|  |
| --- |
| > #display dataframeView(df) >  > #size dim(df) >  > #check datatypesstr(df) >  > #check for missing valuescolSums(is.na(df)) >  > #drop null valuesdf <- na.omit(df) #it will delete all rows >  > df <- read.csv("C:/Users/YAW/Downloads/Data Set- Inc5000 Company List-2014 (2).csv") >  > #drop null values method-2drops <- c("X\_input") >  > names(df)  [1] "X\_input" "X\_num"   [3] "X\_widgetName" "X\_source"   [5] "X\_resultNumber" "X\_pageUrl"   [7] "id" "rank"   [9] "workers" "company"  [11] "url" "state\_l"  [13] "state\_s" "city"  [15] "metro" "growth"  [17] "revenue" "industry"  [19] "yrs\_on\_list"  > View(df) > #drop null values method-3df <- read.csv> df <- read.csv("C:/Users/YAW/Downloads/Data Set- Inc5000 Company List-2014 (2).csv")df <- df[ ,!(names(df) %in% c("X\_input"))] >  > View(df) > #drop null values method-3df <- read.csv> df <- read.csv("C:/Users/YAW/Downloads/Data Set- Inc5000 Company List-2014 (2).csv")df <- df[ ,!(names(df) %in% c("X\_input"))] > View(df) > View(df) > #check duplicatesdf <- df[!duplicated(df$id),] >  > #round of to 2 decimal places#df$growth <- round(df$growth,digit=2) >  > View(df) > #checking outliersinstall.packages("tidyverse") #one time processlibrary(ggplot2) >  > #drawing boxplot and labeling outliersggplot(df, aes(x=revenue, y=growth)) + geom\_boxplot(outlier.colour = "red", outlier.shape = 1)+ scale\_x\_continuous(labels = scales::comma)+coord\_cartesian(ylim = c(0, 1000)) >  > #Calculating IQR growthQ1\_growth <- quantile(df$growth,0.25)Q3\_growth <- quantile(df$growth,0.75) > #using IQR function IQR\_growth\_new <- IQR(df$growth) #calculate using IQR > #Calculate IQR for revenue column  > IQR\_revenue <- IQR (df$revenue) > q1\_revenue <- quantile(df$revenue, 0.25) > q3\_revenue <- quantile(df$revenue, 0.75)  > #removing outliers no\_outliers<-subset (df,df$growth> (q1\_growth-1.5\*IQR\_revenue)&no\_outliers$revenue<(q3\_revenue+1.5\*IQR\_revenue))view(no\_outliers)dim(no\_outliers) > #summary statistics summary(no\_outliers) > #drawing boxplot and labeling outliers  > ggplot(df,aes(x=revneue,y=growth))+ + geom\_boxplot(outlier.colour="red", outlier.shape =1)+ + scale\_x\_continous(labels= scales::comma)+coord\_cartesian(ylim=c(0,1000)) Error in ggplot(df, aes(x = revneue, y = growth)) :   could not find function "ggplot" > #drawing boxplot and labeling outliers ggplot(df,aes(x=revneue,y=growth))+geom\_boxplot(outlier.colour="red", outlier.shape =1)+scale\_x\_continous(labels= scales::comma)+coord\_cartesian(ylim=c(0,1000)) > ggplot(no\_outliers, aes(x=rank, y=workers))+geom\_point()+sale\_y\_continuous(labels=scales::comma)+coord\_cartesian(ylim=c(0,300)) Error in ggplot(no\_outliers, aes(x = rank, y = workers)) :   could not find function "ggplot" > ggplot(no\_outliers, aes(x=rank, y=workers))+geom\_point()+sale\_y\_continuous(labels=scales::comma)+coord\_cartesian(ylim=c(0,300)) Error in ggplot(no\_outliers, aes(x = rank, y = workers)) :   could not find function "ggplot" > save.image("F:/Just IT Bootcamp Data Techician/Week 12 Data Technician Bootcamp/Technical Training Session/Data Analyisis.R-Final Copy 1.RData") > ggplot(no\_outliers,aes(x=rank, y=workers))+geom\_point()+sale\_y\_continuous(labels=scales::comma)+coord\_cartesian(ylim=c(0,300)) eom\_bar()+coord\_cartesian(ylim)=c(0,1500),xlim=c(0,15)),  Error: unexpected symbol in "ggplot(no\_outliers,aes(x=rank, y=workers))+geom\_point()+sale\_y\_continuous(labels=scales::comma)+coord\_cartesian(ylim=c(0,300)) eom\_bar" > write.csv(no\_outliers,"cleaned\_df.csv") Error in eval(expr, p) : object 'no\_outliers' not found > #export cleaned data  > write.csv(no\_outliers,"cleaned\_df.csv") Error in eval(expr, p) : object 'no\_outliers' not found > #export cleaned data  > write.csv(no\_outliers,"cleaned\_df.csv") Error in eval(expr, p) : object 'no\_outliers' not found |
|  |
| |  | | --- | | > #scatter plot  data("mtcars")  plot(mtacars$mpg, mtcars$hp, main="scatter plot",  xlab="MPG", ylab="Horsepower" ,col="blue", pch=19)  view(mtcars)      #Bar plot  barplot(mtcars$mpg, names.arg=mtcars$carb,  main="Bar Plot", xlab="carburetors", ylab="MPG")    #line plot  x<-seq(0,2\*pi, lenght.out=100)  y <- sin(x)  plot(x, y, type= "l" , main="Line plot", xlab="x", ylab="sin(x)", col="red")    #iris dataset  data("iris")  view("iris") | |