```
library(ggplot2)
library(dplyr)
```

Read files

```
train_bike<-read.csv('train.csv')
test_bike<-read.csv('test.csv')
```

Number of column should be same in train and test. Our target is to find total count which is registered+casual. So we can build Model to find count directly insted of finding registered+casual.

Remove registered and causal from training set and then Add count column in test and combine both datasets.

```
test_bike$count<-NA

train_bike<-select(train_bike,-registered,-casual)

bike<-rbind(train_bike,test_bike)
```

DateTime VS Count

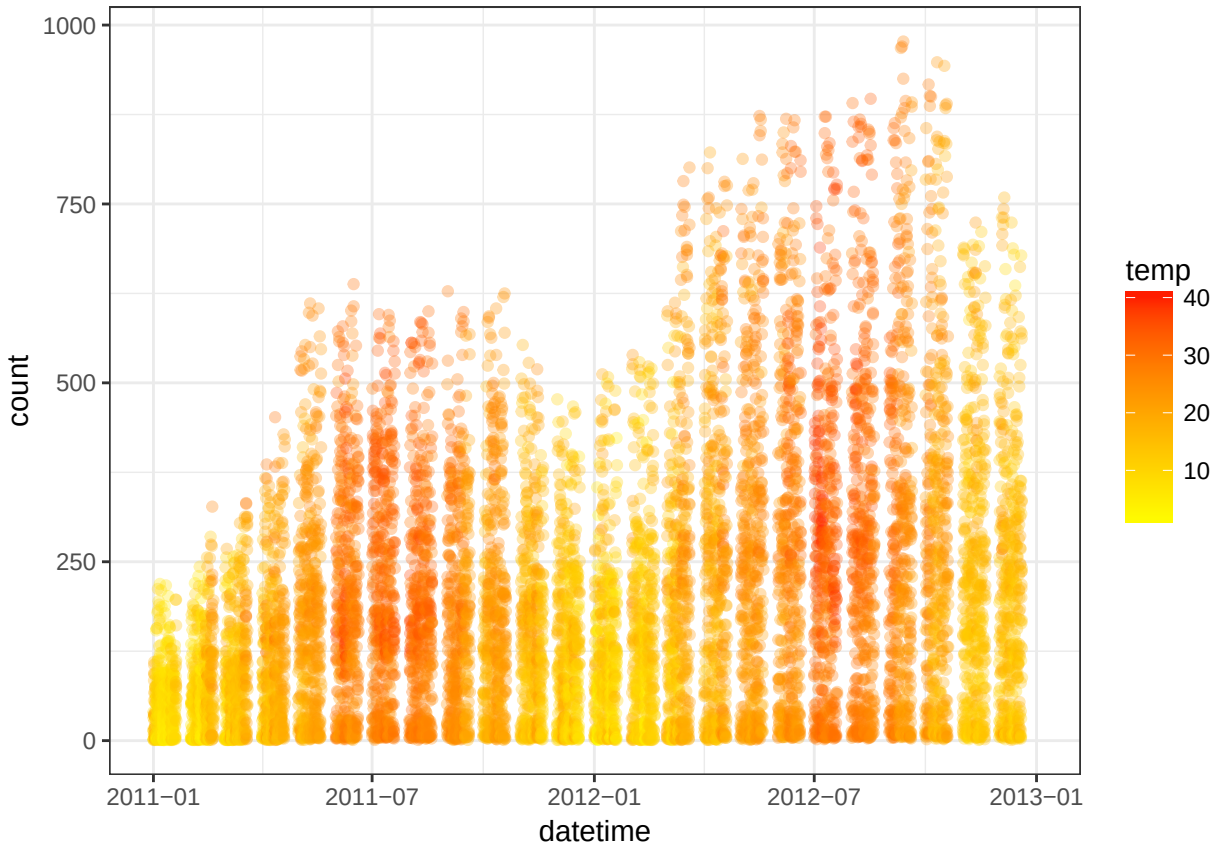
```
class(bike$datetime)
```

```
## [1] "factor"
```

```
bike$datetime<-as.POSIXct(bike$datetime)

class(bike$datetime)
```

```
## [1] "POSIXct" "POSIXt"
```

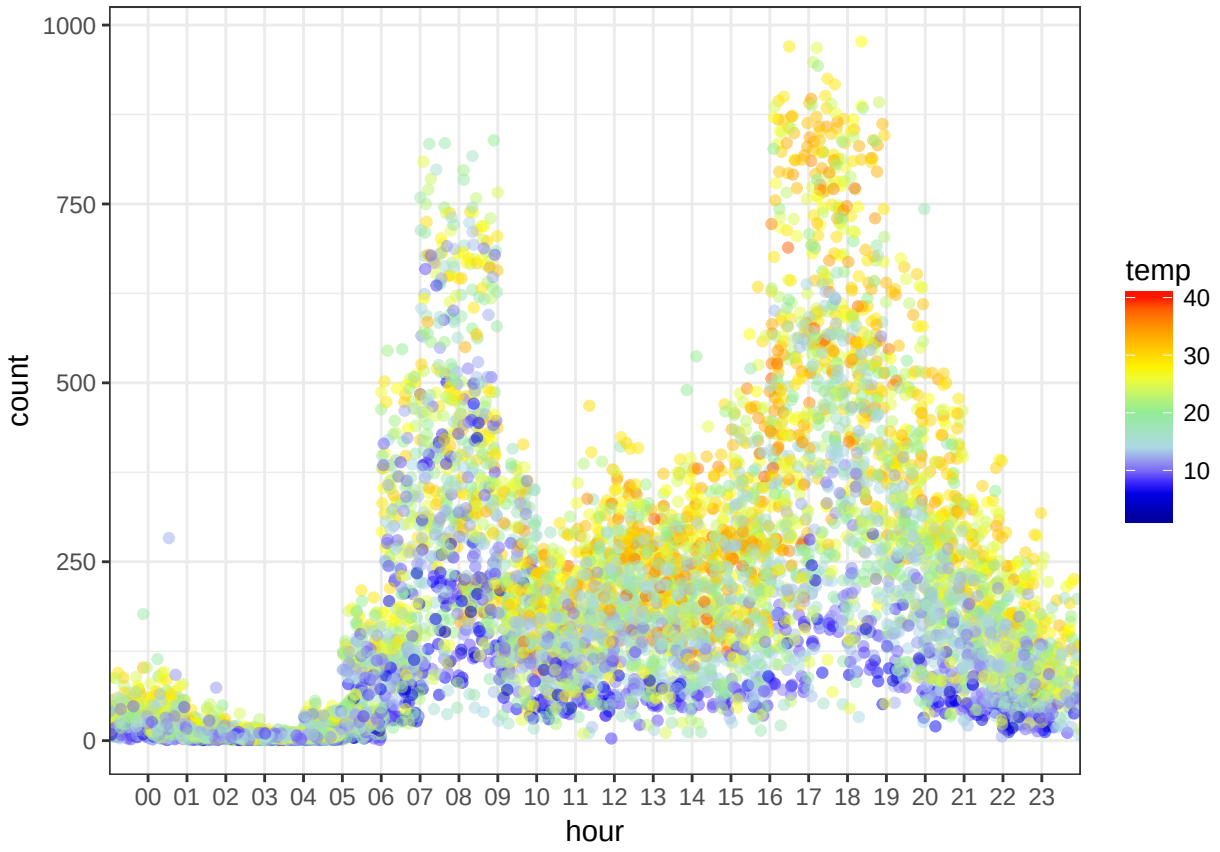
```
ggplot(bike,aes(datetime,count))+geom_point(aes(color=temp),alpha=0.3)+scale_color_continuous(low = 'ye
```



We can see,over the time demand is incresing so linear model is not fit for such data.

Working day vs Demand

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



ggplot(filter(bike,workingday==1),aes(hour,count))+geom_point(aes(color=temp),position = position_jitte
```



Peak hour:6-9 ,12-15,16-19

```
class(bike$hour) #Character
```

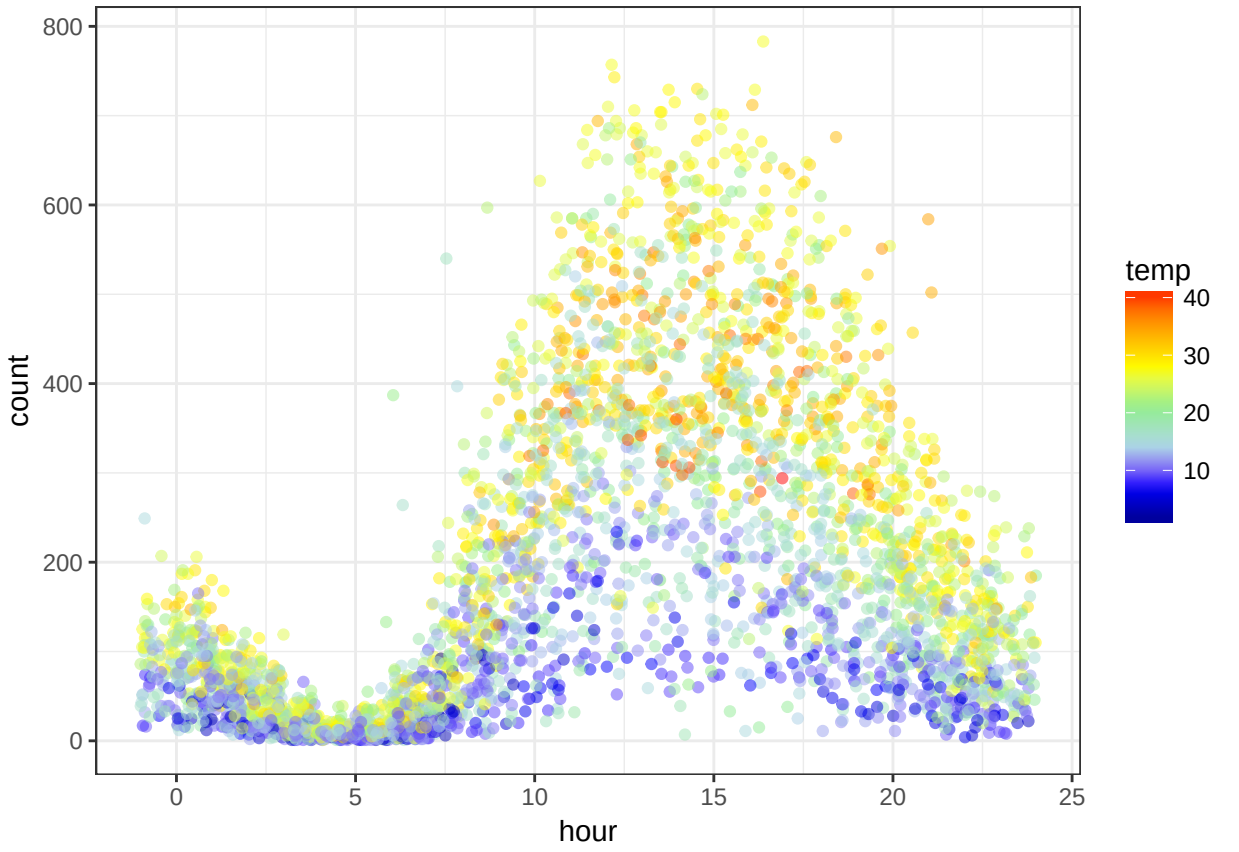
```
## [1] "character"
```

```
bike$hour<-as.numeric(bike$hour)

bike$daypart<-0

bike$daypart[bike$hour>=6 & bike$hour<=9]<-1
bike$daypart[bike$hour>=12 & bike$hour<=15]<-1
bike$daypart[bike$hour>=16 & bike$hour<=19]<-1

bike$daypart<-as.factor(bike$daypart)
```

holiday vs Demand

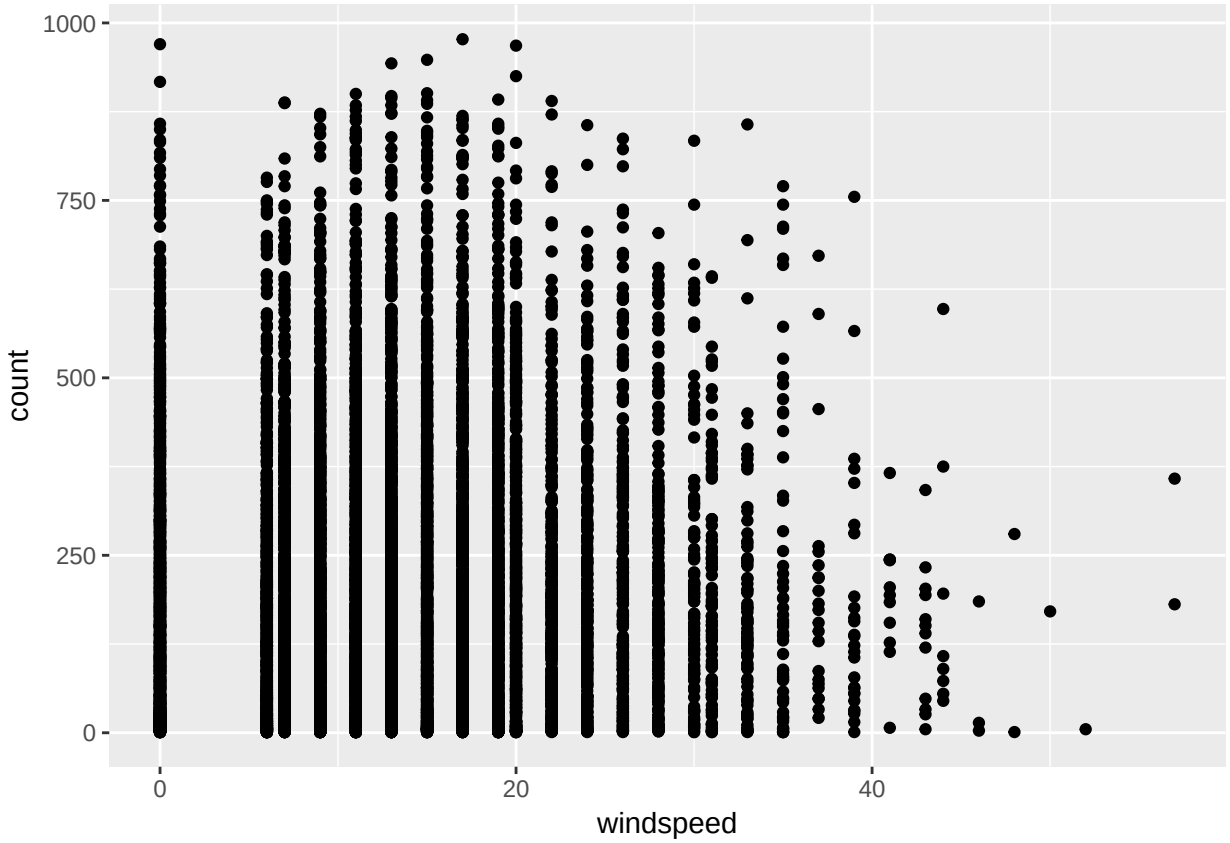
```
ggplot(filter(bike,workingday==0),aes(hour,count))+geom_point(aes(color=temp),position = position_jitte
```

```
## Warning: Removed 2040 rows containing missing values (geom_point).
```



weekday VS Demand

```
bike$weekday<-weekdays(as.Date(bike$datetime))


ggplot(bike,aes(windspeed,count))+geom_point()
```




```
bike$season<-as.factor(bike$season)
bike$holiday<-as.factor(bike$holiday)
bike$workingday<-as.factor(bike$workingday)
bike$weather<-as.factor(bike$weather)
bike$hour<-as.factor(bike$hour)
```

Splitting Dataset

```
#bike<-select(bike,c(-datetime,-hour))
test<-filter(bike,is.na(count))
train<-filter(bike,!is.na(count))
```

xgBoost

```
#install.packages('xgboost')
#library(xgboost)

#xgboost dont work with factors. It needs only numeric variables.

#classifier<-xgboost(data=as.matrix(select(train,c(-datetime,-hour,-weekday,-count))),label = train$cou

#pred<-predict(classifier,newdata = as.matrix(select(train,c(-datetime,-hour,-weekday,-count))))
```

RandomForest

```
library(randomForest)
```

```
## randomForest 4.6-12
```

```
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:dplyr':
##
##     combine
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
rf_classifier<-randomForest(count~season+holiday+workingday+weather+temp+atemp+humidity+windspeed+daypa

rf_pred<-predict(rf_classifier,test)
```

Kaggle submission file

```
#s<-data.frame(datetime=test$datetime,count=rf_pred)
#write.csv(s,file="bike_solution.csv",row.names=FALSE)
```