“Analysis of NYC Taxi Ridership: A Time Series Approach”

(BAN-673)

Author(s) Name(s):

- **Aarthi Sivashankar** (mn6070)
- **Aditya Chache** (hz6941)
- **Chenghui Tan** (sw5299)
- **Nuzhat Fatima** (bw8988)

Instructor: Prof. **Zinovy Radovilsky**

Summary

In this project, we forecasted NYC taxi ridership using time series data from the NYC Taxi and Limousine Commission, aggregated in 1-hour intervals from July 1, 2014, to January 31, 2015. The data revealed strong seasonality, with clear hourly and daily patterns, as well as fluctuations during significant events like the NYC Marathon, Christmas, and a January blizzard. Autocorrelation function (ACF) plots confirmed the presence of trend and seasonal components, guiding us to select models that could capture these patterns.

We tested various forecasting models, including regression with linear and quadratic trends, a two-level forecast with a trailing moving average, Auto ARIMA, and Holt-Winters exponential smoothing. While regression models struggled with trend components and seasonality, Auto ARIMA overfitted the training data. Holt-Winters provided reasonable results but also overfitted, leaving residuals in trend and seasonality. To better capture trend, seasonality, and noise, we refined our models and found that ARIMA(3,1,2)(1,1,2) performed fairly well. Ultimately, the Two-Level Forecast model (combining quadratic trends, seasonality, and a 24-hour trailing moving average) and ARIMA(3,1,2)(1,1,2) emerged as the most accurate, achieving the lowest MAPE and RMSE on the validation set. While ARIMA(3,1,2)(1,1,2) model still had limitations in capturing peak values, it was the best model for forecasting NYC taxi ridership, providing the most reliable results compared to other approaches.

Introduction

In this project, we aim to analyze and forecast the number of taxi passengers in New York City using time series data provided by the NYC Taxi and Limousine Commission. The dataset consists of aggregated passenger counts recorded in 30-minute intervals, capturing the dynamic nature of taxi demand throughout the day. This data provides a rich opportunity to explore patterns, trends, and anomalies in passenger behavior, particularly during significant events that impact transportation demand.

Five notable anomalies are present in the dataset, corresponding to major events in NYC: the NYC Marathon, Thanksgiving, Christmas, New Year's Day, and a snowstorm. These events are expected to significantly influence taxi usage, either by increasing demand due to heightened activity or decreasing it due to disruptions like severe weather. Understanding how these anomalies affect the time series will be a key focus of our analysis.

Our primary objective is to evaluate and compare the performance of various forecasting models in predicting the number of taxi passengers. By leveraging techniques such as ARIMA, SARIMA, Exponential Smoothing, and Regression, we aim to identify the most accurate and reliable model for this dataset. The insights gained from this project will not only enhance our understanding of time series forecasting but also provide practical implications for optimizing taxi services in response to fluctuating demand.

This documentation outlines our methodology, model selection, evaluation process, and findings, culminating in a recommendation for the best-performing forecasting model. Through this project, we hope to contribute to the broader field of time series analytics while addressing a real-world problem with significant economic and logistical implications.

Dataset Source: NYC Taxi Traffic on Kaggle

Eight Steps of Forecasting

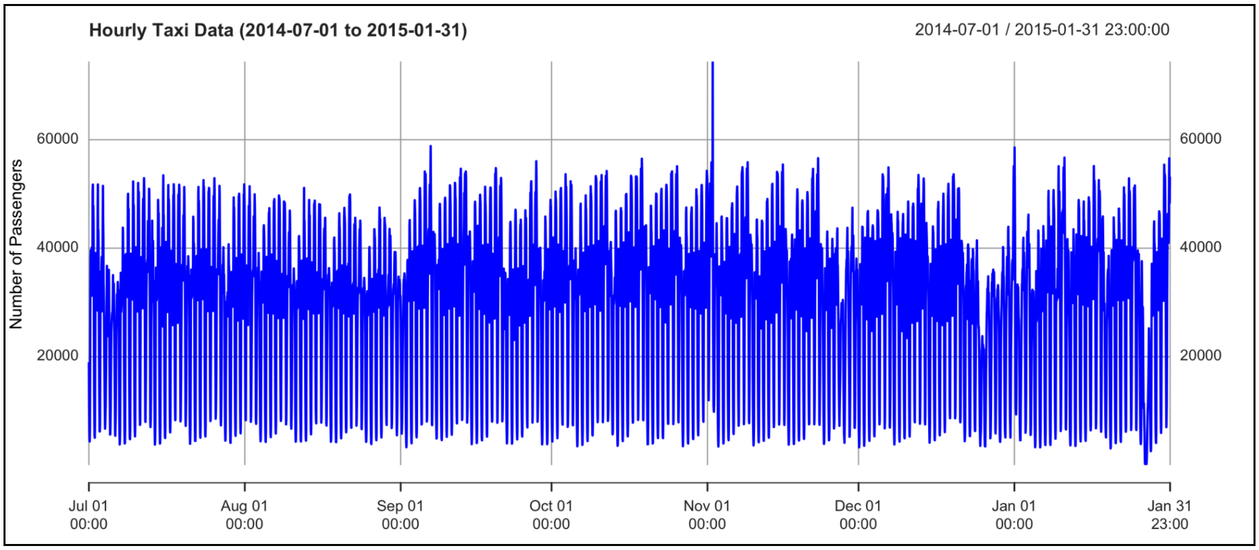
Step 1: Define Goal

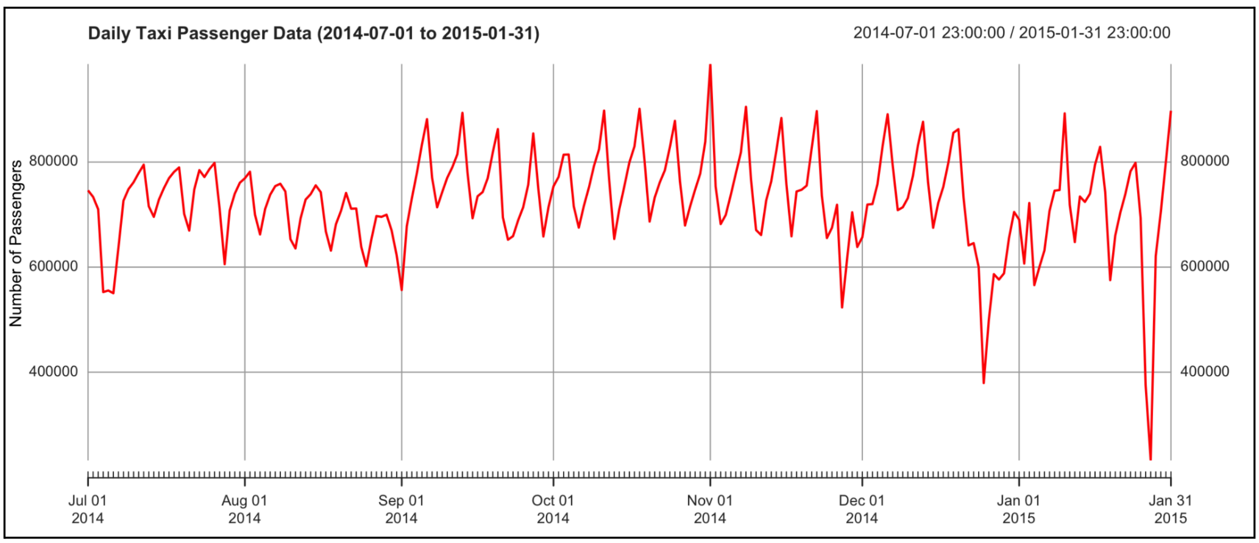
The main goal of this project is to analyze the NYC taxi passenger data, aggregated into hourly intervals (data originally in 30 mins intervals), to detect any irregular patterns or anomalies in passenger counts, particularly during key events like the NYC Marathon, Thanksgiving, Christmas, New Year's Day, and snowstorms. In addition to anomaly detection, the project aims to use the data for forecasting future taxi ridership trends, identifying how factors such as holidays, public events, and extreme weather impact passenger behavior. By uncovering these irregular patterns and creating forecasting models, the analysis will provide valuable insights into taxi demand, which can inform better transportation planning and resource allocation.

Step 2: Get The Data

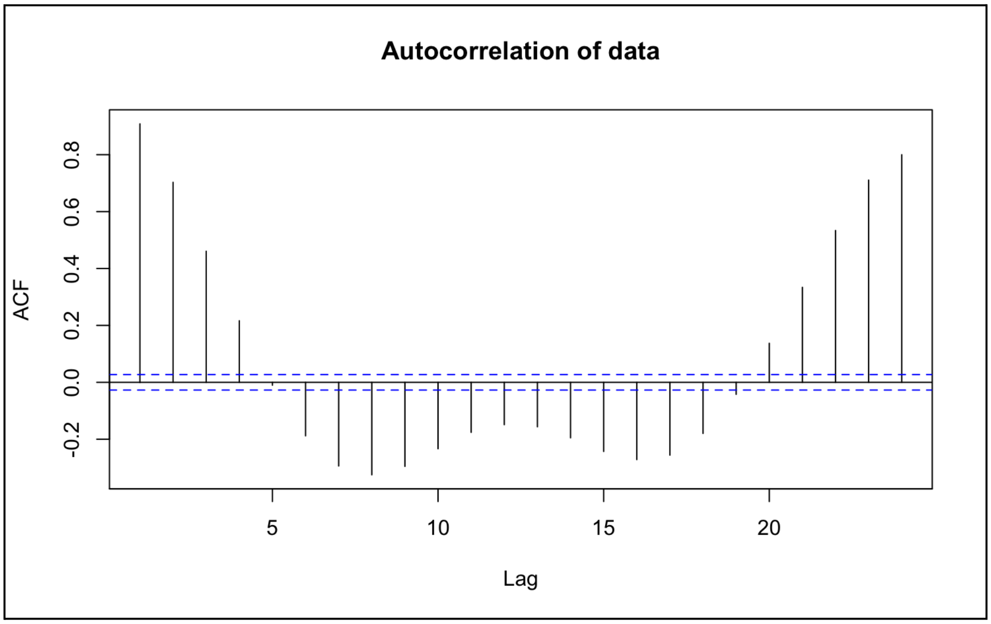
The data for this analysis is sourced from the NYC Taxi and Limousine Commission, which provides detailed records of taxi passenger counts in the city. The dataset aggregates the total number of taxi passengers into 30-minute intervals, allowing for a granular view of passenger demand over time. The data collection timeline spans from July 1st, 2014, to January 31st, 2015, capturing the city's regular days and major events.

Step 3: Explore & Visualize Series





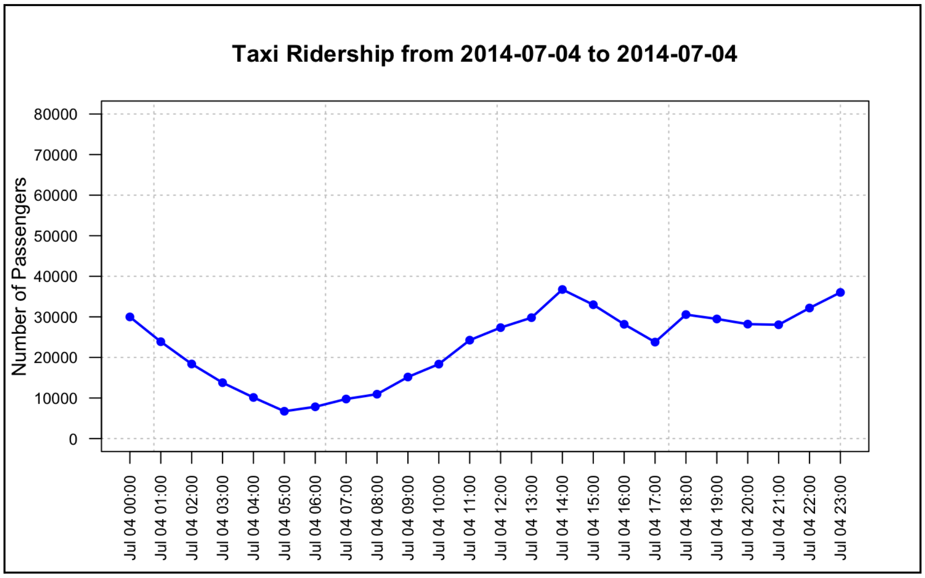
The above plot shows the hourly aggregation of taxi ridership from July 1st, 2014 to January 31st, 2015. On observing the plot, we can see that the data shows an overall stable trend and seasonality, suggesting that the hourly data doesn't fluctuate that much. There is one big spike towards the start of November, and there are also some noticeable lows towards the end of December 2014 and January 2015. If we observe the daily aggregation plot of ridership, we can see that during the end of January, the ridership is at the lowest, suggesting that there might be some event that had affected it. The same goes for November 1st (peak ridership). The overall stable trend can also be seen in this plot. But overall, from the two plots above, it's clear that there is an hourly trend and seasonality in the dataset. And since the patterns are more clear in the hourly aggregation, we are going to use it for training our forecasting models.



From the ACF plot above, we can see that the data is highly correlated since the autocorrelation coefficients in almost all the lags are above the horizontal threshold. There is a positive autocorrelation coefficient in lag 1, which is significantly higher than the threshold, suggesting an upward trend component. The presence of a significant positive autocorrelation coefficient in lag 24 tells that there is a seasonal component present in the data. Hence, we can conclude that the data contain all the 4 components of the time series.

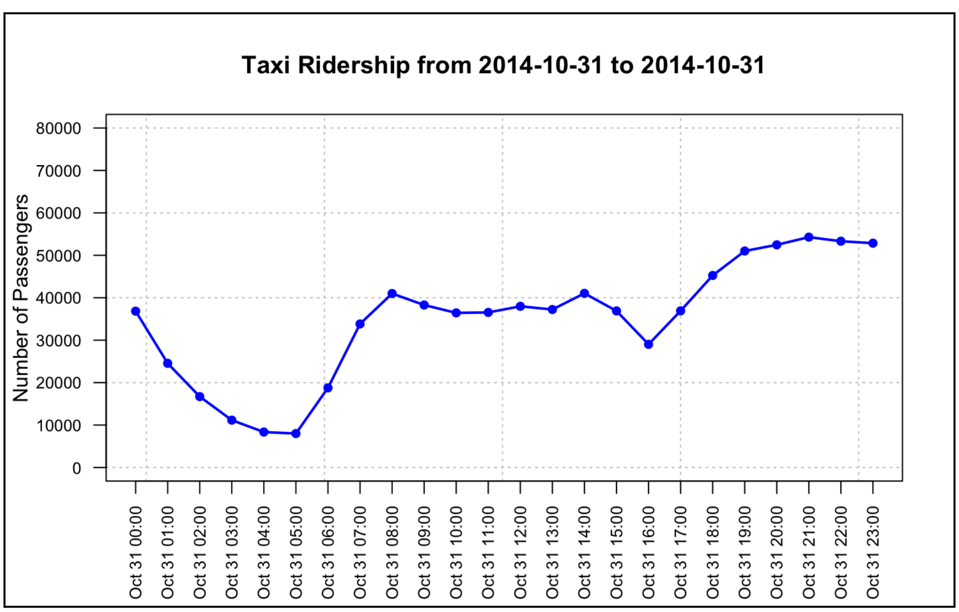
We can also see the visualization for some important events that happened during this period, which might have affected the taxi ridership.

4th of July



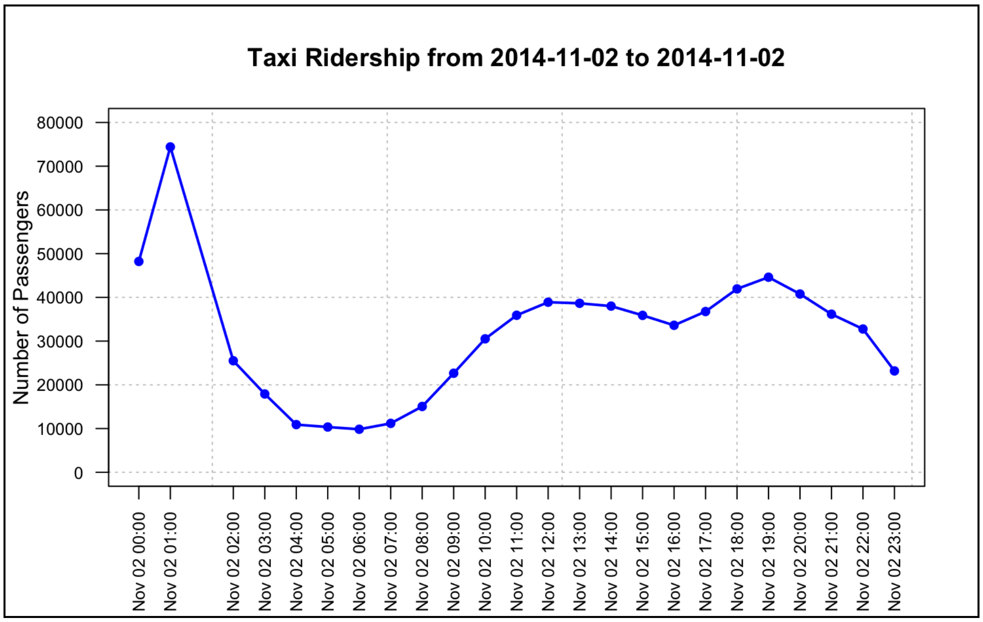
During the 4th of July, we can see that the ridership is pretty stable, with the most number of passengers travelling in the afternoon towards the end of the day. Considering all the celebration and the parades, it doesn't look like it affects the ridership that much.

31st October - Halloween



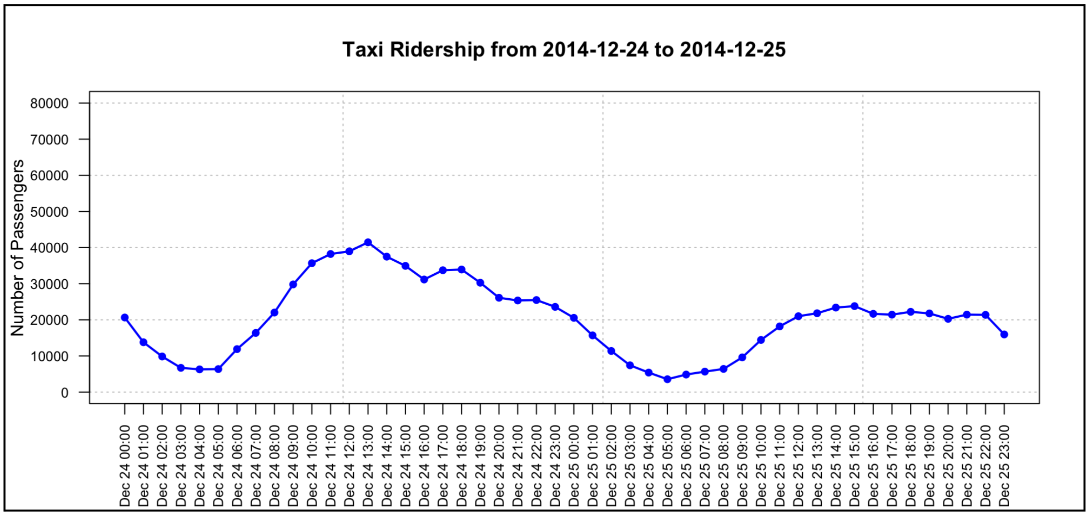
Halloween is another major celebration in the US, and from the above plot, we can see that the ridership is not that affected; instead, we can see a jump in the ridership compared to the 4th of July.

2nd November - NYC Marathon



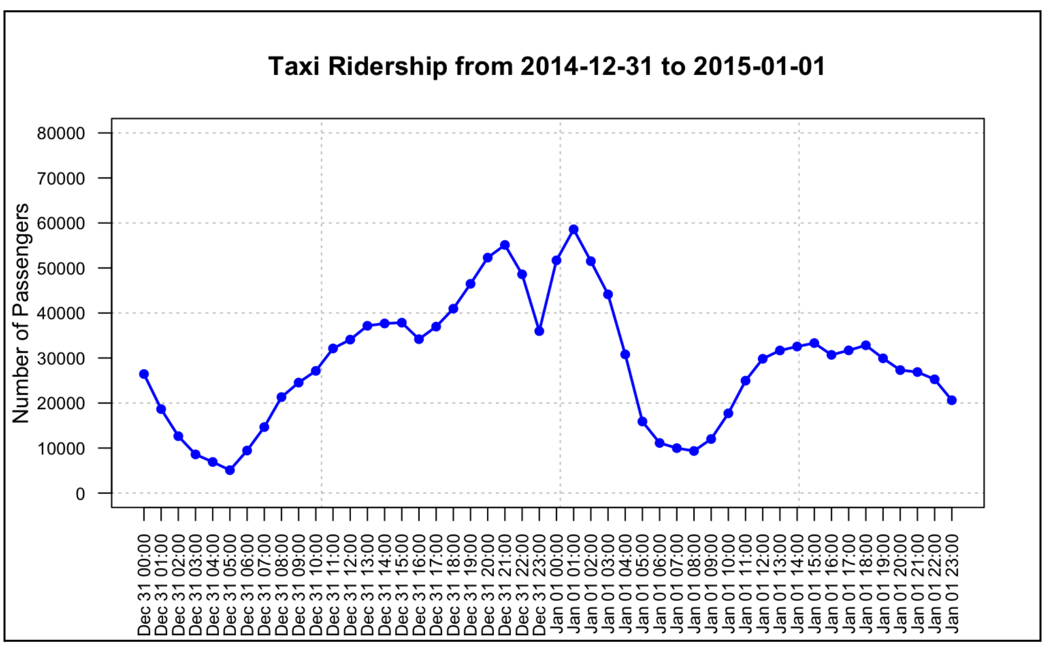
This is another significant event that happened in NYC on 2nd November. The peak that we can see is the time the most ridership was observed. The gap in the timeline is due to the daylight savings time adjustment, and it might've been aggregated into a single hour, but still, the peak might've been because of the street closures due to the marathon, which took place at 6:40 am. Hence, we can see the low ridership during this time.

24th and 25th December - Christmas



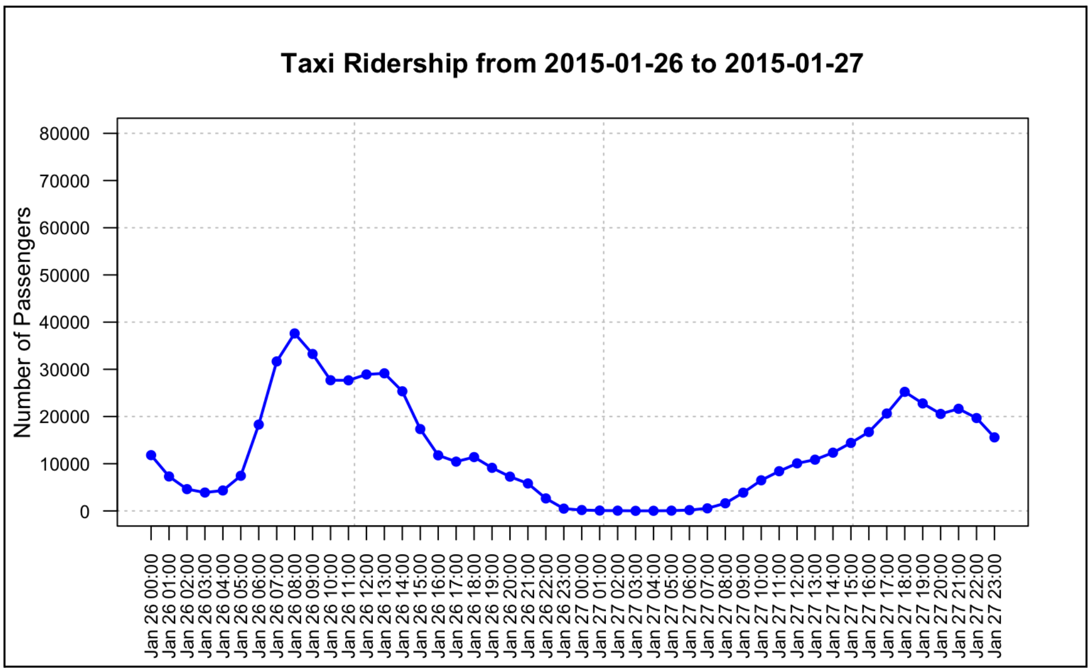
During Christmas (25th), we can see a drop in ridership compared to the previous day, when the peak ridership was above 40000 passengers. The ridership starts dropping towards the end of Christmas Eve and is at its lowest at 5 am early in the morning on the day of Christmas.

31st December and 1st January - New Year



On New Year's Eve and New Year's Day, a distinct trend can be observed in the ridership patterns. Ridership increases towards the end of New Year's Eve, suggesting that people attending parties or gatherings may prefer taking a cab. Additionally, another reason for choosing a taxi could be to avoid driving after drinking. The ridership reaches its peak at 1 am on New Year's Day, which further supports the idea of increased demand during this time.

26th and 27th January - NYC Blizzard



On January 27th, a severe blizzard hit NYC, which is clearly reflected in the above plot. Taxi ridership dropped to almost zero from 11 pm on January 26th to 7 am on January 27th. After 8 am, ridership gradually began to return to normal levels. These low values are extremes of the dataset and hence can also be considered outliers.

Step 4: Data Preprocessing

In this part, we aggregated the data into hourly intervals instead of keeping it in 30-minute intervals and then checked for any missing values. There were a few outliers in the dataset, but we decided to keep them as it is.

Step 5: Partitioning The Series

For partitioning the Series, we have used the months of July to December for the training set and the month of January for validation. There are 4440 points in our training set and 720 points in our validation set.

Step 6 & 7: Forecasting & Evaluating Models

We will first experiment with regression models as they are simple to understand and also work well in many cases. Since our data has both trend and seasonality, we will use regression models that capture both of these components.

Regression with Linear Trend and Seasonality

Since our dataset has all the 4 components of time series, we will first use a regression model with the linear trend and seasonality components. Our approach will be to experiment with a few models and evaluate the accuracies on the validation set. Then, we will select the two best models from them and train those two on the whole dataset for forecasting into the future.

Below is the summary of the model on the training set -

```
> summary(linear_trend_seasonality_model)

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

Residuals:
   Min     1Q Median     3Q    Max 
-24520  -4217     12   4123  52122 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.977e+04  6.007e+02  49.552  < 2e-16 ***
trend        6.971e-02  9.025e-02   0.772  0.43988    
season2     -7.687e+03  8.014e+02  -9.592  < 2e-16 ***
season3     -1.353e+04  8.014e+02 -16.886  < 2e-16 ***
season4     -1.806e+04  8.014e+02 -22.540  < 2e-16 ***
season5     -2.117e+04  8.014e+02 -26.416  < 2e-16 ***
season6     -2.261e+04  8.014e+02 -28.209  < 2e-16 ***
season7     -1.515e+04  8.014e+02 -18.899  < 2e-16 ***
season8     -4.982e+03  8.014e+02  -6.216 5.57e-10 ***
season9      8.678e+02  8.014e+02   1.083  0.27893    
season10     2.287e+03  8.014e+02   2.854  0.00434 ** 
season11     2.532e+03  8.014e+02   3.160  0.00159 ** 
season12     4.192e+03  8.014e+02   5.230 1.77e-07 ***
season13     5.868e+03  8.014e+02   7.322 2.89e-13 ***
season14     5.812e+03  8.014e+02   7.252 4.84e-13 ***
season15     7.082e+03  8.014e+02   8.837  < 2e-16 ***
season16     5.299e+03  8.014e+02   6.612 4.24e-11 ***
season17    -3.081e+01  8.014e+02  -0.038  0.96934    
season18     6.138e+03  8.014e+02   7.658 2.30e-14 ***
season19     1.379e+04  8.014e+02  17.204  < 2e-16 ***
season20     1.608e+04  8.014e+02  20.067  < 2e-16 ***
season21     1.367e+04  8.014e+02  17.061  < 2e-16 ***
season22     1.360e+04  8.014e+02  16.966  < 2e-16 ***
season23     1.240e+04  8.014e+02  15.475  < 2e-16 ***
season24     7.823e+03  8.014e+02   9.762  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7708 on 4415 degrees of freedom
Multiple R-squared:  0.6808,	Adjusted R-squared:  0.679 
F-statistic: 392.3 on 24 and 4415 DF,  p-value: < 2.2e-16
```

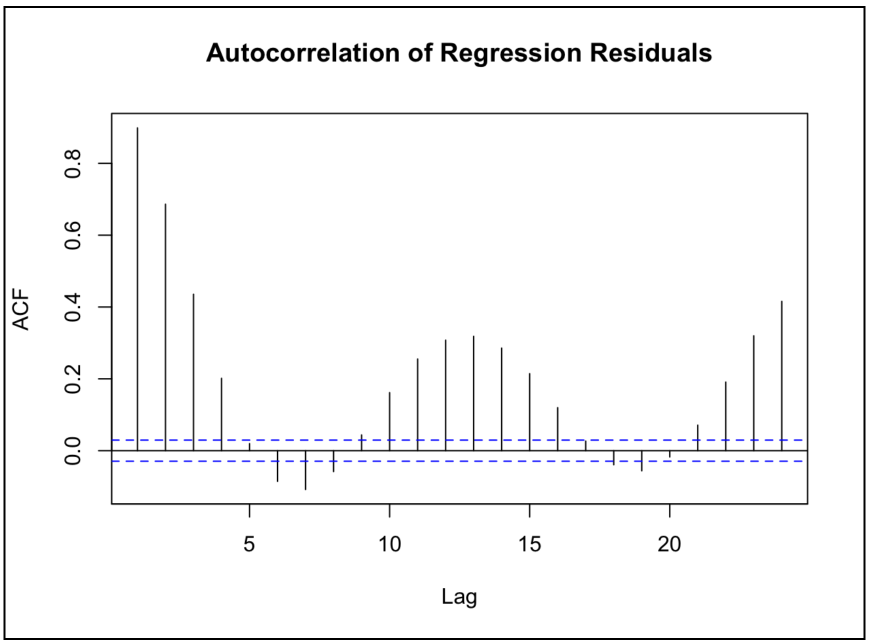
The p-values of most seasonal coefficients are low (<0.05), indicating that they are significant at the 5% significance level. In contrast, the high p-value of the trend coefficient suggests that it may not contribute significantly to the forecasting model. The R-squared value of 0.679 indicates that the model explains approximately 68% of the variance in the training data.

```
                        ME     RMSE      MAE        MPE      MAPE     MASE      ACF1 Theil's U
Training set -4.227921e-13 7685.956 5675.087  -14.57106  29.98128 1.148511 0.8985231        NA
Test set     -1.893436e+03 9651.374 7173.977 -313.67445 327.88465 1.451853 0.9228602  61.39805
```

On evaluating the performance of the model on the test set, we can observe that the MAPE and RMSE is very high, suggesting that the model is not fully capturing the patterns in the data (underfitting).

	Actual_Ridership	Regression_Forecast
1	16601	30076.495
2	10811	22389.533
3	7117	16543.901
4	5090	12012.906
5	4518	8906.468
6	4441	7470.230

Because of the high error, we can see a huge difference between the actual and the forecasted values. To improve this, we can check the ACF plot of the regression residuals and then implement a multi-level forecast system.



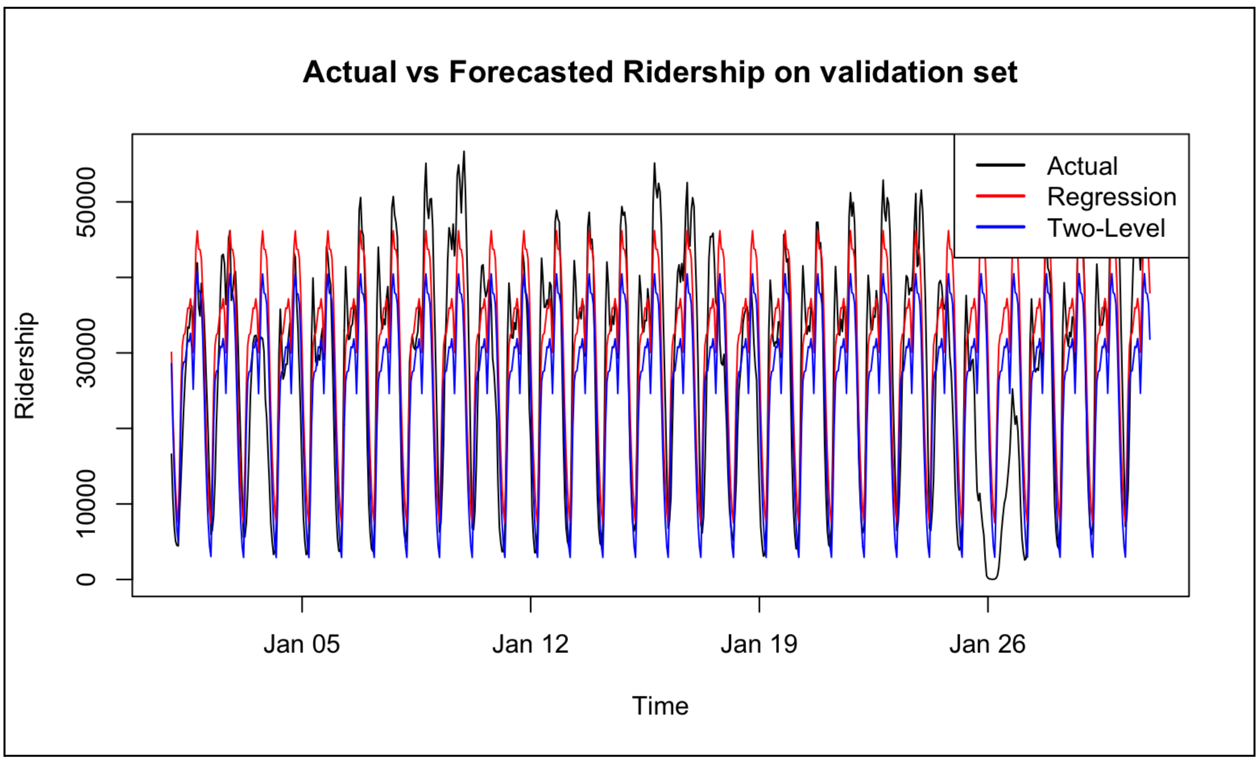
We can observe that the above ACF plot of the regression residuals still shows that there are patterns in the data since most autocorrelation coefficients are above the threshold and are positively correlated.

Two-Level Forecast with Trailing Moving Average of window 24

	Actual_Ridership	Regression_Forecast	Residual_MA_Forecast	Two_Level_Forecast
1	16601	30076.495	-1487.000	28589.495
2	10811	22389.533	-1678.011	20711.522
3	7117	16543.901	-1851.366	14692.535
4	5090	12012.906	-2006.898	10006.008
5	4518	8906.468	-2217.332	6689.136
6	4441	7470.230	-2457.169	5013.061

```
> print(acc_two_level_forecast_lts)
                ME     RMSE     MAE      MPE     MAPE      ACF1 Theil's U
Test set 3268.734 10032.38 8120.65 -187.009 222.1511 0.9226849  35.19437
```

From the above results, we can see that the MAPE of the two-level forecast is lower than just the regression forecast, but the RMSE value is higher, suggesting that there might be more patterns in the data, and even the two-level forecast is not able to fully capture the patterns in the data.



Regression with Quadratic Trend and Seasonality

To try and fully capture the trend in the data, the next model to fit will be a regression model with quadratic trend and seasonality. First, we will evaluate the base regression model, and then we will check the two-level forecast with a trailing moving average of window 24.

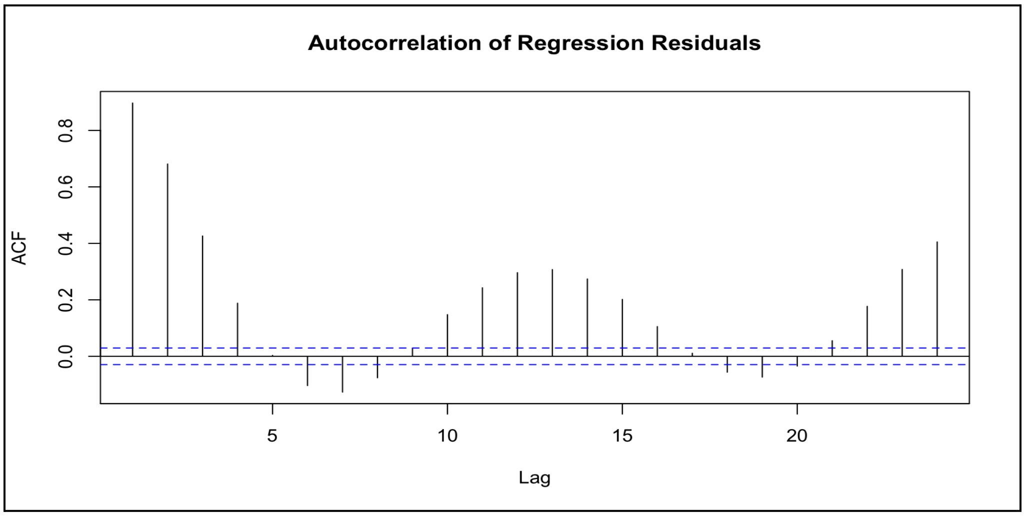
Below is the summary for the model -

```
> summary(quadratic_trend_seasonality_model)

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

Residuals:
   Min     1Q Median     3Q    Max 
-22787  -4292    -87   4155  51381 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.749e+04  6.484e+02  42.396  < 2e-16 ***
trend        3.144e+00  3.579e-01   8.783  < 2e-16 ***
I(trend^2)  -6.922e-04  7.804e-05  -8.870  < 2e-16 ***
season2     -7.687e+03  7.944e+02  -9.676  < 2e-16 ***
season3     -1.353e+04  7.944e+02 -17.034  < 2e-16 ***
season4     -1.806e+04  7.944e+02 -22.738  < 2e-16 ***
season5     -2.117e+04  7.944e+02 -26.648  < 2e-16 ***
season6     -2.261e+04  7.944e+02 -28.456  < 2e-16 ***
season7     -1.515e+04  7.944e+02 -19.065  < 2e-16 ***
season8     -4.982e+03  7.944e+02  -6.271 3.94e-10 ***
season9      8.677e+02  7.944e+02   1.092  0.27479    
season10     2.287e+03  7.944e+02   2.878  0.00402 ** 
season11     2.532e+03  7.944e+02   3.188  0.00144 ** 
season12     4.192e+03  7.944e+02   5.276 1.38e-07 ***
season13     5.868e+03  7.944e+02   7.386 1.80e-13 ***
season14     5.811e+03  7.944e+02   7.315 3.04e-13 ***
season15     7.082e+03  7.944e+02   8.915  < 2e-16 ***
season16     5.299e+03  7.944e+02   6.670 2.88e-11 ***
season17    -3.088e+01  7.944e+02  -0.039  0.96899    
season18     6.137e+03  7.944e+02   7.725 1.37e-14 ***
season19     1.379e+04  7.945e+02  17.355  < 2e-16 ***
season20     1.608e+04  7.945e+02  20.242  < 2e-16 ***
season21     1.367e+04  7.945e+02  17.210  < 2e-16 ***
season22     1.360e+04  7.945e+02  17.114  < 2e-16 ***
season23     1.240e+04  7.945e+02  15.610  < 2e-16 ***
season24     7.823e+03  7.945e+02   9.847  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 7641 on 4414 degrees of freedom
Multiple R-squared:  0.6864,	Adjusted R-squared:  0.6846 
F-statistic: 386.4 on 25 and 4414 DF,  p-value: < 2.2e-16
```

The p-value of almost all the coefficients is less than 0.05 (5% significance level). The Adjusted R-Squared is 68%, which is almost the same as the previous model.

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	3.736303e-13	7618.356	5653.896	-14.42711	30.03015	1.144222	0.8969421	NA
Test set	1.607042e+03	9610.918	7573.666	-202.28800	230.12687	1.532741	0.9224557	36.46256

The MAPE and RMSE are better than the linear trend and seasonality model. We can also test a two-level forecast with a trailing moving average of 24 to see if there is any improvement or not.



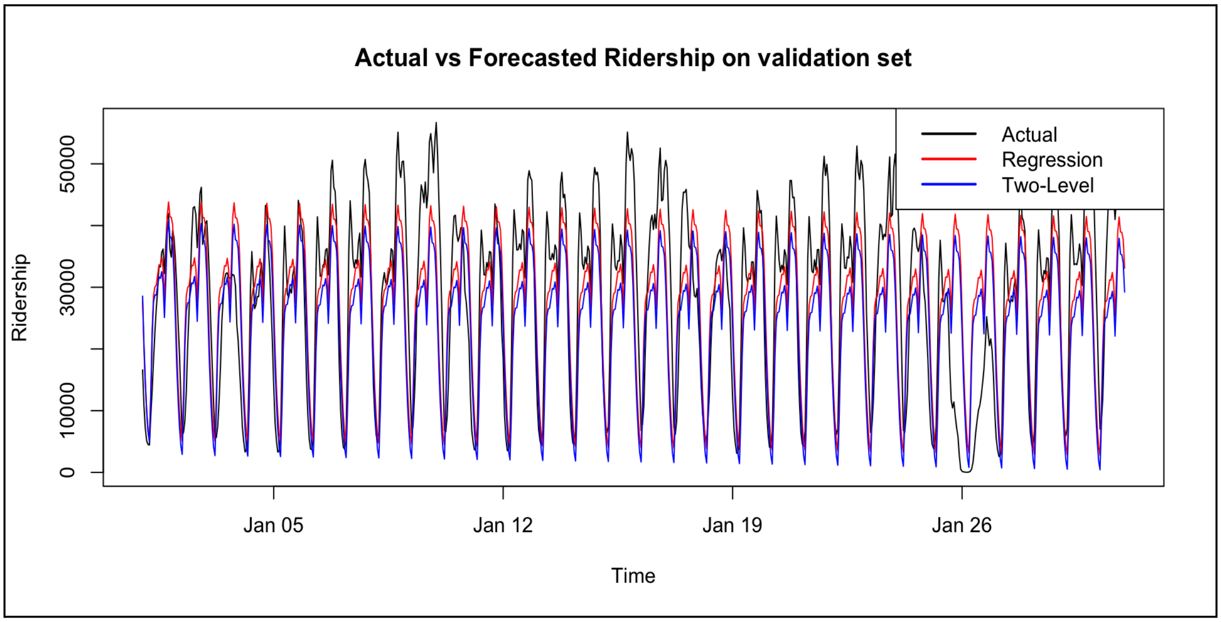
The ACF plot shows that there is still some trend and seasonality that needs to be captured since the autocorrelation coefficients at lags 1 and 24 are above the threshold.

Two-Level Forecast with trailing moving average of window 24

	Actual_Ridership	Regression_Forecast	Residual_MA_Forecast	Two_Level_Forecast
1	16601	27800.619	750.62067	28551.240
2	10811	20110.567	561.64560	20672.213
3	7117	14261.845	389.99742	14651.842
4	5090	9727.760	235.88450	9963.644
5	4518	6618.232	26.62741	6644.859
6	4441	5178.904	-212.18431	4966.720

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
Test set	4523.737	10525.45	8646.925	-135.3707	177.9465	0.9223106	24.6381

The MAPE is lower than the two-level forecast with the linear trend & seasonality model, but the RMSE is still high, suggesting that there are still patterns in the data that the model hasn't captured yet.



In the above plot, we can see that the model performs well while forecasting lower values but poorly while forecasting upper values (peaks).

The next model that we'll test will be the auto ARIMA model, and then we'll also evaluate the model performance on the whole data.

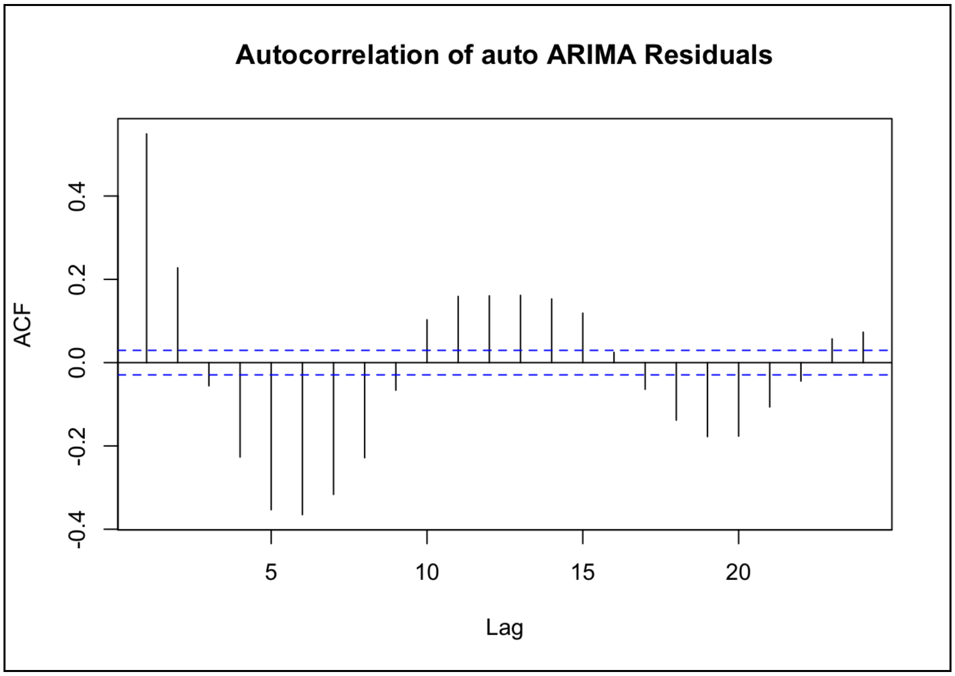
Auto ARIMA Model

```
> auto_arima_model <- auto.arima(train.ts)
> summary(auto_arima_model)
Series: train.ts
ARIMA(1,0,0)(0,1,1)[24]

Coefficients:
         ar1     sma1
      0.8941  -0.1959
s.e.  0.0067   0.0415

sigma^2 = 13700233:  log likelihood = -42550.19
AIC=85106.37   AICc=85106.38   BIC=85125.55

Training set error measures:
                    ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
Training set -4.774441 3690.529 2433.271 -1.052061 12.99655 0.4924397 0.5491684
```

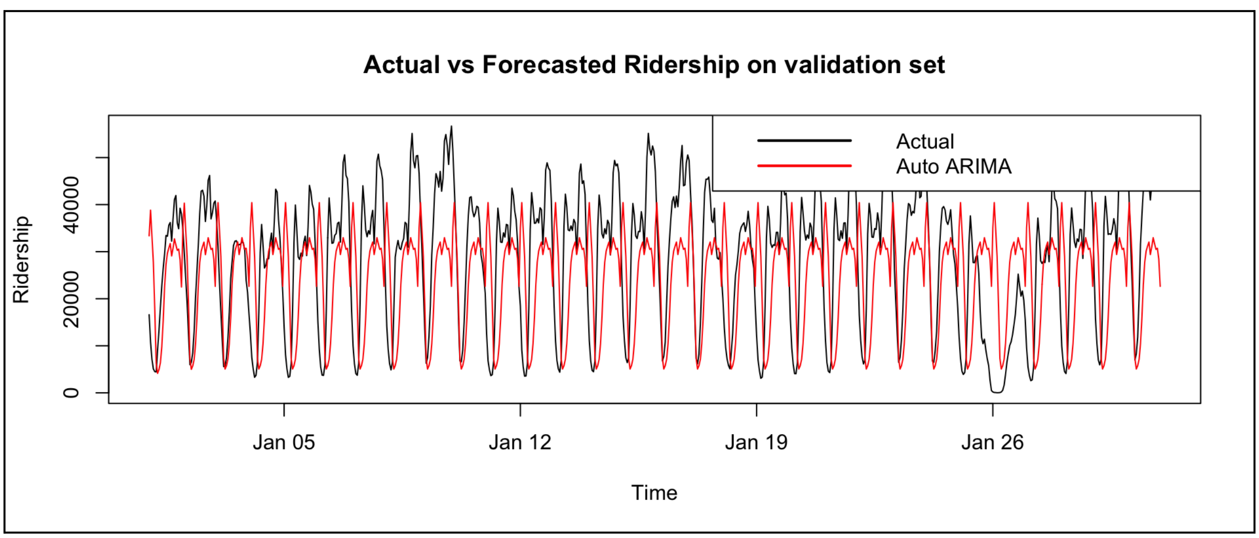
The selected ARIMA model for the training data is ARIMA(1,0,0)(0,1,1)[24], which consists of an AR(1) term and a seasonal MA(1) component with a seasonal period of 24. The AR(1) coefficient is 0.8941, indicating a strong dependence on the previous value, while the seasonal MA(1) coefficient is -0.1959, suggesting a minor seasonal adjustment.

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	-4.774441	3690.529	2433.271	-1.052061	12.99655	0.4924397	0.5491684	NA
Test set	4187.589951	15870.549	12781.414	-545.191671	593.10395	2.5866732	0.8784478	110.1935

The large discrepancy between the MAPE and RMSE in the training and test sets suggests that the model is likely overfitting to the training data. The model is performing well on the training set but poorly on the validation set.



The auto ARIMA model(1,0,0)(0,1,1)[24] captures the seasonality significantly; however, autocorrelation of coefficients larger than 0.4 in lag 1 is significant, so the trend is still not addressed. We can see the below plot for the confirmation because we can see that the model is capturing the repeating patterns correctly, but the trend is not getting captured.



Holt Winter's Model - Advanced Exponential Smoothing

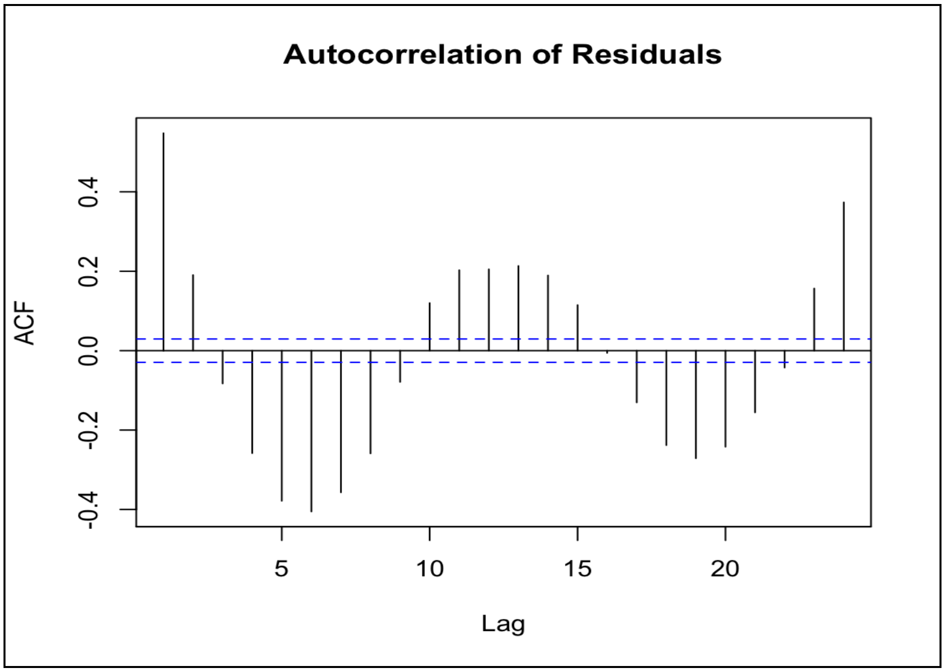
Next, we are going to use exponential smoothing with the Holt Winter's Model since it is also a good model to capture the trend and seasonality. We are going to use the automated selection of error, trend, and seasonality parameters with the 'ZZZ' parameter in the ets() function.

```
> hw.ZZZ <- ets(train.ts, model = "ZZZ")
> summary(hw.ZZZ)
ETS(A,N,A)

Call:
ets(y = train.ts, model = "ZZZ")

  Smoothing parameters:
    alpha = 0.9999
    gamma = 1e-04

  Initial states:
    l = 30827.9535
    s = 7293.074 11865.3 13054.99 13121.47 15525.97 13308.02
           5512.544 -625.4846 4673.906 6478.109 5215.734 5292.637 3592.838 1913.074 1644.282 223.2445 -5618.072 -15744.49 -2302
7.71 -21762.55 -18643.76 -14100.42 -8248.158 -944.5484

  sigma:  3465.709

     AIC     AICc      BIC
109694.8 109695.2 109867.6

Training set error measures:
                    ME     RMSE      MAE       MPE     MAPE      MASE      ACF1
Training set -3.942047 3455.547 2606.015 0.6215096 14.74236 0.5273992 0.5479304
```

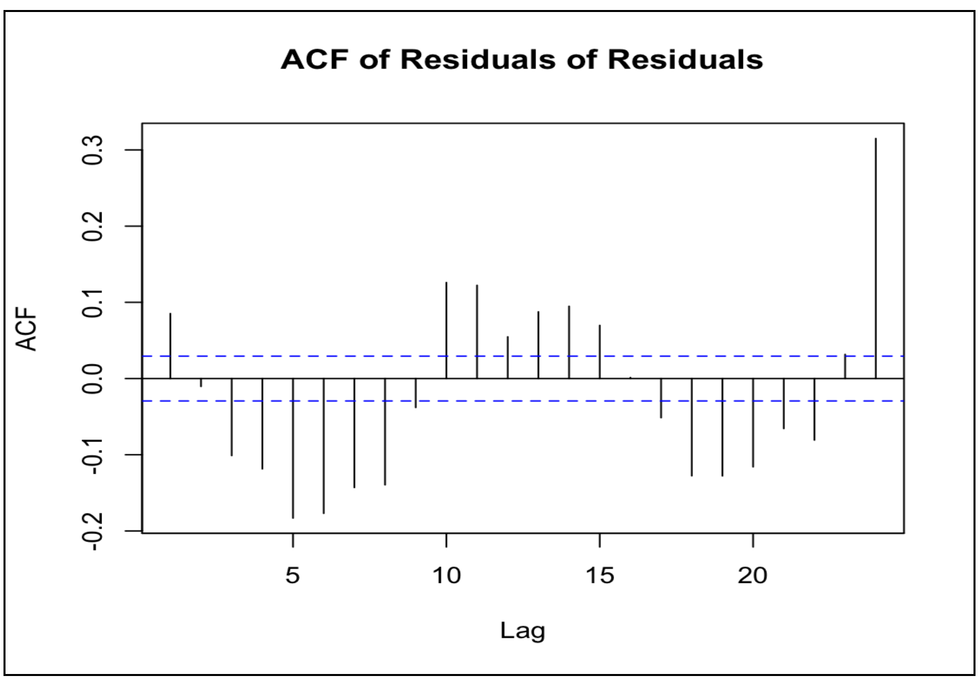
The parameters selected by the ets() function are additive error (A), no trend (N), and additive seasonality (A). The smoothing parameters include α (alpha) = 0.9999, which indicates strong reliance on recent observations, and γ (gamma) = 1e-04, suggesting minimal seasonal adjustment. The RMSE is 3,455.55, while the MAPE is 14.74%, indicating a good level of forecasting accuracy with some variance in predictions.

	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1	Theil's U
Training set	-3.942047	3455.547	2606.015	0.6215096	14.74236	0.5273992	0.5479304	NA
Test set	15473.667405	18137.197	16230.569	148.5304343	209.45447	3.2847052	0.9228906	50.69411

This model, too, is overfitting the data.



The ACF plot suggests that the trend and seasonality is still not fully captured by the model because the lag 1 and 24th coefficients are well above the horizontal threshold. We can try a two-level forecast with the AR(1) model for the Holt Winter's residuals.

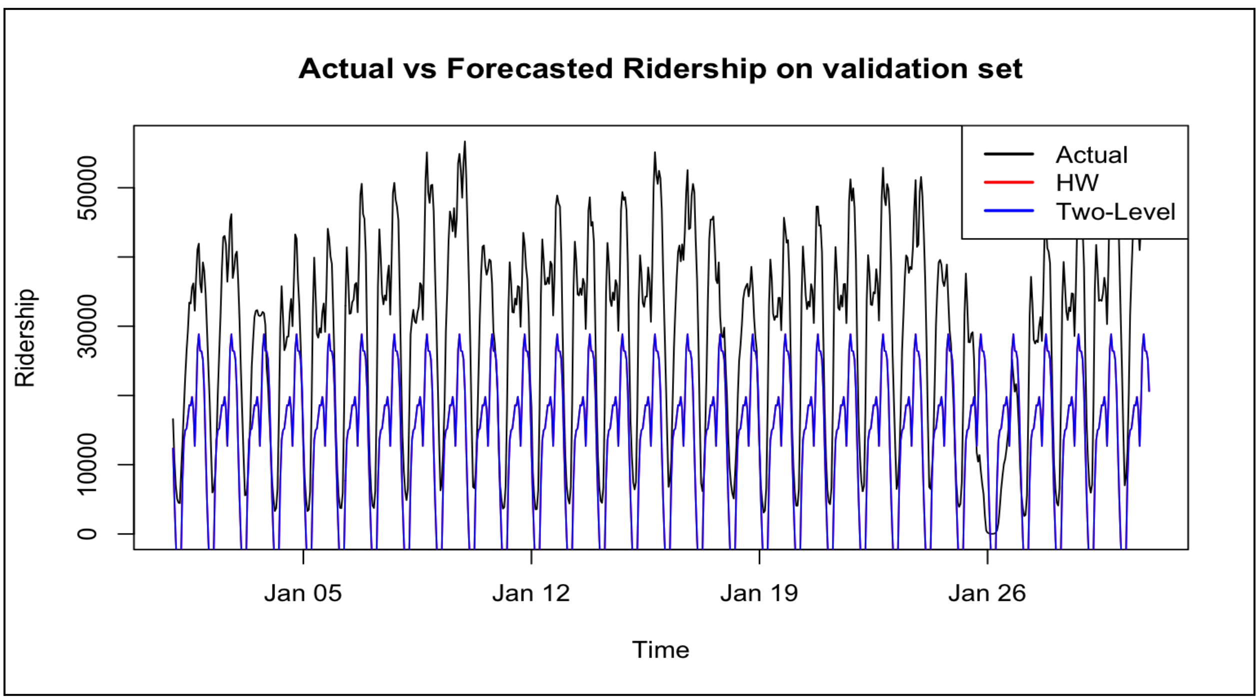


Above is the ACF plot of residuals of AR(1) model (residuals of residuals), and we can see that even with a two-level forecast, the model is not able to fully capture the trend and seasonality.

	ME	RMSE	MAE	MPE	MAPE	ACF1	Theil's U
Test set	15477.59	18140.48	16234.07	148.6329	209.5297	0.9228962	50.71604

	Actual_Ridership	HW_Forecast	AR1_Residual_Forecast	Two_Level_Forecast
1	16601	12388.6111	-40.529466	12348.0816
2	10811	5071.7676	-23.969122	5047.7985
3	7117	-773.3317	-14.876799	-788.2085
4	5090	-5316.4551	-9.884733	-5326.3398
5	4518	-8435.2549	-7.143880	-8442.3988
6	4441	-9703.8499	-5.639037	-9709.4890

The performance in this case is even worse than the other models since it's forecasting negative values for the ridership, which cannot be the case ever.



ARIMA(3,1,2)(1,1,2) Model

```
ARIMA(3,1,2)(1,1,2)[24]

Coefficients:
         ar1      ar2     ar3      ma1     ma2     sar1     sma1     sma2
      1.6088  -0.8266  0.0471  -1.2181  0.2561  -0.0471  -0.6104  -0.3377
s.e.  0.1028   0.1490  0.0653   0.1006  0.0989   0.0441   0.0407   0.0379

sigma^2 = 6049032:  log likelihood = -40759.11
AIC=81536.22   AICc=81536.26   BIC=81593.76

Training set error measures:
                     ME     RMSE      MAE        MPE     MAPE      MASE         ACF1
Training set -2.431018 2450.321 1678.037 -0.7499917 7.581067 0.3395972 4.322969e-05
```

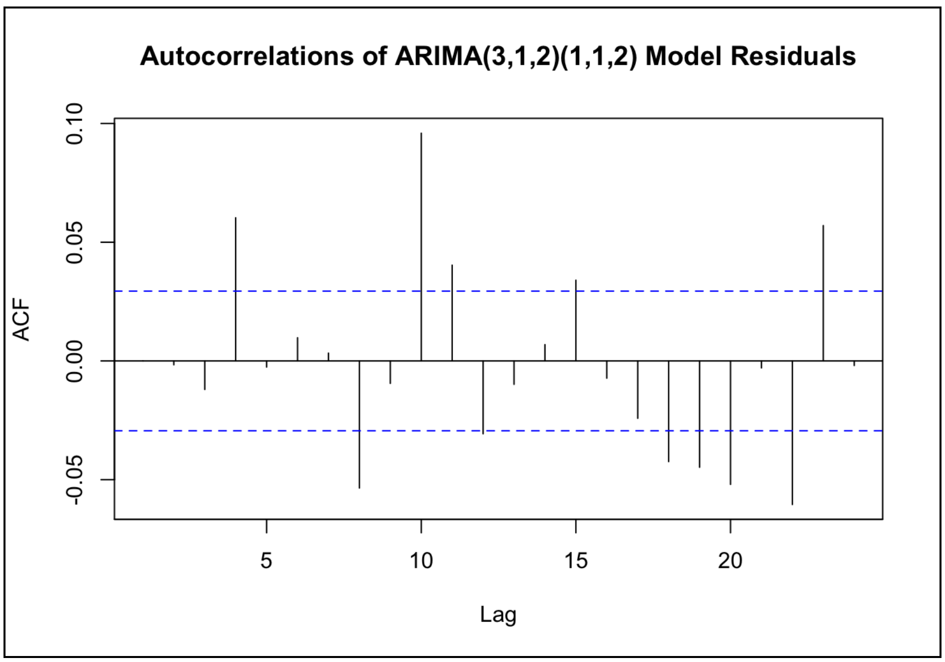
The ARIMA model fitted to the training data is an ARIMA(3,1,2)(1,1,2)[24], which includes three autoregressive terms, one differencing, two moving average terms, and seasonal components with a period of 24. We experimented with various combination of parameters for the ARIMA model and came up with the above one.

The non-seasonal part of the model, denoted by (3,1,2), includes first differencing (d=1) to remove linear trends and stabilize the mean of the time series. It also incorporates three autoregressive (AR) terms to model relationships between current and past observations, capturing short-term dependencies, and two moving average (MA) terms to account for unexpected shocks or noise in the data. On the seasonal side, represented by (1,1,2)[24], the model uses seasonal differencing (D=1) with a period of 24 to eliminate recurring daily patterns, which is useful for hourly data. Additionally, it includes one seasonal AR term to address cyclical dependencies over 24-hour intervals and two seasonal MA terms to model residual seasonal shocks.

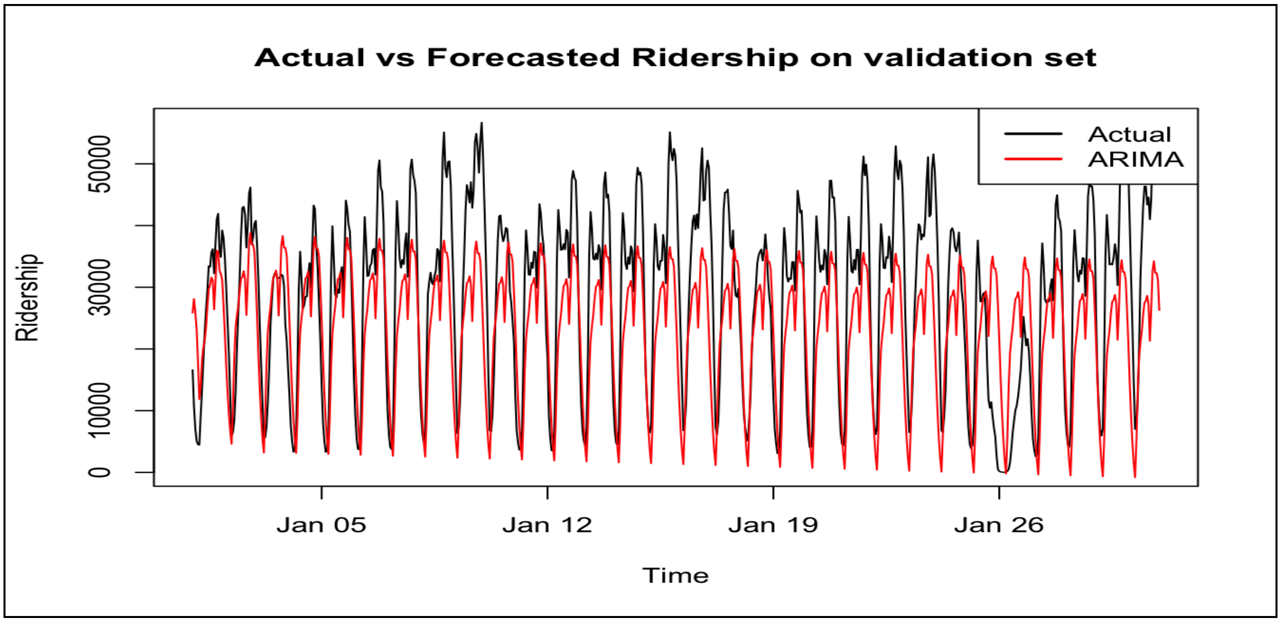
```
               ME     RMSE      MAE      MPE    MAPE  ACF1 Theil's U
Test set 5554.895 11215.91 9212.241 -125.672 173.971 0.922    21.649
```

	Actual_Ridership	ARIMA_forecast
1	16601	25844.76
2	10811	28117.09
3	7117	25983.57
4	5090	23224.61
5	4518	17929.94
6	4441	11838.97

Even though the MAPE is low we can see a huge difference between the actual and the forecasted values. This again suggests that the model is overfitting the data since it is performing well on the training data but not on the validation set.



If we observe the above ACF plot of the residuals we can see that the model effectively handles both trend and seasonality because the autocorrelation coefficients at lags 1 and 24 are almost reduced to 0. This means that the models is able to capture the trend & seasonality and we can also try to experiment with a two-level forecast using the above ARIMA model to see if there are any changes or not.



Comparing the accuracies of all the models on the validation set

```
> print(accuracy_table)
                           Model     MAPE      RMSE
1     Linear Trend & Seasonality 327.8846  9651.374
2  Quadratic Trend & Seasonality 230.1269  9610.918
3                         HW-ZZZ 209.4545 18137.197
4  Two-Level Forecast (LTS+MA-12) 222.1511 10032.379
5  Two-Level Forecast (QTS+MA-12) 177.9465 10525.446
6 Two-Level Forecast (HW-ZZZ+AR1) 209.5297 18140.484
7                     Auto Arima 593.1039 15870.549
8             ARIMA(3,1,2)(1,1,2) 173.9709 11215.908
```

From the above table, we can see that the Two-Level Forecast (Quadratic Trend & Seasonality + Trailing Moving Average of window 24) and ARIMA(3,1,2)(1,1,2) have a relatively low MAPE and RMSE and hence are the best models out of all. Even though these are the best models out of all, they're still not the ideal models in terms of forecasting into the future as they have a relatively high RMSE and the issue of overfitting.

Step 8: Implementing Forecast

Based on the above model selection, we chose the best two models were Two-Level Forecast (Quadratic Trend & Seasonality + Trailing Moving Average of window 24) and ARIMA(3,1,2)(1,1,2).

On the entire data set, ARIMA(3,1,2)(1,1,2) demonstrated significantly superior performance, with a MAPE of 9.8%, compared to 62.2% for the Two-Level model. Additionally, its RMSE of 2,475 was much lower than the 7,905 for the Two-Level model. Furthermore, the ACF1 value of -0.015 in ARIMA suggests minimal autocorrelation in the residuals, whereas the 0.9 ACF1 in the Two-Level model indicates a strong correlation, which is undesirable for reliable forecasting.

```
> round(accuracy(quadratic_trend_seasonality_model_full$fitted.values, taxi_ts), 3)
         ME     RMSE      MAE    MPE   MAPE  ACF1 Theil's U
Test set  0 7905.251 5870.894 -46.03 62.208 0.903    18.882
> round(accuracy(tot_train.arima.seas.pred$fitted, taxi_ts), 3)
            ME    RMSE      MAE    MPE MAPE   ACF1 Theil's U
Test set 6.826 2475.48 1714.715 -2.069  9.8 -0.015     5.053
```

Given these results, ARIMA(3,1,2)(1,1,2) is confirmed as the optimal model for forecasting future taxi ridership.

ARIMA(3,1,2)(1,1,2) on Entire dataset (taxi_ts)

To forecast into the future, we first need to combine the training data and the validation data. After this step, we run our prediction into the future 24 hours using our best model.

Forecast into the future 24 periods:

```
> print(forecast_table_full)
                  Time ARIMA_forecast_future
1  2015-01-31 23:00:00              48535.66
2  2015-02-01 00:00:00              40543.64
3  2015-02-01 01:00:00              32463.43
4  2015-02-01 02:00:00              24937.10
5  2015-02-01 03:00:00              17856.50
6  2015-02-01 04:00:00              12961.53
7  2015-02-01 05:00:00              16859.62
8  2015-02-01 06:00:00              23911.81
9  2015-02-01 07:00:00              30630.04
10 2015-02-01 08:00:00              35823.53
11 2015-02-01 09:00:00              38711.93
12 2015-02-01 10:00:00              42278.48
13 2015-02-01 11:00:00              45236.49
14 2015-02-01 12:00:00              45415.64
15 2015-02-01 13:00:00              45287.57
16 2015-02-01 14:00:00              45154.91
17 2015-02-01 15:00:00              40765.29
18 2015-02-01 16:00:00              44910.64
19 2015-02-01 17:00:00              52137.60
20 2015-02-01 18:00:00              53346.32
21 2015-02-01 19:00:00              48480.84
22 2015-02-01 20:00:00              48250.81
23 2015-02-01 21:00:00              48716.07
24 2015-02-01 22:00:00              44859.21
```

Conclusion

Our analysis confirms that major public events significantly impact taxi ridership, emphasizing the need for authorities to manage passenger traffic effectively. While most models performed well on the training set, their poor test performance indicates challenges in capturing trends or data limitations. To improve accuracy, we can explore alternative models or adjust time intervals. In our case, the best-performing models are Two-Level Forecast (Quadratic Trend & Seasonality + Trailing Moving Average of window 24) and ARIMA(3,1,2)(1,1,2). Given the Two-Level model's weaker performance on the full dataset, refining its approach to forecasting regression residuals could be explored further to enhance accuracy in future analyses.