Flight Data Regression Model

Adding FRED Data to our model

```
library(readxl)
library(data.table)
library(fpp3)
```

```
## -- Attaching packages --------------------------------------- fpp3 0.4.0 --

## v tibble      3.1.5     v tsibble     1.1.0
## v dplyr       1.0.7     v tsibbledata 0.3.0
## v tidyr       1.1.4     v feasts      0.2.2
## v lubridate   1.8.0     v fable       0.3.1
## v ggplot2     3.3.5

## -- Conflicts ------------------------------------------------ fpp3_conflicts --
## x dplyr::between()     masks data.table::between()
## x lubridate::date()    masks base::date()
## x dplyr::filter()      masks stats::filter()
## x dplyr::first()       masks data.table::first()
## x lubridate::hour()    masks data.table::hour()
## x tsibble::intersect() masks base::intersect()
## x tsibble::interval()  masks lubridate::interval()
## x lubridate::isoweek() masks data.table::isoweek()
## x tsibble::key()       masks data.table::key()
## x dplyr::lag()         masks stats::lag()
## x dplyr::last()        masks data.table::last()
## x lubridate::mday()    masks data.table::mday()
## x lubridate::minute()  masks data.table::minute()
## x lubridate::month()   masks data.table::month()
## x lubridate::quarter() masks data.table::quarter()
## x lubridate::second()  masks data.table::second()
## x tsibble::setdiff()   masks base::setdiff()
## x tsibble::union()     masks base::union()
## x lubridate::wday()    masks data.table::wday()
## x lubridate::week()    masks data.table::week()
## x lubridate::yday()    masks data.table::yday()
## x lubridate::year()    masks data.table::year()
```

```
library(car)
```

```
## Loading required package: carData
```

```
##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode
```

```
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(corrplot)
```

```
## corrplot 0.90 loaded
```

```
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(ggplot2)
TCH <- read_excel("C:/RBS/Business Forecasting/Group Project/TCH.xlsx")
View(TCH)
```

```
setDT(TCH)
class(TCH)
```

```
## [1] "data.table" "data.frame"
```

```
TCH[, Quarter := factor(Quarter, ordered = T)]
TCH[, Month := factor(Month, ordered = T)]
summary(TCH)
```

```
##      Year           Quarter     Month      Device_Hrs     Total_Inst_Hrs
##  Length:58          Q1:15   Apr    : 5   Min.   : 222.8   Min.   : 504.6
##  Class :character   Q2:15   Aug    : 5   1st Qu.: 900.2   1st Qu.:1974.4
##  Mode  :character   Q3:15   Dec    : 5   Median :1009.1   Median :2226.3
##                     Q4:13   Feb    : 5   Mean   : 974.7   Mean   :2167.1
##                             Jan    : 5   3rd Qu.:1096.4   3rd Qu.:2453.1
##                             Jul    : 5   Max.   :1412.5   Max.   :2955.8
##                             (Other):28
##      BE_SRG             RPM
##  Min.   :-0.8847   Min.   :  2908236
```

```
##  1st Qu.: 3.9990   1st Qu.: 55277932
##  Median : 4.3928   Median : 79281830
##  Mean   : 4.2143   Mean   : 68488064
##  3rd Qu.: 5.0447   3rd Qu.: 86568943
##  Max.   : 6.4936   Max.   :101794185
##                    NA's   :2
```

Creating a lag for the Business Expectation Sales Revenue Growth and Revenue Passenger Miles

```
TCH[,prev_BE_SRG:=shift(BE_SRG, n=1)]
TCH[,prev_RPM:=shift(RPM, n=2)]
head(TCH)
```

```
##       Year Quarter Month Device_Hrs Total_Inst_Hrs   BE_SRG      RPM
## 1: 2016-12      Q4   Dec     801.83        1853.99 4.039336 76957615
## 2: 2017-01      Q1   Jan     995.09        2446.80 4.473494 71433297
## 3: 2017-02      Q1   Feb     962.00        2169.17 4.302200 64261254
## 4: 2017-03      Q1   Mar    1130.24        2768.35 4.308913 80838984
## 5: 2017-04      Q2   Apr    1054.71        2291.76 4.692054 79494360
## 6: 2017-05      Q2   May    1044.95        2172.54 4.178750 83542041
##    prev_BE_SRG prev_RPM
## 1:          NA       NA
## 2:    4.039336       NA
## 3:    4.473494 76957615
## 4:    4.302200 71433297
## 5:    4.308913 64261254
## 6:    4.692054 80838984
```

```
plot1 <- TCH %>%
 ggplot(aes(x=Device_Hrs, y=prev_BE_SRG, color = Quarter)) +
 ylab("Previous Month's Business Expectations %") +
 xlab("Training Device Hours") +
 geom_point() +
 geom_smooth(method="lm", linetype = 6, se=TRUE, color = 'green')
plot2 <- TCH %>%
 ggplot(aes(x=Device_Hrs, y=prev_RPM, color = Quarter)) +
 ylab("Previous Month's Revenue Passenger Miles %") +
 xlab("Training Device Hours") +
 geom_point() +
 geom_smooth(method="lm", linetype = 6, se=TRUE, color = 'blue')
grid.arrange(plot1,plot2, nrow = 1)
```

```
## `geom_smooth()` using formula 'y ~ x'
```

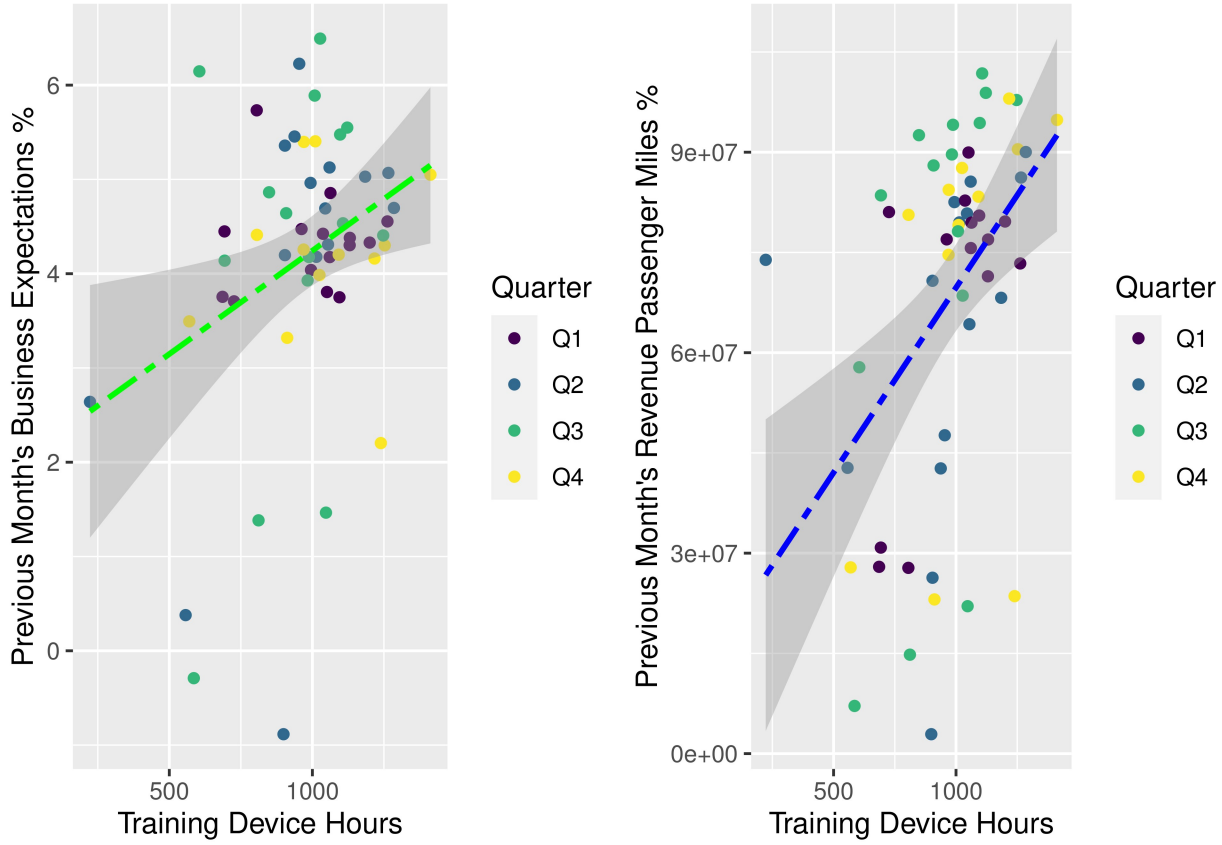
```
## Warning: Removed 1 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```

```
## `geom_smooth()` using formula 'y ~ x'
```

```
## Warning: Removed 2 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 2 rows containing missing values (geom_point).
```



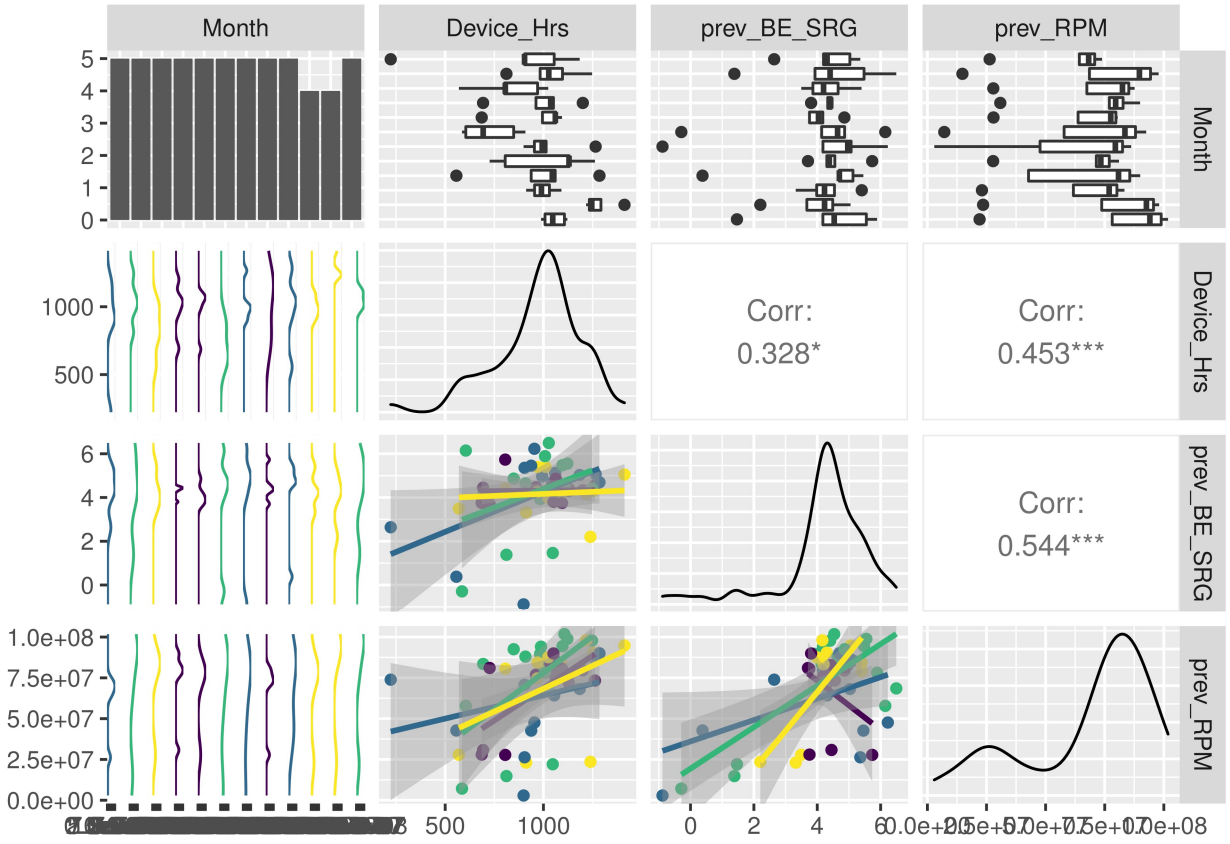

```
str(TCH)
```

```
## Classes 'data.table' and 'data.frame':   58 obs. of  9 variables:
##  $ Year          : chr  "2016-12" "2017-01" "2017-02" "2017-03" ...
##  $ Quarter       : Ord.factor w/ 4 levels "Q1"<"Q2"<"Q3"<..: 4 1 1 1 2 2 2 3 3 3 ...
##  $ Month         : Ord.factor w/ 12 levels "Apr"<"Aug"<"Dec"<..: 3 5 4 8 1 9 7 6 2 12 ...
##  $ Device_Hrs    : num  802 995 962 1130 1055 ...
##  $ Total_Inst_Hrs: num  1854 2447 2169 2768 2292 ...
##  $ BE_SRG        : num  4.04 4.47 4.3 4.31 4.69 ...
##  $ RPM           : num  76957615 71433297 64261254 80838984 79494360 ...
##  $ prev_BE_SRG   : num  NA 4.04 4.47 4.3 4.31 ...
##  $ prev_RPM      : num  NA NA 76957615 71433297 64261254 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
ggpairs(TCH, c(3, 4, 8, 9), lower = list(mapping = aes(color = Quarter), continuous = 'smooth',
 combo = 'facetdensity'))
```

```
## Warning: Removed 1 rows containing non-finite values (stat_boxplot).

## Warning: Removed 2 rows containing non-finite values (stat_boxplot).

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removing 1 row that contained a missing value
```

```
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 2 rows containing missing values

## Warning: Removed 1 rows containing non-finite values (stat_density).

## Warning: Removed 1 rows containing non-finite values (stat_smooth).

## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 1 rows containing non-finite values (stat_density).

## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 2 rows containing missing values

## Warning: Removed 2 rows containing non-finite values (stat_density).

## Warning: Removed 2 rows containing non-finite values (stat_smooth).

## Warning: Removed 2 rows containing missing values (geom_point).

## Warning: Removed 2 rows containing non-finite values (stat_smooth).

## Warning: Removed 2 rows containing missing values (geom_point).

## Warning: Removed 2 rows containing non-finite values (stat_density).
```



```
TH <- na.omit(TCH)
THM <- TH %>%
 select(Device_Hrs, prev_BE_SRG, prev_RPM)
cor(TH$Device_Hrs, TH$prev_BE_SRG)
```

```
## [1] 0.3307934
```

```
summary(TH)
```

```
##      Year            Quarter     Month      Device_Hrs     Total_Inst_Hrs
##  Length:54          Q1:14   Apr    : 5   Min.   : 222.8   Min.   : 504.6
##  Class :character   Q2:15   Feb    : 5   1st Qu.: 900.2   1st Qu.:1974.4
##  Mode  :character   Q3:13   Jul    : 5   Median :1012.0   Median :2266.4
##                     Q4:12   Jun    : 5   Mean   : 976.0   Mean   :2167.4
##                             Mar    : 5   3rd Qu.:1104.4   3rd Qu.:2455.8
##                             May    : 5   Max.   :1412.5   Max.   :2955.8
##                             (Other):24
##      BE_SRG              RPM             prev_BE_SRG         prev_RPM
##  Min.   :-0.8847   Min.   :  2908236   Min.   :-0.8847   Min.   :  2908236
##  1st Qu.: 3.9428   1st Qu.: 50189203   1st Qu.: 3.9428   1st Qu.: 50189203
##  Median : 4.3550   Median : 79477888   Median : 4.3550   Median : 79477888
##  Mean   : 4.1620   Mean   : 68276679   Mean   : 4.1246   Mean   : 68307812
##  3rd Qu.: 5.0123   3rd Qu.: 87287238   3rd Qu.: 4.9380   3rd Qu.: 87287238
##  Max.   : 6.4936   Max.   :101794185   Max.   : 6.2266   Max.   :101794185
##
```

```
cor(TH$Device_Hrs, TH$prev_RPM)
```

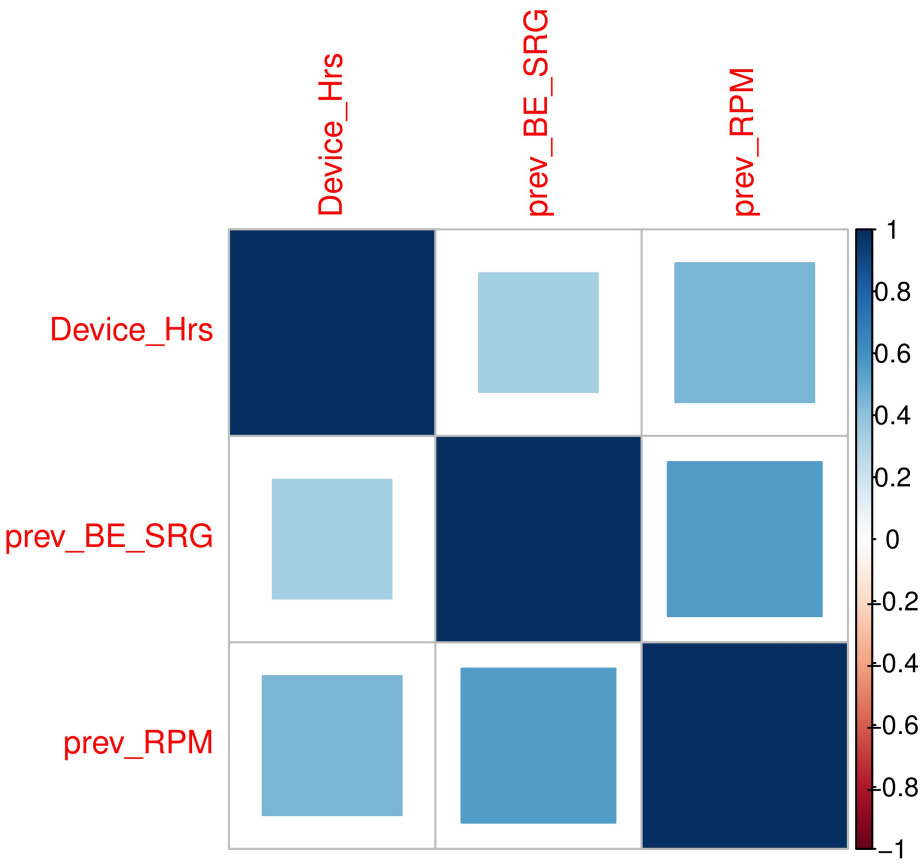
```
## [1] 0.4532032
```

```
THM <- cor(THM)
as.matrix(THM)
```

```
##             Device_Hrs prev_BE_SRG  prev_RPM
## Device_Hrs   1.0000000   0.3307934 0.4532032
## prev_BE_SRG  0.3307934   1.0000000 0.5564797
## prev_RPM     0.4532032   0.5564797 1.0000000
```

Prev RPM Shows it can be used to forecast Device hours. Solid correlation level

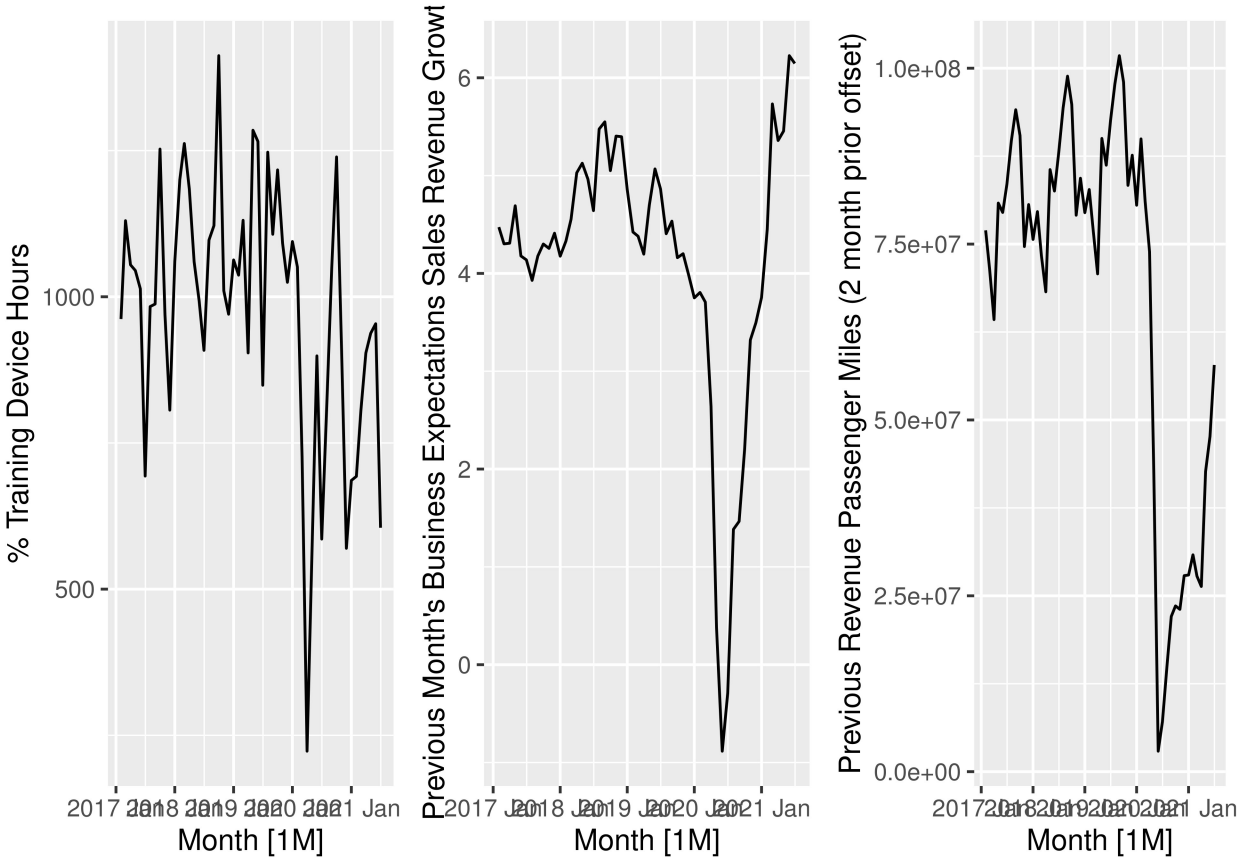
```
corrplot(THM, method = 'square')
```



Possible Multi-collinearity between SRG and RPM?

```
TH_ts <- TH %>%
 mutate(Month = yearmonth(Year)) %>%
 as_tsibble(index = Month)
```

```
plotts1 <- TH_ts %>%
 pivot_longer(c(Device_Hrs), names_to="Series") %>%
 autoplot(value) +
 labs(y = "% Training Device Hours")
plotts2 <- TH_ts %>%
 pivot_longer(c(prev_BE_SRG), names_to="Series") %>%
 autoplot(value) +
 labs(y = "Previous Month's Business Expectations Sales Revenue Growth")
plotts3 <- TH_ts %>%
 pivot_longer(c(prev_RPM), names_to="Series") %>%
 autoplot(value) +
 labs(y = "Previous Revenue Passenger Miles (2 month prior offset)")
grid.arrange(plotts1, plotts2, plotts3, nrow = 1)
```




```
str(TH_ts)
```

```
## tbl_ts [54 x 9] (S3: tbl_ts/tbl_df/tbl/data.frame)
##  $ Year          : chr [1:54] "2017-02" "2017-03" "2017-04" "2017-05" ...
##  $ Quarter       : Ord.factor w/ 4 levels "Q1"<"Q2"<"Q3"<..: 1 1 2 2 2 3 3 3 4 4 ...
##  $ Month         : mth [1:54] 2017 Feb, 2017 Mar, 2017 Apr, 2017 May, 2017 Jun, 2017 Jul,...
##  $ Device_Hrs    : num [1:54] 962 1130 1055 1045 1014 ...
##  $ Total_Inst_Hrs: num [1:54] 2169 2768 2292 2173 2367 ...
##  $ BE_SRG        : num [1:54] 4.3 4.31 4.69 4.18 4.14 ...
##  $ RPM           : num [1:54] 64261254 80838984 79494360 83542041 89667877 ...
##  $ prev_BE_SRG   : num [1:54] 4.47 4.3 4.31 4.69 4.18 ...
##  $ prev_RPM      : num [1:54] 76957615 71433297 64261254 80838984 79494360 ...
##  - attr(*, ".internal.selfref")=<externalptr>
##  - attr(*, "key")= tibble [1 x 1] (S3: tbl_df/tbl/data.frame)
##   ..$ .rows: list<int> [1:1]
##   .. ..$ : int [1:54] 1 2 3 4 5 6 7 8 9 10 ...
##   .. ..@ ptype: int(0)
##  - attr(*, "index")= chr "Month"
##   ..- attr(*, "ordered")= logi TRUE
##  - attr(*, "index2")= chr "Month"
##  - attr(*, "interval")= interval [1:1] 1M
##   ..@ .regular: logi TRUE
```

Time Series Regression Models

```
Tlmb <- lm(Device_Hrs~prev_BE_SRG + prev_RPM + Month + Month:prev_RPM + Month:prev_BE_SRG, data
= TH_ts )
summary(Tlmb)
```

```
##
## Call:
## lm(formula = Device_Hrs ~ prev_BE_SRG + prev_RPM + Month + Month:prev_RPM +
##     Month:prev_BE_SRG, data = TH_ts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -671.30 -121.01   16.79  130.75  400.24
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -1.045e+04  1.008e+04  -1.036   0.3052
## prev_BE_SRG        3.039e+03  1.687e+03   1.801   0.0779 .
## prev_RPM          -6.108e-06  9.227e-05  -0.066   0.9475
## Month              6.059e-01  5.473e-01   1.107   0.2738
## prev_RPM:Month     4.254e-10  5.011e-09   0.085   0.9327
## prev_BE_SRG:Month -1.636e-01  9.150e-02  -1.787   0.0802 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 197.7 on 48 degrees of freedom
## Multiple R-squared:  0.2888, Adjusted R-squared:  0.2147
## F-statistic: 3.898 on 5 and 48 DF,  p-value: 0.004795
```

We see and R2 of 0.2147 which isn't very strong, thogh the p-value of 0.0047 is good, and the F-stat is over 1

Model with one independent variable

```
Tlms <- lm(Device_Hrs~prev_RPM, data = TH_ts)
summary(Tlms)
```

```
##
## Call:
## lm(formula = Device_Hrs ~ prev_RPM, data = TH_ts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -773.90  -78.84   21.10  118.14  429.44
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.223e+02  7.439e+01   9.710 2.85e-13 ***
## prev_RPM    3.714e-06  1.013e-06   3.666 0.000579 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 200.7 on 52 degrees of freedom
## Multiple R-squared:  0.2054, Adjusted R-squared:  0.1901
## F-statistic: 13.44 on 1 and 52 DF,  p-value: 0.0005788
```

We have a stronger p-value and f-stat, thought the R2 is worse

```
anova(Tlms, Tlmb)
```

```
## Analysis of Variance Table
##
## Model 1: Device_Hrs ~ prev_RPM
## Model 2: Device_Hrs ~ prev_BE_SRG + prev_RPM + Month + Month:prev_RPM +
##     Month:prev_BE_SRG
##   Res.Df     RSS Df Sum of Sq      F Pr(>F)
## 1     52 2095414
## 2     48 1875468  4    219946 1.4073 0.2458
```

The P-value is above 0.05 and the F-stat is relatively small so we can reject that the big model is better than the smaller one

Time Series Linear Model

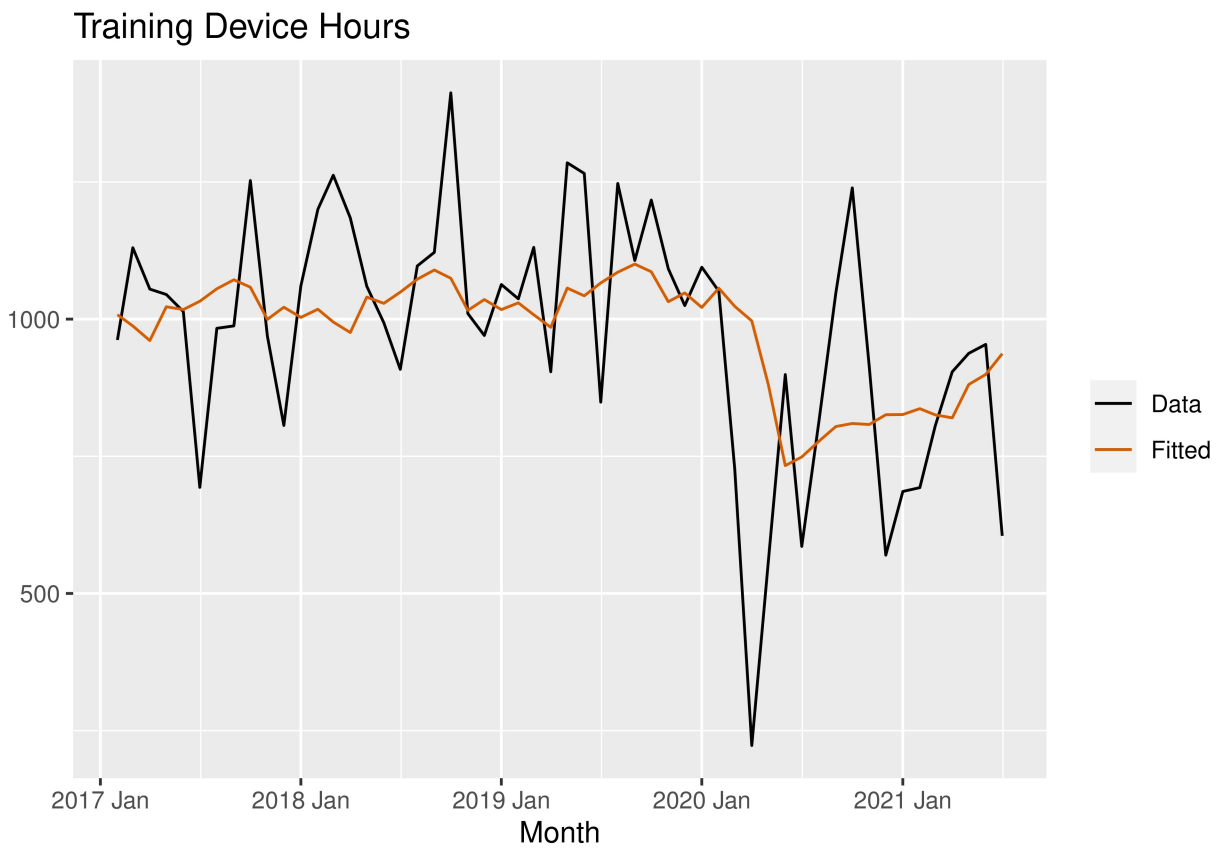
```
TSLMs <- TH_ts %>%
 model(TSLM(Device_Hrs ~ prev_RPM)) %>%
 report()
```

```
## Series: Device_Hrs
## Model: TSLM
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -773.90  -78.84   21.10  118.14  429.44
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.223e+02  7.439e+01   9.710 2.85e-13 ***
## prev_RPM    3.714e-06  1.013e-06   3.666 0.000579 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 200.7 on 52 degrees of freedom
## Multiple R-squared: 0.2054,  Adjusted R-squared: 0.1901
## F-statistic: 13.44 on 1 and 52 DF, p-value: 0.00057878
```

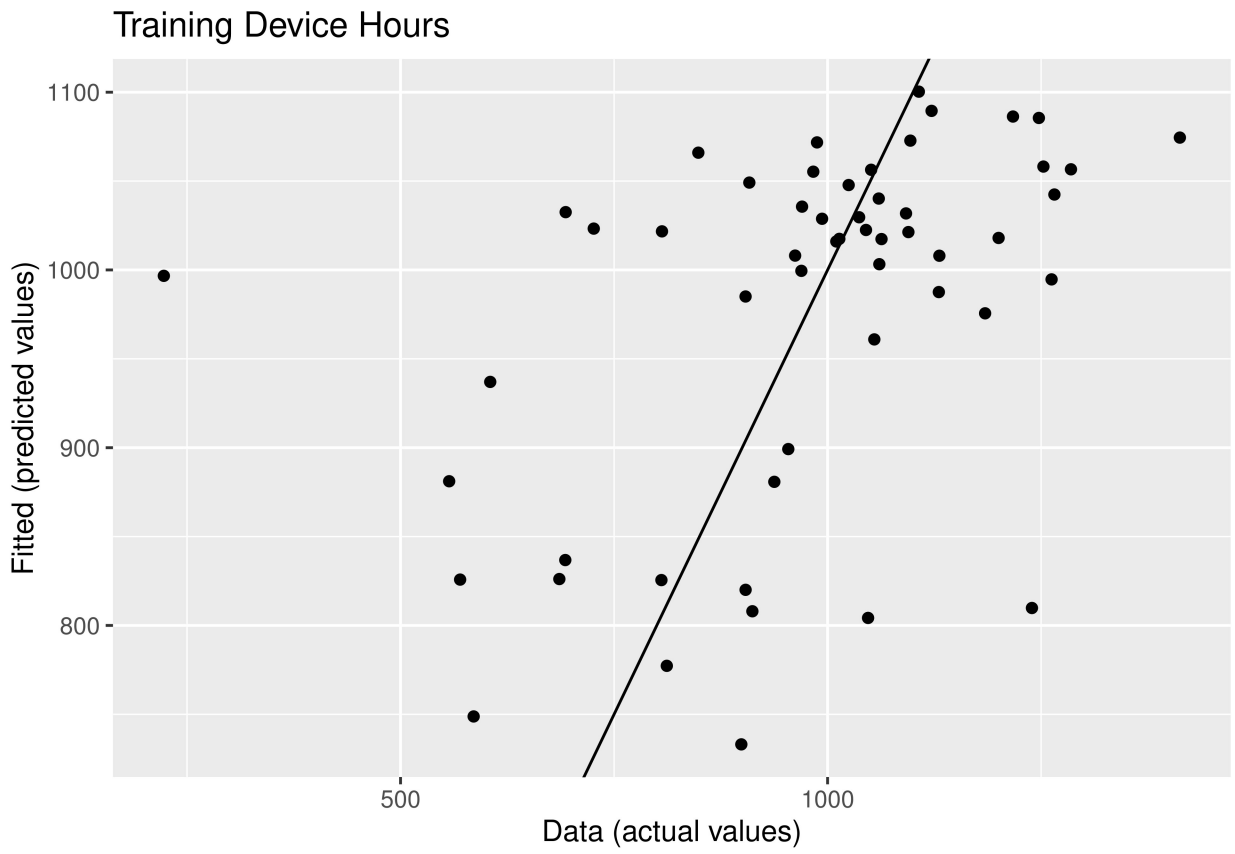
Adj R2 is simailar, but the P-value is very good and thte f-stat is ight

Plotting the Model

```
augment(TSLMs) %>%
 ggplot(aes(x = Month)) +
 geom_line(aes(y = Device_Hrs, colour = "Data")) +
 geom_line(aes(y = .fitted, colour = "Fitted")) +
 labs(y = NULL,
 title = "Training Device Hours"
 ) +
 scale_colour_manual(values=c(Data="black",Fitted="#D55E00")) +
 guides(colour = guide_legend(title = NULL))
```

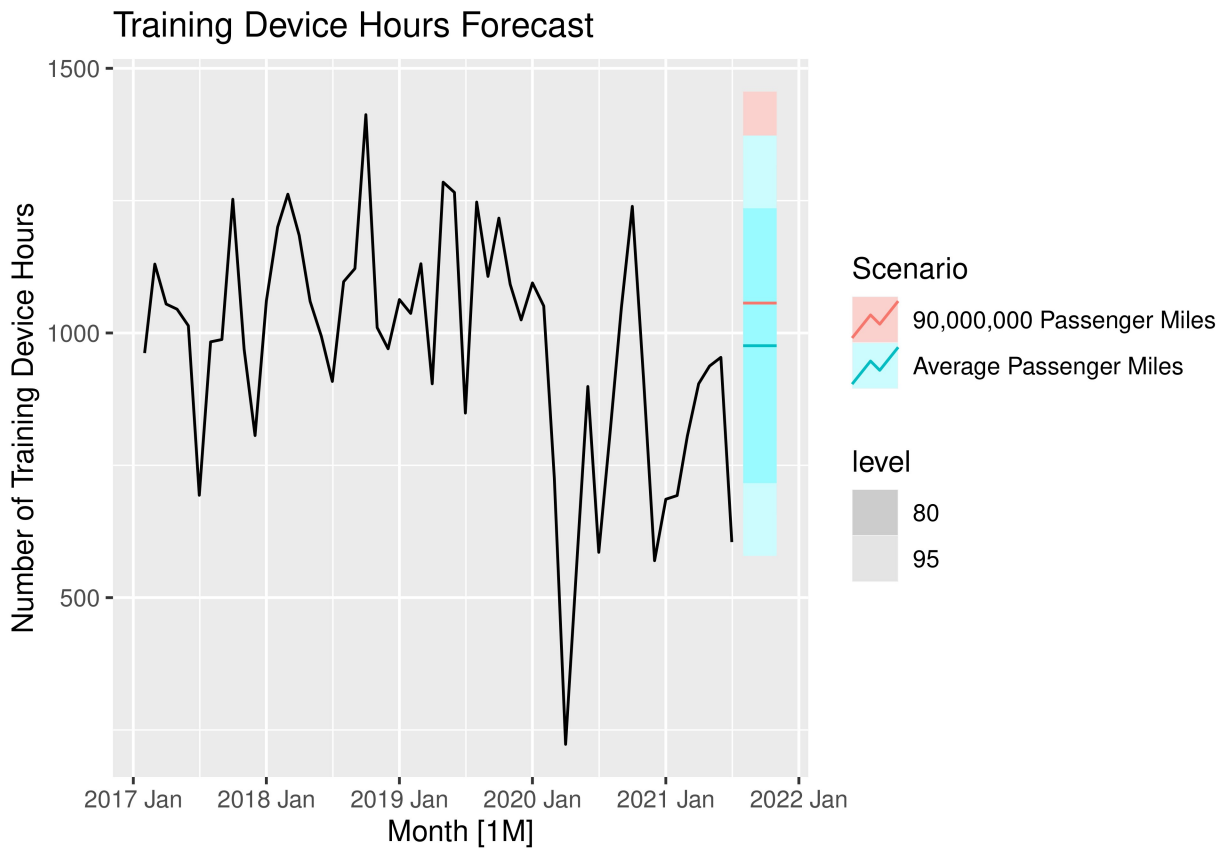



```
augment(TSLMs) %>%
 ggplot(aes(x = Device_Hrs, y = .fitted)) +
 geom_point() +
 labs(
 y = "Fitted (predicted values)",
 x = "Data (actual values)",
 title = "Training Device Hours"
 ) +
 geom_abline(intercept = 0, slope = 1)
```



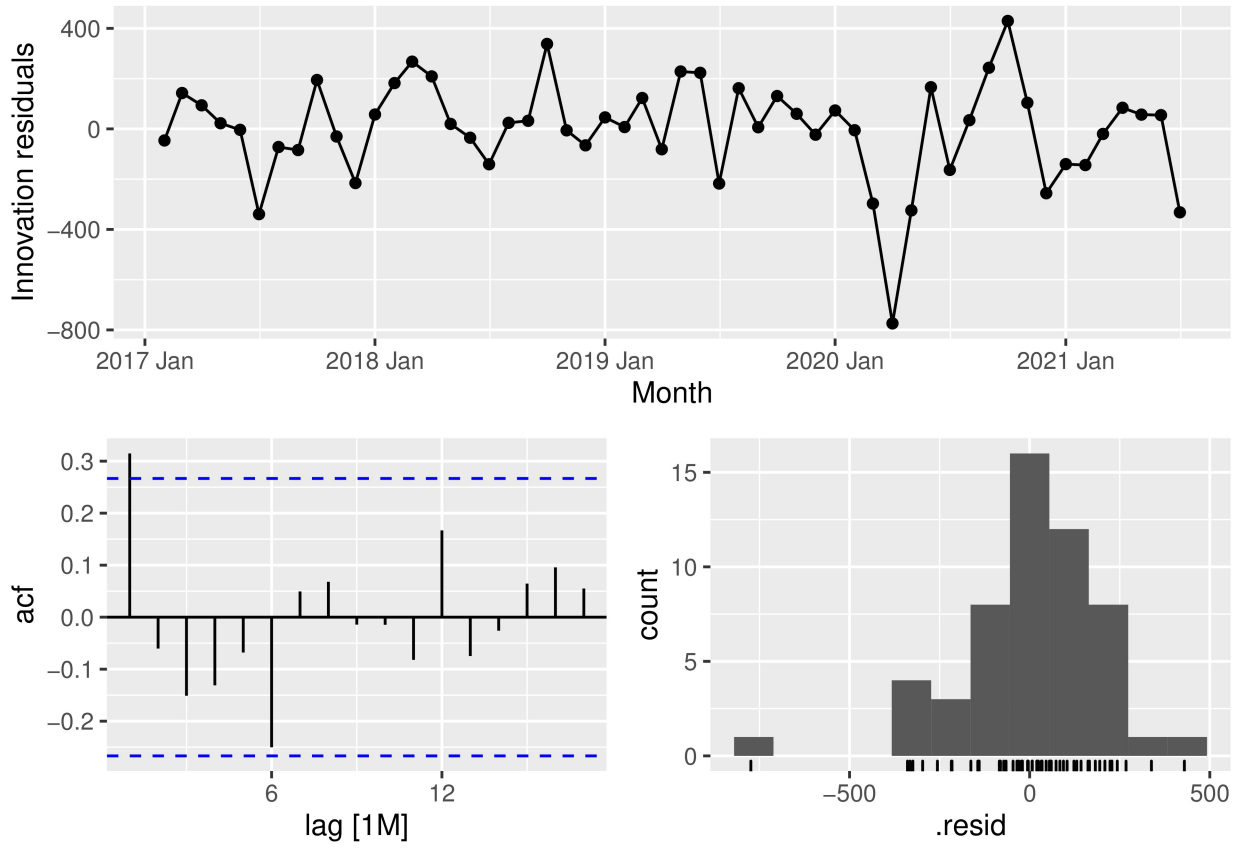
There is some semblance of a trend, though it is not very strong

```
fit_TH <- TH_ts %>%
 model(TSLM(Device_Hrs ~ prev_RPM))
new_TH <- scenarios(
 "Average Passenger Miles" = new_data(TH_ts, 4) %>%
 mutate(prev_RPM = mean(TH_ts$prev_RPM)),
 "90,000,000 Passenger Miles" = new_data(TH_ts, 4) %>%
 mutate(prev_RPM = 90000000),
 names_to = "Scenario")
fcast <- forecast(fit_TH, new_TH)
TH_ts %>%
 autoplot(Device_Hrs) +
 autolayer(fcast) +
 labs(title = "Training Device Hours Forecast", y = "Number of Training Device Hours")
```



Residual Plot

```
fit_TH %>% gg_tsresiduals()
```



Residuals were relatively consistne until COVID struck