ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS
DEPARTMENT OF MANAGEMENT SCIENCE & TECHNOLOGY
MSC BUSINESS ANALYTICS

Bike Sharing Systems

Statistics
For Business
Analytics I

Tutor: Ntzoufras Ioannis
Student: Despotis Spyridon
Dataset Number: 29
Student ID: P2822111

2021-22

Table of Contents

1. Introduction

Nowadays, there is great interest from local governments to invest in bike sharing systems (BSS) due to their numerous benefits on traffic, environment, and health. A bike sharing system is a shared micromobility service that allows users to rent a bike and drop it off at any automated bicycle station within the network. Currently, there are over 500 bike-sharing programs around the world which are composed of over 500 thousands bicycles. Therefore, the data that is generated by such systems is attractive for research in order to export useful insights. For this assignment we analysed a bike-sharing rental dataset from 2011 and 2012 from the Capital Bikeshare program in Washington, D.C, USA. In this assignment, our main objective was to find out the determining factor that plays a significant role in bike rental and to develop statistical models in order to predict rentals for each hour based on the dataset in use. The statistical analysis is conducted via the R computer language (Appendix page 23).

1.1 Data Cleaning & Transformations

Our preliminary training dataset "bike_29" contained 18 variables (columns) and 1500 observations (rows). Before we started our statistical analysis, we had to examine our dataset. We began by looking for missing values and then we proceeded by performing appropriate data type conversions and exclusions. First, by applying the function "is.na" to our dataset, we found that we did not have any missing values. Then, we proceeded by excluding the "X" (record index) and "Instant" (record index) variables from our dataset, since they would not be used later in our analysis. We also excluded the variable "dteday" (hourly date & timestamp) since the information contained within it was already available in "HourF" and "Year" in our dataset. Additionally, we removed the variables "Registered" (the count of registered user rentals, across all stations) and "Casual" (count of non-registered user rentals, across all stations) since they sum up to the variable "Count" (number of total rentals). Then, we converted the variables "holiday", "weekday", "workingday", "weather", "year", "month", "season" and "hourF" from continuous to factor and the variables "count", "temp", "humidity", "windspeed" and "atemp" from int to numeric in order to be the same data type.

Variables	***Description***
count	Number of total rentals
hourF	Morning: 7am-12am, Afternoon:12am-17pm, Evening: 17pm-22pm, Night: 22pm-7am
year	0: 2011, 1:2012
month	Month (1 to 12)
workingday	Day of the week
weekday	If day is neither weekend nor holiday is 1, otherwise is 0
holiday	Weather day is holiday or not
weather	1: Clear, Few clouds, Partly cloudy, Partly cloudy, 2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist, 3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds, 4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
temp	Normalized temperature in Celsius. The values are divided to 41 (max)
atemp	"Feels like" temperature in Celsius The values are divided to 50 (max)
humidity	Normalized humidity. The values are divided to 100 (max)
windspeed	Normalized wind speed. The values are divided to 67 (max)
season	1 = spring, 2 = summer, 3 = fall, 4 = winter

Table 1.1-1: Final Dataset Variables Explanation

2. Descriptive and Exploratory Data Analysis

After the process of data cleaning and transformations that we saw in Section One, let’s have a look to our data from the output with function str() in Table 2-1. At first glance we can see that we have 13 variables (columns), 1500 observations (rows), and the levels of each factor variable.

Table 2-1: Display the Structure of Final Dataset with Function str()

data.frame':	*1500 obs. of 13 variables:*
$ count	num 416 176 220 359 175 28 283 49 30 187 …
$ hourF	Factor w/ 4 levels "Afternoon","Evening",..: 2 1 3 3 1 4 3 4 4 1
$ year	Factor w/ 2 levels "0","1": 1 2 2 2 1 1 2 2 1 1 …
$ month	Factor w/ 12 levels "1","2","3","4",..: 8 5 12 10 5 12 8 9 10 12
$ workingday	Factor w/ 2 levels "nonwkday","wkday": 2 2 1 2 2 1 2 2 2 2 …
$ weekday	Factor w/ 7 levels "0","1","2","3",..: 2 2 7 2 3 7 3 6 3 6 ...
$ holiday	Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 …
$ weather	Ord.factor w/ 4 levels "very bad"<"bad"<..: 4 3 4 2 4 3 3 4 4 4
$ temp	num 0.7 0.62 0.26 0.6 0.9 0.4 0.8 0.52 0.44 0.38 ...
$ humidity	num 0.3 0.73 0.75 0.69 0.37 0.62 0.55 0.72 0.77 0.46 ...
$ windspeed	num 0.284 0.224 0.104 0.388 0.164 ...
$ season	Factor w/ 4 levels "spring","summer",.. 3 2 4 4 2 1 3 3 4 4 …
$ atemp	num 0.636 0.591 0.273 0.591 0.818 …

Moreover, in Table 2-2 we can see the summary statistics (function “summary”()) for our numeric variables. Our response variable is “count” and all other variables are predictors.

	count	**temp**	**humidity**	**windspeed**	**atemp**
Min.:	1	0.0200	0.1500	0.0000	0.0455
1st Qu.:	42	0.3400	0.4700	0.1045	0.3333
Median :	154	0.5000	0.6300	0.1940	0.4848
Mean :	198	0.4964	0.6234	0.1908	0.4756
3rd Qu.:	291	0.6600	0.7800	0.2537	0.6212
Max. :	967	0.9600	1.0000	0.7164	0.9848

Table 2-2: Summary Statistics for Numeric Variables

We can conclude that the response variable “count” does not appear to be symmetrically distributed since the mean and median are not the same. Also, the symmetry can be confirmed from the histogram of the response variable (Figure 2.1). Moreover, we can see that the minimum amount of total rental bikes, including casual and registered users, is 1 which indicates that at a specific hour within a day only 1 bicycle was being rented. The minimum wind speed is 0.0000s which is often reported as “Calm Wind” and indicates that the wind has no direction. Within the continuous variables: “temp”, “atemp”, “humidity” and “windspeed” the mean and median are close, indicating symmetric distributions. Our findings can be confirmed in Figure 2.2. There is also evidence of some extreme values in the histograms, which could possibly indicate the existence of outliers.

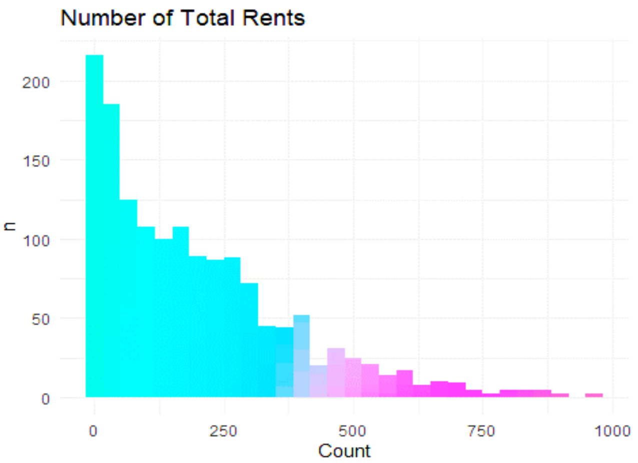

Figure 2-1 Histogram of the Response Variable "count".

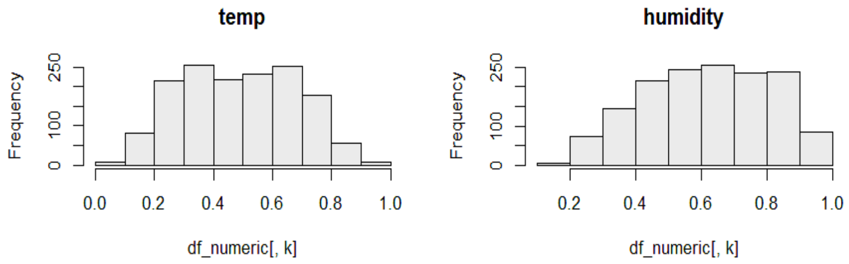

Figure 2-2 Visualising Continuous Variables with Histograms

Visualizing the response with factor variables can give us useful insights about the relationships between the data. From the Figure 2.3 we can see a boxplot comparing the total number of bike rentals (count variable) against the hour. It is evident that the average total bike rentals at night are significant less than the other hours of the day. Also the afternoon seems to be the most preferred period from users since it has the highest average of total bike rentals. Last but not least, the rentals in Morning and Evening have not any significance difference in average rentals.

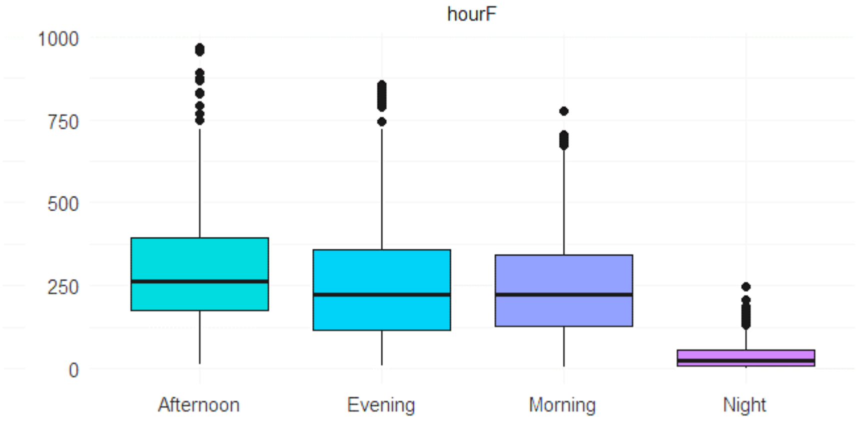

Figure 2-3 Boxplot with Response Variable "count" Against "Hour" Variable

Plotting the weather against the total bike rentals (Figure 2.4), we see that if we have better weather conditions, we will have higher number of bicycle rentals. Therefore, “the very good” weather conditions have the most number of rentals.

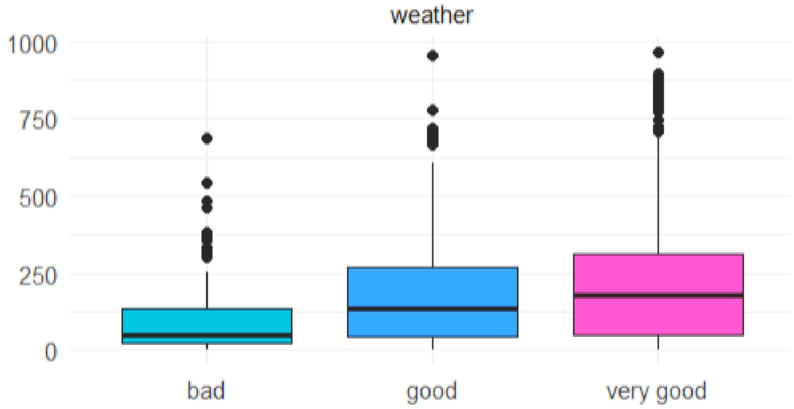

Figure 2-4 Boxplot with the Response Variable "count" Against "weather" Variable

Making a boxplot of the months against the total bike rentals (Figure 2.5), shows that the higher average total rentals are located in summer months, especially in June and July. On the other hand, the lowest average rentals are located in January and February.

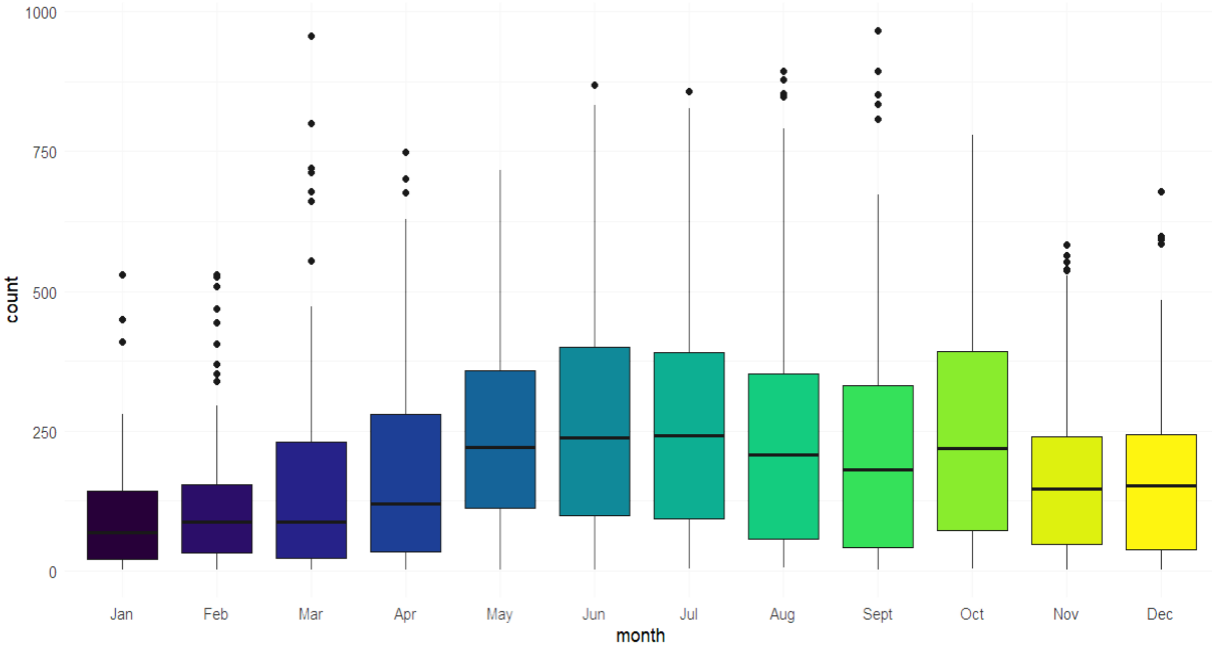

Figure 2-5: Boxplot with the Response Variable "count" Against "weather" Variable

3. Pairwise Comparisons

By using pairwise comparisons we can have a straight forward view of the correlations between our variables. From the Figure 3-1 we can see that there does not seem to have many variables with strong correlation to each other. Specifically, we can see there is high positive correlation between the "temp" and "atemp" predictors which are also moderate positively correlated to the response variable. This means that for higher temperatures or feeling temperature it seems more likely to have more bicycle rentals. There is also moderate negative correlation between count and humidity. The higher humidity has negative impact on the bike rentals.

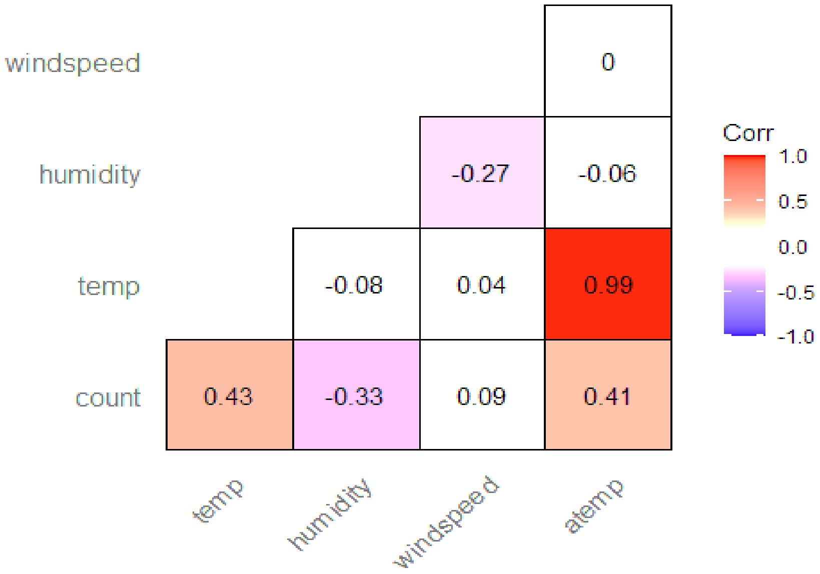

Figure 3-1 Visualising relationship between the numeric variables: Correlations

4. Predictive Descriptive Models

4.1 Implement of Lasso

Lasso regression with k-fold cross validation was applied to select the optimum λ (lambda) and for the variables selection. The tuning parameter λ controls the amount of the coefficient shrinkage. The best λ for the data, can be defined as the λ that minimize the cross-validation prediction error rate (Mean Squared Error, MSE). The next plot (Figure 4-2) shows the MSE vs log(lambda). The vertical lines correspond to lambda.min (1.62401) and to Lambda.1se (12.57412) (Figure 4-1), that is the largest value of lambda such that error is within 1 standard error of the minimum. The latter value was selected, since it leads to more parsimonious models.

lambda	*Output*
lasso1$lambda.min	1.62401
lasso1$lambda.1se	12.57412

Table 4-1 λ 1SE & λ Min values

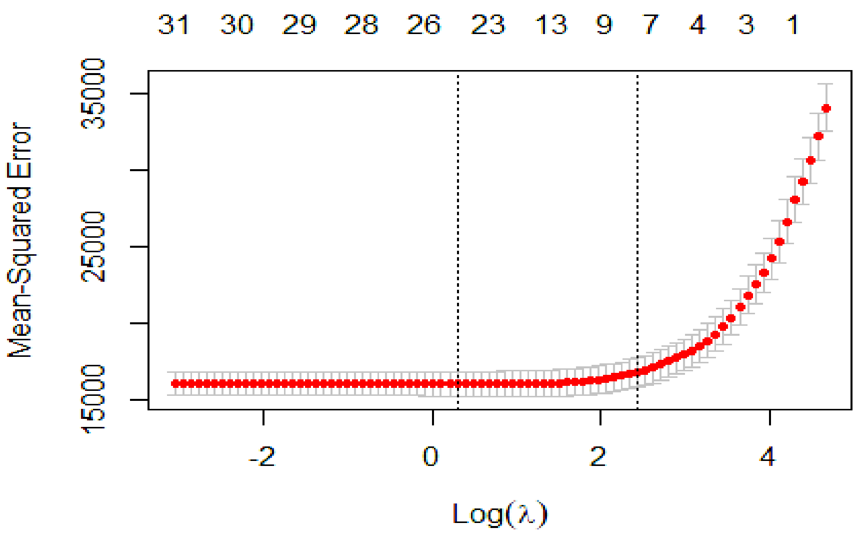

Figure 4-1 Plot with the Mean Squared Error (MSE) vs Log (lambda)

In the next plot (Figure 4-2), each curve corresponds to a variable. It shows the path of its coefficient against the ℓ1-norm of the whole coefficient vector as λ varies. The axis above indicates the number of nonzero coefficients at the current λ, which is the effective degrees of freedom (df) for the lasso. The vertical lines correspond to lambda.min and to lambda.1se.

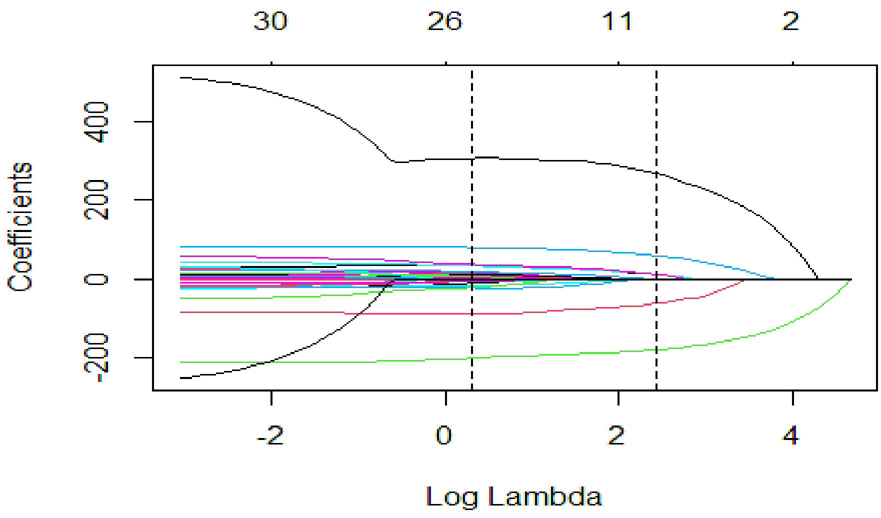

Figure 4-2 Lasso Plot Generated in GLMNET

Next, we can obtain the model coefficients corresponding to lambda.1se (Table 4-2). Thus, according to the LASSO method, the following predictors should be selected: Hour, year, month, weather, season, temp and humidity (the others are shrunk to zero).

coef(lasso1	***s = "lambda.1se")***
## 33 x 1 sparse Matrix of class "dgCMatrix"	
## s1	
(Intercept)	118.38903
hourFEvening	.
hourFMorning	.
hourFNight	-178.79796
year1	60.68456
month2	.
month3	.
month4	.
month5	.
month6	.
month7	.
month8	.
month9	.
month10	11.47975
month11	.
month12	.
workingdaywkday .	
weekday1	.
weekday2	.
weekday3	.
weekday4	.
weekday5	.
weekday6	.
holiday1	.
weather.L	12.94021
weather.Q	.
temp	268.83766
humidity	-61.81185
windspeed	.
seasonsummer	.
seasonfall	.
seasonwinter	12.08726
atemp	.

Table 4-2 Model Coefficients According to Lasso Method

4.2 Implement of Stepwise Methods

We selected the AIC for the stepwise method (Table 4-3), since the model will be used for prediction. Also, we select the stepwise procedure (direction=” both”) as most appropriate because of double checking. The procedure identified the following set of predictors (reduced compared to the Lasso model), that will be used to build the final model: Hour, year , temp , humidity , season and weather.

Table 4-3 Step-wise with AIC Method

	Dependent variable:
	count
hourFEvening	-20.868**
	(10.314)
hourFMorning	-17.229*
	(10.253)
hourFNight	-210.334***
	(10.793)
year1	82.706***
	(6.613)
weather.L	42.458***
	(9.768)
weather.Q	-13.517*
	(7.331)
temp	318.255***
	(28.898)
humidity	-77.808***
	(22.495)
seasonsummer	12.471
	(11.822)
seasonfall	5.814
	(14.949)
seasonwinter	55.024***
	(10.149)
Constant	85.471***
	(18.907)
Observations	1,500
R^2	0.533
Adjusted R^2	0.530
Residual Std. Error	126.668 (df = 1488)
F Statistic	154.366*** (df = 11; 1488)
Note:	*p<0.1; **p<0.05; ***p<0.01

In our final model after Lasso and Stepwise methods, the Adjusted R-squared is 0.530 and most of the variables are statistical significant. Also, we have a p-value <0.05 which means that our model performs better than if we had only the constant variate. The model with the final set of predictors:

model1=lm(formula = count ~ hourF + year + temp + humidity + season + weather, data = df2)

5. Predictive Descriptive Models

Before proceeding with inference, the assumptions of multiple linear regression models should be checked.

5.1 Normality Assumption

The normal QQ-plot (Figure 5-1) of the model studentized residuals, shows strong deviations from the normal distribution. Thus, the assumption of normality is violated.

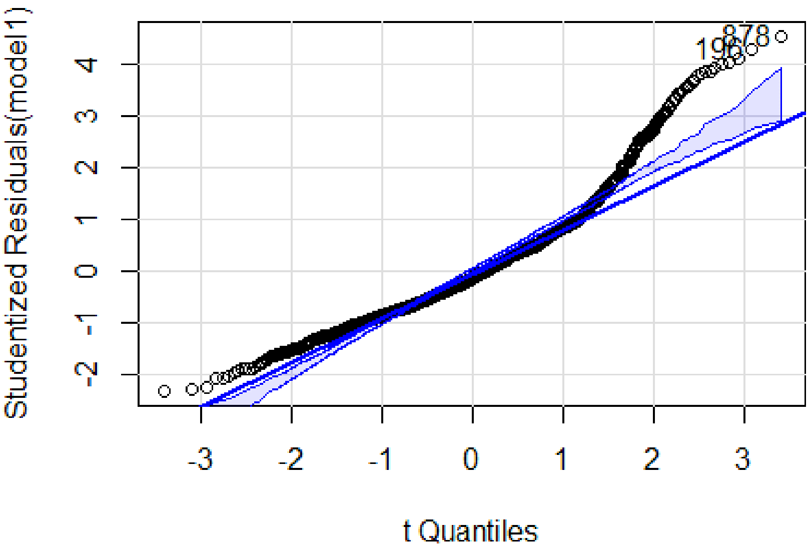

Figure 5-1 QQ-plot Of Studentized Residuals

5.2 Multi Linearity Assumption

The plot of the studentized residuals vs the fitted values (Figure 5-2), shows a non-linear pattern.

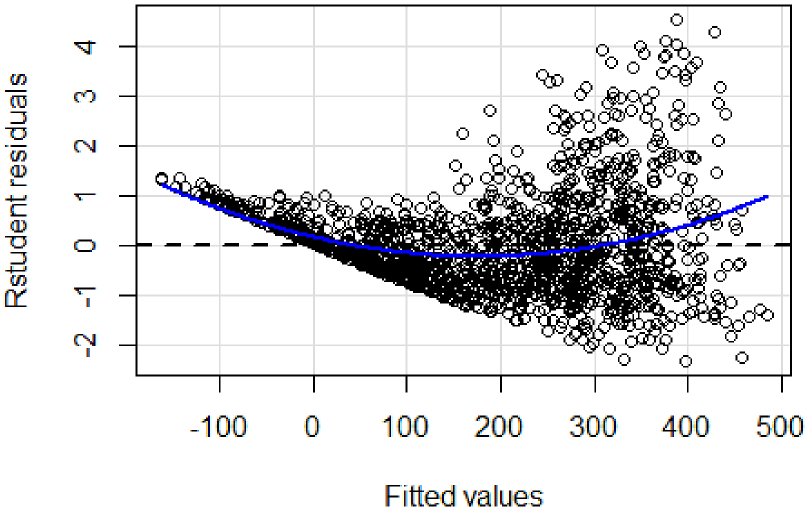

Figure 5-2 Plot of Studentized Residuals vs Fitted Values

A curvature test for each of the numeric predictors was also applied, by adding a quadratic term and testing the quadratic to be zero (Tukey's test for nonadditivity, Table 5-1). The results do not reveal any non-linear pattern with the numeric predictors.

<table>
<tr><th colspan="3"><i>Test stat Pr(>|Test stat|)</i></th></tr>
<tr><td><i>hourF</i></td><td></td><td></td></tr>
<tr><td><i>year</i></td><td></td><td></td></tr>
<tr><td><i>temp</i></td><td>-0.9233</td><td>0.356</td></tr>
<tr><td><i>humidity</i></td><td>-0.5733</td><td>0.5665</td></tr>
<tr><td><i>season</i></td><td></td><td></td></tr>
<tr><td><i>weather</i></td><td></td><td></td></tr>
<tr><td><i>Tukey test</i></td><td>10.7947</td><td><2e-16 ***</td></tr>
<tr><td colspan="3"><i>Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1</i></td></tr>
</table>

Table 5-1 Output from Tukey Test

5.3 Homogeneity of Variance Assumption

The following plots of residuals vs fitted values (Figure 5-3), show strong heteroscedasticity (ie non –constant variance).

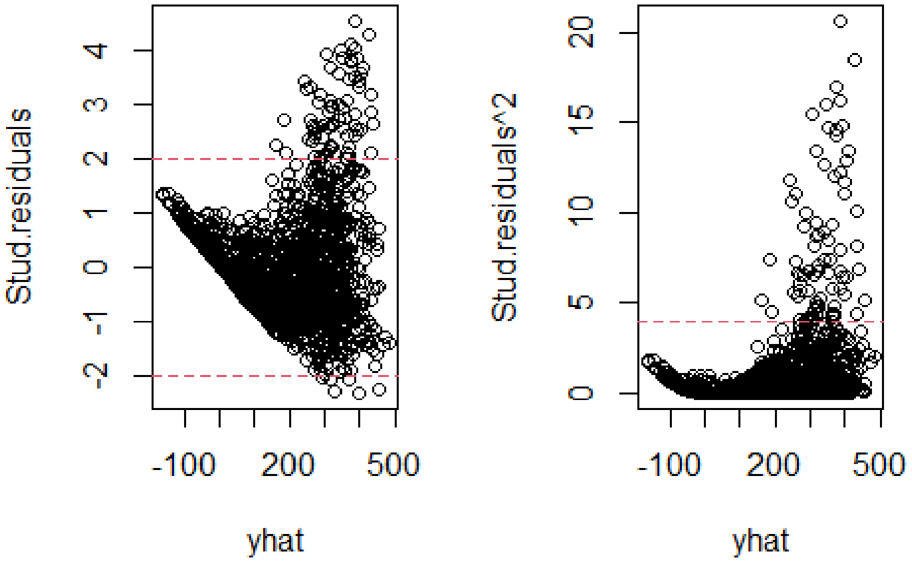

Figure 5-3 Plots of Residuals vs Fitted Values

The pattern is confirmed with the appropriate statistical tests (Score Test for Non-Constant Error Variance and Levene test Table 5-2), where in the null hypothesis of homogeneity is rejected.

Non-constant Variance Score Test

Variance formula:	~ fitted.values	
Chisquare = 342.7035	Df = 1	p = < 2.22e-16

Levene's Test for Homogeneity of Variance (center = median)

	Df	F value	Pr(>F)
group	3	112.47	< 2.2e-16 ***
	1494		

*Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1*

Table 5-2 Statistical tests : Outputs of Score Test and Levene Test

This can be also shown with the boxplots of residuals versus the quantiles of the fitted values (Figure 5-4).

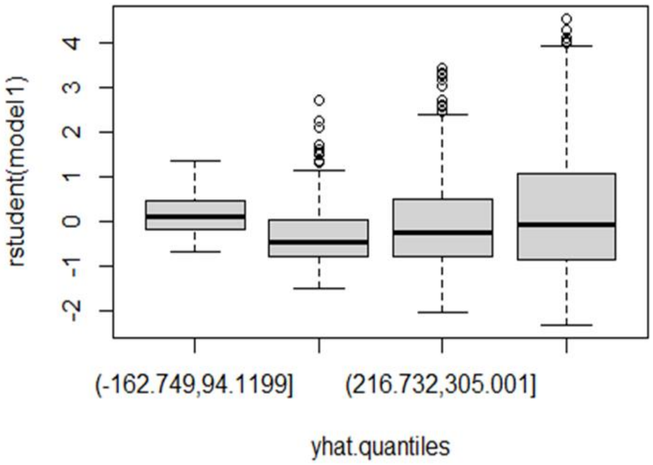

Figure 5-4 Boxplots of residuals versus the quantiles of the fitted values

5.4 Independence Assumption

To check the assumption of independent errors the Durbin-Watson test for autocorrelation was applied to test the null hypothesis that the autocorrelation of the errors is 0. The H0 is not rejected, thus this assumption is fulfilled (Table 5-3).

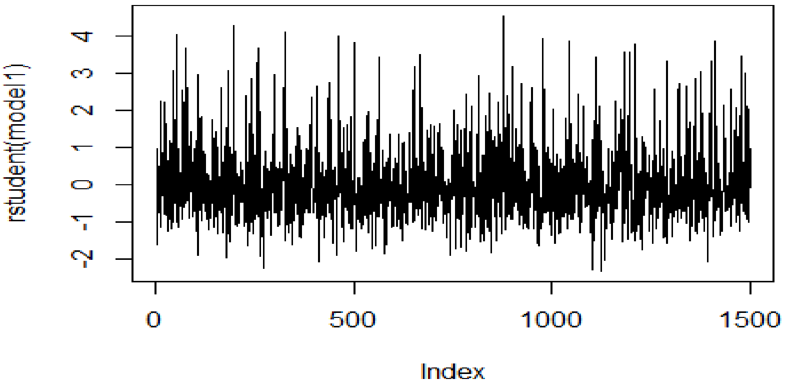

Figure 5-5 Plot Rstudent Model1

lag	*Autocorrelation*	*D-W Statistic*	*p-value*
1	0.02182881	1.955723	0.39
	Alternative hypothesis: rho != 0		

Table 5-3 Durbin-Watson Test Output

5.5 Check of Multi Collinearity

The VIF (Variance Inflation Factors =1/(1-R2)) were used to test for mutli-collinearity (ie (statistically) high linear relationship between one explanatory with (some of) the of the predictors). The square root of the VIFs represents how much larger the standard error is, compared with what it would be if that variable were uncorrelated with the other predictor variables in the model. All the coefficients are close to 1, thus there is no issue of mutli-collinearity (Table 5-4) .

	GVIF	*Df*	*GVIF^(1/(2*Df)*
hourF	1.5	3	1.1
year	1	1	1
temp	2.9	1	1.7
humidity	1.7	1	1.3
season	2.9	3	1.2
weathe	1.4	2	1.1

Table 5-4 VIF Function Output

5.6 Final model

It is necessary to apply transformations in order to correct the above issues of nonnormality, non-linearity and heteroscedasticity. Log and square-root transformations of the response and the predictor variables were tried, but none seemed to correct all the problems. Finally, a Box-Cox transformation was applied to the predictor. This corrected the non-normality problem, but resulted in stronger non-linearity. Thus, polynomials were included for the numeric predictors (humidity and temperature). The above procedure is outlined in the Appendix (10.2 Section «Box Cox Transformation Process»). The final model is the following:

model3 <-lm(formula =((count^lambda-1)/lambda) ~ hourF + year + poly(temp,3) + poly(humidity,2)

+ season + weather , data = df2) where lambda=0.31**.**

The assumptions were re-evaluated for the final model, and all the problems have been fixed (Figures 5-6, 5-7, 5-8 and Tables 5-5, 5-6).

- Normality Assumption

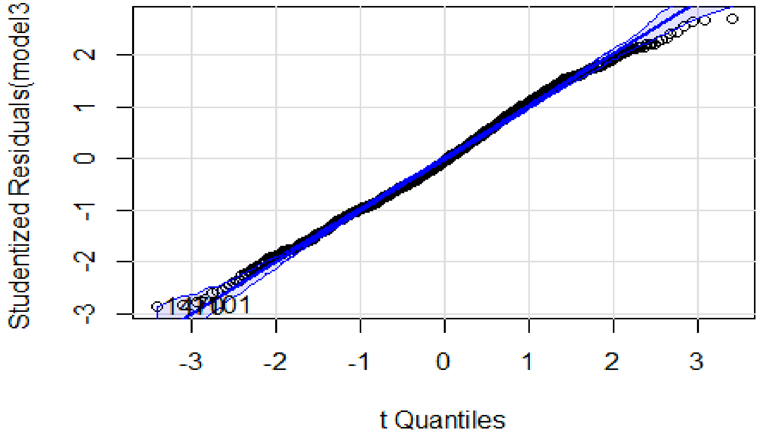

Figure 5-7 1 QQ-plot Of Studentized Residuals

- Constant Variance Assumption (1)

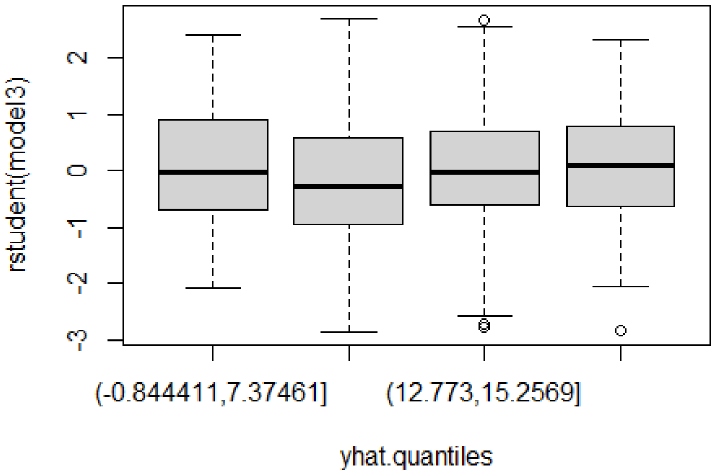

Figure 5-6 Boxplots of residuals versus the quantiles of the fitted values

- Independence Assumption

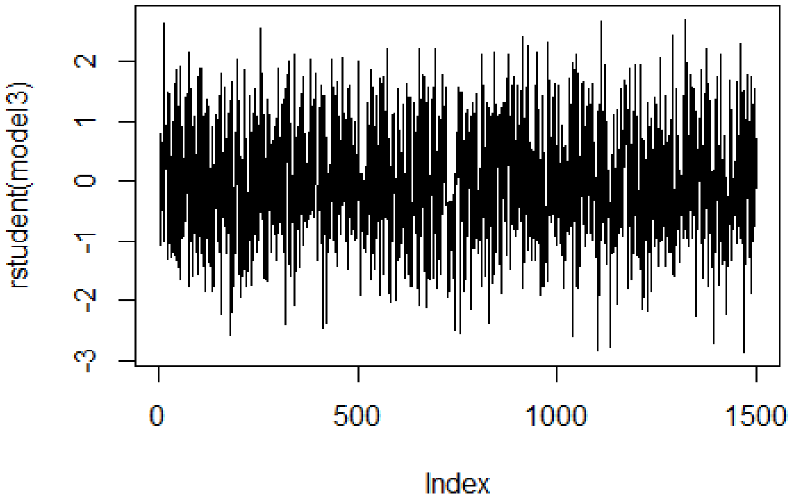

Figure 5-8 Plot Rstudent Model3

- Constant Variance Assumption (2)

Levene's Test for Homogeneity of Variance

(center = median)

	Df	F value	Pr(>F)
group	3	2.198	0.0865 .
	1494		

*Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1*

Table 5-5 Levene's Test for Homogeneity of Variance

- Linearity Assumption

	Test stat	*Pr(>|Test stat|)*
hourF		
year		
poly(temp, 3)		
poly(humidity, 2)		
season		
weather		
Tukey test	6.6965	2.134e-11 ***

*Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1*

Table 5-6 Output from Tukey Test

6. Interpreting Parameters of Final Model

The Interpretation of our final model is the following (Table 6-1):

Count ^ 0.31-1/0.31 = 12.12

– 0.55 * hourFEvening - 0.37 * hourFMorning - 7.90 * hourFNight

+ 1.94 * year1 + 65.52 * temp – 13.88 * temp^2 -12.66* temp^3

- 25.58 * humidity - 5.87 * humidity^2

+ 0.27 *seasonsummer + 0.27 * seasonfall +1.85 * seasonwinter

+ 1.12 *weatherL -0.46 * weather.Q

+ ε ~ N (0, 3.027^2)

Table 6-1 Interpretation of Final Model

The **intercept**, represents the (box-cox transformed) count of bike rentals when Season =Spring, Hour= afternoon, weather= good, year=2011 and temp and humidity are equal to 0.

Regarding the predictors, the count of rental bikes, is affected by:

Year: the year 2012 increases the box-cox transformed count of bike rentals by 1.94. That would be sensible, if the year 2012 the circumstances were more ideal for users to rent bikes (e.g. due newest technology in bike systems).

Hour: The bikes rental count is lower during night time, compared to the Afternoon (included in the Intercept) by 7.90, with all other predictors being equal. The same holds during the Evening, but the difference is lower (-0.55). The difference between Afternoon and Morning is not statistically significant.

Season: The response in higher by 1.85, in Winter compared to Spring (included in the intercept), when all other variables are equal. That means that possibly people may prefer more to use bikes in Winter period than in Spring. The difference of the other seasons compared to Spring is not statistically significant.

Weather: It seems that users do not prefer Bad Weather conditions and it decreases the box-cox transformed count of bike rentals by -0.46 compared to Good (included in the intercept), all other predictors being equal. On the other hand, Very Good Weather conditions increase box-cox transformed count of bike rentals by 1.12. That makes sense, since if its heavily raining or snowing, people will probably prefer a car or other safer transport systems.

	Model 1
(Intercept)	12.12 ***
	[11.58, 12.67]
hourFEvening	-0.55 *
	[-1.04, -0.06]
hourFMorning	-0.37
	[-0.86, 0.11]
hourFNight	-7.90 ***
	[-8.41, -7.39]
year1	1.94 ***
	[1.63, 2.25]
poly(temp, 3)1	65.52 ***
	[55.34, 75.71]
poly(temp, 3)2	-13.88 ***
	[-20.66, -7.11]
poly(temp, 3)3	-12.66 ***
	[-18.87, -6.46]
poly(humidity, 2)1	-25.28 ***
	[-33.30, -17.27]
poly(humidity, 2)2	-5.87
	[-12.15, 0.40]
seasonsummer	0.27
	[-0.29, 0.84]
seasonfall	0.27
	[-0.43, 0.98]
seasonwinter	1.85 ***
	[1.35, 2.34]
weather.L	1.12 ***
	[0.65, 1.60]
weather.Q	-0.46 **
	[-0.81, -0.11]
N	1500
R2	0.7
*** p < 0.001; ** p < 0.01; * p < 0.05.	

Table 6-2 Final Model Summary

The overall **R adjusted** in close to 0.7, indicating a satisfactory fit.

7. Models Predictive Ability

The models predictive ability was assessed using the following metrics:

- **Mean Squared Error (MSE):** it represents the average of the squared difference between the original and predicted values in the data set. It measures the variance of the residuals.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2$$

Figure 7-1 Mean Squared Error Formula

- **Root Mean Squared Error (RMSE)**: the square root of Mean Squared error. It measures the standard deviation of residuals

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2}$$

Figure 7-2 Formula Root Mean Squared Error (RMSE)

- **Mean absolute error (MAE):** it represents the average of the absolute difference between the actual and predicted values in the dataset. It measures the average of the residuals in the dataset. MAE is more robust to data with outliers

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}|$$

Figure 7-3 Formula Mean absolute error (MAE)

The lower value of MAE, MSE, and RMSE implies higher accuracy of a regression model. The full model is the model including all the predictors, while the null model includes only the intercept. For all models the Box-Cox transformed response was used for fitting. The following results were obtained (Table 7-1):

Model	*RMSE*	*MAE*	*MSE*
1 Final	3.104045	2.569887	9.635095
2 Null	5.574803	4.645834	31.078429
3 Full	3.109560	2.590668	9.669361

Table 7-1 Metrics Summary of Predictive Ability of all Models

These results suggest that the Final model outperforms the null model and has comparable results as the full model. However, in comparison to the Full model, it depends on a smaller set of predictors, and it has been formulated appropriately in respect to the linear regression assumptions.

8. Typical Profile of a Day for Each Season

Typical day by year-hour

In order to describe the typical profile for a day, the mean of the numeric predictors needs to be calculated for every level of the factor predictors of the model. In the present model, the mean humidity and temperature were calculated for every year- season-weather and time period (hourF). Subsequently, the final model was used to predict the resulting “count” for each combination of the predictor values. After predicting the count, we applied inverse box-cox transformation, using the following formula :

$$y_t = \begin{cases} \exp(w_t) & \text{if } \lambda = 0; \\ \text{sign}(\lambda w_t + 1)|\lambda w_t + 1|^{1/\lambda} & \text{otherwise.} \end{cases}$$

Figure 8-1 Inverse Box-Cox Transformation Formula

The following results were obtained (Figure 8-2):

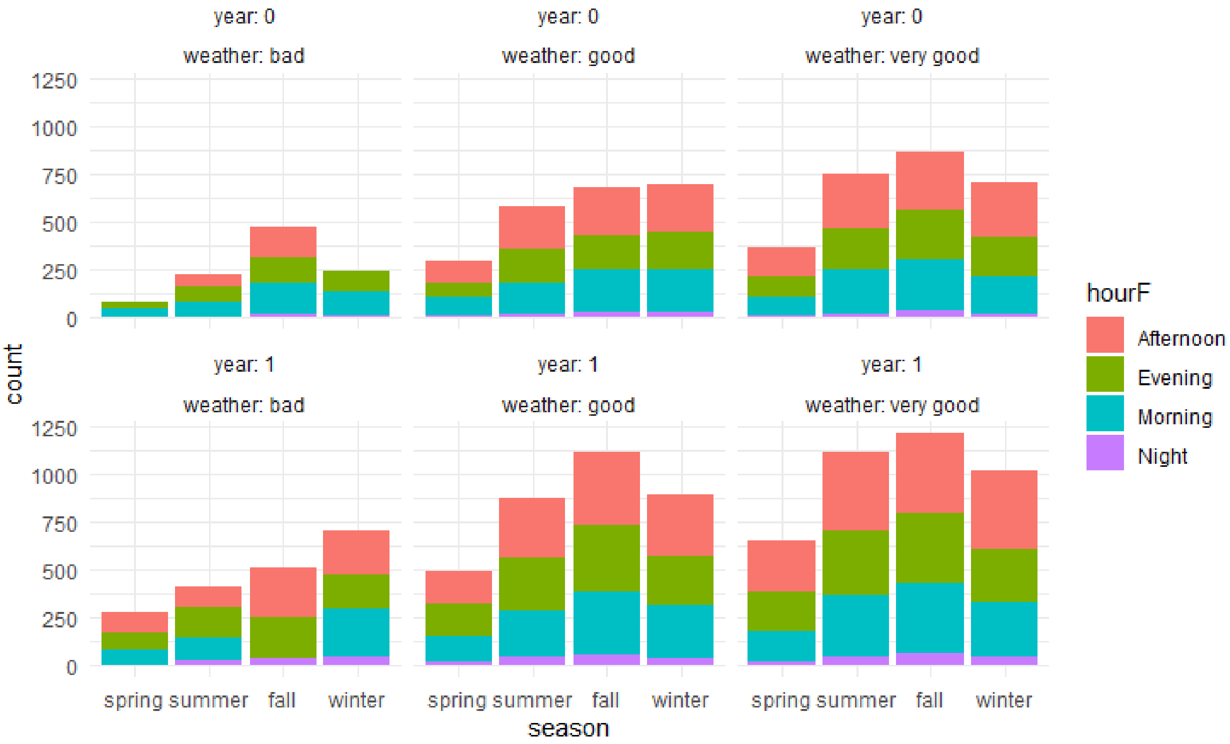

Figure 8-2 Histograms with response variable count against variables “season”, “weather” and “year”

From Figure 8-2 we can conclude that in both years, users on average prefer days with very good weather conditions. Additionally, in both years, users mostly avoid renting bikes at night and prefer to rent bikes in daylight hours. Also, we can see the days of the year 1 (1 = 2012) have on average higher amount of bike rentals than year 0 (0 = 2011). We can also calculate the mean “count” across years and weather conditions, as follows (Figure 8-3):

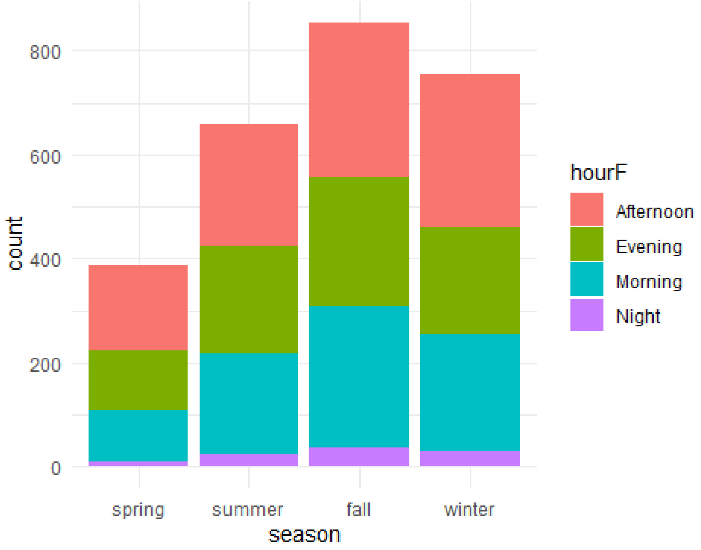

Figure 8-3 Histogram with mean of response variable “count” against “years” and “weather” conditions

9. Conclusions

Conducting the above analysis with different tests and plots we ended up with a final model that fits well with our data set. The methodology followed was to first undergo data cleaning and transformation in order to achieve the highest possible accuracy later on in our analysis. Second, we conducted exploratory data analysis and pairwise comparisons, to allow for a deeper look into our data relationships and patterns, without the need for complex summary statistics. Third, we implemented Lasso and Stepwise methods, in order to aid in the examination of the statistical significance of each predictor variable and to remove the insignificant ones from our model. Fourth, we checked the assumptions of our model and applied the necessary transformations to ensure that all assumptions were met. This allowed us to draw accurate conclusions from the results. Interpreting the model parameters, was key to understand deeper some of our findings. Some useful insights conducting the above analysis are:

- A box cox transformation was deemed necessary, to facilitate the normality and homoscedasticity assumptions of linear regression
- It should be noted that the transformation of the dependent variable, complicates the direct interpretation of the model coefficients. Nevertheless, fulfilling the assumptions, is necessary for obtaining trustworthy predictions.
- A potential improvement in our procedure, would be to investigate the application of Poisson regression, given the count nature of the outcome

10. Appendix

10.1 Additional Figures

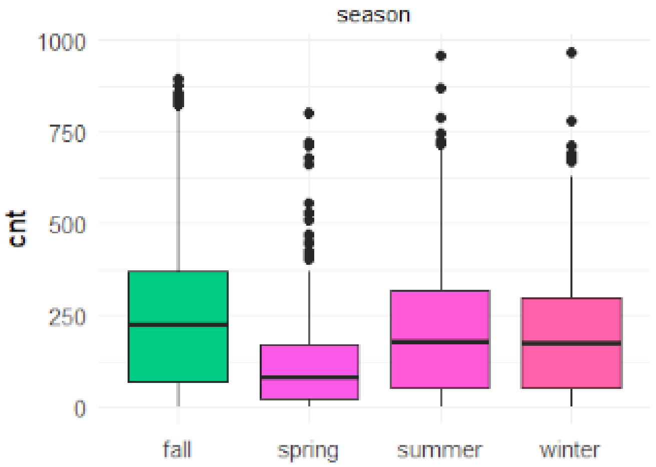

Figure 10-1 Boxplot with the Response Variable "count" Against "season" Variable

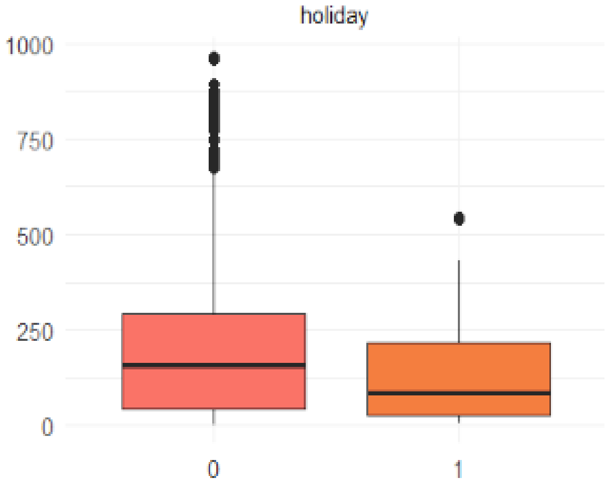

Figure 10-2 Boxplot with the Response Variable "count" Against "holiday" Variable, where “0” stands for non-holiday and “1” for holiday

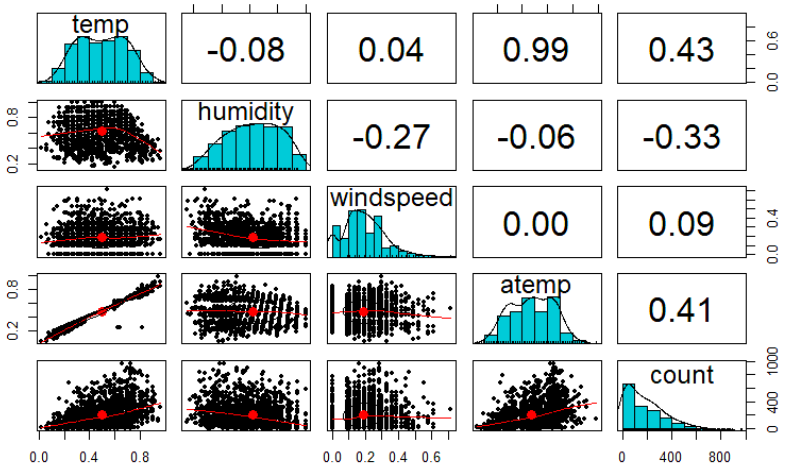

Figure 10-3 Pairwise Comparison for Numeric Variables

10.2 Box Cox Transformation Process

The standard (simple) Box-Cox transform for the response is:

$$Y_i^{(\lambda)} = \begin{cases} \dfrac{Y_i^{\lambda} - 1}{\lambda} & (\lambda \neq 0) \\ \log(Y_i) & (\lambda = 0) \end{cases}$$

The following plot shows the log-likelihood profile versus lambda obtained for the initial model and used to obtain the lambda (0.31), used for the Box-Cox Transformation of the response:

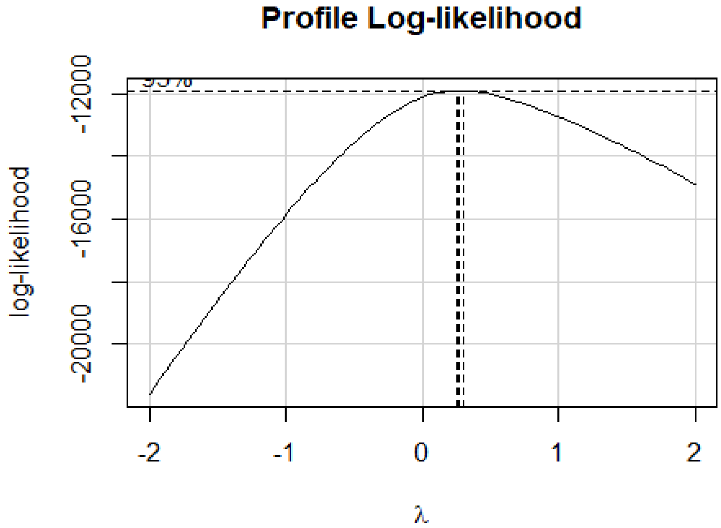

Table 10-1 Log-likelihood profile versus lambda

Check assumptions

The visual investigation of the residual plots for the model with box-cox transformed response and with polynomials for temp and humidity, shows that the assumptions of linear regression are generally fulfilled.

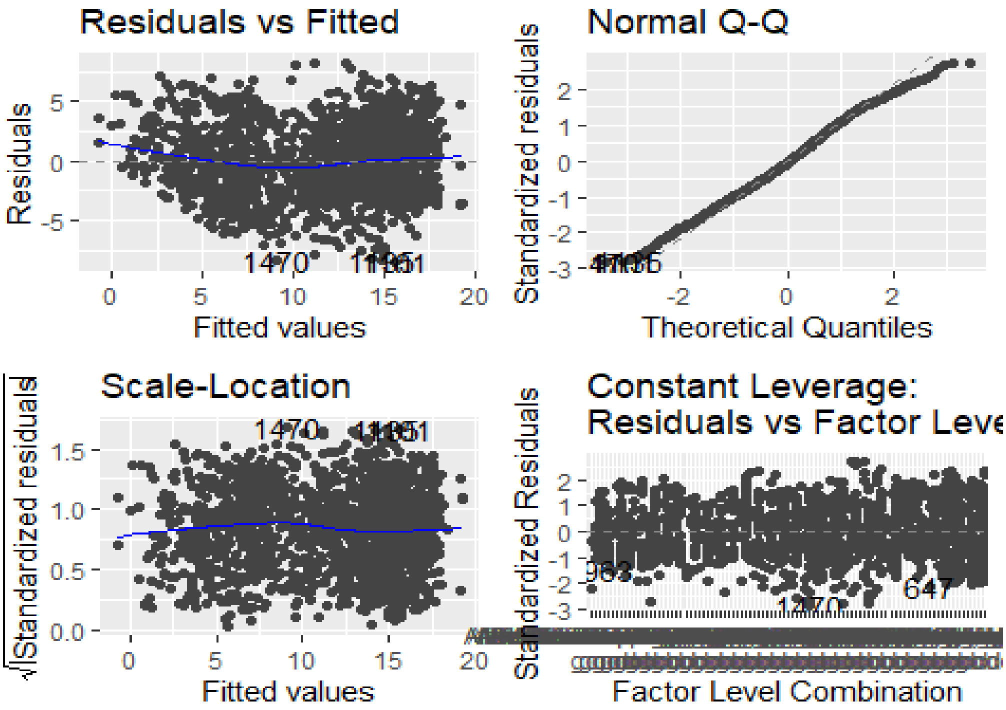

Figure 10-4 Residual Plots of Final Model

<table>
<tr><th colspan="3">Test stat Pr(>|Test stat|)</th></tr>
<tr><td><b>hourF</b></td><td></td><td></td></tr>
<tr><td><b>year</b></td><td></td><td></td></tr>
<tr><td><b>temp</b></td><td>-4.0397</td><td>5.624e-05 ***</td></tr>
<tr><td><b>humidity</b></td><td>-2.0129</td><td>0.0443 *</td></tr>
<tr><td><b>season</b></td><td></td><td></td></tr>
<tr><td><b>weather</b></td><td></td><td></td></tr>
<tr><td><b>Tukey test</b></td><td>4.1012</td><td>4.110e-05 ***</td></tr>
<tr><td colspan="3"><b>Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1</b></td></tr>
</table>

Table 10-2 Tukey Test Output

	Model 1
(Intercept)	12.10 ***
	[11.55, 12.65]
hourFEvening	-0.55 *
	[-1.04, -0.06]
hourFMorning	-0.37
	[-0.85, 0.12]
hourFNight	-7.88 ***
	[-8.39, -7.37]
year1	1.95 ***
	[1.64, 2.26]
poly(temp, 5)1	65.73 ***
	[55.48, 75.98]
poly(temp, 5)2	-14.10 ***
	[-20.93, -7.27]
poly(temp, 5)3	-12.95 ***
	[-19.18, -6.73]
poly(temp, 5)4	-2.13
	[-8.27, 4.01]
poly(temp, 5)5	-0.45
	[-6.49, 5.58]
poly(humidity, 5)1	-25.33 ***
	[-33.41, -17.25]
poly(humidity, 5)2	-5.53
	[-11.85, 0.80]
poly(humidity, 5)3	3.39
	[-2.67, 9.44]
poly(humidity, 5)4	0.65
	[-5.36, 6.65]
poly(humidity, 5)5	2.87
	[-3.10, 8.84]
seasonsummer	0.28
	[-0.28, 0.85]
seasonfall	0.26
	[-0.45, 0.97]
seasonwinter	1.84 ***
	[1.35, 2.34]
weather.L	1.15 ***
	[0.67, 1.63]
weather.Q	-0.47 **
	[-0.82, -0.12]
N	1500
R2	0.70
*** p < 0.001; ** p < 0.01; * p < 0.05.	

Table 10-3 Model Summary

References

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, doi:10.1007/s13748-013-0040-3

Rencher, Alvin C.; Christensen, William F. (2012), "Chapter 10, Multivariate regression – Section 10.1, Introduction", Methods of Multivariate Analysis, Wiley Series in Probability and Statistics, 709 (3rd ed.), John Wiley & Sons, p. 19, ISBN 9781118391679.

Faraway, J. (2002). Practical regression and ANOVA using R; https://cran.r-project.org/doc/contrib/Faraway-PRA.pdf

Fox J. & Weisberg H.S. (2011). An R Companion to Applied Regression.2nd edition. SAGE Publications Inc

Rui Miguel Forte (2015). Mastering Predictive Analytics with R Paperback. Packt Publishing

Elegant Graphics for Data Analysis ggplot2 (https://ggplot2-book.org/) by Wickham, Navarro & Lin Pedersen

R Code Used for Statistical Analysis

```
#' ---
#' title: 'Title Bike Sharing Dataset '
#' author: "Spyridon Despotis"
#'
#'
#' # 1) You should first perform some descriptive data analysis and
visualization. Visualizing the data should give you some insight into
certain particularities of this dataset. Pairwise comparisons will help you
also learn about the association implied by the data.
#'
#' Dataset structure:

library(stargazer)
library(tidyverse)
library(knitr)
library(psych)
library(reshape)
library("ggcorrplot")
library("ggmosaic")
library(ggplot2)
library(MASS)
library(car)
library(lmtest)
library(nortest)
library(jtools)
library(psych)

df <- read.csv("bike_29.csv",sep=";",dec = ",")

str(df)

#'
#' Investigating missing values :
#'
#'


df1 <- df

#mising values
sapply(df1, function(x) sum(is.na(x)))




#'
#' Recoding Factors:
#' Factor levels were recoded from numeric to more meaningful descriptive
strings.
#' The "hour" variable was converted from continuous to factor variable
with the following levels: mornig, afternoon, evening, night




df1$season <- factor(df1$season
                     ,levels = c(1,2,3,4)
                     ,labels = c("spring", "summer", "fall", "winter"))
```

```
df1$workingday <- factor(df1$workingday
                         ,levels = c(0,1)
                         ,labels = c("nonwkday", "wkday"))

df1$weathersit <- factor(df1$weathersit
                      ,levels = c(4,3,2,1)
                      ,labels = c("very bad", "bad", "good", "very good")
                      ,ordered = TRUE)


#Converting hour

df1$hourF <- as.factor(case_when(
df1$hr >= 7 & df1$hr <= 12 ~ 'Morning',
 df1$hr > 12 & df1$hr <= 17 ~ 'Afternoon',
df1$hr > 17 & df1$hr <= 22 ~ 'Evening',
TRUE~ "Night"
))

df2 <- df1 %>%
  mutate(season = as.factor(season), year = as.factor(yr), month =
as.factor(mnth),
         holiday = as.factor(holiday), weekday = as.factor(weekday),
workingday = as.factor(workingday),
         registered = as.numeric(registered), c = as.numeric(casual),
         weather = as.factor(weathersit), hour = as.numeric(hr), count =
as.numeric(cnt), humidity = hum) %>%
  dplyr::select(count, hourF, year, month, workingday, weekday, holiday,
weather, temp, humidity, windspeed,season, atemp)






str(df2)

#'

#Categorizing data
df_numeric= df2 %>% select_if(is.numeric)

df_factor = df2 %>% select_if(is.factor)

#'
#' ## Descriptive analysis
#'
#' ### Visualising the response variable
#'

#visualising count
p <- ggplot(df2, aes(count, fill = cut(count, 100))) +
  geom_histogram(show.legend = FALSE) +
  theme_minimal() +
  labs(x = "Count", y = "n") +
  ggtitle("Number of Total Rents")
p + scale_fill_discrete(h = c(180, 360), c = 150, l = 80)
```

```
#'
#'
#' ### Visualising continious variables: histograms
#'


#visualising continious variables
par(mfrow=c(2,3));
for(k in 1:ncol(df_numeric)){
  hist(df_numeric[,k], main=names(df_numeric)[k])
}

#'
#' ### Visualising factors: barplots


df3=cbind(df_factor %>% mutate_all(as.character),cnt= df2$count) %>%
pivot_longer(-cnt)

df4=df3 %>% dplyr::select(-cnt) %>% group_by(name,value) %>% count()

ggplot(df4) +
  aes(x = value, fill = value, weight = n) +
  geom_bar() +
  scale_fill_hue(direction = 1) +
  theme_minimal() +
  theme(legend.position = "none") +
  facet_wrap(vars(name), scales = "free")


#'
#' ### Visualising relationship between response and factor variables:
boxplots
#'


ggplot(df3) +
  aes(x = value, y = cnt,fill = value) +
  geom_boxplot() +
  theme_minimal() +
  facet_wrap(vars(name), scales = "free")+
  theme(legend.position="none")


#'
#' ### Visualising relationship between the numeric variables: Correlations
#'

#correlation numeric
dumcor=cor(df_numeric)
ggcorrplot(dumcor,method = "square", outline.color = "black", type =
"lower", lab = TRUE)

#'
pairs.panels(df_numeric%>% dplyr::select(-count,everything(),count),
             method = "pearson", # correlation method
             hist.col = "#00AFBB",
             density = TRUE,  # show density plots
             ellipses = TRUE # show correlation ellipses
)
#'
#' #  2. The main aim is to identify the best model for predicting the
number of bike rentals per hour (variable cnt).
#' ## a) Implement Lasso in order to select the covariates of your model.
#'

###Lasso-

require(glmnet)
model1 <- lm(count~.,data=df2 )
X <- model.matrix(model1)[,-1]
lasso <- glmnet(X, df2$count)
# plot(lasso, xvar = "lambda", label = T)

#Use cross validation to find a reasonable value for lambda
lasso1 <- cv.glmnet(X, df2$count, alpha = 1)



#'

# lasso1$lambda
 lasso1$lambda.min
 lasso1$lambda.1se



#'
#'

plot(lasso1)


#'


plot(lasso1$glmnet.fit, xvar = "lambda")
abline(v=log(c(lasso1$lambda.min, lasso1$lambda.1se)), lty =2)

#'



coef(lasso1, s = "lambda.1se")



#'
#' ## b) Select the appropriate features (after implementing lasso) using
stepwise methods in order to select your final model.
#' Be careful, your model should not be over-parameterized
#'
#' ### Step-wise with AIC

#STEPWISE- BIC


model0 <- lm(formula = count ~ hourF + year + month + weather + temp +
humidity + season, data = df2)
```

```
step_m<-step(model0, direction='both',trace=0)

step_m

#'
#'
#'
#' # 3. Check the assumptions of the model and revise your procedure
#'
#'  ### Model with the final set of predictors
#'


model1=lm(formula = count ~ hourF + year + temp + humidity +
season+weather,     data = df2)

#'
#'
#'
#' ## Check the normality assumption
#'

qqPlot(model1)

#'
#' ## Check the Linearity assumption
#'


residualPlot(model1, type='rstudent')
residualPlots(model1, plot=F, type = "rstudent")

#'
#' ## Check the homogeneity of variance assumption
#'


Stud.residuals <- rstudent(model1)
yhat <- fitted(model1)
par(mfrow=c(1,2))
plot(yhat, Stud.residuals)
abline(h=c(-2,2), col=2, lty=2)
plot(yhat, Stud.residuals^2)
abline(h=4, col=2, lty=2)




#'

ncvTest(model1)

#'


yhat.quantiles<-cut(yhat, breaks=quantile(yhat, probs=seq(0,1,0.25)),
dig.lab=6)

leveneTest(rstudent(model1)~yhat.quantiles)
```

```
#'


boxplot(rstudent(model1)~yhat.quantiles)


#'
#' ## Check the independence assumption
#'

# Independence
# --------------------
plot(rstudent(model1), type='l')

 durbinWatsonTest(model1)

#'
#' ## Check Multi Collinearity
#'


#Using VIF
require(car)
round(vif(model1),1)


#'
#' ## Final model
#'

lambda=.31

model3 <-lm(formula =((count^lambda-1)/lambda) ~ hourF +year + poly(temp,3)
+poly( humidity,2) + season+weather , data = df2)

#'
#' # Re- checking assumptions
#'
#' ### Normality

qqPlot(model3)

#'
#'
#' ### Linearity
#'

residualPlots(model3, plot=F, type = "rstudent")

#'
#' ### Constant variance
#'

yhat <- fitted(model3)
yhat.quantiles<-cut(yhat, breaks=quantile(yhat, probs=seq(0,1,0.25)),
dig.lab=6)
```

```
leveneTest(rstudent(model3)~yhat.quantiles)


boxplot(rstudent(model3)~yhat.quantiles)

#'
#' ### Independence


plot(rstudent(model3), type='l')


#'
#' # 4. Interpret the parameters and the predicting performance of the
final model.
#'


export_summs(model3,
             error_format = "[{conf.low}, {conf.high}]")

#'
#' # 5. Use the test dataset to assess the out-of-sample predictive ability
and compare the models selected in Q2. Also include the full and the null
models in your comparison.
#'

# Test dataset

test_dat= read.csv("bike_test.csv",sep=";",dec = ",")

test_dat$season <- factor(test_dat$season
                     ,levels = c(1,2,3,4)
                     ,labels = c("spring", "summer", "fall", "winter"))

test_dat$workingday <- factor(test_dat$workingday
                         ,levels = c(0,1)
                         ,labels = c("nonwkday", "wkday"))

test_dat$weathersit <- factor(test_dat$weathersit
                      ,levels = c(4,3,2,1)
                      ,labels = c("very bad", "bad", "good", "very good")
                      ,ordered = TRUE)


test_dat2 <- test_dat %>%
  mutate(season = as.factor(season), year = as.factor(yr), month =
as.factor(mnth),
         holiday = as.factor(holiday), weekday = as.factor(weekday),
workingday = as.factor(workingday),
         registered = as.numeric(registered), c = as.numeric(casual),
         weather = as.factor(weathersit), hour = as.numeric(hr), count =
as.numeric(cnt), humidity = hum) %>%
 dplyr:: select( count,hour, year, month, workingday, weekday, holiday,
weather, temp, humidity, windspeed,season, atemp)

test_dat2$hourF<- as.factor(case_when(
test_dat$hr >= 7 & test_dat$hr <= 12 ~ 'Morning',
 test_dat$hr > 12 & test_dat$hr <= 17 ~ 'Afternoon',
test_dat$hr > 17 & test_dat$hr <= 22 ~ 'Evening',
TRUE~ "Night"
))

test_dat2$resp=(test_dat2$count^lambda-1)/lambda



#'
#'

# RMSE - Final model

# Predict on test:
pred1 <- predict(model3, newdata = test_dat2, type = "response")

# Compute errors: error
error <- test_dat2$resp - pred1

RMSE_res=data.frame(model=c("Final","Null","Full"),RMSE=c(NA),MAE=NA,MSE=NA
)


RMSE_res$RMSE[1] <- sqrt(mean(error^2))

RMSE_res$MSE[1]= mean((error)^2)
RMSE_res$MAE[1] = mean(abs(error))


#'
#'

# RMSE - Null model

null_mod=lm((count^lambda - 1)/lambda~1,data=df2)

# Predict on test:
pred2 <- predict(null_mod, newdata = test_dat2, type = "response")

# Compute errors: error
error2 <- test_dat2$resp - pred2

RMSE_res$RMSE[2] <- sqrt(mean(error2^2))


RMSE_res$MSE[2]= mean((error2)^2)
RMSE_res$MAE[2] = mean(abs(error2))

#'
#'

# RMSE - Full model

full_mod=lm((count^lambda - 1)/lambda~.,data=df2)

# Predict on test:
pred3 <- predict(full_mod, newdata = test_dat2, type = "response")

# Compute errors: error
```

```r
error3 <- test_dat2$resp - pred3

RMSE_res$RMSE[3] <- sqrt(mean(error3^2))


RMSE_res$MSE[3]= mean((error3)^2)
RMSE_res$MAE[3] = mean(abs(error3))

RMSE_res


# 6--------
#'
#' # 6. Describe the typical profile of a day for each season (autumn,
winter, spring, summer).---------
#'
#'
#' ## Typical day by year-hour



dat_typ=df2 %>% group_by(season,hourF,weather,year) %>%
summarise(temp=mean(temp),humidity=mean(humidity)) %>% ungroup()




#'

# Predict on typical dat:
dat_typ$pred_typ1 <- predict(model3, newdata = dat_typ, type = "response")

# reverse box-cox
invBoxCox <- function(x, lambda)
  if (lambda == 0) exp(x) else (lambda*x + 1)^(1/lambda)


dat_typ$pred_typ <- invBoxCox(dat_typ$pred_typ1,lambda )

ggplot(dat_typ) +
  aes(x = season, fill = hourF, weight = pred_typ) +
  geom_bar() +
  scale_fill_hue(direction = 1) +
  theme_minimal() +
  facet_wrap(~year+weather,labeller = label_both)


dat_typ2=dat_typ %>% group_by(season,hourF) %>%
summarise(mean_cnt=mean(pred_typ))

ggplot(dat_typ2) +
  aes(x = season, fill = hourF, weight = mean_cnt) +
  geom_bar() +
  scale_fill_hue(direction = 1) +
  theme_minimal()


#'
#' # APPENDIX
#'
```

```
#' ## Box Cox transformation
#'


model1 <-lm(formula = ( count ~ hourF +year + temp + humidity+
season+weather), data = df2)

boxCox(model1, plotit = TRUE)




#'
#'
#' ## Checking linearity assumption of Box Cox Model
#'

lambda=0.31

model2.1 <-lm(formula =((count^lambda-1)/lambda) ~ hourF +year + temp +
humidity + season +weather , data = df2)

residualPlots(model2.1, plot=F, type = "rstudent")


#'

model2.3 <-lm(formula =((count^lambda-1)/lambda) ~ hourF +year +
poly(temp,5) + poly(humidity,5) + season +weather , data = df2)


#'
#' Model summary


export_summs(model2.3,
             error_format = "[{conf.low}, {conf.high}]")

#'
#' Check assumptions


library(ggfortify)

autoplot(model2.3)
#'
```