|  |
| --- |
| **#1)Read the file in Zip format and get it into R.** |
|  |  |
|  | x = paste('C:/Users/Windows 10/Desktop/Neelam\_ACADGILD/Assignment', |
|  | 'AirQualityUCI.zip', sep='') |
|  |  |
|  | unzip('AirQualityUCI.zip') |
|  | d = read.csv('AirQualityUCI.csv',sep = ";") |
|  | View(d) |
|  |  |
|  | **#2)Create Univariate for all the columns.** |
|  | #Univariate analysis is the simplest form of analyzing data. "Uni" means "one", |
|  | #so in other words your data has only one variable |
|  |  |
|  | #we can do univariate analysis by the following command too |
|  | summary(airquality) |
|  |  |
|  |  |
|  | #or by visually |
|  | library(purrr) |
|  | library(tidyr) |
|  | library(ggplot2) |
|  |  |
|  | airquality |
|  | keep(is.numeric) |
|  | gather() |
|  | ggplot(aes(value)) + |
|  | facet\_wrap(~ key,scales = "free") + |
|  | geom\_histogram() |
|  |  |
|  | #or we can plot univariate individually for each variable |
|  | #hence plotting histogram |
|  |  |
|  | hist(airquality$Ozone ,xlab = "ozone", ylab = "Frequency",main="Histogram of ozone",col="red") |
|  | hist(airquality$Solar.R ,xlab = "solar.r", ylab = "Frequency",main="Histogram of solar.r",col="blue") |
|  | hist(airquality$Wind ,xlab = "wind", ylab = "Frequency",main="Histogram of wind",col="yellow") |
|  | hist(airquality$Temp ,xlab = "temp", ylab = "Frequency",main="Histogram of temp",col="darkblue") |
|  | hist(airquality$Month ,xlab = "month", ylab = "Frequency",main="Histogram of month",col="pink") |
|  | hist(airquality$Day ,xlab = "day", ylab = "Frequency",main="Histogram of day",col="purple") |
|  |  |
|  |  |
|  | **#3)Check for missing values in all columns.** |
|  | #with the help of summary function we can find which variable has how many NA value |
|  | #or check for missing values |
|  |  |
|  | summary(airquality) |
|  | #thus ozone and solar.r has missing values |
|  |  |
|  | **#4)Impute the missing values using appropriate methods.** |
|  |  |
|  | str(airquality) |
|  |  |
|  | install.packages('mice') |
|  | library(mice) |
|  | md.pattern(airquality) |
|  |  |
|  | #visualizing |
|  | install.packages('VIM') |
|  | library(VIM) |
|  |  |
|  | mice\_plot <- aggr(airquality, col=c('navyblue','yellow'), |
|  | numbers=TRUE, sortVars=TRUE, |
|  | labels=names(airquality), cex.axis=.7, |
|  | gap=3, ylab=c("Missing data","Pattern")) |
|  |  |
|  | # In this case we are using predictive mean matching as imputation method |
|  | imputed\_Data <- mice(airquality, m=5, maxit = 50, method = 'pmm', seed = 500) |
|  | summary(imputed\_Data) |
|  |  |
|  |  |
|  | completeData <- complete(imputed\_Data) |
|  | completeData |
|  |  |
|  |  |
|  | #5)Create bi-variate analysis for all relationships. |
|  |  |
|  |  |
|  | install.packages('psych') |
|  | library(psych) |
|  | pairs.panels( airquality[,c(1,2,3,4,5,6)], |
|  | method = "pearson", |
|  | hist.col = "red", |
|  | density = TRUE, |
|  | ellipses = TRUE, |
|  | lm=TRUE, |
|  | main ="Bivariate Scatter plots with Pearson Correlation & Histogram" |
|  | ) |
|  |  |
|  | #6)Test relevant hypothesis for valid relations. |
|  |  |
|  | str(airquality) |
|  |  |
|  |  |
|  | t.test(x=airquality$Ozone, y=airquality$Solar.R ,alternative = "two.sided",mu=0 ,paired = TRUE) |
|  | t.test(x=airquality$Temp, y=airquality$Wind ,alternative = "two.sided",mu=0 ,paired = TRUE) |
|  | t.test(x=airquality$Ozone, y=airquality$Temp ,alternative = "two.sided",mu=0 ,paired = TRUE) |
|  | t.test(x=airquality$Day, y=airquality$Solar.R ,alternative = "two.sided",mu=0 ,paired = TRUE) |
|  |  |
|  | #as p value of this test is <0.05 we reject the null hypo |
|  | #and accept the alternative hypothesis which says there |
|  | #Mean of 1 variable - Mean of 2 variable is not equal to 0 |
|  | #thus this are some test that we performed |
|  |  |
|  | #7)Create cross tabulations with derived variables. |
|  |  |
|  | attach(airquality) |
|  | unique(Wind) |
|  | unique(Temp) |
|  | #derived variables of wind and temp |
|  | x<- cut(Wind,quantile(Wind)) |
|  | x<- cut(Wind,breaks = seq(1,21,3),labels = c("wind1","wind2","wind3","wind4","wind5","wind6")) |
|  | y<- cut(Temp,quantile(Temp)) |
|  | y<- cut(Temp,breaks = seq(55,100,9),labels = c("temp1","temp2","temp3","temp4","temp5")) |
|  | table(x,y) |
|  |  |
|  |  |
|  | install.packages('gmodels') |
|  | library(gmodels) |
|  | CrossTable(x,y) |
|  |  |
|  |  |
|  | univariateTable(~age +gender + height + weight,data=Diabetes) |