2019

Bike Rental Project Report

LOKANATH MOHAPATRA

CONTENT PAGE NUMBER

1) INTRODUCTION

1.1) Problem statement

A bike rental is a bicycle business that rents bikes for short periods of time. Most rentals are provided by bike shops as a sideline to their main businesses of sales and service, but some shops specialize in rentals. Bike rental shops rent by the day or week as well as by the hour, and these provide an excellent opportunity for people who don't have access to a vehicle, typically travelers and particularly tourists. The fees are set to encourage renting the bikes for a few hours at a time, rarely more than a day. The objective of this Case is to predict the bike rental count based on the environmental and seasonal settings, so that required bikes would be arranged and managed by the shops according to environmental and seasonal conditions.

1.2) Data

Our task is to build regression models which will predict the count of bike rented depending on various environmental and seasonal conditions Given below is a sample of the data set that we are using to predict the count of bike rents:

Table 1.2(a) sample data [columns1-8]

Insert	dteday	season	yr	mnth	holiday	weekday	workingday
1	01-01-2011	1	0	1	0	6	0
2	02-01-2011	1	0	1	0	0	0
3	03-01-2011	1	0	1	0	1	1
4	04-01-2011	1	0	1	0	2	1
5	05-01-2011	1	0	1	0	3	1
6	06-01-2011	1	0	1	0	4	1
7	07-01-2011	1	0	1	0	5	1

Table 1.2(b) sample data [columns 9-16]

weathersit	temp	atemp	hum	windspeed	casual	registered	cnt
2	0.344167	0.363625	0.805833	0.160446	331	654	985
2	0.363478	0.353739	0.696087	0.248539	131	670	801
1	0.196364	0.189405	0.437273	0.248309	120	1229	1349
1	0.2	0.212122	0.590435	0.160296	108	1454	1562
1	0.226957	0.22927	0.436957	0.1869	82	1518	1600
1	0.204348	0.233209	0.518261	0.089565	88	1518	1606
2	0.196522	0.208839	0.498696	0.168726	148	1362	1510

Variables present in the data set are instant, dteday, season, yr, mnth, holiday, weekday, workingday, weathersit, temp, atemp, hum, windspeed, casual, registered, cnt.

The details of data attributes in the dataset are as follows -
instant: Record index
dteday: Date
season: Season (1:spring, 2:summer, 3:fall, 4:winter)
yr: Year (0: 2011, 1:2012)
mnth: Month (1 to 12)
hr: Hour (0 to 23)
holiday: weather day is holiday or not (extracted fromHoliday Schedule)
weekday: Day of the week
workingday: If day is neither weekend nor holiday is 1, otherwise is 0.
weathersit: (extracted fromFreemeteo)
1: Clear, Few clouds, Partly cloudy, Partly cloudy
2: Mist + Cloudy, Mist + Broken clouds, Mist + Few clouds, Mist
3: Light Snow, Light Rain + Thunderstorm + Scattered clouds, Light Rain + Scattered clouds
4: Heavy Rain + Ice Pallets + Thunderstorm + Mist, Snow + Fog
temp: Normalized temperature in Celsius. The values are derived via
(t-t_min)/(t_max-t_min),
t_min=-8, t_max=+39 (only in hourly scale)
atemp: Normalized feeling temperature in Celsius. The values are derived via
(t-t_min)/ (t_maxt_)
min), t_min=-16, t_max=+50 (only in hourly scale)
registered: count of registered user
hum: Normalized humidity. The values are divided to 100 (max)
windspeed: Normalized wind speed. The values are divided to 67 (max)
cnt: count of total rental bikes including both casual & registered
casual: count of casual users.

2) METHODOLOGY

2.1) Data Pre-processing

Any predictive modelling requires that we look at the data before we start modelling. However, in data mining terms *looking at data* refers to so much more than just looking. Looking at data refers to exploring the data, cleaning the data as well as visualizing the data through graphs and plots. This is often called as Exploratory Data Analysis.

2.1.1) Exploratory data analysis

- we have Converted season, mnth, workingday, weathersit into categorical variables
- Feature Engineering: Changed deday variable's date value to day of date and converted to categorical variable having 31 levels as a month has 31 days.
- Deleted instant variable as it is nothing but an index.
- Omitted registered and casual variable as sum of registered and casual is the total count that is what we have to predict.

2.1.2) Univariate analysis

- In Figure 2.1 and 2.2(a,b,c,d) we have plotted the probability density functions numeric variables present in the data including target variable cnt.
    - Target variable 'cnt' is normally distributed.
    - Independent variable 'temp' & 'atemp' data is distributed normally.
    - Other independent variable 'hum' data is slightly skewed to the left. Here, data is in normalised form so outliers are discarded.
    - 'windspeed' data is slightly skewed to the right. So there is chance of getting outliers.

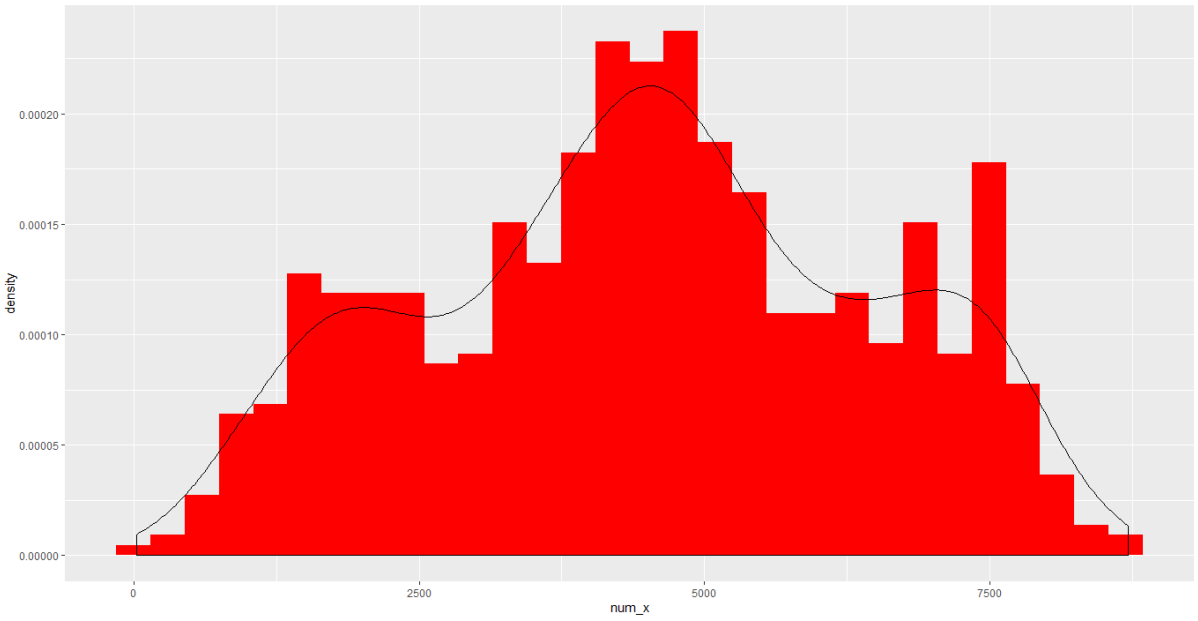


Figure 2.1

The R code for univariate analysis of ‘cnt’ is:

```
# function to create univariate distribution of numeric  variables
univariate_numeric <- function(num_x) {
  ggplot(bike_train)+
    geom_histogram(aes(x=num_x,y=..density..),
                   fill= "red”) +
    geom_density(aes(x=num_x,y=..density..))
}
# analyze the distribution of target variable 'cnt'
univariate_numeric(bike_train$cnt)
```

#for analyzing the distribution of independent variables like ‘temp’, ‘atemp’, ‘hum’, ‘windspeed’ is

```
univariate_numeric(bike_train$temp)
```

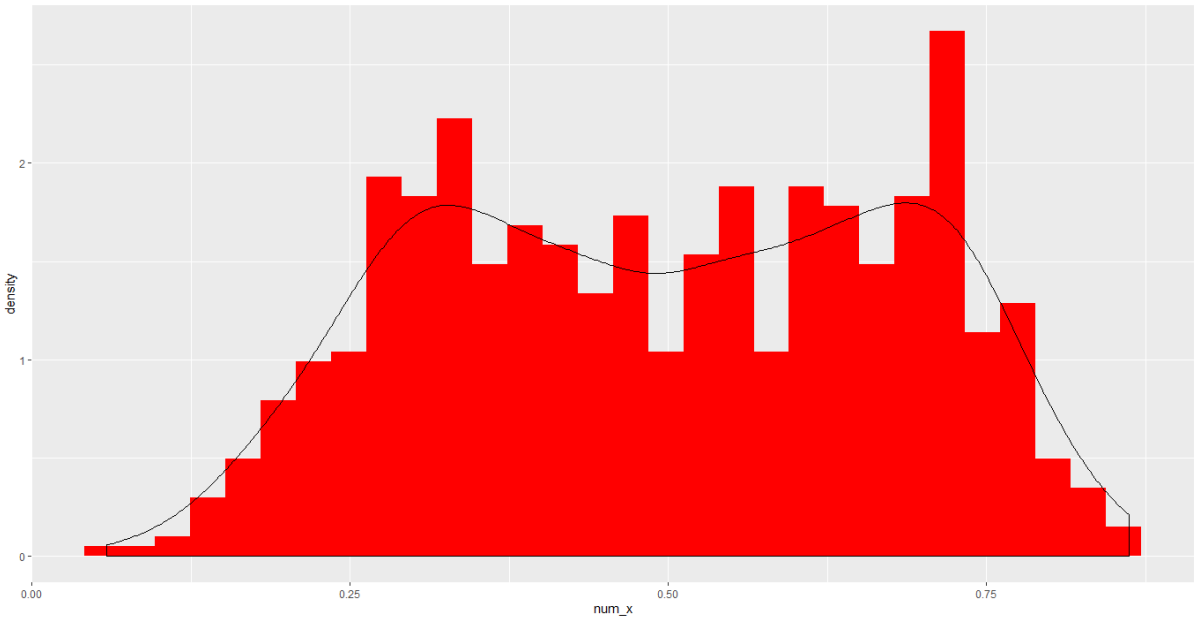


Fig 2.2(a)

```
# analyse the distrubution of independence variable 'atemp'
```

univariate_numeric(bike_train$atemp)

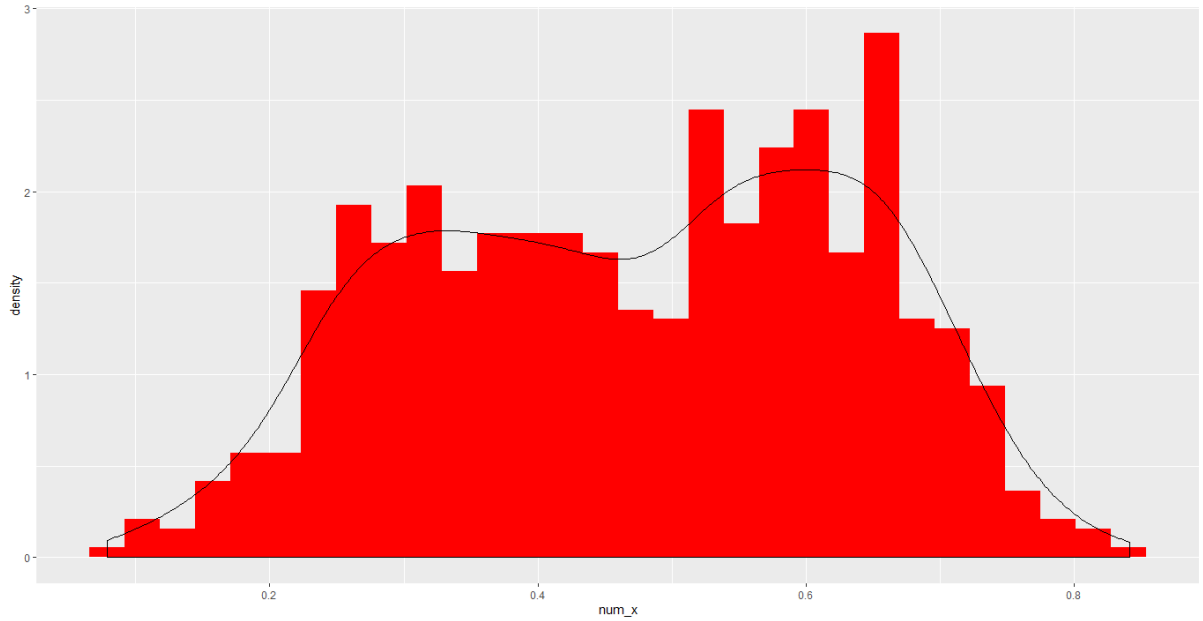


Figure 2.2(b)

analyse the distrubution of independence variable 'hum'

univariate_numeric(bike_train$hum)

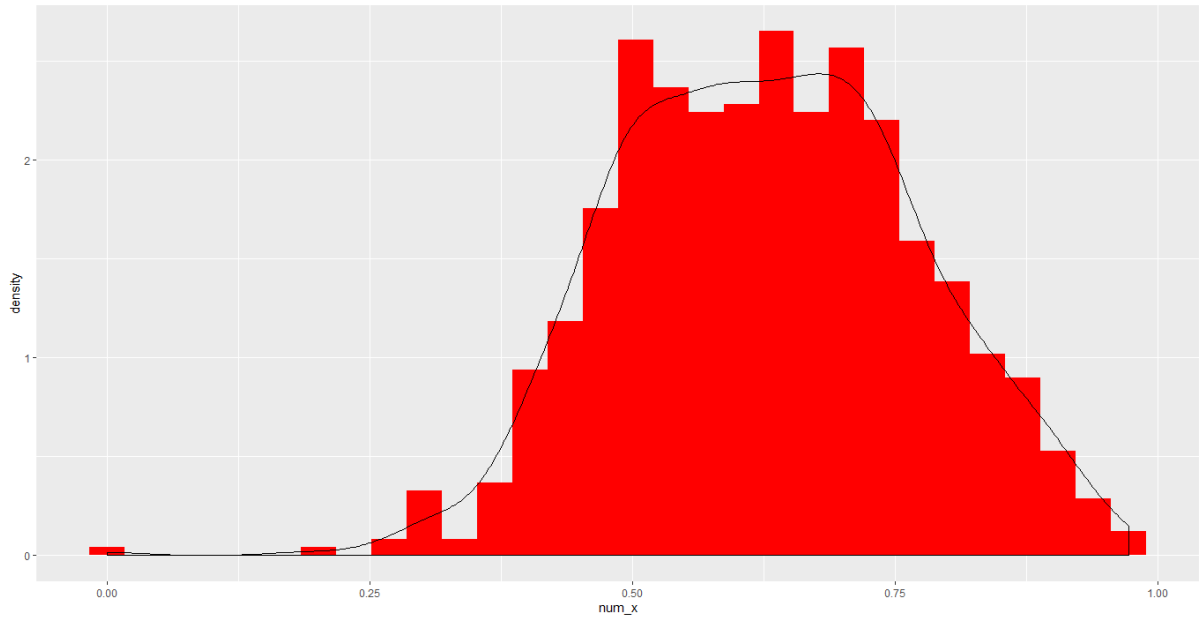


Figure 2.2(c)

analyse the distrubution of independence variable 'windspeed'

univariate_numeric(bike_train$windspeed)

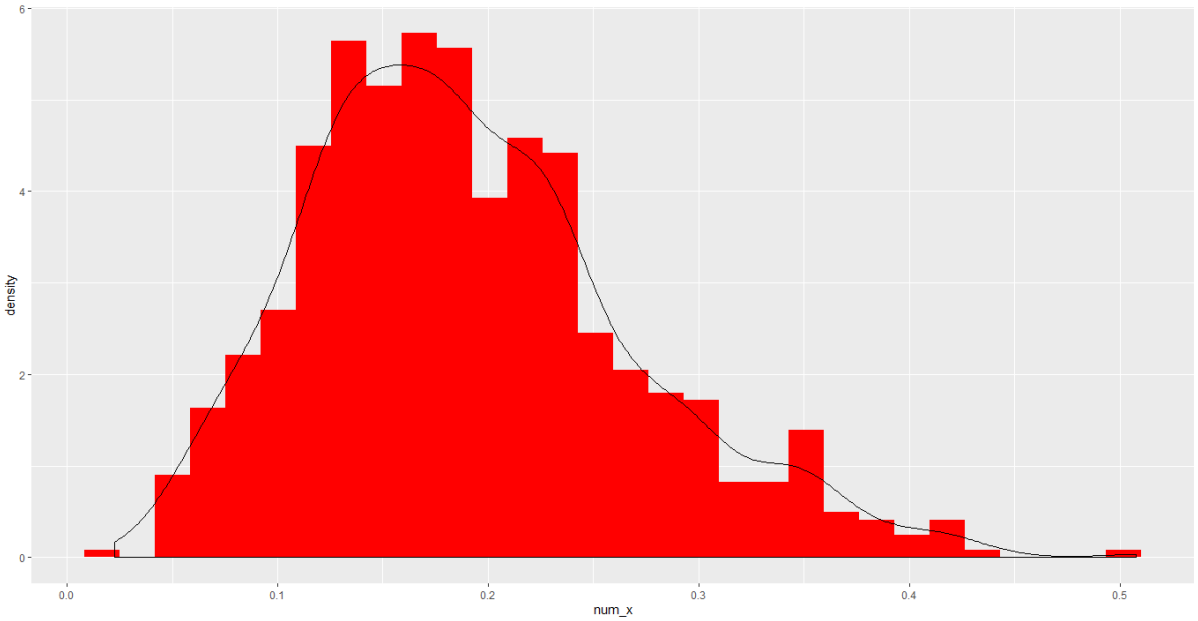


Figure 2.2(d)

the above graph is showing 'cnt' data is normally distributed.

2.1.3) bivariate analysis

Ggpair function built upon ggplot2, GGally provides templates for combining plots into a matrix through the ggpairs function. Such a matrix of plots can be useful for quickly exploring the relationships between multiple columns of data in a data frame.
The lower and upper arguments to the ggpairs function specifies the type of plot or data in each position of the lower or upper diagonal of the matrix, respectively. For continuous X and Y data, one can specify the smooth option to include a regression line.

Below figures shows relationship between independent variables and also with numeric target variable using ggpair:

- Here in graph, the relation between ‘temp’ & ‘atemp’ is very high.
- The relationship between ‘hum’, ‘windspeed’ with target variable is very low.

#check the relationship between 'temp' and 'atemp' variable

```
ggplot(bike_train, aes(x= temp,y=atemp)) +
  geom_point()+
  geom_smooth()
```

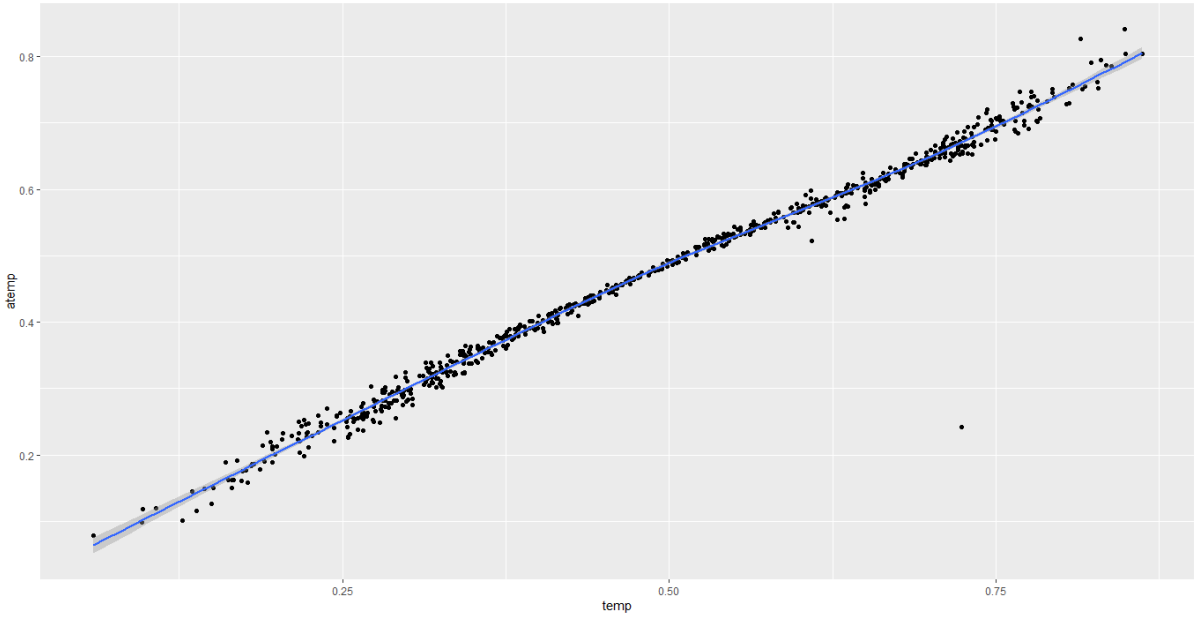


Figure 2.3(a)

#check the relationship between all numeric variable using pair plot

ggpairs(bike_train[,c('atemp','temp','hum','windspeed','cnt')])

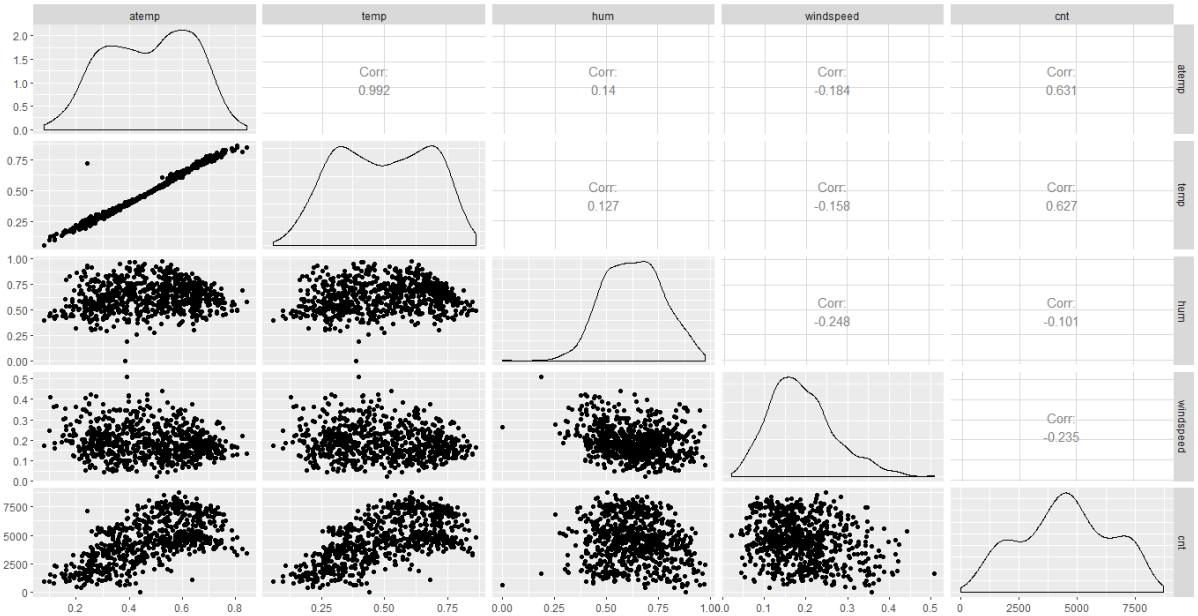


Figure 2.3(b)

that above plot stating that less nagative relationship between

'cnt'-'hum' and cnt-windspeed

and there is strong positive relationship between

temp- cnt and atemp-cnt

2.1.4) Missing value analysis

Missing values in data is a common phenomenon in real world problems. Knowing how to handle missing values effectively is a required step to reduce bias and to produce powerful models.

Below table illustrate no missing value present in the data.

Sl. num	variables	Missing value
1	dteday	0
2	season	0
3	yr	0
4	mnth	0
5	holiday	0
6	weekday	0
7	workingday	0
8	weathersit	0
9	temp	0
10	atemp	0
11	hum	0
12	windspeed	0

There is no missing value found in given dataset.

2.1.5) Outlier Analysis

Outlier analysis is done to handle all inconsistent observations present in given dataset. As outlier analysis can only be done on continuous variable.

Figure 2.1 and 2.2 are visualization of numeric variable present in our dataset to detect outliers using boxplot. Outliers will be detected with red color.

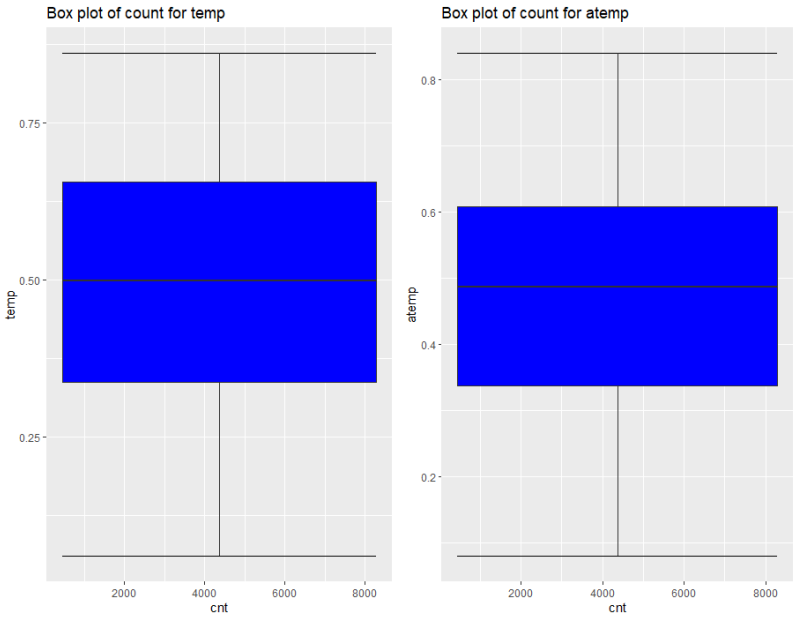


Figure 2.5(a) boxplot graph temp & atemp variables

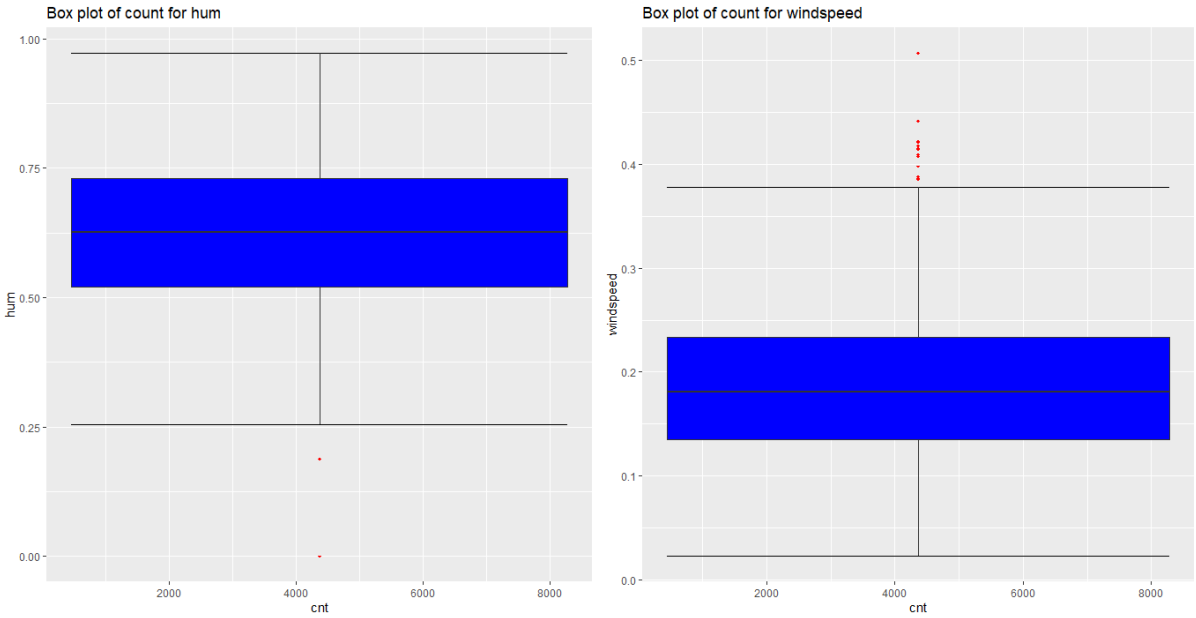


Figure 2.5(b) boxplot graph hum & windspeed

According to above visualizations there is no outlier found in temp and atemp variable but there are few outliers found in windspeed and hum variable.

As windspeed variable defines the windspeed on a particular day and hum defines the humidity of that day so we can neglect these outliers because both these variables define environmental condition. Due to drastic change in weather like storm, heavy rain condition.

2.1.6) Feature selection

Feature selection analysis is done to Select subsets of relevant features (variables, predictors) to be in model construction.

Transforming data using a z-score or t-score. This is usually called standardization. In the vast majority of cases, if a statistics textbook is talking about normalizing data, then this is the definition of “normalization” they are probably using.

Rescaling data to have values between 0 and 1. This is usually called feature scaling. One possible formula to achieve this is.

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

As our target variable is continuous so we can only go for correlation check. As chi-square test is only for categorical variable.

Python code for correlation plot is:

df_corr = bike_train

#Set the width and hieght of the plot

f, ax = plt.subplots(figsize=(7, 5))

#Generate correlation matrix

corr = df_corr.corr()

#Plot using seaborn library

sns.heatmap(corr, mask=np.zeros_like(corr, dtype=np.bool),
cmap=sns.diverging_palette(220, 10, as_cmap=True) square=True, ax=ax);

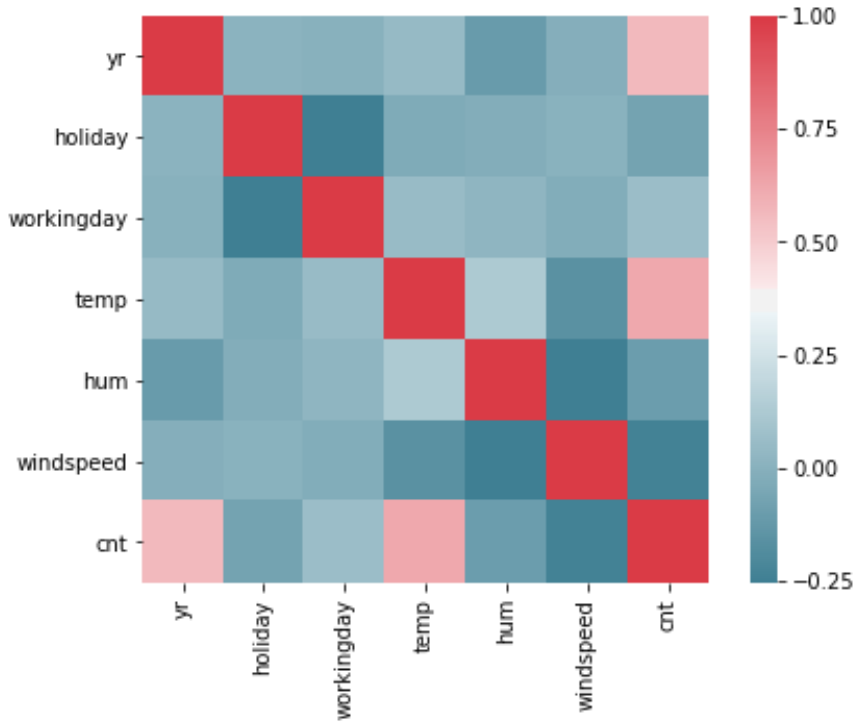


figure 2.6 show a correlation plot for all numeric variable present in dataset.

2.2) Modelling

2.2.1) Model selection

In this case we have to predict the count of bike renting according to environmental and seasonal condition. So the target variable here is a continuous variable. For Continuous we can use various Regression models. Model having less error rate and more accuracy will be our final model.
Models built are
1. c50 (Decision tree for regression target variable)
2. Random Forest (with 200 trees)
3. Linear regression

C50:

This model is also known a Decision tree for regression target variable.

For this model we have divided the dataset into train and test part using random sampling. Where train contains 80% data of data set and test contains 20% data and contains 12 variables where 12th variable is the target variable.

Creating Model In R

```
# ##rpart for regression

fit = rpart(cnt ~ ., data = train, method = "anova")

#Predict for new test cases

predictions_DT = predict(fit, test[,-12])

print(fit)

#  plotting decision tree

par(cex= 0.8)

plot(fit)

text(fit)
```

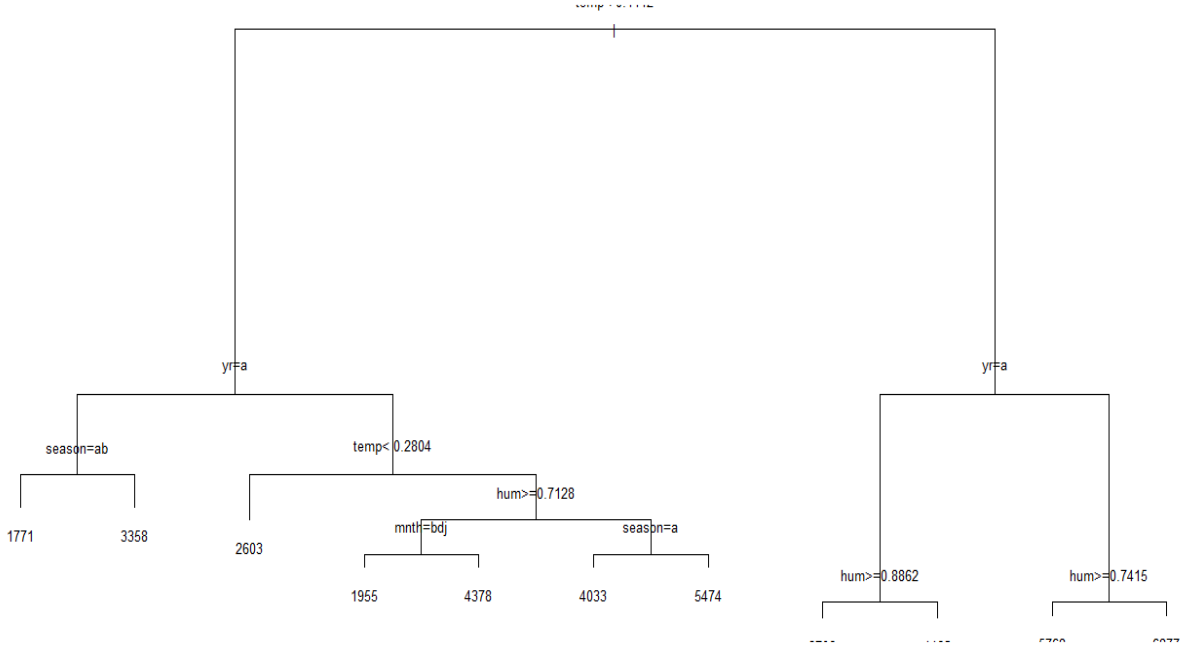


Figure 2.7

Random Forest:

In Random forest we have divided the dataset into train and test part using random sampling. For this model we have divided the dataset into train and test part using random sampling Where train contains 80% data of data set and test contains 20% data and contains 12 variables where 12th variable is the target variable.

Random forest functions in below way

i. Draws a bootstrap sample from training data.
ii. For each sample grow a decision tree and at each node of the tree
    a. Randomly draws a subset of mtry variable and p total of features that are available
    b. Picks the best variable and best split from the subset of mtry variable
    c. Continues until the tree is fully grown.

Code for Random forest is

```
train<- train[,colSums(is.na(train))==0]

Rental_rf=randomForest(cnt ~ . , data = train)

Rental_rf

Plot (Rental_rf)
```

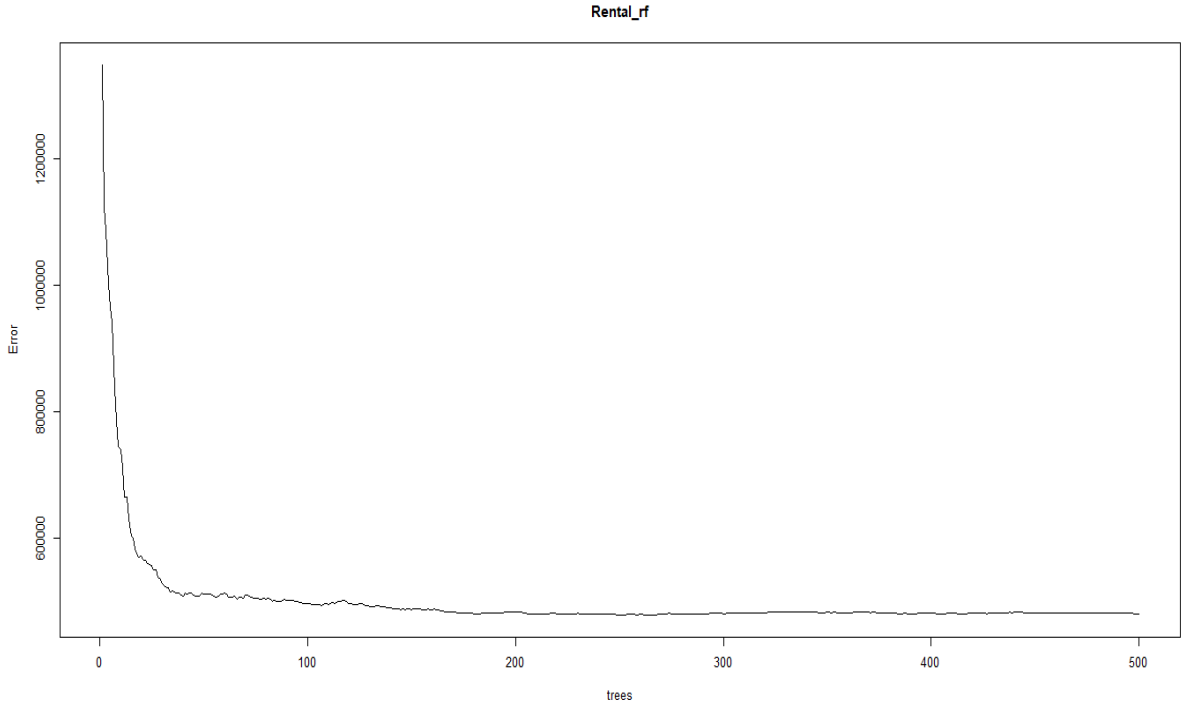


Figure 2.8

Above Figure2.8 represents the curve of error rate as the number of trees increases. After 200 trees the error rate reaches to be constant.
In this model we are using 200 trees to predict the target variable.

Linear regression:

Multiple linear regression is the most common form of linear regression analysis. As a predictive analysis, the multiple linear regression is used to explain the relationship between one continuous dependent variable and two or more independent variables. The independent variables can be continuous or categorical.

VIF (Variance Inflation factor): It quantifies the multicollinearity between the independent variables.

As Linear regression will work well if multicollinearity between the Independent variables are less.

The r code is:

```
vif(train[,-12])

vifcor(train[,-12], th = 0.9)
```

Creating Model

In R

```
#run regression model

lm_model = lm(cnt ~., data = train)

# observe the  residuals and   coefficients  of the linear regression model

# Predict  the Test data

#Predict

predictions_LR= predict(lm_model, test[,1:11])
```

Model summary

```
Call:
lm(formula = cnt ~ ., data = train)

Residuals:
   Min       1Q   Median       3Q      Max
-3534.4   -476.8     48.6    524.8   3038.8

Coefficients: (1 not defined because of singularities)
        Estimate Std. Error t value Pr(>|t|)
```

	estimate	Stand. error	T value	Pr(>|t|)
intercept	20.86			
Dteday	NA	NA	NA	NA
Season	510.14	61.04	8.385	5.09e-16***
Yr	2040.01	73.68	27.688	<2e-16***
Mnth	-33.63	18.94	-1.775	0.0764
Holiday	-595.41	207.53	-2.869	0.00427***
Weekday	60.47	18.56	3.258	0.00119***
Working day	89.89	82.03	1.096	0.27364
Weathersit	-565.35	91.60	-6.171	1.30e-09***
Temp	5111.44	217.76	23.473	<2e-16***
hum	-1168.25	371.32	-3.146	0.00174***
windspeed	-2216.18	544.09	-4.073	5.31e-05***

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

Residual standard error: 868.5 on 559 degrees of freedom
(3 observations deleted due to missingness)
Multiple R-squared: 0.8001, Adjusted R-squared: 0.7966
F-statistic: 223.8 on 10 and 559 DF, p-value: < 2.2e-16

3) Conclusion

3.1) Model evaluation

Now that we have a few models for predicting the target variable, we need to decide which one to choose. There are several criteria that exist for evaluating and comparing models. We can compare the models using any of the following criteria:
1. Predictive Performance
2. Interpretability
3. Computational Efficiency
In our case of Bike Renting, the latter two, *Interpretability* and *Computation Efficiency*, do not hold much significance. Therefore, we will use *Predictive performance* as the criteria to compare and evaluate models.
Predictive performance can be measured by comparing Predictions of the models with real values of the target variables, and calculating some average error measure.

3.1.1) Mean absolute percentage error (MAPE)

```
#defining MAPE function

def MAPE(y_true, y_pred):

    mape = np.mean(np.abs((y_true – y_pred) / y_true))*100

    return mape
```

In above function y_true is the actual value and y_pred is the predicted value.
It will provide the error percentage of model.

MAPE value in Python are as follow

```
#MAPE for decision tree regression

MAPE(test_target_feature, predictions_DT)
```

Error rate: 23.65117086245279

Accuracy: 76.3488291

RMSE: 1059.5692186946853

```
#MAPE for random forest regression

MAPE(test_target_feature,RF_predict)
```

Error rate: 12.807161953851548

Accuracy: 87.1231959

RMSE: 674.1387450178623

#MAPE for linear regression

MAPE(test_target_feature,predict_LR)

Error rate: 19.871597395532454

Accuracy: 80.1284026

RMSE: 926.3465759988094

MAPE value in R as follows

#decision tree

>MAPE(test[,12] , predictions_DT)

21.7133

#RMSE: 908.3552

#random forest, tree=500

>MAPE(test[,12], predictions_RF)

15.52358

#random forest, tree= 200

>MAPE(test[,12],predictions_RF_two)

#RMSE: 612.3454

#linear regression

>MAPE(test[,12],predictions_LR)

19.24258

#RMSE:910.4127

Model Selection (concluded)

As we predicted counts for Bike Rental using three Models Decision Tree, Random Forest and Linear Regression as MAPE is high and RMSE is less for the random forest Model so conclusion is

Conclusion: - For the Bike Rental Data random forest Model is best model to predict the count.

Where predictions_DT are predicted values from C50 model.
predictions_RF are predicted values from random forest model
predictions_LR are predicted values from linear regression model

Relationship between categorical variable & target variable

figure 3.1 Relationship between mnth & cnt

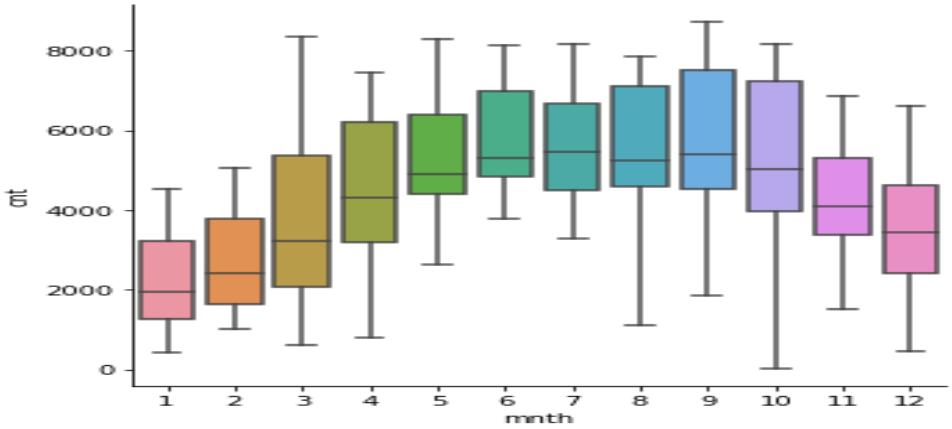


Figure 3.2 Relationship between Weekdays and cnt

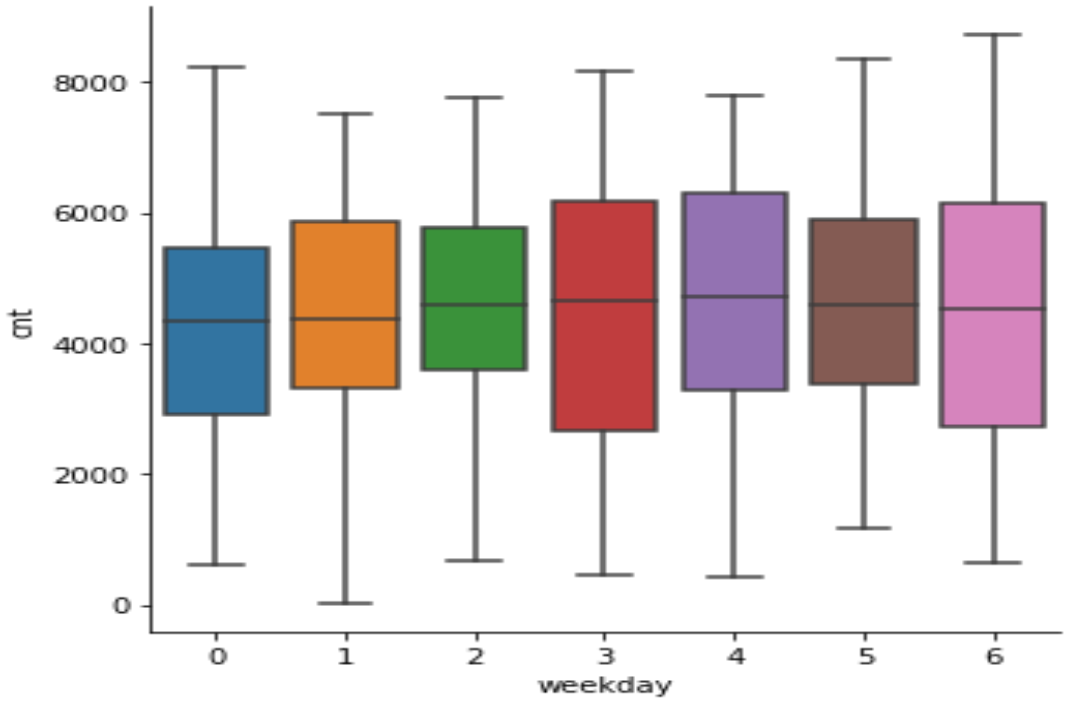


figure 3.2

Bivariate relationship between numeric variables

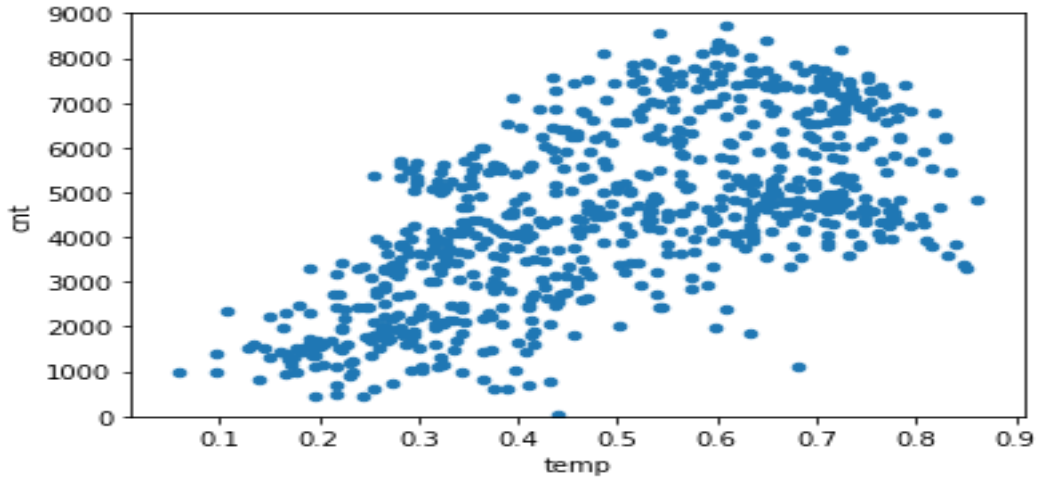
#Now draw scatter plot between 'temp' and 'cnt' variables

var = 'temp'

data = pd.concat([bike_train['cnt'], bike_train[var]], axis=1)

```
data.plot.scatter(x=var, y='cnt', ylim=(0,9000));
```

It is showing there is good relation between 'temp' and 'cnt'




```
#Now draw scatter plot between 'windspeed' and 'cnt' variables
var = 'windspeed'
data = pd.concat([bike_train['cnt'], bike_train[var]], axis=1)
data.plot.scatter(x=var, y='cnt', ylim=(0,9000));
```

It is showing there is nagative relation between 'windspeed' and 'cnt'