Unsupervised Learning

James et al., Ch. 10

Brooke Anderson

April 19, 2016

Unsupervised learning

Example applications:

- Look for subgroups among samples of breast cancer patients
- ID shoppers with similar browsing and purchase histories (recommendation system)

Challenges:

- No well-defined goal
- Hard to assess if a technique did well
- Hard to determine if clusters just result from noise
- Choices like number of clusters, dissimilarity measure, and type of linkage can have a big influence on results

NCI-60 panel of cancer cells

There is an R package called `ISLR` that includes datasets to go along with the James textbook.

```
library(ISLR)
## ?NCI60
```

The `NCI60` dataset is microarray data from the National Cancer Institute, with expression levels on 6,830 genes from 64 cancer cells.

NCI-60 panel of cancer cells

The dataset has two parts:

- `labs`: Labels with the cancer type for each cell line. Vector, length 64.
- `data`: Dataframe, 64 rows, 6,830 columns.

```
nci_labs <- NCI60$labs
nci_data <- NCI60$data


nci_labs[1:4]
```

```
## [1] "CNS"   "CNS"   "CNS"   "RENAL"
```

NCI-60 panel of cancer cells

```
data.frame(cancer = nci_labs) %>% group_by(cancer) %>%
  summarize(n = n()) %>% arrange(desc(n)) %>% kable()
```

cancer	n
NSCLC	9
RENAL	9
MELANOMA	8
BREAST	7
COLON	7
LEUKEMIA	6
OVARIAN	6
CNS	5
PROSTATE	2
K562A-repro	1
K562B-repro	1
MCF7A-repro	1
MCF7D-repro	1
UNKNOWN	1

NCI-60 panel of cancer cells

For the `nci_data` part of the dataset, each row is one of the cell lines and each column gives a measure of gene expression.

```
nci_data[1:5, 1:5]
```

```
##             1         2         3         4         5
## V1 0.300000  1.180000  0.550000  1.140000 -0.265000
## V2 0.679961  1.289961  0.169961  0.379961  0.464961
## V3 0.940000 -0.040000 -0.170000 -0.040000 -0.605000
## V4 0.280000 -0.310000  0.680000 -0.810000  0.625000
## V5 0.485000 -0.465000  0.395000  0.905000  0.200000
```

PCA

PCA can help with exploratory data analysis. If you did pairwise scatterplots of all 6830 gene expressions, you would need loads of plots:

$$\frac{p(p-1)}{2} = \frac{6830(6830-1)}{2} = 2.3321035 \times 10^7$$

PCA

Instead, you can plot pairwise scatterplots of the first few principal components loadings. Based on James et al.,

> *PCA helps create a "low-dimensional representation of the data that captures as much of the information as possible".*

PCA

You can use the `prcomp` function to perform a principal components analysis on a data matrix:

```
pr_out <- prcomp(nci_data, scale = TRUE)
class(pr_out)
```

```
## [1] "prcomp"
```

The output from this function has the class `prcomp`.

PCA

As a reminder, since the output is a special class, it will have special methods of things like `print` and `summary`:

```
## [1] "sdev"       "rotation"   "center"     "scale"      "x"
## [6] "importance"
```

PCA

The `$x` element of the output of `prcomp` are the value of the rotated data (i.e., the centered and scaled data multiplied by the rotation matrix):

```
dim(pr_out$x)
```

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

```
pr_out$x[1:4, 1:4]
```

```
##           PC1       PC2          PC3        PC4
## V1 -19.68245  3.527748  -9.7354382  0.8177816
## V2 -22.90812  6.390938 -13.3725378 -5.5911088
## V3 -27.24077  2.445809  -3.5053437  1.3311502
## V4 -42.48098 -9.691742  -0.8830921 -3.4180227
```

PCA

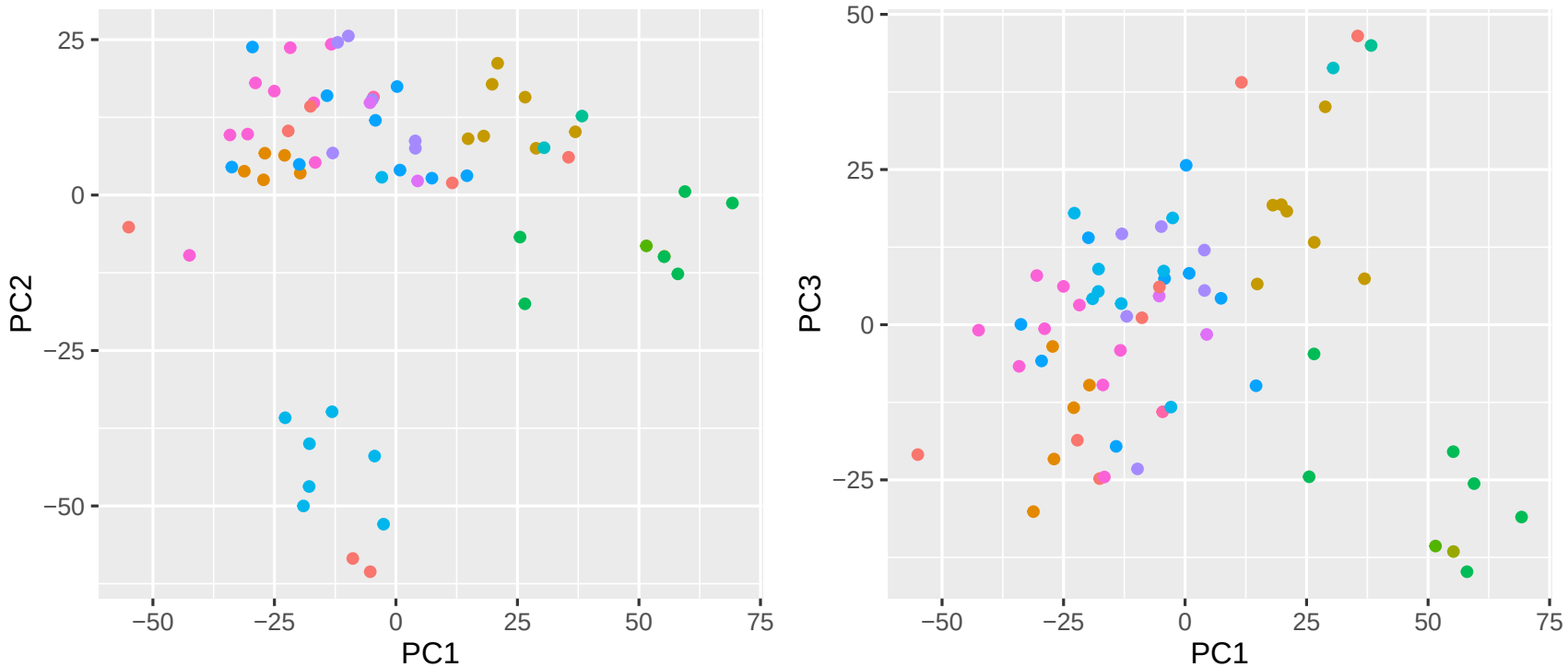
I can use this code to pull just the first three components and add the cell-type labels, to get ready to plot to look for clusters and see if they line up with cell types:

```
pr_groups <- as.data.frame(pr_out$x[ , 1:3]) %>%
  mutate(cell_type = nci_labs)
head(pr_groups)
```

```
##          PC1       PC2         PC3 cell_type
## 1 -19.68245  3.527748  -9.7354382       CNS
## 2 -22.90812  6.390938 -13.3725378       CNS
## 3 -27.24077  2.445809  -3.5053437       CNS
## 4 -42.48098 -9.691742  -0.8830921     RENAL
## 5 -54.98387 -5.158121 -20.9291076    BREAST
## 6 -26.96488  6.727122 -21.6422924       CNS
```

PCA

```
a <- ggplot(pr_groups,
            aes(x = PC1, y = PC2, color = cell_type)) +
  geom_point() + theme(legend.position="none")
b <- a %+% aes(y = PC3)
grid.arrange(a, b, ncol = 2)
```



PCA

To see the standard deviation explained by the first five components:

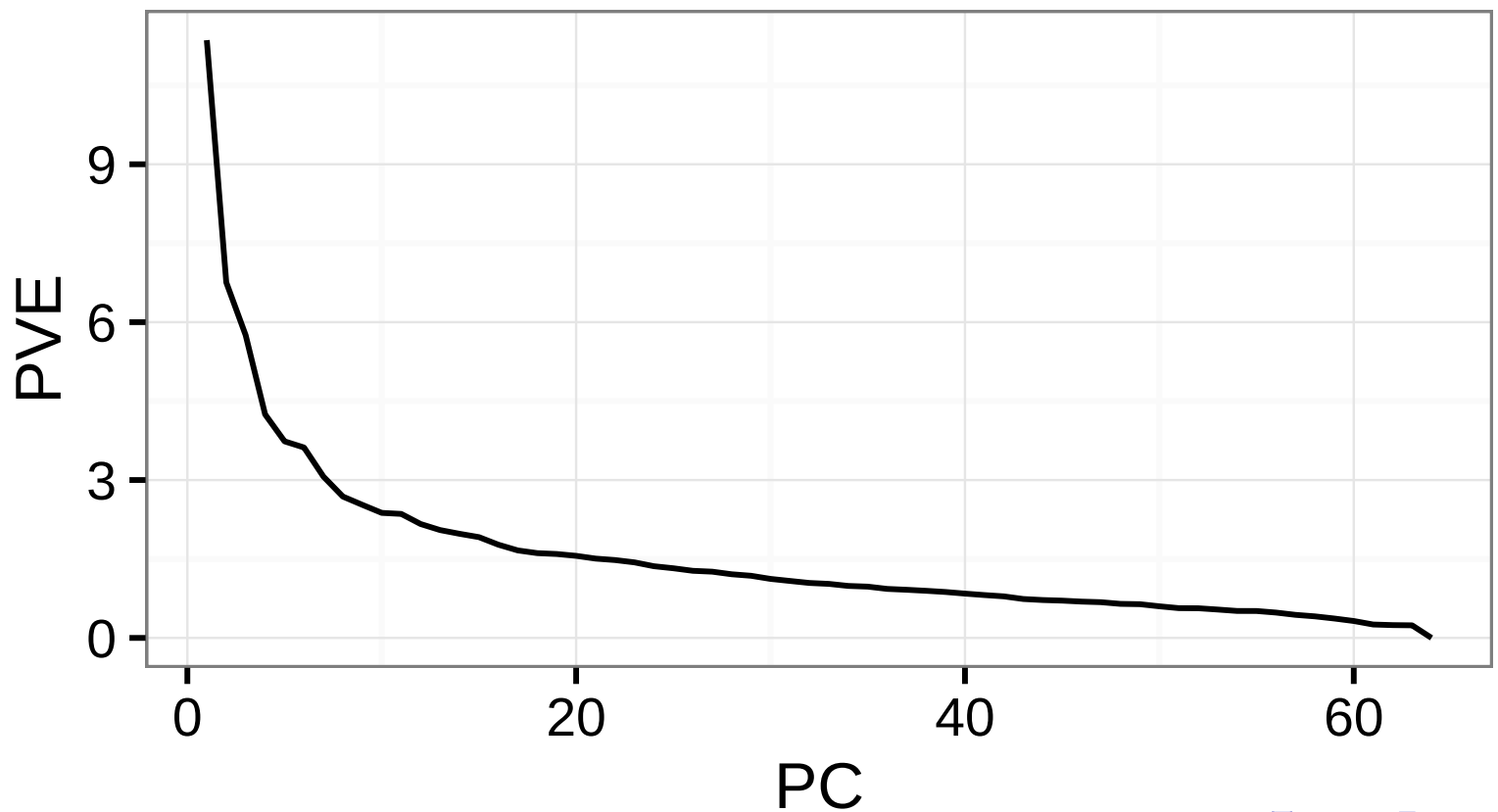
```
pr_out$sdev[1:5]
```

```
## [1] 27.85347 21.48136 19.82046 17.03256 15.97181
```

PCA

To create a scree plot:

```
to_plot <- data.frame(PC = 1:nrow(pr_out$x),
                      PVE = 100 * pr_out$sdev ^ 2 /
                        sum(pr_out$sdev ^ 2))
ggplot(to_plot, aes(x = PC, y = PVE)) + geom_line() +
  theme_bw()
```



PCA

From James et al.:

> *"Unfortunately, there is no well-accepted objective way to decide how many principal components are enough. In Fact, the question of how many principal components are enough is inherently ill-defined, and will depend on the specific area of application and the specific data set."*

Clustering methods

Goal: Create clusters so that the within-cluster variation among observations is as low as possible.

- **Hierarchical clustering**: Create a dendrogram that could be used to pick out clusters of any size.
- **K-means clustering**: Split the observations into a certain number of clusters.

You can cluster observations by features or features by observations.

Hierarchical clustering

Start by standardizing the data:

```
sd_data <- scale(nci_data)
```

Then use the `dist` function to measure Euclidean distance:

```
data_dist <- dist(sd_data, method = "euclidean")
class(data_dist)
```

```
## [1] "dist"
```

Other `method` options: “maximum”, “manhattan”, “canberra”, “binary”, “minkowski”.

Hierarchical clustering

`hclust` can be applied to a `dist` object to identify clusters:

```
nci_clusters <- hclust(data_dist)
names(nci_clusters)
```

```
## [1] "merge"       "height"      "order"       "labels"      "
## [6] "call"        "dist.method"
```

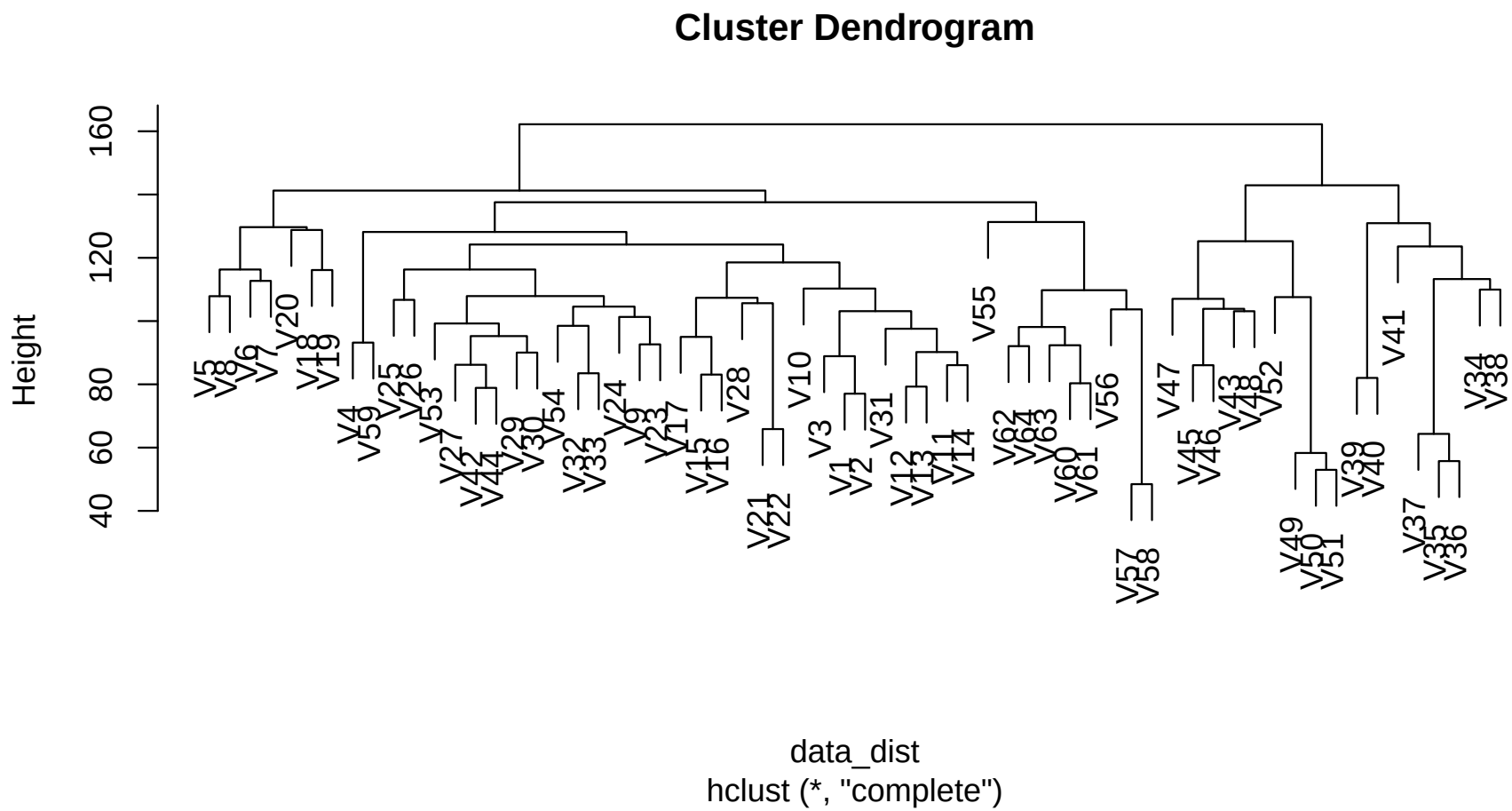
```
class(nci_clusters)
```

```
## [1] "hclust"
```

The default is to cluster using complete linkage.

Hierarchical clustering

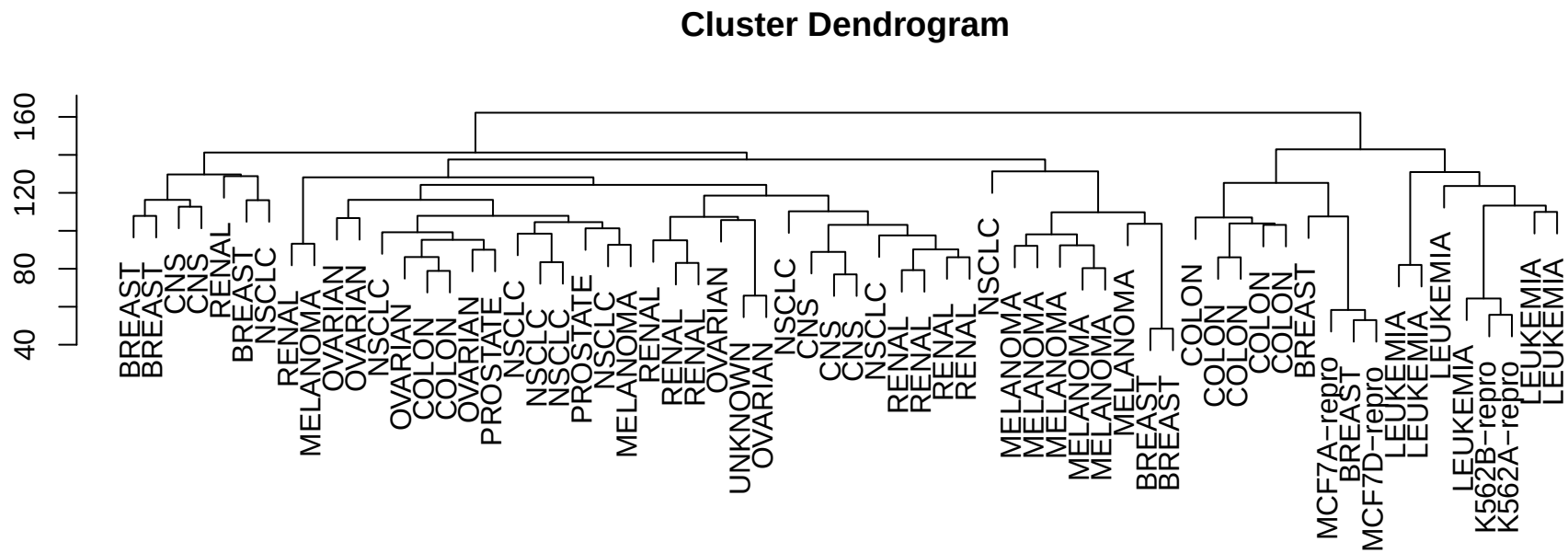
```
plot(nci_clusters)
```



Hierarchical clustering

Use the cancer type for labels:

```
plot(nci_clusters, labels = nci_labs, xlab = "",
     ylab = "", sub = "")
```



Hierarchical clustering

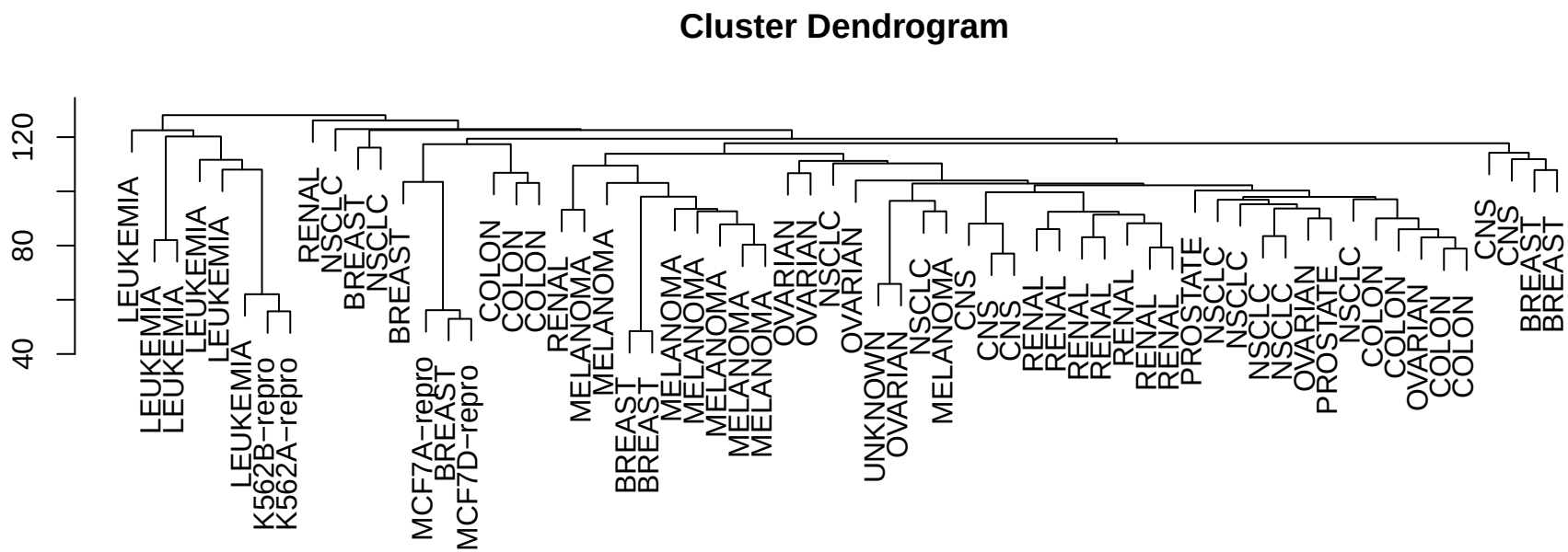
Linkage: The dissimilarity between two groups of observations (see Table 10.2 in James et al.).

- Complete: Largest of all pairwise distances between observations in cluster A and cluster B
- Average: Average of all pairwise distances between observations in cluster A and cluster B
- Single: Smallest of all pairwise distances between observations in cluster A and cluster B
- Centroid: The distance betwee the centroids of each cluster

Hierarchical clustering

By change the `hclust` arguments, you can use average linkage instead:

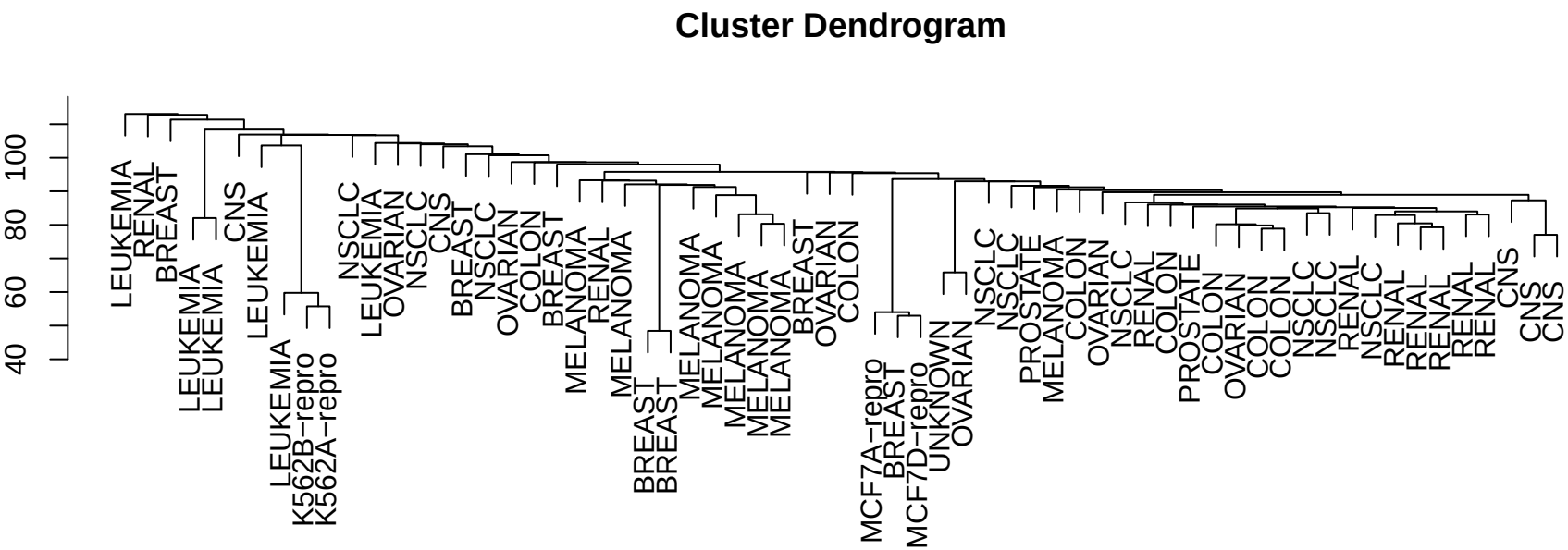
```
plot(hclust(data_dist, method = "average"),
     labels = nci_labs, xlab = "",
     ylab = "", sub = "")
```



Hierarchical clustering

Or single linkage:

```
plot(hclust(data_dist, method = "single"),
     labels = nci_labs, xlab = "",
     ylab = "", sub = "")
```



Cutting down to fewer clusters

You can use the cutree function to cut the cluster dendrogram at a certain height to only get a certain number of clusters. For example, to get four clusters:

```
hc_clusters <- cutree(nci_clusters, 4)
hc_clusters
```

```
##  V1  V2  V3  V4  V5  V6  V7  V8  V9 V10 V11 V12 V13 V14 V15 V
##   1   1   1   1   2   2   2   2   1   1   1   1   1   1   1
## V19 V20 V21 V22 V23 V24 V25 V26 V27 V28 V29 V30 V31 V32 V33 V
##   2   2   1   1   1   1   1   1   1   1   1   1   1   1   1
## V37 V38 V39 V40 V41 V42 V43 V44 V45 V46 V47 V48 V49 V50 V51 V
##   3   3   3   3   3   1   4   1   4   4   4   4   4   4   4
## V55 V56 V57 V58 V59 V60 V61 V62 V63 V64
##   1   1   1   1   1   1   1   1   1   1
```

Cutting down to fewer clusters

```
data.frame(cancer = nci_labs, cluster = hc_clusters) %>%
  group_by(cluster) %>%
  summarize(cancers = paste(unique(cancer), collapse = ", ")) %>
  pander(split.cell = 70)
```

cluster	cancers
1	CNS, RENAL, NSCLC, UNKNOWN, OVARIAN, MELANOMA, PROSTATE, COLON, BREAST
2	BREAST, CNS, NSCLC, RENAL
3	LEUKEMIA, K562B-repro, K562A-repro
4	COLON, MCF7A-repro, BREAST, MCF7D-repro

Cutting down to fewer clusters

```
data.frame(cancer = nci_labs, cluster = hc_clusters) %>%
  group_by(cluster, cancer) %>%
  summarize(n = n(),
            cancers = paste0(cancer[1], " (", n(), ")")) %>%
  arrange(cluster, desc(n)) %>%
  ungroup() %>%
  select(-cancer, -n) %>%
  group_by(cluster) %>%
  summarize(cancers = paste(cancers, collapse = ", ")) %>%
  pander(split.cell = 70)
```

Cutting down to fewer clusters

cluster	cancers
1	MELANOMA (8), NSCLC (8), RENAL (8), OVARIAN (6), CNS (3), BREAST (2), COLON (2), PROSTATE (2), UNKNOWN (1)
2	BREAST (3), CNS (2), NSCLC (1), RENAL (1)
3	LEUKEMIA (6), K562A-repro (1), K562B-repro (1)
4	COLON (5), BREAST (2), MCF7A-repro (1), MCF7D-repro (1)

K-means clustering

```
set.seed(2)
km_out <- kmeans(sd_data, 4, nstart = 20)
class(km_out)
```

```
## [1] "kmeans"
```

```
names(km_out)
```

```
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

K-means clustering

```
km_clusters <- km_out$cluster
table(km_clusters, hc_clusters)
```

```
##             hc_clusters
## km_clusters  1  2  3  4
##           1 11  0  0  9
##           2  0  0  8  0
##           3  9  0  0  0
##           4 20  7  0  0
```

K-means clustering

cluster	cancers
1	COLON (7), NSCLC (5), OVARIAN (3), BREAST (2), MCF7A-repro (1), MCF7D-repro (1), PROSTATE (1)
2	LEUKEMIA (6), K562A-repro (1), K562B-repro (1)
3	MELANOMA (7), BREAST (2)
4	RENAL (9), CNS (5), NSCLC (4), BREAST (3), OVARIAN (3), MELANOMA (1), PROSTATE (1), UNKNOWN (1)

K-means clustering

cluster	cancers
1	RENAL (9), MELANOMA (8), NSCLC (8), OVARIAN (6), BREAST (5), CNS (5), PROSTATE (2), UNKNOWN (1)
2	COLON (7), LEUKEMIA (6), BREAST (2), K562A-repro (1), K562B-repro (1), MCF7A-repro (1), MCF7D-repro (1), NSCLC (1)

K-means clustering

cluster	cancers
1	MELANOMA (7), BREAST (2)
2	LEUKEMIA (6), K562A-repro (1), K562B-repro (1)
3	BREAST (2), CNS (2), NSCLC (1), RENAL (1)
4	COLON (7), NSCLC (1)
5	RENAL (8), NSCLC (7), OVARIAN (6), CNS (3), PROSTATE (2), BREAST (1), MELANOMA (1), UNKNOWN (1)
6	BREAST (2), MCF7A-repro (1), MCF7D-repro (1)