

▼ SI649-20-Winter Lab 2 -> Altair I

Overview

We're going to re-create some of the visualizations we did in Tableau but this time using Altair for the article: [“The Dollar-And-Cents Case Against Hollywood’s Exclusion of Women”](#). We'll be teaching you different pieces of Altair over the next few weeks so we'll focus on just a few visualizations this time:

1. Replicate 2 visualizations in the original article
2. Implementing 2 new visualizations according to our specifications

For this lab, we have done all of the necessary data transformation for you. You do not need to modify any dataframe. You only need to write Altair code. It's fine if your visualization looks slightly different from the example (e.g., getting 1.1 instead of 1.0)

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.

We encourage you to go through the Altair tutorials before next week:

- [UW Course](#)
- [Altair tutorial](#)

Resources

- [Altair Documentation](#)
- [Colab Overview](#)
- [Markdown Cheatsheet](#)
- [Pandas DataFrame Introduction](#)
- [Vega-Lite documentation](#)

- [Vega/Vega-Lite editor](#)

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

# load data and perform basic data processing
# get the CSV
datasetURL="https://raw.githubusercontent.com/LiciaHe/SI649/master/week2/movies_individual_task.csv"
movieDF=pd.read_csv(datasetURL, encoding="latin-1")

# fix the result column, rename the values
movieDF['test_result'] = movieDF['clean_test'].map({
    "ok": "Passes Bechdel Test",
    "men": "Women only talk about men",
    "notalk": "Women don't talk to each other",
    "nowomen": "Fewer than two women",
    "dubious": "dubious"
})

# fix the location column for later use
locationDict = defaultdict(lambda: 'International')
locationDict["United States"]="U.S. and Canada"
locationDict["Canada"]="U.S. and Canada"
movieDF["country_binary"]=movieDF["country"].map(locationDict)

##calculate ROI for 2nd chart
movieDF["roi_dom"]=movieDF["domgross_2013$"]/movieDF["budget_2013$"]
movieDF["int_only_gross"]=movieDF["intgross_2013$"]-movieDF["domgross_2013$"]
movieDF["roi_int"]=movieDF["int_only_gross"]/movieDF["budget_2013$"]

movieDF=movieDF.drop(columns=["Unnamed: 0", "test", "budget", "domgross", "intgross", "code", "period code", "decade code", "director", "genre", "di
movieDF_since_1990=movieDF[movieDF.year>1989]

#take a look at the new dataset
movieDF.sample()
movieDF_since_1990.sample(3)
```

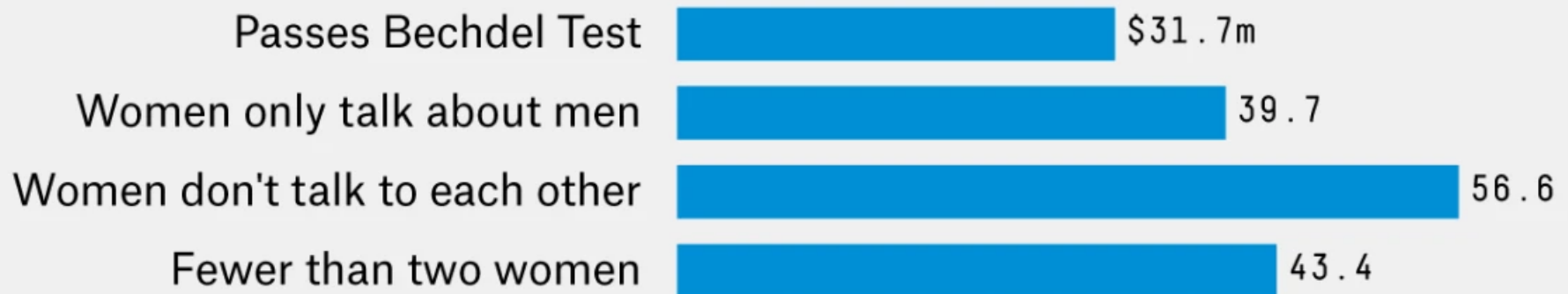


	year	title	clean_test	binary	budget_2013\$	domgross_2013\$	intgross_2013\$	rating	country	language	test_result	co
712	2007	St. Trinian's	ok	PASS	12808396	16853.0	25219695.0	5.8	United Kingdom	English	Passes Bechdel Test	
937	2004	Bridget Jones: The Edge of Reason	ok	PASS	61659644	49578078.0	325432881.0	6.0	United Kingdom	English	Passes Bechdel Test	
1181	2001	Lara Croft: Tomb Raider	nowomen	FAIL	123689520	172565543.0	361389438.0	5.8	United States	English	Fewer than two women	U.

▼ Visualization 1: Recreate this visualization

Median Budget For Films Since 1990

2013 dollars



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

- Data Name: *movieDF_since_1990*
- mark type: `mark_bar()`
- Encoding Specification:
 - `x:median(budget_2013$):Q`
 - `y: test_result:N`

Example encoding, if we had the nominal variable 'movietype' and we wanted to use color, it would be:

`color: movietype:nominal`

▼ Step 2: Create your chart.

Please take a look at the checkpoints below. You can follow the checkpoint to work through the problem step-by-step. Don't forget to 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

#TODO: Replicate visualization 1

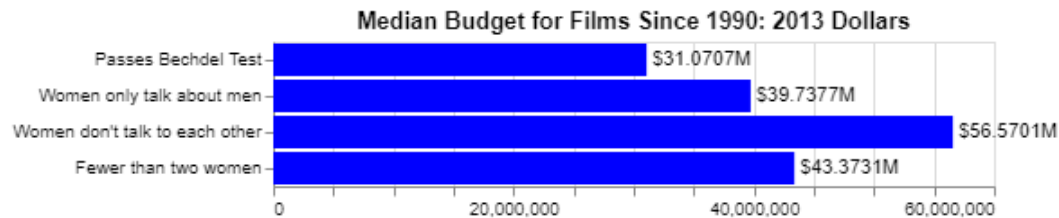
```
bar= alt.Chart(movieDF_since_1990,title="Median Budget for Films Since 1990: 2013 Dollars").mark_bar(color="blue").encode(
  x= alt.X("median(budget_2013$):Q", title= ""),
  y=alt.Y("test_result:N", title= "",
          sort= ["Passes Bechdel Test","Women only talk about men","Women don't talk to each other","Fewer than two women"]
  )).transform_filter(
  alt.FieldOneOfPredicate(field='test_result', oneOf=["Passes Bechdel Test","Women only talk about men","Women don't talk to each ot
)

text = bar.mark_text(
  align='left',
  baseline='middle',
  dx=3 # Nudges text to right so it doesn't appear on top of the bar

).encode(
  text= alt.Text('median(budget_2013$):Q', format = "$~s" )
```

)

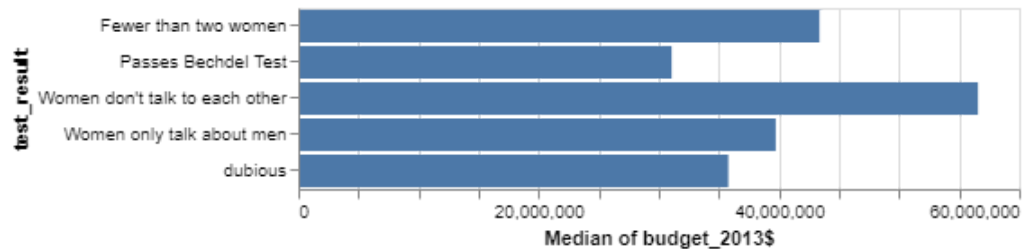
(bar+text)



checkpoint 1: basic bar chart: you get full points if you

- Specify the correct mark
- Use the correct x and y encoding
- Plotting the right data (hint: make sure you examine the data frame and use the correct columns)

You chart should look like:

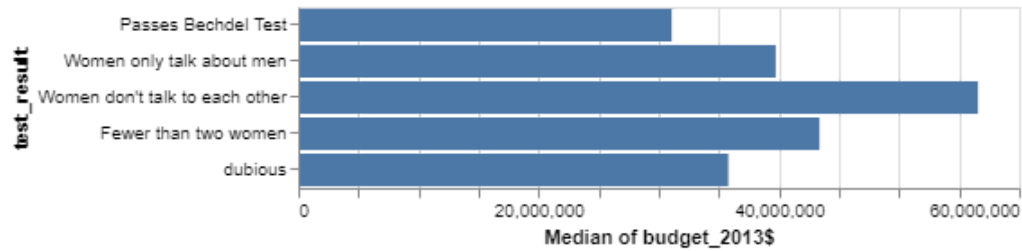


checkpoint 2: basic bar chart with sorted order: you get full points if you

- Completed checkpoint1
- Align the order of your y-axis values with the provided example.
- *i.e., from top to bottom, the order of the bars is "Passes Bechdel Test","Women only talk about men","Women don't talk to each other","Fewer than two women","dubious".*

Hint: [Sort](#)

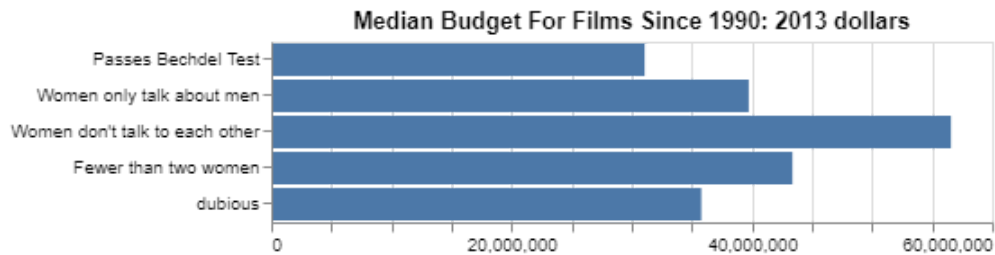
You chart should look like:



checkpoint 3: basic bar chart with title: you get full points if you

- Completed checkpoint2
- Remove labels on x-axis and y-axis
- Add a chart title

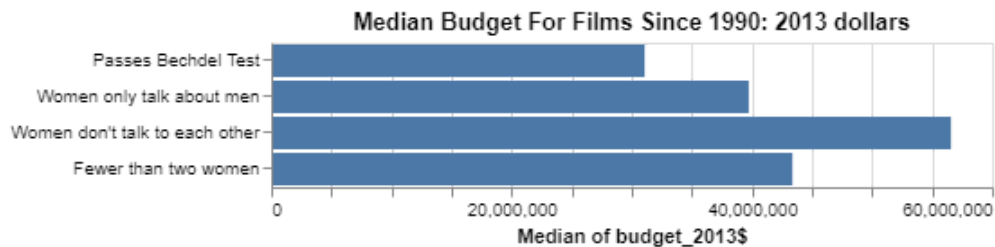
You chart should look like:



checkpoint 4: BONUS: remove dubious. You will get full point if you

- Complete checkpoint 3
- Remove the bar for "dubious" (using Altair, no Pandas)

You chart will look like:

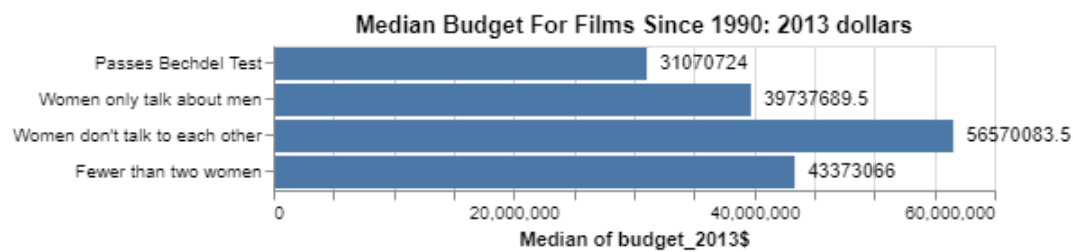


checkpoint 5: BONUS: add number labels.

You will get full point if you

- Complete checkpoint 4
- Add number as labels of your bars

Your chart will look like:

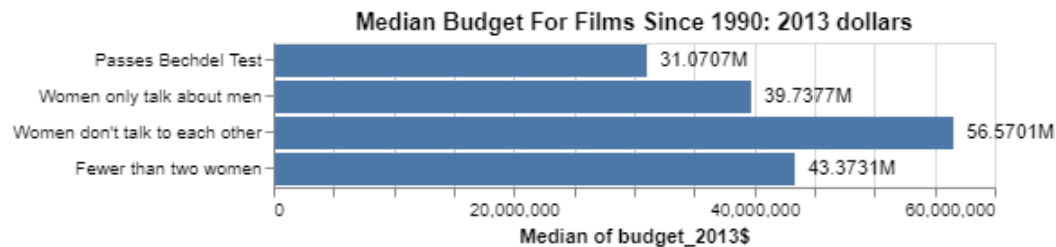


checkpoint 6: BONUS: format numbers.

You will get full points if you

- Complete checkpoint 5
- Adjust number labels to display millions. e.g. (31.4592 M instead of 31459218). You might want to read about [format](#), and [D3's format specification](#).

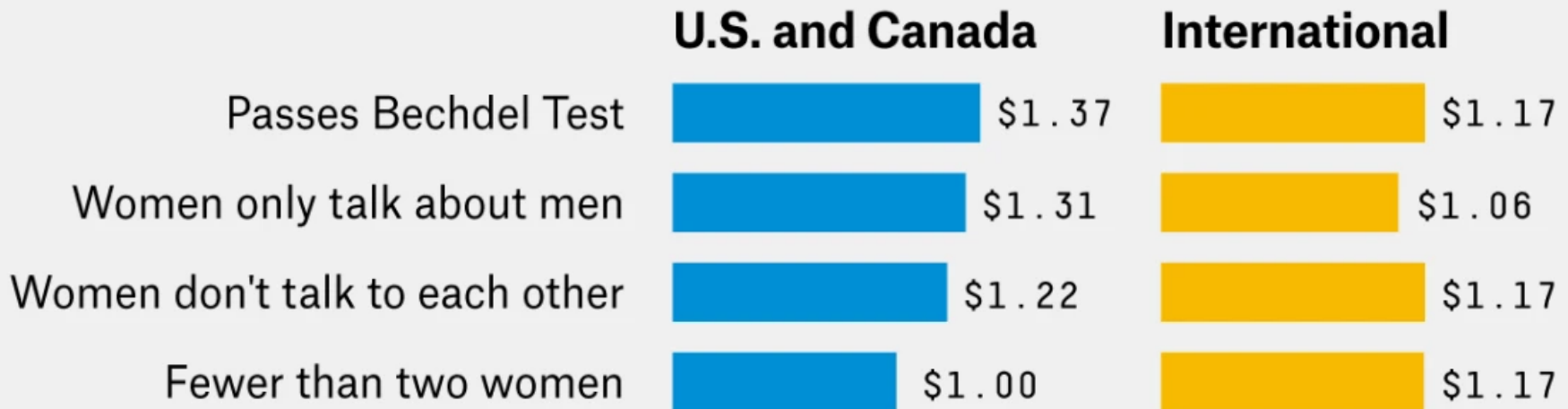
Your chart will look like:



▼ Visualization 2 Replicate this visualization

Dollars Earned for Every Dollar Spent

2013 dollars



FIVETHIRTYEIGHT

SOURCE: BECHDELTEST.COM, THE-NUMBERS.COM

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

Left chart:

- Data Name: *movieDF*
- mark type: *mark_bar*
- Encoding Specification:
 - `x:median(roi_dom):Q`
 - `y:test_result:N`

Right chart:

- Data Name: *movieDF*

- mark type: *mark_bar*
- Encoding Specification:
- `x:median(roi_int):Q`
- `y:test_result:N`

Compound Method (how to join these charts together?): *Horizontal: alt.hconcat(domestic+text2, international+text3) or Vertical: alt.vconcat(domestic+text2, international+text3)*

Example encoding, if we had the nominal variable 'movietype' and we wanted to use color, it would be:
color: movietype:nominal

▼ Step 2: Create your chart.

Please take a look at the checkpoints below. You can follow the checkpoint to work through the problem step-by-step. Don't forget to 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

#TODO: Replicate chart 2

```
domestic = alt.Chart(movieDF,title="U.S and CANADA").mark_bar(color="blue").encode(

    x= alt.X("median(roi_dom):Q", title= "" ),
    y=alt.Y("test_result:N", title= "",
        sort= ["Passes Bechdel Test","Women only talk about men","Women don't talk to each other","Fewer than two women","dubious"]

    )

    )
international = alt.Chart(movieDF,title="INTERNATIONAL").mark_bar(color="orange").encode(
    x= alt.X("median(roi_int):Q",title= "" ),
    y=alt.Y("test_result:N",
        axis=None,
        sort= ["Passes Bechdel Test","Women only talk about men","Women don't talk to each other","Fewer than two women","dubious"]

    ))
```

```

text2 = domestic.mark_text(

    align='left',
    baseline='middle',
    dx=3 # Nudges text to right so it doesn't appear on top of the bar

).encode(

    text= alt.Text('median(roi_dom):Q', format= '.5f')

)
text3 = international.mark_text(

    align='left',
    baseline='middle',
    dx=3 # Nudges text to right so it doesn't appear on top of the bar

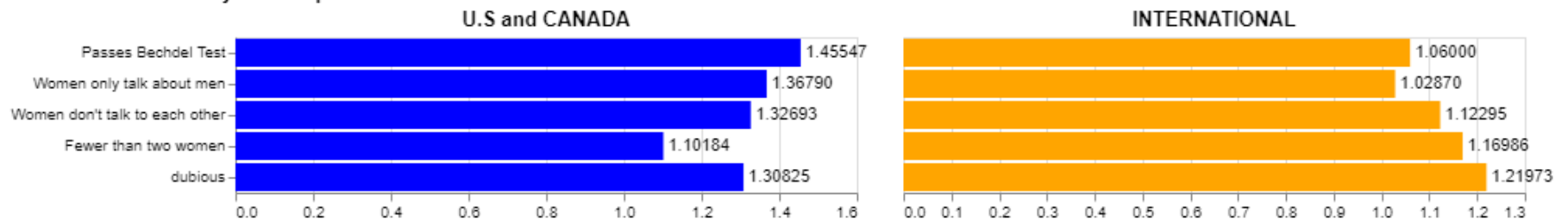
).encode(
    text=alt.Text('median(roi_int):Q',format= '.5f')

)

combined= alt.hconcat(domestic+text2, international+text3).resolve_scale(y="shared")
combined.properties(title="Dollar Earned for Every Dollar Spent")

```

➡ Dollar Earned for Every Dollar Spent

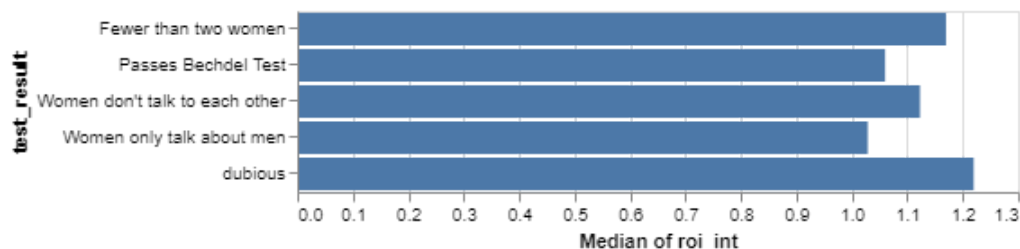
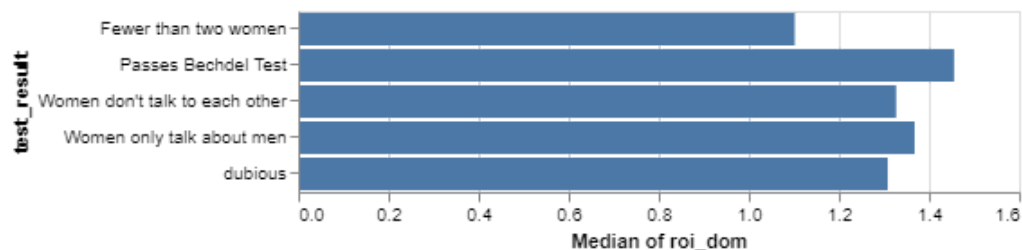


▼ Visualization 2 Checkpoints

checkpoint 1: basic bar charts

- Specify the correct mark
- Use the correct x and y encoding
- Plotting the right data (hint: make sure you examine the data frame and use the correct columns)
- You will have 2 charts, one for U.S.&Canada, one for International

You chart will look like:

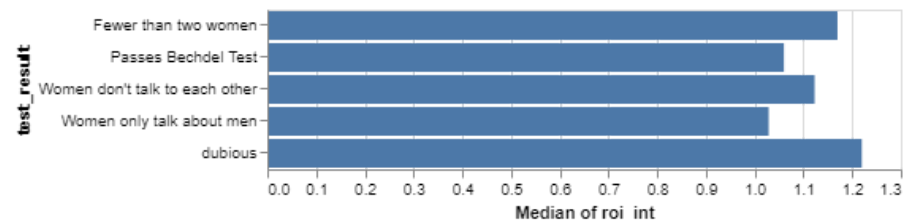
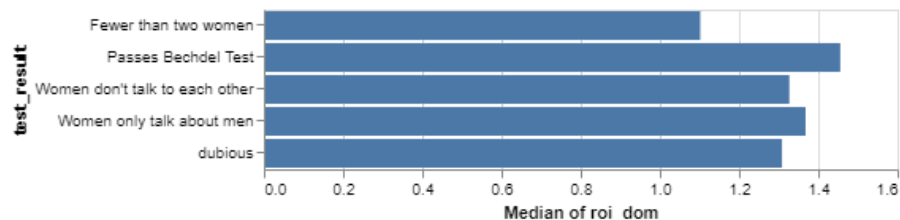


and

checkpoint 2: joining two charts

- completed checkpoint1
- joined two charts

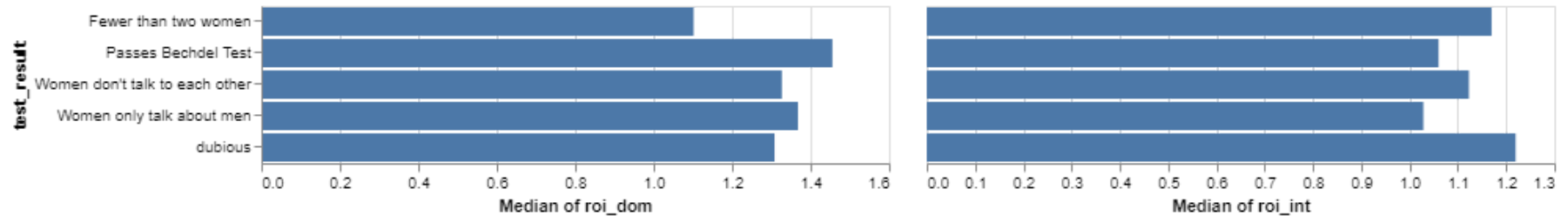
You chart will look like:



checkpoint 3: resolve y scale and hide the second y-axis

- completed checkpoint2
- ensure that two charts are sharing the same y-axis
- remove the second y-axis

You chart will look like:

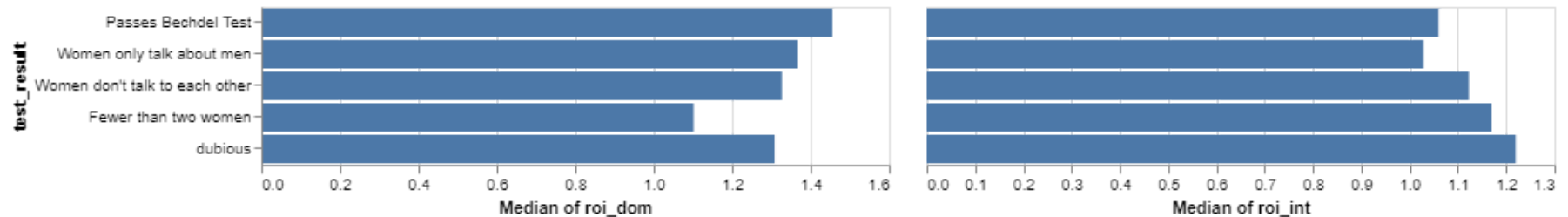


checkpoint 4: sort y-axis

- completed checkpoint 3
- Sort y-axis so that the order of the bars is (from top to bottom):

"Passes Bechdel Test","Women only talk about men","Women don't talk to each other","Fewer than two women","dubious"

You chart will look like:

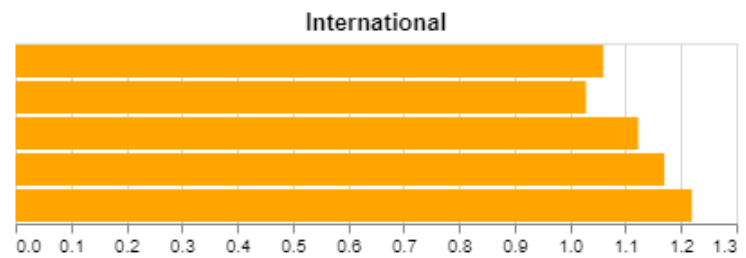
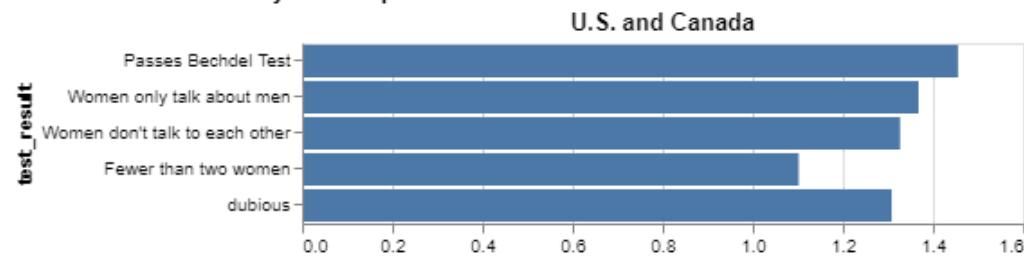


checkpoint 5: Change color and titles

- completed checkpoint 4
- color bars of these two charts with different colors
- add title to the compound chart
- edit axis labels (you can also remove axis label and add chart title to individual chart)

You chart will look like:

Dollars Earned for Every Dollar Spent

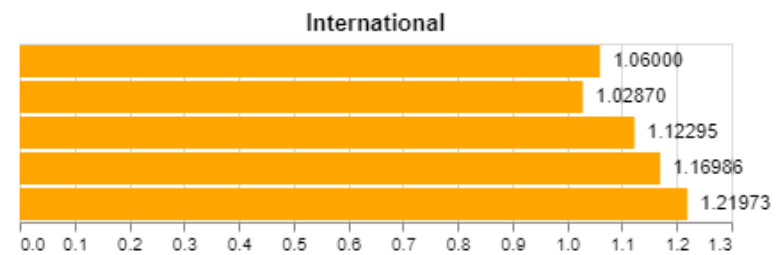
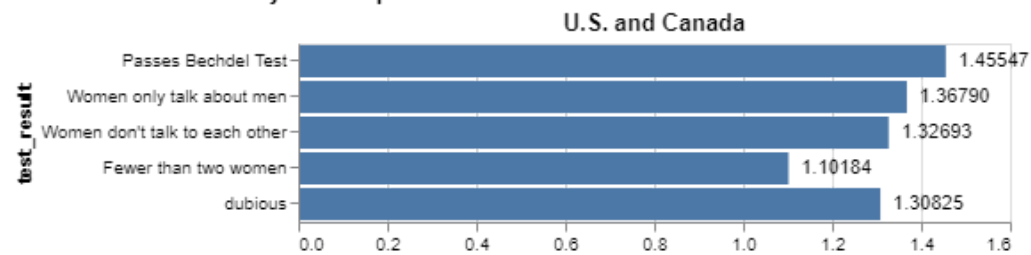


checkpoint 6: BONUS: Add number layer

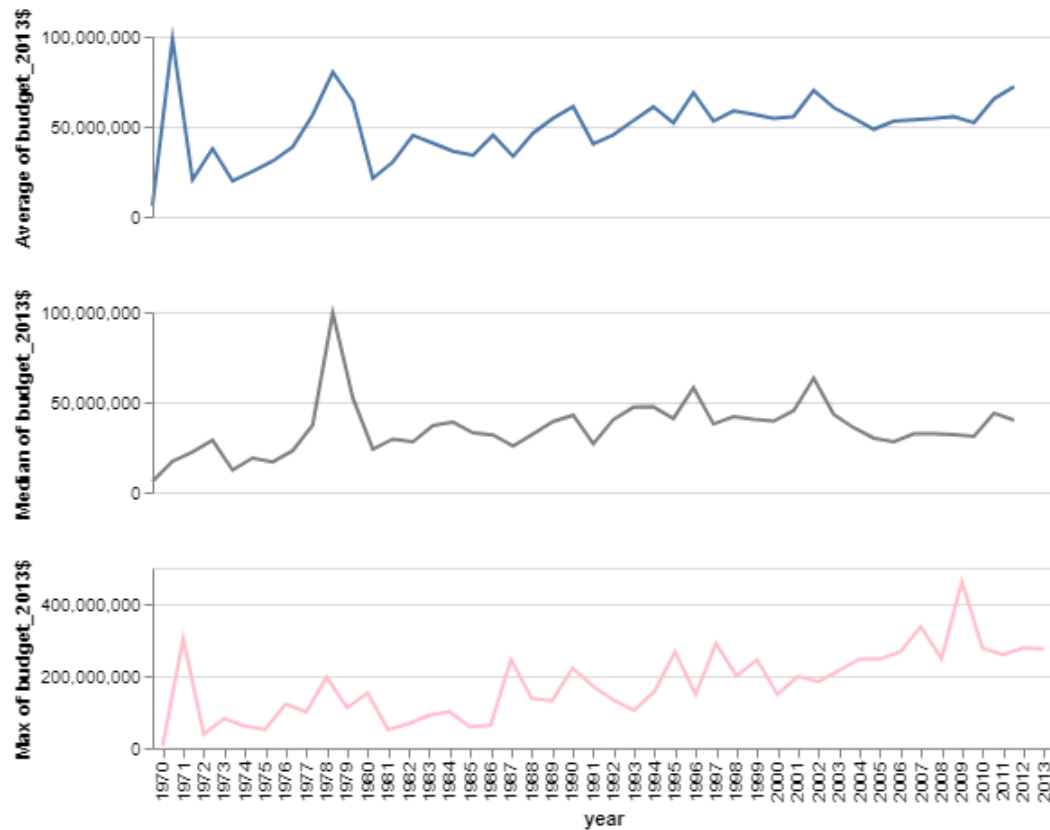
- completed checkpoint 5
- add number annotations

You chart will look like:

Dollars Earned for Every Dollar Spent



▼ Visualization 3: Replicate this visualization



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

- Data Name: *movieDF*
- mark type: *mark_line*
- Encoding Specification (1st chart):
 - *x:year:O*
 - *y: average(budget_2013\$):Q*
- Encoding Specification (2nd chart):
 - *x:year:O*

- `y: median(budget_2013$):Q`
- Encoding Specification (3rd chart):
- `x:year:0`
- `y: max(budget_2013$):Q`

▼ Step 2: Create your chart.

Please take a look at the checkpoints below. You can follow the checkpoint to work through the problem step-by-step. Don't forget to 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

```
#TODO: Replicate visualization 3
average = alt.Chart(movieDF, title = "Average").mark_line(color="blue").encode(
  x=alt.X("year:0", axis=None),
  y = alt.Y("average(budget_2013$):Q", axis=alt.Axis(tickDash=[50000000,100000000], tickColor="red"))

  ).properties(
    width=500,
    height=100

  )
median = alt.Chart(movieDF, title = "Median").mark_line(color="grey").encode(

  x=alt.X("year:0", axis=None),
  y = alt.Y("median(budget_2013$):Q", axis=alt.Axis(tickDash=[50000000,100000000], tickColor="red")),

  ).properties(
    width=500,
    height=100
  )
maximum = alt.Chart(movieDF,title = "Max").mark_line(color="pink").encode(

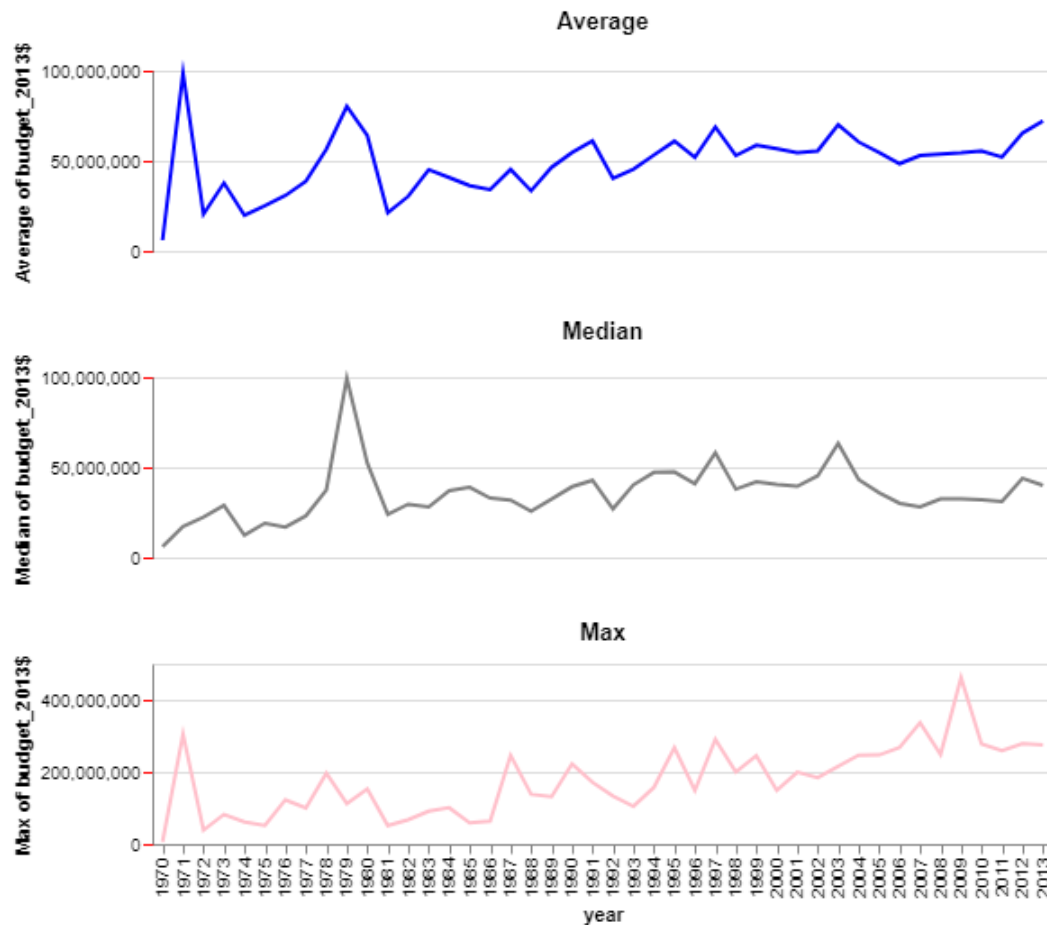
  x=alt.X("year:0"),
  y = alt.Y("max(budget_2013$):Q", axis=alt.Axis(tickDash=[200000000,400000000], tickColor="red")),

  ).properties(
```

```
width=500,
height=100
)
```

```
combined= alt.vconcat(average,median, maximum).resolve_scale( x="shared", y= "independent")
combined.properties(title="Visualization 3")
```

Visualization 3



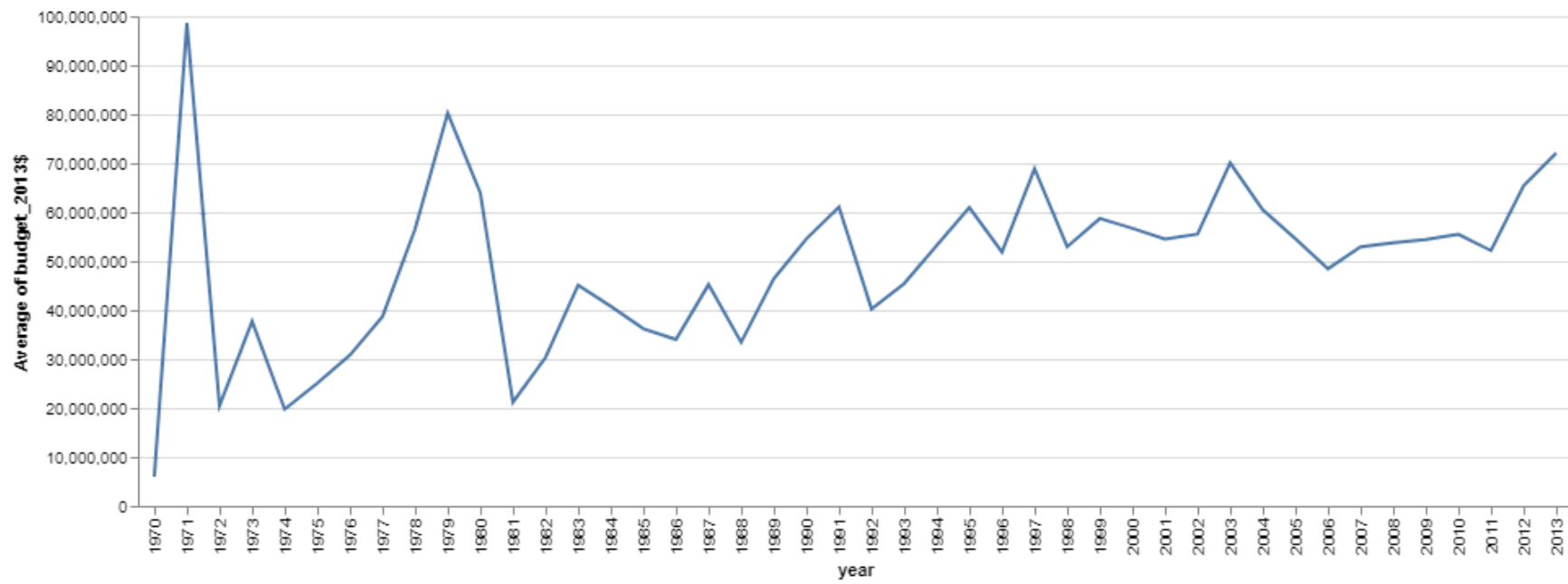
Visualization3 Checkpoints

checkpoint 1: line chart for average, median, and max of budget

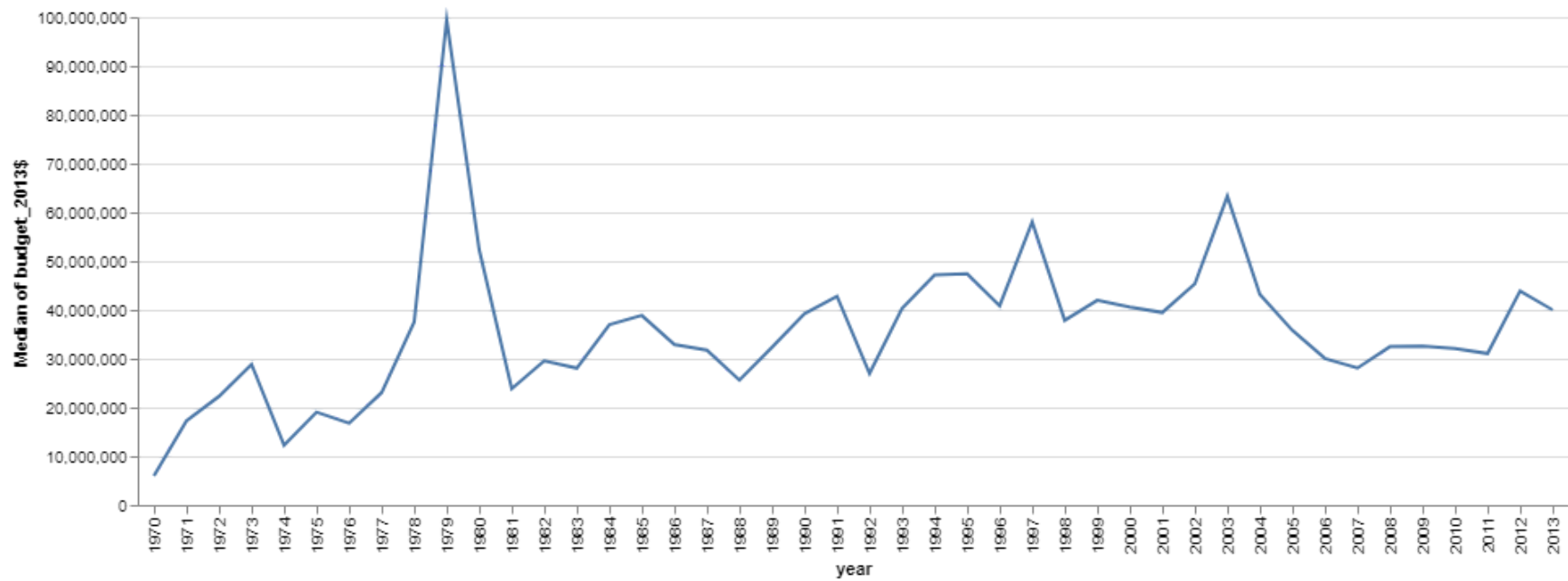
You will get full points if you

- Specify the correct mark
- Use the correct x and y encoding
- Plotting the right data
- Produce 3 line charts

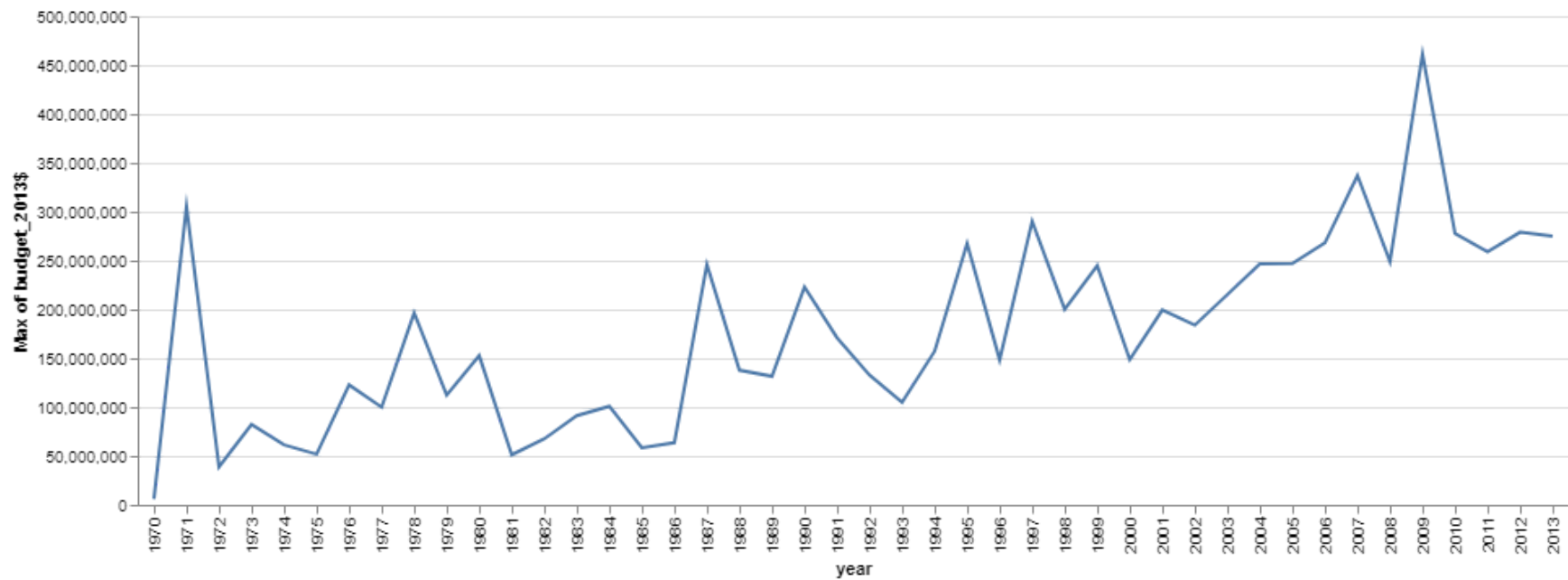
Your chart will look like:



and



and

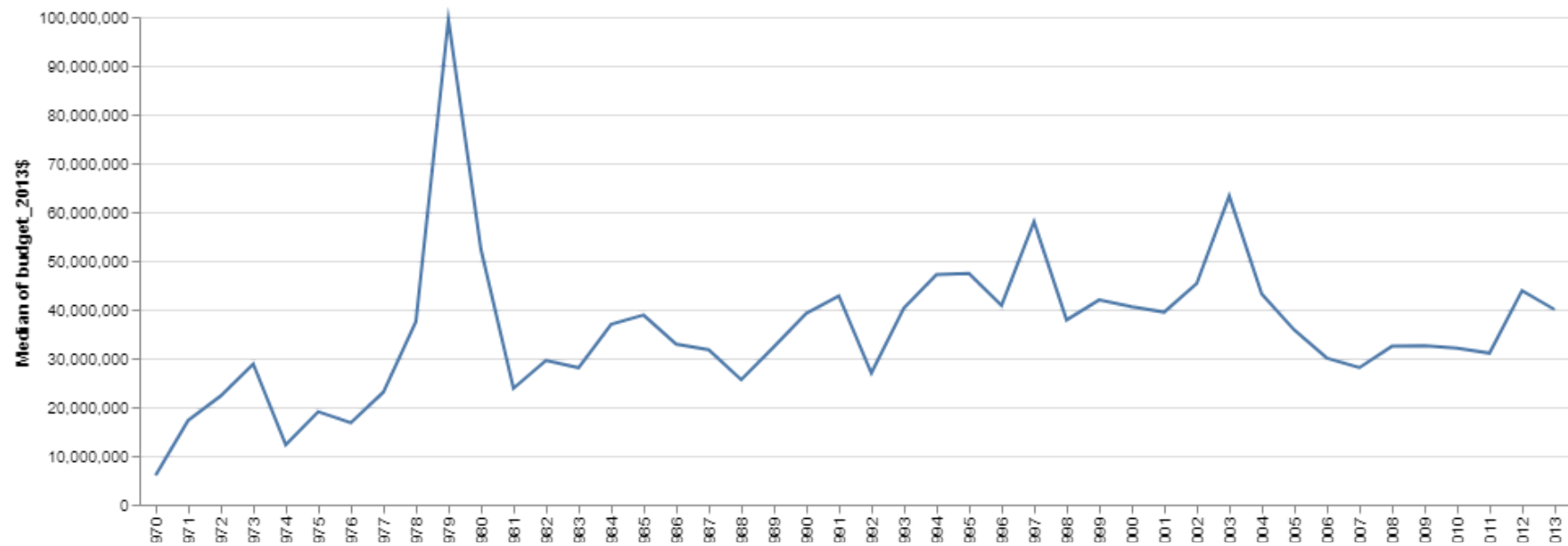
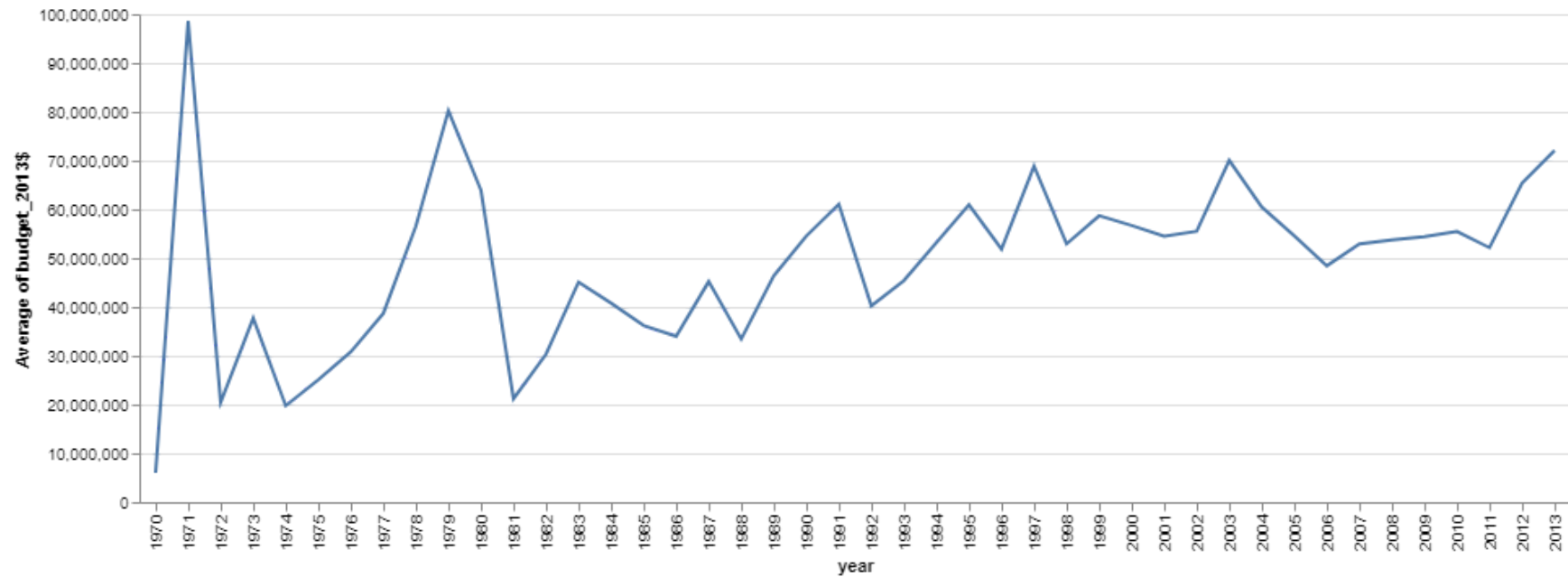


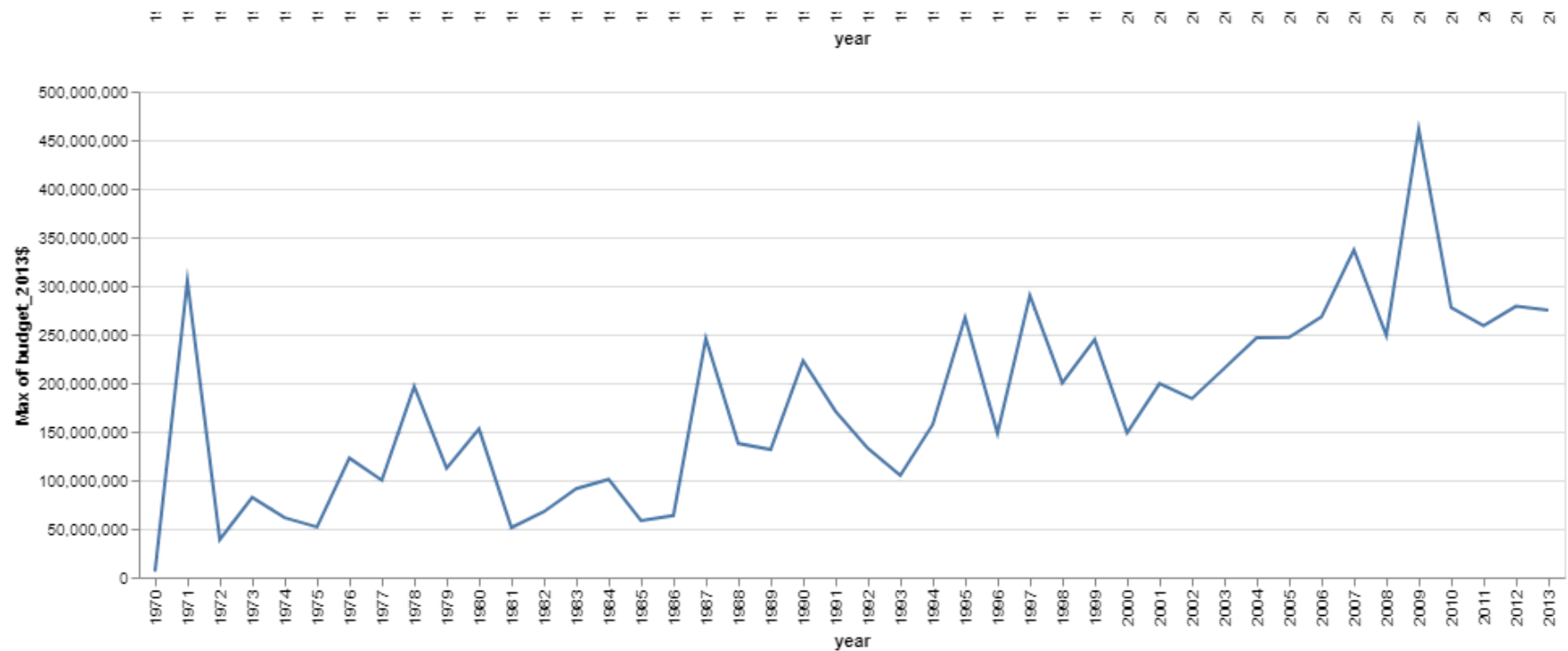
checkpoint 2: concat 3 line charts

You will get full points if you

- Complete checkpoint 1
- Concat 3 charts vertically

You chart will look like:



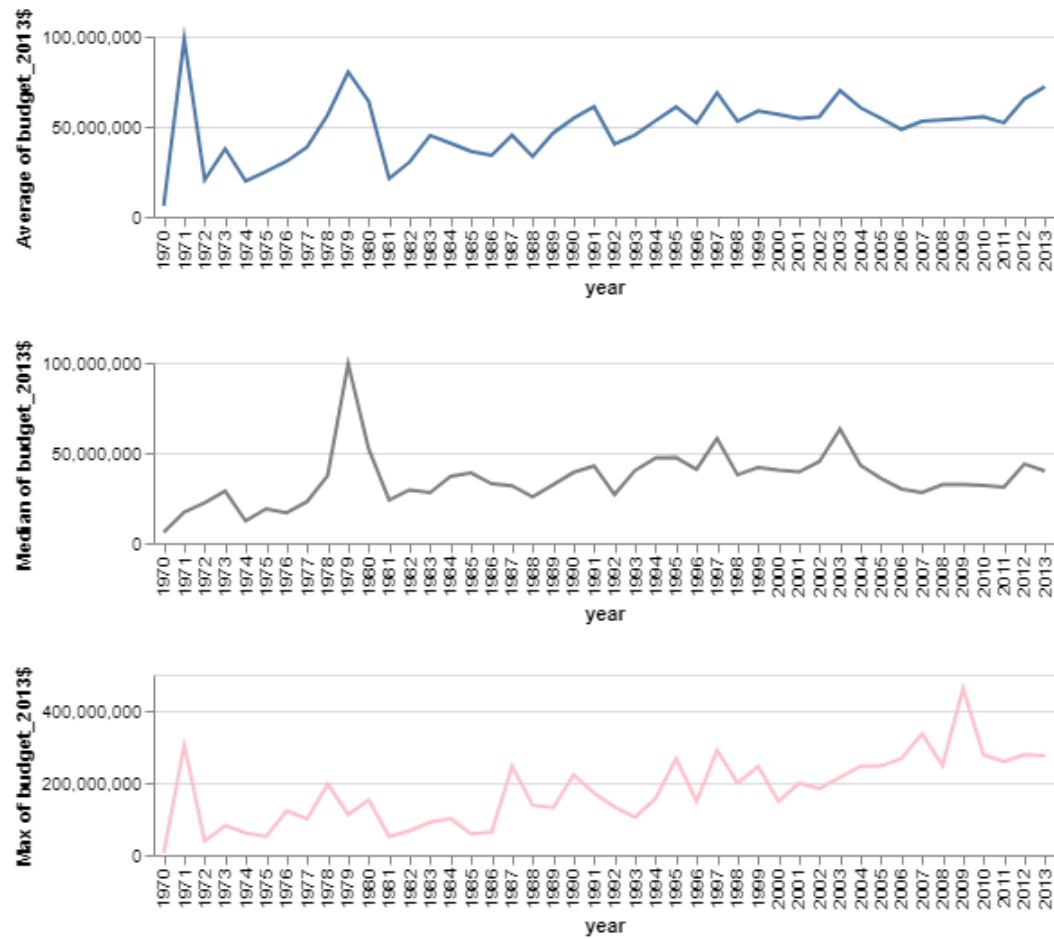


checkpoint 3: adjust width, height and color

Each chart should be 500x100, plotted with different colors

You will get full points if you

- Complete checkpoint 2
- Adjust chart width and height
- Plot charts with different colors



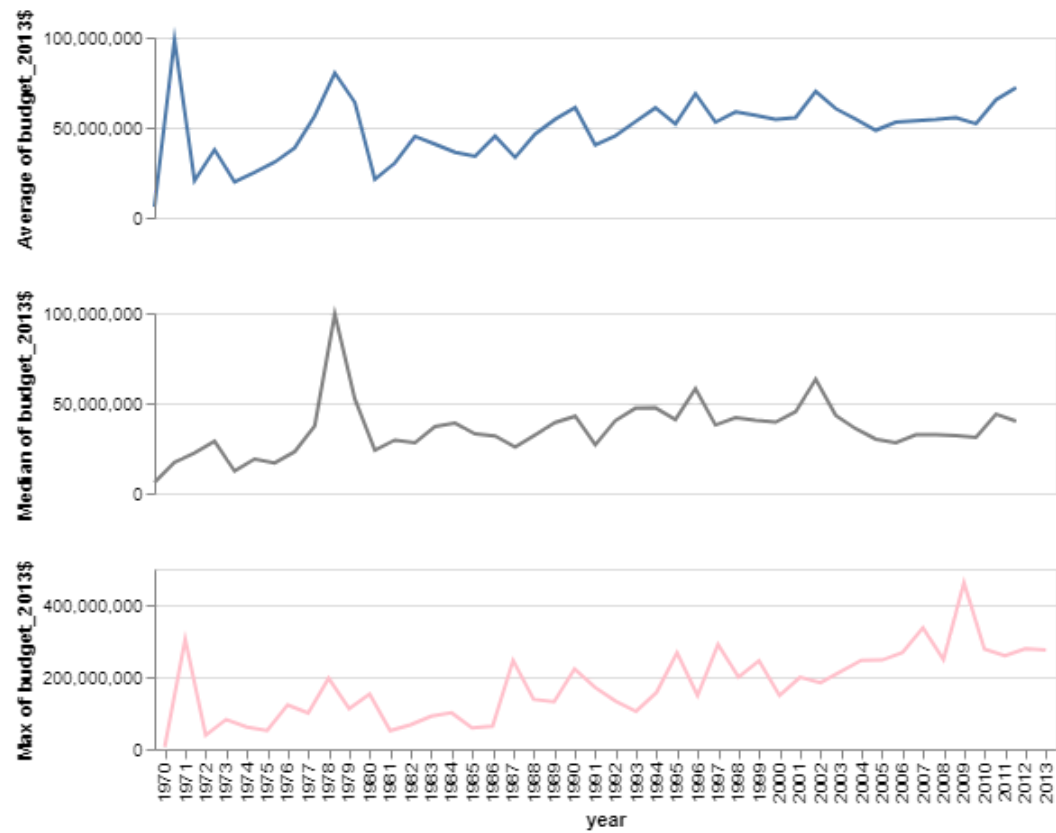
You chart will look like:

checkpoint 4: resolve axis and remove duplicated x-axis

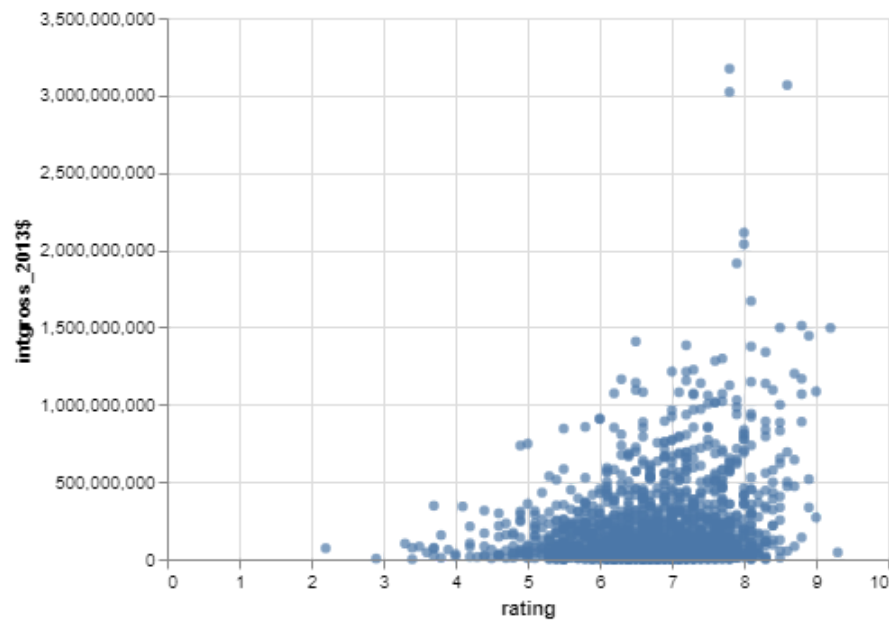
You will get full points if you

- Complete checkpoint 3
- Ensure that 3 charts are sharing the same x-axis
- Remove duplicate axis ticks.

You chart will look like:



- ▼ Visualization 4: Replicate this visualization



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

- Data Name: *movieDF*
- mark type: *mark_point*
- Encoding Specification:
 - `x:rating:Q`
 - `y:intgross_2013$:Q`

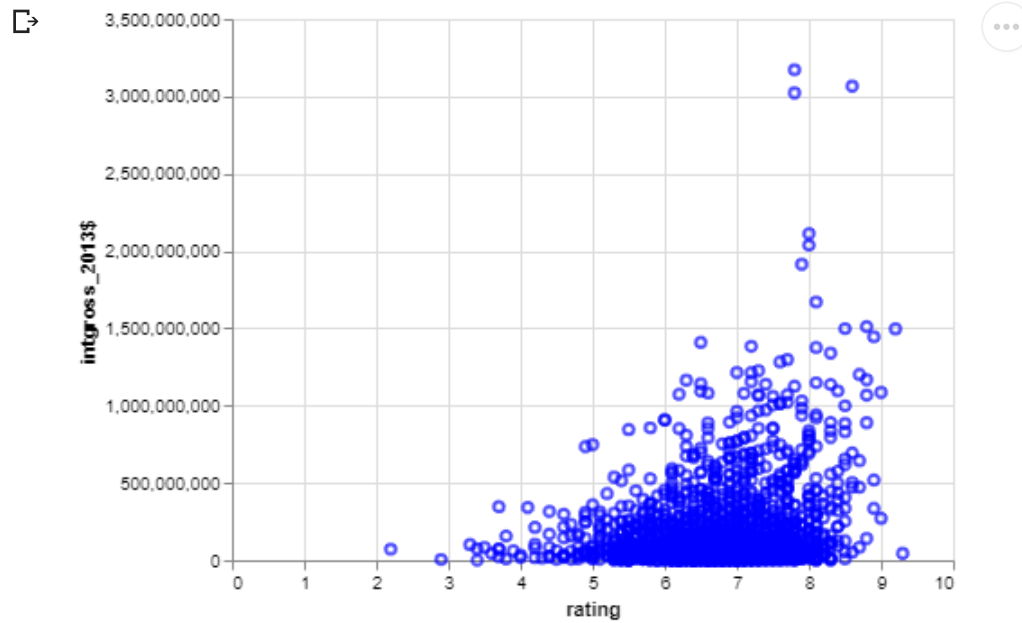
▼ Step 2: Create your chart.

This chart is relatively simple so there's no checkpoint.

#TODO: Replicate visualization 4

```
scatterPlot = alt.Chart(movieDF).mark_point(color="blue").encode(  
  x= alt.X("rating:Q"),  
  y=alt.Y("intgross 2013$:Q")
```

```
))  
scatterPlot
```



End of LAB2

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

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

Rename both files with your username: e.g. username.pdf/ username.ipynb Upload both files to canvas.

