Class 8: PCA mini project

Yaniv Iny (PID:18090586)

Today we will do a complete analysis of some breast cancer biopsy data but first let's revisit the main PCA function in R `prcomp()` and see what 'scale=TRUE/FALSE does.

```
head(mtcars)
```

```
                   mpg cyl disp  hp drat    wt  qsec vs am gear carb
Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

Find the mean value per columnof this dataset

```
apply(mtcars, 2, mean)
```

```
      mpg        cyl       disp         hp       drat         wt       qsec
20.090625   6.187500 230.721875 146.687500   3.596563   3.217250  17.848750
       vs         am       gear       carb
 0.437500   0.406250   3.687500   2.812500
```

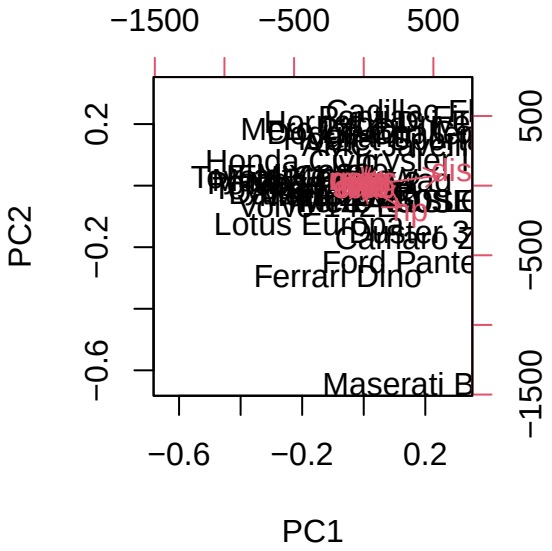
```
apply(mtcars, 2, sd)
```

```
       mpg         cyl        disp          hp        drat          wt
 6.0269481   1.7859216 123.9386938  68.5628685   0.5346787   0.9784574
      qsec          vs          am        gear        carb
 1.7869432   0.5040161   0.4989909   0.7378041   1.6152000
```

It is clear "disp" and "hp" have the highest mean values and the highest standard deviation here. They will dominate any analysis I do on this dataset. Let's see.

```
pc.noscale <- prcomp(mtcars, scale = FALSE)
pc.scale <- prcomp(mtcars, scale = TRUE)
```

```
biplot(pc.noscale)
```




```
pc.noscale$rotation[,1]
```

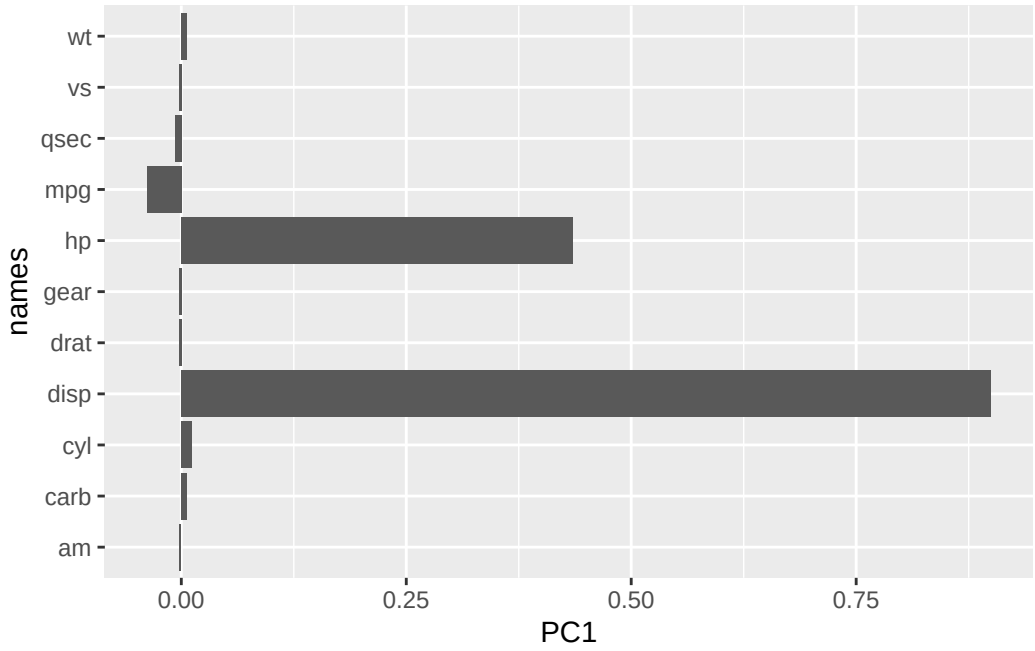
```
         mpg          cyl         disp           hp         drat           wt
-0.038118199  0.012035150  0.899568146  0.434784387 -0.002660077  0.006239405
        qsec           vs           am         gear         carb
-0.006671270 -0.002729474 -0.001962644 -0.002604768  0.005766010
```

plot the loading's

```
library(ggplot2)

r1 <- as.data.frame(pc.noscale$rotation)
r1$names <- rownames(pc.noscale$rotation)
```

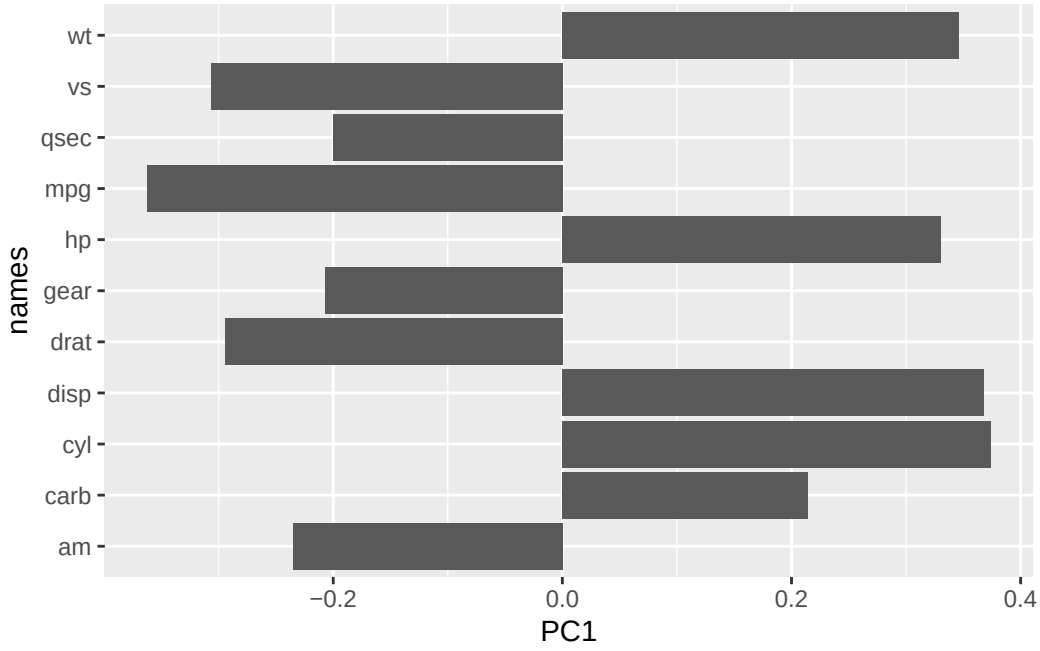
```
ggplot(r1) +
  aes(PC1, names) +
  geom_col()
```



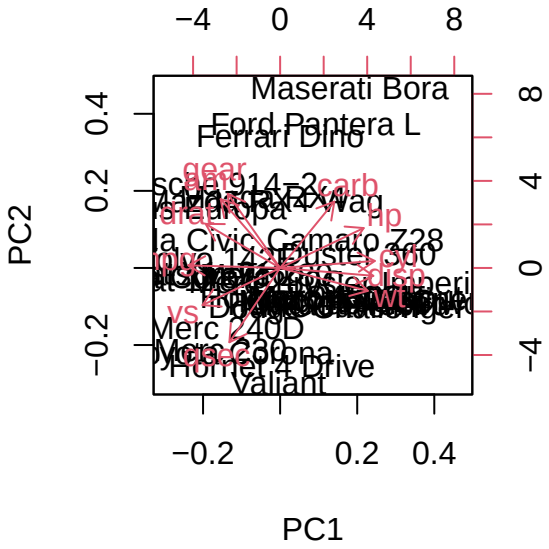

```
library(ggplot2)

r2 <- as.data.frame(pc.scale$rotation)
r2$names <- rownames(pc.scale$rotation)

ggplot(r2) +
  aes(PC1, names) +
  geom_col()
```




```
biplot(pc.scale)
```



Take Home: Generally we always want to set `scale=TRUE` when we do this type

of analysis to avoid our analysis being dominated by individual variables with the largest variance just due to their unit of measurment.

FNA breast cancer data

load the data into R

```
wisc.df <- read.csv("WisconsinCancer.csv")

head(wisc.df)
```

```
        id diagnosis radius_mean texture_mean perimeter_mean area_mean
1   842302         M       17.99        10.38         122.80    1001.0
2   842517         M       20.57        17.77         132.90    1326.0
3 84300903         M       19.69        21.25         130.00    1203.0
4 84348301         M       11.42        20.38          77.58     386.1
5 84358402         M       20.29        14.34         135.10    1297.0
6   843786         M       12.45        15.70          82.57     477.1
  smoothness_mean compactness_mean concavity_mean concave.points_mean
1         0.11840          0.27760         0.3001             0.14710
2         0.08474          0.07864         0.0869             0.07017
3         0.10960          0.15990         0.1974             0.12790
4         0.14250          0.28390         0.2414             0.10520
5         0.10030          0.13280         0.1980             0.10430
6         0.12780          0.17000         0.1578             0.08089
  symmetry_mean fractal_dimension_mean radius_se texture_se perimeter_se
1        0.2419                0.07871    1.0950     0.9053        8.589
2        0.1812                0.05667    0.5435     0.7339        3.398
3        0.2069                0.05999    0.7456     0.7869        4.585
4        0.2597                0.09744    0.4956     1.1560        3.445
5        0.1809                0.05883    0.7572     0.7813        5.438
6        0.2087                0.07613    0.3345     0.8902        2.217
  area_se smoothness_se compactness_se concavity_se concave.points_se
1  153.40      0.006399        0.04904      0.05373           0.01587
2   74.08      0.005225        0.01308      0.01860           0.01340
3   94.03      0.006150        0.04006      0.03832           0.02058
4   27.23      0.009110        0.07458      0.05661           0.01867
5   94.44      0.011490        0.02461      0.05688           0.01885
6   27.19      0.007510        0.03345      0.03672           0.01137
  symmetry_se fractal_dimension_se radius_worst texture_worst perimeter_worst
1     0.03003             0.006193        25.38         17.33          184.60
2     0.01389             0.003532        24.99         23.41          158.80
3     0.02250             0.004571        23.57         25.53          152.50
```

```
4     0.05963             0.009208        14.91         26.50           98.87
5     0.01756             0.005115        22.54         16.67          152.20
6     0.02165             0.005082        15.47         23.75          103.40
  area_worst smoothness_worst compactness_worst concavity_worst
1     2019.0           0.1622            0.6656          0.7119
2     1956.0           0.1238            0.1866          0.2416
3     1709.0           0.1444            0.4245          0.4504
4      567.7           0.2098            0.8663          0.6869
5     1575.0           0.1374            0.2050          0.4000
6      741.6           0.1791            0.5249          0.5355
  concave.points_worst symmetry_worst fractal_dimension_worst
1               0.2654         0.4601                 0.11890
2               0.1860         0.2750                 0.08902
3               0.2430         0.3613                 0.08758
4               0.2575         0.6638                 0.17300
5               0.1625         0.2364                 0.07678
6               0.1741         0.3985                 0.12440
```

Q1. How many observations are in this dataset?

```
nrow(wisc.df)
```

```
[1] 569
```

Q2. How many of the observations have a malignant diagnosis?

```
sum(wisc.df$diagnosis =="M")
```

```
[1] 212
```

The `table()` function is super useful here

```
table(wisc.df$diagnosis)
```

```
  B   M
357 212
```

Q3. How many variables/features in the data are suffixed with _mean?

```
ncol(wisc.df)
```

```
[1] 32
```

```
colnames(wisc.df)
```

```
 [1] "id"                    "diagnosis"
 [3] "radius_mean"           "texture_mean"
 [5] "perimeter_mean"        "area_mean"
 [7] "smoothness_mean"       "compactness_mean"
 [9] "concavity_mean"        "concave.points_mean"
[11] "symmetry_mean"         "fractal_dimension_mean"
[13] "radius_se"             "texture_se"
[15] "perimeter_se"          "area_se"
[17] "smoothness_se"         "compactness_se"
[19] "concavity_se"          "concave.points_se"
[21] "symmetry_se"           "fractal_dimension_se"
[23] "radius_worst"          "texture_worst"
[25] "perimeter_worst"       "area_worst"
[27] "smoothness_worst"      "compactness_worst"
[29] "concavity_worst"       "concave.points_worst"
[31] "symmetry_worst"        "fractal_dimension_worst"
```

A useful function for this is `grep()`

```
grep("_mean", colnames(wisc.df))
```

```
 [1]  3  4  5  6  7  8  9 10 11 12
```

```
length(grep("_mean", colnames(wisc.df)))
```

```
[1] 10
```

Before we go any further we need to exclude the diagnosis column from any future analysis - this tells us whether a sample to cancer or non-cancer.

```
diagnosis <- as.factor(wisc.df$diagnosis)
head(diagnosis)
```

```
[1] M M M M M M
Levels: B M
```

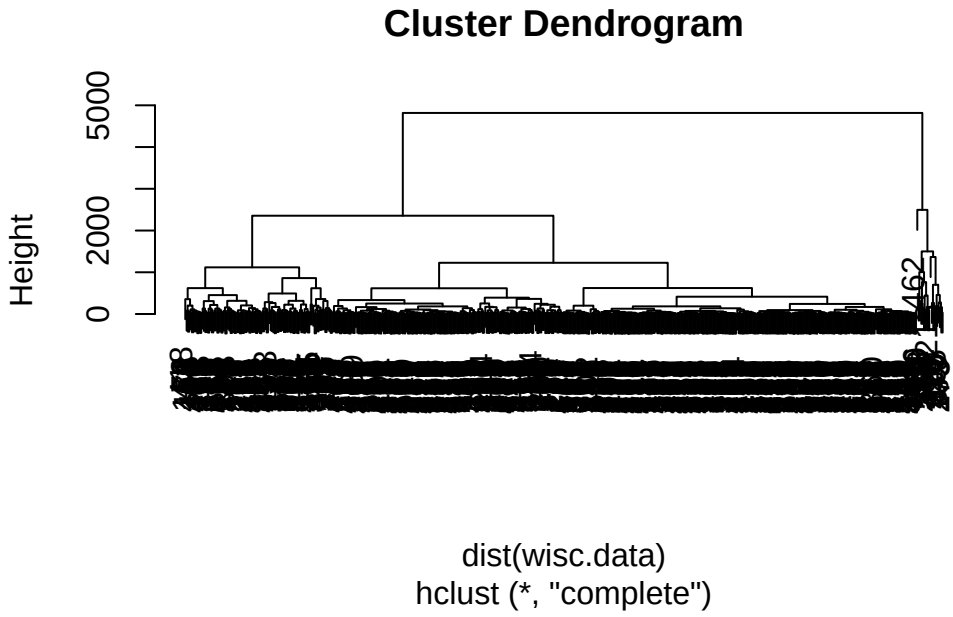
```
wisc.data <- wisc.df[,-1]
```

Lets see if we can cluster the `wisc.data` to find some structure in the dataset.

```
hc <- hclust(dist(wisc.data))
```

```
Warning in dist(wisc.data): NAs introduced by coercion
```

```
plot(hc)
```



Principle Component Analysis (PCA)

```
wisc.data <- wisc.data[sapply(wisc.data, is.numeric)]
```

```
wisc.pr <- prcomp(wisc.data, scale = TRUE)
summary(wisc.pr)
```

```
Importance of components:
                          PC1    PC2     PC3     PC4     PC5     PC6     PC7
Standard deviation     3.6444 2.3857 1.67867 1.40735 1.28403 1.09880 0.82172
Proportion of Variance 0.4427 0.1897 0.09393 0.06602 0.05496 0.04025 0.02251
Cumulative Proportion  0.4427 0.6324 0.72636 0.79239 0.84734 0.88759 0.91010
                           PC8    PC9    PC10   PC11    PC12    PC13    PC14
Standard deviation     0.69037 0.6457 0.59219 0.5421 0.51104 0.49128 0.39624
Proportion of Variance 0.01589 0.0139 0.01169 0.0098 0.00871 0.00805 0.00523
Cumulative Proportion  0.92598 0.9399 0.95157 0.9614 0.97007 0.97812 0.98335
                          PC15    PC16    PC17    PC18    PC19    PC20   PC21
Standard deviation     0.30681 0.28260 0.24372 0.22939 0.22244 0.17652 0.1731
Proportion of Variance 0.00314 0.00266 0.00198 0.00175 0.00165 0.00104 0.0010
Cumulative Proportion  0.98649 0.98915 0.99113 0.99288 0.99453 0.99557 0.9966
                          PC22    PC23   PC24    PC25    PC26    PC27    PC28
Standard deviation     0.16565 0.15602 0.1344 0.12442 0.09043 0.08307 0.03987
Proportion of Variance 0.00091 0.00081 0.0006 0.00052 0.00027 0.00023 0.00005
Cumulative Proportion  0.99749 0.99830 0.9989 0.99942 0.99969 0.99992 0.99997
                          PC29    PC30
Standard deviation     0.02736 0.01153
Proportion of Variance 0.00002 0.00000
Cumulative Proportion  1.00000 1.00000
```

> Q4. From your results, what proportion of the original variance is captured by the first principal components (PC1)?

The proportion of variance that is captured is 0.4427 > Q5. How many principal components (PCs) are required to describe at least 70% of the original variance in the data?

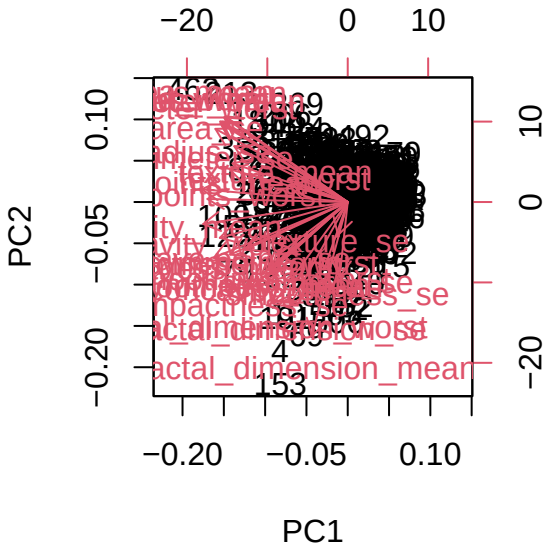
PC 1-3 is required to describe at least 70% of the orignal variance data

> Q6. How many principal components (PCs) are required to describe at least 90% of the original variance in the data?

PC 1-7 describes at least 90% of the orgiinal variance

Plot this data

```
biplot(wisc.pr)
```



Q7. What stands out to you about this plot? Is it easy or difficult to understand? Why?

The first thing that I notice about this plot is how clustered together everything is. This makes the plot extremely hard to understand and can not be read accurately to retrieve data from it.

This biplot sucks! we need to build our own PCA score plot of PC1 vs PC2

```
attributes(wisc.pr)
```

```
$names
[1] "sdev"     "rotation" "center"   "scale"    "x"

$class
[1] "prcomp"
```

```
head(wisc.pr$x)
```

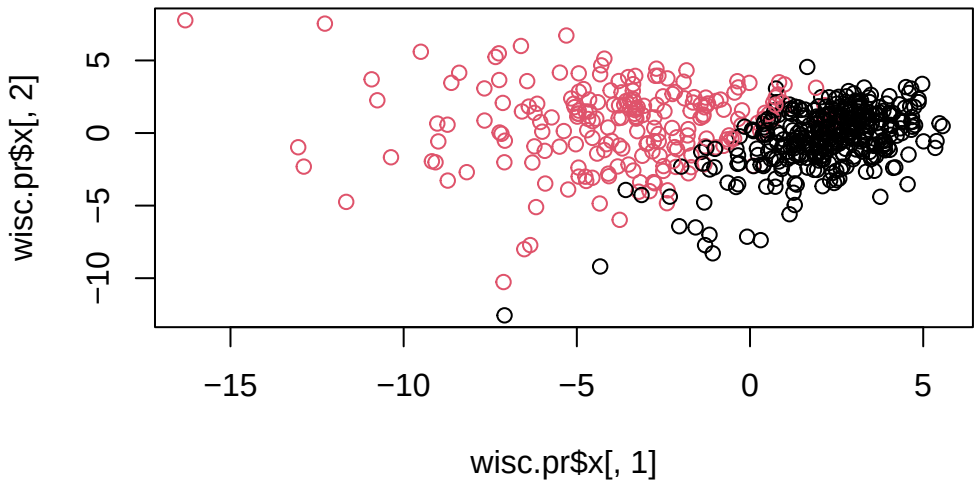
```
           PC1        PC2        PC3        PC4        PC5         PC6
[1,] -9.184755  -1.946870 -1.1221788 3.6305364  1.1940595  1.41018364
[2,] -2.385703   3.764859 -0.5288274 1.1172808 -0.6212284  0.02863116
```

```
[3,] -5.728855   1.074229 -0.5512625 0.9112808  0.1769302  0.54097615
[4,] -7.116691 -10.266556 -3.2299475 0.1524129  2.9582754  3.05073750
[5,] -3.931842   1.946359  1.3885450 2.9380542 -0.5462667 -1.22541641
[6,] -2.378155  -3.946456 -2.9322967 0.9402096  1.0551135 -0.45064213
            PC7         PC8         PC9       PC10       PC11       PC12
[1,]  2.15747152  0.39805698 -0.15698023 -0.8766305 -0.2627243 -0.8582593
[2,]  0.01334635 -0.24077660 -0.71127897  1.1060218 -0.8124048  0.1577838
[3,] -0.66757908 -0.09728813  0.02404449  0.4538760  0.6050715  0.1242777
[4,]  1.42865363 -1.05863376 -1.40420412 -1.1159933  1.1505012  1.0104267
[5,] -0.93538950 -0.63581661 -0.26357355  0.3773724 -0.6507870 -0.1104183
[6,]  0.49001396  0.16529843 -0.13335576 -0.5299649 -0.1096698  0.0813699
           PC13         PC14         PC15        PC16        PC17        PC18
[1,]  0.10329677 -0.690196797  0.601264078  0.74446075 -0.26523740 -0.54907956
[2,] -0.94269981 -0.652900844 -0.008966977 -0.64823831 -0.01719707  0.31801756
[3,] -0.41026561  0.016665095 -0.482994760  0.32482472  0.19075064 -0.08789759
[4,] -0.93245070 -0.486988399  0.168699395  0.05132509  0.48220960 -0.03584323
[5,]  0.38760691 -0.538706543 -0.310046684 -0.15247165  0.13302526 -0.01869779
[6,] -0.02625135  0.003133944 -0.178447576 -0.01270566  0.19671335 -0.29727706
          PC19        PC20         PC21        PC22        PC23         PC24
[1,]  0.1336499  0.34526111  0.096430045 -0.06878939  0.08444429  0.175102213
[2,] -0.2473470 -0.11403274 -0.077259494  0.09449530 -0.21752666 -0.011280193
[3,] -0.3922812 -0.20435242  0.310793246  0.06025601 -0.07422581 -0.102671419
[4,] -0.0267241 -0.46432511  0.433811661  0.20308706 -0.12399554 -0.153294780
[5,]  0.4610302  0.06543782 -0.116442469  0.01763433  0.13933105  0.005327110
[6,] -0.1297265 -0.07117453 -0.002400178  0.10108043  0.03344819 -0.002837749
            PC25         PC26        PC27          PC28         PC29
[1,]  0.150887294 -0.201326305 -0.25236294 -0.0338846387  0.045607590
[2,]  0.170360355 -0.041092627  0.18111081  0.0325955021 -0.005682424
[3,] -0.171007656  0.004731249  0.04952586  0.0469844833  0.003143131
[4,] -0.077427574 -0.274982822  0.18330078  0.0424469831 -0.069233868
[5,] -0.003059371  0.039219780  0.03213957 -0.0347556386  0.005033481
[6,] -0.122282765 -0.030272333 -0.08438081  0.0007296587 -0.019703996
             PC30
[1,]  0.0471277407
[2,]  0.0018662342
[3,] -0.0007498749
[4,]  0.0199198881
[5,] -0.0211951203
[6,] -0.0034564331
```

Plot of PC1 vs PC2 the first two columns >Q8. Generate a similar plot for principal components 1 and 3. What do you notice about these plots?

Immediately after creating this plot we are able to see that it is much more clear and easier to understand the data from it. We see the difference between the red(malignant), and the black(benign)

```
plot(wisc.pr$x[,1], wisc.pr$x[,2], col=diagnosis)
```



Make a ggplot version of this score plot

Data frame

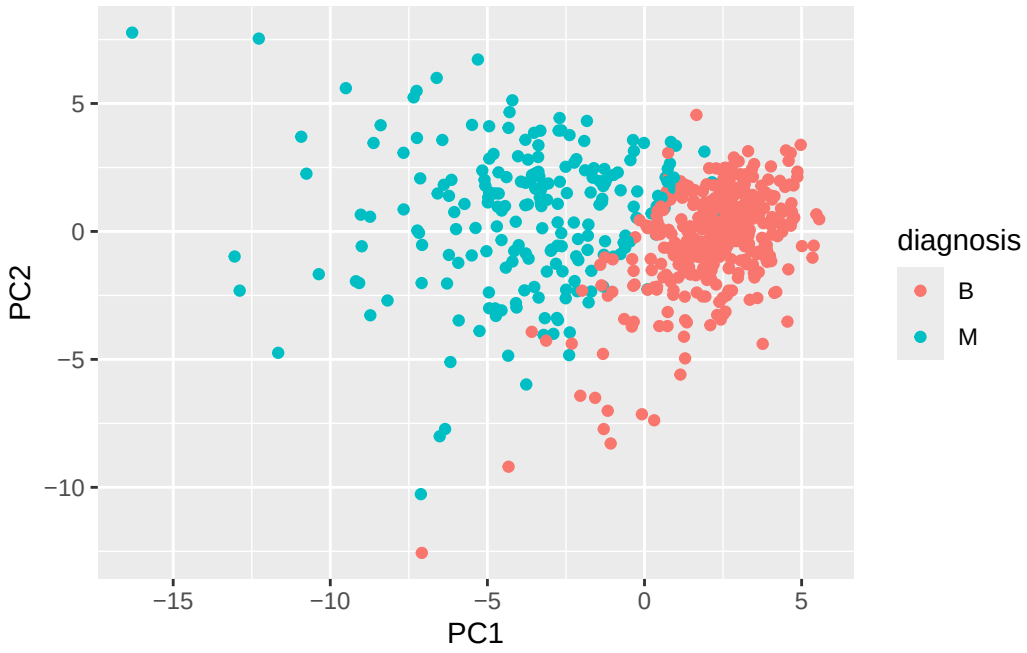
```
df <- as.data.frame(wisc.pr$x)
df$diagnosis <- diagnosis
```

Load ggplot

```
library(ggplot2)
```

Make a scatter plot

```
ggplot(df) +
  aes(PC1, PC2, col=diagnosis) +
  geom_point()
```



Variance Explained

We want to find out if there is an elbow in the amount of variance, to calculate this we will write some code.

Variance of each component

```
pr.var <- wisc.pr$sdev^2
head(pr.var)
```

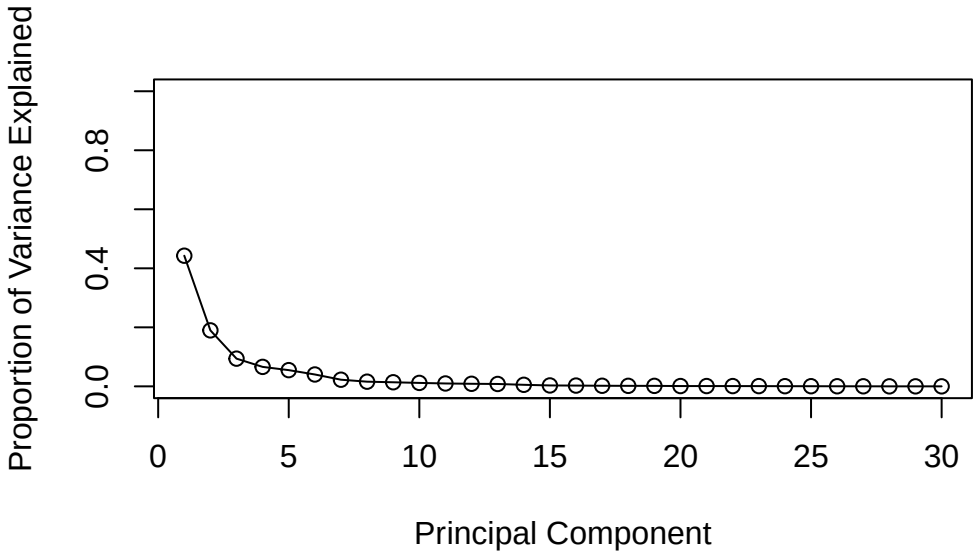
```
[1] 13.281608  5.691355  2.817949  1.980640  1.648731  1.207357
```

Variance by principal component

```
pve <- (pr.var)/ (sum(pr.var))
```

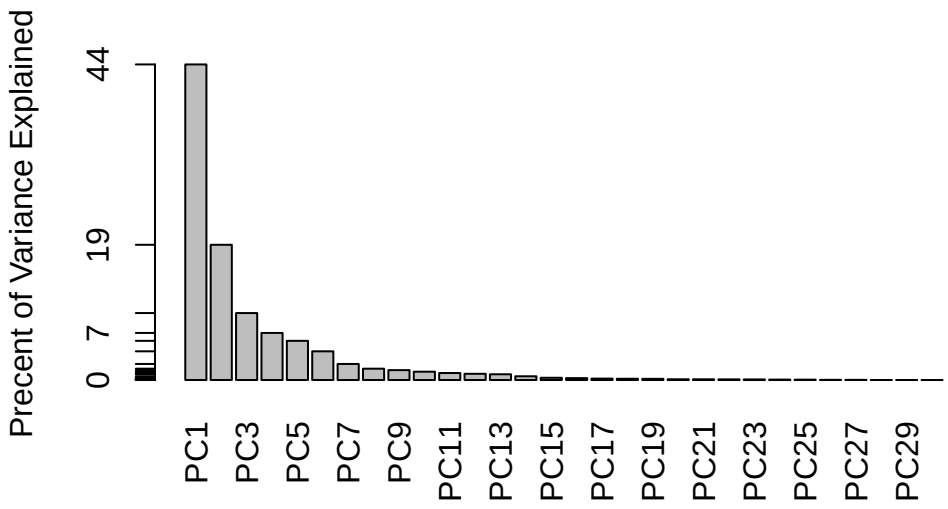
Plotting variance explained for each principle component

```
plot(pve, xlab = "Principal Component",
     ylab = "Proportion of Variance Explained",
     ylim = c(0, 1), type = "o")
```



Alternative scree plot of the same data, note data driven y-axis

```
barplot(pve, ylab = "Precent of Variance Explained",
     names.arg=paste0("PC",1:length(pve)), las=2, axes = FALSE)
axis(2, at=pve, labels=round(pve,2)*100 )
```

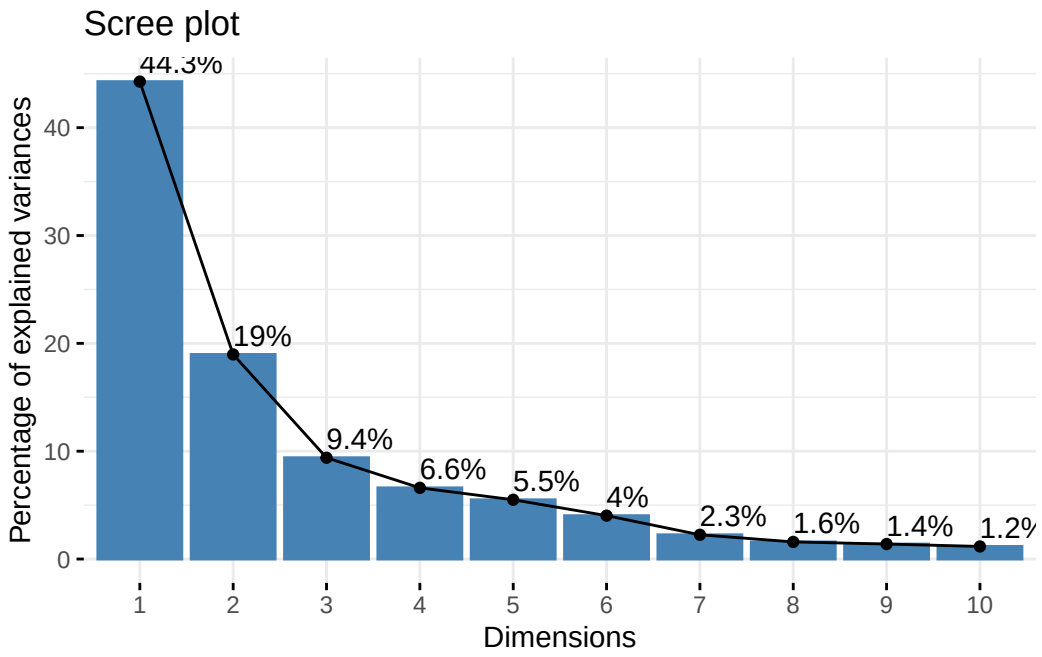


GGplot example of this graph

```
## ggplot based graph
#install.packages("factoextra")
library(factoextra)
```

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

```
fviz_eig(wisc.pr, addlabels = TRUE)
```



Q9. For the first principal component, what is the component of the loading vector (i.e. wisc.pr$rotation[,1]) for the feature concave.points_mean?

```
wisc.pr$rotation["concave.points_mean", 1]
```

```
[1] -0.2608538
```

Q10. What is the minimum number of principal components required to explain 80% of the variance of the data?

The miniimum number of principle components is 5

##Hierarchical Clustering

```
data.scaled <- scale(wisc.data)
```

Calculate the (Euclidean) distances

```
data.dist <- dist(data.scaled)
```

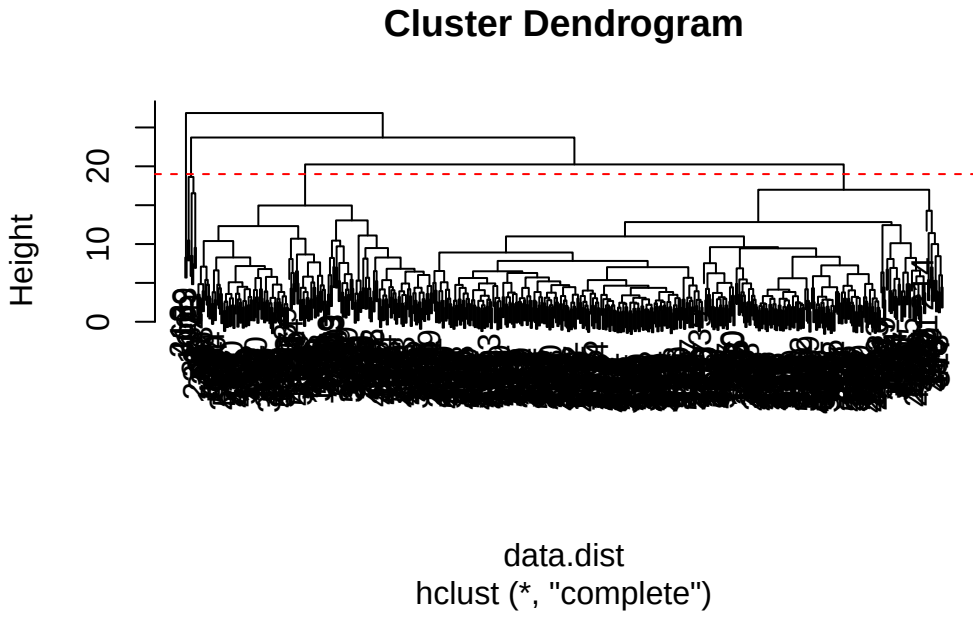
hierarchical clustering model using complete linkage. Manually specify the method argument to hclust() and assign the results to wisc.hclust.

```
wisc.hclust <- hclust(data.dist, method="complete")
```

Q11. Using the plot() and abline() functions, what is the height at which the clustering model has 4 clusters?

At 19 the clustering tree has 4 clusters

```
plot(wisc.hclust)
abline(h=19, col="red", lty=2)
```




```
wisc.hclust.clusters <- cutree(wisc.hclust, (k=4))
```

We can use the table() function to compare the cluster membership to the actual diagnoses.

```
table(wisc.hclust.clusters, diagnosis)
```

```
                    diagnosis
wisc.hclust.clusters   B   M
                   1  12 165
                   2   2   5
                   3 343  40
                   4   0   2
```

Q12. Can you find a better cluster vs diagnoses match by cutting into a different number of clusters between 2 and 10?

Changing the number of clusters (k) can improve how well they match the actual diagnoses. Using k=2 works well since there are two main groups (malignant and benign), but k=3 or 4 might show more details. Testing different k values helps find the best match.

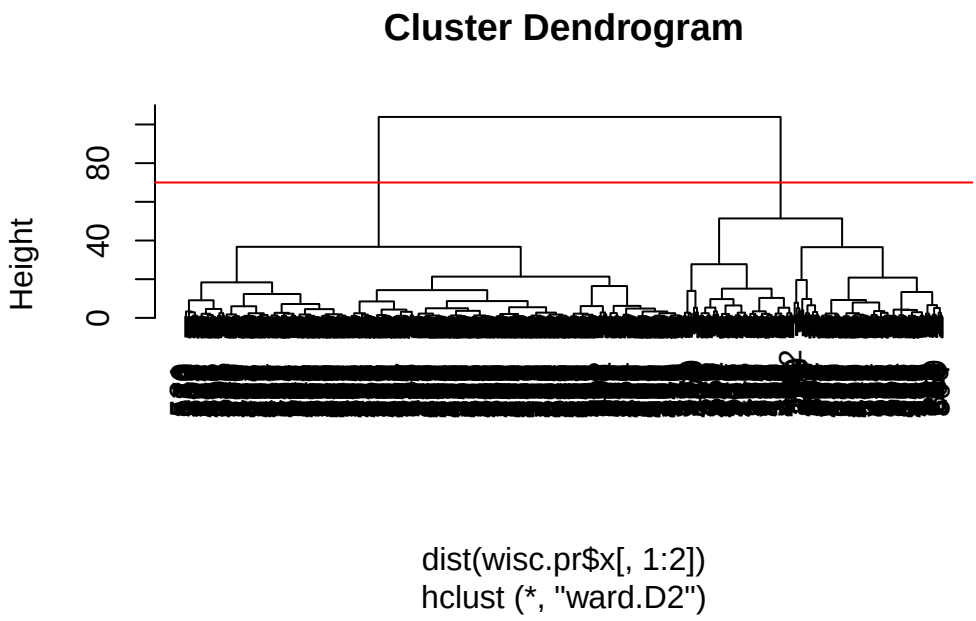
Q13. Which method gives your favorite results for the same data.dist dataset? Explain your reasoning.

The best method depends on how well the clusters match the actual diagnoses. The "complete" method is good at keeping similar cases together, but "ward.D2" can create more balanced clusters. Testing different methods helps find the best one.

Clustering in PC space

```
hc <- hclust(dist(wisc.pr$x[,1:2]), method="ward.D2")

plot(hc)
abline(h=70, col="red")
```



Cluster membership vector

```
grps <- cutree(hc, h=70)
table(grps)
```

```
grps
  1   2
195 374
```

```
table(diagnosis)
```

```
diagnosis
  B   M
357 212
```

Cross table to see how my clustering groups correspond to the expert diagnosis vector of M and B values.

```
table(grps, diagnosis)
```

```
    diagnosis
grps   B   M
   1  18 177
   2 339  35
```

Positive would => cancer M Negative => non-cancer B

True = cluster/grp 1 False = grp 2

True Positive 177 False Positive 18 True Negative 339 False Negative 35

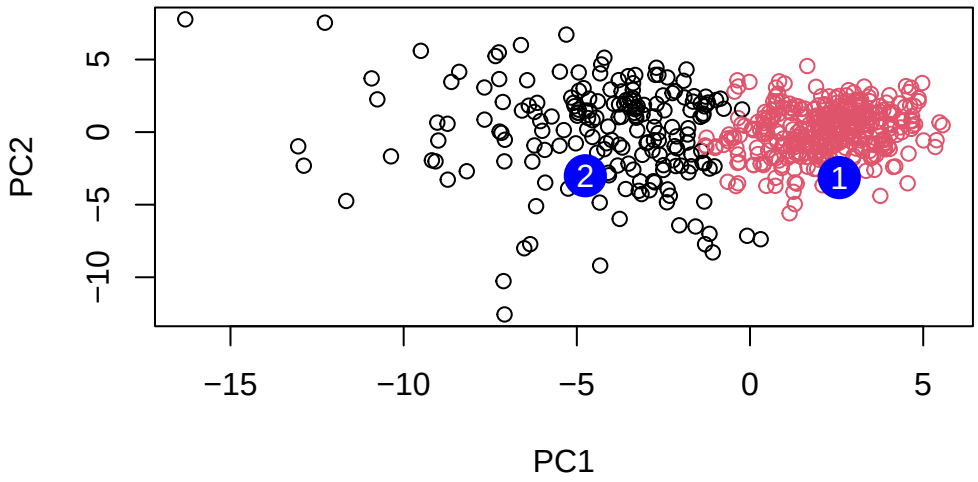
We can use our PCA results (wis.pr) to make prediections on new unseen data.

```
url <- "https://tinyurl.com/new-samples-CSV"
new <- read.csv(url)
npc <- predict(wisc.pr, newdata=new)
npc
```

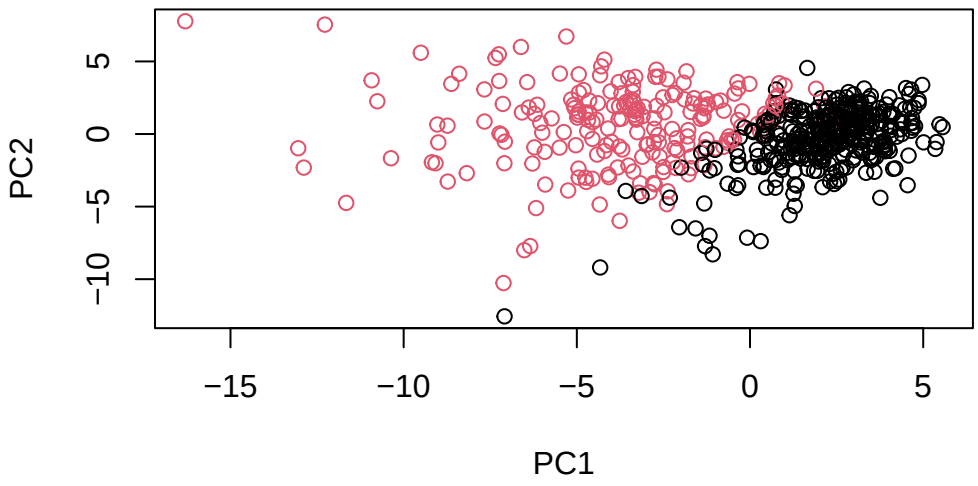
```
          PC1       PC2        PC3        PC4       PC5        PC6        PC7
[1,]  2.576616 -3.135913  1.3990492 -0.7631950  2.781648 -0.8150185 -0.3959098
[2,] -4.754928 -3.009033 -0.1660946 -0.6052952 -1.140698 -1.2189945  0.8193031
           PC8       PC9       PC10      PC11      PC12      PC13     PC14
```

```
[1,] -0.2307350 0.1029569 -0.9272861 0.3411457  0.375921 0.1610764 1.187882
[2,] -0.3307423 0.5281896 -0.4855301 0.7173233 -1.185917 0.5893856 0.303029
          PC15       PC16        PC17        PC18        PC19       PC20
[1,] 0.3216974 -0.1743616 -0.07875393 -0.11207028 -0.08802955 -0.2495216
[2,] 0.1299153  0.1448061 -0.40509706  0.06565549  0.25591230 -0.4289500
           PC21       PC22       PC23       PC24        PC25          PC26
[1,]  0.1228233 0.09358453 0.08347651  0.1223396  0.02124121  0.078884581
[2,] -0.1224776 0.01732146 0.06316631 -0.2338618 -0.20755948 -0.009833238
             PC27        PC28          PC29         PC30
[1,]  0.220199544 -0.02946023 -0.015620933  0.005269029
[2,] -0.001134152  0.09638361  0.002795349 -0.019015820
```

```
plot(wisc.pr$x[,1:2], col=grps)
points(npc[,1], npc[,2], col="blue", pch=16, cex=3)
text(npc[,1], npc[,2], c(1,2), col="white")
```




```
plot(wisc.pr$x[,1:2], col=diagnosis)
```




```
g <- as.factor(grps)
levels(g)
```

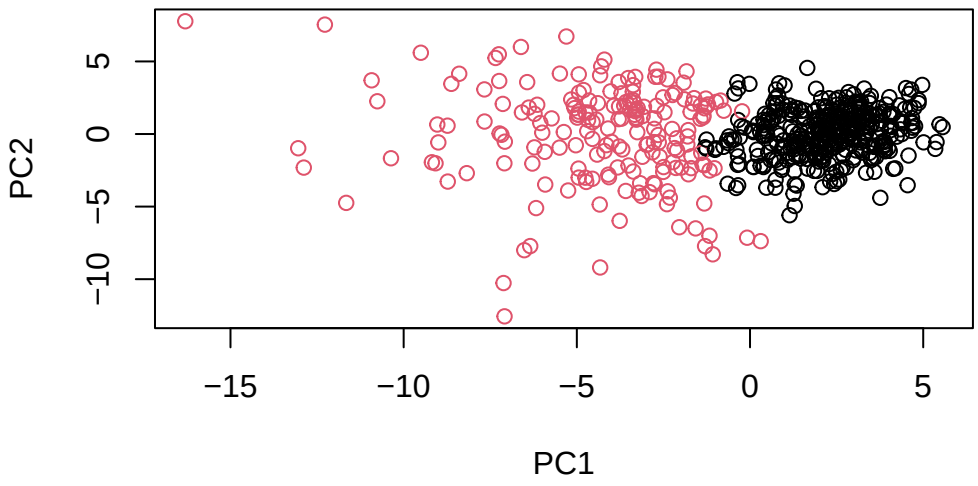
```
[1] "1" "2"
```

```
g <- relevel(g,2)
levels(g)
```

```
[1] "2" "1"
```

Plot using our re-ordered factor

```
plot(wisc.pr$x[,1:2], col=g)
```



Q15. How well does the newly created model with four clusters separate out the two diagnoses?

We checked how well the four clusters match the diagnoses. The clusters separate malignant and benign samples fairly well, but it's not perfect. We can look at the table for true positives and true negatives.

diagnosis

wisc.pr.hclust.clusters B M

1 28 188

2 329 24

Q16. How well do the k-means and hierarchical clustering models you created in previous sections (i.e. before PCA) do in terms of separating the diagnoses? Again, use the table() function to compare the output of each model (wisc.km$cluster and wisc.hclust.clusters) with the vector containing the actual diagnoses.

The k-means and hierarchical clustering models both show some success in separating the diagnoses, but neither perfectly matches the true labels. By examining the tables, we can see how each model's clusters align with benign (B) and malignant (M) diagnoses.

diagnosis

B M

1 14 175

2 343 37

diagnosis

wisc.hclust.clusters B M

1 12 165

2 2 5

3 343 40

4 0 2

Sensitivty/Specifity

Sensitivity refers to a test's ability to correctly detect ill patients who do have the condition. In our example here the sensitivity is the total number of samples in the cluster identified as predominantly malignant (cancerous) divided by the total number of known malignant samples. In other words: TP/(TP+FN).

Specificity relates to a test's ability to correctly reject healthy patients without a condition. In our example specificity is the proportion of benign (not cancerous) samples in the cluster identified as predominantly benign that are known to be benign. In other words: TN/(TN+FN).

> Q17. Which of your analysis procedures resulted in a clustering model with the best specificity? How about sensitivity?

We compare the clusters to the actual diagnoses. The clustering method with the highest specificity correctly identifies benign samples, while the one with the highest sensitivity better detects malignant samples.

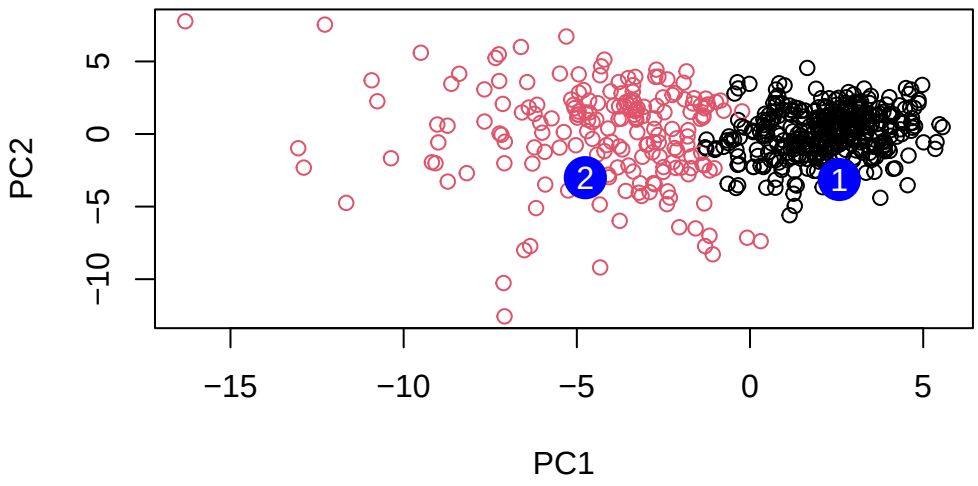
Prediction

We will use the predict() function that will take our PCA model from before and new cancer cell data and project that data onto our PCA space.

```
#url <- "new_samples.csv"
url <- "https://tinyurl.com/new-samples-CSV"
new <- read.csv(url)
npc <- predict(wisc.pr, newdata=new)
npc
```

```
          PC1       PC2        PC3        PC4       PC5        PC6        PC7
[1,]  2.576616 -3.135913  1.3990492 -0.7631950  2.781648 -0.8150185 -0.3959098
[2,] -4.754928 -3.009033 -0.1660946 -0.6052952 -1.140698 -1.2189945  0.8193031
           PC8       PC9       PC10      PC11      PC12      PC13     PC14
[1,] -0.2307350 0.1029569 -0.9272861 0.3411457  0.375921 0.1610764 1.187882
[2,] -0.3307423 0.5281896 -0.4855301 0.7173233 -1.185917 0.5893856 0.303029
          PC15       PC16        PC17        PC18        PC19       PC20
[1,] 0.3216974 -0.1743616 -0.07875393 -0.11207028 -0.08802955 -0.2495216
[2,] 0.1299153  0.1448061 -0.40509706  0.06565549  0.25591230 -0.4289500
          PC21       PC22       PC23       PC24        PC25         PC26
[1,]  0.1228233 0.09358453 0.08347651  0.1223396  0.02124121  0.078884581
[2,] -0.1224776 0.01732146 0.06316631 -0.2338618 -0.20755948 -0.009833238
             PC27        PC28         PC29         PC30
[1,]  0.220199544 -0.02946023 -0.015620933  0.005269029
[2,] -0.001134152  0.09638361  0.002795349 -0.019015820
```

```
plot(wisc.pr$x[,1:2], col=g)
points(npc[,1], npc[,2], col="blue", pch=16, cex=3)
text(npc[,1], npc[,2], c(1,2), col="white")
```



Q18. Which of these new patients should we prioritize for follow up based on your results?

The ones closer to the malignant group (with higher PC1 and PC2 values in that direction) should be prioritized for follow-up. in our case it would be patient 1.