Times Series Analysis and Forecasting Report

Le Tran Ngoc Minh * & Nguyen Duc Linh **

* minh.le2018@qcf.jvn.edu.vn
** linh.nguyen2018@qcf.jvn.edu.vn

May 28, 2019

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Time series analysis is an important and exciting research fields. By using the various of advanced statistical method, the model is constructed base on the historical data. By applying in economic, this technique could forecasting in near future and gain knowledge insight. Thus, the improvement could be archived. For instance, the prediction of clothing's revenue, the analytic can find trend of fashion.

Different from another forecasting like regression, the time series uses the relationship between previous data point, rather than finding more explanatory factor. This leads to two school of analysis, the *frequency-domain* and *time-domain*. In this project, we will focus on the *time-domain* class.

Three models we are going to use for analysis are Linear Regression, Holt-Winters and ARIMA. Linear Regression, which is the main method in regression analysis, is widely used to capture the relationship between predictor and target variable. When it is applied into time series analysis, the only purpose is to capture the trend and seasonal. However, in this report, we want to use more attributes to explain the model. Thus, we combine with the trend and seasonal predictor with special attributes with bike sharing dataset. Then, we compare and evaluate it with the results of Holt-Winters and ARIMA.

This report is organized as follows. In chapter 2, we introduce basic formation of time series, and the model we used. In Chapter 3, we aim to describe our work flow step by step. Finally, chapter 4 presents the results and chapter 5 points out our conclusions. The reference and the appendix can be found at the end.

Chapter 2

Literature Review

2.1 Definition and Properties of Time Series

According to Cochrane [4], the time series could defined as follow:

Definition 2.1. *A* **Univariate Time Series** *is a sequence of measurements of the same random variable X collected over time t such that:*

$$X_t \in \mathbb{R}, \quad \forall t \in \mathbb{Z}.$$

The measurements taken during an event in a time series are arranged in a proper chronological order. A time series can be continuous or discrete. In a continuous time series observations are measured at every instance of time, whereas a discrete time series contains observations measured at discrete points of time [2].

The procedure of finding a time series to a proper mathematical model is called **Time Series Analysis**. There are two main aim of Time Series Analysis:

- Identifying the nature of the phenomenon represented by the sequence of observation.
- Forecasting (predicting future values of the time series variable).

The time series contain three components: seasonal, trend, and cycle [1]:

- **Seasonal** component refers to fluctuations in the data related to calendar cycles,for instance, quarter or month of the year.
- **Trend** component is the overall pattern (increase or decrease) of the series.
- **Cycle** component consists of decreasing or increasing patterns that are not seasonal. Usually, trend and cycle components are grouped together.

We could assume that the time series X_t could be decomposition as additive:

$$X_t = S_t + T_t + R_t$$

or multiplicative:

$$X_t = S_t \times T_t \times R_t.$$

where S_t is the seasonal component, T_t is the trend-cycle component, and R_t is the remainder component [1].

A common assumption in many time series techniques is that the data are **Stationary**. According to [7] we have definitions:

Definition 2.2. *The time series X_t, $t \in \mathbb{Z}$ is said to be* **Weakly Stationary** *or* **Second Order Stationary** *if and only if:*

(i) $Var(X_t) < \infty, \quad \forall t \in \mathbb{Z}$

(ii) $E(X_t) = \mu, \quad \forall t \in \mathbb{Z}$

(iii) $Cov(X_s, X_r) = Cov(X_{s+k}, X_{r+k}) = \gamma(k), \quad \forall s, r, k \in \mathbb{Z}$

Definition 2.3. *A time series X_t is called* **Strictly Stationary** *if and only if the random vectors $(X_{t_1}, ..., X_{t_n})^T$ and $(X_{t_{1+\tau}}, ..., X_{t_{n+\tau}})^T$ have the same joint distribution for all sets of indices $\{t_1, ..., t_n\}$ and for all integers τ and $n > 0$. It is written as*

$$(X_{t_1}, ..., X_{t_n})^T \overset{d}{=} (X_{t_{1+\tau}}, ..., X_{t_{n+\tau}})^T,$$

where $\overset{d}{=}$ means "equal in distribution".

From the assumption above, the stationary time series are easy to forecast because the value of future is dependence and have the similar statistical properties with the past. Unfortunately, in the economic and financial field, the time series data is far from stationary. However we could using the mathematics method, which will be described later, to transform the former data into approximated stationary process.

2.2 Forecasting Models

2.2.1 Smoothing

Simple Exponential Smoothing
This method aim to estimate the values of series at times $T + k$:

$$\hat{X}_{T+k} = (1-\beta)\sum_{i=0}^{T-1} \beta^i x_{T-i} \ , \ \forall k \geq 1$$

where T observation x_i are available, β is the smoothing factor, $\beta \in (0, 1)$ [8].

The forecasting point are calculated using weighted averages, where the weights decrease exponentially as observations come from further in the past. Generally speaking, the smallest weights are associated with the oldest observations [1].

The smoothing factor could be find by optimize the sum of square error function:

$$\begin{aligned} &\underset{\beta}{\arg\min} \ \sum_{t=1}^{T-1} (x_{t+1} - (1-\beta)\sum_{i=0}^{t-1} \beta^i x_t - i)^2 \\ &\text{subject to: } \ 0 < \beta < 1 \end{aligned}$$

where x_{t+1} is the real value obtain at time t+1.

This method is suitable for forecasting data with no clear trend or seasonal pattern.

Double Exponential Smoothing
To detect trend, the double exponential smoothing evaluate from simple smoothing by adding the slope k into the estimation X_{T+k}:

$$\hat{X}_{T+k} = \hat{a}_1(T) + k\hat{a}_2(T), \ \forall k \geq 1$$

The constants a_1 and a_2 could be defined differently by various methods

Brown's Method This is the double exponential smoothing recommended by Brown. The constants a_1 and a_2 could be defined as following:

$$\begin{cases} S_1(T) = (1-\beta)\sum_{j=0}^{t-1} \beta^j x_{T-j} \\ S_2'(T) = (1-\beta)\sum_{j=0}^{t-1} \beta^j j x_{T-j} \\ \hat{a}_1(T) = (1+\beta)S_1(T) - (1-\beta)S_2'(T) \\ \hat{a}_2(T) = (1-\beta)S_1(T) - \dfrac{(1-\beta)^2}{\beta} S_2'(T) \end{cases}$$

Holt-Winters' Method This is the exponential smoothing recommended by Holt and Winters [6] [9]. The constants a_1 and a_2 could be defined as following:

- In the **Linear Trend** case (no seasonal components), this is the double exponential smoothing. We define $a_1(T) = \ell_t$, $a_2(T) = b_t$, the ℓ_1 and b_t imply the level and trend coefficients.

$$\begin{array}{rr} \text{Forecast equation} & \hat{x}_{t+k} = \ell_t + hb_t \\ \text{Level equation} & \ell_t = \alpha x_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\ \text{Trend equation} & b_t = \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1}, \end{array}$$

- In the **Additive Seasonal** case, the smoothing become triple exponential with the seasonal term:

$$\begin{array}{rr} \text{Forecast equation} & \hat{x}_{t+k} = \ell_t + hb_t + s_{t+h-m(k+1)} \\ \text{Level equation} & \ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\ \text{Trend equation} & b_t = \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1} \\ \text{Seasonal equation} & s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}, \end{array}$$

- In case of **Multiplicative Seasonal**, we also have triple exponential smoothing:

$$\begin{array}{rr} \text{Forecast equation} & \hat{x}_{t+k} = (\ell_t + hb_t)s_{t+h-m(k+1)} \\ \text{Level equation} & \ell_t = \alpha\dfrac{y_t}{s_{t-m}} + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\ \text{Trend equation} & b_t = \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1} \\ \text{Seasonal equation} & s_t = \gamma\dfrac{y_t}{(\ell_{t-1} + b_{t-1})} + (1-\gamma)s_{t-m} \end{array}$$

2.2.2 Regression

The aim of Regression method is fitting the point by a function $f(z_t)$. Some common functions is :

$$\begin{aligned}
f(z_t) = P(z_t) & \qquad P(.) \text{ is the polynomial with } n \text{ degree} \\
a \log z_t + b & \qquad \text{logarithm function} \\
\frac{1}{a + b z_t} & \qquad \text{reciprocal function} \\
a b^{z_t}; a b^{z_t} + c & \qquad \text{exponential and modified exponential function} \\
a z_t^b; a z_t^b + c & \qquad \text{power and modified power function} \\
\frac{a}{1 + b c^{z_t}} & \qquad \text{logistic function} \\
\exp\left(a b^{z_t} + c\right) & \qquad \text{Gompertz function}
\end{aligned}$$

where z_t is the preditor. The fitting method is minimizing the least square error:

$$\underset{a_1, \ldots a_m}{\arg\min} \sum_{i=1}^{n} (x_i - f(z_{t_i}))^2$$

where $a_j,\ j \in [1, m]$ is the parameter of functions $f(z_t)$. z_t could be generalize by vector if we have a set of predictor.

2.2.2.1 Linear Regression

Simple Linear Regression
In the simplest case, the regression model allows for a linear relationship between the forecast variable x and a single predictor variable z:

$$\begin{aligned}
x_t &= f(z_t) + \varepsilon_t \\
&= \beta_0 + \beta_1 z_t + \varepsilon_t.
\end{aligned}$$

After minimizing least square error, we have the solution has the form:

$$\begin{aligned}
\widehat{\beta_1} &= \frac{\sum_{i=1}^{T} \left(x_i - \overline{x}\right) \left(z_i - \overline{z}\right)}{\sum_{i=1}^{T} \left(z_i - \overline{z}\right)^2} \\
\widehat{\beta_0} &= \overline{y} - \widehat{\beta_1} \overline{x}
\end{aligned}$$

Multiple Linear Regression
When there are two or more predictor variables $z_{1,t}, \ldots z_{k_t}$, the model is called a multiple regression model. The general form of a multiple regression model is:

$$\begin{aligned}
& x_t = \beta_0 + \beta_1 z_{1,t} + \beta_2 z_{2,t} + \cdots + \beta_k z_{k,t} + \varepsilon_t \\
& \mathbf{x} = \mathbf{Z} \boldsymbol{\beta} + \epsilon
\end{aligned}$$

where :

$$Z = \begin{pmatrix} 1 & z_{1,1} & \ldots & z_{k,1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & z_{1,T} & \ldots & z_{k,T} \end{pmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_k \end{pmatrix}, \quad \boldsymbol{\epsilon} = \begin{pmatrix} \epsilon_0 \\ \vdots \\ \epsilon_T \end{pmatrix},$$

The least square solution is:

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{Z}^{\mathrm{T}} \mathbf{Z}\right)^{-1} \mathbf{Z}^{\mathrm{T}} \mathbf{x}$$

Assumptions
Those two linear model is called **Ordinary Least Squares** model. The **OLS** have five critical assumption to make the meaningful result [3]:

- **Strict exogeneity** The expectation of error is 0.
- **Homoscedasticity** The variance of error is constants
- **Independence of errors** All the error is independent
- **Linear Dependence** All predictor will independent with each other.
- **Normality** The error is normally distributed.

Coefficient Significant Test
To testing each coefficient of linear model is significant or not. We using the t test with hypothesis and test statistic as follow:

$$H_0 : \beta_j = 0$$
$$H_1 : \beta_j \neq 0$$
$$test\_statistic = \frac{\hat{\beta}_j}{\operatorname{se}\left(\hat{\beta}_j\right)}$$

2.2.3 ARMA, ARIMA and SARIMA Model

Definition 2.4. *A* **Moving Average** *process X_t, order q is define by:*

$$X_t = \sum_{i}^{q} \beta_i \epsilon_{t-i}$$

and the shorthand are denote by MA(q) [7].

Definition 2.5. Autoregressive Models *order p, is a process where future values somehow depend on the recent past.*

$$X_t = \sum_{i=0}^{p} \alpha_i X_{t-i} + \epsilon$$

Both AR and MA models express different kinds of stochastic dependence. AR processes encapsulate a Markov-like quality where the future depends on the past, whereas MA processes combine el- ements of randomness from the past using a moving window. An obvious step is to combine both types of behaviour into an ARMA(p, q) model which is obtained by a simple concatentation [7]. Therefore we have:

Definition 2.6. ARMA Models *order p,q of the process X_t is define by:*

$$\begin{aligned} X_t &= \alpha_1 X_{t-1} + ... + \alpha_p X_{t-p} + \epsilon_t + \beta_1 \epsilon_{t-1} + ... + \beta_q \epsilon_{t-q}. \\ &= \sum_{i=1}^{p} \alpha_i X_{t-i} + \epsilon_t + \sum_{j=1}^{q} \beta_j \epsilon_{t-j} \end{aligned}$$

From its definition, we could see that an MA(q) process is **second-order stationary** for any $\beta_1, ..., \beta_q$. However the AR(p) and ARMA(p, q) models do not necessarily define **second-order stationary** time series. Therefore, we must transform the non-stationary into stationary times series by **differencing** procedure.

Definition 2.7. *The* **difference operator** ∇ *is given by*

$$\nabla X_t = X_t - X_{t-1}$$

These differences form a new time series ∇X *(of length* $n-1$ *) if the original series had length* n*). Similarly, the second-order difference:*

$$\nabla^2 X_t = \nabla(\nabla X_t) = X_t - 2X_{t-1} + X_{t-2}$$

and so on.

If our original time series is not stationary, we can look at the first order difference process ∇X, or second order differences $\nabla^2 X$, and so on. If we find that a differenced process is a stationary process, we can look for an ARMA model of that differenced process. This procedure turn ARMA into ARIMA process.

Definition 2.8. *The process* X_t *is said to be an* **autoregressive integrated moving average process** *ARIMA(p, d, q) if its dth difference* $\nabla^d X$ *is an ARMA(p, q) process.*

However, a problem with ARIMA is that it does not support seasonal data. ARIMA expects data that is either not seasonal or has the seasonal component removed. Therefore, the Seasonal ARIMA process (or SARIMA) are extended in order to modelling seasonal data [1].

Definition 2.9. *A* **seasonal ARIMA** *or* **SARIMA** *model is formed by including additional seasonal terms in the ARIMA models we have seen so far. It is written as follows:*

$$ARIMA(p,d,q)(P,D,Q)_m$$

Box-Jenkins' Method

In order to fitting the times series into ARIMA familiar models. We could use the Box-Jenkins three-step methods: **model identification**, **parameter estimation**, and **model diagnostics** [2].

- **Model identification**

  The first step in developing a model identification is to determine if the time series is stationary and if there is any significant seasonality that needs to be modelled. The stationary time series could be observed by series plot to see whether the mean or the variance of the time series changes over time or compute autocovariance on two different parts of the time series. If the two quantities from the different regions look very different then this provides some evidence of non-stationarity [7]. More technical approach is using the hypothesis testing for stationary, one of the test is **Augmented Dickey-Fuller Test**.

  The second step is decide which autoregressive or moving average component should be used in the model by plotting of the **Auto Correlation Functions (ACF)** and **Partial Auto Correlation Functions (PACF)** of the dependent time series to.

  Auto Correlation Function $\rho\colon k \mapsto \rho(k) = \dfrac{\gamma(k)}{\gamma(0)}$

  Partial Auto Correlation Functions $r\colon \mathbb{N} \mapsto \mathbb{R};\ r(p-n) = cor(X_n, X_p / X_{n+1}, \ldots, X_{p-1}), \quad p > n$

  $$= \frac{cov(X_n - X_n^*; X_p - X_p^*)}{\sqrt{Var(X_n - X_n^*)Var(X_p - X_p^*)}}$$

  ACF is used to identify order of MA term, and PACF for AR.The lag where ACF,PACF shuts off suddenly is the order of MA and AR.

- **Parameter Estimation**

  The parameters $\alpha_1, \ldots, \alpha_p, \beta_1, \ldots, \beta_q$ will be estimated in this step by using **Maximum Likelihood Estimation** method. Estimators are always sufficient, efficient, and consistent for Gaussian distribution and which are asymptotically normal with efficient for several non-Gaussian distribution.

- **Model Diagnostics**

  The model diagnostic is necessary to test the appropriateness of the selected model. Model selection can be made based on the values of certain criteria like **log likelihood**, **Akaike Information Criteria (AIC) / Bayesian Information Criteria (BIC)**

Akaike Information Criteria	$AIC = n(1 + \log 2\pi) + n \log \sigma^2 + 2m$
Bayesian Information Criteria	$BIC = -2 \log(L) + k \log(n)$

  If the model is not edaquate, repeated the procedure.

Chapter 3

Methodology

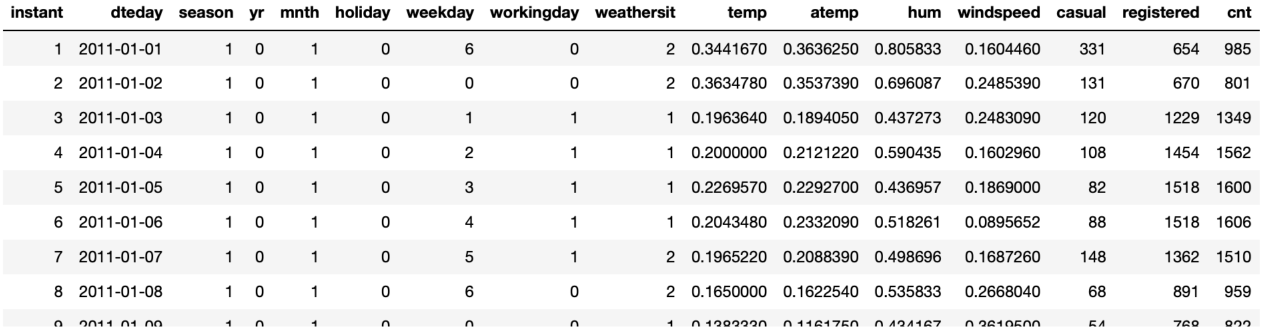
3.1 Dataset Information

In this empirical project we using the bike sharing dataset from [5]:

”Bike sharing systems are new generation of traditional bike rentals where whole process from membership, rental and return back has become automatic. Through these systems, user is able to easily rent a bike from a particular position and return back at another position. Currently, there are about over 500 bike-sharing programs around the world which is composed of over 500 thousands bicycles. Today, there exists great interest in these systems due to their important role in traffic, environmental and health issues.

Apart from interesting real world applications of bike sharing systems, the characteristics of data being generated by these systems make them attractive for the research. Opposed to other transport services such as bus or subway, the duration of travel, departure and arrival position is explicitly recorded in these systems. This feature turns bike sharing system into a virtual sensor network that can be used for sensing mobility in the city. Hence, it is expected that most of important events in the city could be detected via monitoring these data.”

The data has 16 attributes. The head of data is shown in figure 3.1. We will hold out the last 31 instances of this dataset for testing the accuracy of the model.



instant	dteday	season	yr	mnth	holiday	weekday	workingday	weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
1	2011-01-01	1	0	1	0	6	0	2	0.3441670	0.3636250	0.805833	0.1604460	331	654	985
2	2011-01-02	1	0	1	0	0	0	2	0.3634780	0.3537390	0.696087	0.2485390	131	670	801
3	2011-01-03	1	0	1	0	1	1	1	0.1963640	0.1894050	0.437273	0.2483090	120	1229	1349
4	2011-01-04	1	0	1	0	2	1	1	0.2000000	0.2121220	0.590435	0.1602960	108	1454	1562
5	2011-01-05	1	0	1	0	3	1	1	0.2269570	0.2292700	0.436957	0.1869000	82	1518	1600
6	2011-01-06	1	0	1	0	4	1	1	0.2043480	0.2332090	0.518261	0.0895652	88	1518	1606
7	2011-01-07	1	0	1	0	5	1	2	0.1965220	0.2088390	0.498696	0.1687260	148	1362	1510
8	2011-01-08	1	0	1	0	6	0	2	0.1650000	0.1622540	0.535833	0.2668040	68	891	959
9	2011-01-09	1	0	1	0	0	0	1	0.1383330	0.1161750	0.434167	0.3619500	54	768	822

Figure 3.1: Bike Sharing Data

3.2 Research Method

The research procedure are following as 4 step:

- **Analysis Time Series**: we will visualize the time series and tried to decomposition trend and seasonal components. If there were the outlier, the smoothing procedure is required.

- **Testing on Different Model**: Smoothing Exponential, Linear Regression and ARIMA model will be applied in data.
- **Compare and Evaluate Model**: The fitting model above will be evaluated by using criteria like the **Mean Square Error (MSE)**, **Mean Absolute Error (MAE)** and, **Mean Absolute Percentage Error (MAPE)** calculated on testing dataset.
- **Conclude**: From the information above, we can concluding the best model for data.

Chapter 4

Empirical Result

4.1 Exploratory Data Analysis

First, we conduct a *Data Exploratory Analysis* to analyze:

- Outliers total rental bikes count including both casual and registered.
- Data volatility.
- Whether trends are existed.

We begin by plotting date time as season format to see duration of every seasons.

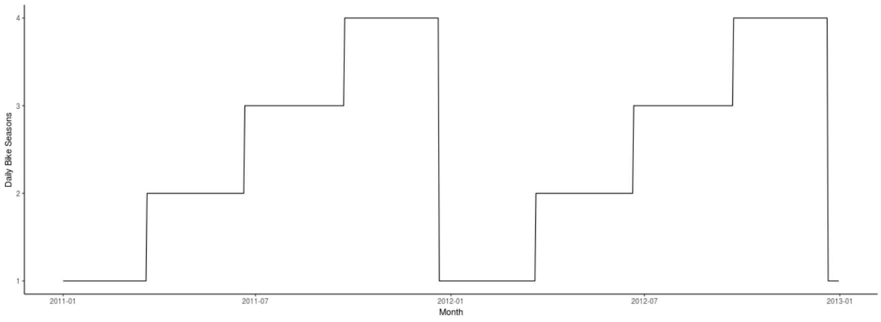


Figure 4.1: Total bike checkout in every seasons

We conclude that the season varies: from Spring to Winter, from January as starting and to December as ending. Next, we visualize the total rental bikes count including both casual and registered with date time index to see how our data distributed

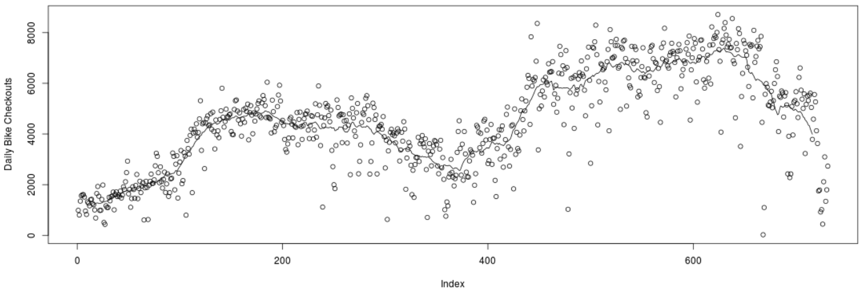


Figure 4.2: Daily bike checkout with Index

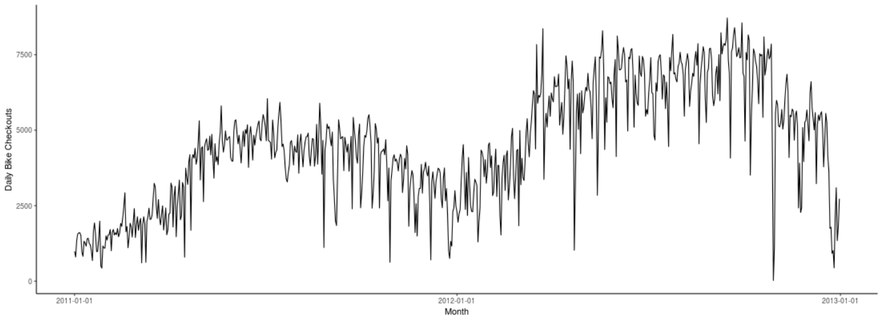


Figure 4.3: Daily bike checkout with Date

There is a lot of fluctuation of bike checkout from day to day. However, we can see some patterns here. Firstly, we see that the usage tends to *increase in summer*, but there is *low usage in winter*. So there must be trend here. Secondly, in some cases, the number of bicycles checked out dropped below 100 on day and rose to over 4,000 the next day. Thus, we will check for outliers that could bias the model by skewing statistical summaries.

To see how outliers are distributed in each month, we use box-plot to visualize them.

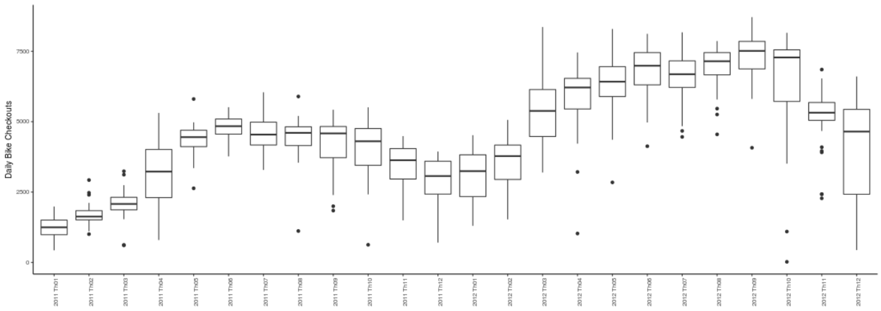


Figure 4.4: Daily bike checkout Box Plot with Month

By identifying outliers with Box Plot, we can remove outliers and replacing outliers using series smoothing and decomposition. Data Visualization after being cleaned is as shown in Figure 4.5

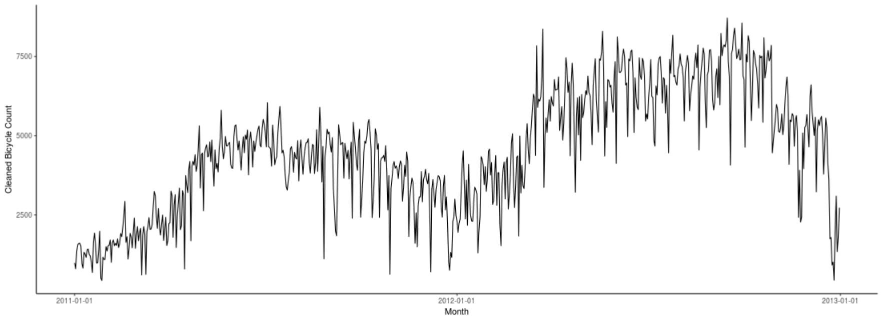
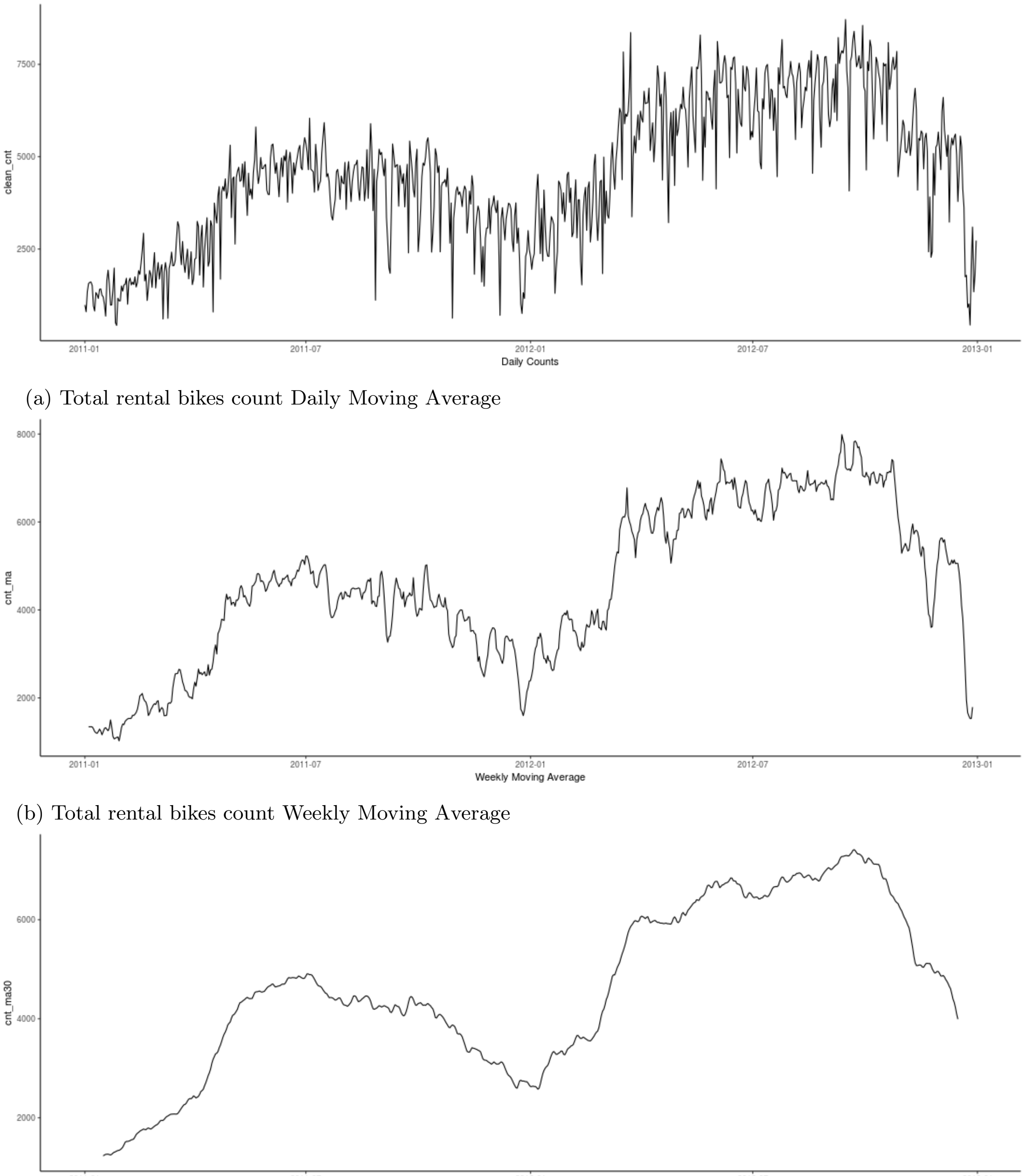


Figure 4.5: Cleaned daily bike checkout with Date

The daily data is still volatile even after we remove the outliers. We consider to use moving average to smooth out noisy fluctuations.

There are ups and downs with a general upward trend except in winter. The data is volatile so time series is not stationary. Also, the wider the rolling of the moving average, the smoother the series becomes. We can both use the weekly moving average or monthly moving average. For simplicity, we will model the smoothed series of *weekly moving average*.



(a) Total rental bikes count Daily Moving Average

(b) Total rental bikes count Weekly Moving Average

(c) Total rental bikes count Monthly Moving Average

Figure 4.6: Total rental bikes count with different Moving Average

4.2 Modeling

4.2.1 Linear Regression

4.2.1.1 Feature Selection

In this method, we only use the data cleaned from outliers. We want to conduct the linear model that is able to capture the time series of counting bike. Therefore, the attributes of dataset will be considered by checking correlation with *count* attribute.

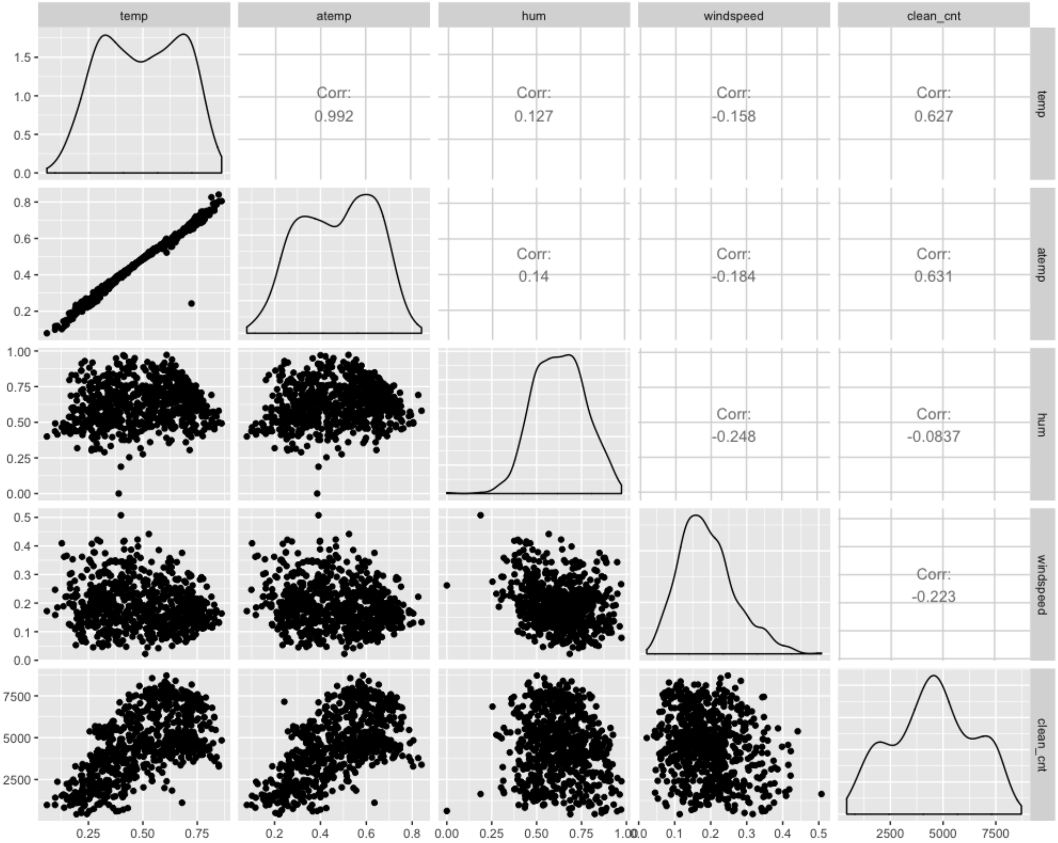


Figure 4.7: Correlation of Attributes

First, we consider the continuous data column. *casual* and *register* are the subset of bike count, so we do not use them. The correlation between them and counting bike are shown in 4.7. In the figure 4.7 we can conclude:

- The *temp* (temperature) and *atemp* (assumption temperature) are highly correlated, it is also suitable with the dataset description. Thus, it should not be the selected predictor together. atemp also has the higher correlation with number of counted bike.
- Others attributes also have significant correlation with target output.

Thus, we obtain three continuous predictors: **atemp, hum, windspeed**. For the group and indicator attributes: *season, yr, mnth, holiday, weekday, workingday, weathersit* . We choose *weathersit, season, holiday, workingday* as predictors because:

- In the exploratory section, we observe that the time series has the additive seasonal component and each period could be quarter. We want to capture that property. The *mnth* (month) interval is not clear in this situation. Thus, the season attribute is chosen.
- Heuristically, the *holiday* and *working* will affect on the counted bike.

- Using the **One-Way ANOVA** test in table 4.1, we conduct that *weathersit* and *season* is not significantly correlated with **atemp, hum, windspeed**.

Group	P-value of attributes		
	atemp	hum	windspeed
weathersit	0.0044	$< 2e-16$	0.00527
season	$< 2e-16$	$2.37e-07$	$1.16e-09$

Table 4.1: One-Way ANOVA testing

The type data of *holiday* and *workingday* is the binary (0,1) ,but *weathersit* and *season* is the group factor (1,2,3,4) . Thus we need to transform *weathersit,season* into dummies variables as table 4.2. Because each of them have 4 and 3 groups so we have 5 more dummies variables.

Finally, to capture linear trend , we will add the instant variable (as time). The final linear model is:

$$\begin{aligned} y = \beta_0 + \beta_1 x_{\text{instant}} + \beta_2 x_{\text{atemp}} + \beta_3 x_{\text{hum}} + \beta_4 x_{\text{windspeed}} + \\ \beta_5 x_{\text{weathersit.1}} + \beta_6 x_{\text{weathersit.2}} + \beta_7 x_{\text{weathersit.3}} + \\ \beta_8 x_{\text{season.1}} + \beta_9 x_{\text{season.2}} + \beta_{10} x_{\text{season.3}} \end{aligned}$$

4.2.1.2 Fitting and Diagnosing model

After running procedure in R, we have the coefficients' value and its significant level in table 4.3, the regression result is visualized in figure 4.8 and 4.9.

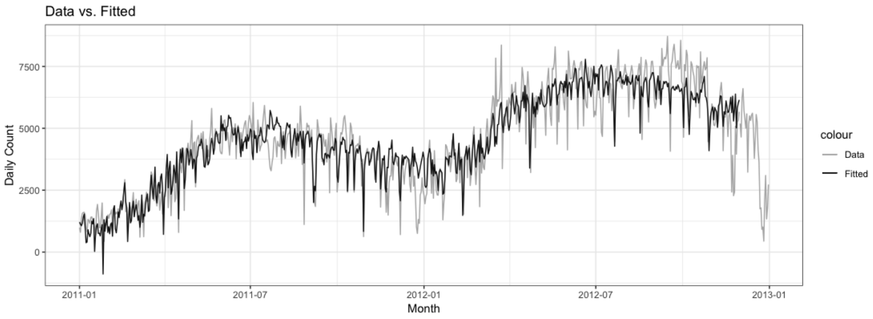


Figure 4.8: Fitted v.s Real Data Time Series

From the table, despite the high value of Adjusted R^2, the *season.3* and *workingday* is not significant. However, we must contain those two dummies variable.

Then, we check the assumption about residual. The result is show in 4.10 and 4.11. As the ACF plot of residual, there are significant correlations between the residual of model. Also, from the histogram and normal Q-Q plot, the residual is also not normally distributed. It is

Season Group	Season.1	Season.2	Season.3
1	1	0	0
2	0	1	0
3	0	0	1
4	0	0	0

Weathersit Group	weathersit.1	weathersit.2
1	1	0
2	0	1
3	0	0

Table 4.2: Dummies Variables

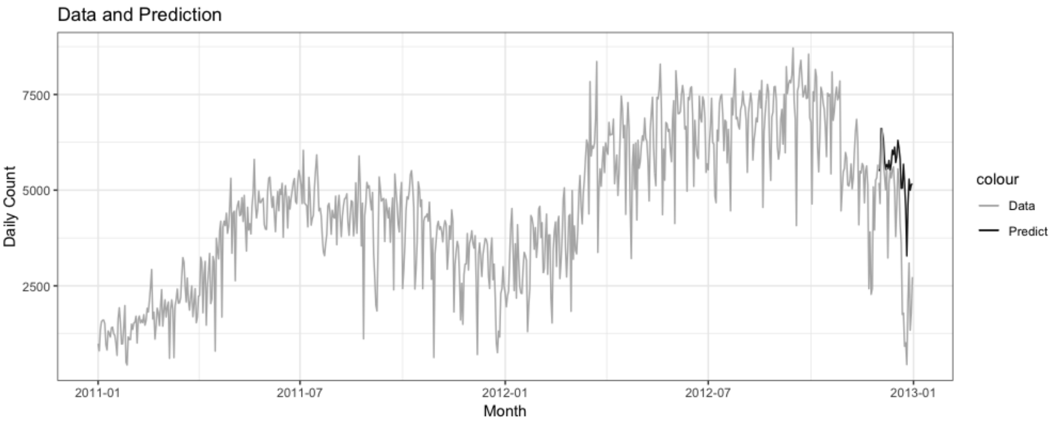


Figure 4.9: Predict v.s Real Data Time Series

Predictor	Coefficient
instant	5.829*** (0.168)
season.1	$-$223.058** (104.065)
season.2	643.384*** (99.327)
season.3	$-$50.160 (117.051)
weathersit.1	1,472.145*** (205.352)
weathersit.2	1,148.211*** (193.105)
holiday	$-$634.760*** (186.748)
workingday	77.926 (67.189)
atemp	5,011.562*** (330.122)
hum	$-$1,369.098*** (290.016)
windspeed	$-$2,203.394*** (425.727)
Constant	$-$85.385 (362.286)
Observations R^2 Adjusted R^2 Residual Std. Error F Statistic	700 0.832 0.829 795.050 (df = 688) 309.845*** (df = 11; 688)
Note:	*p<0.1; **p<0.05; ***p<0.01

Table 4.3: Linear Regression Results

violated the **Independence of errors** and also the **Normality** assumption. From that result, despite the constant term is not significant, the model still contain information which can utilize to analyze.

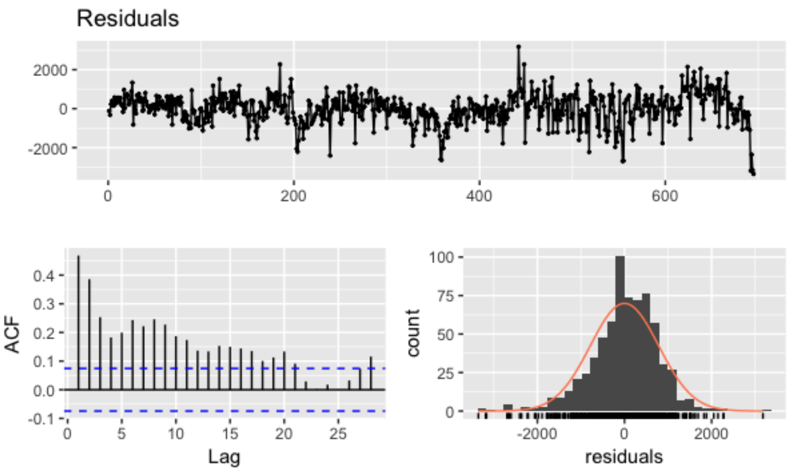


Figure 4.10: Linear Model Residual Checking

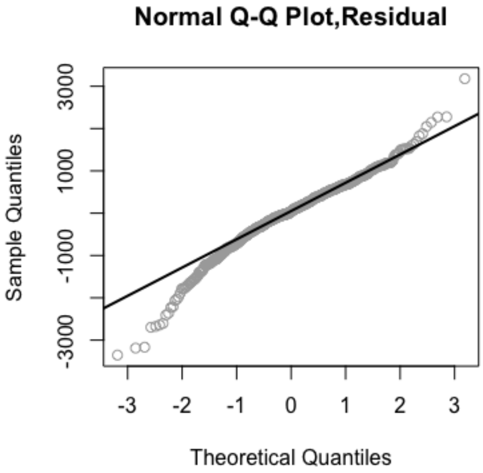


Figure 4.11: Normal QQ Plot for Linear Model Residual

4.2.2 Holt-Winters' Method

Because the time series have the additive seasonal component , in the exponential smoothing family, we will using triple exponential smoothing or Holt-Winters additive seasonal method. The weakly smoothing MA 7 (cleaned data) of bike count are used with this method. The result is shown in table 4.4 and figure 4.12, 4.13.

Smoothing parameters	
α	0.9429644
β	0.00216804
γ	1

Table 4.4: Holt-Winters' Smoothing Variable

As we can see, in both graph, the data is plotted in black-line, and the prediction is shown in grey. The darkest line shows the mean of prediction. Two lighter grey layers show 80% (the darker grey) and 95% confidence interval of the prediction.

Next, we check the residual of this model. The result is shown in 4.14 and 4.15. As we observe, the ACF plot of error is better than the linear model because it only has significant correlation on lag 7. The Ljung-Box test which is shown below also confirms the auto correlation

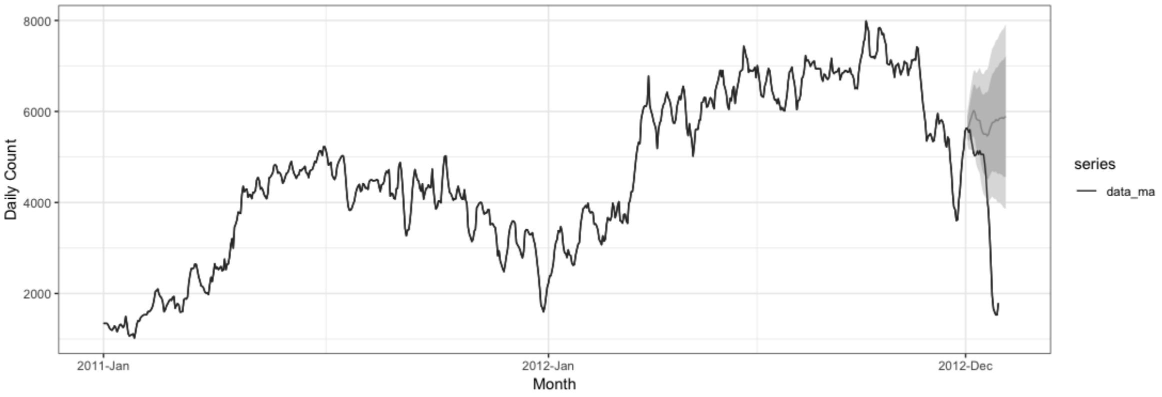


Figure 4.12: Holt-Winter's Result with the Weekly Moving Average

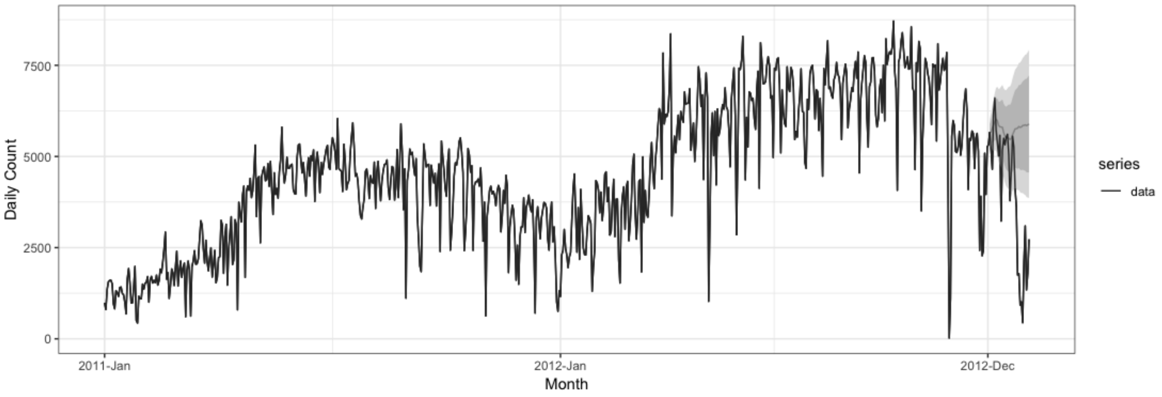


Figure 4.13: Holt-Winter's Result with Actual Data

of residual.

```
    Box-Ljung test

data:  hwforcast$residual
X-squared = 242.34, df = 20, p-value < 2.2e-16
```

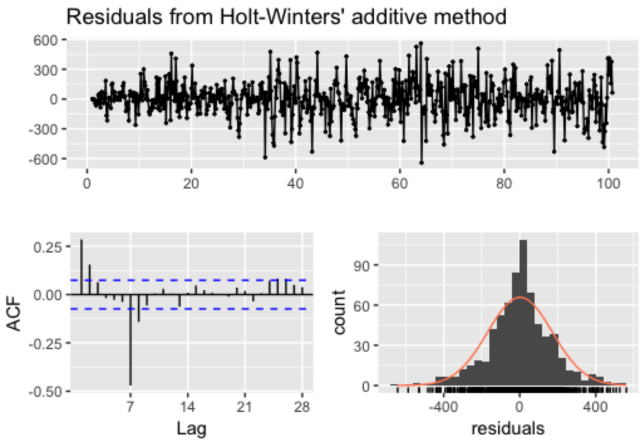


Figure 4.14: Holt-Winter's Residual Checking

Also, the Normal QQplot shows that, in two tail, the error does not follow the normal distribution. Therefore, the Holt-Winter is not a good model for this data.

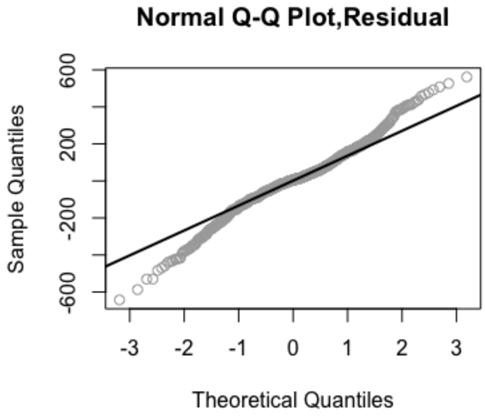
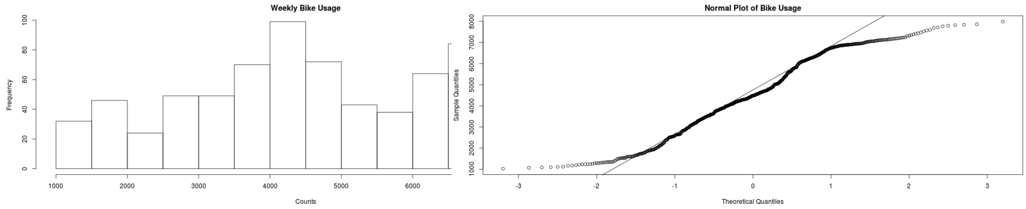


Figure 4.15: Holt-Winter's Residual Normal QQ plot

4.2.3 ARIMA

Firstly, we check if the weekly data follows a normal distribution by creating a normal probability plot



(a) Weekly bike usage histogram

(b) Weekly bike usage Q-Q plot

Figure 4.16: Weekly bike usage normal probability plot

We see that the weekly data does not follow a normal distribution since both tails are not linear.

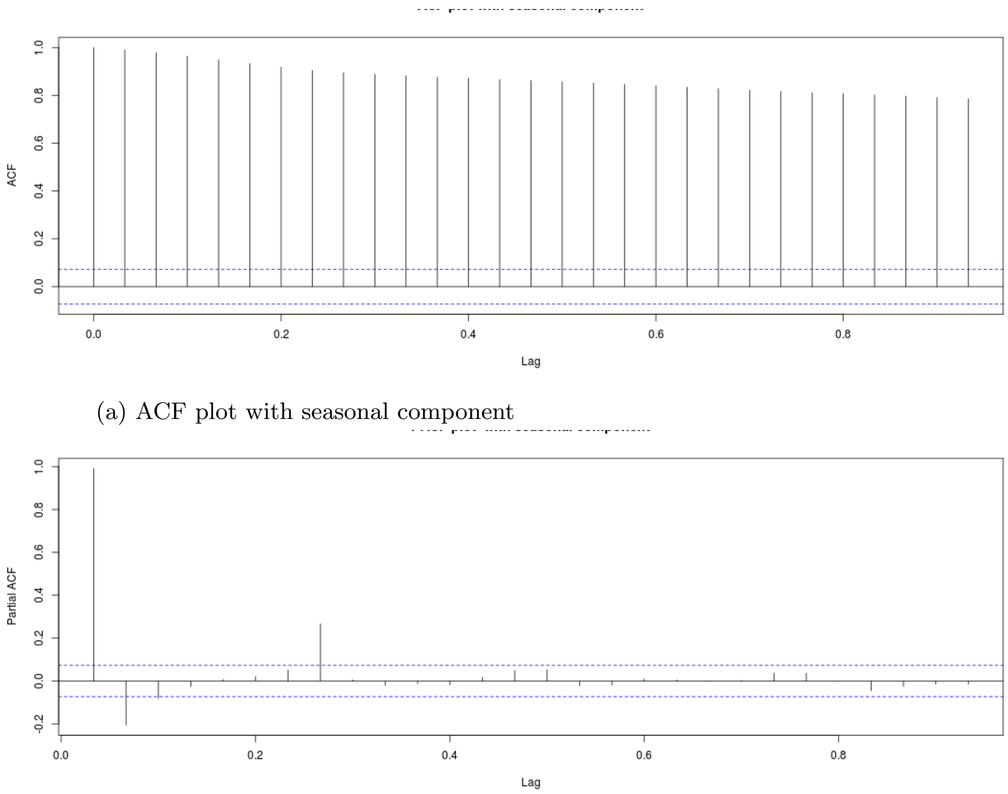
Then, we check for data stationarity by using Dickey-Fuller Test. Below table shows the result of Dickey-Fuller Test for weekly bike usage.

Augmented Dickey-Fuller Test	
data	weekly bike usage
Dickey-Fuller	-0.2557
Lag order	8
p-value	**0.99**
alternative hypothesis:	stationary

Table 4.5: Augmented Dickey-Fuller Test

We observe p-value is large so we cannot reject the null hypothesis. Thus, data is not stationarity. We can also check stationarity using acf and pacf plot.

Here, we cannot choose the MA(q) from ACF plot. That's why we will de-seasonalize and difference the series to remove seasonal and trend-cycle components. We calculate the seasonal component of the series using smoothing, and adjusts the original series by subtracting



(a) ACF plot with seasonal component

(b) PACF plot with seasonal component

Figure 4.17: ACF and PACF plot weekly bike usage

seasonality. Furthermore, we consider to use additive decomposition since the magnitude of the seasonal fluctuations in every summer and winter.

$$Y_t = S_t + T_t + R_t$$

where, Y_t is the cleaned weekly bike usage data by removing outliers and shift moving average of 7, T_t is the trend-cycle component and R_t is the remainder components.
Seasonal adjusted formulas and visualization Figure 4.18 are as following.

$$Y_t - S_t = T_t + R_t$$

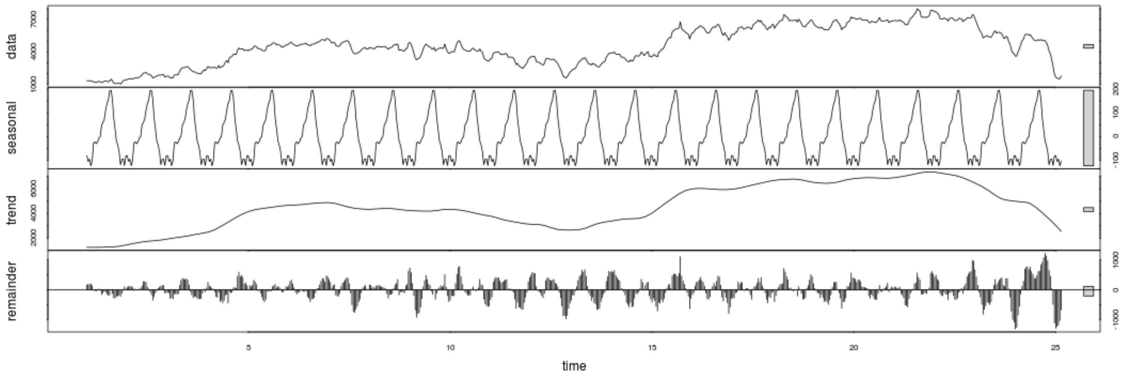


Figure 4.18: Seasonal adjusted weekly bike usage

Then, we consider to remove the trend-cycle components by differencing the series. Firstly, we can start with the order of $d = 1$ and re-evaluate whether further differencing is needed.

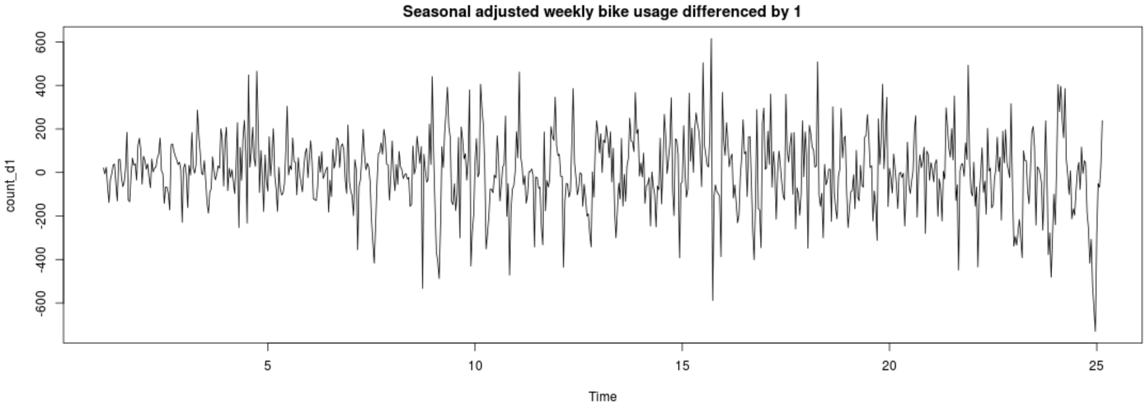


Figure 4.19: Seasonal adjusted weekly bike usage differenced by 1

We see that the oscillating pattern around 0 with no visible strong trend. This suggests that differencing of order 1 terms is sufficient and should be included in the model. Again, we use the Augmented Dickey-Fuller Test to check if it is stationarity.

Augmented Dickey-Fuller Test	
data	seasonal adjusted weekly bike usage differenced by 1
Dickey-Fuller	-9.9255
Lag order	8
p-value	**0.01**
alternative hypothesis:	stationary

Table 4.6: Augmented Dickey-Fuller Test 2

So, we conclue that the data is stationarity since we can reject the null hypothesis. For the next step, we plot ACF and PACF to determine the order of MA(q) and AR(p) term.

Following the rule that, spikes at particular lags of the differenced series can help inform the choice of p or q for our model.
We consider following lags are significant for MA(q): Lag 1, Lag 2, Lag 3, Lag 7, Lag 8 and following lags are significant for AR(p): Lag 1, Lag 2, Lag 6, Lag 7, Lag 8, Lag 13, Lag 14

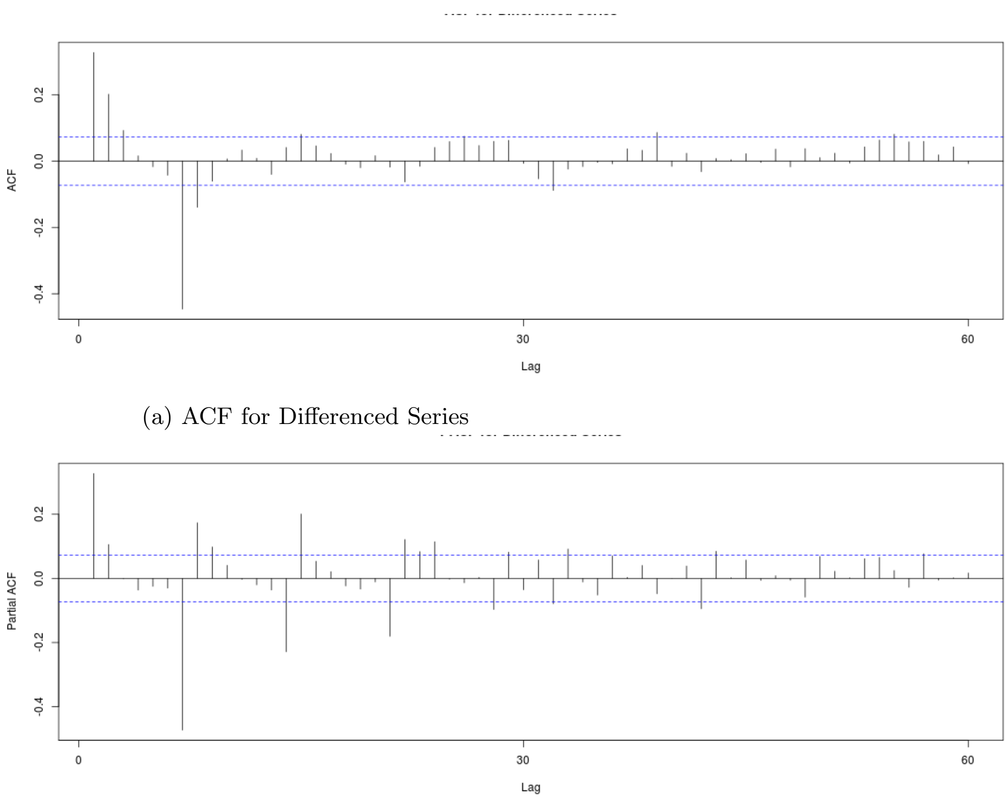
Then, we try to find an $ARIMA(p, d, q)$ model. Keeping parsimony principle in mind, we limit $p + d + q \leq 6$. We evaluate the model using SSE and AIC.

Model	AIC	SSE	p-value
ARIMA(0,1,0)	9508.759	21376391	0
ARIMA(0,1,1)	9447.215	19578285	1.446001e-05
ARIMA(0,1,2)	9427.638	19002032	0.02841553
ARIMA(0,1,3)	9424.381	18864103	0.224242
ARIMA(1,1,0)	9428.742	19083921	0.008932232
ARIMA(1,1,1)	9423.817	18901820	0.287052
ARIMA(1,1,2)	9424.348	18863342	0.2757869
ARIMA(2,1,0)	**9422.642**	**18871035**	**0.2980812**
ARIMA(2,1,1)	9424.643	18871052	0.2871248
ARIMA(3,1,0)	9424.64	18870976	0.2894379

Table 4.7: Simple ARIMA model with different lag p,d,q

We find that ARIMA(2,1,0) model has the smallest AIC value and ARIMA(1,1,2) model has the smallest SSE (sum of squared errors) value. Here, we choose model with smallest AIC, ARIMA(2,1,0).

Finally, we evaluate out model by examining ACF and PACF plots for model residuals. If



(a) ACF for Differenced Series

(b) PACF for Differenced Series

Figure 4.20: ACF and PACF for Differenced Series

ar1	ar2	SE 1	SE 2	sigma$\hat{2}$	log likelihood	AIC
0.2929	0.1055	0.0370	0.0370	26065	-4708.32	9422.64

Table 4.8: ARIMA(2,1,0)

model order parameters and structure are correctly specified, we would expect no significant autocorrelations present. We get model residual then ACF, PACF by subtracting real data with fitted ARIMA(2,1,0) with weekly bike usage.

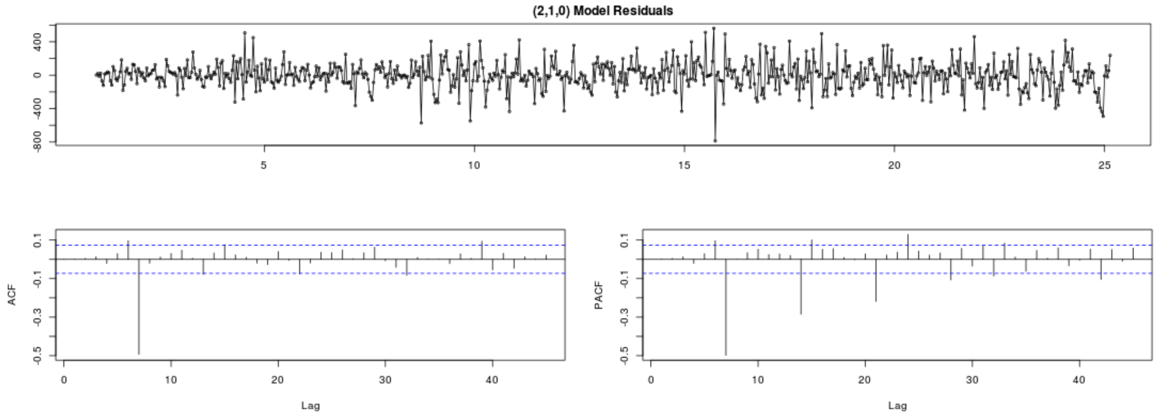


Figure 4.21: ARIMA(2,1,0) Residual, ACF and PACF

There is a clear pattern present in ACF/PACF and model residuals plots repeating at lag 7. This suggests that our model may be better off with a complicate model with p or $q = 7$, hence $p + d + q \geq 6$. Thus, we tune our model again with $p + d + q \leq 12$.

Model	AIC	SSE	p-value
ARIMA(0,1,0)	9508.759	21376391	0
ARIMA(0,1,1)	9447.215	19578285	1.446001e-05
ARIMA(0,1,2)	9427.638	19002032	0.02841553
ARIMA(0,1,3)	9424.381	18864103	0.224242
ARIMA(0,1,4)	9412.845	18484407	2.467475e-08
ARIMA(0,1,5)	9380.69	17566041	2.210454e-13
ARIMA(0,1,6)	9231.937	14138719	4.218847e-15
ARIMA(0,1,7)	9046.189	10762568	0.0450277
ARIMA(1,1,0)	9428.742	19083921	0.008932232
ARIMA(1,1,1)	9423.817	18901820	0.287052
ARIMA(1,1,2)	9424.348	18863342	0.2757869
ARIMA(1,1,3)	9425.803	18848729	0.2156899
ARIMA(1,1,4)	9411.642	18398347	7.960588e-09
ARIMA(1,1,5)	9368.901	17226190	6.439294e-15
ARIMA(1,1,6)	9129.21	12104882	3.866307e-07
ARIMA(1,1,7)	**9024.564**	**10419601**	**0.447178**
ARIMA(2,1,0)	9422.642	18871035	0.2980812
ARIMA(2,1,1)	9424.643	18871052	0.2871248
ARIMA(2,1,2)	9426.325	18862759	0.2729236
ARIMA(2,1,3)	9428.29	18861834	0.2753095
ARIMA(2,1,4)	9309.766	15773420	0.01828572
ARIMA(2,1,5)	9255.432	14658607	3.377965e-12
ARIMA(2,1,6)	9095.232	11523857	7.651121e-05
ARIMA(2,1,7)	9025.909	10408619	0.498702
ARIMA(3,1,0)	9424.64	18870976	0.2894379
...	...	...	...
ARIMA(7,1,0)	9237.426	14371989	0.3030682
ARIMA(7,1,1)	9226.47	14114951	0.8543826
ARIMA(7,1,2)	9220.447	13957603	0.9780944

Table 4.9: Complicate ARIMA model with different lag p,d,q

As result, we choose ARIMA(1,1.7) which has the smallest AIC value.

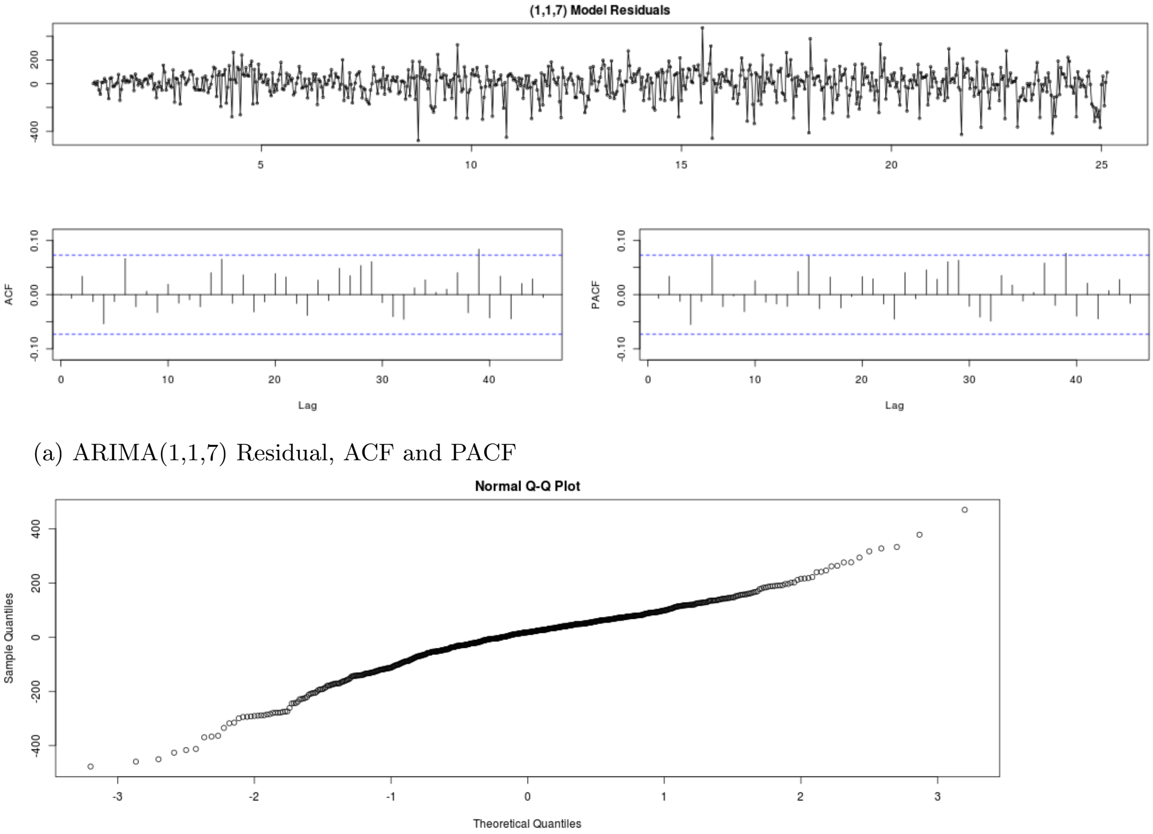
ar1	ma1	ma2	ma3	ma4	ma5	ma6	ma7
0.2803	0.1465	0.1524	0.1263	0.1225	0.1291	0.1471	0.8353

SE1	SE2	SE3	SE4	SE5	SE6	SE7	SE8
0.0478	0.0289	0.0266	0.0261	0.0263	0.0257	0.0265	0.0285

sigma$\hat{2}$	log likelihood	AIC
14392	-4503.28	9024.56

Table 4.10: ARIMA(1,1,7)

We check again ARIMA(1,1,7) model residuals, ACF and PACF. We also plot Q-Q.



(a) ARIMA(1,1,7) Residual, ACF and PACF

(b) ARIMA(1,1,7) Q-Q plot

Figure 4.22: ARIMA(1,1,7) Residual, ACF and PACF and Q-Q plot

This time, there are no significant auto correlations present. Then, we use this model to forecast next 25 steps with 95% confidence interval.

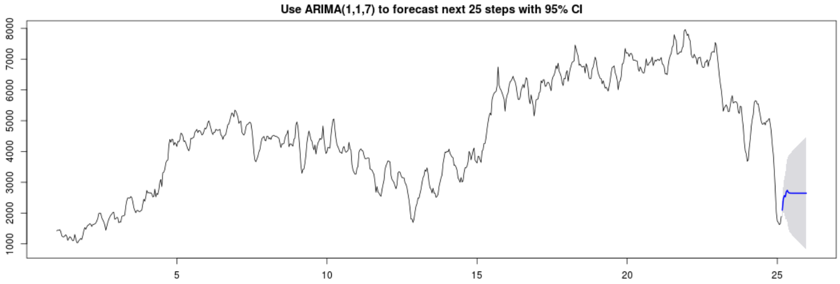


Figure 4.23: ARIMA(1,1,7) to forecast next 25 steps

To verify the prediction, we cut the last 25 data and use ARIMA(1,1,7) to forecast with 95% confidence interval.

We see that the blue line representing forecast seems very linear. It goes close to a straight line fairly soon, which seems unlikely given past behavior of the series since the model is assuming a series with no seasonality, and is differencing the original non-stationary data. Thus, in order to add back seasonal components to our forecasting, it would be better to use SARIMA for better results.

There are other methods to gain better result, such as choosing MSE instead of AIC, or we can differentize the model in higher order.

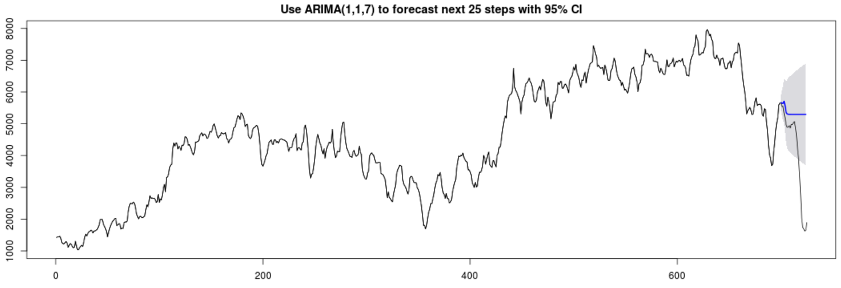


Figure 4.24: Evaluate ARIMA(1,1,7) to forecast next 31 steps

Finally, we conclude our model as following:

$$X_t = X_{t-1} + 0.2803\phi_1 + Z_t + 0.1465Z_{t-1} + 0.1524Z_{t-2} + 0.1263Z_{t-3} + 0.1225Z_{t-4} + 0.1291Z_{t-5} + 0.1471Z_{t-6} + 0.8353Z_{t-7}$$

where $Z_t \sim Normal(0, 14392)$

4.3 Compare and Evaluate

In this section, we will use the MSE, MAPE, MAE to compare the mean prediction of models. The results are shown in the tables 4.11 bellow:

Model	MSE	MAE	MAPE
Linear Regression	**4144798**	1573.858	**0.9669045**
Holt-Winters'	6393452	1846.064	1.288976
ARIMA(1,1,7)	4791037	**1552.695**	1.123162

Table 4.11: MSE, MAE and MAPE of models in test dataset

The MSE and MAPE of Linear Regression method are the lowest, and ARIMA(1,1,7) has the smallest MAE. Because the MAE does not give the weight to the outlier, the MSE and MAE of Linear Regression and ARIMA(1,1,7) are different. Thus, we can conclude that the linear regression model is better than ARIMA and Holt-Winters' method in test dataset.

Chapter 5

Conclusion

This project used time series analysis and forecasting method on the data of Bike Sharing [5] from 2011 to 2013. Three learned methods: Linear Regression, Holt-Winters' Methods and ARIMA model are used to analysis and forecast the series.

At first, because the time series is very fluctuating, so a cleaning the outlier of this times series is required. Then, the Linear Regression model are evaluated. The predictor are selected by heuristic and by checking correlation. Time instance and season variable are using to capture the trend and seasonal component. Others predictors capture the correlation between selected attribute and target value. Next, The Holt-Winter method are used because of additive seasonal component. Finally, the ARIMA model are analysed by Box-Jenkins' Method. By checking stationary, we found that the first differencing is the stationary. The ACF and PACF which were plotted show that the ARIMA(2,1,0) is required. However, after checking again the ACF, PACF of model residuals, we decide to fit the model again and gain the final result at ARIMA(1,1,7).

The comparision between 3 models shows that, the Linear Regression could predict more accurate, but slightly, than ARIMA(1,1,7). This could be because the Linear Regression have higher correlation predictor.

Bibliography

[1] Forecasting principle and practices OTexts. URL `https://www.otexts.org/`.

[2] R. Adhikari and R. K. Agrawal. An introductory study on time series modeling and forecasting. URL `https://arxiv.org/pdf/1302.6613.pdf`.

[3] C. Brooks. *Classical linear regression model assumptions and diagnostic tests*, page 179–250. Cambridge University Press, 3 edition, 2014. doi: 10.1017/CBO9781139540872.006.

[4] J. H. Cochrane. Time series for macroeconomics and finance. manuscript, 2005. URL `http://econ.lse.ac.uk/staff/wdenhaan/teach/cochrane.pdf`.

[5] H. Fanaee-T and J. Gama. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, pages 1–15, 2013. ISSN 2192-6352. doi: 10.1007/s13748-013-0040-3. URL `[WebLink]`.

[6] C. C. Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *International journal of forecasting*, 20(1):5–10, 2004.

[7] G. P. Nason. Stationary and non-stationary time series. *Statistics in Volcanology. Special Publications of IAVCEI*, 1:000–000, 2006.

[8] M. Pontier. Time series and forecasting lecture note, 2019.

[9] P. R. Winters. Forecasting sales by exponentially weighted moving averages. *Management science*, 6(3):324–342, 1960.

Appendix A

Data Attributes

Both hour.csv and day.csv have the following fields, except hr which is not available in day.csv

- instant: record index
- dteday : date
- season : season (1:springer, 2:summer, 3:fall, 4:winter)
- yr : year (0: 2011, 1:2012)
- mnth : month (1 to 12)
- hr : hour (0 to 23)
- holiday : weather day is holiday or not (extracted from [Web Link])
- weekday : day of the week
- workingday : if day is neither weekend nor holiday is 1, otherwise is 0.
- weathersit :
  - 1: Clear, Few clouds, Partly cloudy, Partly cloudy
  - 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
  - 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
  - 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
- temp : Normalized temperature in Celsius. The values are derived via $(t-t_{min})/(t_{max}-t_{min})$, $t_{min}=-8$, $t_{max}=+39$ (only in hourly scale)
- atemp: Normalized feeling temperature in Celsius. The values are derived via $(t-t_{min})/(t_{max}-t_{min})$, $t_{min}=-16$, $t_{max}=+50$ (only in hourly scale)
- hum: Normalized humidity. The values are divided to 100 (max)
- windspeed: Normalized wind speed. The values are divided to 67 (max)
- casual: count of casual users
- registered: count of registered users
- cnt: count of total rental bikes including both casual and registered

Appendix B

Source Code

```
# Import Package
library(ggplot2)      # Data visualization
library(readr)        # CSV file I/O, e.g. the read_csv function
library(forecast)
library(tseries)
library(astsa)
library(GGally)
library(fastDummies)
library(lmtest)
library(stargazer)
library(Metrics)
library(seasonal)

rm(list = ls(all = TRUE))
graphics.off()

# Get data Setworking Dir First!
daily_data <- read.csv(file="./Bike-Sharing-Dataset/day.csv",
                       header=TRUE, sep=",",
                       na.strings=c(""))

# View data
# View(daily_data)

# --> Data sets started from 2011-01-01 to 2012-12-31
daily_data$Date = as.Date(daily_data$dteday) # create toDate with column dteday

# Plot Time Series
ggplot(daily_data, aes(Date, cnt)) +
  geom_line(size=0.5) +
  scale_x_date('Month', breaks=('year')) +
  theme_bw() +
  ylab("Daily␣Bike␣Checkouts") + xlab("")


# Function to check outlier with box-plot
date_range <- daily_data$Date
measure <- daily_data$cnt

# create new columns for the months and years, and
# and a year_month column for x-axis labels
daily_data$month <- format(date_range, format="%b")
daily_data$year <- as.POSIXlt(date_range)$year + 1900
daily_data$year_month <- paste(daily_data$year, daily_data$month)
daily_data$sort_order <- daily_data$year *100 + as.POSIXlt(date_range)$mon
```

```
#plot it
ggplot(daily_data) +
      geom_boxplot(aes(x=reorder(year_month,
                                 sort_order),
                       y=measure)) +
      ylab("Daily␣Bike␣Checkouts") + xlab("") +
      theme_bw() +
          theme(text = element_text(size=10),
                  axis.text.x = element_text(angle=90, hjust=1))

# Delete Outlier
count_ts = ts(daily_data[, c('cnt')],frequency = 30)
daily_data$cnt_ma = ma(tsclean(count_ts), order=7)
daily_data$clean_cnt = as.numeric(tsclean(count_ts))


#------------------------ Linear Regression --------------------#
# Correlation Exploring
keep <- c('instant','Date','season','holiday','workingday','weathersit'
          ,'temp','atemp','hum','windspeed','cnt','clean_cnt')
temp <- daily_data[keep]
ggpairs(temp,columns = c('temp','atemp','hum','windspeed','clean_cnt'))

# ANOVA testing
temp$season <- as.factor(temp$season)
aov1 = aov(atemp ~ season ,data = temp )
summary(aov1)
aov2 = aov(hum ~ season ,data = temp )
summary(aov2)
aov3 = aov(windspeed ~ season ,data = temp )
summary(aov3)

temp$weathersit <- as.factor(temp$weathersit)
aov4 = aov(atemp ~ weathersit ,data = temp )
summary(aov4)
aov5 = aov(hum ~ weathersit ,data = temp )
summary(aov5)
aov6 = aov(windspeed ~ weathersit ,data = temp )
summary(aov6)


# Create Dummies Variable
temp <- dummy_cols(temp,select_columns = c('season'))
temp <- dummy_cols(temp,select_columns = c('weathersit'))
temp$weathersit_3 <- NULL
temp$season_4 <- NULL

# Slipt Train, Test set.
train_count <- ts(count_ts[-(701:731)],frequency = 30)
test_count <- count_ts[(701:731)]

train <- temp[-(701:731),]
test   <- temp[(701:731),]

# Linear Model with dummies Variables
lm.cnt <- lm(
  clean_cnt ~ instant + season_1 + season_2 + season_3 +
        weathersit_1 + weathersit_2 + holiday + workingday +
        atemp + hum + windspeed,
  data=train)
summary(lm.cnt)
```

```
# Test
bptest(lm.cnt)  # Variance test
checkresiduals(lm.cnt) # Residual test
# Checking Nornal Residual
qqnorm(resid(lm.cnt), main = "Normal␣Q-Q␣Plot,Residual", col = "darkgrey")
qqline(resid(lm.cnt), col = "dodgerblue", lwd = 2)


# Violate the Error Assumption
# Fitting data test set
cnt_predict <- predict(lm.cnt,test)
# Plot two timeseries
ggplot() +
  geom_line(data = test,aes(x=Date,y=cnt_predict,
                            colour = "Predict")) +
  geom_line(data = train,aes(x=Date, y = clean_cnt,
                             colour = "Data")) +
  geom_line(data = test,aes(x=Date,y=clean_cnt  ,
                            colour = "Data")) +
  scale_color_grey(start = 0.7, end = 0.1) +
  theme_bw() +
  xlab("Month") + ylab("Daily␣Count") +
  ggtitle("Data␣and␣Prediction")


ggplot() +
  geom_line(data = train,aes(x=Date, y = clean_cnt,
                             colour = "Data")) +
  geom_line(data = train,aes(x=Date, y = fitted(lm.cnt),
                              colour = "Fitted")) +
  geom_line(data = test,aes(x=Date,y=clean_cnt  ,
                            colour = "Data")) +
  scale_color_grey(start = 0.7, end = 0.1) +
  xlab("Month") + ylab("Daily␣Count") +
  theme_bw() +
  ggtitle("Data␣vs.␣Fitted␣")



#------------------------- Holtwinter Forecast ---------------------#
daily_data$cnt_ma
count_ma = ts(na.omit(daily_data$cnt_ma), frequency=30)
train_count_ma <-  ts(frequency = 30,count_ma[-(700:725)])
hwfit <-  HoltWinters(train_count_ma)
hwforcast <- forecast(hwfit,31)




data_ma <- count_ma
autoplot(data_ma) +
  autolayer(hwforcast,colour = 'grey') +
  autolayer(data_ma) +
  theme_bw() +
  scale_color_grey(start = 0.2, end = 0) +
  scale_x_continuous(breaks= c(1,13,24.25),
                     labels= c("2011-Jan","2012-Jan","2012-Dec")) +
  xlab("Month") + ylab("Daily␣Count")

data <- count_ts
autoplot(data) +
  autolayer(hwforcast,colour = 'grey') +
  autolayer(data) +
  theme_bw() +
  scale_color_grey(start = 0.2, end = 0) +
  scale_x_continuous(breaks= c(1,13,24.25),
                     labels= c("2011-Jan","2012-Jan","2012-Dec")) +
  xlab("Month") + ylab("Daily␣Count")

# Testing Residual
Box.test(hwforcast$residual, lag=20, type="Ljung-Box")


#-------------------ARIMA model-----------------------------------#

# Calculate seasonal component of the data using stl()

# count_ma = ts(na.omit(daily_data$cnt_ma), frequency=30)
# train_count_ma_log <- log(train_count_ma)
# plot(train_count_ma_log)

decomp = stl(train_count_ma, s.window="periodic") # STL calculates the seasonal compo
deseasonal_cnt <- seasadj(decomp)
plot(decomp)
acf2(deseasonal_cnt)


# Check stationarity
adf.test(train_count_ma, alternative = "stationary")
# Autocorrelations and Choosing Model Order

acf(train_count_ma, main='ACF␣plot␣with␣seasonal␣component') # ACF plots display corr
pacf(train_count_ma, main='PACF␣plot␣␣with␣seasonal␣component') # PACF display correl
acf2(train_count_ma)
# Plotting the differenced series,
#  we see an oscillating pattern around 0 with no visible strong trend

count_d1 = diff(train_count_ma, differences = 1)
plot(count_d1, main="Bicycle␣usage␣after␣decomposion,␣difference␣1␣and␣moving␣average
adf.test(count_d1, alternative = "stationary")



# Spikes at particular lags of the differenced series
# can help inform the choice of p or q
Acf(count_d1, main='ACF␣for␣Differenced␣Series')
# Following lags are significant for MA(q)
# Lag 1, Lag 2, Lag 3, Lag 7, Lag 8
# The order q of MA terms can be 1,2,3,7
Pacf(count_d1, main='PACF␣for␣Differenced␣Series')
acf2(count_d1)
# Following lags are significant for AR(p)
# Lag 1, Lag 2, Lag 6, Lag 7, Lag 8, Lag 13, Lag 14
# The order p of AR terms can be 1,2,3,7



# Manual calculation with arima d=1 # difference
d = 1
for(p in 1:4){
  for(q in 1:4){ if(p+d+q<=6){
    model<-arima(x=deseasonal_cnt, order = c((p-1),d,(q-1)))
    pval<-Box.test(model$residuals, lag=log(length(model$residuals)))
    sse<-sum(model$residuals^2)
    cat(p-1,d,q-1, 'AIC=', model$aic, '␣SSE=',sse,'␣p-VALUE=', pval$p.value,'\n')
  } }
}

# We choose model with smallest AIC
# (2 1 0) AIC= 9422.642 SSE= 18871035 p-VALUE= 0.2980812
arima(x=deseasonal_cnt, order = c(2,1,0))
fit<-auto.arima(deseasonal_cnt, seasonal=FALSE)
tsdisplay(residuals(fit), lag.max=45, main='(2,1,0)␣Model␣Residuals')

# There is a clear pattern present in ACF/PACF
# Model residuals plots repeating at lag 7.
# our model may be better off with a different specification,
# such as p = 7 or q = 7

# Make p+d+q<=12 means that the model will be very complex
for(p in 1:8){
  for(q in 1:8){
    if(p+d+q<=12){
      model<-arima(x=deseasonal_cnt, order = c((p-1),d,(q-1)))
      pval<-Box.test(model$residuals, lag=log(length(model$residuals)))
      sse<-sum(model$residuals^2)
      cat(p-1,d,q-1, 'AIC=', model$aic, '␣SSE=',sse,'␣p-VALUE=', pval$p.value,'\n')
    }
  }
}

# ARIMA(1 1 7) model has the smallest AIC value
# (1 1 7) AIC= 8688.101  SSE= 9857393  p-VALUE= 0.2897584
model.arima <- arima(x=deseasonal_cnt, order = c(1,1,7))
arima.forcast <- forecast(model.arima, h=31,level=95)

data <- count_ma
autoplot(data) +
  autolayer(arima.forcast,colour = 'grey') +
  autolayer(data) +
  theme_bw() +
  scale_color_grey(start = 0.2, end = 0) +
  scale_x_continuous(breaks= c(1,13,24.25),
                     labels= c("2011-Jan","2012-Jan","2012-Dec")) +
  xlab("Month") + ylab("Daily␣Count")

hold <- window(ts(deseasonal_cnt), start=700)

fit_no_holdout = arima(ts(deseasonal_cnt[-c(700:725)]), order=c(1,1,7))

fcast_no_holdout <- forecast(fit_no_holdout,h=25,level=95)
plot(fcast_no_holdout, main="Use␣ARIMA(1,1,7)␣to␣forecast␣next␣25␣steps")
lines(ts(deseasonal_cnt))
#-------------------Compare Model on Testing Set-----------------#

# Holt
mse(test_count,hwforcast$mean)
mae(test_count,hwforcast$mean)
mape(test_count,hwforcast$mean)
# Linear
mse(test$clean_cnt,cnt_predict)
mae(test$clean_cnt,cnt_predict)
mape(test$clean_cnt,cnt_predict)
#ARIMA
mse(test_count,arima.forcast$mean)
mae(test_count,arima.forcast$mean)
mape(test_count,arima.forcast$mean)
```