SI649-21-Fall Lab 4 -> Altair Transformation

Overview

We will work with a dataset from the article "Joining The Avengers Is As Deadly As Jumping Off A Four-Story Building". We'll focus on **transformation**. In this lab, you will practice:

1. transform_calculation
2. transform_aggregate
3. transform_join_aggregate
4. transform_window
5. transform_filter
6. exporting to HTML

Data

The source data is available at this link. You can also find a description of each column on the website. We preprocessed the dataset for you, and you do not need to write any data processing code.

For this lab, please write Altair code to answer the questions. In many situations, you could also solve the problem using Pandas. However, we want code that can be deployed without using pandas, so it's better to practice to just do as much as we can in Altair. You can complete the entire lab without writing any pandas transformation.

It's fine if your visualization looks slightly different from the example (e.g., getting 1.1 instead of 1.0, use orange instead of red)

Lab Instructions (read the full version on the handout of the previous lab)

- Save, rename, and submit the ipynb file (use your uniqname in the name).
- Run every cell (do Runtime -> Restart and run all to make sure you have a clean working version), print to pdf, submit the pdf file.
- For each visualization, we will ask you to write down a "Grammar of Graphics" plan first (basically a description of what you'll code).
- 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.

In [ ]:

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

In [ ]:

```
#obtain the data and process the data
# you do not need to edit this block, just run it
# and use avenger_df as your starting point
avenger_link="https://raw.githubusercontent.com/fivethirtyeight/data/master/a
```

```
avenger_return=pd.read_csv(avenger_link,encoding="latin-1")
articleLink="https://fivethirtyeight.com/features/avengers-death-comics-age-o
#process the dataset
convertColumnList=[]
modified=[]
for i in range(5):
  convertColumnList.append("Death"+str(i+1))
  convertColumnList.append("Return"+str(i+1))

for c in convertColumnList:
  avenger_return[c]=avenger_return[c].astype(str)

for j in range(len(avenger_return)):
  modified.append([])
  storage=modified[-1]
  for c in convertColumnList:
    item=avenger_return[c][j]
    v = 1 if item=="YES" else 0
    storage.append(v)

organizedDR=pd.DataFrame(modified,columns=convertColumnList)
shortlist=avenger_return.drop(columns=convertColumnList)
avenger_df=shortlist.join(organizedDR)
```

In [ ]:

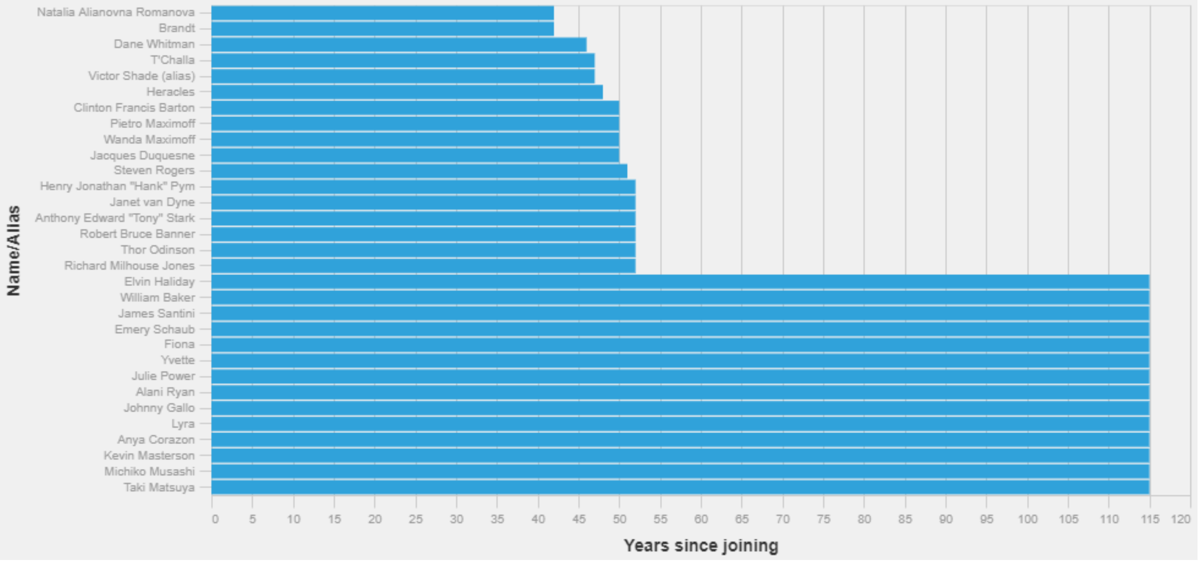
```
# you can set to whatever theme you prefer.
alt.themes.enable('fivethirtyeight')
avenger_df.sample(2)
```

Out[ ]:

	URL	Name/Alias	Appearances	Current?	Gende
136	http://marvel.wikia.com/Kevin_Masterson_(Earth...	Kevin Masterson	62	YES	MAL
147	http://marvel.wikia.com/Eden_Fesi_(Earth-616)#	Eden Fesi	65	YES	MAL

2 rows × 21 columns

Visualization 1: Top 30 avengers who have been in the series for the longest time



Description of the visualization:

We want to find avengers who hae been in the series for the longest time. We'd like to build a bar chart with the top 30 avengers ranked by their years since join.

- The number of years that they have been in the series (column "Years since joined") is plotted on the X axis
- The avenger's name is displayed on the y axis
- Only the top 30 avengers are included in the chart

Step 1: Write down your plan for the visualization (edit this cell)

- Data Name: avenger_df
- mark type: *mark_bar*
- Encoding Specification:
- > x:*Years since joining:Q*
- > y: *Name/Alias:N*
- Transformation Plan:
- > step 1:*Use the window transform to sort the avengers by the number of years the avengers have joined, then rank the number of years the avengers have joined*
- > step 2:*Use the filter transform to select only the top 30 avengers*

Hint

- You need to rank avengers by the number of years they have joined. Review our lab demo and the altair documentation. Which transformation can generate the rank?
- Because we only want to include the top 30 avengers, which transformation can we use?
- Adding tooltips can be a helpful way to check if your calculation is correct.

Step 2: Create your chart.

Please paste your FINAL answer to the cell immediately blow this block (it will allow us to grade). You can search for the keyword "TODO" to locate cells that need your edits

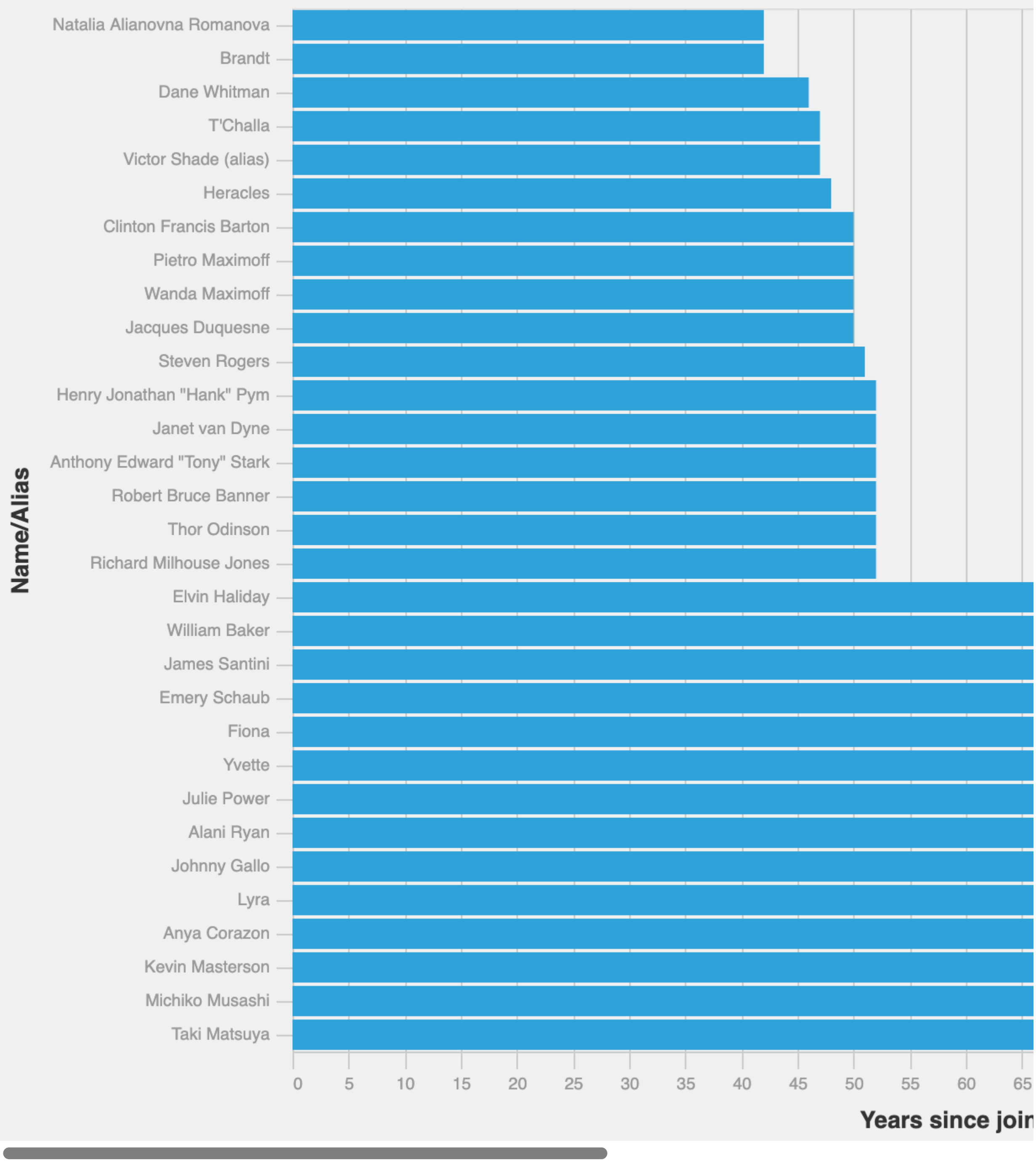
In [ ]:

```
# TODO: (vis1) top 30 avengers who have been in the series for the longest ti

chart1 = alt.Chart(avenger_df, width=800).transform_window(
  sort=[alt.SortField('Years since joining', order='descending')], #ordering
  year_rank='rank(*)' #creates a new column, decides rank relative to the lis
).transform_filter(
    alt.datum.year_rank <= 30
).mark_bar().encode(
    alt.X('Years since joining:Q', axis=alt.Axis(values=list(range(0, 125, 5)
    alt.Y('Name/Alias:N',sort=alt.EncodingSortField(   #need an additional so
            field="year_rank",order="descending"))
)
```

```
chart1.encode(
    tooltip=['Name/Alias:N','year_rank:O', 'Years since joining:Q']
).interactive()
```

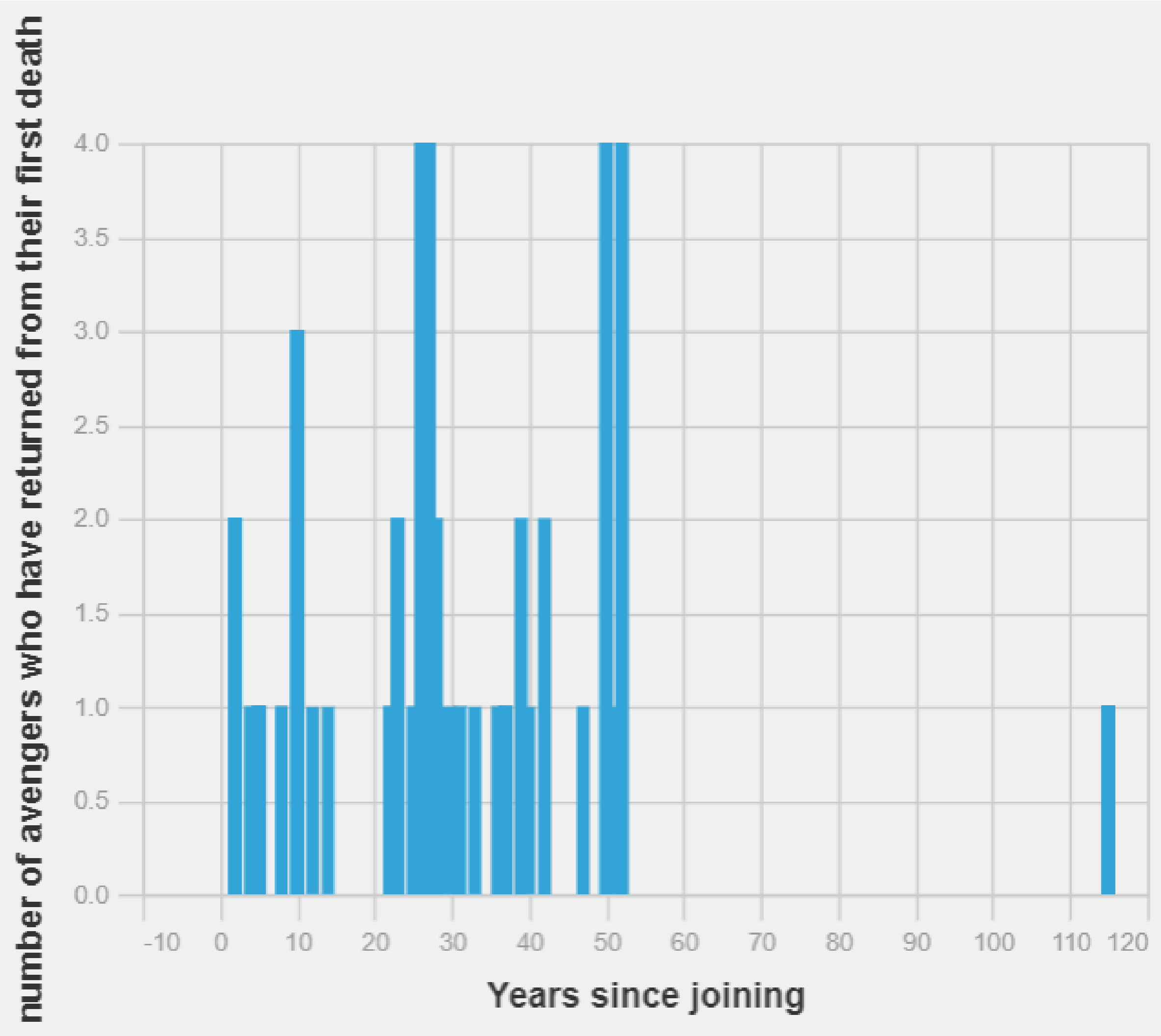
Out[ ]:



In [ ]:

```
chart1.save('chart1.json')
```

Visualization 2: Count of avengers who return from their first death



Description of the data: We want to count the number of avengers who have returned from their first death. We have processed the data so that for every Death/Return column:

- 0 indicate that they did not die or did not return
- 1 indicate that they died or returned
  - e.g., if an avenger died at least once and did not return, their data looks like this: (Death1=1, Return1=0)
  - e.g., if an avenger did not die twice (Death2==0), then they could not have returned twice (Return2==0)

Description of the visualization:

- a bar chart
- y-axis plot the total number of avengers who have returned within an "age group"
- x-axis plot the "age group": number of years since they have joined.
- y-axis uses the label: *number of avengers who have returned from their first death*

Step 1: Write down your plan for the visualization (edit this cell)

- Data Name: avenger_df
- mark type: *mark_bar*
- Encoding Specification:

- > x:*Years since joining:Q*

- > y:*sum(Return1):Q*
- Transformation Plan:
- > step 1:*Use the aggergate transform to group the data by the number of years the avengers have joined, then calculate the sum of Return1*

Hint

1. We want to plot a count of avengers who have returned from their first death. For these avengers, their Return1 value would be equal to 1. Do you need to look at other columns (such as Death1)?
2. Do you want to use joinaggregate or aggergate?
3. How can you produce a calculated/aggregated value within a group?

Step 2: Create your chart.

Please paste your FINAL answer to the cell immediately blow this block (it will allow us to grade). You can search for the keyword "TODO" to locate cells that need your edits

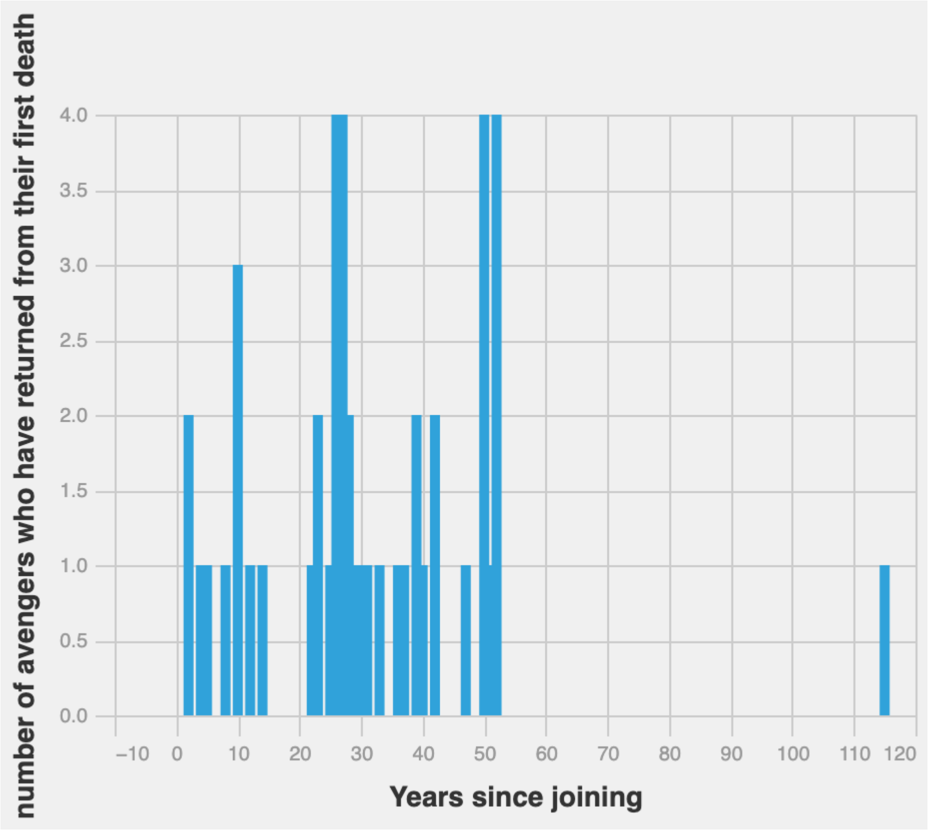
In [ ]:

```
#TODO: vis2
# number of avengers who returned from the 1st death, breakdown by years sinc

chart2 = alt.Chart(avenger_df).transform_aggregate(
    groupby=['Years since joining'],
    sum_Return1='sum(Return1)'
).mark_bar().encode(
    alt.X('Years since joining:Q'),
    alt.Y('sum_Return1:Q', title='number of avengers who have returned from t
)

chart2
```

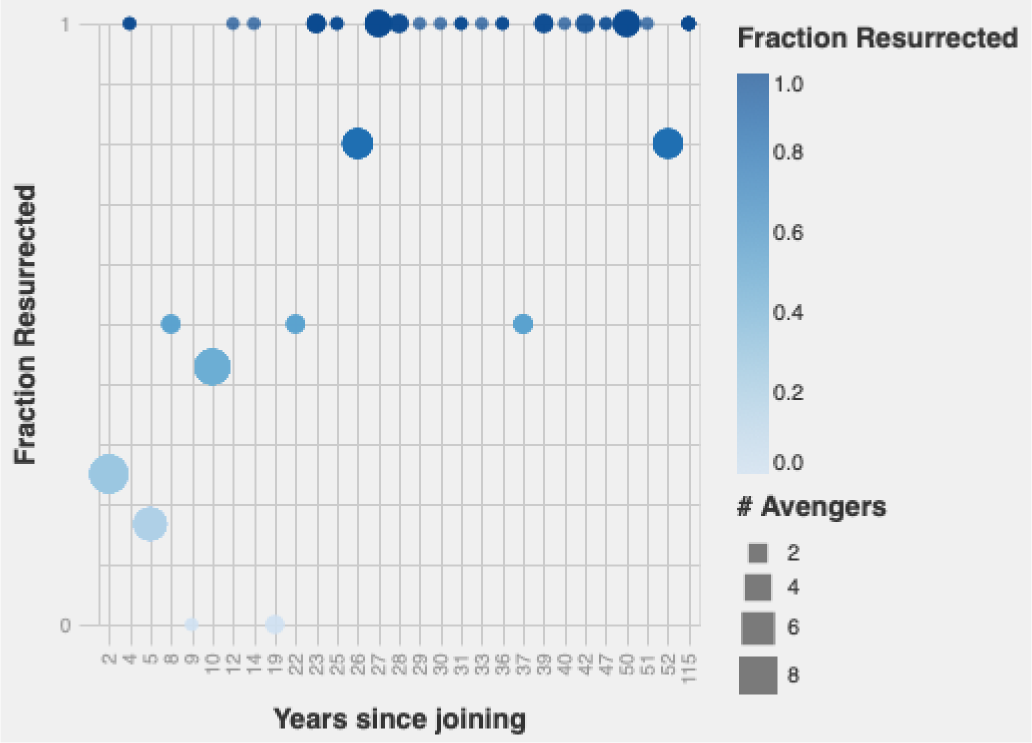
Out[ ]:



In [ ]:

```
chart2.save('chart2.json')
```

Visualization 3: Chance of resurrection increases with years in service



The previous chart displays the number of time that avengers have returned from their first death, broken down by their years since joining. We also want to see how likely it is for them to be resurrected. Therefore, we are going to make a scatter plot with the fraction of avengers resurrected, the years since they've joined, and how many avengers died.

Description of the visualization:

- a scatterplot
- x-axis plot the number of years that they have joined
- y-axis plot the fraction of avengers that were resurrected, which is double encoded with color
    - y-axis uses the label: *Fraction Resurrected*
- circle size encodes the number of avengers who died at least once.
- (optional) Tooltips that display the encoded data

Hint:

- Just like the previous chart, you want to produce a count of avengers who have returned after their first death. Do you also need a count of avengers who have died at least one time?
- Should you use aggregate or join aggregate?
- How can you calculate the fraction of avengers that were resurrected? What is the numerator of the fraction and what is the denominator?

Step 1: Write down your plan for the visualization (edit this cell)

- Data Name: avenger_df
- mark type: *mark_circle*
- Encoding Specification:

- > x:*Years since joining:N*

- > y: *Fraction Reserrected:Q*

- Transformation Plan:

- > step 1:*Use the aggregate transeform to group the data by the number of years the avengers have joined, then calculate the sum of Death1 and Return1*

- > step 2:*Use the calculate transform to calculate the fraction by dividing sum_Return1 by sum_Death1*

Step 2: Create your chart.

Please paste your FINAL answer to the cell immediately blow this block (it will allow us to grade). You can search for the keyword "TODO" to locate cells that need your edits

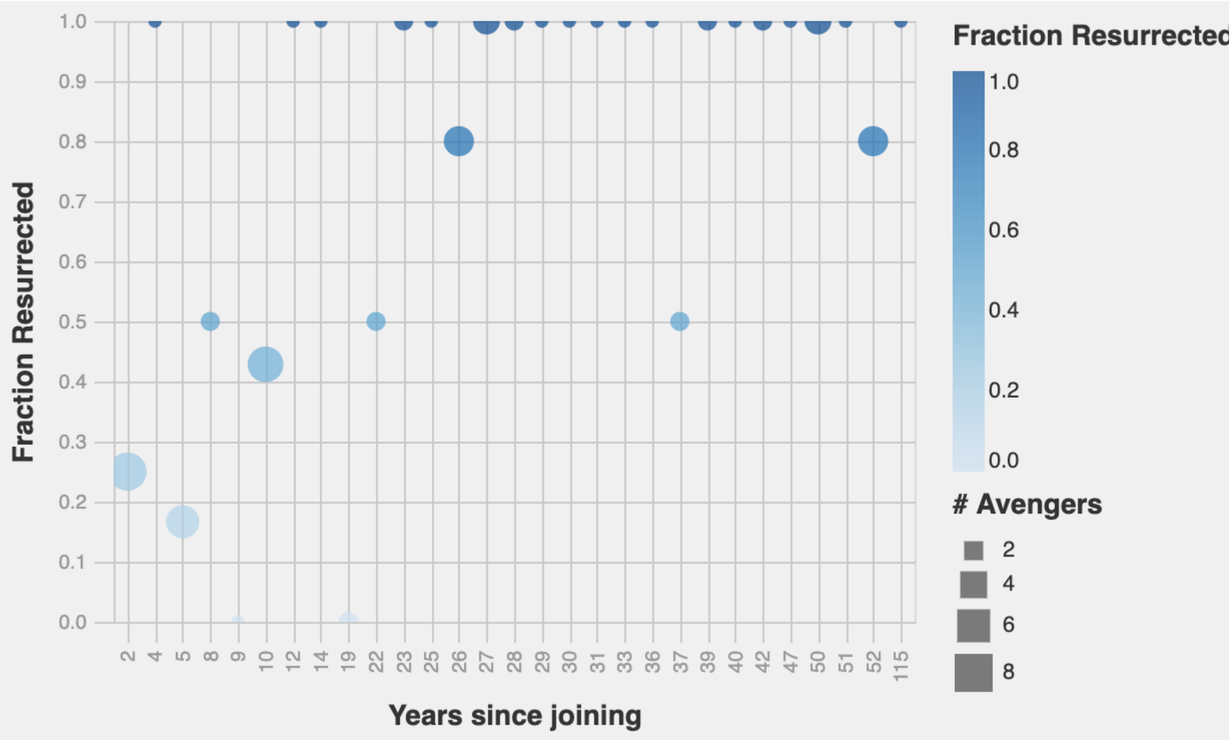
In [ ]:

```
#TODO vis 3
#chance of resurrection and years since joining

chart3 = alt.Chart(avenger_df, width=400).transform_aggregate(
    groupby=['Years since joining'],
    sum_Death1='sum(Death1)',
    sum_Return1='sum(Return1)'
).transform_calculate(
    fraction=alt.datum.sum_Return1 / alt.datum.sum_Death1
).mark_circle().encode(
    alt.X('Years since joining:N'),
    alt.Y('fraction:Q', title='Fraction Resurrected'),
    alt.Color('fraction:Q', title='Fraction Resurrected'),
    alt.Size('sum_Death1:Q', title='# Avengers')
)

chart3.encode(
    tooltip=['Years since joining:N','fraction:Q', 'sum_Death1:Q']
).interactive()
```

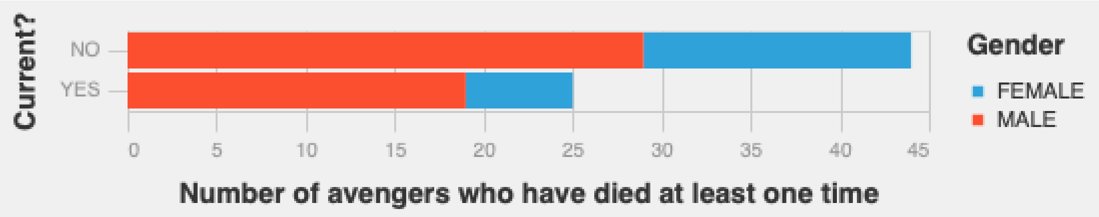
Out[ ]:



In [ ]:

```
chart3.save('chart3.json')
```

Vis 4: Number of avengers who have died at least once, divided by current membership and gender



Instead of looking at years since joined, we want to investigate in the relationship between count of avengers who have died at least once and their other properties such as whether they are a current member or not, and gender.

Description of the visualization:

- a bar chart
- x-axis displays the total number of avengers who have **died at least one time** within an "age group"
    - x-axis uses the label: *number of avengers who have died at least one time*
- y-axis displays whether they are a current member or not

- use color to display the gender information

Hint:

- Do you want to use joinaggregate or aggregate?

Step 1: Write down your plan for the visualization (edit this cell)

- Data Name: avenger_df
- mark type: *mark_bar*
- Encoding Specification:

- > x:*sum_Death1:Q*

- > y: *Current?:N*

- Transformation Plan:

- > step 1:*Use the aggregate transform to group the data by Current? and Gender, then calculate the sum of Death1*

Step 2: Create your chart.

Please paste your FINAL answer to the cell immediately below this block (it will allow us to grade). You can search for the keyword "TODO" to locate cells that need your edits

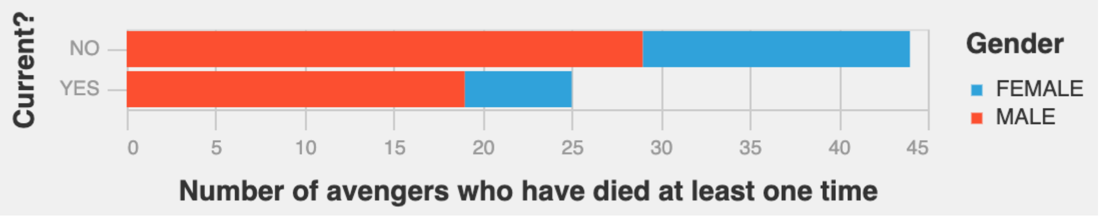
In [ ]:

```
#TODO Vis 4
  #number of avenger who has died at least once,
  #divided by current membership and gender

chart4 = alt.Chart(avenger_df, width=400).transform_aggregate(
    groupby=['Current?', 'Gender'],
    sum_Death1='sum(Death1)'
).mark_bar().encode(
    alt.X('sum_Death1:Q', title='Number of avengers who have died at least on
    alt.Y('Current?:N'),
    alt.Color('Gender:N'),
    order=alt.Order(
      'Gender:N',
      sort='descending'
    )
)

chart4
```

Out[ ]:



In [ ]:

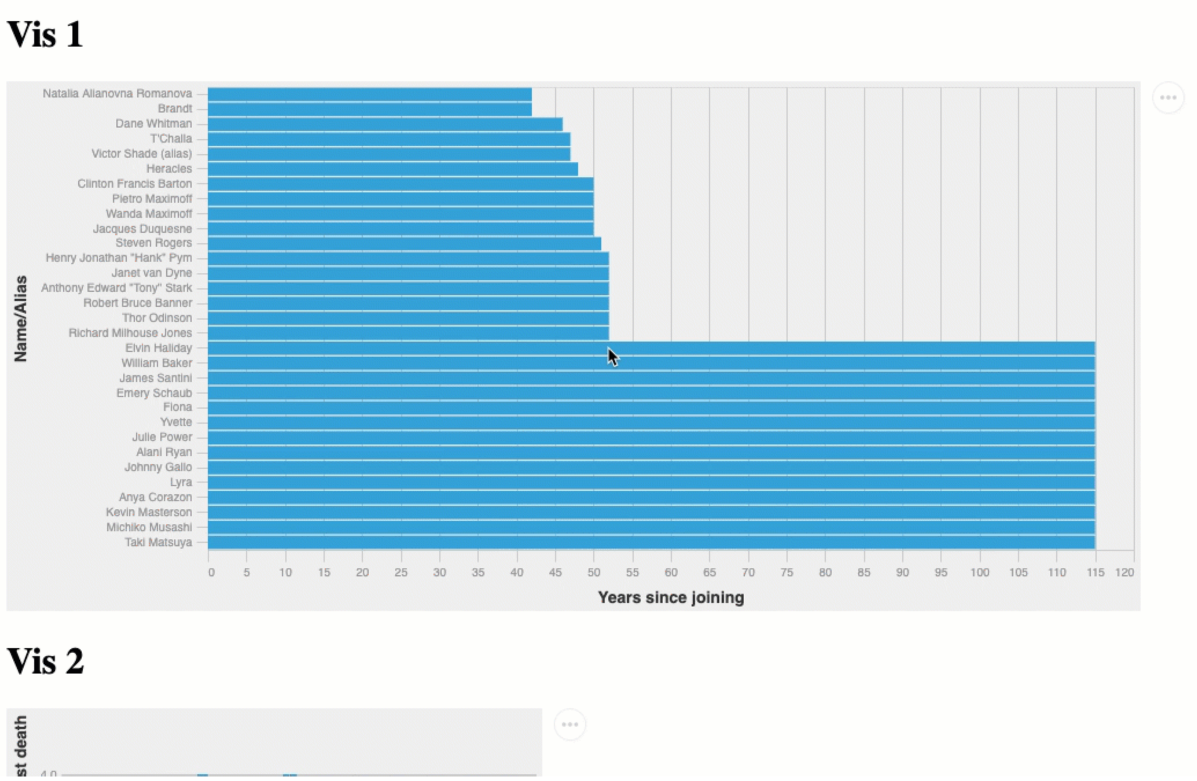
```
chart4.save('chart4.json')
```

Submission

Once you are done with all the altair-related coding, we want you to embed these charts in an HTML file:

1. export these chart specifications to JSON
2. embed them into an HTML file.
3. If you use multiple files (e.g., HTML + JSON + javascript instead of pasting JSON into the HTML file), make sure you zip the entire folder.

The final website will look similar to this:



Vis 1

Vis 2

We also want to have a copy of your python script. Please run all cells (Runtime->Run all), and

1. save to PDF (File->Print->Save PDF)
2. save to ipynb

Rename all files with your uniqname: e.g.,

1. uniqname.pdf
2. uniqname.ipynb
3. uniquename.zip (html)

Upload all files to canvas. *End of Lab 4 Homework*