Met4 Fall 19 home exam: Solution proposal

Assignment 1

Start by reading in the dataset, and storing it as a dataframe:

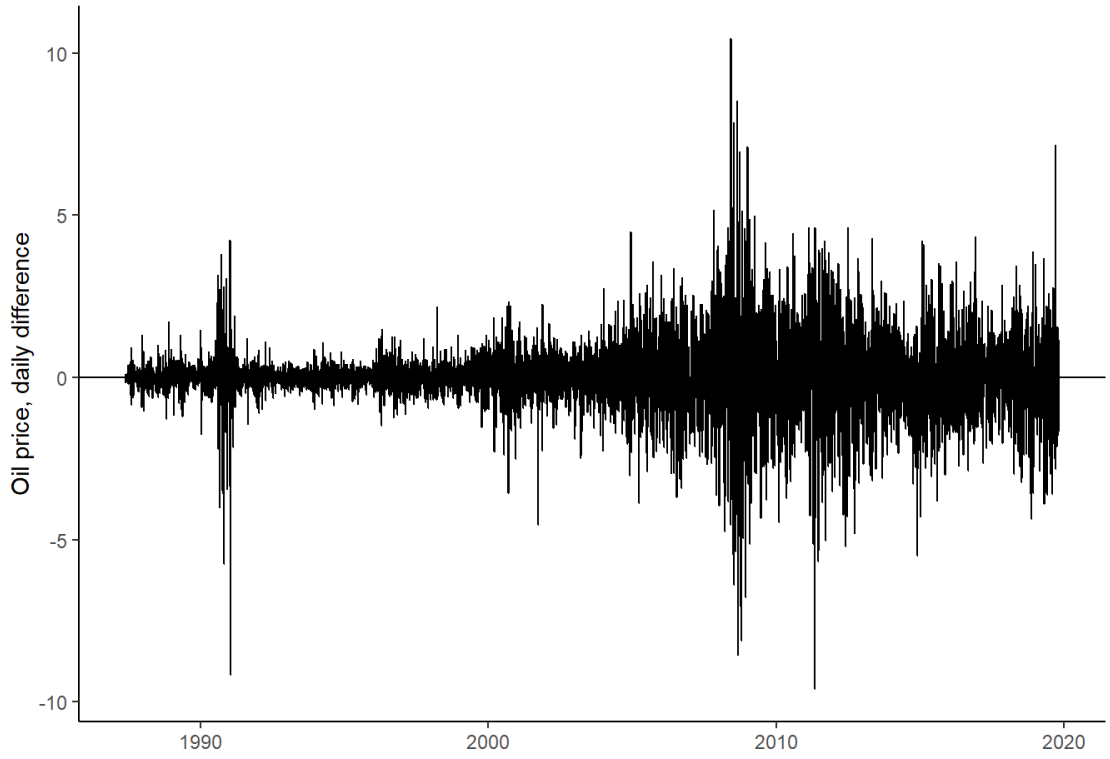
```
oil <-
  read_excel("RBRTEd.xls", sheet = "Data 1", skip = 2) %>%
  transmute(
    date = as.Date(Date),
    oil = `Europe Brent Spot Price FOB (Dollars per Barrel)`) %>%
  mutate(
    doil = oil - lag(oil, order_by = date),
    roil = (oil - lag(oil, order_by = date))/lag(oil, order_by = date))
```

Visualising the series:

```
oil %>%
  ggplot(aes(x=date, y = oil))+
  geom_line()+
  ylab("Oil price")+
  xlab("")+
  theme_classic()
```

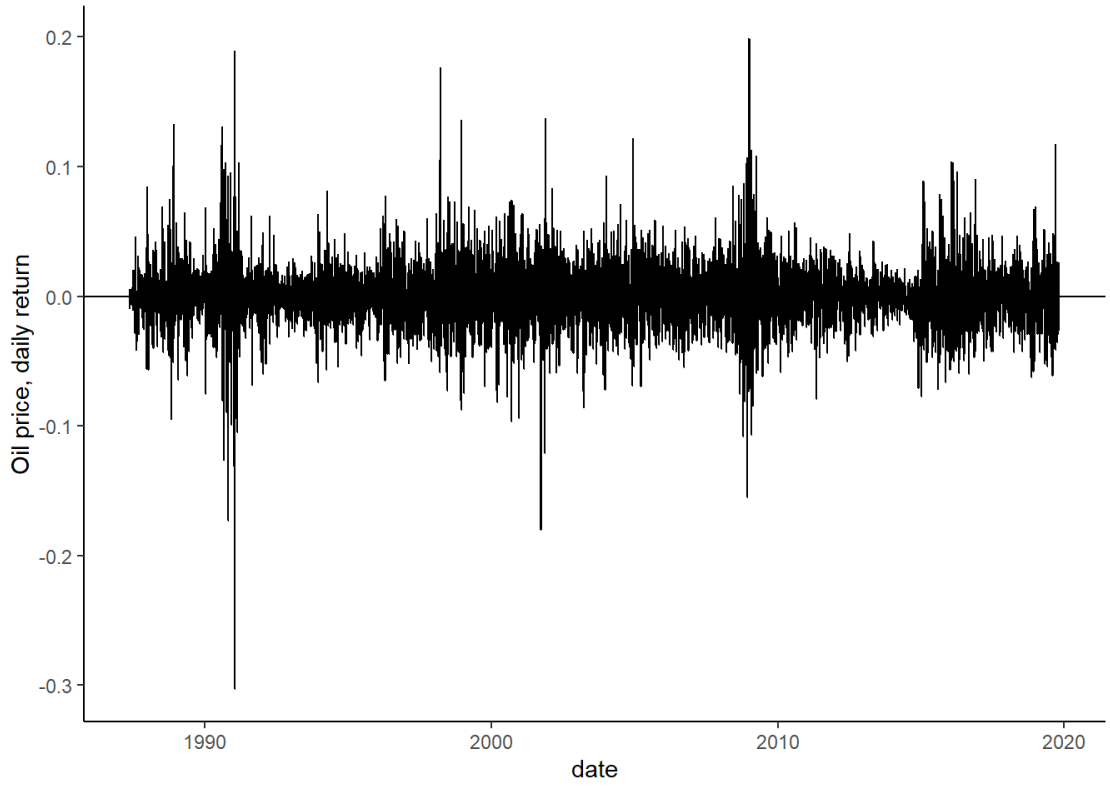

```
oil %>%
  ggplot(aes(x=date, y = doil))+
  geom_line()+
  geom_hline(yintercept = 0)+
  ylab("Oil price, daily difference")+
  xlab("")+
  theme_classic()
```

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




```
oil %>%
  ggplot(aes(x=date, y = roil))+
  geom_line()+
  geom_hline(yintercept = 0)+
  ylab("Oil price, daily return")+
  theme_classic()
```

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



Summary statistics:

```
stargazer::stargazer(as.data.frame(oil), type="html")
```

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
oil	8,236	46.365	32.671	9.100	18.730	67.297	143.950
doil	8,235	0.005	1.076	-9.620	-0.370	0.400	10.450
roil	8,235	0.0004	0.023	-0.303	-0.011	0.012	0.199

There are several things the students can focus on in the first assignment. For the assessment of exams, an important issue is that i: The students capture relevant features of the time series, and ii: The topics they discuss in the solution of assignment 1 should be revisitied in lates assignments. An example is e.g. that the oil price has developed through something that might be called “regimes”, where regimes are different both in terms of the level/growth rate of the oil price as well as differing variance.

Assignment 2

From the estimated ARIMA(1,1,1)-model we see that neither the AR(1)-coefficient nor the MA(1) coeffient are significant. From the figure where we plot data, forecasted values as well as the 80 and 95% prediction bands, we can observe that the model almost only predicts a “no-change” in the oil price, with a growing uncertainty.

Further, there is an observation (Sept. 16th) with a somewhat extreme observation. It turns out this particular date was the same date a drone attacked a Saudi oil facility, causing a large spike in oil prices.

Estimating an ARIMA-model has been covered in class, however, not all the plotting below and e.g. joining datasets on dates. Students should be rewarded for creating informative figures, showing an understanding of the arima-model and interpretation of estimated parameters.

Students should also note the very large uncertainty when predicting more than a few days ahead.

```
ts_r <- ts(oil$oil)
before_2019_08_30 <- oil$date<=as.Date('2019-08-30')
arima_111 <- Arima(ts_r[before_2019_08_30], order = c(1,1,1))


# A
summary(arima_111)
```

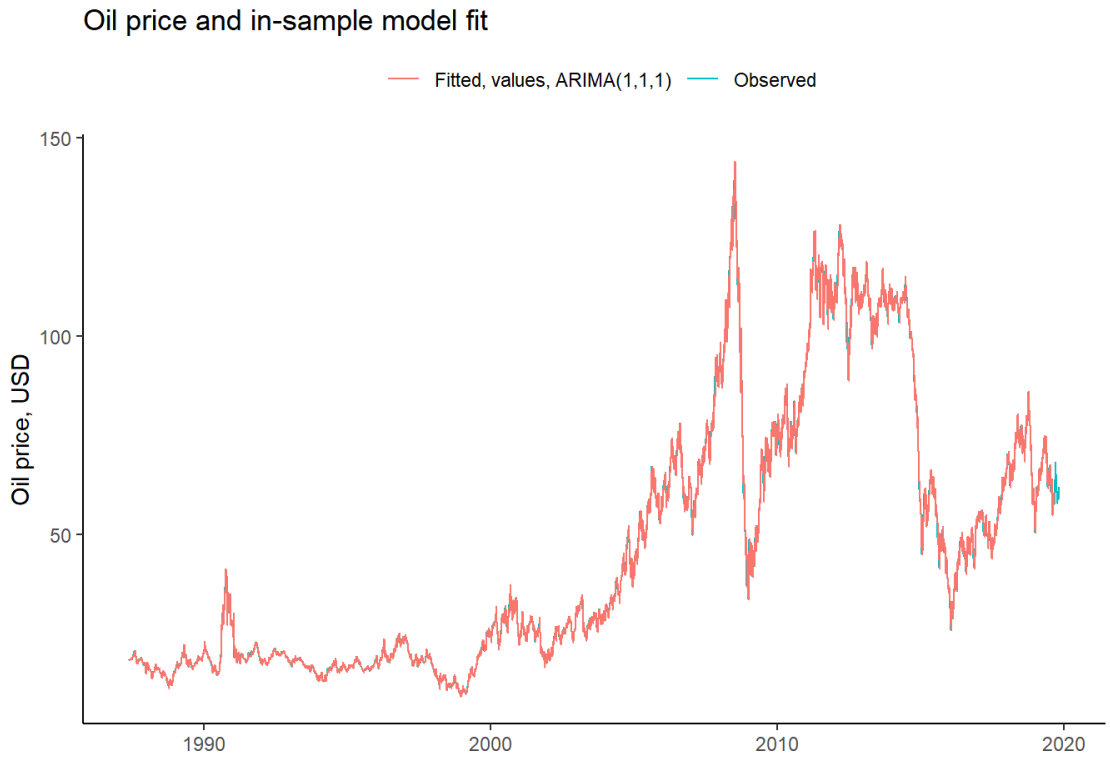
```
## Series: ts_r[before_2019_08_30]
## ARIMA(1,1,1)
##
## Coefficients:
##           ar1     ma1
##       -0.0856  0.1166
## s.e.   0.2911  0.2904
##
## sigma^2 estimated as 1.148:  log likelihood=-12190.76
## AIC=24387.52   AICc=24387.53   BIC=24408.56
##
## Training set error measures:
##                       ME     RMSE       MAE         MPE     MAPE      MASE
## Training set 0.005035162 1.071187 0.6796017 -0.01086752 1.580075 0.9999258
##                       ACF1
## Training set -0.0001869755
```

```
lmtest::coeftest(arima_111)
```

```
##
## z test of coefficients:
##
##     Estimate Std. Error z value Pr(>|z|)
## ar1 -0.08562    0.29108 -0.2941   0.7686
## ma1  0.11664    0.29035  0.4017   0.6879
```

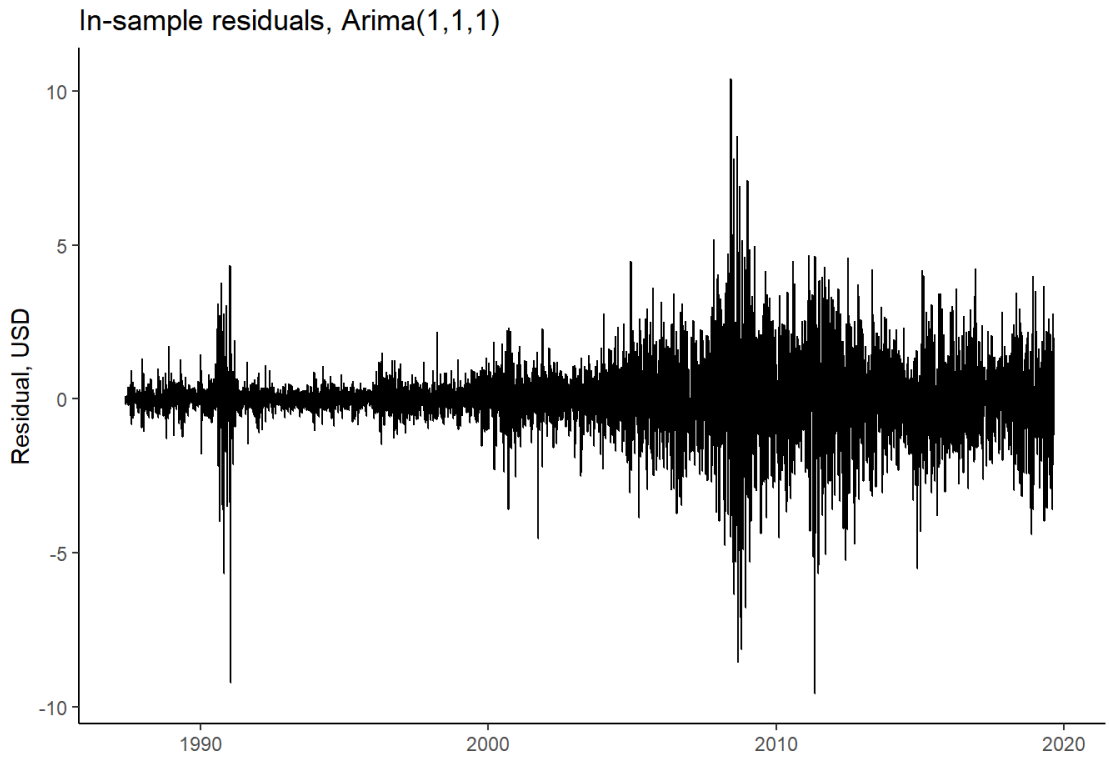
```
# B
arima_111_pred <- forecast(arima_111, h = sum(!before_2019_08_30))

in_sample <-
  data.frame(
  date = oil$date[before_2019_08_30],
  arima_111_residuals = arima_111$residuals,
  arima_111_fitted = arima_111$fitted)

out_of_sample <-
  data.frame(
    date = oil$date[!before_2019_08_30],
    arima_111_pred   = as.numeric(arima_111_pred$mean),
    arima_111_lwr80  = as.numeric(arima_111_pred$lower[,1]),
    arima_111_lwr95  = as.numeric(arima_111_pred$lower[,2]),
    arima_111_upr80  = as.numeric(arima_111_pred$upper[,1]),
    arima_111_upr95  = as.numeric(arima_111_pred$upper[,2]))

oil <-
  oil %>%
  left_join(in_sample, by="date") %>%
  left_join(out_of_sample, by="date")

oil %>%
  ggplot(aes(x=date))+
  geom_line(aes(y = oil, col = "Observed"))+
  geom_line(aes(y = arima_111_fitted, col="Fitted, values, ARIMA(1,1,1)"))+
  ggtitle("Oil price and in-sample model fit")+
  xlab("")+
  ylab("Oil price, USD")+
  theme_classic()+
  theme(legend.position="top")+
  guides(col=guide_legend(title=""))
```

```
## Warning: Removed 41 rows containing missing values (geom_path).
```

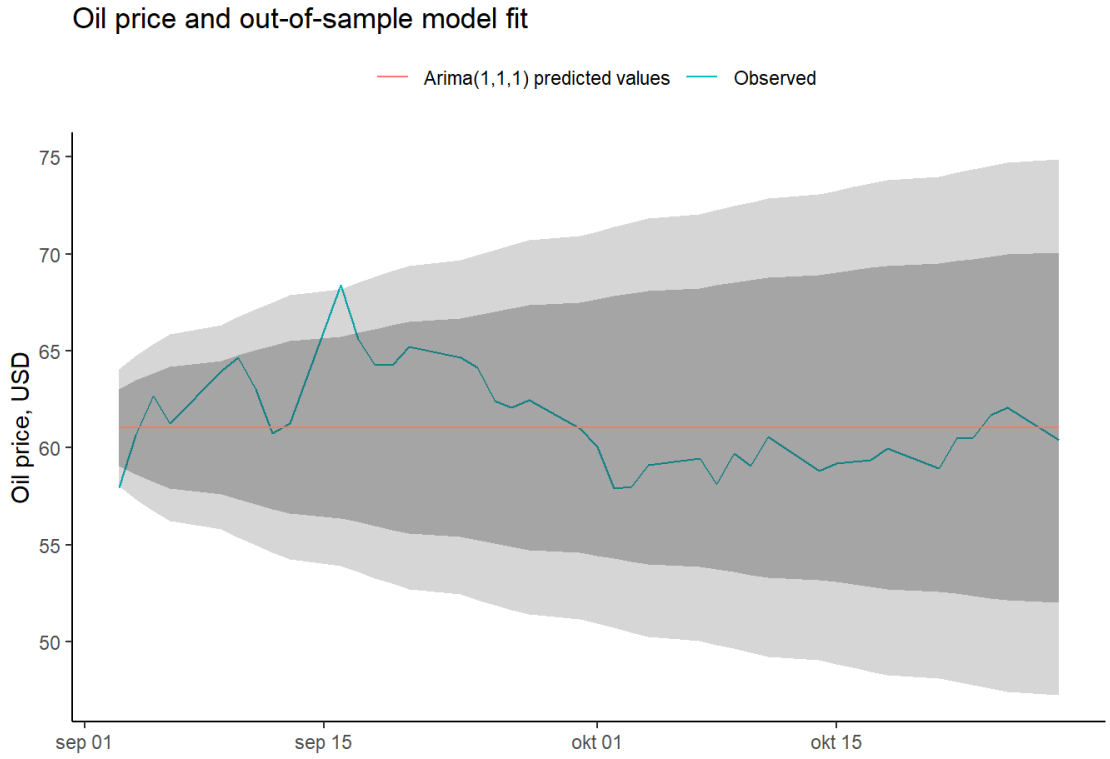


```
oil %>%
  ggplot(aes(x=date, y = arima_111_residuals))+
  geom_line(aes())+
  ggtitle("In-sample residuals, Arima(1,1,1)")+
  xlab("")+
  ylab("Residual, USD")+
  theme_classic()
```

```
## Warning: Removed 41 rows containing missing values (geom_path).
```




```
oil %>%
  tail(40) %>%
  ggplot(aes(x=date))+
  geom_line(aes(y = oil, col = "Observed"))+
  geom_ribbon(aes(ymin = arima_111_lwr95, ymax = arima_111_upr95), alpha=.2)+
  geom_ribbon(aes(ymin = arima_111_lwr80, ymax = arima_111_upr80), alpha=.3)+
  geom_line(aes(y=arima_111_pred, col="Arima(1,1,1) predicted values"))+
  ggtitle("Oil price and out-of-sample model fit")+
  xlab("")+
  ylab("Oil price, USD")+
  theme_classic()+
  theme(legend.position="top")+
  guides(col=guide_legend(title=""))
```



Assignment 3

Model evaluation has been discussed in class. They have seen several methods for doing this: in-sample, out of sample and iteratively updating the model and predictions through the data set. Any one of these could be acceptable answers, but very good answers should opt for an out-of-sample estimation.

In this solution, we show a “gold standard” method for assessing model performance. We re-estimate the model for all dates after a fixed start date, and use the estimated model to predict N periods ahead. Hence, we can measure the predictive performance N-timesteps into the future. The reason for doing this is that in assignment 4, we will be predicting the oil price a few days into the future. Hence, we want to choose a model that does a good job at exactly that.

The curriculum covers ARIMA and exponential smoothing. Students might at this assignment submit very different models. Creativity (e.g. subsetting the data in clever ways, using more advanced models etc) should be rewarded. A simple benchmark (e.g. using the last observation as a prediction) could also be a useful benchmark model - and also potentially one that is hard to beat.

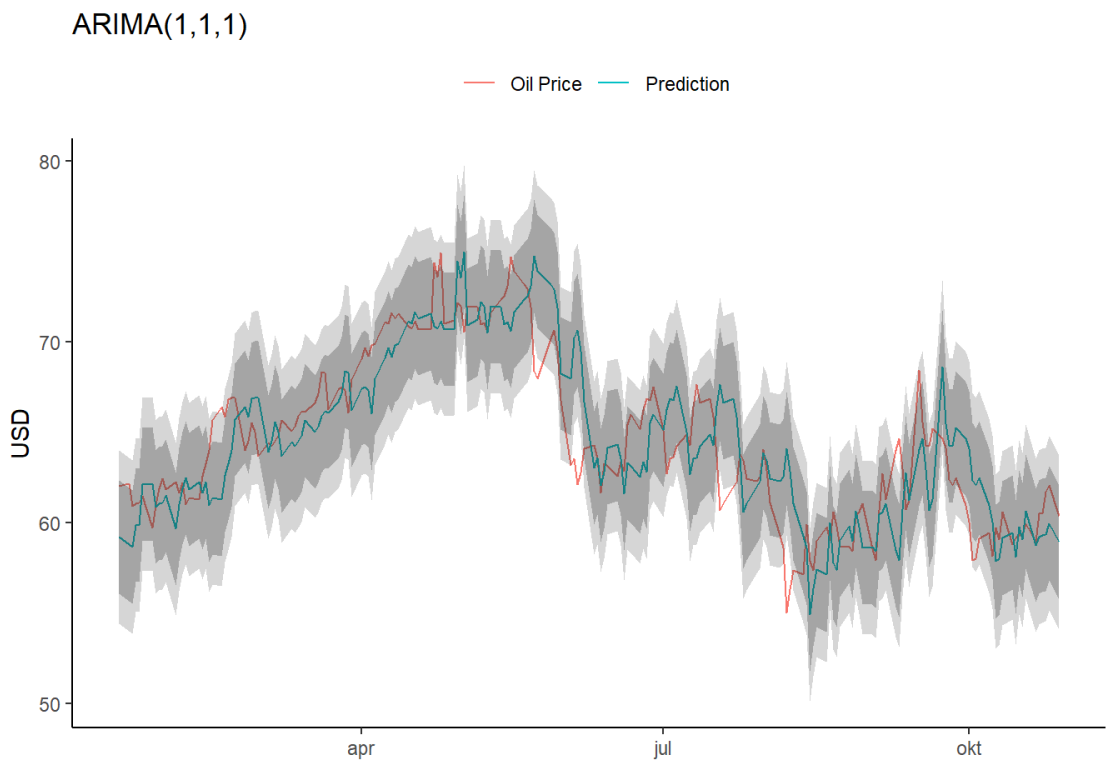
Comparing three methods in this solution proposal (Arima(1,1,1), ETS and Holt) we see that none of the models are very good, but ARIMA is marginally better than ETS, and Holt. However, is is also interesting to note that the prediction intervals (both 80 and 95%) are closest to the values they should be (i.e. 20% and 5%) for the ARIMA-model. Hence, among these three models, it seems that the Arima(1,1,1) model captures the uncertainty of the N-ahead predictions fairly well. This is somewhat unexpected, as the variance of the oil price is not homoskedastic. Note that the results from this benchmark may vary, depending on e.g. start date and number of periods ahead used in the prediction.

```
start_date_estimation <-
  oil %>%
  select(date) %>%
  filter(date >= as.Date('2017-01-01')) %$%
  min(date)

N_ahead <- 5

pred_mod1 <- pred_mod2 <-  pred_mod3 <-
  oil %>%
  select(date, oil) %>%
  mutate(
    `Point Forecast` = NA,
    `Lo 80`          = NA,
    `HI 80`          = NA,
    `Lo 95`          = NA,
    `HI 95`          = NA)

predvars <- c("Point Forecast","Lo 80","HI 80","Lo 95","HI 95")

r_train   <- as.ts(oil$oil)

for(i in which(oil$date==start_date_estimation):(nrow(oil)-N_ahead)){

  pred_mod1[i+N_ahead,predvars] <-
    forecast::Arima(r_train[1:i], order = c(1,1,1)) %>%
    forecast(h=N_ahead) %>%
    as.data.frame() %>%
    tail(1)

  pred_mod2[i+N_ahead,predvars] <-
    forecast::ses(r_train[1:i], h = N_ahead) %>%
    as.data.frame() %>%
    tail(1)

  pred_mod3[i+N_ahead,predvars] <-
    forecast::holt(r_train[1:i], h = N_ahead) %>%
    as.data.frame() %>%
    tail(1)
}


summarise_predictions <-
  function(predframe, name){
    predframe %>%
      mutate(
        cov_80 =
          case_when(
            oil > `HI 80` ~ 1,
            oil < `Lo 80` ~ 1,
            is.na(`HI 80`) ~ NA_real_,
            TRUE ~ 0),
        cov_95 =
          case_when(
            oil > `HI 95` ~ 1,
            oil < `Lo 95` ~ 1,
            is.na(`HI 95`) ~ NA_real_,
            TRUE ~ 0)) %>%
      summarise(
        mse = mean((oil - `Point Forecast`)^2, na.rm = T),
        mad = median((oil - `Point Forecast`), na.rm = T),
        cov80 = mean(cov_80, na.rm = T),
        cov95 = mean(cov_95, na.rm = T)) %>%
      cbind(data.frame(model = name))
  }

mod_results <-
  summarise_predictions(pred_mod1, "ARIMA(1,1,1)") %>%
  rbind(summarise_predictions(pred_mod2, "SES")) %>%
```

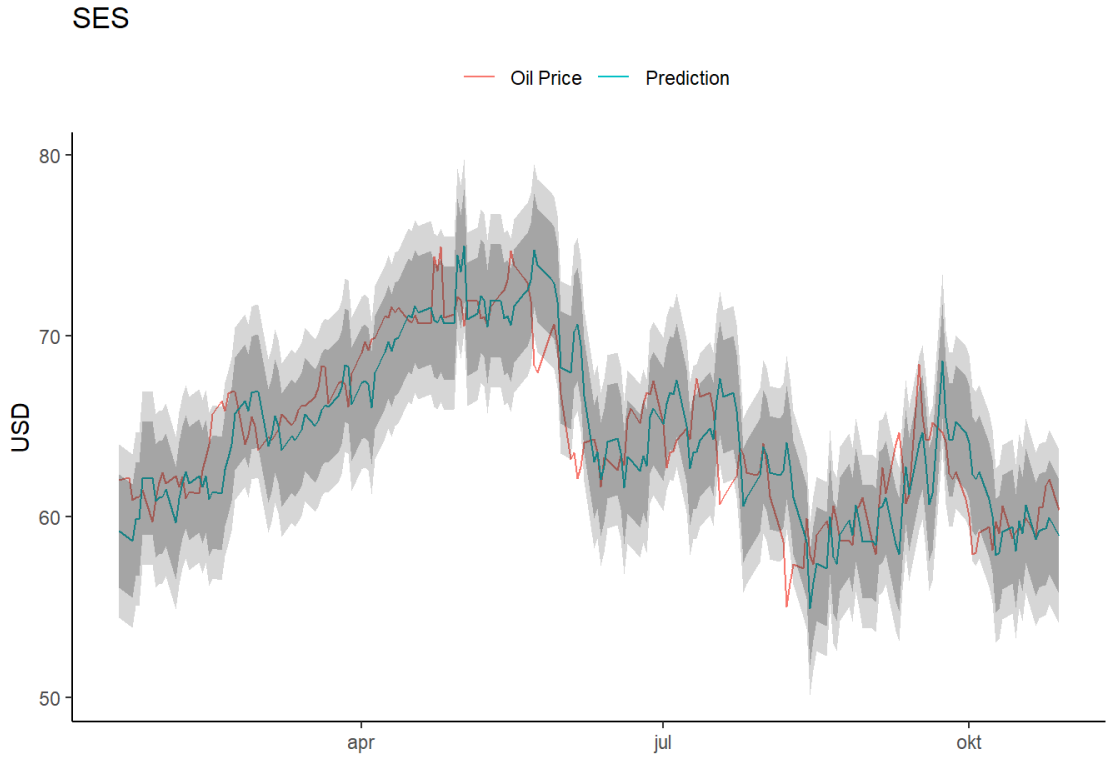
```
  rbind(summarise_predictions(pred_mod2, "Holt"))

plot_preds <-
  function(predframe, name){
    pred_mod1 %>%
      tail(200) %>%
      filter(complete.cases(.)) %>%
      ggplot(aes(x=date))+
      geom_line(aes(y=oil, col="Oil Price"))+
      geom_line(aes(y=`Point Forecast`, col="Prediction"))+
      ggtitle(name)+
      geom_ribbon(aes(ymin = `Lo 95`, ymax = `HI 95`), alpha=.2)+
      geom_ribbon(aes(ymin = `Lo 80`, ymax = `HI 80`), alpha=.3)+
      ylab("USD")+
      xlab("")+
      theme_classic()+
      theme(legend.position="top")+
      guides(col=guide_legend(title=""))
  }

mod_results
```

```
##        mse       mad     cov80      cov95        model
## 1 6.623357 0.2949717 0.2125874 0.06013986 ARIMA(1,1,1)
## 2 6.622126 0.2899339 0.2167832 0.06713287          SES
## 3 6.622126 0.2899339 0.2167832 0.06713287         Holt
```

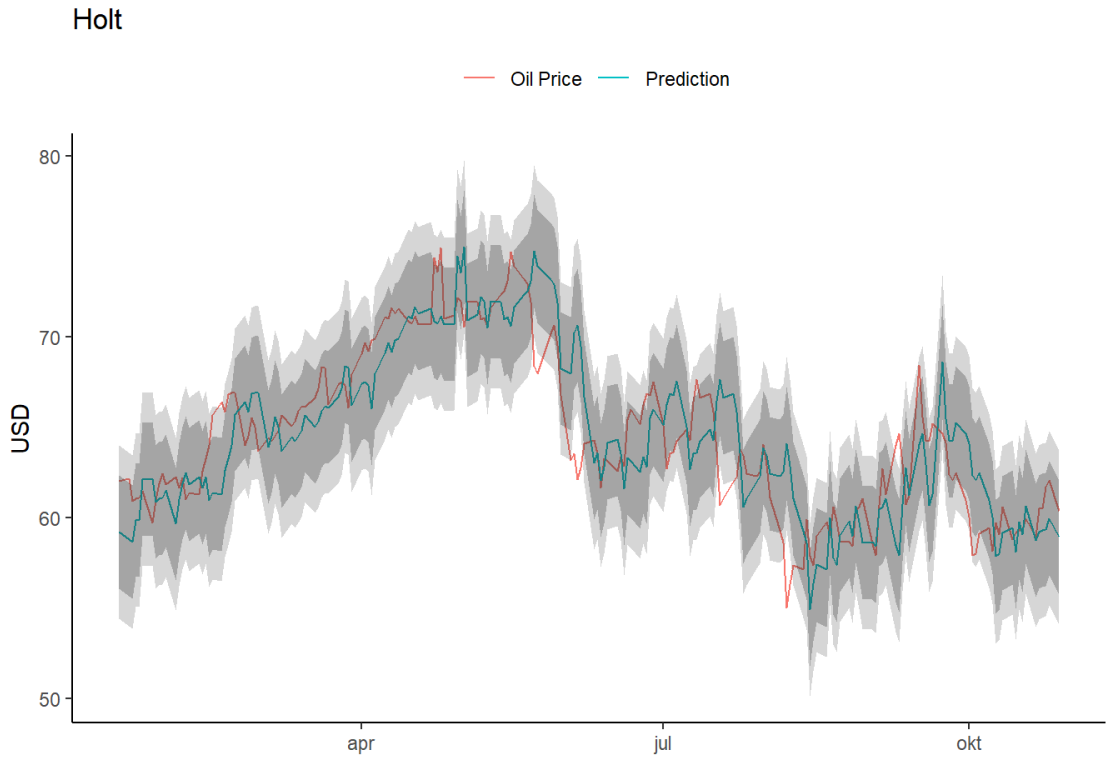
```
plot_preds(pred_mod1, "ARIMA(1,1,1)")
```




```
plot_preds(pred_mod2, "SES")
```




```
plot_preds(pred_mod3, "Holt")
```



Assignment 4

As the Arima(1,1,1) had the best performance, we'll use that for the final assignment as well. However, we'll re-estimate it using all available observations of the oil price, and predict the oil price for 2019-11-15.

The expected profit - assuming the errors have an expected value of zero - is

$$E\left[50 - \left(oil_{2019-11-15} - \hat{oil}_{2019-11-15}\right)^2\right] = 50 - Var\left[oil_{2019-11-15} - \hat{oil}_{2019-11-15}\right] - E\left[oil_{2019-11-15} - \hat{oil}_{2019-11-15}\right]^2$$

If the Arima-model produces unbiased predictions, then $E\left[oil_{2019-11-15} - \hat{oil}_{2019-11-15}\right]^2 = 0$. This implies that the expression reduces to:

$$E\left[50 - \left(oil_{2019-11-15} - \hat{oil}_{2019-11-15}\right)^2\right] = 50 - \sigma^2.$$

If we take the arima-model very literally, where errors are normally distributed, then $\frac{(oil_{2019-11-15} - \hat{oil}_{2019-11-15})}{\sigma}$ is a standard normal variable. Therefore, $\frac{(oil_{2019-11-15} - \hat{oil}_{2019-11-15})^2}{\sigma^2}$ should be χ^2-distributed with one degree of freedom (this is because the χ^2-distribution is a distribution of the squares of standard normally distributed variables). Note that we divide by σ^2, and not σ. Hence, we can use the χ^2 distribution to find the probability of a negative profit.

In order to calculate the expected profit and probability of a negative profit, we need an estimate of σ^2. A key point here, however, is that uncertainty grows the further into the future we predict, so we need to find the σ^2 correspondning to the 2019-11-15-prediction.

We could calculate the variance of the prediction error directly. However, we can also use the prediction intervals that are reported from the forecasts. The 95% prediction intervals is calculated as $\hat{oil} \pm 1.96\sigma$. Hence, we can e.g use the upper bound for the prediction interval, and solve it for σ, which gives

$$\sigma_{2019-11-15} = \frac{\hat{oil}^{97.5}_{2019-11-15} - \hat{oil}_{2019-11-15}}{1.96}.$$

However, given that errors are indeed not normally distributed, the answer below is unlikely to reflect the true probability of a negative profit.

```
final_model <- forecast::Arima(as.ts(oil$oil), order = c(1,1,1))
#
final_prediction <-
  final_model %>%
  forecast(h=as.Date("2019-11-15")-max(oil$date)) %>%
  as.data.frame() %>%
  tail(1)

print(final_prediction)
```

```
##      Point Forecast    Lo 80    Hi 80   Lo 95    Hi 95
## 8254       60.34725 54.35566 66.33885 51.1839 69.51061
```

```
r <- 50
sigma2 <- ((final_prediction$`Hi 95`-final_prediction$`Point Forecast`)/1.96)^2
Expected_profit <- r - sigma2
Prob_neg_profit <- 1-pchisq(r/sigma2, df=1)
#
print(Expected_profit)
```

```
## [1] 28.14267
```

```
print(Prob_neg_profit)
```

```
## [1] 0.1304146
```

Another way of finding this probability would be to repeat an exercise similar to assignment 3, except that we count how many times in the past predictions would by so far off that profits turn negative. Returning to the article in the reference (Makridakis et al), there is no limit to the potential loss from the proposed trade. In principle, any trader, no matter how deep pockets, might potentially become ruined from commiting to such a profit function. The probability of such events might be very small, but are not impossible as seen from the history of the oil price. I would therefore not take the bet - but students' preferences might be different.