Data Analysis and Visualization - Assignment 3 & 4

Ma Jingchun, 2020111235

1. 利用你的学号，生成一个 1000×p 的矩阵 X，如下所示。

```
library(glmnet)
library(MASS)
library(factoextra)
library(tidyverse)
library(cluster)
library(broom)
library(gclus)
```

```
######### Please write your R code in this chunk #########
### Solution to Q1
studno <- 2020111235 # 改成你的学号！！！！！
set.seed(studno)
n <- 1000
p <- 10
beta0 <- 1
beta <- c(c(1,2,3,4,5), rep(0, p-5))
X <- matrix(rnorm(n*p, 0, 1), nrow=n, ncol=p)
e <- rnorm(n, 0, 0.2)
Y <- beta0 + X %*% beta + e
dat <- data.frame(Y,X)
colnames(dat) <- c("Y", paste("X", 1:p, sep=""))
```

- 请描述目前生成的响应变量中，有用的自变量是哪些

```
######### Please write your R code in this chunk #########
### Solution to Q1.1
model1 = lm(Y~., data=dat)
summary(model1)
```

```
##
## Call:
## lm(formula = Y ~ ., data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68942 -0.13741 -0.00005  0.12912  0.66300
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.007e+00  6.354e-03 158.448   <2e-16 ***
## X1           9.954e-01  6.552e-03 151.909   <2e-16 ***
## X2           1.994e+00  6.233e-03 319.895   <2e-16 ***
## X3           3.002e+00  6.706e-03 447.655   <2e-16 ***
## X4           3.997e+00  6.133e-03 651.807   <2e-16 ***
## X5           4.993e+00  6.421e-03 777.670   <2e-16 ***
## X6          -5.415e-05  6.244e-03  -0.009    0.993
## X7          -4.716e-03  6.529e-03  -0.722    0.470
## X8           4.675e-03  6.596e-03   0.709    0.479
## X9           7.207e-03  6.155e-03   1.171    0.242
## X10         -5.122e-03  6.349e-03  -0.807    0.420
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1999 on 989 degrees of freedom
## Multiple R-squared:  0.9993, Adjusted R-squared:  0.9993
## F-statistic: 1.495e+05 on 10 and 989 DF,  p-value: < 2.2e-16
```

有用的自变量为X1、X2、X3、X4、X5

- 请用 AIC 估计 Y ~ X 的线性回归中，依次估计出来的系数非零的变量分别是哪些。

```
######### Please write your R code in this chunk #########
### Solution to Q1.2
model.for <- step(model1,direction = 'forward')
```

```
## Start:  AIC=-3209.41
## Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10
```

```
summary(model.for)
```

```
##
## Call:
## lm(formula = Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 +
##     X10, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68942 -0.13741 -0.00005  0.12912  0.66300
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.007e+00  6.354e-03 158.448   <2e-16 ***
## X1           9.954e-01  6.552e-03 151.909   <2e-16 ***
## X2           1.994e+00  6.233e-03 319.895   <2e-16 ***
## X3           3.002e+00  6.706e-03 447.655   <2e-16 ***
## X4           3.997e+00  6.133e-03 651.807   <2e-16 ***
## X5           4.993e+00  6.421e-03 777.670   <2e-16 ***
## X6          -5.415e-05  6.244e-03  -0.009    0.993
## X7          -4.716e-03  6.529e-03  -0.722    0.470
## X8           4.675e-03  6.596e-03   0.709    0.479
## X9           7.207e-03  6.155e-03   1.171    0.242
## X10         -5.122e-03  6.349e-03  -0.807    0.420
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1999 on 989 degrees of freedom
## Multiple R-squared:  0.9993, Adjusted R-squared:  0.9993
## F-statistic: 1.495e+05 on 10 and 989 DF,  p-value: < 2.2e-16
```

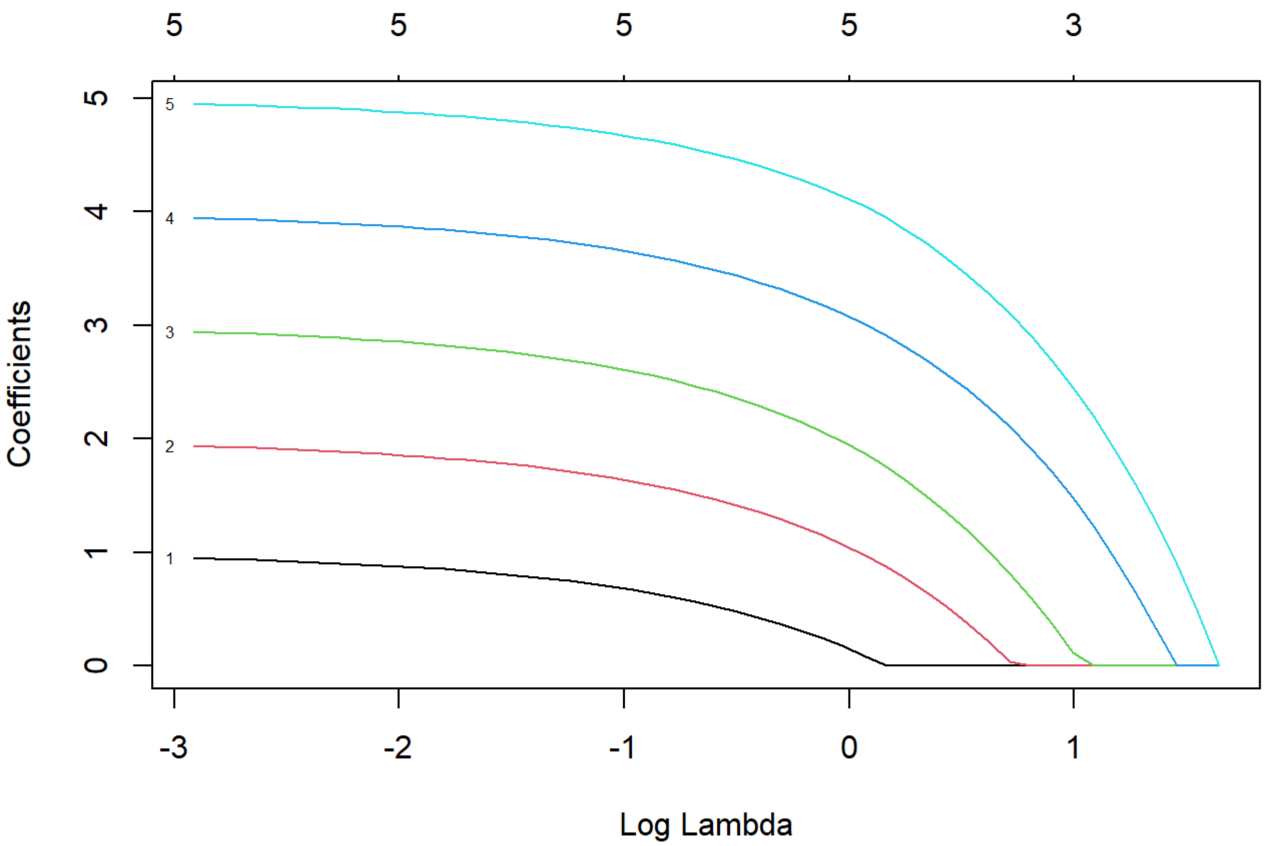
依次引入的变量为X1，X2，X3，X4，X5

- 请用 lasso 和 ridge，依次估计出来的系数非零的变量分别是哪些，绘制 solution path

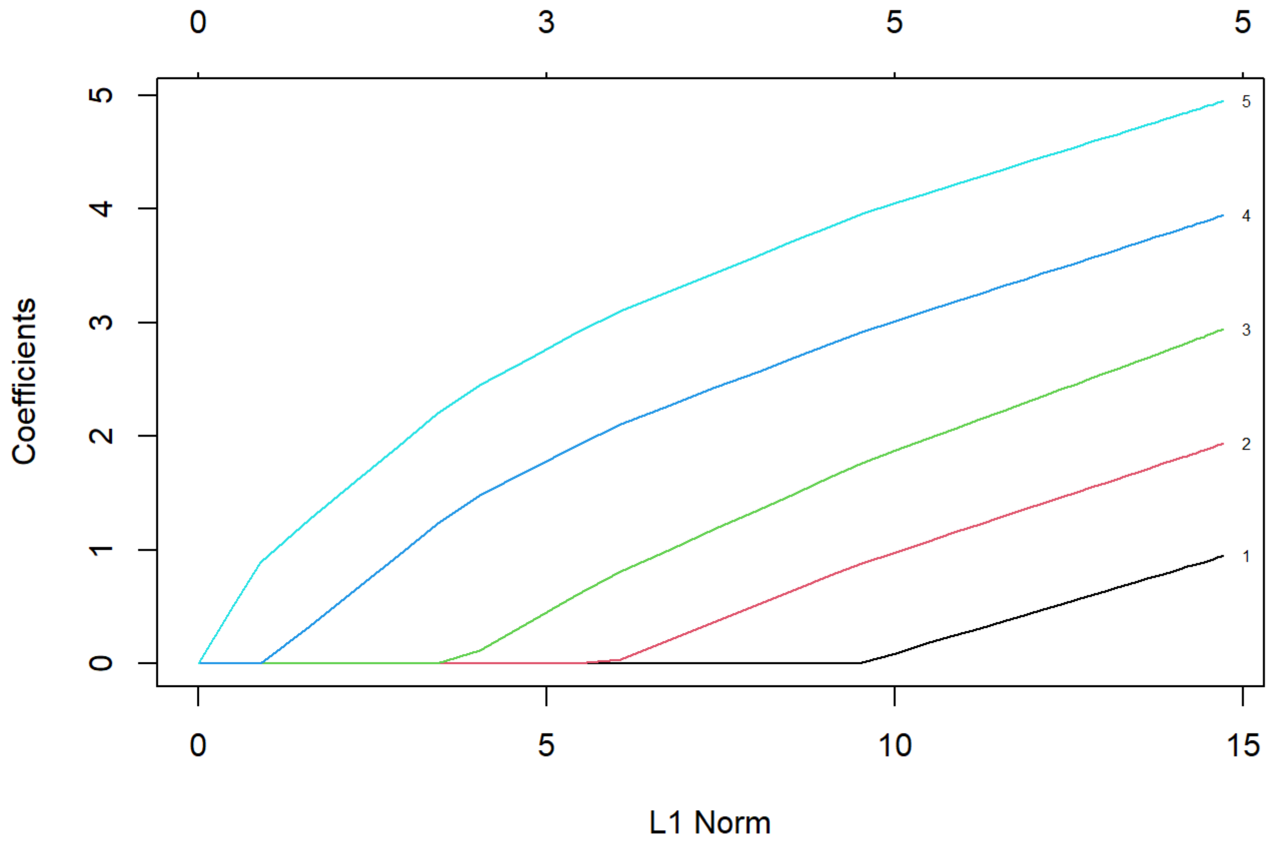
```
######### Please write your R code in this chunk #########
### Solution to Q1.3
X=model.matrix(Y~.,dat)[,-1]
train=sample(1:n, n/2)
test=(-train)
Y.test=Y[test]
#lasso
cv.lasso <- cv.glmnet(X[train,], Y[train], alpha=1/2)
M.lasso <- glmnet(X[train,],Y[train],alpha=1,
                  lambda=cv.lasso$lambda.min)
coef(M.lasso) # variable selected given lambda
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                    s0
## (Intercept) 1.0073598
## X1          0.9228755
## X2          1.9086649
## X3          2.9116840
## X4          3.9185217
## X5          4.9251443
## X6          .
## X7          .
## X8          .
## X9          .
## X10         .
```

```
M.lasso <- glmnet(X[train,],Y[train],alpha=1)
plot(M.lasso, label=T, xvar="lambda")
```



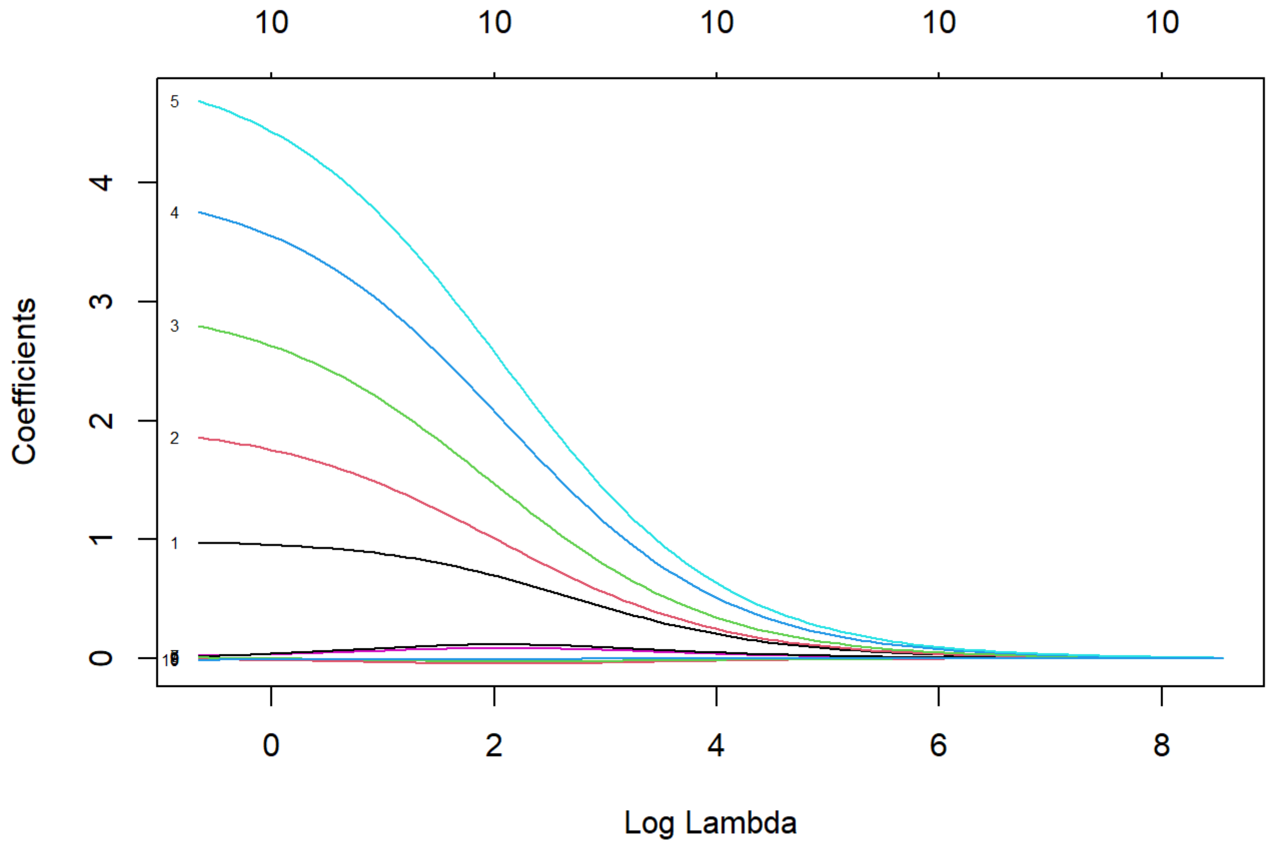

```
plot(M.lasso, label = T, xvar="norm")
```



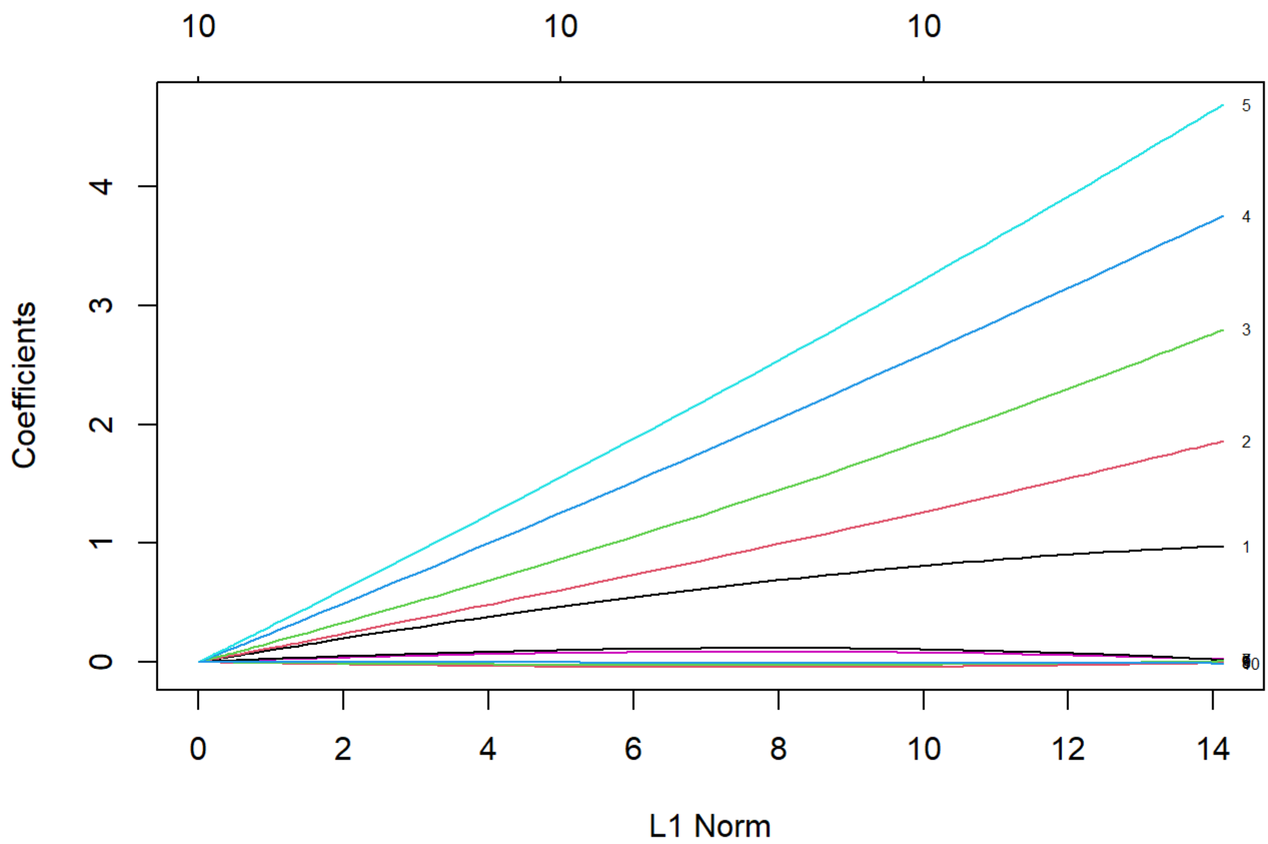

```
#ridge
cv.out=cv.glmnet(X[train,], Y[train],alpha=0)
ridge.mod=glmnet(X[train,], Y[train],alpha=0,
                 lambda = cv.out$lambda.min)
coef(ridge.mod)
```

```
## 11 x 1 sparse Matrix of class "dgCMatrix"
##                       s0
## (Intercept)  1.010589105
## X1           0.973804548
## X2           1.858640461
## X3           2.794646856
## X4           3.752052974
## X5           4.686440780
## X6           0.022684863
## X7           0.018284978
## X8          -0.006674086
## X9           0.006614583
## X10         -0.011930856
```

```
M.ridge <- glmnet(X[train,],Y[train],alpha=0)
plot(M.ridge, label=T, xvar="lambda")
```




```
plot(M.ridge, label = T, xvar="norm")
```



- 设定 p 为 100 重复（b）-（c），结果有什么变化?

```
######### Please write your R code in this chunk #########
### Solution to Q1.4
#(a)
p <- 100
beta <- c(c(1,2,3,4,5), rep(0, p-5))
X <- matrix(rnorm(n*p, 0, 1), nrow=n, ncol=p)
Y <- beta0 + X %*% beta + e
dat <- data.frame(Y,X)
colnames(dat) <- c("Y", paste("X", 1:p, sep=""))
#(b)
model.for <- step(model1,direction = 'forward')
```

```
## Start:  AIC=-3209.41
## Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10
```

```
summary(model.for)
```

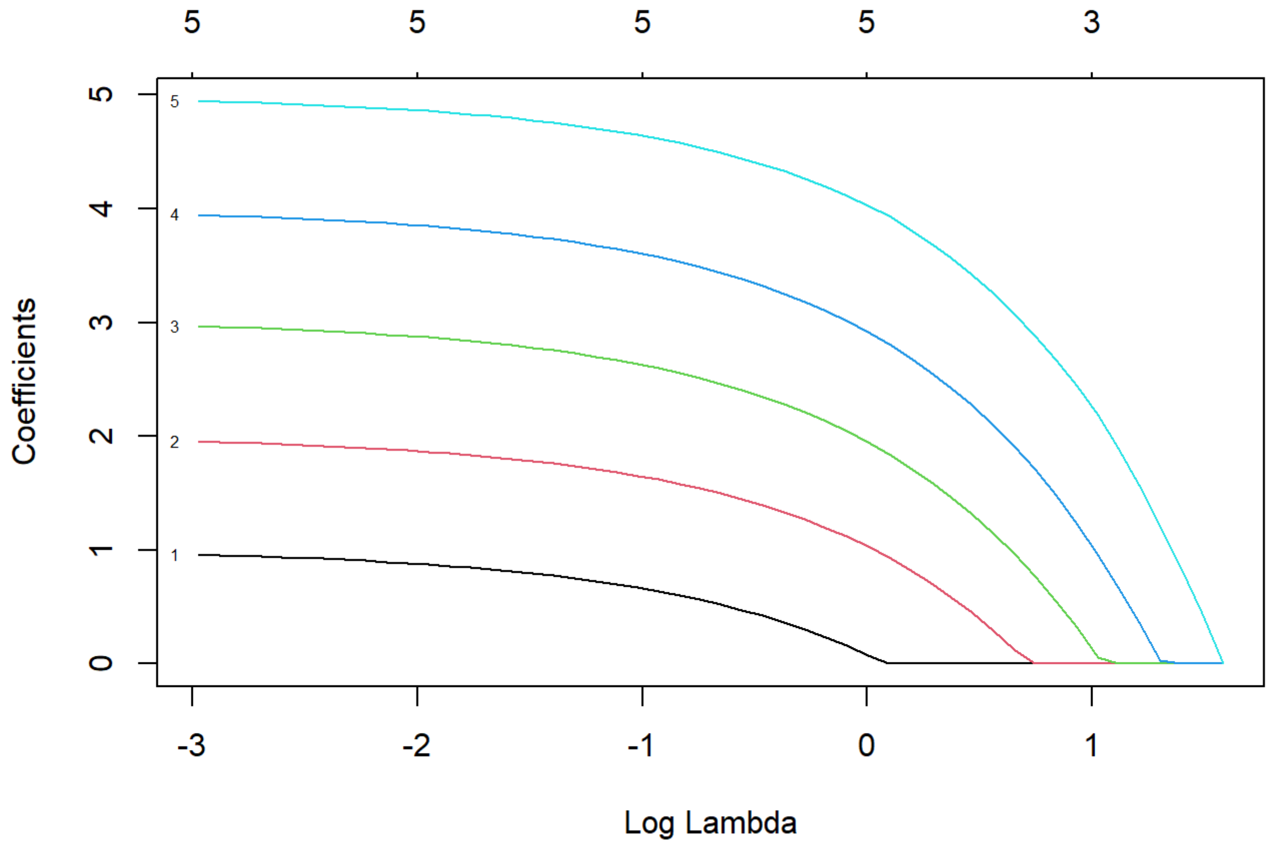
```
##
## Call:
## lm(formula = Y ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 +
##     X10, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68942 -0.13741 -0.00005  0.12912  0.66300
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.007e+00  6.354e-03 158.448   <2e-16 ***
## X1           9.954e-01  6.552e-03 151.909   <2e-16 ***
## X2           1.994e+00  6.233e-03 319.895   <2e-16 ***
## X3           3.002e+00  6.706e-03 447.655   <2e-16 ***
## X4           3.997e+00  6.133e-03 651.807   <2e-16 ***
## X5           4.993e+00  6.421e-03 777.670   <2e-16 ***
## X6          -5.415e-05  6.244e-03  -0.009    0.993
## X7          -4.716e-03  6.529e-03  -0.722    0.470
## X8           4.675e-03  6.596e-03   0.709    0.479
## X9           7.207e-03  6.155e-03   1.171    0.242
## X10         -5.122e-03  6.349e-03  -0.807    0.420
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1999 on 989 degrees of freedom
## Multiple R-squared:  0.9993, Adjusted R-squared:  0.9993
## F-statistic: 1.495e+05 on 10 and 989 DF,  p-value: < 2.2e-16
```

```
#(c)
X=model.matrix(Y~.,dat)[,-1]
train=sample(1:n, n/2)
test=(-train)
Y.test=Y[test]
#lasso
cv.lasso <- cv.glmnet(X[train,], Y[train], alpha=1/2)
M.lasso <- glmnet(X[train,],Y[train],alpha=1,
                  lambda=cv.lasso$lambda.min)
coef(M.lasso) # variable selected given lambda
```

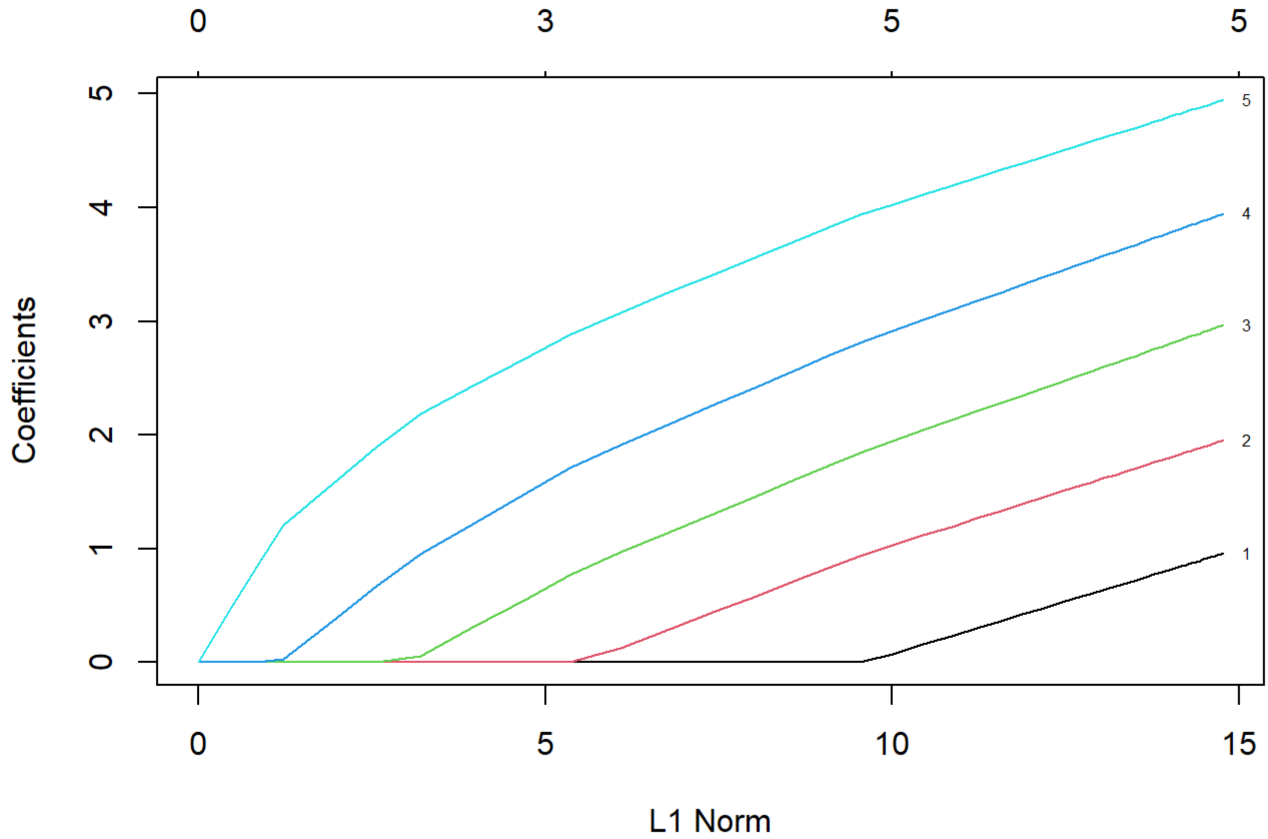
```
## 101 x 1 sparse Matrix of class "dgCMatrix"
##                    s0
## (Intercept) 1.0128242
## X1          0.9394396
## X2          1.9335884
## X3          2.9451934
## X4          3.9234777
## X5          4.9285700
## X6          .
## X7          .
## X8          .
## X9          .
## X10         .
## X11         .
## X12         .
## X13         .
## X14         .
## X15         .
## X16         .
## X17         .
## X18         .
## X19         .
## X20         .
## X21         .
## X22         .
## X23         .
## X24         .
## X25         .
## X26         .
## X27         .
## X28         .
## X29         .
## X30         .
## X31         .
## X32         .
## X33         .
## X34         .
## X35         .
## X36         .
## X37         .
## X38         .
## X39         .
## X40         .
## X41         .
## X42         .
## X43         .
## X44         .
## X45         .
## X46         .
## X47         .
## X48         .
## X49         .
## X50         .
## X51         .
## X52         .
```

```
## X53         .
## X54         .
## X55         .
## X56         .
## X57         .
## X58         .
## X59         .
## X60         .
## X61         .
## X62         .
## X63         .
## X64         .
## X65         .
## X66         .
## X67         .
## X68         .
## X69         .
## X70         .
## X71         .
## X72         .
## X73         .
## X74         .
## X75         .
## X76         .
## X77         .
## X78         .
## X79         .
## X80         .
## X81         .
## X82         .
## X83         .
## X84         .
## X85         .
## X86         .
## X87         .
## X88         .
## X89         .
## X90         .
## X91         .
## X92         .
## X93         .
## X94         .
## X95         .
## X96         .
## X97         .
## X98         .
## X99         .
## X100        .
```

```
M.lasso <- glmnet(X[train,],Y[train],alpha=1)
plot(M.lasso, label=T, xvar="lambda")
```




```
plot(M.lasso, label = T, xvar="norm")
```

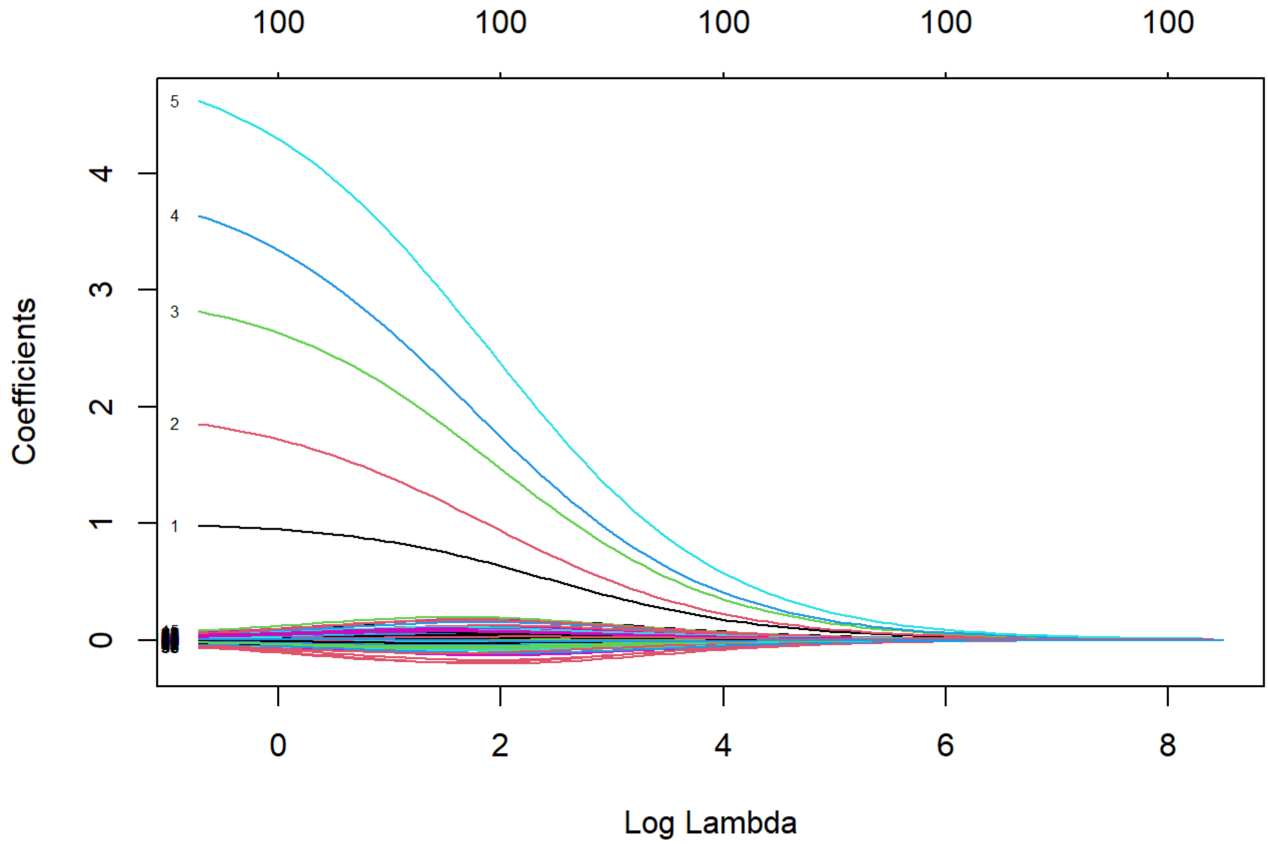


```
#ridge
cv.out=cv.glmnet(X[train,], Y[train],alpha=0)
ridge.mod=glmnet(X[train,], Y[train],alpha=0,
                 lambda = cv.out$lambda.min)
coef(ridge.mod)
```

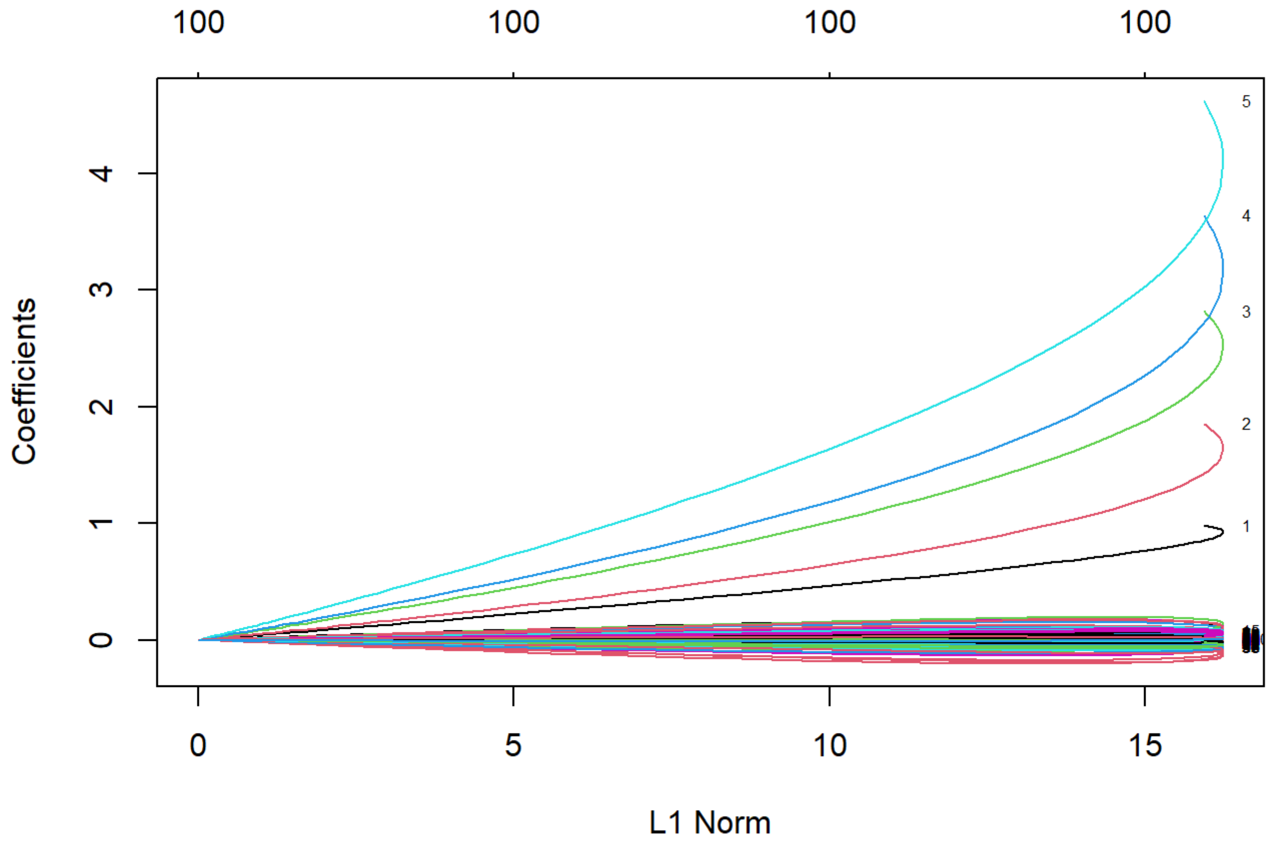
```
## 101 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept)  1.0123845529
## X1           0.9793376484
## X2           1.8544233341
## X3           2.8190826759
## X4           3.6433176931
## X5           4.6241471260
## X6          -0.0166127169
## X7           0.0463588400
## X8           0.0604474509
## X9          -0.0062017967
## X10          0.0437142906
## X11          0.0287422629
## X12         -0.0330228955
## X13          0.0094059397
## X14          0.0079572532
## X15          0.0791132303
## X16          0.0328846445
## X17          0.0166288351
## X18          0.0595502069
## X19         -0.0077817022
## X20          0.0261443760
## X21          0.0016203666
## X22         -0.0176577272
## X23         -0.0178830260
## X24          0.0076468973
## X25         -0.0034465257
## X26         -0.0504126322
## X27         -0.0243282861
## X28         -0.0255723625
## X29         -0.0074189329
## X30          0.0067559453
## X31          0.0384906339
## X32          0.0030734964
## X33         -0.0039699124
## X34          0.0161321973
## X35          0.0155557953
## X36          0.0016277629
## X37          0.0224225827
## X38         -0.0643723148
## X39          0.0193088906
## X40         -0.0185520098
## X41          0.0118675982
## X42         -0.0253195810
## X43          0.0142508059
## X44          0.0065891652
## X45          0.0215691220
## X46         -0.0024646561
## X47         -0.0416799215
## X48          0.0091124025
## X49         -0.0085658434
## X50          0.0333089137
## X51          0.0311598488
## X52         -0.0233365995
```

```
## X53          0.0040660481
## X54         -0.0190069620
## X55          0.0007025708
## X56         -0.0640835139
## X57          0.0222981919
## X58          0.0037357594
## X59         -0.0242832356
## X60         -0.0263291386
## X61         -0.0069347189
## X62          0.0024383253
## X63          0.0249659772
## X64         -0.0091055536
## X65         -0.0178437171
## X66         -0.0142816594
## X67          0.0331795778
## X68          0.0536158389
## X69          0.0069902957
## X70          0.0284463301
## X71         -0.0126818858
## X72          0.0224567851
## X73          0.0060537636
## X74          0.0370510285
## X75         -0.0178467985
## X76          0.0050533547
## X77          0.0381534022
## X78         -0.0024744732
## X79          0.0090469795
## X80          0.0013811775
## X81         -0.0202900108
## X82         -0.0240325021
## X83         -0.0235661383
## X84          0.0318600370
## X85          0.0183738625
## X86         -0.0232762374
## X87          0.0035195842
## X88         -0.0367999026
## X89          0.0173848242
## X90          0.0416978820
## X91         -0.0290914755
## X92         -0.0503610911
## X93         -0.0230274584
## X94         -0.0152181831
## X95          0.0144086499
## X96          0.0249713897
## X97         -0.0007545796
## X98          0.0028349384
## X99         -0.0230499188
## X100         0.0063769492
```

```
M.ridge <- glmnet(X[train,],Y[train],alpha=0)
plot(M.ridge, label=T, xvar="lambda")
```




```
plot(M.ridge, label = T, xvar="norm")
```



2. 请基于 wineTrain 数据集，进行主成分分析。

- 请计算 wineTrain 的主成分，并输出计算结果，你应该得到一个 13*13 的得分矩阵。

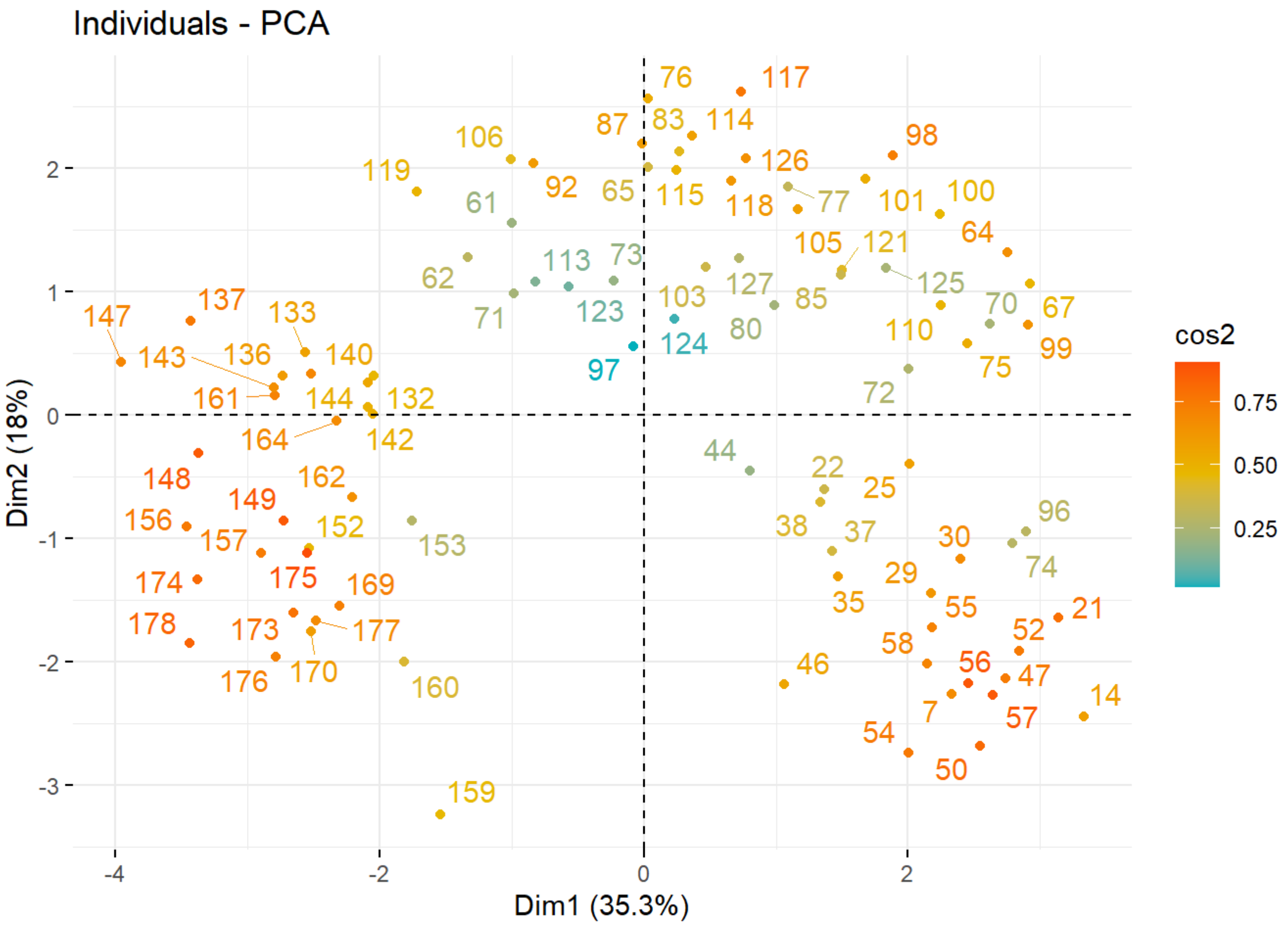
```
######### Please write your R code in this chunk #########
### Solution to Q2.1
data("wine")
train=sample(1:nrow(wine), nrow(wine)/2)
test=(-train)
wineTrain <- wine[train,2:14]
res.pca <- prcomp(wineTrain, scale = TRUE)
res.pca$rotation
```

```
##                        PC1         PC2         PC3         PC4         PC5
## Alcohol         -0.01699901 -0.53892649  0.18699763 -0.18488455  0.11890574
## Malic           -0.27064199 -0.13106142 -0.04757265 -0.33271419 -0.14468587
## Ash             -0.09905796 -0.27708565 -0.59259487  0.01088589  0.46116170
## Alcalinity      -0.20132049  0.06144272 -0.68974966 -0.04908662 -0.13192059
## Magnesium        0.09047081 -0.27565853 -0.18608460  0.68615429 -0.25693311
## Phenols          0.36852231 -0.15141592 -0.13969152 -0.33450798  0.01751657
## Flavanoids       0.42740497 -0.03758125 -0.04655429 -0.24153551  0.04372058
## Nonflavanoid    -0.30466091  0.03404841  0.08843751 -0.14739240  0.51203194
## Proanthocyanins  0.32237402 -0.16079193 -0.13652131 -0.16563965 -0.35181705
## Intensity       -0.22063712 -0.46691560  0.12137882 -0.10755687 -0.18148181
## Hue              0.33746242  0.18160571  0.02318342  0.24127210  0.42264190
## OD280            0.39902393  0.04896639 -0.13478220 -0.19039098  0.05411730
## Proline          0.17586317 -0.48072951  0.14014692  0.23739805  0.25725256
##                          PC6          PC7          PC8         PC9        PC10
## Alcohol         -0.107544666  0.235534761 -0.592828133 -0.15900485  0.09545434
## Malic           -0.749082697 -0.270285510  0.013348848  0.03671071  0.19390942
## Ash              0.014587394  0.100137393  0.380497525 -0.20263381  0.32160709
## Alcalinity       0.106488050  0.122859512 -0.442021961  0.05281206 -0.33210787
## Magnesium       -0.169556391 -0.335865642 -0.107944449  0.31892763  0.07372546
## Phenols          0.147604593 -0.106359564 -0.004911892  0.59761604  0.12798613
## Flavanoids       0.008766959 -0.003531857  0.092122897  0.18889174  0.18604322
## Nonflavanoid     0.155619788 -0.651725005 -0.097120144  0.15051818 -0.28792901
## Proanthocyanins  0.195071311 -0.517240615  0.030500296 -0.60808374 -0.02431480
## Intensity        0.458178022  0.003574441 -0.003682476  0.11975010  0.05658313
## Hue             -0.021111293 -0.127312568 -0.462859381 -0.15828427  0.31807991
## OD280           -0.262330198  0.087202363 -0.033804475 -0.00103774 -0.51095813
## Proline         -0.153893347  0.078152082  0.249588409 -0.05624693 -0.48392547
##                        PC11         PC12        PC13
## Alcohol         -0.29918079 -0.285202635  0.04742204
## Malic            0.25258219  0.194921223 -0.02412479
## Ash             -0.20082228  0.028100700 -0.09393769
## Alcalinity       0.30061918 -0.024602113  0.18506281
## Magnesium       -0.28340110  0.006720251  0.06048859
## Phenols          0.17631862 -0.284518802 -0.43687550
## Flavanoids      -0.01973551  0.105175714  0.81435551
## Nonflavanoid    -0.17113554 -0.052467826  0.13529822
## Proanthocyanins  0.04547045 -0.134669259 -0.05735670
## Intensity        0.08578989  0.660887327 -0.04944688
## Hue              0.33981241  0.351667252 -0.13963724
## OD280           -0.44546503  0.443075975 -0.21826525
## Proline          0.50156881 -0.097802360  0.07227708
```

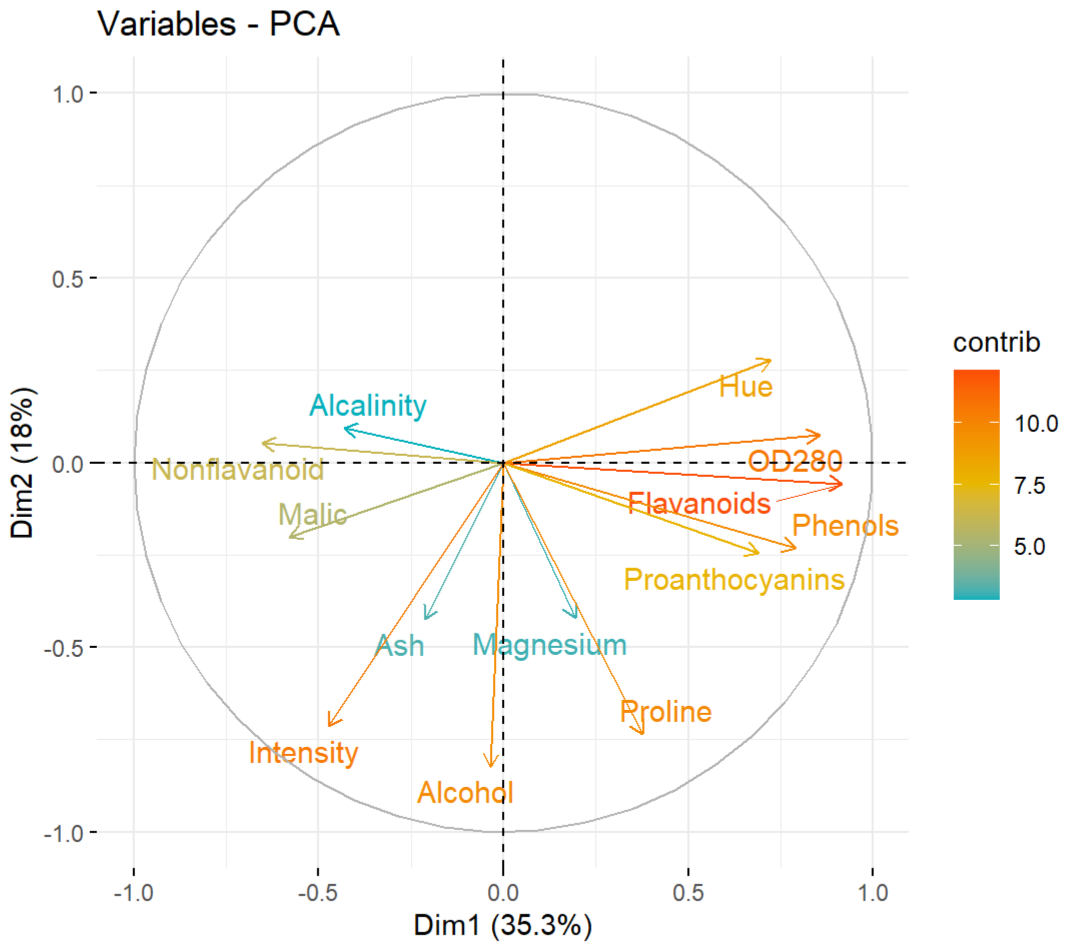
- 通过合适的图表，将 fviz_pca_ind(), fviz_pca_var() 和 fviz_pca_var()进行展示。这三个图展示的分别是什么?

```
######### Please write your R code in this chunk #########
### Solution to Q2.2
fviz_pca_ind(res.pca,
             col.ind = "cos2", # Color by the quality of representation
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE     # Avoid text overlapping
)
```

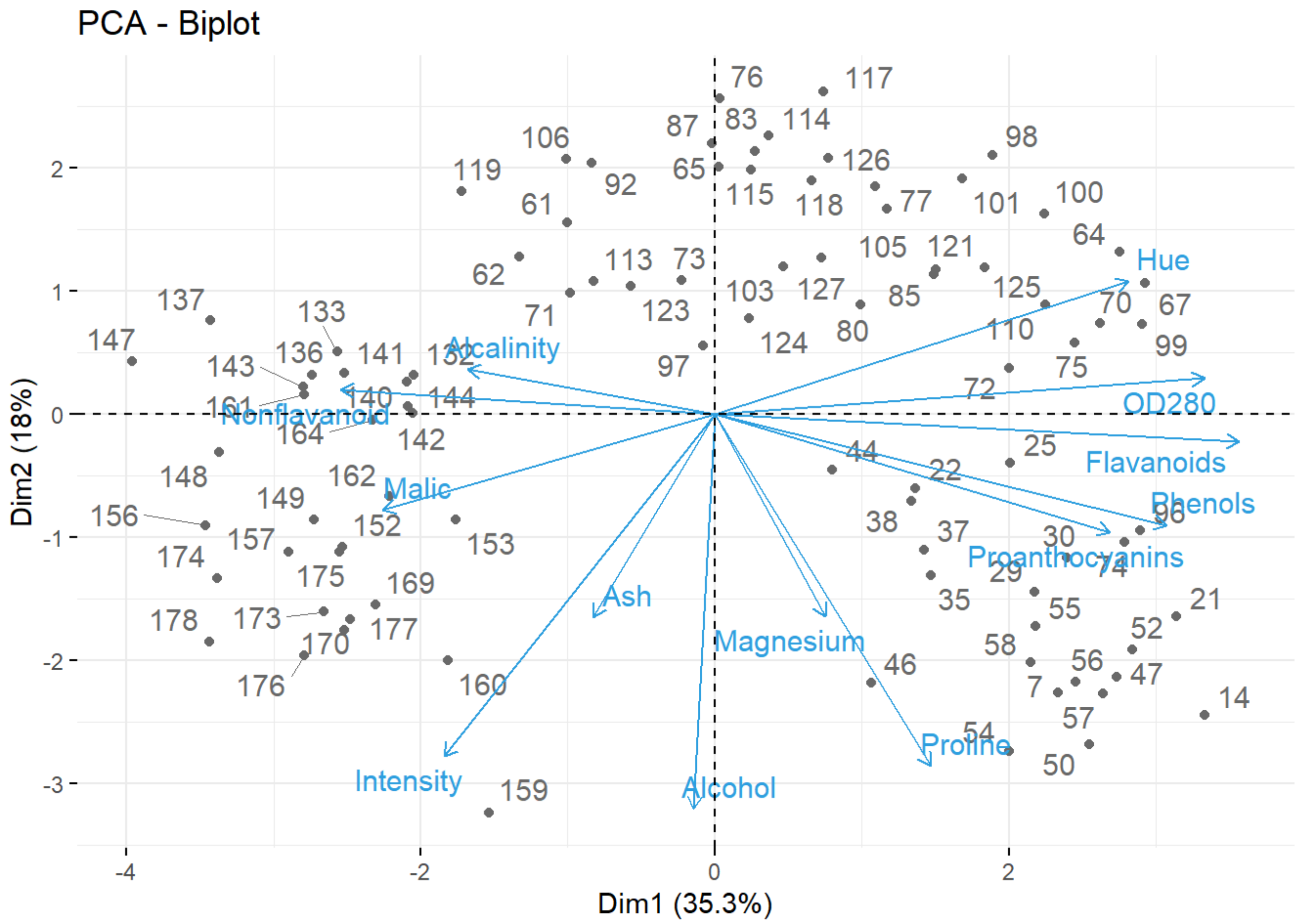
```
## Warning: ggrepel: 1 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```




```
fviz_pca_var(res.pca,
             col.var = "contrib", # Color by contributions to the PC
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE     # Avoid text overlapping
)
```




```
fviz_pca_biplot(res.pca, repel = TRUE,
                col.var = "#2E9FDF", # Variables color
                col.ind = "#696969"  # Individuals color
)
```



第一张图代表训练样本投影到主成分一和主成分二的坐标位置。

第二张图代表原来十三个变量对主成分一和主成分二的影响

第三张图是样本数据和变量在主成分一主成分二上的联合投影

- 计算每个主成分对原始变量的解释程度，按照降序排列，并计算累计贡献率。你觉得选多少个主成分比较合适?

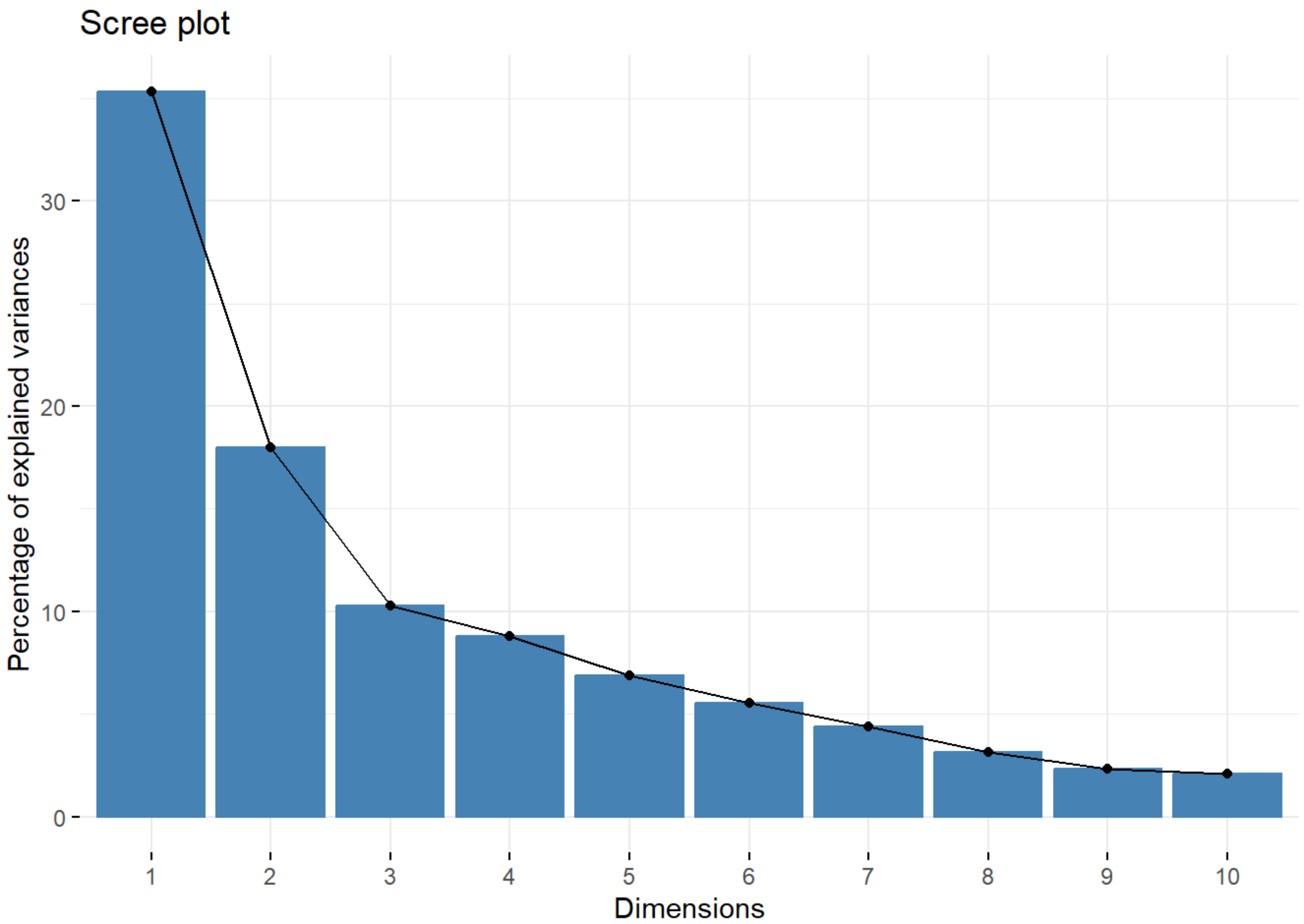
```
######### Please write your R code in this chunk #########
### Solution to Q2.3
eig.val <- get_eigenvalue(res.pca)
eig.val
```

	eigenvalue <dbl>	variance.percent <dbl>	cumulative.variance.percent <dbl>
Dim.1	4.59464684	35.3434372	35.34344
Dim.2	2.33625296	17.9711766	53.31461
Dim.3	1.33778997	10.2906921	63.60531
Dim.4	1.14518076	8.8090828	72.41439
Dim.5	0.89674421	6.8980324	79.31242
Dim.6	0.72043695	5.5418227	84.85424
Dim.7	0.56817734	4.3705950	89.22484

	eigenvalue <dbl>	variance.percent <dbl>	cumulative.variance.percent <dbl>
Dim.8	0.40597856	3.1229120	92.34775
Dim.9	0.30318460	2.3321893	94.67994
Dim.10	0.26972552	2.0748117	96.75475

1-10 of 13 rows Previous **1** 2 Next

```
fviz_eig(res.pca)
```



我认为选择四个主成分比较合适，因为拐点出现在主成分为4时，累计概率达到74.93009%

- 展示原始变量到前 m 个主成分的变换矩阵，其中 m 为你在(c)中的主成分的个数。在前 m 个主成分上，原始的 13 个变量对每个主成分的影响有多少是正的影响，有多少是负影响？请通过 apply 函数进行展示。

```
########## Please write your R code in this chunk ##########
### Solution to Q2.4
m = 4
res.pca$rotation[,1:m]
```

```
##                         PC1         PC2         PC3         PC4
## Alcohol         -0.01699901 -0.53892649  0.18699763 -0.18488455
## Malic           -0.27064199 -0.13106142 -0.04757265 -0.33271419
## Ash             -0.09905796 -0.27708565 -0.59259487  0.01088589
## Alcalinity      -0.20132049  0.06144272 -0.68974966 -0.04908662
## Magnesium        0.09047081 -0.27565853 -0.18608460  0.68615429
## Phenols          0.36852231 -0.15141592 -0.13969152 -0.33450798
## Flavanoids       0.42740497 -0.03758125 -0.04655429 -0.24153551
## Nonflavanoid    -0.30466091  0.03404841  0.08843751 -0.14739240
## Proanthocyanins  0.32237402 -0.16079193 -0.13652131 -0.16563965
## Intensity       -0.22063712 -0.46691560  0.12137882 -0.10755687
## Hue              0.33746242  0.18160571  0.02318342  0.24127210
## OD280            0.39902393  0.04896639 -0.13478220 -0.19039098
## Proline          0.17586317 -0.48072951  0.14014692  0.23739805
```

```
res.var <- get_pca_var(res.pca)
# 计数正影响
countp_func <- function(x){
  y <- ifelse(x >0, 1, 0)
  sum(y)
}
apply(res.var$coord[,1:m],2,countp_func)
```

```
## Dim.1 Dim.2 Dim.3 Dim.4
##     7     4     5     4
```

```
# 计数负影响
countn_func <- function(x){
  y <- ifelse(x <0, 1, 0)
  sum(y)
}
apply(res.var$coord[,1:m],2,countn_func)
```

```
## Dim.1 Dim.2 Dim.3 Dim.4
##     6     9     8     9
```