ASSIGNMENT 2

Xiying Zhao(xiyingz2)

2023-03-21

Team members: “Crystal Wang (cw30)” “Xinyu Zhang (xinyuz6)”

```
setwd("C:/Users/zhaox/OneDrive/Desktop/UIUC/BADM 575/Assignment/hw2")
d = read.csv("BikeDemandDaily.csv")

colnames(d)

##  [1] "Index"         "year"          "month"         "day"
##  [5] "season"        "holiday"       "workingday"    "meanatemp"
##  [9] "maxatemp"      "minatemp"      "sdatemp"       "meanhumidity"
## [13] "maxhumidity"   "minhumidity"   "sdhumidity"    "meanwindspeed"
## [17] "maxwindspeed"  "minwindspeed"  "sdwindspeed"   "Casual"
## [21] "Registered"    "Total"

d$month = as.factor(d$month)
d$season = as.factor(d$season)
d$holiday = as.factor(d$holiday)
d$workingday = as.factor(d$workingday)
summary(d)

##      Index            year          month          day       season  holida
y
##  Min.   :  1.0   Min.   :1.0   1      : 38   Min.   : 1   1:114   0:443

##  1st Qu.:114.8   1st Qu.:1.0   2      : 38   1st Qu.: 5   2:114   1: 13

##  Median :228.5   Median :1.5   3      : 38   Median :10   3:114

##  Mean   :228.5   Mean   :1.5   4      : 38   Mean   :10   4:114

##  3rd Qu.:342.2   3rd Qu.:2.0   5      : 38   3rd Qu.:15

##  Max.   :456.0   Max.   :2.0   6      : 38   Max.   :19

##                                (Other):228
##  workingday   meanatemp        maxatemp         minatemp        sdatem
p
##  0:145      Min.   : 5.083   Min.   : 8.335   Min.   : 0.76   Min.   :0.
000
##  1:311      1st Qu.:16.989   1st Qu.:21.210   1st Qu.:12.12   1st Qu.:
1.981
##             Median :24.495   Median :30.305   Median :20.45   Median :
```

```
2.689
##             Mean   :23.607   Mean   :27.764   Mean   :19.54   Mean   :2.
726
##             3rd Qu.:30.089   3rd Qu.:33.335   3rd Qu.:25.95   3rd Qu.:
3.390
##             Max.   :40.246   Max.   :45.455   Max.   :35.60   Max.   :5.
840
##

##   meanhumidity    maxhumidity     minhumidity     sdhumidity
##  Min.   : 0.00   Min.   :  0.00   Min.   : 0.00   Min.   : 0.000
##  1st Qu.:51.22   1st Qu.: 72.75   1st Qu.:33.00   1st Qu.: 9.025
##  Median :61.85   Median : 83.00   Median :40.50   Median :12.314
##  Mean   :61.89   Mean   : 81.52   Mean   :42.84   Mean   :12.484
##  3rd Qu.:71.84   3rd Qu.: 93.00   3rd Qu.:51.00   3rd Qu.:15.687
##  Max.   :97.04   Max.   :100.00   Max.   :88.00   Max.   :31.648
##
##  meanwindspeed    maxwindspeed     minwindspeed     sdwindspeed
##  Min.   : 1.50   Min.   : 8.998   Min.   : 0.000   Min.   : 2.245
##  1st Qu.: 9.20   1st Qu.:19.001   1st Qu.: 0.000   1st Qu.: 4.864
##  Median :12.15   Median :23.999   Median : 0.000   Median : 5.877
##  Mean   :12.81   Mean   :24.542   Mean   : 2.439   Mean   : 6.205
##  3rd Qu.:15.61   3rd Qu.:30.003   3rd Qu.: 6.003   3rd Qu.: 7.327
##  Max.   :34.00   Max.   :56.997   Max.   :20.000   Max.   :13.709
##
##      Casual          Registered       Total
##  Min.   :  11.0   Min.   : 516   Min.   : 635
##  1st Qu.: 324.2   1st Qu.:2720   1st Qu.:3329
##  Median : 725.0   Median :3726   Median :4612
##  Mean   : 864.9   Mean   :3733   Mean   :4598
##  3rd Qu.:1150.0   3rd Qu.:4826   3rd Qu.:6008
##  Max.   :3410.0   Max.   :6949   Max.   :8736
##
```

Part A. Disaggregated Demand Forecasting and aggregating up

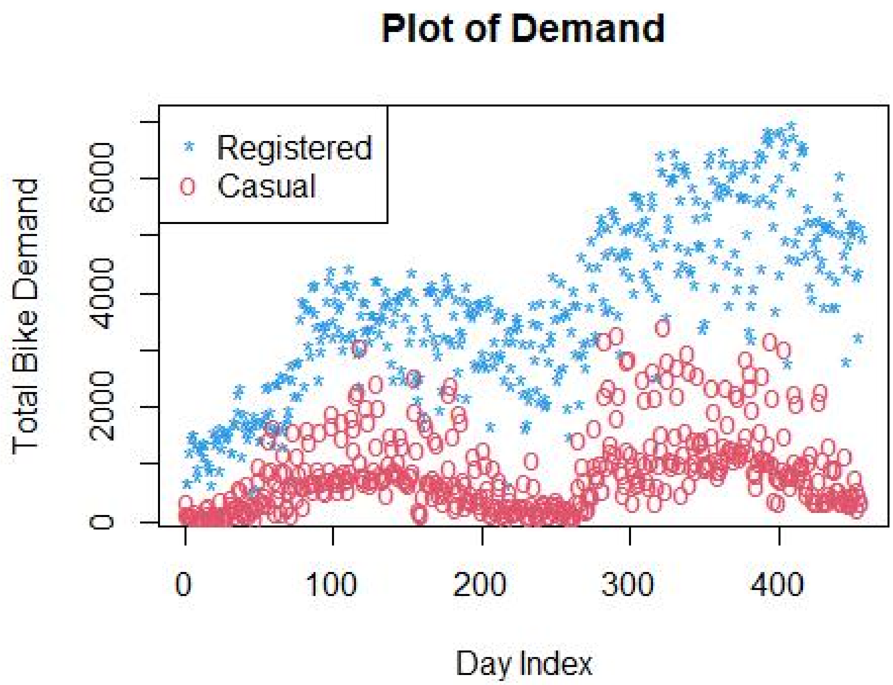
A1. Graphical comparison of registered and casual demand.

i. Plot the daily registered and causal demand on the y-axis and the day index on the x-axis.Comment on the observations.

(Hint. Think of the patterns of demand for the two groups of customers, which group shows more dispersion around the daily mean demand? Think of the time trends. Where do you observe greater growth? …)

```
plot(d$Index, d$Registered, pch = "*", col = 4, xlab = "Day Index",
     ylab = "Total Bike Demand", main = "Plot of Demand",
     ylim = c(200,7000))
points(d$Index, d$Casual, pch = "o", col = 2)
```

```
legend("topleft", c("Registered", "Casual"), pch = c("*", "o"),
       col = c(4,2))
```



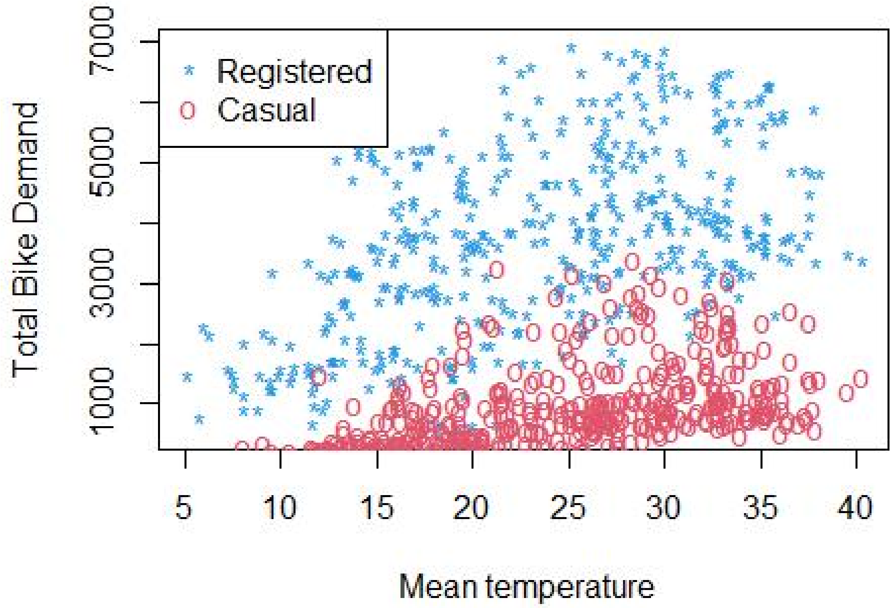

```
cor(d$Index, d$Casual)

## [1] 0.3024345

cor(d$Index, d$Registered)

## [1] 0.7738318
```

Observations: 1. Thinking of the patterns of demand for the two groups of customers, causal demand shows more dispersion around the daily mean demand, and it is lower than registered demand. 2. Thinking of the time trends, registered demand has greater growth trend. 3. In general, demand for bike peaks at around day 150 and 350 for both registered and casual groups, with the greater growth being observed between day 0 to 150 and day 250 to 400.

ii. Plot the following scatter diagrams

a. Mean temperature versus registered and causal demand.

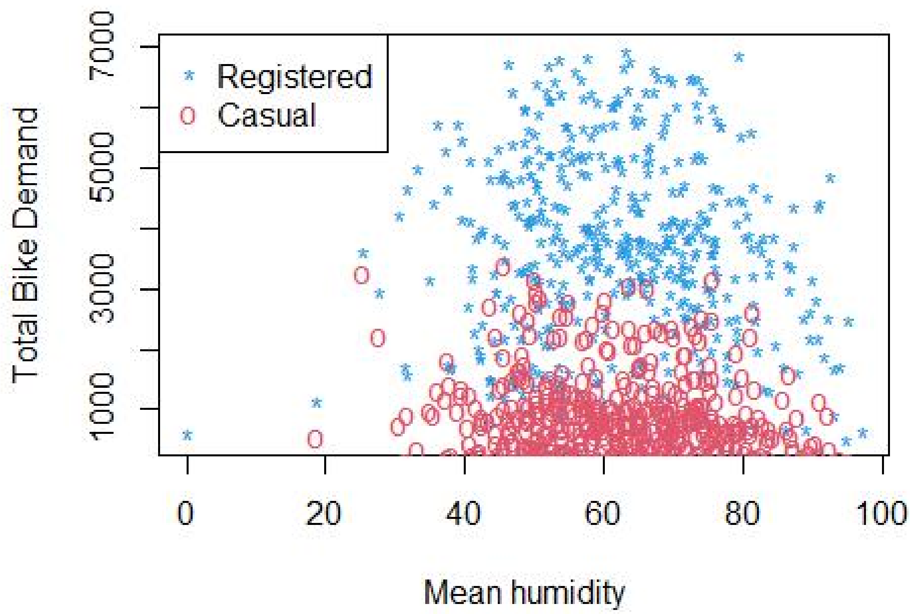
```
plot(d$meanatemp, d$Registered, pch = "*", col = 4, xlab = " Mean temper
ature",
     ylab = "Total Bike Demand")
points(d$meanatemp, d$Casual, pch = "o", col = 2)
legend("topleft", c("Registered", "Casual"), pch = c("*", "o"),
       col = c(4,2))
```




```
cor(d$meanatemp, d$Casual)

## [1] 0.5483651

cor(d$meanatemp, d$Registered)

## [1] 0.5102346
```

b. Mean humidity versus registered and causal demand.

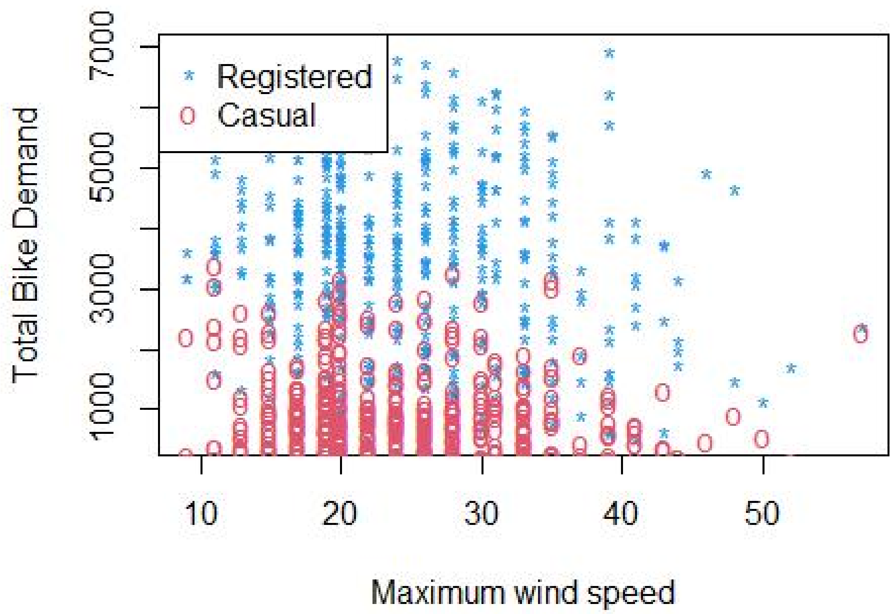
```
plot(d$meanhumidity, d$Registered, pch = "*", col = 4, xlab = "Mean humi
dity",
     ylab = "Total Bike Demand")
points(d$meanhumidity, d$Casual, pch = "o", col = 2)
legend("topleft", c("Registered", "Casual"), pch = c("*", "o"),
       col = c(4,2))
```




```
cor(d$meanhumidity, d$Casual)

## [1] -0.08329068

cor(d$meanhumidity, d$Registered)

## [1] -0.05748222
```

c. Maximum wind speed versus registered and casual demand.

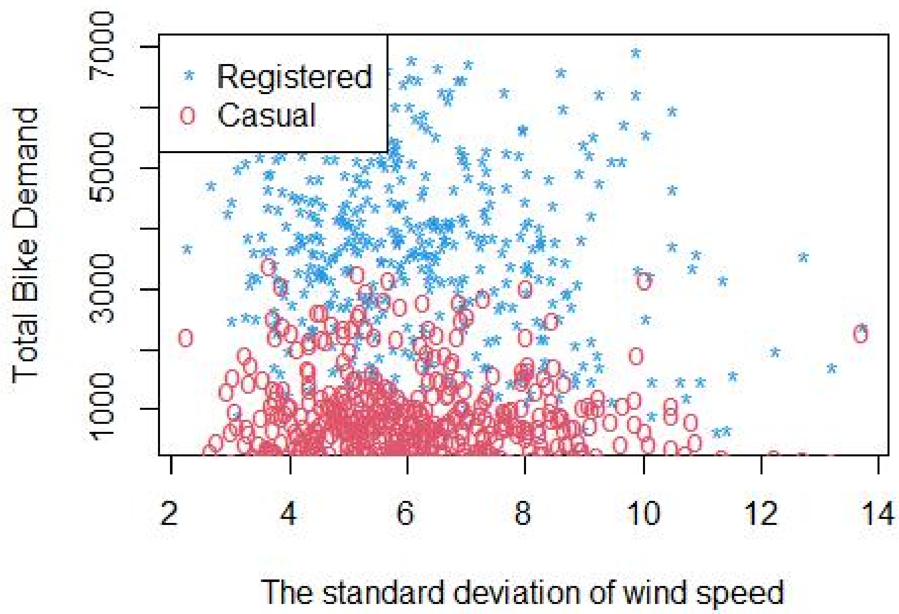
```
plot(d$maxwindspeed, d$Registered, pch = "*", col = 4, xlab = "Maximum w
ind speed",
     ylab = "Total Bike Demand")
points(d$maxwindspeed, d$Casual, pch = "o", col = 2)
legend("topleft", c("Registered", "Casual"), pch = c("*", "o"),
       col = c(4,2))
```




```
cor(d$maxwindspeed, d$Casual)

## [1] -0.1529884

cor(d$maxwindspeed, d$Registered)

## [1] -0.2254846
```

d. The standard deviation of wind speed versus registered and causal demand.

```
plot(d$sdwindspeed, d$Registered, pch = "*", col = 4, xlab = "The standa
rd deviation of wind speed",ylab = "Total Bike Demand")
points(d$sdwindspeed, d$Casual, pch = "o", col = 2)
legend("topleft", c("Registered", "Casual"), pch = c("*", "o"),
       col = c(4,2))
```




```
cor(d$sdwindspeed, d$Casual)

## [1] -0.1240596

cor(d$sdwindspeed, d$Registered)

## [1] -0.1847814
```

e. Comment on our observations.

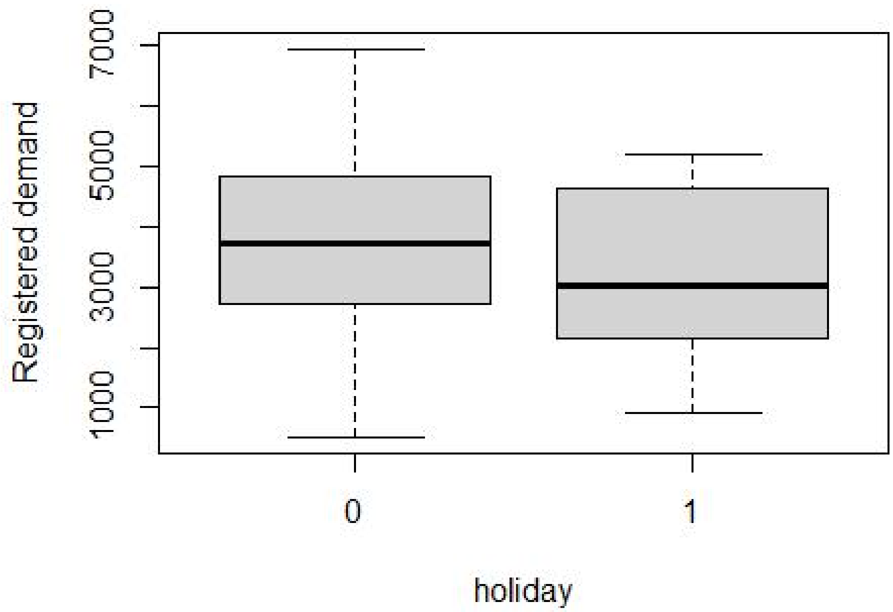
(Think of which factors seem to be more correlated with demand and which type of demand. What could be the potential reasons for what you observe?)

Observations: 1. Mean temperature has a positive impact on demand. The correlation is 0.5483651 for Causal and 0.5102346 for Registered. The reason behind might be that the nicer weather prompts people to go out more. And it contributes to the increase of causal demand. 2. Mean humidity has a slight negative impact on demand. The correlation is -0.08329068 for Causal and -0.05748222 for Registered. 3. Maximum wind speed has a negative impact on demand. The correlation is -0.1529884 for Causal and -0.2254846 for Registered. The standard deviation of wind speed has a negative impact on demand. The correlation is -0.1240596 for Causal and -0.1847814 for Registered. The reason behind might be that the windy weather makes cycling more uncomfortable. And registered demand are more likely to be influenced since people may change to other transportations.

iii Create box plots for the following. (5 Points)

a. Registered demand versus holiday.

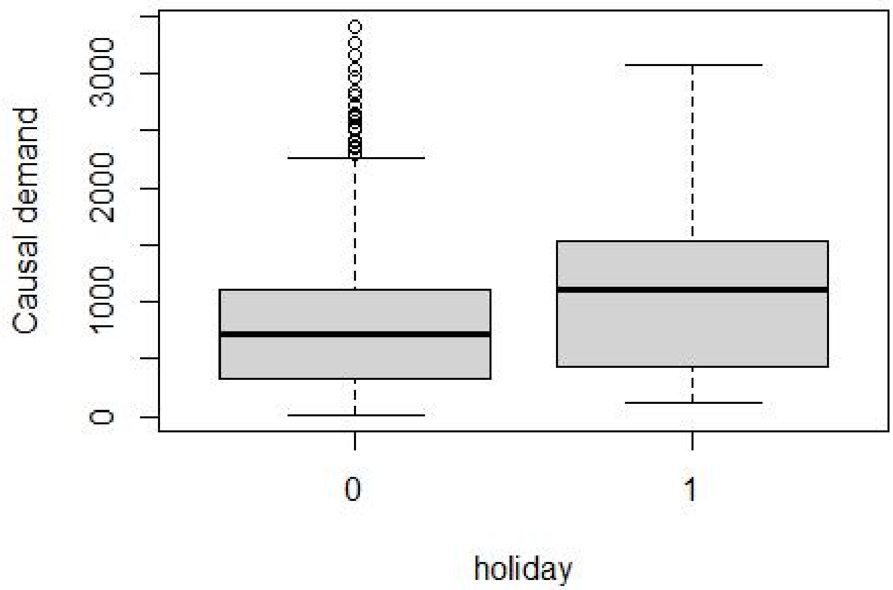
```
boxplot(d$Registered~d$holiday, xlab = "holiday",ylab = "Registered dem
and")
```



b.

Causal demand versus holiday.

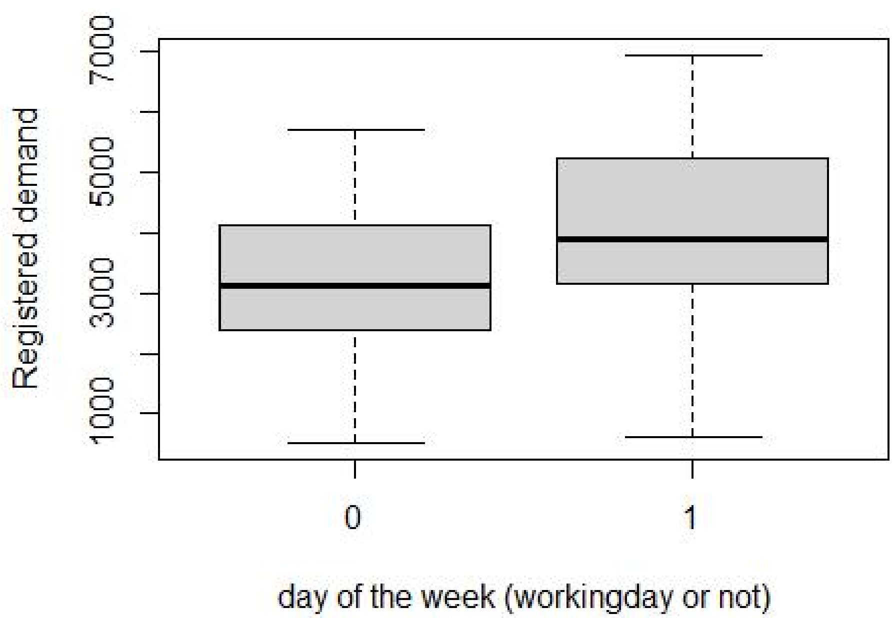
```
boxplot(d$Casual~d$holiday, xlab = "holiday",ylab = "Causal demand")
```



####

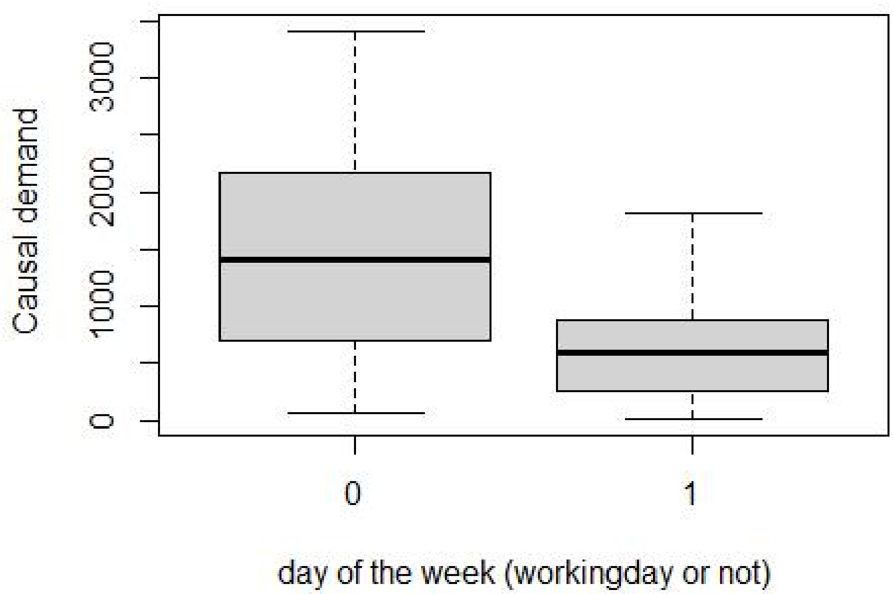
c. Registered demand versus day of the week.

```
boxplot(d$Registered~d$workingday, xlab = "day of the week (workingday o
r not)",ylab = "Registered demand")
```



d. Causal demand versus day of the week.

```
boxplot(d$Casual~d$workingday, xlab = "day of the week (workingday or no
t)",ylab = " Causal demand")
```



e. Comment on your observations.

(Think of which factors seem to be more correlated with demand and which type of demand. What could be the potential reasons for what you observe?)

Observations: In summary, we can observe that holidays and non-working days negatively effect the registered demand but positively effect casual demand. The reasoning behind this could be that registered group rely on bikes for their daily commute, whereas casual group do not. On holidays, casual group have higher demand for bikes as they are likely enjoying a ride and doing something outside of their normal routine.

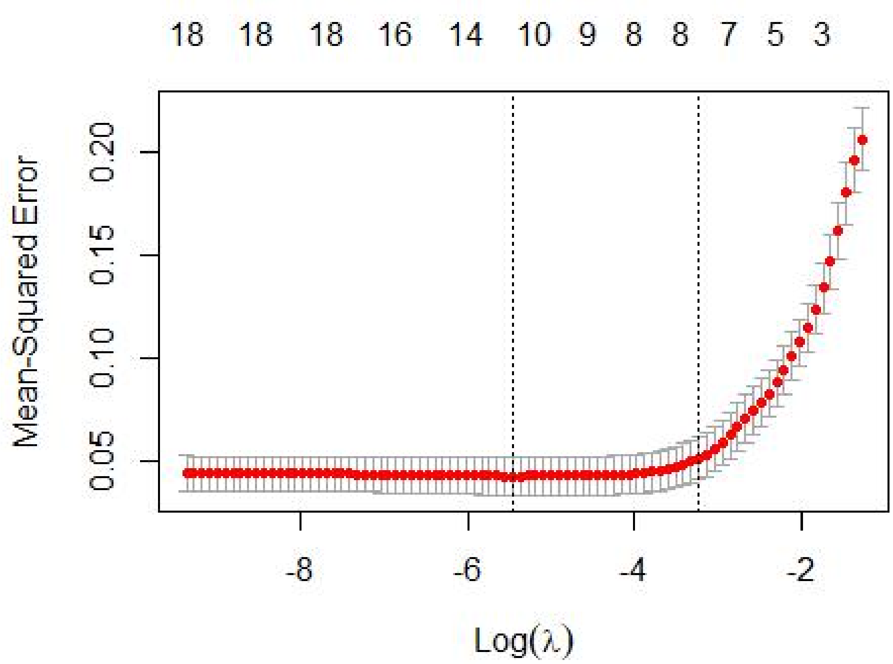
A2. Create LASSO models for registered demand and causal demand separately.

i. Split the sample by day Index. All observations less than or equal to day index 300 are in the training set and the remaining is in the testing set.

```
ind = c(1:300)
train = d[ind,]
test = d[-ind,]
```

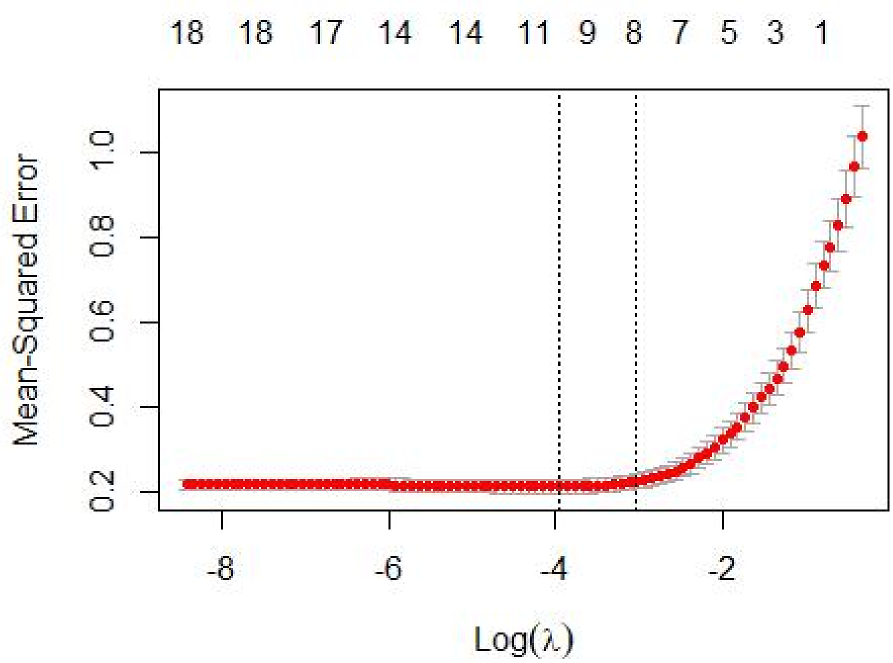
ii. Follow the class code to set up the cross-validation for registered and casual demand separately. Report the optimal cross-validation penalty for both models. Use the training set only.

```
library(glmnet)

## Warning: package 'glmnet' was built under R version 4.2.3

## Loading required package: Matrix

## Loaded glmnet 4.1-6

# Registered
m1 = lm(log(Registered)~Index+season+holiday+workingday, data = d)
x1 = model.matrix(m1)
x1 = cbind(x1, as.matrix(d[,c(8:19)]))
y1 = log(d$Registered)
trainx1 = x1[ind,]
trainy1 = y1[ind]
testx1 = x1[-ind,]
testy1 = y1[-ind]
#Cross validation of penalty parameter.
l1 = cv.glmnet(trainx1, trainy1)
plot(l1)
```




```
print(l1$lambda.min)

## [1] 0.004266442
```

```
# Casual
m2 = lm(log(Casual)~Index+season+holiday+workingday, data = d)
x2 = model.matrix(m2)
x2 = cbind(x2, as.matrix(d[,c(8:19)]))
y2 = log(d$Casual)
trainx2 = x2[ind,]
trainy2 = y2[ind]
testx2 = x2[-ind,]
testy2 = y2[-ind]
#Cross validation of penalty parameter.
l2 = cv.glmnet(trainx2, trainy2)
plot(l2)
```




```
print(l2$lambda.min)

## [1] 0.01911986
```

iii. Construct the final LASSO model using the optimal penalty found at the cross-validation stage. Comment on the selected variables. Again, use the training set only.

```
#Registered Final model
l1f = glmnet(trainx1, trainy1, lambda = l1$lambda.min)
l1f$beta

## 19 x 1 sparse Matrix of class "dgCMatrix"
##                          s0
## (Intercept)    .
## Index         0.003139861
```

```
## season2       0.165628413
## season3       0.012254524
## season4       0.051562552
## holiday1       .
## workingday1    0.223000397
## meanatemp      .
## maxatemp       .
## minatemp      0.027172770
## sdatemp       0.016080776
## meanhumidity  -0.001866801
## maxhumidity    .
## minhumidity   -0.004308781
## sdhumidity     0.007663606
## meanwindspeed  .
## maxwindspeed  -0.009367549
## minwindspeed   .
## sdwindspeed    .
```

```
#Casual Final model
l2f = glmnet(trainx2, trainy2, lambda = l2$lambda.min)
l2f$beta
```

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept)    .
## Index         0.0022886520
## season2       0.3828250649
## season3        .
## season4       0.0060466802
## holiday1     -0.0586304403
## workingday1   -0.9449170548
## meanatemp      0.0685796946
## maxatemp       .
## minatemp      0.0158729697
## sdatemp       0.0752276791
## meanhumidity  -0.0005923095
## maxhumidity    .
## minhumidity   -0.0105232919
## sdhumidity     .
## meanwindspeed  .
## maxwindspeed  -0.0127076339
## minwindspeed   .
## sdwindspeed    .
```

Observations: 1. Since the lasso model utilizes a regularization technique to shrink coefficients to 0, the remaining non-zero variables are the ones that are truly relevant to predicting the total bike demand. 2. For registered group, the selected predictors are index, seasons (2,3,4), workingday1, min and sdatemp, mean and minhumidity, sdhumidity and maxwindspeed. Most predictors have coefficients are very close to 0, but workingday1 and season2 have larger coefficients that indicate high importance to the prediction of bike demand. 3. For casual group, the selected predictors are index, seasons (2,3,4), holiday1, workingday1, (mean, min, have the largest positive coefficients; whereas workingday1 and holiday1 have the largest negative coefficients. This indicates the season and temperature positively effects casual group demand for bikes but working days and holidays negatively impact the demand.

iv. Predict the testing set. Report the Root Mean Squared Error (RMSE) on the testing set.

```
#Registered Predict the the future.
p1 = predict(l1f, newx = testx1)
#Casual Predict the the future.
p2 = predict(l2f, newx = testx2)

#Registered RMSE
sqrt(mean(testy1 - p1)^2)

## [1] 0.2654592

#Casual RMSE
sqrt(mean(testy2 - p2)^2)

## [1] 0.1996924
```

v. Add the predictions of registered and casual demand to get the prediction for total demand. Report the RMSE.

```
# total demand
p_total = log(exp(p1)+exp(p2))
test_total = log(test$Total)
sqrt(mean(test_total - p_total)^2)

## [1] 0.2640262
```

A3. Time series prediction models – Auto Regressive Moving Average (ARMA).

i. Using the variables selected by the LASSO Models create linear regression models using the training set for both registered and casual demand. Report the summary of the regression and comment on the important predictors for both registered and causal demand.

```
l1f$beta

## 19 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept)    .
## Index         0.003139861
## season2       0.165628413
## season3       0.012254524
## season4       0.051562552
## holiday1      .
```

```
## workingday1    0.223000397
## meanatemp      .
## maxatemp       .
## minatemp       0.027172770
## sdatemp        0.016080776
## meanhumidity  -0.001866801
## maxhumidity    .
## minhumidity   -0.004308781
## sdhumidity     0.007663606
## meanwindspeed  .
## maxwindspeed  -0.009367549
## minwindspeed   .
## sdwindspeed    .
```

```
# Registered
mr = lm(log(Registered)~Index+season+workingday+
        minatemp+sdatemp+meanhumidity+minhumidity+sdhumidity+
        maxwindspeed, data = train)
summary(mr)
```

```
##
## Call:
## lm(formula = log(Registered) ~ Index + season + workingday +
##     minatemp + sdatemp + meanhumidity + minhumidity + sdhumidity +
##     maxwindspeed, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.21130 -0.09192  0.02324  0.11125  0.41404
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.1513919  0.0860988  83.060  < 2e-16 ***
## Index         0.0031761  0.0001356  23.420  < 2e-16 ***
## season2       0.1965208  0.0377998   5.199 3.81e-07 ***
## season3       0.0529514  0.0521800   1.015   0.3111
## season4       0.0810657  0.0351236   2.308   0.0217 *
## workingday1   0.2316486  0.0239695   9.664  < 2e-16 ***
## minatemp      0.0263638  0.0024623  10.707  < 2e-16 ***
## sdatemp       0.0197402  0.0116499   1.694   0.0913 .
## meanhumidity -0.0039821  0.0031460  -1.266   0.2066
## minhumidity  -0.0024609  0.0033452  -0.736   0.4625
## sdhumidity    0.0104562  0.0045008   2.323   0.0209 *
## maxwindspeed -0.0099979  0.0014108  -7.087 1.06e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1913 on 288 degrees of freedom
## Multiple R-squared:  0.8306, Adjusted R-squared:  0.8241
## F-statistic: 128.3 on 11 and 288 DF,  p-value: < 2.2e-16
```

```
l2f$beta

## 19 x 1 sparse Matrix of class "dgCMatrix"
##                         s0
## (Intercept)    .
## Index         0.0022886520
## season2       0.3828250649
## season3       .
## season4       0.0060466802
## holiday1     -0.0586304403
## workingday1  -0.9449170548
## meanatemp     0.0685796946
## maxatemp      .
## minatemp      0.0158729697
## sdatemp       0.0752276791
## meanhumidity  -0.0005923095
## maxhumidity    .
## minhumidity   -0.0105232919
## sdhumidity     .
## meanwindspeed  .
## maxwindspeed  -0.0127076339
## minwindspeed   .
## sdwindspeed    .

# Casual
mc = lm(log(Casual)~Index+season+holiday+workingday+
        meanatemp+minatemp+sdatemp+meanhumidity+minhumidity+
        maxwindspeed, data = train)
summary(mc)

##
## Call:
## lm(formula = log(Casual) ~ Index + season + holiday + workingday +
##     meanatemp + minatemp + sdatemp + meanhumidity + minhumidity +
##     maxwindspeed, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.72185 -0.24098  0.00567  0.26417  1.00106
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.3304490  0.1956825  27.240  < 2e-16 ***
## Index         0.0024038  0.0003119   7.708 2.11e-13 ***
## season2       0.5075101  0.0876852   5.788 1.87e-08 ***
## season3       0.1539585  0.1214600   1.268  0.20598
## season4       0.1161598  0.0806987   1.439  0.15112
## holiday1     -0.2194656  0.1631989  -1.345  0.17976
## workingday1  -1.0065754  0.0567408 -17.740  < 2e-16 ***
## meanatemp     0.0195046  0.0313937   0.621  0.53490
```

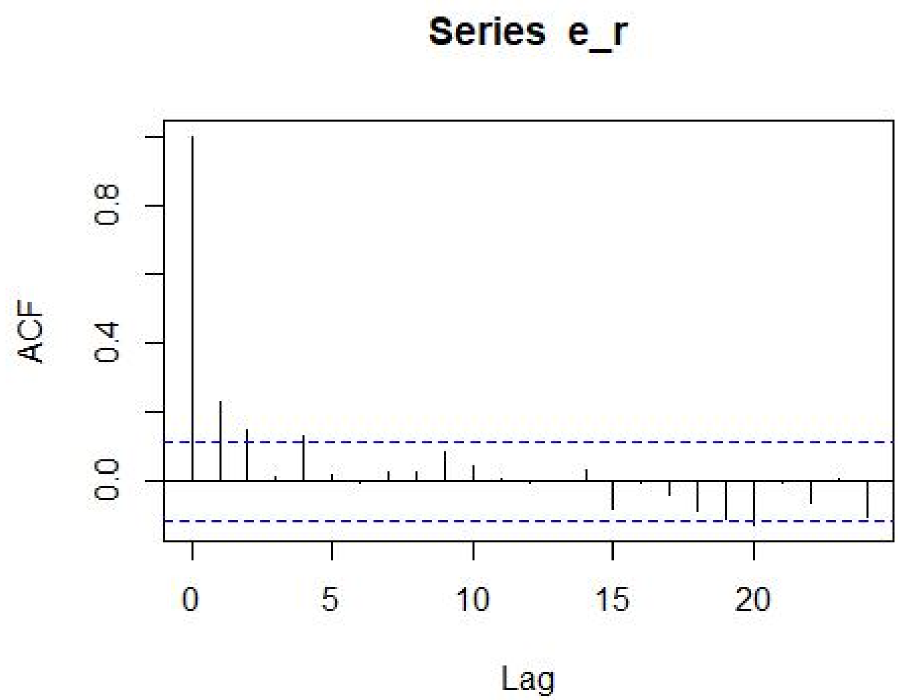
```
## minatemp      0.0615685  0.0307285   2.004  0.04605 *
## sdatemp       0.1526794  0.0484223   3.153  0.00179 **
## meanhumidity -0.0031855  0.0042513  -0.749  0.45429
## minhumidity  -0.0100404  0.0042318  -2.373  0.01832 *
## maxwindspeed -0.0149466  0.0032538  -4.594 6.53e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4398 on 287 degrees of freedom
## Multiple R-squared:  0.8222, Adjusted R-squared:  0.8148
## F-statistic: 110.6 on 12 and 287 DF,  p-value: < 2.2e-16
```

Observations: 1. For registered group, the most significant predictors are index, intercept, season2, workingday1, minatemp, and maxwindspeed(with 99% level of significance), in which maxwindspeed negatively effects total bike demand, whereas the other predictors positively effect demand, especially season2 and workingday1.

2. For casual group, the most significant(with 99% level of significance) predictors are the intercept(which means that the model might not contain good independent variables), index, season2, workingday1, and maxwindseed. Among the most significant predictors, season2 have the largest positive coefficient; whereas maxwindspeed and workingday1 have the largest negative coefficients. This indicates the season positively effects casual group demand for bikes but working days and maximum windspeed negatively impact the demand.

ii. Use the residuals of the regression models to plot the Autocorrelation Function for both types of demand (ACF). Which demand type demonstrates greater auto-correlation? Why?

```
# Registered
e_r = mr$residuals
ar_r = acf(e_r)
plot(ar_r)
```

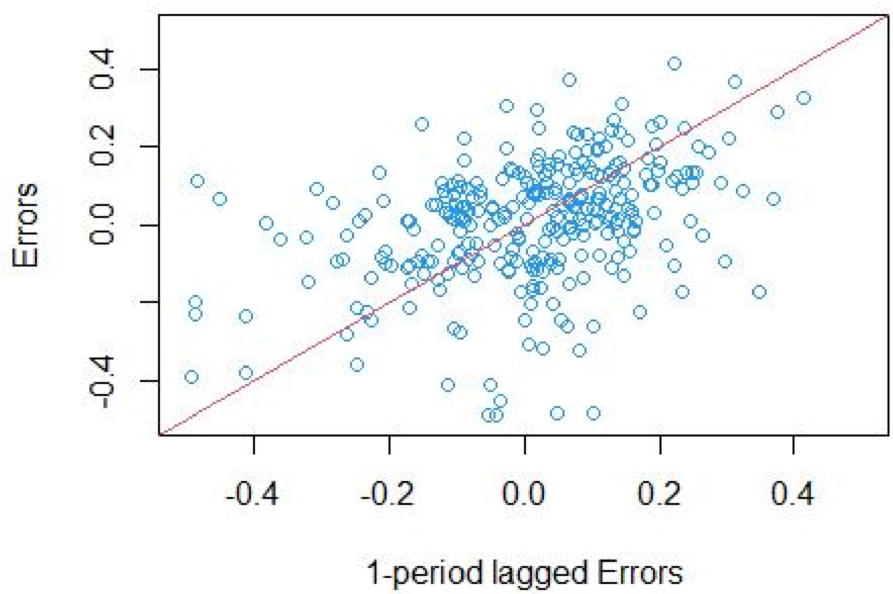



```
ar_r$acf[1:10]

##  [1]  1.000000000  0.232496890  0.146639219  0.013440368  0.129867846
##  [6]  0.021329332 -0.006927451  0.027769258  0.027653102  0.082963488
```

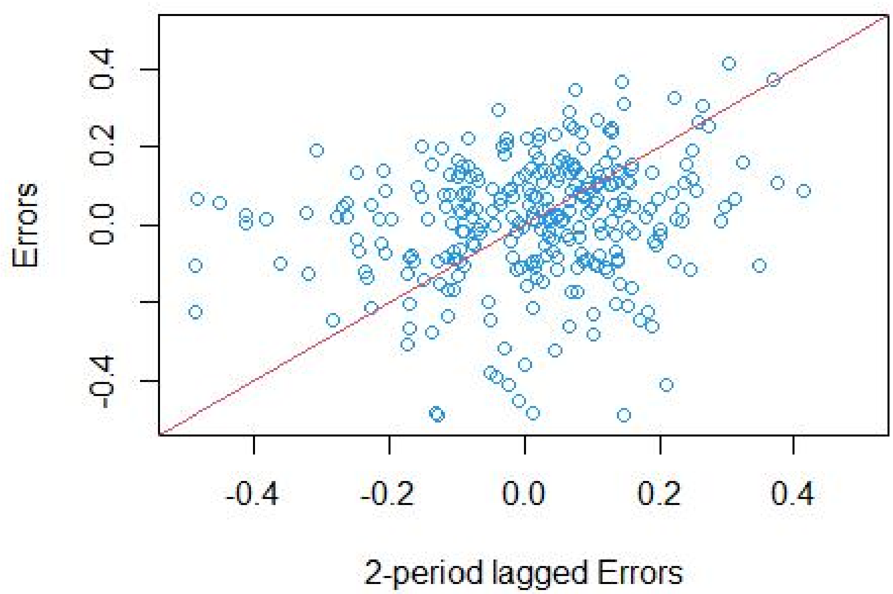
Observations:

Errors are more correlated from period 1 to 2. As we can observe the autocorrelation of order 1 of the error is 0.23. The autocorrelation of order 2 is 0.15. This can be understood by plotting the errors with the corresponding one-period and two-period lagged errors as shown below.

```
#Store the residuals
tr = ts(mr$residuals)
#Create one-period and two-period lagged vector.
tr1 = lag(tr,1L)
tr2 = lag(tr, 2L)

#  plot the Autocorrelation Function
plot(tr1,tr,xlab = "1-period lagged Errors",
     ylab = "Errors",xlim = c(-0.5,0.5),
     ylim=c(-0.5,0.5),col=4)
abline(lm(tr~tr1),col=2)
```



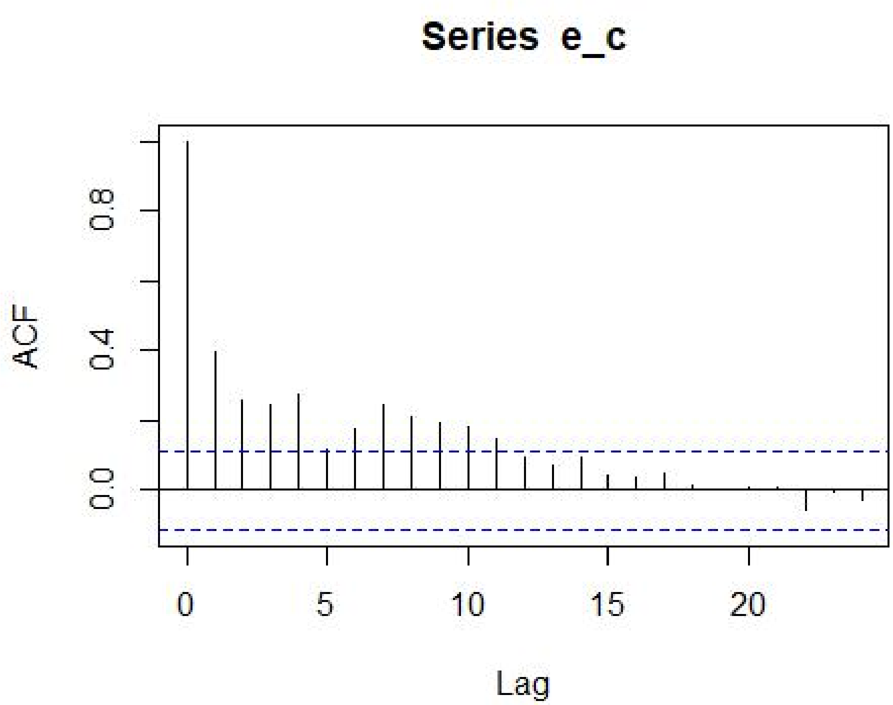

```
plot(tr2,tr,xlab = "2-period lagged Errors",
     ylab = "Errors",xlim = c(-0.5,0.5),
     ylim=c(-0.5,0.5),col=4)
abline(lm(tr~tr2),col=2)
```




```
# auto-regression coefficient
lr1 = ar.ols(tr,
       order.max = 1,
       demean = F,
       intercept = T)
lr1

##
## Call:
## ar.ols(x = tr, order.max = 1, demean = F, intercept = T)
##
## Coefficients:
##      1
## 0.2331
##
## Intercept: 0.001185 (0.01048)
##
## Order selected 1  sigma^2 estimated as  0.03281

lr2 = ar.ols(tr,
       order.max = 2,
       demean = F,
       intercept = T)
lr2

##
## Call:
```

```
## ar.ols(x = tr, order.max = 2, demean = F, intercept = T)
##
## Coefficients:
##      1       2
## 0.1936  0.1018
##
## Intercept: 0.002555 (0.01037)
##
## Order selected 2  sigma^2 estimated as  0.03202
```

```
# Casual

e_c = mc$residuals
ar_c = acf(e_c)
plot(ar_c)
```



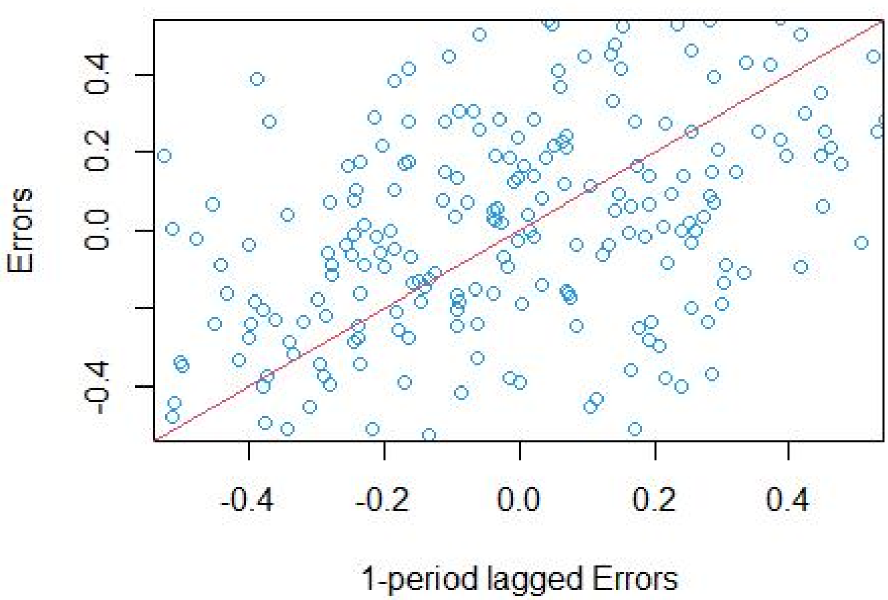

```
ar_c$acf[1:10]
```

```
##  [1] 1.0000000 0.3967231 0.2578667 0.2451766 0.2769184 0.1190487 0.17
77531
##  [8] 0.2432626 0.2082756 0.1914882
```

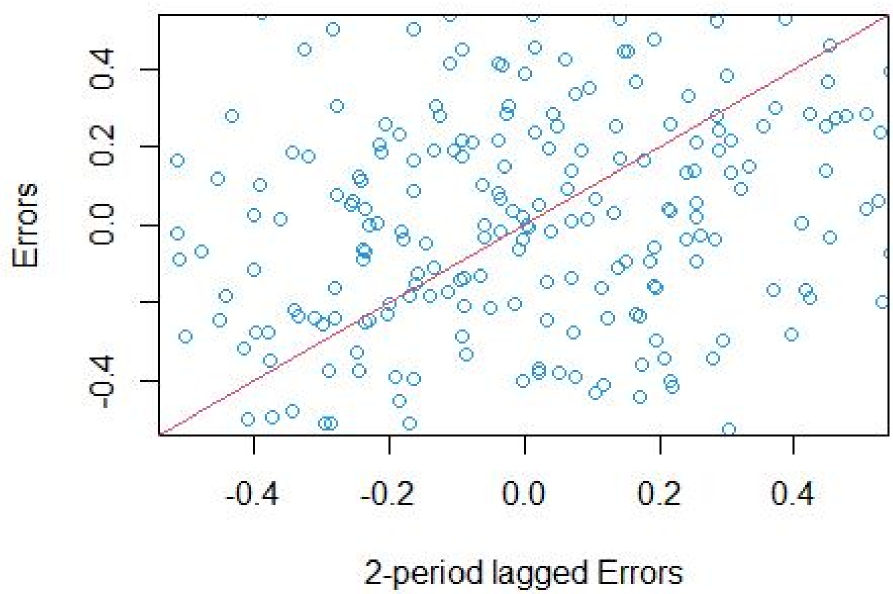
Observations:

Errors are correlated from period 1 to 10. As we can observe the autocorrelation of order 1 of the error is 0.40. The autocorrelation of order 2 is 0.26. This can be understood by plotting the errors with the corresponding one-period and two-period lagged errors as shown below.

```
#Store the residuals
tc = ts(mc$residuals)
#Create one-period and two-period lagged vector.
tc1 = lag(tc,1L)
tc2 = lag(tc, 2L)

#  plot the Autocorrelation Function
plot(tc1,tc,xlab = "1-period lagged Errors",
     ylab = "Errors",xlim = c(-0.5,0.5),
     ylim=c(-0.5,0.5),col=4)
abline(lm(tc~tc1),col=2)
```




```
plot(tc2,tc,xlab = "2-period lagged Errors",
     ylab = "Errors",xlim = c(-0.5,0.5),
     ylim=c(-0.5,0.5),col=4)
abline(lm(tc~tc2),col=2)
```




```
# auto-regression coefficient
lc1 = ar.ols(tc,
       order.max = 1,
       demean = F,
       intercept = T)
lc1

##
## Call:
## ar.ols(x = tc, order.max = 1, demean = F, intercept = T)
##
## Coefficients:
##     1
## 0.397
##
## Intercept: -0.0004162 (0.02287)
##
## Order selected 1  sigma^2 estimated as  0.1564

lc2 = ar.ols(tc,
       order.max = 2,
       demean = F,
       intercept = T)
lc2

##
## Call:
```

```
## ar.ols(x = tc, order.max = 2, demean = F, intercept = T)
##
## Coefficients:
##      1       2
## 0.3506  0.1190
##
## Intercept: 0.002788 (0.02255)
##
## Order selected 2  sigma^2 estimated as  0.1515
```

iii. Fit an ARMA(2,2) time-series model on the training set for both registered and casual demand. Report the summary.

```
#Load the packages
library(nlme)
library(car)

## Warning: package 'car' was built under R version 4.2.3

## Loading required package: carData

## Warning: package 'carData' was built under R version 4.2.3

# Registered
# Create a GLS estimate
mr1 = gls(log(Registered)~Index+season+workingday+
        minatemp+sdatemp+meanhumidity+minhumidity+sdhumidity+
        maxwindspeed, data = train,
        correlation = corARMA(p=2, q=2), method = "ML")
summary(mr1)

## Generalized least squares fit by maximum likelihood
##   Model: log(Registered) ~ Index + season + workingday + minatemp + s
datemp +      meanhumidity + minhumidity + sdhumidity + maxwindspeed
##   Data: train
##         AIC       BIC   logLik
##   -143.4437 -80.47943 88.72187
##
## Correlation Structure: ARMA(2,2)
##  Formula: ~1
##  Parameter estimate(s):
##        Phi1        Phi2      Theta1      Theta2
## -0.55027934 -0.01294438  0.79601696  0.28658952
##
## Coefficients:
##                   Value  Std.Error  t-value p-value
## (Intercept)    7.157318 0.08788936 81.43554  0.0000
## Index         0.003182 0.00017163 18.53854  0.0000
## season2       0.210908 0.04529956  4.65586  0.0000
## season3       0.076987 0.06110761  1.25986  0.2087
## season4       0.096217 0.04339639  2.21716  0.0274
## workingday1    0.231773 0.02371144  9.77473  0.0000
```

```
## minatemp      0.025358 0.00277503  9.13794  0.0000
## sdatemp       0.017481 0.01124074  1.55516  0.1210
## meanhumidity -0.005106 0.00294510 -1.73356  0.0841
## minhumidity  -0.001573 0.00317926 -0.49463  0.6212
## sdhumidity    0.012341 0.00426876  2.89104  0.0041
## maxwindspeed -0.009470 0.00132791 -7.13121  0.0000
##
##  Correlation:
##              (Intr) Index  seasn2 seasn3 seasn4 wrkng1 mintmp sdatmp m
enhmdty
## Index        -0.296

## season2       0.058  0.097

## season3       0.118  0.053  0.636

## season4       0.031 -0.215  0.417  0.423

## workingday1  -0.170  0.018  0.052  0.051  0.026

## minatemp     -0.260 -0.089 -0.587 -0.752 -0.257 -0.083

## sdatemp      -0.339 -0.080 -0.092 -0.018  0.018  0.028  0.059

## meanhumidity  0.003  0.031  0.083  0.110 -0.067  0.049 -0.186 -0.184

## minhumidity  -0.187 -0.003 -0.080 -0.106  0.030 -0.038  0.144  0.244
-0.959
## sdhumidity   -0.238  0.023 -0.050 -0.066  0.030 -0.106  0.104  0.009
-0.815
## maxwindspeed -0.494  0.016  0.004  0.027  0.098  0.065  0.039 -0.061
 0.114
##              minhmdty sdhmdt
## Index
## season2
## season3
## season4
## workingday1
## minatemp
## sdatemp
## meanhumidity
## minhumidity
## sdhumidity    0.843
## maxwindspeed -0.077   -0.060
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -6.4812898 -0.4922786  0.1387196  0.6367704  2.2553054
##
```

```
## Residual standard error: 0.1877561
## Degrees of freedom: 300 total; 288 residual
```

```r
# Casual
# Create a GLS estimate
mc1 = gls(log(Casual)~Index+season+holiday+workingday+
         meanatemp+minatemp+sdatemp+meanhumidity+minhumidity+
         maxwindspeed, data = train,
         correlation = corARMA(p=2, q=2), method = "ML")
summary(mc1)
```

```
## Generalized least squares fit by maximum likelihood
##   Model: log(Casual) ~ Index + season + holiday + workingday + meanat
emp +      minatemp + sdatemp + meanhumidity + minhumidity + maxwindspee
d
##   Data: train
##        AIC      BIC    logLik
##   297.5582 364.2263 -130.7791
##
## Correlation Structure: ARMA(2,2)
##  Formula: ~1
##  Parameter estimate(s):
##       Phi1       Phi2     Theta1     Theta2
##  0.8987330  0.0533060 -0.5525916 -0.1776733
##
## Coefficients:
##                  Value  Std.Error    t-value p-value
## (Intercept)   5.697475 0.28870717  19.734441  0.0000
## Index         0.003004 0.00123236   2.438010  0.0154
## season2       0.321142 0.15564149   2.063347  0.0400
## season3       0.344562 0.21277395   1.619379  0.1065
## season4       0.019491 0.19350566   0.100728  0.9198
## holiday1     -0.133201 0.13072120  -1.018969  0.3091
## workingday1  -0.948639 0.04841342 -19.594549  0.0000
## meanatemp     0.021277 0.02622808   0.811232  0.4179
## minatemp      0.042092 0.02523961   1.667683  0.0965
## sdatemp       0.090304 0.04130552   2.186249  0.0296
## meanhumidity -0.001417 0.00350495  -0.404219  0.6864
## minhumidity  -0.011812 0.00354107  -3.335651  0.0010
## maxwindspeed -0.015978 0.00274844  -5.813511  0.0000
##
##  Correlation:
##              (Intr) Index  seasn2 seasn3 seasn4 holdy1 wrkng1 mentmp m
intmp
## Index        -0.597

## season2      -0.142 -0.080

## season3      -0.116 -0.055  0.454
```

```
## season4      -0.095 -0.145  0.283  0.477

## holiday1     -0.093 -0.004  0.022 -0.011  0.022

## workingday1  -0.108  0.009  0.006 -0.015 -0.003  0.223

## meanatemp    -0.113  0.002 -0.026 -0.077 -0.016 -0.009  0.087

## minatemp      0.036 -0.017  0.000  0.006  0.003  0.014 -0.111 -0.962

## sdatemp      -0.027 -0.018  0.017  0.061  0.031  0.040 -0.056 -0.829
0.806
## meanhumidity -0.211  0.031  0.010  0.043 -0.053  0.046 -0.048 -0.005
-0.028
## minhumidity   0.022 -0.012  0.002 -0.016  0.045  0.003  0.076 -0.024
 0.031
## maxwindspeed -0.293  0.003 -0.017  0.010  0.020  0.003  0.032 -0.114
 0.117
##              sdatmp menhmdty minhmdty
## Index
## season2
## season3
## season4
## holiday1
## workingday1
## meanatemp
## minatemp
## sdatemp
## meanhumidity -0.181
## minhumidity   0.275 -0.867
## maxwindspeed  0.063  0.134   -0.067
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -3.7419288 -0.6323531  0.1068052  0.6855016  2.4586472
##
## Residual standard error: 0.4643663
## Degrees of freedom: 300 total; 287 residual
```

iv.Predict the demand for registered and causal customers for the testing set. Aggregate the individual demands to create the total demand forecast. Report the RMSE.

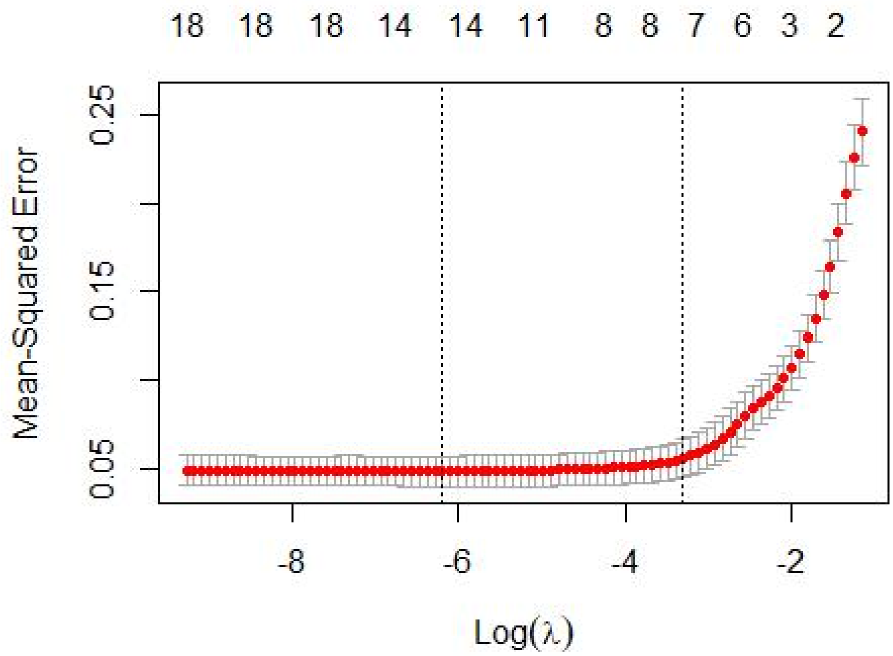
```
# Registered
pr = predict(mr1, newx=test)
sqrt(mean(log(test$Registered) - pr)^2)

## Warning in log(test$Registered) - pr: longer object length is not a m
ultiple of
## shorter object length
```

```
## [1] 0.6366267

# Casual
pc = predict(mc1, newx=test)
sqrt(mean(log(test$Casual) - pc)^2)

## Warning in log(test$Casual) - pc: longer object length is not a multi
ple of
## shorter object length

## [1] 0.8201903

# total
pt = log(exp(pr)+exp(pc))
sqrt(mean(log(test$Total) - pt)^2)

## Warning in log(test$Total) - pt: longer object length is not a multip
le of
## shorter object length

## [1] 0.6637521
```

Part B. Aggregate Demand Forecast. This part is essentially a repeat of the class code. (25 Points)

i. Follow A2 to create a LASSO model from the training set for total demand. Predict the testing set and report the RMSE.

```
# Total
m3 = lm(log(Total)~Index+season+holiday+workingday, data = d)
x3 = model.matrix(m3)
x3 = cbind(x3, as.matrix(d[,c(8:19)]))
y3 = log(d$Total)
trainx3 = x3[ind,]
trainy3 = y3[ind]
testx3 = x3[-ind,]
testy3 = y3[-ind]
#Cross validation of penalty parameter.
l3 = cv.glmnet(trainx3, trainy3)
plot(l3)
```




```
print(l3$lambda.min)

## [1] 0.002055015

# create a LASSO model from the training set for total demand
l3f = glmnet(trainx3, trainy3, lambda = l3$lambda.min)
l3f$beta

## 19 x 1 sparse Matrix of class "dgCMatrix"
##                         s0
## (Intercept)    .
## Index          0.003057413
## season2        0.229607317
## season3        0.062876900
## season4        0.059784451
## holiday1      -0.022159756
## workingday1    .
## meanatemp      .
## maxatemp       .
## minatemp       0.033983612
## sdatemp        0.046926705
## meanhumidity  -0.006023936
## maxhumidity    0.001792347
## minhumidity   -0.002741895
## sdhumidity     0.006602418
## meanwindspeed -0.003679423
## maxwindspeed  -0.009402411
```

```
## minwindspeed   0.002268196
## sdwindspeed    .
```

```
#Predict the test set.
p3 = predict(l3f, newx = testx3)
#RMSE
sqrt(mean(log(test$Total) - p3)^2)
```

```
## [1] 0.2826229
```

ii. Follow A3 to create ARMA(2,2) time-series model for total demand using the training set. Predict the testing set and report the RMSE.

```
# Create a GLS estimate
mt1 = gls(log(Total)~Index+season+holiday+
        maxatemp+minatemp+sdatemp+meanhumidity+maxhumidity+minhumidity+
sdhumidity+
        meanwindspeed+maxwindspeed+minwindspeed, data = train,
        correlation = corARMA(p=2, q=2), method = "ML")
summary(mt1)
```

```
## Generalized least squares fit by maximum likelihood
##   Model: log(Total) ~ Index + season + holiday + maxatemp + minatemp
+      sdatemp + meanhumidity + maxhumidity + minhumidity + sdhumidity +
     meanwindspeed + maxwindspeed + minwindspeed
##   Data: train
##         AIC       BIC   logLik
##   -92.03636 -14.25693 67.01818
##
## Correlation Structure: ARMA(2,2)
##  Formula: ~1
##  Parameter estimate(s):
##       Phi1       Phi2     Theta1     Theta2
## -0.8390776 -0.3148061  1.0859631  0.6081537
##
## Coefficients:
##                    Value  Std.Error  t-value p-value
## (Intercept)    7.431828 0.10249595 72.50850  0.0000
## Index          0.003062 0.00017533 17.46501  0.0000
## season2        0.268268 0.04732073  5.66913  0.0000
## season3        0.115232 0.06430717  1.79190  0.0742
## season4        0.081196 0.04476497  1.81383  0.0708
## holiday1      -0.004094 0.06760613 -0.06056  0.9518
## maxatemp      -0.014909 0.01057050 -1.41047  0.1595
## minatemp       0.046684 0.01050995  4.44190  0.0000
## sdatemp        0.080012 0.02922073  2.73818  0.0066
## meanhumidity  -0.008533 0.00340613 -2.50507  0.0128
## maxhumidity    0.003355 0.00285040  1.17709  0.2401
## minhumidity   -0.002198 0.00378416 -0.58080  0.5618
## sdhumidity     0.006888 0.00734915  0.93725  0.3494
## meanwindspeed -0.010418 0.00484977 -2.14806  0.0326
```

```
## maxwindspeed  -0.006738 0.00245488 -2.74480  0.0064
## minwindspeed   0.008383 0.00363347  2.30728  0.0218
##
##  Correlation:
##              (Intr) Index  seasn2 seasn3 seasn4 holdy1 maxtmp mintmp
sdatmp
## Index        -0.286

## season2       0.115  0.092

## season3       0.166  0.046  0.651

## season4       0.071 -0.213  0.424  0.432

## holiday1     -0.082  0.005 -0.023 -0.057 -0.059

## maxatemp     -0.120  0.045 -0.042 -0.091  0.018 -0.096

## minatemp      0.033 -0.066 -0.129 -0.126 -0.095  0.108 -0.960

## sdatemp      -0.040 -0.073 -0.010  0.065 -0.017  0.095 -0.908  0.881

## meanhumidity   0.103  0.015  0.079  0.114 -0.038  0.108 -0.088  0.036
 -0.010
## maxhumidity   -0.338  0.037 -0.058 -0.061 -0.094  0.002 -0.072  0.096
  0.115
## minhumidity   -0.020 -0.015 -0.035 -0.065  0.070 -0.083  0.136 -0.112
 -0.048
## sdhumidity     0.123 -0.014  0.032  0.020  0.096 -0.045  0.089 -0.094
-0.116
## meanwindspeed -0.173  0.043 -0.054  0.001  0.015  0.093 -0.023  0.030
 -0.006
## maxwindspeed  -0.133 -0.024  0.026  0.001  0.042 -0.078 -0.059  0.063
  0.073
## minwindspeed   0.161 -0.044  0.099  0.063 -0.006 -0.014 -0.062  0.042
  0.025
##              menhmdty mxhmdt minhmdty sdhmdt menwndspd mxwnds
## Index
## season2
## season3
## season4
## holiday1
## maxatemp
## minatemp
## sdatemp
## meanhumidity
## maxhumidity   -0.326
## minhumidity   -0.701   -0.401
## sdhumidity    -0.227   -0.780  0.798
```

```
## meanwindspeed  0.217   -0.034 -0.149   -0.057
## maxwindspeed  -0.130    0.061  0.070   -0.008 -0.783
## minwindspeed  -0.070   -0.038  0.083    0.115 -0.600     0.306
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -6.4601438 -0.4471409  0.1479855  0.6039427  1.9436373
##
## Residual standard error: 0.2030871
## Degrees of freedom: 300 total; 284 residual
```

```
pt1 = predict(mt1, newx=test)
sqrt(mean(log(test$Total) - pt1)^2)
```

```
## Warning in log(test$Total) - pt1: longer object length is not a multi
ple of
## shorter object length

## [1] 0.6479056
```

iii.Compare the RMSE for the total demand from part B with those from part A for both LASSO and ARMA(2,2) models. Comment on the observations. Which forecasting (aggregate or disaggregated) seems to be more precise? What could be the reasons for what you observe?

```
sqrt(mean(log(test$Total) - pt)^2)
```

```
## Warning in log(test$Total) - pt: longer object length is not a multip
le of
## shorter object length

## [1] 0.6637521
```

```
sqrt(mean(log(test$Total) - pt1)^2)
```

```
## Warning in log(test$Total) - pt1: longer object length is not a multi
ple of
## shorter object length

## [1] 0.6479056
```

Observations:

When comparing the 2 RMSE values for total demand from part A and B, we can observe that part B has the lower RMSE value of approximately 0.648. This indicates that aggregated forecasting using ARMA(2,2) model seem to yield better forecasting for total demand.

As aggregated forecasting focuses on forecasting demand for the all customers (not considering individual segments), it is likely that ARMA(2,2) model yielded better results because the model focuses on the overall trend and not individual detail.

Additionally, the aggregated model is more stable over time, so it is more accurate and reliable for forecasting total demand.