HW1-Fangzhou Song

Fangzhou Song

```r
rm(list=ls())
library(tidyverse)
library(GGally)
```

Question 1

Install the titanic package from CRAN and load the titanic_train dataset, and check its help file to learn what the dataset contains.

```r
library(titanic)
data("titanic_train")
glimpse(titanic_train)
```

```
## Observations: 891
## Variables: 12
## $ PassengerId <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,...
## $ Survived    <int> 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0,...
## $ Pclass      <int> 3, 1, 3, 1, 3, 3, 1, 3, 3, 2, 3, 1, 3, 3, 3, 2, 3,...
## $ Name        <chr> "Braund, Mr. Owen Harris", "Cumings, Mrs. John Bra...
## $ Sex         <chr> "male", "female", "female", "female", "male", "mal...
## $ Age         <dbl> 22, 38, 26, 35, 35, NA, 54, 2, 27, 14, 4, 58, 20, ...
## $ SibSp       <int> 1, 1, 0, 1, 0, 0, 0, 3, 0, 1, 1, 0, 0, 1, 0, 0, 4,...
## $ Parch       <int> 0, 0, 0, 0, 0, 0, 0, 1, 2, 0, 1, 0, 0, 5, 0, 0, 1,...
## $ Ticket      <chr> "A/5 21171", "PC 17599", "STON/O2. 3101282", "1138...
## $ Fare        <dbl> 7.2500, 71.2833, 7.9250, 53.1000, 8.0500, 8.4583, ...
## $ Cabin       <chr> "", "C85", "", "C123", "", "", "E46", "", "", "", ...
## $ Embarked    <chr> "S", "C", "S", "S", "S", "Q", "S", "S", "S", "C", ...
```

Remove the PassengerId, Name, Ticket and Cabin columns, and transform the Fare variable by taking its log.

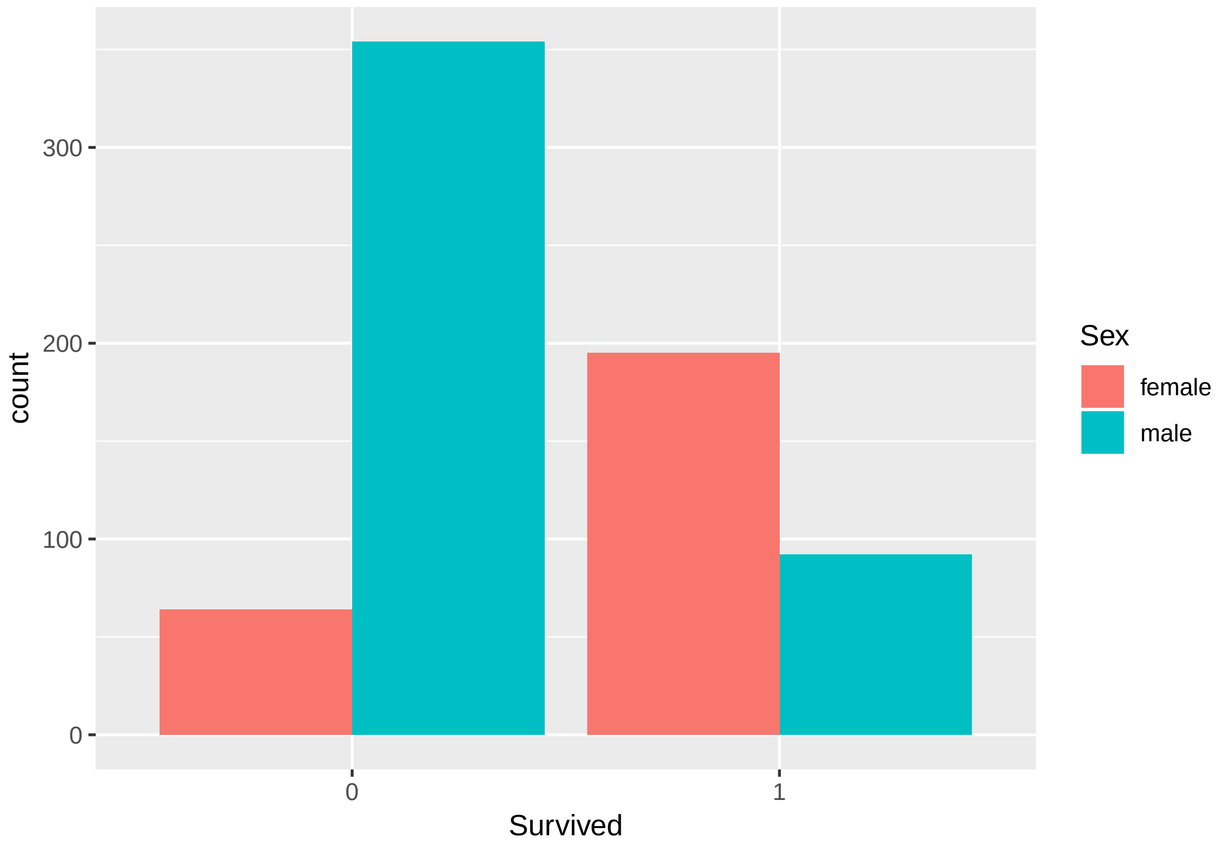
```r
data1= titanic_train %>%
  filter(Embarked!="",Fare!=0,is.na(Age)==FALSE) %>%
  select(Survived,Pclass,Sex:Parch,Fare,Embarked) %>%
  mutate(
    Fare=log(Fare),
    Survived=as.factor(Survived),
    Pclass=as.factor(Pclass),
    Sex=as.factor(Sex)
  )
head(data1,10)
```

```
##    Survived Pclass    Sex Age SibSp Parch     Fare Embarked
## 1         0      3   male  22     1     0 1.981001        S
## 2         1      1 female  38     1     0 4.266662        C
## 3         1      3 female  26     0     0 2.070022        S
## 4         1      1 female  35     1     0 3.972177        S
## 5         0      3   male  35     0     0 2.085672        S
## 6         0      1   male  54     0     0 3.948596        S
## 7          0       3   male   2     3     1 3.048088         S
## 8          1       3 female  27     0     2 2.409941         S
## 9          1       2 female  14     1     0 3.403555         C
## 10         1       3 female   4     1     1 2.815409         S
```

Choose three variable pairs (for instance, "Sex" and "Survived" is one pair) and plot their distribution in the dataset by using appropriate plots. You can try mosaic plots, densities, histograms. You are asked to provide at least one plot for each pair.

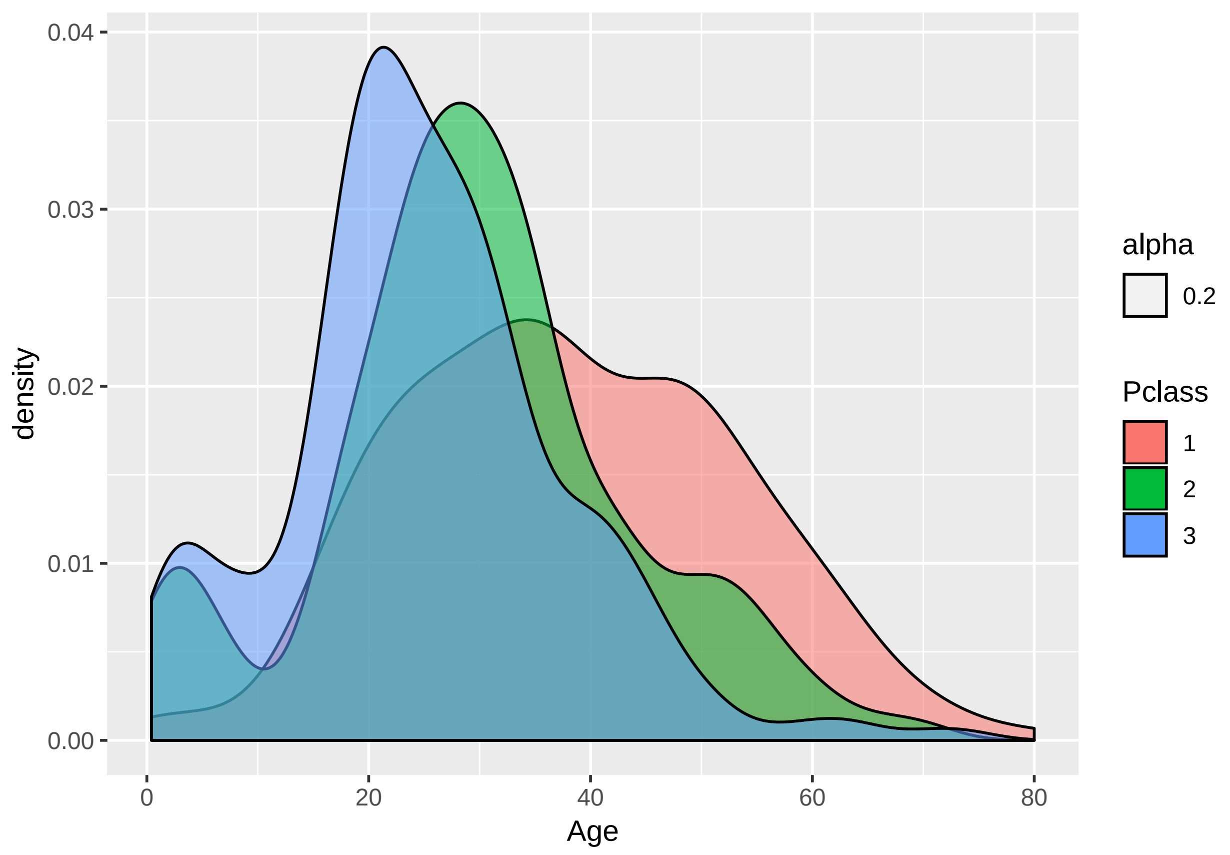
Sex v.s. Survived

```
data1 %>%
  ggplot() +
  geom_bar(mapping = aes(x=Survived,fill=Sex),position = "dodge")
```



Pclass v.s. Age

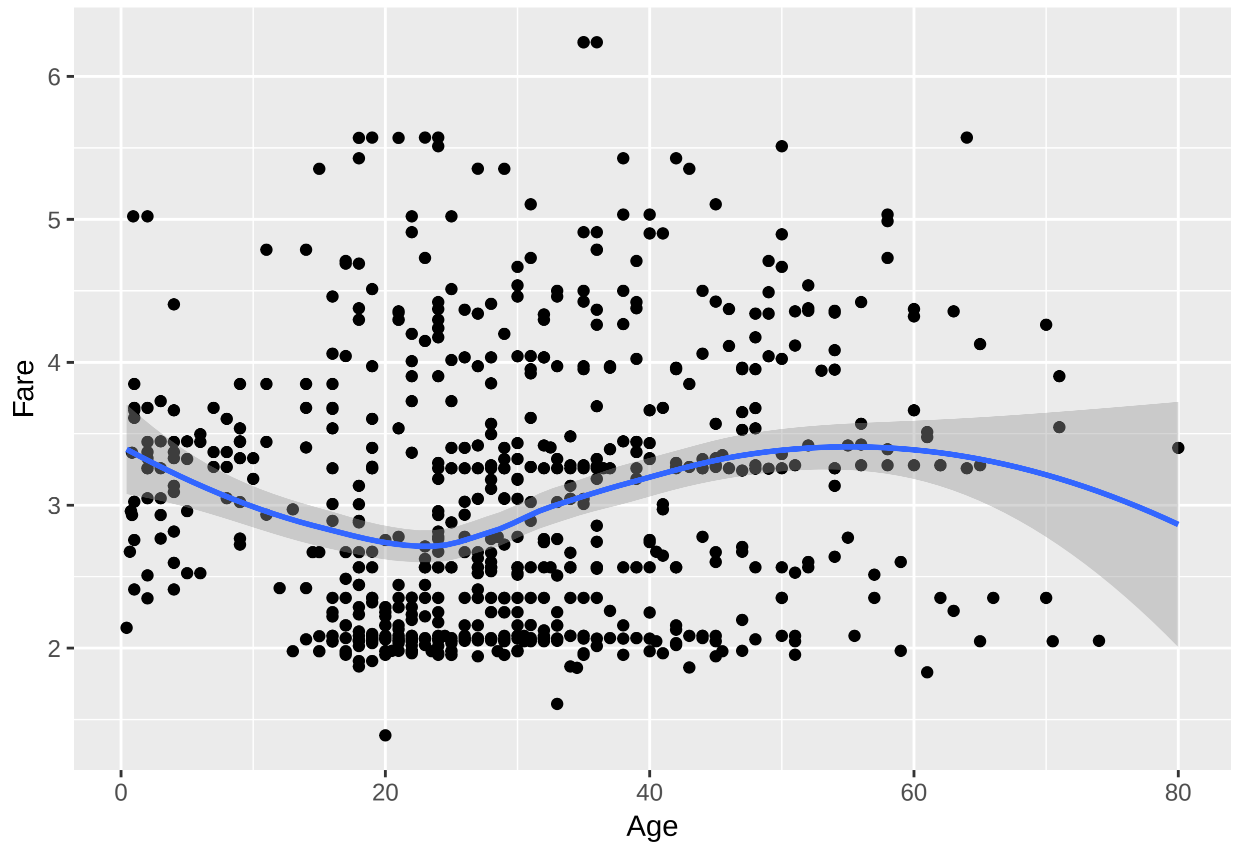
```
data1 %>%
  ggplot()+
  geom_density(mapping = aes(x=Age,fill=Pclass,alpha=0.2))
```



Age v.s. Fare

```
data1 %>%
  ggplot(mapping = aes(x=Age,y=Fare))+
  geom_point()+
  geom_smooth()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



Create a model matrix of the dataset that only contains numbers (no factors!) by using the model.matrix function. Then, remove the Survived variable from this dataset.

Matrix that only contains numbers

```
data1_a=model.matrix(~Survived+Pclass+Age+SibSp+Parch+Fare+Embarked,data1)
data1_a=data1_a[,-1]
head(data1_a,5)
```

```
##   Survived1 Pclass2 Pclass3 Age SibSp Parch     Fare EmbarkedQ EmbarkedS
## 1         0       0       1  22     1     0 1.981001         0         1
## 2         1       0       0  38     1     0 4.266662         0         0
## 3         1       0       1  26     0     0 2.070022         0         1
## 4         1       0       0  35     1     0 3.972177         0         1
## 5         0       0       1  35     0     0 2.085672         0         1
```

Remove the Survived variable from this dataset

```
data1_b=data1_a[,-1]
head(data1_b,5)
```

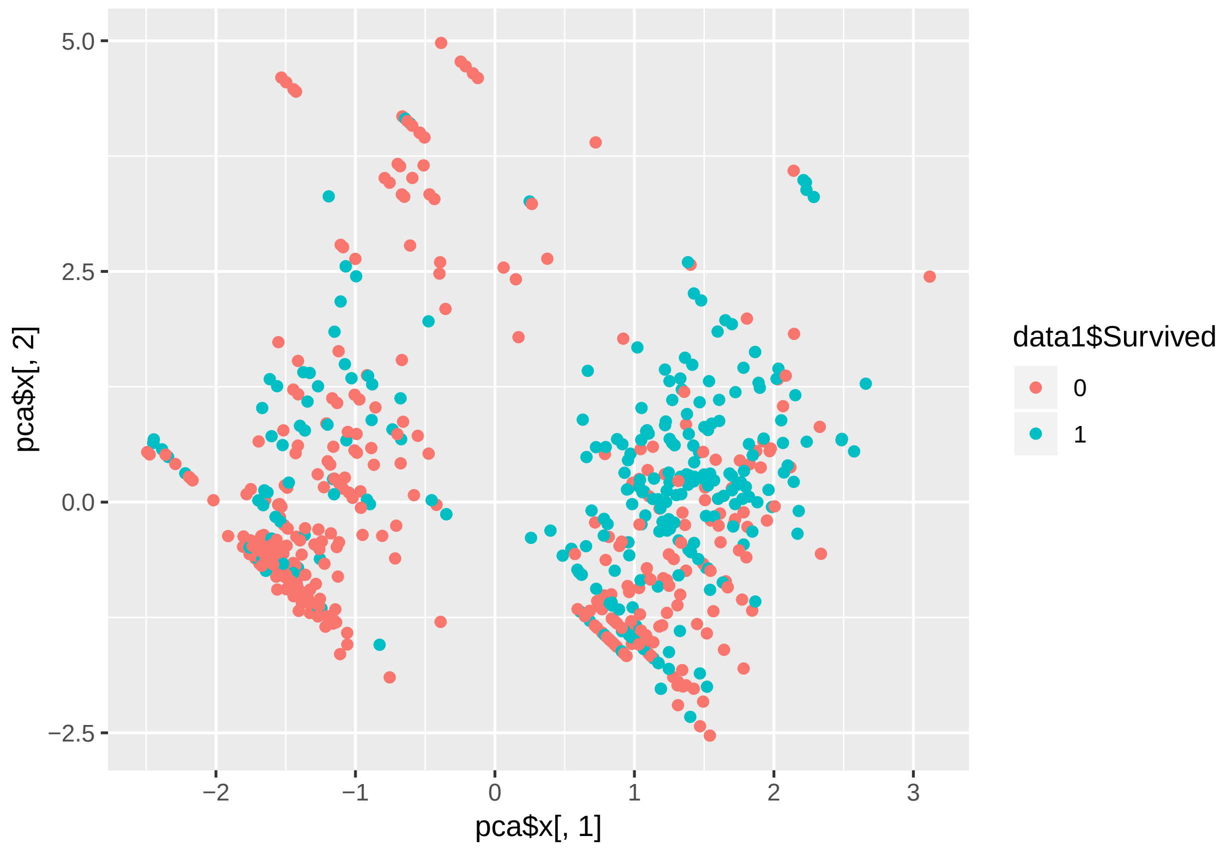
```
##   Pclass2 Pclass3 Age SibSp Parch     Fare EmbarkedQ EmbarkedS
## 1       0       1  22     1     0 1.981001         0         1
## 2       0       0  38     1     0 4.266662         0         0
## 3       0       1  26     0     0 2.070022         0         1
## 4       0       0  35     1     0 3.972177         0         1
## 5       0       1  35     0     0 2.085672         0         1
```

Fit a PCA to your matrix from the previous step. Plot the scores of the observations (use only the first 2 dimensions) and color them according to the Survived variable.

```r
pca=prcomp(data1_b,center = TRUE,scale. = TRUE)
pca$rotation
```

```
##                   PC1        PC2         PC3         PC4         PC5
## Pclass2    0.36714425 -0.2393812 -0.28344550  0.66825184 -0.11222375
## Pclass3   -0.67037012  0.1416326 -0.04110184 -0.13741885 -0.10814542
## Age        0.25507885 -0.3685565  0.27098802 -0.43446975 -0.60282680
## SibSp      0.05726306  0.5861042 -0.17492109  0.09317291 -0.04938814
## Parch      0.13194660  0.5353955 -0.17551390 -0.04906136 -0.48701851
## Fare       0.54556639  0.3347634  0.14999258 -0.28109825  0.16086501
## EmbarkedQ -0.16515616  0.1320152  0.54523246  0.49037236 -0.45321671
## EmbarkedS -0.07166301 -0.1647483 -0.68070262 -0.12337779 -0.37603338
##                   PC6         PC7           PC8
## Pclass2    0.06940775  0.28178992  0.4291651398
## Pclass3    0.03890987  0.19340600  0.6777577183
## Age       -0.19886195  0.35916690  0.0703943506
## SibSp     -0.67181835  0.37692696 -0.1344293745
## Parch      0.63076006  0.08823134 -0.1410657186
## Fare      -0.07596361 -0.37895124  0.5599327673
## EmbarkedQ -0.17801302 -0.42480285  0.0006722524
## EmbarkedS -0.25974074 -0.52940296  0.0032317068
```

```r
ggplot(data=data.frame(pca$x))+
  geom_point(mapping = aes(x=pca$x[,1],y=pca$x[,2],color=data1$Survived))
```



Repeat the previous question with NMF (non-negative matrix factorization). Use rank=2 for the fit. Note which variables were chosen.

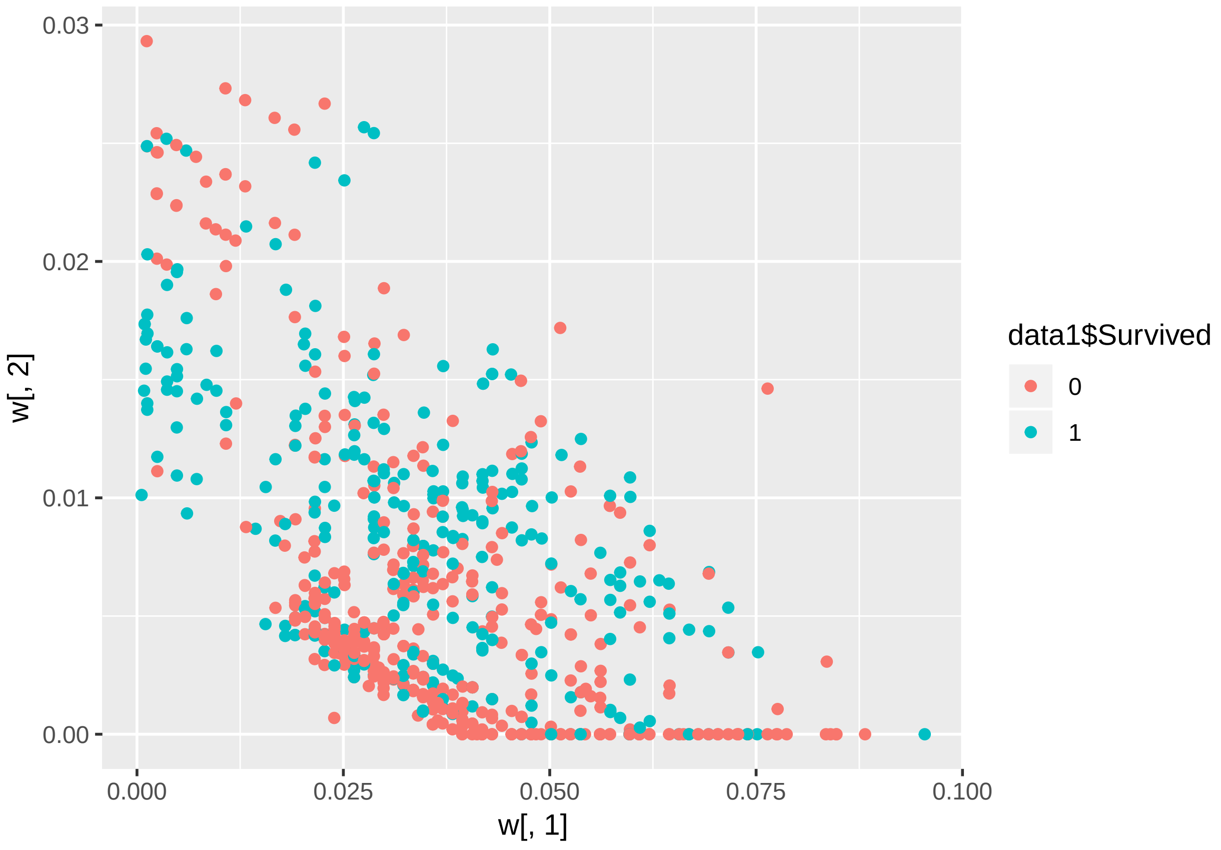
```
library(pkgmaker)
```

```
## Loading required package: registry

##
## Attaching package: 'pkgmaker'

## The following object is masked from 'package:base':
##
##     isFALSE
```

```
library(registry)
library(rngtools)
library(cluster)
library(NMF)
```

```
## NMF - BioConductor layer [NO: missing Biobase] | Shared memory capabilities [NO: windows] | Cores 7/

##   To enable the Bioconductor layer, try: install.extras('
## NMF
## ') [with Bioconductor repository enabled]
```

```
res_nmf=nmf(data1_b,rank = 2,method = "snmf/r")
w=basis(res_nmf)
h=coef(res_nmf)
ggplot(data=data.frame(w))+
  geom_point(mapping = aes(x=w[,1],y=w[,2],color=data1$Survived))
```



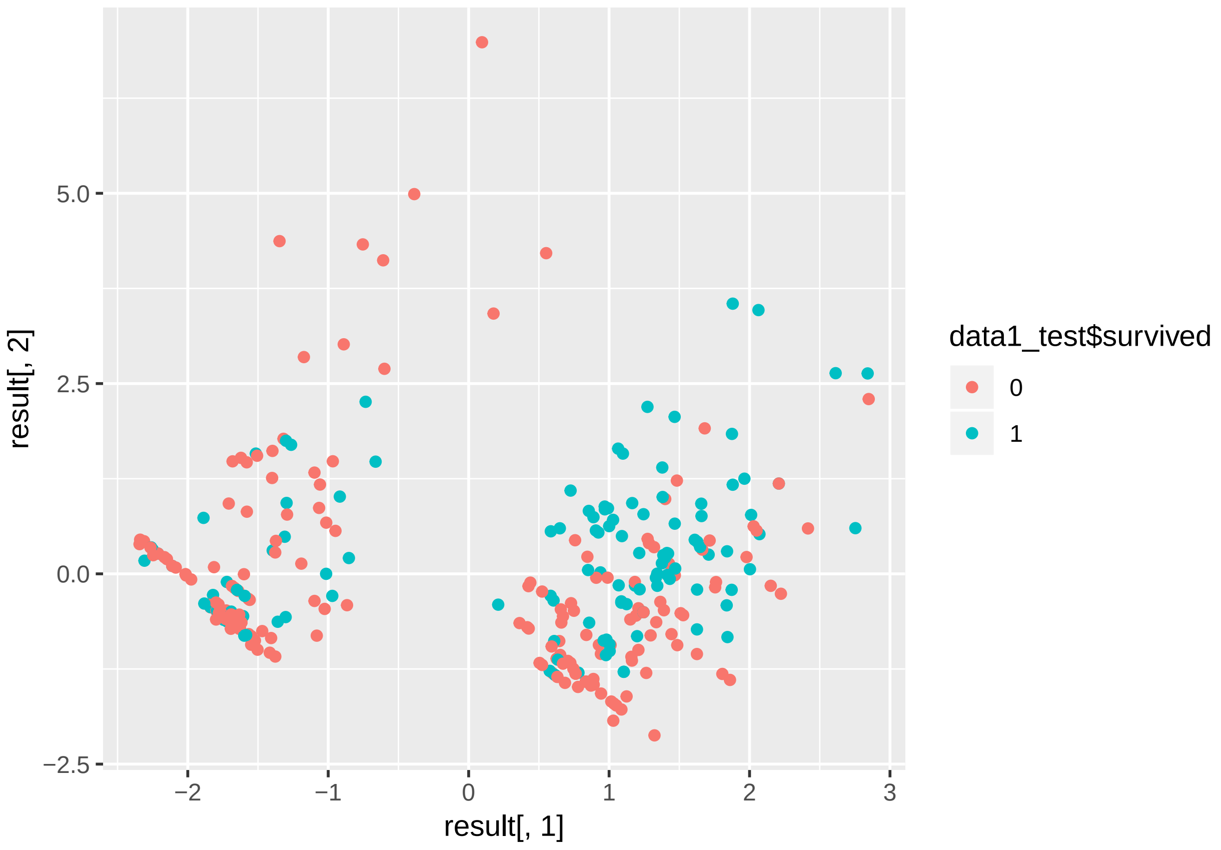
Finally, load the titanic_test dataset. Using your fitted PCA from the previous stages, obtain the 2 dimensional projections of the test dataset. Plot the scores with respect to the Survived variable.

```
data("titanic_test")
sur=read_csv("titanic3.csv")
sur=rename(sur,Name=name)
```

```
data1_test= titanic_test %>%
  inner_join(sur,by="Name") %>%
  filter(Embarked!="",Fare!=0,is.na(Age)==FALSE) %>%
  select(survived,Pclass,Sex:Parch,Fare,Embarked) %>%
  mutate(
    survived=as.factor(survived),
    Fare=log(Fare),
    Pclass=as.factor(Pclass),
    Sex=as.factor(Sex)
  )
data1_test_m=model.matrix(~Pclass+Age+SibSp+Parch+Fare+Embarked,data1_test)
data1_test_m=data1_test_m[,-1]
head(data1_test_m,5)
```

```
##   Pclass2 Pclass3  Age SibSp Parch     Fare EmbarkedQ EmbarkedS
## 1       0       1 34.5     0     0 2.057860         1         0
## 2       0       1 34.5     0     0 2.057860         1         0
## 3       0       1 47.0     1     0 1.945910         0         1
## 4       1       0 62.0     0     0 2.270836         1         0
## 5       0       1 27.0     0     0 2.159003         0         1
```

```
result=scale(data1_test_m) %*% pca$rotation[,1:2]
ggplot(data=data.frame(result))+
  geom_point(mapping = aes(x=result[,1],y=result[,2],color=data1_test$survived))
```



Question 2

Choose a dataset from Kaggle, any of the R packages or generate your own dataset by capturing photos from Amazon.com with the "Image Downloader" extension of Google Chrome as we did in class. If the dataset size is too large, randomly choose 2000 samples. Feel free to discard any variables that are not numbers or factors.

```
library(jpeg)
library(EBImage)
```

```
##
## Attaching package: 'EBImage'

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

Load cat pictures, use gray images, change their size into 400x300 and matrix form

```
cat_name=paste0("cats/",list.files("cats/"))

convert_image=function(x){
  a=readImage(x)
  a=channel(a,"gray")
  c=resize(a,w=400,h=300)
  return(c)
}
```

```
cat_pic_mat=sapply(cat_name,function(x) matrix(convert_image(x),nrow=1))
```

Display a example of pictures

```
cat_pic_array=array(sapply(cat_name[1:3],convert_image),dim = c(400,300,3))

display(cat_pic_array[,,3])
```

> Use PCA and at least one other method (this could be Logistic PCA, Sparse PCA, NMF or any other factorization method) to obtain a low dimensional representation of the data.Provide plots of the low dimensional representations.

```
dim(cat_pic_mat)
```

```
## [1] 120000   1000
```

Apply PCA algorithm

```
cat_pic_mat_t=t(cat_pic_mat)
cat_pca_t=prcomp(cat_pic_mat_t,center = TRUE)
```

Apply NMF algorithm

```
#cat_nmf=nmf(cat_pic_mat_t,rank = 10,method = "snmf/r")
#It takes about 2 hour to run
```

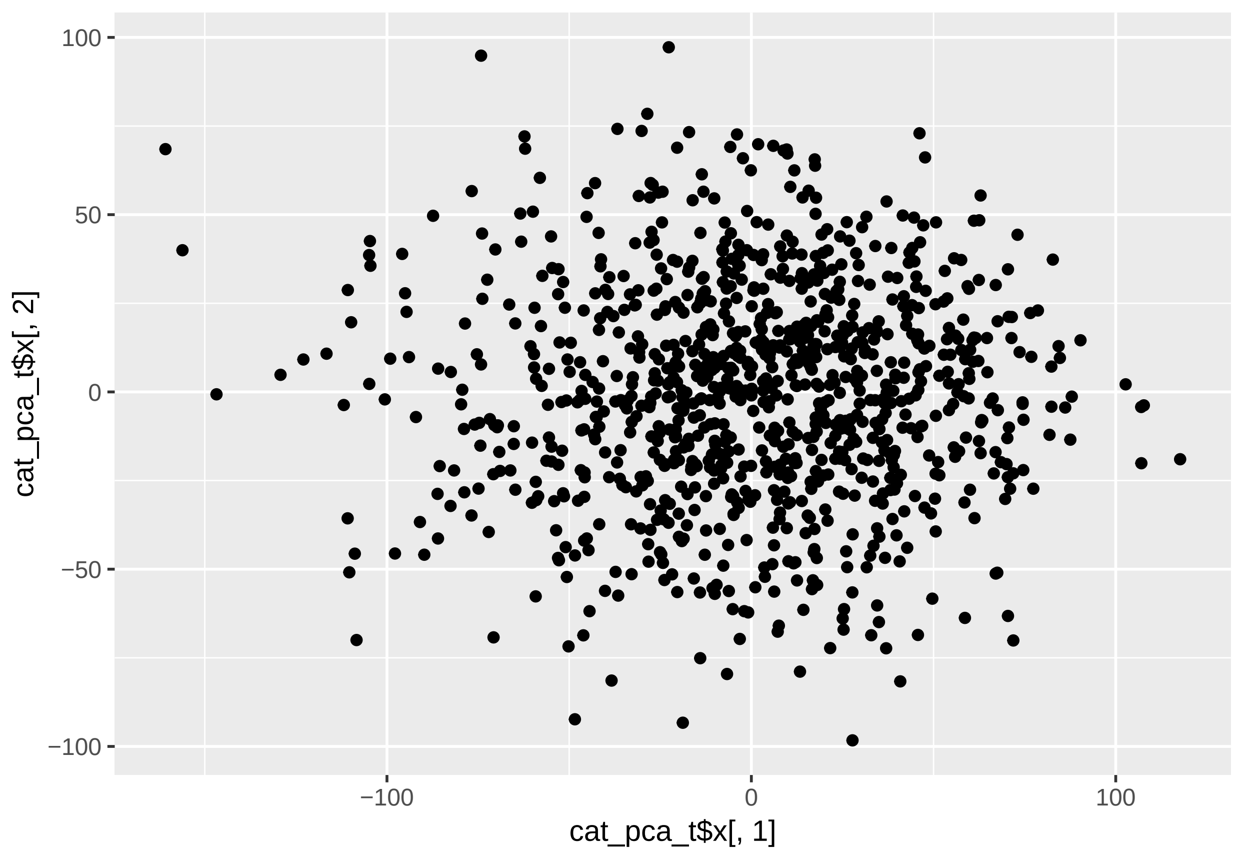
Low dimensional representations

```r
theMin=min(apply(cat_pca_t$rotation[,1:100],2,min))
theMax=max(apply(cat_pca_t$rotation[,1:100],2,max))

rots=(cat_pca_t$rotation[,1:100]-theMin)/(theMax-theMin)

display(rots[,2] %>% matrix(nrow=400,ncol=300) )
```

> Analyze the fitted models in detail. Which variables are chosen or have a larger magnitude? Do the 2-dimensional plots suggest anything specific about the dataset? Write a summary of your results; your summary should have at least 150 words.

2-dimensional plots

```r
ggplot(data=data.frame(cat_pca_t$x))+
  geom_point(mapping = aes(x=cat_pca_t$x[,1],y=cat_pca_t$x[,2]))
```

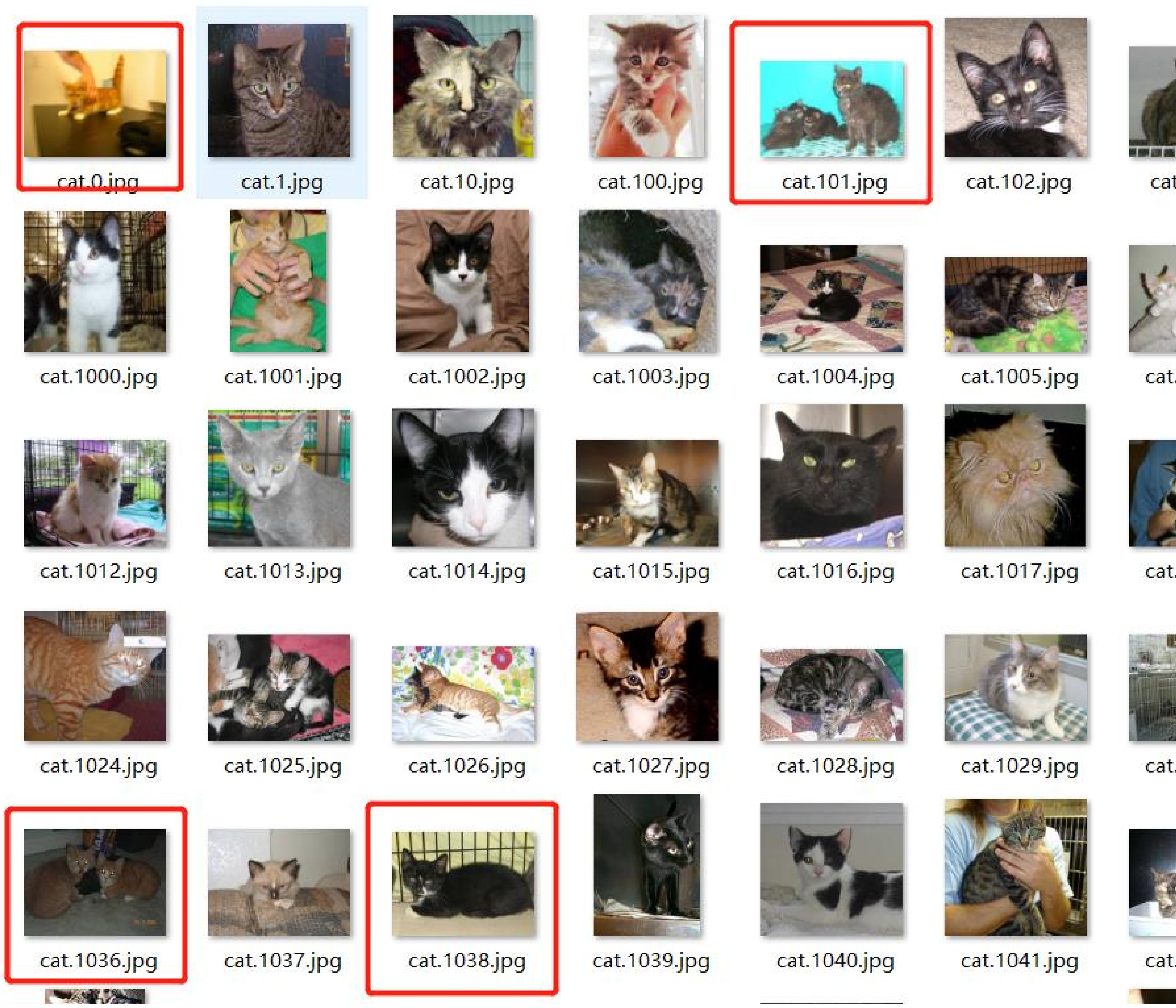


Summary

In question 2, I choose to apply PCA algorithm on cat pictures instead of some row data like Titanic in question 1. Obviously, PCA have quite different performance on these two types of data. In cat pictures, the variables are not ones that have some specific meaning like age, price, etc, but ones that represent the level of gray color of each pixel using float number. Therefore, it hard to show and explain Which variables are chosen or have a larger magnitude. Instead, we can plot loadings or principal components to regrad it as the main structure or appearance of cats. And every picture can be represented as the combination of those characteristcs.

From the 2 dimesional plot, we can see that most of points are gathering around (0,0) and some of them scatter randomly all around. I think this can be explained by the poor quality of pictures.

```r
knitr::include_graphics("pic.jpg")
```



From the screen shot , we can see that "good" pictures are ones which mainly focus on the face of cats, "bad" pictures tend to have other factors in pictures which have less common features compared to those "good" ones. So, next time I will definitely collect some better quality pictures