Final Project: Bike Sharing Dataset

Northeastern: ALY 6015

G. Holt Williams

6/28/2020

# 

# Introduction

The purpose of this assignment is to showcase the material the group learned throughout ALY 6015. This is accomplished first through exploratory data analysis of a large data set with accompanying tables and data visualizations for emphasis. Second, will be a showcase of statistical hypothesis testing to answer questions that would be relevant to a business. The final section delves into using linear modeling to understand the relationship that different time periods have on the data, followed by regularization methods to determine a more predictive model with all available variables. Each section will compare Casual vs Registered vs Total users.

# Data Import and Cleaning

The following R code imports the dataset and creates factor/date variables for the time variables which will be essential for further analysis. The code also renames all levels for the factor variables for ease of readability. Other useful cleaning steps include making the data “tall” instead of “wide” by combining the Total, Registered, and Casual columns into two columns of type and count.

## load Libraries  
library(tidyverse)  
library(psych)  
library(tidymodels)  
library(forecast)  
## load data  
hour<-read.csv("hour.csv")  
## Clean and create factor variables  
cols<-colnames(hour)[3:10]  
hour<-hour%>%mutate\_at(cols, funs(factor(.)))  
hour$dteday<-as.Date(hour$dteday)  
long<-hour%>%gather("Type","Count",15:17)  
long$Type<-as.factor(long$Type)

# Exploratory Data Analysis

This section explores the structure of the data. First is a table with basic summary statistics of the non-time/non-factorized variables. This includes weather variables and the outcome variables Casual, for Casual users, Registered, for Registered users, and Total, for total count of users.

hour[2:17]%>%  
 describe(quant=c(.25,.75),omit=TRUE)%>%  
 select(n, sd,mean, median, min, max, Q0.25, Q0.75)%>%round(digits=2)

Table

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | n | sd | mean | median | min | max | Q0.25 | Q0.75 |
| Temperature | 17379 | 0.19 | 0.50 | 0.50 | 0.02 | 1.00 | 0.34 | 0.66 |
| atemp | 17379 | 0.17 | 0.48 | 0.48 | 0.00 | 1.00 | 0.33 | 0.62 |
| Humidity | 17379 | 0.19 | 0.63 | 0.63 | 0.00 | 1.00 | 0.48 | 0.78 |
| Windspeed | 17379 | 0.12 | 0.19 | 0.19 | 0.00 | 0.85 | 0.10 | 0.25 |
| Casual | 17379 | 49.31 | 35.68 | 17.00 | 0.00 | 367.00 | 4.00 | 48.00 |
| Registered | 17379 | 151.36 | 153.79 | 115.00 | 0.00 | 886.00 | 34.00 | 220.00 |
| Total | 17379 | 181.39 | 189.46 | 142.00 | 1.00 | 977.00 | 40.00 | 281.00 |

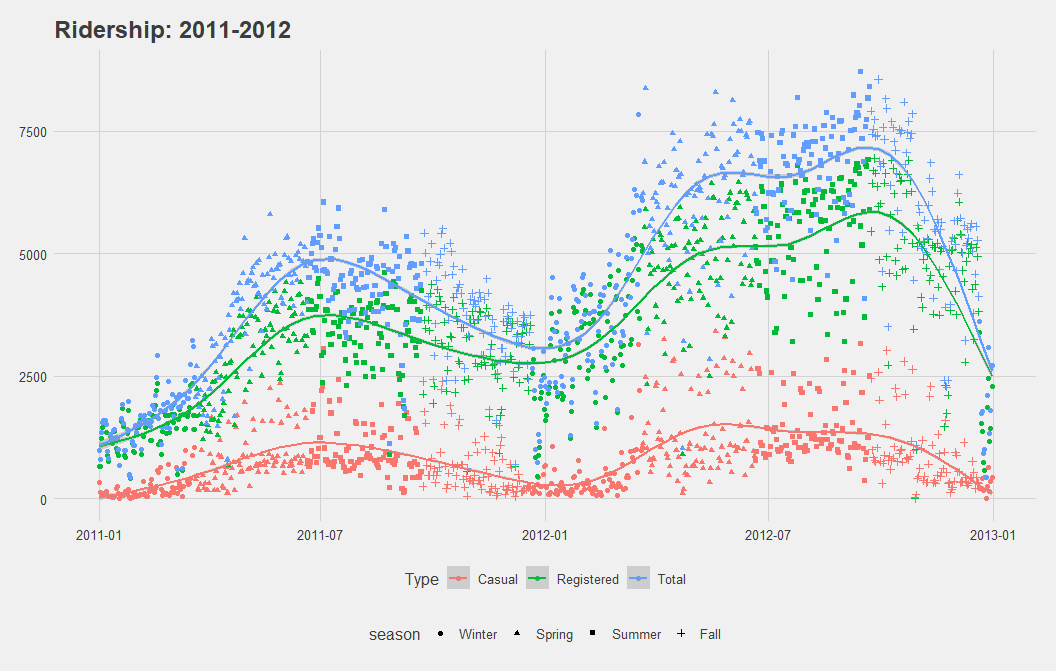
According to the authors, the weather variables (Temperature, atemp, Humidity, and Windspeed) have been normalized (Dua 2019). As is evident from the above table, the dataset has 17379 entries of data for every hour over a two-year period.

This period being from 2011-01-01 through 2012-12-31. Next the analysis digs into the different time periods that are available in this dataset: Yearly, Seasonal, Monthly, and Daily

## Yearly

long%>%  
 group\_by(Type,dteday)%>%  
 mutate(sum=sum(Count))%>%  
 ggplot(aes(x=dteday,y=sum, color=Type))+  
 geom\_point(aes(shape=season))+stat\_smooth()+   
 theme\_fivethirtyeight()+  
 xlab("Time")+  
 ylab("Ridership")+ggtitle("Ridership: 2011-2012")

Figure



hour%>%  
 group\_by(Year)%>%   
 summarise(Casual=sum(Casual),Registered=sum(Registered),Total=sum(Total))

Table

|  |  |  |  |
| --- | --- | --- | --- |
| Year | Casual | Registered | Total |
| 2011 | 247252 | 995851 | 1243103 |
| 2012 | 372765 | 1676811 | 2049576 |

In the above table and chart, overall ridership increases over the two-year time period. The total ridership increases by almost 800,000 trips, that is mostly contained in its Registered users which rose just over 680,000 trips from 2011 to 2012. This increase is likely due to an increase in the quantity of bikes plus the normalization of local populations to using the bicycles as a reliable mode of transportation.

## Seasonal and Monthly

hour%>%  
 group\_by(Year,season)%>%   
 summarise(Casual=sum(Casual),Registered=sum(Registered),Total=sum(Total))%>%  
 knitr::kable()

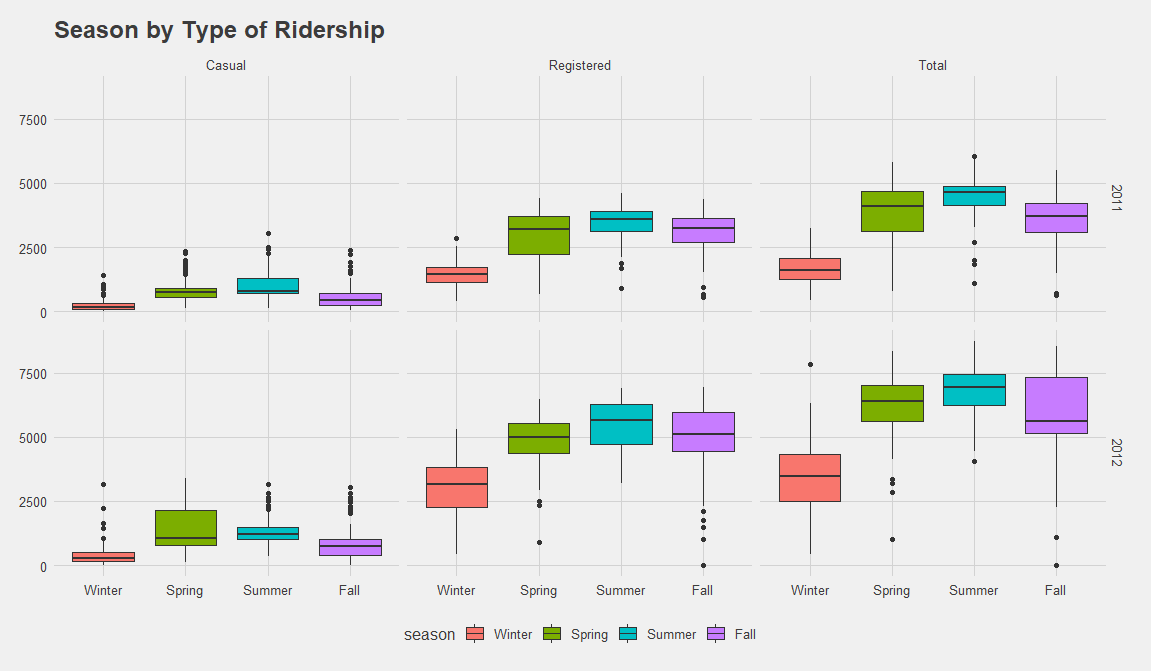
Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Year | season | Casual | Registered | Total |
| 2011 | Winter | 21425 | 128575 | 150000 |
| 2011 | Spring | 77564 | 269752 | 347316 |
| 2011 | Summer | 95450 | 324200 | 419650 |
| 2011 | Fall | 52813 | 273324 | 326137 |
| 2012 | Winter | 39197 | 282151 | 321348 |
| 2012 | Spring | 125958 | 445315 | 571273 |
| 2012 | Summer | 130641 | 510838 | 641479 |
| 2012 | Fall | 76969 | 438507 | 515476 |

From this table it is easy to see that the winter season features the smallest quantity of riders, which follows logical sense as snow and cold weather can easily dissuade users from taking a bicycle. Similarly, Summer has the highest ridership’s as the weather is more likely to be amenable to biking.

long%>%  
 group\_by(Type,dteday)%>%  
 mutate(sum=sum(Count))%>%  
 ggplot(aes(x=sum, y=season, fill=season))+  
 geom\_boxplot()+  
 facet\_grid(Year~Type)+  
 coord\_flip()+  
 theme\_fivethirtyeight()+theme()+  
 xlab("Ridership")+ggtitle("Season by Type of Ridership")

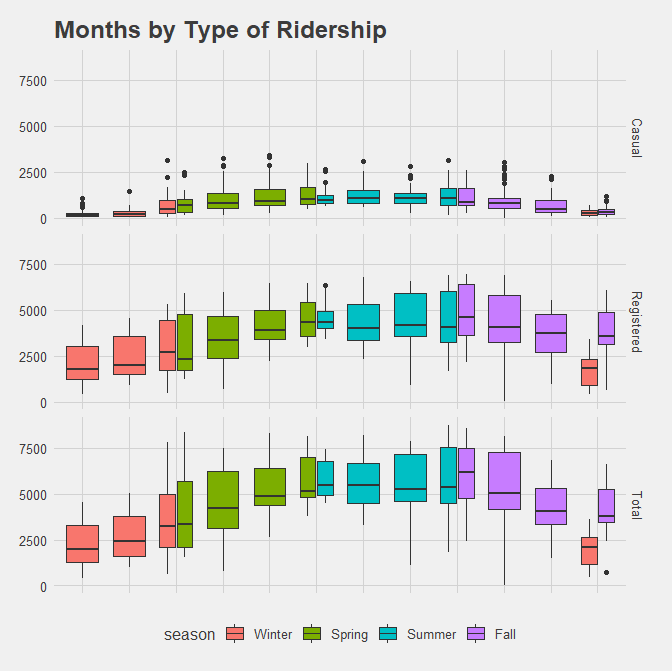
Figure



In the boxplots we can see this relationship playout with all three groups of data. One interesting takeaway is that mean Casual fall ridership is below mean Casual spring ridership, but this is not the case for Registered ridership. This is most likely due to fewer tourists, who likely have a large impact on Casual ridership, in the fall, as families have kids going back to school. The flip side of this looking at Registered users suggests that many users depend on the bicycles for their daily commutes. This hypothesis becomes more apparent in the daily and hourly sections.

long%>%  
 group\_by(Type,dteday)%>%  
 mutate(sum=sum(Count))%>%  
 ggplot(aes(x=sum, y=Month, fill=season))+  
 geom\_boxplot()+  
 facet\_grid(Type~.)+coord\_flip()+  
 theme\_fivethirtyeight()+theme(axis.text.x=element\_blank(),  
 axis.ticks.x=element\_blank())+  
 xlab("Ridership")+ylab("Months")+ggtitle("Months by Type of Ridership")

Figure

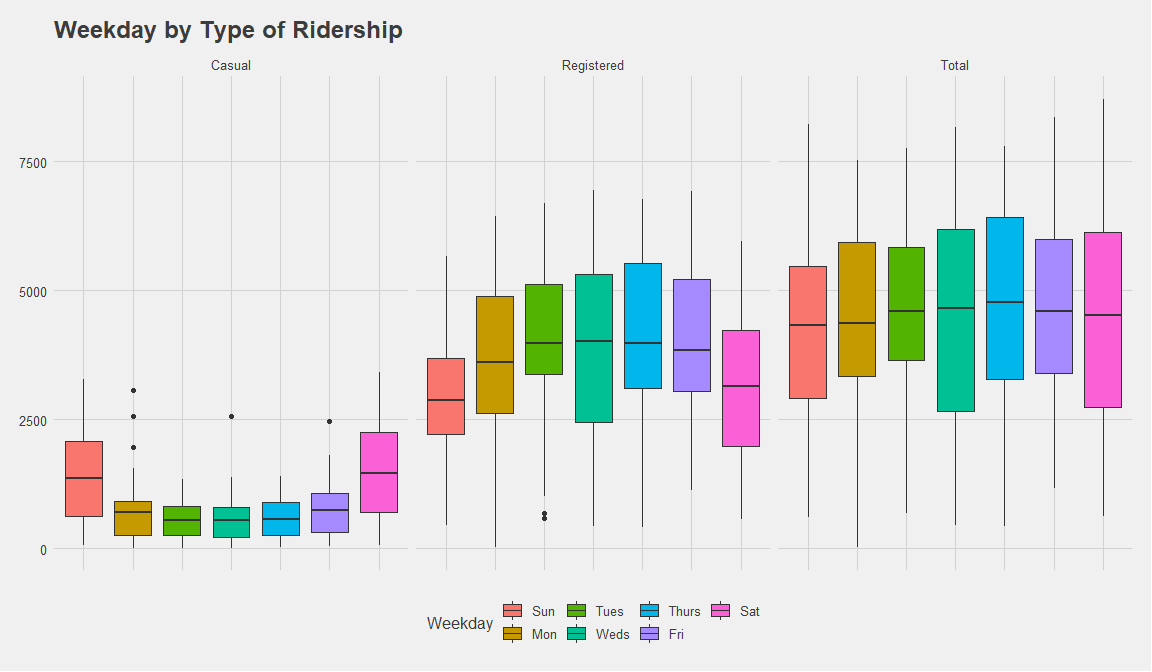


Here we see the same patterns discussed above, with the data broken down by each month and season. The extra boxplots are the months where the seasons shift mid-month. One interesting takeaway from this visualization is how in December casual ridership has a very small change from spring to winter, but registered users have a large leap, likely due to Christmas and having days off where they are not commuting but may instead be using a car to travel longer distances.

## Weekly and Daily

long%>%  
 group\_by(Type,dteday)%>%  
 mutate(sum=sum(Count))%>%  
 ggplot(aes(x=sum, y=Weekday, fill=Weekday))+  
 geom\_boxplot()+  
 facet\_wrap(~Type)+coord\_flip()+  
 theme\_fivethirtyeight()+theme(axis.text.x=element\_blank(),  
 axis.ticks.x=element\_blank())+  
 xlab("Ridership")+ggtitle("Weekday by Type of Ridership")

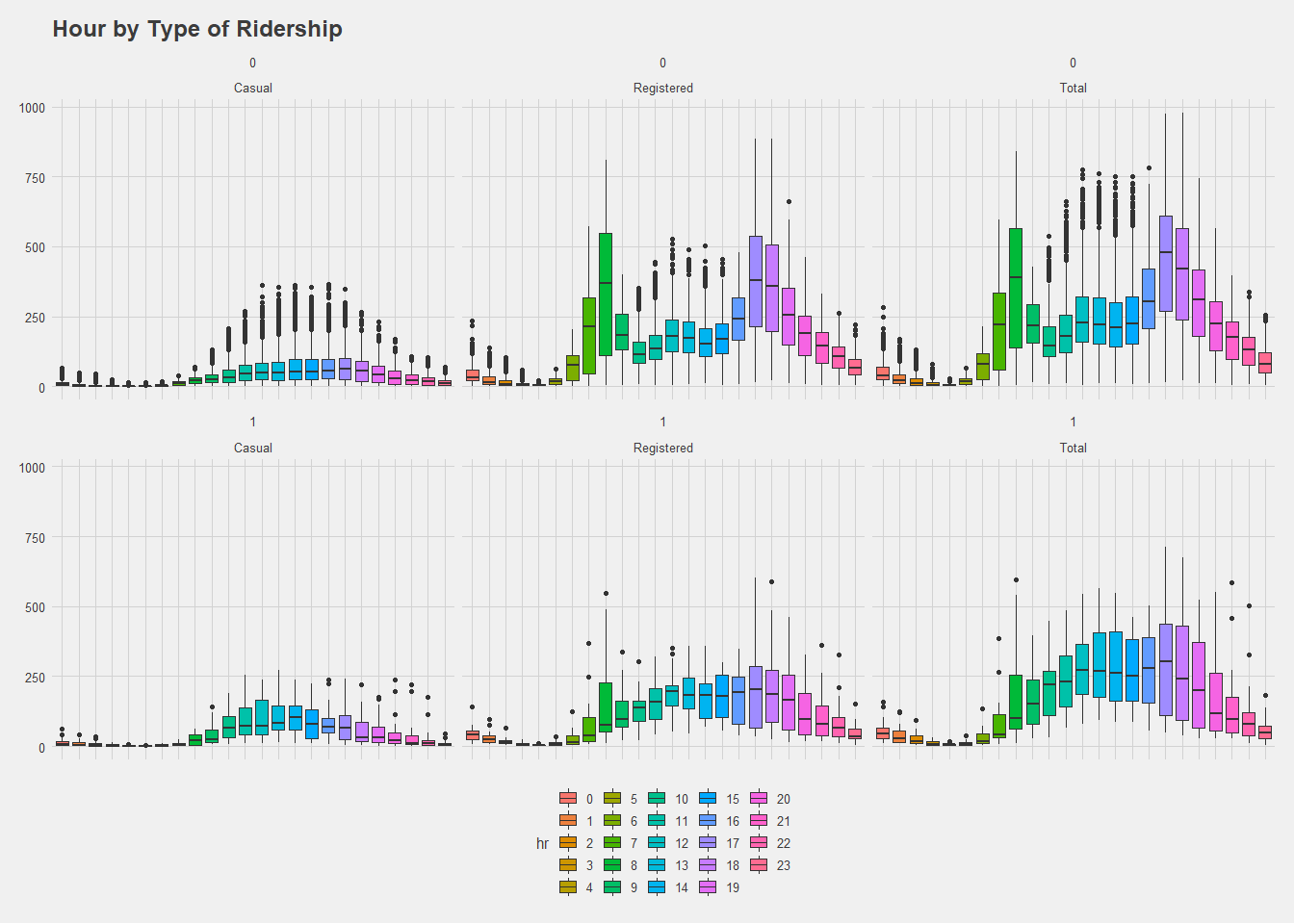
Figure



Looking at weekly data, the data begins to show the stark differences in the types of users of the bicycle sharing program. The Registered users have about average ridership from Monday through Friday while the Casual ridership has the opposite pattern. This disparity likely comes from Registered users utilizing the bicycles as their main daily commute to and from work. This theory is confirmed when looking at the different times of day that ridership is high for the two groups. The casual ridership spikes on the weekends as that is when casual users have free time during sunlight hours

long%>%  
 group\_by(Type)%>%  
 ggplot(aes(x=Count, y=hr, fill=hr))+  
 geom\_boxplot()+  
 facet\_wrap(Holiday~Type)+  
coord\_flip()+  
 theme\_fivethirtyeight()+theme(axis.text.x=element\_blank(),  
 axis.ticks.x=element\_blank())+  
 xlab("Ridership")+ylab("Hour")+ggtitle("Hour by Type of Ridership")

Figure



Here is visualization with a few dimensions. From left to right is the day broken down into 24 hours by each type of riders, which is then facetted by a day being a Holiday from top to bottom. Looking at the top, the daily commute of registered riders is highest during normal commute hours, whereas casual users don’t tend to start using the cycles until 10 at a sustained level till night. While looking at the holiday section the fact that registered users use the bicycles is reconfirmed as the spike during commute hours is no longer present. That time is still higher than casual users, but this is likely due to the few registered users who still must work on Holidays. Interestingly, casual users have a higher ridership in the middle of the day on Holidays as they likely don’t have to work on those days.

# Hypothesis Testing

Next, is a few hypothesis tests to explore the output variables. First will be a t-test with the following hypothesis and outputs

Ho: The mean ridership of Registered and Casual users is equal

Ha: The mean ridership of Registered and Casual users is not equal

alpha = 0.05

t.test(value~key , data=gather(hour[15:16]))

##   
## Welch Two Sample t-test  
##   
## data: value by key  
## t = -97.813, df = 21025, p-value < 2.2e-16  
## alternative hypothesis: true difference in means is not equal to 0  
## 95 percent confidence interval:  
## -120.4775 -115.7438  
## sample estimates:  
## mean in group Casual mean in group Registered   
## 35.67622 153.78687

t: -97.813

P-value: < 2.2e-16

Using a two-sample t-test where the null is that there is no difference in mean ridership of Registered and Casual users, the outcome is a p-value less than 0.05, which means the null hypothesis is rejected at the 95% confidence interval.

This was an expected outcome as Registered users receive a better deal on bikes and are more likely to live in the area and use the bikes as a daily or weekly commute vehicle, versus Casual users are more likely to use the bikes for one off adventures. Because this was an expected outcome, it would be prudent to perform an f-test to the variance of the ridership’s between Casual and Registered.

Ho: There is no difference between the variances of ridership of Registered and Casual users

Ha: There is a difference between the variances of ridership of Registered and Casual users

alpha = 0.05

var.test(value~key , data=gather(hour[15:16]))

##   
## F test to compare two variances  
##   
## data: value by key  
## F = 0.10611, num df = 17378, denom df = 17378, p-value < 2.2e-16  
## alternative hypothesis: true ratio of variances is not equal to 1  
## 95 percent confidence interval:  
## 0.1030057 0.1093176  
## sample estimates:  
## ratio of variances   
## 0.1061148

F: 0.10611

P-value: < 2.2e-16

Utilizing an f-test to test if there is a difference in the variances of the ridership has a p-value less than 0.05. This means the null hypothesis is rejected at a 95% confidence interval, the null hypothesis being that the difference is zero, while the alternative hypothesis is that the difference is not equal to zero. This is an unsurprising result after exploring the visualizations of the two types of ridership. Registered users more often utilize the bikes for daily commuting, while Casual users are more likely to use them for more one-off instances and will not have the same reliance on the bicycles resulting in different variances.

# Modelling

## Model Creation and Analysis: Time Variables

The next section delves into the relationship that different time periods have on the different types of ridership. This will be explored by comparing a model with all-time variables versus multiple models using just one time variable.

Utilizing the tidyverse set of functions, specifically the broom and purr packages a list data frame grouped by each type of ridership with models for Year, Season, Month, Weekday, and Hour is created.

# Packages  
  
# Nest Data  
test<- long%>%group\_by(Type)%>%nest()  
# Example Model (only one shown for brevity)  
ALL\_model <- function(df) {  
 lm(Count ~ Year+season+Month+Weekday+hr, data = df)  
} …  
# Apply models  
test <- test%>%   
 mutate(ALL\_model = map(data, ALL\_model))…  
test[c(1,3:8)]

## # A tibble: 3 x 7  
## # Groups: Type [3]  
## Type ALL\_model yr\_model season\_model mnth\_model Weekday\_model hr\_model  
## <fct> <list> <list> <list> <list> <list> <list>   
## 1 Casual <lm> <lm> <lm> <lm> <lm> <lm>   
## 2 Regist~ <lm> <lm> <lm> <lm> <lm> <lm>   
## 3 Total <lm> <lm> <lm> <lm> <lm> <lm>

The above table features each model created and the comparisons are delved into below.

## Model Outputs

Linear modelling was utilized to compare different models possible using the multiple time variables in the dataset. The table contains the R^2, AIC, BIC, and Adjusted R^2 for all the models run. First, a model with all the time variables was run, then models with just singular time variables was run as shown in the table.

model\_table<-test%>%mutate(  
 All = map(ALL\_model, broom::glance),…  
)  
output\_table<-model\_table[9:14]%>%unnest(c(All,Year,Season,Month,Weekday,Hour),names\_sep=".", .drop = TRUE)%>%  
 select(matches("r.squared"),matches("AIC"),matches("BIC"))  
tidy(t(output\_table))%>%arrange(.rownames)%>%rename(Model.by.Statistic=.rownames, Causal=X1, Registered=X2, Total=X3)

Table

|  |  |  |  |
| --- | --- | --- | --- |
| Model.by.Statistic | Causal | Registered | Total |
| All.adj.r.squared | 5.388591e-01 | 6.551296e-01 | 6.512228e-01 |
| All.AIC | 1.714018e+05 | 2.053377e+05 | 2.118244e+05 |
| All.BIC | 1.717589e+05 | 2.056948e+05 | 2.121815e+05 |
| All.r.squared | 5.400267e-01 | 6.560028e-01 | 6.521059e-01 |
| Hour.adj.r.squared | 3.131703e-01 | 5.118838e-01 | 5.008328e-01 |
| Hour.AIC | 1.783043e+05 | 2.113540e+05 | 2.180340e+05 |
| Hour.BIC | 1.784984e+05 | 2.115480e+05 | 2.182281e+05 |
| Hour.r.squared | 3.140793e-01 | 5.125299e-01 | 5.014934e-01 |
| Month.adj.r.squared | 1.045385e-01 | 5.171070e-02 | 7.446290e-02 |
| Month.AIC | 1.829021e+05 | 2.228835e+05 | 2.287524e+05 |
| Month.BIC | 1.830031e+05 | 2.229844e+05 | 2.288533e+05 |
| Month.r.squared | 1.051053e-01 | 5.231090e-02 | 7.504870e-02 |
| Season.adj.r.squared | 8.246890e-02 | 4.892240e-02 | 6.582670e-02 |
| Season.AIC | 1.833173e+05 | 2.229265e+05 | 2.289058e+05 |
| Season.BIC | 1.833561e+05 | 2.229653e+05 | 2.289446e+05 |
| Season.r.squared | 8.262730e-02 | 4.908660e-02 | 6.598790e-02 |
| Weekday.adj.r.squared | 9.194980e-02 | 1.533000e-02 | 8.582000e-04 |
| Weekday.AIC | 1.831398e+05 | 2.235327e+05 | 2.300773e+05 |
| Weekday.BIC | 1.832019e+05 | 2.235948e+05 | 2.301394e+05 |
| Weekday.r.squared | 9.226330e-02 | 1.567000e-02 | 1.203200e-03 |
| Year.adj.r.squared | 2.032930e-02 | 6.430190e-02 | 6.269380e-02 |
| Year.AIC | 1.844541e+05 | 2.226412e+05 | 2.289620e+05 |
| Year.BIC | 1.844774e+05 | 2.226645e+05 | 2.289853e+05 |
| Year.r.squared | 2.038570e-02 | 6.435570e-02 | 6.274770e-02 |

From the table one can see that overall just using time in different formats does not create a strong model. Interpreting the All model from here, one can say the model predicted 65% of the variation (R^2) when applied to the Total ridership. Next, the terms, estimates, and p-values of the Season model are shown and interpreted.

test %>%   
 mutate(tidy = map(season\_model, broom::tidy)) %>%   
 unnest(tidy, .drop = TRUE)%>%select(term,estimate, p.value)

## # A tibble: 12 x 4  
## # Groups: Type [3]  
## Type term estimate p.value  
## <fct> <chr> <dbl> <dbl>  
## 1 Casual (Intercept) 14.3 1.58e- 85  
## 2 Casual seasonSpring 31.9 3.03e-210  
## 3 Casual seasonSummer 36.0 4.17e-268  
## 4 Casual seasonFall 16.4 6.22e- 57  
## 5 Registered (Intercept) 96.8 0.   
## 6 Registered seasonSpring 65.4 4.50e- 93  
## 7 Registered seasonSummer 88.9 2.12e-170  
## 8 Registered seasonFall 71.4 3.12e-108  
## 9 Total (Intercept) 111. 0.   
## 10 Total seasonSpring 97.2 6.27e-144  
## 11 Total seasonSummer 125. 1.53e-235  
## 12 Total seasonFall 87.8 1.03e-115

The above estimates show the same patterns seen in the previous visualizations. The casual model has summer and spring both estimating twice as much of an impact to fall when compared against the winter, while the registered model has fall having a greater impact on ridership than spring when compared to the winter season. All the terms are statistically significant at the 95% confidence interval when looking over the p-value.

## Forecasting: Arima and TBATS

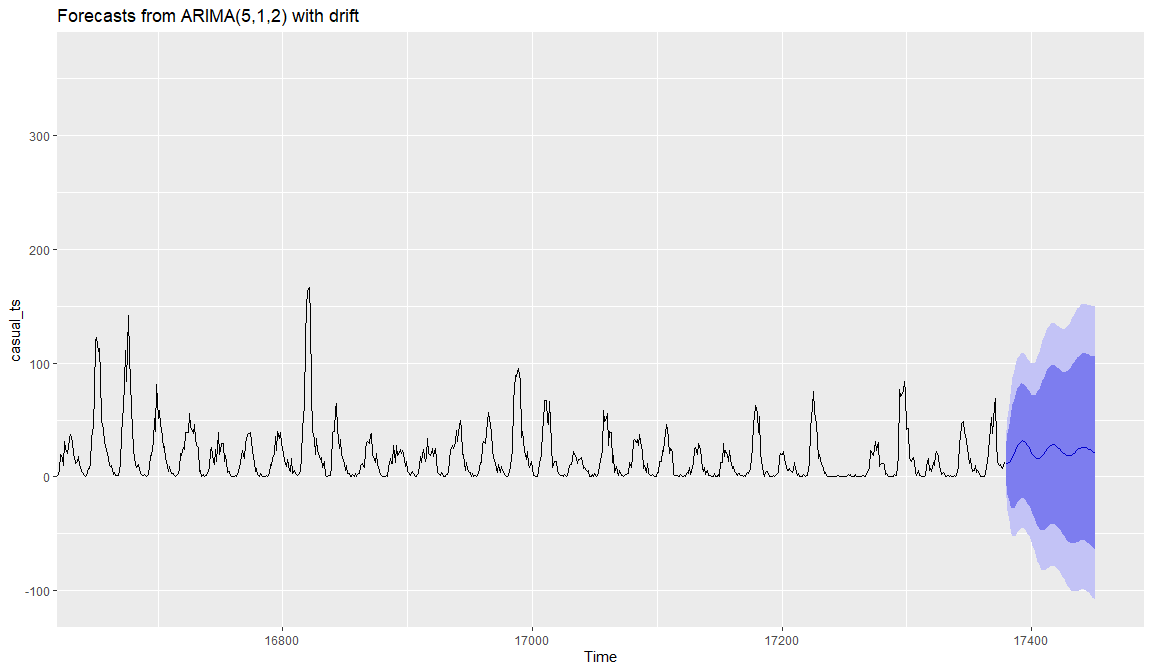
The previous sections of this analysis focused on identifying the trends through visualization and linear modelling, but the upcoming section will explore time series analysis with auto.arima and TBATS models. The following code chunks creates the time series required for analysis

# Packages  
Casual<-long[c(15:16)]%>%filter(Type=="Casual")  
Casual<-Casual[2]  
Registered<-long[c(15:16)]%>%filter(Type=="Registered")  
Registered<-Registered[2]  
Total<-long[c(15:16)]%>%filter(Type=="Total")  
Total<-Total[2]  
casual\_ts<-ts(Casual)  
Registered\_ts<-ts(Registered)  
Total\_ts<-ts(Total)

This section utilizes the autoArima() function in R to forecast the next 72 hours:

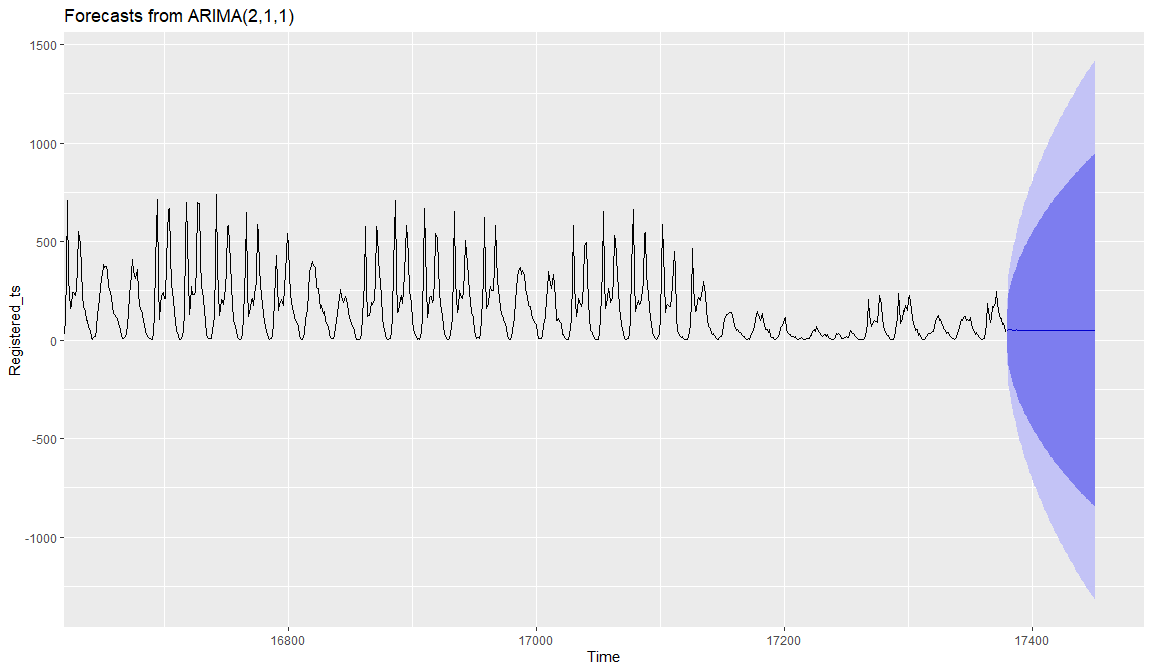
Casual\_arima<-auto.arima(casual\_ts)  
autoplot(forecast(Casual\_arima, h=72))+  
 coord\_cartesian(xlim = c(nrow(Casual)-720, nrow(Casual)+72))

Figure



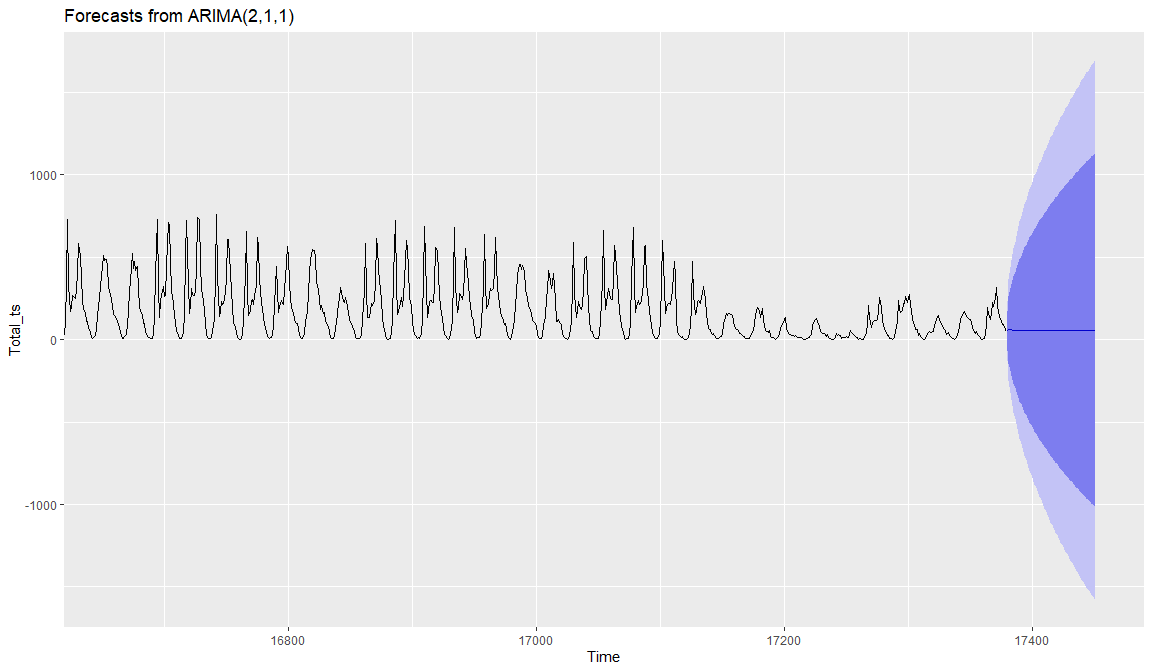
Registered\_arima<-auto.arima(Registered\_ts)  
autoplot(forecast(Registered\_arima, h=72))+  
 coord\_cartesian(xlim = c(nrow(Casual)-720, nrow(Casual)+72))

Figure



Total\_arima<-auto.arima(Total\_ts)  
autoplot(forecast(Total\_arima, h=72))+  
 coord\_cartesian(xlim = c(nrow(Casual)-720, nrow(Casual)+72))

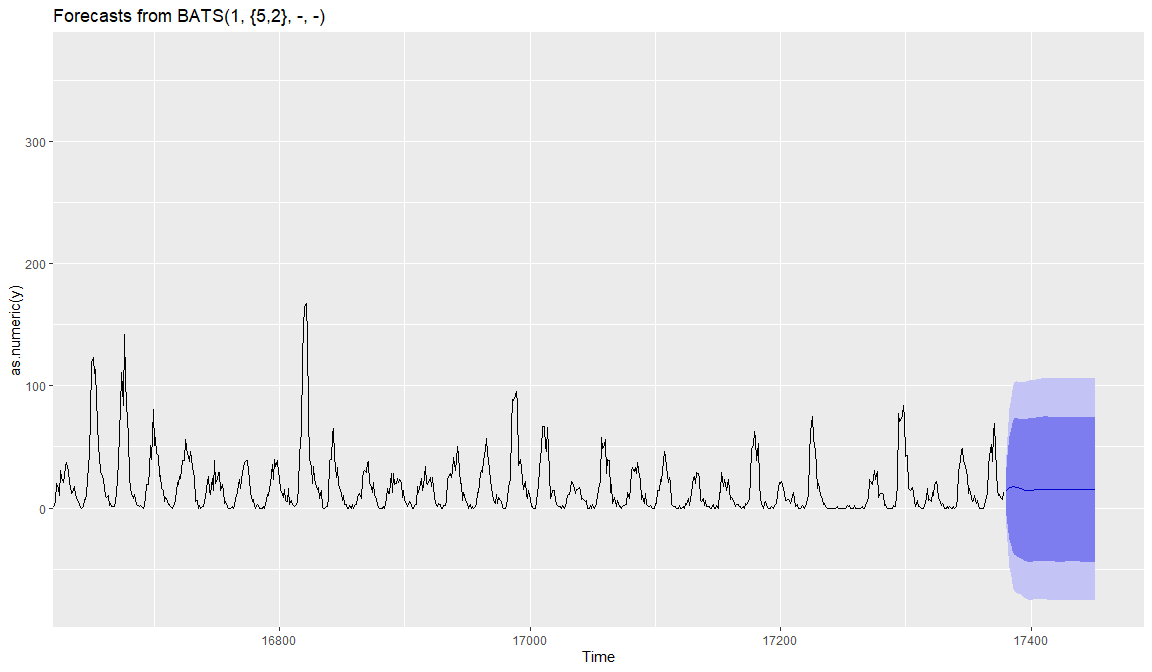
Figure



The Casual model looks to capture the daily trend the best, followed by the Registered and Total models. This is not too surprising as the Casual ridership was overall less influenced by Holidays and other specified seasonal events. Next a TBATS model is utilized to try and pull out further seasonality trends that may have been missed through the autoARIMA() function.

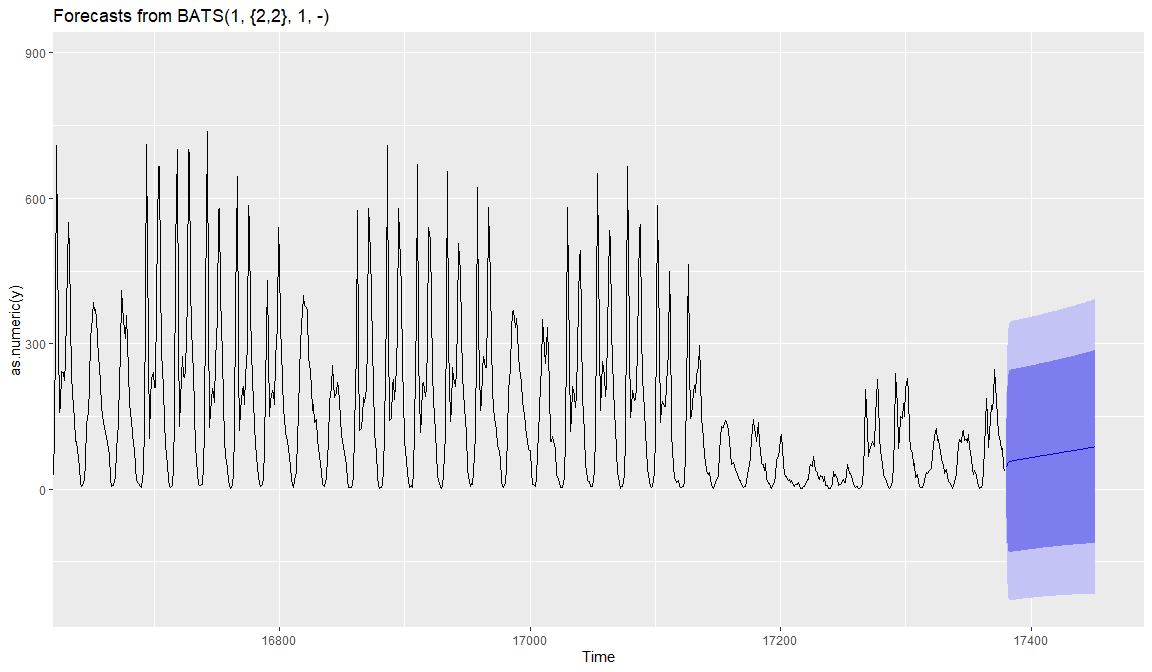
casual\_tbats<-tbats(casual\_ts)  
autoplot(forecast(casual\_tbats, h=72))+  
 coord\_cartesian(xlim = c(nrow(Casual)-720, nrow(Casual)+72))

Figure



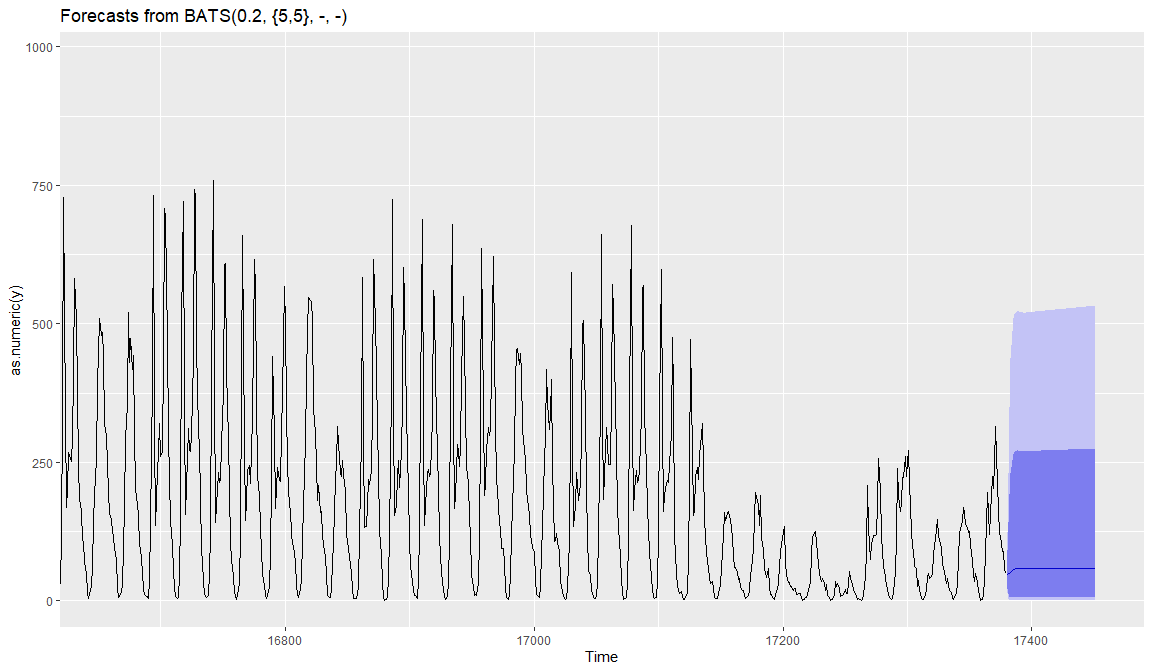
Registered\_tbats<-tbats(Registered\_ts)  
autoplot(forecast(Registered\_tbats, h=72))+  
 coord\_cartesian(xlim = c(nrow(Casual)-720, nrow(Casual)+72))

Figure



Total\_tbats<-tbats(Total\_ts)  
autoplot(forecast(Total\_tbats, h=72))+  
 coord\_cartesian(xlim = c(nrow(Casual)-720, nrow(Casual)+72))

Figure

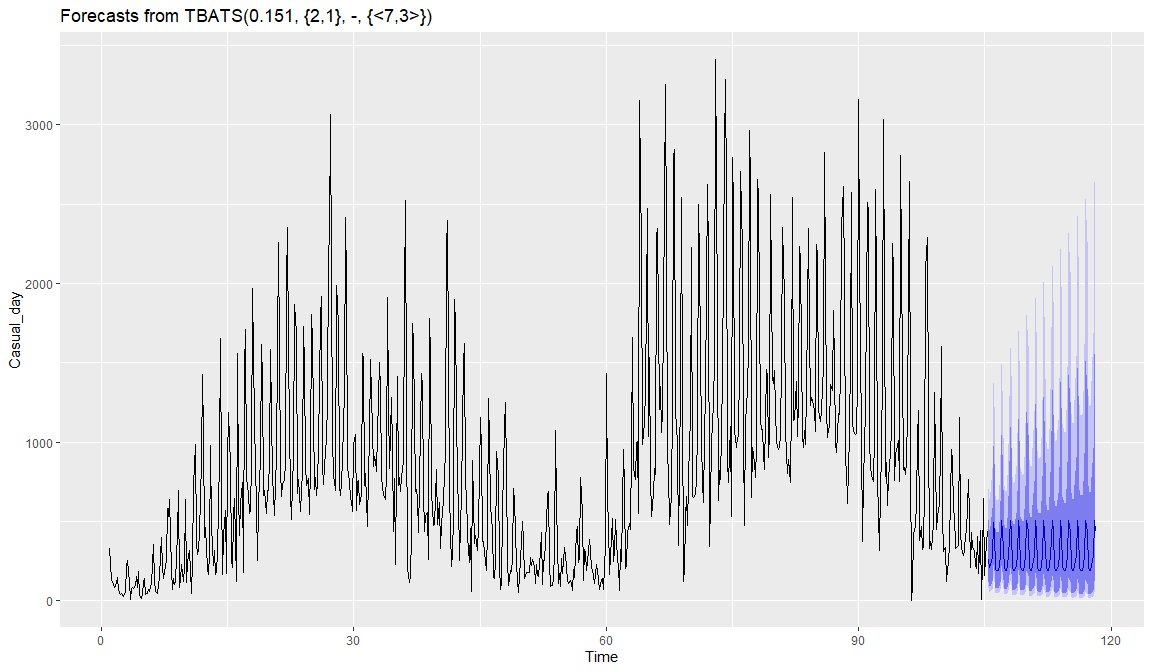


Overall, using time series analysis with autroARIMA() and TBATS() to predict hourly usage is difficult due to the many trends that occur throughout the year that were explored earlier in the report. Examples would be the season change with winter having the smallest change, down to the hourly changes where it spikes during commuting hours for registered users and when usage is around zero in the early morning before sunrise. Interestingly, the TBATS() function used on the Total Ridership is the only automated model that understands ridership will not go negative. Using the chosen model there on the other ridership Types may produce interesting results and would be a subject of future study. It is also possible to transform the dataset into daily data and perform the TBATS() function on the daily ridership, which produced similar model results for Casual Ridership as the Hourly Total Ridership above.

It is also possible to transform the dataset into daily data and perform the TBATS() function on the daily ridership, which produced similar model results for Casual Ridership as the Hourly Total Ridership above.

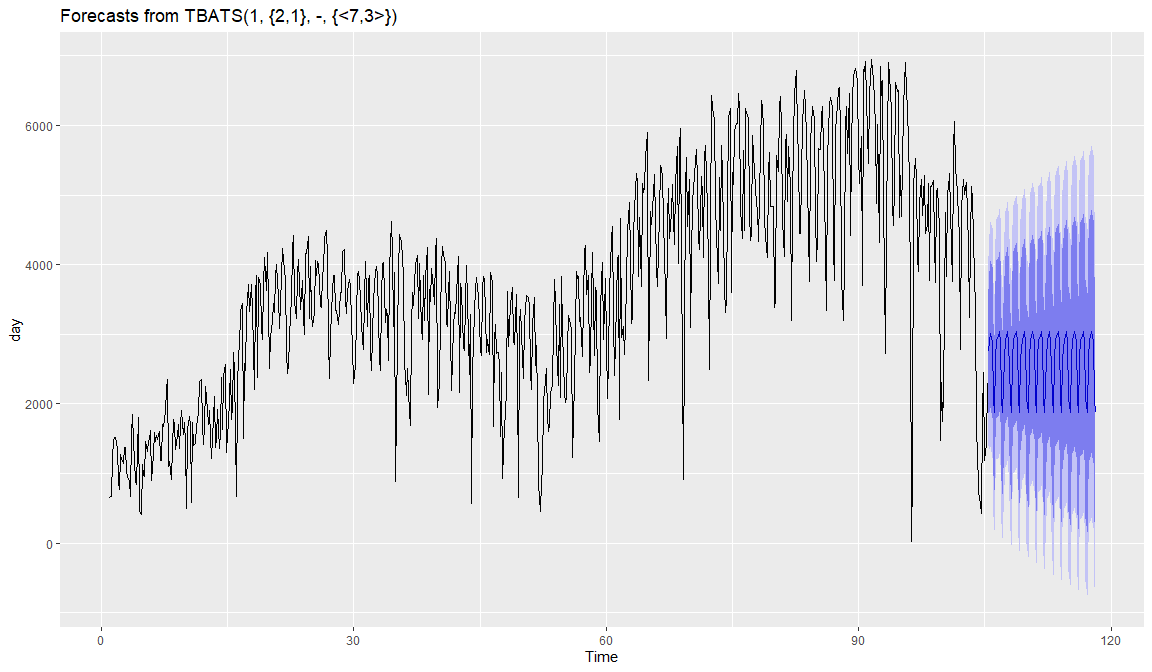
Casual\_day<-long%>%group\_by(Type,dteday)%>%summarise(day\_cnt=sum(Count))%>%filter(Type=="Casual")  
Casual\_day<-Casual\_day[3]%>%ts(frequency=7)  
Casual\_arima<-tbats(Casual\_day)  
autoplot(forecast(Casual\_arima, h=90))

Figure



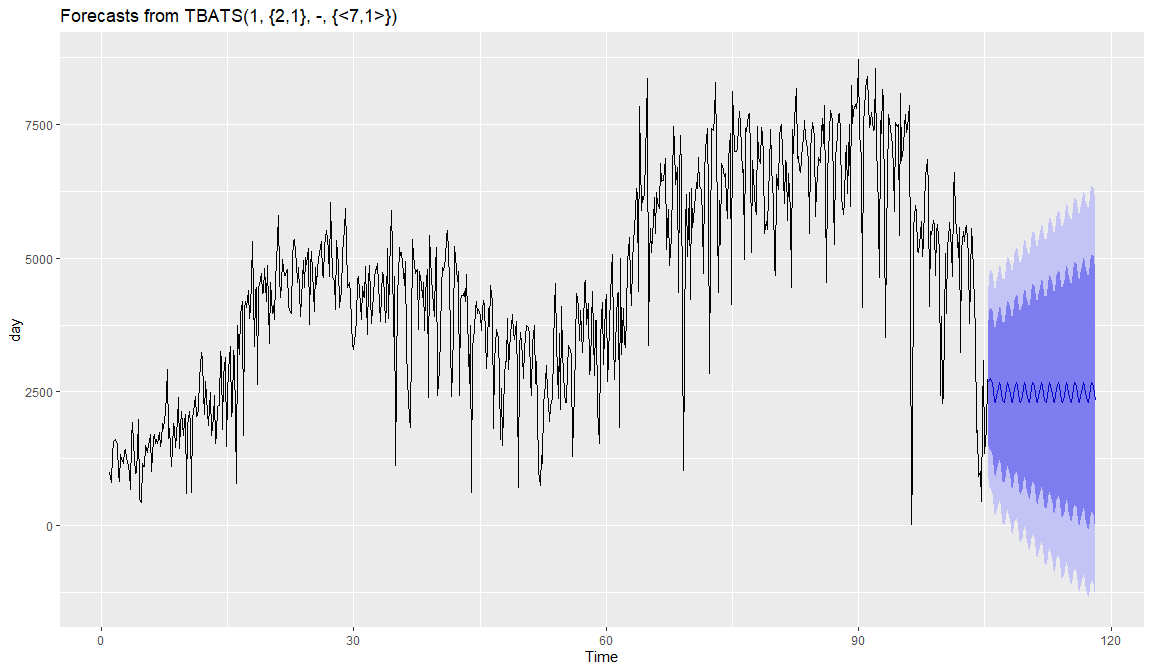
day<-long%>%group\_by(Type,dteday)%>%summarise(day\_cnt=sum(Count))%>%filter(Type=="Registered")  
day<-day[3]%>%ts(frequency=7)  
arima<-tbats(day)  
autoplot(forecast(arima, h=90))

Figure



day<-long%>%group\_by(Type,dteday)%>%summarise(day\_cnt=sum(Count))%>%filter(Type=="Total")  
day<-day[3]%>%ts(frequency=7)  
arima<-tbats(day)  
autoplot(forecast(arima, h=90))

Figure



Using the two models above which produced predictions above zero can be useful to predict the usage for the next day, which is a business problem bikeshare companies will come across. Understanding their market and timing bike repairs and expanding inventory is a crucial element to ensure riders can rely on the bike sharing company for their daily commute requirements.

# 

# References

[1] Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science.

[2] Fanaee-T, Hadi, and Gama, Joao, ‘Event labeling combining ensemble detectors and background knowledge’, Progress in Artificial Intelligence (2013): pp. 1-15, Springer Berlin Heidelberg

Web Source: <http://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset>