DataMining_HW_5

Pratibha Chitta

5/1/2022

R Markdown

```
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.1.3
```

```
library(plyr)
```

```
## Warning: package 'plyr' was built under R version 4.1.2
```

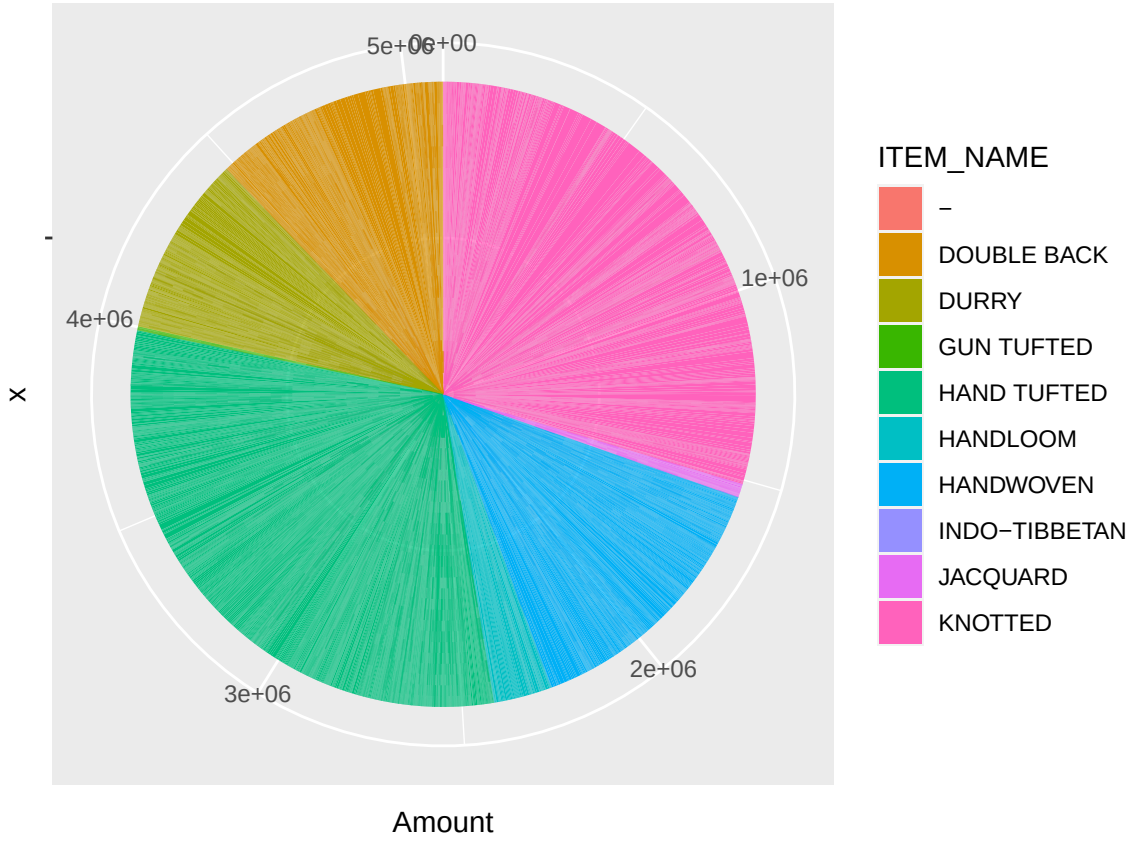
```
location <- 'C:/Users/pchitt2/Downloads'
setwd(location)
file <- 'Champo_Carpets.xlsx'

library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

```
library(tidyverse)
```

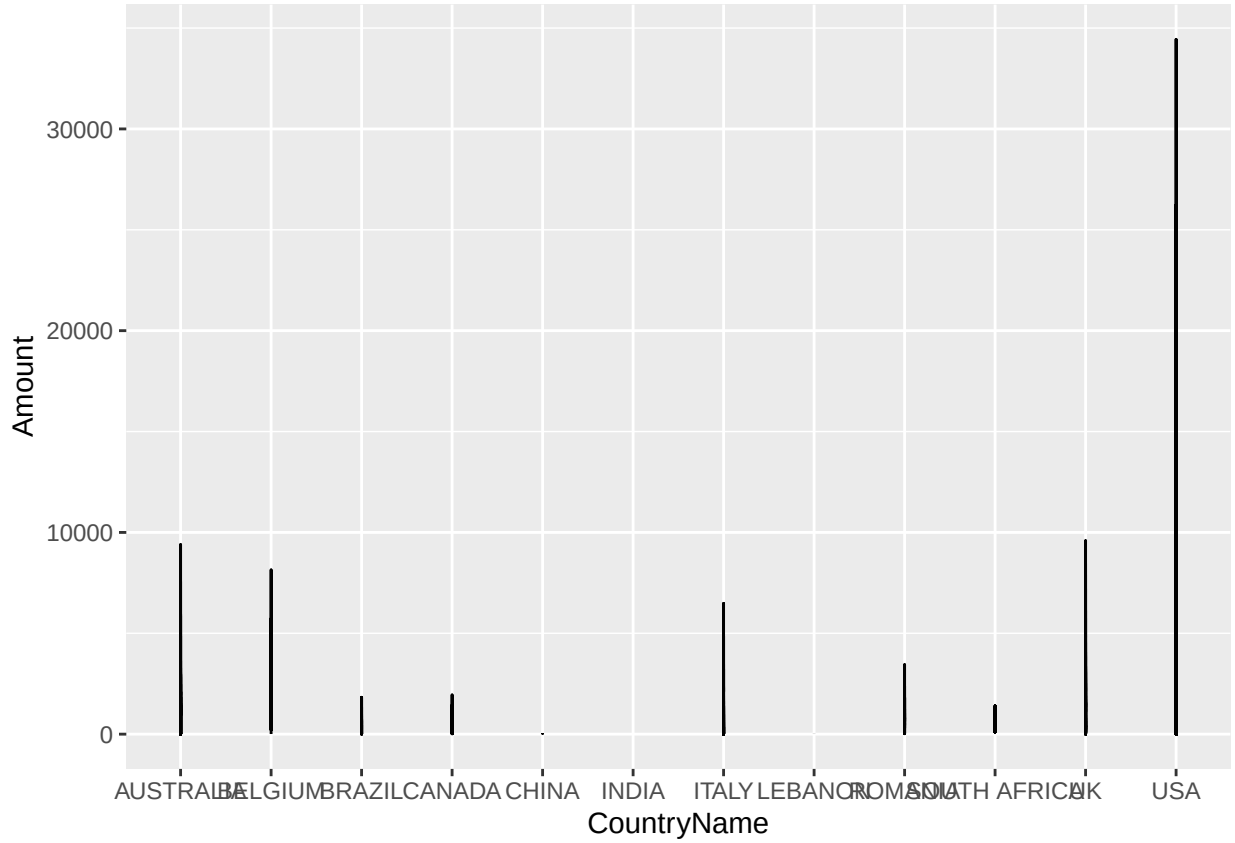
```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
## v purrr   0.3.4

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::arrange()   masks plyr::arrange()
## x purrr::compact()   masks plyr::compact()
## x dplyr::count()     masks plyr::count()
## x dplyr::failwith()  masks plyr::failwith()
## x dplyr::filter()    masks stats::filter()
## x dplyr::id()        masks plyr::id()
## x dplyr::lag()       masks stats::lag()
## x dplyr::mutate()    masks plyr::mutate()
## x dplyr::rename()    masks plyr::rename()
## x dplyr::summarise() masks plyr::summarise()
## x dplyr::summarize() masks plyr::summarize()
```

```
data1<-read_excel("Champo Carpets.xlsx",sheet=2)
data<-data1[1:5000, ]


#1 Pie Chart
ggplot(data, aes(x = "", y = Amount, fill = ITEM_NAME)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar(theta = "y", start = 0)
```

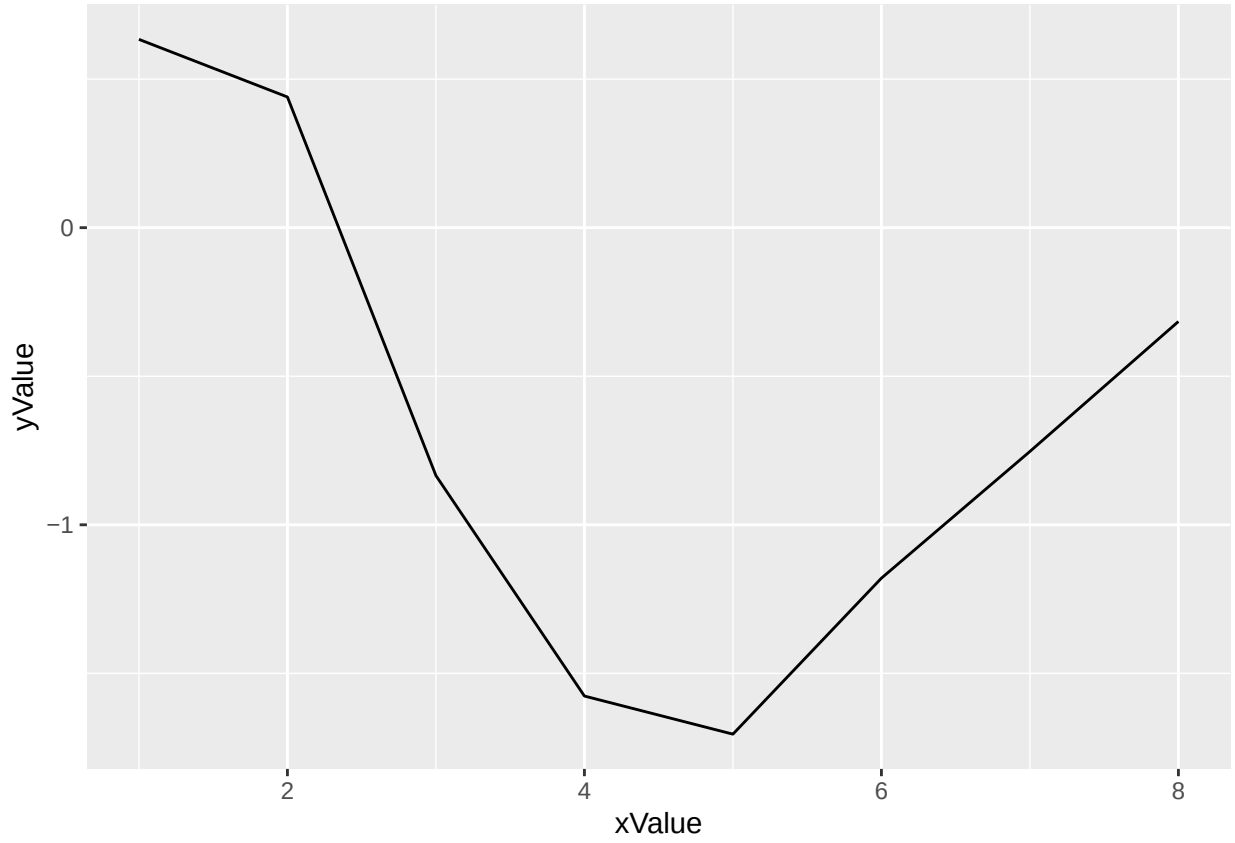



```
#2
ggplot() + geom_line (data = data, mapping=aes(x=CountryName
                                               , y=Amount, fill = Custorderdate),alpha=9)
```

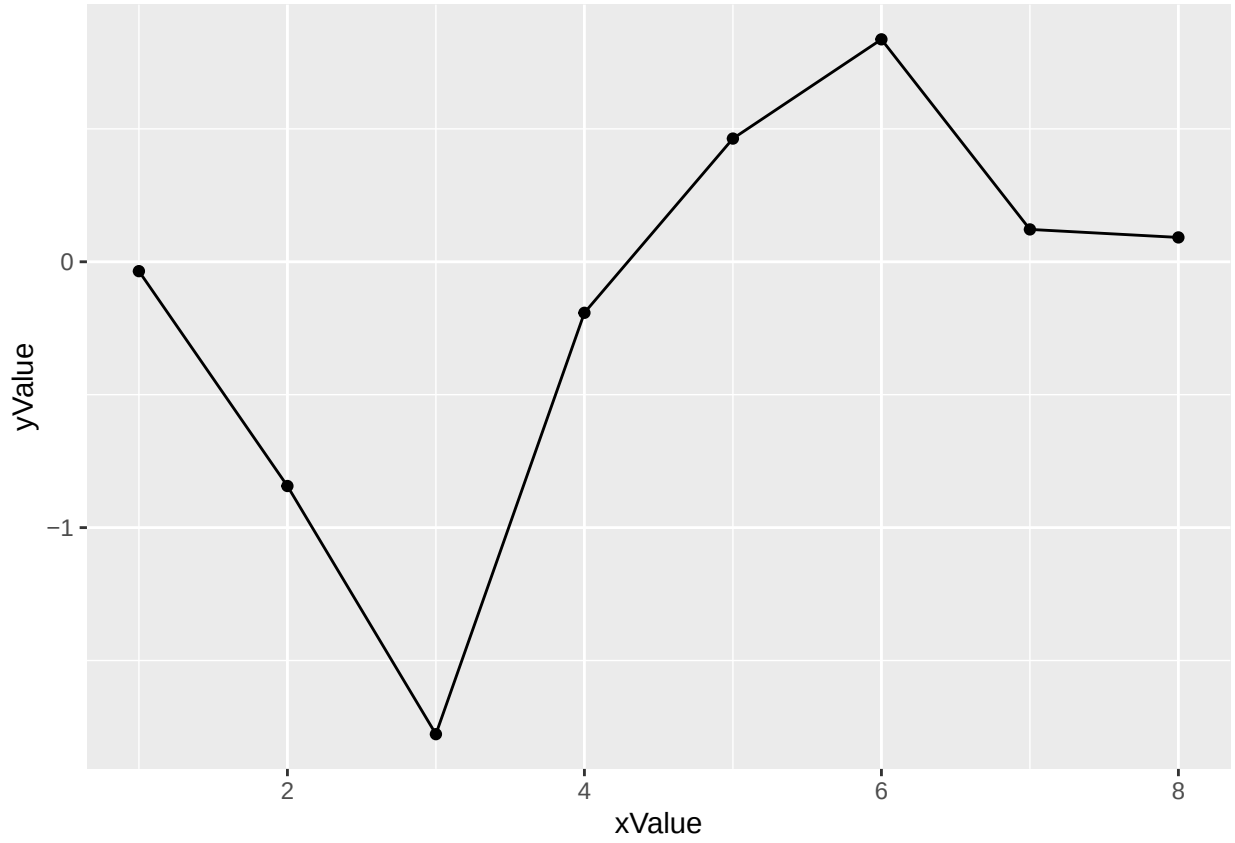
```
## Warning: Ignoring unknown aesthetics: fill
```



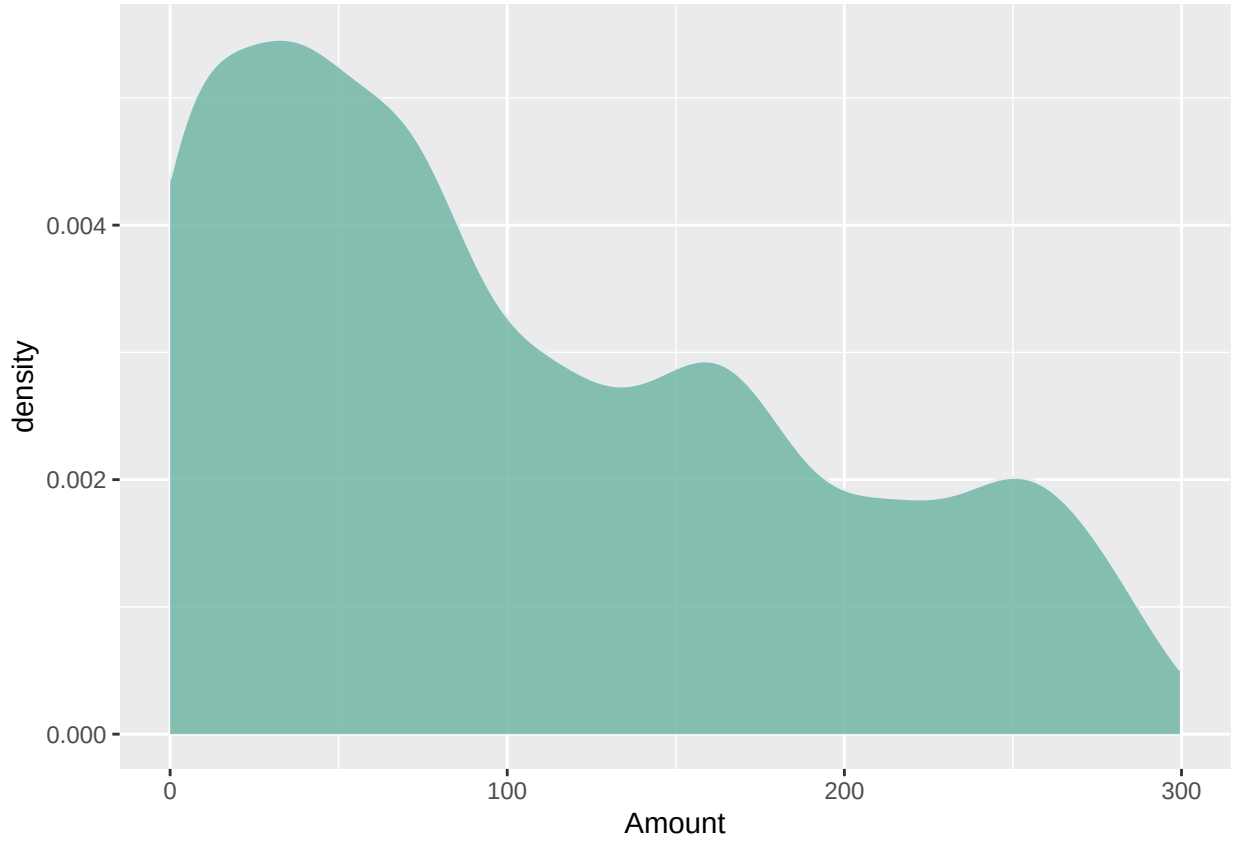

```
#3
library(ggplot2)
xValue <- 1:8
yValue <- cumsum(rnorm(8))
data2 <- data.frame(xValue,yValue)
ggplot(data2, aes(x=xValue, y=yValue)) +  geom_line()
```



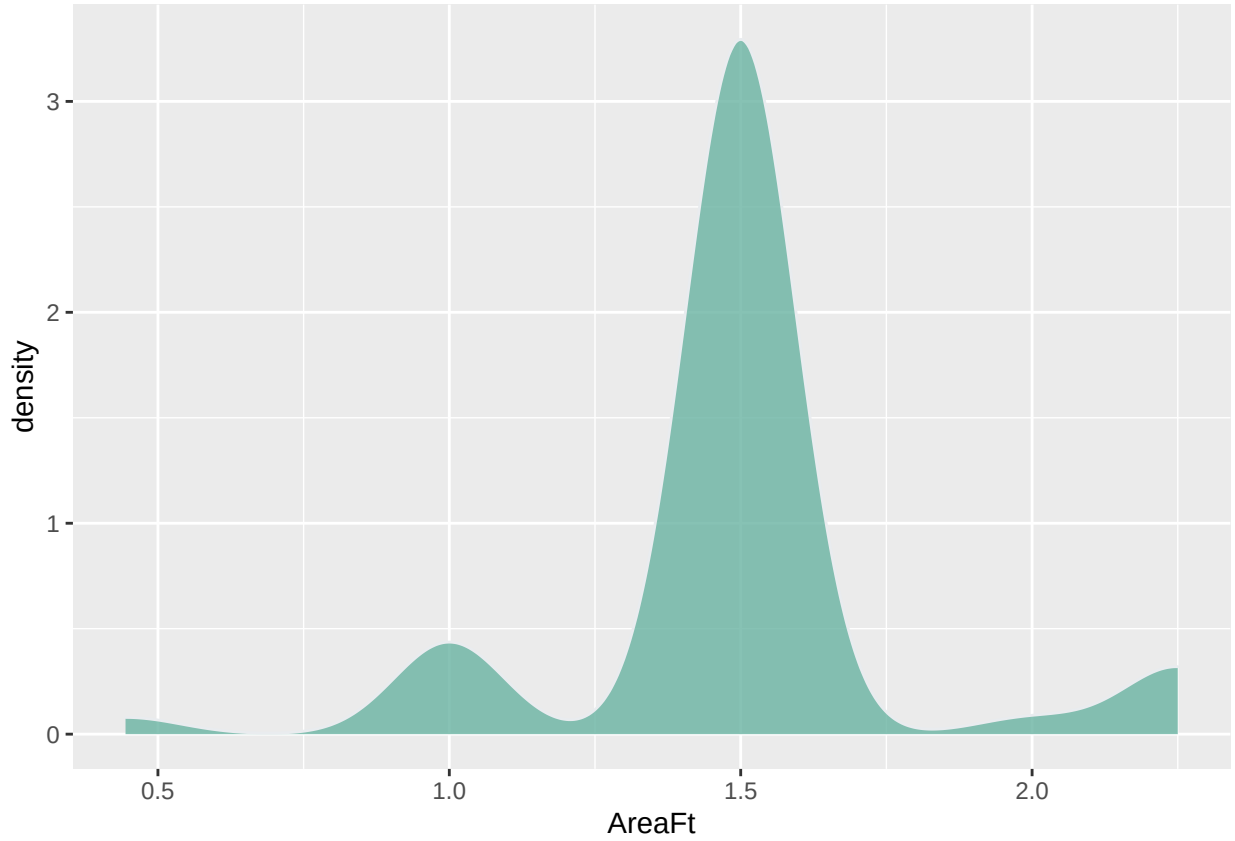

```
#4
library(ggplot2)
xValue <- 1:8
yValue <- cumsum(rnorm(8))
data3 <- data.frame(xValue,yValue)
data3 %>%
  tail(10) %>%
  ggplot( aes(x=xValue, y=yValue)) +
  geom_line() +
  geom_point()
```



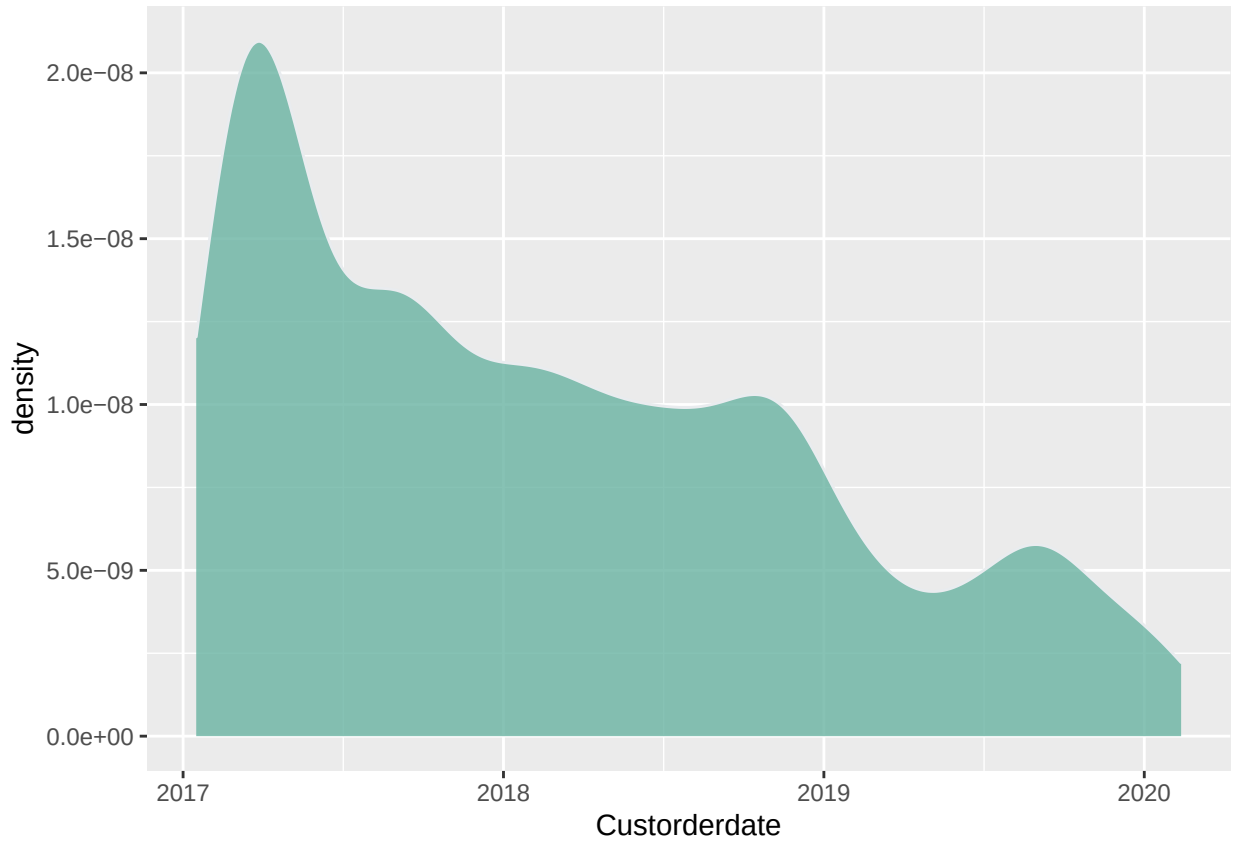

```
#5
data %>%
  filter( Amount<300 ) %>%
  ggplot( aes(x=Amount)) +
  geom_density(fill="#69b3a2", color="#e9ecef", alpha=0.8)
```



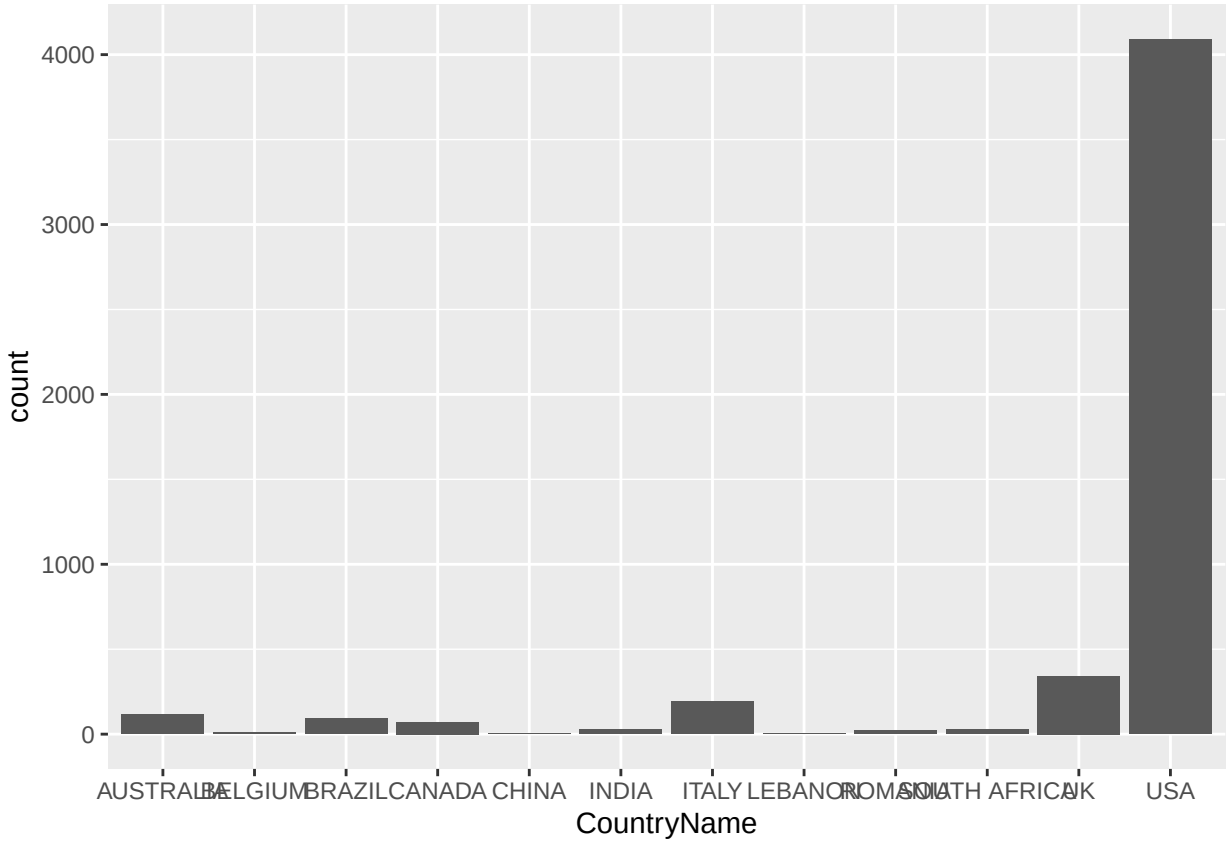

```
#6
data %>%
  filter( AreaFt<3 ) %>%
  ggplot( aes(x=AreaFt)) +
  geom_density(fill="#69b3a2", color="#e9ecef", alpha=0.8)
```



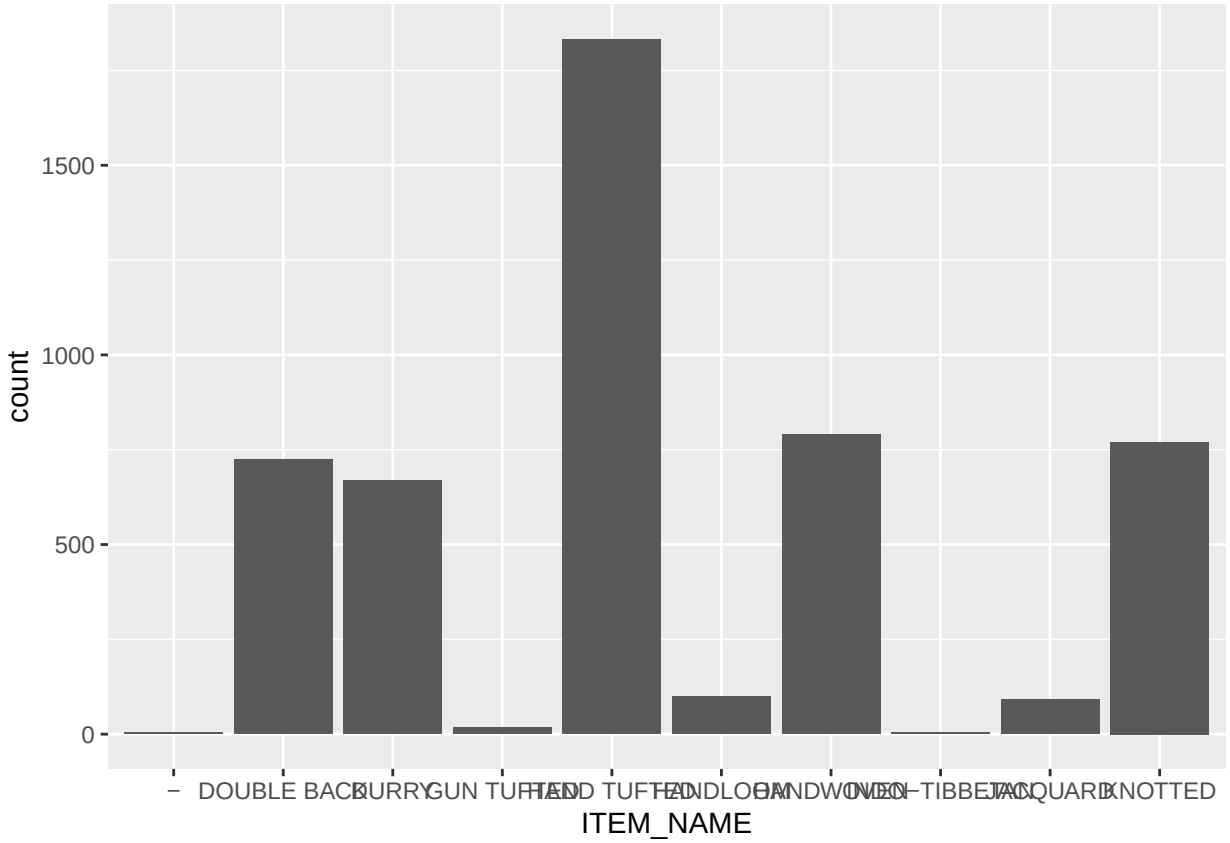

```
data %>%
  ggplot( aes(x=Custorderdate)) +
  geom_density(fill="#69b3a2", color="#e9ecef", alpha=0.8)
```



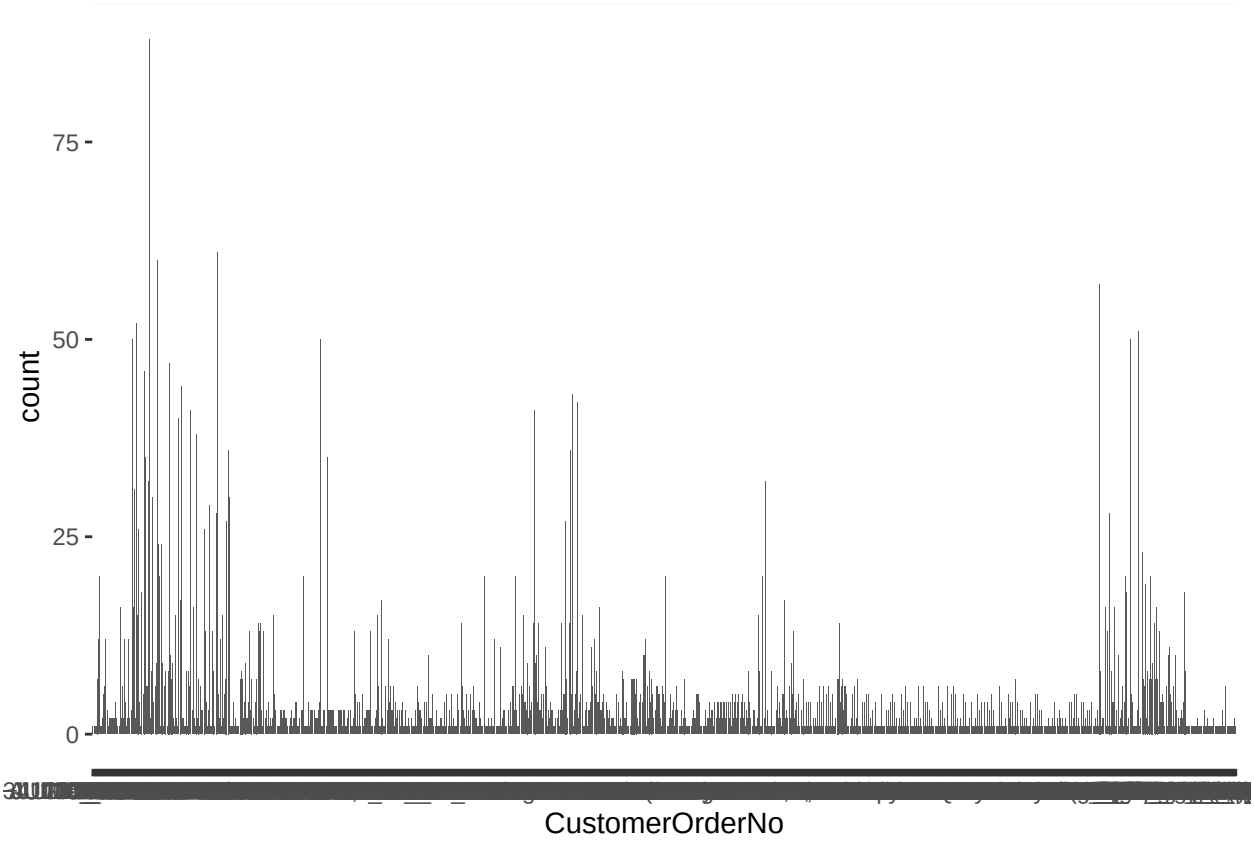

```
#7
ggplot(data = data) +  geom_bar(mapping = aes(x = CountryName))
```



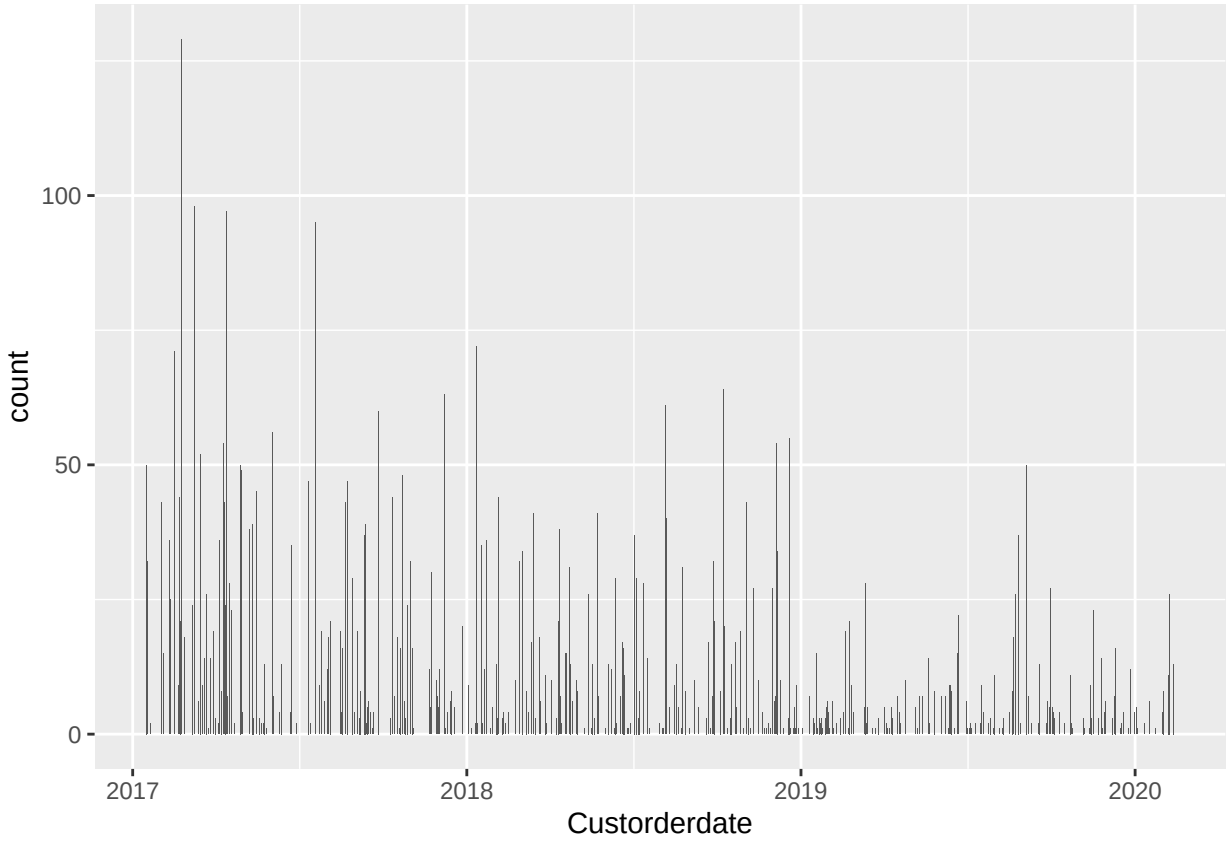

```
ggplot(data = data) +  geom_bar(mapping = aes(x = ITEM_NAME))
```



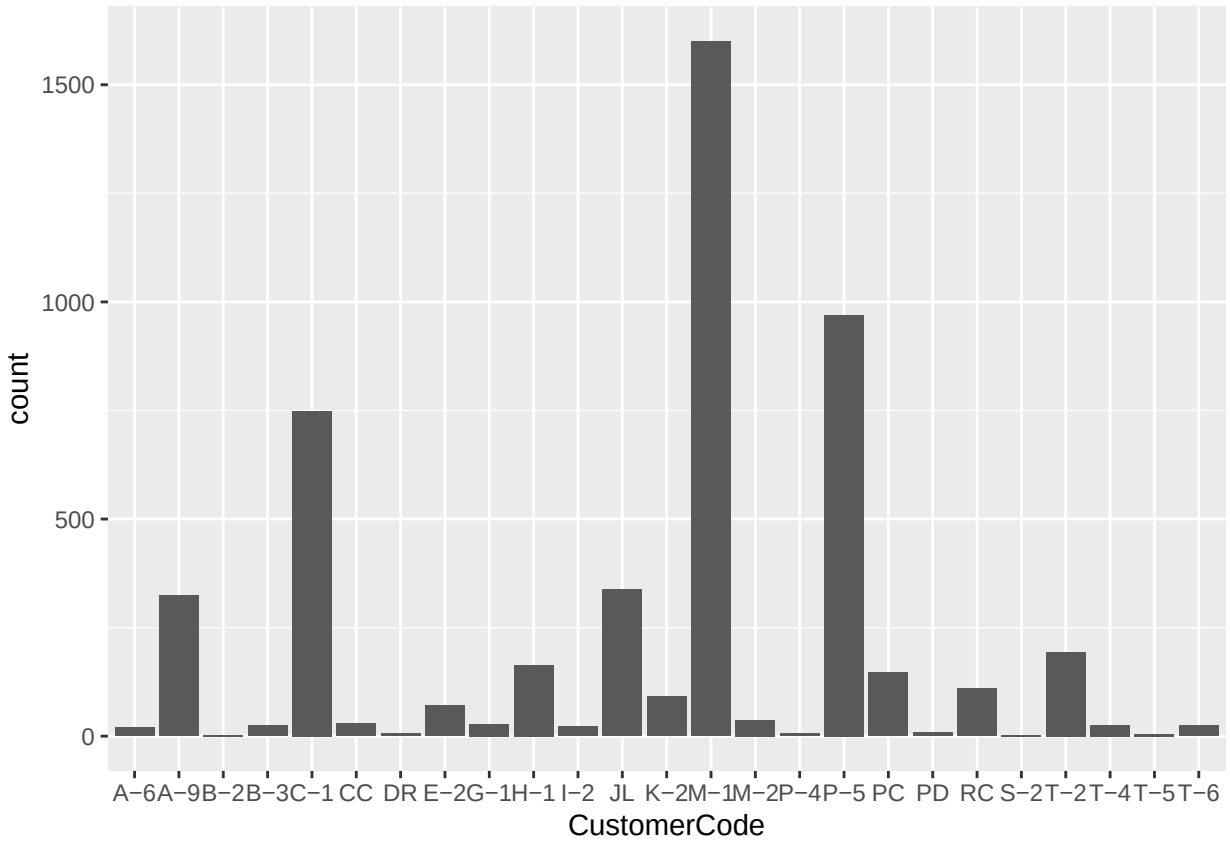

```
#8
ggplot(data = data) +  geom_bar(mapping = aes(x =CustomerOrderNo))
```



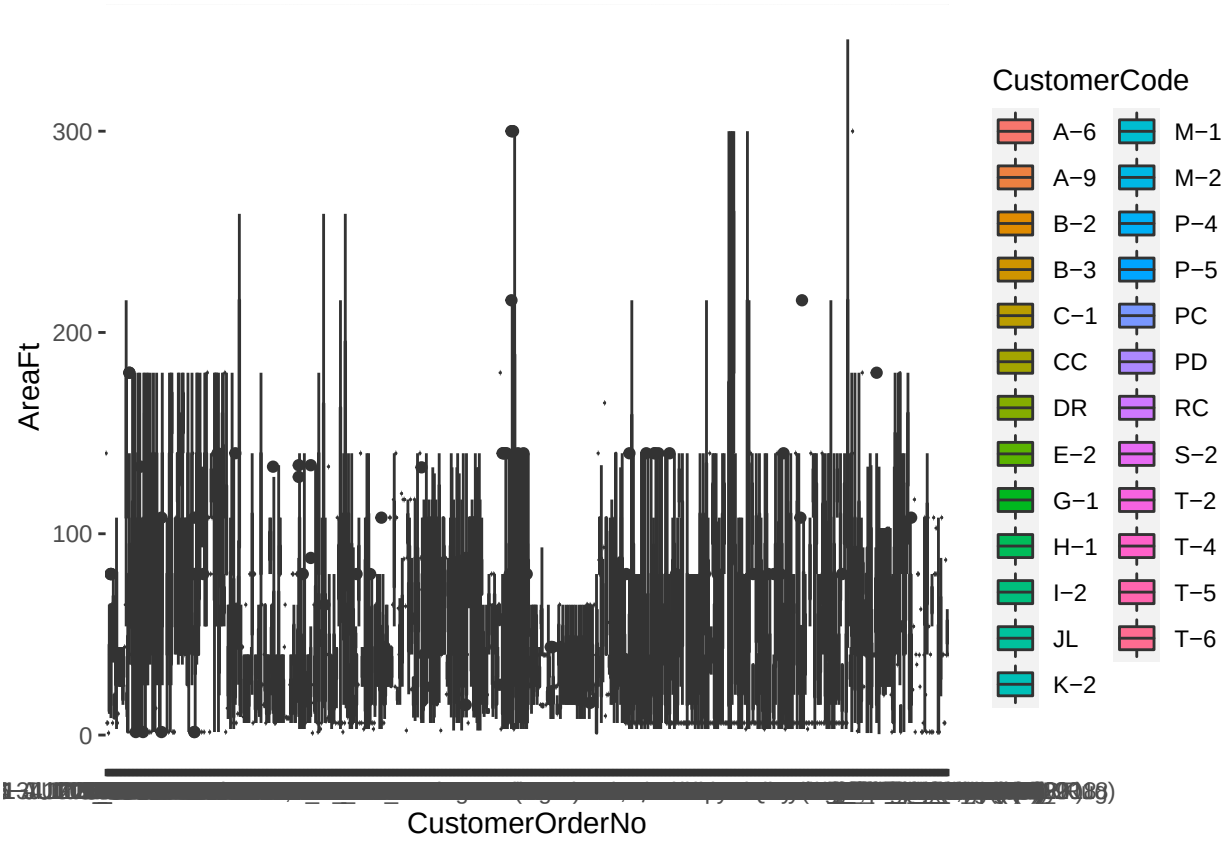

```
#9
ggplot(data = data) +  geom_bar(mapping = aes(x = Custorderdate))
```




```
#10
ggplot(data = data) +  geom_bar(mapping = aes(x = CustomerCode
))
```




```
#11
ggplot() + geom_boxplot(data=data, mapping=aes(x=CustomerOrderNo, y=AreaFt, fill = CustomerCode),alpha=
```



From the visuals we can say,

1. As Compared to other countries sales in US are higher and Majority of sales are coming from USA.
2.TABLE TUFFTED item has more sales
3.Highest contributor of revenue is the "Gun Tufted"
carpet type followed by Jackquardand & Handloom has the
lowest sales among all types
4. Number of orders increased were highest in
2017 and gradually decreased from 2018 till 2020.

##Q2
Champo Carpets can use various classification
models to identify the important attributes
that determine the conversion of samples sent
to the customers. This models will enable us
to assign test data into specific categories
and recognize specific entities within the
dataset to draw conclusions on how those
entities should be labeled or defined. The
model's that we think would be most suitable
are used in Question 3 below.

##Q4
The data strategy for constructing a recommender system

for this would be based on client segmentation using
clustering.
1. Customers who order regularly may be offered samples
depending on their previous orders.
2. Customers who buy on a regular basis but not often
enough might be recommended based on the order
information of similar customers with a good sales
record.
3. We may categorize and separate them based on
"Customer Code" using information from "Raw Data
Order and Sample."
4. Buyers from different countries have different
interests in carpet kinds and styles, therefore
recommendations would differ by country based
on the sales and revenue generated for each of
the carpet styles and country

##Q5
1.We can develop K-means clustering & emphasize
on optimal number of clusters, significant
variables, and cluster characteristics for
segmenting Champo Carpets's customers.
2.The Euclidean distance is calculated in
the K-means clustering method to interpret the
distance between data points. The elbow technique
may be used to determine the number of clusters K.
3.We also can develop dendrogram by the method
of hierarchical clustering to find similar
clusters and compare clusters and it's characteristics
for segmentation.

```r
Data<-read_excel("Champo Carpets.xlsx",sheet=6)

DF_KM<-data.frame(Data$`Sum of QtyRequired`,Data$`Sum of TotalArea`,Data$`Sum of Amount`,Data$DURRY,Data
head(DF_KM)
```

```
##   Data..Sum.of.QtyRequired. Data..Sum.of.TotalArea. Data..Sum.of.Amount.
## 1                      2466               139.5900            185404.10
## 2                       131              2086.0000              6247.46
## 3                     18923             53625.6544           1592079.79
## 4                       624               202.8987             14811.16
## 5                       464              8451.5625             58626.87
## 6                       692              3244.2500             26242.50
##   Data.DURRY Data.HANDLOOM Data..DOUBLE.BACK. Data.JACQUARD Data..HAND.TUFTED.
## 1       1021          1445                  0            0                   0
## 2          0             0                 25          106                   0
## 3       3585             0                175          714               11716
## 4        581             0                  0            2                   0
## 5          0             0                459            5                   0
## 6         80           102                  0            0                 510
##   Data..HAND.WOVEN. Data.KNOTTED Data..GUN.TUFTED. Data..Powerloom.Jacquard.
```

```
## 1                 0            0                 0                         0
## 2                 0            0                 0                         0
## 3              2116          617                 0                         0
## 4                41            0                 0                         0
## 5                 0            0                 0                         0
## 6                 0            0                 0                         0
##   Data..INDO.TEBETAN.
## 1                   0
## 2                   0
## 3                   0
## 4                   0
## 5                   0
## 6                   0
```

```
str(DF_KM)
```

```
## 'data.frame':    45 obs. of  13 variables:
##  $ Data..Sum.of.QtyRequired.: num  2466 131 18923 624 464 ...
##  $ Data..Sum.of.TotalArea.  : num  140 2086 53626 203 8452 ...
##  $ Data..Sum.of.Amount.     : num  185404 6247 1592080 14811 58627 ...
##  $ Data.DURRY               : num  1021 0 3585 581 0 ...
##  $ Data.HANDLOOM            : num  1445 0 0 0 0 ...
##  $ Data..DOUBLE.BACK.       : num  0 25 175 0 459 0 0 0 0 3 ...
##  $ Data.JACQUARD            : num  0 106 714 2 5 0 0 0 0 0 ...
##  $ Data..HAND.TUFTED.       : num  0 0 11716 0 0 ...
##  $ Data..HAND.WOVEN.        : num  0 0 2116 41 0 ...
##  $ Data.KNOTTED             : num  0 0 617 0 0 0 453 0 0 0 ...
##  $ Data..GUN.TUFTED.        : num  0 0 0 0 0 0 0 0 0 19 ...
##  $ Data..Powerloom.Jacquard.: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Data..INDO.TEBETAN.      : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
summary(DF_KM)
```

```
##  Data..Sum.of.QtyRequired. Data..Sum.of.TotalArea. Data..Sum.of.Amount.
##  Min.   :     2            Min.   :     1.35       Min.   :     329
##  1st Qu.:   565            1st Qu.:   376.77       1st Qu.:   39701
##  Median :  1566            Median :  2120.00       Median :  116778
##  Mean   : 12978            Mean   : 13056.59       Mean   :  698210
##  3rd Qu.: 11146            3rd Qu.:  8451.56       3rd Qu.:  426626
##  Max.   :183206            Max.   :209725.22       Max.   :11341053
##    Data.DURRY     Data.HANDLOOM    Data..DOUBLE.BACK. Data.JACQUARD
##  Min.   :     0   Min.   :   0.0   Min.   :   0.0     Min.   :  0.00
##  1st Qu.:     0   1st Qu.:   0.0   1st Qu.:   0.0     1st Qu.:  0.00
##  Median :   289   Median :   0.0   Median :   0.0     Median :  0.00
##  Mean   :  7103   Mean   : 185.5   Mean   : 407.9     Mean   : 89.42
##  3rd Qu.:  1560   3rd Qu.:   0.0   3rd Qu.: 175.0     3rd Qu.: 72.00
##  Max.   :139618   Max.   :3673.0   Max.   :5439.0     Max.   :714.00
##  Data..HAND.TUFTED. Data..HAND.WOVEN.  Data.KNOTTED    Data..GUN.TUFTED.
##  Min.   :    0      Min.   :    0.0   Min.   :   0.0   Min.   :  0.000
##  1st Qu.:    0      1st Qu.:    0.0   1st Qu.:   0.0   1st Qu.:  0.000
##  Median :  510      Median :    0.0   Median :   0.0   Median :  0.000
##  Mean   : 3651      Mean   :  867.7   Mean   : 365.8   Mean   :  8.133
##  3rd Qu.: 3544      3rd Qu.:  269.0   3rd Qu.:  18.0   3rd Qu.:  0.000
```

```
##  Max.   :60685      Max.   :14314.0   Max.   :9502.0   Max.   :195.000
##  Data..Powerloom.Jacquard. Data..INDO.TEBETAN.
##  Min.   :   0.0            Min.   : 0.0000
##  1st Qu.:   0.0            1st Qu.: 0.0000
##  Median :   0.0            Median : 0.0000
##  Mean   : 216.7            Mean   : 0.7111
##  3rd Qu.:   0.0            3rd Qu.: 0.0000
##  Max.   :9753.0            Max.   :20.0000
```

```
library(dplyr)
myscale <- function(x) {
  (x - min(x)) / (max(x) - min(x))
}
SET <- DF_KM %>% mutate_if(is.numeric, myscale)
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.1.3
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
# Convert vector to numeric
DF_KM <- as.numeric(DF_KM$`Row Labels`)
# Convert vector to numeric <- DF[!is.na(DF)] # Remove NA values from vector
Data <- na.omit(DF_KM)

head(DF_KM)
```

```
## numeric(0)
```

```
km1 <- kmeans(SET, centers = 2, nstart = 100)
str(km1)
```

```
## List of 9
##  $ cluster     : int [1:45] 1 1 1 1 1 1 1 1 1 1 ...
##  $ centers     : num [1:2, 1:13] 0.0436 0.4516 0.0316 0.491 0.0474 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:2] "1" "2"
##   .. ..$ : chr [1:13] "Data..Sum.of.QtyRequired." "Data..Sum.of.TotalArea." "Data..Sum.of.Amount." "
##  $ totss       : num 17.7
##  $ withinss    : num [1:2] 8.24 4.76
##  $ tot.withinss: num 13
##  $ betweenss   : num 4.7
##  $ size        : int [1:2] 42 3
##  $ iter        : int 1
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

```
km1
```

```
## K-means clustering with 2 clusters of sizes 42, 3
##
```

```
## Cluster means:
##   Data..Sum.of.QtyRequired. Data..Sum.of.TotalArea. Data..Sum.of.Amount.
## 1                0.04362532              0.03162803           0.04739456
## 2                0.45163133              0.49095247           0.25954037
##   Data.DURRY Data.HANDLOOM Data..DOUBLE.BACK. Data.JACQUARD Data..HAND.TUFTED.
## 1 0.02619529    0.02238342         0.03600977     0.1020742         0.05203569
## 2 0.39638394    0.44432344         0.62082491     0.4495798         0.17390898
##   Data..HAND.WOVEN. Data.KNOTTED Data..GUN.TUFTED. Data..Powerloom.Jacquard.
## 1        0.04594403  0.008346614        0.02087912                 0.0000000
## 2        0.26605654  0.460534624        0.33333333                 0.3333333
##   Data..INDO.TEBETAN.
## 1          0.03809524
## 2          0.00000000
##
## Clustering vector:
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 1 1 1
## [39] 1 1 1 1 1 1 1
##
## Within cluster sum of squares by cluster:
## [1] 8.237222 4.760036
##  (between_SS / total_SS =  26.6 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
km1$cluster
```

```
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 1 1 1
## [39] 1 1 1 1 1 1 1
```

```
km1$centers
```

```
##   Data..Sum.of.QtyRequired. Data..Sum.of.TotalArea. Data..Sum.of.Amount.
## 1                0.04362532              0.03162803           0.04739456
## 2                0.45163133              0.49095247           0.25954037
##   Data.DURRY Data.HANDLOOM Data..DOUBLE.BACK. Data.JACQUARD Data..HAND.TUFTED.
## 1 0.02619529    0.02238342         0.03600977     0.1020742         0.05203569
## 2 0.39638394    0.44432344         0.62082491     0.4495798         0.17390898
##   Data..HAND.WOVEN. Data.KNOTTED Data..GUN.TUFTED. Data..Powerloom.Jacquard.
## 1        0.04594403  0.008346614        0.02087912                 0.0000000
## 2        0.26605654  0.460534624        0.33333333                 0.3333333
##   Data..INDO.TEBETAN.
## 1          0.03809524
## 2          0.00000000
```

```
km1$withinss
```

```
## [1] 8.237222 4.760036
```

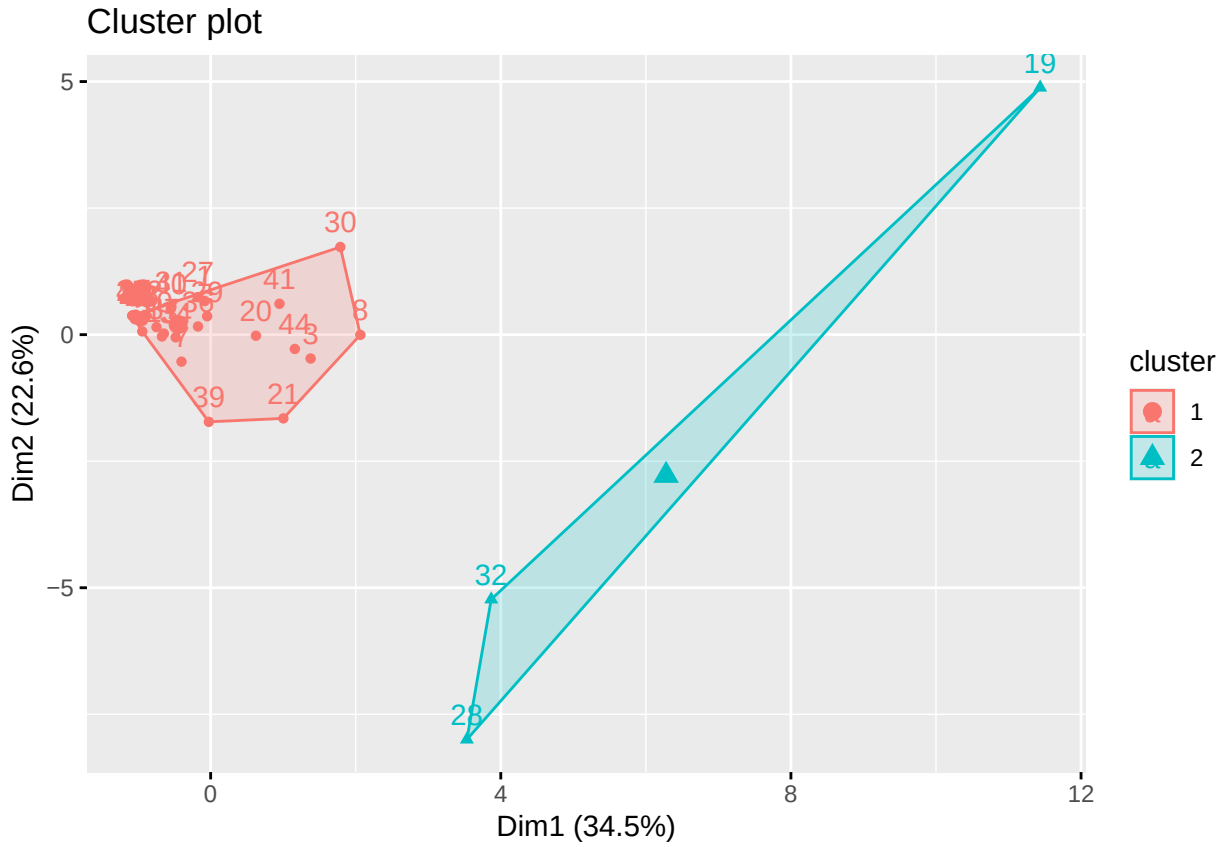
```
km1$betweenss
```

```
## [1] 4.699134
```

```
km1$size
```

```
## [1] 42  3
```

```
fviz_cluster(km1,data=SET)
```



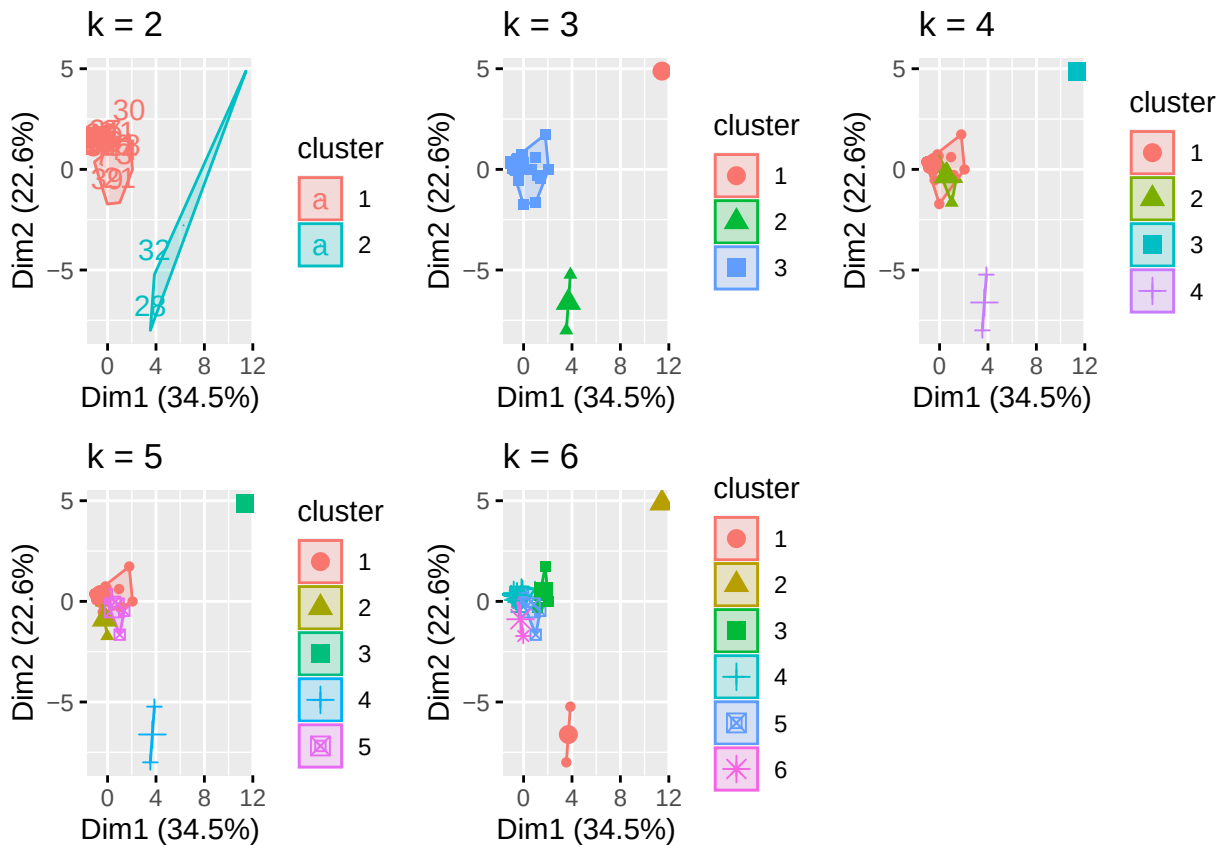

```
p1 <- fviz_cluster(km1, geom = "text", data = SET) + ggtitle("k = 2")

km2 <- kmeans(SET, centers = 3, nstart = 100)
km3 <- kmeans(SET, centers = 4, nstart = 100)
km4 <- kmeans(SET, centers = 5, nstart = 100)
km5 <- kmeans(SET, centers = 6, nstart = 100)

# plots to compare
p2 <- fviz_cluster(km2, geom = "point", data = SET) + ggtitle("k = 3")
p3 <- fviz_cluster(km3, geom = "point", data = SET) + ggtitle("k = 4")
p4 <- fviz_cluster(km4, geom = "point", data = SET) + ggtitle("k = 5")
p5 <- fviz_cluster(km5, geom = "point", data = SET) + ggtitle("k = 6")

library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 4.1.2
```

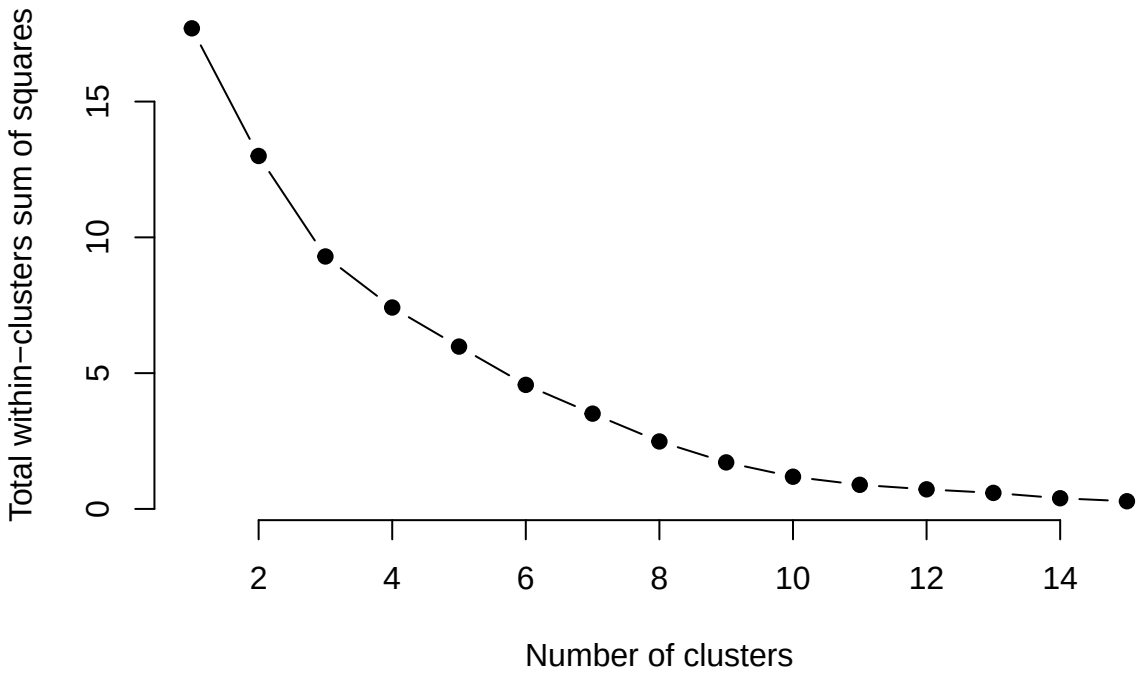
```
##
## Attaching package: 'gridExtra'
```

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

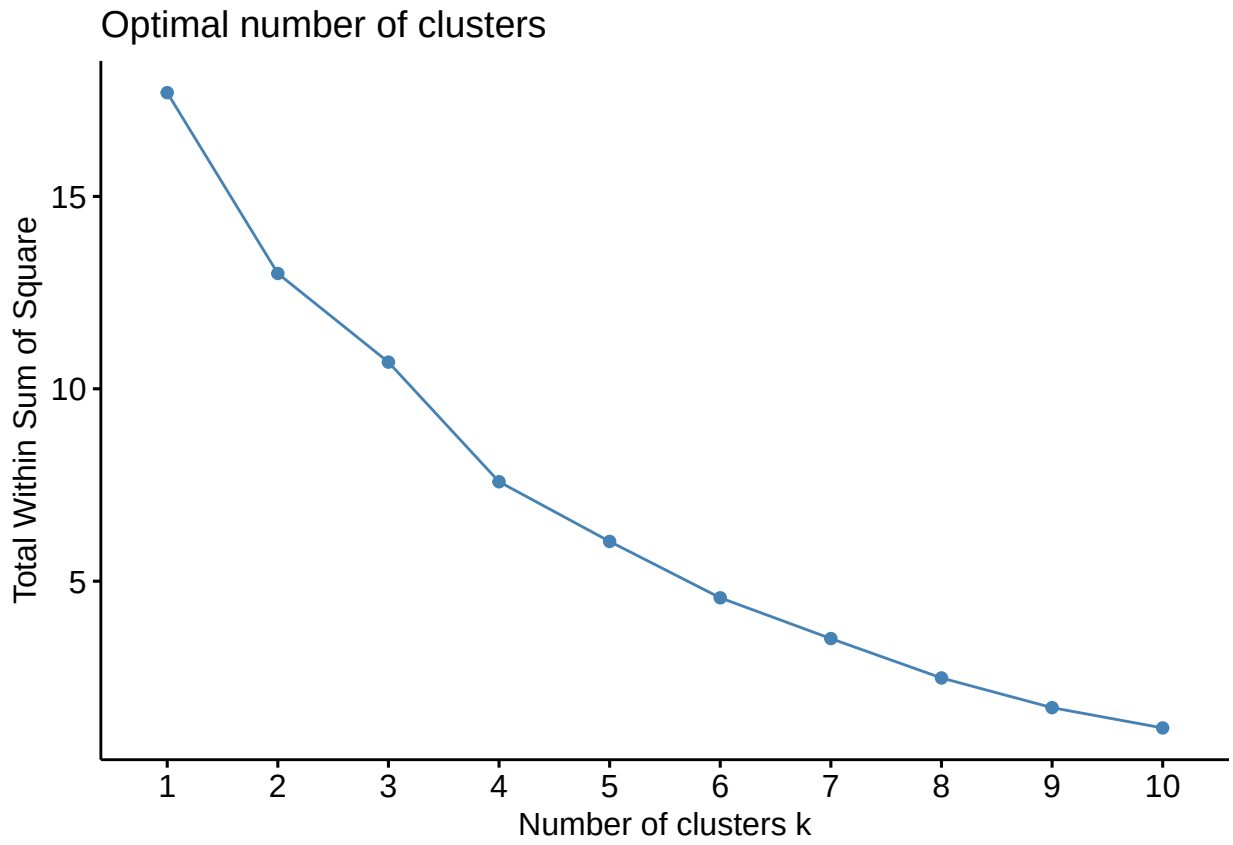
```
grid.arrange(p1, p2, p3, p4,p5, nrow = 2)
```



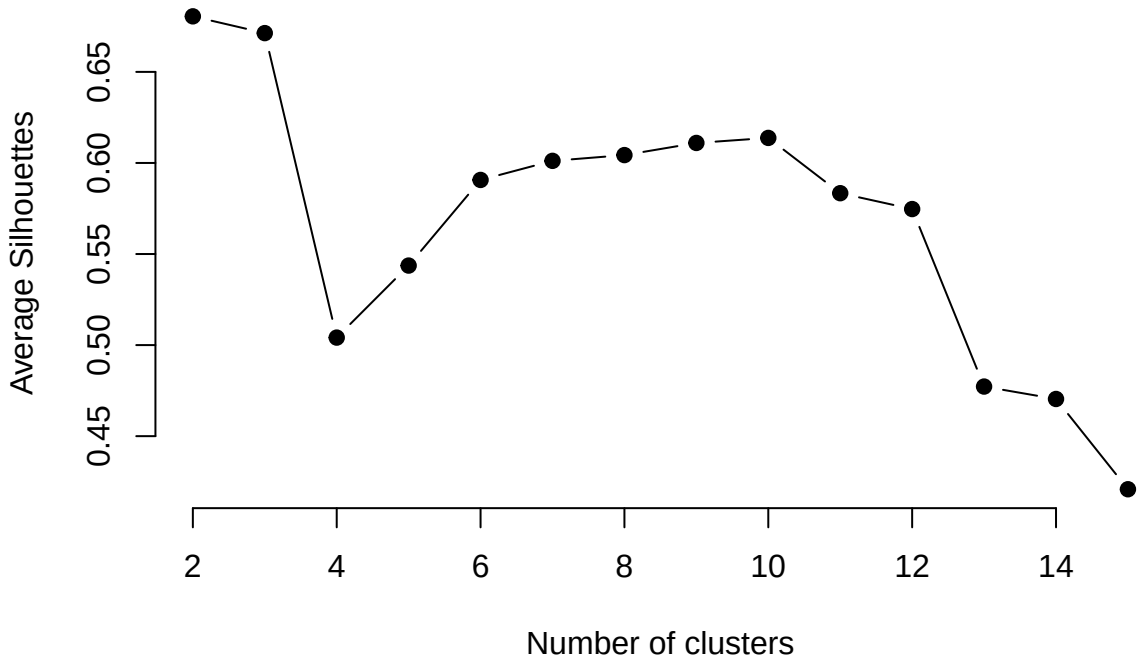

```
set.seed(123)
# function to compute total within-cluster sum of square
wss <- function(k) {
  kmeans(SET, centers = k, nstart = 100)$tot.withinss
}
# Compute and plot wss for k = 1 to k = 15
k.values <- 1:15

# extract wss for 2-15 clusters
library(tidyverse)
wss_values <- map_dbl(k.values, wss)
plot(k.values, wss_values,
     type="b", pch = 19, frame = FALSE,
     xlab="Number of clusters",
     ylab="Total within-clusters sum of squares")
```



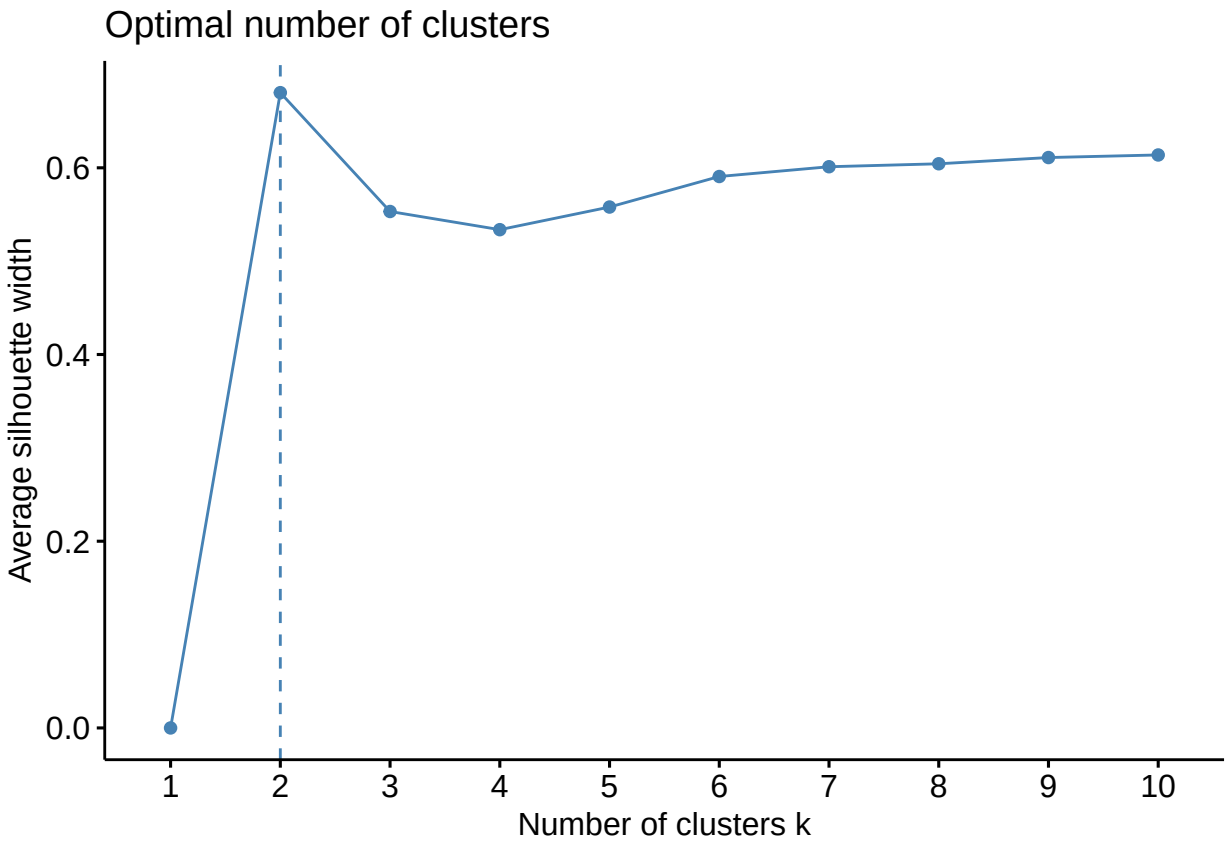

```
set.seed(123)
fviz_nbclust(SET, kmeans, method = "wss")
```



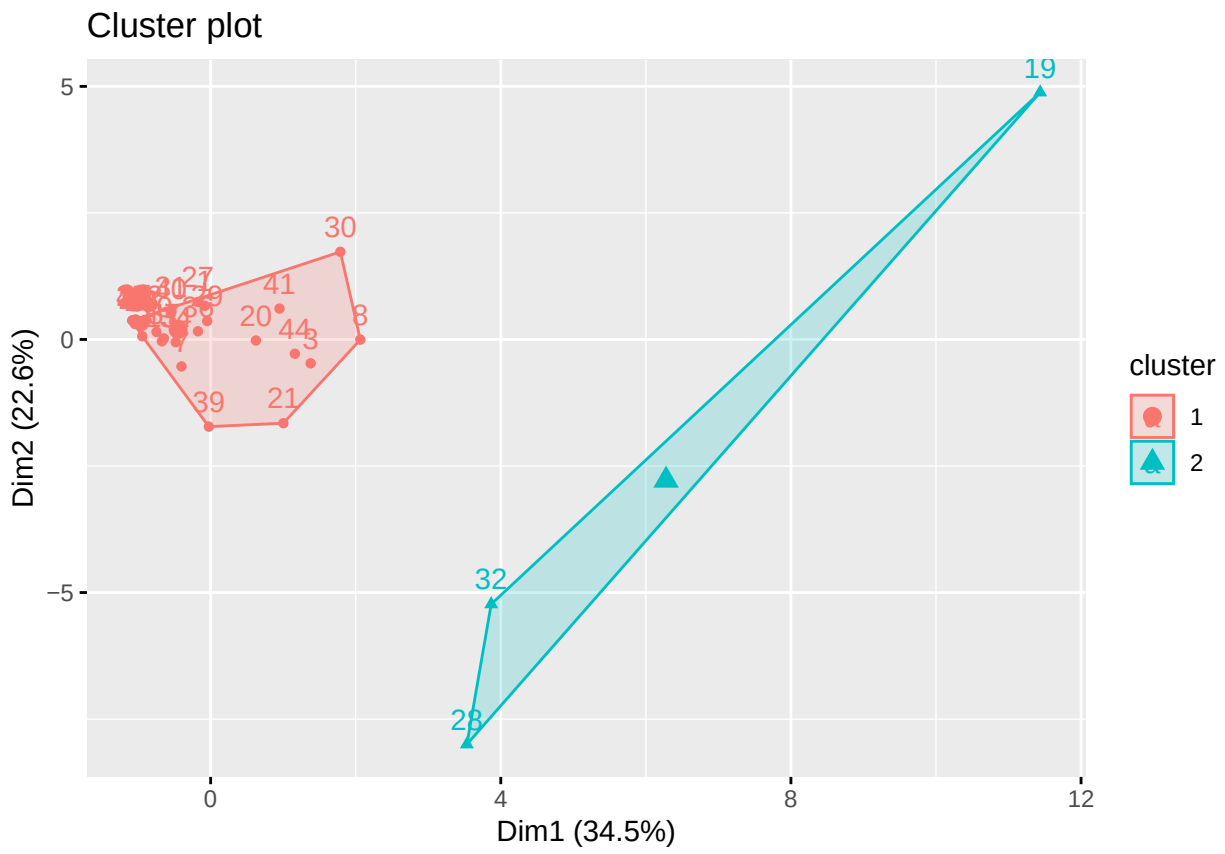

```r
# function to compute average silhouette for k clusters
library(cluster)
avgsil <- function(k) {
  kmModel <- kmeans(SET, centers = k, nstart = 100)
  ss <- silhouette(kmModel$cluster, dist(SET))
  mean(ss[, 3])
}
# Compute and plot wss for k = 2 to k = 15
k.values <- 2:15
# extract avg silhouette for 2-15 clusters
avgsil_values <- map_dbl(k.values, avgsil)
plot(k.values, avgsil_values,
     type = "b", pch = 19, frame = FALSE,
     xlab = "Number of clusters",
     ylab = "Average Silhouettes")
```




```
fviz_nbclust(SET, kmeans, method = "silhouette")
```




```
fviz_cluster(km1, data = SET)
```




```
SET %>%
  mutate(Cluster = km3$cluster) %>% group_by(Cluster) %>%
  summarise_all("mean")
```

```
## # A tibble: 4 x 14
##   Cluster Data..Sum.of.QtyRequ~ Data..Sum.of.Total~ Data..Sum.of.Amo~ Data.DURRY
##     <int>                 <dbl>                <dbl>             <dbl>      <dbl>
## 1       1                0.0414               0.0271            0.0445     0.0274
## 2       2                0.0603               0.0650            0.0690     0.0176
## 3       3                1                    0.0930            0.335      1
## 4       4                0.177                0.690             0.222      0.0946
## # ... with 9 more variables: Data.HANDLOOM <dbl>, Data..DOUBLE.BACK. <dbl>,
## #   Data.JACQUARD <dbl>, Data..HAND.TUFTED. <dbl>, Data..HAND.WOVEN. <dbl>,
## #   Data.KNOTTED <dbl>, Data..GUN.TUFTED. <dbl>,
## #   Data..Powerloom.Jacquard. <dbl>, Data..INDO.TEBETAN. <dbl>
```

```
##Hierrachical Clutering
library(readxl)
Data<-read_excel("Champo Carpets.xlsx",sheet=6)

DF_KM<-data.frame(Data$`Sum of QtyRequired`,Data$`Sum of TotalArea`,Data$`Sum of Amount`,Data$DURRY,Data
head(DF_KM)
```

```
##   Data..Sum.of.QtyRequired. Data..Sum.of.TotalArea. Data..Sum.of.Amount.
## 1                      2466                139.5900             185404.10
```

```
## 2                      131              2086.0000             6247.46
## 3                    18923             53625.6544          1592079.79
## 4                      624               202.8987            14811.16
## 5                      464              8451.5625            58626.87
## 6                      692              3244.2500            26242.50
##   Data.DURRY Data.HANDLOOM Data..DOUBLE.BACK. Data.JACQUARD Data..HAND.TUFTED.
## 1       1021          1445                  0             0                  0
## 2          0             0                 25           106                  0
## 3       3585             0                175           714              11716
## 4        581             0                  0             2                  0
## 5          0             0                459             5                  0
## 6         80           102                  0             0                510
##   Data..HAND.WOVEN. Data.KNOTTED Data..GUN.TUFTED. Data..Powerloom.Jacquard.
## 1                 0            0                 0                        0
## 2                 0            0                 0                        0
## 3              2116          617                 0                        0
## 4                41            0                 0                        0
## 5                 0            0                 0                        0
## 6                 0            0                 0                        0
##   Data..INDO.TEBETAN.
## 1                   0
## 2                   0
## 3                   0
## 4                   0
## 5                   0
## 6                   0
```

```
str(DF_KM)
```

```
## 'data.frame':    45 obs. of  13 variables:
##  $ Data..Sum.of.QtyRequired.: num  2466 131 18923 624 464 ...
##  $ Data..Sum.of.TotalArea.  : num  140 2086 53626 203 8452 ...
##  $ Data..Sum.of.Amount.     : num  185404 6247 1592080 14811 58627 ...
##  $ Data.DURRY               : num  1021 0 3585 581 0 ...
##  $ Data.HANDLOOM            : num  1445 0 0 0 0 ...
##  $ Data..DOUBLE.BACK.       : num  0 25 175 0 459 0 0 0 0 3 ...
##  $ Data.JACQUARD            : num  0 106 714 2 5 0 0 0 0 0 ...
##  $ Data..HAND.TUFTED.       : num  0 0 11716 0 0 ...
##  $ Data..HAND.WOVEN.        : num  0 0 2116 41 0 ...
##  $ Data.KNOTTED             : num  0 0 617 0 0 0 453 0 0 0 ...
##  $ Data..GUN.TUFTED.        : num  0 0 0 0 0 0 0 0 0 19 ...
##  $ Data..Powerloom.Jacquard.: num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Data..INDO.TEBETAN.      : num  0 0 0 0 0 0 0 0 0 0 ...
```

```
summary(DF_KM)
```

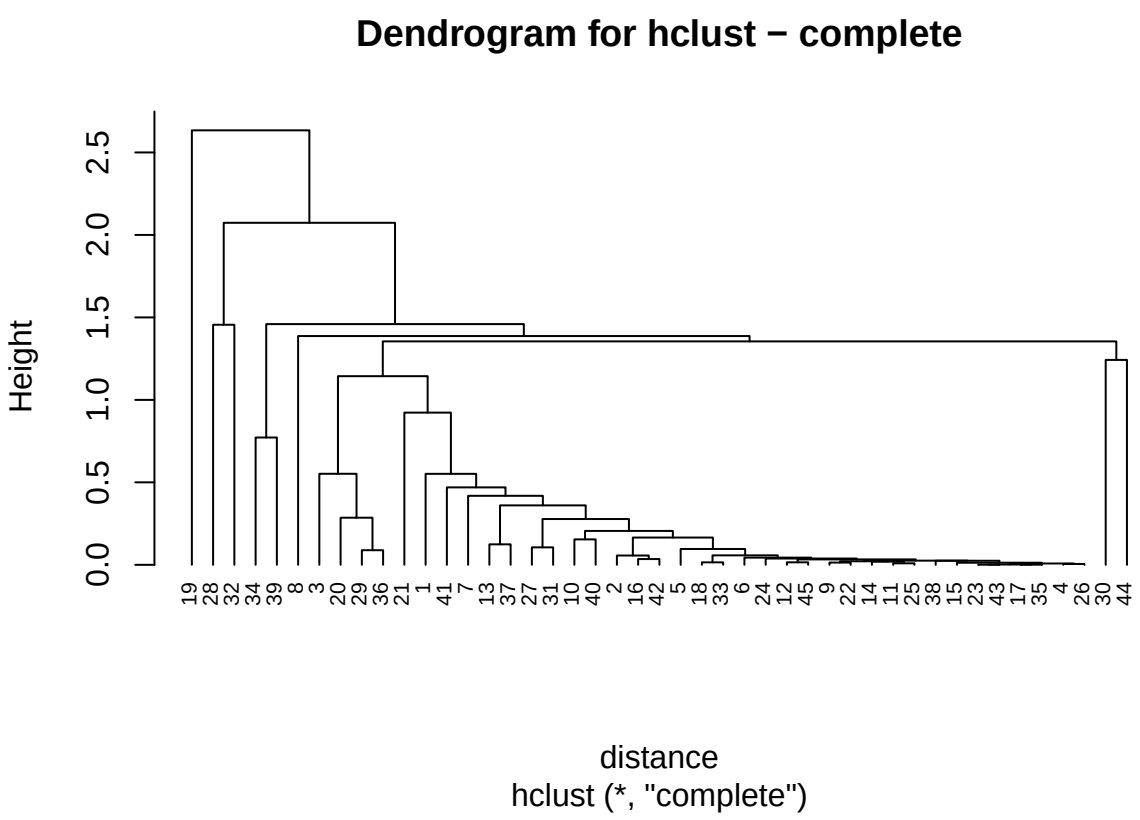
```
##  Data..Sum.of.QtyRequired. Data..Sum.of.TotalArea. Data..Sum.of.Amount.
##  Min.   :     2            Min.   :     1.35       Min.   :     329
##  1st Qu.:   565            1st Qu.:   376.77       1st Qu.:   39701
##  Median :  1566            Median :  2120.00       Median :  116778
##  Mean   : 12978            Mean   : 13056.59       Mean   :  698210
##  3rd Qu.: 11146            3rd Qu.:  8451.56       3rd Qu.:  426626
##  Max.   :183206            Max.   :209725.22       Max.   :11341053
```

```
##    Data.DURRY     Data.HANDLOOM    Data..DOUBLE.BACK. Data.JACQUARD
##  Min.   :     0   Min.   :   0.0   Min.   :   0.0     Min.   :  0.00
##  1st Qu.:     0   1st Qu.:   0.0   1st Qu.:   0.0     1st Qu.:  0.00
##  Median :   289   Median :   0.0   Median :   0.0     Median :  0.00
##  Mean   :  7103   Mean   : 185.5   Mean   : 407.9     Mean   : 89.42
##  3rd Qu.:  1560   3rd Qu.:   0.0   3rd Qu.: 175.0     3rd Qu.: 72.00
##  Max.   :139618   Max.   :3673.0   Max.   :5439.0     Max.   :714.00
##  Data..HAND.TUFTED. Data..HAND.WOVEN.  Data.KNOTTED    Data..GUN.TUFTED.
##  Min.   :    0      Min.   :    0.0   Min.   :   0.0   Min.   :  0.000
##  1st Qu.:    0      1st Qu.:    0.0   1st Qu.:   0.0   1st Qu.:  0.000
##  Median :  510      Median :    0.0   Median :   0.0   Median :  0.000
##  Mean   : 3651      Mean   :  867.7   Mean   : 365.8   Mean   :  8.133
##  3rd Qu.: 3544      3rd Qu.:  269.0   3rd Qu.:  18.0   3rd Qu.:  0.000
##  Max.   :60685      Max.   :14314.0   Max.   :9502.0   Max.   :195.000
##  Data..Powerloom.Jacquard. Data..INDO.TEBETAN.
##  Min.   :   0.0            Min.   : 0.0000
##  1st Qu.:   0.0            1st Qu.: 0.0000
##  Median :   0.0            Median : 0.0000
##  Mean   : 216.7            Mean   : 0.7111
##  3rd Qu.:   0.0            3rd Qu.: 0.0000
##  Max.   :9753.0            Max.   :20.0000
```

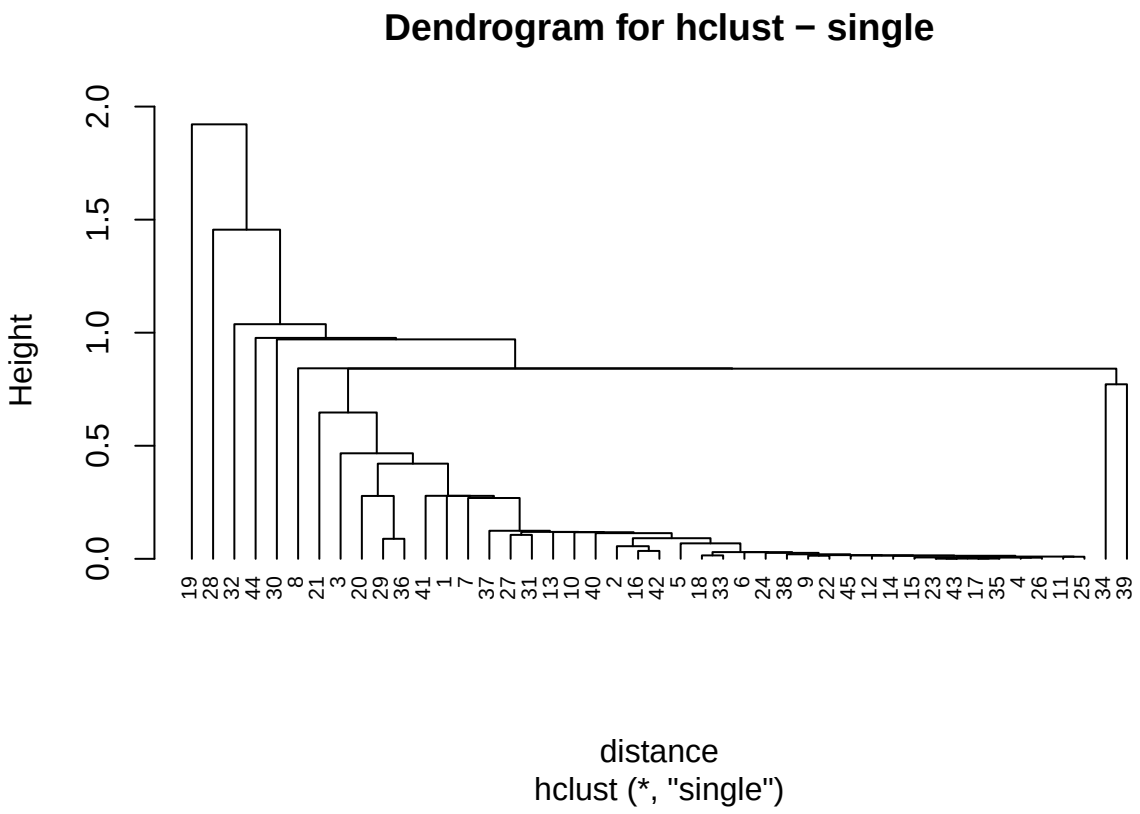
```
library(dplyr)
myscale <- function(x) {
  (x - min(x)) / (max(x) - min(x))
}
SET <- DF_KM %>% mutate_if(is.numeric, myscale)
library(factoextra)
DF_KM <- as.numeric(DF_KM$`Row Labels`)        # Convert vector to numeric
DF_KM <- DF_KM[!is.na(DF_KM)] # Remove NA values from vector
Data <- na.omit(DF_KM)

distance <- dist(SET, method = "euclidean")
head(distance)
```

```
## [1] 0.4211715 1.1436707 0.3938603 0.4047369 0.3664961 0.5025482
```

```
hcomplete <- hclust(distance, method = "complete")
plot(hcomplete, cex = 0.7, hang = -2, main = "Dendrogram for hclust - complete")
```



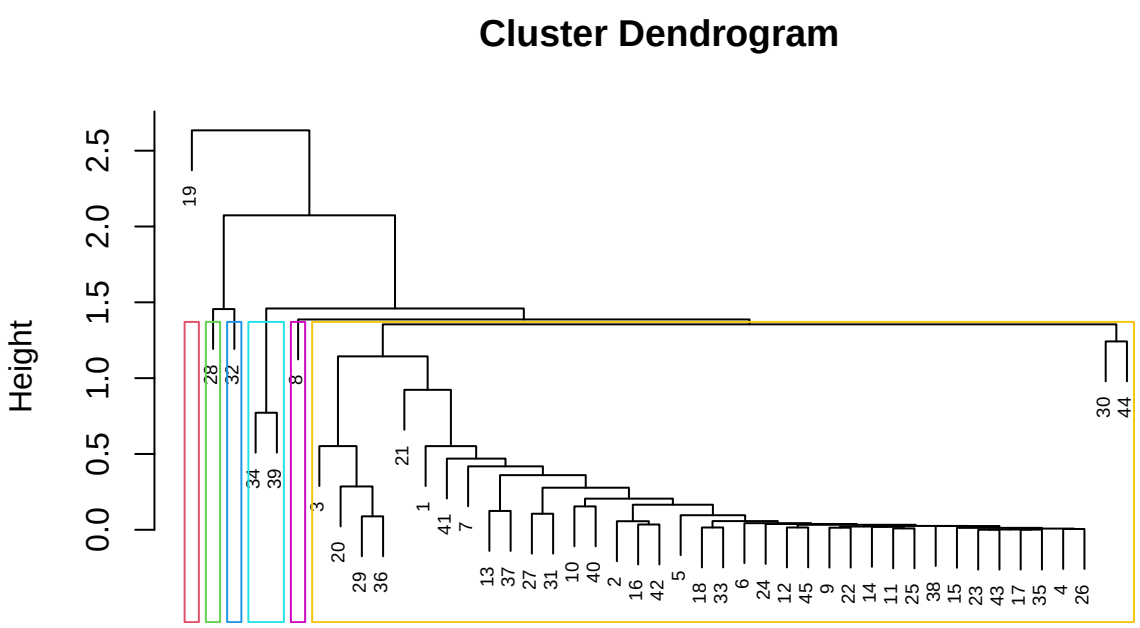

```
hsingle <- hclust(distance, method = "single")
plot(hsingle, cex = 0.7, hang = -2, main = "Dendrogram for hclust - single")
```




```
clusters <- cutree(hcomplete, k =6)
table(clusters)
```

```
## clusters
##  1  2  3  4  5  6
## 39  1  1  1  1  2
```

```
plot(hcomplete, cex = 0.6)
rect.hclust(hcomplete, k =6, border = 2:8)
```



Based on the results from K-means clustering (elbow method), optimal number of clusters are six. Significant Variables: • Customer Code, • Quantity, • Amount, • Design Name, • Item name, • Color, • Shape

Cluster Characteristics: Cluster No. 1:• Cluster with the most revenue-generating Customers from the United States and the United Kingdom make up this cluster. • Customer N-1 purchased a big amount of hand tufted items in a variety of colors, resulting in a substantial rise in revenue. • Customers prefer embroidered designs in Durry goods, which are popular in the United States and Israel. Cluster No. 2: • The most profitable Customers in this cluster are from the United States and Brazil, and they like Durry, Double Back, Jaquard, and Knotted fabrics. Cluster 3: The bulk of customers are from the United States and place small orders; popular items include double back and hand tufted. Cluster4: Cluster with the most revenue-generating potential Customers in this cluster are mostly from Australia and Brazil, and they want hand-tufted, double-backed, knotted items. Cluster 5: The majority of consumers are from Canada and submit small-quantity orders. • A variety of products are popular, including double back, hand tufted, and hand woven, although strictly rectangular forms are recommended. Cluster6: Australian customers produce the largest revenue, with a preference for hand-tufted Bombay print designs from this cluster.

```
##Q7
##Cosine Filtering
library(lsa)
```

```
## Warning: package 'lsa' was built under R version 4.1.3
```

```
## Loading required package: SnowballC
```

```
library(readxl)
```

```
reco <- read_excel("C:/Users/pchitt2/Downloads/Champo Carpets.xlsx",
                              sheet = "Data for Recommendation", range = "A1:U21")

test<-subset(reco, reco$Customer == 'T-2')
cos_cust<-c()
cos <-c()
for(i in 1:nrow(reco)){
  if (as.character(reco[i,][1])==as.character(test[1])) {next
  }
  cos_cust<-c(cos_cust,as.character(reco[i,][1]))
  cos<-c(cos,cosine(as.numeric(test[,-1]),as.numeric(reco[i, ][-1])))
}

nearest<-cos_cust[which.max(cos)]
mat_nonzero_1 <- which(test == 0, arr.ind = T)
mat_nonzero_2 <- which(subset(reco, reco$Customer == nearest) == 0, arr.ind = T)


setdif <- function(a, b) {

  comp <- vector()

  for (i in a) {
    if (i %in% a && !(i %in% b)) {
      comp <- append(comp, i)
    }
  }

  return(comp)
}

recommendations <- setdif(mat_nonzero_1[,2],mat_nonzero_2[,2])
colnames(reco)[recommendations]
```

```
## [1] "Knotted"   "Jacquared" "Purple"    "Navy"      "PINK"      "BLUE"
## [7] "NEUTRAL"   "NAVY"
```

##Q8

Champo Carpets has a diverse portfolio that connect the world. As sampling is a costly operation, Champo should seek for a cost-effective strategy to choose optimal sample designs that would earn the most revenue for the company.The company can identify the key factors that influence order conversion and take additional action. As can be seen, characteristics such as CountryName, QtyRequired, ITEM NAME, ShapName, and AreaFt play a crucial role.• Companies may learn about their clients' distributions by using the K-means clustering approach, which will help them develop better strategies for different segments.