
INFO 2950: Intro to Data Science

Lecture 21
2023-11-08

Agenda

1. Naïve Bayes Classifier in action
2. Logistic Regression (Review)
3. Clustering
4. Phase IV/V rubric

Probabilities in the wild

- So far, we've been using probabilistic thinking to support data arguments in...

Example: halloween candy

Which bowl did we grab candy from?

Categories: Houses

Observations: a list of candy brands from a bag

Example: Yelp reviews

Is this review positive or negative?

Categories: Positive, Negative

Observations: a list of words from a new review

Example: baby names

The US Social Security Administration published the frequency of given names for each year for all names that occur at least five times for anyone with a US social security number

$$P(\text{"David"} \mid 1932) = \frac{\text{\# people with name David born in 1932}}{\text{\# people born in 1932}}$$

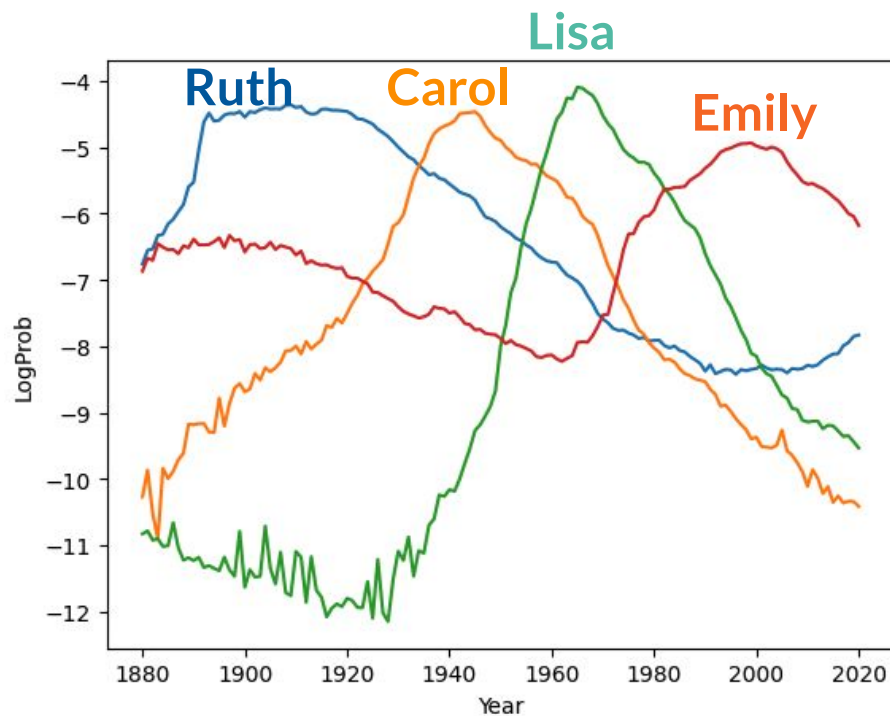
**Rank these in order by age in the
United States**

Carol Emily Lisa Ruth

Probably...

Ruth > Carol > Lisa > Emily

Pr(name|year) peaks



Example: baby names

In which year were current 2950 students born?

Categories: Years, 1980-2010 (equally likely?)

Observations: a list of given names from the 2950 class roster

Example: baby names

In which year were current 2950 students born?

Categories: Years, 1980-2010 (equally likely?)

Observations: a list of given names from the 2950 class roster

**Which year has the highest probability of generating the
~200 names?**

Can we guess the average birth year of 2950 students based only on given names?

1. Yes, within 1-2 years
2. Yes, within 5-10 years
3. No, it's too random

Calculate log of
 $\Pr(\text{name} \mid \text{year})$

In-class code example

```
def year_log_prob(names, year):
    year_counts, year_total = year_data[year]
    seen = []
    unseen = []
    log_prob = 0.0
    for name in names:
        if name in year_counts:
            log_prob += np.log(year_counts[name] / year_total)
            seen.append(name)
        else:
            log_prob += np.log(4 / year_total) #assume small number of uncommon names (4 per year)
            unseen.append(name)
    return log_prob

def plot_names(counter, start_year=1970, end_year=2020):
    years = list(range(start_year, end_year + 1))
    year_log_probs = [year_log_prob(counter, year) for year in years]

    student_age_df = pd.DataFrame({"Year": years, "LogProb": year_log_probs})

    seaborn.lineplot(data=student_age_df, x="Year", y="LogProb")
```

In-class code example

```
def year_log_prob(names, year):
    year_counts, year_total = year_data[year]
    seen = []
    unseen = []
    log_prob = 0.0
    for name in names:
        if name in year_counts:
            log_prob += np.log(year_counts[name] / year_total)
            seen.append(name)
        else:
            log_prob += np.log(4 / year_total) #assume small number of uncommon names (4 per year)
            unseen.append(name)
    return log_prob
```

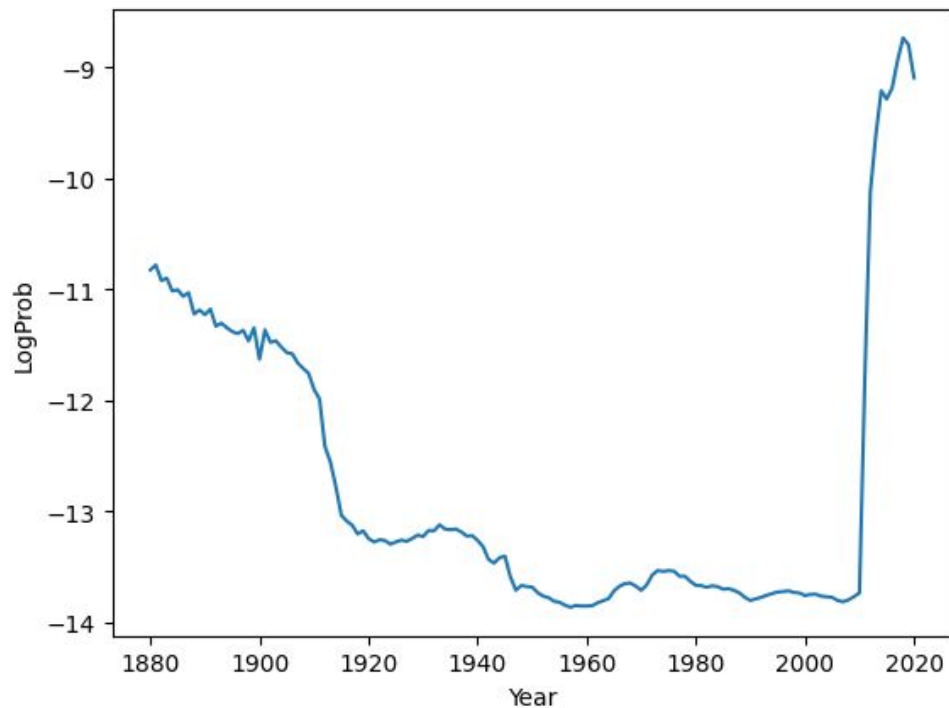
Plot the summed log probabilities given a list of names/counts called *counter*

```
def plot_names(counter, start_year=1970, end_year=2020):
    years = list(range(start_year, end_year + 1))
    year_log_probs = [year_log_prob(counter, year) for year in years]

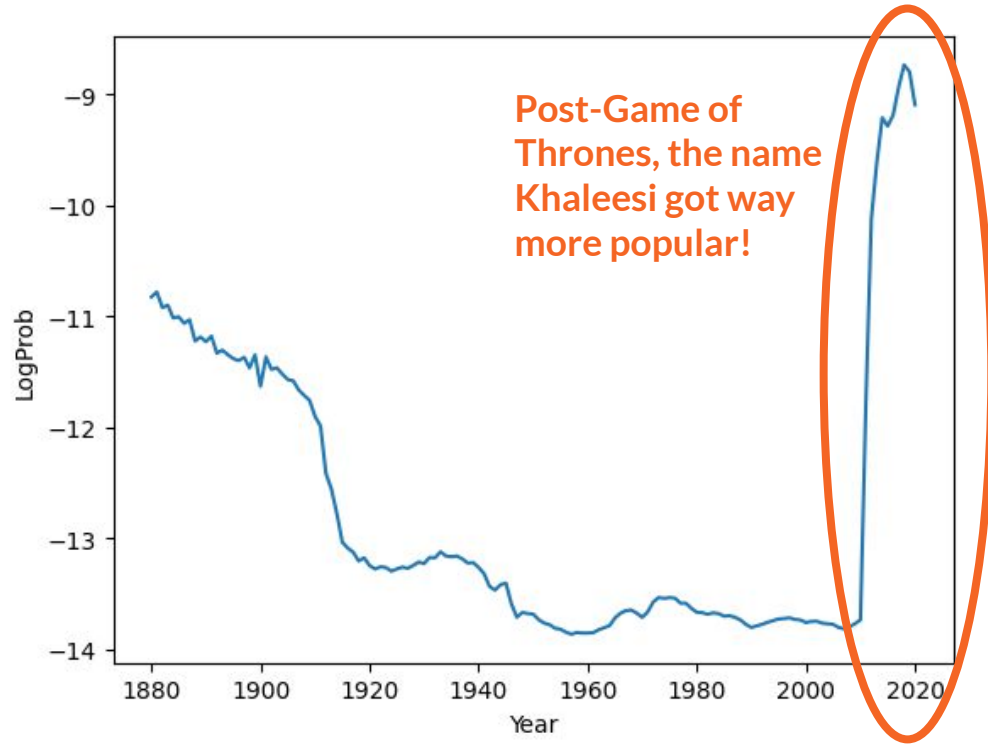
    student_age_df = pd.DataFrame({"Year": years, "LogProb": year_log_probs})

    seaborn.lineplot(data=student_age_df, x="Year", y="LogProb")
```

```
plot_names(Counter(["Khaleesi"]), start_year=1880)
```



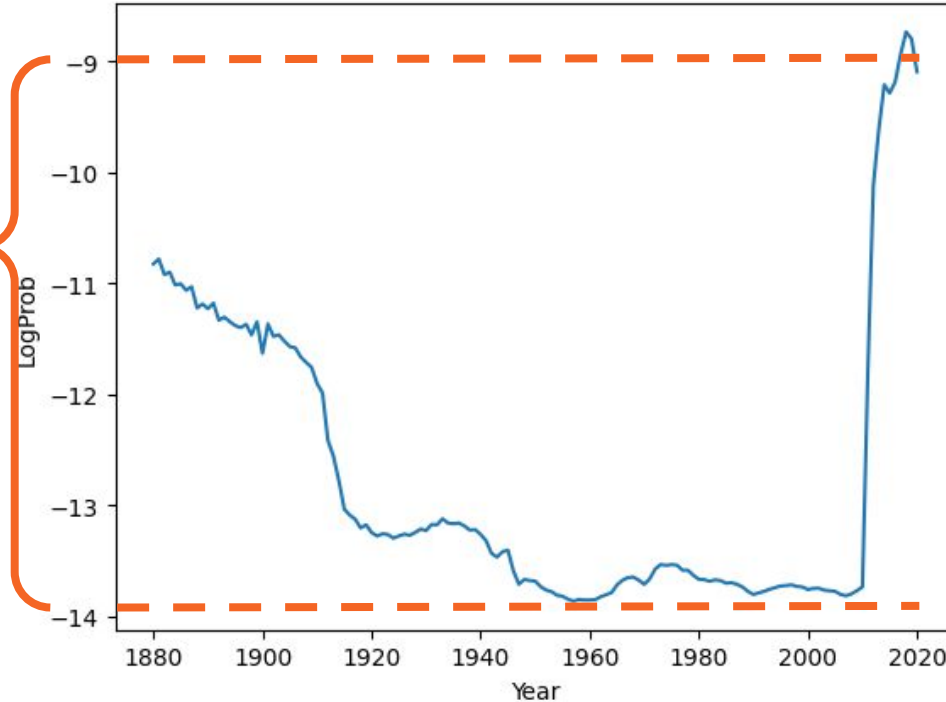
```
plot_names(Counter(["Khaleesi"]), start_year=1880)
```



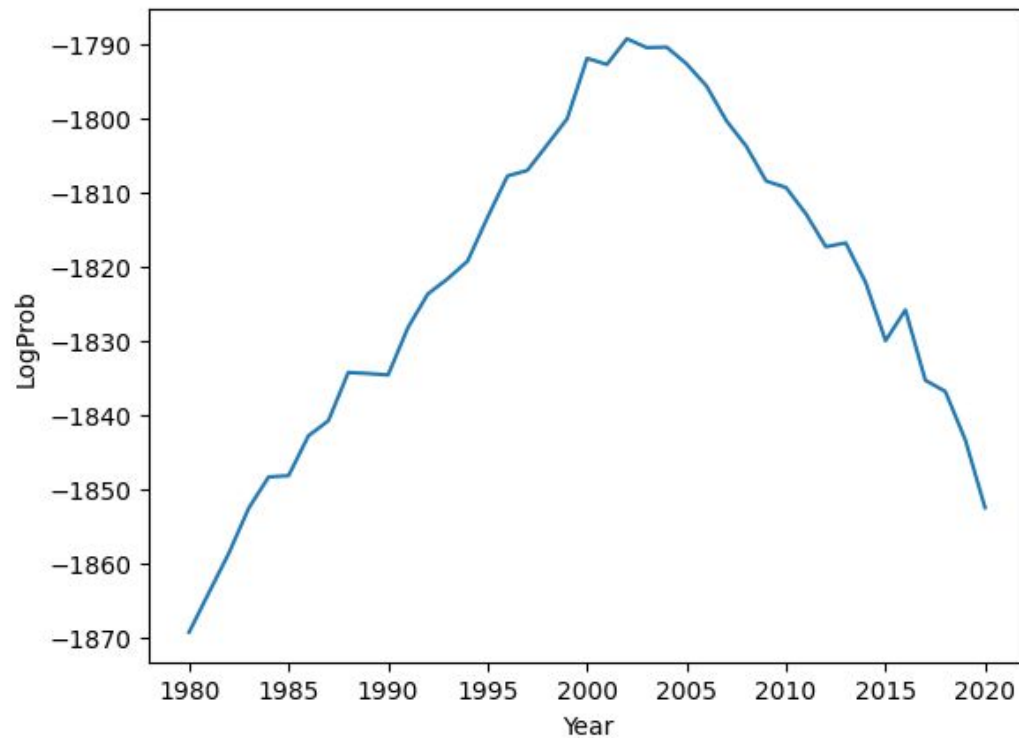

```
plot_names(Counter(["Khaleesi"]), start_year=1880)
```

The difference in log probabilities in 2020 vs. 1960 is about $-9 - (-14) = 5$

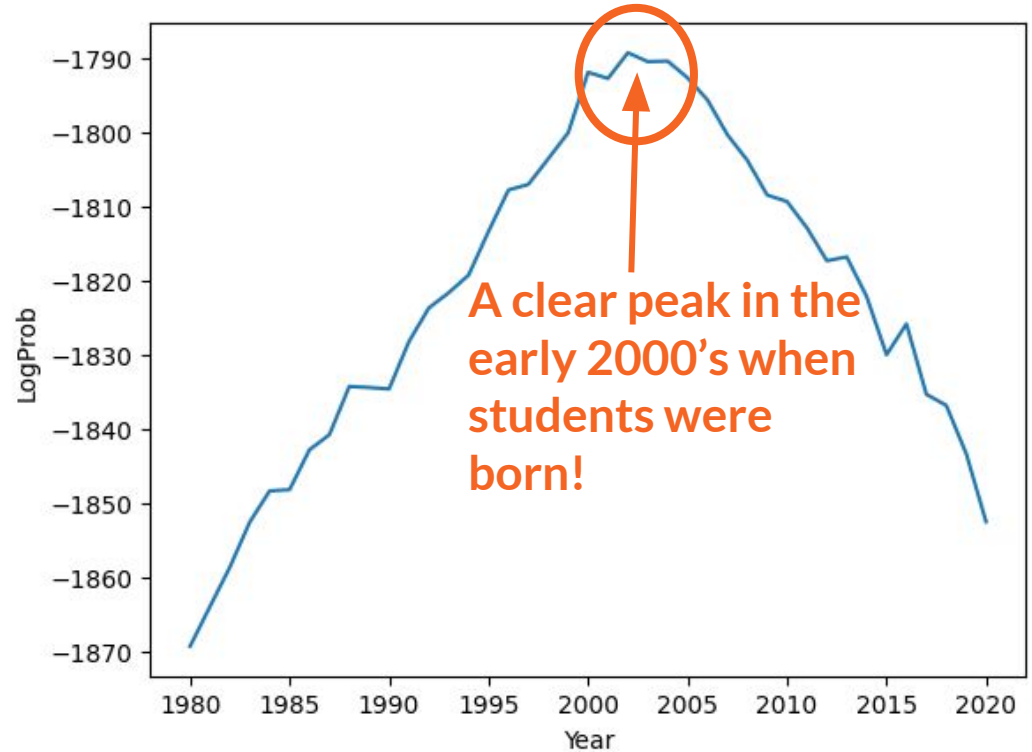
So, the name Khaleesi was about $e^5 = 148$ times more popular in 2020 than in 1960!



```
plot_names(INFO2950StudentNamesFA23, start_year=1880)
```



```
plot_names(INFO2950StudentNamesFA23, start_year=1880)
```



Naive Bayes for non-text data

- We can use Naive Bayes for other sorts of classification, too!
 - For some **dataframe with multiple inputs** x :
 - Apply the independent x 's assumption in Bayes' rule
 - Generate a contingency table
 - Calculate relevant probabilities

Naive Bayes for non-text data


Training data

Type	Freezing	Windy	Wet	Total
Sleet	400	350	450	500
Rain	0	150	300	300
Other	100	150	50	200
Total	500	650	800	1000

Question: I have weather that is freezing, windy, and wet.
What type of weather is it?

Naive Bayes for non-text data

Training data



Type	Freezing	Windy	Wet	Total
Sleet	400	350	450	500
Rain	0	150	300	300
Other	100	150	50	200
Total	500	650	800	1000

How do we write the conditional probability for weather type being freezing, windy, and wet?

Naive Bayes for non-text data

Type	Freezing	Windy	Wet	Total
Sleet	400	350	450	500
Rain	0	150	300	300
Other	100	150	50	200
Total	500	650	800	1000

$P(\text{Type} \mid \text{Freezing, Windy, Wet})$

For example:

$P(\text{Sleet} \mid \text{Freezing, Windy, Wet})$

$P(\text{Rain} \mid \text{Freezing, Windy, Wet})$

$P(\text{Other} \mid \text{Freezing, Windy, Wet})$

Naive Bayes for non-text data

Type	Freezing	Windy	Wet	Total
Sleet	400	350	450	500
Rain	0	150	300	300
Other	100	150	50	200
Total	500	650	800	1000

$P(\text{Type} \mid \text{Freezing, Windy, Wet})$

What is the Naive Bayes (NB) probability? **a** ...

$$P(y|x_1, \dots, x_n) \propto P(y) \prod_{i=1}^n P(x_i|y)$$

Naive Bayes for non-text data

Type	Freezing	Windy	Wet	Total
Sleet	400	350	450	500
Rain	0	150	300	300
Other	100	150	50	200
Total	500	650	800	1000

$P(\text{Type} \mid \text{Freezing, Windy, Wet})$

a $P(\text{Freezing} \mid \text{Type})$
* $P(\text{Windy} \mid \text{Type})$
* $P(\text{Wet} \mid \text{Type})$
* $P(\text{Type})$

Let's start by looking at Type = Sleet.

Naive Bayes for non-text data

Type	Freezing	Windy	Wet	Total
Sleet	400	350	450	500
Rain	0	150	300	300
Other	100	150	50	200
Total	500	650	800	1000

$P(\text{Sleet} \mid \text{Freezing, Windy, Wet})$

a

- $P(\text{Freezing} \mid \text{Sleet}) = 4/5$
- $* P(\text{Windy} \mid \text{Sleet}) = 7/10$
- $* P(\text{Wet} \mid \text{Sleet}) = 9/10$
- $* P(\text{Sleet}) = 1/2$

Naive Bayes for non-text data

Type	Freezing	Windy	Wet	Total
Sleet	400	350	450	500
Rain	0	150	300	300
Other	100	150	50	200
Total	500	650	800	1000

$P(\text{Sleet} \mid \text{Freezing, Windy, Wet})$

a $P(\text{Freezing} \mid \text{Sleet}) = 4/5$
* $P(\text{Windy} \mid \text{Sleet}) = 7/10$
* $P(\text{Wet} \mid \text{Sleet}) = 9/10$
* $P(\text{Sleet}) = 1/2$

Naive Bayes for non-text data

$P(\text{Sleet} \mid \text{Freezing, Windy, Wet}) \propto 4/5 * 7/10 * 9/10 * 1/2 = 0.252$

$P(\text{Rain} \mid \text{Freezing, Windy, Wet}) \propto 0$

$P(\text{Other} \mid \text{Freezing, Windy, Wet}) \propto 0.01875$

Naive Bayes for non-text data

$P(\text{Sleet} \mid \text{Freezing, Windy, Wet}) \propto 4/5 * 7/10 * 9/10 * 1/2 = 0.252$

$P(\text{Rain} \mid \text{Freezing, Windy, Wet}) \propto 0$

$P(\text{Other} \mid \text{Freezing, Windy, Wet}) \propto 0.01875$

Why do we ignore the denominator and use “proportional to” for these probabilities?

Naive Bayes for non-text data

$P(\text{Sleet} \mid \text{Freezing, Windy, Wet}) \propto 4/5 * 7/10 * 9/10 * 1/2 = 0.252$

$P(\text{Rain} \mid \text{Freezing, Windy, Wet}) \propto 0$

$P(\text{Other} \mid \text{Freezing, Windy, Wet}) \propto 0.01875$

Why do we ignore the denominator and use “proportional to” for these probabilities?

For all three of the conditional probabilities, the denominator would just normalize (i.e. divides by ~ 0.27) since $P(\text{Sleet} \mid \text{F,W,W}) + P(\text{Rain} \mid \text{F,W,W}) + P(\text{O} \mid \text{F,W,W}) = 1$

Naive Bayes for non-text data

$P(\text{Sleet} \mid \text{Freezing, Windy, Wet}) \propto 4/5 * 7/10 * 9/10 * 1/2 = 0.252$

$P(\text{Rain} \mid \text{Freezing, Windy, Wet}) \propto 0$ 0.252+0+0.01875 \approx 0.27

$P(\text{Other} \mid \text{Freezing, Windy, Wet}) \propto 0.01875$

Why do we ignore the denominator and use “proportional to” for these probabilities?

For all three of the conditional probabilities, the denominator would just normalize (i.e. divides by ~ 0.27) since $P(\text{Sleet} \mid \text{F,W,W}) + P(\text{Rain} \mid \text{F,W,W}) + P(\text{O} \mid \text{F,W,W}) = 1$

Naive Bayes for non-text data

$P(\text{Sleet} \mid \text{Freezing, Windy, Wet}) \propto 4/5 * 7/10 * 9/10 * 1/2 = 0.252$

$P(\text{Rain} \mid \text{Freezing, Windy, Wet}) \propto 0$

$P(\text{Other} \mid \text{Freezing, Windy, Wet}) \propto 0.01875$

Which one of these is biggest: 0.252, 0, or 0.0187?

Naive Bayes for non-text data

$P(\text{Sleet} \mid \text{Freezing, Windy, Wet}) \propto 4/5 * 7/10 * 9/10 * 1/2 = 0.252$

$P(\text{Rain} \mid \text{Freezing, Windy, Wet}) \propto 0$

$P(\text{Other} \mid \text{Freezing, Windy, Wet}) \propto 0.01875$

Which one of these is biggest: 0.252, 0, or 0.0187?

Which one of these is biggest: $0.252/0.27$, $0/0.27$, or $0.0187/0.27$?
 $\approx 93\%$ $\approx 0\%$ $\approx 7\%$

Naive Bayes for non-text data

$P(\text{Sleet} \mid \text{Freezing, Windy, Wet}) \propto 4/5 * 7/10 * 9/10 * 1/2 = 0.252$

$P(\text{Rain} \mid \text{Freezing, Windy, Wet}) \propto 0$

$P(\text{Other} \mid \text{Freezing, Windy, Wet}) \propto 0.01875$

Which one of these is biggest: 0.252, 0, or 0.0187?

Which one of these is biggest: $0.252/0.27 \approx 93\%$, $0/0.27 \approx 0\%$, or $0.0187/0.27 \approx 7\%$?

Naive Bayes for non-text data

$P(\text{Sleet} \mid \text{Freezing, Windy, Wet}) \propto 4/5 * 7/10 * 9/10 * 1/2 = 0.252$

$P(\text{Rain} \mid \text{Freezing, Windy, Wet}) \propto 0$

$P(\text{Other} \mid \text{Freezing, Windy, Wet}) \propto 0.01875$

A constant denominator (unaffected by weather type) can be ignored when comparing probability magnitudes!

Naive Bayes for non-text data

$$P(\text{Sleet} \mid \text{Freezing, Windy, Wet}) \propto 4/5 * 7/10 * 9/10 * 1/2 = 0.252$$

$$P(\text{Rain} \mid \text{Freezing, Windy, Wet}) \propto 0$$

$$P(\text{Other} \mid \text{Freezing, Windy, Wet}) \propto 0.01875$$

The probability is highest for weather type = Sleet, so using Naive Bayes, we would classify a freezing, windy, and wet day as having sleet

Naive Bayes for non-text data

$P(\text{Sleet} \mid \text{Freezing, Windy, Wet}) \propto 4/5 * 7/10 * 9/10 * 1/2 = 0.252$

$P(\text{Rain} \mid \text{Freezing, Windy, Wet}) \propto 0$

$P(\text{Other} \mid \text{Freezing, Windy, Wet}) \propto 0.01875$

Are we missing anything we talked about last class?

Naive Bayes for non-text data

Type	Freezing	Windy	Wet	Total
Sleet	400	350	450	500
Rain	0	150	300	300
Other	100	150	50	200
Total	500	650	800	1000

1.) Note that we have a 0 here, so we'd want to do Laplace correction!

Naive Bayes for non-text data

$P(\text{Sleet} \mid \text{Freezing, Windy, Wet}) \propto 4/5 * 7/10 * 9/10 * 1/2 = 0.252$

$P(\text{Rain} \mid \text{Freezing, Windy, Wet}) \propto 0$

$P(\text{Other} \mid \text{Freezing, Windy, Wet}) \propto 0.01875$

2.) It's best to compare log probabilities to avoid small number issues (we got lucky again here)

Naive Bayes

- **Naive Bayes allows us to classify our outcomes**
 - These can be binary (Sports / not sports) or multi-category (sleet / rain / other)
 - We calculate probabilities based on the frequencies of these categories in our data for each independent input x (whether x_i 's are words or weather characteristics)

Naive Bayes

- Naive Bayes allows us to classify our outcomes
 - These can be binary (Sports / not sports) or multi-category (sleet / rain / other)
 - We calculate probabilities based on the frequencies of these categories in our data for each independent input x (whether x_i 's are words or weather characteristics)

What if we have
numeric input
 X 's instead?

Gaussian Naive Bayes

- Naive Bayes allows us to classify our outcomes

- These can be binary (Sports / not sports) or multi-category (sleet / rain / other)

- We calculate probabilities based on the

normal probability density function ~~frequencies of these categories~~ in our data for each independent input x (whether x_i 's are words or weather characteristics)

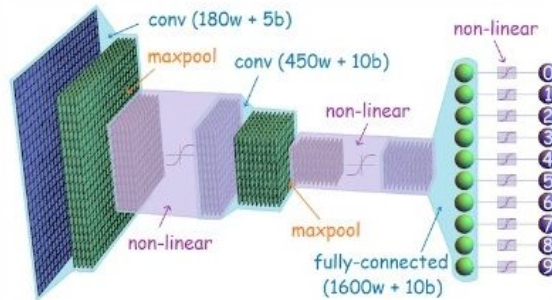
Classification

- Aside from Naive Bayes, we've learned about “classifying” with one other model in class
 - **Logistic regression** (for binary output)!
- How does Naive Bayes compare to logistic regression?

One minute break!

WHO WOULD WIN?

**AN INCREDIBLY COMPLEX
MULTI-LAYER CONVOLUTIONAL
NEURAL NETWORK**



ONE NAIVE BOI



Classification for binary output

Naive Bayes	Logistic Regression
Assume independent x 's	No independence assumption

Classification for binary output

Naive Bayes	Logistic Regression
Assume independent x 's	No independence assumption
Okay with small data	Overfits with small data

Classification for binary output

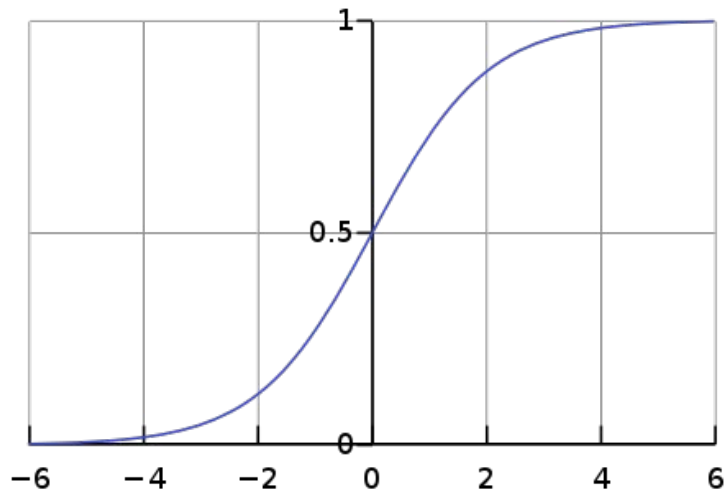
Naive Bayes	Logistic Regression
Assume independent x 's	No independence assumption
Okay with small data	Overfits with small data
Requires explicit distribution over inputs x (Bernoulli, normal)	Inputs x are just numbers

Classification for binary output

Naive Bayes	Logistic Regression
Assume independent x 's	No independence assumption
Okay with small data	Overfits with small data
Requires explicit distribution over inputs x (Bernoulli, normal)	Inputs x are just numbers
Model fitting is easy (counting words, mean/variance)	Need a gradient-based method (SGD, L-BFGS) to get coefficients α, β

Logistic Regression Refresher

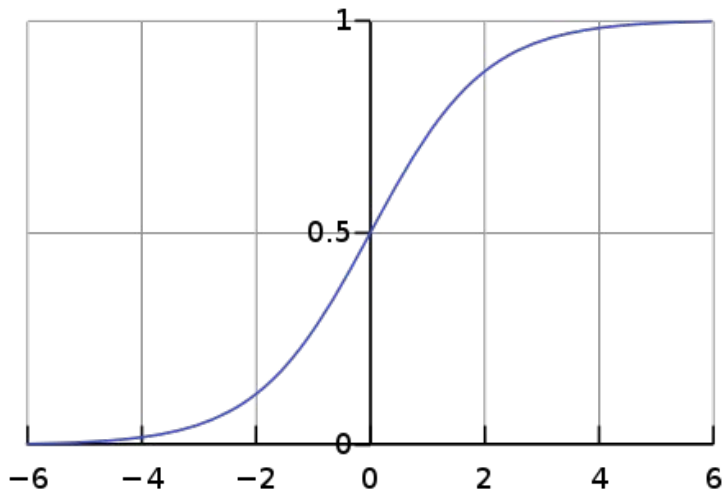
Logistic Function $\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$



Goal: to estimate whether our output is TRUE or FALSE

Logistic Regression Refresher

Logistic Function $\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$

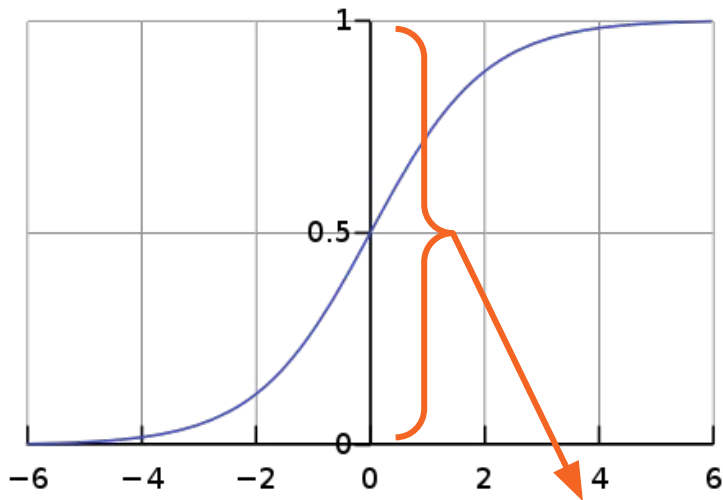


Goal: to estimate whether our output is TRUE or FALSE

Method: Make our regression tell us about the probability (between 0 and 1) that $y = \text{TRUE}$.

Logistic Regression Refresher

Logistic Function $\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$



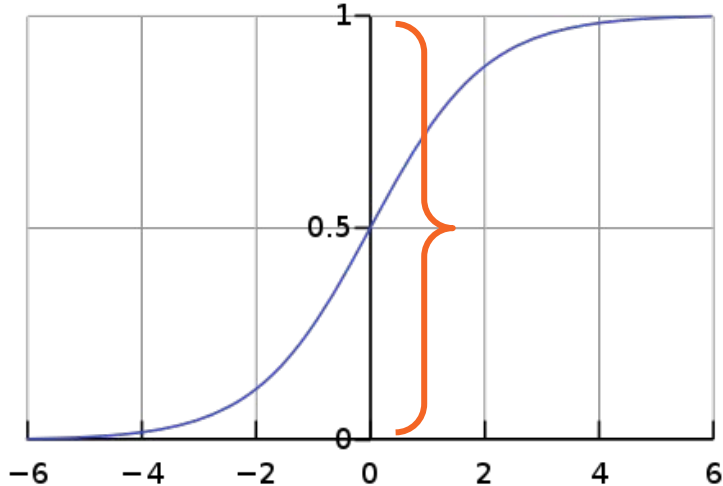
Goal: to estimate whether our output is TRUE or FALSE

Method: Make our regression tell us about the probability (between 0 and 1) that $y = \text{TRUE}$. To force our regression to give us an output between 0 and 1, we use a **transformation** on our regression's prediction for $\alpha + \beta_1 x_1 + \beta_2 x_2 + \dots$

Probability (between 0 and 1)

Logistic Regression Refresher

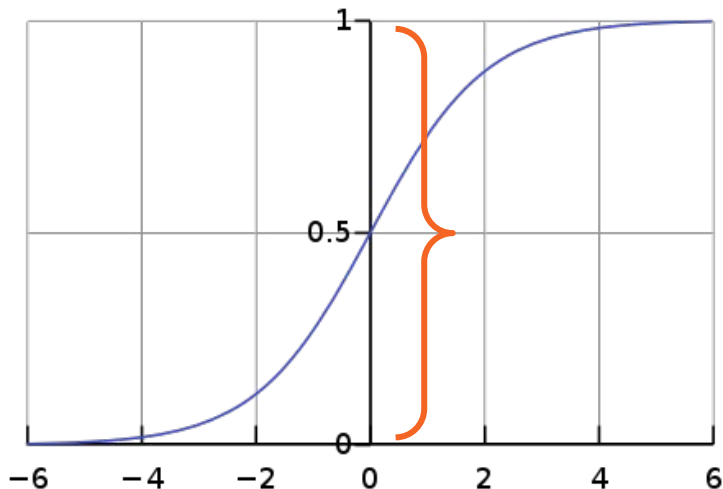
Logistic Function $\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$



We want to find some output = σ
($\alpha + \beta_1 x_1 + \beta_2 x_2 + \dots$)

Logistic Regression Refresher

Logistic Function $\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$

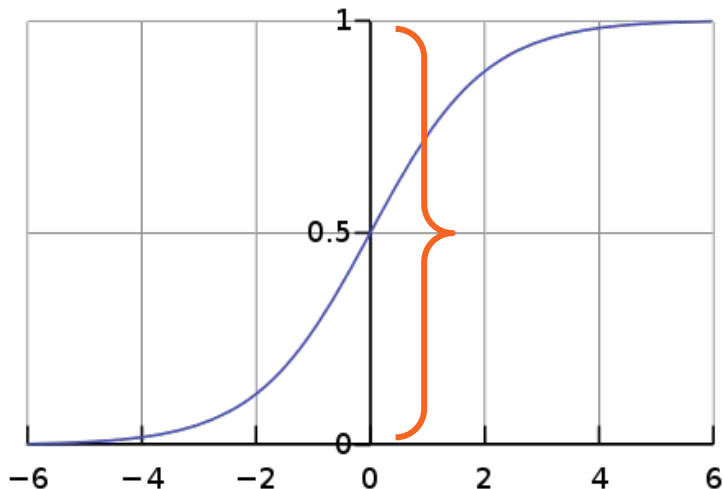


Key thing to remember: this isn't the value of y anymore, it's a probability *about* y

We want to find some output = σ
($\alpha + \beta_1 x_1 + \beta_2 x_2 + \dots$)

Logistic Regression Refresher

Logistic Function $\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$



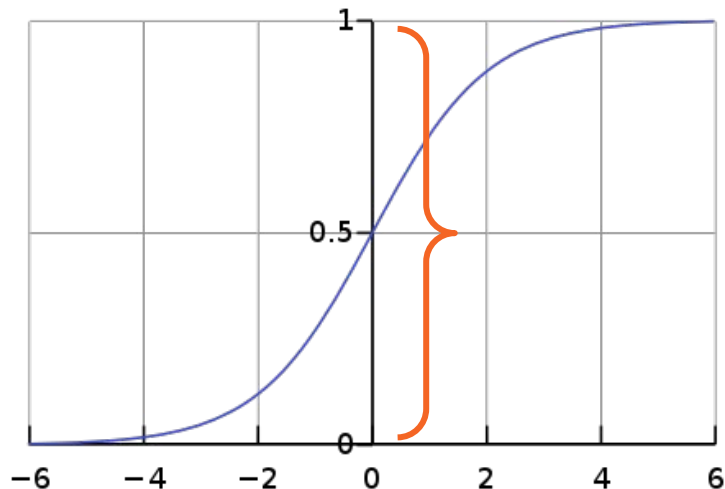
Key thing to remember: this isn't the value of y anymore, it's a probability *about* y

We want to find some output = σ
($\alpha + \beta_1 x_1 + \beta_2 x_2 + \dots$)

The probability that $y=\text{TRUE}$ is p

Logistic Regression Refresher

Logistic Function $\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$

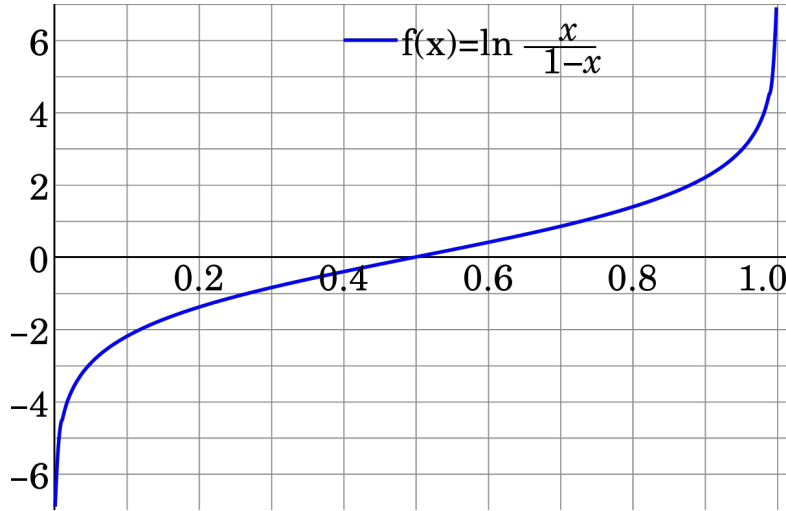


We want to find some output = σ
 $(\alpha + \beta_1 x_1 + \beta_2 x_2 + \dots)$

Remember, the “logit” function is the inverse of the sigmoid, so:

$$\text{logit}(p) = \alpha + \beta_1 x_1 + \beta_2 x_2 + \dots$$

Logistic Regression Refresher

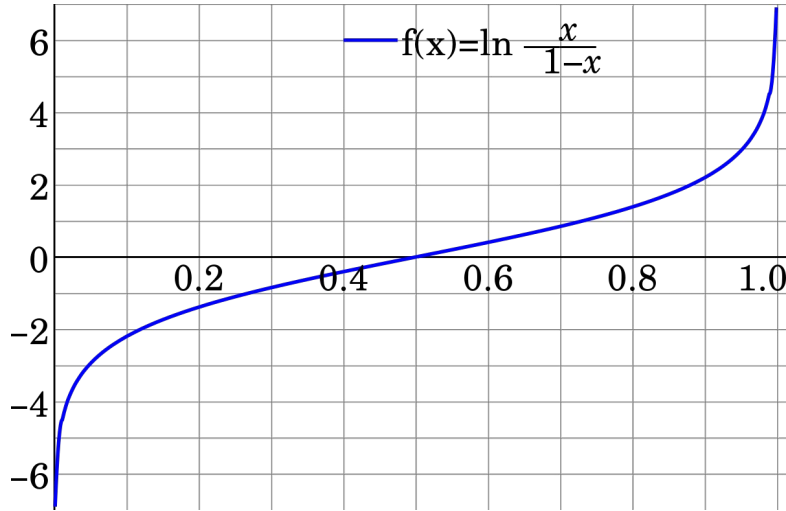


We want to find some output = σ
 $(\alpha + \beta_1 x_1 + \beta_2 x_2 + \dots)$

Remember, the “logit” function is the inverse of the sigmoid, so:

$$\text{logit}(p) = \alpha + \beta_1 x_1 + \beta_2 x_2 + \dots$$

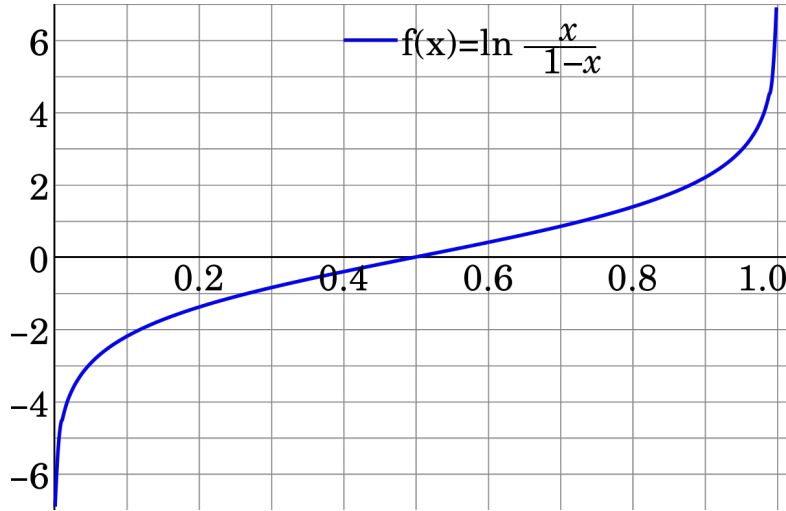
Logistic Regression Refresher



$$\text{logit}(p) = \alpha + \beta_1 x_1 + \beta_2 x_2 + \dots$$

And, $\text{logit}(p) = \log(p / [1-p])$ = the log odds ratio

Logistic Regression Refresher

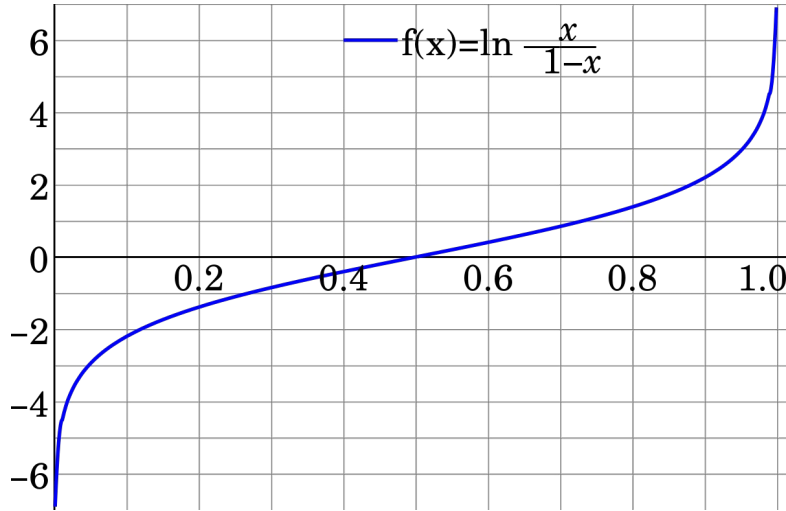


$$\text{logit}(p) = \alpha + \beta_1 x_1 + \beta_2 x_2 + \dots$$

And, $\text{logit}(p) = \log(p / [1-p])$ = the log odds ratio

Probability that y
= 1 (TRUE)

Logistic Regression Refresher

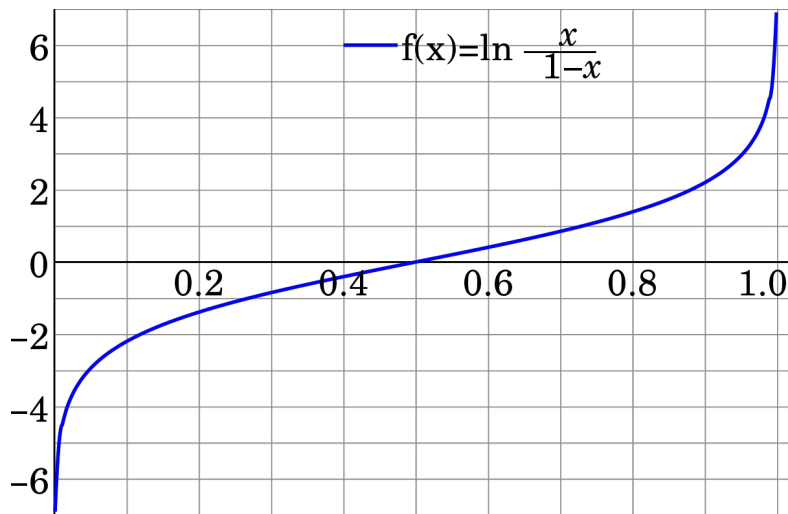


$$\text{logit}(p) = \alpha + \beta_1 x_1 + \beta_2 x_2 + \dots$$

And, $\text{logit}(p) = \log(p / [1-p])$ = the log odds ratio

Probability that y
= 0 (FALSE)

Logistic Regression Refresher

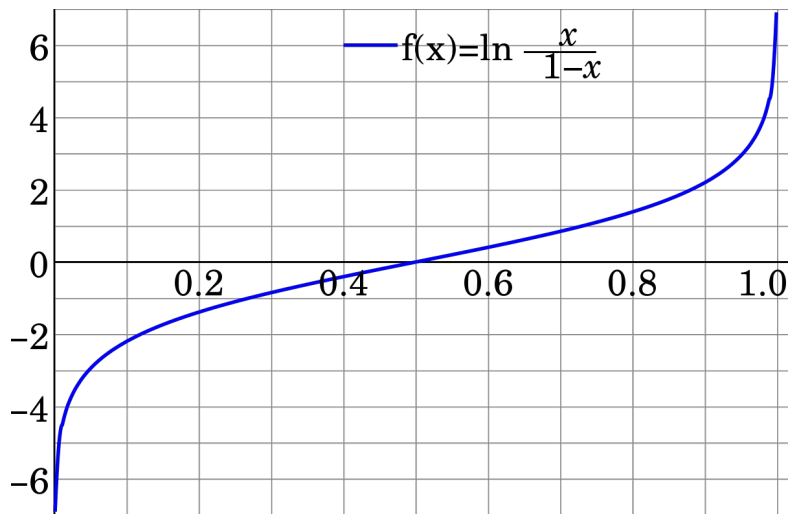


$$\text{logit}(p) = \alpha + \beta_1 x_1 + \beta_2 x_2 + \dots$$

And, $\text{logit}(p) = \log(p / [1-p])$ = the log odds ratio

A 1 unit increase in x_1 corresponds to a β_1 increase in _____

Logistic Regression Refresher



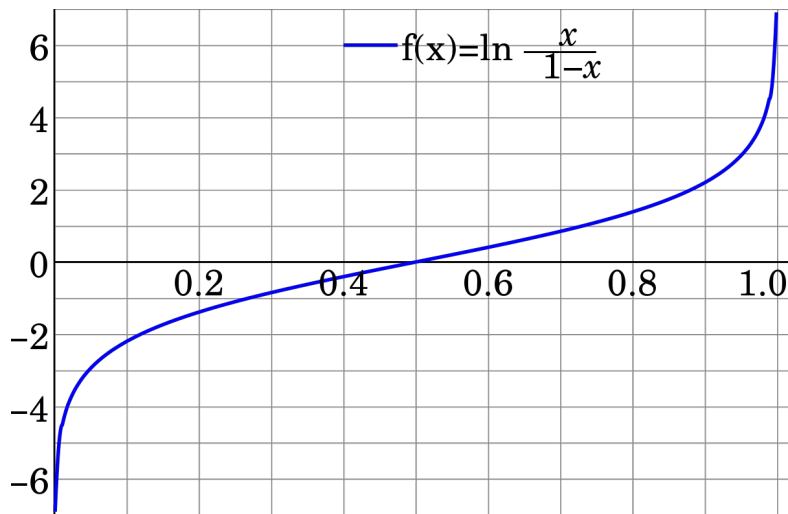
$$\text{logit}(p) = \alpha + \beta_1 x_1 + \beta_2 x_2 + \dots$$

And, $\text{logit}(p) = \log(p / [1-p])$ = the log odds ratio

A 1 unit increase in x_1 corresponds to a β_1 increase in $\text{logit}(p)$

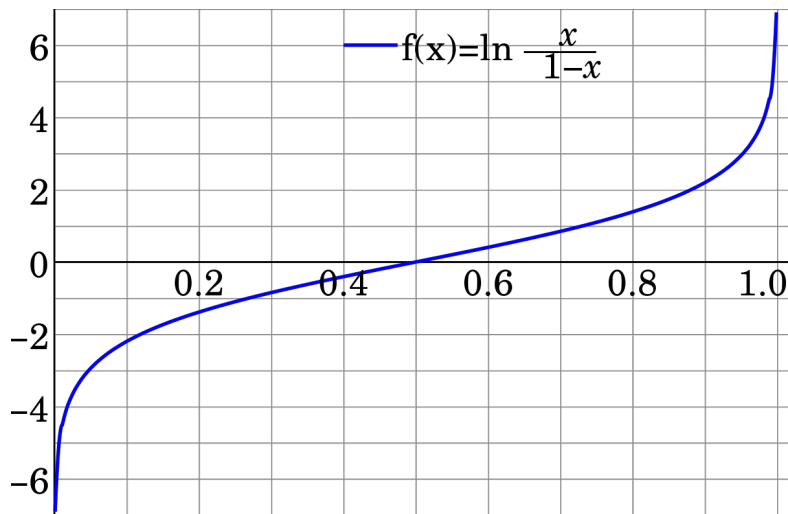
a.k.a. the *log-odds ratio* of
 $y=\text{TRUE} : y=\text{FALSE}$

Logistic Regression Refresher



What does it mean to increase the log-odds ratio by some amount?

Logistic Regression Refresher



What does it mean to increase the log-odds ratio by some amount?

Increasing the log-odds ratio also means increasing the odds ratio (logarithm is monotonically increasing!)

If increasing x_1 yields an increase in log-odds ratio, it'll also increase the odds ratio.

Logistic Regression Refresher

How do we interpret the odds ratio?

Flipping heads

Rolling a 1 on a 6-sided die

Which is more likely?

Logistic Regression Refresher

How do we interpret the odds ratio?

Flipping heads

Rolling a 1 on a 6-sided die

Which is more likely? **Flipping heads (50% chance) is more likely than rolling a 1 ($\frac{1}{6}$ chance)**

Logistic Regression Refresher

How do we interpret the odds ratio?

Flipping heads has $2:1$ odds

Rolling a 1 on a 6-sided die has $1:5$ odds

Logistic Regression Refresher

How do we interpret the odds ratio?

Flipping heads has 1:1 odds

Rolling a 1 on a 6-sided die has 1:5 odds

Logistic Regression Refresher

How do we interpret the odds ratio?

Flipping heads has **1:1** odds

Rolling a 1 on a 6-sided die has **1:5** odds

Not rolling a 1 on a 6-sided die has **5:1** odds

Logistic Regression Refresher

How do we interpret the odds ratio?

Flipping heads has 1:1 odds

Rolling a 1 on a 6-sided die has 1:5 odds

Not rolling a 1 on a 6-sided die has 5:1 odds

Logistic Regression Refresher

Odds ratio = 1 \rightarrow as likely that $y=\text{TRUE}$ (relative to $y=\text{FALSE}$)

Odds ratio < 1 \rightarrow less likely that $y=\text{TRUE}$ (relative to $y=\text{FALSE}$)

Odds ratio > 1 \rightarrow more likely that $y=\text{TRUE}$ (relative to $y=\text{FALSE}$)

How do we interpret the odds ratio?

Flipping heads has 1:1 odds

Rolling a 1 on a 6-sided die has 1:5 odds

Not rolling a 1 on a 6-sided die has 5:1 odds

Log intuition quiz!!!

If the log odds ratio is -2.3, the odds ratio is 1 to _____, which is {small, large}

If the odds ratio is 100,000 to 1 (large), the log odds ratio is _____

Log intuition quiz!!!

If the log odds ratio is -2.3, the odds ratio is 1 to __10__, which is {small}

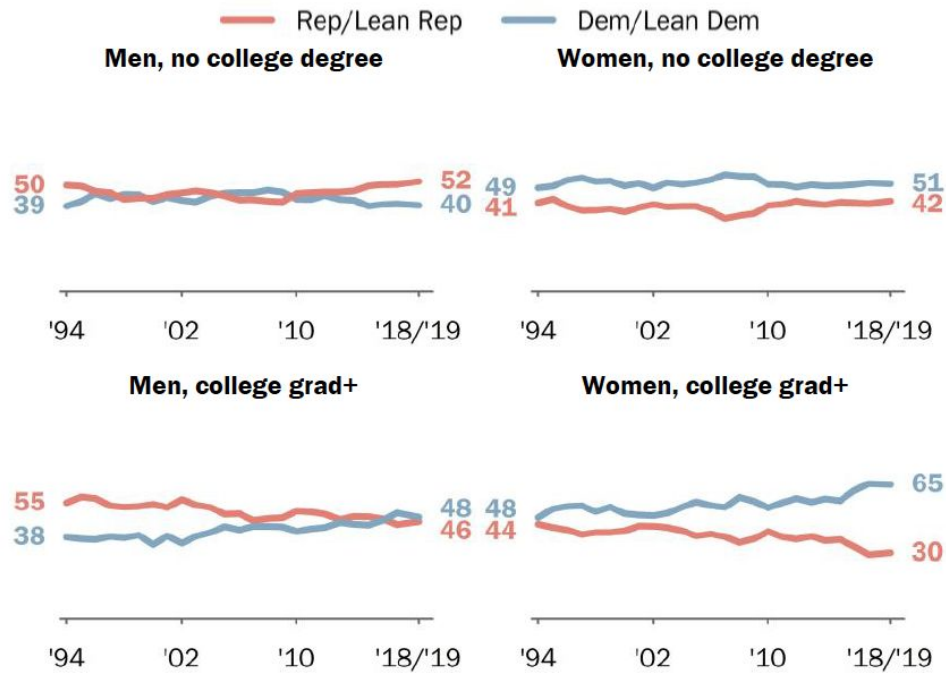
If the odd ratio is 100,000 to 1 (large), the log odds ratio is __11.5__

Logistic Regression: Example

- Now that we know how to interpret coefficients, can we derive some coefficients given probabilistic data?
 - Note: *this is different from getting coefficients given input data, which requires e.g. SGD*
- And, what about interaction effects?

US party affiliation by gender and education

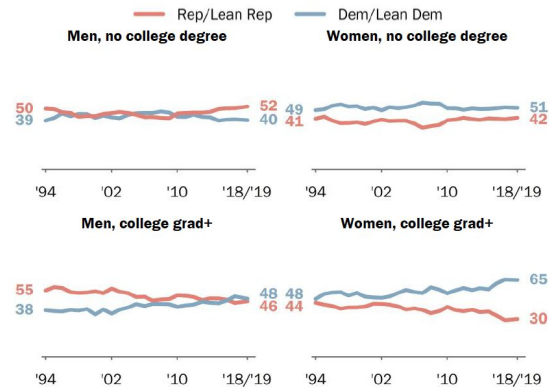
% of registered voters who identify as/lean toward ...



Example: party affiliation by gender and education

Growing Democratic advantage among women college graduates; men with a college degree remain divided

% of registered voters who identify as/lean toward ...



Notes: Based on registered voters. Due to smaller sample sizes in 2018 and 2019, the data from those years has been combined. Don't know responses not shown.
Source: Annual totals of Pew Research Center survey data (U.S. adults).

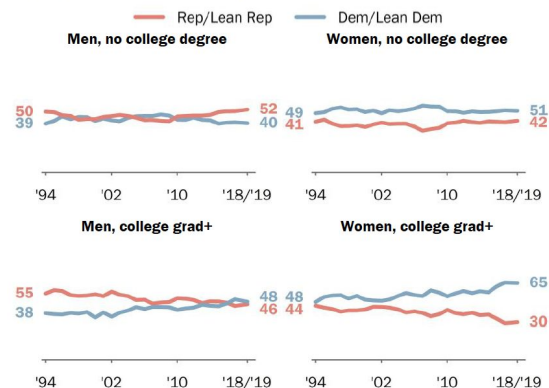
PEW RESEARCH CENTER

% Democrat	Men	Women
No college degree	40	51
College degree	48	65

Example: party affiliation by gender and education

Growing Democratic advantage among women college graduates; men with a college degree remain divided

% of registered voters who identify as/lean toward ...



Notes: Based on registered voters. Due to smaller sample sizes in 2018 and 2019, the data from those years has been combined. Don't know responses not shown.
Source: Annual totals of Pew Research Center survey data (U.S. adults).

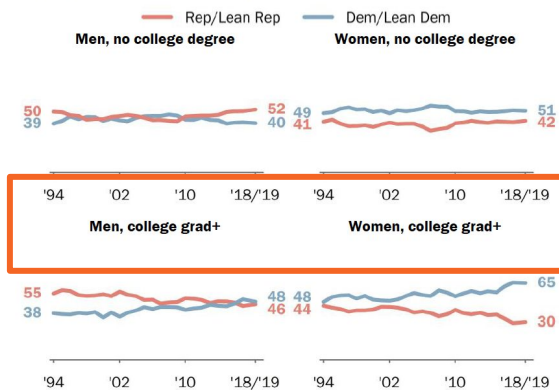
PEW RESEARCH CENTER

P(Democrat)	Men	Women
No college degree	0.40	0.51
College degree	0.48	0.65

Example: party affiliation by gender and education

Growing Democratic advantage among women college graduates; men with a college degree remain divided

% of registered voters who identify as/lean toward ...



Notes: Based on registered voters. Due to smaller sample sizes in 2018 and 2019, the data from those years has been combined. Don't know responses not shown.
Source: Annual totals of Pew Research Center survey data (U.S. adults).

PEW RESEARCH CENTER

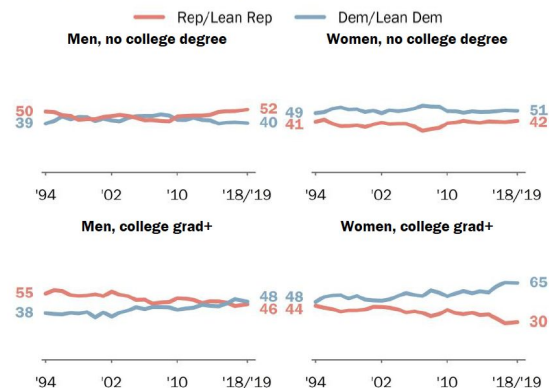
P(Democrat)	Men	Women
No college degree	0.40	0.51
College degree	0.48	0.65

For the purposes of *this study*, gender = binary

Example: party affiliation by gender and education

Growing Democratic advantage among women college graduates; men with a college degree remain divided

% of registered voters who identify as/lean toward ...



Notes: Based on registered voters. Due to smaller sample sizes in 2018 and 2019, the data from those years has been combined. Don't know responses not shown.
Source: Annual totals of Pew Research Center survey data (U.S. adults).

PEW RESEARCH CENTER

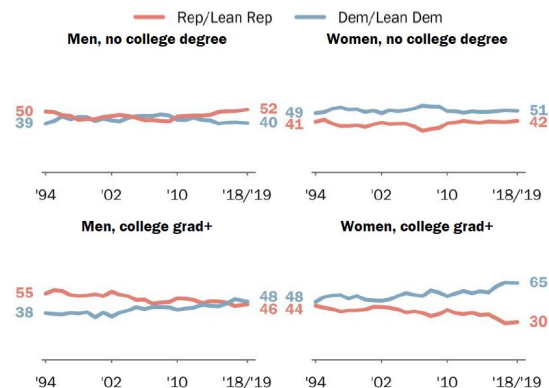
P(Democrat)	Men	Women
No college degree	0.40	0.51
College degree	0.48	0.65

Note: these rows aren't summable (is there a 116% probability that women are democrats?)

Example: party affiliation by gender and education

Growing Democratic advantage among women college graduates; men with a college degree remain divided

% of registered voters who identify as/lean toward ...



Notes: Based on registered voters. Due to smaller sample sizes in 2018 and 2019, the data from those years has been combined. Don't know responses not shown.
Source: Annual totals of Pew Research Center survey data (U.S. adults).

PEW RESEARCH CENTER

P(Democrat)	Men	Women
No college degree	0.40	0.51
College degree	0.48	0.65


Note: these columns aren't summable (is there a 113% probability that college degree holders are democrats?)

Example: party affiliation by gender and education

Specifically, this is the
 $P(\text{Democrat} \mid \text{binary college status, binary gender})$

$P(\text{Democrat})$	Men	Women
No college degree	0.40	0.51
College degree	0.48	0.65

Step 1: code categories as {0,1}



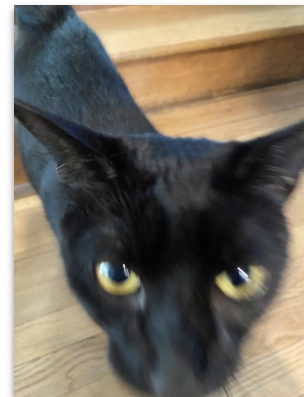
P(Democrat)	Men	Women
No college degree	0.40	0.51
College degree	0.48	0.65

It makes sense to model this as a logistic regression because our output here is a probability, i.e. the thing that the sigmoid function will return

Step 1: code categories as {0,1}

P(Democrat)	Men	Women
No college degree	0.40	0.51
College degree	0.48	0.65

Koenecke: Female=1, College=1
Thalken: Female=1, College=1
Sparky: Female=0, College=0



Step 1: code categories as {0,1}

P(Democrat)	Men	Women
No college degree	0.40	0.51
College degree	0.48	0.65

Koenecke: Female=1, College=1

Thalken: Female=1, College=1

Sparky: Female=0, College=0

Example inputs to your logistic model

x_F = Female binary

x_C = College binary

Step 1: code categories as {0,1}

P(Democrat)	Men	Women
No college degree	0.40	0.51
College degree	0.48	0.65

Koenecke: Female=1, College=1

Thalken: Female=1, College=1

Sparky: Female=0, College=0

Example inputs to your logistic model

x_F = Female binary

x_C = College binary

$$P(\text{Democrat}) = \sigma(\alpha + \beta_F x_F + \beta_C x_C)$$

Step 1: code categories as {0,1}

P(Democrat)	Men	Women
No college degree	0.40	0.51
College degree	0.48	0.65

Koenecke: Female=1, College=1
Thalken: Female=1, College=1
Sparky: Female=0, College=0

Will we need an interaction
variable Female+College?



$$P(\text{Democrat}) = \sigma(\alpha + \beta_F x_F + \beta_C x_C)$$

$$P(\text{Democrat}) = \sigma(\alpha + \beta_F x_F + \beta_C x_C + \beta_{FC} x_F * x_C)$$

Step 1: code categories as {0,1}

With no interaction terms:

P(Democrat)	Men	Women
No college degree	$\sigma(\alpha + 0\beta_F + 0\beta_C) = 0.40$	$\sigma(\alpha + 1\beta_F + 0\beta_C) = 0.51$
College degree	$\sigma(\alpha + 0\beta_F + 1\beta_C) = 0.48$	$\sigma(\alpha + 1\beta_F + 1\beta_C) = 0.65$

Step 1: code categories as {0,1}

P(Democrat)	Men	Women
No college degree	$\sigma(\alpha + 0\beta_F + 0\beta_C) = 0.40$	$\sigma(\alpha + 1\beta_F + 0\beta_C) = 0.51$
College degree	$\sigma(\alpha + 0\beta_F + 1\beta_C) = 0.48$	$\sigma(\alpha + 1\beta_F + 1\beta_C) = 0.65$

Step 1: code categories as {0,1}

P(Democrat)	Men	Women
No college degree	$\sigma(\alpha + 0\beta_F + 0\beta_C) = 0.40$	$\sigma(\alpha + 1\beta_F + 0\beta_C) = 0.51$
College degree	$\sigma(\alpha + 0\beta_F + 1\beta_C) = 0.48$	$\sigma(\alpha + 1\beta_F + 1\beta_C) = 0.65$

Step 2: find α for base case

“Base case”: $x_F = 0, x_C = 0$, representing men with no college degree

- $\sigma(\alpha + 0\beta_F + 0\beta_C) = 0.40$
- $\sigma(\alpha) = 0.40$

Step 2: find α for base case

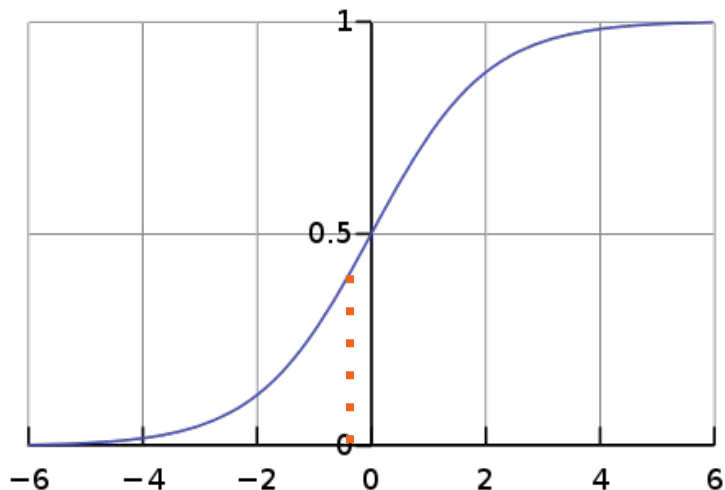
“Base case”: $x_F = 0$, $x_C = 0$, representing men with no college degree

- $\sigma(\alpha + 0\beta_F + 0\beta_C) = 0.40$
- $\sigma(\alpha) = 0.40$

Interpreting the intercept: the probability that x 's = 0 (i.e., men w/ no college degree) yields probability(Democrat = 1) is 0.40.

Step 2: find α for base case

Logistic Function $\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$

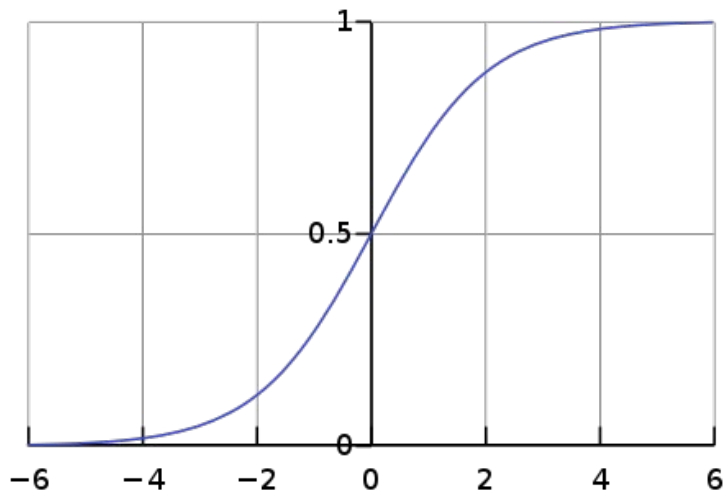


“Base case”: $x_F = 0$, $x_C = 0$, representing men with no college degree

- $\sigma(\alpha + 0\beta_F + 0\beta_C) = 0.40$
- $\sigma(\alpha) = 0.40$
- $e^\alpha / (e^\alpha + 1) = 0.4$
- $\alpha = \log(0.4 / (1 - 0.4)) = -0.405$

Step 2: find α for base case

Logistic Function $\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$



“Base case”: $x_F = 0$, $x_C = 0$, representing men with no college degree

- $\sigma(\alpha + 0\beta_F + 0\beta_C) = 0.40$
- $\sigma(\alpha) = 0.40$
- $e^\alpha / (e^\alpha + 1) = 0.4$
- $\alpha = \log(0.4 / (1 - 0.4)) = -0.405$

Now we know α

Step 3: find β 's for input variables

P(Democrat)	Men	Women
No college degree	$\sigma(-.405 + 0\beta_F + 0\beta_C) = 0.40$	$\sigma(\alpha + 1\beta_F + 0\beta_C) = 0.51$
College degree	$\sigma(\alpha + 0\beta_F + 1\beta_C) = 0.48$	$\sigma(\alpha + 1\beta_F + 1\beta_C) = 0.65$

Step 3: find β 's for input variables

P(Democrat)	Men	Women
No college degree	$\sigma(-.405 + 0\beta_F + 0\beta_C) = 0.40$	$\sigma(-.405 + 1\beta_F + 0\beta_C) = 0.51$
College degree	$\sigma(\alpha + 0\beta_F + 1\beta_C) = 0.48$	$\sigma(\alpha + 1\beta_F + 1\beta_C) = 0.65$

We can solve for β_F the same way!

Logistic Function $\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$

$$\sigma(\alpha + \beta_F) = 0.51$$

$$\exp(\alpha + \beta_F) / (1 + \exp(\alpha + \beta_F)) = 0.51$$

$$\alpha + \beta_F = \log(0.51 / (1 - 0.51)) = 0.040$$

From before, $\alpha = -0.405$

$$\beta_F = 0.040 - (-0.405) = \mathbf{0.445}$$

Women

$$\sigma(-.405 + 1\beta_F + 0\beta_C) = 0.51$$

Step 3: find β 's for input variables

P(Democrat)	Men	Women
No college degree	$\sigma(-.405 + 0\beta_F + 0\beta_C) = 0.40$	$\sigma(-.405 + 0.445 + 0\beta_C) = 0.51$
College degree	$\sigma(\alpha + 0\beta_F + 1\beta_C) = 0.48$	$\sigma(\alpha + 1\beta_F + 1\beta_C) = 0.65$

We can solve for β_C the same way!

P(Democrat)	Men	Women
No college degree	$\sigma(-.405 + 0\beta_F + 0\beta_C) = 0.40$	$\sigma(-.405 + 0.445 + 0\beta_C) = 0.51$
College degree	$\sigma(-.405 + 0\beta_F + 1\beta_C) = 0.48$	$\sigma(\alpha + 1\beta_F + 1\beta_C) = 0.65$

We can solve for β_C the same way!

Logistic Function $\sigma(t) = \frac{e^t}{e^t + 1} = \frac{1}{1 + e^{-t}}$

P(Democrat)	Men	$\sigma(\alpha + \beta_C) = 0.48$
No college degree		$\exp(\alpha + \beta_C) / (1 + \exp(\alpha + \beta_C)) = 0.48$ $\alpha + \beta_C = \log(0.48 / (1 - 0.48)) = -0.080$
College degree	$\sigma(-.405 + 0\beta_F + 1\beta_C) = 0.48$	From before, $\alpha = -0.405$ $\beta_C = -0.080 - (-0.405) = 0.325$

Plug in values for parameters

$$\alpha = -0.405, \beta_F = 0.445, \beta_C = 0.325$$

P(Democrat)	Men	Women
No college degree	$\sigma(\alpha + 0\beta_F + 0\beta_C) = 0.40$	$\sigma(\alpha + 1\beta_F + 0\beta_C) = 0.51$
College degree	$\sigma(\alpha + 0\beta_F + 1\beta_C) = 0.48$	$\sigma(\alpha + 1\beta_F + 1\beta_C)$

Plug in values for parameters

$$\alpha = -0.405, \beta_F = 0.445, \beta_C = 0.325$$

P(Democrat)	Men	Women
No college degree	$\sigma(\alpha + 0\beta_F + 0\beta_C) = 0.40$	$\sigma(\alpha + 1\beta_F + 0\beta_C) = 0.51$
College degree	$\sigma(\alpha + 0\beta_F + 1\beta_C) = 0.48$	$\begin{aligned}\sigma(\alpha + 1\beta_F + 1\beta_C) \\ &= \sigma(-0.405 + 0.445 + 0.325) \\ &= e^{0.365} / (1 + e^{0.365}) \\ &= 0.59\end{aligned}$

Our data has a different number!

P(Democrat)	Men	Women
No college degree	$\sigma(\alpha + 0\beta_F + 0\beta_C) = 0.40$	$\sigma(\alpha + 1\beta_F + 0\beta_C) = 0.51$
College degree	$\sigma(\alpha + 0\beta_F + 1\beta_C) = 0.48$	$\sigma(\alpha + 1\beta_F + 1\beta_C) = 0.65$

Plug in values for parameters

$$\alpha = -0.405, \beta_F = 0.445, \beta_C = 0.325$$

P(Democrat)	Men	Women
No college degree	$\sigma(\alpha + 0\beta_F + 0\beta_C) = 0.40$	$\sigma(\alpha + 1\beta_F + 0\beta_C) = 0.51$
College degree	$\sigma(\alpha + 0\beta_F + 1\beta_C) = 0.48$	$\begin{aligned}\sigma(\alpha + 1\beta_F + 1\beta_C) \\ &= \sigma(-0.405 + 0.445 + 0.325) \\ &= e^{0.365} / (1 + e^{0.365}) \\ &= 0.59 \neq 0.65\end{aligned}$

Plug in values for parameters

$$\alpha = -0.405, \beta_F = 0.445, \beta_C = 0.325$$

P(Democrat)	Men	Women
No college degree	$\sigma(\alpha + 0\beta_F + 0\beta_C) = 0.40$	$\sigma(\alpha + 1\beta_F + 0\beta_C) = 0.51$
College degree	$\sigma(\alpha + 0\beta_F + 1\beta_C) = 0.48$	$\sigma(\alpha + 1\beta_F + 1\beta_C) = 0.59 \neq 0.65$

Independence assumption is wrong! We can't infer $P(\text{Dem} \mid \text{College, Female})$ from $P(\text{Dem} \mid \text{College})$ and $P(\text{Dem} \mid \text{Female})$

Plug in values for parameters

$$\alpha = -0.405, \beta_F = 0.445, \beta_C = 0.325, \beta_{CF} = 0.255$$

P(Democrat)	Men	Women
No college degree	$\sigma(\alpha + 0\beta_F + 0\beta_C) = 0.40$	$\sigma(\alpha + 1\beta_F + 0\beta_C) = 0.51$
College degree	$\sigma(\alpha + 0\beta_F + 1\beta_C) = 0.48$	$\sigma(\alpha + 1\beta_F + 1\beta_C + 1\beta_{CF}) = 0.65$

Independence assumption is wrong! We can't infer $P(\text{Dem} \mid \text{College, Female})$ from $P(\text{Dem} \mid \text{College})$ and $P(\text{Dem} \mid \text{Female})$

P(Democrat)	Men	Women
No college degree	0.40	0.51
College degree	0.48	0.65

Koenecke: Female=1, College=1
 Thalken: Female=1, College=1
 Sparky: Female=0, College=0

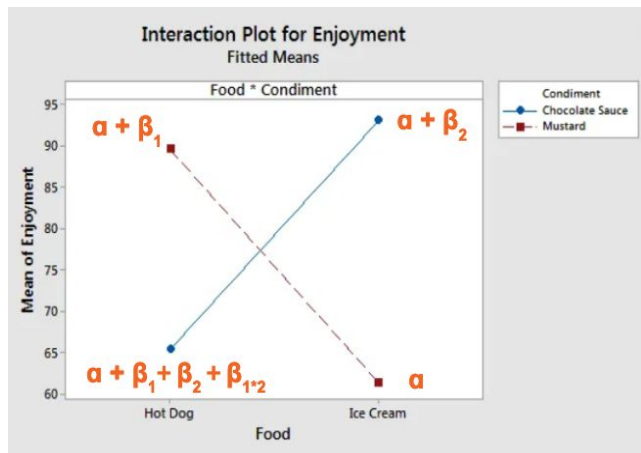
We do need an interaction variable

$$P(\text{Democrat}) = \sigma(\alpha + \beta_F x_F + \beta_C x_C)$$

$$P(\text{Democrat}) = \sigma(\alpha + \beta_F x_F + \beta_C x_C + \beta_{FC} x_F * x_C)$$


Do you need regression interactions?

- Use interaction plots
(harder to interpret with logit)



- Use probabilistic thinking

P(Democrat)	Men	Women
No college degree	$\sigma(\alpha + 0\beta_F + 0\beta_C) = 0.40$	$\sigma(\alpha + 1\beta_F + 0\beta_C) = 0.51$
College degree	$\sigma(\alpha + 0\beta_F + 1\beta_C) = 0.48$	$\sigma(\alpha + 1\beta_F + 1\beta_C) = 0.59^* \text{ 0.65}$

One minute break & attendance!



tinyurl.com/32fh9s9p

Classification

- We've learned about two methods (Naive Bayes and logistic regression) to classify outcomes into discrete "classes"
 - Binary outcome 0/1
 - Weather type (sleet, rain, other)

Classification

- We've learned about two methods (Naive Bayes and logistic regression) to classify outcomes into discrete "classes"
 - Binary outcome 0/1
 - Weather type (sleet, rain, other)
- What if we want to classify things, but we don't know the actual "classes"?

Clustering!

- Clustering allows you to classify your data rows, *even if you don't have an output variable*

Clustering!

- Clustering allows you to classify your data rows, *even if you don't have an output variable*

This is called an “unsupervised learning” algorithm

Clustering!

- Clustering allows you to classify your data rows, *even if you don't have an output variable*

This is called an “unsupervised learning” algorithm

(as opposed to “supervised learning”, e.g. regressions where we have examples of the desired output)

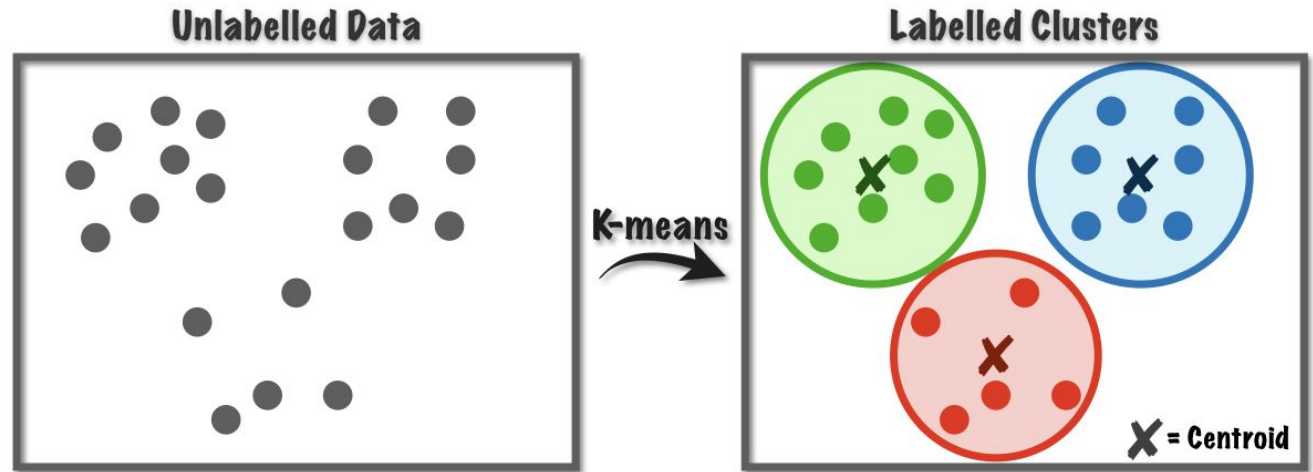
Clustering!

- Clustering allows you to classify your data rows, *even if you don't have an output variable*
 - **Algorithm Input:** the number of classes k
 - **Algorithm Output:** k clusters of datapoints based on spatial distance from each other

Example: Netflix categories

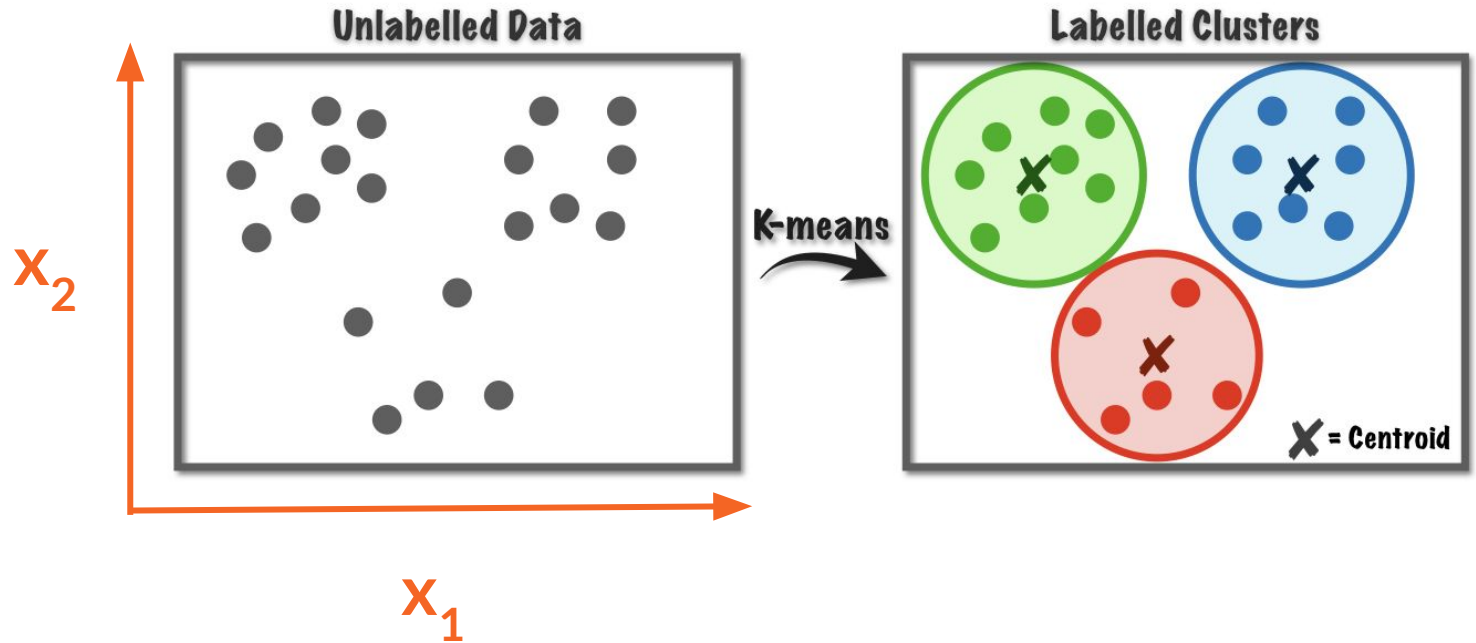
Campy Psychological Movies	3899
Campy Sci-Fi & Fantasy	1194
Campy Sci-Fi Horror Movies	4140
Campy Slasher and Serial Killer Movies	1646
Campy Thrillers	3226
Campy Zombie Movies	1515
Car Culture Shows	753
Cardio & Aerobics Workouts	711
Career & Finance	2560
Cerebral Action & Adventure	4778
Cerebral Biographical Dramas	127
Cerebral Biographical Movies	4518

Clustering!



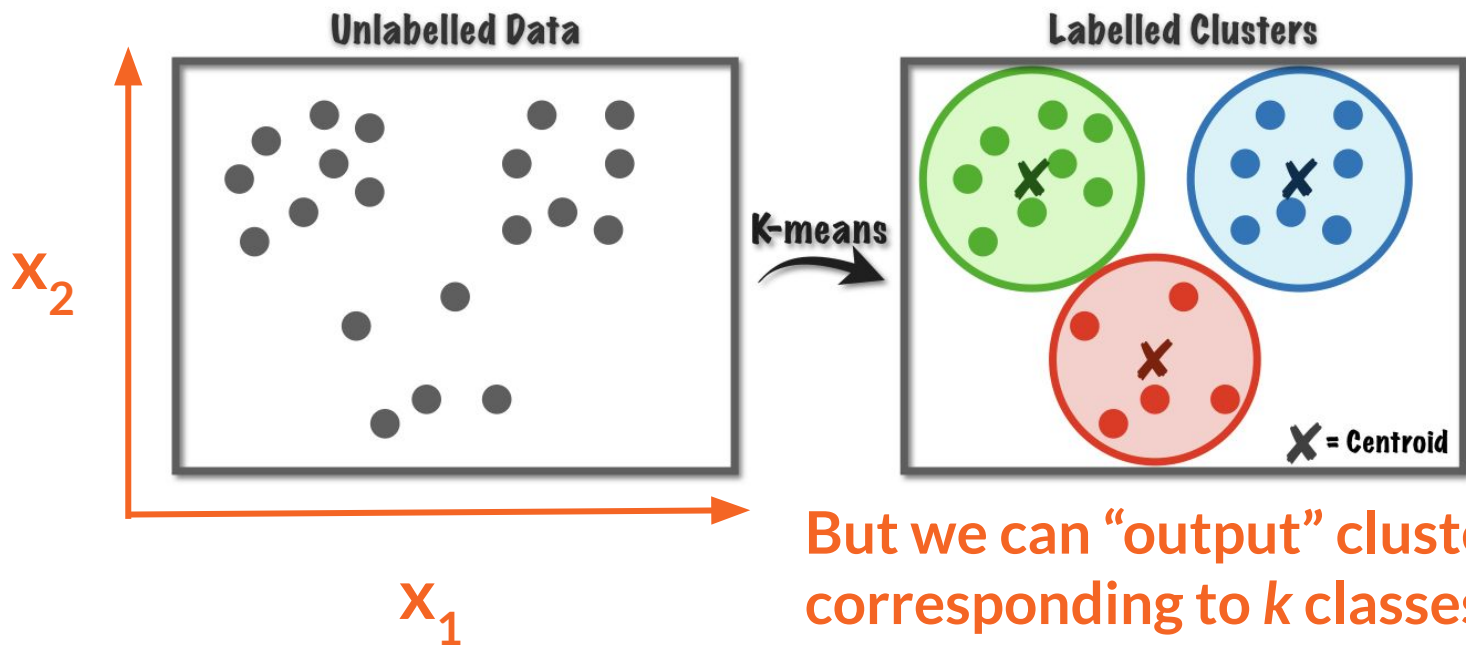
Clustering!

Example: 2-dimensional data



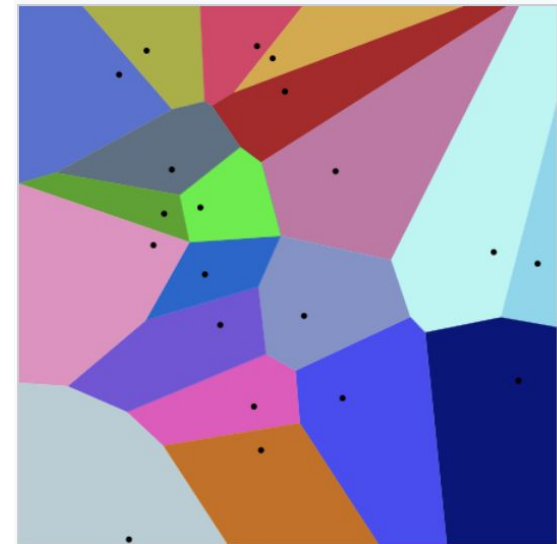
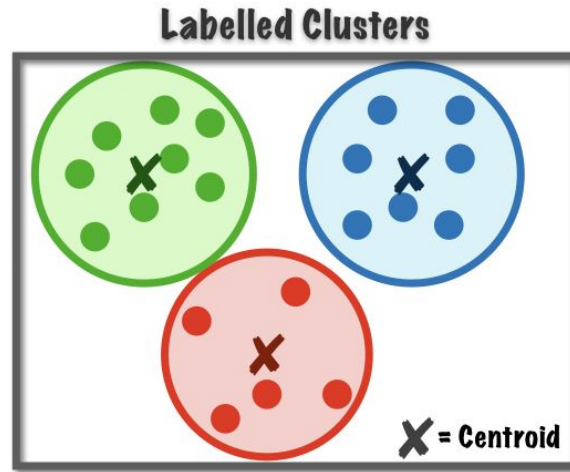
Clustering!

We don't have y 's



But we can “output” clusters corresponding to k classes

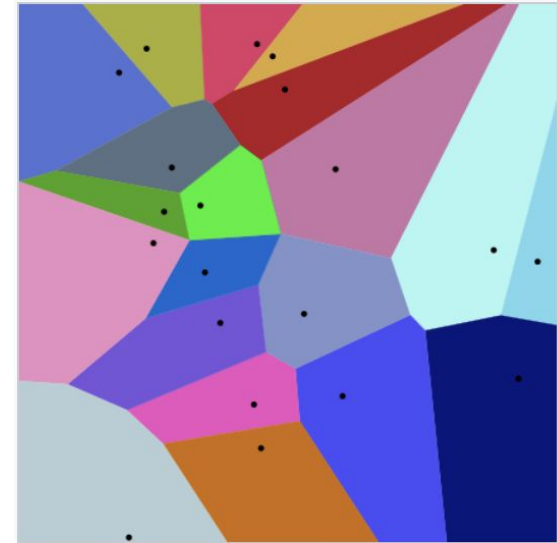
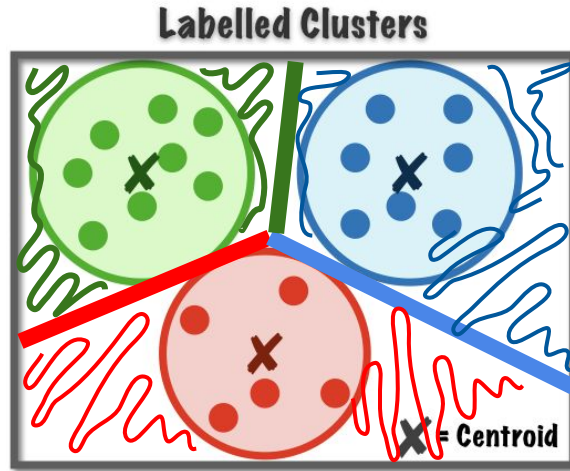
You can think of clusters as Voronoi diagrams



20 points and their Voronoi cells
(larger version [below](#))

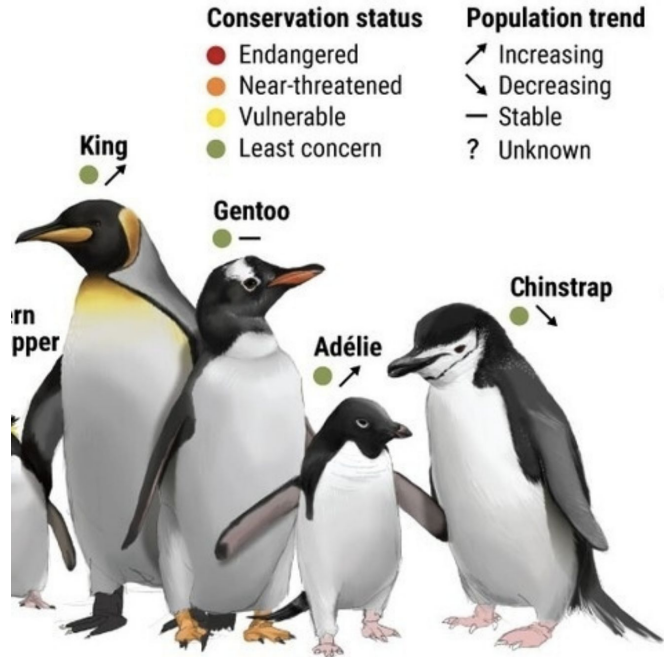
You can think of clusters as Voronoi diagrams

You're actually dividing up the space in n dimensions, not just drawing circles around some data



20 points and their Voronoi cells
(larger version [below](#))

Dataset: Penguins



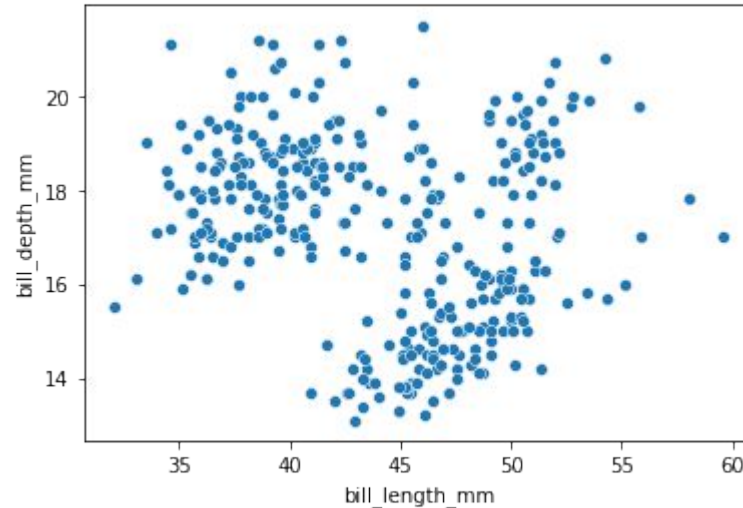
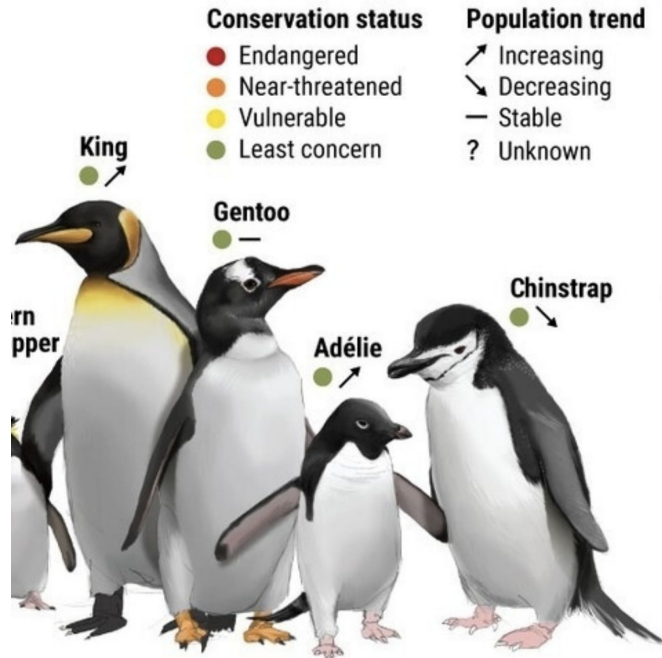
```
penguins_df =  
seaborn.load_dataset("penguins")
```

Dataset: Penguins

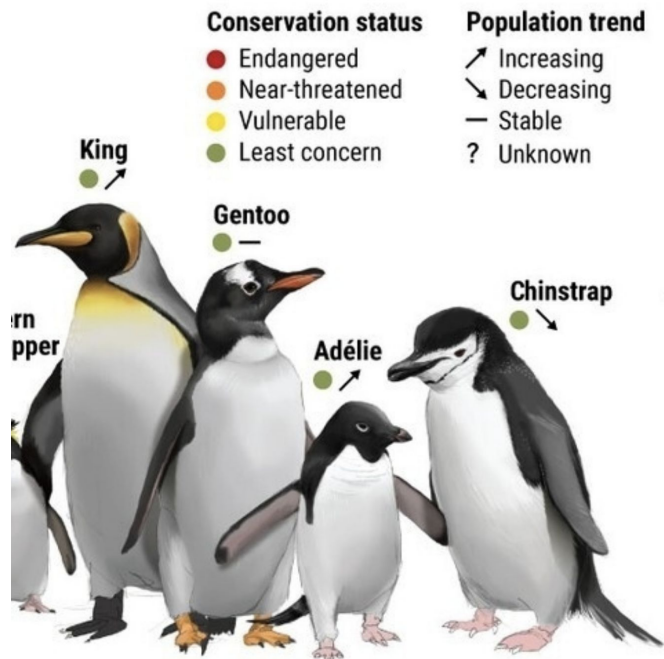
```
penguins_df = seaborn.load_dataset("penguins")
penguins_df = penguins_df.dropna()
penguins_df.head()
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	Male
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	Female
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	Female
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	Female
5	Adelie	Torgersen	39.3	20.6	190.0	3650.0	Male

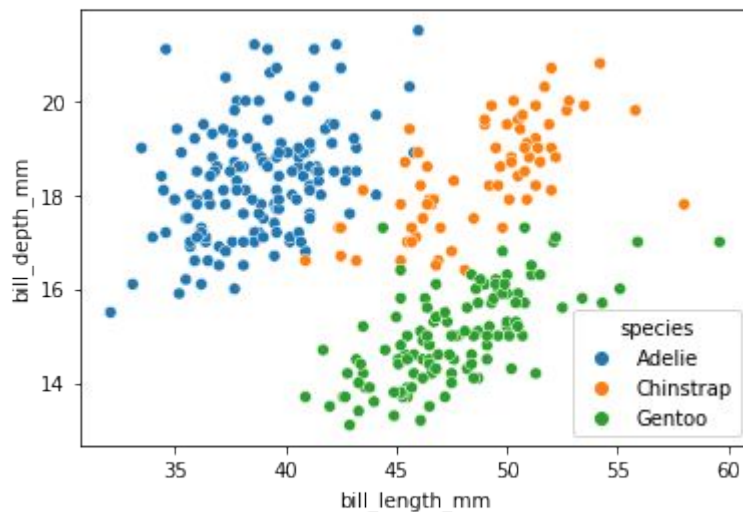
Measurements of penguins



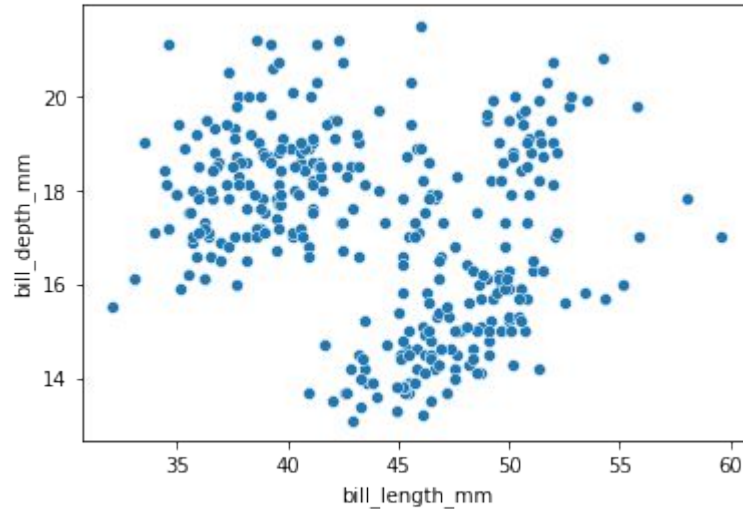
Penguins by species



```
seaborn.scatterplot(data=penguins_df,  
x="bill_length_mm", y="bill_depth_mm",  
hue="species")
```

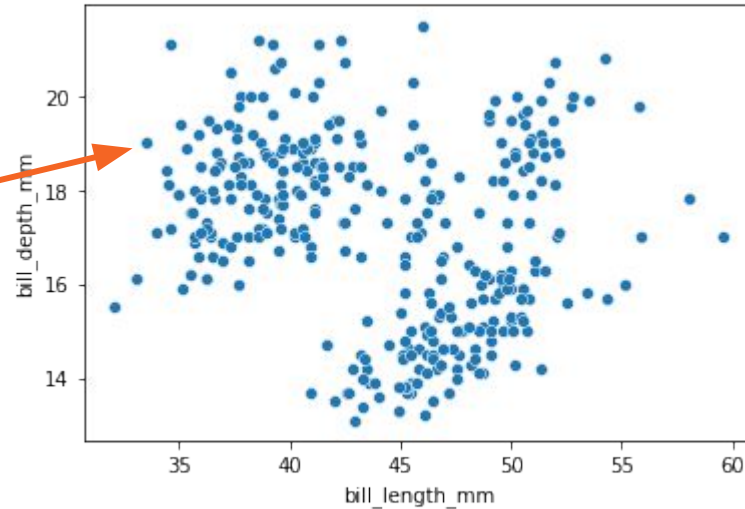


What if we didn't observe species?

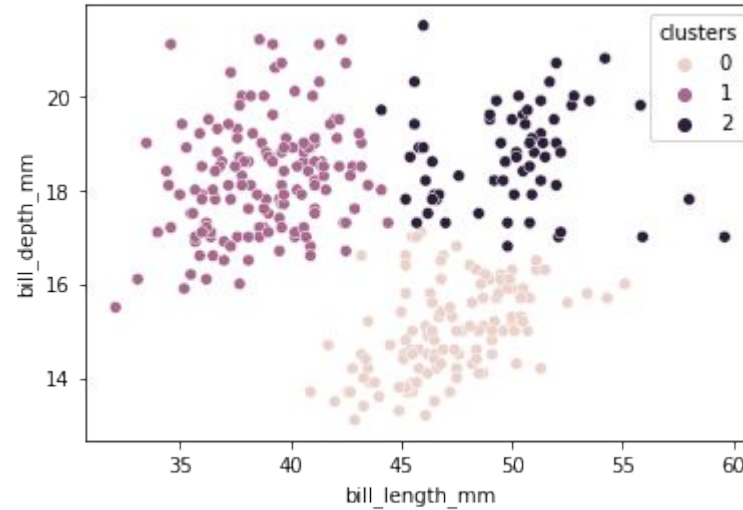


What if we didn't observe species?

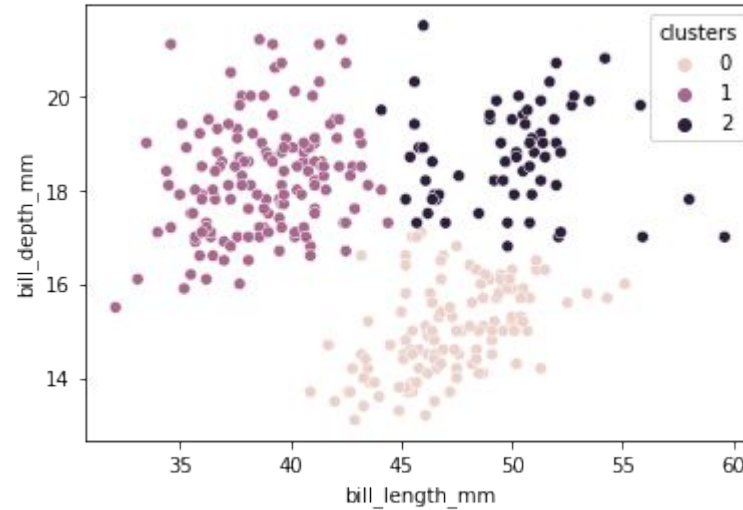
Each data point
is one penguin



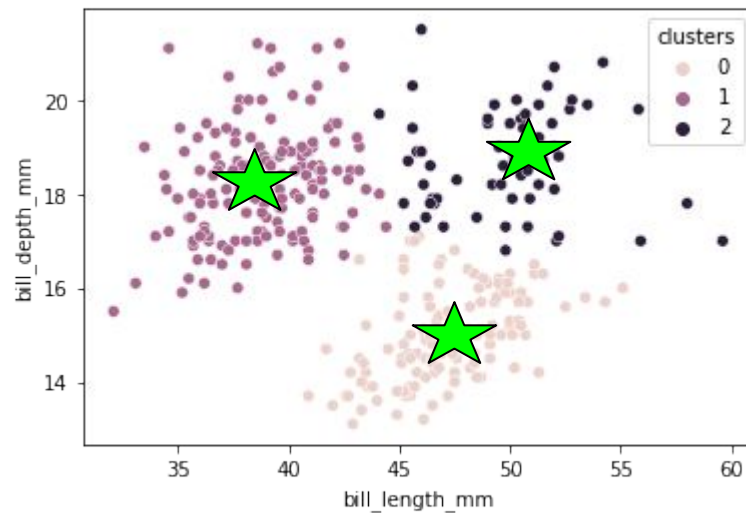
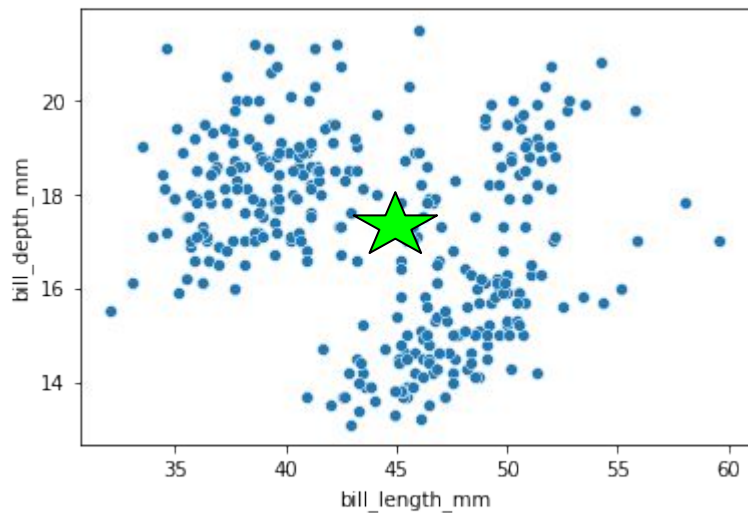
Output: k-means clustering



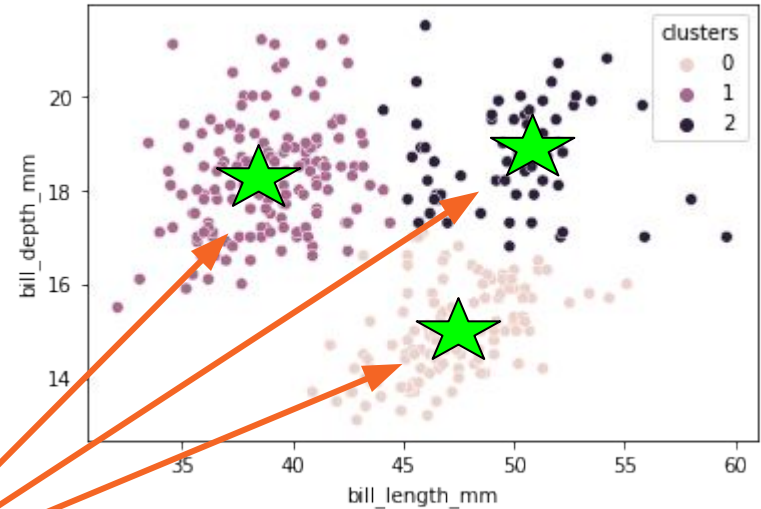
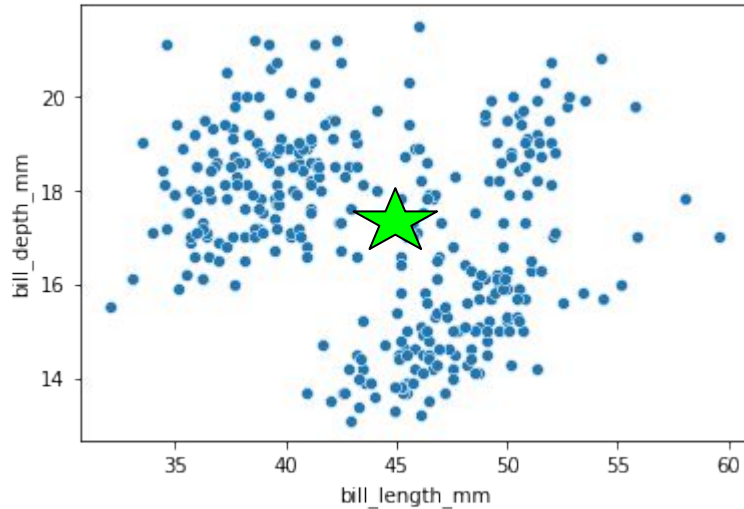
Output: k-means clustering



Global mean vs. category means

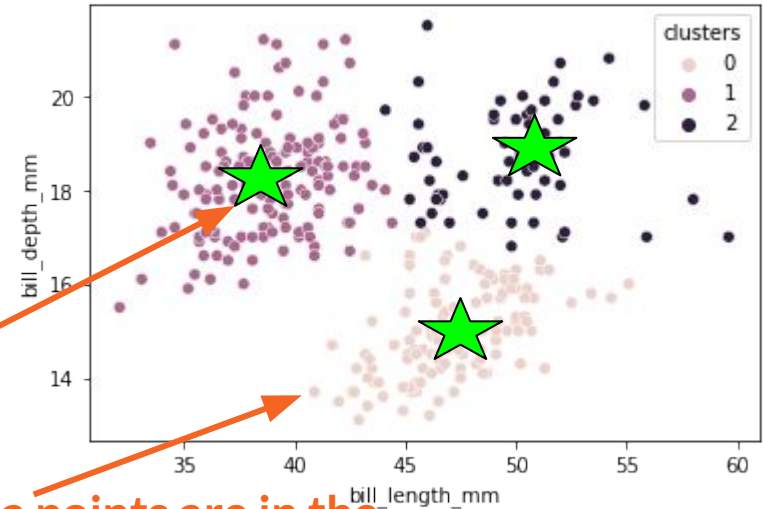
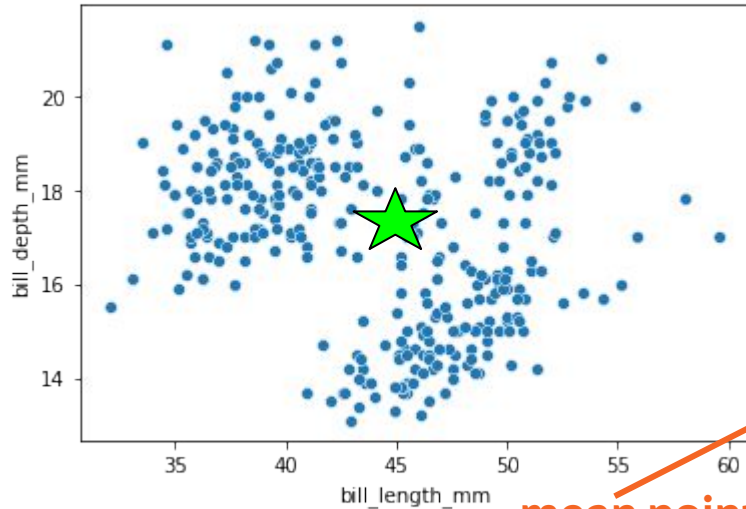


Global mean vs. category means



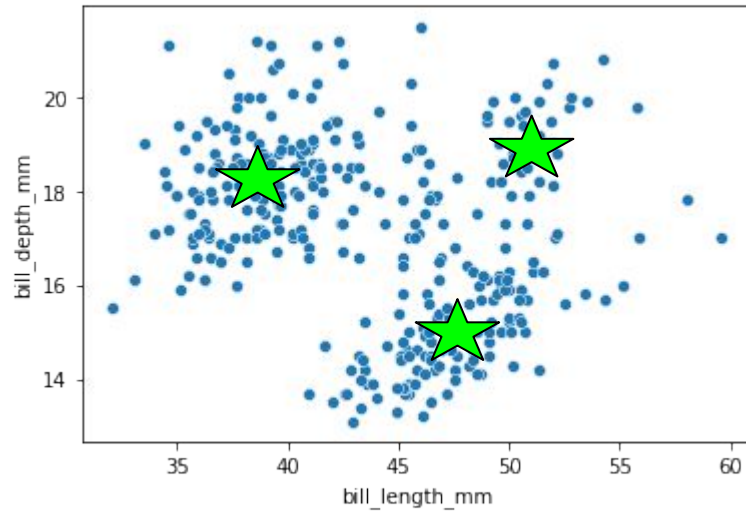
K=3 mean points, one for each labeled cluster!

Global mean vs. category means

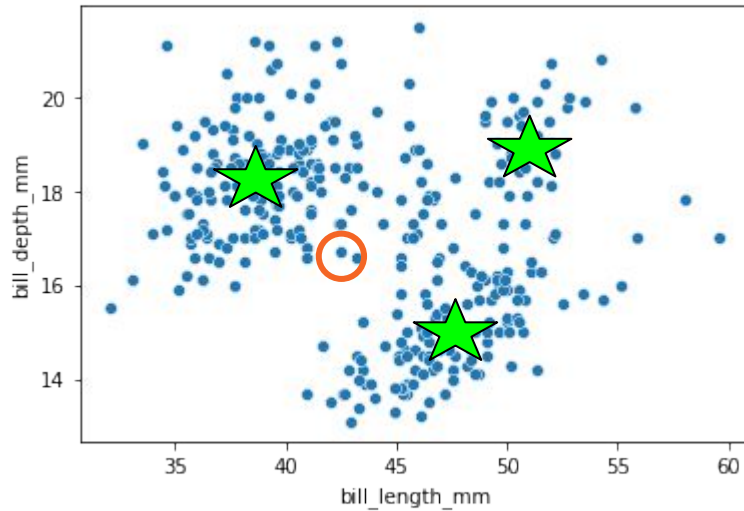


mean points and data points are in the
same coordinate system (length, depth)

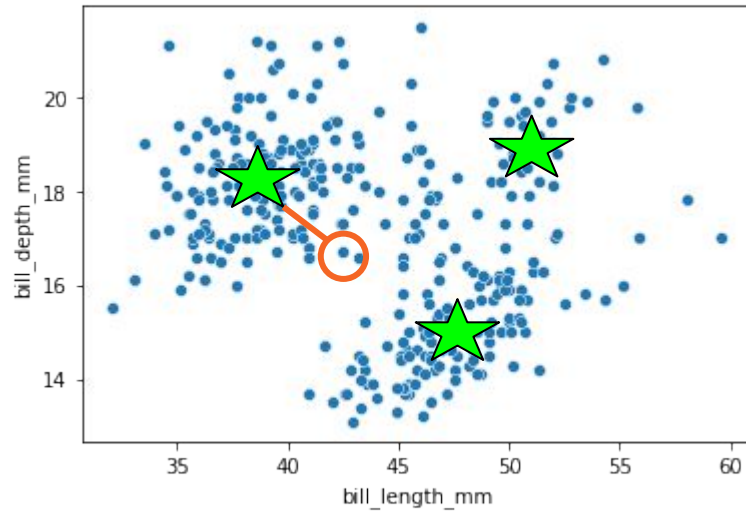
Label points given means



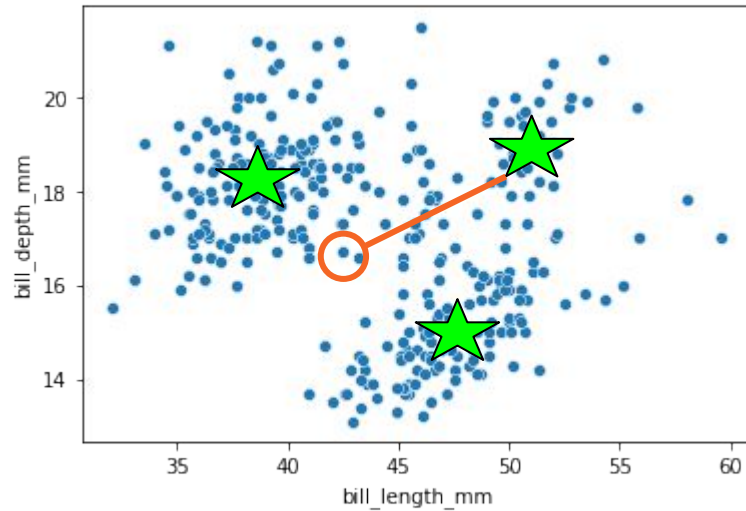
Label points given means



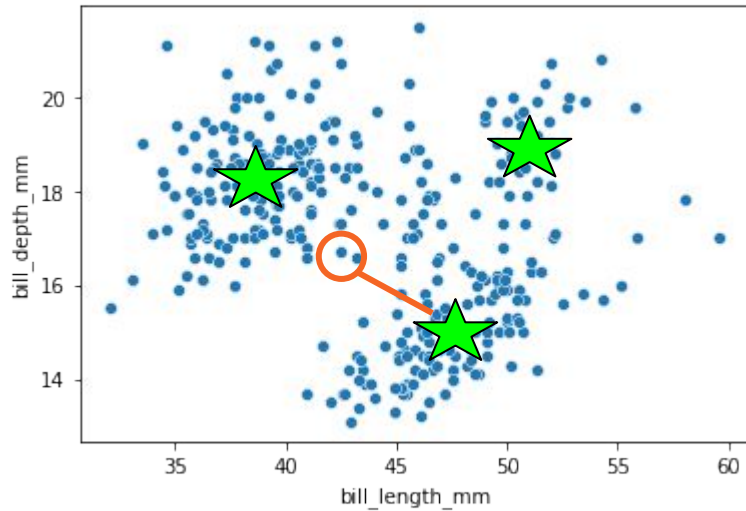
Label points given means



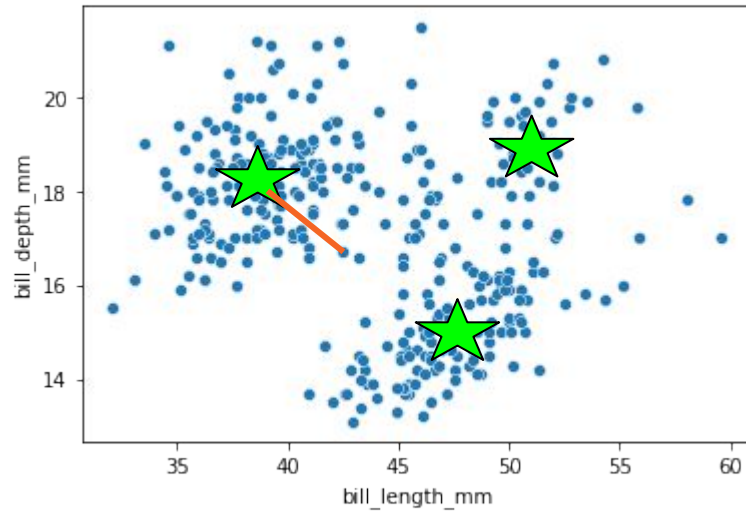
Label points given means



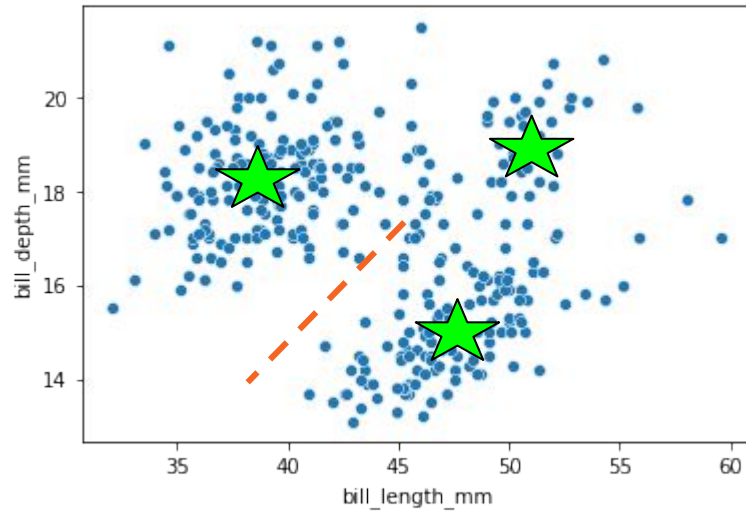
Label points given means



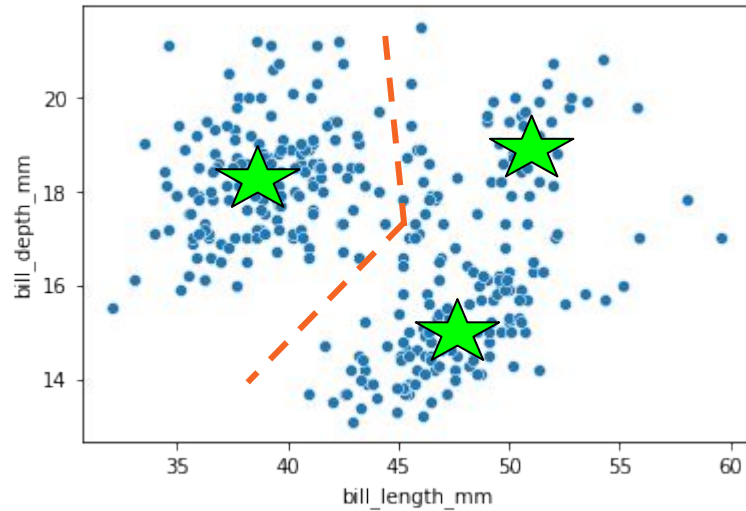
Label points given means



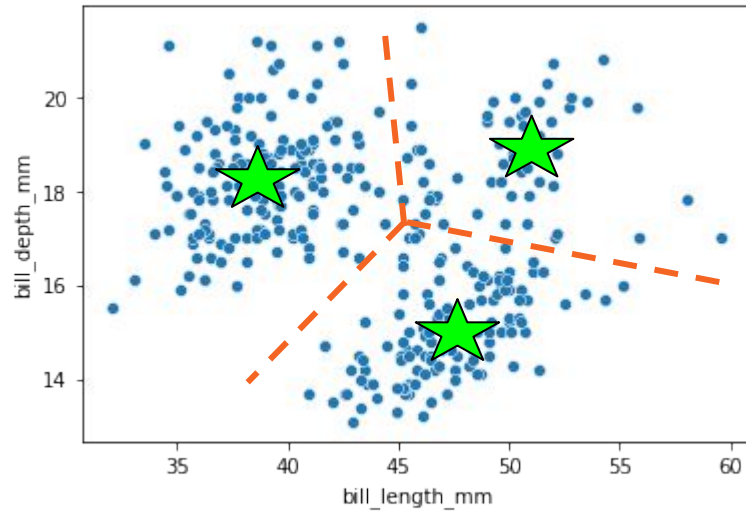
Label points given means



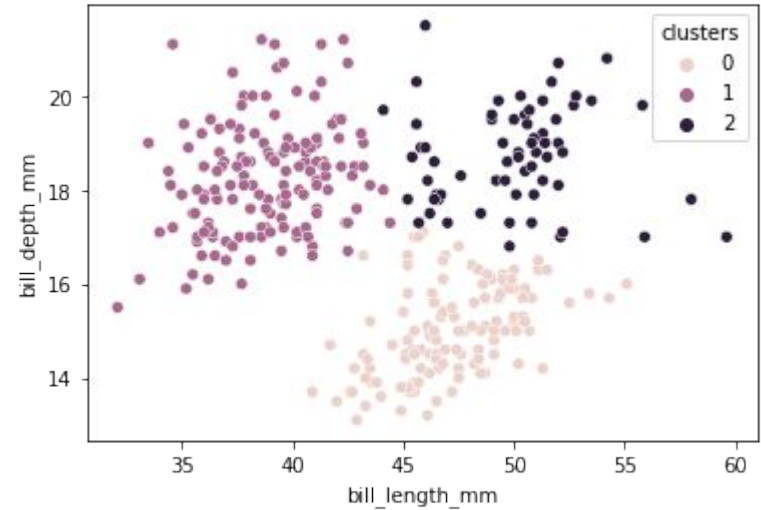
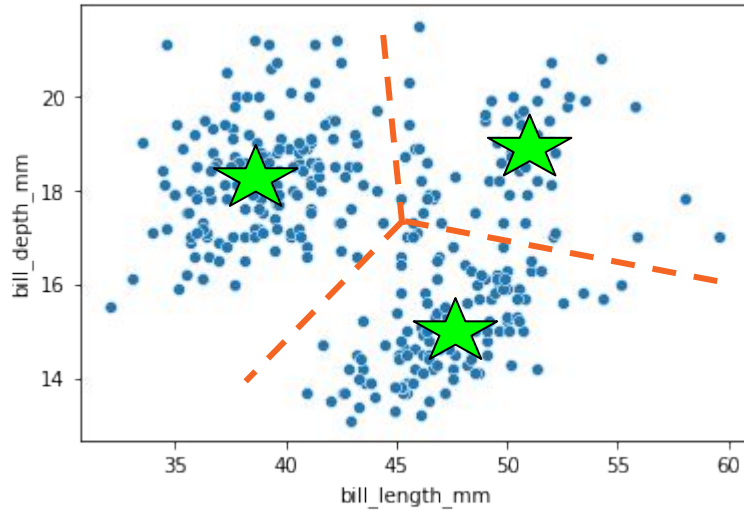
Label points given means



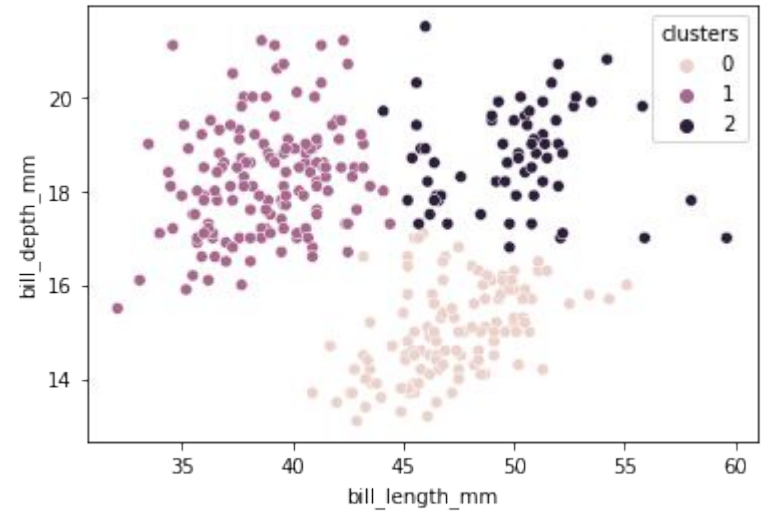
Label points given means



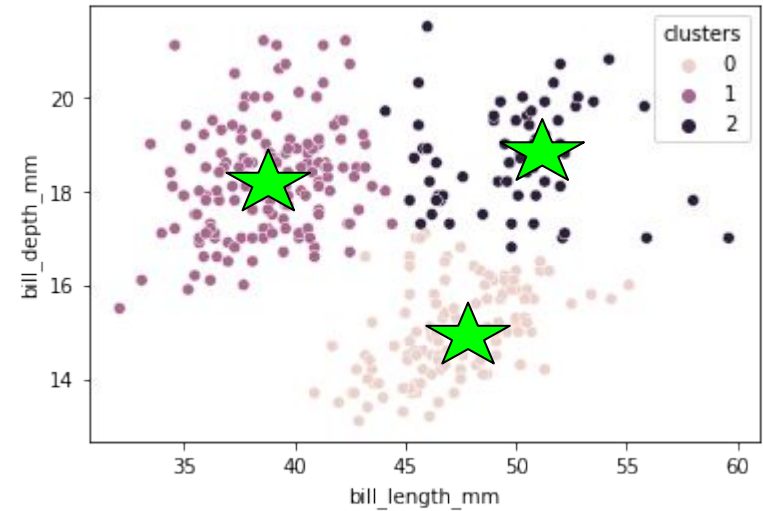
Label points given means



Find the mean given labels



Find the mean given labels



Iterative algorithm

If we knew the position of the K mean points, we could easily assign each data point to one of the K clusters

If we knew the assignment of data points to clusters, we could easily find the position of the K mean points

Iterative algorithm

If we knew the position of the K mean points, we could easily assign each data point to one of the K clusters

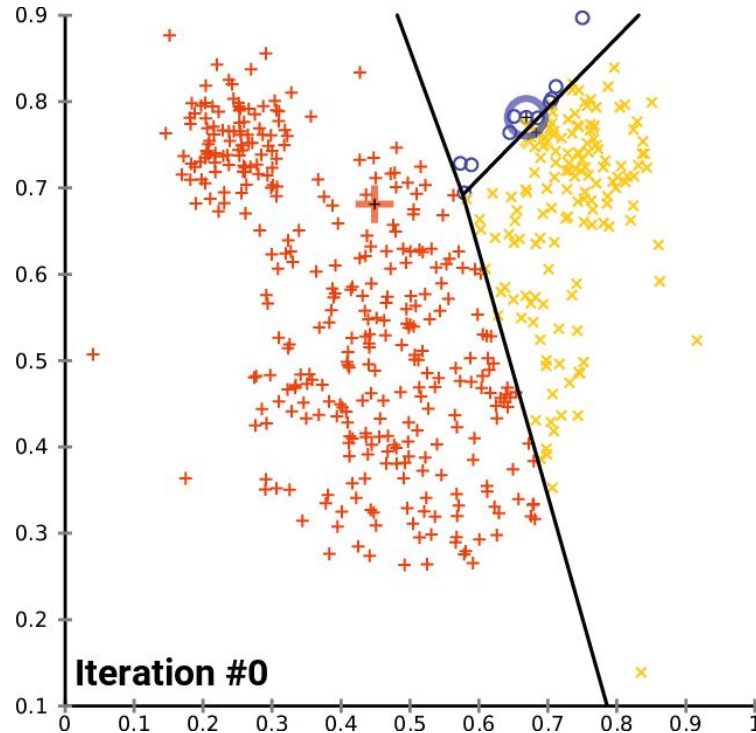
If we knew the assignment of data points to clusters, we could easily find the position of the K mean points

But at the start we don't know either one!

Iterative algorithm

1. Pick K random points
2. While not converged:
 - a. Assign data points to nearest mean
 - b. Set means to average of assigned data points

Iterative algorithm



Ways of measuring distance

Euclidean / ℓ_2 / "as the crow flies"

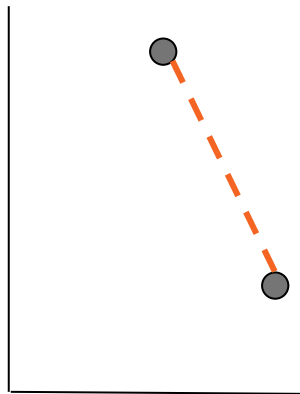
Use Pythagorean theorem: square root of sum of squares

Absolute / ℓ_1 / Manhattan / "city block"

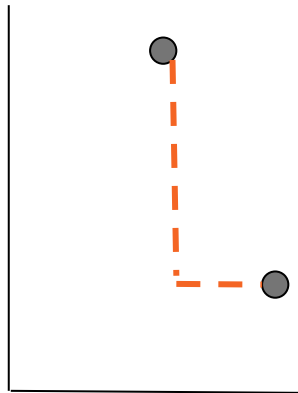
Sum of absolute values for each variable

Cosine / inner product

Ignore magnitude, compare angle between vectors



Ways of measuring distance



Euclidean / ℓ_2 / "as the crow flies"

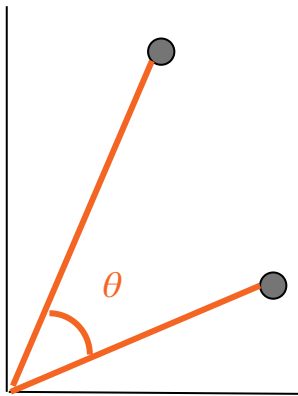
Use Pythagorean theorem: square root of sum of squares

Absolute / ℓ_1 / Manhattan / "city block"

Sum of absolute values for each variable

Cosine / inner product

Ignore magnitude, compare angle between vectors



Ways of measuring distance

Euclidean / ℓ_2 / "as the crow flies"

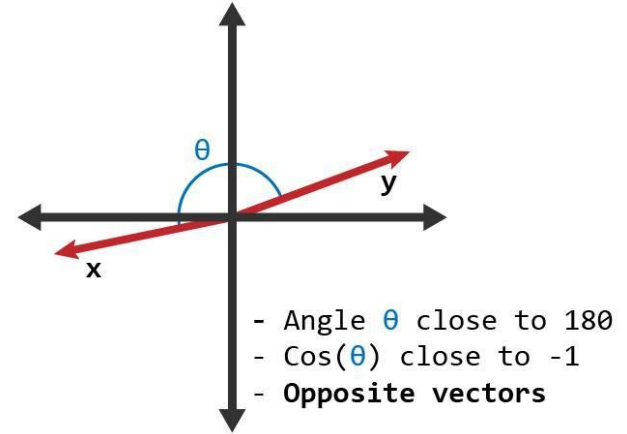
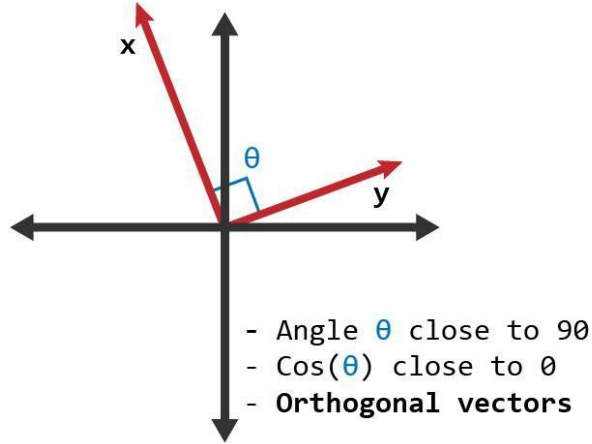
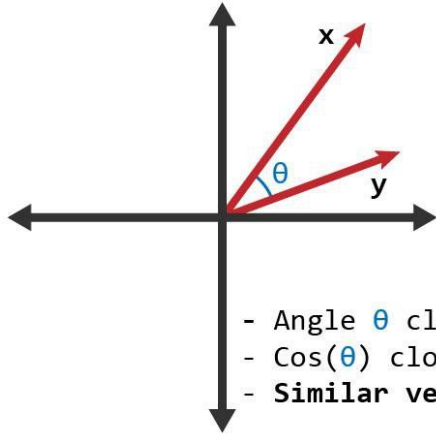
Use Pythagorean theorem: square root of sum of squares

Absolute / ℓ_1 / Manhattan / "city block"

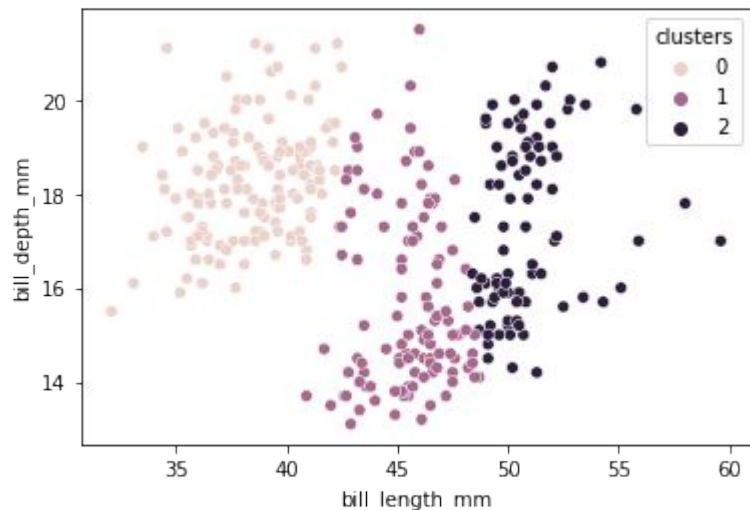
Sum of absolute values for each variable

Cosine / inner product

Ignore magnitude, compare angle between vectors



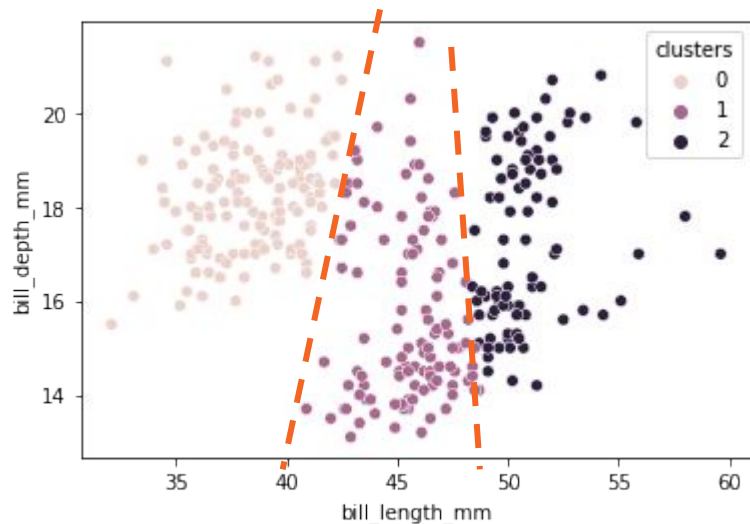
Variables must be comparable



The previous clusters were *displayed* using bill measurements in mm, but they were *clustered* using z-scores

This is what clusters look like with the original measurements

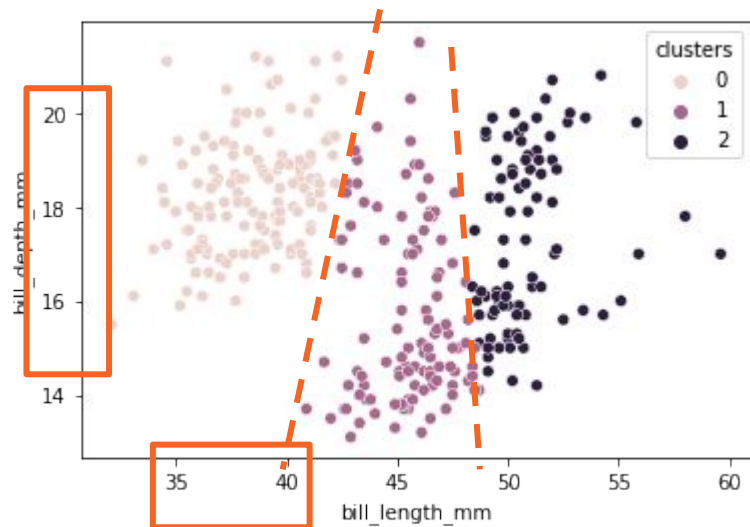
Variables must be comparable



The previous clusters were *displayed* using bill measurements in mm, but they were *clustered* using z-scores

This is what clusters look like with the original measurements

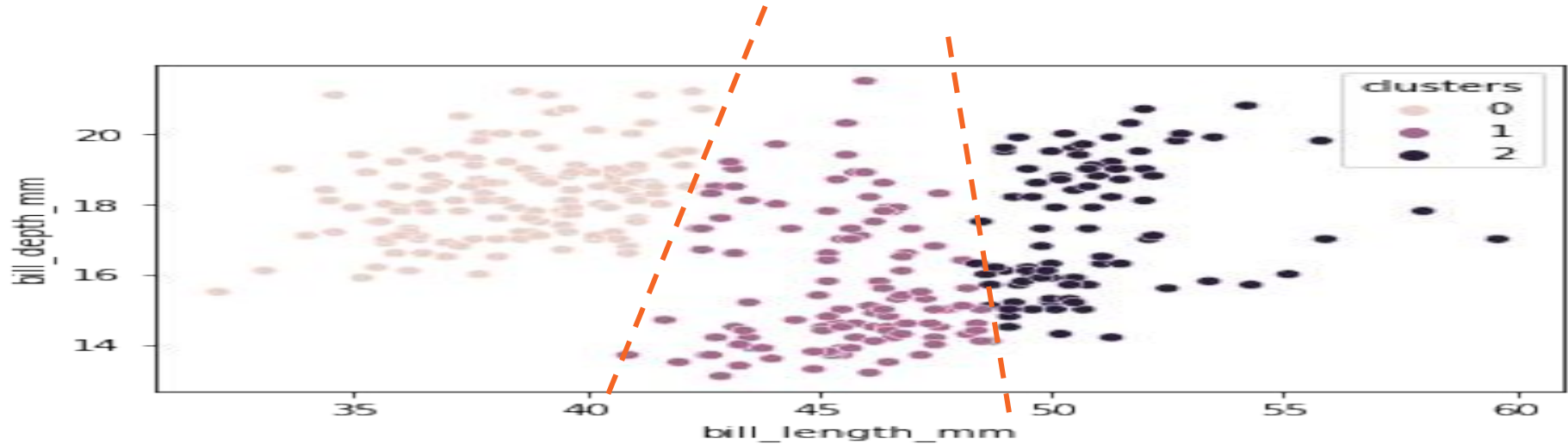
Variables must be comparable



The previous clusters were *displayed* using bill measurements in mm, but they were *clustered* using z-scores

This is what clusters look like with the original measurements

Variables must be comparable

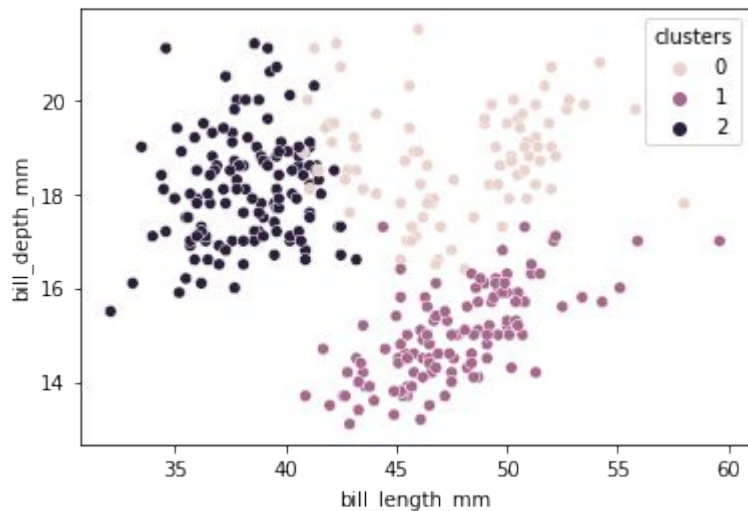


Dataset: Penguins

```
penguins_df = seaborn.load_dataset("penguins")  
penguins_df = penguins_df.dropna()  
penguins_df.head()
```

	species	island	bill_length_mm	bill_depth_mm	flipper_length_mm	body_mass_g	sex
0	Adelie	Torgersen	39.1	18.7	181.0	3750.0	Male
1	Adelie	Torgersen	39.5	17.4	186.0	3800.0	Female
2	Adelie	Torgersen	40.3	18.0	195.0	3250.0	Female
4	Adelie	Torgersen	36.7	19.3	193.0	3450.0	Female
5	Adelie	Torgersen	39.3	20.6	190.0	3650.0	Male

More than 2 dimensions

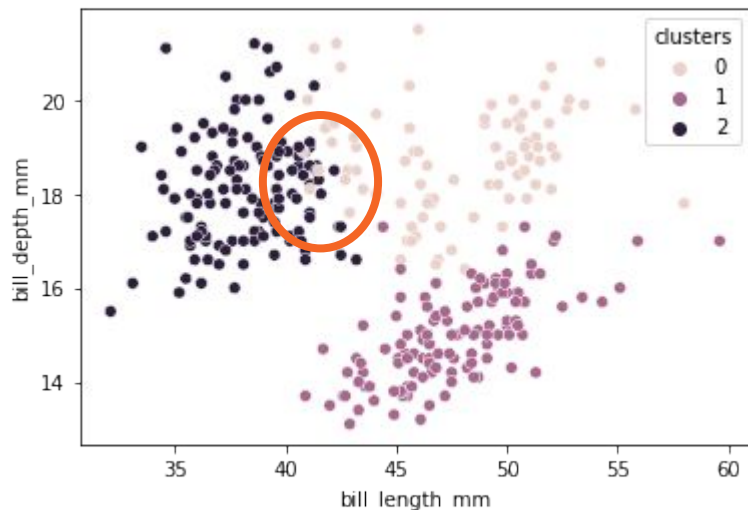


```
penguin_clusters =  
KMeans(n_clusters=3).fit(penguins_df[[  
    "bill_length_z", "bill_depth_z",  
    "flipper_length_z", "body_mass_z" ]])
```

```
penguins_df["clusters"] =  
penguin_clusters.labels_
```

```
seaborn.scatterplot(data=penguins_df,  
x="bill_length_mm", y="bill_depth_mm",  
hue="clusters")
```

More than 2 dimensions

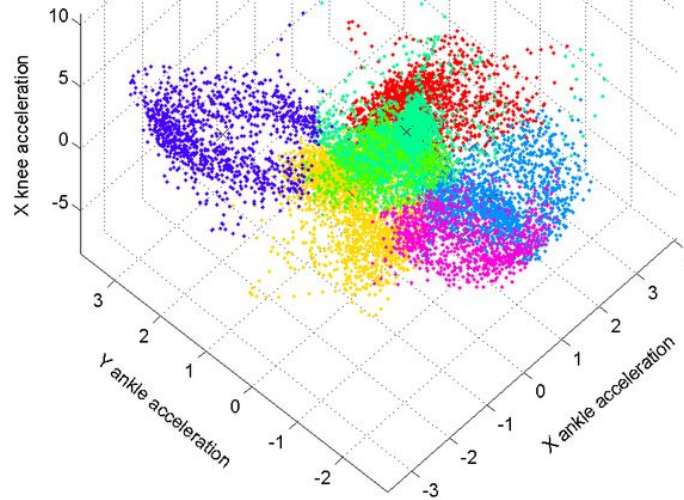


```
penguin_clusters =  
KMeans(n_clusters=3).fit(penguins_df[[  
    "bill_length_z", "bill_depth_z",  
    "flipper_length_z", "body_mass_z" ]])
```

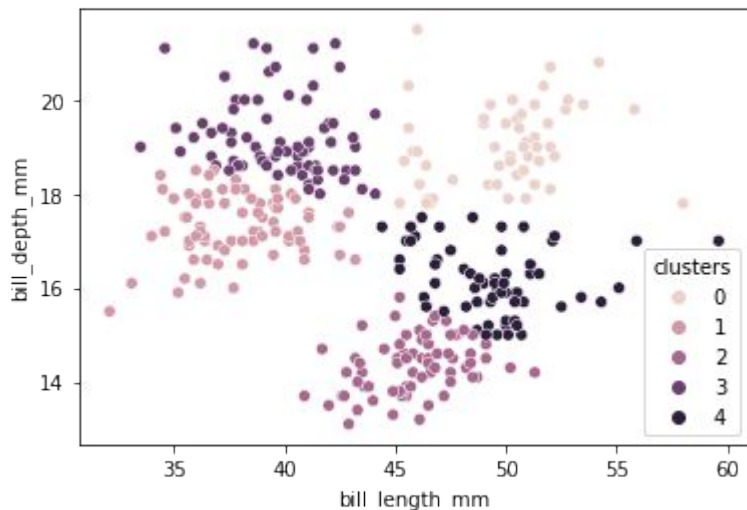
```
penguins_df["clusters"] =  
penguin_clusters.labels_
```

```
seaborn.scatterplot(data=penguins_df,  
x="bill_length_mm", y="bill_depth_mm",  
hue="clusters")
```

3D clustering plot: different human runners' gaits



Choosing the number of clusters



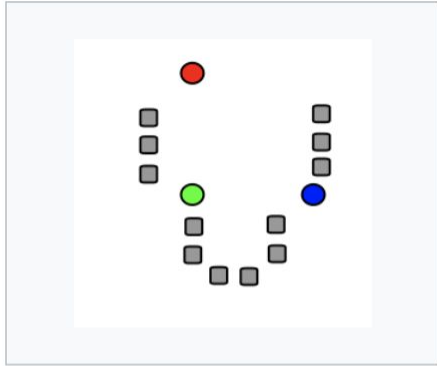
Selecting K is a hard problem

Anyone who says they have a formula for deciding the "right" number is wrong

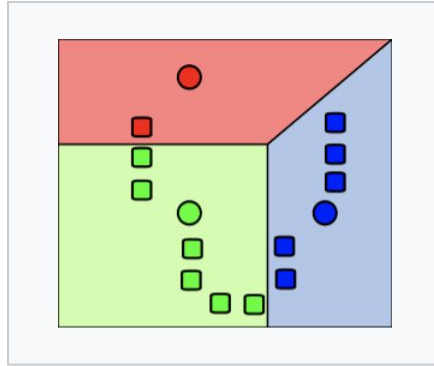
"How do I choose K?" is the single most frequent question Instructor Thalken has gotten in her career and there are no good answers

Try a few K values and see what makes sense and suits your specific needs

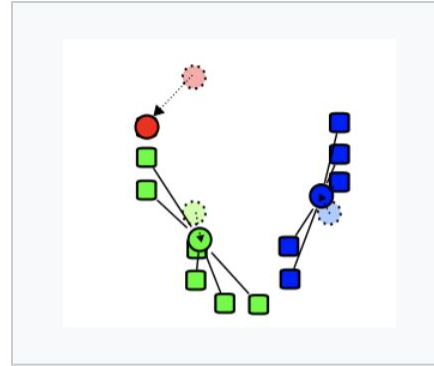
K-Means summary



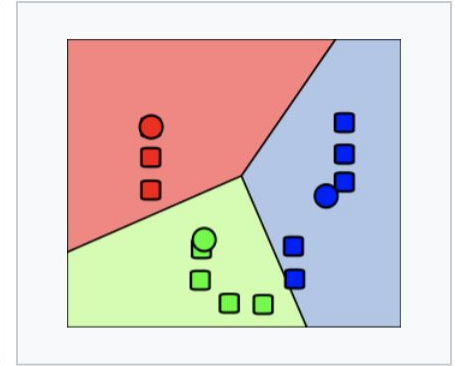
1. k initial "means" (in this case $k=3$) are randomly generated within the data domain (shown in color).



2. k clusters are created by associating every observation with the nearest mean.



3. The **centroid** of each of the k clusters becomes the new mean.



4. Steps 2 and 3 are repeated until convergence has been reached.

Admin

- If you're interested in TAing, apply via <https://cis-student-hiring.coecis.cornell.edu/> by Nov 15th!
- Phase 4 is due Nov 16, 11:59pm

Phase IV/V Rubric Walkthrough

You can find this content on Canvas under:

Modules > Friday Discussions > Discussion11_FA23.zip

Remember that the other Discussion 11 course content will be checked for completion on Friday!