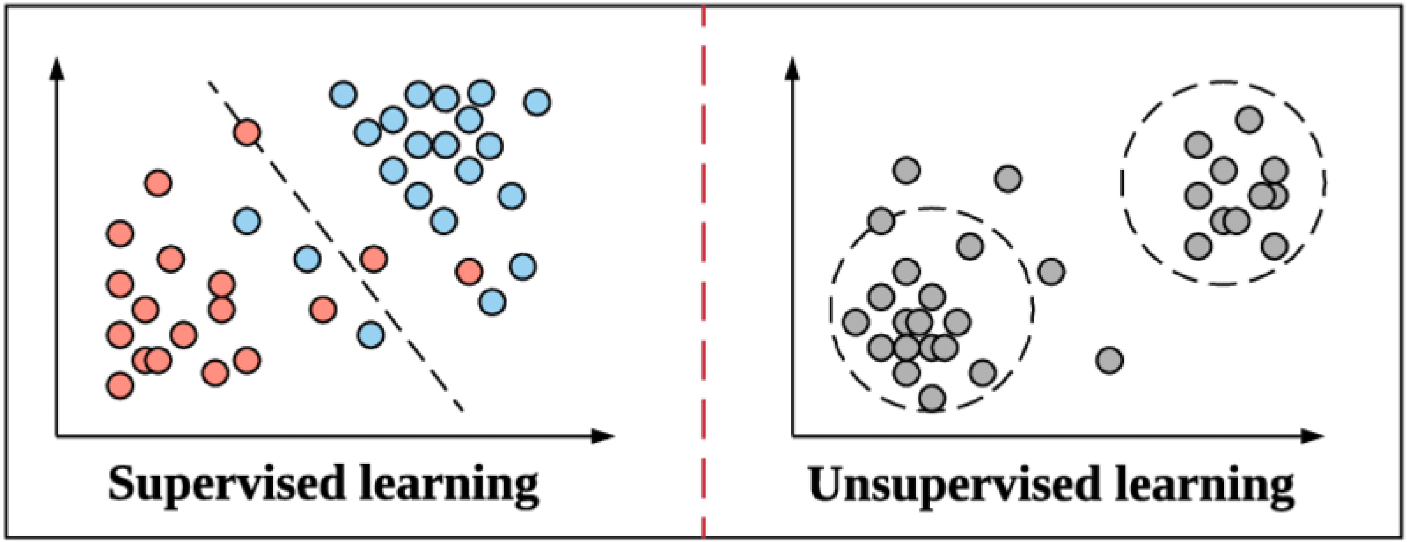
# Introduction to clustering analysis

Stephanie J. Spielman

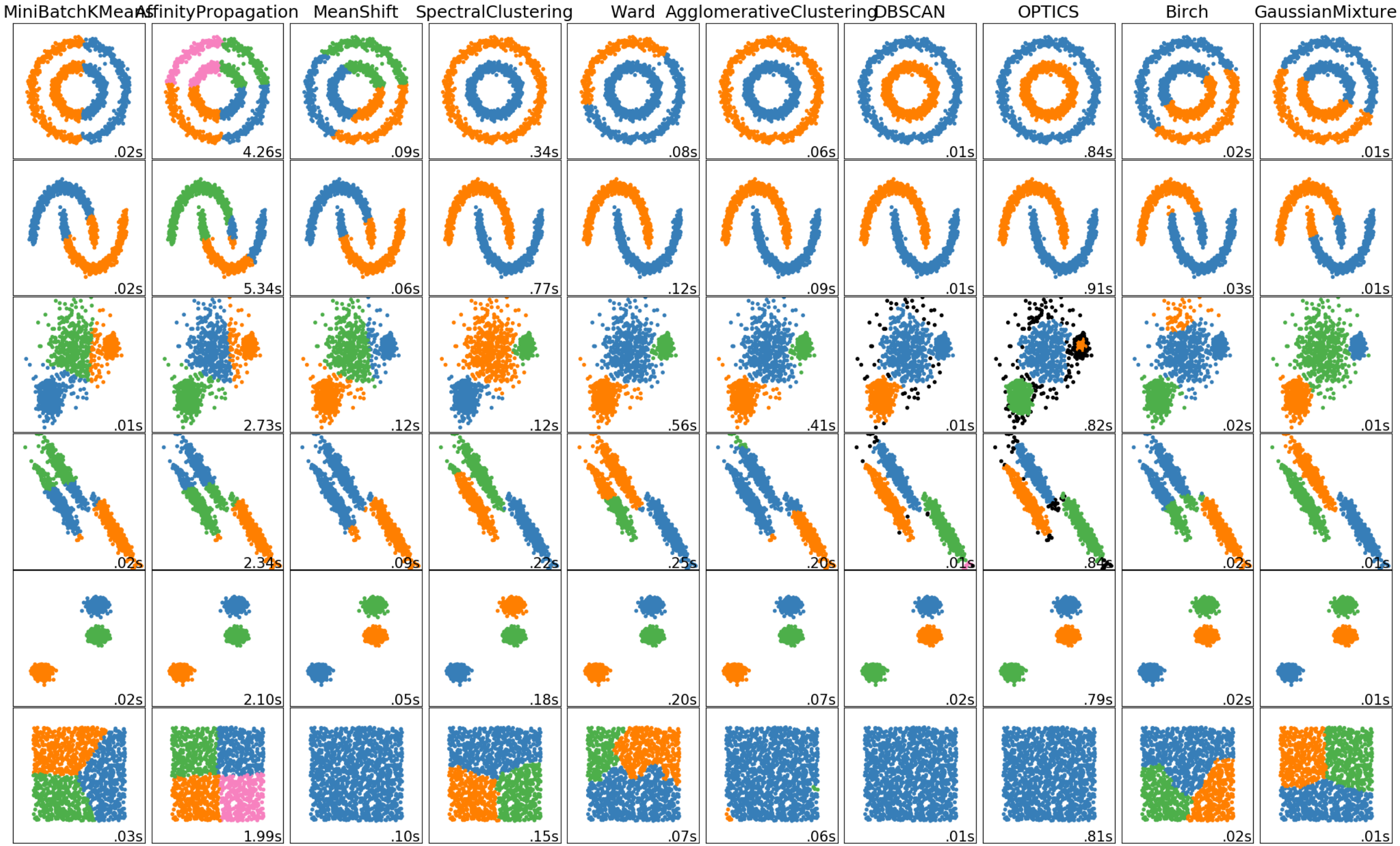
Data Science for Biologists, Spring 2020

# Clustering

- An **unsupervised** approach to placing observations into clusters
- Cluster = previously **unknown/undetected** groupings
- Requires some approach to measuring distance/similarity among observations



# There are MANY algorithms for this



- Image from https://scikit-learn.org/stable/modules/clustering.html

# GARBAGE IN, GARBAGE OUT

- All based one some kind of mathematical comparison among data points

# k-means clustering

1. Place k (*determined in advanced*) "centroids" in the data
2. Assign point to cluster k based on Euclidian distance
3. Re-compute each k centroid based on means of associated points
4. Re-assign centroids
5. Repeat until convergence (stops changing)

**Thanks, internet!**

- https://www.naftaliharris.com/blog/visualizing-k-means-clustering/
- https://twitter.com/allison_horst/status/1250477975130140672?s=20
- https://en.wikipedia.org/wiki/K-means_clustering#/media/File:K-means_convergence.gif

# Let's cluster

```
set.seed(1011)

## just making the URL fit..
wine_url <- paste0("https://raw.githubusercontent.com/sjspielman/",
                   "datascience_for_biologists/master/data/wine.csv")
wine <- read_csv(wine_url)
dplyr::glimpse(wine)
## Observations: 178
## Variables: 9
## $ Cultivar       <chr> "A", "A", "A", "A", "A", "A", "A", "A", "A", "A",…
## $ Alcohol        <dbl> 14.23, 13.20, 13.16, 14.37, 13.24, 14.20, 14.39, …
## $ MalicAcid      <dbl> 1.71, 1.78, 2.36, 1.95, 2.59, 1.76, 1.87, 2.15, 1…
## $ Ash            <dbl> 2.43, 2.14, 2.67, 2.50, 2.87, 2.45, 2.45, 2.61, 2…
## $ Magnesium      <dbl> 127, 100, 101, 113, 118, 112, 96, 121, 97, 98, 10…
## $ TotalPhenol    <dbl> 2.80, 2.65, 2.80, 3.85, 2.80, 3.27, 2.50, 2.60, 2…
## $ Flavanoids     <dbl> 3.06, 2.76, 3.24, 3.49, 2.69, 3.39, 2.52, 2.51, 2…
## $ NonflavPhenols <dbl> 0.28, 0.26, 0.30, 0.24, 0.39, 0.34, 0.30, 0.31, 0…
## $ Color          <dbl> 5.64, 4.38, 5.68, 7.80, 4.32, 6.75, 5.25, 5.05, 5…
```

# Let's cluster with k=3

- **ONLY NUMERIC DATA!!!** You must remove any categorical columns!!!

```
k <- 3 # don't hardcode!

wine %>%
  # remove the categorical column first
  select(-Cultivar) %>%
  kmeans(k) -> wine_k3
```

```
wine_k3
## K-means clustering with 3 clusters of sizes 73, 26, 79
##
## Cluster means:
##    Alcohol MalicAcid      Ash Magnesium TotalPhenol Flavanoids
## 1 12.70356  2.298356 2.257671  87.21918    2.136438   1.829315
## 2 13.32769  2.066538 2.511923 125.11538    2.531538   2.463462
## 3 13.16747  2.460253 2.419241 102.96203    2.363924   2.071139
##   NonflavPhenols    Color
## 1      0.3880822 4.363973
## 2      0.3065385 5.521154
## 3      0.3558228 5.547089
##
## Clustering vector:
##   [1] 2 3 3 3 2 3 3 2 3 3 3 1 1 1 3 3 2 2 3 2 2 3 3 1 3 2 1 1 3 3 3 3 3 2 3
##  [36] 3 3 3 3 2 2 1 3 3 3 3 3 3 3 3 1 1 3 2 2 2 2 3 3 1 3 3 1 1 3 3 1 1 3 2
##  [71] 3 1 1 2 3 3 1 3 2 3 1 1 1 1 1 3 1 1 1 1 1 1 1 1 3 2 2 1 1 1 3 1 3 1 1
## [106] 1 1 1 1 1 3 1 3 1 1 1 1 3 1 1 3 2 3 1 1 1 1 1 1 1 2 3 3 3 1 1 1 3 1 3
## [141] 3 1 3 1 3 3 1 1 1 3 2 3 2 3 3 1 1 3 3 1 1 3 3 3 1 1 3 1 3 3 3 1 1 3 3
## [176] 2 2 3
##
## Within cluster sum of squares by cluster:
## [1] 2256.940 3462.511 2843.740
##  (between_SS / total_SS =  77.3 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

# Which row is in which cluster?

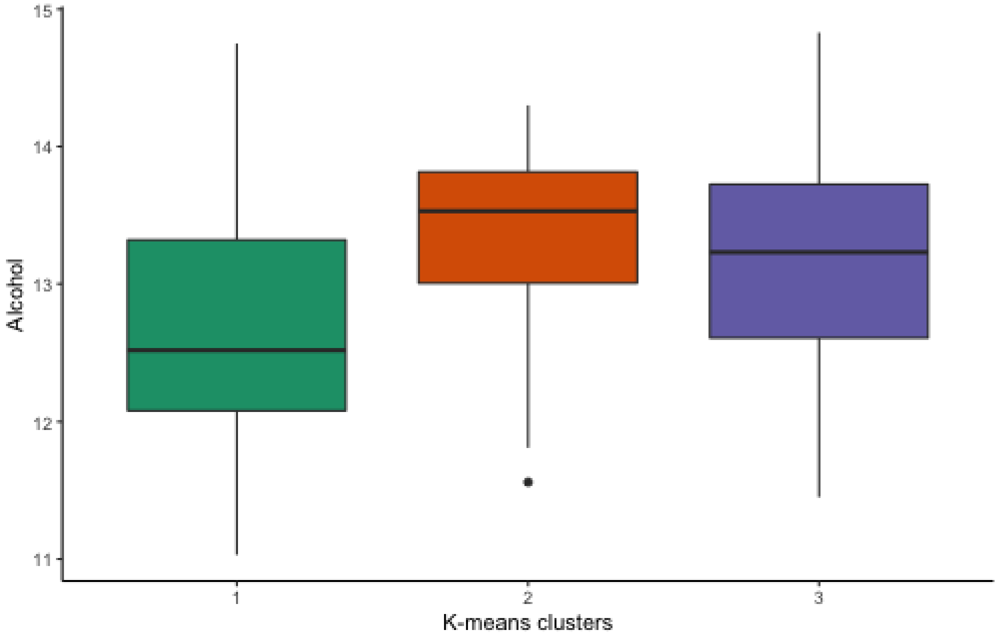
```
wine_k3$cluster
##   [1] 2 3 3 3 2 3 3 2 3 3 3 1 1 1 3 3 2 2 3 2 2 3 3 1 3 2 1 1 3 3 3 3 3 2 3
##  [36] 3 3 3 3 2 2 1 3 3 3 3 3 3 3 3 1 1 3 2 2 2 2 3 3 1 3 3 1 1 3 3 1 1 3 2
##  [71] 3 1 1 2 3 3 1 3 2 3 1 1 1 1 1 3 1 1 1 1 1 1 1 1 3 2 2 1 1 1 3 1 3 1 1
## [106] 1 1 1 1 1 3 1 3 1 1 1 1 3 1 1 3 2 3 1 1 1 1 1 1 1 2 3 3 3 1 1 1 3 1 3
## [141] 3 1 3 1 3 3 1 1 1 3 2 3 2 3 3 1 1 3 3 1 1 3 3 3 1 1 3 1 3 3 3 1 1 3 3
## [176] 2 2 3

wine %>%
  mutate(cluster_k3 = factor(wine_k3$cluster)) -> wine_with_clusters
wine_with_clusters %>%select(Alcohol, Cultivar, cluster_k3)
## # A tibble: 178 x 3
##    Alcohol Cultivar cluster_k3
##      <dbl> <chr>    <fct>
##  1    14.2 A        2
##  2    13.2 A        3
##  3    13.2 A        3
##  4    14.4 A        3
##  5    13.2 A        2
##  6    14.2 A        3
##  7    14.4 A        3
##  8    14.1 A        2
##  9    14.8 A        3
## 10    13.9 A        3
## # … with 168 more rows
```

# Average column values within each cluster?

```
wine_k3$centers
##    Alcohol MalicAcid      Ash Magnesium TotalPhenol Flavanoids
## 1 12.70356  2.298356 2.257671  87.21918    2.136438   1.829315
## 2 13.32769  2.066538 2.511923 125.11538    2.531538   2.463462
## 3 13.16747  2.460253 2.419241 102.96203    2.363924   2.071139
##   NonflavPhenols    Color
## 1      0.3880822 4.363973
## 2      0.3065385 5.521154
## 3      0.3558228 5.547089
```

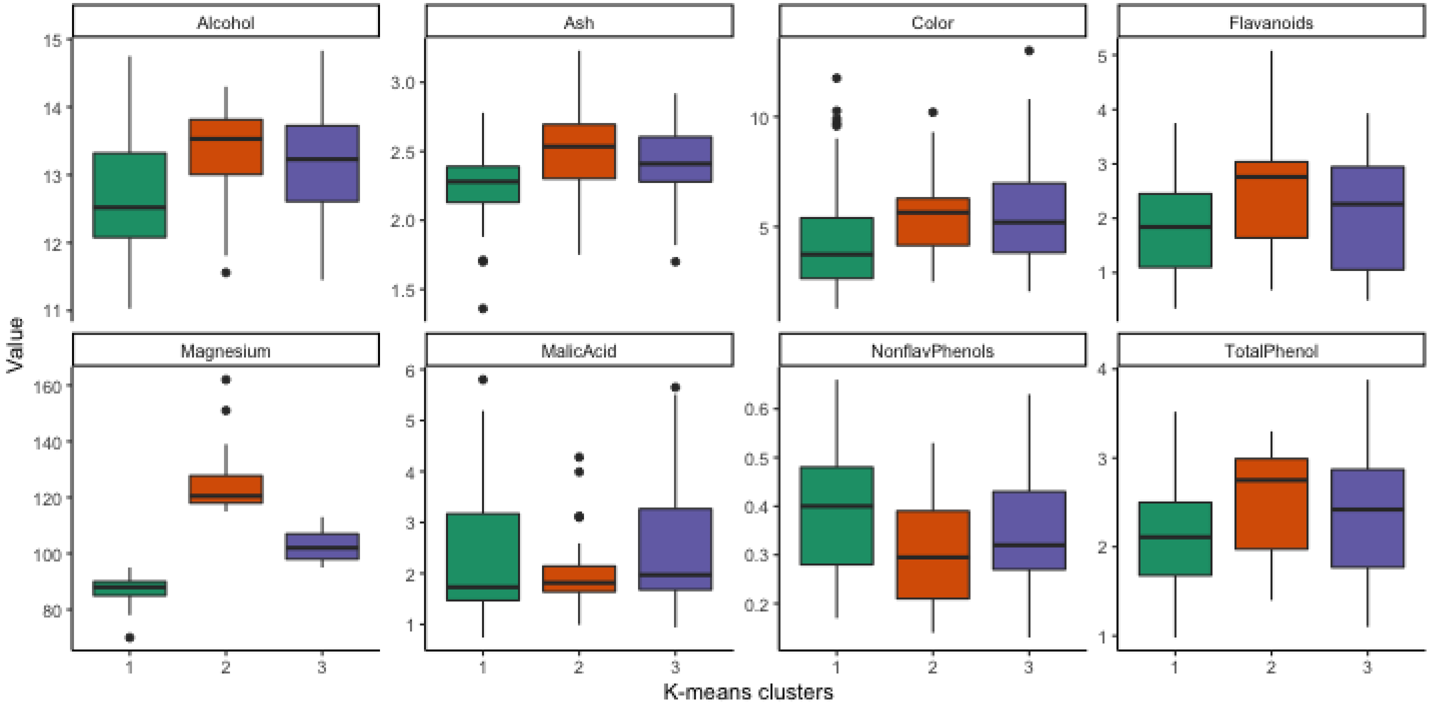
# Visualizing the clustering: distributions of variables across clusters

```
ggplot(wine_with_clusters, aes(x = cluster_k3, y = Alcohol, fill = cluster_k3)) +
  geom_boxplot() +
  labs(x = "K-means clusters") +
  scale_fill_brewer(palette = "Dark2") +
  theme(legend.position = "none")
```



```
names(wine_with_clusters)
##  [1] "Cultivar"       "Alcohol"        "MalicAcid"      "Ash"
##  [5] "Magnesium"      "TotalPhenol"    "Flavanoids"     "NonflavPhenols"
##  [9] "Color"          "cluster_k3"
wine_with_clusters %>%
  pivot_longer(Alcohol:Color, names_to = "quantity", values_to = "value")
## # A tibble: 1,424 x 4
##    Cultivar cluster_k3 quantity        value
##    <chr>    <fct>      <chr>           <dbl>
##  1 A        2          Alcohol         14.2
##  2 A        2          MalicAcid        1.71
##  3 A        2          Ash              2.43
##  4 A        2          Magnesium      127
##  5 A        2          TotalPhenol      2.8
##  6 A        2          Flavanoids       3.06
##  7 A        2          NonflavPhenols   0.28
##  8 A        2          Color            5.64
##  9 A        3          Alcohol         13.2
## 10 A        3          MalicAcid        1.78
## # … with 1,414 more rows
```

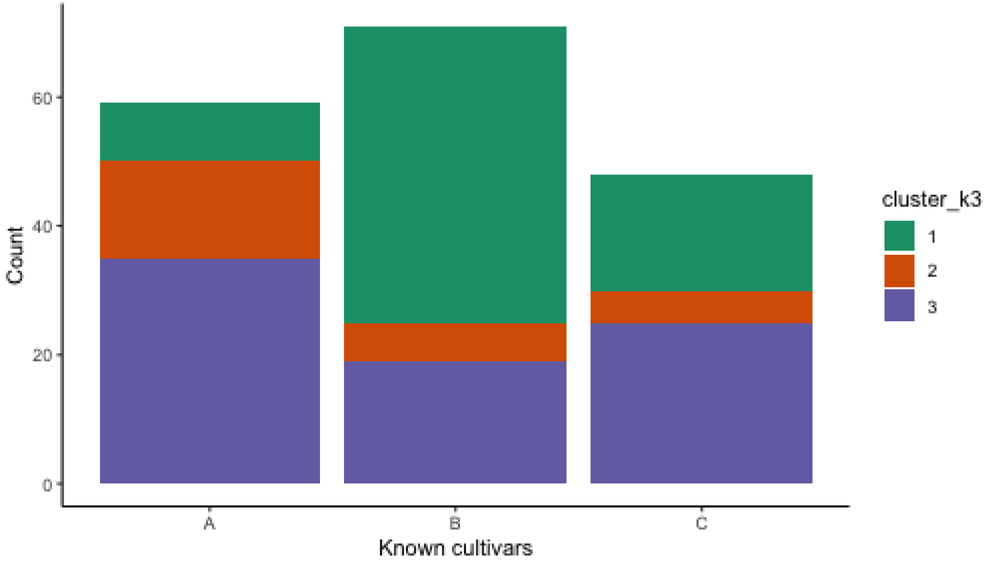
```
wine_with_clusters %>%
  pivot_longer(Alcohol:Color, names_to = "quantity", values_to = "value") %>%
  ggplot(aes(x = cluster_k3, y = value, fill = cluster_k3)) +
    geom_boxplot() +
    theme(legend.position = "none") +
    labs(x = "K-means clusters", y = "Value") +
    scale_fill_brewer(palette = "Dark2") +
    ## different Y-axis for each panel in grid
    facet_wrap(~quantity, scales = "free_y", nrow=2)
```



# Visualizing the clustering: compare clusters with any other known groupings

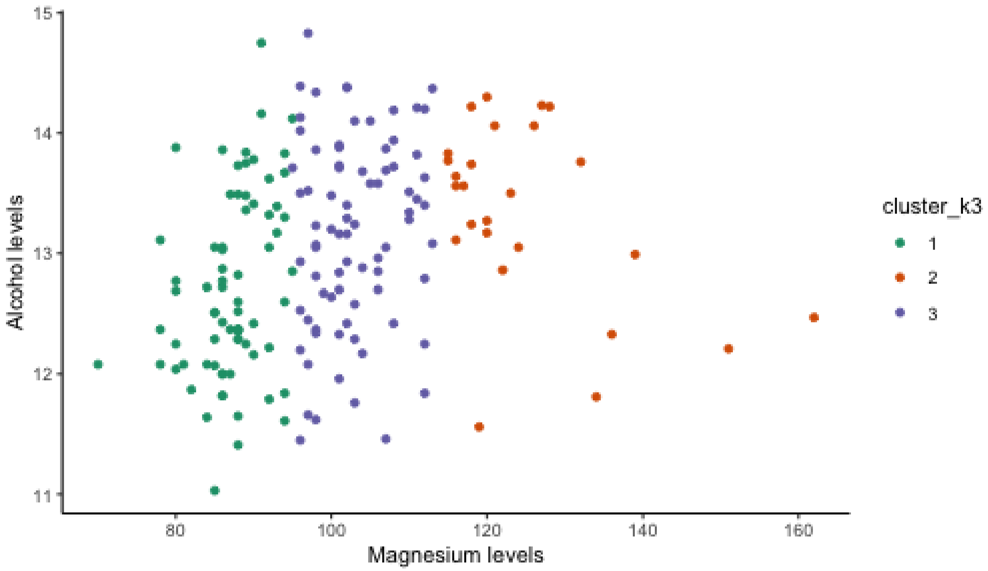
- Does clustering match with the known cultivars? **Not really!**

```
ggplot(wine_with_clusters, aes(x = Cultivar, fill = cluster_k3)) +
  geom_bar() +
  scale_fill_brewer(palette = "Dark2") +
  xlab("Known cultivars") + ylab("Count") -> bark
bark
```



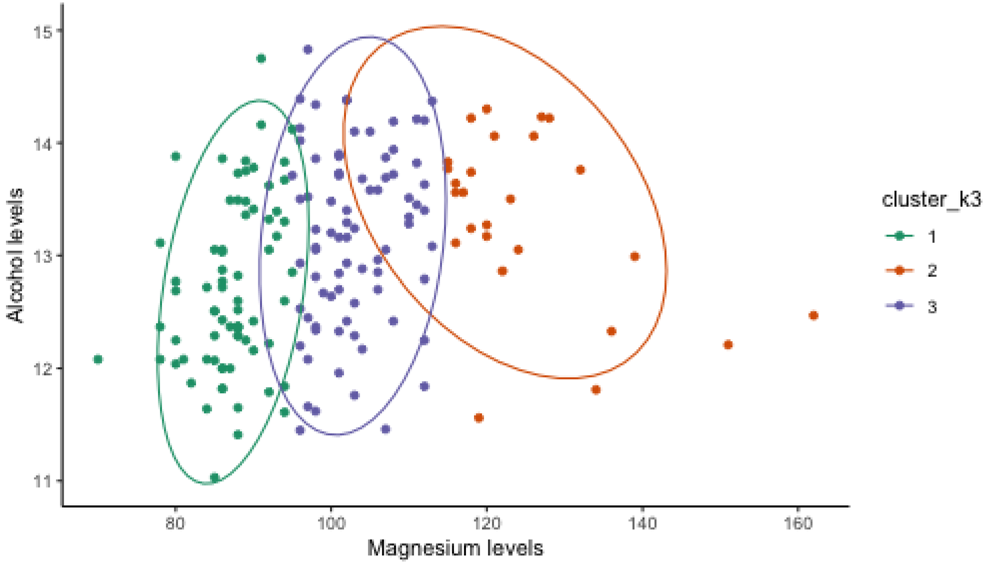
# Plot variables against each other

```
ggplot(wine_with_clusters, aes(x = Magnesium,
                               y = Alcohol,
                               color = cluster_k3)) +
  geom_point() +
  scale_color_brewer(palette = "Dark2") +
  xlab("Magnesium levels") + ylab("Alcohol levels")
```



# Plot variables against each other

```
ggplot(wine_with_clusters, aes(x = Magnesium,
                               y = Alcohol,
                               color = cluster_k3)) +
  geom_point() +
  scale_color_brewer(palette = "Dark2") +
  xlab("Magnesium levels") + ylab("Alcohol levels") +
  stat_ellipse()
```



# K means is STOCHASTIC (random)

```
wine %>%
  select(-Cultivar) %>%
  kmeans(3) -> wine_k3_secondtime

wine %>%
  mutate(new_clusters = factor(wine_k3_secondtime$cluster)) %>%
  ggplot(aes(x = Cultivar, fill = new_clusters)) +
    geom_bar() +
    scale_fill_brewer(palette = "Dark2") +
    xlab("Known cultivars") + ylab("Count") -> bark_secondtime

bark + bark_secondtime
```

# Choosing the right $k$: .........

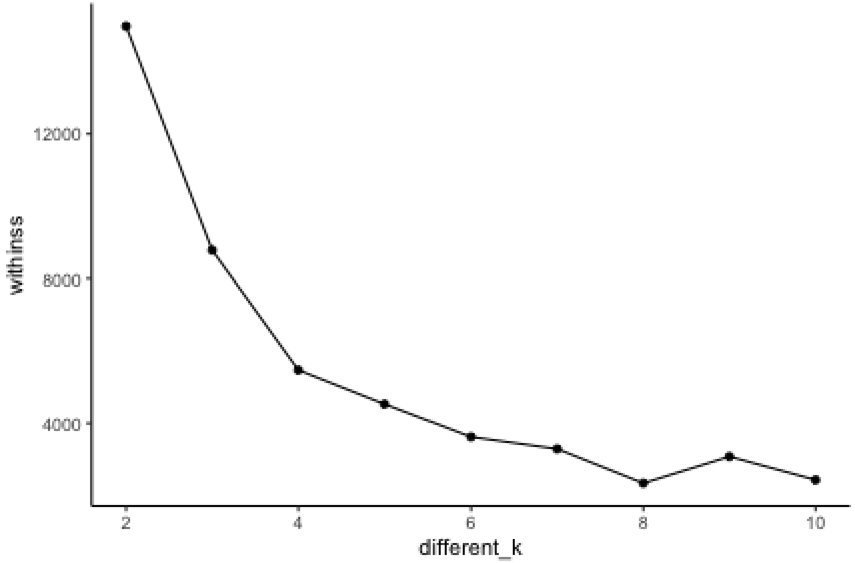
# Choosing the right k: .........

- Using sum of squares and the "elbow method"

```
wine_k3$withinss
## [1] 2256.940 3462.511 2843.740
wine_k3$tot.withinss
## [1] 8563.191
wine_k3$betweenss
## [1] 29093.74
wine_k3$totss
## [1] 37656.93
```

```
numeric_wine <- wine %>% select(-Cultivar)
run_wine_kmeans <- function(k)
{
  output_kmeans <- kmeans(numeric_wine, k)
  output_kmeans$tot.withinss[[1]]
}

tibble(different_k = 2:10) %>%
  mutate(withinss = map_dbl(different_k, run_wine_kmeans))
## # A tibble: 9 x 2
##   different_k withinss
##         <int>    <dbl>
## 1           2   14979.
## 2           3    8783.
## 3           4    5472.
## 4           5    4228.
## 5           6    3763.
## 6           7    2937.
## 7           8    2668.
## 8           9    2472.
## 9          10    2976.
```

```
tibble(different_k = 2:10) %>%
  mutate(withinss = map_dbl(different_k, run_wine_kmeans)) %>%
  ggplot(aes(x = different_k, y = withinss)) +
    geom_point() + geom_line()
```



- This approach is incredibly unsatisfying. It is also the easiest to do.
- There is no possibly way to know if more complex approaches "get it right"!!!