Multivariate Statistics HW2

Namwoo Kwon 20236002

2023-03-11

Goal : Perform the PCA of IRIS Dataset.

Process :

1. Intro 1 ~ 2
2. Quiz a - 1 : How many components do you need to adequately describe the data?
3. Quiz a - 2 : How would you interpret them?
4. Quiz b - 1 : Plot the average PC Scores for each of the three different types of iris for the fist two PC
5. Quiz b - 2 : Describe your findings.

Intro 1 : Set path of file

```
setwd('/Users/namwoo/Desktop/UNIST/lecture/1-1/Multivaraiate Statistics')
getwd()
```

```
## [1] "/Users/namwoo/Desktop/UNIST/lecture/1-1/Multivaraiate Statistics"
```

Intro 2 : Load IRIS Dataset

```
#install.packages('readxl')
library(readxl)
iris <- read_excel('IRIS.xlsx')
```

```
str(iris)
```

```
## tibble [150 × 5] (S3: tbl_df/tbl/data.frame)
##  $ X1: num [1:150] 1 1 1 1 1 1 1 1 1 1 ...
##  $ X2: num [1:150] 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ X3: num [1:150] 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ X4: num [1:150] 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ X5: num [1:150] 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
```

a - 1 : How many components do you need to adequately describe the data?

Solution Step 1 : Select Feature

```
a_iris <- iris[, 2:5]
str(a_iris)
```

```
## tibble [150 × 4] (S3: tbl_df/tbl/data.frame)
##  $ X2: num [1:150] 5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ X3: num [1:150] 3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ X4: num [1:150] 1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ X5: num [1:150] 0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
```

```
summary(a_iris)
```

```
##        X2              X3              X4              X5
##  Min.   :4.300   Min.   :2.000   Min.   :1.000   Min.   :0.100
##  1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.600   1st Qu.:0.300
##  Median :5.800   Median :3.000   Median :4.350   Median :1.300
##  Mean   :5.843   Mean   :3.057   Mean   :3.758   Mean   :1.199
##  3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
##  Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
```

Solution Step 2 : Check Coefficient

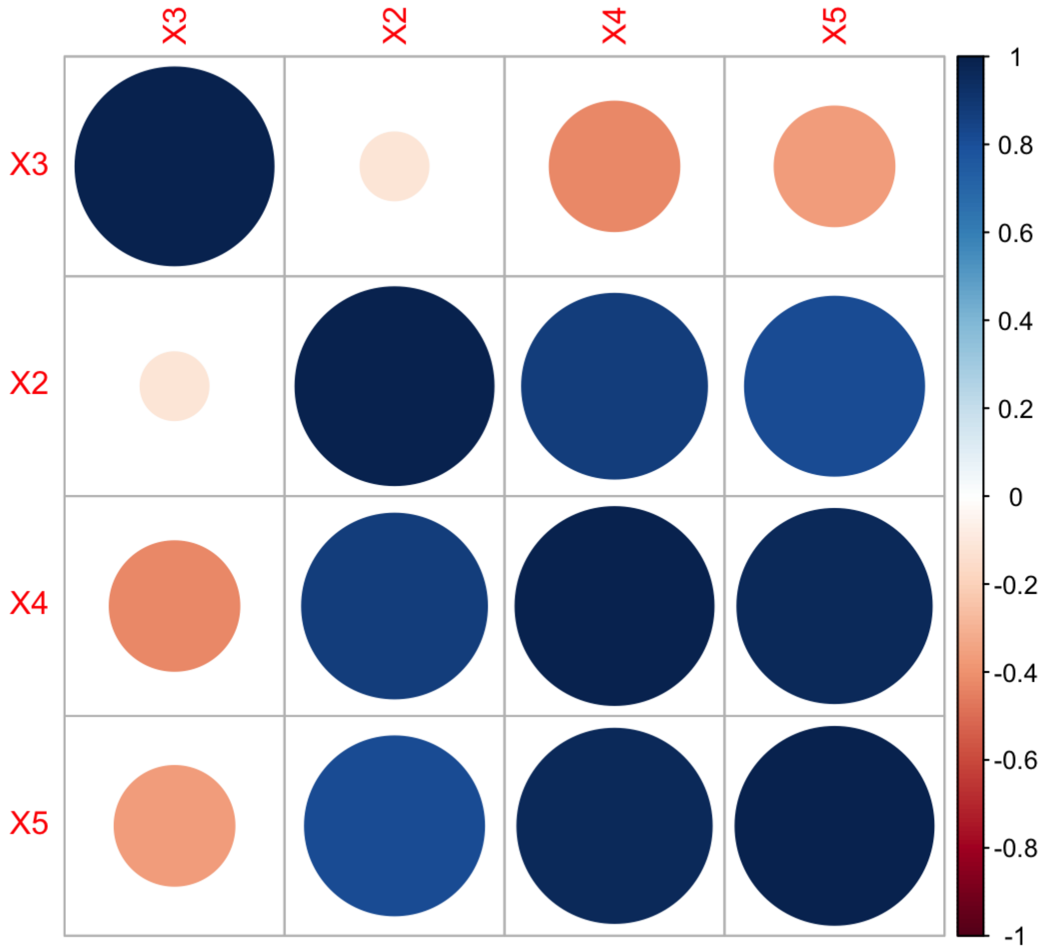
```
cor(a_iris)
```

```
##            X2         X3         X4         X5
## X2  1.0000000 -0.1175698  0.8717538  0.8179411
## X3 -0.1175698  1.0000000 -0.4284401 -0.3661259
## X4  0.8717538 -0.4284401  1.0000000  0.9628654
## X5  0.8179411 -0.3661259  0.9628654  1.0000000
```

```
#install.packages("corrplot")
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
corrplot(cor(a_iris), order="hclust")
```



Multicollinearity Issues

- You can see when variables have a high correlation between them
- EX : corr(X2 & X4), corr(X2 & X5)
- High correlation coefficients between variables can lead to multicollinearity.

Solution Step 3 : Perform the PCA of IRIS Dataset

```
set.seed(42)


iris.pca <- prcomp(a_iris,
                   # Standardization
                   center=TRUE, # E(Xi) = 0
                   scale=TRUE) # Var(Xi) = 1
```

Set PCA parameters

- 'center=TRUE' : Make the mean of all the variavles to be zero
- 'scale=TRUE' : I don't know what unit of measure each variable uses -> Need to scaled Data

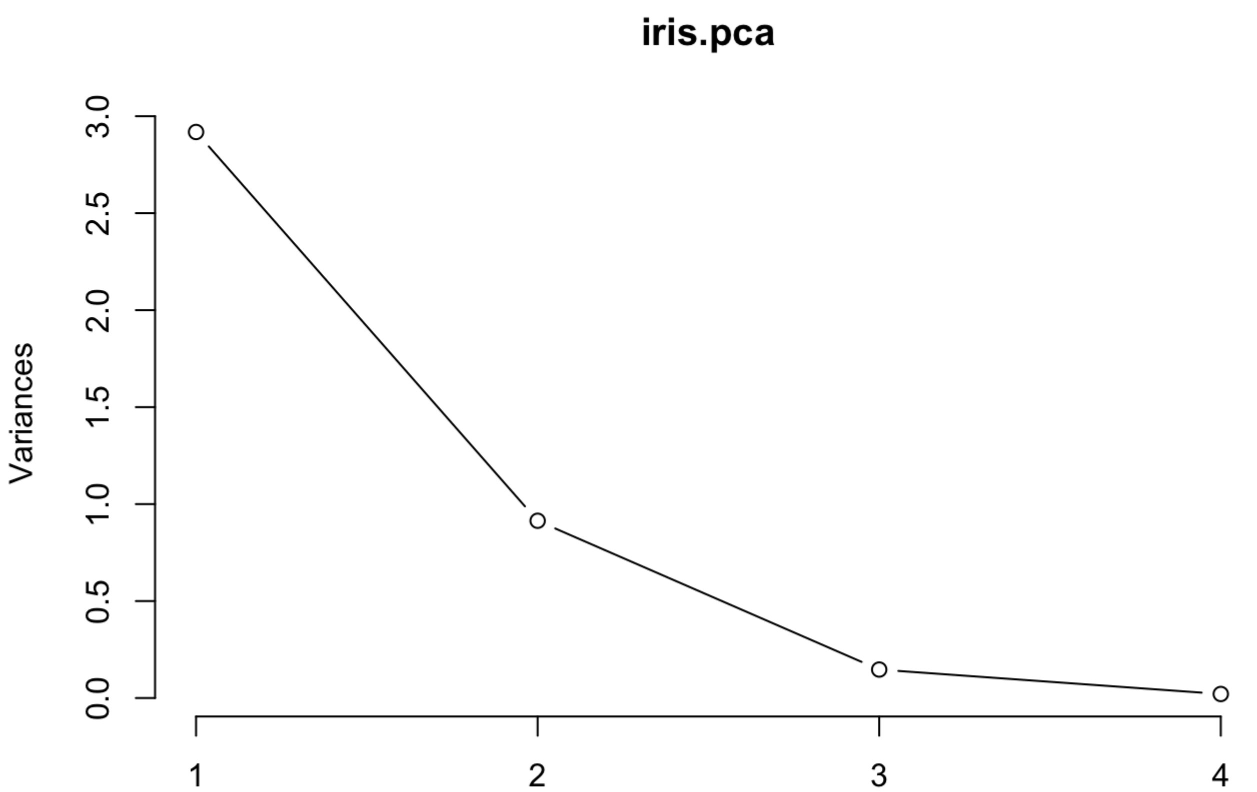
```
print(iris.pca)
```

```
## Standard deviations (1, .., p=4):
## [1] 1.7083611 0.9560494 0.3830886 0.1439265
##
## Rotation (n x k) = (4 x 4):
##           PC1         PC2        PC3        PC4
## X2  0.5210659 -0.37741762  0.7195664  0.2612863
## X3 -0.2693474 -0.92329566 -0.2443818 -0.1235096
## X4  0.5804131 -0.02449161 -0.1421264 -0.8014492
## X5  0.5648565 -0.06694199 -0.6342727  0.5235971
```

```
summary(iris.pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.7084 0.9560 0.38309 0.14393
## Proportion of Variance 0.7296 0.2285 0.03669 0.00518
## Cumulative Proportion  0.7296 0.9581 0.99482 1.00000
```

```
plot(iris.pca, type='l')
```

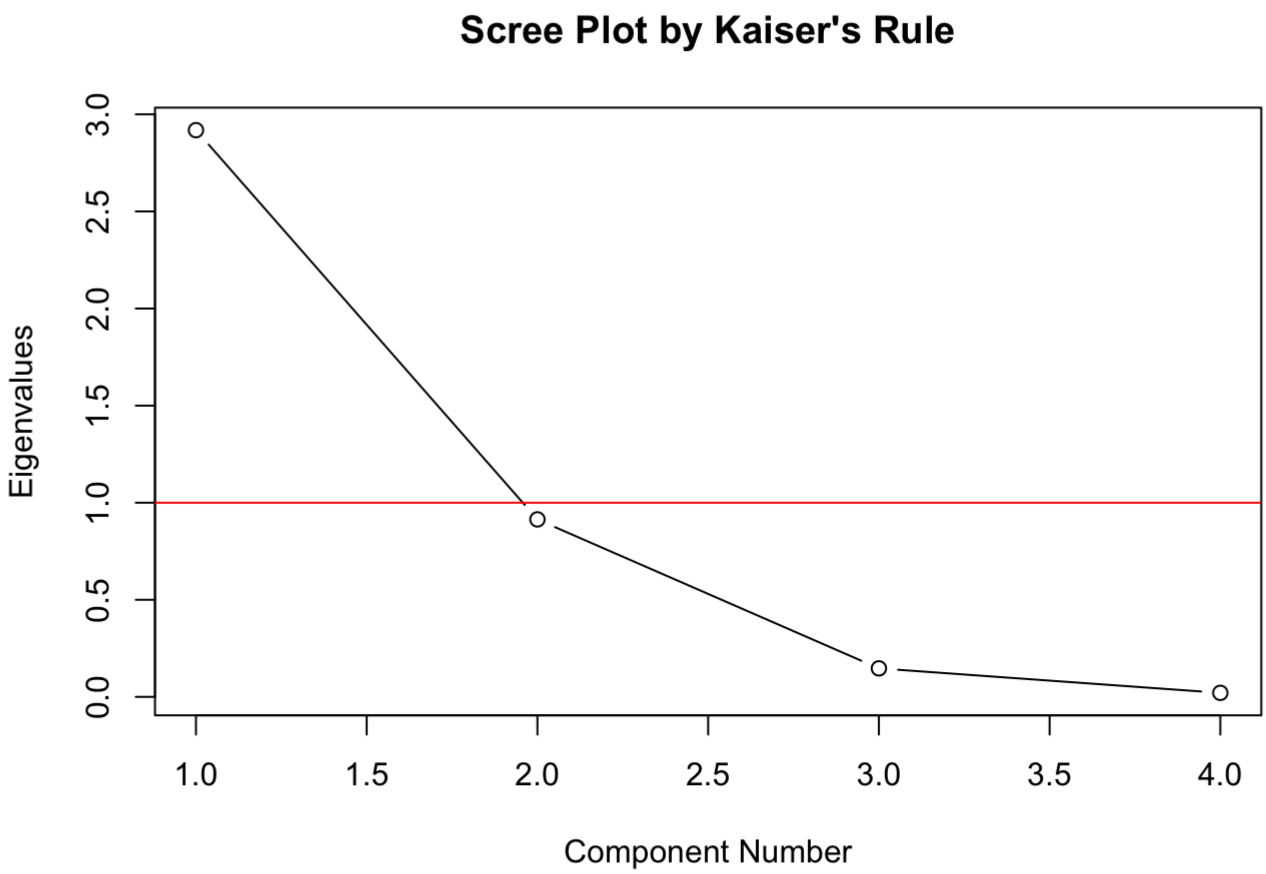


Solution Step 4 : Check Eigenvalues

```
iris.mat <- as.matrix(a_iris)
cov.mat <- cor(iris.mat)
eigen(cov.mat)
```

```
## eigen() decomposition
## $values
## [1] 2.91849782 0.91403047 0.14675688 0.02071484
##
## $vectors
##            [,1]        [,2]       [,3]       [,4]
## [1,]  0.5210659 -0.37741762  0.7195664  0.2612863
## [2,] -0.2693474 -0.92329566 -0.2443818 -0.1235096
## [3,]  0.5804131 -0.02449161 -0.1421264 -0.8014492
## [4,]  0.5648565 -0.06694199 -0.6342727  0.5235971
```

```
plot(eigen(cov.mat)$values,
     main="Scree Plot by Kaiser's Rule",
     xlab='Component Number',
     ylab='Eigenvalues',
     type='b')
abline(h=1, col='red')
```



Answer

- It seems desirable to use 1 Principal Components by Kaiser's Rule.
- Kaiser's Rule recommends using the number of principal components with an eigenvalue greater than 1 as the final number of dimensions to use.
- Personal Suggestion : Suggest using two principal components.
    - Using only one principal component seems insufficient to explain the variance of the original data. (72.96 %)

  - However, using two principal components can explain a significant portion of the variance in the original data (95.81%)

a - 2 : How would you interpret them?

Solution Step 5 : Check Principal Component Scores

```
iris.pca$x[,1:2]
```

```
##              PC1          PC2
##   [1,] -2.25714118 -0.478423832
##   [2,] -2.07401302  0.671882687
##   [3,] -2.35633511  0.340766425
##   [4,] -2.29170679  0.595399863
##   [5,] -2.38186270 -0.644675659
##   [6,] -2.06870061 -1.484205297
##   [7,] -2.43586845 -0.047485118
##   [8,] -2.22539189 -0.222403002
##   [9,] -2.32684533  1.111603700
##  [10,] -2.17703491  0.467447569
##  [11,] -2.15907699 -1.040205867
##  [12,] -2.31836413 -0.132633999
##  [13,] -2.21104370  0.726243183
##  [14,] -2.62430902  0.958296347
##  [15,] -2.19139921 -1.853846555
##  [16,] -2.25466121 -2.677315230
##  [17,] -2.20021676 -1.478655729
##  [18,] -2.18303613 -0.487206131
##  [19,] -1.89223284 -1.400327567
##  [20,] -2.33554476 -1.124083597
##  [21,] -1.90793125 -0.407490576
##  [22,] -2.19964383 -0.921035871
##  [23,] -2.76508142 -0.456813301
##  [24,] -1.81259716 -0.085272854
##  [25,] -2.21972701 -0.136796175
##  [26,] -1.94532930  0.623529705
##  [27,] -2.04430277 -0.241354991
##  [28,] -2.16133650 -0.525389422
##  [29,] -2.13241965 -0.312172005
##  [30,] -2.25769799  0.336604248
##  [31,] -2.13297647  0.502856075
##  [32,] -1.82547925 -0.422280389
##  [33,] -2.60621687 -1.787587272
##  [34,] -2.43800983 -2.143546796
##  [35,] -2.10292986  0.458665270
##  [36,] -2.20043723  0.205419224
##  [37,] -2.03831765 -0.659349230
##  [38,] -2.51889339 -0.590315163
##  [39,] -2.42152026  0.901161067
##  [40,] -2.16246625 -0.267981199
##  [41,] -2.27884081 -0.440240541
##  [42,] -1.85191836  2.329610745
##  [43,] -2.54511203  0.477501017
##  [44,] -1.95788857 -0.470749613
##  [45,] -2.12992356 -1.138415464
##  [46,] -2.06283361  0.708678586
##  [47,] -2.37677076 -1.116688691
##  [48,] -2.38638171  0.384957230
##  [49,] -2.22200263 -0.994627669
##  [50,] -2.19647504 -0.009185585
##  [51,]  1.09810244 -0.860091033
##  [52,]  0.72889556 -0.592629362
##  [53,]  1.23683580 -0.614239894
##  [54,]  0.40612251  1.748546197
```

```
##  [55,]  1.07188379  0.207725147
##  [56,]  0.38738955  0.591302717
##  [57,]  0.74403715 -0.770438272
##  [58,] -0.48569562  1.846243998
##  [59,]  0.92480346 -0.032118478
##  [60,]  0.01138804  1.030565784
##  [61,] -0.10982834  2.645211115
##  [62,]  0.43922201  0.063083852
##  [63,]  0.56023148  1.758832129
##  [64,]  0.71715934  0.185602819
##  [65,] -0.03324333  0.437537419
##  [66,]  0.87248429 -0.507364239
##  [67,]  0.34908221  0.195656268
##  [68,]  0.15827980  0.789451008
##  [69,]  1.22100316  1.616827281
##  [70,]  0.16436725  1.298259939
##  [71,]  0.73521959 -0.395247446
##  [72,]  0.47469691  0.415926887
##  [73,]  1.23005729  0.930209441
##  [74,]  0.63074514  0.414997441
##  [75,]  0.70031506  0.063200094
##  [76,]  0.87135454 -0.249956017
##  [77,]  1.25231375  0.076998069
##  [78,]  1.35386953 -0.330205463
##  [79,]  0.66258066  0.225173502
##  [80,] -0.04012419  1.055183583
##  [81,]  0.13035846  1.557055553
##  [82,]  0.02337438  1.567225244
##  [83,]  0.24073180  0.774661195
##  [84,]  1.05755171  0.631726901
##  [85,]  0.22323093  0.286812663
##  [86,]  0.42770626 -0.842758920
##  [87,]  1.04522645 -0.520308714
##  [88,]  1.04104379  1.378371048
##  [89,]  0.06935597  0.218770433
##  [90,]  0.28253073  1.324886147
##  [91,]  0.27814596  1.116288852
##  [92,]  0.62248441 -0.024839814
##  [93,]  0.33540673  0.985103828
##  [94,] -0.36097409  2.012495825
##  [95,]  0.28762268  0.852873116
##  [96,]  0.09105561  0.180587142
##  [97,]  0.22695654  0.383634868
##  [98,]  0.57446378  0.154356489
##  [99,] -0.44617230  1.538637456
## [100,]  0.25587339  0.596852285
## [101,]  1.83841002 -0.867515056
## [102,]  1.15401555  0.696536401
## [103,]  2.19790361 -0.560133976
## [104,]  1.43534213  0.046830701
## [105,]  1.86157577 -0.294059697
## [106,]  2.74268509 -0.797736709
## [107,]  0.36579225  1.556289178
## [108,]  2.29475181 -0.418663020
## [109,]  1.99998633  0.709063226
## [110,]  2.25223216 -1.914596301
```

```
## [111,]  1.35962064 -0.690443405
## [112,]  1.59732747  0.420292431
## [113,]  1.87761053 -0.417849815
## [114,]  1.25590769  1.158379741
## [115,]  1.46274487  0.440794883
## [116,]  1.58476820 -0.673986887
## [117,]  1.46651849 -0.254768327
## [118,]  2.41822770 -2.548124795
## [119,]  3.29964148 -0.017721580
## [120,]  1.25954707  1.701046715
## [121,]  2.03091256 -0.907427443
## [122,]  0.97471535  0.569855257
## [123,]  2.88797650 -0.412259950
## [124,]  1.32878064  0.480202496
## [125,]  1.69505530 -1.010536476
## [126,]  1.94780139 -1.004412720
## [127,]  1.17118007  0.315338060
## [128,]  1.01754169 -0.064131184
## [129,]  1.78237879  0.186735633
## [130,]  1.85742501 -0.560413289
## [131,]  2.42782030 -0.258418706
## [132,]  2.29723178 -2.617554417
## [133,]  1.85648383  0.177953334
## [134,]  1.11042770  0.291944582
## [135,]  1.19845835  0.808606364
## [136,]  2.78942561 -0.853942542
## [137,]  1.57099294 -1.065013214
## [138,]  1.34179696 -0.421020154
## [139,]  0.92173701 -0.017165594
## [140,]  1.84586124 -0.673870645
## [141,]  2.00808316 -0.611835930
## [142,]  1.89543421 -0.687273065
## [143,]  1.15401555  0.696536401
## [144,]  2.03374499 -0.864624030
## [145,]  1.99147547 -1.045665670
## [146,]  1.86425786 -0.385674038
## [147,]  1.55935649  0.893692855
## [148,]  1.51609145 -0.268170747
## [149,]  1.36820418 -1.007877934
## [150,]  0.95744849  0.024250427
```

```
pcscores<-data.frame(iris.pca$x[,1:2])
pcscores$species<-iris$X1
head(pcscores)
```

```
##         PC1        PC2 species
## 1 -2.257141 -0.4784238       1
## 2 -2.074013  0.6718827       1
## 3 -2.356335  0.3407664       1
## 4 -2.291707  0.5953999       1
## 5 -2.381863 -0.6446757       1
## 6 -2.068701 -1.4842053       1
```

```
# Calculate the average PC score for each Iris species
iris.pca.mean <- aggregate (.~species , data=pcscores , mean)
iris.pca.mean
```

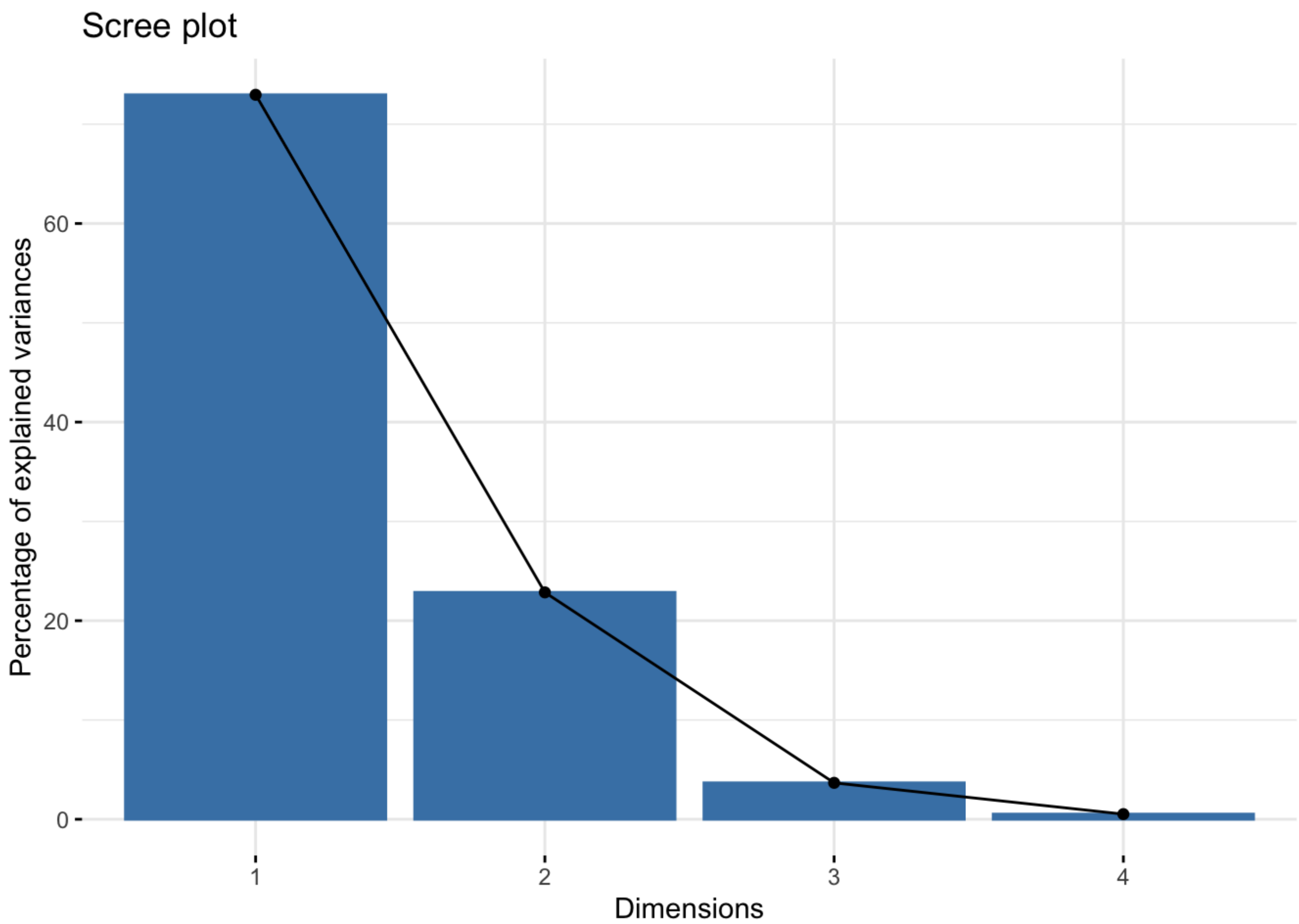
```
##   species        PC1        PC2
## 1       1 -2.2173249 -0.2879627
## 2       2  0.4947904  0.5483335
## 3       3  1.7225345 -0.2603708
```

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

```
## Loading required package: ggplot2
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve
3WBa
```

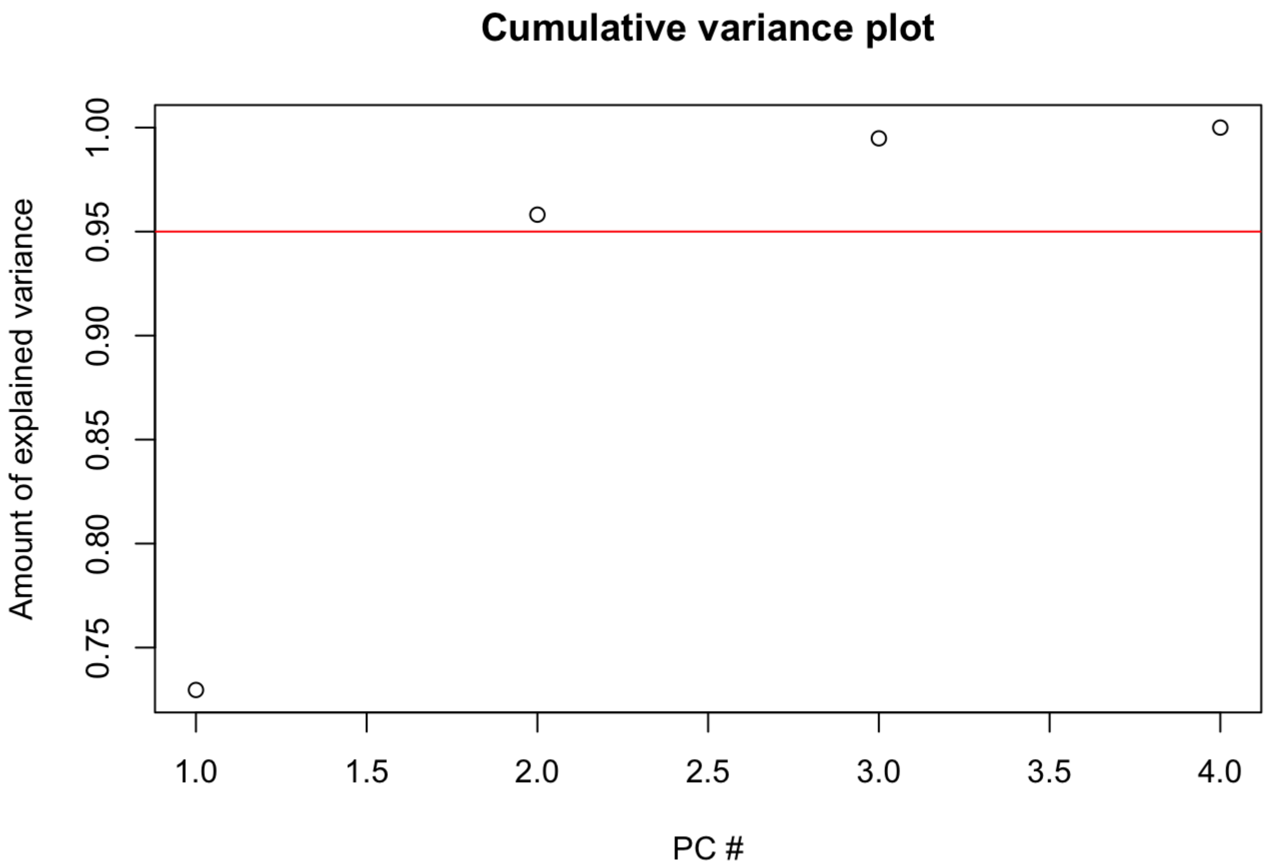
```
fviz_eig(iris.pca)
```



Percentage of explained variance by Number of Dimensions

- The first through second dimensions explain at least 25% of the variance in the original data.

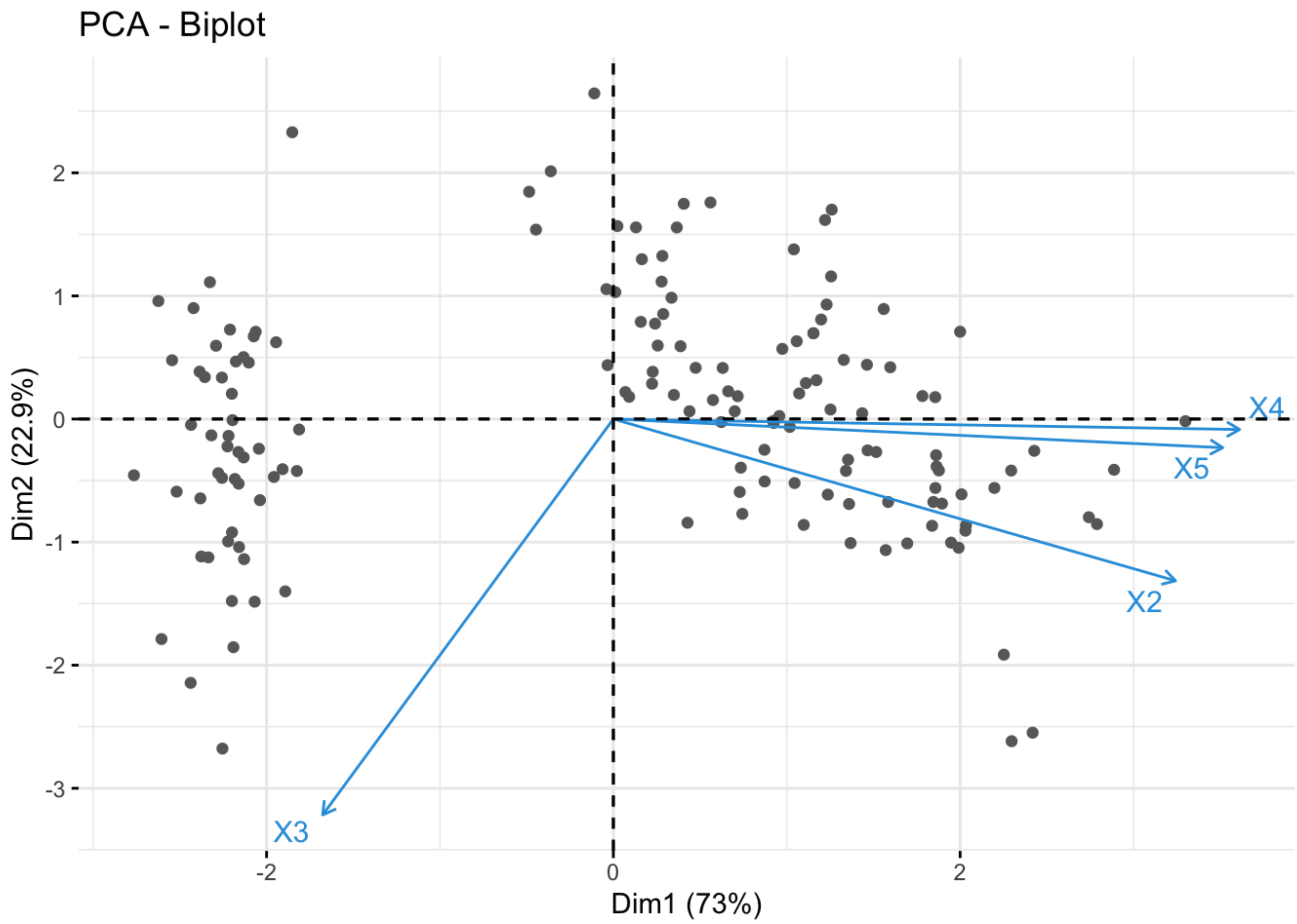
```
p<-length(iris.pca$sdev)
cumpro <- cumsum(iris.pca$sdev^2 / sum(iris.pca$sdev^2))
plot(cumpro[0:p], xlab = "PC #", ylab = "Amount of explained variance", main = "Cumul
ative variance plot")
abline(h=0.95, col='red')
```



Cumulative Amount of explained variance by Number of Dimensions

- Just two dimensions can explain more than 95% of the variance in your source data.

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

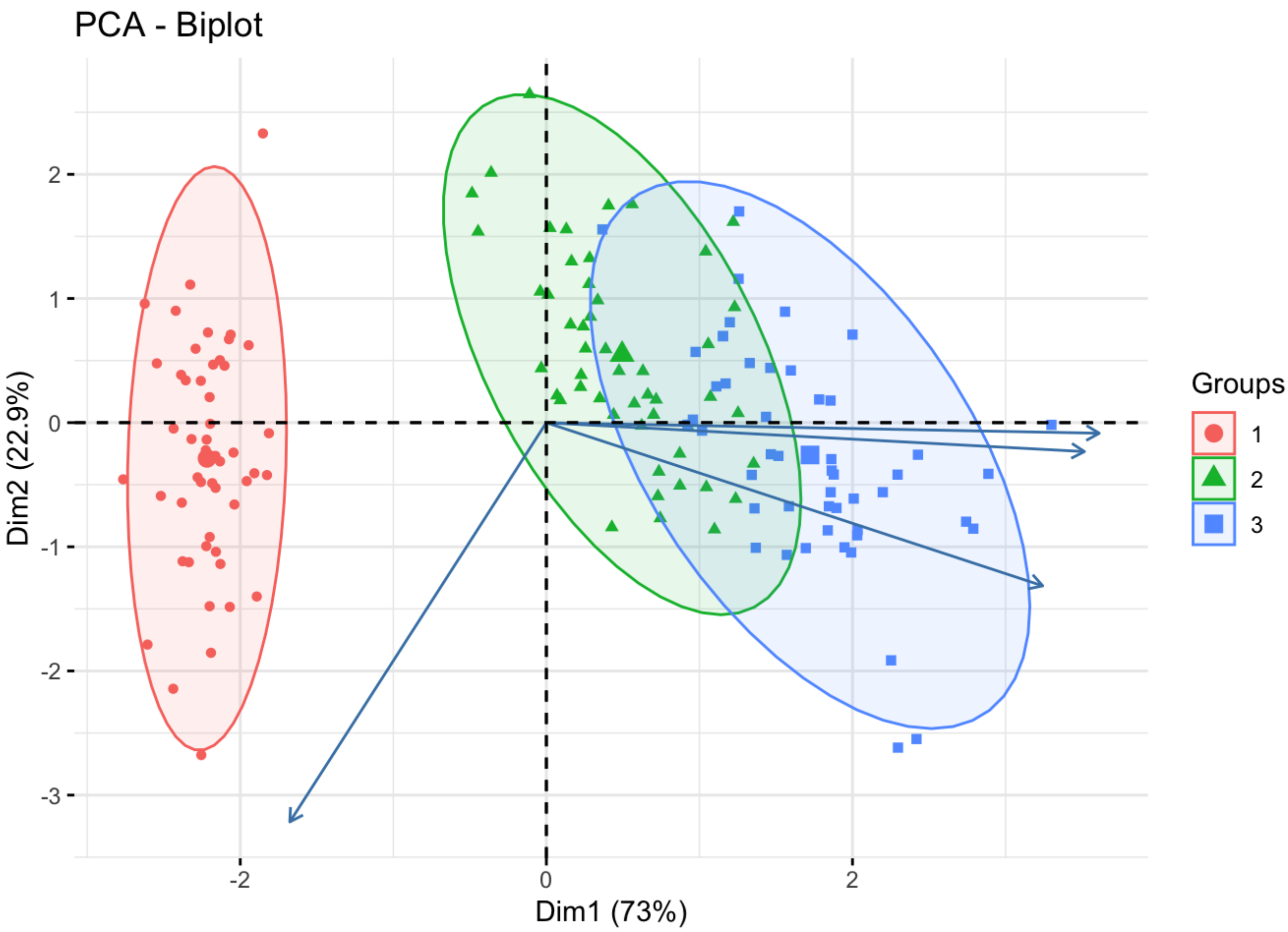


Explain the relationship between PC and source data variables with Biplot

- The variables that had the most impact on PC1 are X4, X5 and X2 (In order of greatest impact)
- The variables that had the most impact on PC2 are X3

b - 1 : Plot the average PC Scores for each of the three different types of iris for the fist two PC

```
fviz_pca_biplot(iris.pca,
                label='none',
                habillage=iris$X1,
                addEllipses = TRUE,
                palette = 'jco',
)
```



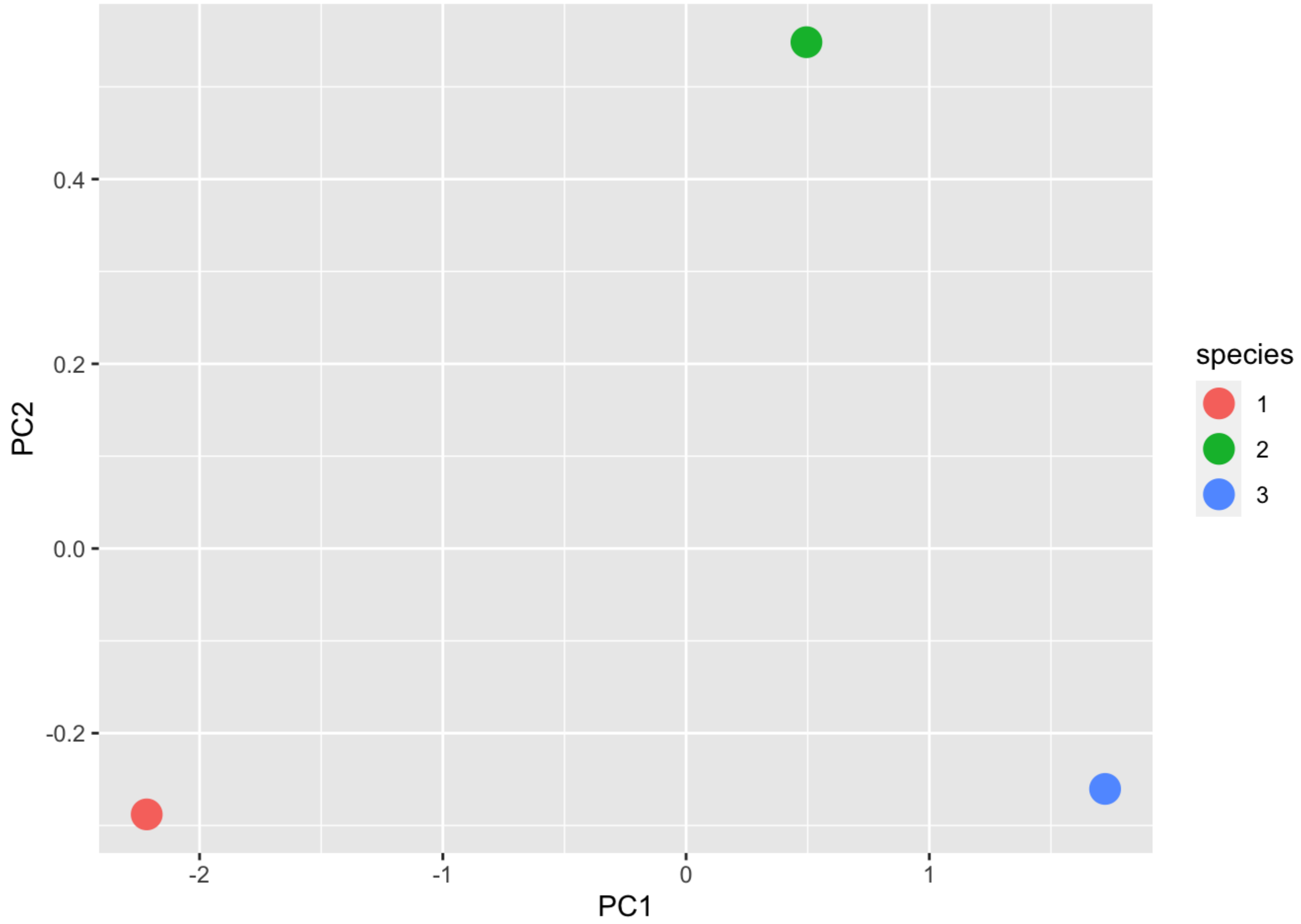

```
library("ggplot2")

iris.pca.mean
```

```
##   species        PC1        PC2
## 1       1 -2.2173249 -0.2879627
## 2       2  0.4947904  0.5483335
## 3       3  1.7225345 -0.2603708
```

```
iris.pca.mean$species = as.character(iris.pca.mean$species)

ggplot(data=iris.pca.mean, aes(x=PC1, y=PC2, color=species)) + geom_point(size=5)
```



b - 2 : Describe your findings.

- Based on PC1, it becomes easier to distinguish Specie 1 from the rest.
- Based on PC2, it becomes easier to distinguish Specie 2 from the rest
- Drawing a diagonal line between PC1 and PC2 makes it easier to distinguish PC3 from the rest.