K-Means Clustering: in R!

Courtney Miller

K-Means Clustering: in R!

Courtney Miller

Example code at
https://github.com/Courtney-E-Miller/K-MeansClusteringPresentation

The 3 kinds of ML

- Supervised - predictions/inference on labeled data
  - (i.e., regression or classification)
- Unsupervised - finding structure in unlabeled data
- Reinforcement - learning through feedback in artificial environments

Supervised Learning

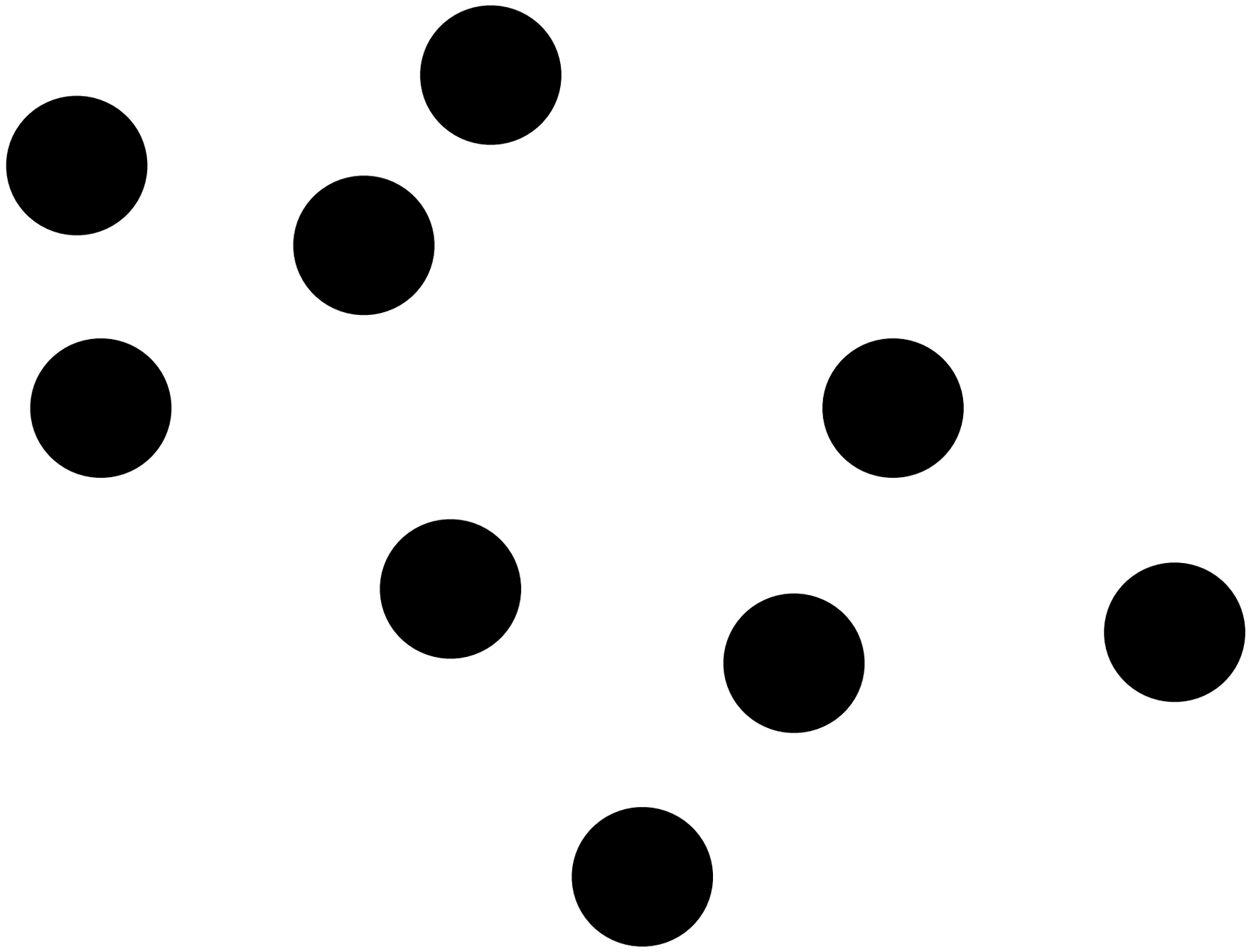
- Goal 1: clustering data

- Goal 2: finding patterns in data

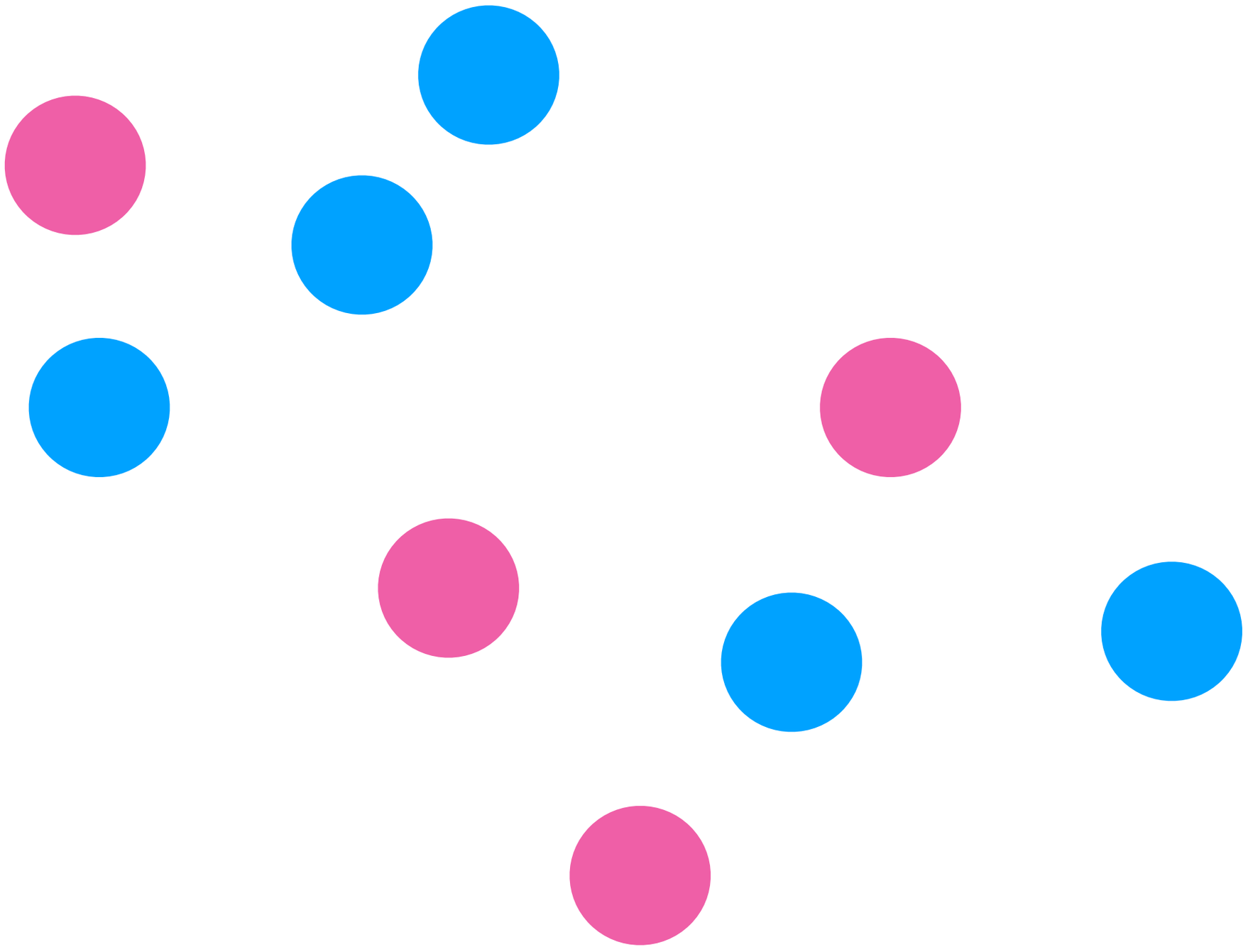
K-Means Clustering

- Assign each point to a cluster at random
- Calculate the center points for each cluster
- Reassign each point to the cluster whose center is closest
    - Repeat (2) and (3) until none of the points change their cluster assignment

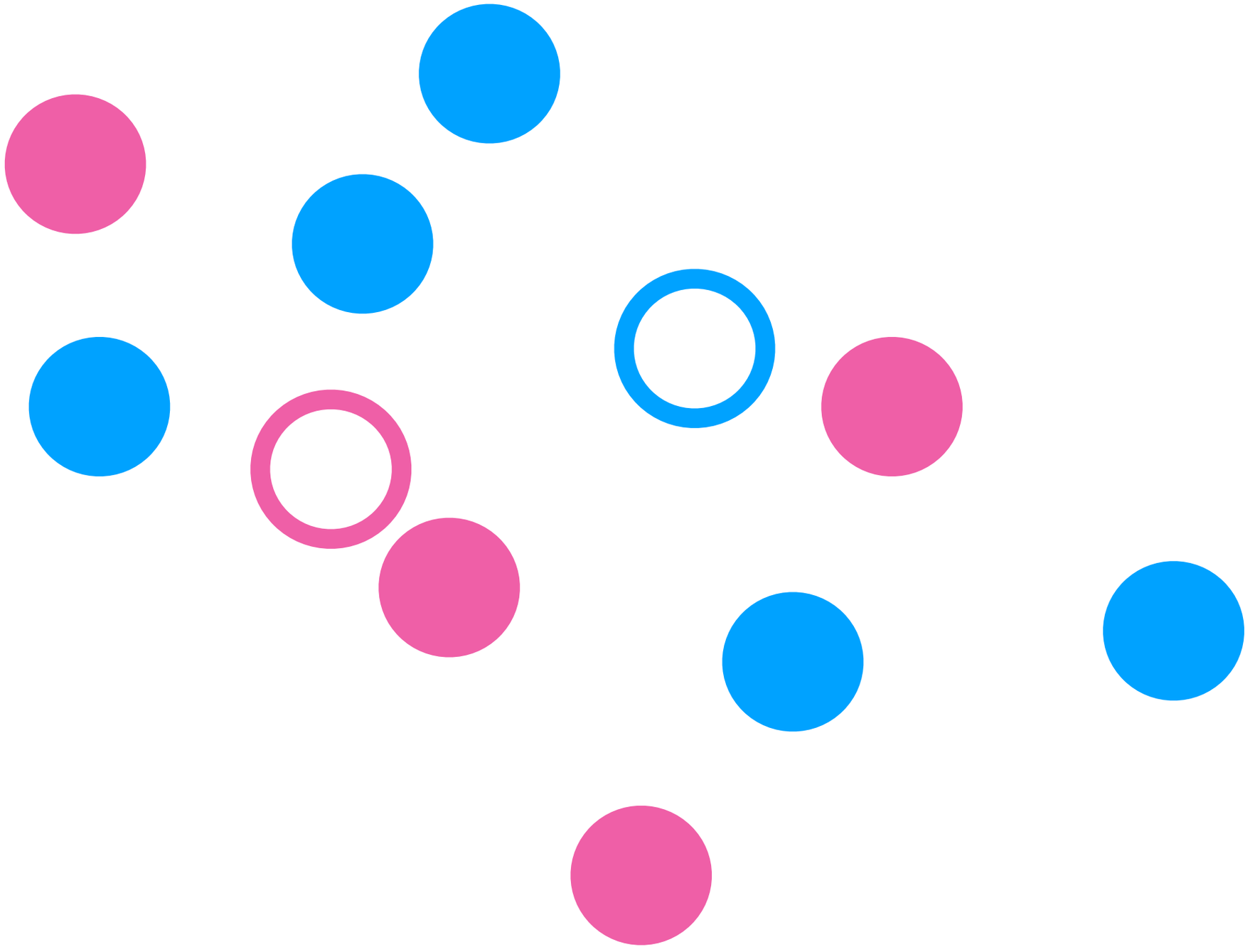
K-Means Clustering



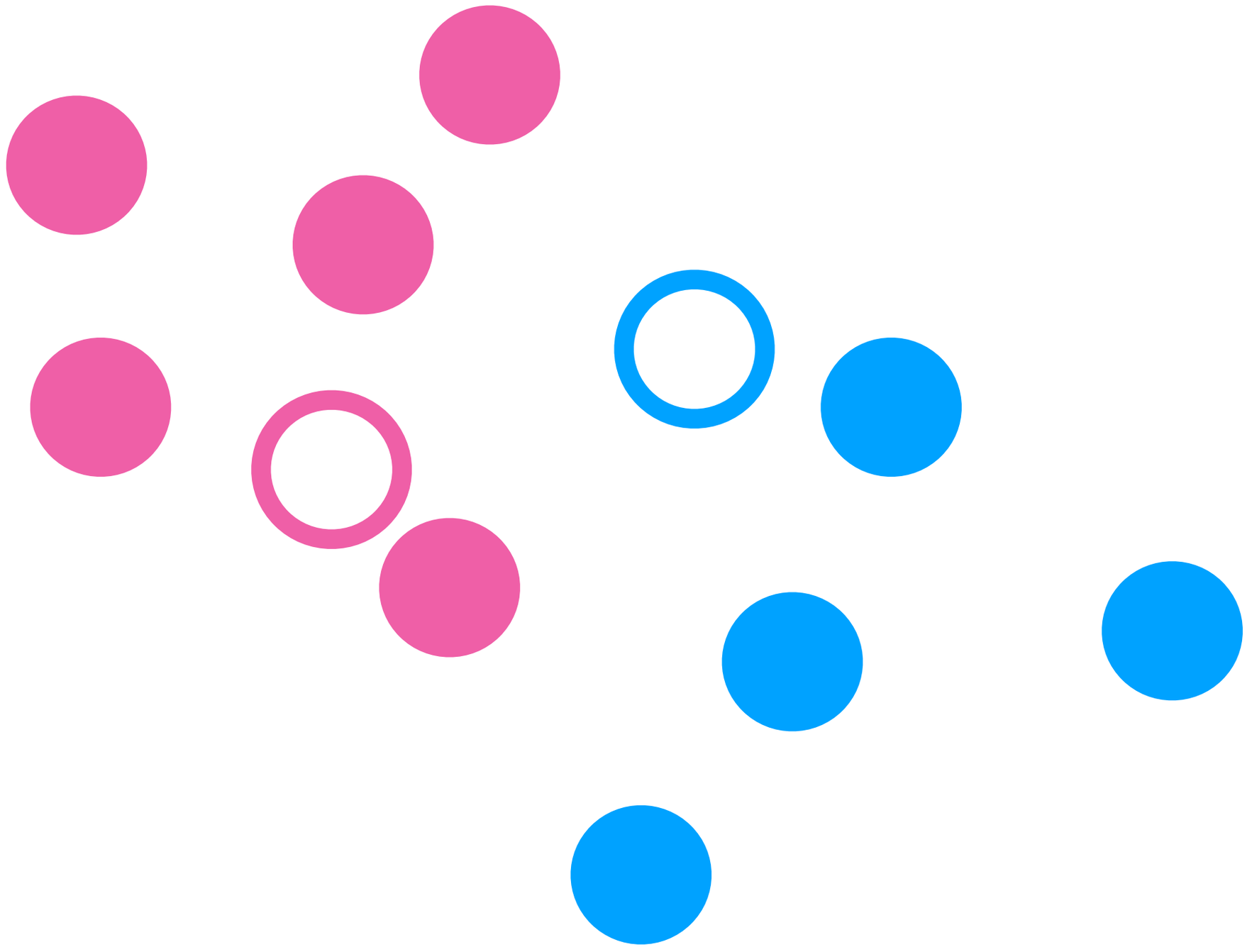
K-Means Clustering



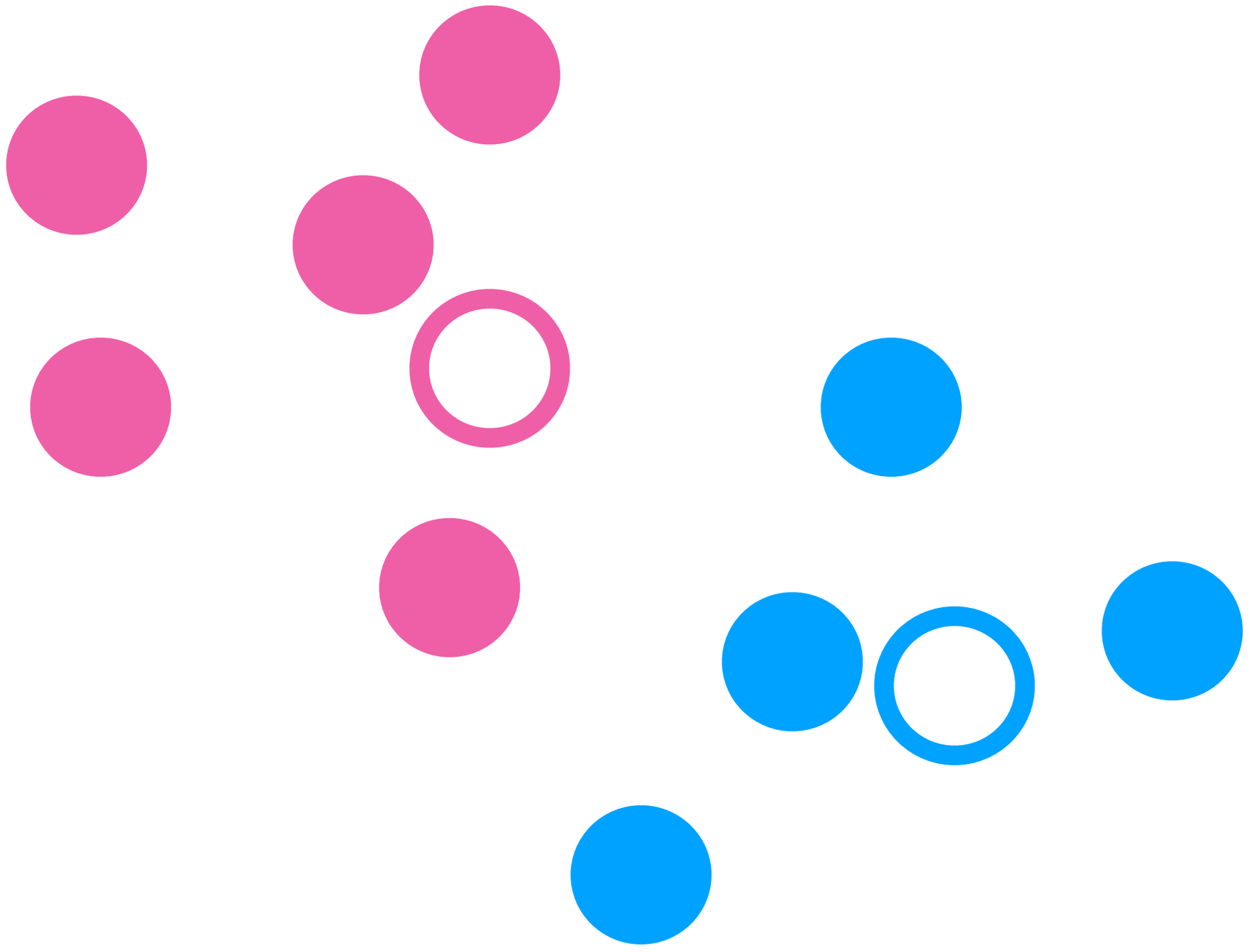
K-Means Clustering



K-Means Clustering



K-Means Clustering



K-means in R

1. Determine which variables to use

2. Find approximation for the number of clusters

3. Run K-means

4. Visualize results

kmeans() in R

```
kmeans(x, centers=2, nstart=10)
```

kmeans() in R

```
kmeans(x, centers=2, nstart=10)
```

dataset

kmeans() in R

```
kmeans(x, centers=2, nstart=10)
```

number of clusters

kmeans() in R

```
kmeans(x, centers=2, nstart=10)
```

number of iterations

kmeans() in R

```
size

iter.max

cluster
```

kmeans() in R

`size` - # of points per cluster

`iter.max`

`cluster`

kmeans() in R

```
size
```

```
iter.max - max # iterations
```

```
cluster
```

kmeans() in R

`size`

`iter.max`

`cluster` - cluster assignments

Example:

2019 World Happiness Report

https://www.kaggle.com/unsdsn/world-happiness

K-means in R

1. Determine which variables to use

2. Find approximation for the number of clusters

3. Run K-means

4. Visualize results

Determine which variables to use

```
> View(happiness)
```

ExampleCode.Rmd* × happiness ×

Filter

	Overall rank	Country or region	Score	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption
1	1	Finland	7.769	1.340	1.587	0.986	0.596	0.153	0.393
2	2	Denmark	7.600	1.383	1.573	0.996	0.592	0.252	0.410
3	3	Norway	7.554	1.488	1.582	1.028	0.603	0.271	0.341
4	4	Iceland	7.494	1.380	1.624	1.026	0.591	0.354	0.118
5	5	Netherlands	7.488	1.396	1.522	0.999	0.557	0.322	0.298
6	6	Switzerland	7.480	1.452	1.526	1.052	0.572	0.263	0.343
7	7	Sweden	7.343	1.387	1.487	1.009	0.574	0.267	0.373
8	8	New Zealand	7.307	1.303	1.557	1.026	0.585	0.330	0.380
9	9	Canada	7.278	1.365	1.505	1.039	0.584	0.285	0.308
10	10	Austria	7.246	1.376	1.475	1.016	0.532	0.244	0.226
11	11	Australia	7.228	1.372	1.548	1.036	0.557	0.332	0.290
12	12	Costa Rica	7.167	1.034	1.441	0.963	0.558	0.144	0.093
13	13	Israel	7.139	1.276	1.455	1.029	0.371	0.261	0.082
14	14	Luxembourg	7.090	1.609	1.479	1.012	0.526	0.194	0.316
15	15	United Kingdom	7.054	1.333	1.538	0.996	0.450	0.348	0.278
16	16	Ireland	7.021	1.499	1.553	0.999	0.516	0.298	0.310
17	17	Germany	6.985	1.373	1.454	0.987	0.495	0.261	0.265
18	18	Belgium	6.923	1.356	1.504	0.986	0.473	0.160	0.210
19	19	United States	6.892	1.433	1.457	0.874	0.454	0.280	0.128
20	20	Czech Republic	6.852	1.269	1.487	0.920	0.457	0.046	0.036
21	21	United Arab Emirates	6.825	1.503	1.310	0.825	0.598	0.262	0.182
22	22	Malta	6.726	1.300	1.520	0.999	0.564	0.375	0.151
23	23	Mexico	6.595	1.070	1.323	0.861	0.433	0.074	0.073

Showing 1 to 23 of 156 entries

Determine which variables to use

```
> View(happiness)
```

ExampleCode.Rmd* × happiness ×

Filter

	Overall rank	Country or region	Score	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption
1	1	Finland	7.769	1.340	1.587	0.986	0.596	0.153	0.393
2	2	Denmark	7.600	1.383	1.573	0.996	0.592	0.252	0.410
3	3	Norway	7.554	1.488	1.582	1.028	0.603	0.271	0.341
4	4	Iceland	7.494	1.380	1.624	1.026	0.591	0.354	0.118
5	5	Netherlands	7.488	1.396	1.522	0.999	0.557	0.322	0.298
6	6	Switzerland	7.480	1.452	1.526	1.052	0.572	0.263	0.343
7	7	Sweden	7.343	1.387	1.487	1.009	0.574	0.267	0.373
8	8	New Zealand	7.307	1.303	1.557	1.026	0.585	0.330	0.380
9	9	Canada	7.278	1.365	1.505	1.039	0.584	0.285	0.308
10	10	Austria	7.246	1.376	1.475	1.016	0.532	0.244	0.226
11	11	Australia	7.228	1.372	1.548	1.036	0.557	0.332	0.290
12	12	Costa Rica	7.167	1.034	1.441	0.963	0.558	0.144	0.093
13	13	Israel	7.139	1.276	1.455	1.029	0.371	0.261	0.082
14	14	Luxembourg	7.090	1.609	1.479	1.012	0.526	0.194	0.316
15	15	United Kingdom	7.054	1.333	1.538	0.996	0.450	0.348	0.278
16	16	Ireland	7.021	1.499	1.553	0.999	0.516	0.298	0.310
17	17	Germany	6.985	1.373	1.454	0.987	0.495	0.261	0.265
18	18	Belgium	6.923	1.356	1.504	0.986	0.473	0.160	0.210
19	19	United States	6.892	1.433	1.457	0.874	0.454	0.280	0.128
20	20	Czech Republic	6.852	1.269	1.487	0.920	0.457	0.046	0.036
21	21	United Arab Emirates	6.825	1.503	1.310	0.825	0.598	0.262	0.182
22	22	Malta	6.726	1.300	1.520	0.999	0.564	0.375	0.151
23	23	Mexico	6.595	1.070	1.323	0.861	0.433	0.074	0.073

Showing 1 to 23 of 156 entries

Determine which variables to use

```
> happiness <- happiness[,-c(1,2,3)]
```

ExampleCode.Rmd* × happiness ×

Filter

	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption
1	1.340	1.587	0.986	0.596	0.153	0.393
2	1.383	1.573	0.996	0.592	0.252	0.410
3	1.488	1.582	1.028	0.603	0.271	0.341
4	1.380	1.624	1.026	0.591	0.354	0.118
5	1.396	1.522	0.999	0.557	0.322	0.298
6	1.452	1.526	1.052	0.572	0.263	0.343
7	1.387	1.487	1.009	0.574	0.267	0.373
8	1.303	1.557	1.026	0.585	0.330	0.380
9	1.365	1.505	1.039	0.584	0.285	0.308
10	1.376	1.475	1.016	0.532	0.244	0.226
11	1.372	1.548	1.036	0.557	0.332	0.290
12	1.034	1.441	0.963	0.558	0.144	0.093
13	1.276	1.455	1.029	0.371	0.261	0.082
14	1.609	1.479	1.012	0.526	0.194	0.316
15	1.333	1.538	0.996	0.450	0.348	0.278
16	1.499	1.553	0.999	0.516	0.298	0.310
17	1.373	1.454	0.987	0.495	0.261	0.265
18	1.356	1.504	0.986	0.473	0.160	0.210
19	1.433	1.457	0.874	0.454	0.280	0.128
20	1.269	1.487	0.920	0.457	0.046	0.036
21	1.503	1.310	0.825	0.598	0.262	0.182
22	1.300	1.520	0.999	0.564	0.375	0.151
23	1.070	1.323	0.861	0.433	0.074	0.073

Showing 1 to 23 of 156 entries

K-means in R

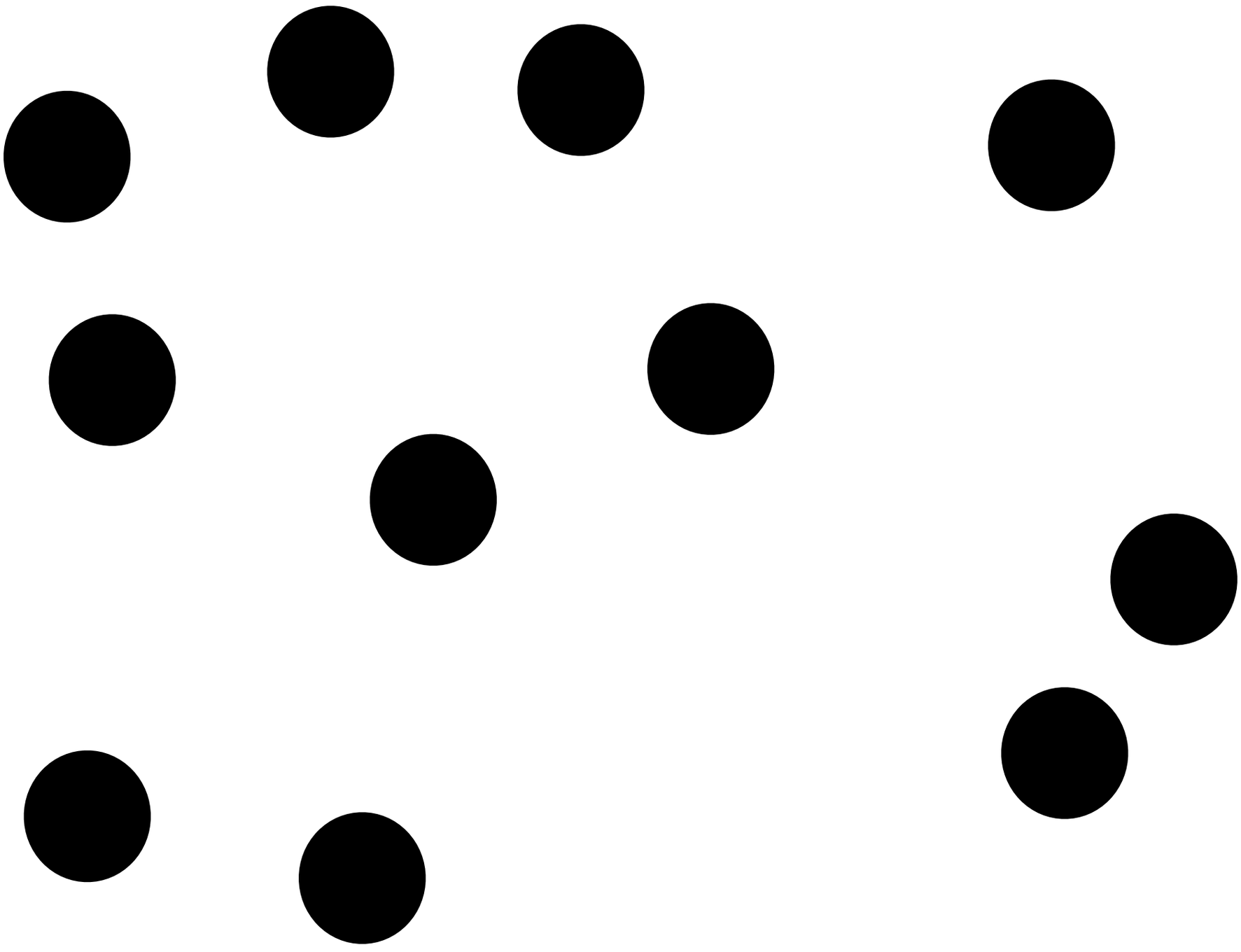
1. Determine which variables to use

2. Find approximation for the number of clusters

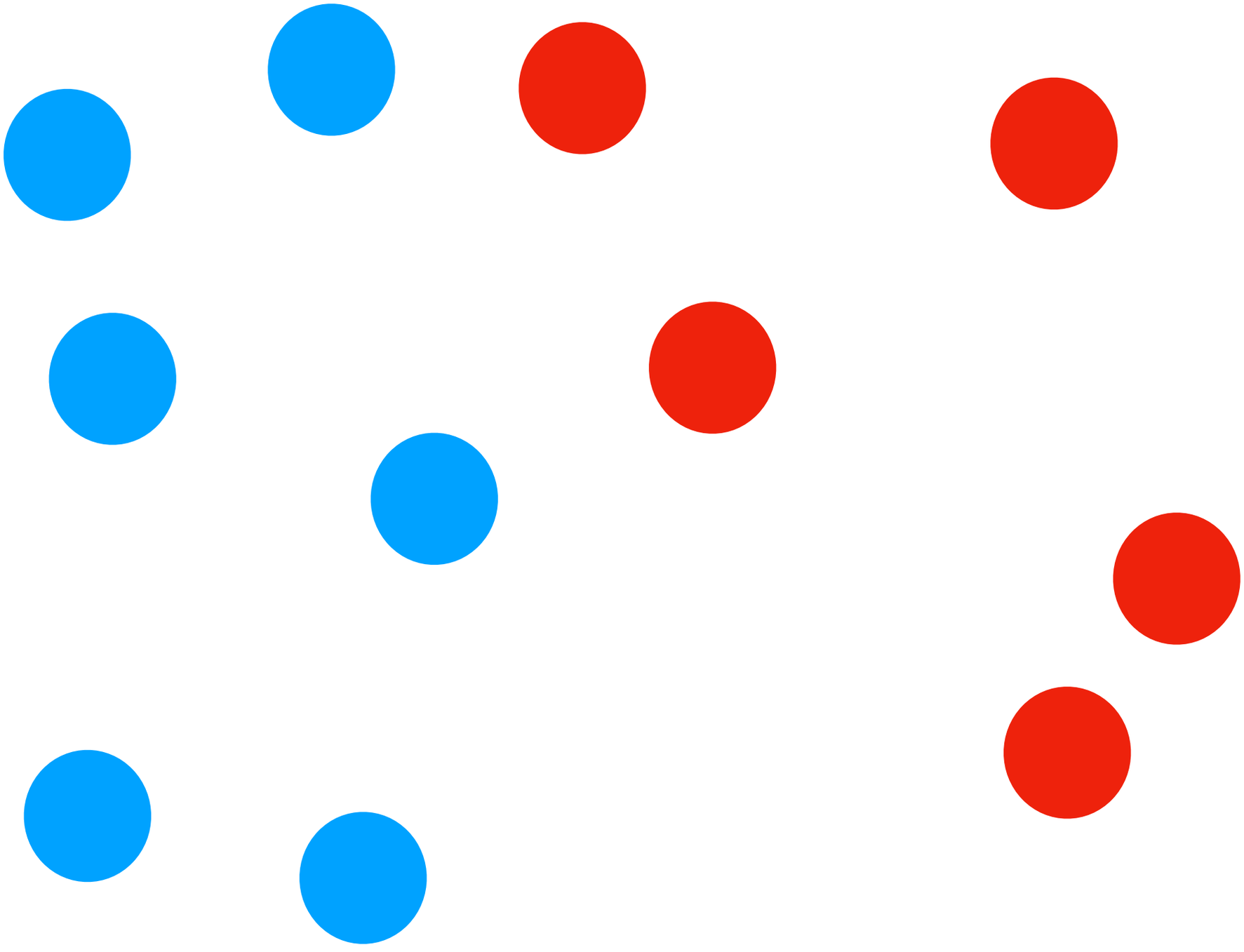
3. Run K-means

4. Visualize results

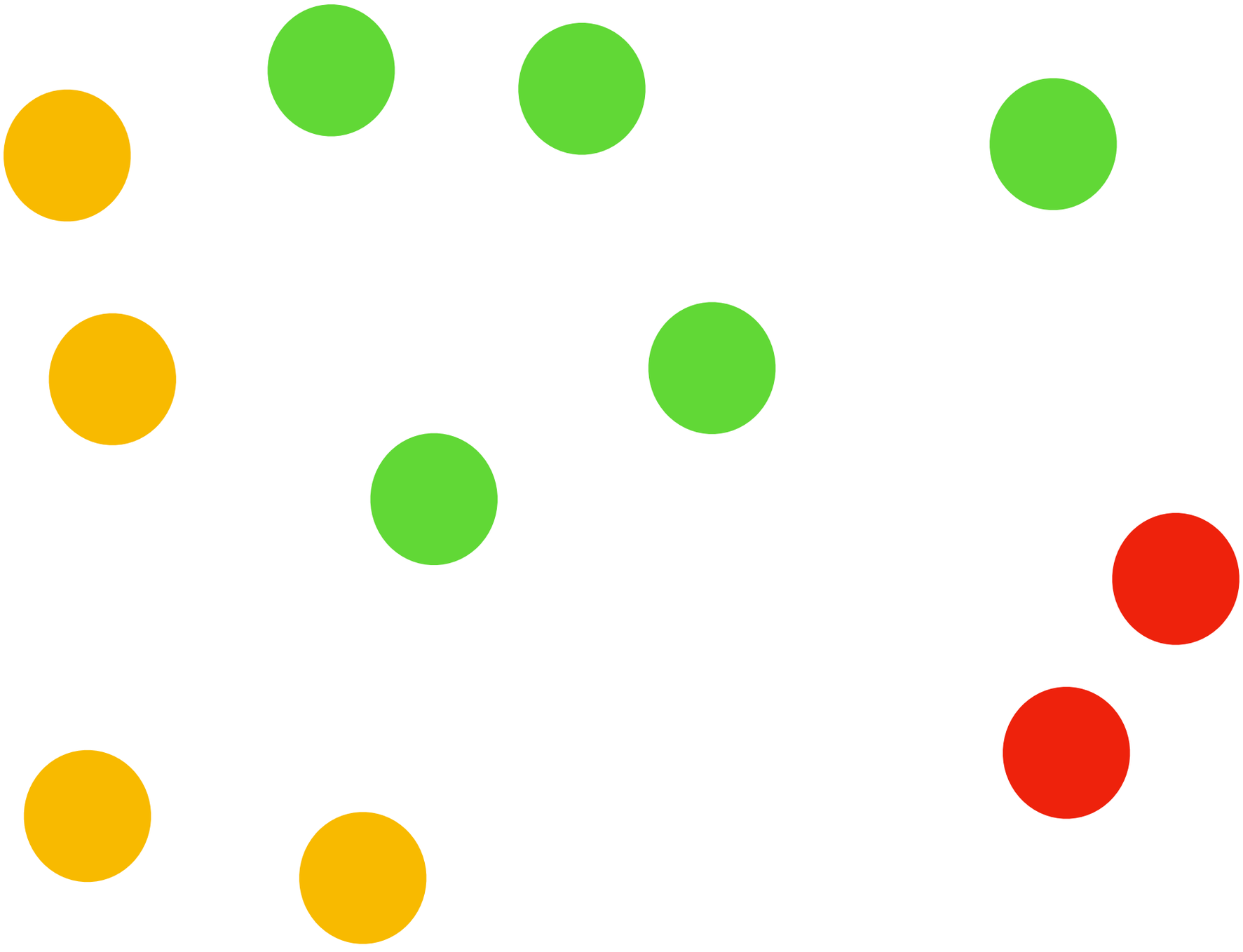
Number of Clusters



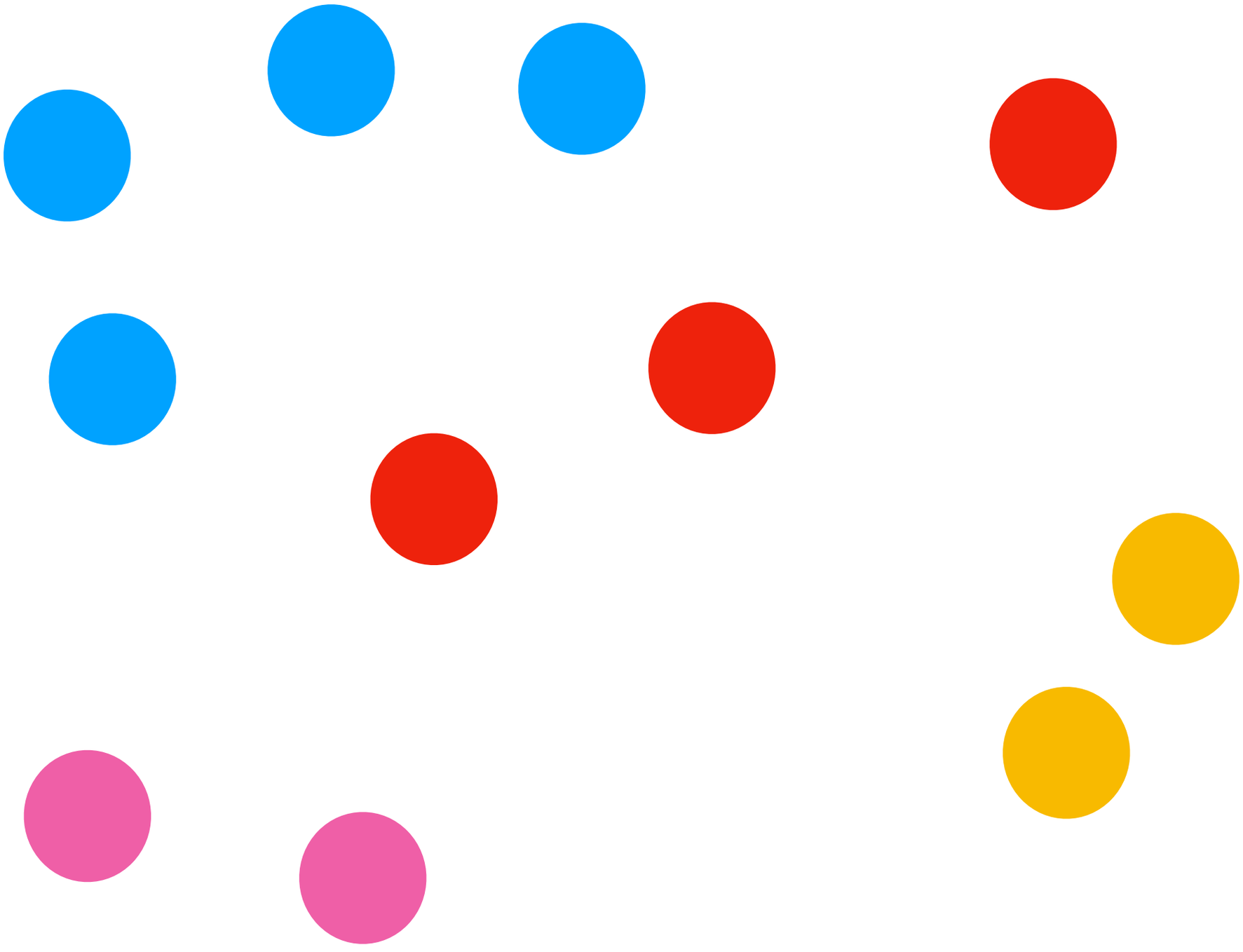
Number of Clusters



Number of Clusters



Number of Clusters



The Elbow Method

- ∀ potential number of clusters, k
    - Run k-means with k clusters
        - Record the performance of the model
- Plot the performance vs. number of clusters
- Pick the elbow! This is your k!

The Elbow Method

```
set.seed(111711)
```

<- !!

The Elbow Method

```
set.seed(111711)
```

Run each model
Save performance

```
# Vector that will store performance for each k
modelWithiness <- c()
# Determine what number of clusters we should use as our approximation
for(i in 1:10) {
  kmeans.model <- kmeans(happiness, centers=i, nstart=30)
  modelWithiness <- append(modelWithiness, kmeans.model$tot.withinss)
}
```

The Elbow Method

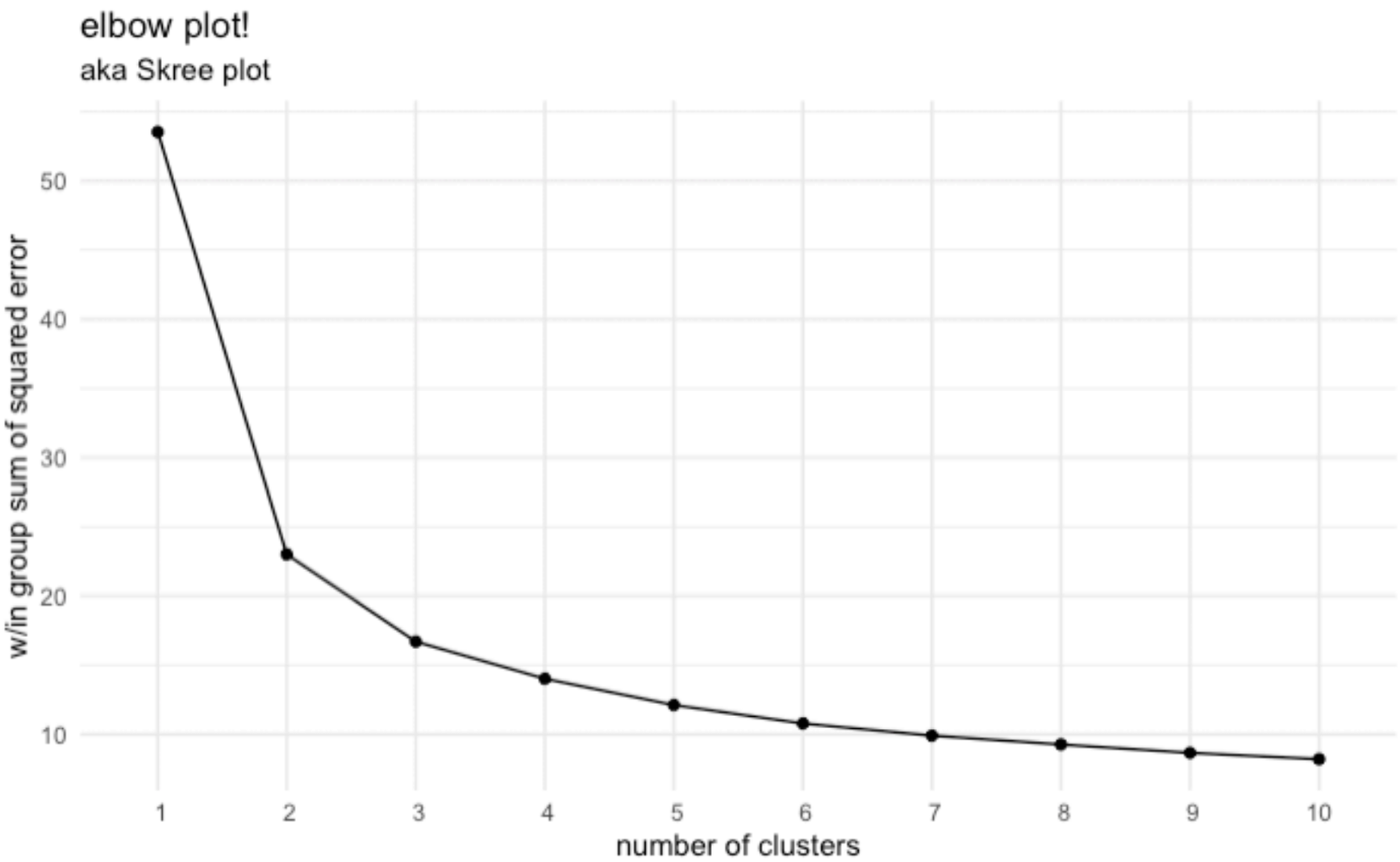
```
set.seed(111711)
```

```
# Vector that will store performance for each k
modelWithiness <- c()
# Determine what number of clusters we should use as our approximation
for(i in 1:10) {
  kmeans.model <- kmeans(happiness, centers=i, nstart=30)
  modelWithiness <- append(modelWithiness, kmeans.model$tot.withinss)
}
```

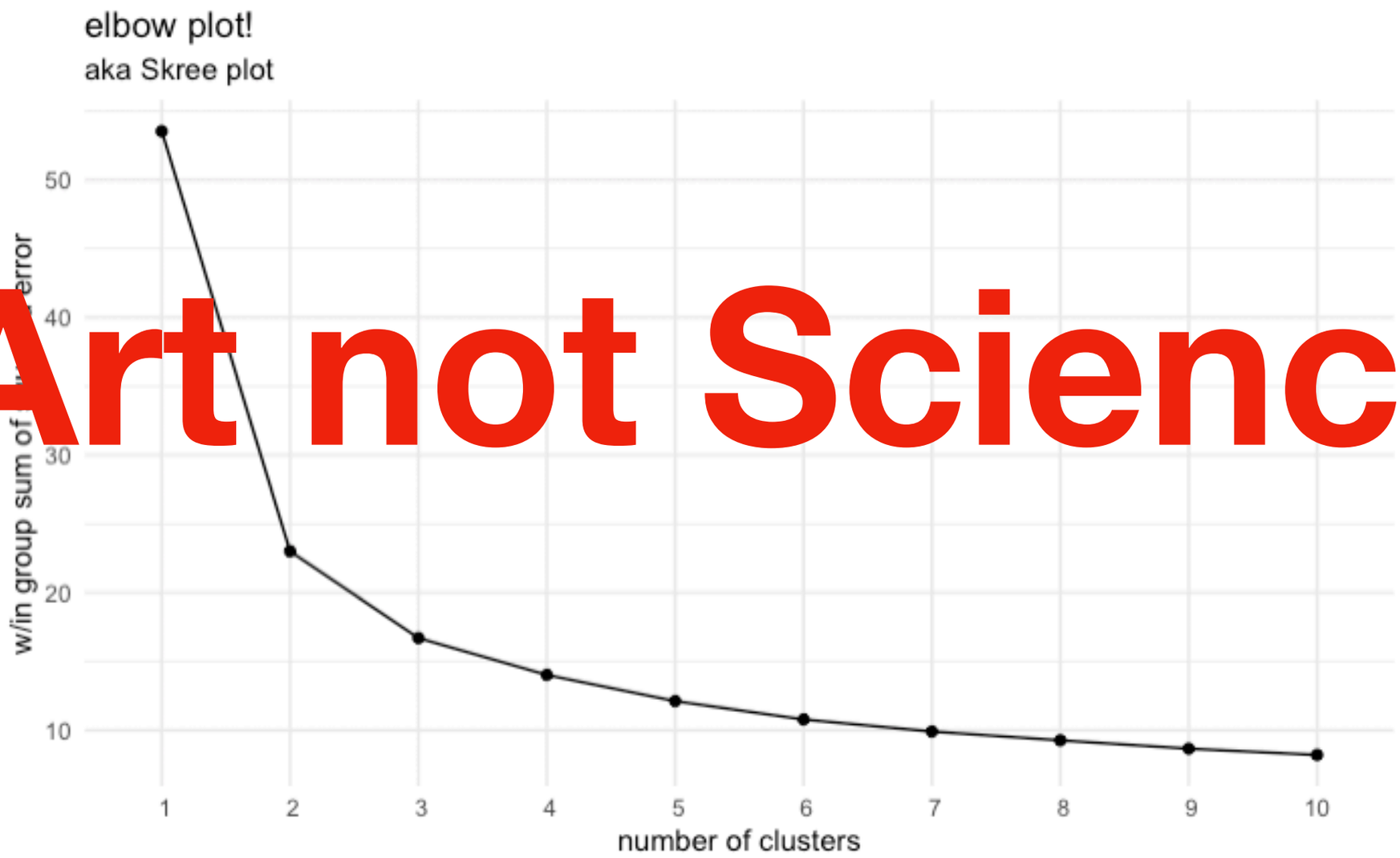
Create elbow plot!

```
# Make the elow plot
nums <- c(1:10)
ggplot(data.frame(nums, modelWithiness), aes(x=nums,y=modelWithiness)) +
    geom_line() + geom_point()
    + scale_x_discrete(limits=nums) + labs(y = "w/in group sum of squared error", x= "number of clusters", title =
"elbow plot!", subtitle = "aka Skree plot") + theme_minimal()
```

The Elbow Plot!

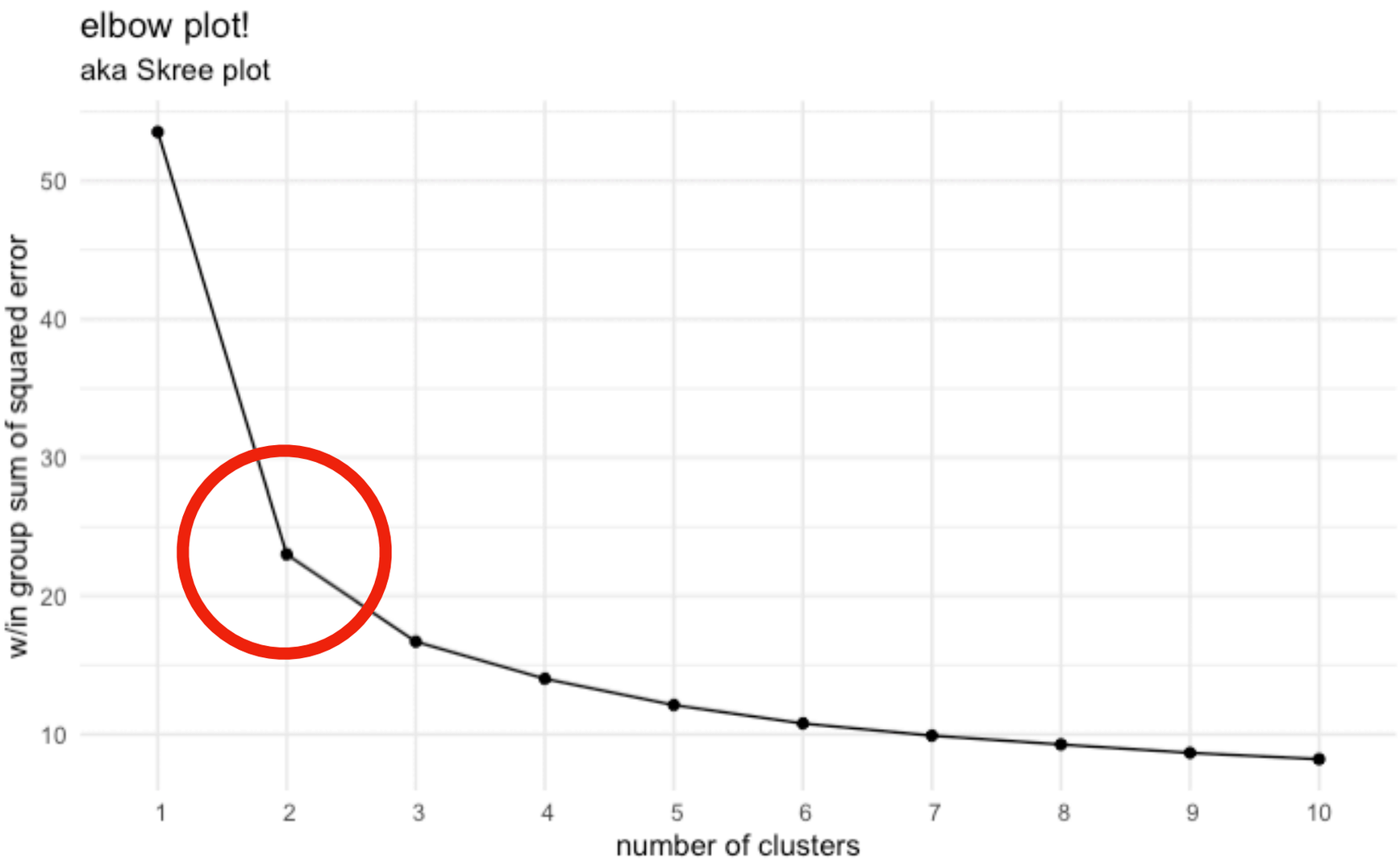


The Elbow Plot!

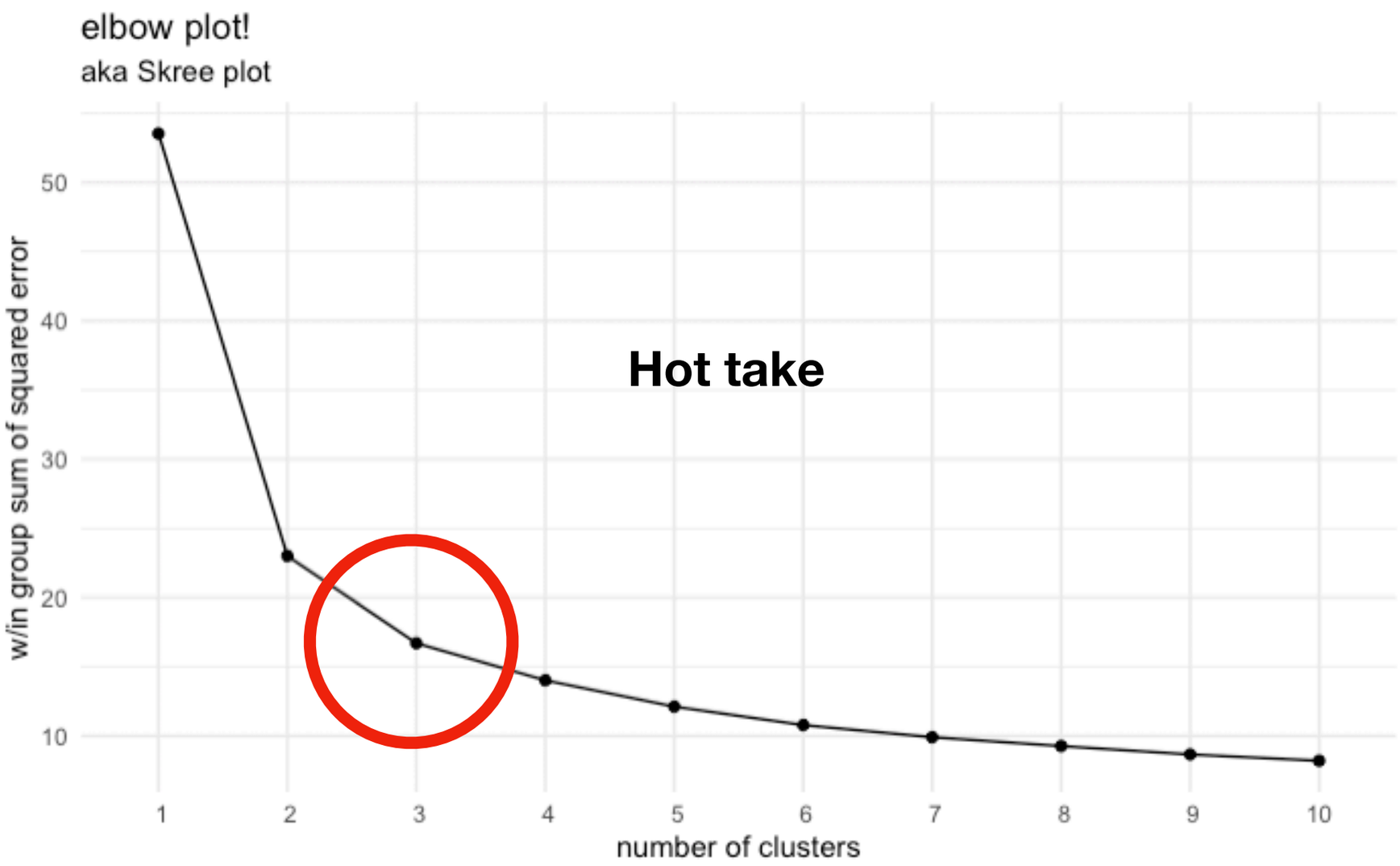


Art not Science

The Elbow Plot!



The Elbow Plot!



K-means in R

1. Determine which variables to use

2. Find approximation for the number of clusters

3. Run K-means

4. Visualize results

Run kmeans() in R

```r
# Construct model with 3 clusters
kmeans.real.model <- kmeans(happiness, centers = 3, nstart = 30)
```

```
> # View the resulting model
> kmeans.real.model
K-means clustering with 3 clusters of sizes 48, 64, 44

Cluster means:
  GDP per capita Social support Healthy life expectancy Freedom to make life choices Generosity Perceptions of corruption
1      1.3341875      1.4569375                0.9596458                     0.4763542  0.2004375                0.17395833
2      0.9368281      1.2612500                0.7605000                     0.3839375  0.1573750                0.07090625
3      0.3910227      0.8618636                0.4182500                     0.3137273  0.2077955                0.09922727

Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 2 1 1 2 1 1 1 1 2 1 2 1 2 2 2 2 1 2 1 2 1 1 1 2 2 1 2 1 2 2 2 1
 [65] 2 1 3 2 2 2 2 2 2 3 2 1 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 3 2 3 3 3 2 3 3 2 2 2 2 2 2 2 3 3 2 3 3 2 2 3 2 3 3 3 3 2 3 2 3 3
[129] 3 2 2 3 2 3 3 3 2 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3

Within cluster sum of squares by cluster:
[1] 3.261740 6.545869 6.909874
 (between_SS / total_SS =  68.8 %)

Available components:

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

Run kmeans() in R

```r
# Construct model with 3 clusters
kmeans.real.model <- kmeans(happiness, centers = 3, nstart = 30)
```

```
> # View the resulting model
> kmeans.real.model
K-means clustering with 3 clusters of sizes 48, 64, 44

Cluster means:
  GDP per capita Social support Healthy life expectancy Freedom to make life choices Generosity Perceptions of corruption
1      1.3341875      1.4569375                0.9596458                     0.4763542  0.2004375                0.17395833
2      0.9368281      1.2612500                0.7605000                     0.3839375  0.1573750                0.07090625
3      0.3910227      0.8618636                0.4182500                     0.3137273  0.2077955                0.09922727

Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 1 2 1 1 2 1 1 1 1 1 2 1 2 1 2 2 2 2 1 2 1 2 1 1 1 2 2 1 2 1 2 2 2 1
 [65] 2 1 3 2 2 2 2 2 2 3 2 1 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 3 2 3 3 3 2 3 3 2 2 2 2 2 2 2 3 3 2 3 3 2 2 3 2 3 3 3 3 2 3 2 3 3
[129] 3 2 2 3 2 3 3 3 2 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3

Within cluster sum of squares by cluster:
[1] 3.261740 6.545869 6.909874
 (between_SS / total_SS =  68.8 %)

Available components:

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

Run kmeans() in R

```r
# Construct model with 3 clusters
kmeans.real.model <- kmeans(happiness, centers = 3, nstart = 30)
```

```
> # View the resulting model
> kmeans.real.model
K-means clustering with 3 clusters of sizes 48, 64, 44

Cluster means:
  GDP per capita Social support Healthy life expectancy Freedom to make life choices Generosity Perceptions of corruption
1      1.3341875      1.4569375                0.9596458                     0.4763542  0.2004375                0.17395833
2      0.9368281      1.2612500                0.7605000                     0.3839375  0.1573750                0.07090625
3      0.3910227      0.8618636                0.4182500                     0.3137273  0.2077955                0.09922727

Clustering vector:
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 2 1 1 1 2 1 1 2 1 1 1 1 2 1 2 1 2 2 2 2 1 2 1 2 1 1 1 2 2 1 2 1 2 2 2 1
 [65] 2 1 3 2 2 2 2 2 2 3 2 1 2 2 2 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 3 2 3 3 3 2 3 3 2 2 2 2 2 2 2 3 3 2 3 3 2 2 3 2 3 3 3 3 2 3 2 3 3
[129] 3 2 2 3 2 3 3 3 2 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3

Within cluster sum of squares by cluster:
[1] 3.261740 6.545869 6.909874
 (between_SS / total_SS =  68.8 %)

Available components:

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

K-means in R

1. Determine which variables to use

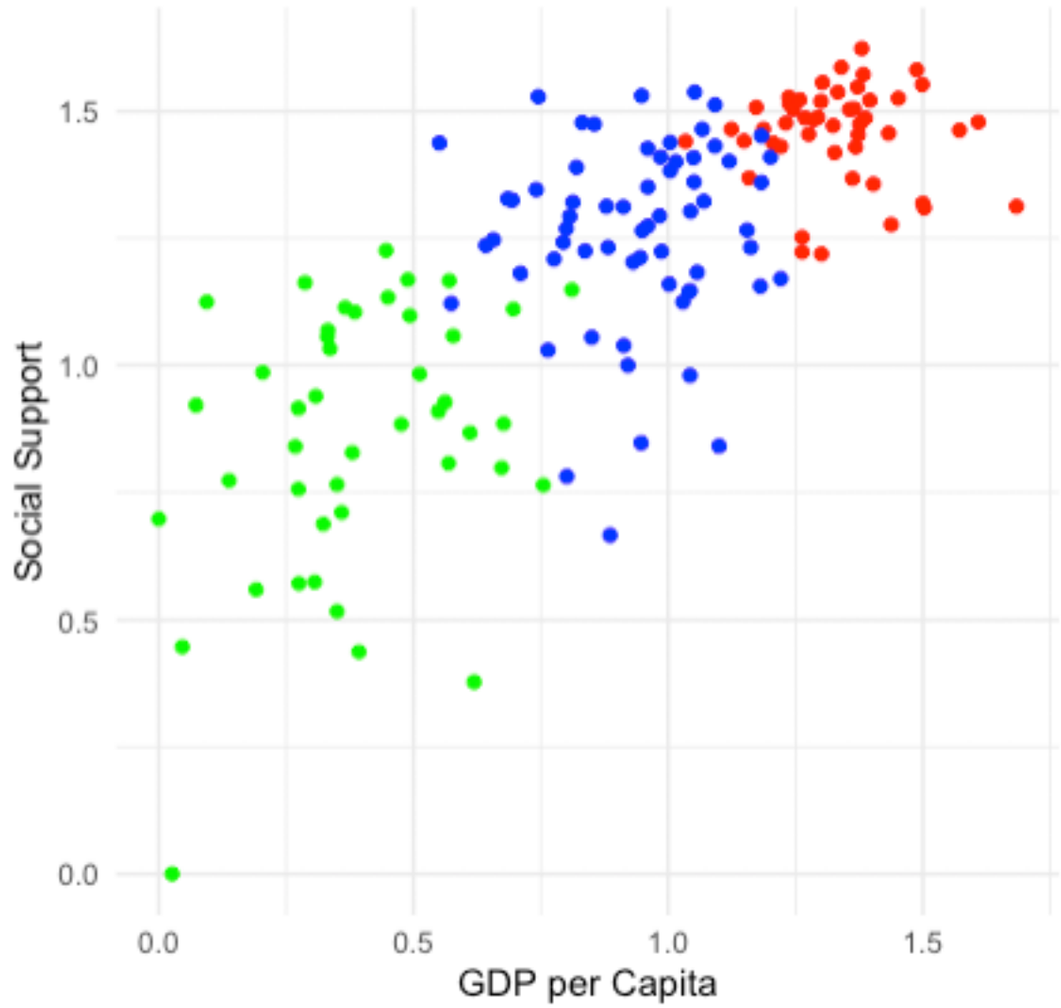
2. Find approximation for the number of clusters

3. Run K-means

4. Visualize results

Visualizing k-means

```
# Visualizing GDP Per Capita vs Social Support grouped by cluster
ggplot(data=happiness, aes(x=happiness$`GDP per capita`, y=happiness$`Social support`, col = as.factor(kmeans.real.model$cluster))) + geom_point() +  labs(y =
"Social Support", x= "GDP per Capita", title = "K-means clustering of countries by happiness criteria with 3 clusters") + scale_color_manual(breaks = c("1", "2",
"3"), values=c("red", "blue", "green"))  + theme_minimal()
```



K-means in R

1. Determine which variables to use

2. Find approximation for the number of clusters

3. Run K-means

4. Visualize results

kmeans()

Elbow Plots!

The End!