Project1_diagnostic

2023-02-07

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
setwd("C:/Users/qingc/Dropbox/STAT 27850/Project 1")
bikeshare_2010 <- read.csv("2010-capitalbikeshare-tripdata.csv")
bikeshare_2011 <- read.csv("2011-capitalbikeshare-tripdata.csv")
bikeshare <- rbind(bikeshare_2010,bikeshare_2011)
head(bikeshare)
```

```
##   Duration          Start.date            End.date Start.station.number
## 1     1012 2010-09-20 11:27:04 2010-09-20 11:43:56                31208
## 2       61 2010-09-20 11:41:22 2010-09-20 11:42:23                31209
## 3     2690 2010-09-20 12:05:37 2010-09-20 12:50:27                31600
## 4     1406 2010-09-20 12:06:05 2010-09-20 12:29:32                31600
## 5     1413 2010-09-20 12:10:43 2010-09-20 12:34:17                31100
## 6      982 2010-09-20 12:14:27 2010-09-20 12:30:50                31109
##                   Start.station End.station.number
## 1      M St & New Jersey Ave SE              31108
## 2                1st & N St  SE              31209
## 3                 5th & K St NW              31100
## 4                 5th & K St NW              31602
## 5 19th St & Pennsylvania Ave NW              31201
## 6                 7th & T St NW              31200
##                          End.station Bike.number Member.type
## 1                      4th & M St SW      W00742      Member
## 2                     1st & N St  SE      W00032      Member
## 3        19th St & Pennsylvania Ave NW      W00993      Member
## 4              Park Rd & Holmead Pl NW      W00344      Member
## 5                     15th & P St NW      W00883      Member
## 6 Massachusetts Ave & Dupont Circle NW      W00850      Member
```

```
bikeshare$routes <- with(bikeshare,paste0(Start.station.number,"-",End.station.number))

bikeshare$Start.date <- strptime(bikeshare$Start.date, format = "%Y-%m-%d %H:%M:%S")
bikeshare$End.date <- strptime(bikeshare$End.date, format = "%Y-%m-%d %H:%M:%S")
bikeshare$Start.date <- as.POSIXct(bikeshare$Start.date, format = "%Y-%m-%d %H:%M:%S")
bikeshare$End.date <- as.POSIXct(bikeshare$End.date, format = "%Y-%m-%d %H:%M:%S")


bikeshare$weekday <- weekdays(as.Date(bikeshare$Start.date))
bikeshare$weekday <- ifelse(bikeshare$weekday %in% c("Firday","Saturday","Sunday"),0,1)



drops <-c("End.date","Start.station.number","Start.station","End.station.number","End.station")

df <- bikeshare[ , !(names(bikeshare) %in% drops)]

df$Start.date <- as.numeric(df$Start.date- df$Start.date[1])/(3600*24)


head(df)
```

```
##   Duration  Start.date Bike.number Member.type      routes weekday
## 1     1012 0.000000000      W00742      Member 31208-31108       1
## 2       61 0.009930556      W00032      Member 31209-31209       1
## 3     2690 0.026770833      W00993      Member 31600-31100       1
## 4     1406 0.027094907      W00344      Member 31600-31602       1
## 5     1413 0.030312500      W00883      Member 31100-31201       1
## 6      982 0.032905093      W00850      Member 31109-31200       1
```

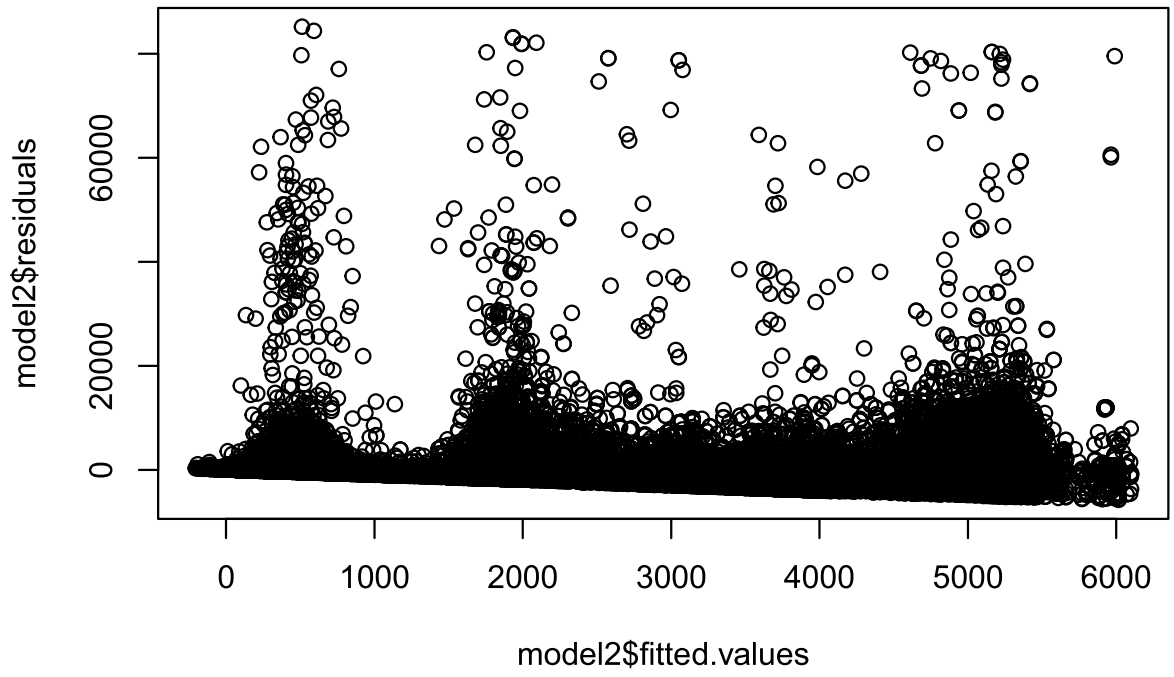
```
count_routes <- df %>%group_by(routes)%>%summarise(n=n())
count_routes[order(count_routes$n,decreasing = TRUE),]
```

```
## # A tibble: 12,179 x 2
##    routes          n
##    <chr>       <int>
##  1 31104-31106  4833
##  2 31106-31104  4831
##  3 31613-31619  4820
##  4 31619-31613  3806
##  5 31200-31201  3674
##  6 31217-31217  3373
##  7 31229-31200  3270
##  8 31201-31200  3229
##  9 31101-31200  2779
## 10 31623-31611  2753
## # ... with 12,169 more rows
```

```
count_routes <- df %>%
  group_by(routes) %>%
  summarise(count = n()) %>%
  filter(count >= 1000)
```

```
df_reduced <- df[df$routes %in% count_routes$routes,]
rm(df)
rm(bikeshare)
rm(bikeshare_2010)
rm(bikeshare_2011)
model2 <- lm(Duration~Start.date*routes+weekday+Member.type,data = df_reduced)
```

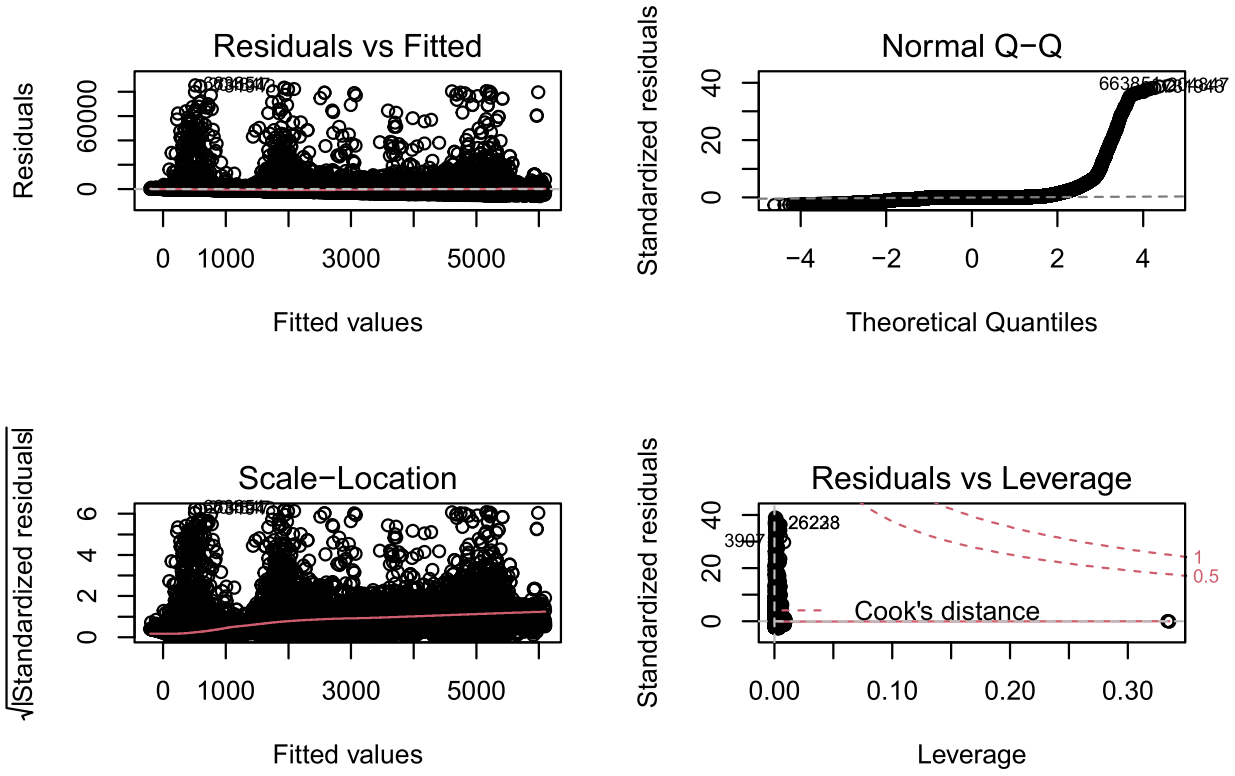
```
plot(model2$fitted.values,model2$residuals)
```




```
# Outliers
# X = model.matrix(model2)
# H = X%*%solve(t(X)%*%X,t(X))
# hist(diag(H),breaks=10)
```

Error message; data size is too large.

```
# Heteroscedasticity
# Residuals vs. Fitted
par(mfrow = c(2, 2))
plot(model2)
```



Strong evidence of non-constant variance in the Residuals vs. Fitted values plot.

```
# Breusch-Pagan
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
lmtest::bptest(model2)
```

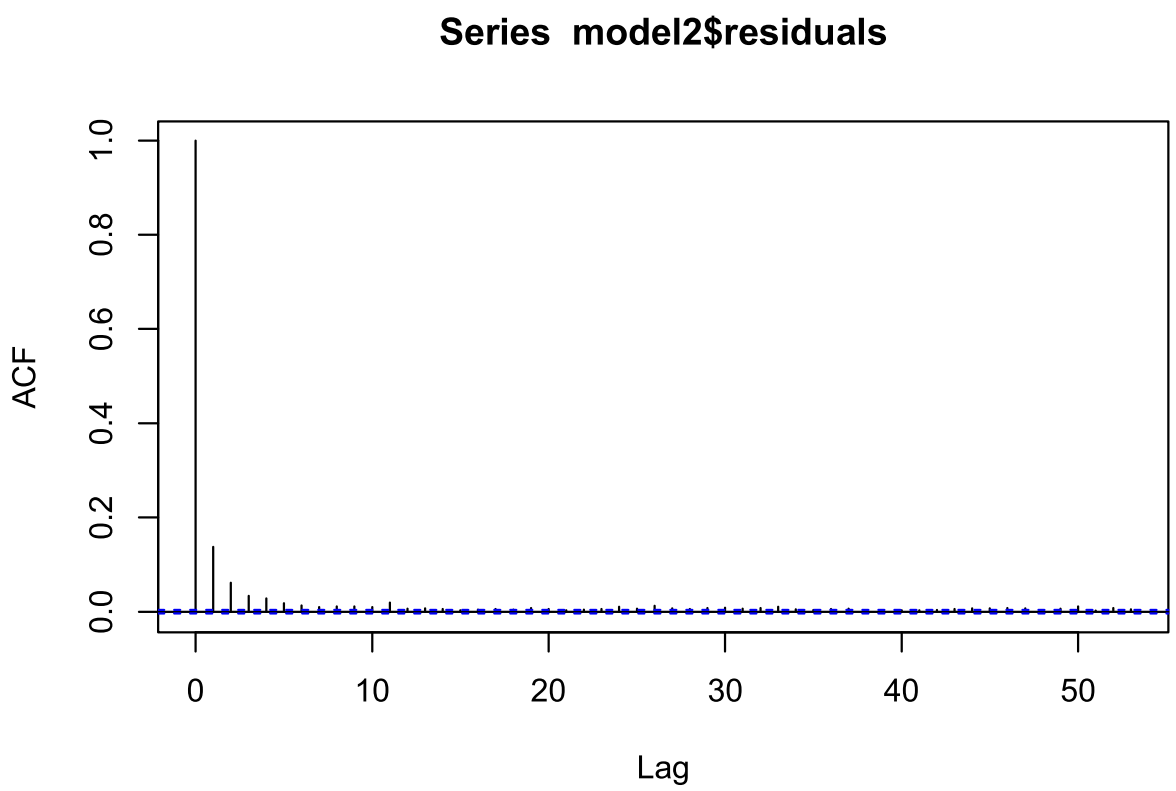
```
##
##  studentized Breusch-Pagan test
##
## data:  model2
## BP = 1957.7, df = 314, p-value < 2.2e-16
```

Extremely small P-value; strong evidence of heteroscedasticity.

```
# White Test
# library(skedastic)
# skedastic:::white_lm(model2)
```

Error message; data size is too large.

```
# Autocorrelation
# ACF
acf(model2$residuals, type = "correlation")
```



Multiple bars above the blue dotted line; evidence of heteroscedasticity.

```
# Durbin-Watson
library(lmtest)
lmtest::dwtest(model2)
```

```
##
##  Durbin-Watson test
##
## data:  model2
## DW = 1.7254, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

Extremely small P-value; strong evidence of autocorrelation.

```r
# Breusch-Godfrey
lmtest::bgtest(model2, order = 3)
```

```
##
##  Breusch-Godfrey test for serial correlation of order up to 3
##
## data:  model2
## LM test = 5257.9, df = 3, p-value < 2.2e-16
```

Extremely small P-value; strong evidence of autocorrelation.