Final Project Exploration

Luis Calleja

December 16, 2018

```r
library(dplyr)
library(tidyverse)
library(stringr)
library(data.table)
library(ggplot2)
library(lubridate)
library(zoo)
library(forecast)
library(nlme)
library(skimr)
```

MTA data

```r
subway <- read.csv('/home/lechuza/Documents/CUNY/data_621/finalProject/MTA_Performance_Agencies.csv', st

str(subway)
```

```
## 'data.frame':    15258 obs. of  17 variables:
##  $ INDICATOR_SEQ : int  74039 74039 74039 74039 74039 74039 74039 74039 74039 74039 ...
##  $ PARENT_SEQ    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AGENCY_NAME   : chr  "Bridges and Tunnels" "Bridges and Tunnels" "Bridges and Tunnels" "Bridges an
##  $ INDICATOR_NAME: chr  "Collisions with Injury Rate" "Collisions with Injury Rate" "Collisions with
##  $ DESCRIPTION   : chr  "All customer collisions with injuries on B&T property.  The rate is collisi
##  $ CATEGORY      : chr  "Safety Indicators" "Safety Indicators" "Safety Indicators" "Safety Indicato
##  $ FREQUENCY     : chr  "M" "M" "M" "M" ...
##  $ DESIRED_CHANGE: chr  "D" "D" "D" "D" ...
##  $ INDICATOR_UNIT: chr  "-" "-" "-" "-" ...
##  $ DECIMAL_PLACES: int  2 2 2 2 2 2 2 2 2 2 ...
##  $ PERIOD_YEAR   : int  2008 2008 2008 2008 2008 2008 2008 2008 2008 2008 ...
##  $ PERIOD_MONTH  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ YTD_TARGET    : num  0.75 0.89 0.9 0.97 0.99 1.03 1.07 1.09 1.09 1.09 ...
##  $ YTD_ACTUAL    : num  0.54 0.75 0.77 0.79 0.94 0.94 0.98 0.97 0.97 0.98 ...
##  $ MONTHLY_TARGET: num  0.75 1.02 0.92 1.2 1.08 1.18 1.35 1.2 1.15 1.02 ...
##  $ MONTHLY_ACTUAL: num  0.54 0.98 0.8 0.84 1.49 0.97 1.19 0.91 1.02 0.99 ...
##  $ YYYY_MM       : chr  "2008-01" "2008-02" "2008-03" "2008-04" ...
```

Ride hailing data

```r
#For Hire Vehicle data
rh <- read.csv('/home/lechuza/Documents/CUNY/data_621/finalProject/rideHailing/FHV_Base_Aggregate_Report

str(rh)
```

```
## 'data.frame':    21078 obs. of  9 variables:
##  $ Base.License.Number          : chr  "B01326" "B01957" "B03018" "B02539" ...
##  $ Base.Name                    : chr  "DEBORAH C/L SVC INC" "ANIMO, INC" "AUREUS LLC" "HAWK CAR AND
##  $ DBA                          : chr  "" "CHRIS LIMOUSINE U.S.A." "" "JRIDE" ...
```

```
##  $ Year                        : int  2016 2015 2017 2018 2017 2017 2017 2016 2015 2018 ...
##  $ Month                       : int  8 6 8 1 5 5 8 3 10 1 ...
##  $ Month.Name                  : chr  "August" "June" "August" "January" ...
##  $ Total.Dispatched.Trips      : int  154 140 179 1015 669 27204 5570 1323 4298 3745 ...
##  $ Total.Dispatched.Shared.Trips: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ Unique.Dispatched.Vehicles   : int  2 5 12 6 60 134 19 6 25 52 ...
```

Iden's code to join the target variables.

```
rh %>%
  filter(grepl('uber|UBER|lyft|LYFT', rh$Base.Name)) -> ma

dt <- data.table(ma)

test <- dt[,sum(Total.Dispatched.Trips),by = c('Year','Month')]

names(test)[3] <- 'total.dispatched.trips'

#prep the MTA data:
subway <- subway[, c(4, 11, 12, 16)]

mdf <- subway %>%
  filter(INDICATOR_NAME == "Mean Distance Between Failures - Subways") %>%
  spread(INDICATOR_NAME, MONTHLY_ACTUAL)
tr <- subway %>%
  filter(INDICATOR_NAME == "Total Ridership - Subways") %>%
  spread(INDICATOR_NAME, MONTHLY_ACTUAL)
cir <- subway %>%
  filter(INDICATOR_NAME == "Customer Injury Rate - Subways") %>%
  spread(INDICATOR_NAME, MONTHLY_ACTUAL)
elev <- subway %>%
  filter(INDICATOR_NAME == "Elevator Availability - Subways") %>%
  spread(INDICATOR_NAME, MONTHLY_ACTUAL)
esc <- subway %>%
  filter(INDICATOR_NAME == "Escalator Availability - Subways") %>%
  spread(INDICATOR_NAME, MONTHLY_ACTUAL)
otp <- subway %>%
  filter(INDICATOR_NAME == "On-Time Performance (Terminal)") %>%
  spread(INDICATOR_NAME, MONTHLY_ACTUAL)
wait <- subway %>%
  filter(INDICATOR_NAME == "Subway Wait Assessment ") %>%
  spread(INDICATOR_NAME, MONTHLY_ACTUAL)

final.subway <- left_join(mdf, tr, by = c("PERIOD_YEAR", "PERIOD_MONTH"))
final.subway <- left_join(final.subway, cir, by = c("PERIOD_YEAR", "PERIOD_MONTH"))
final.subway <- left_join(final.subway, elev, by = c("PERIOD_YEAR", "PERIOD_MONTH"))
final.subway <- left_join(final.subway, esc, by = c("PERIOD_YEAR", "PERIOD_MONTH"))
final.subway <- left_join(final.subway, otp, by = c("PERIOD_YEAR", "PERIOD_MONTH"))
final.subway <- left_join(final.subway, wait, by = c("PERIOD_YEAR", "PERIOD_MONTH"))
```

Iden renames the dataframe and finalizes it.

```
final.subway <- final.subway[-c(1:17),]
colnames(final.subway) <- c("YEAR", "MONTH", "FAILURE", "RIDERSHIP", "INJURY", "ELEV", "ESCA", "OTP","W

#merge the FHV to MTA
```

```r
names(final.subway)
```

```
## [1] "YEAR"      "MONTH"     "FAILURE"   "RIDERSHIP" "INJURY"    "ELEV"
## [7] "ESCA"      "OTP"       "WAIT"
```

```r
dt.fs <- data.table(final.subway)
```

Merge MTA dataset to the FHV data.

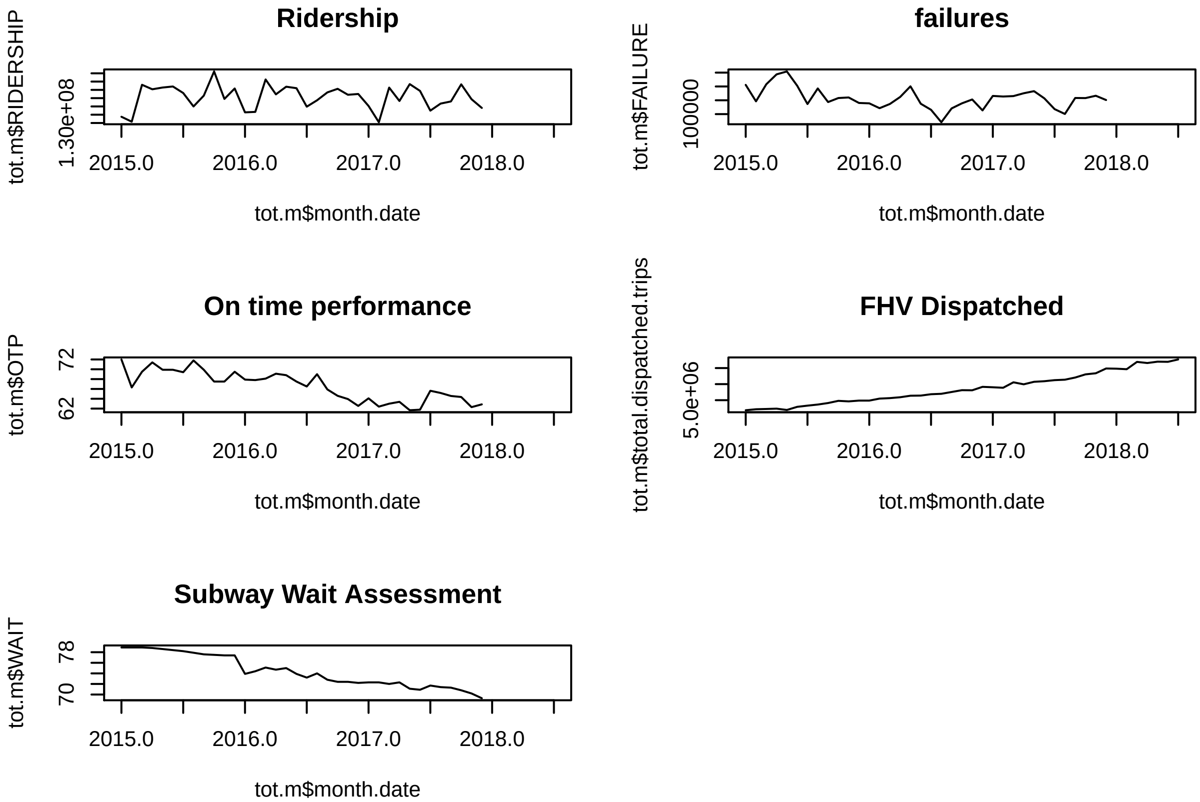
```r
tot <- merge(test,dt.fs, by.x = c('Year','Month'), by.y = c("YEAR","MONTH"), suffixes = c(".fhv",".mta")
#36 observations... is that expected? Yes, the MTA data doesn't include 2018
tot.m <- merge(test,dt.fs, by.x = c('Year','Month'), by.y = c("YEAR","MONTH"), suffixes = c(".fhv",".mta

names(tot.m)
```

```
##  [1] "Year"                   "Month"
##  [3] "total.dispatched.trips" "FAILURE"
##  [5] "RIDERSHIP"              "INJURY"
##  [7] "ELEV"                   "ESCA"
##  [9] "OTP"                    "WAIT"
```

```r
z <- zoo::as.yearmon(paste(tot.m$Year, tot.m$Month, rep('01', length(tot.m$Year)),sep = '-'))

tot.m$month.date <- z
```

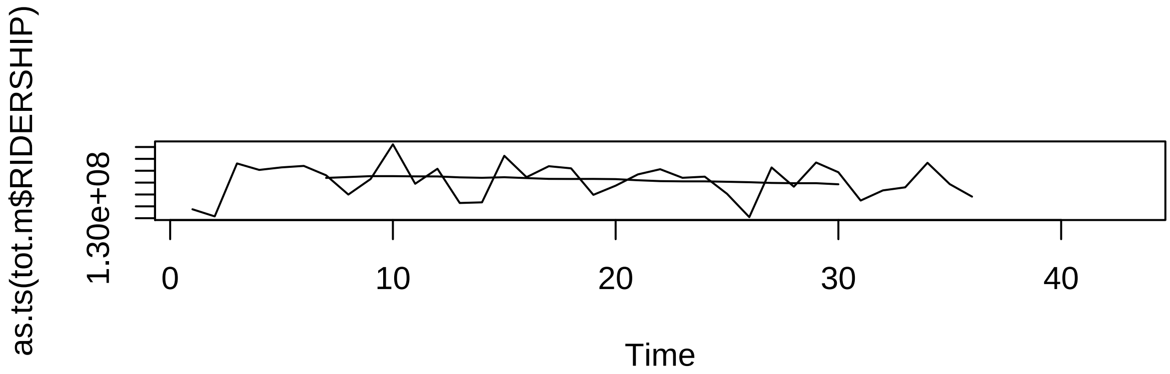
Overlay monthly public ridership, FHV ridership, mta on time performance

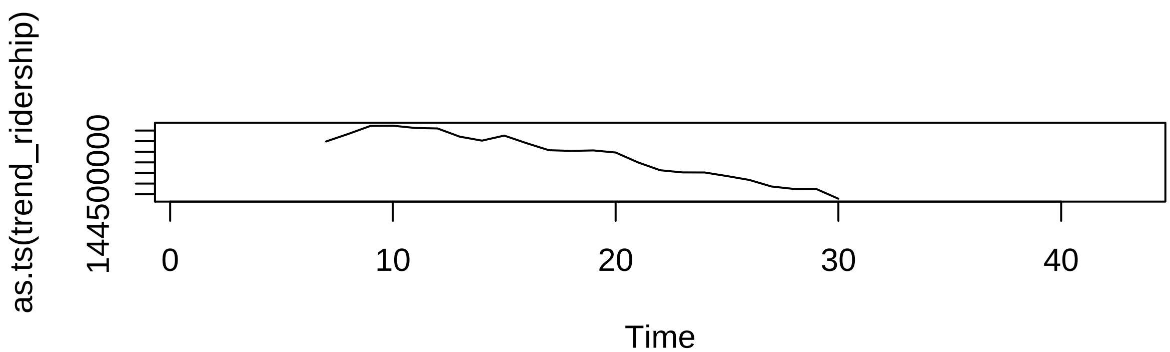
```r
par(mfrow = c(3,2))
plot.zoo(tot.m$month.date, tot.m$RIDERSHIP, type = 'l', main = "Ridership")
plot.zoo(tot.m$month.date, tot.m$FAILURE, type = 'l', main ="failures")
plot.zoo(tot.m$month.date, tot.m$OTP, type = 'l', main = "On time performance")
plot.zoo(tot.m$month.date, tot.m$total.dispatched.trips, type = 'l', main = "FHV Dispatched")
plot.zoo(tot.m$month.date, tot.m$WAIT, type = 'l', main = "Subway Wait Assessment")
```



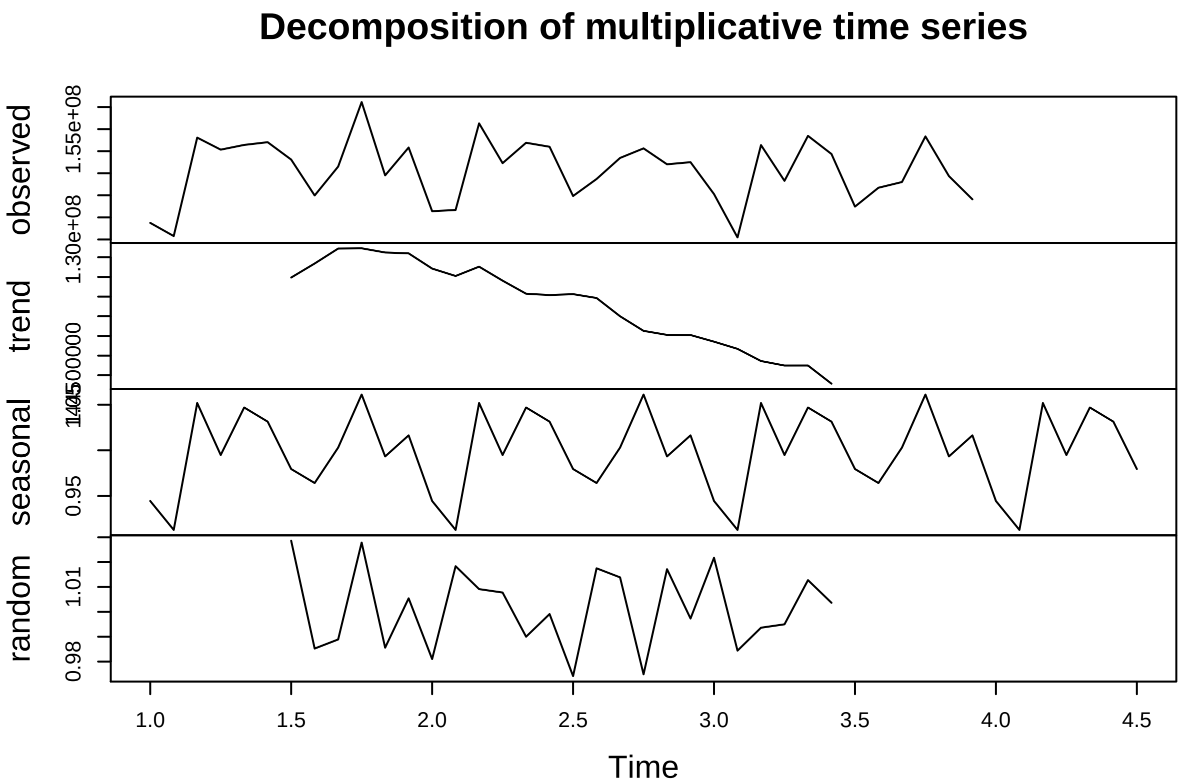
Analyze seasonality and trend for the response and predictor variables as they are all temporal.

```
trend_ridership <- ma(tot.m$RIDERSHIP, order = 12, centre = TRUE)
par(mfrow = c(2,1))
plot(as.ts(tot.m$RIDERSHIP))
lines(trend_ridership)
plot(as.ts(trend_ridership))
```

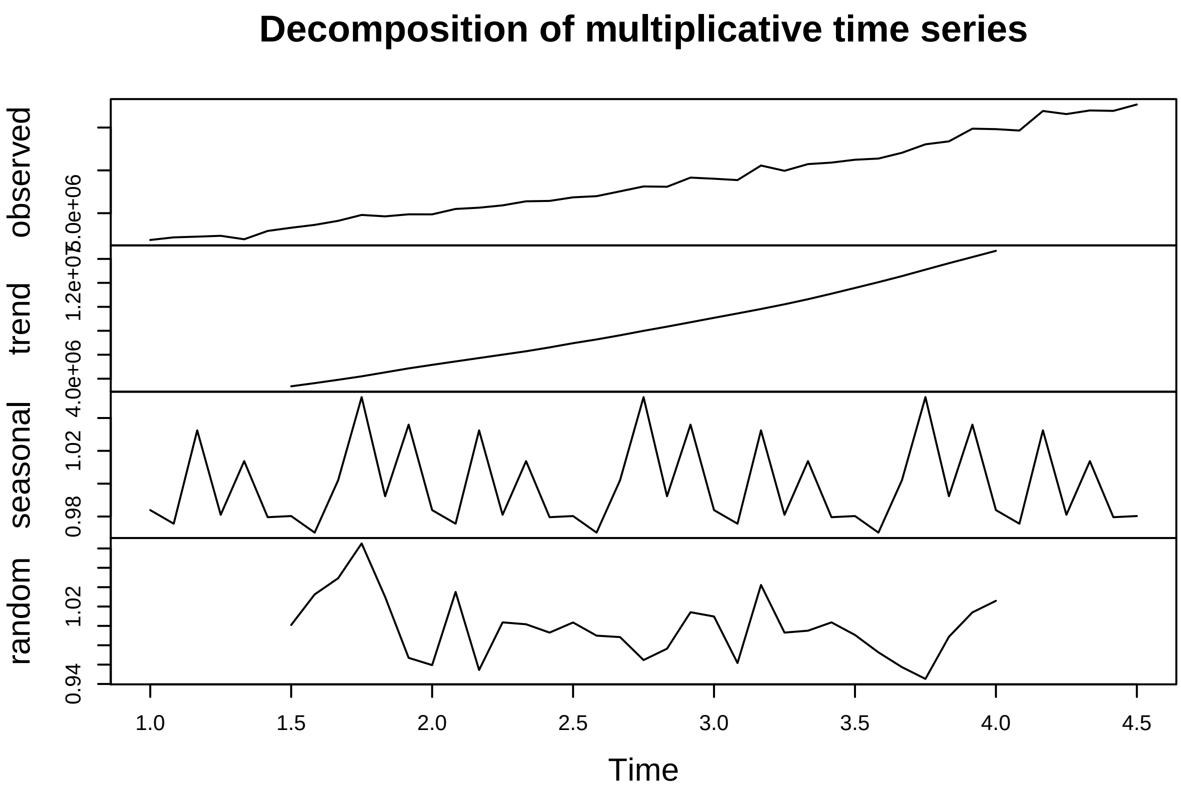




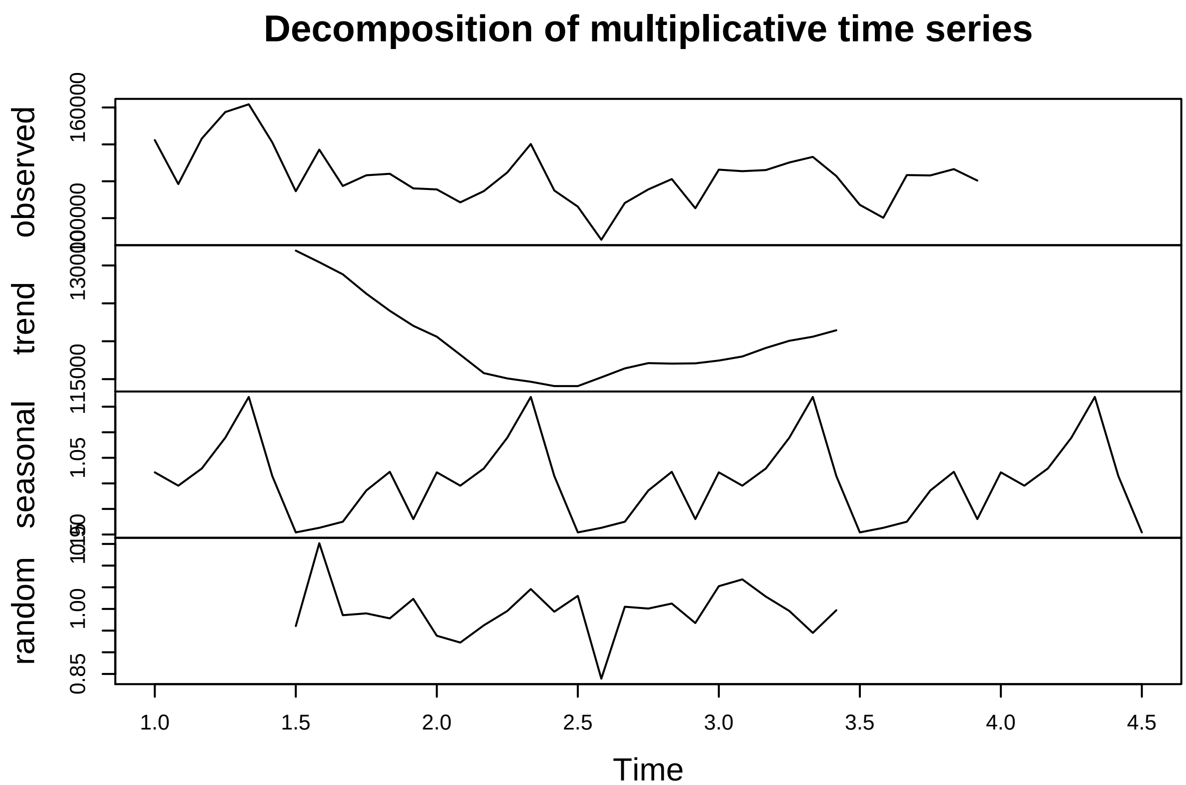

```r
ts_ride <- ts(tot.m$RIDERSHIP, frequency = 12)
ts.fhv <- ts(tot.m$total.dispatched.trips, frequency = 12)
ts.fail <- ts(tot.m$FAILURE, frequency = 12)
ts.otp <- ts(tot.m$OTP, frequency = 12)
ts.wait <- ts(tot.m$WAIT, frequency = 12)

ts.ride.de <- decompose(ts_ride, "multiplicative")
ts.fhv.de <- decompose(ts.fhv, "multiplicative")
ts.fail.de <- decompose(ts.fail, "multiplicative")
ts.otp.de <- decompose(ts.otp, "multiplicative")

#plot ridership deconstructed time series
plot(decompose(ts_ride, "multiplicative"))
```



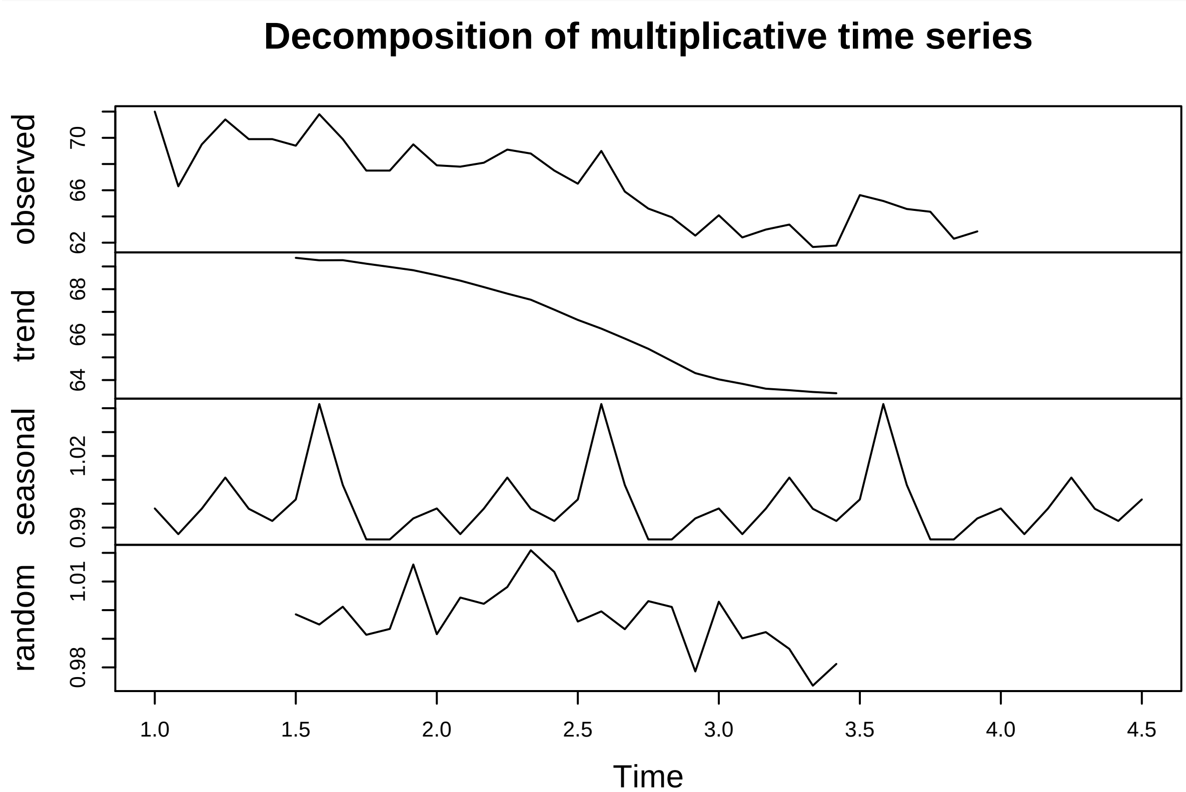

```r
#plot ride hailing dispatched trips deconstructed time series
plot(decompose(ts.fhv, "multiplicative"))
```




```r
#plot failure MTA deconstructed time series
plot(decompose(ts.fail, "multiplicative"))
```




```r
#plot on time performance deconstructed time series
plot(decompose(ts.otp, "multiplicative"))
```

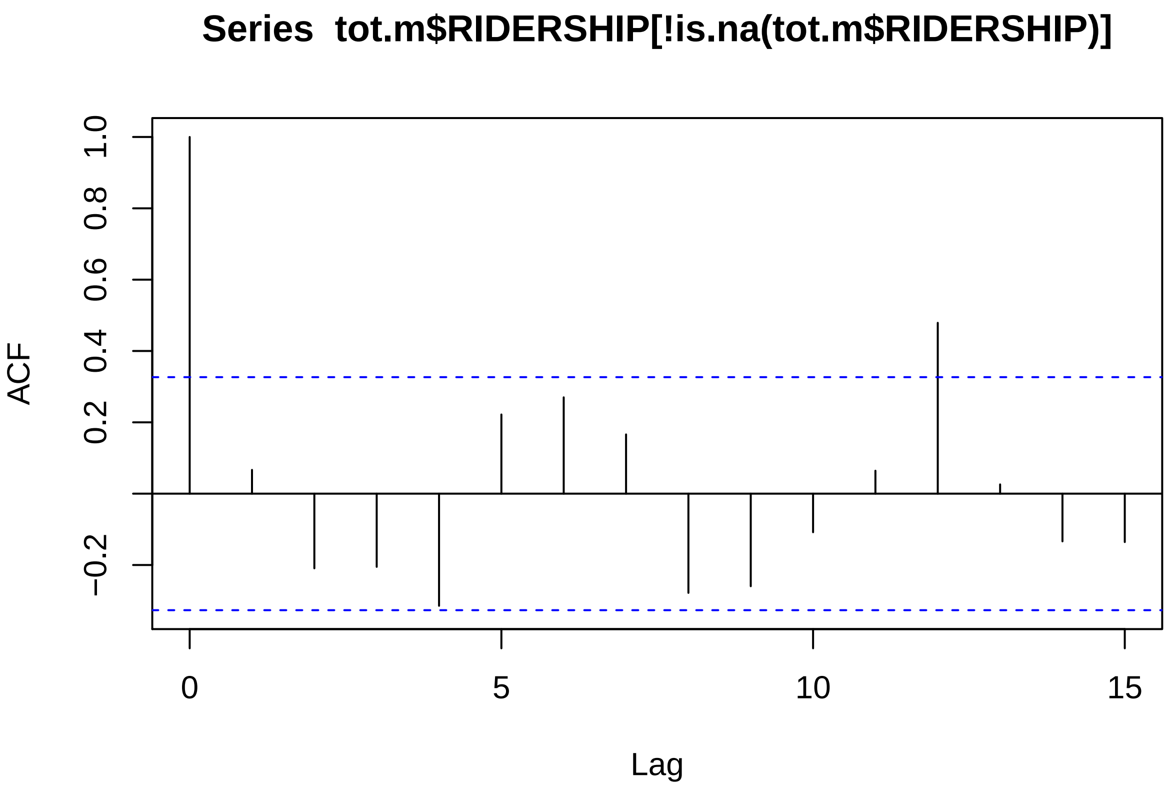


Look at the autocorrelation tendencies from each variable.

```r
head(ts.ride.de$random)
```

```
##   Jan Feb Mar Apr May Jun
## 1  NA  NA  NA  NA  NA  NA
```

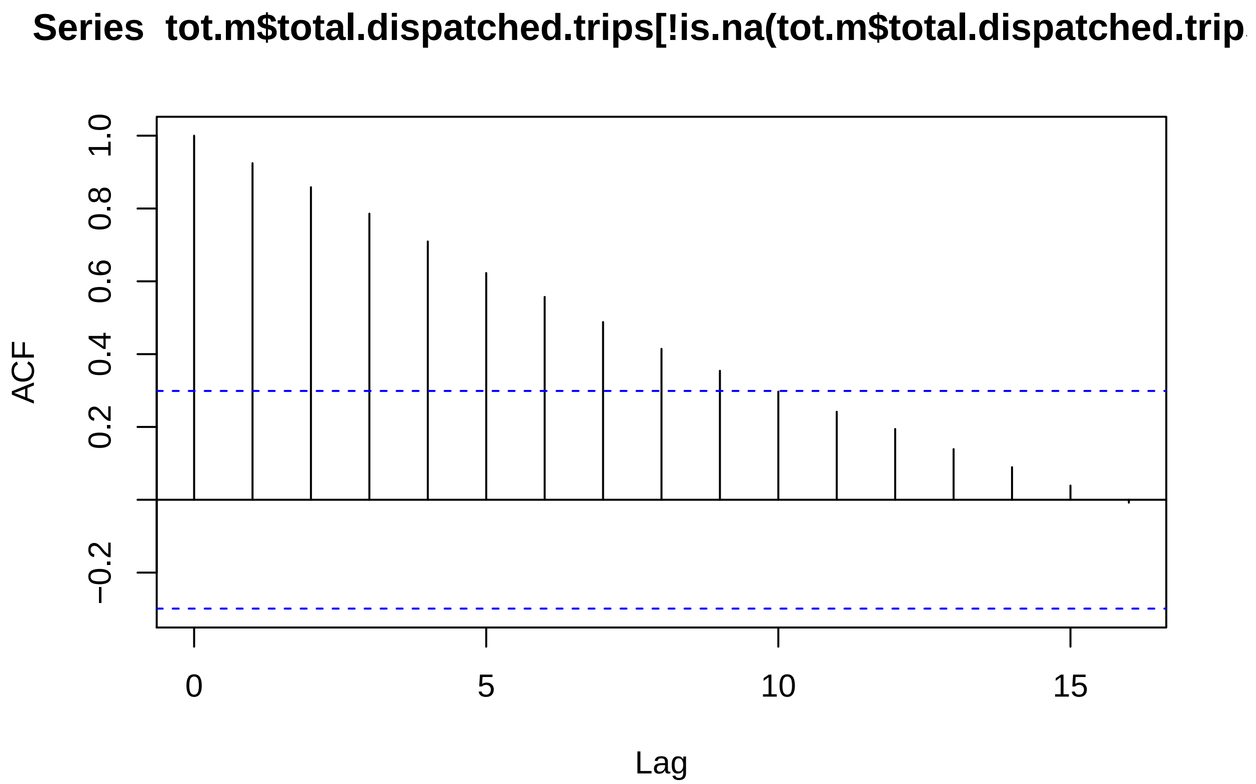
```r
#acf(ts.ride.de$random)
par(mfrow=c(1,1))
acf(tot.m$RIDERSHIP[!is.na(tot.m$RIDERSHIP)])
```



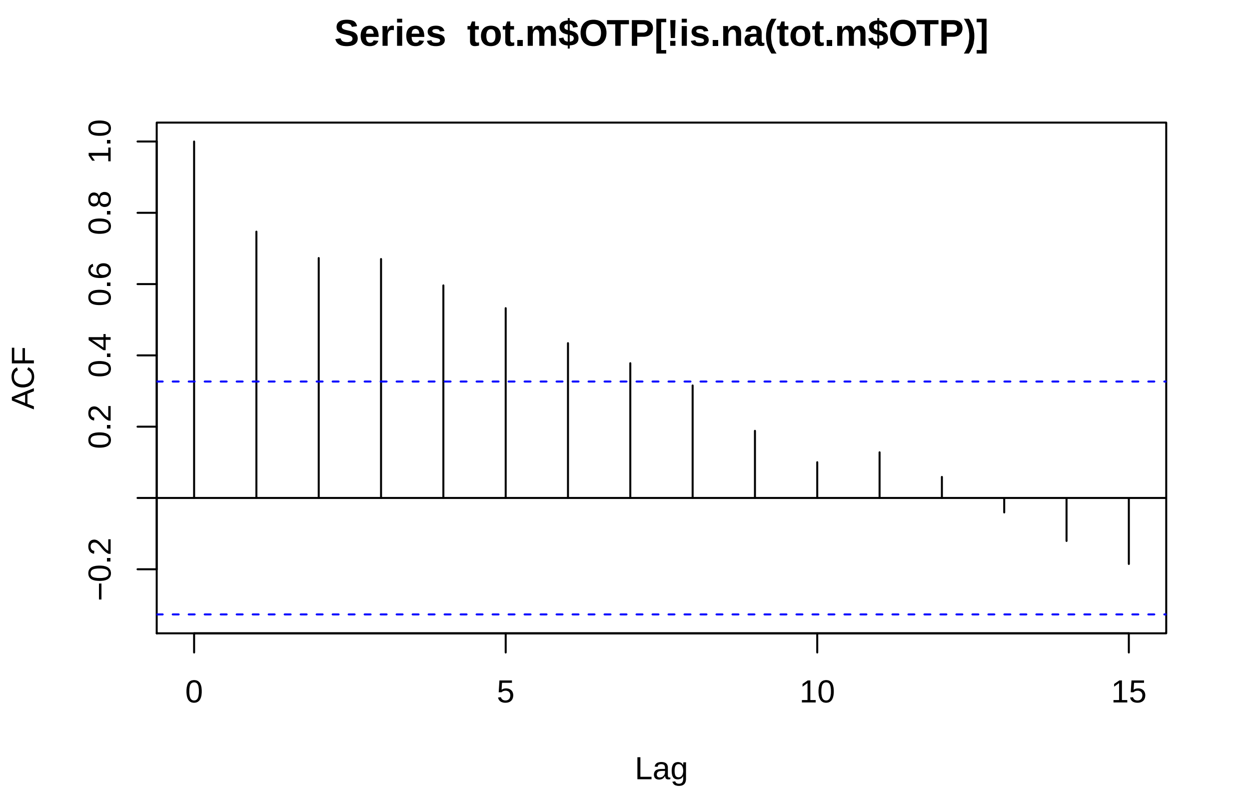
This confirms the seasonality inherent in the ridership time series.

Plot the autocorrelation measure among the predictors.

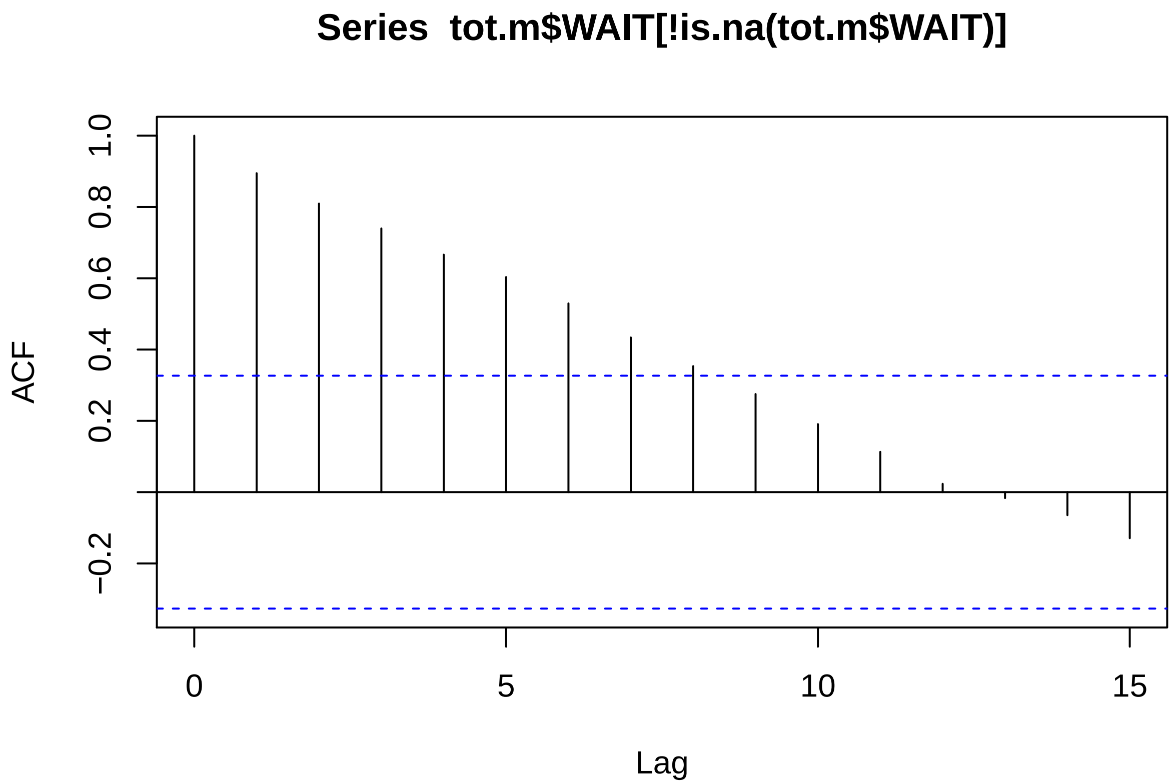
```r
acf(tot.m$total.dispatched.trips[!is.na(tot.m$total.dispatched.trips)])
```




```
acf(tot.m$OTP[!is.na(tot.m$OTP)])
```




```
acf(tot.m$WAIT[!is.na(tot.m$WAIT)])
```



Initially... my desire was to build two models: 1) The first model is built using the entirely decomposed time series with trend and seasonality removed. 2) Second model is built using the trend datasets (with seasonality removed) from each variable.

This is not very feasible as the de-trending and seasonality removal would leave us too few observations (<30). Instead, we'll fit a more naive lm and account for any serial correlation via the GLS approach.

Plot a time series model on lags of first and second order of each of the predictors.

```
tot.m %>%
  dplyr::select(-INJURY) -> temp

temp[complete.cases(temp)] -> tot.m.cc

lagged.set = data.frame(
  ridership = lag(tot.m.cc$RIDERSHIP,1),
fhv = tot.m.cc$total.dispatched.trips,
failure = tot.m.cc$FAILURE,
otp = tot.m.cc$OTP,
wait = tot.m.cc$WAIT,
month = tot.m.cc$Month)

lagged.set <- lagged.set[complete.cases(lagged.set),]
```

Fit a model on the lagged predictors

```
model.ts <- lm(data = lagged.set, ridership ~ .)

summary(model.ts)
```

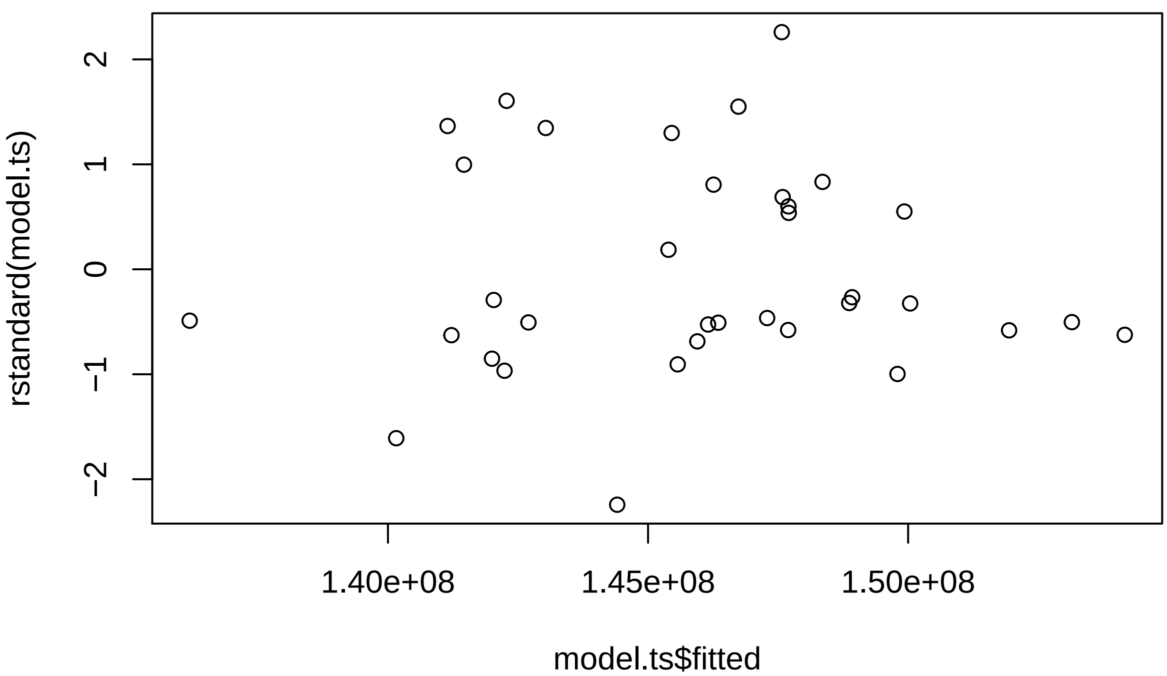
```
##
## Call:
```

```
## lm(formula = ridership ~ ., data = lagged.set)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -13628748  -3499868  -1940525   4448872  13553427
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.692e+08  1.141e+08   3.236  0.00303 **
## fhv         -2.700e+00  1.236e+00  -2.184  0.03724 *
## failure      2.293e+02  9.120e+01   2.514  0.01772 *
## otp          5.011e+05  7.396e+05   0.678  0.50345
## wait        -3.674e+06  1.538e+06  -2.389  0.02364 *
## month        1.179e+06  4.025e+05   2.929  0.00656 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6561000 on 29 degrees of freedom
## Multiple R-squared:  0.2931, Adjusted R-squared:  0.1712
## F-statistic: 2.405 on 5 and 29 DF,  p-value: 0.06098
```

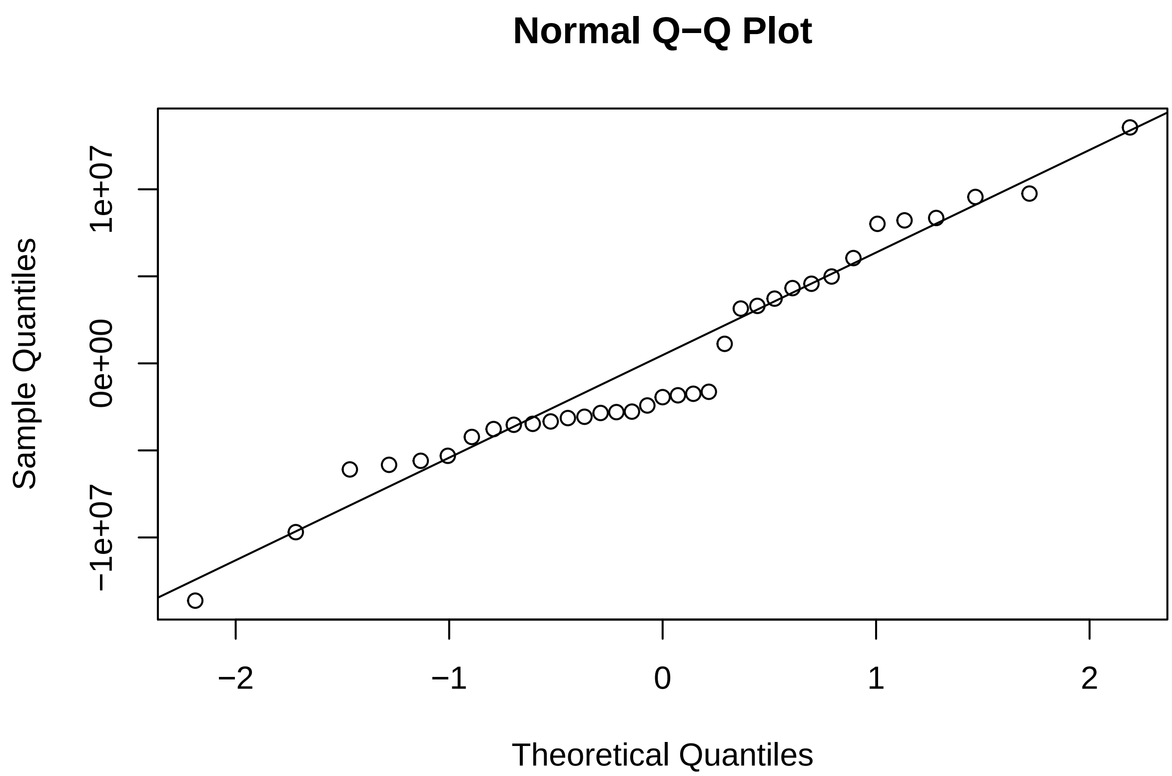
Wow, the model is hot garbage!

Model diagnostics

```r
par(mfrow=c(1,1))
plot(model.ts$fitted, rstandard(model.ts))
```



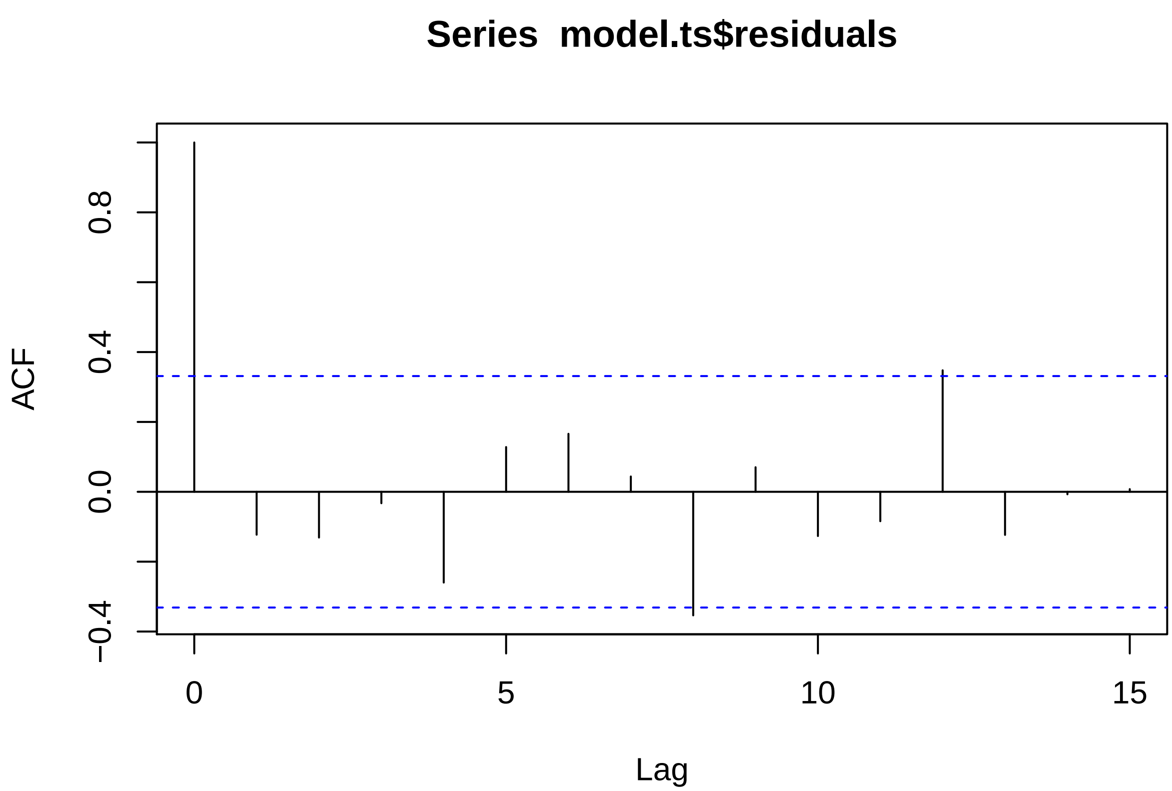

```r
qqnorm(residuals(model.ts))
qqline(residuals(model.ts))
```



The qqplot demonstrates a pattern among the residuals, suggesting the model is not a good fit.

Investigate whether there is any autocorrelation among the error terms.

```
acf(model.ts$residuals)
```



There

is an indication of lag 12 serial correlation of the errors.

Treat the response variable, then re-fit.

```r
model.ts.log <- lm(data = lagged.set, I(log(ridership) ~ .))

summary(model.ts.log)
```

```
##
## Call:
## lm(formula = I(log(ridership) ~ .), data = lagged.set)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.09671 -0.02447 -0.01270  0.03054  0.08932
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.037e+01  7.871e-01  25.880  < 2e-16 ***
## fhv         -1.892e-08  8.531e-09  -2.217  0.03462 *
## failure      1.597e-06  6.293e-07   2.537  0.01681 *
## otp          3.694e-03  5.103e-03   0.724  0.47496
## wait        -2.601e-02  1.061e-02  -2.451  0.02049 *
## month        8.289e-03  2.777e-03   2.985  0.00571 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04527 on 29 degrees of freedom
## Multiple R-squared:  0.3003, Adjusted R-squared:  0.1797
## F-statistic: 2.489 on 5 and 29 DF,  p-value: 0.0541
```

Fit a generalized least squares model

```r
summary(gls(data = lagged.set, ridership ~ .))
```

```
## Generalized least squares fit by REML
##   Model: ridership ~ .
##   Data: lagged.set
##        AIC      BIC    logLik
##   1079.792 1089.363 -532.8958
##
## Coefficients:
##                 Value Std.Error   t-value p-value
## (Intercept) 369178705 114081530  3.236095  0.0030
## fhv                -3         1 -2.183519  0.0372
## failure           229        91  2.514409  0.0177
## otp            501057    739553  0.677513  0.5035
## wait         -3673739   1537963 -2.388704  0.0236
## month         1178919    402476  2.929168  0.0066
##
##  Correlation:
##         (Intr) fhv    failur otp    wait
## fhv     -0.967
## failure  0.325 -0.334
## otp     -0.168  0.155  0.116
## wait    -0.893  0.865 -0.454 -0.284
```

```
## month    0.453 -0.524  0.341 -0.092 -0.428
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -2.0772619 -0.5334417 -0.2957703  0.6780867  2.0657817
##
## Residual standard error: 6560919
## Degrees of freedom: 35 total; 29 residual
```

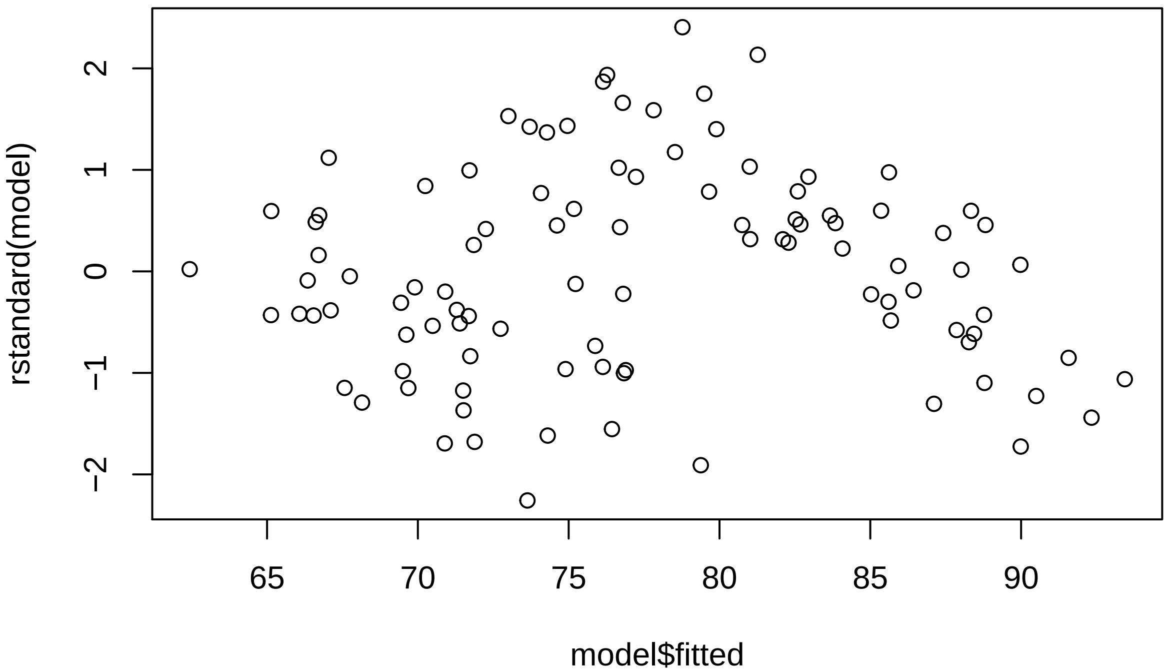
Iden's model

```
fs.cc <- final.subway[complete.cases(final.subway),]
model <- lm(OTP ~ FAILURE + RIDERSHIP + INJURY + ELEV + ESCA, data = fs.cc)

summary(model)
```

```
##
## Call:
## lm(formula = OTP ~ FAILURE + RIDERSHIP + INJURY + ELEV + ESCA,
##     data = fs.cc)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.2324  -3.5214  -0.0817   2.9602  11.8279
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.735e+02  7.285e+01  -2.382   0.0192 *
## FAILURE      1.582e-04  1.890e-05   8.369 5.34e-13 ***
## RIDERSHIP   -3.371e-07  7.612e-08  -4.428 2.56e-05 ***
## INJURY      -1.728e+00  1.603e+00  -1.078   0.2837
## ELEV         3.470e+00  7.784e-01   4.457 2.29e-05 ***
## ESCA        -5.904e-01  4.241e-01  -1.392   0.1671
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.122 on 94 degrees of freedom
## Multiple R-squared:  0.7036, Adjusted R-squared:  0.6878
## F-statistic: 44.62 on 5 and 94 DF,  p-value: < 2.2e-16
```

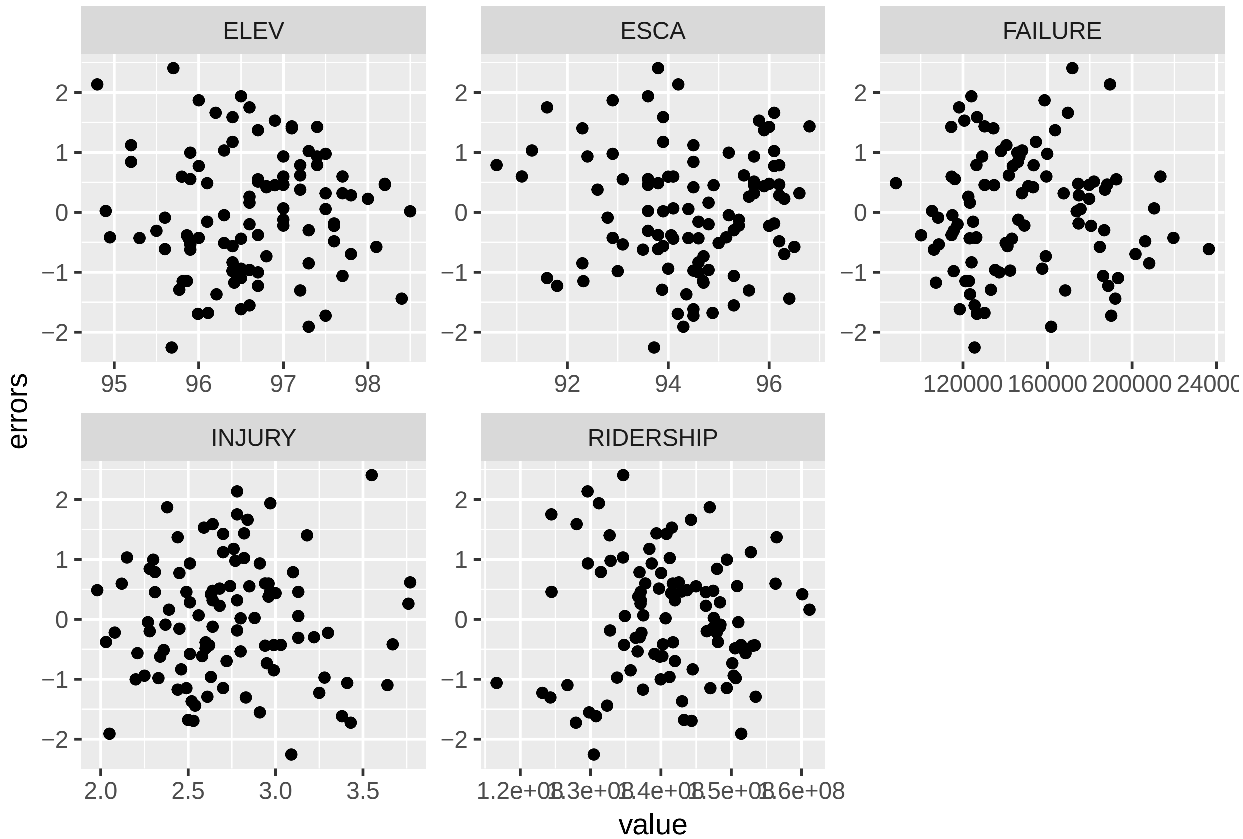
Plot the residuals… looks like there is a pattern of the residuals… almost a negative quadratic

```
par(mfrow= c(1,1))
plot(model$fitted, rstandard(model))
```



Plot each variable against the residuals

```
fs.cc %>%
  dplyr::select(FAILURE, RIDERSHIP, ELEV, ESCA, INJURY) %>%
  gather() -> long.form

long.form$errors <- rep(rstandard(model),5)

ggplot(long.form,aes(x = value, y = errors)) +geom_point() + facet_wrap(~key, scales = "free")
```


ELEV
ESCA
FAILURE
INJURY
RIDERSHIP
errors
value