lukasz grzybek - hw5

a. Data gathering and integration

```
library(readr)
layoffs <- read_csv("layoffs.csv")
```

```
## Rows: 1651 Columns: 9
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (5): company, location, industry, stage, country
## dbl  (3): total_laid_off, percentage_laid_off, funds_raised
## date (1): date
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
View(layoffs)
```

```
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✓ ggplot2 3.3.6      ✓ dplyr   1.0.10
## ✓ tibble  3.1.8      ✓ stringr 1.4.1
## ✓ tidyr   1.2.1      ✓ forcats 0.5.2
## ✓ purrr   0.3.4
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
#creates new data frame

df <- as.data.frame(layoffs)
```

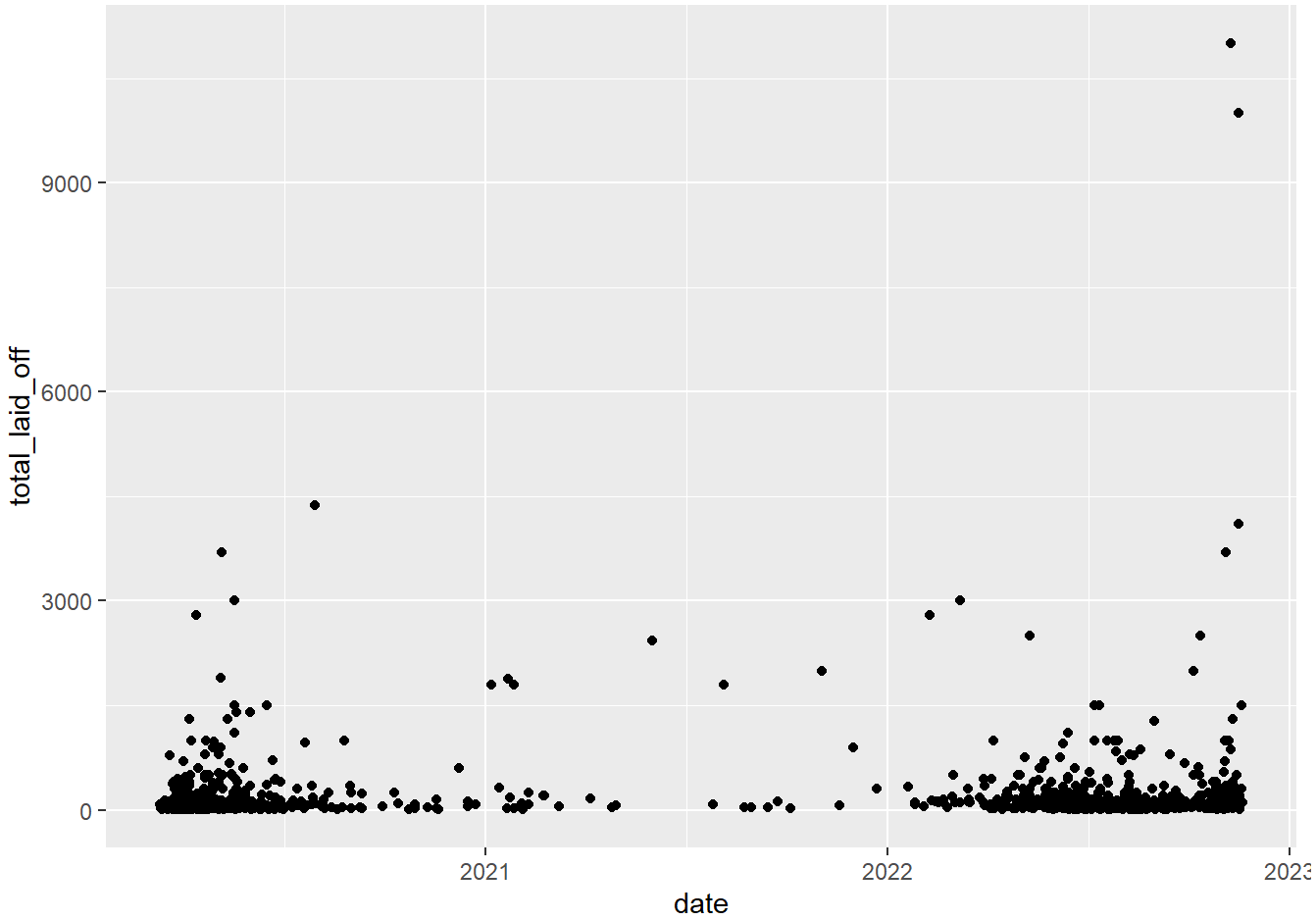
```
summary(df)
```

```
##    company            location           industry         total_laid_off
##  Length:1651        Length:1651        Length:1651        Min.   :    3.0
##  Class :character   Class :character   Class :character   1st Qu.:   31.0
##  Mode  :character   Mode  :character   Mode  :character   Median :   70.0
##                                                           Mean   :  198.3
##                                                           3rd Qu.:  150.0
##                                                           Max.   :11000.0
##                                                           NA's   :476
##  percentage_laid_off      date               stage             country
##  Min.   :0.0000      Min.   :2020-03-11   Length:1651        Length:1651
##  1st Qu.:0.1000      1st Qu.:2020-05-05   Class :character   Class :character
##  Median :0.1900      Median :2022-06-02   Mode  :character   Mode  :character
##  Mean   :0.2751      Mean   :2021-09-13
##  3rd Qu.:0.3200      3rd Qu.:2022-08-10
##  Max.   :1.0000      Max.   :2022-11-19
##  NA's   :546
##   funds_raised
##  Min.   :     0.0
##  1st Qu.:    42.0
##  Median :   129.0
##  Mean   :   890.7
##  3rd Qu.:   375.2
##  Max.   :121900.0
##  NA's   :115
```

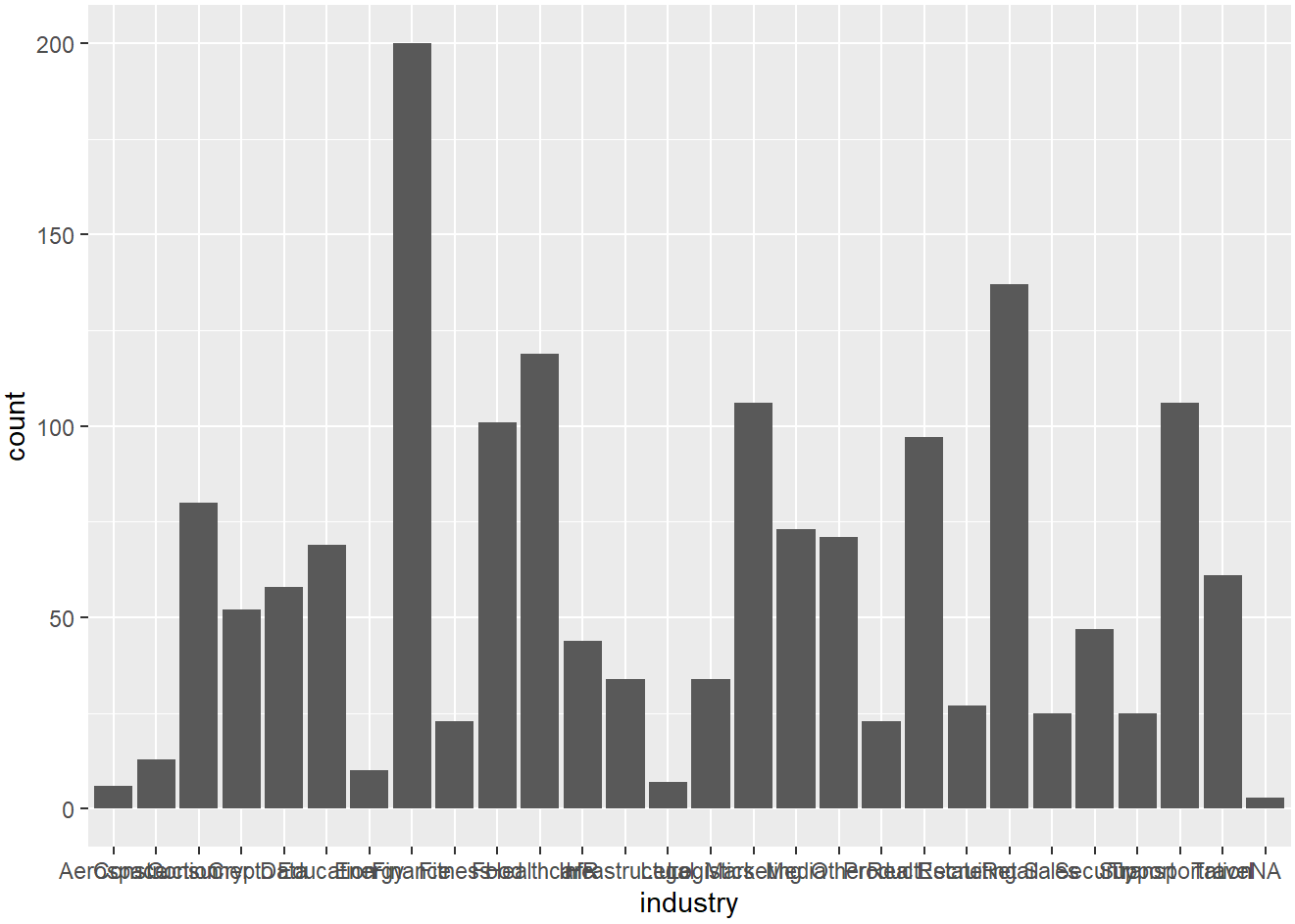
b. Data Exploration

```
#plots the the total number of layoffs over time using a scatter plot
ggplot(df, aes(date,total_laid_off)) + geom_point()
```

```
## Warning: Removed 476 rows containing missing values (geom_point).
```

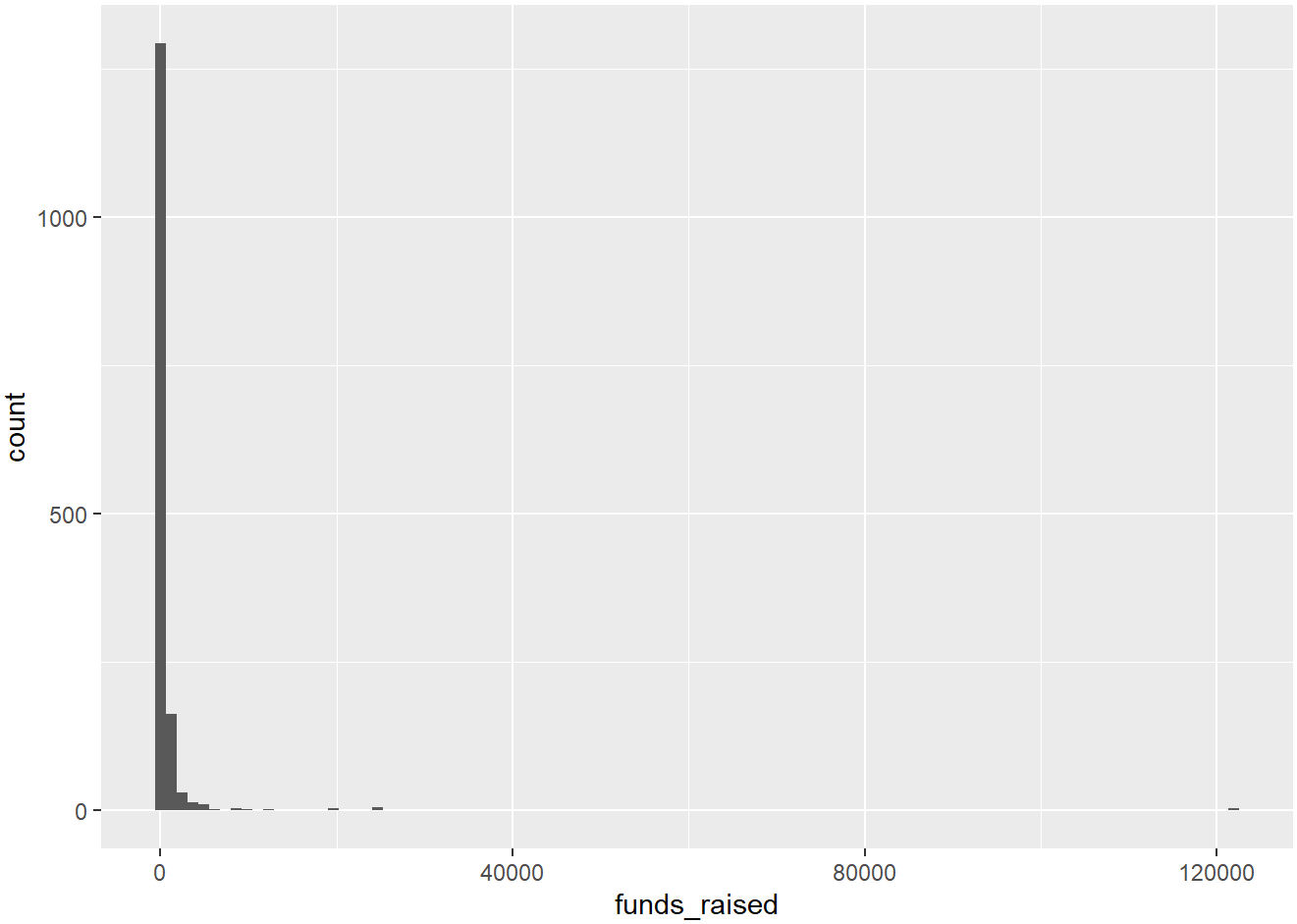



```
#plots the most commonly appearing industries in the data
ggplot(df, aes(x=industry)) + geom_bar()
```



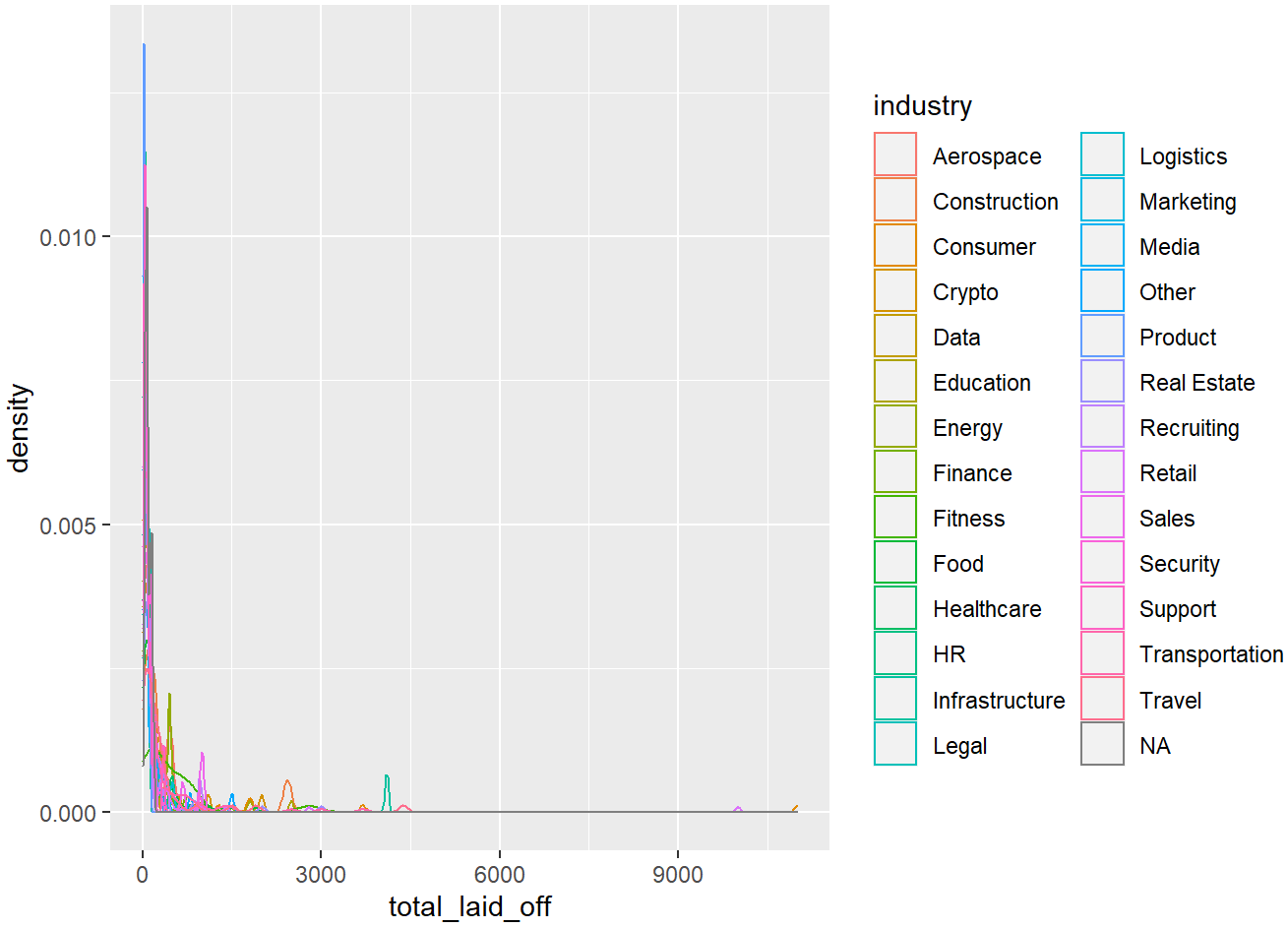

```
#plots a histogram of funds_raised using 100 bins
ggplot(df, aes(funds_raised)) + geom_histogram(bins = 100)
```

```
## Warning: Removed 115 rows containing non-finite values (stat_bin).
```




```
#density plot is used to show total number of layoffs per industry

denplot1 <- ggplot(df, aes(x=total_laid_off, color = industry)) +
geom_density()
denplot1
```

```
## Warning: Removed 476 rows containing non-finite values (stat_density).
```


density
0.010
0.005
0.000
0
3000
6000
9000
total_laid_off
industry
Aerospace
Construction
Consumer
Crypto
Data
Education
Energy
Finance
Fitness
Food
Healthcare
HR
Infrastructure
Legal
Logistics
Marketing
Media
Other
Product
Real Estate
Recruiting
Retail
Sales
Security
Support
Transportation
Travel
NA

```
summary(df)
```

```
##    company            location           industry         total_laid_off
##  Length:1651        Length:1651        Length:1651        Min.   :    3.0
##  Class :character   Class :character   Class :character   1st Qu.:   31.0
##  Mode  :character   Mode  :character   Mode  :character   Median :   70.0
##                                                           Mean   :  198.3
##                                                           3rd Qu.:  150.0
##                                                           Max.   :11000.0
##                                                           NA's   :476
##  percentage_laid_off      date               stage             country
##  Min.   :0.0000      Min.   :2020-03-11   Length:1651        Length:1651
##  1st Qu.:0.1000      1st Qu.:2020-05-05   Class :character   Class :character
##  Median :0.1900      Median :2022-06-02   Mode  :character   Mode  :character
##  Mean   :0.2751      Mean   :2021-09-13
##  3rd Qu.:0.3200      3rd Qu.:2022-08-10
##  Max.   :1.0000      Max.   :2022-11-19
##  NA's   :546
##   funds_raised
##  Min.   :     0.0
##  1st Qu.:    42.0
##  Median :   129.0
##  Mean   :   890.7
##  3rd Qu.:   375.2
##  Max.   :121900.0
##  NA's   :115
```

c. Data Cleaning and d. Data Preprocessing

```
#check for missing values
df2 <- df
summary(df2$company)
```

```
##    Length     Class      Mode
##      1651 character character
```

```
summary(df2$location)
```

```
##    Length     Class      Mode
##      1651 character character
```

```
summary(df2$industry)
```

```
##    Length     Class      Mode
##      1651 character character
```

```
summary(df2$total_laid_off)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     3.0    31.0    70.0   198.3   150.0 11000.0     476
```

```
summary(df2$percentage_laid_off)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##  0.0000  0.1000  0.1900  0.2751  0.3200  1.0000     546
```

```
summary(df2$date)
```

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
## "2020-03-11" "2020-05-05" "2022-06-02" "2021-09-13" "2022-08-10" "2022-11-19"
```

```
summary(df2$stage)
```

```
##    Length     Class      Mode
##      1651 character character
```

```
summary(df2$country)
```

```
##    Length     Class      Mode
##      1651 character character
```

```
summary(df2$funds_raised)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
##      0.0     42.0    129.0    890.7    375.2 121900.0      115
```

```
library(dbplyr)
```

```
##
## Attaching package: 'dbplyr'
```

```
## The following objects are masked from 'package:dplyr':
##
##     ident, sql
```

```
df2 <- df2 %>% drop_na(total_laid_off)
summary(df2$total_laid_off)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     3.0    31.0    70.0   198.3   150.0 11000.0
```

```
df2 <- df2 %>% drop_na(percentage_laid_off)
summary(df2$percentage_laid_off)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.100   0.180   0.244   0.300   1.000
```

```
df2 <- df2 %>% drop_na(funds_raised)
summary(df2$funds_raised)
```

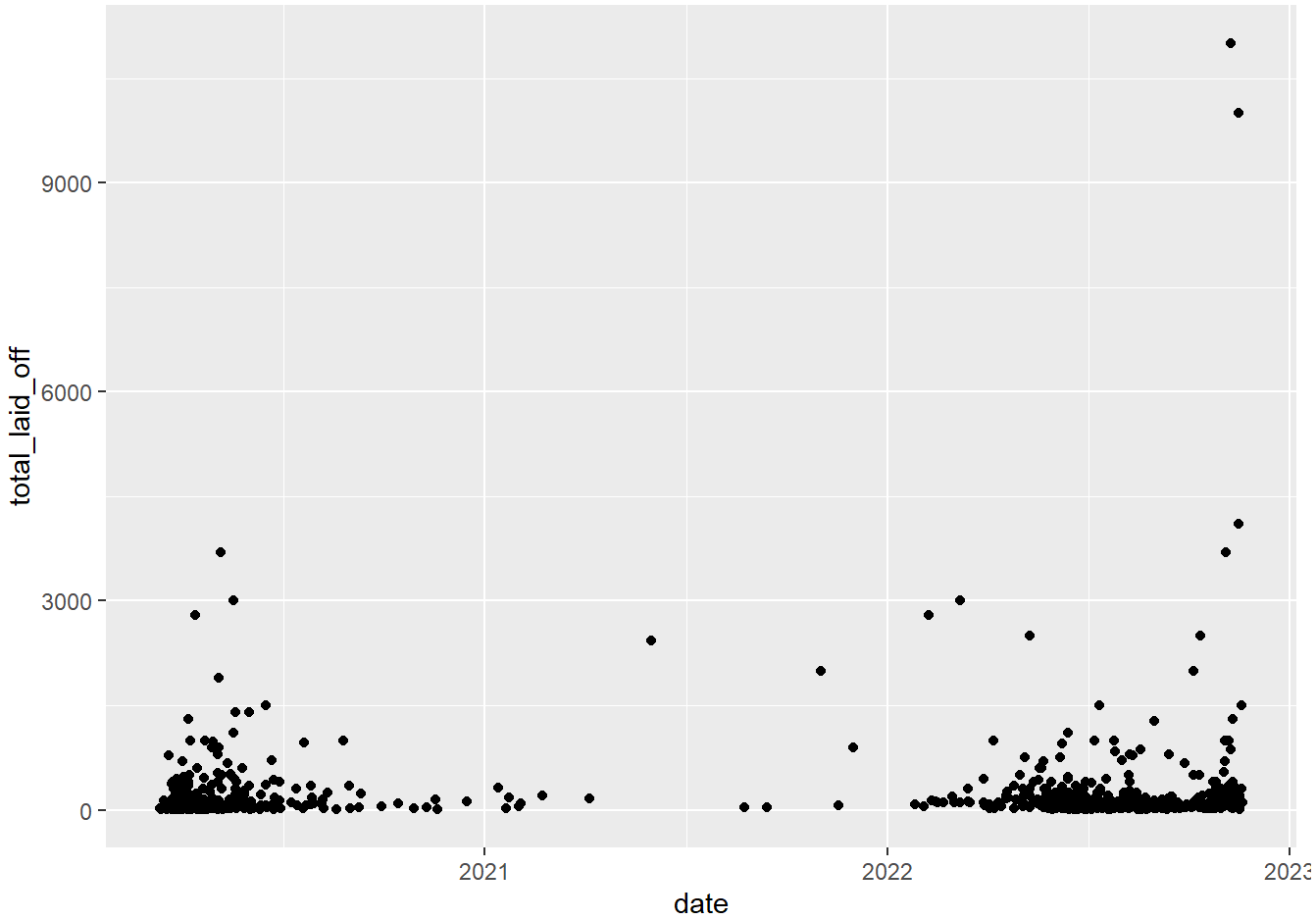
```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##      0.0     50.0    149.5    905.4    423.0 121900.0
```

```
df2 <- df2 %>% drop_na(industry)
summary(df2$industry)
```

```
##    Length     Class      Mode
##       813 character character
```

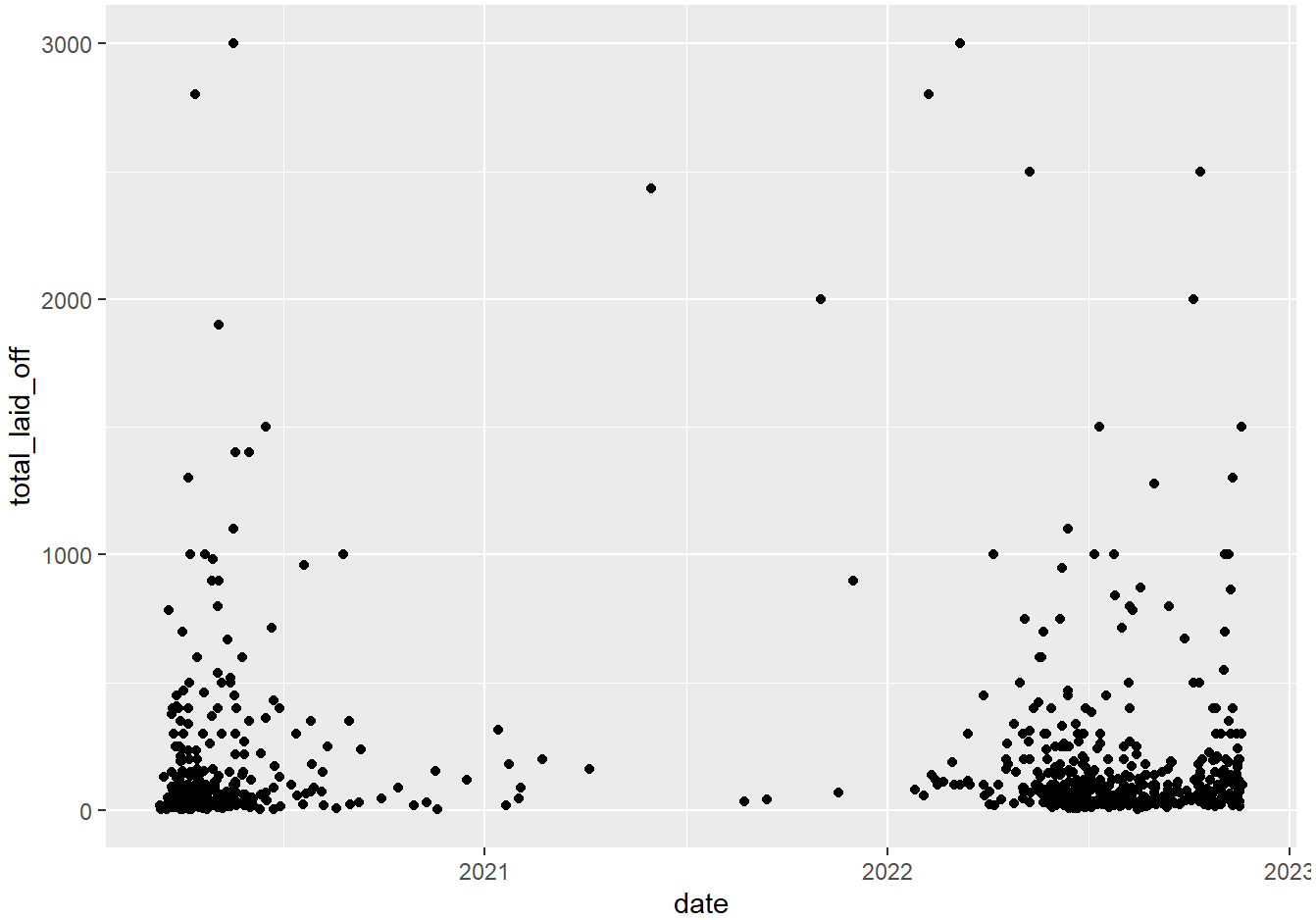
```
df2 <- df2 %>% drop_na(company)
df2 <- df2 %>% drop_na(location)
df2 <- df2 %>% drop_na(stage)
df2 <- df2 %>% drop_na(date)
df2 <- df2 %>% drop_na(country)
```

```
#scatter plot still difficult to see
ggplot(df2, aes(date,total_laid_off)) + geom_point()
```

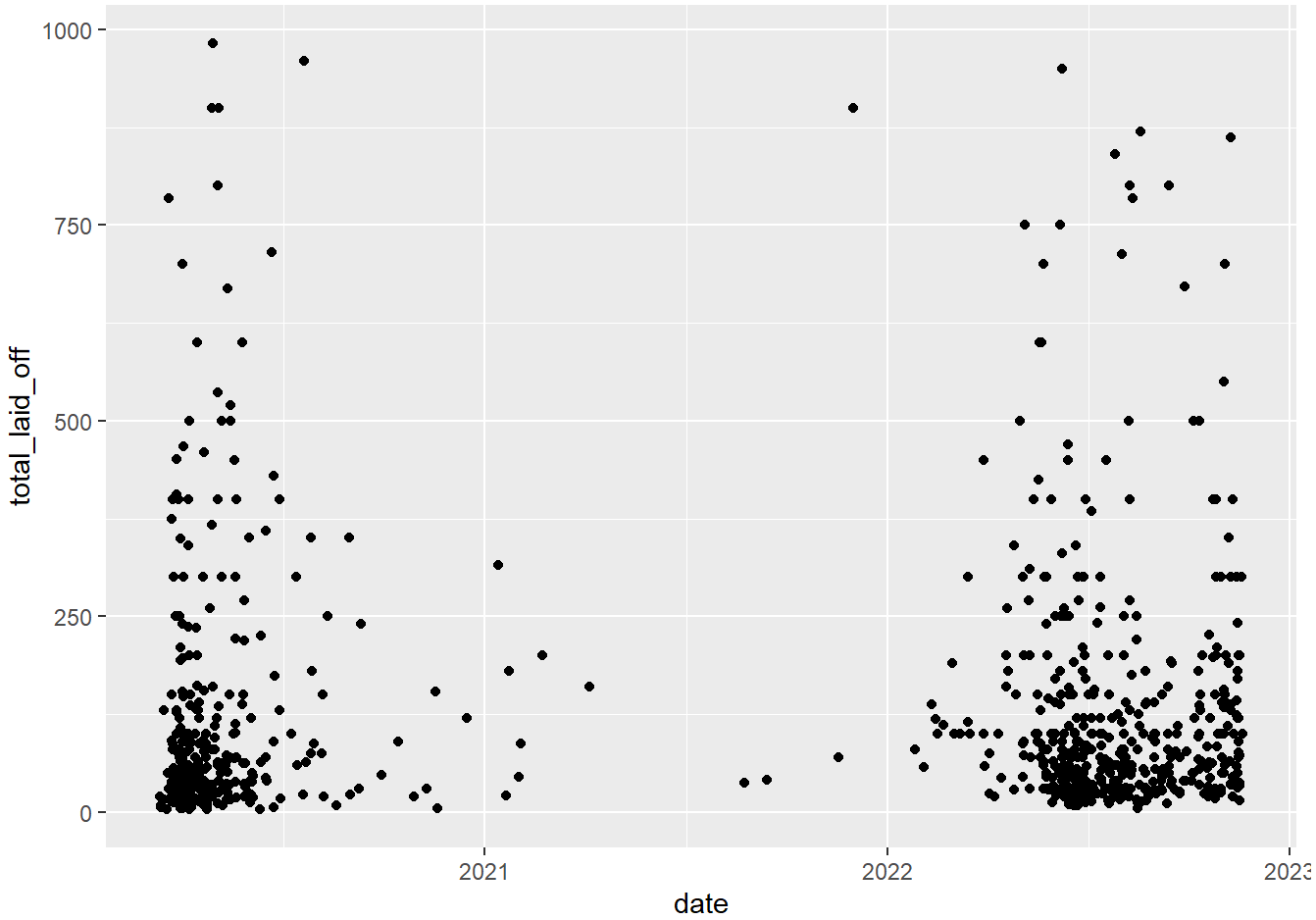



```
#remove absurd outliers

df3 <- df2
df3 <- df3 %>% filter(total_laid_off <= 3000)
```

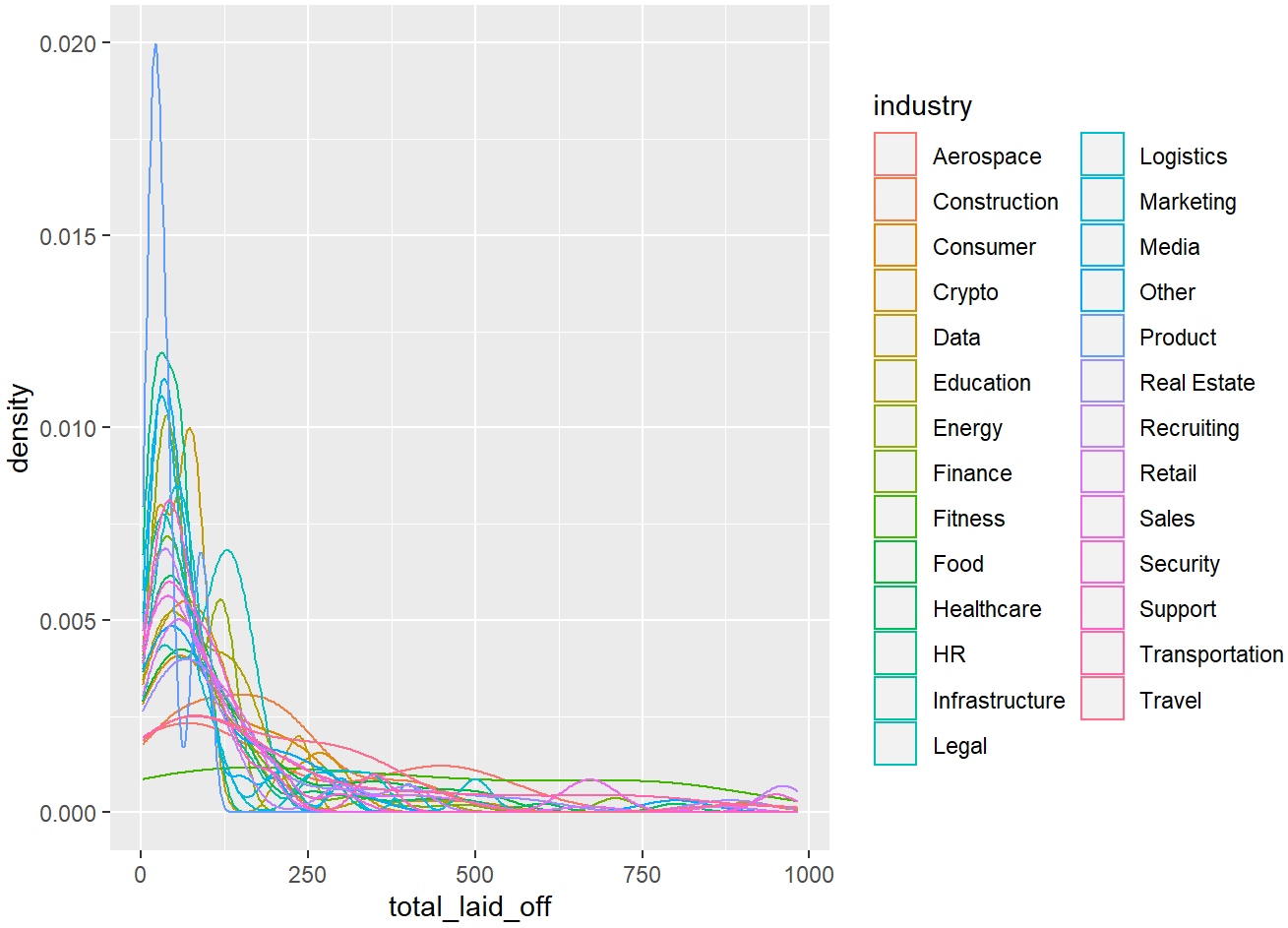
```
#still some outliers present

ggplot(df3, aes(date,total_laid_off)) + geom_point()
```




```
#scatter plot now looks a lot more readable
df4 <- df3
df4 <- df4 %>% filter(total_laid_off < 1000)
ggplot(df4, aes(date,total_laid_off)) + geom_point()
```



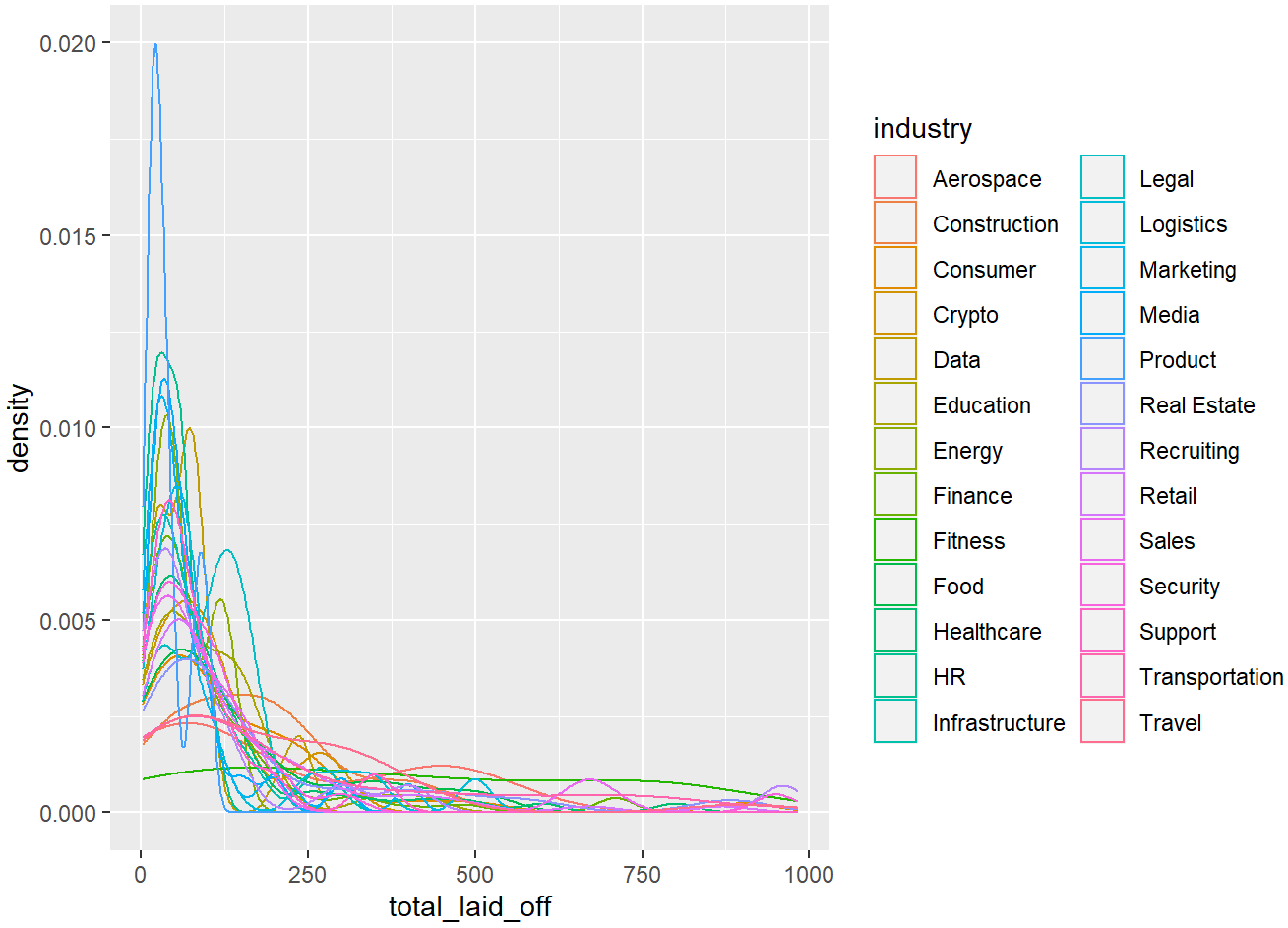

```
#density plot slightly more readable but too many categories present and a "other" category p
resent

denplot2 <- ggplot(df4, aes(x=total_laid_off, color = industry)) +
geom_density()
denplot2
```



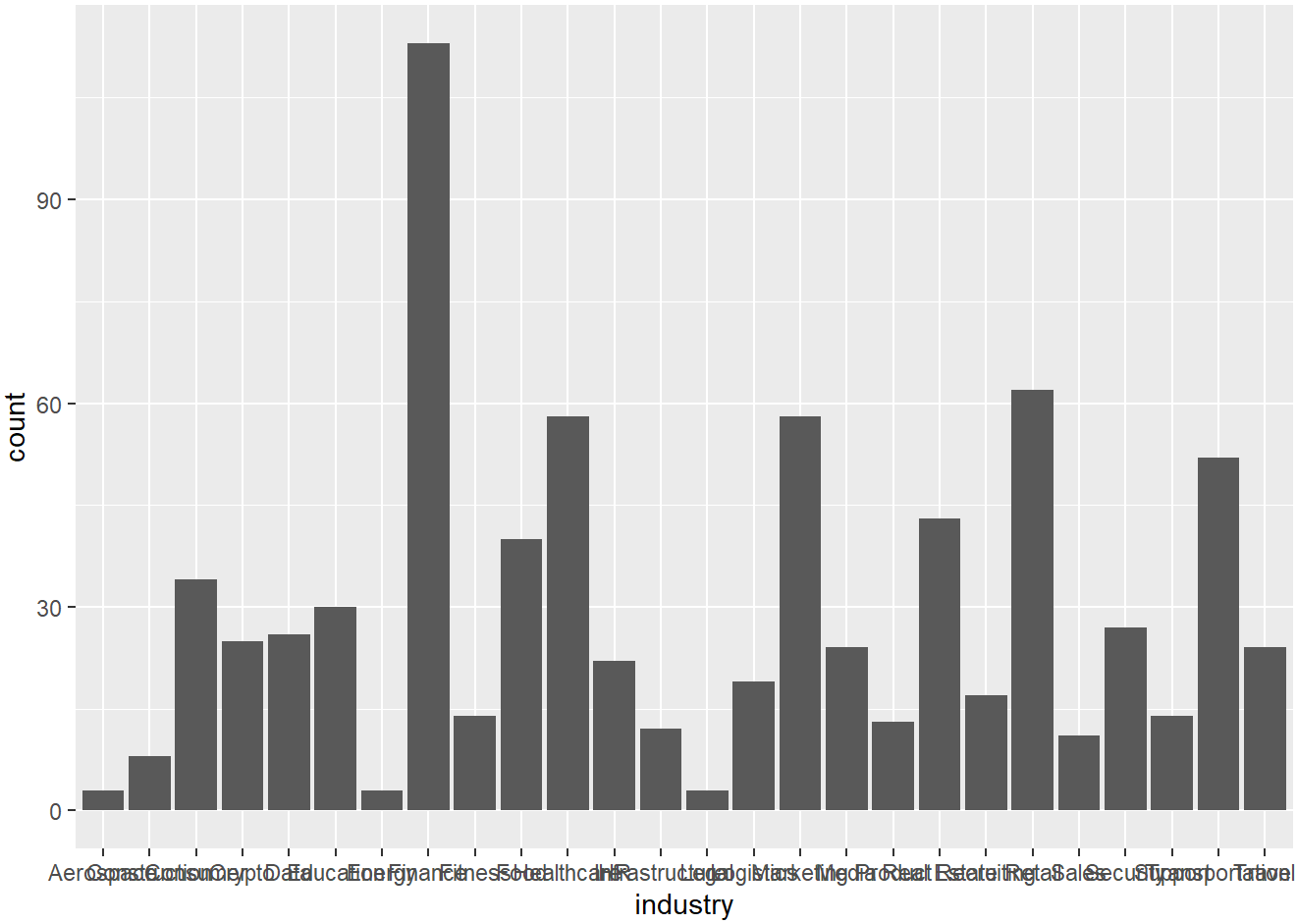

```
df5 <- df4
df5 <- df5 %>% filter(industry != 'Other')
summary(df5)
```

```
##    company            location           industry         total_laid_off
##  Length:755         Length:755         Length:755         Min.   :  3.0
##  Class :character   Class :character   Class :character   1st Qu.: 32.5
##  Mode  :character   Mode  :character   Mode  :character   Median : 70.0
##                                                           Mean   :130.4
##                                                           3rd Qu.:150.0
##                                                           Max.   :982.0
##  percentage_laid_off      date               stage             country
##  Min.   :0.0000      Min.   :2020-03-12   Length:755         Length:755
##  1st Qu.:0.1000      1st Qu.:2020-04-24   Class :character   Class :character
##  Median :0.1800      Median :2022-05-23   Mode  :character   Mode  :character
##  Mean   :0.2412      Mean   :2021-08-14
##  3rd Qu.:0.3000      3rd Qu.:2022-07-31
##  Max.   :1.0000      Max.   :2022-11-19
##   funds_raised
##  Min.   :     0.0
##  1st Qu.:    48.0
##  Median :   143.0
##  Mean   :   789.6
##  3rd Qu.:   377.5
##  Max.   :121900.0
```

```
#no more 'other' category but still too many industries shown
denplot3 <- ggplot(df5, aes(x=total_laid_off, color = industry)) +
geom_density()
denplot3
```




```
#too many categories present

ggplot(df5, aes(x=industry)) + geom_bar()
```

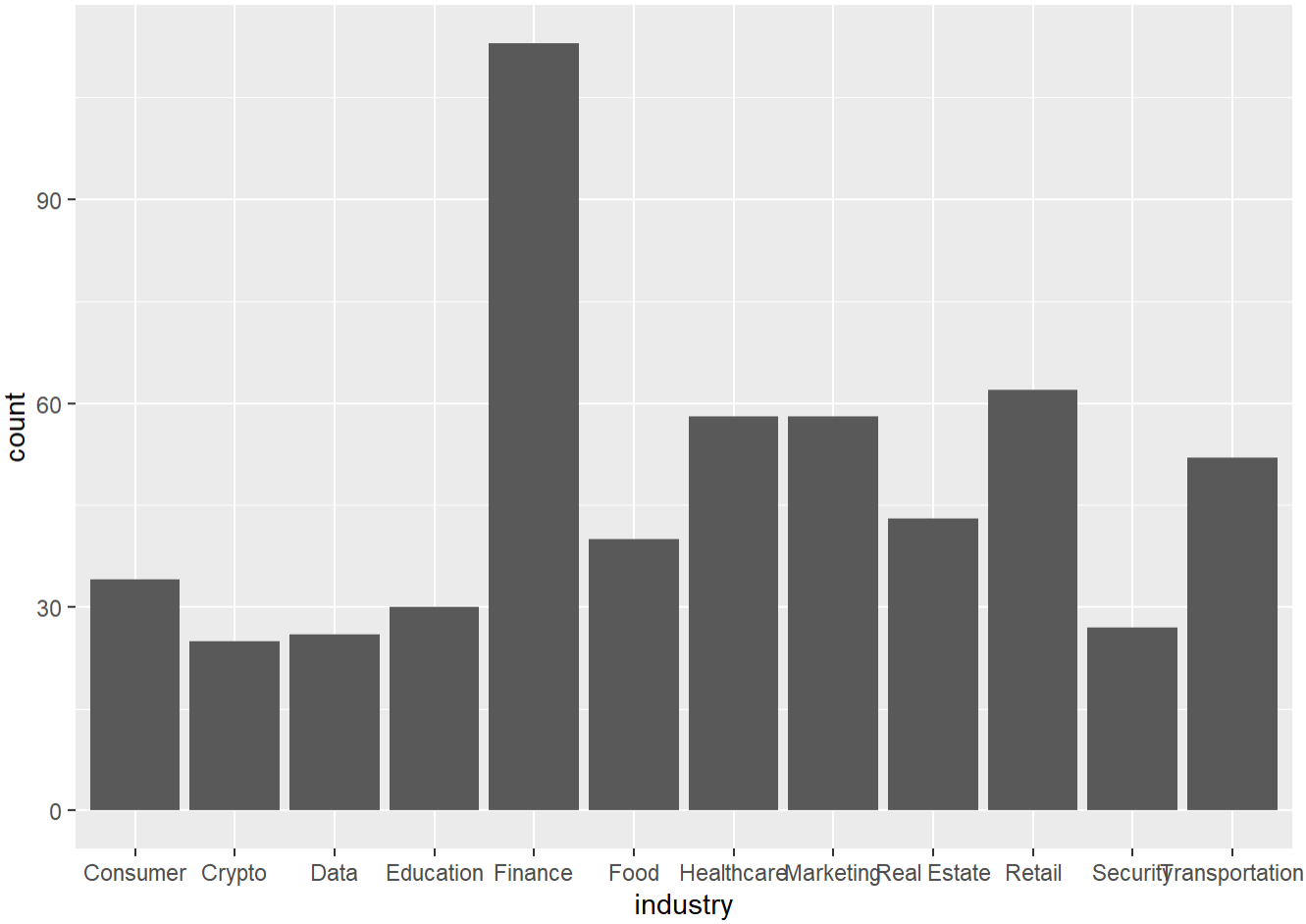



```
#check the frequency of how often each category appears in the data
table(df5$industry)
```

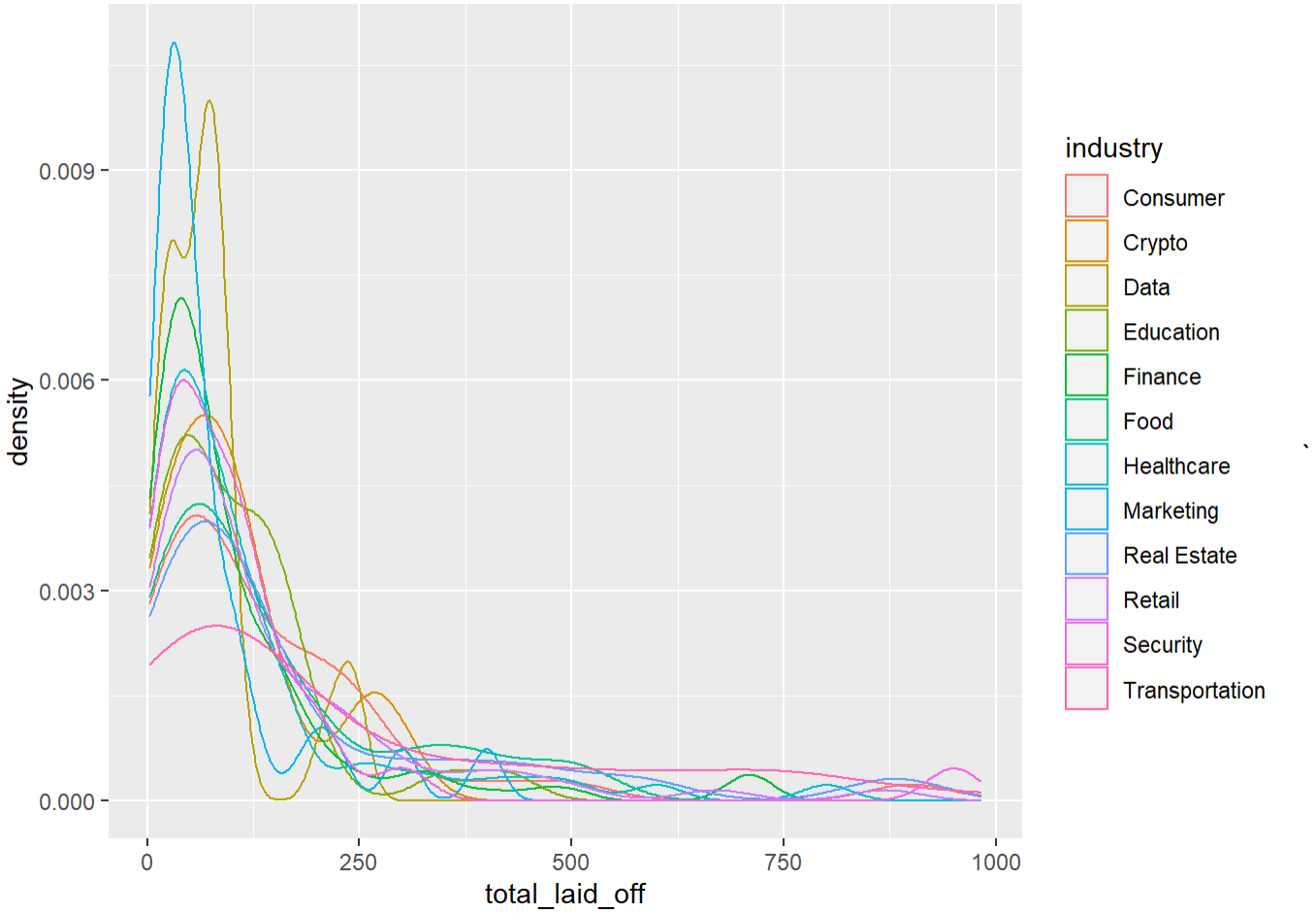
```
##
##      Aerospace   Construction       Consumer         Crypto           Data
##              3              8             34             25             26
##      Education         Energy        Finance        Fitness           Food
##             30              3            113             14             40
##     Healthcare             HR Infrastructure          Legal      Logistics
##             58             22             12              3             19
##      Marketing          Media        Product    Real Estate     Recruiting
##             58             24             13             43             17
##         Retail          Sales       Security        Support Transportation
##             62             11             27             14             52
##         Travel
##             24
```

```
#remove categories that appear less than 25 times in the data
df6 <- df5
df6 <- df6 %>% filter(industry != "Aerospace")
df6 <- df6 %>% filter(industry != "Construction")
df6 <- df6 %>% filter(industry != "Energy")
df6 <- df6 %>% filter(industry != "Fitness")
df6 <- df6 %>% filter(industry != "HR")
df6 <- df6 %>% filter(industry != "Infrastructure")
df6 <- df6 %>% filter(industry != "Legal")
df6 <- df6 %>% filter(industry != "Logistics")
df6 <- df6 %>% filter(industry != "Media")
df6 <- df6 %>% filter(industry != "Product")
df6 <- df6 %>% filter(industry != "Recruiting")
df6 <- df6 %>% filter(industry != "Sales")
df6 <- df6 %>% filter(industry != "Support")
df6 <- df6 %>% filter(industry != "Travel")
```

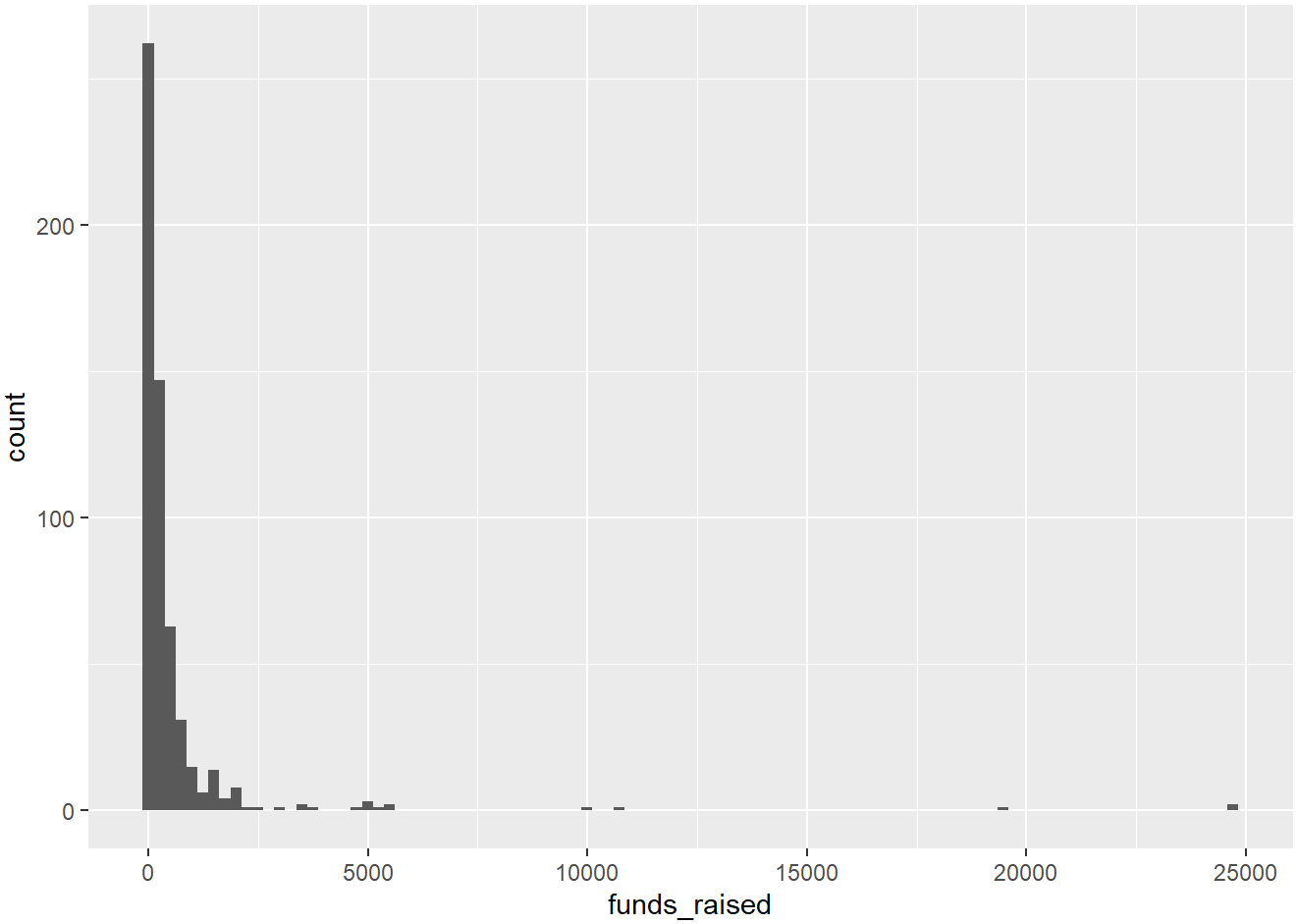
```
#bar chart in now a lot more readable

ggplot(df6, aes(x=industry)) + geom_bar()
```



```
#density plot in now a lot more readable

denplot4 <- ggplot(df6, aes(x=total_laid_off, color = industry)) +
geom_density()
denplot4
```

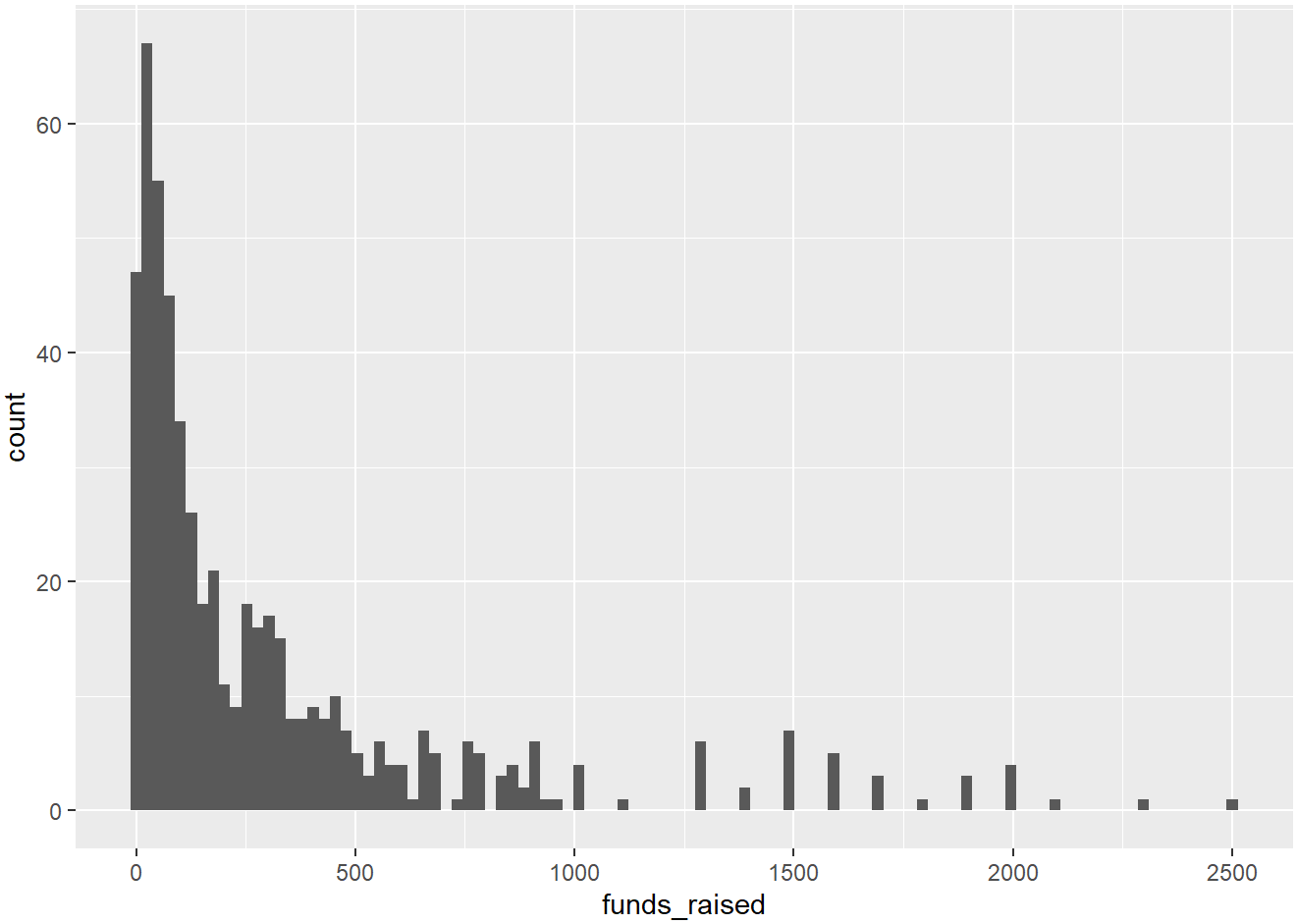



```
ggplot(df6, aes(funds_raised)) + geom_histogram(bins = 100)
```

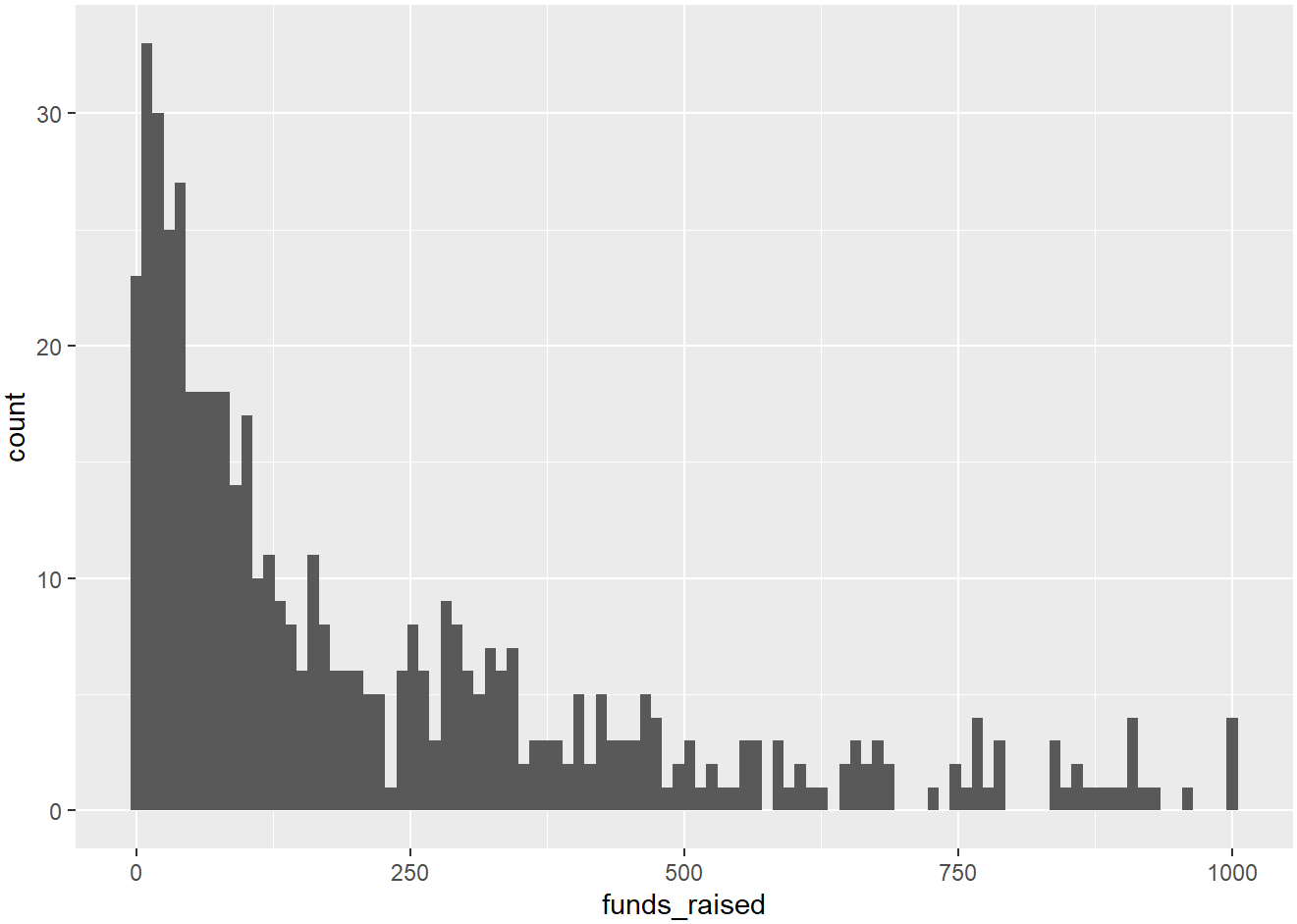



```
#remove absurd outliers

df7 <- df6
df7 <- df7 %>% filter(funds_raised <= 3000)
```

```
#data still contains some outliers
ggplot(df7, aes(funds_raised)) + geom_histogram(bins = 100)
```




```
#data is now much more readable

df7 <- df7 %>% filter(funds_raised <= 1000)
ggplot(df7, aes(funds_raised)) + geom_histogram(bins = 100)
```




```
#517 columns remain after removing outliers, missing, and unnecessary rows
summary(df7)
```

```
##    company            location           industry         total_laid_off
##  Length:517         Length:517         Length:517         Min.   :  3.0
##  Class :character   Class :character   Class :character   1st Qu.: 30.0
##  Mode  :character   Mode  :character   Mode  :character   Median : 69.0
##                                                           Mean   :108.1
##                                                           3rd Qu.:130.0
##                                                           Max.   :950.0
##  percentage_laid_off      date               stage             country
##  Min.   :0.0100      Min.   :2020-03-12   Length:517         Length:517
##  1st Qu.:0.1000      1st Qu.:2020-04-28   Class :character   Class :character
##  Median :0.2000      Median :2022-05-27   Mode  :character   Mode  :character
##  Mean   :0.2531      Mean   :2021-08-31
##  3rd Qu.:0.3000      3rd Qu.:2022-07-27
##  Max.   :1.0000      Max.   :2022-11-19
##   funds_raised
##  Min.   :   0.0
##  1st Qu.:  42.0
##  Median : 120.0
##  Mean   : 220.7
##  3rd Qu.: 320.0
##  Max.   :1000.0
```

```
#funds_raised does not display meaningful data as unit of measure is not provided
#normalizing this variable to 0-1 range will allow for better understanding of this data

library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
df8a <- df7
preproc <- preProcess(df8a, method=c("range"))
norm <- predict(preproc, df8a)

summary(norm)
```

```
##    company            location           industry         total_laid_off
##  Length:517         Length:517         Length:517         Min.   :0.00000
##  Class :character   Class :character   Class :character   1st Qu.:0.02851
##  Mode  :character   Mode  :character   Mode  :character   Median :0.06969
##                                                           Mean   :0.11096
##                                                           3rd Qu.:0.13411
##                                                           Max.   :1.00000
##  percentage_laid_off      date               stage             country
##  Min.   :0.00000     Min.   :2020-03-12   Length:517         Length:517
##  1st Qu.:0.09091     1st Qu.:2020-04-28   Class :character   Class :character
##  Median :0.19192     Median :2022-05-27   Mode  :character   Mode  :character
##  Mean   :0.24553     Mean   :2021-08-31
##  3rd Qu.:0.29293     3rd Qu.:2022-07-27
##  Max.   :1.00000     Max.   :2022-11-19
##   funds_raised
##  Min.   :0.0000
##  1st Qu.:0.0420
##  Median :0.1200
##  Mean   :0.2207
##  3rd Qu.:0.3200
##  Max.   :1.0000
```

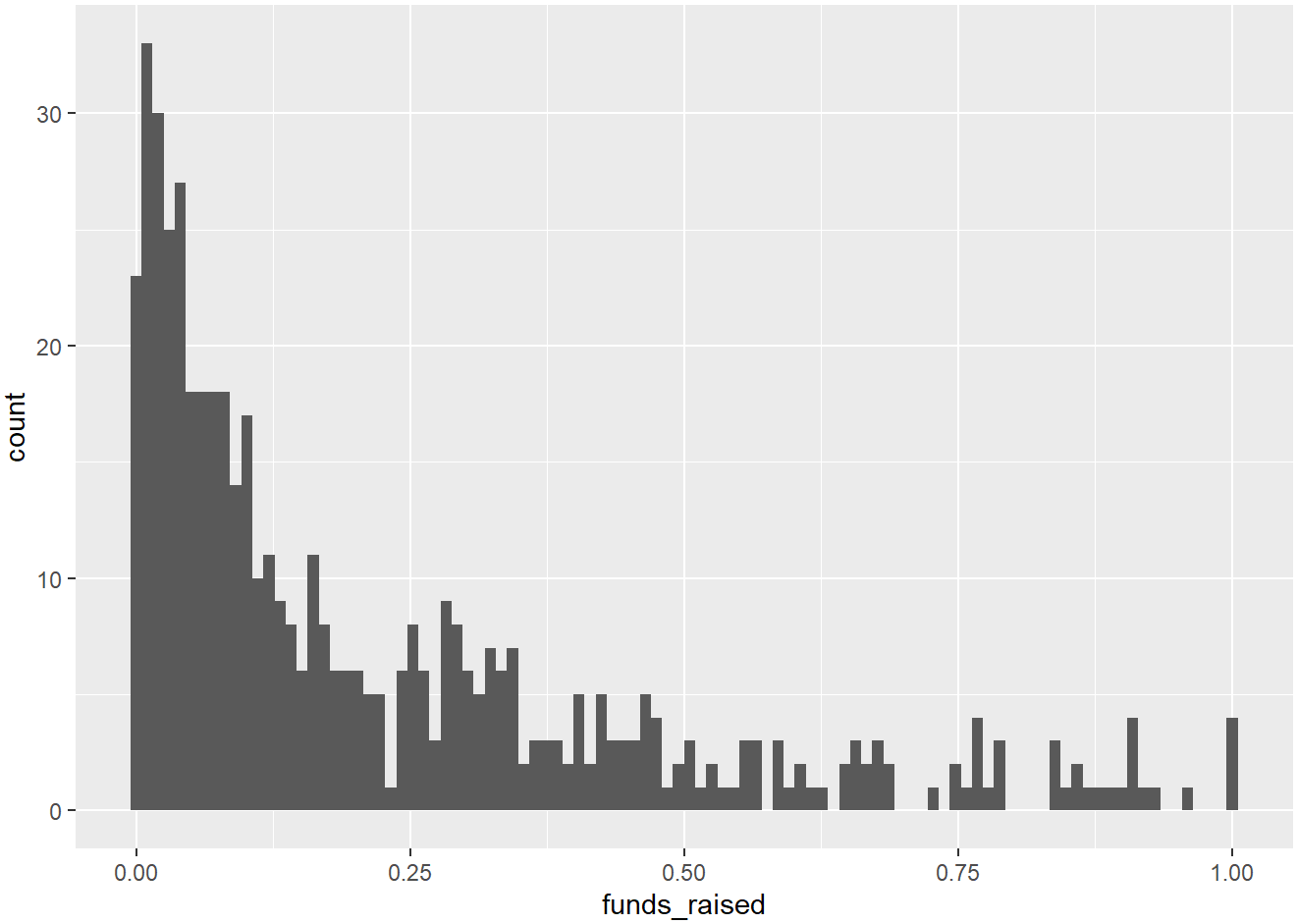
```
#old funds_raised is removed from dataset
df8 <- df7
df8 <- df8 %>% select(-c( "funds_raised"))
```

```
#old funds_raised is replaced by the normalized version
df8$funds_raised <- norm$funds_raised
```

```
summary(df8)
```

```
##    company            location           industry         total_laid_off
##  Length:517         Length:517         Length:517         Min.   :  3.0
##  Class :character   Class :character   Class :character   1st Qu.: 30.0
##  Mode  :character   Mode  :character   Mode  :character   Median : 69.0
##                                                           Mean   :108.1
##                                                           3rd Qu.:130.0
##                                                           Max.   :950.0
##  percentage_laid_off      date               stage             country
##  Min.   :0.0100      Min.   :2020-03-12   Length:517         Length:517
##  1st Qu.:0.1000      1st Qu.:2020-04-28   Class :character   Class :character
##  Median :0.2000      Median :2022-05-27   Mode  :character   Mode  :character
##  Mean   :0.2531      Mean   :2021-08-31
##  3rd Qu.:0.3000      3rd Qu.:2022-07-27
##  Max.   :1.0000      Max.   :2022-11-19
##   funds_raised
##  Min.   :0.0000
##  1st Qu.:0.0420
##  Median :0.1200
##  Mean   :0.2207
##  3rd Qu.:0.3200
##  Max.   :1.0000
```

```
#the histogram now displays more meaningful info
ggplot(df8, aes(funds_raised)) + geom_histogram(bins = 100)
```

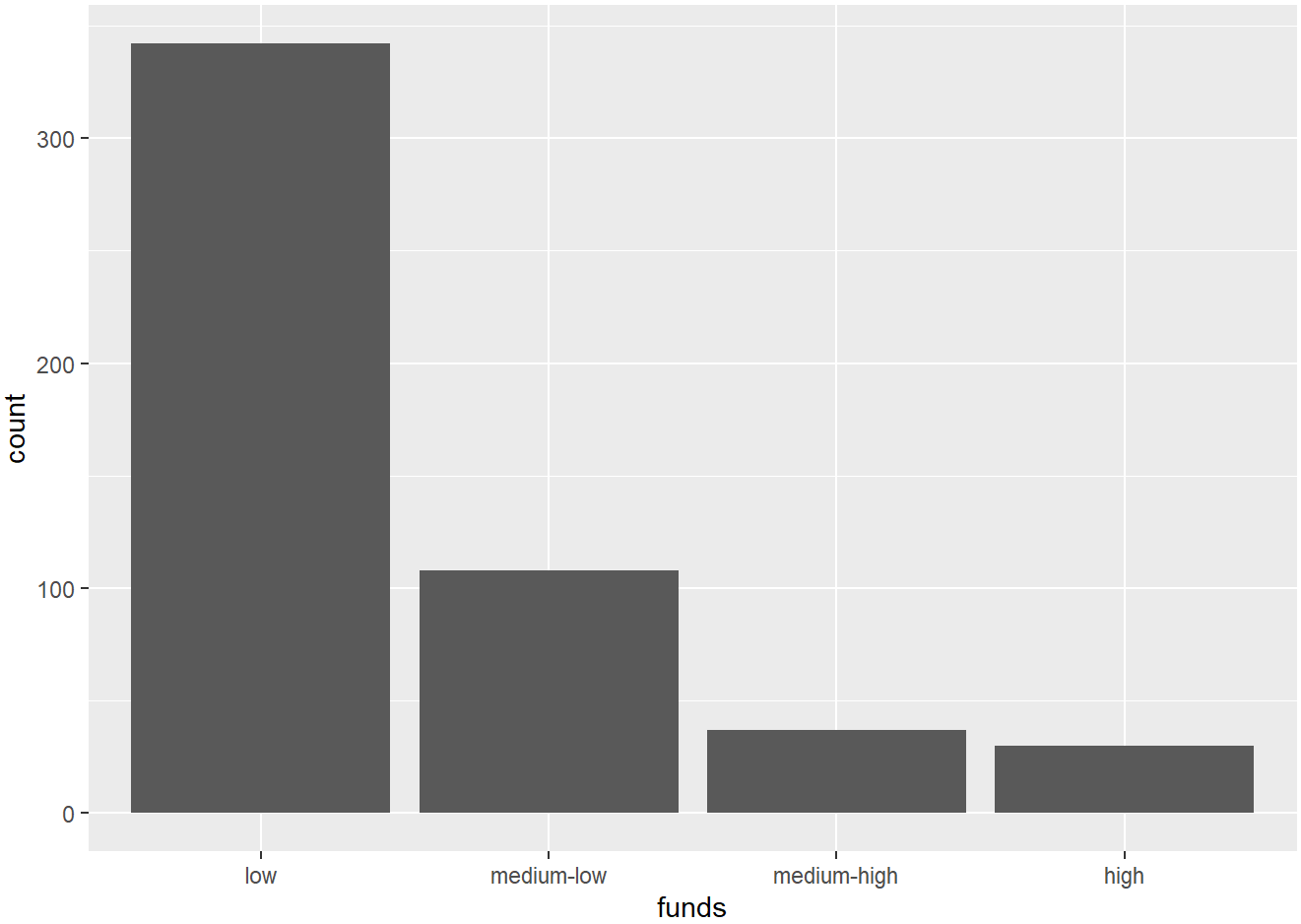



```
#data is further smoothed and divided into 4 bins in order to provide more consise informatio
n
df9 <- df8 %>%
mutate(funds = cut(funds_raised, breaks = 4,
labels=c("low","medium-low","medium-high","high")))
```

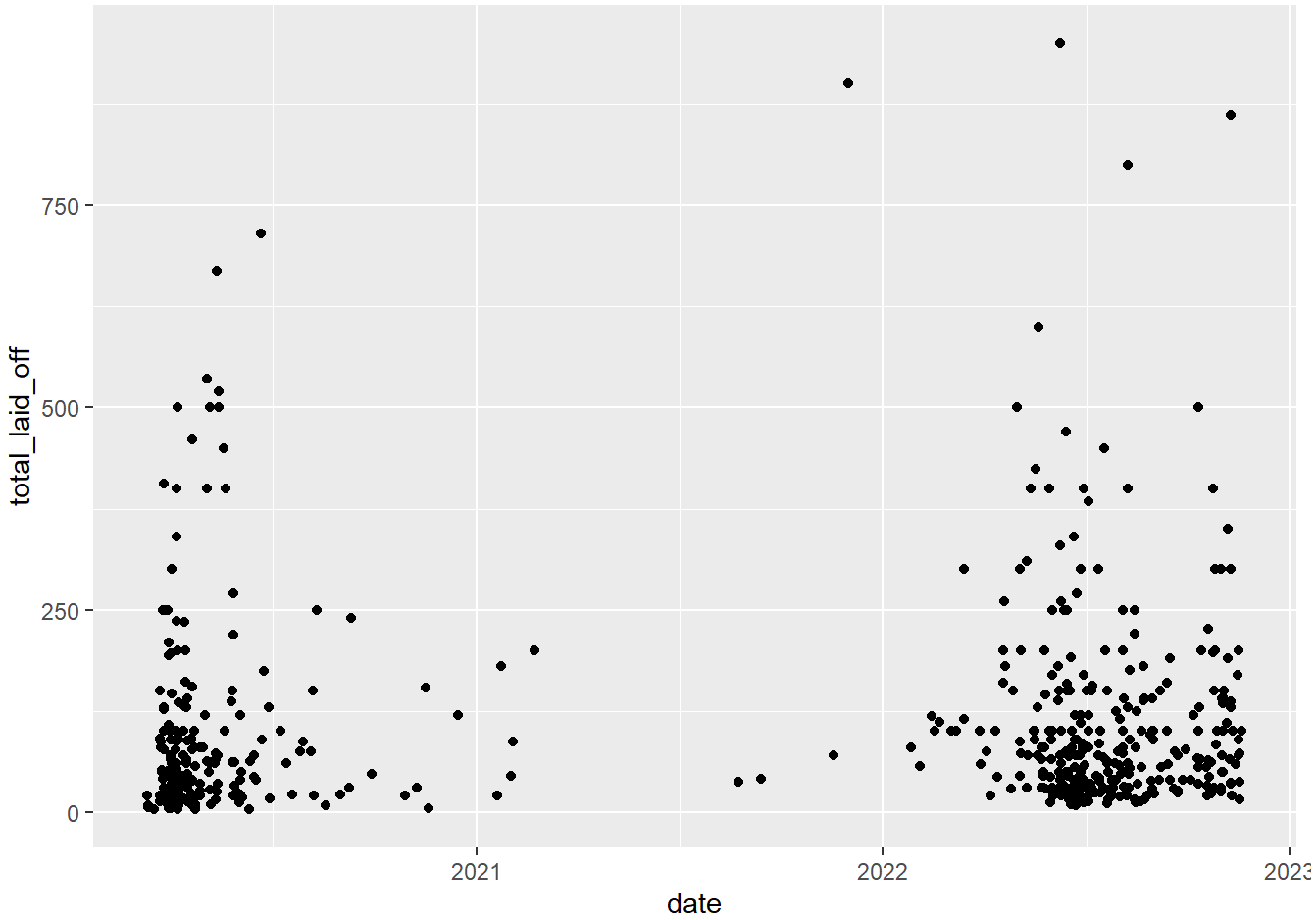
```
df9 <- df9 %>% select(-c( "funds_raised"))
summary(df9)
```

```
##    company            location           industry         total_laid_off
##  Length:517         Length:517         Length:517         Min.   :  3.0
##  Class :character   Class :character   Class :character   1st Qu.: 30.0
##  Mode  :character   Mode  :character   Mode  :character   Median : 69.0
##                                                           Mean   :108.1
##                                                           3rd Qu.:130.0
##                                                           Max.   :950.0
##  percentage_laid_off      date               stage             country
##  Min.   :0.0100      Min.   :2020-03-12   Length:517         Length:517
##  1st Qu.:0.1000      1st Qu.:2020-04-28   Class :character   Class :character
##  Median :0.2000      Median :2022-05-27   Mode  :character   Mode  :character
##  Mean   :0.2531      Mean   :2021-08-31
##  3rd Qu.:0.3000      3rd Qu.:2022-07-27
##  Max.   :1.0000      Max.   :2022-11-19
##          funds
##  low        :342
##  medium-low :108
##  medium-high: 37
##  high       : 30
##
##
```

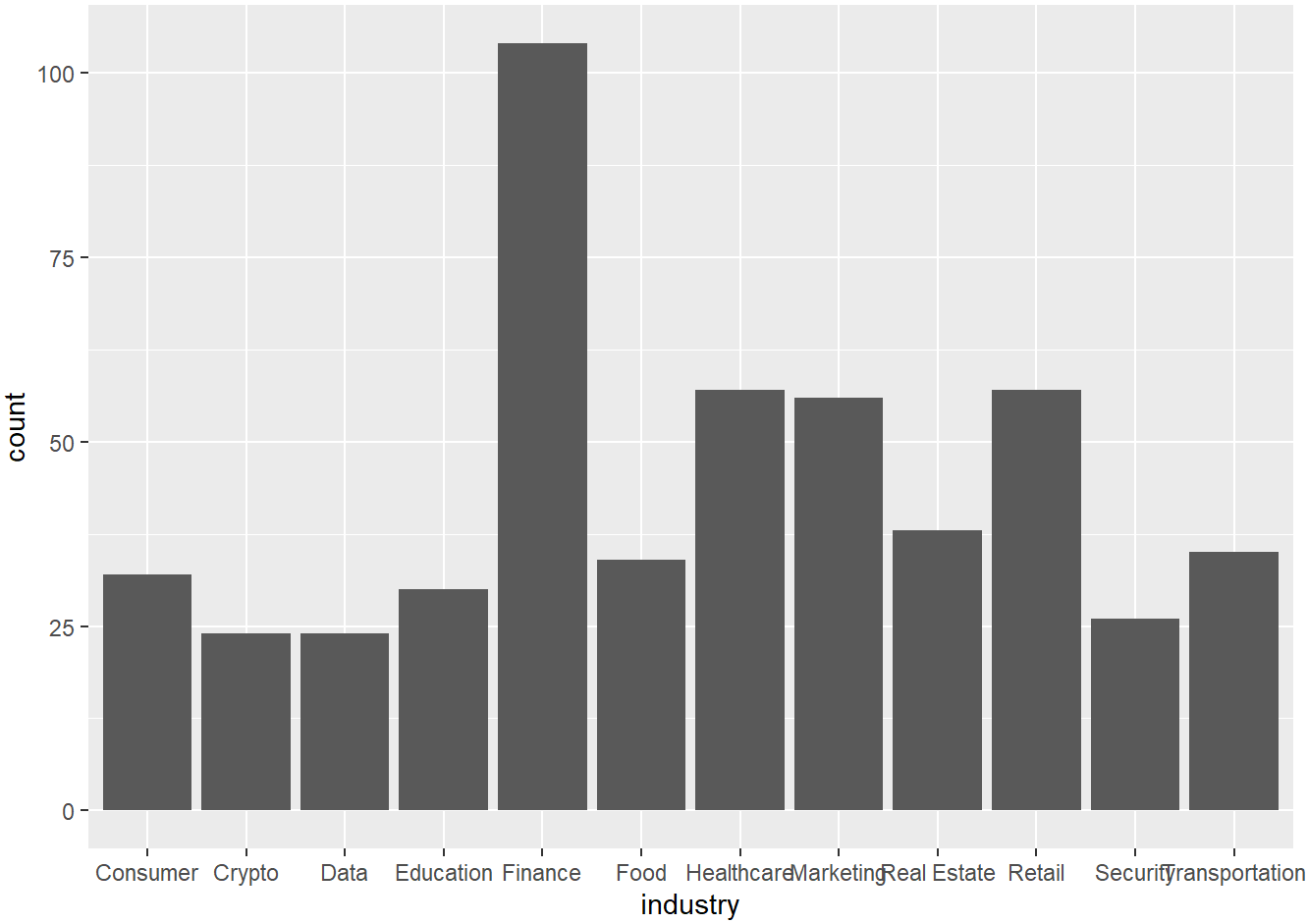
```
#a bar chart can now be used to plot the funds data

ggplot(df9, aes(x=funds)) + geom_bar()
```



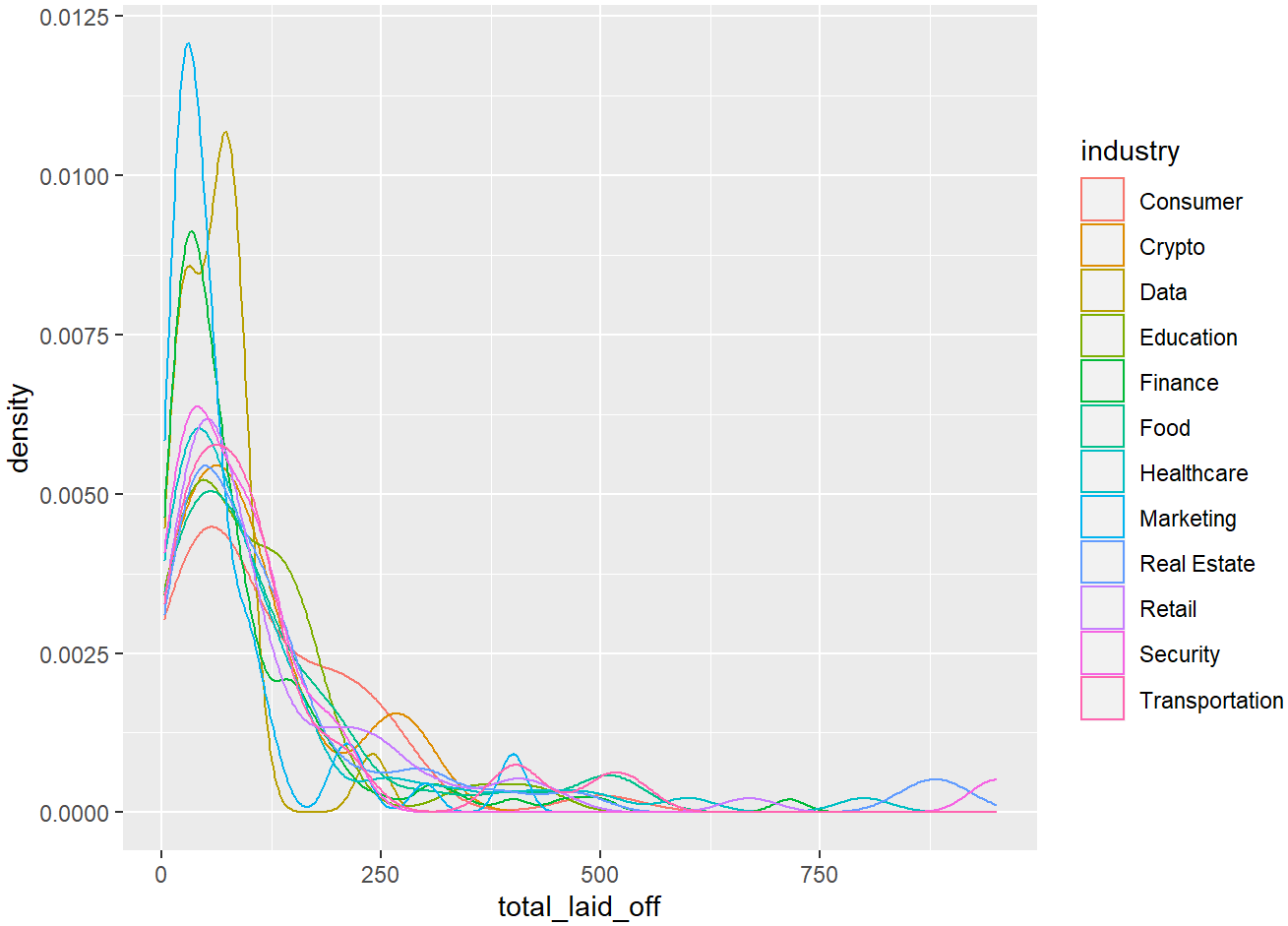

```
#after some data cleaning and some pre-processing, the final charts appear as follow

ggplot(df9, aes(date,total_laid_off)) + geom_point()
```



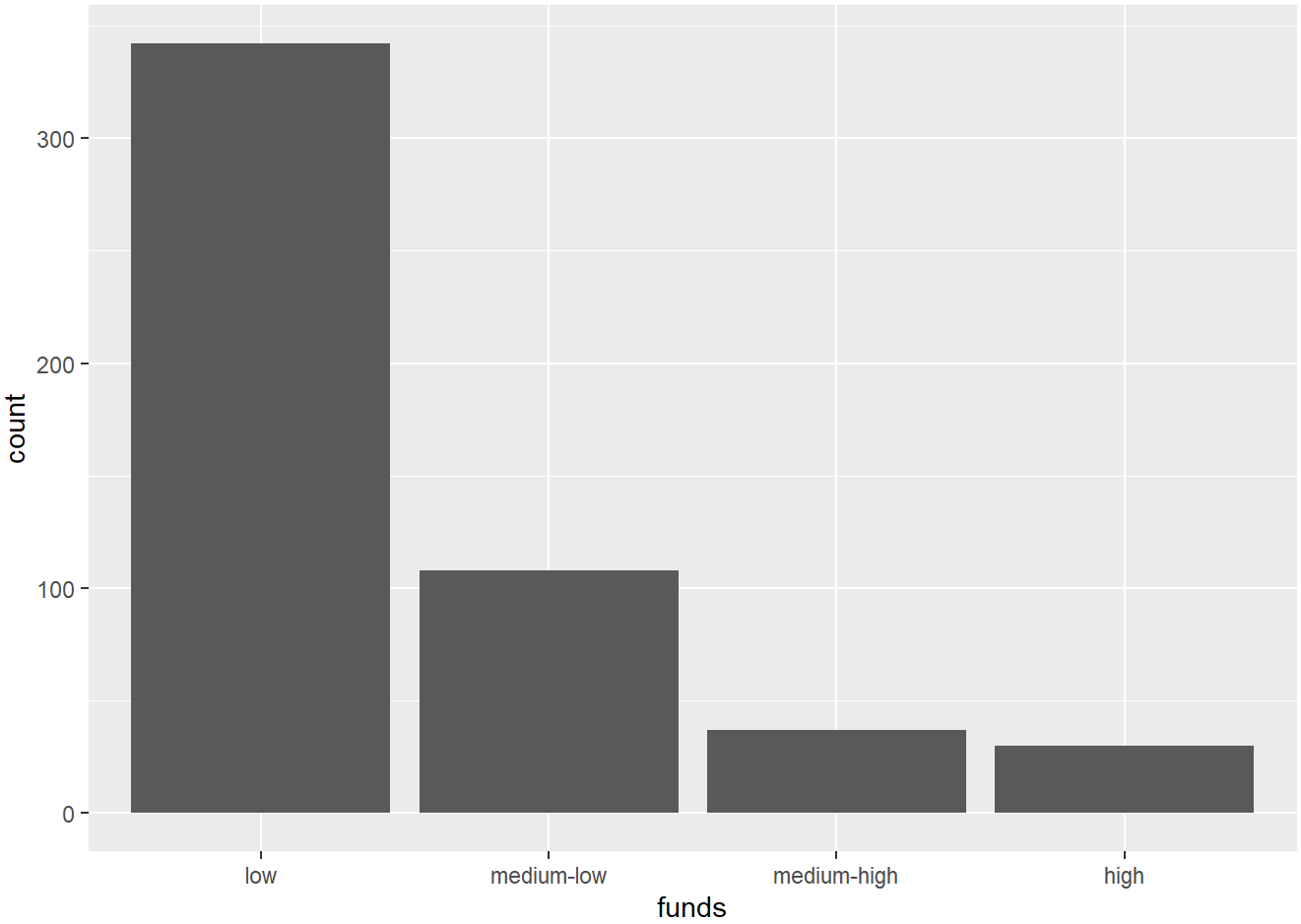

```
ggplot(df9, aes(x=industry)) + geom_bar()
```




```
ggplot(df9, aes(x=total_laid_off, color = industry)) + geom_density()
```




```
ggplot(df9, aes(x=funds)) + geom_bar()
```




```
#There are many columns of the data that don't have much meaningful relevance to the topic of interest and are mainly categorical. Before proceeding to further steps, it will be best to remove the columns that are unnecessary.

summary(df9)
```

```
##    company            location           industry         total_laid_off
##  Length:517         Length:517         Length:517         Min.   :  3.0
##  Class :character   Class :character   Class :character   1st Qu.: 30.0
##  Mode  :character   Mode  :character   Mode  :character   Median : 69.0
##                                                           Mean   :108.1
##                                                           3rd Qu.:130.0
##                                                           Max.   :950.0
##  percentage_laid_off      date               stage             country
##  Min.   :0.0100      Min.   :2020-03-12   Length:517         Length:517
##  1st Qu.:0.1000      1st Qu.:2020-04-28   Class :character   Class :character
##  Median :0.2000      Median :2022-05-27   Mode  :character   Mode  :character
##  Mean   :0.2531      Mean   :2021-08-31
##  3rd Qu.:0.3000      3rd Qu.:2022-07-27
##  Max.   :1.0000      Max.   :2022-11-19
##          funds
##  low        :342
##  medium-low :108
##  medium-high: 37
##  high       : 30
##
##
```

```
df10 <- df9
df10 <- df10 %>% select(-c( "company", "location", "date", "stage", "country"))

summary(df10)
```

```
##    industry         total_laid_off  percentage_laid_off         funds
##  Length:517         Min.   :  3.0   Min.   :0.0100      low        :342
##  Class :character   1st Qu.: 30.0   1st Qu.:0.1000      medium-low :108
##  Mode  :character   Median : 69.0   Median :0.2000      medium-high: 37
##                     Mean   :108.1   Mean   :0.2531      high       : 30
##                     3rd Qu.:130.0   3rd Qu.:0.3000
##                     Max.   :950.0   Max.   :1.0000
```

```
library(stats)
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.2.2
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(ggplot2)
library(tidyverse)
library(caret)
```

```
#funds column replaced with the numerical normalized version

df10a <- df10
df10a <- df10a %>% select(-c( "funds"))

df10a$funds <- df8$funds_raised

df10a$industry <- as.factor(df10a$industry)
summary(df10a)
```

```
##            industry   total_laid_off  percentage_laid_off     funds
##  Finance       :104   Min.   :  3.0   Min.   :0.0100      Min.   :0.0000
##  Healthcare    : 57   1st Qu.: 30.0   1st Qu.:0.1000      1st Qu.:0.0420
##  Retail        : 57   Median : 69.0   Median :0.2000      Median :0.1200
##  Marketing     : 56   Mean   :108.1   Mean   :0.2531      Mean   :0.2207
##  Real Estate   : 38   3rd Qu.:130.0   3rd Qu.:0.3000      3rd Qu.:0.3200
##  Transportation: 35   Max.   :950.0   Max.   :1.0000      Max.   :1.0000
##  (Other)       :170
```

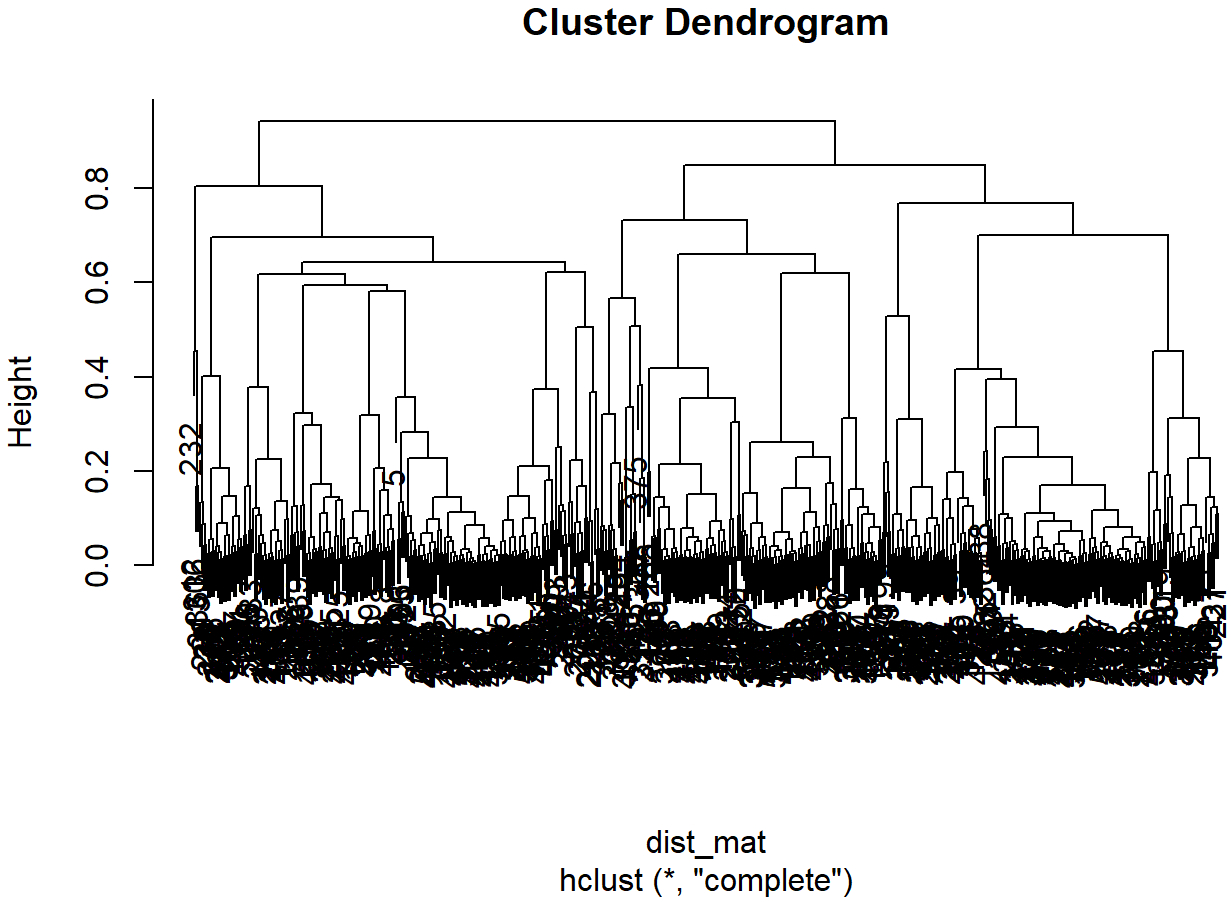
e Clustering

```
#pre-processing for clustering using HAC

set.seed(123)
preproc2 <- preProcess(df10a, method=c("center", "scale"))

df10a <- predict(preproc2, df10a)
```

```
#daisy function used due to it working with both both categorical and numerical data
#a dissimilarity matrix was created with the complete method

library(cluster)
```

```
## Warning: package 'cluster' was built under R version 4.2.2
```

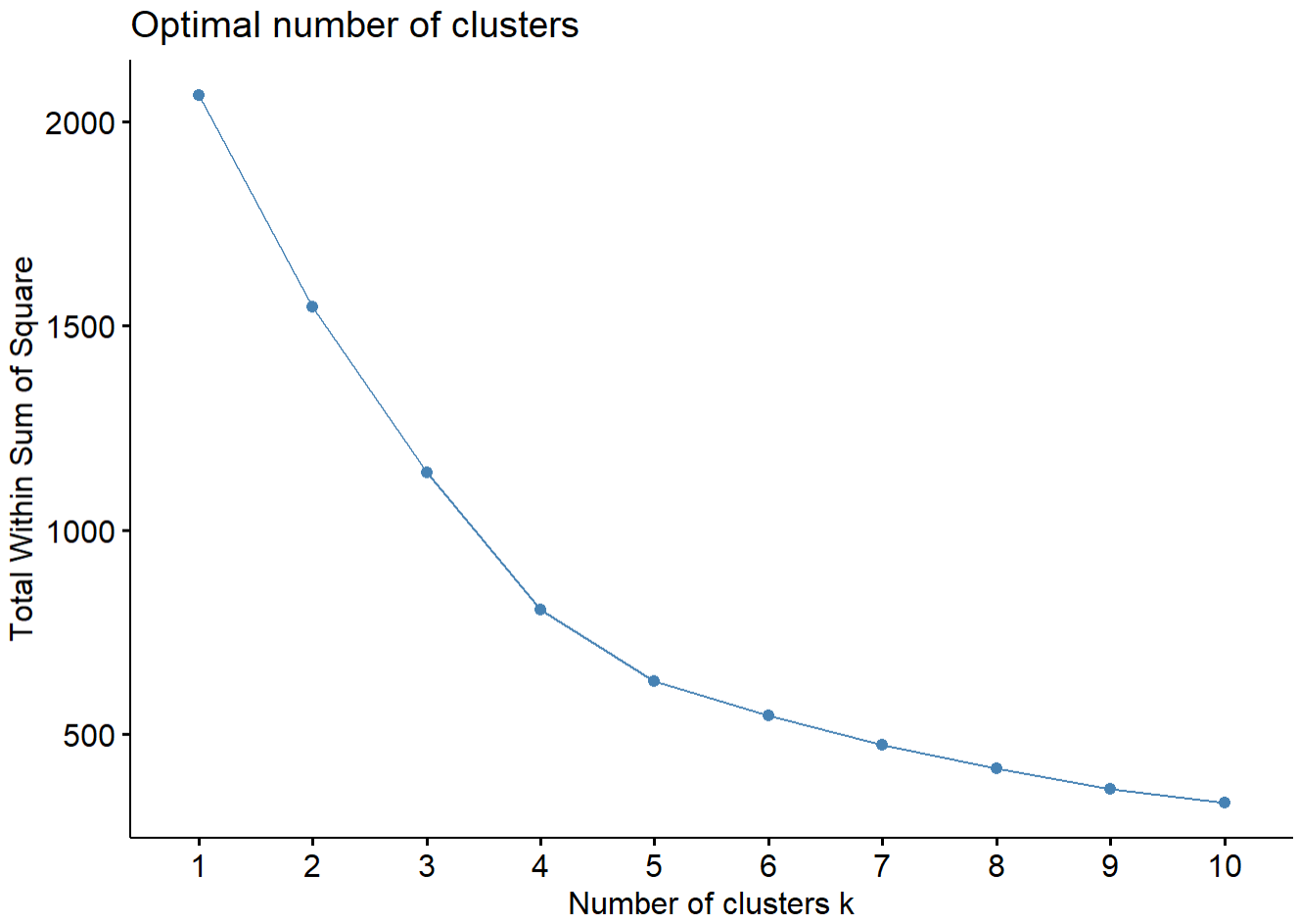
```
dist_mat <- daisy(df10a, metric = "gower")
hfit <- hclust(dist_mat, method = 'complete')
plot(hfit)
```




```
fviz_nbclust(df10a, FUN = hcut, method = "wss")
```

```
## Warning in stats::dist(x): NAs introduced by coercion
```

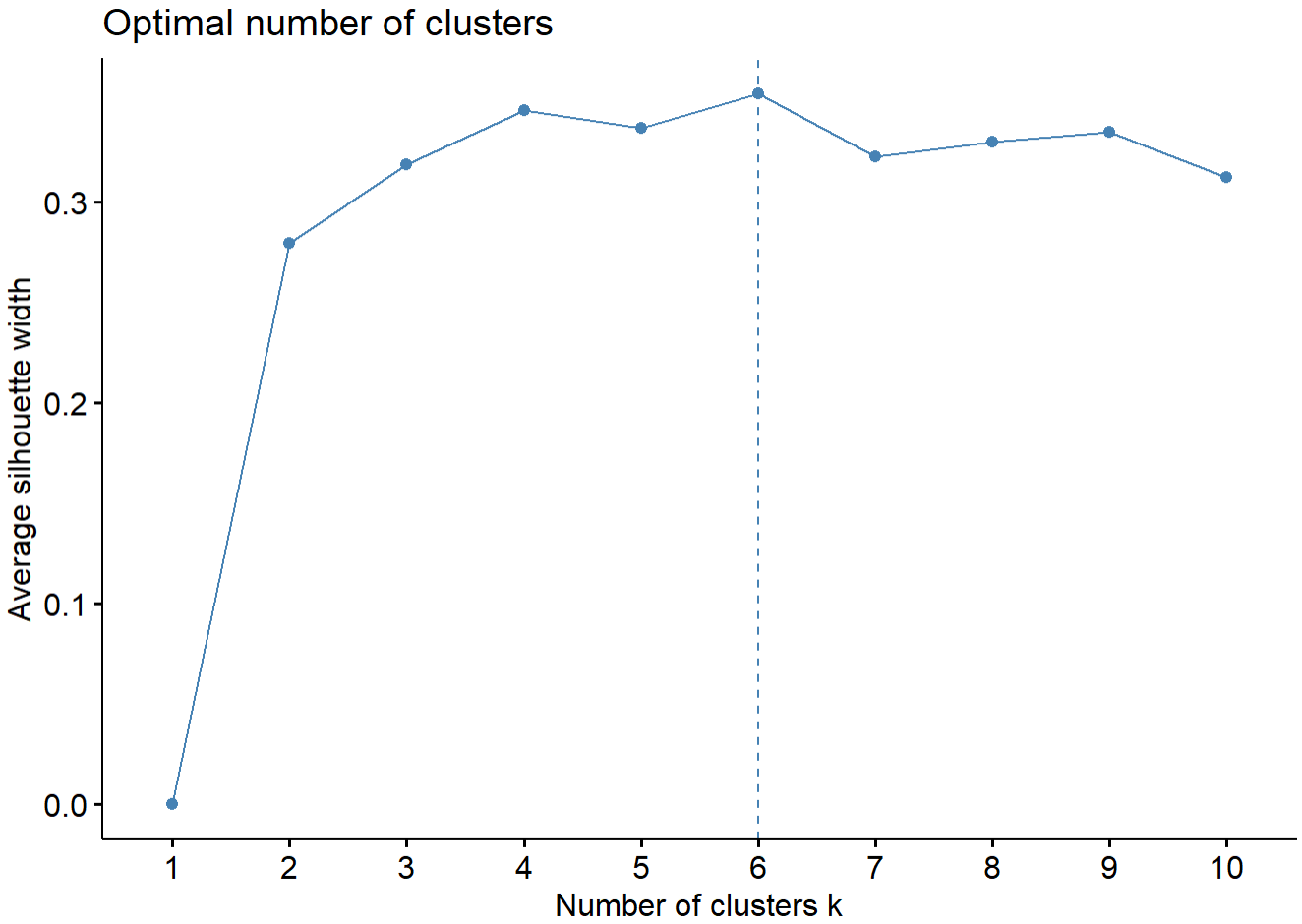
```
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion
```




```
fviz_nbclust(df10a, FUN = hcut, method = "silhouette")
```

```
## Warning in stats::dist(x): NAs introduced by coercion
```

```
## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion

## Warning in stats::dist(x, method = method, ...): NAs introduced by coercion
```



Based on both methods, k = 6 was chosen.

```
h6 <- cutree(hfit, k=6)
```

```
#dummy variables created to change categorical columns to numerical


dummy_industry <- dummyVars(industry ~ ., data = df10a)
dummies_industry <- as.data.frame(predict(dummy_industry, newdata = df10a))
```

```
## Warning in model.frame.default(Terms, newdata, na.action = na.action, xlev =
## object$lvls): variable 'industry' is not a factor
```

```
#creates pca

dfpca = prcomp(dummies_industry)
summary(dfpca)
```

```
## Importance of components:
##                           PC1    PC2    PC3
## Standard deviation     1.1809 1.0481 0.7121
## Proportion of Variance 0.4648 0.3662 0.1690
## Cumulative Proportion  0.4648 0.8310 1.0000
```

```
#pca data scatter plotted and labeled due to the clusters created by HAC
rotated_data = as.data.frame(dfpca$x)

rotated_data$Clusters = as.factor(h6)

ggplot(data = rotated_data, aes(x = PC1, y = PC2, col = Clusters)) + geom_point()
```

f Classification

```
#data will be classified using svm
library(caret)
library(e1071)
```

```
# target column returned
rotated_data$industry <-as.factor(df10a$industry)
```

```
#70-30 train test split
index = createDataPartition(y=rotated_data$industry, p=0.7, list=FALSE)
train = rotated_data[index,]
test = rotated_data[-index,]
```

```
#svm was applied to rotated_data to evaluate the prediction of the industry labels using 10-f
old  cross validation
train_control = trainControl(method = "cv", number = 10)
preproc = c("center", "scale")

svm <- train(industry ~., data = train, method = "svmLinear", trControl = train_control, preP
rocess = preproc)
svm
```

```
## Support Vector Machines with Linear Kernel
##
## 366 samples
##   4 predictor
##  12 classes: 'Consumer', 'Crypto', 'Data', 'Education', 'Finance', 'Food', 'Healthcare', '
Marketing', 'Real Estate', 'Retail', 'Security', 'Transportation'
##
## Pre-processing: centered (8), scaled (8)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 329, 327, 329, 330, 330, 330, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.4514571  0.3753848
##
## Tuning parameter 'C' was held constant at a value of 1
```

```
#knn was applied to rotated_data to evaluate the prediction of the industry labels using 10-f
old  cross validation


set.seed(123)
train_control2 <- trainControl(method="cv", number = 10)
preproc = c("center", "scale")
knnFit <- train(industry ~ ., data = train, method = "knn", trControl = train_control2, prePr
ocess = preproc, tuneLength = 20)


knnFit
```

```
## k-Nearest Neighbors
##
## 366 samples
##   4 predictor
##  12 classes: 'Consumer', 'Crypto', 'Data', 'Education', 'Finance', 'Food', 'Healthcare', '
Marketing', 'Real Estate', 'Retail', 'Security', 'Transportation'
##
## Pre-processing: centered (8), scaled (8)
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 328, 331, 332, 328, 330, 331, ...
## Resampling results across tuning parameters:
##
##   k   Accuracy   Kappa
##    5  0.4534309  0.3822675
##    7  0.4902723  0.4220975
##    9  0.4680895  0.3970639
##   11  0.4606201  0.3883167
##   13  0.4771909  0.4053026
##   15  0.4822867  0.4113612
##   17  0.4742332  0.4023748
##   19  0.4825281  0.4109586
##   21  0.4835570  0.4133277
##   23  0.4877737  0.4171546
##   25  0.4962000  0.4266209
##   27  0.4880297  0.4165809
##   29  0.4925007  0.4214408
##   31  0.4954379  0.4249364
##   33  0.4903209  0.4189895
##   35  0.4955729  0.4243690
##   37  0.4852595  0.4119248
##   39  0.4869997  0.4135832
##   41  0.4784958  0.4038903
##   43  0.4869663  0.4134563
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 25.
```

svm returned a accuracy value = 0.4751183 with a kappa of 0.3997081 knn returned a accuracy value = 0.4814461 with a kappa of 0.4077361

knn therefore slightly more accurate

gEvaluation

```
library(caret)

library(rpart)
library(tidyverse)

library(rattle)
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(ggplot2)
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
#confusionmatrix created...
pred <- predict(knnFit, test)
cm = confusionMatrix(test$industry, pred)
cm
```

```
## Confusion Matrix and Statistics
##
##                Reference
## Prediction       Consumer Crypto Data Education Finance Food Healthcare
##   Consumer              1      1    0         0       0    2          0
##   Crypto                1      0    0         0       0    0          0
##   Data                  0      0    0         0       0    0          4
##   Education             0      0    0         9       0    0          0
##   Finance               0      0    0         0      30    0          1
##   Food                  0      0    0         0       0    5          0
##   Healthcare            3      0    0         0       0    0          7
##   Marketing             2      0    0         0       0    2          0
##   Real Estate           0      0    0         0      10    0          1
##   Retail                0      0    0         0       0    2         10
##   Security              0      1    0         0       0    0          0
##   Transportation        3      0    0         0       0    0          0
##                Reference
## Prediction       Marketing Real Estate Retail Security Transportation
##   Consumer               4           0      0        0              1
##   Crypto                 5           0      0        0              1
##   Data                   0           0      2        1              0
##   Education              0           0      0        0              0
##   Finance                0           0      0        0              0
##   Food                   4           0      0        0              1
##   Healthcare             0           0      6        0              1
##   Marketing             10           0      0        1              1
##   Real Estate            0           0      0        0              0
##   Retail                 0           0      4        0              1
##   Security               2           0      0        1              3
##   Transportation         7           0      0        0              0
##
## Overall Statistics
##
##                Accuracy : 0.4437
##                  95% CI : (0.363, 0.5267)
##     No Information Rate : 0.2649
##     P-Value [Acc > NIR] : 1.668e-06
##
##                   Kappa : 0.3673
##
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: Consumer Class: Crypto Class: Data Class: Education
## Sensitivity                 0.100000       0.00000          NA           1.0000
## Specificity                 0.943262       0.95302     0.95364           1.0000
## Pos Pred Value              0.111111       0.00000          NA           1.0000
## Neg Pred Value              0.936620       0.98611          NA           1.0000
## Prevalence                  0.066225       0.01325     0.00000           0.0596
## Detection Rate              0.006623       0.00000     0.00000           0.0596
```

```
## Detection Prevalence        0.059603       0.04636     0.04636          0.0596
## Balanced Accuracy           0.521631       0.47651          NA          1.0000
##                      Class: Finance Class: Food Class: Healthcare
## Sensitivity                  0.7500     0.45455           0.30435
## Specificity                  0.9910     0.96429           0.92188
## Pos Pred Value               0.9677     0.50000           0.41176
## Neg Pred Value               0.9167     0.95745           0.88060
## Prevalence                   0.2649     0.07285           0.15232
## Detection Rate               0.1987     0.03311           0.04636
## Detection Prevalence         0.2053     0.06623           0.11258
## Balanced Accuracy            0.8705     0.70942           0.61311
##                      Class: Marketing Class: Real Estate Class: Retail
## Sensitivity                   0.31250                 NA       0.33333
## Specificity                   0.94958            0.92715       0.90647
## Pos Pred Value                0.62500                 NA       0.23529
## Neg Pred Value                0.83704                 NA       0.94030
## Prevalence                    0.21192            0.00000       0.07947
## Detection Rate                0.06623            0.00000       0.02649
## Detection Prevalence          0.10596            0.07285       0.11258
## Balanced Accuracy             0.63104                 NA       0.61990
##                      Class: Security Class: Transportation
## Sensitivity                 0.333333               0.00000
## Specificity                 0.959459               0.92958
## Pos Pred Value              0.142857               0.00000
## Neg Pred Value              0.986111               0.93617
## Prevalence                  0.019868               0.05960
## Detection Rate              0.006623               0.00000
## Detection Prevalence        0.046358               0.06623
## Balanced Accuracy           0.646396               0.46479
```

```
#find metrics of the confusion matrix...
metrics <- as.data.frame(cm$byClass)
metrics
```

	Sensitivity <dbl>	Specificity <dbl>	Pos Pred Value <dbl>	Neg Pred Value <dbl>
Class: Consumer	0.1000000	0.9432624	0.1111111	0.9366197
Class: Crypto	0.0000000	0.9530201	0.0000000	0.9861111
Class: Data	*NA*	0.9536424	*NA*	*NA*
Class: Education	1.0000000	1.0000000	1.0000000	1.0000000
Class: Finance	0.7500000	0.9909910	0.9677419	0.9166667
Class: Food	0.4545455	0.9642857	0.5000000	0.9574468
Class: Healthcare	0.3043478	0.9218750	0.4117647	0.8805970
Class: Marketing	0.3125000	0.9495798	0.6250000	0.8370370

	Sensitivity <dbl>	Specificity <dbl>	Pos Pred Value <dbl>	Neg Pred Value <dbl>
Class: Real Estate	*NA*	0.9271523	*NA*	*NA*
Class: Retail	0.3333333	0.9064748	0.2352941	0.9402985

1-10 of 12 rows | 1-5 of 12 columns Previous **1** 2 Next

```
#precision is found
metrics %>% select(c(Precision))
```

	Precision <dbl>
Class: Consumer	0.1111111
Class: Crypto	0.0000000
Class: Data	0.0000000
Class: Education	1.0000000
Class: Finance	0.9677419
Class: Food	0.5000000
Class: Healthcare	0.4117647
Class: Marketing	0.6250000
Class: Real Estate	0.0000000
Class: Retail	0.2352941

1-10 of 12 rows Previous **1** 2 Next

```
#recall is found
metrics %>% select(c(Recall))
```

	Recall <dbl>
Class: Consumer	0.1000000
Class: Crypto	0.0000000
Class: Data	*NA*
Class: Education	1.0000000
Class: Finance	0.7500000
Class: Food	0.4545455
Class: Healthcare	0.3043478

	Recall <dbl>
Class: Marketing	0.3125000
Class: Real Estate	*NA*
Class: Retail	0.3333333

1-10 of 12 rows Previous **1** 2 Next

```
#class probabilities for KNN
prob <- predict(knnFit, test, type = "prob")
```

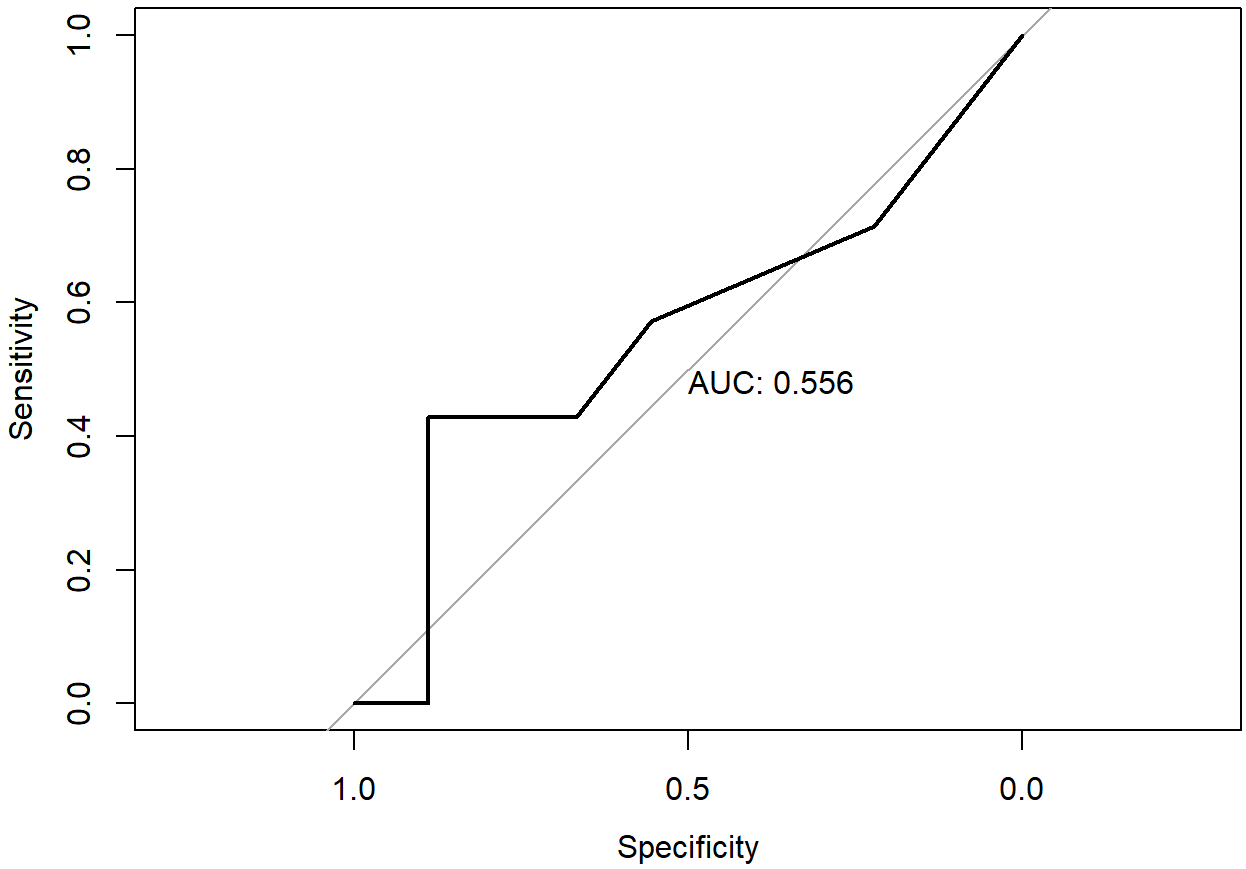
```
#roc object cfreated

roc_obj <- roc((test$industry), prob[,1])
```

```
## Warning in roc.default((test$industry), prob[, 1]): 'response' has more than two
## levels. Consider setting 'levels' explicitly or using 'multiclass.roc' instead
```

```
## Setting levels: control = Consumer, case = Crypto
```

```
## Setting direction: controls < cases
```

```
plot(roc_obj, print.auc=TRUE)
```



i Reflection

Coming from a completely different background than data science, I have learned a lot from the course. I never realized how much work data cleaning requires. I initially believed that a typical data analyst spent most time writing complex algorithms or looking at excel documents. I learned how useful various classifiers work, and how tuning various parameters can change the result by a lot, such as the type of distances used in HAC clustering. I always thought of analyzing data to be limited to mostly visualizations, but the course has shown me how machine learning can be used to help predict various models to facilitate in analyzing such data. Confusion matrices helped me understand better how to evaluate the data I am working with. Furthermore, I am excited to learn more about machine learning in future classes.