The U.S. Election throughout History - Process Book

Duanduan Liu
Kanrong Yu
Weina Chen

Overview and Motivation

As the eye-catching 2016 US presidential eventually settled, there had been so many politically motivated visualizations like those provided by the famous FiveThirtyEight and NYTimes, and the public expectation towards the final result was somehow affected or even misled by them. Therefore, what prompts us to do this project is to create an unbiased data-driven visualization, which won't mislead people's interpretation from viz. Moreover, from this year's presidential result, we should admit election results are generally hard to predict despite many different polls made on large group of samples. However, we consider it still interesting and meaningful to look at the whole idea of the evolution of US presidential election results and investigate if there is correlation between the results and some other factors.

To be more specific, we aim to obtain a general picture of the US election through demographic of election in the following aspects:
- Discover presidential shift from 1920 to 2012.
- Detect how the states stacked up in every election and how they have swung during the time frame. (In progress)
- Expose the relative relations with the preference of newspaper endorsements e.g. NYT, the external environment like recession or recovered economy, and the trend of growth of the executive branch e.g. Federal Budget Receipts & Outlays, and GDP.

Related Work

- There Are Many Ways to Map Election Results. We've Tried Most of Them. - by NYTimes
  http://www.nytimes.com/interactive/2016/11/01/upshot/many-ways-to-map-election-results.html?_r=0
- LIVE RESULTS AND MAPS Election Results 2016
  http://graphics.wsj.com/elections/2016/results/
- The various Live US election results we watched in class

Data

1. Data Source

The main reference is from the Library of Congress, a post named "U.S. Election Statistics: A

Resource Guide" (https://www.loc.gov/rr/program/bib/elections/statistics.html). This resource guide compiles a list of online and print resources that contain U.S. election statistics for both federal and state elections. To be more specific, we mainly use three online resources listed in this guide:

- Dave Leip's Atlas of U.S. Presidential Elections, which is our main data source that includes the voting statistics for every election by state level.
http://uselectionatlas.org/

- American Presidency Project: Presidential Election Data, from which we grabbed the financial data (Federal Budget Receipts & Outlays, and GDP)
http://www.presidency.ucsb.edu/elections.php

- Newspaper presidential endorsements, where we excerpted the endorsement of top 10 US newspapers (by circulation)
http://noahveltman.com/endorsements/
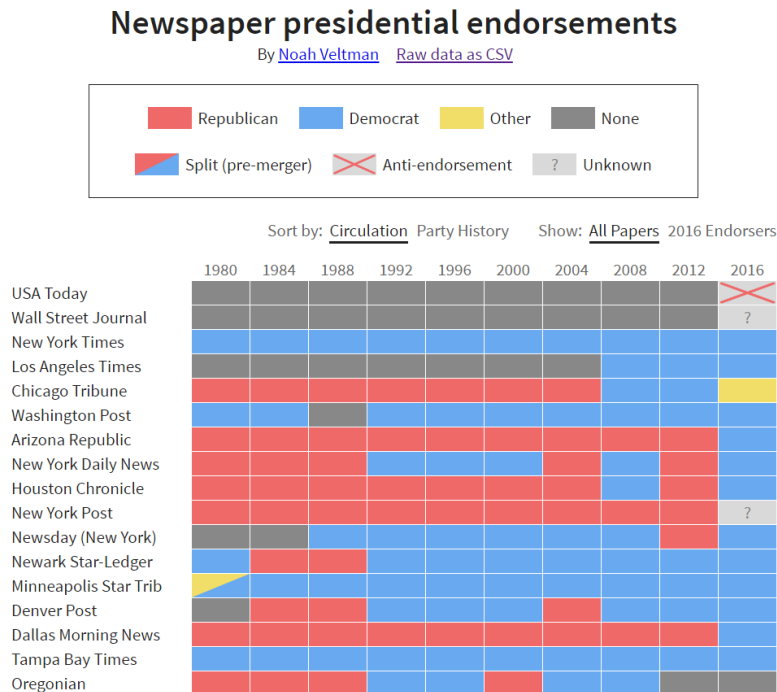
2. Data Processing

For Data Cleaning,

1) To keep the column consistency in "each year.csv", we uses Python to manualLy add EV_I, I_Percen, I_Actual with "0" value when there's only two parties (Democratic and Republican) at that year. By doing this, we got same number of columns in each file and could easily import relevant columns from the csv files using JavaScript.

2) Because of the admission of Alaska and Hawaii as states in 1959, the presidential election data before 1960 didn't have their information and similar situation goes with Washington DC, we added new rows with "0" values for these states in order to keep the row consistency for every year data file.

3) Also, we deleted % for the D_Percen, R_Percen and I_Percen columns and delete the thousand separators.

To avoid too much calculation in later JavaScript code, we did the processing work in Python in advance. Including:

1) Add a RD_Diff column as R_Percen - D_Percen
2) Put the state abbreviation name in State_Abbr
3) Based on RD_Diff, attach relevant color HEX towards it (in ten levels, from deep red to deep blue)

4) Besides the HEX, color rank is assigned (for sake of the sorting of state blocks in the stacked bar chart)
5) Recalculate Other_Percen as 1 - D_Percen - R_Percen - I_Percen as in the raw data, these 4 columns add up to more than 1

```python
# Sort by index
df.sort_index(inplace=True)

# Delete the final Row
df = df[df['State'] != "Total"]

# Add RD_Diff
df = add_RD_Diff(df)

# Add State_Abbr
df = add_State_Abbr(df)

# Add Color
df = add_color(df)

# Re caculate Other_Percen
df = tranform_Other_Percen(df, year)

# Add Color Rank
df = add_color_rank(df)

# Save df to CSV
folder_name = "Data/Cleaning/CSV/" + str(year) + ".csv"
df.to_csv(folder_name, index=False)
```

6) To make the state electoral shift chart, df_now["Shift"] = (-df_now["D_Percen"] + df_last["D_Percen"]) + (df_now["R_Percen"] - df_last["R_Percen"]) is added.

Here's the final look of each year.csv.

State	EV_D	EV_R	Total_Vote	D_Percen	R_Percen	Other_Percen	D_Actual	R_Actual	Other_Actual	I_Actual	EV_I	I_Percen	RD_Diff	State_Abbr	Color	colorRank	Last_Color	Shift
Alabama	0	9	2074338	38.36	60.55	1.09	795696	1255925	22717	0	0	0	22.19	AL	#fcae91	2	#fcae91	0.61
Alaska	0	3	300495	40.81	54.8	4.39	122640	164676	13179	0	0	0	13.99	AK	#fee5d9	1	#fcae91	-7.54
Arizona	0	11	2306559	44.45	53.48	2.07	1025232	1233654	47673	0	0	0	9.03	AZ	#fee5d9	1	#fee5d9	0.55
Arkansas	0	6	1069468	36.88	60.57	2.55	394409	647744	27315	0	0	0	23.69	AR	#fcae91	2	#fee5d9	3.83
California	55	0	13055815	60.16	37.07	2.77	7854285	4839958	361572	0	0	0	-23.09	CA	#6baed6	-3	#6baed6	0.93
Colorado	9	0	2571846	51.45	46.09	2.46	1323102	1185243	63501	0	0	0	-5.36	CO	#eff3ff	-1	#eff3ff	3.59
Connecticut	7	0	1558993	58.06	40.72	1.22	905109	634899	18985	0	0	0	-17.34	CT	#bdd7e7	-2	#6baed6	5.03
Delaware	3	0	413921	58.61	39.98	1.41	242584	165484	5853	0	0	0	-18.63	DE	#bdd7e7	-2	#6baed6	6.35
D. C.	3	0	293764	90.91	7.28	1.81	267070	21381	5313	0	0	0	-83.63	DC	#08519c	-5	#08519c	2.3

Apart from 29 election year state level data files, we also generated a total file, with state level electoral statistics, media endorsement, financial data and background information.

Year	Color	D_Percen	R_Percen	EV_Total	USA Today	Wall Street J	New York Tir	Los Angeles	Chicago Tribu	Washington	Arizona Repu	New York Da	Houston Chr	New York Po	D_EV	R_EV	I_EV	Keyword
1976	#08519c	50.08	48.01	537			#08519c								297	240	0	Watergate scand:
1980	#a50f15	41.01	50.75	538	#888888	#888888	#08519c	#888888	#a50f15	#08519c	#a50f15	#a50f15	#a50f15	#a50f15	49	489	0	Iran hostage crisi
1984	#a50f15	40.56	58.77	538	#888888	#888888	#08519c	#888888	#a50f15	#08519c	#a50f15	#a50f15	#a50f15	#a50f15	13	525	0	Strong economic
1988	#a50f15	45.65	53.37	537	#888888	#888888	#08519c	#888888	#a50f15	#888888	#a50f15	#a50f15	#a50f15	#a50f15	111	426	0	Good economy, S
1992	#08519c	43.01	37.45	538	#888888	#888888	#08519c	#888888	#a50f15	#08519c	#a50f15	#08519c	#a50f15	#a50f15	370	168	0	Economy in reces
1996	#08519c	49.23	40.72	538	#888888	#888888	#08519c	#888888	#a50f15	#08519c	#a50f15	#08519c	#a50f15	#a50f15	379	159	0	Recovered econo
2000	#a50f15	48.38	47.87	537	#888888	#888888	#08519c	#888888	#a50f15	#08519c	#a50f15	#08519c	#a50f15	#a50f15	266	271	0	Controversy over
2004	#a50f15	48.26	50.73	537	#888888	#888888	#08519c	#888888	#a50f15	#08519c	#a50f15	#a50f15	#a50f15	#a50f15	251	286	0	911 in 2001, War
2008	#08519c	52.86	45.6	538	#888888	#888888	#08519c	#08519c	#08519c	#08519c	#a50f15	#08519c	#08519c	#a50f15	365	173	0	2008 economic c
2012	#08519c	51.01	47.15	538	#888888	#888888	#08519c	#08519c	#08519c	#08519c	#a50f15	#a50f15	#a50f15	#a50f15	332	206	0	Economic recove

Exploratory Data Analysis

After we prepared and processed the raw data, we leveraged Tableau dashboard to come up with a general idea of how our data looks like and where each part should be arranged.

Here are some insights we obtained from Tableau dashboard, which helped us in our future viz design with D3.js.

- Titles are needed for some charts, to help the audience better understand what is displayed here.
- When interacting, tooltips can be a good choice; for mouse-click that cause changes in coordinated views, there should be a clear emphasis (possible ways can be change of color, size, etc.) on the according data and automatically reset afterwards (Tableau does poor in that).
- Scrolling down to check the viz in the bottom can be annoying, therefore we would try to set the layout in one screen, adaptive to any resolution.

Design Evolution

1. Early Stage

We sketched two different designs for historical data, as shown below. The first design focuses on the map display, and attempts to include various types of information in the vis; the second design focuses on the trend of every state, which may show some pattern of swing states.
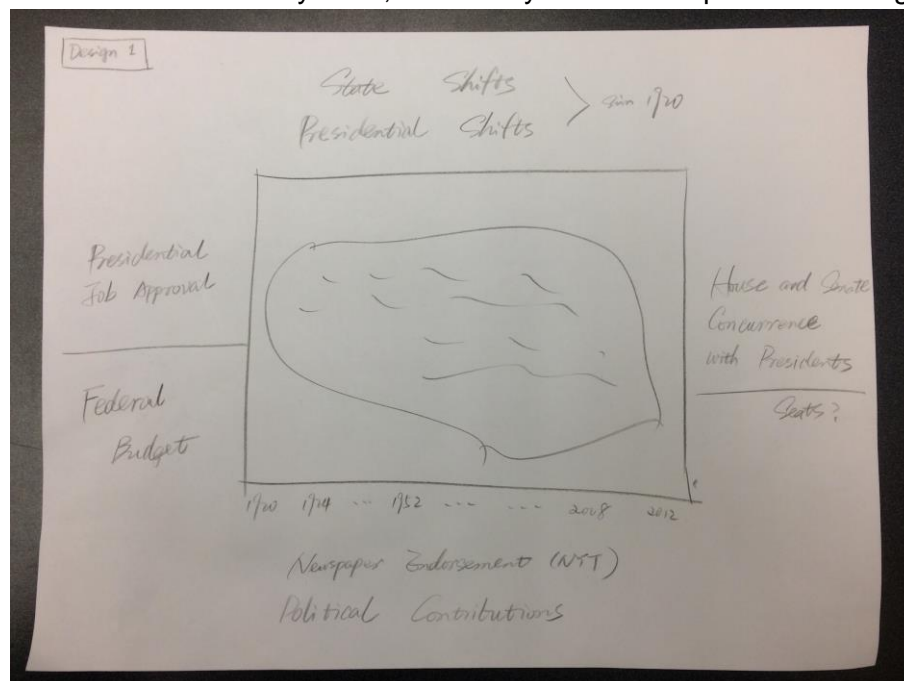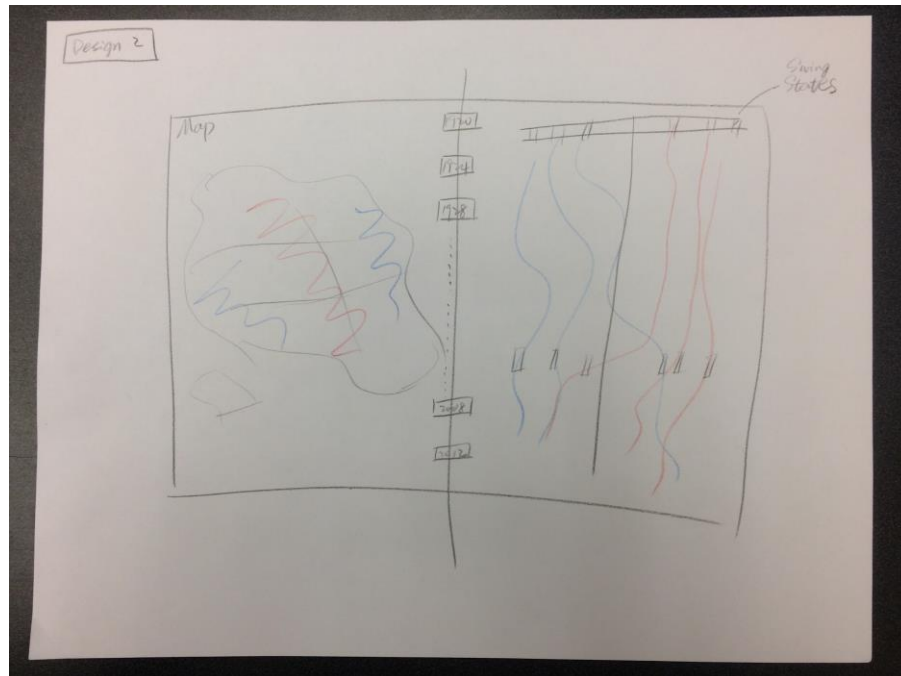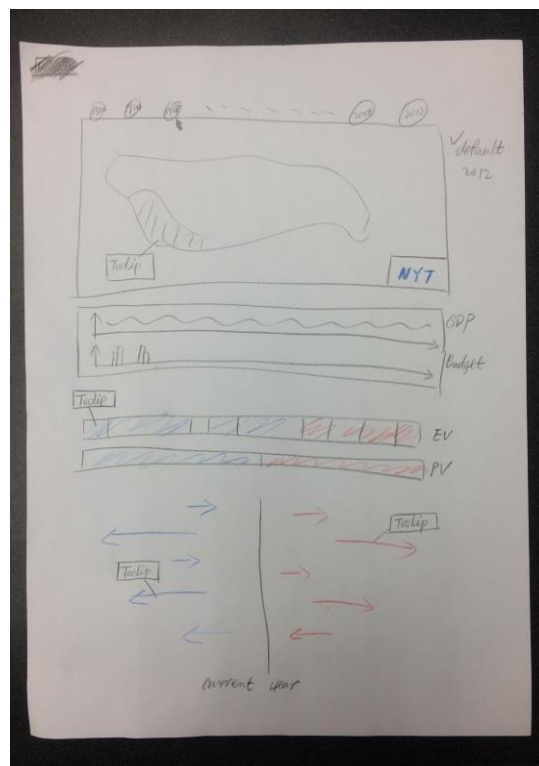
After discussion, we decided to use the first design, which is more consistent with our motivation to show as much information as possible in the vis.
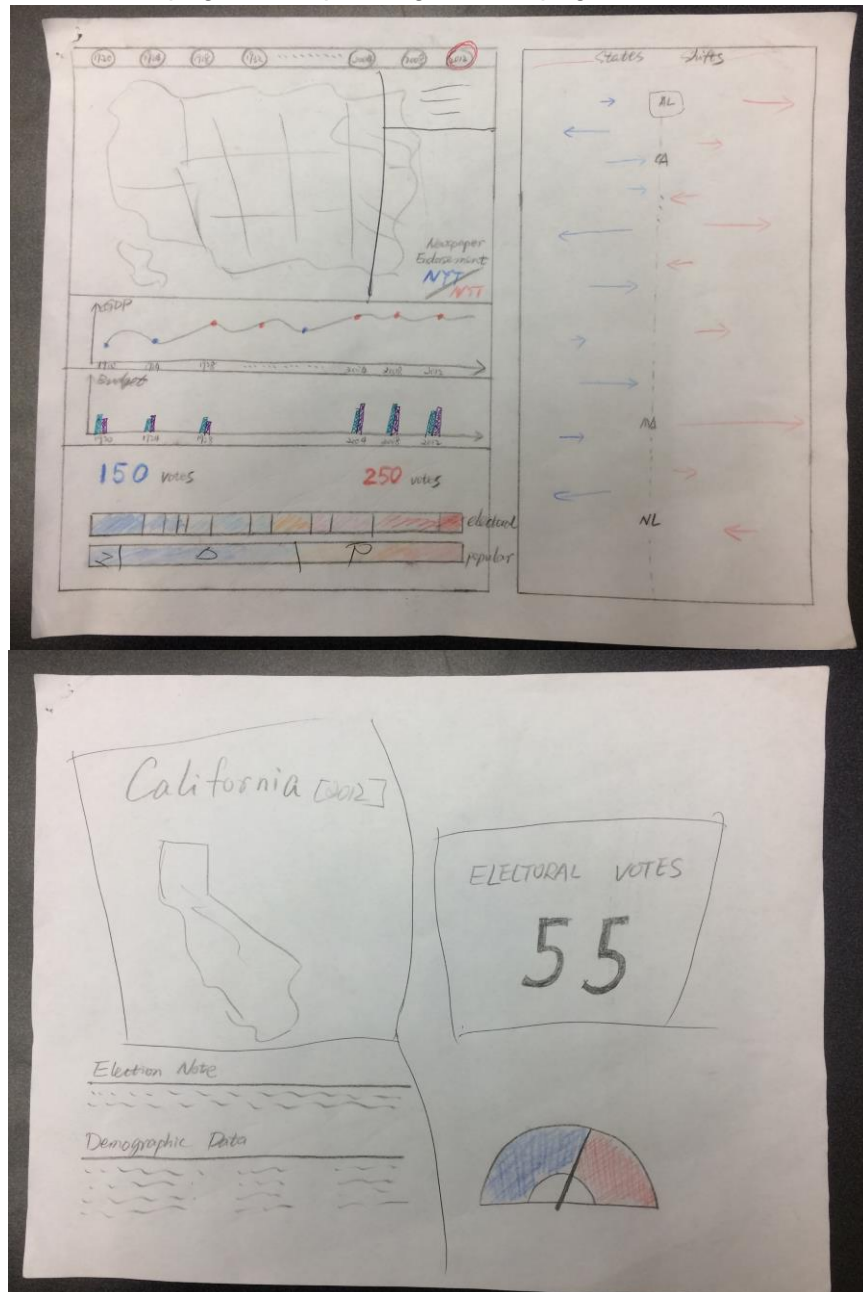
2. Vis Draft

Once we decided to use design 1, the next step is trying to generate a draft of our vis in more detail.

First, we created the whole viz in vertical-fashion:

But considering it will expand the whole page by using the scroll, it may not straightforward for audience to capture an entire idea at each year. Thus, we rearranged the page to a horizontal-fashion, and add a second page corresponding the first page.

Furthermore, we reconsidered the second page which is explicitly a zoom-in version of the first page, and concluded that the second page was totally redundant information. Therefore, the final vis draft is just the first page in horizontal type.

3. Operation Design

For timeline: "click" should trigger all the other vis elements.
For map: "hanover" should trigger tooltip of every state; "click" should trigger relevant highlights in other vis elements.

For stack bar: "hanover" should trigger tooltip of every state; "click" should trigger relevant highlights in map.
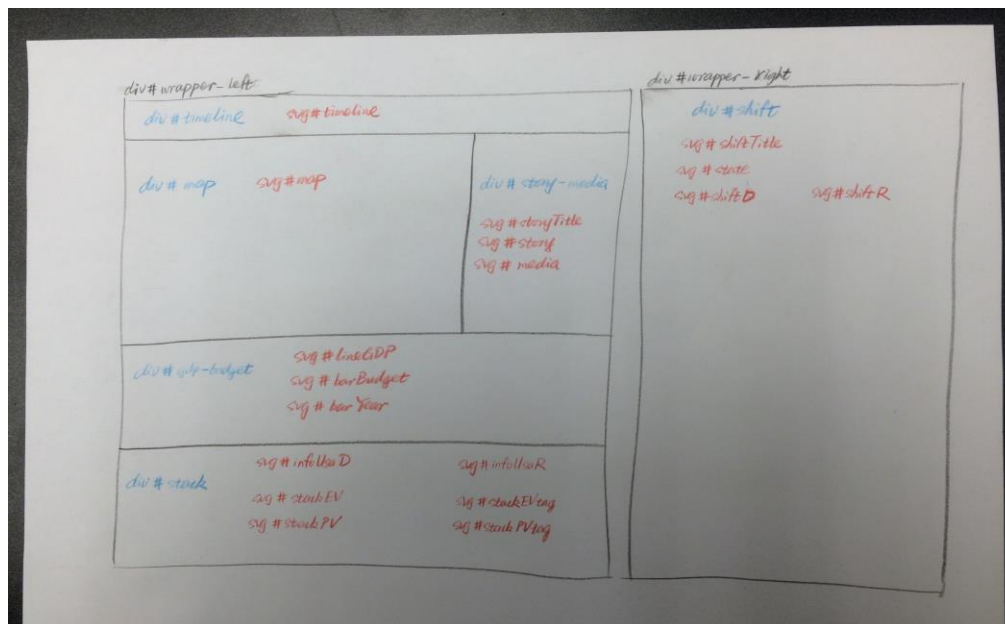For shift slope chart: "hanover" should trigger tooltip of every state.

4. Current Vis
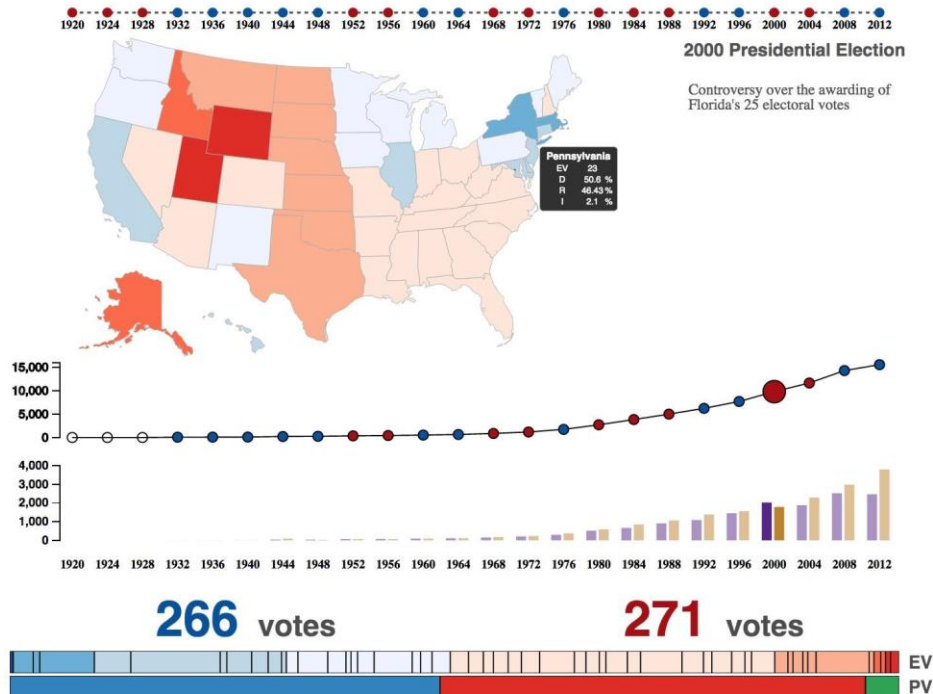
Implementation

1. Vis Elements Structure

For div#wrapper-left, it includes:
- div#timeline
  - svg#timeline
- div#map
  - svg#map
- div#story-media
  - svg#storyTitle
  - svg#story
  - svg#media
- div#gdp-budget
  - svg#lineGDP
  - svg#barBudget
  - svg#barYear
- div#stack
  - svg#infoD
  - svg#infoR
  - svg#stackEV
  - svg#stackPV
  - svg#stackEVtag
  - svg#stackPVtag

For div#wrapper-right, it includes:
- div#shift
  - svg#shiftTitle
  - svg#state
  - svg#shiftD
  - svg#shiftR

2. Code Architecture

Generally speaking, Utils includes the interaction code and Chart includes all the static charts; with run function, they could be connected and executed together.

```
Utils = {⸱⸱⸱
}

Chart = {⸱⸱⸱
}

function run(){⸱⸱⸱
}
```

1) Utils

When a specific year on the timeline is clicked, the information in other charts like map, vote text, and stacked bars would be updated.

```
Utils = {

    updateMap: function(year){▯
    },

    updateStoryTitle: function(year){▯
    },

    updateStory: function(year){▯
    },

    updateStackEV: function(year){▯
    },

    updateStackPV: function(year){▯
    },

    updateUSADR: function(year){▯
    },

    updateGDPBudget: function(year){▯
    },

    updateStackEVForMap: function(id){▯
    },

    /*
    updateShift: function(data, category){}*/

    update(d){▯
    },

    updateForMap(d){▯
    }
}
```

2) Chart

So far, 11 Vis elements are included in Chart, other one like the state shift map is expected to be done in the next stage.

```
Chart = {

    timeline: function(id){▯
    },

    map: function(id, year){▯
    },

    storyTitle: function(id, year) {▯
    },

    story: function(id, year) {▯
    },

    lineGDP: function(id){▯
    },

    barBudget: function(id){▯
    },

    barYear: function(id) {▯
    },

    infoUSAD: function(id, year){▯
    },

    infoUSAR: function(id, year){▯
    },

    stackEV: function(id, year){▯
    },
```

Evaluation

What did you learn about the data by using your visualizations? How did you answer your questions? How well does your visualization work, and how could you further improve it?
TBD