

liujr

February 14, 2025

## 1 SI649 W25 Altair Theme Homework #3

### 2 Overview

We are focusing on **custom themes & small multiples** in this lab! For this assignment, we will be looking at *Star Wars* character dataset from [America's Favorite 'Star Wars' Movies \(And Least Favorite Characters\)](#) by Walt Hickey.

#### 2.0.1 Lab Instructions

- Save, rename, and submit the ipynb file (use your username in the name).
- Complete all the checkpoints, to create the required visualization at each cell
- Run every cell (do Runtime -> Restart and run all to make sure you have a clean working version), and upload your .ipynb file to Canvas.
- For each visualization, there is a space to write down a “Grammar of Graphics” plan, but this is optional for this assignment.
- If you end up stuck, show us your work by including links (URLs) that you have searched for. You'll get partial credit for showing your work in progress.

```
[1]: # suppress warnings about future deprecations
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)

# imports we will use
import altair as alt
import pandas as pd

# for large data sets
alt.data_transformers.disable_max_rows()

# read in data
df = pd.read_csv('https://scmcqueen.github.io/si649_hw/skyelerStarWars.
↳ csv', header=[0, 1], skip_blank_lines=True)
```

#### 2.1 Part 1: Character Rating Facet Charts

Using an Altair Facet chart, recreate the ‘Star Wars’ Character Favorability Ratings chart from the [Star Wars article](#). It should look like this:

### 2.1.1 Step 1: Get data in the correct format

We will get this started for you

```
[2]: # Read in the data with Character name & review
download = pd.read_csv('https://scmcqueen.github.io/si649_hw/reviews_by_char.
↳csv',index_col=0)
download.head()
```

```
[2]:  character      rating
0  Han Solo    Very favorably
2  Han Solo  Somewhat favorably
3  Han Solo    Very favorably
4  Han Solo    Very favorably
5  Han Solo    Very favorably
```

```
[3]: # TODO: Map the 'rating' column to a new column that matches the chart's
↳Favorable, Unfavorable, etc.
download['class'] = download['rating'].map({'Very favorably':
↳'Favorable','Somewhat favorably':'Favorable','Unfamiliar (N/A)':
↳'Unfamiliar', 'Somewhat unfavorably':'Unfavorable','Neither favorably nor
↳unfavorably (neutral)':'Neutral', 'Very unfavorably':'Unfavorable'})
```

```
[4]: # TODO: in Pandas, get the percent of each rating per character
rating_counts = download.groupby(['character', 'class']).size().
↳reset_index(name='count')
total_counts = download.groupby(['character']).size().reset_index(name='total')
df_percent = rating_counts.merge(total_counts, on='character')
df_percent['percent'] = df_percent['count'] / df_percent['total'] * 100
favorable_sort = df_percent[df_percent['class'] == 'Favorable']
df_percent = df_percent.merge(
    favorable_sort[['character', 'percent']],
    on='character',
    suffixes=('', '_favorable')
)
```

### 2.1.2 Step 2: Create your charts

Hints: \* Layer the charts before faceting!

```
[5]: bar1 = alt.Chart(df_percent).mark_bar().encode(
    x=alt.X('percent:Q', axis=alt.Axis(title=None, grid=False, labels=False,
↳ticks=False, domain=False)),
    y=alt.Y('character:N', axis=alt.Axis(title=None, grid=False),sort=alt.
↳EncodingSortField(field="percent_favorable", order="descending")),
    color=alt.Color('class:N', legend=None),
).properties(
    width=80,
```

```

)
text1 = bar1.mark_text(
    align='left',
    baseline='middle',
    dx=3,
    fontSize=10,
).encode(
    text=alt.Text('percent:Q', format=".0f")
)
(bar1 + text1).facet(
    column=alt.Column('class:N', sort=['Favorable', 'Neutral', 'Unfavorable', 'Unfamiliar'], header=alt.Header(title=None)),
).configure_view(
    stroke=None
).properties(
    title={
        "text": "'Star Wars' Favorability Rating",
        "subtitle": "By 834 respondents",
        "anchor": "start"
    }
)

```

[5]: alt.FacetChart(...)

[ ]:

## 2.2 Part 2: Star Wars Theme

### 2.2.1 Step 1: Create Star Wars Theme

We want you to try implementing a custom theme in Altair based on this style guide:

We will give you some starter code, so you aren't creating a theme fully from scratch. The Altair documentation on [Chart Customization](#) and this Towards Data Science Article [Consistently Beautiful Visualizations with Altair Themes](#) should serve as helpful guides.

You can set the spacing, color palettes, font schemes, etc.

Run the cell below to get the font you need.

```

[6]: %%html

<style>
@import url('https://fonts.googleapis.com/css?family=Lato');
</style>

```

<IPython.core.display.HTML object>

```
[7]: # TO DO: Modify this code to fit the above style guide
# @alt.theme.register("star_wars", enable=True) # Comment this line out for
↳ Altair 5.2
# the theme is defined as a function
def star_wars_solution():
    font = "Lato"
    backgroundColor = None

    return {
        "config": {
            "title": {
                "anchor": "start",
                "fontSize": 20,
                "font": font,
                "color": "#000000"
            },
            "subtitle": {
                "fontSize": 18,
                "font": font,
                "color": "#000000"
            },
            "header": {
                "titleColor": "#00274C",
                "titleFontSize": 1,
                "titleFontWeight": "bold",
                "titleBaseline": "top"
            },
            "axisX": {
                "titleFont": font,
                "titleFontSize": 16,
                "titleColor": "#000000",
                "labelFont": font,
                "labelFontSize": 14,
                "labelColor": "#000000",
                "domain": True,
                "grid": False,
                "ticks": False,
            },
            "axisY": {
                "titleFont": font,
                "titleFontSize": 16,
                "titleColor": "#000000",
                "titleAngle": 0,
                "titleAlign": "left",
                "titleAnchor": "start",
                "titleY": -10,
                "labelFont": font,
```

```

        "labelFontSize": 14,
        "labelColor": "#000000",
        "domain": False,
        "grid": True,
        "gridColor": "#aab0c0",
        "gridWidth": 1
    },
    "legend": {
        "labelFont": font,
        "labelFontSize": 14,
        "titleFont": font,
        "titleFontSize": 16,
    },
    "background": backgroundColor,
    "view": {
        "stroke": "transparent",
    },
    "range": {
        "category": ["#B62321", "#A1A332", "#15509F", "#3B444B", ↵
↵ "#D7C078", "#DC5026"],
        "ordinal": ["#B62321", "#BD4232", "#C36244", "#CA8155", ↵
↵ "#D0A167", "#D7C078"],
        "ramp": ["#BD4232", "#C36244", "#CA8155", "#D0A167", "#D7C078"],
        "diverging": ["#15509F", "#638ABF", "#B1C5DF", "#FFFFFF", ↵
↵ "#F3C5B7", "#E88A6E", "#DC5026"]
    },
    "text": {
        "font": font,
        "fontSize": 14,
        "color": "match"
    },
    "note": {
        "font": font,
        "fontSize": 12,
        "color": "#000000",
        "align": "left"
    },
    "source": {
        "font": font,
        "fontSize": 12,
        "color": "#000000",
        "align": "right"
    },
    },
}
}

```

*# For Altair version 5.2, uncomment the lines below*

```
alt.themes.register('star_wars', star_wars_solution)
# # enable the newly registered theme
alt.themes.enable('star_wars')
```

```
[7]: ThemeRegistry.enable('star_wars')
```

### 2.2.2 Step 2: Recreate Favorability Chart with new theme

To check that your theme looks correct, recreate your chart of Star Wars characters favorability. If your theme is enabled correctly, the visualization should look like this:

```
[8]: bar1 = alt.Chart(df_percent).mark_bar().encode(
    x=alt.X('percent:Q', axis=alt.Axis(title=None, grid=False, labels=False,
    ↪ticks=False, domain=False)),
    y=alt.Y('character:N', axis=alt.Axis(title=None, grid=False), sort=alt.
    ↪EncodingSortField(field="percent_favorable", order="descending")),
    color=alt.Color('class:N', legend=None),
).properties(
    width=80,
)
text1 = bar1.mark_text(
    align='left',
    baseline='middle',
    dx=3,
    fontSize=10,
).encode(
    text=alt.Text('percent:Q', format=".0f")
)
(bar1 + text1).facet(
    column=alt.Column('class:N', sort=['Favorable', 'Neutral', 'Unfavorable',
    ↪'Unfamiliar'], header=alt.Header(title=None)),
).configure_view(
    stroke=None
).properties(
    title={
        "text": "'Star Wars' Favorability Rating",
        "subtitle": "By 834 respondents",
        "anchor": "start"
    }
)
```

```
[8]: alt.FacetChart(...)
```

### 2.2.3 Step 3: Recreate the following charts to test your theme

#### Chart 1: Favorable Ratings by Number of Ratings

```
[9]: # TODO: Create favorability chart
df_c1 = df_percent[df_percent['class'] == 'Favorable']
df_c1['c1'] = df_c1['percent']/100
```

/var/folders/gp/bnf8n57s1nlgccs8xl1kk3mc0000gn/T/ipykernel\_95052/999980883.py:3:  
 SettingWithCopyWarning:  
 A value is trying to be set on a copy of a slice from a DataFrame.  
 Try using .loc[row\_indexer,col\_indexer] = value instead

See the caveats in the documentation: [https://pandas.pydata.org/pandas-docs/stable/user\\_guide/indexing.html#returning-a-view-versus-a-copy](https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)  
 df\_c1['c1'] = df\_c1['percent']/100

```
[10]: alt.Chart(df_c1).mark_point(filled=True, size=50).encode(
    x=alt.X('total:Q', scale=alt.Scale(domain=[800, 850])), axis=alt.
    ↪Axis(title='Number of Ratings')),
    y=alt.Y('c1:Q', axis=alt.Axis(title='Percent of Favorable Ratings',
    ↪tickCount=6)),
    color=alt.Color('c1:Q', title='Percent'),
).properties(
    title='Favorable Ratings by Number of Ratings',
    width=300,
    height=300
)
```

```
[10]: alt.Chart(...)
```

## Chart 2: Who shot first?

```
[11]: # read in shooting data
shot_first = pd.read_csv('https://scmcqueen.github.io/si649_hw/shot_first.
    ↪csv', index_col=0)
shot_first.head()
```

```
[11]:          shooter
0  I don't understand this question
2  I don't understand this question
3  I don't understand this question
4                      Greedo
5                      Han
```

```
[12]: df_count = shot_first['shooter'].value_counts().reset_index()
df_count.columns = ['shooter', 'count']
total = df_count['count'].sum()
df_count['percent'] = df_count['count'] / total
order_list = ['Greedo', 'Han', "I don't understand this question"]
```

```
df_count['shooter'] = pd.Categorical(df_count['shooter'],
    ↪categories=order_list, ordered=True)
df_count.sort_values('shooter', inplace=True)
```

```
[13]: bar = (
    alt.Chart(df_count)
    .mark_bar(height=30)
    .encode(
        y=alt.Y('shooter:N', sort=None, title='', axis=alt.Axis(grid=False,
    ↪ticks=False)),
        x=alt.X('percent:Q',
            title='',
            axis=alt.Axis(title=None, grid=False, labels=False,
    ↪ticks=False, domain=False)),
        color=alt.Color('percent:Q', title='Percent of Responders',
            scale=alt.Scale(domain=[0.2, 0.3, 0.4],
    ↪range=["#15509F", "#FFFFFF", "#DC5026"] ))
    ).properties(
        title='Who shot first?',
        width=300,
        height=100
    )
)
text = bar.mark_text(
    align='right',
    baseline='middle',
    dx=-3,
    color='white'
).encode(
    text=alt.Text('percent:Q', format='.2%'),
    color=alt.value('white')
)

final_chart = (bar + text)
final_chart
```

```
[13]: alt.LayerChart(...)
```

## 2.3 Part 3: Repeat Chart of Movies Seen by Key Demographics

Recreate this chart using `.repeat()`

```
[14]: # read in the formatted data
x = pd.read_csv('https://scmcqueen.github.io/si649_hw/movies_demo.
    ↪csv', index_col=0)
x.head()
```



```
[14]:      Age Gender \
2  18-29   Male
3  18-29   Male
4  18-29   Male
5  18-29   Male
8  18-29   Male

      Do you consider yourself to be a fan of the Star Wars film franchise? \
2                                                    No
3                                                    Yes
4                                                    Yes
5                                                    Yes
8                                                    Yes

      movies_seen
2           3
3           6
4           6
5           6
8           6
```

```
[15]: alt.Chart(x).mark_boxplot(extent='min-max', box=alt.MarkConfig(fill="#b62321",
↪stroke="#b62321")).encode(
    x=alt.X(alt.repeat("column"), type='nominal', axis=alt.
↪Axis(labelAngle=0,ticks=True)),
    y=alt.Y("movies_seen:Q", title=None,
        scale=alt.Scale(domain=[0, 6])),
).properties(
    width=200, height=300
).repeat(
    column=[
        "Age",
        "Gender",
        "Do you consider yourself to be a fan of the Star Wars film franchise?"
    ]
).properties(
    title="Number of Movies Seen by Key Demographics"
)
```

```
[15]: alt.RepeatChart(...)
```

## 2.4 Part 4: Make your own theme!

Make your own theme. This is your opportunity to be creative and define a consistent, reusable style. The theme must be significantly different than the Star Wars one above. The more unique, the better!

To get full points, you must: \* Create at least 3 color schemes for your theme \* Choose a new font

for your theme \* Change the default font sizes \* Set a new default mark color \* Choose to enable or disable grid lines \* Change the default padding of the titles

You can choose one of the themes from [this list of Data Visualization Style Guidelines](#) as a starting point. To generate interesting color palettes, you can use sites like [Coolers](#), [Canva](#), or [ColorMagic](#).

```
[16]: import altair as alt

def new_theme():
    font = "Roboto Slab"
    backgroundColor = "#F9F9F5"

    return {
        "config": {
            "title": {
                "font": font,
                "fontSize": 24,
                "color": "#1A3C40",
                "offset": 20
            },
            "subtitle": {
                "font": font,
                "fontSize": 18,
                "color": "#1A3C40",
                "offset": 15
            },
        },
        "axis": {
            "labelFont": font,
            "labelFontSize": 14,
            "labelColor": "#1A3C40",
            "titleFont": font,
            "titleFontSize": 16,
            "titleColor": "#1A3C40"
        },
        "axisX": {
            "grid": False
        },
        "axisY": {
            "grid": True,
            "gridColor": "#D2D7D3",
            "gridWidth": 1
        },
        "legend": {
            "labelFont": font,
            "labelFontSize": 14,
            "titleFont": font,
```

```

        "titleFontSize": 16,
        "titleColor": "#1A3C40"
    },
    "mark": {
        "color": "#A2323E"
    },
    "range": {
        "category": [
            "#586F6B",
            "#9DBF9E",
            "#D8DBB2",
            "#E9B872",
            "#CA7E8D",
            "#B5515D"
        ],
        "ordinal": [
            "#FCE1CE",
            "#F8B6A3",
            "#F18578",
            "#DE5858",
            "#B02339"
        ],
        "ramp": [
            "#F0E5CF",
            "#E2D19B",
            "#C3B177",
            "#988753",
            "#6B5F34"
        ],
        "diverging": [
            "#404E5C",
            "#7A9E9F",
            "#C9DBC7",
            "#F8F8F8",
            "#FCDBBE",
            "#ED9B67",
            "#A0412C"
        ]
    },
    "background": backgroundColor,
    "view": {
        "stroke": "transparent"
    }
}
}

```

```
alt.themes.register("ljr", new_theme)
```

```
alt.themes.enable("ljr")
```

```
[16]: ThemeRegistry.enable('ljr')
```

### 2.4.1 Recreate all of the charts from above in your theme!

```
[17]: # insert facet chart
bar1 = alt.Chart(df_percent).mark_bar().encode(
    x=alt.X('percent:Q', axis=alt.Axis(title=None, grid=False, labels=False,
    ↪ticks=False, domain=False)),
    y=alt.Y('character:N', axis=alt.Axis(title=None, grid=False), sort=alt.
    ↪EncodingSortField(field="percent_favorable", order="descending")),
    color=alt.Color('class:N', legend=None),
).properties(
    width=80,
)
text1 = bar1.mark_text(
    align='left',
    baseline='middle',
    dx=3,
    fontSize=10,
).encode(
    text=alt.Text('percent:Q', format=".0f")
)
theme_1 = (bar1 + text1).facet(
    column=alt.Column('class:N', sort=['Favorable', 'Neutral', 'Unfavorable',
    ↪'Unfamiliar'], header=alt.Header(title=None)),
).configure_view(
    stroke=None
).properties(
    title={
        "text": "'Star Wars' Favorability Rating",
        "subtitle": "By 834 respondents",
        "anchor": "start"
    }
)
theme_1
```

```
[17]: alt.FacetChart(...)
```

```
[18]: # insert scatter plot
alt.Chart(df_c1).mark_point(filled=True, size=50).encode(
    x=alt.X('total:Q', scale=alt.Scale(domain=[800, 850]), axis=alt.
    ↪Axis(title='Number of Ratings')),
    y=alt.Y('c1:Q', axis=alt.Axis(title='Percent of Favorable Ratings',
    ↪tickCount=6)),
```

```

        color=alt.Color('c1:Q', title='Percent'),
    ).properties(
        title='Favorable Ratings by Number of Ratings',
        width=300,
        height=300
    )

```

[18]: alt.Chart(...)

```

[19]: bar = (
    alt.Chart(df_count)
    .mark_bar(height=30)
    .encode(
        y=alt.Y('shooter:N', sort=None, title='', axis=alt.Axis(grid=False,
↪ticks=False)),
        x=alt.X('percent:Q',
            title='',
            axis=alt.Axis(title=None, grid=False, labels=False,
↪ticks=False, domain=False)),
        color=alt.Color('percent:Q', title='Percent of Responders',
            scale=alt.Scale(domain=[0.2, 0.3, 0.4] ))
    ).properties(
        title='Who shot first?',
        width=300,
        height=100
    )
)
text = bar.mark_text(
    align='right',
    baseline='middle',
    dx=-3,
    color='white'
).encode(
    text=alt.Text('percent:Q', format='.2%'),
    color=alt.value('white')
)

final_chart = (bar + text)
final_chart

```

[19]: alt.LayerChart(...)

```

[20]: alt.Chart(x).mark_boxplot(extent='min-max').encode(
    x=alt.X(alt.repeat("column"), type='nominal', axis=alt.
↪Axis(labelAngle=0,ticks=True)),
    y=alt.Y("movies_seen:Q", title=None,
        scale=alt.Scale(domain=[0, 6])),

```

```

).properties(
    width=200, height=300
).repeat(
    column=[
        "Age",
        "Gender",
        "Do you consider yourself to be a fan of the Star Wars film franchise?"
    ]
).properties(
    title="Number of Movies Seen by Key Demographics"
)

```

[20]: alt.RepeatChart(...)

## 2.5 Part 5: Upload one of your charts to a github page

Choose one of the four charts above (in your own style) and export it as an html, using `chart.save('alt_theme.html')`.

Open the html file using any text editor, and add a link to the data you used for this chart, by adding a line of HTML to the body (i.e., between the `<body>` and `</body>` tags); something like `<a href="link">Data</a>`, where “link” is the URL for the data. This link to the data should appear on your html page.

Using the same process as we used for the matplotlib assignment, upload your html page to github, such that it is accessible on github pages, and insert the link to your page below:

TODO: [https://richardl1111.github.io/si649/lab/lab5/alt\\_theme.html](https://richardl1111.github.io/si649/lab/lab5/alt_theme.html)

Finally, open the link, and verify that you are able to download the data using the link that you added.

Next, click on the three dots at the upper right of the plot, and click on “View Source”. Note that this plot thus supports some basic interaction, even though this is a static html page.

After clicking “View Source”, copy the plot’s source description and paste it below:

```

{ "config": { "title": { "font": "Roboto Slab", "fontSize": 24, "color": "#1A3C40", "offset":
20 }, "subtitle": { "font": "Roboto Slab", "fontSize": 18, "color": "#1A3C40", "offset": 15
}, "axis": { "labelFont": "Roboto Slab", "labelFontSize": 14, "labelColor": "#1A3C40", "ti
tleFont": "Roboto Slab", "titleFontSize": 16, "titleColor": "#1A3C40" }, "axisX": { "grid":
false}, "axisY": { "grid": true, "gridColor": "#D2D7D3", "gridWidth": 1}, "legend": { "label
Font": "Roboto Slab", "labelFontSize": 14, "titleFont": "Roboto Slab", "titleFontSize": 16, "ti
tleColor": "#1A3C40" }, "mark": { "color": "#A2323E", "range": { "category": [ "#586F6B",
"#9DBF9E", "#D8DBB2", "#E9B872", "#CA7E8D", "#B5515D" ], "ordinal": [ "#FCE1CE",
"#F8B6A3", "#F18578", "#DE5858", "#B02339", "ramp": [ "#F0E5CF", "#E2D19B",
"#C3B177", "#988753", "#6B5F34", "diverging": [ "#404E5C", "#7A9E9F", "#C9DBC7",
"#F8F8F8", "#FCDBBE", "#ED9B67", "#A0412C" ] }, "background": "#F9F9F5", "view":
{ "stroke": null } }, "data": { "name": "data-2297fa504f870cf2a894d2b78016e577", "facet": {
"column": { "field": "class", "header": { "title": null, "sort": [ "Favorable", "Neutral", "Un
favorable", "Unfamiliar", "type": "nominal" } }, "spec": { "layer": [ { "mark": { "type":

```

"bar"}, "encoding": { "color": { "field": "class", "legend": null, "type": "nominal"}, "x": {  
 "axis": { "domain": false, "grid": false, "labels": false, "ticks": false, "title": null }, "field":  
 "percent", "type": "quantitative" }, "y": { "axis": { "grid": false, "title": null }, "field":  
 "character", "sort": { "field": "percent\_favorable", "order": "descending", "type": "nominal"  
 } } }, { "mark": { "type": "text", "align": "left", "baseline": "middle", "dx": 3, "font-  
 Size": 10 }, "encoding": { "color": { "field": "class", "legend": null, "type": "nominal"},  
 "text": { "field": "percent", "format": ".0f", "type": "quantitative"}, "x": { "axis": { "do-  
 main": false, "grid": false, "labels": false, "ticks": false, "title": null }, "field": "percent",  
 "type": "quantitative" }, "y": { "axis": { "grid": false, "title": null }, "field": "character",  
 "sort": { "field": "percent\_favorable", "order": "descending", "type": "nominal" } } } ],  
 "width": 80 }, "title": { "text": "'Star Wars' Favorability Rating", "subtitle": "By 834 respon-  
 dents", "anchor": "start" }, "\$schema": "https://vega.github.io/schema/vega-lite/v5.16.3.json",  
 "datasets": { "data-2297fa504f870cf2a894d2b78016e577": [ { "character": "Anakin Skywalker",  
 "class": "Favorable", "count": 514, "total": 823, "percent": 62.454434993924664, "per-  
 cent\_favorable": 62.454434993924664 }, { "character": "Anakin Skywalker", "class": "Neu-  
 tral", "count": 135, "total": 823, "percent": 16.403402187120292, "percent\_favorable":  
 62.454434993924664 }, { "character": "Anakin Skywalker", "class": "Unfamiliar", "count": 52,  
 "total": 823, "percent": 6.318347509113001, "percent\_favorable": 62.454434993924664 }, {  
 "character": "Anakin Skywalker", "class": "Unfavorable", "count": 122, "total": 823, "per-  
 cent": 14.823815309842042, "percent\_favorable": 62.454434993924664 }, { "character": "Boba  
 Fett", "class": "Favorable", "count": 291, "total": 812, "percent": 35.83743842364532, "per-  
 cent\_favorable": 35.83743842364532 }, { "character": "Boba Fett", "class": "Neutral", "count":  
 248, "total": 812, "percent": 30.541871921182267, "percent\_favorable": 35.83743842364532 },  
 { "character": "Boba Fett", "class": "Unfamiliar", "count": 132, "total": 812, "percent":  
 16.25615763546798, "percent\_favorable": 35.83743842364532 }, { "character": "Boba Fett",  
 "class": "Unfavorable", "count": 141, "total": 812, "percent": 17.364532019704434, "per-  
 cent\_favorable": 35.83743842364532 }, { "character": "C-3P0", "class": "Favorable", "count":  
 703, "total": 827, "percent": 85.00604594921403, "percent\_favorable": 85.00604594921403  
 }, { "character": "C-3P0", "class": "Neutral", "count": 79, "total": 827, "percent":  
 9.552599758162032, "percent\_favorable": 85.00604594921403 }, { "character": "C-3P0", "class":  
 "Unfamiliar", "count": 15, "total": 827, "percent": 1.8137847642079807, "percent\_favorable":  
 85.00604594921403 }, { "character": "C-3P0", "class": "Unfavorable", "count": 30, "total":  
 827, "percent": 3.6275695284159615, "percent\_favorable": 85.00604594921403 }, { "character":  
 "Darth Vader", "class": "Favorable", "count": 481, "total": 826, "percent": 58.23244552058111,  
 "percent\_favorable": 58.23244552058111 }, { "character": "Darth Vader", "class": "Neutral",  
 "count": 84, "total": 826, "percent": 10.16949152542373, "percent\_favorable": 58.23244552058111  
 }, { "character": "Darth Vader", "class": "Unfamiliar", "count": 10, "total": 826, "per-  
 cent": 1.2106537530266344, "percent\_favorable": 58.23244552058111 }, { "character": "Darth  
 Vader", "class": "Unfavorable", "count": 251, "total": 826, "percent": 30.387409200968523,  
 "percent\_favorable": 58.23244552058111 }, { "character": "Emperor Palpatine", "class": "Fa-  
 vorable", "count": 253, "total": 814, "percent": 31.08108108108108, "percent\_favorable":  
 31.08108108108108 }, { "character": "Emperor Palpatine", "class": "Neutral", "count": 213, "to-  
 tal": 814, "percent": 26.16707616707617, "percent\_favorable": 31.08108108108108 }, { "char-  
 acter": "Emperor Palpatine", "class": "Unfamiliar", "count": 156, "total": 814, "percent":  
 19.164619164619165, "percent\_favorable": 31.08108108108108 }, { "character": "Emperor Pal-  
 patine", "class": "Unfavorable", "count": 192, "total": 814, "percent": 23.587223587223587, "per-  
 cent\_favorable": 31.08108108108108 }, { "character": "Han Solo", "class": "Favorable", "count":  
 761, "total": 829, "percent": 91.79734620024126, "percent\_favorable": 91.79734620024126

}, { "character": "Han Solo", "class": "Neutral", "count": 44, "total": 829, "percent": 5.307599517490953, "percent\_favorable": 91.79734620024126 }, { "character": "Han Solo", "class": "Unfamiliar", "count": 15, "total": 829, "percent": 1.8094089264173705, "percent\_favorable": 91.79734620024126 }, { "character": "Han Solo", "class": "Unfavorable", "count": 9, "total": 829, "percent": 1.0856453558504222, "percent\_favorable": 91.79734620024126 }, { "character": "Jar Jar Binks", "class": "Favorable", "count": 242, "total": 821, "percent": 29.476248477466505, "percent\_favorable": 29.476248477466505 }, { "character": "Jar Jar Binks", "class": "Neutral", "count": 164, "total": 821, "percent": 19.97563946406821, "percent\_favorable": 29.476248477466505 }, { "character": "Jar Jar Binks", "class": "Unfamiliar", "count": 109, "total": 821, "percent": 13.276492082825822, "percent\_favorable": 29.476248477466505 }, { "character": "Jar Jar Binks", "class": "Unfavorable", "count": 306, "total": 821, "percent": 37.27161997563947, "percent\_favorable": 29.476248477466505 }, { "character": "Lando Calrissian", "class": "Favorable", "count": 365, "total": 820, "percent": 44.51219512195122, "percent\_favorable": 44.51219512195122 }, { "character": "Lando Calrissian", "class": "Neutral", "count": 236, "total": 820, "percent": 28.780487804878046, "percent\_favorable": 44.51219512195122 }, { "character": "Lando Calrissian", "class": "Unfamiliar", "count": 148, "total": 820, "percent": 18.048780487804876, "percent\_favorable": 44.51219512195122 }, { "character": "Lando Calrissian", "class": "Unfavorable", "count": 71, "total": 820, "percent": 8.658536585365853, "percent\_favorable": 44.51219512195122 }, { "character": "Luke Skywalker", "class": "Favorable", "count": 771, "total": 831, "percent": 92.7797833935018, "percent\_favorable": 92.7797833935018 }, { "character": "Luke Skywalker", "class": "Neutral", "count": 38, "total": 831, "percent": 4.57280385078219, "percent\_favorable": 92.7797833935018 }, { "character": "Luke Skywalker", "class": "Unfamiliar", "count": 6, "total": 831, "percent": 0.7220216606498195, "percent\_favorable": 92.7797833935018 }, { "character": "Luke Skywalker", "class": "Unfavorable", "count": 16, "total": 831, "percent": 1.9253910950661854, "percent\_favorable": 92.7797833935018 }, { "character": "Obi Wan Kenobi", "class": "Favorable", "count": 750, "total": 825, "percent": 90.9090909090909, "percent\_favorable": 90.9090909090909 }, { "character": "Obi Wan Kenobi", "class": "Neutral", "count": 43, "total": 825, "percent": 5.212121212121213, "percent\_favorable": 90.9090909090909 }, { "character": "Obi Wan Kenobi", "class": "Unfamiliar", "count": 17, "total": 825, "percent": 2.0606060606060606, "percent\_favorable": 90.9090909090909 }, { "character": "Obi Wan Kenobi", "class": "Unfavorable", "count": 15, "total": 825, "percent": 1.8181818181818181, "percent\_favorable": 90.9090909090909 }, { "character": "Padme Amidala", "class": "Favorable", "count": 351, "total": 814, "percent": 43.12039312039312, "percent\_favorable": 43.12039312039312 }, { "character": "Padme Amidala", "class": "Neutral", "count": 207, "total": 814, "percent": 25.429975429975432, "percent\_favorable": 43.12039312039312 }, { "character": "Padme Amidala", "class": "Unfamiliar", "count": 164, "total": 814, "percent": 20.14742014742015, "percent\_favorable": 43.12039312039312 }, { "character": "Padme Amidala", "class": "Unfavorable", "count": 92, "total": 814, "percent": 11.302211302211303, "percent\_favorable": 43.12039312039312 }, { "character": "Princess Leia Organa", "class": "Favorable", "count": 757, "total": 831, "percent": 91.0950661853189, "percent\_favorable": 91.0950661853189 }, { "character": "Princess Leia Organa", "class": "Neutral", "count": 48, "total": 831, "percent": 5.776173285198556, "percent\_favorable": 91.0950661853189 }, { "character": "Princess Leia Organa", "class": "Unfamiliar", "count": 8, "total": 831, "percent": 0.9626955475330927, "percent\_favorable": 91.0950661853189 }, { "character": "Princess Leia Organa", "class": "Unfavorable", "count": 18, "total": 831, "percent": 2.166064981949458, "percent\_favorable": 91.0950661853189 }, { "character": "R2 D2", "class": "Favorable", "count": 747, "total": 830, "percent": 90, "percent\_favorable": 90 }, { "character": "R2 D2", "class":



```

“Neutral”, “count”: 57, “total”: 830, “percent”: 6.8674698795180715, “percent_favorable”:
90 }, { “character”: “R2 D2”, “class”: “Unfamiliar”, “count”: 10, “total”: 830, “percent”:
1.2048192771084338, “percent_favorable”: 90 }, { “character”: “R2 D2”, “class”: “Unfavorable”,
“count”: 16, “total”: 830, “percent”: 1.9277108433734942, “percent_favorable”: 90 }, { “char-
acter”: “Yoda”, “class”: “Favorable”, “count”: 749, “total”: 826, “percent”: 90.67796610169492,
“percent_favorable”: 90.67796610169492 }, { “character”: “Yoda”, “class”: “Neutral”, “count”: 51,
“total”: 826, “percent”: 6.174334140435835, “percent_favorable”: 90.67796610169492 }, { “char-
acter”: “Yoda”, “class”: “Unfamiliar”, “count”: 10, “total”: 826, “percent”: 1.2106537530266344,
“percent_favorable”: 90.67796610169492 }, { “character”: “Yoda”, “class”: “Unfavorable”,
“count”: 16, “total”: 826, “percent”: 1.937046004842615, “percent_favorable”: 90.67796610169492
} ] } }

```

```
[21]: theme_1.save('alt_theme.html')
```