Understanding and Predicting Bicycle Sharing Rental Patterns: A Linear Modeling Approach

MA575 C1 Group 4: Yusen Wu

Abstract: In this paper, we explore and analyze the factors affecting bike rental behaviors using a linear modeling approach. We first conduct an in-depth exploratory data analysis to investigate the relationships between various variables and bike rentals. Our findings reveal rental demand are strongly associated with certain weather-related variables and are increasing as an overall trend. We then apply the Lasso algorithm for variable selection and construct an OLS model with the selected variables to predict daily total bike rentals. In addition, we develop separate models for different user types, discovering that these user groups exhibit distinct behavioral patterns. Future directions, limitations, and business implications are discussed.

1. Introduction

In recent years, bike-sharing rental systems have experienced a surge in popularity as an environmentally friendly, economical, and convenient mode of transportation in urban environments. This shift to bicycle sharing has had a significant impact on traffic congestion, air quality, and overall urban mobility. Recognizing these advantages, a growing number of cities around the world are adopting policies to implement the bike-sharing infrastructure. However, many of them face challenges in effectively allocating limited resources to maximize the benefits of these systems[1]. Therefore, statistical analysis to understand the key determinants and underlying trends of bike rentals is beyond significant. This study aims to comprehend and forecast bike-sharing rental patterns using a linear modeling approach. Our objectives are as follows: 1) Identify and select significant predictors that influence bicycle rentals. 2) Understand the underlying trends in bike rentals. 3) Construct a straightforward and interpretable model to track and predict daily bike rental usage. 4) Provide actionable insights to stakeholders for optimizing bike-sharing systems.

2. Background

2.1 Problem definition

Bike-sharing rental systems can be classified as docked and dockless systems. Docked systems enable users to rent bicycles from a designated docking station and return them to another docking station within the system. Dockless systems, on the other hand, rely on advanced technology to facilitate bicycle rentals without physical stations[2]. Our primary objective in this study is to identify a set of predictive variables that can effectively contribute to creating a robust linear model for understanding and forecasting daily bike rental demand, thus enabling stakeholders to make informed decisions for optimizing resource allocation.

2.2 Dataset and Preprocessing

Our research employs a dataset[3] gathered from Capital Bikeshare, the docked bike-sharing system operating in Washington, D.C. By January 2023, the network boasted more than 700 stations and 5,400 bicycles[4]. It was recognized as the largest bike-sharing service in the United States before the launch of Citi Bike in New York City. The dataset includes 731 daily bicycle rental counts in Washington, DC, from 2011 and 2012, arranged as a time series where observations are anticipated to exhibit correlation and dependence on one another. Daily rentals are categorized by casual and registered users, whose combined usage constitutes the total daily rentals. Weather-related data, such as standardized perceived temperature, humidity, and wind speed, have been incorporated from an external source. Time-related data, including season, month, holiday, weekday, and working day, is also provided in accordance with the local calendar.

No missing data has been identified. Normalized temperature and perceived temperature, humidity, and wind speed have been transformed back to their original scale, where the unit is available in the detailed column description in Table.1 below. Furthermore, weather conditions are categorized into good, mild, and bad, with their respective definitions also detailed in Table.1. A random sample of 5 data points is available in Appendix.1. Exploratory data analysis and model construction will be based on data in 2011, and model validation will be based on data in 2012 for a fair performance estimate.

Variable	Description

Year/Month/Season/Weekday	Current year/month/season/weekday of the data entry
Workingday	1 if day is neither weekend or holiday, 0 otherwise
Holiday	1 if day is a holiday, 0 otherwise
Weather	Good: Clear/Few clouds Mild: Mist/Cloudy/Broken clouds Bad: Light Snow/Rain/Thunderstorm/Scattered clouds
Perceived_Temp/Temp	Perceived temperature in °C/Temperature in °C
Humidity	Humidity in %
Windspeed	Wind speed in mph
Total/Registered/Casual Rentals	Total/registered/casual bike rentals per day

Table.1. Original Variable Information

2.3 Variable selection with Lasso

To obtain a simple and interpretable model, we aim to conduct a variable selection on a chosen set of features that are potentially predictive (outlined in section 3). The Least Absolute Shrinkage and Selection Operator (Lasso) is a regularization method developed by Robert Tibshirani in 1996 to tackle overfitting by incorporating an L1 penalty term into the OLS loss function[5]. This penalty promotes sparsity by forcing certain coefficients to zero, effectively excluding them from the model to diminish multicollinearity. We will apply the Lasso technique for variable selection and test the Lasso model for modeling and prediction.

2.4 Data modeling and prediction with OLS

Ordinary Least Squares (OLS) is a widely used statistical method in linear models. OLS determines the best-fitting line through the observations by minimizing the squared differences between observed and predicted responses. Its simple nature makes OLS suitable for numerous applications where a simple linear relationship exists between dependent and independent variables. Although the Lasso model mentioned above, with its selected variables, can be employed for modeling and prediction, there's potential for misuse with mistaken underlying assumptions for linear models. To this end, we will incorporate the selected variables into an additional OLS model, which offers a more convenient framework for diagnosing linear model assumptions.

3. Exploratory data analysis

Before applying more advanced Lasso variable selection and constructing our models, we conducted an exploratory data analysis to visualize and comprehend the key features of our dataset. This process helps us identify internal patterns, trends, outliers, and potential issues within the data. Throughout this analysis, we manually selected a set of potentially predictive variables from the original dataset and incorporated additional engineered variables based on our insights. A scatter plot/correlation matrix featuring all chosen features for variable selection and model construction will be provided at the end of this section.

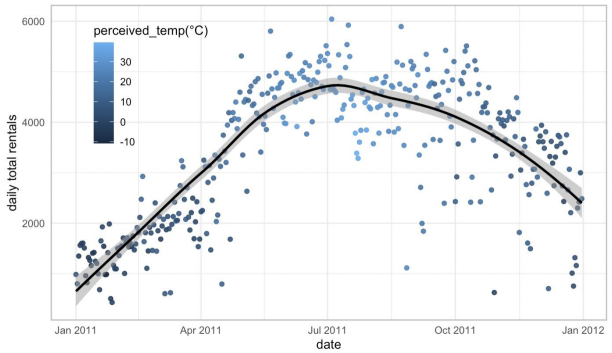


Figure.1: daily total rentals (2011)

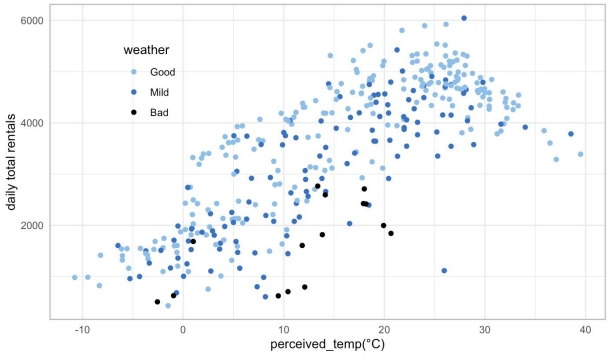


Figure.2: perceived_temp against daily total rentals (2011)

3.1 Bike rentals follow a clear seasonal pattern but are increasing as an overall trend

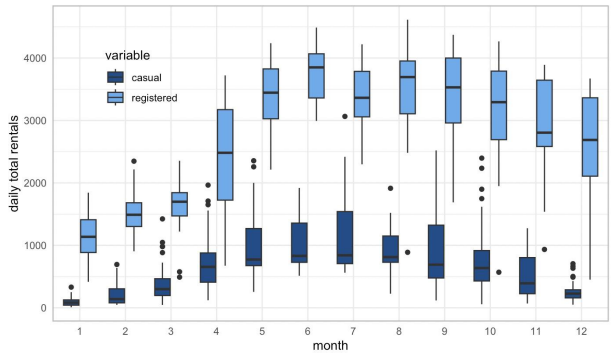
Figure.1 depicts daily total rentals from January to December in 2011, with each point color-coded to represent the perceived temperature on that day and a loess fit to show the trend. Lighter shades of blue indicate warmer days, while darker shades represent colder days. A distinct seasonal pattern is evident, with bike rental usage increasing from early spring, peaking in late summer, and declining through autumn until it reaches the bottom in winter, which is consistent with the boxplot between seasons and total rentals available as Appendix.2. Such seasonal variations also seem to strongly associate with change in perceived temperature across different times of the year, with summer temperatures typically ranging between 20 and 30 degrees Celsius, and winter temperatures frequently falling below 0 degrees Celsius. A one-way ANOVA is conducted to compare the mean perceived temperature across different seasons. Result in Appendix.3 indicate a significant difference in mean perceived temperature among different seasons. To further investigate these differences, we conducted a pairwise TukeyHSD test available as Appendix.4 and revealed important differences in perceived temperature between each pair of seasons. This is also consistent with the boxplot comparing perceived temperature among different seasons in Appendix.5. Therefore, seasonal variations for bike rentals could be well captured by change in perceived temperature, and season and month variables should be excluded for our model to avoid multicollinearity. In addition to the seasonal patterns, a general upward trend in daily bike rentals can also be discerned. A considerable increase in total rentals is evident by the end of 2011 compared to the beginning, even when perceived temperature and weathers are similar. This pattern may be attributed to user growth, prompting us to incorporate a day variable, which begins at 0 and increments by 1 for each subsequent day, to account for this upward trend.

3.2 Bike rentals are non-linearly correlated with perceived temperature

Figure.2 depicts the nonlinear relationship between daily total rentals and perceived temperature. The positive correlation between bike rentals and perceived temperatures ranging from -10 to 25 degrees Celsius indicates that as the temperature becomes more comfortable, bike rentals increase. However, when perceived temperatures exceed 25 degrees Celsius, bike rentals tend to decrease, suggesting that extremely hot temperatures might discourage people from biking. Additionally, the rate of change in daily bike rentals with respect to changes in perceived temperature is not consistent. For instance, the increase in bike rentals from -10°C to 0°C is more significant than the increase observed between 0°C and 25°C. Given the nonlinear pattern observed, we have chosen to include quadratic and cubic terms for perceived temperature in our variable selection and modeling to satisfy the linearity assumption required for our models.

3.3 There are substantially fewer bike rentals in mild and bad weather conditions

The scatterplot between perceived temperature and daily total bike rentals in Figure.2 features points color-coded based on their corresponding weather conditions, which are categorized as Good, Mild, or Bad, as previously defined. From the scatterplot, it is evident that there are more bike rentals during good weather conditions, such as clear and partly cloudy days. This suggests that people are more likely to rent bikes when the weather is comfortable and pleasant. On the other hand, mild weather conditions, characterized by misty and cloudy weather, lead to a decrease in bike rentals. There are almost no bike rentals during bad weather conditions, such as snow, rain, or thunderstorms as most people may choose not to rent bikes when the weather is potentially hazardous. A one-way ANOVA test is conducted and suggests significant difference in the mean daily total rentals across different weathers available in Appendix.6. In addition, total bike rentals seem to react differently to temperature change under different weathers depicted in Figure.2, so we include an interaction term between perceived temperature and weather for variable selection.



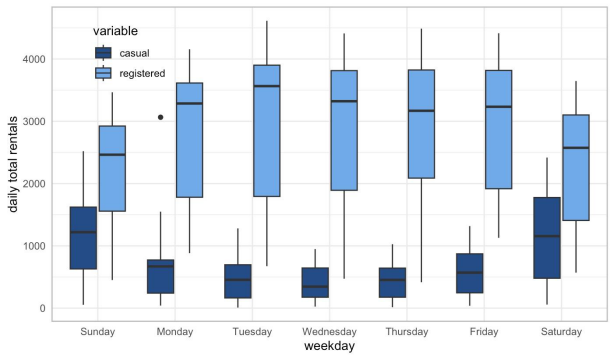


Figure.3: daily bike rentals against month by user types (2011)

Figure.4: daily bike rentals against weekday by user types (2011)

3.4 Casual and registered users exhibit different behaviors

Figure.3 and Figure.4 present boxplots of daily rentals categorized by months and weekdays and are color-coded to distinguish between registered and casual users. Analyzing Figure.3, we observed a similar seasonal pattern in both registered and casual users that was seen in Figure.1 for total rentals. However, a more pronounced growth trend is apparent for registered users, while such a trend is almost negligible for casual users. When examining Figure.4, we find that registered users are more active from Monday to Friday, while casual users are more prevalent on weekends. Registered users might primarily use the rental service for commuting to work or school during the weekdays, while casual users could be using the service for leisure activities or occasional errands, which tend to occur more frequently on weekends. We will create separate models for registered and casual users later to verify the observed patterns discovered above.

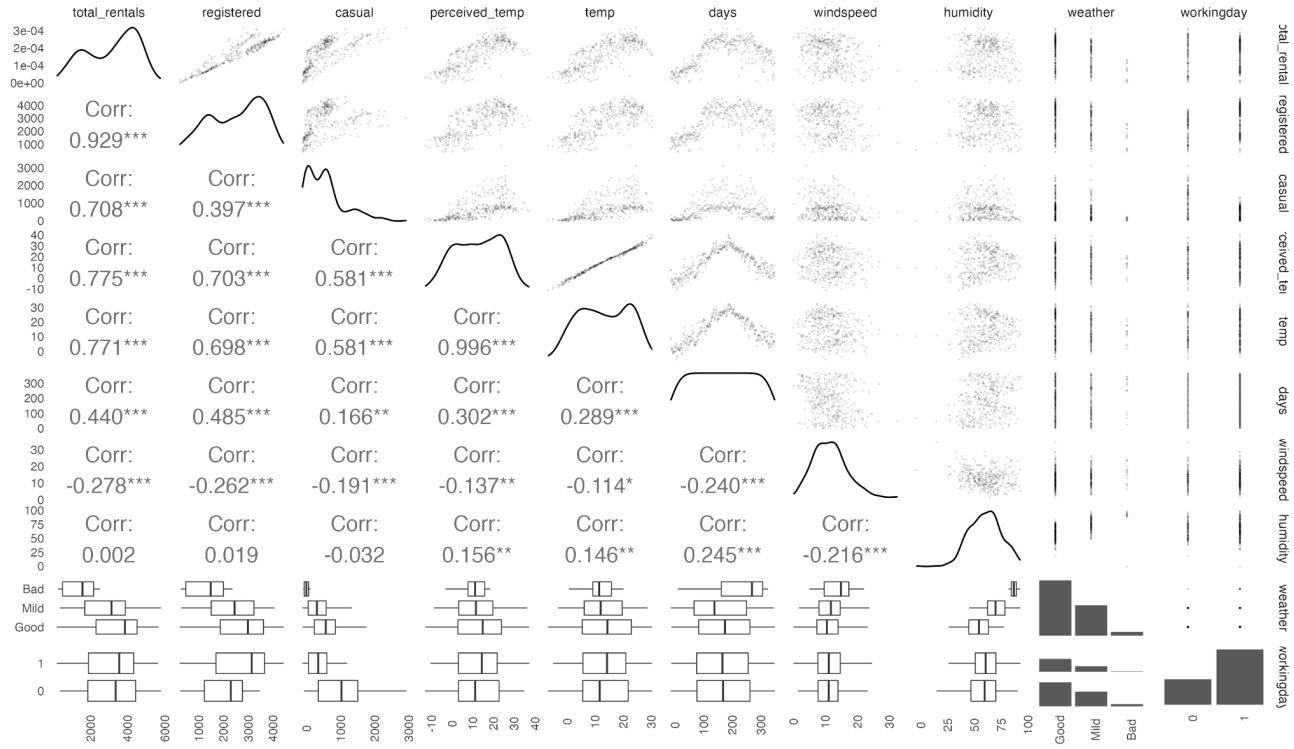


Figure.5: Scatter Plot/Correlation Matrix (2011)

3.5 Most time and weather-related variables exhibit a strong correlation with usage-related data

As stated in section 2, our objective is to create a linear model capable of efficiently predicting, modeling, and controlling daily bike rentals using pertinent information. In the context of linear modeling, our independent variables consist of weather and time-related data, while the dependent variables are usage-related data, including total, registered, and casual rentals each day.

In Figure.5, a scatterplot and correlation matrix are used to examine the linearity between various independent and dependent variables. Most continuous variables, such as days and wind speed, demonstrate a linear correlation with total rentals. The relationship between perceived temperature and total rentals is particularly strong, with a correlation coefficient of approximately 0.775, although a nonlinear connection may exist. Humidity exhibits a nearly zero correlation with all three dependent variables. Still, we would like to include it in the variable selection process as it might be predictive when controlling effects from other independent variables. It is also worth noting that the days variable exhibits a higher correlation with registered users compared to casual users and total rentals. This observation supports our earlier assumption that the growth trend is more pronounced among registered users.

Multicollinearity does not appear to be a major issue, as correlations among independent variables are generally not overly strong, except for the relationship between perceived temperature and temperature. We choose perceived temperature over temperature because it correlates more strongly with all three dependent variables total, registered, and casual rentals. Also logically, perceived temperature is what people truly feel

while outside, which can directly influence the choice of whether a user will choose to use the bike rental system on a particular day or not.

The matrix's diagonal displays histograms for each variable to help check if transformation is needed, with most continuous variables exhibiting a normal-like distribution—which is optimal for building linear models. However, wind speed and casual rentals show right-skewed distributions. To further determine if a transformation is required, we apply the Box-Cox algorithm to each continuous variable. As anticipated, most variables suggest a lambda value of approximately 1, while wind speed and casual rentals have a recommended lambda of 0.5(available as Appendix.7). Therefore, we perform a square root transformation on wind speed and casual rentals before fitting a linear model.

3.6 Final variables considered for variable selection and model construction

Table.2 provides a summary of the variables considered for model construction and selection. To avoid multicollinearity with perceived temperature, month and season variables were excluded. The holiday variable was also excluded due to its limited number of positive values and varying rental patterns across different holidays. For instance, Independence Day had the highest number of rentals in 2011, whereas Christmas had the lowest. All remaining variables from the original dataset along with the newly engineered variables(bolded in Table.2) will proceed for variable selection and model construction. Categorical variables including weather, workingday, and weekday are transformed into dummy variables being fit to the models.

Independent Variable (Continuous)	days, humidity, sqrt(windspeed), perceived_temp**, perceived_temp^2, perceived_temp^3, perceived_temp:weathermild, perceived_temp:weatherbad**
Independent Variable (Categorical)	workingday, Tuesday, Wednesday, Thursday, Friday, Saturday, Sunday, weathermild, weatherbad
Dependent Variable	total_rentals, sqrt(casual_rentals), registered_rentals

Table.2. Variables for Selection and Model Construction

4. Variable selection and model construction

In this paper, our final prediction target will be daily total rentals as it offers a holistic representation of bike usage and is directly influenced by both registered and casual rentals. In this section, we will start by constructing a Lasso model for variable selection. As followed, we will analyze and refine the set of selected variables, and then employ it to develop an OLS model. Both models will have daily total rentals as their dependent variables. In addition, daily registered and casual rentals exhibit distinct behaviors, yet their combined sum constitutes daily total rentals. Therefore, we will construct two more OLS models separately for each user type to investigate their differences. During the validation phase, we will combine the predictions from these two models to forecast daily total rentals. All models mentioned above will be trained with data in 2011 and validate with data in 2012.

4.1 Lasso model for variable selection(total rentals in 2011)

After transforming categorical variables into dummy variables, we now have a total of 9 dummy and 8 continuous variables to predict daily total bike rentals. To obtain a simple and interpretable model, we will begin by applying the Lasso algorithm for variable selection. The effectiveness of the Lasso algorithm can be attributed to its L1 penalty term, which efficiently shrinks the coefficients of insignificant variables toward zero. We would like to build a Lasso model with all independent variables mentioned above to model total rentals. Variables with coefficients that were not reduced to zero by Lasso will be deemed as important. The intensity of the penalty is governed by a hyperparameter denoted as lambda. A common approach for selecting an optimal lambda is through cross-validation. This technique evaluates the performance of the Lasso algorithm under various lambda values by dividing the training data into multiple folds. The algorithm is trained on certain folds and validated on a remaining fold, with this process repeated for each fold.

However, this is not applicable to our dataset as a time series, because traditional cross-validation with random sampling can disrupt the temporal patterns across time-dependent observations. To tackle this issue, we customized a rolling cross-validation method designed specifically for a time series, dividing it into multiple

non-overlapping windows[6]. Each training set is created using an earlier window, while the corresponding testing set is made by the subsequent window. The windows then "roll" or "slide" forward by a certain step size, generating new training and validation sets. This process continues until the entire dataset has been utilized for testing. This approach effectively preserves the integrity of the temporal patterns in the data during model evaluation. Figure.6 demonstrates the workflow of this method.

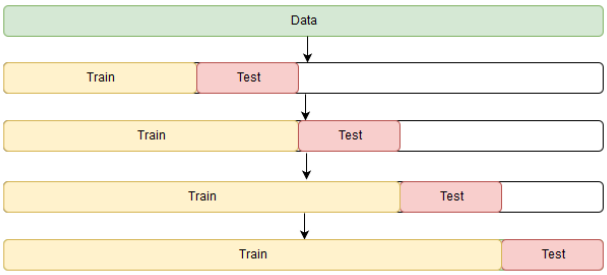


Figure.6: Rolling cross-validation diagram

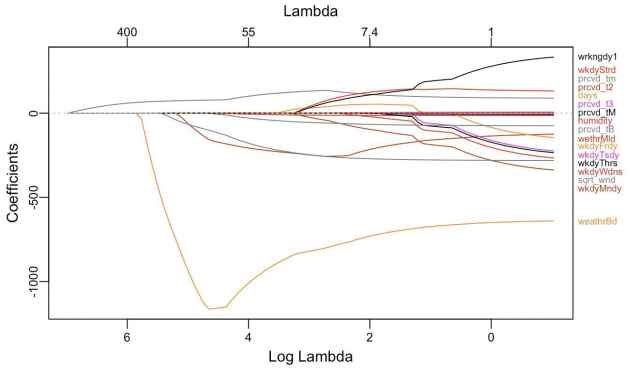


Figure.7: Lasso plot

We conducted a hyperparameter search over a range from 0.001 to 1000 with 100 values suggested by the glmnet package in R and documented the cross-validation performance of each penalty strength. All independent variables were standardized to ensure a fair penalty. The Lasso plot in Figure.7 demonstrates how each covariate shrinks to 0 as the penalty increases with larger lambda values. As we anticipated, perceived temperature and weather appear to be the most predictive variables that shrink to 0 only when lambda is larger than 400. The days variable is also significant, demonstrating our previous assumption of a general upward trend in user growth. The optimal lambda value is around 103 resulting in a parsimonious model with only 7 variables summarized as the formula below:

*total_rentals = 2299.52 - 160.21*weathermild - 1164.07*weatherbad + 77.66*perceived_temp - 0.04*humidity + 2.31*days - 2.51*perceived_temp:weatherbad - 102.55*sqrt(windspeed)*

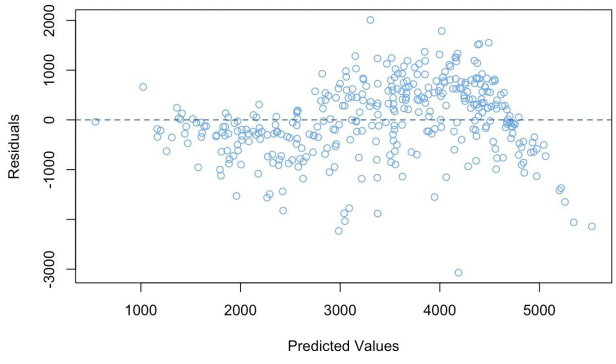


Figure.8: Training residual plot for Lasso model

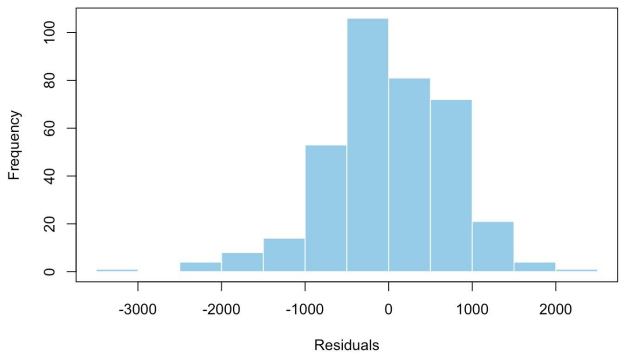


Figure.9: Histogram for training residuals

Although the Lasso model effectively selects a set of predictive variables, we need to determine if it adequately models the training data. The variable selection by the Lasso model appears to underperform, as evidenced by the training residual plot in Figure.8. We can observe heteroscedasticity, with residuals showing larger variance when the predicted value is less than 2500 or greater than 4500. Additionally, a fluctuating pattern is detected, which could be caused by the exclusion of quadratic and cubic terms for perceived temperature. The normality assumption of the linear model is not met demonstrated in Figure 9, where the residuals exhibit a left-skewed distribution. Furthermore, a few outliers with extremely low residuals are deleted: points 238 (Hurricane Irene) and 358 (Christmas in 2011).

After deleting the outliers, we decided to tackle the fluctuating pattern in the training residual plot. As shown in Figure.7, perceived temperature appears to be the most significant predictor while it is non-linearly correlated with the dependent variable. Therefore, we decided to include the quadratic and cubic terms of perceived temperature to account for the nonlinear pattern. The quadratic and cubic terms along with the other variables selected by Lasso will build an improved OLS model.

4.2 OLS model(total rentals in 2011)

	Dependent variable:
	total_rentals
perceived_temp	84.762*** (10.904)
perceived_temp2	5.694*** (0.831)
perceived_temp3	-0.187*** (0.018)
weatherBad	-700.269** (298.283)
sqrt_windspeed	-247.059*** (44.009)
days	3.032*** (0.330)
humidity	-13.923*** (2.762)
weatherMild	-218.592*** (77.793)
perceived_temp:weatherBad	-72.547*** (20.483)
Constant	3,163.185*** (256.408)
Observations	363
R2	0.840
Adjusted R2	0.836
Residual Std. Error	555.196 (df = 353)
F Statistic	206.222*** (df = 9; 353)
Note:	*p<0.1; **p<0.05; ***p<0.01

Table.3: OLS model output(total rentals)

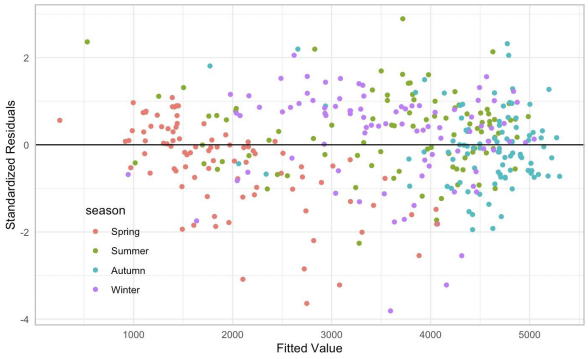


Figure.10: Standardized residual plot for OLS model(total rentals)

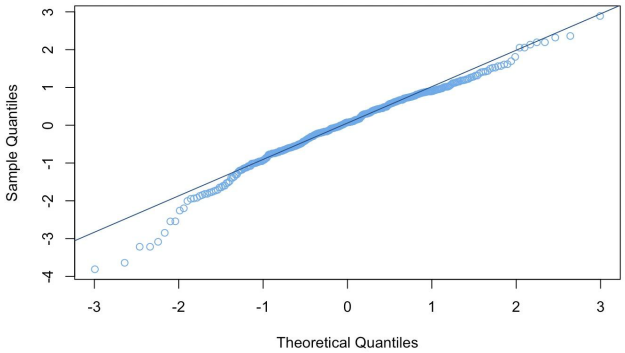
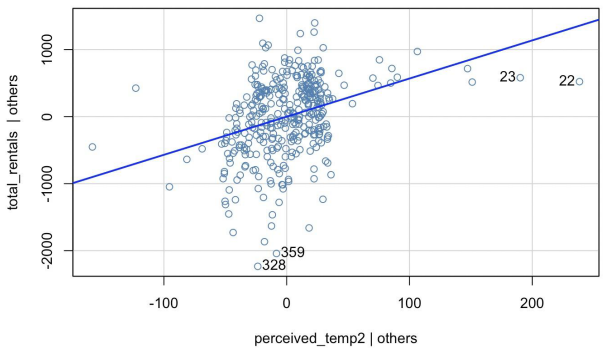


Figure.11: Normal QQ plot for OLS model(total rentals)

This improved OLS model includes all variables selected by Lasso and the quadratic and cubic terms of perceived temperature to predict daily total rentals. The regression output is available in Table.3, and full output with t-value, p value, and confidence interval for each parameter in available in Appendix.8. The model demonstrates overall significance, with a p-value lower than 0.01 for its F-statistic, and accounts for 84.5% of the variance in the dependent variable.

Figure.10 displays the standardized residual plot, which reveals only slight heteroscedasticity as residuals with middle-fitted values exhibiting a larger variance. Despite this, there is no fluctuating pattern detected, marking a notable improvement compared to the Lasso model. Residuals are color-coded by season, and it is observed that most residuals smaller than -2 occur in spring and winter. This is understandable, as more extreme weather in these seasons could lead the model to overestimate. The normal QQ plot shown in Figure.11 and histogram in Appendix.9 indicate that residuals are not perfectly normally distributed, as points deviate from the diagonal line at the tails and head, which could pose a potential limitation for the model.

Most variables are statistically significant with a p-value less than 0.01. The days variable has a positive coefficient of around 3, which again demonstrates our observations of a general upward growth trend in total rentals. We also see fewer rentals when weather conditions are mild or bad. Perceived temperature and its quadratic and cubic terms are all significant. As the quadratic and cubic terms of perceived temperature are added variables in addition to the Lasso model, two added variable plots are given as Figure.12 and Figure.13 to demonstrate that including them is a wise decision as it is observable that both quadratic and cubic terms exhibit a significant slope. The addition of these two variables could account for the disappearance of the fluctuating pattern in the standardized residual plot for the OLS model compared to the Lasso Model.



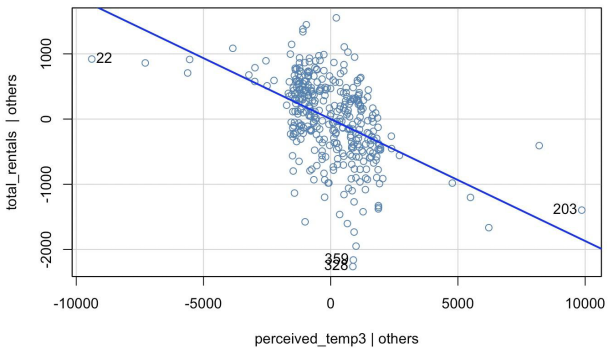


Figure.12: Added variable plot for percevied_temp^2

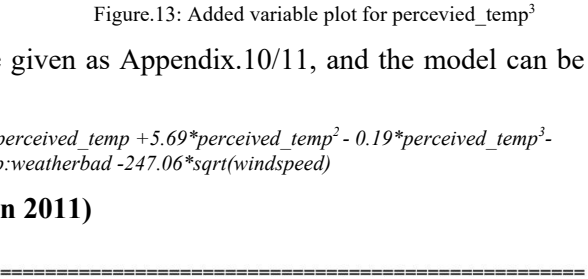


Figure.13: Added variable plot for percevied_temp^3

Complete marginal model plot and added variable plot are given as Appendix.10/11, and the model can be summarized as the following formula:

*total_rentals = 3163.18 - 218.59*weathermild -700.27*weatherbad + 84.76*perceived_temp +5.69*perceived_temp^2 - 0.19*perceived_temp^3- 13.92*humidity + 3.03*days -72.54*perceived_temp:weatherbad -247.06*sqrt(windspeed)*

4.3 Separate OLS models(registered and casual rentals in 2011)

	Dependent variable:
	registered
perceived_temp	49.745*** (8.848)
perceived_temp2	4.172*** (0.674)
perceived_temp3	-0.132*** (0.015)
days	3.258*** (0.268)
weatherBad	-728.954*** (242.092)
humidity	-9.543*** (2.243)
sqrt_windspeed	-156.117*** (35.711)
weatherMild	-142.992** (63.398)
workingday1	690.643*** (51.611)
perceived_temp:weatherBad	-45.964*** (16.639)
Constant	1,894.354*** (211.149)
Observations	363
R2	0.822
Adjusted R2	0.817
Residual Std. Error	450.509 (df = 352)
F Statistic	162.943*** (df = 10; 352)
Note:	*p<0.1; **p<0.05; ***p<0.01

Table.4: OLS model output(registered rentals)

	Dependent variable:
	log(casual)
perceived_temp	0.135*** (0.008)
perceived_temp2	-0.002*** (0.001)
perceived_temp3	-0.00003* (0.00001)
weatherBad	-0.719*** (0.224)
humidity	-0.011*** (0.002)
sqrt_windspeed	-0.122*** (0.032)
weatherMild	-0.199*** (0.058)
workingday1	-0.870*** (0.048)
perceived_temp:weatherBad	-0.028* (0.015)
Constant	6.714*** (0.189)
Observations	363
R2	0.840
Adjusted R2	0.835
Residual Std. Error	0.418 (df = 353)
F Statistic	205.192*** (df = 9; 353)
Note:	*p<0.1; **p<0.05; ***p<0.01

Table.5: OLS model output(casual rentals)

In addition to the models mentioned above, we constructed two OLS models for registered and casual users individually to capture their distinct behaviors. For both models, we primarily utilized the same set of independent variables from the earlier OLS model for total rentals, with minor adjustments tailored to each user type. In the following discussion, we will refer to the prior OLS model developed for total rentals when mentioning the model for total users.

Table.4 presents the regression results for the OLS model tailored to registered users. The model is overall significant, and all variables are important with a p-value less than 0.01. Distinct from the model for total rentals, we included the workingday variable, as a higher number of registered users were observed on weekdays. The days variable, which accounts for user growth trends, is more prominent in the registered users model compared to the model of the total rentals, where coefficient t-values are approximately 12.16 for the former and 9.18 for the latter. This corroborates our observations of more registered users by the end of 2011 compared to that of the start as depicted in Figure.3. The workingday predictor also indicates more registered rentals during the weekdays as depicted in Figure.4. Full model output and diagnostic plots are available as Appendix.12.

Table.4 showcases the regression outcome for the OLS model designed for casual users. The model is overall significant, but not all variables are important. Different from the model of the total rentals, we incorporated the workingday variable to determine whether there is a higher prevalence of casual users during weekends and holidays. The days variable is excluded from this model, as it was found to be insignificant in a prior experiment. This finding aligns with the observation that the number of users by the end of 2011 was largely similar to that at the beginning of the year. As previously mentioned, we conduct a log transformation to the dependent variable as it was found to be heavily right skewed. Similar to the model for registered users, the workingday variable is significant with a negative coefficient, suggesting more casual rentals during weekends or holidays. In addition, the coefficient t-value for perceived temperature is around 17.95 for casual rentals and 5.6 for registered rentals, which suggests that casual users might be more sensitive to temperature change. Full model output and diagnostic plots are available as Appendix.13.

In the validation stage, we will add the prediction from the models for registered and casual users to predict total rentals. These models can be summarized as follows:

$$registered\ rentals = 1894.35 - 142.99*weathermild - 728.95*weatherbad + 49.75*perceived\_temp + 4.17*perceived\_temp^2 - 0.13*perceived\_temp^3 - 9.54*humidity + 3.25*days - 45.96*perceived\_temp{:}weatherbad - 156.11*sqrt(windspeed) + 690.64*workingday$$

$$log(casual\ rentals) = 6.71 - 0.19*weathermild - 0.72*weatherbad + 0.14*perceived\_temp + 0.002*perceived\_temp^2 - 0.00003*perceived\_temp^3 - 0.01*humidity - 0.03*perceived\_temp{:}weatherbad - 0.12*sqrt(windspeed) - 0.87*workingday$$

$$total\ rentals = registered\ rentals + exp(log(casual\ rentals))$$

5. Model Prediction

Before proceeding with model validation on data in 2012, we must choose a final model from the following alternatives: the Lasso model for total rentals, the OLS model for total rentals, and the combination of the OLS models for registered and casual rentals. While the Lasso model is the simplest, it fails to meet the linearity assumption required for linear models. Combining OLS models for registered and casual users is anticipated to be the most accurate but is overly complicated. The OLS model for total rentals is simple, interpretable, and overall valid, which becomes our final model.

	Lasso (total rentals)	OLS (total rentals)	OLS (Registered + Casual rental)
Training Root MSE	736.46	**547.49**	575.90
Validating Root MSE	1739.94	1313.71	**1265.76**
Validating Relative MSE	0.087	0.049	**0.046**

Table.6: Final prediction results (2012)

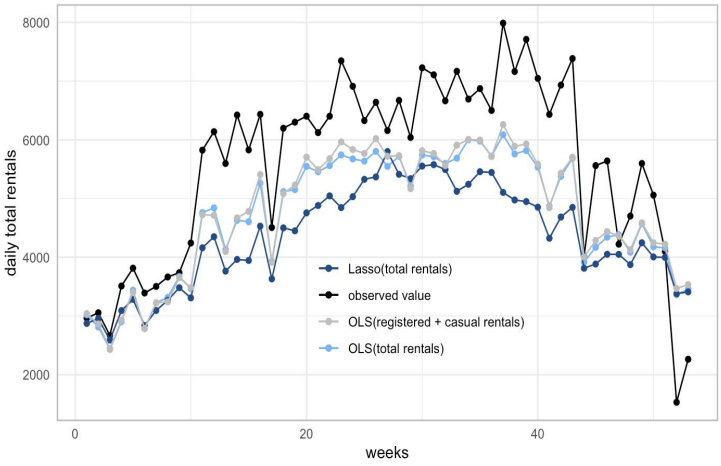


Figure.14: Observed value against predictions ordered by week (2012)

Table.6 summarizes model training and validating performance. The Lasso model appears to underfit the data, as evidenced by the low values for both the training and validation root MSE. As expected, merging the predictions from the OLS models for registered and casual rentals leads to better performance, as evidenced by the lowest validation root MSE and relative MSE. However, the improvement gained by such a combination is only marginal when compared to the performance of our final model: the OLS model for total rentals. This demonstrates our choice of final models is sensible. Even with a simple and interpretable design, our final model correctly predicts and monitors the seasonal fluctuations shown in Figure.14.

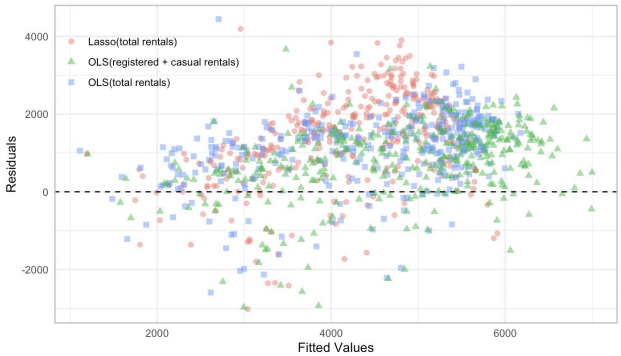


Figure.15: Predicted residual plot (2012)

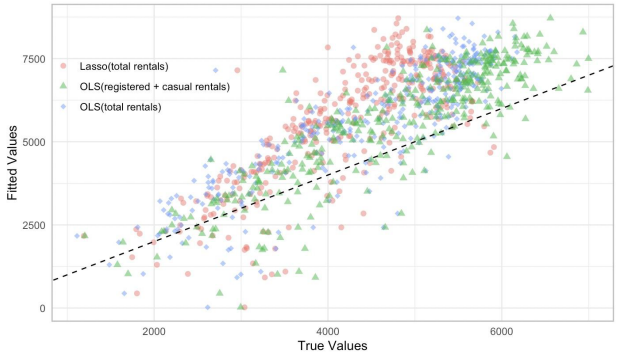


Figure.16: Observed value against predictions (2012)

However, Figure.14 reveals a significant discrepancy between the predictions from all three models and the observed values in 2012. Each model tends to underestimate the daily total rentals from late spring to early winter of 2012, with the largest discrepancy of around 2000 occurring in late summer for our final model. This observation is also evident in Figure.15, where a substantial portion of predicted residuals are positive for all three models. The predicted residuals for all three models on the validation data exhibit an upward trend,

indicating that the underestimation increases with the growth of predicted values. As illustrated in Figure.16, the underestimation appears to be more pronounced for the Lasso model and less severe for the OLS model for total rentals and the combined OLS model for registered and casual rentals. A training residual plot for all three models is available as Appendix.14.

To investigate why the model succeeded in predicting fluctuations but failed to capture the trend, we recreated the plot for daily total rentals in Figure.1 but included the year 2012 (available as Appendix.15). We discovered that while the variation in perceived temperature and weather conditions was similar in 2011 and 2012, the surge in daily total rentals was beyond what we could observe from the 2011 data. The summer peak period for bike rentals in 2012 reached around 7500 per day, compared to 5000 for the same period in 2011. We also remade the boxplot between registered and casual rentals and month in Figure.3 to observe user growth from 2011 to 2012 (available as Appendix.16). Consistent with our previous observation in 2011, registered users experienced a pronounced increase in 2012, while the count of casual users was similar in both 2011 and 2012. Therefore, to account for the discrepancy between our prediction and observed value, it is necessary to estimate the changing user base for each type of users and incorporate such information into our models.

5. Conclusion and business implications

In conclusion, this study successfully applied a linear modeling approach to analyze and predict bicycle-sharing rental patterns. Our findings and respective business implications are as follows:

1) Our analysis successfully pinpointed perceived temperature and weather conditions as two of the most influential variables for predicting daily total bike rentals. We observed that bike rentals tend to be more frequent when the perceived temperature is comfortable, typically ranging between 15 to 25 degrees Celsius, and during days with clear skies or few clouds. Bike rentals also exhibit a clear seasonal trend with respect to perceived temperature variations, where summer experiences more rentals and winter sees significantly fewer. Therefore, it is beyond significant for bike-sharing systems to closely monitor weather and anticipated temperature trends to optimize their resource allocation, ensuring adequate bike availability and maintenance during favorable conditions. Furthermore, bike-sharing systems should proactively prepare for slumps and surges in demand across different seasons.

2) Our study indicates that there are distinct rental patterns and behaviors among registered and casual users. Over the course of 2011, a significant upward trend was seen in the number of registered users, while casual users did not display a similar growth trajectory. Rentals from registered users often take place on weekdays, likely for commuting reasons, while casual rentals tend to be more common on weekends and holidays, possibly for recreational purposes. For bike-sharing systems, registered and casual users should be regarded as two separate markets, each requiring distinct growth strategies. For registered users, companies could design specialized loyalty programs to encourage repeat usage and long-term commitment, such as offering discounted rates or rewards for frequent users. For casual users, companies should aim to convert them into registered users by offering trial membership and flexible membership options.

3) We successfully constructed three linear models to forecast daily total rentals, and we recommend the final OLS model for total rentals as a simple and interpretable approach for predicting future bike-sharing demand. However, our linear models training on 2011 data underestimates the demand of the validating data in 2012. Even if we correctly identified the user growth trend in 2011 and tried to account for that in our model, we still failed to track its progression into 2012. Considering such projected growth, bike-sharing companies should proactively expand the existing infrastructure, including bike stations, bike lanes, and parking facilities, to accommodate the increasing demand. Related government agencies should collaborate with these companies to create a comprehensive regulatory framework that ensures safety and accessibility.

Our study also comes with certain limitations: 1) Incorporating the days variable alone is insufficient to fully exploit the time-dependent patterns present in our dataset. An autocorrelation plot in Appendix.17 shows that total rentals are strongly correlated with their past value. We tried implementing a generalized least square regression with an autocorrelation of 1, but the performance is still limited. More sophisticated time series models with an autoregressive fashion can be promising including MA, ARMA, ARIMA, or SARIMA. 2) Our model, trained on 2011 data from Washington DC, might not produce reliable predictions for Boston in 2023 due to disparities in both location and time frame. To improve the model's generalizability, it would be essential to include more recent data and local information.

Reference

[1] Taylor & Francis, “Promoting Cycling for Transport: Research Needs and Challenges.” https://www.tandfonline.com doi/full/10.1080/01441647.2013.860204.

[2] “Bicycle-Sharing System.” *Wikipedia*, Wikimedia Foundation, 20 Mar. 2023, https://en.wikipedia.org/wiki/Bicycle-sharing_system.

[3] *UCI Machine Learning Repository: Bike Sharing Dataset Data Set*, https://archive.ics.uci.edu/ml/datasets/bike+sharing+dataset.

[4] “Capital Bikeshare.” *Wikipedia*, Wikimedia Foundation, 10 Apr. 2023, https://en.wikipedia.org/wiki/Capital_Bikeshare.

[5] Ranstam, J, and J A Cook. “Lasso Regression.” *OUP Academic*, Oxford University Press, 7 Aug. 2018, https://academic.oup.com/bjs/article/105/10/1348/6122951.

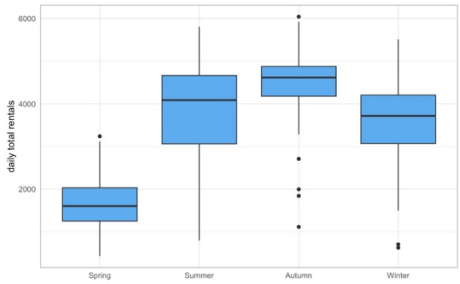
[6] Bergmeir, et al. “On the Use of Cross-Validation for Time Series Predictor Evaluation.” *Information Sciences*, Elsevier, 4 Jan. 2012, https://www.sciencedirect.com/science /article/pii/S0020025511006773.

Appendix

Appendix.1: 5 random samples from the original dataset

date	season	year	month	holiday	weekday	workingday	weather	temp	perceived_temp	humidity	windspeed	casual	registered	total rentals
2/10/11	1	0	2	0	4	1	1	0.144348	0.149548	0.437391	0.221935	47	1491	1538
3/11/11	1	0	3	0	5	1	2	0.316522	0.305	0.649565	0.23297	247	1730	1977
10/14/11	4	0	10	0	5	1	2	0.550833	0.529675	0.71625	0.223883	529	3115	3644
6/19/11	2	0	6	0	0	0	2	0.699167	0.645846	0.666667	0.102	1639	3105	4744
12/16/11	4	0	12	0	5	1	2	0.375	0.359825	0.500417	0.260575	178	3399	3577

Appendix.2: Boxplot between total rentals and seasons (2011)



Appendix.3: One way ANOVA between perceived temperature and seasons

```
             Df Sum Sq Mean Sq F value Pr(>F)
season        3  30365   10122   246.3 <2e-16 ***
Residuals   361  14833      41
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

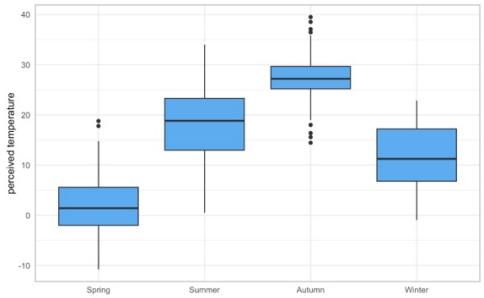
Appendix.4: Pairwise TukeyHSD test between perceived temperature and seasons

```
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = perceived_temp ~ season, data = bikedata %>% filter(yr == 0))

$season
                    diff        lwr        upr p adj
Summer-Spring  15.481326  13.028442  17.934211     0
Autumn-Spring  24.982237  22.542290  27.422183     0
Winter-Spring   9.418552   6.945309  11.891795     0
Autumn-Summer   9.500910   7.074551  11.927270     0
Winter-Summer  -6.062774  -8.522614  -3.602934     0
Winter-Autumn -15.563684 -18.010623 -13.116745     0
```

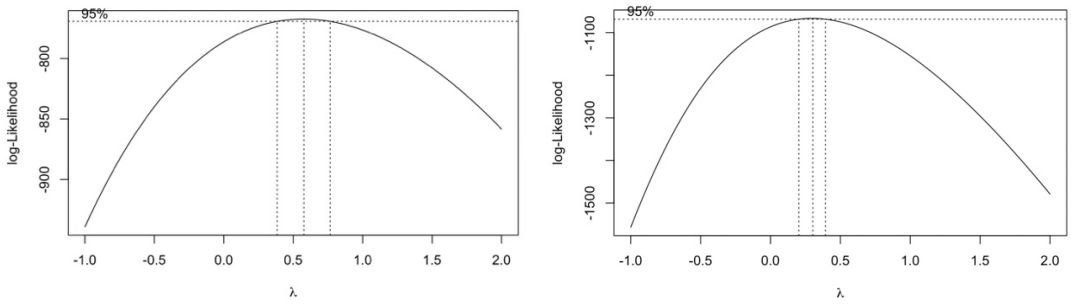
Appendix.5: Boxplot between perceived temperature and seasons



Appendix.6: One way ANOVA between total rentals and weather

```
             Df    Sum Sq  Mean Sq F value   Pr(>F)
weather       2  76396169 38198084   22.46 6.36e-10 ***
Residuals   362 615553880  1700425
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Appendix.7: Box-Cox result for windspeed and casual rentals

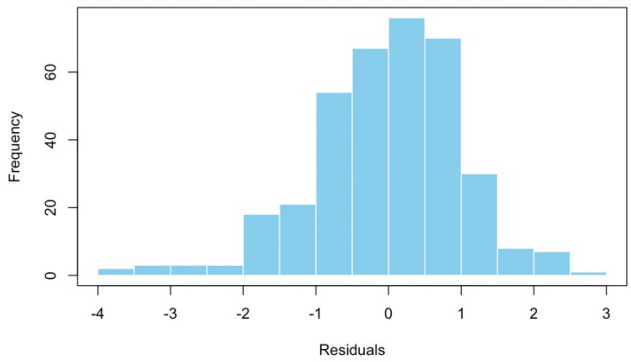


Windspeed casual rentals

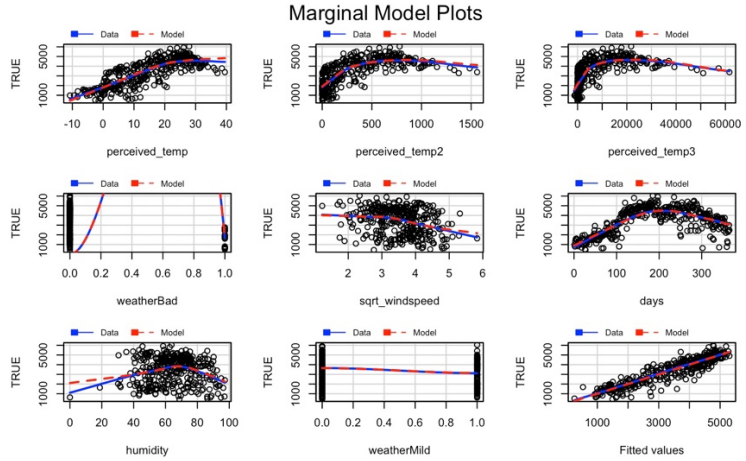
Appendix.8: Full regression output for OLS model for total rentals

	Dependent variable:
	total_rentals
perceived_temp	84.762 (63.390, 106.134)
	t = 7.773
	p = 0.000
perceived_temp2	5.694 (4.065, 7.322)
	t = 6.854
	p = 0.000
perceived_temp3	-0.187 (-0.223, -0.151)
	t = -10.244
	p = 0.000
weatherBad	-700.269 (-1,284.893, -115.645)
	t = -2.348
	p = 0.020
sqrt_windspeed	-247.059 (-333.314, -160.803)
	t = -5.614
	p = 0.00000
days	3.032 (2.385, 3.679)
	t = 9.186
	p = 0.000
humidity	-13.923 (-19.337, -8.509)
	t = -5.040
	p = 0.00000
weatherMild	-218.592 (-371.064, -66.120)
	t = -2.810
	p = 0.006
perceived_temp:weatherBad	-72.547 (-112.692, -32.401)
	t = -3.542
	p = 0.0005
Constant	3,163.185 (2,660.635, 3,665.735)
	t = 12.337
	p = 0.000
Observations	363
R^2	0.840
Adjusted R^2	0.836
Residual Std. Error	555.196 (df = 353)
F Statistic	206.222^{***} (df = 9; 353)
Note:	*p<0.1; **p<0.05; ***p<0.01

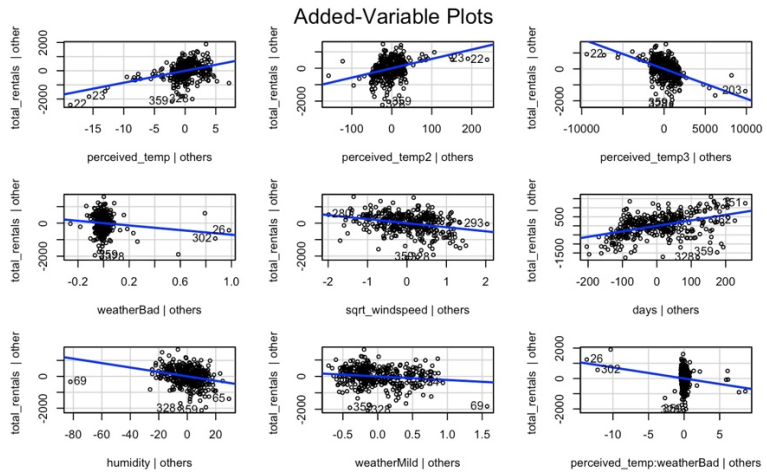
Appendix.9: Histogram of standardized residuals for OLS model for total rentals



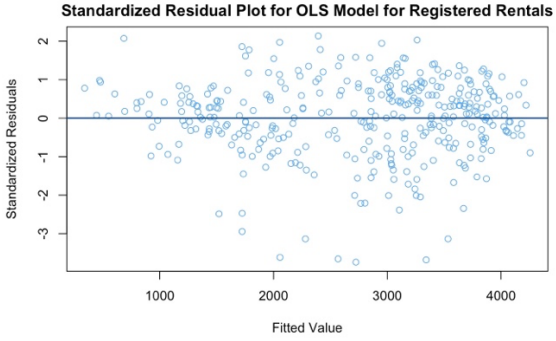
Appendix.10: Marginal model plots for OLS model for total rentals

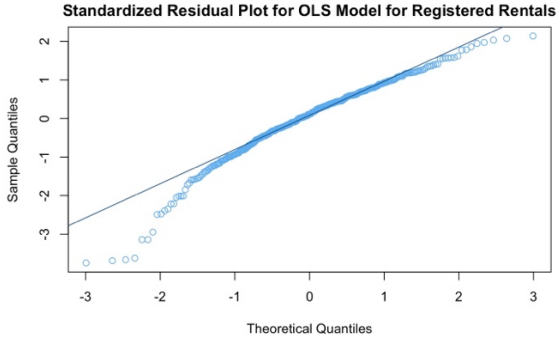


Appendix.11: Added-variable plots for OLS model for total rentals



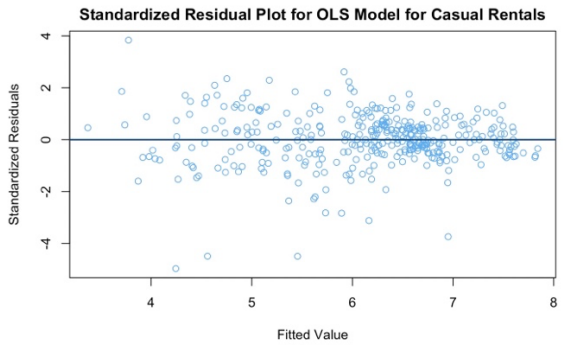
Appendix.12: Full output/standardized residual plot/normal QQ plot for OLS model for registered rentals

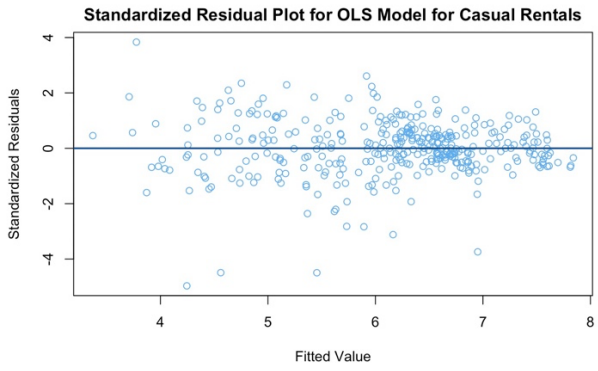




	Dependent variable:
	registered
perceived_temp	49.745 (32.402, 67.087)
	t = 5.622
	p = 0.00000
perceived_temp2	4.172 (2.850, 5.493)
	t = 6.187
	p = 0.000
perceived_temp3	-0.132 (-0.161, -0.103)
	t = -8.890
	p = 0.000
days	3.258 (2.733, 3.783)
	t = 12.162
	p = 0.000
weatherBad	-728.954 (-1,203.446, -254.462)
	t = -3.011
	p = 0.003
humidity	-9.543 (-13.939, -5.146)
	t = -4.254
	p = 0.00003
sqrt_windspeed	-156.117 (-226.109, -86.125)
	t = -4.372
	p = 0.00002
weatherMild	-142.992 (-267.250, -18.734)
	t = -2.255
	p = 0.025
workingday1	690.643 (589.488, 791.799)
	t = 13.382
	p = 0.000
perceived_temp:weatherBad	-45.964 (-78.575, -13.353)
	t = -2.762
	p = 0.007
Constant	1,894.354 (1,480.510, 2,308.198)
	t = 8.972
	p = 0.000
Observations	363
R^2	0.822
Adjusted R^2	0.817
Residual Std. Error	450.509 (df = 352)
F Statistic	162.943^{***} (df = 10; 352)
Note:	*p<0.1; **p<0.05; ***p<0.01

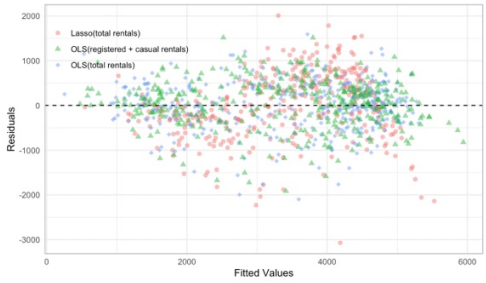
Appendix.13: Full output/standardized residual plot/normal QQ plot for OLS model for casual rentals



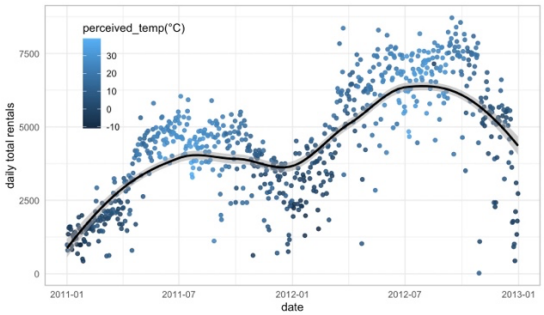


	Dependent variable:
	log(casual)
perceived_temp	0.135 (0.121, 0.150)
	t = 17.955
	p = 0.000
perceived_temp2	-0.002 (-0.003, -0.0005)
	t = -2.715
	p = 0.007
perceived_temp3	-0.00003 (-0.0001, 0.00000)
	t = -1.888
	p = 0.060
weatherBad	-0.719 (-1.159, -0.280)
	t = -3.207
	p = 0.002
humidity	-0.011 (-0.015, -0.007)
	t = -5.207
	p = 0.00000
sqrt_windspeed	-0.122 (-0.185, -0.059)
	t = -3.795
	p = 0.0002
weatherMild	-0.199 (-0.313, -0.085)
	t = -3.414
	p = 0.001
workingday1	-0.870 (-0.963, -0.776)
	t = -18.229
	p = 0.000
perceived_temp:weatherBad	-0.028 (-0.059, 0.002)
	t = -1.847
	p = 0.066
Constant	6.714 (6.343, 7.085)
	t = 35.448
	p = 0.000
Observations	363
R^2	0.840
Adjusted R^2	0.835
Residual Std. Error	0.418 (df = 353)
F Statistic	205.192^{***} (df = 9; 353)
Note:	*p<0.1; **p<0.05; ***p<0.01

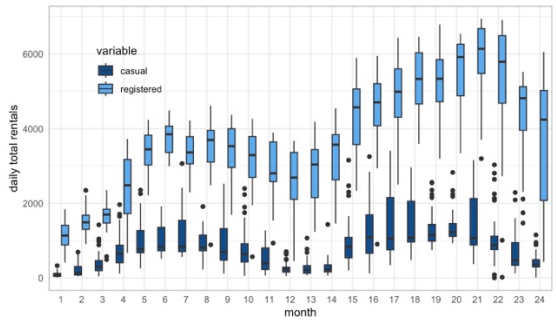
Appendix.14: Training residual plot



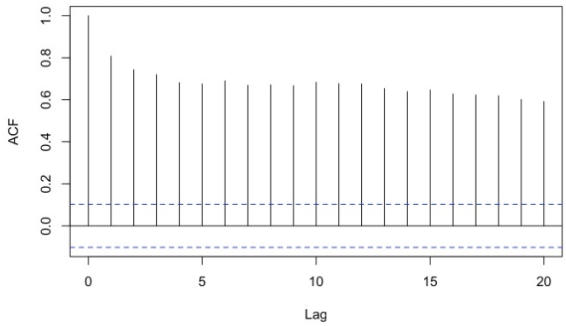
Appendix.15: Daily total rentals from 2011 to 2012



Appendix.16: daily bike rentals against month by user types (2011)



Appendix.17: Autocorrelation plot for total rentals in 2011



Appendix.18 Part of our code:

```
#formatting numerical variables
bikedata$temp = as.numeric(bikedata$temp)
bikedata$perceived_temp = as.numeric(bikedata$atemp)
bikedata$atemp = NULL
bikedata$total_rentals = as.numeric(bikedata$cnt)
bikedata$humidity = as.numeric(bikedata$hum)
bikedata$hum = NULL
bikedata$windspeed = as.numeric(bikedata$windspeed)
bikedata$days = as.numeric(bikedata$instant)
bikedata$instant = NULL
bikedata$total_rentals = bikedata$cnt
bikedata$cnt = NULL

#formatting categorical variables
bikedata$holiday = as.factor(bikedata$holiday)
bikedata$workingday = as.factor(bikedata$workingday)
bikedata$weekday = as.factor(bikedata$weekday)
bikedata$month = as.factor(bikedata$mnth)
bikedata$season = as.factor(bikedata$season)
bikedata$weather = as.factor(bikedata$weathersit)

weekday_names <- c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")
# Transform the DayNumber column to weekday names
bikedata$weekday <- factor(bikedata$weekday, levels = 0:6, labels = weekday_names)

season_names <- c("Spring", "Summer", "Autumn", "Winter")
# Transform the DayNumber column to weekday names
bikedata$season <- factor(bikedata$season, levels = 1:4, labels = season_names)

weather_names <- c("Good", "Mild", "Bad")
# Transform the DayNumber column to weekday names
bikedata$weather <- factor(bikedata$weather, levels = 1:3, labels = weather_names)
```

```{r}
t_min = -8
t_max = 39
# Transform the normalized temperatures back to real temperatures
bikedata$temp <- bikedata$temp * (t_max - t_min) + t_min

t_min = -16
t_max = 50
# Transform the normalized feels like temperatures back to real temperatures
bikedata$perceived_temp <- bikedata$perceived_temp * (t_max - t_min) + t_min

# Transform the normalized windspeed back to real windspeed
bikedata$windspeed <- bikedata$windspeed*67

## Transform the normalized humidity back to real humidity
bikedata$humidity <- bikedata$humidity*100
```

```{r}
bikedata %>% filter(yr == 0) %>% ggplot(aes(x = perceived_temp, y = total_rentals, color = weather)) + geom_point() + theme_minimal() + labs(title = "", x = "perceived_temp(°C)", y = "daily total rentals") +
scale_color_manual(values = c("skyblue2","dodgerblue3","black"))+ theme_minimal() + theme(panel.border = element_rect(colour = "grey", fill=NA, size=1)) + theme(legend.position = c(0.18,0.75))
```

```{r}
bikedata %>% ggplot() + geom_boxplot(aes(x = weather, y = total_rentals),fill = 'steelblue2') + theme_minimal()+ theme(axis.title.x = element_blank())+ theme(panel.border = element_rect(colour = "grey",
fill=NA, size=1)) + ylab('Bike User Count') + ggtitle('Boxplot of Total Rentals Per Day against Weather')
```

```{r}
ggplot(bikedata %>% filter(yr == 0), aes(x = dteday, y = total_rentals)) + geom_point(aes(color = perceived_temp),alpha = 0.9) + geom_smooth(color = 'black',alpha = 0.5) + theme_minimal() + labs(title = '', y =
'daily total rentals', x = 'date', color = 'perceived_temp(°C)' ,shape = 'weather') + theme(panel.border = element_rect(colour = "grey", fill=NA, size=1)) + theme(legend.position = c(0.18,0.75))
```
```{r}
bikedata %>% filter(yr == 0) %>% ggplot() + geom_boxplot(aes(x = season, y = total_rentals),fill = 'steelblue2') + theme_minimal()+ theme(axis.title.x = element_blank())+ theme(panel.border =
element_rect(colour = "grey", fill=NA, size=1)) + ylab('daily total rentals')
```

```{r}
anova_result <- aov(perceived_temp ~ season, data = bikedata %>% filter(yr == 0))
summary(anova_result)
posthoc_result <- TukeyHSD(anova_result)
posthoc_result
```
```{r}
anova_result <- aov(total_rentals ~ weather, data = bikedata %>% filter(yr == 0))
summary(anova_result)
```
```{r}
bikedata %>% filter(yr == 0) %>% ggplot() + geom_boxplot(aes(x = season, y = perceived_temp),fill = 'steelblue2') + theme_minimal()+ theme(axis.title.x = element_blank())+ theme(panel.border =
element_rect(colour = "grey", fill=NA, size=1)) + ylab('perceived temperature')
```

```{r}
data_weekly <- bikedata %>% filter(yr == 0)
data_weekly <- data_weekly[,c('casual','registered','month')]

data_long <- gather(data_weekly, key = "variable", value = "value", -month)

ggplot(data_long, aes(month, value)) + geom_boxplot(aes(fill = variable)) + xlab("month") + ylab("daily total rentals") + theme_minimal() + scale_fill_manual(values = c("registered" = "steelblue2", "casual" =
"dodgerblue4")) + ggtitle('')+ theme(panel.border = element_rect(colour = "grey", fill=NA, size=1)) + theme(legend.position = c(0.16,0.77))
```

```{r}
data_weekly <- bikedata %>% filter(yr == 0)
data_weekly <- data_weekly[,c('casual','registered','weekday')]

data_long <- gather(data_weekly, key = "variable", value = "value", -weekday)
```

```
ggplot(data_long, aes(weekday, value)) + geom_boxplot(aes(fill = variable)) + xlab("weekday") + ylab("daily total rentals") + theme_minimal() + scale_fill_manual(values = c("registered" = "steelblue2", "casual" = "dodgerblue4")) + ggtitle('')+ theme(panel.border = element_rect(colour = "grey", fill=NA, size=1))+ theme(legend.position = c(0.14,0.85))
```

```{r }
data_weekly <- bikedata %>% filter(yr == 0)
data_weekly <- data_weekly[,c('casual','registered','season')]

data_long <- gather(data_weekly, key = "variable", value = "value", -season)

ggplot(data_long, aes(season, value)) + geom_boxplot(aes(fill = variable)) + xlab("weekday") + ylab("daily total rentals") + theme_minimal() + scale_fill_manual(values = c("registered" = "steelblue2", "casual" = "dodgerblue4")) + ggtitle('')+ theme(panel.border = element_rect(colour = "grey", fill=NA, size=1))+ theme(legend.position = c(0.14,0.85))
```

```{r }
corr_data <- bikedata %>% filter(yr == 0) %>% select(total_rentals,registered,casual,perceived_temp,temp,days, windspeed,humidity,weather,workingday)

custom_upper <- function(data, mapping, ...){
  p <- ggplot() + theme_void()
 return(p)
}
custom_diag <- function(data, mapping, ...){
 p <- ggplot(data = data, mapping = mapping) + geom_boxplot(outlier.shape = NA,alpha = 0.5,size = 0.3)
 return(p)
}

plot_stat <-
 theme_minimal() +
 theme( axis.line=element_blank(),
      panel.background = element_blank(),
      plot.title = element_text(size=10, color="grey20"),
      panel.grid.minor = element_blank() ,
      axis.title = element_text(size=10, color="grey20"),
      panel.grid.major = element_blank(),
      legend.key = element_blank() ,
      legend.title = element_text(size=6, color="grey30") ,
      legend.text = element_text(size=6, color="grey30") )

plot <- ggpairs(corr_data,
            lower = list(combo = custom_diag,
                  continuous = wrap("cor", size=5)),
            upper=list(combo=wrap("points", alpha=0.1, size=0.01 ),
                  continuous=wrap("points", alpha=0.1, size=0.01),
                  discrete=wrap("points", alpha=0.1, size=0.01) ),
            diag = list(continuous = wrap("densityDiag", alpha = 0.5)), progress=F) +plot_stat+
 theme(text= element_text(size = 11),
 axis.text.x = element_text(size=8,angle = 90),
 axis.text.y = element_text(size=8,angle = 0),
 axis.ticks = element_blank(),
 axis.title.y.right = element_text(size = 8, angle = 45, margin = margin(t = 0, r = 0, b = 0, l = 2)))
plot
```

```{r }
#Box Cox Transformation
boxcox(windspeed~1, data = bikedata %>% filter(yr == 0), lambda = seq(-1, 2, by = 0.1))

#boxcox(perceived_temp+20~1, data = bikedata %>% filter(yr == 0), lambda = seq(-1, 2, by = 0.1))

#boxcox(humidity+20~1, data = bikedata %>% filter(yr == 0), lambda = seq(-1, 2, by = 0.1))

#boxcox(total_rentals~1, data = bikedata %>% filter(yr == 0), lambda = seq(-1, 2, by = 0.1))

#boxcox(registered~1, data = bikedata %>% filter(yr == 0), lambda = seq(-1, 2, by = 0.1))

boxcox(casual~1, data = bikedata %>% filter(yr == 0), lambda = seq(-1, 2, by = 0.1))
```

```{r }
data_for_training <- bikedata[,c('yr','registered','casual','total_rentals','weather','perceived_temp','humidity','windspeed','weekday',"workingday","days")]

data_for_training <- data.frame(model.matrix( ~ .-1, data_for_training))
data_for_training$weatherGood = NULL

Train_data <- data_for_training %>% filter(yr == 0)
Train_data$perceived_temp2 = Train_data$perceived_temp^2
Train_data$perceived_temp3 = Train_data$perceived_temp^3
Train_data$perceived_tempBad = Train_data$perceived_temp*Train_data$weatherBad
Train_data$perceived_tempMild = Train_data$perceived_temp*Train_data$weatherMild

Train_data$sqrt_windspeed = sqrt(Train_data$windspeed)
Train_data$windspeed = NULL
X_train <- Train_data[,5:ncol(Train_data)]
y_train <- Train_data$total_rentals

Test_data <- data_for_training %>% filter(yr == 1)
Test_data$perceived_temp2 = Test_data$perceived_temp^2
Test_data$perceived_temp3 = Test_data$perceived_temp^3
Test_data$perceived_tempBad = Test_data$perceived_temp*Test_data$weatherBad
Test_data$perceived_tempMild = Test_data$perceived_temp*Test_data$weatherMild
Test_data$sqrt_windspeed = sqrt(Test_data$windspeed)
Test_data$windspeed = NULL
Test_data %>% slice(-303)

X_test <- Test_data[,5:ncol(Test_data)]
y_test <- Test_data$total_rentals
```

```{r }
time_series_cv_lasso <- function(x, y, alpha = 1, n_folds = 7) {
 n <- nrow(x)
 fold_size <- floor(n / n_folds)
```

```
  lambda_result <- list()
  lambda_list <- cv.glmnet(as.matrix(x),y,nlambda = 100)$lambda
  for (lambda in lambda_list){
    result <- list()
    for (fold in seq_len(n_folds)) {
      train_end <- (fold) * fold_size
      x_train <- x[1:train_end, ]
      y_train <- y[1:train_end]
      x_test <- x[train_end:nrow(x),]
      y_test <- y[train_end:nrow(x)]
      cv_fit <- cv.glmnet(x_train,y_train)
      lasso_model <- glmnet(x_train, y_train, alpha = 1, lambda = lambda)
      preds = predict(lasso_model,x_test)
      mse <- mean((preds - y_test)**2)
      result <- append(result, mse[1])
    }
    lambda_result <- append(lambda_result, mean(unlist(result)))
  }
  return(data.frame(lambda_list,unlist(lambda_result)))
}
result <- time_series_cv_lasso(as.matrix(X_train),as.matrix(y_train))
```

```{r}
lasso_test <- glmnet(X_train,y_train, alpha = 1,family = "gaussian", standardize = TRUE)
plot_glmnet(lasso_test,label = 20)

lasso_model<- glmnet(X_train,y_train, alpha = 1, lambda =  103, family = "gaussian", standardize = TRUE)
res <- data.frame(as.matrix(coef(lasso_model, lasso_model$lambda.1se))) %>% filter(s0 != 0)
write.csv(res, "/Users/raywu/Desktop/output.csv", row.names = TRUE)
```

```{r }
preds <- predict(lasso_model, newx = as.matrix(X_train))
residuals <- y_train - preds
plot(preds, residuals,
     xlab = "Predicted Values",
     ylab = "Residuals",
     main = "",
     col = "steelblue2")
abline(h = 0, col = "dodgerblue4", lty = 2)

hist(residuals,
     main = "",
     xlab = "Residuals",
     col = "skyblue",
     border = "white")
box()
```

```{r warning = FALSE}
model_2 <- lm(total_rentals ~ perceived_temp+ perceived_temp2 + perceived_temp3  + perceived_temp*weatherBad  + sqrt_windspeed + days + humidity + weatherMild + weatherBad, data = Train_data %>%
slice(-c(239,358)))

stargazer(model_2, type = "text", single.row = TRUE, header = FALSE, digits = 3,report = "vcstp",ci = TRUE,out = "stargazer_output1.html")

avPlot(model_2, "perceived_temp3",main = ",col = 'steelblue')
avPlot(model_2, "perceived_temp2",main = ",col = 'steelblue')
```

```{r }
avPlots(model_2)
mmps(model_2)
```

```{r warning=FALSE}
standardized_residuals <- rstandard(model_2)
hist(standardized_residuals,
     main = "",
     xlab = "Residuals",
     col = "skyblue",
     border = "white")
box()
sqrt_standardized_residuals <- sqrt(abs(standardized_residuals))
fitted_values <- fitted(model_2)
#fitted_values <- data.frame(fitted_values) %>% slice(-26) %>% slice(-89)
#standardized_residuals <- data.frame(standardized_residuals) %>% slice(-26) %>% slice(-89)
qqnorm(standardized_residuals, main = "", col = 'steelblue2')
qqline(standardized_residuals, col = 'dodgerblue4')

season <- (bikedata %>% filter(yr == 0) %>% slice(-239) %>% slice(-357))$season
data <- data.frame(fit = fitted_values, resid = standardized_residuals, season = season)
ggplot(data, aes(fit,resid)) + geom_point(aes(color = season))+theme_minimal()+ theme(legend.position = c(0.15,0.23))+ theme(panel.border = element_rect(colour = "grey", fill=NA, size=1))+
geom_abline(intercept = 0, slope = 0, color = "black")+ylab('Standardized Residuals') + xlab('Fitted Value')
```

```{r warning = FALSE}
#seasonSummer +seasonWinter
model_3 <- lm(registered ~ perceived_temp + perceived_temp2 + perceived_temp3 + days + perceived_temp*weatherBad + humidity + sqrt_windspeed + weatherBad + weatherMild + workingday1, data =
Train_data %>% slice(-239) %>% slice(-357))

stargazer(model_3, type = "text", single.row = TRUE, header = FALSE, digits = 3,report = "vcstp",ci = TRUE,out = "stargazer_output2.html")
```

```{r }
model_4 <- lm(log(casual) ~ perceived_temp + perceived_temp2 + perceived_temp3 + perceived_temp*weatherBad + humidity + sqrt_windspeed+ weatherBad + weatherMild+ workingday1, data =
Train_data %>% slice(-239) %>% slice(-357))


stargazer(model_4, type = "text", single.row = TRUE, header = FALSE,t.auto = TRUE, digits = 3,report = "vcstp",out = "stargazer_output3.html",ci = TRUE)
```

```{r warning=FALSE}
standardized_residuals <- rstandard(model_3)
sqrt_standardized_residuals <- sqrt(abs(standardized_residuals))
fitted_values <- fitted(model_3)
#fitted_values <- data.frame(fitted_values) %>% slice(-26) %>% slice(-89)
#standardized_residuals <- data.frame(standardized_residuals) %>% slice(-26) %>% slice(-89)
qqnorm(standardized_residuals, main = "Standardized Residual Plot for OLS Model for Registered Rentals", col = 'steelblue2')
qqline(standardized_residuals, col = 'dodgerblue4')
plot(fitted_values, standardized_residuals,label = "Standardized Residuals", main = "Standardized Residual Plot for OLS Model for Registered Rentals",col = 'steelblue2',xlab = 'Fitted Value', ylab = 'Standardized Residuals')
abline(h = 0, col = 'dodgerblue4', lwd = 2)
```

```{r warning=FALSE}
standardized_residuals <- rstandard(model_4)
sqrt_standardized_residuals <- sqrt(abs(standardized_residuals))
fitted_values <- fitted(model_4)
#fitted_values <- data.frame(fitted_values) %>% slice(-26) %>% slice(-89)
#standardized_residuals <- data.frame(standardized_residuals) %>% slice(-26) %>% slice(-89)
qqnorm(standardized_residuals, main = "Normal QQ Plot for OLS Model for Casual Rentals", col = 'steelblue2')
qqline(standardized_residuals, col = 'dodgerblue4')
plot(fitted_values, standardized_residuals,label = "Standardized Residuals", main = "Standardized Residual Plot for OLS Model for Casual Rentals",col = 'steelblue2',xlab = 'Fitted Value', ylab = 'Standardized Residuals')
abline(h = 0, col = 'dodgerblue4', lwd = 2)
```

```{r }
preds_ols <- predict(model_2,Test_data)
preds_combined <- predict(model_3,Test_data) + exp(predict(model_4,Test_data))
preds_lasso <- predict(lasso_model,as.matrix(X_test))
test <- bikedata %>% filter(yr == 1)
test$preds_lasso <- preds_lasso
test <- test %>% mutate(preds_lasso = replace(preds_lasso, preds_lasso < 0, 0))
test$preds_ols <- preds_ols
test$preds_combined <- preds_combined

test %>% group_by(week) %>% summarise(total_rentals = mean(total_rentals),preds_ols = mean(preds_ols),preds_combined = mean(preds_combined),preds_lasso = mean(preds_lasso), total_rentals = mean(total_rentals)) %>% ggplot() + geom_line(aes(x = week, y = total_rentals, color = 'observed value')) + geom_point(aes(x = week, y = total_rentals,color = 'observed value')) + geom_line(aes(x = week, y = preds_ols,color = 'OLS(total rentals)')) + geom_point(aes(x = week, y = preds_ols,color = 'OLS(total rentals)')) + geom_point(aes(x = week, y = preds_lasso,color = 'Lasso(total rentals)')) + geom_line(aes(x = week, y = preds_lasso,color = 'Lasso(total rentals)')) + geom_point(aes(x = week, y = preds_combined,color = 'OLS(registered + casual rentals)')) + geom_line(aes(x = week, y = preds_combined,color = 'OLS(registered + casual rentals)')) + scale_color_manual(values = c("observed value" = "black", "Lasso(total rentals)" = "dodgerblue4","OLS(total rentals)" = "steelblue1","OLS(registered + casual rentals)" = "grey")) + theme_minimal() + theme(panel.border = element_rect(colour = "grey", fill=NA, size=1)) + ylab("daily total rentals") + xlab('weeks') + labs(title = '', color = '')+ theme(plot.title = element_text(family = "Arial", size = 12),legend.position = c(0.55, 0.3))
```

```{r }
fitted_values1 <- predict(lasso_model, as.matrix(X_test))
fitted_values2 <- predict(model_2, X_test)
fitted_values3 <- predict(model_3, X_test) + exp(predict(model_4, X_test))

# Assume residuals1, residuals2, and residuals3 have different lengths
residuals1 <- y_test - fitted_values1
residuals2 <- y_test - fitted_values2
residuals3 <- y_test - fitted_values3

# Create data frames for each model's residuals and fitted values
df1 <- data.frame(Fitted = fitted_values1, Residuals = residuals1, Model = "Lasso(total rentals)")
colnames(df1) <- c("Fitted", "Residuals", "Model")
df2 <- data.frame(Fitted = fitted_values2, Residuals = residuals2, Model = "OLS(total rentals)")
df3 <- data.frame(Fitted = fitted_values3, Residuals = residuals3, Model = "OLS(registered + casual rentals)")

# Combine the data frames into a single long-format data frame
residuals_long <- rbind(df1, df2, df3)

# Create the residual plot for three models with different colors
ggplot(residuals_long, aes(x = Fitted, y = Residuals, color = Model, shape = Model)) +
  geom_point(alpha = 0.5, size = 2)+geom_hline(yintercept = 0, linetype = "dashed", color = "black", size = 0.5) + theme_minimal() + labs(title = "", x = "Fitted Values", y = "Residuals",color = ",shape = ")+theme(panel.border = element_rect(colour = "grey", fill=NA, size=1)) + theme(legend.position = c(0.17,0.85))+ scale_shape_manual(values = c(16, 17, 18))
```

```{r }
fitted_values1 <- predict(lasso_model, as.matrix(X_train))
fitted_values2 <- model_2$fitted.values
fitted_values3 <- model_3$fitted.values + exp(model_4$fitted.values)

# Assume residuals1, residuals2, and residuals3 have different lengths
residuals1 <- y_train - fitted_values1
residuals2 <- residuals(model_2)
residuals3 <- (Train_data %>% slice(-c(239,358)))$total_rentals - fitted_values3

# Create data frames for each model's residuals and fitted values
df1 <- data.frame(Fitted = fitted_values1, Residuals = residuals1, Model = "Lasso(total rentals)")
colnames(df1) <- c("Fitted", "Residuals", "Model")
df2 <- data.frame(Fitted = fitted_values2, Residuals = residuals2, Model = "OLS(total rentals)")
df3 <- data.frame(Fitted = fitted_values3, Residuals = residuals3, Model = "OLS(registered + casual rentals)")

# Combine the data frames into a single long-format data frame
residuals_long <- rbind(df1, df2, df3)

# Create the residual plot for three models with different colors
ggplot(residuals_long, aes(x = Fitted, y = Residuals, color = Model, shape = Model)) +
  geom_point(alpha = 0.5, size = 2)+geom_hline(yintercept = 0, linetype = "dashed", color = "black", size = 0.5) + theme_minimal() + labs(title = "", x = "Fitted Values", y = "Residuals",color = ",shape = ")+theme(panel.border = element_rect(colour = "grey", fill=NA, size=1)) + theme(legend.position = c(0.17,0.85))+ scale_shape_manual(values = c(16, 17, 18))
```

```{r }
# Create data frames for each model's true values and fitted values
df1 <- data.frame(True = y_test, Fitted = fitted_values1, Model = "Lasso(total rentals)")
colnames(df1) <- c('True','Fitted','Model')
df2 <- data.frame(True = y_test, Fitted = fitted_values2, Model = "OLS(total rentals)")
```

```
df3 <- data.frame(True = y_test, Fitted = fitted_values3, Model = "OLS(registered + casual rentals)")

# Combine the data frames into a single long-format data frame
values_long <- rbind(df1, df2, df3)

# Create the fitted vs. true plot for three models with different colors
ggplot(values_long, aes(x = Fitted, y = True, color = Model, shape = Model)) +
  geom_point(alpha = 0.5, size = 2) +
  geom_abline(intercept = 0, slope = 1, linetype = "dashed", color = "black", size = 0.5) +
  theme_minimal() +
  labs(title = "", x = "True Values", y = "Fitted Values",color = ",shape = '') +
  scale_shape_manual(values = c(16, 17, 18))+theme(panel.border = element_rect(colour = "grey", fill=NA, size=1)) + theme(legend.position = c(0.16,0.77))+ scale_shape_manual(values = c(16, 17, 18))
```

```{r }
ggplot(bikedata, aes(x = dteday, y = total_rentals)) + geom_point(aes(color = perceived_temp),alpha = 0.9) + geom_smooth(color = 'black',alpha = 0.5) + theme_minimal() + labs(title = '', y = 'daily total rentals',
x = 'date', color = 'perceived_temp(°C)' ,shape = 'weather') + theme(panel.border = element_rect(colour = "grey", fill=NA, size=1)) + theme(legend.position = c(0.18,0.75))
```

```{r }
data_weekly_1 <- bikedata %>% filter(yr == 0)
data_weekly_2 <- bikedata %>% filter(yr == 1)
data_weekly_2$mnth <- data_weekly_2$mnth + 12
data_weekly <- rbind(data_weekly_1,data_weekly_2)
data_weekly_ <- data_weekly[,c('casual','registered','mnth')]

data_long <- gather(data_weekly, key = "variable", value = "value", -mnth)

ggplot(data_long, aes(as.factor(mnth), value)) + geom_boxplot(aes(fill = variable)) + xlab("month") + ylab("daily total rentals") + theme_minimal() + scale_fill_manual(values = c("registered" = "steelblue2",
"casual" = "dodgerblue4")) + ggtitle('')+ theme(panel.border = element_rect(colour = "grey", fill=NA, size=1)) + theme(legend.position = c(0.16,0.77))
```

```{r }
#validation
preds_combined <- predict(model_3,X_test) + exp(predict(model_4,X_test))
preds_lasso <- predict(lasso_model,as.matrix(X_test))
preds_ols <- predict(model_2,X_test)
#root mse
print(sqrt(mean((y_test - preds_lasso)**2)))
print(sqrt(mean((y_test - preds_ols)**2)))
print(sqrt(mean((y_test - preds_combined)**2)))
#relative mse
print(mean((y_test - preds_lasso)**2)/mean(y_test**2))
print(mean((y_test - preds_ols)**2)/mean(y_test**2))
print(mean((y_test - preds_combined)**2)/mean(y_test**2))
```

```{r }
acf(y_train, lag.max = 20, main = "Autocorrelation Function (ACF)")
```