R Notebook

The following is your first chunk to start with. Remember, you can add chunks using the menu above (Insert -> R) or using the keyboard shortcut Ctrl+Alt+I. A good practice is to use different code chunks to answer different questions. You can delete this comment if you like.

Other useful keyboard shortcuts include Alt- for the assignment operator, and Ctrl+Shift+M for the pipe operator. You can delete these reminders if you don't want them in your report.

```
#setwd("C:/") #Don't forget to set your working directory before you start!

library("tidyverse")
```

```
## ── Attaching packages ──────────────────────────────────────────── tidyverse 1.3.0 ──

## ✓ ggplot2 3.2.1     ✓ purrr   0.3.3
## ✓ tibble  2.1.3     ✓ dplyr   0.8.5
## ✓ tidyr   1.0.0     ✓ stringr 1.4.0
## ✓ readr   1.3.1     ✓ forcats 0.4.0

## ── Conflicts ────────────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library("fpp3")
```

```
## ── Attaching packages ─────────────────────────────────────────────── fpp3 0.2 ──

## ✓ lubridate   1.7.4     ✓ feasts      0.1.3
## ✓ tsibble     0.8.6     ✓ fable       0.1.2
## ✓ tsibbledata 0.1.0

## ── Conflicts ──────────────────────────────────────────────────── fpp3_conflicts ──
## x lubridate::date()       masks base::date()
## x dplyr::filter()         masks stats::filter()
## x tsibble::id()           masks dplyr::id()
## x tsibble::interval()     masks lubridate::interval()
```

```
## x dplyr::lag()            masks stats::lag()
## x tsibble::new_interval() masks lubridate::new_interval()
```

```
library("plotly")
```

```
##
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':
##
##     last_plot

## The following object is masked from 'package:stats':
##
##     filter

## The following object is masked from 'package:graphics':
##
##     layout
```

```
library("skimr")
library("lubridate")
library("dplyr")
library("readr")
library("ggplot2")
library("tsibble")
library("feasts")
library("forecast")
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

##
## Attaching package: 'forecast'

## The following objects are masked from 'package:fabletools':
##
##     GeomForecast, StatForecast
```

Load retail sales data

Q1a

```
tsretail <- read_csv("retailSales.csv")
```

```
## Parsed with column specification:
## cols(
##   date = col_character(),
##   sales = col_double()
## )
```

```
tsretail
```

```
## # A tibble: 338 x 2
##    date     sales
##    <chr>    <dbl>
##  1 1/1/92  130683
##  2 2/1/92  131244
##  3 3/1/92  142488
##  4 4/1/92  147175
##  5 5/1/92  152420
##  6 6/1/92  151849
##  7 7/1/92  152586
##  8 8/1/92  152476
##  9 9/1/92  148158
## 10 10/1/92 155987
## # … with 328 more rows
```

###convert date from char to date class

```
tsretail <- tsretail%>%
            mutate(date = mdy(date))
tsretail
```

```
## # A tibble: 338 x 2
##    date        sales
##    <date>      <dbl>
##  1 1992-01-01 130683
##  2 1992-02-01 131244
##  3 1992-03-01 142488
##  4 1992-04-01 147175
##  5 1992-05-01 152420
##  6 1992-06-01 151849
##  7 1992-07-01 152586
##  8 1992-08-01 152476
##  9 1992-09-01 148158
## 10 1992-10-01 155987
## # … with 328 more rows
```

Convert to tsibble

Q1b

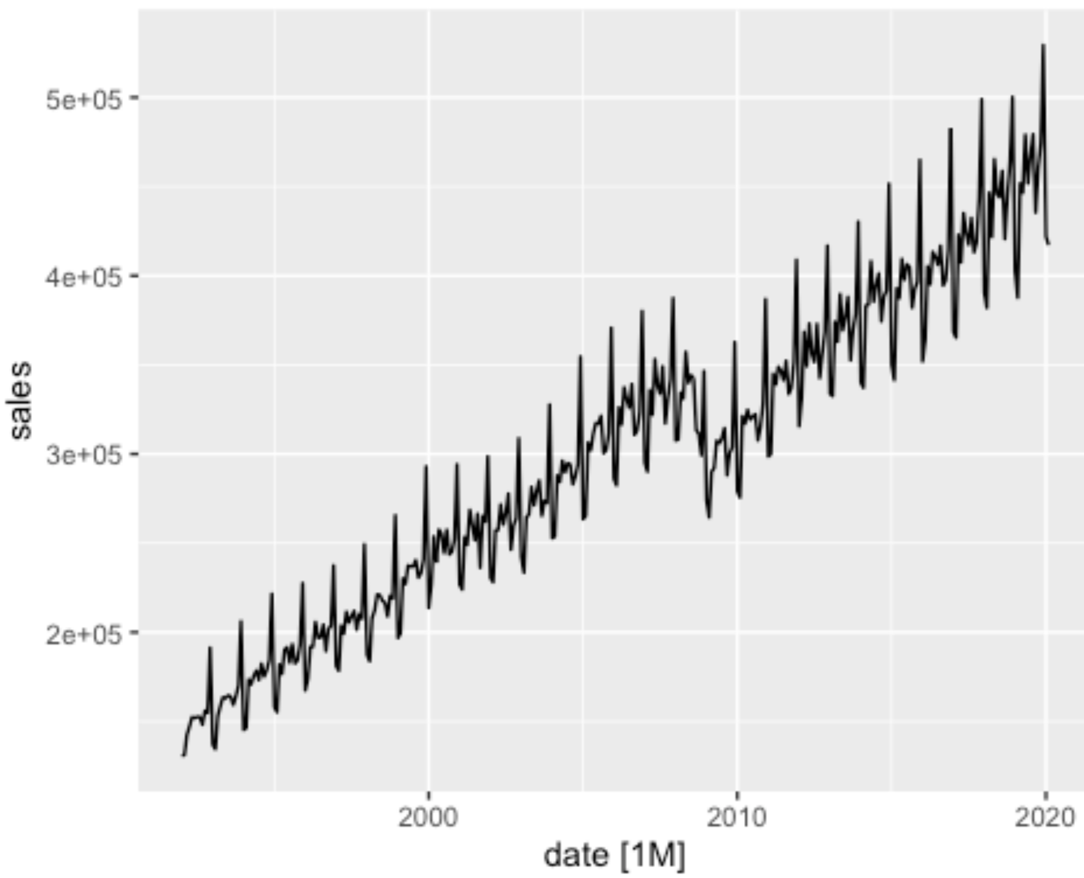
```
tsretail <-
  tsretail %>%
  mutate(date = yearmonth(date))%>%
  as_tsibble(index = date)
tsretail
```

```
## # A tsibble: 338 x 2 [1M]
##        date  sales
##       <mth>  <dbl>
##  1 1992 Jan 130683
##  2 1992 Feb 131244
```

```
##   3 1992 Mar 142488
##   4 1992 Apr 147175
##   5 1992 May 152420
##   6 1992 Jun 151849
##   7 1992 Jul 152586
##   8 1992 Aug 152476
##   9 1992 Sep 148158
##  10 1992 Oct 155987
## # … with 328 more rows
```

#Q1c

```
full_retail <- tsretail %>%
  autoplot(sales)
full_retail
```




```
retail_2010<-tsretail %>% filter(date > '2009-12-01')
retail_2010
```

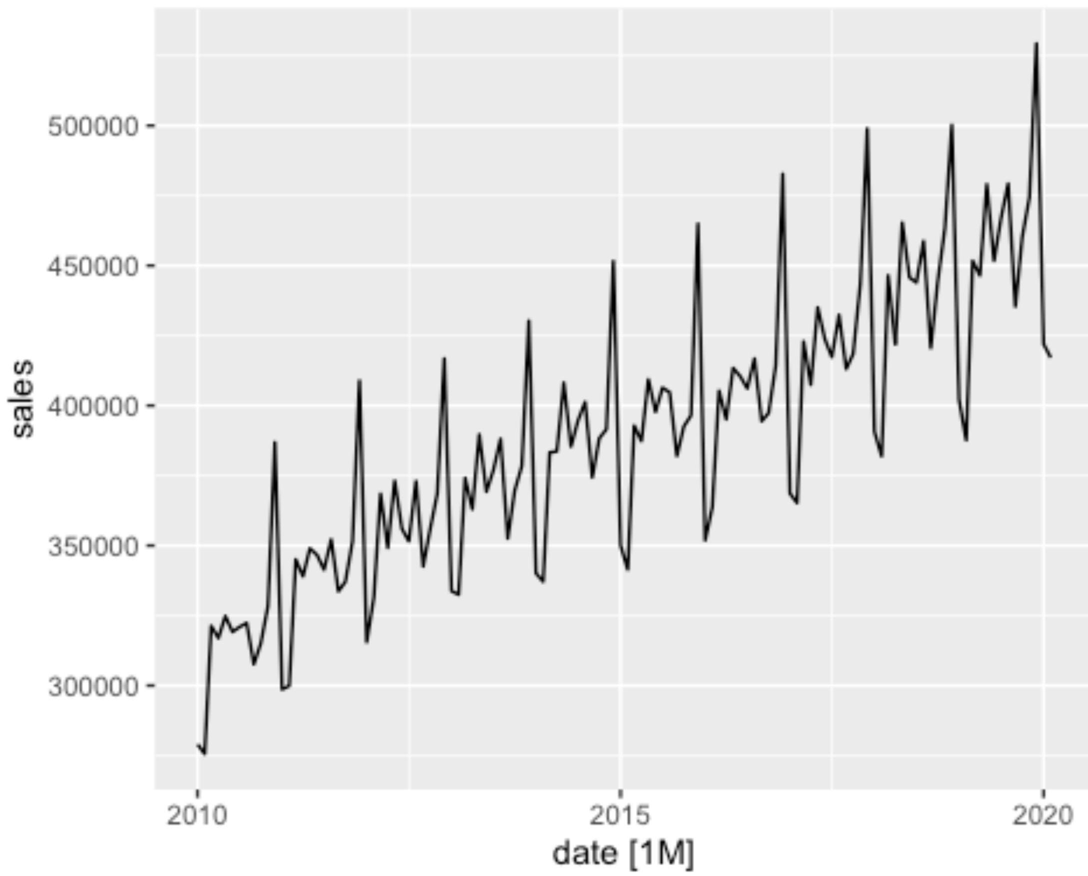
```
## # A tsibble: 122 x 2 [1M]
##         date  sales
##        <mth>  <dbl>
##   1 2010 Jan 279044
##   2 2010 Feb 275566
```

```
##   3 2010 Mar 321305
##   4 2010 Apr 316940
##   5 2010 May 324820
##   6 2010 Jun 319183
##   7 2010 Jul 320915
##   8 2010 Aug 322319
##   9 2010 Sep 307638
## 10 2010 Oct 315059
## # … with 112 more rows
```

```
retail_2010 <-
  retail_2010 %>%
  mutate(date = yearmonth(date))%>%
  as_tsibble(index = date)
retail_2010
```

```
## # A tsibble: 122 x 2 [1M]
##         date  sales
##        <mth>  <dbl>
##   1 2010 Jan 279044
##   2 2010 Feb 275566
##   3 2010 Mar 321305
##   4 2010 Apr 316940
##   5 2010 May 324820
##   6 2010 Jun 319183
##   7 2010 Jul 320915
##   8 2010 Aug 322319
##   9 2010 Sep 307638
## 10 2010 Oct 315059
## # … with 112 more rows
```

```
retail_2010 <- retail_2010 %>%
               autoplot(sales)
retail_2010
```

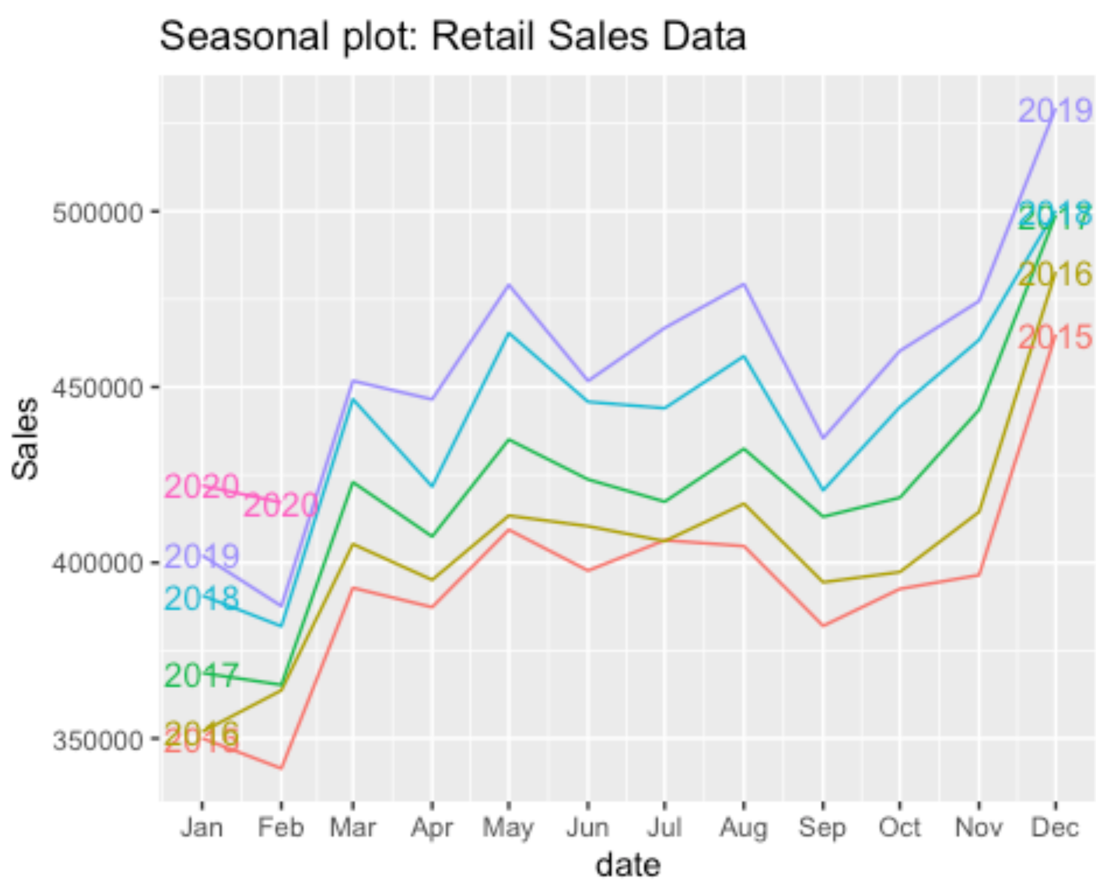


Q2a

```
retail_2015<-tsretail %>% filter(date > '2014-12-01')
retail_2015

## # A tsibble: 62 x 2 [1M]
##        date  sales
##       <mth>  <dbl>
##  1 2015 Jan 350067
##  2 2015 Feb 341459
##  3 2015 Mar 392848
##  4 2015 Apr 387352
##  5 2015 May 409376
##  6 2015 Jun 397752
##  7 2015 Jul 406393
##  8 2015 Aug 404729
##  9 2015 Sep 382020
## 10 2015 Oct 392545
## # … with 52 more rows

retail_2015 <-
  retail_2015 %>%
  mutate(date = yearmonth(date))%>%
  as_tsibble(index = date)
retail_2015
```

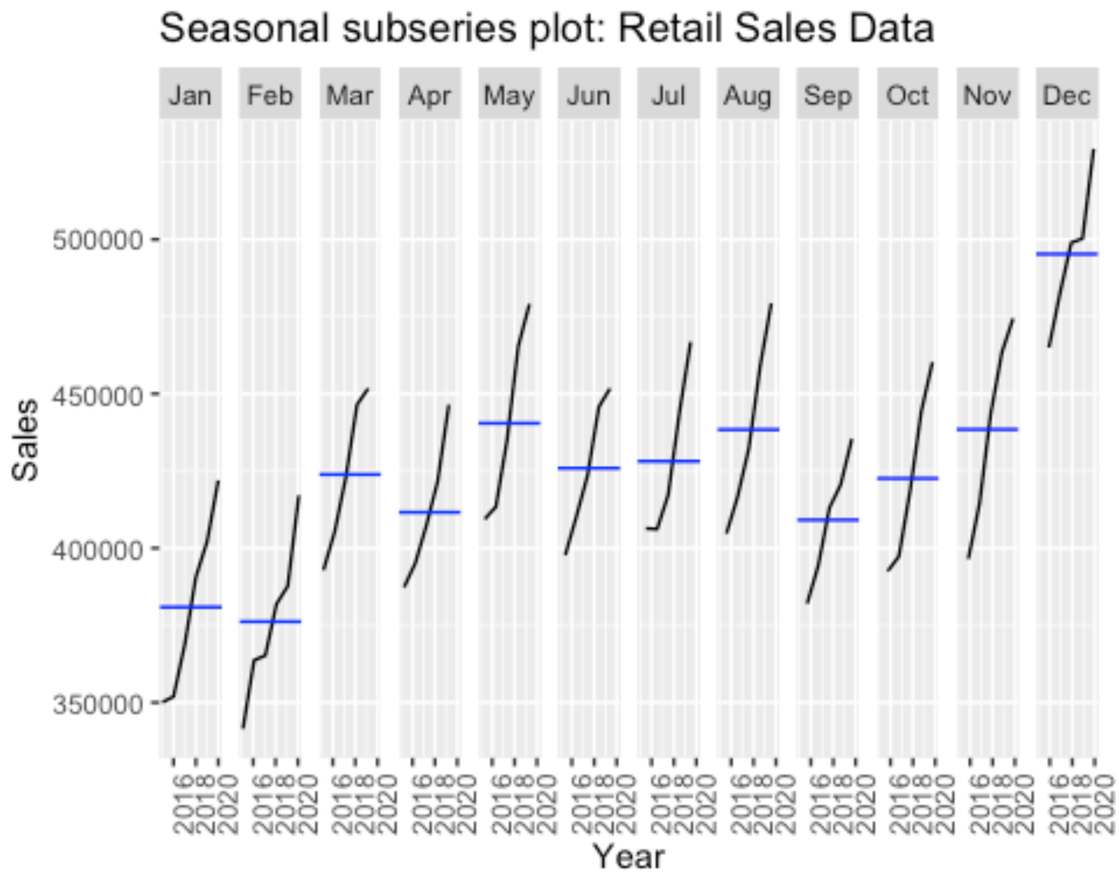
```
## # A tsibble: 62 x 2 [1M]
##         date  sales
##        <mth>  <dbl>
##   1 2015 Jan 350067
##   2 2015 Feb 341459
##   3 2015 Mar 392848
##   4 2015 Apr 387352
##   5 2015 May 409376
##   6 2015 Jun 397752
##   7 2015 Jul 406393
##   8 2015 Aug 404729
##   9 2015 Sep 382020
## 10 2015 Oct 392545
## # … with 52 more rows
```

```
retail_2015 %>% gg_season(sales, labels = "both") +
  ylab("Sales") +
  ggtitle("Seasonal plot: Retail Sales Data")
```



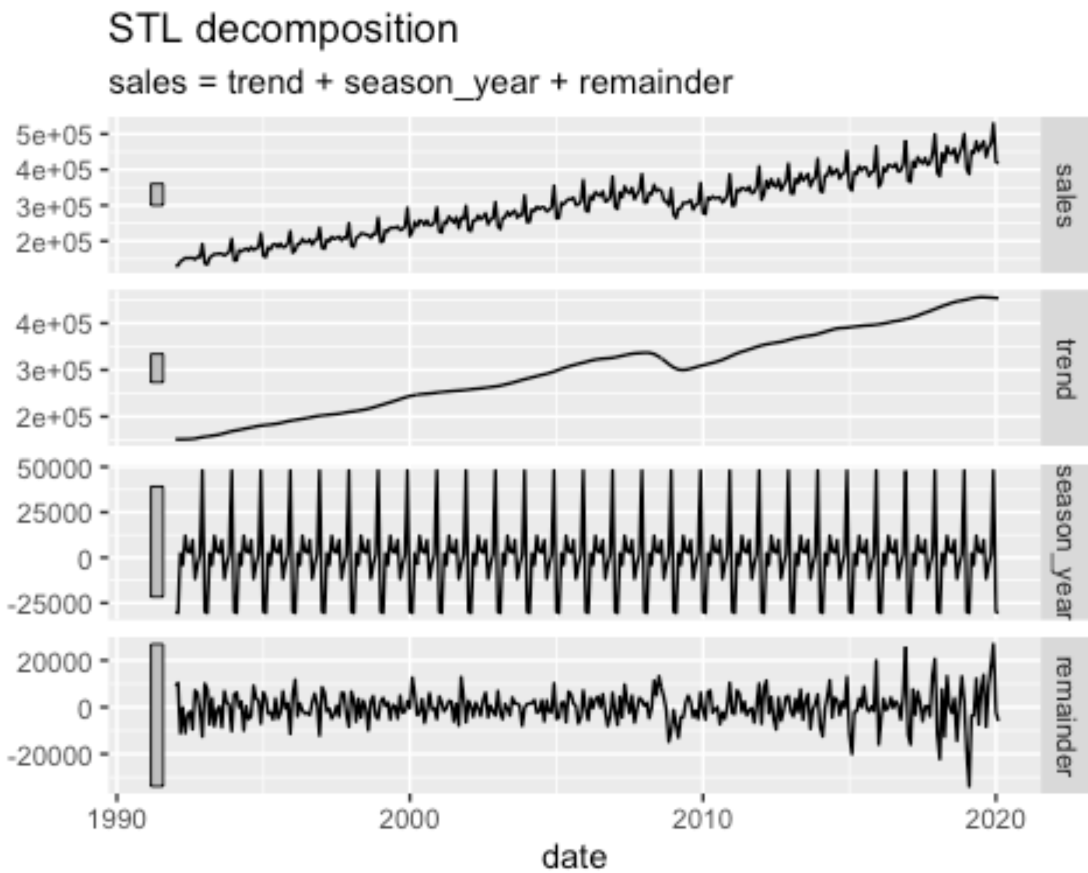

```
retail_2015 %>%
    gg_subseries(sales) +
    ylab("Sales") +
```

```
    xlab("Year") +
    ggtitle("Seasonal subseries plot: Retail Sales Data")
```



Q2b

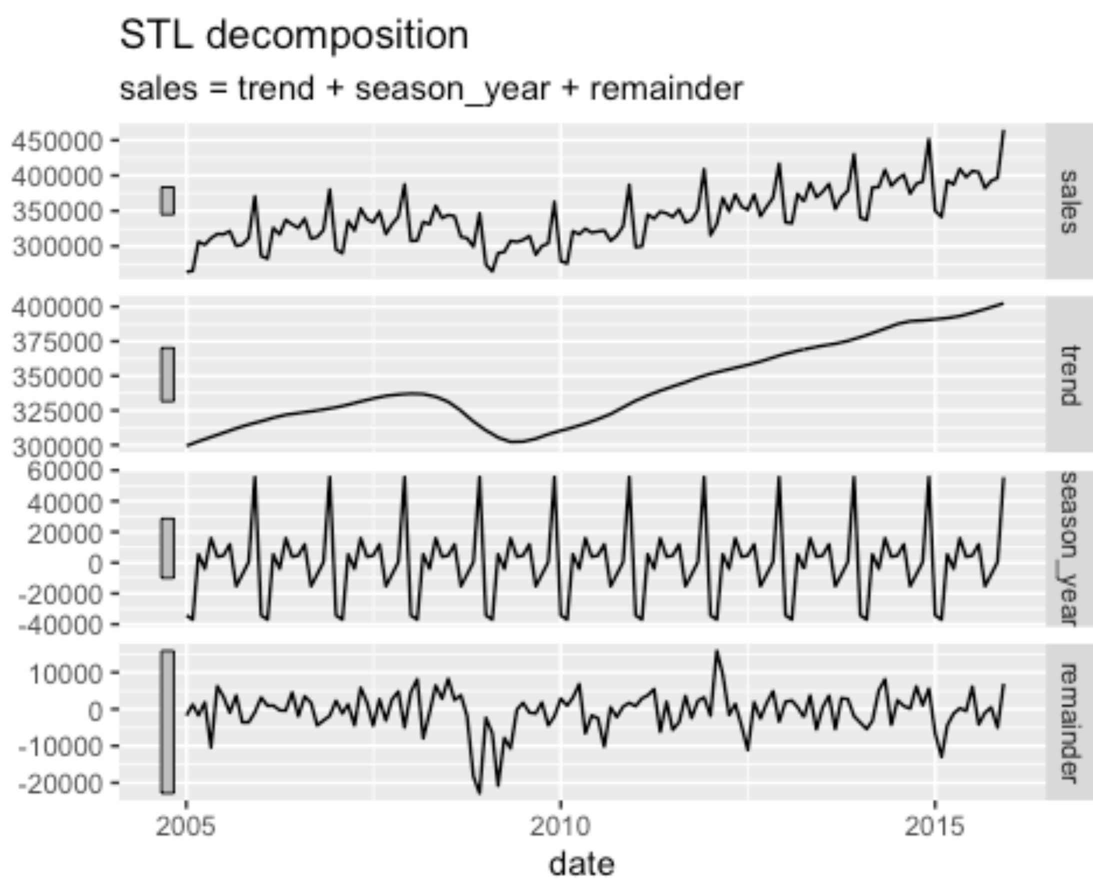
```
tsretail %>%
  model(STL(sales ~ trend(window=17) + season(window='periodic'),robust =
TRUE)) %>%
  components() %>%
  autoplot()
```




```
retail_515 <-tsretail %>% filter(date > '2004-12-01' & date < '2016-01-01')
retail_515

## # A tsibble: 132 x 2 [1M]
##        date  sales
##       <mth>  <dbl>
##  1 2005 Jan 263469
##  2 2005 Feb 265320
##  3 2005 Mar 306384
##  4 2005 Apr 302054
##  5 2005 May 311292
##  6 2005 Jun 317375
##  7 2005 Jul 316887
##  8 2005 Aug 321409
##  9 2005 Sep 300439
## 10 2005 Oct 302213
## # … with 122 more rows

retail_515 <-
  retail_515 %>%
  mutate(date = yearmonth(date))%>%
  as_tsibble(index = date)
retail_515
```

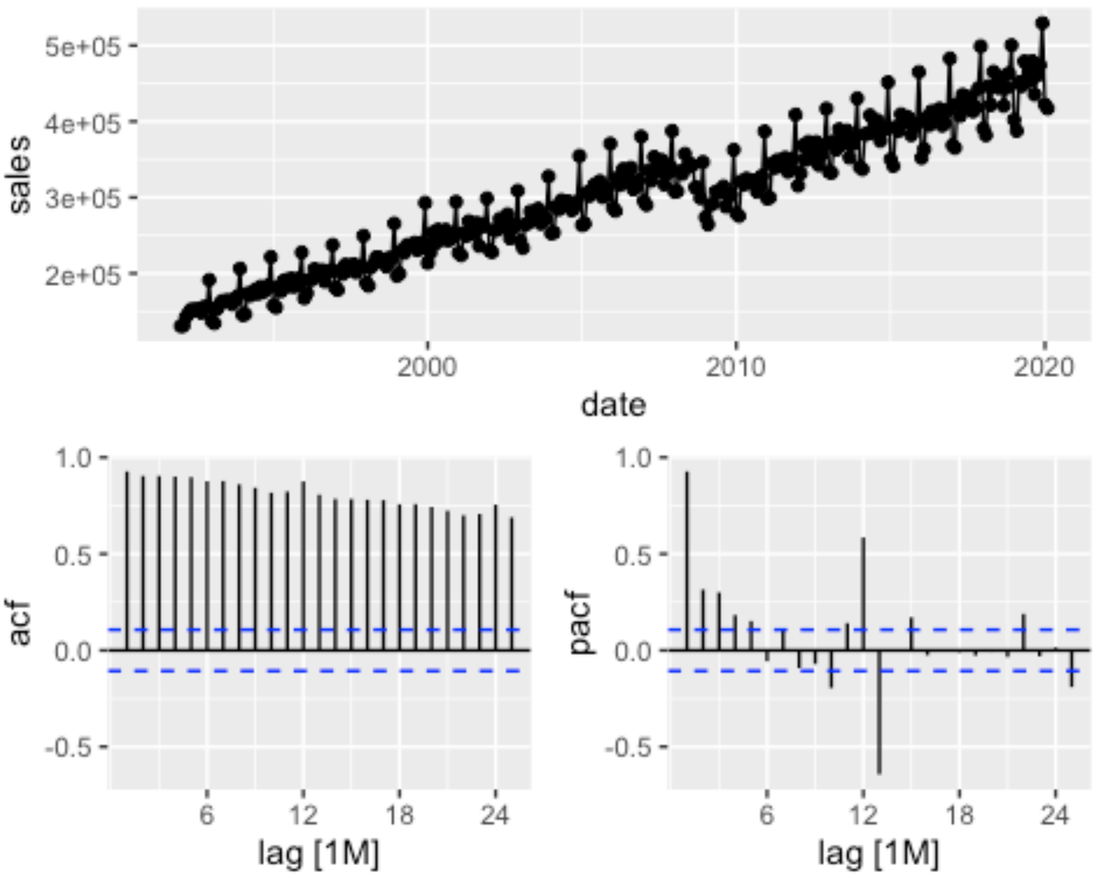
```
## # A tsibble: 132 x 2 [1M]
##        date  sales
##       <mth>  <dbl>
##   1 2005 Jan 263469
##   2 2005 Feb 265320
##   3 2005 Mar 306384
##   4 2005 Apr 302054
##   5 2005 May 311292
##   6 2005 Jun 317375
##   7 2005 Jul 316887
##   8 2005 Aug 321409
##   9 2005 Sep 300439
## 10 2005 Oct 302213
## # … with 122 more rows
```

```
retail_515 %>%
  model(STL(sales ~ trend(window=17) + season(window='periodic'),robust =
TRUE)) %>%
  components() %>%
  autoplot()
```



Q2c

```
tsretail %>%
  gg_tsdisplay(sales, plot_type='partial')
```

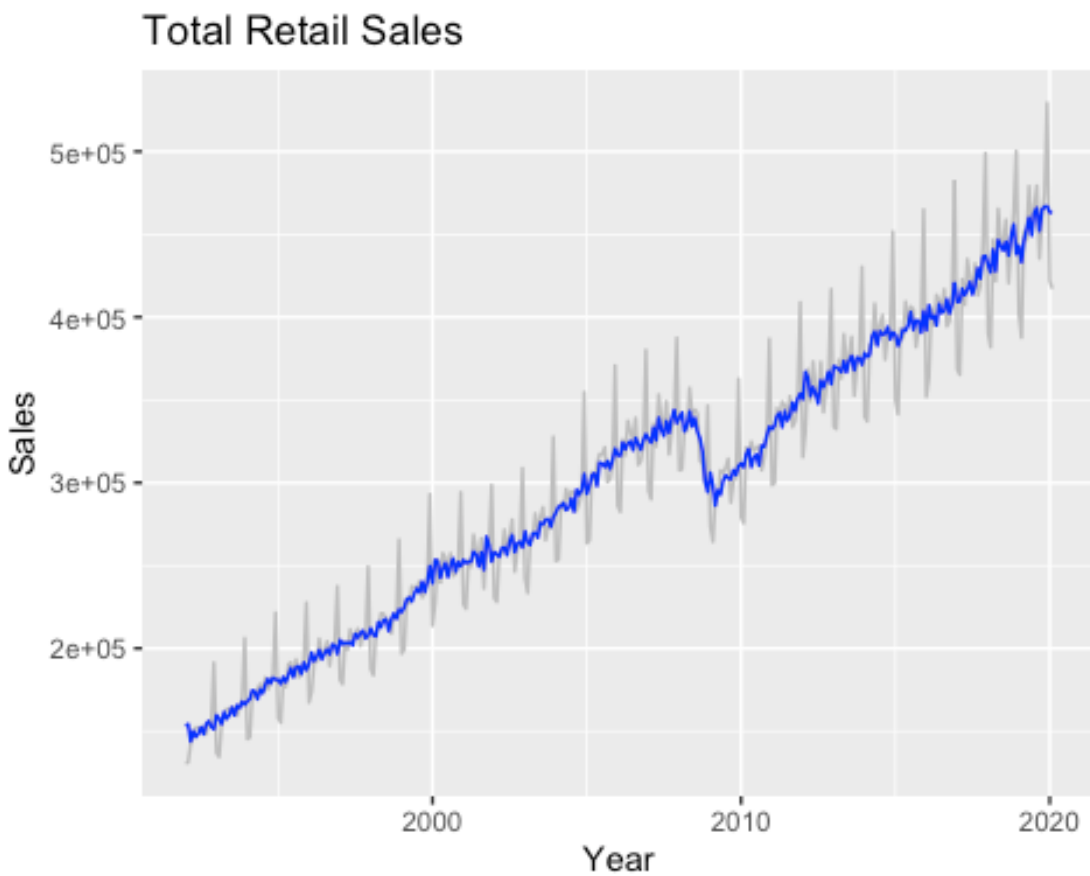


Q2d

```
dcmp <- tsretail %>%
  model(STL(sales))
components(dcmp)
```

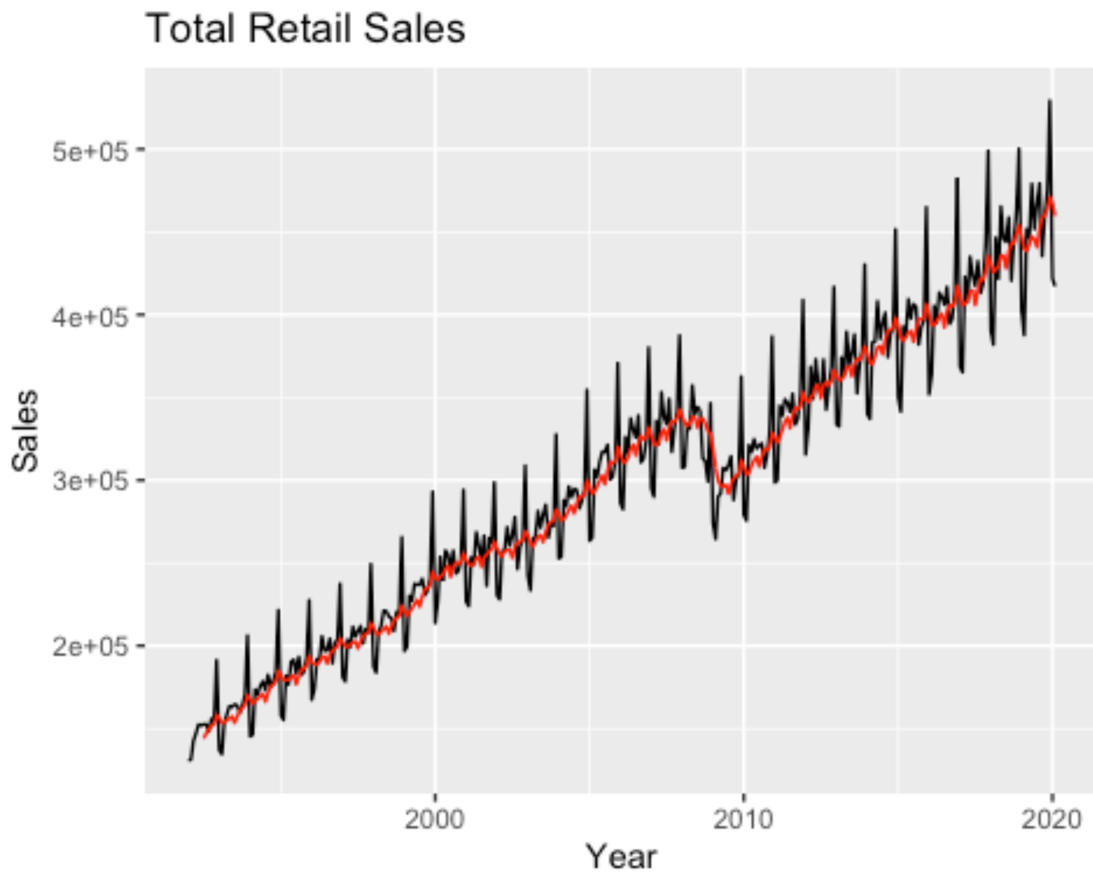
```
## # A dable:           338 x 7 [1M]
## # Key:               .model [1]
## # STL Decomposition: sales = trend + season_year + remainder
##    .model         date  sales   trend season_year remainder season_adjust
##    <chr>         <mth>  <dbl>   <dbl>       <dbl>     <dbl>         <dbl>
##  1 STL(sales) 1992 Jan 130683 148453.     -22505.     4735.       153188.
##  2 STL(sales) 1992 Feb 131244 148960.     -23009.     5292.       154253.
##  3 STL(sales) 1992 Mar 142488 149468.      -1326.    -5654.       143814.
##  4 STL(sales) 1992 Apr 147175 149976.      -2978.      177.       150153.
##  5 STL(sales) 1992 May 152420 150513.       5927.    -4020.       146493.
##  6 STL(sales) 1992 Jun 151849 151051.       3205.    -2407.       148644.
##  7 STL(sales) 1992 Jul 152586 151589.        294.      703.       152292.
##  8 STL(sales) 1992 Aug 152476 152155.       4343.    -4022.       148133.
##  9 STL(sales) 1992 Sep 148158 152722.      -6162.     1598.       154320.
## 10 STL(sales) 1992 Oct 155987 153289.        -33.3    2732.       156020.
## # … with 328 more rows
```

```
adj_retail<-tsretail %>%
  autoplot(sales, color='gray') +
  autolayer(components(dcmp), season_adjust, color='blue') +
  xlab("Year") + ylab("Sales") +
  ggtitle("Total Retail Sales")
adj_retail
```

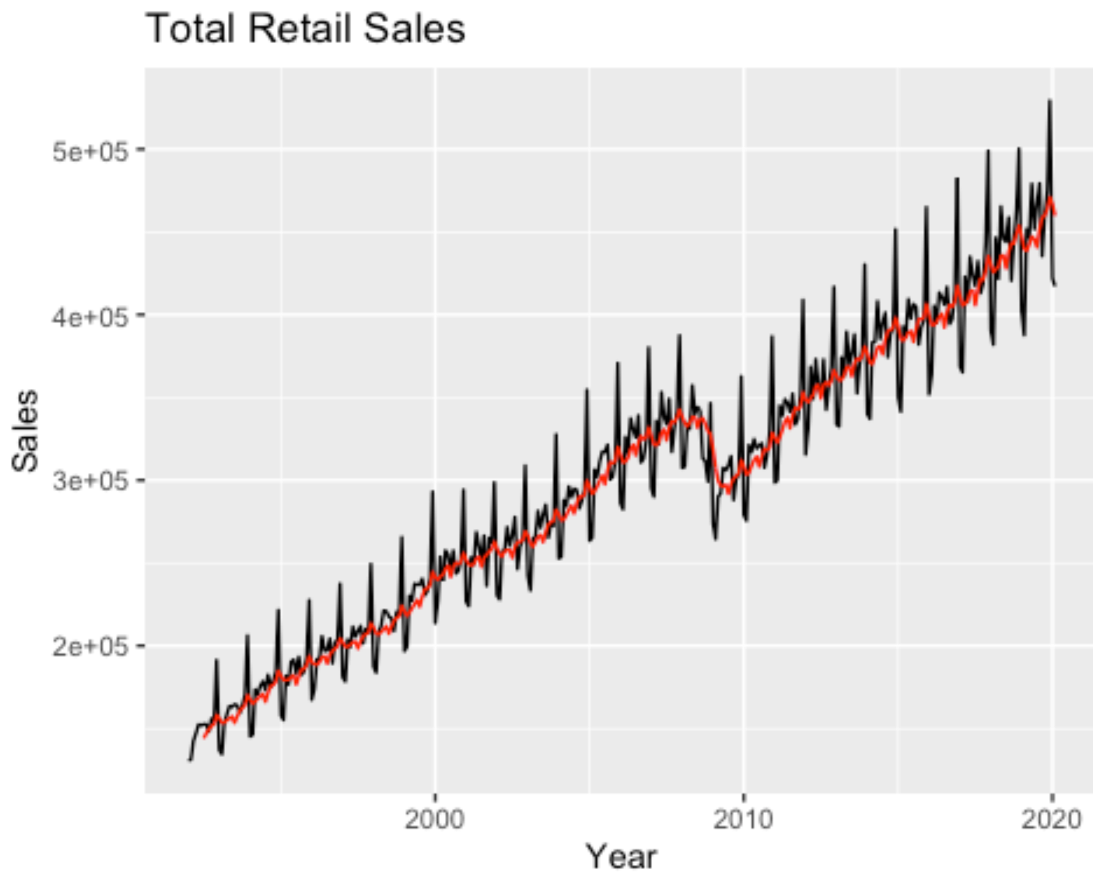


Q2e

```
tsretail_2ma <- tsretail %>%
  mutate(`2-MA` = slide_dbl(sales, mean, .size = 7))

tsretail_2ma %>%
  autoplot(sales) +
  autolayer(tsretail_2ma, `2-MA`, color='red') +
  xlab("Year") + ylab("Sales") +
  ggtitle("Total Retail Sales") +
  guides(colour=guide_legend(title="series"))

## Warning: Removed 6 rows containing missing values (geom_path).
```




```
tsretail_7ma <- tsretail %>%
  mutate(`7-MA` = slide_dbl(sales, mean, .size = 7))

tsretail_7ma %>%
  autoplot(sales) +
  autolayer(tsretail_7ma, `7-MA`, color='red') +
  xlab("Year") + ylab("Sales") +
  ggtitle("Total Retail Sales") +
  guides(colour=guide_legend(title="series"))

## Warning: Removed 6 rows containing missing values (geom_path).
```



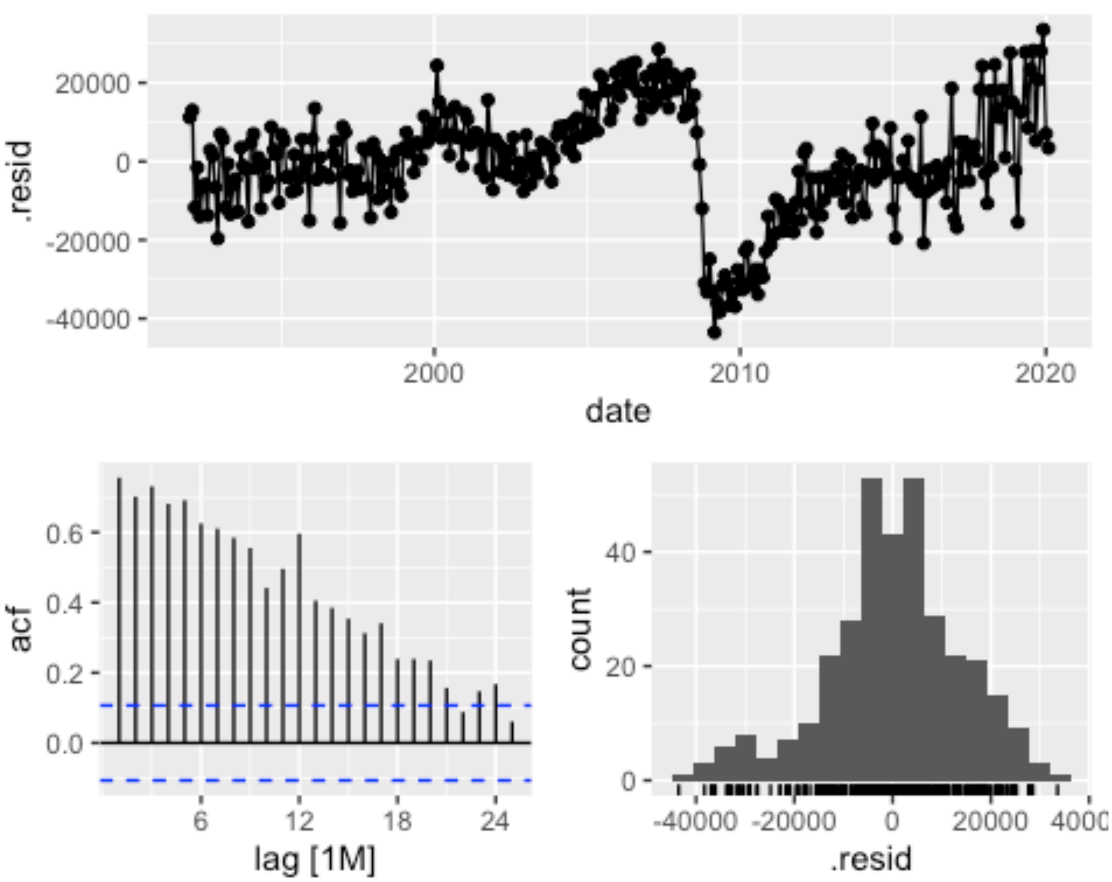
Q3a

```
fit_retail <- tsretail %>%
  model(TSLM(sales ~ trend() + season()))
report(fit_retail)

## Series: sales
## Model: TSLM
##
## Residuals:
##     Min      1Q Median      3Q     Max
## -43506  -6799    329   7662  33529
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)    118607.944   2948.209  40.231  < 2e-16 ***
## trend()           879.249      7.895 111.365  < 2e-16 ***
## season()year2   -2107.214   3717.967  -0.567    0.571
## season()year3   32961.493   3751.141   8.787  < 2e-16 ***
## season()year4   26615.138   3751.083   7.095 8.13e-12 ***
## season()year5   43380.853   3751.041  11.565  < 2e-16 ***
## season()year6   34385.747   3751.017   9.167  < 2e-16 ***
## season()year7   33746.927   3751.008   8.997  < 2e-16 ***
## season()year8   40570.572   3751.017  10.816  < 2e-16 ***
```

```
## season()year9   18758.787   3751.041   5.001 9.35e-07 ***
## season()year10  27201.181   3751.083   7.252 3.03e-12 ***
## season()year11  33160.718   3751.141   8.840  < 2e-16 ***
## season()year12  81780.970   3751.216  21.801  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14160 on 325 degrees of freedom
## Multiple R-squared: 0.9759,  Adjusted R-squared: 0.975
## F-statistic:  1098 on 12 and 325 DF, p-value: < 2.22e-16
```

Residual Diagnostics

```
fit_retail %>% gg_tsresiduals()
```

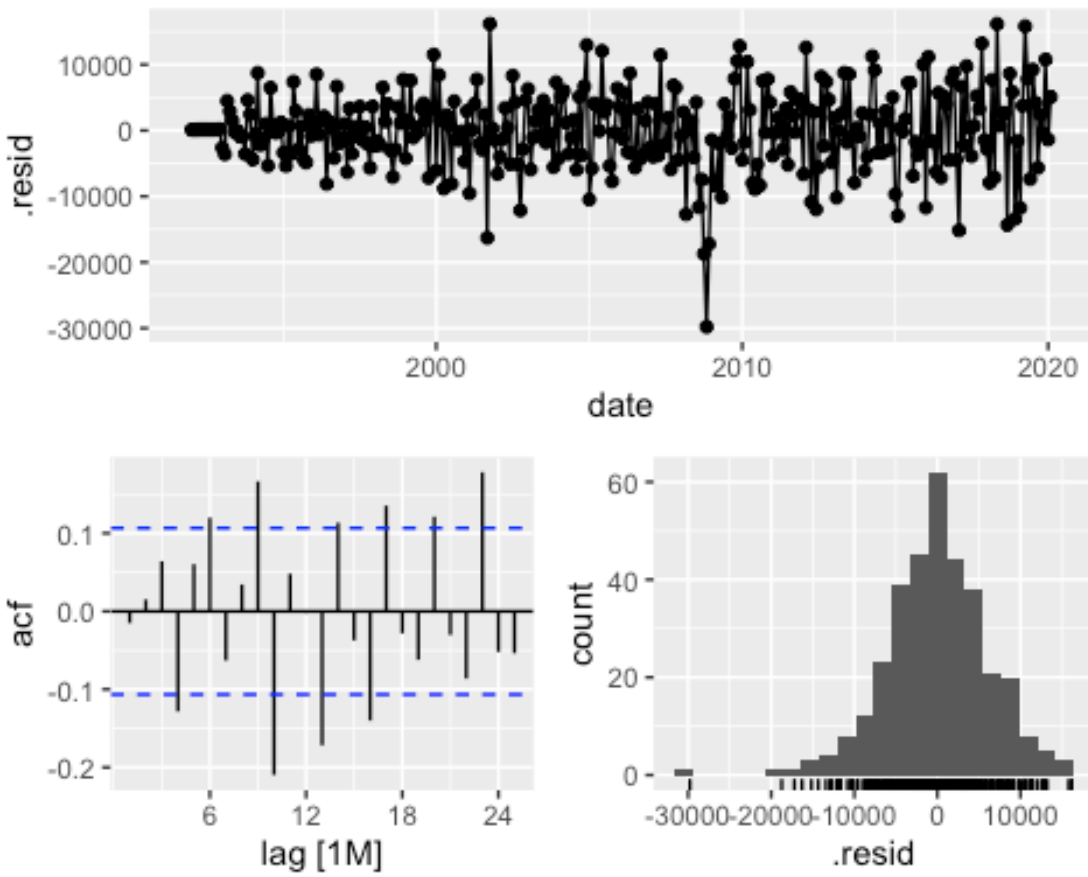


Q3b.

```
fit_retail_arima <- tsretail %>%
  model(ARIMA(sales)) %>%
report(fit_retail_arima)

## Series: sales
## Model: ARIMA(1,0,3)(0,1,2)[12] w/ drift
##
## Coefficients:
```

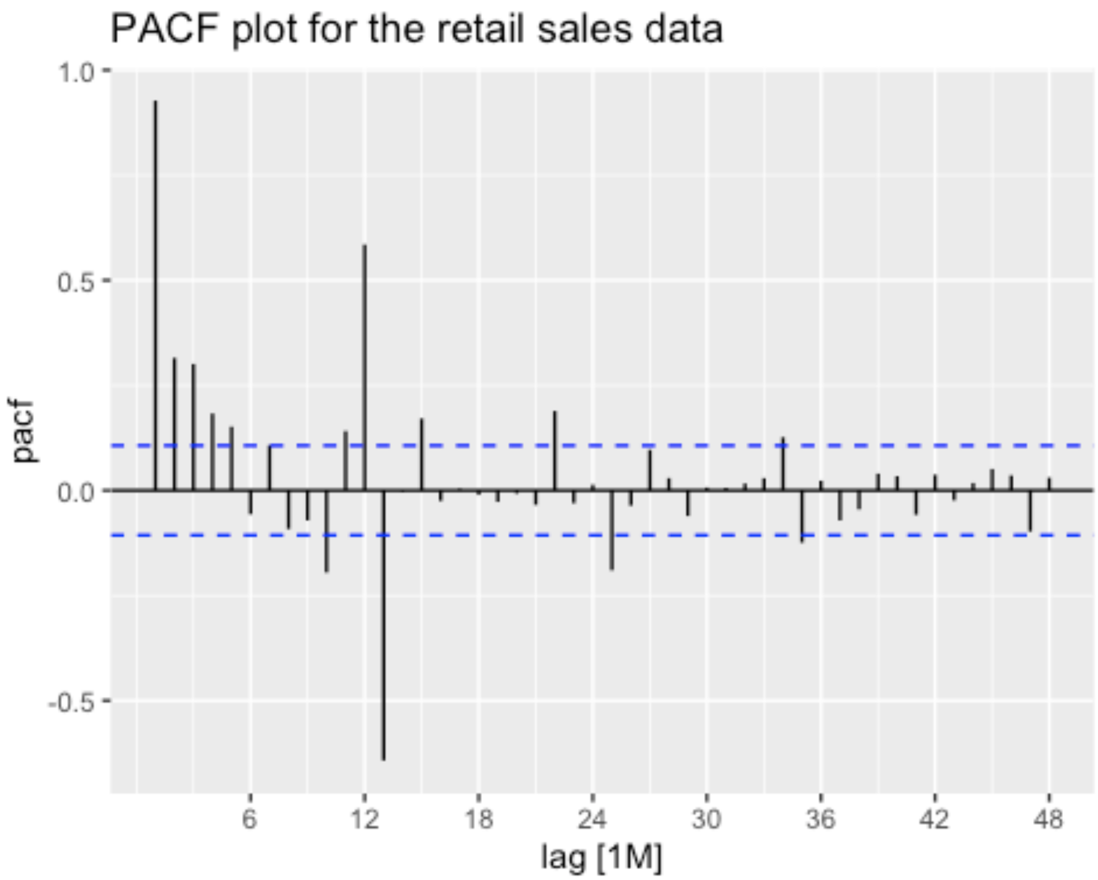
```
##          ar1      ma1     ma2     ma3     sma1     sma2  constant
##       0.9260  -0.4786  0.0103  0.1839  -0.3376  -0.2596  826.7518
## s.e.  0.0269   0.0638  0.0739  0.0608   0.0548   0.0484  103.0485
##
## sigma^2 estimated as 39034949:  log likelihood=-3311.12
## AIC=6638.23   AICc=6638.69   BIC=6668.53
```

```
fit_retail_arima %>% gg_tsresiduals()
```



Q3c

```
plotPACF <-
  tsretail %>%
  PACF(sales, lag_max = 48) %>%
  autoplot() +  ggtitle("PACF plot for the retail sales data")
plotPACF
```



kpss before differencing

```
kpss_test_before <- tsretail %>%
  features(sales, unitroot_kpss)
kpss_test_before

## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##       <dbl>       <dbl>
## 1      5.53        0.01
```

The p value is lower than the critical of 1%,5% and 10% value. Meaning the, the null hypothesis is rejected. The test statistic is much bigger than the 1% critical value, indicating that the null hypothesis is rejected. That is, the data are not stationary. We can difference the data, and apply the test again.

#unitroot before differencing

```
tsretail %>%
  mutate(retail = sales) %>%
  features(retail, unitroot_nsdiffs)
```

```
## # A tibble: 1 x 1
##   nsdiffs
##     <int>
## 1       1
```

#unitroot after differencing

```
tsretail %>%
  mutate(retail = difference((sales), 12)) %>%
  features(retail, unitroot_nsdiffs)
```

```
## # A tibble: 1 x 1
##   nsdiffs
##     <int>
## 1       0
```

#unitroot after differencing

```
tsretail %>%
  mutate(retail = difference((sales), 12)) %>%
  features(retail, unitroot_ndiffs)
```

```
## # A tibble: 1 x 1
##   ndiffs
##    <int>
## 1      0
```

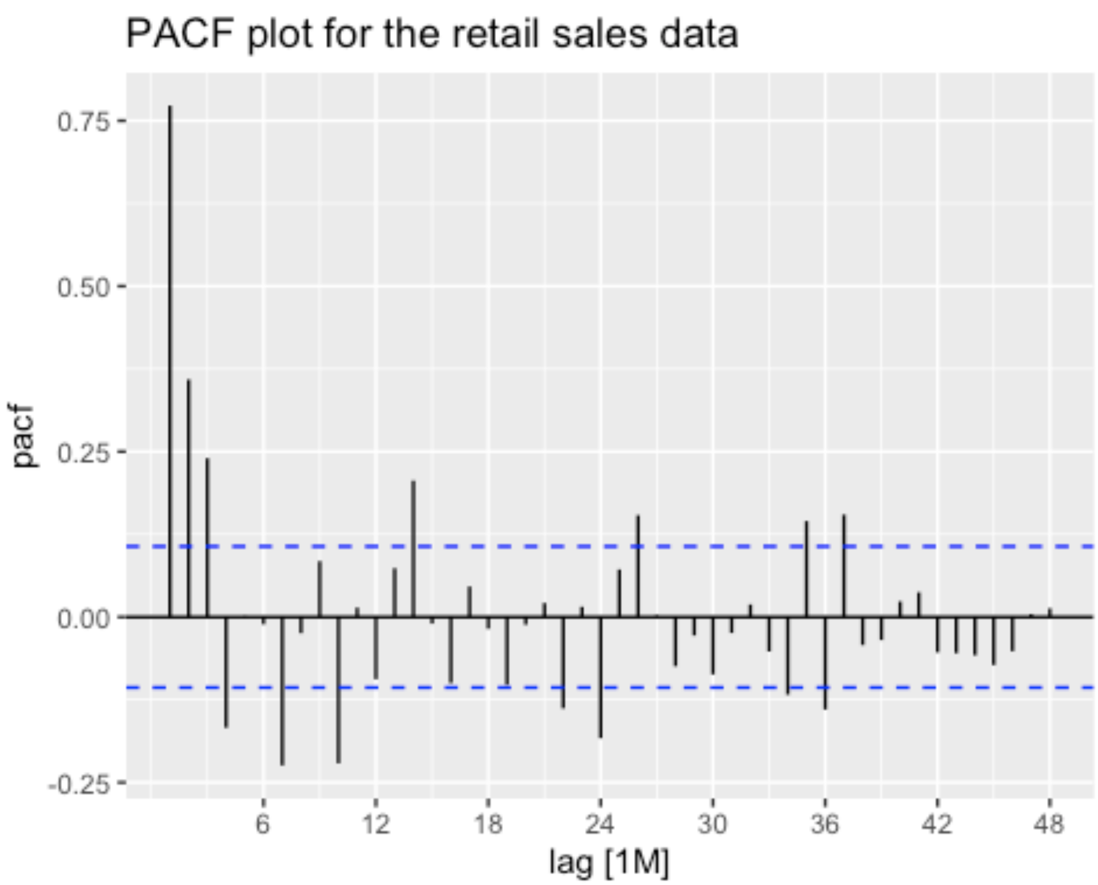
kpss after differencing

```
kpss_test<-tsretail %>%
  mutate(retail = difference((sales), 12)) %>%
  features(retail, unitroot_kpss)
kpss_test
```

```
## # A tibble: 1 x 2
##   kpss_stat kpss_pvalue
##       <dbl>       <dbl>
## 1     0.157         0.1
```

PACF after differencing

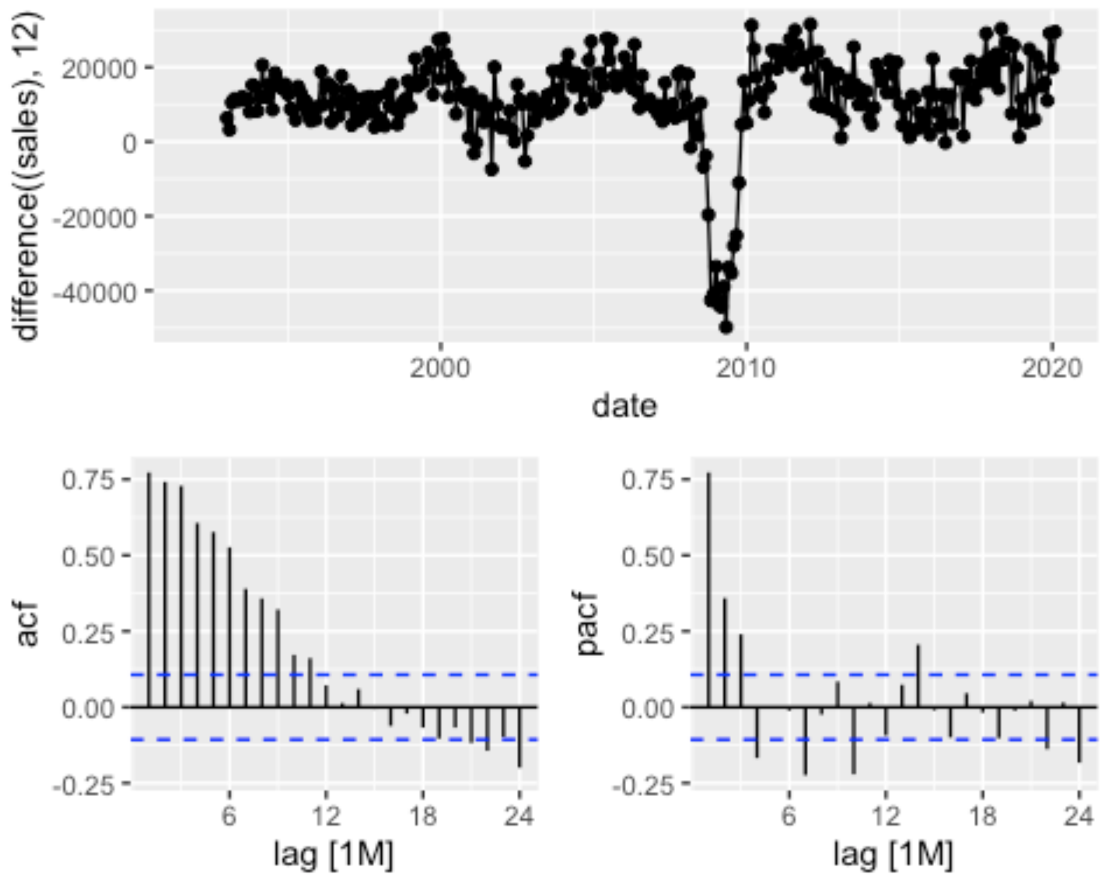
```
plotPACFafter <-
  tsretail %>%
  mutate(retail = (difference((sales), 12))) %>%
  PACF(retail, lag_max = 48) %>%
  autoplot() +  ggtitle("PACF plot for the retail sales data")
plotPACFafter
```



Q3d.

```
set.seed(333)
tsretailtrain11 <- tsretail %>% filter(date < '2011-01-01')
tsretailtest11 <- tsretail %>% filter(date >= '2011-01-01')

fit_ts11 <- tsretailtrain11 %>%
  model(TSLM(sales ~ trend() + season()))
fc_sales <- fit_ts11 %>%
  forecast(h = "10 years")

forecast::accuracy(fc_sales, tsretailtest11)

## Warning: The future dataset is incomplete, incomplete out-of-sample data
will be treated as missing.
## 10 observations are missing between 2020 Mar and 2020 Dec

## # A tibble: 1 x 9
##   .model                      .type    ME   RMSE    MAE    MPE  MAPE  MASE
ACF1
##   <chr>                       <chr> <dbl>  <dbl>  <dbl>  <dbl> <dbl> <dbl>
<dbl>
## 1 TSLM(sales ~ trend() + sea… Test   969. 14250. 10815. -0.119  2.70   NaN
0.409
```

```
tsretail %>%
  gg_tsdisplay(difference((sales), 12), plot_type='partial', lag_max = 24)

## Warning: Removed 12 rows containing missing values (geom_path).

## Warning: Removed 12 rows containing missing values (geom_point).
```




```
fit_arima1 <- tsretailtrain11 %>%
  model(arima = ARIMA(sales ~ pdq(1:3,0, 0:2) + PDQ(0:2, 1, 0:2),stepwise =
FALSE, approximation = FALSE))

fc_sales1 <-
  fit_arima1 %>%
  forecast(h = "10 years")

forecast::accuracy(fc_sales1, tsretailtest11)

## Warning: The future dataset is incomplete, incomplete out-of-sample data
will be treated as missing.
## 10 observations are missing between 2020 Mar and 2020 Dec

## # A tibble: 1 x 9
##    .model .type     ME    RMSE    MAE    MPE   MAPE   MASE   ACF1
```

```
##   <chr>  <chr>  <dbl>  <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 arima  Test  17534. 22136. 17829.  4.13  4.22   NaN 0.603
```

#Q3e

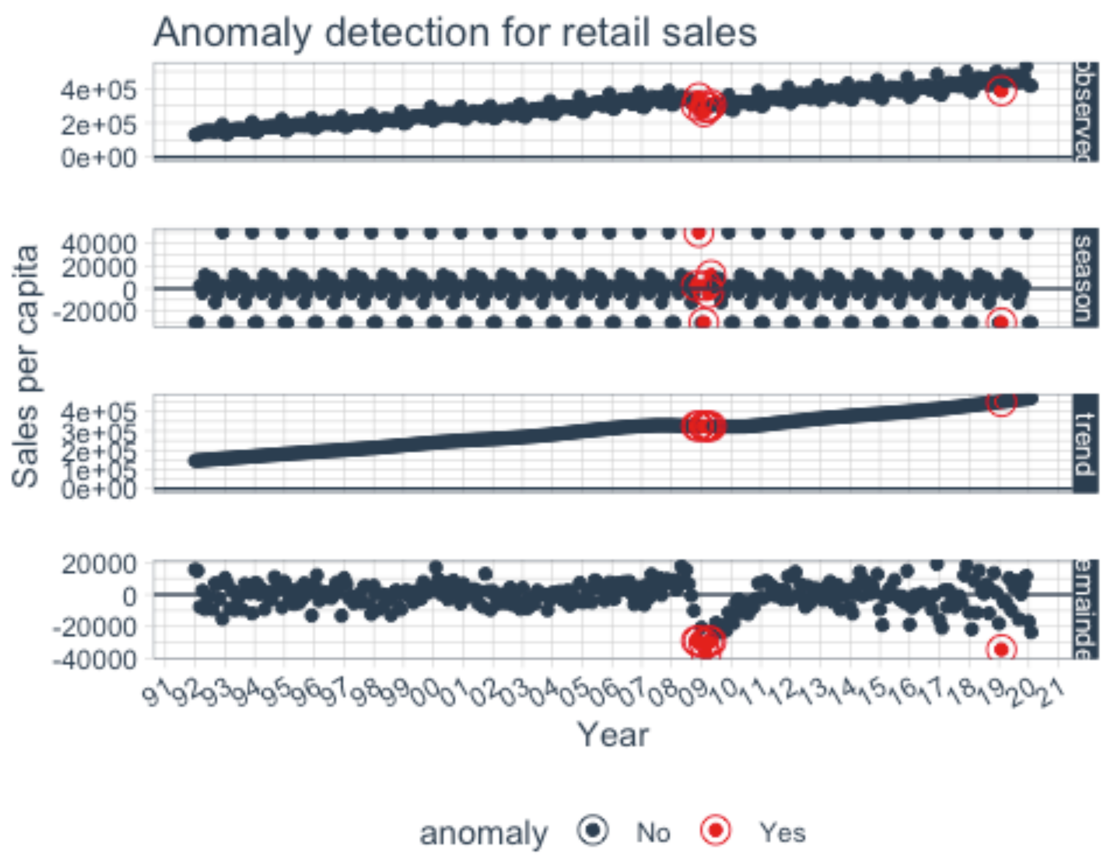
```
set.seed(333)
tsretailtrain16 <- tsretail %>% filter(date < '2016-01-01')
tsretailtest16 <- tsretail %>% filter(date >= '2016-01-01')

fit_ts16 <- tsretailtrain16 %>%
  model(TSLM(sales ~ trend() + season()))
fc_sales2 <- fit_ts16 %>%
  forecast(h = "5 years")

forecast::accuracy(fc_sales2, tsretailtest16)

## Warning: The future dataset is incomplete, incomplete out-of-sample data
will be treated as missing.
## 10 observations are missing between 2020 Mar and 2020 Dec

## # A tibble: 1 x 9
##   .model                      .type     ME   RMSE    MAE   MPE  MAPE  MASE
ACF1
##   <chr>                       <chr>  <dbl>  <dbl>  <dbl> <dbl> <dbl> <dbl>
<dbl>
## 1 TSLM(sales ~ trend() + sea… Test  11405. 18692. 14567.  2.39  3.24   NaN
0.366

fit_arima2 <- tsretailtrain16 %>%
  model(arima = ARIMA(sales ~ pdq(1:3,0, 0:2) + PDQ(0:2, 1, 0:2), stepwise =
FALSE, approximation = FALSE))

fc_sales3 <-
  fit_arima1 %>%
  forecast(tsretailtest16)

forecast::accuracy(fc_sales3, tsretailtest16)

## # A tibble: 1 x 9
##   .model .type     ME   RMSE    MAE   MPE  MAPE  MASE  ACF1
##   <chr>  <chr>  <dbl>  <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 arima  Test  75351. 76602. 75351.  17.4  17.4   NaN 0.423
```

#Q4

```
library(anomalize)

## ══ Use anomalize to improve your Forecasts by 50%! ═════════════════════════

## Business Science offers a 1-hour course - Lab #18: Time Series Anomaly
Detection!
```

```
## </> Learn more at: https://university.business-science.io/p/learning-labs-pro </>

library(tibbletime)

##
## Attaching package: 'tibbletime'

## The following object is masked from 'package:stats':
##
##     filter

library(tsibbledata)

tsretail

## # A tsibble: 338 x 2 [1M]
##        date  sales
##       <mth>  <dbl>
##  1 1992 Jan 130683
##  2 1992 Feb 131244
##  3 1992 Mar 142488
##  4 1992 Apr 147175
##  5 1992 May 152420
##  6 1992 Jun 151849
##  7 1992 Jul 152586
##  8 1992 Aug 152476
##  9 1992 Sep 148158
## 10 1992 Oct 155987
## # … with 328 more rows

tsretail_last = read_csv("retailSales.csv")

## Parsed with column specification:
## cols(
##   date = col_character(),
##   sales = col_double()
## )

tsretail_last %>%
  mutate(date = mdy(date))  %>%
  as_tbl_time(index = date) %>%
  as_period("month") %>%
  time_decompose(sales, method = "stl") %>%
  anomalize(remainder, method = "gesd") %>%
  #plot_anomalies() +
  plot_anomaly_decomposition()+
  labs(title = "Anomaly detection for retail sales") +
  xlab("Year") + ylab("Sales per capita ") +
  scale_x_date(date_breaks = "years" , date_labels = "%y")

## frequency = 12 months
```

```
## trend = 60 months
```

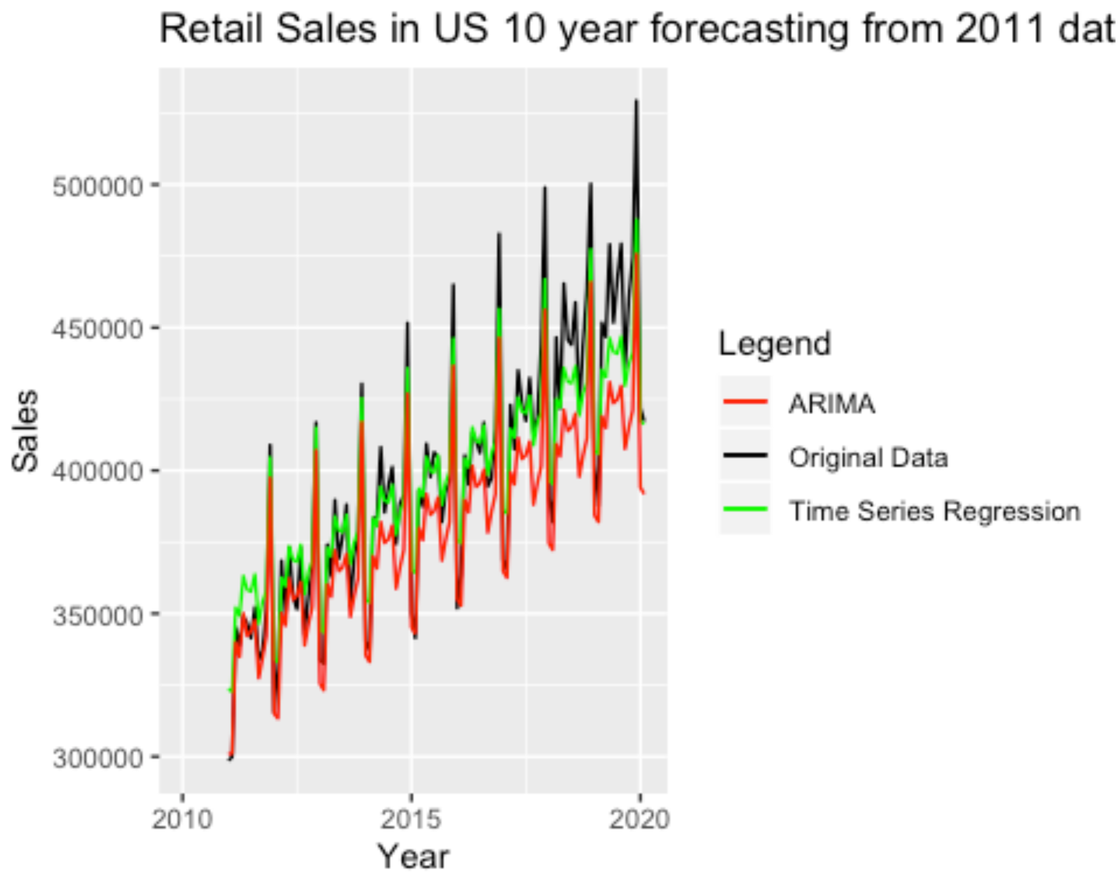



```
# fc_NYDrift <-
#   tsNY %>%
#   model(RW(loansPerCapita ~ drift())) %>%
#   forecast(h = "5 years") %>%
#   autoplot(tsNY, colour = "#769ECB") +
#   geom_line(linetype = 'dashed', colour = '#000000') +
#   xlab("Year (monthly data)") + ylab("Loans per capita") +
#   ggtitle("Number of loansPerCapita in US data")
# fc_NYDrift
```

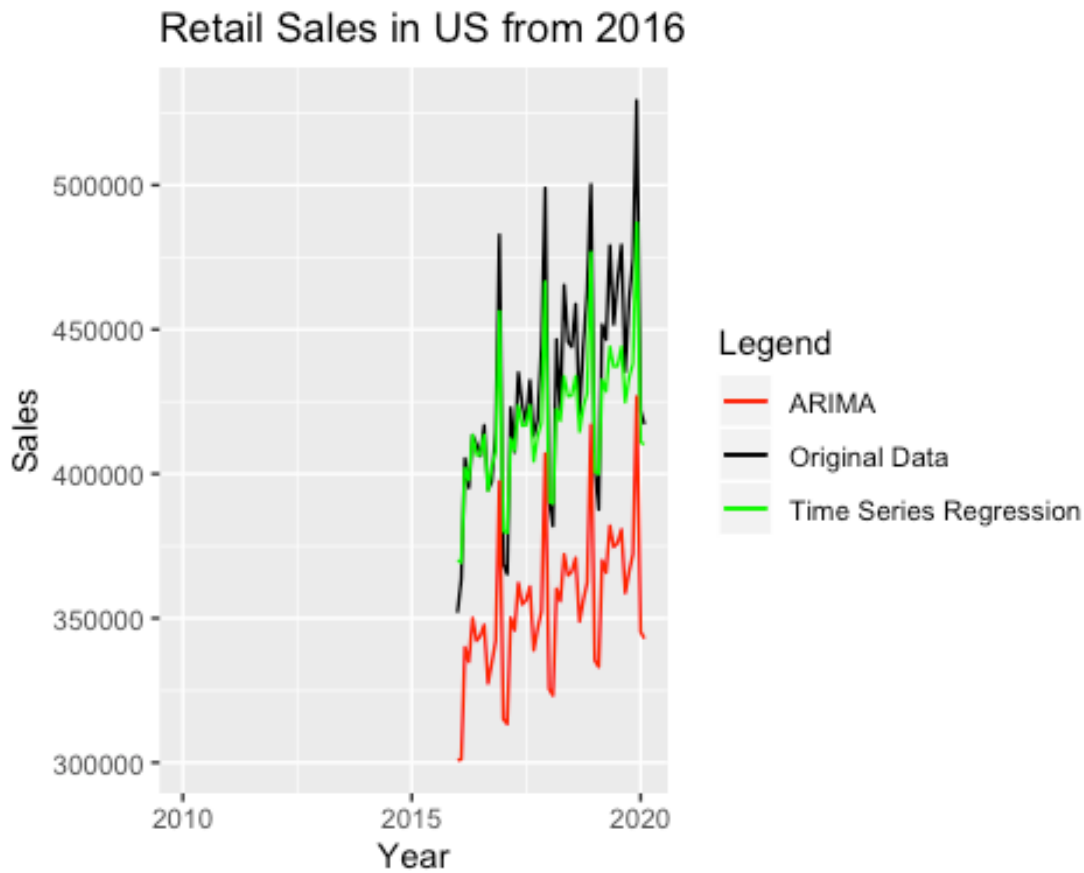
#Q4b

```
# fit_ts11 <- tsretailtrain11 %>%
#   model(TSLM(sales ~ trend() + season()))
#
# fc_sales <- fit_ts11 %>%
#   forecast(h = "10 years")
#
# fit_arima1 <- tsretailtrain11 %>%
#   model(arima = ARIMA(sales ~ pdq(1:3,0, 0:2) + PDQ(0:2, 1, 0:2),stepwise =
FALSE, approximation = FALSE))
#
```

```r
# fc_sales1 <-
#   fit_arima1 %>%
#   forecast(h = "10 years")
#
# fit_ts16 <- tsretailtrain16 %>%
#   model(TSLM(sales ~ trend() + season()))
# fc_sales2 <- fit_ts16 %>%
#   forecast(h = "5 years")
#
# fit_arima2 <- tsretailtrain16 %>%
#   model(arima = ARIMA(sales ~ pdq(1:3,0, 0:2) + PDQ(0:2, 1, 0:2), stepwise
= FALSE, approximation = FALSE))
#
# fc_sales3 <-
#   fit_arima1 %>%
#   forecast(tsretailtest16)

plot_2011 <-
  ggplot() +
  geom_line(data = tsretailtest11, aes(x = date, y = sales, color = "Original
Data")) +
  geom_line(data = fc_sales , aes(x = date, y = sales, color = "Time Series
Regression")) +
  geom_line(data = fc_sales1 , aes(x = date, y = sales, color = "ARIMA")) +
  xlim(c(as.Date('2010-01-01'),as.Date('2020-02-01')))+
  ggtitle("Retail Sales in US 10 year forecasting from 2011 data") +
  labs(x='Year',y='Sales', color="Legend")+
  scale_color_manual(values = c("red","black","green"))

plot_2011

## Warning: Removed 20 rows containing missing values (geom_path).

## Warning: Removed 20 rows containing missing values (geom_path).
```




```r
plot_2016 <-
  ggplot() +
  geom_line(data = tsretailtest16, aes(x = date, y = sales, color = "Original
Data")) +
  geom_line(data = fc_sales2 , aes(x = date, y = sales, color = "Time Series
Regression")) +
  geom_line(data = fc_sales3 , aes(x = date, y = sales, color = "ARIMA")) +
  xlim(c(as.Date('2010-01-01'),as.Date('2020-02-01')))+
  ggtitle("Retail Sales in US from 2016") +
  labs(x='Year',y='Sales', color="Legend")+
  scale_color_manual(values = c("red","black","green"))

plot_2016

## Warning: Removed 20 rows containing missing values (geom_path).
```



Retail Sales in US from 2016
Sales
500000
450000
400000
350000
300000
2010
2015
2020
Year
Legend
ARIMA
Original Data
Time Series Regression