

Bike Share Demand Forecast

November 19, 2019

```
[ ]: # ----- Step 1a: Define and categorize the problem statement
    ↪ -----

# The problem statement is to "Predict the daily bike rental count based on the
    ↪ environmental and seasonal settings"
# This is clearly a 'Supervised machine learning regression problem' to predict
    ↪ a number based on the input features.

# ----- Step 1a ends here -----

[1]: # -----Step 1b: Import all the required libraries -----

#---- for data transformations----
#install.packages("lubridate")
library(lubridate)

#---- for EDA Visualizations -----
#install.packages("corrplot")
library(corrplot)
#install.packages("ggplot2")
library(ggplot2)
#install.packages("GGally")
library("GGally")
#install.packages("ggExtra")
library(ggExtra)

#---- for model building----
library(caret)
#install.packages("Metrics")
library(Metrics)
#install.packages("randomForest")
library(randomForest)

# ----- Step 1b ends here -----
```

Attaching package: lubridate

The following object is masked from package:base:

date

corrplot 0.84 loaded

Registered S3 method overwritten by 'GGally':

method from

+.gg ggplot2

Loading required package: lattice

Attaching package: Metrics

The following objects are masked from package:caret:

precision, recall

randomForest 4.6-14

Type rfNews() to see new features/changes/bug fixes.

Attaching package: randomForest

The following object is masked from package:ggplot2:

margin

```
[3]: # ----- Step 2: Gather the data -----  
  
# Data is provided as .csv file and already split into Test and Train.  
# The training set is comprised of the first 19 days of each month, while the  
→test set is the 20th to the end of the month.  
# Let's import the data  
bike= read.csv("/Users/snehashrungarpawar/Documents/Master in Data Science/  
→DPA/Project/Data/train.csv", header=TRUE)  
bike_test = read.csv("/Users/snehashrungarpawar/Documents/Master in Data_  
→Science/DPA/Project/Data/test.csv", header=TRUE)  
# ----- Step 2 ends here -----  
  
[ ]: # ----- Step 3: Data Preparation -----  
  
# 3a. Analyze Attributes: Check properties of data  
# 3b. Complete Data Perform missing value analysis and Impute if needed  
# 3c. Correct Data: Check for any invalid data points  
# 3d. Create Derived Attributes - Feature Extraction  
# 3e. Convert - Converting data to proper formats  
  
[4]: # 3a. Analyze Attributes: Check properties of data  
dim(bike)
```

```

str(bike)
head(bike, 10)
# 3a -> Inference:
#i. The dataset has 10,886 observations (n=10886) and 12 columns of
→type int, num and factor.
#ii. Season, Holiday, Working day and weather are categorical variables.
#ii. temp, atemp, humidity, windspeed, casual, registered and count are
→continuous numerical variables.

```

1. 10886 2. 12

```

'data.frame': 10886 obs. of 12 variables:
 $ datetime : Factor w/ 10886 levels "2011-01-01 00:00:00",...: 1 2 3 4 5 6 7 8
9 10 ...
 $ season : int 1 1 1 1 1 1 1 1 1 1 ...
 $ holiday : int 0 0 0 0 0 0 0 0 0 0 ...
 $ workingday: int 0 0 0 0 0 0 0 0 0 0 ...
 $ weather : int 1 1 1 1 1 2 1 1 1 1 ...
 $ temp : num 9.84 9.02 9.02 9.84 9.84 ...
 $ atemp : num 14.4 13.6 13.6 14.4 14.4 ...
 $ humidity : int 81 80 80 75 75 75 80 86 75 76 ...
 $ windspeed : num 0 0 0 0 0 ...
 $ casual : int 3 8 5 3 0 0 2 1 1 8 ...
 $ registered: int 13 32 27 10 1 1 0 2 7 6 ...
 $ count : int 16 40 32 13 1 1 2 3 8 14 ...

```

	datetime	season	holiday	workingday	weather	temp	atemp	humidity	windspeed
	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	81	0.0000
	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	80	0.0000
	2011-01-01 02:00:00	1	0	0	1	9.02	13.635	80	0.0000
	2011-01-01 03:00:00	1	0	0	1	9.84	14.395	75	0.0000
	2011-01-01 04:00:00	1	0	0	1	9.84	14.395	75	0.0000
	2011-01-01 05:00:00	1	0	0	2	9.84	12.880	75	6.0032
	2011-01-01 06:00:00	1	0	0	1	9.02	13.635	80	0.0000
	2011-01-01 07:00:00	1	0	0	1	8.20	12.880	86	0.0000
	2011-01-01 08:00:00	1	0	0	1	9.84	14.395	75	0.0000
	2011-01-01 09:00:00	1	0	0	1	13.12	17.425	76	0.0000

[5]: # 3b. Complete Data Perform missing value analysis and Impute if needed

```

table(is.na(bike))
# 3b -> Inference: There are no null values in the dataset. If it had, then
→either the rows/columns had to be
# dropped or the null values be imputed based on the % of null values

```

FALSE
130632

```
[ ]: # 3c. Correct Data: Check for any invalid data points
      # From above observations data doesnot seem to have any invalid datatypes
      →to be handled.
      # Let's check for the outliers in EDA step

[6]: # 3d. Create Derived Attributes - Feature Extraction
      # Lets extract 'date', 'month', 'weekday' and 'year' from 'datetime' column
      →as we will be needing it for analysis

      bike$date=as.Date(substr(bike$datetime,1,10))
      bike$year = as.factor(year(bike$datetime))
      bike$month = as.factor(month(bike$datetime))
      bike$hour = as.factor(hour(bike$datetime))
      bike$wkday = as.factor(wday(bike$datetime))

      bike_test$date=as.Date(substr(bike_test$datetime,1,10))
      bike_test$year = as.factor(year(bike_test$datetime))
      bike_test$month = as.factor(month(bike_test$datetime))
      bike_test$hour = as.factor(hour(bike_test$datetime))
      bike_test$wkday = as.factor(wday(bike_test$datetime))

      # Drop datetime as we have extracted all the above needed information
      →from it
      bike = bike[-c(1)]
      bike_test = bike_test[-c(1)]

      head(bike, 5)
      head(bike_test, 5)

      # 3d -> Inference: There are no null values in the dataset. If it had, then
      →either the rows/columns had to be
      #dropped or the null values be imputed based on the % of
      →null values.
```

season	holiday	workingday	weather	temp	atemp	humidity	windspeed	casual	registered
1	0	0	1	9.84	14.395	81	0	3	13
1	0	0	1	9.02	13.635	80	0	8	32
1	0	0	1	9.02	13.635	80	0	5	27
1	0	0	1	9.84	14.395	75	0	3	10
1	0	0	1	9.84	14.395	75	0	0	1
season	holiday	workingday	weather	temp	atemp	humidity	windspeed	date	year
1	0	1	1	10.66	11.365	56	26.0027	2011-01-20	2011
1	0	1	1	10.66	13.635	56	0.0000	2011-01-20	2011
1	0	1	1	10.66	13.635	56	0.0000	2011-01-20	2011
1	0	1	1	10.66	12.880	56	11.0014	2011-01-20	2011
1	0	1	1	10.66	12.880	56	11.0014	2011-01-20	2011

```
[7]: # 3e. Convert - Converting data to proper formats
```

```

# We can clearly see that "season",
→ "yr", "mnth", "holiday", "weekday", "workingday", "weather", "date" are
→ categories, rather than continuous variable.

# Let's convert them to categories
names = c("season", "holiday", "workingday", "weather")
bike[,names] = lapply(bike[,names], factor)
bike_test[,names] = lapply(bike_test[,names], factor)

str(bike)
str(bike_test)

# ----- Step 3: Data Preparation ends here
→ -----

```

```

'data.frame': 10886 obs. of 16 variables:
 $ season      : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
 $ holiday     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ workingday  : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ weather     : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 2 1 1 1 1 ...
 $ temp        : num  9.84 9.02 9.02 9.84 9.84 ...
 $ atemp       : num  14.4 13.6 13.6 14.4 14.4 ...
 $ humidity    : int   81 80 80 75 75 75 80 86 75 76 ...
 $ windspeed   : num   0 0 0 0 0 ...
 $ casual      : int   3 8 5 3 0 0 2 1 1 8 ...
 $ registered  : int  13 32 27 10 1 1 0 2 7 6 ...
 $ count       : int  16 40 32 13 1 1 2 3 8 14 ...
 $ date        : Date, format: "2011-01-01" "2011-01-01" ...
 $ year        : Factor w/ 2 levels "2011","2012": 1 1 1 1 1 1 1 1 1 1 ...
 $ month       : Factor w/ 12 levels "1","2","3","4",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ hour        : Factor w/ 24 levels "0","1","2","3",...: 1 2 3 4 5 6 7 8 9 10 ...
 $ wkday       : Factor w/ 7 levels "1","2","3","4",...: 7 7 7 7 7 7 7 7 7 7 ...
'data.frame': 6493 obs. of 13 variables:
 $ season      : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
 $ holiday     : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
 $ workingday  : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...
 $ weather     : Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 2 ...
 $ temp        : num  10.7 10.7 10.7 10.7 10.7 ...
 $ atemp       : num  11.4 13.6 13.6 12.9 12.9 ...
 $ humidity    : int   56 56 56 56 56 60 60 55 55 52 ...
 $ windspeed   : num  26 0 0 11 11 ...
 $ date        : Date, format: "2011-01-20" "2011-01-20" ...
 $ year        : Factor w/ 2 levels "2011","2012": 1 1 1 1 1 1 1 1 1 1 ...
 $ month       : Factor w/ 12 levels "1","2","3","4",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ hour        : Factor w/ 24 levels "0","1","2","3",...: 1 2 3 4 5 6 7 8 9 10 ...
 $ wkday       : Factor w/ 7 levels "1","2","3","4",...: 5 5 5 5 5 5 5 5 5 5 ...

```

```

[ ]: # ----- Step 4: Exploratory Data Analysis -----
      # 4a. Outlier Analysis

      # 4a(1). Visualize continuos variables wrt target variable

      # 4a(2). Visualize categorical variables wrt target variable

[ ]: # 4b. Correlation Analysis

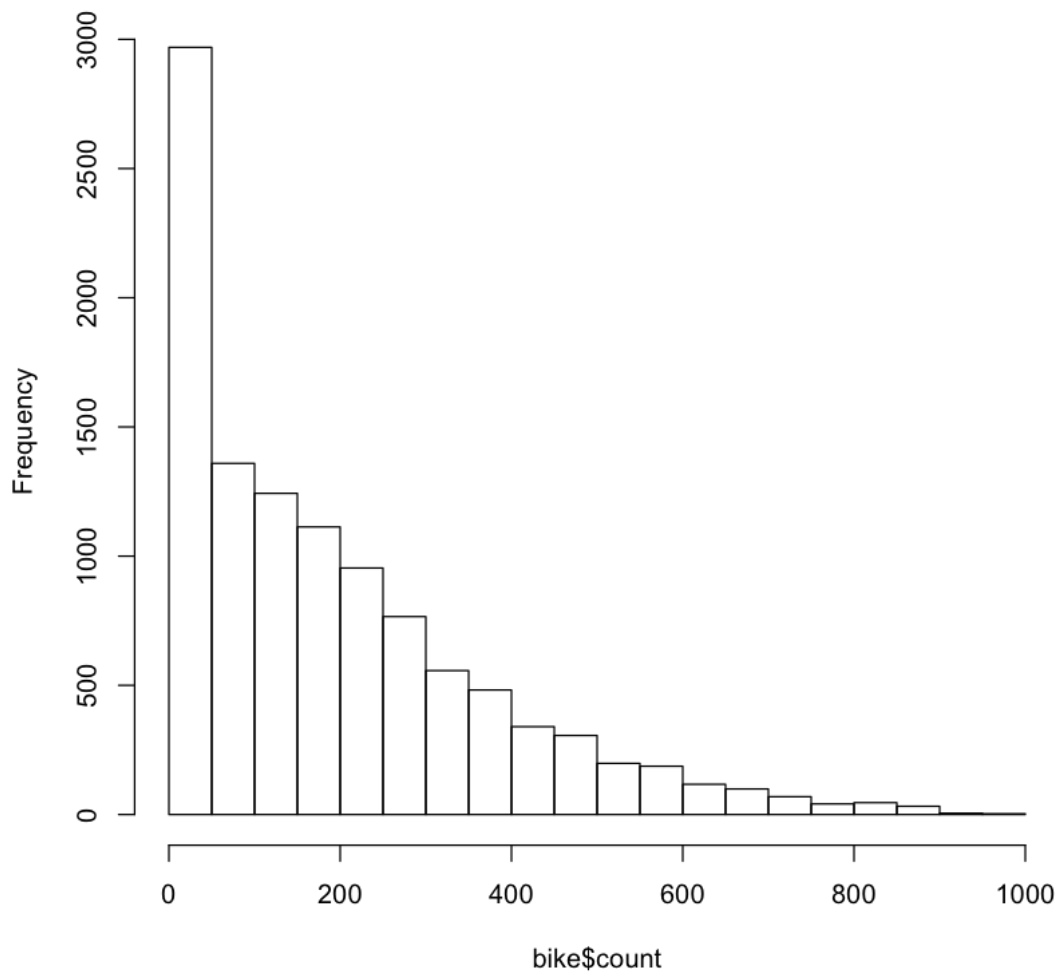
      # ----- Explore Continuous Variables-----

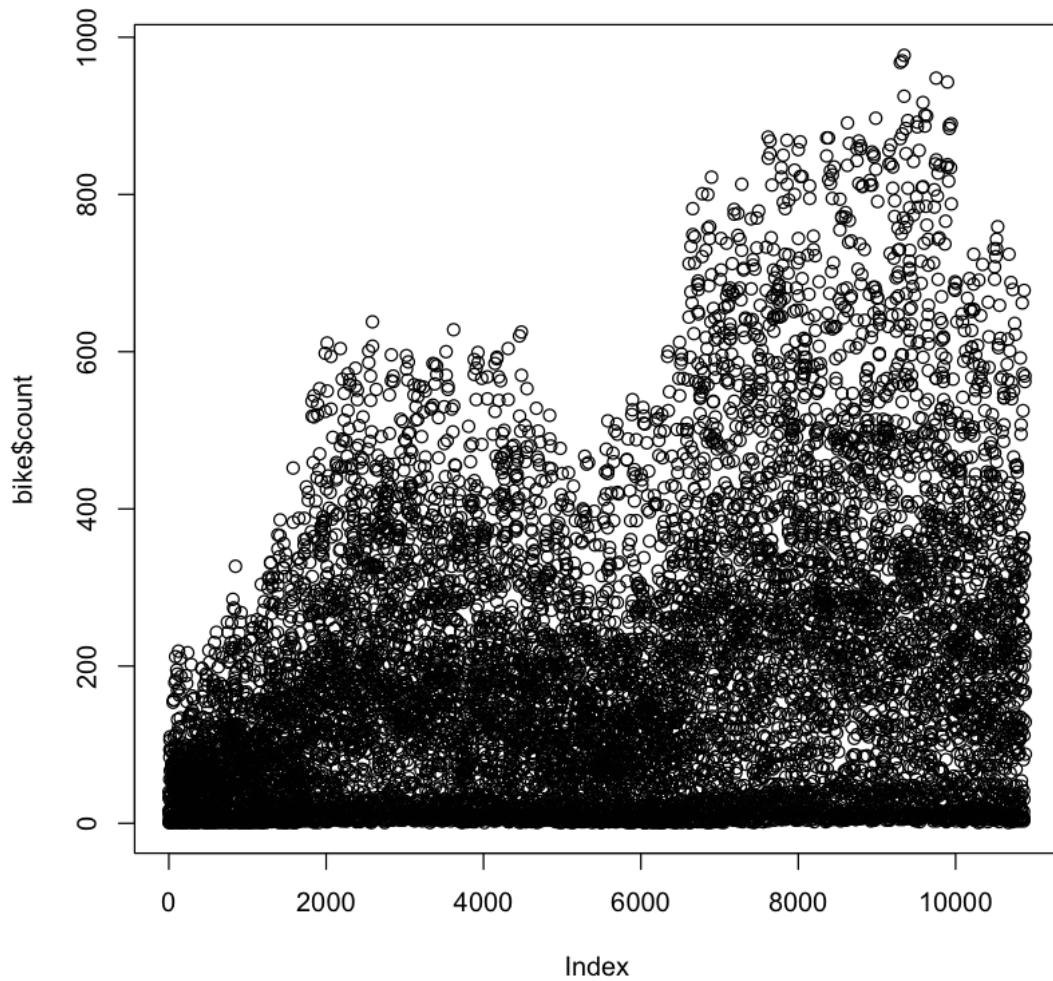
      # 4b(1). Explore continous features
          # i. Check distribution of target variable
          # ii. Explore correlation between independent continuous variables with
      →target variable
          # iii. Plot heatmap for correlation matrix (to check for
      →multicollinearity)
          # iv. Visualize the relationship among all continuous variables using
      →pairplots
          # v. Explore relationship between independent continuous variables and
      →dependent variables using Joint Plot

[8]: # 4b(1) i. Check distribution of target variable
      hist(bike$count)
      plot(bike$count)
      # Inference: Target variable "count" is almost normally distributed.

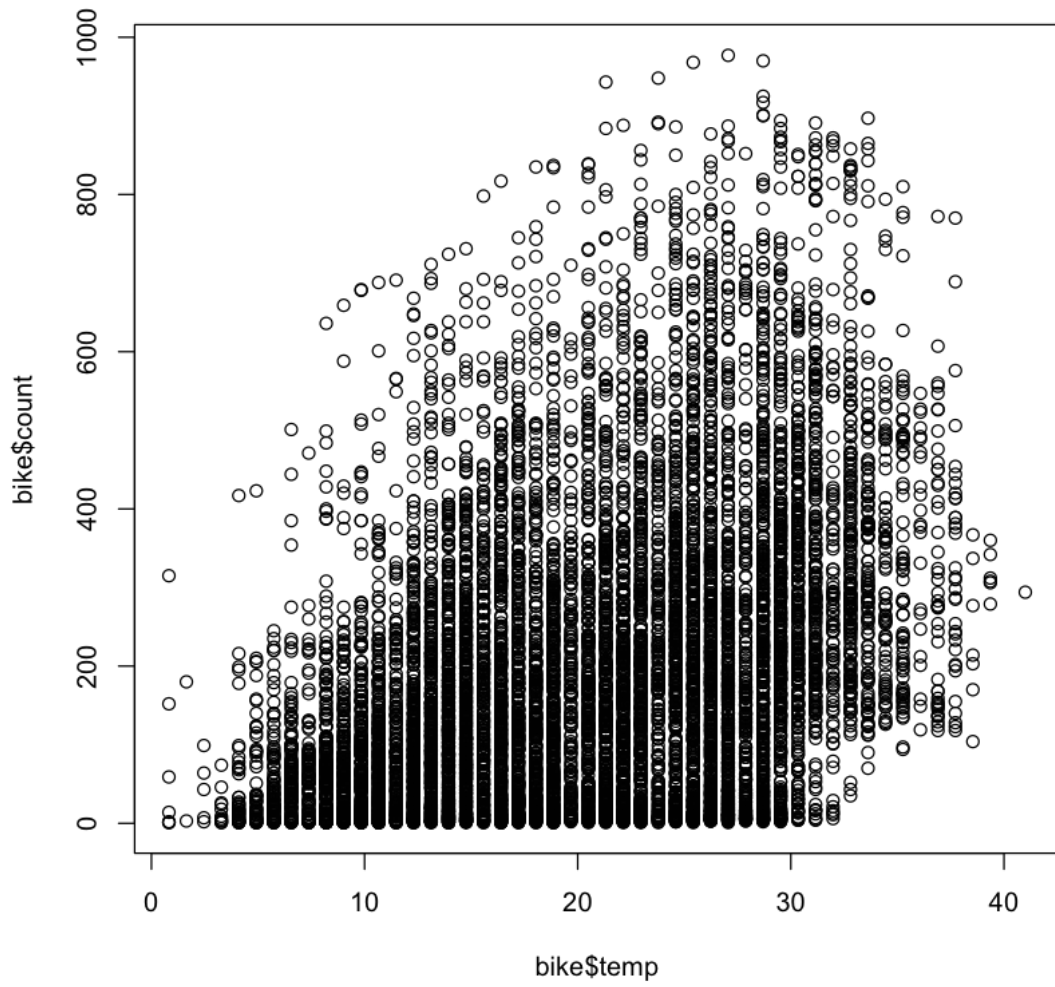
```

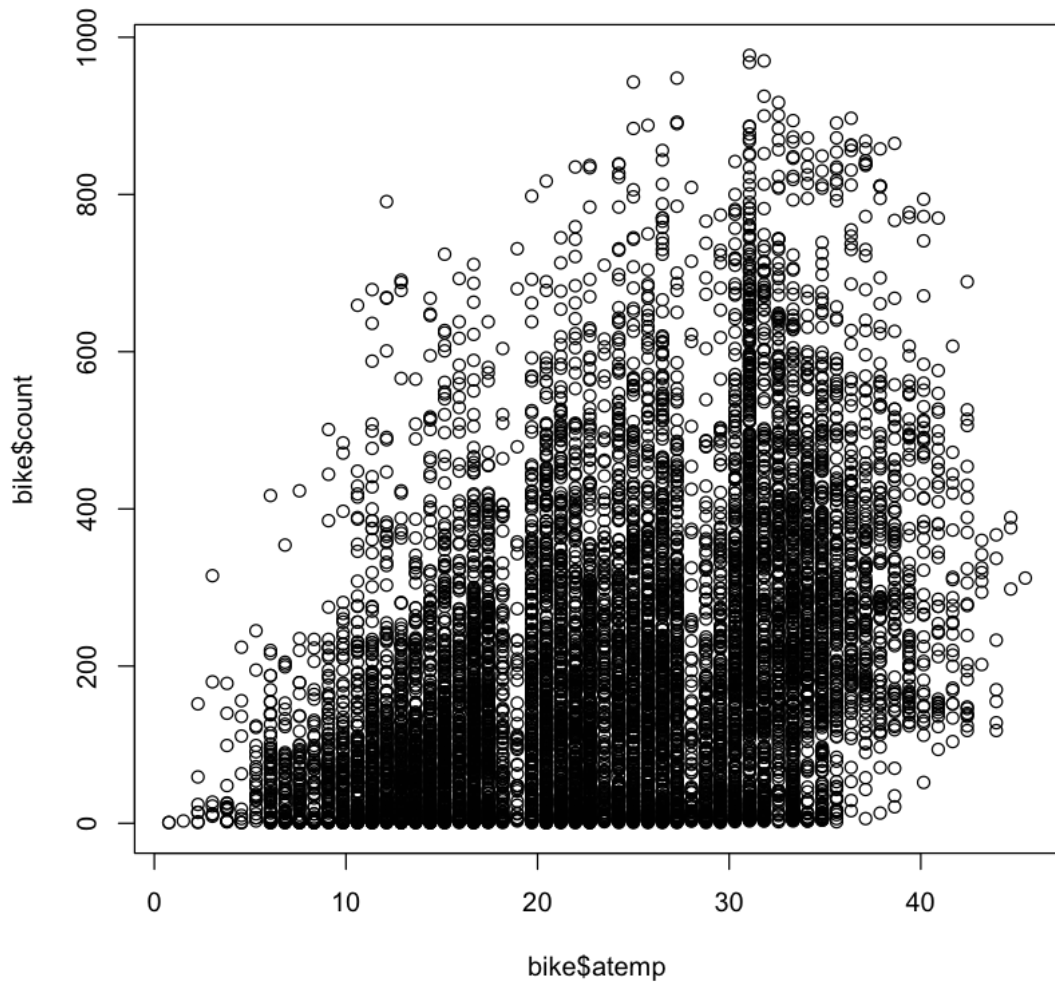
Histogram of bike\$count

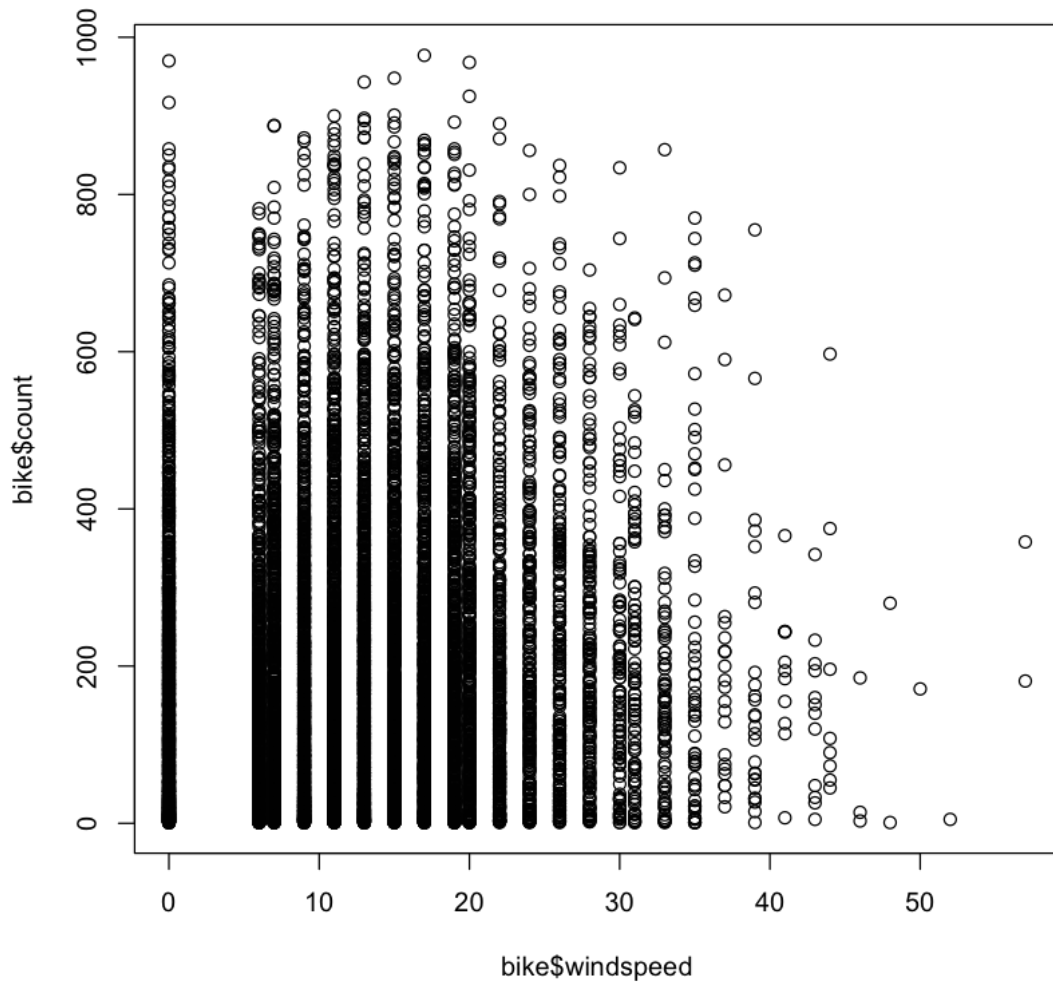


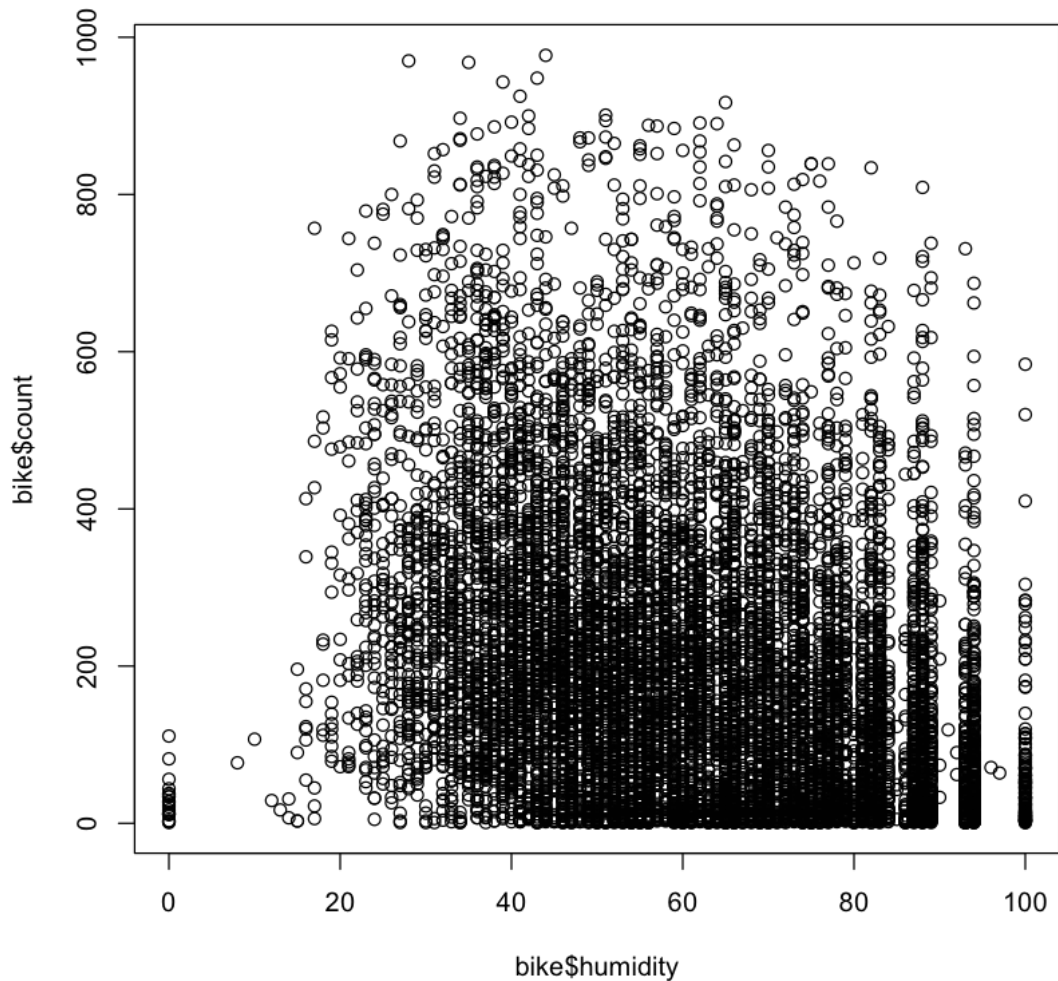


```
[9]: # 4b(1) ii. Explore correlation between independent continuous variables with
      ↪ target variable
      plot(bike$temp,bike$count)
      plot(bike$atemp,bike$count)
      plot(bike$windspeed,bike$count)
      plot(bike$humidity,bike$count)
```

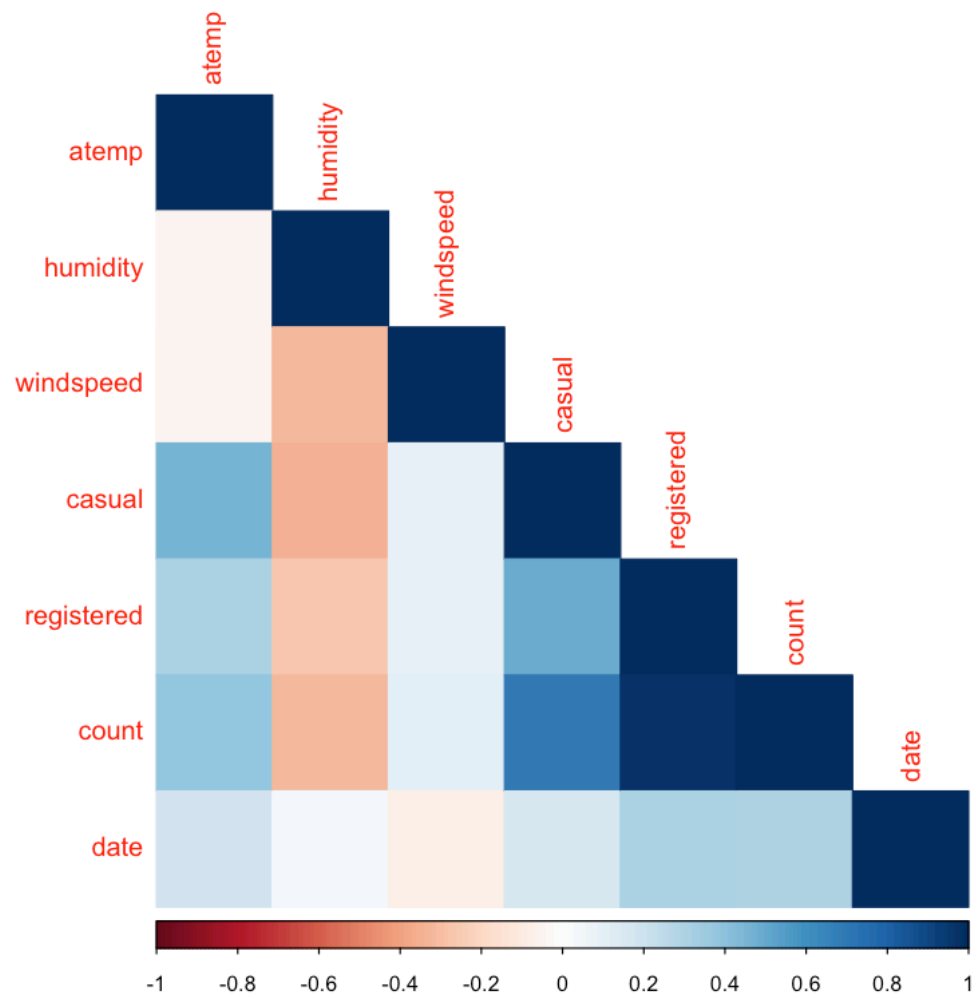





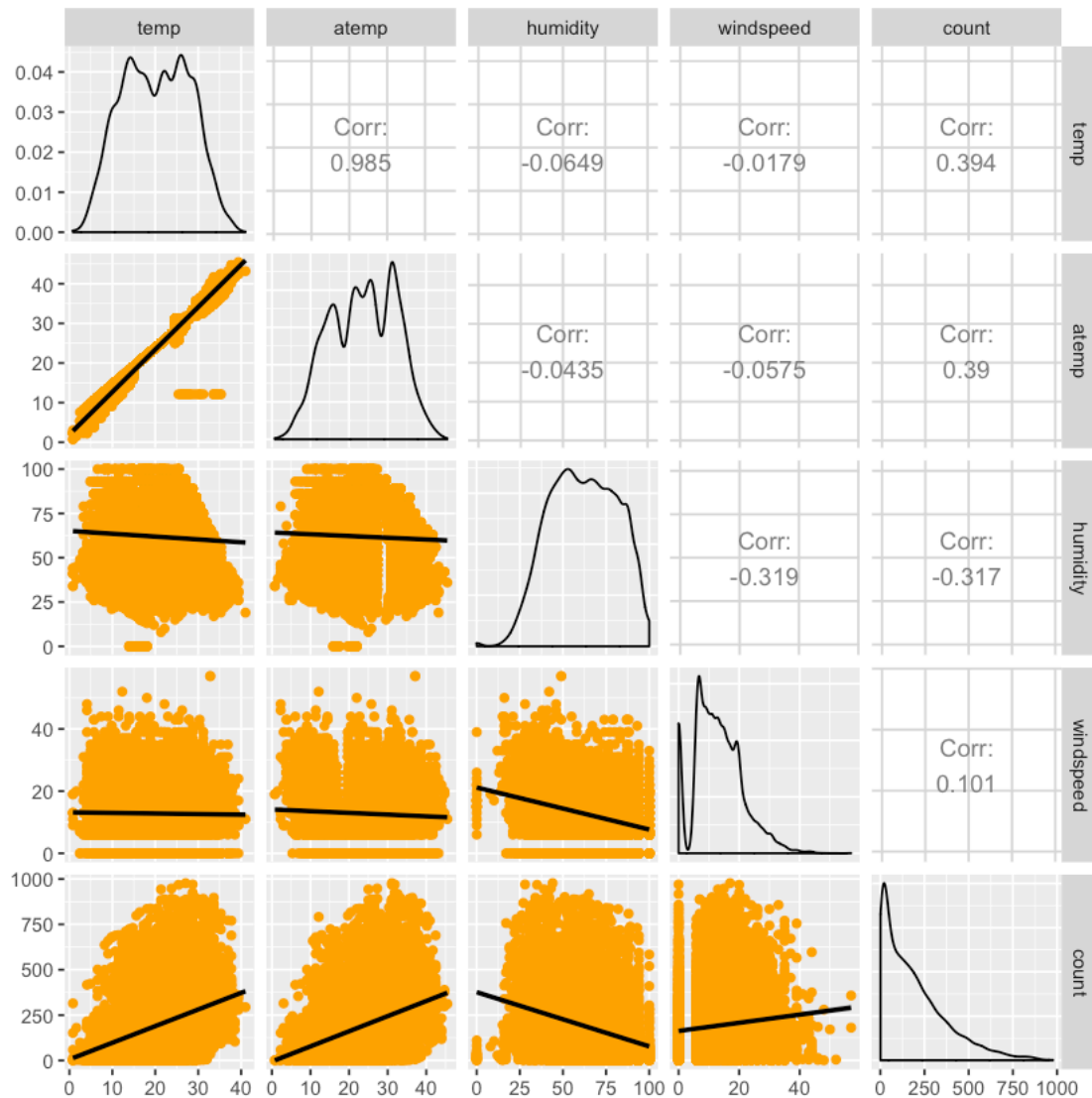




```
[10]: # 4b(1) iii. Plot heatmap for correlation matrix (to check for
      ↪multicollinearity)
      corr <- as.data.frame(lapply(bike[c(6:12)], as.numeric))
      corrplot(cor(corr), method = "color", type='lower')
      # Inference:
      # i. temp and atemp are highly correlated, we would need to drop one of
      ↪them to remove multicollinearity.
      # ii. We can also drop Registered and Casual from our analysis as
      ↪Counts are categorized as Registered and Casual
      # and we will be predicting "Count" variable only.
```

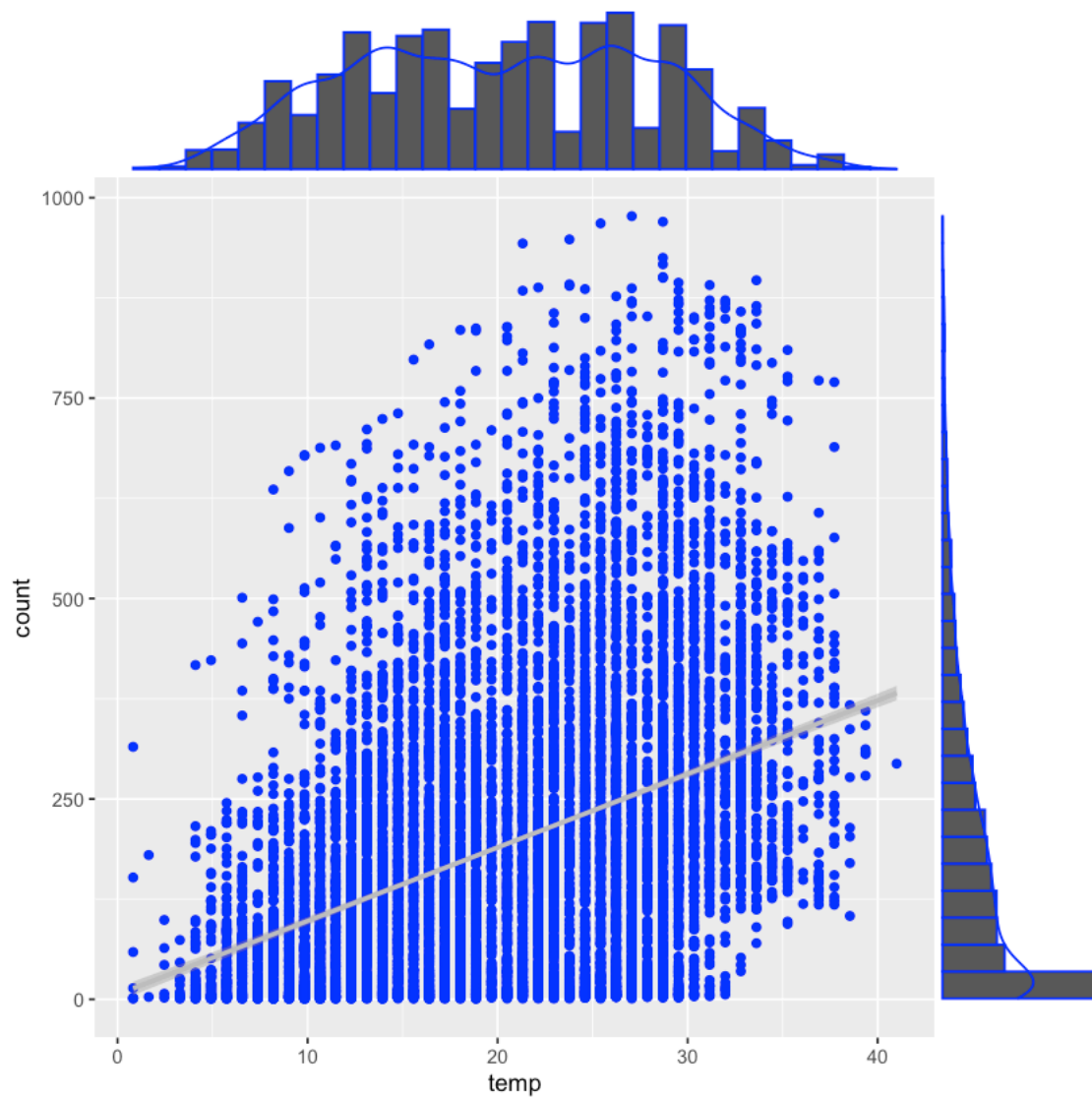


```
[11]: # 4b(1) iv. Visualize the relationship among all continuous variables using
      ↪ pairplots
      ggpairs(bike[c(5:8, 11)], lower=list(continuous=wrap("smooth",
      ↪ colour="orange"))) )
```

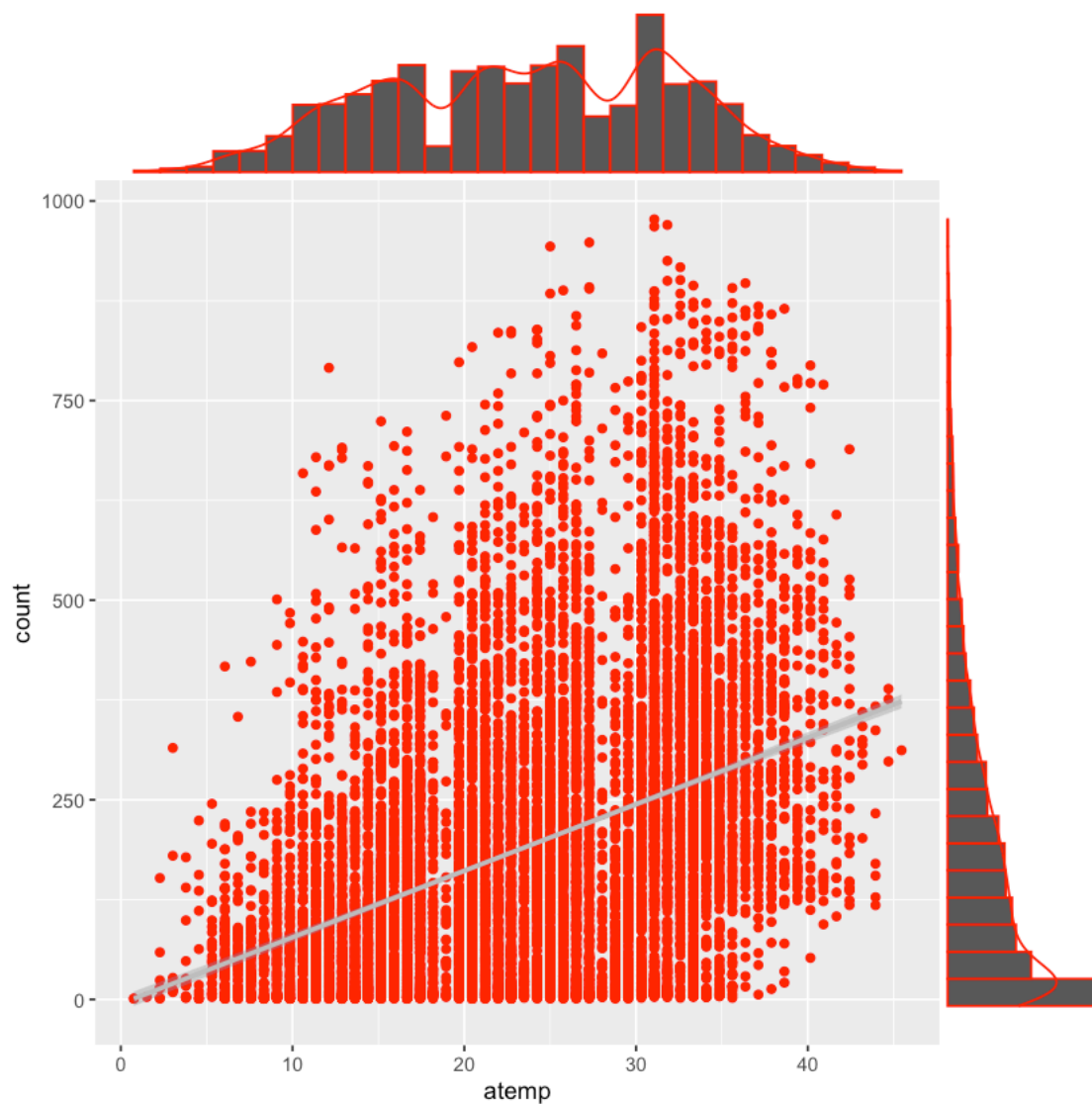


[15]: # 4b(1) v. Explore relationship between independent continuous variables and
 → dependent variables using Joint Plot

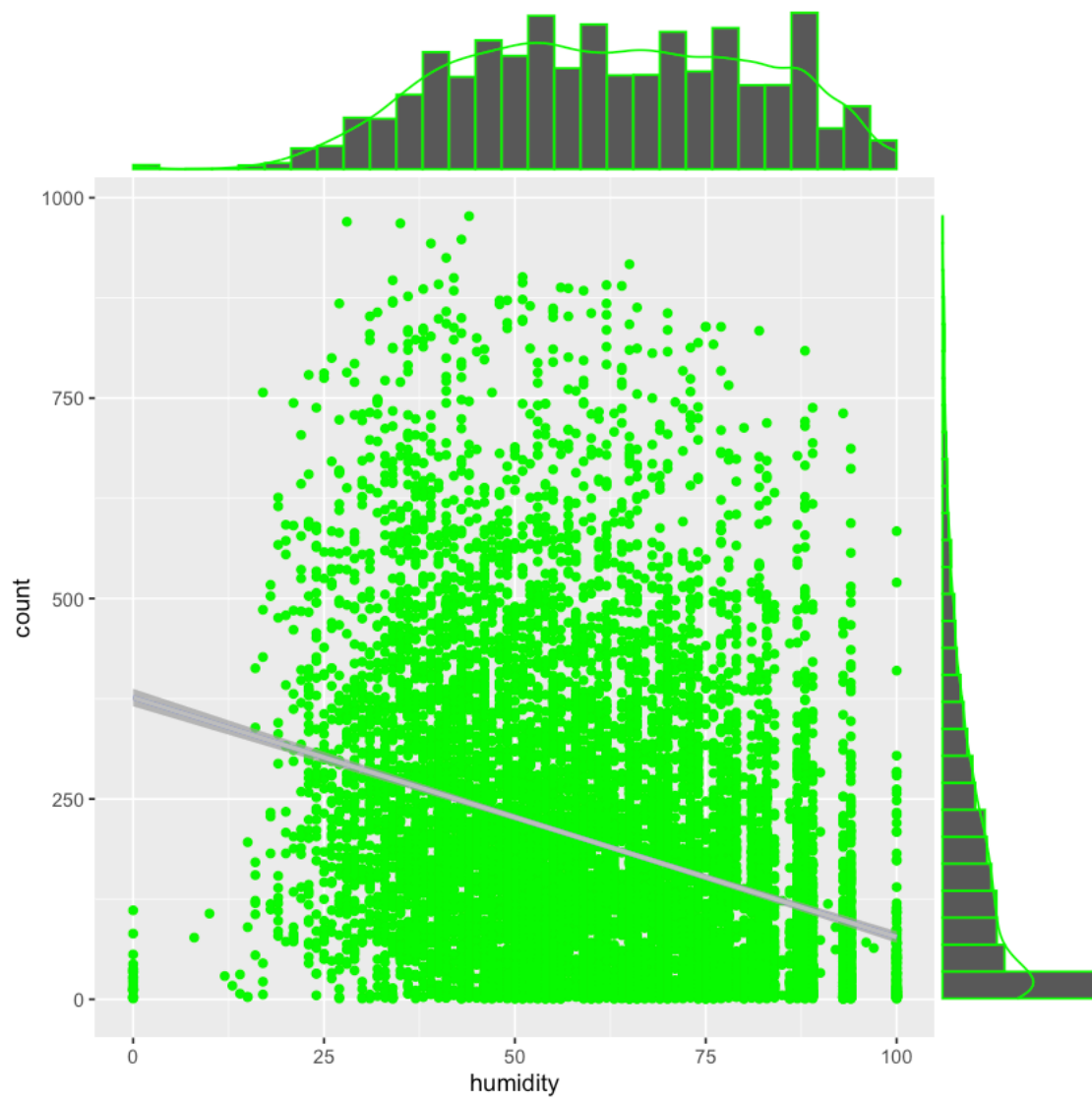
```
# 1. temp vs Count
plot_center = ggplot(bike, aes(x=temp,y=count)) +
  → geom_point(colour="blue") + geom_smooth(method="lm", colour="grey")
  ggMarginal(plot_center, type="densigram", colour="blue")
# Inference: temp has good correlation with count.
```



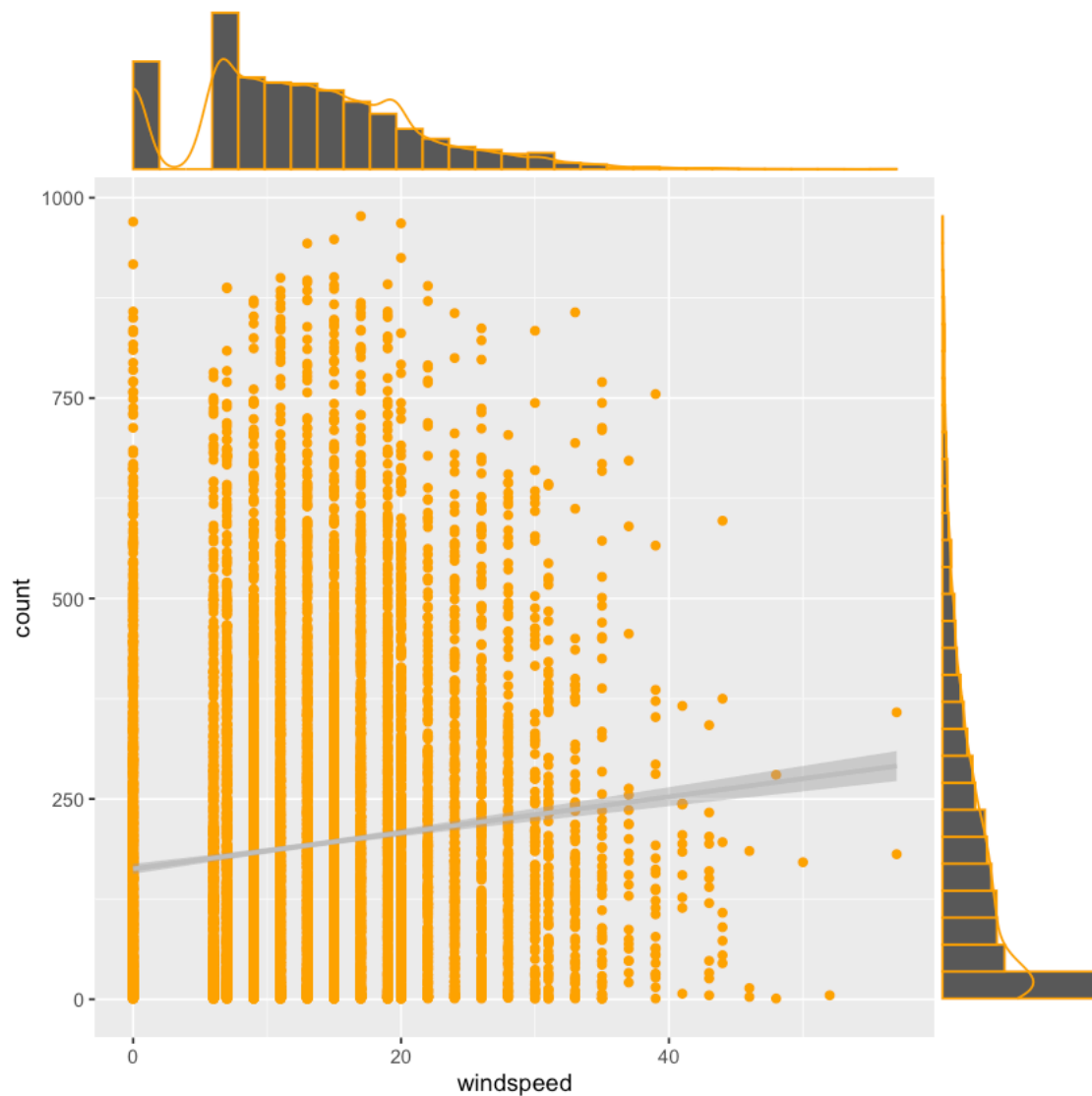
```
[16]: # 4b(1).v.2. atemp vs Count
      plot_center = ggplot(bike, aes(x=atemp,y=count)) +
      ↪geom_point(colour="red") + geom_smooth(method="lm", colour="grey")
      ggMarginal(plot_center, type="densigram", colour="red")
      # Inference: atemp has good correlation with count.
```



```
[17]: # 4b(1).v.3. humidity vs Count
      plot_center = ggplot(bike, aes(x=humidity,y=count)) +
        geom_point(colour="green") + geom_smooth(method="lm") +
        geom_smooth(method="lm", colour="grey")
      ggMarginal(plot_center, type="densigram", colour="green")
      # Inference: Humidity has low correlation with count.
```

```
[18]: # 4b(1).v.4. windspeed vs Count
      plot_center = ggplot(bike, aes(x=windspeed,y=count)) +
      ↪geom_point(colour="orange") + geom_smooth(method="lm", colour="grey")
      ggMarginal(plot_center, type="densigram", colour="orange")
```



```
[ ]: # 4b(1) Inferences Summary - Analysis of continuous variables
      # 1. Target variable 'count' is almost normally distributed.
      # 2. From correlation with dependent variable "count", we can see that
      → 'casual', 'registered' are very
         # highly correlated to cnt. Needs to be dropped from the dataset.
      # 3. 'humidity' has low correlation with 'count'. For now, let's keep it.
      # 4. atemp and temp has good correlation with 'count'
      # 5. From heatmap, we can see that atemp and temp are highly correlated.
      → So we need to drop 1 to remove multicollinearity.
         # 6. Since, as seen from jointplot,  $p(\text{atemp}) < p(\text{temp})$ , we can drop
      → 'temp' and retain 'atemp' in the dataset.
```

```

[ ]: # ----- Explore Catogorical Variables-----
      # 4b(2) Explore categorical features
      # i. Check distribution of categorical variables
      # ii. Check how individual categorical features affects the target
      →variable
      # iii. Explore trends over time

[ ]: # 4c. Drop some variables from the dataset based on the analysis so far
      # drop temp, casual, registered and date
      bike_subset = bike[-c(5,9:10, 12)]
      head(bike_subset,5)

[ ]: #----- Step 4: Exploratory Data Analysis ENDS Here-----
      # Final observations:
      #1.) 'casual' and 'registered' needs to be dropped from the dataset
      #2.) 'atemp' and 'temp' are very strongly correlated . Drop 'atemp' from the
      →dataset (since it has higher p-value
      #than 'temp')
      #3.) 'date' does not seem to have any affect on count of bikes, it can be
      →dropped from the dataset
      #-----

[ ]: #-----Part 5 : Model Builing starts here -----
      # 5a. Split data into test and train set
      # 5b. Linear Regression
      # 5c. Random Forest
      # 5d. Gradient Boosting

[19]: # 5a. Split data into test and train set
      sample_size = floor(0.8 * nrow(bike))
      set.seed(1)
      train_index = sample(nrow(bike), size = sample_size)
      train <- bike[train_index, ]
      test <- bike[-train_index, ]

[22]: # 5b. Linear Regression
      # Fit Linear Model
      train_subset = train[-c(5,9:10, 12)]
      test_subset = test[-c(5,9:10, 12)]

      lm_fit = lm(count ~ ., data = train_subset)
      summary(lm_fit)

      # Choosing the best model by AIC in a Stepwise Algorithm
      # The step() function iteratively removes insignificant features from
      →the model.
      step(lm_fit)
      summary(lm_fit)

```

```

# Calculate Train RMSLE
y_act_train <- abs(train_subset$count)
y_pred_train <- abs(predict(lm_fit, train_subset))
lm_train_RMSLE = rmsle(y_act_train, y_pred_train)

# Calculate Test RMSLE
y_act_test <- abs(test_subset$count)
y_pred_test <- abs(predict(lm_fit, test_subset))
lm_test_RMSLE = rmsle(y_act_test, y_pred_test)

# Save the results
lm_results = predict(lm_fit, bike_test)
hist(lm_results)

```

Call:

```
lm(formula = count ~ ., data = train_subset)
```

Residuals:

Min	1Q	Median	3Q	Max
-354.13	-61.80	-6.72	51.10	432.28

Coefficients: (4 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-88.06071	9.41903	-9.349	< 2e-16 ***
season2	76.02615	7.50253	10.133	< 2e-16 ***
season3	85.24449	7.28896	11.695	< 2e-16 ***
season4	75.84881	5.62763	13.478	< 2e-16 ***
holiday1	15.21998	7.81590	1.947	0.05153 .
workingday1	16.46915	4.06865	4.048	5.21e-05 ***
weather2	-12.18425	2.67532	-4.554	5.33e-06 ***
weather3	-69.93783	4.47891	-15.615	< 2e-16 ***
weather4	-178.26616	100.80800	-1.768	0.07703 .
atemp	4.15523	0.27807	14.943	< 2e-16 ***
humidity	-0.79181	0.07847	-10.091	< 2e-16 ***
windspeed	-0.36893	0.14426	-2.557	0.01056 *
year2012	87.60396	2.18764	40.045	< 2e-16 ***
month2	10.99388	5.39353	2.038	0.04155 *
month3	31.14268	5.73979	5.426	5.93e-08 ***
month4	-23.36018	5.81874	-4.015	6.00e-05 ***
month5	8.92497	5.39782	1.653	0.09828 .
month6	NA	NA	NA	NA
month7	-33.47592	5.52279	-6.061	1.41e-09 ***
month8	-22.39220	5.38767	-4.156	3.27e-05 ***
month9	NA	NA	NA	NA
month10	24.88559	5.68662	4.376	1.22e-05 ***
month11	2.20392	5.35091	0.412	0.68044
month12	NA	NA	NA	NA

hour1	-11.56700	7.48918	-1.544	0.12250	
hour2	-24.20057	7.45303	-3.247	0.00117	**
hour3	-37.65763	7.55672	-4.983	6.37e-07	***
hour4	-38.36125	7.44394	-5.153	2.62e-07	***
hour5	-23.58264	7.47710	-3.154	0.00162	**
hour6	36.34707	7.41526	4.902	9.68e-07	***
hour7	170.36551	7.39766	23.030	< 2e-16	***
hour8	311.27579	7.44319	41.820	< 2e-16	***
hour9	164.66110	7.38386	22.300	< 2e-16	***
hour10	114.20073	7.46609	15.296	< 2e-16	***
hour11	141.94037	7.50231	18.920	< 2e-16	***
hour12	179.11916	7.56096	23.690	< 2e-16	***
hour13	178.36299	7.65620	23.297	< 2e-16	***
hour14	163.86705	7.63461	21.464	< 2e-16	***
hour15	170.24754	7.57631	22.471	< 2e-16	***
hour16	233.35385	7.60976	30.665	< 2e-16	***
hour17	389.19729	7.65071	50.871	< 2e-16	***
hour18	361.87509	7.57348	47.782	< 2e-16	***
hour19	246.72022	7.42580	33.225	< 2e-16	***
hour20	164.41265	7.51298	21.884	< 2e-16	***
hour21	114.40167	7.44445	15.367	< 2e-16	***
hour22	75.69491	7.45131	10.159	< 2e-16	***
hour23	37.52871	7.37358	5.090	3.66e-07	***
wkday2	-12.53471	4.16922	-3.006	0.00265	**
wkday3	-9.06690	4.11349	-2.204	0.02754	*
wkday4	-5.40134	4.11238	-1.313	0.18907	
wkday5	-3.99999	4.07558	-0.981	0.32640	
wkday6	NA	NA	NA	NA	
wkday7	16.55806	3.99635	4.143	3.46e-05	***

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

Residual standard error: 100.5 on 8659 degrees of freedom

Multiple R-squared: 0.6934, Adjusted R-squared: 0.6917

F-statistic: 408 on 48 and 8659 DF, p-value: < 2.2e-16

Start: AIC=80337.58

count ~ season + holiday + workingday + weather + atemp + humidity +
windspeed + year + month + hour + wkday

Step: AIC=80337.58

count ~ season + holiday + weather + atemp + humidity + windspeed +
year + month + hour + wkday

Step: AIC=80337.58

count ~ holiday + weather + atemp + humidity + windspeed + year +
month + hour + wkday

	Df	Sum of Sq	RSS	AIC
- holiday	1	330	87440395	80336
<none>			87440065	80338
- windspeed	1	66045	87506111	80342
- wkday	6	287513	87727579	80354
- humidity	1	1028198	88468263	80437
- atemp	1	2254957	89695022	80557
- weather	3	2485536	89925602	80576
- month	11	5549955	92990020	80851
- year	1	16193442	103633507	81815
- hour	23	102266438	189706503	87036

Step: AIC=80335.61

count ~ weather + atemp + humidity + windspeed + year + month +
hour + wkday

	Df	Sum of Sq	RSS	AIC
<none>			87440395	80336
- windspeed	1	66045	87506441	80340
- wkday	6	291781	87732176	80353
- humidity	1	1028605	88469000	80435
- atemp	1	2254678	89695074	80555
- weather	3	2485209	89925605	80574
- month	11	5553725	92994121	80850
- year	1	16193145	103633541	81813
- hour	23	102271521	189711917	87034

Call:

```
lm(formula = count ~ weather + atemp + humidity + windspeed +  
year + month + hour + wkday, data = train_subset)
```

Coefficients:

(Intercept)	weather2	weather3	weather4	atemp	humidity
-88.1022	-12.1887	-69.9304	-178.0143	4.1544	-0.7919
windspeed	year2012	month2	month3	month4	month5
-0.3689	87.6027	11.0857	31.2433	52.7041	85.0598
month6	month7	month8	month9	month10	month11
76.1375	51.8193	62.9666	85.2886	100.7816	78.0935
month12	hour1	hour2	hour3	hour4	hour5
75.9544	-11.5714	-24.1994	-37.6623	-38.3622	-23.5839
hour6	hour7	hour8	hour9	hour10	hour11
36.3411	170.3596	311.2796	164.6600	114.2076	141.9441
hour12	hour13	hour14	hour15	hour16	hour17
179.1188	178.3688	163.8666	170.2423	233.3501	389.2004

hour18	hour19	hour20	hour21	hour22	hour23
361.8761	246.7257	164.4022	114.4089	75.6880	37.5286
wkday2	wkday3	wkday4	wkday5	wkday6	wkday7
3.7447	7.4038	11.0473	12.4676	16.4273	16.5561

Call:

```
lm(formula = count ~ ., data = train_subset)
```

Residuals:

Min	1Q	Median	3Q	Max
-354.13	-61.80	-6.72	51.10	432.28

Coefficients: (4 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	-88.06071	9.41903	-9.349	< 2e-16 ***
season2	76.02615	7.50253	10.133	< 2e-16 ***
season3	85.24449	7.28896	11.695	< 2e-16 ***
season4	75.84881	5.62763	13.478	< 2e-16 ***
holiday1	15.21998	7.81590	1.947	0.05153 .
workingday1	16.46915	4.06865	4.048	5.21e-05 ***
weather2	-12.18425	2.67532	-4.554	5.33e-06 ***
weather3	-69.93783	4.47891	-15.615	< 2e-16 ***
weather4	-178.26616	100.80800	-1.768	0.07703 .
atemp	4.15523	0.27807	14.943	< 2e-16 ***
humidity	-0.79181	0.07847	-10.091	< 2e-16 ***
windspeed	-0.36893	0.14426	-2.557	0.01056 *
year2012	87.60396	2.18764	40.045	< 2e-16 ***
month2	10.99388	5.39353	2.038	0.04155 *
month3	31.14268	5.73979	5.426	5.93e-08 ***
month4	-23.36018	5.81874	-4.015	6.00e-05 ***
month5	8.92497	5.39782	1.653	0.09828 .
month6	NA	NA	NA	NA
month7	-33.47592	5.52279	-6.061	1.41e-09 ***
month8	-22.39220	5.38767	-4.156	3.27e-05 ***
month9	NA	NA	NA	NA
month10	24.88559	5.68662	4.376	1.22e-05 ***
month11	2.20392	5.35091	0.412	0.68044
month12	NA	NA	NA	NA
hour1	-11.56700	7.48918	-1.544	0.12250
hour2	-24.20057	7.45303	-3.247	0.00117 **
hour3	-37.65763	7.55672	-4.983	6.37e-07 ***
hour4	-38.36125	7.44394	-5.153	2.62e-07 ***
hour5	-23.58264	7.47710	-3.154	0.00162 **
hour6	36.34707	7.41526	4.902	9.68e-07 ***
hour7	170.36551	7.39766	23.030	< 2e-16 ***
hour8	311.27579	7.44319	41.820	< 2e-16 ***

hour9	164.66110	7.38386	22.300	< 2e-16	***
hour10	114.20073	7.46609	15.296	< 2e-16	***
hour11	141.94037	7.50231	18.920	< 2e-16	***
hour12	179.11916	7.56096	23.690	< 2e-16	***
hour13	178.36299	7.65620	23.297	< 2e-16	***
hour14	163.86705	7.63461	21.464	< 2e-16	***
hour15	170.24754	7.57631	22.471	< 2e-16	***
hour16	233.35385	7.60976	30.665	< 2e-16	***
hour17	389.19729	7.65071	50.871	< 2e-16	***
hour18	361.87509	7.57348	47.782	< 2e-16	***
hour19	246.72022	7.42580	33.225	< 2e-16	***
hour20	164.41265	7.51298	21.884	< 2e-16	***
hour21	114.40167	7.44445	15.367	< 2e-16	***
hour22	75.69491	7.45131	10.159	< 2e-16	***
hour23	37.52871	7.37358	5.090	3.66e-07	***
wkday2	-12.53471	4.16922	-3.006	0.00265	**
wkday3	-9.06690	4.11349	-2.204	0.02754	*
wkday4	-5.40134	4.11238	-1.313	0.18907	
wkday5	-3.99999	4.07558	-0.981	0.32640	
wkday6	NA	NA	NA	NA	
wkday7	16.55806	3.99635	4.143	3.46e-05	***

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

Residual standard error: 100.5 on 8659 degrees of freedom

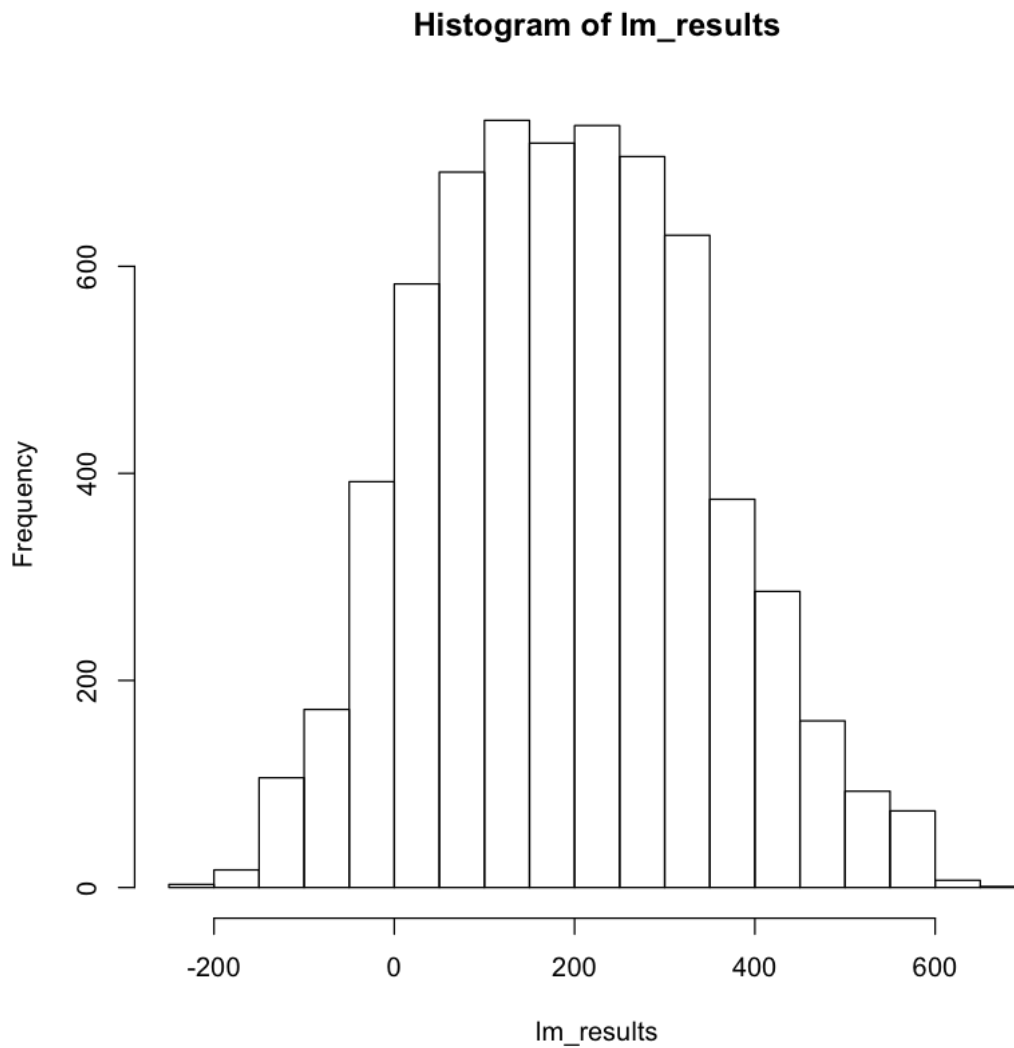
Multiple R-squared: 0.6934, Adjusted R-squared: 0.6917

F-statistic: 408 on 48 and 8659 DF, p-value: < 2.2e-16

Warning message in predict.lm(lm_fit, test_subset):

prediction from a rank-deficient fit may be misleadingWarning message in

predict.lm(lm_fit, bike_test):



```
[24]: # 5b. Random Forest
      Ntree=500
      Mtry = 5
      myImportance = TRUE

      # Predict Casual Counts
      set.seed(1)
      CasualData <- subset(train, select = -c(count, registered, date))
      CasualFit <- randomForest(casual ~ ., data=CasualData, ntree=Ntree,
      ↪mtry=Mtry,
                                importance=myImportance)
```

```

# Predict Registered Counts
RegisteredData <- subset(train, select = -c(count, casual, date))
RegisteredFit <- randomForest(registered ~ ., data=RegisteredData,
  ↪ntree=Ntree, mtry=Mtry,
                                importance=myImportance)

```

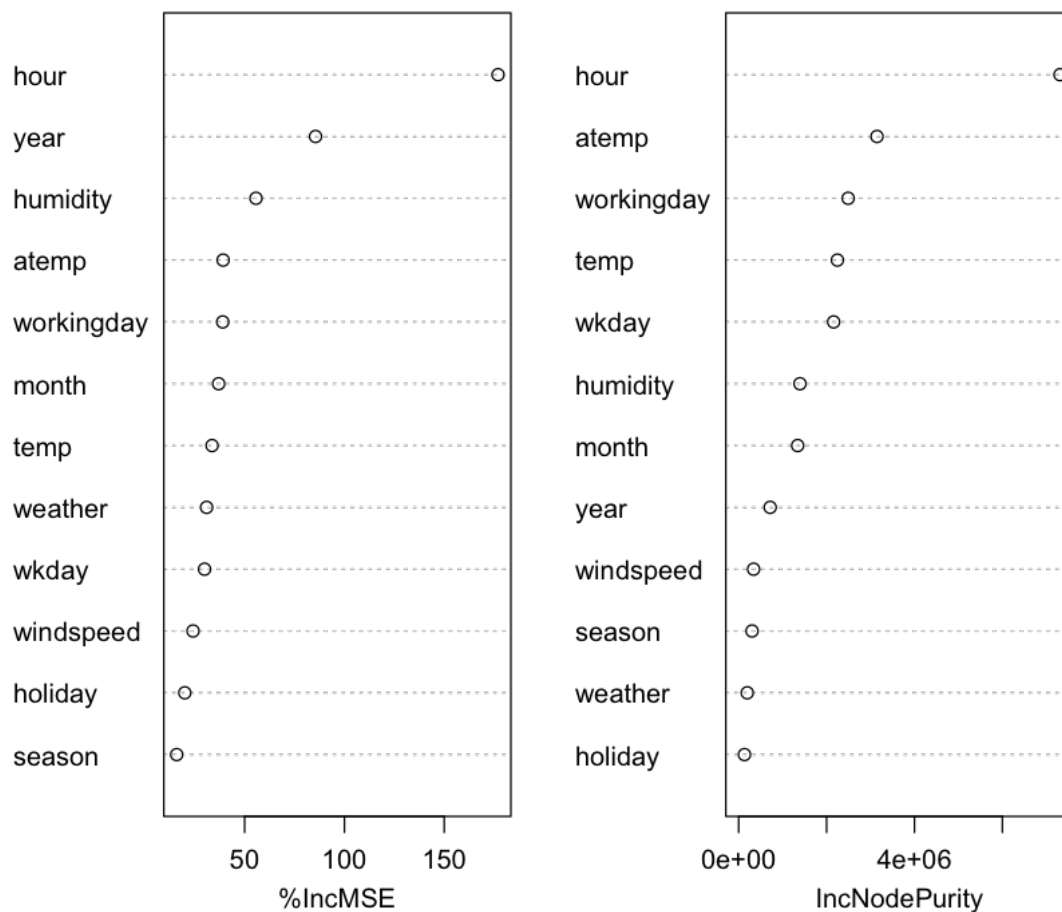
```

[25]: varImpPlot(CasualFit)
      varImpPlot(RegisteredFit)

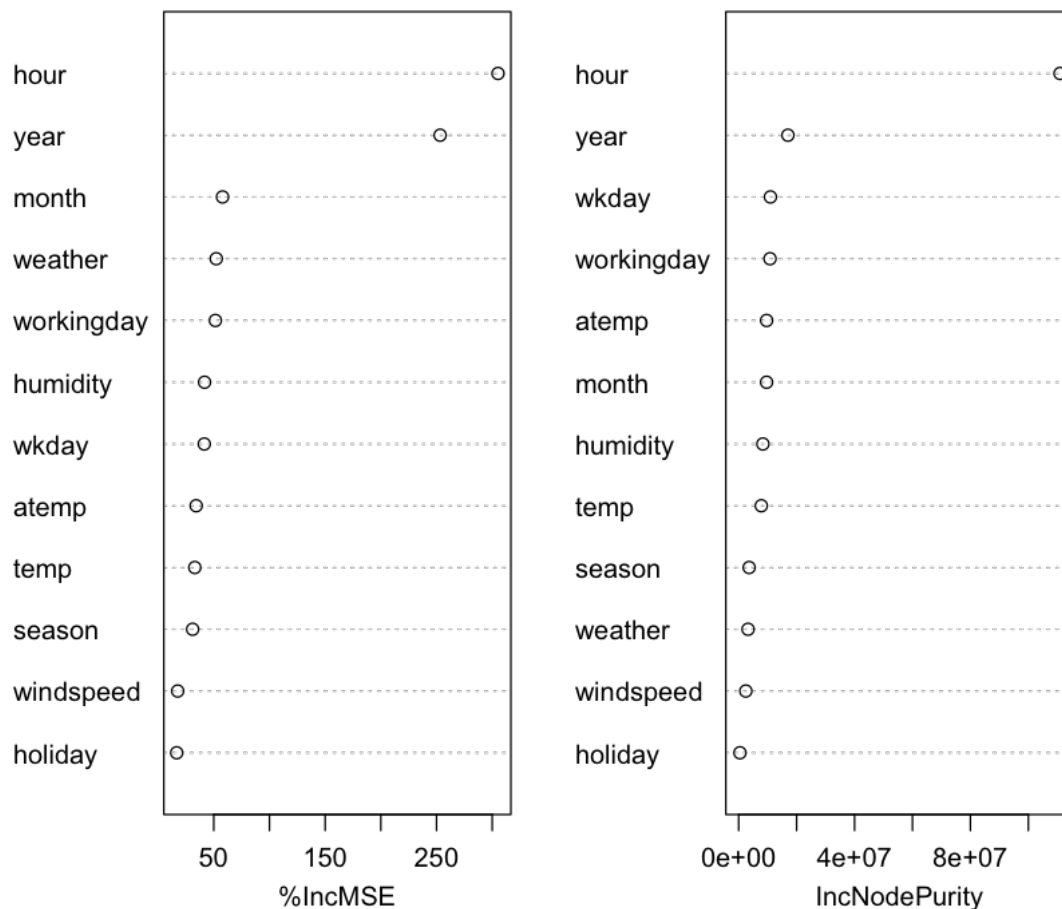
#Inference - Casual Fit: season, holiday, windspeed and weather are not
↪much significant here.
#Inference - Registered Fit: season, holiday, windspeed and temp are not
↪much significant here.

```

CasualFit



RegisteredFit



```
[26]: casualFitFinal <- randomForest(casual ~ hour + year + humidity + month + temp +
  ↳ atemp + workingday + wkday,
                                     data=CasualData, ntree=Ntree,
  ↳ mtry=Mtry, importance=myImportance)
  RegisteredFitFinal <- randomForest(registered ~ hour + year + month +
  ↳ weather + workingday + humidity + atemp
                                     + wkday, data=RegisteredData,
  ↳ ntree=Ntree, mtry=Mtry, importance=myImportance)
```

```
[46]: # Prediction on train data

      # Prediction on train data - casual users
      PredTrainCasual = round(predict(CasualFit, train),0)
      PredTrainCasualFinal = round(predict(casualFitFinal, train),0)
```

```

# Prediction on train data - Registered users
PredTrainRegistered = round(predict(RegisteredFit, train),0)
PredTrainRegisteredFinal = round(predict(RegisteredFitFinal,
→train),0)

# Sum up Casual and Registered to get Total Count
PredTrainCount = PredTrainCasual+PredTrainRegistered
PredTrainCountFinal = PredTrainCasualFinal+PredTrainRegisteredFinal

# Calculate Train RMSLE
rf_train_rmsle_full = rmsle(train$count, PredTrainCount)
rf_train_rmsle2_reduced = rmsle(train$count, PredTrainCountFinal)

# Prediction on test data
# Prediction on test data - casual users
PredTestCasual = round(predict(CasualFit, test),0)
PredTestCasualFinal = round(predict(casualFitFinal, test),0)

# Prediction on test data - registered users
PredTestRegistered = round(predict(RegisteredFit, test),0)
PredTestRegisteredFinal = round(predict(RegisteredFitFinal, test),0)

# Sum up Casual and Registered to get Total Count
PredTestCount = PredTestCasual+PredTestRegistered
PredTestCountFinal = PredTestCasualFinal+PredTestRegisteredFinal

# Calculate Train RMSLE
rf_test_rmsle_full = rmsle(test$count, PredTestCount)
rf_test_rmsle2_reduced = rmsle(test$count, PredTestCountFinal)

```

```

[42]: cat("Training RMSLE - Linear Regression: ", lm_train_RMSLE)
cat("\nTraining RMSLE - Random Forest (Full Model): ", rf_train_rmsle_full)
cat("\nTraining RMSLE - Random Forest (Reduced Model): : ",
→rf_train_rmsle2_reduced)

cat("\n\nTest RMSLE - Linear Regression: ", lm_test_RMSLE)
cat("\nTest RMSLE - Random Forest (Full Model): ", rf_test_rmsle_full)
cat("\nTest RMSLE - Random Forest (Reduced Model): ", rf_test_rmsle2_reduced)

```

```

Training RMSLE - Linear Regression: 1.031166
Training RMSLE - Random Forest (Full Model): 0.2706077
Training RMSLE - Random Forest (Reduced Model): : 0.2091858

```

```

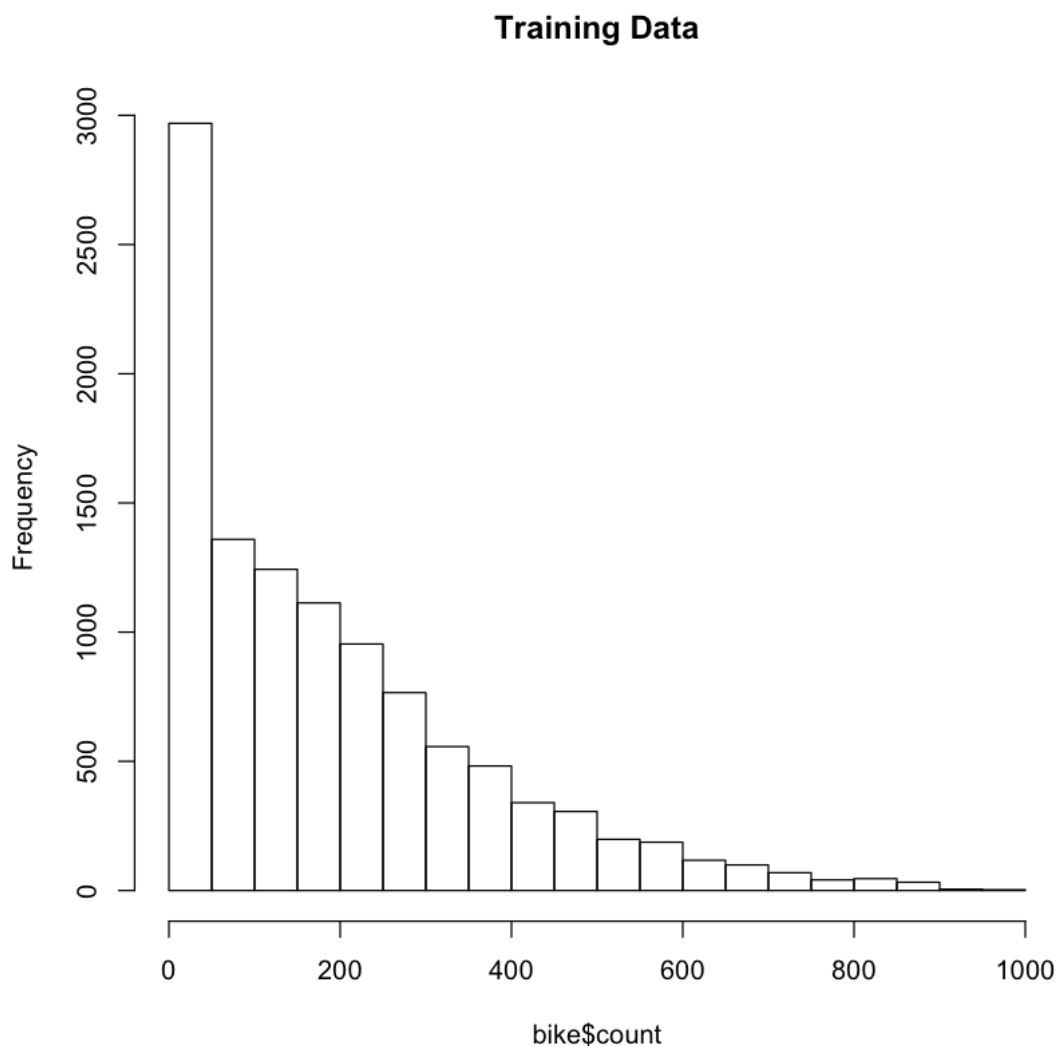
Test RMSLE - Linear Regression: 0.9982687
Test RMSLE - Random Forest (Full Model): 0.4446371

```

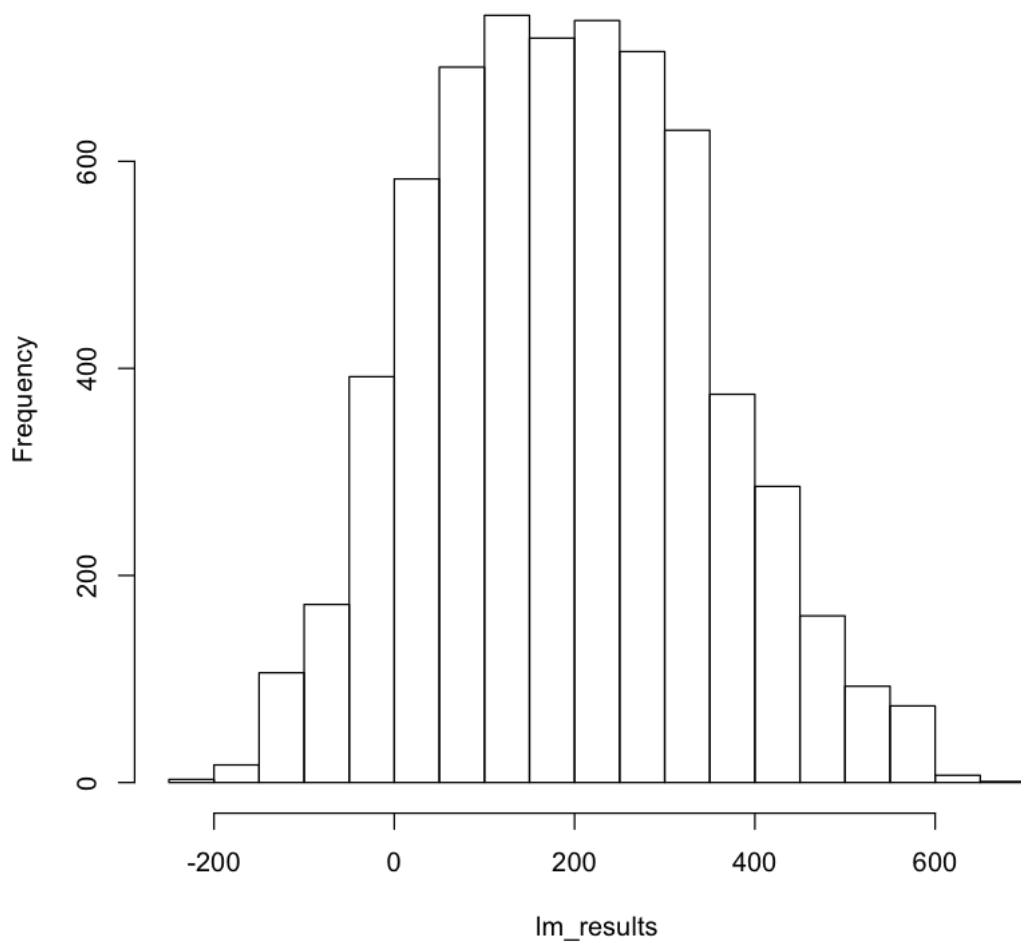
```
[50]: # Save the RF results
      #rf_test_casual = round(predict(casualFitFinal, bike_test),0)
      #rf_test_registered = round(predict(RegisteredFitFinal, bike_test),)
      #rf_results = rf_test_casual + rf_test_registered

      hist(bike$count, main="Training Data")
      hist(lm_results, main="Linear Regression Fit")
      hist(rf_results, main="Random Forest Fit")

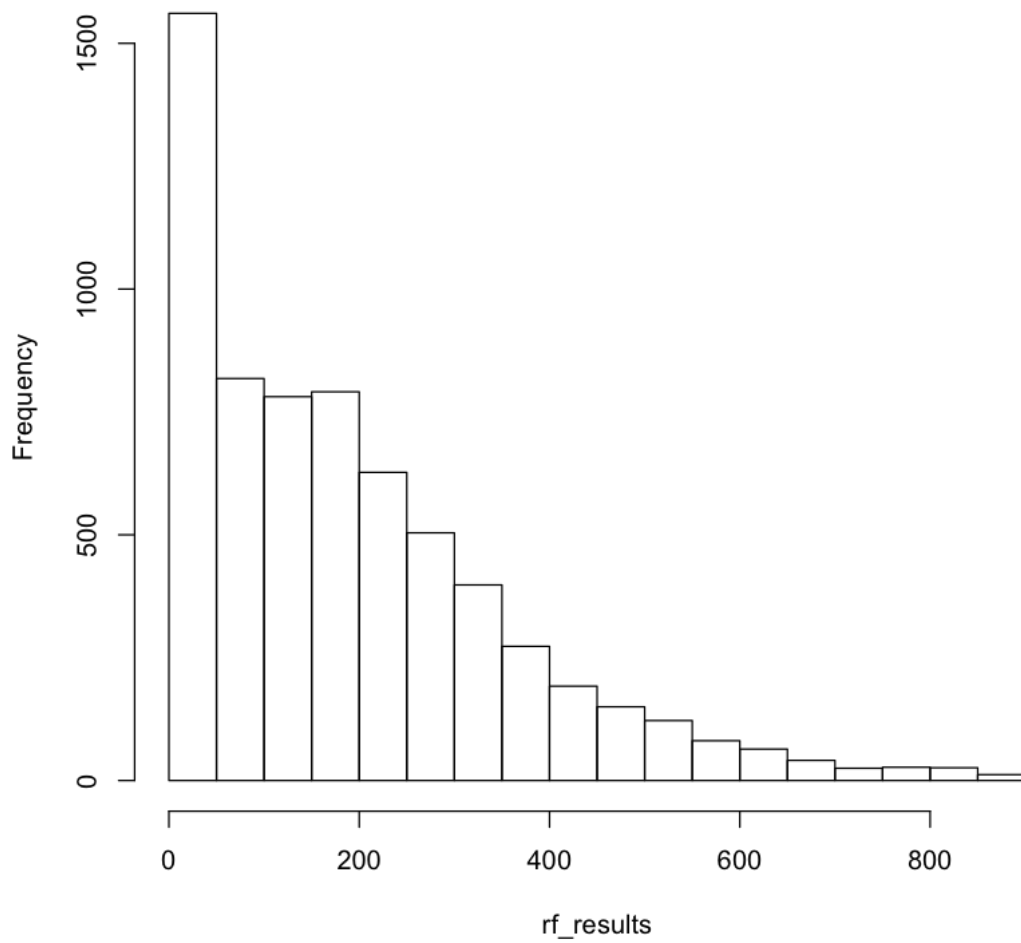
      # Inference: The distribution of predicted count looks similar to that
      → of train data.
```



Linear Regression Fit



Random Forest Fit



[]: