AAEC 4984/5484 Applied Economic Forecasting

EXAM 1 – Spring 2022

MASTER KEY

Due: Thursday, March 24, 2022 (6:59 AM)

Instructions (PLEASE READ):

1. This is an open-book exam. You are free to use your notes, homework, and any available resource to aid with your answers. **However, you are not allowed to collaborate with anyone else.**
2. Where necessary, please ensure that your graphs and visuals have proper titles and axes labels.
3. Please refer to the output, whenever appropriate, when discussing the results.

Question 1: Time Series Regressions [Points: 30]

The EIA reports data on **Monthly** U.S. Product Supplied of Crude Oil and Petroleum Products (Thousand Barrels) from January 1936 – December 2021. The data are available at http://www.eia.gov/dnav/pet/pet_cons_psup_dc_nus_mbbl_m.htm.

Our task here to use a time series regression to forecast the data for the last 2 years of data and perform model diagnostic tests.

a. Using the codes below, pull the data into

```r
#store file in temp file and read into R
tmp <- tempfile(fileext = ".xls")
download.file(url = "https://www.eia.gov/dnav/pet/xls/PET_CONS_PSUP_DC_NUS_MBBL_M.xls",
              destfile = tmp, mode = "wb")
oil.raw <- readxl::read_excel(tmp, sheet = 2, skip = 2)
```

b. Unmute and modify the code chunk below to

- since January 1981 is the first time a data is observed drop all rows between 1:541. *I have already accounted for this in the code below.*
- keep only the second column of `oil.raw` ("U.S. Product Supplied of Crude Oil and Petroleum Products (Thousand Barrels)"),
- convert the series to "Million Barrels" by diving by 1000
- now declare as `ts()` object with a start date of January 1981.

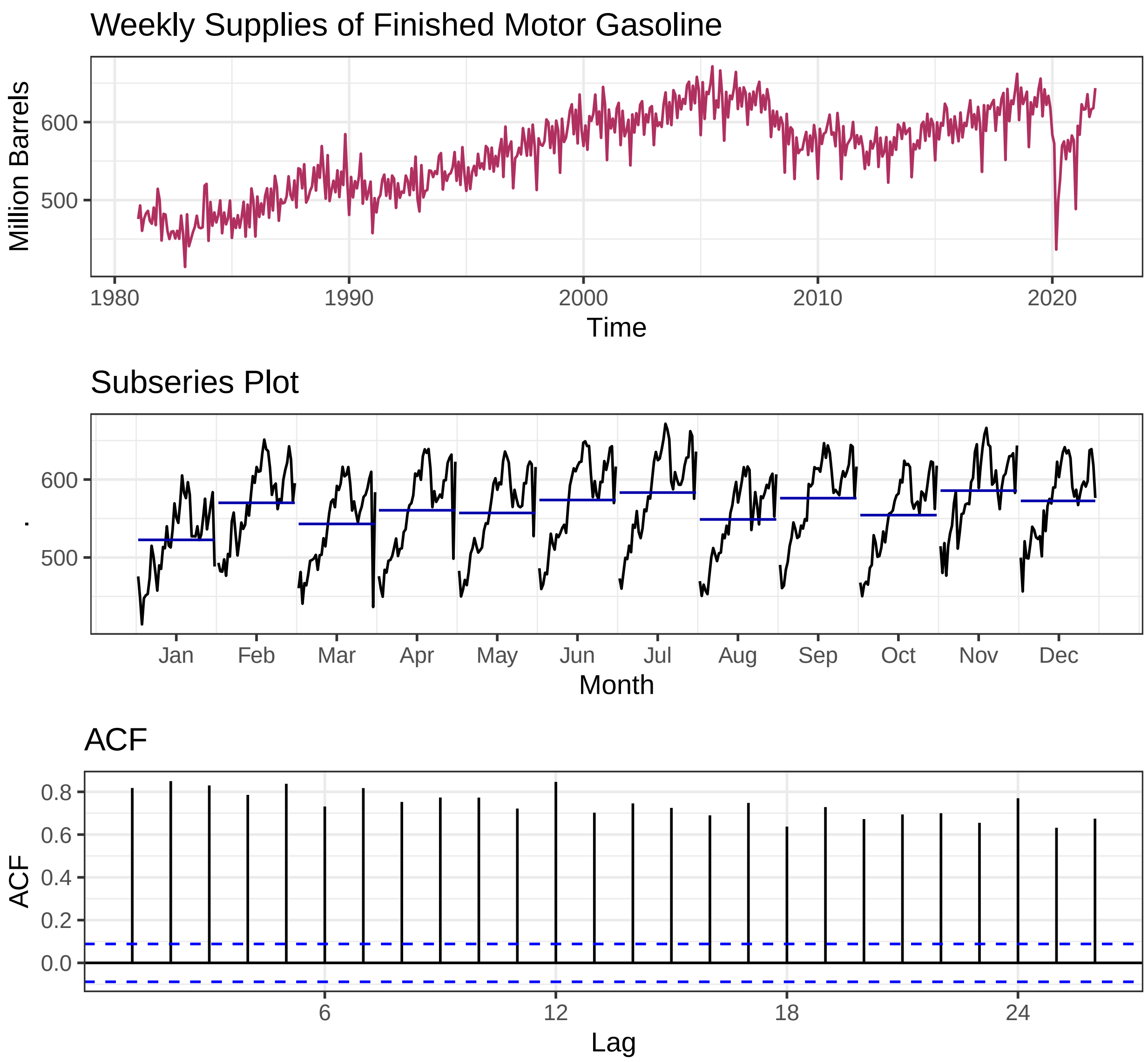
Your task is ultimately to store this into a variable called `oil`.

```r
#Converting oil.raw to a ts object
# ___ <- (__[-c(1:541),__]/____) %>% ____(frequency = ____, start = 1981)

oil <- (oil.raw[-c(1:541),2]/1000) %>% ts(frequency = 12, start = 1981)
```

c. Report the `autoplot`, `ggsubseriesplot`, and `ggAcf` plots of `oil`. **Briefly comment on the plots. Be sure to talk about any visible trends and seasonality. Be explicit in explaining how you came to your conclusions.**

Be sure to add appropriate labels to your plot.

```r
p1 <- oil %>% autoplot(colour = "maroon") +
  labs(title = "Weekly Supplies of Finished Motor Gasoline",
       y = "Million Barrels")
p2 <- oil %>% ggsubseriesplot() + ggtitle("Subseries Plot")
p3 <- oil %>% ggAcf() + ggtitle("ACF")

gridExtra::grid.arrange(p1,p2,p3, ncol = 1)
```



Brief Comments:

- **The autoplot reveals a general upward trend in the data. We observe that February 2020, there was a huge fall off in supplies. It has since increased. There are also fluctuations in the data that points to seasonality.**
- **The trend and seasonality are confirmed by the ggACF. The first lag is pretty large and significant. The lags are also decaying slowly. The seasonality is evident since there are large spikes at the multiple of the data's frequency.**
- **The ggsubseriesplot reveals some variablity across the months**.

d. Because 2020 was such an unusual period, it would not be "fair" to include this in the test sample. Use the `window` command to end the data at December 2019 instead. Store this as `oil2`.

```
oil2 <- oil %>% window(end = c(2019,12))
```

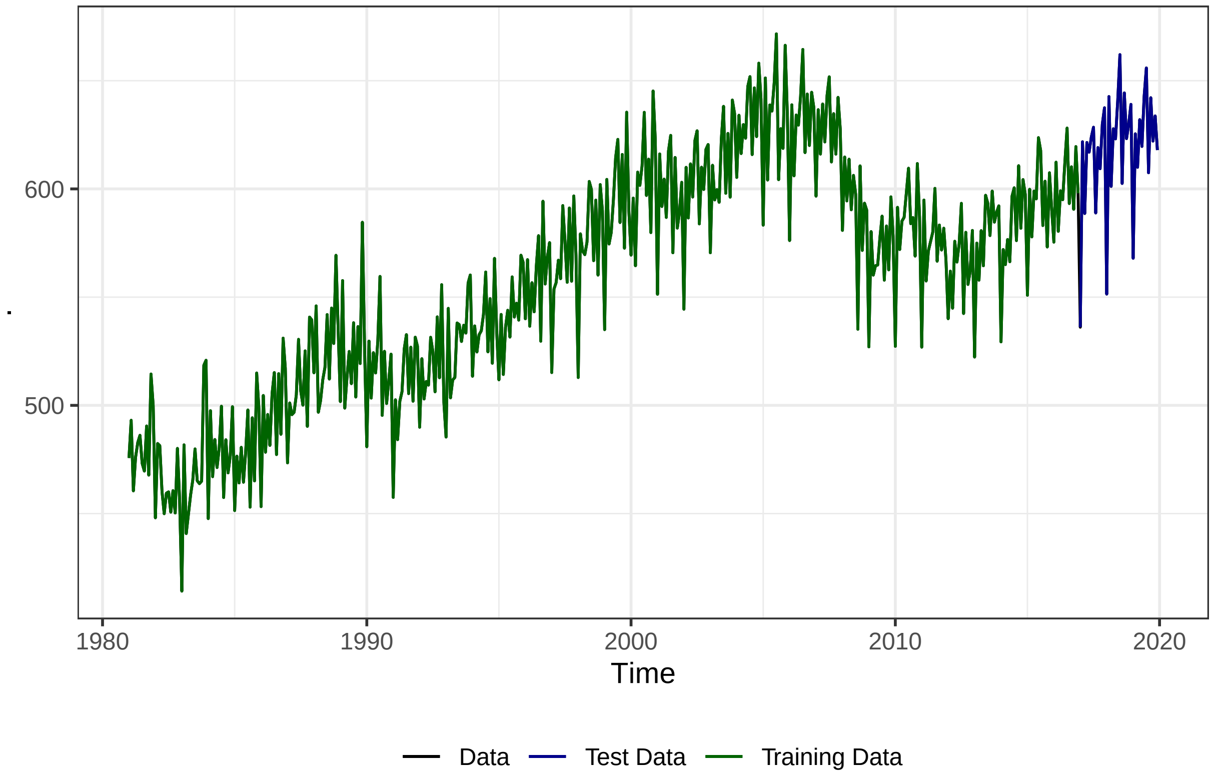
e. Split `oil2` into a training and test set.

- Using the window command, assign data up to (and including) December 2016 to the training set. This period should be January 1981 - December 2016. Call this `oil.train`.
- Assign data from January 2017 to December 2019 to the test set. Call this `oil.test`.

```
oil.train <- oil2 %>% window(end = c(2016,12))
oil.test <- oil2 %>% window(start = c(2017,1))
```

f. Confirm that your data in `oil2` is properly split by producing a `autoplot` (with `autolayers`). Be sure to include `oil2`, `oil.test`, and `oil.train`.

```
oil2 %>% autoplot(series = "Data") +
  autolayer(oil.train, series = "Training Data") +
  autolayer(oil.test, series = "Test Data") +
  scale_color_manual(values = c("black", "darkblue", "darkgreen"),
                     name = "")
```



g. Run a regression of the training data, `oil.train`, on a linear trend and monthly/seasonal dummies and report your regression summary. **Store your regression results as `reg1` before you produce the summary.**

```
reg1 <- tslm(oil.train ~ trend + season)
reg1 %>% summary()
```

```
##
## Call:
## tslm(formula = oil.train ~ trend + season)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -71.889 -24.492  -4.338  25.572  69.318
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 449.77300    5.61476  80.105  < 2e-16 ***
## trend         0.33009    0.01168  28.256  < 2e-16 ***
## season2      44.66088    7.13438   6.260 9.54e-10 ***
## season3      20.07996    7.13441   2.815  0.00512 **
## season4      34.60471    7.13446   4.850 1.74e-06 ***
## season5      30.32532    7.13452   4.251 2.63e-05 ***
## season6      46.40559    7.13461   6.504 2.23e-10 ***
## season7      55.18131    7.13472   7.734 7.82e-14 ***
## season8      21.17025    7.13484   2.967  0.00318 **
## season9      47.96877    7.13498   6.723 5.84e-11 ***
## season10     24.64143    7.13515   3.454  0.00061 ***
## season11     57.72949    7.13533   8.091 6.48e-15 ***
## season12     44.53609    7.13553   6.241 1.06e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30.27 on 419 degrees of freedom
## Multiple R-squared:   0.69,  Adjusted R-squared:  0.6811
## F-statistic: 77.71 on 12 and 419 DF,  p-value: < 2.2e-16
```

h. Carefully interpret the coefficients on the `intercept`, `trend`, and `season10` variables. Interpret the R^2 value.

Interpretation:

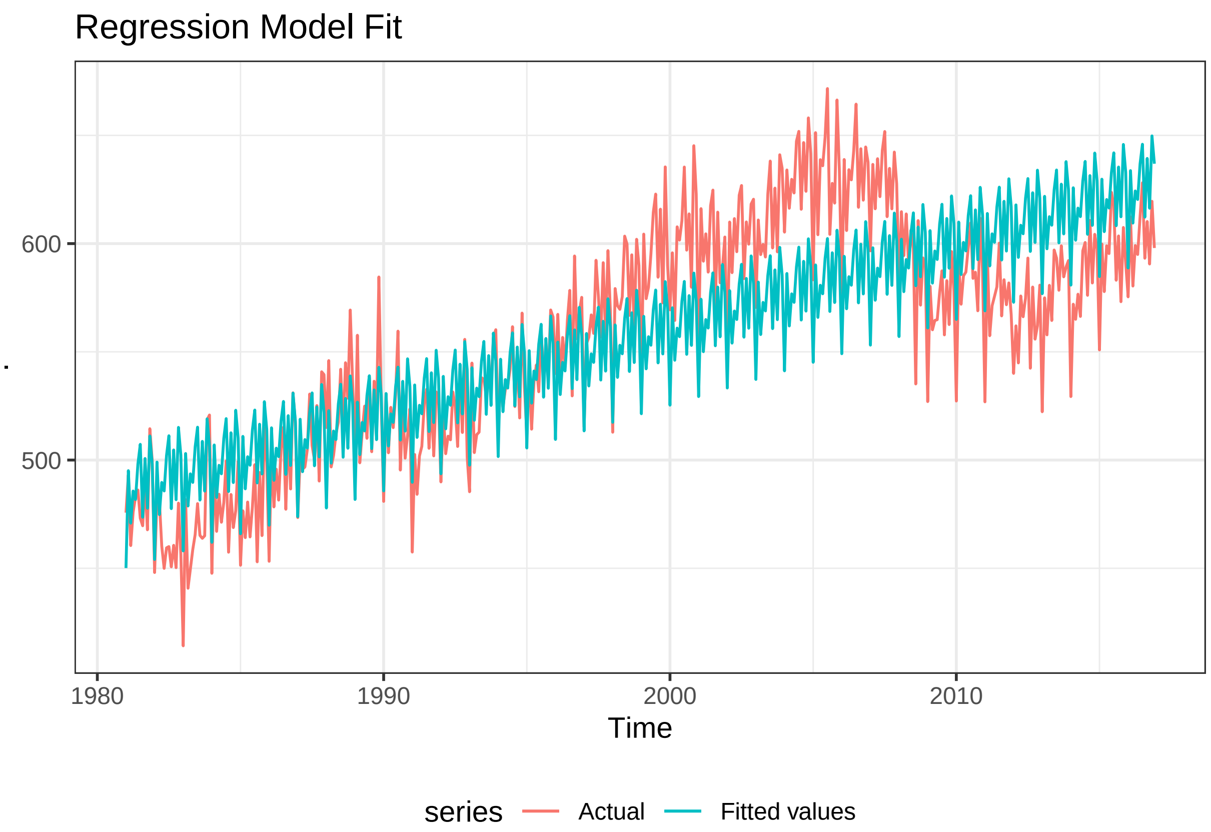
- **The coefficient on the intercept suggests that, the average supplies of crude and petroleum products in January is 449.773 Million barrels, holding all other things constant.**
- **The coefficient on the trend suggests that the supplies of crude and petroleum products increases by approximately 0.33 Million Barrels per month, holding all other things constant.**
- **The coefficient on the `season10` suggests that, on average, the supplies of crude and petroleum products is 24.641 Million Barrels higher in October than in January, holding all other things constant.**
- **The R^2 value here indicates that almost 69%of the variation in the supplies of crude and petroleum products can be explained by a linear trend and seasonal dummies.**

i. Explain why the above model does not include a dummy variable for quarter 1.

Solution: The *January* dummy is omitted to avoid the dummy variable trap. Instead, β_0 (the intercept) will be used to represent the average supplies for January

j. Produce a `autoplot` of the actual values (`oil.train`) and `autolayer` the fitted values from `reg1`. **Comment on the model fit.**

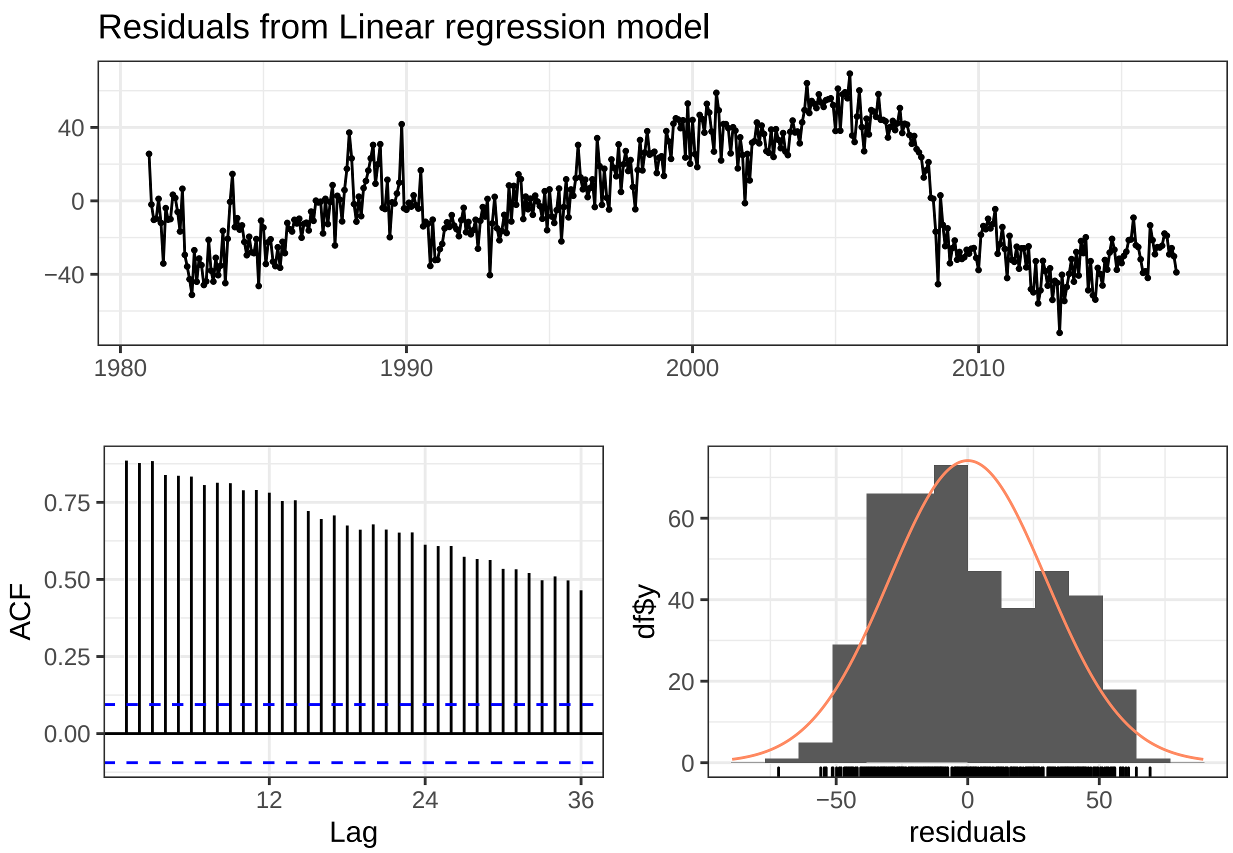
```
oil.train %>% autoplot(series = "Actual") +
  autolayer(fitted(reg1), series = "Fitted values" ) +
  ggtitle("Regression Model Fit")
```



Comment: While the linear trend and seasonal dummies regression does a fairly good job of predicting the series, there are periods where we are systematically over- (and under-) predicting the actual data.

k. Conduct a residual check of `reg1`. Comment on whether the residuals are white noise and if there is any possible serial correlation. **I would like you to be detailed in your comments but you must explain how you arrive at your conclusions as plainly as possible.**

```
reg1 %>% checkresiduals()
```



```
##
##  Breusch-Godfrey test for serial correlation of order up to 24
##
## data:  Residuals from Linear regression model
## LM test = 374.14, df = 24, p-value < 2.2e-16
```

Comment: The diagnostic check suggest that the residuals are not white noise. In fact, the ACF reveals that a trend might still be present in the residuals. This is evident from the fact that the residuals are slowly decaying and the first lag is so significant. The Breusch-Godfrey test strongly rejects the null of no serial correlation up to lag 24. In sum, the residuals do not appear to be WN and are also serially correlated.

l. Franco contends that there should potentially be a quadratic trend in the model. You are skeptical about this but opt to include one anyways. After all, if he's wrong, he will owe you a meal at Chipotle.

Re-estimate the regression in g. but including the squared trend as well. Remember to report your regression summary.

- **Store your regression results as `reg2` before you produce the summary.**
- Recall that you can produce the squared trend term within the `tslm` function using I(trend^2)

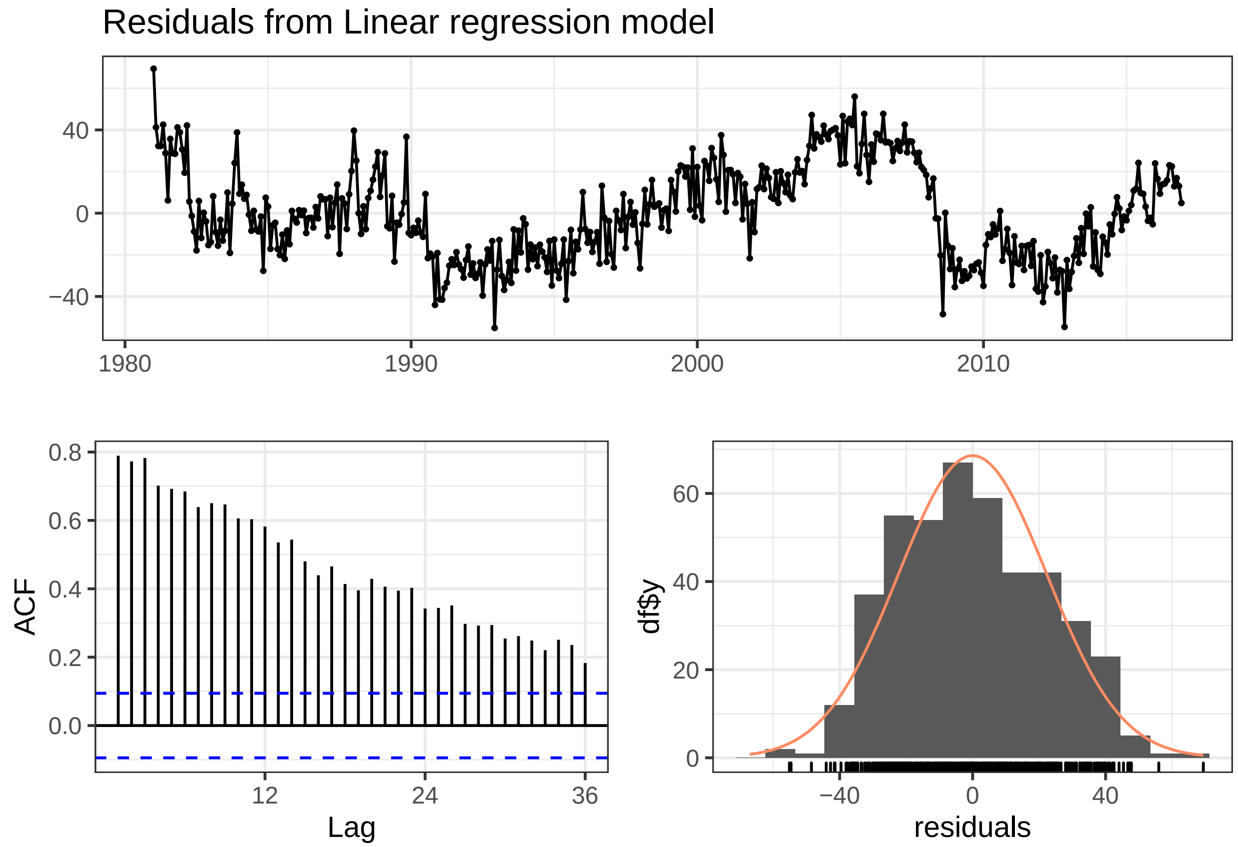
```
reg2 <- tslm(oil.train ~ trend + season + I(trend^2))
reg2 %>% summary()
```

```
##
## Call:
## tslm(formula = oil.train ~ trend + season + I(trend^2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -55.109 -17.333  -2.189  16.522  69.409
##
## Coefficients:
```

```
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.053e+02  4.876e+00  83.128  < 2e-16 ***
## trend        9.448e-01  3.511e-02  26.908  < 2e-16 ***
## season2      4.465e+01  5.351e+00   8.344 1.06e-15 ***
## season3      2.005e+01  5.351e+00   3.748 0.000203 ***
## season4      3.457e+01  5.351e+00   6.461 2.91e-10 ***
## season5      3.029e+01  5.351e+00   5.660 2.82e-08 ***
## season6      4.636e+01  5.351e+00   8.664  < 2e-16 ***
## season7      5.514e+01  5.351e+00  10.304  < 2e-16 ***
## season8      2.113e+01  5.351e+00   3.949 9.22e-05 ***
## season9      4.793e+01  5.351e+00   8.957  < 2e-16 ***
## season10     2.462e+01  5.352e+00   4.600 5.61e-06 ***
## season11     5.772e+01  5.352e+00  10.785  < 2e-16 ***
## season12     4.454e+01  5.352e+00   8.322 1.24e-15 ***
## I(trend^2)  -1.420e-03  7.852e-05 -18.079  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 22.7 on 418 degrees of freedom
## Multiple R-squared:  0.826,  Adjusted R-squared:  0.8206
## F-statistic: 152.7 on 13 and 418 DF,  p-value: < 2.2e-16
```

m. Conduct a residual check of Franco's model stored in `reg2`. Comment on whether the residuals now pass the Breusch-Godfrey test for serial correlation. **Be sure to state to state the null and what exactly you are concluding**

```
reg2 %>% checkresiduals()
```



```
##
##   Breusch-Godfrey test for serial correlation of order up to 24
##
```

```
## data:  Residuals from Linear regression model
## LM test = 321.6, df = 24, p-value < 2.2e-16
```

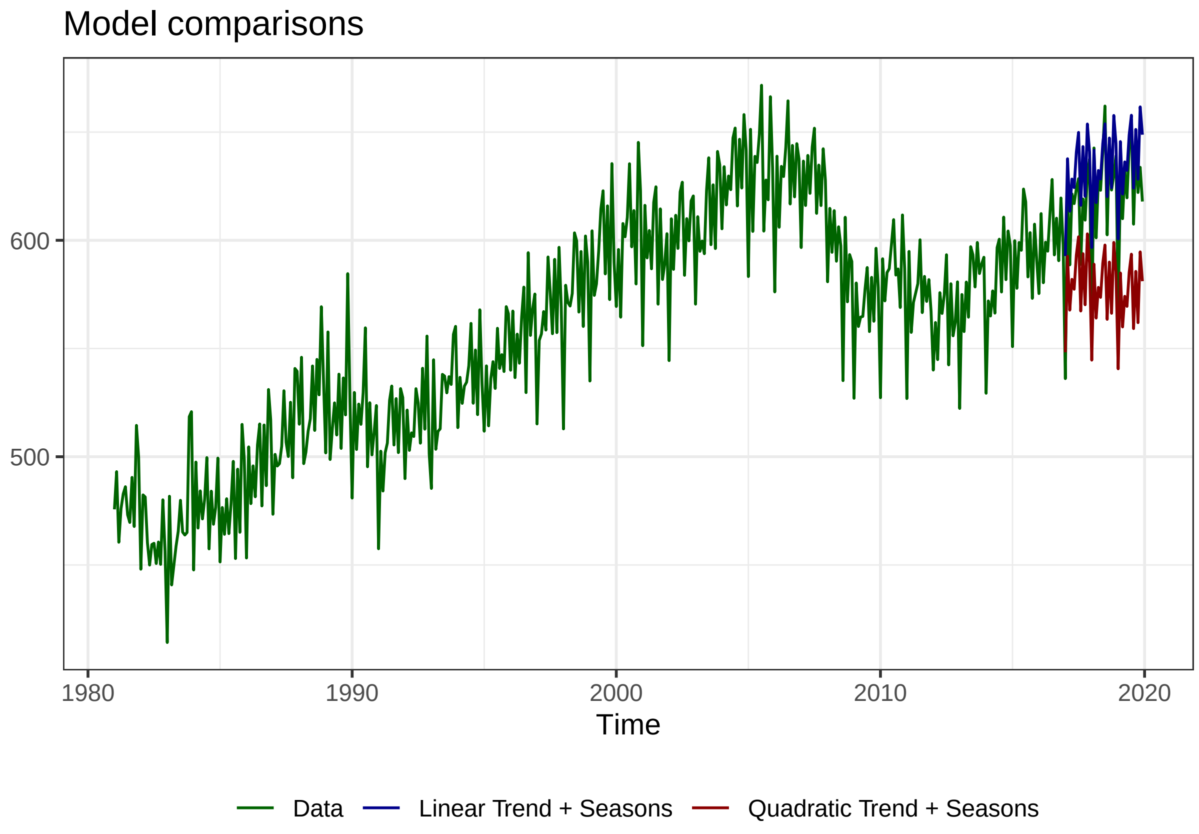
Comment: Franco's model still fails the BG test. In fact, the p-value is extremely small so we reject the null of no serial correlation and conclude that there is indeed serial correlation in the model. Also, you can see that the ACF still has some time series structure so it is not WN.

n. Let us put both models to the test. Using your models stored in `reg1` and `reg2` to forecast the next 3 years of data (h = 3*12). Store the forecast into `fore.reg1` and `fore.reg2`, respectively. **In this step, I only require that you store the forecast results.**

```
H <- 3*12

fore.reg1 <- reg1 %>% forecast(h = H)
fore.reg2 <- reg2 %>% forecast(h = H)
```

o. `cbind` the `oil2` data and the point forecast (the mean columns) stored in `fore.reg1` and `fore.reg2`. Produce a `autoplot` of each of these three variables. Be sure to give each series appropriate names in your plot.

```
cbind("Data" = oil2,
      "Linear Trend + Seasons" = fore.reg1$mean,
      "Quadratic Trend + Seasons" = fore.reg2$mean) %>%
  autoplot() + ggtitle("Model comparisons") +
  scale_color_manual(values = c("darkgreen", "darkblue", "darkred"),
                     name = "")
```



p. You might notice by now that though Franco's model might have had a higher R^2 it isn't necessarily the case that his model will be the best out of sample.

Use the `accuracy` command to formally test your forecasts in `fore.reg1` and `fore.reg2` against the test data, `oil.test`.

- Extract the `RMSE`, `MAPE`, and `MAE` columns.

- Manually compute the MSE for each model. Recall that the $MSE = RMSE^2$
- Using appropriate column and row names, report the results of the 4 model selection criteria above using `knitr::kable()` with `digits = 3`.
- **Explain** which is the preferred model under each test.

```
tab <- matrix(NA, nrow = 2, ncol = 4,
              dimnames = list(c("Linear Model", "Franco's Model"),
                              c("RMSE", "MAPE","MAE", "MSE"))
              )
tab[1,1:3] <- accuracy(fore.reg1, oil.test)[2, c("RMSE", "MAPE","MAE")]
tab[2,1:3] <- accuracy(fore.reg2, oil.test)[2, c("RMSE", "MAPE","MAE")]

tab[1,4] <- tab[1,1]^2
tab[2,4] <- tab[2,1]^2

knitr::kable(tab, align = "c", digits = 3,
             caption = "Model Comparisons")
```

Table 1: Model Comparisons

	RMSE	MAPE	MAE	MSE
Linear Model	20.036	2.583	15.508	401.428
Franco's Model	43.907	6.635	41.488	1927.861

Comments: From the table above, we see that the simple linear model with dummies does a better job than Franco's model out of sample. It looks like Franco owes you dinner after all.

[GRADS ONLY!!!!] Question 2: Basic Forecasting Models [Points: 30]

In general, this is a free response question that tests your ability to appropriately present time series results and perform model comparisons. I would like for you to write a **brief** but thorough report that addressed the questions and tasks below. We are focused on using the `ibmclose` series from the `fpp2` package. **Remember you can use the `help` function in your *console* to better understand the `ibmclose` series.**

Tasks:

- Produce an `autoplot` and `ggAcf` of the `ibmclose`. Your aim is to get familiar with the data as well as discuss the properties observed in both graphs.
  - If it helps, feel free to set your `lag.max` to a large number, say 50?
- Using my codes below split the data into a training set of the first 300 observations and a test set of 69 observations.

```
# xxxx %>% subset(start = 1, end = 300)
# xxxx %>% subset(start = 301)
```

- Using various benchmark methods (i.e. mean, naïve, drift, where appropriate) forecast the training set (the next 69 observations) and compare the results on the test set. Which method did best?
- Check the residuals of your preferred method. Do they resemble white noise?