#Fit Holt Winters Model

DemandDataApplesHWModel1 = HoltWinters(TrainDemandDataApples, alpha = 0.2, beta = F, gamma = F)

#Predict using Holt Winters Model

DemandDataApplesHWModel1.Predict = predict(DemandDataApplesHWModel1, n.ahead = 41, prediction.interval = T)

#Plot the prediction

plot.ts(TrainDemandDataApples, xlim = c(1,145))

lines(DemandDataApplesHWModel1$fitted[,1], col = "blue")

lines(DemandDataApplesHWModel1.Predict[,1], col = "red")

lines(DemandDataApplesHWModel1.Predict[,2], col = "yellow")

lines(DemandDataApplesHWModel1.Predict[,3], col = "orange")

#Check Accuracy

sqrt(mean((DemandDataApplesHWModel1.Predict[,1] - TestDemandDataApples)^2))

MAPE(DemandDataApplesHWModel1.Predict[,1],TestDemandDataApples)

RMSE = 61.38

MAPE = 22.48

Holt Winters with Trend

#Holt Winters Model with Trend

DemandDataApplesHWModel2 = HoltWinters(TrainDemandDataApples, alpha = 0.2, beta = 0.3, gamma = F)

#Predict using Holt Winters Model

DemandDataApplesHWModel2.Predict = predict(DemandDataApplesHWModel2, n.ahead = 41, prediction.interval = T)

#Plot the prediction

plot.ts(TrainDemandDataApples, xlim = c(1,145))

lines(DemandDataApplesHWModel2$fitted[,1], col = "blue")

lines(DemandDataApplesHWModel2.Predict[,1], col = "red")

lines(DemandDataApplesHWModel2.Predict[,2], col = "yellow")

lines(DemandDataApplesHWModel2.Predict[,3], col = "orange")

#Check Accuracy

sqrt(mean((DemandDataApplesHWModel2.Predict[,1] - TestDemandDataApples)^2))

MAPE(DemandDataApplesHWModel2.Predict[,1],TestDemandDataApples)

RMSE = 79.69629

MAPE = 27.97346

ARIMA Model

#Auto Arima Model

ArimaApples1 = auto.arima(TrainDemandDataApples, trace = T, ic = "bic")

summary(ArimaApples1)

confint(ArimaApples1)

#Check Residuals

plot(ArimaApples1$residuals)

```
#Check ACF and PACF plots of the Train data
acf(diff(TrainDemandDataApples))
pacf(diff(TrainDemandDataApples))

#Forecast with the Arima Model (0,1,1)
ArimaApples1.Predict = forecast(ArimaApples1, h = 41)
plot(ArimaApples1.Predict)
print(ArimaApples1.Predict)

#Check Accuracy of Arima Model (0,1,1)
sqrt(mean((ArimaApples1.Predict$mean - TestDemandDataApples)^2))
MAPE(ArimaApples1.Predict$mean,TestDemandDataApples)
```

AIC=1075.77   AICc=1075.9   BIC=1080.96

RMSE = 59.01684

MAPE = 21.79119

SIMPLE MOVING AVERAGE MODEL

```
#Simple Moving Average Model
library(smooth)
SMAModel = sma(TrainDemandDataApples, h = 41)
summary(SMAModel)
SMAModel.predict= forecast(SMAModel, h= 41)
```

```
plot(SMAModel.predict)
SMAModel.predict$forecast

#Check Accuracy
sqrt(mean((SMAModel.predict$forecast - TestDemandDataApples)^2))
MAPE(SMAModel.predict$forecast,TestDemandDataApples)
```

```
  AIC     AICc      BIC
1080.630 1080.753 1085.840
```

RMSE = 68.05288

MAPE = 24.5

Neural Network Model

```
#Neural Network Model
NNetarModel = nnetar(TrainDemandDataApples, p=12, repeats = 40, size = 30)
summary(NNetarModel)
NNetarModel.Predict = forecast(NNetarModel, h = 41)
plot(DemandDataApples$Qty, xlim = c(0,141), col = "black", type = "l", lwd = 3)
lines(NNetarModel.Predict$fitted, col = "blue", type = "l", lwd = 3)
lines(NNetarModel.Predict$mean, col = "red", lwd = 3)



#Check Accuracy
sqrt(mean((NNetarModel.Predict$mean - TestDemandDataApples)^2))
```

```
MAPE(NNetarModel.Predict$mean,TestDemandDataApples)

accuracy(NNetarModel.Predict$mean,TestDemandDataApples)

> sqrt(mean((NNetarModel.Predict$mean - TestDemandDataApples)^2))

[1] 58.37707

> MAPE(NNetarModel.Predict$mean,TestDemandDataApples)

[1] 0.242

> accuracy(NNetarModel.Predict$mean,TestDemandDataApples)

               ME     RMSE      MAE       MPE     MAPE      ACF1 Theil's U

Test set -12.08772 58.37707 47.70783 -13.44414 30.00843 0.2009425   1.03829
```

Neural Network Model (18 lags)

```
#Neural Network Model

NNetarModel = nnetar(TrainDemandDataApples, p=18, repeats = 40, size = 30)

summary(NNetarModel)

NNetarModel.Predict = forecast(NNetarModel, h = 41)

plot(DemandDataApples$Qty, xlim = c(0,141), col = "black", type = "l", lwd = 3)

lines(NNetarModel.Predict$fitted, col = "blue", type = "l", lwd = 3)

lines(NNetarModel.Predict$mean, col = "red", lwd = 3)

#Check Accuracy

sqrt(mean((NNetarModel.Predict$mean - TestDemandDataApples)^2))

MAPE(NNetarModel.Predict$mean,TestDemandDataApples)

accuracy(NNetarModel.Predict$mean,TestDemandDataApples)
```

Neural Network (15 lags)

```
#Neural Network Model
NNetarModel = nnetar(TrainDemandDataApples, p=15, repeats = 20, size = 30)
summary(NNetarModel)
NNetarModel.Predict = forecast(NNetarModel, h = 41)
plot(DemandDataApples$Qty, xlim = c(0,141), col = "black", type = "l", lwd = 3)
lines(NNetarModel.Predict$fitted, col = "blue", type = "l", lwd = 3)
lines(NNetarModel.Predict$mean, col = "red", lwd = 3)
```

```
#Check Accuracy
sqrt(mean((NNetarModel.Predict$mean - TestDemandDataApples)^2))
MAPE(NNetarModel.Predict$mean,TestDemandDataApples)
accuracy(NNetarModel.Predict$mean,TestDemandDataApples)
```

```
> sqrt(mean((NNetarModel.Predict$mean - TestDemandDataApples)^2))
[1] 55.13629
> MAPE(NNetarModel.Predict$mean,TestDemandDataApples)
[1] 0.217
> accuracy(NNetarModel.Predict$mean,TestDemandDataApples)
              ME     RMSE      MAE       MPE     MAPE       ACF1 Theil's U
Test set -11.59865 55.13629 43.41573 -13.09864 27.38196 0.08055744 0.9547129
```

FFV Total

```
#Fit Holt Winters Model
HWModel = HoltWinters(TrainFFV, alpha = 0.1, beta = 0.04, gamma= F)

#Predict using Holt Winters Model
HWModel.Predict = predict(HWModel, n.ahead = 41, prediction.interval = T)

#Plot the prediction
plot.ts(TrainFFV, xlim = c(1,145), lwd = 3)
lines(HWModel.Predict$fitted[,1], col = "blue", lwd = 3)
lines(HWModel.Predict[,1], col = "red", lwd =3 )
lines(HWModel.Predict[,2], col = "yellow", lwd = 3)
lines(HWModel.Predict[,3], col = "orange", lwd = 3)

#Check Accuracy
accuracy(HWModel.Predict, TestFFV)
```

```
> accuracy(HWModel.Predict, TestFFV)
         ME     RMSE      MAE       MPE     MAPE       ACF1 Theil's U
Test set 77.59908 4033.125 3245.118 -2.963546 14.40886 0.01577585 0.6687877
```

SMA Model

```
#SMA Model
library(smooth)
SMAModel = sma(TrainFFV,order = 18, h = 41)
summary(SMAModel)
```

```
SMAModel.predict= forecast(SMAModel, h= 41)
plot(SMAModel.predict)
SMAModel.predict$forecast
```

```
#Check Accuracy
accuracy(SMAModel.predict, TestFFV)
> accuracy(SMAModel.predict, TestFFV)
                  ME     RMSE      MAE       MPE     MAPE      MASE       ACF1 Theil's U
Training set  439.3428 3815.065 2428.233 -3.334160 15.08866 0.7340936 -0.1805095        NA
Test set     1422.6090 4678.196 3931.840  2.434277 16.27409 1.1886578  0.1739120 0.7791964
```

Neural Network Model

```
#Neural Network Model
set.seed(100)
NNetarModel = nnetar(TrainFFV, p=12, repeats = 20, size = 30)
summary(NNetarModel)
NNetarModel.Predict = forecast(NNetarModel, h = 41)
plot(DemandDataFFVTotal, xlim = c(0,141), col = "black", type = "l", lwd = 3, pch = 3)
lines(NNetarModel.Predict$fitted, col = "blue", type = "l", lwd = 3, pch = 1)
```

```
lines(NNetarModel.Predict$mean, col = "red", lwd = 3, pch = 2)

par(xpd = T)

legend(x = 10, y = 50224, title = "FFV Forecast", fill = c("blue", "red", "black"), legend = c("Fitted Values","Forecast Line","Actual Values"),  ncol = 3, cex = 0.75)


#Check Accuracy

accuracy(NNetarModel.Predict$mean,TestFFV)


> accuracy(NNetarModel.Predict$mean,TestFFV)

              ME     RMSE      MAE       MPE     MAPE       ACF1 Theil's U

Test set -232.7717 4973.444 4139.896 -4.179754 18.56874 0.07926527 0.8088902
```

Neural Network Model (18 lags)

ETS Model

```
        #ETS Model

ETSModel = ets(TrainFFV)

ETSModel.Predict = forecast(NNetarModel, h = 41)

plot(DemandDataFFVTotal, xlim = c(0,141), col = "black", type = "l", lwd = 3)

lines(ETSModel.Predict$fitted, col = "blue", type = "l", lwd = 3)
```

```
lines(ETSModel.Predict$mean, col = "red", lwd = 3)

#Check Accuracy
accuracy(ETSModel.Predict$mean,TestFFV)

#Output
ETS(M,A,N)

Call:
 ets(y = TrainFFV)

  Smoothing parameters:
    alpha = 0.0299
    beta  = 0.0139

  Initial states:
    l = 17115.8168
    b = 185.2127

  sigma:  0.1763

     AIC     AICc      BIC
2102.616 2103.255 2115.642

  ME     RMSE      MAE       MPE     MAPE      ACF1 Theil's U
Test set 315.7937 5058.038 4148.663 -2.581361 18.25278 0.0378333 0.8790792
```