R Notebook

Code ▾

Hide

```
bike <- read.csv('C:/Users/Jai Katariya/Desktop/Jai Katariya/Self Learning/R/Cours
e 2-R-Course-HTML-Notes/R-Course-HTML-Notes/R-for-Data-Science-and-Machine-Learnin
g/Training Exercises/Machine Learning Projects/CSV files for ML Projects/bikeshar
e.csv')
bike <- as.data.frame(bike)
head(bike)
```

	datetime	sea…	holiday	workingday	weather	t…	atemp	humidity	winds
	<fctr>	<int>	<int>	<int>	<int>	<dbl>	<dbl>	<int>	
1	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	81	0
2	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	80	0
3	2011-01-01 02:00:00	1	0	0	1	9.02	13.635	80	0
4	2011-01-01 03:00:00	1	0	0	1	9.84	14.395	75	0
5	2011-01-01 04:00:00	1	0	0	1	9.84	14.395	75	0
6	2011-01-01 05:00:00	1	0	0	2	9.84	12.880	75	6

6 rows | 1-10 of 12 columns

Hide

```
str(bike)
```

```
'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 ...
```

Hide

```
summary(bike)
```

```
                datetime          season         holiday           workingday    
weather            temp           atemp          humidity      
 2011-01-01 00:00:00:    1   Min.   :1.000   Min.   :0.00000   Min.   :0.0000   Mi
n.   :1.000   Min.   : 0.82   Min.   : 0.76   Min.   :  0.00  
 2011-01-01 01:00:00:    1   1st Qu.:2.000   1st Qu.:0.00000   1st Qu.:0.0000   1s
t Qu.:1.000   1st Qu.:13.94   1st Qu.:16.66   1st Qu.: 47.00  
 2011-01-01 02:00:00:    1   Median :3.000   Median :0.00000   Median :1.0000   Me
dian :1.000   Median :20.50   Median :24.24   Median : 62.00  
 2011-01-01 03:00:00:    1   Mean   :2.507   Mean   :0.02857   Mean   :0.6809   Me
an   :1.418   Mean   :20.23   Mean   :23.66   Mean   : 61.89  
 2011-01-01 04:00:00:    1   3rd Qu.:4.000   3rd Qu.:0.00000   3rd Qu.:1.0000   3r
d Qu.:2.000   3rd Qu.:26.24   3rd Qu.:31.06   3rd Qu.: 77.00  
 2011-01-01 05:00:00:    1   Max.   :4.000   Max.   :1.00000   Max.   :1.0000   Ma
x.   :4.000   Max.   :41.00   Max.   :45.45   Max.   :100.00  
 (Other)            :1088                                                         
0                                                             

   windspeed          casual         registered        count      
 Min.   : 0.000   Min.   :  0.00   Min.   :  0.0   Min.   :  1.0  
 1st Qu.: 7.002   1st Qu.:  4.00   1st Qu.: 36.0   1st Qu.: 42.0  
 Median :12.998   Median : 17.00   Median :118.0   Median :145.0  
 Mean   :12.799   Mean   : 36.02   Mean   :155.6   Mean   :191.6  
 3rd Qu.:16.998   3rd Qu.: 49.00   3rd Qu.:222.0   3rd Qu.:284.0  
 Max.   :56.997   Max.   :367.00   Max.   :886.0   Max.   :977.0  
```

Hide

```
tail(bike)
```

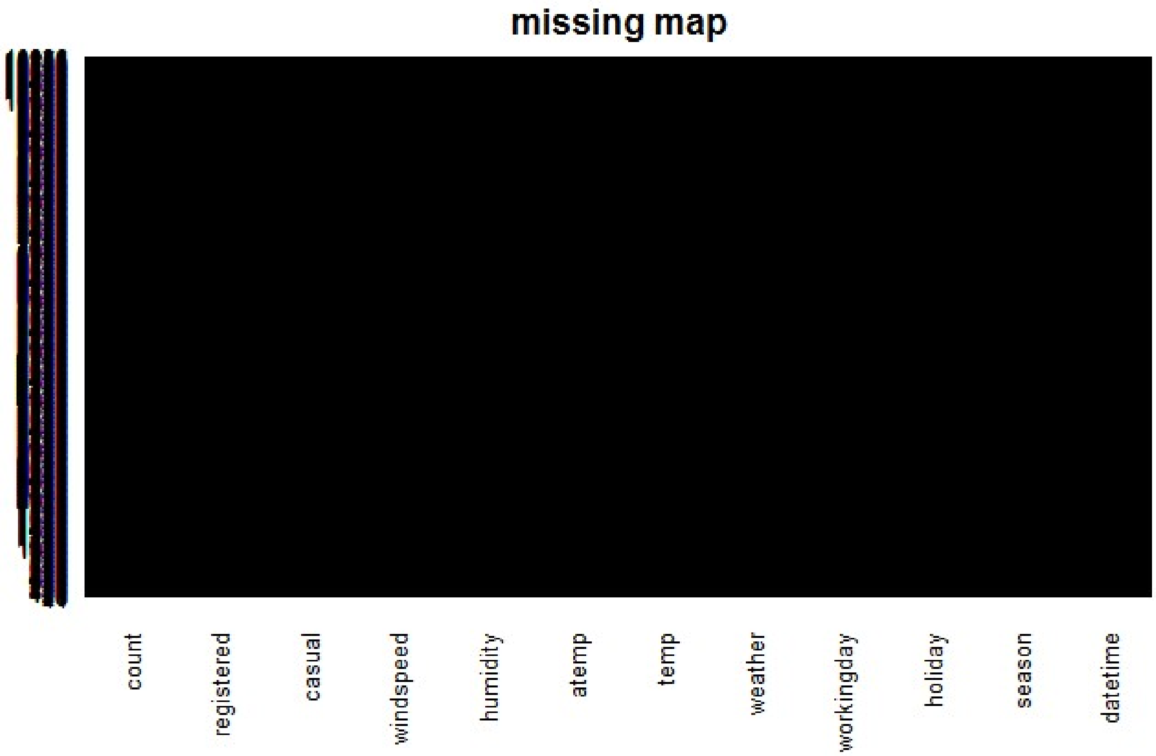
	datetime <fctr>	sea… <int>	holiday <int>	workingday <int>	weather <int>	temp <dbl>	atemp <dbl>	humidity <int>
10881	2012-12-19 18:00:00	4	0	1	1	15.58	19.695	50
10882	2012-12-19 19:00:00	4	0	1	1	15.58	19.695	50
10883	2012-12-19 20:00:00	4	0	1	1	14.76	17.425	57
10884	2012-12-19 21:00:00	4	0	1	1	13.94	15.910	61
10885	2012-12-19 22:00:00	4	0	1	1	13.94	17.425	61
10886	2012-12-19 23:00:00	4	0	1	1	13.12	16.665	66

6 rows | 1-10 of 12 columns

Exploratory Data Analysis¶

Hide

```
library(Amelia)
missmap(bike, main = 'missing map', col = c('yellow', 'black'), legend =F)
```



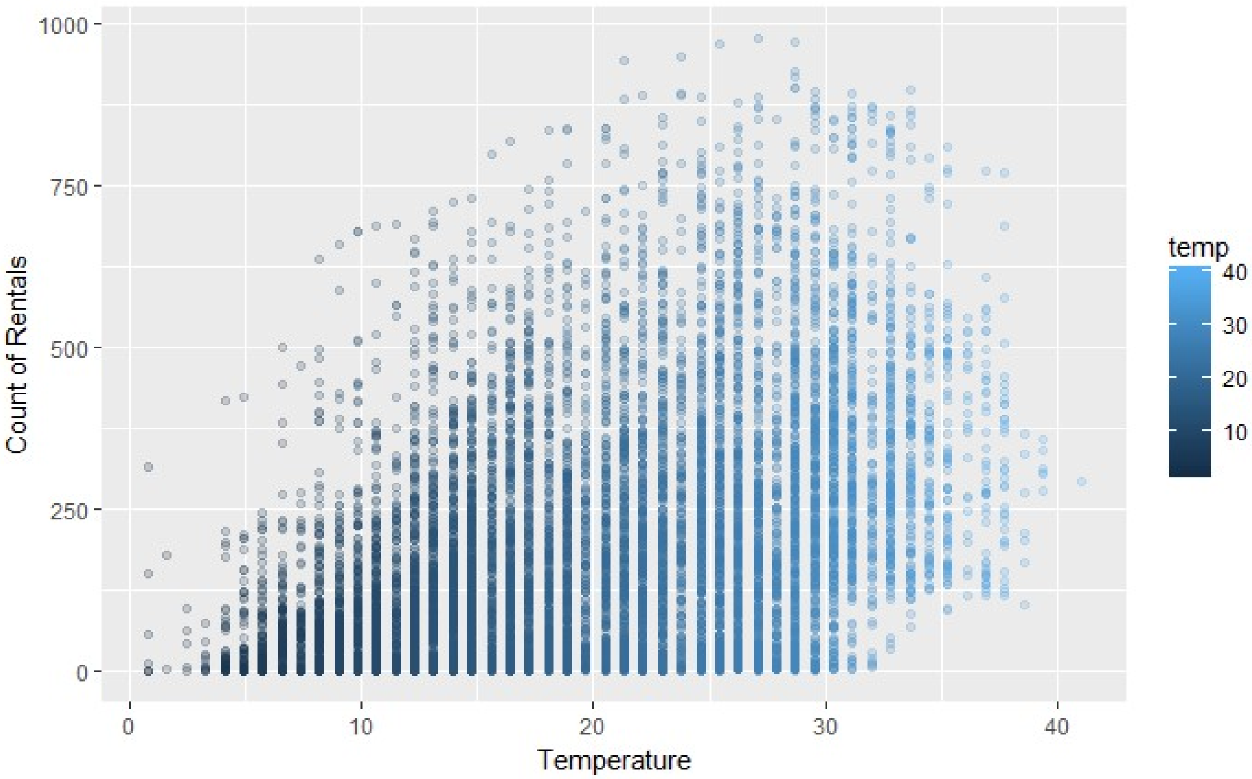
Create a scatter plot of count vs temp.

Hide

```
library(ggplot2)
temp_scatterplot <- ggplot(bike, aes(temp, count)) +

                    geom_point(aes(color = temp),alpha = 0.2)+

                    xlab('Temperature') + ylab("Count of Rentals")
print(temp_scatterplot)
```



Plot count versus datetime as a scatterplot with a color gradient based on temperature. Convert the datetime column into POSIXct before plotting.

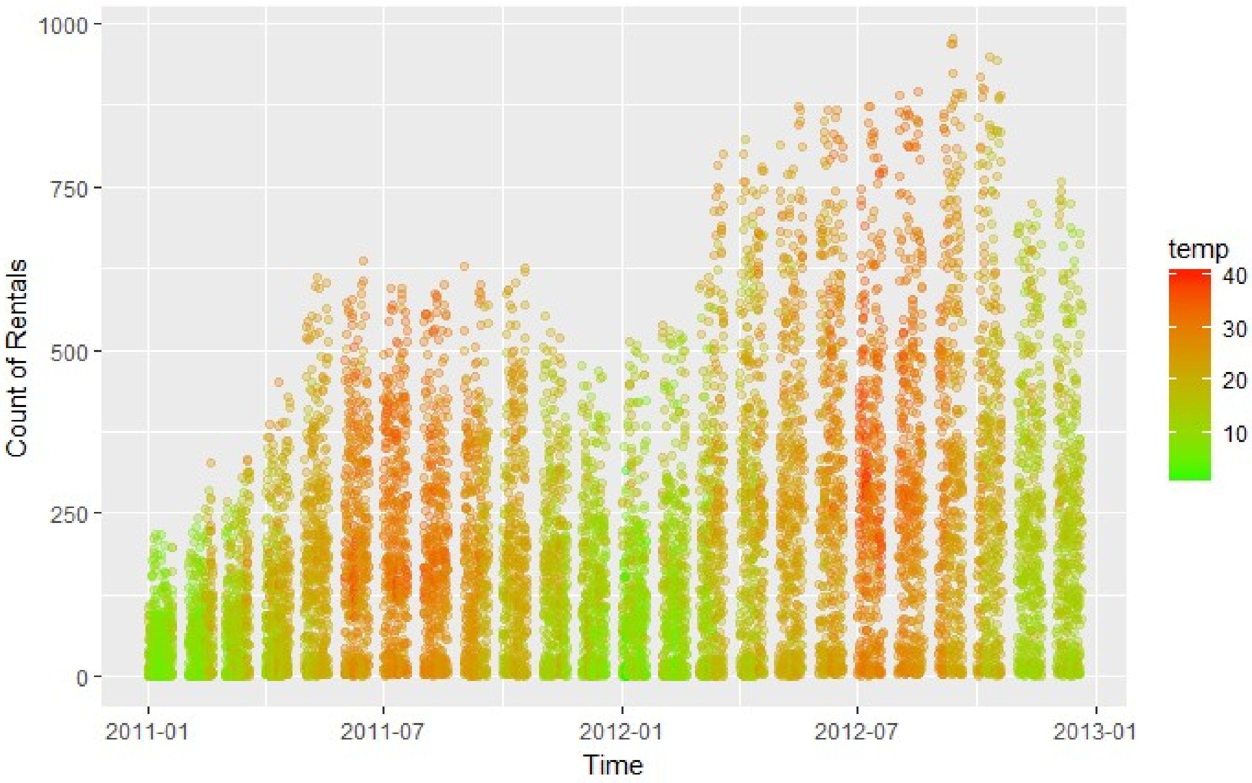
Hide

```
bike$datetime <- as.POSIXct(bike$datetime)
datetime_scatterplot <- ggplot(bike, aes(datetime, count)) +

                        geom_point(alpha = 0.3, aes(color = temp))+

                        scale_color_gradient(low = 'green', high = 'red') +

                        xlab('Time') + ylab("Count of Rentals")
print(datetime_scatterplot)
```



Hide

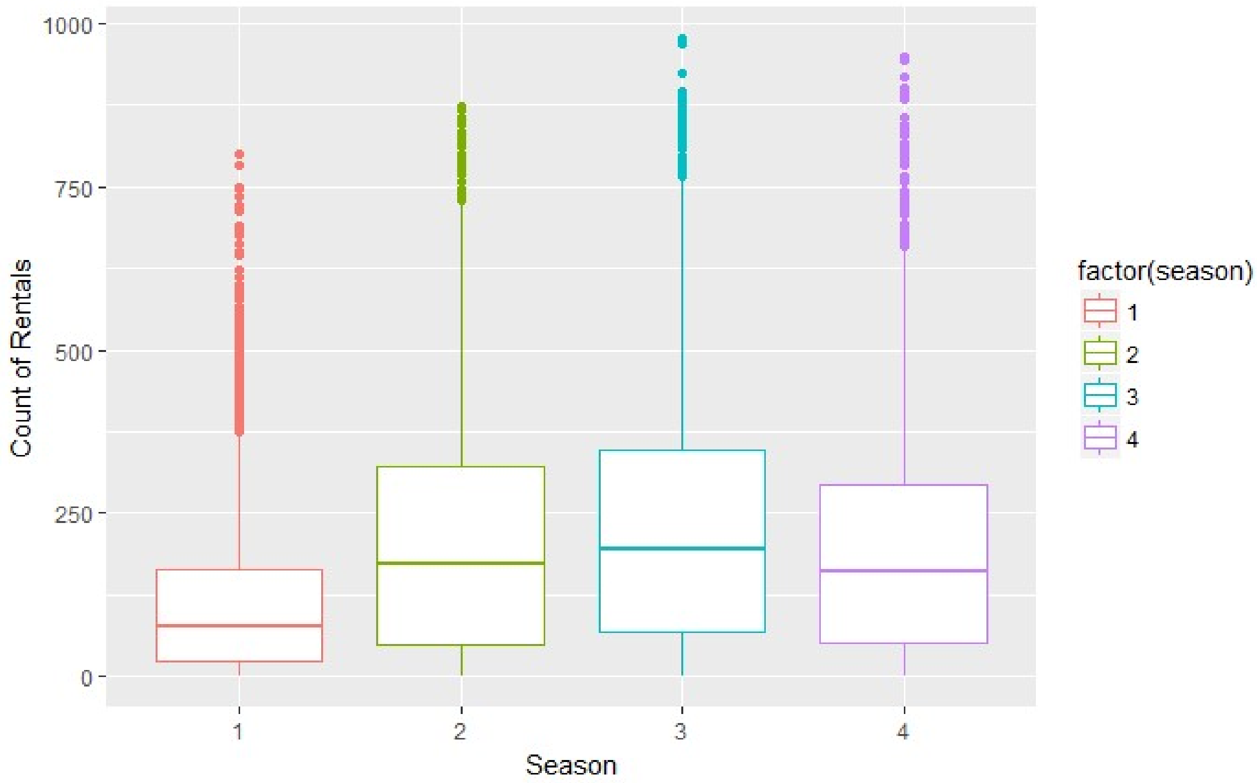
```
cor_tempVscount <- cor(bike[, c('temp', 'count')])
print(cor_tempVscount)
```

```
           temp     count
temp  1.0000000 0.3944536
count 0.3944536 1.0000000
```

Created a boxplot to explore the season data with the y axis indicating count and the x axis begin a box for each season.

Hide

```
seasons <- ggplot(bike, aes(factor(season), count)) + geom_boxplot(aes(color = fac
tor(season))) + xlab('Season') + ylab("Count of Rentals")
print(seasons)
```



Feature Engineering

Before dealing with date time column, we need to feature it.

Created an “hour” column that takes the hour from the datetime column.

Hide

```
bike$hour <- sapply(bike$datetime, function(x){format(x, '%H')})
head(bike)
```

	datetime	sea…	holiday	workingday	weather	t…	atemp	humidity	winds
	<S3: POSIXct>	<int>	<int>	<int>	<int>	<dbl>	<dbl>	<int>	
1	2011-01-01 00:00:00	1	0	0	1	9.84	14.395	81	0
2	2011-01-01 01:00:00	1	0	0	1	9.02	13.635	80	0
3	2011-01-01 02:00:00	1	0	0	1	9.02	13.635	80	0
4	2011-01-01 03:00:00	1	0	0	1	9.84	14.395	75	0
5	2011-01-01 04:00:00	1	0	0	1	9.84	14.395	75	0
6	2011-01-01 05:00:00	1	0	0	2	9.84	12.880	75	6

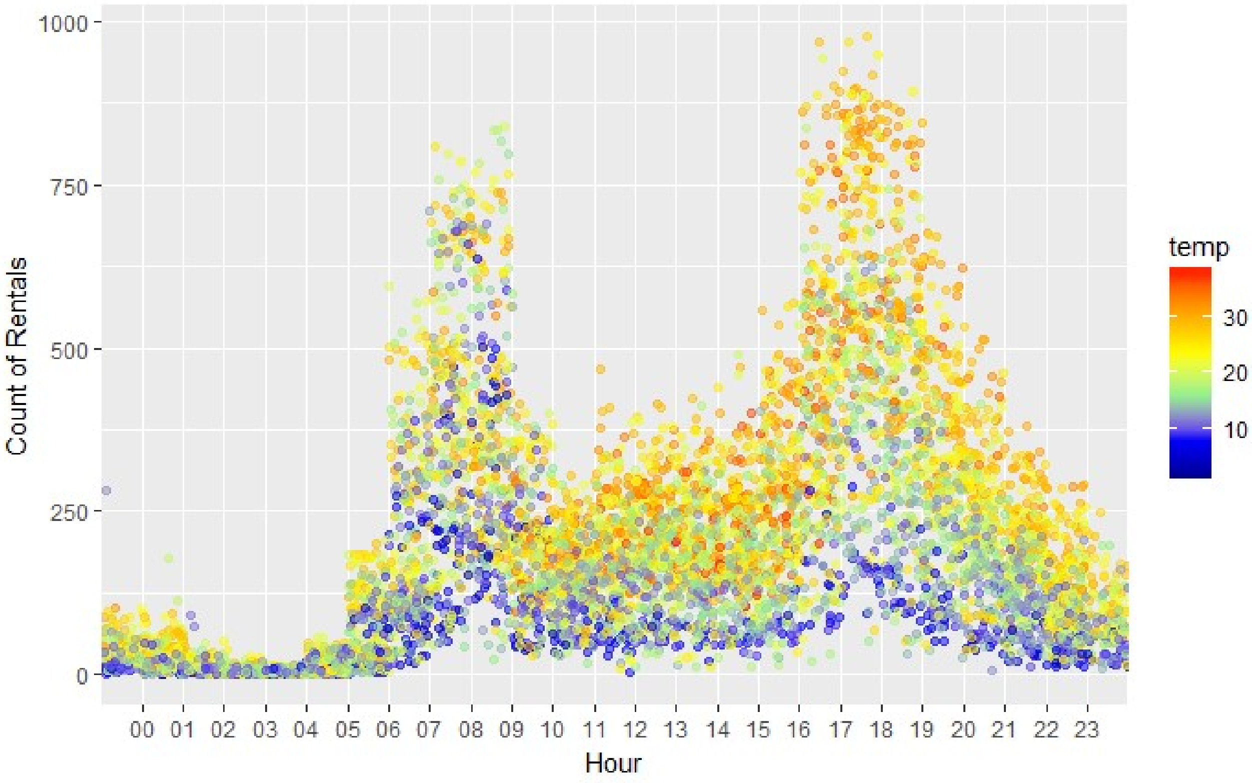
6 rows | 1-10 of 13 columns

Now create a scatterplot of count versus hour, with color scale based on temp. Only use bike data where workingday==1. Additions:Used the additional layer: scale_color_gradientn(colors=c(‘color1’,color2,etc..)) where the colors argument is a vector gradient of colors you choose, not just high and low. Used position=position_jitter(w=1, h=0) inside of geom_point() and check out what it does.

Hide

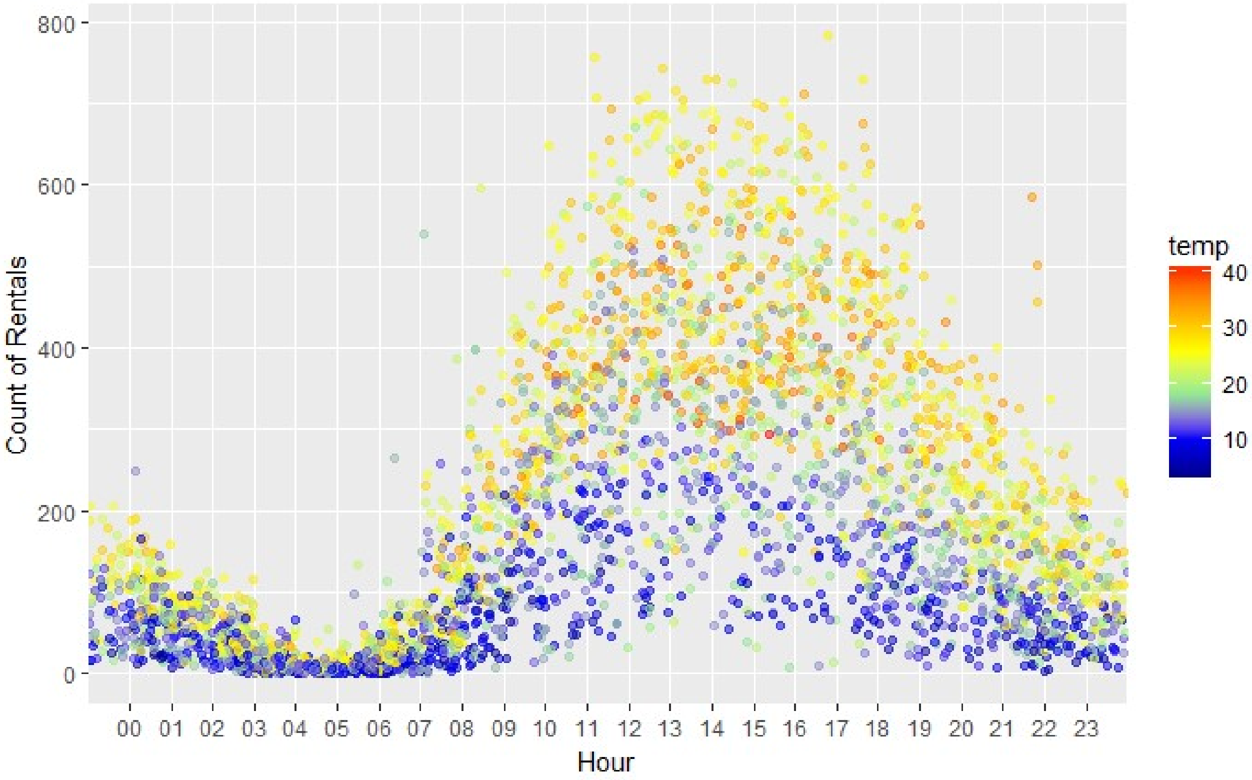
```
bike_data <- subset(bike, bike$workingday == 1)
hour_scatterplot <- ggplot(bike_data,aes(hour, count)) + geom_point(aes(color = temp),

                    position =  position_jitter(w = 1, h=0), alpha = 0.5)+ scale_color_gradientn(colors= c('dark blue', 'blue', 'light green', 'yellow', 'orange', 'red')) + xlab('Hour') + ylab("Count of Rentals")
print(hour_scatterplot)
```



Hide

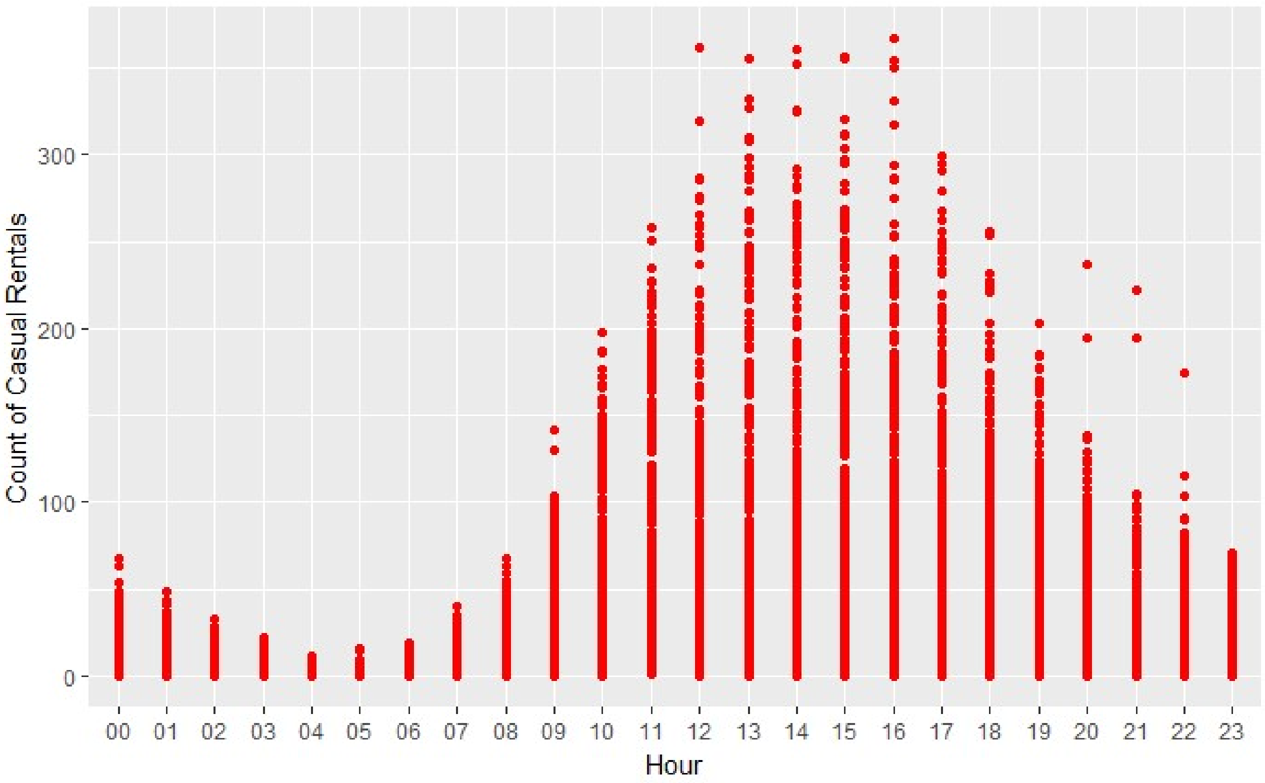
```
library(dplyr)
hour2_scatterplot <- ggplot(filter(bike, workingday ==0), aes(hour, count)) +
                    geom_point(aes(color = temp), position = position_jitter(w = 1, h = 0),
                    alpha = 0.5) + scale_color_gradientn(colors= c('dark blue', 'blue',
                    'light green', 'yellow', 'orange', 'red')) + xlab('Hour') + ylab("Count of Rentals")
print(hour2_scatterplot)
```



Compare Count of Rentals with Casual and Registered.

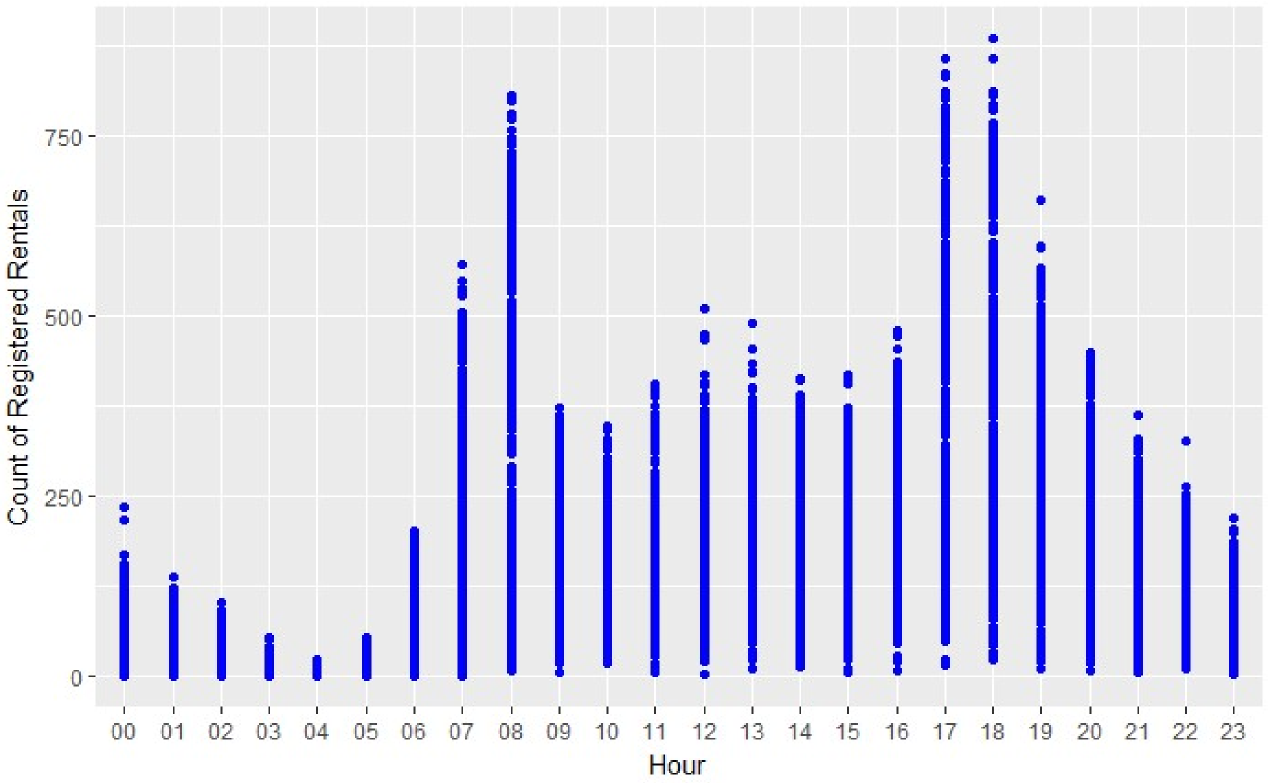
Hide

```
Ca_scatterplot <- ggplot(bike,aes(hour, casual)) + geom_point(colour ='Red') + xla
b('Hour') + ylab("Count of Casual Rentals")
print(Ca_scatterplot)
```



Hide

```
Re_scatterplot <- ggplot(bike,aes(hour, registered)) + geom_point(colour ='blue')
+ xlab('Hour') + ylab("Count of Registered Rentals")
print(Re_scatterplot)
```



For an interactive plot, I installed plotly library.

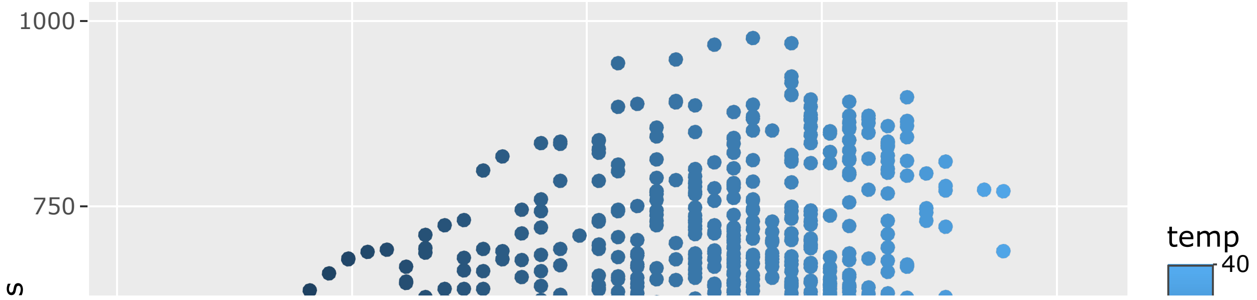
Hide

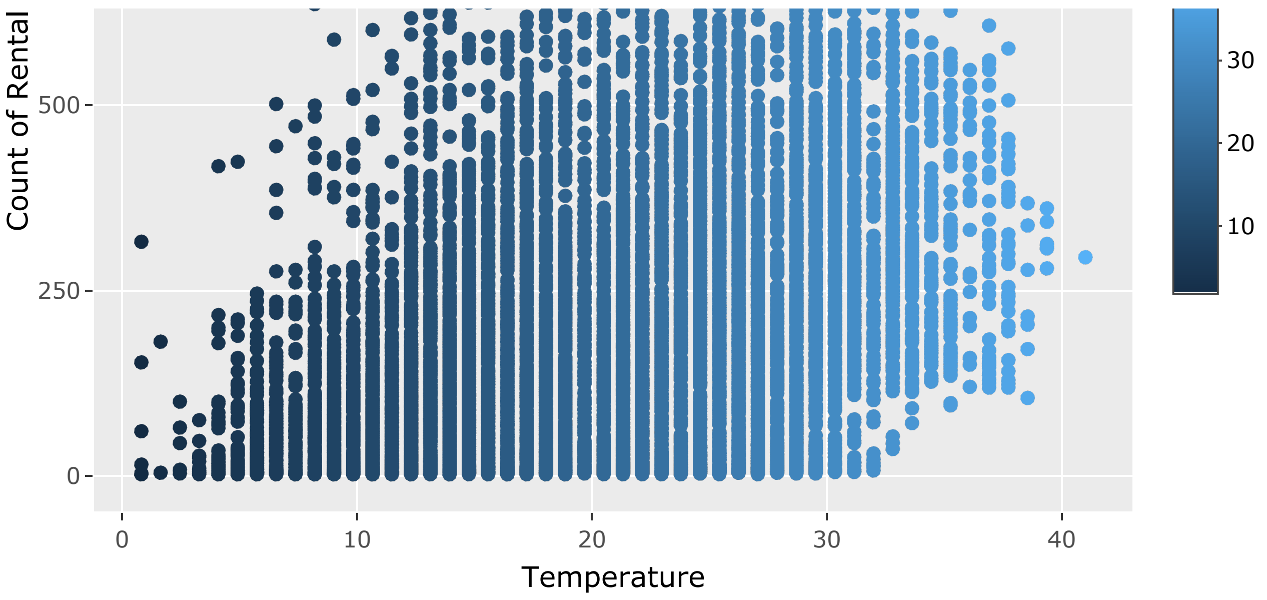
```
#install.packages('plotly')
library(ggplot2)
library(plotly)
CountVsTemp <- ggplot(bike, aes(temp, count)) + geom_point(aes(color = temp)) + xlab("Temperature") + ylab("Count of Rentals")
gpl <- ggplotly(CountVsTemp)
```

```
We recommend that you use the dev version of ggplot2 with `ggplotly()`
Install it with: `devtools::install_github('hadley/ggplot2')`
```

Hide

```
print(gpl)
```





```
NULL
```