

```
[50] import warnings
warnings.filterwarnings('ignore')
#!pip install rpy2
#!pip install pandas
#!pip install tzlocal
#!pip install ggplot
#!pip install leather
#!pip install rpy2==2.8.6
```

```
[59] ## Lets try out some code
import pandas as pd          # do all processing in pandas, and
                              # convert Pandas DataFrame to R DataFrame

#imports required from rpy2
from rpy2.robjects import pandas2ri

pandas2ri.activate()
%reload_ext rpy2.ipython
```

```

----
ModuleNotFoundError                                Traceback (most recent call
last)
<ipython-input-59-2b44dbb04a12> in <module>
      3
      4 #imports required from rpy2
----> 5 from rpy2.objects import pandas2ri
      6
      7 pandas2ri.activate()

ModuleNotFoundError: No module named 'rpy2'

```

```
[1]      ## HTML changes to align the images to the center of the screen

      from IPython.core.display import HTML
      HTML("""
      """)
```

```
[2] df = pd.DataFrame({'Letter': ['a', 'a', 'a', 'b', 'b', 'b', 'c',  
                                'c', 'c'],  
                    'X': [4, 3, 5, 2, 1, 7, 7, 5, 9],  
                    'Y': [0, 4, 3, 6, 7, 10, 11, 9, 13],  
                    'Z': [1, 2, 3, 1, 2, 3, 1, 2, 3]})
```

```

-----
---
NameError                                Traceback (most recent call
last)
<ipython-input-2-5ee69aa1aded> in <module>
----> 1 df = pd.DataFrame({'Letter': ['a', 'a', 'a', 'b','b', 'b', 'c',
    'c','c'],
    2                                'X': [4, 3, 5, 2, 1, 7, 7, 5, 9],
    3                                'Y': [0, 4, 3, 6, 7, 10, 11, 9, 13],
    4                                'Z': [1, 2, 3, 1, 2, 3, 1, 2, 3]})

NameError: name 'pd' is not defined

```

```
[3] df
```

```

-----
---
NameError                                Traceback (most recent call
last)
<ipython-input-3-00cf07b74dcd> in <module>
----> 1 df

NameError: name 'df' is not defined

```

```
[58] %%R -i df -w 900 -h 480 -u px

## Everything in here is ** R ** (magic)

print(df) # Tada, df is now an R dataframe

library("ggplot2") # If this line does not work, make sure you
have R installed on your laptop

ggplot(data = df) + geom_point(aes(x = X, y = Y, color = Letter,
size = Z))

```

UsageError: Cell magic `%%R` not found.

```
[ ] import pandas as pd
# midwest= pd.read_csv('http://goo.gl/G1K41K')
midwest= pd.read_csv('midwest.csv')
# Filtering
midwest= midwest[midwest.poptotal<50000]

```

```
[ ] midwest.head().loc[:, ['area']]
```

```
[ ] from ggplot import *

ggplot(aes(x='area', y='poptotal', color='state',
size='popdensity'), data=midwest) +\
    geom_point() +\
    theme_bw() +\
    xlab("Area") +\
    ylab("Population") +\
    ggtitle("Area vs Population")
```

```
[ ] import seaborn as sns
import matplotlib.pyplot as plt
sizes = [10, 40, 70, 100]
marker_size = pd.cut(midwest['popdensity'], range(0, 2500, 500),
labels=sizes)
sns.lmplot('area', 'poptotal', data=midwest, hue='state',
fit_reg=False, scatter_kws={'s':marker_size})
plt.ylim((0, 50000))
```

```
[ ] import random

import leather

dot_data = [(random.randint(0, 250), random.randint(0, 250)) for
i in range(100)]

def colorizer(d):
    return 'rgb(%i, %i, %i)' % (d.x, d.y, 150)

chart = leather.Chart('Colorized dots')
chart.add_dots(dot_data, fill_color=colorizer)
chart.to_svg('colorized_dots.svg')
```

```
[ ] import leather
import pandas as pd

df=pd.read_csv("train.csv")

pclass_one=df[df['Pclass']==1]['Pclass'].count().astype(float)
pclass_two=df[df['Pclass']==2]['Pclass'].count().astype(float)
pclass_three=df[df['Pclass']==3]['Pclass'].count().astype(float)

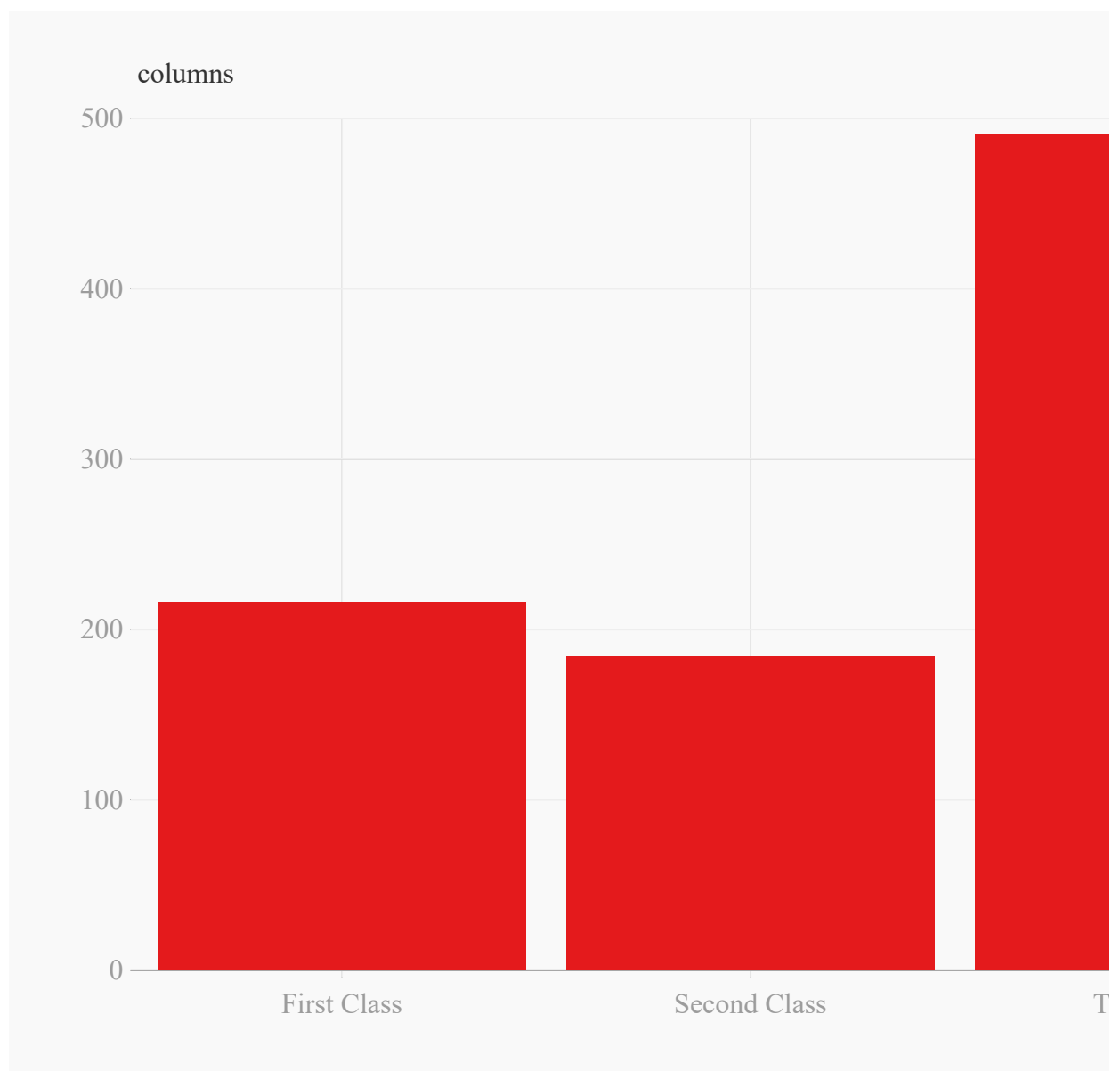
data = [
    ('First Class', pclass_one),
    ('Second Class',pclass_two),
```

```
    ('Third Class', pclass_three)
]

chart = leather.Chart('columns')
chart.add_columns(data)
chart.to_svg('leather_distribution_by_class.svg')
```

```
[37] from IPython.display import SVG
def show_svg():
    return SVG('leather_distribution_by_class.svg')
```

```
[38] show_svg()
```



```
[ ] from bokeh.io import show, output_file
from bokeh.plotting import figure
```

```

import pandas as pd

df=pd.read_csv("train.csv")

survived=df[df['Survived']==1]['Survived'].count().astype(float)
not_survived=df[df['Survived']==0]
['Survived'].count().astype(float)

survival_values= [survived,not_survived]
output_file("bokeh_survival_bar_chart.html")
survival_labels = ['Survived','Not Survived']

bkh = figure(x_range=survival_labels, plot_height=300,
title="Survival Distribution", toolbar_location=None, tools="")
bkh.vbar(x=survival_labels, top=survival_values, width=0.9)

bkh.xgrid.grid_line_color = None
bkh.y_range.start = 0
show(bkh)

```

```

[ ] import pandas as pd
import random
import matplotlib.pyplot as plt
import seaborn as sns

df = pd.DataFrame()
df=pd.read_csv("train.csv")

g = sns.FacetGrid(df, row='Survived', col='Pclass')
g.map(sns.distplot, "Age")
sns.plt.show()

```

```

[1] import pandas as pd
import seaborn as sns

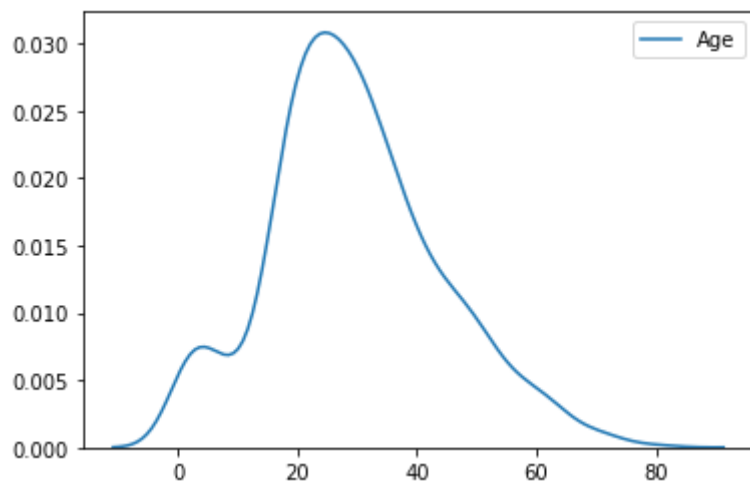
df = pd.DataFrame()
df=pd.read_csv("train.csv")

df['Age'] = df['Age']

sns.kdeplot(df.Age)

```

<matplotlib.axes.\_subplots.AxesSubplot at 0x22aeb9205f8>



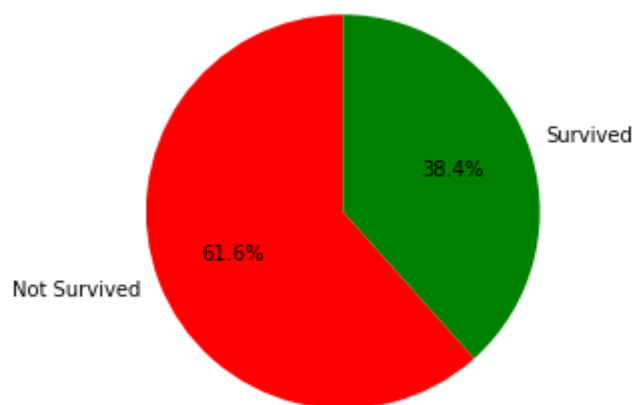
```
[2] import matplotlib.pyplot as plt
import pandas as pd

df=pd.read_csv("train.csv")

pie_chart=df.groupby(df['Survived'])['Survived'].count()
labels=['Not Survived','Survived']
color = ['r','g']

plt.pie(pie_chart, labels=labels, colors=color, startangle=90,
shadow= False, autopct='%1.1f%%')

plt.axis('equal')
plt.show()
```



```
[3] import matplotlib.pyplot as plt
import pandas as pd

df=pd.read_csv("train.csv")

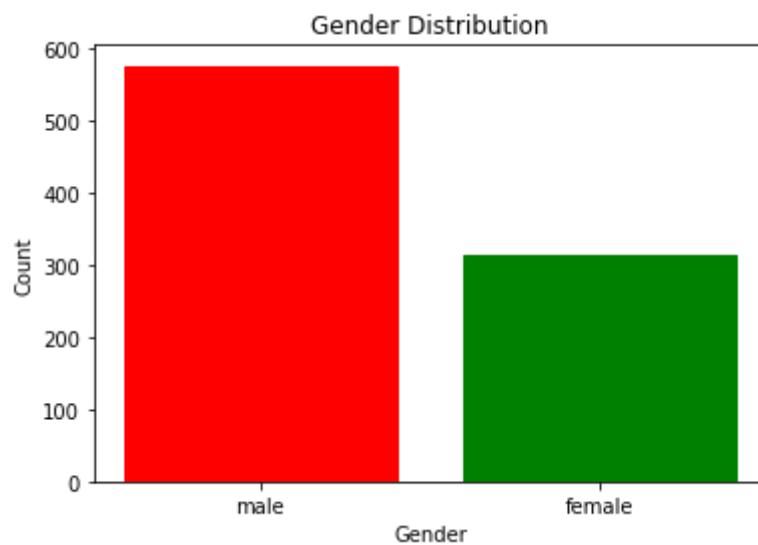
male=df[df['Sex']=='male']['Sex'].count().astype(float)
female=df[df['Sex']=='female']['Sex'].count().astype(float)
```

```
x = [2,3]
y = [male,female]

bar=plt.bar(x, y, align='center')
bar[0].set_color('r')
bar[1].set_color('g')

plt.title('Gender Distribution')
plt.ylabel('Count')
plt.xlabel('Gender')
plt.xticks(x , ['male','female'])

plt.show()
```



[ ]

[ ]