IST-719

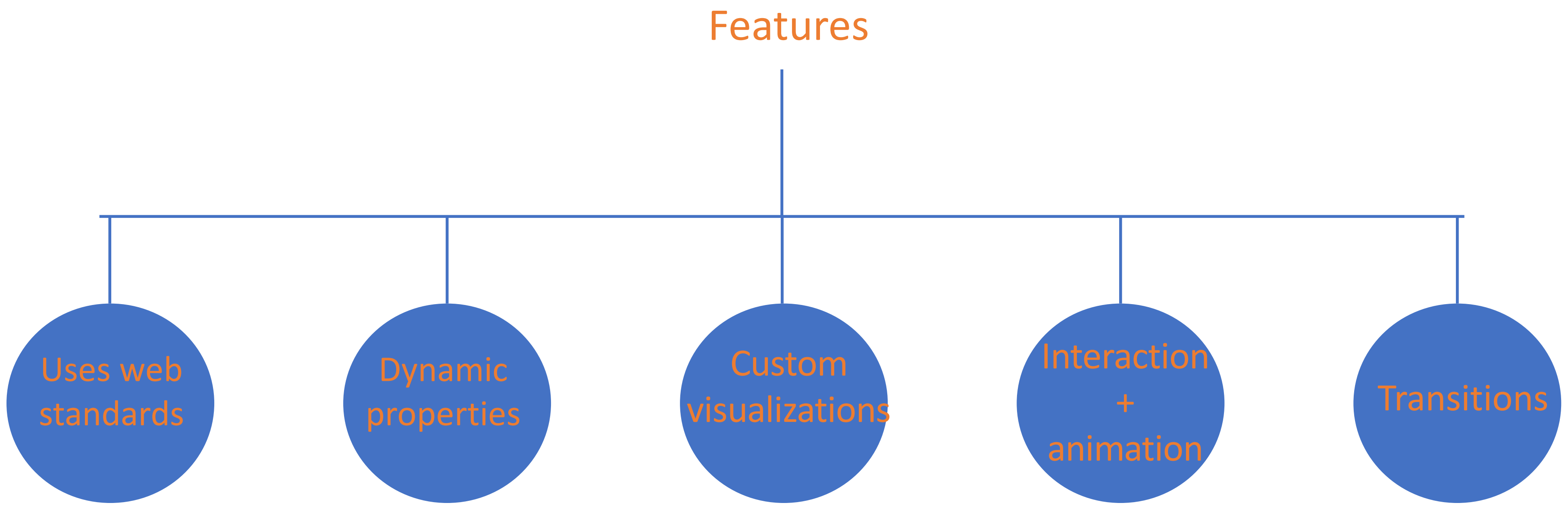
Advanced Topic Presentation (Week-8)

Topic: Using D3.js to visualize the wine sales dataset

Sharat Sripada (vssripad@syr.edu)

D3 - Introduction

D3 stands for Data-Driven Documents. It is an open-source JavaScript library developed to create custom interactive data visualizations in web-browser

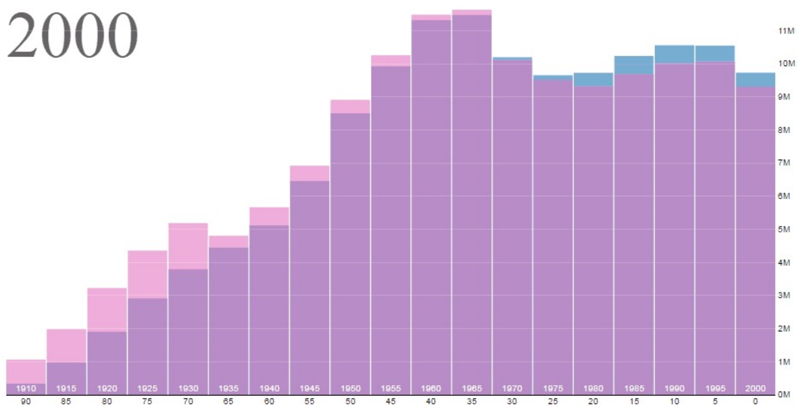


D3 - Advantages

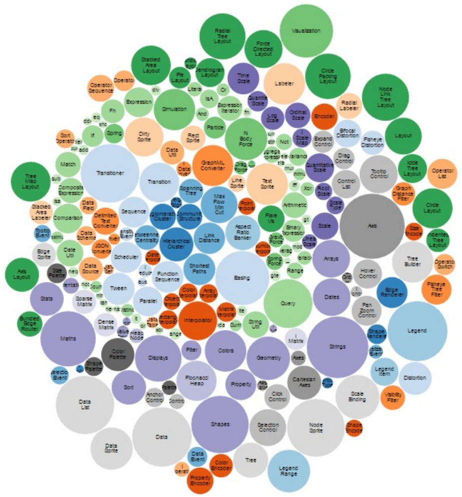
Attribute	**Description**
D3 is a javascript (JS) library	It can be used with any JS framework
D3 focuses on data	It is the most appropriate and specialized tool for data visualizations
D3 is open-source	Add your own features
It works with web standards	You don't need any other technology or plugin other than a browser to make use of D3
D3 works with web standards like HTML, CSS and SVG	No new learning or debugging tool required
D3 does not provide any specific feature	Gives you complete control over your visualization to customize it the way you want. This gives it an edge over other popular tools like Tableau or QlikView.
D3 is lightweight, and works directly with web standards	Extremely fast and works well with large datasets.

D3 - Visualization examples

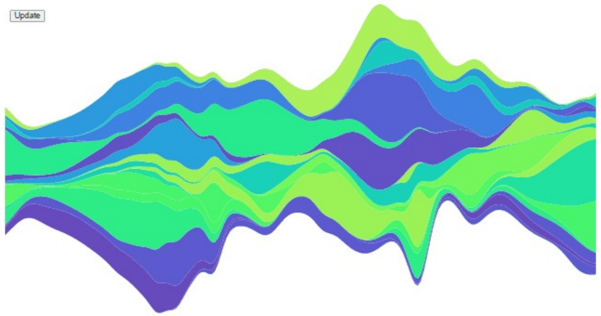
Bar Chart



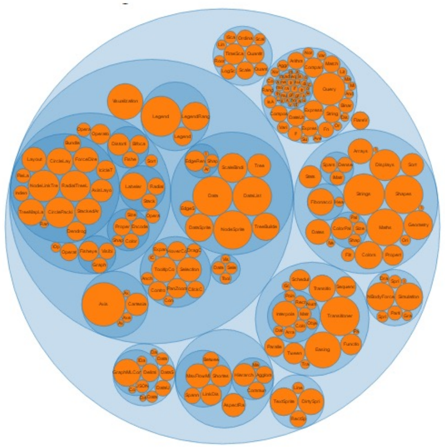
Bubble Chart



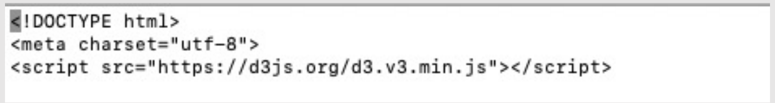
Stream Graph



Circle Packing



Building a simple dashboard with Python and D3.js



```
<!DOCTYPE html>
<meta charset="utf-8">
<script src="https://d3js.org/d3.v3.min.js"></script>
```

Fig: index.html showing d3.js invocation

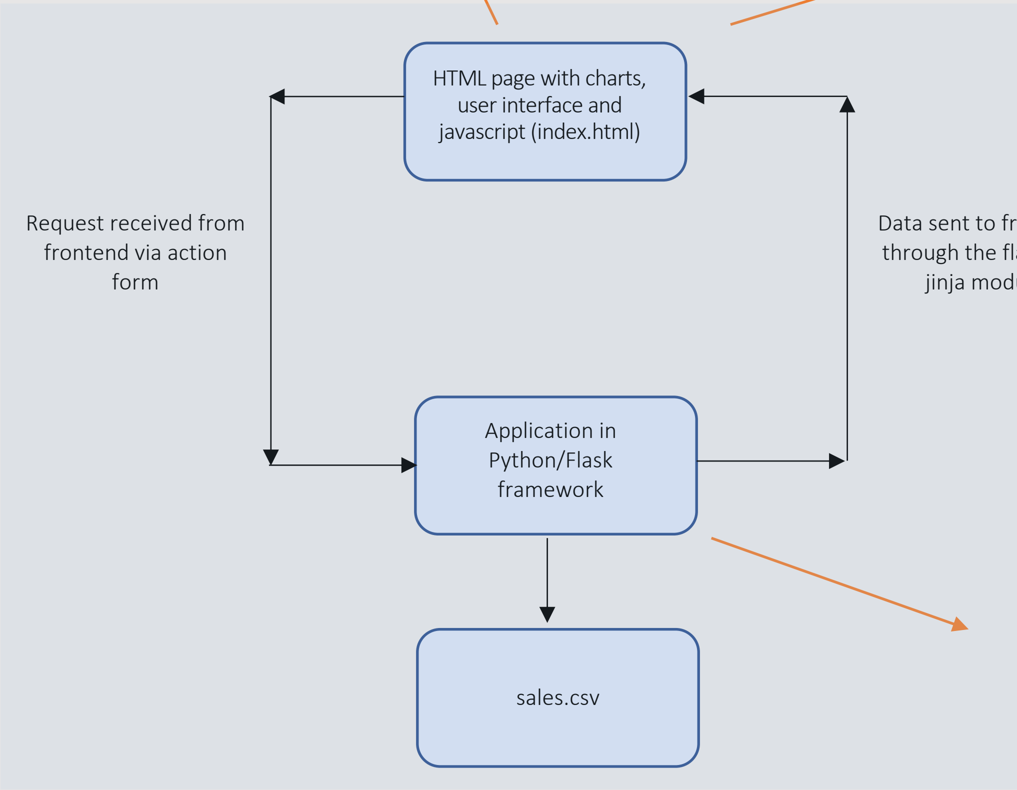
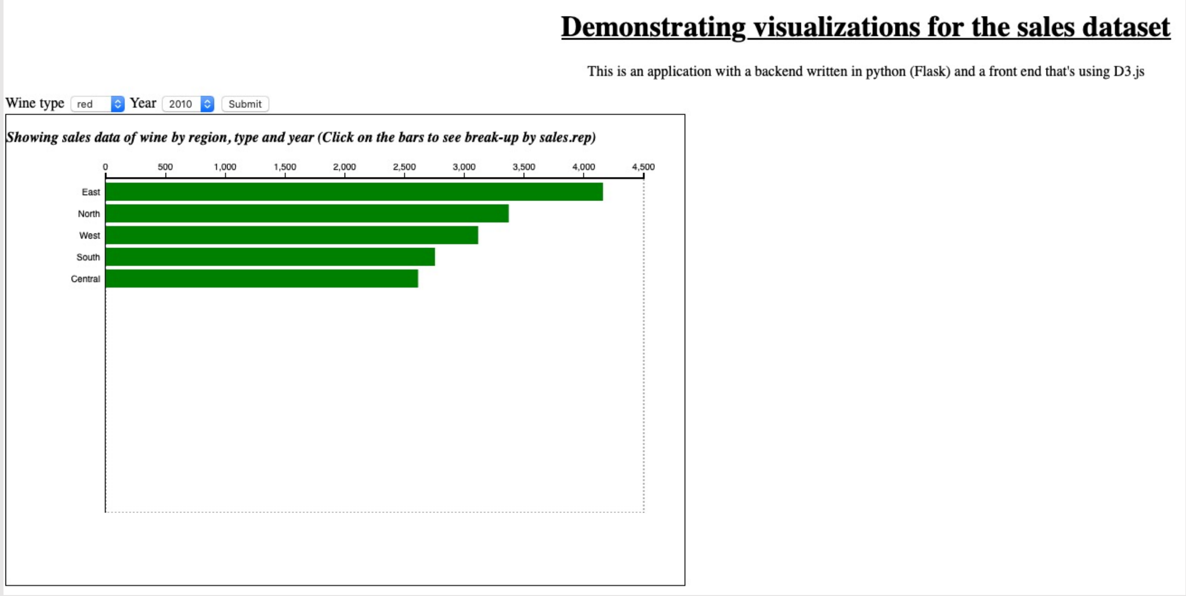

HTML page with charts, user interface and javascript (index.html)
Request received from frontend via action form
Data sent to front end through the flask and jinja module
Application in Python/Flask framework
sales.csv

Fig: High-level architecture



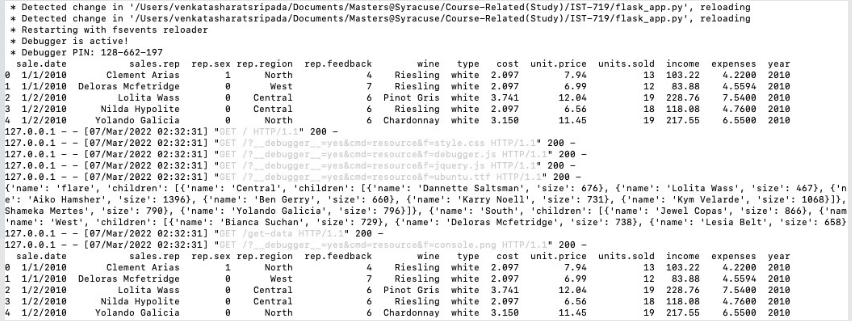
Demonstrating visualizations for the sales dataset

This is an application with a backend written in python (Flask) and a front end that's using D3.js

Wine type red Year 2010 Submit

Showing sales data of wine by region, type and year (Click on the bars to see break-up by sales.rep)

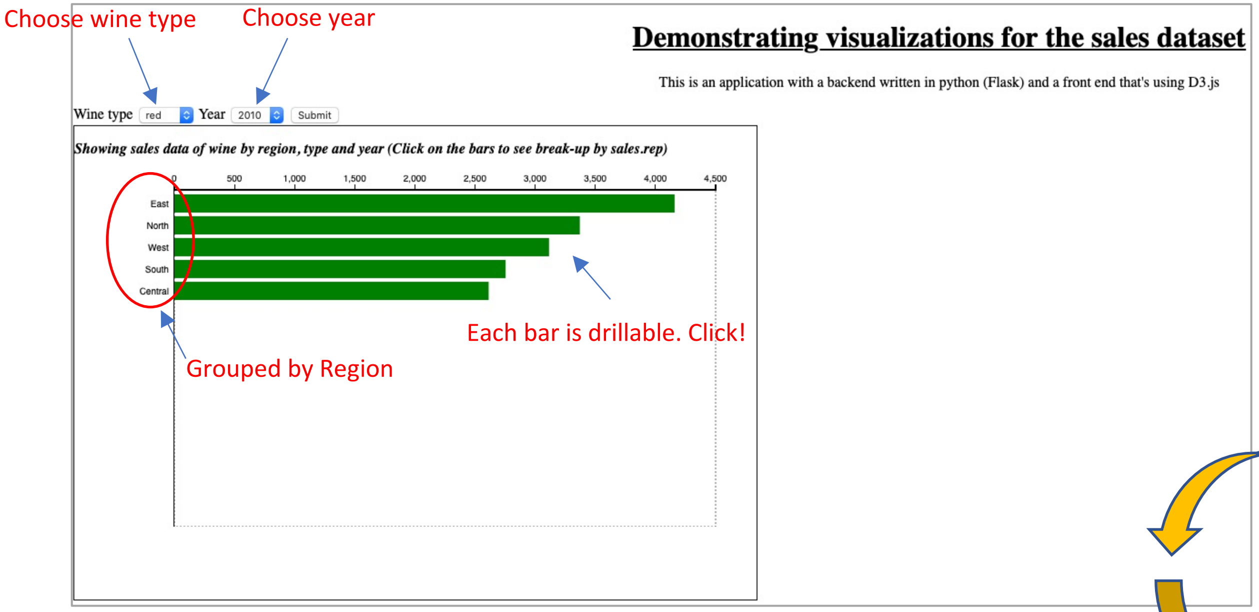
Fig: Front-end GUI with d3.js charts



```
 * Detected change in '/Users/venkatasharatsripada/Documents/Masters@Syracuse/Course-Related(Study)/IST-719/flask_app.py', reloading
 * Detected change in '/Users/venkatasharatsripada/Documents/Masters@Syracuse/Course-Related(Study)/IST-719/flask_app.py', reloading
 * Restarting with fsevents reloader
 * Debugger is active!
 * Debugger PIN: 128-662-197
  sale.date           sales.rep  rep.sex rep.region  rep.feedback        wine  type   cost  unit.price  units.sold  income  expenses  year
0  1/1/2010       Clement Arias        1      North             4    Riesling white  2.097        7.94          13  103.22    4.2200  2010
1  1/1/2010  Deloras Mcfetridge        0       West             7    Riesling white  2.097        6.99          12   83.88    4.5594  2010
2  1/2/2010         Lolita Wass        0    Central             6  Pinot Gris white  3.741       12.04          19  228.76    7.5400  2010
3  1/2/2010      Nilda Hypolite        0    Central             6    Riesling white  2.097        6.56          18  118.08    4.7600  2010
4  1/2/2010     Yolando Galicia        0      North             6  Chardonnay white  3.150       11.45          19  217.55    6.5500  2010
127.0.0.1 - - [07/Mar/2022 02:32:31] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [07/Mar/2022 02:32:31] "GET /?__debugger__=yes&cmd=resource&f=style.css HTTP/1.1" 200 -
127.0.0.1 - - [07/Mar/2022 02:32:31] "GET /?__debugger__=yes&cmd=resource&f=debugger.js HTTP/1.1" 200 -
127.0.0.1 - - [07/Mar/2022 02:32:31] "GET /?__debugger__=yes&cmd=resource&f=jquery.js HTTP/1.1" 200 -
127.0.0.1 - - [07/Mar/2022 02:32:31] "GET /?__debugger__=yes&cmd=resource&f=ubuntu.ttf HTTP/1.1" 200 -
{'name': 'flare', 'children': [{'name': 'Central', 'children': [{'name': 'Dannette Saltsman', 'size': 676}, {'name': 'Lolita Wass', 'size': 467}, {'n
e': 'Aiko Hamsher', 'size': 1396}, {'name': 'Ben Gerry', 'size': 660}, {'name': 'Karry Noell', 'size': 731}, {'name': 'Kym Velarde', 'size': 1068}]},
Shameka Mertes', 'size': 790}, {'name': 'Yolando Galicia', 'size': 796}]}, {'name': 'South', 'children': [{'name': 'Jewel Copas', 'size': 866}, {'nam
'name': 'West', 'children': [{'name': 'Bianca Suchan', 'size': 729}, {'name': 'Deloras Mcfetridge', 'size': 738}, {'name': 'Lesia Belt', 'size': 658}
127.0.0.1 - - [07/Mar/2022 02:32:31] "GET /get-data HTTP/1.1" 200 -
127.0.0.1 - - [07/Mar/2022 02:32:31] "GET /?__debugger__=yes&cmd=resource&f=console.png HTTP/1.1" 200 -
  sale.date           sales.rep  rep.sex rep.region  rep.feedback        wine  type   cost  unit.price  units.sold  income  expenses  year
0  1/1/2010       Clement Arias        1      North             4    Riesling white  2.097        7.94          13  103.22    4.2200  2010
1  1/1/2010  Deloras Mcfetridge        0       West             7    Riesling white  2.097        6.99          12   83.88    4.5594  2010
2  1/2/2010         Lolita Wass        0    Central             6  Pinot Gris white  3.741       12.04          19  228.76    7.5400  2010
3  1/2/2010      Nilda Hypolite        0    Central             6    Riesling white  2.097        6.56          18  118.08    4.7600  2010
4  1/2/2010     Yolando Galicia        0      North             6  Chardonnay white  3.150       11.45          19  217.55    6.5500  2010
```

Fig: Python Flask app in action

Demo



Choose wine type

Choose year

Demonstrating visualizations for the sales dataset

This is an application with a backend written in python (Flask) and a front end that's using D3.js

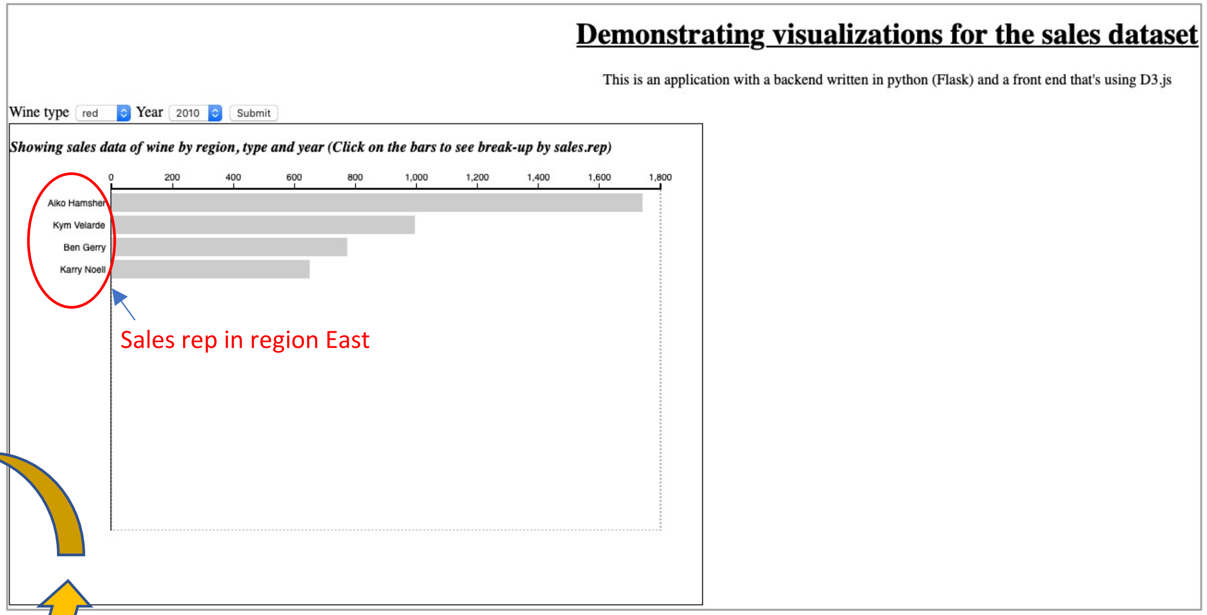
Wine type red Year 2010 Submit

Showing sales data of wine by region, type and year (Click on the bars to see break-up by sales.rep)

Each bar is drillable. Click!

Grouped by Region

Fig: Home page (front-end)



Demonstrating visualizations for the sales dataset

This is an application with a backend written in python (Flask) and a front end that's using D3.js

Wine type red Year 2010 Submit

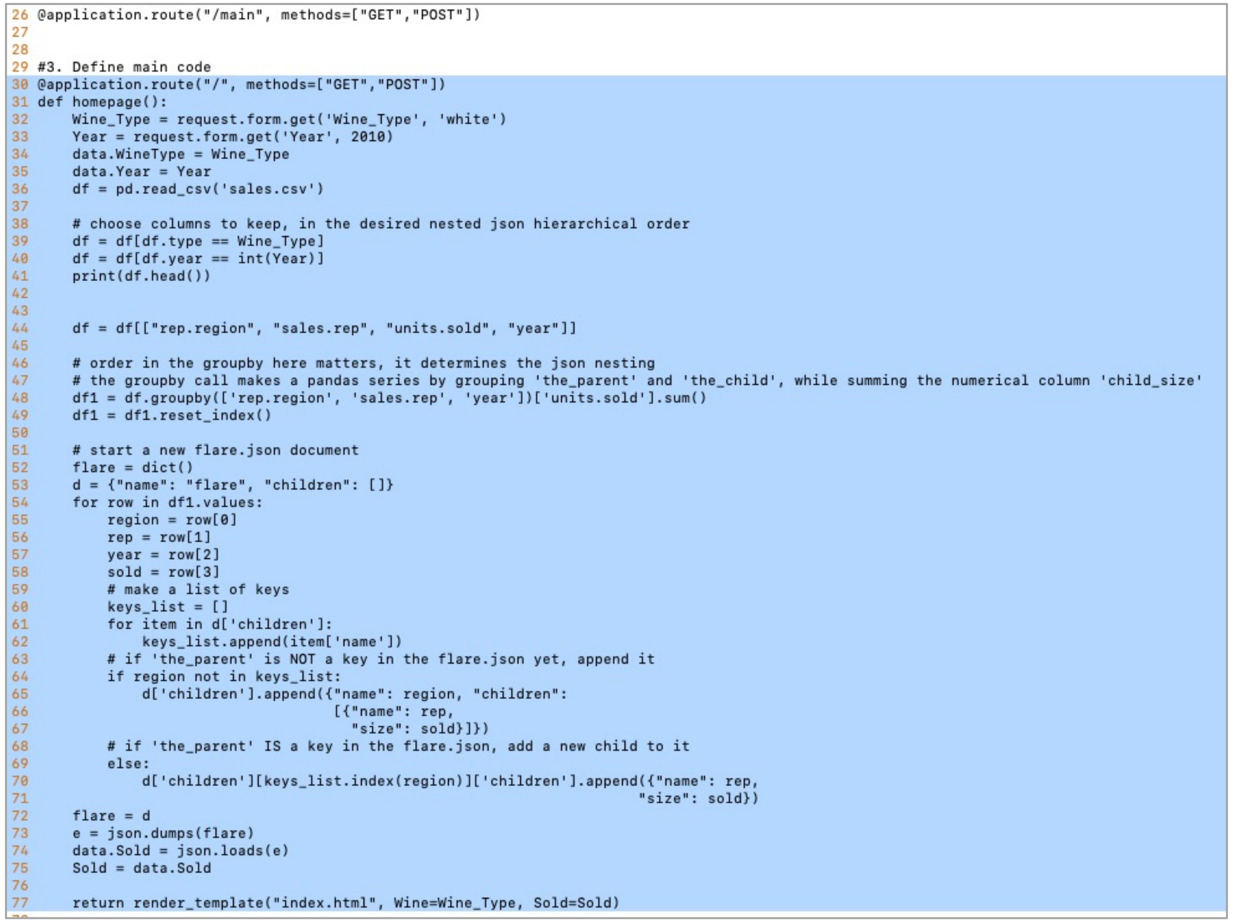
Showing sales data of wine by region, type and year (Click on the bars to see break-up by sales.rep)

Sales rep in region East

Fig: Children under parent

Interact with bar-plot to navigate between parent/child objects

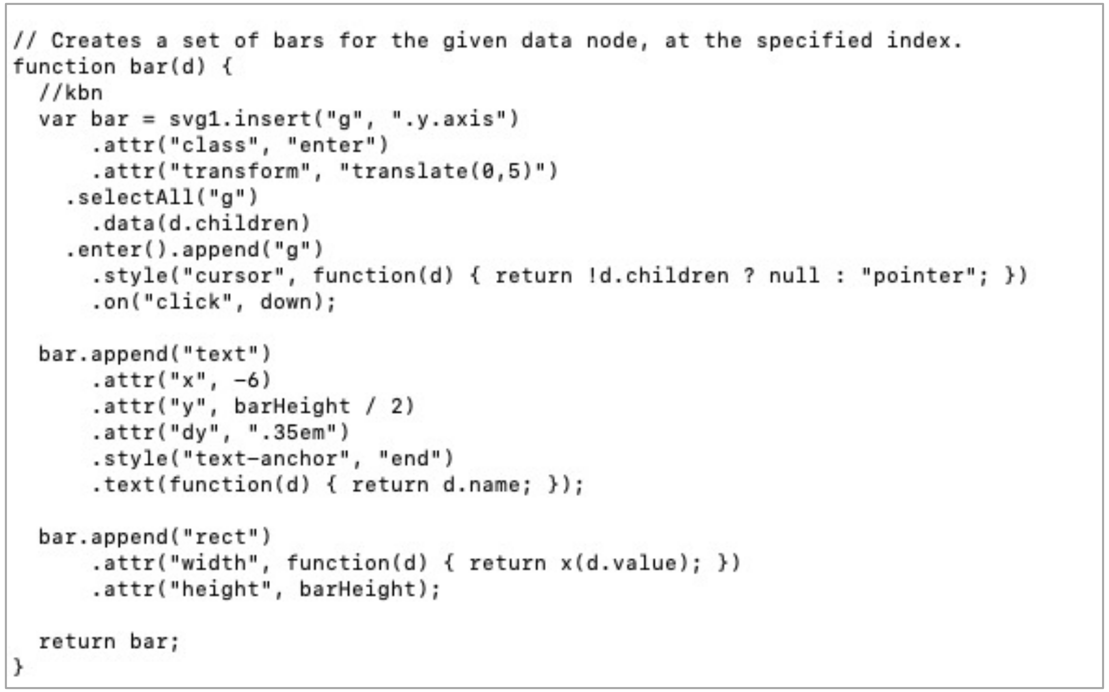
Demo



```python
@application.route("/main", methods=["GET","POST"])


#3. Define main code
@application.route("/", methods=["GET","POST"])
def homepage():
    Wine_Type = request.form.get('Wine_Type', 'white')
    Year = request.form.get('Year', 2010)
    data.WineType = Wine_Type
    data.Year = Year
    df = pd.read_csv('sales.csv')

    # choose columns to keep, in the desired nested json hierarchical order
    df = df[df.type == Wine_Type]
    df = df[df.year == int(Year)]
    print(df.head())


    df = df[["rep.region", "sales.rep", "units.sold", "year"]]

    # order in the groupby here matters, it determines the json nesting
    # the groupby call makes a pandas series by grouping 'the_parent' and 'the_child', while summing the numerical column 'child_size'
    df1 = df.groupby(['rep.region', 'sales.rep', 'year'])['units.sold'].sum()
    df1 = df1.reset_index()

    # start a new flare.json document
    flare = dict()
    d = {"name": "flare", "children": []}
    for row in df1.values:
        region = row[0]
        rep = row[1]
        year = row[2]
        sold = row[3]
        # make a list of keys
        keys_list = []
        for item in d['children']:
            keys_list.append(item['name'])
        # if 'the_parent' is NOT a key in the flare.json yet, append it
        if region not in keys_list:
            d['children'].append({"name": region, "children":
                                  [{"name": rep,
                                    "size": sold}]})
        # if 'the_parent' IS a key in the flare.json, add a new child to it
        else:
            d['children'][keys_list.index(region)]['children'].append({"name": rep,
                                                                  "size": sold})
    flare = d
    e = json.dumps(flare)
    data.Sold = json.loads(e)
    Sold = data.Sold

    return render_template("index.html", Wine=Wine_Type, Sold=Sold)
```

Fig: Python code to convert data to json for D3

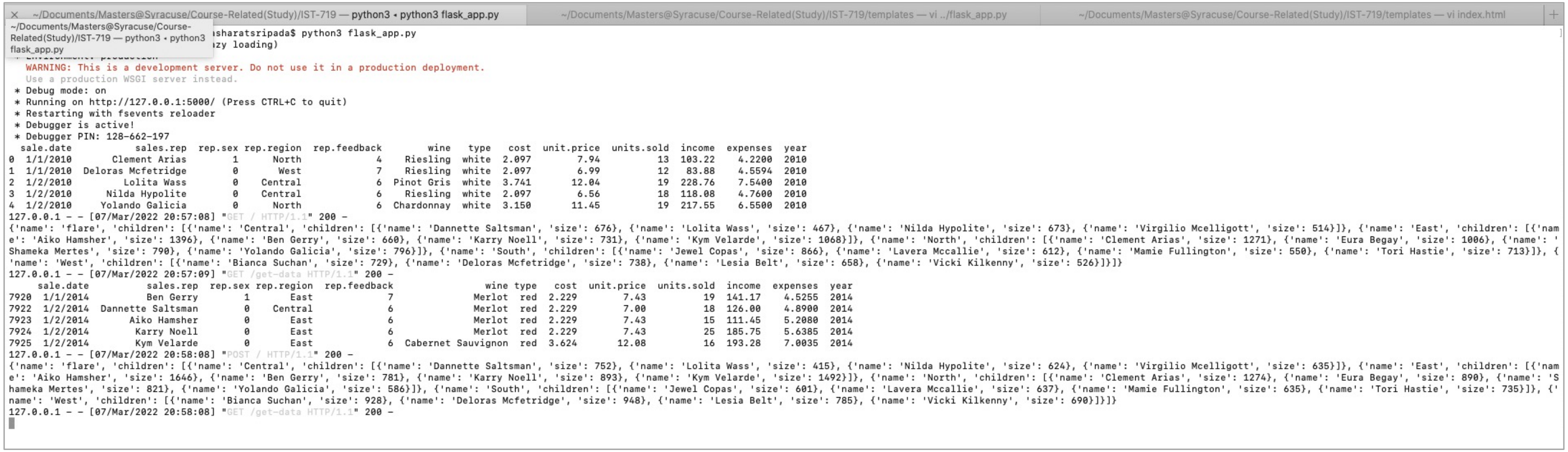
Demo



```
// Creates a set of bars for the given data node, at the specified index.
function bar(d) {
  //kbn
  var bar = svg1.insert("g", ".y.axis")
      .attr("class", "enter")
      .attr("transform", "translate(0,5)")
    .selectAll("g")
      .data(d.children)
    .enter().append("g")
      .style("cursor", function(d) { return !d.children ? null : "pointer"; })
      .on("click", down);

  bar.append("text")
      .attr("x", -6)
      .attr("y", barHeight / 2)
      .attr("dy", ".35em")
      .style("text-anchor", "end")
      .text(function(d) { return d.name; });

  bar.append("rect")
      .attr("width", function(d) { return x(d.value); })
      .attr("height", barHeight);

  return bar;
}
```

Fig: Code in D3 to make an interactive bar-plot

Demo


~/Documents/Masters@Syracuse/Course-Related(Study)/IST-719 — python3 ◂ python3 flask_app.py
~/Documents/Masters@Syracuse/Course-Related(Study)/IST-719/templates — vi ../flask_app.py
~/Documents/Masters@Syracuse/Course-Related(Study)/IST-719/templates — vi index.html
~/Documents/Masters@Syracuse/Course-Related(Study)/IST-719 — python3 ◂ python3 flask_app.py

```
sharatsripada$ python3 flask_app.py
zy loading)
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
 * Restarting with fsevents reloader
 * Debugger is active!
 * Debugger PIN: 128-662-197
  sale.date          sales.rep  rep.sex rep.region  rep.feedback        wine  type   cost  unit.price  units.sold  income  expenses  year
0  1/1/2010      Clement Arias        1      North             4    Riesling  white  2.097        7.94          13  103.22    4.2200  2010
1  1/1/2010  Deloras Mcfetridge       0       West             7    Riesling  white  2.097        6.99          12   83.88    4.5594  2010
2  1/2/2010         Lolita Wass       0    Central             6  Pinot Gris  white  3.741       12.04          19  228.76    7.5400  2010
3  1/2/2010      Nilda Hypolite       0    Central             6    Riesling  white  2.097        6.56          18  118.08    4.7600  2010
4  1/2/2010     Yolando Galicia       0      North             6  Chardonnay  white  3.150       11.45          19  217.55    6.5500  2010
127.0.0.1 - - [07/Mar/2022 20:57:08] "GET / HTTP/1.1" 200 -
{'name': 'flare', 'children': [{'name': 'Central', 'children': [{'name': 'Dannette Saltsman', 'size': 676}, {'name': 'Lolita Wass', 'size': 467}, {'name': 'Nilda Hypolite', 'size': 673}, {'name': 'Virgilio Mcelligott', 'size': 514}]}, {'name': 'East', 'children': [{'nam
e': 'Aiko Hamsher', 'size': 1396}, {'name': 'Ben Gerry', 'size': 660}, {'name': 'Karry Noell', 'size': 731}, {'name': 'Kym Velarde', 'size': 1068}]}, {'name': 'North', 'children': [{'name': 'Clement Arias', 'size': 1271}, {'name': 'Eura Begay', 'size': 1006}, {'name': '
Shameka Mertes', 'size': 790}, {'name': 'Yolando Galicia', 'size': 796}]}, {'name': 'South', 'children': [{'name': 'Jewel Copas', 'size': 866}, {'name': 'Lavera Mccallie', 'size': 612}, {'name': 'Mamie Fullington', 'size': 550}, {'name': 'Tori Hastie', 'size': 713}]}, {
'name': 'West', 'children': [{'name': 'Bianca Suchan', 'size': 729}, {'name': 'Deloras Mcfetridge', 'size': 738}, {'name': 'Lesia Belt', 'size': 658}, {'name': 'Vicki Kilkenny', 'size': 526}]}]}
127.0.0.1 - - [07/Mar/2022 20:57:09] "GET /get-data HTTP/1.1" 200 -
     sale.date          sales.rep  rep.sex rep.region  rep.feedback                wine type   cost  unit.price  units.sold  income  expenses  year
7920  1/1/2014          Ben Gerry        1       East             7              Merlot  red  2.229        7.43          19  141.17    4.5255  2014
7922  1/2/2014  Dannette Saltsman        0    Central             6              Merlot  red  2.229        7.00          18  126.00    4.8900  2014
7923  1/2/2014        Aiko Hamsher       0       East             6              Merlot  red  2.229        7.43          15  111.45    5.2080  2014
7924  1/2/2014         Karry Noell       0       East             6              Merlot  red  2.229        7.43          25  185.75    5.6385  2014
7925  1/2/2014         Kym Velarde       0       East             6  Cabernet Sauvignon  red  3.624       12.08          16  193.28    7.0035  2014
127.0.0.1 - - [07/Mar/2022 20:58:08] "POST / HTTP/1.1" 200 -
{'name': 'flare', 'children': [{'name': 'Central', 'children': [{'name': 'Dannette Saltsman', 'size': 752}, {'name': 'Lolita Wass', 'size': 415}, {'name': 'Nilda Hypolite', 'size': 624}, {'name': 'Virgilio Mcelligott', 'size': 635}]}, {'name': 'East', 'children': [{'nam
e': 'Aiko Hamsher', 'size': 1646}, {'name': 'Ben Gerry', 'size': 781}, {'name': 'Karry Noell', 'size': 893}, {'name': 'Kym Velarde', 'size': 1492}]}, {'name': 'North', 'children': [{'name': 'Clement Arias', 'size': 1274}, {'name': 'Eura Begay', 'size': 890}, {'name': 'S
hameka Mertes', 'size': 821}, {'name': 'Yolando Galicia', 'size': 586}]}, {'name': 'South', 'children': [{'name': 'Jewel Copas', 'size': 601}, {'name': 'Lavera Mccallie', 'size': 637}, {'name': 'Mamie Fullington', 'size': 635}, {'name': 'Tori Hastie', 'size': 735}]}, {'
name': 'West', 'children': [{'name': 'Bianca Suchan', 'size': 928}, {'name': 'Deloras Mcfetridge', 'size': 948}, {'name': 'Lesia Belt', 'size': 785}, {'name': 'Vicki Kilkenny', 'size': 690}]}]}
127.0.0.1 - - [07/Mar/2022 20:58:08] "GET /get-data HTTP/1.1" 200 -
```

Fig: Application (back-end) in action

Thank you!

Sharat Sripada (vssripad@syr.edu)