Tios Capital Homework

Haosheng Luo

January 29, 2016

Check Package

Before carrying out any analysis, let's define a R function that automatically loads the packages we need.

```
check_packages = function(names)
{
  for(name in names)
  {
    if (!(name %in% installed.packages()))
      install.packages(name, repos="http://cran.us.r-project.org")

    library(name, character.only=TRUE)
  }
}
```

Question 1

Introduction

In this question, we model the 30-minute transaction data for the electricity settlement price and total demand in Victoria, Australia in 2014. The modeling process can be separated into two parts: modeling the association between settlement price and total demand, and forecasting total demand. Linear models and neural networks are applied. In the both process, we select the model based on Mean Squared Error (MSE) and Mean Absolute Percentage Error (MAPE) and cross-validation. The error analysis also involves comparison with a naive model. Finally we provide a short-term forecast of settlement prices from Jan 1, 2015 00:30:00 am to Jan 3, 2015 00:00:00 am, based on the model selected.

Data Loading and Validation

The dataset is supposed to contain $365 \times 24 \times 2 = 17520$ rows of datapoints. Before any exploration into the data, we need to construct the intermediate variables and check the data quality, to make sure no duplicated or missing datapoints, and that the data are collected in 2014 (except one at 0:00:00 am on Jan 1, 2015). Intermediate variables include:

1. **YEAR**: the year of a datapoint.
2. **MONTH**: the month of a datapoint.
3. **DAY**: the day of a datapoint.
4. **TIME**: the clock time of a datapoint.
5. **HOUR**: the hour of a datapoint.
6. **MINUTE**: the minute of a datapoint.
7. **WDAY**: the days of a week of a datapoint.
8. **WDAY2**: 0.1 if it is a weekday datapoint, and 0 if it is a weekend datapoint. 0.1 is chosen due to the computation concern arising in the neural network.

Following are the results of dataset validation.

```
check_packages(c("stats","devtools","utils","lubridate","dplyr","tseries","neuralnet"))

data = read.csv('~/Tios/Homework/australia-victoria-energy.csv')
data = dplyr::select(data,SETTLEMENTDATE,TOTALDEMAND,RRP)%>%
  mutate(YEAR=year(SETTLEMENTDATE),MONTH=month(SETTLEMENTDATE),DAY=day(SETTLEMENTDATE),
         HOUR = hour(SETTLEMENTDATE),MINUTE = minute(SETTLEMENTDATE),
         TIME = paste(HOUR, MINUTE),WDAY=wday(SETTLEMENTDATE,label=T),WDAY2=0.1)

data[data$WDAY%in%c("Sat","Sun"),"WDAY2"]=0
year = transmute(data, YEAR=year(SETTLEMENTDATE))
n = nrow(data)
(paste("Number of all the datapoints:", n))
```

```
## [1] "Number of all the datapoints: 17520"
```

```
(paste("Number of the datapoint in 2015:", nrow(data[year$YEAR!=2014,])))
```

```
## [1] "Number of the datapoint in 2015: 1"
```

```
(paste("Number of duplicated data:", nrow(data[duplicated(data$SETTLEMENTDATE),])))
```

```
## [1] "Number of duplicated data: 0"
```

```
(paste("Number of datapints that contain missing values:", nrow(data[!complete.cases(data),])))
```

```
## [1] "Number of datapints that contain missing values: 0"
```

As a result, we can conclude that the dataset is intact.

A Quick Glance at the Settlement Price and Total Demand

To recognize the basic nature of the price and demand time-series, we need to take a quick scan over the time-series. The result will help us to build assumptions and the associated models.

Settlement Price and Total Demand Over 2014

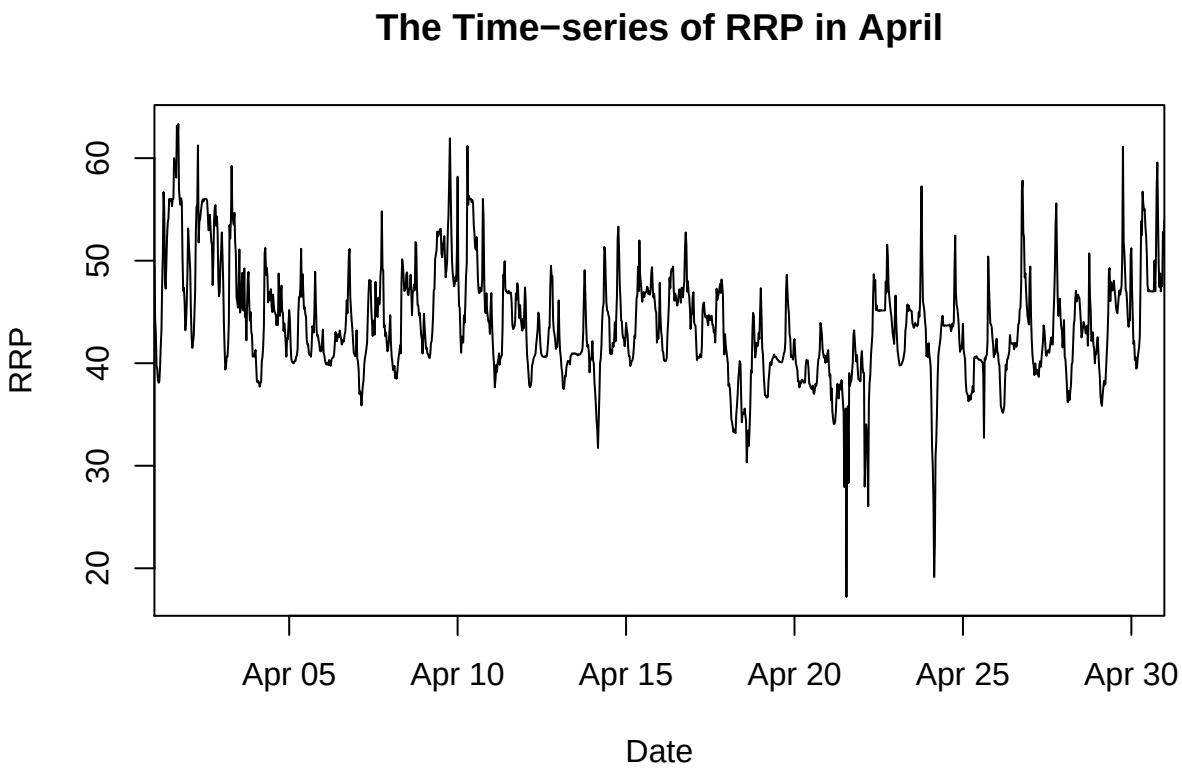
```
#plot(data$RRP,type="l",xlab="Index", ylab="RRP", main="The Time-series of RRP")
#plot(data$TOTALDEMAND,type="l",xlab="Index",ylab="Total Demand",main="The Time-series of Total Demand"
```

Given the nature of electricity generation, transmission, storage and auction design, the settlement price series exhibit much more complexity than the demand series [1]. In the degegulated market, the settlement price can be highly-volatile, and much more likely to become negative or exceptionally high than other commodities [2]. In the above plots we can discover a major price and demand series fluctuation from Jan 14th to 17th, due to the 4 consecutive days of "heatwave not seen for 100 years" [3], and several minor fluctuations in days such as Feb 02, due to the similar hot weather [4].

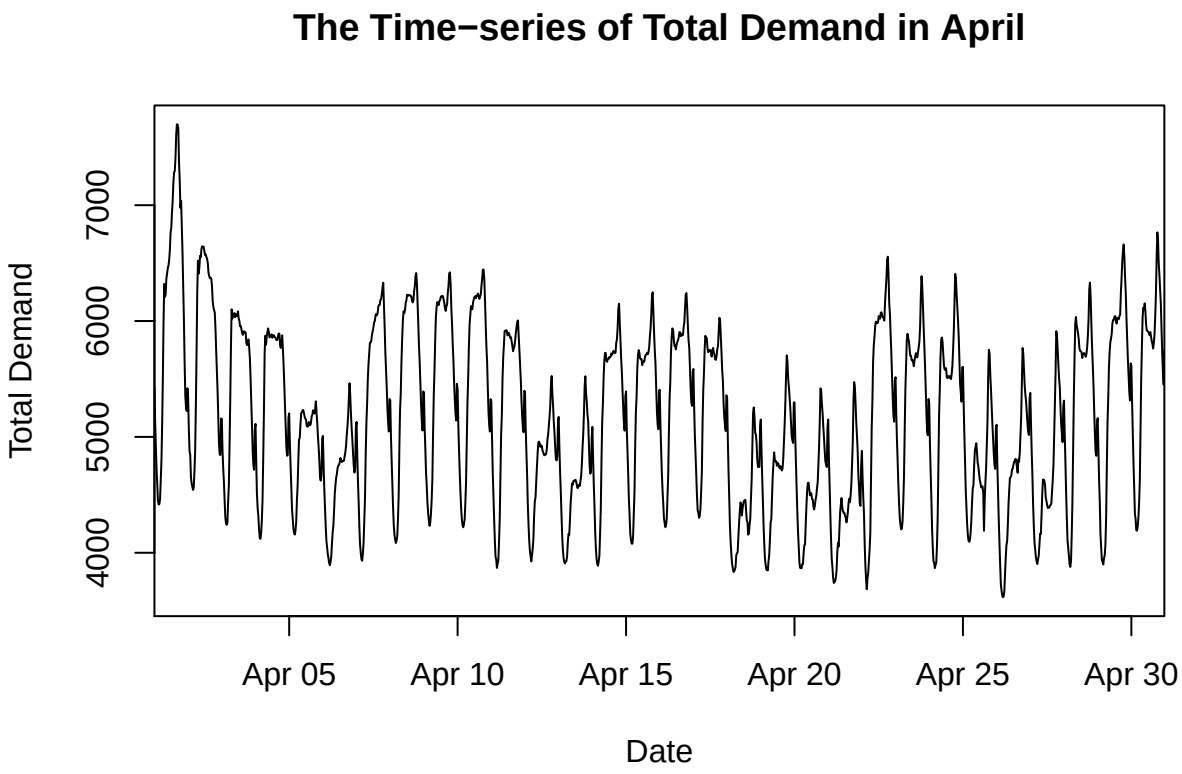
Also,

Seasonality

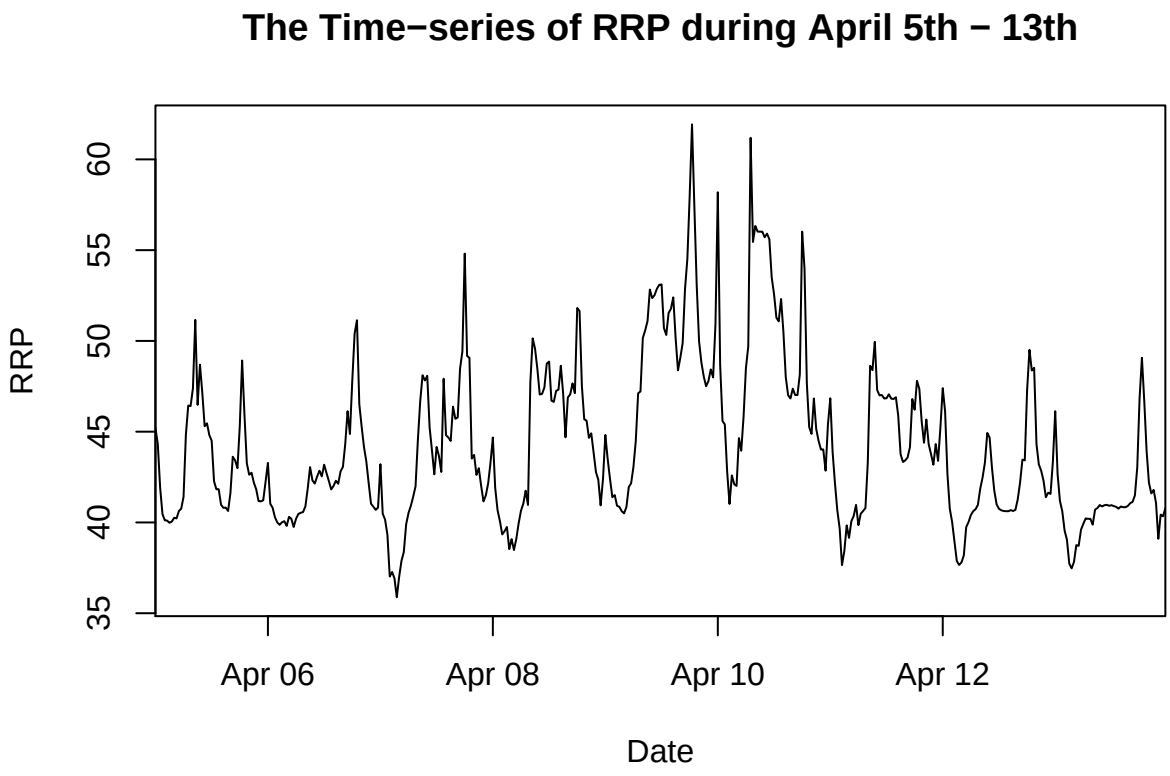
```
datetime = seq(from=as.POSIXct("2014-4-1 0:00"), to=as.POSIXct("2014-4-30 23:30"),
               by="30 mins")
plot(datetime,data[data$MONTH==4,"RRP"], type="l", xaxs='i', xlab="Date",
     ylab="RRP", main="The Time-series of RRP in April")
```



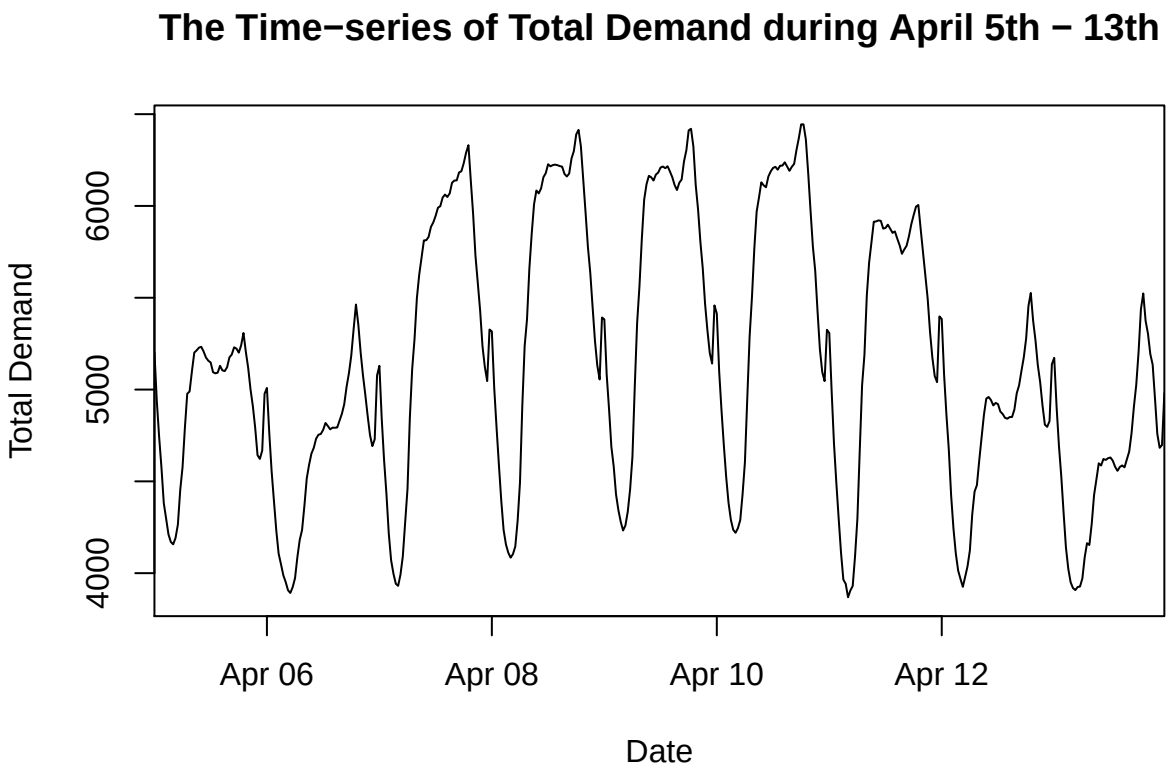

```
plot(datetime,data[data$MONTH==4,"TOTALDEMAND"], type="l", xaxs='i', xlab="Date",
     ylab="Total Demand", main="The Time-series of Total Demand in April",
     xlim = as.POSIXct(c("2014-4-1 0:00","2014-4-30 23:30")))
```




```
datetime = seq(from=as.POSIXct("2014-4-5 0:00"), to=as.POSIXct("2014-4-13 23:30"),
               by="30 mins")
plot(datetime, data[(data$MONTH==4)&(data$DAY>4)&(data$DAY<14),"RRP"], type="l",
     xaxs='i', xlab="Date",ylab="RRP", main="The Time-series of RRP during April 5th - 13th")
```




```
plot(datetime, data[(data$MONTH==4)&(data$DAY>4)&(data$DAY<14),"TOTALDEMAND"],
     type="l", xaxs='i', xlab="Date",ylab="Total Demand",
     main="The Time-series of Total Demand during April 5th - 13th")
```



Selecting data in April, where the price series exhibits mild fluctuation, we observe strong weekly seasonality and daily seasonality in the total demand series, while they are weaker in the price series. Due to the large number of the clock times, we merely incorporate the weekly effect in the model by adding a dummy variable indicating weekday, and assume that the daily seasonality in the price series can be sufficiently captured by the daily seasonality in the demand series. Also, the effect of holiday effect is also modeled as a potential variables, as suggested in [1]. The public holiday (including both national and regional) database is generated according to [5].

```
# generate public holiday database
Holiday = as.data.frame(matrix(c(1,1,1,27,3,10,4,18,4,21,4,25,6,9,11,4,12,25,12,26),
                nrow = 10, ncol = 2, byrow=T))
colnames(Holiday) = c("MONTH","DAY")
Holiday$holiday = 0.1
```

Data Smoothing and Truncating

To address the problem of high volatility in the price series, we choose to smooth the data of the major volatility event during Tuesday, Jan. 14th to Friday, Jan. 17th, by replacing the total demand and price series with the average of demand and price at each half hours on weekdays. However, we choose to truncate the price spike in the other minor event at 80 dollars, and try to model the occasional yet relatively milder price volativity through incorporating higher orders of the total demand data. Also, to stablize the data and improve the computation efficiency, our analysis is on the logarithm scale. As a result, we replace negative price data with 0.01, whose log value can still be sufficiently small.

```
data2 = data
hot_index = rownames(data[data$MONTH==1&data$DAY%in%14:17,])
normal_wday = data[((data$MONTH!=1)|(!data$DAY%in%14:17))&data$WDAY2==0.1,]
normal_wday = group_by(normal_wday,HOUR,MINUTE)
normal_wday = as.data.frame(summarise(normal_wday,TOTALDEMAND=mean(TOTALDEMAND),RRP=mean(RRP)))
normal_wday = rbind(normal_wday,normal_wday,normal_wday,normal_wday)
data2[hot_index,c("TOTALDEMAND","RRP")]=normal_wday[c("TOTALDEMAND","RRP")]
data2[data2$RRP<0,"RRP"]=0.01
data2[data2$RRP>80,"RRP"]=80
```

Following we can find a strong positive correlation between the total demand for electricity and settlement price.

```
#plot(data$RRP, data$TOTALDEMAND, type = "p", pch=20, xlab="RRP ($/MWh)",ylab="Total Demand (MW)",main=
```

Model Setting

Inspired by the five-mintue electricity forecasting method adopted by the Australian Energy Market Operator (AEMO) [6], we will utilize both the linear and neural network model, and data on the logarithmatic scale to forecast the demand and price. Also, similar to [6], we include "4 log changes immediately prior to the period being predicted and 5 leading up to and including the period occuring exactly one week before the time of the desired prediction" [6], in the demand forecasting model. In our 30-min case, the corresponding lag orders are 1 to 4, and $7 \times 24 \times 2 = 336$ to $7 \times 24 \times 2 + 5 = 340$.

Therefore, we create new variables as follows:

1. **LCRRP**: the log change of the settlement price by 30 minutes.
2. **LCTOTD**: the log change of the total demand by 30 minutes.
3. **LCTOTD2** and **LCTOTD3**: the square and cubic of the log change of the settlement price by 30 minutes.
4. **L1LCTOTD** to **L340LCTOTD**: the nine lags of the log change of the settlement price by 30 minutes discussed above.

The weekend effect is considered by assigning 0.1 to weekdays and 0 to weekends, and the holiday effect by assigning 0.1 to public holidays and 0 to the other days (choosing 0.1 due to the computation concern in the neural network models).

In short, the proposed models are as follows:

Price Models: (5-fold cross-validation is carried out *due to computation concern with neural network*) 1. LCRRP ~ LCTOTD + LCTOTD2 + LCTOTD3 2.

Demand Models: (10-fold cross-validation is carried out) 1.