

# Information Visualization I

## School of Information, University of Michigan

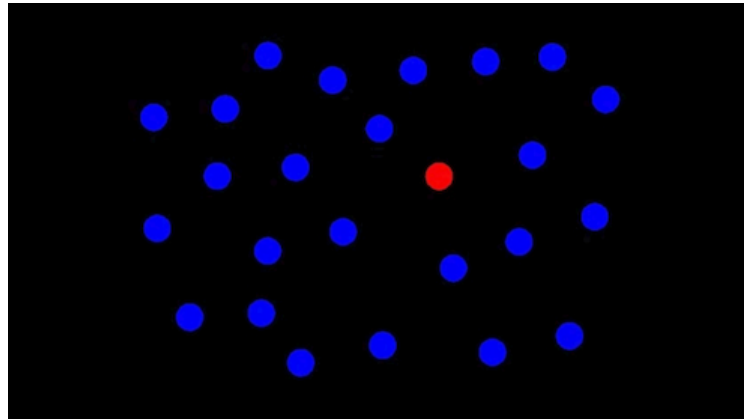
### Week 3:

- 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:

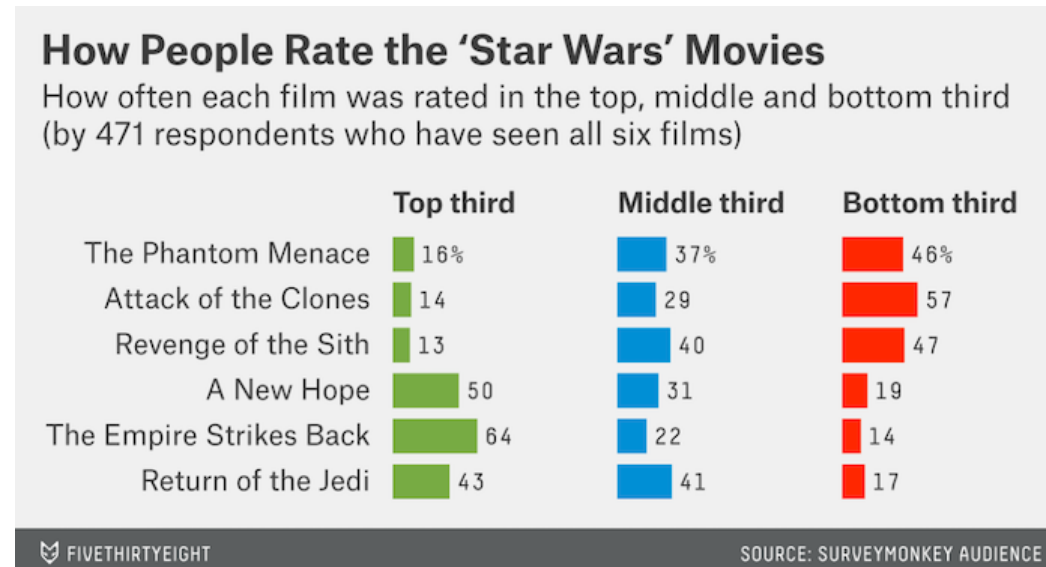
- 1) There will be a couple of places where the numbers you get when you select rows may be a little different than 538, but the percents should still work (e.g., 828 instead of 834). You'll see this in our examples. If you can somehow get the data to match exactly, that's great too.
- 2) Grading for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

## 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

#### A. Chart 1:

1. Variable: movie name, Type: nominal, Encoding: Y-axis
2. Variable: percentage, Type: quantitative, Encoding: length of bars (X-axis) and numbers next to bars
3. Variable: movie rating, Type: ordinal, Encoding: color of bars and position of bars

#### B. Chart 2:

1. Variable: movie name, Type: nominal, Encoding: Y-axis
2. Variable: percentage, Type: quantitative, Encoding: length of bars (X-axis) and numbers next to bars

### 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



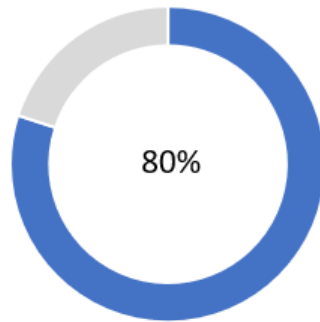
FIVETHIRTYEIGHT

SOURCE: SURVEYMONKEY AUDIENCE

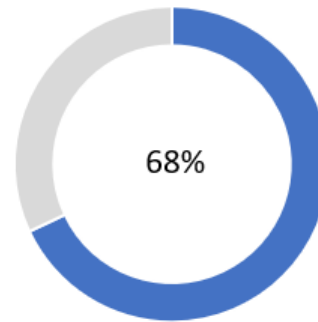
1.2 Answer

## Which 'Star Wars' Movies Have You Seen?

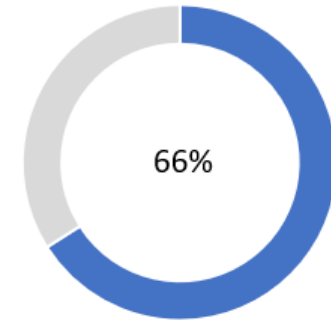
The Phantom Menace



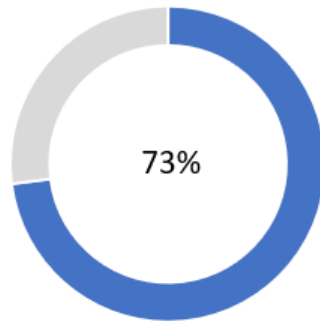
Attack of the Clones



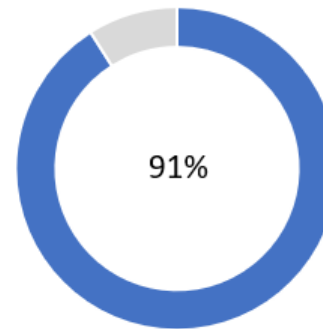
Revenge of the Sith



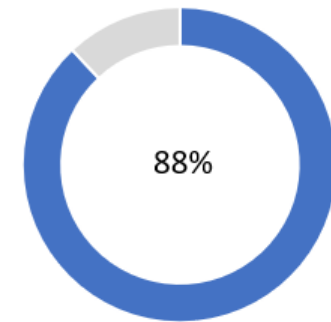
A New Hope



The Empire Strikes Back



Return of the Jedi



In terms of comparing movies, donut charts (a type of pie chart) are an alternative way to visualise the percentage of respondents who have seen them. The FiveThirtyEight image shows the percentage of respondents who have seen the Star Wars films based on the lengths of the bars, while my alternative encoding shows the same information but in terms of the colored parts of the donuts denoting each movie.

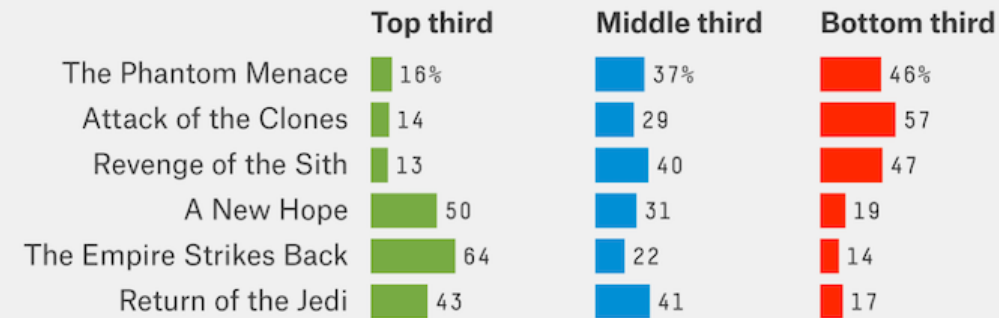
In this case, a bar chart (the FiveThirtyEight image) is probably more effective for the purpose at hand since it is easier to visually compare the lengths of the bars as opposed to comparing the colored parts of the donut charts. In terms of perception, the bar chart in this case enables more accurate detection and estimation compared to the donut charts. Donut charts are probably more useful when answering part-versus-whole type questions.

### 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).

## 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)

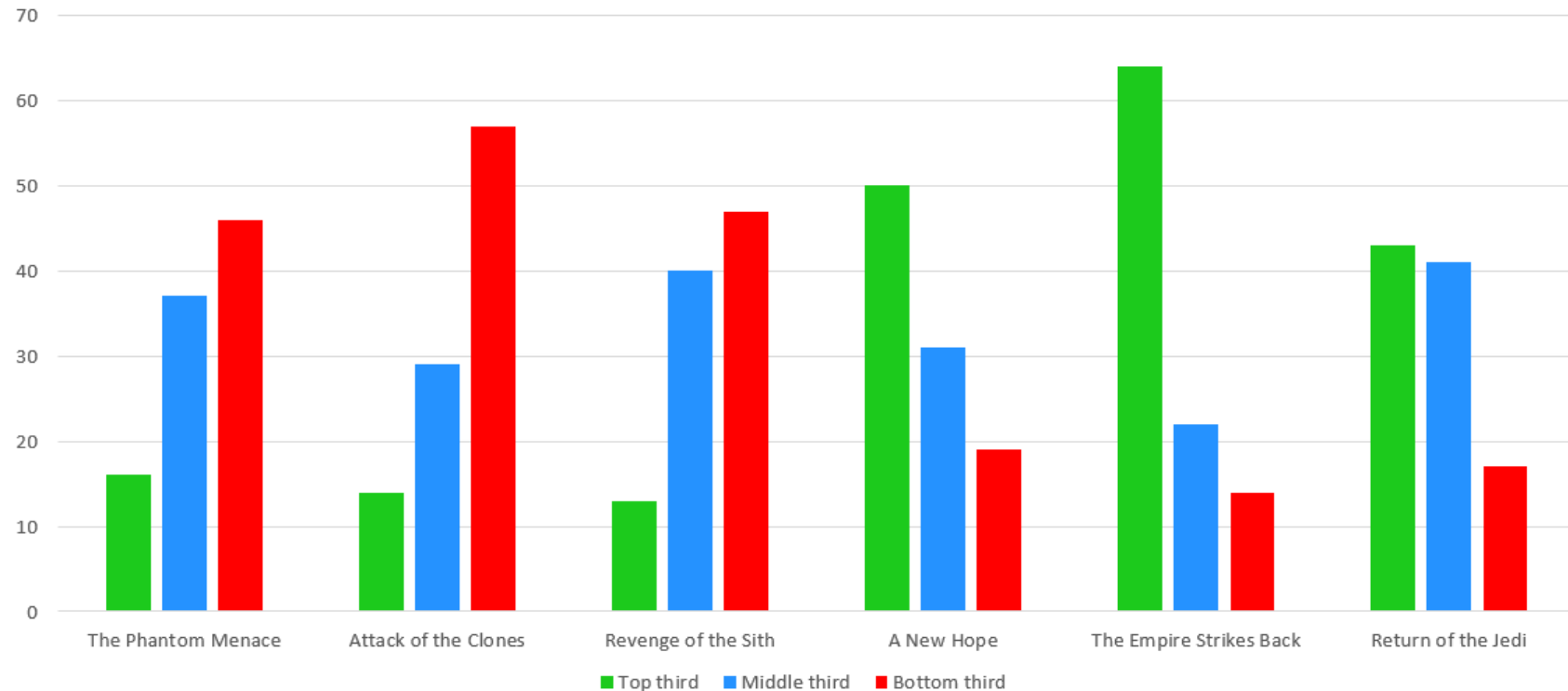


FIVETHIRTYEIGHT

SOURCE: SURVEYMONKEY AUDIENCE

1.3 Answer

## How People Rate the 'Star Wars' Movies



In terms of comparing movies, grouped bar charts are an alternative way to visualise the percentage of respondents who have rated the Star Wars films and their standing for each film in the top, middle and bottom third of the rankings. The FiveThirtyEight image shows the same information but in terms of a horizontally concatenated set of bar charts, which one bar chart for each of the 3 categories (top third, middle third and bottom third). The colors are preserved as is in both visualizations.

In this case, the grouped bar chart solution proposed by me is more perceptive, as it is visually easier to compare the lengths of the bars belonging to different categories since they're side-by-side. This enables seamless comparisons and estimations across different movies and different categories. On the other hand, the bars in the FiveThirtyEight image are more disjointed and placed far apart, making comparisons among movies and categories cumbersome since it is harder to compare the lengths of the bars.

## 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>) (<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 [1]: import pandas as pd
import altair as alt
import numpy as np
import math
```

```
In [2]: # enable correct rendering
alt.renderers.enable('default')
```

```
Out[2]: RendererRegistry.enable('default')
```

```
In [3]: # uses intermediate json files to speed things up
alt.data_transformers.enable('json')
```

```
Out[3]: DataTransformerRegistry.enable('json')
```



```

In [4]: 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 f
                                '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.' : 'Han Solo',
                                '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-3P0',
                                '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 [5]: # take a peak to look at the data
sw.sample(5)
```

Out[5]:

	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	Hous In
713	3.289886e+09	Yes	No	Star Wars: Episode I The Phantom Menace	Star Wars: Episode II Attack of the Clones	NaN	NaN	Star Wars: Episode V The Empire Strikes Back	Star Wars: Episode VI Return of the Jedi	3	...	Very favorably	I don't understand this question	No	NaN	Yes	Male	> 60	100,000–149,999
518	3.290351e+09	Yes	Yes	NaN	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	...	Somewhat favorably	Greedo	No	NaN	Yes	Female	18-29	50,000–99,999
890	3.289450e+09	Yes	No	NaN	NaN	NaN	NaN	Star Wars: Episode V The Empire Strikes Back	NaN	NaN	...	Very favorably	I don't understand this question	No	NaN	No	Female	> 60	50,000–99,999
529	3.290326e+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	...	Neither favorably nor unfavorably (neutral)	Han	Yes	Yes	Yes	Male	30-44	25,000–49,999
1068	3.288578e+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	4	...	Very favorably	I don't understand this question	No	NaN	Yes	Female	45-60	\$150,000+

5 rows × 38 columns

# America’s Favorite ‘Star Wars’ Movies (And Least Favorite Characters)

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

By [Walt Hickey \(https://fivethirtyeight.com/contributors/walt-hickey/\)](https://fivethirtyeight.com/contributors/walt-hickey/).

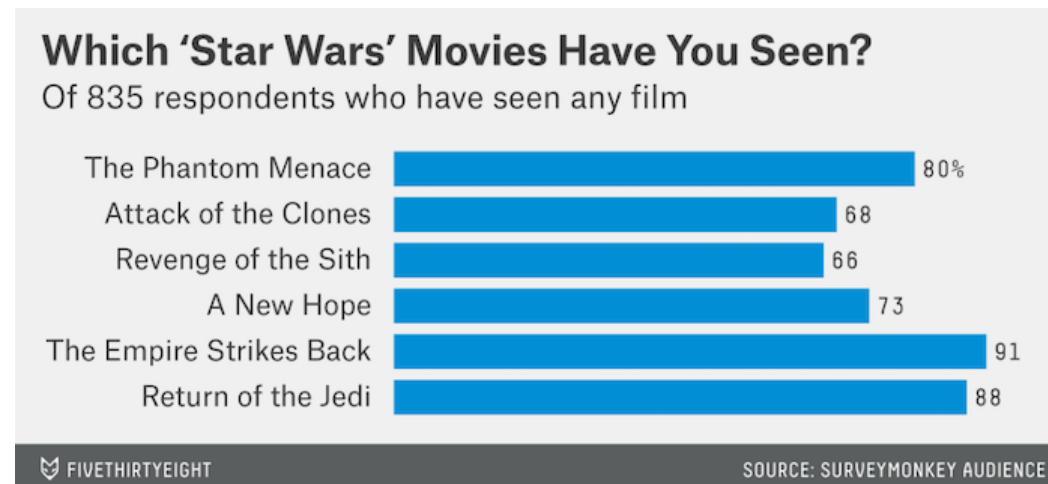
Filed under [Movies](https://fivethirtyeight.com/tag/movies/) (<https://fivethirtyeight.com/tag/movies/>).

Get the data on [GitHub](https://github.com/fivethirtyeight/data/tree/master/star-wars-survey) (<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 [6]: # 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_l will now have our sort order
    return episodes, names, [names[ep] for ep in episodes]

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

```
In [7]: # 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_1)
```

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 [8]: # 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 [9]: 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:

	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	Household Income
96	3.291588e+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	1	...	Very favorably	Han	No	NaN	Yes	Male	> 60	50,000–99,999
392	3.290679e+09	Yes	No	NaN	NaN	NaN	NaN	NaN	Star Wars: Episode VI Return of the Jedi	3	...	Very favorably	I don't understand this question	No	NaN	No	Male	30-44	100,000–149,999
154	3.291268e+09	Yes	No	NaN	NaN	NaN	Star Wars: Episode IV A New Hope	Star Wars: Episode V The Empire Strikes Back	Star Wars: Episode VI Return of the Jedi	4	...	Somewhat favorably	I don't understand this question	No	NaN	Yes	Male	45-60	100,000–149,999
84	3.291655e+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	Greedo	No	NaN	Yes	Male	> 60	50,000–99,999
44	3.292218e+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	18-29	\$150,000+

5 rows × 38 columns



```
In [10]: # 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 [11]: seen_per_df = gen_seen_percent_df(seen_at_least_one)
```

```
In [12]: # let's see what's inside
seen_per_df
```

Out[12]:

	Name	Percentage
0	The Phantom Menace	0.805988
1	Attack of the Clones	0.683832
2	Revenge of the Sith	0.658683
3	A New Hope	0.726946
4	The Empire Strikes Back	0.907784
5	The Return of the Jedi	0.883832

```

In [13]: 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_l
        'Name:N',
        axis=alt.Axis(tickCount=5, title=''),
        # we give the sorting order to avoid alphabetical order
        sort=names_l
    )
)

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

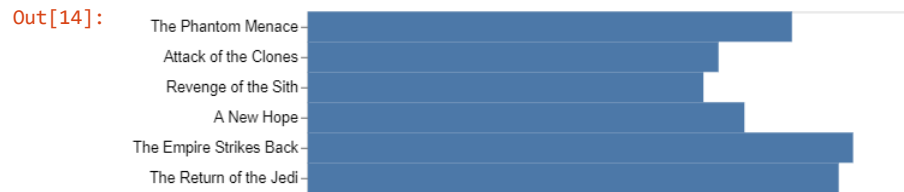
bars = gen_percent_vis(seen_per_df)

```

```

In [14]: # display it
bars

```



```
In [15]: 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

# change to fivethirtyeight theme
alt.renderers.set_embed_options(theme="fivethirtyeight")

text = base.mark_text(
    align='left',
    baseline='middle',
    dx=3, # nudges text to right so it doesn't appear on top of the bar
    fontStyle="bold"
).encode(
    # we'll use the percentage as the text
    text=alt.Text('Percentage:Q',format='.0%')
)

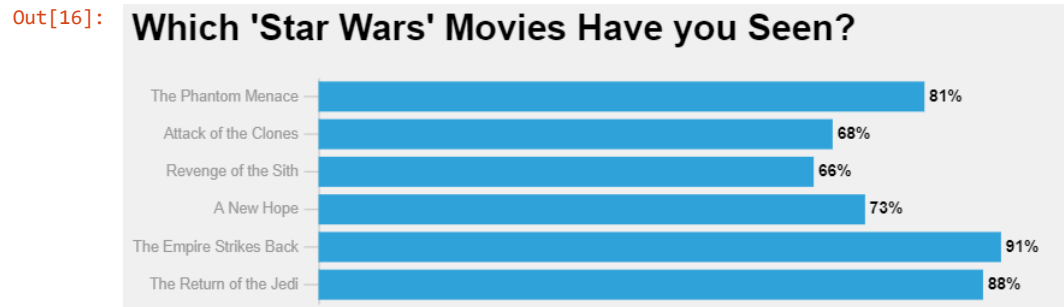
# finally, we're going to combine the bars and the text and do some styling
seen_movies = (text + base).configure_mark(
    # we don't love the blue
    color='#008fd5'
).configure_view(
    # we don't want a stroke around the bars
    strokeWidth=0
).configure_scale(
    # add some padding
    bandPaddingInner=0.2
).properties(
    # set the dimensions of the visualization
    width=500,
    height=150
).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
```

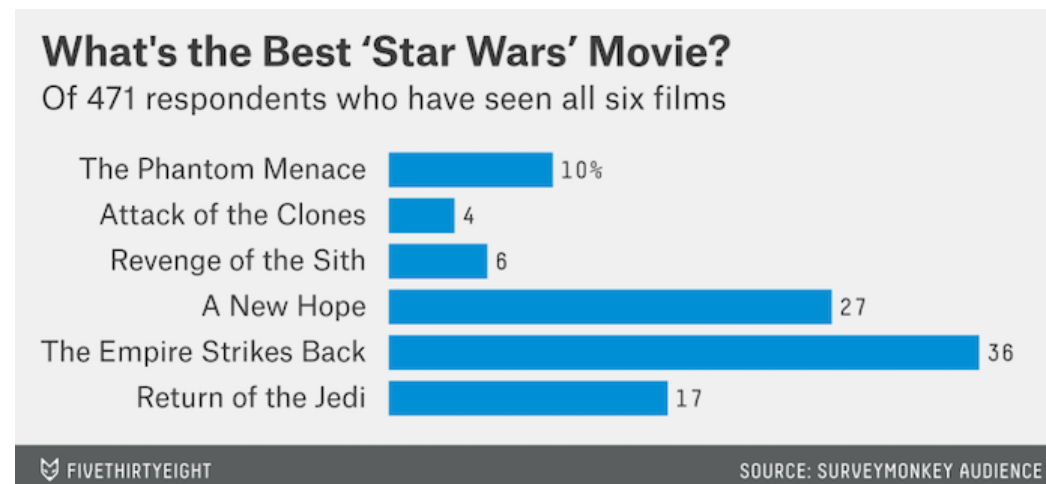


```
In [16]: # Let's see it
augment_percent_vis(bars)
```



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 [17]: # 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):
    # input: inpf, the star wars dataset
    # input: eps, the list of episodes
    # YOUR CODE HERE

    # subset for 471 respondents who have seen all of the Star Wars movies
    seen_all=inpf.dropna(subset=list(inpf.columns[3:9]))

    # create a list of the rank columns
    rank_list=['rank_' + ep for ep in eps]

    # create a dictionary with the count of rank 1 for each movie
    rank_dict={}
    for i in rank_list:
        rank_dict[i]=seen_all[i].value_counts()['1']

    # create a dataframe from the dictionary
    best_sw=pd.DataFrame(rank_dict.values(),index=rank_dict.keys())

    # assign a column name to the column in the dataframe
    best_sw.columns=["Count of Rank 1"]

    # create a percentage column
    best_sw["Percentage"]=best_sw["Count of Rank 1"]/len(seen_all)

    # create a column for the movie names
    best_sw["Name"]=names.values()

    # create a bar chart
    bars=gen_percent_vis(best_sw)

    # change to fivethirtyeight theme
    alt.renderers.set_embed_options(theme="fivethirtyeight")

    text = bars.mark_text(
        align='left',
        baseline='middle',
        dx=3, # Nudges text to right so it doesn't appear on top of the bar
        fontStyle="bold"
    ).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 + bars).configure_mark(
        # we don't love the blue
        color='#008fd5'
    ).configure_view(
        # we don't want a stroke around the bars
        strokeWidth=0
    ).configure_scale(

```

```

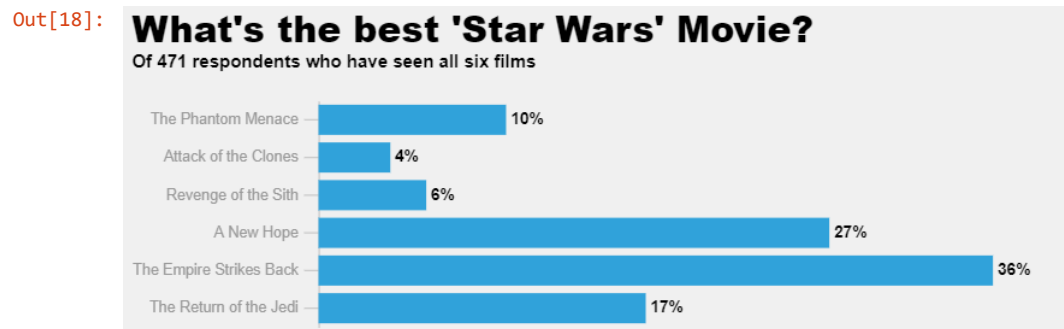
    # add some padding
    bandPaddingInner=0.2
).properties(
    # set the dimensions of the visualization
    width=500,
    height=150
).properties(
    # add a title
    title={
        "text":"What's the best 'Star Wars' Movie?",
        "fontWeight":900,
        "subtitle":"Of 471 respondents who have seen all six films",
        "subtitleFontStyle":"bold",
    }
)

return(seen_movies)

# raise NotImplementedError()

```

In [18]: `# test your code`  
`gen_best_vis(sw,episodes)`



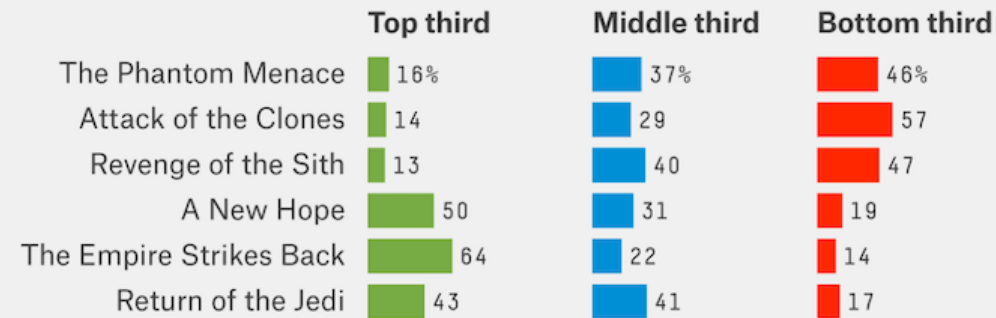
## Make sure to *style* your visualization to match the original the best you can

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:

## 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)



FIVETHIRTYEIGHT

SOURCE: SURVEYMONKEY AUDIENCE

\*\* 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 [19]: # Recreate this image using altair here (10 POINTS)

```
def gen_rate_vis(inp,eps):
    # input: inp, the star wars dataset
    # input: eps, the list of episodes

    # YOUR CODE HERE

    # change to fivethirtyeight theme
    alt.renderers.set_embed_options(theme="fivethirtyeight")

    # subset for 471 respondents who have seen all of the Star Wars movies
    seen_all=inp.dropna(subset=list(inp.columns[3:9]))

    # create a list of the rank columns
    rank_list=['rank_' + ep for ep in eps]

    # create a dictionary with the count of each rank for each movie
    rank_1_dict={}
    rank_2_dict={}
    rank_3_dict={}
    rank_4_dict={}
    rank_5_dict={}
    rank_6_dict={}

    for i in rank_list:
        rank_1_dict[i]=seen_all[i].value_counts()['1']
        rank_2_dict[i]=seen_all[i].value_counts()['2']
        rank_3_dict[i]=seen_all[i].value_counts()['3']
        rank_4_dict[i]=seen_all[i].value_counts()['4']
        rank_5_dict[i]=seen_all[i].value_counts()['5']
        rank_6_dict[i]=seen_all[i].value_counts()['6']

    # create a dataframe from the dictionary and create a category column (Top: 1 and 2, Mid: 3 and 4, Bot: 5 and 6)
    rank_1_df=pd.DataFrame(rank_1_dict.values(),index=rank_1_dict.keys())
    rank_1_df["Category"]="Top third"
    rank_2_df=pd.DataFrame(rank_2_dict.values(),index=rank_2_dict.keys())
    rank_2_df["Category"]="Top third"
    rank_3_df=pd.DataFrame(rank_3_dict.values(),index=rank_3_dict.keys())
    rank_3_df["Category"]="Middle third"
    rank_4_df=pd.DataFrame(rank_4_dict.values(),index=rank_4_dict.keys())
    rank_4_df["Category"]="Middle third"
    rank_5_df=pd.DataFrame(rank_5_dict.values(),index=rank_5_dict.keys())
    rank_5_df["Category"]="Bottom third"
    rank_6_df=pd.DataFrame(rank_6_dict.values(),index=rank_6_dict.keys())
    rank_6_df["Category"]="Bottom third"

    # assign column names to the columns in the dataframe
    rank_1_df.columns=["Count", "Category"]
    rank_2_df.columns=["Count", "Category"]
    rank_3_df.columns=["Count", "Category"]
    rank_4_df.columns=["Count", "Category"]
    rank_5_df.columns=["Count", "Category"]
    rank_6_df.columns=["Count", "Category"]
```

```

# merge rank 1 and 2 dataframes, add the counts, calculate percentage and add movie name for Top third dataframe
top_third=pd.merge(rank_1_df, rank_2_df, left_index=True, right_index=True)
top_third["Count of Top third"]=top_third["Count_x"]+top_third["Count_y"]
top_third["Percentage"]=top_third["Count of Top third"]/len(seen_all)
top_third["Name"]=names.values()

# merge rank 3 and 4 dataframes, add the counts, calculate percentage and add movie name for Middle third dataframe
mid_third=pd.merge(rank_3_df, rank_4_df, left_index=True, right_index=True)
mid_third["Count of Middle third"]=mid_third["Count_x"]+mid_third["Count_y"]
mid_third["Percentage"]=mid_third["Count of Middle third"]/len(seen_all)
mid_third["Name"]=names.values()

# merge rank 5 and 6 dataframes, add the counts, calculate percentage and add movie name for Bottom third dataframe
bot_third=pd.merge(rank_5_df, rank_6_df, left_index=True, right_index=True)
bot_third["Count of Bottom third"]=bot_third["Count_x"]+bot_third["Count_y"]
bot_third["Percentage"]=bot_third["Count of Bottom third"]/len(seen_all)
bot_third["Name"]=names.values()

# create green base bars for Top third
base_top = alt.Chart(top_third).mark_bar(size=20, color="#4da74a").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_l
        'Name:N',
        axis=alt.Axis(tickCount=5, title=''),
        # we give the sorting order to avoid alphabetical order
        sort=names_l
    )
).properties(
    # add a title
    title="Top third")

# add percent text
text_top = base_top.mark_text(
    align='left',
    baseline='middle',
    dx=3, # Nudges text to right so it doesn't appear on top of the bar
    fontStyle="bold",
    color="#4da74a"
).encode(
    # we'll use the percentage as the text
    text=alt.Text('Percentage:Q', format='.0%')
)

# store Top third vis
top_third_chart=(text_top+base_top).properties(width=80,height=150)

# create blue base bars for Middle third
base_mid = alt.Chart(mid_third).mark_bar(size=20, color="#008fd5").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_L
        'Name:N',
        # axis=alt.Axis(tickCount=5, title=''),
        axis=None,
        # we give the sorting order to avoid alphabetical order
        sort=names_L
    )
).properties(
    # add a title
    title="Middle third")

# add percent text
text_mid = base_mid.mark_text(
    align='left',
    baseline='middle',
    dx=3, # Nudges text to right so it doesn't appear on top of the bar
    fontStyle="bold",
    color="#008fd5"
).encode(
    # we'll use the percentage as the text
    text=alt.Text('Percentage:Q', format='.0%')
)

# store Middle third vis
mid_third_chart=(text_mid+base_mid).properties(width=50,height=150)

# create red base bars for Bottom third
base_bot = alt.Chart(bot_third).mark_bar(size=20, color="#ff2600").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_L
        'Name:N',
        # axis=alt.Axis(tickCount=5, title=''),
        axis=None,
        # we give the sorting order to avoid alphabetical order
        sort=names_L
    )
).properties(
    # add a title
    title="Bottom third")

# add percent text
text_bot = base_bot.mark_text(
    align='left',
    baseline='middle',
    dx=3, # Nudges text to right so it doesn't appear on top of the bar
    fontStyle="bold",
    color="#ff2600"
).encode(
    # we'll use the percentage as the text
    text=alt.Text('Percentage:Q', format='.0%')
)

```

```

# store Bottom third vis
bot_third_chart=(text_bot+base_bot).properties(width=70,height=150)

# create horizontally concatenated chart with all three vis
hcon=alt.HConcatChart(hconcat=[top_third_chart, mid_third_chart, bot_third_chart]
                      ).configure_view(strokeWidth=0
                      ).configure_title(fontSize=10, anchor="middle", offset=10)

hcon=hcon.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)"],
    "fontWeight":900,
    "fontSize":24,
    "dy":-10,
    "anchor":"start",
    "subtitleFontStyle":"bold"
})

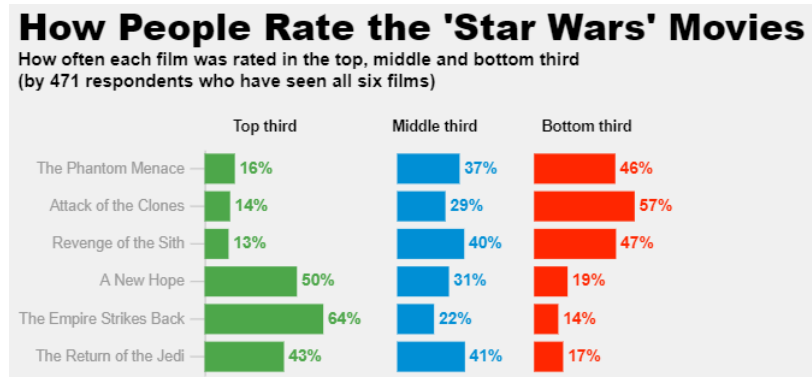
return hcon

# raise NotImplementedError()

```

In [20]: `# Let's check our solution`  
`gen_rate_vis(sw,episodes)`

Out[20]:

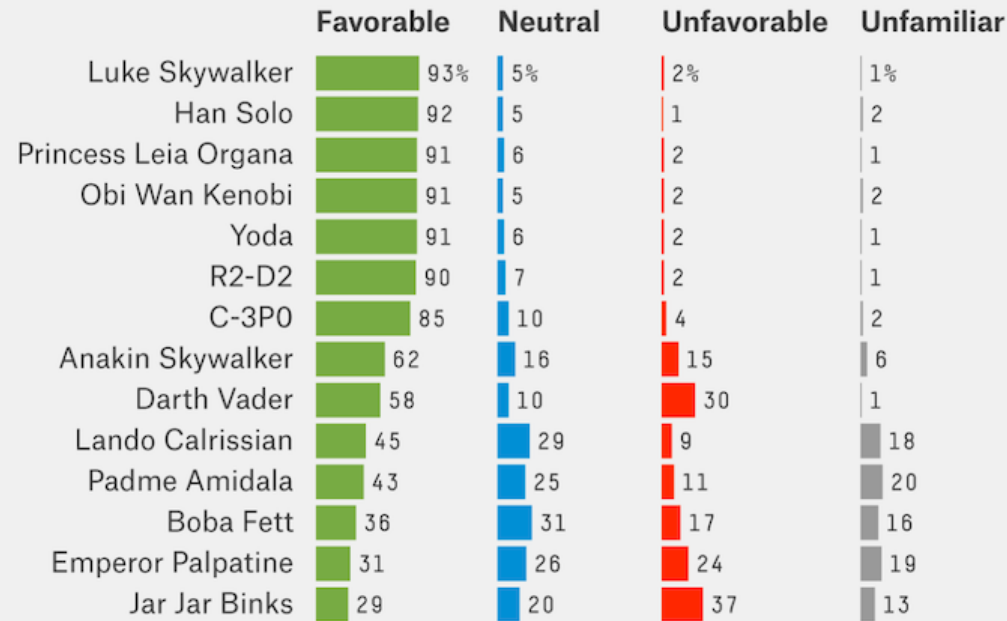


Finally, we took a boilerplate format used by political favorability polls — “Please state whether you view the following characters favorably, unfavorably, or are unfamiliar with him/her” — and asked respondents to rate characters in the series.



## 'Star Wars' Character Favorability Ratings

By 834 respondents



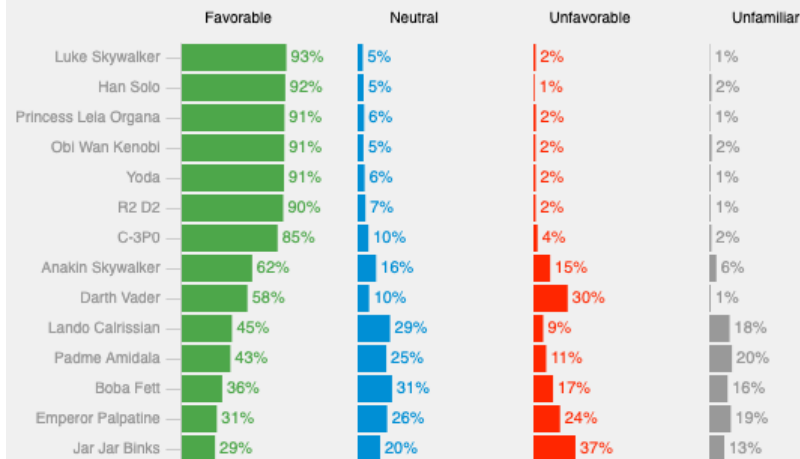
FIVETHIRTYEIGHT

SOURCE: SURVEYMONKEY AUDIENCE

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

## 'Star Wars' Characters Favorability Ratings

By 826 respondents



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

In [21]: *# Recreate this image using altair here (10 POINTS)*

```
def gen_favor_vis(inp):  
    # input: inpf, the star wars dataset  
  
    # change to fivethirtyeight theme  
    alt.renderers.set_embed_options(theme="fivethirtyeight")  
  
    # create a list of the character rating columns  
    char_list=inpf.columns[15:29]  
  
    # create a dictionary with the count of each rating for each character  
    favorable_dict={}  
    neutral_dict={}  
    unfavorable_dict={}  
    unfamiliar_dict={}  
  
    for i in char_list:  
        favorable_dict[i]=sw[i].value_counts()["Very favorably"]+sw[i].value_counts()["Somewhat favorably"]  
        neutral_dict[i]=sw[i].value_counts()["Neither favorably nor unfavorably (neutral)"]  
        unfavorable_dict[i]=sw[i].value_counts()["Very unfavorably"]+sw[i].value_counts()["Somewhat unfavorably"]  
        unfamiliar_dict[i]=sw[i].value_counts()["Unfamiliar (N/A)"]  
  
    # create a dataframe from the dictionary and create a category column  
    favorable=pd.DataFrame(favorable_dict.values(), index=favorable_dict.keys())  
    favorable["Category"]="Favorable"  
    neutral=pd.DataFrame(neutral_dict.values(), index=neutral_dict.keys())  
    neutral["Category"]="Neutral"  
    unfavorable=pd.DataFrame(unfavorable_dict.values(), index=unfavorable_dict.keys())  
    unfavorable["Category"]="Unfavorable"  
    unfamiliar=pd.DataFrame(unfamiliar_dict.values(), index=unfamiliar_dict.keys())  
    unfamiliar["Category"]="Unfamiliar"  
  
    # assign column names to the columns in the dataframe  
    favorable.columns=["Count", "Category"]  
    neutral.columns=["Count", "Category"]  
    unfavorable.columns=["Count", "Category"]  
    unfamiliar.columns=["Count", "Category"]  
  
    # calculate percentage for each of the ratings for each character  
    for i in favorable.index:  
        favorable["Percentage"]=favorable["Count"]/inpf.dropna(subset=[i]).shape[0]  
    for i in neutral.index:  
        neutral["Percentage"]=neutral["Count"]/inpf.dropna(subset=[i]).shape[0]  
    for i in unfavorable.index:  
        unfavorable["Percentage"]=unfavorable["Count"]/inpf.dropna(subset=[i]).shape[0]  
    for i in unfamiliar.index:  
        unfamiliar["Percentage"]=unfamiliar["Count"]/inpf.dropna(subset=[i]).shape[0]  
  
    # reset index to get the character names as a column in each of the dataframes  
    favorable.reset_index(inplace=True)  
    favorable.columns.values[0]="Character Name"  
    neutral.reset_index(inplace=True)  
    neutral.columns.values[0]="Character Name"
```

```

unfavorable.reset_index(inplace=True)
unfavorable.columns.values[0]="Character Name"
unfamiliar.reset_index(inplace=True)
unfamiliar.columns.values[0]="Character Name"

# create a character sorting order for the final chart
char_sorting_order=["Luke Skywalker", "Han Solo", "Princess Leia Organa", "Obi Wan Kenobi", "Yoda", "R2 D2", "C-3P0",
                    "Anakin Skywalker", "Darth Vader", "Lando Calrissian", "Padme Amidala", "Boba Fett", "Emperor Palpatine",
                    "Jar Jar Binks"]

# create green base bars for Favorable
base_fav = alt.Chart(favorable).mark_bar(size=20, color="#4da74a").encode(
    # encode x as the percent, and hide the axis
    x=alt.X(
        'Percentage',
        axis=None),
    y=alt.Y(
        # encode y using the character name, use the character name to label the axis, sort using the char_sorting_order
        'Character Name:N',
        axis=alt.Axis(tickCount=5, title=''),
        # we give the sorting order to avoid alphabetical order
        sort=char_sorting_order
    )
).properties(
    # add a title
    title="Favorable")

# add percent text
text_fav = base_fav.mark_text(
    align='left',
    baseline='middle',
    dx=3, # Nudges text to right so it doesn't appear on top of the bar
    fontStyle="bold",
    color="#4da74a"
).encode(
    # we'll use the percentage as the text
    text=alt.Text('Percentage:Q', format='.0%')
)

# store Favorable vis
favorable_chart=(text_fav+base_fav).properties(width=80,height=300)

# create blue base bars for Neutral
base_neu = alt.Chart(neutral).mark_bar(size=20, color="#008fd5").encode(
    # encode x as the percent, and hide the axis
    x=alt.X(
        'Percentage',
        axis=None),
    y=alt.Y(
        # encode y using the character name, use the character name to label the axis, sort using the char_sorting_order
        'Character Name:N',
        # axis=alt.Axis(tickCount=5, title=''),
        axis=None,
        # we give the sorting order to avoid alphabetical order
        sort=char_sorting_order
    )
)

```

```

).properties(
  # add a title
  title="Neutral")

# add percent text
text_neu = base_neu.mark_text(
  align='left',
  baseline='middle',
  dx=3, # Nudges text to right so it doesn't appear on top of the bar
  fontStyle="bold",
  color="#008fd5")
).encode(
  # we'll use the percentage as the text
  text=alt.Text('Percentage:Q',format='.0%')
)

# store Neutral vis
neutral_chart=(text_neu+base_neu).properties(width=30,height=300)

# create red base bars for Unfavorable
base_unf = alt.Chart(unfavorable).mark_bar(size=20, color="#ff2600").encode(
  # encode x as the percent, and hide the axis
  x=alt.X(
    'Percentage',
    axis=None),
  y=alt.Y(
    # encode y using the character name, use the character name to label the axis, sort using the char_sorting_order
    'Character Name:N',
    # axis=alt.Axis(tickCount=5, title=''),
    axis=None,
    # we give the sorting order to avoid alphabetical order
    sort=char_sorting_order
  )
).properties(
  # add a title
  title="Unfavorable")

# add percent text
text_unf = base_unf.mark_text(
  align='left',
  baseline='middle',
  dx=3, # Nudges text to right so it doesn't appear on top of the bar
  fontStyle="bold",
  color="#ff2600")
).encode(
  # we'll use the percentage as the text
  text=alt.Text('Percentage:Q',format='.0%')
)

# store Unfavorable vis
unfavorable_chart=(text_unf+base_unf).properties(width=30,height=300)

# create red base bars for Unfamiliar
base_unfam = alt.Chart(unfamiliar).mark_bar(size=20, color="#999999").encode(
  # encode x as the percent, and hide the axis
  x=alt.X(

```

```

        'Percentage',
        axis=None),
    y=alt.Y(
        # encode y using the character name, use the character name to label the axis, sort using the char_sorting_order
        'Character Name:N',
        # axis=alt.Axis(tickCount=5, title=''),
        axis=None,
        # we give the sorting order to avoid alphabetical order
        sort=char_sorting_order
    )
).properties(
    # add a title
    title={
        "text": "Unfamiliar",
        "align": "center"
    }
)

# add percent text
text_unfam = base_unfam.mark_text(
    align='left',
    baseline='middle',
    dx=3, # Nudges text to right so it doesn't appear on top of the bar
    fontStyle="bold",
    color="#999999"
).encode(
    # we'll use the percentage as the text
    text=alt.Text('Percentage:Q', format='.0%')
)

# store Unfamiliar vis
unfamiliar_chart=(text_unfam+base_unfam).properties(width=18,height=300)

# create horizontally concatenated chart with all four vis
hcon=alt.HConcatChart(hconcat=[favorable_chart, neutral_chart, unfavorable_chart, unfamiliar_chart],
    spacing=40
    ).configure_view(strokeWidth=0
    ).configure_title(fontSize=10, anchor="middle", offset=10, dx=25)

hcon=hcon.properties(title={
    "text": "'Star Wars' Characters Favorability Ratings",
    "subtitle": "By 826 respondents",
    "fontWeight": 900,
    "fontSize": 24,
    "dy": -10,
    "dx": -10,
    "anchor": "start",
    "subtitleFontStyle": "bold"
})

return hcon

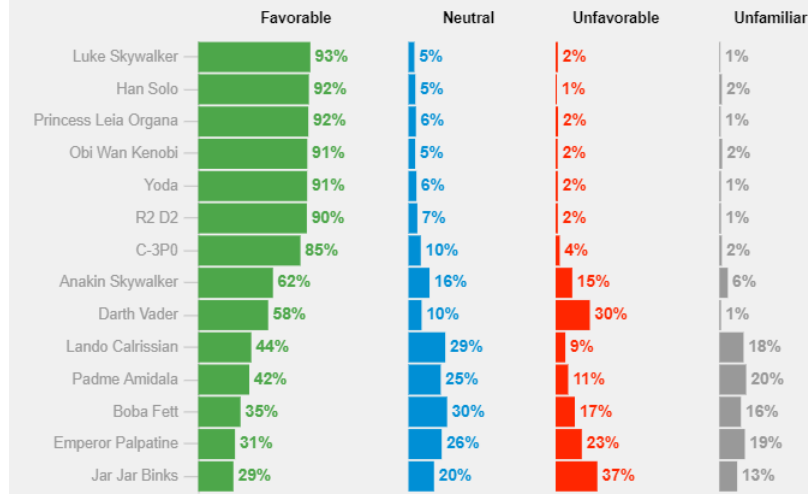
# YOUR CODE HERE
# raise NotImplementedError()

```

```
In [22]: # Let's test the solution
gen_favor_vis(sw)
```

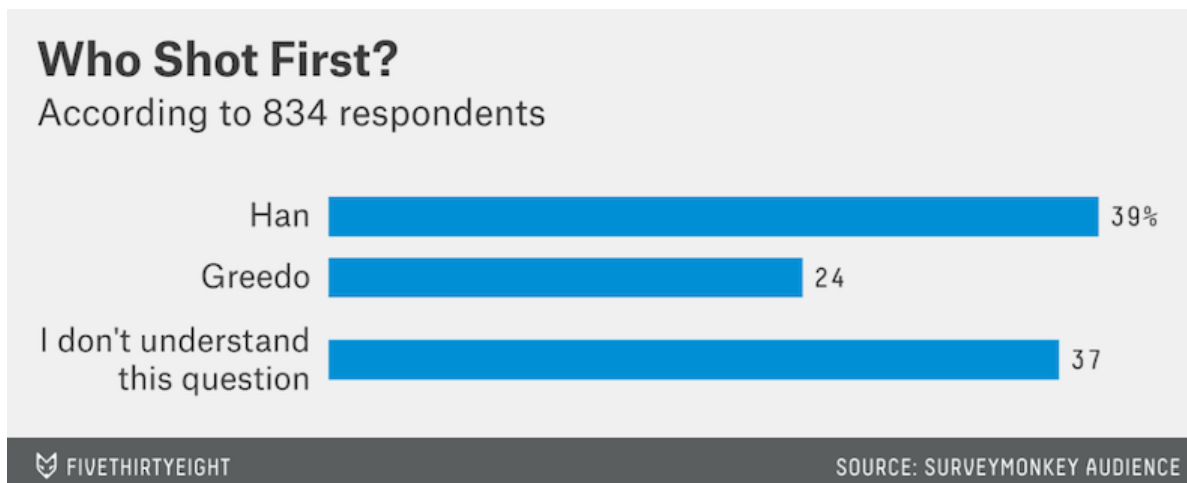
Out[22]: **'Star Wars' Characters Favorability Ratings**

By 826 respondents



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](http://en.wikipedia.org/wiki/Han_shot_first)) can have on a society:



\*\* Homework note: Click [here](#) ([assets/shot\\_first.png](#)) 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 [23]: *# Recreate this image using altair here (10 POINTS)*

```
def gen_first_shot_vis(inp):  
    # input: inpf, the star wars dataset  
    # YOUR CODE HERE  
  
    # create a dataframe with the count of each answer  
    shot=pd.DataFrame(inp["Which character shot first?"].value_counts())  
  
    # create a percentage column for the count of each answer  
    shot["Percentage"]=shot["Which character shot first?"]/inp.dropna(subset=["Which character shot first?"]).shape[0]  
  
    # reset index to get the answers as a column in the dataframe  
    shot.reset_index(inplace=True)  
    shot.columns.values[0]="Answer"  
  
    # create an answer sorting order for the final chart  
    answer_sort_order=["Han", "Greedo", "I don't understand this question"]  
  
    # create blue base bars  
    base = alt.Chart(shot).mark_bar(size=20, color="#008fd5").encode(  
        # encode x as the percent, and hide the axis  
        x=alt.X(  
            'Percentage',  
            axis=None,  
        ),  
        y=alt.Y(  
            # encode y using the Answer column, use the Answer column to label the axis, sort using the answer_sort_order  
            'Answer:N',  
            axis=alt.Axis(tickCount=5, title=''),  
            # we give the sorting order to avoid alphabetical order  
            sort=answer_sort_order  
        )  
    ).properties(  
        # add a title  
        title={  
            "text":"Who shot first?",  
            "fontWeight":900,  
            "subtitle":"According to 828 respondents",  
            "subtitleFontStyle":"bold"  
        }  
    )  
  
    # add percent text  
    text = base.mark_text(  
        align='left',  
        baseline='middle',  
        dx=3, # Nudges text to right so it doesn't appear on top of the bar  
        fontStyle="bold",  
        color="black"  
    ).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
```

```

chart = (text + base).configure_mark(
    # we don't love the blue
    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=100)

return chart

# raise NotImplementedError()

```

In [24]: `# 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.

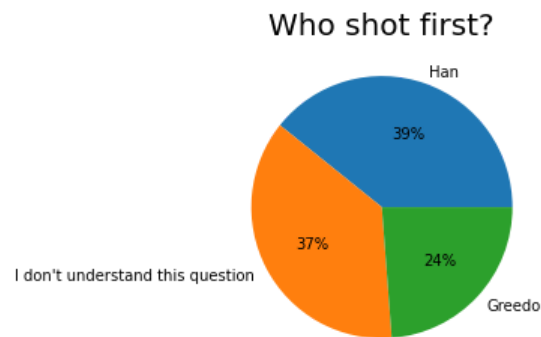
```
In [26]: # add your code here
import matplotlib.pyplot as plt

# create a dataframe with the count of each answer
shot=pd.DataFrame(sw["Which character shot first?"].value_counts())

# create a percentage column for the count of each answer
shot["Percentage"]=shot["Which character shot first?"]/sw.dropna(subset=["Which character shot first?"]).shape[0]

# reset index to get the answers as a column in the dataframe
shot.reset_index(inplace=True)
shot.columns.values[0]="Answer"

# create pie chart
plt.pie(shot["Percentage"], labels=shot["Answer"], autopct='%0f%%')
plt.title("Who shot first?", fontsize=20)
plt.show()
```



*Provide your justification here:*

The alternative visualization I have proposed is a pie chart for the "Who shot first?" question from 2.4. I have used Matplotlib for this visualization instead of Altair since `mark_arc()` is not supported by the current version of Altair.

Since this is a part-versus-whole question, I would argue that my visualization is probably more effective than the one in the article (the bar chart) as pie charts are better for comparing parts of a whole. While bar charts enable easy comparison due to bar lengths, in this case I would prefer pie charts since it becomes immediately obvious that all slices of the pie add up to a 100% and are all part of the same set, which makes intuitive sense to compare the slice sizes and gain insights.

## 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 [28]: # add your code here

# get the count of fans of the Star Wars franchise grouped by gender
gen_fan=seen_at_least_one.groupby(["fan", "Gender"])["Gender"].count()
gen_fan=pd.DataFrame(gen_fan)

# create a copy of the dataframe for modifications
calc=gen_fan.copy()

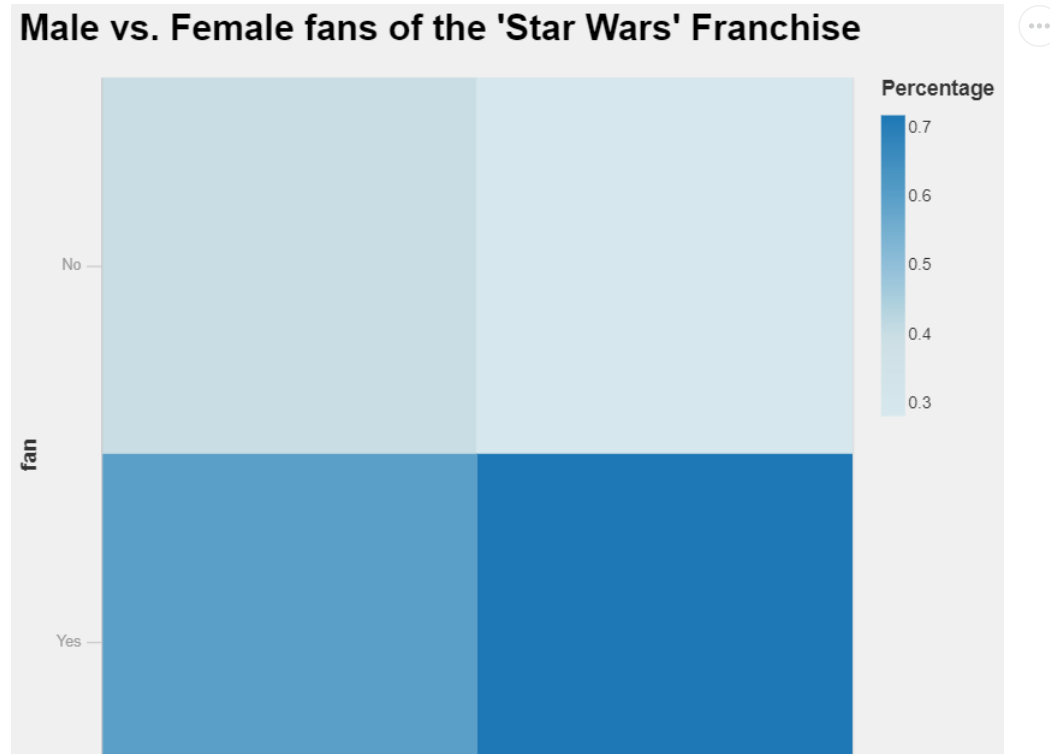
# calculate percentages
calc.loc["No", "Female"]=calc.loc["No", "Female"]/seen_at_least_one.Gender.value_counts()["Female"]
calc.loc["No", "Male"]=calc.loc["No", "Male"]/seen_at_least_one.Gender.value_counts()["Male"]
calc.loc["Yes", "Female"]=calc.loc["Yes", "Female"]/seen_at_least_one.Gender.value_counts()["Female"]
calc.loc["Yes", "Male"]=calc.loc["Yes", "Male"]/seen_at_least_one.Gender.value_counts()["Male"]
calc.columns=["Percentage"]

# reset index to remove hierarchical indices
calc.reset_index(inplace=True)

# create heatmap
alt.Chart(calc).mark_rect().encode(
    x='Gender',
    y='fan',
    color='Percentage'
).properties(
    width=500,
    height=500,
    title="Male vs. Female fans of the 'Star Wars' Franchise")

```

Out[28]:



*Provide your justification here:*

The complementary visualization I have proposed is a heatmap for the percentage of male versus female fans of the 'Star Wars' movies.

The article had the following information: "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." Since visualization for this particular information was missing in the article, I thought it would be a good idea to create one.

I feel like this is an effective way to visualize this information since the intensity of the color would make it immediately apparent that men seem to be more avid fans of the franchise than women. The heatmap shows that even among male and female fans, the percentage of male fans far exceeds the percentage of female fans of the franchise. Among the men and women who aren't fans of the franchise, the percentage of men is lesser than the percentage of women. The color bar also makes it easy to relate to which block contains which percentage (or a ballpark), at least enough to make comparisons if not be able to make out the precise percentage value.