Predicting Monthly Changes in Commercial and Residential Electricity Consumption in the United States

Anubhav Nikunj Singh Sachan

November 1, 2018

Introduction

According to data from the US Energy Information Administration (EIA) in 2017 around 63% of US electricity was generated using fossil fuels, 20% was generated using nuclear energy, and only about 17% was generated from renewable energy sources [1]. Since electricity is a fundamental necessity in modern times and its current production depends heavily on non-renewable resources, it is vital to have a good administration of the resources necessary to generate it. To achieve an adequate management of resources, it is necessary to know and predict the behavior of consumers. Producing less electricity than what is needed can have negative effects on industry and people's lifestyle, but producing too much electricity is not a sustainable practice in the long term. In addition to this, since several of the non-renewable resources used to produce electricity are also needed for other purposes, their mismanagement can have several harmful repercussions.

In this project we will be analyzing data published by EIA and accessed using the following URL:

https://www.eia.gov/electricity/data/browser/#/topic/5?geo=g&agg=0,1&endsec=vg

The dataset used will have information about monthly electricity usage from January 2002 until September 2018.

We will be interested in trying to fit a linear regression model that can predict the monthly changes in residential and commercial consumption of electricity in the United States. The consumption of electricity will be measured in the volume of retail sales of electricity. The reason for modelling monthly change in electricity consumption and not monthly electricity consumption will be explained in the next section.

After reorganizing the data, the dataset we will work with has the following variables:

- Date: A monthly period variable with a month and a year (i.e. Jan-06, Feb-17)
- Month: A categorical variable indicating the month of the observation. For example Jan-06 corresponds to January.
- Year: A numerical variable indicating the year of the observation. For example Jan-06 corresponds to 2006.
- Time_Index: A numerical variable indicating the number of months elapsed since January 2002. For example February 2002 has a value of 1, March 2002 has a value of 2, etc.
- ResidentialSales: A numerical variable measured in million KWh that indicates the volume of electricity sold during the respective month for the residential sector.
- CommercialSales: A numerical variable measured in million KWh that indicates the volume of electricity sold during the respective month for the residential sector.
- ChangeResidentialSales: The change in the volume of electricity sold in comparison to the previous month for the residential sector. For example if for the residential sector in February 2002 there were 96593 KWh sold and in January 2002 there were 116892 KWh sold, then ChangeResidentialSales for February 2002 is 96593 - 116892 = -20299
- ChangeCommercialSales: The change in the volume of electricity sold in comparison to the previous month for the commercial sector. For example if for the commercial sector in February 2002 there were 80951 KWh sold and in January 2002 there were 87429 KWh sold, then ChangeResidentialSales for February 2002 is 80951 - 87429 = -6478

Our dataset has 201 datapoints in total (one for each month ranging January 2002 - September 2018). Our response variables will be ChangeResidentialSales and ChangeCommercialSales.

Our main question of interest is: **Can we predict the monthly changes in the consumption of electrical energy based on the available variables using linear regression?**

Exploratory Analysis

Before carrying out any regression analysis, we will carry out some exploratory analysis of the data.

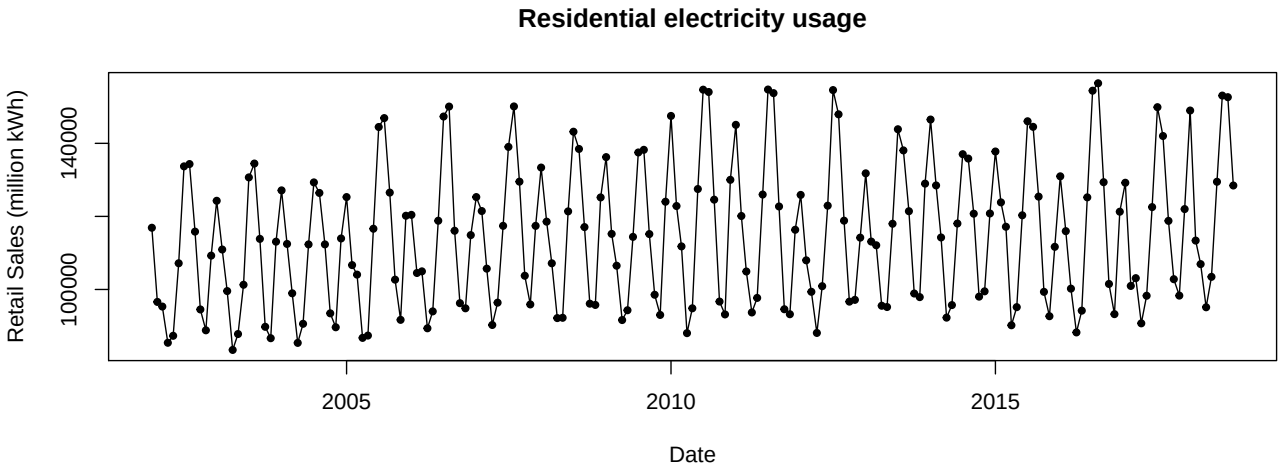
We first read and print out the first few observations of our dataset using the following code:

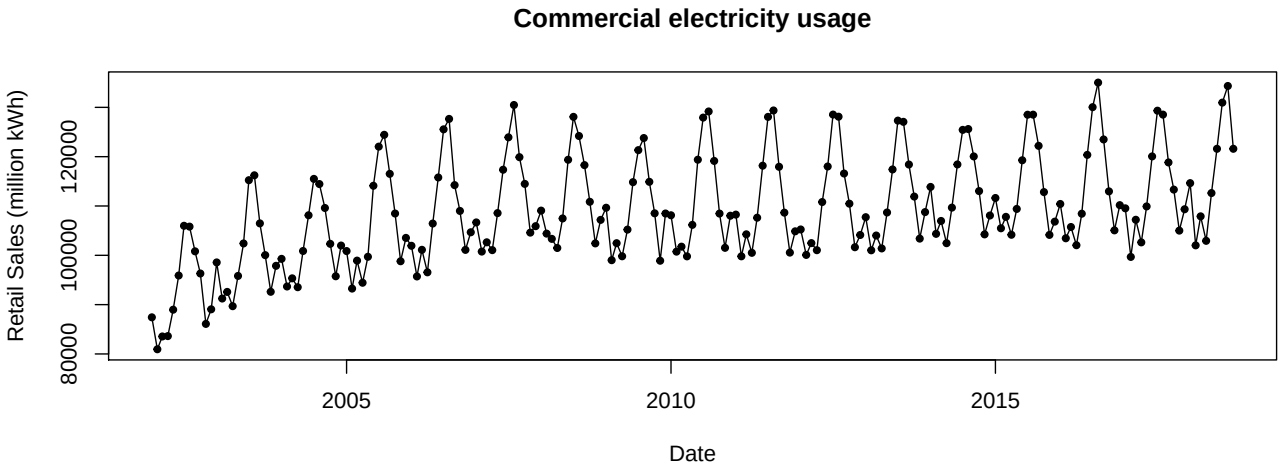
```
data = read.csv("Dataset2.csv")
data$Date = as.Date(paste0("01-", data$Date), format = "%d-%b-%y")
data$Month = factor(months.Date(data$Date), levels = c("January", "February", "March", "April",
    "May", "June", "July", "August", "September", "October", "November", "December"))
data$Year = as.numeric(format(data$Date, "%Y"))
head(data)
```

```
##         Date Time_Index ResidentialSales ChangeResidentialSales
## 1 2002-01-01          0            116892                  20708
## 2 2002-02-01          1             96593                 -20299
## 3 2002-03-01          2             95319                  -1274
## 4 2002-04-01          3             85408                  -9911
## 5 2002-05-01          4             87319                   1911
## 6 2002-06-01          5            107170                  19851
##   CommercialSales ChangeCommercialSales    Month Year
## 1           87429                  1993  January 2002
## 2           80951                 -6478 February 2002
## 3           83545                  2594    March 2002
## 4           83619                    74    April 2002
## 5           88974                  5355      May 2002
## 6           95903                  6929     June 2002
```

We now proceed to make a plot of of electricity consumption over time from January 2002 to September 2018.

```
par(mfrow = c(2, 1))
plot(data$Date, data$ResidentialSales, xlab = "Date", ylab = "Retail Sales (million kWh)",
    main = "Residential electricity usage", type = "o", pch = 20)
plot(data$Date, data$CommercialSales, xlab = "Date", ylab = "Retail Sales (million kWh)", main = "Commercial e
    type = "o", pch = 20)
```





From the plot made above there seems to be significant periodicity. There might be some trend to but it is hard to see from just looking at the above graph. For residential electricity consumption there might be some trend, but it is hard to spot from the graph. For commercial usage there seems to be a somewhat logarithmic trend.

To see the general trends more clearly we use the following code to make plots of yearly electricity consumption per sector for 2002-2017:

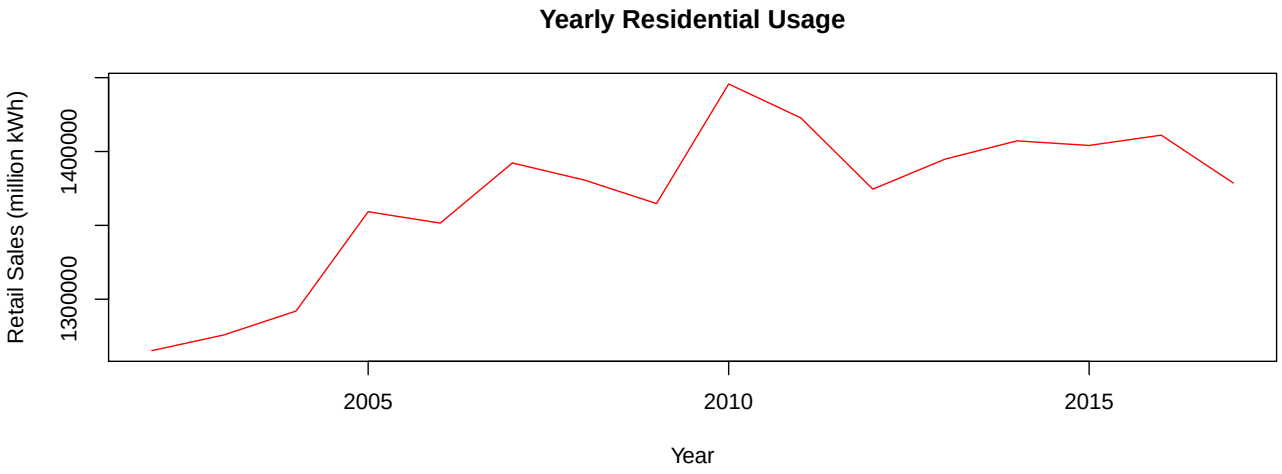
```
# Calculating yearly consumption for each sector

Residential_use = rep(0, (2017 - 2002 + 1))
Commercial_use = rep(0, (2017 - 2002 + 1))

for (i in 1:NROW(data)) {
    Year = as.numeric(format(data$Date, "%Y"))[i]
    if (Year < 2018) {
        Residential_use[(Year - 2002 + 1)] = Residential_use[(Year - 2002 + 1)] + data$ResidentialSales[i]
        Commercial_use[(Year - 2002 + 1)] = Commercial_use[(Year - 2002 + 1)] + data$CommercialSales[i]
    }

}

Residential_use
```

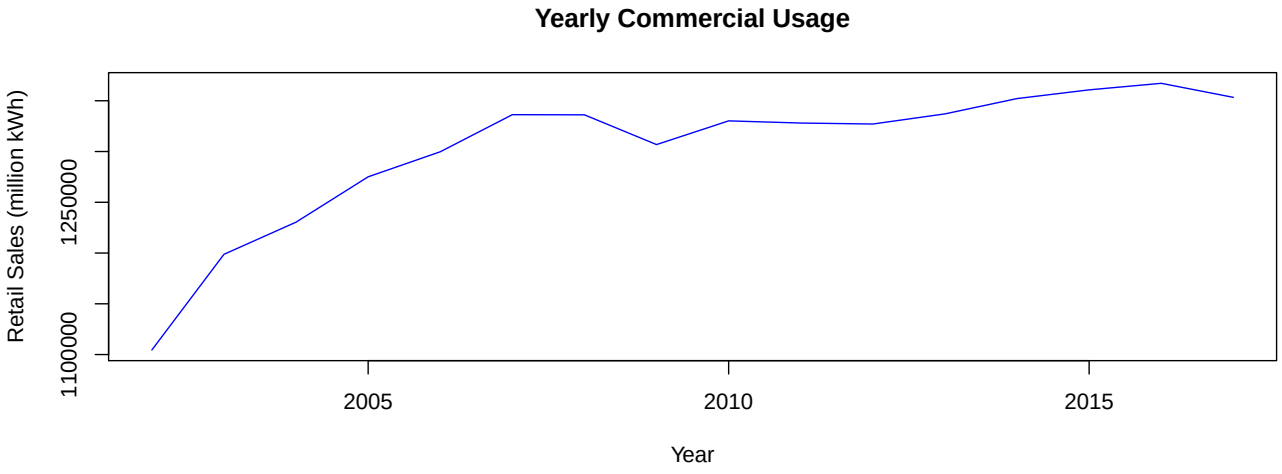
```
##  [1] 1265180 1275824 1291980 1359227 1351519 1392241 1380661 1364757
##  [9] 1445708 1422801 1374513 1394813 1407209 1404097 1411058 1378648
```

```
Commercial_use
```

```
##  [1] 1104497 1198728 1230423 1275079 1299743 1336314 1336134 1306853
##  [9] 1330199 1328057 1327099 1337079 1352159 1360750 1367191 1353358
```

```
##################################################

# Plotting yearly consumption

par(mfrow = c(2, 1))

Year = 2002:2017
plot(Year, Residential_use, type = "l", col = "red", main = "Yearly Residential Usage", xlab = "Year",
    ylab = "Retail Sales (million kWh)")
plot(Year, Commercial_use, type = "l", col = "blue", main = "Yearly Commercial Usage", xlab = "Year",
    ylab = "Retail Sales (million kWh)")
```



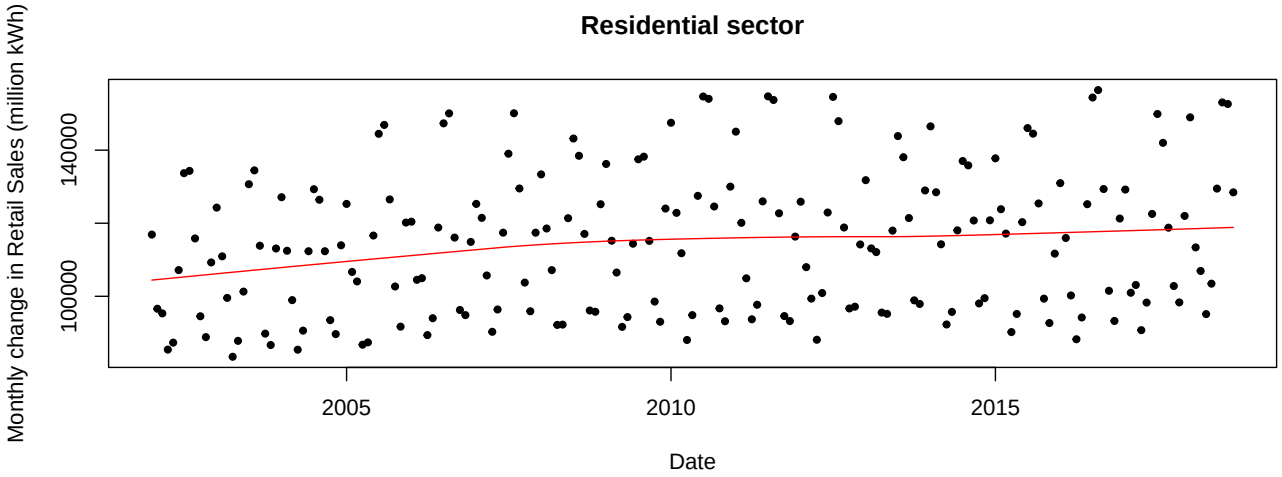


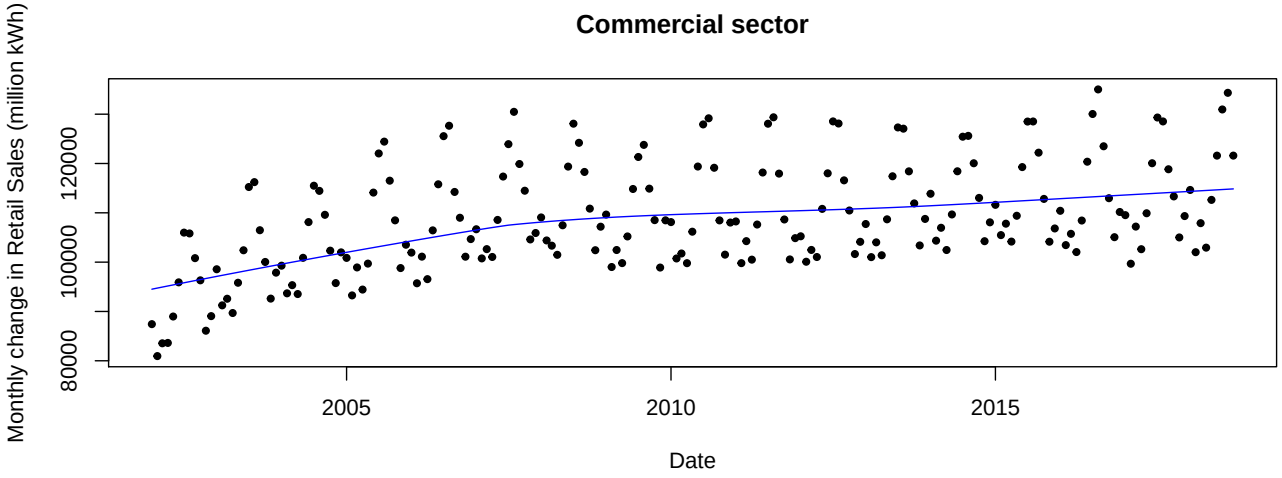
From these plots there seems to be an overall increasing trend for both sectors, though it is definitely not linear. It is interesting to see how years there there are increases and decreases in electricity usage seem to coincide in both sectors (though the intensity of increase or decrease seems to be different for each sector). It is also interesting to see how electricity consumption decreased between 2008-2009 which corresponds to the recent economic recession. There is also a decrease in electricity usage between 2010-2012 and between 2016-2017 which might be worth investigating.

We can also see this trend by making plots with Lowess lines:

```
par(mfrow = c(2, 1))
plot(data$Date, data$ResidentialSales, xlab = "Date", ylab = "Monthly change in Retail Sales (million kWh)",
```

```
    main = "Residential sector", pch = 20)
lines(lowess(data$Date, data$ResidentialSales), col = "red")
plot(data$Date, data$CommercialSales, xlab = "Date", ylab = "Monthly change in Retail Sales (million kWh)",
    main = "Commercial sector", pch = 20)
lines(lowess(data$Date, data$CommercialSales), col = "blue")
```





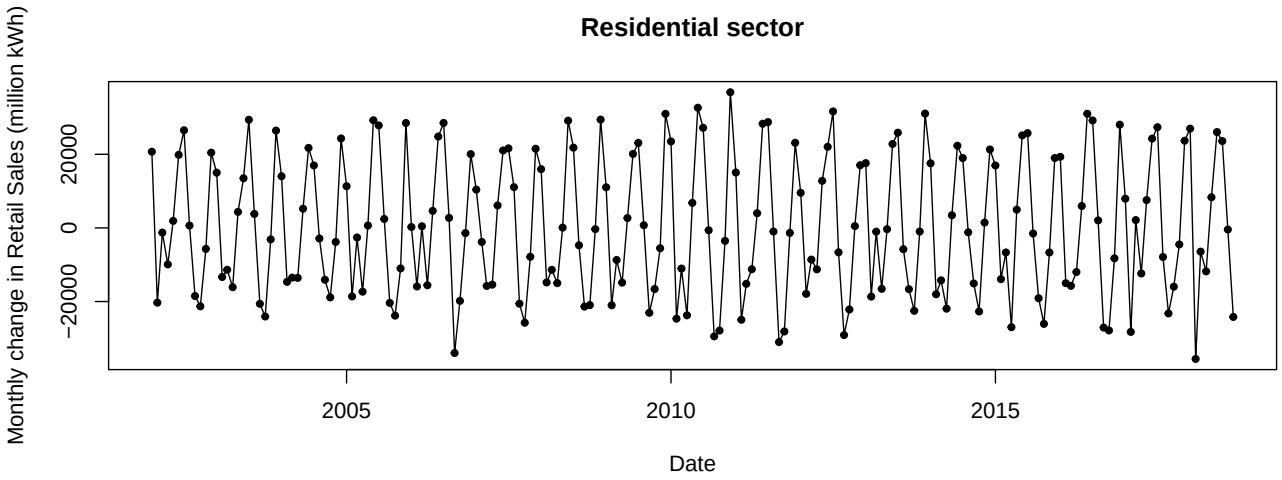
Originally for this project we had planned to use a model of the form:

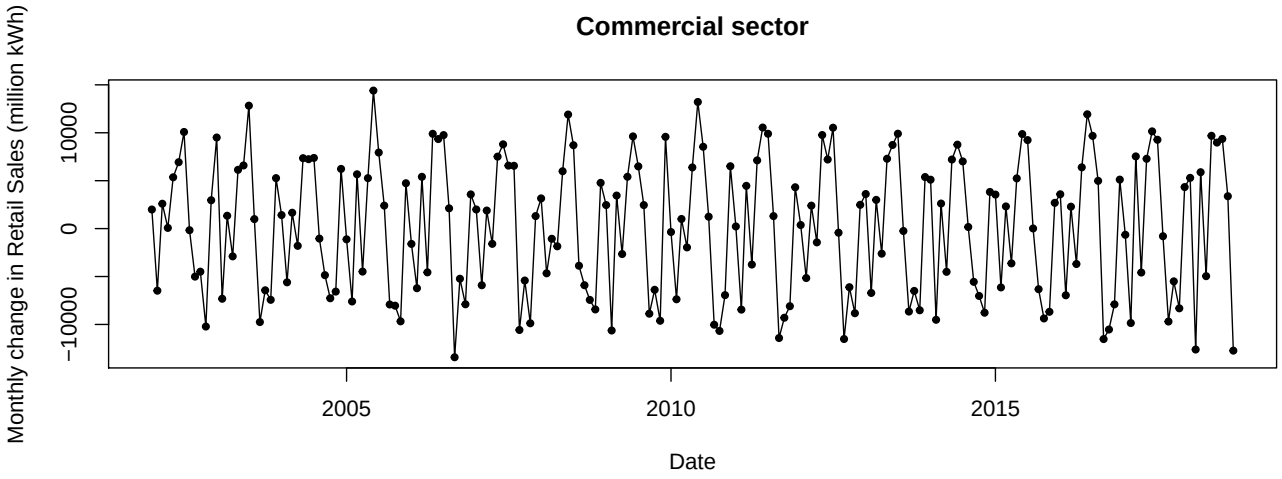
$$Y_t = \alpha_0 + \alpha_1 g(t) + \alpha_2 I_t\{February\} + \dots \alpha_{12} I_t\{December\} + \epsilon_t$$

where Y_t represented the electricity consumption at time-point t, g(t) modeled the trend and $I_t(w)$ was an indicator function that is 1 if timepoint t corresponds to month w and 0 otherwise. Nevertheless, it is hard to figure out what the real form of g is. Furthermore, it also seems to be that for the last year (2017) the electricity consumption decreased and so making predictions assuming a generally increasing trend might be fallacious.

We therefore decided to see if modelling monthly change in electricity consumption might be less challenging in this aspect. To do so, we first make a plot of of monthly change in electricity consumption over time from January 2002 to September 2018:

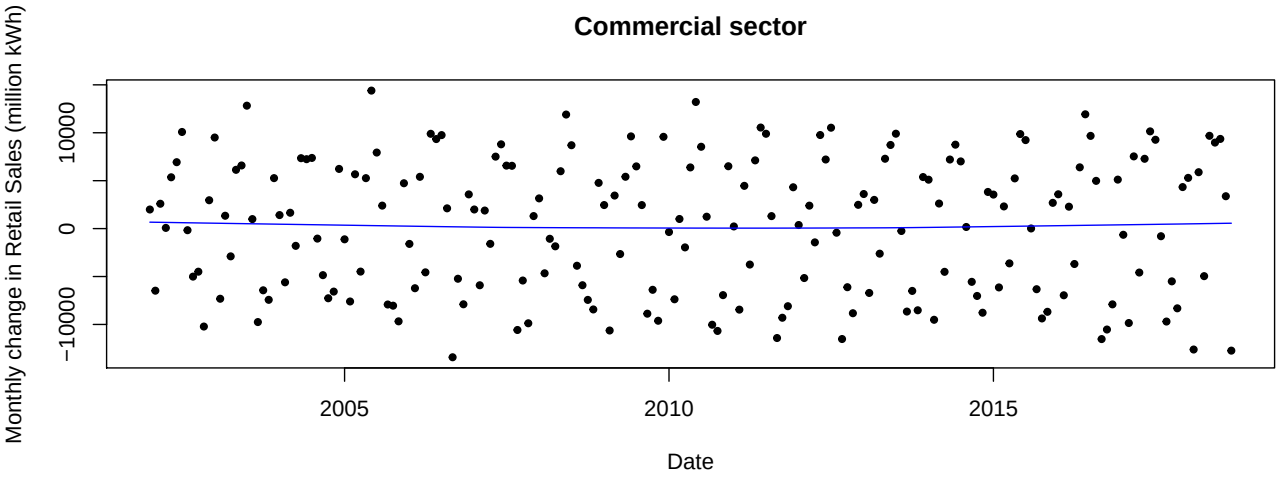
```
par(mfrow = c(2, 1))
plot(data$Date, data$ChangeResidentialSales, xlab = "Date", ylab = "Monthly change in Retail Sales (million kW
    main = "Residential sector", type = "o", pch = 20)
plot(data$Date, data$ChangeCommercialSales, xlab = "Date", ylab = "Monthly change in Retail Sales (million kWh
    main = "Commercial sector", type = "o", pch = 20)
```





From these plots there doesn't seem to be any noticeably trend for the monthly change in electricity consumption per sector. We can also see this by making a plot of the data with a Lowess line:

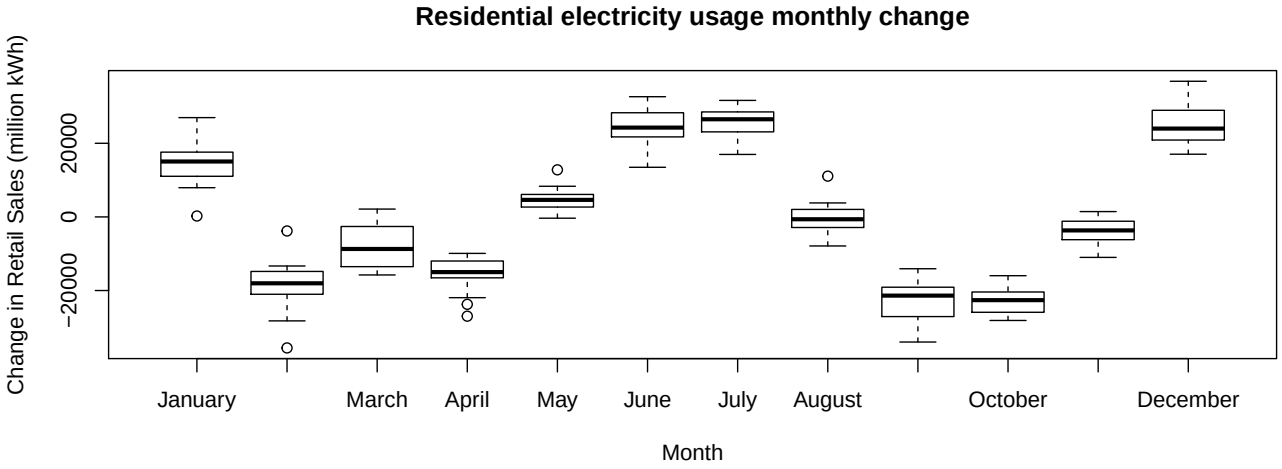
```
par(mfrow = c(2, 1))
plot(data$Date, data$ChangeResidentialSales, xlab = "Date", ylab = "Monthly change in Retail Sales (million kW
    main = "Residential sector", pch = 20)
lines(lowess(data$Date, data$ChangeResidentialSales), col = "red")
plot(data$Date, data$ChangeCommercialSales, xlab = "Date", ylab = "Monthly change in Retail Sales (million kWh
    main = "Commercial sector", pch = 20)
lines(lowess(data$Date, data$ChangeCommercialSales), col = "blue")
```

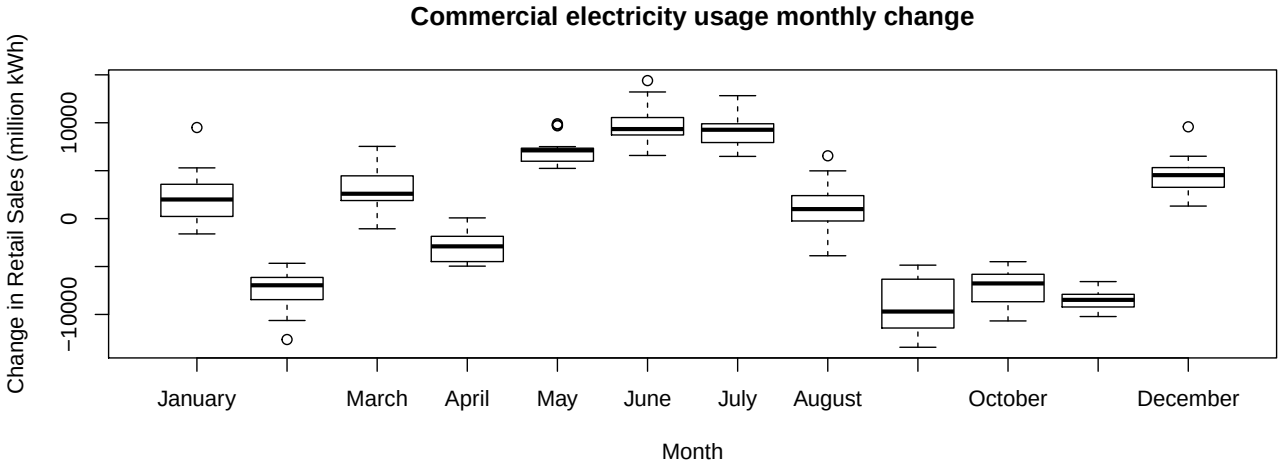


The Lowess lines seem to be horizontal in both cases, indicating that we don't have the same issue with trend as before.

We now illustrate the periodicity of the data by making boxplots:

```
par(mfrow = c(2, 1))
boxplot(data$ChangeResidentialSales ~ data$Month, xlab = "Month", ylab = "Change in Retail Sales (million kWh)
    main = "Residential electricity usage monthly change")
boxplot(data$ChangeCommercialSales ~ data$Month, xlab = "Month", ylab = "Change in Retail Sales (million kWh)"
    main = "Commercial electricity usage monthly change")
```





From these plots it can be seen that in both sectors the highest monthly increase in electricity usage happens during the summer months (July and August) and winter months (December and January). Also for both sectors the biggest monthly decreases in electricity usage happen in February, September and October.

These box plots also show us that even though we can remove trend by modelling monthly change in electricity consumption instead of monthly consumption, there is still periodicity. To deal with the periodicity we will try to fit a model of the form:

$$\Delta Y_t = \beta_0 + \beta_1 I_t\{February\} + \ldots + \beta_{11}\ I_t\{December\} + \epsilon_t$$

where ΔY_t represents the monthly change in electricity consumption. Also it is worth noticing that for residential sector ΔY_t approximately varies between -35000 and 35000, the whereas for the residential sector it varies between -15000 and 15000.

Originally we planned to incorporate all the data into a single regression model.However after trying out various methods to incorporate the data and different types of models, we couldn't find a model where the variances of the residuals were homogenous or approximately normally distributed. To tackle this issue, a better initial approach seemed to be to fit two regression models separately - one for residential electricity usage and another one for commercial electricity usage. This in part helped alleviate the issue because it allowed for us to have two estimates for the variance of the residuals (one estimate per sector). In the following 2 sections we will present our regression analysis for the two sectors.

Regression Analysis for the Residential Sector

We start out by dividing our dataset into a training set and a testing set. The training set consists of all observations from January 2002 until September 2016, whereas the testing set consists of all observations from October 2016 until September 2018.

```
data.train = data[data$Time_Index < (201 - 24), ]
data.test = data[data$Time_Index >= (201 - 24), ]
```

We will use the training set to fit various initial models and the testing set to analyze the fit of these models and try to decided which one might be the most appropriate for making predictions.

We first consider a model of the form:

$$\Delta Y_t = \beta_0 + \beta_1 I_t\{February\} + \ldots + \beta_{11}\ I_t\{December\} + \epsilon_t$$

Here ΔY_t is the electricity consumption for the sector of interest at the timepoint t. We also assume that for the sector of interest the ϵ_t are independent and identically distributed as normal distributions with mean 0 (and constant variance). Also, as mentioned before, $I_t(w)$ was an indicator function that is 1 if timepoint t corresponds to month w and 0 otherwise.

We now fit the model using R:

```
model1_residential = lm(ChangeResidentialSales ~ Month, data.train)
summary(model1_residential)
```

```
##
## Call:
## lm(formula = ChangeResidentialSales ~ Month, data = data.train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -14305  -3215    403   2980  13221
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)       14547       1247  11.662  < 2e-16 ***
## MonthFebruary    -31590       1764 -17.907  < 2e-16 ***
## MonthMarch       -23661       1764 -13.413  < 2e-16 ***
## MonthApril       -30643       1764 -17.371  < 2e-16 ***
## MonthMay         -10334       1764  -5.858 2.47e-08 ***
## MonthJune          9694       1764   5.495 1.45e-07 ***
## MonthJuly         10996       1764   6.233 3.68e-09 ***
## MonthAugust      -14611       1764  -8.282 3.95e-14 ***
## MonthSeptember   -37222       1764 -21.100  < 2e-16 ***
## MonthOctober     -37456       1795 -20.863  < 2e-16 ***
## MonthNovember    -18074       1795 -10.068  < 2e-16 ***
## MonthDecember     10452       1795   5.822 2.96e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4831 on 165 degrees of freedom
## Multiple R-squared:  0.9351, Adjusted R-squared:  0.9308
## F-statistic: 216.2 on 11 and 165 DF,  p-value: < 2.2e-16
```

In order to verify if the assumptions of our model are met, we make the following plots:

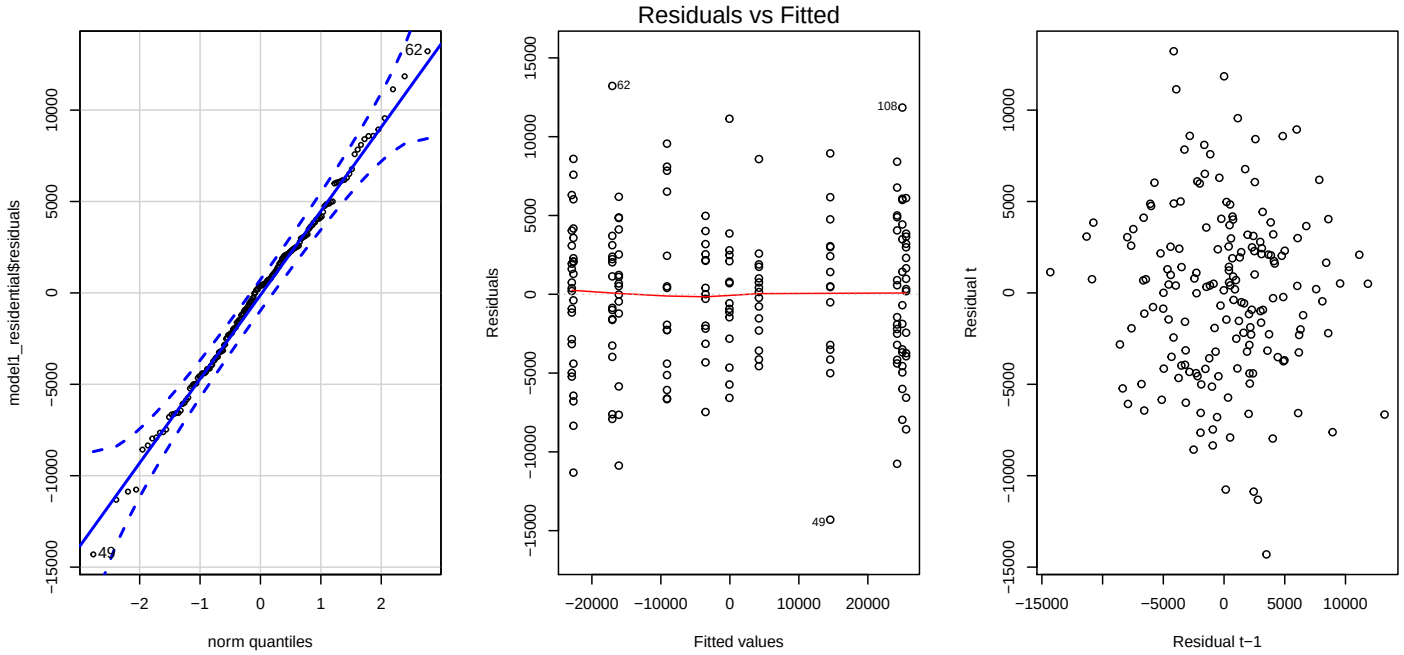
```
n = NROW(data.train)

library(car)
```

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

```
par(mfrow = c(1, 3))
qqPlot(model1_residential$residuals)
```

```
## [1] 49 62
```

```
plot(model1_residential, 1)
plot(model1_residential$residuals[-n], model1_residential$residuals[-1], xlab = "Residual t-1",
    ylab = "Residual t")
```



From the first plot it seems like the residuals are indeed normally distributed. From the second plot it seems to be that all residuals are homoscedastic and that their mean is 0. Finally from the third plot, there doesn't seem to be any indication of serial correlation between the residuals.

To verify that there is indeed no serial correlation, we run a regression between ϵ_t as a function of ϵ_{t-1}:

```
summary(lm(model1_residential$residuals[-1] ~ model1_residential$residuals[-n] - 1))
```

```
##
## Call:
## lm(formula = model1_residential$residuals[-1] ~ model1_residential$residuals[-n] -
##     1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14135.3  -3261.0    249.2   2636.5  13019.1
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## model1_residential$residuals[-n] -0.04864    0.07532  -0.646    0.519
##
## Residual standard error: 4662 on 175 degrees of freedom
## Multiple R-squared:  0.002377,   Adjusted R-squared:  -0.003323
## F-statistic: 0.417 on 1 and 175 DF,  p-value: 0.5193
```

Here the coefficient is shown to have a p-value of 0.519 which is quite high and thus indicates that there is no evidence for serial correlation.

Therefore, it would seem like the underlying assumptions for our model to be valid for making inferences is met.

We now try to see if we can group some months into a single variable. To do so, we will calculate the mean monthly change in electricity consumption for every month, sort it and then make groups of 2 months, 3 months, 4 months and 6 months.

```
sort(tapply(data$ChangeResidentialSales, data$Month, mean))
```

```
##   September     October    February       April       March    November
## -22796.0588 -22784.0625 -18792.8235 -15625.0000  -8295.6471  -3883.3750
##      August         May     January        June    December        July
##   -546.2353   4651.6471  14888.8824  24347.4706  25107.0625  25534.8235
```

For model2_residential we will group the months as follows: {{September October} {February April} {March November} {August May} {January June} {December July}}

For model3_residential we will group the months as follows: {{September October February} {April March November} {August May January} {June December July}}

For model4_residential we will group the months as follows: {{September October February April} {March November August May} {January June December July}}

For model6_residential we will group the months as follows: {{September October February April March November} {August May January June December July}}

Notice that there is no model5_residential. This is because to make it easier to keep track of the models, we are using the naming "modeli_residential" to indicate that the model is assuming groups of i months.

We will also consider an alternative model7_residential which is based on grouping variables based on a visual analysis of the box-plot made in the previous section. In this case we will group the months as {{September October February April} {March November} {August May} {January} {June December July}}

We now define and implement these models in R. We also calculate the AIC, Adjusted R squared and MSE (mean squared error).

```
# Create levels
ordering = order(c(9, 10, 2, 4, 3, 11, 8, 5, 1, 6, 12, 7))

levelsm1 = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)
levelsm2 = c(1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6)[ordering]
levelsm3 = c(1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4)[ordering]
levelsm4 = c(1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3)[ordering]
levelsm6 = c(1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2)[ordering]
levelsm7 = c(1, 1, 1, 1, 2, 2, 3, 3, 4, 5, 5, 5)[ordering]

LevelMatrix = rbind(levelsm1, levelsm2, levelsm3, levelsm4, levelsm6, levelsm7)

## Residential sales
ResidentialModelNames = c("model1_residential", "model2_residential", "model3_residential",
    "model4_residential", "model6_residential", "model7_residential")
ResidentialModels = list()
AIC.Residential.train = c()
R2.Residential.train = c()
adjR2.Residential.train = c()
MSE.Residential.train = c()

for (i in 1:NROW(LevelMatrix)) {
    MonthGrouped = data.train$Month
    levels(MonthGrouped) = LevelMatrix[i, ]
    fitted.model = lm(data.train$ChangeResidentialSales ~ MonthGrouped)

    summ = summary(fitted.model)

    ResidentialModels = c(ResidentialModels, list(fitted.model))
    AIC.Residential.train = c(AIC.Residential.train, AIC(fitted.model))
    R2.Residential.train = c(R2.Residential.train, summ$r.squared)
    adjR2.Residential.train = c(adjR2.Residential.train, summ$adj.r.squared)
    MSE.Residential.train = c(MSE.Residential.train, mean(summ$residuals^2))
```

```
}

### We print out each of the criterion
ResidentialModelNames
```

```
## [1] "model1_residential" "model2_residential" "model3_residential"
## [4] "model4_residential" "model6_residential" "model7_residential"
```

```
AIC.Residential.train
```

```
## [1] 3518.803 3550.461 3609.471 3608.401 3767.028 3543.953
```

```
R2.Residential.train
```

```
## [1] 0.9351223 0.9169734 0.8814720 0.8808478 0.7047261 0.9190614
```

```
adjR2.Residential.train
```

```
## [1] 0.9307971 0.9145457 0.8794166 0.8794783 0.7030389 0.9171791
```

```
MSE.Residential.train
```

```
## [1] 21757575 27844030 39749879 39959191 99023839 27143795
```

From this it can be seen that the model with highest R squared and lowest mean squared error is the full model (model1_residential). This is expected for the training set because it has more variables and the other models are just nested versions of it. The full model also has the lowest AIC and highest adjusted R squared, though models "model2_residential" and "model7_residential" also seem to perform well for these criteria. Due to the small number of datapoints we decide not to perform any outlier analysis.

We now fit the models that were generated into the testing set and calculate the values of the R squared and mean squared error.

```
# Create levels
ordering = order(c(9, 10, 2, 4, 3, 11, 8, 5, 1, 6, 12, 7))

levelsm1 = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)
levelsm2 = c(1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6)[ordering]
levelsm3 = c(1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4)[ordering]
levelsm4 = c(1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3)[ordering]
levelsm6 = c(1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2)[ordering]
levelsm7 = c(1, 1, 1, 1, 2, 2, 3, 3, 4, 5, 5, 5)[ordering]

LevelMatrix = rbind(levelsm1, levelsm2, levelsm3, levelsm4, levelsm6, levelsm7)

## Residential sales
ResidentialModelNames = c("model1_residential", "model2_residential", "model3_residential",
    "model4_residential", "model6_residential", "model7_residential")
R2.Residential.test = c()
MSE.Residential.test = c()

for (i in 1:NROW(LevelMatrix)) {
    MonthGrouped = data.test$Month
    levels(MonthGrouped) = LevelMatrix[i, ]
    fitted.model = ResidentialModels[[i]]

    predicted.values = predict.lm(ResidentialModels[[i]], MonthGrouped)

    MSE.Residential.test = c(MSE.Residential.test, mean((data.test$ChangeResidentialSales -
        predicted.values)^2))

    SSE = sum((data.test$ChangeResidentialSales - predicted.values)^2)
    SST = sum((data.test$ChangeResidentialSales - mean(data.test$ChangeResidentialSales))^2)

    R2.Residential.test = c(R2.Residential.test, (1 - SSE/SST))
```

```
}

### We print out each of the criterion
ResidentialModelNames
```

```
## [1] "model1_residential" "model2_residential" "model3_residential"
## [4] "model4_residential" "model6_residential" "model7_residential"
```

```
R2.Residential.test
```

```
## [1] 0.8907272 0.8803895 0.8681316 0.8758581 0.6676359 0.8919308
```

```
MSE.Residential.test
```

```
## [1]  43454402  47565364  52439956  49367347 132170859  42975755
```

In this case we can see that the model with the lowest mean squared error and highest R squared is model7_residential. The full model (model1_residential) also comes close, but there seems to be evidence that there was more overfitting for the full model than model7_residential. We also see that model7_residential performed well in the training set and so we decide that the most appropriate model for our problem will be this one.

We now print out a summary for this model:

```
model7_residential = ResidentialModels[[6]]
summary(model7_residential)
```

```
##
## Call:
## lm(formula = data.train$ChangeResidentialSales ~ MonthGrouped)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14366.1  -3434.1    459.9   3879.4  15803.9
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)      14547       1365  10.660  < 2e-16 ***
## MonthGrouped1   -34173       1528 -22.361  < 2e-16 ***
## MonthGrouped2   -20964       1681 -12.472  < 2e-16 ***
## MonthGrouped3   -12472       1671  -7.463 4.03e-12 ***
## MonthGrouped5    10379       1580   6.568 5.81e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5285 on 172 degrees of freedom
## Multiple R-squared:  0.9191, Adjusted R-squared:  0.9172
## F-statistic: 488.3 on 4 and 172 DF,  p-value: < 2.2e-16
```

Therefore our fitted model will have the following form:

$$\Delta Y_t = \beta_0 + \beta_1 I_t\{1\} + \beta_2 I_t\{2\} + \beta_3 I_t\{3\} + \beta_5 I_t\{5\} + \epsilon_t$$

Where the $I_t(w)$ take the following values: $I_t(1)$: 1 if $Month_t \in \{September, October, February, April\}$. 0 otherwise. $I_t(2)$: 1 if $Month_t \in \{March, November\}$. 0 otherwise. $I_t(3)$: 1 if $Month_t \in \{August, May\}$. 0 otherwise. $I_t(5)$: 1 if $Month_t \in \{June, December, July\}$. 0 otherwise.

From our summary, we have that the least square estimates have the following values:

$$\hat{\beta_0} = 14547$$

$$\hat{\beta_1} = -34173$$

$$\hat{\beta_2} = -20964$$

$$\hat{\beta_3} = -12472$$

$$\hat{\beta}_5 = 10379$$

We can interpret these the following way:

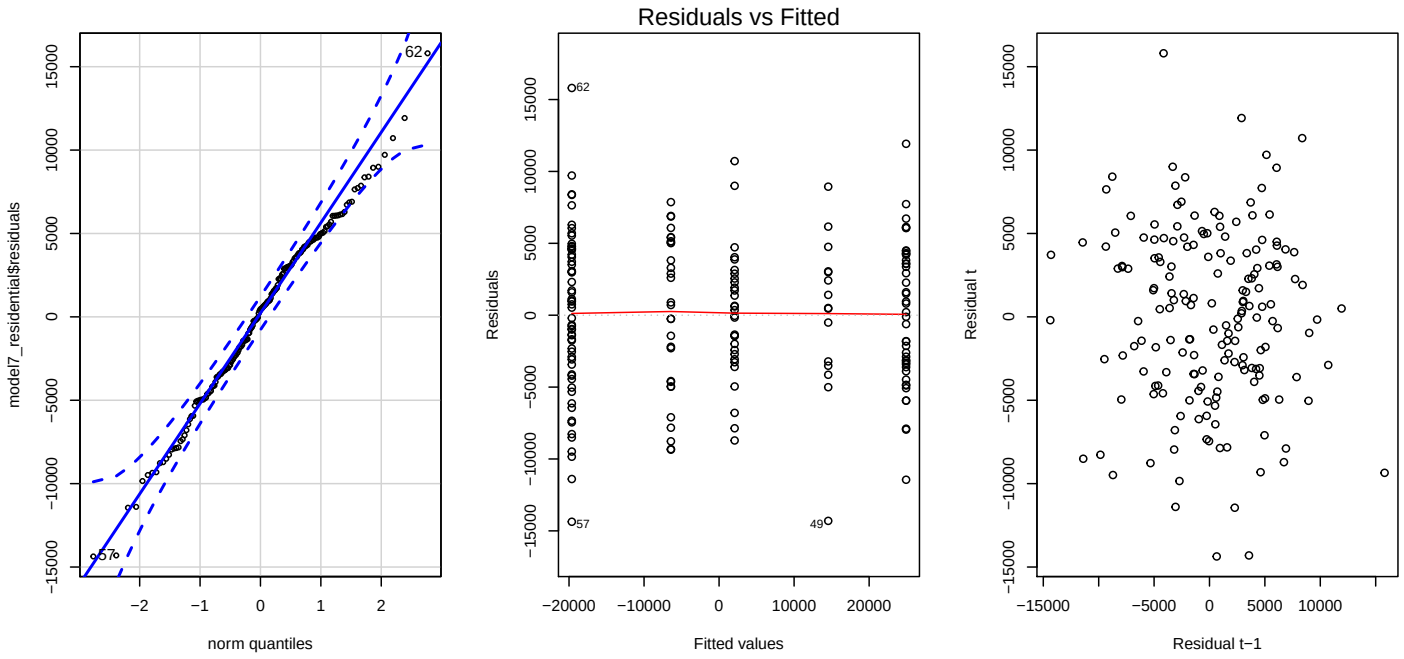
- During the month of January there is an average increase of 14547 units in ΔY_t.
- During the months of September,October, February and April there is an average decrease of 14547 - 34173 = -19626 units in ΔY_t.
- During the months of March and November there is an average decrease of 14547 - 20964 = -6417 units in ΔY_t.
- During the months of August and May there is an average increase of 14547 - 12472 = 2075 units in ΔY_t.
- During the months of June, December and July there is an average increase of 14547 + 10379 =24926 units in ΔY_t

Now we see whether the assumptions are met for this model by making the following plots:

```
n = NROW(data.train)

library(car)
par(mfrow = c(1, 3))
qqPlot(model7_residential$residuals)
```

```
## [1] 62 57
```

```
plot(model7_residential, 1)
plot(model7_residential$residuals[-n], model7_residential$residuals[-1], xlab = "Residual t-1",
    ylab = "Residual t")
```



From the first plot it seems like the residuals are indeed normally distributed. From the second plot it seems to be that all residuals are homoscedastic and that their mean is 0. Finally from the third plot, there doesn't seem to be any indication of serial correlation between the residuals.

To verify that there is indeed no serial correlation, we run a regression between ϵ_t as a function of ϵ_{t-1}:

```
summary(lm(model7_residential$residuals[-1] ~ model7_residential$residuals[-n] - 1))
```

```
##
## Call:
## lm(formula = model7_residential$residuals[-1] ~ model7_residential$residuals[-n] -
##     1)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -14328.3  -3520.0    389.9   3816.9  15563.7
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## model7_residential$residuals[-n] -0.05798    0.07561  -0.767    0.444
##
## Residual standard error: 5210 on 175 degrees of freedom
## Multiple R-squared:  0.00335,    Adjusted R-squared:  -0.002346
## F-statistic: 0.5882 on 1 and 175 DF,  p-value: 0.4442
```

Here the coefficient is shown to have a p-value of 0.444 which is quite high and thus indicates that there is no evidence for serial correlation.

Therefore, it would seem like the underlying assumptions for our model to be valid for making inferences is met.

We now proceed to calculate 95 percent confidence intervals for each of our coefficients:

```
confint(model7_residential, level = 0.95)
```

```
##                     2.5 %     97.5 %
## (Intercept)    11853.573  17240.694
## MonthGrouped1 -37189.674 -31156.491
## MonthGrouped2 -24281.859 -17646.201
## MonthGrouped3 -15771.424  -9173.576
## MonthGrouped5   7260.015  13498.173
```

This means that if we were to gather additional data and recalculate these confidence intervals in a similar way, then the true values of the coefficient would lie in the confidence intervals 95 percent of the time. Also, since 0 is not contained in any of these confidence intervals, we can reject the null hypothesis $H_0 : \beta_i = 0$ at a level $\alpha = 0.05$ for each of the coefficients.

We now calculate fitted values for ΔY_t for January - December.

```
ordering = order(c(9, 10, 2, 4, 3, 11, 8, 5, 1, 6, 12, 7))
levelsm7 = c(1, 1, 1, 1, 2, 2, 3, 3, 4, 5, 5, 5)[ordering]

MonthGrouped = data.train$Month[1:12]
levels(MonthGrouped) = levelsm7

predicted.values = predict.lm(model7_residential, MonthGrouped)
predicted.values
```

```
##          1          2          3          4          5          6
##  14547.133 -19625.949  -6416.897 -19625.949   2074.633  24926.227
##          7          8          9         10         11         12
##  24926.227   2074.633 -19625.949 -19625.949  -6416.897  24926.227
```

So for January the fitted value for ΔY_t is 14547.133, for February it is -19625.949, and so on. This means that for example in January we would expect residential electricity consumption to increase on average by 14547.133 KWh when compared to the previous month.

We now calculate the 95 percent confidence interval for the mean responses for January-December:

```
ordering = order(c(9, 10, 2, 4, 3, 11, 8, 5, 1, 6, 12, 7))
levelsm7 = c(1, 1, 1, 1, 2, 2, 3, 3, 4, 5, 5, 5)[ordering]

MonthGrouped = data.train$Month[1:12]
levels(MonthGrouped) = levelsm7

predict.lm(model7_residential, MonthGrouped, interval = "confidence", level = 0.95)
```

```
##           fit         lwr        upr
## 1   14547.133  11853.5731  17240.694
## 2  -19625.949 -20984.0947 -18267.804
## 3   -6416.897  -8354.0915  -4479.702
## 4  -19625.949 -20984.0947 -18267.804
```

```
## 5    2074.633    169.9986   3979.268
## 6   24926.227  23353.5269  26498.928
## 7   24926.227  23353.5269  26498.928
## 8    2074.633    169.9986   3979.268
## 9  -19625.949 -20984.0947 -18267.804
## 10 -19625.949 -20984.0947 -18267.804
## 11  -6416.897  -8354.0915  -4479.702
## 12  24926.227  23353.5269  26498.928
```

We do the same for the responses:

```
ordering = order(c(9, 10, 2, 4, 3, 11, 8, 5, 1, 6, 12, 7))
levelsm7 = c(1, 1, 1, 1, 2, 2, 3, 3, 4, 5, 5, 5)[ordering]

MonthGrouped = data.train$Month[1:12]
levels(MonthGrouped) = levelsm7

predict.lm(model7_residential, MonthGrouped, interval = "prediction", level = 0.95)
```

```
##           fit        lwr       upr
## 1   14547.133   3772.892 25321.374
## 2  -19625.949 -30146.099 -9105.799
## 3   -6416.897 -17027.350  4193.557
## 4  -19625.949 -30146.099 -9105.799
## 5    2074.633  -8529.924 12679.191
## 6   24926.227  14376.233 35476.222
## 7   24926.227  14376.233 35476.222
## 8    2074.633  -8529.924 12679.191
## 9  -19625.949 -30146.099 -9105.799
## 10 -19625.949 -30146.099 -9105.799
## 11  -6416.897 -17027.350  4193.557
## 12  24926.227  14376.233 35476.222
```

Note that the confidence interval for the mean response is shorter than the confidence interval for the response. This is because as the name says, the confidence interval for the mean response corresponds to the **average** ΔY_t whereas the confidence interval for the response corresponds to ΔY_t

Finally, it might be useful to make forecasts for monthly residential electricity consumption Y_t instead of monthly change in residential electricity consumption ΔY_t. From our results we can derive a method to do so.

Assume that the last observed electricity consumption is Y_s and we are interested in the electricity consumption $Y_{s+\tau}$. We can see that:

$$Y_{s+\tau} = Y_s + \sum_{k=s+1}^{s+\tau} \Delta Y_k$$

We can treat Y_s as a constant, and under the assumptions of our model we have that

$$\Delta Y_k \sim Normal(E[\Delta Y_k], \sigma^2)$$

Here $E[\Delta Y_k]$ corresponds to the fitted value at timepoint k.

We will get that under the assumptions of our model:

$$Y_{s+\tau} \sim Normal(Y_s + \sum_{k=s+1}^{s+\tau} E[\Delta Y_k], \tau\sigma^2)$$

For our predictions we will substitute σ with $\hat{\sigma} = 5285$. We now proceed to make predictions for our testing test. We will take $Y_s = 129363$ (corresponding to the residential sales in the last timepoint of the training set). We will do pointwise inference and calculate a 95 percent confidence band for our testing set. We will make a plot to compare this to the real values from the testing set.

```r
ordering = order(c(9, 10, 2, 4, 3, 11, 8, 5, 1, 6, 12, 7))
levelsm7 = c(1, 1, 1, 1, 2, 2, 3, 3, 4, 5, 5, 5)[ordering]

MonthGrouped = data.test$Month
levels(MonthGrouped) = levelsm7

predicted.values = predict.lm(model7_residential, MonthGrouped)
sigmas = sqrt(((summ$sigma)^2) * (1:NROW(data.test)))

LowerBounds = c()
ExpectedValues = c()
UpperBounds = c()

for (i in 1:NROW(data.test)) {
    LowerBounds[i] = 129363 + sum(predicted.values[1:i]) - 1.96 * sigmas[i]
    ExpectedValues[i] = 129363 + sum(predicted.values[1:i])
    UpperBounds[i] = 129363 + sum(predicted.values[1:i]) + 1.96 * sigmas[i]
}


LowerBounds
```

```
##  [1]  99378.15  88670.45 110304.24 122075.71 100004.35  91376.64  69717.64
##  [8]  69899.95  93048.87 116294.08 116769.85  95616.20  74524.99  66698.18
## [15]  90264.01 103495.39  82594.21  74939.06  54108.80  55010.42  78792.62
## [22] 102601.74 103584.38  82889.93
```

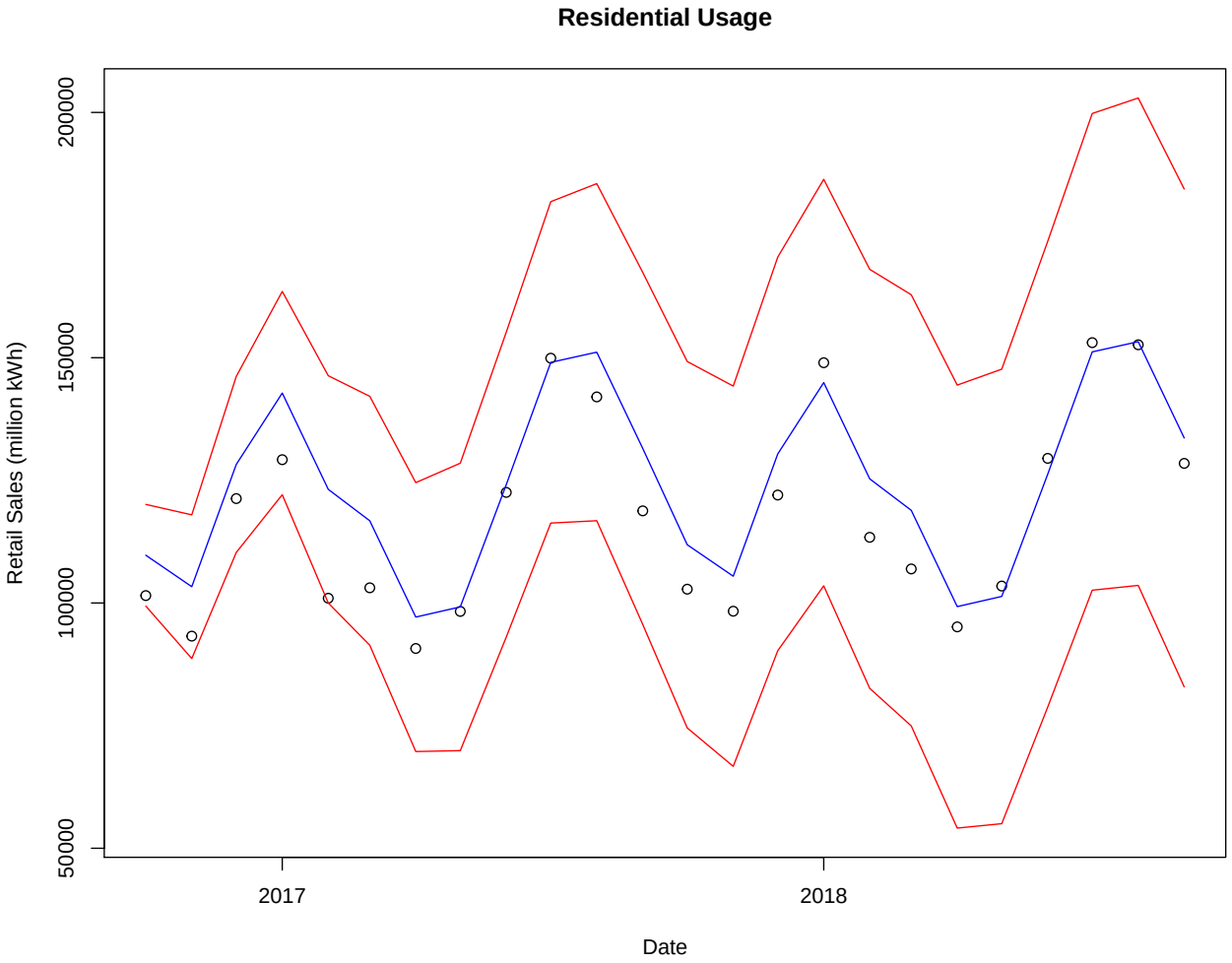
```r
ExpectedValues
```

```
##  [1] 109737.05 103320.15 128246.38 142793.51 123167.57 116750.67  97124.72
##  [8]  99199.35 124125.58 149051.81 151126.44 131500.49 111874.54 105457.65
## [15] 130383.87 144931.01 125305.06 118888.16  99262.21 101336.85 126263.07
## [22] 151189.30 153263.93 133637.98
```

```r
UpperBounds
```

```
##  [1] 120096.0 117969.9 146188.5 163511.3 146330.8 142124.7 124531.8
##  [8] 128498.8 155202.3 181809.5 185483.0 167384.8 149224.1 144217.1
## [15] 170503.7 186366.6 168015.9 162837.3 144415.6 147663.3 173733.5
## [22] 199776.9 202943.5 184386.0
```

```r
par(mfrow = c(1, 1))
plot(data.test$Date, data.test$ResidentialSales, col = "black", main = "Residential Usage",
    xlab = "Date", ylab = "Retail Sales (million kWh)", ylim = c(min(LowerBounds), max(UpperBounds)))
lines(data.test$Date, ExpectedValues, col = "blue")
lines(data.test$Date, LowerBounds, col = "red")
lines(data.test$Date, UpperBounds, col = "red")
```



Here the points represent the actual values, the blue line represents the predicted consumption and and red lines represent the 95 percent confidence bounds.

It is a good sign that all the observations seem to lie within the expected confidence band. Nevertheless it seems to be like our point-wise predictions tend to in general be higher than the actual observations. This might be worth investigating in a future project.

Finally, we decide to use our model to make predictions for the period of October 2018 - September 2019. Here we take $Y_s = 128458$ which corresponds to the observation for September 2018.

```
ordering = order(c(9, 10, 2, 4, 3, 11, 8, 5, 1, 6, 12, 7))
levelsm7 = c(1, 1, 1, 1, 2, 2, 3, 3, 4, 5, 5, 5)[ordering]

MonthGrouped = data.test$Month[1:12]
levels(MonthGrouped) = levelsm7

predicted.values = predict.lm(model7_residential, MonthGrouped)
sigmas = sqrt(((summ$sigma)^2) * (1:12))

LowerBounds = c()
ExpectedValues = c()
UpperBounds = c()

for (i in 1:NROW(data.test$Month[1:12])) {
    LowerBounds[i] = 128458 + sum(predicted.values[1:i]) - 1.96 * sigmas[i]
    ExpectedValues[i] = 128458 + sum(predicted.values[1:i])
    UpperBounds[i] = 128458 + sum(predicted.values[1:i]) + 1.96 * sigmas[i]
}
```

```
LowerBounds
```

```
##  [1]  98473.15  87765.45 109399.24 121170.71  99099.35  90471.64  68812.64
##  [8]  68994.95  92143.87 115389.08 115864.85  94711.20
```

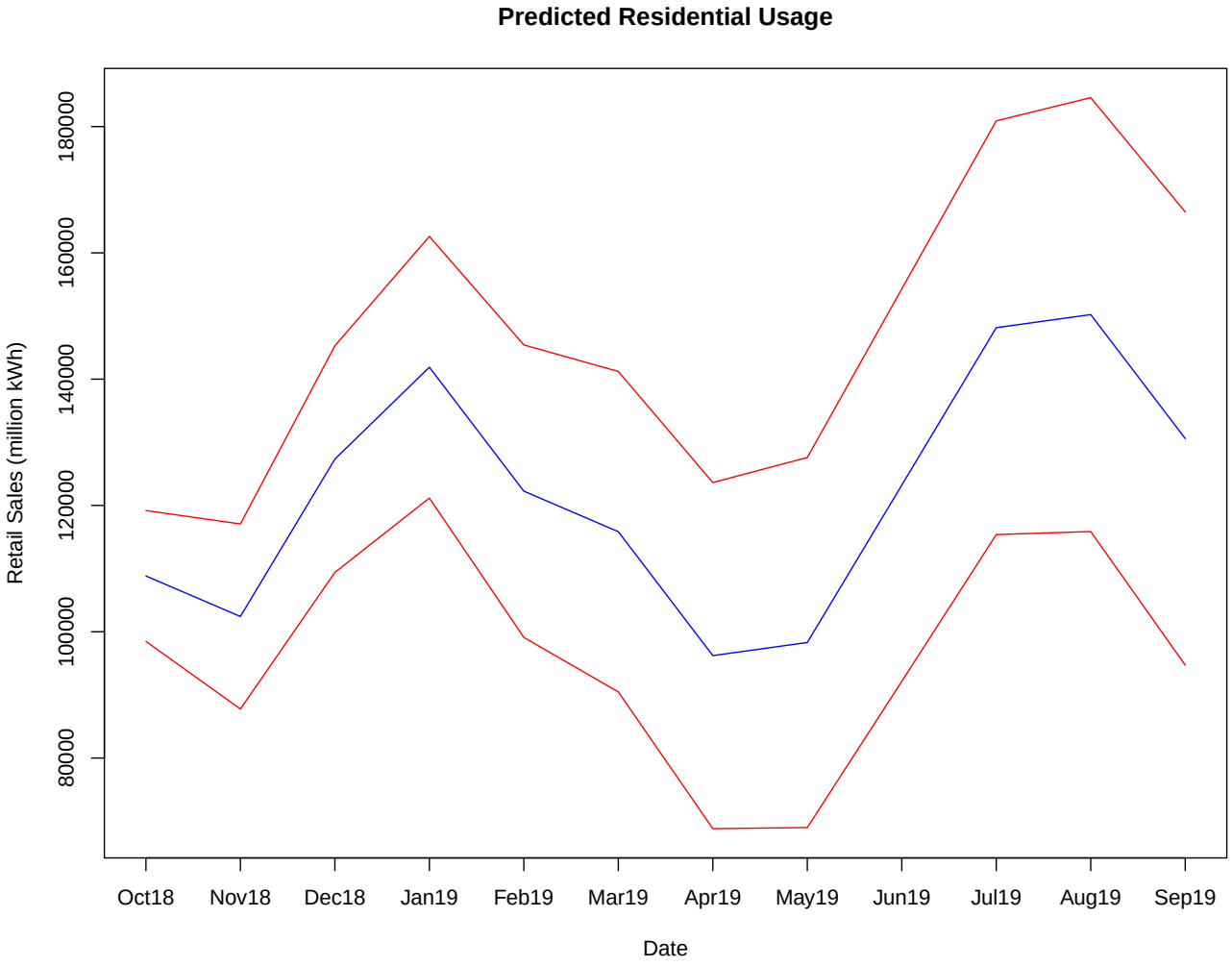
```
ExpectedValues
```

```
##  [1] 108832.05 102415.15 127341.38 141888.51 122262.57 115845.67  96219.72
##  [8]  98294.35 123220.58 148146.81 150221.44 130595.49
```

```
UpperBounds
```

```
##  [1] 119191.0 117064.9 145283.5 162606.3 145425.8 141219.7 123626.8
##  [8] 127593.8 154297.3 180904.5 184578.0 166479.8
```

```
Period = c("Oct18", "Nov18", "Dec18", "Jan19", "Feb19", "Mar19", "Apr19", "May19", "Jun19",
    "Jul19", "Aug19", "Sep19")


par(mfrow = c(1, 1))
plot(ExpectedValues, type = "l", col = "blue", main = "Predicted Residential Usage", xlab = "Date",
    ylab = "Retail Sales (million kWh)", ylim = c(min(LowerBounds), max(UpperBounds)), xaxt = "n")
axis(1, at = 1:12, labels = Period)
lines(LowerBounds, col = "red")
lines(UpperBounds, col = "red")
```



Regression Analysis for the Commercial Sector

We start out by dividing our dataset into a training set and a testing set. The training set consists of all observations from January 2002 until September 2016, whereas the testing set consists of all observations from October 2016 until September 2018.

```
data.train = data[data$Time_Index < (201 - 24), ]
data.test = data[data$Time_Index >= (201 - 24), ]
```

As before we will use the training set to fit various initial models and the testing set to analyze the fit of these models and try to decided which one might be the most appropriate for making predictions.

We first consider a model of the form:

$$\Delta Y_t = \beta_0 + \beta_2 I_t\{February\} + \ldots + \beta_{12}\ I_t\{December\} + \epsilon_t$$

Here ΔY_t is the electricity consumption for the sector of interest at the timepoint t. We also assume that for the sector of interest the ϵ_t are independent and identically distributed as normal distributions with mean 0 (and constant variance). Also, as mentioned before, $I_t(w)$ was an indicator function that is 1 if timepoint t corresponds to month w and 0 otherwise.

We now fit the model using R:

```
model1_commercial = lm(ChangeCommercialSales ~ Month, data.train)
summary(model1_commercial)
```

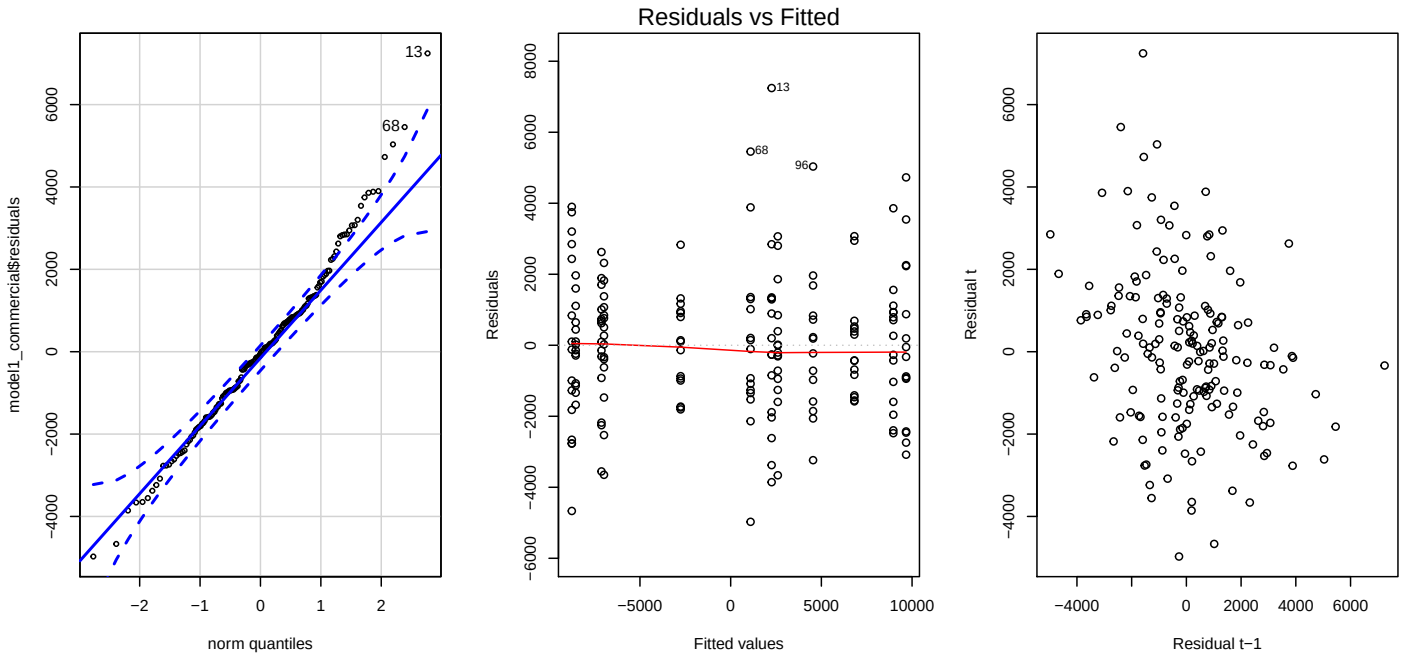
```
##
## Call:
## lm(formula = ChangeCommercialSales ~ Month, data = data.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4973.1 -1264.1   -50.4   958.1  7244.5
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2258.5      516.4   4.374 2.16e-05 ***
## MonthFebruary   -9241.3      730.3 -12.655  < 2e-16 ***
## MonthMarch        341.3      730.3   0.467  0.64083
## MonthApril      -5015.5      730.3  -6.868 1.26e-10 ***
## MonthMay         4560.3      730.3   6.245 3.47e-09 ***
## MonthJune        7411.9      730.3  10.150  < 2e-16 ***
## MonthJuly        6709.3      730.3   9.187  < 2e-16 ***
## MonthAugust     -1161.3      730.3  -1.590  0.11369
## MonthSeptember -11015.0      730.3 -15.083  < 2e-16 ***
## MonthOctober    -9380.6      743.2 -12.622  < 2e-16 ***
## MonthNovember  -10799.3      743.2 -14.531  < 2e-16 ***
## MonthDecember    2284.4      743.2   3.074  0.00247 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2000 on 165 degrees of freedom
## Multiple R-squared:  0.9192, Adjusted R-squared:  0.9138
## F-statistic: 170.7 on 11 and 165 DF,  p-value: < 2.2e-16
```

In order to verify if the assumptions of our model are met, we make the following plots:

```
n = NROW(data.train)

library(car)
par(mfrow = c(1, 3))
qqPlot(model1_commercial$residuals)
```

```
## [1] 13 68
```

```
plot(model1_commercial, 1)
plot(model1_commercial$residuals[-n], model1_commercial$residuals[-1], xlab = "Residual t-1",
    ylab = "Residual t")
```



From the first plot it seems like the residuals are not completely normally distributed. From the second plot it seems to be that all residuals are homoscedastic, though the Lowess line seems to show that the residuals don't have mean zero all the time. Finally from the third plot, there seems to be some indication of serial correlation between the residuals (one can almost fit a line with a negative slope).

To check that there is serial correlation we run the following:

```
summary(lm(model1_commercial$residuals[-1] ~ model1_commercial$residuals[-n] - 1))
```

```
##
## Call:
## lm(formula = model1_commercial$residuals[-1] ~ model1_commercial$residuals[-n] -
##     1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5052.0 -1026.4     7.6   997.9  6779.7
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## model1_commercial$residuals[-n] -0.29347    0.07272  -4.035 8.13e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1857 on 175 degrees of freedom
## Multiple R-squared:  0.08514,    Adjusted R-squared:  0.07991
## F-statistic: 16.29 on 1 and 175 DF,  p-value: 8.132e-05
```

Here the coefficient is shown to have a p-value of 0.0000813 which is very low and thus indicates that there is evidence for serial correlation if we take a significance level of $\alpha = 0.05$.

Several different transformations were tried, but we couldn't find one that corrected the issue with normality and serial correlation. We also tried out weighted least squares but also encountered similar problems. At the end we decided to keep on using linear models without weighting or transformations, but with the caveat that the pointwise estimates might be biased.

We now try to see if we can group some months into a single variable. To do so, we will calculate the mean monthly change in electricity consumption for every month, sort it and then make groups of 2 months, 3 months, 4 months and 6 months.

```
sort(tapply(data$ChangeCommercialSales, data$Month, mean))
```

```
## September  November  February   October     April    August   January
## -9045.706 -8486.938 -7482.941 -7234.062 -2994.235  1119.941  2266.118
##     March  December       May      July      June
##  3083.235  4565.312  7014.588  9008.412  9657.353
```

For model2_commercial we will group the months as follows: {{September November} {February October} {April August} {January March} {December May} {July June}}

For model3_commercial we will group the months as follows: {{September November February} {October April August} {January March December} {May July June}}

For model4_commercial we will group the months as follows: {{September November February October} {April August January March} {December May July June}}

For model6_commercial we will group the months as follows: {{September November February October April August} {January March December May July June}}

Notice that there is no model5_commercial. This is because to make it easier to keep track of the models, we are using the naming "modeli_commercial" to indicate that the model is assuming groups of i months.

We will also consider an alternative model7_commercial which is based on grouping variables based on a visual analysis of the box-plot made in the Exploratory Analysis Section. In this case we will group the months as {{September November February October} {April} {August January March} {December} {May} {July June}}

We will also consider another model called model8_commercial in which we group the following way: {{September} {November} {February} {October} {April} {August January March} {December} {May} {July} {June}}. This arises because in our summary it seems to be that the coefficients for August and March are not significantly different from zero at a level $\alpha = 0.05$, indicating that they could perhaps be grouped with January.

We now define and implement these models in R. We also calculate the AIC, Adjusted R squared and MSE (mean squared error).

```
# Create levels
ordering = order(c(9, 11, 2, 10, 4, 8, 1, 3, 12, 5, 7, 6))

levelsm1 = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)
levelsm2 = c(1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6)[ordering]
levelsm3 = c(1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4)[ordering]
levelsm4 = c(1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3)[ordering]
levelsm6 = c(1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2)[ordering]
levelsm7 = c(1, 1, 1, 1, 2, 3, 3, 3, 4, 5, 6, 6)[ordering]
levelsm8 = c(1, 2, 3, 4, 5, 6, 6, 6, 7, 8, 9, 10)[ordering]

LevelMatrix = rbind(levelsm1, levelsm2, levelsm3, levelsm4, levelsm6, levelsm7, levelsm8)

## commercial sales
commercialModelNames = c("model1_commercial", "model2_commercial", "model3_commercial", "model4_commercial",
    "model6_commercial", "model7_commercial", "model8_commercial")
commercialModels = list()
AIC.commercial.train = c()
R2.commercial.train = c()
adjR2.commercial.train = c()
MSE.commercial.train = c()

for (i in 1:NROW(LevelMatrix)) {
    MonthGrouped = data.train$Month
    levels(MonthGrouped) = LevelMatrix[i, ]
    fitted.model = lm(data.train$ChangeCommercialSales ~ MonthGrouped)

    summ = summary(fitted.model)
```

```
    commercialModels = c(commercialModels, list(fitted.model))
    AIC.commercial.train = c(AIC.commercial.train, AIC(fitted.model))
    R2.commercial.train = c(R2.commercial.train, summ$r.squared)
    adjR2.commercial.train = c(adjR2.commercial.train, summ$adj.r.squared)
    MSE.commercial.train = c(MSE.commercial.train, mean(summ$residuals^2))

}

### We print out each of the criterion
commercialModelNames
```

```
## [1] "model1_commercial" "model2_commercial" "model3_commercial"
## [4] "model4_commercial" "model6_commercial" "model7_commercial"
## [7] "model8_commercial"
```

```
AIC.commercial.train
```

```
## [1] 3206.586 3231.703 3308.793 3294.634 3425.029 3209.994 3207.510
```

```
R2.commercial.train
```

```
## [1] 0.9192226 0.9003766 0.8424833 0.8529406 0.6892995 0.9118758 0.9169441
```

```
adjR2.commercial.train
```

```
## [1] 0.9138375 0.8974636 0.8397518 0.8512503 0.6875241 0.9092991 0.9124680
```

```
MSE.commercial.train
```

```
## [1]  3728572  4598478  7270754  6788058 14341507  4067691  3833747
```

From this it can be seen that the model with highest R squared and lowest mean squared error is the full model (model1_commercial). This is expected for the training set because it has more variables and the other models are just nested versions of it. The full model also has the lowest AIC and highest adjusted R squared, though models "model2_commercial", "model7_commercial" and "model8_commercial" also seem to perform well for these criteria. Due to the small number of datapoints we decide not to perform any outlier analysis.

We now fit the models that were generated into the testing set and calculate the values of the R squared and mean squared error.

```
# Create levels Create levels
ordering = order(c(9, 11, 2, 10, 4, 8, 1, 3, 12, 5, 7, 6))

levelsm1 = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12)
levelsm2 = c(1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6)[ordering]
levelsm3 = c(1, 1, 1, 2, 2, 2, 3, 3, 3, 4, 4, 4)[ordering]
levelsm4 = c(1, 1, 1, 1, 2, 2, 2, 2, 3, 3, 3, 3)[ordering]
levelsm6 = c(1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2)[ordering]
levelsm7 = c(1, 1, 1, 1, 2, 3, 3, 3, 4, 5, 6, 6)[ordering]
levelsm8 = c(1, 2, 3, 4, 5, 6, 6, 6, 7, 8, 9, 10)[ordering]

LevelMatrix = rbind(levelsm1, levelsm2, levelsm3, levelsm4, levelsm6, levelsm7, levelsm8)

## commercial sales
commercialModelNames = c("model1_commercial", "model2_commercial", "model3_commercial", "model4_commercial",
    "model6_commercial", "model7_commercial", "model8_commercial")
R2.commercial.test = c()
MSE.commercial.test = c()

for (i in 1:NROW(LevelMatrix)) {
    MonthGrouped = data.test$Month
    levels(MonthGrouped) = LevelMatrix[i, ]
    fitted.model = commercialModels[[i]]

    predicted.values = predict.lm(commercialModels[[i]], MonthGrouped)
```

```
    MSE.commercial.test = c(MSE.commercial.test, mean((data.test$ChangeCommercialSales - predicted.values)^2))

    SSE = sum((data.test$ChangeCommercialSales - predicted.values)^2)
    SST = sum((data.test$ChangeCommercialSales - mean(data.test$ChangeCommercialSales))^2)

    R2.commercial.test = c(R2.commercial.test, (1 - SSE/SST))

}

### We print out each of the criterion
commercialModelNames
```

```
## [1] "model1_commercial" "model2_commercial" "model3_commercial"
## [4] "model4_commercial" "model6_commercial" "model7_commercial"
## [7] "model8_commercial"
```

```
R2.commercial.test
```

```
## [1] 0.9001897 0.8693058 0.8519562 0.8229337 0.7353704 0.8953488 0.8922702
```

```
MSE.commercial.test
```

```
## [1]  6284613  8229236  9321667 11149086 16662561  6589425  6783270
```

In this case we can see that the model with the lowest mean squared error and highest R squared is the full model (model1_commercial) and so we decide that this is the most appropriate model for our problem. Therefore the model we will use for this problem has the following form:

$$\Delta Y_t = \beta_0 + \beta_2 I_t\{February\} + \ldots + \beta_{12}\ I_t\{December\} + \epsilon_t$$

We now print out a summary for this model:

```
model1_commercial = lm(ChangeCommercialSales ~ Month, data = data.train)
summary(model1_commercial)
```

```
##
## Call:
## lm(formula = ChangeCommercialSales ~ Month, data = data.train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4973.1 -1264.1   -50.4   958.1  7244.5
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2258.5      516.4   4.374 2.16e-05 ***
## MonthFebruary   -9241.3      730.3 -12.655  < 2e-16 ***
## MonthMarch        341.3      730.3   0.467  0.64083
## MonthApril      -5015.5      730.3  -6.868 1.26e-10 ***
## MonthMay         4560.3      730.3   6.245 3.47e-09 ***
## MonthJune        7411.9      730.3  10.150  < 2e-16 ***
## MonthJuly        6709.3      730.3   9.187  < 2e-16 ***
## MonthAugust     -1161.3      730.3  -1.590  0.11369
## MonthSeptember -11015.0      730.3 -15.083  < 2e-16 ***
## MonthOctober    -9380.6      743.2 -12.622  < 2e-16 ***
## MonthNovember  -10799.3      743.2 -14.531  < 2e-16 ***
## MonthDecember    2284.4      743.2   3.074  0.00247 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2000 on 165 degrees of freedom
## Multiple R-squared:  0.9192, Adjusted R-squared:  0.9138
```

```
## F-statistic: 170.7 on 11 and 165 DF,  p-value: < 2.2e-16
```

The least squares estimates for this model are the following:

$$\hat{\beta}_0 = 2258.5$$
$$\hat{\beta}_2 = -9241.3$$
$$\hat{\beta}_3 = 341.3$$
$$\hat{\beta}_4 = -5015.5$$
$$\hat{\beta}_5 = 4560.3$$
$$\hat{\beta}_6 = 7411.9$$
$$\hat{\beta}_7 = 6709.3$$
$$\hat{\beta}_8 = -1161.3$$
$$\hat{\beta}_9 = -11015.0$$
$$\hat{\beta_{10}} = -9380.6$$
$$\hat{\beta_{11}} = -10799.3$$
$$\hat{\beta_{12}} = 2284.4$$

This can be interpreted as follows:

- During the month of January there is an average increase of 2258.5 units in ΔY_t.
- During the i-th month of the year there is an average change of $2258.5 + \hat{\beta}_i$ units in ΔY_t. For example in June there is an average change of $2258.5 + 7411.9$ units in ΔY_t.

It is worth mentioning however that there might be some bias in these estimates since the residuals didn't seem to be centered around 0.

Unfortunately, since the errors are also not independent, performing traditional statistical tests (such as t-tests and F-tests) and calculating traditional confidence bands will give us fallacious results.

We decide to calculate confidence intervals for the coefficients using bootstrap samples, however the lack of independence might still pose a problem when interpreting these intervals.

```
set.seed(242112)
B = 1000
coef.boot <- matrix(0, B, 12)
for (b in 1:B) {
    indices = sample(seq(1, NROW(data.train)), replace = T)
    m.boot = lm(data.train$ChangeCommercialSales[indices] ~ data.train$Month[indices])
    coef.boot[b, ] = m.boot$coefficients
}

beta0 = coef.boot[, 1]
beta2 = coef.boot[, 2]
beta3 = coef.boot[, 3]
beta4 = coef.boot[, 4]
beta5 = coef.boot[, 5]
beta6 = coef.boot[, 6]
beta7 = coef.boot[, 7]
beta8 = coef.boot[, 8]
beta9 = coef.boot[, 9]
beta10 = coef.boot[, 10]
beta11 = coef.boot[, 11]
beta12 = coef.boot[, 12]

coefnames = c("beta0", "beta2", "beta3", "beta4", "beta5", "beta6", "beta7", "beta8", "beta9",
    "beta10", "beta11", "beta12")
```

```r
for (i in 1:12) {
    output = paste0(coefnames[i], ": [", quantile(coef.boot[, i], 0.025), ",", quantile(coef.boot[,
        i], 0.975), "]")
    print(output)
}
```

```
## [1] "beta0: [925.767647058813,3643.76875]"
## [1] "beta2: [-10898.2160714285,-7707.68203463205]"
## [1] "beta3: [-1329.53737077067,1943.21190476189]"
## [1] "beta4: [-6616.13886217948,-3564.20387019231]"
## [1] "beta5: [2894.56916302447,6025.47249999998]"
## [1] "beta6: [5596.84318438915,9257.46868100649]"
## [1] "beta7: [5050.78955182072,8277.3023634454]"
## [1] "beta8: [-3007.86527777778,821.396493506494]"
## [1] "beta9: [-12934.4293478261,-9033.75382936508]"
## [1] "beta10: [-11078.8336363636,-7771.92791666666]"
## [1] "beta11: [-12407.7857142857,-9341.05004774637]"
## [1] "beta12: [405.042187499997,3999.61724264705]"
```

If we gathered more data and recalculated confidence intervals in this manner, then we would expect the true value of the parameter to be inside these intervals 95 percent of the time. We also see that the bootstrap intervals for β_3 and β_8 contain a zero which means that there is little evidence to reject the null hypothesis that the effect that March and August have on ΔY_t is the same as the effect that January has on ΔY_t at a level $\alpha = 0.05$. For all other months since their confidence intervals don't contain the value zero, there is evidence that the effects of the othe other months on ΔY_t is significantly different from the effect of January on ΔY_t.

Conclusions

Our main question of interest for this project was: **Can we predict the monthly changes in the consumption of electrical energy based on the available variables?**

For the residential sector we seemed to be quite succesful in doing so, and were able to fit a model that predicted residential electricity consumption quite well. The underlying assumptions of this model were also met which made it possible to carry out several interesting inferences.

Unfortunately the commercial sector posed more difficulties, and at least the approach we used for this project was not the most adequate. Perhaps one small success was getting a model with a large R squared for the training set and testing set, indicating that it works well for making pointwise predictions in the short-term. Nevertheless for the longer term it is necessary to be able to calculate confidence intervals. Unfortunately, since the underlying assumptions of Gaussian regression models couldn't be met, we were not able to calculate prediction intervals for the consumption of electricity in the commercial sector.

In the end, it seems to be that this project illustrated both the strengths and weaknesses of regression modelling. There are some scenarios in which it is easy to carry out and can be used to make interesting inferences, but there are other scenarios in which it is harder to construct a model that satisfies the underlying assumptions and in turn it becomes more difficult to make inferences about the phenomena in question.

References

[1] "Electricity Explained: Electricity in the United States" U.S. Energy Information Administration (EIA), https://www.eia.gov/energyexplained/index.php?page=electricity_in_the_united_states . Last Accessed December 6 2018.

[2] "Electricity Data Browser." U.S. Energy Information Administration (EIA), https://www.eia.gov/electricity/data/browser/#/topic/5?geo=g&agg=0,1&endsec=vg . Last Accessed December 6 2018.