Data Processing and Analysis in Python
Lecture 15
Visualization and matplotlib

UNIVERSITY OF MARYLAND
ROBERT H. SMITH SCHOOL OF BUSINESS

DR. ADAM LEE

Matplotlib

- Matrix Plot Library
- A plotting library for the Python programming language and its numerical mathematics extension NumPy
- An open source, BSD-licensed library (module)
- SciPy makes use of Matplotlib
- Pyplot is a Matplotlib module which provides a MATLAB-like interface:
  - Designed to be as usable as MATLAB
  - Advantage of being free and open-source
- Several toolkits are available to extend functionality

Use Pyplot in Matplotlib

```
# Import the whole matplotlib
>>> from matplotlib import *
# Import the whole pyplot module
>>> import matplotlib.pyplot
# Import the whole pyplot module using alias
>>> import matplotlib.pyplot as plt
```

https://matplotlib.org/

```
>>> dir(matplotlib)
>>> dir(matplotlib.pyplot)
>>> help(matplotlib)
>>> help(matplotlib.pyplot)
```

UNIVERSITY OF MARYLAND

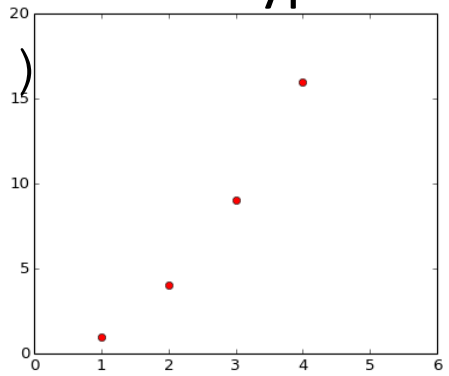
Plot

- plot() is a versatile function, and will take an arbitrary number of arguments
  - For example, to plot x versus y, you can issue the command:

```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4])
plt.plot([1,2,3,4],[1,4,9,16],'b-')
```

  - For every x, y pair of arguments, there is an optional third argument to format string that indicates the color and line type

```
plt.plot([1,2,3,4],[1,4,9,16],'ro')
plt.axis([0,6,0,20])
plt.show()
```



Format Strings:

```
fmt='[marker][line][color]'
```

	Marker
'.'	point marker
'o'	circle marker
'v'	triangle_down marker
'^'	triangle_up marker
'<'	triangle_left marker
'>'	triangle_right marker
's'	square marker
'p'	pentagon marker
'h'	hexagon marker
'*'	star marker
'+'	plus marker
'x'	x marker
'd'	diamond marker

	Line Style
'-'	solid line style
'--'	dashed line style
'-.'	dash-dot line style
':'	dotted line style
	Color
'b'	blue
'g'	green
'r'	red
'c'	cyan
'm'	magenta
'y'	yellow
'k'	black
'w'	white

UNIVERSITY OF MARYLAND

Matplotlib Figure Functions

- **annotate(text, xy, ...)** Annotate the point xy with text
- **colorbar(mappable, ...)** Add a colorbar to a plot
- **figlegend(...)** Place a legend on the figure
- **figure(num, ...)** Create a new figure or activate an existing
- **grid(which, axis, ...)** Configure the grid lines
- **plot_date(x, y, xdate, ydate, ...)** Plot data that contains dates
- **savefig(fname, dpi, format, ...)** Save the current figure
- **show(...)** Display all open figures
- **subplot(nrows, ncols, index, ...)** Add subplot to current figure
- **subplots(nrows, ncols, ...)** Create a figure and a set of subplots
- **suptitle(t, x, y, ...)** Add a centered title to the figure
- **text(x, y, s, ...)** Add text to the axes
- **title(label, loc, y, ...)** Set a title for the axes

Matplotlib Axis Functions

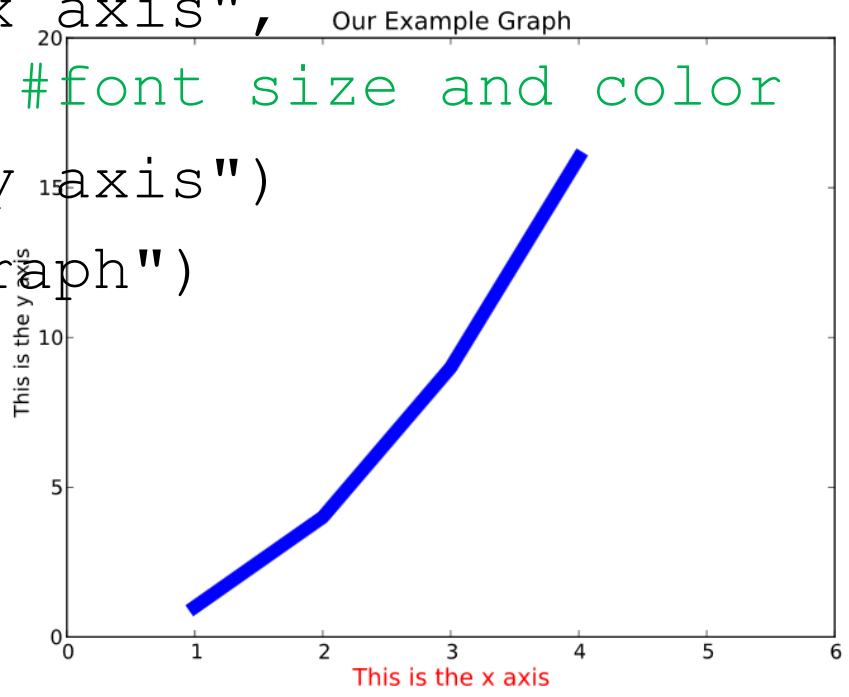
- **axis(xmin, xmax, ymin, ymax)** To get or set some axis properties
- **twinx(ax)** Make and return a second axes that shares x-axis
- **twiny(ax)** Make and return a second axes that shares y-axis
- **xlabel(xlabel, loc, ...)** Set the label for x-axis
- **xlim(left, right)** Get or set x limits of the current axes
- **xscale(value, ...)** Set x-axis scale
- **xticks(ticks, labels)** Get or set the current tick locations of x-axis
- **ylabel(ylabel, loc, ...)** Set the label for the y-axis
- **ylim(bottom, top)** Get or set y-limits of the current axes
- **yscale(value, ...)** Set the y-axis scale
- **yticks(ticks, labels)** Get or set the current tick locations of y-axis

Matplotlib Chart Functions

- **bar(x, height[, width, bottom, align, data])** Make a bar plot
- **barh(y, width[, height, left, align])** Make a horizontal bar plot
- **boxplot(x[, notch, sym, vert, whis, ...])** Make a box and whisker plot
- **broken_barh(...)** Plot a horizontal sequence of rectangles
- **fill(data, ...)** Plot filled polygons
- **hist(x[, bins, range, density, weights, ...])** Plot a histogram
- **hist2d(x, y[, bins, range, density, ...])** Make a 2D histogram plot
- **loglog(...)** Make a plot with log scaling on both axes
- **scatter(x, y[, s, c, marker, ...])** A scatter plot of y versus x
- **semilogx(...)** Make a plot with log scaling on the x axis
- **semilogy(...)** Make a plot with log scaling on the y axis
- **stackplot(x[, labels, colors, ...])** Draw a stacked area plot

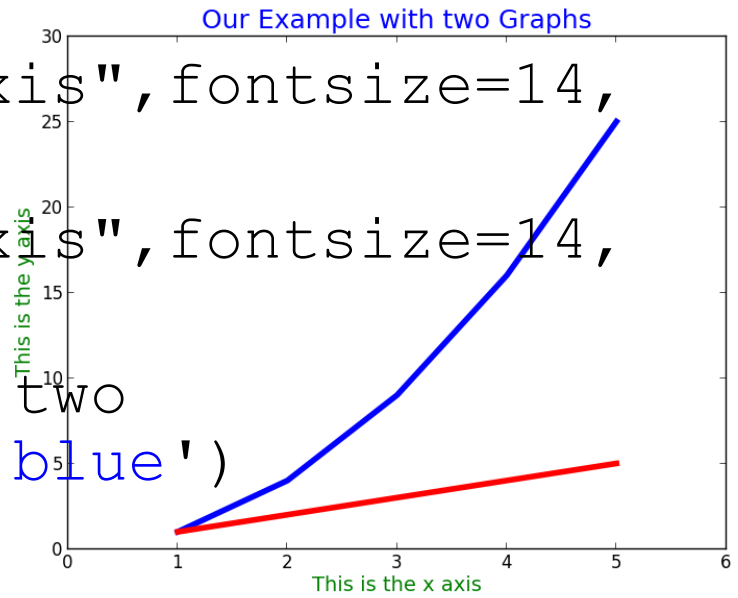
Line and Text Properties

```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4],[1,4,9,16],linewidth=8)
#thick line
plt.axis([0,6,0,20]) #sets x and y axis ranges
plt.xlabel("This is the x axis",
fontsize=14,color='red') #font size and color
plt.ylabel("This is the y axis")
plt.title("Our Example Graph")
plt.show()
```



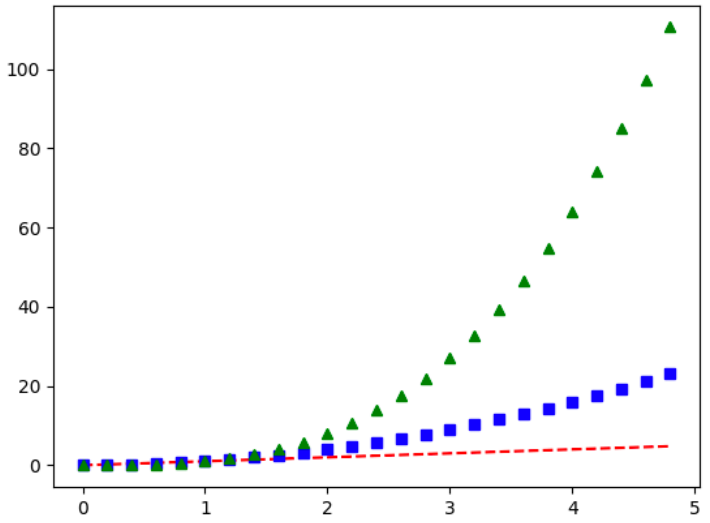
Two Lines

```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4,5],[1,4,9,16,25],linewidth=4,
color='blue') #line 1
plt.plot([1,2,3,4,5],[1,2,3,4,5],linewidth=4,
color='red') #line 2
plt.axis([0,6,0,30])
plt.xlabel("This is the x axis",fontsize=14,
color='green')
plt.ylabel("This is the y axis",fontsize=14,
color='green')
plt.title("Our Example with two
Graphs",fontsize=18, color='blue')
plt.show()
```



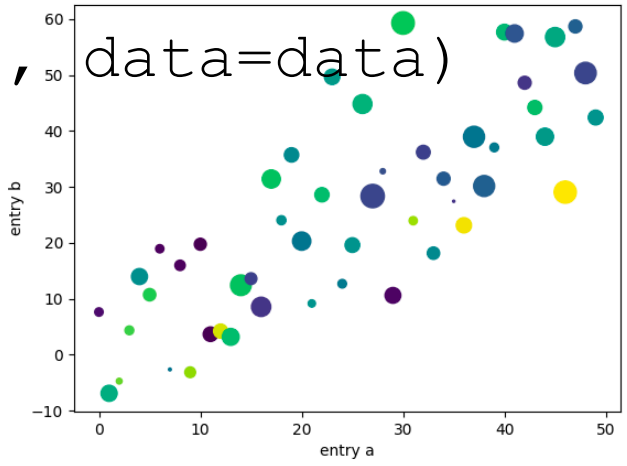
Different Styles

```
import numpy as np
import matplotlib.pyplot as plt
# evenly sampled time at 200ms intervals
t = np.arange(0.,5.,0.2)
# red dashes, blue squares and green triangles
plt.plot(t,t,'r--',t,t**2,'bs',t,t**3,'g^')
plt.show()
```



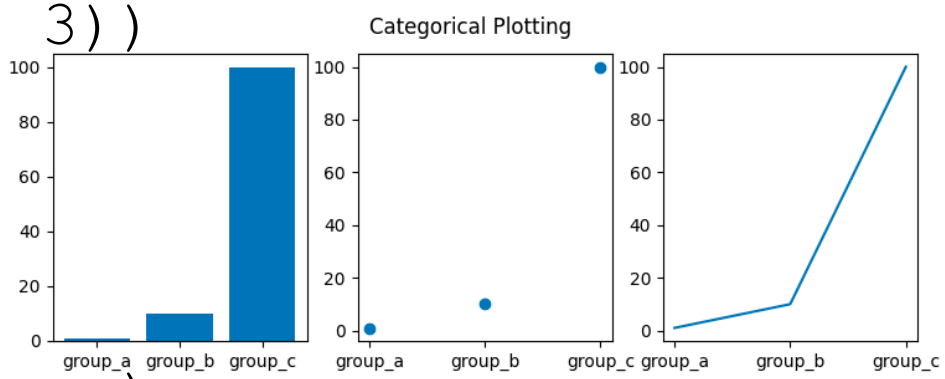
Scatter Plot

```
import numpy as np
import matplotlib.pyplot as plt
data = {'a':np.arange(50)}
data['b'] = data['a']+10*np.random.randn(50)
data['c'] = np.random.randint(0,50,50)
data['d'] = np.abs(np.random.randn(50))*100
# c=color, s=scalar
plt.scatter('a','b',c='c',s='d',data=data)
plt.xlabel("entry a")
plt.ylabel("entry b")
plt.show()
```



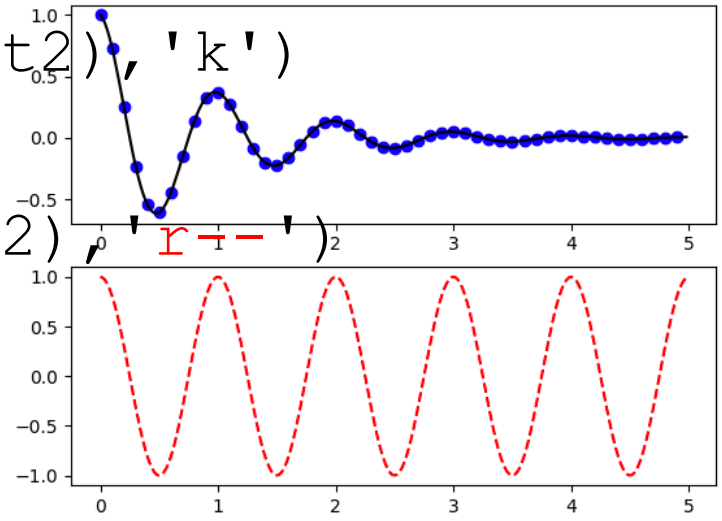
Categorical Plots

```
names = ["group_a", "group_b", "group_c"]
values = [1, 10, 100]
plt.figure(figsize=(9, 3))
plt.subplot(131)
plt.bar(names, values)
plt.subplot(132)
plt.scatter(names, values)
plt.subplot(133)
plt.plot(names, values)
plt.suptitle("Categorical Plotting")
plt.show()
```



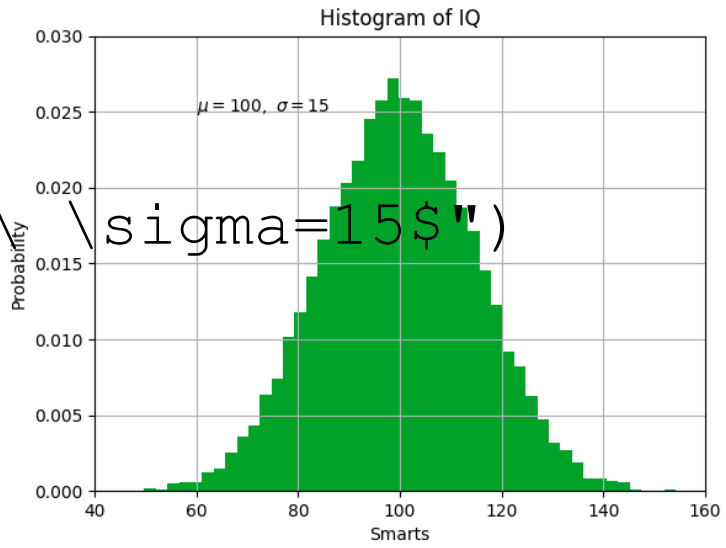
Multiple Plots

```
def f(t):
    return np.exp(-t) * np.cos(2*np.pi*t)
t1 = np.arange(0.0,5.0,0.1)
t2 = np.arange(0.0,5.0,0.02)
plt.figure()
plt.subplot(211)
plt.plot(t1,f(t1),'bo',t2,f(t2),'k')
plt.subplot(212)
plt.plot(t2,np.cos(2*np.pi*t2),'r--')
plt.show()
```



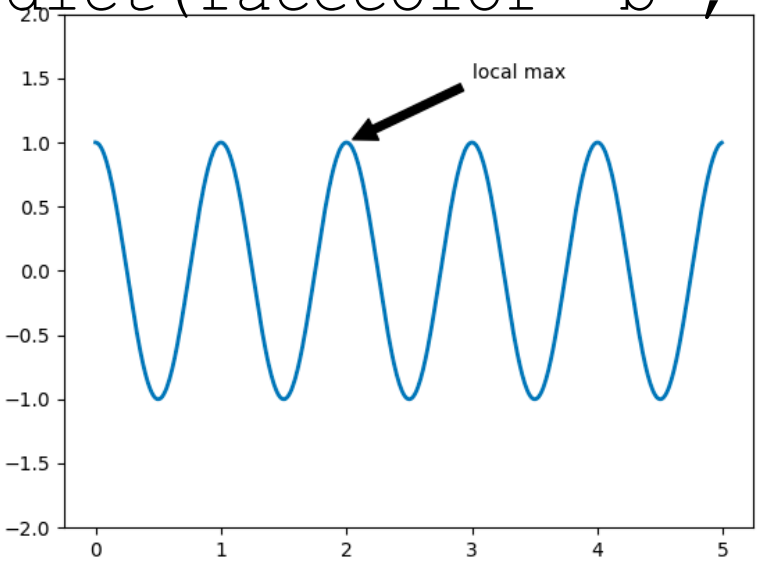
Histogram

```
mu, sigma = 100, 15
x = mu + sigma * np.random.randn(10000)
# the histogram of the data
n, bins, patches = plt.hist(x, 50, density=1,
facecolor='g', alpha=0.75)
plt.xlabel("Smarts")
plt.ylabel("Probability")
plt.title("Histogram of IQ")
plt.text(60,.025,r"$\mu=100,\ \sigma=15$")
plt.axis([40,160,0,0.03])
plt.grid(True)
plt.show()
```



Annotating Text

```
ax = plt.subplot(111)
t = np.arange(0.0, 5.0, 0.01)
s = np.cos(2 * np.pi * t)
line, = plt.plot(t, s, lw=2)
plt.annotate("local max", xy=(2,1),
xytext=(3,1.5), arrowprops=dict(facecolor='b',
shrink=0.05), )
plt.ylim(-2,2)
plt.show()
```



NBA Player Data Analysis

UNIVERSITY OF MARYLAND

Step 1: Import Data from Text File

```
import pandas as pd
filepath = "NBA_Player_Data.csv"
df = pd.read_csv(filepath)
print(df.head())
```

```
  First Name   Last Name    Salary Position       Team  Division
0       Kobe      Bryant  27849149       SG  LA Lakers   Pacific
1    Carmelo     Anthony  20463024       SF  NY Knicks  Atlantic
2     Amar'e  Stoudemire  19948799       PF  NY Knicks  Atlantic
3     Dwight      Howard  19536360        C  LA Lakers   Pacific
4        Pau       Gasol  19000000       PF  LA Lakers   Pacific
```

Step 2: Report Contents on a Column

```
print(df["First Name"])
print(df["Last Name"])
print(df["Salary"])
print(df["Team"][0])
print(df.head(5)["Division"])
```

```
0        Kobe
1      Carmelo
2       Amar'e
3      Dwight
4         Pau
5       Chris
6      LeBron
7      Dwyane
Name: First Name, dtype: object
0      Bryant
1     Anthony
2    Stoudemire
3      Howard
4       Gasol
5        Bosh
6       James
7        Wade
Name: Last Name, dtype: object
0    27849149
1    20463024
2    19948799
3    19536360
4    16000000
Name: Salary, dtype: int64
LA Lakers
0     Pacific
1    Atlantic
2    Atlantic
3     Pacific
4     Pacific
Name: Division, dtype: object
```

Step 3: Search Players (i.e. Rows) by Name

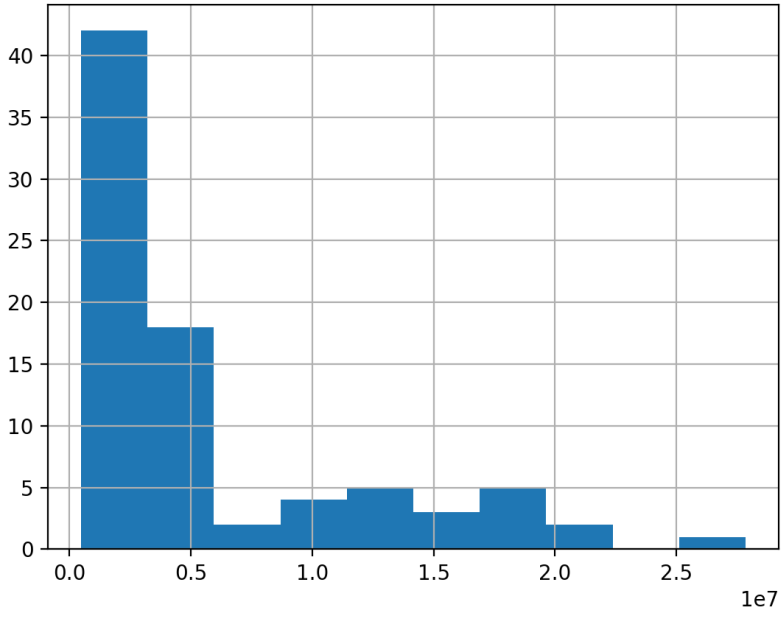
```
print(df.loc[df["First Name"]=="James"])
print(df.loc[df["Last Name"]=="Anthony"])
```

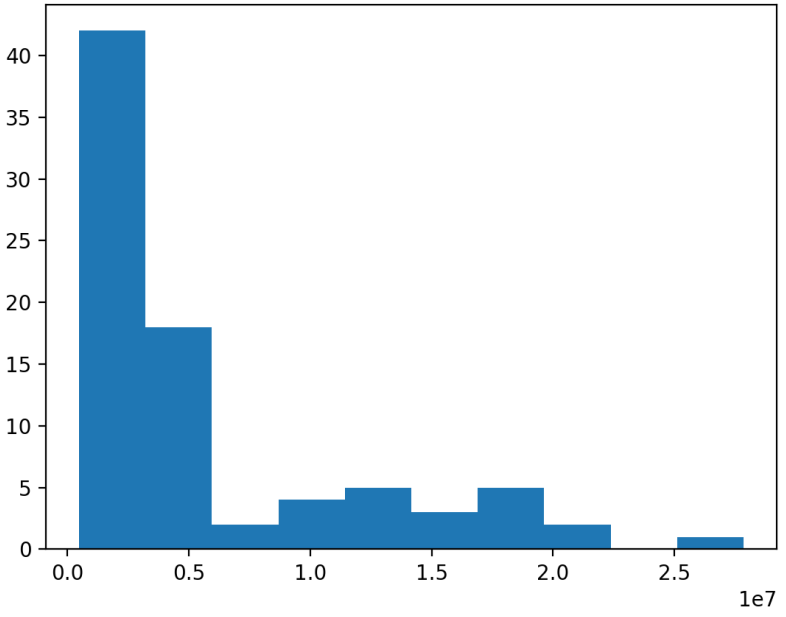
	First Name	Last Name	Salary	Position	Team	Division
22	James	Harden	5820417	SG	Oklahoma Thunder	Northwest
54	James	Jones	1500000	SF	Miami Heat	Southeast
78	James	White	854389	SF	NY Knicks	Atlantic

	First Name	Last Name	Salary	Position	Team	Division
1	Carmelo	Anthony	20463024	SF	NY Knicks	Atlantic
33	Joel	Anthony	3750000	C	Miami Heat	Southeast

Step 4: Report Salaries in Histogram

```
import matplotlib.pyplot as plt
df["Salary"].hist()
plt.show()
plt.hist(df["Salary"])
plt.show()
```





Step 5: Which Position Earns More in a Team? Center or Shooting Guard?

```
C_player = df.loc[df["Position"]=="C"]
print(C_player)
SG_player = df.loc[df["Position"]=="SG"]
print(SG_player)
```

	First Name	Last Name	Salary	Position	Team	Division
0	Kobe	Bryant	27849149	SG	LA Lakers	Pacific
6	LeBron	James	17545000	SG	Miami Heat	Southeast
11	Manu	Ginobili	14107492	SG	San Antonio Spurs	Southeast
22	James	Harden	5820417	SG	Oklahoma Thunder	Northwest
24	Richard	Hamilton	5000000	SG	Chicago Bulls	Central
35	Thabo	Sefolosha	3600000	SG	Oklahoma Thunder	Northwest

	First Name	Last Name	Salary	Position	Team	Division
3	Dwight	Howard	19536360	C	LA Lakers	Pacific
5	Chris	Bosh	17545000	C	Miami Heat	Southeast
10	Carlos	Boozer	15000000	C	Chicago Bulls	Central
13	Tyson	Chandler	13604188	C	NY Knicks	Atlantic
16	Joakim	Noah	11300000	C	Chicago Bulls	Central
18	Tim	Duncan	9638554	C	San Antonio Spurs	Southeast

Step 6: Rename Columns

```
C_player.columns=["First","Last","Salary_C",
 "Pos","Team","Div"]
print(C_player)
SG_player.columns=["First","Last","Salary_SG",
 "Pos","Team","Div"]
print(SG_player)
```

	First	Last	Salary_C	Pos	Team	Div
3	Dwight	Howard	19536360	C	LA Lakers	Pacific
5	Chris	Bosh	17545000	C	Miami Heat	Southeast
10	Carlos	Boozer	15000000	C	Chicago Bulls	Central
13	Tyson	Chandler	13604188	C	NY Knicks	Atlantic

	First	Last	Salary_SG	Pos	Team	Div
0	Kobe	Bryant	27849149	SG	LA Lakers	Pacific
6	LeBron	James	17545000	SG	Miami Heat	Southeast
11	Manu	Ginobili	14107492	SG	San Antonio Spurs	Southeast
22	James	Harden	5820417	SG	Oklahoma Thunder	Northwest

Step 7: Renumber Rows

```
C_player = C_player.reset_index()
print(C_player)
SG_player = SG_player.reset_index()
print(SG_player)
```

	index	First	Last	Salary_C	Pos	Team	Div
0	3	Dwight	Howard	19536360	C	LA Lakers	Pacific
1	5	Chris	Bosh	17545000	C	Miami Heat	Southeast
2	10	Carlos	Boozer	15000000	C	Chicago Bulls	Central
3	13	Tyson	Chandler	13604188	C	NY Knicks	Atlantic
4	16	Joakim	Noah	11300000	C	Chicago Bulls	Central

	index	First	Last	Salary_SG	Pos	Team	Div
0	0	Kobe	Bryant	27849149	SG	LA Lakers	Pacific
1	6	LeBron	James	17545000	SG	Miami Heat	Southeast
2	11	Manu	Ginobili	14107492	SG	San Antonio Spurs	Southeast
3	22	James	Harden	5820417	SG	Oklahoma Thunder	Northwest
4	24	Richard	Hamilton	5000000	SG	Chicago Bulls	Central

Step 8: Partition Top-15 Salaries

```
salary = pd.concat(
  [C_player["Salary_C"][0:15],
   SG_player["Salary_SG"][0:15]],
   axis=1)
print(salary)
```

	Salary_C	Salary_SG
0	19536360	27849149
1	17545000	17545000
2	15000000	14107492
3	13604188	5820417
4	11300000	5000000
5	9638554	3600000
6	8300531	3500000
7	4590338	3090000
8	3944000	2806452
9	3750000	1957000
10	2445480	1633440
11	2253062	1500000
12	1352181	1208400
13	1200000	1074720
14	1054000	1066920

Step 9: Descriptive Statistics

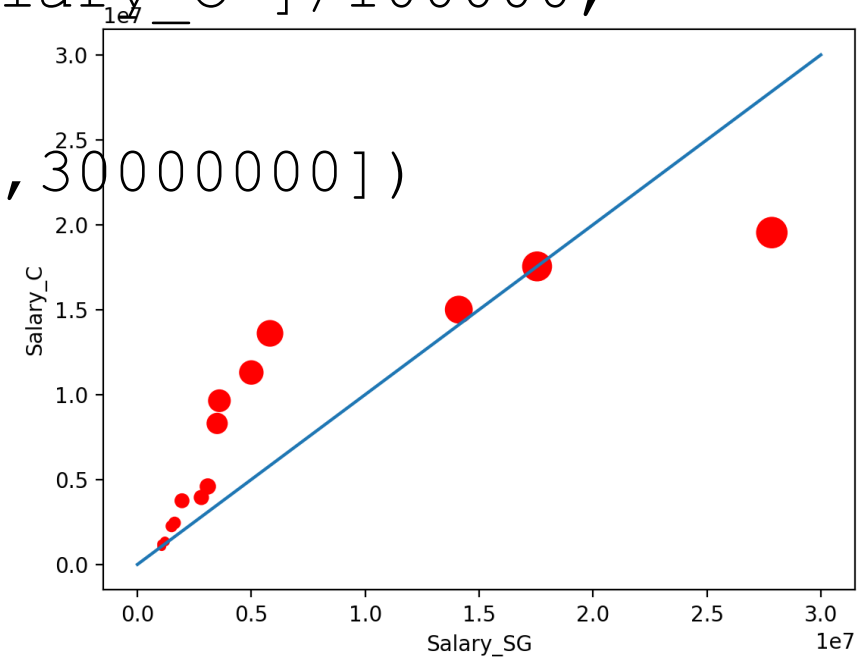
```
print(df['Salary'].max())
print(df['Salary'].min())
print(df['Salary'].mean())
print(df['Salary'].describe())
```

```
27849149
473604
5692881.609756097
count    8.200000e+01
mean     5.692882e+06
std      6.309205e+06
min      4.736040e+05
25%      1.180560e+06
50%      3.112971e+06
75%      8.040138e+06
max      2.784915e+07
Name: Salary, dtype: float64
```

UNIVERSITY OF MARYLAND

Step 10: Which Position Earns More in a Team? Center or Shooting Guard?

```
import matplotlib.pyplot as plt
salary.plot(kind="scatter",
            x="Salary_SG",
            y="Salary_C",
            s=salary["Salary_C"]/100000,
            color='r')
plt.plot([0,30000000],[0,30000000])
plt.show()
```



NBA Team Data Analysis

UNIVERSITY OF MARYLAND

Step 1: Change Display Options

```
import pandas as pd
# default: 0
pd.set_option("display.max_columns", 5)
# default: 60 rows
pd.set_option("display.max_rows", 5)
# default: 80 characters
pd.set_option("display.width", 75)
```

UNIVERSITY OF MARYLAND

Step 2: Import Data from Text and Excel Files

```
dataPlayer = pd.read_csv("NBA_Player_Data.csv")
print(dataPlayer)
dataTeam =
pd.read_excel("NBA_Team_Salary.xlsx",
sheet_name="Summary")
print(dataTeam)
```

```
   First Name Last Name  ...       Team  Division
0        Kobe    Bryant  ...  LA Lakers   Pacific
1     Carmelo   Anthony  ...  NY Knicks  Atlantic
..        ...       ...  ...        ...       ...
80      Chris  Copeland  ...  NY Knicks  Atlantic
81      Pablo  Prigioni  ...  NY Knicks  Atlantic

[82 rows x 6 columns]
    Rk                    Team  ...  2012-13  2013-14
0    1             Miami Heat  ...      NaN      NaN
1    2  Golden State Warriors  ...      NaN      NaN
..  ..                    ...  ...      ...      ...
28  29       Sacramento Kings  ...      NaN      NaN
29  30       Dallas Mavericks  ...      NaN      NaN

[30 rows x 8 columns]
```

Step 3: Combine Player and Team Data

```
pd.set_option("display.max_columns", 0)
pd.set_option("display.width", 0)
dataCombine =
dataPlayer.join(dataTeam.set_index("Team"),
on="Team")
print(dataCombine)
```

```
   First Name Last Name    Salary Position       Team  Division  Rk  2008-09  20
09-10  2010-11  2011-12  2012-13  2013-14
0        Kobe    Bryant  27849149       SG  LA Lakers   Pacific NaN      NaN
  NaN      NaN      NaN      NaN      NaN
1     Carmelo   Anthony  20463024       SF  NY Knicks  Atlantic NaN      NaN
  NaN      NaN      NaN      NaN      NaN
..        ...      ...      ...      ...        ...       ...  ..      ...
  ...      ...      ...      ...      ...
80      Chris  Copeland    473604       SF  NY Knicks  Atlantic NaN      NaN
  NaN      NaN      NaN      NaN      NaN
81      Pablo  Prigioni    473604       PG  NY Knicks  Atlantic NaN      NaN
  NaN      NaN      NaN      NaN      NaN

[82 rows x 13 columns]
```

Step 4: Analyze Unique Team Names

```
print(dataPlayer["Team"].unique())
print(dataTeam["Team"].unique())
```

```
['LA Lakers' 'NY Knicks' 'Miami Heat' 'Oklahoma Thunder' 'Chicago Bulls'
 'San Antonio Spurs']
['Miami Heat' 'Golden State Warriors' 'Oklahoma City Thunder'
 'Toronto Raptors' 'Portland Trail Blazers' 'Milwaukee Bucks'
 'Boston Celtics' 'Detroit Pistons' 'Memphis Grizzlies'
 'Cleveland Cavaliers' 'Washington Wizards' 'Houston Rockets'
 'New York Knicks' 'Charlotte Hornets' 'Minnesota Timberwolves'
 'San Antonio Spurs' 'Los Angeles Clippers' 'Brooklyn Nets'
 'Denver Nuggets' 'New Orleans Pelicans' 'Orlando Magic' 'Utah Jazz'
 'Philadelphia 76ers' 'Chicago Bulls' 'Phoenix Suns' 'Indiana Pacers'
 'Atlanta Hawks' 'Los Angeles Lakers' 'Sacramento Kings'
 'Dallas Mavericks']
```

UNIVERSITY OF MARYLAND

Step 5: Match Full Team Names

```
dataPlayer.loc[dataPlayer["Team"]=="LA Lakers",
"Team"] = "Los Angeles Lakers"
dataPlayer.loc[dataPlayer["Team"]=="NY Knicks",
"Team"] = "New York Knicks"
dataPlayer.loc[dataPlayer["Team"]=="Oklahoma
Thunder", "Team"] = "Oklahoma City Thunder"
print(dataPlayer["Team"].unique())
```

```
['Los Angeles Lakers' 'New York Knicks' 'Miami Heat'
 'Oklahoma City Thunder' 'Chicago Bulls' 'San Antonio Spurs']
```

Step 6: Combine Player and Team Data

```
dataCombine =
dataPlayer.join(dataTeam.set_index("Team"),
on="Team")
print(dataCombine)
```

```
  First Name Last Name    Salary Position                Team  Division  Rk
 2008-09     2009-10     2010-11     2011-12  2012-13  2013-14
0       Kobe    Bryant  27849149       SG  Los Angeles Lakers   Pacific  28  10
7194758.0  66047804.0  67577154.0  48154193.0      NaN      NaN
1     Carmelo   Anthony  20463024       SF     New York Knicks  Atlantic  13  12
2961124.0  36089890.0  24547462.0  14079701.0      NaN      NaN
..        ...       ...       ...       ...                 ...       ...  ..
     ...         ...         ...         ...      ...      ...
80      Chris  Copeland    473604       SF     New York Knicks  Atlantic  13  12
2961124.0  36089890.0  24547462.0  14079701.0      NaN      NaN
81      Pablo  Prigioni    473604       PG     New York Knicks  Atlantic  13  12
2961124.0  36089890.0  24547462.0  14079701.0      NaN      NaN

[82 rows x 13 columns]
```

Step 7: Calculate Percentage

```
pd.set_option("display.max_rows", 0)
dataAnalysis = dataCombine[["First Name",
"Team", "Salary", "2008-09", "Last Name"]]
dataAnalysis.loc[:, "First Name"] =
dataAnalysis["First Name"] + ' ' +
dataAnalysis["Last Name"]
dataAnalysis.loc[:, "Last Name"] =
dataAnalysis["Salary"] /
dataAnalysis["2008-09"]
dataAnalysis.columns=["Name", "Team", "Salary",
"Budget", "Percentage"]
print(dataAnalysis)
```

UNIVERSITY OF MARYLAND

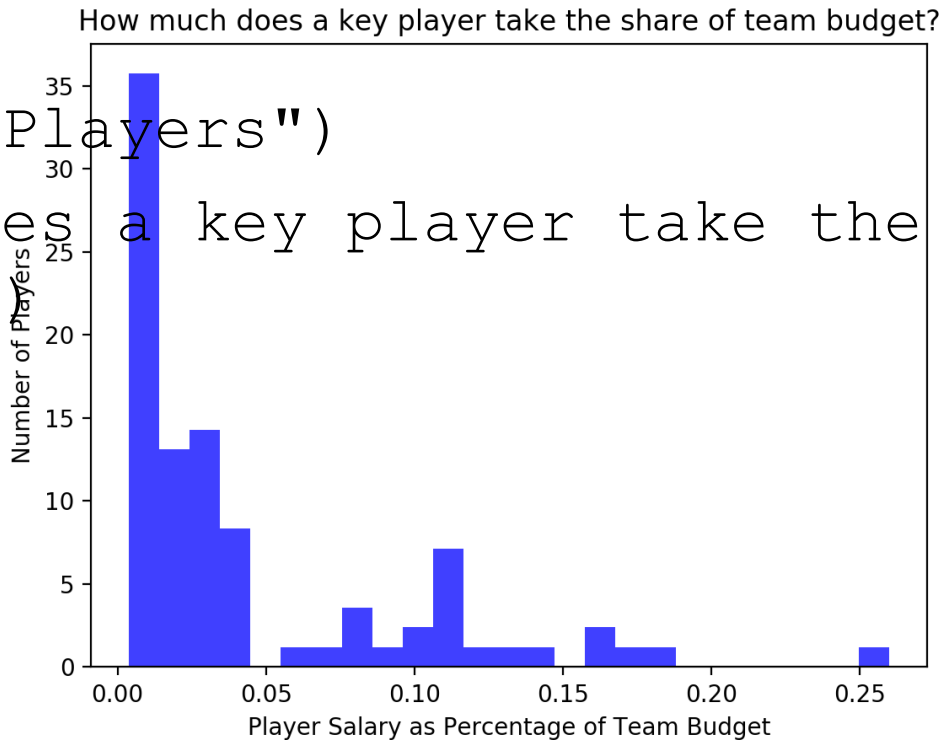
Step 7: Calculate Percentage

	Name	Team	Salary	Budget	Percentage
0	Kobe Bryant	Los Angeles Lakers	27849149	107194758.0	0.259800
1	Carmelo Anthony	New York Knicks	20463024	122961124.0	0.166419
2	Amar'e Stoudemire	New York Knicks	19948799	122961124.0	0.162237
3	Dwight Howard	Los Angeles Lakers	19536360	107194758.0	0.182251
4	Pau Gasol	Los Angeles Lakers	19000000	107194758.0	0.177247
5	Chris Bosh	Miami Heat	17545000	151687967.0	0.115665
6	LeBron James	Miami Heat	17545000	151687967.0	0.115665
7	Dwyane Wade	Miami Heat	17182000	151687967.0	0.113272
8	Kevin Durant	Oklahoma City Thunder	16669630	145625023.0	0.114470
..	...	...	...	...	...
73	Antawn Jamison	Los Angeles Lakers	854389	107194758.0	0.007970
74	Rodney Carney	Miami Heat	854389	151687967.0	0.005633
75	Dexter Pittman	Miami Heat	854389	151687967.0	0.005633
76	Garrett Temple	Miami Heat	854389	151687967.0	0.005633
77	Ronnie Brewer	New York Knicks	854389	122961124.0	0.006948
78	James White	New York Knicks	854389	122961124.0	0.006948
79	Josh Harrellson	Miami Heat	762195	151687967.0	0.005025
80	Chris Copeland	New York Knicks	473604	122961124.0	0.003852
81	Pablo Prigioni	New York Knicks	473604	122961124.0	0.003852

[82 rows x 5 columns]

Step 8: How much does a key player take the share of team budget?

```
import matplotlib.pyplot as plt
plt.hist(dataAnalysis["Percentage"], 25,
density=1, facecolor='b', alpha=0.75)
plt.xlabel("Player Salary as Percentage of Team
Budget")
plt.ylabel("Number of Players")
plt.title("How much does a key player take the
share of team budget?")
plt.show()
```



Step 9: Treemap

```
import squarify
squarify.plot(sizes=dataAnalysis["Percentage"],
label=dataAnalysis["Name"], alpha=.8)
plt.axis("off")
plt.show()
```

Step 10: Hierarchical Treemap

```
import plotly.express as px
fig = px.treemap(dataAnalysis, path=["Team", "Name"], values="Percentage")
fig.show()
```