homework 3

In [1]:

```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

Numpy

Cosine Similarity

Calculate the cosine similarity of 2 vectors (as numpy arrays).

Give vector $\vec{\mu}$ and vector $\vec{v}$, the cosine similarity of them is

$$S_{\vec{\mu}\vec{v}} = \frac{\vec{\mu} \cdot \vec{v}}{||\vec{\mu}||_2 \times ||\vec{v}||_2}$$

Where $|| \cdot ||_2$ is the L-2 norm.

Question: Define a CosSim function, with inputs are two one-D arrays, and the output is their cosine similarity.

In [2]:

```
# your code here
def CosSim(u, v):
    return u.dot(v) / (np.sqrt(u.dot(u) * v.dot(v)))
```

In [3]:

```
# validate your results

x = np.linspace(-1,1,100)
y = np.linspace(1,-1,100)

print(
    """The cosine similarity between x and x is {:.5f}
The cosine similarity between x and y is {:.5f}
    """.format(CosSim(x,x), CosSim(x,y))
     )
```

```
The cosine similarity between x and x is 1.00000
The cosine similarity between x and y is -1.00000
```

Linear Algebra

Solve the following Linear equation system:

$$\begin{cases} 4x + 3y + 2z = 25 \\ -2x + 2y + 3z = -10 \\ 3x - 5y + 2z = -4 \end{cases}$$

Question: Define a 2-d array A and a 1-d array b according to the above equation, which satisfy $AX = b$, where $X = (x, y, z)^T$

In [4]:

```
# your code here
A = np.array([[4,3,2],[-2,2,3],[3,-5,2]])
b = np.array([25,-10,-4])

print (A)
print (b)
```

```
[[ 4  3  2]
 [-2  2  3]
 [ 3 -5  2]]
[ 25 -10  -4]
```

Question: Solve the equation, to get the values of X.

Hint: you may use `numpy.linalg.inv()` to get the inverse of a matrix.

In [5]:

```
print(np.matmul(np.linalg.inv(A),b))
```

```
[ 5.  3. -2.]
```

Baby Names Data Analysis

Load the `baby_name_NY.txt` dataset, which contains the baby names in New York city.

In [6]:

```
# load the data
column_names = ['State', 'Sex', 'Year', 'Name', 'Count']
babynames = pd.read_csv("./baby_name_NY.TXT", header=None, names=column_names)

babynames.head(5)
```

Out[6]:

	State	Sex	Year	Name	Count
0	NY	F	1910	Mary	1923
1	NY	F	1910	Helen	1290
2	NY	F	1910	Rose	990
3	NY	F	1910	Anna	951
4	NY	F	1910	Margaret	926

Question: What's the data types of each column?

In [7]:

```
babynames.dtypes
```

Out[7]:

```
State    object
Sex      object
Year      int64
Name     object
Count     int64
dtype: object
```

Question: Find the most popular baby name in NY in 2018

In [8]:

```
print(babynames.loc[(babynames.Year == 2018) & (babynames.State == "NY")].sort_values("Count",ascen
```

```
Liam
```

Question: For female and male, what's the most popular baby names in 2017 respectively?

In [9]:

```
print("Female:",babynames.loc[(babynames.Sex == "F")].loc[(babynames.Year == 2017)].sort_values("Co
print("Male:",babynames.loc[(babynames.Sex == "M")].loc[(babynames.Year == 2017)].sort_values("Coun
```

```
Female: Olivia
Male: Liam
```

Question: List all baby names that start with J.

In [10]:

```
for i in babynames.loc[((babynames.Name < "K") & (babynames.Name > "J"))]["Name"].unique():
    print(i)
```

```
Josephine
Jean
Julia
Jennie
Jane
Joan
Jeanette
Jessie
Jeanne
Jeannette
Janet
Johanna
June
Janice
Judith
Joyce
John
Jacqueline
Jenny
```

Question: Sort names by their length, then print the top 5 by length.

In [11]:

```
# babynames.sort_values("Name",ascending = False,key = lambda x,y: len(x) < len(y) ? True : (len(x)

for i in sorted(babynames.Name.unique(), key = lambda x: -len(x))[:5]:
    print(i)
```

```
Michaelanthony
Maryelizabeth
Marycatherine
Samanthamarie
Oluwadarasimi
```

Question: Name whose popularity has changed the most.

Hint: First you may need to define change in popularity, i.e., for each name, you need to find the difference between the name's maximum occurrence and minimum occurrence.

In [12]:

```
num = len(babynames)
name_map = {}

for i in range(num):
    record = babynames.iloc[i]
    name = record["Name"]
    ct = record["Count"]

    if (name in name_map):
        val = name_map[name]
        val["min"] = min(val["min"], ct)
        val["max"] = max(val["max"], ct)
    else:
        name_map[name] = {"min":ct, "max":ct}

max_change = 0
max_name = None
for key, val in name_map.items():
    change = val["max"] - val["min"]
    if (change > max_change):
        max_change = change
        max_name = key

print("Name:", max_name, ", Change:", max_change)
```

```
Name: Robert , Change: 10020
```

Single variable analysis

The `Salaries.csv` dataset contains the salaries of employees in San Francisco, see details here (https://transparentcalifornia.com/salaries/san-francisco/).

Question: Read the data using pandas, save it as a Dataframe called `Salaries_df`

In [13]:

```
Salaries_df = pd.read_csv("Salaries.csv", index_col = "Id")

Salaries_df
```

Out[13]:

Id	EmployeeName	TotalPayBenefits	Year
1	NATHANIEL FORD	567595.43	2011
2	GARY JIMENEZ	538909.28	2011
3	ALBERT PARDINI	335279.91	2011
4	CHRISTOPHER CHONG	332343.61	2011
5	PATRICK GARDNER	326373.19	2011
6	DAVID SULLIVAN	316285.74	2011
7	ALSON LEE	315981.05	2011
8	DAVID KUSHNER	307899.46	2011
9	MICHAEL MORRIS	303427.55	2011
10	JOANNE HAYES-WHITE	302377.73	2011
11	ARTHUR KENNEY	299494.17	2011
12	PATRICIA JACKSON	297608.92	2011
13	EDWARD HARRINGTON	294580.02	2011
14	JOHN MARTIN	292671.62	2011
15	DAVID FRANKLIN	286347.05	2011
16	RICHARD CORRIEA	286213.86	2011
17	AMY HART	284720.43	2011
18	SEBASTIAN WONG	278569.21	2011
19	MARTY ROSS	276434.22	2011
20	ELLEN MOFFATT	274550.25	2011
21	VENUS AZAR	274190.27	2011
22	JUDY MELINEK	273771.21	2011
23	GEORGE GARCIA	273702.71	2011
24	VICTOR WYRSCH	270672.63	2011
25	JOSEPH DRISCOLL	270324.91	2011
26	GREGORY SUHR	267992.59	2011
27	JOHN HANLEY	265784.56	2011
28	RAYMOND GUZMAN	265463.46	2011
29	DENISE SCHMITT	264074.60	2011
30	MONICA FIELDS	261366.14	2011

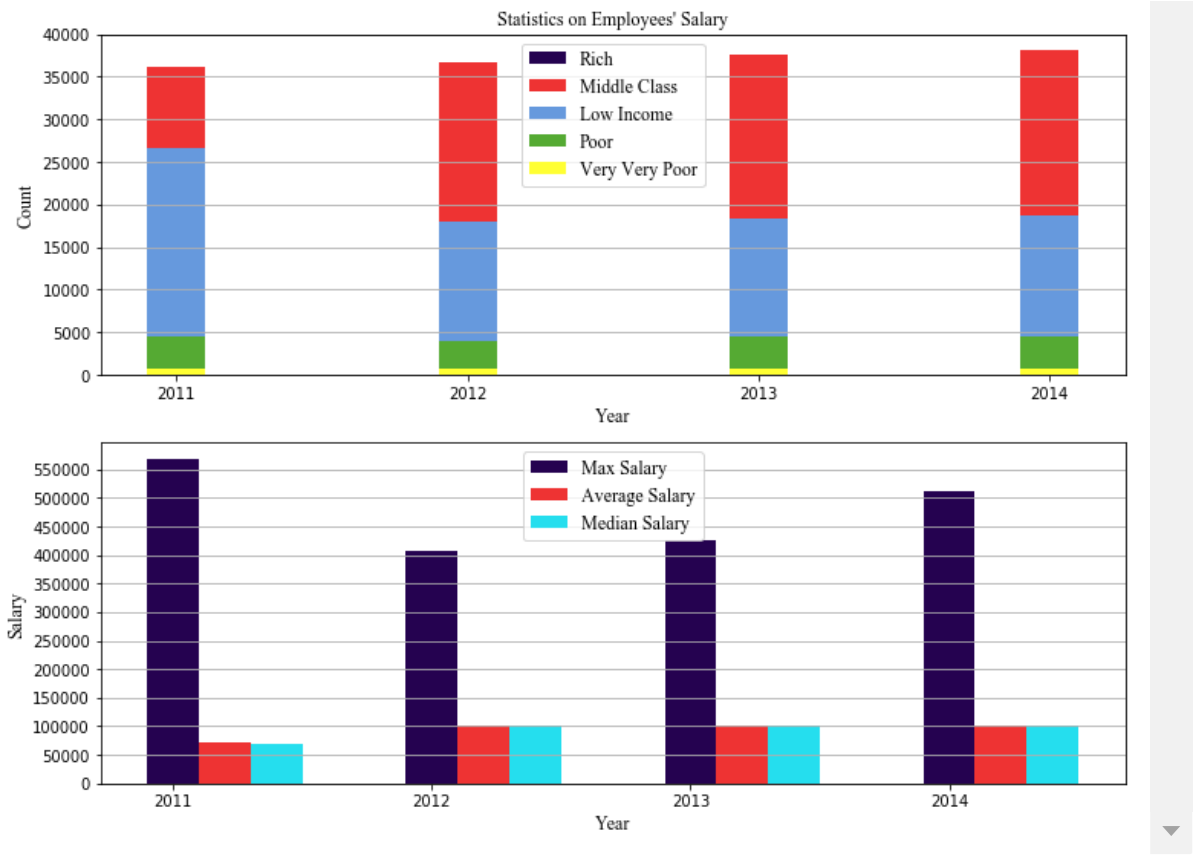
Id	EmployeeName	TotalPayBenefits	Year
...	...	...	...
148625	Lorraine Rosenthal	12.89	2014
148626	Renato C Gurion	7.24	2014
148627	Paulet Gaines	0.00	2014
148628	Brett A Lundberg	0.00	2014
148629	Mark W Mcclure	0.00	2014
148630	Elizabeth Iniguez	0.00	2014
148631	Randy J Keys	0.00	2014
148632	Andre M Johnson	0.00	2014
148633	Sharon D Owens-Webster	0.00	2014
148634	Edward Ferdinand	0.00	2014
148635	David M Turner	0.00	2014
148636	James S Kibblewhite	0.00	2014
148637	Andrew J Enzi	0.00	2014
148638	Kadeshra D Green	0.00	2014
148639	Lennard B Hutchinson	0.00	2014
148640	Richard A Talbert	0.00	2014
148641	Charlene D Mccully	0.00	2014
148642	Raphael Marquis Goins	0.00	2014
148643	Dominic C Marquez	0.00	2014
148644	Kim Brewer	0.00	2014
148645	Randy D Winn	0.00	2014
148646	Carolyn A Wilson	0.00	2014
148647	Not provided	0.00	2014
148648	Joann Anderson	0.00	2014
148649	Leon Walker	0.00	2014
148650	Roy I Tillery	0.00	2014
148651	Not provided	0.00	2014
148652	Not provided	0.00	2014
148653	Not provided	0.00	2014
148654	Joe Lopez	-618.13	2014

148654 rows × 3 columns

Question: Make a bar plot to show the number of DataRecords by year.

In [68]:

```
font = {"family":"Times New Roman","weight":"bold","size":12}

x = Salaries_df.sort_values("Year")["Year"].unique()
y1 = []
y11 = []
y12 = []
y13 = []
y14 = []
y2 = []
y3 = []
y4 = []
for i in x:
    y1.append(Salaries_df.loc[Salaries_df.Year == i]["Year"].count())
    y11.append(Salaries_df.loc[(Salaries_df.Year == i) & (Salaries_df.TotalPayBenefits < 865)]["Yea
    y12.append(Salaries_df.loc[(Salaries_df.Year == i) & (Salaries_df.TotalPayBenefits < 10000)]["Y
    y13.append(Salaries_df.loc[(Salaries_df.Year == i) & (Salaries_df.TotalPayBenefits < 100000)]["
    y14.append(Salaries_df.loc[(Salaries_df.Year == i) & (Salaries_df.TotalPayBenefits < 500000)]["
    y2.append(Salaries_df.loc[Salaries_df.Year == i]["TotalPayBenefits"].max())
    y3.append(Salaries_df.loc[Salaries_df.Year == i]["TotalPayBenefits"].mean())
    y4.append(Salaries_df.loc[Salaries_df.Year == i]["TotalPayBenefits"].median())

fig = plt.figure(figsize=(12, 9))

ax1 = fig.add_subplot(2, 1, 1)
ax2 = fig.add_subplot(2, 1, 2)

ax1.grid(b = True,axis = "y")
ax1.set_xlabel('Year',font)
ax1.set_ylabel('Count',font)
ax1.bar(x, y1, width=0.2, color='#250250', label='Rich')
ax1.bar(x, y14, width=0.2, color='#EE3333', label='Middle Class')
ax1.bar(x, y13, width=0.2, color='#6699DD', label='Low Income')
ax1.bar(x, y12, width=0.2, color='#55AA33', label='Poor')
ax1.bar(x, y11, width=0.2, color='#FFFF33', label='Very Very Poor')
ax1.legend(loc='upper center',prop = font)
ax1.xaxis.set_major_locator(plt.MultipleLocator(1))
ax1.set_ylim(0,40000)
ax1.yaxis.set_major_locator(plt.MultipleLocator(5000))
ax1.set_title("Statistics on Employees' Salary",font)

ax2.grid(b = True,axis = "y")
ax2.set_xlabel('Year',font)
ax2.set_ylabel('Salary',font)
ax2.bar(x, y2, width=0.2, color='#250250', label='Max Salary')
ax2.bar(x+0.2, y3, width=0.2, color='#EE3333', label='Average Salary')
ax2.bar(x+0.4, y4, width=0.2, color='#25DEEE', label='Median Salary')
ax2.legend(loc='upper center',prop = font)
ax2.xaxis.set_major_locator(plt.MultipleLocator(1))
ax2.yaxis.set_major_locator(plt.MultipleLocator(50000))
```



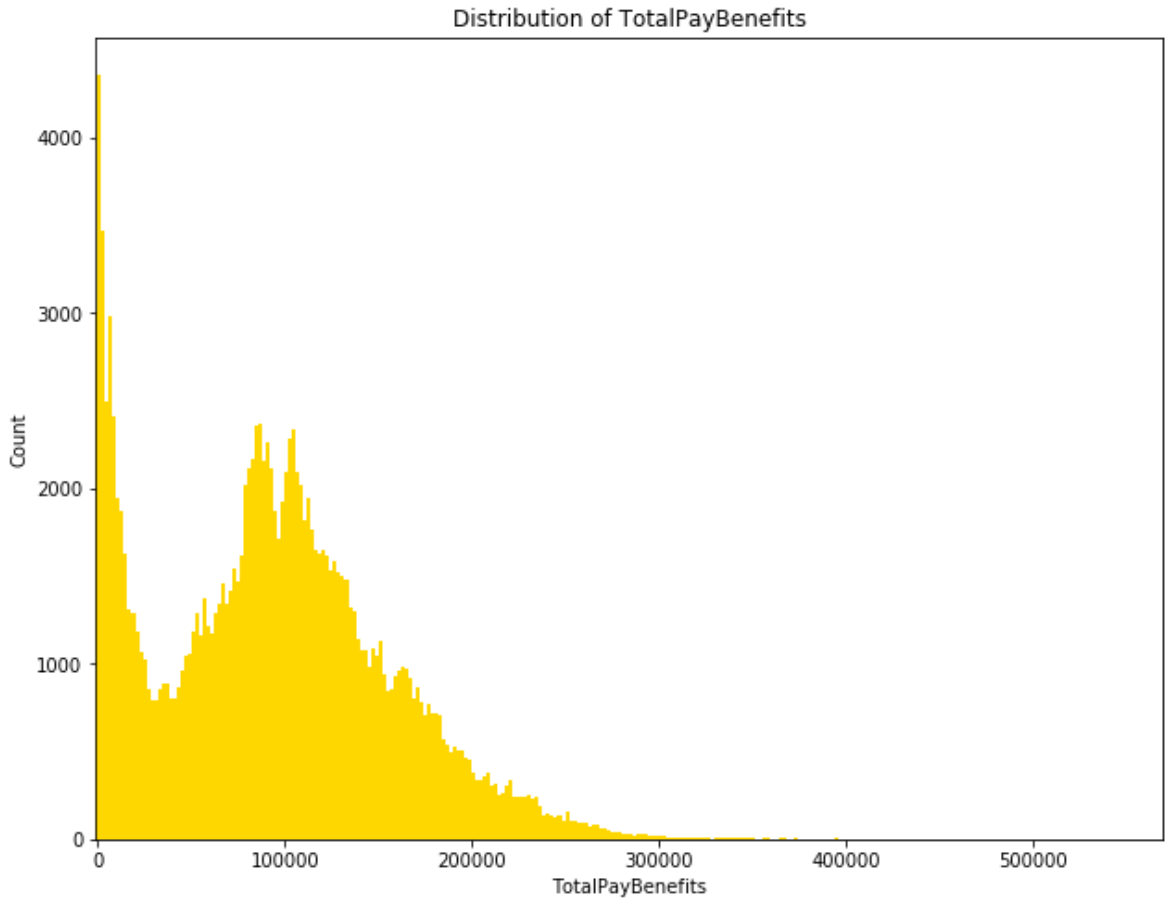
Question: Make a histogram to show distribution of the variable `TotalPayBenefits`.

In [16]:

```
x = []
y = []
step = 2000
for i in range(-1000,570000,step):
    x.append(i)
for i in range(len(x)):
    y.append(Salaries_df.loc[(Salaries_df.TotalPayBenefits >= x[i]) & (Salaries_df.TotalPayBenefits

fig = plt.figure(figsize=(8, 6))
axes = fig.add_axes([0, 0, 1, 1]) # left, bottom, width, height (range 0 to 1)
axes.bar(x,y,fc = "gold",width = step)
plt.xlim(-1000,570000)
axes.set_xlabel('TotalPayBenefits')
axes.set_ylabel('Count')
axes.set_title('Distribution of TotalPayBenefits')
```

Out[16]:

```
Text(0.5,1,'Distribution of TotalPayBenefits')
```



Question: Calculate the (min, 1st quartile, median, 3rd quartile, max) of the variable `TotalPayBenefits`.

In [17]:

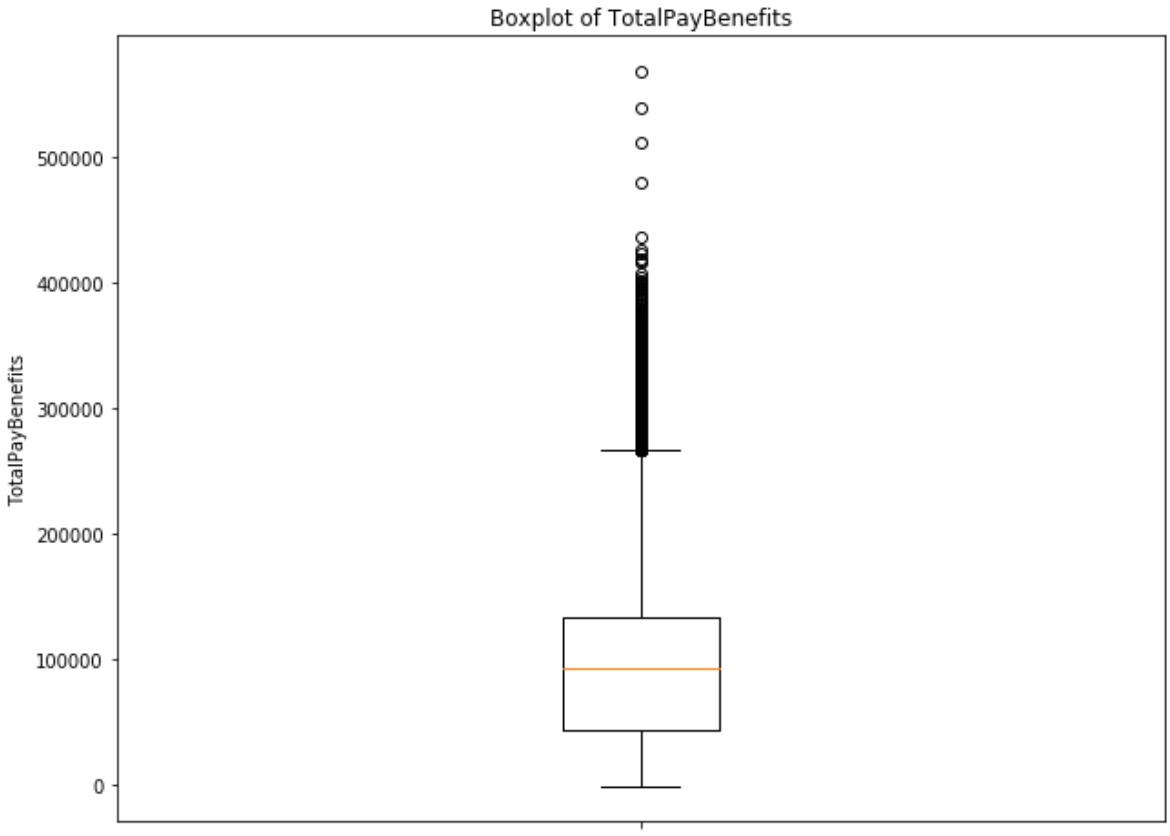
```
t = Salaries_df.sort_values("TotalPayBenefits")
t = t["TotalPayBenefits"]

print("min",t.iloc[0])
print("1st quartile",t.iloc[int(len(t) / 4)])
print("median",t.iloc[int(len(t) / 2)])
print("3st quartile",t.iloc[int(3 * len(t) / 4)])
print("max",t.iloc[int(len(t) - 1)])
```

```
min -618.13
1st quartile 44064.41
median 92405.97
3st quartile 132876.5
max 567595.43
```

Question: Make a box plot for the variable `TotalPayBenefits`.

In [69]:

```
fig = plt.figure(figsize=(8, 6))
axes = fig.add_axes([0, 0, 1, 1])
axes.boxplot(t)
axes.set_ylabel('TotalPayBenefits')
axes.set_xticklabels('')
axes.set_title('Boxplot of TotalPayBenefits');
```



Question: What you can conclude from the box plot above?

```
Most of people hold a salary lower than 300,000￥, while there literally exists a handful of
people have an extremely high salary.
Over a half of people(person-years) EVEN can't get a salary over 100,000￥ annually.
```

The end