SI649-22 Fall Lab 3 -> 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" (https://fivethirtyeight.com/features/the-dollar-and-cents-case-against-hollywoods-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.

TIPS: You can search through this file for "TODO" to see the parts you need to complete.

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

- UW Course (https://github.com/uwdata/visualization-curriculum)
- Altair tutorial (https://github.com/altair-viz/altair-tutorial)

Resources

- Altair Documentation (https://altair-viz.github.io/index.html)
- Colab Overview (https://colab.research.google.com/notebooks/basic_features_overview.ipynb)
- Markdown Cheatsheet (https://www.markdownguide.org/cheat-sheet/)
- Pandas DataFrame Introduction (https://pandas.pydata.org/pandas-docs/stable/user_guide/10min.html#min)
- Vega-Lite documentation
- Vega/Vega-Lite editor

In [1]:
```
# imports we will use
import altair as alt
import pandas as pd
from collections import defaultdict
alt.renderers.enable('html') #run this line if you are running jupyter notebook
```

Out[1]:
```
RendererRegistry.enable('html')
```

In [2]:
```
# load data and perform basic data processing
# get the CSV
datasetURL="https://raw.githubusercontent.com/eytanadar/si649public/master/lab3/data/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)
```

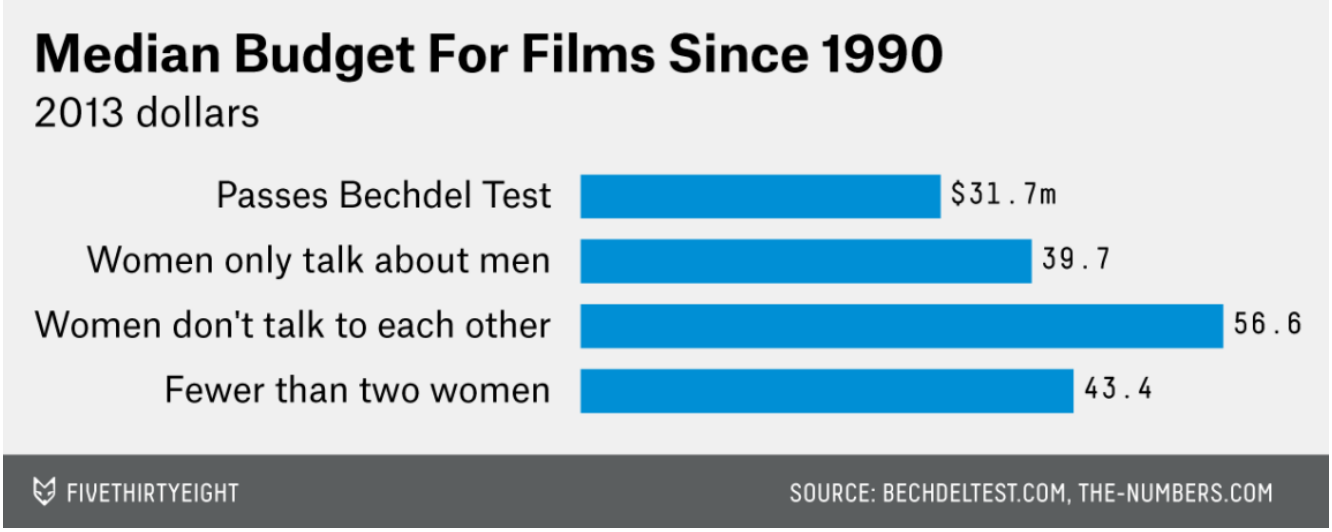
In [3]:
```

##calculate ROI (Return on Investment) 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","di
movieDF_since_1990=movieDF[movieDF.year>1989]
```

In [4]:
```
#take a look at the new dataset
movieDF.sample(3)
# movieDF_since_1990.sample(3)
```

Out[4]:

	year	title	clean_test	binary	budget_2013$	domgross_2013$	intgross_2013$	rating	country	language	test_result	country_binary	roi_dom	int_
920	2005	Walk the Line	ok	PASS	34597961	142590606.0	223941260.0	7.9	United States	English	Passes Bechdel Test	U.S. and Canada	4.121359	
1449	1996	Broken Arrow	nowomen	FAIL	96535720	104920957.0	220318272.0	6.0	United States	English	Fewer than two women	U.S. and Canada	1.086861	1
1473	1996	Striptease	dubious	FAIL	74258246	49173429.0	168283656.0	4.4	United States	English	dubious	U.S. and Canada	0.662195	1

Visualization 1: Recreate this visualization



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

- Data Name: *movieDF_since_1990*
- mark type: bar
- Encoding Specification:
  - x:median(budget_2013$):Quantitative
  - y: clean_test:Nominal

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

color: movietype:nominal

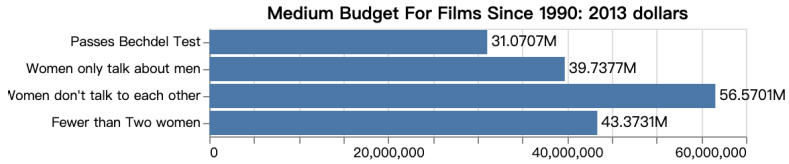
Step 2: Create your chart.

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 easier).

In [5]:
```
from altair import datum

#TODO: Replicate visualization 1
names = {'men': 'Women only talk about men', 'dubious': 'dubious', 'notalk': "Women don't talk to each other", 'nowomer
movieDF_since_1990['clean_test'] = movieDF_since_1990['clean_test'].replace(names)
movieDF_since_1990


bar = alt.Chart(movieDF_since_1990).mark_bar(filled=True).encode(
    alt.Y('clean_test:N', sort=['Passes Bechdel Test', 'Women only talk about men',  "Women don't talk to each other",
    alt.X('median(budget_2013$):Q', title=None)
).transform_filter((datum.clean_test != 'dubious'))
text = bar.mark_text(
    align='left',
    baseline='middle',
    dx=3  # Nudges text to right so it doesn't appear on top of the bar
).encode(
    text=alt.Text('median(budget_2013$):Q', format='.6s')
)

(bar + text).properties(title='Medium Budget For Films Since 1990: 2013 dollars')
```

```
/var/folders/wd/7mmm5v2n75b_cxpswz3tf4100000gn/T/ipykernel_28459/3830119292.py:5: 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)
  movieDF_since_1990['clean_test'] = movieDF_since_1990['clean_test'].replace(names)
```

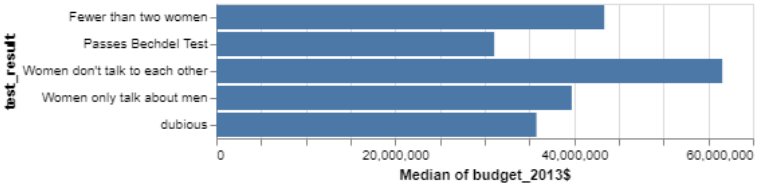
Out[5]:



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:

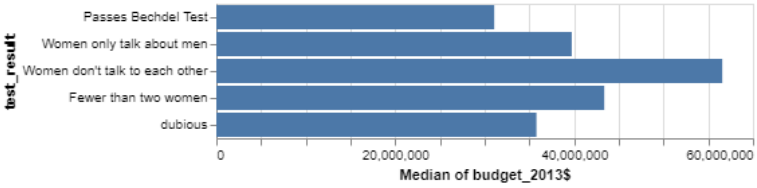


checkpoint 2: basic bar chart with sorted order: you get full points if you

- 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 (https://altair-viz.github.io/user_guide/generated/core/altair.Sort.html?highlight=sort)

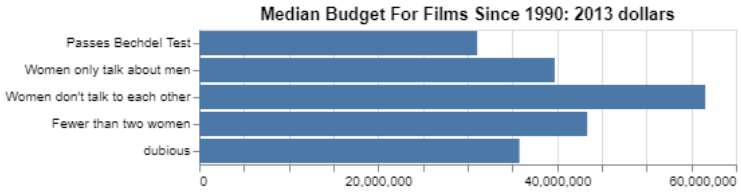
You chart should look like:



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

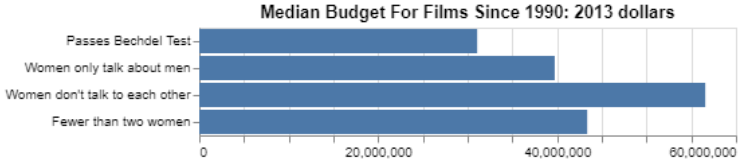
You chart should look like:



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:

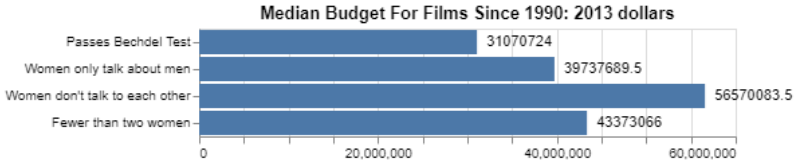


checkpoint 5: BONUS: add number labels.

You will get full point if you

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

You chart will look like:

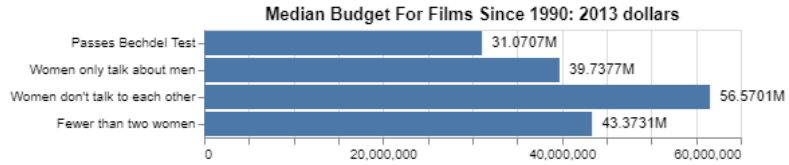


checkpoint 6: BONUS: format numbers.

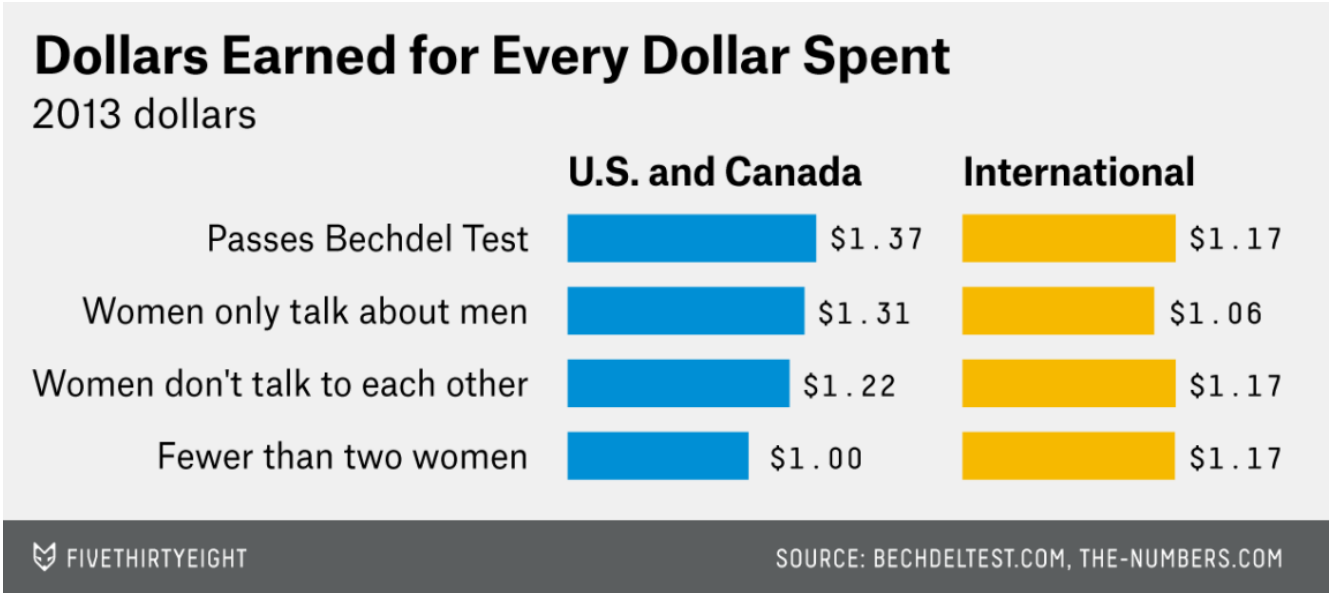
You will get full points if you

- Complete checkpoint 5
- Adjust number labels to display millions. e.g. (31.4592 M instead instead of 31459218). You might want to read about format (https://altair-viz.github.io/user_guide/encoding.html?highlight=format%20type), and D3's format specification (https://github.com/d3/d3-format#locale_format), or search around.

You chart will look like:



Visualization 2 Replicate this visualization



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

Left chart:

- Data Name: *movieDF*
- mark type: bar
- Encoding Specification:
  - x:clean_test:N
  - y: roi_dom:Q

Right chart:

- Data Name: *movieDF*
- mark type: bar
- Encoding Specification:
  - x: clean_test:N
  - y: roi_dom:Q

Compound Method (how to join these charts together?): (left_chart|right_chart)

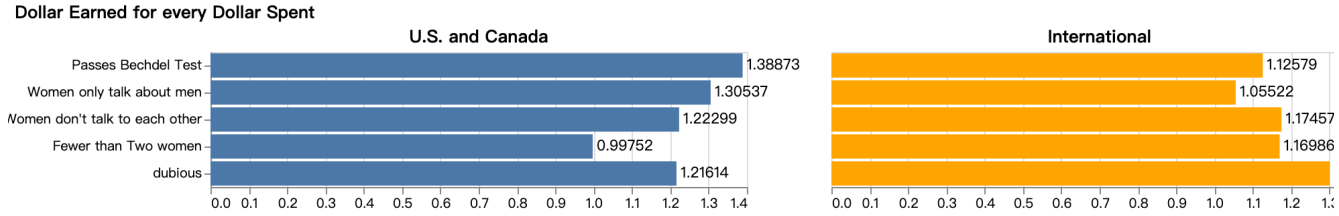
Step 2: Create your chart.

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 below.

In [6]:
```
#TODO: Replicate chart 2
names = {'men': 'Women only talk about men', 'dubious': 'dubious', 'notalk': "Women don't talk to each other", 'nowomer
movieDF_since_1990['clean_test'] = movieDF_since_1990['clean_test'].replace(names)

# movieDF['int_only'] = movieDF['intgross_2013$'] - movieDF['domgross_2013$']
# movieDF['roi_dom'] = (movieDF['domgross_2013$']) / movieDF['budget_2013$']
# movieDF['roi_int'] = (movieDF['int_only']) / movieDF['budget_2013$']

bar_dom = alt.Chart(movieDF_since_1990).mark_bar(filled=True).encode(
    alt.Y('clean_test:N', title=None, sort=['Passes Bechdel Test', 'Women only talk about men',  "Women don't talk to e
    alt.X('median(roi_dom):Q', title=None)
).properties(title='U.S. and Canada')
bar_int = alt.Chart(movieDF_since_1990).mark_bar(filled=True, color='Orange').encode(
    alt.Y('clean_test:N', title=None, axis=None,  sort=['Passes Bechdel Test', 'Women only talk about men',  "Women dor
    alt.X('median(roi_int):Q', title=None)
).properties(title='International')

text_dom = bar_dom.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')
)

text_int = bar_int.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")
)
# (bar_dom | bar_int).resolve_scale(y='shared').properties(title='Dollar Earned for every Dollar Spent')
(bar_dom + text_dom | bar_int + text_int).resolve_scale(y='shared').properties(title='Dollar Earned for every Dollar Sp
```

```
/var/folders/wd/7mmm5v2n75b_cxpswz3tf4100000gn/T/ipykernel_28459/3302642807.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)
  movieDF_since_1990['clean_test'] = movieDF_since_1990['clean_test'].replace(names)
```

Out[6]:

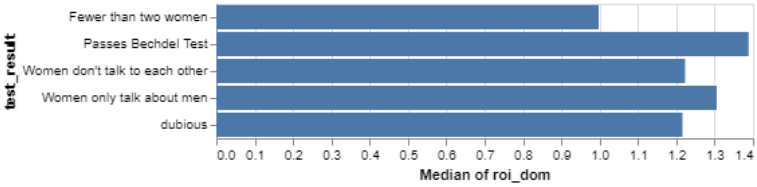


Visualization 2 Checkpoints

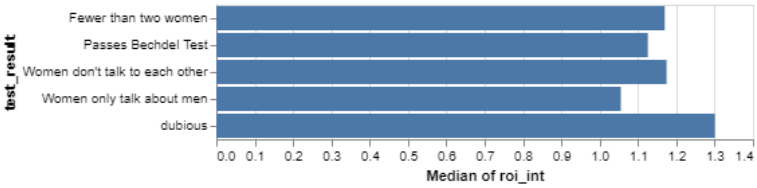
checkpoint 1: basic bar charts

- 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:



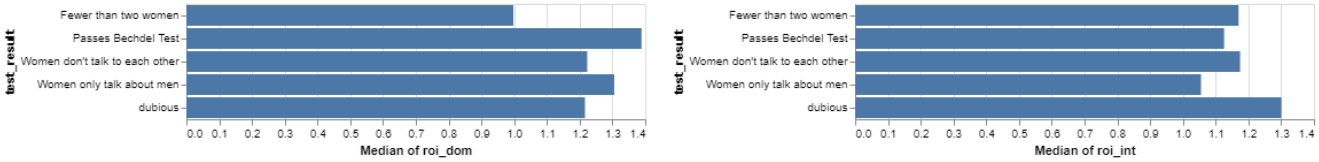
and



checkpoint 2: joining two charts

- completed checkpoint1
- joined two charts

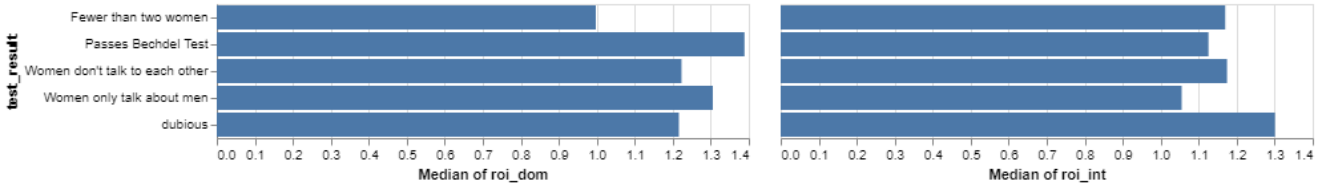
You chart will look like:



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:

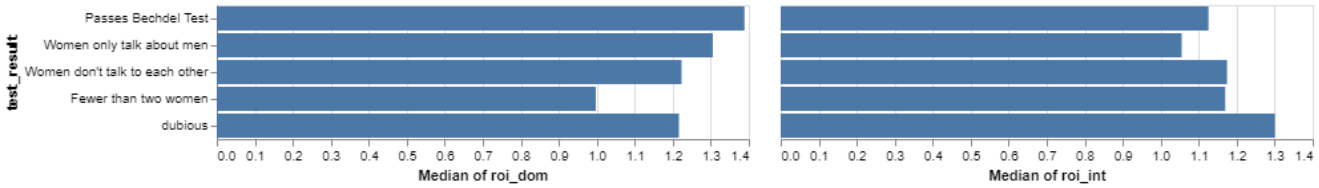


checkpoint 4: sort y-axis

- 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"
```

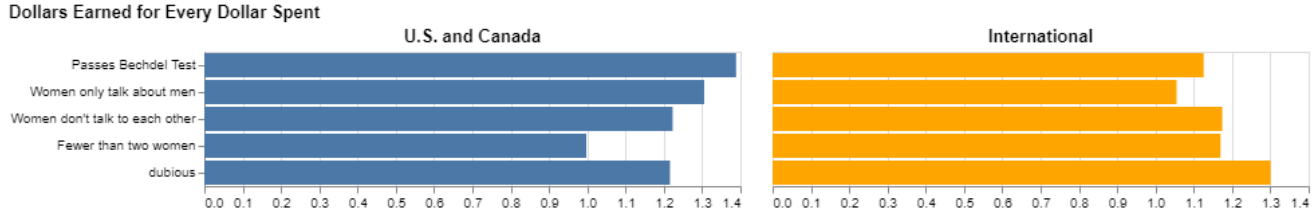
You chart will look like:



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)
- remove y axis label "test_result"

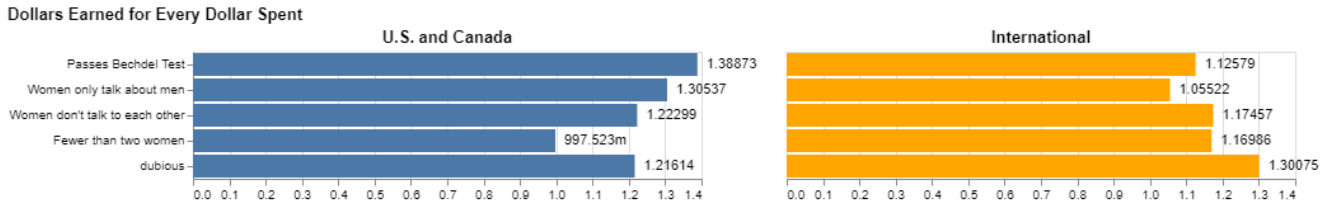
You chart will look like:



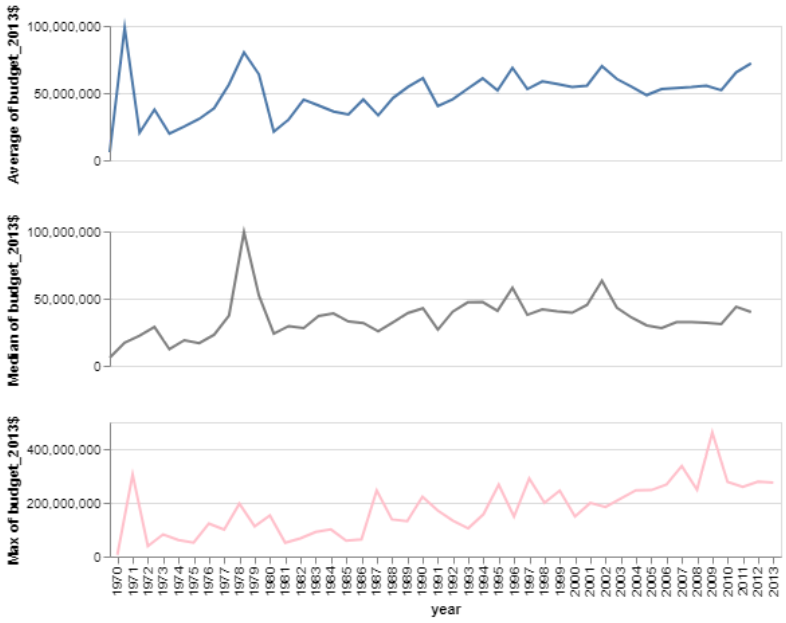
checkpoint 6: BONUS: Add number layer

- completed checkpoint 5
- add number annotations

You chart will look like:



Visualization 3: Replicate this visualization



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

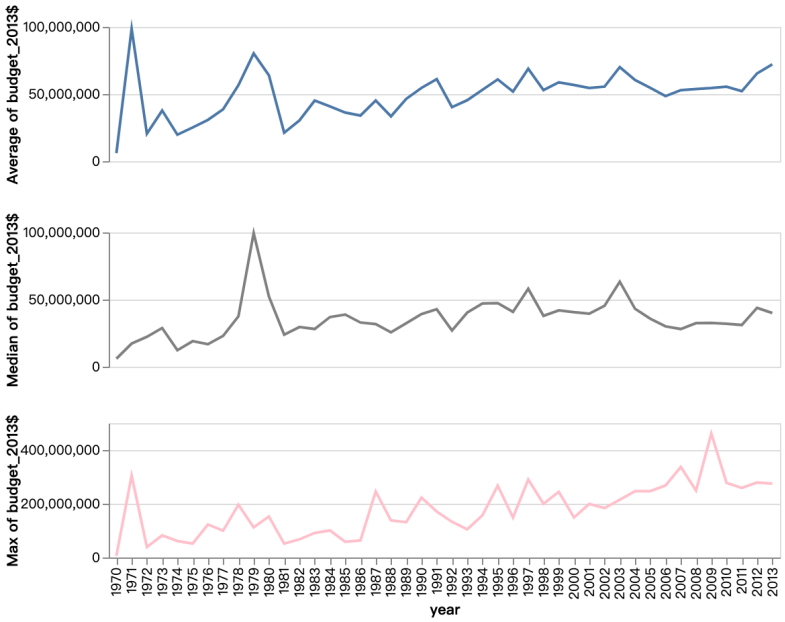
- Data Name: *movieDF*
- mark type: line
- Encoding Specification (1st chart):
  - x: year:O
  - y: average(budget_2013): $Q * EncodingSpecification(2ndchart) : *x : year : O * y : median(budget_2013)$:Q
- Encoding Specification (3rd chart):
  - x: year:O
  - y:max(budget_2013$):Q

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.

In [7]:
```
#TODO: Replicate visualization 3
line_average = alt.Chart(movieDF).mark_line().encode(
    alt.Y('average(budget_2013$):Q'),
    alt.X('year:O', axis=None)
).properties(
    width=500,
    height=100
)

line_max = alt.Chart(movieDF).mark_line(color='pink', width=500,height=100).encode(
    alt.Y('max(budget_2013$):Q'),
    alt.X('year:O', title='year')
).properties(
    width=500,
    height=100
)

line_median = alt.Chart(movieDF).mark_line(color='grey', width=500,height=100).encode(
    alt.Y('median(budget_2013$):Q'),
    alt.X('year:O', axis=None)
).properties(
    width=500,
    height=100
)

(line_average & line_median & line_max).resolve_scale(x='shared')
```

Out[7]:



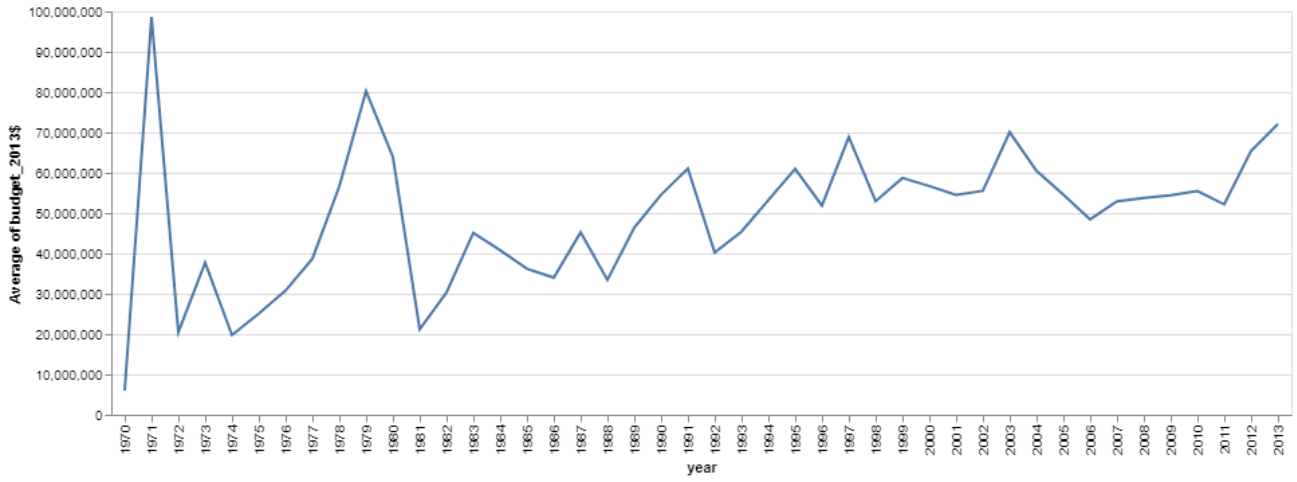
Visualization 3 Checkpoints

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

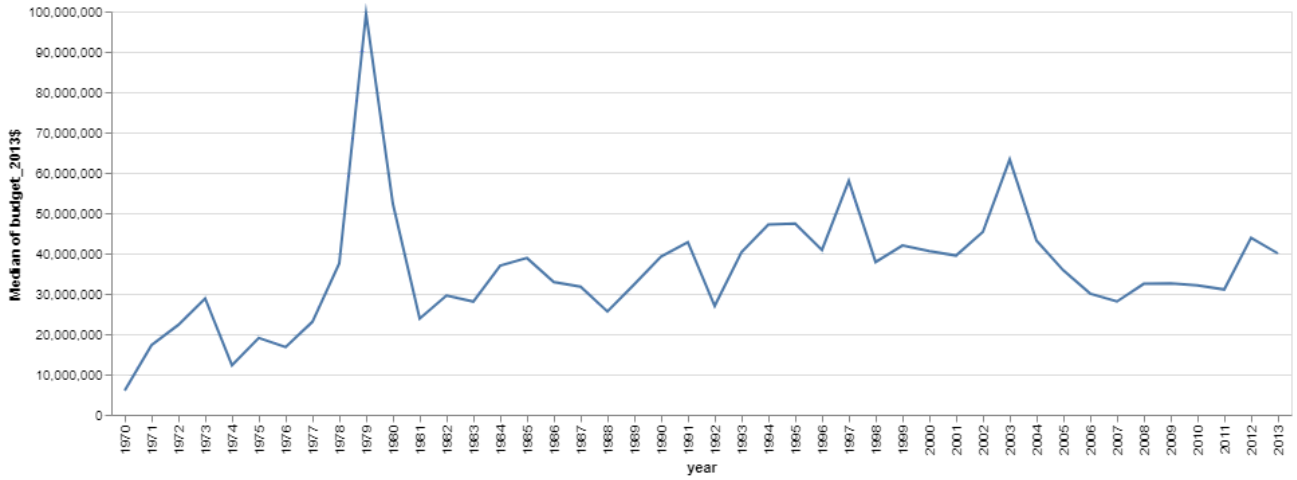
You will get full points if you

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

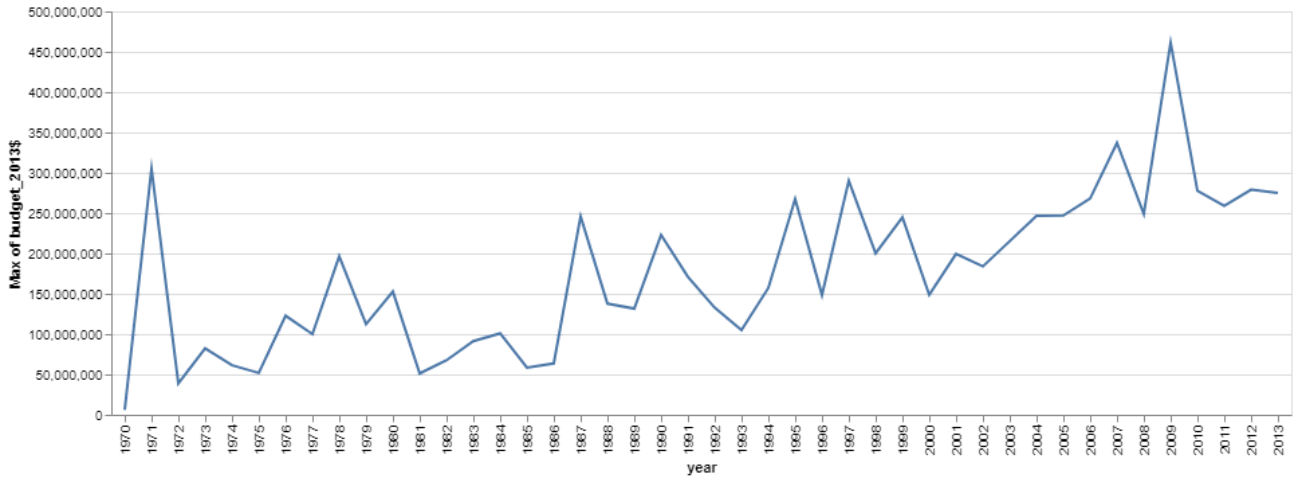
You chart will look like:



and



and

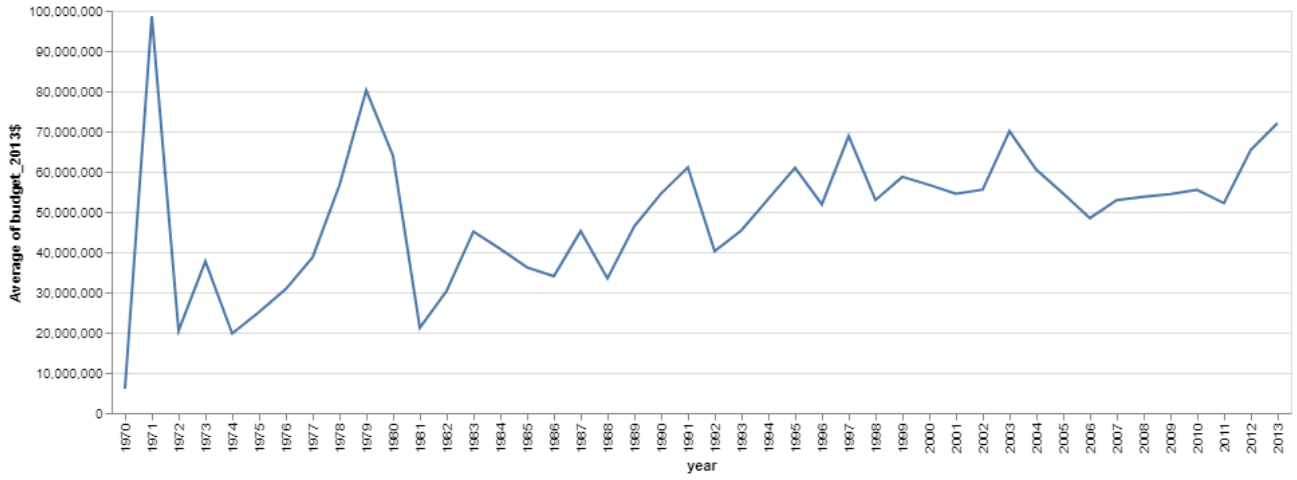


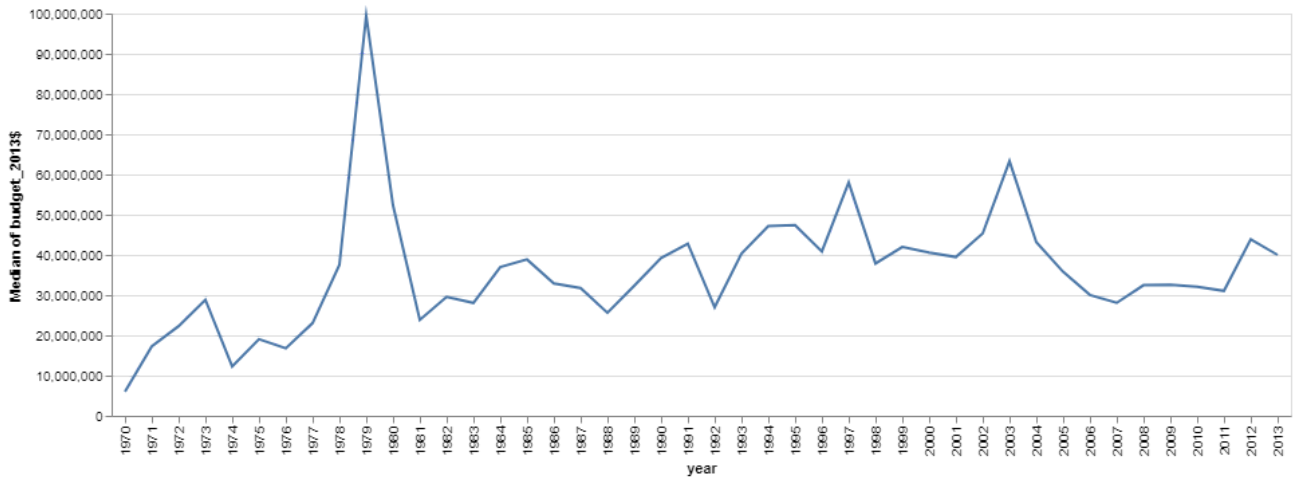
checkpoint 2: concat 3 line charts

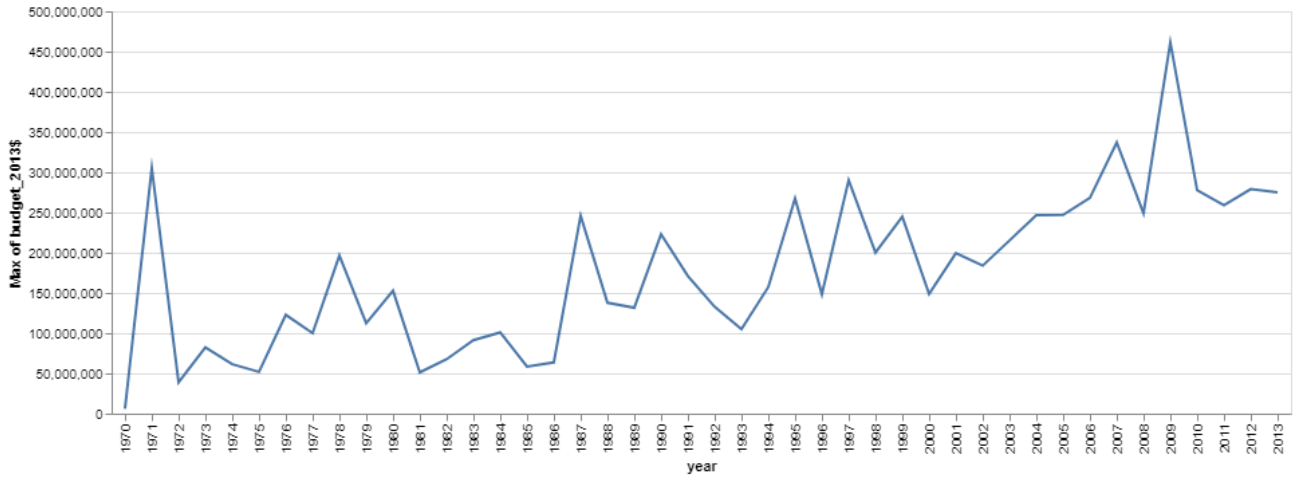
You will get full points if you

- Complete checkpoint 1
- Concat 3 charts vertically

You chart will look like:







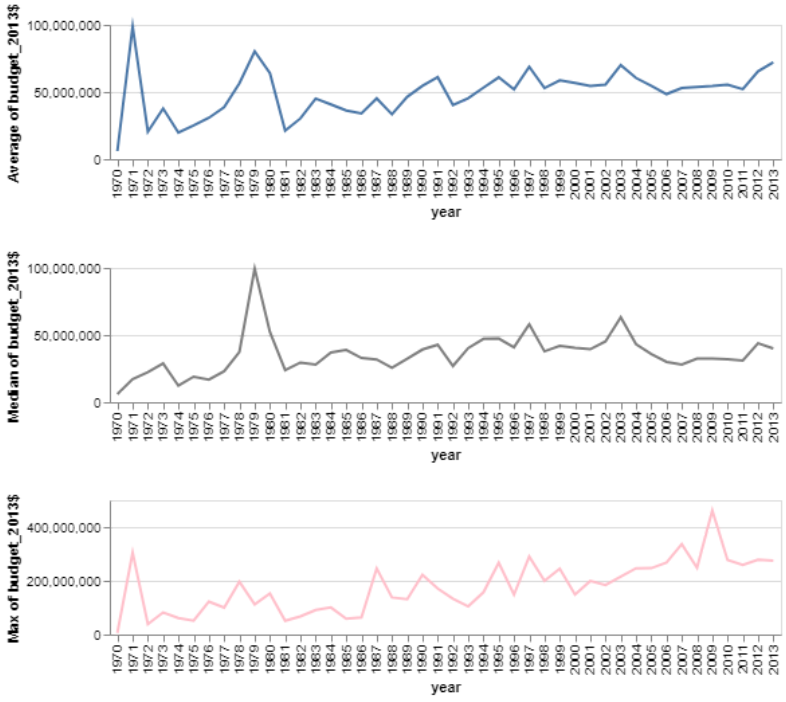
checkpoint 3: adjust width, height and color

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:

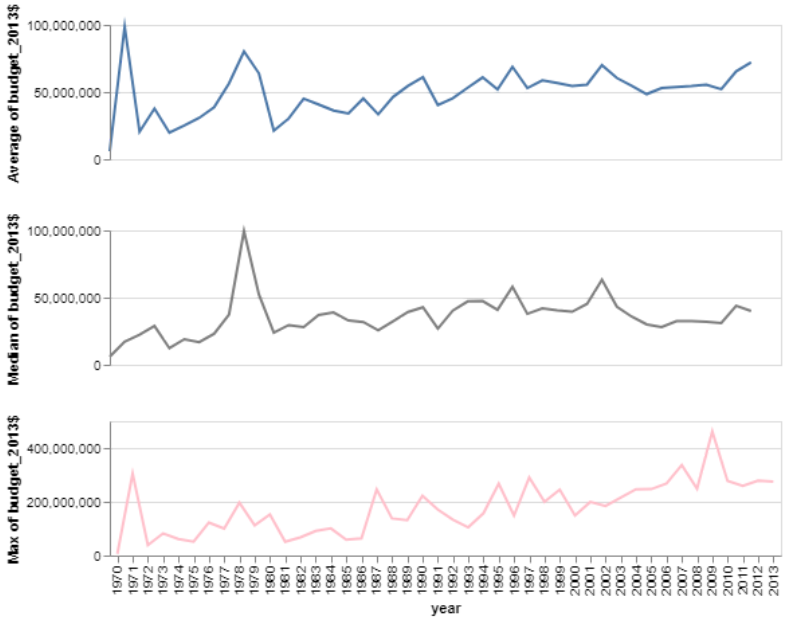


checkpoint 4: resolve axis and remove duplicated x-axis

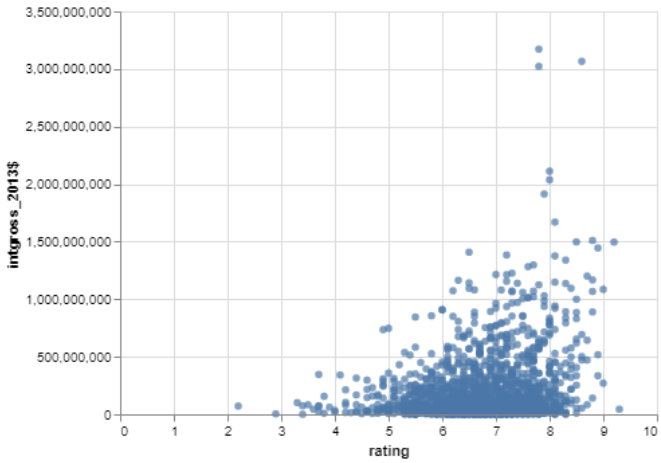
You will get full points if you

- 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)

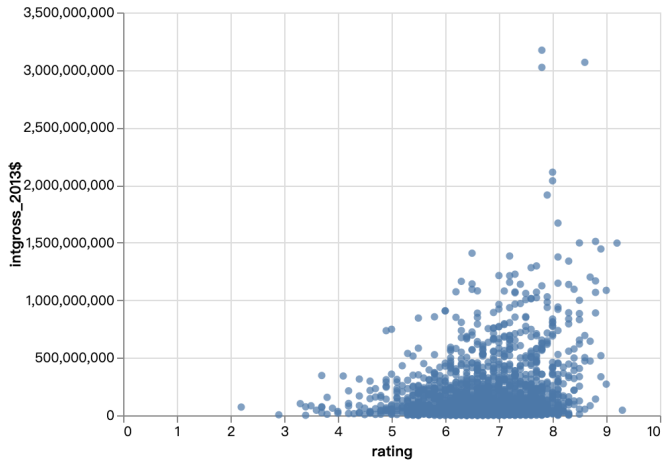
- Data Name: *movieDF*
- mark type: mark_circle
- Encoding Specification:
  - x: intgross_2013$:Q
  - y: rating:Q

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.

In [8]:
```
#TODO: Replicate visualization 4

point = alt.Chart(movieDF).mark_circle().encode(
    alt.Y('intgross_2013$:Q'),
    alt.X('rating:Q')
)
point
```

Out[8]:



End of Lab

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

1. save to PDF
   - We suggest using your browser's print feature: File->Print->Save PDF. You can try the notebook File->Download As->PDF, but we've noticed this doesn't work as well. If you're a Windows user and need help, take a look here (https://www.digitaltrends.com/computing/print-pdf-windows/)
   - We recommend doing this in Landscape mode to fit the notebook better.
2. save to ipynb (File -> Download as -> Notebook (.ipynb))

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