
# Daily Electricity Price and Demand Analysis

Statistical learning Project

Mojtaba Amini  Javier Alberto Bernal Sigala  Carmen Rocio Ortiz Benitez
Saeed Soufeh

July 19, 2022

# Contents



# 1 Introduction

Victoria is the second smallest state in surface, and the second most populous state in Australia, with a population of 6.7 million (in 2020) it is the state with the highest density. This state maintains an electricity consumption that corresponds to 21% of the entire country, especially in the capital, Melbourne (5 million). Aware that electricity is essential for people, in the following work we will try to draw conclusions about the demand and prices of electricity consumption of the inhabitants of Victoria using observation and statistical inference methods.

Statistical inference is defined as the process of using data analysis to infer on the properties of an underlying distribution, in other contexts the concept of inference is used as “learning” or “training”. Regression methods are a type of inference in which models are used to establish the relationship between a target variable and its predictor variables, these targets can belong to a normal distribution as in linear regression, to even a binomial as in logistic regression.

Thus, the aim of this project is to build a regression model to predict demand that takes into account the factors that influence it the most, and secondly building a prediction model with enough accuracy and efficiency. Additionally, we will build a logistic regression model to classify whether RRP (Recommended Retail Price) will be high or low.

## 1.1 Objectives

- Explore the data in depth in order to understand relations between variables and their behavior.
- Build a regression model to predict electricity demand.
- Build a logistic regression model to classify the RRP as above or below the median.

# 2 Data Collection

This dataset has 2,106 observations and 14 variables, and it is a record from January 2015 to October 2020 of daily electricity price and demand in the state of Victoria, Australia. It is available publicly (in https://www.kaggle.com/datasets/aramacus/electricity-demand-in-victoria-australia). Daily electricity may depend on different factors such as rainfall, solar exposure, whether it was a school day, etc; and also the correlations between them. It is clear to see how any of these variables could be meaningful in our analysis.

### 2.1 Variables description

The variables present on our dataset are:

- date: the date of the recording;
- demand: total daily electricity demand in MWh (megawatt hour);
- RRP: a recommended retail price in AUD$ / MWh (Australian Dollars per MWh);
- demand_pos_RRP: total daily demand at positive RRP in MWh;
- RRP_positive: averaged positive RRP, weighted by the corresponding daily demand in AUD$ / MWh (Australian Dollars per MWh);
- demand_neg_RRP: total daily demand at negative RRP in MWh (megawatt hour);
- RRP_negative: average negative RRP, weighted by the corresponding daily demand in AUD$ / MWh (Australian Dollars per MWh);
- frac_at_neg_RRP: fraction of the day when the demand was traded at negative RRP;
- min_temperature: minimum temperature during the day in Celsius;
- max_temperature: maximum temperature during the day in Celsius;
- solar_exposure: total daily sunlight energy in MJ/m^2 (megajoule per square meter);
- rainfall: rainfall during the day in mm;
- school_day: whether students were at school on that day;
- holiday: whether the day was a holiday.

It is worth noting that the RRP is calculated considering different components such as wholesale costs, network charges, and retail margin, all information which we do not have access too. Furthermore, the RRP and its related features (demand_pos_RRP, RRP_positive, demand_neg_RRP, RRP_negative, frac_at_neg_RRP) will depend strongly on demand and on each other, and that is why we choose not to consider them when building our model.

```
data <- read.csv(file = "complete_dataset.csv")
summary(data)
```

```
##      date               demand            RRP           demand_pos_RRP
##  Length:2106        Min.   : 85094   Min.   :  -6.076   Min.   : 41988
##  Class :character   1st Qu.:109964   1st Qu.:  38.707   1st Qu.:109246
##  Mode  :character   Median :119586   Median :  66.597   Median :119148
##                     Mean   :120035   Mean   :  76.080   Mean   :119252
##                     3rd Qu.:130436   3rd Qu.:  95.075   3rd Qu.:130119
##                     Max.   :170654   Max.   :4549.645   Max.   :170654
##
##   RRP_positive     demand_neg_RRP     RRP_negative      frac_at_neg_RRP
##  Min.   :  13.57   Min.   :    0.0   Min.   :-342.220   Min.   :0.000000
##  1st Qu.:  39.12   1st Qu.:    0.0   1st Qu.:   0.000   1st Qu.:0.000000
##  Median :  66.87   Median :    0.0   Median :   0.000   Median :0.000000
##  Mean   :  76.55   Mean   :  783.2   Mean   :  -2.686   Mean   :0.008547
##  3rd Qu.:  95.13   3rd Qu.:    0.0   3rd Qu.:   0.000   3rd Qu.:0.000000
##  Max.   :4549.65   Max.   :57597.6   Max.   :   0.000   Max.   :0.625000
##
##  min_temperature max_temperature solar_exposure     rainfall
##  Min.   : 0.60   Min.   : 9.00   Min.   : 0.70   Min.   : 0.000
##  1st Qu.: 8.50   1st Qu.:15.53   1st Qu.: 8.20   1st Qu.: 0.000
##  Median :11.30   Median :19.10   Median :12.70   Median : 0.000
##  Mean   :11.58   Mean   :20.41   Mean   :14.74   Mean   : 1.506
##  3rd Qu.:14.60   3rd Qu.:23.90   3rd Qu.:20.70   3rd Qu.: 0.800
##  Max.   :28.00   Max.   :43.50   Max.   :33.30   Max.   :54.600
##                                  NA's   :1       NA's   :3
##   school_day          holiday
##  Length:2106        Length:2106
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
##
```

In terms of continuous variables, the values from the RRP 3rd quantile and the maximum differ excessively, which suggests that this might be an outlier. Furthermore, there are only 4 null values and they located in the solar_exposure and rainfall columns. Distribution of demand itself does not vary too much, with values ranging from around 85,000 MWh to 170,000 MWh.

## 2.2 Data preparation

We check for "duplicate" rows and find there are none.

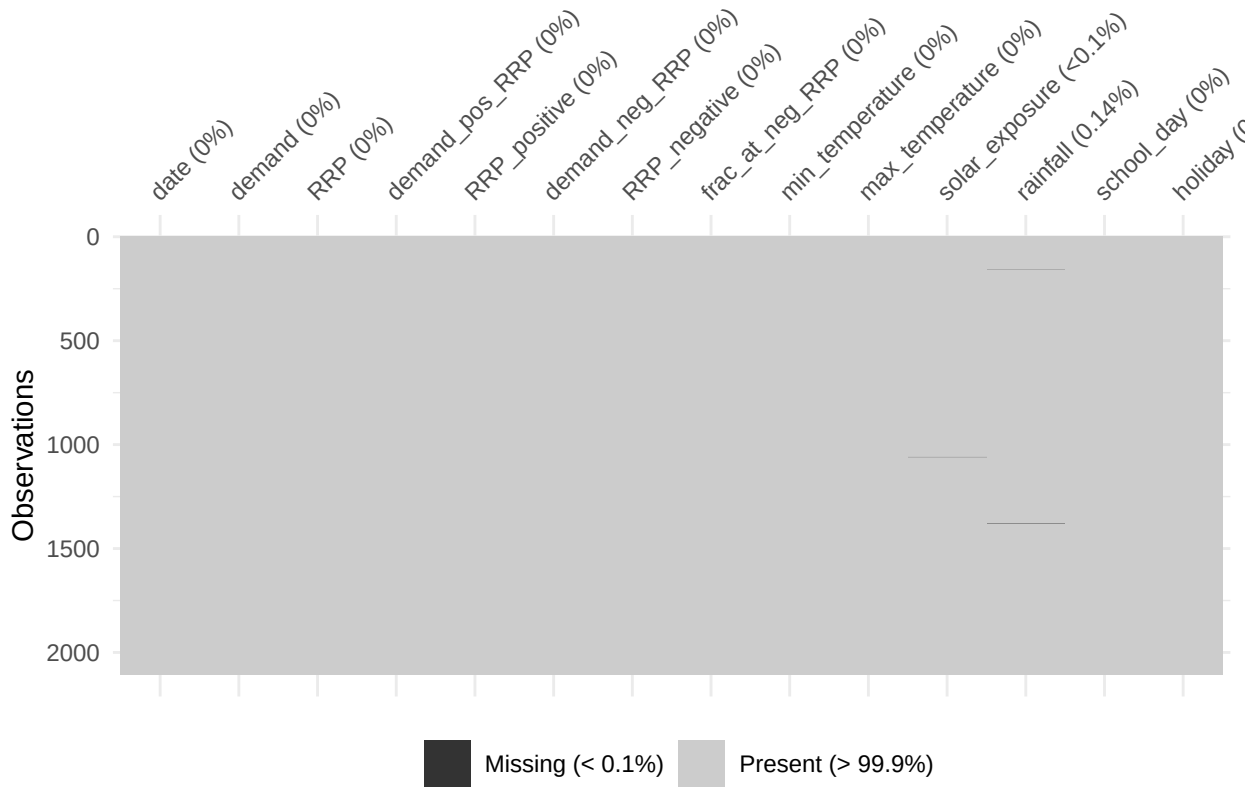
```
cat("Cheking for duplicated rows...", "The dataset has", sum(duplicated(data)), "duplicated rows.\n")
```

```
## Cheking for duplicated rows... The dataset has 0 duplicated rows.
```

In order to ease the analysis, we proceed to drop the find the missing values and remove them.

```
#see the missing values
vis_miss(data)
```

```
## Warning: `gather_()` was deprecated in tidyr 1.2.0.
## Please use `gather()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```




```
#drop missing solar_exposure
data <- data[!is.na(data$solar_exposure),]

#drop missing rainfall
data <- data[!is.na(data$rainfall),]
```

Next, we transform the date column from char to date, the min and max temperature into one combined mean temperature, and the holiday and school_day columns from char to categorical (binary).

```
#convert the date
data$date <- as.POSIXct(data$date)

#add the mean temperature variable
data$mean_temperature <- (data$max_temperature + data$min_temperature)/2

#turn the holidays to binary
data$holiday <- as.factor(data$holiday)
data <- data %>%
    mutate(holiday = recode(holiday,
                      "N" = FALSE,
```

```
                        "Y" = TRUE))

#turn the school_day to binary
data$school_day <- as.factor(data$school_day)

data <- data %>%
    mutate(school_day = recode(school_day,
                        "N" = FALSE,
                        "Y" = TRUE))
```

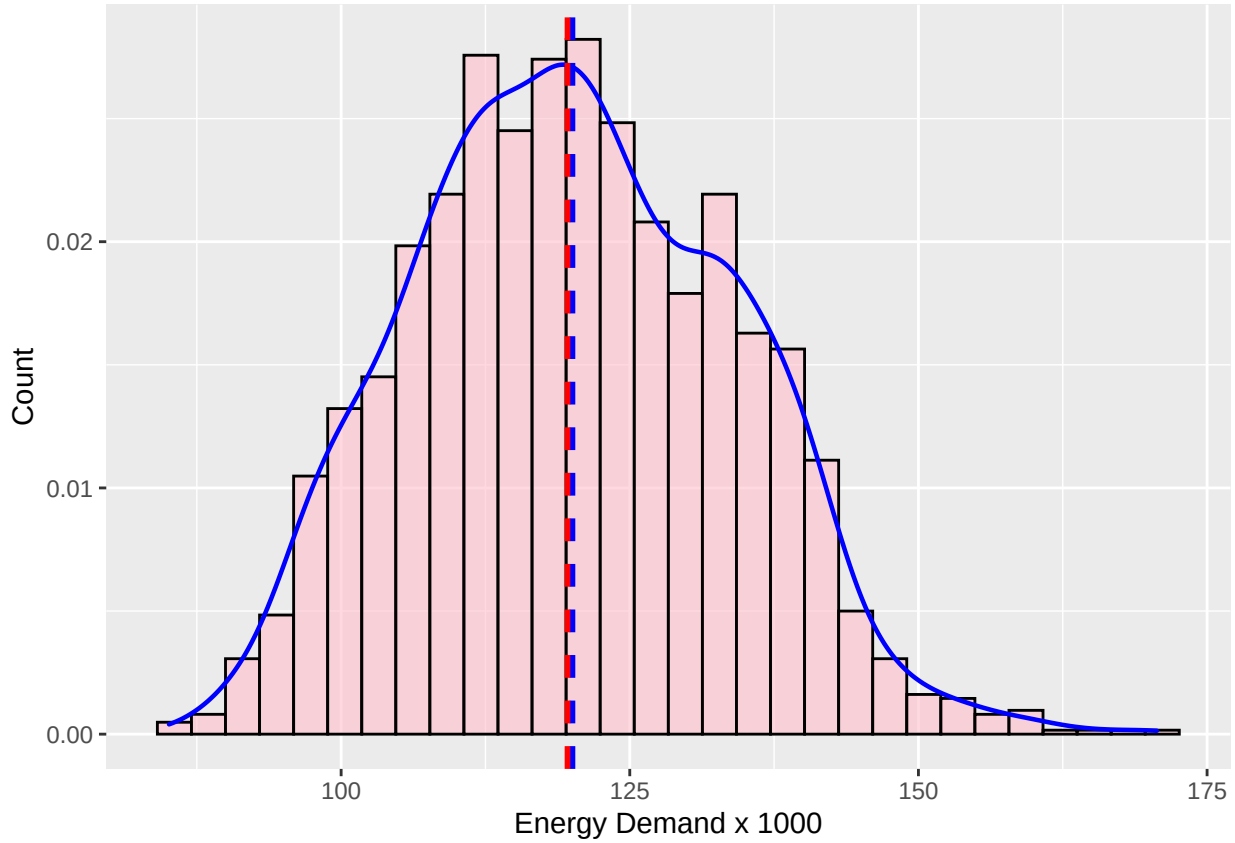
# 3 Exploratory Data Analysis (EDA)

In this section we will use some visual tools to discover hidden trends and patterns in the dataset that can help us build an accurate and efficient model to make predictions on daily electricity demand.

We start by doing some general analysis on demand, since this is what our model will be centered around.

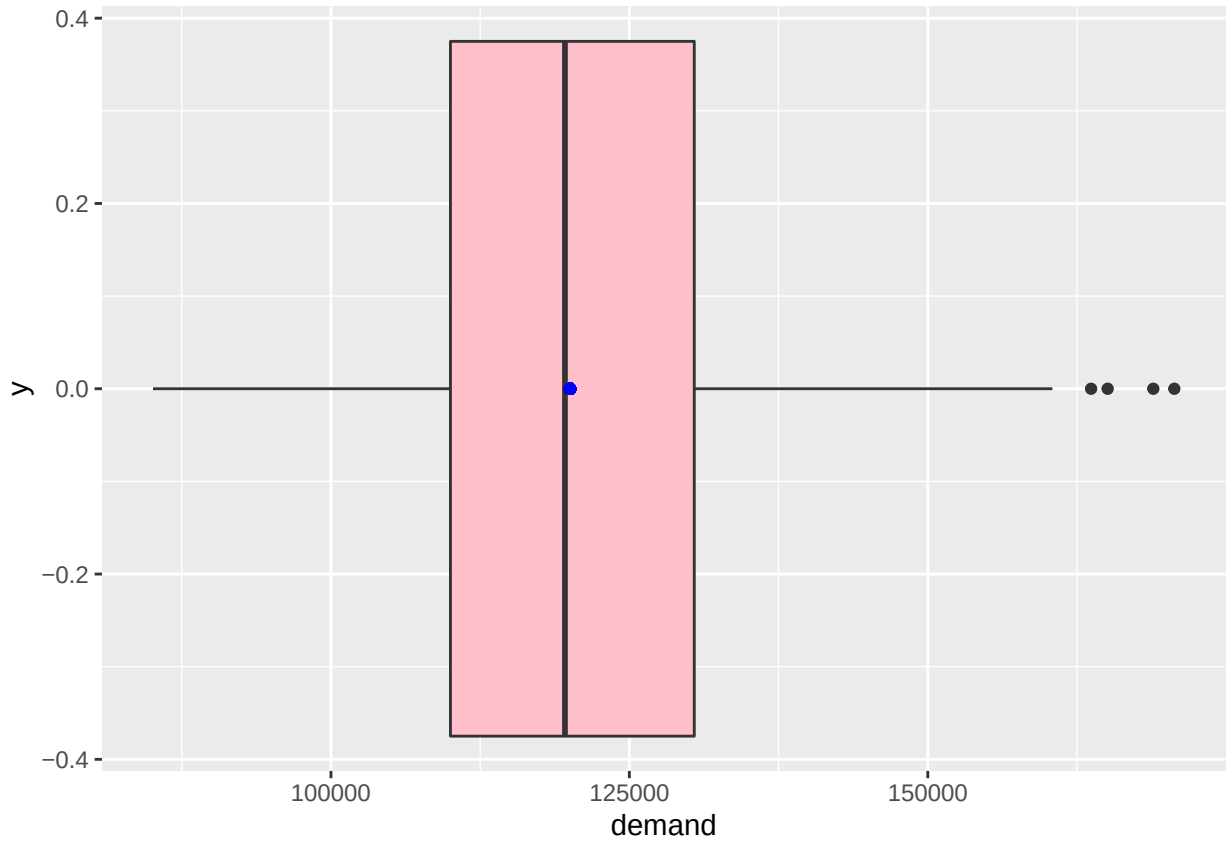
```
options(repr.plot.width=7, repr.plot.height=5)
ggplot(data, aes(x= demand/1000)) +
geom_histogram(aes(y =..density..), fill="pink", color="black", alpha=0.6) +
geom_density(color= "blue", size =0.8)+
geom_vline(aes(xintercept= mean(demand/1000)), color="blue",linetype="dashed", size=1) +
geom_vline(aes(xintercept= median(demand/1000)), color="red",linetype="dashed", size=1) +

labs(x ="Energy Demand x 1000", y="Count")
```

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




The histogram above shows that the distribution of the energy demand is almost symmetrical with the mean and median values both at around 120,000 Mwh.

```
deamnd_plot <- ggplot(data, aes(x = demand)) +
geom_boxplot(fill = "pink") +
geom_point(aes(x= mean(demand), y=0), color="blue")

deamnd_plot
```

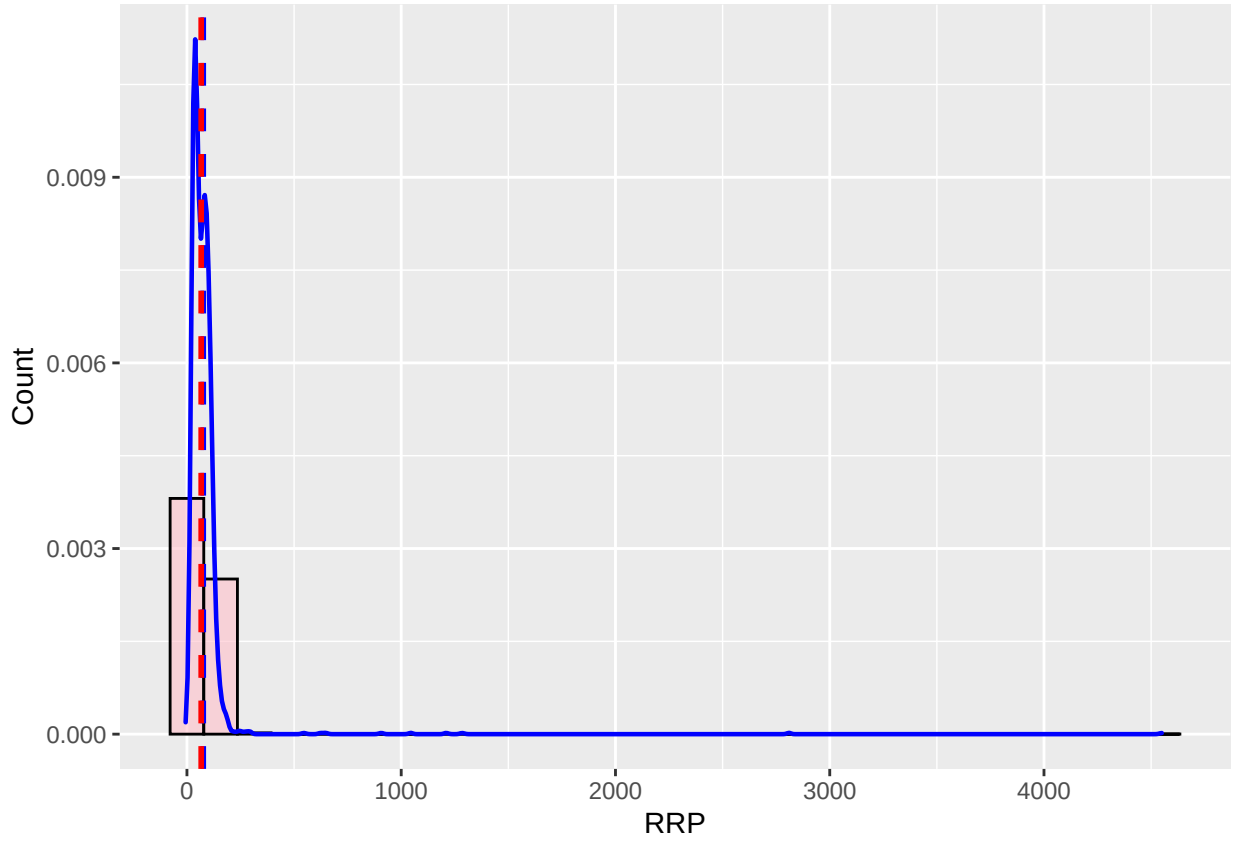



Furthermore, from the box-plot shown above, we can see that the energy demand has a longer tail on the right side, with all the outliers being there. Since the outliers are few and do not differ too much from the values at the end of the right tail, we keep them.

We move on to do some analysis on RRP.

```
options(repr.plot.width=7, repr.plot.height=5)
ggplot(data, aes(x= RRP)) +
geom_histogram(aes(y =..density..), fill="pink", color="black", alpha=0.6) +
geom_density(color= "blue", size =0.8)+
geom_vline(aes(xintercept= mean(RRP)), color="blue",linetype="dashed", size=1) +
geom_vline(aes(xintercept= median(RRP)), color="red",linetype="dashed", size=1) +

labs(x ="RRP", y="Count")
```

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



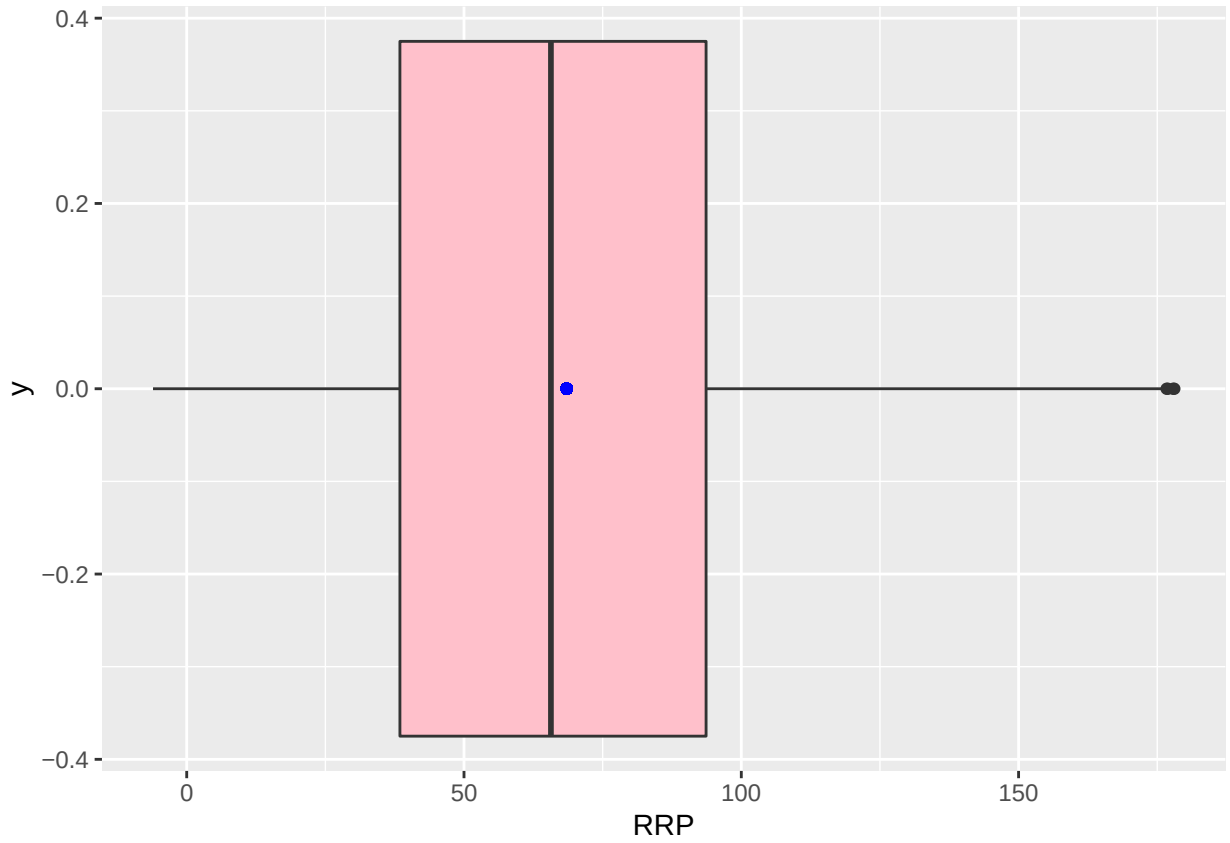

The above histogram of the RRP shows an extremely long right tail, especially when compared to the left tail. This indicates the presence of outliers, so we decide to further investigate this.

```
#OUTLIARS
Q <- quantile(data$RRP)
data_ol <- subset(data, data$RRP >= Q[4]+1.5*(Q[4]-Q[2]) | data$RRP<= Q[2]-1.5*(Q[4]-Q[2]))

dim(data_ol)
```

```
## [1] 28 15
```

Indeed, there are 28 outliers for the RRP variable. These values differ excessively from the rest, which indicates there may have been an error when collecting the data. Furthermore, since the values are so great they will have a big impact other variables, thus we decide to remove these outliers.

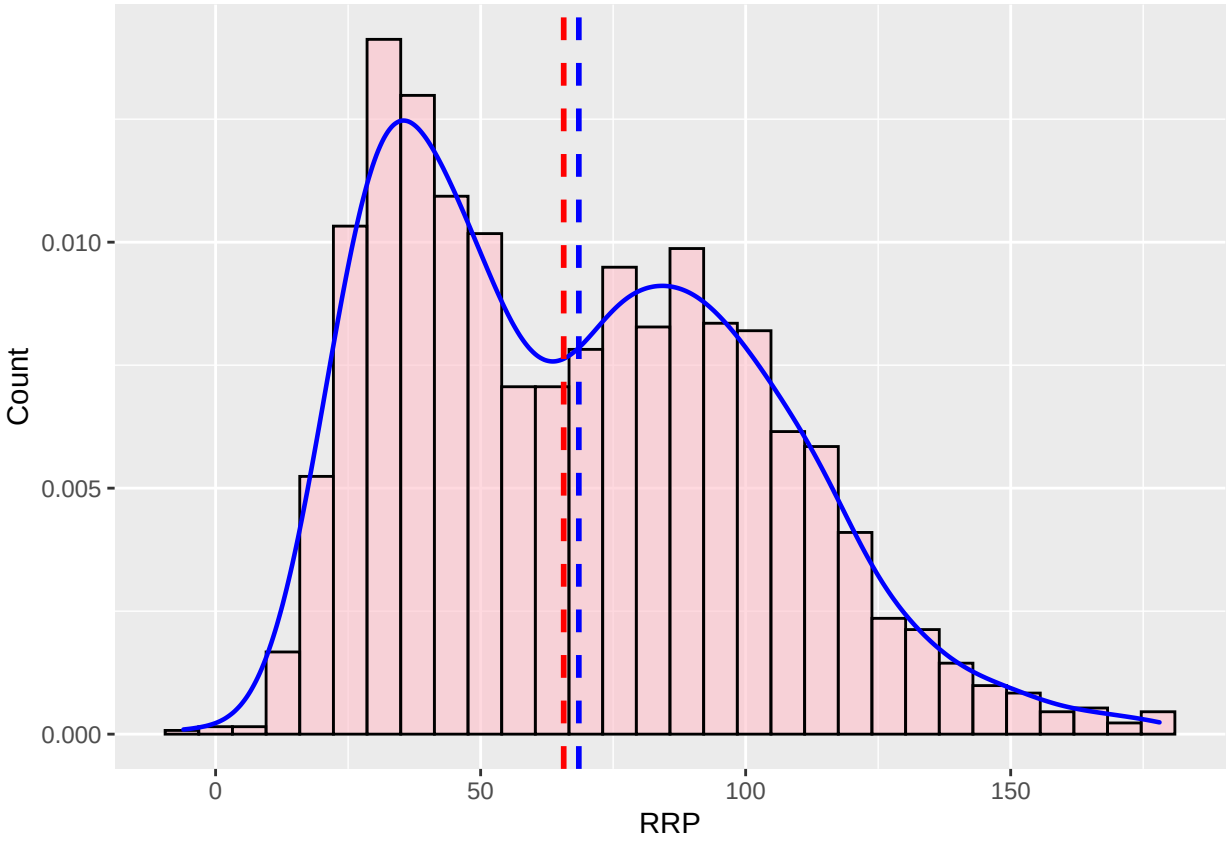
```
#WITHOUT
data <- subset(data, data$RRP <= Q[4]+1.5*(Q[4]-Q[2]) & data$RRP>= Q[2]-1.5*(Q[4]-Q[2]))

RRP_plot <- ggplot(data, aes(x = RRP)) +
geom_boxplot(fill = "pink") +
geom_point(aes(x= mean(RRP), y=0), color="blue")

RRP_plot
```




After removing the outliers, the RRP distribution is almost symmetrical, but the right tail is still longer, as seen in the box-plot above. We can derive much more information from this subset.

```
options(repr.plot.width=7, repr.plot.height=5)
ggplot(data, aes(x= RRP)) +
geom_histogram(aes(y =..density..), fill="pink", color="black", alpha=0.6) +
geom_density(color= "blue", size =0.8)+
geom_vline(aes(xintercept= mean(RRP)), color="blue",linetype="dashed", size=1) +
geom_vline(aes(xintercept= median(RRP)), color="red",linetype="dashed", size=1) +

labs(x ="RRP", y="Count")
```

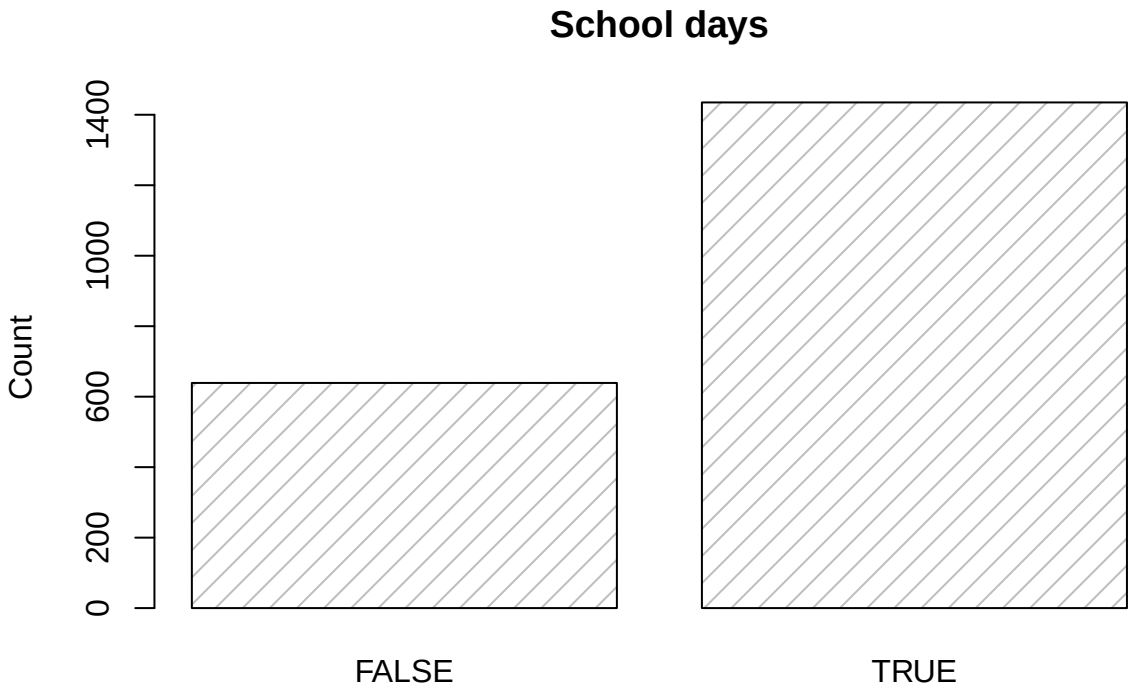
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




As for the histogram, without the outliers, RRP is shown to have a bimodal distribution, with the most frequent values in the RRP axis around 40 and 85. This histogram is much more informative than the previous one.

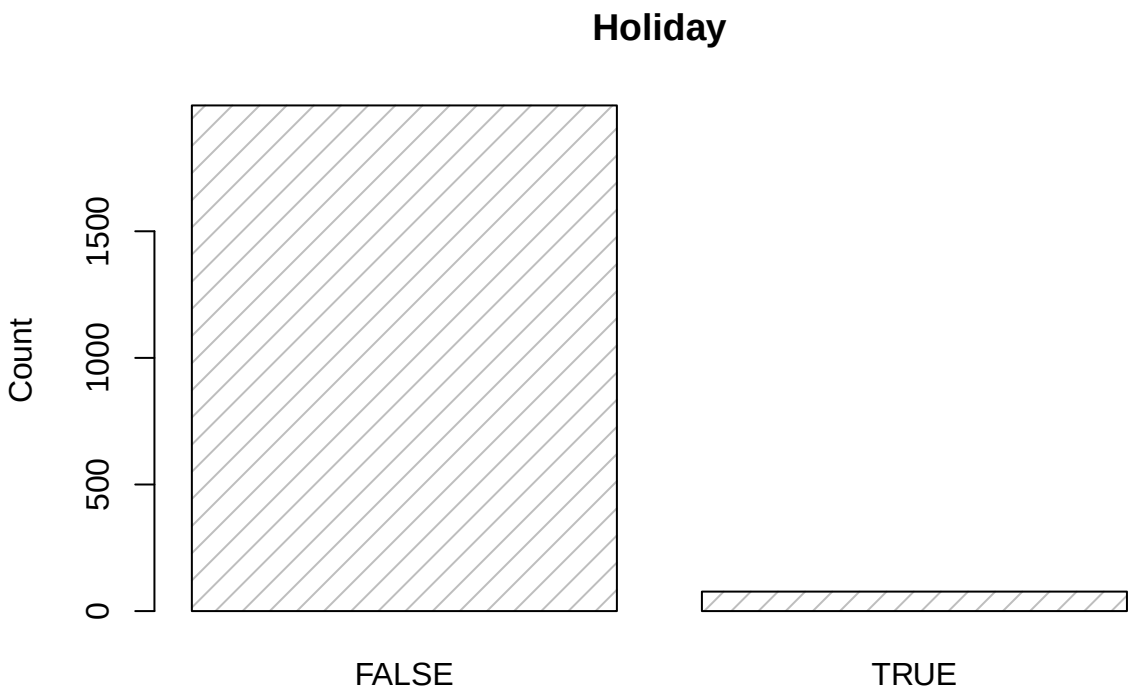
We proceed to analyse the relationship of some categorical variables with electricity demand.

```
bplot2 <- barplot(table(data$school_day),
            main="School days",
            ylab="Count",
            density=10
            )
```




In the graph above, we can see that the data is unbalanced.

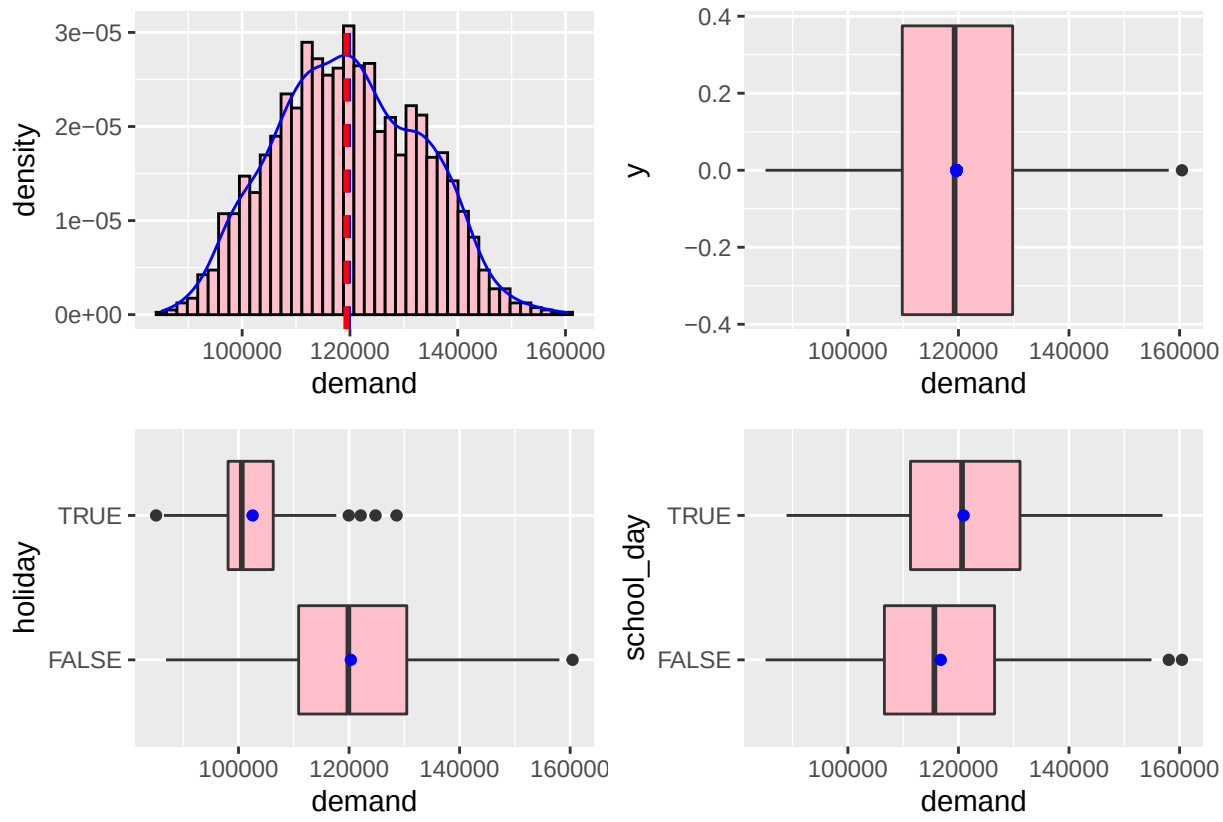
```
bplot1 <- barplot(table(data$holiday),
            main="Holiday",
            ylab="Count",
            density=10
            )
```




In the above case, the data is even more severely unbalanced than the school days.

```
p1 <- ggplot(data, aes(x=demand)) +
geom_histogram(aes(y=..density..),color="black", fill="pink", bins=40) +
geom_density(color="blue") +
geom_vline(aes(xintercept= mean(demand)), color="blue", linetype="dashed",size=1) +
geom_vline(aes(xintercept= median(demand)), color="red", linetype="dashed",size=1)


p2 <- ggplot(data, aes(x = demand)) +
geom_boxplot(fill = "pink") +
geom_point(aes(x= mean(demand), y=0), color="blue")


p3 <- ggplot(data, aes(x=holiday, y=demand, group=holiday)) +
  geom_boxplot(fill="pink") +
  stat_summary(fun=mean, geom="point", color="blue") +
  coord_flip()


p4 <- ggplot(data, aes(x=school_day, y=demand, group=school_day)) +
  geom_boxplot(fill="pink") +
  stat_summary(fun=mean, geom="point", color="blue") +
  coord_flip()
```

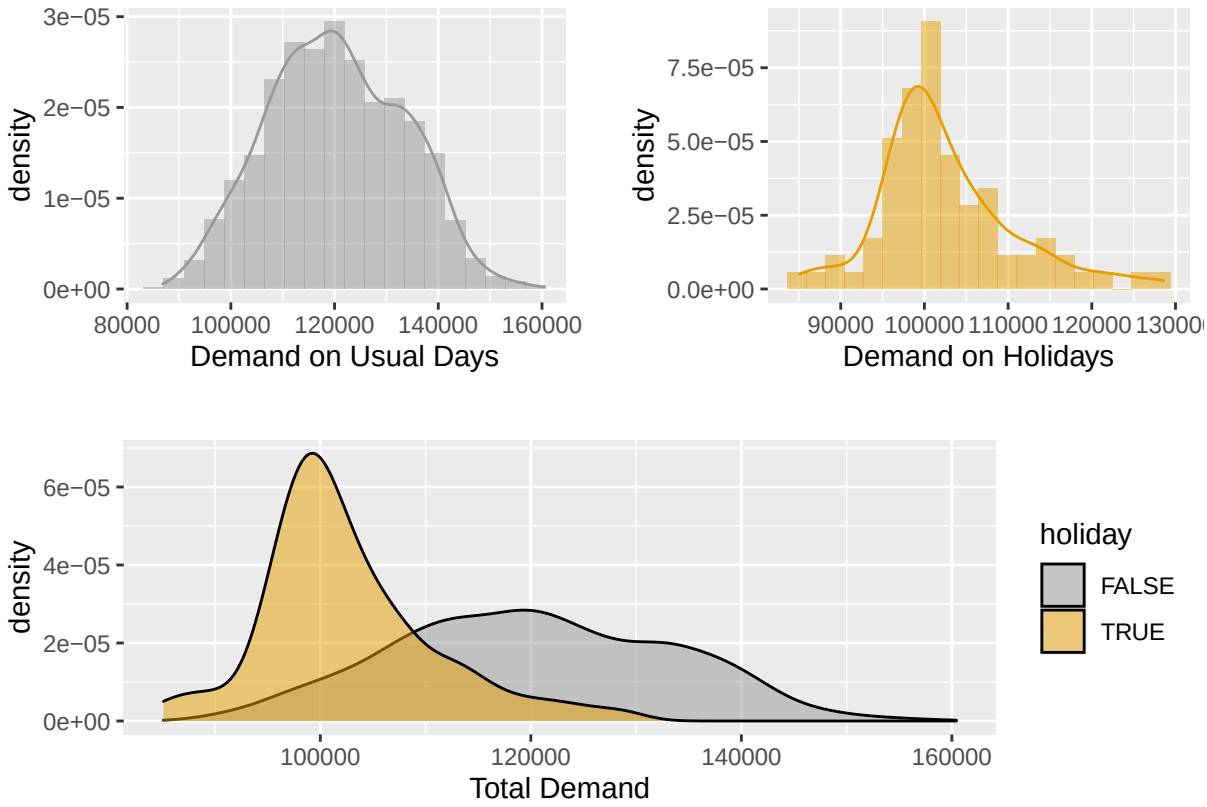
```
p1+p2 +p3+ p4
```




In the plots above, we can see that the distribution of electricity demand on its own is almost symmetrical. However, when plotted with relation to "holiday", the mean is much lower when the values are True, and also the variance is significantly smaller. As for demand plotted with respect to "school_day", demand is generally higher on school days.

```
options(repr.plot.width=15, repr.plot.height=5)
sd.colors <- c("#999999", "#E69F00")

p1 <- ggplot(data[data$holiday == FALSE,], aes(x=demand, y= ..density..))+
geom_histogram(fill=sd.colors[1],bins=20, alpha=.5)+
geom_density(color=sd.colors[1])+
labs(x = "Demand on Usual Days")

p2 <- ggplot(data[data$holiday == TRUE,], aes(x=demand, y= ..density..))+
geom_histogram(fill=sd.colors[2],bins=20, alpha=.5)+
geom_density(color=sd.colors[2])+
labs(x = "Demand on Holidays")

p3 <- ggplot(data, aes(x = demand, fill = holiday)) +
geom_density(alpha = 0.5) +
scale_fill_manual(values=sd.colors)+
labs(x = "Total Demand")
```

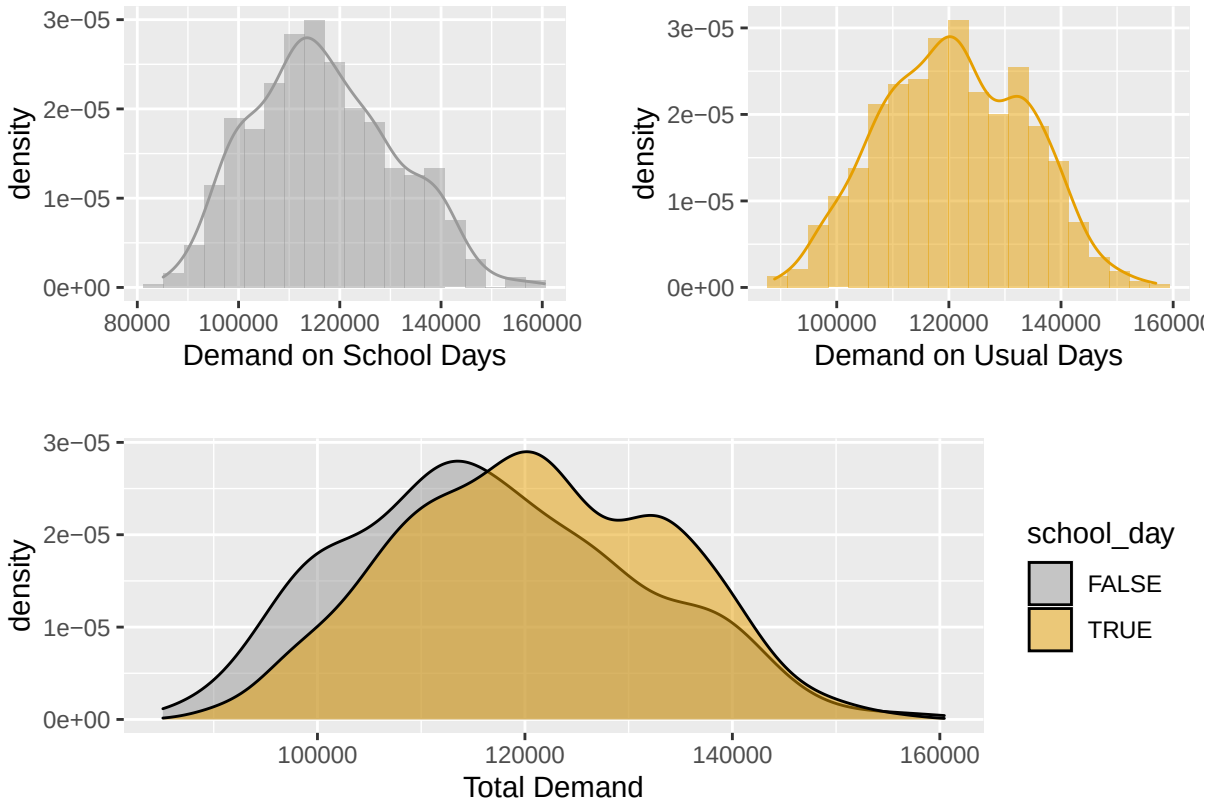
```
figure1 <- multi_panel_figure(columns = 2, rows = 2, panel_label_type = "none")
figure1 %<>%
  fill_panel(p1, column = 1, row = 1) %<>%
  fill_panel(p2, column = 2, row = 1) %<>%
  fill_panel(p3, column = 1:2, row = 2)
figure1
```




As seen in the density graphs above, the difference in distributions in demand for holidays and non-holidays is evident. Mean demand on holidays is much lower than on non-holidays, and the variance is also smaller. For non-holidays, daily demand values are less concentrated.

```
options(repr.plot.width=15, repr.plot.height=5)
sd.colors <- c("#999999", "#E69F00")

p1 <- ggplot(data[data$school_day == FALSE,], aes(x=demand, y= ..density..))+
geom_histogram(fill=sd.colors[1],bins=20, alpha=.5)+
geom_density(color=sd.colors[1])+
labs(x = "Demand on School Days")

p2 <- ggplot(data[data$school_day == TRUE,], aes(x=demand, y= ..density..))+
geom_histogram(fill=sd.colors[2],bins=20, alpha=.5)+
geom_density(color=sd.colors[2])+
labs(x = "Demand on Usual Days")
```

```
p3 <- ggplot(data, aes(x = demand , fill = school_day)) +
geom_density(alpha = 0.5) +
scale_fill_manual(values=sd.colors)+
labs(x = "Total Demand")


figure1 <- multi_panel_figure(columns = 2, rows = 2, panel_label_type = "none")

figure1 %<>%
  fill_panel(p1, column = 1, row = 1) %<>%
  fill_panel(p2, column = 2, row = 1) %<>%
  fill_panel(p3, column = 1:2, row = 2)
figure1
```



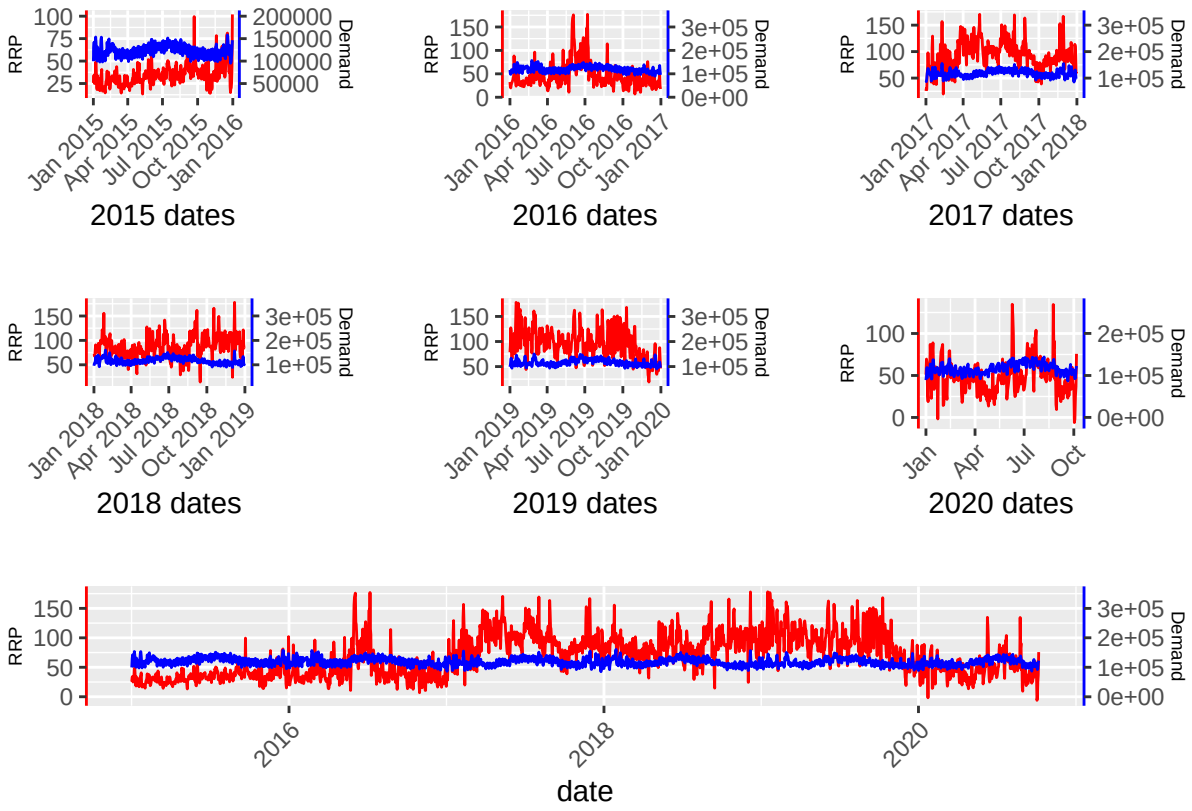

Regarding the school days feature, the differences are not as dramatic as with the holidays. Mean demand on school days is a little higher, but the variance does not differ by a lot.

Next, we visualize the evolution of RRP and demand throughout the years.

```
coeff <- 2000
df2015 <- subset(data, date < "2016-01-01 00:00:00")
df2016 <- subset(data, date < "2017-01-01 00:00:00" & date >= "2016-01-01 00:00:00")
df2017 <- subset(data, date < "2018-01-01 00:00:00" & date >= "2017-01-01 00:00:00")
df2018 <- subset(data, date < "2019-01-01 00:00:00" & date >= "2018-01-01 00:00:00")
df2019 <- subset(data, date < "2020-01-01 00:00:00" & date >= "2019-01-01 00:00:00")
df2020 <- subset(data, date < "2021-01-01 00:00:00" & date >= "2020-01-01 00:00:00")
```

```
p2015 <- ggplot(df2015, aes(x=date)) +
  geom_line( aes(y=RRP), color = 'red') +
  geom_line( aes(y=demand/coeff),color = 'blue') +
  scale_y_continuous(name = "RRP",sec.axis = sec_axis(~.*coeff,
                                                      name="Demand")) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.title.y = element_text(size = 7),
        axis.line.y.right = element_line(color = "blue"),
        axis.line.y.left = element_line(color = "red"))+
  labs(x = "2015 dates")
p2016 <- ggplot(df2016, aes(x=date)) +
  geom_line( aes(y=RRP), color = 'red') +
  geom_line( aes(y=demand/coeff),color = 'blue') +
  scale_y_continuous(name = "RRP",sec.axis = sec_axis(~.*coeff,
                                                      name="Demand")) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.title.y = element_text(size = 7),
        axis.line.y.right = element_line(color = "blue"),
        axis.line.y.left = element_line(color = "red"))+
  labs(x = "2016 dates")
p2017 <- ggplot(df2017, aes(x=date)) +
  geom_line( aes(y=RRP), color = 'red') +
  geom_line( aes(y=demand/coeff),color = 'blue') +
  scale_y_continuous(name = "RRP",sec.axis = sec_axis(~.*coeff,
                                                      name="Demand")) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.title.y = element_text(size = 7),
        axis.line.y.right = element_line(color = "blue"),
        axis.line.y.left = element_line(color = "red"))+
  labs(x = "2017 dates")
p2018 <- ggplot(df2018, aes(x=date)) +
  geom_line( aes(y=RRP), color = 'red') +
  geom_line( aes(y=demand/coeff),color = 'blue') +
  scale_y_continuous(name = "RRP",sec.axis = sec_axis(~.*coeff,
                                                      name="Demand")) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.title.y = element_text(size = 7),
        axis.line.y.right = element_line(color = "blue"),
        axis.line.y.left = element_line(color = "red"))+
  labs(x = "2018 dates")
p2019 <- ggplot(df2019, aes(x=date)) +
  geom_line( aes(y=RRP), color = 'red') +
  geom_line( aes(y=demand/coeff),color = 'blue') +
  scale_y_continuous(name = "RRP",sec.axis = sec_axis(~.*coeff,
                                                      name="Demand")) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.title.y = element_text(size = 7),
        axis.line.y.right = element_line(color = "blue"),
        axis.line.y.left = element_line(color = "red"))+
  labs(x = "2019 dates")
p2020 <- ggplot(df2020, aes(x=date)) +
  geom_line( aes(y=RRP), color = 'red') +
  geom_line( aes(y=demand/coeff),color = 'blue') +
```

```
  scale_y_continuous(name = "RRP",sec.axis = sec_axis(~.*coeff,
                                                      name="Demand")) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.title.y = element_text(size = 7),
        axis.line.y.right = element_line(color = "blue"),
        axis.line.y.left = element_line(color = "red"))+
  labs(x = "2020 dates")

pfull <- ggplot(data, aes(x=date)) +
  geom_line( aes(y=RRP), color = 'red') +
  geom_line( aes(y=demand/coeff),color = 'blue') +
  scale_y_continuous(name = "RRP",sec.axis = sec_axis(~.*coeff,
                                                      name="Demand")) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.line.y.right = element_line(color = "blue"),
        axis.line.y.left = element_line(color = "red"),
        axis.title.y = element_text(size = 7))



figure1 <- multi_panel_figure(columns = 3, rows = 3, panel_label_type = "none")
figure1 %<>%
  fill_panel(p2015, column = 1, row = 1) %<>%
  fill_panel(p2016, column = 2, row = 1) %<>%
  fill_panel(p2017, column = 3, row = 1) %<>%
  fill_panel(p2018, column = 1, row = 2) %<>%
  fill_panel(p2019, column = 2, row = 2) %<>%
  fill_panel(p2020, column = 3, row = 2) %<>%
  fill_panel(pfull, column = 1:3, row = 3)
figure1
```

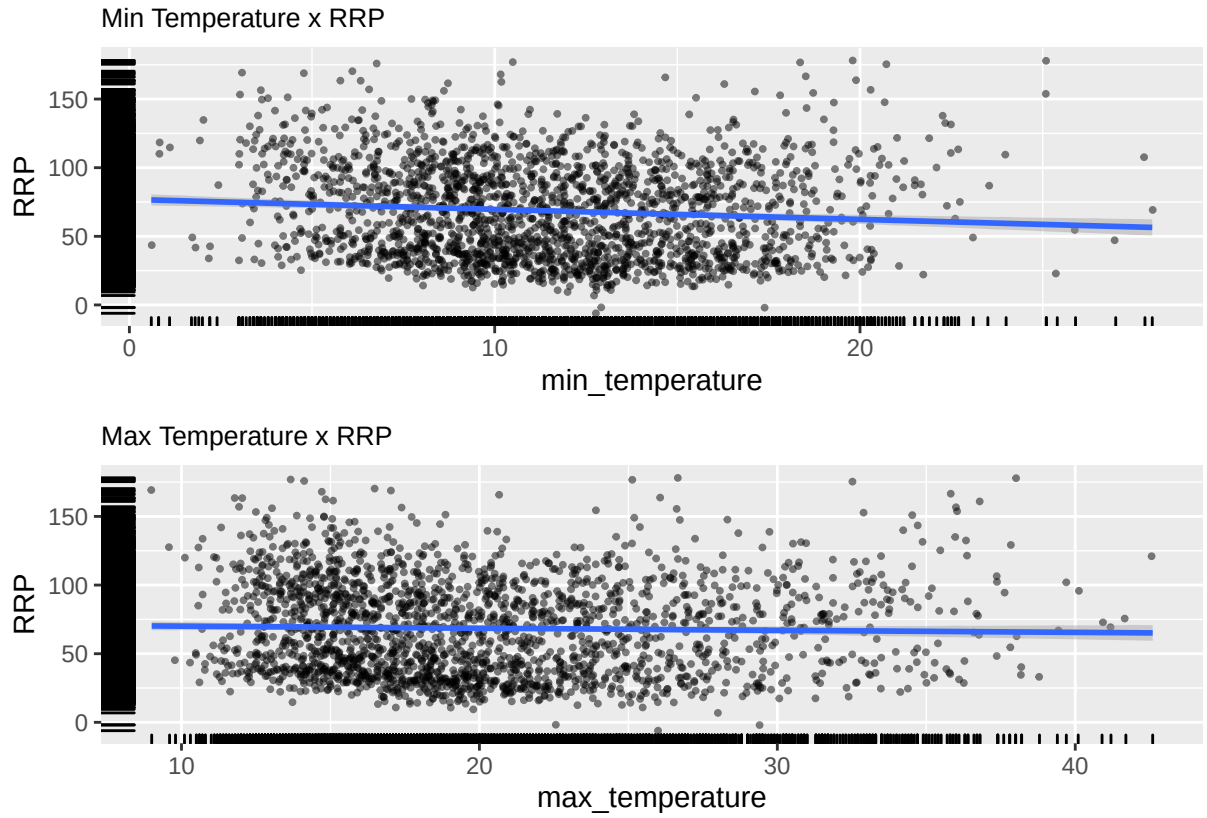



The above graphs show that while demand does not fluctuate much over the years, RRP does shift considerably during and throughout the years. There there was a notable peak in RRP around 2017 that lasted until almost 2020, when it went down again. Furthermore, it is shown that around the time Covid-19 related measures were put in place (March 2020), behavior of demand did not change. Although we are interested in further investigating this, the available dataset did not provide any useful insights to do it.

Next, we investigate the relations between temperature (min and max) and the other features, starting with RRP.

```
p1 <- ggplot(data, aes(x=min_temperature, y=RRP)) +
geom_jitter(alpha=0.5, size=0.7) +
scale_color_manual("mediumorchid1") +
geom_rug() +
geom_smooth(method=lm, formula=y~x) +
labs(title="Min Temperature x RRP") +
theme(plot.title = element_text(size=10))

p2 <- ggplot(data, aes(x=max_temperature, y=RRP)) +
geom_jitter(alpha=0.5, size=0.7) +
scale_color_manual("mediumorchid1") +
geom_rug() +
geom_smooth(method=lm, formula=y~x) +
labs(title="Max Temperature x RRP") +
theme(plot.title = element_text(size=10))


options(repr.plot.width=5, repr.plot.height=60)
```

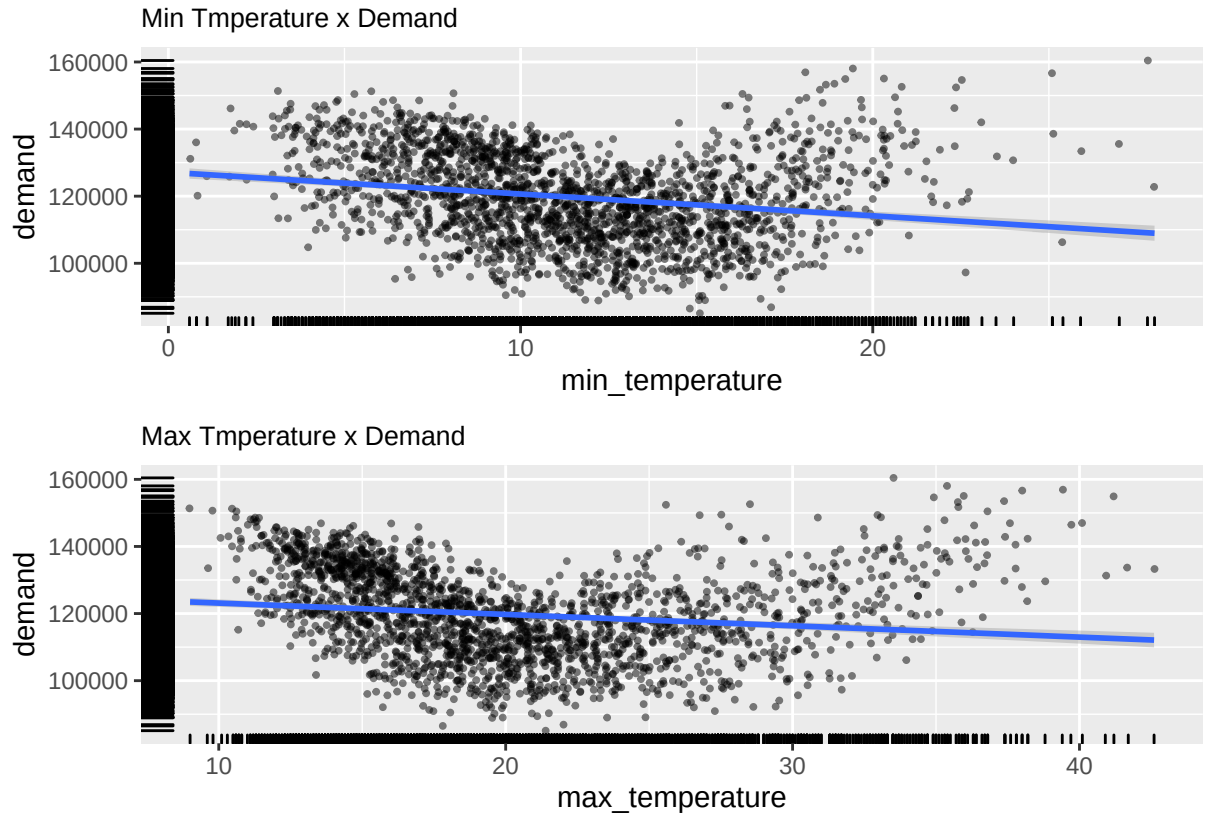
```
layout <- "A \n B"
p1+p2 + plot_layout(design = layout)
```




We can see that the relations of min and max temperature with RRP are extremely similar. Then, we move on to check the relations with demand.

```
p1 <- ggplot(data, aes(x=min_temperature, y=demand)) +
geom_jitter(alpha=0.5, size=0.7) +
scale_color_manual("mediumorchid1") +
geom_rug() +
geom_smooth(method=lm, formula=y~x) +
labs(title="Min Tmperature x Demand") +
theme(plot.title = element_text(size=10))

p2 <- ggplot(data, aes(x=max_temperature, y=demand)) +
geom_jitter(alpha=0.5, size=0.7) +
scale_color_manual("mediumorchid1") +
geom_rug() +
geom_smooth(method=lm, formula=y~x) +
labs(title="Max Tmperature x Demand") +
theme(plot.title = element_text(size=10))


options(repr.plot.width=5, repr.plot.height=60)
layout <- "A \n B "
p1+p2+p3 + plot_layout(design = layout)
```

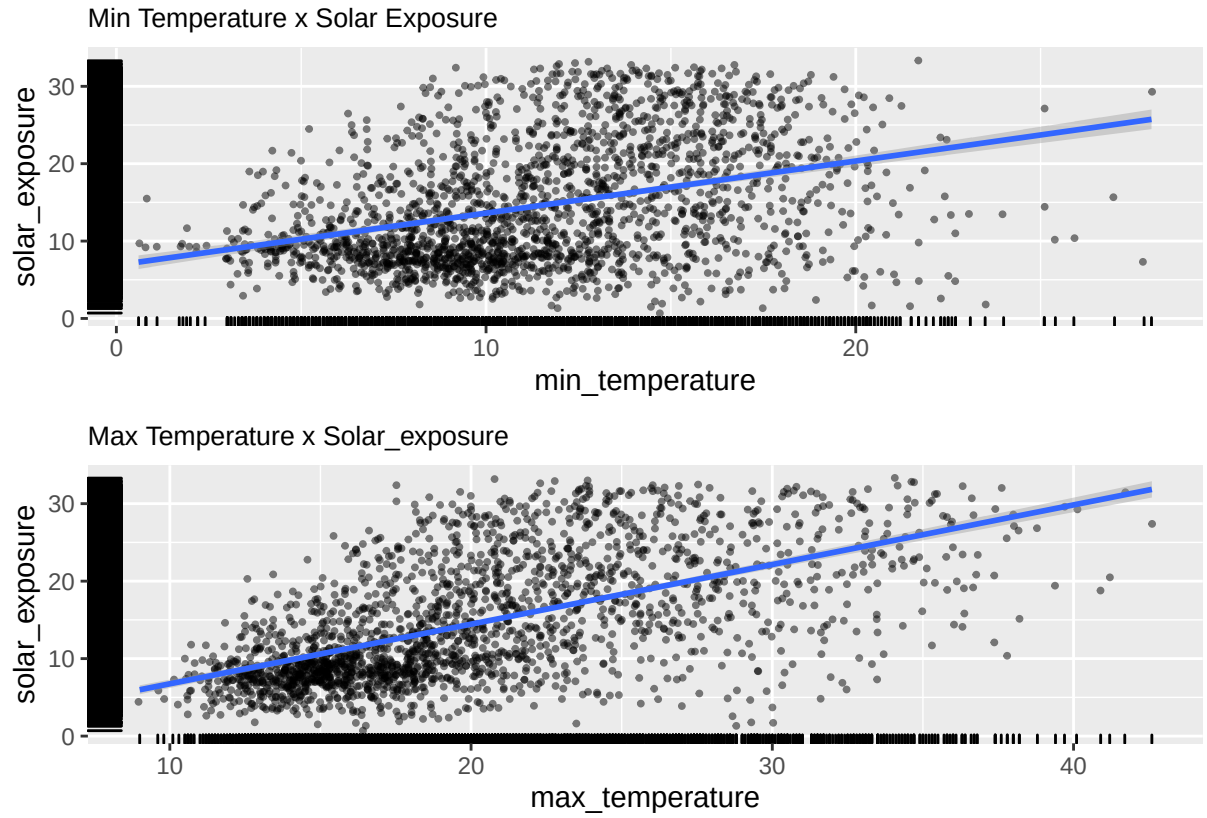
```
## Warning: Too few patch areas to hold all plots. Dropping plots
```




The similarity in relations here is a bit less than with RRP, but still existent. We proceed to check the relation between temperatures and solar exposure.

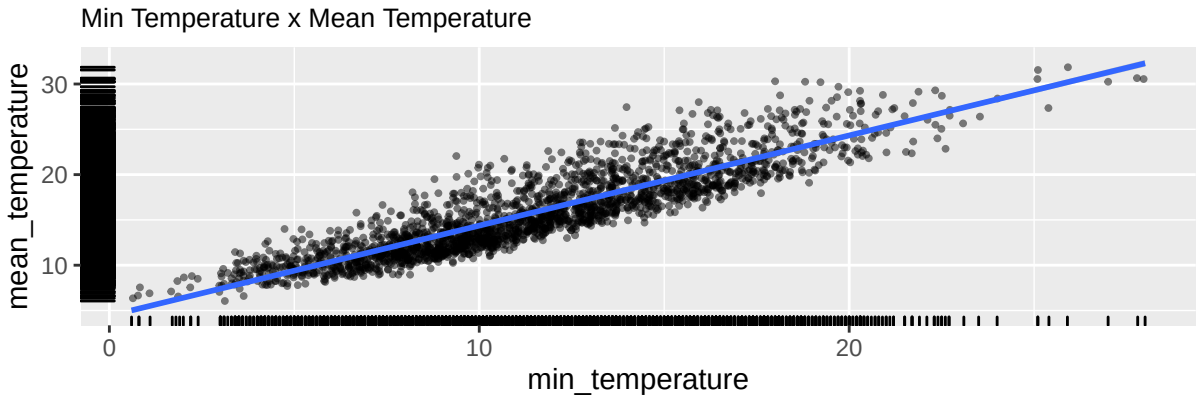
```
p1 <- ggplot(data, aes(x=min_temperature, y=solar_exposure)) +
geom_jitter(alpha=0.5, size=0.7) +
scale_color_manual("mediumorchid1") +
geom_rug() +
geom_smooth(method=lm, formula=y~x) +
labs(title="Min Temperature x Solar Exposure") +
theme(plot.title = element_text(size=10))

p2 <- ggplot(data, aes(x=max_temperature, y=solar_exposure)) +
geom_jitter(alpha=0.5, size=0.7) +
scale_color_manual("mediumorchid1") +
geom_rug() +
geom_smooth(method=lm, formula=y~x) +
labs(title="Max Temperature x Solar_exposure") +
theme(plot.title = element_text(size=10))


options(repr.plot.width=5, repr.plot.height=60)
```

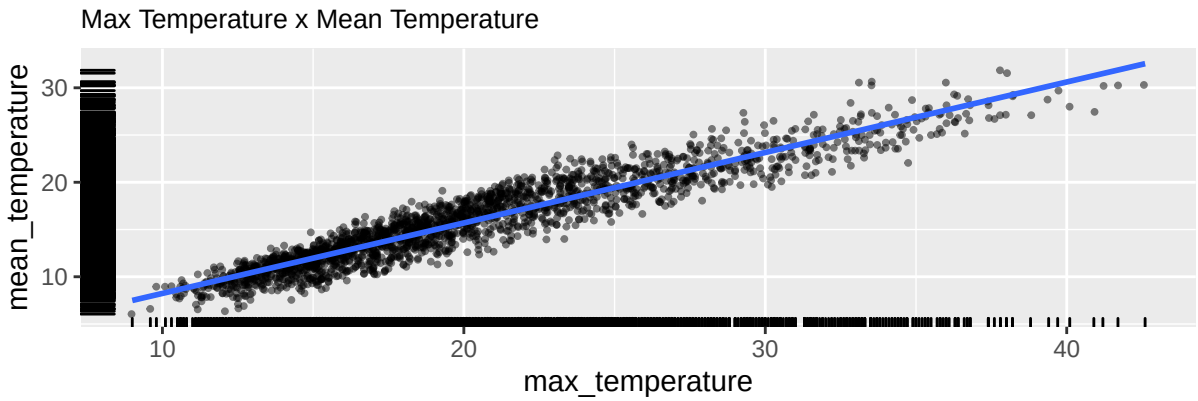
```
layout <- "A \n B"
p1+p2 + plot_layout(design = layout)
```




Here, they are extremely similar again. Finally, we check the relations between mean temperature with min and max temperature.

```
p1 <- ggplot(data, aes(x=min_temperature, y=mean_temperature)) +
geom_jitter(alpha=0.5, size=0.7) +
scale_color_manual("mediumorchid1") +
geom_rug() +
geom_smooth(method=lm, formula=y~x) +
labs(title="Min Temperature x Mean Temperature") +
theme(plot.title = element_text(size=10))

p2 <- ggplot(data, aes(x=max_temperature, y=mean_temperature)) +
geom_jitter(alpha=0.5, size=0.7) +
scale_color_manual("mediumorchid1") +
geom_rug() +
geom_smooth(method=lm, formula=y~x) +
labs(title="Max Temperature x Mean Temperature") +
theme(plot.title = element_text(size=10))


options(repr.plot.width=5, repr.plot.height=60)
layout <- "A \n B"
p1+p2 + plot_layout(design = layout)
```



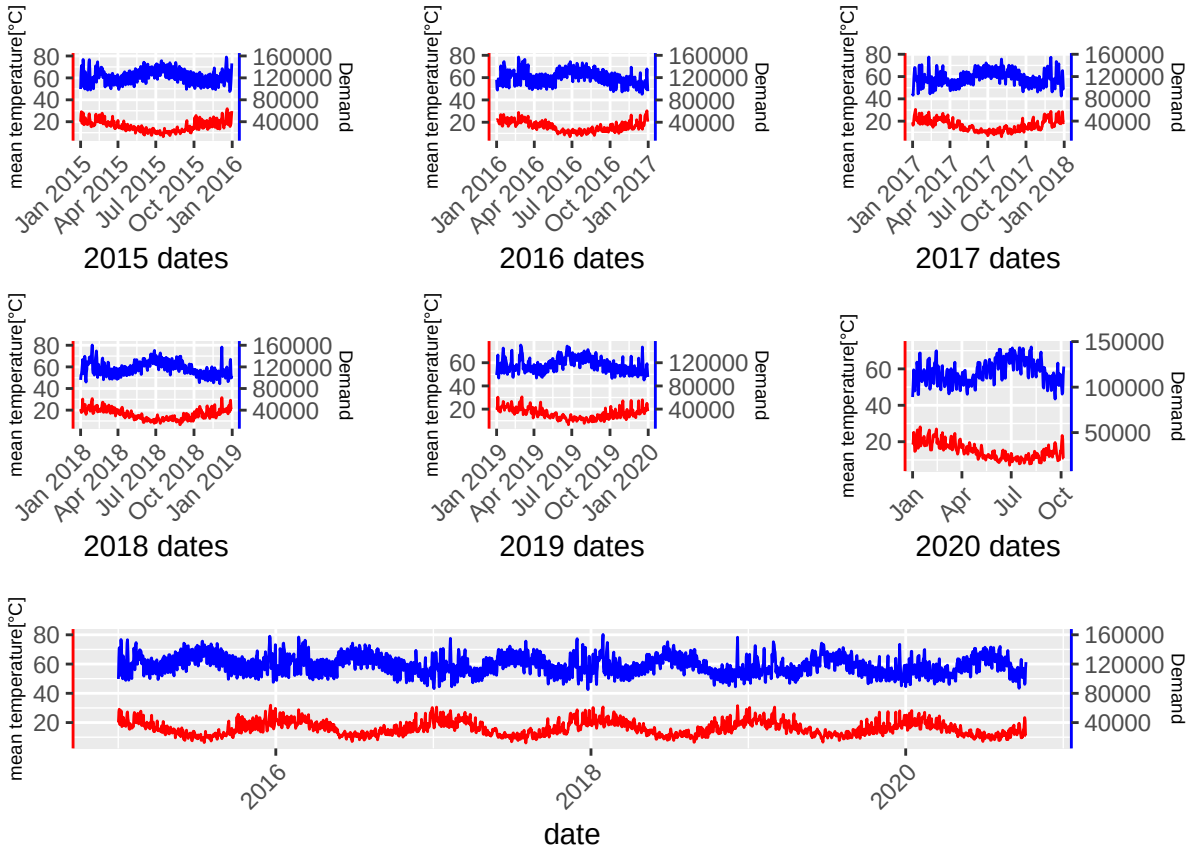

Again, the relations are very similar. From this analyses we conclude that the min and max temperature can be combined into one single mean temperature variable.

We move on to visualize the evolution of mean temperature and demand throughout the years.

```
p2015 <- ggplot(df2015, aes(x=date)) +
  geom_line( aes(y=mean_temperature), color = 'red') +
  geom_line( aes(y=demand/coeff),color = 'blue') +
  scale_y_continuous(name = "mean temperature[°C]",sec.axis = sec_axis(~.*coeff, name="Demand")) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.title.y = element_text(size = 7),
        axis.line.y.right = element_line(color = "blue"),
        axis.line.y.left = element_line(color = "red"))+
  labs(x = "2015 dates")
p2016 <- ggplot(df2016, aes(x=date)) +
  geom_line( aes(y=mean_temperature), color = 'red') +
  geom_line( aes(y=demand/coeff),color = 'blue') +
  scale_y_continuous(name = "mean temperature[°C]",sec.axis = sec_axis(~.*coeff, name="Demand")) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.title.y = element_text(size = 7),
        axis.line.y.right = element_line(color = "blue"),
        axis.line.y.left = element_line(color = "red"))+
  labs(x = "2016 dates")
p2017 <- ggplot(df2017, aes(x=date)) +
  geom_line( aes(y=mean_temperature), color = 'red') +
```

```
  geom_line( aes(y=demand/coeff),color = 'blue') +
  scale_y_continuous(name = "mean temperature[°C]",sec.axis = sec_axis(~.*coeff, name="Demand")) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.title.y = element_text(size = 7),
        axis.line.y.right = element_line(color = "blue"),
        axis.line.y.left = element_line(color = "red"))+
  labs(x = "2017 dates")
p2018 <- ggplot(df2018, aes(x=date)) +
  geom_line( aes(y=mean_temperature), color = 'red') +
  geom_line( aes(y=demand/coeff),color = 'blue') +
  scale_y_continuous(name = "mean temperature[°C]",sec.axis = sec_axis(~.*coeff, name="Demand")) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.title.y = element_text(size = 7),
        axis.line.y.right = element_line(color = "blue"),
        axis.line.y.left = element_line(color = "red"))+
  labs(x = "2018 dates")
p2019 <- ggplot(df2019, aes(x=date)) +
  geom_line( aes(y=mean_temperature), color = 'red') +
  geom_line( aes(y=demand/coeff),color = 'blue') +
  scale_y_continuous(name = "mean temperature[°C]",sec.axis = sec_axis(~.*coeff, name="Demand")) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.title.y = element_text(size = 7),
        axis.line.y.right = element_line(color = "blue"),
        axis.line.y.left = element_line(color = "red"))+
  labs(x = "2019 dates")
p2020 <- ggplot(df2020, aes(x=date)) +
  geom_line( aes(y=mean_temperature), color = 'red') +
  geom_line( aes(y=demand/coeff),color = 'blue') +
  scale_y_continuous(name = "mean temperature[°C]",sec.axis = sec_axis(~.*coeff, name="Demand")) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.title.y = element_text(size = 7),
        axis.line.y.right = element_line(color = "blue"),
        axis.line.y.left = element_line(color = "red"))+
  labs(x = "2020 dates")

pfull <- ggplot(data, aes(x=date)) +
  geom_line( aes(y=mean_temperature), color = 'red') +
  geom_line( aes(y=demand/coeff),color = 'blue') +
  scale_y_continuous(name = "mean temperature[°C]",sec.axis = sec_axis(~.*coeff, name="Demand")) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.title.y = element_text(size = 7),
        axis.line.y.right = element_line(color = "blue"),
        axis.line.y.left = element_line(color = "red"))


figure1 <- multi_panel_figure(columns = 3, rows = 3, panel_label_type = "none")
figure1 %<>%
  fill_panel(p2015, column = 1, row = 1) %<>%
  fill_panel(p2016, column = 2, row = 1) %<>%
  fill_panel(p2017, column = 3, row = 1) %<>%
  fill_panel(p2018, column = 1, row = 2) %<>%
  fill_panel(p2019, column = 2, row = 2) %<>%
  fill_panel(p2020, column = 3, row = 2) %<>%
```

```
  fill_panel(pfull, column = 1:3, row = 3)
figure1
```




From the above graph, it is clear that when mean temperature peaks, demand goes down and viceversa. This indicates that more electricity is used when the weather is colder.

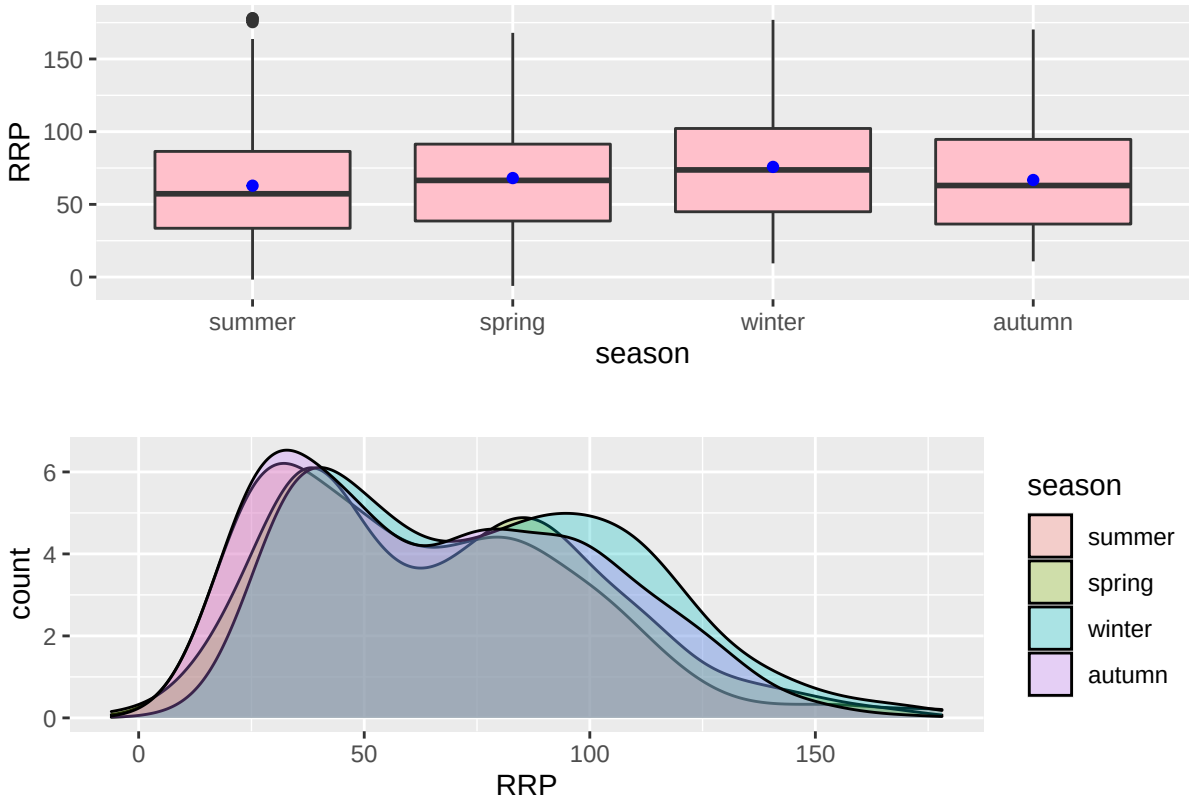
We think that the behavior of demand may change depending on the seasons. To investigate this, we proceed to separate the data on seasons (summer, autumn, winter, spring) as categorical values.

```
data$month<-as.numeric(format(data$date,"%m"))
data$season <- "summer"
data$season[data$month>2 & data$month<6] <- "autumn"
data$season[data$month>5 & data$month<9] <- "winter"
data$season[data$month>8 & data$month<12] <- "spring"
data$season<-factor(data$season,levels=c("summer","spring","winter","autumn"))
summary(data$season)
```

```
## summer spring winter autumn
##    490    487    548    549
```

```
options(repr.plot.width=15, repr.plot.height=10)
p1 <- ggplot(data, aes(x=season, y=RRP, group=season)) +
  geom_boxplot(fill="pink") +
  stat_summary(fun=mean, geom="point", color="blue")
```

```
p2 <- ggplot(data, aes(x=RRP, y=..count.., fill=season))+
geom_density(alpha=0.3)

figure1 <- multi_panel_figure(columns = 1, rows = 2, panel_label_type = "none")
figure1 %<>%
  fill_panel(p1, column = 1, row = 1) %<>%
  fill_panel(p2, column = 1, row = 2)
figure1
```

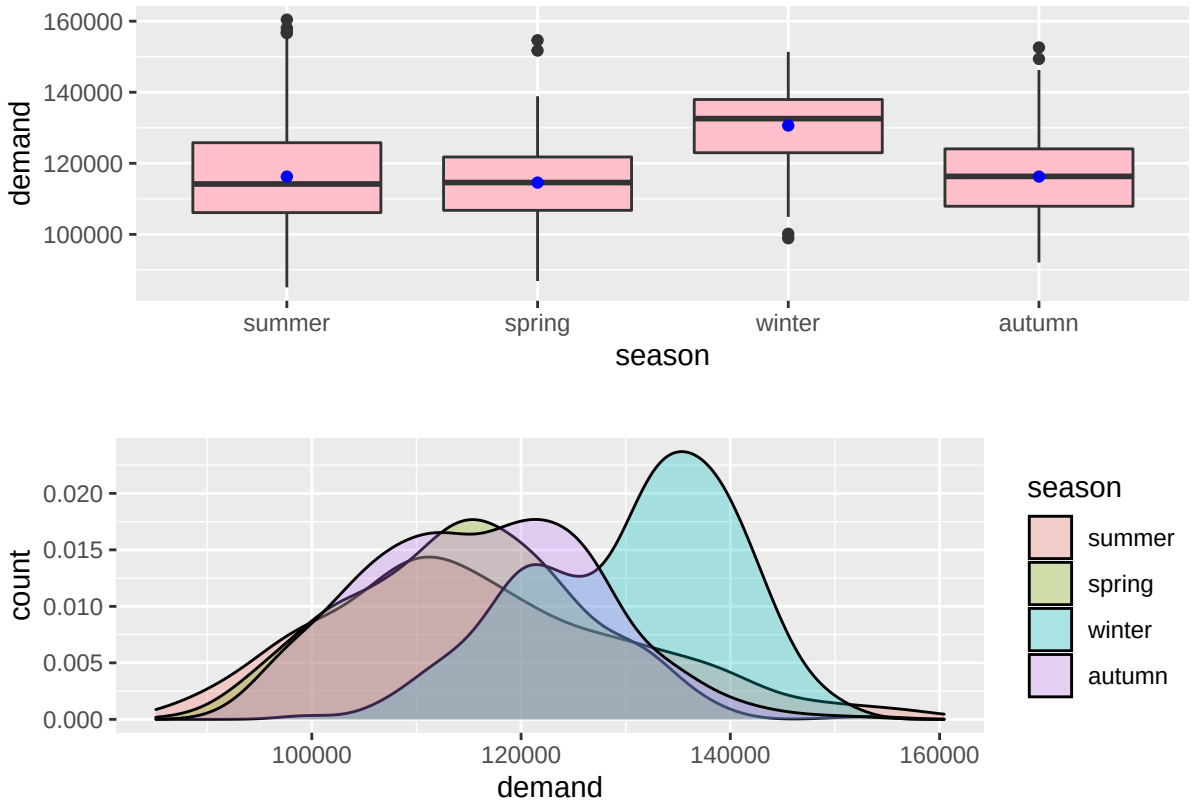



From the above graphs, we can see that RRP values are the highest in winter and the lowest in summer. As for spring and autumn, values don't differ a lot. Variance in all seasons is similar.

We continue to investigate the effect of the seasons, this time in the electricity demand.

```
options(repr.plot.width=15, repr.plot.height=10)
p1 <- ggplot(data, aes(x=season, y=demand, group=season)) +
  geom_boxplot(fill="pink") +
  stat_summary(fun=mean, geom="point", color="blue")

p2 <- ggplot(data, aes(x=demand, y=..count.., fill=season))+
geom_density(alpha=0.3)

figure1 <- multi_panel_figure(columns = 1, rows = 2, panel_label_type = "none")
figure1 %<>%
  fill_panel(p1, column = 1, row = 1) %<>%
  fill_panel(p2, column = 1, row = 2)
```

```
figure1
```

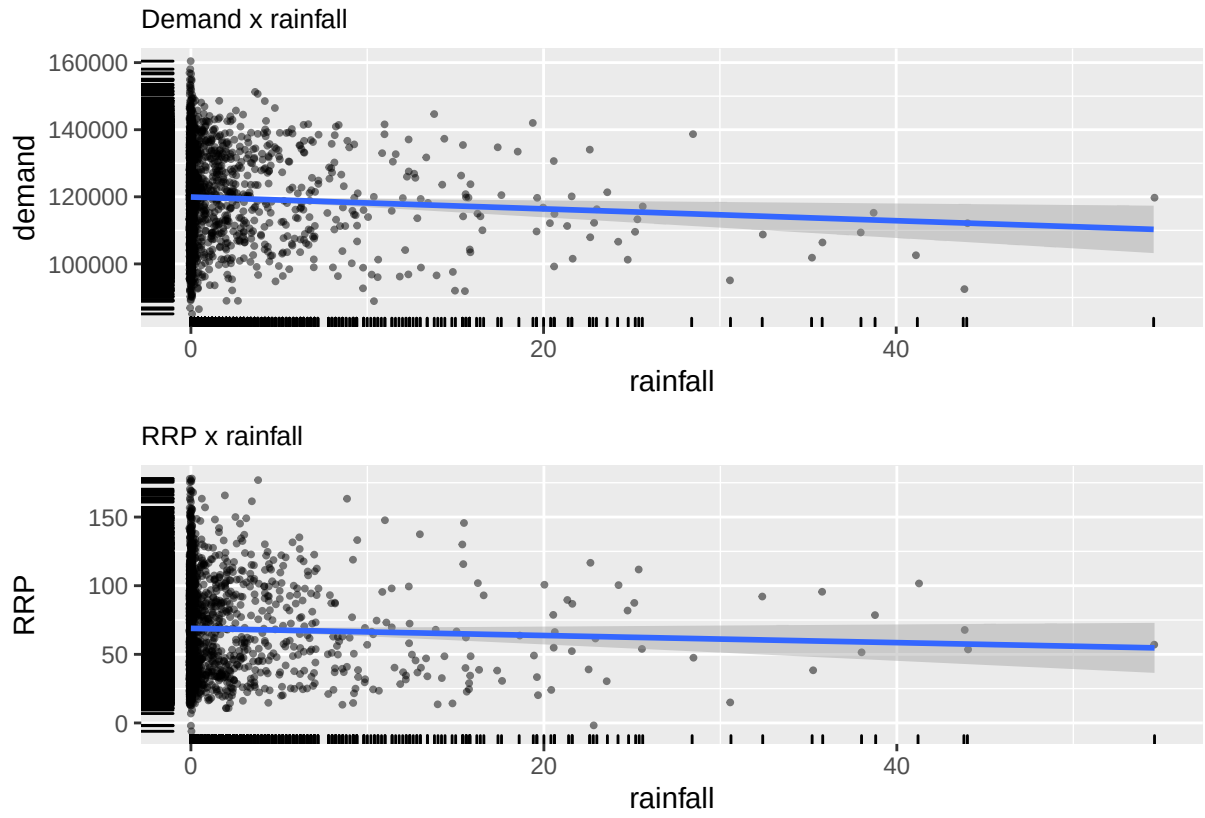



In the graphs shown above, it is clear that the demand on winter is higher than in the other seasons. Summer has the lowest demand comparatively, but it does not differ much from the spring and autumn ones. Variance of demand during summer is the highest. Based on these insights, we conclude that using seasons as a categorical value provides useful information, and continue to do so.

We move on to investigate the effect of rainfall in both electricity demand and RRP.

```
p1 <- ggplot(data, aes(x=rainfall, y=demand)) +
geom_jitter(alpha=0.5, size=0.7) +
scale_color_manual("mediumorchid1") +
geom_rug() +
geom_smooth(method=lm, formula=y~x) +
labs(title="Demand x rainfall") +
theme(plot.title = element_text(size=10))

p2 <- ggplot(data, aes(x=rainfall, y=RRP)) +
geom_jitter(alpha=0.5, size=0.7) +
scale_color_manual("mediumorchid1") +
geom_rug() +
geom_smooth(method=lm, formula=y~x) +
labs(title="RRP x rainfall") +
theme(plot.title = element_text(size=10))

options(repr.plot.width=5, repr.plot.height=60)
```

```
layout <- "A \n B"
p1+p2+plot_layout(design = layout)
```

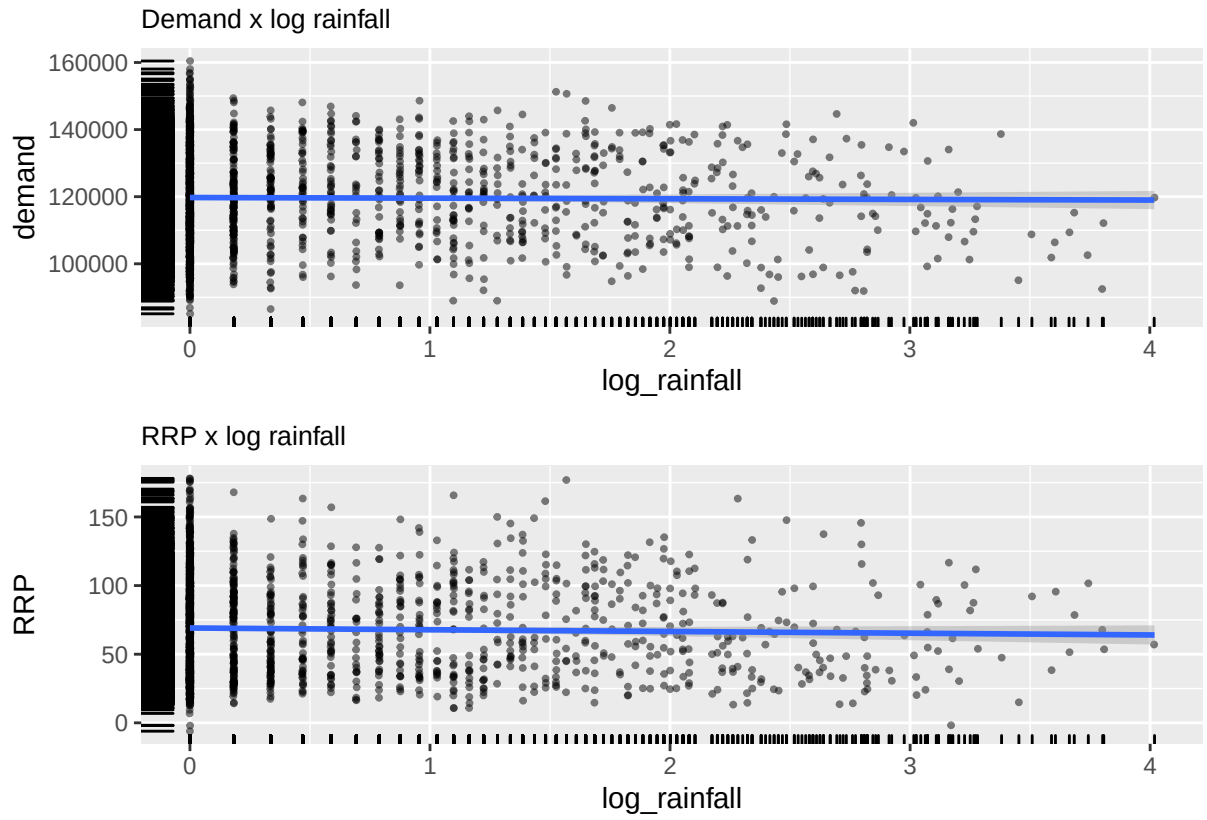



The above graphs indicate that as rainfall increases, demand and RRP decrease, although not much. The shape of the graph also indicates that doing a transformation will be beneficial, so we proceed to investigate the behavior of the logarithm of rainfall and add 1, to avoid errors when the rainfall is 0.

```
data$log_rainfall <- log(data$rainfall+1)
```

```
p1 <- ggplot(data, aes(x=log_rainfall, y=demand)) +
geom_jitter(alpha=0.5, size=0.7) +
scale_color_manual("mediumorchid1") +
geom_rug() +
geom_smooth(method=lm, formula=y~x) +
labs(title="Demand x log rainfall") +
theme(plot.title = element_text(size=10))

p2 <- ggplot(data, aes(x=log_rainfall, y=RRP)) +
geom_jitter(alpha=0.5, size=0.7) +
scale_color_manual("mediumorchid1") +
geom_rug() +
geom_smooth(method=lm, formula=y~x) +
labs(title="RRP x log rainfall") +
theme(plot.title = element_text(size=10))
```

```
options(repr.plot.width=5, repr.plot.height=60)
layout <- "A \n B"
p1+p2+plot_layout(design = layout)
```

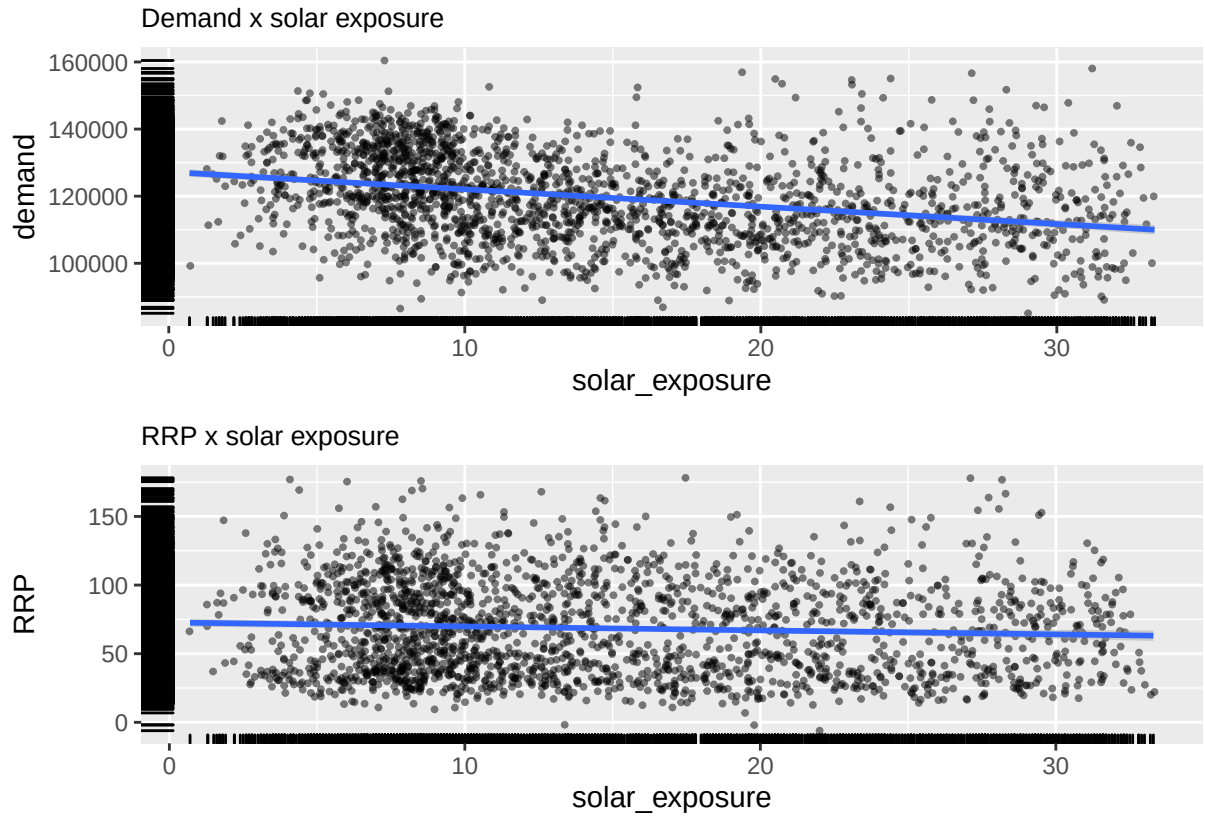



From the above graphs, we can see the after the transformation of rainfall, the effect in demand and RRP is still not significant, especially for demand. RRP decreases very slightly as log rainfall increases.

Next, we plot some graphs to see the effect of solar exposure on demand.

```
p1 <- ggplot(data, aes(x=solar_exposure, y=demand)) +
geom_jitter(alpha=0.5, size=0.7) +
scale_color_manual("mediumorchid1") +
geom_rug() +
geom_smooth(method=lm, formula=y~x) +
labs(title="Demand x solar exposure") +
theme(plot.title = element_text(size=10))

p2 <- ggplot(data, aes(x=solar_exposure, y=RRP)) +
geom_jitter(alpha=0.5, size=0.7) +
scale_color_manual("mediumorchid1") +
geom_rug() +
geom_smooth(method=lm, formula=y~x) +
labs(title="RRP x solar exposure") +
theme(plot.title = element_text(size=10))

options(repr.plot.width=5, repr.plot.height=60)
```

```
layout <- "A \n B"
p1+p2+plot_layout(design = layout)
```

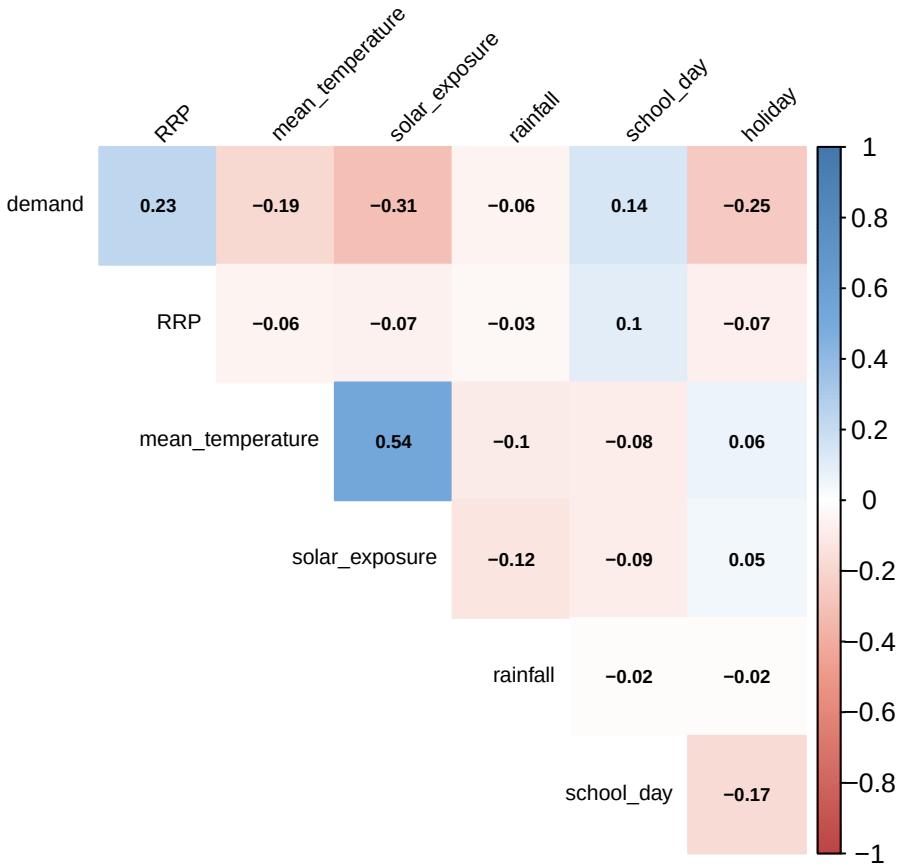



As seen above, as solar exposure increases demand decreases. As for RRP, the effect of solar exposure is almost imperceptible.

Next, we check the correlation matrix in order to find potentially significant relations between variables.

```
data_copy <- copy(data)
options(repr.plot.width=7, repr.plot.height=7)
data_copy$holiday= as.numeric(data_copy$holiday)
data_copy$school_day = as.numeric(data_copy$school_day)
data_copy$date = as.numeric(data_copy$date)

numerical <- subset(data_copy, select = c(demand,
                                          RRP,
                                          mean_temperature,
                                          solar_exposure,
                                          rainfall,
                                          school_day,
                                          holiday
                                          ))

correlation <- cor(numerical)

col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))
corrplot(correlation, method="color", col=col(200),
```

```
        type="upper",
        addCoef.col = "black",
        tl.col="black", tl.srt=45,
        sig.level = 0.01, insig = "blank",
        diag=FALSE,tl.cex=0.7, number.cex=0.6)
```




This correlation plot suggests that we start by analyzing the relation between demand and solar_exposure, and demand and holiday. The feature that has less correlation with demand is rainfall.

## 3.1 Variable statistics

In this part, we will test some hypotheses to further understand the effect of different variables on demand. From the previously performed EDA, we see that it is possible that the categorical variables have an effect on the electricity demand. In this section, we wish further investigate this.

### 3.1.1 Hypothesis 1

The equality of means of the demand on school days and non-school days is going to be examined. For this, we want to perform a t-test, and before checking the equality of the means, we have to check the equality of the variances. To do so, an F-test is going to be carried out first.

$\sigma_{school}$ is the standard deviation of demand for school days and $\sigma_{notschool}$ is the standard deviation for the other days.

- $H_0$: $\sigma_{school} = \sigma_{notschool}$

- $H_1$: $\sigma_{school} \neq \sigma_{notschool}$

```
school_days <- data[data$school_day == TRUE,]$demand
no_school_days <- data[data$school_day == FALSE,]$demand

var.test(school_days, no_school_days , alternative="two.sided")
```

```
##
##  F test to compare two variances
##
## data:  school_days and no_school_days
## F = 0.84801, num df = 1434, denom df = 638, p-value = 0.01305
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.7418882 0.9660391
## sample estimates:
## ratio of variances
##          0.8480142
```

We cannot perform a t-test to analyze the equality of the mean, since we do not have evidence to conclude that the variances of demand on school day vs not school days are equal. This is not surprising, given that as seen in the EDA section, the data is very unbalanced.

### 3.1.2 Hypothesis 2

We want to examine the equality of means of the demand on holidays and non-holidays. As before, we have to check the equality of the variances before carrying out the t-test.

$\sigma_{holiday}$ is the standard deviation of demand for holidays and $\sigma_{notholiday}$ is the standard deviation for the other days.

- $H_0$: $\sigma_{holiday} = \sigma_{notholiday}$
- $H_1$: $\sigma_{holiday} \neq \sigma_{notholiday}$

```
holidays <- data[data$holiday == TRUE,]$demand
no_holidays <- data[data$holiday == FALSE,]$demand

var.test(holidays, no_holidays , alternative="two.sided")
```

```
##
##  F test to compare two variances
##
## data:  holidays and no_holidays
## F = 0.38706, num df = 76, denom df = 1996, p-value = 7.539e-07
## alternative hypothesis: true ratio of variances is not equal to 1
## 95 percent confidence interval:
##  0.2863989 0.5497053
## sample estimates:
## ratio of variances
##          0.3870597
```

Again, we cannot conclude that the variances of the two populations are equal and thus we cannot perform a t-test. Once more, this is not surprising, since in this case the data is even more severely unbalanced as previously seen on the EDA phase.

# 4 Model Data & Analysis

## 4.1 Linear Regression Model

### 4.1.1 Numerical variables

In this section we will consider the effect of numerical variables on electricity demand. As we know from the correlation matrix performed during the EDA, the solar exposure (solar_exposure) has high correlation with demand. So, we start by building a linear regression model with only this variable and continue by adding one variable at a time. Performance in this section will be analyzed in terms of BIC ((Bayesian Information Criterion) and Adjusted $R^2$.

We start by building a model with only the solar exposure variable.

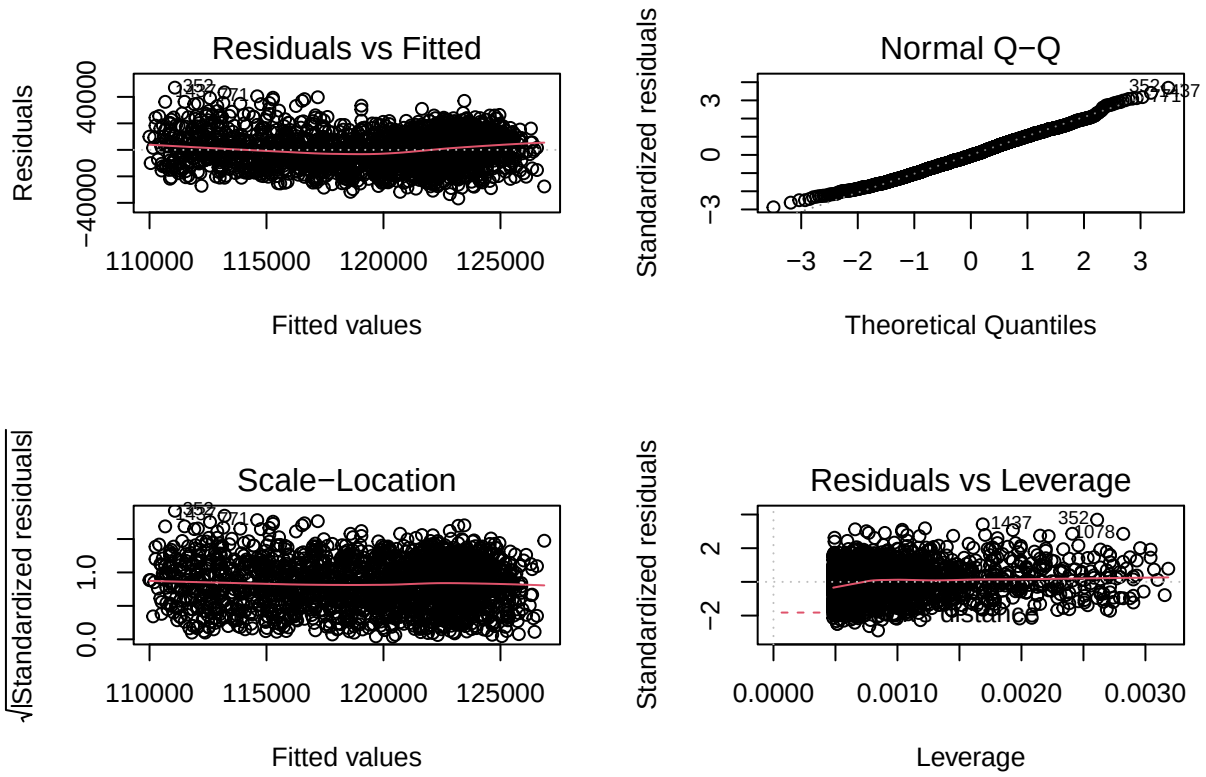
```
mod.num0 <- lm(data=data, demand ~ solar_exposure)
summary(mod.num0)
```

```
##
## Call:
## lm(formula = demand ~ solar_exposure, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -36690  -8868   -460   8980  46968
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    127234.81     589.12   216.0   <2e-16 ***
## solar_exposure   -517.63      35.45   -14.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12730 on 2072 degrees of freedom
## Multiple R-squared:  0.0933, Adjusted R-squared:  0.09287
## F-statistic: 213.2 on 1 and 2072 DF,  p-value: < 2.2e-16
```

```
BIC(mod.num0)
```

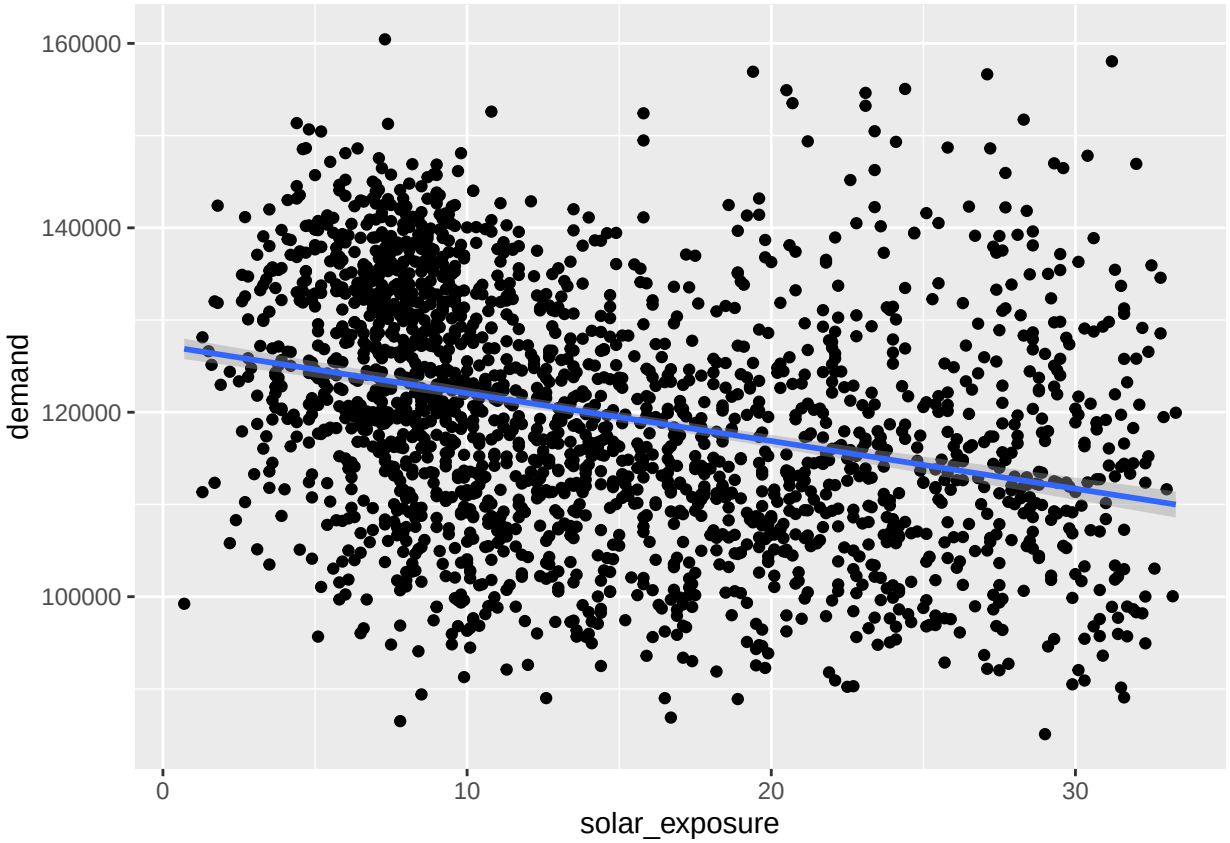
```
## [1] 45113.74
```

```
# Diagnostic
par(mfrow=c(2,2))
plot(mod.num0)
```




The behavior of residuals seen above are admissible. We proceed to check the shape of this model.

```
ggplot(data, aes(x = solar_exposure, y = demand)) +
geom_point() +
stat_smooth(method = "lm", formula = y ~ x) +
labs(x = 'solar_exposure', y = 'demand')
```

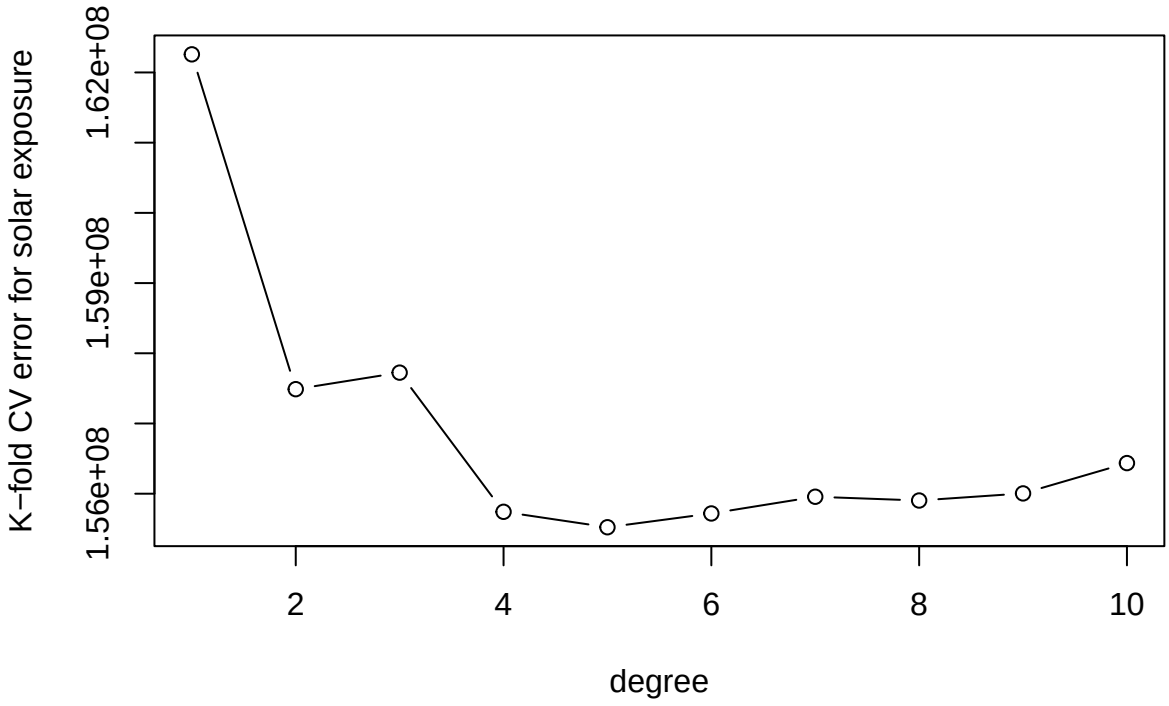



From the above plot, we can see that our model may benefit by increasing the polynomial degree. After comparing different degree models, we find that the best performance is obtained by the fourth degree one, as confirmed by performing the following cross validation.

```
#polynomial for solar exposure
set.seed(1)
cv.error.10.ex <- rep(0,10)
for (i in 1:10){
  glm.fit <- glm(demand~poly(solar_exposure,i),data=data)
  cv.error.10.ex[i] <- cv.glm(data,glm.fit,K=10)$delta[1]
}
cv.error.10.ex
```

```
##  [1] 162257738 157489052 157724480 155742166 155522398 155718510 155956707
##  [8] 155903209 156004406 156436097
```

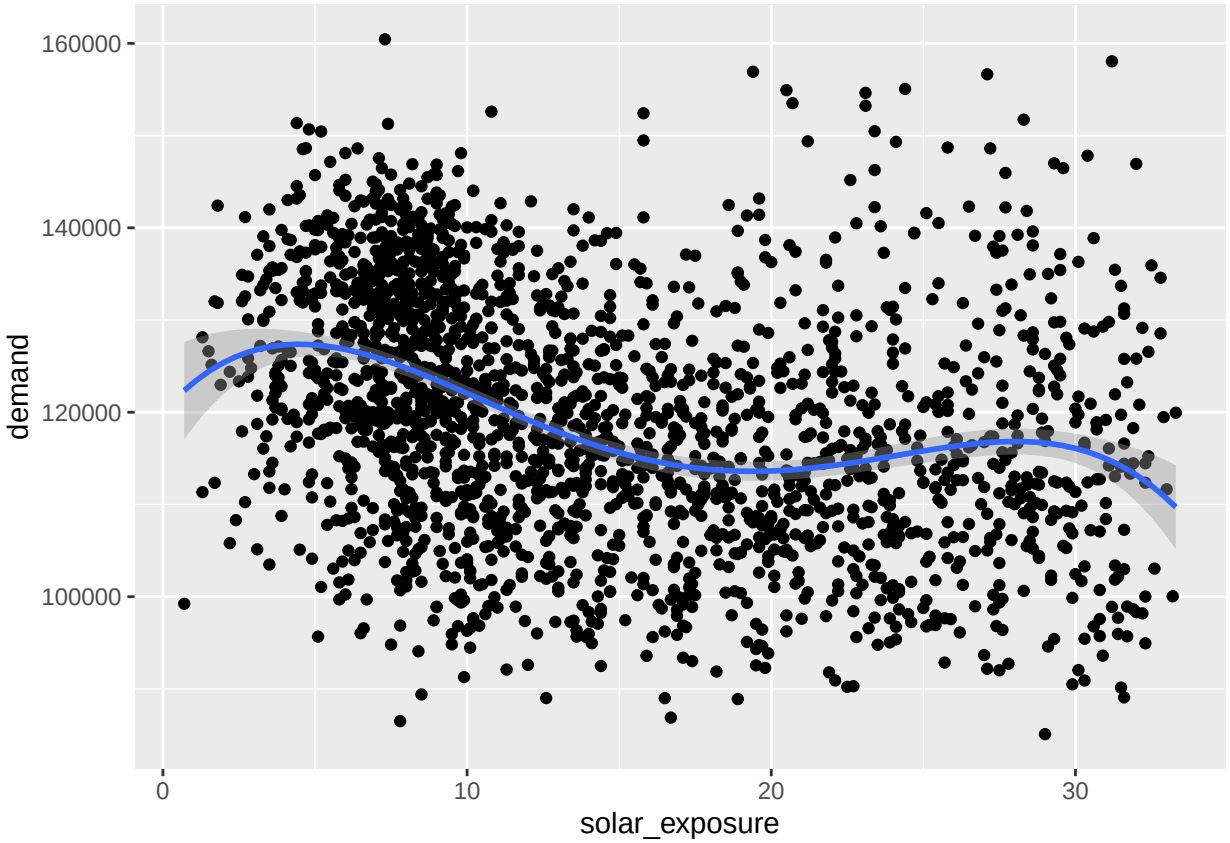
```
expo.temp <- plot(1:10, cv.error.10.ex, type="b", xlab="degree", ylab="K-fold CV error for solar exposu
```




```
expo.temp
```

```
## NULL
```

```
ggplot(data, aes(x = solar_exposure, y = demand)) +
geom_point() +
stat_smooth(method = "lm", formula = y ~ poly(x, 4)) +
labs(x = 'solar_exposure', y = 'demand')
```



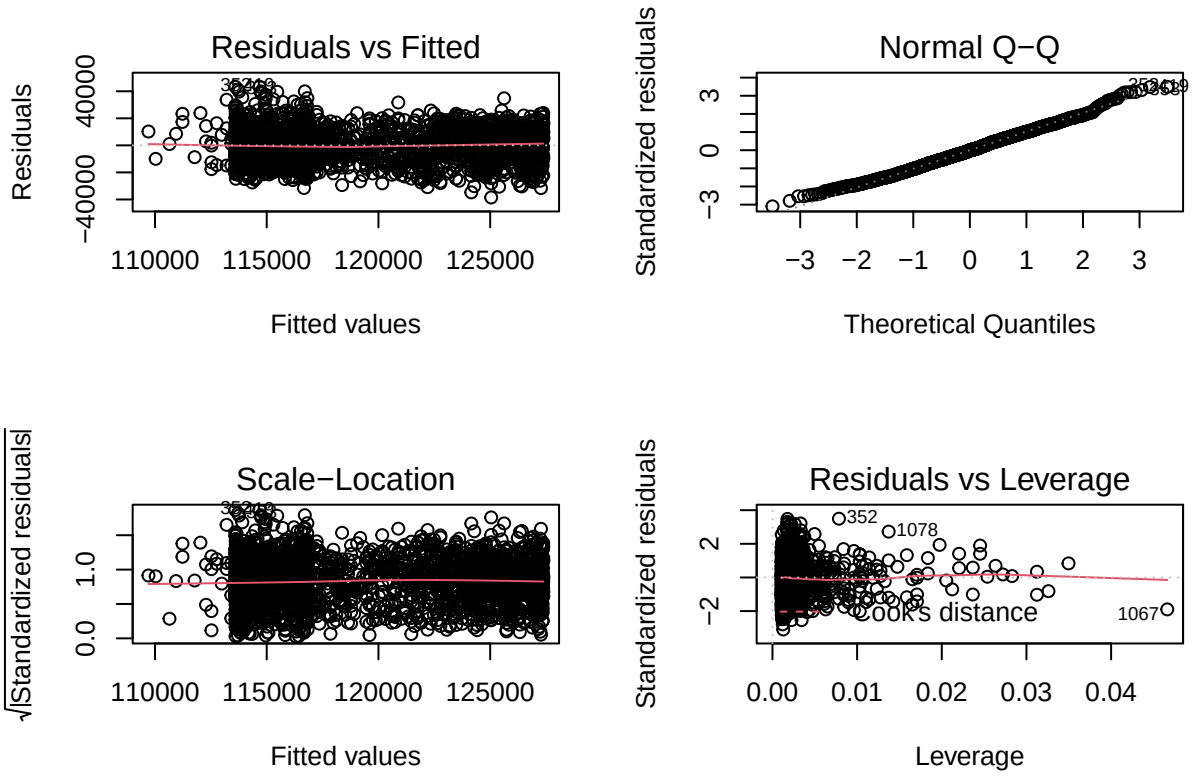

```
mod.num1 <- lm(data=data, demand ~ poly(solar_exposure, 4))
summary(mod.num1)
```

```
##
## Call:
## lm(formula = demand ~ poly(solar_exposure, 4), data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -38538  -8620   -172   8603  43377
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)                119663.2      273.7 437.234  < 2e-16 ***
## poly(solar_exposure, 4)1 -185943.6    12463.8 -14.919  < 2e-16 ***
## poly(solar_exposure, 4)2   99653.8    12463.8   7.995 2.13e-15 ***
## poly(solar_exposure, 4)3    9404.9    12463.8   0.755    0.451
## poly(solar_exposure, 4)4  -67532.0    12463.8  -5.418 6.72e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12460 on 2069 degrees of freedom
## Multiple R-squared:  0.1326, Adjusted R-squared:  0.131
## F-statistic: 79.11 on 4 and 2069 DF,  p-value: < 2.2e-16
```

```
BIC(mod.num1)
```

```
## [1] 45044.64
```

```
# Diagnostic
par(mfrow=c(2,2))
plot(mod.num1)
```




The behavior of the model is acceptable, since residuals are distributed uniformly across the fitted values, and our model has constant variance.

From the above plots, we conclude that keeping the solar exposure feature can be useful and thus we keep the feature in our model.

Next, we evaluate the effect of adding mean temperature to the linear model.

```
mod.num2 <- lm(data=data, demand ~ poly(solar_exposure, 4) + mean_temperature)
summary(mod.num2)
```

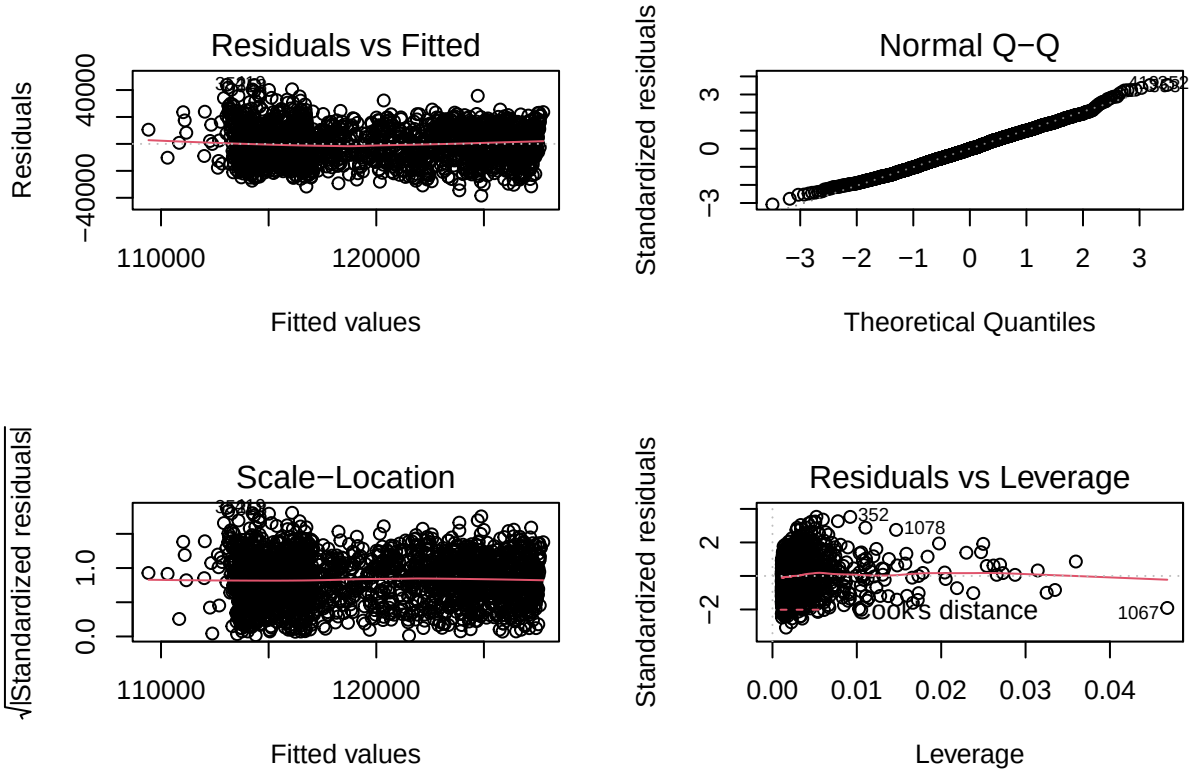
```
##
## Call:
## lm(formula = demand ~ poly(solar_exposure, 4) + mean_temperature,
##     data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -38370  -8531   -183   8663  43870
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                120478.72    1134.73 106.174  < 2e-16 ***
## poly(solar_exposure, 4)1 -179965.14   14850.96 -12.118  < 2e-16 ***
## poly(solar_exposure, 4)2   99660.03   12465.17   7.995 2.13e-15 ***
## poly(solar_exposure, 4)3    8101.84   12588.74   0.644    0.520
## poly(solar_exposure, 4)4  -66410.21   12556.88  -5.289 1.36e-07 ***
## mean_temperature             -51.38      69.38  -0.741    0.459
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12470 on 2068 degrees of freedom
## Multiple R-squared:  0.1329, Adjusted R-squared:  0.1308
## F-statistic: 63.38 on 5 and 2068 DF,  p-value: < 2.2e-16
```

```
BIC(mod.num2)
```

```
## [1] 45051.73
```

```
# Diagnostic
par(mfrow=c(2,2))
plot(mod.num2)
```



For this model the behavior of residuals is still acceptable but BIC is a little worse. We proceed to check the behavior of the model with only the mean temperature variable in order to have a better idea of the effect of this variable.

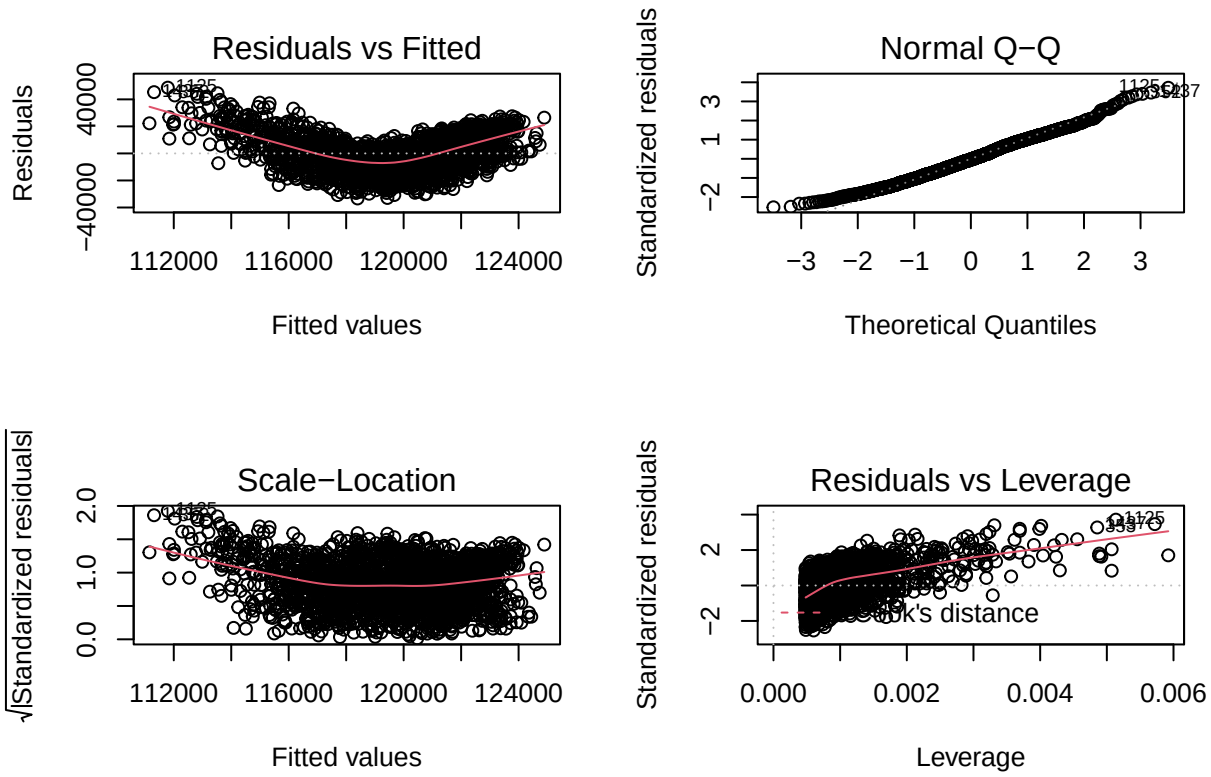
```
mod.num3 <- lm(data=data, demand ~ mean_temperature)
summary(mod.num3)
```

```
##
## Call:
## lm(formula = demand ~ mean_temperature, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -33302  -9577   -477   9599  48644
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      128115.40    1004.27 127.570   <2e-16 ***
## mean_temperature   -532.52      60.61  -8.786   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13130 on 2072 degrees of freedom
## Multiple R-squared:  0.03592,    Adjusted R-squared:  0.03545
## F-statistic:  77.2 on 1 and 2072 DF,  p-value: < 2.2e-16
```

```
BIC(mod.num3)
```

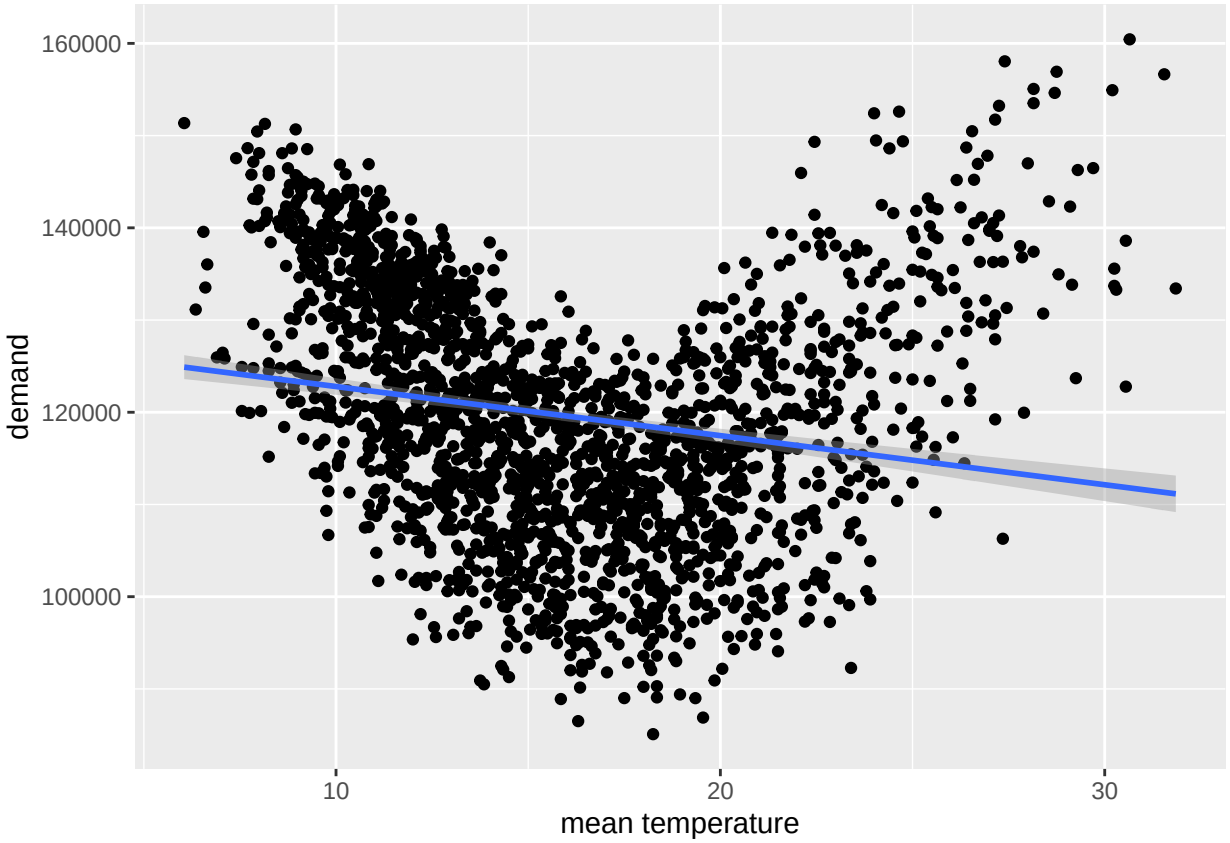
```
## [1] 45241.01
```

```
# Diagnostic
par(mfrow=c(2,2))
plot(mod.num3)
```




The residual plot exhibits a U-shape, which indicates that the data is non-linear.

```
ggplot(data, aes(x = mean_temperature, y = demand)) +
geom_point() +
stat_smooth(method = "lm", formula = y ~ x) +
labs(x = 'mean temperature', y = 'demand')
```

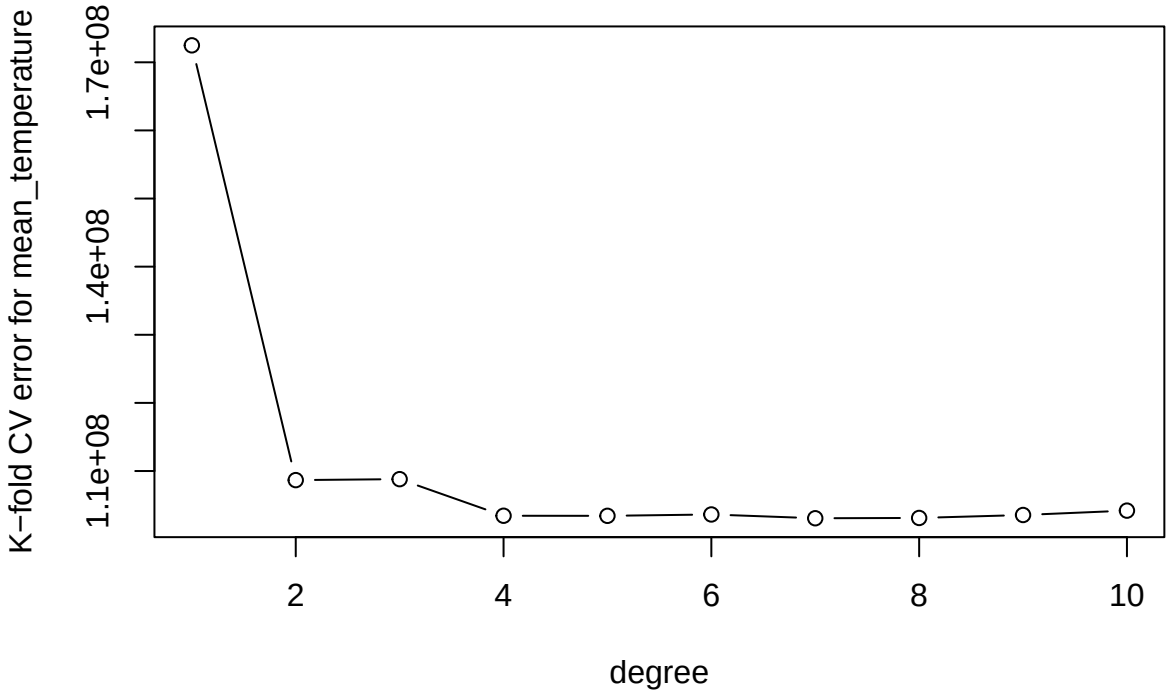



The behavior of mean temperature as seen above suggests that increasing the degree of the mean temperature variable can improve the model. By doing the cross validation, we see that the lowest error is achieved with degree four.

```
#polynomial for mean_temperature
set.seed(1)
cv.error.10.tm <- rep(0,10)
for (i in 1:10){
  glm.fit <- glm(demand~poly(mean_temperature,i),data=data)
  cv.error.10.tm[i] <- cv.glm(data,glm.fit,K=10)$delta[1]
}
cv.error.10.tm
```

```
##  [1] 172488197 108664192 108806093 103431869 103419951 103619940 103066645
##  [8] 103108967 103542567 104184716
```

```
temp.pol = plot(1:10, cv.error.10.tm, type="b", xlab="degree", ylab="K-fold CV error for mean_temperatu
```

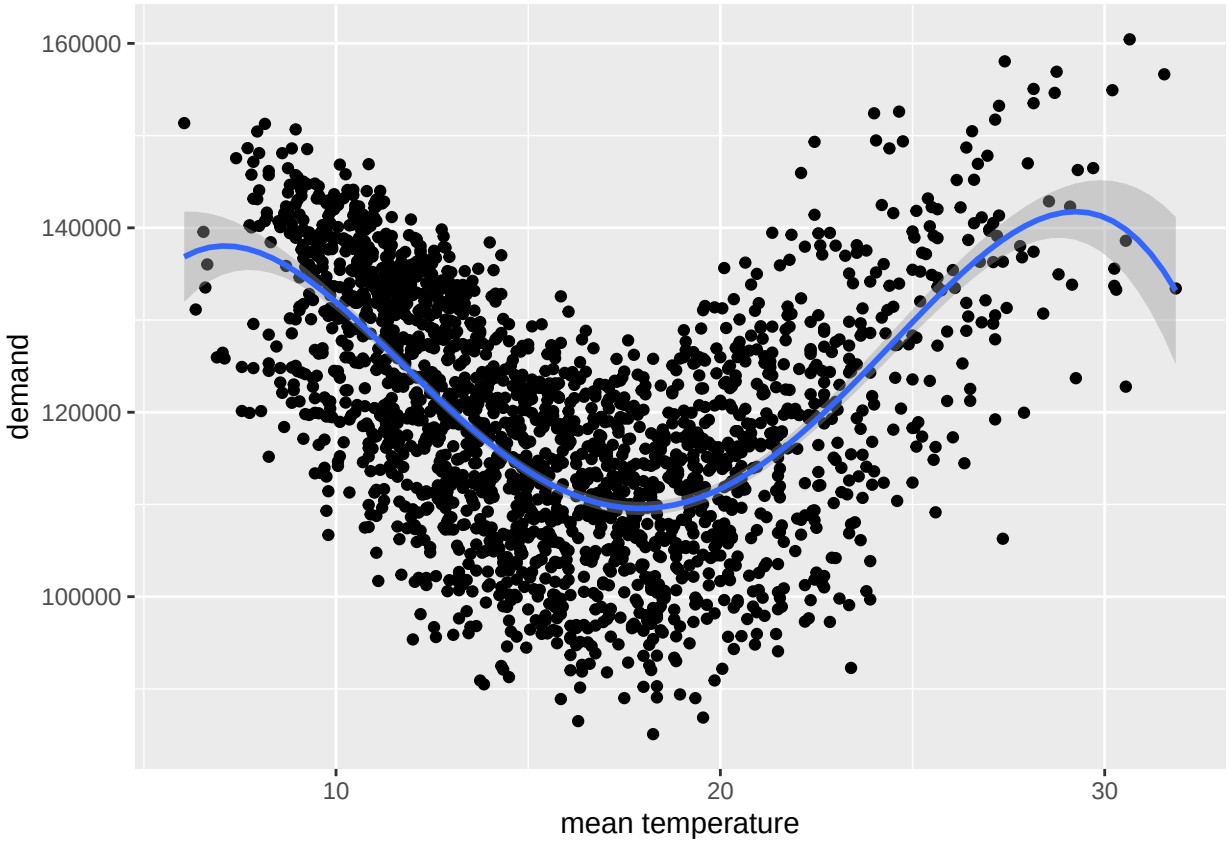



```
temp.pol
```

```
## NULL
```

We produce the fourth degree model:

```
ggplot(data, aes(x = mean_temperature, y = demand)) +
geom_point() +
stat_smooth(method = "lm", formula = y ~ poly(x, 4)) +
labs(x = 'mean temperature', y = 'demand')
```




```
mod.num4 <- lm(data=data, demand ~ poly(mean_temperature, 4))
summary(mod.num4)
```

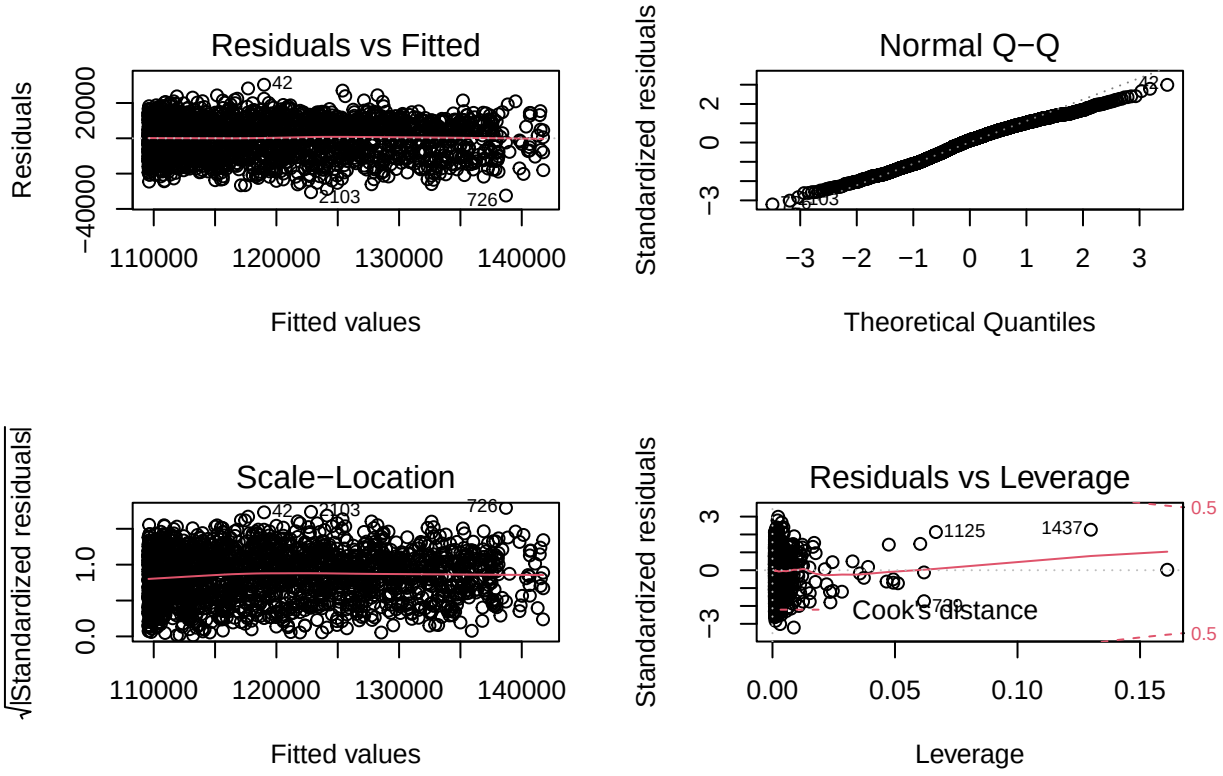
```
##
## Call:
## lm(formula = demand ~ poly(mean_temperature, 4), data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -32429  -7511   1192   7700  30329
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   119663.2      222.9 536.766   <2e-16 ***
## poly(mean_temperature, 4)1 -115370.4    10152.7 -11.364   <2e-16 ***
## poly(mean_temperature, 4)2  364207.6    10152.7  35.873   <2e-16 ***
## poly(mean_temperature, 4)3  -15234.3    10152.7  -1.501    0.134
## poly(mean_temperature, 4)4 -105412.8    10152.7 -10.383   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10150 on 2069 degrees of freedom
## Multiple R-squared:  0.4245, Adjusted R-squared:  0.4234
## F-statistic: 381.5 on 4 and 2069 DF,  p-value: < 2.2e-16
```

```
BIC(mod.num4)
```

```
## [1] 44193.91
```

The BIC for this fourth degree model is better than for the first degree one. We also compared with the BIC for the second and third degree models, and this had the best performance.

```
# Diagnostic
par(mfrow=c(2,2))
plot(mod.num4)
```




Behavior of the residuals is also better.

We proceed to add this new variable to the model and check the performance.

```
mod.num5 <- lm(data=data, demand ~ poly(solar_exposure, 4) + poly(mean_temperature, 4))
summary(mod.num5)
```

```
##
## Call:
## lm(formula = demand ~ poly(solar_exposure, 4) + poly(mean_temperature,
##     4), data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

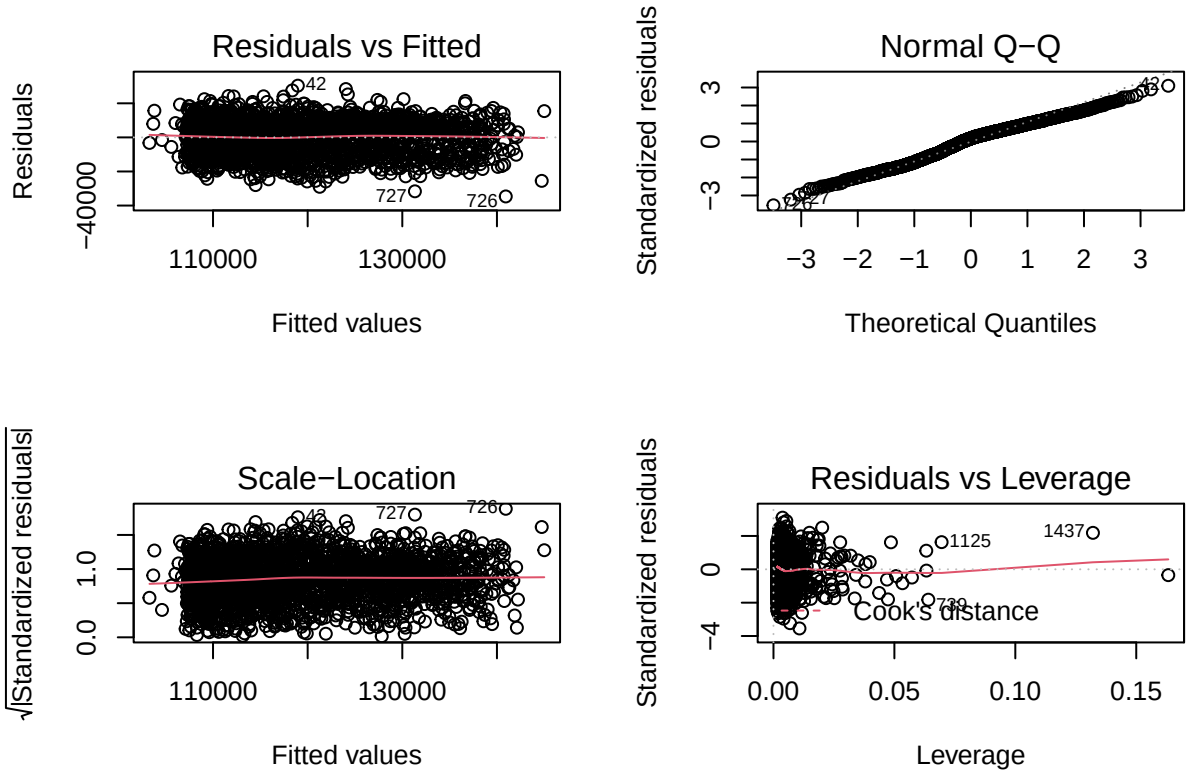
```
## -34648  -7406   1689   7130  30360
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  119663.2      215.7 554.731  < 2e-16 ***
## poly(solar_exposure, 4)1   -108604.2    12000.1  -9.050  < 2e-16 ***
## poly(solar_exposure, 4)2     65449.6     9878.8   6.625 4.41e-11 ***
## poly(solar_exposure, 4)3    -16374.4     9950.5  -1.646   0.1000
## poly(solar_exposure, 4)4    -43833.8     9917.4  -4.420 1.04e-05 ***
## poly(mean_temperature, 4)1  -54578.5    11948.3  -4.568 5.22e-06 ***
## poly(mean_temperature, 4)2  343618.4    10018.8  34.297  < 2e-16 ***
## poly(mean_temperature, 4)3  -22614.6     9911.3  -2.282   0.0226 *
## poly(mean_temperature, 4)4  -94529.1     9878.7  -9.569  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9824 on 2065 degrees of freedom
## Multiple R-squared:  0.4622, Adjusted R-squared:  0.4601
## F-statistic: 221.8 on 8 and 2065 DF,  p-value: < 2.2e-16
```

```
BIC(mod.num5)
```

```
## [1] 44083.89
```

With this model, the BIC improves.

```
# Diagnostic
par(mfrow=c(2,2))
plot(mod.num5)
```




Residual behavior is also acceptable. Since the model behavior improved, we keep this fourth degree term and go on to add a variable that we previously examined on the EDA section, log_rainfall.

```
mod.num6 <- lm(data=data, demand ~ poly(solar_exposure, 4) +
                 poly(mean_temperature, 4) + log_rainfall)
summary(mod.num6)
```

```
##
## Call:
## lm(formula = demand ~ poly(solar_exposure, 4) + poly(mean_temperature,
##     4) + log_rainfall, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -34899  -7493   1707   7038  30087
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   120313.5      250.4 480.517  < 2e-16 ***
## poly(solar_exposure, 4)1   -115104.6    11999.8  -9.592  < 2e-16 ***
## poly(solar_exposure, 4)2     67421.6     9828.9   6.859 9.10e-12 ***
## poly(solar_exposure, 4)3    -20840.5     9932.2  -2.098   0.0360 *
## poly(solar_exposure, 4)4    -41422.6     9871.1  -4.196 2.83e-05 ***
## poly(mean_temperature, 4)1  -62710.1    11987.9  -5.231 1.86e-07 ***
## poly(mean_temperature, 4)2  342992.2     9961.1  34.433  < 2e-16 ***
## poly(mean_temperature, 4)3  -20527.4     9862.1  -2.081   0.0375 *
```

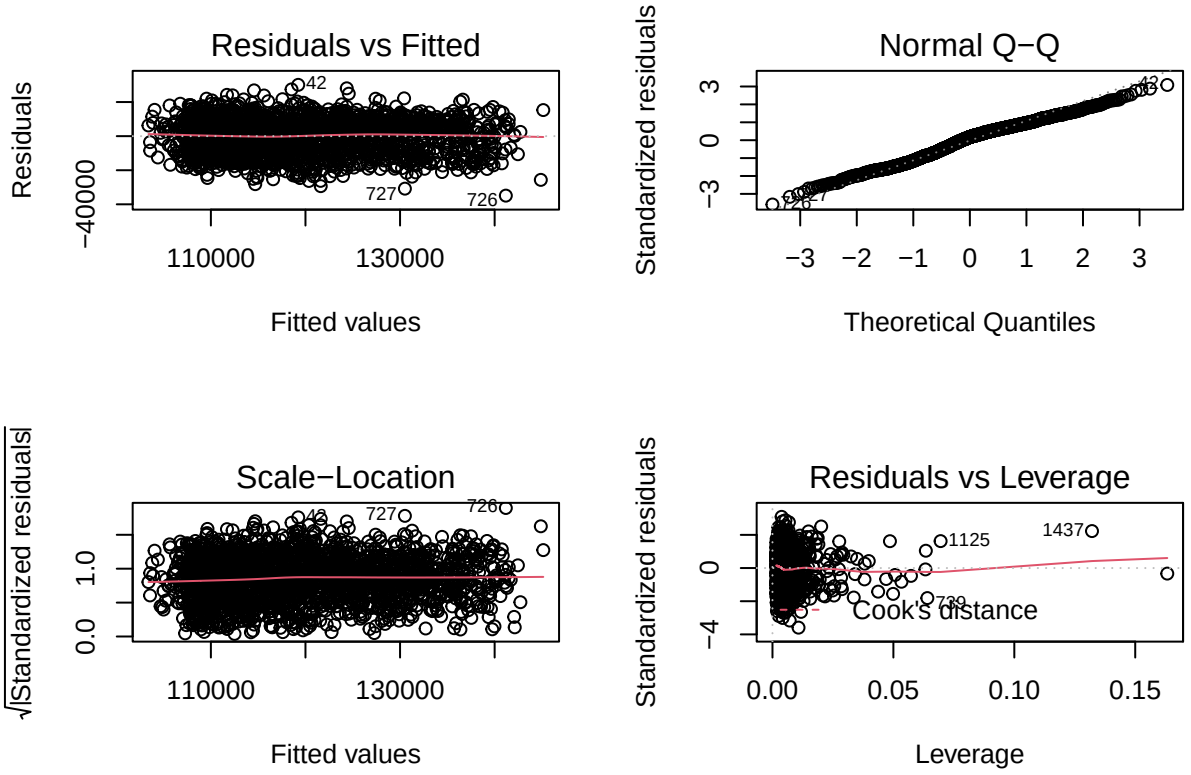
```
## poly(mean_temperature, 4)4  -96424.8     9828.3  -9.811  < 2e-16 ***
## log_rainfall                 -1449.9      288.1  -5.032 5.27e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9767 on 2064 degrees of freedom
## Multiple R-squared:  0.4687, Adjusted R-squared:  0.4664
## F-statistic: 202.3 on 9 and 2064 DF,  p-value: < 2.2e-16
```

```
BIC(mod.num6)
```

```
## [1] 44066.24
```

By adding the rainfall feature to the model, BIC improves a little.

```
# Diagnostic
par(mfrow=c(2,2))
plot(mod.num6)
```




Residual plots show an admissible behavior. Now, we check the behavior of the model with only the log_rainfall variable.

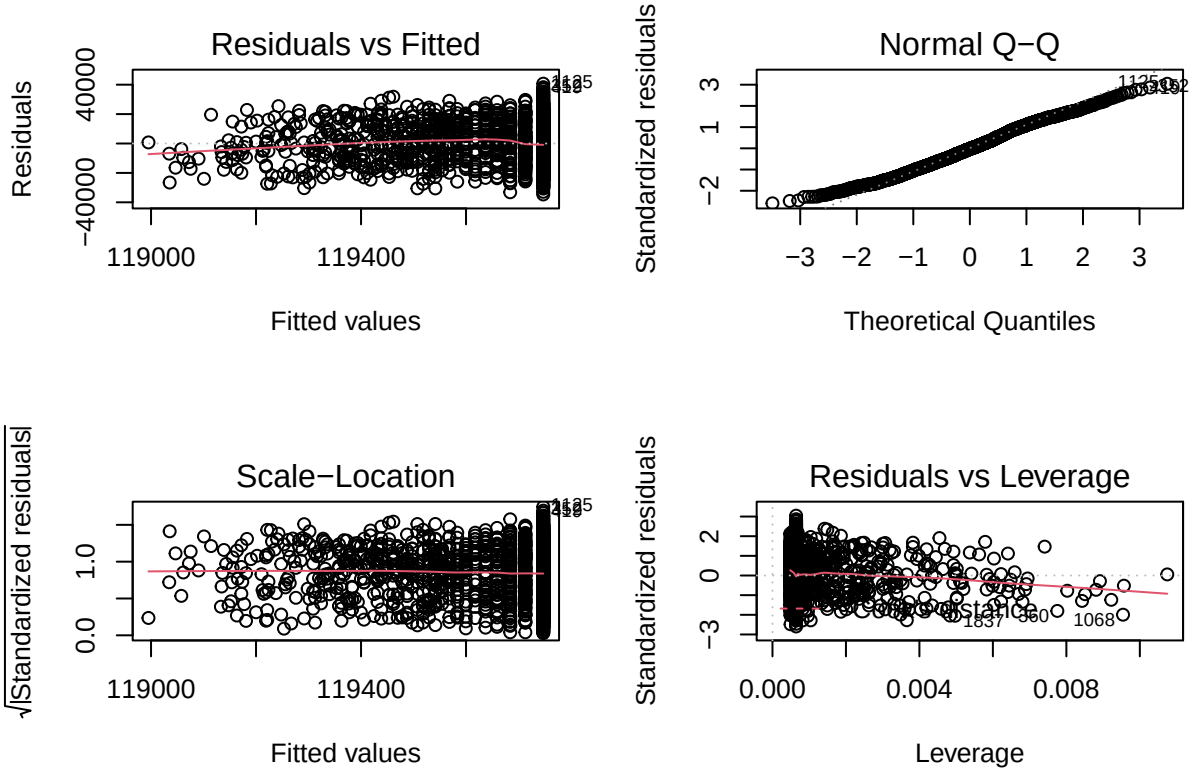
```
mod.num7 <- lm(data=data, demand ~ log_rainfall)
summary(mod.num7)
```

```
##
## Call:
## lm(formula = demand ~ log_rainfall, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -34653  -9776   -424  10081  40690
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  119747.1      339.4 352.804   <2e-16 ***
## log_rainfall   -187.2      379.5  -0.493    0.622
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13370 on 2072 degrees of freedom
## Multiple R-squared:  0.0001173,  Adjusted R-squared:  -0.0003652
## F-statistic: 0.2431 on 1 and 2072 DF,  p-value: 0.622
```

```
BIC(mod.num7)
```

```
## [1] 45316.64
```

```
# Diagnostic
par(mfrow=c(2,2))
plot(mod.num7)
```



The "log_rainfall" does not have a big impact on the model, but we keep it for now and will further analyze and decide afterwards. We proceed to build a table to compare the BIC and Adjusted $R^2$ of the different models more easily.

| Variables_num | BIC_num | Adjusted_R2_num |
|---|---|---|
| solar exposure^4/mean temperature | 45119.29 | 0.1308000 |
| solar exposure^4/mean temperature^4 | 44083.89 | 0.4601000 |
| solar exposure^4/mean temperature^4/log rainfall | 44066.24 | 0.4664000 |
| solar exposure^4 | 45044.64 | 0.1310000 |
| solar exposure | 45113.74 | 0.0928700 |
| mean temperature^4 | 44193.91 | 0.4234000 |
| mean temperature | 45241.01 | -0.0354500 |
| log rainfall | 45316.64 | -0.0003652 |

From the above table we can see that the model behavior does not improve much by adding log rainfall so we decide to remove this feature. Therefore, the resulting model has only the fourth degree solar exposure and mean temperature variables.

### 4.1.2 Adding categorical variables

Next, we build a model considering only the categorical variables. We start considering all three of them.

```
mod.cat0 <- lm(demand ~ school_day + season + holiday, data=data)
summary(mod.cat0)
```

```
##
## Call:
## lm(formula = demand ~ school_day + season + holiday, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -30163  -7995    230   7229  44663
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    115774.2      600.7 192.747  < 2e-16 ***
## school_dayTRUE   2278.9      549.5   4.148  3.5e-05 ***
## seasonspring    -2597.9      726.1  -3.578 0.000355 ***
## seasonwinter    13305.7      707.3  18.813  < 2e-16 ***
## seasonautumn     -212.6      703.6  -0.302 0.762571
## holidayTRUE    -14468.1     1330.8 -10.872  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11220 on 2068 degrees of freedom
## Multiple R-squared:  0.297,  Adjusted R-squared:  0.2953
## F-statistic: 174.7 on 5 and 2068 DF,  p-value: < 2.2e-16
```

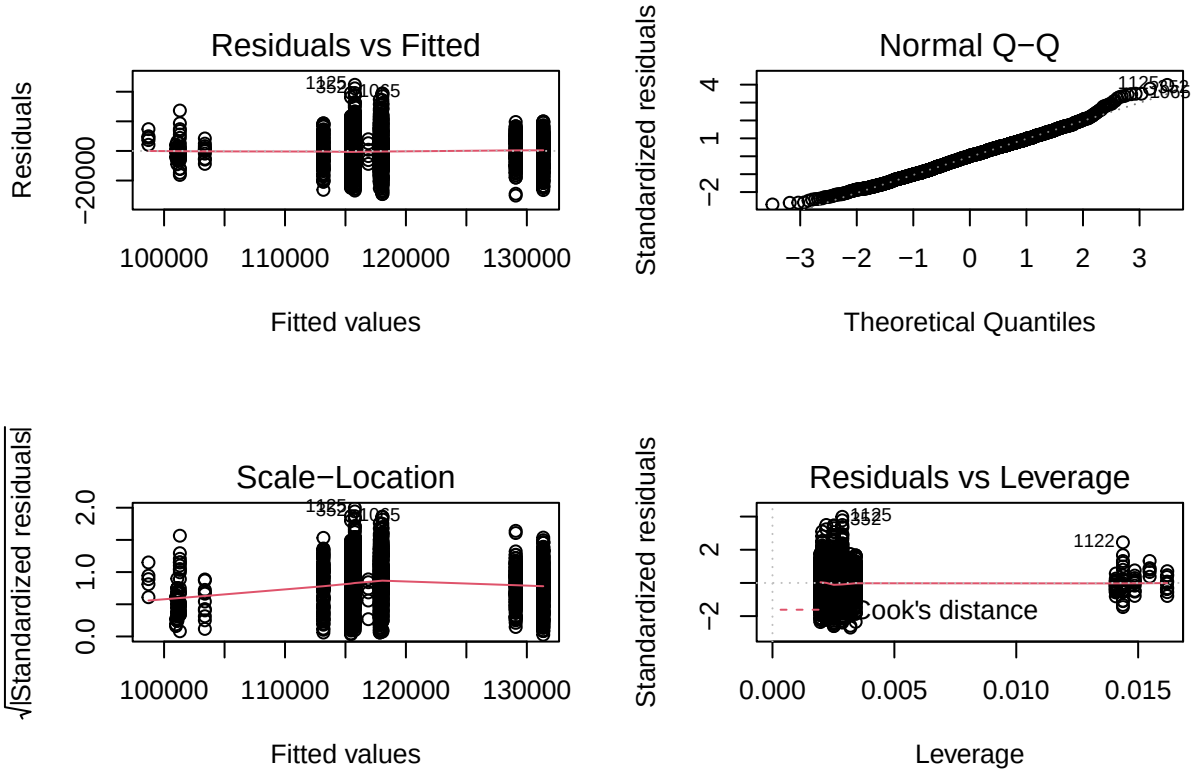
```
summary(mod.cat0)$adj.r.squared
```

```
## [1] 0.2953041
```

```
BIC(mod.cat0)
```

```
## [1] 44616.54
```

```
par(mfrow=c(2,2))
plot(mod.cat0)
```




Next, since they are not many, we consider all possible subsets of the variables to find the model which results in the best performance in terms of BIC and Adjusted $R^2$. We start with a combination of the school day and season features.

```
mod.cat1 <- lm(demand ~ school_day + season , data=data)
summary(mod.cat1)
```

```
##
## Call:
## lm(formula = demand ~ school_day + season, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -29369  -8596    304   7496  45974
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)    114463.8      604.9 189.237  < 2e-16 ***
## school_dayTRUE   3263.8      557.1   5.859 5.41e-09 ***
## seasonspring    -2316.7      745.9  -3.106  0.00192 **
## seasonwinter    13717.3      726.0  18.895  < 2e-16 ***
## seasonautumn     -526.0      722.6  -0.728  0.46674
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11540 on 2069 degrees of freedom
## Multiple R-squared:  0.2568, Adjusted R-squared:  0.2554
## F-statistic: 178.7 on 4 and 2069 DF,  p-value: < 2.2e-16
```

```
summary(mod.cat1)$adj.r.squared
```

```
## [1] 0.2553871
```

```
BIC(mod.cat1)
```

```
## [1] 44724.18
```

Next, we combine the holiday and season features.

```
mod.cat2 <- lm(demand ~ holiday + season , data=data)
summary(mod.cat2)
```

```
##
## Call:
## lm(formula = demand ~ holiday + season, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -31886  -8075    228   7345  43359
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  117078.2      513.8 227.870  < 2e-16 ***
## holidayTRUE  -15378.1     1317.7 -11.670  < 2e-16 ***
## seasonspring  -2189.2      722.2  -3.031  0.00247 **
## seasonwinter  13725.0      702.7  19.531  < 2e-16 ***
## seasonautumn    164.5      700.4   0.235  0.81437
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11270 on 2069 degrees of freedom
## Multiple R-squared:  0.2912, Adjusted R-squared:  0.2898
## F-statistic: 212.5 on 4 and 2069 DF,  p-value: < 2.2e-16
```

```
summary(mod.cat2)$adj.r.squared
```

```
## [1] 0.2897857
```

```
BIC(mod.cat2)
```

```
## [1] 44626.09
```

Then, we try a combination of the holiday and school day features.

```
mod.cat3 <- lm(demand ~ holiday + school_day , data=data)
summary(mod.cat3)
```

```
##
## Call:
## lm(formula = demand ~ holiday + school_day, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -32288  -9195   -685   9775  42217
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    118221.0      525.9 224.802  < 2e-16 ***
## holidayTRUE    -16509.6     1518.3 -10.874  < 2e-16 ***
## school_dayTRUE   2970.2      621.7   4.777  1.9e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12880 on 2071 degrees of freedom
## Multiple R-squared:  0.07334,    Adjusted R-squared:  0.07244
## F-statistic: 81.95 on 2 and 2071 DF,  p-value: < 2.2e-16
```

```
summary(mod.cat3)$adj.r.squared
```

```
## [1] 0.07244284
```

```
BIC(mod.cat3)
```

```
## [1] 45166.55
```

We move on to consider only school day.

```
mod.cat4 <- lm(demand ~ school_day , data=data)
summary(mod.cat4)
```

```
##
## Call:
## lm(formula = demand ~ school_day, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -32035  -9710   -557  10108  43638
##
```

```
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    116800.0      523.6 223.068  < 2e-16 ***
## school_dayTRUE   4138.1      629.5   6.574 6.19e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13240 on 2072 degrees of freedom
## Multiple R-squared:  0.02043,    Adjusted R-squared:  0.01996
## F-statistic: 43.22 on 1 and 2072 DF,  p-value: 6.185e-11
```

```
summary(mod.cat4)$adj.r.squared
```

```
## [1] 0.01995818
```

```
BIC(mod.cat4)
```

```
## [1] 45274.07
```

Next, we only consider holiday.

```
mod.cat5 <- lm(demand ~ holiday , data=data)
summary(mod.cat5)
```

```
##
## Call:
## lm(formula = demand ~ holiday, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -33431  -9111   -584   9797  40115
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 120322.7      289.7  415.39   <2e-16 ***
## holidayTRUE -17762.6     1503.3  -11.82   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12940 on 2072 degrees of freedom
## Multiple R-squared:  0.06313,    Adjusted R-squared:  0.06267
## F-statistic: 139.6 on 1 and 2072 DF,  p-value: < 2.2e-16
```

```
summary(mod.cat5)$adj.r.squared
```

```
## [1] 0.0626738
```

```
BIC(mod.cat5)
```

```
## [1] 45181.64
```

Finally, we consider only the season feature.

```
mod.cat6 <- lm(demand ~ season , data=data)
summary(mod.cat6)
```

```
##
## Call:
## lm(formula = demand ~ season, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -31718  -8376    270   7794  44175
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.163e+05  5.254e+02  221.29   <2e-16 ***
## seasonspring -1.689e+03  7.441e+02   -2.27   0.0233 *
## seasonwinter  1.437e+04  7.231e+02   19.88   <2e-16 ***
## seasonautumn  6.229e-02  7.228e+02    0.00   0.9999
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11630 on 2070 degrees of freedom
## Multiple R-squared:  0.2445, Adjusted R-squared:  0.2434
## F-statistic: 223.3 on 3 and 2070 DF,  p-value: < 2.2e-16
```

```
summary(mod.cat6)$adj.r.squared
```

```
## [1] 0.2433993
```

```
BIC(mod.cat6)
```

```
## [1] 44750.67
```

| Variables_cat | BIC_cat | Adjusted_R2_cat |
|---|---|---|
| school day/season/holiday | 44616.54 | 0.2953041 |
| school day/season | 44724.16 | 0.2553871 |
| season/holiday | 44626.09 | 0.2897857 |
| school day/holiday | 45166.55 | 0.0724428 |
| school day | 45274.07 | 0.0199582 |
| holiday | 45181.64 | 0.0626738 |
| season | 44750.67 | 0.2433993 |

After comparing the BIC and Adjusted $R^2$ for all the different models, we conclude that the best model performance is achieved when considering all three categorical variables. However, since during the exploratory phase we determined that the holiday feature is severely unbalanced, we remove it from our model and keep only the season and school day variables.

### 4.1.3 Unified model

Combining all the numerical and categorical variables we get the following model:

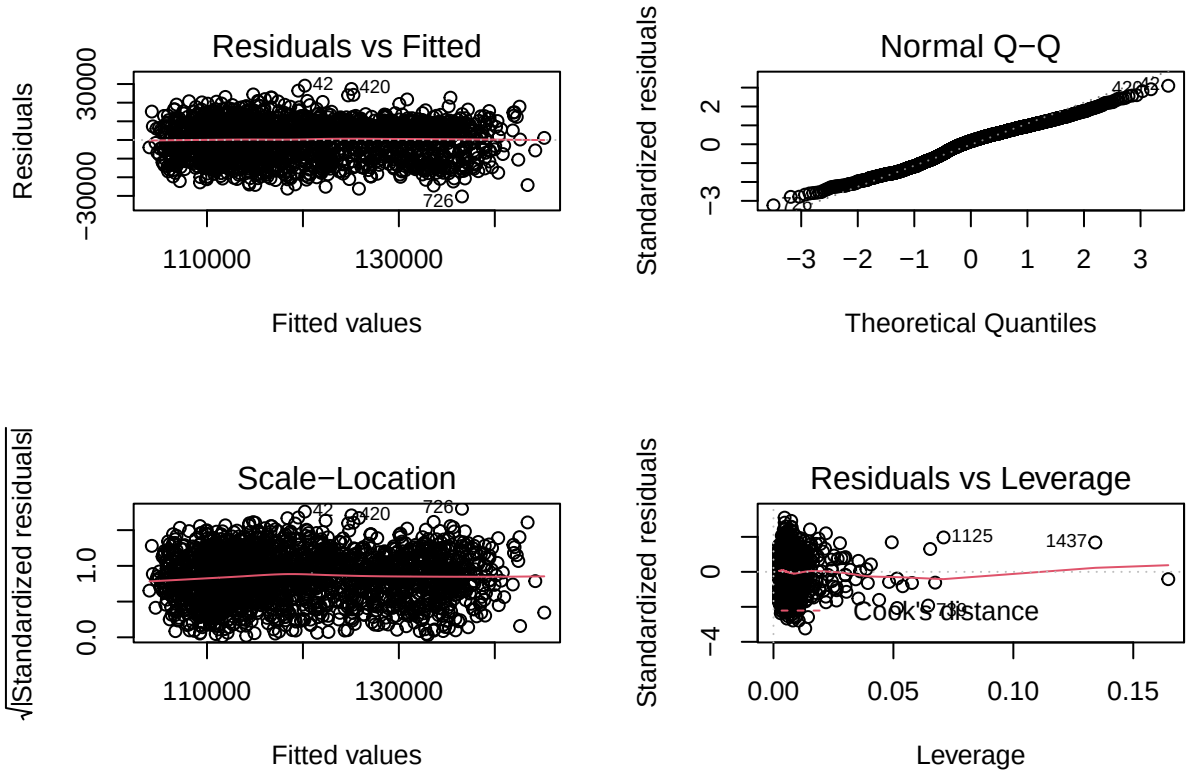
```
mod.tot0 <- lm(demand ~ poly(solar_exposure, 4) + poly(mean_temperature, 4) +
                 season + school_day , data=data)
summary(mod.tot0)
```

```
##
## Call:
## lm(formula = demand ~ poly(solar_exposure, 4) + poly(mean_temperature,
##     4) + season + school_day, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -30293  -7238   1546   6874  29124
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  114831.2      633.0 181.414  < 2e-16 ***
## poly(solar_exposure, 4)1     -57979.2    13818.9  -4.196 2.84e-05 ***
## poly(solar_exposure, 4)2      54558.5    10003.5   5.454 5.52e-08 ***
## poly(solar_exposure, 4)3     -21922.1     9779.6  -2.242   0.0251 *
## poly(solar_exposure, 4)4     -19137.2     9699.4  -1.973   0.0486 *
## poly(mean_temperature, 4)1     8373.3    14693.1   0.570   0.5688
## poly(mean_temperature, 4)2 307063.1    10730.4  28.616  < 2e-16 ***
## poly(mean_temperature, 4)3    -4693.3     9710.9  -0.483   0.6289
## poly(mean_temperature, 4)4 -84327.0     9627.0  -8.759  < 2e-16 ***
## seasonspring                  -557.8      734.3  -0.760   0.4475
## seasonwinter                  7393.9      989.6   7.472 1.16e-13 ***
## seasonautumn                  1372.1      753.1   1.822   0.0686 .
## school_dayTRUE                3824.4      462.8   8.264 2.49e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9427 on 2061 degrees of freedom
## Multiple R-squared:  0.5057, Adjusted R-squared:  0.5029
## F-statistic: 175.7 on 12 and 2061 DF,  p-value: < 2.2e-16
```

```
BIC(mod.tot0)
```

```
## [1] 43939.4
```

```
par(mfrow=c(2,2))
plot(mod.tot0)
```



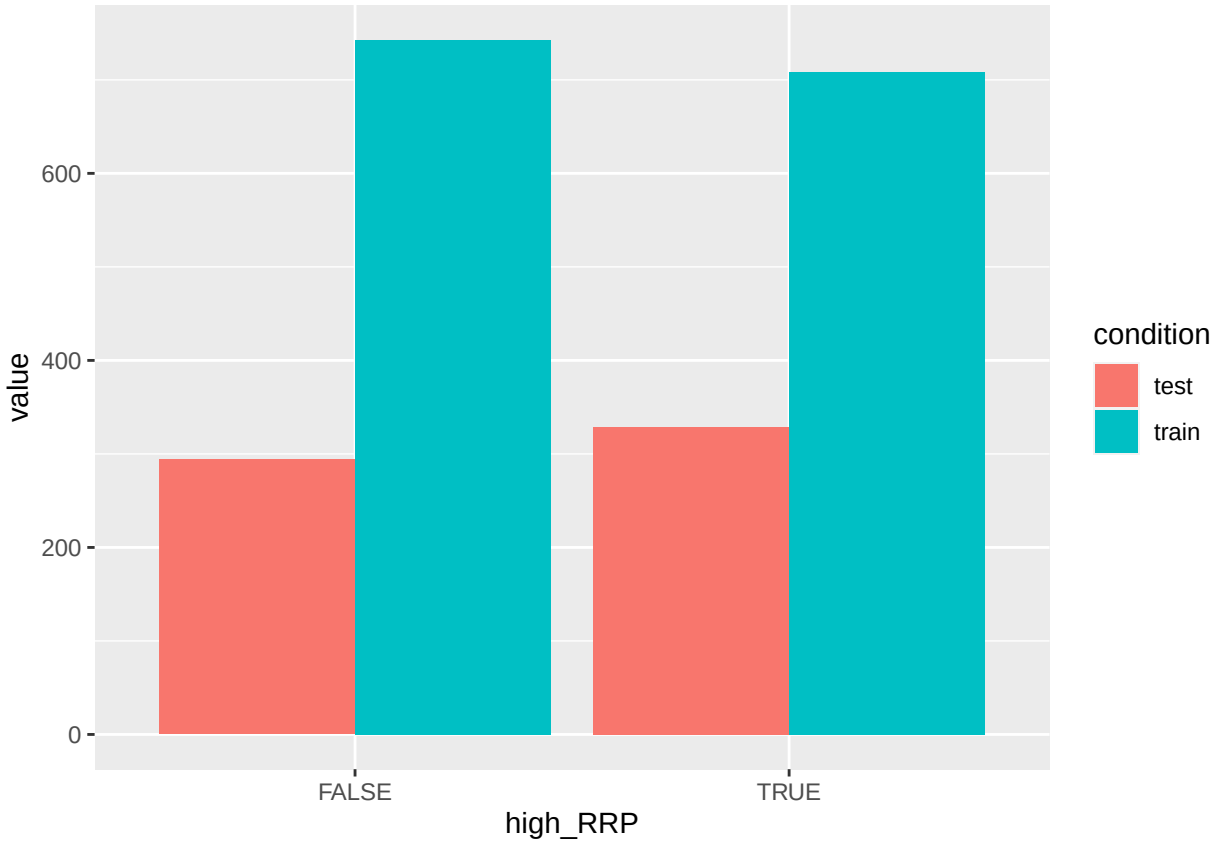

Behavior of the residuals is acceptable since they are distributed uniformly across the fitted values, and the model has constant variance. The BIC and Adjusted $R^2$ are better with respect to the all the previous models.

## 4.2 Logistic Regression

In this section we will build a logistic regression model in order to classify RRP as high or not, depending on whether it is above or below the median. To measure performance we will compare the different AUC (Area Under the Curve).

```
attach(data)
data$month <- as.factor(data$month)


#add the high RRP column
data$high_RRP <- TRUE
data$high_RRP[data$RRP <= median(data$RRP)] <- FALSE

#split in validation and train sets
dt = sort(sample(nrow(data), nrow(data)*.7))
train<-data[dt,]
test<-data[-dt,]

train$source <- "train"
test$source<- "test"
```

```r
toplot <- rbind(train,test)

# create a dataset
high_RRP <- c(TRUE, TRUE, FALSE, FALSE )
condition <- c("train" , "test", "train", "test")
value <- c(table(toplot$source, toplot$high_RRP)[2,2], table(toplot$source, toplot$high_RRP)[1,2], table
toplot2 <- data.frame(high_RRP,condition,value)

# Grouped
ggplot(toplot2, aes(fill=condition, y=value, x=high_RRP)) +
  geom_bar(position="dodge", stat="identity")
```




The sets were separated into 70% training and 30% validation, obtaining samples of the original set randomly. The TRUE and FALSE labels are equally balanced in the train and test sets. This is to be expected since the split was at the median.

### 4.2.1 Logistic Regression Using Seasons

For this first part, a model will be tested using the season variable, then it will be replaced by the month variable, in order to compare and select the best performance.

We will start by building an initial model using the variables "demand", "mean_temperature", "solar_exposure", the logarithm of "rainfall", "holiday", "season", that the EDA showed are the most descriptive for the RRP. Then, through a process of backward elimination, the variables that were making less contribution to the model will be simplified, observing the p-value and the deviance as criteria to obtain the best model.

We build the first model:

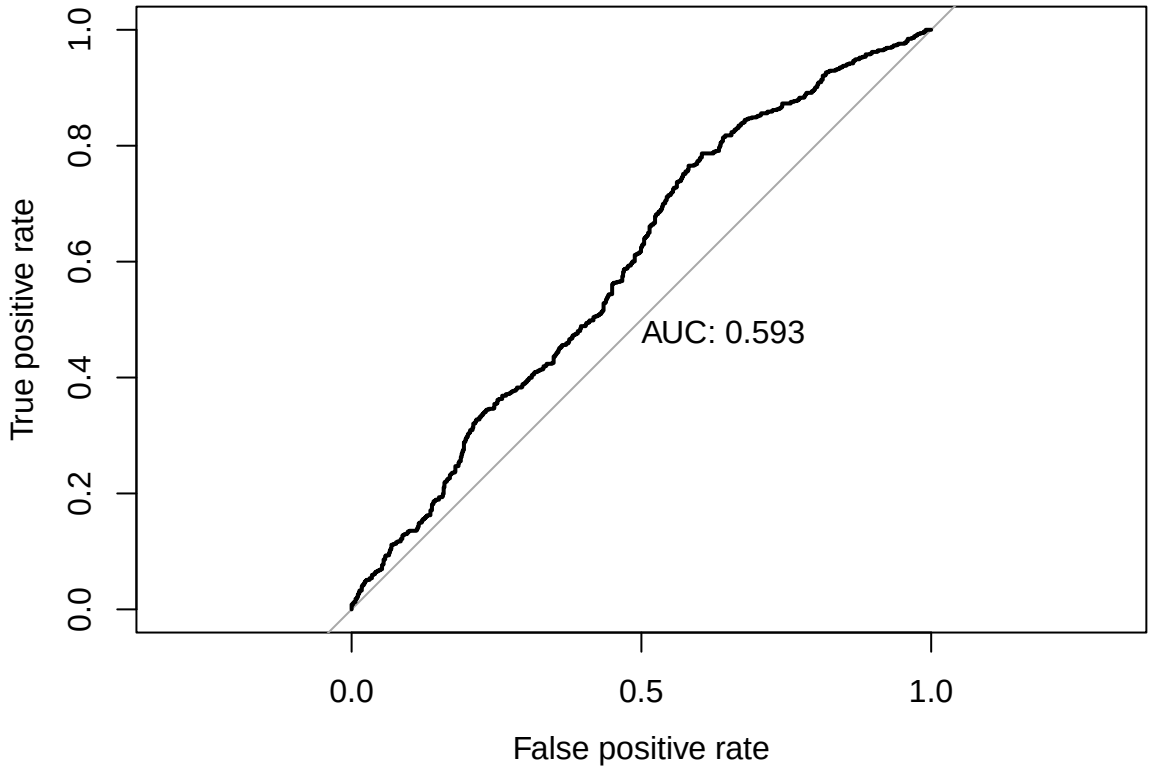
```
logit.out1 <- glm(high_RRP ~ demand + mean_temperature + solar_exposure + log(rainfall +1) + holiday + s

summary(logit.out1)
```

```
##
## Call:
## glm(formula = high_RRP ~ demand + mean_temperature + solar_exposure +
##     log(rainfall + 1) + holiday + school_day + season, family = binomial,
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4381  -1.1535  -0.8043   1.1648   1.6414
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -2.208e+00  6.802e-01  -3.247 0.001168 **
## demand             1.270e-05  4.907e-06   2.589 0.009629 **
## mean_temperature   2.177e-03  1.726e-02   0.126 0.899646
## solar_exposure     9.523e-03  9.884e-03   0.963 0.335302
## log(rainfall + 1) -1.093e-01  7.405e-02  -1.477 0.139789
## holidayTRUE       -2.815e-01  3.157e-01  -0.892 0.372539
## school_dayTRUE     4.050e-01  1.203e-01   3.365 0.000765 ***
## seasonspring       3.904e-01  1.831e-01   2.132 0.032991 *
## seasonwinter       4.295e-01  2.575e-01   1.668 0.095413 .
## seasonautumn       1.437e-01  1.918e-01   0.749 0.453744
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2010.7  on 1450  degrees of freedom
## Residual deviance: 1966.3  on 1441  degrees of freedom
## AIC: 1986.3
##
## Number of Fisher Scoring iterations: 4
```

```
logistic.prob <- predict(logit.out1, type="response")
roc.out <- roc(train$high_RRP, logistic.prob, levels=c(FALSE, TRUE))
```

```
## Setting direction: controls < cases
```

```
plot(roc.out, print.auc=TRUE, legacy.axes=TRUE, xlab="False positive rate", ylab="True positive rate")
```




It can be understood that it is not possible to find a model with a high "true positive rate" and a low "false positive rate" to define the threshold as seen in the ROC (receiver operating characteristic curve. In any case, this process will continue.

Based on the P-values obtained from the previous model, we decide to remove the holiday feature, as it's the highest, and continue like this until significant changes in the deviance are obtained, which would indicate the end of this process

```
logit.out2 <- glm(high_RRP ~ demand + mean_temperature + solar_exposure + log(rainfall +1) + school_day
summary(logit.out2)
```

```
##
## Call:
## glm(formula = high_RRP ~ demand + mean_temperature + solar_exposure +
##     log(rainfall + 1) + school_day + season, family = binomial,
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4444  -1.1520  -0.8466   1.1672   1.6594
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -2.341e+00  6.644e-01  -3.523 0.000426 ***
```

```
## demand             1.375e-05  4.769e-06   2.883 0.003938 **
## mean_temperature   1.370e-03  1.724e-02   0.079 0.936662
## solar_exposure     9.707e-03  9.873e-03   0.983 0.325519
## log(rainfall + 1) -1.079e-01  7.408e-02  -1.457 0.145061
## school_dayTRUE     4.197e-01  1.193e-01   3.519 0.000433 ***
## seasonspring       3.947e-01  1.831e-01   2.156 0.031071 *
## seasonwinter       4.180e-01  2.573e-01   1.625 0.104220
## seasonautumn       1.395e-01  1.917e-01   0.728 0.466858
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2010.7  on 1450  degrees of freedom
## Residual deviance: 1967.1  on 1442  degrees of freedom
## AIC: 1985.1
##
## Number of Fisher Scoring iterations: 4
```

```
logit.out3 <- glm(high_RRP ~ demand + mean_temperature + log(rainfall+1) + school_day + season, data = t
summary(logit.out3)
```

```
##
## Call:
## glm(formula = high_RRP ~ demand + mean_temperature + log(rainfall +
##     1) + school_day + season, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4471  -1.1596  -0.8433   1.1650   1.6975
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -2.117e+00  6.229e-01  -3.398 0.000678 ***
## demand             1.317e-05  4.729e-06   2.785 0.005354 **
## mean_temperature   4.655e-03  1.690e-02   0.276 0.782929
## log(rainfall + 1) -1.229e-01  7.249e-02  -1.695 0.090006 .
## school_dayTRUE     4.199e-01  1.192e-01   3.522 0.000429 ***
## seasonspring       3.620e-01  1.798e-01   2.014 0.044029 *
## seasonwinter       3.199e-01  2.369e-01   1.350 0.177034
## seasonautumn       4.828e-02  1.675e-01   0.288 0.773162
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2010.7  on 1450  degrees of freedom
## Residual deviance: 1968.1  on 1443  degrees of freedom
## AIC: 1984.1
##
## Number of Fisher Scoring iterations: 4
```

```r
logit.out4 <- glm(high_RRP ~ demand + mean_temperature + school_day + season, data = train, family = bi

summary(logit.out4)
```

```
##
## Call:
## glm(formula = high_RRP ~ demand + mean_temperature + school_day +
##     season, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4499  -1.1566  -0.8699   1.1713   1.5754
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -2.283e+00  6.149e-01  -3.713 0.000205 ***
## demand            1.305e-05  4.724e-06   2.763 0.005722 **
## mean_temperature  1.132e-02  1.642e-02   0.690 0.490468
## school_dayTRUE    4.185e-01  1.191e-01   3.514 0.000441 ***
## seasonspring      3.760e-01  1.795e-01   2.095 0.036180 *
## seasonwinter      3.598e-01  2.355e-01   1.527 0.126666
## seasonautumn      7.115e-02  1.668e-01   0.427 0.669669
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2010.7  on 1450  degrees of freedom
## Residual deviance: 1971.0  on 1444  degrees of freedom
## AIC: 1985
##
## Number of Fisher Scoring iterations: 4
```

```r
logit.out5 <- glm(high_RRP ~ demand + school_day + season, data = train, family = binomial)

summary(logit.out5)
```

```
##
## Call:
## glm(formula = high_RRP ~ demand + school_day + season, family = binomial,
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4432  -1.1546  -0.8708   1.1681   1.5568
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.101e+00  5.542e-01  -3.791 0.000150 ***
## demand          1.350e-05  4.677e-06   2.887 0.003891 **
## school_dayTRUE  4.200e-01  1.191e-01   3.528 0.000419 ***
## seasonspring    3.181e-01  1.584e-01   2.008 0.044641 *
```

```
## seasonwinter    2.446e-01  1.660e-01   1.473 0.140767
## seasonautumn    2.269e-02  1.512e-01   0.150 0.880669
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2010.7  on 1450  degrees of freedom
## Residual deviance: 1971.5  on 1445  degrees of freedom
## AIC: 1983.5
##
## Number of Fisher Scoring iterations: 4
```

AUC improved but not by much.

Then, we try a model without considering the seasons:

```
logit.out6 <- glm(high_RRP ~ demand + school_day ,data = train, family = binomial)

summary(logit.out6)
```

```
##
## Call:
## glm(formula = high_RRP ~ demand + school_day, family = binomial,
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4235  -1.1613  -0.8931   1.1659   1.5210
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -2.161e+00  4.888e-01  -4.421 9.84e-06 ***
## demand          1.501e-05  4.078e-06   3.681 0.000232 ***
## school_dayTRUE  4.515e-01  1.173e-01   3.850 0.000118 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2010.7  on 1450  degrees of freedom
## Residual deviance: 1977.8  on 1448  degrees of freedom
## AIC: 1983.8
##
## Number of Fisher Scoring iterations: 4
```

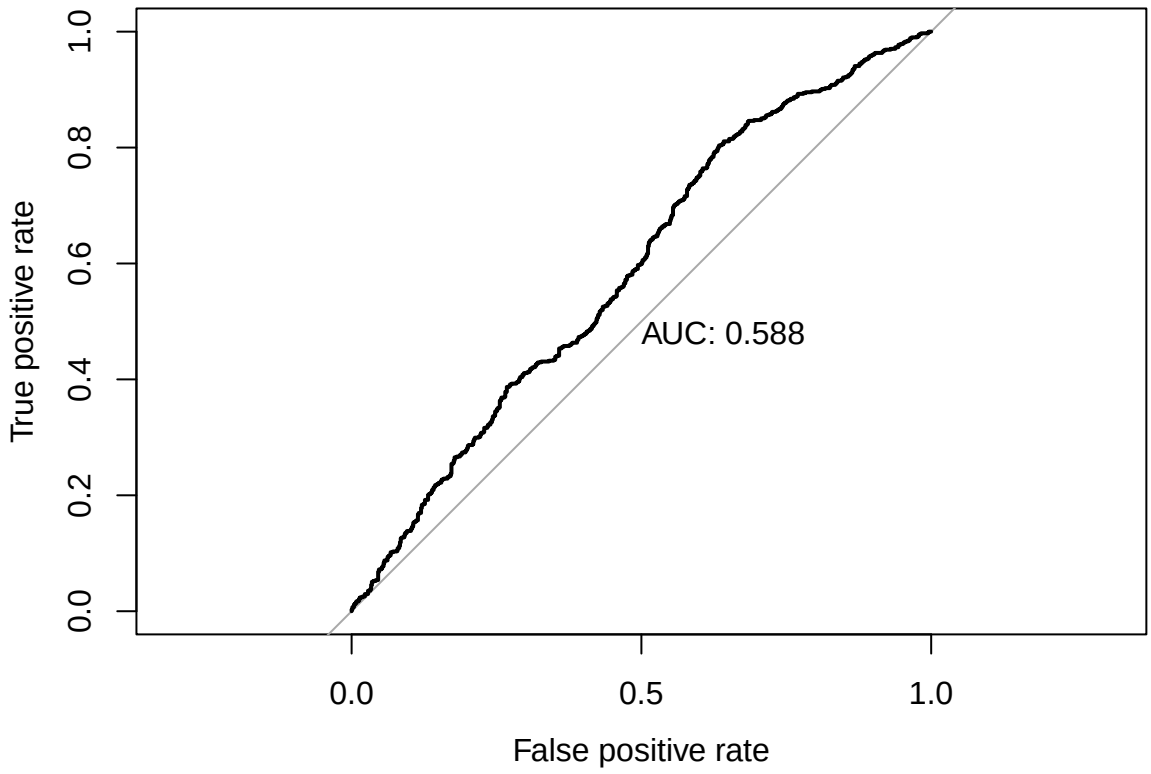
The second model contains the results without the holiday variable, in the following models the variables "solar_exposure", logarithm of "rainfall", "mean_temperature" and finally the seasons were removed consecutively. After removing the variable "mean_temperature", the deviance begins to have relatively significant changes, that is why the final model is the logit.out4 that contains: "demand", "mean_temperature", "school_day", and "season".

In the following plot it is possible to see the AUC of the ROC to understand how much the discarding of the other variables has affected the model.

```
logistic.prob <- predict(logit.out4, type="response")
roc.out <- roc(train$high_RRP, logistic.prob, levels=c(FALSE, TRUE))
```

```
## Setting direction: controls < cases
```

```
plot(roc.out, print.auc=TRUE, legacy.axes=TRUE, xlab="False positive rate", ylab="True positive rate")
```




#### 4.2.1 Logistic Regression Using Months

As an alternative, we try to see if we can increase the performance of the model by using the months variables instead of the seasons, repeating the same procedure above but skipping some steps.

We start by using all the variables.

```
logit.out1 <- glm(high_RRP ~ demand + mean_temperature + solar_exposure + log(rainfall +1) + holiday + s

summary(logit.out1)
```

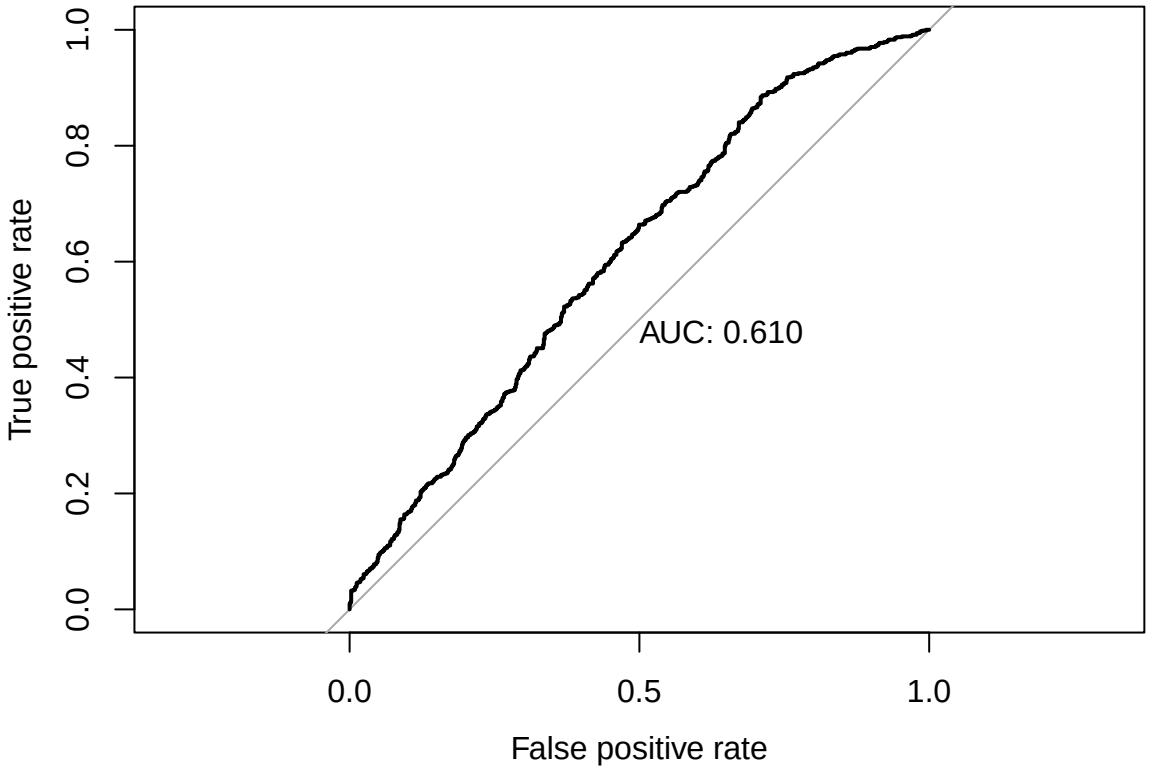
```
##
## Call:
## glm(formula = high_RRP ~ demand + mean_temperature + solar_exposure +
##     log(rainfall + 1) + holiday + school_day + month, family = binomial,
##     data = train)
```

```
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5420  -1.1329  -0.7893   1.1720   1.7766
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)       -2.231e+00  7.408e-01  -3.012   0.0026 **
## demand             1.073e-05  5.113e-06   2.098   0.0359 *
## mean_temperature   1.426e-02  1.953e-02   0.730   0.4652
## solar_exposure     1.563e-02  1.095e-02   1.428   0.1533
## log(rainfall + 1) -1.037e-01  7.556e-02  -1.373   0.1699
## holidayTRUE       -3.652e-01  3.198e-01  -1.142   0.2534
## school_dayTRUE     6.131e-01  1.456e-01   4.210 2.55e-05 ***
## month2            -5.002e-01  2.998e-01  -1.668   0.0952 .
## month3            -3.348e-01  2.975e-01  -1.125   0.2604
## month4             2.961e-01  3.138e-01   0.944   0.3454
## month5            -1.050e-02  3.583e-01  -0.029   0.9766
## month6             6.463e-01  3.990e-01   1.620   0.1053
## month7             5.623e-01  3.935e-01   1.429   0.1530
## month8            -7.092e-02  3.785e-01  -0.187   0.8514
## month9             3.713e-01  3.348e-01   1.109   0.2674
## month10            9.165e-02  3.072e-01   0.298   0.7654
## month11            8.853e-02  3.115e-01   0.284   0.7762
## month12           -2.438e-01  2.911e-01  -0.838   0.4023
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2010.7  on 1450  degrees of freedom
## Residual deviance: 1950.0  on 1433  degrees of freedom
## AIC: 1986
##
## Number of Fisher Scoring iterations: 4
```

```
logistic.prob <- predict(logit.out1, type="response")
roc.out <- roc(train$high_RRP, logistic.prob, levels=c(FALSE, TRUE))
```

```
## Setting direction: controls < cases
```

```
plot(roc.out, print.auc=TRUE, legacy.axes=TRUE, xlab="False positive rate", ylab="True positive rate")
```




As before, we proceed to remove the feature that had the highest P-value, which in this case is holiday.

```
logit.out2 <- glm(high_RRP ~ demand + mean_temperature + school_day + month, data = train, family = bin

summary(logit.out2)
```

```
##
## Call:
## glm(formula = high_RRP ~ demand + mean_temperature + school_day +
##     month, family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5345  -1.1239  -0.8293   1.1771   1.6511
##
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -2.173e+00  6.556e-01  -3.314  0.00092 ***
## demand            1.161e-05  4.975e-06   2.334  0.01958 *
## mean_temperature  2.107e-02  1.896e-02   1.111  0.26641
## school_dayTRUE    6.243e-01  1.447e-01   4.314 1.61e-05 ***
## month2           -5.406e-01  2.947e-01  -1.835  0.06658 .
## month3           -4.275e-01  2.820e-01  -1.516  0.12952
## month4            1.070e-01  2.788e-01   0.384  0.70103
## month5           -2.061e-01  3.115e-01  -0.662  0.50825
## month6            4.228e-01  3.529e-01   1.198  0.23084
```

```
## month7            3.715e-01  3.480e-01   1.068  0.28570
## month8           -2.376e-01  3.446e-01  -0.689  0.49055
## month9            2.484e-01  3.134e-01   0.792  0.42814
## month10           6.598e-02  2.985e-01   0.221  0.82507
## month11           4.710e-02  3.080e-01   0.153  0.87847
## month12          -2.258e-01  2.895e-01  -0.780  0.43533
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2010.7  on 1450  degrees of freedom
## Residual deviance: 1956.6  on 1436  degrees of freedom
## AIC: 1986.6
##
## Number of Fisher Scoring iterations: 4
```

```
logit.out3 <- glm(high_RRP ~ demand + mean_temperature + school_day, data = train, family = binomial)

summary(logit.out3)
```

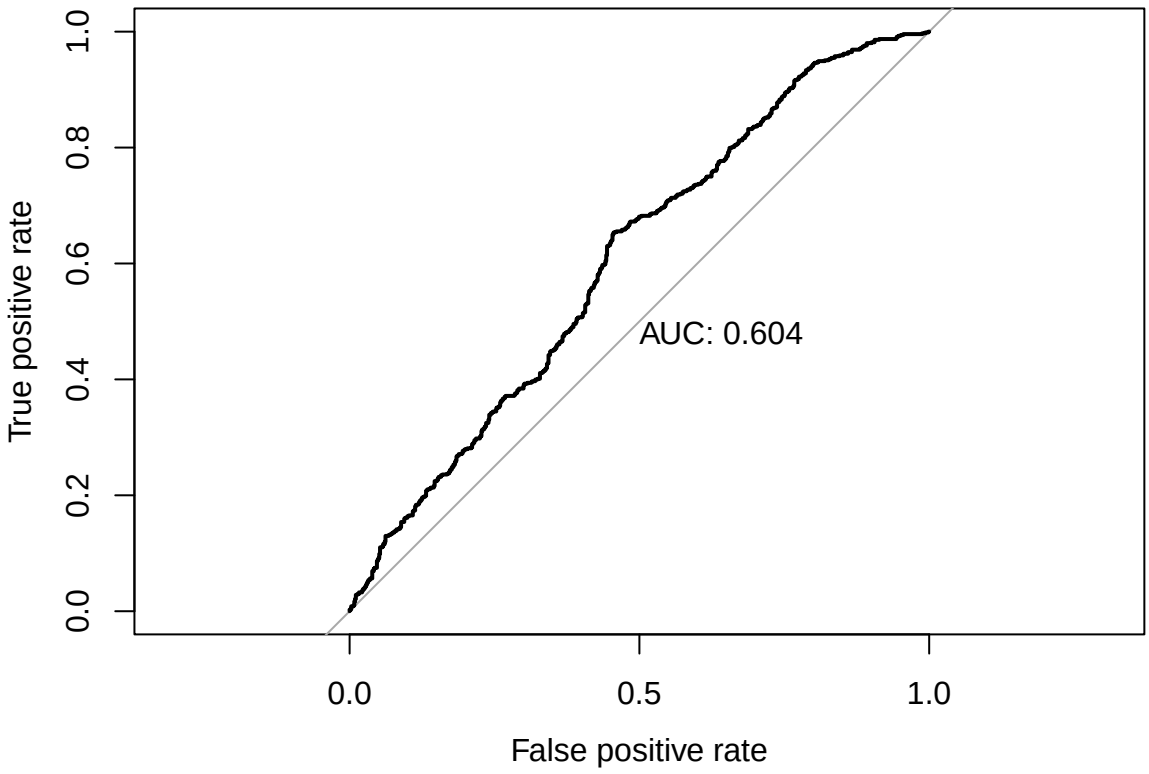
```
##
## Call:
## glm(formula = high_RRP ~ demand + mean_temperature + school_day,
##     family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.4440  -1.1611  -0.8911   1.1719   1.5135
##
## Coefficients:
##                     Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -1.963e+00  5.611e-01  -3.498 0.000468 ***
## demand            1.448e-05  4.144e-06   3.496 0.000473 ***
## mean_temperature -8.235e-03  1.152e-02  -0.715 0.474544
## school_dayTRUE    4.468e-01  1.175e-01   3.804 0.000143 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2010.7  on 1450  degrees of freedom
## Residual deviance: 1977.3  on 1447  degrees of freedom
## AIC: 1985.3
##
## Number of Fisher Scoring iterations: 4
```

It is possible to see that the general result is better than using the seasons. It is also noticeable how when changing the second model, removing the months, the deviance drops considerably. Because of that, we keep the second model (logit.out2). Finally, we make the ROC plot to see how much the area under the curve has changed with respect to the first model.

```
logistic.prob <- predict(logit.out2, type="response")
roc.out <- roc(train$high_RRP, logistic.prob, levels=c(FALSE, TRUE))
```

```
## Setting direction: controls < cases
```

```
plot(roc.out, print.auc=TRUE, legacy.axes=TRUE, xlab="False positive rate", ylab="True positive rate")
```




# 5 Model Evaluation

## 5.1 Linear Regression Evaluation

To evaluate the performance of our models, we will first get the MSE (Mean Squared Errors) of the test set by doing a simple test-train-split. We assign the 80 percent of the data to the training set and the rest of the data to the test set.

```
###split the data into train and test and see their error(MSE)

set.seed(1)
n = dim(data)[1] #number of samples
ntrain = floor(dim(data)[1]*0.8)  #1678 samples for test(80 percent of the data)

train <- sample(1:n, size=ntrain, replace= FALSE) #index of training samples
```

```
# linear model with training data
mod.tot0.train <- lm(demand ~ poly(solar_exposure, 4) +
                       poly(mean_temperature, 4) + season +
                       school_day , data=data , subset=train)

#predict the demand of the test data based on fitted model
y.pred <- predict(mod.tot0.train, newdata=data[-train, ])

# mean squared error on test data
MSE.val <- mean((data$demand[-train]-y.pred)^2)
MSE.val
```

```
## [1] 80388633
```

Then also we use a 10-fold cross validation to have the better evaluation of our model and its error based on training data with 10 different parts of the data. For training the data, we split the data to 10 folds. Each time we choose one of these folds as the test set and other nine parts as training set. After training the data and computing the MSE errors, we get an average of the errors achieved from 10 different test sets. At the end, the average amount of MSE error would be the best evaluation of the model's error.

```
# k-fold cross-validation

set.seed(1)
k=10
folds <- sample(1:k, nrow(data), replace=TRUE)
#table(folds)

cv.errors <- matrix(NA,k,1)
colnames(cv.errors) <- 1


for(j in 1:k){
  best.fit <- lm(demand ~ poly(solar_exposure, 4) + poly(mean_temperature, 4) +
                   season + school_day, data=data[folds!=j,])
  test.mat <- model.matrix(demand ~ poly(solar_exposure, 4) +
                             poly(mean_temperature, 4) + season +
                             school_day, data=data[folds==j,])
  y.pred <- predict(mod.tot0.train, newdata=data[folds==j,])
  cv.errors[j,1] <- mean( (data$demand[folds==j]-y.pred)^2)

}

mean.cv.errors <- apply(cv.errors,2,mean)
mean.cv.errors
```

```
##        1
## 88238532
```

The error values obtained by cross validation are less than the ones obtained by calculating the MSE, but not by much.

```
##    cv.errors
```

```
## 1   82540383
## 2   84387844
## 3   94685089
## 4   89769280
## 5   92233213
## 6   85484240
## 7   96069217
## 8   81960217
## 9   91930535
## 10  83325300
```

### 5.2 Logistic Regression Evaluation

In this part, the performance of the logistic regression model selected among those tested (second model using months) will be evaluated. The matrix evaluated in the test-set (623 samples) will be printed using the best threshold

```
th <- coords(roc.out, "best")
probabilities <- logit.out2 %>% predict(test, type = "response")


trivial.pred <- rep(FALSE, 623)
logistic.pred <- rep(FALSE, 623)

logistic.pred[probabilities>th[1,1]] <- TRUE

CM <- table(logistic.pred, test$high_RRP)
CM
```

```
##
## logistic.pred FALSE TRUE
##         FALSE   160  126
##         TRUE    134  203
```

From the confusion matrix, an accuracy close to 56%, a specificity of 45.69%, and a sensitivity of 66%.

## 6 Conclusion

Based on the insights gained during the EDA part about the behavior of the different variables that have impact on RRP and demand, we build a regression model to predict the demand and determined that the best results are obtained by considering the "solar_exposure", "mean_temperature", "seasons" and "school_days" variables. As we saw, the behavior of the regression models had non linear correlation with two variables, "solar_exposure" and "mean_temperature". Furthermore, while performing the hypothesis test on variances in the EDA section, we could not find evidence that the variances were equal, because the data was very unbalanced. Thus, we could not use the t-test to check equality of the means. We couldn't find any other tests on the mean of the demand on different categories or any kind of ANOVA/ANCOVA test for seasons.

Regarding logistic regression, the model that obtained the best results used the variables: "demand", "mean_temperature", "school_day", "month", and reported an area under the ROC curve equal to 0.608. The best threshold for this method was 0.43 and it is possible to see that the accuracy is quite low (56%),

although the overall performance is quite low and should not be considered a good predictor of the RRP(due to the distribution of the data), the The model appears to have a decent sensitivity (66%), as reported in the confusion matrix in the previous section.