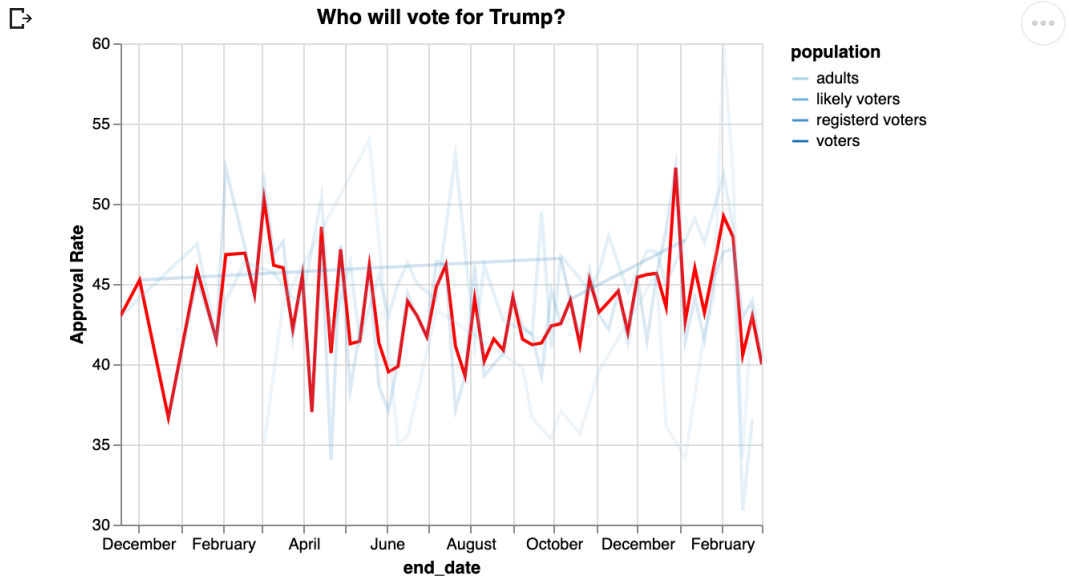
Information Visualization Altair Lab 5

1. Interactive Visualizations

```
import altair as alt
import pandas as pd
from vega_datasets import data as vega_data# load the data
import panel as pn
from altair import datum
import datetime as dt
datasetURL='https://projects.fivethirtyeight.com/polls-page/president_polls.csv'
df=pd.read_csv(datasetURL, encoding="latin-1")
df['end_date']= pd.to_datetime(df.end_date)



df1 = df.copy
df1 = df.set_index('end_date')
df_weekly = pd.DataFrame()
df_weekly['pct'] = df1.groupby('answer').pct.resample('W').mean()
df_weekly = df_weekly.dropna().reset_index()


adult = df[df['population']=='a']
registerd_voters = df[df['population']=='rv']
voters = df[df['population']=='v']
likely_voters = df[df['population']=='lv']

adult = adult.set_index('end_date')# a new dataframe to hold our weekly average
adult_weekly = pd.DataFrame()# group by the candidate name, and then calculate weekly averages
adult_weekly['pct'] = adult.groupby('answer').pct.resample('W').mean()# clean up weeks with no data and gets
adult_weekly = adult_weekly.dropna().reset_index()
adult_weekly['population'] = 'adults'

registerd_voters = registerd_voters.set_index('end_date')# a new dataframe to hold our weekly average
registerd_voters_weekly = pd.DataFrame()# group by the candidate name, and then calculate weekly averages
registerd_voters_weekly['pct'] = registerd_voters.groupby('answer').pct.resample('W').mean()# clean up weeks
registerd_voters_weekly = registerd_voters_weekly.dropna().reset_index()
registerd_voters_weekly['population'] = 'registerd voters'

voters = voters.set_index('end_date')# a new dataframe to hold our weekly average
voters_weekly = pd.DataFrame()# group by the candidate name, and then calculate weekly averages
voters_weekly['pct'] = voters.groupby('answer').pct.resample('W').mean()# clean up weeks with no data and ge
voters_weekly = voters_weekly.dropna().reset_index()
voters_weekly['population'] = 'voters'


likely_voters = likely_voters.set_index('end_date')# a new dataframe to hold our weekly average
likely_voters_weekly = pd.DataFrame()# group by the candidate name, and then calculate weekly averages
likely_voters_weekly['pct'] = likely_voters.groupby('answer').pct.resample('W').mean()# clean up weeks with
likely_voters_weekly = likely_voters_weekly.dropna().reset_index()
likely_voters_weekly['population'] = 'likely voters'

import pandas as pd
weekly = pd.concat([adult_weekly,registerd_voters_weekly,voters_weekly,likely_voters_weekly],ignore_index=Tr


# define radio button
population = list(["adults","registerd voters", "voters","likely voters"])
select_radio = alt.selection_single(
    fields=['population'],
    bind=alt.binding_radio(options=population, name="Please select one other candidate to compare: "),
```

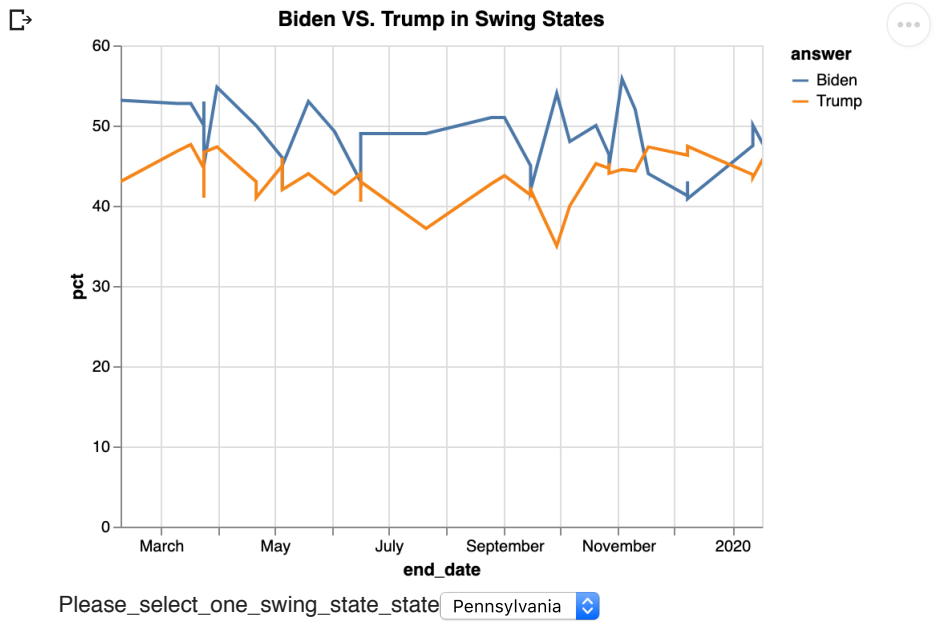
```
    name="Population",
on='keyup', clear="false",
    empty = "none"
)
selection_zoom = alt.selection_interval(bind="scales", encodings = ["x"])
colorCondition = alt.condition(select_radio,alt.value(1),alt.value(0.2))

# create the static chart for Trump
chart1 = alt.Chart(df_weekly).mark_line(color = 'red').transform_filter(
    (alt.datum.answer == "Trump")
).encode(
    x=alt.X('end_date:T'),            # time on the X axis
    y=alt.Y('pct:Q',title='Approval Rate'),                 # pct polling on the Y
)

chart2 = alt.Chart(weekly).mark_line().transform_filter(
    (alt.datum.answer == "Trump")
).encode(
    alt.X("end_date:T"),
    alt.Y("pct:Q", scale=alt.Scale(zero=False)),
    alt.Color("population:N",scale=alt.Scale(scheme='blues')),
    opacity = colorCondition,
    tooltip = ("pct", "end_date:T")
).add_selection(
    select_radio
).add_selection(
    selection_zoom
)

vis1 = chart1 + chart2
vis1.properties(title='Who will vote for Trump?')
```



Please select one other candidate to compare: adults registerd voters voters likely voters

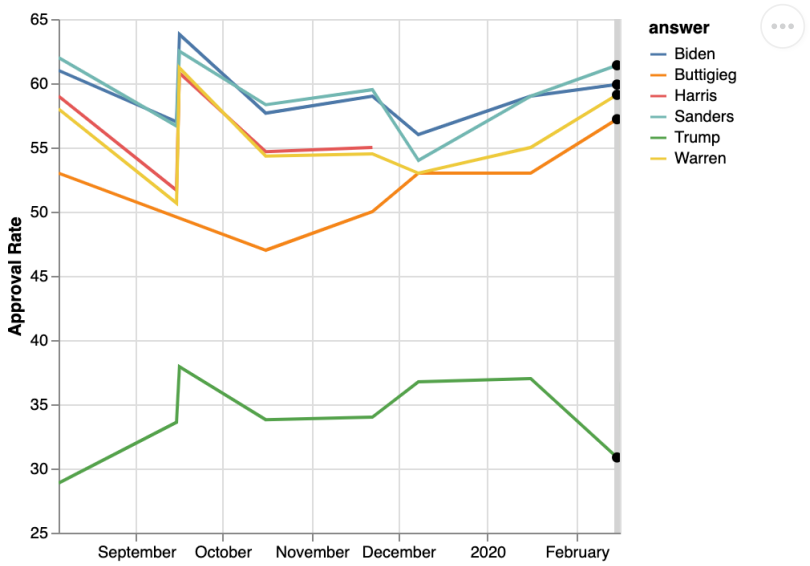
```
MI_polls = df[df['state']=='Michigan']
MI_polls = MI_polls.set_index('end_date')# a new dataframe to hold our weekly average
MI_weekly = pd.DataFrame()# group by the candidate name, and then calculate weekly averages
MI_weekly['pct'] = MI_polls.groupby('answer').pct.resample('W').mean()# clean up weeks with no data and gets
MI_weekly = MI_weekly.dropna().reset_index()
MI_weekly['state']='Michigan'

WI_polls = df[df['state']=='Wisconsin']
WI_polls = WI_polls.set_index('end_date')# a new dataframe to hold our weekly average
```

```
WI_weekly = pd.DataFrame()# group by the candidate name, and then calculate weekly averages
WI_weekly['pct'] = WI_polls.groupby('answer').pct.resample('W').mean()# clean up weeks with no data and gets
WI_weekly = WI_weekly.dropna().reset_index()
WI_weekly['state']='Wisconsin'

PA_polls = df[df['state']=='Pennsylvania']
PA_polls = PA_polls.set_index('end_date')# a new dataframe to hold our weekly average
PA_weekly = pd.DataFrame()# group by the candidate name, and then calculate weekly averages
PA_weekly['pct'] = PA_polls.groupby('answer').pct.resample('W').mean()# clean up weeks with no data and gets
PA_weekly = PA_weekly.dropna().reset_index()
PA_weekly['state']='Pennsylvania'
swing_weekly = pd.concat([PA_weekly,WI_weekly,MI_weekly],ignore_index=True, sort =False)
selection_zoom = alt.selection_interval(bind="scales", encodings = ["x"])

swing_state=list(swing_weekly['state'].unique())

state_dropdown = alt.binding_select(options=swing_state)
state_select = alt.selection_single(fields=['state'], bind=state_dropdown, name="Please select one swing sta

Swing = alt.Chart(swing_weekly).mark_line().transform_filter(
    alt.FieldOneOfPredicate(field='answer', oneOf=['Trump','Biden'])
).encode(
    x=alt.X('end_date:T'),            # time on the X axis
    y=alt.Y('pct:Q'),                 # pct polling on the Y
    color=alt.Y('answer:N')
)

vis2 = Swing.add_selection(
    state_select
).transform_filter(
    state_select
).add_selection(
    selection_zoom
).properties(title='Biden VS. Trump in Swing States')

vis2
```

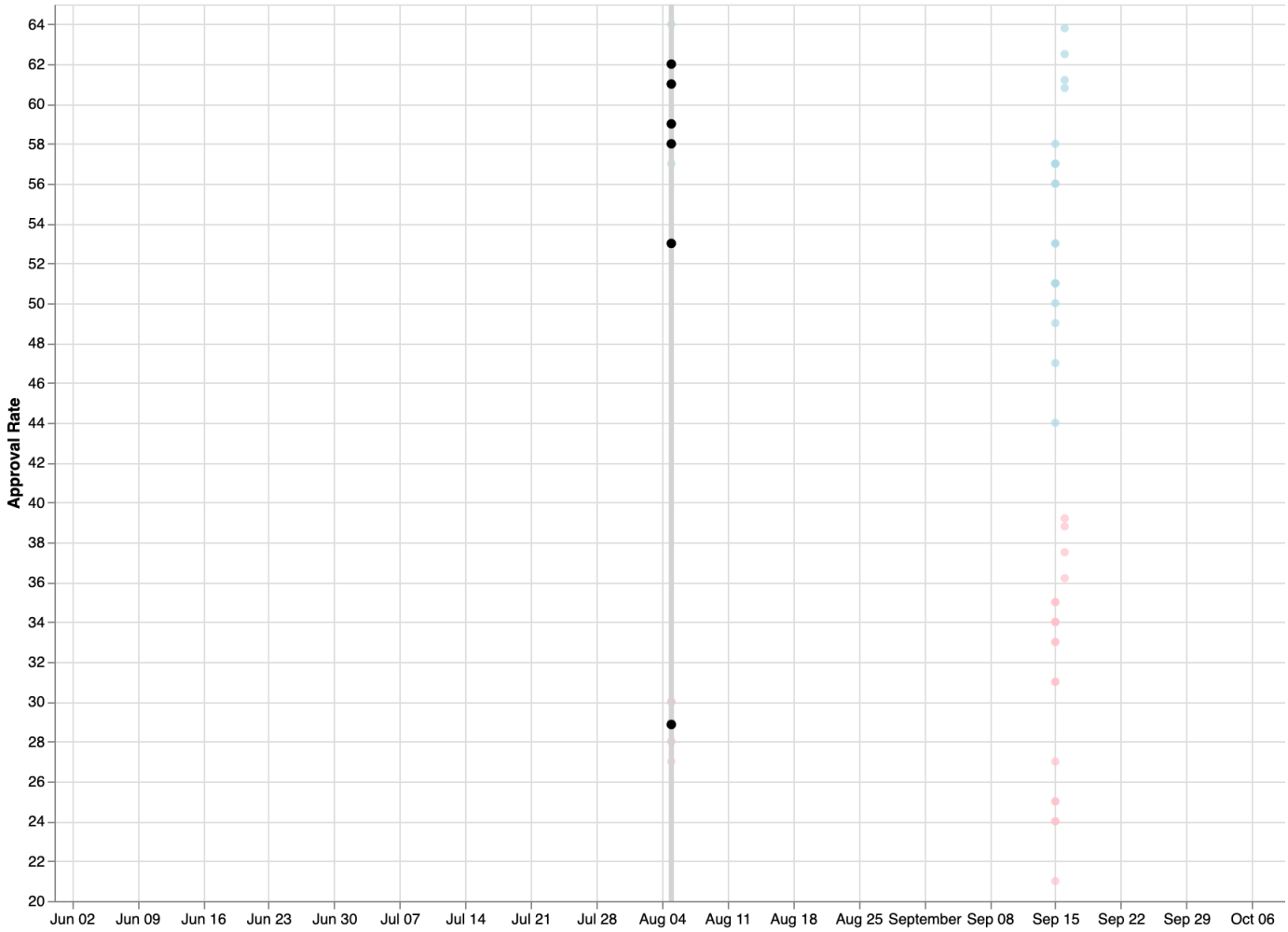



```
CA_polls = df[df['state']=='California']
CA_polls = CA_polls.set_index('end_date')# a new dataframe to hold our weekly average
CA_weekly = pd.DataFrame()# group by the candidate name, and then calculate weekly averages
CA_weekly['pct'] = CA_polls.groupby('answer').pct.resample('D').mean()# clean up weeks with no data and gets
CA_weekly = CA_weekly.dropna().reset_index()
```

```
selection=alt.selection_interval(
    bind='scales',#when you interact with this chart, you could change the scales
    encodings =['x'])#select the axis

nearest = alt.selection(type='single', nearest=True, on='mouseover',
                        fields=['end_date'], empty='none')

poll_CA = alt.Chart(CA_weekly).mark_line().transform_filter(
    alt.FieldOneOfPredicate(field='answer', oneOf=['Trump',"Warren","Biden",'Sanders','Harris','Buttigieg'])
).encode(
    alt.X("end_date:T",title=None),
    alt.Y("pct:Q", scale=alt.Scale(zero=False),title='Approval Rate'),
    alt.Color("answer:N"),
    #tooltip = ("pct", "end_date:T")
)

selectors = alt.Chart(CA_weekly).mark_point().encode(
    x='end_date:T',
    opacity=alt.value(0),
).add_selection(
    nearest
)

# Draw points on the line, and highlight based on selection
points = poll_CA.mark_point(filled = True,size=40).encode(
    opacity = alt.condition(nearest, alt.value(1), alt.value(0)),
    color=alt.value('black')
)

# Draw a rule at the location of the selection
vline = alt.Chart(CA_weekly).mark_rule(size=3,color="lightgray",opacity=1).encode(
    x='end_date:T',
).transform_filter(
    nearest
)

poll_CA2 = alt.Chart(df).mark_circle().transform_filter(
    (alt.datum.state == "California")
).encode(
    alt.X("end_date:T",title=None),
    alt.Y("pct:Q", scale=alt.Scale(zero=False),title='Approval Rate'),
    alt.Color("candidate_party:N",scale=alt.Scale(
            domain=['DEM', 'REP'],
            range=['lightblue', 'pink'])),
    #tooltip = ("pct", "end_date:T")
)

# Put the layers into a chart and bind the data
vis3 = alt.layer(
    poll_CA, selectors, vline, points
).add_selection(selection).encode(
    tooltip=['pct:Q','end_date:T','answer:N']
).properties(
    width=350,
    height=320
)
vis3
```




```
poll_CA2 = alt.Chart(df, width = 1000, height = 600).mark_circle().transform_filter(
    (alt.datum.state == "California")
).encode(
    alt.X("end_date:T",title=None),
    alt.Y("pct:Q", scale=alt.Scale(zero=False),title='Approval Rate'),
    alt.Color("candidate_party:N",scale=alt.Scale(
            domain=['DEM', 'REP'],
            range=['lightblue', 'pink'])),
    #tooltip = ("pct", "end_date:T")
)

poll_CA2
#I tried to add the scatter plot to vis3 as background to help reader have a better understand of the partis
```



2. Export to json

```
vis1.save('vis1.json')
vis2.save('vis2.json')
vis3.save('vis3.json')
```