18_Nishan

Nishan Neupane

2024-05-31

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Including Plots

You can also embed plots, for example:

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```r
#QN.9
# Load required libraries
library(stats)
library(ggplot2)
library(ggfortify)

# a)
city_names <- c("Atlanta", "Chicago", "Denver", "Houston", "Los Angeles", "Miami",
                "New York", "San Francisco", "Seattle", "Washington D.C")

city.dissimilarity <- matrix(c(
  0, 587, 1212, 701, 1936, 604, 748, 2139, 2182, 543,
  587, 0, 920, 940, 1745, 1188, 713, 1858, 1737, 597,
  1212, 920, 0, 879, 831, 1726, 1631, 949, 1021, 1494,
  701, 940, 879, 0, 1374, 968, 1420, 1645, 1891, 1220,
  1936, 1745, 831, 1374, 0, 2339, 2451, 347, 959, 2300,
  604, 1188, 1726, 968, 2339, 0, 1092, 2594, 2734, 923,
  748, 713, 1631, 1420, 2451, 1092, 0, 2571, 2408, 205,
  2139, 1858, 949, 1645, 347, 2594, 2571, 0, 678, 2442,
  2182, 1737, 1021, 1891, 959, 2734, 2408, 678, 0, 2329,
  543, 597, 1494, 1220, 2300, 923, 205, 2442, 2329, 0
), nrow = 10, byrow = TRUE)

rownames(city.dissimilarity) <- city_names
colnames(city.dissimilarity) <- city_names

city_dissimilarity <- as.dist(city.dissimilarity)
```

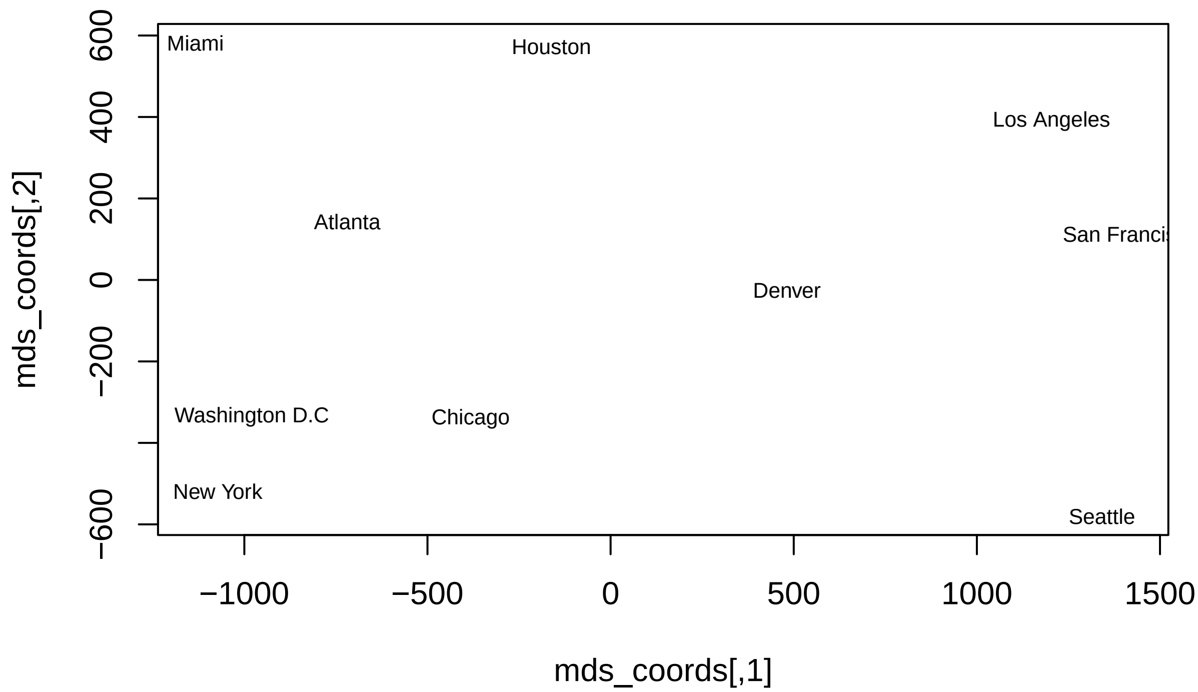
```r
#Dissimilarity distance is reduce form of given matrix we will get 9*9 matrix from above 10*10 matrix a
# b) Fit a classical MDS model
mds_fit <- cmdscale(city.dissimilarity, eig = TRUE, k = 2)
mds_fit
```

```
## $points
##                      [,1]       [,2]
## Atlanta         -718.7594  142.99427
## Chicago         -382.0558 -340.83962
## Denver           481.6023  -25.28504
## Houston         -161.4663  572.76991
## Los Angeles     1203.7380  390.10029
## Miami          -1133.5271  581.90731
## New York       -1072.2357 -519.02423
## San Francisco   1420.6033  112.58920
## Seattle         1341.7225 -579.73928
## Washington D.C  -979.6220 -335.47281
##
## $eig
##  [1]  9.582144e+06  1.686820e+06  8.157298e+03  1.432870e+03  5.086687e+02
##  [6]  2.514349e+01 -4.312942e-10 -8.977013e+02 -5.467577e+03 -3.547889e+04
##
## $x
## NULL
##
## $ac
## [1] 0
##
## $GOF
## [1] 0.9954096 0.9991024
```

```r
#Multidimensional model gives the information about actual location of the city without removing the ac

#c. Summary of the model
mds_coords <- mds_fit$points
print(mds_coords)
```

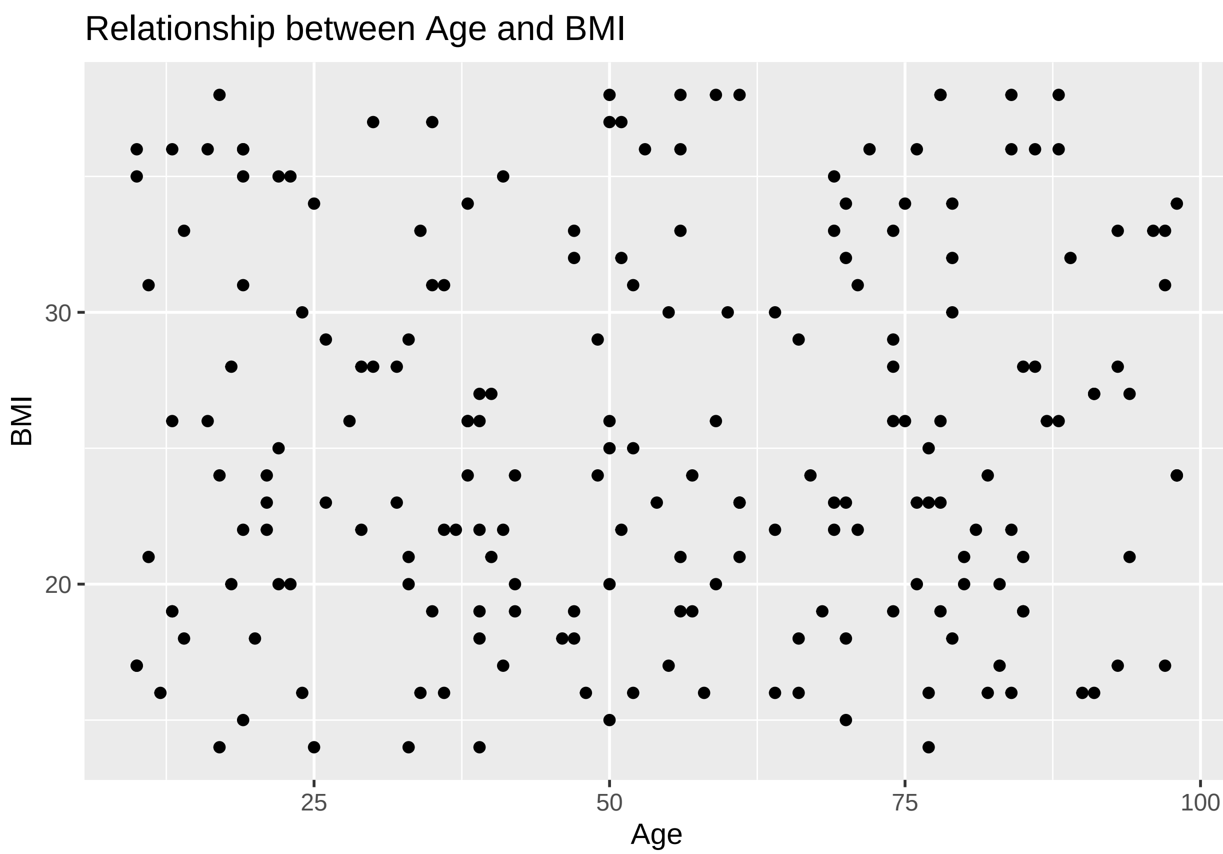
```
##                      [,1]       [,2]
## Atlanta         -718.7594  142.99427
## Chicago         -382.0558 -340.83962
## Denver           481.6023  -25.28504
## Houston         -161.4663  572.76991
## Los Angeles     1203.7380  390.10029
## Miami          -1133.5271  581.90731
## New York       -1072.2357 -519.02423
## San Francisco   1420.6033  112.58920
## Seattle         1341.7225 -579.73928
## Washington D.C  -979.6220 -335.47281
```

```r
#d. Create the bi-plot of the MDS model
#Bi-plot of the model
plot(mds_coords, type = "n")
text(mds_coords, labels = city_names, cex = 0.7)
```

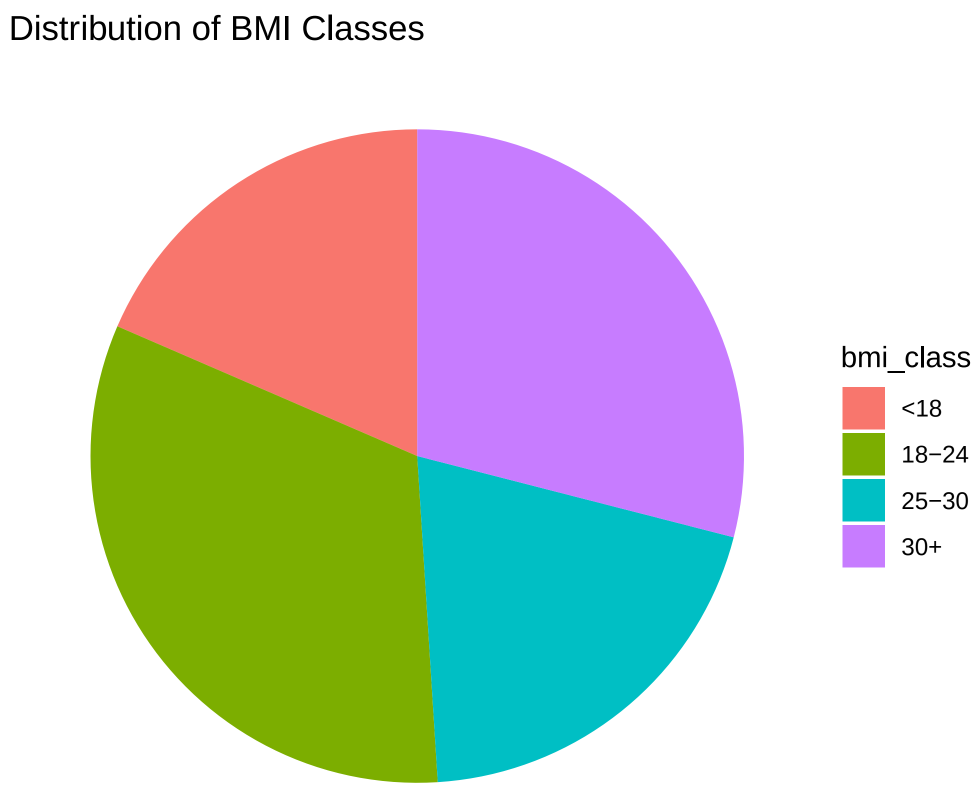



```
#from the above graph we can see tha position of the data in 2d
```

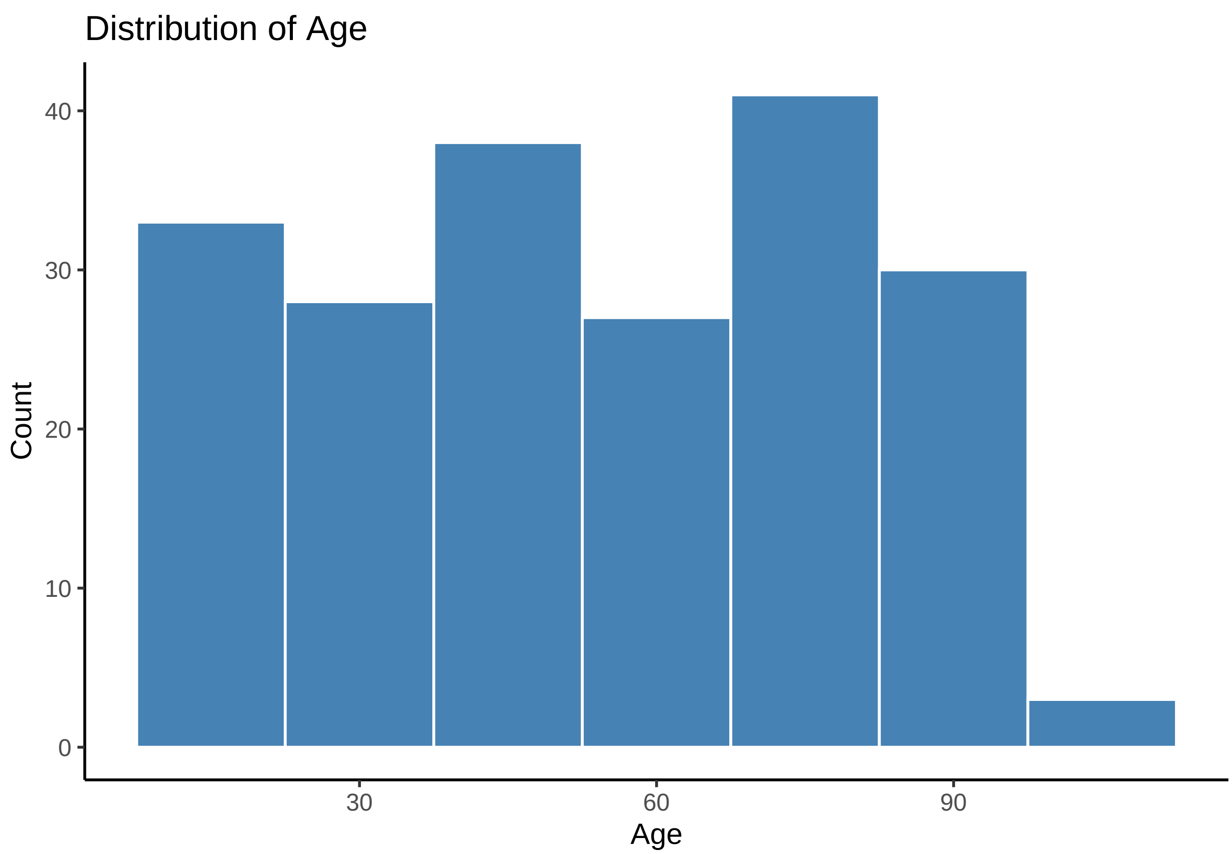
```
# Question No 6

library(ggplot2)
set.seed(18)

# a
age <- sample(10:99, 200, replace = TRUE)
sex <- sample(c("Male", "Female"), 200, replace = TRUE)
education <- sample(c("No education", "Primary", "Secondary", "Beyond secondary"), 200, replace = TRUE)
socioeconomic_status <- sample(c("Low", "Middle", "High"), 200, replace = TRUE)
bmi <- sample(14:38, 200, replace = TRUE)
#data is injected as per the question

# b

ggplot(data = data.frame(age, bmi), aes(x = age, y = bmi)) +
  geom_point() +
  labs(x = "Age", y = "BMI", title = "Relationship between Age and BMI")
```




```
# No trend is seen fro the data that means the data is spred all over the graph

# c
bmi_class <- cut(bmi, breaks = c(0, 18, 24, 30, Inf), labels = c("<18", "18-24", "25-30", "30+"))

#For pie chart
ggplot(data.frame(bmi_class), aes(x = "", fill = bmi_class)) +
  geom_bar(width = 1) +
  coord_polar("y", start = 0) +
  labs(title = "Distribution of BMI Classes") +
  theme_void() +
  theme(legend.position = "right")
```




```
#from the pie chart we can see the maximum part of the data is covered by group 25-30 and 30+
#and minimum part of the data is from <18

# d

ggplot(data.frame(age), aes(x = age)) +
  geom_histogram(binwidth = 15, fill = "steelblue", color = "white") +
  labs(x = "Age", y = "Count", title = "Distribution of Age") +
  theme_classic()
```




```
#From above plot we can see thet all the data has simmilar frequency except highest one
```

```
#8
library(car)
```

```
## Loading required package: carData
```

```
library(e1071)
# a
data <- Arrests
ind <- sample(2, nrow(data),
              replace = T, prob = c(0.8, 0.2))
train <- data[ind==1,]
test <- data[ind==2,]
#data is divided in 80-20 portion

# b

# For the logistic regression model
logistic_model <- glm(released ~ ., data = train, family = "binomial")

# For the Naive Bayes model
nb_model <- naiveBayes(released ~ ., data = train)

# c
```

```r
# to get  predictions.
logistic_pred <- predict(logistic_model, newdata = test, type = "response")
nb_pred <- predict(nb_model, newdata = test)


logistic_pred_class <- ifelse(logistic_pred > 0.5, 1, 0)
logistic_conf_matrix <- table(Predicted = logistic_pred_class, Actual = test$released)
logistic_conf_matrix
```

```
##          Actual
## Predicted  No Yes
##         0   9  11
##         1 175 852
```

```r
logistic_accuracy <- sum(diag(logistic_conf_matrix)) / sum(logistic_conf_matrix)
print(paste("Logistic Regression Accuracy:", logistic_accuracy))
```

```
## [1] "Logistic Regression Accuracy: 0.822349570200573"
```

```r
# True Negatives (TN): 9
# in this case actual result is no and our model predect no
# False Negatives (FN): 3
# in this case actual result is yes and our model predect no
# False Positives (FP): 176
# in this case actual result is yes and our model predect no.
# True Positives (TP): 837
# in this case actual result is yes and our model predect yes
#it has 83% accuracy

nb_conf_matrix <- table(Predicted = nb_pred, Actual = test$release)
nb_conf_matrix
```

```
##          Actual
## Predicted  No Yes
##       No   29  38
##       Yes 155 825
```

```r
nb_accuracy <- sum(diag(nb_conf_matrix)) / sum(nb_conf_matrix)
print(paste("Naive Bayes Accuracy:", nb_accuracy))
```

```
## [1] "Naive Bayes Accuracy: 0.815663801337154"
```

```r
# True Negatives (TN): 27
# in this case actual result is no and our model predect no
# False Negatives (FN): 32
# in this case actual result is yes and our model predect no
# False Positives (FP): 158
# in this case actual result is yes and our model predect no.
# True Positives (TP): 808
# in this case actual result is yes and our model predect yes
```

```r
# accuracy of the naive bayes is 81%
# d
# from their accuracy we can say that logistic regression is best because it gives 83% accuracy
```

```r
# QN 7
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':
##
##     recode

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
data <- airquality
data$Month <- as.factor(data$Month)

#a.

# For checking sample size per month
per_month_count <- data %>% group_by(Month) %>% summarize(count = n())
per_month_count
```

```
## # A tibble: 5 x 2
##   Month count
##   <fct> <int>
## 1 5        31
## 2 6        30
## 3 7        31
## 4 8        31
## 5 9        30
```

```r
#Shapiro-Wilk test is performed for normality within each month
result <- tapply(data$Temp, data$Month, shapiro.test)
print(result)
```

```
## $`5`
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.94771, p-value = 0.1349
```

```
##
##
## $`6`
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.97158, p-value = 0.5832
##
##
## $`7`
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.94579, p-value = 0.1194
##
##
## $`8`
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.96391, p-value = 0.3688
##
##
## $`9`
##
##  Shapiro-Wilk normality test
##
## data:  X[[i]]
## W = 0.9513, p-value = 0.1831
```

```r
# it is in normal distribution because p value is greater than 0.07

#b.

airquality$Month <- factor(airquality$Month)
bartlett_result <- bartlett.test(Temp ~ Month, data = airquality)
print(bartlett_result)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  Temp by Month
## Bartlett's K-squared = 12.023, df = 4, p-value = 0.01718
```

```r
#since the p value is less than 0.05 this shows that they donot have equal variance



#c.
#we need to use Bartlett's test in the above case suggests that the "Temp" variable's variances
#one-way ANOVA is appropriate.
```

```r
#d.
data("airquality")
anova_model <- aov(Temp ~ Month, data = airquality)
summary(anova_model)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## Month         1   2413  2413.0   32.52 6.03e-08 ***
## Residuals   151  11205    74.2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
airquality$Month <- factor(airquality$Month)
anova_model <- aov(Temp ~ Month, data = airquality)
tukey_result <- TukeyHSD(anova_model)
print(tukey_result)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = Temp ~ Month, data = airquality)
##
## $Month
##            diff          lwr       upr     p adj
## 6-5 13.55161290   8.84386422 18.259362 0.0000000
## 7-5 18.35483871  13.68583759 23.023840 0.0000000
## 8-5 18.41935484  13.75035372 23.088356 0.0000000
## 9-5 11.35161290   6.64386422 16.059362 0.0000000
## 7-6  4.80322581   0.09547713  9.510974 0.0430674
## 8-6  4.86774194   0.15999325  9.575491 0.0388654
## 9-6 -2.20000000  -6.94617992  2.546180 0.7038121
## 8-7  0.06451613  -4.60448499  4.733517 0.9999995
## 9-7 -7.00322581 -11.71097449 -2.295477 0.0006215
## 9-8 -7.06774194 -11.77549062 -2.359993 0.0005376
```

```r
# here we can see relationship between temp and month of (6-5),(7-5),(8-5),(9-5) are less significant
#as compared to month of (9-6),(8-7).
```