Introduction to R for Biologists

Day 4 – Data analysis

*Several slides have been contributed by or modified from slides by Dr. Claus Wilke

Day 4 Outline

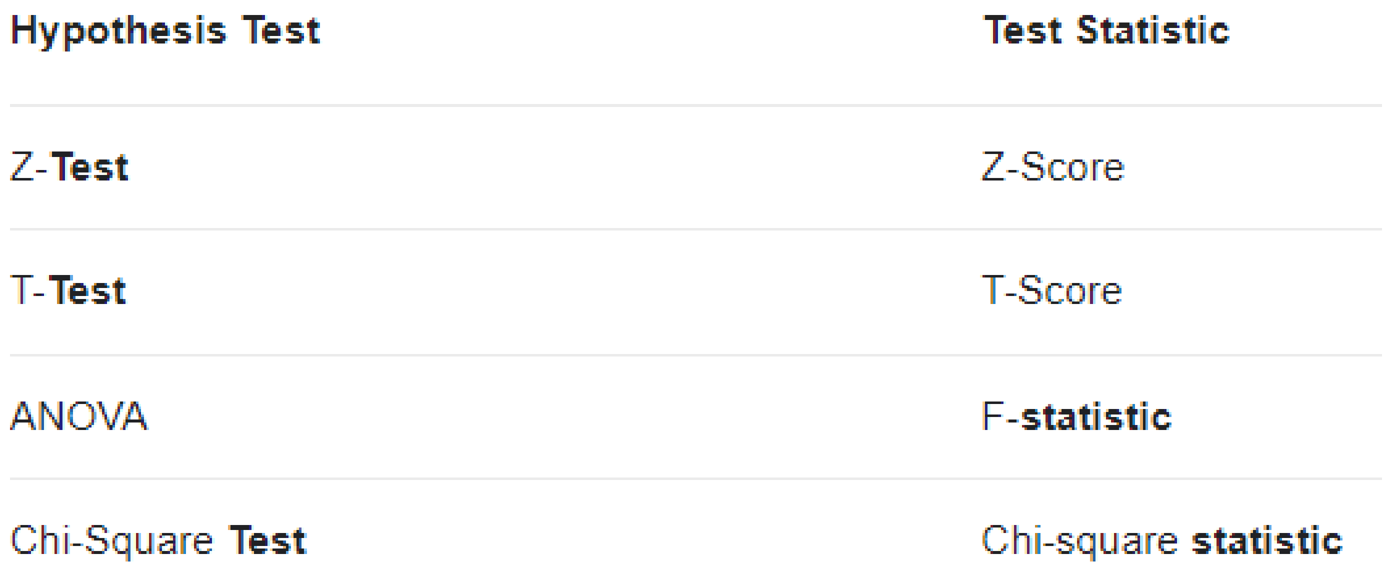
1. Hypothesis testing
   1. Test statistics
   2. p-values
   3. False discovery rate
2. Exploratory analysis
   A. Dimensionality reduction
   B. Clustering

Many, many types of statistical tests

Your choice will depend on your experiment:

http://www.biostathandbook.com/testchoice.html

But generally these are the work horses:



Hypothesis Test	**Test Statistic**
Z-**Test**	Z-Score
T-**Test**	T-Score
ANOVA	F-**statistic**
Chi-Square **Test**	Chi-square **statistic**

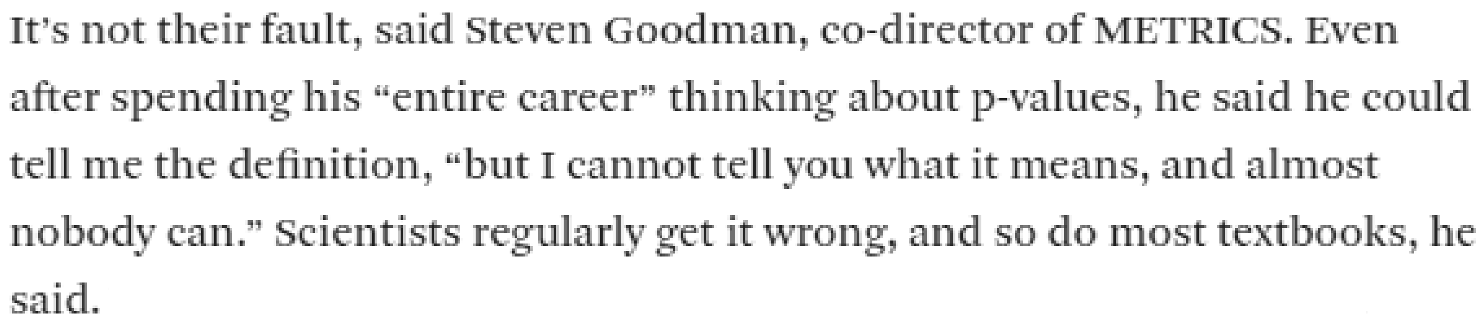
But they all follow the same pattern

1. Calculate a sample statistic (δ) from your real data
   - Mean, difference in means, median, proportion, difference in proportions, chi-squared value, etc
2. Use simulation to create a null distribution
3. Compare δ to the null distribution – how does it fit?
4. Calculate probability (p-value) that δ could exist in a null world
5. Decide if δ is statistically significant

Conceptualizing the p-value

Technical definition:

- Probability of observing the results by chance given that the null hypothesis is correct



It's not their fault, said Steven Goodman, co-director of METRICS. Even after spending his "entire career" thinking about p-values, he said he could tell me the definition, "but I cannot tell you what it means, and almost nobody can." Scientists regularly get it wrong, and so do most textbooks, he said.

Conceptualizing the p-value

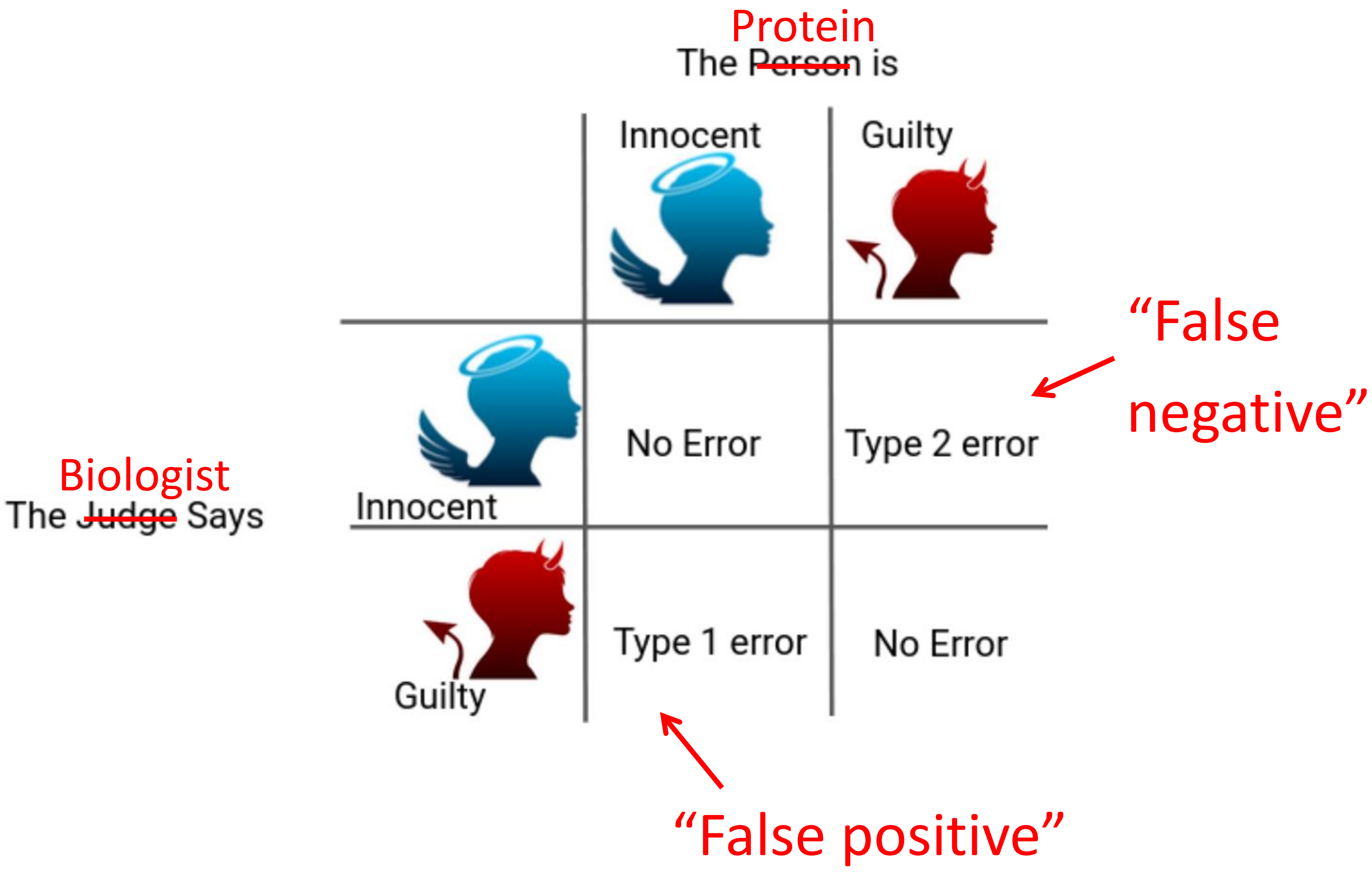
The Presumption of Innocence:

All molecules are innocent until proven guilty $\underline{\text{beyond a reasonable doubt.}}$

$\downarrow$

Significance level
or p-value cutoff

Conceptualizing the p-value

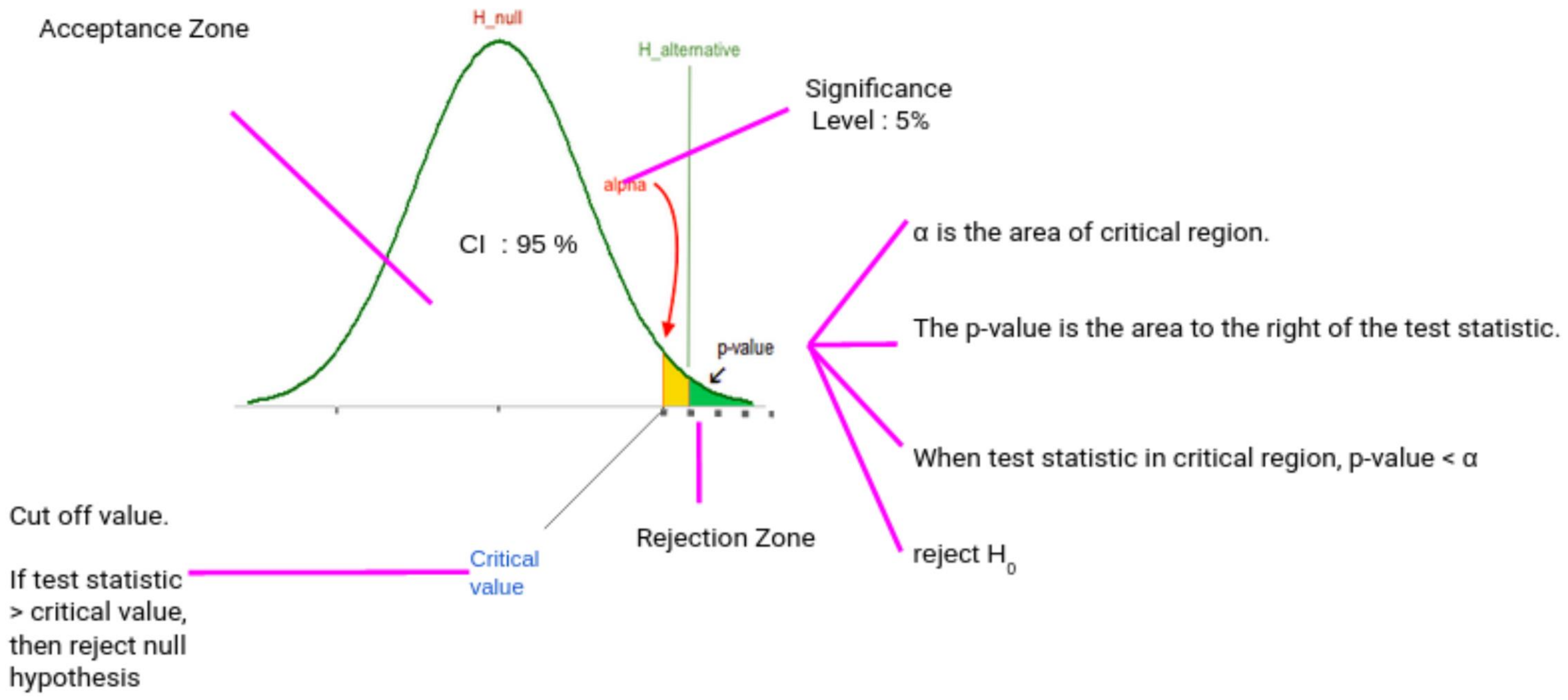


		The ~~Person~~ Protein is	
		Innocent	Guilty
The ~~Judge~~ Biologist Says	Innocent	No Error	Type 2 error
	Guilty	Type 1 error	No Error

"False negative"

"False positive"

Conceptualizing the p-value



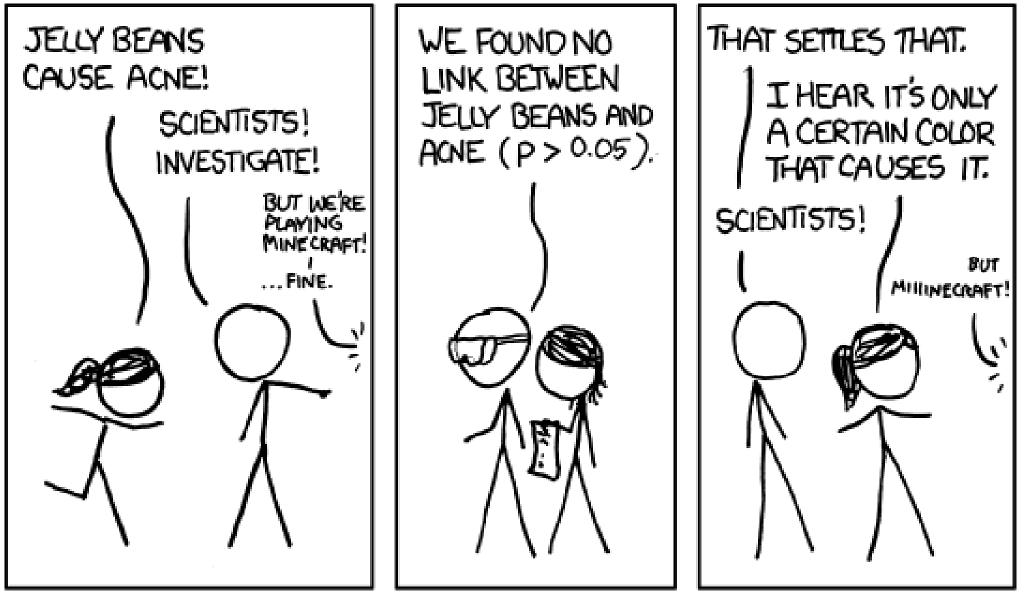
Acceptance Zone
H_null
H_alternative
Significance Level : 5%
alpha
CI : 95 %
p-value
α is the area of critical region.
The p-value is the area to the right of the test statistic.
When test statistic in critical region, p-value < α
reject H_0
Cut off value.
If test statistic > critical value, then reject null hypothesis
Critical value
Rejection Zone

How good is your evidence?

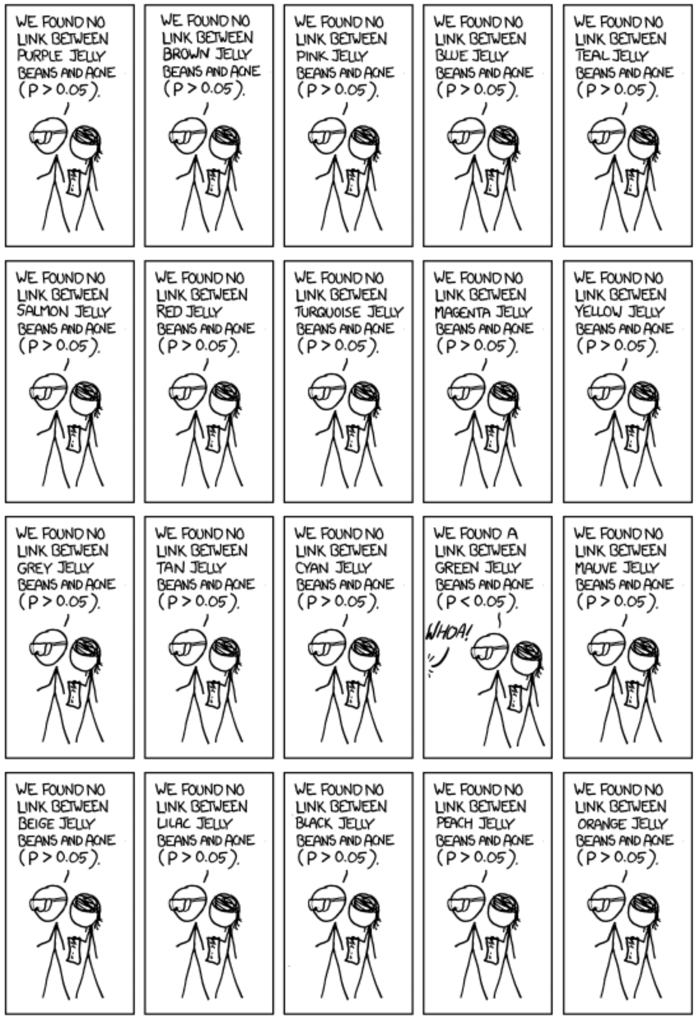
- i.e., how much statistical power does your experiment have?
- Two factors:
  - Effect size
  - Sample size

Multiple comparisons problem

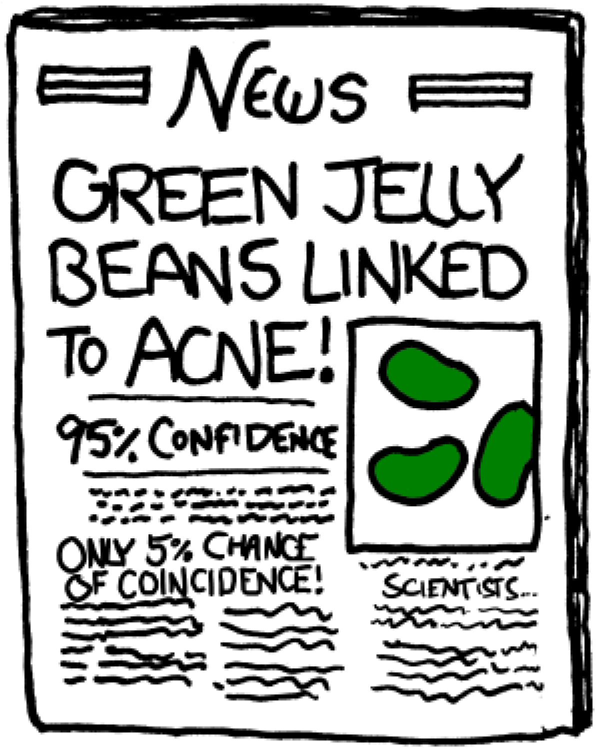
As the # of comparisons go up → probability of a false positive goes up


JELLY BEANS CAUSE ACNE!
SCIENTISTS! INVESTIGATE!
BUT WE'RE PLAYING MINECRAFT!
...FINE.
WE FOUND NO LINK BETWEEN JELLY BEANS AND ACNE (P > 0.05).
THAT SETTLES THAT.
I HEAR IT'S ONLY A CERTAIN COLOR THAT CAUSES IT.
SCIENTISTS!
BUT MIIIINECRAFT!

Multiple comparisons problem


WE FOUND NO LINK BETWEEN PURPLE JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN BROWN JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN PINK JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN BLUE JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN TEAL JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN SALMON JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN RED JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN TURQUOISE JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN MAGENTA JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN YELLOW JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN GREY JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN TAN JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN CYAN JELLY BEANS AND ACNE (P > 0.05).
WE FOUND A LINK BETWEEN GREEN JELLY BEANS AND ACNE (P < 0.05).
WHOA!
WE FOUND NO LINK BETWEEN MAUVE JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN BEIGE JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN LILAC JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN BLACK JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN PEACH JELLY BEANS AND ACNE (P > 0.05).
WE FOUND NO LINK BETWEEN ORANGE JELLY BEANS AND ACNE (P > 0.05).

So if you perform the same test a bunch of times, eventually you're going to get a false positive


NEWS
GREEN JELLY BEANS LINKED TO ACNE!
95% CONFIDENCE
ONLY 5% CHANCE OF COINCIDENCE!
SCIENTISTS...

Multiple comparisons problem

- Consider an RNA-seq differential expression experiment in HEK293 cell lines, +/- drug treatment

Multiple comparisons problem

- You measure 12,000 transcripts in both the control and treated samples

$$\downarrow$$

- You perform 12,000 statistical tests for mRNA in control vs treated (p-value <= 0.05)

$$\downarrow$$

- Probability of **at least** 1 false positive increases to essentially 100%

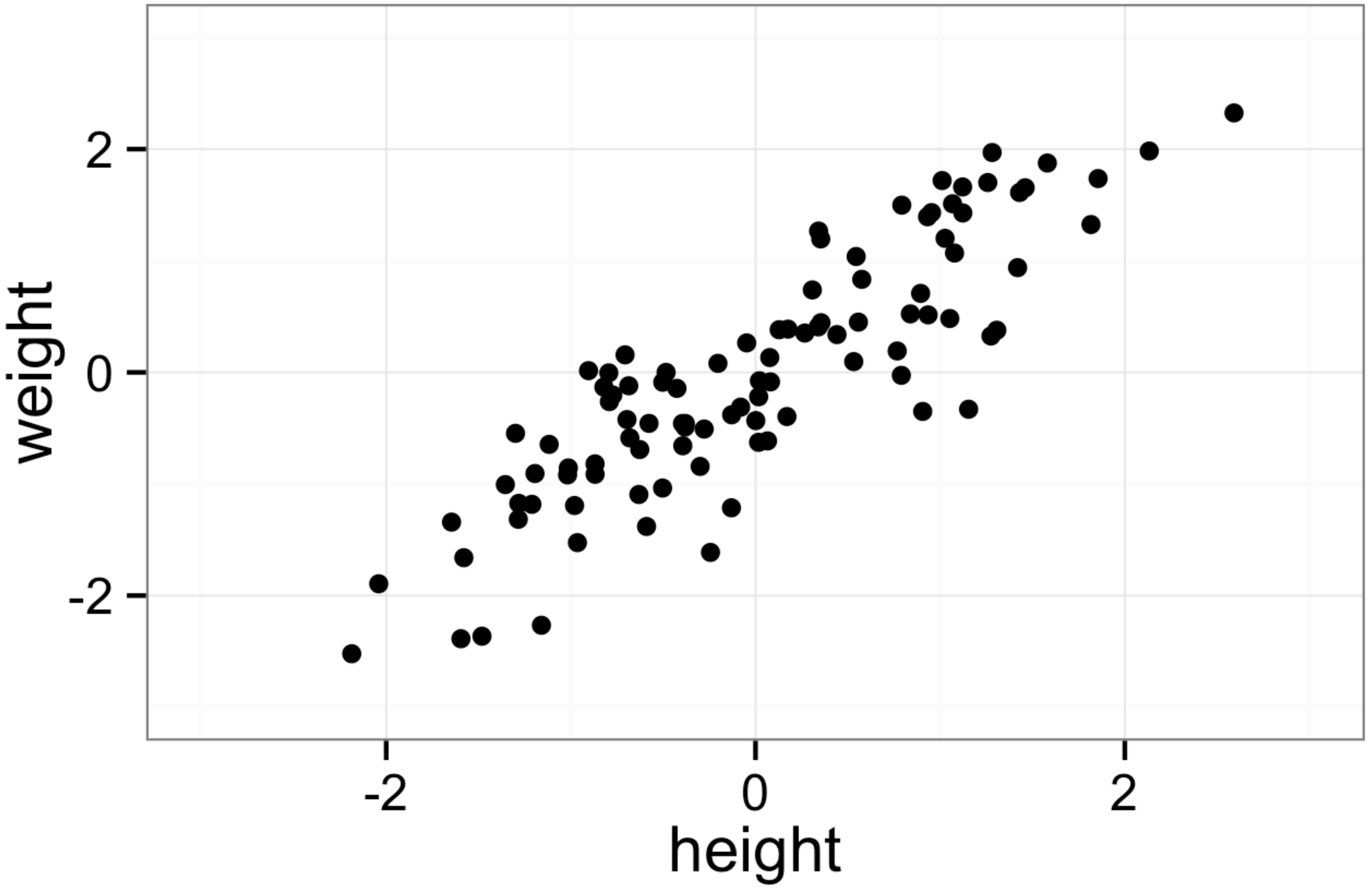
Multiple comparisons problem

- Active area of research
- No universally accepted approach
- Corrective algorithms span the extremely conservative (e.g., Bonferroni correction) to the less conservative (e.g., Benjamini-Hochberg procedure)

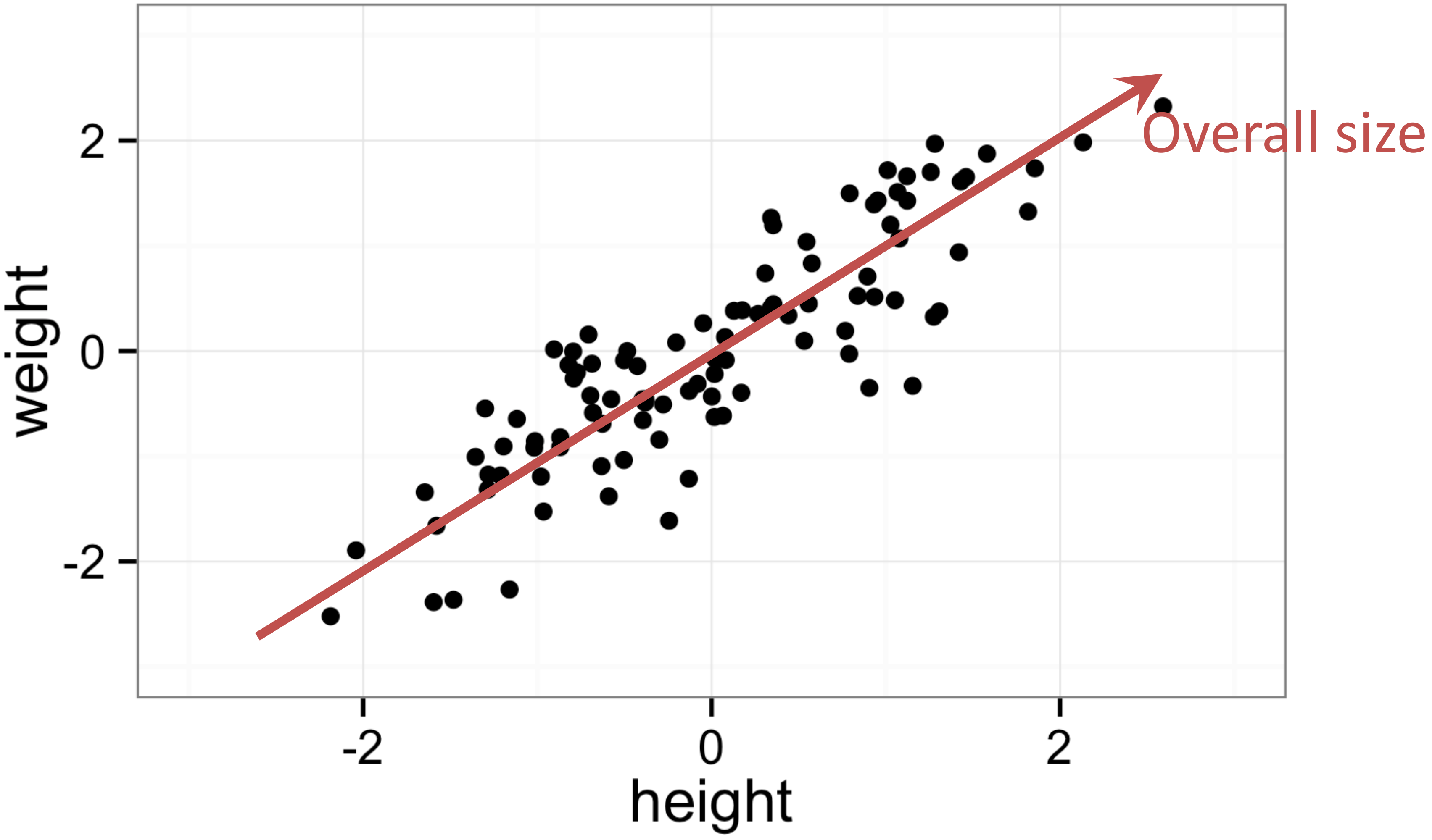
Principal Components Analysis (PCA)

- Dimension reduction
- Useful for exploratory data analysis of high-dimensional data sets.

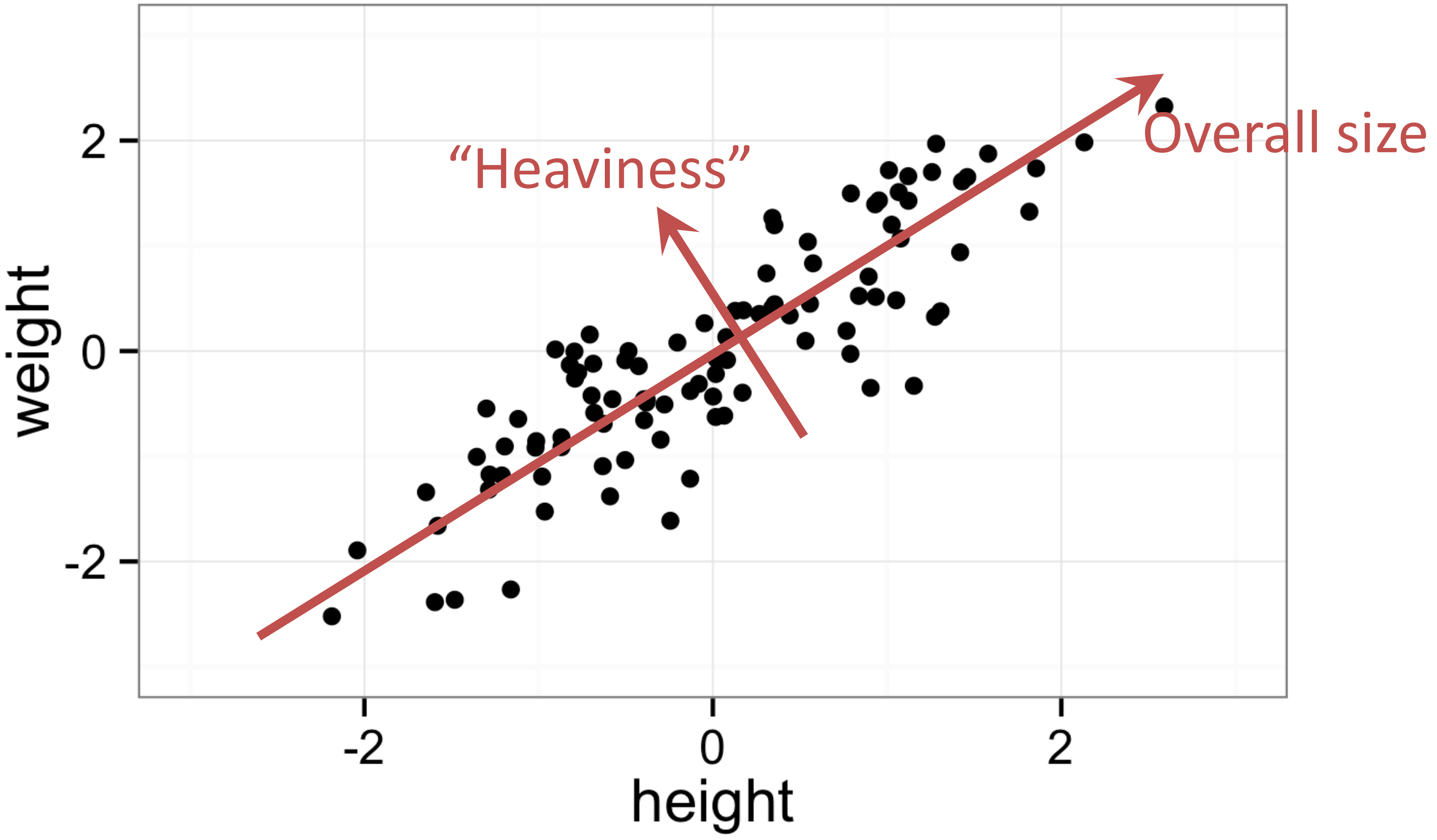
Example: Consider a data set of heights and weights of people



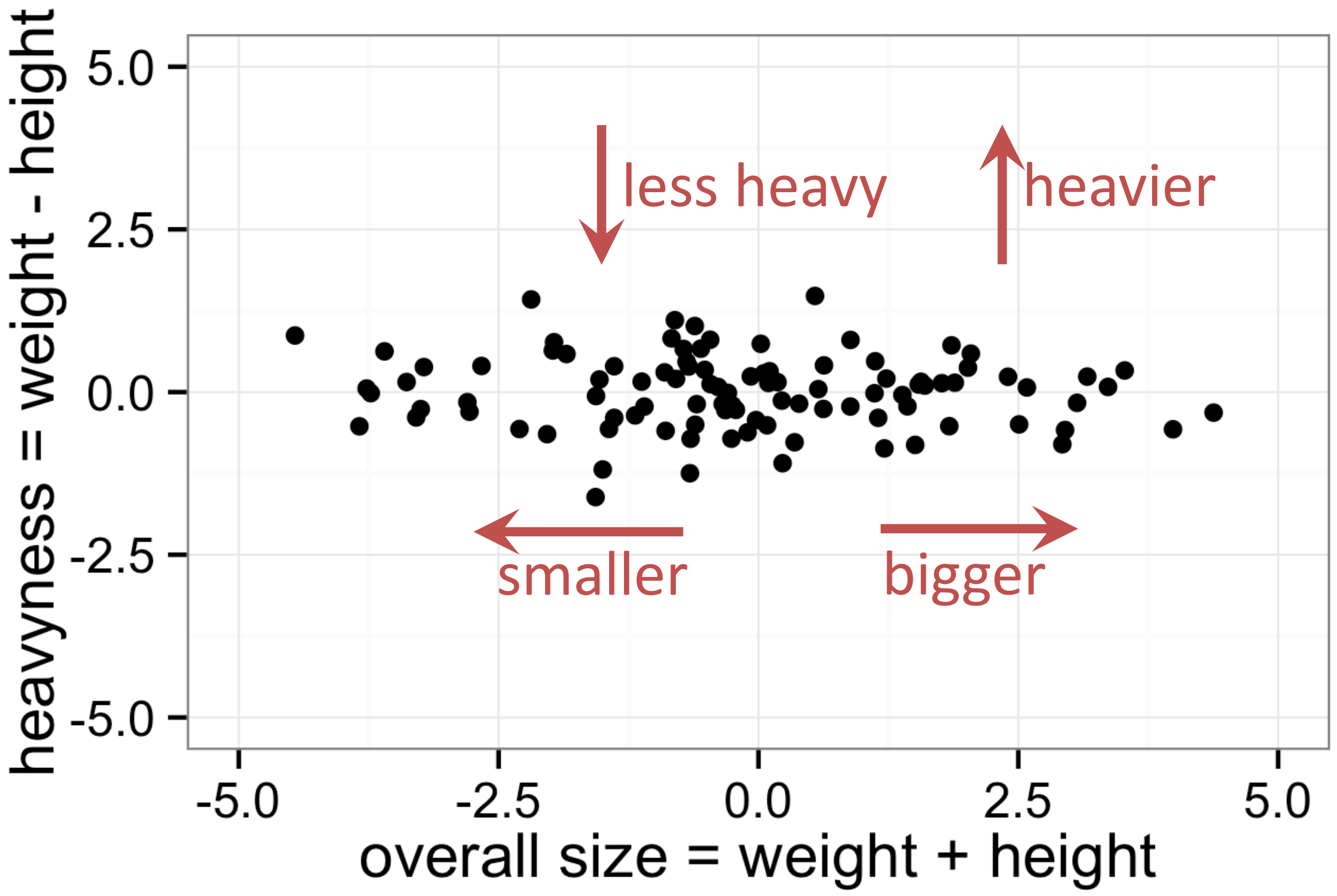
Example: Consider a data set of heights and weights of people



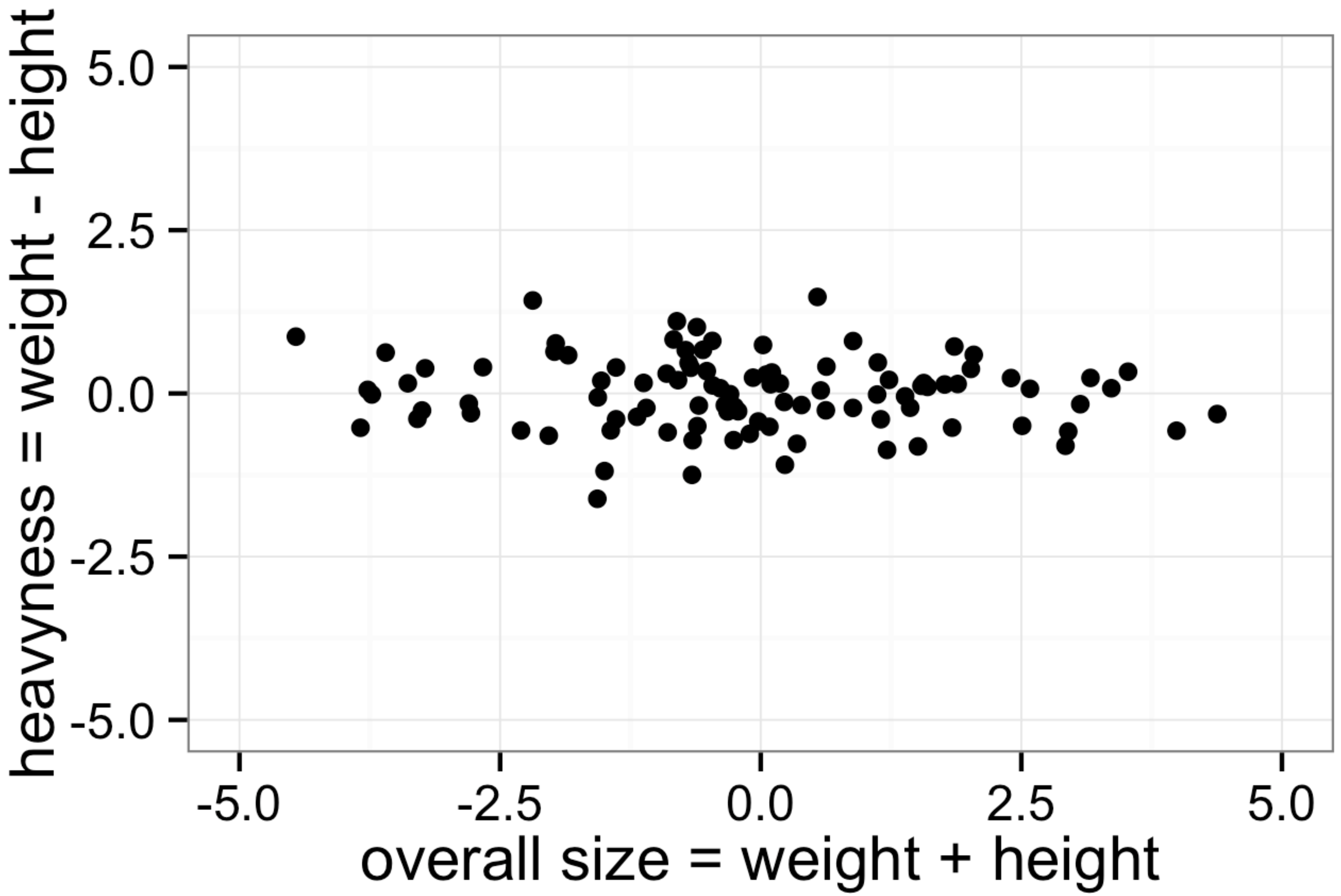
Example: Consider a data set of heights and weights of people



PCA on this data set reframes data in terms of overall size and heavyness



In our earlier example, overall size and heaviness are uncorrelated



Doing a PCA in R

```
iris %>%
     select(-Species) %>%    # remove Species column
     scale() %>%             # scale to zero mean
                             # and unit variance
     prcomp() ->             # do PCA
     pca                     # store result
                             # in variable “pca”
```

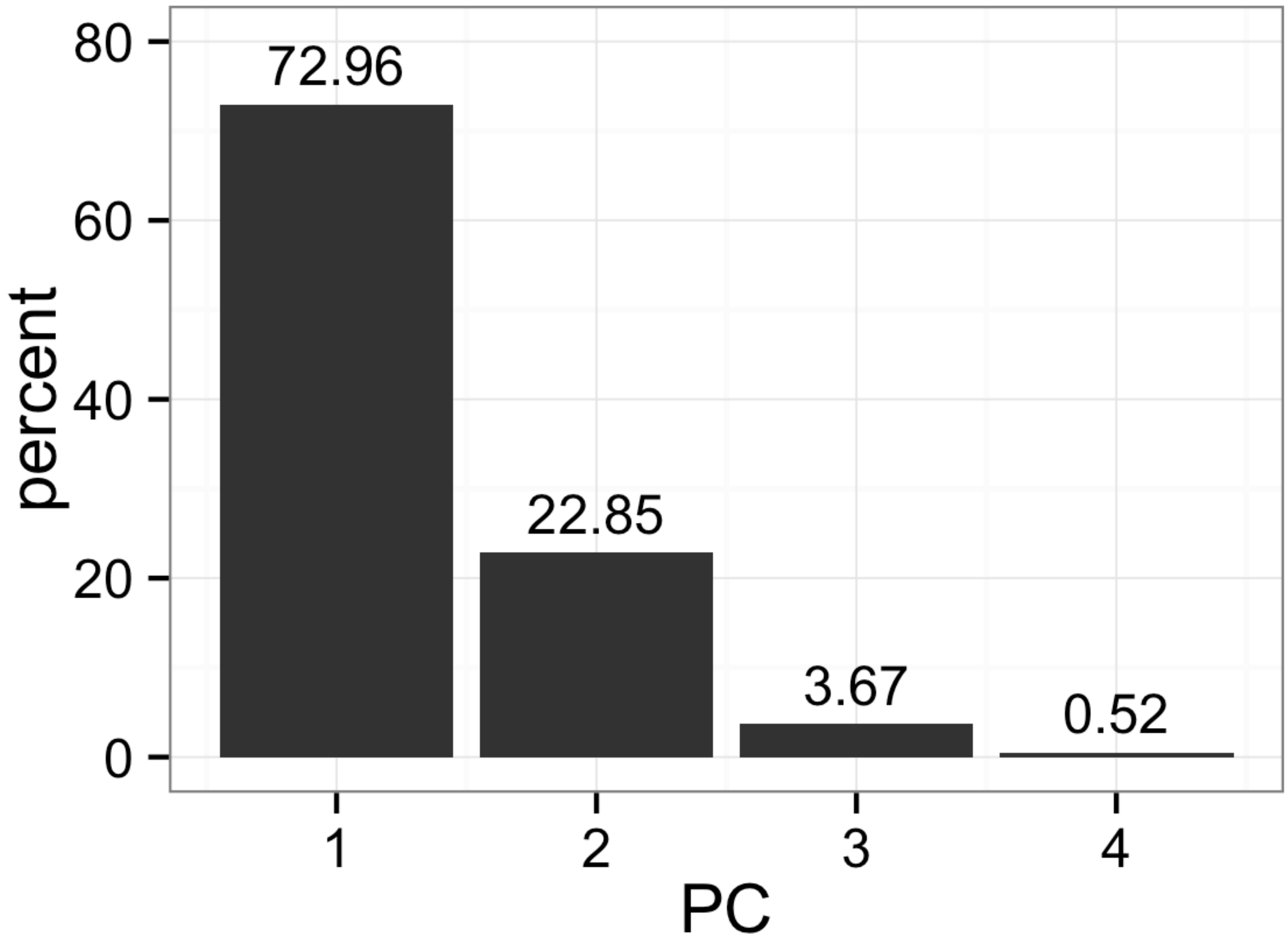
Doing a PCA in R

```
> pca
Standard deviations:
[1] 1.7083611 0.9560494 0.3830886 0.1439265

Rotation:
                    PC1         PC2        PC3        PC4
Sepal.Length  0.5210659 -0.37741762  0.7195664  0.2612863
Sepal.Width  -0.2693474 -0.92329566 -0.2443818 -0.1235096
Petal.Length  0.5804131 -0.02449161 -0.1421264 -0.8014492
Petal.Width   0.5648565 -0.06694199 -0.6342727  0.5235971
```

Doing a PCA in R

```
> pca
Standard deviations:
[1] 1.7083611 0.9560494 0.3830886 0.1439265

Rotation:
                    PC1         PC2        PC3        PC4
Sepal.Length  0.5210659 -0.37741762  0.7195664  0.2612863
Sepal.Width  -0.2693474 -0.92329566 -0.2443818 -0.1235096
Petal.Length  0.5804131 -0.02449161 -0.1421264 -0.8014492
Petal.Width   0.5648565 -0.06694199 -0.6342727  0.5235971
```

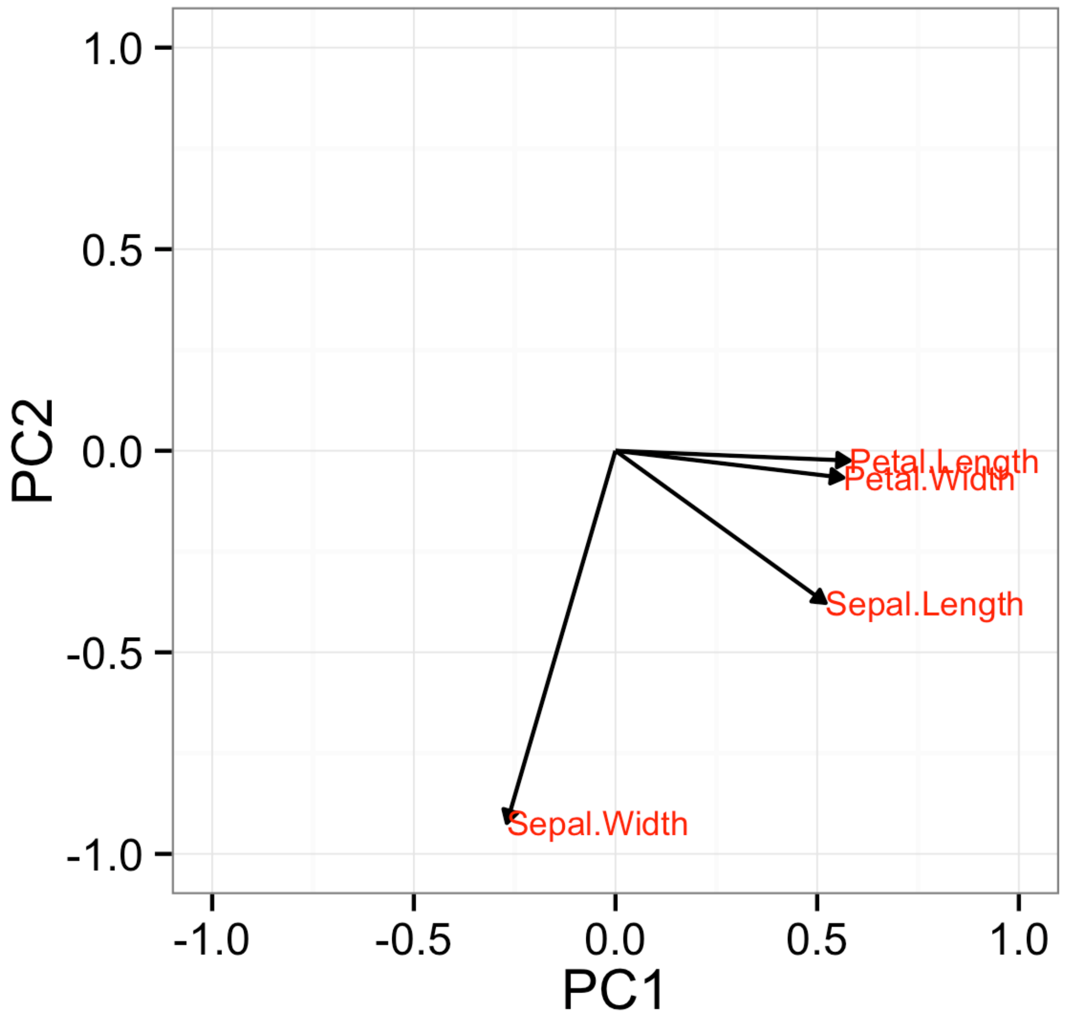
Squares of the std. devs represent the % variance explained by each PC



Doing a PCA in R

```
> pca
Standard deviations:
[1] 1.7083611 0.9560494 0.3830886 0.1439265

Rotation:
                    PC1         PC2        PC3        PC4
Sepal.Length  0.5210659 -0.37741762  0.7195664  0.2612863
Sepal.Width  -0.2693474 -0.92329566 -0.2443818 -0.1235096
Petal.Length  0.5804131 -0.02449161 -0.1421264 -0.8014492
Petal.Width   0.5648565 -0.06694199 -0.6342727  0.5235971
```

The rotation matrix tells us which variables contribute to which PCs



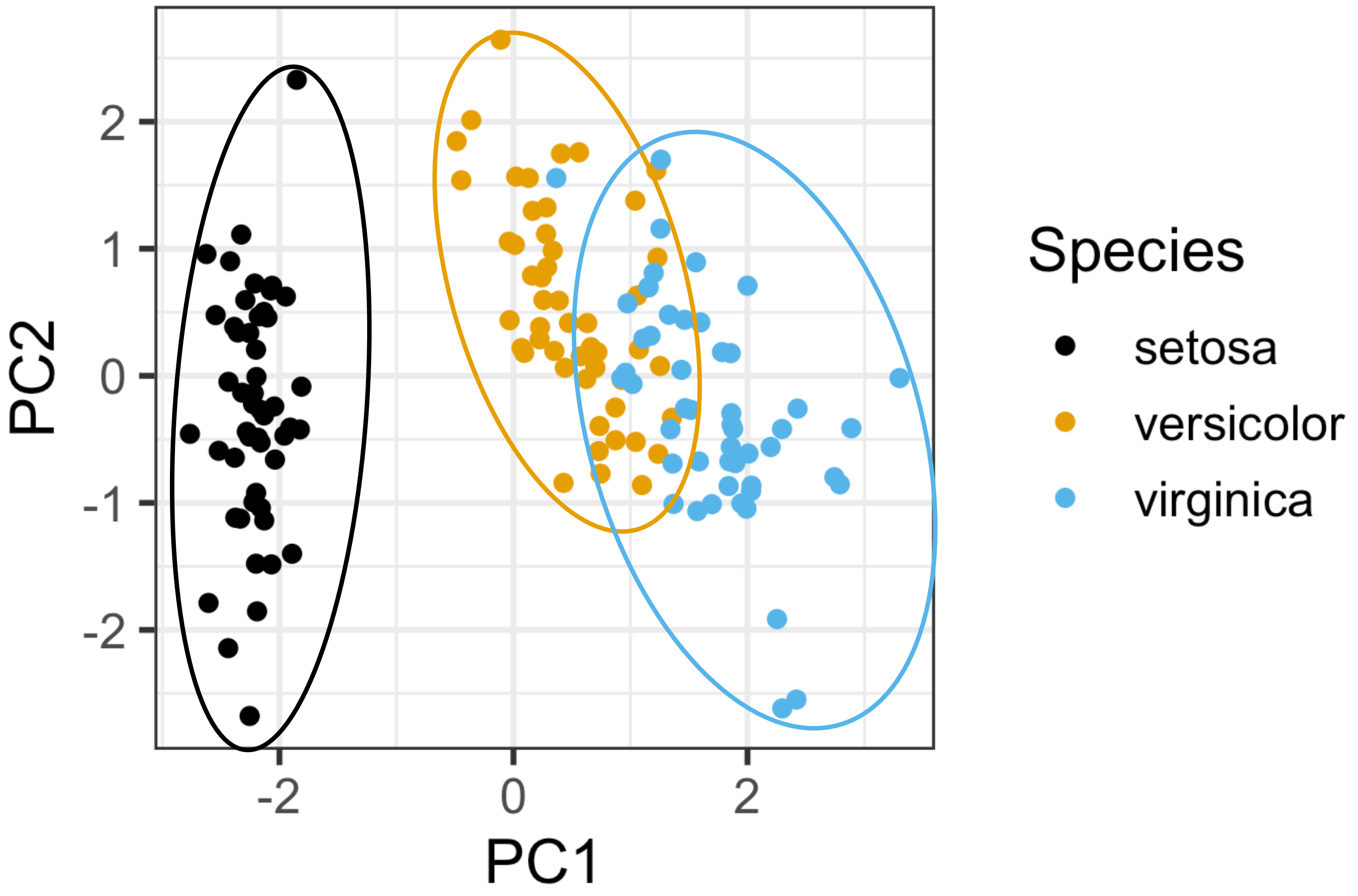
We can also recover each original observation expressed in PC coordinates

```
> pca$x
```

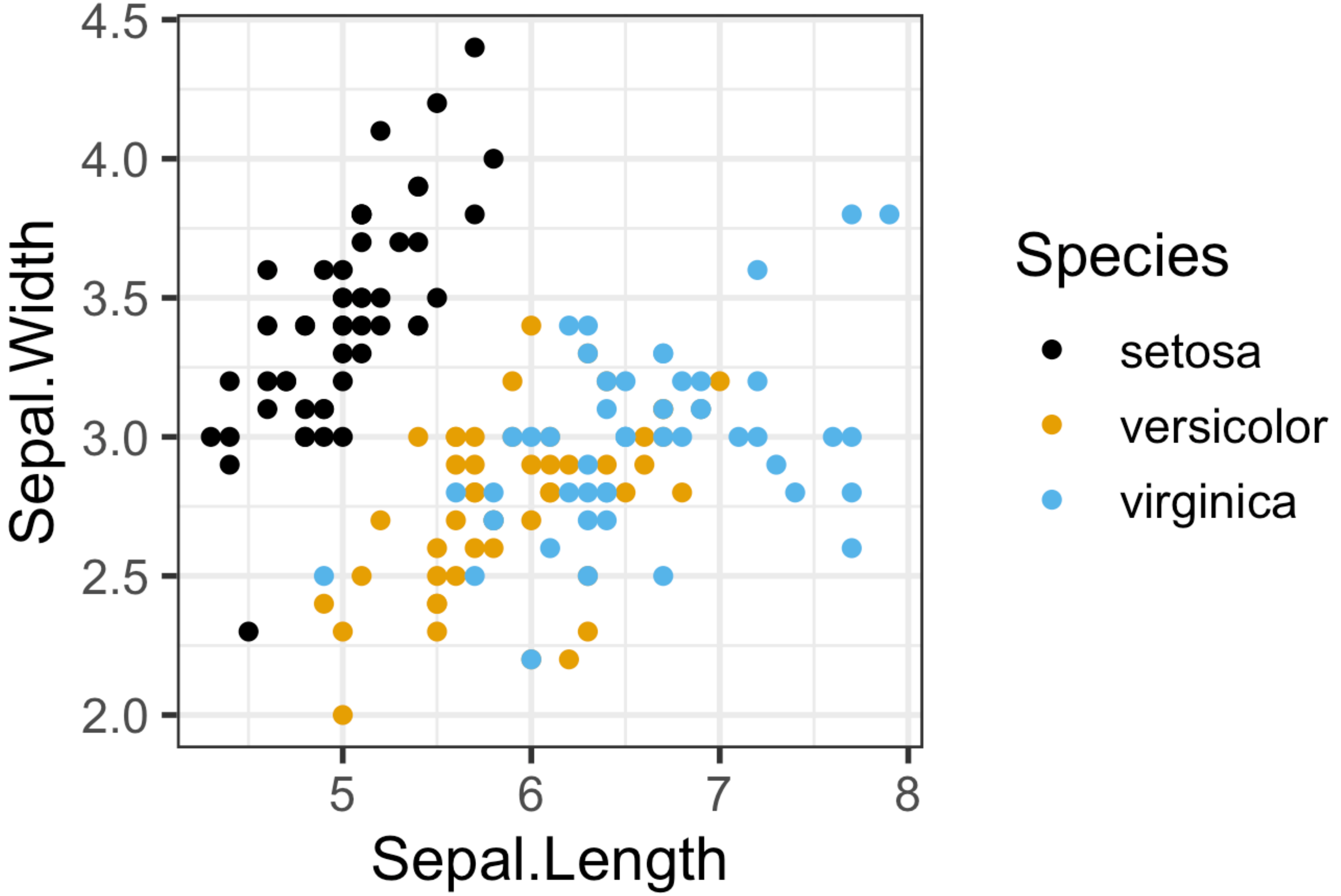
We can also recover each original observation expressed in PC coordinates

```
> pca$x
             PC1          PC2          PC3          PC4
 [1,] -2.25714118 -0.478423832  0.127279624  0.024087508
 [2,] -2.07401302  0.671882687  0.233825517  0.102662845
 [3,] -2.35633511  0.340766425 -0.044053900  0.028282305
 [4,] -2.29170679  0.595399863 -0.090985297 -0.065735340
 [5,] -2.38186270 -0.644675659 -0.015685647 -0.035802870
 [6,] -2.06870061 -1.484205297 -0.026878250  0.006586116
 [7,] -2.43586845 -0.047485118 -0.334350297 -0.036652767
 [8,] -2.22539189 -0.222403002  0.088399352 -0.024529919
 [9,] -2.32684533  1.111603700 -0.144592465 -0.026769540
[10,] -2.17703491  0.467447569  0.252918268 -0.039766068
[11,] -2.15907699 -1.040205867  0.267784001  0.016675503
[12,] -2.31836413 -0.132633999 -0.093446191 -0.133037725
[13,] -2.21104370  0.726243183  0.230140246  0.002416941
```

Plot of iris plants in PC coordinates reveals differences among species



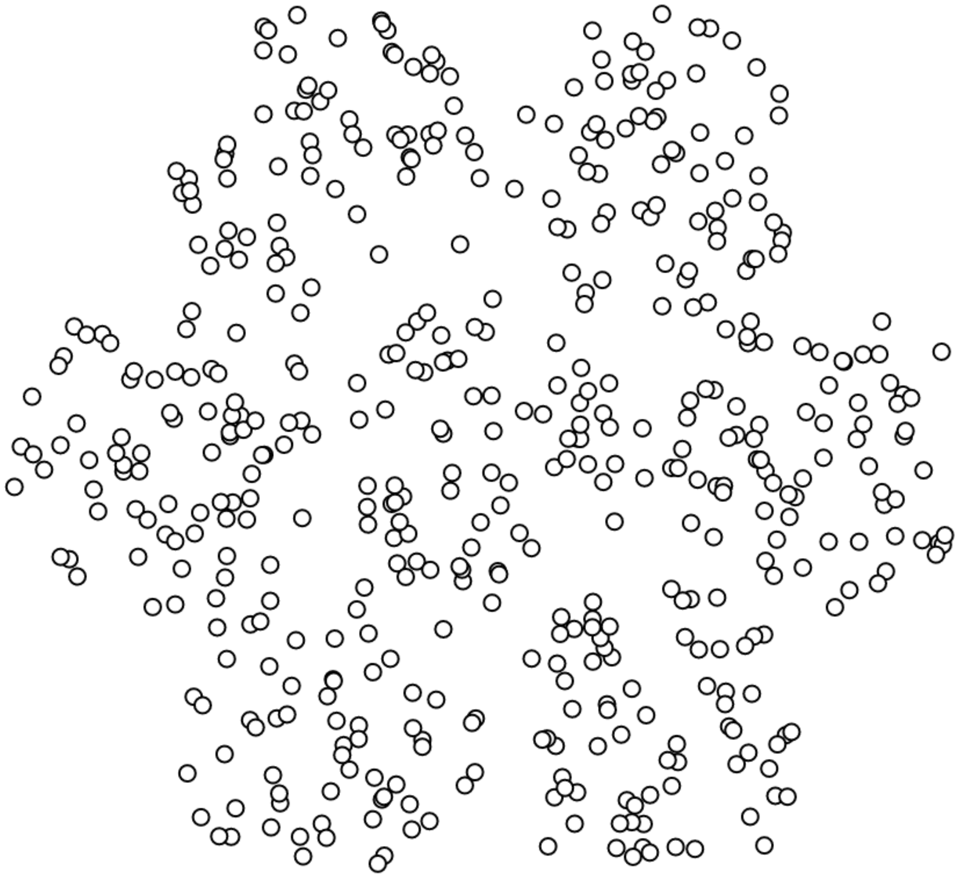
These differences are much harder to see in the original variables



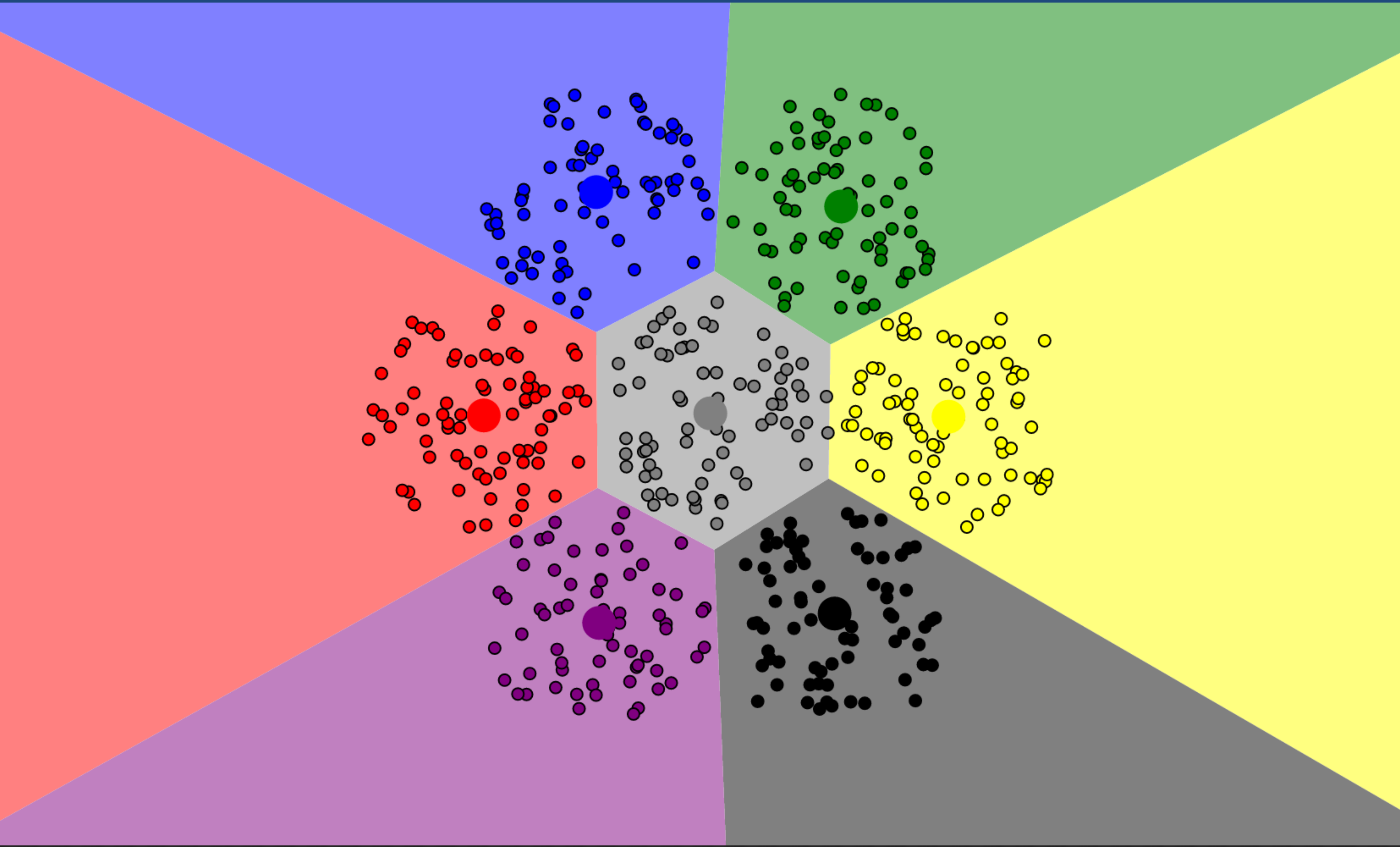
k-means clustering

Method to automatically separate data sets into distinct groups.

Clustering example



Clustering example

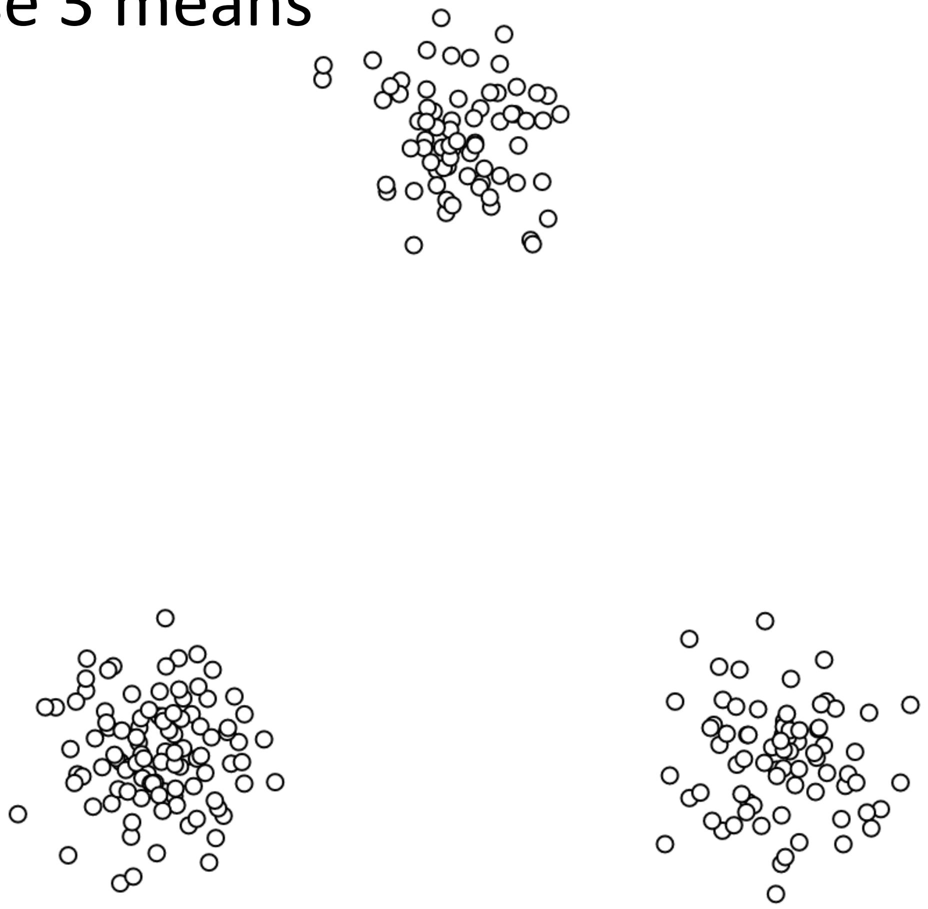


k-means clustering algorithm

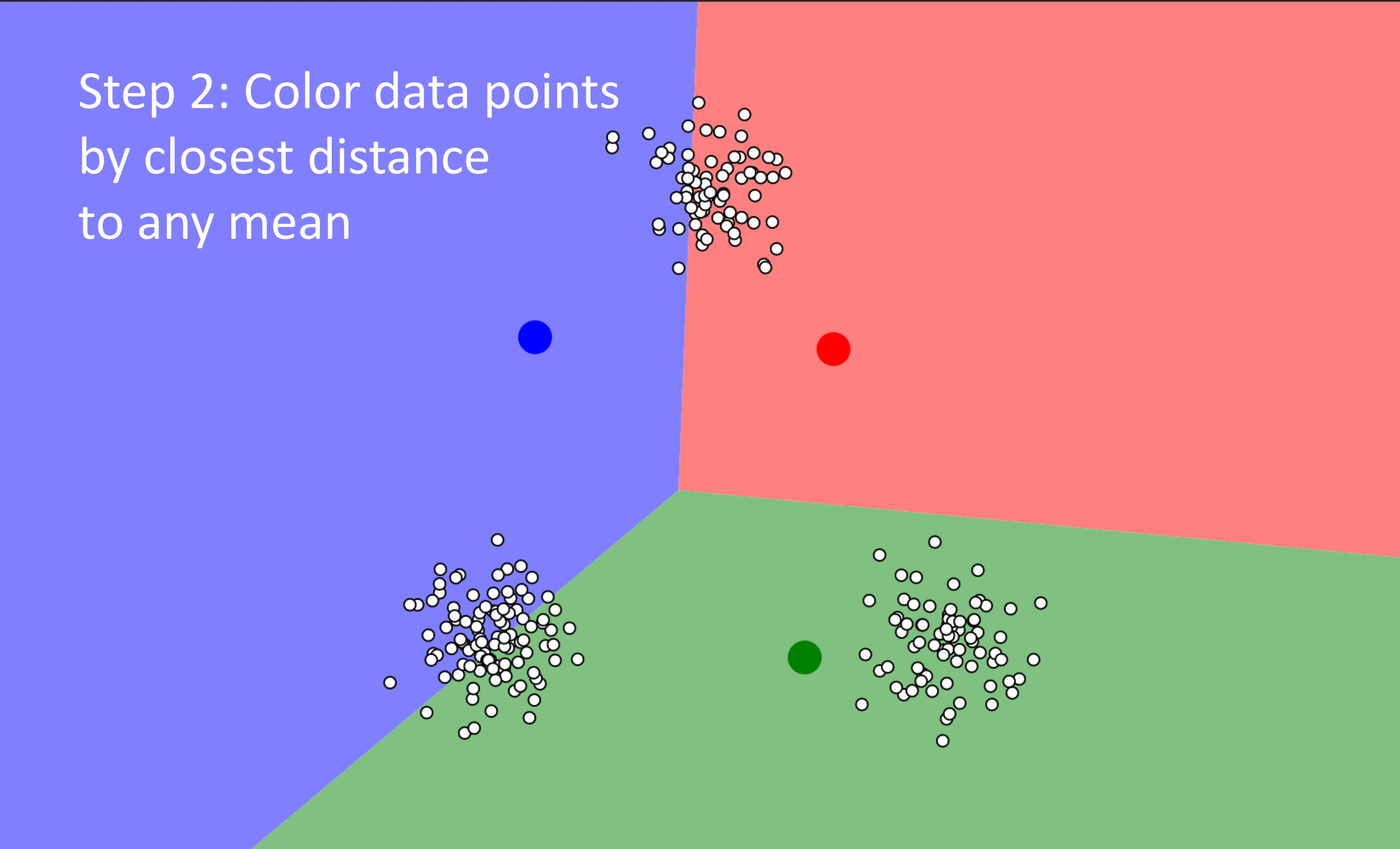
1. Start with *k* randomly chosen means
2. Color data points by the shortest distance to any mean
3. Move means to centroid position of each group of points
4. Repeat from step 2 until convergence

Algorithm example (k = 3)

Step 1: Choose 3 means at random

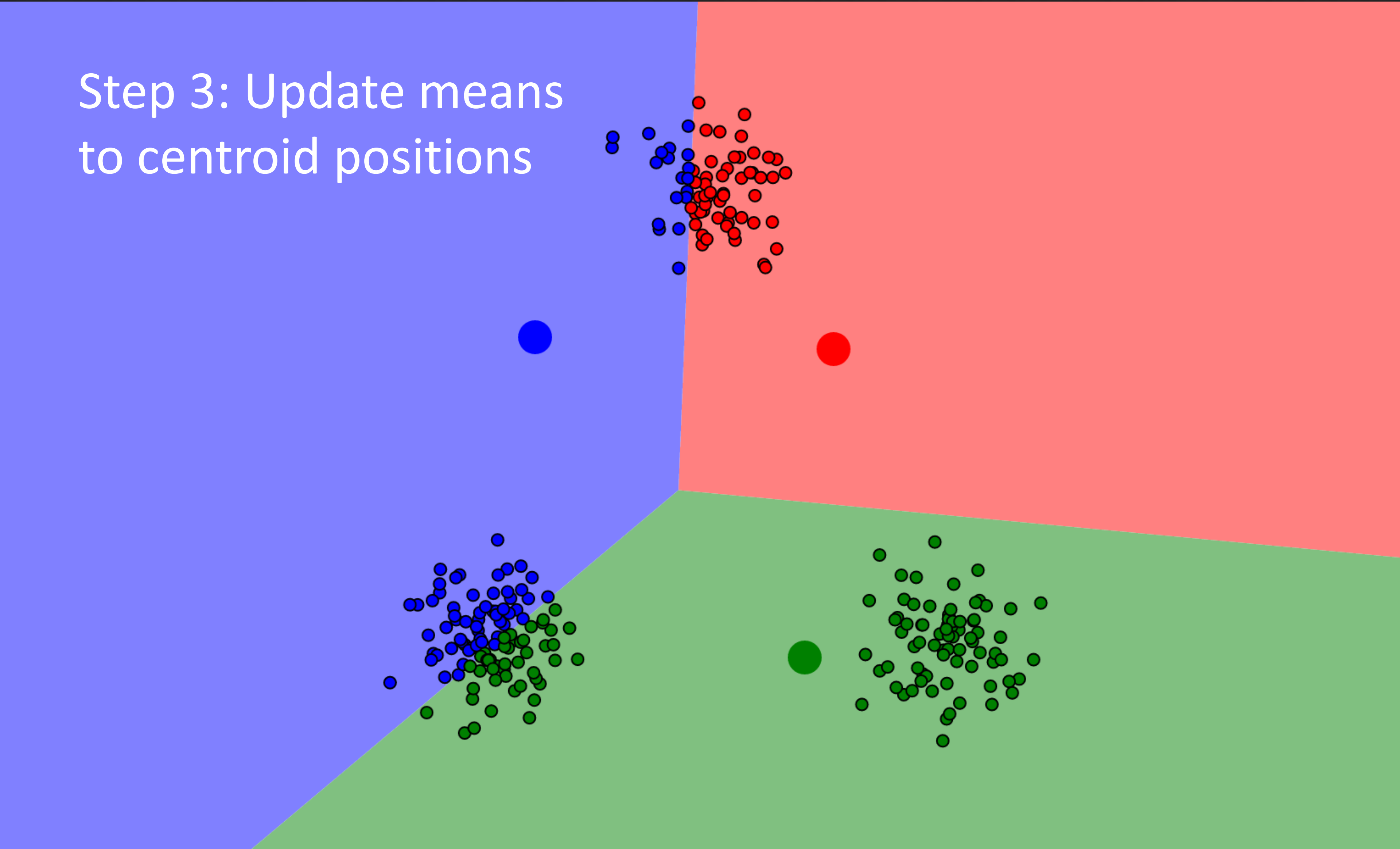


Algorithm example ($k = 3$)



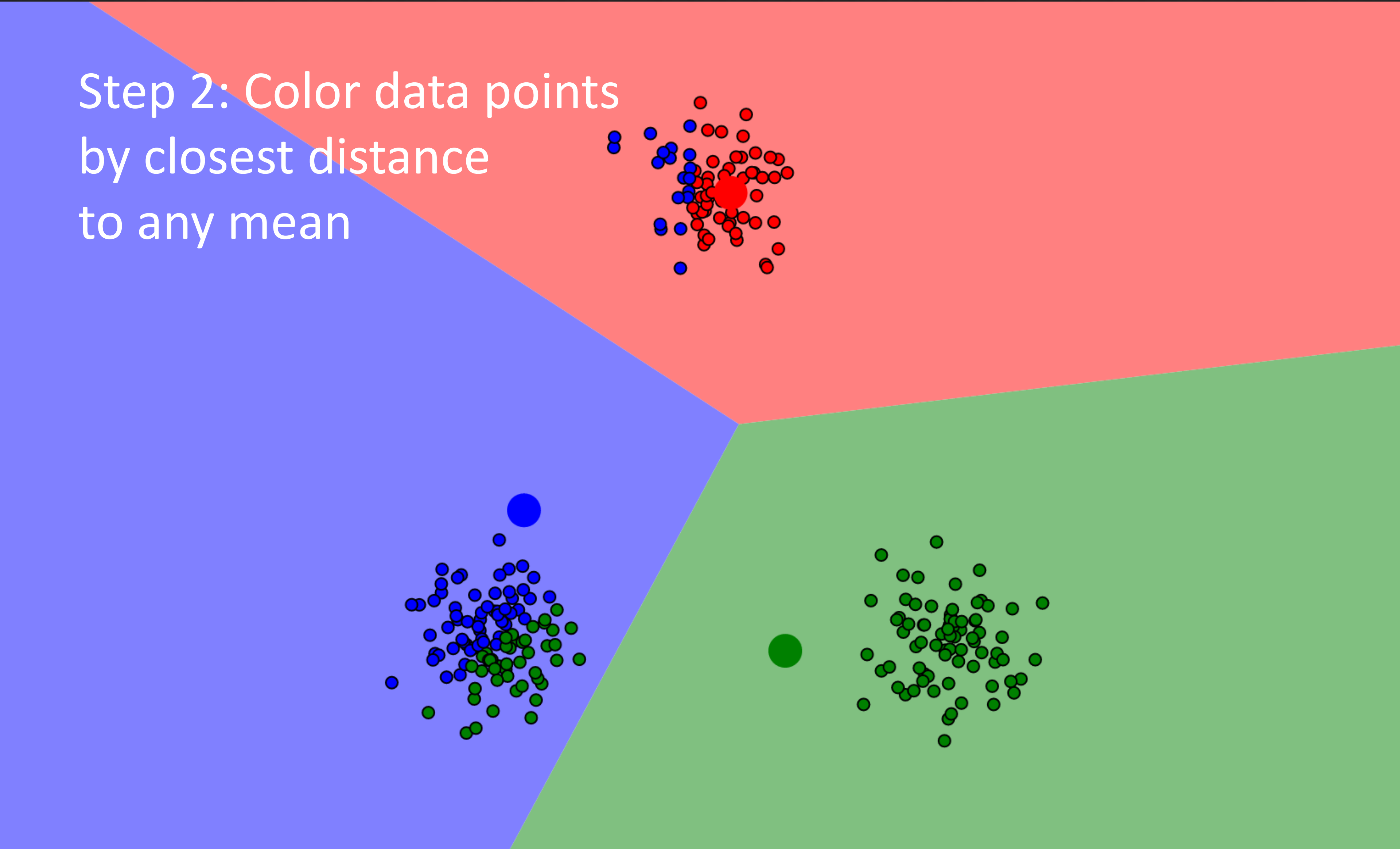
Step 2: Color data points by closest distance to any mean

Algorithm example (k = 3)



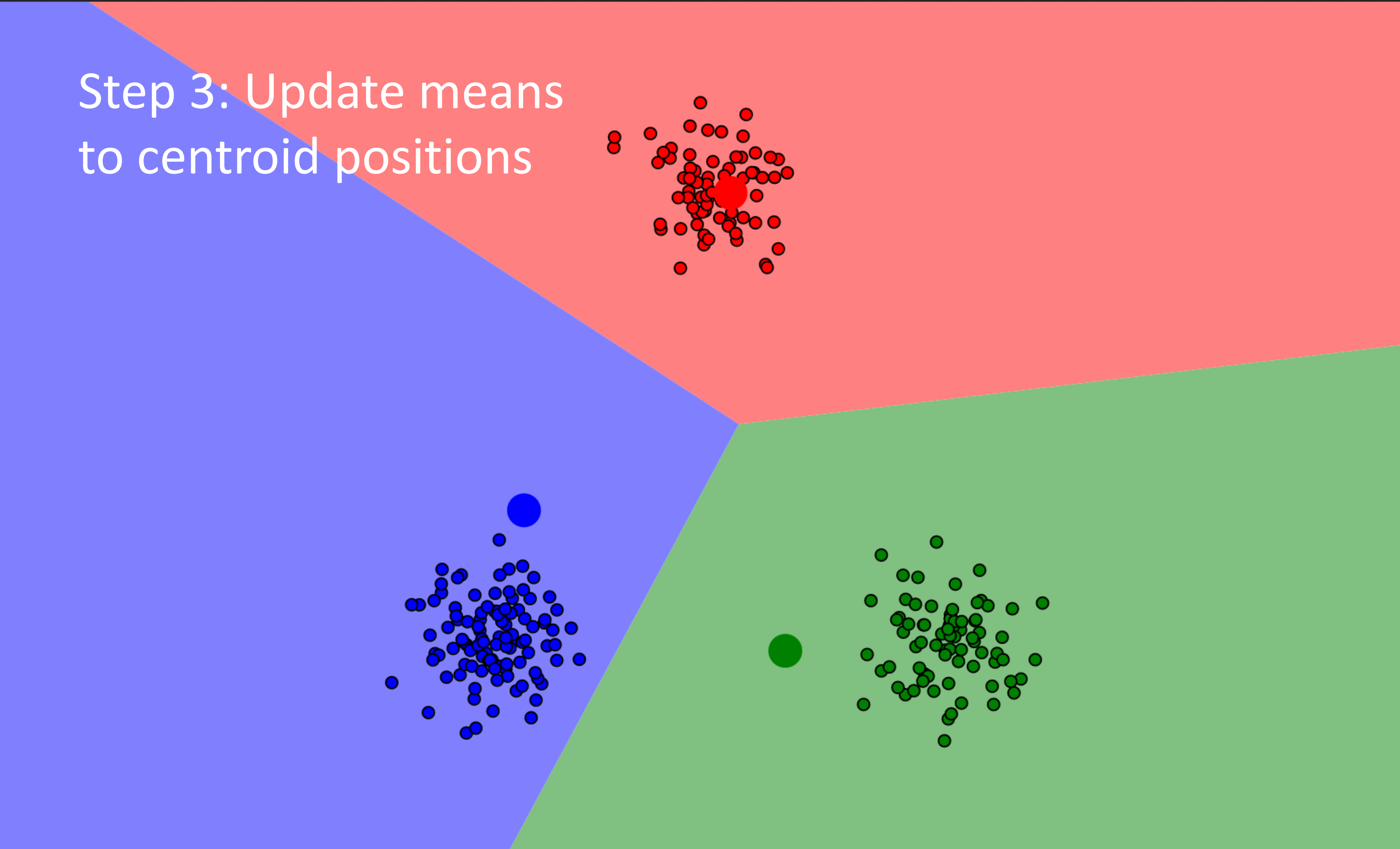
Step 3: Update means to centroid positions

Algorithm example ($k = 3$)



Step 2: Color data points by closest distance to any mean

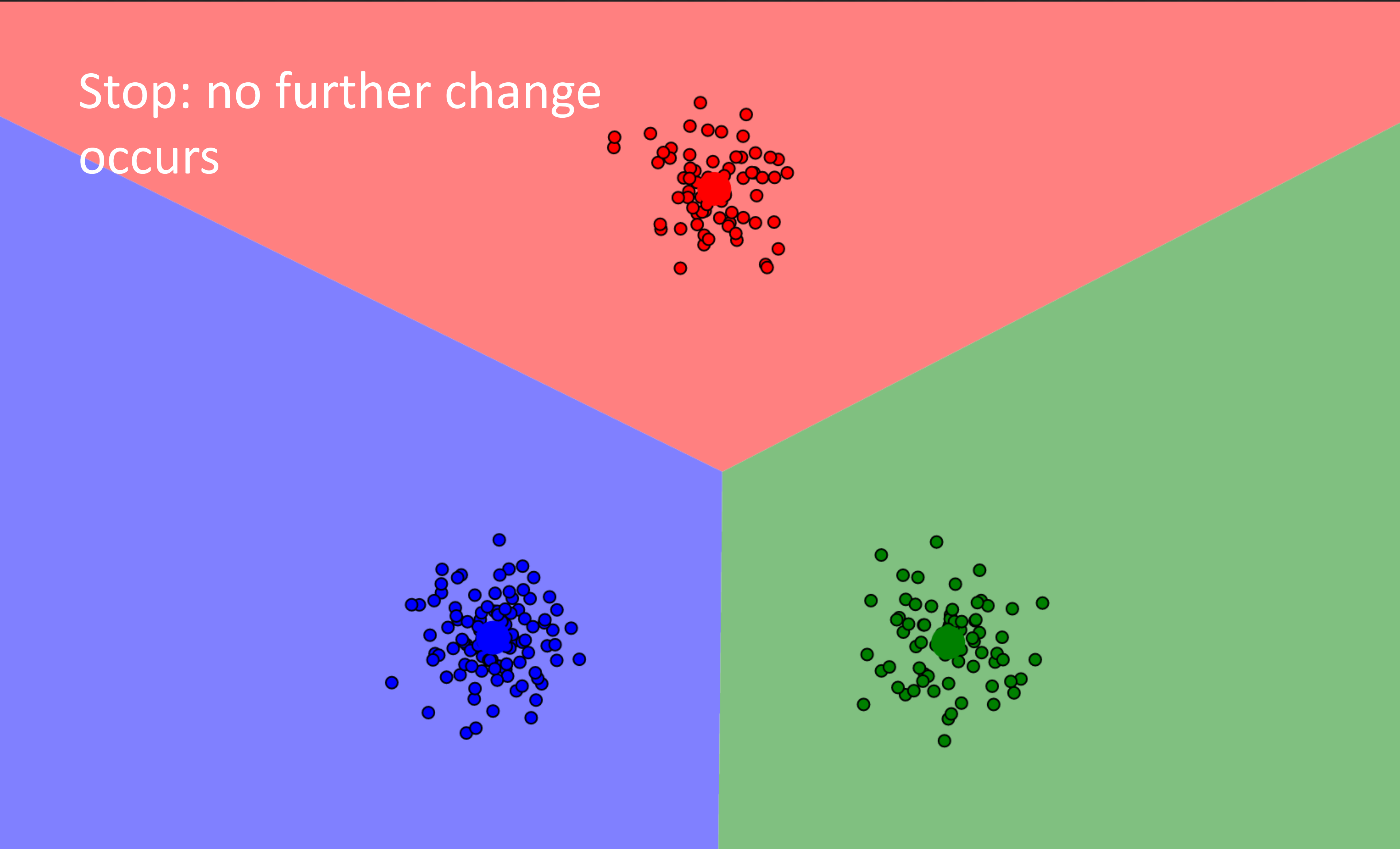
Algorithm example (k = 3)



Step 3: Update means to centroid positions

Algorithm example ($k = 3$)

Stop: no further change occurs



k-means in R (example: iris data set)

```
iris %>%
  select(-Species) %>%       # remove Species column
  kmeans(centers=3) ->       # do k-means clustering
                             # with 3 centers
  km                         # store result as "km"
```

k-means in R (example: iris data set)

```
> km
K-means clustering with 3 clusters of sizes 38, 62, 50

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     6.850000    3.073684     5.742105    2.071053
2     5.901613    2.748387     4.393548    1.433871
3     5.006000    3.428000     1.462000    0.246000

Clustering vector:
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [38] 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [75] 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1 1 2 1 1 1 1
[112] 1 1 2 2 1 1 1 1 2 1 2 1 2 1 1 2 2 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2 1 1 1 2 1
[149] 1 2

Within cluster sum of squares by cluster:
[1] 23.87947 39.82097 15.15100
 (between_SS / total_SS =  88.4 %)
```

k-means in R (example: iris data set)

```
> km
K-means clustering with 3 clusters of sizes 38, 62, 50

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     6.850000    3.073684     5.742105    2.071053
2     5.901613    2.748387     4.393548    1.433871
3     5.006000    3.428000     1.462000    0.246000

Clustering vector:
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [38] 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [75] 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1 1 2 1 1 1 1
[112] 1 1 2 2 1 1 1 1 2 1 2 1 2 1 1 2 2 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2 1 1 1 2 1
[149] 1 2

Within cluster sum of squares by cluster:
[1] 23.87947 39.82097 15.15100
 (between_SS / total_SS =  88.4 %)
```

Cluster means: the location of the final centroids

k-means in R (example: iris data set)

```
> km
K-means clustering with 3 clusters of sizes 38, 62, 50

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     6.850000    3.073684     5.742105    2.071053
2     5.901613    2.748387     4.393548    1.433871
3     5.006000    3.428000     1.462000    0.246000

Clustering vector:
   [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
  [38] 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
  [75] 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1 1 2 1 1 1 1
 [112] 1 1 2 2 1 1 1 1 2 1 2 1 2 1 1 2 2 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2 1 1 1 2 1
 [149] 1 2

Within cluster sum of squares by cluster:
[1] 23.87947 39.82097 15.15100
 (between_SS / total_SS =  88.4 %)
```

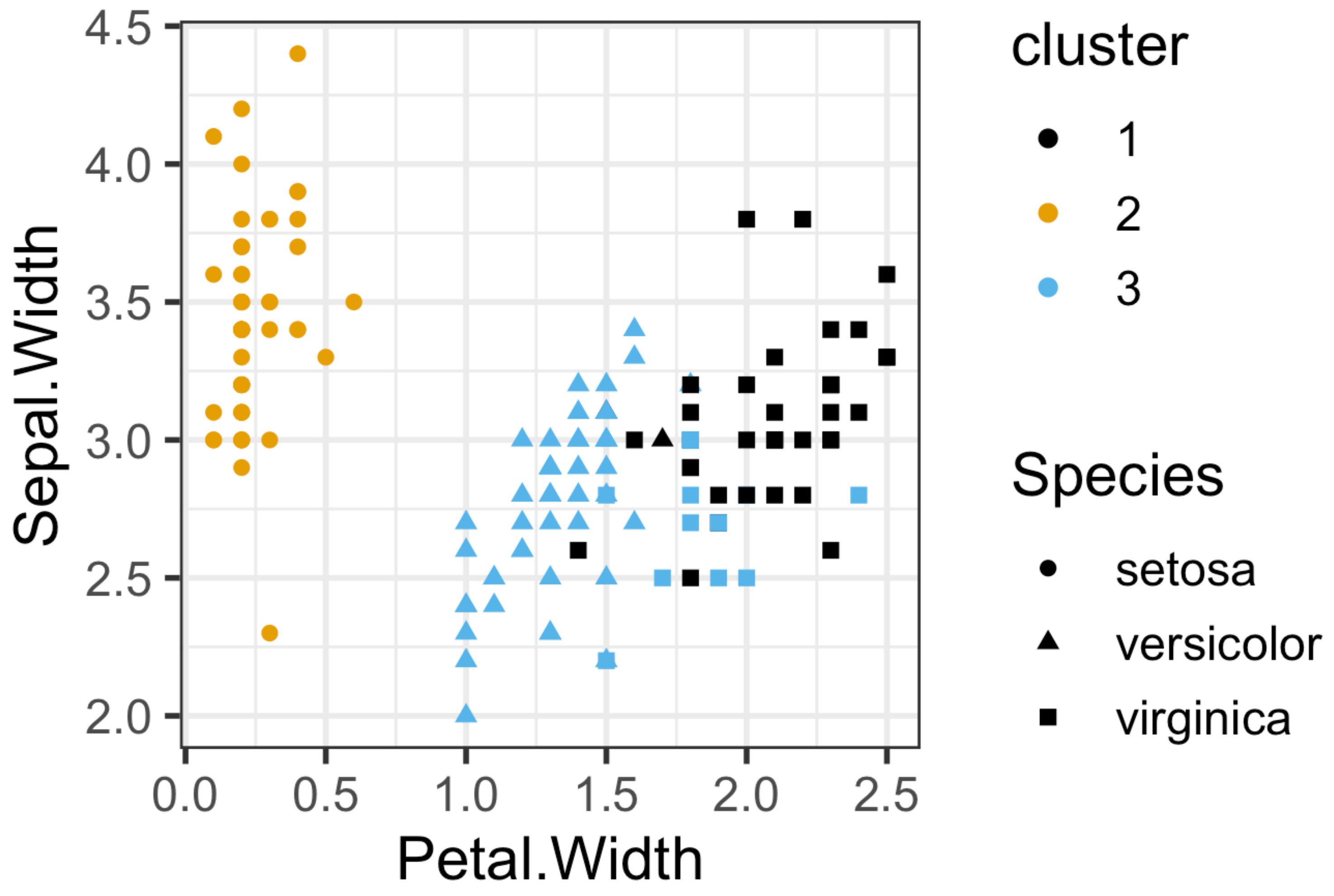
Clustering vector: provides the cluster to which each observation belongs

k-means in R (example: iris data set)

```
> km
K-means clustering with 3 clusters of sizes 38, 62, 50

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     6.850000    3.073684     5.742105    2.071053
2     5.901613    2.748387     4.393548    1.433871
3     5.006000    3.428000     1.462000    0.246000

Clustering vector:
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [38] 3 3 3 3 3 3 3 3 3 3 3 3 3 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [75] 2 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1 1 2 1 1 1 1
[112] 1 1 2 2 1 1 1 1 2 1 2 1 2 1 1 2 2 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2 1 1 1 2 1
[149] 1 2

Within cluster sum of squares by cluster:
[1] 23.87947 39.82097 15.15100
 (between_SS / total_SS =  88.4 %)
```

Within cluster sum of squares: measures quality of the clustering (lower is better)

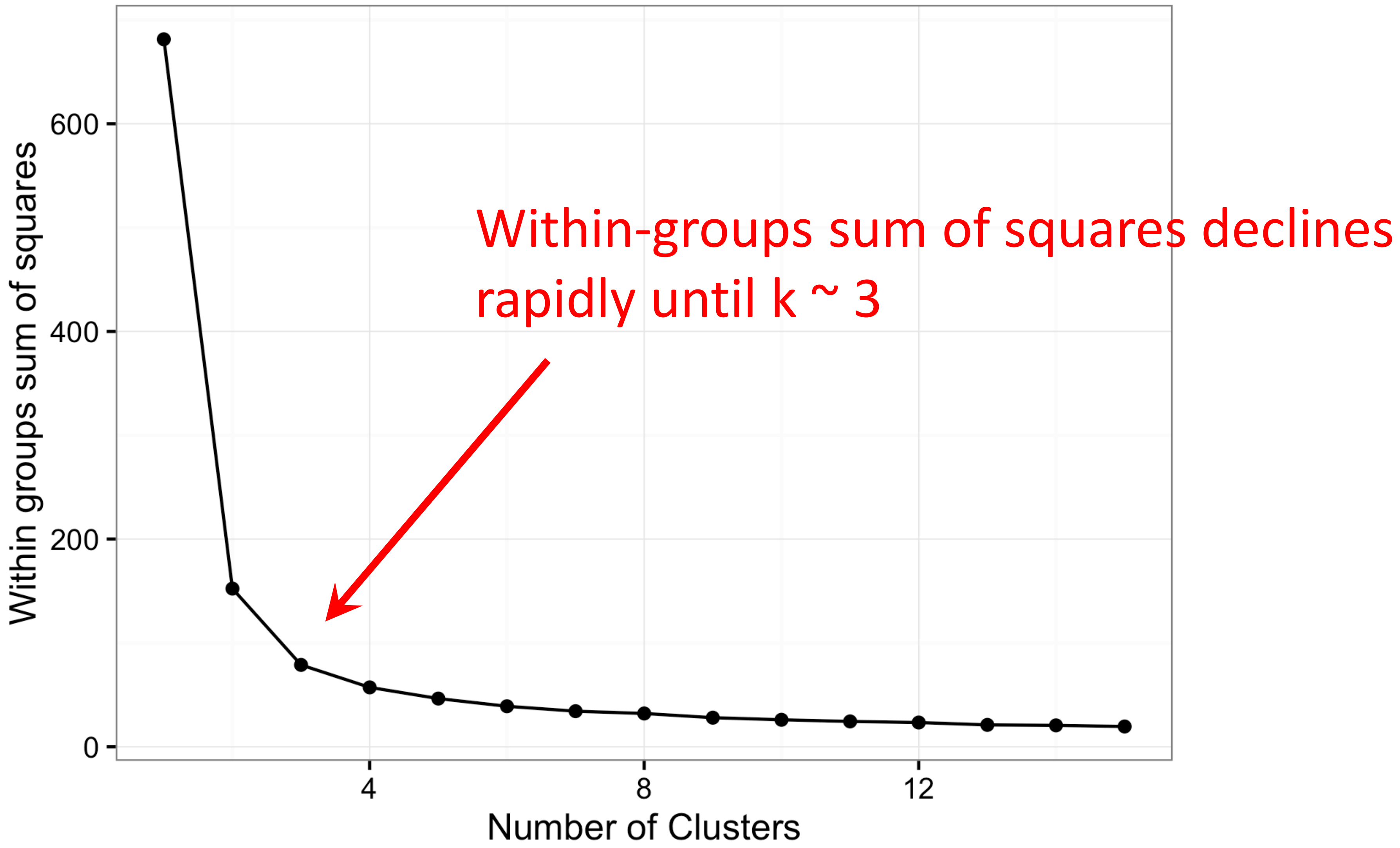
The clusters mostly but not exactly recapitulate the species assignments



How do we determine the right number of means *k*?

- Many different methods, see e.g.: http://stackoverflow.com/a/15376462/4975218
- Simplest: plot within-sum-of-squares against *k*

A bend in within-sum-of-squares indicates the ideal number of clusters


Within-groups sum of squares declines rapidly until k ~ 3
Within groups sum of squares
Number of Clusters