Lab3 Report

> Course: 2021 HCI Lab3, School of Software Engineering, Tongji Univ.
>
> Data Visualization
>
> Name：沈益立
>
> Student Number: 1851009

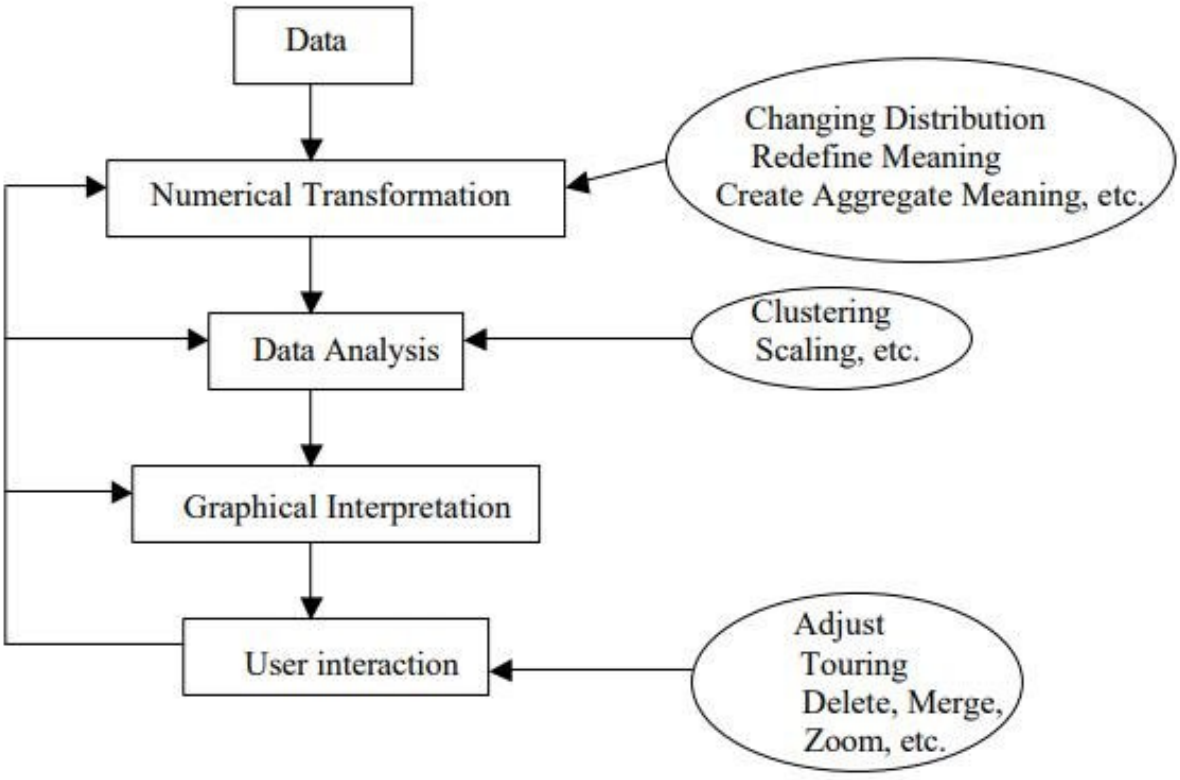
1. Introduction

This project is a cross-platform data visualization web page, based on dash for python, plotly engine, react front-end platform and flask for server deployment.

This project reads the data of graduated college students' salaries in different stages of their career life and their college propoties, and then visualize their relationships.

2. Architecture

2.1 Data Visualization



The process of data visualization can be shown as the picture. To be more specified, the process can be sum merized as:

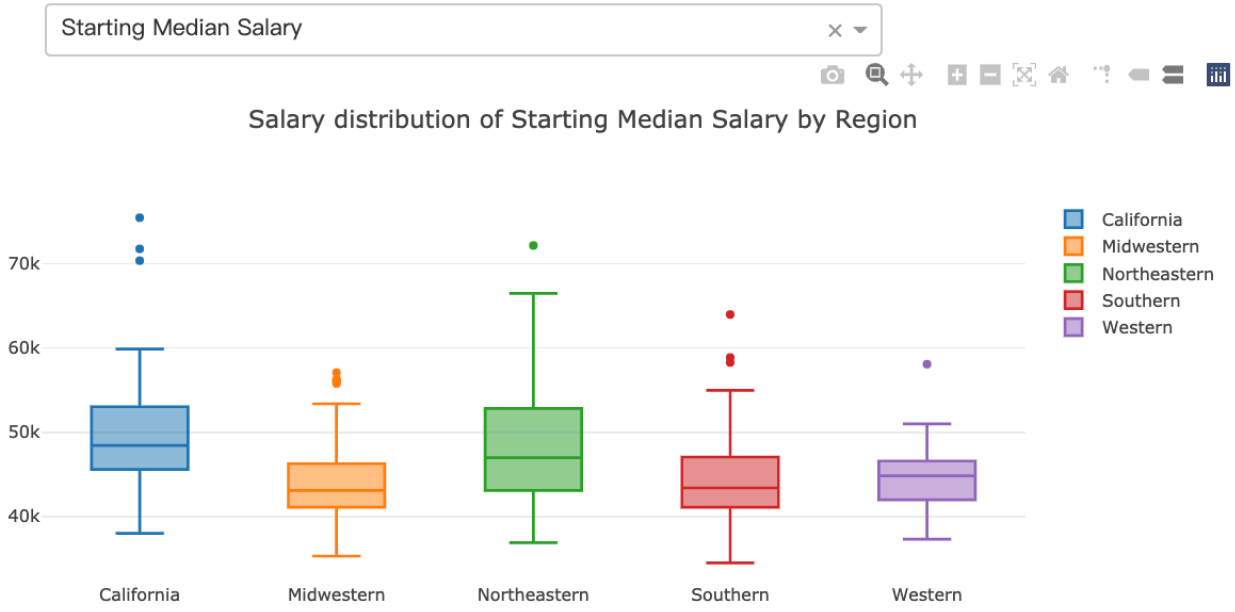
1. Read the raw data into memory
2. Pre-process the raw data(such as converting string values with specified pattern to bool or numerical values)
3. Find a proper pattern to express it, such as scatter or hist
4. Change the color fonts and other specified styles, to make the UI more friendly to users
5. Deal with the front-end arrangement, to make it easy to read.
6. Render and deploy on server.

3. GUI

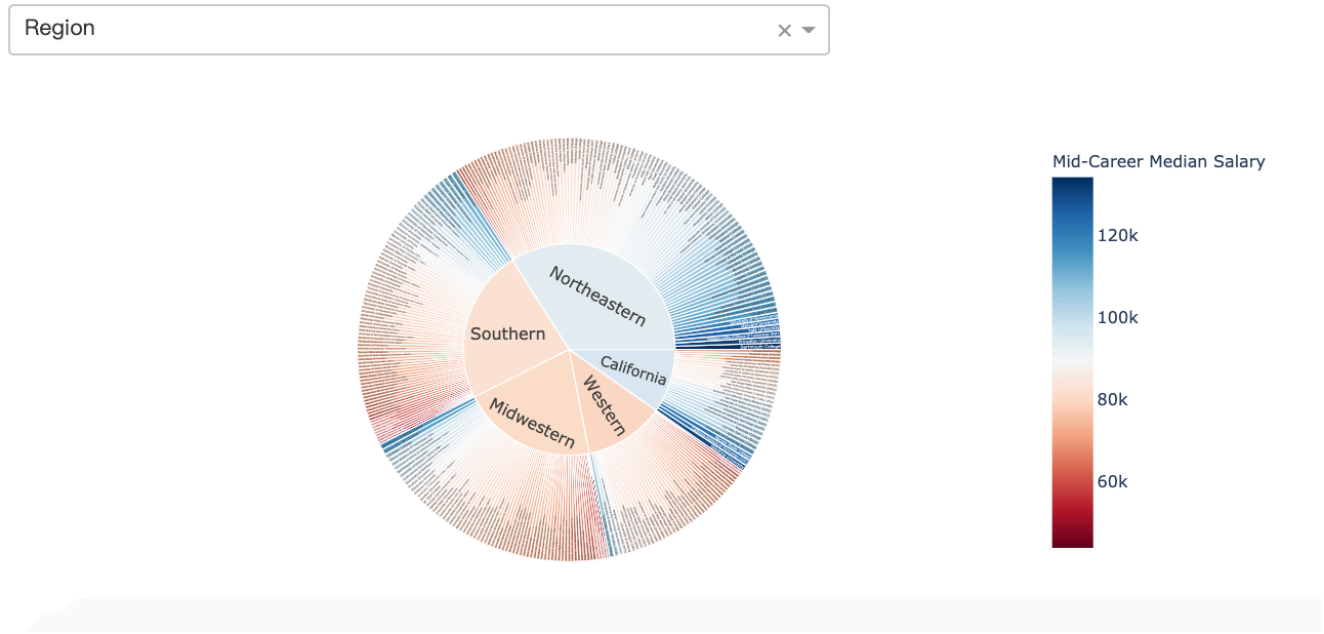
This project is based on dash, which is able to automatically render the front-end UI with React framework. And the programmer just need to bind certain data logics and the arrangement of HTML and components. In this project, I've created three parts of data visualization.

1. The scattering graph of average salaries of different career stages of graduated students distinguished by different properties.

2. The box graph of the distribution of certain career stage of graduated students distinguished by different properties.



3. The sunbrust graph of the distribution of students distinguished by different properties.



3.1 Code

```
app = dash.Dash()


app.layout = html.Div([
    html.Div([
        html.Div([
            html.Div([
                dcc.Dropdown(
                    id='filter',
                    options=[{'label': i, 'value': i} for i in ['Region',
'Undergraduate Major', 'School Type']],
                    value='Region',
                    style={'width': '80%'}
                ),

                dcc.Graph(id='avg_salary', figure=reader.get_salary_plot_by_region()),
                html.Br(),
            ], style={'width': '49%'}),
            html.Div([
                dcc.Dropdown(
                    id='stage-selector',
                    options=[{'label': i, 'value': i} for i in ['Starting Median
Salary',
                                                                'Mid-Career 10th
Percentile Salary',
```

```
                                                        'Mid-Career 25th
Percentile Salary',
                                                        'Mid-Career Median
Salary',
                                                        'Mid-Career 75th
Percentile Salary',
                                                        'Mid-Career 90th
Percentile Salary']],
                    value='Starting Median Salary',
                    style={'width': '80%', 'margin-left': '5%'}
                ),
                dcc.Graph(id='selected_part')
            ], style={'width': '49%'})
        ], style={'display': 'flex'}),
        html.Div([
            dcc.Dropdown(id='distribution-selector',
                         options=[{'label': i, 'value': i} for i in ['Region', 'School
Type']],
                         value='Region',
                         style={'width': '80%'}
                         ),

            dcc.Graph(id='distribution')
        ], style={'width': '50%'})
    ])
])
```

4 Functions and Code

4.1 Visualize average salaries at different career stages group by region / school types / undergraduate majors

Firstly, group the data by the corresponding properties, then get each numerical column's mean value. Then, feed them into a scattering plot, to draw the scattering points and use solid lines to fill them.

4.1.1 Code

```
def get_salary_plot_by_school_type():
    df = pd.read_csv('dataset/college-salaries/salaries-by-college-type.csv')
    for i in range(len(df.columns)):
        if df.columns[i] != 'School Name' and df.columns[i] != 'School Type':
            df[df.columns[i:]] = df[df.columns[i:]].replace('[\$,]', '',
regex=True).astype(float)

    group = df.groupby('School Type').mean() # type: pd.DataFrame
    data = []
```

```
    # print(group.columns)
    l = ['Starting Median Salary',
           'Mid-Career 10th Percentile Salary',
           'Mid-Career 25th Percentile Salary',
           'Mid-Career Median Salary',
           'Mid-Career 75th Percentile Salary',
           'Mid-Career 90th Percentile Salary']
    idx = [0, 2, 3, 1, 4, 5]
    # print((group.iloc[0].name))
    for i in range(len(group)):
        trace = go.Scatter(x=l, y=group.iloc[i][idx],
                           name=group.iloc[i].name,
                          showlegend=True)
        data.append(trace)
    layout = dict(                  title='Average salaries of different stages of
career by school type')

    # 将data与layout组合为一个图像
    fig = dict(data = data, layout = layout)
    return fig
# fig = get_salary_plot_by_school_type()
# iplot(fig)
def update_avg_fig(stage, filter):
    # ['Region', 'Undergraduate Major', 'School Type']
    if filter == 'Region':
        df = pd.read_csv('dataset/college-salaries/salaries-by-region.csv')
    elif filter == 'Undergraduate Major':
        df = pd.read_csv('dataset/college-salaries/degrees-that-pay-back.csv')
    else:
        df = pd.read_csv('dataset/college-salaries/salaries-by-college-type.csv')
    for i in range(len(df.columns)):
        if df.columns[i] != 'School Name' and df.columns[i] != filter:
            df[df.columns[i:]] = df[df.columns[i:]].replace('[\$,]', '',
regex=True).astype(float)
    data = []
    group = df.groupby(filter)
    for idx, item in group:
        trace = go.Box(y=item[stage], name=idx)
        data.append(trace)
    layout = dict(title='Average salaries of different stages of career by school
type')

    # 将data与layout组合为一个图像
    fig = dict(data=data, layout=layout)
    return fig

def get_salary_plot_by_degree():
    df = pd.read_csv('dataset/college-salaries/degrees-that-pay-back.csv')
    for i in range(len(df.columns)):
```

```
        if df.columns[i] != 'School Name' and df.columns[i] != 'Undergraduate Major':
            df[df.columns[i:]] = df[df.columns[i:]].replace('[\$,]', '',
regex=True).astype(float)

    group = df.groupby('Undergraduate Major').mean() # type: pd.DataFrame
    data = []
    # print(group.columns)
    l = ['Starting Median Salary',
           'Mid-Career 10th Percentile Salary',
           'Mid-Career 25th Percentile Salary',
           'Mid-Career Median Salary',
           'Mid-Career 75th Percentile Salary',
           'Mid-Career 90th Percentile Salary']
    idx = [0, 3, 4, 1, 5, 6]
    # print((group.iloc[0].name))
    for i in range(len(group)):
        trace = go.Scatter(x=l, y=group.iloc[i][idx],
                           name=group.iloc[i].name,
                          showlegend=True)
        data.append(trace)
    layout = dict(                  title='Average salaries of different stages of
career by degree major')

    # 将data与layout组合为一个图像
    fig = dict(data = data, layout = layout)
    return fig
```

4.2 Visualize the distribution of certain career stage of graduated students distinguished by different properties

Firstly, group the data by the corresponding properties, then get each numerical column. Draw the box graph for each property.

4.2.1 Code

```
@app.callback(
    dash.dependencies.Output('selected_part', 'figure'),
    [dash.dependencies.Input('stage-selector', 'value'),
     dash.dependencies.Input('filter', 'value')]
)
def update_avg_fig(stage, filter):
    # ['Region', 'Undergraduate Major', 'School Type']
    if filter == 'Region':
        df = pd.read_csv('dataset/college-salaries/salaries-by-region.csv')
    elif filter == 'Undergraduate Major':
```

```
        df = pd.read_csv('dataset/college-salaries/degrees-that-pay-back.csv')
    else:
        df = pd.read_csv('dataset/college-salaries/salaries-by-college-type.csv')
    for i in range(len(df.columns)):
        if df.columns[i] != 'School Name' and df.columns[i] != filter:
            df[df.columns[i:]] = df[df.columns[i:]].replace('[\$,]', '',
regex=True).astype(float)
    data = []
    group = df.groupby(filter)
    for idx, item in group:
        trace = go.Box(y=item[stage], name=idx)
        data.append(trace)
    layout = dict(title='Salary distribution of ' + stage + ' by ' + filter)

    # 将data与layout组合为一个图像
    fig = dict(data=data, layout=layout)
    return fig
```

4.3 Visualize the distribution(region / school type) of origin data

Using the sunbrust graph to visualize the distribution of students distinguished by different properties.

4.3.1 Code

```
def get_sunbrast():
    df = pd.read_csv('dataset/college-salaries/salaries-by-region.csv')
    for i in range(len(df.columns)):
        if df.columns[i] != 'School Name' and df.columns[i] != 'Region':
            df[df.columns[i:]] = df[df.columns[i:]].replace('[\$,]', '',
regex=True).astype(float)

    fig = px.sunburst(df, path=['Region', 'School Name'], values='Mid-Career Median
Salary',
                      color='Mid-Career Median Salary',
                      color_continuous_scale='RdBu'
                      )

    return fig
def get_sunbrast_by_school_type():
    df = pd.read_csv('dataset/college-salaries/salaries-by-college-type.csv')
    for i in range(len(df.columns)):
        if df.columns[i] != 'School Name' and df.columns[i] != 'School Type':
            df[df.columns[i:]] = df[df.columns[i:]].replace('[\$,]', '',
regex=True).astype(float)
```

```
    fig = px.sunburst(df, path=['School Type', 'School Name'], values='Mid-Career
Median Salary',
                      color='Mid-Career Median Salary',
                      color_continuous_scale='RdBu'
                      )

    return fig
```

5. How To Run My Code?

1. 开发环境为Python3.9 + dash + plotly
2. 安装好对应环境后，输入命令`python3 ./app.py`，将服务器在本地持久化运行之后，进入`http://localhost:8050/`，即可浏览页面。您可以选择对应的标签来查看对应的可视化数据。