Loess Regression for Predicting Wait Time

Gaurav Awade

14/12 /2019

Data Pre processing and prepration

LOESS is method of fitting a smooth surface between an outcome and up to four predictor variables. This is a nonparametric method because the linearity assumptions of conventional regression methods have been relaxed. Instead of estimating parameters like m and c in y = mx +c, a nonparametric regression focuses on the fitted curve. It is called local regression because the fitting at say point x is weighted toward the data nearest to x. The distance from x that is considered near to it is controlled by the span setting alpha.

```
library(magicfor)
library(writexl)
library(lubridate)
library(dplyr)
library(Metrics)
library(ggplot2)
library(data.table)

dot <- fread("dot.csv")
dot=dot%>%mutate(Month=ifelse(dot$Month=="A) January",1,ifelse(dot$Month=="B) February",2,
                        ifelse(dot$Month=="C) March",3,ifelse(dot$Month=="D) April",4,
                        ifelse(dot$Month=="E) May",5,ifelse(dot$Month=="F) June",6,
                        ifelse(dot$Month=="G) July",7,ifelse(dot$Month=="H) August",8,
                        ifelse(dot$Month=="I) September",9,ifelse(dot$Month=="J) October",10,
                        ifelse(dot$Month=="K) November",11,12))))))))))))
dot$Month=as.factor(dot$Month)
dot=data.frame(Branch=as.factor(dot$Branch_Name),w=as.numeric(dot$waiting_time_seconds),
               day=as.numeric(dot$DayNumber),hour=as.numeric(dot$Hour),month=as.numeric(dot$Month), week=
               year=as.factor(dot$Year),date=as.Date(dot$day))
dot=arrange(dot,date)
```

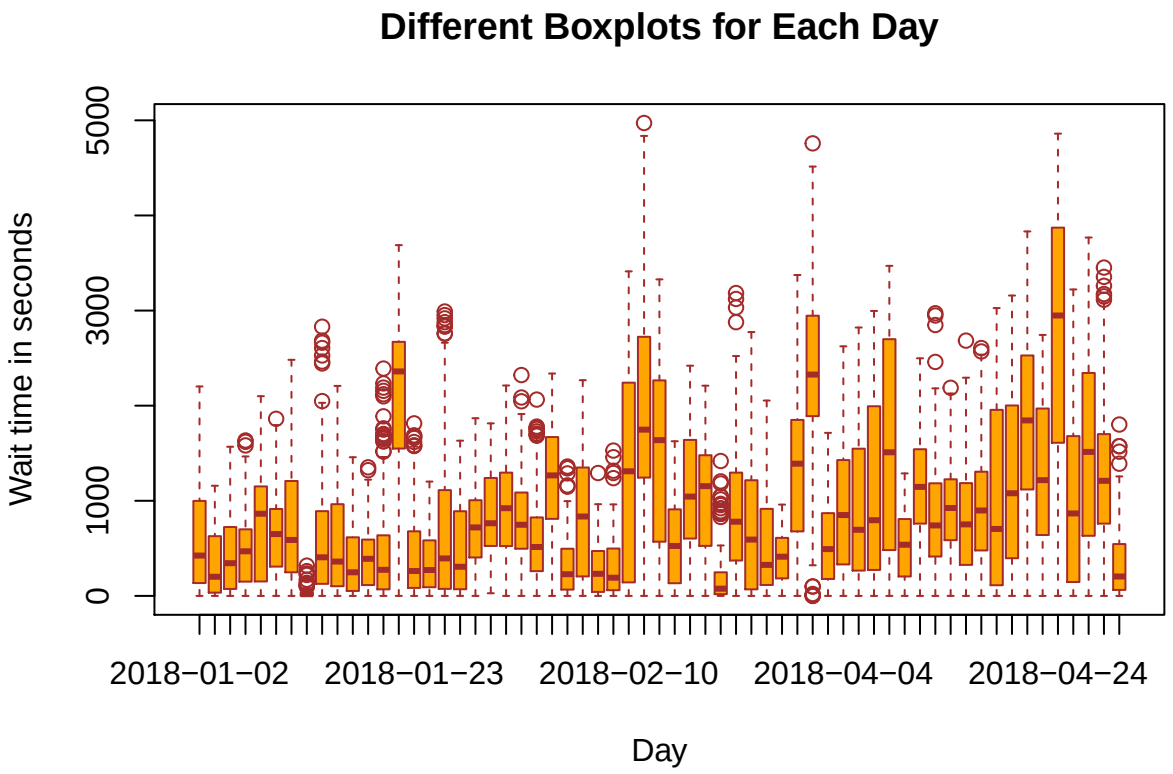
In this our goal is to predict wait time for a customer before he / she get served at a DOT centre. For this we have just time series data. The data has date, wait time and transaction times as features. We converted the given date in the following format – Year, Month number, Week number, Day Number, and hour of the day. This means that all the data points between 9AM to 10 AM will be displayed in the bracket of 9th hour of that day. The data set has various DOT stations and has data for various years form 2016-2019.

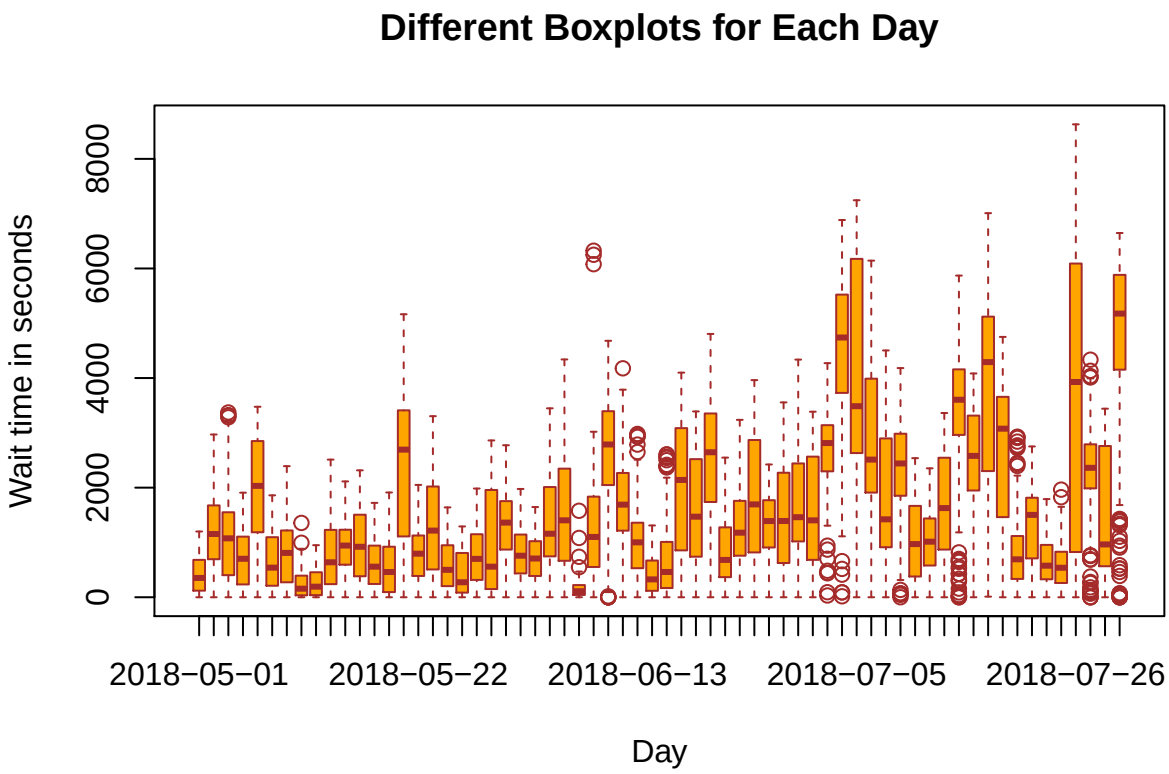
Data Exploration

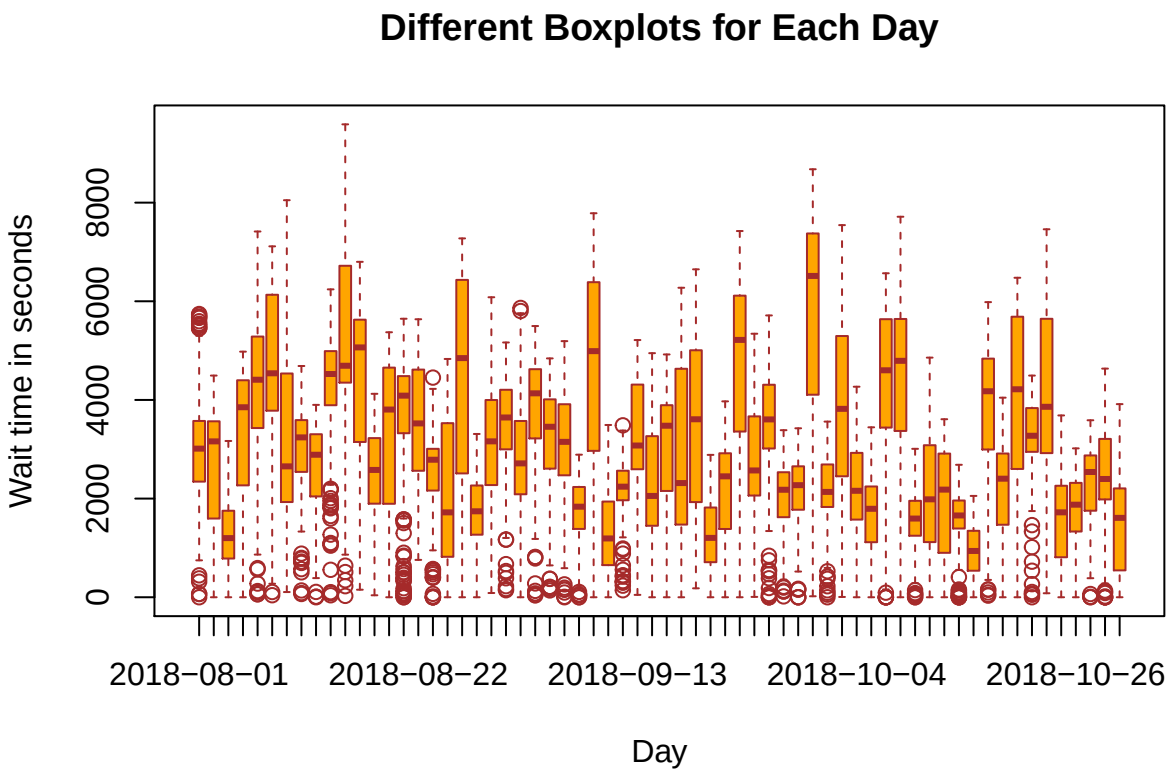
Ames

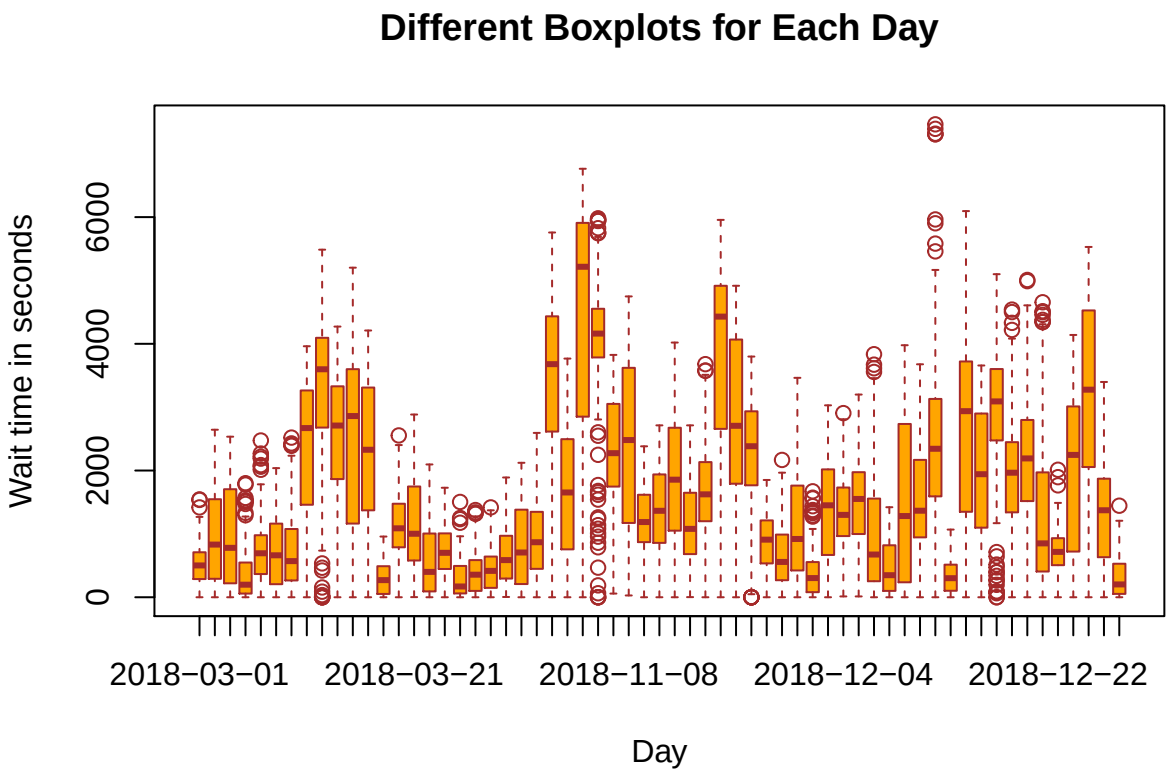
The following box plots give you wait time distribution for 3 stations for entire year grouped by date. We can see that there is very high volatility in the wait time over the year. The last 6 months of the year have larger wait times than the first 6 months. We can also see that mean wait time is different for different DoT stations. The wait time for bigger cities such as Ames is consistently higher as compared to other smaller cities. So we need to build individual models for each city. The wait time for adjacent box plots is mostly similar or it follows some local pattern so local regression can be useful here. **Because of this LOESS regression is ideal for this application as it is based on the neighbouring datapoints.**

```
for(i in c(1,4,7,10)) {
  ankeny=subset(dot,Branch=="Ames" & year=='2018'& (month==i | month==i+1| month==i+2))
  boxplot(w~date,
          data=ankeny,
          main="Different Boxplots for Each Day",
          xlab="Day",
          ylab="Wait time in seconds",
          col="orange",
          border="brown")
}
```

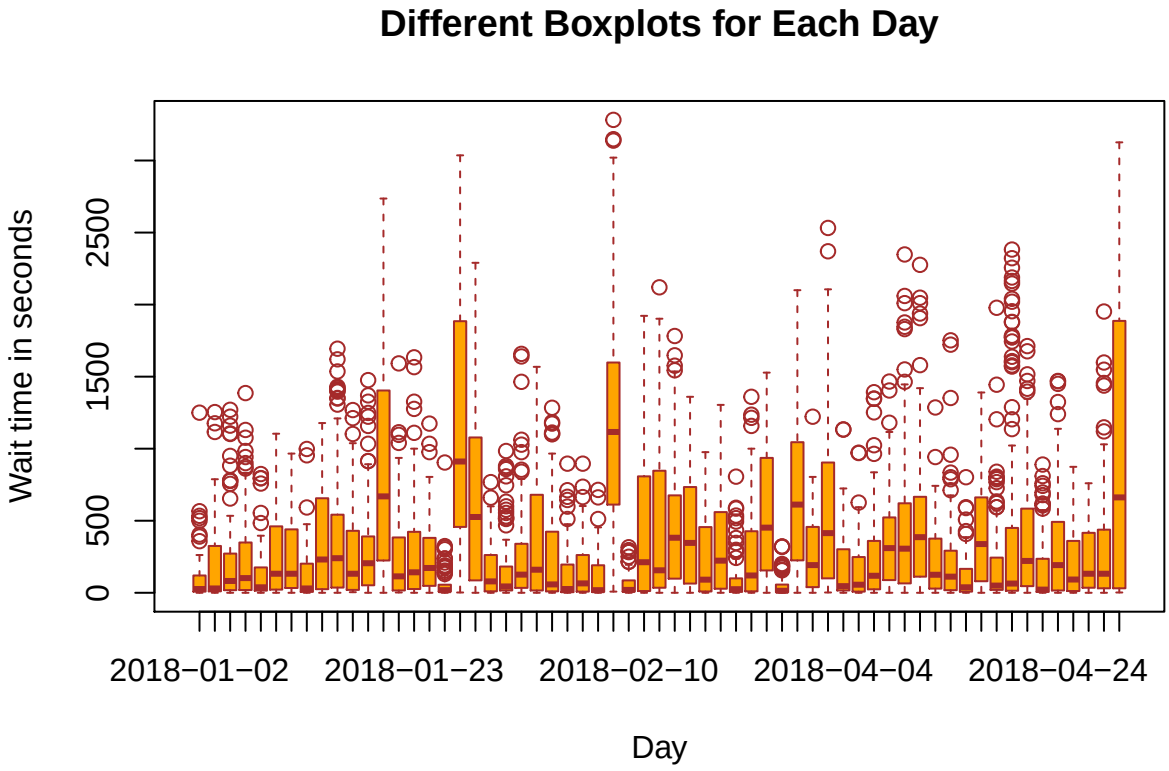


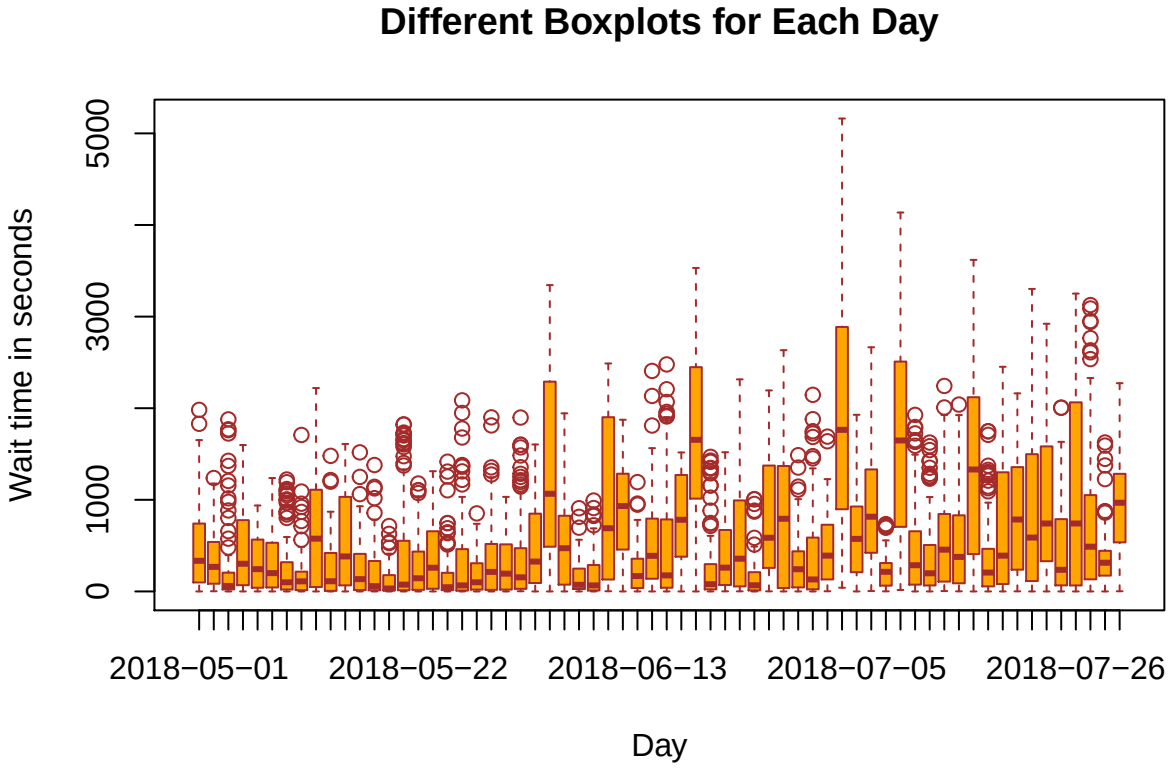

Different Boxplots for Each Day
Wait time in seconds
0
2000
4000
6000
8000
2018−05−01
2018−05−22
2018−06−13
2018−07−05
2018−07−26
Day

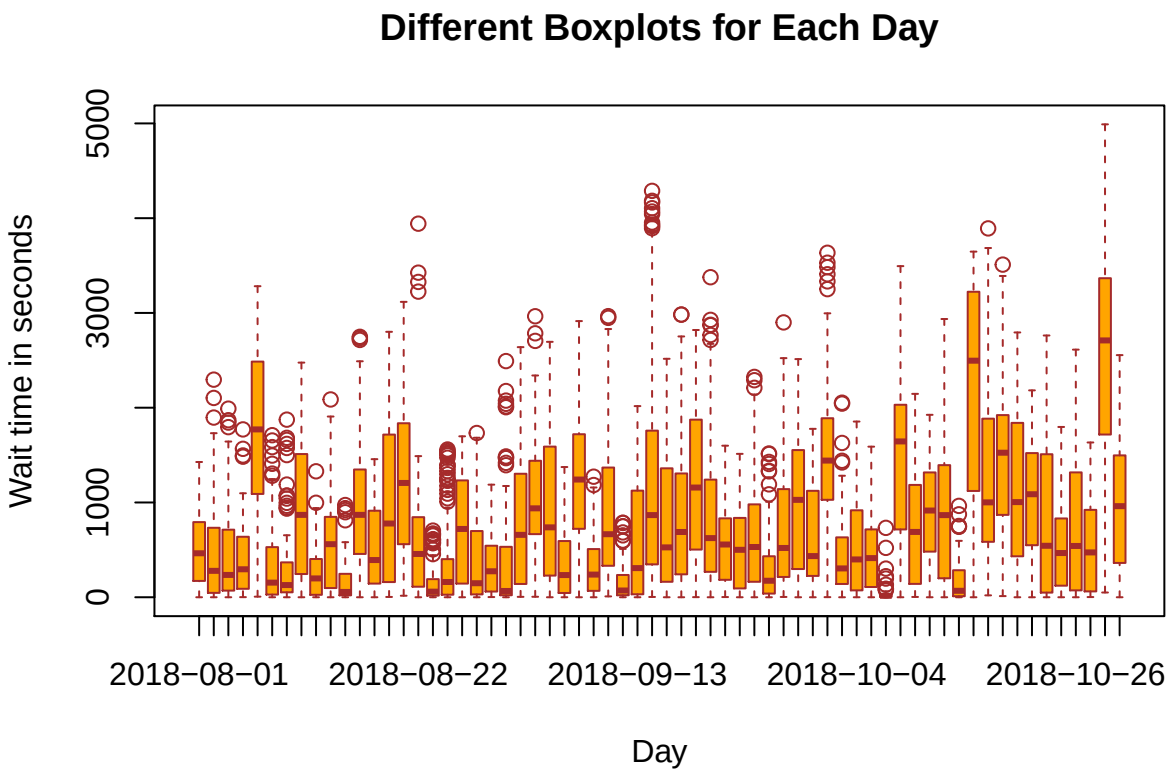

Different Boxplots for Each Day
Wait time in seconds
0
2000
4000
6000
8000
2018−08−01
2018−08−22
2018−09−13
2018−10−04
2018−10−26
Day

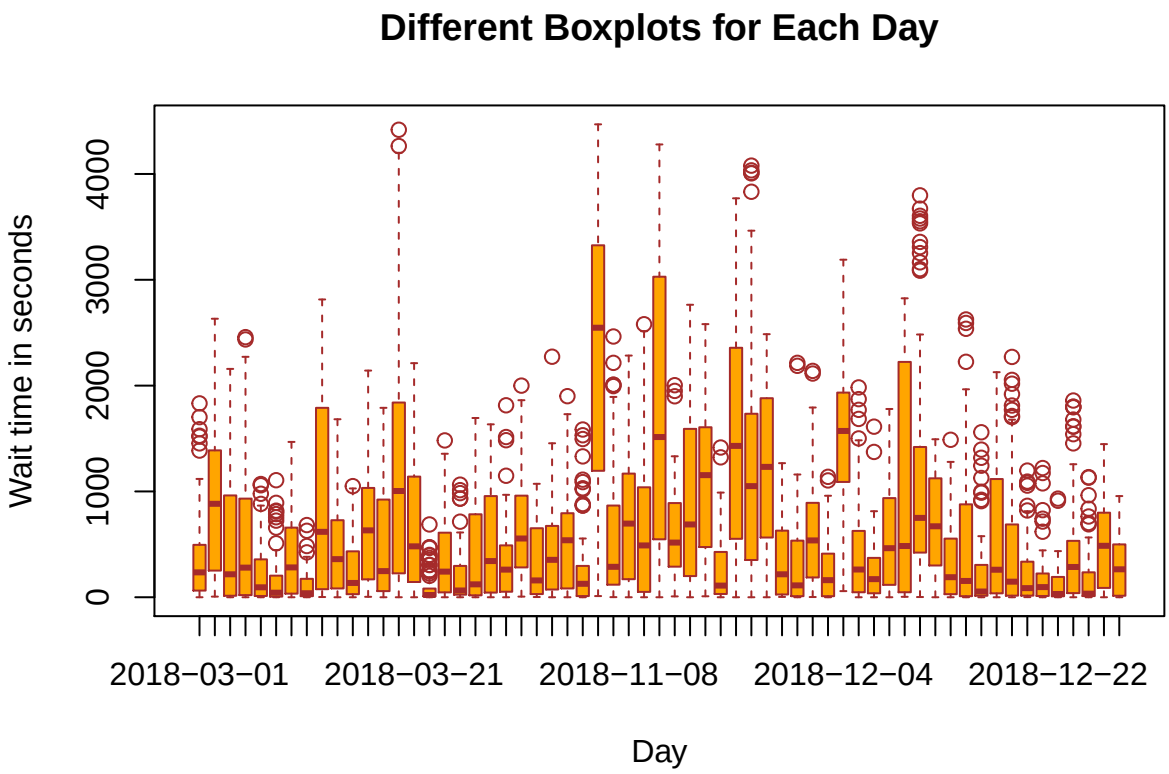

Different Boxplots for Each Day
Wait time in seconds
0
2000
4000
6000
2018−03−01
2018−03−21
2018−11−08
2018−12−04
2018−12−22
Day

Burlington

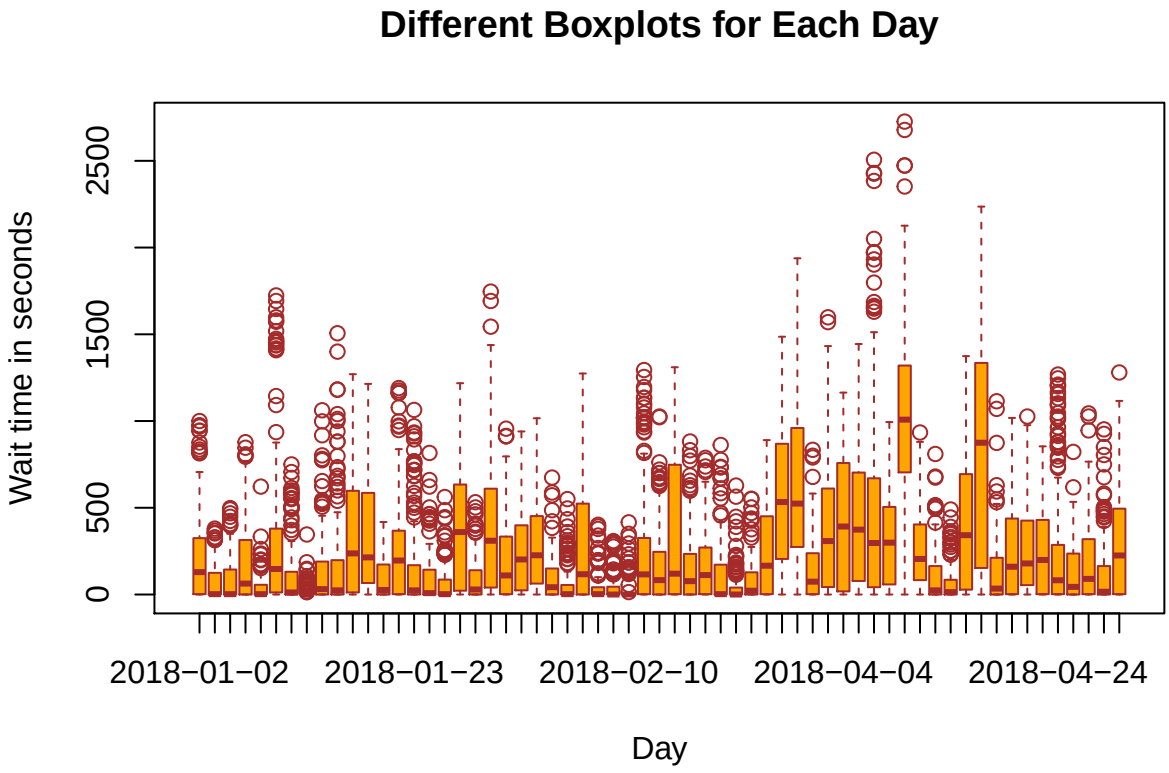

Different Boxplots for Each Day
Wait time in seconds
0
500
1500
2500
2018−01−02
2018−01−23
2018−02−10
2018−04−04
2018−04−24
Day

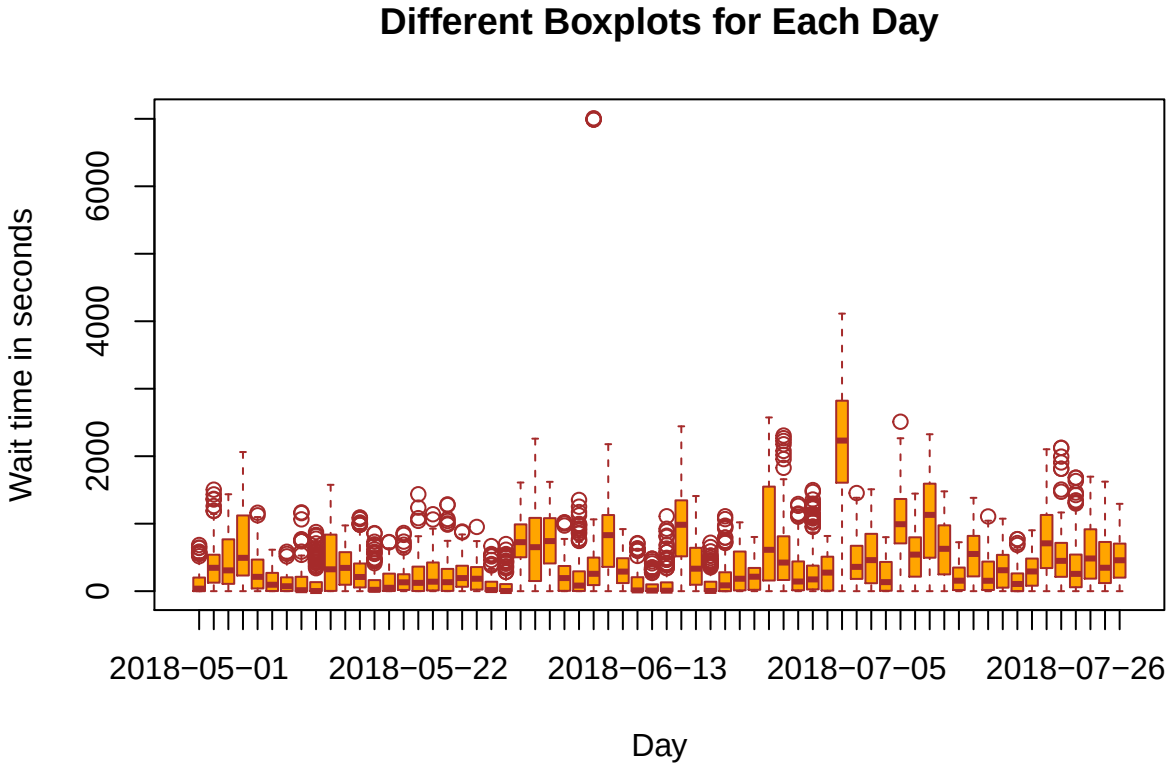

Different Boxplots for Each Day
Wait time in seconds
0
1000
3000
5000
2018−05−01
2018−05−22
2018−06−13
2018−07−05
2018−07−26
Day

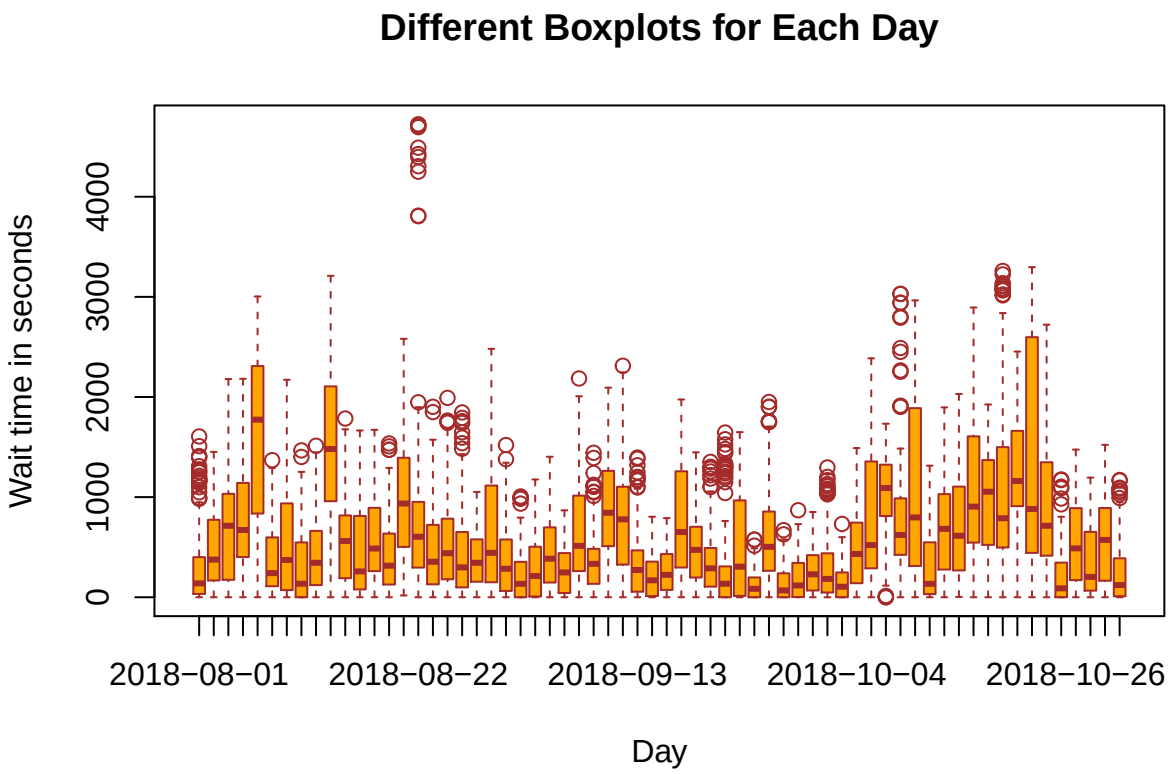

Different Boxplots for Each Day
Wait time in seconds
0
1000
3000
5000
2018−08−01
2018−08−22
2018−09−13
2018−10−04
2018−10−26
Day

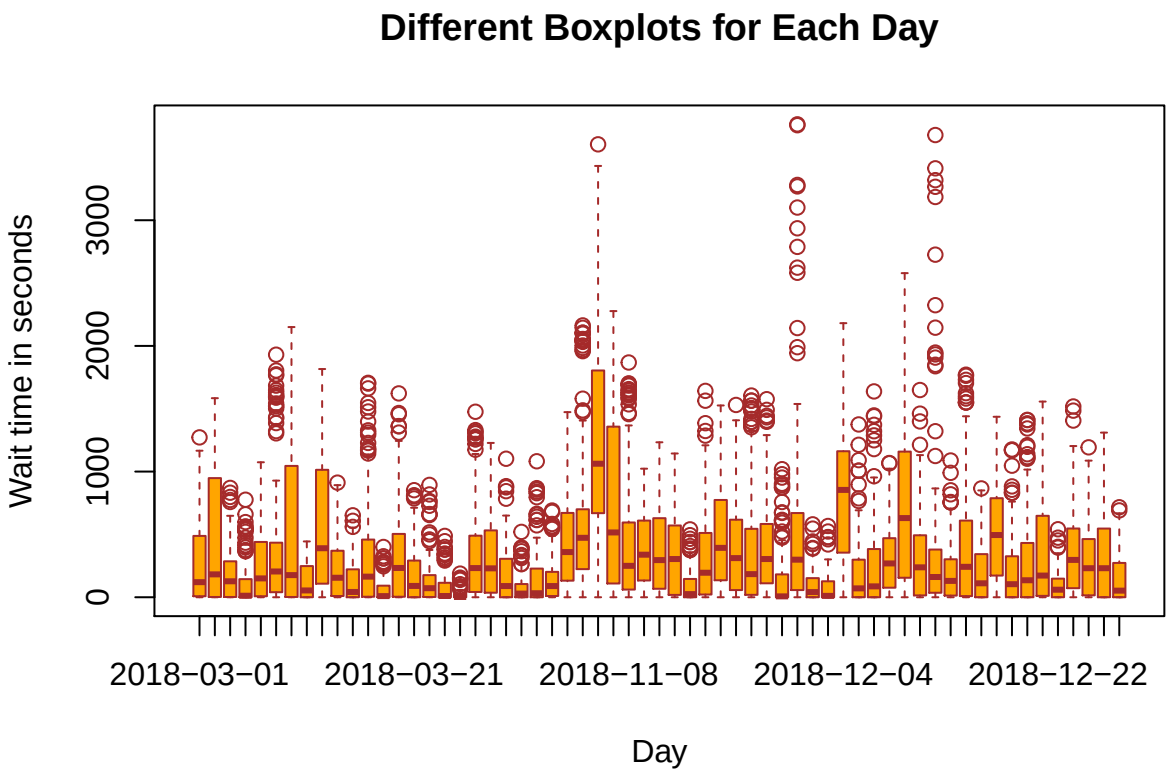

Different Boxplots for Each Day
Wait time in seconds
0
1000
2000
3000
4000
2018−03−01
2018−03−21
2018−11−08
2018−12−04
2018−12−22
Day

Dubuque


Different Boxplots for Each Day
Wait time in seconds
0
500
1500
2500
2018−01−02
2018−01−23
2018−02−10
2018−04−04
2018−04−24
Day


Different Boxplots for Each Day
Wait time in seconds
0
2000
4000
6000
2018−05−01
2018−05−22
2018−06−13
2018−07−05
2018−07−26
Day


Different Boxplots for Each Day
Wait time in seconds
0
1000
2000
3000
4000
2018−08−01
2018−08−22
2018−09−13
2018−10−04
2018−10−26
Day



Loess

The loess regression has two hyperparameters one degree of polynomial and second span. Degree of polynomial controls the degree of linear regression and span controls the number of datapoints from the neighbours to be considered for regression. The higher the span more data points from the neighbour are considered and more the generalised model is. For our application after experimenting with various data from various year and various DoT station we found that optimum span is 0,3 and degree is 1. We use a strategy of training on two weeks and predicting the next week. For example if we train the model on 1st and 2nd week we predict the 3rd week and so on. This means that we are assuming that wait times of adjecent weeks are dependent. In simple words what happened in last two weeks will happen in third week as well.

```
db=na.omit(subset(dot,Branch=="Dubuque" & year=='2018' & (week==6 | week==7)))
test=na.omit(subset(dot,Branch=="Dubuque" & year=='2018' & week==8))
test.db=data.frame(b=test$day,c=test$hour,d=test$month,e=test$week)
test_w= test$w
a=db$w
b=db$day
c=db$hour
d=db$month
e=db$week
loessMod10 <- loess(lm(a~b+c+e), span=0.3, degree = 1, control = loess.control(surface = "direct"))
p=predict(loessMod10, newdata = test.db )
MAE=mae(test_w,p)
RMSE=rmse(test_w,p)
```

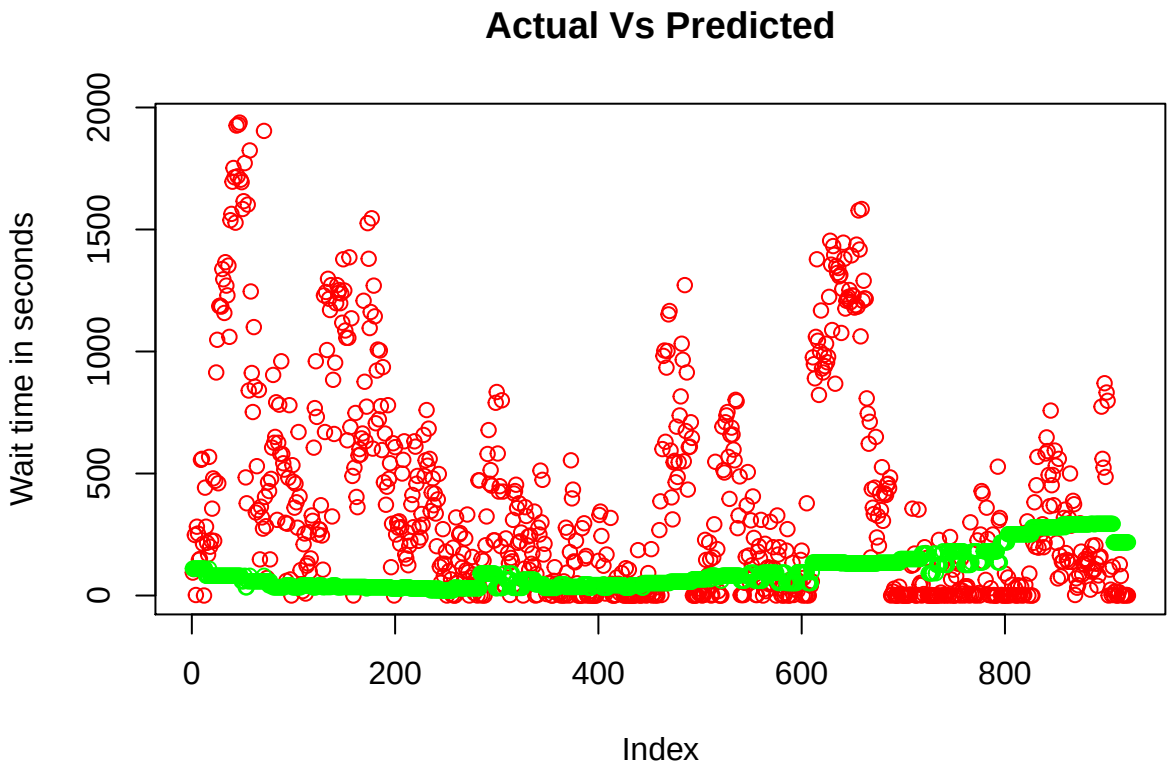
```
cat("The RMSE is:",RMSE)
```

```
## The RMSE is: 523.5009
```

```
cat("The MAE is:",MAE)
```

```
## The MAE is: 347.8454
```

```
x=1:length(p)
plot(x,test_w,col="red",
     main="Actual Vs Predicted",
     xlab="Index",
     ylab="Wait time in seconds")
points(x,p,col="green")
```



For Dubuque we trained on 6th and 7th week and predicted 8th week from 2018. These MAE and RMSE values can look higher but the wait time is in seconds so MAE of 350 means there is an error of 6 minutes which is acceptable given we have just time series data.

```
db=na.omit(subset(dot,Branch=="Ames" & year=='2018' & (week==20 | week==21)))
test=na.omit(subset(dot,Branch=="Ames" & year=='2018' & week==22))
test.db=data.frame(b=test$day,c=test$hour,d=test$month,e=test$week)
test_w= test$w
a=db$w
```

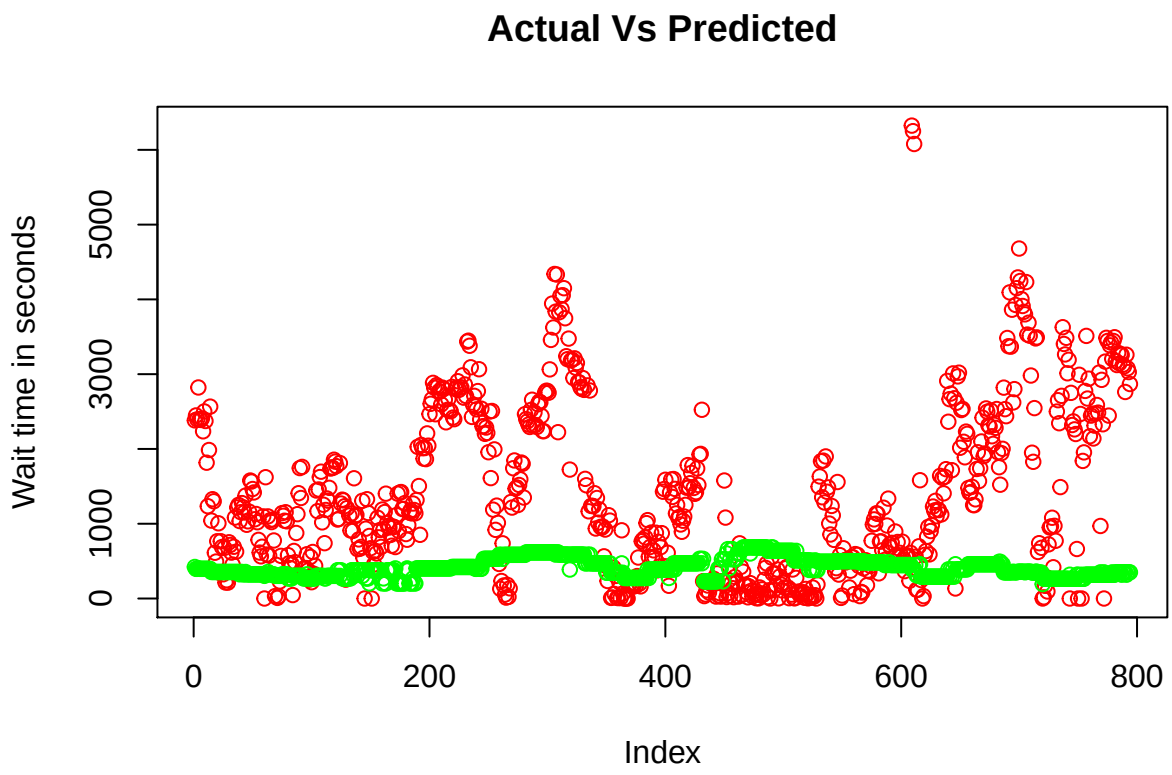
```
b=db$day
c=db$hour
d=db$month
e=db$week
loessMod10 <- loess(lm(a~b+c+e), span=0.3, degree = 1, control = loess.control(surface = "direct"))
p=predict(loessMod10, newdata = test.db )
MAE=mae(test_w,p)
RMSE=rmse(test_w,p)
cat("The RMSE is:",RMSE)
```

```
## The RMSE is: 1512.713
```

```
cat("The MAE is:",MAE)
```

```
## The MAE is: 1137.22
```

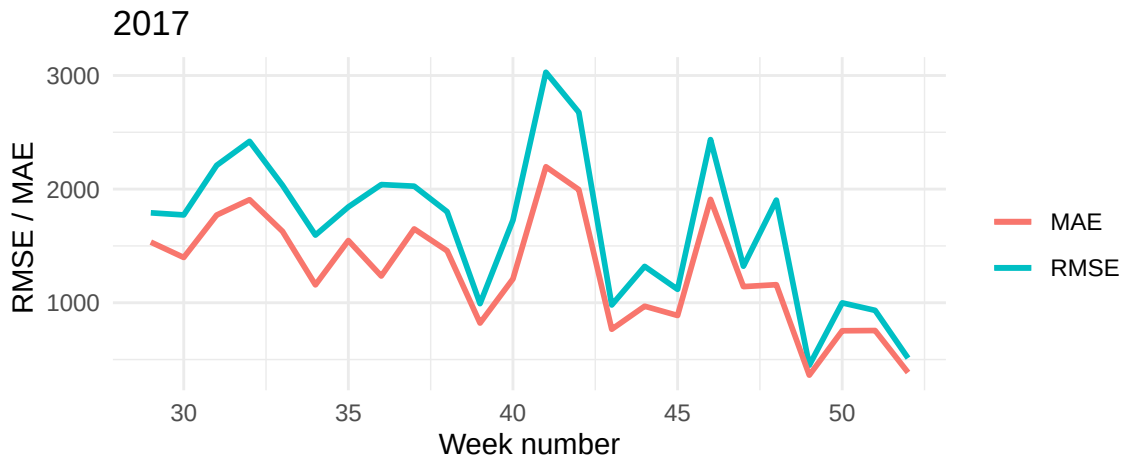
```
x=1:length(p)
plot(x,test_w,col="red",
     main="Actual Vs Predicted",
     xlab="Index",
     ylab="Wait time in seconds")
points(x,p,col="green")
```

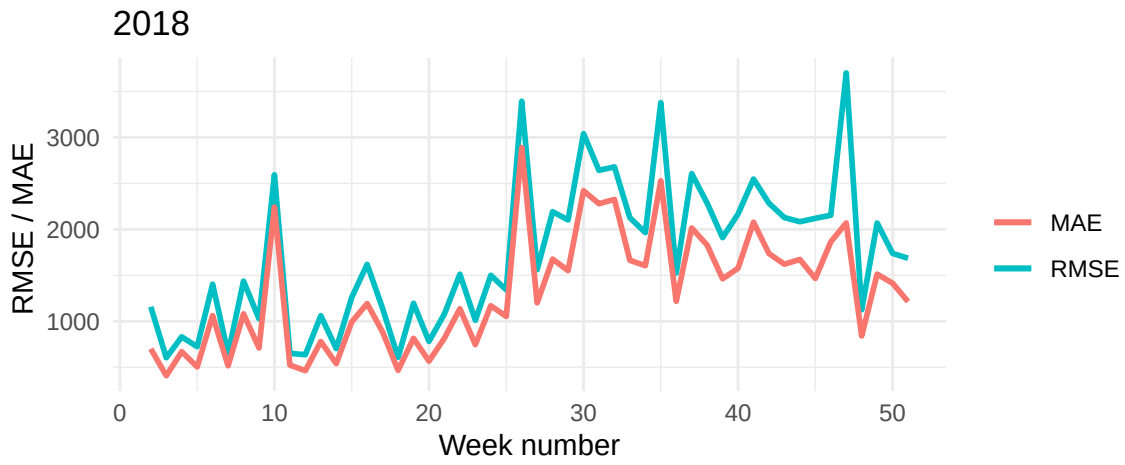


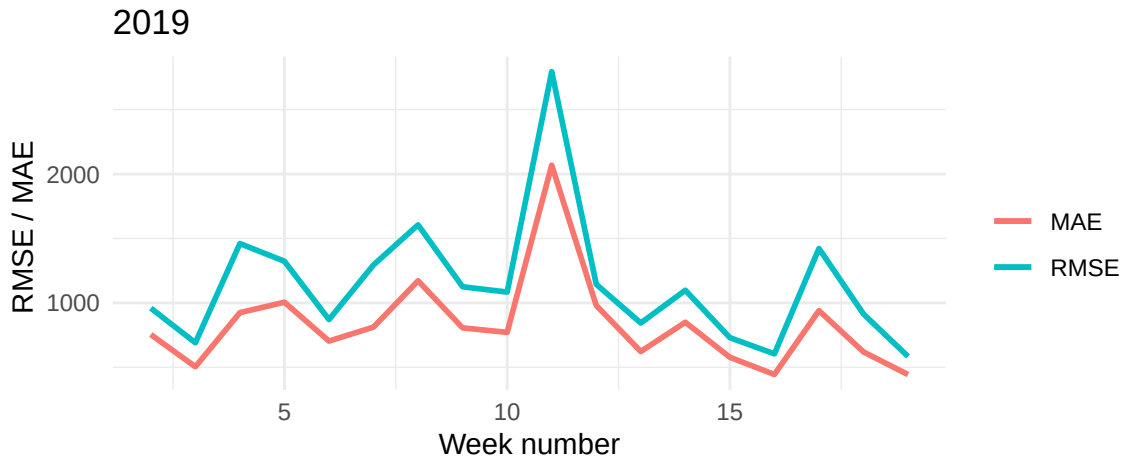
Ames

The following graph shows the RMSE and MAE values for 3 years 2017, 2018 and 2019 for three different stations Ames, Burlington and Dubuque. We train our moel on two weks and predict the next week.

```
ames=na.omit(subset(dot,Branch=="Ames" & year=='2017'))
  magic_for(silent = TRUE)
  for (i in ames$week[1]:ames$week[1]:(ames$week[dim(ames)[1]]-2)){
    db=subset(ames,(week==i | week==i+1))
    test=subset(ames, week==i+2)
    test.db=data.frame(b=test$day,c=test$hour,d=test$month,e=test$week)
    test_w= test$w
    a=db$w
    b=db$day
    c=db$hour
    d=db$month
    e=db$week
    loessMod10 <- loess(lm(a~b+c+e), span=0.3, degree = 1, control = loess.control(surface = "direct"))
    p=predict(loessMod10, newdata = test.db )
    MAE=mae(test_w,p)
    RMSE=rmse(test_w,p)
    put(i+2,MAE,RMSE)
  }
  result=magic_result_as_dataframe()
  theme_set(theme_minimal())
  q=ggplot(result,aes(i+2,RMSE))+geom_line(aes(color="RMSE"),size=1)+
    geom_line(data=result,aes(i+2,MAE,color="MAE"),size=1)+labs(x="Week number",y="RMSE / MAE ",color="
    ggtitle('2017')
  print(q)
```



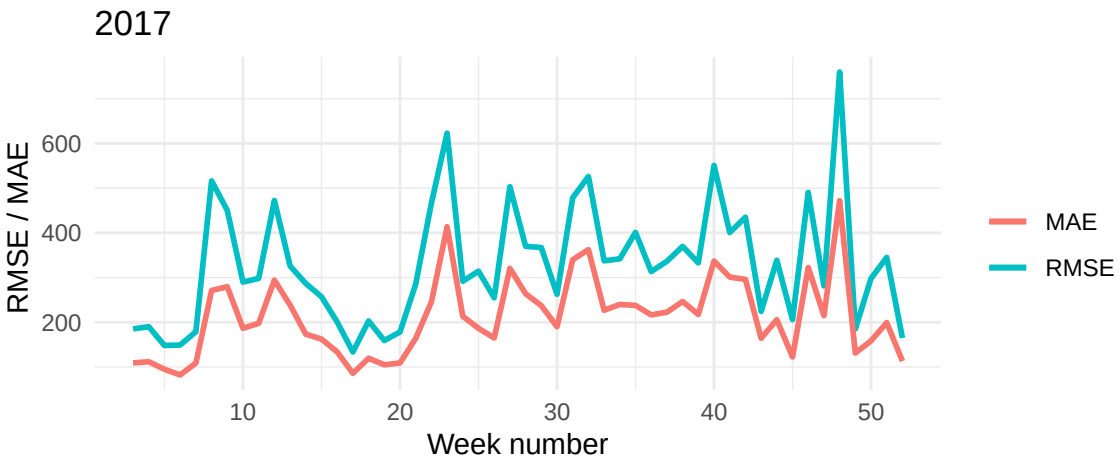


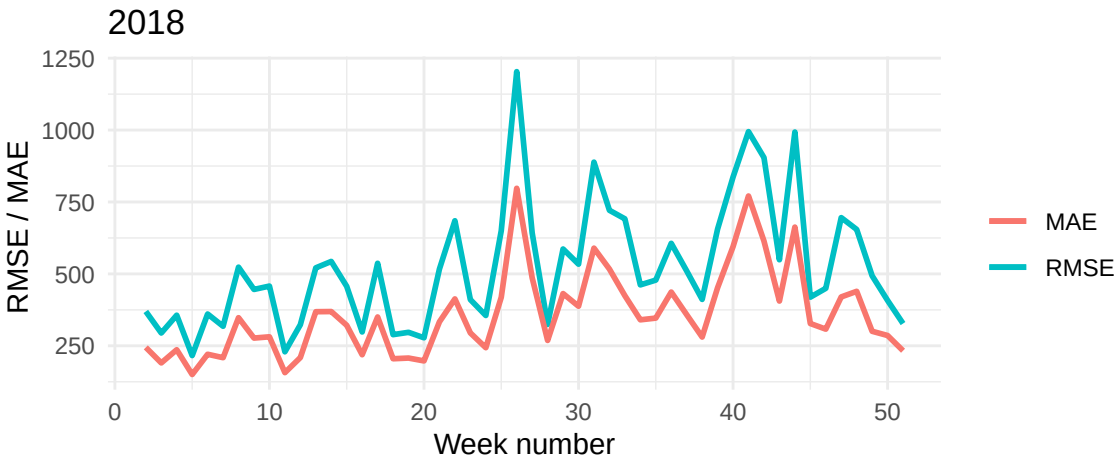


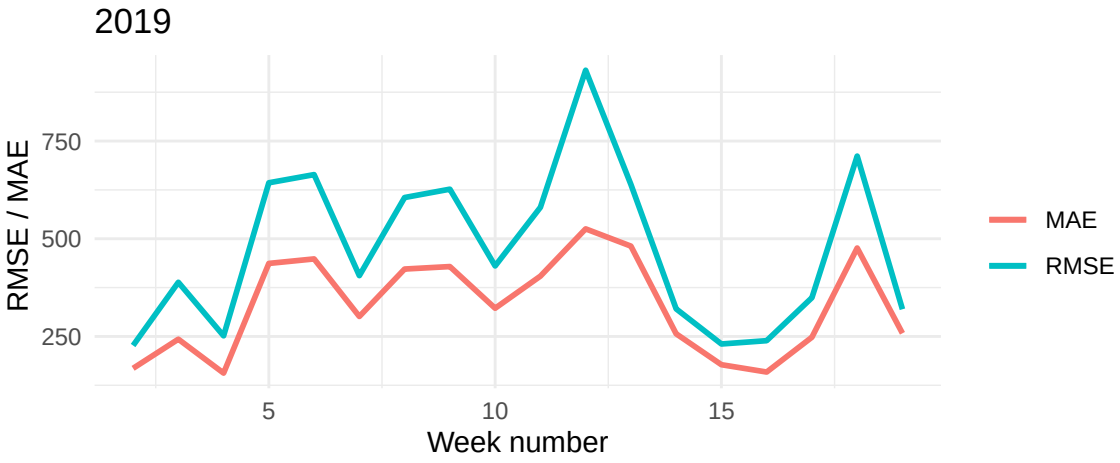
Dubuque

```
ames=na.omit(subset(dot,Branch=="Dubuque" & year=='2017'))
  magic_for(silent = TRUE)
  for (i in ames$week[1]:ames$week[1]:(ames$week[dim(ames)[1]]-2)){
    db=subset(ames,(week==i | week==i+1))
    test=subset(ames, week==i+2)
    test.db=data.frame(b=test$day,c=test$hour,d=test$month,e=test$week)
    test_w= test$w
    a=db$w
    b=db$day
    c=db$hour
    d=db$month
    e=db$week
    loessMod10 <- loess(lm(a~b+c+e), span=0.3, degree = 1, control = loess.control(surface = "direct"))
    p=predict(loessMod10, newdata = test.db )
    MAE=mae(test_w,p)
    RMSE=rmse(test_w,p)
    put(i+2,MAE,RMSE)
  }
```

```
result=magic_result_as_dataframe()
theme_set(theme_minimal())
q=ggplot(result,aes(i+2,RMSE))+geom_line(aes(color="RMSE"),size=1)+
  geom_line(data=result,aes(i+2,MAE,color="MAE"),size=1)+labs(x="Week number",y="RMSE / MAE ",color="
  ggtitle('2017')
print(q)
```

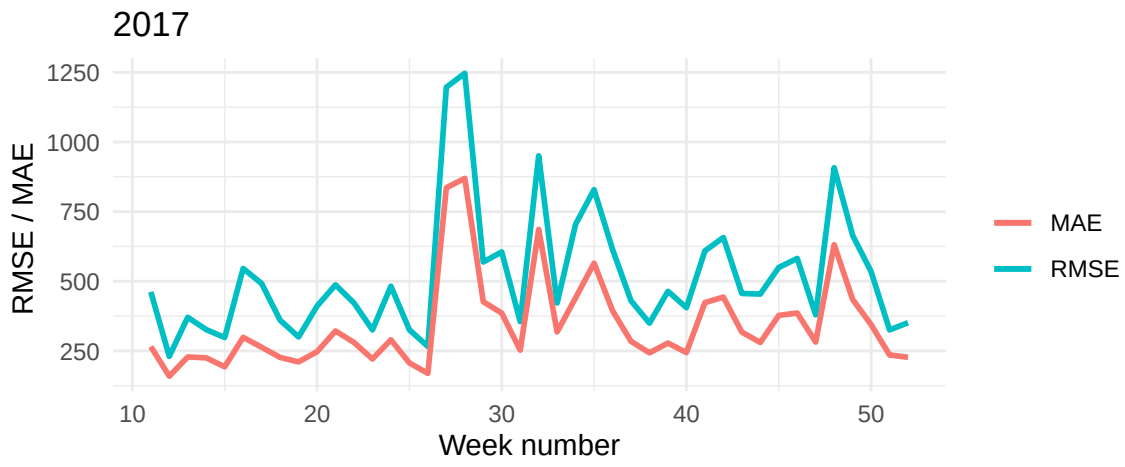


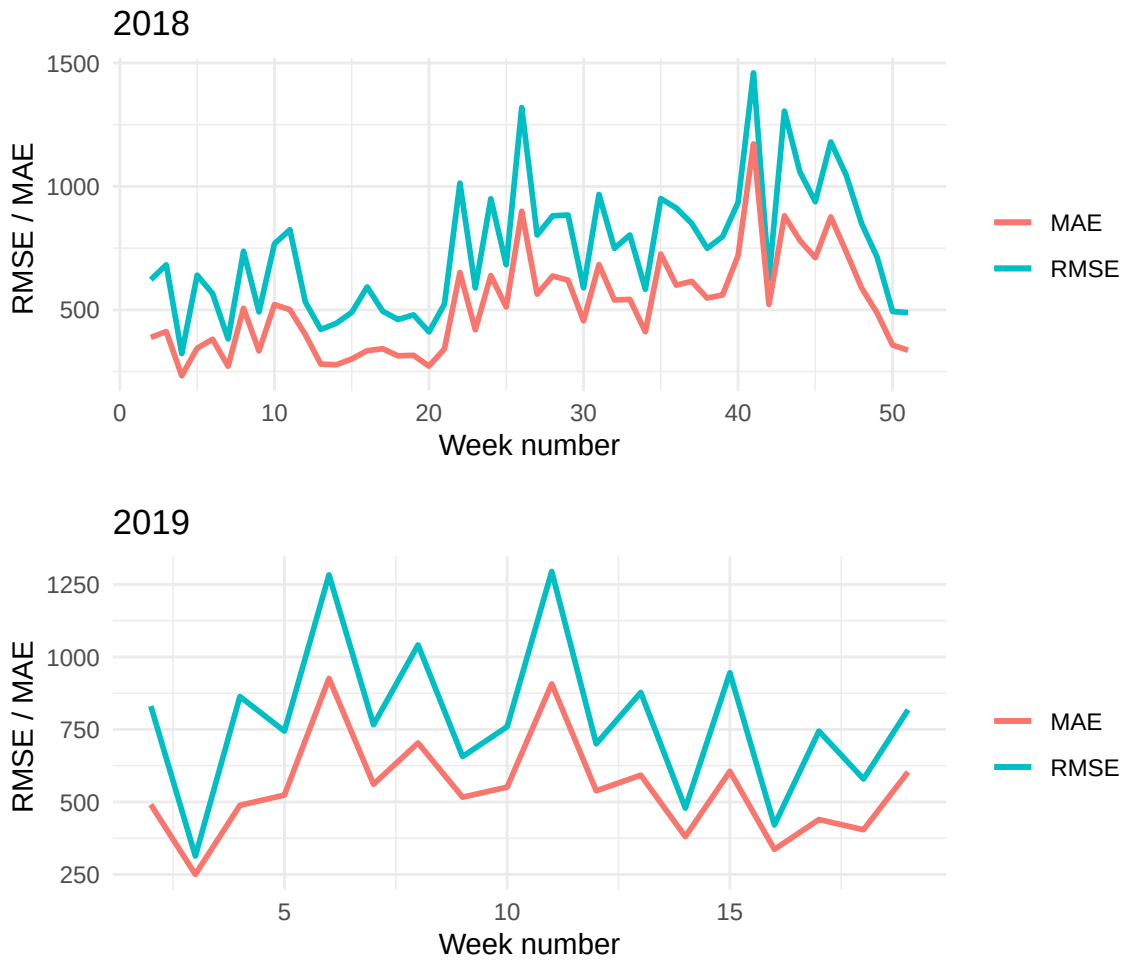




Burlington

```
ames=na.omit(subset(dot,Branch=="Burlington" & year=='2017'))
  magic_for(silent = TRUE)
  for (i in ames$week[1]:ames$week[1]:(ames$week[dim(ames)[1]]-2)){
    db=subset(ames,(week==i | week==i+1))
    test=subset(ames, week==i+2)
    test.db=data.frame(b=test$day,c=test$hour,d=test$month,e=test$week)
    test_w= test$w
    a=db$w
    b=db$day
    c=db$hour
    d=db$month
    e=db$week
    loessMod10 <- loess(lm(a~b+c+e), span=0.3, degree = 1, control = loess.control(surface = "direct"))
    p=predict(loessMod10, newdata = test.db )
    MAE=mae(test_w,p)
    RMSE=rmse(test_w,p)
    put(i+2,MAE,RMSE)
  }
  result=magic_result_as_dataframe()
  theme_set(theme_minimal())
  q=ggplot(result,aes(i+2,RMSE))+geom_line(aes(color="RMSE"),size=1)+
    geom_line(data=result,aes(i+2,MAE,color="MAE"),size=1)+labs(x="Week number",y="RMSE / MAE ",color="
    ggtitle('2017')
  print(q)
```





Conclusion

The LOESS regression gives satisfactory performance for Burlington and Dubuque. The MAE associated with this consistently below 450 seconds which means we miss our prediction by an margin 7 to 8 mins which is not vary large given the fact that we only time series data The Ames has higher MAE and RMSE values because it is an unique case. Iowa State University is located in the Ames so the flow of people in the DoT station is not consistent we can see that summer has the lowest waiting time. This indicates that many other factors other than time affect this station that's why we couldn't predict it accurately Most important conclusion is that time series data is not sufficient to predict the wait time at DoT stations. There can be many other factors behind the higher wait times other than time of the day. These high wait time can be because of absence of an server or nature of work of other people standing in the line. So the accuracy can be improved by inroducing more data such as nature of work of the people standing in line and using some complex tree based models.