# 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)

# Reading in the Data

ins\_data <- read.csv("Insurance.csv")

# Getting familiar with the data set

class(ins\_data)

names(ins\_data)

length(ins\_data)

dim(ins\_data)

head(ins\_data)

tail(ins\_data)

summary(ins\_data)

str(ins\_data)

View(ins\_data)

NA\_check <- is.na(ins\_data)

NA\_check

summary(NA\_check)

# Please calculate the following descriptive statistics for the variable BMI

mean(ins\_data$bmi)

bmi <- ins\_data$bmi

charges <- ins\_data$charges

View(charges)

View(bmi)

bmi\_charg <- cbind(bmi, charges)

View(bmi\_charg)

median(ins\_data$bmi)

class(bmi\_charg)

bmi\_charg <- as.data.frame(bmi\_charg)

summary(ins\_data)

str(ins\_data)

# Please calculate the following measures of dispersion for the variable age

var(ins\_data$age, y=NULL)

var(bmi\_charg)

sqrt(var(ins\_data$age, y=NULL)) #St.Dev

sqrt(var(bmi\_charg)) #St.Dev

range(ins\_data$age)

range(ins\_data$bmi)

IQR(ins\_data$age) # 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.

heaviest\_person <- subset(ins\_data, bmi =="53.13")

View(heaviest\_person)

quantile(ins\_data$age, c(.25, .50, .99))

# let us set the display area for more than one chart

par(mfrow = c(1, 1))

plot(ins\_data)

plot(ins\_data$age, ins\_data$charges, type = "p",

main = "Correlation between Age X and Charges as Y",

sub = "Correlation Coefficient r = 0.299", xlab = "Age in years",

ylab = "Charges in USD", col = "blue")

cor(ins\_data$age, ins\_data$charges)

?plot#what else can we add?

plot(ins\_data$sex, ins\_data$charges)

plot(ins\_data$bmi, ins\_data$charges)

plot(ins\_data$children, ins\_data$charges)

str(ins\_data$smoker)

ins\_data$smoker <- as.factor(ins\_data$smoker)

plot(ins\_data$smoker, ins\_data$charges,

main = "Comparison of insurance cost for smokers and non smokers",

ylab = "Insurance Cost", xlab =" Smoking Status", col ="red")

plot(ins\_data$sex, ins\_data$charges,

main = "Comparison of insurance cost for smokers and non smokers",

ylab = "Insurance Cost", xlab =" Sex", col ="red")

ins\_data$smoker <- as.numeric(ins\_data$smoker)

cor(ins\_data$smoker, ins\_data$charges)

class(ins\_data$smoker)

cor(ins\_data$age, ins\_data$charges) # create some more

# Please create the following plots:

# heatmap of both BMI and charges

bmi <- ins\_data$bmi

charges <- ins\_data$charges

class(bmi)

# barchart

barchart <- barplot(ins\_data$age, breaks = 20)

?hist

hist(ins\_data$bmi, breaks = 30, main = "Histogram of BMI", xlab = "BMI")

mean(ins\_data$bmi)

hist(ins\_data$age, breaks = 80)

# boxplot

boxplot(ins\_data$bmi)

Distribution

# Histogram overlaid with kernel density curve

plot1 <- ggplot(ins\_data, aes(x=bmi)) +

geom\_histogram(aes(y=..density..),

binwidth=0.5,

colour="black", fill="white") +

geom\_density(alpha=.2, fill="blue") +

ggtitle("Density Curve and Histogram for BMI Measurements")

plot1 + geom\_vline(xintercept = 26)

# Generating a linear model to estimate insurance charges

model <- lm(ins\_data$charges ~ ins\_data$age + ins\_data$sex +

ins\_data$bmi + ins\_data$children + ins\_data$smoker)

summary(model)

# Generating a linear model to estimate charges on BMI

model\_bmi <- lm(ins\_data$charges ~ ins\_data$smoker)

summary(model\_bmi)

# Using step-wise backward regression to optimize our model

improved.model <- step(model, direction = "backward")

summary(improved.model)

# Use machine learning to estimate insurance cost based on the improved regression model

# Splitting the data into train and test sets

sample.split(ins\_data$charges, SplitRatio = 0.65) -> Split\_Values

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

subset(ins\_data, 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(charges~., Train\_Set) -> mod\_regress

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

cbind(Actual = Test\_Set$charges, 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

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

RMSE

plot(Test\_Set$charges, result\_regress)