Project 2

Team Members: Tejeswini Anjaneya Reddy, Shashank Mankala

Part 1

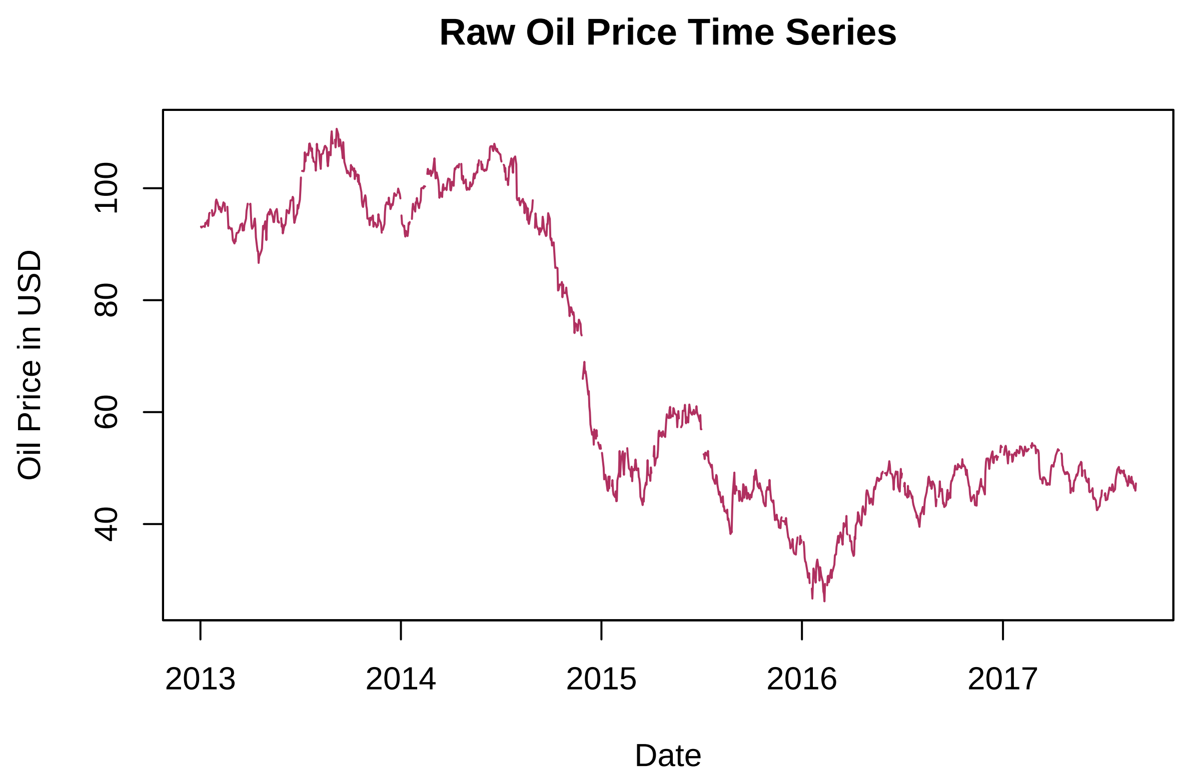
```
library(zoo)

oil_data <- read.csv("oil.csv")
```

Part 2

```
oil_data$date <- as.Date(oil_data$date, format = "%Y-%m-%d")
oil_ts <- zoo(oil_data$dcoilwtico, oil_data$date)

plot(oil_ts,
     main = "Raw Oil Price Time Series",
     xlab = "Date", ylab = "Oil Price in USD",
     col = "maroon", type = "l")
```



Part 3

```
summary(oil_data)
```

```
##       date              dcoilwtico
##  Min.   :2013-01-01   Min.   : 26.19
##  1st Qu.:2014-03-03   1st Qu.: 46.41
##  Median :2015-05-02   Median : 53.19
##  Mean   :2015-05-02   Mean   : 67.71
##  3rd Qu.:2016-06-30   3rd Qu.: 95.66
##  Max.   :2017-08-31   Max.   :110.62
##                       NA's   :43
```

```
sum(is.na(oil_data$dcoilwtico))
```

```
## [1] 43
```

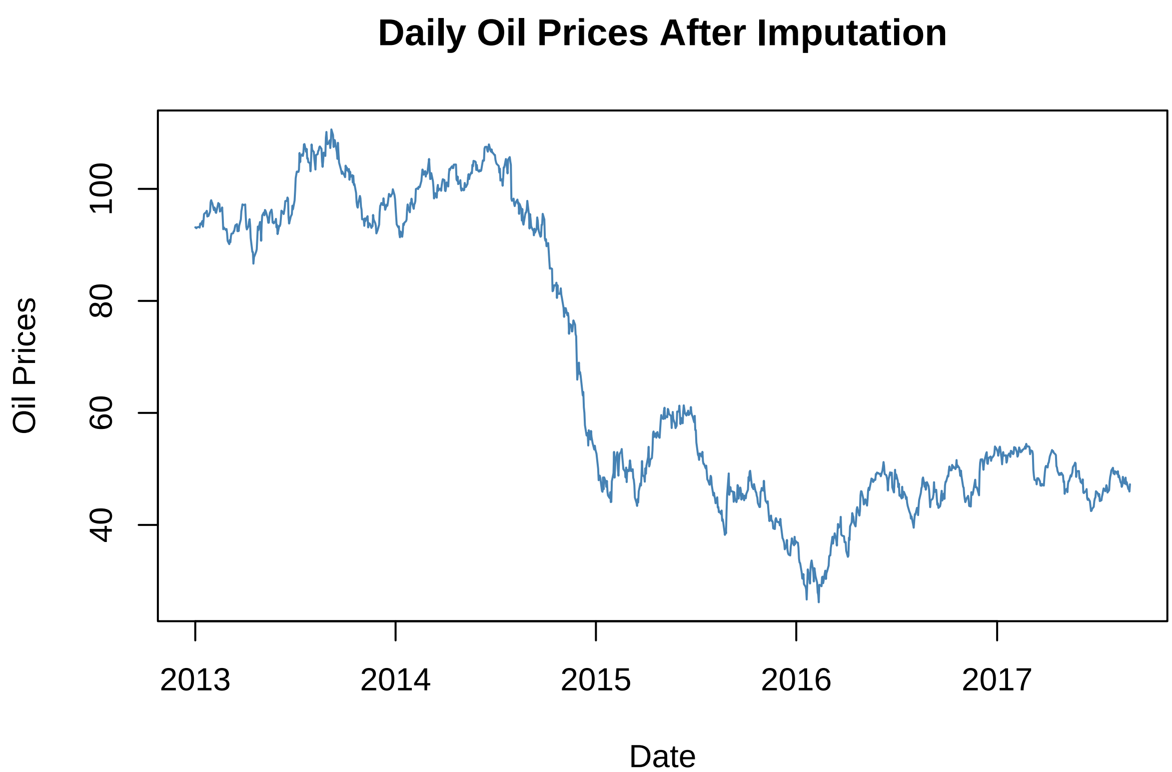
```
library(imputeTS)

imputed_oil_data <- na.interpolation(oil_data$dcoilwtico)
```

Part 4

```
imputed_oil_ts <- ts(imputed_oil_data, start = c(2013, 1), frequency = 365)

imputed_oil_data <- data.frame(
  date = oil_data$date,
  dcoilwtico = coredata(imputed_oil_ts)
)
plot(imputed_oil_data$date, imputed_oil_data$dcoilwtico, type = "l", col = "steelblue",
     main = "Daily Oil Prices After Imputation",
     xlab = "Date", ylab = "Oil Prices")
```



After imputing missing values using linear interpolation, the time series is continuous without noticeable artifacts from imputation. The imputed series preserved the overall pattern and major shifts in oil prices, notably the sharp drop in mid-2014. We cannot see any clear seasonality but medium-term trends and cycles appear.

Part 5

The ETS models are of exponential smoothing methods that model time series data by decomposing it into three components namely error, trend, and seasonality.
These models are useful for handling data with clear trend and seasonal patterns.

ETS models directly select the best combination of components based on information criteria like AIC. We will use the ets() function from the forecast package to fit these models and the forecast() function will be used to generate future values.
The Holt-Winters method is a specific case of ETS models. It extends simple exponential smoothing by adding components for trend and seasonality. It has 2 forms namely additive and multiplicative. The additive version can be used when the seasonal variation is roughly constant over time, while the multiplicative version is appropriate when the variation changes proportionally with the level of the series. The method updates the level, trend, and seasonal components at each time step. We will implement it using HoltWinters() function.
We can say that Holt-Winters is limited in flexibility and ETS models offer a broader range of configurations and support for damped trends. We can evaluate the model performance using tools like checkresiduals() and accuracy().

```r
library(forecast)
ets_model <- ets(imputed_oil_ts)

summary(ets_model)
```

```
## ETS(A,N,N)
##
## Call:
## ets(y = imputed_oil_ts)
##
##   Smoothing parameters:
##     alpha = 0.9683
##
##   Initial states:
##     l = 93.1395
##
##   sigma:  1.1771
##
##      AIC     AICc      BIC
## 9055.133 9055.153 9070.448
##
## Training set error measures:
##                       ME     RMSE       MAE         MPE     MAPE       MASE
## Training set -0.03893581 1.176176 0.8950884 -0.08222931 1.548288 0.03256606
##                       ACF1
## Training set -0.0007500942
```

```r
hw_model <- HoltWinters(imputed_oil_ts, seasonal = "additive")

hw_model
```

```
## Holt-Winters exponential smoothing with trend and additive seasonal component.
##
## Call:
## HoltWinters(x = imputed_oil_ts, seasonal = "additive")
##
## Smoothing parameters:
##  alpha: 0.918104
##  beta : 0
##  gamma: 1
```

```
##
## Coefficients:
##              [,1]
## a     51.08090472
## b     -0.09657653
## s1    -2.40677664
## s2    -2.51681642
## s3    -3.31555151
## s4    -3.41167794
## s5    -2.27299329
## s6    -1.29063747
## s7    -1.76034147
## s8    -3.07676790
## s9    -3.30047111
## s10   -6.85914575
## s11  -10.64228675
## s12   -8.07150106
## s13   -9.38470294
## s14   -8.90573563
## s15   -9.41001142
## s16  -10.14343447
## s17  -12.68409010
## s18  -12.17004565
## s19  -14.39890182
## s20  -15.28624093
## s21  -17.25130505
## s22  -19.07012773
## s23  -18.87051882
## s24  -18.26628566
## s25  -20.24276615
## s26  -17.82511323
## s27  -18.85003149
## s28  -17.46289210
## s29  -18.25661528
## s30  -18.66797912
## s31  -19.10923033
## s32  -19.95634394
## s33  -19.29751894
## s34  -19.63401669
## s35  -19.94021592
## s36  -20.11843642
## s37  -22.48716819
## s38  -24.44274823
## s39  -23.80183770
## s40  -23.50285202
## s41  -23.79493809
## s42  -25.66127133
## s43  -25.83425906
## s44  -23.33527602
## s45  -25.06653692
## s46  -23.18349675
## s47  -23.56836234
## s48  -24.25634339
## s49  -23.21542147
```

```
## s50  -24.85461257
## s51  -25.17112578
## s52  -25.51952738
## s53  -24.61274421
## s54  -25.96791946
## s55  -25.90479410
## s56  -22.62012234
## s57  -20.72343826
## s58  -16.69535939
## s59  -20.45162410
## s60   -7.76761616
## s61    5.37943426
## s62    5.77261988
## s63    4.42053585
## s64    6.86361385
## s65    5.62197785
## s66    4.20478800
## s67    2.88879021
## s68    2.55283840
## s69    1.53849206
## s70    1.98507316
## s71    1.76598191
## s72    1.26158493
## s73    0.88957655
## s74    2.82907351
## s75    1.95460344
## s76    2.63432055
## s77    1.86000856
## s78    2.24580605
## s79    0.68740959
## s80    1.44443345
## s81    0.86248316
## s82    0.93412457
## s83   -0.22574929
## s84    0.79925092
## s85   -0.48413413
## s86   -0.79207434
## s87   -1.98111123
## s88   -3.68689841
## s89   -4.51134534
## s90   -4.76783209
## s91   -4.07681199
## s92   -2.86052163
## s93   -3.22760938
## s94   -4.44630646
## s95   -4.92531727
## s96   -6.95539002
## s97   -7.25576779
## s98   -8.15807115
## s99   -7.12714637
## s100  -7.47784829
## s101  -7.13694346
## s102  -6.73329958
## s103  -8.34818572
```

```
## s104  -8.46875603
## s105  -8.33250502
## s106  -8.04851626
## s107  -8.82236014
## s108  -8.61266962
## s109  -8.44283074
## s110  -6.80167525
## s111  -7.67382286
## s112  -8.07712281
## s113  -8.79499582
## s114  -9.94356767
## s115  -9.81510288
## s116  -9.75980245
## s117  -8.75601953
## s118  -6.57445594
## s119  -5.19314328
## s120  -4.93957927
## s121  -4.93728124
## s122  -4.97216398
## s123  -4.01801413
## s124  -4.53293978
## s125  -5.00822681
## s126  -5.73552778
## s127  -5.03856733
## s128  -5.06780229
## s129  -4.60465771
## s130  -3.91990777
## s131  -3.28522487
## s132  -3.54588798
## s133  -3.44962194
## s134  -3.15312090
## s135  -2.93176270
## s136  -2.49854974
## s137  -3.29769870
## s138  -3.99862173
## s139  -5.24446912
## s140  -6.95011599
## s141  -8.15622631
## s142  -9.06865749
## s143  -8.98862187
## s144 -10.35523079
## s145 -10.89596908
## s146 -10.06511817
## s147 -10.94061867
## s148 -10.12983963
## s149  -8.78466375
## s150  -8.84319871
## s151  -8.50915088
## s152  -8.06903971
## s153  -7.75643519
## s154  -6.17036387
## s155  -5.18078589
## s156  -5.64432203
## s157  -6.23392568
```

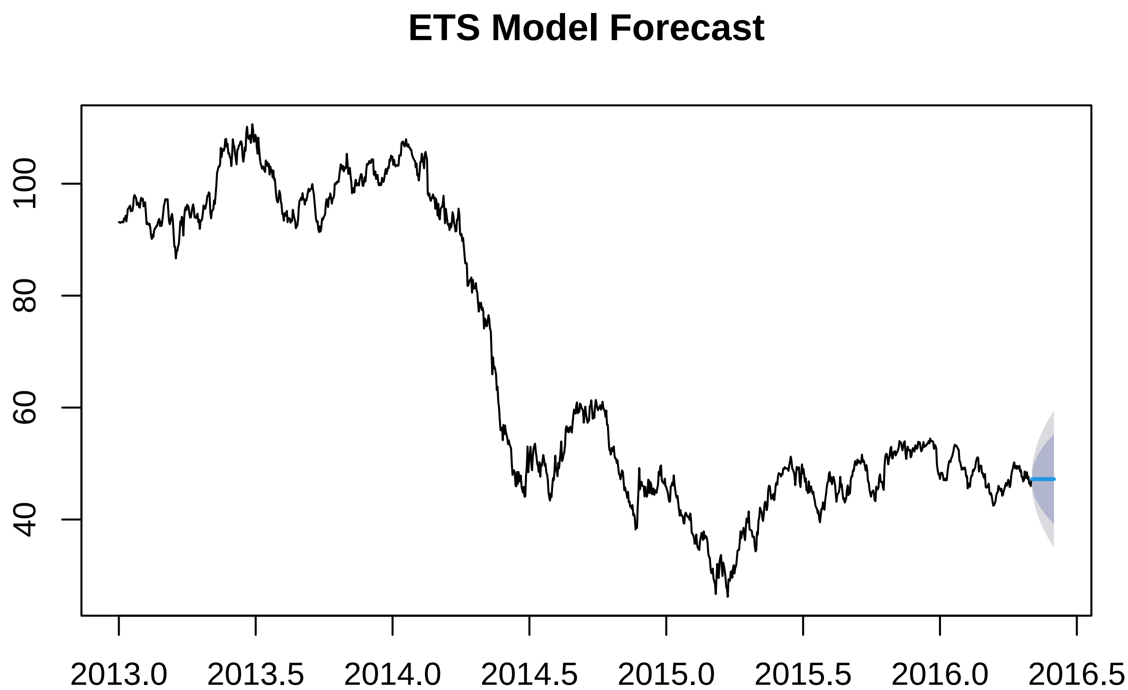
```
## s158  -4.63877281
## s159  -4.88657753
## s160  -4.07155049
## s161  -4.43542094
## s162  -5.35799180
## s163  -4.74706854
## s164  -4.90798059
## s165  -4.22722913
## s166  -2.46008190
## s167  -2.11488369
## s168  -1.96057789
## s169  -1.23874943
## s170  -1.27637251
## s171  -1.38347729
## s172  -0.51651354
## s173   0.64292231
## s174   1.42802370
## s175   1.62890090
## s176   1.13002753
## s177   1.61793305
## s178   0.93693172
## s179   1.17020381
## s180   1.22956316
## s181   1.61216422
## s182   3.66993675
## s183   2.73701704
## s184   0.94812733
## s185   0.80099444
## s186   1.54026853
## s187   0.50633439
## s188  -0.37089844
## s189  -2.16758320
## s190  -2.27508388
## s191  -1.67125365
## s192  -2.09258063
## s193  -0.91307199
## s194   0.17626794
## s195  -0.56512398
## s196  -0.30218618
## s197  -0.33046185
## s198  -0.40104436
## s199   0.49287035
## s200   1.37327243
## s201   1.83654565
## s202   1.81695902
## s203   0.56958731
## s204   0.17404752
## s205   1.30383672
## s206   2.30069447
## s207   1.99069321
## s208   3.20968450
## s209   4.72223145
## s210   4.99745983
## s211   5.20805098
```

```
## s212   5.68543847
## s213   5.64725422
## s214   5.71615232
## s215   6.50720430
## s216   6.61091606
## s217   6.68938996
## s218   4.67065134
## s219   4.51182922
## s220   5.06839123
## s221   4.02172312
## s222   4.38804261
## s223   4.87209418
## s224   3.70780304
## s225   3.52384612
## s226   3.96268227
## s227   3.76346203
## s228   4.06970864
## s229   5.17352019
## s230   5.13143247
## s231   5.05521615
## s232   5.71698122
## s233   7.01701154
## s234   7.66254387
## s235   7.24496847
## s236   7.85484640
## s237   8.35173379
## s238   8.59126507
## s239   9.87996945
## s240  10.13995837
## s241  11.23289752
## s242  11.32284942
## s243  10.49907421
## s244  10.25193503
## s245  11.07687484
## s246  10.58559059
## s247  10.32977265
## s248  10.68834857
## s249  11.10027072
## s250  10.94744705
## s251  11.24987881
## s252  12.77782691
## s253  13.39180276
## s254  13.34163740
## s255  15.28889558
## s256  15.93735275
## s257  16.17834582
## s258  16.00997664
## s259  15.66590902
## s260  16.14716814
## s261  17.09453427
## s262  16.59248007
## s263  16.27651998
## s264  16.71532537
## s265  16.46135581
```

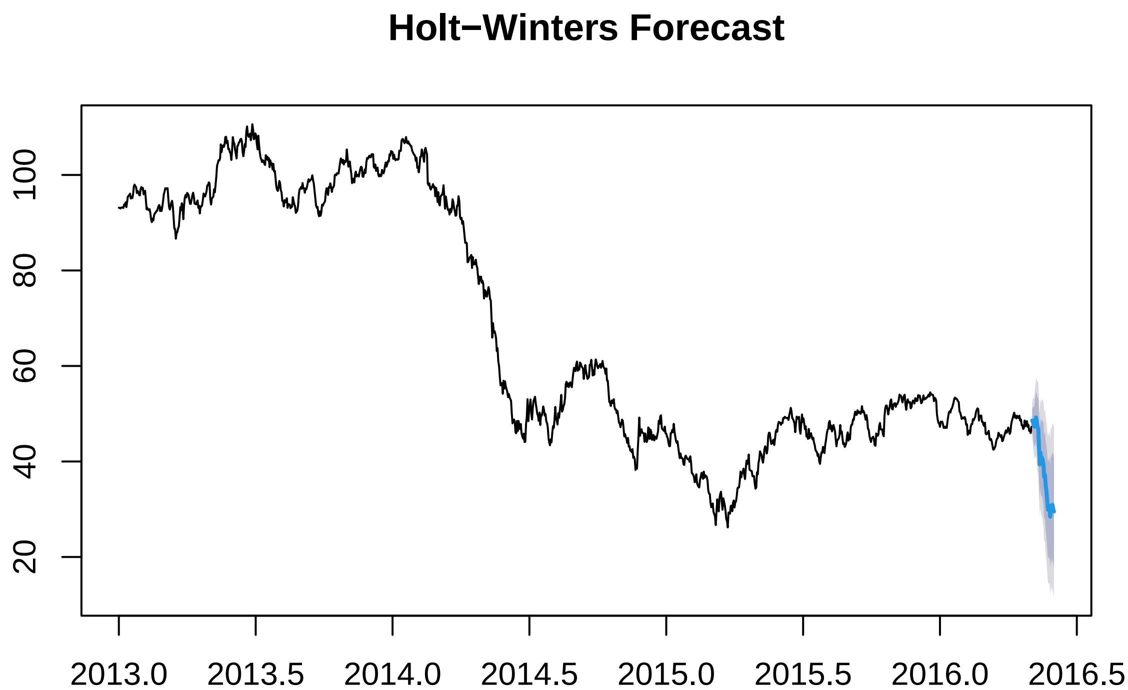
```
## s266  16.40979020
## s267  16.27994364
## s268  16.51554429
## s269  15.69603284
## s270  15.43949638
## s271  15.24461019
## s272  15.10197775
## s273  15.20126860
## s274  14.29145765
## s275  14.94974009
## s276  13.27067883
## s277  13.14526286
## s278  12.36621844
## s279  13.56138030
## s280  15.30919277
## s281  15.88627340
## s282  17.42176807
## s283  17.10105626
## s284  16.44785690
## s285  15.55945001
## s286  17.57916845
## s287  18.48779092
## s288  18.09497636
## s289  17.58870807
## s290  12.50127442
## s291  11.39313083
## s292  11.62636052
## s293  11.13560982
## s294  10.55155955
## s295  10.84461358
## s296  11.45506662
## s297  11.93724003
## s298  11.43480101
## s299  11.41157184
## s300  10.00820004
## s301  11.33172231
## s302  10.82869729
## s303   9.20150656
## s304  10.57263281
## s305   9.10970444
## s306   8.45834203
## s307  10.05200407
## s308  10.73558630
## s309  10.69992598
## s310  11.76935068
## s311  13.23821949
## s312  11.08803858
## s313   8.77854866
## s314  10.95311605
## s315  10.51294967
## s316   9.50249457
## s317   8.67452825
## s318   8.78715960
## s319   8.30805416
```

```
## s320   9.06323424
## s321   8.76203868
## s322   9.21806727
## s323  11.07415644
## s324  11.06416951
## s325  10.05260235
## s326   9.61609509
## s327   8.69022701
## s328   8.72643441
## s329  10.65303610
## s330  10.93574562
## s331  12.58981585
## s332  12.21691945
## s333   9.36918692
## s334   8.49395031
## s335   8.78690832
## s336   7.75791449
## s337   8.33176605
## s338   7.42918739
## s339   5.83479959
## s340   4.33587152
## s341   4.39874233
## s342   4.41915528
## s343   0.76643750
## s344   0.63161228
## s345   1.00191417
## s346   1.68874664
## s347   1.63579001
## s348   2.15899198
## s349   0.34024683
## s350   1.91672230
## s351   0.93417145
## s352   0.82148053
## s353   1.19481854
## s354   1.69317024
## s355   1.20063541
## s356   0.88437294
## s357  -0.75906311
## s358  -2.20870612
## s359  -0.88514642
## s360  -1.41038352
## s361  -0.65270456
## s362  -1.58817239
## s363  -1.05197330
## s364  -1.24186127
## s365  -3.82090472
```

```
ets_forecast <- forecast(ets_model, h = 30)
plot(ets_forecast, main = "ETS Model Forecast")
```




```
hw_forecast <- forecast(hw_model, h = 30)
plot(hw_forecast, main = "Holt-Winters Forecast")
```



Part 6

Since we observed a long-term trend especially in 2014 when it first started to decline, we have recommended the following models:
1) ETS(A,A,N) or Holt's Linear Trend Model: We recommended this model because it captures the level and an additive trend which corresponds with the long-term trend. It is also simple and deals with the trend directly therefore it can be useful for short-term forecasting. This model may not give indication of any medium-term cyclical patterns because it assumes a linear trend.
2) ETS(A,Ad,N) - Holt's Damped Trend Model: It is a damped trend model which is a variation of Holt's linear trend that reduces the trend's influence over time. It can handle cyclical patterns better that aren't strictly linear. Thus this model can be suitable for oil prices, where the trend may stabilize or fluctuate. This model is more flexible than ETS(A,A,N) for capturing medium-term fluctuations and it also prevents over forecasting extreme trends but it can be slightly more complex to interpret due to the damping parameter.
3) ETS(A,N,N) - Simple Exponential Smoothing: This model smooths the level without modeling trend or seasonality. We considered this model to test if the trend is weak enough to be ignored or if differencing the series makes it stationary. Our assumption is that this model may underperform as we are seeing a trend.
4) Holt-Winters Additive (ETS(A,A,A)): This is a test to confirm that seasonality is negligible, but we can expect it to not perform well due to the lack of fixed seasonal cycles.

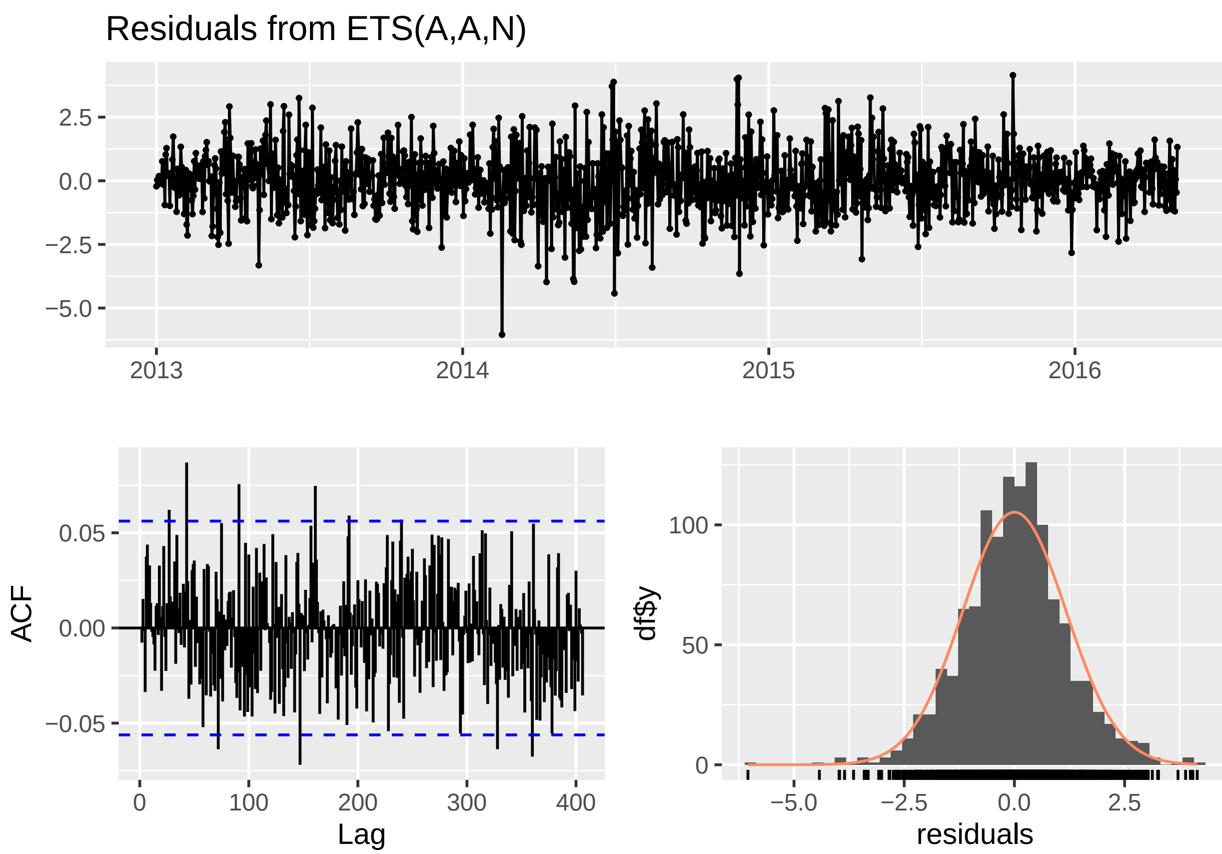
Part 7

```r
aan_ets <- ets(imputed_oil_ts, model = "AAN")

summary(aan_ets)
```

```
## ETS(A,A,N)
##
## Call:
## ets(y = imputed_oil_ts, model = "AAN")
##
##   Smoothing parameters:
##     alpha = 0.9676
##     beta  = 1e-04
##
##   Initial states:
##     l = 93.3935
##     b = -0.0379
##
##   sigma:  1.1775
##
##      AIC     AICc      BIC
## 9057.952 9058.002 9083.477
##
## Training set error measures:
##                        ME     RMSE       MAE         MPE     MAPE       MASE
## Training set 0.0002500013 1.175605 0.8948867 -0.01426733 1.546043 0.03255872
##                       ACF1
## Training set -5.143552e-05
```

```r
checkresiduals(aan_ets)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,A,N)
## Q* = 259.56, df = 244, p-value = 0.2358
##
## Model df: 0.   Total lags used: 244
```

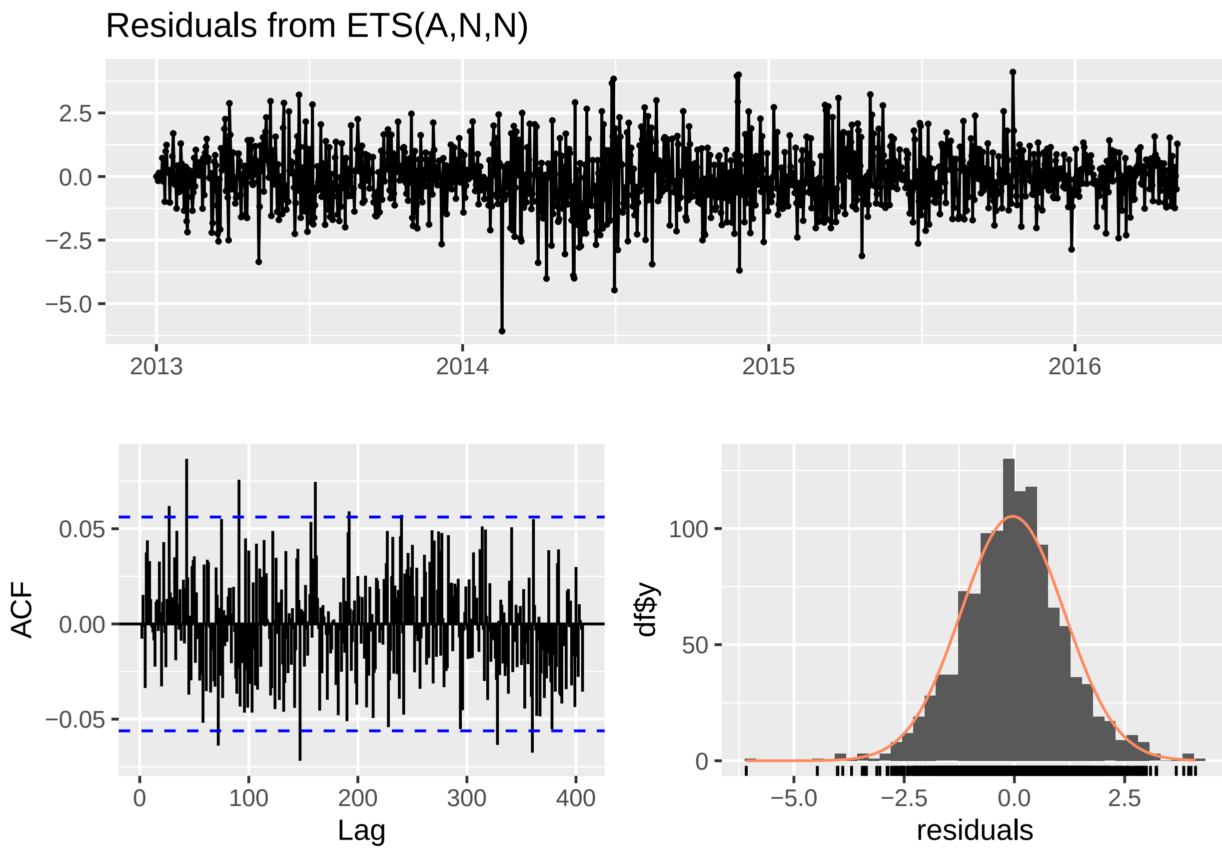
```r
ann_ets <- ets(imputed_oil_ts, model = "ANN")

summary(ann_ets)
```

```
## ETS(A,N,N)
##
## Call:
## ets(y = imputed_oil_ts, model = "ANN")
##
##   Smoothing parameters:
##     alpha = 0.9683
##
##   Initial states:
##     l = 93.1395
##
##   sigma:  1.1771
##
```

```
##       AIC     AICc      BIC
## 9055.133 9055.153 9070.448
##
## Training set error measures:
##                       ME     RMSE       MAE         MPE     MAPE       MASE
## Training set -0.03893581 1.176176 0.8950884 -0.08222931 1.548288 0.03256606
##                       ACF1
## Training set -0.0007500942
```

```r
checkresiduals(ann_ets)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,N,N)
## Q* = 259.69, df = 244, p-value = 0.234
##
## Model df: 0.   Total lags used: 244
```

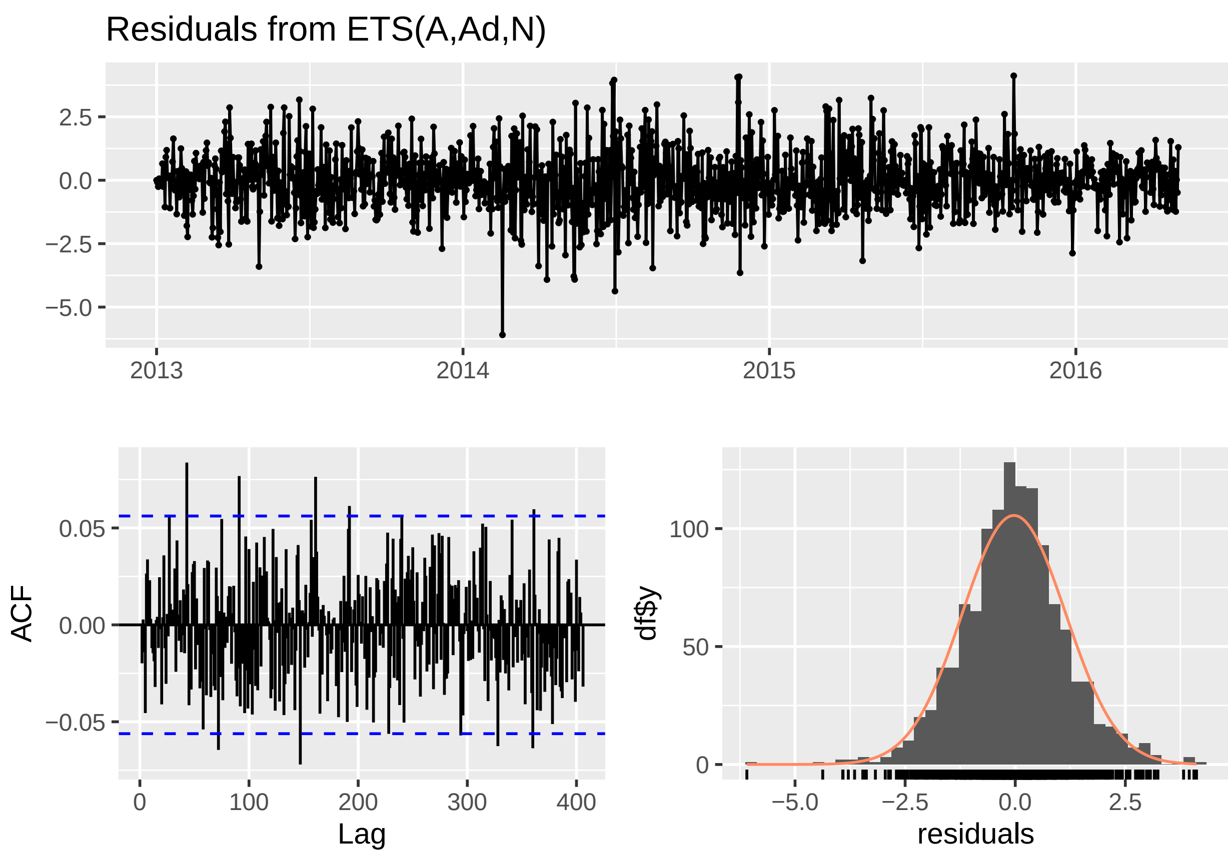
```r
aadn_ets <- ets(imputed_oil_ts, model = "AAN", damped = TRUE)

summary(aadn_ets)
```

```
## ETS(A,Ad,N)
```

```
##
## Call:
## ets(y = imputed_oil_ts, model = "AAN", damped = TRUE)
##
##   Smoothing parameters:
##     alpha = 0.9545
##     beta  = 0.0108
##     phi   = 0.9752
##
##   Initial states:
##     l = 93.0527
##     b = 0.0895
##
##   sigma:  1.1772
##
##      AIC     AICc      BIC
## 9058.238 9058.308 9088.868
##
## Training set error measures:
##                       ME     RMSE       MAE         MPE    MAPE       MASE
## Training set -0.02944197 1.174779 0.8936797 -0.05865279 1.54416 0.03251481
##                     ACF1
## Training set 0.0003706787
```

```
checkresiduals(aadn_ets)
```

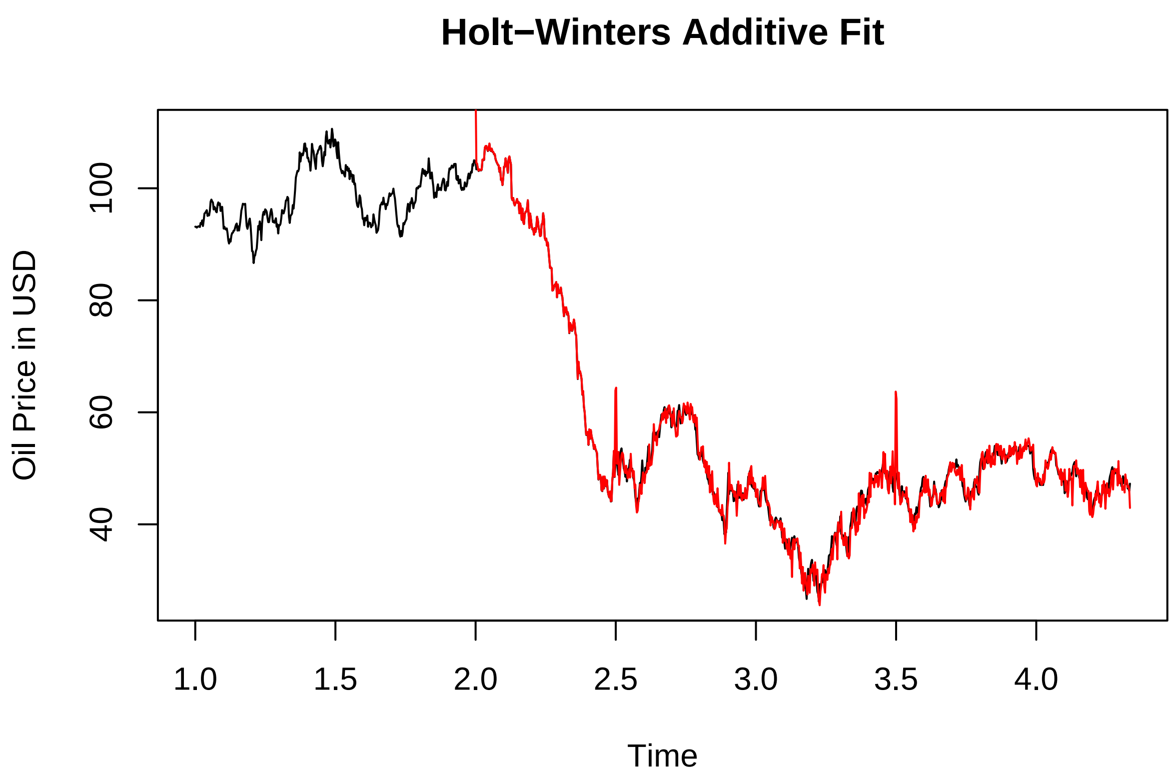


```
##
##  Ljung-Box test
##
## data:  Residuals from ETS(A,Ad,N)
## Q* = 259.43, df = 244, p-value = 0.2375
##
## Model df: 0.   Total lags used: 244
```

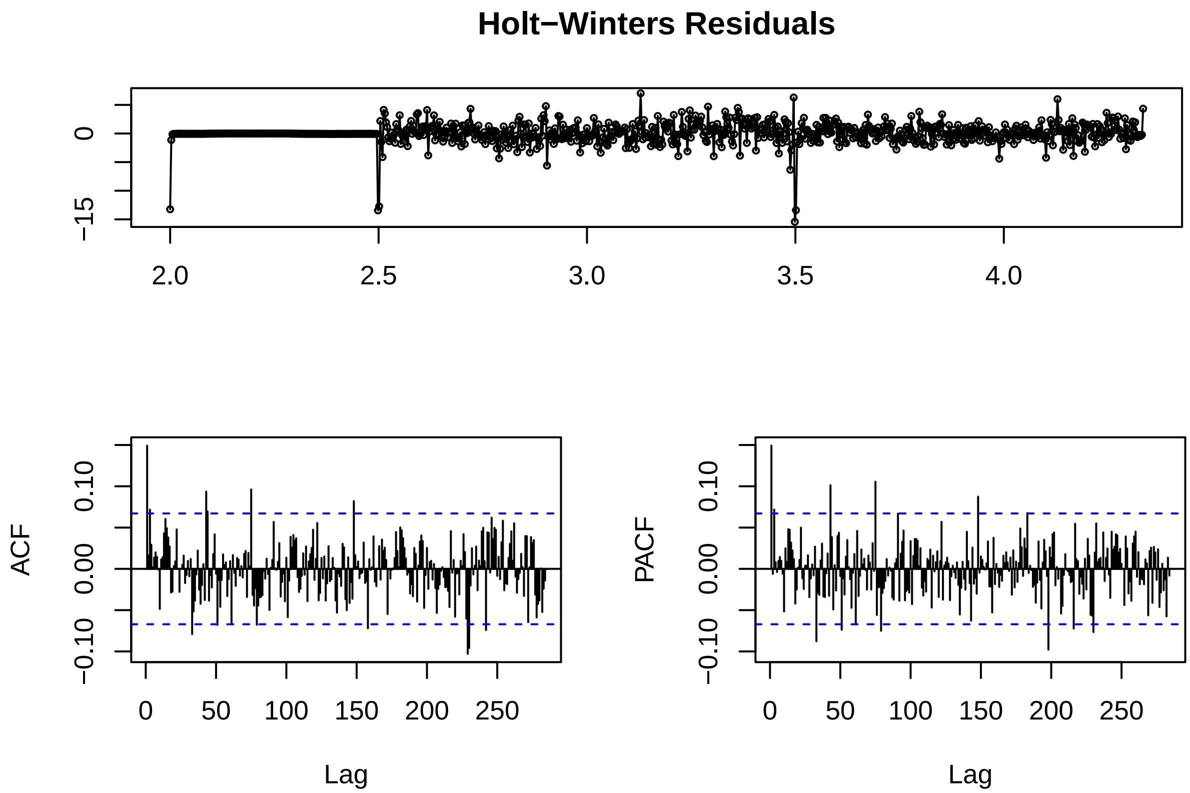
```r
oil_ts_daily <- ts(imputed_oil_ts, frequency = 365)

hw_additive <- HoltWinters(oil_ts_daily, seasonal = "additive")

plot(oil_ts_daily, main = "Holt-Winters Additive Fit", ylab = "Oil Price in USD")
lines(fitted(hw_additive)[,1], col = "red")
```




```r
hw_residuals <- oil_ts_daily - fitted(hw_additive)[,1]
tsdisplay(hw_residuals, main = "Holt-Winters Residuals")
```



Step 7: Adequacy Analysis Based on Plots

1. ETS(A,N,N) - We can see residuals fluctuating around zero in the residual plot.
For the most part, we did not notice any trend in the residuals over time.
However, we did see large spikes, similar to around 2014, that we observe a drop in oil prices, which the model did not account for.
In the ACF Plot, most lags fall within the blue confidence bounds which suggests us that there is no significant autocorrelation. We also noticed that there was no significant spikes suggesting us the residuals are mostly white noise. This is very reassuring in terms of model adequacy since it can capture key patterns.
In the Histogram, the residuals are roughly normally distributed, around center 0, but with some very large negative residuals.
As we see no significant spikes in the ACF section, the Ljung-Box p-value > 0.05 tells us that the residuals are uncorrelated.

2. ETS(A,A,N) Model - The residual plot is similar to that of ETS(A,N,N) residual plot, fluctuating around zero and having some significant outliers.
The ACF plot shows no substantial autocorrelation, and the p-value is 0.2358 > 0.05 from the Ljung-Box test tells us that the residuals were white noise.
The ACF plot suggests a very slight sign of a normal distribution. The regressions errors are roughly near normal distribution and slightly skewed just like the other model at the 1% level.
We can see that there is no autocorrelation in the residuals from the histogram plots.

3. ETS(A,Ad,N) - In the Residual Plot, the residuals vary about zero like in ETS(A,N,N), except that there are much larger spikes around 2014. These spikes may be a little smaller than with ETS(A,N,N) because the model is dampening the trend and is allowing for some of the

trend error to decrease. Also, there is no clear trend in the residuals which is a good thing. The ACF Plot is useful here again because most of the lags are within the blue confidence bounds with no meaningful autocorrelation in the residuals making it look like white noise In the histogram the residuals again are approximately normally distributed with slight skewness, similar to ETS(A,N,N) model.
We do not see any significant spikes in the ACF and also the Ljung-Box p-value is > 0.05, suggesting these residuals are uncorrelated.

4. Holt-Winters Additive (ETS(A,A,A))
In the Fitted Values Plot the red line (fitted values) follows the overall downward trend like drop in 2014 but smooths out much of the data's volatility and medium-term cyclical fluctuations. In the Residual Plot, residuals show clear patterns, large positive residuals early and negative residuals later, reflecting the model's systematic failure to fit the data.
In the ACF Plot several lags exceed the blue confidence bounds like at lags 1, 2 indicating significant autocorrelation.
PACF Plot shows similar significant spikes, further confirming autocorrelation in residuals. Here Residuals are not white noise as there is significant autocorrelation, and the fitted values show systematic deviations. Thus Holt-Winters Additive is inadequate for this data due to the lack of seasonality and its inability to capture the data's volatility and cyclical patterns.

Part 8

```r
n <- length(imputed_oil_ts)
train_size <- floor(0.9 * n)
train_data <- imputed_oil_ts[1:train_size]
test_data <- imputed_oil_ts[(train_size + 1):n]

h <- length(test_data)
```

```r
ann_ets <- ets(train_data, model = "ANN")

fc_ann <- forecast(ann_ets, h = h)

errors_ann <- test_data - fc_ann$mean
rmse_ann <- sqrt(mean(errors_ann^2))
mae_ann <- mean(abs(errors_ann))
mape_ann <- mean(abs(errors_ann / test_data)) * 100
me_ann <- mean(errors_ann)

naive_errors <- diff(train_data)[1:(length(train_data)-1)]
scale <- mean(abs(naive_errors))
mase_ann <- mae_ann / scale

aicc_ann <- ann_ets$aicc
```

```r
aan_ets <- ets(train_data, model = "AAN")

fc_aan <- forecast(aan_ets, h = h)

errors_aan <- test_data - fc_aan$mean
rmse_aan <- sqrt(mean(errors_aan^2))
```

```r
mae_aan <- mean(abs(errors_aan))
mape_aan <- mean(abs(errors_aan / test_data)) * 100
me_aan <- mean(errors_aan)
naive_errors <- diff(train_data)[1:(length(train_data)-1)]
scale <- mean(abs(naive_errors))
mase_aan <- mae_aan / scale
aicc_aan <- aan_ets$aicc
```

```r
aadn_ets <- ets(train_data, model = "AAN", damped = TRUE)

fc_aadn <- forecast(aadn_ets, h = h)

errors_aadn <- test_data - fc_aadn$mean
rmse_aadn <- sqrt(mean(errors_aadn^2))
mae_aadn <- mean(abs(errors_aadn))
mape_aadn <- mean(abs(errors_aadn / test_data)) * 100
me_aadn <- mean(errors_aadn)
naive_errors <- diff(train_data)[1:(length(train_data)-1)]
scale <- mean(abs(naive_errors))
mase_aadn <- mae_aadn / scale
aicc_aadn <- aadn_ets$aicc
```

```r
hw_model <- HoltWinters(imputed_oil_ts, seasonal = "additive")

fc_hw <- forecast(hw_model, h = h)

errors_hw <- test_data - fc_hw$mean
rmse_hw <- sqrt(mean(errors_hw^2))
mae_hw <- mean(abs(errors_hw))
mape_hw <- mean(abs(errors_hw / test_data)) * 100
me_hw <- mean(errors_hw)
naive_errors <- diff(train_data)[1:(length(train_data)-1)]
scale <- mean(abs(naive_errors))
mase_hw <- mae_hw / scale

n <- length(imputed_oil_ts)
k <- 3 + frequency(imputed_oil_ts)
sse <- sum(residuals(hw_model)^2)
log_lik <- -n/2 * (log(2 * pi) + log(sse/n) + 1)
aic <- -2 * log_lik + 2 * k  # AIC
aicc_hw <- aic + (2 * k * (k + 1)) / (n - k - 1)
```

```r
metrics_table <- data.frame(
  Model = c("ETS(A,N,N)", "ETS(A,A,N)", "ETS(A,Ad,N)", "Holy-Winters"),
  RMSE = c(rmse_ann, rmse_aan, rmse_aadn, rmse_hw),
  MAE = c(mae_ann, mae_aan, mae_aadn, mae_hw),
  MAPE = c(mape_ann, mape_aan, mape_aadn, mape_hw),
  ME = c(me_ann, me_aan, me_aadn, me_hw),
  MASE = c(mase_ann, mase_aan, mase_aadn, mase_hw),
  AICc = c(aicc_ann, aicc_aan, aicc_aadn, aicc_hw)
)
print(metrics_table)
```

```
##          Model      RMSE       MAE      MAPE         ME      MASE     AICc
## 1   ETS(A,N,N)  2.467000  1.969859  4.073964  0.6237022  2.121357 8099.984
## 2   ETS(A,A,N)  3.927298  3.323798  6.795486  3.2083447  3.579425 8102.799
## 3  ETS(A,Ad,N)  3.150155  2.571321  5.226341  2.2554688  2.769076 8102.929
## 4 Holy-Winters 16.098853 13.350858 27.353083 12.3945754 14.377646 5518.061
```

ETS(A,N,N) is the best model by all performance metrics with the minimum RMSE (2.4670),
along with MAE, MAPE, ME, MASE, and AICc. It has the best trade-off between accuracy and bias.