Social Media Analytics – Yelp

Get the Yelp data

There are several kinds of datasets one could obtain from Yelp from this Website: https://www.yelp.com/dataset/challenge. Yelp is providing these datasets as a part of running various rounds of data challenges. We directly download it from Yelp.

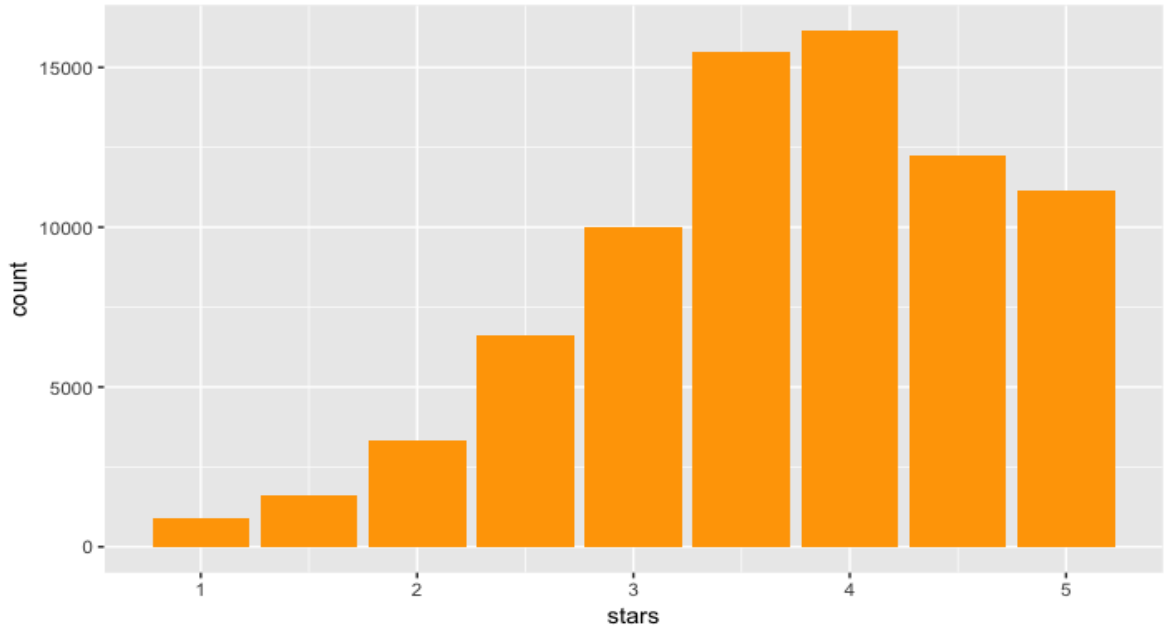
Visualize the data

We create bar chart, pie chart, histogram, scatter plot, stacked bar graph using R.

Regression Analysis

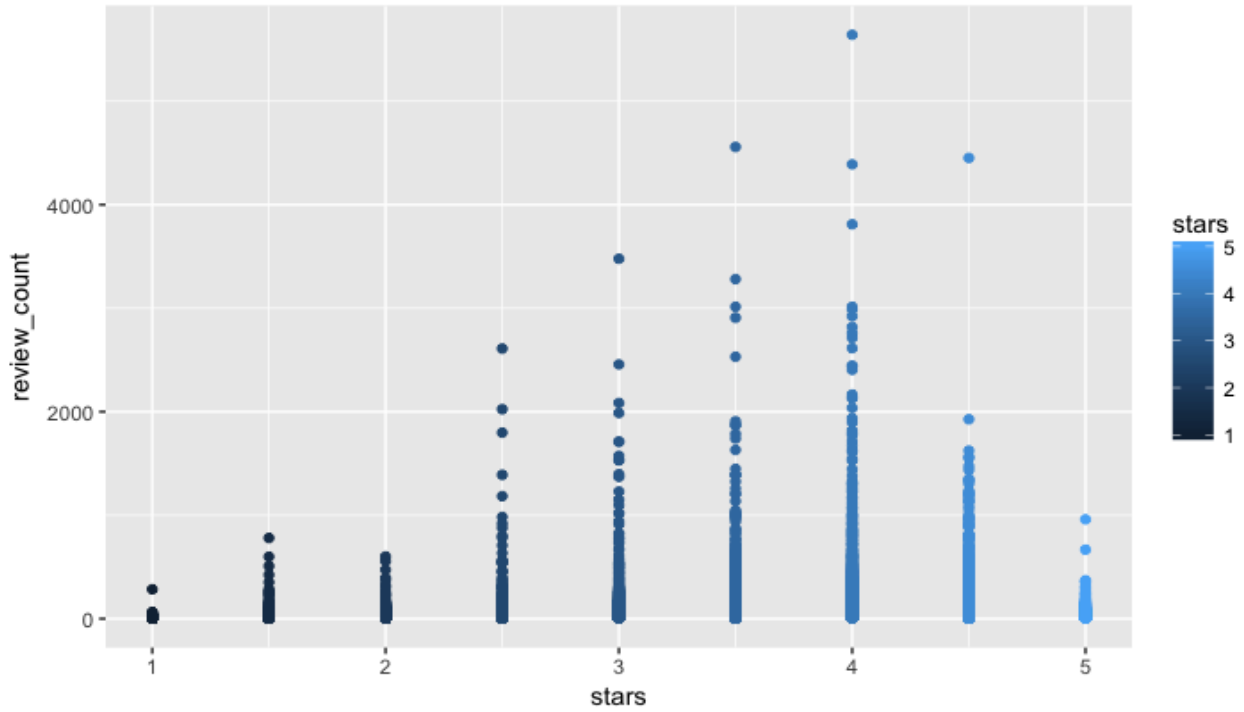
We use the dataset to see if there are any relationships among reviewcount, cool_votes, useful_votes and funny_votes.. We create a R script (.R) for analysing the relationships. We find correlations among variables in the dataset. We create regression model, plot and line for varaibles having medium to high level positive/negative correlation. We check the R-square value and the p-value. Based on the regression plot and model, we state the hypothesis.

```
ggplot(business_data) +geom_bar(aes(x=stars),fill="orange")
```



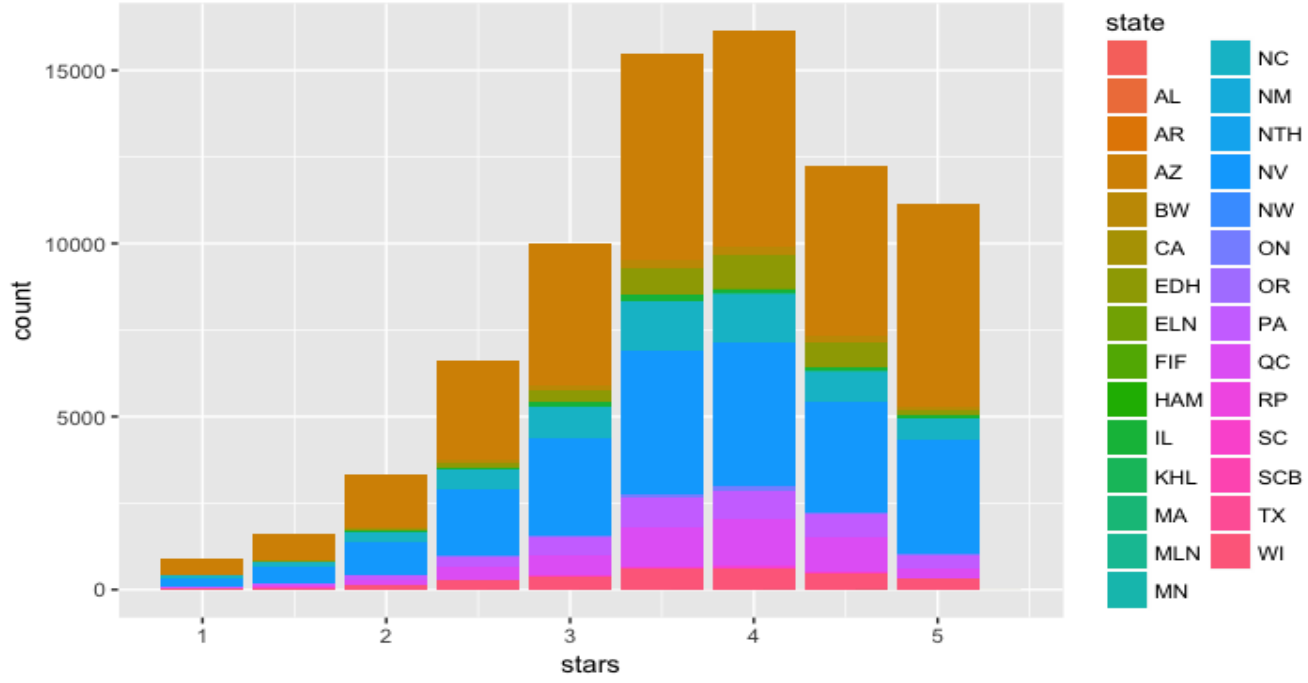
The above histogram shows the count of each stars in the business dataset.

ggplot(business_data,aes(stars,review_count,color=stars)) + geom_point()



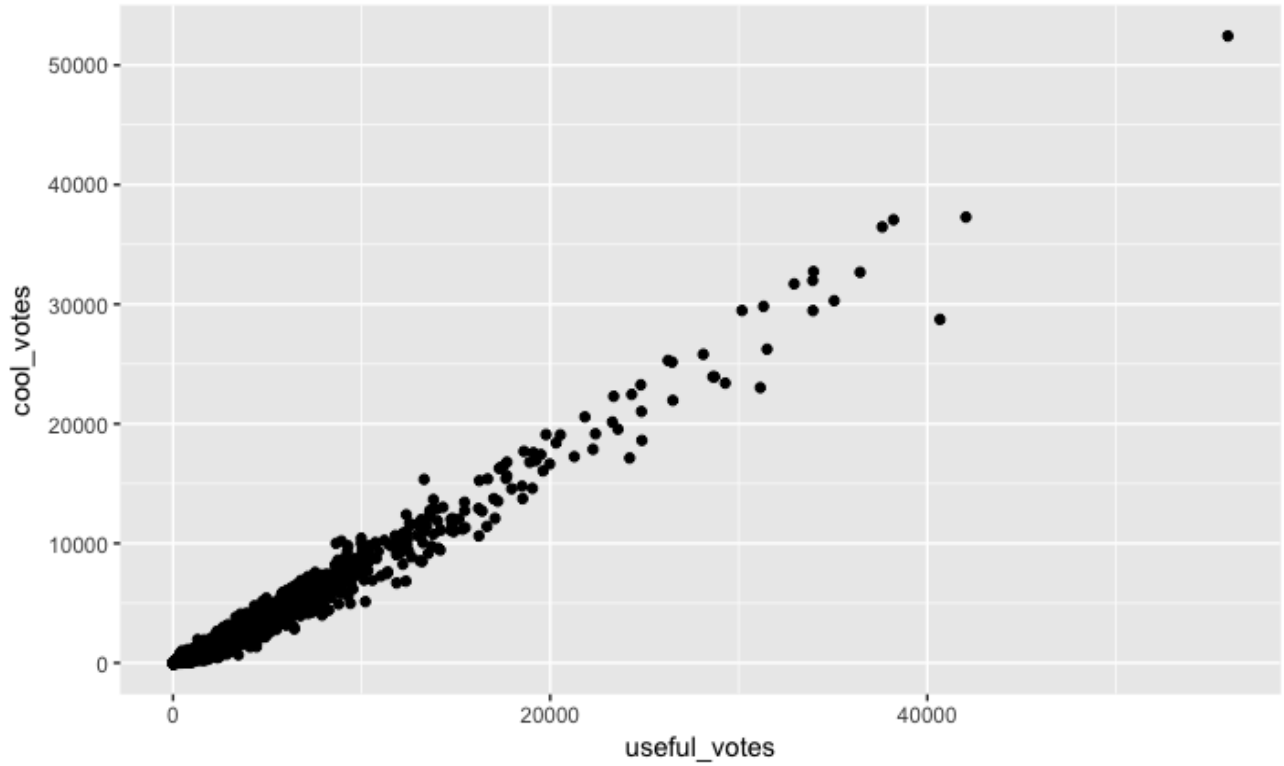
The above scatterplot shows for each stars, the reviews businesses got in business dataset. So one of the business got more than 5000 reviews for 4 stars.

ggplot(business_data, aes(stars, fill = state)) + geom_bar()



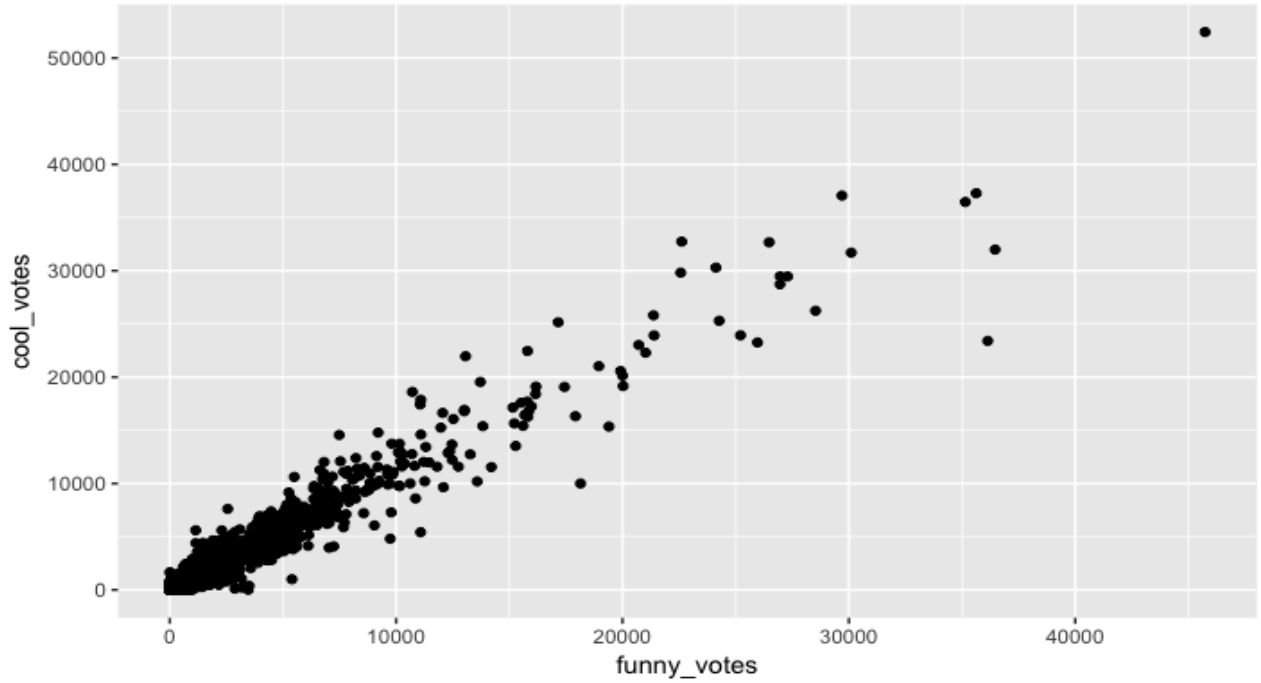
The stacked bar graph shows those states which belong to each stars. So for Star 4, there are AZ, NV, PA, etc.

ggplot(user_data,aes(x=useful_votes,y=cool_votes)) + geom_point()



The above scatterplot shows a linear relationship between useful_votes and cool_votes variables in the user dataset.

ggplot(user_data,aes(x= funny_votes,y=cool_votes)) + geom_point()



The above scatterplot shows a linear relationship between funny_votes and cool_votes variables in the user dataset.

Based on the above 2 plots, we do more analysis by finding correlation:

```
> cor(user_votes)
              review_count average_stars  cool_votes funny_votes useful_votes       fans
review_count   1.000000000   0.004493621 0.559061268 0.527872320  0.665702285 0.584905868
average_stars  0.004493621   1.000000000 0.005729636 0.001755019  0.001898496 0.009102177
cool_votes     0.559061268   0.005729636 1.000000000 0.976411252  0.983270786 0.752437116
funny_votes    0.527872320   0.001755019 0.976411252 1.000000000  0.954654087 0.731249538
useful_votes   0.665702285   0.001898496 0.983270786 0.954654087  1.000000000 0.789978217
fans           0.584905868   0.009102177 0.752437116 0.731249538  0.789978217 1.000000000
```

We see a strong positive correlation among cool_votes, useful_votes and funny_votes. We get 0.9832 correlation between cool_votes and useful_votes and 0.9764 correlation between cool_votes and funny_votes. More someone get votes to be cool, more useful and funny they receive. So we use useful_votes and funny_votes to predict cool_votes.

#Making a model
lm_model<-lm(cool_votes~useful_votes+funny_votes, data=user_data)
coeffs = coefficients (lm_model)

```
> lm_model<-lm(cool_votes~useful_votes+funny_votes, data=user_data)
> coeffs = coefficients (lm_model)
> coeffs
 (Intercept) useful_votes  funny_votes
  -8.1152793    0.4765104    0.4759054
```

Cool_votes = -8.1152793 + 0.4765 * useful_votes + 0.4759 * funny_votes

There maybe some interaction effect between useful_votes and funny_votes as they have a strong positive correlation. So adding interaction effect in the regression model.

```
> lm_model<-lm(cool_votes~useful_votes+funny_votes+useful_votes*funny_votes, data=user_data)
> coeffs = coefficients (lm_model)
> coeffs
            (Intercept)             useful_votes              funny_votes
          -7.351068e+00             4.934134e-01             3.969604e-01
useful_votes:funny_votes
           2.852543e-06
```

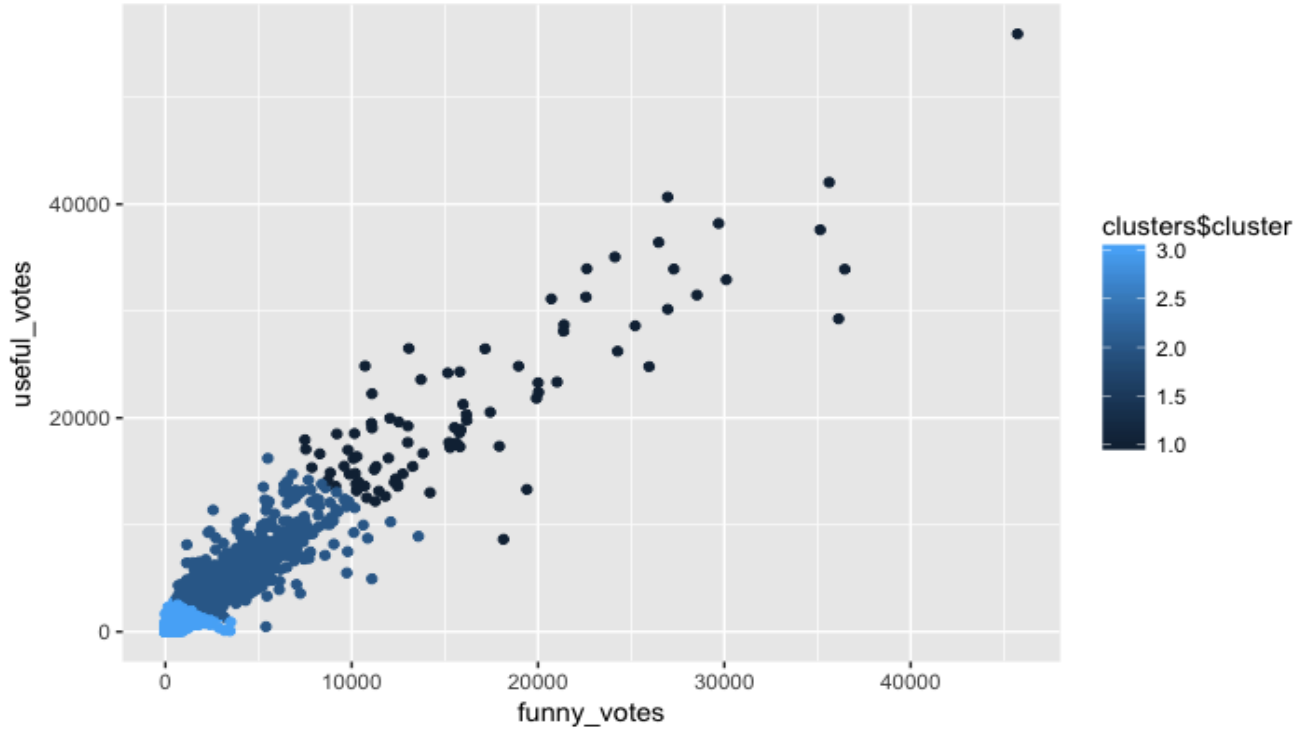
We have a coefficient for that interaction effect for the new regression model which is really small so we can ignore that factor.
#Using funny_votes and usefuk_votes for clustering

Taking K=3
clusters=kmeans(user_data[,c(6,7)],3)

```
> clusters=kmeans(user_data[,c(6,7)],3)
> clusters["centers"]
$centers
  funny_votes useful_votes
1 16714.70455  21455.11364
2  3005.00475   4479.64846
3    15.30856     37.59505

> clusters["size"]
$size
[1]     88   1263 550988
```

ggplot(user_data,aes(funny_votes,useful_votes,color=clusters$cluster)) +geom_point()



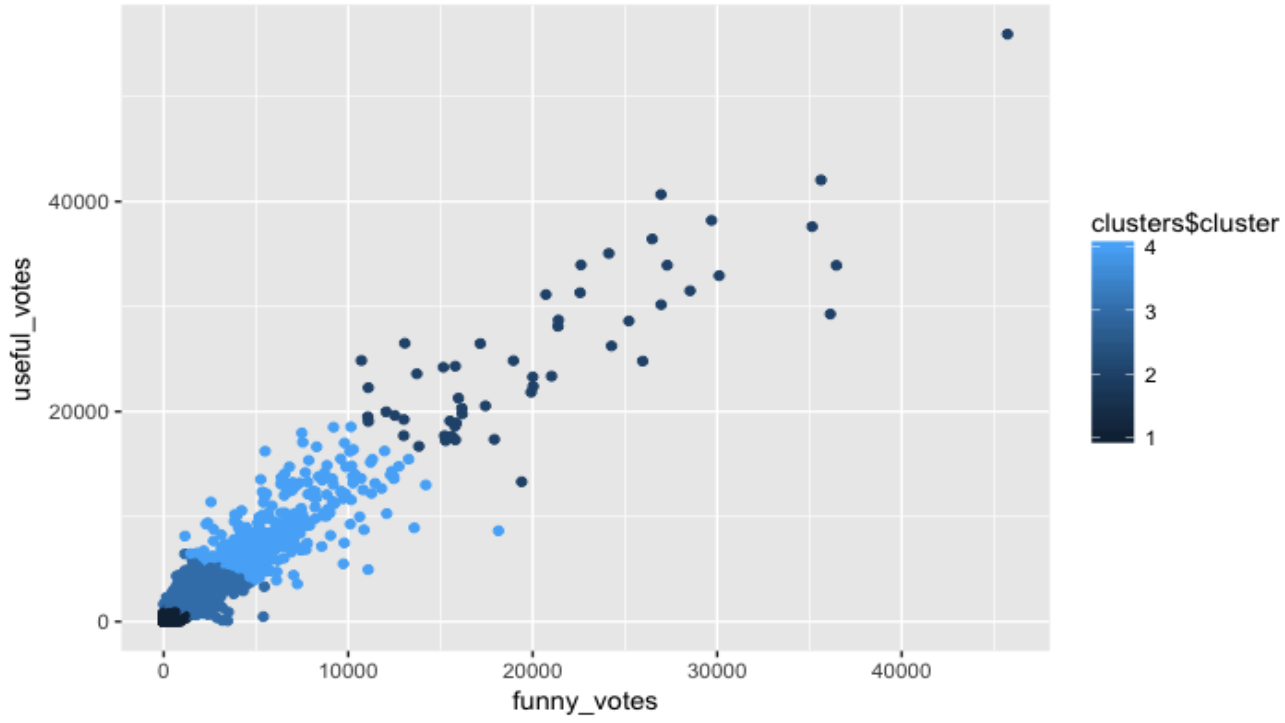
Taking K=4

clusters=kmeans(user_data[,c(6,7)],3)

```
> clusters=kmeans(user_data[,c(6,7)],4)
> clusters["centers"]
$centers
  funny_votes useful_votes
1     10.8372     29.72662
2  20452.7037  25685.09259
3   1084.3028   1823.49662
4   5707.6143   8094.63391

> clusters["size"]
$size
[1] 547879     54   3999    407
```

ggplot(user_data,aes(funny_votes,useful_votes,color=clusters$cluster)) +geom_point()

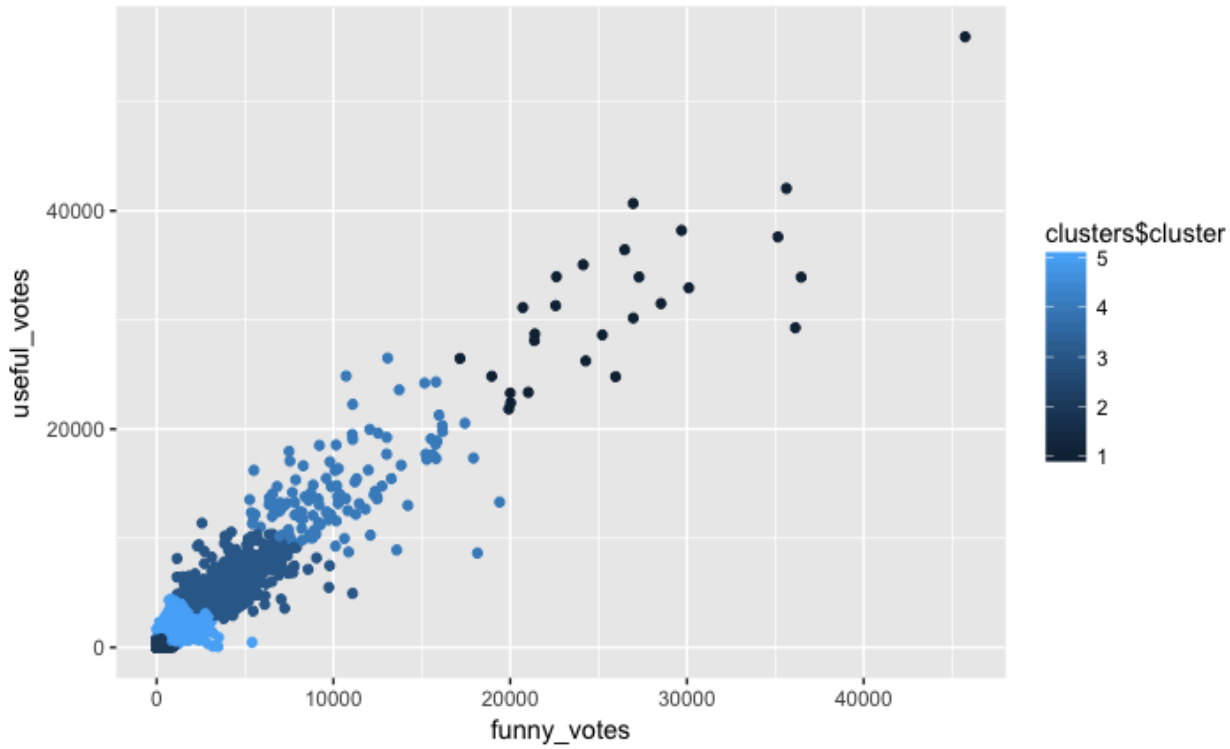


Taking k=5

```
> clusters=kmeans(user_data[,c(6,7)],5)
> clusters["centers"]
$centers
   funny_votes useful_votes
1 26313.000000  31586.96296
2     9.677207     27.34561
3  3806.190789   5491.41283
4 10373.669565  14569.73043
5   792.164590   1407.03228

> clusters["size"]
$size
[1]     27 546291    608    115   5298
```

ggplot(user_data,aes(funny_votes,useful_votes,color=clusters$cluster)) +geom_point()



Perhaps I see that for K=5 the points are clustered better than K= 3 or 4. Perhaps there are 5 clusters.