ROLL NO: 13

Kaushal Khatiwada

2024-05-31

Q no 6

```r
library(ggplot2)
```

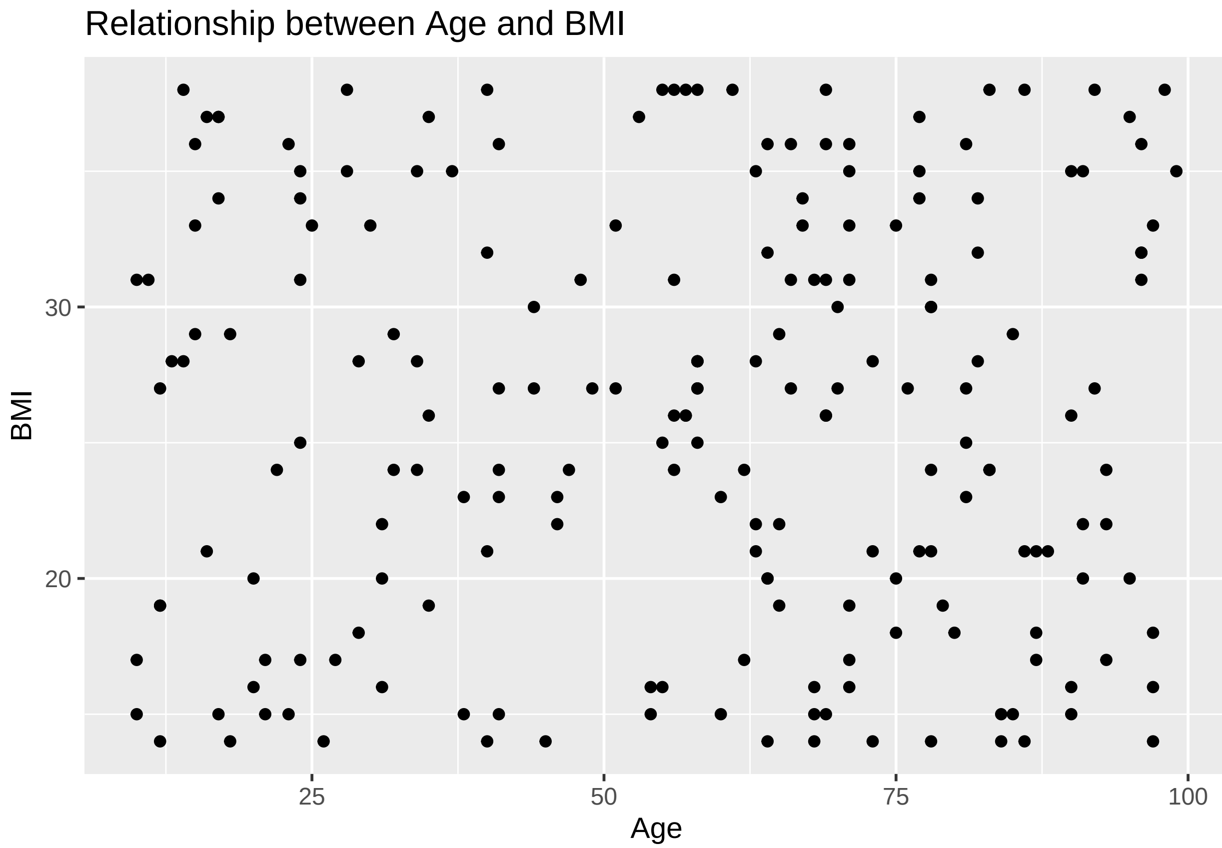
```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

A)

```r
set.seed(13)
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)
```

B)

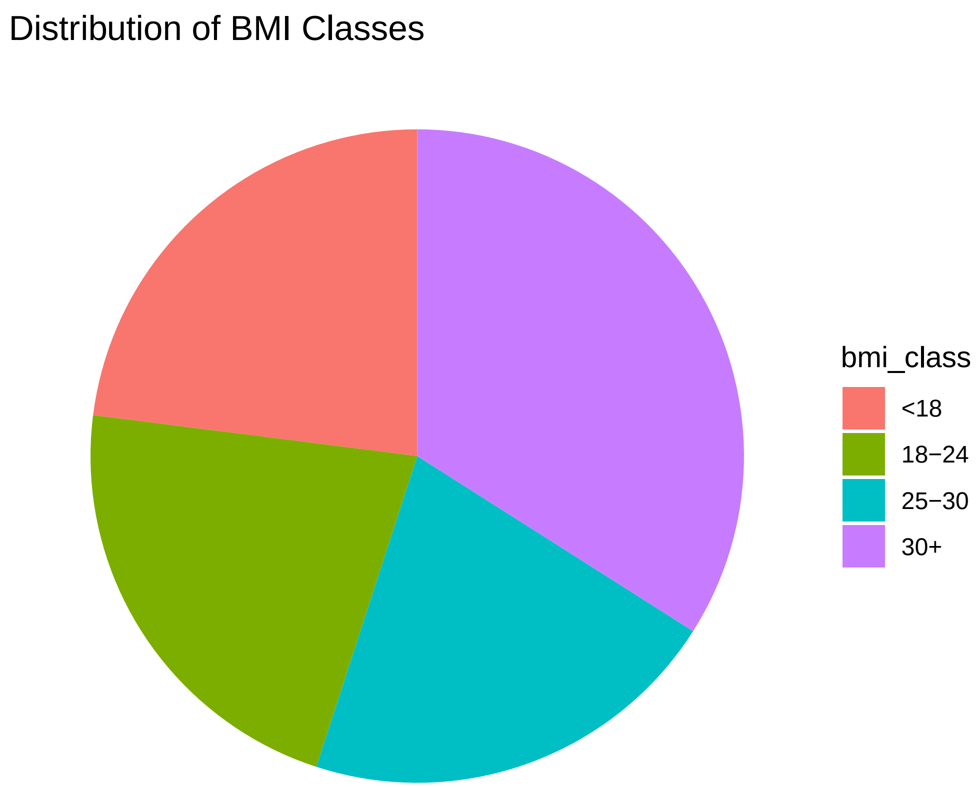
```r
ggplot(data = data.frame(age, bmi), aes(x = age, y = bmi)) +
  geom_point() +
  labs(x = "Age", y = "BMI", title = "Relationship between Age and BMI")
```



No trend is seen from the data that means the data is well spread

C)

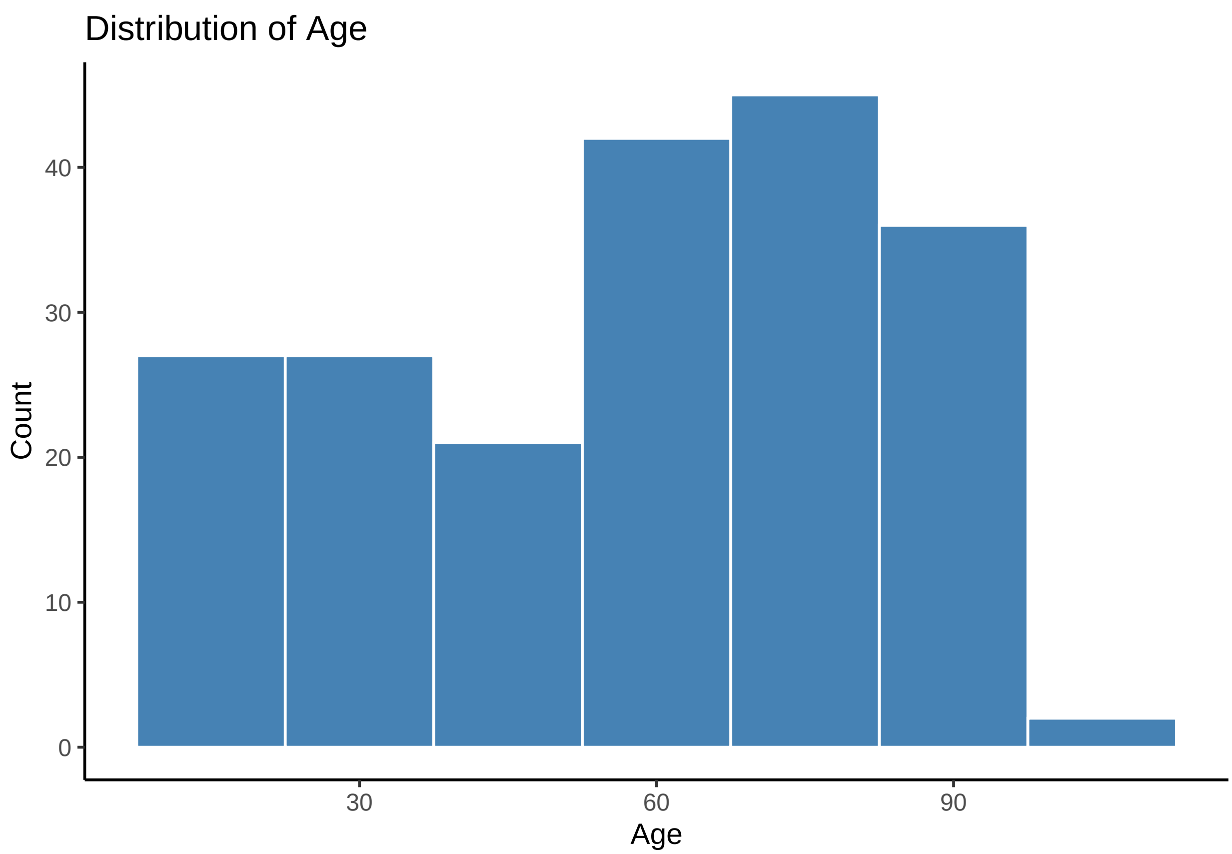
```
bmi_class <- cut(bmi, breaks = c(0, 18, 24, 30, Inf), labels = c("<18", "18-24", "25-30", "30+"))

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

Q no 7

```
library(dplyr)
```

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

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

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

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

A)

```r
# Check 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
```

Perform Shapiro-Wilk test for normality within each month

```r
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
```

The data follows a normal distribution within each month as p value is greater than 0.05.

B)

```r
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
```

c)

Bartlett's test in the above case suggests that the "Temp" variable's variances are roughly equal between months. Consequently, the conventional one-way ANOVA is appropriate.

D) perform the best independent sample statistical test for this data now and interpret the result carefully.

```r
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
```

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).

Q no 8

```r
library(car)
```

```
## Warning: package 'car' was built under R version 4.3.3
```

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

```
## Warning: package 'carData' was built under R version 4.3.3
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```r
arrest <- Arrests
```

A)

```r
arrest$year <- as.factor(arrest$year)
arrest$age <- as.factor(arrest$age)
arrest$checks <- as.factor(arrest$checks)
```

```r
set.seed(13)
ind <- sample(2,nrow(arrest), replace = T, prob = c(0.8,0.2))
arrest.train <- arrest[ind==1,]
arrest.test <- arrest[ind==2,]
```

##B)

```
model.lr <- glm(released ~ colour+age+sex+employed+citizen  , data = arrest.train, family = binomial)
summary(model.lr)
```

```
##
## Call:
## glm(formula = released ~ colour + age + sex + employed + citizen,
##     family = binomial, data = arrest.train)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -0.98559    1.25153  -0.788    0.431
## colourWhite   0.50072    0.09627   5.201 1.98e-07 ***
## age13         0.87377    1.39388   0.627    0.531
## age14         1.60247    1.30691   1.226    0.220
## age15         0.75878    1.24949   0.607    0.544
## age16         0.92943    1.24455   0.747    0.455
## age17         1.26297    1.24337   1.016    0.310
## age18         1.28647    1.24200   1.036    0.300
## age19         1.16267    1.24122   0.937    0.349
## age20         1.44178    1.24477   1.158    0.247
## age21         1.50770    1.24543   1.211    0.226
## age22         1.53365    1.24855   1.228    0.219
## age23         1.53844    1.25256   1.228    0.219
## age24         1.06483    1.24926   0.852    0.394
## age25         1.65273    1.26542   1.306    0.192
## age26         1.08761    1.25991   0.863    0.388
## age27         0.68232    1.25630   0.543    0.587
## age28         0.91598    1.26150   0.726    0.468
## age29         1.06796    1.26780   0.842    0.400
## age30         1.88633    1.29147   1.461    0.144
## age31         2.13675    1.34053   1.594    0.111
## age32         1.41084    1.29441   1.090    0.276
## age33         1.37905    1.27773   1.079    0.280
## age34         1.31521    1.27705   1.030    0.303
## age35         0.58994    1.28456   0.459    0.646
## age36         1.67926    1.29970   1.292    0.196
## age37         0.67932    1.27574   0.532    0.594
## age38         1.98395    1.34768   1.472    0.141
## age39         0.87658    1.28190   0.684    0.494
## age40         0.66840    1.27985   0.522    0.601
## age41         0.66566    1.29863   0.513    0.608
## age42         0.44811    1.30116   0.344    0.731
## age43         0.80743    1.32674   0.609    0.543
## age44         1.59954    1.38457   1.155    0.248
## age45         0.45125    1.30737   0.345    0.730
## age46        -0.19274    1.36389  -0.141    0.888
## age47         0.99054    1.42086   0.697    0.486
## age48         0.18319    1.35157   0.136    0.892
## age49         2.08316    1.64470   1.267    0.205
## age50         0.96049    1.67433   0.574    0.566
## age51        16.29795  613.08402   0.027    0.979
## age52        15.42899  608.15313   0.025    0.980
```

```
## age53          1.06072    1.65756   0.640    0.522
## age54          0.74743    1.48934   0.502    0.616
## age55         15.24418  825.78643   0.018    0.985
## age57         16.77565 1455.39806   0.012    0.991
## age58         16.28129 1455.39806   0.011    0.991
## age59         15.65132  967.34272   0.016    0.987
## age60         15.20030 1455.39806   0.010    0.992
## age62         16.27494 1455.39806   0.011    0.991
## age64         15.46232 1022.45451   0.015    0.988
## age66        -15.93184 1455.39806  -0.011    0.991
## sexMale       -0.22400    0.16647  -1.346    0.178
## employedYes    1.08099    0.09451  11.438  < 2e-16 ***
## citizenYes     0.49436    0.11187   4.419 9.90e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3785.8  on 4133  degrees of freedom
## Residual deviance: 3499.8  on 4079  degrees of freedom
## AIC: 3609.8
##
## Number of Fisher Scoring iterations: 14
```

```r
library(e1071)
```

```
## Warning: package 'e1071' was built under R version 4.3.3
```

```r
model.nb <- naiveBayes(released ~ colour+age+sex+employed+citizen, data = arrest.train)

summary(model.nb)
```

```
##           Length Class  Mode
## apriori   2      table  numeric
## tables    5      -none- list
## levels    2      -none- character
## isnumeric 5      -none- logical
## call      4      -none- call
```

C)

```r
#arrest.predict<-predict(model.lr,newdata =arrest.test,type="response")
#predict.lr<-as.factor((ifelse(predict>0.5,1,0)))
```

Q no 9

```r
library(stats)
city_distances <- matrix(c(
```

```r
  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, 1611, 1949, 2204, 1494,
  701, 940, 879, 0, 1374, 968, 1420, 1645, 1891, 1220,
  1936, 1745, 831, 1374, 0, 2339, 2451, 347, 2734, 2300,
  604, 1188, 1726, 968, 2339, 0, 1092, 2594, 2408, 923,
  748, 713, 1611, 1420, 2451, 1092, 0, 2571, 678, 205,
  2139, 1858, 1949, 1645, 347, 2594, 2571, 0, 678, 2442,
  2182, 1737, 2204, 1891, 2734, 2408, 678, 678, 0, 2329,
  543, 597, 1494, 1220, 2300, 923, 205, 2442, 2329, 0
), nrow = 10, byrow = TRUE)

# Assigning names to row and columns
city_names <- c("Atlanta", "Chicago", "Denver", "Houston", "Los Angeles", "Miami",
                "New York", "San Francisco", "Seattle", "Washington D.C.")
rownames(city_distances) <- city_names
colnames(city_distances) <- city_names
```

A)

Get dissimilarity distance as city.dissimilarity object

```r
city.dissimilarity <- as.dist(city_distances)
print(city.dissimilarity)
```

```
##                 Atlanta Chicago Denver Houston Los Angeles Miami New York
## Chicago             587
## Denver             1212     920
## Houston             701     940    879
## Los Angeles        1936    1745    831    1374
## Miami               604    1188   1726     968        2339
## New York            748     713   1611    1420        2451  1092
## San Francisco      2139    1858   1949    1645         347  2594     2571
## Seattle            2182    1737   2204    1891        2734  2408      678
## Washington D.C.     543     597   1494    1220        2300   923      205
##                 San Francisco Seattle
## Chicago
## Denver
## Houston
## Los Angeles
## Miami
## New York
## San Francisco
## Seattle                   678
## Washington D.C.          2442    2329
```

B)

Fit the classical MDS model using city.dissimilarity object

```r
mds.model <- cmdscale(city.dissimilarity, eig = TRUE, k = 2)  # Dimension 2
```

C)

Summary of model

```r
mds.points <- mds.model$points
print(mds.points)
```

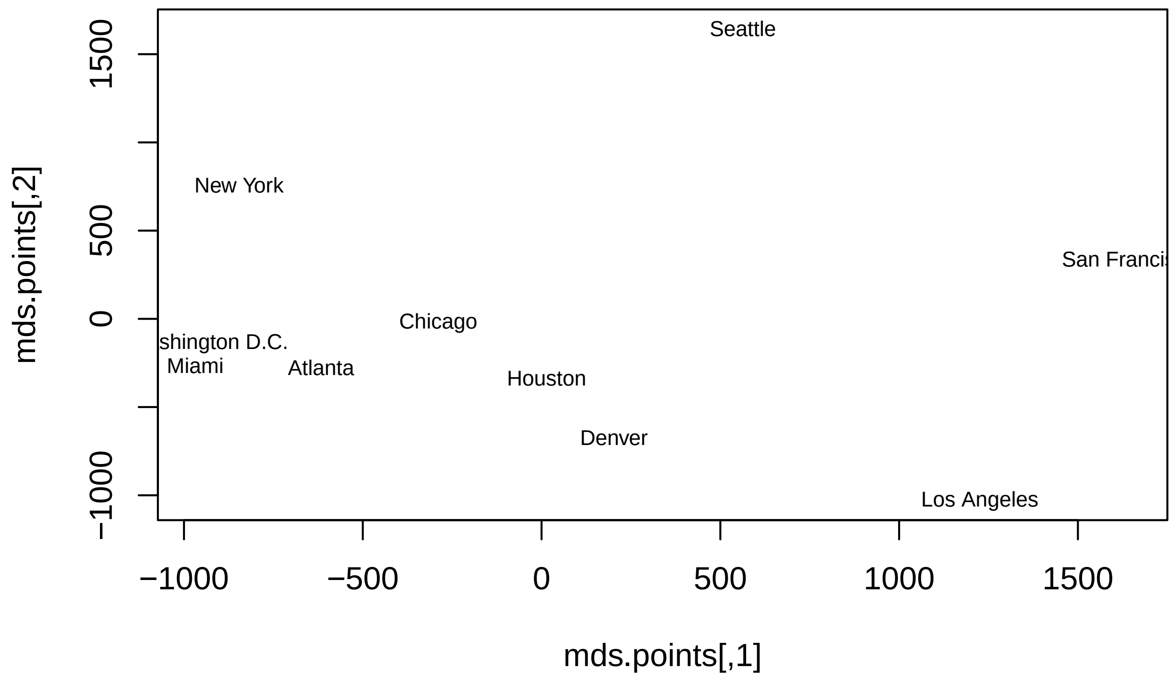
```
##                        [,1]        [,2]
## Atlanta         -616.46326  -277.03319
## Chicago         -288.61063   -22.16151
## Denver           202.61148  -672.61019
## Houston           14.25242  -335.54496
## Los Angeles     1225.78174 -1033.78934
## Miami           -968.45797  -264.31832
## New York        -845.50822   757.66327
## San Francisco   1645.58380   339.92746
## Seattle          563.12009  1646.43854
## Washington D.C. -932.30945  -138.57175
```

Interpretation

D)

Bi-plot of the model

```r
plot(mds.points, type = "n")
text(mds.points, labels = city_names, cex = 0.7)
```



Interpretation

Q no 10

head(iris)