Worksheet 3 Group 1111

Ammara Akhtar, Afia Ibnath, and Reuben Walker

5 May 2024

###Exercise 1 Stratification

```
library(tidyverse)
```

Load libraries

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggplot2)
```

Data summary

```
lung_data <- read.csv("C:/Users/Reuben/Documents/Code/DSLS/lung_data_all.csv")
head(lung_data)
```

```
##   Subject.id Lung.function Trial.arm Sex
## 1          1          11.0   Control   M
## 2          2          11.1   Control   M
## 3          3           9.5   Control   F
## 4          4          10.1   Control   F
## 5          5           9.7   Control   F
## 6          6           9.4   Control   F
```

```
# Summary statistics
summary(lung_data)
```

```
##    Subject.id    Lung.function    Trial.arm            Sex
##  Min.   : 1.00   Min.   : 9.40   Length:40          Length:40
##  1st Qu.:10.75   1st Qu.:10.07   Class :character   Class :character
##  Median :20.50   Median :10.30   Mode  :character   Mode  :character
##  Mean   :20.50   Mean   :10.31
##  3rd Qu.:30.25   3rd Qu.:10.62
##  Max.   :40.00   Max.   :11.40
```

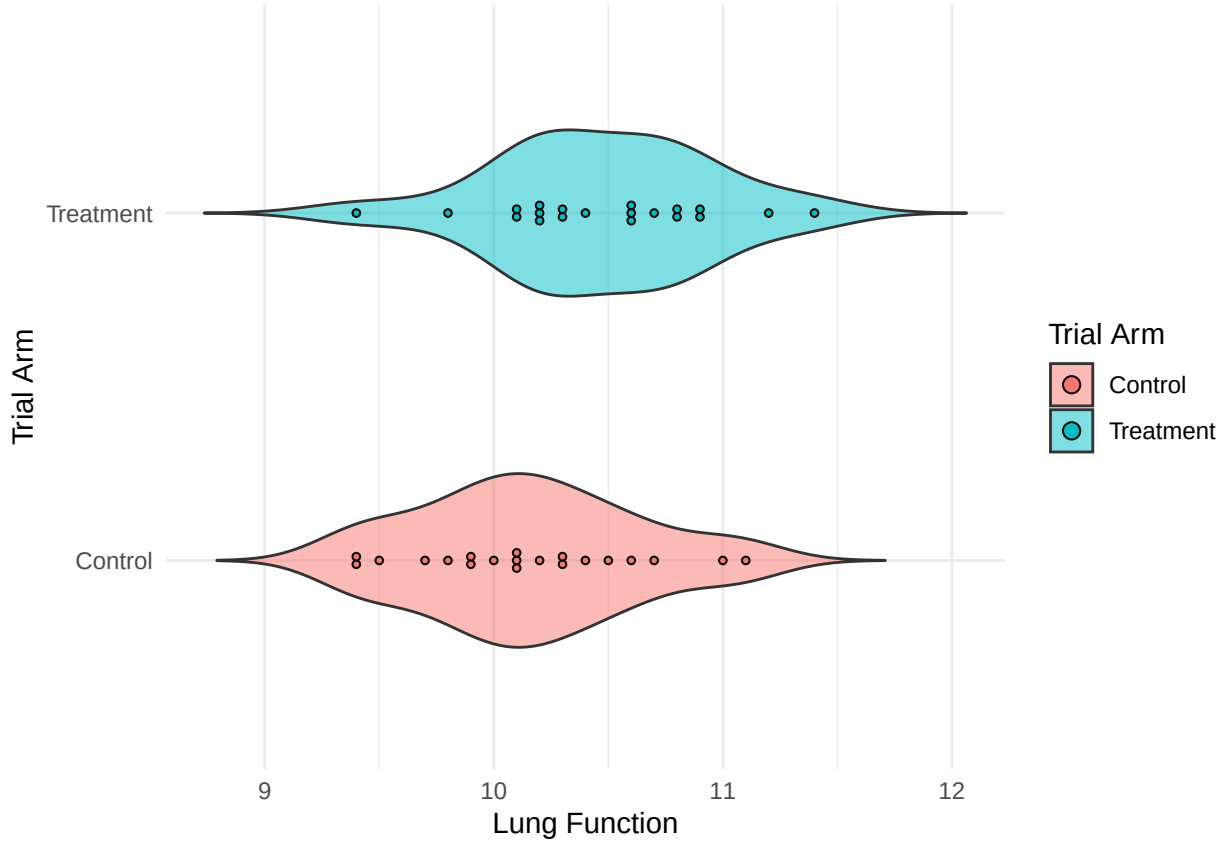
Test whether the treatment shows any effect overall

```
t.test(filter (lung_data, Trial.arm == "Treatment") $Lung.function,
filter (lung_data, Trial.arm == "Control") $Lung.function) $p.value
```

```
## [1] 0.0374047
```

```
fig1<- ggplot(lung_data, aes(x = Trial.arm, y = Lung.function, fill = Trial.arm)) +
  geom_violin(trim = FALSE, alpha = 0.5, width = 0.5) +
  geom_dotplot(binaxis = "y",
               stackdir = "center",
               dotsize = 0.5,
               aes(fill = Trial.arm)) +
  labs(x = "Trial Arm", y = "Lung Function", fill = "Trial Arm") +
  theme_minimal() +
  coord_flip()
fig1
```

```
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
```



A t-test comparing lung function between the treatment and control groups shows a significant difference (p = 0.0374),which is below the standard significance level of 0.05, suggesting that the treatment has an overall effect on lung function.

Analysis stratified by sex

```
t.test(filter (lung_data, Sex=='M', Trial.arm == "Treatment") $Lung.function,
filter (lung_data,  Sex=='M', Trial.arm == "Control") $Lung.function) $p.value
```

```
## [1] 0.1267617
```

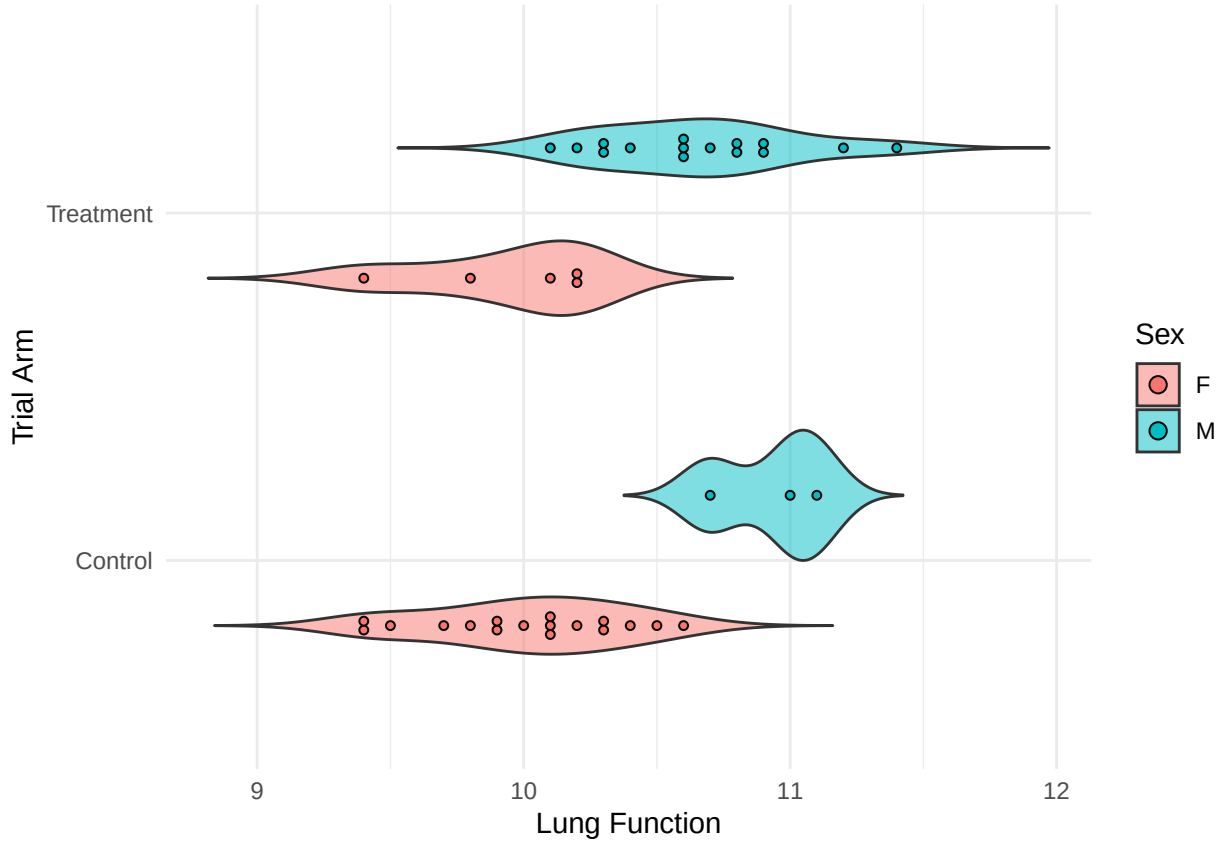
```
t.test(filter (lung_data, Sex=='F', Trial.arm == "Treatment") $Lung.function,
filter (lung_data,  Sex=='F', Trial.arm == "Control") $Lung.function) $p.value
```

```
## [1] 0.697984
```

```
fig2<- ggplot(lung_data, aes(x = Trial.arm, y = Lung.function, fill = Sex)) +
  geom_violin(trim = FALSE, alpha = 0.5, width = 0.75,position = position_dodge(0.75)) +
  geom_dotplot(binaxis = "y",
               stackdir = "center",
               dotsize = 0.5,
               aes(fill = Sex),
               position = position_dodge(0.75)) +  # Adjust position to overlay correctly
```

```
  labs(x = "Trial Arm", y = "Lung Function", fill = "Sex") +
  theme_minimal() +
  coord_flip()
fig2
```

```
## Bin width defaults to 1/30 of the range of the data. Pick better value with
## `binwidth`.
```



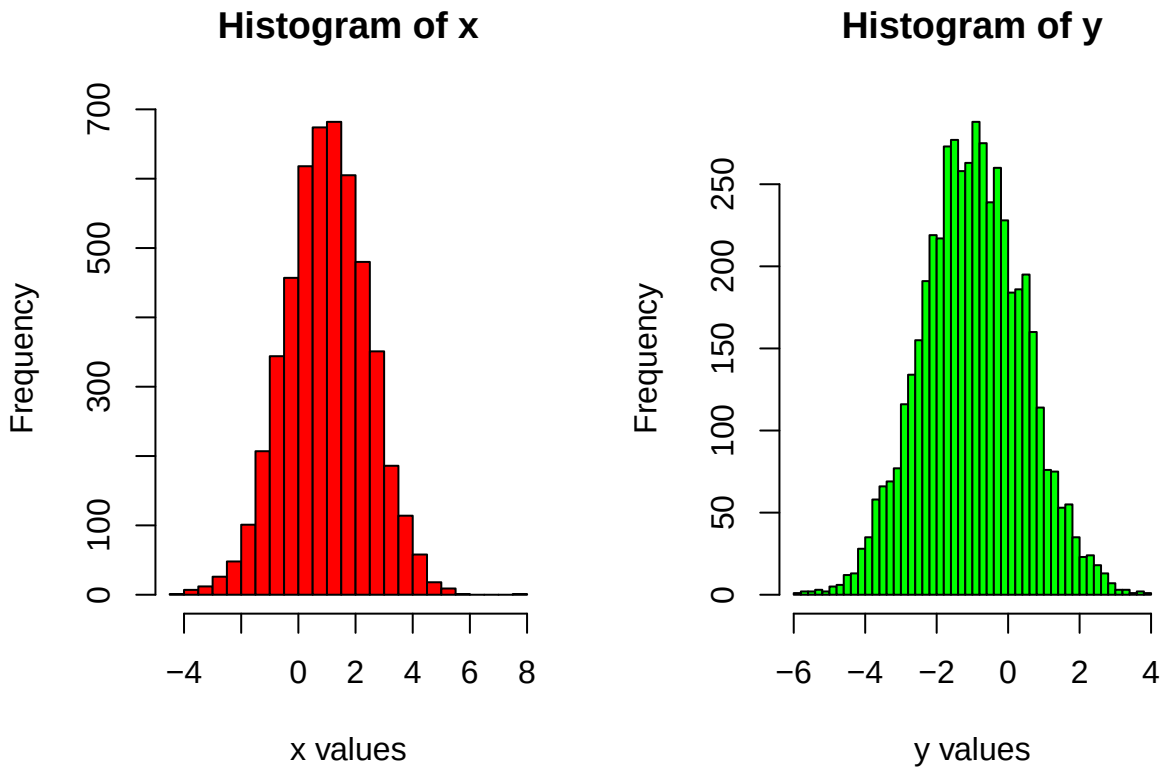
- Males: The t-test for males indicates no significant difference in lung function between the treatment and control groups ($p = 0.1268$).
- Females: Similarly, the t-test for females shows no significant difference in lung function between the treatment and control groups ($p = 0.698$).

Statistical Analysis

- While the overall analysis suggests a significant effect of the treatment on lung function, further stratified analysis by sex reveals non-significant differences within both male and female groups.
- These results imply that the observed treatment effect might be influenced by random sampling from both genders. There are more men in the treatment group and more women in the control. And men tend to have a higher baseline level of lung function.

###Exercise 2 Confounders #Distributation of x and y

```r
#command line
args <- commandArgs(trailingOnly = TRUE)
N <- as.integer(args[1])

N = 5000
if (is.na(N)) {
  stop("5000")
}
set.seed(57)

#Data
w <- 1 + rnorm(N)
x_0 <- rnorm(N)
y_0 <- rnorm(N)
x <- x_0 + w
y <- y_0 - w
#Ploting parameter of ! row and 2 coloums
par(mfrow=c(1,2))
#print(y)

#Plot the histogram of x
hist(x, main="Histogram of x", xlab="x values", ylab="Frequency", col="red", breaks=40)
#plot the histagram of y
hist(y, main="Histogram of y", xlab="y values", ylab="Frequency", col="green", breaks=40)
```



#Compute the t-test between x and y

```r
t_test_result <-t.test(x,y)
print(t_test_result)
```

```
##
##  Welch Two Sample t-test
##
## data:  x and y
## t = 69.463, df = 9997.3, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  1.923890 2.035626
## sample estimates:
##  mean of x  mean of y
##  0.9963053 -0.9834524
```

#Report The Welch Two Sample t-test between x anf y calculated the "t-statistic" is approximately 69.463. The "degrees of freedom" are approximately 9997.3. AS the p-values is very close to zero it indicates the strong evidance against the null hypothesis. The 95% confidence interval for the difference in means between x and y is approximately [1.924, 2.036]. The mean of x is approximately 0.996 and the mean of y is approximately -0.983. #Interpretation of findings Low p-value indicates that there is a statistically significant difference between the means of x and y. As the means of x is positive and highr than y. It indicates that, on average, the effect of adding w to x is greater than subtracting w from y. The practical significance of the material is determined by its context and the specific challenge at hand. A difference of about 2 units on some scales may or may not be significant, depending on the domain.

#generate two new vectors x`and y`

```r
#Data
w <- 1 + rnorm(N)
x_0 <- rnorm(N)
y_0 <- rnorm(N)

#Randomly decide to multiply w by either +1 or -1
signs <- sample(c(-1, 1), N, replace = TRUE)
w_p <- w * signs
# Generate xp and yp
x_p <- x_0 + w_p
y_p <- y_0 - w_p
#Display
summary(x_p)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -5.377754 -1.142570  0.006873  0.011656  1.179158  6.222975
```

```r
summary(y_p)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -6.17646 -1.16937  0.02416  0.01929  1.21674  5.67760
```

#Compute the t-test

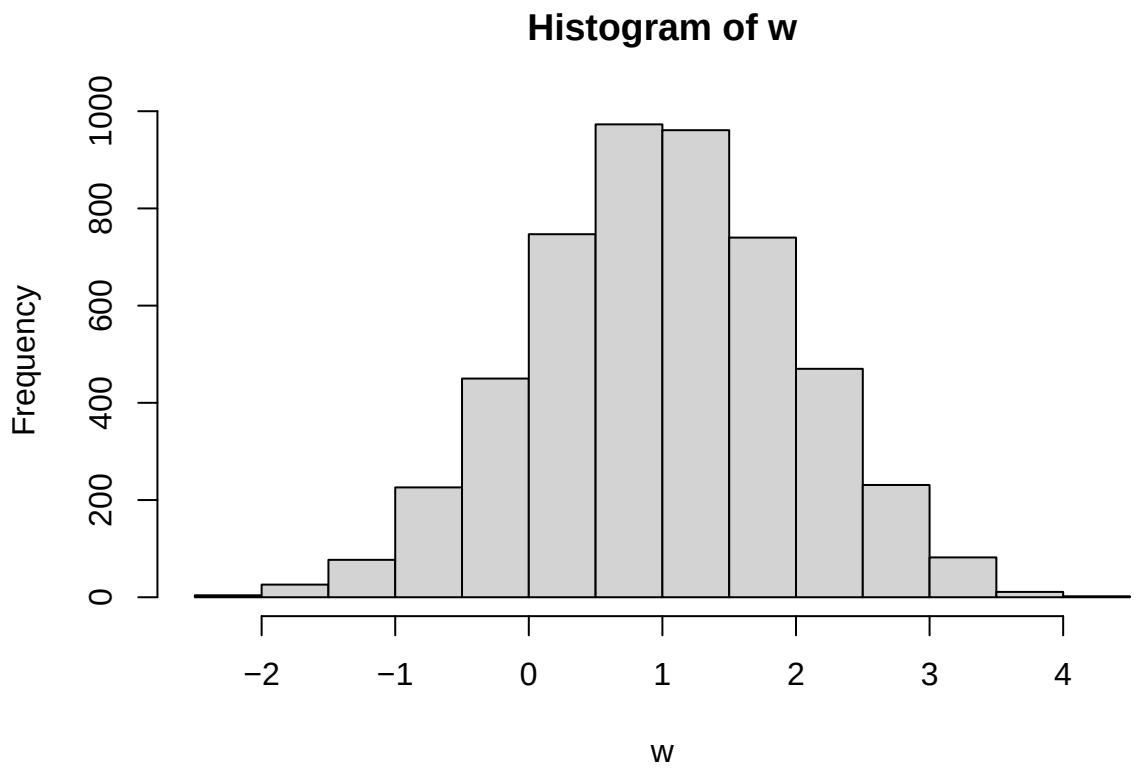
```r
t_test_result <-t.test(x_p, y_p)
print(t_test_result)
```

```
##
##  Welch Two Sample t-test
##
## data:  x_p and y_p
## t = -0.22457, df = 9997.2, p-value = 0.8223
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.07429094  0.05901837
## sample estimates:
##  mean of x  mean of y
## 0.01165608 0.01929237
```

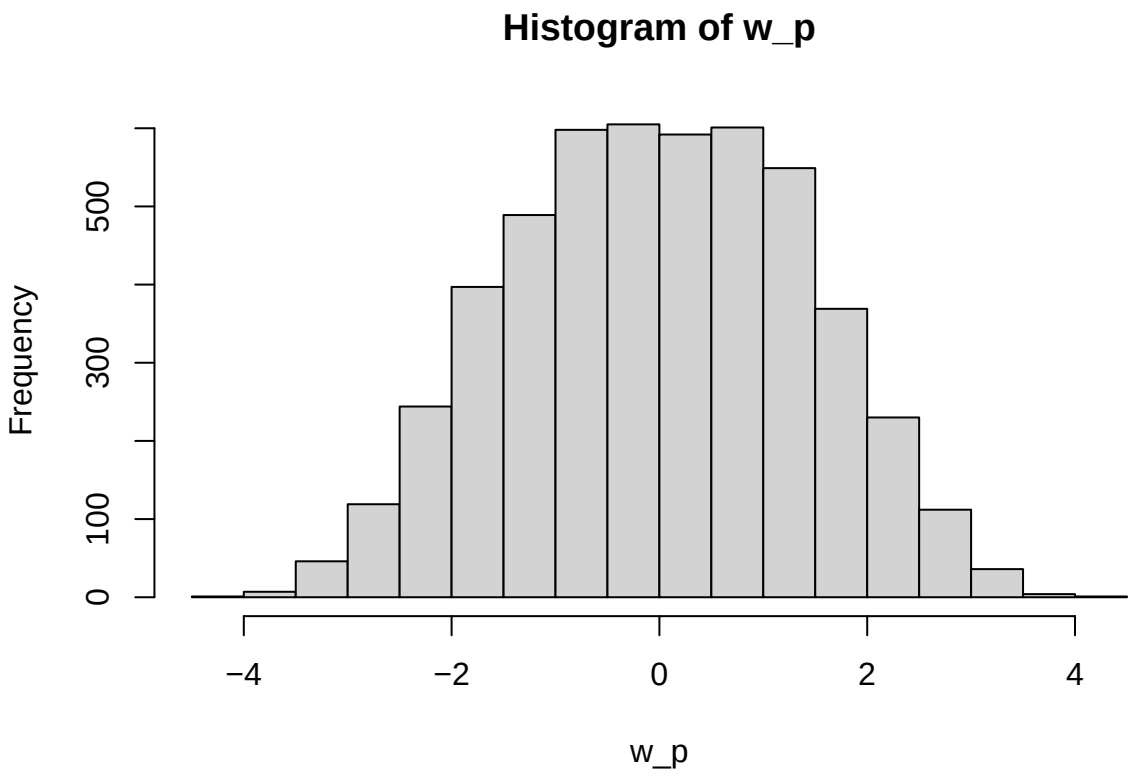
#Report The Welch Two Sample t-test between x anf y calculated the "t-statistic" is approximately -0.225. The "degrees of freedom" are approximately 9997.2. The p-value is 0.8223, that is greater than the commonly used significance level of p-value (0.05). This indicates that there is no statistically significant difference between the means of xp and yp. The 95% confidence interval for the difference in means between xp and yp is approximately (-0.074, 0.059). The mean of xp is approximately 0.012 and the mean of yp is approximately 0.019. #Interpretation of findings The p-value of 0.8223 suggests that there is no evidence to reject the null hypothesis. This indicates that the true difference in means between xp and yp is likely to be zero and means of xp and yp are very close with yp having a slightly higher mean than xp.This difference is not statistically significant. The slight difference in means between xp and yp may not be practically important, especially given the narrow confidence interval and the fact that it contains zero.

Randomly swapping the signs of an almost uniformly positive distribution is going to lead to the w_p distribution being more centered around zero. It is not surprising that the null hypothesis cannot be rejected in this case. Below the original w distribution and the uniformly sign-shifted w_p distribution.

```r
hist(w)
```




```
hist(w_p)
```



###Exercise 3 Clustering #Load microarray data

```
library(tidyverse)
library(ISLR2)
library(cluster)
library(clValid)
nci.labs <- NCI60$labs
nci.data <- NCI60$data
#Check size of array
dim(nci.data)
```

```
## [1]   64 6830
```

```
#Gene expression as double
#nci.data[,1]
```

#So as stated in the assignment, we have 6,830 gene expression measurements on 64 cancer cell lines.

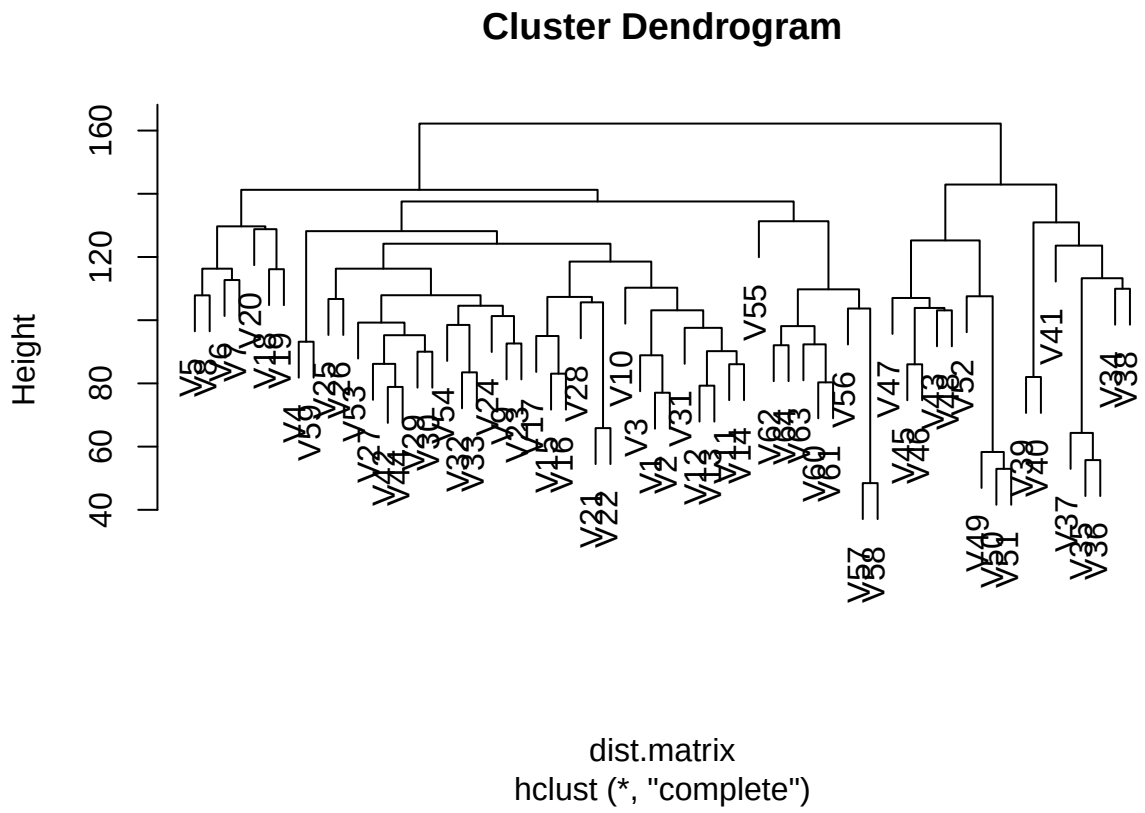
```
#Let's transpose the data so the gene expressions are the rows
#nci.data.t <- t(nci.data)
#nci.labs.t <- t(nci.labs)
#Remove NAs and scale
nci.data.cleaned <- na.omit(nci.data)#.t)
nci.data.scaled <- scale(nci.data.cleaned)

#Calculate distance matrix
```

```
dist.matrix <- dist(nci.data.scaled, method="euclidean")
str(dist.matrix)
```

```
##  'dist' num [1:2016] 77 87.3 103.2 113.7 108.3 ...
##  - attr(*, "Size")= int 64
##  - attr(*, "Labels")= chr [1:64] "V1" "V2" "V3" "V4" ...
##  - attr(*, "Diag")= logi FALSE
##  - attr(*, "Upper")= logi FALSE
##  - attr(*, "method")= chr "euclidean"
##  - attr(*, "call")= language dist(x = nci.data.scaled, method = "euclidean")
```

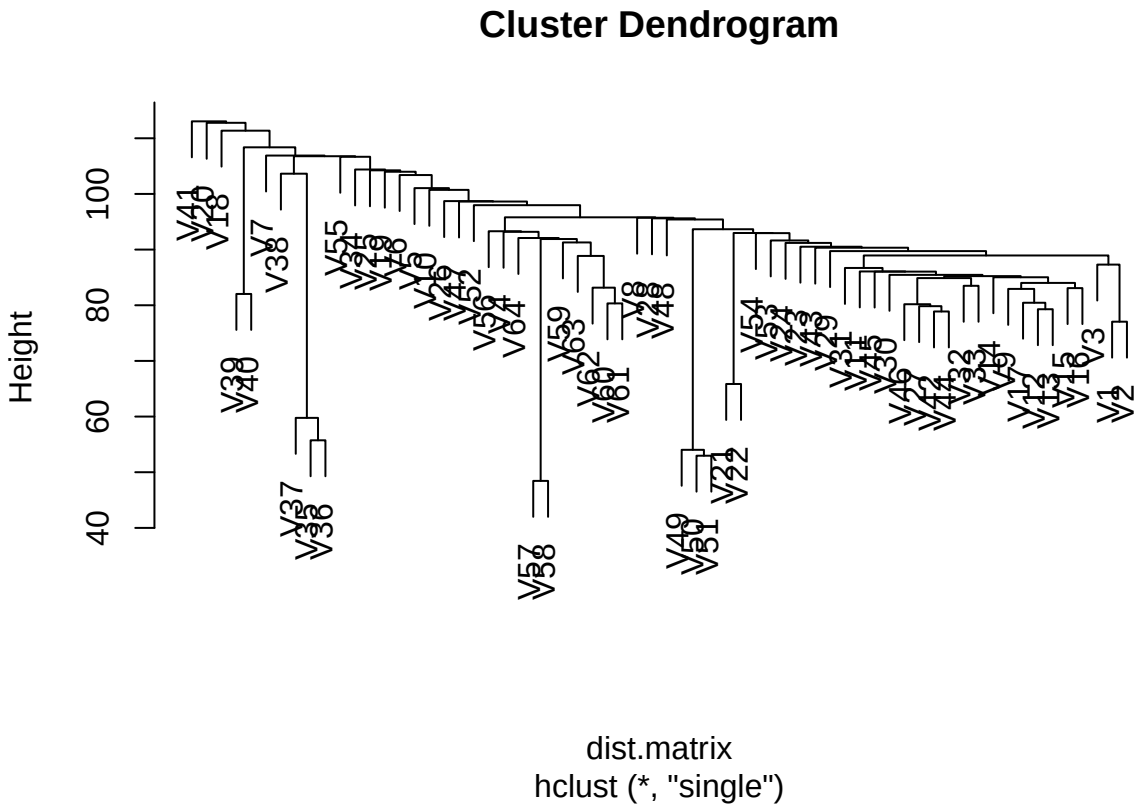
#Hierarchical Clustering #Complete

```
clustersCom <- hclust(dist.matrix,method = "complete")
plot(clustersCom)
```



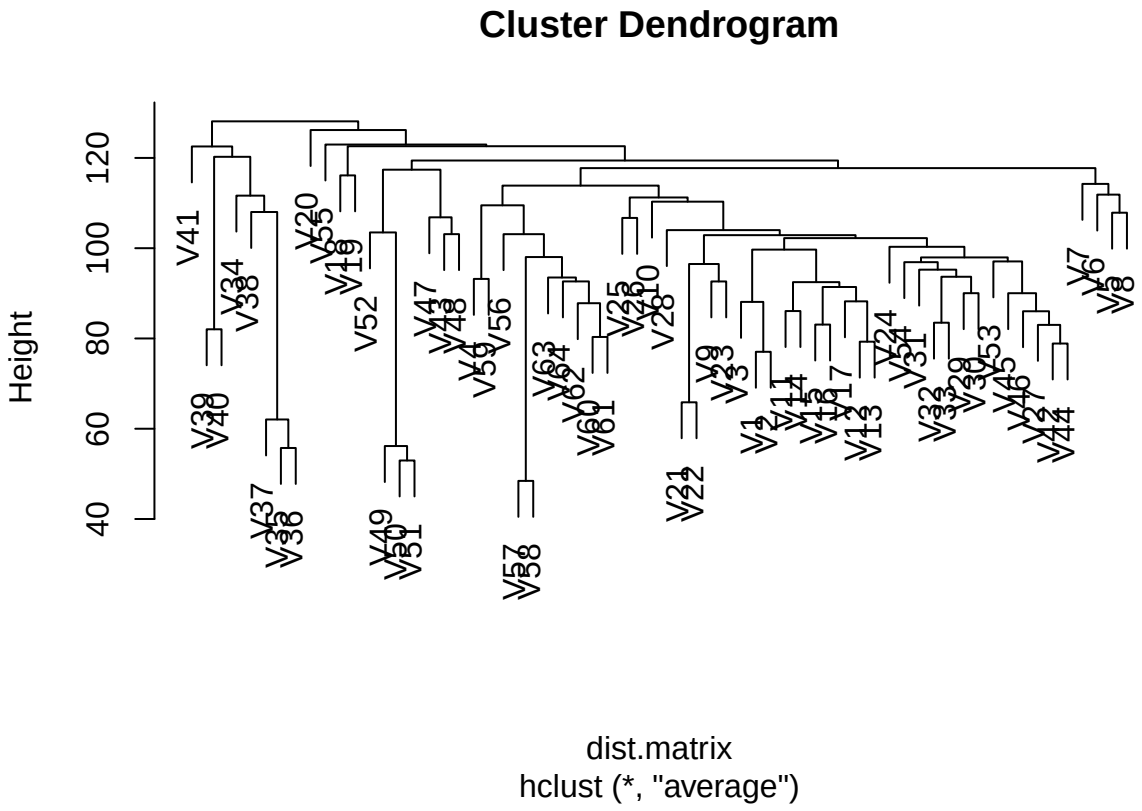
#Single

```
clustersS <- hclust(dist.matrix,method = "single")
plot(clustersS)
```



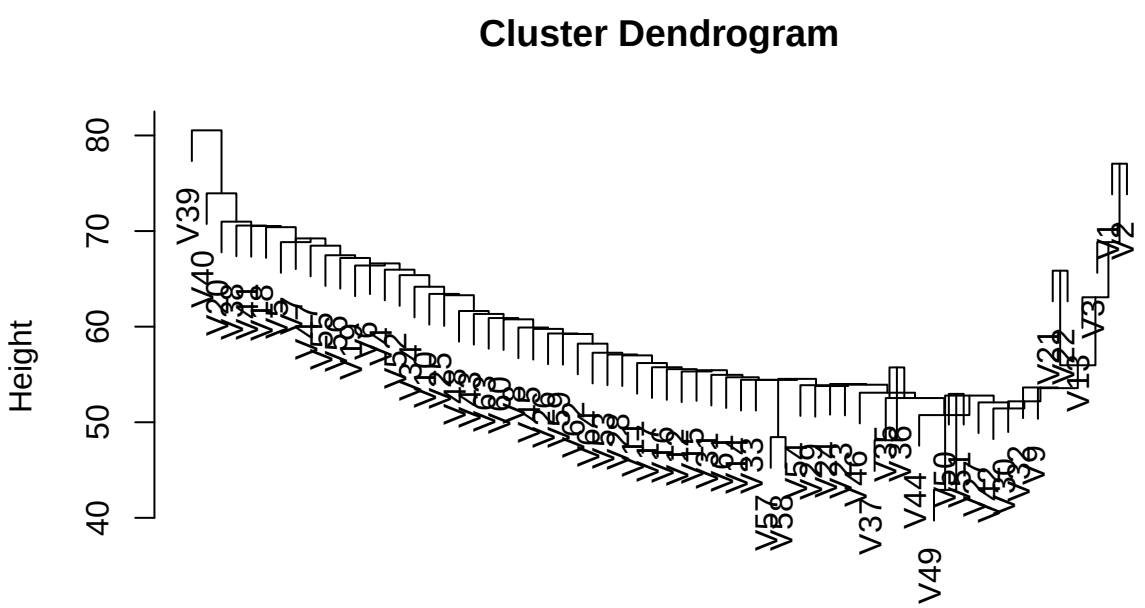
#Average

```
clustersA <- hclust(dist.matrix,method = "average")
plot(clustersA)
```



#Centroid

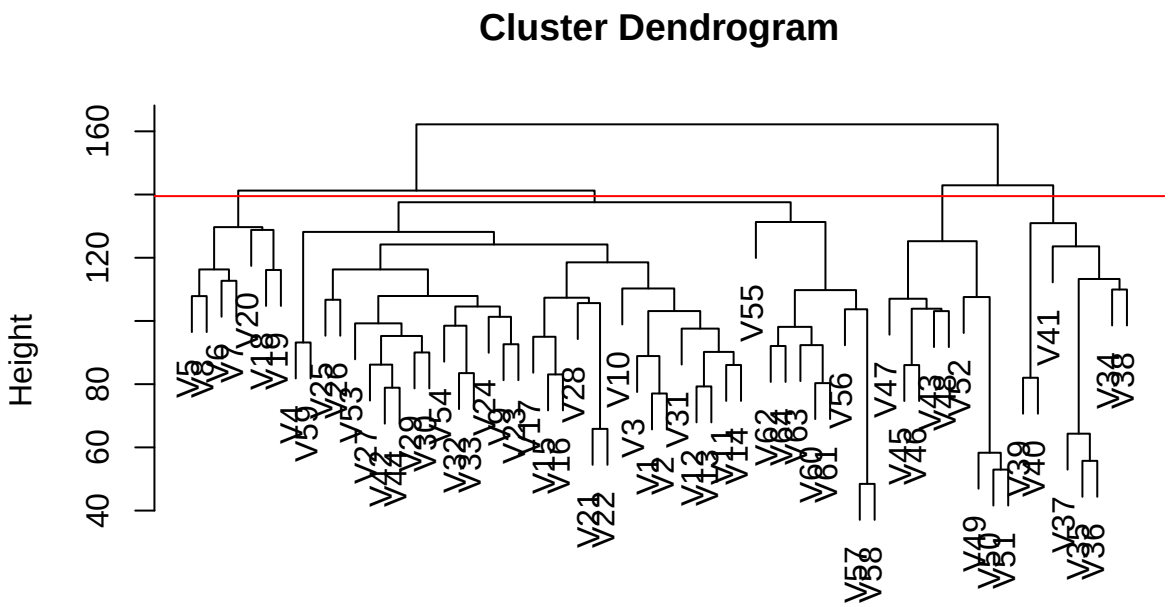
```
clustersCen <- hclust(dist.matrix,method = "centroid")
plot(clustersCen)
```



The form of each dendrogram reflects the method used for clustering. Single linkage begins with the most similar points (minimum distance) between clusters, which seems to reduce dimensions in a way that there are still a lot of similar groups as we approach the top of the dendrogram. Centroid linkage compares the centroid difference between clusters and reveals more clustering structure higher up the dendrogram than single linkage, but is still difficult to decipher where one could make cuts. Complete linkage uses the maximal dissimilarity between objects in a cluster and as a result, seems to visually do a good job of gathering the branches into identifiable groupings while average linkage computes the average pairwise dissimilarity between all objects in clusters. As complete linkage seems to have the clearest differentiation near the top of the dendrogram, we'll use it for our cut to analyze the groupings.

#Cutting the dendrogram

```
cut <- cutree(clustersCom, k = 4)#c(4,6))
plot(clustersCom) + abline(h=139.5, col='red')
```




```
## integer(0)
```

#Compare results

```
df <- data.frame(nci.labs, cut)
df %>% count(nci.labs, cut)
```

```
##       nci.labs cut n
## 1       BREAST   1 2
## 2       BREAST   2 3
## 3       BREAST   4 2
## 4          CNS   1 3
## 5          CNS   2 2
## 6        COLON   1 2
## 7        COLON   4 5
## 8  K562A-repro   3 1
## 9  K562B-repro   3 1
## 10    LEUKEMIA   3 6
## 11 MCF7A-repro   4 1
## 12 MCF7D-repro   4 1
## 13    MELANOMA   1 8
## 14       NSCLC   1 8
## 15       NSCLC   2 1
## 16     OVARIAN   1 6
## 17    PROSTATE   1 2
```

```
## 18       RENAL   1 8
## 19       RENAL   2 1
## 20     UNKNOWN   1 1
```

So it looks like we have some misclassified labels, where breast cancer is spread across 3 groups, for example. That brings us back to the question what a "reasonable" number of groupings would be, for which I'd need more expertise. How many different labels do we have?

```
length(unique(nci.labs))
```

```
## [1] 14
```

```
#So let's try 14 groups
cut14 <- cutree(clustersCom, k = 14)#c(4,6))

df14 <- data.frame(nci.labs, cut14)
df14 %>% count(nci.labs, cut14)
```

```
##       nci.labs cut14 n
## 1       BREAST     3 2
## 2       BREAST     6 1
## 3       BREAST    12 2
## 4       BREAST    14 2
## 5          CNS     1 3
## 6          CNS     3 2
## 7        COLON     4 2
## 8        COLON    11 5
## 9  K562A-repro     8 1
## 10 K562B-repro     8 1
## 11    LEUKEMIA     8 3
## 12    LEUKEMIA     9 2
## 13    LEUKEMIA    10 1
## 14 MCF7A-repro    12 1
## 15 MCF7D-repro    12 1
## 16    MELANOMA     2 1
## 17    MELANOMA     4 1
## 18    MELANOMA    14 6
## 19       NSCLC     1 2
## 20       NSCLC     4 5
## 21       NSCLC     6 1
## 22       NSCLC    13 1
## 23     OVARIAN     4 4
## 24     OVARIAN     5 2
## 25    PROSTATE     4 2
## 26       RENAL     1 4
## 27       RENAL     2 1
## 28       RENAL     5 3
## 29       RENAL     7 1
## 30     UNKNOWN     5 1
```

#Still not completely impressed. Let's try somewhere in between 4 and 14: 10?

```r
length(unique(nci.labs))
```

```
## [1] 14
```

```r
#So let's try 10 groups
cut10 <- cutree(clustersCom, k = 10)#c(4,6))

df10 <- data.frame(nci.labs, cut10)
df10 %>% count(nci.labs, cut10)
```

```
##       nci.labs cut10 n
## 1       BREAST     3 2
## 2       BREAST     4 1
## 3       BREAST     8 2
## 4       BREAST    10 2
## 5          CNS     1 3
## 6          CNS     3 2
## 7        COLON     1 2
## 8        COLON     8 5
## 9  K562A-repro     6 1
## 10 K562B-repro     6 1
## 11    LEUKEMIA     6 4
## 12    LEUKEMIA     7 2
## 13 MCF7A-repro     8 1
## 14 MCF7D-repro     8 1
## 15    MELANOMA     1 1
## 16    MELANOMA     2 1
## 17    MELANOMA    10 6
## 18       NSCLC     1 7
## 19       NSCLC     4 1
## 20       NSCLC     9 1
## 21     OVARIAN     1 6
## 22    PROSTATE     1 2
## 23       RENAL     1 7
## 24       RENAL     2 1
## 25       RENAL     5 1
## 26     UNKNOWN     1 1
```

What if we try it with another algorithm?

```r
length(unique(nci.labs))
```

```
## [1] 14
```

```r
#So let's try 14 groups
cut4c <- cutree(clustersCom, k = 4)#c(4,6))

df4c <- data.frame(nci.labs, cut4c)
df4c %>% count(nci.labs, cut4c)
```

```
##       nci.labs cut4c n
```

```
## 1       BREAST     1 2
## 2       BREAST     2 3
## 3       BREAST     4 2
## 4          CNS     1 3
## 5          CNS     2 2
## 6        COLON     1 2
## 7        COLON     4 5
## 8  K562A-repro     3 1
## 9  K562B-repro     3 1
## 10    LEUKEMIA     3 6
## 11 MCF7A-repro     4 1
## 12 MCF7D-repro     4 1
## 13    MELANOMA     1 8
## 14       NSCLC     1 8
## 15       NSCLC     2 1
## 16     OVARIAN     1 6
## 17    PROSTATE     1 2
## 18       RENAL     1 8
## 19       RENAL     2 1
## 20     UNKNOWN     1 1
```

This seems a little better using the centroid linkage. Breast cancer appears in 1,2, and 4, but colon cancer, melanoma, leukemia, ovarian, prostate, and the repro types are all isolated in their own groups even with k=4. Now let's take a look at k means

```
set.seed(123)
cl <- kmeans(nci.data.scaled, 4)
str(cl)
```

```
## List of 9
##  $ cluster     : Named int [1:64] 4 4 4 4 2 2 2 2 2 2 ...
##   ..- attr(*, "names")= chr [1:64] "V1" "V2" "V3" "V4" ...
##  $ centers     : num [1:4, 1:6830] 0.0205 -0.2254 -0.3974 0.54 -0.0821 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:4] "1" "2" "3" "4"
##   .. ..$ : chr [1:6830] "1" "2" "3" "4" ...
##  $ totss       : num 430290
##  $ withinss    : num [1:4] 37150 98055 110313 107250
##  $ tot.withinss: num 352767
##  $ betweenss   : num 77523
##  $ size        : int [1:4] 9 17 18 20
##  $ iter        : int 2
##  $ ifault      : int 0
##  - attr(*, "class")= chr "kmeans"
```

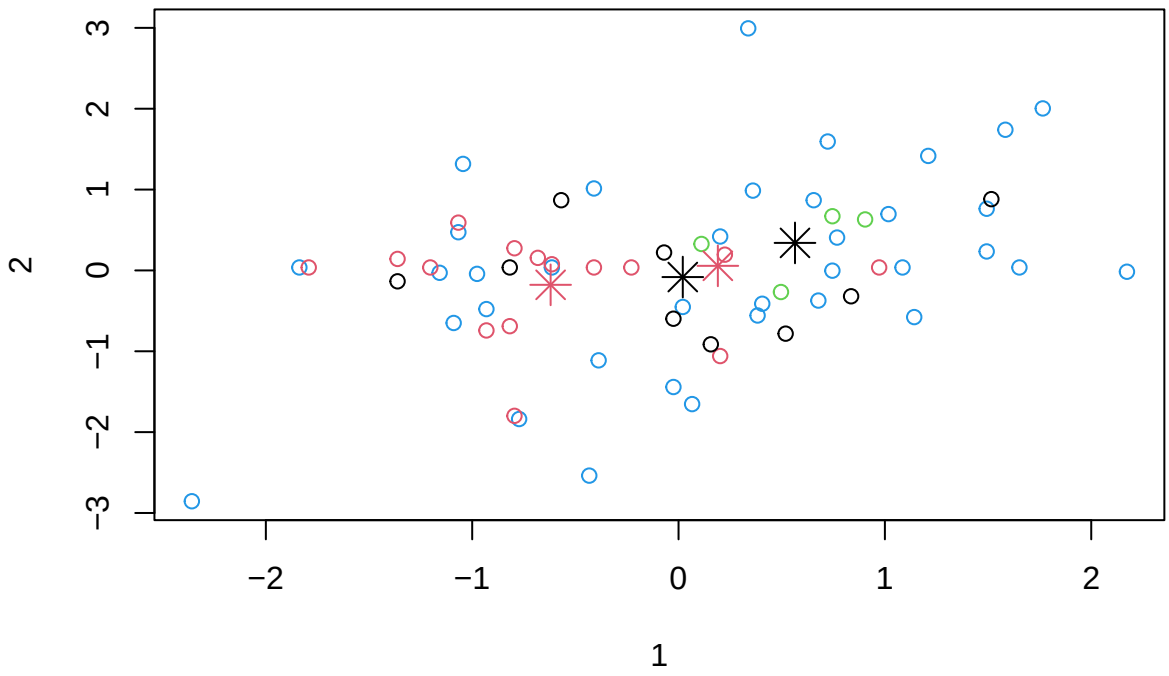
```
df_k <- data.frame(nci.labs,cl$cluster)
df_k %>% count(nci.labs, cl$cluster)
```

```
##       nci.labs cl$cluster n
## 1       BREAST          1 2
## 2       BREAST          2 3
## 3       BREAST          3 1
## 4       BREAST          4 1
```

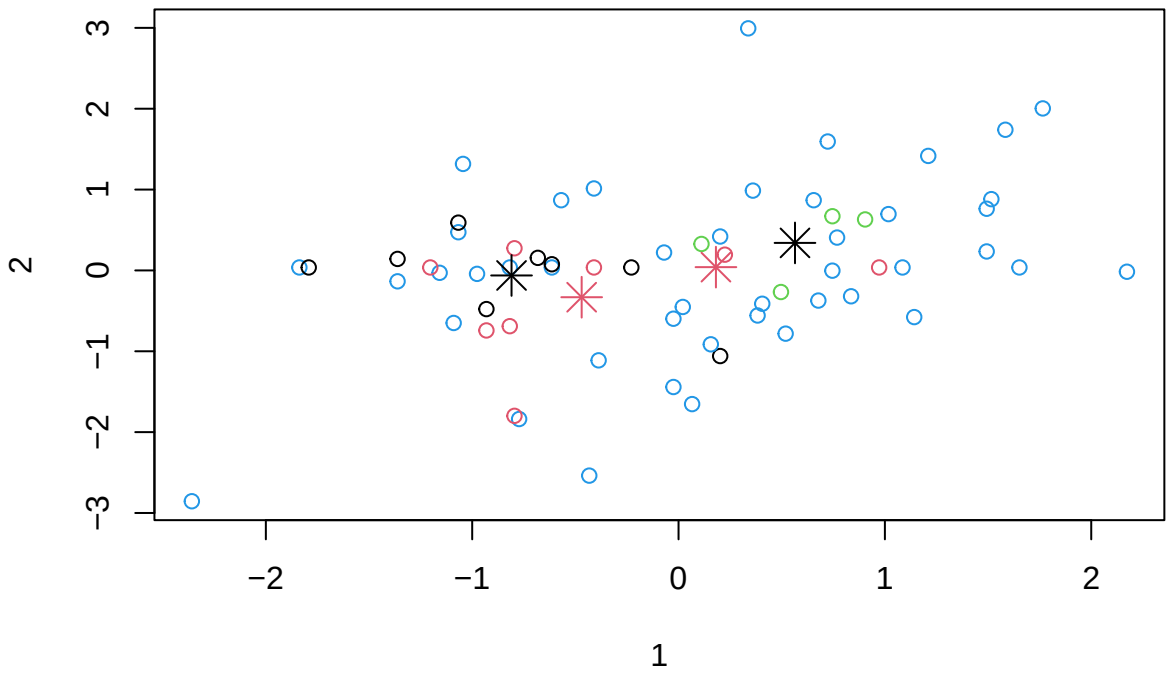
```
## 5          CNS          2 2
## 6          CNS          4 3
## 7        COLON          3 7
## 8  K562A-repro          3 1
## 9  K562B-repro          3 1
## 10    LEUKEMIA          3 6
## 11 MCF7A-repro          3 1
## 12 MCF7D-repro          3 1
## 13    MELANOMA          1 7
## 14    MELANOMA          2 1
## 15       NSCLC          2 3
## 16       NSCLC          4 6
## 17     OVARIAN          2 1
## 18     OVARIAN          4 5
## 19    PROSTATE          4 2
## 20       RENAL          2 6
## 21       RENAL          4 3
## 22     UNKNOWN          2 1
```

The clustering seems to have done similarly, with breast cancer (4), CNS (2), melanoma(2), and renal (2) appearing across multiple groupings.

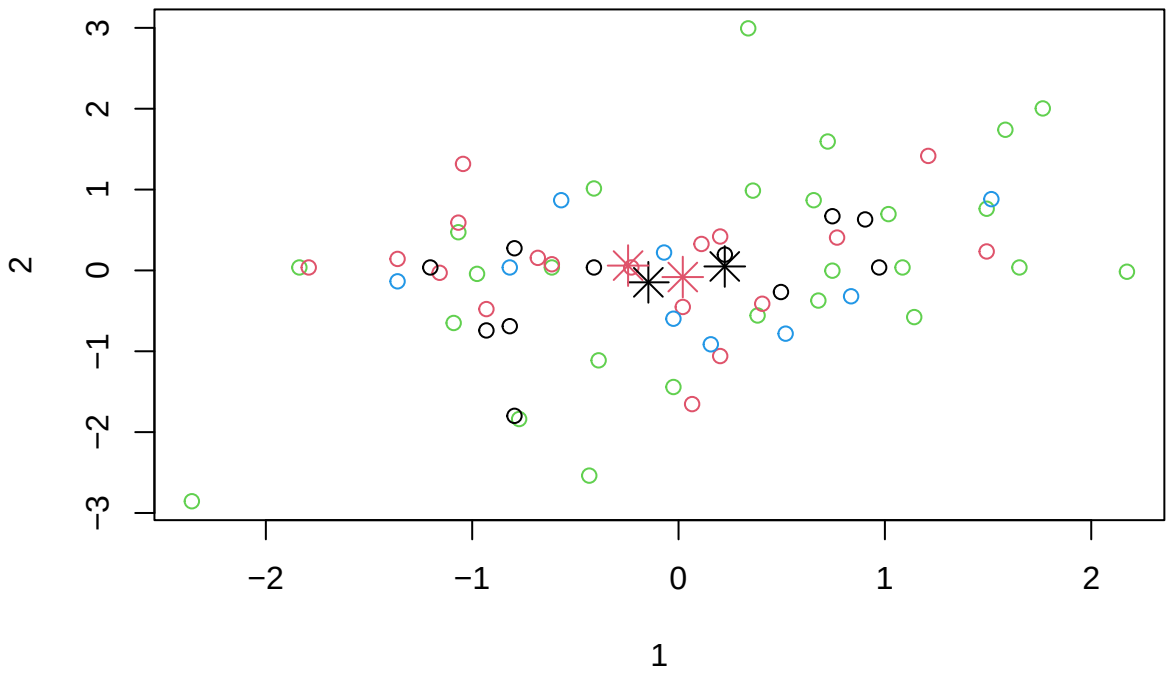
```
lengthTest <- 64
meansList <- list()
for (t in 1:5) {
  (cl <- kmeans(nci.data.scaled, 4))
  plot(nci.data.scaled, col = cl$cluster)
  points(cl$centers, col = 1:2, pch = 8, cex = 2)
  df_i <- data.frame(nci.labs,cl$cluster)
  resultDF <- df_i %>% count(nci.labs, cl$cluster)
  lengthTest_i <-length(resultDF$n)
  meansList <- c(meansList, cl)
  if (lengthTest_i < lengthTest) {
    print(lengthTest_i)
    lengthTest <- lengthTest_i
    t_f <- t
    cl_f <- cl
    resultDF_f <- resultDF
  }

}
```

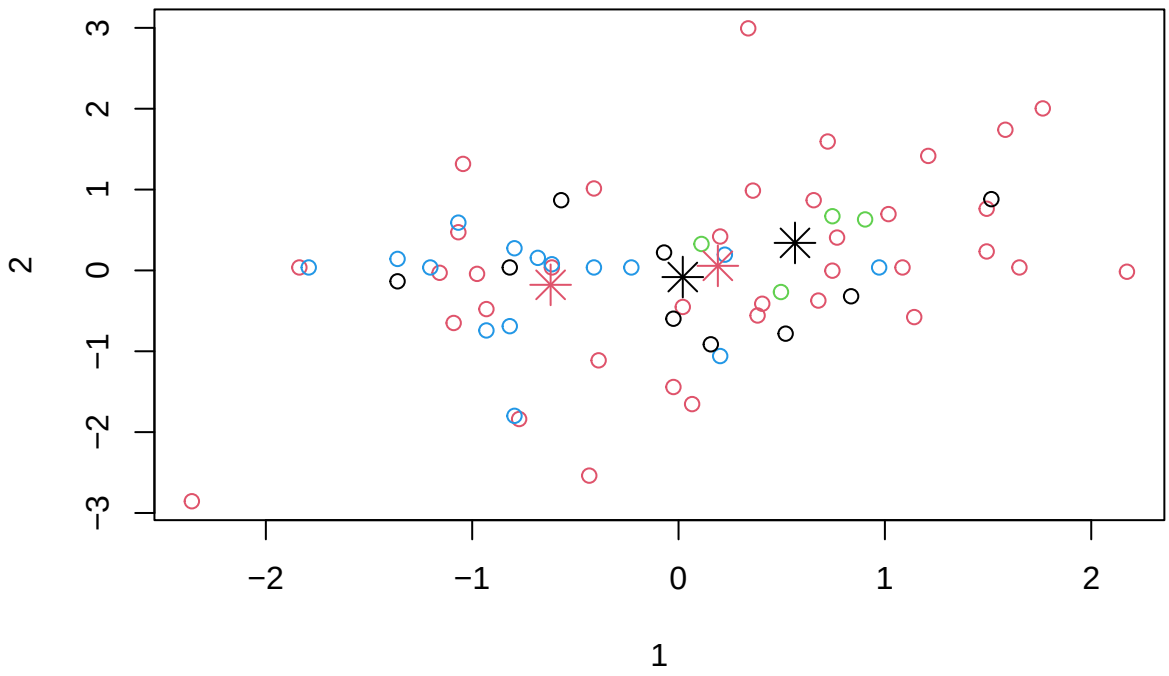


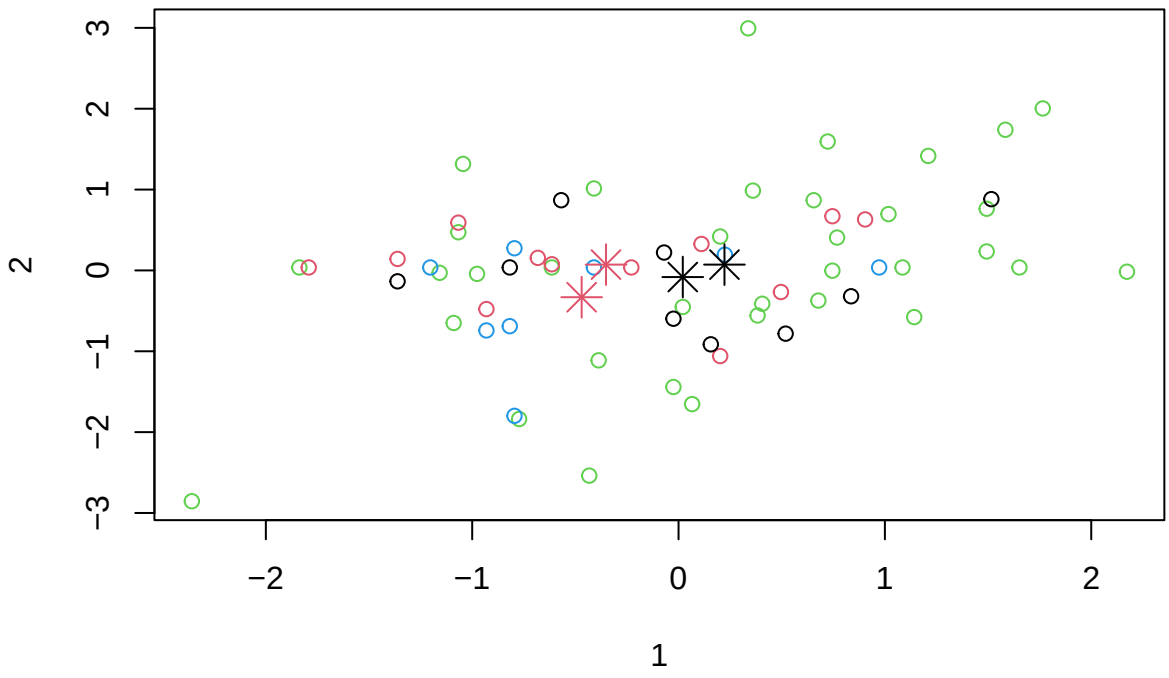

```
## [1] 17
```




```
## [1] 16
```


3
2
1
0
−1
−2
−3
2
−2
−1
0
1
2
1


3
2
1
0
−1
−2
−3
2
−2
−1
0
1
2
1




```
resultDF_f
```

```
##       nci.labs cl$cluster n
## 1       BREAST          3 2
## 2       BREAST          4 5
## 3          CNS          4 5
## 4        COLON          1 7
## 5  K562A-repro          2 1
## 6  K562B-repro          2 1
## 7     LEUKEMIA          2 6
## 8  MCF7A-repro          3 1
## 9  MCF7D-repro          3 1
## 10    MELANOMA          4 8
## 11       NSCLC          1 1
## 12       NSCLC          4 8
## 13     OVARIAN          4 6
## 14    PROSTATE          4 2
## 15       RENAL          4 9
## 16     UNKNOWN          4 1
```

Obviously, 64 dimensional data is difficult to visualize in 2d, so I'll print the best fit based on the labels. One of the K means split all values into their own groups outside of breast cancer (2) and NSCLC (2). This isn't necessarily a "good" clustering result though, as this is information that we shouldn't have. Let's use a validation package to check what we "should" do.

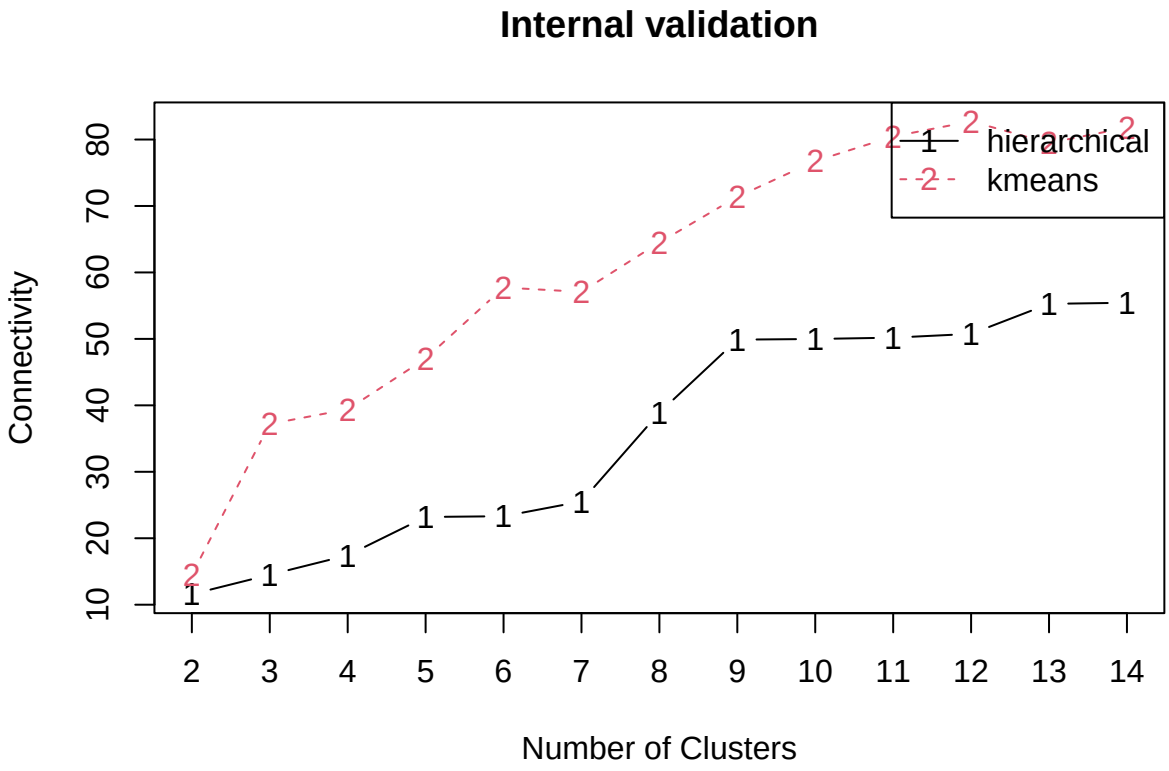
```r
intern <- clValid(nci.data.scaled, 2:14, clMethods=c("hierarchical","kmeans"),
                  validation="internal")
#metric = "euclidean" and method = "average" are default
summary(intern)
```

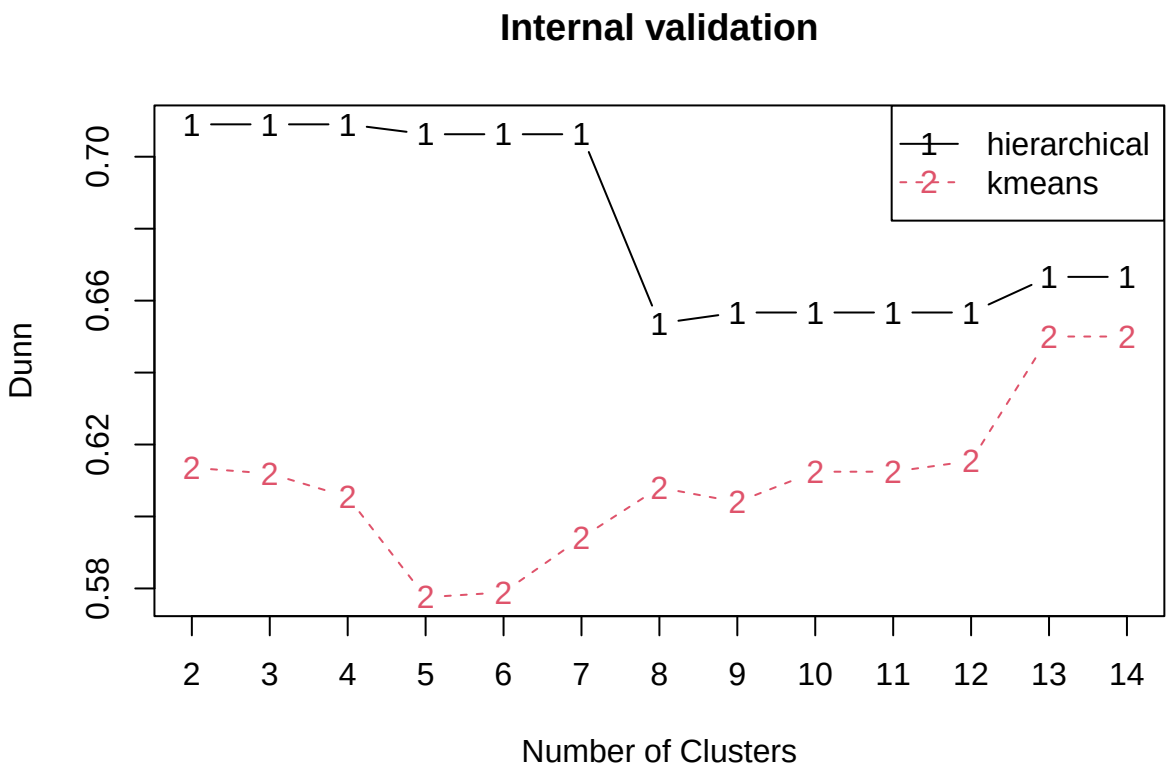
```
##
## Clustering Methods:
##  hierarchical kmeans
##
## Cluster sizes:
##  2 3 4 5 6 7 8 9 10 11 12 13 14
##
## Validation Measures:
##                                  2       3       4       5       6       7       8       9      10
##
## hierarchical Connectivity  11.5754 14.5044 17.4333 23.2024 23.3135 25.5325 38.9063 49.8861 49.9861 50
##              Dunn           0.7090  0.7090  0.7090  0.7063  0.7063  0.7063  0.6537  0.6567  0.6567  0
##              Silhouette     0.1214  0.0940  0.0785  0.0717  0.0697  0.0781  0.0855  0.0800  0.0877  0
## kmeans       Connectivity  14.5115 37.2020 39.3996 47.0385 57.7298 57.0742 64.5091 71.3786 76.7627 80
##              Dunn           0.6136  0.6119  0.6054  0.5776  0.5789  0.5941  0.6080  0.6040  0.6124  0
##              Silhouette     0.1142  0.0609  0.0632  0.0847  0.0855  0.1003  0.0906  0.0958  0.0949  0
##
## Optimal Scores:
##
##              Score   Method       Clusters
## Connectivity 11.5754 hierarchical 2
## Dunn          0.7090 hierarchical 2
## Silhouette    0.1214 hierarchical 2
```

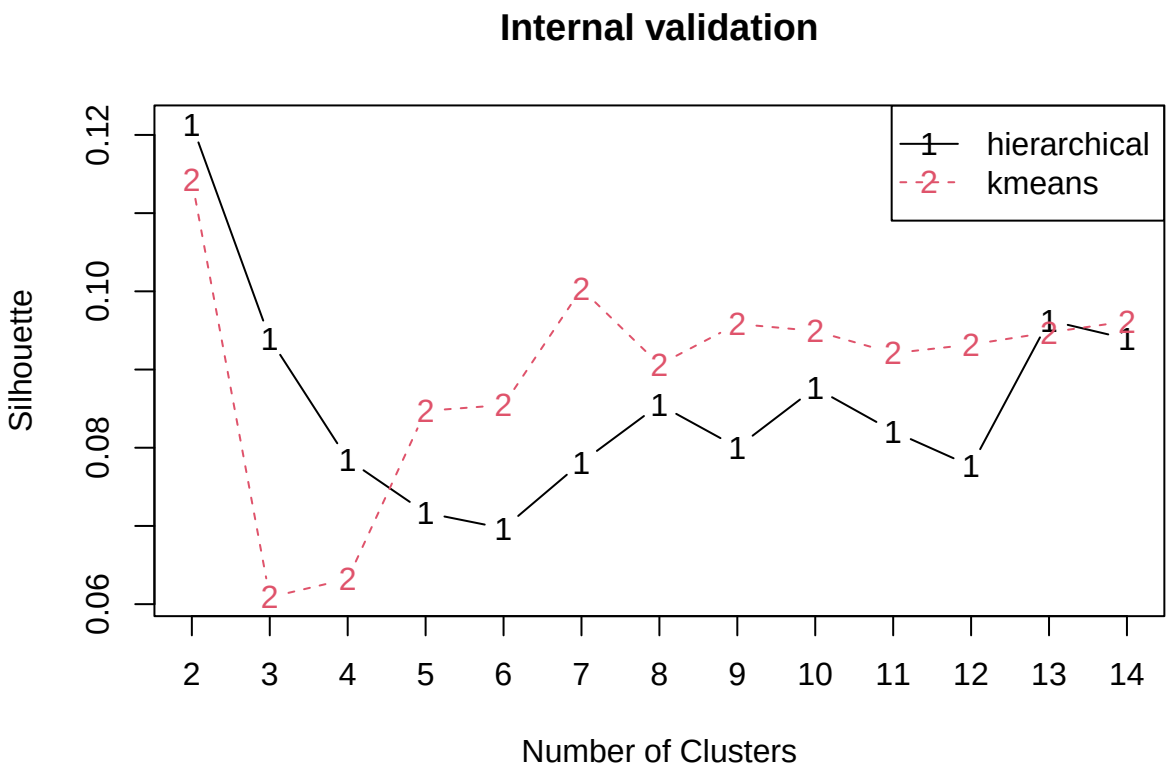
```r
optimalScores(intern)
```

```
##                   Score       Method Clusters
## Connectivity 11.5753968 hierarchical        2
## Dunn          0.7089980 hierarchical        2
## Silhouette    0.1213521 hierarchical        2
```

```r
plot(intern)
```


Internal validation
Connectivity
Number of Clusters
hierarchical
kmeans
10
20
30
40
50
60
70
80
2
3
4
5
6
7
8
9
10
11
12
13
14


Internal validation
Dunn
0.58
0.62
0.66
0.70
2
3
4
5
6
7
8
9
10
11
12
13
14
Number of Clusters
hierarchical
kmeans



So it seems like the optimal k for kMeans would be seven groups as the silhouette score has a local maximum(?). Let's take a look:

```
lengthTest <- 64
meansList <- list()
for (t in 1:5) {
  (cl <- kmeans(nci.data.scaled, 7))
  df_i <- data.frame(nci.labs,cl$cluster)
  resultDF <- df_i %>% count(nci.labs, cl$cluster)
  lengthTest_i <-length(resultDF$n)
  meansList <- c(meansList, cl)
  if (lengthTest_i < lengthTest) {
    print(lengthTest_i)
    lengthTest <- lengthTest_i
    t_f <- t
    cl_f <- cl
    resultDF_f <- resultDF
  }

}
```

```
## [1] 24
```

```
resultDF_f
```

```
##       nci.labs cl$cluster n
```

```
## 1       BREAST          3 2
## 2       BREAST          4 2
## 3       BREAST          5 1
## 4       BREAST          6 2
## 5          CNS          3 4
## 6          CNS          5 1
## 7        COLON          1 7
## 8  K562A-repro          2 1
## 9  K562B-repro          2 1
## 10    LEUKEMIA          2 6
## 11 MCF7A-repro          4 1
## 12 MCF7D-repro          4 1
## 13    MELANOMA          6 7
## 14    MELANOMA          7 1
## 15       NSCLC          1 1
## 16       NSCLC          3 1
## 17       NSCLC          5 1
## 18       NSCLC          7 6
## 19     OVARIAN          7 6
## 20    PROSTATE          7 2
## 21       RENAL          3 3
## 22       RENAL          5 1
## 23       RENAL          7 5
## 24     UNKNOWN          7 1
```

So even with the optimized k for the kMeans method, we end up with some cancer types split across the groupings. The repro variants are all isolated, as well as leukemia and colon cancer, which is uniquely grouped with a single instance of NSCLC.

However, the highest score seems to be just two groups for average linkage hierarchical clustering:

```
cutA <- cutree(clustersA, k=2)
df <- data.frame(nci.labs, cutA)
df %>% count(nci.labs, cutA)
```

```
##       nci.labs cutA n
## 1       BREAST    1 7
## 2          CNS    1 5
## 3        COLON    1 7
## 4  K562A-repro    2 1
## 5  K562B-repro    2 1
## 6     LEUKEMIA    2 6
## 7  MCF7A-repro    1 1
## 8  MCF7D-repro    1 1
## 9     MELANOMA    1 8
## 10       NSCLC    1 9
## 11     OVARIAN    1 6
## 12    PROSTATE    1 2
## 13       RENAL    1 9
## 14     UNKNOWN    1 1
```

Not a single cancer type appears across both groupings. Leukemia, K562A-repro, and K562B-repro appear in their own group and all other cancer types are grouped together.

###Exercise 4 a) As we are comparing a continuous variable in blood pressure, a chi square test would only be appropriate if we converted that continuous variable into groupings (high/low, for example). A more appropriate test would be an analysis of variance (ANOVA). b) In order to do a power analysis, we need to know the variance (standard deviation) of the data as well as the hypothesized mean difference between groups. The other quantities in this case are known (significance level 0.05, group size 20, number of groups 2). c) The easiest factor to change to increase the power would be the sample size of the study. d) TRUE Putting an equal amount of male and female participants in the high and low blood alcohol groups is an example of stratified sampling where we identify a potential confounding factor and make our groups as similar as we can, a process known as blocking. e) FALSE The number of subjects in a study is not related to multiple testing corrections, however, a correction may be necessary as we are running multiple tests on the same research question. f) FALSE, we should ALSO randomly select from subgroups we have identified as potentially confounding factors in the data. g) For a study of the effects of alcohol on blood pressure, a smoking habit would be an example of a counfounder or confounding factor. h) The second study showed that both obese and non-obese people showed lower blood pressure in a control group than in a high-alcohol group. However, when the entire population was observed, the high-alcohol group had lower blood pressure. This suggests that the mean difference in blood pressure between men and women is driving the observed mean differences in the total population, likely through differing distribution and the control and treatment groups. In order to estimate if the effect is present in the total population, one could adjust for these baseline differences by mean shifting blood pressure for the obese population with the difference from the non-obese population and running the comparison. i) FALSE, the second test has mostly confirmed two confounding factors that we likely should have blocked in the first study. j) FALSE, this ad-hoc comparison shows that mean blood pressure in a control group is lower than in a high alcohol group in both obese and non-obese groups than would be expected due to random variation. Mostly, the second study shows that there are complex relationships between multiple confounding factors and that a future experimental design should take those into account.