Applications of Data Science and Statistics: Main Assignment

Jonathan Ludolph

03/01/2020

Contents

Initial Data Analysis Task

Question 1

```
library(tidyverse)
library(ggplot2)
library(Rmisc)
library(GGally)
library(chron)
library(cclust)
library(taRifx)
library(cluster)
library(factoextra)
library(dendextend)
library(kableExtra)
library(ggpubr)
# setting language for local time to english for future
# name of day retrieval
# (no need to run on english computer)
Sys.setlocale("LC_TIME","en_US.UTF-8")
```

```
# Loading Characteristics data set for initial analysis
Characteristics <- read.csv('Characteristics.csv',stringsAsFactors = F)
```

```r
head(Characteristics)
```

```
##   SUBSTATION_NUMBER         TRANSFORMER_TYPE TOTAL_CUSTOMERS
## 1            511016 Grd Mtd Dist. Substation             206
## 2            511017 Grd Mtd Dist. Substation               0
## 3            511028 Grd Mtd Dist. Substation             280
## 4            511029 Grd Mtd Dist. Substation             268
## 5            511030 Grd Mtd Dist. Substation             299
## 6            511032 Grd Mtd Dist. Substation             108
##   Transformer_RATING Percentage_IC LV_FEEDER_COUNT GRID_REFERENCE
## 1                750    0.70308406               5 ST187800775900
## 2                500    0.09264679               0 ST181000782000
## 3                500    0.24804607               5 ST188400769800
## 4                500    0.16029786               3 ST188200771500
## 5                500    0.28333084               5 ST187300772600
## 6                800    0.89802973               3 ST191800779200
```

```r
# summarising data frame to look for missing values
summary(is.na(Characteristics)) # No missing values
```

```
##  SUBSTATION_NUMBER TRANSFORMER_TYPE TOTAL_CUSTOMERS Transformer_RATING
##  Mode :logical     Mode :logical    Mode :logical   Mode :logical
##  FALSE:948         FALSE:948        FALSE:948       FALSE:948
##  Percentage_IC   LV_FEEDER_COUNT GRID_REFERENCE
##  Mode :logical   Mode :logical   Mode :logical
##  FALSE:948       FALSE:948       FALSE:948
```

```r
# ordinary summary of data frame
summary(Characteristics)
```

```
##  SUBSTATION_NUMBER TRANSFORMER_TYPE   TOTAL_CUSTOMERS Transformer_RATING
##  Min.   :511016    Length:948         Min.   :  0.0   Min.   :   0.0
##  1st Qu.:521516    Class :character   1st Qu.:  3.0   1st Qu.: 200.0
##  Median :532652    Mode  :character   Median : 67.5   Median : 315.0
##  Mean   :534344                       Mean   :104.3   Mean   : 389.1
##  3rd Qu.:552386                       3rd Qu.:179.2   3rd Qu.: 500.0
##  Max.   :564512                       Max.   :569.0   Max.   :1000.0
##  Percentage_IC     LV_FEEDER_COUNT  GRID_REFERENCE
##  Min.   :0.00000   Min.   : 0.000   Length:948
##  1st Qu.:0.01048   1st Qu.: 1.000   Class :character
##  Median :0.17849   Median : 3.000   Mode  :character
##  Mean   :0.37982   Mean   : 2.762
##  3rd Qu.:0.90271   3rd Qu.: 4.000
##  Max.   :1.00000   Max.   :16.000
```

```r
p1 <- ggplot(Characteristics,aes(x=Percentage_IC))+
  geom_histogram(colour='black',fill='blue',bins=20)+
  labs(x='Percentage_IC',
       y='frequency') +
  ggtitle('Distribution of Percentage_IC')

length(levels(factor(Characteristics$Transformer_RATING))) # 17 levels
```

```
## [1] 17
```

```r
p2 <- ggplot(Characteristics,aes(x=Transformer_RATING))+
  geom_histogram(colour='black',fill='blue',bins = 17)+
```

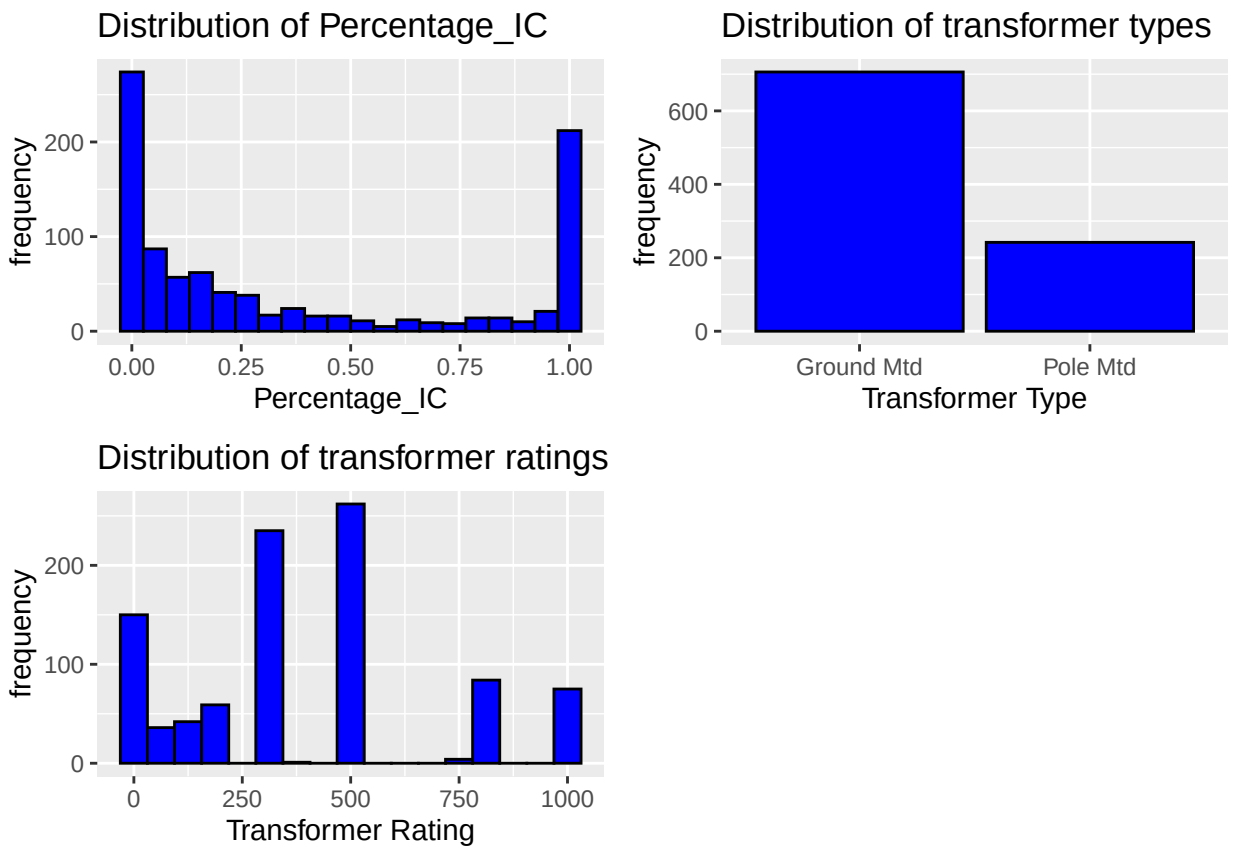
```
  # bins=17 because of 17 levels in Transformer_RATING
  labs(x='Transformer Rating',
       y='frequency')+
  ggtitle('Distribution of transformer ratings')

levels(factor(Characteristics$TRANSFORMER_TYPE)) # 2 levels with very long names
```

```
## [1] "Grd Mtd Dist. Substation"  "Pole Mtd Dist. Substation"
```

```
# let's rename them for easier use:
TT_fct <- factor(Characteristics$TRANSFORMER_TYPE)
levels(TT_fct) <- c('Ground Mtd','Pole Mtd')
Characteristics$TRANSFORMER_TYPE <- TT_fct

p3 <- ggplot(Characteristics,aes(x=as.factor(TRANSFORMER_TYPE)))+
  geom_bar(colour='black',fill='blue',bins = 2)+
  labs(x='Transformer Type',
       y='frequency')+
  ggtitle('Distribution of transformer types')
```

```
## Warning: Ignoring unknown parameters: bins
```

```
multiplot(p1,p2,p3,cols=2,
          title='Distributions for variables of Substations characteristics ')
```

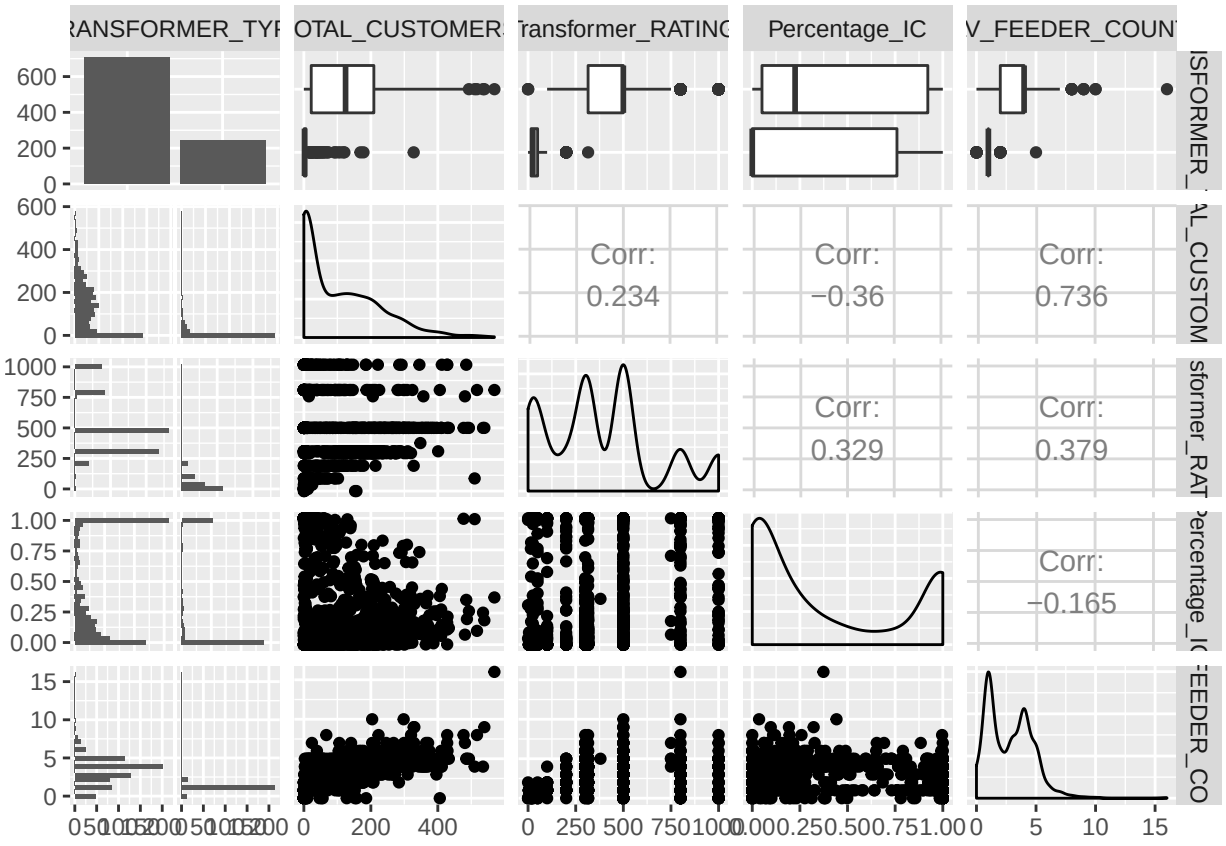



```
## [1] "Distributions for variables of Substations characteristics "
```

Question 2

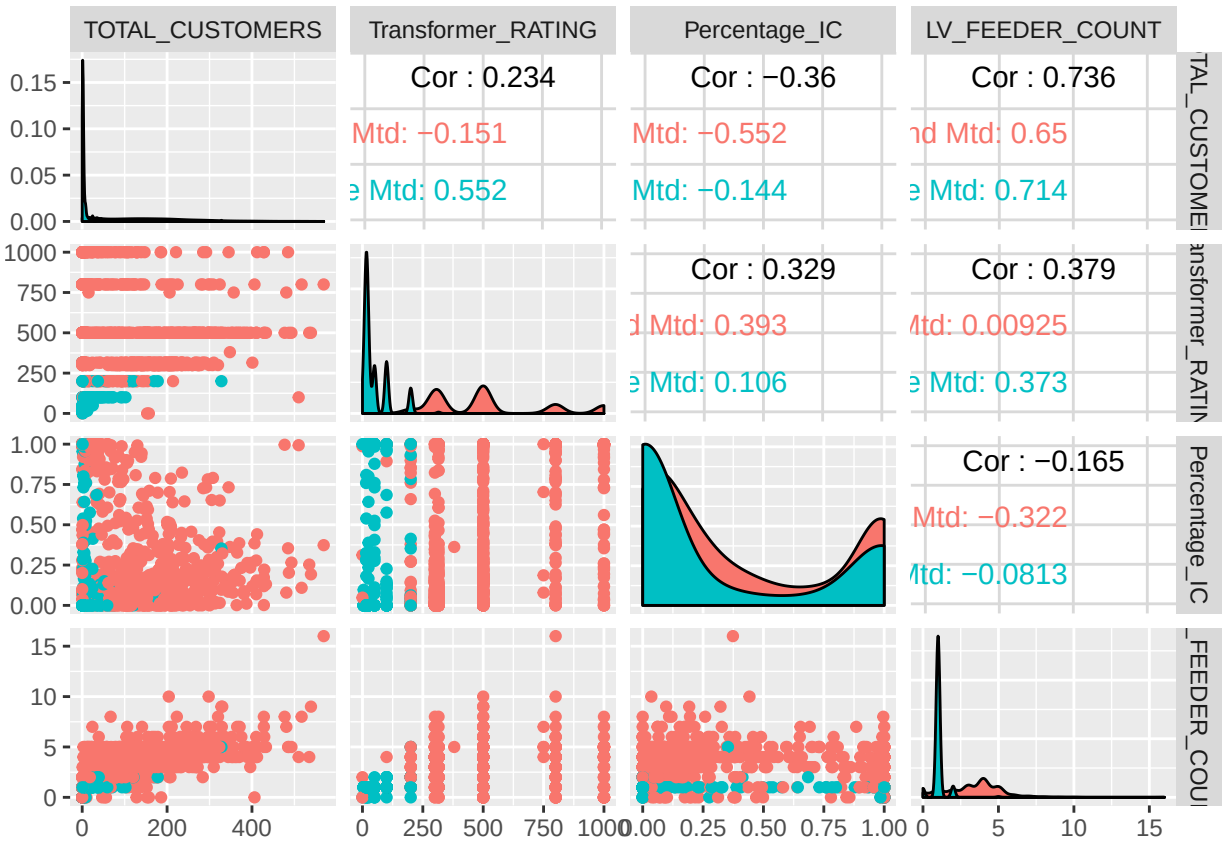
```
# pair plot for substation characteristics
ggpairs(Characteristics,columns = 2:6)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



On this pair-plot, we can see that the variables LV Feeder Count and Total customers are quite highly correlated. This could be expected since more feeders are required if there is a higher number of power users. Moreover, there are a few differences regarding Transformer types: the distribution of the number of total customers is quite focused onto the lower end for pole mounted transformers, whereas for ground mounted transformers the number of total customers is more spread out. Similarly, the transformer ratings and their feeder counts are skewed towards the lower end for pole mounted transformers and more spread out for ground mounted transformers.
Furthermore, the percentage of industrial and commercial customers is distributed in a relatively same way: we can observe peaks towards both ends (towards 0% and towards 100%). On the following pair-plot the distinction between transformer types is depicted more clearly to complement these relationships:

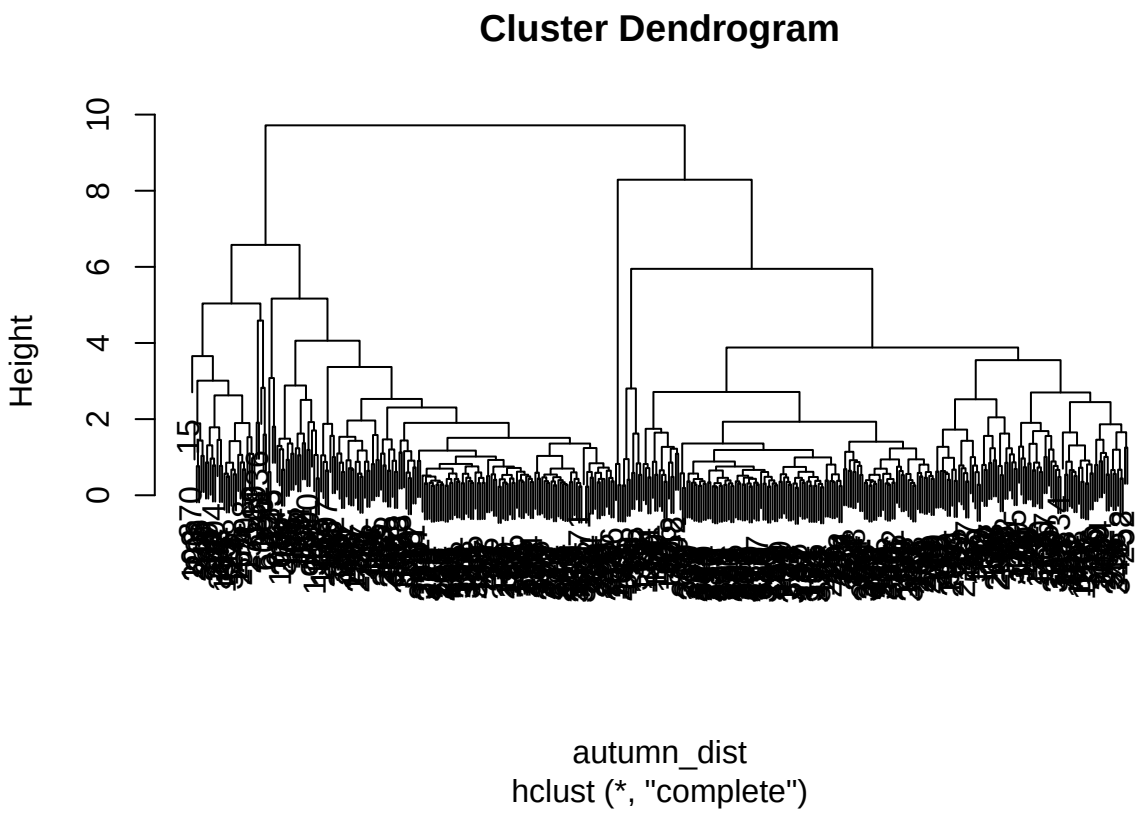
```
# pair plot for substation characteristics facetted by transformer type
# (orange is Ground Mtd and Blue is Pole Mtd)
ggpairs(Characteristics,columns = 3:6,ggplot2::aes(colour=TRANSFORMER_TYPE))
```



Initial Clustering Task

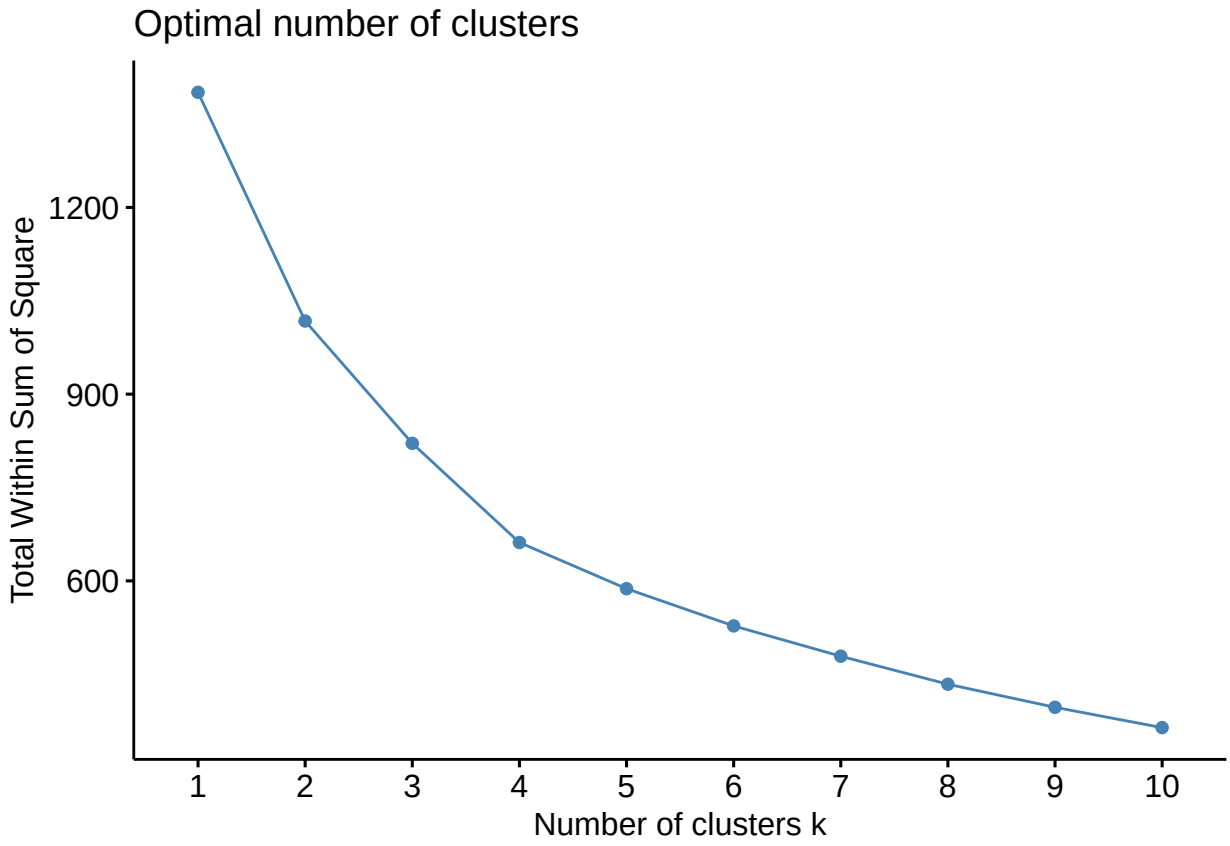
Question 3

```
# loading Autumn Data
load('Autumn_2012.rdata')
AutumnData <- Autumn_2012
AutumnReal <- AutumnData[,c(1,2,148:291)]
## Creating the distance Matrix

# Separating Scaled measurements
ScaledAutumn <- AutumnData[,1:146]
# Averages per interval per substation
DailyAverages <- ScaledAutumn[-2] %>%
  group_by(Station) %>%
  summarise_all(mean) %>%
  mutate(Station=as.factor(Station))
# distance matrix
autumn_dist <- dist(DailyAverages[-1])

##  hclust & dendogram plot

autumn_hclust <- hclust(autumn_dist,method = 'complete')
# let's visualize this dendogram
plot(hclust(autumn_dist))
```

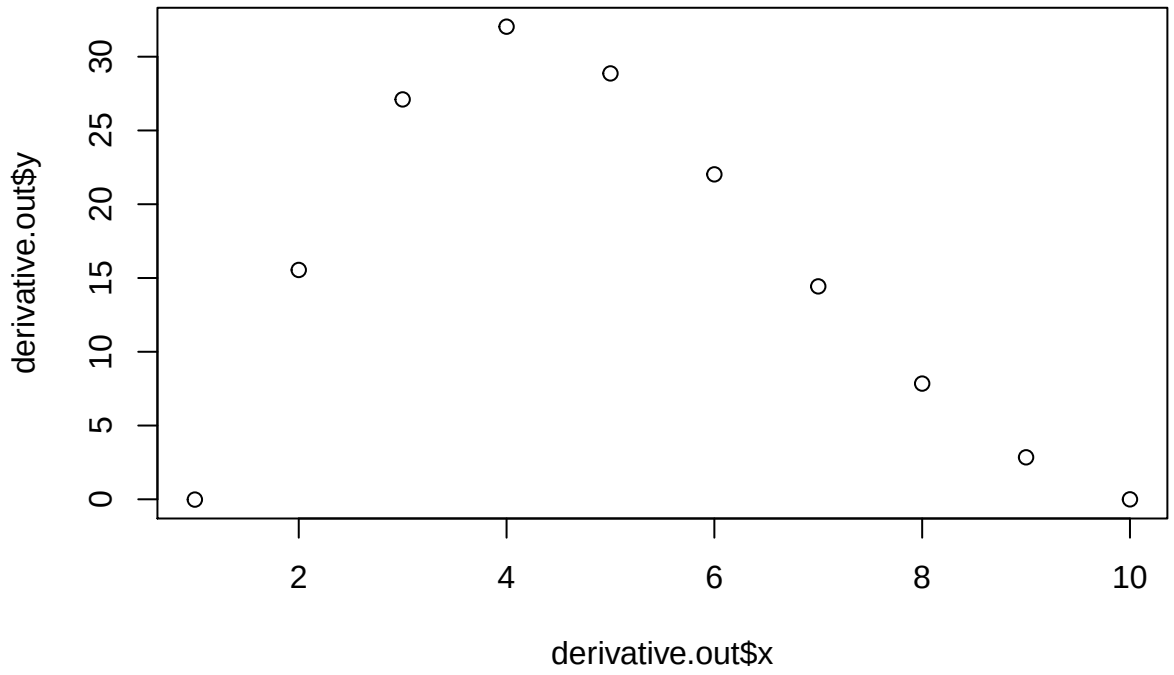


Question 4

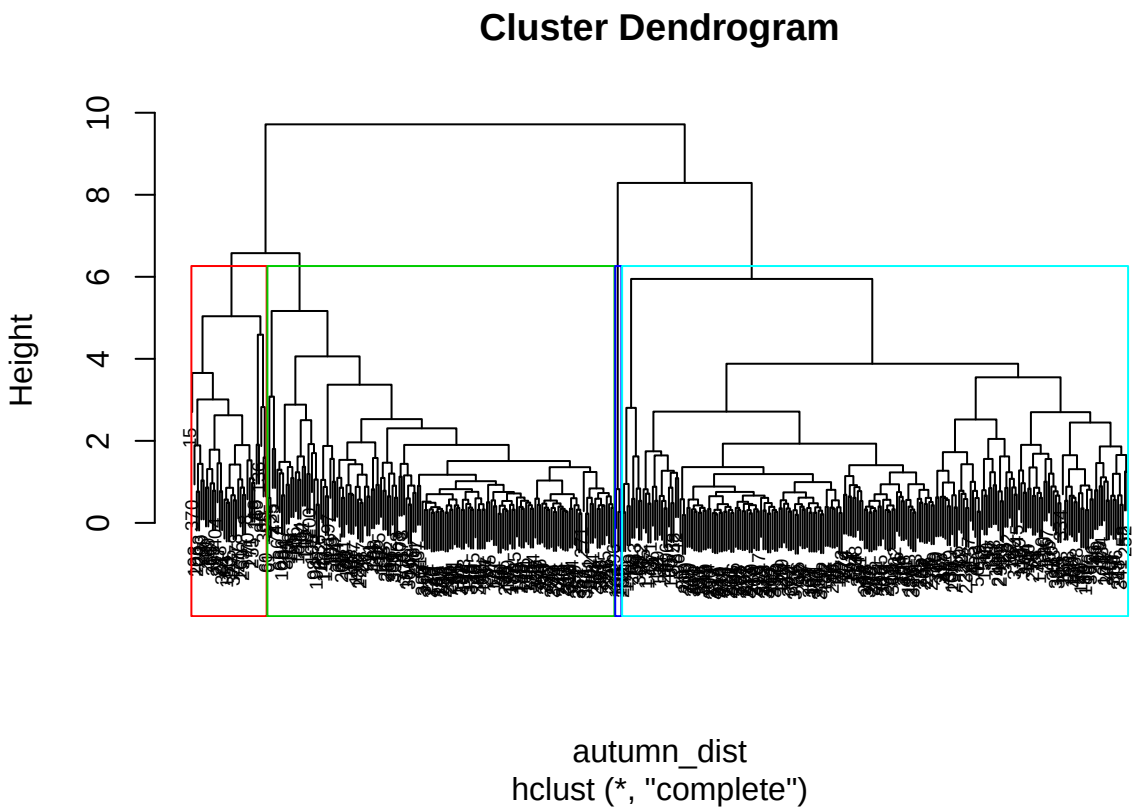
```
## Optimal number of cluster  testing with elbow method

# fviz_nbclust allows us to visualize within sum of squared plot for
# different number of clusters
twss <- fviz_nbclust(DailyAverages[-1],FUN=hcut,method='wss')
twss # elbow
```




```
# we retrieve the actual values in order to use them for the derivative
totwithins.df <- twss$data
totwithins.df$clusters <- as.numeric(totwithins.df$clusters)
withinss <- totwithins.df$y
# we then model a smooth spline to the data and predict derivatives from it for
# better visualization of point of maximum curvature
out.spl <- with(totwithins.df, smooth.spline(clusters, y=y, df = 3))
derivative.out <- with(totwithins.df, predict(out.spl, x = clusters, deriv = 2))

plot(derivative.out$x,derivative.out$y)
```




```
# Optimal number of clusters is clearly 4, let's visualize them

plot(autumn_hclust, cex = 0.6)
rect.hclust(autumn_hclust, k = 4, border = 2:5)
```




```
# Now let's assign cluster numbers to the substations
AutumnClusters <- cutree(autumn_hclust,k=4)
```

```r
# now let's label each Substation with it's respective cluster
DailyAverages$cluster <- as.factor(AutumnClusters)
```

Question 5

```r
# date and days modifications
ScaledAutumn$Julian_Date <- ScaledAutumn$Date
ScaledAutumn$Date <- dates(ScaledAutumn[,2], origin = c(month = 1,day = 1,year = 1970))
ScaledAutumn$Day <- weekdays(as.Date(ScaledAutumn$Date,'%m/%d/%y'))
# renaming the day levels for easier filtering in the future
day_fact <- factor(ScaledAutumn$Day)
levels(day_fact) <- c('Weekday','Weekday',
                      'Saturday','Sunday',
                      'Weekday','Weekday',
                      'Weekday')

ScaledAutumn$Day <- day_fact

# Separating days


###  Cluster 1 stations
stations1 <- DailyAverages %>%
  filter(cluster==1) %>%
  select(Station)
# filtering for cluster 1 stations for different days
autumn1 <- ScaledAutumn %>%
  filter(Station %in% stations1$Station)

## All days

Station1All <- autumn1 %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean of all days
s1meanAll <- autumn1 %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Week days

Station1Week <- autumn1 %>%
  filter(Day=='Weekday') %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
```

```
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean
s1meanWeek <- autumn1 %>%
  filter(Day=='Weekday') %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Saturdays

Station1Sat <- autumn1 %>%
  filter(Day=='Saturday') %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean
s1meanSat <- autumn1 %>%
  filter(Day=='Saturday') %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Sundays

Station1Sun <- autumn1 %>%
  filter(Day=='Sunday')%>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean

s1meanSun <- autumn1 %>%
  filter(Day=='Sunday') %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Plots
```

```
s1p <- ggplot(Station1All, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s1meanAll,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Scaled Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('All Days Cluster 1')

s1p1 <- ggplot(Station1Week, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s1meanWeek,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Scaled Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Week Days Cluster 1')

s1p2 <- ggplot(Station1Sat, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s1meanSat,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Scaled Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Saturdays Cluster 1')

s1p3 <- ggplot(Station1Sun, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s1meanSun,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Scaled Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Sundays Cluster 1')


### Cluster 2 stations
stations2 <- DailyAverages %>%
  filter(cluster==2) %>%
  select(Station)

# filtering for cluster 2 stations
autumn2 <- ScaledAutumn %>%
  filter(Station %in% stations2$Station)

## All days

Station2All <- autumn2 %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
```

```
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean
s2meanAll <- autumn2 %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Week days

Station2Week <- autumn2 %>%
  filter(Day=='Weekday')%>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean
s2meanWeek <- autumn2 %>%
  filter(Day=='Weekday')%>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Saturdays

Station2Sat <- autumn2 %>%
  filter(Day=='Saturday')%>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean
s2meanSat <- autumn2 %>%
  filter(Day=='Saturday')%>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Sundays

Station2Sun <- autumn2 %>%
  filter(Day=='Sunday')%>%
  select(-Date,-Day,-Julian_Date)%>%
```

```
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean
s2meanSun <- autumn2 %>%
  filter(Day=='Sunday')%>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Plots

s2p <- ggplot(Station2All, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s2meanAll,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Scaled Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('All Days Cluster 2')

s2p1 <- ggplot(Station2Week, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s2meanWeek,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Scaled Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Week Days Cluster 2')

s2p2 <- ggplot(Station2Sat, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s2meanSat,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Scaled Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Saturdays Cluster 2')

s2p3 <- ggplot(Station2Sun, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s2meanSun,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Scaled Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Sundays Cluster 2')

###  Cluster 3 stations
```

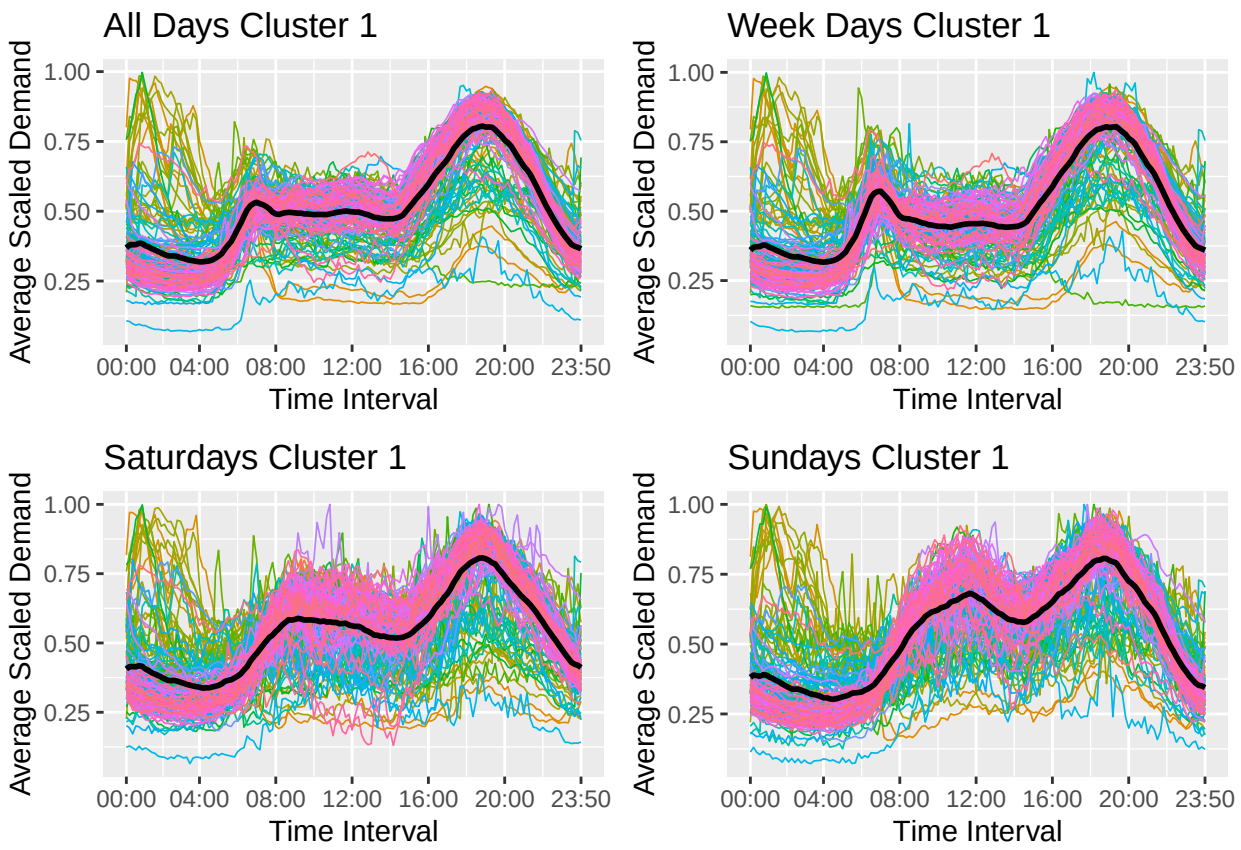
```
stations3 <- DailyAverages %>%
  filter(cluster==3) %>%
  select(Station)
# filtering for cluster 3 stations for different days
autumn3 <- ScaledAutumn %>%
  filter(Station %in% stations3$Station)

## All days

Station3All <- autumn3 %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean of all days
s3meanAll <- autumn3 %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Week days

Station3Week <- autumn3 %>%
  filter(Day=='Weekday') %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean
s3meanWeek <- autumn3 %>%
  filter(Day=='Weekday') %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Saturdays

Station3Sat <- autumn3 %>%
  filter(Day=='Saturday') %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
```

```
# mean
s3meanSat <- autumn3 %>%
  filter(Day=='Saturday') %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Sundays

Station3Sun <- autumn3 %>%
  filter(Day=='Sunday')%>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean

s3meanSun <- autumn3 %>%
  filter(Day=='Sunday') %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Plots

s3p <- ggplot(Station3All, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s3meanAll,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Scaled Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('All Days Cluster 3')

s3p1 <- ggplot(Station3Week, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s3meanWeek,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Scaled Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Week Days Cluster 3')

s3p2 <- ggplot(Station3Sat, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s3meanSat,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Scaled Demand') +
```

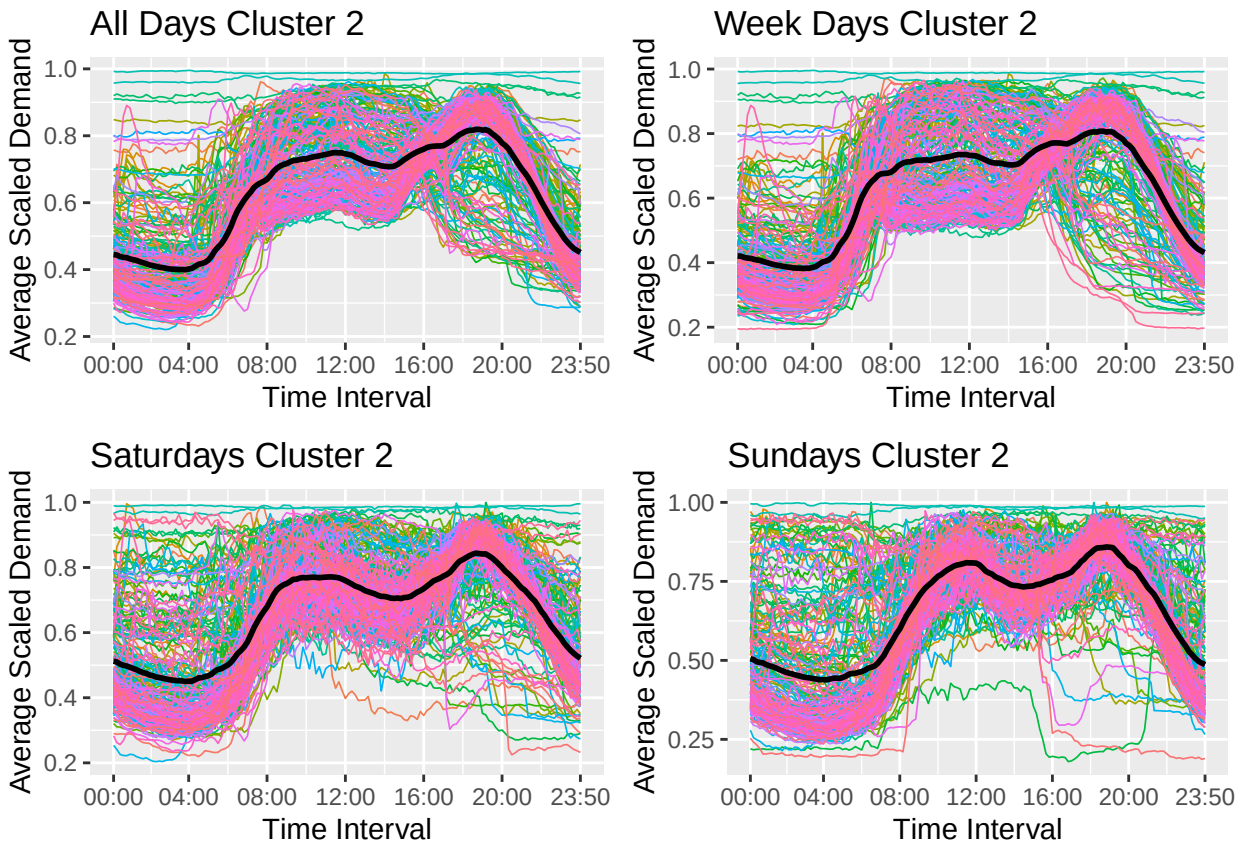
```
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Saturdays Cluster 3')

s3p3 <- ggplot(Station3Sun, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s3meanSun,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Scaled Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Sundays Cluster 3')

###  Cluster 4 stations
stations4 <- DailyAverages %>%
  filter(cluster==4) %>%
  select(Station)
# filtering for cluster 4 stations for different days
autumn4 <- ScaledAutumn %>%
  filter(Station %in% stations4$Station)

## All days

Station4All <- autumn4 %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean of all days
s4meanAll <- autumn4 %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Week days

Station4Week <- autumn4 %>%
  filter(Day=='Weekday') %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean
s4meanWeek <- autumn4 %>%
  filter(Day=='Weekday') %>%
  select(-Date,-Day,-Julian_Date)%>%
```

```
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Saturdays

Station4Sat <- autumn4 %>%
  filter(Day=='Saturday') %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean
s4meanSat <- autumn4 %>%
  filter(Day=='Saturday') %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Sundays

Station4Sun <- autumn4 %>%
  filter(Day=='Sunday')%>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean

s4meanSun <- autumn4 %>%
  filter(Day=='Sunday') %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Plots

s4p <- ggplot(Station4All, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s4meanAll,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Scaled Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
```

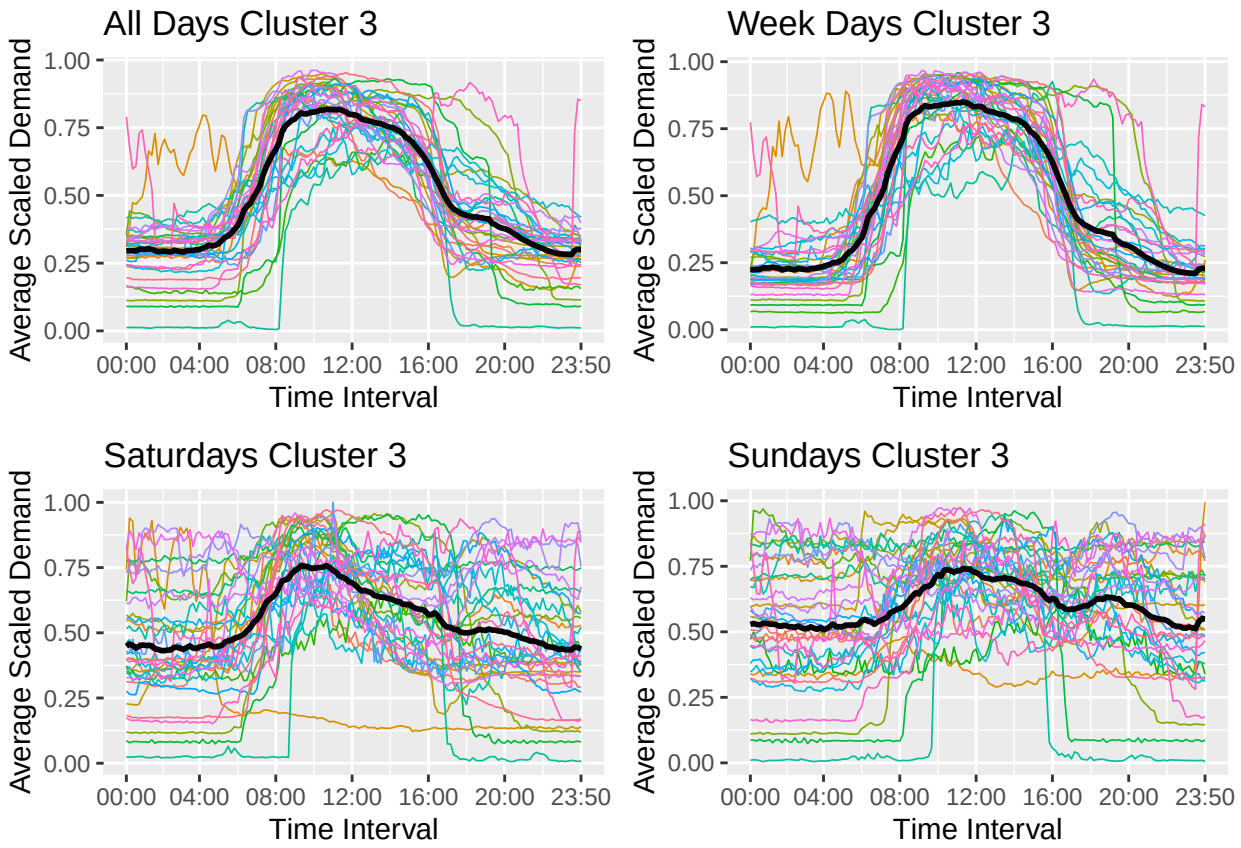
```
  ggtitle('All Days Cluster 4')

s4p1 <- ggplot(Station4Week, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s4meanWeek,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Scaled Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Week Days Cluster 4')

s4p2 <- ggplot(Station4Sat, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s4meanSat,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Scaled Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Saturdays Cluster 4')

s4p3 <- ggplot(Station4Sun, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s4meanSun,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Scaled Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Sundays Cluster 4')


#### For all daily averages plots, the black line represents the mean values ####
multiplot(s1p,s1p1,s1p2,s1p3,layout=matrix(c(1,2,3,4), nrow=2, byrow=TRUE))
```



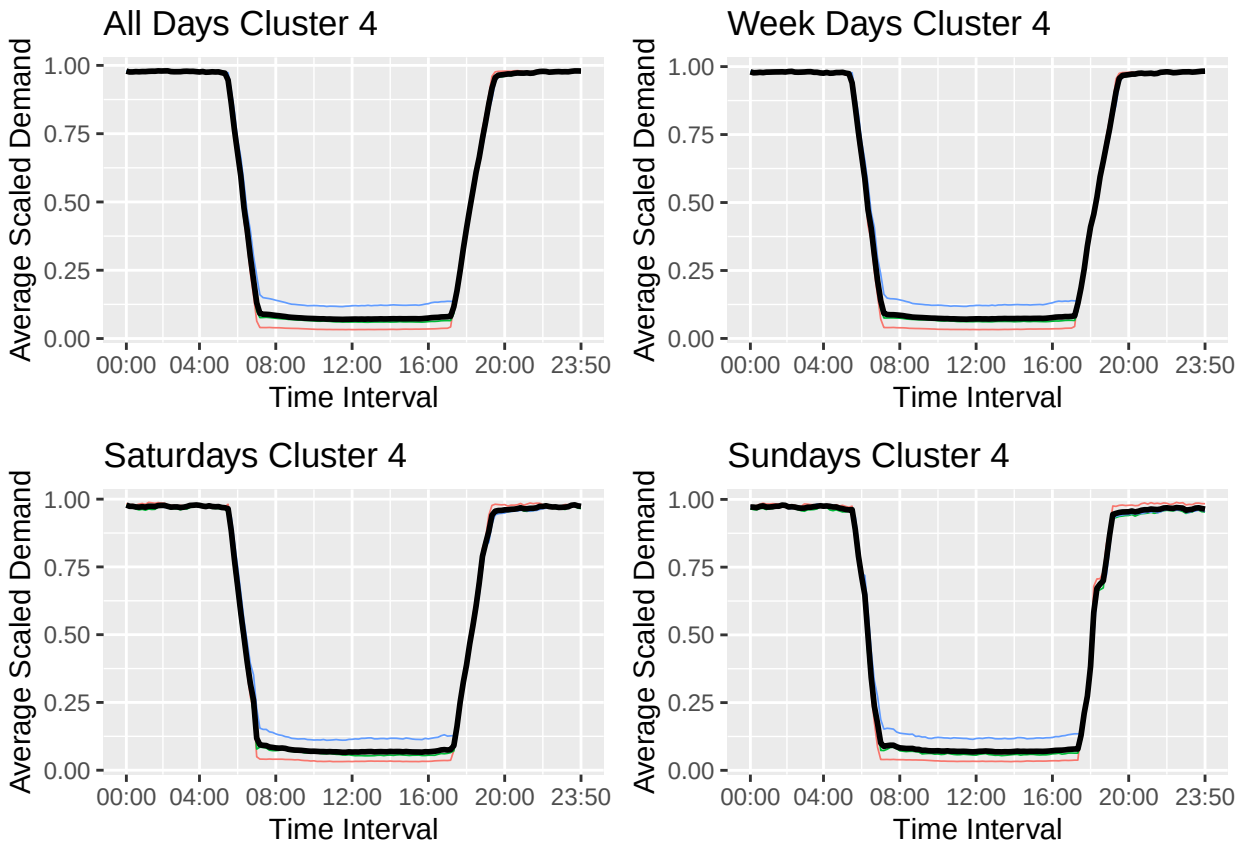

```
multiplot(s2p,s2p1,s2p2,s2p3,layout=matrix(c(1,2,3,4), nrow=2, byrow=TRUE))
```




```
multiplot(s3p,s3p1,s3p2,s3p3,layout=matrix(c(1,2,3,4), nrow=2, byrow=TRUE))
```




```
multiplot(s4p,s4p1,s4p2,s4p3,layout=matrix(c(1,2,3,4), nrow=2, byrow=TRUE))
```



Question 6

```
Characteristics1 <- Characteristics %>%
  filter(SUBSTATION_NUMBER %in% stations1$Station)

Characteristics2 <- Characteristics %>%
  filter(SUBSTATION_NUMBER %in% stations2$Station)

Characteristics3 <- Characteristics %>%
  filter(SUBSTATION_NUMBER %in% stations3$Station)

Characteristics4 <- Characteristics %>%
  filter(SUBSTATION_NUMBER %in% stations4$Station)



# Cluster 1
summary(Characteristics1)
```

```
##  SUBSTATION_NUMBER   TRANSFORMER_TYPE TOTAL_CUSTOMERS  Transformer_RATING
##  Min.   :511029    Ground Mtd:120     Min.   :  0.00   Min.   :  15.0
##  1st Qu.:513430    Pole Mtd  : 32     1st Qu.: 56.25   1st Qu.: 200.0
##  Median :532654                       Median :110.00   Median : 315.0
##  Mean   :534547                       Mean   :113.87   Mean   : 375.7
##  3rd Qu.:552173                       3rd Qu.:156.75   3rd Qu.: 500.0
##  Max.   :563737                       Max.   :301.00   Max.   :1000.0
```

```
##  Percentage_IC     LV_FEEDER_COUNT GRID_REFERENCE
##  Min.   :0.00000   Min.   :0.000   Length:152
##  1st Qu.:0.00000   1st Qu.:2.000   Class :character
##  Median :0.03889   Median :3.000   Mode  :character
##  Mean   :0.12604   Mean   :2.928
##  3rd Qu.:0.14073   3rd Qu.:4.000
##  Max.   :1.00000   Max.   :6.000
```

```
# Cluster 2
summary(Characteristics2)
```

```
##  SUBSTATION_NUMBER   TRANSFORMER_TYPE TOTAL_CUSTOMERS  Transformer_RATING
##  Min.   :511030    Ground Mtd:215     Min.   :  0.00   Min.   :   0.0
##  1st Qu.:513180    Pole Mtd  :  5     1st Qu.: 16.75   1st Qu.: 300.0
##  Median :532690                       Median :155.00   Median : 500.0
##  Mean   :533782                       Mean   :150.36   Mean   : 507.6
##  3rd Qu.:552601                       3rd Qu.:233.25   3rd Qu.: 500.0
##  Max.   :564285                       Max.   :485.00   Max.   :1000.0
##  Percentage_IC     LV_FEEDER_COUNT  GRID_REFERENCE
##  Min.   :0.00000   Min.   : 0.000   Length:220
##  1st Qu.:0.09198   1st Qu.: 2.000   Class :character
##  Median :0.31399   Median : 4.000   Mode  :character
##  Mean   :0.45064   Mean   : 3.495
##  3rd Qu.:0.92310   3rd Qu.: 5.000
##  Max.   :1.00000   Max.   :10.000
```

```
# Cluster 3
summary(Characteristics3)
```

```
##  SUBSTATION_NUMBER   TRANSFORMER_TYPE TOTAL_CUSTOMERS Transformer_RATING
##  Min.   :511150    Ground Mtd:28      Min.   : 0.00   Min.   :  50.0
##  1st Qu.:513050    Pole Mtd  : 5      1st Qu.: 2.00   1st Qu.: 315.0
##  Median :532229                       Median : 8.00   Median : 500.0
##  Mean   :532570                       Mean   :18.52   Mean   : 491.8
##  3rd Qu.:551996                       3rd Qu.:25.00   3rd Qu.: 500.0
##  Max.   :564224                       Max.   :94.00   Max.   :1000.0
##  Percentage_IC    LV_FEEDER_COUNT GRID_REFERENCE
##  Min.   :0.0000   Min.   :0.000   Length:33
##  1st Qu.:0.9210   1st Qu.:1.000   Class :character
##  Median :1.0000   Median :2.000   Mode  :character
##  Mean   :0.9067   Mean   :2.333
##  3rd Qu.:1.0000   3rd Qu.:3.000
##  Max.   :1.0000   Max.   :8.000
```

```
# Cluster 4
summary(Characteristics4)
```

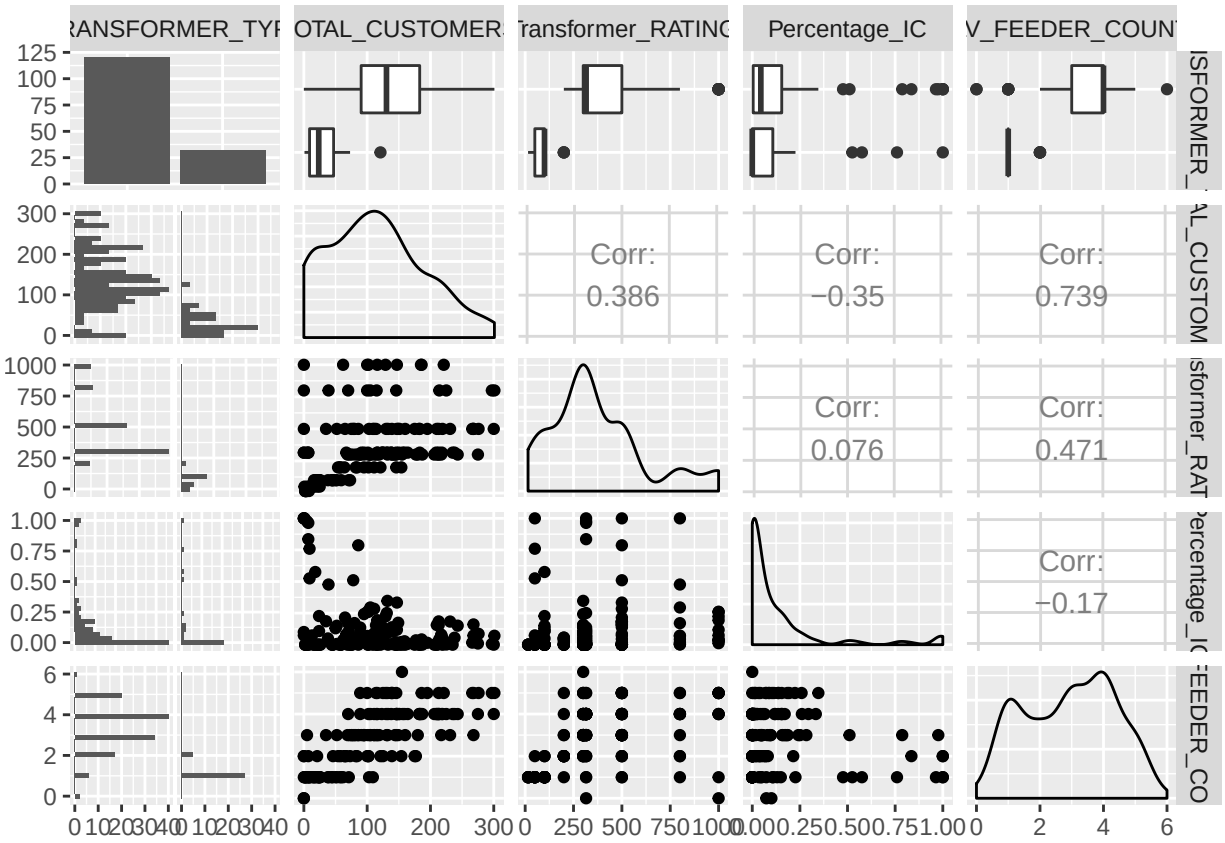
```
##  SUBSTATION_NUMBER   TRANSFORMER_TYPE TOTAL_CUSTOMERS Transformer_RATING
##  Min.   :531057    Ground Mtd:0       Min.   :0.0     Min.   : 25.00
##  1st Qu.:531644    Pole Mtd  :3       1st Qu.:0.5     1st Qu.: 37.50
##  Median :532232                       Median :1.0     Median : 50.00
##  Mean   :531841                       Mean   :1.0     Mean   : 58.33
##  3rd Qu.:532234                       3rd Qu.:1.5     3rd Qu.: 75.00
##  Max.   :532235                       Max.   :2.0     Max.   :100.00
##  Percentage_IC    LV_FEEDER_COUNT  GRID_REFERENCE
##  Min.   :0.0000   Min.   :0.0000   Length:3
```

```
##  1st Qu.:0.5000   1st Qu.:0.5000   Class :character
##  Median :1.0000   Median :1.0000   Mode  :character
##  Mean   :0.6667   Mean   :0.6667
##  3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000
```

Question 7

```
# Let's dive deeper into the Substations characteristics:


# for cluster 1
ggpairs(Characteristics1,columns = 2:6)
```

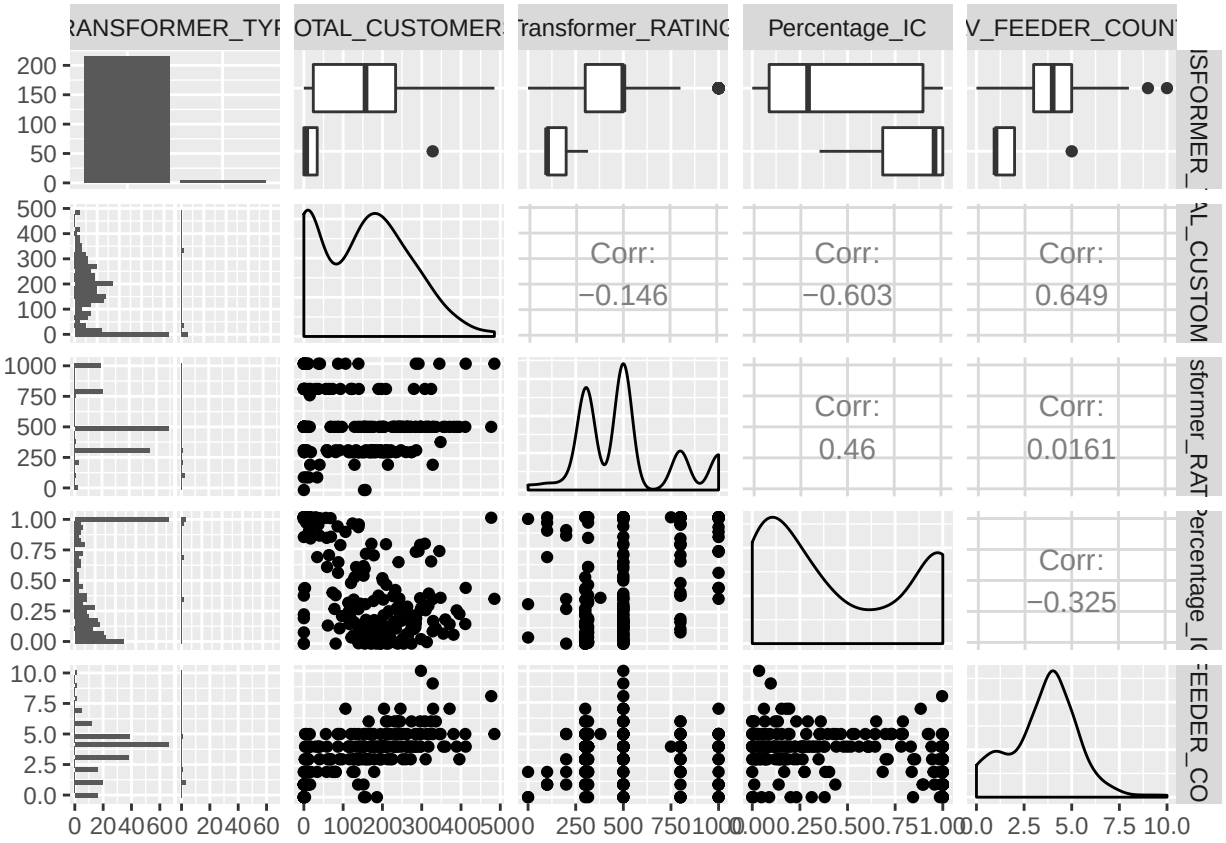
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



- Cluster 1:
  On the pair-plot for cluster 1, we can see that the percentages of industrial and commercial customers are mostly very low. There is a major peak around 0% which indicates that the vast majority of customers in the areas where the substations for cluster 1 are based are domestic customers. Furthermore, we can see from the total number of customers that those areas are not extremely populated and have for the most part transformer ratings that are on the lower end.

```
# for cluster 2
ggpairs(Characteristics2,columns = 2:6)
```

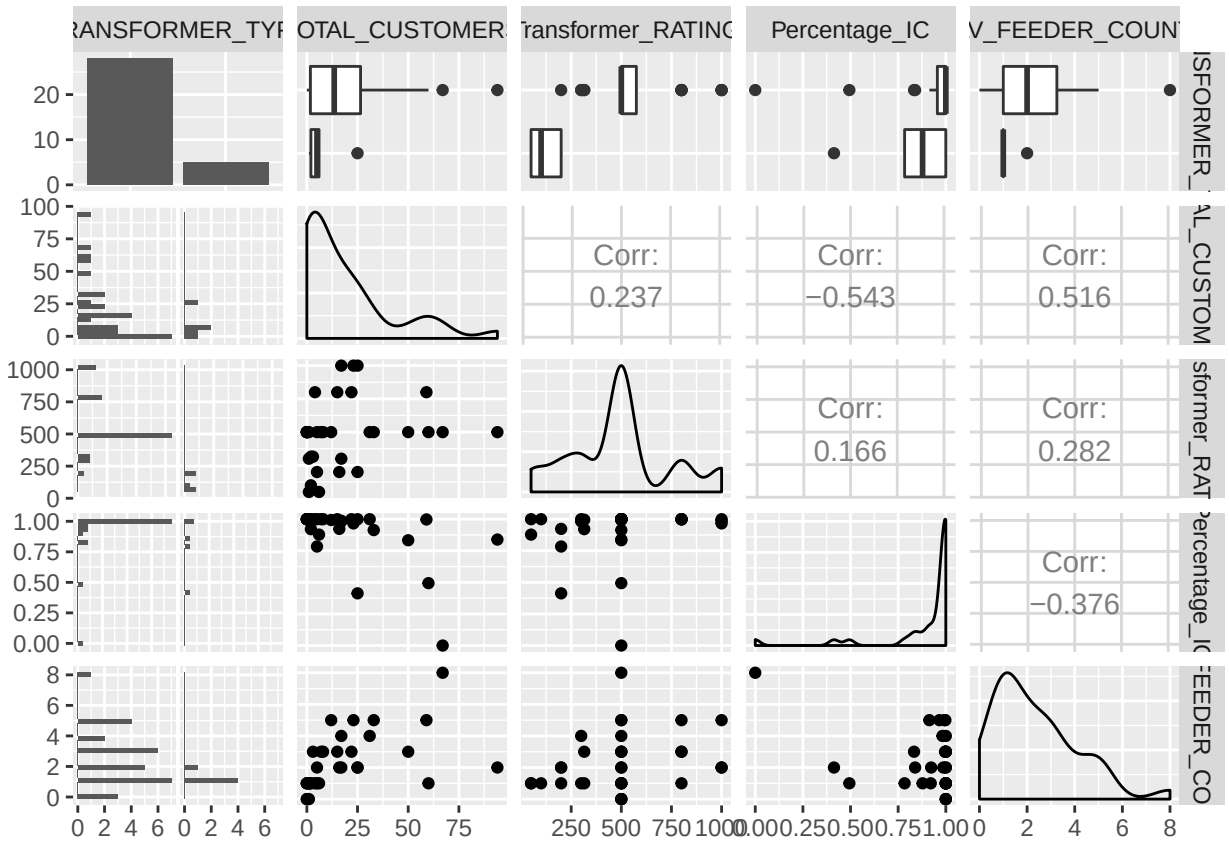
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



- Cluster 2:
  On the pair-plot for cluster 2, the percentages of industrial and commercial customers are more versatile. On one hand, there many substations which have customers of 100% industrial and commercial nature whereas on the other hand there also is a large portion of them ranging between 0 and 25% of industrial and commercial customers. It indicates that some of these areas are very industrial and commercial based but simultaneously also have many domestic customers. Moreover, the total number of customers distribution is torn between a major peak at about 0 customers and a more stretched out distribution ranging from about a 100 to 300 customers with an array of different transformer ratings.

```
# for cluster 3
ggpairs(Characteristics3,columns = 2:6)
```

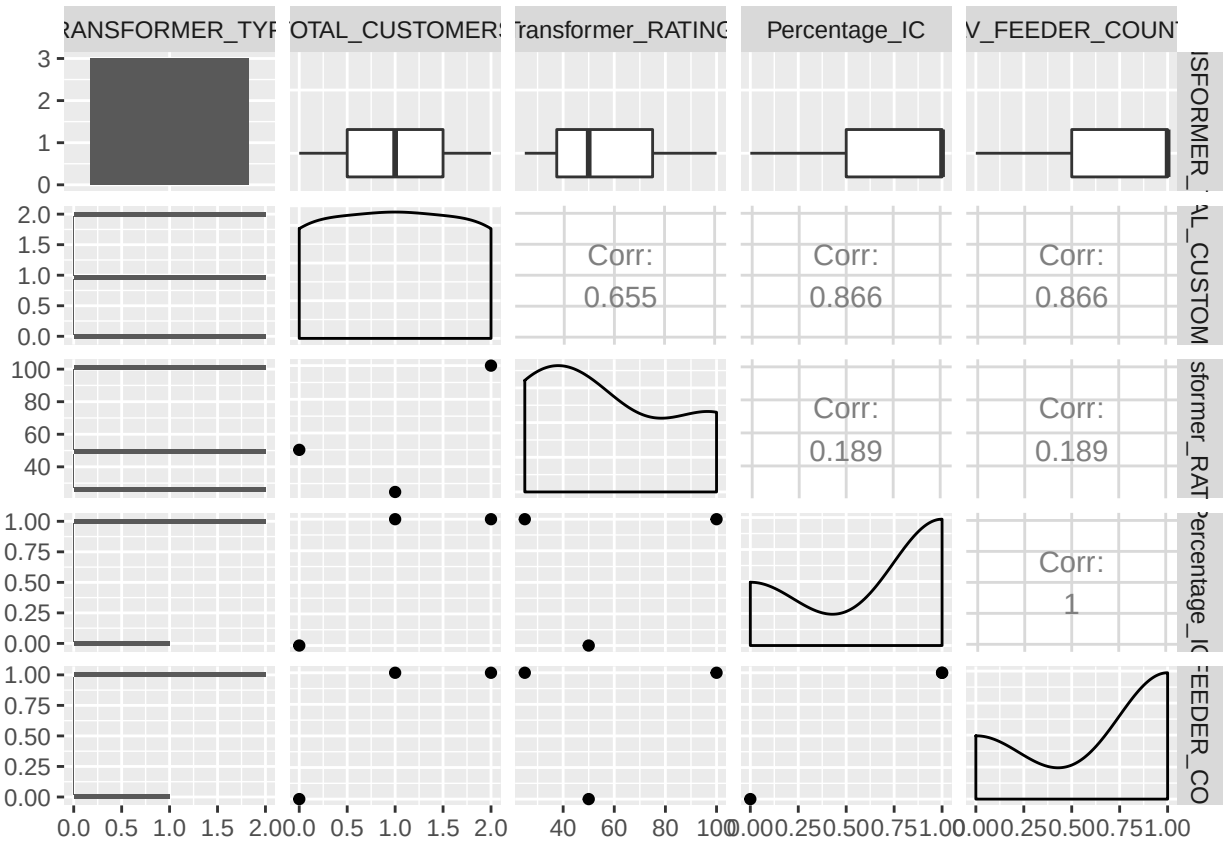
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



- Cluster 3:
  On the pair-plot for cluster 3, the percentages of customers are focused on a 100% of industrial and commercial. This indicates that there aren't barely any domestic customers in the areas covered by the substations in cluster 3. Regarding the total number of customers, they are mainly focused between 0 and 25 and the transformer ratings are mainly centered around 500 with more than half the values being exactly 500.

```
# for cluster 4
ggpairs(Characteristics4,columns = 2:6)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



- Cluster 4:
  Finally, on the fourth pair-plot (for cluster 4), the percentages distribution is simple: 2 substations have 100% industrial and commercial customers whereas 1 has 0% of industrial and commercial customers. This cluster is slightly different because of the low amount of substations actually contained in it. We can see, however that the total numbers of customers are very low (0,1 and 2).
  In order to investigate the clusters in a more thorough way, I computed a table of transformer type proportions for each cluster:

```
# Let's have a look at the percentages of the number of transformers per type per cluster:
Transformer_types_clstr1 <- table(Characteristics1$TRANSFORMER_TYPE)
Transformer_types_clstr2 <- table(Characteristics2$TRANSFORMER_TYPE)
Transformer_types_clstr3 <- table(Characteristics3$TRANSFORMER_TYPE)
Transformer_types_clstr4 <- table(Characteristics4$TRANSFORMER_TYPE)

perc_types_clstr1 <- data.frame(
  Ground=round(Transformer_types_clstr1[1]/sum(Transformer_types_clstr1),3),
  Pole=round(Transformer_types_clstr1[2]/sum(Transformer_types_clstr1),3)
  )
rownames(perc_types_clstr1) <- 'Cluster 1'

perc_types_clstr2 <- data.frame(
  Ground=round(Transformer_types_clstr2[1]/sum(Transformer_types_clstr2),3),
  Pole=round(Transformer_types_clstr2[2]/sum(Transformer_types_clstr2),3)
  )
rownames(perc_types_clstr2) <- 'Cluster 2'

perc_types_clstr3 <- data.frame(
  Ground=round(Transformer_types_clstr3[1]/sum(Transformer_types_clstr3),3),
```

```
  Pole=round(Transformer_types_clstr3[2]/sum(Transformer_types_clstr3),3)
  )
rownames(perc_types_clstr3) <- 'Cluster 3'

perc_types_clstr4 <- data.frame(
  Ground=round(Transformer_types_clstr4[1]/sum(Transformer_types_clstr4),3),
  Pole=round(Transformer_types_clstr4[2]/sum(Transformer_types_clstr4),3)
  )
rownames(perc_types_clstr4) <- 'Cluster 4'

Transformer_types_percentages <- rbind(perc_types_clstr1,
                                       perc_types_clstr2,
                                       perc_types_clstr3,
                                       perc_types_clstr4)

kable(Transformer_types_percentages)
```

	Ground	Pole
Cluster 1	0.789	0.211
Cluster 2	0.977	0.023
Cluster 3	0.848	0.152
Cluster 4	0.000	1.000

By comparing these values with what we found out from the individual pair plots for each cluster, we now have a good description of our clusters:
- Cluster 1 mainly consists of domestic customers, which indicates that these substations are based in residential areas and from the number of customers, we can deduce that it consists of less populated areas such as villages, which can be seen again on the transformer type table, with about 21% of pole mounted transformers and 79% of ground mounted transformers. The transformers can be placed on the ground but sometimes have to be placed on poles, showing some degree of rurality.
- Cluster 2 has a more diverse distribution of customer types. This indicates a different scale of customers, which could be translated to more diverse areas which consist of more populated towns or even cities. From the numbers of customers and the fact that we saw peaks at high numbers as well as low numbers, it indicates that we are also looking at some industrial zones.
- Cluster 3 solely consists of industries and commercial customers. It indicates areas with very low amounts of residents, such as industrial estates or shopping centers.
- Cluster 4 only contains three substations, and after inspecting their average power consumptions as well as the characteristics, we can see that it only consumes power during the night. Therefore, they must be close to motorways, or in remote and rural areas near a road which only requires power during the night (for lighting).

For the Village cluster, we can see clear patterns of low power consumption during the night, for all days. For Saturdays and week days, small peaks around early morning and then a fairly flat consumption throughout the day until evening, where we have the biggest peak of the day. Sundays are somewhat similar, except for a later and longer morning/afternoon peak.
For the Town and City cluster, the night time power demand is mostly low for all days. For week days, the demand increases in the morning and stays fairly constant during the afternoon and finally, peaks during the evening. For Saturdays and Sundays, it increases in the morning, decreases slightly in the afternoon and similarly to week days, peaks in the evening.
For the Industrial cluster, the demand is low during week days during evenings and during the night. During mornings and afternoons, however it largely increases. Saturdays and Sundays, the demand is higher than week days over the course of evenings and nights and increases around late morning and early afternoon.
Finally, for the Motorway cluster, there only is power demand during evenings and night time and barely has any during the day.

Allocating new substations

Question 8

```
# Loading the data set
NewSubstations <- read.csv('NewSubstations.csv')
NewSubstations$Date <- as.Date(NewSubstations$Date)
NewSubstations$Day <- weekdays(NewSubstations$Date)
# renaming the day levels for easier filtering in the future
new_day_fact <- factor(NewSubstations$Day)
levels(new_day_fact) <- c('Weekday','Weekday',
                          'Saturday','Sunday',
                          'Weekday','Weekday',
                          'Weekday')

NewSubstations$Day <- new_day_fact


# find names of Substations to select them
levels(factor(NewSubstations$Substation))
```

```
## [1] "511079" "512457" "532697" "552863" "563729"
```

```
## Substation 511079
Sub1 <- NewSubstations %>%
  filter(Substation=='511079')

# Filter data frame by type of day
Sub1All <- Sub1 %>%
  select(-Date,-Day,-X) %>%
  group_by(Substation) %>%
  summarise_all(mean)

Sub1Week <- Sub1 %>%
  filter(Day=='Weekday') %>%
  select(-Date,-Day,-X) %>%
  group_by(Substation) %>%
  summarise_all(mean)

Sub1Sat <- Sub1 %>%
  filter(Day=='Saturday') %>%
  select(-Date,-Day,-X) %>%
  group_by(Substation) %>%
  summarise_all(mean)

Sub1Sun <- Sub1 %>%
  filter(Day=='Sunday') %>%
  select(-Date,-Day,-X) %>%
  group_by(Substation) %>%
  summarise_all(mean)

# combining to prepare for plotting
Sub1DF <- rbind(Sub1All,Sub1Week,Sub1Sat,Sub1Sun)

# Adding the type of day for plotting multiple lines per type of day
```

```
Sub1DF$Day_Type <- c('All days','Week days','Saturdays','Sundays')

# Collapsing columns for plotting
Sub1DF <- Sub1DF %>%
  gather(key='time_interval',value='average_power',-Substation,-Day_Type) %>%
  mutate(daytype=as.factor(Day_Type))

# Remove the 'X' in the interval name to be able to convert it to numeric
Sub1DF$time_interval <- as.numeric(gsub('X','',Sub1DF$time_interval))

NewPlot1 <- ggplot(Sub1DF,aes(x=time_interval,y=average_power,colour=Day_Type))+
  geom_line()+
  labs(x='',y='') + # Same x y labels for all Substations
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Substation 511079')

## Substation 512457
Sub2 <- NewSubstations %>%
  filter(Substation=='512457')

# Filter data frame by type of day
Sub2All <- Sub2 %>%
  select(-Date,-Day,-X) %>%
  group_by(Substation) %>%
  summarise_all(mean)

Sub2Week <- Sub2 %>%
  filter(Day=='Weekday') %>%
  select(-Date,-Day,-X) %>%
  group_by(Substation) %>%
  summarise_all(mean)

Sub2Sat <- Sub2 %>%
  filter(Day=='Saturday') %>%
  select(-Date,-Day,-X) %>%
  group_by(Substation) %>%
  summarise_all(mean)

Sub2Sun <- Sub2 %>%
  filter(Day=='Sunday') %>%
  select(-Date,-Day,-X) %>%
  group_by(Substation) %>%
  summarise_all(mean)

# combining to prepare for plotting
Sub2DF <- rbind(Sub2All,Sub2Week,Sub2Sat,Sub2Sun)

# Adding the type of day for plotting multiple lines per type of day
Sub2DF$Day_Type <- c('All days','Week days','Saturdays','Sundays')

# Collapsing columns for plotting
```

```
Sub2DF <- Sub2DF %>%
  gather(key='time_interval',value='average_power',-Substation,-Day_Type) %>%
  mutate(daytype=as.factor(Day_Type))

# Remove the 'X' in the interval name to be able to convert it to numeric
Sub2DF$time_interval <- as.numeric(gsub('X','',Sub2DF$time_interval))

NewPlot2 <- ggplot(Sub2DF,aes(x=time_interval,y=average_power,colour=Day_Type))+
  geom_line()+
  labs(x='',y='' )+ # Same x y labels for all Substations
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Substation 512457')

## Substation 532697
Sub3 <- NewSubstations %>%
  filter(Substation=='532697')

# Filter data frame by type of day
Sub3All <- Sub3 %>%
  select(-Date,-Day,-X) %>%
  group_by(Substation) %>%
  summarise_all(mean)

Sub3Week <- Sub3 %>%
  filter(Day=='Weekday') %>%
  select(-Date,-Day,-X) %>%
  group_by(Substation) %>%
  summarise_all(mean)

Sub3Sat <- Sub3 %>%
  filter(Day=='Saturday') %>%
  select(-Date,-Day,-X) %>%
  group_by(Substation) %>%
  summarise_all(mean)

Sub3Sun <- Sub3 %>%
  filter(Day=='Sunday') %>%
  select(-Date,-Day,-X) %>%
  group_by(Substation) %>%
  summarise_all(mean)

# combining to prepare for plotting
Sub3DF <- rbind(Sub3All,Sub3Week,Sub3Sat,Sub3Sun)

# Adding the type of day for plotting multiple lines per type of day
Sub3DF$Day_Type <- c('All days','Week days','Saturdays','Sundays')

# Collapsing columns for plotting
Sub3DF <- Sub3DF %>%
  gather(key='time_interval',value='average_power',-Substation,-Day_Type) %>%
  mutate(daytype=as.factor(Day_Type))
```

```
# Remove the 'X' in the interval name to be able to convert it to numeric
Sub3DF$time_interval <- as.numeric(gsub('X','',Sub3DF$time_interval))

NewPlot3 <- ggplot(Sub3DF,aes(x=time_interval,y=average_power,colour=Day_Type))+
  geom_line()+
  labs(x='',y='')+ # same x y labels for all substations
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Substation 532697')

## Substation 552863
Sub4 <- NewSubstations %>%
  filter(Substation=='552863')

# Filter data frame by type of day
Sub4All <- Sub4 %>%
  select(-Date,-Day,-X) %>%
  group_by(Substation) %>%
  summarise_all(mean)

Sub4Week <- Sub4 %>%
  filter(Day=='Weekday') %>%
  select(-Date,-Day,-X) %>%
  group_by(Substation) %>%
  summarise_all(mean)

Sub4Sat <- Sub4 %>%
  filter(Day=='Saturday') %>%
  select(-Date,-Day,-X) %>%
  group_by(Substation) %>%
  summarise_all(mean)

Sub4Sun <- Sub4 %>%
  filter(Day=='Sunday') %>%
  select(-Date,-Day,-X) %>%
  group_by(Substation) %>%
  summarise_all(mean)

# combining to prepare for plotting
Sub4DF <- rbind(Sub4All,Sub4Week,Sub4Sat,Sub4Sun)

# Adding the type of day for plotting multiple lines per type of day
Sub4DF$Day_Type <- c('All days','Week days','Saturdays','Sundays')

# Collapsing columns for plotting
Sub4DF <- Sub4DF %>%
  gather(key='time_interval',value='average_power',-Substation,-Day_Type) %>%
  mutate(daytype=as.factor(Day_Type))

# Remove the 'X' in the interval name to be able to convert it to numeric
Sub4DF$time_interval <- as.numeric(gsub('X','',Sub4DF$time_interval))
```

```
NewPlot4 <- ggplot(Sub4DF,aes(x=time_interval,y=average_power,colour=Day_Type))+
  geom_line()+
  labs(x='',y='')+ # same x y labels for all substations
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Substation 552863')

## Substation 563729
Sub5 <- NewSubstations %>%
  filter(Substation=='563729')

# Filter data frame by type of day
Sub5All <- Sub5 %>%
  select(-Date,-Day,-X) %>%
  group_by(Substation) %>%
  summarise_all(mean)

Sub5Week <- Sub5 %>%
  filter(Day=='Weekday') %>%
  select(-Date,-Day,-X) %>%
  group_by(Substation) %>%
  summarise_all(mean)

Sub5Sat <- Sub5 %>%
  filter(Day=='Saturday') %>%
  select(-Date,-Day,-X) %>%
  group_by(Substation) %>%
  summarise_all(mean)

Sub5Sun <- Sub5 %>%
  filter(Day=='Sunday') %>%
  select(-Date,-Day,-X) %>%
  group_by(Substation) %>%
  summarise_all(mean)

# combining to prepare for plotting
Sub5DF <- rbind(Sub5All,Sub5Week,Sub5Sat,Sub5Sun)

# Adding the type of day for plotting multiple lines per type of day
Sub5DF$Day_Type <- c('All days','Week days','Saturdays','Sundays')

# Collapsing columns for plotting
Sub5DF <- Sub5DF %>%
  gather(key='time_interval',value='average_power',-Substation,-Day_Type) %>%
  mutate(daytype=as.factor(Day_Type))

# Remove the 'X' in the interval name to be able to convert it to numeric
Sub5DF$time_interval <- as.numeric(gsub('X','',Sub5DF$time_interval))

NewPlot5 <- ggplot(Sub5DF,aes(x=time_interval,y=average_power,colour=Day_Type))+
  geom_line()+
  labs(x='',y='')+ # Same x y labels for all Substations
```

```
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Substation 563729')

gg_fig <- ggarrange(NewPlot1,NewPlot2,NewPlot3,NewPlot4,NewPlot5,nrow=3)
gg_annotated <- annotate_figure(p=gg_fig,
                left = text_grob("Tooth length", color = "green", rot = 90)
                )
```

Question 9

```
## We need to find centers of clusters and measure the distances with new substations

## We need to compute euclidean distance of each sample to each center
# for each column, return min distance and therefore evaluate which cluster is most
# appropriate for each new substation

# Scaling Values for new substations to match cluster values and
# computing their daily averages
NewSubScaled <- data.frame()
NewSubNumeric <- NewSubstations[,c(-1,-2,-3,-148)]
for(i in 1:dim(NewSubNumeric)[1]){
  NewSubScaledRow <- NewSubNumeric[i,]/max(NewSubNumeric[i,])
  NewSubScaled <- rbind(NewSubScaled,NewSubScaledRow)
}
NewSubScaled$Substation <- NewSubstations$Substation
NewDailyAverages <- NewSubScaled %>%
  group_by(Substation) %>%
  summarise_all(mean)

# Then, we need to calculate the centroids of our clusters in order
# to be able to fit our new substations into them

clust1center <- colMeans(autumn1[,3:146])
clust2center <- colMeans(autumn2[,3:146])
clust3center <- colMeans(autumn3[,3:146])
clust4center <- colMeans(autumn4[,3:146])

centers <- rbind(clust1center,
                 clust2center,
                 clust3center,
                 clust4center)
# Now we can assign a cluster to the new substations by computing distances
# to cluster centers
center_dist <- sapply(seq_len(nrow(NewDailyAverages[,-1])),
                      function(i) apply(centers, 1,
                                        function(v) sum((NewDailyAverages[,-1][i, ]-v)^2)))
NewDailyAverages$Cluster <- as.factor(max.col(-t(center_dist)))
NewClust <- NewDailyAverages[,c(1,146)]
# clean table
kable(NewClust)
```

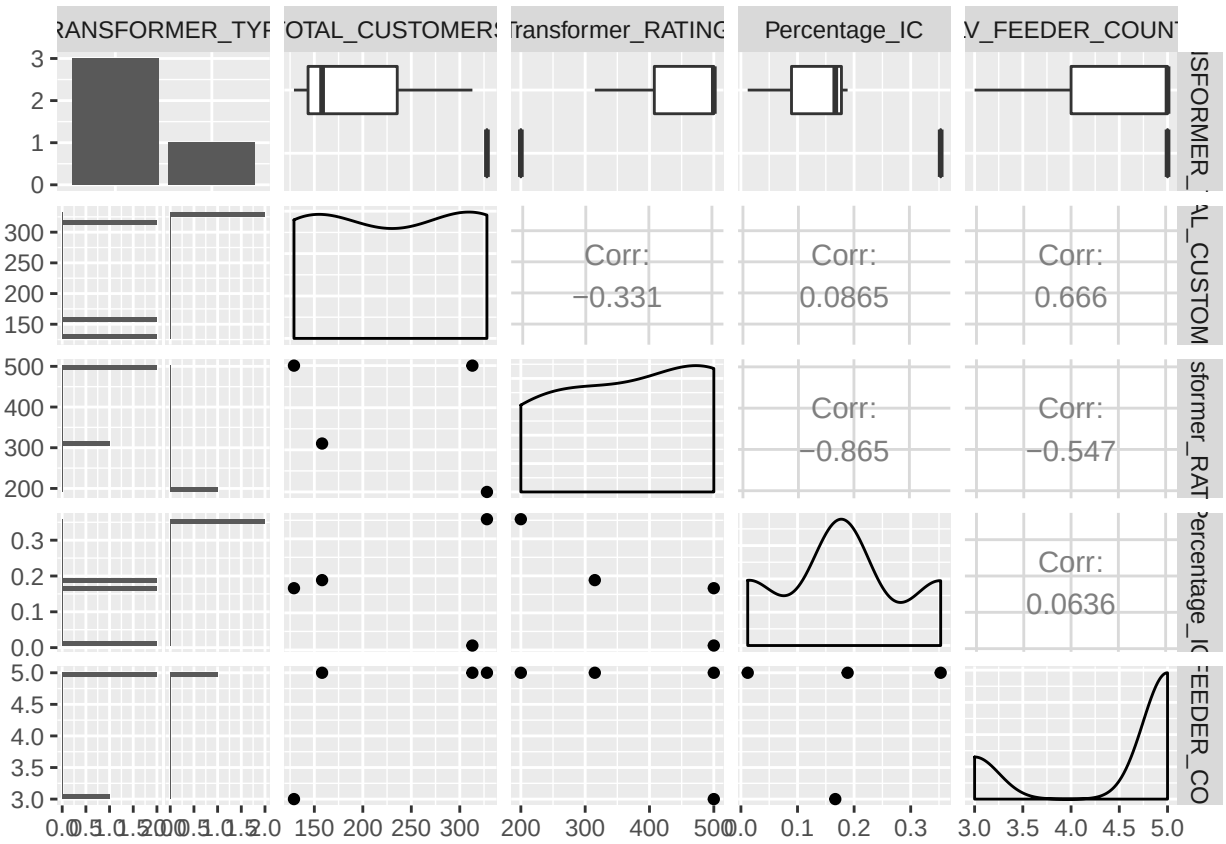
Substation	Cluster
511079	2
512457	3
532697	2
552863	2
563729	2

Question 10

As we can see, Substation number 512457 was allocated cluster 3 and the rest of the Substations were allocated cluster 2. When we look at their daily power demand plots, we can see that they fit the clusters they were allocated and that the allocations are not surprising.
Substation 512457's demand has the typical cluster 3 shape: very low over evenings and nights and has an arc-shaped increase over the day with a peak at around midday.
The rest of the Substations, however, have a cluster-2-like shape: very low during the night, with a first increase around early morning and then a second increase leading to the peak demand between mid-afternoon and evenings. The characteristics of the Substations also indicate a cluster 2 membership, as we can see on the following pair-plot:

```
NewClust2 <- NewClust%>%
  filter(Cluster == 2)

NewSubChar2 <- Characteristics %>%
  filter(SUBSTATION_NUMBER %in% NewClust2$Substation)

ggpairs(NewSubChar2[2:6])
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

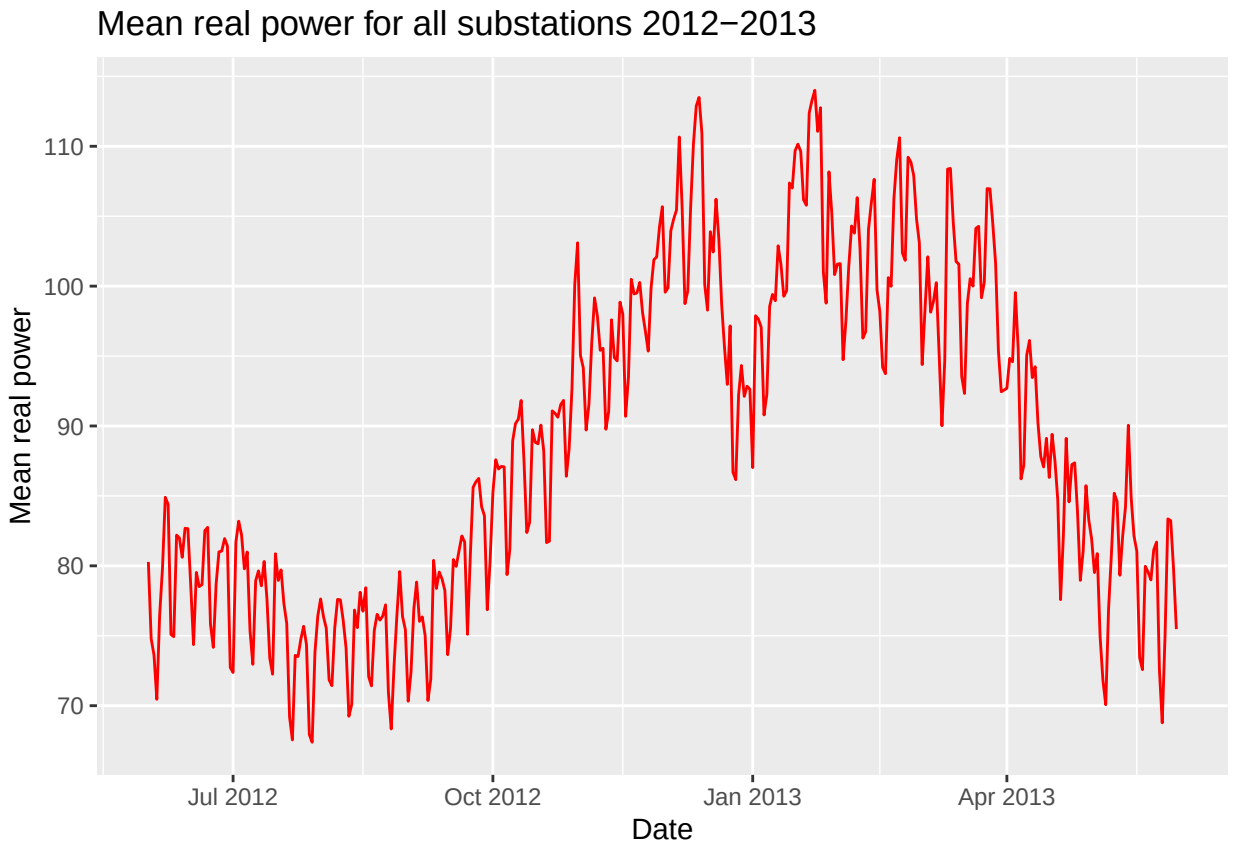

RANSFORMER_TYP
OTAL_CUSTOMERS
Transformer_RATING
Percentage_IC
V_FEEDER_COUNT
ISFORMER_
AL_CUSTOM
sformer_RAT
ercentage_IC
EEDER_CO
Corr:
−0.331
Corr:
0.0865
Corr:
0.666
Corr:
−0.865
Corr:
−0.547
Corr:
0.0636

Exploring differences between seasons

Question 11

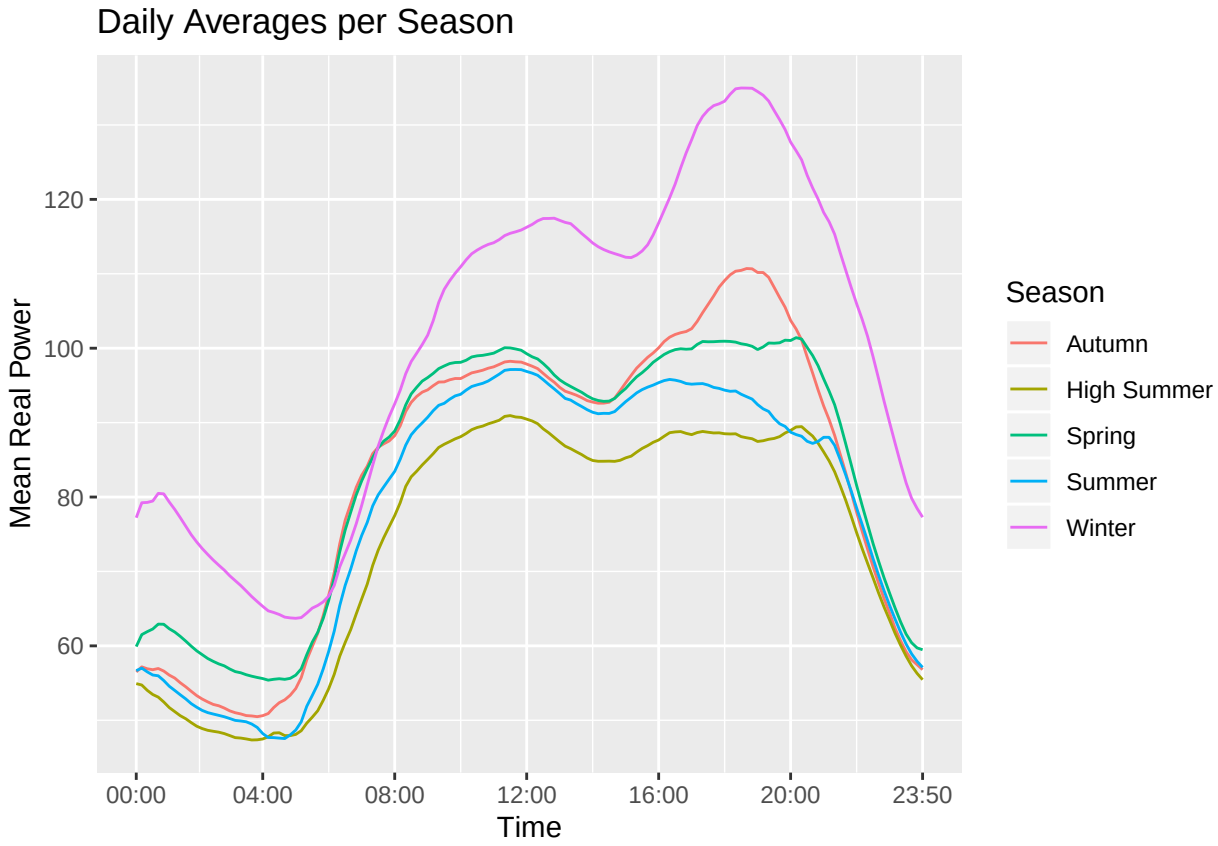
Analysis/Results

The aim of this analysis is to investigate seasonal patterns and detect any changes that may occur on cluster levels and to compare similarities between seasons. In order to initiate this investigation, an overall power demand per day can be observed in order to visualise the changes in power demands over the years 2012 and 2013. The following plot gives us an idea of the mean demand over the year for all substations.



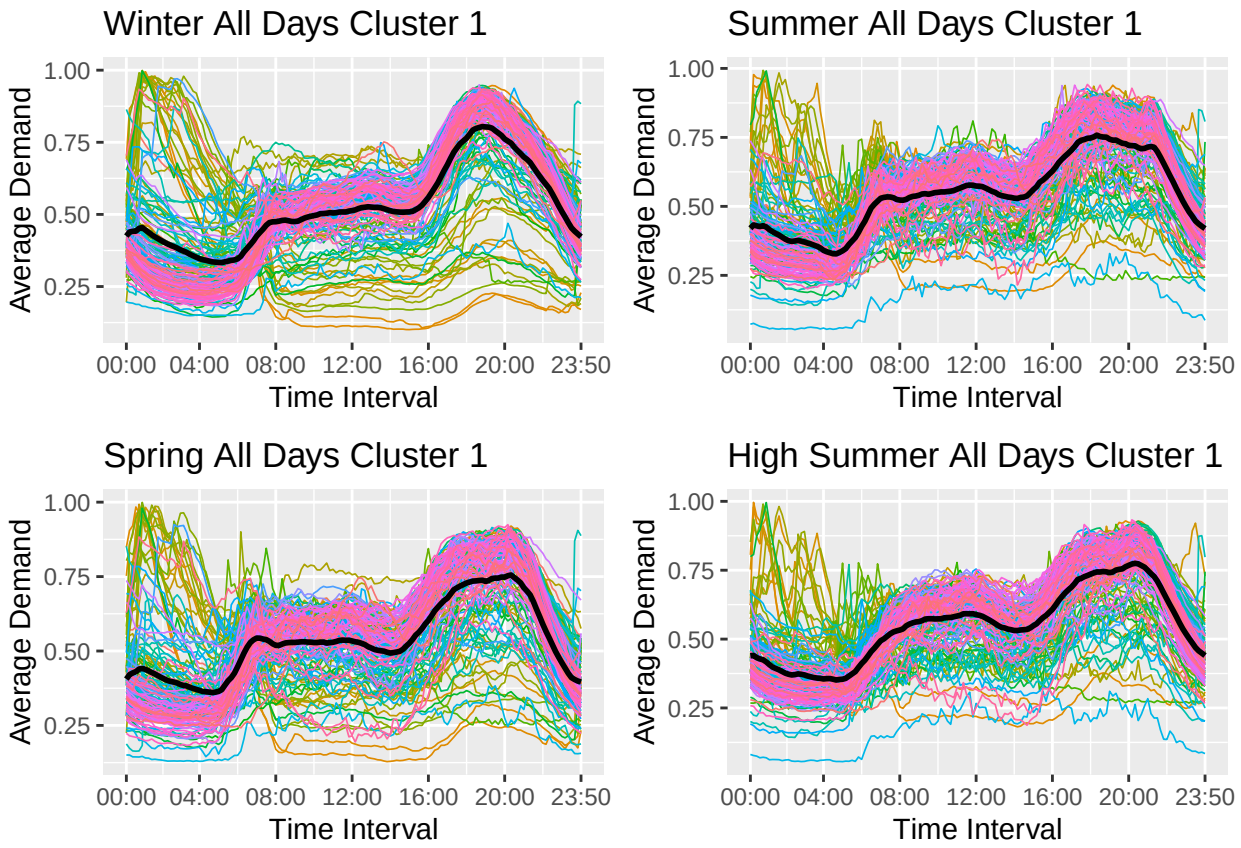
We can see that the real power demands are as expected; the demand during Spring, Summer and High Summer are the lowest, followed by and increase in Autumn, a peak in Winter (except for an odd slight dip in January) and a decrease again coming back to Spring.

```
ggplot(YearlyRealMean, aes(x=time_interval,y=real_power,colour=Season)) +
  geom_line() +
  labs(x='Time',y='Mean Real Power') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Daily Averages per Season')
```



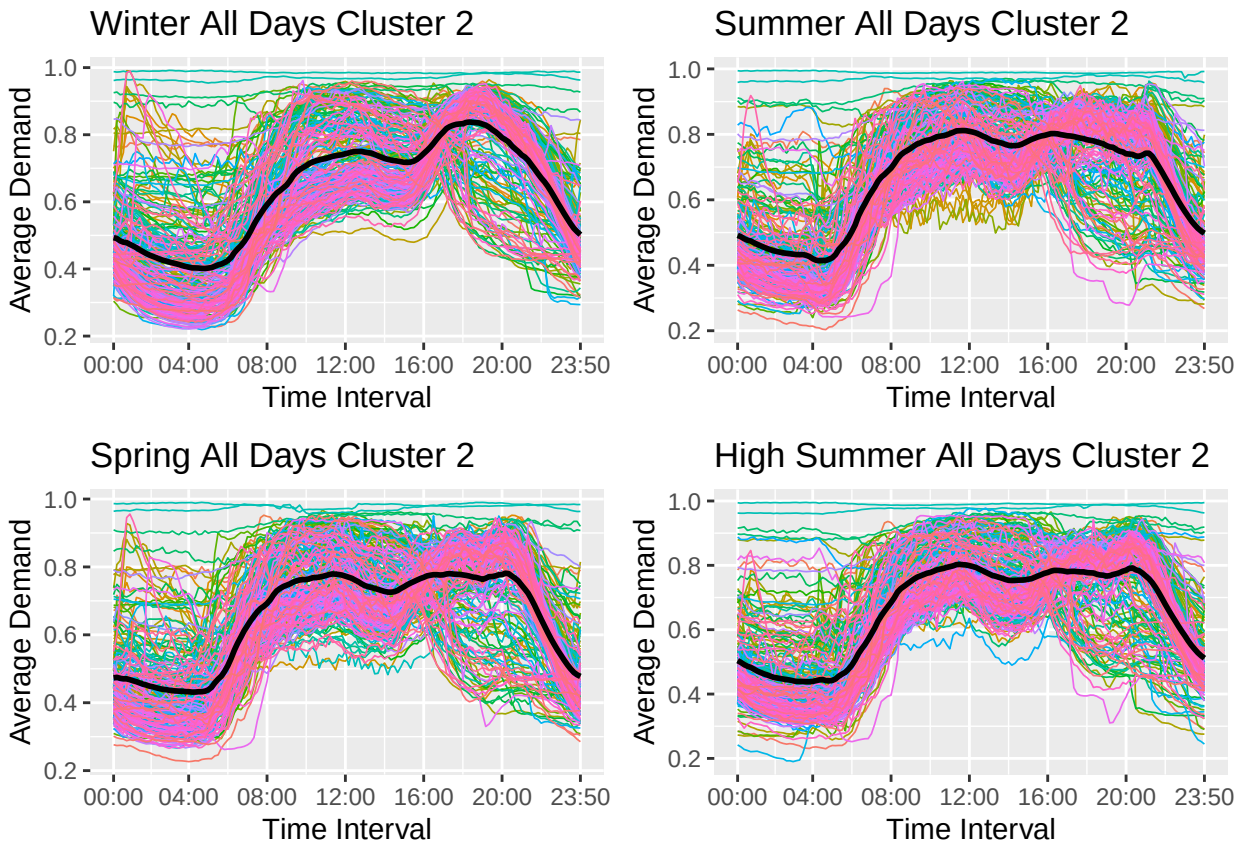
This plot puts into perspective what we have learned about the power consumption over seasons. In fact it is fairly as expected: Winter has the highest power demand, then Spring/Autumn, Summer and finally High Summer. Which brings us to the actual patterns within seasons. Let's investigate how the customers' power usage is spread out within seasons and within regions (clusters).
By scaling values within each Substation within each day, we'll be able to investigate interseasonal patterns and clustering similaritities and changes, in order to learn more about power distributions for specific seasons and localizations. First, let's observe how the power demands clusters look like by seasons. The following plots show the average demand for the clusters obtained for the new seasons.

Here is the demand for each season for cluster 1:



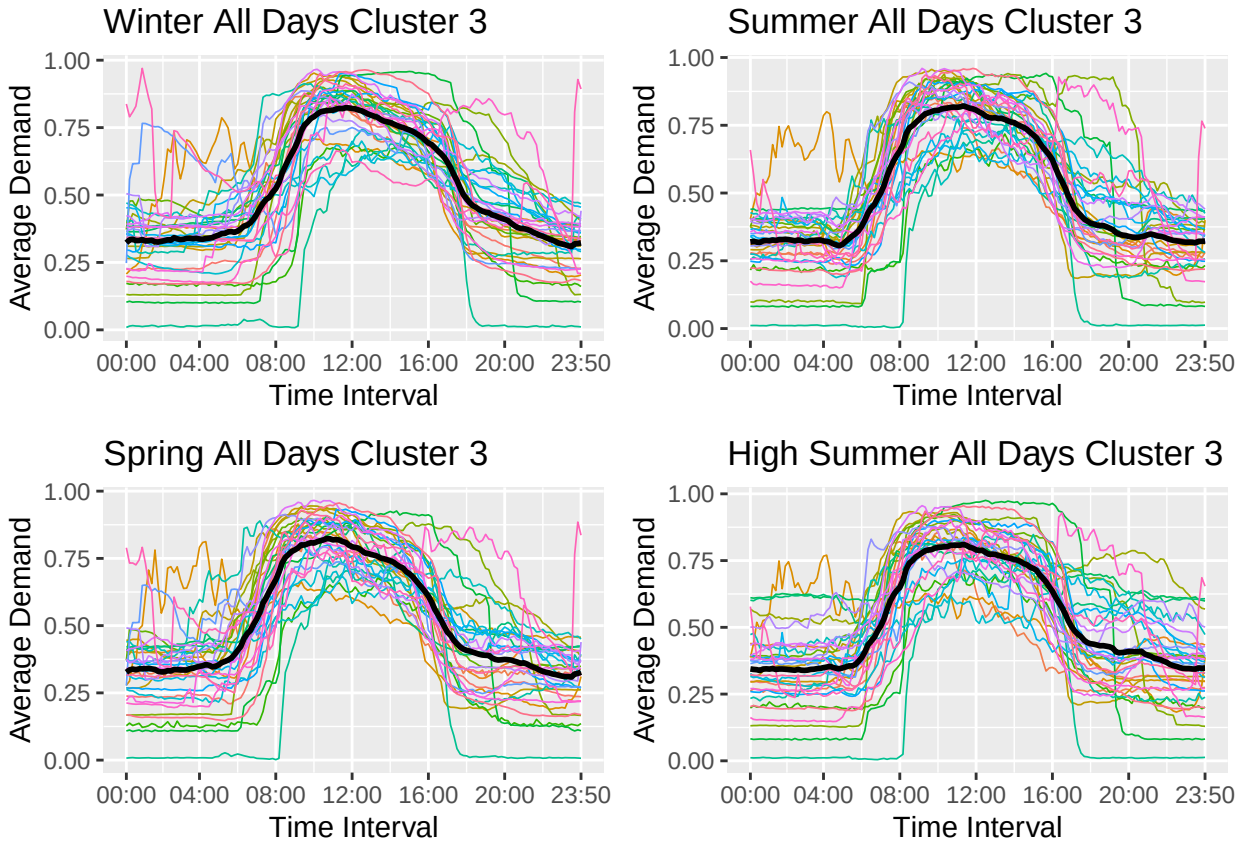
- For Cluster 1, we can see overall very similar patterns within the clusters of each seasons with some minor differences. Spring, Summer and Autumn are the most similar: we can see a clear distinct peak during the morning, although the power demands for the evening are slightly different during Autumn. The evening power demand is more focused around 19:30-20:00 whereas the demand for Spring and Summer is more spread between around 17:00 and 20:00 for Spring and even 17:00 and 21:00 for Summer. This spread over the evening can be found again during High Summer, although the morning pattern during High Summer indicates a fairly unique daily pattern in a way: it also has a spread out morning demand , showing versatility between substations.
  Cluster 1 is the village cluster and therefore the electricity is focused mainly on domestic customers for whome those power demand patterns make sense. Spring and Summer show the arrival of sunny days. We can clearly see on the plot how people are more laid back, drawing out evenings. The Autumn and Winter patterns are immediately more rough, showing a sort of more straight to the point evening routine, where much relaxation is not introduced. The days are also shorter and the Sun goes done much quicker, which explains the sudden peak around early evening. Finally, during High Summer the daily demand is increasing more gradually, where the mornings are spread over a couple of hours and so are the evenings. Winter and High Summer also are the seasons during which most people have longer holidays, which is depcited on the plots.

Then, there is the demand for each season for cluster 2:



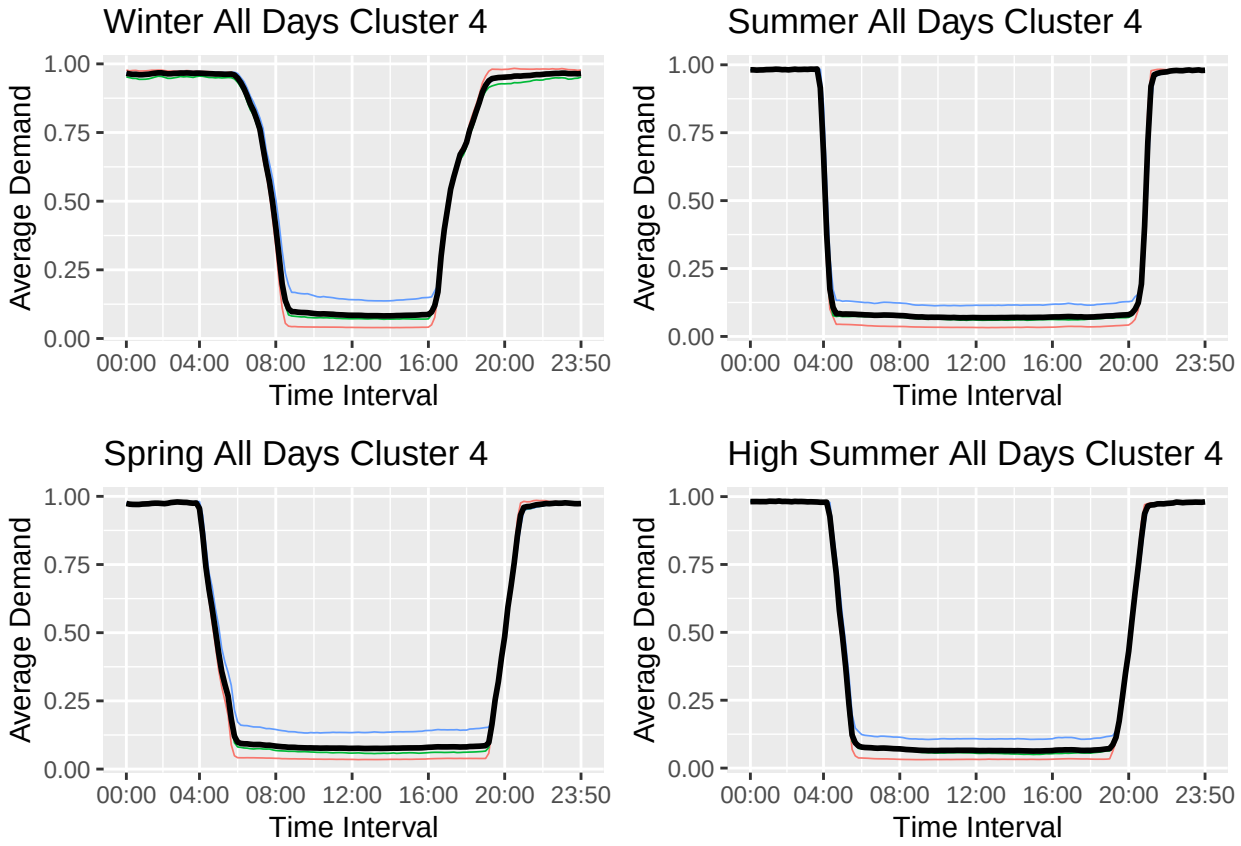
- For Cluster 2, we can see again similarities between the same seasons: Winter and Autumn and then Spring, Summer and High Summer. The overall morning patterns are more or less the same for all seasons, an arc like pattern that spreads out from around 5:00 to 12:00. The major difference here in the intensity rather than the pattern itself: for Winter and Autumn the afternoon shows a slight dip but has a larger increase than during the morning late afternoon/early evening, whereas for Spring, Summer and High Summer the difference in increase between the morning and the late afternoon increase is not important. It is almost less important for the evenings than for the mornings.
  Since Cluster 2 is focused on larger agglomerations, towns and cities, those patterns make sense: the need for power in the evening is larger than during the night since it gets colder and lighting will be turned on earlier on almost simultaneously, creating a high demand peak around the time where the Sun goes down. During Spring, Summer and High Summer, however, the days get warmer and the need for heating is not present anymore, resulting in a fairly constant power demand over the course of the day (excluding night time).

And the demand during each season for cluster 3:



- For Cluster 3, the pattern is very similar for every season, which is understandable since it is the industrial/commercial cluster. Whatever the seasons, the power consumption will be more or less the same all year round. The only difference between seasons might be a slight movement of the peak within 10:00 to 12:00 but overall, seasonal change doesn't affect the power demands.

And finally, the demand during each season for cluster 4:



- Finally, for cluster 4 the main change is as expected: the major differences are due to the length of the days during the season in question since the cluster is focused on rural areas and motorways. For example, there is a blatant difference Winter and Summer, with Winter having shorter days, therefore increasing the power demand over longer periods of time.

Looking at those plots, we can see how the power demand changes over the seasons by clusters. Some clusters go through more changes than others, which is understandable, due to the locations the substations are at and therefore the different power needs from different places with different activities and daily routines.

Conclusion

We have gained insight in power demand patterns throughout the year. Some were expected such as the yearly real power demands and the overall daily averages for each season, and some were trickier to predict: the cluster patterns. We have seen that villages, towns, larger cities and secluded rural areas have interesting power consumption routines which coincide with the seasonal changes. But we have also seen that commercial and industrial regions have a fairly similar consumption throughout the year and throughout the seasons. This showed us which clusters to focus on for future analysis and monitoring, in order to improve our understanding of power demand.

Appendix

```
library(tidyverse)
library(ggplot2)
library(Rmisc)
library(GGally)
library(chron)
```

```r
library(cclust)
library(taRifx)

### Question 11

load('HighSummer_2012.rdata')
load('Spring_2013.rdata')
load('Summer_2012.rdata')
load('Winter_2012.rdata')

HighSummer <- HighSummer_2012
Summer <- Summer_2012
Winter <- Winter_2012
Spring <- Spring_2013

# Let's take a look at an overview of means over the whole year

# Separating Scaled and Real values

HighSummerScaled <- HighSummer[,c(1:146)]
SummerScaled <- Summer[,c(1:146)]
WinterScaled <- Winter[,c(1:146)]
SpringScaled <- Spring[,c(1:146)]

HighSummerReal <- HighSummer[,c(1,2,148:291)]
SummerReal <- Summer[,c(1,2,148:291)]
WinterReal <- Winter[,c(1,2,148:291)]
SpringReal <- Spring[,c(1,2,148:291)]
AutumnReal <- AutumnData[,c(1,2,148:291)]

# new df with all real seasonal values
RealYear <- rbind(HighSummerReal,
                  SummerReal,
                  WinterReal,
                  SpringReal,
                  AutumnReal)

RealYear$Date <- dates(RealYear$Date,origin = c(month = 1,day = 1,year = 1970))

# grouping by date
YearlyMean <- RealYear %>%
  select(-Station)%>%
  mutate(Date=as.Date(Date)) %>%
  group_by(Date) %>%
  summarise_all(mean)

# computing means per date
OverallMeans <- data.frame(Date=YearlyMean$Date,
                           Mean_Power=rowMeans(YearlyMean[,-1]))

ggplot(OverallMeans,aes(x=Date,y=Mean_Power))+
  geom_line(colour='red') +
  labs(x='Date',y='Mean real power')+
```

```
  ggtitle('Mean real power for all substations 2012-2013')

# computing real daily averages to see overall demands per time
# interval and adding the season factor for future grouping

HighSummerReal$Season <- 'High Summer'
HighSummerRealAvg <- HighSummerReal[c(-1,-2)]

SummerReal$Season <- 'Summer'
SummerRealAvg <- SummerReal[c(-1,-2)]

SpringReal$Season <- 'Spring'
SpringRealAvg <- SpringReal[c(-1,-2)]

WinterReal$Season <- 'Winter'
WinterRealAvg <- WinterReal[c(-1,-2)]

AutumnReal$Season <- 'Autumn'
AutumnRealAvg <- AutumnReal[c(-1,-2)]

# new df with all real seasonal values and season factor
YearlyRealSeason <- rbind(HighSummerRealAvg,
                          SummerRealAvg,
                          WinterRealAvg,
                          SpringRealAvg,
                          AutumnRealAvg)

# grouping by season
YearlyRealMean <- YearlyRealSeason %>%
  group_by(Season) %>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=real_power,-Season) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Season = as.factor(Season))

ggplot(YearlyRealMean, aes(x=time_interval,y=real_power,colour=Season)) +
  geom_line() +
  labs(x='Time',y='Mean Real Power') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Daily Averages per Season')



WinterAverages <- WinterScaled[-2] %>%
  group_by(Station) %>%
  summarise_all(mean) %>%
  mutate(Station=as.factor(Station))

SummerAverages <- SummerScaled[-2] %>%
  group_by(Station) %>%
  summarise_all(mean) %>%
```

```
  mutate(Station=as.factor(Station))

HighSummerAverages <- HighSummerScaled[-2] %>%
  group_by(Station) %>%
  summarise_all(mean) %>%
  mutate(Station=as.factor(Station))

SpringAverages <- SpringScaled[-2] %>%
  group_by(Station) %>%
  summarise_all(mean) %>%
  mutate(Station=as.factor(Station))


###################################

# At this point, each season was run on a separate script to avoid confusion
# with some of the variables having the same names
# (code was replicated for each season)

############## WINTER #############

WinterScaled$Julian_Date <- WinterScaled$Date
WinterScaled$Date <- dates(WinterScaled[,2], origin = c(month = 1,day = 1,year = 1970))
WinterScaled$Day <- weekdays(as.Date(WinterScaled$Date,'%m/%d/%y'))
# renaming the day levels for easier filtering in the future
day_fact <- factor(WinterScaled$Day)
levels(day_fact) <- c('Sunday','Weekday',
                      'Weekday','Weekday',
                      'Weekday','Saturday',
                      'Weekday')

WinterScaled$Day <- day_fact

# Separating days

###  Cluster 1 stations
# filtering for cluster 1 stations for different days
winter1 <- WinterScaled %>%
  filter(Station %in% stations1$Station)

## All days

Station1All <- winter1 %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean of all days
s1meanAll <- winter1 %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
```

```
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Plots

Ws1p <- ggplot(Station1All, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s1meanAll,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Winter All Days Cluster 1')


### Cluster 2 stations

# filtering for cluster 2 stations
winter2 <- WinterScaled %>%
  filter(Station %in% stations2$Station)

## All days

Station2All <- winter2 %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean
s2meanAll <- winter2 %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Plots

Ws2p <- ggplot(Station2All, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s2meanAll,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Winter All Days Cluster 2')

###  Cluster 3 stations
# filtering for cluster 3 stations for different days
winter3 <- WinterScaled %>%
```

```
  filter(Station %in% stations3$Station)

## All days

Station3All <- winter3 %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean of all days
s3meanAll <- winter3 %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Plots

Ws3p <- ggplot(Station3All, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s3meanAll,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Winter All Days Cluster 3')

###  Cluster 4 stations
# filtering for cluster 4 stations
winter4 <- WinterScaled %>%
  filter(Station %in% stations4$Station)

## All days

Station4All <- winter4 %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean of all days
s4meanAll <- winter4 %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Plots
```

```
Ws4p <- ggplot(Station4All, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s4meanAll,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Winter All Days Cluster 4')


############## HIGH SUMMER ##############

HighSummerScaled$Julian_Date <- HighSummerScaled$Date
HighSummerScaled$Date <- dates(HighSummerScaled[,2], origin = c(month = 1,day = 1,year = 1970))
HighSummerScaled$Day <- weekdays(as.Date(HighSummerScaled$Date,'%m/%d/%y'))
# renaming the day levels for easier filtering in the future
day_fact <- factor(HighSummerScaled$Day)
levels(day_fact) <- c('Sunday','Weekday',
                      'Weekday','Weekday',
                      'Weekday','Saturday',
                      'Weekday')

HighSummerScaled$Day <- day_fact

# Separating days


###  Cluster 1 stations
# filtering for cluster 1 stations for different days
HighSummer1 <- HighSummerScaled %>%
  filter(Station %in% stations1$Station)

## All days

Station1All <- HighSummer1 %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean of all days
s1meanAll <- HighSummer1 %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Plots

Hs1p <- ggplot(Station1All, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
```

```
  geom_line(data=s1meanAll,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('High Summer All Days Cluster 1')


### Cluster 2 stations
# filtering for cluster 2 stations
HighSummer2 <- HighSummerScaled %>%
  filter(Station %in% stations2$Station)

## All days

Station2All <- HighSummer2 %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean
s2meanAll <- HighSummer2 %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Plots

Hs2p <- ggplot(Station2All, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s2meanAll,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('High Summer All Days Cluster 2')

###  Cluster 3 stations
# filtering for cluster 3 stations for different days
HighSummer3 <- HighSummerScaled %>%
  filter(Station %in% stations3$Station)

## All days

Station3All <- HighSummer3 %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
```

```
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean of all days
s3meanAll <- HighSummer3 %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Plots

Hs3p <- ggplot(Station3All, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s3meanAll,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('High Summer All Days Cluster 3')

###  Cluster 4 stations
# filtering for cluster 4 stations for different days
HighSummer4 <- HighSummerScaled %>%
  filter(Station %in% stations4$Station)

## All days

Station4All <- HighSummer4 %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean of all days
s4meanAll <- HighSummer4 %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Plots

Hs4p <- ggplot(Station4All, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s4meanAll,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('High Summer All Days Cluster 4')
```

```
############## SPRING ##############

SpringScaled$Julian_Date <- SpringScaled$Date
SpringScaled$Date <- dates(SpringScaled[,2], origin = c(month = 1,day = 1,year = 1970))
SpringScaled$Day <- weekdays(as.Date(SpringScaled$Date,'%m/%d/%y'))
# renaming the day levels for easier filtering in the future
day_fact <- factor(SpringScaled$Day)
levels(day_fact) <- c('Sunday','Weekday',
                      'Weekday','Weekday',
                      'Weekday','Saturday',
                      'Weekday')

SpringScaled$Day <- day_fact

# Separating days


###  Cluster 1 stations
# filtering for cluster 1 stations for different days
Spring1 <- SpringScaled %>%
  filter(Station %in% stations1$Station)

## All days

Station1All <- Spring1 %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean of all days
s1meanAll <- Spring1 %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Plots

SPs1p <- ggplot(Station1All, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s1meanAll,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Spring All Days Cluster 1')


### Cluster 2 stations
# filtering for cluster 2 stations
```

```
Spring2 <- SpringScaled %>%
  filter(Station %in% stations2$Station)

## All days

Station2All <- Spring2 %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean
s2meanAll <- Spring2 %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Plots

SPs2p <- ggplot(Station2All, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s2meanAll,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Spring All Days Cluster 2')

###  Cluster 3 stations
# filtering for cluster 3 stations for different days
Spring3 <- SpringScaled %>%
  filter(Station %in% stations3$Station)

## All days

Station3All <- Spring3 %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean of all days
s3meanAll <- Spring3 %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
```

```
## Plots

SPs3p <- ggplot(Station3All, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s3meanAll,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Spring All Days Cluster 3')

###  Cluster 4 stations
# filtering for cluster 4 stations for different days
Spring4 <- SpringScaled %>%
  filter(Station %in% stations4$Station)

## All days

Station4All <- Spring4 %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean of all days
s4meanAll <- Spring4 %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Plots

SPs4p <- ggplot(Station4All, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s4meanAll,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Spring All Days Cluster 4')









############## SUMMER ##############

SummerScaled$Julian_Date <- SummerScaled$Date
SummerScaled$Date <- dates(SummerScaled[,2], origin = c(month = 1,day = 1,year = 1970))
```

```
SummerScaled$Day <- weekdays(as.Date(SummerScaled$Date,'%m/%d/%y'))
# renaming the day levels for easier filtering in the future
day_fact <- factor(SummerScaled$Day)
levels(day_fact) <- c('Sunday','Weekday',
                      'Weekday','Weekday',
                      'Weekday','Saturday',
                      'Weekday')

SummerScaled$Day <- day_fact

# Separating days


###  Cluster 1 stations
# filtering for cluster 1 stations for different days
Summer1 <- SummerScaled %>%
  filter(Station %in% stations1$Station)

## All days

Station1All <- Summer1 %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean of all days
s1meanAll <- Summer1 %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Plots

Ss1p <- ggplot(Station1All, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s1meanAll,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Summer All Days Cluster 1')


### Cluster 2 stations
# filtering for cluster 2 stations
Summer2 <- SummerScaled %>%
  filter(Station %in% stations2$Station)

## All days
```

```
Station2All <- Summer2 %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean
s2meanAll <- Summer2 %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Plots

Ss2p <- ggplot(Station2All, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s2meanAll,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Summer All Days Cluster 2')

###  Cluster 3 stations
# filtering for cluster 3 stations for different days
Summer3 <- SummerScaled %>%
  filter(Station %in% stations3$Station)

## All days

Station3All <- Summer3 %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean of all days
s3meanAll <- Summer3 %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Plots

Ss3p <- ggplot(Station3All, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s3meanAll,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
```

```
  labs(x='Time Interval',y='Average Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Summer All Days Cluster 3')

###  Cluster 4 stations
# filtering for cluster 4 stations for different days
Summer4 <- SummerScaled %>%
  filter(Station %in% stations4$Station)

## All days

Station4All <- Summer4 %>%
  select(-Date,-Day,-Julian_Date)%>%
  group_by(Station)%>%
  summarise_all(mean)%>%
  gather(key=time_interval,value=scaled_power,-Station) %>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))
# mean of all days
s4meanAll <- Summer4 %>%
  select(-Date,-Day,-Julian_Date)%>%
  summarise_all(mean) %>%
  gather(key=time_interval,value=scaled_power,-Station)%>%
  mutate(time_interval=as.numeric(time_interval)) %>%
  mutate(Station=as.factor(Station))

## Plots

Ss4p <- ggplot(Station4All, aes(x=time_interval,y=scaled_power,colour=Station))+
  geom_line(show.legend=FALSE,size=0.3)+
  geom_line(data=s4meanAll,aes(x=time_interval,y=scaled_power),colour='black',size=1)+
  labs(x='Time Interval',y='Average Demand') +
  scale_x_continuous(breaks=c(1,24,48,72,96,120,144),
labels=c("00:00","04:00","08:00","12:00",
"16:00","20:00","23:50"))+
  ggtitle('Summer All Days Cluster 4')


############################

# FINAL CLUSTER PLOTS

# Clusters 1
multiplot(Ws1p,SPs1p,Ss1p,Hs1p,cols=2)
# Clusters 2
multiplot(Ws2p,SPs2p,Ss2p,Hs2p,cols=2)
# Clusters 3
multiplot(Ws3p,SPs3p,Ss3p,Hs3p,cols=2)
# Clusters 4
multiplot(Ws4p,SPs4p,Ss4p,Hs4p,cols=2)
```