K-means Clustering

Into Statistical Learning with Unsupervised Learning

Statistical Learning: refers to a set of tools for modeling and understanding complex datasets.

Unsupervised Learning: none of the variables are *response* variables (i.e., there is not labeled data).

- Unsupervised learning is like an extension of EDA

What is clustering (aka cluster analysis)?

very broad set of techniques for finding subgroups (clusters) within a dataset

So, observations within clusters are more similar to each other, AND observations in different clusters are more different from each other.

To find distance:

- e.g. Euclidean distance between two observations i and j

BUT...

units matter!

- we might standardize each variable/column of the dataset to have a mean of 0 and standard deviation with `scale()`

Clustering's Objective

- If observation i is in cluster k then $i \in C_k$
- Minimize the within-cluster variation $W(C_k)$ for each cluster C_k.
- Can define $W(C_k)$ using the **squared Euclidean distance**.
- Commonly referred to as the within-cluster sum of squares (WSS)

Lloyd's Algorithm (aka K-means)

(1) Choose K random centers, aka **centroids**

(2) Assign each obsrvation closest center (using the Euclidean distance)

(3) Repeat until cluster assignments stop changing

- Computer new centroids as the averages of the updated groups

- Reassign each observations to closest center

Converges to a local optimum, not the global

Results will change from run to run (set the seed!)

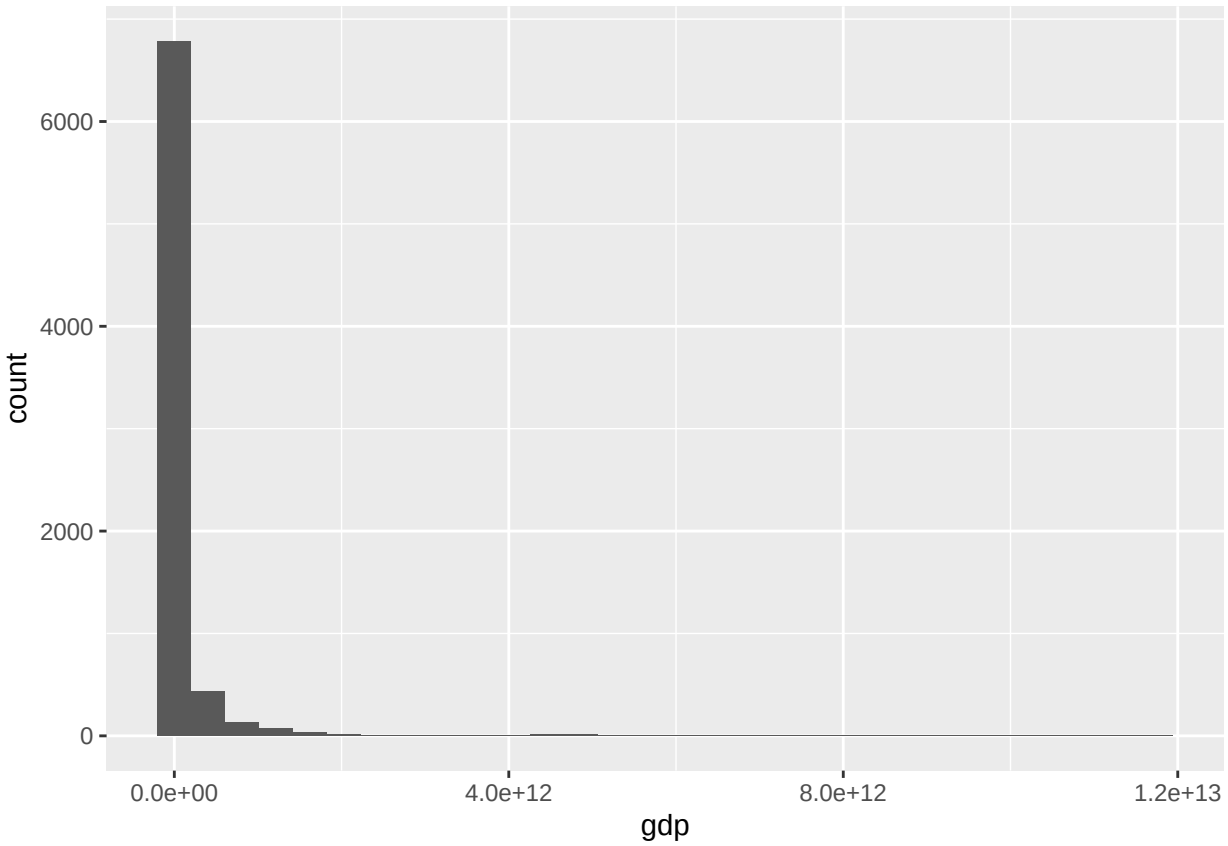
Takes K as an input!

Gapminder data

```
gapminder <- as_tibble(gapminder)
head(gapminder)
```

```
## # A tibble: 6 x 9
##   country    year infant_mortality life_expectancy fertility population      gdp
##   <fct>     <int>            <dbl>           <dbl>     <dbl>      <dbl>    <dbl>
## 1 Albania    1960            115.             62.9      6.19    1636054 NA
## 2 Algeria    1960            148.             47.5      7.65   11124892  1.38e10
## 3 Angola     1960            208              36.0      7.32    5270844 NA
## 4 Antigua ~  1960             NA              63.0      4.43      54681 NA
## 5 Argentina  1960             59.9            65.4      3.11   20619075  1.08e11
## 6 Armenia    1960             NA              66.9      4.55    1867396 NA
## # i 2 more variables: continent <fct>, region <fct>
```

```
gapminder %>%
  ggplot(aes(x = gdp))+
  geom_histogram()
```

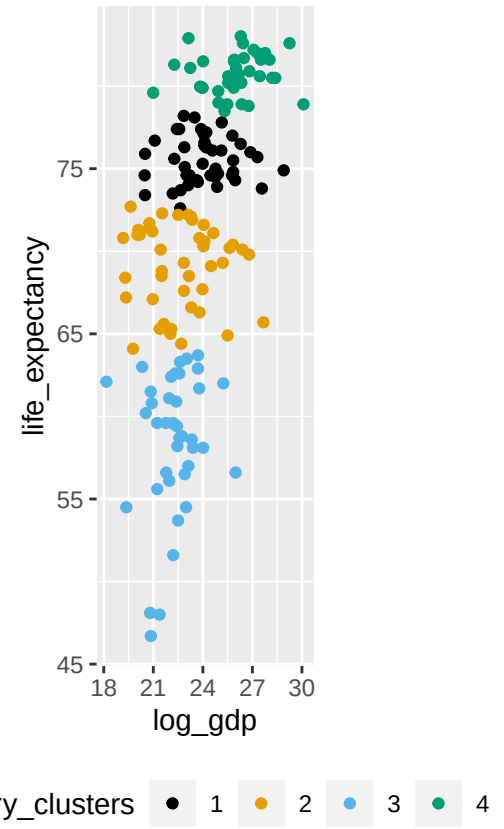



```
clean_gapminder <- gapminder %>%
  filter(year == 2011, !is.na(gdp)) %>%
  mutate(log_gdp = log(gdp))
```

K-means clustering example

```
init_kmeans <-
  kmeans(dplyr::select(clean_gapminder, log_gdp, life_expectancy), algorithm = "Lloyd", centers = 4, ns

clean_gapminder %>%
  mutate(country_clusters = as.factor(init_kmeans$cluster)) %>%
  ggplot(aes(x = log_gdp, y = life_expectancy, color = country_clusters))+
  geom_point()+
  ggthemes::scale_color_colorblind()+
  theme(legend.position = "bottom")+
  coord_fixed()
```



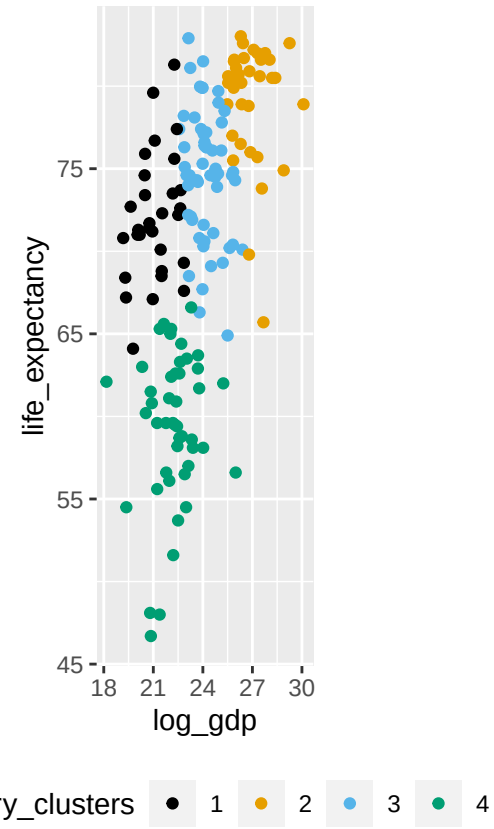
Standardize the Variables

as a rule of thumb, we usually standardize the variables

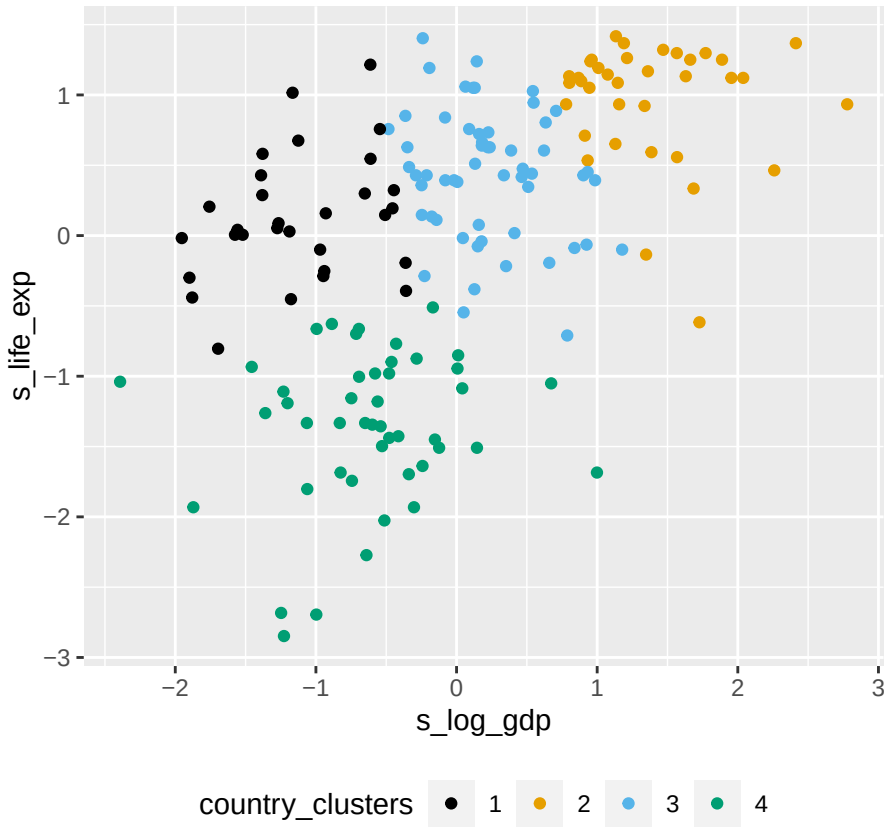
```
clean_gapminder <- clean_gapminder %>%
  mutate(s_log_gdp = as.numeric(scale(log_gdp, center = TRUE, scale = TRUE)),
         s_life_exp = as.numeric(scale(life_expectancy, center = TRUE, scale = TRUE)))

s_kmeans <- kmeans(dplyr::select(clean_gapminder, s_log_gdp, s_life_exp), algorithm = "Lloyd", centers =

clean_gapminder %>%
  mutate(country_clusters = as.factor(s_kmeans$cluster)) %>%
  ggplot(aes(x = log_gdp, y = life_expectancy, color = country_clusters))+
  geom_point()+
  ggthemes::scale_color_colorblind()+
  theme(legend.position = "bottom")+
  coord_fixed()
```




```
clean_gapminder %>%
  mutate(country_clusters = as.factor(s_kmeans$cluster)) %>%
  ggplot(aes(x = s_log_gdp, y = s_life_exp, color = country_clusters))+
  geom_point()+
  ggthemes::scale_color_colorblind()+
  theme(legend.position = "bottom")+
  coord_fixed()
```



Randomness in Clustering

Fixes

- nstart: run the algorithm `nstart` times and pick the results with the lowest total within-cluster variation (up to permutation of labels)
- K-means++

Choosing Number of Clusters

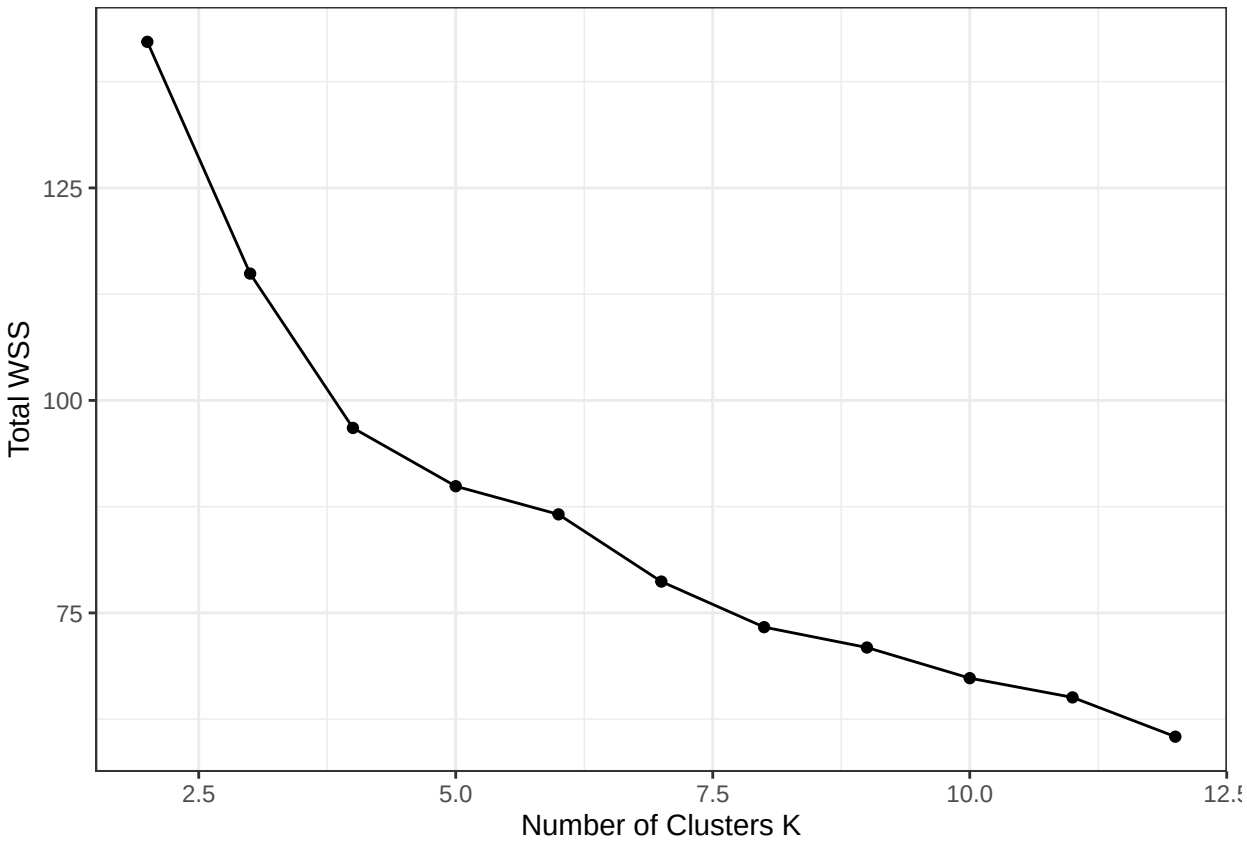
Options

- Elbow plot (use with caution)

```
n_clusters_search <- 2:12

tibble(total_wss = sapply(n_clusters_search, function(k_choice){
  kmeans_results <- kcca(dplyr::select(clean_gapminder, s_log_gdp, s_life_exp),
                         k = k_choice,
                         control = list(initcent = "kmeanspp"))
  return(sum(kmeans_results@clusinfo$size * kmeans_results@clusinfo$av_dist))
})) %>%
  mutate(k = n_clusters_search) %>%
  ggplot(aes(x = k, y = total_wss)) +
  geom_line()+
```

```
  geom_point()+
  labs(x = "Number of Clusters K", y = "Total WSS")+
  theme_bw()
```



- Coming later this week: a model-based approach to choosing number of clusters