

Information Visualization I

School of Information, University of Michigan

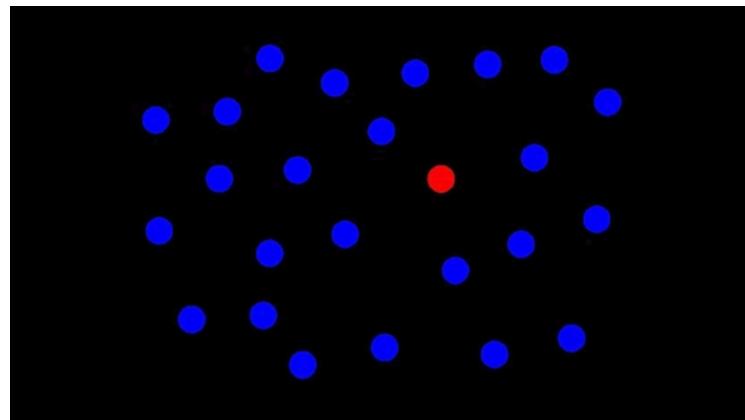
Week 3:

- Perception / Cognition

Assignment Overview

This assignment's objectives include:

- Review, reflect, and apply the concepts of the perception pipeline. Justify how different encodings impact the effectiveness of a visualization depending on the human perception process.



Preattentive Processing

- Recreate visualizations and propose new and alternative visualizations using [Altair](https://altair-viz.github.io/) (<https://altair-viz.github.io/>)

The total score of this assignment will be 100 points consisting of:

- Case study reflection: America's Favorite 'Star Wars' Movies (And Least Favorite Characters) (30 points)
- Altair programming exercise (70 points)

Resources:

- Article by [FiveThirtyEight](https://fivethirtyeight.com) (<https://fivethirtyeight.com>) available [online](https://fivethirtyeight.com/features/americas-favorite-star-wars-movies-and-least-favorite-characters/) (<https://fivethirtyeight.com/features/americas-favorite-star-wars-movies-and-least-favorite-characters/>) (Hickey, 2014)
- Datasets from FiveThirtyEight, we have downloaded a subset of this data in the folder [./assets](#) ([assets](#))
 - The original dataset can be found at [FiveThirtyEight Star Wars Survey](https://github.com/fivethirtyeight/data/tree/master/star-wars-survey) (<https://github.com/fivethirtyeight/data/tree/master/star-wars-survey>).

Important notes:

1) There will be a couple of places where the numbers you get when you select rows may be a little different than 538, but the percents should still work (e.g., 828 instead of 834). You'll see this in our examples. If you can somehow get the data to match exactly, that's great too.

2) Grading for this assignment is entirely done by a human grader. They will be running tests on the functions we ask you to create. This means there is no autograding (submitting through the autograder will result in an error). You are expected to test and validate your own code.

3) Keep your notebooks clean and readable. If your code is highly messy or inefficient you will get a deduction.

4) Follow the instructions for submission on Coursera. You will be providing us a generated link to a read-only version of your notebook and a PDF. When turning in your PDF, please use the File -> Print -> Save as PDF option **from your browser**. Do **not** use the File->Download as->PDF option. Complete instructions for this are under Resources in the Coursera page for this class. If you're having trouble with printing, take a look at [this video](https://youtu.be/PiO-K7AoWjk) (<https://youtu.be/PiO-K7AoWjk>).

```
In [1]: !pip install vega
```

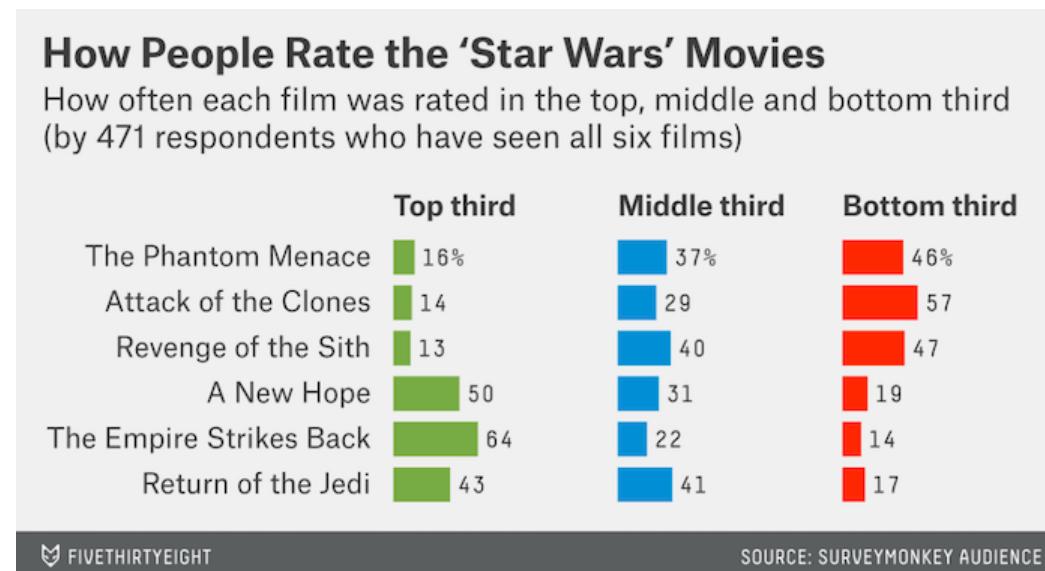
```
Requirement already satisfied: vega in /opt/conda/lib/python3.8/site-packages (2.6.0)
```

Part 1. Perception and Cognition (30 points)

Read the article "[America's Favorite 'Star Wars' Movies \(And Least Favorite Characters\)](https://fivethirtyeight.com/features/americas-favorite-star-wars-movies-and-least-favorite-characters/)." (<https://fivethirtyeight.com/features/americas-favorite-star-wars-movies-and-least-favorite-characters/>) and answer the following questions:

1.1 List the different data types in the following visualizations and their encodings (10 points)

Look at the following visualizations. For each, list the variable, their type, and the encoding used (e.g., Weight, quantitative, color, ...)



Which 'Star Wars' Movies Have You Seen?

Of 835 respondents who have seen any film



 FIVETHIRTYEIGHT

SOURCE: SURVEYMONKEY AUDIENCE

1.1 Answer

How People Rate the 'Star Wars' Movies

- Variable: Film rating frequency --> Data Type: Quantitative --> Color: alt.Color
- Variable: Star Wars movie names --> Data Type: Nominal
- Weight: Top Third --> Color: Green --> alt.Chart(color = 'green') #77AB43
- Weight: Middle Third --> Color: Blue --> alt.Chart(color ='blue') #008FD5
- Weight: Bottom Third --> Color: Red --> alt.Chart(color ='red') #eb1e1e

Which 'Star Wars' Movies Have You Seen?

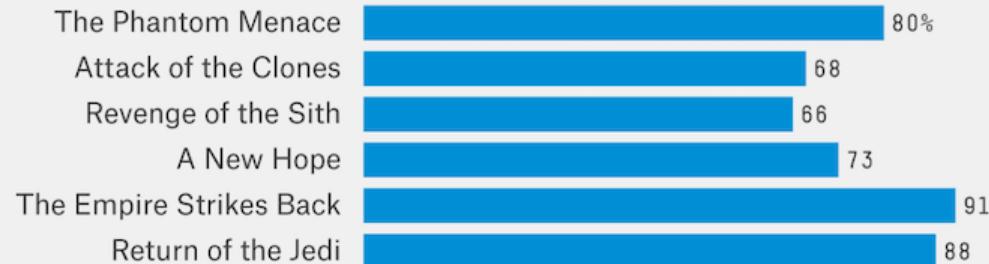
- Variable: # of respondents who saw each movie --> Data Type: Quantitative --> Color: Blue alt.Chart(color ='blue') #008FD5
- Variable: Star Wars movie names --> Data Type: Nominal

1.2 Propose an alternative encoding for the following visualization. Compare the visualizations based on perception. (10 points)

Either hand-draw or use an application to create a sketched solution. Upload an image and describe the differences between your solution and the FiveThirtyEight image in terms of perception (specifically for the task of comparing one movie to another).

Which 'Star Wars' Movies Have You Seen?

Of 835 respondents who have seen any film



 FIVETHIRTYEIGHT

SOURCE: SURVEYMONKEY AUDIENCE

1.2 Answer

which Star Wars Movies
Have You Seen?
(The survey had 835 respondents)

- The Phantom Menace
- Attack of the Clones
- Revenge of the Sith
- A New Hope
- The Empire Strikes Back
- Return of the Jedi



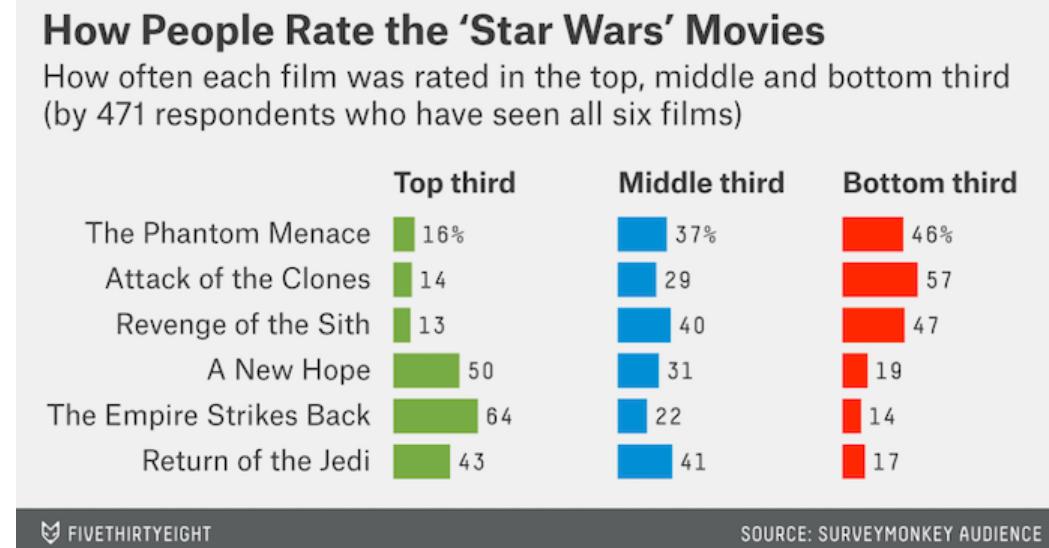
Gestalt's principle of proximity is achieved in both the example and my visualization.

The example is better in terms of perception because it follows the law of continuity & similitarity.

Additionally, my visualization should have been ordered in terms of smallest percentage seen on the inner most circle and high percentage seen on the outer most circle. While this is an effectiveness issue, I found that it'd still be important to note.

1.3 Propose an alternative encoding for the following visualization. Compare the visualizations based on perception. (10 points)

Again, either-hand draw or use an application to create a sketched solution. Upload an image and describe the differences between your solution and the FiveThirtyEight image in terms of perception (specifically for the task of comparing one movie to another).

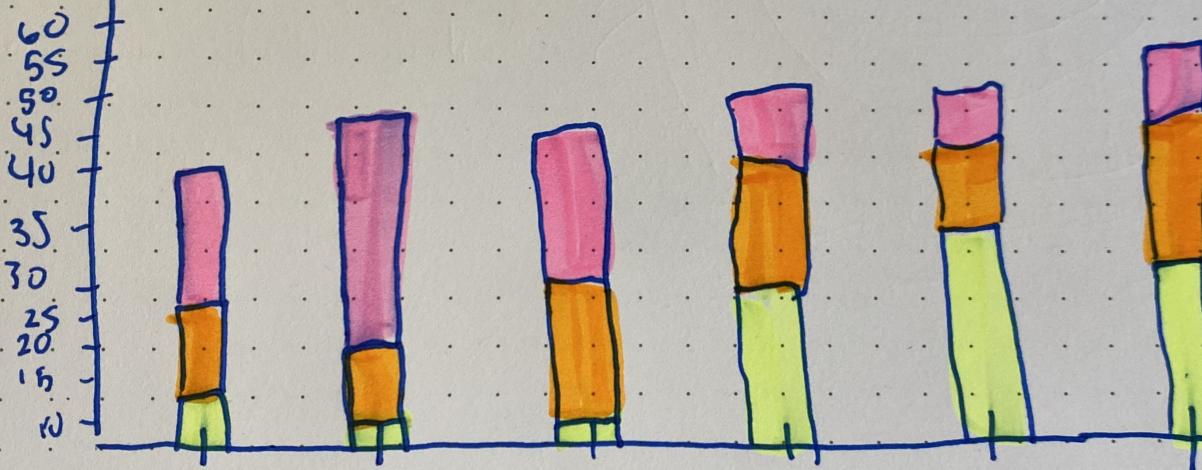


1.3 Answer

% Green Rating

How People Rate the 'Star Wars' Movies

TOP THIRD
MID THIRD
BOTTOM THIRD



The Phantom Menace Attack of the Clones Revenge of the Sith
A New Hope Empire strikes back Return of the Jedi.

MOVIES

fiserv.

I believe a stacked bar chart more closely follows Gestalt's principles of perception, in particular, the principle of connectivity. The example lacks connectivity, so I chose to show it in my proposed visualization.

Part 2. Altair programming exercise (70 points)

We have provided you with some code and parts of the article [America's Favorite 'Star Wars' Movies \(And Least Favorite Characters\)](https://fivethirtyeight.com/features/americas-favorite-star-wars-movies-and-least-favorite-characters/) (<https://fivethirtyeight.com/features/americas-favorite-star-wars-movies-and-least-favorite-characters/>). This article is based on the dataset:

1. [StarWars \(data/StarWars.csv\)](https://github.com/fivethirtyeight/data/tree/master/star-wars-survey) Created by FiveThirtyEight based on a survey ran through SurveyMonkey Audience, surveying 1,186 respondents from June 3 to 6 2014. Available [online] (<https://github.com/fivethirtyeight/data/tree/master/star-wars-survey>)

To earn points for this assignment, you must:

- Recreate the visualizations in the article (replace the images in the article with a code cell that creates a visualization). We provide one example. Each visualization is worth 10 points (40 points/ 10 each x 4 total).
 - *Partial credit can be granted for each visualization (up to 5 points) if you provide the grammar of graphics description of the visualization without a functional Altair implementation*
- Propose one alternative visualization for one of the article visualizations. Add a short paragraph describing why your visualization is more effective based on principles of perception/cognition. (15 points/ 10 points plot + 5 justification)
- Propose a new visualization to complement a part of the article. Add a short paragraph justifying your decisions in terms of Perception/Cognition processes. (15 points/ 10 points plot + 5 justification)

```
In [2]: import pandas as pd
import altair as alt
import numpy as np
import math
```

```
In [3]: # enable correct rendering
alt.renderers.enable('default')
```

```
Out[3]: RendererRegistry.enable('default')
```

```
In [4]: # uses intermediate json files to speed things up
alt.data_transformers.enable('json')
```

```
Out[4]: DataTransformerRegistry.enable('json')
```

```
In [5]: def load_starwars_data(filename='assets/StarWars.csv'):
    sw = pd.read_csv(filename, encoding='latin1')

    # Some format is needed for the survey dataframe, we provide the formatted dataset in a dataframe
    sw = sw.rename(columns={'Have you seen any of the 6 films in the Star Wars franchise?': 'seen_any_movie',
                           'Do you consider yourself to be a fan of the Star Wars film franchise?': 'fan',
                           'Which of the following Star Wars films have you seen? Please select all that apply.': 'seen_EI',
                           'Unnamed: 4': 'seen_EII',
                           'Unnamed: 5': 'seen_EIII',
                           'Unnamed: 6': 'seen_EIV',
                           'Unnamed: 7': 'seen_EV',
                           'Unnamed: 8': 'seen_EVI',
                           'Please rank the Star Wars films in order of preference with 1 being your favorite film in the franchise and 6 being your least favorite. ': 'rank_EI',
                           'Unnamed: 10': 'rank_EII',
                           'Unnamed: 11': 'rank_EIII',
                           'Unnamed: 12': 'rank_EIV',
                           'Unnamed: 13': 'rank_EV',
                           'Unnamed: 14': 'rank_EVI',
                           'Please state whether you view the following characters favorably, unfavorably, or are unfamiliar with him/her. ': 'char_fav',
                           'Unnamed: 16': 'Luke Skywalker',
                           'Unnamed: 17': 'Princess Leia Organa',
                           'Unnamed: 18': 'Anakin Skywalker',
                           'Unnamed: 19': 'Obi Wan Kenobi',
                           'Unnamed: 20': 'Emperor Palpatine',
                           'Unnamed: 21': 'Darth Vader',
                           'Unnamed: 22': 'Lando Calrissian',
                           'Unnamed: 23': 'Boba Fett',
                           'Unnamed: 24': 'C-3PO',
                           'Unnamed: 25': 'R2 D2',
                           'Unnamed: 26': 'Jar Jar Binks',
                           'Unnamed: 27': 'Padme Amidala',
                           'Unnamed: 28': 'Yoda',
                           })
    sw = sw.drop([0])
    return(sw)

sw = load_starwars_data()
```

```
In [6]: # take a peak to look at the data  
sw.sample(5)
```

Out[6]:

RespondentID	seen_any_movie	fan	seen_EI	seen_EII	seen_EIII	seen_EIV	seen_EV	seen_EVI	rank_EI	...	Yoda	Which character shot first?	Are you familiar with the Expanded Universe?	Do you consider yourself to be a fan of the Expanded Universe? æ	Do you consider yourself to be a fan of the Star Trek franchise?	Gender	Age	
563	3.290225e+09	No	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	No	Female	> 60
32	3.292339e+09	Yes	Yes	Star Wars: Episode I The Phantom Menace	Star Wars: Episode II Attack of the Clones	Star Wars: Episode III Revenge of the Sith	Star Wars: Episode IV A New Hope	Star Wars: Episode V The Empire Strikes Back	Star Wars: Episode VI Return of the Jedi	6	...	Very favorably	Han	Yes	Yes	Yes	Male	30-44
907	3.289410e+09	Yes	Yes	Star Wars: Episode I The Phantom Menace	Star Wars: Episode II Attack of the Clones	Star Wars: Episode III Revenge of the Sith	Star Wars: Episode IV A New Hope	Star Wars: Episode V The Empire Strikes Back	Star Wars: Episode VI Return of the Jedi	5	...	Very favorably	I don't understand this question	Yes	No	Yes	Male	18-29
1165	3.288415e+09	Yes	No	Star Wars: Episode I The Phantom Menace	Star Wars: Episode II Attack of the Clones	Star Wars: Episode III Revenge of the Sith	NaN	Star Wars: Episode V The Empire Strikes Back	Star Wars: Episode VI Return of the Jedi	5	...	Neither favorably nor unfavorably (neutral)	Han	No	NaN	No	Male	45-60
136	3.291391e+09	Yes	Yes	NaN	NaN	NaN	NaN	Star Wars: Episode V The Empire Strikes Back	Star Wars: Episode VI Return of the Jedi	4	...	Very favorably	Han	No	NaN	Yes	Female	> 60

America's Favorite 'Star Wars' Movies (And Least Favorite Characters)

Original article available at FiveThirtyEight (<https://fivethirtyeight.com/features/americas-favorite-star-wars-movies-and-least-favorite-characters/>)

By Walt Hickey (<https://fivethirtyeight.com/contributors/walt-hickey/>)

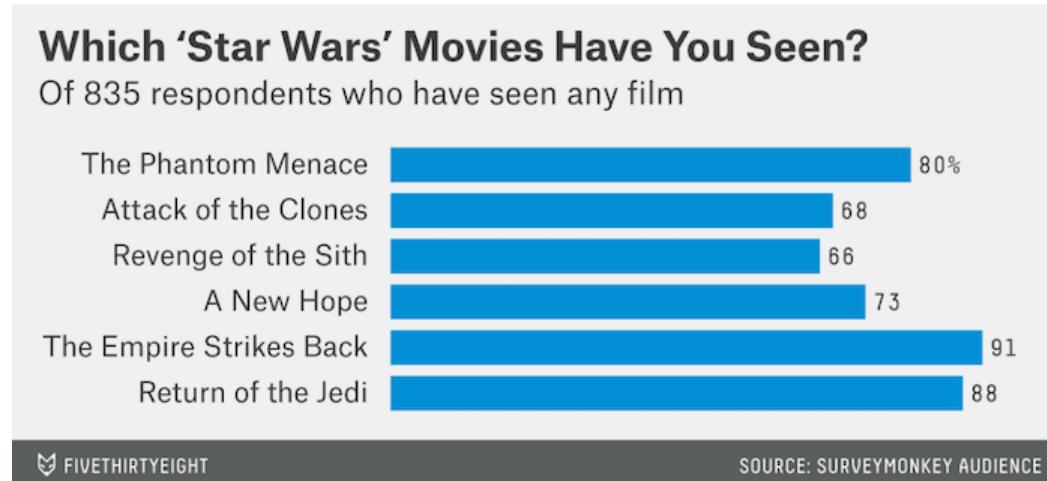
Filed under [Movies](https://fivethirtyeight.com/tag/movies/)

Get the data on GitHub (<https://github.com/fivethirtyeight/data/tree/master/star-wars-survey>)

This week, I caught a sneak peek [of the X-Wing fighter](http://www.wired.com/2014/07/star-wars-episode-vii-x-wing) (<http://www.wired.com/2014/07/star-wars-episode-vii-x-wing>) from the new “Star Wars” films in production. The forthcoming movies — and the middling response to the most recent trilogy — provide a perfect excuse to examine some questions I’ve long wanted answers to: How many people are “Star Wars” fans? Does the rest of America realize that “The Empire Strikes Back” is clearly the best of the bunch? Which characters are most well-liked and most hated? And who shot first, Han Solo or Greedo?

We ran a poll through [SurveyMonkey Audience](https://www.surveymonkey.com/mp/audience) (<https://www.surveymonkey.com/mp/audience>), surveying 1,186 respondents from June 3 to 6 (the [data](https://github.com/fivethirtyeight/data/tree/master/star-wars-survey) (<https://github.com/fivethirtyeight/data/tree/master/star-wars-survey>) is available [on GitHub](https://github.com/fivethirtyeight/data) (<https://github.com/fivethirtyeight/data>)). Seventy-nine percent of those respondents said they had watched at least one of the “Star Wars” films. This question, incidentally, had a substantial difference by gender: 85 percent of men have seen at least one “Star Wars” film compared to 72 percent of women. Of people who have seen a film, men were also more likely to consider themselves a fan of the franchise: 72 percent of men compared to 60 percent of women.

We then asked respondents which of the films they had seen. With 835 people responding, here’s the probability that someone has seen a given “Star Wars” film given that they have seen any Star Wars film:



```
In [7]: # We're going to fix the labels a bit so will create a mapping to the full names
# and have the full sort order
def gen_episode_names_df():
    episodes = ['EI', 'EII', 'EIII', 'EIV', 'EV', 'EVI']
    names = {
        'EI' : 'The Phantom Menace', 'EII' : 'Attack of the Clones', 'EIII' : 'Revenge of the Sith',
        'EIV': 'A New Hope', 'EV': 'The Empire Strikes Back', 'EVI' : 'The Return of the Jedi'
    }

    # we're also going to use this order to sort, so names_1 will now have our sort order
    return episodes, names, [names[ep] for ep in episodes]

episodes, names, names_1 = gen_episode_names_df()
```

```
In [8]: # let's inspect what we've generated. These will be useful to you below
print("abbreviated list (sorted):\n ",episodes)
print("\nmapping between abbreviated names and full titles:\n ",names)
print("\nfull titles, sorted:\n ",names_l)

abbreviated list (sorted):
['EI', 'EII', 'EIII', 'EIV', 'EV', 'EVI']

mapping between abbreviated names and full titles:
{'EI': 'The Phantom Menace', 'EII': 'Attack of the Clones', 'EIII': 'Revenge of the Sith', 'EIV': 'A New Hope', 'EV': 'The Empire Strikes Back', 'EVI': 'The Return of the Jedi'}

full titles, sorted:
['The Phantom Menace', 'Attack of the Clones', 'Revenge of the Sith', 'A New Hope', 'The Empire Strikes Back', 'The Return of the Jedi']
```

```
In [9]: # let's do some data pre-processing... recall that sw (star wars) has everything
```

```
def get_seen_at_least_one_df(indf,eps):
    # input: indf the data file as formatted above
    # input: eps a list of episodes (movies)
    # returns a subset of the dataset

    # We want to only use those people who have seen at least one movie, let's get the people, toss NAs
    # and get the total count

    # find people who have at least one of the columns (seen_*) not NaN
    salo = indf.dropna(subset=['seen_' + ep for ep in eps],how='all')

    return(salo)
```

```
In [10]: seen_at_least_one = get_seen_at_least_one_df(sw, episodes)
print("total who have seen at least one: ", len(seen_at_least_one),"\nSample:")
display(seen_at_least_one.sample(5))
```

```
total who have seen at least one:  835
Sample:
```

```
In [11]: # for each movie, we're going to calculate the percents and generate a new data frame

def gen_seen_percent_df(inpf):
    total = len(inpf)
    percs = []

    # loop over each column and calculate the number of people who have seen the movie
    # specifically, filter out the people who are *NaN* for a specific episode (e.g., ep_EII), count them
    # and divide by the percent
    for seen_ep in ['seen_' + ep for ep in episodes]:
        perc = len(seen_at_least_one[~ pd.isna(seen_at_least_one[seen_ep])]) / total
        percs.append(perc)

    # at this point percs is holding our percentages

    # now we're going use a trick to make tuples--pairing names with percents--using "zip" and then make a dataframe
    tuples = list(zip([names[ep] for ep in episodes],percs))
    seen_per_df = pd.DataFrame(tuples, columns = ['Name', 'Percentage'])
    return(seen_per_df)
```

```
In [12]: seen_per_df = gen_seen_percent_df(seen_at_least_one)
```

```
In [13]: # let's see what's inside
```

```
In [14]: def gen_percent_vis(indf):
    # input: indf, the dataframe as seen_per_df
    # output: simple altair bar chart

    # ok, time to make the chart... let's make a bar chart (use mark_bar)
    bars = alt.Chart(indf).mark_bar(size=20).encode(
        # encode x as the percent, and hide the axis
        x=alt.X(
            'Percentage',
            axis=None),
        y=alt.Y(
            # encode y using the name, use the movie name to label the axis, sort using the names_1
            'Name:N',
            axis=alt.AxisTickCount=5, title=''),
            # we give the sorting order to avoid alphabetical order
            sort=names_1
        )
    )

    # at this point we don't really have a great plot (it's missing the annotations, titles, etc.)
    return(bars)

bars = gen_percent_vis(seen_per_df)
```

```
In [15]: # display it  
bars
```

Out[15]:

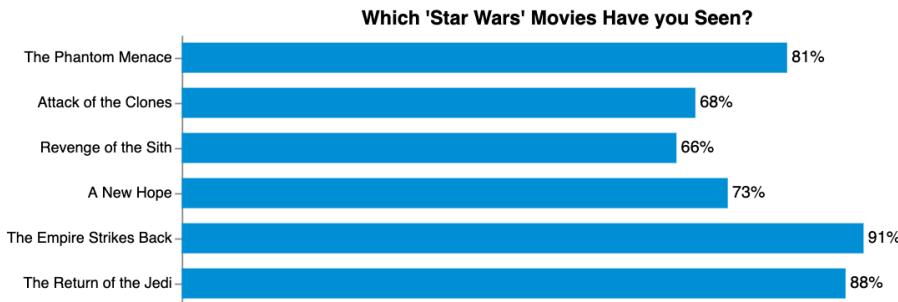


```
In [16]: def augment_percent_vis(base):  
    # input: base (the base vis, i.e., bars as above)  
    # we're going to overlay the text with the percentages, so let's make another visualization  
    # that's just text labels  
  
    text = base.mark_text(  
        align='left',  
        baseline='middle',  
        dx=3  # Nudges text to right so it doesn't appear on top of the bar  
    ).encode(  
        # we'll use the percentage as the text  
        text=alt.Text('Percentage:Q',format='.0%')  
    )  
  
    # finally, we're going to combine the bars and the text and do some styling  
seen_movies = (text + base).configure_mark(  
    # we don't love the blue4  
    color="#008fd5"  
).configure_view(  
    # we don't want a stroke around the bars  
    strokeWidth=0  
).configure_scale(  
    # add some padding  
    bandPaddingInner=0.2  
).properties(  
    # set the dimensions of the visualization  
    width=500,  
    height=180  
).properties(  
    # add a title  
    title="Which 'Star Wars' Movies Have you Seen?"  
)  
  
return(seen_movies)
```

note that we are NOT formatting this in the Five Thirty Eight Style yet... we'll leave that to you to figure out

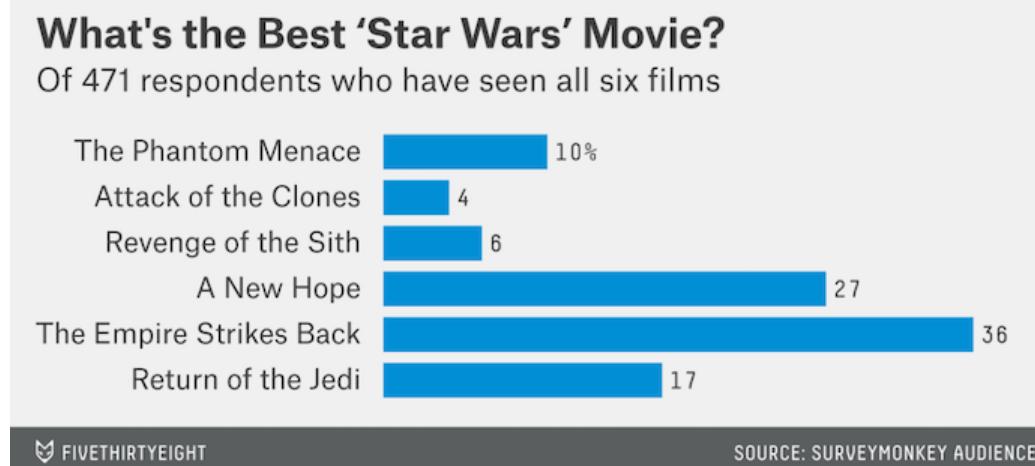
```
In [17]: # let's see it  
augment_percent_vis(bars)
```

Out[17]:



So we can see that "Star Wars: Episode V — The Empire Strikes Back" is the film seen by the most number of people, followed by "Star Wars: Episode VI — Return of the Jedi." Appallingly, more people reported seeing "Star Wars: Episode I — The Phantom Menace" than the original "Star Wars" (renamed "Star Wars: Episode IV — A New Hope").

So, which movie is the best? We asked the subset of 471 respondents who indicated they have seen every "Star Wars" film to rank them from best to worst. From that question, we calculated the share of respondents who rated each film as their favorite.



** Homework note: Click [here](#) (`assets/best_movie.png`) to see a version of this plot generated in Altair.

2.1 What's the best 'Star Wars' movie? Recreate the above image using altair (10 POINTS)

Recreate the image above using Altair. Match the "538" style as best you can (hint: look at the altair lab at the start of the semester). We expect you to *at least* match the [our version](#) (`assets/best_movie.png`) of the chart that was created in Altair.

```
In [18]: # Recreate this image using Altair
# match the "538 style" as best you can (hint: look at the altair lab at the start of the semester)

def gen_best_vis(inpf,eps):

    best_sw = inpf.dropna(subset=['seen_' + ep for ep in episodes], how ='any')

    total = len(best_sw)
    percs = []

    for ep_rank in ['rank_' + ep for ep in episodes]:
        perc_1 = len(best_sw[best_sw[ep_rank] == '1']) / total
        percs.append(perc_1)

    tuples = list(zip([names[ep] for ep in episodes], percs))
    rank_df = pd.DataFrame(tuples, columns = ['Name', 'Percentage'])

    bars = alt.Chart(rank_df).mark_bar(size=20).encode(
        x = alt.X(
            'Percentage:Q',
            axis = None),
        y = alt.Y(
            'Name:N',
            axis=alt.AxisTickCount=5, title=''),
        sort=names_1
    )

    text = bars.mark_text(
        align = 'left',
        baseline = 'middle',
        dx=3
    ).encode(
        text = alt.Text('Percentage:Q', format=' .0%'))

    rank_movies = (text +bars).configure_mark(
        color = '#008fd5'
    ).configure_view(
        strokeWidth=0
    ).configure_scale(
        bandPaddingInner=0.2
    ).properties(
        width=500,
        height=180
    ).properties(
        title="What's the Best 'Star Wars' Movie?"
        #ADD subtitle
    ).configure_title(
        anchor='start'
    ).configure_axis(
        labelColor='gray'
    )

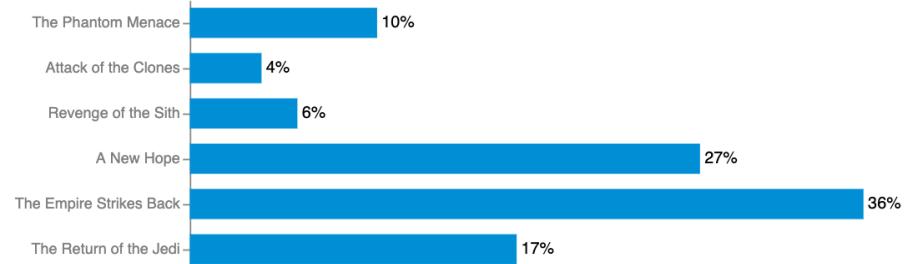
    return rank_movies
# input: inpf, the star wars dataset
# input: eps, the list of episodes
```

```
# YOUR CODE HERE
#raise NotImplemented()

alt.themes.enable('fivethirtyeight')
```

In [19]: # test your code
gen_best_vis(sw,episodes)

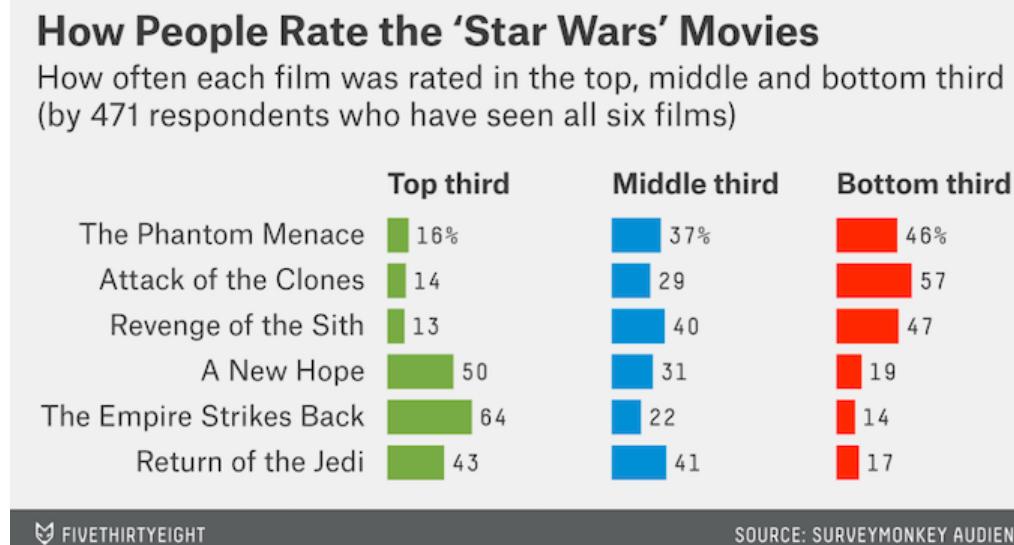
Out[19]: What's the Best 'Star Wars' Movie?



Make sure to style your visualization to match the original the best you can

We can also drill down and find out, generally, how people rate the films. Overall, fans broke into two camps: those who preferred the original three movies and those who preferred the three prequels. People who said “The Empire Strikes Back” was their favorite were also likely to rate “A New Hope” and “Return of the Jedi” higher as well. Those who rated “The Phantom Menace” as the best film were more likely to rate prequels higher.

This chart shows how often each film was rated in the top third (best or second-best), the middle third (third or fourth) or the bottom third (second-worst or worst). It's a more nuanced take on the series:



** Homework note: Click [here](#) (`assets/people_rate.png`) to see a version of this plot generated in Altair.

2.2 How people rate the 'Star Wars' movie? Recreate the above image using altair (10 POINTS)

In [20]: #

```
In [21]: #def gen_rate_vis(inpf,eps):  
  
seen = sw.dropna(subset=['seen_' + ep for ep in episodes],how='any')  
total = len(seen)  
  
percs = []  
  
for rank_ep in [f'rank_{ep}' for ep in episodes]:  
    perc = (seen[seen[rank_ep] == '1'].shape[0] + seen[seen[rank_ep] == '2'].shape[0]) / total  
    percs.append(perc)  
  
tuples = list(zip([names[ep] for ep in episodes],percs))  
top = pd.DataFrame(tuples, columns = ['Name', 'Percentage'])  
top  
  
top_bars = alt.Chart(top).mark_bar(size=20, color="#77AB43").encode(  
  
    x=alt.X(  
        'Percentage',  
        axis=None),  
    y=alt.Y(  
  
        'Name:N',  
        axis=alt.AxisTickCount=0, title='',  
        sort=names_l  
    )  
)  
  
top_bars  
  
text = top_bars.mark_text(  
    align='left',  
    baseline='middle',  
    dx=3  
)  
.encode(  
    text=alt.Text('Percentage:Q',format='.0%')  
)  
  
top_movies = (text + top_bars).properties(  
    width=60,  
    height=160  
)  
.properties(  
    title={  
        "text": ["Top third"]},  
)  
  
top_movies  
  
percs = []  
  
for rank_ep in [f'rank_{ep}' for ep in episodes]:  
    perc = (seen[seen[rank_ep] == '3'].shape[0] + seen[seen[rank_ep] == '4'].shape[0]) / total  
    percs.append(perc)
```

```

tuples = list(zip([names[ep] for ep in episodes],percs))
mid = pd.DataFrame(tuples, columns = ['Name', 'Percentage'])
mid

mid_bars = alt.Chart(mid).mark_bar(size=20, color="#008FD5").encode(
    x=alt.X(
        'Percentage',
        axis=None),
    y=alt.Y(
        'Name:N',
        axis=None,
        sort=names_1
    )
)

mid_bars

text = mid_bars.mark_text(
    align='left',
    baseline='middle',
    dx=3
).encode(
    text=alt.Text('Percentage:Q',format='.0%')
)

mid_movies = (text + mid_bars).properties(
    width=60,
    height=160
).properties(
    title={
        "text": ["Middle third"]},
)

mid_movies

percs = []

for rank_ep in [f'rank_{i}' + ep for ep in episodes]:
    perc = (seen[seen[rank_ep] == '5'].shape[0] + seen[seen[rank_ep] == '6'].shape[0]) / total
    percs.append(perc)

tuples = list(zip([names[ep] for ep in episodes],percs))
bot = pd.DataFrame(tuples, columns = ['Name', 'Percentage'])
bot

bot_bars = alt.Chart(bot).mark_bar(size=20, color='red').encode(

```

```

x=alt.X(
    'Percentage',
    axis=None),
y=alt.Y(
    'Name:N',
    axis=None,
    sort=names_1
)
)

bot_bars

text = bot_bars.mark_text(
    align='left',
    baseline='middle',
    dx=3
).encode(
    text=alt.Text('Percentage:Q',format='.0%')
)

bot_movies = (text + bot_bars).properties(
    width=60,
    height=160
).properties(
    title={
        "text": ["Bottom third"]
    }
)

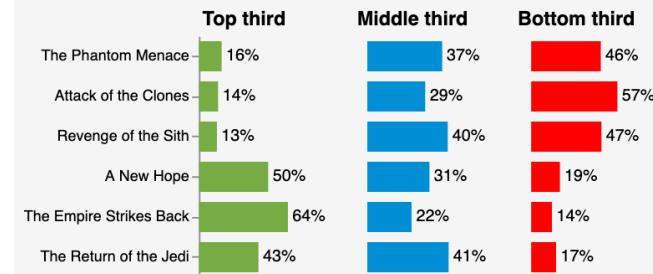
bot_movies

alt.hconcat(top_movies , mid_movies ,bot_movies).properties(
    title={
        "text": ["How People Rate the 'Star Wars' Movies"],
        "subtitle": ["How often each film was rated in the top, middle and bottom third", '(by 471 respondents who have seen all six films)', ''],
        "fontSize": 16
    }
).configure_scale(
    bandPaddingInner=1
).configure(
    # customize background color
    background="whitesmoke"
).configure_view(
    strokeWidth=0
).configure_axis(
    grid=False, domain=False
).configure_axis(
    labelFontSize=10
).configure_legend(
    gradientLength=800,
    gradientThickness=60
)

```

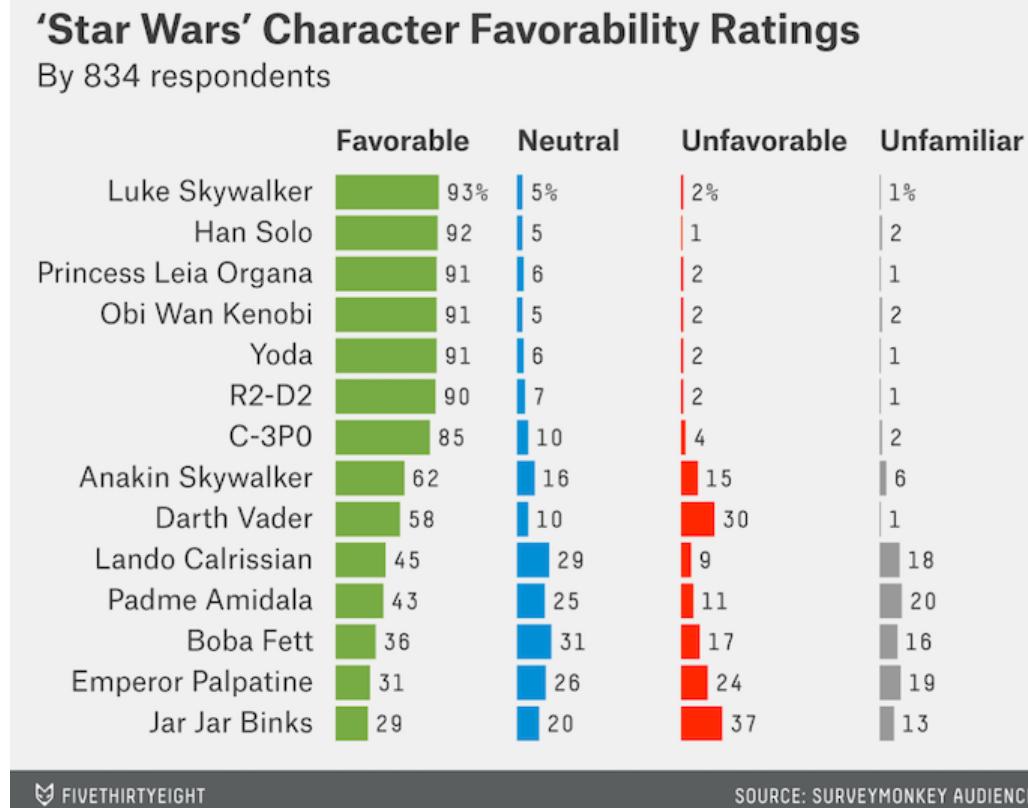
Out[21]:

How People Rate the 'Star Wars' Movies
How often each film was rated in the top, middle and bottom third
(by 471 respondents who have seen all six films)

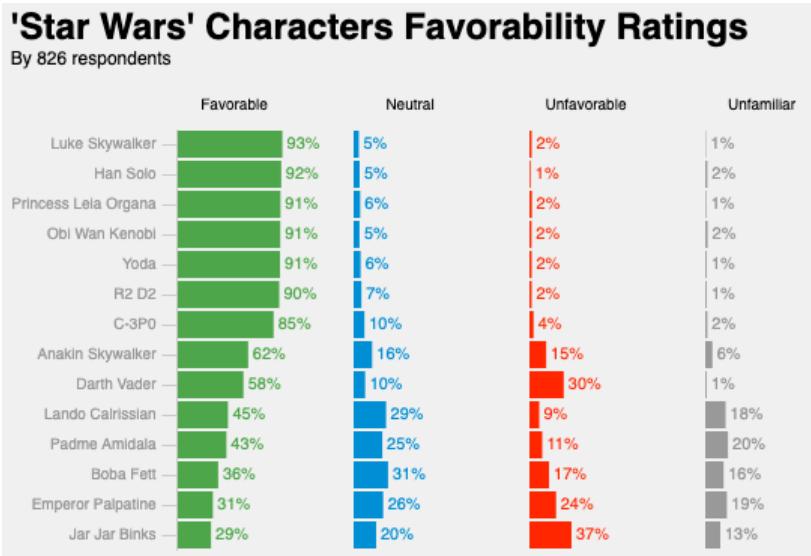


```
In [22]: # let's check our solution
#gen_rate_vis(sw,episodes)
```

Finally, we took a boilerplate format used by political favorability polls — “Please state whether you view the following characters favorably, unfavorably, or are unfamiliar with him/her” — and asked respondents to rate characters in the series.



** Homework note. Here's an example solution generated in Altair:



2.3 Star Wars' Characters Favorability Ratings. Recreate the above image using altair (10 POINTS)

```
In [23]: # Recreate this image using altair here (10 POINTS)

def gen_favor_vis(inpf):
    # input: inpf, the star wars dataset

    # YOUR CODE HERE
    #raise NotImplemented()
```

```
In [25]: # let's test the solution
gen_favor_vis(sw)
```

You read that correctly. Jar Jar Binks has a lower favorability rating than the actual personification of evil in the galaxy.

And for those of you who want to know the impact that [historical revisionism](http://en.wikipedia.org/wiki/Han_shot_first) (http://en.wikipedia.org/wiki/Han_shot_first) can have on a society:

Who Shot First?

According to 834 respondents



 FIVETHIRTYEIGHT

SOURCE: SURVEYMONKEY AUDIENCE

** Homework note: Click [here](#) to see a version of this plot generated in Altair. You may find that you don't get 834 rows (as 538 did) but the percents should still work.

2.4 Who shot first? Recreate the above image using altair (10 POINTS)

```
In [26]: # Recreate this image using altair here (10 POINTS)
#def gen_first_shot_vis(inpf):
    # input: inpf, the star wars dataset

def gen_first_shot_vis(inpf):

    best_sw = inpf.dropna(subset=['seen_' + ep for ep in episodes], how ='any')

    Han = seen["Which character shot first?"] == 'Han'
    count_Han = sum(Han)

    Greedo = seen["Which character shot first?"] == 'Greedo'
    count_Greedo = sum(Greedo)

    question = seen["Which character shot first?"] == "I don't understand this question"
    count_question = sum(question)

    perc_Han = count_Han / total
    perc_Greedo = count_Greedo / total
    perc_question = count_question / total

    names = {'Name':[ "Han", "Greedo", "I don't understand this question"], "percentage": [perc_Han, perc_Greedo, perc_question]}
    characters = pd.DataFrame(data=names)
    characters

    names =[ 'Han', 'Greedo', "I don't understand this question"]

    shot_bars = alt.Chart(characters).mark_bar(size=20, color="#008FD5").encode(
        x=alt.X(
            'percentage',
            axis=None),
        y=alt.Y(
            'Name:N',
            axis=alt.Axis(tickCount=5, title=''),
            sort=names
        )
    )

    shot_bars

    text = shot_bars.mark_text(
        align='left',
        baseline='middle',
        dx=3
    ).encode(
        text=alt.Text('percentage:Q', format='.0%')
    )

    shot_movies = (text + shot_bars).configure_mark(
        color="#008FD5"
    ).configure_scale(
        bandPaddingInner=0.1
    ).properties(
        width=600,
        height=200,
        title={
            "text": ["Who Shot First?"]
        },
        padding=10
    ).configure_view(strokeWidth=0)
```

```

        "subtitle": ["By 828 respondents", ' ']}
    ).configure(
background="whitesmoke"
).configure_title(
    fontSize=30, anchor='start'
).configure_view(
    strokeWidth=0
).configure_axis(
    grid=False, domain=False
)

return shot_movies

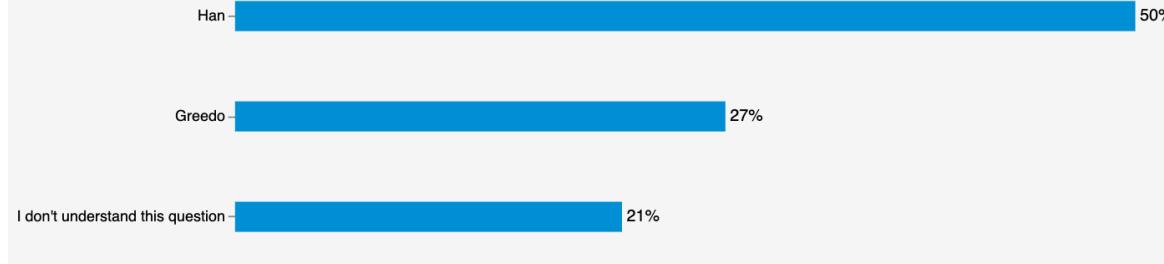
alt.themes.enable('fivethirtyeight')

```

In [27]: # test our solution
gen_first_shot_vis(sw)

Out[27]: **Who Shot First?**

By 828 respondents



In [28]: # Recreate this image using altair here (10 POINTS)

#def gen_first_shot_vis(inpf):
 # input: inpf, the star wars dataset

In [29]: # test our solution
#gen_first_shot_vis(sw)

2.5.1 Make your own (15 points/ 10 points plot + 5 justification)

Propose and code an alternative visualization for one of the visualizations *already in the article*. Add a short paragraph describing why your visualization is more (or less -- you can go out of your way to make a bad vis) effective based on principles of perception/cognition.

If you feel your visualization is worse, that's ok! Just tell us why.

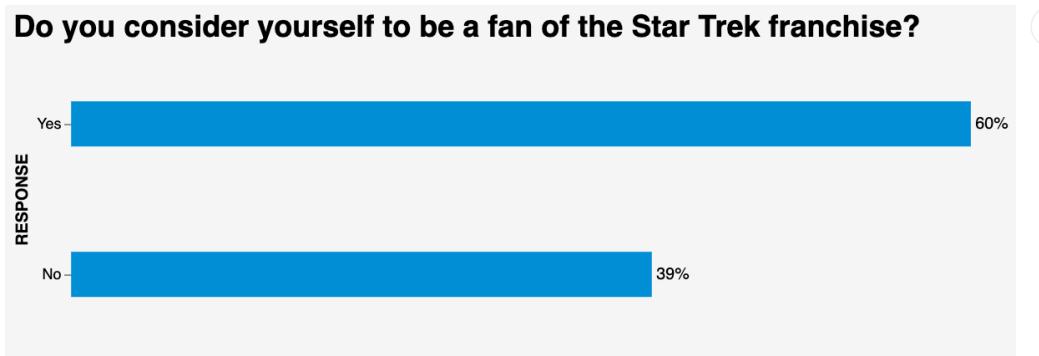
```
In [30]: def gen_first_shot_vis(inpf):  
  
    best_sw = inpf.dropna(subset=['seen_' + ep for ep in episodes], how ='any')  
  
    yes = seen["Do you consider yourself to be a fan of the Star Trek franchise?"] == 'Yes'  
    count_yes = sum(yes)  
  
    no = seen["Do you consider yourself to be a fan of the Star Trek franchise?"] == 'No'  
    count_no = sum(no)  
  
    perc_yes = count_yes / total  
    perc_no = count_no / total  
  
    options = {'Name': ["Yes", "No"], "percentage": [perc_yes, perc_no]}\br/>    characters = pd.DataFrame(data=options)  
    characters  
  
    options = ['Yes', 'No']  
  
    bars = alt.Chart(characters).mark_bar(size=30, color="#008FD5").encode(  
        x=alt.X(  
            'percentage',  
            axis=None),  
        y=alt.Y(  
            'Name:N',  
            axis=alt.Axis(title='RESPONSE'),  
            sort=options  
        )  
    )  
  
    text = bars.mark_text(  
        align='left',  
        baseline='middle',  
        dx=3  
    ).encode(  
        text=alt.Text('percentage:Q', format='.0%')  
    )  
  
    question = (text + bars).configure_mark(  
        color='palegreen'  
    ).configure_scale(  
        bandPaddingInner=0.5  
    ).properties(  
        width=600,  
        height=200,  
        title={  
            "text": ["Do you consider yourself to be a fan of the Star Trek franchise?"]}  
    ).configure(  
        background="whitesmoke"  
    ).configure_title(  
        fontSize=20, anchor='start'  
    ).configure_view(  
        strokeWidth=0  
    ).configure_axis(  
        grid=False, domain=False  
    )
```

```
return question

alt.themes.enable('fivethirtyeight')
```

```
In [31]: # test our solution
gen_first_shot_vis(sw)
```

Out[31]: **Do you consider yourself to be a fan of the Star Trek franchise?**



Propose and code an alternative visualization for one of the visualizations already in the article. Add a short paragraph describing why your visualization is more (or less -- you can go out of your way to make a bad vis) effective based on principles of perception/cognition.

While my visualization is informative, it is less effective than the other visualizations. Gestalt's principle of proximity says that objects that are close to one another are seen as more related. The distance that exists between the 'Yes' and 'No' responses are much too far apart. Another principle is the importance of making a focal point stand out. There isn't a focal point within the 'Yes' versus 'No' responses, therefore the visualization lacks a focal point. Should I have added in another attribute such as gender, this could've made all of the difference.

The visualization does follow Gestalt's principles of continuity (elements on a line), similarity, and parallelism.

Doing this comparison tells me that Gestalt Visualizations should actually have multiple attributes being compared within them and if all of the principles are followed, it will be a strong visualization. Mine was just too simple.

2.5.2 Make your own (15 points/ 10 points plot + 5 justification)

Propose and code a *new visualization* to complement a part of the article. Add a short paragraph justifying your decisions in terms of Perception/Cognition processes.

If you feel your visualization is worse, that's ok! Just tell us why.

```
In [32]: def gen_first_shot_vis(inpf):

    best_sw = inpf.dropna(subset=['seen_'] + ep for ep in episodes], how ='any')

    yes = seen["Are you familiar with the Expanded Universe?"] == 'Yes'
    count_yes = sum(yes)

    no = seen["Are you familiar with the Expanded Universe?"] == 'No'
    count_no = sum(no)

    female = seen["Gender"] == 'Female' #
    count_female = sum(female)

    male = seen["Gender"] == 'Male' #
    count_male = sum(male)

    perc_yes = count_yes / total
    perc_no = count_no / total
    perc_female = count_female / total
    perc_male = count_male / total

    options = {'Name':["Yes", "No", "Female", "Male"], "percentage": [perc_yes, perc_no, perc_female, perc_male]}
    characters = pd.DataFrame(data=options)
    characters

    options =[ 'Yes', 'No', 'Female', 'Male']

    bars = alt.Chart(characters).mark_bar(size=30, color="#008FD5").encode(
        x=alt.X(
            'percentage',
            axis=None),
        y=alt.Y(
            'Name:N',
            axis=alt.Axis(title=''),
            sort=options
        )
    )

    text = bars.mark_text(
        align='left',
        baseline='middle',
        dx=3
    ).encode(
        text=alt.Text('percentage:Q', format='.0%')
    )

    question = (text + bars).configure_mark(
        color='palegreen'
    ).configure_scale(
        bandPaddingInner=0.5
    ).properties(
        width=600,
        height=200,
        title={
```

```

        "text": ["Are you familiar with the Expanded Universe?"]}

).configure(
background="whitesmoke"
).configure_title(
    fontSize=20, anchor='start'
).configure_view(
    strokeWidth=0
).configure_axis(
    grid=False, domain=False
)

)

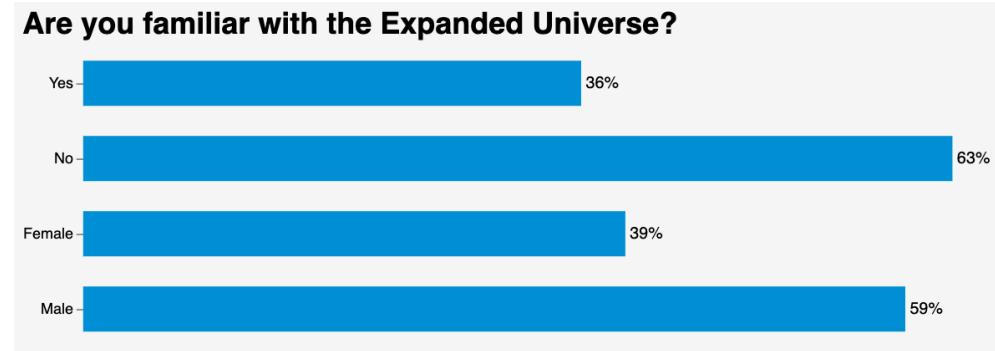
return question

alt.themes.enable('fivethirtyeight')

```

In [33]: # test our solution
gen_first_shot_vis(sw)

Out[33]: Are you familiar with the Expanded Universe?



Provide your justification here

This visualization is not great in terms of perception and cognition. 1st of all, the calculated number of response from female versus male would've been nice. As is, we are just looking at the percentage of M/F who took part in the survey.

In terms of perception/cognition:

- fulfills the principle of closure (complete shapes)
- It would be a better visualization if it fulfilled the proximity principle and was separated based on Female Yes/No and Male Yes/No.

In []: