Ss2405_BAN_673_CASE_2:

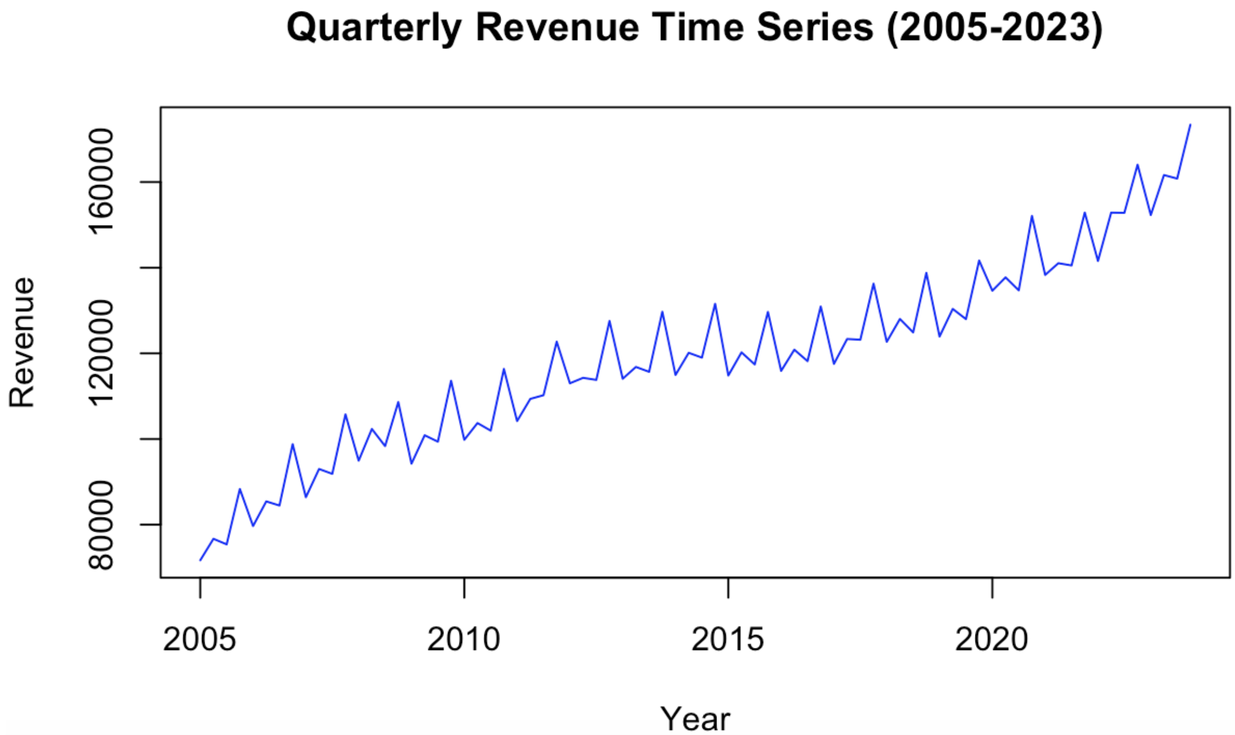
Name – Sarath Kumar, Vatyam
NET ID – ss2405

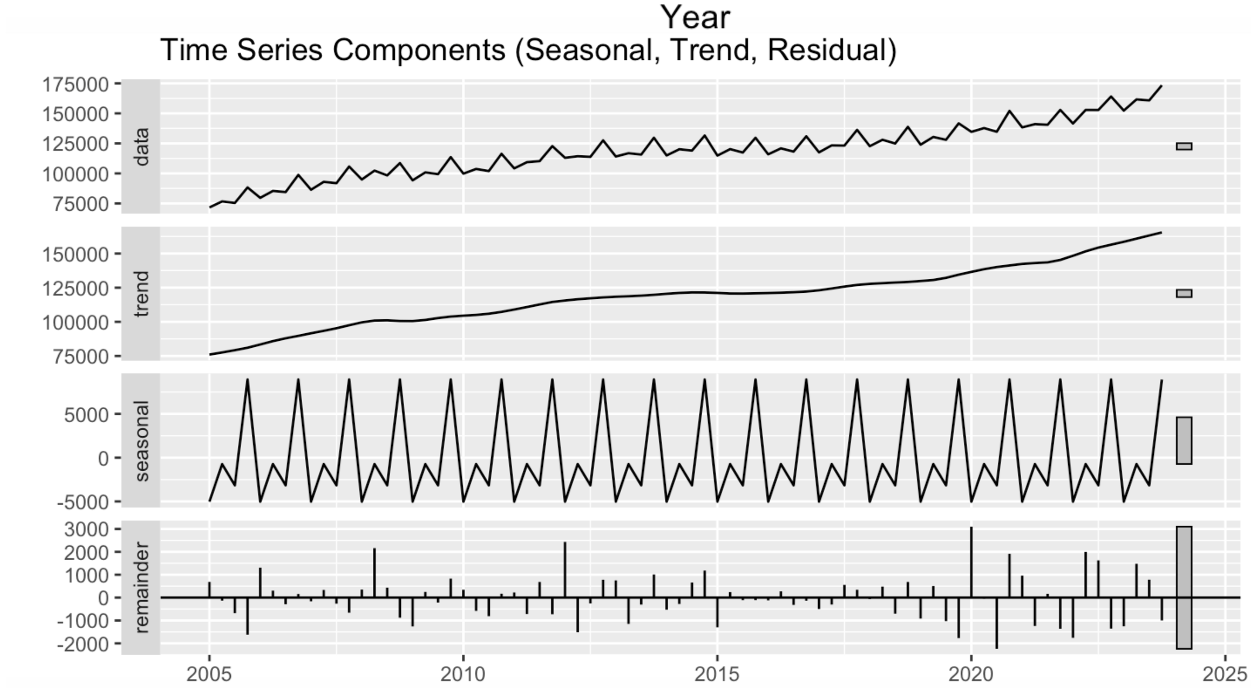
Q1A - Create time series data set in R using the ts() function.

```
> revenue <- walmart.data$Revenue
> walmart.ts <- ts(walmart.data$Revenue,
+                  start = c(2005, 1), end = c(2023, 4), freq = 4)
>
> walmart.ts
       Qtr1   Qtr2   Qtr3   Qtr4
2005  71680  76697  75397  88327
2006  79676  85430  84467  98795
2007  86410  92999  91865 105749
2008  94940 102342  98345 108627
2009  94242 100876  99373 113594
2010  99811 103726 101952 116360
2011 104189 109366 110226 122728
2012 113010 114282 113800 127559
2013 114070 116830 115688 129706
2014 114960 120125 119001 131565
2015 114826 120229 117408 129667
2016 115904 120854 118179 130936
2017 117542 123355 123179 136267
2018 122690 128028 124894 138793
2019 123925 130377 127991 141671
2020 134622 137742 134708 152079
2021 138310 141048 140525 152871
2022 141569 152859 152813 164048
2023 152301 161632 160804 173388
```

Q1B- Plotting the data using plot() function –

```
> plot(walmart.ts,
+      main = "Quarterly Revenue Time Series (2005-2023)",
+      xlab = "Year",
+      ylab = "Revenue",
+      type = "l",  # 'l' for line plot
+      col = "blue"
```





Observation –

As per above plot() function we can observe that there is an continuous upward trend in the revenue. We can also say that the revenue has been doubled from 2005 to 2023.

We can also see the case time series components using the autopilot function

Q2A-

Partitioning the data 16 periods for validation data and rest is with training partitions –

```
> # Data Partition
> nValid <- 16
> nTrain <- length(walmart.ts) - nValid
> train.ts <- window(walmart.ts, start = c(2005, 1), end = c(2019, 4))
> valid.ts <- window(walmart.ts, start = c(2020, 1), end = c(2023, 4))
> train.ts
       Qtr1   Qtr2   Qtr3   Qtr4
2005  71680  76697  75397  88327
2006  79676  85430  84467  98795
2007  86410  92999  91865 105749
2008  94940 102342  98345 108627
2009  94242 100876  99373 113594
2010  99811 103726 101952 116360
2011 104189 109366 110226 122728
2012 113010 114282 113800 127559
2013 114070 116830 115688 129706
2014 114960 120125 119001 131565
2015 114826 120229 117408 129667
2016 115904 120854 118179 130936
2017 117542 123355 123179 136267
2018 122690 128028 124894 138793
2019 123925 130377 127991 141671
> valid.ts
       Qtr1   Qtr2   Qtr3   Qtr4
2020 134622 137742 134708 152079
2021 138310 141048 140525 152871
2022 141569 152859 152813 164048
2023 152301 161632 160804 173388
```

Q2B –

Regression Model with Linear Trend

```
> train.lin <- tslm(train.ts ~ trend)
>
> # See summary of linear trend model and associated parameters.
> summary(train.lin)

Call:
tslm(formula = train.ts ~ trend)

Residuals:
     Min       1Q   Median       3Q      Max
-13713.4  -5045.9   -416.3   4058.6  15335.8

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 84527.92    1829.19   46.21   <2e-16 ***
trend         865.48      52.15   16.59   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6996 on 58 degrees of freedom
Multiple R-squared:  0.826,	Adjusted R-squared:  0.823
F-statistic: 275.4 on 1 and 58 DF,  p-value: < 2.2e-16
```

Observation –
The regression model exhibits a strong fit to the data, explaining approximately 82.6% of the variance in the dependent variable. The residual standard error, at 6996, indicates that the model's predictions typically deviate from the actual values by around 6996 units.

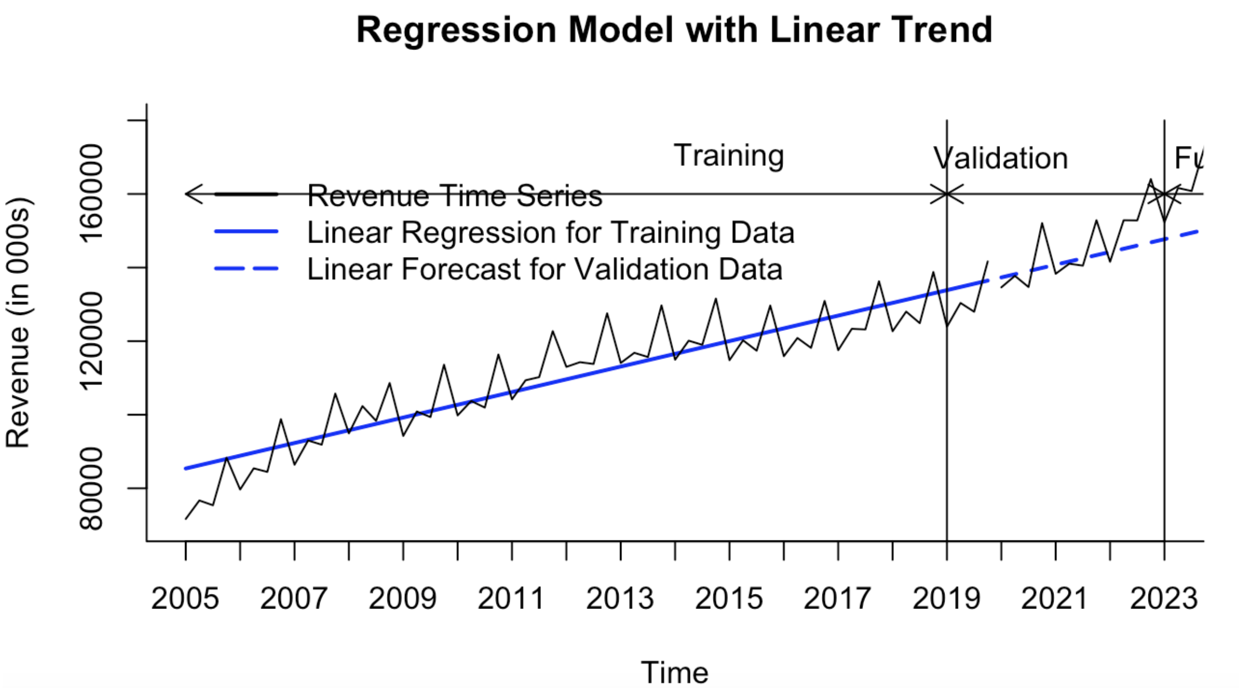
Model Equation –

y = 84527.92+865.48t (Co-efficient of intercept - 84527.92 , Co-efficient of trend - 865.48)

Forecast for the validation period using Linear Regression Model –

```
> # Apply forecast() function to make forecast for validation period.
> train.lin.pred <- forecast(train.lin, h = nValid, level = 0)
> train.lin.pred
        Point Forecast     Lo 0     Hi 0
2020 Q1       137322.1 137322.1 137322.1
2020 Q2       138187.6 138187.6 138187.6
2020 Q3       139053.0 139053.0 139053.0
2020 Q4       139918.5 139918.5 139918.5
2021 Q1       140784.0 140784.0 140784.0
2021 Q2       141649.5 141649.5 141649.5
2021 Q3       142515.0 142515.0 142515.0
2021 Q4       143380.4 143380.4 143380.4
2022 Q1       144245.9 144245.9 144245.9
2022 Q2       145111.4 145111.4 145111.4
2022 Q3       145976.9 145976.9 145976.9
2022 Q4       146842.3 146842.3 146842.3
2023 Q1       147707.8 147707.8 147707.8
2023 Q2       148573.3 148573.3 148573.3
2023 Q3       149438.8 149438.8 149438.8
2023 Q4       150304.3 150304.3 150304.3
```

Plotting Linear trend –



Observation –

From the plot the model is predicting decently but not with utmost accuracy for the validation partition in terms of revenue.

REGRESSION MODEL WITH QUADRATIC TREND –

```
> # Use tslm() function to create quadratic (polynomial) trend model.
> train.quad <- tslm(train.ts ~ trend + I(trend^2))
> # See summary of quadratic trend model and associated parameters.
> summary(train.quad)

Call:
tslm(formula = train.ts ~ trend + I(trend^2))

Residuals:
   Min     1Q Median     3Q    Max 
 -8848  -4356  -1331   5045  12581 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 76386.191   2442.424  31.275  < 2e-16 ***
trend        1653.387    184.749   8.949 1.87e-12 ***
I(trend^2)    -12.917      2.936  -4.400 4.80e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6097 on 57 degrees of freedom
Multiple R-squared:  0.8701,	Adjusted R-squared:  0.8656 
F-statistic:   191 on 2 and 57 DF,  p-value: < 2.2e-16
```

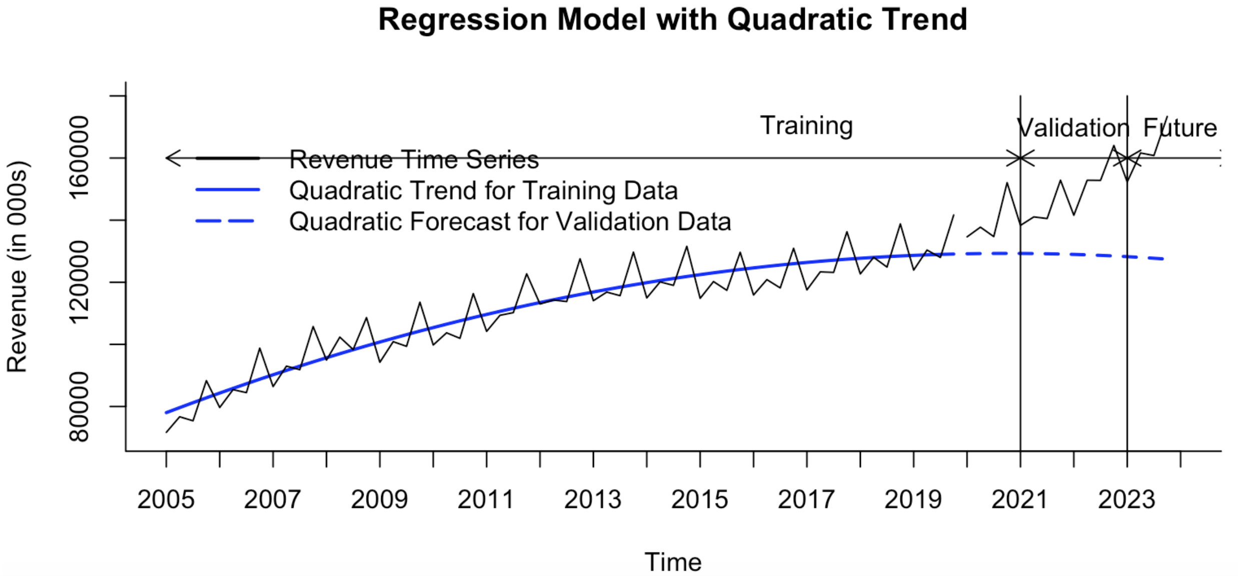
Observation –
The quadratic trend model shows a robust fit (Adj. R-squared = 0.866), signifying significant trend and curvature effects on the data, with a diminishing quadratic impact over time.

MODEL EQUATION – yt = 76386.191+1653.387t – 12.917t^2
(Estimated intercept – 76386.191 , Trend t = 1653.387 , coefficient t^2 = 12.917)

FORECAST OF VALIDATION PERIOD USING QUADRATIC REGRESSION –

```
> # Apply forecast() function to make predictions for ts data in
> # validation set.
> train.quad.pred <- forecast(train.quad, h = nValid, level = 0)
> train.quad.pred
        Point Forecast    Lo 0     Hi 0
2020 Q1       129180.4 129180.4 129180.4
2020 Q2       129245.0 129245.0 129245.0
2020 Q3       129283.8 129283.8 129283.8
2020 Q4       129296.8 129296.8 129296.8
2021 Q1       129284.0 129284.0 129284.0
2021 Q2       129245.3 129245.3 129245.3
2021 Q3       129180.8 129180.8 129180.8
2021 Q4       129090.4 129090.4 129090.4
2022 Q1       128974.3 128974.3 128974.3
2022 Q2       128832.2 128832.2 128832.2
2022 Q3       128664.4 128664.4 128664.4
2022 Q4       128470.7 128470.7 128470.7
2023 Q1       128251.2 128251.2 128251.2
2023 Q2       128005.9 128005.9 128005.9
2023 Q3       127734.7 127734.7 127734.7
2023 Q4       127437.7 127437.7 127437.7
```

Plotting Quadratic Regression Trend –



Observation –

From the plot the model is clearly doesn't look good for the validation partition where it is over predicting the revenue.

REGRESSION MODEL WITH SEASONALITY TREND –

```
> # Use tslm() function to create seasonal model.
> train.season <- tslm(train.ts ~ season)
>
> # See summary of seasonal model and associated parameters.
> summary(train.season)

Call:
tslm(formula = train.ts ~ season)

Residuals:
   Min     1Q Median     3Q    Max
-33029  -9632   5943  10617  20676

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   104525       4072  25.667   <2e-16 ***
season2         5176       5759   0.899    0.373
season3         3593       5759   0.624    0.535
season4        16831       5759   2.922    0.005 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15770 on 56 degrees of freedom
Multiple R-squared:  0.1463,    Adjusted R-squared:  0.1006
F-statistic: 3.199 on 3 and 56 DF,  p-value: 0.03017
```

Observation –

The model's coefficients indicate that only the fourth season has a significant positive effect on the target variable, while the other seasons do not show statistically significant impacts. The low adjusted R-squared value suggests that the seasonal trend alone does not explain much of the variability in the data.

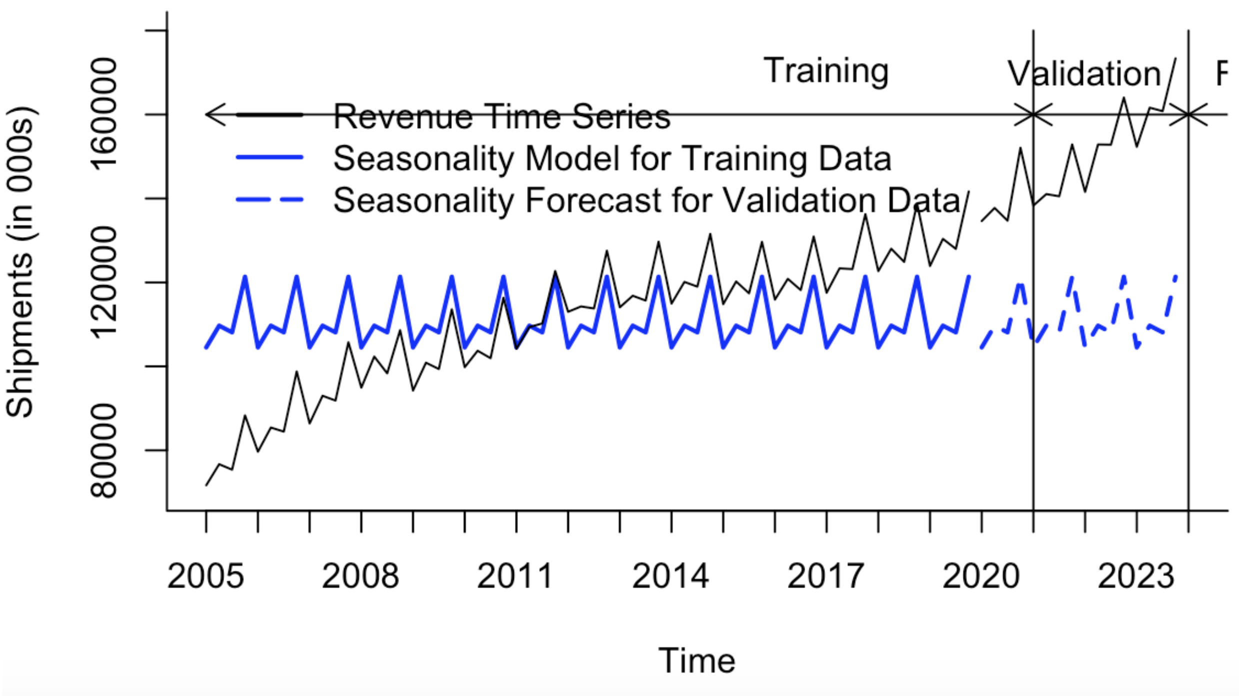
MODEL EQUATION – Yt = 104525+ 5176D2+3593D3+16831D4+e

USING FOREAST() TO CHECK SEASONALITY REGRESSION MODEL –

```
> # Apply forecast() function to make predictions for ts with
> # seasonality data in validation set.
> train.season.pred <- forecast(train.season, h = nValid, level = 0)
> train.season.pred
        Point Forecast     Lo 0     Hi 0
2020 Q1       104525.0 104525.0 104525.0
2020 Q2       109701.1 109701.1 109701.1
2020 Q3       108117.7 108117.7 108117.7
2020 Q4       121356.3 121356.3 121356.3
2021 Q1       104525.0 104525.0 104525.0
2021 Q2       109701.1 109701.1 109701.1
2021 Q3       108117.7 108117.7 108117.7
2021 Q4       121356.3 121356.3 121356.3
2022 Q1       104525.0 104525.0 104525.0
2022 Q2       109701.1 109701.1 109701.1
2022 Q3       108117.7 108117.7 108117.7
2022 Q4       121356.3 121356.3 121356.3
2023 Q1       104525.0 104525.0 104525.0
2023 Q2       109701.1 109701.1 109701.1
2023 Q3       108117.7 108117.7 108117.7
2023 Q4       121356.3 121356.3 121356.3
```

Plotting Seasonal Regression Model –

Regression Model with Seasonality



Observation –
From the plot we can see the model doesn't include the trend but only considers the seasonality.

REGRESSION MODEL WITH LINEAR TREND AND SEASONALITY:

```
> # Use tslm() function to create linear trend and seasonal model.
> train.lin.season <- tslm(train.ts ~ trend + season)
>
> # See summary of linear trend and seasonality model and associated parameter
s.
> summary(train.lin.season)

Call:
tslm(formula = train.ts ~ trend + season)

Residuals:
    Min      1Q  Median      3Q     Max
-9267.6 -3135.2   307.5  3637.7  8485.0

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 79914.73    1455.18  54.917  < 2e-16 ***
trend         848.63      32.25  26.312  < 2e-16 ***
season2      4327.44    1576.86   2.744  0.00817 **
season3      1895.41    1577.85   1.201  0.23480
season4     14285.38    1579.50   9.044  1.8e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4318 on 55 degrees of freedom
Multiple R-squared:  0.9372,    Adjusted R-squared:  0.9326
F-statistic: 205.1 on 4 and 55 DF,  p-value: < 2.2e-16
```

Observation –

The linear regression model with seasonality shows a high level of significance with an Adjusted R-squared value of 0.9326, indicating that around 93.26% of the variability in the response variable (train.ts) is explained by the predictors (trend and season).

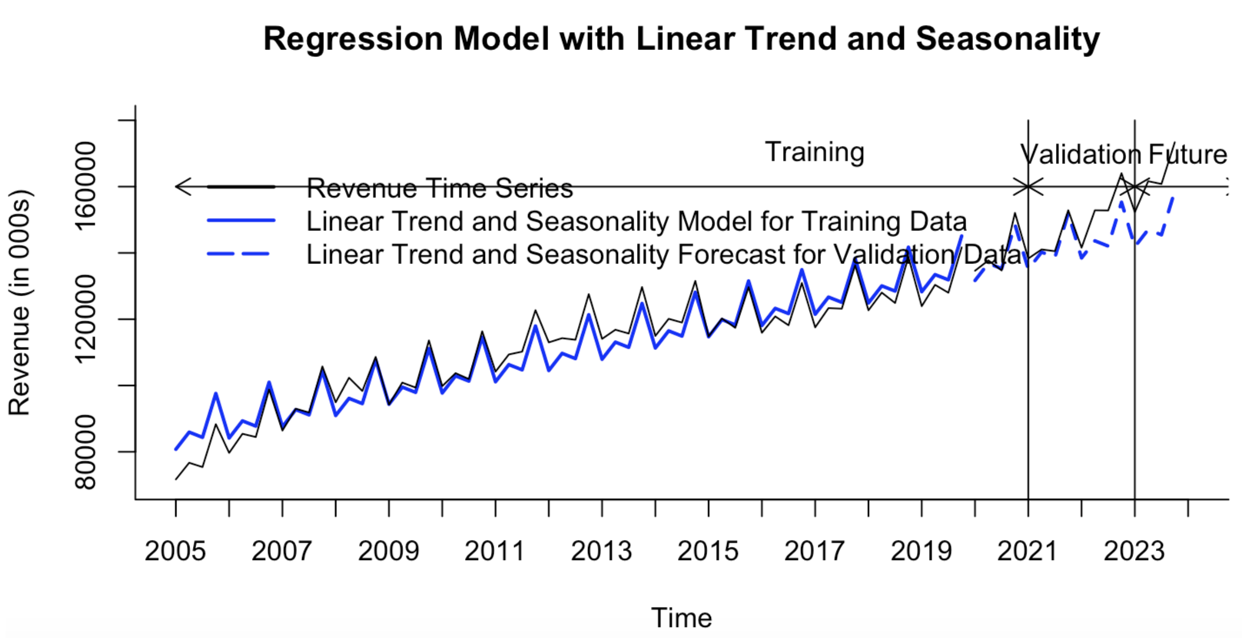
MODEL EQUATION – Yt = 79914.73+848.63t+4327.44D2+1895.41D3+14285.38D4+e

This model suggests that the intercept is approximately 79914.73, and for each unit increase in time (trend), the response variable increases by about 848.63

Forecast() for validation period using Linear Trend and Seasonality regression model:

```
> # Apply forecast() function to make predictions for ts with
> # linear trend and seasonality data in validation set.
> train.lin.season.pred <- forecast(train.lin.season, h = nValid, level = 0)
> train.lin.season.pred
        Point Forecast     Lo 0     Hi 0
2020 Q1       131681.2 131681.2 131681.2
2020 Q2       136857.2 136857.2 136857.2
2020 Q3       135273.8 135273.8 135273.8
2020 Q4       148512.4 148512.4 148512.4
2021 Q1       135075.7 135075.7 135075.7
2021 Q2       140251.8 140251.8 140251.8
2021 Q3       138668.4 138668.4 138668.4
2021 Q4       151907.0 151907.0 151907.0
2022 Q1       138470.2 138470.2 138470.2
2022 Q2       143646.3 143646.3 143646.3
2022 Q3       142062.9 142062.9 142062.9
2022 Q4       155301.5 155301.5 155301.5
2023 Q1       141864.7 141864.7 141864.7
2023 Q2       147040.8 147040.8 147040.8
2023 Q3       145457.4 145457.4 145457.4
2023 Q4       158696.0 158696.0 158696.0
>
```

Plotting the Forecast():



Observation –
The above model is pretty much accurate and able to predict the revenue with the trend as shown in the plot.

Regression model with Quadratic trend and seasonality:

```
> # Use tslm() function to create quadratic trend and seasonal model.
> train.quad.season <- tslm(train.ts ~ trend + I(trend^2) + season)
>
> # See summary of quadratic trend and seasonality model and associated parameters.
> summary(train.quad.season)

Call:
tslm(formula = train.ts ~ trend + I(trend^2) + season)

Residuals:
    Min      1Q  Median      3Q     Max 
-4072.5 -1738.4    33.4  1486.5  5873.8 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 71768.162   1072.465  66.919  < 2e-16 ***
trend        1638.260     71.710  22.846  < 2e-16 ***
I(trend^2)    -12.945      1.139 -11.362 6.15e-16 ***
season2      4301.547    864.246   4.977 6.95e-06 ***
season3      1869.517    864.788   2.162   0.0351 *  
season4     14285.376    865.688  16.502  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2366 on 54 degrees of freedom
Multiple R-squared:  0.9815,	Adjusted R-squared:  0.9798 
F-statistic:   572 on 5 and 54 DF,  p-value: < 2.2e-16

> 
```

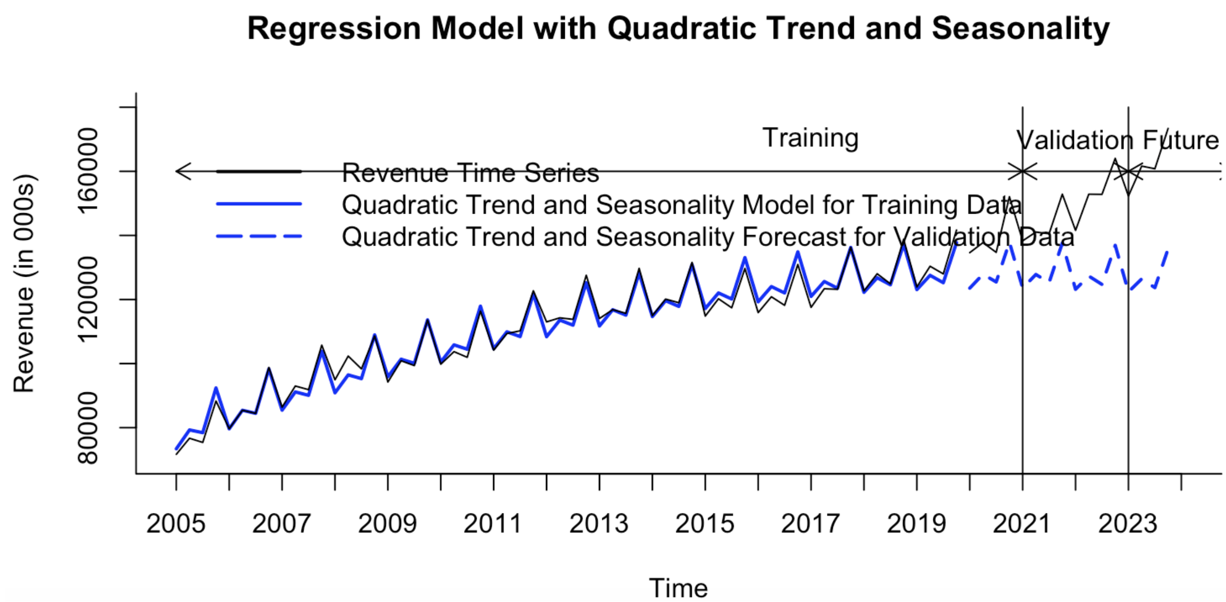
Observation –
The model combines a quadratic trend and seasonality to capture the underlying patterns in the data effectively, resulting in a high adjusted R-squared value of 0.9798.

MODEL EQUATION :
Yt =
71768.16+1638.26t-
12.95t^2+4301.55(season2)+1869.52(season3)+14285.38(season4)+e

Forecast() with Quadratic trend and Seasonality -

```
> # Apply forecast() function to make predictions for ts with
> # trend and seasonality data in validation set.
> train.quad.season.pred <- forecast(train.quad.season, h = nValid, level = 0)
> train.quad.season.pred
        Point Forecast     Lo 0     Hi 0
2020 Q1       123534.6 123534.6 123534.6
2020 Q2       127882.2 127882.2 127882.2
2020 Q3       125470.3 125470.3 125470.3
2020 Q4       137880.5 137880.5 137880.5
2021 Q1       123563.5 123563.5 123563.5
2021 Q2       127807.5 127807.5 127807.5
2021 Q3       125292.1 125292.1 125292.1
2021 Q4       137598.7 137598.7 137598.7
2022 Q1       123178.1 123178.1 123178.1
2022 Q2       127318.6 127318.6 127318.6
2022 Q3       124699.6 124699.6 124699.6
2022 Q4       136902.7 136902.7 136902.7
2023 Q1       122378.6 122378.6 122378.6
2023 Q2       126415.5 126415.5 126415.5
2023 Q3       123692.9 123692.9 123692.9
2023 Q4       135792.4 135792.4 135792.4
```

Plotting the forecast ()



Observation - This forecast plot depicts the quadratic trend and seasonality model's performance on the revenue data. Here, we see its not much in line with trend.

Q2C. Below are the summary Accuracy measures of all the models used above –

```
> round(accuracy(train.lin.pred$mean, valid.ts),3)
               ME      RMSE       MAE   MPE  MAPE  ACF1 Theil's U
Test set 5644.266 9805.521 7548.395 3.415 4.799 0.147     0.997
> round(accuracy(train.quad.pred$mean, valid.ts),3)
               ME      RMSE       MAE    MPE   MAPE  ACF1 Theil's U
Test set 20696.34 23801.48 20696.34 13.344 13.344 0.475     2.422
> round(accuracy(train.season.pred$mean, valid.ts),3)
               ME      RMSE       MAE    MPE   MAPE  ACF1 Theil's U
Test set 38532.44 39550.73 38532.44 25.558 25.558 0.838     4.007
> round(accuracy(train.lin.season.pred$mean, valid.ts),3)
               ME      RMSE       MAE   MPE  MAPE  ACF1 Theil's U
Test set 6284.492 8282.813 6355.221 4.005 4.058 0.804     0.824
> round(accuracy(train.quad.season.pred$mean, valid.ts),3)
               ME      RMSE       MAE    MPE   MAPE ACF1 Theil's U
Test set 21369.44 23464.22 21369.44 13.957 13.957 0.83     2.362
>
```

Observation –

1. The **linear trend with seasonality model achieves the lowest MAPE of 4.058**, **RMSE – 8282.813** indicating its superior accuracy in revenue forecasting compared to other models.

2. The **linear trend model,** while simpler, still demonstrates reasonable accuracy with an **MAPE of 4.799**, **RMSE – 9805.521** positioning it as a viable option for revenue forecasting.

3. The **quadratic Model** offers competitive performance, with **an MAPE of 13.344**, **RMSE – 23801.48**

Q3A –

Model with Linear Trend and Seasonality :

```
> # 1. Linear Trend with Seasonality
> lin.season <- tslm(walmart.ts ~ trend + season)
>
> # See summary of linear trend and seasonality equation and associated parameters.
> summary(lin.season)

Call:
tslm(formula = walmart.ts ~ trend + season)

Residuals:
    Min      1Q  Median      3Q     Max 
-7427.9 -4275.5   524.9  3108.0 10593.4 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 77548.36    1460.13  53.111  < 2e-16 ***
trend         940.62      25.46  36.945  < 2e-16 ***
season2      4539.38    1577.91   2.877   0.0053 ** 
season3      2115.49    1578.52   1.340   0.1845    
season4     14444.08    1579.55   9.144 1.27e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4863 on 71 degrees of freedom
Multiple R-squared:  0.9547,	Adjusted R-squared:  0.9522 
F-statistic: 374.4 on 4 and 71 DF,  p-value: < 2.2e-16
```

Observation –
1. The linear trend with seasonality model shows a strong relationship between Walmart's revenue and time, as indicated by the high R-squared value of 0.9547.
2. Seasonality seems to have a significant impact on revenue, with coefficients for seasons 2, 3, and 4 being statistically significant.

Model Equation:

$$Revenue=77548.36+940.62\times Trend+4539.38\times Season2+2115.49\times Season3+14444.08\times Season4+\epsilon$$

Forecast for Q1-Q4 of 2024-2025:

```
> lin.season.pred <- forecast(lin.season, h = 8, level = 0)
> lin.season.pred
        Point Forecast     Lo 0     Hi 0
2024 Q1       149976.4 149976.4 149976.4
2024 Q2       155456.4 155456.4 155456.4
2024 Q3       153973.1 153973.1 153973.1
2024 Q4       167242.3 167242.3 167242.3
2025 Q1       153738.8 153738.8 153738.8
2025 Q2       159218.8 159218.8 159218.8
2025 Q3       157735.6 157735.6 157735.6
2025 Q4       171004.8 171004.8 171004.8
> 
```

Linear Trend Model :

```
> lin.trend <- tslm(walmart.ts ~ trend)
>
> # See summary of linear trend equation and associated parameters.
> summary(lin.trend)

Call:
tslm(formula = walmart.ts ~ trend)

Residuals:
   Min     1Q Median     3Q    Max
-12710  -5841     70   3462  18679

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 82414.01    1703.65   48.38   <2e-16 ***
trend         951.25      38.45   24.74   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7353 on 74 degrees of freedom
Multiple R-squared:  0.8922,	Adjusted R-squared:  0.8907
F-statistic: 612.1 on 1 and 74 DF,  p-value: < 2.2e-16
```

Observation:

1. The linear trend model indicates a strong positive relationship between time and Walmart's revenue, supported by a high R-squared value of 0.8922.

2. The coefficient for the trend variable is statistically significant, suggesting that revenue has been increasing over time at a rate of $951.25 per quarter.

Model Equation:
$Revenue=82414.01+951.25\times Trend+\epsilon$

Forecast() for Q1-Q4 for 2024-2025 :

```
> # Apply forecast() function to make predictions for ts with
> # linear trend  data in 8 future periods.
> lin.trend.pred <- forecast(lin.trend, h = 8, level = 0)
> lin.trend.pred
        Point Forecast     Lo 0     Hi 0
2024 Q1       155660.2 155660.2 155660.2
2024 Q2       156611.4 156611.4 156611.4
2024 Q3       157562.7 157562.7 157562.7
2024 Q4       158513.9 158513.9 158513.9
2025 Q1       159465.2 159465.2 159465.2
2025 Q2       160416.4 160416.4 160416.4
2025 Q3       161367.7 161367.7 161367.7
2025 Q4       162318.9 162318.9 162318.9
>
```

3.QUAD TREND –

```
> #3 - QUAD Trend
> # Use tslm() function to create quadratic trend model.
> quad <- tslm(walmart.ts ~ trend + I(trend^2))
>
> # See summary of quadratic trend model and associated parameters.
> summary(quad)

Call:
tslm(formula = walmart.ts ~ trend + I(trend^2))

Residuals:
     Min       1Q   Median       3Q      Max
-12609.4  -5842.9    -15.6   3522.0  18008.2

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 8.314e+04  2.614e+03  31.809  < 2e-16 ***
trend       8.954e+02  1.567e+02   5.716 2.23e-07 ***
I(trend^2)  7.253e-01  1.972e+00   0.368    0.714
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7396 on 73 degrees of freedom
Multiple R-squared:  0.8924,    Adjusted R-squared:  0.8894
F-statistic: 302.6 on 2 and 73 DF,  p-value: < 2.2e-16
```

OBSERVATION –

The quadratic trend model equation is:

Revenue=83140+895.4×Time+0.7253×Time2

This model combines both linear and quadratic trends to capture the non-linear patterns in the data. However, the significance of the quadratic term is questionable, suggesting that the linear trend might suffice for forecasting.

Q3B.

```
> round(accuracy(lin.trend.pred$fitted, walmart.ts),3)
         ME     RMSE      MAE    MPE  MAPE  ACF1 Theil's U
Test set  0 7255.488 5912.276 -0.396 5.027 0.072     0.738
> round(accuracy(lin.season.pred$fitted, walmart.ts),3)
         ME    RMSE      MAE   MPE  MAPE  ACF1 Theil's U
Test set  0 4700.12 4017.852 -0.17 3.428 0.875     0.478
> round(accuracy(quad.pred$fitted, walmart.ts),3)
         ME     RMSE      MAE    MPE  MAPE  ACF1 Theil's U
Test set  0 7248.771 5911.592 -0.407 5.051 0.072     0.742
> round(accuracy((naive(walmart.ts))$fitted, walmart.ts), 3)
               ME     RMSE      MAE   MPE  MAPE   ACF1 Theil's U
Test set 1356.107 9705.706 8166.427 0.834 6.928 -0.709         1
> round(accuracy((snaive(walmart.ts))$fitted, walmart.ts), 3)
           ME     RMSE      MAE   MPE  MAPE  ACF1 Theil's U
Test set 4667 5863.128 4827.806 3.938 4.081 0.741     0.596
```

Observation –
Among the forecasting models evaluated for Walmart's quarterly revenue in 2024-2025, the **linear trend model with seasonality emerges as the most accurate**. It exhibits the lowest Mean Absolute Percentage Error (MAPE) at 3.428% and the smallest Root Mean Square Error (RMSE) of 4700.12. This indicates its superior performance in capturing the quarterly revenue patterns. In contrast, the naïve model and the quadratic trend model demonstrate higher errors, with MAPE values of 6.928% and 5.051%, respectively, and larger RMSE values. The seasonal naïve model also performs competitively, showing a MAPE of 4.081% and an RMSE of 5863.128, suggesting its effectiveness in leveraging seasonal patterns for forecasting. Overall, the linear trend model with seasonality proves to be the most reliable choice for forecasting Walmart's revenue during the specified period.