BAN 673 – Time Series Analytics

Final Project

Walmart Sales Prediction Across 45 stores

Submitted by-

Osheen Gupta (je7800)

1. Summary

This report presents a time series forecasting analysis of Walmart's aggregated weekly sales across 45 stores using historical data from 2010 to 2012. The goal was to identify patterns of trend and seasonality in the data and to develop forecasting models capable of accurately predicting future sales. Three models were implemented and compared: (1) a linear regression model with trend and weekly seasonality, (2) a two-level model that combined regression with a trailing moving average of residuals, and (3) an automatic ARIMA model.

Each model was evaluated using performance metrics such as RMSE, MAE, MAPE, ACF1, and Theil's U. The two-level model performed the best overall, achieving the lowest RMSE and MAPE values, while the linear regression model also delivered strong accuracy and interpretability. The ARIMA model underperformed in comparison, and a seasonal naïve model was included as a baseline reference. All models were retrained on the full dataset and used to forecast the next 12 weeks of sales. The findings demonstrate that regression-based models, especially with residual smoothing, are highly effective for forecasting aggregated retail sales.

2. Introduction

This project focuses on forecasting Walmart's weekly retail sales using time series methods. The data was obtained from Kaggle's Walmart dataset, which contains weekly sales records for 45 Walmart stores across the United States between February 2010 and October 2012. In addition to weekly sales, the dataset includes variables such as store number, date, holiday indicators, temperature, fuel price, CPI, and unemployment.

For the purpose of this project, the data was aggregated across all stores to create a univariate time series of total weekly sales. The analysis concentrates on detecting and modeling long-term trends and seasonal patterns to produce reliable short-term sales forecasts. Forecasting models were implemented and evaluated using R, with a focus on practical performance and interpretability for real-world planning applications.

Although the original dataset included additional economic variables such as unemployment rate, fuel price, and consumer price index (CPI), these were not included in the final forecasting models. A series of Pearson correlation tests were conducted to evaluate the strength and significance of the relationships between each of these variables and Walmart's aggregated weekly sales. The results showed that all three variables had very weak correlations with sales (correlation coefficients near zero) and none were statistically significant (p-values > 0.5). Given their lack of explanatory power, these external variables were excluded from the final models to maintain simplicity, interpretability, and to avoid introducing unnecessary noise.

3. Steps of Time Series Forecasting

3.1 Define Goal

The goal of this project is to forecast Walmart's aggregated weekly sales across 45 stores using univariate time series methods. The objective is to build predictive models that effectively capture underlying trend and seasonal patterns in the sales data. Forecast accuracy will be evaluated using performance metrics such as RMSE, MAE, MAPE, and ACF1 to determine the most suitable model for short-term retail forecasting. All analysis and modeling were conducted using the R programming language.

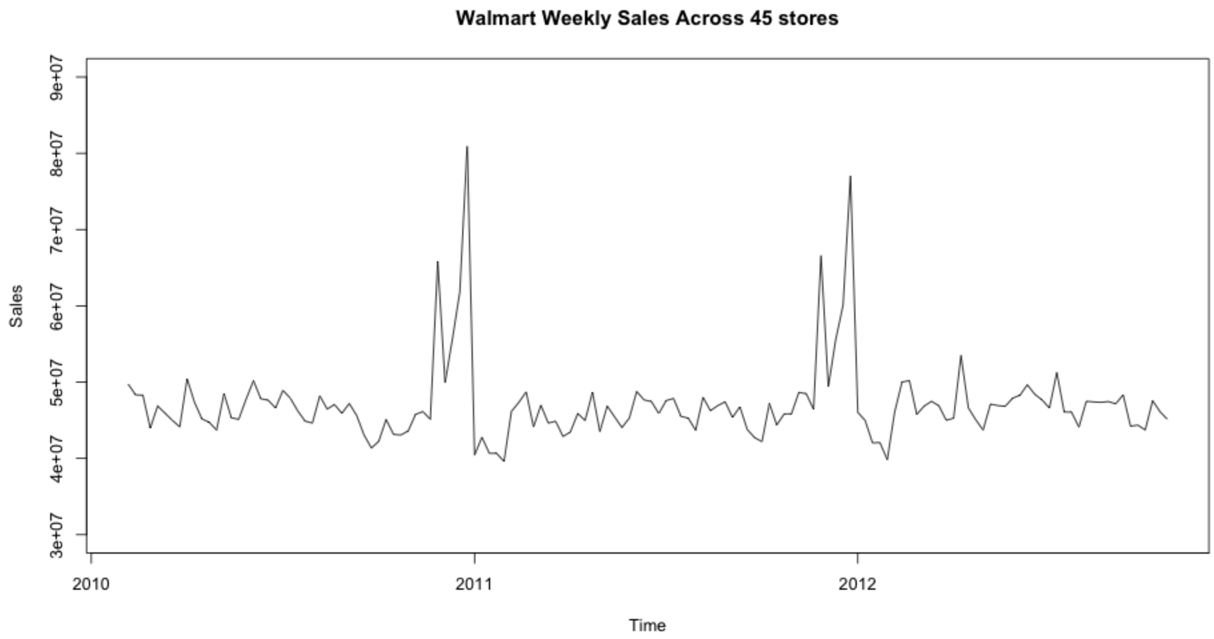
3.2 Get Data

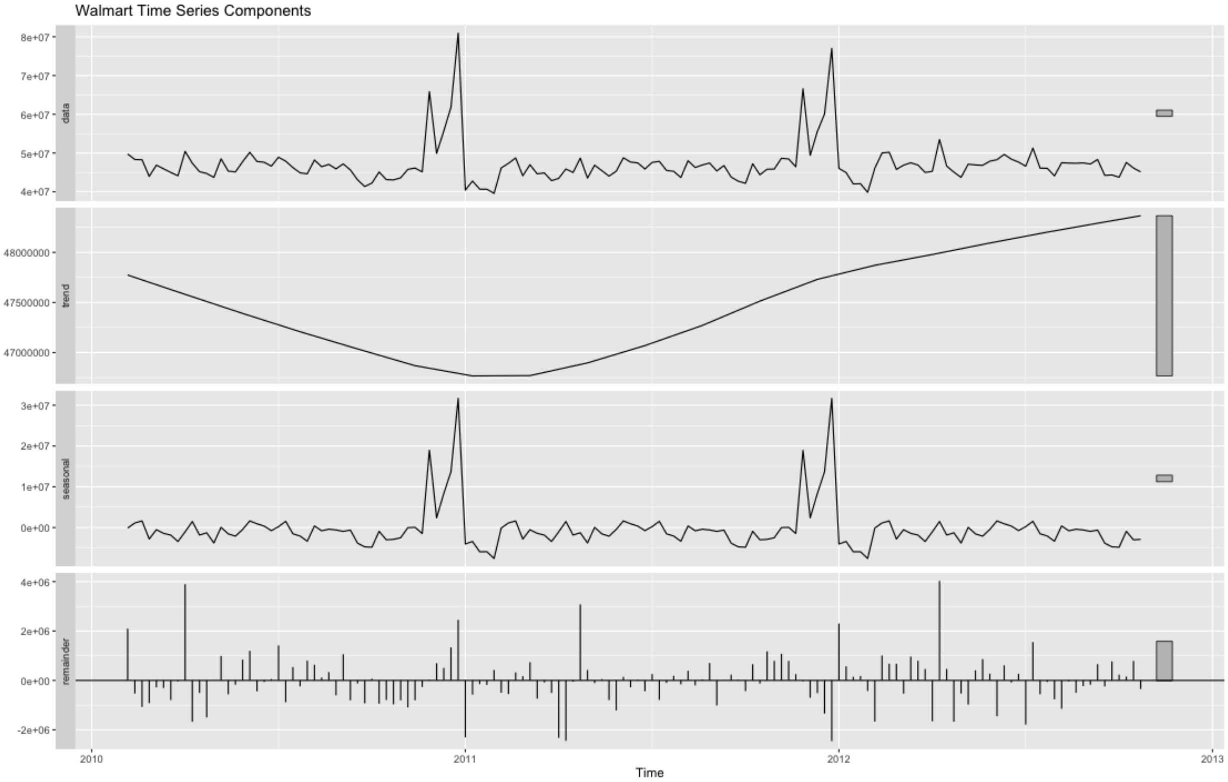
The dataset used for this project was sourced from a publicly available **Walmart Retail Sales dataset**. It contains weekly sales figures for **45 Walmart stores** across the United States, spanning from **February 2010 to October 2012**. The dataset includes variables such as Store, Date, Weekly_Sales, Holiday_Flag, Temperature, Fuel_Price, CPI, and Unemployment.

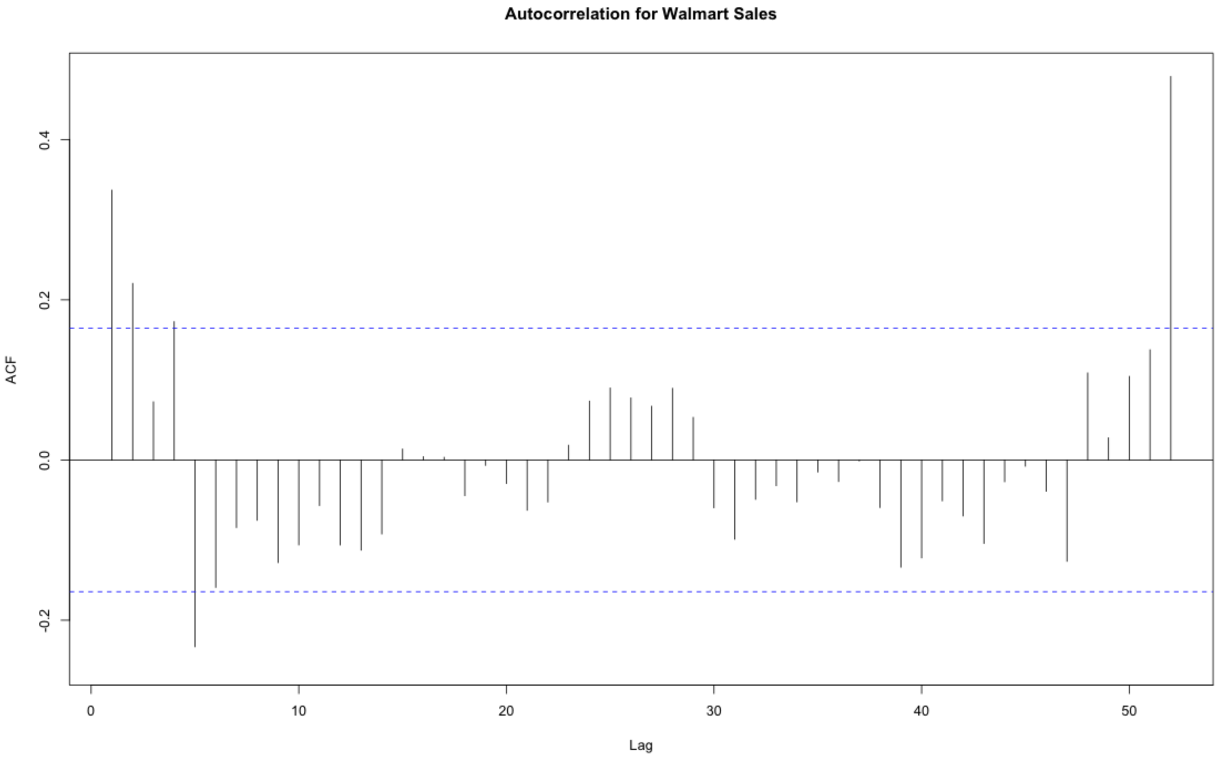
For this project, the analysis focuses solely on the aggregated weekly sales across all 45 stores. The Weekly_Sales variable was summed across all stores for each unique week to create a univariate time series for modeling. The resulting time series consists of **143 observations**, representing weekly total sales over the analysis period.

The data was imported and processed using R, with time series objects created using the ts() function and a frequency of 52 to reflect weekly data.

3.3 Explore & Visualize Series







The aggregated weekly sales time series for 45 Walmart stores was visualized to identify key patterns. As shown in the first plot, there are clear **seasonal spikes**, which align with major U.S. retail events such as Black Friday and the holiday season, typically observed in November and December. Apart from these peaks, the sales exhibit moderate variation throughout the year.

The time series decomposition plot reveals three components:

- A **trend** component that shows a slight dip followed by a gradual upward movement over time.
- A **seasonal** component with sharp recurring increases, consistent with expected retail cycles.

- A **remainder** component with relatively small random fluctuations, suggesting that most of the structure is captured by trend and seasonality.

The autocorrelation function (ACF) plot supports the presence of **seasonality**, with significant spikes at lag 1 and lag 52. These indicate strong weekly autocorrelation and an annual seasonal pattern, which justifies the use of models that incorporate both trend and seasonal components.

3.4 Pre-Process Data

The original dataset contained weekly-level sales data for 45 Walmart stores over a span of approximately three years. To prepare the data for modeling, weekly sales across all stores were **aggregated** by date to create a single univariate time series of total weekly sales. This resulted in a time series with 143 weekly observations, starting from February 2010 and ending in October 2012.

The Date column was converted to a proper Date class in R, and the time series object was constructed using the ts() function with a frequency of 52 to reflect weekly seasonality. The aggregated series was then sorted chronologically to ensure proper temporal structure.

Additionally, the data was inspected for missing values and found to be complete. This ensured that no imputation or filtering was required prior to modeling.

3.5 Partition Time Series

To evaluate model performance effectively, the aggregated time series was partitioned into a **training set** and a **validation set**. The training set was used to build the models, while the validation set was reserved for out-of-sample evaluation.

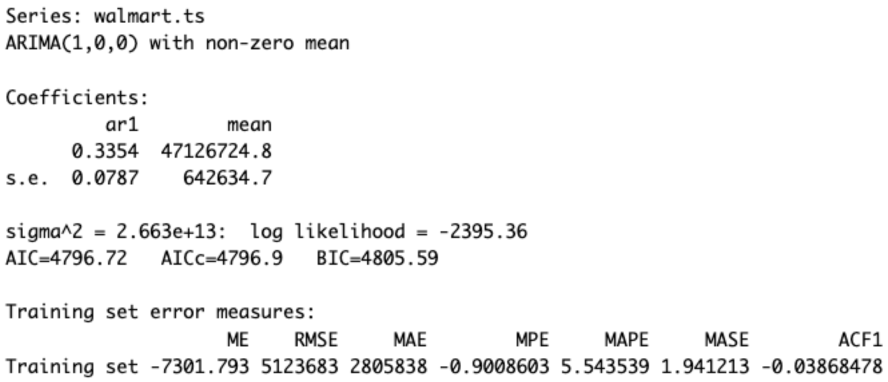
The series was split such that the first **115 weekly observations** (approximately 80% of the data) were used for training, and the **last 28 observations** (20%) were used for validation. This split was chosen to ensure that the validation period covered more than six months of sales activity, allowing the models to be tested across various seasonal points.

The window() function in R was used to create both partitions. The training data was extracted from the start of the series (February 2010) through the 115th week, and the validation data covered the remaining weeks up to October 2012.

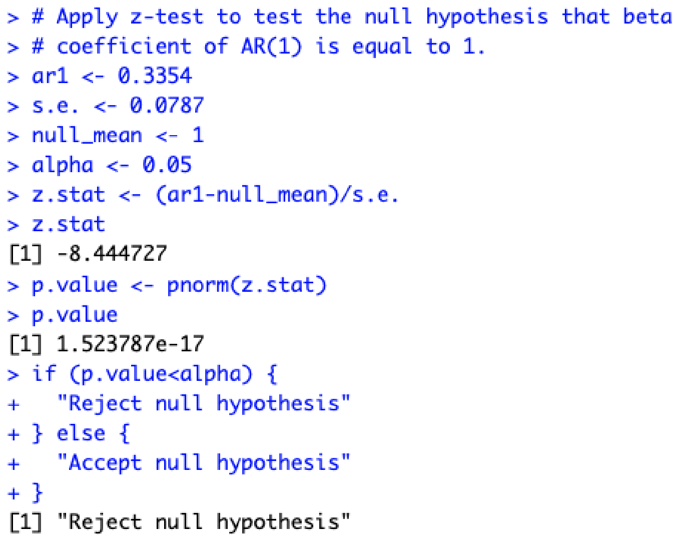
This setup ensured that the forecasting models were trained only on past data and evaluated on unseen future values, providing a fair and time-consistent assessment of forecast accuracy.

3.6 Apply Forecasting Methods

3.6.1 Check Predictability



```
Series: walmart.ts
ARIMA(1,0,0) with non-zero mean

Coefficients:
         ar1        mean
      0.3354  47126724.8
s.e.  0.0787    642634.7

sigma^2 = 2.663e+13:  log likelihood = -2395.36
AIC=4796.72   AICc=4796.9   BIC=4805.59

Training set error measures:
                   ME    RMSE     MAE        MPE     MAPE     MASE        ACF1
Training set -7301.793 5123683 2805838 -0.9008603 5.543539 1.941213 -0.03868478
```

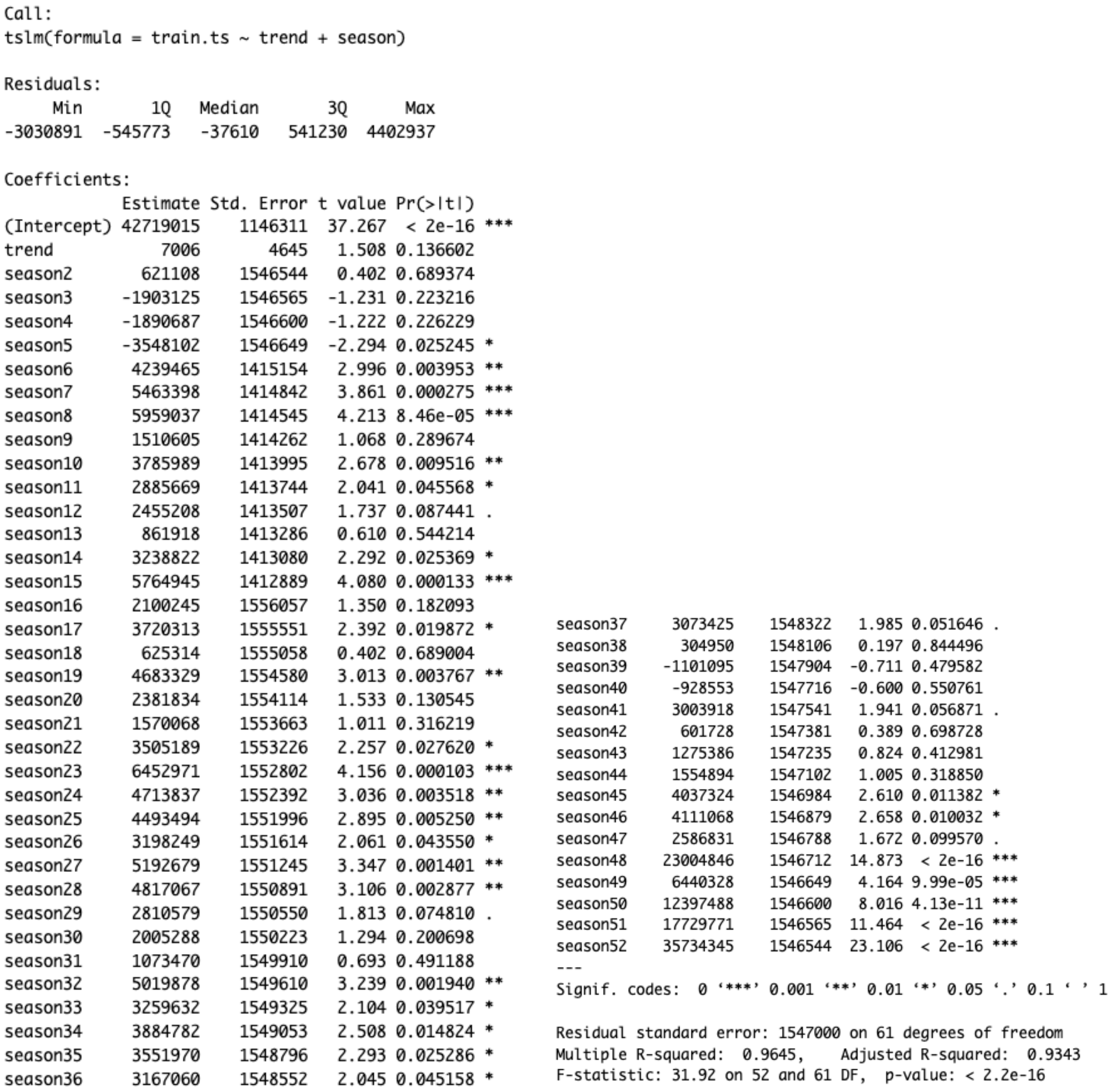


```
> # Apply z-test to test the null hypothesis that beta
> # coefficient of AR(1) is equal to 1.
> ar1 <- 0.3354
> s.e. <- 0.0787
> null_mean <- 1
> alpha <- 0.05
> z.stat <- (ar1-null_mean)/s.e.
> z.stat
[1] -8.444727
> p.value <- pnorm(z.stat)
> p.value
[1] 1.523787e-17
> if (p.value<alpha) {
+   "Reject null hypothesis"
+ } else {
+   "Accept null hypothesis"
+ }
[1] "Reject null hypothesis"
```

To validate the predictability of the series, an AR(1) model was fit to the data, and a Z-test was performed on the AR(1) coefficient. The null hypothesis tested whether the coefficient was equal to 1, which would indicate a **random walk** and lack of predictability. The test produced a z-statistic of –8.44 and a p-value near zero, leading to a **rejection of the null hypothesis**. This confirmed that the series is predictable and not a random walk.

3.6.2 Model 1: Linear Regression Model with Trend and Seasonality

The model using the summary() function is presented below:



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

Residuals:
     Min       1Q   Median       3Q      Max 
-3030891  -545773   -37610   541230  4402937 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 42719015    1146311  37.267  < 2e-16 ***
trend           7006       4645   1.508 0.136602    
season2       621108    1546544   0.402 0.689374    
season3     -1903125    1546565  -1.231 0.223216    
season4     -1890687    1546600  -1.222 0.226229    
season5     -3548102    1546649  -2.294 0.025245 *  
season6      4239465    1415154   2.996 0.003953 ** 
season7      5463398    1414842   3.861 0.000275 ***
season8      5959037    1414545   4.213 8.46e-05 ***
season9      1510605    1414262   1.068 0.289674    
season10     3785989    1413995   2.678 0.009516 ** 
season11     2885669    1413744   2.041 0.045568 *  
season12     2455208    1413507   1.737 0.087441 .  
season13      861918    1413286   0.610 0.544214    
season14     3238822    1413080   2.292 0.025369 *  
season15     5764945    1412889   4.080 0.000133 ***
season16     2100245    1556057   1.350 0.182093    
season17     3720313    1555551   2.392 0.019872 *  
season18      625314    1555058   0.402 0.689004    
season19     4683329    1554580   3.013 0.003767 ** 
season20     2381834    1554114   1.533 0.130545    
season21     1570068    1553663   1.011 0.316219    
season22     3505189    1553226   2.257 0.027620 *  
season23     6452971    1552802   4.156 0.000103 ***
season24     4713837    1552392   3.036 0.003518 ** 
season25     4493494    1551996   2.895 0.005250 ** 
season26     3198249    1551614   2.061 0.043550 *  
season27     5192679    1551245   3.347 0.001401 ** 
season28     4817067    1550891   3.106 0.002877 ** 
season29     2810579    1550550   1.813 0.074810 .  
season30     2005288    1550223   1.294 0.200698    
season31     1073470    1549910   0.693 0.491188    
season32     5019878    1549610   3.239 0.001940 ** 
season33     3259632    1549325   2.104 0.039517 *  
season34     3884782    1549053   2.508 0.014824 *  
season35     3551970    1548796   2.293 0.025286 *  
season36     3167060    1548552   2.045 0.045158 *  
season37     3073425    1548322   1.985 0.051646 .  
season38      304950    1548106   0.197 0.844496    
season39    -1101095    1547904  -0.711 0.479582    
season40     -928553    1547716  -0.600 0.550761    
season41     3003918    1547541   1.941 0.056871 .  
season42      601728    1547381   0.389 0.698728    
season43     1275386    1547235   0.824 0.412981    
season44     1554894    1547102   1.005 0.318850    
season45     4037324    1546984   2.610 0.011382 *  
season46     4111068    1546879   2.658 0.010032 *  
season47     2586831    1546788   1.672 0.099570 .  
season48    23004846    1546712  14.873  < 2e-16 ***
season49     6440328    1546649   4.164 9.99e-05 ***
season50    12397488    1546600   8.016 4.13e-11 ***
season51    17729771    1546565  11.464  < 2e-16 ***
season52    35734345    1546544  23.106  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1547000 on 61 degrees of freedom
Multiple R-squared:  0.9645,    Adjusted R-squared:  0.9343 
F-statistic: 31.92 on 52 and 61 DF,  p-value: < 2.2e-16
```

The regression model contains 52 predictors: a time trend index and 51 seasonal dummy variables representing weekly seasonality (with one baseline week omitted). The model equation takes the general form:

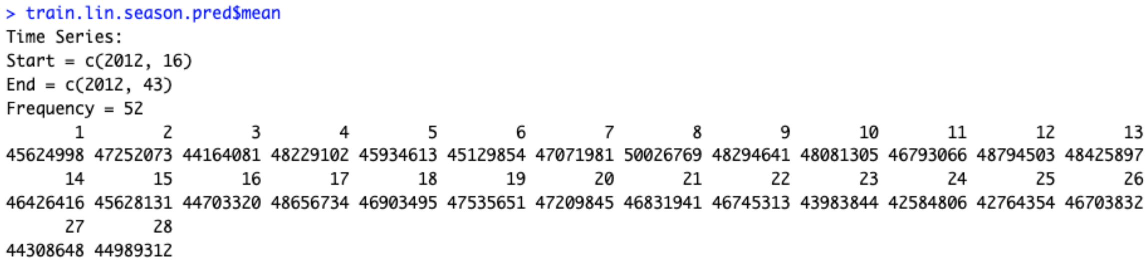
$$\mathbf{y_t = 427191015 + 7006\,t + 621108\,D_2 - 1903125\,D_3 - 1890687\,D_4 - 3548102\,D_5 + 4239465\,D_6 + \cdots + 35734345\,D_{52}}$$

Here:

- D_i is a dummy variable for week *i* (e.g., $D_2 = 1$ if week 2, else 0)
- The baseline (omitted) category is **season1 (week 1)**

The regression summary shows a **Multiple R-squared** of **0.9645** and an **Adjusted R-squared** of **0.9343**, indicating that the model explains approximately 93% of the variance in the training data. The overall model is highly statistically significant with a **p-value < 2.2e-16.** This regression model is a very good fit and statistically significant and thus can be applied for time series forecasting.

The forecast for the validation period is the following:

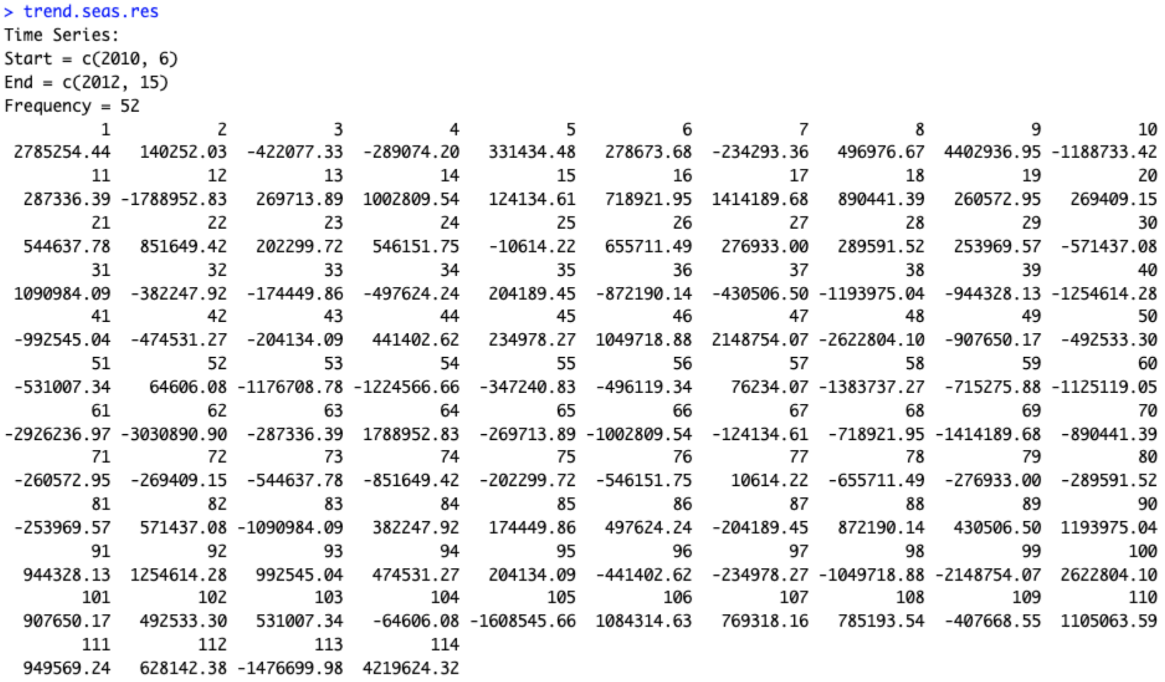


```
> train.lin.season.pred$mean
Time Series:
Start = c(2012, 16)
End = c(2012, 43)
Frequency = 52
       1        2        3        4        5        6        7        8        9       10       11       12       13 
45624998 47252073 44164081 48229102 45934613 45129854 47071981 50026769 48294641 48081305 46793066 48794503 48425897 
      14       15       16       17       18       19       20       21       22       23       24       25       26 
46426416 45628131 44703320 48656734 46903495 47535651 47209845 46831941 46745313 43983844 42584806 42764354 46703832 
      27       28 
44308648 44989312 
```

3.6.3 Model 2: Two-Level Forecasting (Regression + Trailing Moving Average on Residuals)

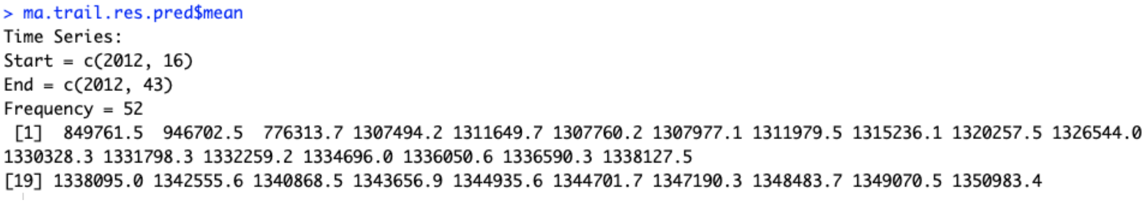
To enhance the predictive accuracy of the linear regression model with trend and seasonality, a two-level forecasting approach was applied.

Residuals from the regression model are presented below:



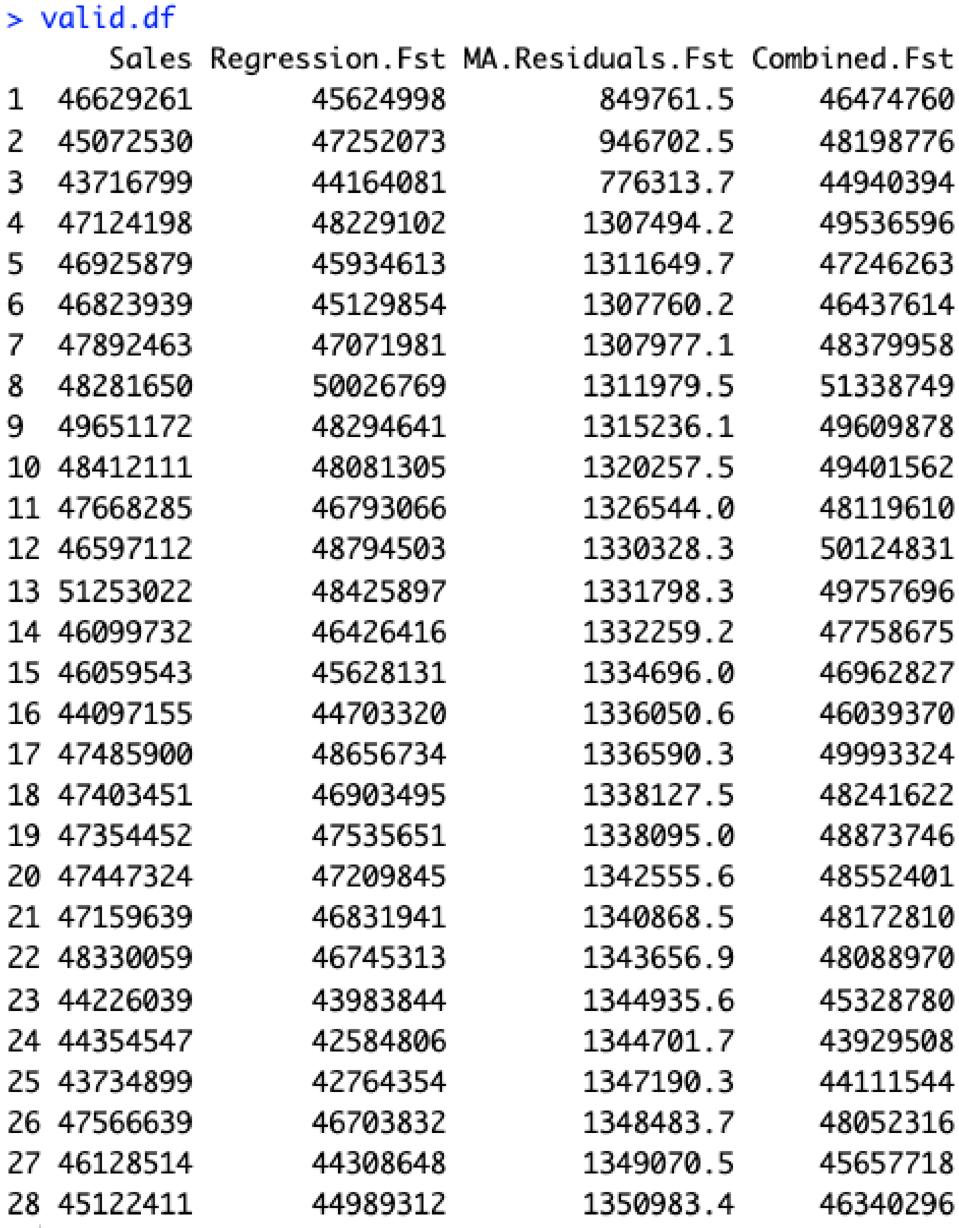
```
> trend.seas.res
Time Series:
Start = c(2010, 6) 
End = c(2012, 15) 
Frequency = 52 
          1           2           3           4           5           6           7           8           9          10 
 2785254.44   140252.03  -422077.33  -289074.20   331434.48   278673.68  -234293.36   496976.67  4402936.95 -1188733.42 
         11          12          13          14          15          16          17          18          19          20 
  287336.39 -1788952.83   269713.89  1002809.54   124134.61   718921.95  1414189.68   890441.39   260572.95   269409.15 
         21          22          23          24          25          26          27          28          29          30 
  544637.78   851649.42   202299.72   546151.75   -10614.22   655711.49   276933.00   289591.52   253969.57  -571437.08 
         31          32          33          34          35          36          37          38          39          40 
 1090984.09  -382247.92  -174449.86  -497624.24   204189.45  -872190.14  -430506.50 -1193975.04  -944328.13 -1254614.28 
         41          42          43          44          45          46          47          48          49          50 
 -992545.04  -474531.27  -204134.09   441402.62   234978.27  1049718.88  2148754.07 -2622804.10  -907650.17  -492533.30 
         51          52          53          54          55          56          57          58          59          60 
 -531007.34    64606.08 -1176708.78 -1224566.66  -347240.83  -496119.34    76234.07 -1383737.27  -715275.88 -1125119.05 
         61          62          63          64          65          66          67          68          69          70 
-2926236.97 -3030890.90  -287336.39  1788952.83  -269713.89 -1002809.54  -124134.61  -718921.95 -1414189.68  -890441.39 
         71          72          73          74          75          76          77          78          79          80 
 -260572.95  -269409.15  -544637.78  -851649.42  -202299.72  -546151.75    10614.22  -655711.49  -276933.00  -289591.52 
         81          82          83          84          85          86          87          88          89          90 
 -253969.57   571437.08 -1090984.09   382247.92   174449.86   497624.24  -204189.45   872190.14   430506.50  1193975.04 
         91          92          93          94          95          96          97          98          99         100 
  944328.13  1254614.28   992545.04   474531.27   204134.09  -441402.62  -234978.27 -1049718.88 -2148754.07  2622804.10 
        101         102         103         104         105         106         107         108         109         110 
  907650.17   492533.30   531007.34   -64606.08 -1608545.66  1084314.63   769318.16   785193.54  -407668.55  1105063.59 
        111         112         113         114 
  949569.24   628142.38 -1476699.98  4219624.32 
```

The trailing MA forecast (window width of 4) for the regression residuals in the validation period is presented below:



```
> ma.trail.res.pred$mean
Time Series:
Start = c(2012, 16) 
End = c(2012, 43) 
Frequency = 52 
 [1]  849761.5  946702.5  776313.7 1307494.2 1311649.7 1307760.2 1307977.1 1311979.5 1315236.1 1320257.5 1326544.0
1330328.3 1331798.3 1332259.2 1334696.0 1336050.6 1336590.3 1338127.5
[19] 1338095.0 1342555.6 1340868.5 1343656.9 1344935.6 1344701.7 1347190.3 1348483.7 1349070.5 1350983.4
```

The table below contains validation partition data (Sales), regression forecast (Regression.Fst), MA forecast for regression residuals (MA.Residuals.Fst), and combined (2-level) forecast (Combined.Fst) that combines the two previous forecasts:

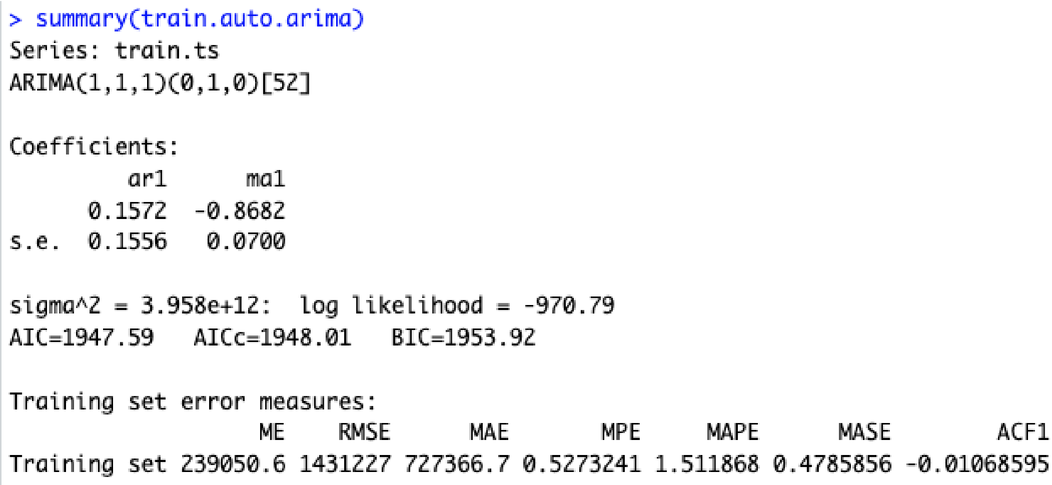


```
> valid.df
      Sales Regression.Fst MA.Residuals.Fst Combined.Fst
1  46629261       45624998         849761.5     46474760
2  45072530       47252073         946702.5     48198776
3  43716799       44164081         776313.7     44940394
4  47124198       48229102        1307494.2     49536596
5  46925879       45934613        1311649.7     47246263
6  46823939       45129854        1307760.2     46437614
7  47892463       47071981        1307977.1     48379958
8  48281650       50026769        1311979.5     51338749
9  49651172       48294641        1315236.1     49609878
10 48412111       48081305        1320257.5     49401562
11 47668285       46793066        1326544.0     48119610
12 46597112       48794503        1330328.3     50124831
13 51253022       48425897        1331798.3     49757696
14 46099732       46426416        1332259.2     47758675
15 46059543       45628131        1334696.0     46962827
16 44097155       44703320        1336050.6     46039370
17 47485900       48656734        1336590.3     49993324
18 47403451       46903495        1338127.5     48241622
19 47354452       47535651        1338095.0     48873746
20 47447324       47209845        1342555.6     48552401
21 47159639       46831941        1340868.5     48172810
22 48330059       46745313        1343656.9     48088970
23 44226039       43983844        1344935.6     45328780
24 44354547       42584806        1344701.7     43929508
25 43734899       42764354        1347190.3     44111544
26 47566639       46703832        1348483.7     48052316
27 46128514       44308648        1349070.5     45657718
28 45122411       44989312        1350983.4     46340296
```

3.6.4 Model 3: Auto ARIMA

The third model applied to the Walmart aggregated weekly sales series was an **automatic ARIMA model** using the auto.arima() function from the forecast package in R. The Auto ARIMA model automatically selects the best-fitting ARIMA structure based on the data, using information criteria such as AICc.

The output from using the auto.arima() function for the training partition is presented below.



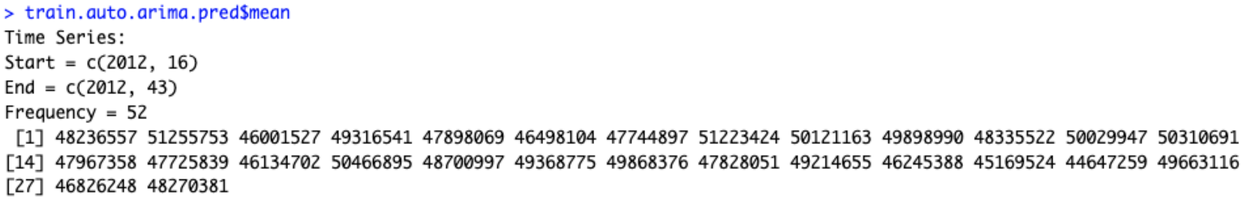
```
> summary(train.auto.arima)
Series: train.ts
ARIMA(1,1,1)(0,1,0)[52]

Coefficients:
         ar1      ma1
      0.1572  -0.8682
s.e.  0.1556   0.0700

sigma^2 = 3.958e+12:  log likelihood = -970.79
AIC=1947.59   AICc=1948.01   BIC=1953.92

Training set error measures:
                   ME    RMSE      MAE       MPE     MAPE      MASE         ACF1
Training set 239050.6 1431227 727366.7 0.5273241 1.511868 0.4785856 -0.01068595
```

This is a seasonal ARIMA model, ARIMA(p, d, q)(P, D, Q)m, where:

- p = 1, order 1 autoregressive model AR(1)
- d = 1, first differencing
- q = 1, order 1 moving average MA(1) for error lags
- P = 0, no autoregressive model for the seasonal part
- D = 1, first differencing for the seasonal part
- Q = 0, no moving average for the seasonal error lags
- m = 52

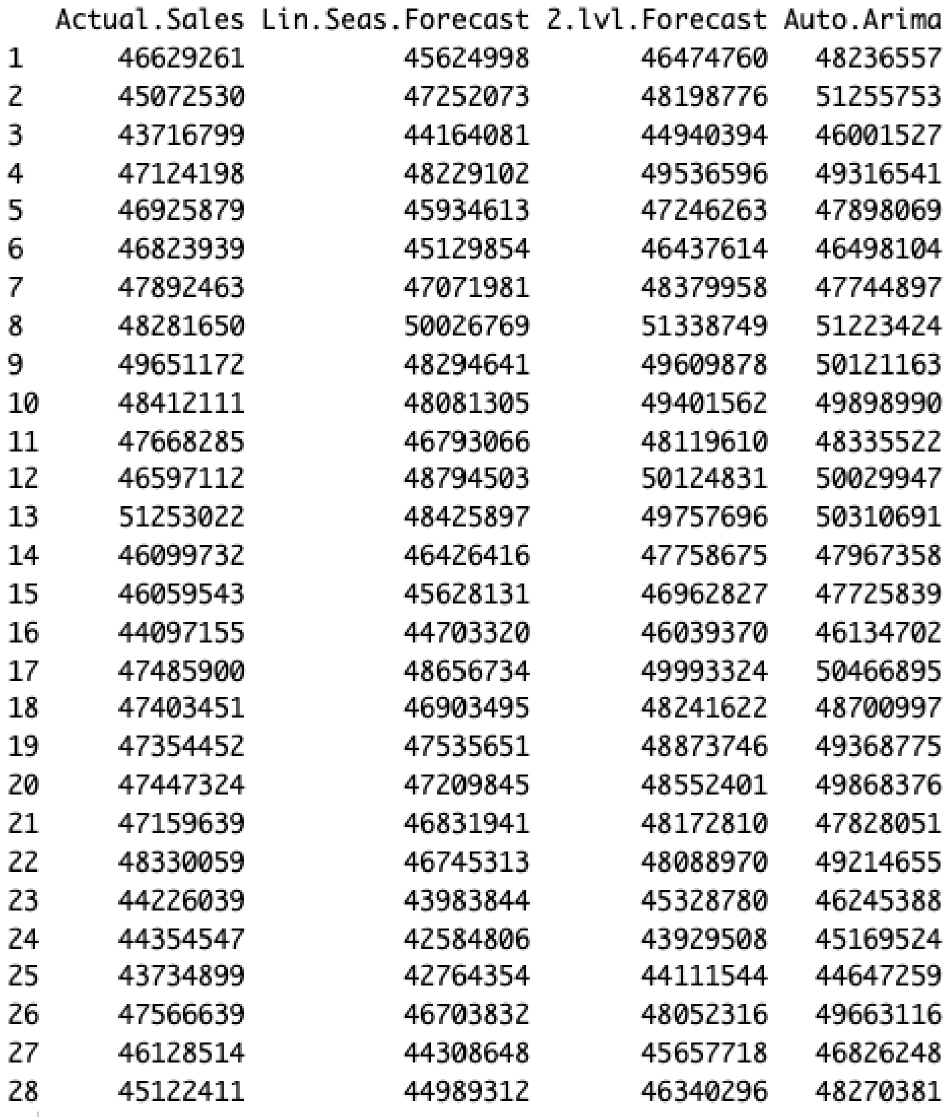
Model Equation : $y_t - y_{t-1} = 0.01572\ (y_{t-1} - y_{t-2}) - 0.8682\ \varepsilon_{t-1}$

Using the model’s equation, see below the forecast for the validation period:



```
> train.auto.arima.pred$mean
Time Series:
Start = c(2012, 16)
End = c(2012, 43)
Frequency = 52
 [1] 48236557 51255753 46001527 49316541 47898069 46498104 47744897 51223424 50121163 49898990 48335522 50029947 50310691
[14] 47967358 47725839 46134702 50466895 48700997 49368775 49868376 47828051 49214655 46245388 45169524 44647259 49663116
[27] 46826248 48270381
```

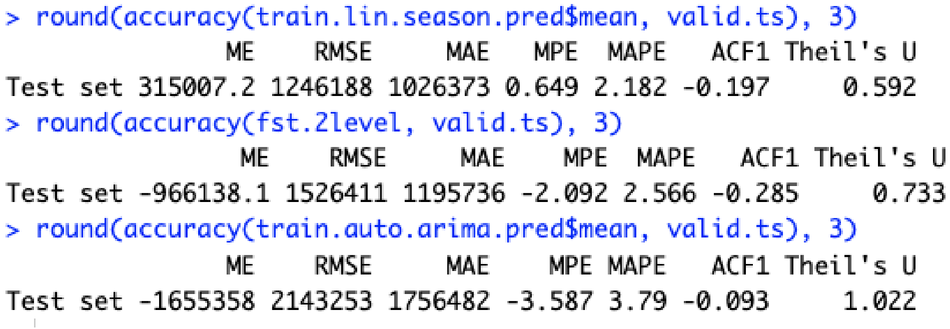
A comparison of forecast for validation period from all the models are presented below:



```
  Actual.Sales Lin.Seas.Forecast Z.lvl.Forecast Auto.Arima
1     46629261          45624998       46474760   48236557
2     45072530          47252073       48198776   51255753
3     43716799          44164081       44940394   46001527
4     47124198          48229102       49536596   49316541
5     46925879          45934613       47246263   47898069
6     46823939          45129854       46437614   46498104
7     47892463          47071981       48379958   47744897
8     48281650          50026769       51338749   51223424
9     49651172          48294641       49609878   50121163
10    48412111          48081305       49401562   49898990
11    47668285          46793066       48119610   48335522
12    46597112          48794503       50124831   50029947
13    51253022          48425897       49757696   50310691
14    46099732          46426416       47758675   47967358
15    46059543          45628131       46962827   47725839
16    44097155          44703320       46039370   46134702
17    47485900          48656734       49993324   50466895
18    47403451          46903495       48241622   48700997
19    47354452          47535651       48873746   49368775
20    47447324          47209845       48552401   49868376
21    47159639          46831941       48172810   47828051
22    48330059          46745313       48088970   49214655
23    44226039          43983844       45328780   46245388
24    44354547          42584806       43929508   45169524
25    43734899          42764354       44111544   44647259
26    47566639          46703832       48052316   49663116
27    46128514          44308648       45657718   46826248
28    45122411          44989312       46340296   48270381
```

3.7 Evaluate & Compare Performance

To assess the performance of each forecasting method, the forecasts generated from the three models were compared against the actual values in the validation set. The evaluation metrics used included **RMSE (Root Mean Squared Error)**, **MAE (Mean Absolute Error)**, **MAPE (Mean Absolute Percentage Error)**, **ACF1** of residuals, and **Theil's U** statistic.



```
> round(accuracy(train.lin.season.pred$mean, valid.ts), 3)
                ME    RMSE     MAE   MPE  MAPE   ACF1 Theil's U
Test set 315007.2 1246188 1026373 0.649 2.182 -0.197     0.592
> round(accuracy(fst.2level, valid.ts), 3)
                 ME    RMSE     MAE    MPE  MAPE   ACF1 Theil's U
Test set -966138.1 1526411 1195736 -2.092 2.566 -0.285     0.733
> round(accuracy(train.auto.arima.pred$mean, valid.ts), 3)
                ME    RMSE     MAE    MPE MAPE   ACF1 Theil's U
Test set -1655358 2143253 1756482 -3.587 3.79 -0.093     1.022
```

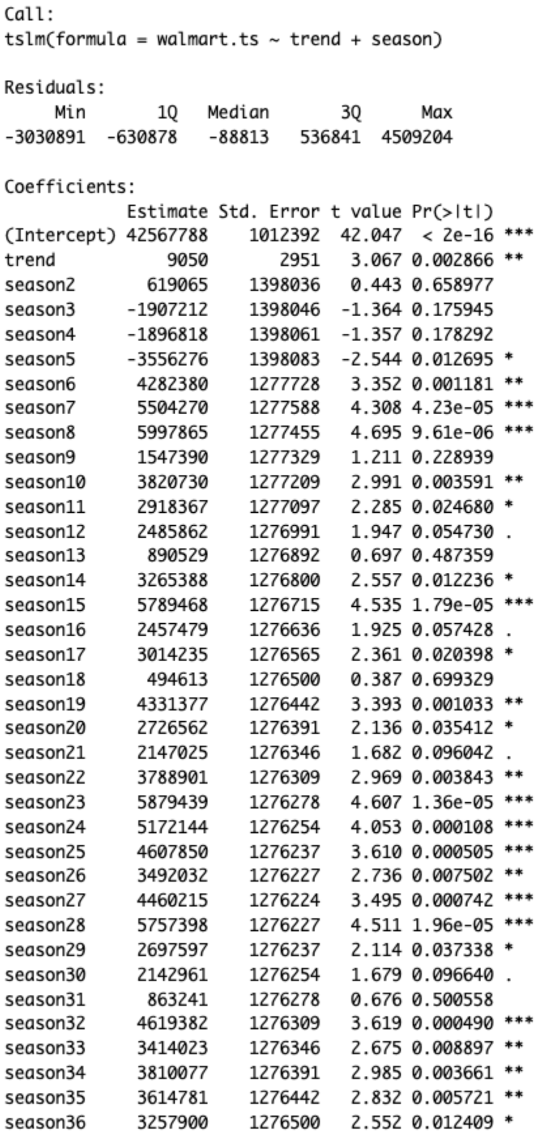
The **linear trend and seasonality model** performed best overall, achieving the **lowest RMSE, MAE, MAPE, and Theil's U**. We still need to fit the entire dataset to make the final decision on the best model.

3.8 Implement Forecast to entire Dataset

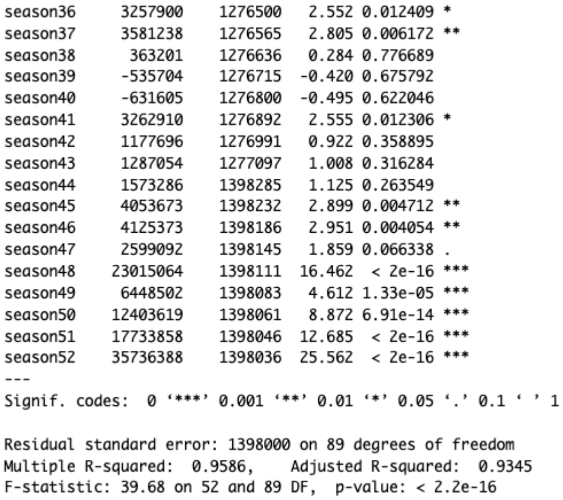
Following model evaluation, all three forecasting models were retrained using the entire dataset (combining both training and validation sets) to fully utilize all available observations and improve forecast precision. Each model was then used to generate forecasts for the next 12 weeks, providing insight into Walmart's expected aggregated weekly sales in the near future.

3.8.1 Model 1: Linear Regression Model with Trend and Seasonality

The model using the summary() function is presented below with entire dataset:



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

Residuals:
     Min       1Q   Median       3Q      Max 
-3030891  -630878   -88813   536841  4509204 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 42567788    1012392  42.047  < 2e-16 ***
trend           9050       2951   3.067 0.002866 ** 
season2       619065    1398036   0.443 0.658977    
season3     -1907212    1398046  -1.364 0.175945    
season4     -1896818    1398061  -1.357 0.178292    
season5     -3556276    1398083  -2.544 0.012695 *  
season6      4282380    1277728   3.352 0.001181 ** 
season7      5504270    1277588   4.308 4.23e-05 ***
season8      5997865    1277455   4.695 9.61e-06 ***
season9      1547390    1277329   1.211 0.228939    
season10     3820730    1277209   2.991 0.003591 ** 
season11     2918367    1277097   2.285 0.024680 *  
season12     2485862    1276991   1.947 0.054730 .  
season13      890529    1276892   0.697 0.487359    
season14     3265388    1276800   2.557 0.012236 *  
season15     5789468    1276715   4.535 1.79e-05 ***
season16     2457479    1276636   1.925 0.057428 .  
season17     3014235    1276565   2.361 0.020398 *  
season18      494613    1276500   0.387 0.699329    
season19     4331377    1276442   3.393 0.001033 ** 
season20     2726562    1276391   2.136 0.035412 *  
season21     2147025    1276346   1.682 0.096042 .  
season22     3788901    1276309   2.969 0.003843 ** 
season23     5879439    1276278   4.607 1.36e-05 ***
season24     5172144    1276254   4.053 0.000108 ***
season25     4607850    1276237   3.610 0.000505 ***
season26     3492032    1276227   2.736 0.007502 ** 
season27     4460215    1276224   3.495 0.000742 ***
season28     5757398    1276227   4.511 1.96e-05 ***
season29     2697597    1276237   2.114 0.037338 *  
season30     2142961    1276254   1.679 0.096640 .  
season31      863241    1276278   0.676 0.500558    
season32     4619382    1276309   3.619 0.000490 ***
season33     3414023    1276346   2.675 0.008897 ** 
season34     3810077    1276391   2.985 0.003661 ** 
season35     3614781    1276442   2.832 0.005721 ** 
season36     3257900    1276500   2.552 0.012409 *  
```



```
season36     3257900    1276500   2.552 0.012409 *  
season37     3581238    1276565   2.805 0.006172 ** 
season38      363201    1276636   0.284 0.776689    
season39     -535704    1276715  -0.420 0.675792    
season40     -631605    1276800  -0.495 0.622046    
season41     3262910    1276892   2.555 0.012306 *  
season42     1177696    1276991   0.922 0.358895    
season43     1287054    1277097   1.008 0.316284    
season44     1573286    1398285   1.125 0.263549    
season45     4053673    1398232   2.899 0.004712 ** 
season46     4125373    1398186   2.951 0.004054 ** 
season47     2599092    1398145   1.859 0.066338 .  
season48    23015064    1398111  16.462  < 2e-16 ***
season49     6448502    1398083   4.612 1.33e-05 ***
season50    12403619    1398061   8.872 6.91e-14 ***
season51    17733858    1398046  12.685  < 2e-16 ***
season52    35736388    1398036  25.562  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1398000 on 89 degrees of freedom
Multiple R-squared:  0.9586,	Adjusted R-squared:  0.9345 
F-statistic: 39.68 on 52 and 89 DF,  p-value: < 2.2e-16
```

The regression model contains 52 predictors: a **time trend index** and **51 seasonal dummy variables** representing weekly seasonality (with one baseline week omitted). The model equation takes the general form:

Model Equation-

$$y_t = 42567788 + 9050\,t + 619065\,D_2 - 1907212\,D_3 - 1896818\,D_4 - 3556276\,D_5 + 4282380\,D_6 + \cdots + 35736388\,D_{52}$$

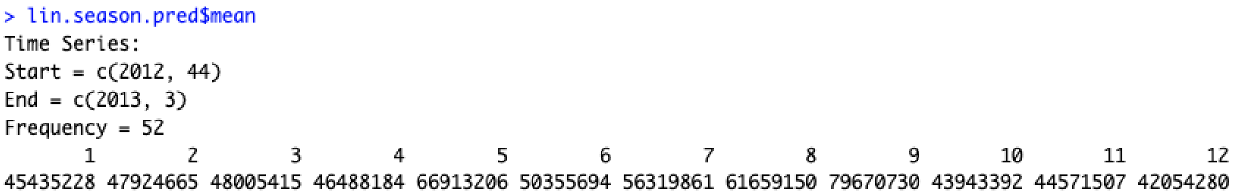
Here:

- D_i is a dummy variable for week *i* (e.g., $D_2 = 1$ if week 2, else 0)
- The baseline (omitted) category is **season1 (week 1)**

The regression summary shows a **Multiple R-squared of 0.9586** and an **Adjusted R-squared of 0.9345**, indicating that the model explains approximately **93.4% of the variance** in the full dataset. The overall model is **highly statistically significant** with a p-value < 2.2e-16. Many weekly dummy coefficients were individually significant at the 1%, 5%, or 10% level, confirming strong seasonality in the series.

This final regression model is a **very strong fit** and statistically robust, making it highly suitable for generating forecasts of Walmart's aggregated weekly sales.

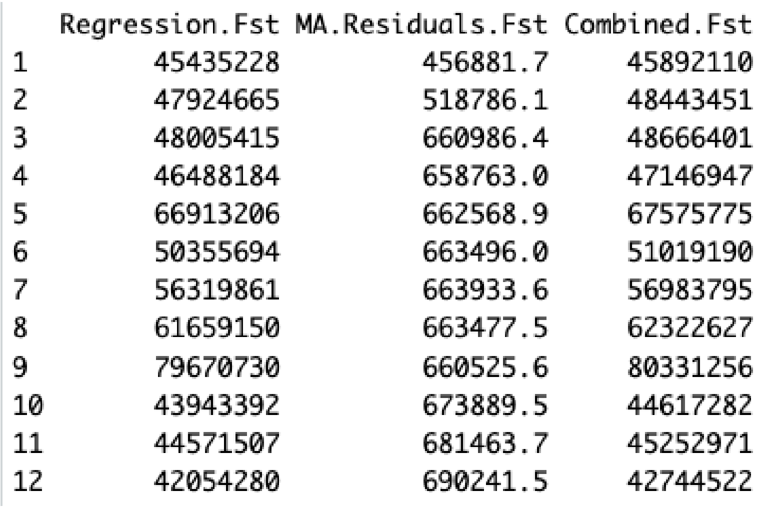
Below is the forecast for Walmarts Sales for next 12 weeks -



```
> lin.season.pred$mean
Time Series:
Start = c(2012, 44)
End = c(2013, 3)
Frequency = 52
       1        2        3        4        5        6        7        8        9       10       11       12 
45435228 47924665 48005415 46488184 66913206 50355694 56319861 61659150 79670730 43943392 44571507 42054280
```

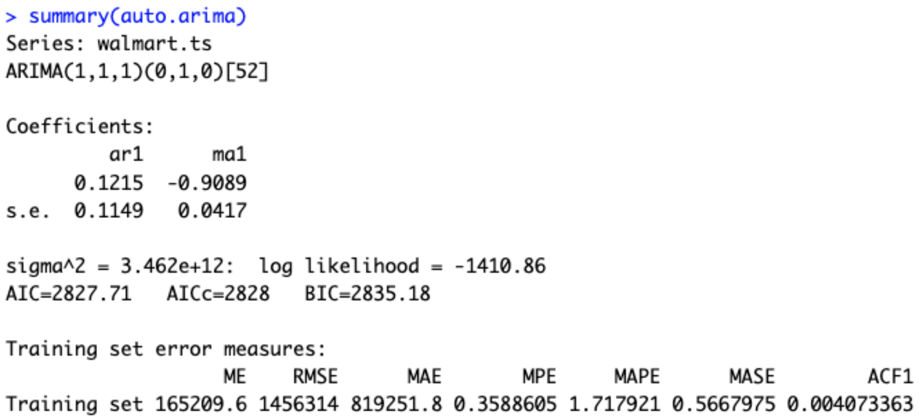
3.8.2 Two-Level Forecasting (Regression + Trailing Moving Average on Residuals)

Forecast for next 12 weeks using Two-level Forecasting model



```
   Regression.Fst MA.Residuals.Fst Combined.Fst
1        45435228         456881.7     45892110
2        47924665         518786.1     48443451
3        48005415         660986.4     48666401
4        46488184         658763.0     47146947
5        66913206         662568.9     67575775
6        50355694         663496.0     51019190
7        56319861         663933.6     56983795
8        61659150         663477.5     62322627
9        79670730         660525.6     80331256
10       43943392         673889.5     44617282
11       44571507         681463.7     45252971
12       42054280         690241.5     42744522
```

3.8.3 Auto-ARIMA Model



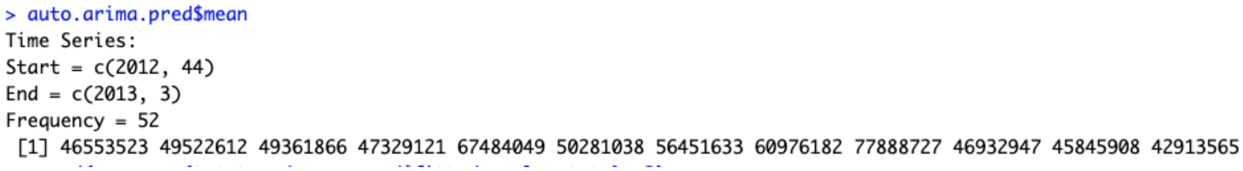
```
> summary(auto.arima)
Series: walmart.ts
ARIMA(1,1,1)(0,1,0)[52]

Coefficients:
         ar1      ma1
      0.1215  -0.9089
s.e.  0.1149   0.0417

sigma^2 = 3.462e+12:  log likelihood = -1410.86
AIC=2827.71   AICc=2828   BIC=2835.18

Training set error measures:
                   ME    RMSE      MAE       MPE     MAPE      MASE        ACF1
Training set 165209.6 1456314 819251.8 0.3588605 1.717921 0.5667975 0.004073363
```

This is a seasonal ARIMA model, ARIMA(p, d, q)(P, D, Q)m, where:

- p = 1, order 1 autoregressive model AR(1)
- d = 1, first differencing
- q = 1, order 1 moving average MA(1) for error lags
- P = 0, no autoregressive model for the seasonal part
- D = 1, first differencing for the seasonal part
- Q = 0, no moving average for the seasonal error lags
- m = 52

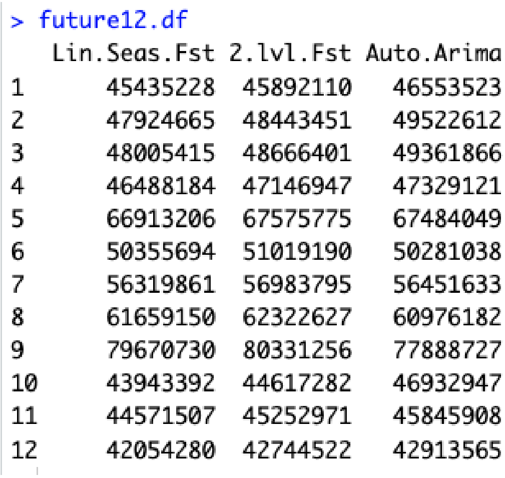
Model Equation : $y_t - y_{t-1} = 0.1215\ (y_{t-1} - y_{t-2}) - 0.9889\ \varepsilon_{t-1}$

Forecast for next 12 weeks using Auto ARIMA model



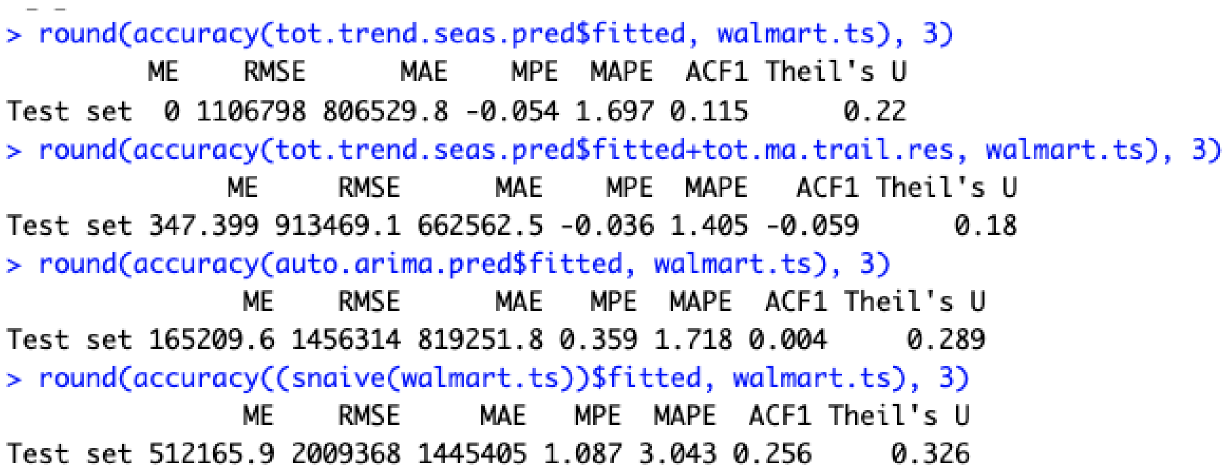
```
> auto.arima.pred$mean
Time Series:
Start = c(2012, 44)
End = c(2013, 3)
Frequency = 52
 [1] 46553523 49522612 49361866 47329121 67484049 50281038 56451633 60976182 77888727 46932947 45845908 42913565
```

Below are forecasts of Walmart sales for next 12 weeks using all three models



```
> future12.df
   Lin.Seas.Fst 2.lvl.Fst Auto.Arima
1      45435228  45892110   46553523
2      47924665  48443451   49522612
3      48005415  48666401   49361866
4      46488184  47146947   47329121
5      66913206  67575775   67484049
6      50355694  51019190   50281038
7      56319861  56983795   56451633
8      61659150  62322627   60976182
9      79670730  80331256   77888727
10     43943392  44617282   46932947
11     44571507  45252971   45845908
12     42054280  42744522   42913565
```

Below are the **accuracy measures** obtained from all the models for the entire dataset.



```
> round(accuracy(tot.trend.seas.pred$fitted, walmart.ts), 3)
         ME    RMSE      MAE    MPE  MAPE  ACF1 Theil's U
Test set  0 1106798 806529.8 -0.054 1.697 0.115      0.22
> round(accuracy(tot.trend.seas.pred$fitted+tot.ma.trail.res, walmart.ts), 3)
              ME     RMSE      MAE    MPE  MAPE   ACF1 Theil's U
Test set 347.399 913469.1 662562.5 -0.036 1.405 -0.059      0.18
> round(accuracy(auto.arima.pred$fitted, walmart.ts), 3)
               ME    RMSE      MAE   MPE  MAPE  ACF1 Theil's U
Test set 165209.6 1456314 819251.8 0.359 1.718 0.004     0.289
> round(accuracy((snaive(walmart.ts))$fitted, walmart.ts), 3)
               ME    RMSE     MAE   MPE  MAPE  ACF1 Theil's U
Test set 512165.9 2009368 1445405 1.087 3.043 0.256     0.326
```

After fitting all models on the full dataset, performance metrics confirmed the superiority of the two-level model combining linear regression with trailing moving averages on residuals. This model achieved the lowest RMSE (913,469) and lowest MAPE (1.405%), along with the best residual behavior (ACF1 = –0.059) and Theil's U of 0.18, indicating strong predictive accuracy and minimal remaining autocorrelation. The simpler linear trend and seasonality model also performed well, with RMSE of 1,106,798 and Theil's U of 0.22, offering a solid balance of performance and interpretability. The auto ARIMA model, while automated, lagged behind with higher RMSE and Theil's U (0.289), while the seasonal naïve model performed the worst overall, with MAPE exceeding 3% and Theil's U of 0.326. These results confirm that the two-level model offers the most accurate forecasts for Walmart's aggregated weekly sales, followed closely by the linear regression model.

4. Conclusion

This project successfully developed and compared three time series forecasting models to predict Walmart's aggregated weekly sales across 45 stores. The models included a linear regression with trend and seasonality, a two-level model incorporating a trailing moving average on residuals, and an automatic ARIMA model. These were evaluated using multiple accuracy metrics, including RMSE, MAE, MAPE, ACF1, and Theil's U.

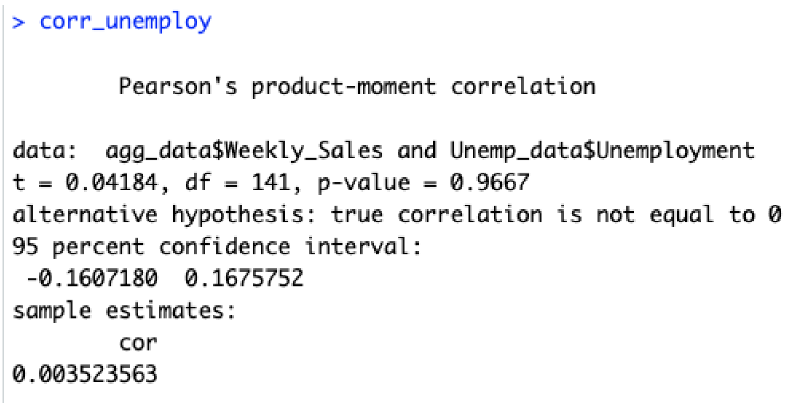
The results demonstrated that the two-level model delivered the best overall performance, achieving the lowest RMSE and MAPE, and showing minimal residual autocorrelation. The linear trend and seasonality model also performed strongly and remains a simpler yet interpretable alternative. The auto ARIMA model, while useful as a benchmark, underperformed relative to the other methods. A seasonal naïve model was also included as a baseline and confirmed the value of more sophisticated approaches.

All three models were retrained on the full dataset and used to generate 12-week forecasts, offering Walmart actionable projections for upcoming sales trends. The two-level model, in particular, offers a powerful forecasting solution by capturing both long-term structure and short-term variation, making it highly suitable for operational use.

In conclusion, the combination of statistical modeling and layered forecasting strategies provided valuable insights and robust short-term predictions.

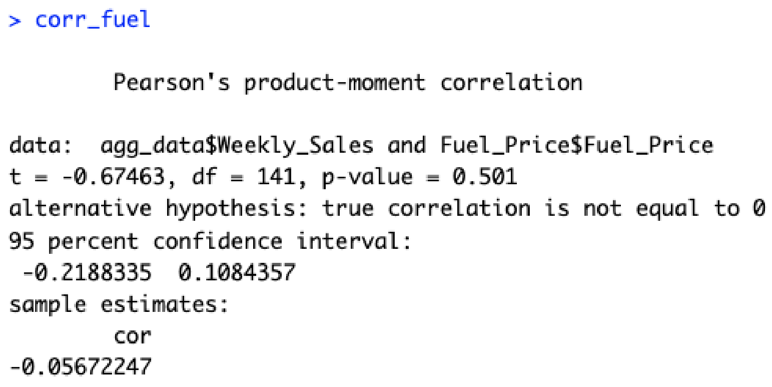
5. Appendix

Correlation of Unemployment with Weekly Sales



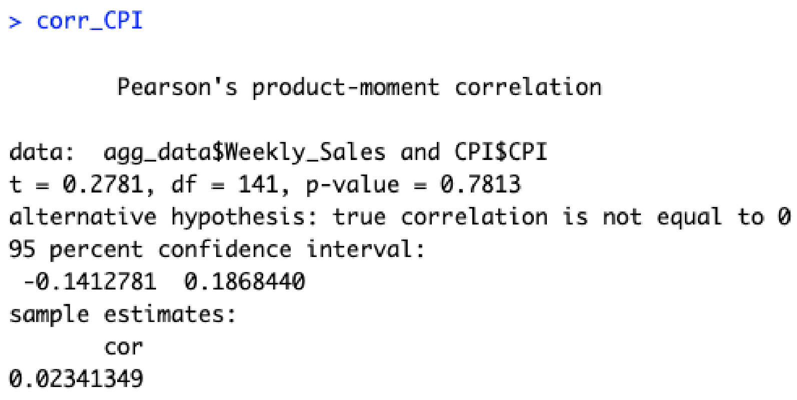
```
> corr_unemploy

	Pearson's product-moment correlation

data:  agg_data$Weekly_Sales and Unemp_data$Unemployment
t = 0.04184, df = 141, p-value = 0.9667
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.1607180  0.1675752
sample estimates:
        cor
0.003523563
```

Correlation of Fuel Price with Weekly Sales



```
> corr_fuel

	Pearson's product-moment correlation

data:  agg_data$Weekly_Sales and Fuel_Price$Fuel_Price
t = -0.67463, df = 141, p-value = 0.501
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.2188335  0.1084357
sample estimates:
        cor
-0.05672247
```

Correlation of CPI with Weekly Sales



```
> corr_CPI

	Pearson's product-moment correlation

data:  agg_data$Weekly_Sales and CPI$CPI
t = 0.2781, df = 141, p-value = 0.7813
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.1412781  0.1868440
sample estimates:
       cor
0.02341349
```