SI649-20winter Lab 3 -> Altair II

Overview

We will continue working with the dataset from the article “The Dollar-And-Cents Case Against Hollywood’s Exclusion of Women”--don't worry, we'll switch soon. We'll focus on **transformation** and build the following charts:

1. Line chart within a range
2. Heatmap with additional annotations
3. Bar chart with additional annotation.
4. Bar Chart with fold
5. Ugliest chart

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 Python so it's better 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 username 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.
- There are many bonus point opportunities in this lab.

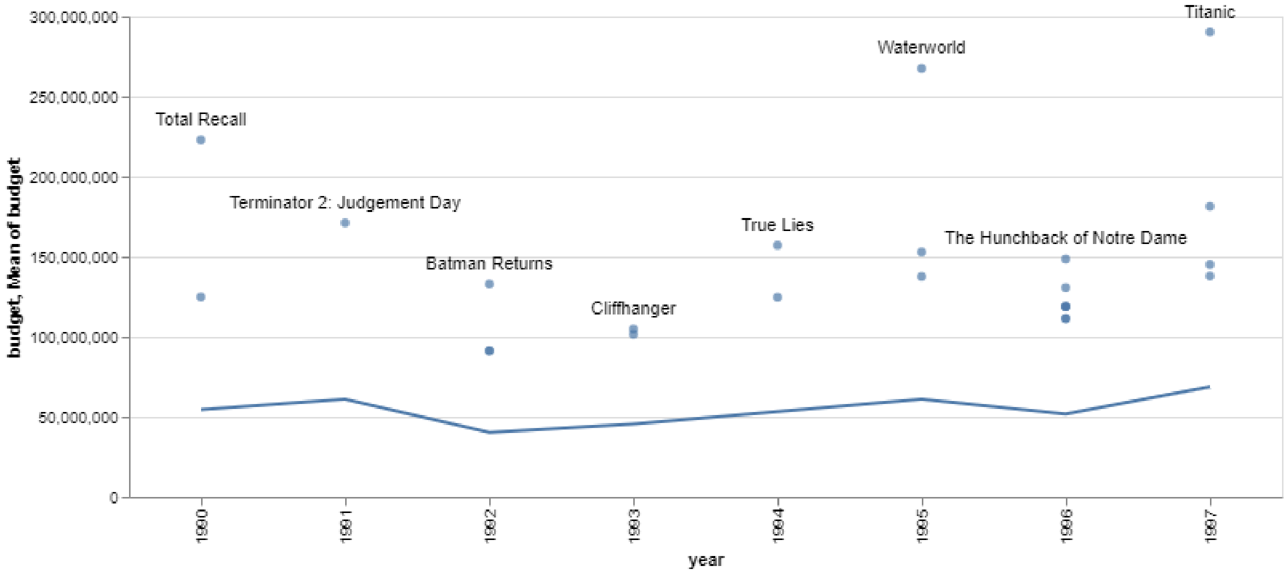
```
# imports we will use
import altair as alt
import pandas as pd
datasetURL="https://raw.githubusercontent.com/LiciaHe/SI649/master/week3/movie_after_1990.csv"
movies_test=pd.read_csv(datasetURL, encoding="latin-1")
```

Here's the processed Bechdel test dataset.

```
movies_test.sample(2)
```

	Unnamed: 0	year	title	binary	budget	dom_gross	int_gross	rating	country	language	test_result	country_binary	roi_dom	roi_int
1281	1281	1999	American Pie	FAIL	16781544	142364760.0	328251410.0	7.0	United States	English	Women only talk about men	U.S. and Canada	8.483412	11.07685

Visualization 1: Line chart within a range



Description of the visualization:

We want to see a visualization of movies that were significantly above budget relative to the mean. We'd like to know the name of the top movie that year but also to have a sense of how many other outliers (we'll defined those as at least 2x the mean budget) were produced that year.

- plot a line chart using year and average budget for movies between 1990 and 1998.
- plot dots to represent movies with budgets that are at least 2 times bigger than the mean budget of that year. (e.g., if the mean budget of 1990 is 50M, plot movies made in 1990 whose budgets are at least 100M)
- For each year, annotate the title of the movie with the highest budget of that year.

Visualization 1 Plan:

TODO: edit this cell to write your visualization plan. You can write in altair syntax, in full sentence, or in bullet points, whichever way that helps you to plan your chart.

line chart

- Describe the encoding rules:

  > x: year:O
  >
  > y: mean(budget):Q

- Describe the transformations:

  > filter: year >= 1990 & year < 1998

dot chart

- Describe the encoding rules:

  > x: year:O
  >
  > y: budget:Q

- Describe the transformations:

  > filter: year >= 1990 & year < 1998
  >
  > joinaggregate: mean_budget="mean(budget)", groupby=["year"],
  >
  > filter: budget >= 2*mean_budget

text annotation

- Describe the encoding rules:

  > text: title:N

- Describe the transformations:

  > filter: year >= 1990 & year < 1998
  >
  > joinaggregate: max_budget="max(budget)", groupby=["year"],
  >
  > filter: budget == max_budget

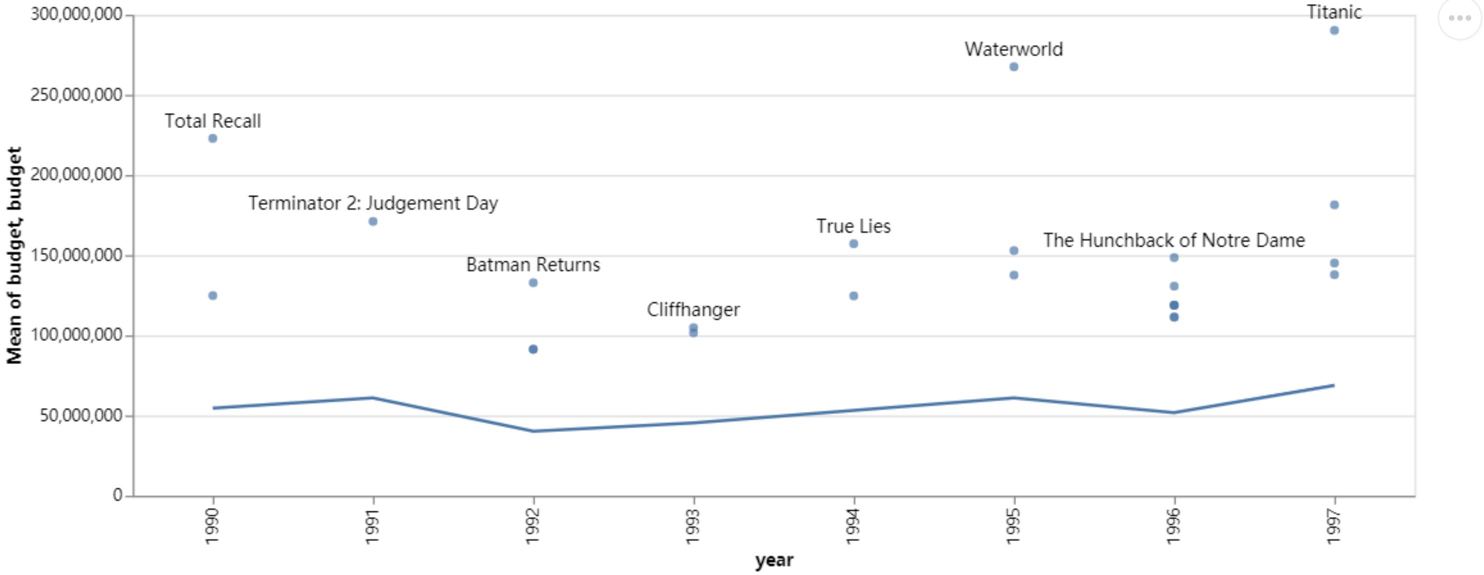
Replicate Vis 1

Hint:

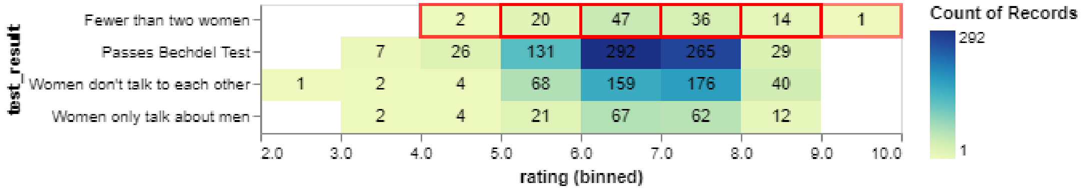
- Line chart is the simplest one, you can start from there.

- Think about the difference between transform_aggregate and transform_joinaggregate

```
#TODO: Replicate visualization 1
line_chart=alt.Chart(movies_test).transform_filter(
    alt.datum.year>=1990 & alt.datum.year<1998
).mark_line().encode(
    x="year:O",
    y="mean(budget)",
).properties(width=800)



dot_chart=alt.Chart(movies_test).transform_filter(
    alt.datum.year>=1990 & alt.datum.year<1998
).transform_joinaggregate(
    mean_budget="mean(budget)",
    groupby=["year"],
).transform_filter(
    alt.datum.budget>=2*alt.datum.mean_budget
).mark_point(filled=True).encode(
    x="year:O",
    y="budget",
)

text_chart=line_chart.transform_joinaggregate(
    max_budget="max(budget)",
    groupby=["year"],
).transform_filter(
    alt.datum.budget==alt.datum.max_budget,
).mark_text(align='center',dy=-10).encode(
    text=alt.Text('title')
)

line_chart + dot_chart + text_chart
```



Visualization 2 heatmap with additional annotations



Description of this visualization:

We want to produce a heatmap contrasting different types of bechdel test results to the IMDB scores of those movies. Each cell in the heatmap will tell us how many movies (not normalized here, we want raw count) are in that category. The author of the article wants to point out some property of the really well-regarded movies so wants those highlighted.

- Plot a heatmap with rating and test result. Encode the count of movies as color. Remove the category "dubious".
- For each cell, add text to indicate the count of movies.
- If a test_result category has at least one movie whose rating is higher than 9, highlight that entire category. You can highlight by adding a layer of heatmap that is not filled.

Visualization2 Plan:

TODO: edit this cell to write your visualization plan. You can write in altair syntax, in full sentence, or in bullet points, whichever way that helps you to plan your chart.

heatmap

- Describe the encoding rules:

  > x: rating:Q, bin=True
  >
  > y: test_result:N
  >
  > color: count():Q

- Describe the transformations:

  > filter: test_result != "dubious"

text annotation

- Describe the encoding rules:

  > x: rating:Q, bin=True
  >
  > y: test_result:N text: count():Q

- Describe the transformations:

  > filter: test_result != "dubious"

highlight

- Describe the encoding rules:

  > x: rating:Q, bin=True
  >
  > y: test_result:N

- Describe the transformations:

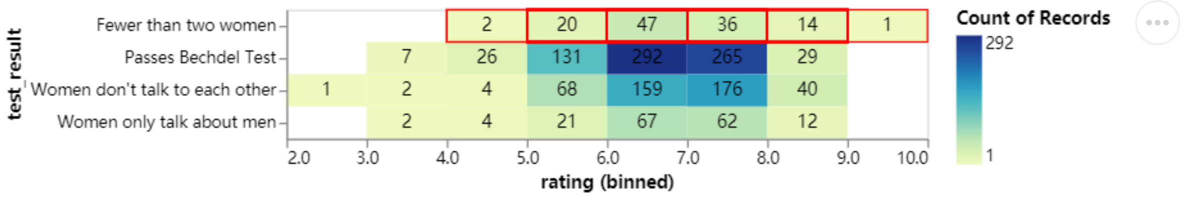
  > filter: test_result != "dubious"
  >
  > joinaggregate: max_rating="max(rating)", groupby=["test_result"]
  >
  > filter: max_rating>9

Replicate Vis 2

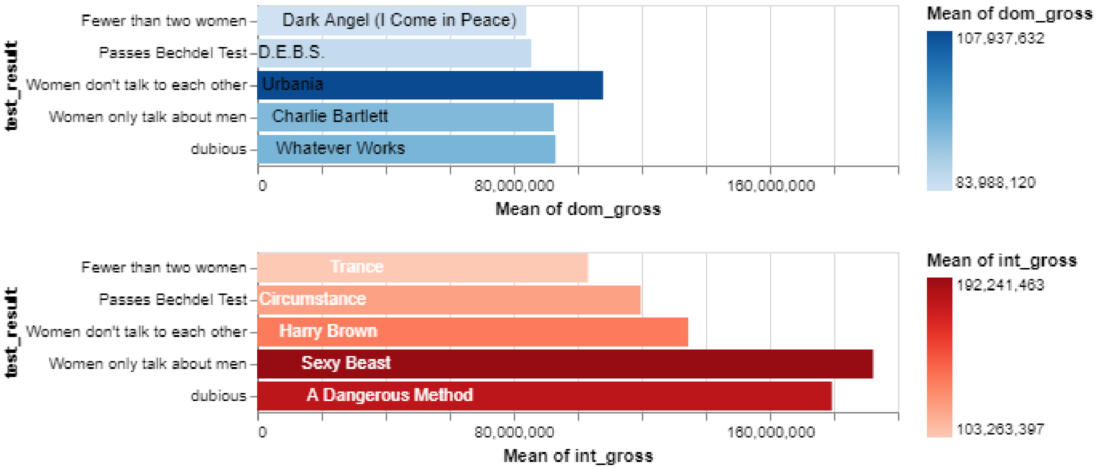
Hint:

- When you pass the heatmap to your text chart, you are passing all of the heatmap's encoding settings as well, which include the color setting.
- How do you translate "at least one movie with rating > 9 " into code?

```
#TODO: Replicate vis 2
heatmap=alt.Chart(movies_test).transform_filter(
    alt.datum.test_result != ″dubious″,
).mark_rect().encode(
        x=alt.X(″rating″,bin=True),
        y=’test_result’,
        color=’count()’
)

text_annotation=alt.Chart(movies_test).transform_filter(
    alt.datum.test_result != ″dubious″,
).encode(
    x=alt.X(″rating″,bin=True),
    y=’test_result’,
).mark_text(color=’black’).encode(
    text=alt.Text(’count()’),
)

highlight=text_annotation.transform_joinaggregate(
    max_rating=″max(rating)″,
    groupby=[″test_result″]
).transform_filter(
    alt.datum.max_rating>9,
).mark_rect(filled=False,color=″red″).encode(
    x=alt.X(″rating″,bin=True),
    y=’test_result’,
)



heatmap + text_annotation + highlight
```



Visualization 3: Bar chart with additional annotations



Description of this visualization: We want to contrast the domestic and international gross based on bedchel test passing category. We also want an example movie for each category so we're going to pick the 10th most popular one.

- Plot a bar chart *for movies made in US and Canada*, using the test result and mean of domestic gross.
- For test result category (i.e., each bar), find out the movie whose domestic gross ranks as the 10th lowest (sort by ascending order, the rank = 10). Annotate the title of this movie as text.
- BONUS: Plot a similar bar chart *for movies made internationally* using the international gross. Concatenate the international and domestic charts charts. Make sure they share the same x axis but have independent color scales.

Visualization3 Plan:

TODO: edit this cell to write your visualization plan. You can write in altair syntax, in full sentence, or in bullet points, whichever way that helps you to plan your chart.

bar chart

- Describe the encoding rules:

  > x: mean(dom_gross):Q
  > y: test_result:N
  > color: mean(dom_gross):Q

- Describe the transformations:

> filter: country_binary == "U.S. and Canada"

text annotation

- Describe the encoding rules:

> text: title:N

- Describe the transformations::

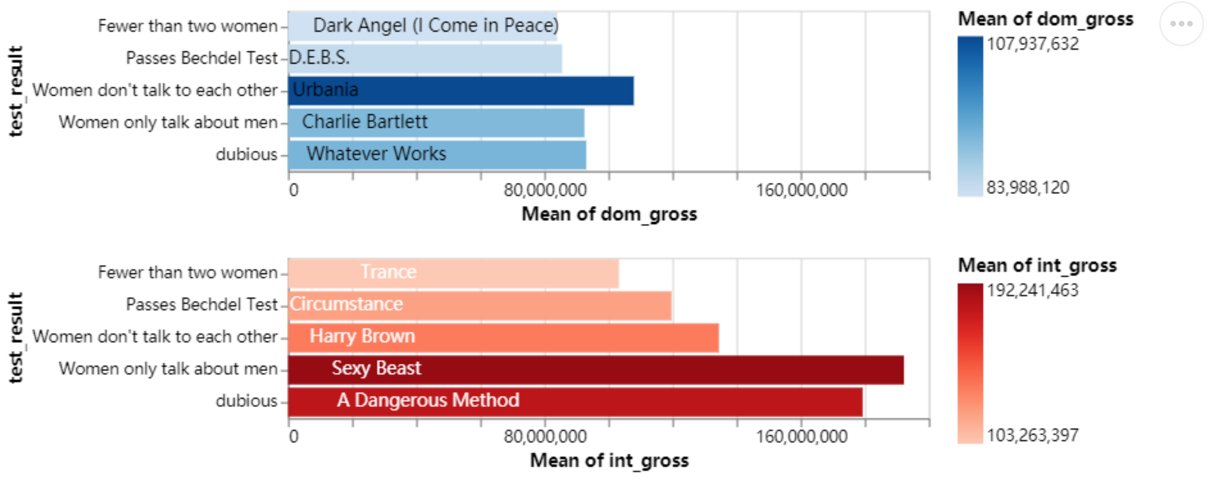
> filter: country_binary == "U.S. and Canada"
> window: rank='rank(dom_gross)', groupby=["test_result"], sort=[alt.SortField('dom_gross', order='ascending')]
> filter: rank == 10

Replicate Vis 3

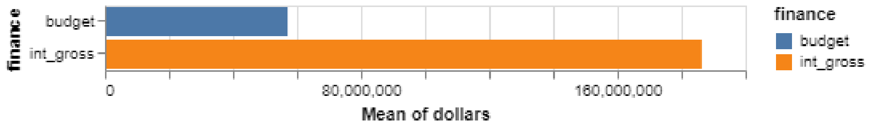
Hint:

- You want to generate the rank using a transformation method that we have covered in class.
- It's fine if text annotations don't align perfectly with your bars.
- There are several transformations in this task. If your chart looks slightly different from ours, you are still likely to get full points.

```
#TODO: Replicate Vis 3
bar_chart_dom=alt.Chart(movies_test).transform_filter(
        alt.datum.country_binary == "U.S. and Canada",
).mark_bar().encode(
        x='mean(dom_gross)',
        y='test_result',
        color='mean(dom_gross)'
)

text_dom=alt.Chart(movies_test).transform_filter(
        alt.datum.country_binary == "U.S. and Canada",
).encode(
        x='mean(dom_gross)',
        y='test_result',
).transform_window(
        rank='rank(dom_gross)',
        groupby=["test_result"],
        sort=[alt.SortField('dom_gross', order='ascending')]
).transform_filter(
        alt.datum.rank == 10
).mark_text(align='left').encode(
).mark_text(align='left').encode(
        text=alt.Text(’title’)
)

bar_chart_int=alt.Chart(movies_test).transform_filter(
        alt.datum.country_binary  !=  ”U.S.  and  Canada”,
).mark_bar().encode(
        x=’mean(int_gross)’,
        y=’test_result’,
        color=alt.Color(’mean(int_gross)’,  scale=alt.Scale(scheme=’reds’))
)

text_int=alt.Chart(movies_test).transform_filter(
        alt.datum.country_binary  !=  ”U.S.  and  Canada”,
).encode(
        x=’mean(int_gross)’,
        y=’test_result’,
).transform_window(
        rank=’rank(int_gross)’,
        groupby=[”test_result”],
        sort=[alt.SortField(’int_gross’,  order=’ascending’)]
).transform_filter(
        alt.datum.rank  ==  10
).mark_text(align=’left’,color=’white’).encode(
        text=alt.Text(’title’)
)


((bar_chart_dom  +  text_dom)  &  (bar_chart_int  +  text_int)).resolve_scale(x=”shared”,color=’independent’)
```



Visualization 4: Bar chart



Description of this visualization:

- Plot a bar chart that enables the comparison between mean budget and mean international gross across the whole dataset.
- Two bars should be colored differently

Visualization4 Plan:

TODO: edit this cell to write your visualization plan. You can write in altair syntax, in full sentence, or in bullet points, whichever way that helps you to plan your chart.

- What kind of transformation do I need to plot this bar chart?

> x: mean(dollars):Q
> y: finance:N
> color: finance:N

- What kind of encoding do I want to use?

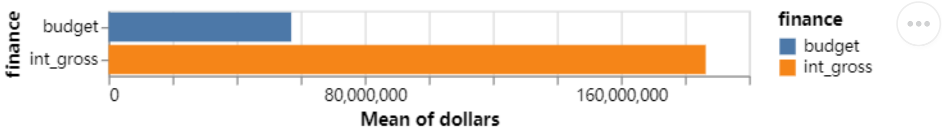
> fold: ["budget","int_gross"], as_=["finance","dollars"]

Replicate Vis 4

Hint:

- Do I need a long form or wide form data? Which type of data do I have?

```
# TODO: replicate vis 4
alt.Chart(movies_test).transform_fold(
        [”budget”,”int_gross”],
        as_=[”finance”,”dollars”]
).mark_bar().encode(
        y = alt.Y(’finance:N’),
        x = alt.X(’mean(dollars):Q’),
        color = alt.Color(’finance:N’)
        color = alt.Color('finance:N')
)
```



5. Ugly Visualization

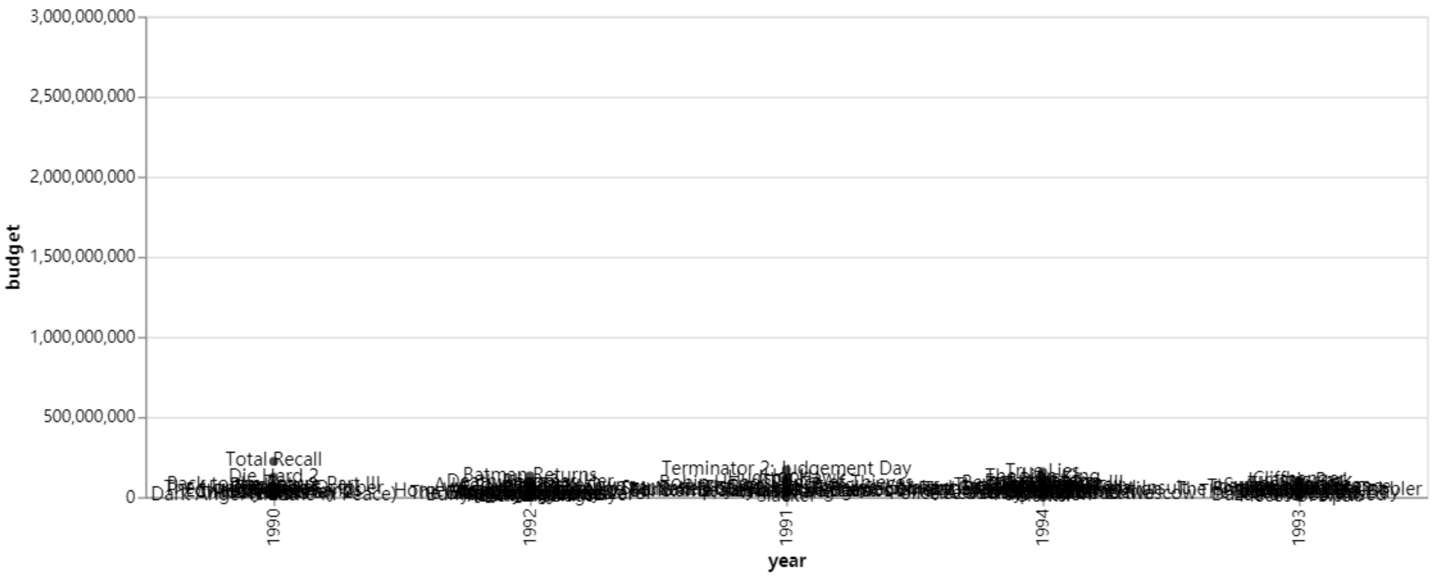
For this task, we're going to ask that you make the worst(!), ugliest, chart possible. Please use the movies dataset as a starting point, but you can add to it if you want.

You don't have to do it programatically. You can start with the Altair version or go back to Tableau, Excel, or whatever you want... You can download the chart and edit it in photoshop, illustrator, inkscape... any software you want to use. Please don't just turn in just a blank screen... :) Your chart should express **something** useful. Just do it badly.

After making the chart, please upload it to canvas (jpg or png or pdf).

When you're done crafting your terrible chart, use the following cell to tell us what perception rules you are violating that make your chart really bad!

```
dot_chart=alt.Chart(movies_test).transform_filter(
    alt.datum.year>=1990  &  alt.datum.year<1995
).mark_point(color="black",filled=True).encode(
    x=alt.X("year:N",sort=[1990,1992,1991,1994,1993]),
    y=alt.Y("budget:Q",scale=alt.Scale(domain=[0,  3000000000]))
).properties(width=800)

text_chart=dot_chart.mark_text().encode(
    text=alt.Text('title')
)

dot_chart  +  text_chart
```



TODO: Edit this cell: Write down perception/cognition rules that your chart is violating, and why you think this chart is ugly

The chart violates both the expressiveness principle and the effectiveness principle. The underlying order of the "year" data is not shown in the chart above, which violates the expressiveness principle. Also, the too large scale， too many text labels and points with the same color clustering together makes it alomost impossible to find out anything important from the chart. The effectiveness principle is thus violated. The messy labels and points makes it ugly.

This is the end of lab 3.

Please run all cells (Runtime->Restart and run all), and

1. save to PDF (File->Print->Save PDF -> landscape, shrink to 80%)
2. save to ipynb (File -> Download .ipynb)

Rename both files with your uniqname: e.g. uniqname.pdf/ uniqname.ipynb

Upload both files AND the ugliest chart to canvas