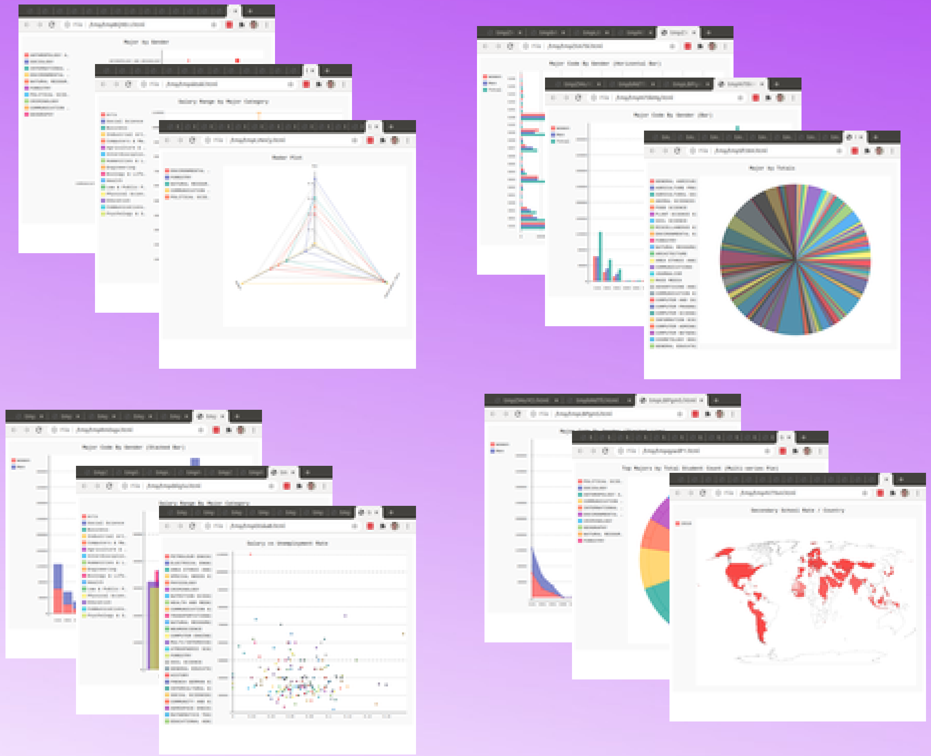
Data Visualization with Pygal

pymntos



FSK INC.
CONSULTING

What is Pygal?

- Python module that creates interactive Scalable Vector Graphics (SVG) graphs/charts
- One of many data visualization modules (e.g. Matplotlib, Seaborn, Bokeh, …)
- In search of honing my data visualization chops, a course in Coursera introduced this module
- Simple, interactive graph/chart, readily integrated in web user interfaces and web pages

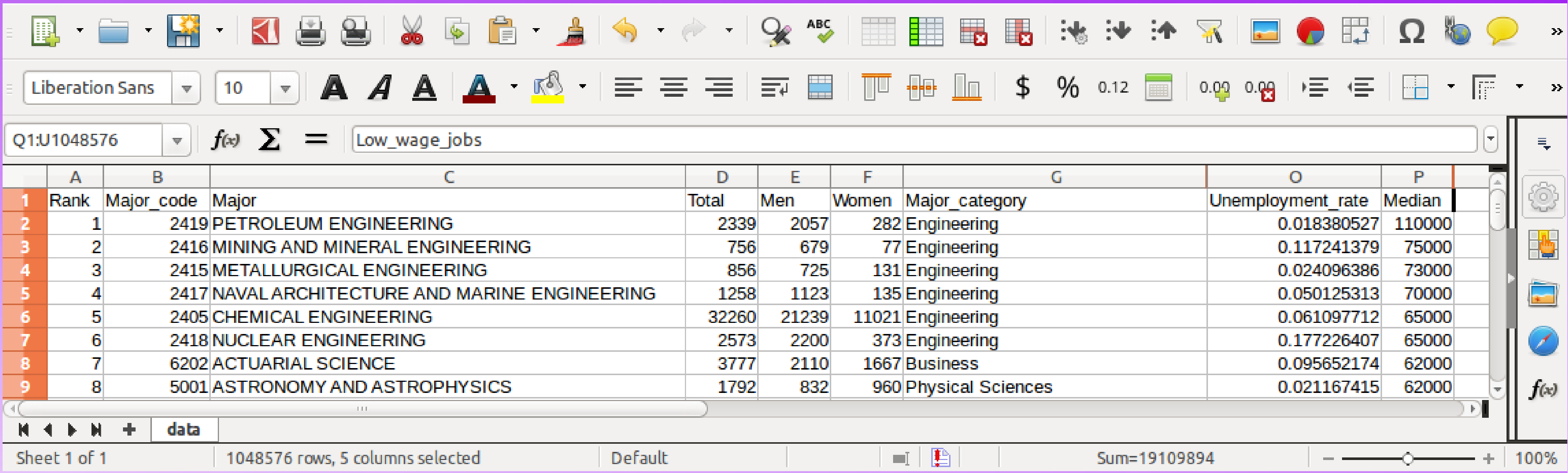
Agenda

- What is Pygal?
- Chart/Graph Example Sampler
- Performing data analysis on debugging logs to attain system performance/behaviors has been an emphasis on last couple contracts
- ‘Visualization’ of even modest data sets gives us a better understanding of the collective

Example Data Set

- https://raw.githubusercontent.com/fivethirtyeight/data/master/college-majors/recent-grads.csv
- FiveThirtyEight
  - The Economic Guide to Picking A College Major
    - https://fivethirtyeight.com/features/the-economic-guide-to-picking-a-college-major/
  - Just an interesting dataset; not and endorsement of the paper
  - Wanted a useful dataset that could be used to demonstrate a variety of means of plotting

Data Overview



Liberation Sans 10

Q1:U1048576 Low_wage_jobs

	A	B	C	D	E	F	G	O	P
1	Rank	Major_code	Major	Total	Men	Women	Major_category	Unemployment_rate	Median
2	1	2419	PETROLEUM ENGINEERING	2339	2057	282	Engineering	0.018380527	110000
3	2	2416	MINING AND MINERAL ENGINEERING	756	679	77	Engineering	0.117241379	75000
4	3	2415	METALLURGICAL ENGINEERING	856	725	131	Engineering	0.024096386	73000
5	4	2417	NAVAL ARCHITECTURE AND MARINE ENGINEERING	1258	1123	135	Engineering	0.050125313	70000
6	5	2405	CHEMICAL ENGINEERING	32260	21239	11021	Engineering	0.061097712	65000
7	6	2418	NUCLEAR ENGINEERING	2573	2200	373	Engineering	0.177226407	65000
8	7	6202	ACTUARIAL SCIENCE	3777	2110	1667	Business	0.095652174	62000
9	8	5001	ASTRONOMY AND ASTROPHYSICS	1792	832	960	Physical Sciences	0.021167415	62000

data

Sheet 1 of 1 | 1048576 rows, 5 columns selected | Default | Sum=19109894 | 100%

- 174 Rows, 21 Columns of data organized by university major
- Focus out attention on 9 key columns in our examples

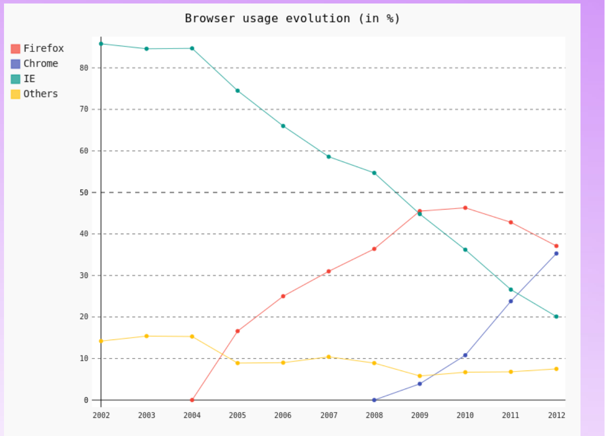
FSK INC. CONSULTING

CSV File Reader

- `def readCsvAsDict(fileName, keyField, separator=',', quote='"'):`
- `data = readCsvAsDict(‘data.csv’,keyField=’Major_code’)`
  - Returns dictionary, keyed by ‘Major_code’ column value, value is dictionary of all column field names
    - {"1301",
    - {
    - "Major":"ENVIRONMENTAL SCIENCE",
    - "Men":"10787",
    - "Unemployment_rate":"0.078584681",
    - "Major_code":"1301",
    - "Median":"35600",
    - "Rank":"93",
    - "Major_category":"Biology & Life Science",
    - "Women":"15178"
    - }
    - ...
    - }

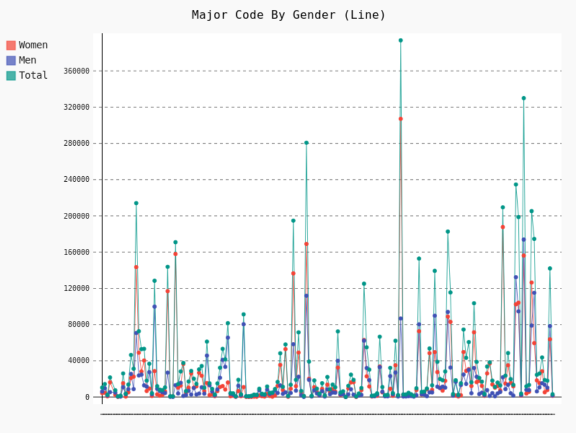
Simple Pygal Example

```
$ cat -n example.py
1     #!/usr/bin/python3
2     import pygal
3     import csv
4     chart = pygal.Line()
5     chart.title = 'Browser usage evolution (in %)'
6     chart.x_labels = map(str, range(2002, 2013))
7     chart.add('Firefox', [None, None,    0, 16.6,   25,   31, 36.4, 45.5, 46.3, 42.8, 37.1])
8     chart.add('Chrome',  [None, None, None, None, None, None,    0,  3.9, 10.8, 23.8, 35.3])
9     chart.add('IE',      [85.8, 84.6, 84.7, 74.5,   66, 58.6, 54.7, 44.8, 36.2, 26.6, 20.1])
10    chart.add('Others',  [14.2, 15.4, 15.3,  8.9,    9, 10.4,  8.9,  5.8,  6.7,  6.8,  7.5])
11    chart.render_in_browser()
```



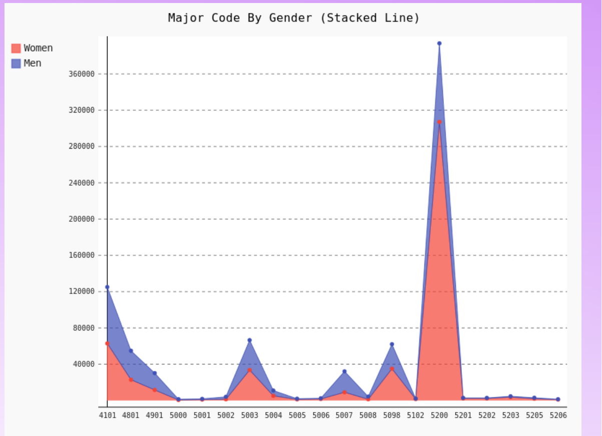
Line

```
$ cat example.py
#!/usr/bin/python3
import pygal
import csv
data=readCsvAsDict('data.csv','Major_code')
Fields=['Women','Men','Total']
plotData=dict()
for key in Fields:
  D=[(k,v[key]) for (k,v) in sorted(data.items())]
  L=([int(el[1]) if el[1].isdigit() else None for el in D])
  xLabel=([el[0] for el in D])
  plotData[key]=L
chart=pygal.Line()
chart.title='Major Code By Gender (Line)'
for key in Fields:
  chart.add(key,plotData[key])
chart.x_labels = xLabel
chart.render_in_browser()
```



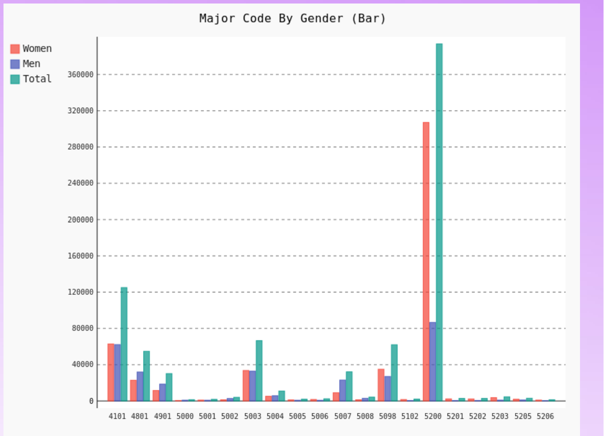
Stacked Line

```
#!/usr/bin/python3
import pygal
import csv
data=readCsvAsDict('data.csv','Major_code')
Fields=['Women','Men']
plotData=dict()
for key in Fields:
  D=[(k,v[key]) for (k,v) in sorted(data.items())[100:120]]
  L=([int(el[1]) if el[1].isdigit() else None for el in D])
  xLabel=([el[0] for el in D])
  plotData[key]=L
chart=pygal.StackedLine(fill=True)
chart.title='Major Code By Gender (Stacked Line)'
for key in Fields:
  chart.add(key,plotData[key])
chart.x_labels = xLabel
chart.render_in_browser()
```



Bar

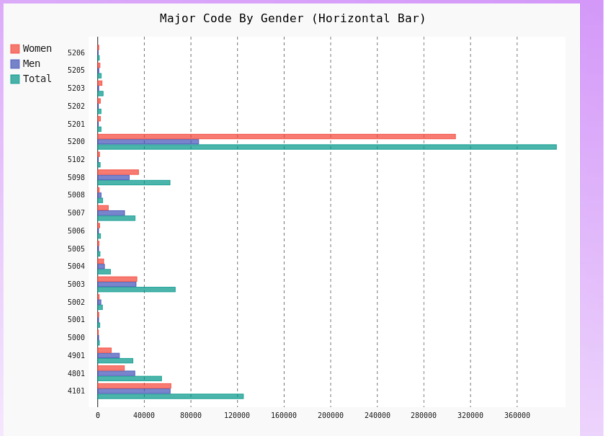
```
#!/usr/bin/python3
import pygal
import csv
data=readCsvAsDict('data.csv','Major_code')
Fields=['Women','Men','Total']
plotData=dict()
for key in Fields:
  D=[(k,v[key]) for (k,v) in sorted(data.items())[100:120]]
  L=([int(el[1]) if el[1].isdigit() else None for el in D])
  xLabel=([el[0] for el in D])
  plotData[key]=L
chart=pygal.Bar()
chart.title='Major Code By Gender (Bar)'
for key in Fields:
  chart.add(key,plotData[key])
chart.x_labels = xLabel
chart.render_in_browser()
```



FSK INC. CONSULTING

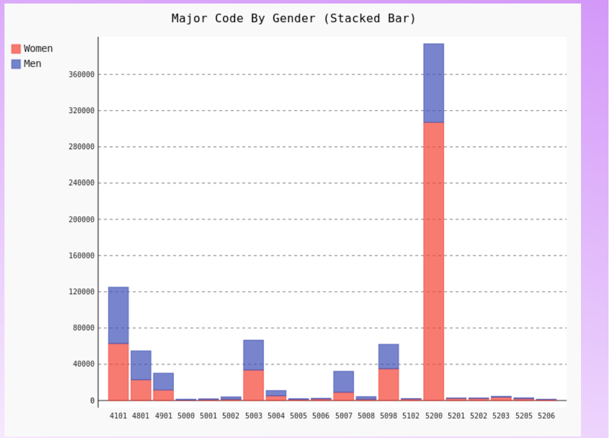
Horizontal Bar

```
#!/usr/bin/python3
import pygal
import csv
data=readCsvAsDict('data.csv','Major_code')
Fields=['Women','Men','Total']
plotData=dict()
for key in Fields:
  D=[(k,v[key]) for (k,v) in sorted(data.items())[100:120]]
  L=([int(el[1]) if el[1].isdigit() else None for el in D])
  xLabel=([el[0] for el in D])
  plotData[key]=L
chart=pygal.HorizontalBar()
chart.title='Major Code By Gender (Horizontal Bar)'
for key in Fields:
  chart.add(key,plotData[key])
chart.x_labels = xLabel
chart.render_in_browser()
```



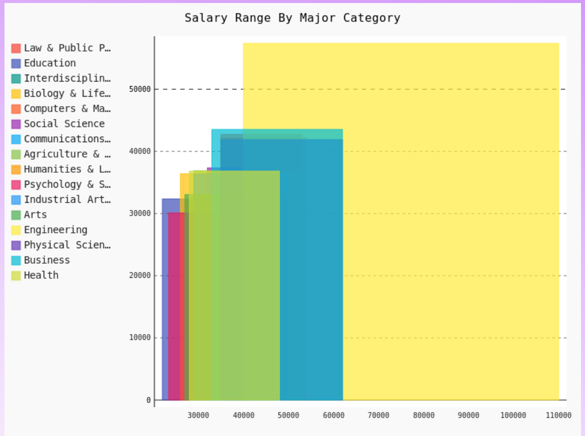
Stacked Bar

```
#!/usr/bin/python3
import pygal
import csv
data=readCsvAsDict('data.csv','Major_code')
Fields=['Women','Men']
plotData=dict()
for key in Fields:
  D=[(k,v[key]) for (k,v) in sorted(data.items())[100:120]]
  L=([int(el[1]) if el[1].isdigit() else None for el in D])
  xLabel=([el[0] for el in D])
  plotData[key]=L
chart=pygal.StackedBar()
chart.title='Major Code By Gender (Stacked Bar)'
for key in Fields:
  chart.add(key,plotData[key])
chart.x_labels = xLabel
chart.render_in_browser()
```



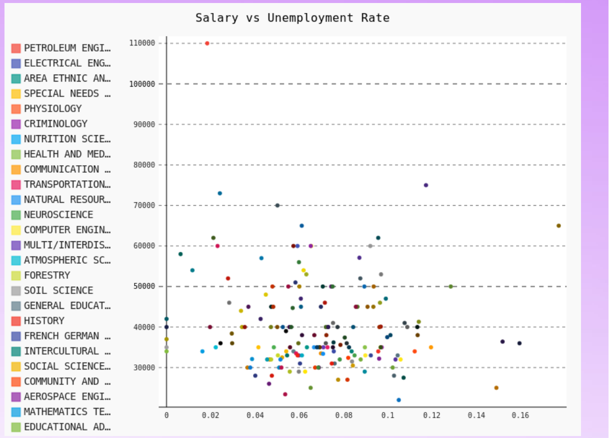
Histogram

```
#!/usr/bin/python3
import pygal
import csv
data=readCsvAsDict('data.csv','Major_code')
plotData=dict()
for (key,val) in data.items():
  try:
    plotData[val['Major_category']].append(int(val['Median']))
  except:
    plotData[val['Major_category']]=list()
    plotData[val['Major_category']].append(int(val['Median']))
categories=[val['Major_category'] for (k,val) in data.items()]
chart = pygal.Histogram()
chart.title='Salary Range By Major Category'
for k in set(categories):
  x0=min(plotData[k])
  x1=max(plotData[k])
  y=sum(plotData[k])/float(len(plotData[k]))
  chart.add(k,[(y,x0,x1)])
chart.render_in_browser()
```



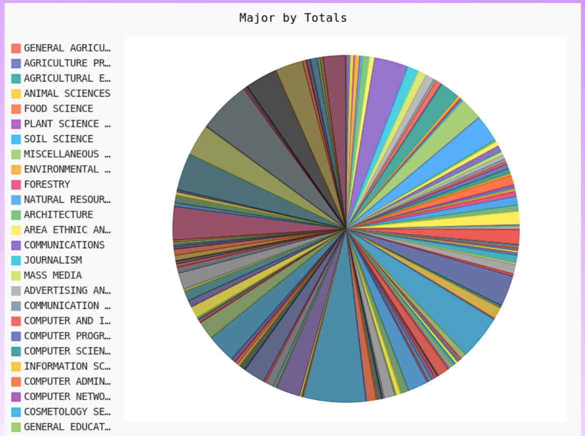
XY

```
#!/usr/bin/python3
import pygal
import csv
data=readCsvAsDict('data.csv','Rank')
chart = pygal.XY()
chart.title='Salary vs Unemployment Rate'
for (k,v) in sorted(data.items()):
  chart.add(v['Major'],[(float(v['Unemployment_rate']),int(v['Median']))])
chart.render_in_browser()
```



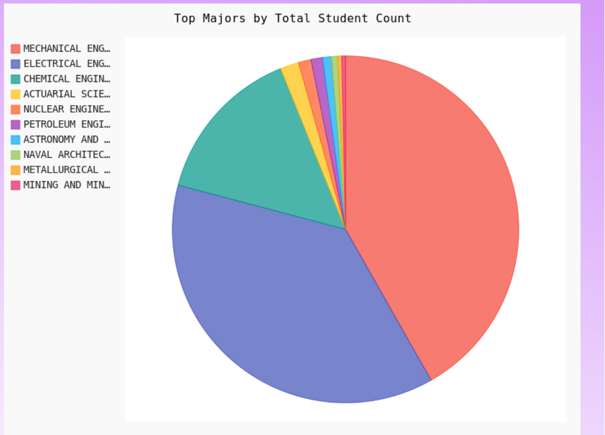
Pie

```
#!/usr/bin/python3
import pygal
import csv
data=readCsvAsDict('data.csv','Major_code')
chart = pygal.Pie()
chart.title='Major by Totals'
for (k,v) in sorted(data.items()):
  val = int(v['Total']) if v['Total'].isdigit() else None
  chart.add(v['Major'],val)
chart.render_in_browser()
```



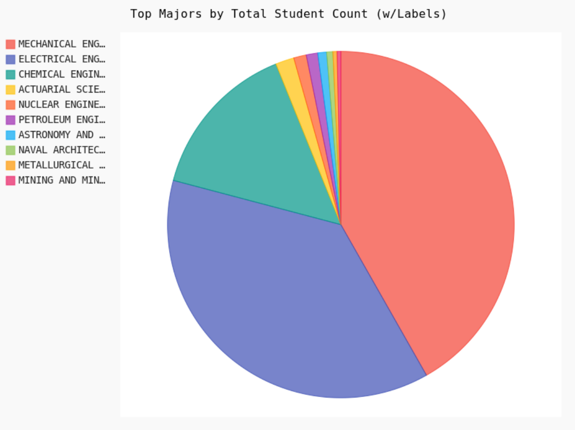
Pie

```
#!/usr/bin/python3
import pygal
import csv
data=readCsvAsDict('data.csv','Major_code')
chart = pygal.Pie()
chart.title='Top Majors by Total Student Count'
L=[(int(v['Total']),v['Major']) for (k,v) in data.items() if v['Total'].isdigit()]
L=L[0:10]
N=sum([v for (v,k) in L])
for (t,k) in sorted(L,reverse=True):
  chart.add(k,t)
chart.render_in_browser()
```



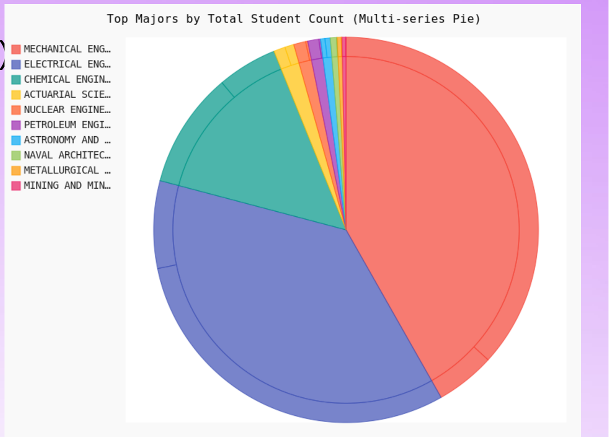
Pie w/Labels

```
#!/usr/bin/python3
import pygal
import csv
data=readCsvAsDict('data.csv','Major_code')
chart = pygal.Pie()
chart.title='Top Majors by Total Student Count (w/Labels)'
L=[(int(v['Total']),v['Major']) for (k,v) in data.items() if v['Total'].isdigit()]
L=L[0:10]
N=sum([v for (v,k) in L])
for (t,k) in sorted(L,reverse=True):
  chart.add(k,[{'value': t, 'label': "%0.2f%%"%(float(100*t)/N)}
chart.render_in_browser()
```



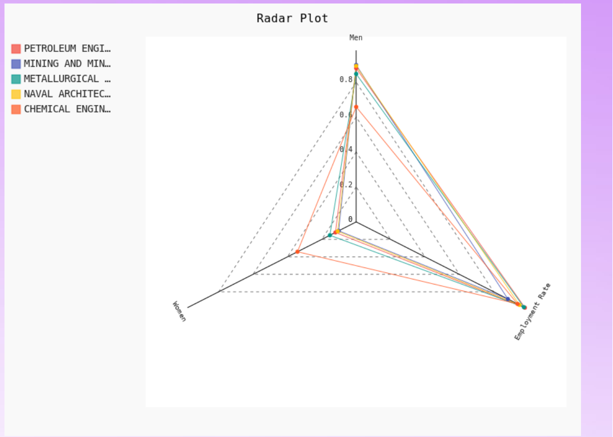
Multi-Series Pie

```
#!/usr/bin/python3

import pygal

import csv

data=readCsvAsDict('data.csv','Major_code')

chart = pygal.Pie()

chart.title='Top Majors by Total Student Count (Multi-series Pie)'

L=[(int(v['Total']),v['Major'],int(v['Men']),int(v['Women'])) for (k,v) in data.items() if
v['Total'].isdigit()]

L=L[0:10]

N=sum([v for (v,t,m,w) in L])

for (t,k,m,w) in sorted(L,reverse=True):

  chart.add(k,[{'value':m,'label':'men: %02f%%'%(float(100*m)/t)
%02f%%'%(float(100*w)/t)}])

chart.render_in_browser()
```



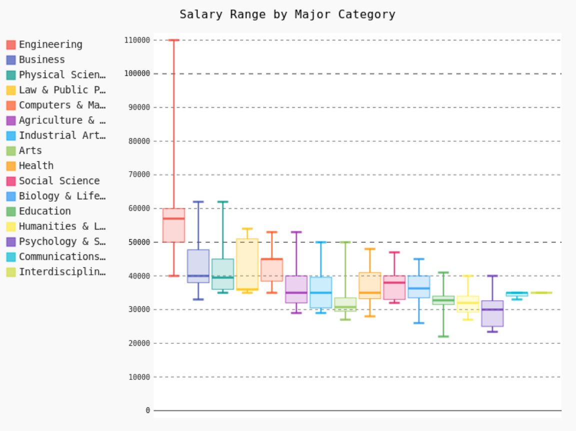
Radar

```
#!/usr/bin/python3
import pygal
import csv
data=readCsvAsDict('data.csv','Major_code')
chart = pygal.Radar()
chart.title = 'Radar Plot'
chart.x_labels=['Men','Women','Employment Rate']
for val in [v for (k,v) in data.items() if v['Total'].isdigit()][0:5]:
  L=[]
  L.append(float(val['Men'])/float(val['Total']));
  L.append(float(val['Women'])/float(val['Total']));
  L.append(1.0-float(val['Unemployment_rate']));
  chart.add(val['Major'],L)
chart.render_in_browser()
```



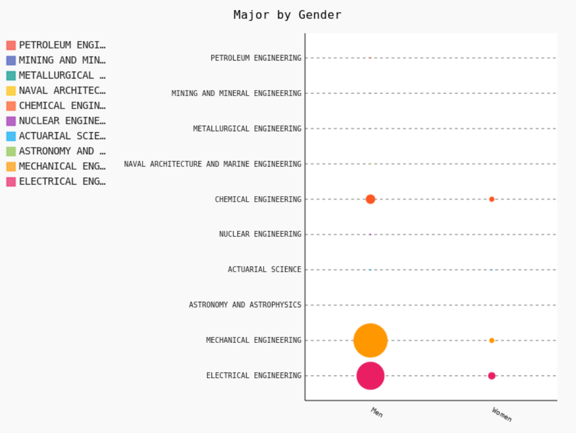
Box

```
#!/usr/bin/python3
import pygal
import csv
data=readCsvAsDict('data.csv','Major_code')
plotData=dict()
for val in [v for (k,v) in data.items() if v['Median'].isdigit()]:
 category=val['Major_category']
 try:
  plotData[category].append(int(val['Median']))
 except(KeyError):
  plotData[category]=list()
  plotData[category].append(int(val['Median']))
chart = pygal.Box()
chart.title = 'Salary Range by Major Category'
for (k,v) in plotData.items():
 chart.add(k,v)
chart.render_in_browser()
```



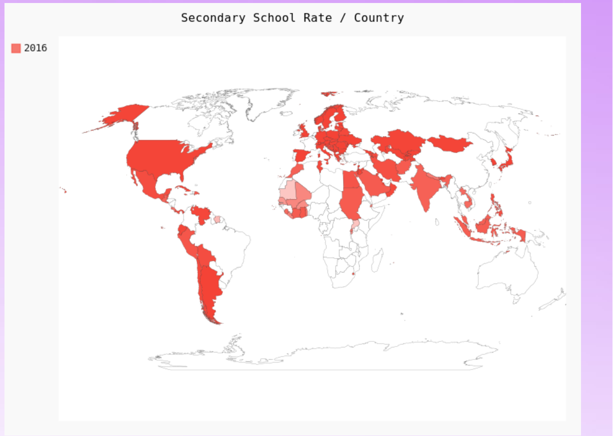
Dot

```
#!/usr/bin/python3
import pygal
import csv
data=readCsvAsDict('data.csv','Major_code')
plotData=dict()
for val in [v for (k,v) in data.items() if v['Total'].isdigit()][0:10]:
  category=val['Major']
  try:
    plotData[category].append(int(val['Men']))
    plotData[category].append(int(val['Women']))
  except(KeyError):
    plotData[category]=list()
    plotData[category].append(int(val['Men']))
    plotData[category].append(int(val['Women']))
chart = pygal.Dot(x_label_rotation=30)
chart.title = 'Major by Gender'
chart.x_labels = ['Men', 'Women']
for (k,v) in plotData.items():
  chart.add(k, v)
chart.render_in_browser()
```



World Map

```
#!/usr/bin/python3
import pygal
import csv
def convertCountryCodeToPygal(countryCode):
  convertCountryCodeToPygal.data=readCsvAsDict('WDICountry.csv','Country Code')
  return convertCountryCodeToPygal.data[countryCode]['2-alpha code'].lower()

data=readCsvAsDict('school.csv','Country Code')
chart = pygal.maps.world.World()
chart.title = 'Secondary School Rate / Country'
year=2016
plotData=dict()
for (k,v) in data.items():
  try:
    plotData[convertCountryCodeToPygal(k)]=float(v[str(year)])
  except:
    pass
chart.add(str(year),plotData)
chart.render_to_png('./example10.png')
chart.render_in_browser()
```



References

- http://www.pygal.org/
  - Official Site
- https://github.com/fivethirtyeight/data/tree/master/college-majors/
- https://datacatalog.worldbank.org/dataset/world-development-indicators/

Contact Info

- Slides:
  - https://github.com/fsk-software/pub/
- Blog: http://dragonquest64.blogspot.com
- Slack: pymntos.slack.com lipeltgm