# Statistics Lecture 1

-------------------------------------------------------------------------------------------------------------------

# Explanation of R and Gitbash

# 1. Installing R Studio (or even better the entire Anaconda Navigator)

# 2. Downloading Gitbash \

# 3. Creating A Github Acount by signing up for GitHub and cliclking the "New" Tile

# Some useful Gitbash Commands

# pwd #prints current working directory

# cd # changes the working directory

# ls # lists the items in the working direcory

# mkdir # creates a new folder in the working directory

# touch # creates a test file

# rm # stands for remove, be very carefull, as removed files cannot be retrieved

# mv # Move files between direcories

# echo # Echos whatever argument is provided

# date # Gives you the current date

# Linking your GitHub account to your working directory on your local machine

# git config --global user.name "Your Name"

# git config --global user.email "Your E-mail"

# git config --list #See all changes you made

# mkdir ~/Data\_Science

# git init

# git remote add origin "URL of Git Repo"

# Synching your local repo to the GitHub repo

# git add . - adds new files

# git add -U # Updates tracking of files

# git add # does both

# git commit -m "This is an addition I made to my repo"

# git pull # pull the status of the online repo and check if it is in sync with the last update

# git push # push your repo to Gitup .... and DONE ;)

-------------------------------------------------------------------------------------------------------------------

# In R Studio Setting the working directory

getwd() # See which directory you are currently in

setwd("C:/Users/awaldert/Desktop/Data\_Science") # Set your Working Directory to your desired folder

# But first we need to load some tools for later ...

library(dplyr)

library(rpart)

library(ggplot2)

library(caTools)

library(pdftools)

library(plotrix)

# The data science process: "https://towardsdatascience.com/5-steps-of-a-data-science-project-lifecycle-26c50372b492"

# Exploratory analysis in R

data <- mtcars

head(data)

str(data)

summary(data)

mean(data$mpg)

var(data$hp)

sqrt(var(data$hp))

hp <- as.vector(unlist(data$hp))

mpg <- as.vector(unlist(data$mpg))

cyl <- as.vector(unlist(data$cyl))

wt <- as.vector(unlist(data$wt))

plot(cyl, mpg)

plot(wt, mpg)

regression <- lm(mpg ~ hp + cyl + wt)

summary(regression)

plot(hp, mpg)

cor(hp, mpg)

# Plotting in R

# Setting up a linear model in R

# The Laws on Probability

# Probability = Number of desired outcomes/Number of possible outomces

# Hence the probability getting a 2 from a die roll is 1/6

# Probability of event A and B which are not dependent (in odher words they are independent)

# Indenpendent means that the outcome of event A, has no impact on the probability of B occurring

# Example Two coins being tossed

p.head <- 0.5

p.tail <- 0.5

# What is the probability of tossing a coin and getting a) two heads b) 4 tails and c) 10 heads

p.head^2

p.head^4

p.head^10

# Proving that I am correct in a) ...

p.head <- rbinom(100000, 1, .5) #You can chose different probabilites as you please

p.tail <- rbinom(100000, 1, .5) #You can chose different probabilites as you please

mean(p.head & p.tail)

# Probability of event A or B which are not dependent

# You toss a coin two times, what is the probability of getting either heads or tail

p.head <- 0.5

p.tail <- 0.5

p <- p.head + p.tail -(p.head\*p.tail)

print(p)

# Problem: If you throw a six-sided die and then flip a coin, what is the probability that you will get either a 6 on the die or a head on the coin flip?

p.six <- 1/6

p.head <- 0.5

p <- p.head + p.six -(p.head\*p.six)

print(p)

# Problem: You are in the process of applying for our first job and have sent out two applications. You rate the probability of getting Job A to be 20% and Job B to be 60%.

# What is the probability of getting at least one of the two jobs? Note they are idependent of each other.

p.jobA <- .2

p.jobB <- .6

p.not.jobA <- .8

p.not.jobB <- .4

p <- 1-(p.not.jobB\*p.not.jobA)

print(p)

-------------------------------------------------------------------------------------------------------------------

# The Normal Distribution and Z Scores

# Generally, it is observed that the collection of random data from independent sources is distributed normally.

#We get a bell shape curve on plotting a graph with the value of the variable on the horizontal axis and the count of the values in the vertical axis.

#The centre of the curve represents the mean of the dataset.

# First let us have a look at what a normal distribution looks like:

x <- seq(0, 10, by = .1)

y <- dnorm(x, mean = 5, sd =1)

plot(x, y, main = "Normal Distribution", col = "blue", type = "l")

# Calculating Probabilities on the Normal Distribution using the slides and our calculator ;)!

# And now lets do that in R

# 1. Test scores are normally distributed with a Mean of 45% and a Standard Deviation of 6%. What is the probability that a student scores one of the following cases:

# Up to 60%

# More than 70%

# Between 60% and 80%

p <- pnorm(60, mean=45, sd=25, lower.tail = TRUE)

print(p)

p <- pnorm(70, mean=45, sd=25, lower.tail = FALSE)

print(p)

p <- pnorm(70, mean=45, sd=25, lower.tail = TRUE)-

pnorm(60, mean=45, sd=25, lower.tail = TRUE)

print(p)

# Enough with the boring stuff, let us get into some data!

-------------------------------------------------------------------------------------------------------------------

# Reading in the House Price data

data\_full <- read.csv("data.csv") # This csv file is in my working directory, to read it maje sure it is in yours and you have set your current working directory in R accordingly, and correct spelling helps

head(data\_full)

dim(data\_full)

class(data\_full)

str(data\_full)

# Descriptive Statistics

# Definition A descriptive statistic (in the count noun sense) is a summary statistic that quantitatively describes or summarizes features from a collection of information,[1]

# while descriptive statistics (in the mass noun sense) is the process of using and analysing those statistics.

# Descriptive Statistics and Measures of Location and Dispersion

# Measures of Location

mean(data\_full$price) # Average if the data

median(data\_full$price) #M iddle Point of the data in ascending order

mode(data\_full$price) #M ost frequent number

weighted.mean(data\_full$price) #The sum of all values times a weight/sum of their weights

# Measures of Dispersion

var(data\_full$price, y=NULL)

sqrt(var(data\_full$price, y=NULL)) #St.Dev

range(data\_full$price)

IQR(data\_full$price) # The interquartile range of an observation variable is the difference of its upper and lower quartiles. It is a measure of how far apart the middle portion of data spreads in value.

quantile(data\_full$price, c(.25, .44, .99))

-------------------------------------------------------------------------------------------------------------------

# Plotting

# After having looked at your preliminary data, let us have a look at it by creaing simple and not so simple plots

# Scatterplot (For this we need two variables X, for the X Axis and Y for the Y Axis)

x <- data\_full$sqft\_living

y <- data\_full$price

scatter <- plot(x, y, main = "Correlation between Sqft Living Space and Price of Houses", ylab = "Price of houses", xlab = "Living Space Size in Sqft")

scatter

# Uppss. this looks a bit off due to some extreme values, lets fix this

scatter <- plot(x, y, main = "Correlation between Sqft Living Space and Price of Houses", ylab = "Price of houses", xlab = "Living Space Size in Sqft", ylim = c(0, 5000000))

scatter

# Correlation Coefficient R (how to calculate this one)

cor(data\_full$price, data\_full$sqft\_living)

# Pie Chart

df <- data.frame(

group = c("Male", "Female", "Child"),

value = c(25, 25, 50)

)

head(df)

bp<- ggplot(df, aes(x="", y=value, fill=group))+

geom\_bar(width = 1, stat = "identity")

bp

pie <- bp + coord\_polar("y", start=0)

pie

# Histogram using a dataset built in in R

hist(AirPassengers,

main="Histogram for Air Passengers",

xlab="Passengers",

border="blue",

col="green",

xlim=c(100,700),

las=1,

breaks=5)

# Boxplot

bp <- boxplot(data\_full$sqft\_living)

bp

# Heatmap

data <- as.matrix(mtcars)

heatmap(data, scale = "column")

-------------------------------------------------------------------------------------------------------------------

# Calculating Confidence Intervals

# Calculating Z Scores and the Z Score Table

# Hypothesis Test (One and Two Tailed)

# Inferential Statistics (linear and Multiple Regression)

# Machine Learning (Case Example)

# Learning how ML works using the diamionds Dataset

View(diamonds)

str(diamonds)

dim(diamonds)

length(diamonds$price)

# Splitting the data into train and test sets

sample.split(diamonds$price, SplitRatio = 0.65) -> Split\_Values

subset(diamonds, Split\_Values==T) -> Train\_Set

subset(diamonds, Split\_Values==F) -> Test\_Set

head(Train\_Set)

dim(Train\_Set)

dim(Test\_Set)

# Building a linear model on top of the training dataset

lm(price~., Train\_Set) -> mod\_regress #Lets use stepwise backward regression to optimize our model and compare to the RMSE obtained before

predict(mod\_regress, Test\_Set) -> result\_regress

cbind(Actual = Test\_Set$price, Predicted = result\_regress) -> Final\_Data

as.data.frame(Final\_Data)-> Final\_Data

head(Final\_Data)

# Finding theb RMSE (Root Mean Squared Error)

(Final\_Data$Actual- Final\_Data$Predicted) -> error

cbind(Final\_Data, error) -> Final\_Data

Initial\_RMSE <- sqrt(mean(Final\_Data$error^2))

RMSE