DATA ANALYTICS WITH R, EXCEL AND TABLAEU

ASSIGNMENT 10.1 ANSWERS

By ASHISH S SHANBHAG

ashishshanbhag108@gmail.com

Question no:

5)

1.Read the file in Zip format and get it into R.

2.Create Univariate for all the columns.

3.Check for missing values in all columns.

4.Impute the missing values using appropriate methods.

5.Create bi-variate analysis for all relationships.

6.Test relevant hypothesis for valid relations.

7.Create cross tabulations with derived variables
.
8.Check for trends and patterns in time series.

9.Find out the most polluted time of the day and the name of the chemical compound.

Ans

```
require("datasets")
data("airquality")
str(airquality)
## 'data.frame':    153 obs. of  6 variables:
##  $ Ozone  : int  41 36 12 18 NA 28 23 19 8 NA ...
##  $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
##  $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
##  $ Temp   : int  67 72 74 62 56 66 65 59 61 69 ...
##  $ Month  : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...
head(airquality)
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5## 6    28      NA 14.9   66     5   6
```

```
col1<- mapply(anyNA,airquality) # apply function anyNA() on all columns of airquality dataset
col1
##   Ozone Solar.R    Wind    Temp   Month     Day
##    TRUE    TRUE   FALSE   FALSE   FALSE   FALSE
```

The output shows that only Ozone and Solar.R attributes have NA i.e. some missing value.

```
# Impute monthly mean in Ozone
for (i in 1:nrow(airquality)){
  if(is.na(airquality[i,"Ozone"])){
    airquality[i,"Ozone"]<- mean(airquality[which(airquality[,"Month"]==airquality[i,"Month"]),"Ozone"],na.rm = TRUE)
  }
# Impute monthly mean in Solar.R
    if(is.na(airquality[i,"Solar.R"])){
    airquality[i,"Solar.R"]<- mean(airquality[which(airquality[,"Month"]==airquality[i,"Month"]),"Solar.R"],na.rm = TRUE)
  }

}
#Normalize the dataset so that no particular attribute has more impact on clustering algorithm than others.
normalize<- function(x){
  return((x-min(x))/(max(x)-min(x)))
}
airquality<- normalize(airquality) # replace contents of dataset with normalized values
str(airquality)
## 'data.frame':    153 obs. of  6 variables:
##  $ Ozone  : num  0.1201 0.1051 0.033 0.0511 0.0679 ...
##  $ Solar.R: num  0.568 0.351 0.444 0.937 0.541 ...
##  $ Wind   : num  0.0192 0.021 0.0348 0.0315 0.0399 ...
##  $ Temp   : num  0.198 0.213 0.219 0.183 0.165 ...
##  $ Month  : num  0.012 0.012 0.012 0.012 0.012 ...
##  $ Day    : num  0 0.003 0.00601 0.00901 0.01201 ...
```

```
Y<- airquality[,"Ozone"] # select Target attribute
X<- airquality[,"Solar.R"] # select Predictor attribute

model1<- lm(Y~X)
model1 # provides regression line coefficients i.e. slope and y-intercept
##
## Call:
## lm(formula = Y ~ X)
##
## Coefficients:
## (Intercept)            X
##     0.06509      0.09849
```

```
p1<- predict(model1,data.frame("X"=10))
p1
##        1
## 1.049993
```

The predicted value of “Ozone” is 1.0499933 when “Solar.R”= 10

```
# Prediction of 'Ozone' when 'Wind'= 5
p2<- predict(model2,data.frame("X"=5))
p2
##         1
## -21.46849
```