(x, 2), size = 1)

scatter2

#density plot

scatter3<-ggplot(assignment4, aes(PHYSINT , democ))

scatter3<-scatter3 + stat\_density\_2d() + geom\_smooth(method = "glm")

scatter3

#Need scatterplot3d package

# note you can also do 3dbarplotsd

# the arguments are x,y, and z variable.

# The angle changes the way you look at it

# You will need the scatterplot3d package to do this

# angle adjust the angle of the graph. Alter it to look at the graph from different angles

# highlight.3d=TRUE creates the red and black color to tell you where in the graph the dots are

scatterplot3d(assignment4$gdp\_log, assignment4$democ, assignment4$PHYSINT, angle=75, highlight.3d = TRUE, length(1))

linearMod <- **lm**(dist ~ speed, data=cars) *# build linear regression model on full data*  
**print**(linearMod)

Linear regression <- lm(y ~ x1 + x2…., data = my data)

summary  
*#> Call:*  
*#> lm(formula = dist ~ speed, data = cars)*  
*#>*   
*#> Coefficients:*  
*#> (Intercept) speed*   
*#> -17.579 3.932*

#Need summary statistics to see if it is a good fit

**summary**(linearMod) *# model summary*  
*#> Call:*  
*#> lm(formula = dist ~ speed, data = cars)*  
*#>*   
*#> Residuals:*  
*#> Min 1Q Median 3Q Max*   
*#> -29.069 -9.525 -2.272 9.215 43.201*   
*#>*   
*#> Coefficients:*  
*#> Estimate Std. Error t value Pr(>|t|)*   
*#> (Intercept) -17.5791 6.7584 -2.601 0.0123 \**   
*#> speed 3.9324 0.4155 9.464 1.49e-12 \*\*\**  
*#> ---*  
*#> Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1*  
*#>*   
*#> Residual standard error: 15.38 on 48 degrees of freedom*  
*#> Multiple R-squared: 0.6511, Adjusted R-squared: 0.6438*   
*#> F-statistic: 89.57 on 1 and 48 DF, p-value: 1.49e-12*

