Altair Exercises

This notebook will explore multiple different visualizations in Altair.

Part 6

The following exercise is based on the 538 article here (https://fivethirtyeight.com/features/women-in-comic-books/). What you should know is that there are two major comic book companies: DC (Batman, Superman, Wonder Woman, etc.) and Marvel (Black Widow, Iron Man, Hulk, etc.).

We have a dataset of characters, their sex, when they were introduced, if their identify is secret, their eye and hair color, the number of appearances, etc. Lots of dimensions on which to build our visualizations.

In [1]:
```
import pandas as pd
import numpy as np
import altair as alt
```

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

# uses intermediate json files to speed things up
alt.data_transformers.enable('json')

# use the 538 theme
alt.themes.enable('fivethirtyeight')
```

Out[2]:
```
ThemeRegistry.enable('fivethirtyeight')
```

In [3]:
```
# load up the two datasets, one for Marvel and one for DC
dc = pd.read_csv('../assets/dc-wikia-data.csv')
marvel = pd.read_csv('../assets/marvel-wikia-data.csv')
```

In [4]:
```
# Some pre-processing

# Add columns
dc['publisher'] = 'DC'
marvel['publisher'] = 'Marvel'

# rename some columns
marvel.rename(columns={'Year': 'YEAR'}, inplace=True)

# create the table with everything
comic = pd.concat([dc, marvel])

# drop years with na values
comic.dropna(subset=['YEAR'], inplace=True)

comic.sample(5)
```

Out[4]:

Out[4]:

	page_id	name	urlslug	ID	ALIGN	EYE	HAIR
13800	680869	Tokk (Earth-616)	\/Tokk_(Earth-616)	NaN	Bad Characters	NaN	Black Hair
9218	157140	Anarchy (ULTIMATUM) (Earth-616)	\/Anarchy_(ULTIMATUM)_(Earth-616)	Secret Identity	Bad Characters	NaN	Red Hair
1002	17302	Norton Fester (Earth-616)	\/Norton_Fester_(Earth-616)	Secret Identity	Bad Characters	Brown Eyes	Brown Hair
6408	285122	El Dorado (New Earth)	\/wiki\/El_Dorado_(New_Earth)	Secret Identity	Bad Characters	Black Eyes	NaN
6389	529273	Leviathan (smuggler) (Earth-616)	\/Leviathan_(smuggler)_(Earth-616)	Secret Identity	Bad Characters	Blue Eyes	NaN

Comic Books Are Still Made By Men, For Men And About Men

Original article available at [FiveThirtyEight (https://fivethirtyeight.com/features/women-in-comic-books/)](https://fivethirtyeight.com/features/women-in-comic-books/)

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

Get the data on [GitHub (https://github.com/fivethirtyeight/data/tree/master/comic-characters)](https://github.com/fivethirtyeight/data/tree/master/comic-characters)

We are going to be revising and adding to the visualizations for this article. While they're nice, we think we can do better by adding some interactivity.

New Comic Book Characters Introduced per Year

We'd like to build an interactive visualization that allows us to compare the distributions of characters over time as well. The top two charts will represent the total characters over time (as bar charts). The bottom two will be a line chart with separate lines for female and male characters.

In [5]:
```
# let's pre-process the data. We're going to focus on just Female and
# for the moment and will only consider those
comic_ch1_df = comic[(comic['YEAR'] >= 1940) & (comic['SEX'].isin(['Fe
```

In [6]:
```
comic_ch1_df.sample(5)
```

Out[6]:

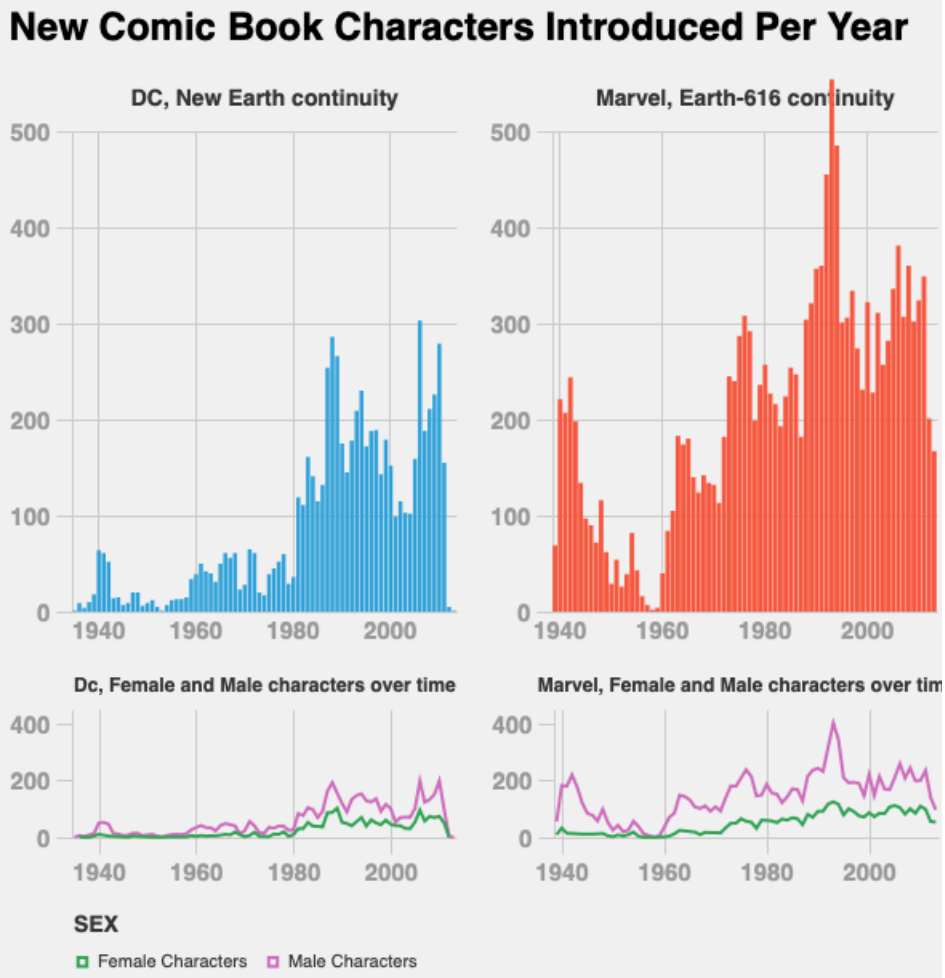
	page_id	name	urlslug	ID	ALIGN	EYE	HAIR
4770	45671	Alec Dalton (Earth-616)	\/Alec_Dalton_(Earth-616)	Secret Identity	Good Characters	Brown Eyes	Black Hair
4405	116756	Magus Eximus (New Earth)	\/wiki\/Magus_Eximus_(New_Earth)	NaN	Good Characters	NaN	NaN
8405	445678	Corky Grogan (Earth-616)	\/Corky_Grogan_(Earth-616)	Public Identity	Good Characters	NaN	Black Hair
13036	183809	Bouncer (Speedball Foe) (Earth-616)	\/Bouncer_(Speedball_Foe)_(Earth-616)	NaN	Bad Characters	NaN	NaN
12322	24327	Drexxon (Earth-616)	\/Drexxon_(Earth-616)	Secret Identity	Bad Characters	Green Eyes	No Hair

In [7]:
```
p1_bar_base = alt.Chart(comic).mark_bar(size=2.5).encode(
    alt.Y('count():Q',
          axis=alt.Axis(values=[0, 100, 200, 300, 400, 500],
                        title=None,
                        labelFontWeight="bold",
                        labelFontSize=15),
          scale=alt.Scale(domain=[0, 500]))).properties(
                width=240,
                height=300
)



# let's create the bar chart for DC. We'll take the "base" chart
bar_dc = p1_bar_base.encode(alt.X('YEAR:N',  # create the X axis based
                                  axis=alt.Axis(values=[1940, 1960, 1980,
                                                        title="DC, New Earth cont
                                                        titlePadding=-347,
                                                        labelAngle=360,
                                                        labelFontWeight="bold",
                                                        labelFontSize=15,)),
        ).transform_filter(
            # we will use Altair's filter to only keep DC for this cha
            alt.datum.publisher == 'DC'
        )

# let's do the same thing for marvel
```

```
bar_marvel = p1_bar_base.mark_bar(color='#f6573f').encode(alt.X('YEAR:
                            # fix the look of the axes
                           axis=alt.Axis(values=[1940, 1960, 1980, 200
                                         title="Marvel, Earth-616 cont
                                         titlePadding=-347,
                                         labelAngle=360,
                                         labelFontWeight="bold",
                                         labelFontSize=15)),
        ).transform_filter(
            # we will use Altair's filter to only keep DC for this cha
            alt.datum.publisher == 'Marvel'
        )



# let's create a new "base" chart for the two line charts. We'll take
# and modify it to use a line chart
p1_line_base = p1_bar_base.mark_line().encode(
     # the X axis will be year
     alt.X('YEAR:N'),
     # the Y axis will be the count (the number of points that year)
     alt.Y('count():Q', axis=alt.Axis(grid=False,
                                      labelFontWeight="bold",
                                      labelFontSize=15,
                                      title=None)),
     # let's split the data and color by SEX
     alt.Color('SEX',
               scale = alt.Scale(domain=['Female Characters', 'Male Cha
               legend=alt.Legend(orient="bottom"))
    ).properties(
                width=240, height=80
     )


line_dc = p1_line_base.encode(alt.X('YEAR:N',
                                    axis=alt.Axis(values=[1940, 196
                                                          grid=Tru
                                                          labelAng
                                                          labelFor
                                                          labelFor
                                                          title =
                                                          titlePad
                                                          titleFor
                                                         )
                                     )
            ).transform_filter(
                # this is the DC line chart, so we only want DC
                alt.datum.publisher == 'DC'
            )



line_marvel = p1_line_base.encode(alt.X('YEAR:N',
                                      axis=alt.Axis(values=[1940, 1960,
                                                            grid=True,
                                                            labelAngle=3
                                                            labelFontWei
                                                            labelFontSiz
                                                            title = 'Mar
                                                            titlePadding
                                                            titleFontSiz
                                                           )
                                      )
            ).transform_filter(
                # this is the Marvel line chart, so we only want Marve
                alt.datum.publisher == 'Marvel'
            )



# let's put everything together
# top piece
```

```
top_charts = alt.hconcat(bar_dc,bar_marvel).resolve_scale(y='shared'
           ).properties(
                    title='New Comic Book Characters Introduced Per Ye
           )

# bottom piece
bottom_charts = alt.hconcat(line_dc,line_marvel).resolve_scale(y='shar

alt.vconcat(top_charts,bottom_charts).configure_view(
    strokeWidth=0
)
```

Out[7]:



Use the code below to create a "brush" object (a "selection" in Altair speak) that will let us select a time range. We will then create a condition for the DC chart, and add both the condition and selection to the DC chart. We will then repeat with Marvel. This will create interactivity with the chart.

In [8]:

```
## DC
# Create brush object
brush = alt.selection_interval(encodings=['x'])

# Create condition - DC
colorConditionDC = alt.condition(brush,alt.value("#2182bd"),alt.value(

# Add condition and selection - DC
i_bar_dc = bar_dc.encode(
    color=colorConditionDC
).add_selection(
    brush
)

# Create condition - Marvel
colorConditionMarvel = alt.condition(brush,alt.value("#f6573f"),alt.va
```

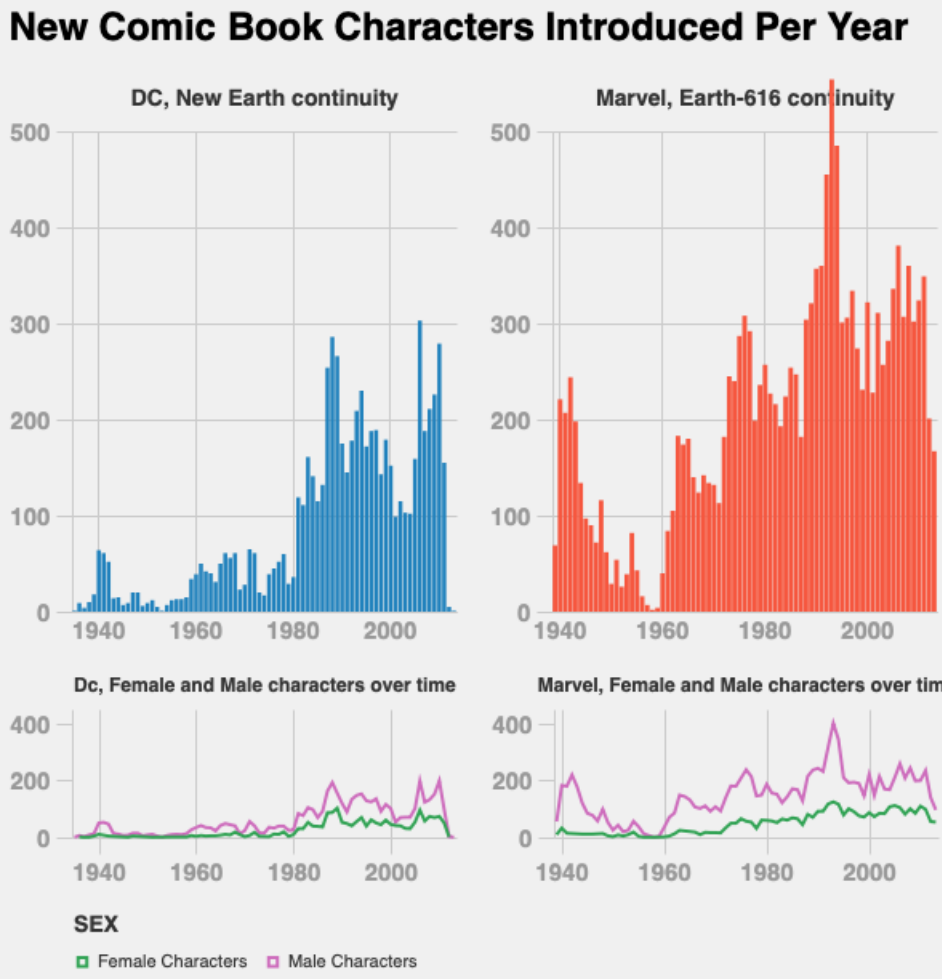
```
# Add condition and selection - Marvel
i_bar_marvel = bar_marvel.encode(
    color=colorConditionMarvel
).add_selection(
    brush
)

# top piece
top_charts = alt.hconcat(i_bar_dc,i_bar_marvel).resolve_scale(y='share
           ).properties(
                    title='New Comic Book Characters Introduced Per Ye
           )
```

Now we modify the two line charts.

In [9]:
```
i_line_dc = line_dc.add_selection(
    brush
).transform_filter(
    brush
)

i_line_marvel = line_marvel.add_selection(
    brush
).transform_filter(
    brush
)

# bottom piece
bottom_charts = alt.hconcat(i_line_dc,i_line_marvel).resolve_scale(y='
```

In [10]:
```
# let's put everything together with the new interactive charts.

# bottom piece
bottom_charts = alt.hconcat(i_line_dc,i_line_marvel).resolve_scale(y='

alt.vconcat(top_charts,bottom_charts).configure_view(
    strokeWidth=0
)
```

Out[10]:



Comics Aren't Gaining Many Female Characters

This chart will only present one point of interest: Percent female in any given year. It might help us understand the claim that there's a relatively trending change in this percent by plotting year-over-year percent changes. Also, it's possible that there are more characters being introduced in later years. So even one or two good years in the 2000's may make up for lots of bad years in the past (it turns out that this is not the case, but it is a question we might ask).

In [11]:
```
def generatePercentTable(publisher):
    _df = comic[comic.publisher == publisher]
    _df = _df[['SEX','YEAR']]
    _df = pd.get_dummies(_df)
```

```
    _df.YEAR = _df.YEAR.astype('int')
    _df = _df.groupby(['YEAR']).sum()

    _df['total'] = 0
    _df['total'] = _df['total'].astype('int')
    for col in list(comic[comic.publisher == publisher].SEX.unique()):
        col = str(col)
        if (col != 'nan'):
            _df['total'] = _df['total'].astype('int') + _df["SEX_"+col

    _df['% Female'] = _df['SEX_Female Characters'] / _df.total
    _df = _df.reset_index()
    _df = _df[['YEAR','% Female','SEX_Female Characters','SEX_Male Cha
    _df['publisher'] = publisher
    _df = _df[(_df.YEAR >= 1979)]
    _df['Year-over-year change in % Female'] = _df['% Female'].pct_cha
    toret = _df[(_df.YEAR > 1980) & (_df.YEAR < 2013)].copy()
    t2 = toret.cumsum()
    toret['% Female characters to date'] = list(t2['SEX_Female Charact
    return(toret)

changedata = pd.concat([generatePercentTable("Marvel"),generatePercent

changedata = pd.melt(changedata,id_vars=['YEAR','publisher'],value_var
                                                            'Year-ove
                                                            '% Female
```

In [12]:
```
changedata.sample(5)
```

Out[12]:

	YEAR	publisher	variable	value
164	1985	DC	% Female characters to date	0.307937
80	1997	Marvel	Year-over-year change in % Female	-0.070558
69	1986	Marvel	Year-over-year change in % Female	-0.018952
108	1993	DC	Year-over-year change in % Female	0.171688
122	2007	DC	Year-over-year change in % Female	-0.026070

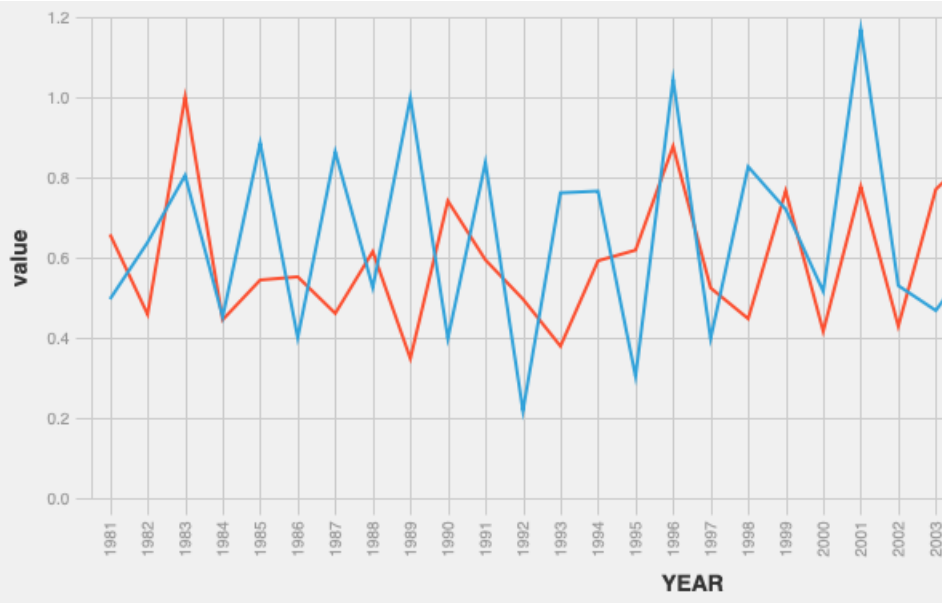
The first job will be to create an interactive chart that has a drop-down box that allows us to select the variable of interest.

In [13]:
```
def generateLineChartP21():

    metricOptions = ['% Female','Year-over-year change in % Female','%
    input_dropdown = alt.binding_select(options=metricOptions)

    dropdown_selection = alt.selection_single(fields=['variable'], bin

    line = alt.Chart(changedata).mark_line().encode(
        x=alt.X('YEAR:N'),
        y=alt.Y("sum(value)", title='value'),
        color='publisher'
    ).add_selection(
        dropdown_selection
    ).transform_filter(
        dropdown_selection
    ).properties(width=750, height=300)

    return(line)
```

```
    return(line)
```

In [14]:
```
generateLineChartP21()
```

Out[14]:



Data__variable % Female

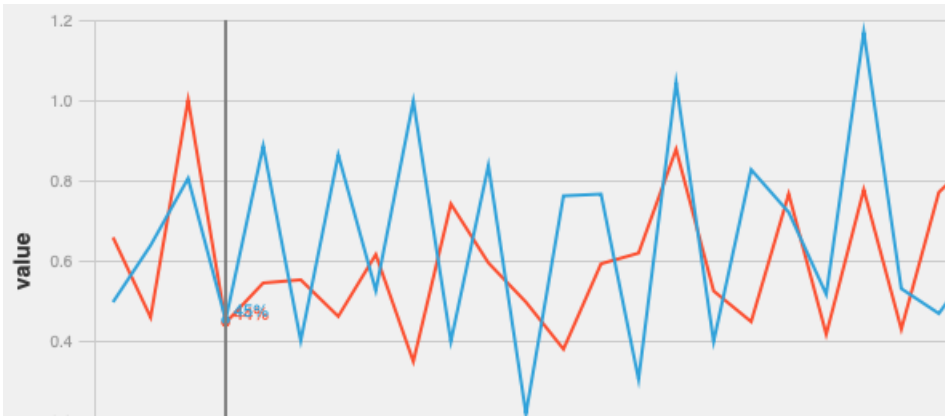
This is pretty static, so let's add some annotations and interactivity.

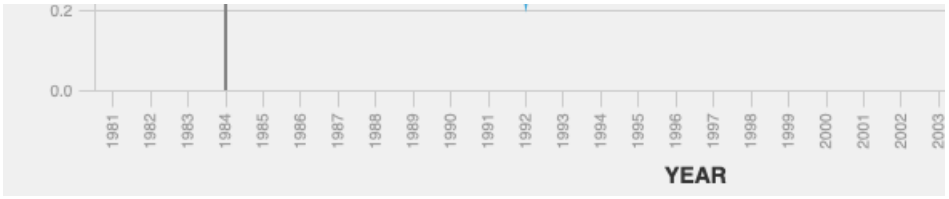
In [15]:
```
def generateLineChartP22():

    metricOptions = ['% Female','Year-over-year change in % Female','%
    input_dropdown = alt.binding_select(options=metricOptions)

    dropdown_selection = alt.selection_single(fields=['variable'], bir

    nearest = alt.selection(type='single',nearest=True, on='mouseover'
                            fields=['YEAR'],empty='none')

    line2 = alt.Chart(changedata).mark_line().encode(
        x=alt.X('YEAR:N'),
        y=alt.Y("sum(value):Q", title='value'),
        color='publisher:N'
    )

    selectors = alt.Chart(changedata).mark_point().encode(
        x=alt.X('YEAR:N',axis=alt.Axis(labels=True)),
        opacity=alt.value(0)
    ).add_selection(
        nearest
    )
```

```
    points = line2.mark_point().encode(
        opacity=alt.condition(
            nearest,
            alt.value(1),
            alt.value(0)
        )
    )

    text = line2.mark_text(align='left', dx=5, dy=-5).encode(
        text=alt.condition(
            nearest,
            "sum(value):Q",
            alt.value(' '),
            format='.0%'
        )
    )

    rules = alt.Chart(changedata).mark_rule(color='gray').encode(
        x=alt.X('YEAR:N',),
    ).transform_filter(
        nearest
    )

    final = alt.layer(line2, selectors, points, rules, text).add_selec
        dropdown_selection
    ).transform_filter(
        dropdown_selection
    ).properties(width=750,height=300)

    return(final)
```

In [16]:
```
generateLineChartP22()
```

Out[16]:





Data__variable % Female

Exercise adapted and modified from UMSI homework assignment for SIADS 622.