LIBRARIES

```r
library(knitr)
library(readr)
library(ggplot2)
library(ggthemes)
library(wesanderson)
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
```

```r
library(tidyr)
library(magrittr)
```

```
##
## Attaching package: 'magrittr'

## The following object is masked from 'package:tidyr':
##
##     extract
```

```r
library(stats)
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```r
library(factoextra)
```

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

```r
library(DataExplorer)
library("psych")
```

```
##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```r
custom = read_csv("/Users/chiarasaini/Desktop/uni/SL lectures/SL new/Statistical learning project - Chia
```

```
## Rows: 200 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): Gender
## dbl (4): CustomerID, Age, Annual Income (k$), Spending Score (1-100)
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
head(custom)
```

```
## # A tibble: 6 x 5
##   CustomerID Gender   Age `Annual Income (k$)` `Spending Score (1-100)`
##        <dbl> <chr>  <dbl>                <dbl>                    <dbl>
## 1          1 Male      19                   15                       39
## 2          2 Male      21                   15                       81
## 3          3 Female    20                   16                        6
## 4          4 Female    23                   16                       77
## 5          5 Female    31                   17                       40
## 6          6 Female    22                   17                       76
```

```r
custom <- custom %>%
  rename(customer_id = 1,
         gender = 2,
         age = 3,
         annual_income = 4,
         spending_score = 5)
```

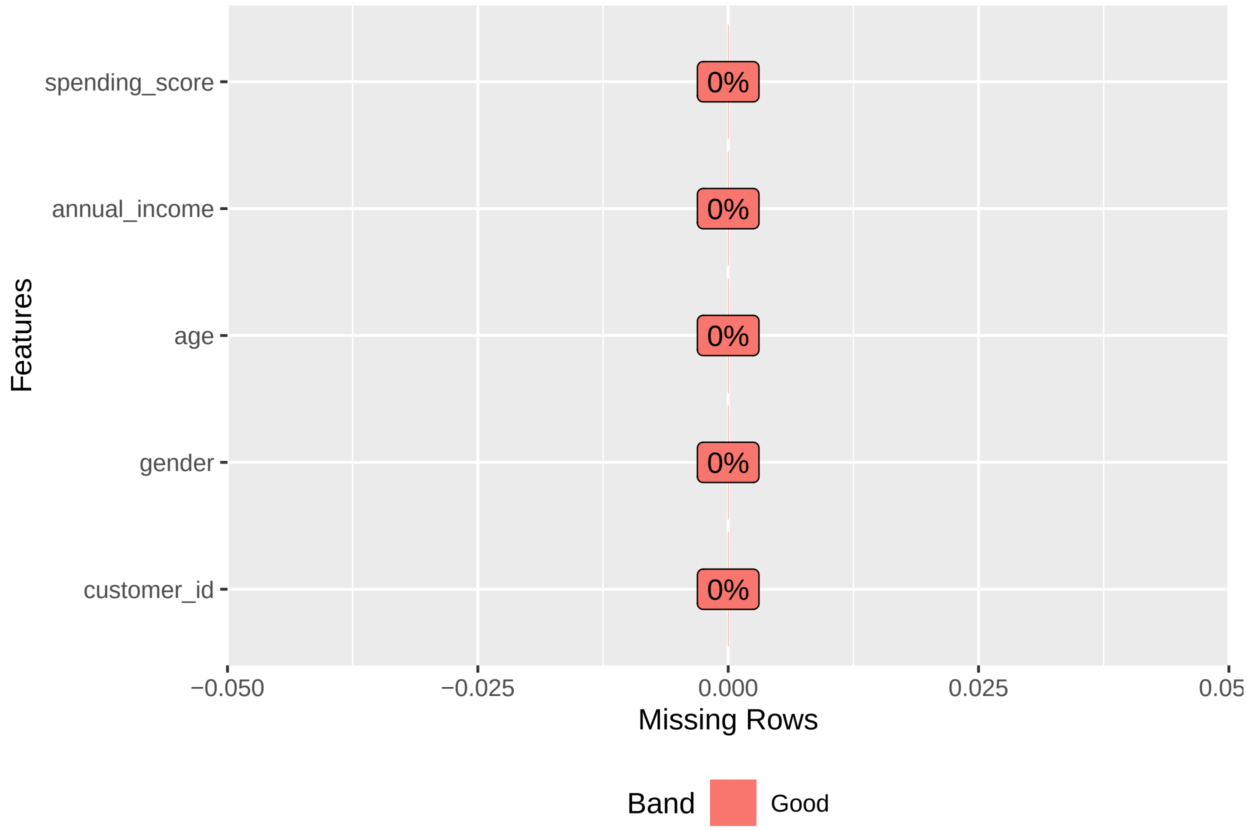
```r
head(custom)
```

```
## # A tibble: 6 x 5
##   customer_id gender   age annual_income spending_score
##         <dbl> <chr>  <dbl>         <dbl>          <dbl>
## 1           1 Male      19            15             39
## 2           2 Male      21            15             81
## 3           3 Female    20            16              6
## 4           4 Female    23            16             77
## 5           5 Female    31            17             40
## 6           6 Female    22            17             76
```

```r
# unique customers
n_distinct(custom$customer_id)
```

```
## [1] 200
```

```r
 plot_missing(custom)
```




```r
# Summary of dataset
summary_custom <- psych::describe(custom, fast = TRUE)
```

```
## Warning in FUN(newX[, i], ...): no non-missing arguments to min; returning Inf

## Warning in FUN(newX[, i], ...): no non-missing arguments to max; returning -Inf
```

```r
summary_custom
```

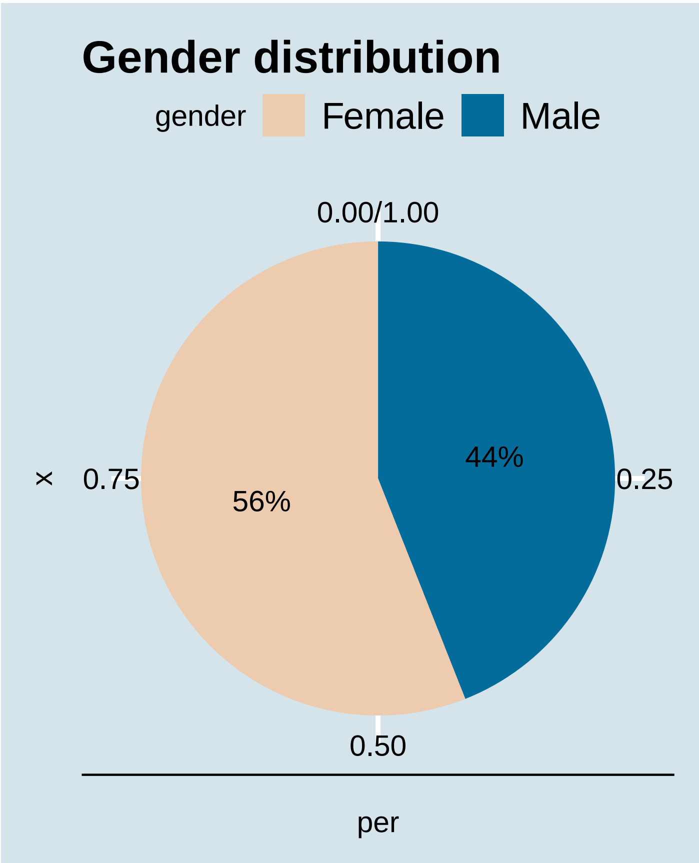
```
##                vars   n   mean    sd min  max range   se
## customer_id       1 200 100.50 57.88   1  200   199 4.09
## gender            2 200    NaN    NA Inf -Inf  -Inf   NA
## age               3 200  38.85 13.97  18   70    52 0.99
## annual_income     4 200  60.56 26.26  15  137   122 1.86
## spending_score    5 200  50.20 25.82   1   99    98 1.83
```

```r
#path_out = '' #Set path
#write.csv(summary_custom,paste(path_out,'summary_custom.csv'))
```

Data analysis

```
theme_set(theme_economist())
```

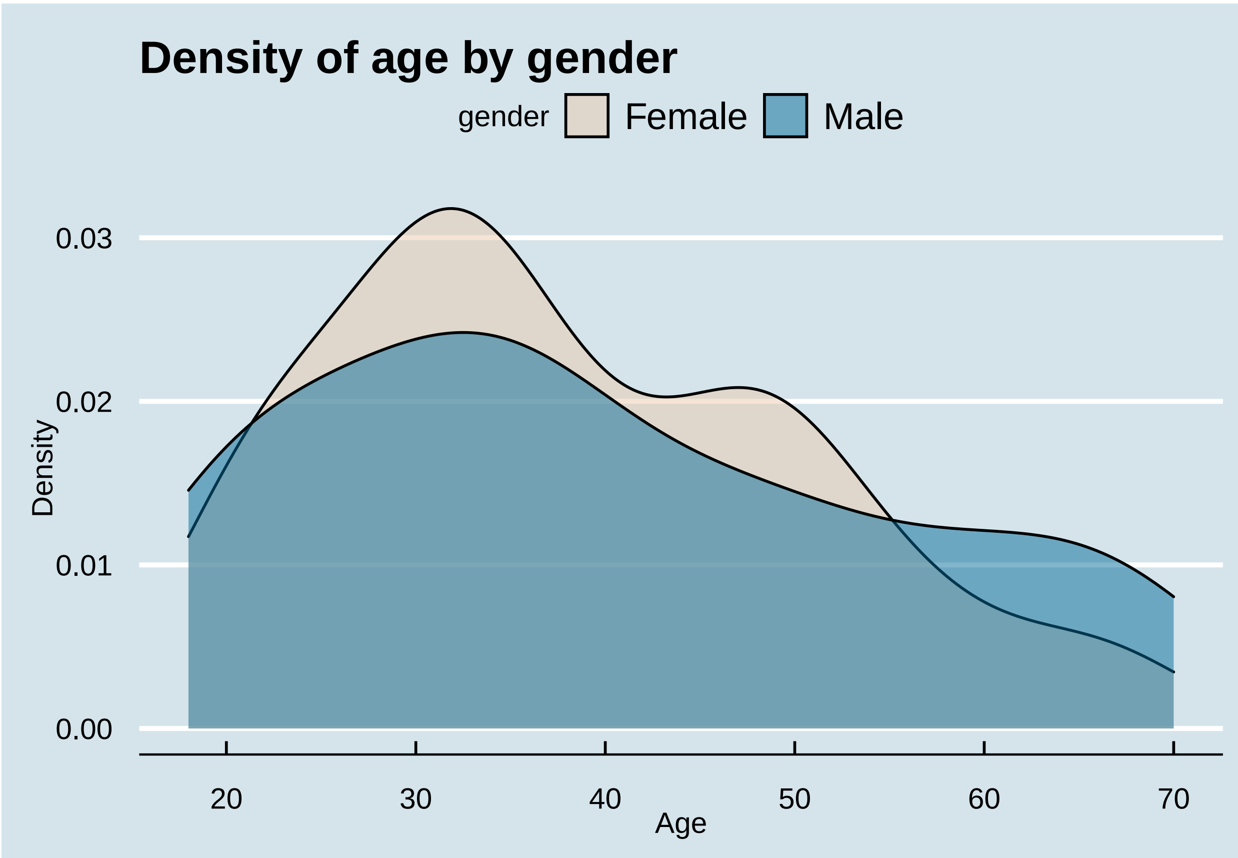
```
df1 <- custom %>%
  group_by(gender) %>%
  count() %>%
  ungroup() %>%
  mutate(per=`n`/sum(`n`)) %>%
  arrange(desc(gender))
df1$label <- scales::percent(df1$per)
gender_plt <- ggplot(data=df1)+
  geom_bar(aes(x="", y=per, fill=gender), stat="identity", width = 1)+
  coord_polar("y", start=0)+
  ggtitle(paste0("Gender distribution")) +
  geom_text(aes(x=1, y = cumsum(per) - per/2, label=label)) +
  scale_fill_manual(values=wes_palette(n=2, name="Darjeeling2"))
gender_plt
```




```
#ggsave("gender_plt.png", plot = gender_plt, path = "") #Set path
```

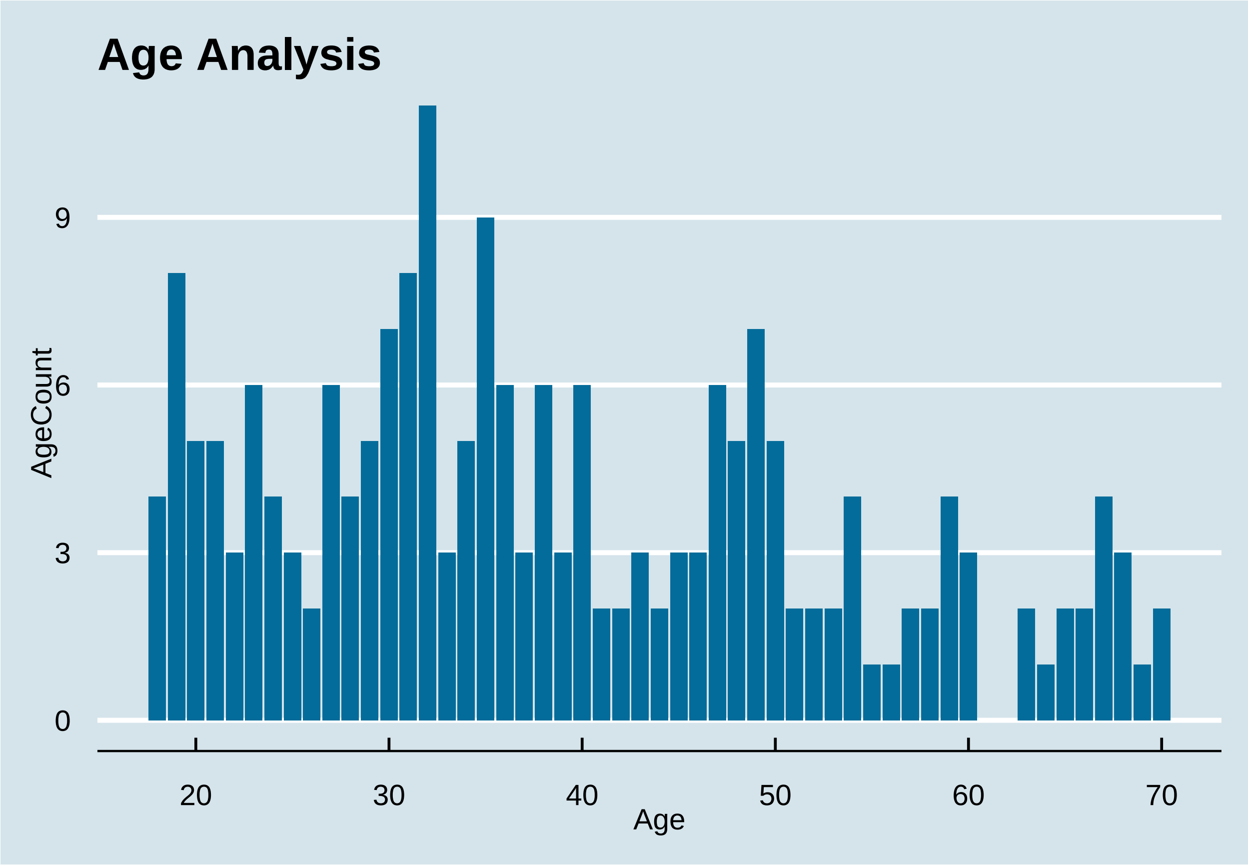
```
# Density of age by gender
pl1 <- custom %>%
  ggplot( aes(x=age, fill = gender)) +
  geom_density(alpha = 0.5) +
    xlab("Age")  +
```

```r
  ylab("Density") +
  ggtitle("Density of age by gender")+ scale_fill_manual(values=wes_palette(n=2, name="Darjeeling2"))
pl1
```



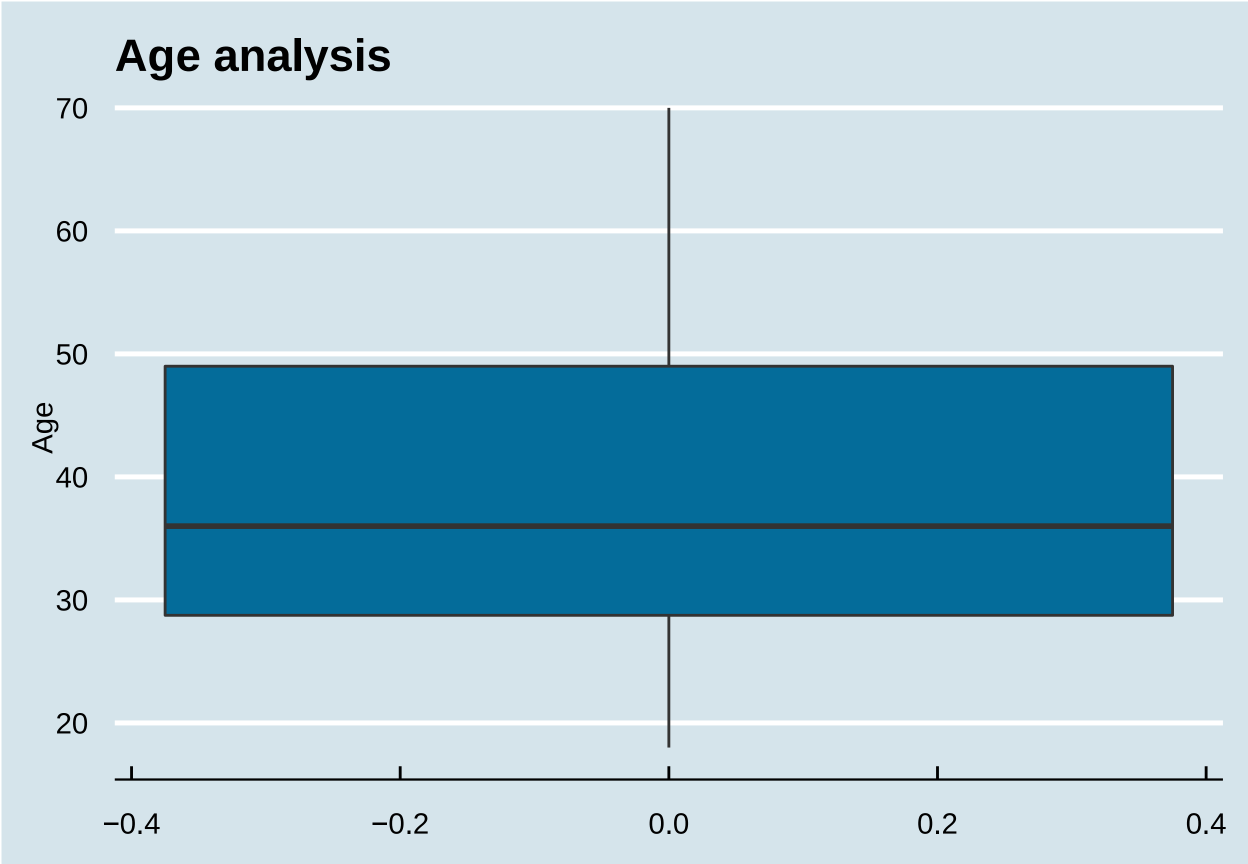

```r
#ggsave("pl1.png", plot = pl1, path = "") #Set path
```

```r
# Counting the frequency of the values of the age
age_analysis <- custom %>%
  group_by(age) %>%
  count() %>%
  ggplot()+
  geom_col(aes(age, n), fill = "#046C9A") +
  ggtitle("Age Analysis") +
  xlab("Age")  +
  ylab("AgeCount")
age_analysis
```



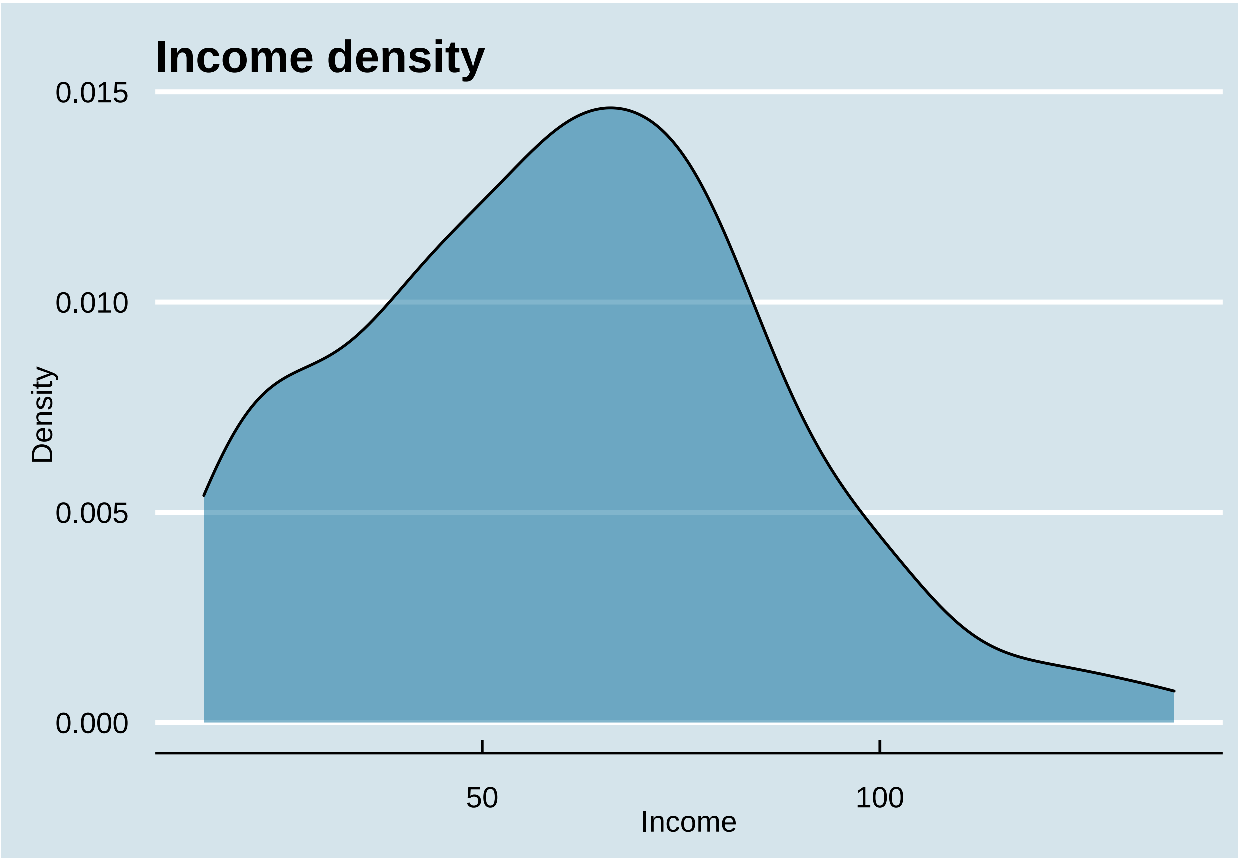

```
#ggsave("age_analysis.png", plot = age_analysis, path = "") #Set path
```

```
age_analysis_boxpl<-ggplot(custom, aes(y = age)) +
  geom_boxplot(fill='#046C9A')+
  ggtitle("Age analysis") +
  ylab("Age")
age_analysis_boxpl
```



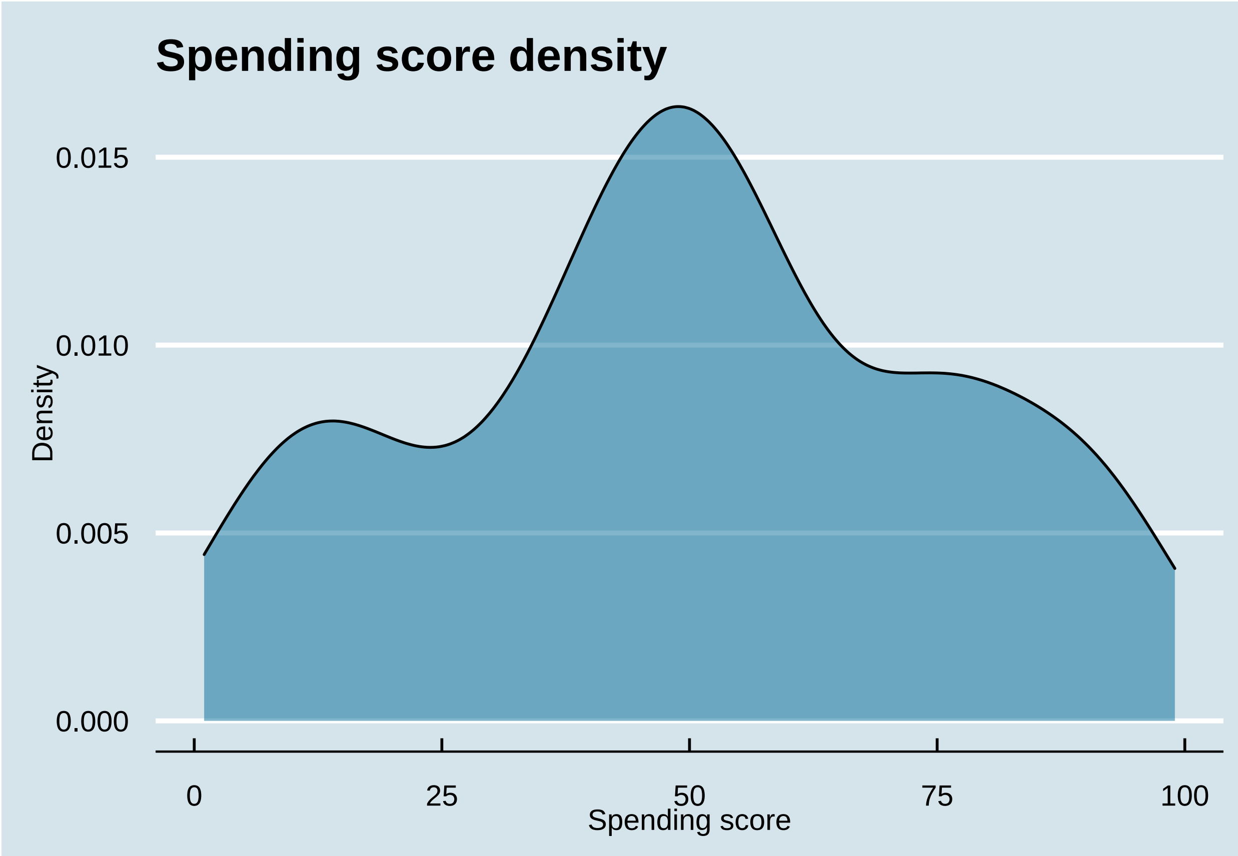

```
#ggsave("age_analysis_boxpl.png", plot = age_analysis_boxpl, path = "") #Set path
```

```
# Density income
income_dens<-custom %>%
  ggplot( aes(x=annual_income)) +
  geom_density(alpha = 0.5, fill = "#046C9A") +
  xlab("Income")  +
  ylab("Density") +
  ggtitle("Income density")
income_dens
```




```
#ggsave("income_dens.png", plot = income_dens, path = "") #Set path
```

```
# Spending score
spending_dens <- custom %>%
  ggplot( aes(x=spending_score)) +
  geom_density(alpha = 0.5, fill = "#046C9A") +
    xlab("Spending score")  +
  ylab("Density") +
  ggtitle("Spending score density")
spending_dens
```



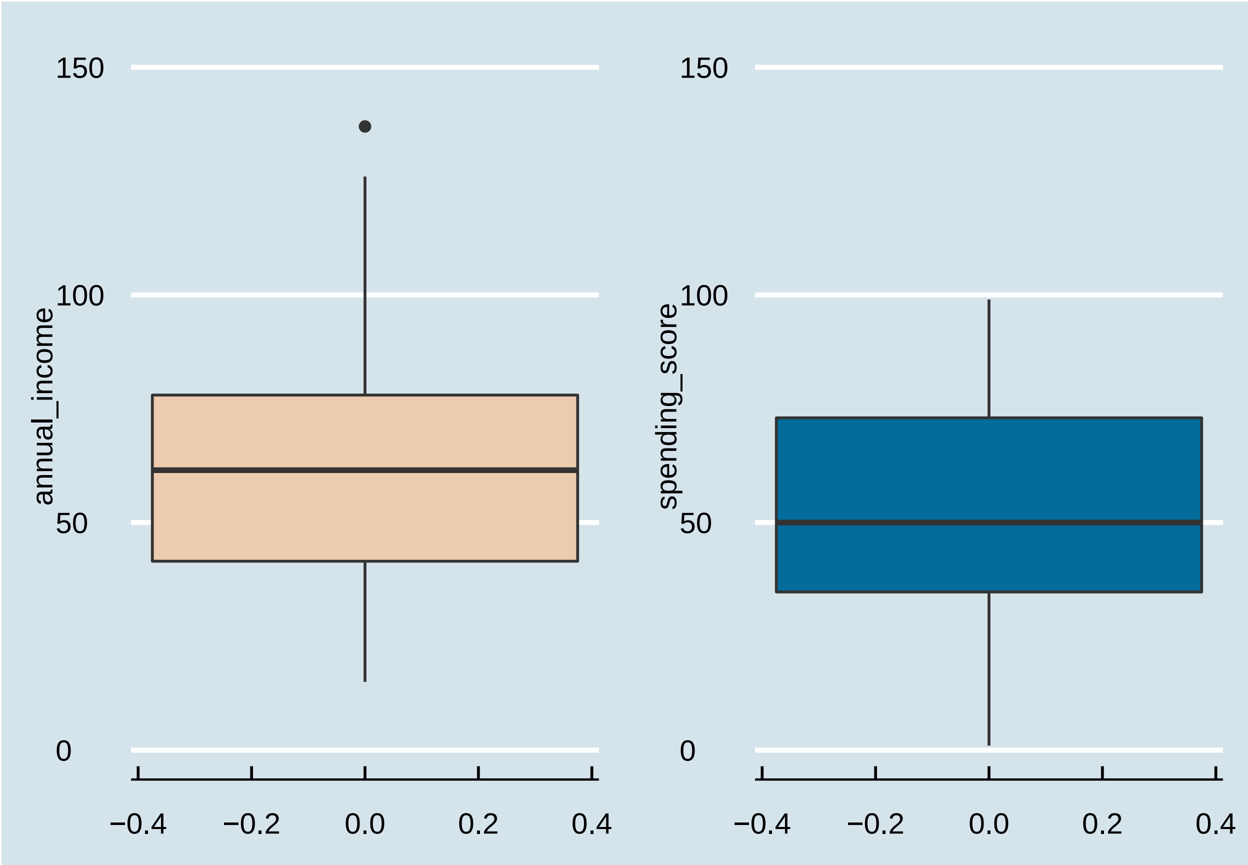

```r
#ggsave("spending_dens.png", plot = spending_dens, path = "") #Set path
```

```r
#Annual income and spending score bo
library(gridExtra)
```

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

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

```r
p1 <- ggplot(custom, aes(y = annual_income)) + geom_boxplot(fill='#ECCBAE') + ylim(c(1,150))
p2 <- ggplot(custom, aes(y = spending_score)) + geom_boxplot(fill='#046C9A') + ylim(c(1,150))
p_1_2 <- grid.arrange(p1, p2, ncol = 2)
```




```
p_1_2
```

```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
```

```
p_1_2
```

```
## TableGrob (1 x 2) "arrange": 2 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
```

```
#ggsave("p_1_2.png", plot = p_1_2, path = "") #Set path
```

```
custom$gender <- as.factor(custom$gender)
custom$gender <- as.numeric(custom$gender)
lapply(custom, class)
```

```
## $customer_id
## [1] "numeric"
##
```

```
## $gender
## [1] "numeric"
##
## $age
## [1] "numeric"
##
## $annual_income
## [1] "numeric"
##
## $spending_score
## [1] "numeric"
```

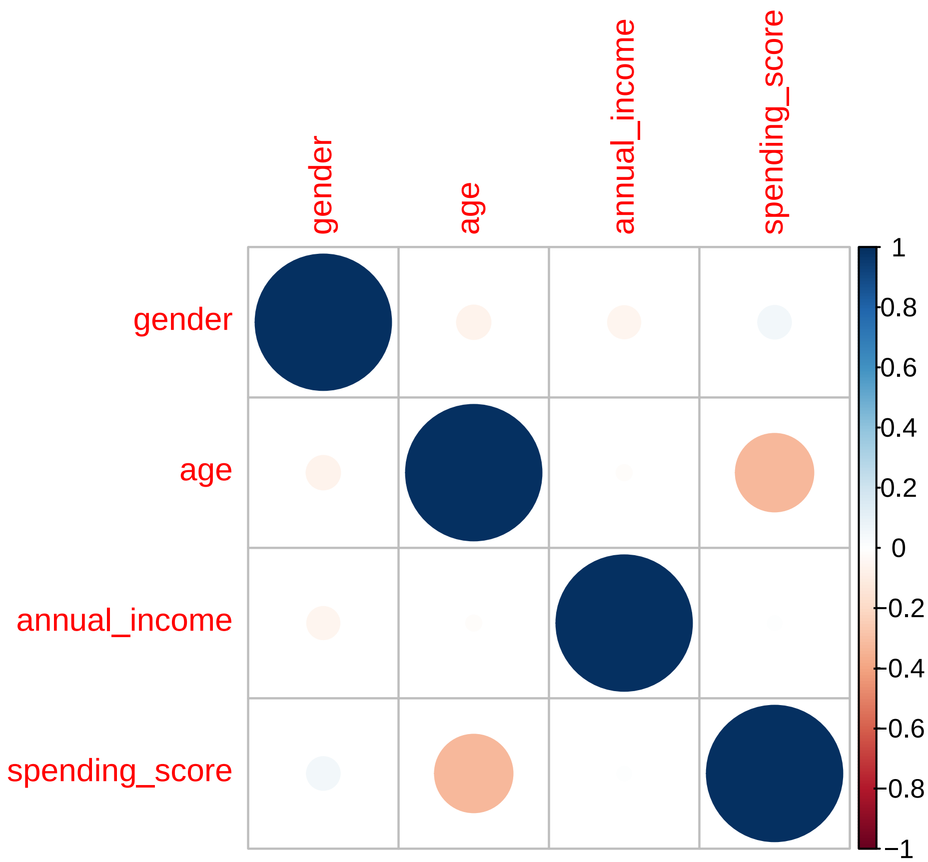
```
custom$gender[custom$gender == 2 ] <- 0 # Male = 0 Female = 1
head(custom)
```

```
## # A tibble: 6 x 5
##   customer_id gender   age annual_income spending_score
##         <dbl>  <dbl> <dbl>         <dbl>          <dbl>
## 1           1      0    19            15             39
## 2           2      0    21            15             81
## 3           3      1    20            16              6
## 4           4      1    23            16             77
## 5           5      1    31            17             40
## 6           6      1    22            17             76
```

```
cor_custom<- cor(custom[,-1])
cor_custom
```

```
##                     gender         age annual_income spending_score
## gender          1.00000000 -0.06086739  -0.056409810    0.058108739
## age            -0.06086739  1.00000000  -0.012398043   -0.327226846
## annual_income  -0.05640981 -0.01239804   1.000000000    0.009902848
## spending_score  0.05810874 -0.32722685   0.009902848    1.000000000
```

```
corrplot(cor_custom)
```

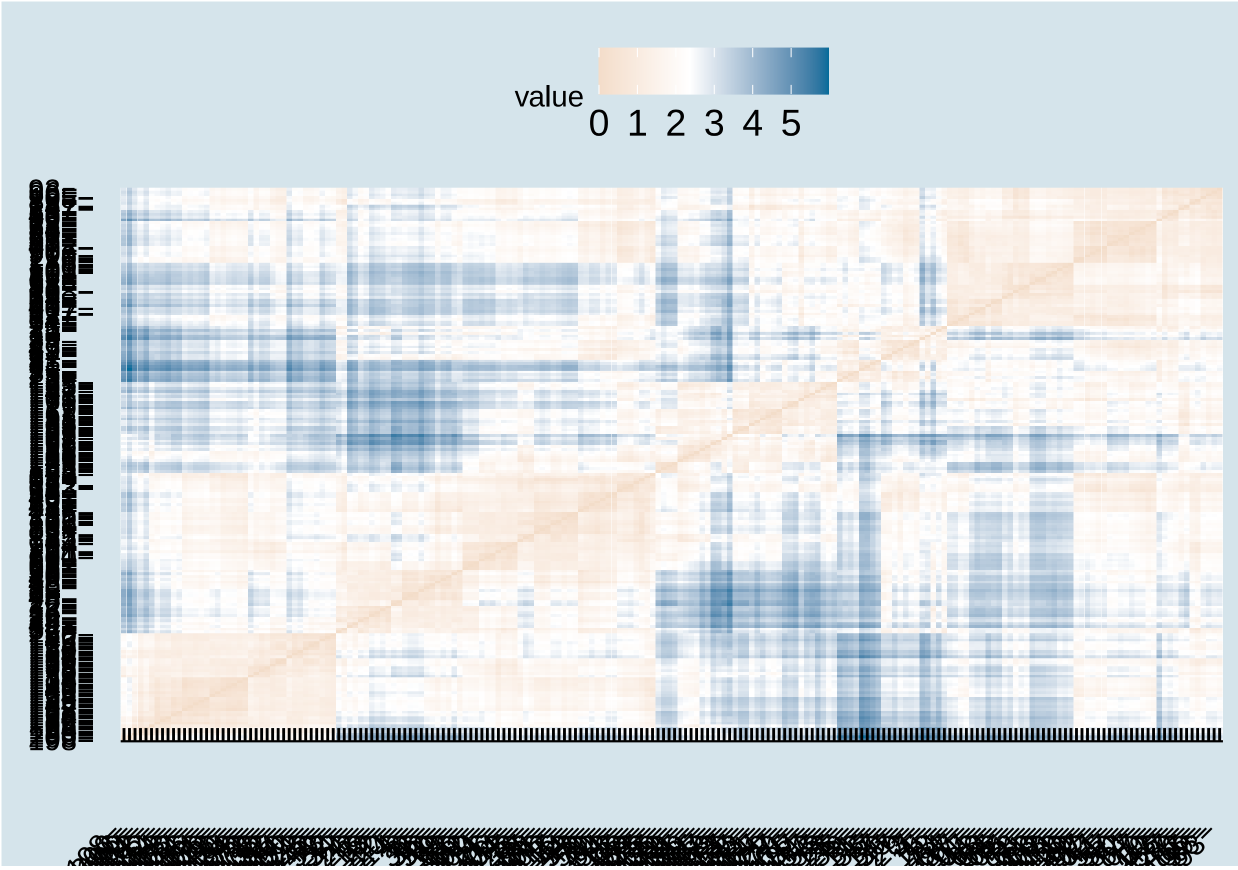



```
custom_norm <- lapply(custom[c(3,4,5)], function(x) c(scale(x)))
custom_norm <- as.data.frame(custom_norm)
custom_norm[ , 'gender'] <- custom$gender
custom_norm <- custom_norm %>%
  select(gender, everything())
head(as_tibble(custom_norm))
```

```
## # A tibble: 6 x 4
##   gender    age annual_income spending_score
##    <dbl>  <dbl>         <dbl>          <dbl>
## 1      0 -1.42          -1.73         -0.434
## 2      0 -1.28          -1.73          1.19
## 3      1 -1.35          -1.70         -1.71
## 4      1 -1.13          -1.70          1.04
## 5      1 -0.562         -1.66         -0.395
## 6      1 -1.21          -1.66          0.999
```

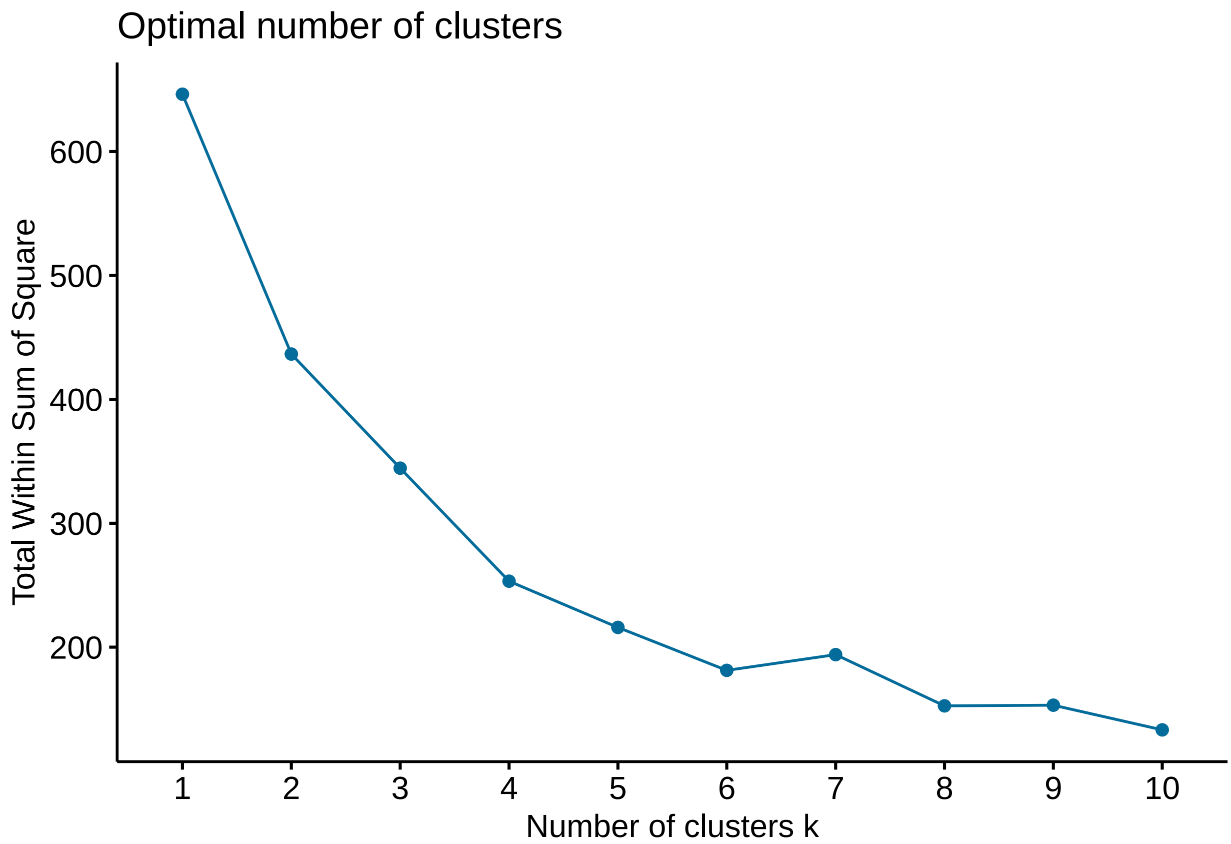
```
distance <- get_dist(custom_norm)
```

```
distance <- fviz_dist(distance, gradient = list(low = "#ECCBAE", mid = "white", high = "#046C9A"))
distance
```




```
#ggsave("distance.png", plot = distance, path = "") #Set path
```

```
# Elbow Method
set.seed(86)
elb_pl <- fviz_nbclust(custom_norm, kmeans, method = "wss", linecolor = "#046C9A")
elb_pl
```




```
#ggsave("elb_pl.png", plot = elb_pl, path = "") #Set path
```

```
# K = 4
kmodel_4 = kmeans(custom_norm, centers = 4)
kmodel_4
```

```
## K-means clustering with 4 clusters of sizes 38, 65, 57, 40
##
## Cluster means:
##      gender         age annual_income spending_score
## 1 0.5000000  0.03711223     0.9876366     -1.1857814
## 2 0.5692308  1.08344244    -0.4893373     -0.3961802
## 3 0.5964912 -0.96008279    -0.7827991      0.3910484
## 4 0.5500000 -0.42773261     0.9724070      1.2130414
##
## Clustering vector:
##   [1] 3 3 3 3 3 3 2 3 2 3 2 3 2 3 2 3 3 3 2 3 3 3 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2
##  [38] 3 2 3 2 3 2 3 2 3 3 3 2 3 3 2 2 2 2 2 3 2 2 3 2 2 2 3 2 2 3 3 2 2 2 2 2
##  [75] 2 3 2 2 3 2 2 3 2 2 3 2 2 3 3 2 2 3 2 2 3 3 2 3 2 3 3 2 2 3 2 3 2 2 2 2 2
## [112] 3 1 3 3 3 2 2 2 2 3 1 4 4 1 4 1 4 2 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4
## [149] 1 4 1 4 1 4 1 4 1 4 1 4 2 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1
## [186] 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4
##
## Within cluster sum of squares by cluster:
## [1] 53.51863 90.77126 75.15145 33.81544
```

```
##  (between_SS / total_SS =  60.8 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
table(kmodel_4$cluster)
```

```
##
##  1  2  3  4
## 38 65 57 40
```

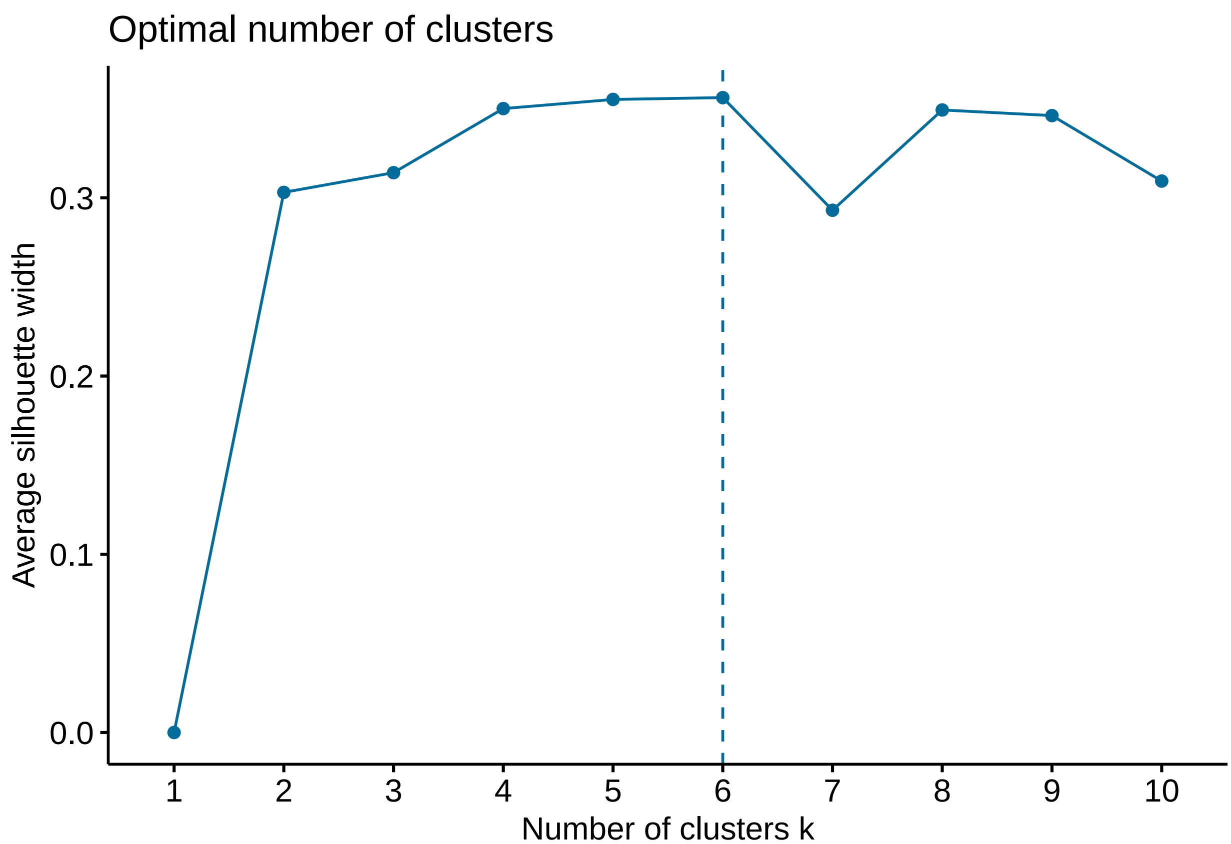
```
# K = 5
kmodel_5 = kmeans(custom_norm, centers = 5)
kmodel_5
```

```
## K-means clustering with 5 clusters of sizes 39, 47, 58, 34, 22
##
## Cluster means:
##      gender        age annual_income spending_score
## 1 0.5384615 -0.4408110     0.9891010      1.2364001
## 2 0.6595745 -0.7797657    -0.4020602     -0.2153735
## 3 0.5689655  1.1956271    -0.4598275     -0.3262196
## 4 0.4117647  0.1728617     1.0637844     -1.2947612
## 5 0.5909091 -0.9719569    -1.3262173      1.1293439
##
## Clustering vector:
##   [1] 2 5 2 5 2 5 2 5 3 5 3 5 3 5 2 5 2 5 3 5 2 5 3 5 3 5 3 5 2 5 3 5 3 5 3 5 3
##  [38] 5 2 5 3 5 3 2 3 5 3 2 2 2 3 2 2 3 3 3 3 3 2 3 3 2 3 3 3 2 3 3 2 2 3 3 3 3
##  [75] 3 2 3 2 2 3 3 2 3 3 2 3 3 2 2 3 3 2 3 2 2 2 3 2 3 2 2 3 3 2 3 2 3 3 3 3 3
## [112] 2 2 2 2 2 3 3 3 3 2 2 2 1 2 1 4 1 4 1 4 1 2 1 4 1 4 1 4 1 4 1 2 1 4 1 4 1
## [149] 4 1 4 1 4 1 4 1 4 1 4 1 3 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1 4
## [186] 1 4 1 4 1 4 1 4 1 4 1 4 1 4 1
##
## Within cluster sum of squares by cluster:
## [1] 32.05498 53.46660 71.15581 45.75370 13.51000
##  (between_SS / total_SS =  66.6 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
table(kmodel_5$cluster)
```

```
##
##  1  2  3  4  5
## 39 47 58 34 22
```

```r
# Average Silouette Method
avg_sil <- fviz_nbclust(custom_norm, kmeans, method = "silhouette", k.max = 10, linecolor = "#046C9A")
avg_sil
```




```r
#ggsave("avg_sil.png", plot = avg_sil, path = "") #Set path
```

```r
# K = 6
kmodel_6 = kmeans(custom_norm, centers = 6)
kmodel_6
```

```
## K-means clustering with 6 clusters of sizes 24, 58, 22, 10, 39, 47
##
## Cluster means:
##      gender         age annual_income spending_score
## 1 0.2500000  0.21058523     0.7449156     -1.3985957
## 2 0.5689655  1.19562713    -0.4598275     -0.3262196
## 3 0.5909091 -0.97195688    -1.3262173      1.1293439
## 4 0.8000000  0.08232511     1.8290695     -1.0455584
## 5 0.5384615 -0.44081102     0.9891010      1.2364001
## 6 0.6595745 -0.77976573    -0.4020602     -0.2153735
##
## Clustering vector:
##   [1] 6 3 6 3 6 3 6 3 2 3 2 3 2 3 6 3 6 3 2 3 6 3 2 3 2 3 2 3 6 3 2 3 2 3 2 3 2
##  [38] 3 6 3 2 3 2 6 2 3 2 6 6 6 2 6 6 2 2 2 2 2 6 2 2 6 2 2 2 6 2 2 6 6 2 2 2 2
```

```
##  [75] 2 6 2 6 6 2 2 6 2 2 6 2 2 6 6 2 2 6 2 6 6 6 2 6 2 6 6 2 2 6 2 6 2 2 2 2 2
## [112] 6 6 6 6 6 2 2 2 2 6 6 6 5 6 5 1 5 1 5 1 5 6 5 1 5 1 5 1 5 1 5 6 5 1 5 1 5
## [149] 1 5 1 5 1 5 1 5 1 5 1 5 2 5 1 5 1 5 1 5 4 5 1 5 1 5 1 5 1 5 1 5 4 5 1 5 4
## [186] 5 4 5 4 5 4 5 4 5 4 5 4 5 4 5
##
## Within cluster sum of squares by cluster:
## [1] 26.365728 71.155809 13.510004  7.959912 32.054981 53.466595
##  (between_SS / total_SS =  68.4 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

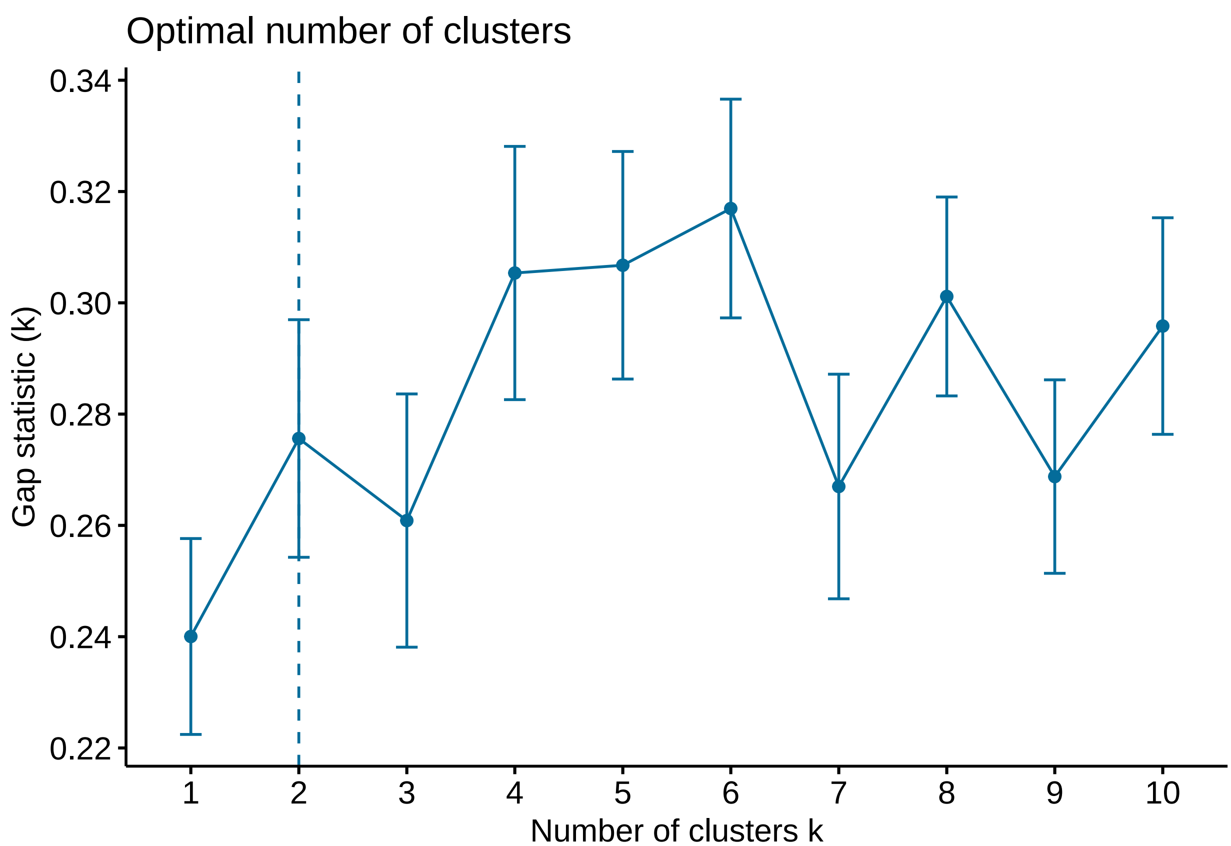
```
table(kmodel_6$cluster)
```

```
##
##  1  2  3  4  5  6
## 24 58 22 10 39 47
```

```
library(cluster)
set.seed(86)
gap_stat <- clusGap(custom_norm, FUN = kmeans, K.max = 10)
print(gap_stat, method = "firstmax")
```

```
## Clustering Gap statistic ["clusGap"] from call:
## clusGap(x = custom_norm, FUNcluster = kmeans, K.max = 10)
## B=100 simulated reference sets, k = 1..10; spaceH0="scaledPCA"
##  --> Number of clusters (method 'firstmax'): 2
##           logW   E.logW       gap     SE.sim
##  [1,] 4.774565 5.014596 0.2400310 0.01760184
##  [2,] 4.564499 4.840114 0.2756152 0.02134867
##  [3,] 4.441858 4.702724 0.2608659 0.02275679
##  [4,] 4.289733 4.595089 0.3053559 0.02275647
##  [5,] 4.203516 4.510262 0.3067463 0.02044863
##  [6,] 4.116303 4.433249 0.3169463 0.01965395
##  [7,] 4.106016 4.373006 0.2669894 0.02018157
##  [8,] 4.018735 4.319877 0.3011418 0.01787485
##  [9,] 4.008020 4.276791 0.2687713 0.01738714
## [10,] 3.939734 4.235560 0.2958267 0.01946202
```

```
opt_clus<-fviz_gap_stat(gap_stat, linecolor = "#046C9A")
opt_clus
```




```r
#ggsave("opt_clus.png", plot = opt_clus, path = "") #Set path
```

```r
# K = 2
kmodel_2 = kmeans(custom_norm, centers = 2)
kmodel_2
```

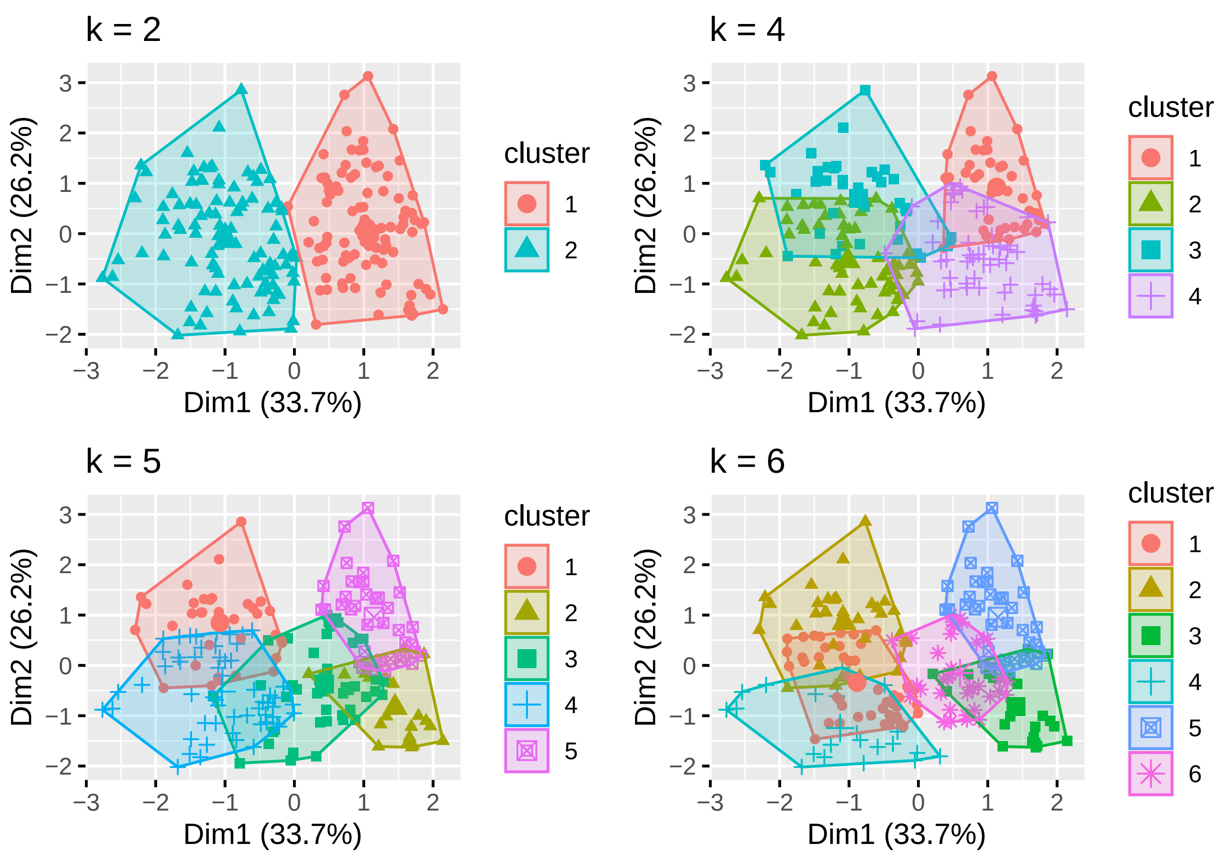
```
## K-means clustering with 2 clusters of sizes 97, 103
##
## Cluster means:
##      gender        age annual_income spending_score
## 1 0.5876289 -0.7508891   0.002621995      0.7407935
## 2 0.5339806  0.7071480  -0.002469258     -0.6976405
##
## Clustering vector:
##   [1] 1 1 2 1 1 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2
##  [38] 1 2 1 2 1 2 1 2 1 2 1 1 1 2 1 1 2 2 2 2 2 1 2 2 1 2 2 2 1 2 2 1 1 2 2 2 2
##  [75] 2 1 2 2 1 2 2 1 2 2 1 2 2 1 1 2 2 1 2 2 1 1 2 1 2 1 1 2 2 1 2 1 2 2 2 2 2
## [112] 1 2 1 1 1 2 2 2 2 1 2 1 1 1 1 1 2 1 2 1 2 1 1 1 1 2 1 2 1 2 1 2 1 1 1 2 1 2 1
## [149] 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1 2
## [186] 1 2 1 2 1 2 1 2 1 2 1 2 1 2 1
##
## Within cluster sum of squares by cluster:
## [1] 193.1955 243.3800
##  (between_SS / total_SS =  32.4 %)
##
```

```
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
table(kmodel_2$cluster)
```

```
##
##   1   2
##  97 103
```

```
k2 <- kmeans(custom_norm, centers = 2, nstart = 25)
k4 <- kmeans(custom_norm, centers = 4, nstart = 25)
k5 <- kmeans(custom_norm, centers = 5, nstart = 25)
k6 <- kmeans(custom_norm, centers = 6, nstart = 25)
p1 <- fviz_cluster(k2, geom = "point", data = custom_norm)+
  ggtitle("k = 2")
p3 <- fviz_cluster(k4, geom = "point", data = custom_norm)+
  ggtitle("k = 4")
p4 <- fviz_cluster(k5, geom = "point", data = custom_norm)+
  ggtitle("k = 5")
p5 <- fviz_cluster(k6, geom = "point", data = custom_norm)+
  ggtitle("k = 6")
library(gridExtra)
kmn_division <- grid.arrange(p1,p3,p4,p5, nrow = 2)
```



```r
kmn_division
```

```
## TableGrob (2 x 2) "arrange": 4 grobs
##   z     cells    name           grob
## 1 1 (1-1,1-1) arrange gtable[layout]
## 2 2 (1-1,2-2) arrange gtable[layout]
## 3 3 (2-2,1-1) arrange gtable[layout]
## 4 4 (2-2,2-2) arrange gtable[layout]
```

```r
#ggsave("kmn_division.png", plot = kmn_division, path = "") #Set path
```

```r
library(radiant.data)
```

```
## Loading required package: lubridate
```

```
##
## Attaching package: 'lubridate'
```

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

```
##
## Attaching package: 'radiant.data'
```

```
## The following objects are masked from 'package:lubridate':
##
##     month, wday
```

```
## The following object is masked from 'package:psych':
##
##     describe
```

```
## The following object is masked from 'package:magrittr':
##
##     set_attr
```

```
## The following object is masked from 'package:ggplot2':
##
##     diamonds
```

```
## The following object is masked from 'package:base':
##
##     date
```

```r
library(ggiraphExtra)
```

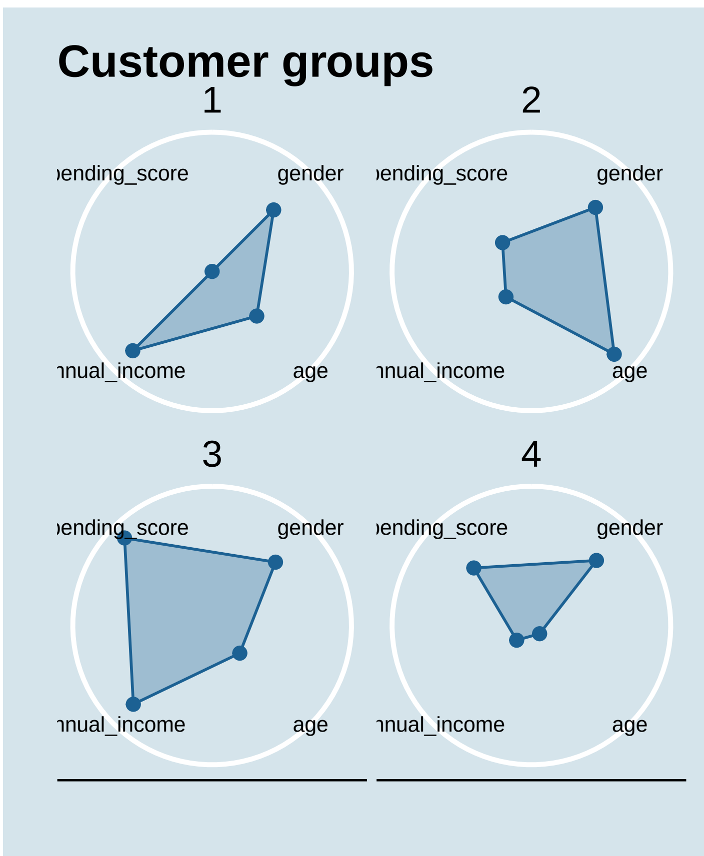
```
##
## Attaching package: 'ggiraphExtra'
```

```
## The following object is masked from 'package:ggthemes':
##
##     theme_clean
```

```r
final <- kmeans(custom_norm, 4, nstart = 30)
custom_df <- as.data.frame(custom_norm) %>% rownames_to_column()
cluster_pos <- as.data.frame(final$cluster) %>% rownames_to_column()
colnames(cluster_pos) <- c("rowname", "cluster")
custom_final <- inner_join(cluster_pos, custom_df)
```

```
## Joining, by = "rowname"
```

```r
radar_plt<-ggRadar(custom_final[-1], aes(group = cluster), rescale = FALSE, legend.position = "none", si
  facet_wrap(~cluster) +
  scale_y_discrete(breaks = NULL) +
  theme(axis.text.x = element_text(size = 8)) +
  scale_fill_manual(values = rep("#1c6193", nrow(custom_final))) +
  scale_color_manual(values = rep("#1c6193", nrow(custom_final))) +
  ggtitle("Customer groups")
radar_plt
```

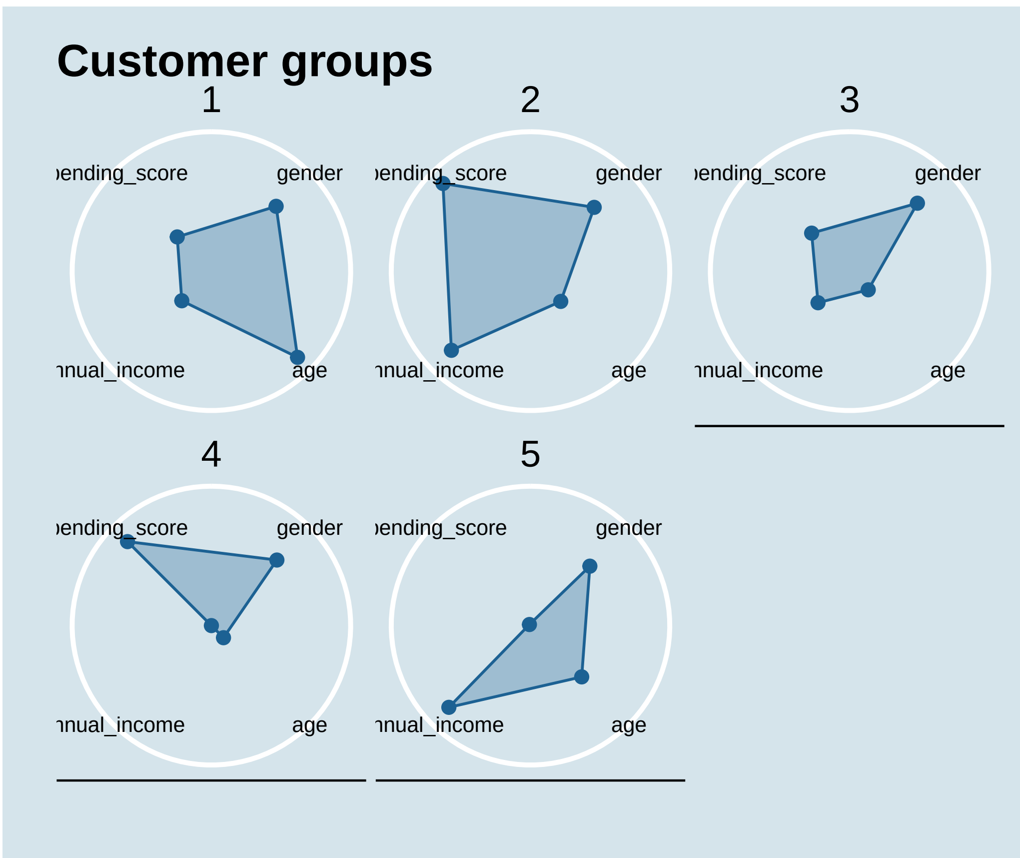



```r
#ggsave("radar_plt.png", plot = radar_plt, path = "") #Set path
```

```r
final_5 <- kmeans(custom_norm, 5, nstart = 30)
custom_df_5 <- as.data.frame(custom_norm) %>% rownames_to_column()
cluster_pos_5 <- as.data.frame(final_5$cluster) %>% rownames_to_column()
colnames(cluster_pos_5) <- c("rowname", "cluster")
custom_final_5 <- inner_join(cluster_pos_5, custom_df_5)
```

```
## Joining, by = "rowname"
```

```r
radar_plt_5<-ggRadar(custom_final_5[-1], aes(group = cluster), rescale = FALSE, legend.position = "none
  facet_wrap(~cluster) +
  scale_y_discrete(breaks = NULL) +
  theme(axis.text.x = element_text(size = 8)) +
  scale_fill_manual(values = rep("#1c6193", nrow(custom_final_5))) +
  scale_color_manual(values = rep("#1c6193", nrow(custom_final_5))) +
  ggtitle("Customer groups")
radar_plt_5
```

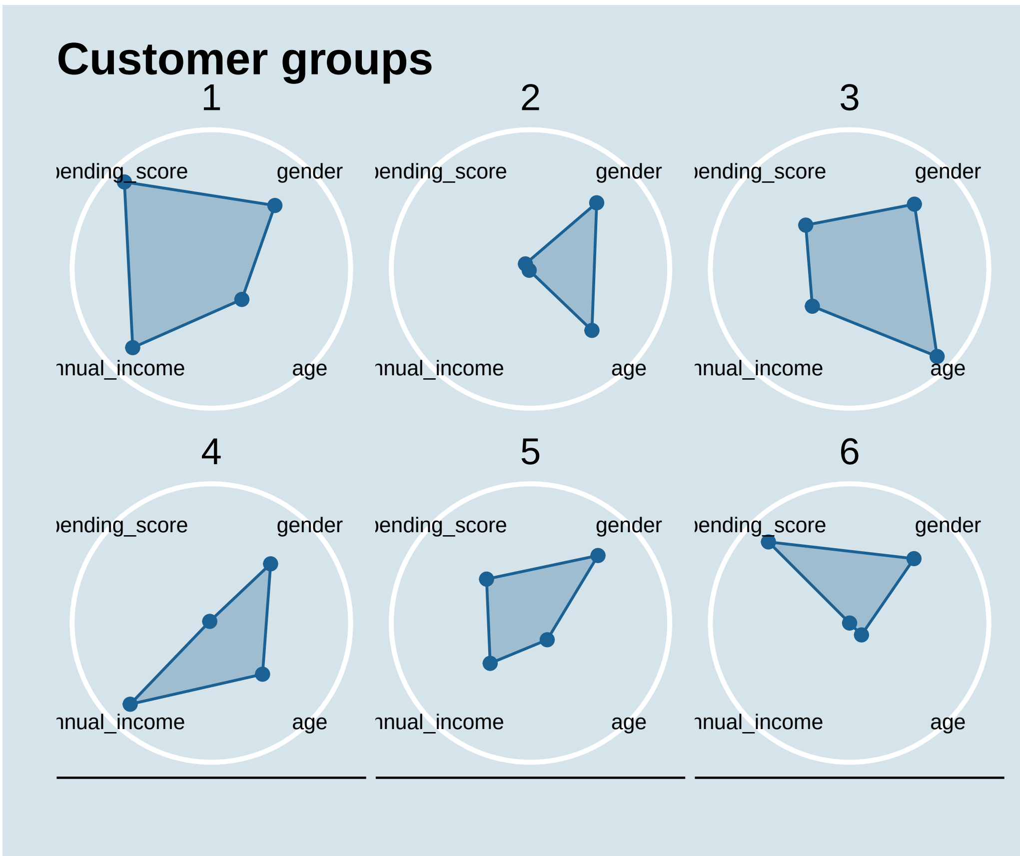



```r
#ggsave("radar_plt_5.png", plot = radar_plt_5, path = "") #Set path
```

```r
final_6 <- kmeans(custom_norm, 6, nstart = 30)
custom_df_6 <- as.data.frame(custom_norm) %>% rownames_to_column()
cluster_pos_6 <- as.data.frame(final_6$cluster) %>% rownames_to_column()
colnames(cluster_pos_6) <- c("rowname", "cluster")
custom_final_6 <- inner_join(cluster_pos_6, custom_df_6)
```

```
## Joining, by = "rowname"
```

```
radar_plt_6<-ggRadar(custom_final_6[-1], aes(group = cluster), rescale = FALSE, legend.position = "none
  facet_wrap(~cluster) +
  scale_y_discrete(breaks = NULL) +
  theme(axis.text.x = element_text(size = 8)) +
  scale_fill_manual(values = rep("#1c6193", nrow(custom_final_6))) +
  scale_color_manual(values = rep("#1c6193", nrow(custom_final_6))) +
  ggtitle("Customer groups")
radar_plt_6
```




```
#ggsave("radar_plt_6.png", plot = radar_plt_6, path = "") #Set path
```