**Faculty of Technology, Design & Environment**

# Assessment cover

OXFORD BROOKES UNIVERSITY

**STUDENTS, PLEASE COPY THIS PAGE AND USE AS THE COVER PAGE FOR YOUR SUBMISSION**

| | | | |
|---|---|---|---|
| Module No: | DALT7010 | Module title: | **Time Series Analysis** |

| | |
|---|---|
| Assessment title: | Coursework Section 1 |

| | |
|---|---|
| Due date and time**:** | 3 May 2024 5PM |

| | |
|---|---|
| Estimated total time to be spent on assignment: | 76 hours per student |

**LEARNING OUTCOMES**

| **On successful completion of this module, students will be able to achieve the module following learning outcomes (LOs):** *LO numbers and text copied and pasted from the module descriptor* |
|---|
| LO1 Demonstrate understanding of the dynamic nature of the interrelationships between deterministic trend, seasonality and stochasticity within time series models. |
| LO2 Define and devise autoregressive integrated moving average (ARIMA) and seasonal ARIMA (SARIMA) models and evaluate their properties. |
| LO3 Apply Box-Jenkins methodology to select appropriate models for time series data, critically evaluate the merits of outcomes and create solutions for shortcomings. |

| **Engineering Council AHEP4 LOs assessed (from S1 2022-23)** *LOs copied and pasted from the AHEP4 matrix* | |
|---|---|
| **M1** | Apply a comprehensive knowledge of mathematics, statistics, natural science and engineering principles to the solution of complex problems. Much of the knowledge will be at the forefront of the particular subject of study and informed by a critical awareness of new developments and the wider context of engineering |
| **M2** | Formulate and analyse complex problems to reach substantiated conclusions. This will involve evaluating available data using first principles of mathematics, statistics, natural science and engineering principles, and using engineering judgment to work with information that may be uncertain or incomplete, discussing the limitations of the techniques employed |
| **M3** | Select and apply appropriate computational and analytical techniques to model complex problems, discussing the limitations of the techniques employed |
| **M17** | Communicate effectively on complex engineering matters with technical and non-technical audiences, evaluating the effectiveness of the methods used |

**Statement of Compliance *(please tick to sign)***

☐ I declare that the work submitted is my own and that the work I submit is fully in accordance with the University regulations regarding assessments *(www.brookes.ac.uk/uniregulations/current)*

**Assignment, Section 1**

Introduction

The Movin data is used to plot the timeseries. As a part of this experiment, time series is being initialized at year 2011 and set the frequency as 12 to indicate monthly data.

tsdata = ts(trend, start = c(2011,1), frequency = 12)

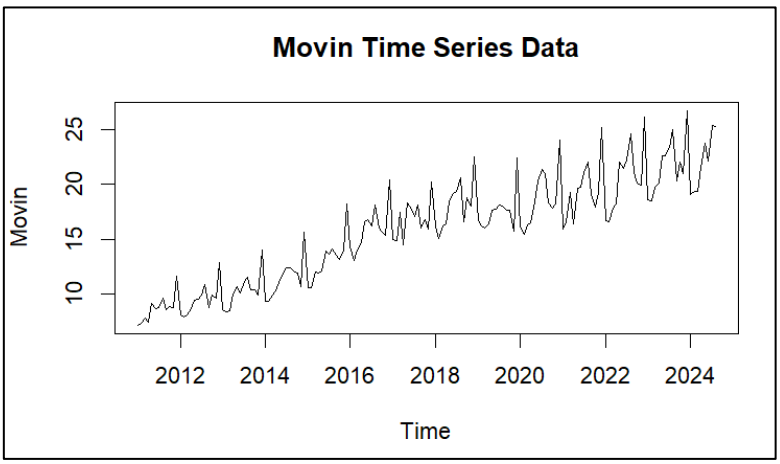


*Fig 1: Movin timeseries*

A. Trend Analysis: Fitting the timeseries with linear, polynomial and exponential

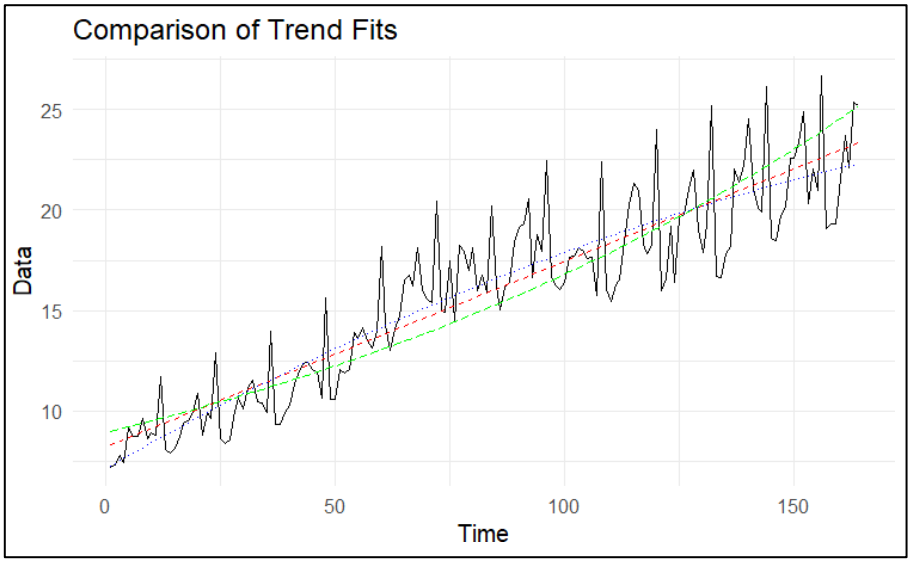


*Fig 2: Comparison of trend fits*

1. Linear model

```
> trend_model <- lm(tsdata ~ time(tsdata))
> summary(trend_model)

Call:
lm(formula = tsdata ~ time(tsdata))

Residuals:
    Min      1Q  Median      3Q     Max
-3.9586 -1.2185 -0.2157  0.9619  5.5791

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -2.217e+03  7.961e+01  -27.85   <2e-16 ***
time(tsdata)  1.107e+00  3.946e-02   28.05   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.993 on 162 degrees of freedom
Multiple R-squared:  0.8292,    Adjusted R-squared:  0.8282
F-statistic: 786.7 on 1 and 162 DF,  p-value: < 2.2e-16
```

By fitting the time series with a linear model, we can see that R – squared is coming as 82.92%, which states that our model is doing reasonably good and statistically significant.

2. Polynomial model

```
> summary(fit_poly)

Call:
lm(formula = tsdata ~ poly(time_index, 2, raw = TRUE))

Residuals:
    Min      1Q  Median      3Q     Max
-3.8616 -1.2011 -0.2531  1.1488  5.0762

Coefficients:
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                     7.143e+00  4.605e-01   15.51  < 2e-16 ***
poly(time_index, 2, raw = TRUE)1  1.312e-01  1.289e-02   10.18  < 2e-16 ***
poly(time_index, 2, raw = TRUE)2 -2.360e-04  7.564e-05   -3.12  0.00214 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.942 on 161 degrees of freedom
Multiple R-squared:  0.839,     Adjusted R-squared:  0.837
F-statistic: 419.4 on 2 and 161 DF,  p-value: < 2.2e-16
```

The R-squared value of polynomial model is slightly better than the linear model which says that the model is 83.90% accurate.

3. Exponential model

```
> summary(fit_exp)

Call:
lm(formula = log(tsdata) ~ time_index)

Residuals:
     Min       1Q   Median       3Q      Max
-0.23416 -0.10363 -0.00359  0.07763  0.37277

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 2.1896932  0.0213653  102.49   <2e-16 ***
time_index  0.0063192  0.0002246   28.13   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1362 on 162 degrees of freedom
Multiple R-squared:  0.8301,    Adjusted R-squared:  0.829
F-statistic: 791.5 on 1 and 162 DF,  p-value: < 2.2e-16
```

By looking at the exponential model after taking the log of the data, we can see that R – squared is coming as 83.01%, which is also close to the R-squared value of Linear model and polynomial model.

As per above observations it can be seen that all the models have similar R-squared values. So based on the principle of parsimony as stated by (Sober, 1981) the linear model makes and attractive choice for this trend. And also, by visually looking at the timeseries plot we can say that there exists a linear increasing trend in this timeseries and the trend is calculated as

**The equation of the trend line is: Y = 0.09222155 * Time + 8.219917**

Residuals plot

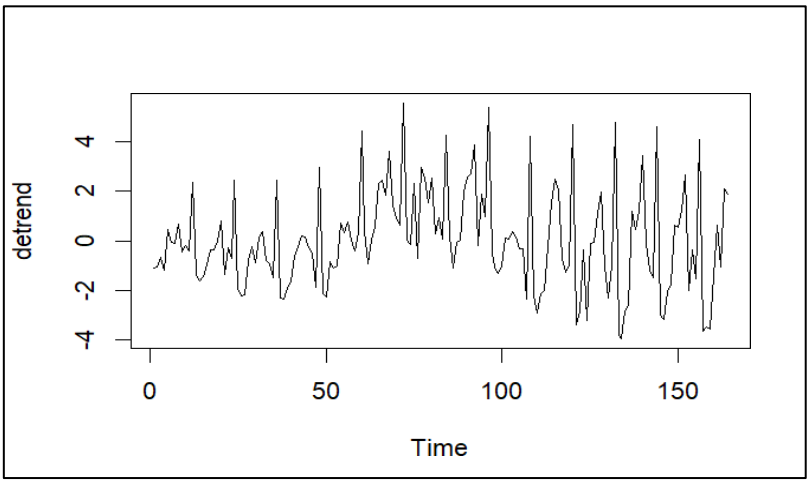


*Fig 3: Residuals plot*

Fig 3 is the residual plot which shows the measure of fluctuations. It is observed that the variance of the residuals is randomly scattered around the x-axis and do not show any pattern. This suggests homoscedasticity. Although there are some signs of seasonality in the data as there are spikes at regular intervals.

Thus, from the above trend fits and detrend, we can conclude that the timeseries has a trend as well as a seasonal component.

## B. Smoothing

- Moving Averages

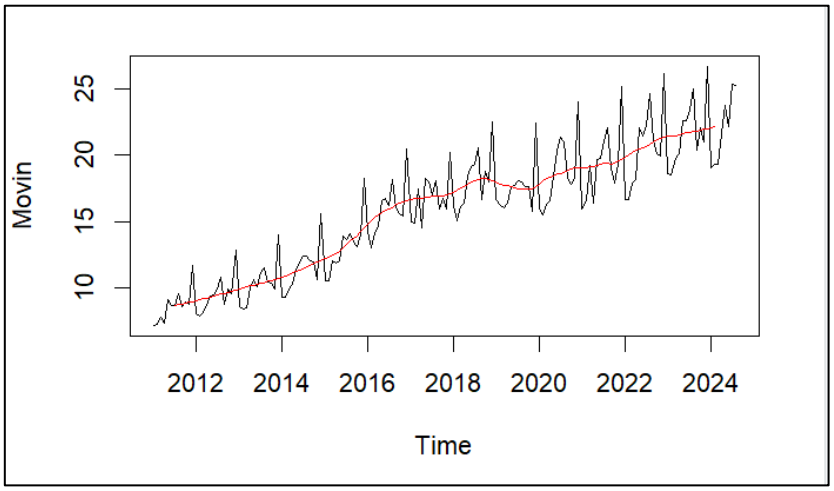


*Fig 4: Moving Average Smoothing*

The moving averages method smooths the series by averaging the datapoints. The Fig 4 shows a moving upward trend in the data but as we have both trend and seasonality in our series this method will not be appropriate.

| Model Type | Alpha | Beta | Gamma | AIC | BIC | RMSE | MAE | MASE | ACF1 |
|---|---|---|---|---|---|---|---|---|---|
| Simple Exponential Smoothing (fit1) | 0.2028 | N/A | N/A | 1075.09 | 1084.39 | 2.032951 | 1.563872 | 1.220539 | -0.0067997 |
| Holt's Linear (fit2) | 0.0207 | 0.0207 | N/A | 1066.896 | 1082.395 | 1.958756 | 1.517055 | 1.184 | 0.1179642 |
| Holt-Winters' Additive (fit3) | 0.2621 | 0.0001 | 0.3926 | 803.7638 | 856.4616 | 0.8162029 | 0.6223727 | 0.4857368 | -0.0548645 |
| Holt-Winters' Multiplicative (fit4) | 0.2 | 0.2 | 0.0133 | 795.2006 | 841.6986 | 0.8561945 | 0.6695824 | 0.5225821 | 0.1594505 |
| ETS(M,A,M) (fit5_auto) | 0.3831 | 0.0001 | 0.0005 | 738.9388 | 791.6365 | 0.7239404 | 0.5423039 | 0.4232463 | -0.1427266 |

*Table1*: *Summary statistics of smoothing techniques*

1. Simple exponential smoothing

The alpha value of 0.2028 tells us that around 20% of the weight is considered from the recent observations and 80% of it from older data. This implies to heavy smoothing.

2. Holts smoothing

In holts smoothing both the parameters alpha and beta are 0.0207 which is low that states historical observations are considered rather than the recent changes for prediction. The AIC, BIC values are high and also the error rate is more than 1.

3. Holt Winters additive method

The low value of beta says that the trend component changes at a very slow rate over time. And the seasonal smoothing coefficient gamma = 0.3926 indicating a reasonable weight on seasonal component. Also, the AIC, BIC and error rates are reasonably low as compared to the holt and ses model.

4. Holts winters multiplicative method

The AIC, BIC and RMSE values are low as compared to holt winters additive smoothing which can a better fit for our timeseries.

5. ETS smoothing

The ETS model suggests that the Error and Seasonality component are multiplicative and the trend component is additive. It also has very low beta and gamma values from which we can say that both the trend and seasonality changes very gradually. Also, the AIC, BIC and RMSE values are lowest among above.

Considering the AIC, BIC, ME and RMSE values for the above models the lowest values are given by the ETS model as highlighted in the table above. Thus, we can say that the ETS smoothing is the best fit or our data.

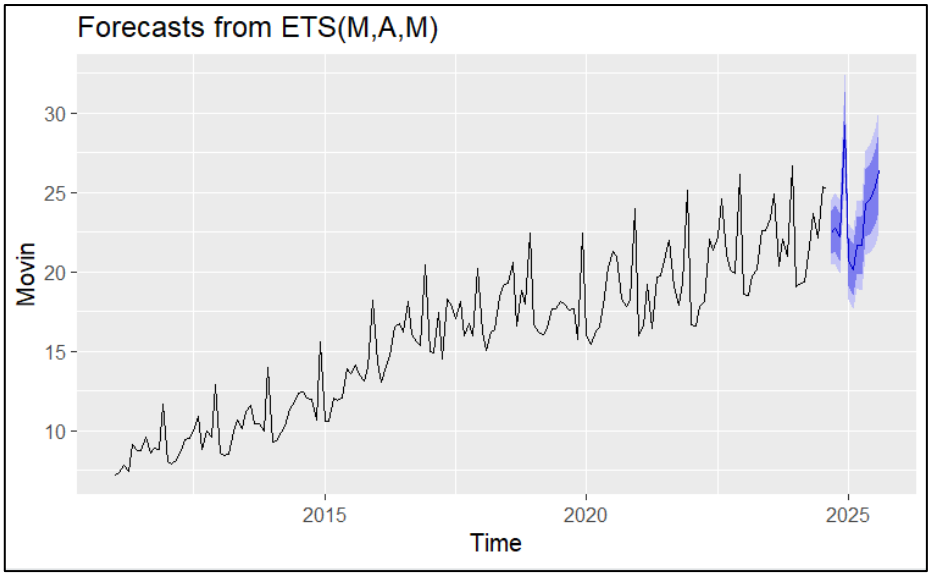


*Fig 5: Smoothing using ETS best fit*

Fig 5 shows ETS smoothing with 1 year of forecasting.

## C. Seasonal adjustment and Forecasting

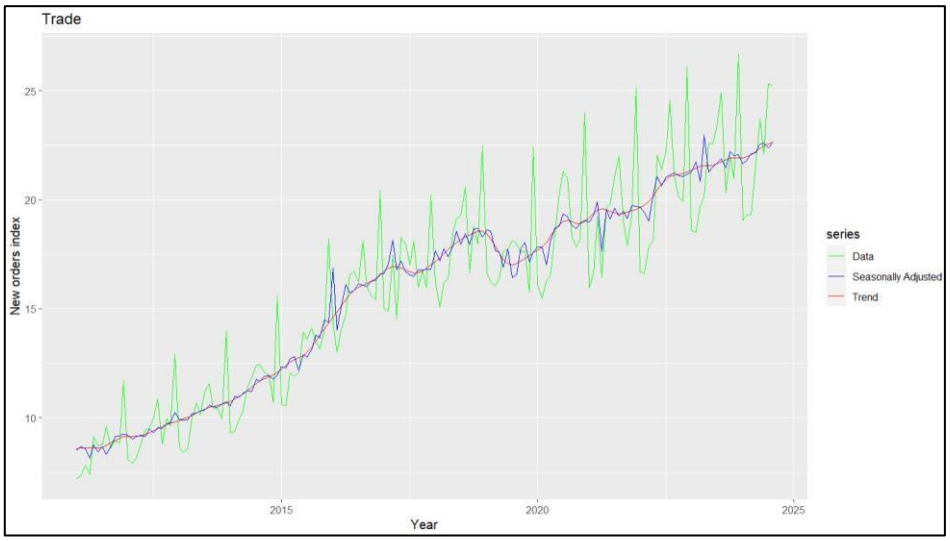


*Fig 6: Seasonal adjustment of Movin timeseries*

Fig 6 shows the trend and seasonally adjusted data with respect to the original time series with the red line indicating adjusted trend by moving average and green line representing the seasonally adjusted data.

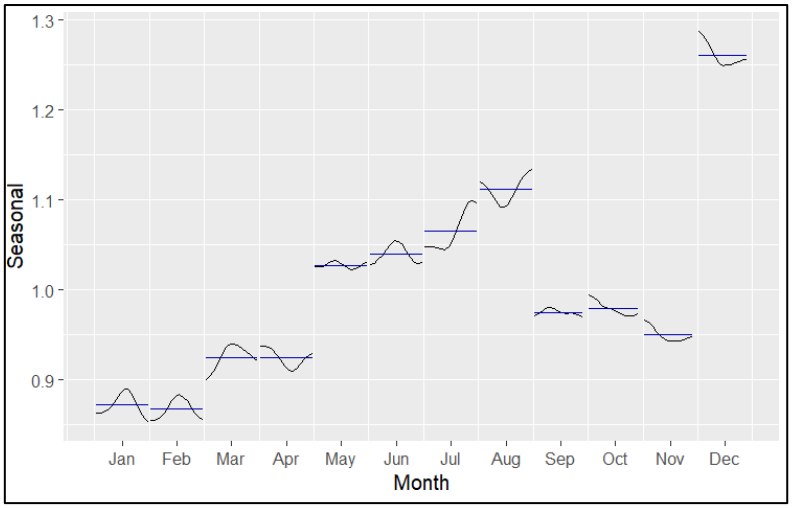


*Fig 7: Month wise seasonal adjustment*

Fig 7 indicates that there are consistent seasonal patterns, but inter-year variability is also observed. There is seasonal effect that repeats every year, although the magnitude of this effect is not constant.

Forecasting the time series one year ahead

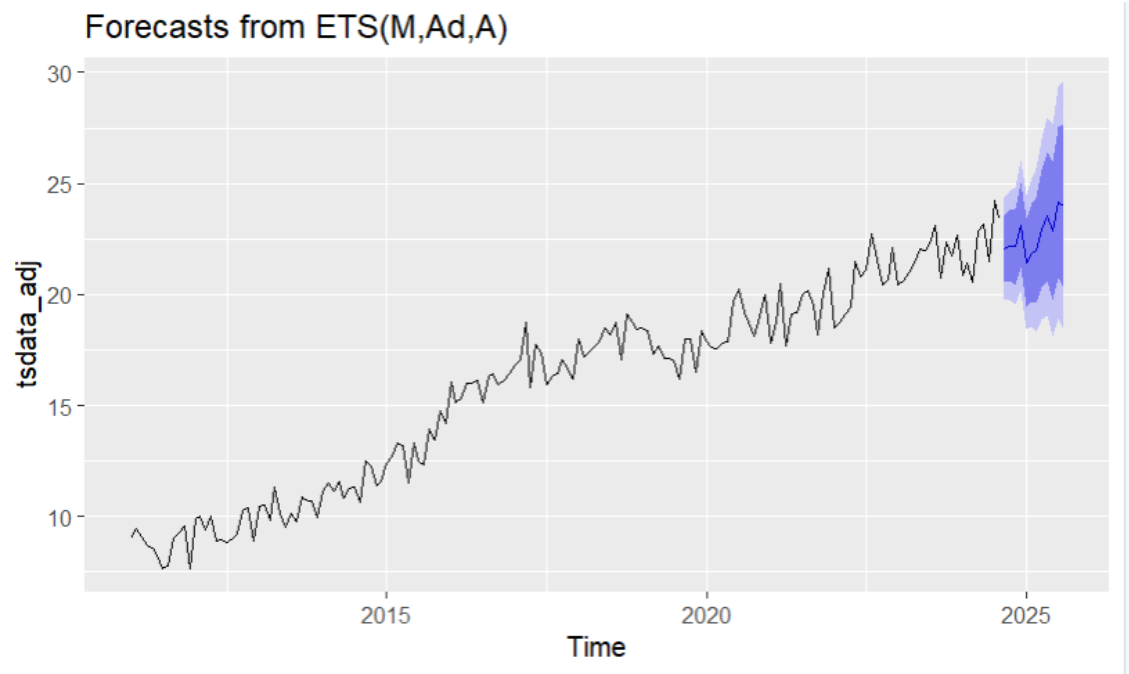


*Fig 8: Forecasts of the seasonally adjusted data*

The fig 8 shows the forecast one-year forecast of the seasonally adjusted data and as per the forecast the trend is upward. The lower confidence level) and upper confidence level) which gives us the upper bound and lower bound of the predicted values which is wide which says that there is uncertainty in the forecasts.

References

Sober, E. (1981) 'The principle of parsimony', *The British Journal for the Philosophy of Science,* 32(2), pp. 145-156.

Appendix

```
library(readxl)
library(forecast)
library(seasonal)
library(seasonalview)
library(ggplot2)
library(lmtest)

#code to load the dataset
trend <- read_xlsx("C:/Users/Ashish Khatavkar/Downloads/Individualized datasets for Section 1 and
2/Individualized datasets for Section 1 and 2/Ashish.xlsx")

#code to convert the series into time series
tsdata = ts(trend, start = c(2011,1), frequency = 12)
plot(tsdata, main = "Movin Time Series Data")

time <- time(tsdata)
#code to fit Linear Model
fit_linear <- lm(tsdata ~ time)

#code to fit Polynomial Model
fit_poly <- lm(tsdata ~ poly(time, 2, raw = TRUE))

#code to check non positive values
if(any(trend <= 0)) {
  stop("Data contains non-positive values, which are unsuitable for logarithmic transformation.")
}

#code to fit Exponential Model
fit_exp <- lm(log(tsdata) ~ time)

#code to prepare data for ggplot to plot above 3 models
data_frame <- data.frame(
  Time = time,
  Data = tsdata,
  Linear_Pred = predict(fit_linear),
  Poly_Pred = predict(fit_poly),
  Exp_Pred = exp(predict(fit_exp))
)

#code to plot the graph
ggplot(data_frame, aes(x = Time)) +
  geom_line(aes(y = tsdata)) +  # Changed 'size' to 'linewidth'
  geom_line(aes(y = Linear_Pred), color = "red", linetype = "dashed") +
  geom_line(aes(y = Poly_Pred), color = "blue", linetype = "dotted") +
  geom_line(aes(y = Exp_Pred), color = "green", linetype = "longdash") +
  labs(title = "Comparison of Trend Fits", x = "Time", y = "Data") +
  theme_minimal() +
  theme(legend.position = "top") +
  scale_color_manual(values = c("Data" = "black", "Linear" = "red", "Polynomial" = "blue", "Exponential"
= "green"))
```

```
#code to display model summaries for comparison
summary(fit_linear)
summary(fit_poly)
summary(fit_exp)

#code to create a numerical time index that starts at 1 for the first observation to get the equation
start_year <- start(tsdata)[1]
years_since_start <- floor(time(tsdata)) - start_year
months_since_start <- cycle(tsdata) + years_since_start * frequency(tsdata)
time_index <- months_since_start

# Fit the linear regression model
trend_model_for_equation <- lm(tsdata ~ time_index)

# Summary of the model to see coefficients
summary(trend_model_for_equation)

################################################################################

#code to plot the residuals plot
detrend=residuals(trend_model)
plot.ts(detrend)

#smoothing
#calculate moving average
plot(tsdata)
tsm <- ma(tsdata,order=12,centre = TRUE)
lines(tsm,col="red")
summary(tsm)


#simple exponential smoothing
ses <- ses(tsdata,h=12, alpha = NULL)
autoplot(ses, main = "Simple exponential smoothing", xlab = "Month", ylab = "Trade")
summary(ses)

#holt smoothing
holt <- holt(tsdata,h=12)
autoplot(holt, xlab = "Month", ylab = "Trade")
summary(holt)

#holt winter additive
hwa <- hw(tsdata,h=12)
autoplot(hwa, xlab = "Month", ylab = "Trade")
summary(hwa)

#holt winter multiplicative
hwm <- hw(tsdata,h=12, alpha=.2, beta=.2, level=c(95),
        seasonal = "multiplicative")
autoplot(hwm, xlab = "Month", ylab = "Trade")
summary(hwm)
```

```
#code to plot ETS smoothing
ets <- ets(tsdata)
autoplot(ets)
summary(ets)

accuracy(ses)
accuracy(holt)
accuracy(hwa)
accuracy(hwm)
accuracy(ets)

#code to decompose the timeseries
tsdata_dec <- tsdata %>% decompose(type="multiplicative")%>%
  autoplot() + xlab("Month") +
  ggtitle("Multiplicative Decomposition of Movin Timeseries")

#X11 decomposition
fit <- tsdata %>% seas(x11="")
autoplot(fit) + ggtitle("X11 decomposition of Movin")

fit %>% seasonal() %>% ggsubseriesplot() + ylab("Seasonal")

#Forecasting with decomposition
tsdata_uni <- ts(tsdata[,1], start=start(tsdata), frequency=frequency(tsdata))

fitstl <- stl(tsdata_uni, s.window = "periodic")
tsdata_adj <- seasadj(fitstl)

ets_model <- ets(tsdata_adj)

ets_forecast <- forecast(ets_model, h = 12)
autoplot(ets_forecast)
```