



jupyter project Last Checkpoint: a few seconds ago (autosaved) Current Kernel Logo

Logout

Menu

Python 3 (ipykernel)

Not Trusted

- [File](#)
 - [New Notebook](#)

Dropdown

 - [Python 3 \(ipykernel\)](#)
 - [Open...](#)
 -
 - [Make a Copy...](#)
 - [Save as...](#)
 - [Rename...](#)
 - [Save and Checkpoint](#)

Ctrl-S
 -
 - [Revert to Checkpoint](#)

Dropdown

 - [Tuesday, 25 July 2023 14:22](#)
 -
 - [Print Preview](#)
 - [Download as](#)

Dropdown

 - [AsciiDoc \(.asciidoc\)](#)
 - [HTML \(.html\)](#)
 - [LaTeX \(.tex\)](#)
 - [Markdown \(.md\)](#)
 - [Notebook \(.ipynb\)](#)
 - [PDF via LaTeX \(.pdf\)](#)
 - [PDF via HTML \(.html\)](#)
 - [PNG via HTML \(.html\)](#)
 - [reST \(.rst\)](#)
 - [Python \(.py\)](#)
 - [Reveal.js slides \(.slides.html\)](#)
 - [PDF via HTML \(.html\)](#)
 - [Deploy as](#)
 -
 - [Trust Notebook](#)
 -
 - [Close and Halt](#)
- [Edit](#)
 - [Cut Cells](#)

x
 - [Copy Cells](#)

c
 - [Paste Cells Above](#)

Shift-V
 - [Paste Cells Below](#)

v
 - [Paste Cells & Replace](#)
 - [Delete Cells](#)

D
 - [Undo Delete Cells](#)

Z
 -
 - [Split Cell](#)

Ctrl-Shift-Minus
 - [Merge Cell Above](#)
 - [Merge Cell Below](#)
 -
 - [Move Cell Up](#)
 - [Move Cell Down](#)
 -
 - [Edit Notebook Metadata](#)
 -

- [Find and Replace](#)
-
- [Cut Cell Attachments](#)
- [Copy Cell Attachments](#)
- [Paste Cell Attachments](#)
-
- [Insert Image](#)
- [View](#)
 - [Toggle Header](#)
 - [Toggle Toolbar](#)
 - [Toggle Line Numbers](#)[Shift-L](#)
 - [Cell Toolbar](#)
 - [None](#)
 - [Edit Metadata](#)
 - [Raw Cell Format](#)
 - [Slideshow](#)
 - [Attachments](#)
 - [Tags](#)
- [Insert](#)
 - [Insert Cell Above](#)[A](#)
 - [Insert Cell Below](#)[B](#)
- [Cell](#)
 - [Run Cells](#)[Ctrl-Enter](#)
 - [Run Cells and Select Below](#)[Shift-Enter](#)
 - [Run Cells and Insert Below](#)[Alt-Enter](#)
 - [Run All](#)
 - [Run All Above](#)
 - [Run All Below](#)
 -
 - [Cell Type](#)
 - [Code](#)[y](#)
 - [Markdown](#)[M](#)
 - [Raw NB](#)[Convert](#)[R](#)
 -
 - [Current Outputs](#)
 - [Toggle](#)[0](#)
 - [Toggle Scrolling](#)[Shift-0](#)
 - [Clear](#)
 - [All Output](#)
 - [Toggle](#)
 - [Toggle Scrolling](#)
 - [Clear](#)
- [Kernel](#)
 - [Interrupt](#)[I,I](#)
 - [Restart](#)[0,0](#)
 - [Restart & Clear Output](#)
 - [Restart & Run All](#)
 - [Reconnect](#)
 - [Shutdown](#)
 -
 - [Change kernel](#)
 - [Python 3 \(ipykernel\)](#)
- [Widgets](#)
 - [Save Notebook Widget State](#)
 - [Clear Notebook Widget State](#)
 - [Download Widget State](#)
 - [Embed Widgets](#)
- [Help](#)
 - [User Interface Tour](#)
 - [Keyboard Shortcuts](#)[H](#)
 - [Edit Keyboard Shortcuts](#)

-
- [Notebook Help](#)
- [Markdown](#)
-
- [Python Reference](#)
- [IPython Reference](#)
- [NumPy Reference](#)
- [SciPy Reference](#)
- [Matplotlib Reference](#)
- [SymPy Reference](#)
- [pandas Reference](#)
-
- [About](#)

Run

Code

▼

xxxxxxxxxx

```
# ANALYSIS REPORT OF ABC COMPANY FOR THE DURATION
# 2022-2023
```

ANALYSIS REPORT OF ABC COMPANY FOR THE DURATION

2022-2023

In [1]:

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
```

xxxxxxxxxx

```
## 1)Reading Preprocessed Dataset
```

1)Reading Preprocessed Dataset

In [6]:

```
df=pd.read_csv("C://Users//dell//Downloads//myexcel.csv")
df
```

Out[6]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	06-Feb	180	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99	SF	25	06-Jun	235	Marquette	6796117.0
2	John Holland	Boston Celtics	30	SG	27	06-May	205	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28	SG	22	06-May	185	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8	PF	29	06-Oct	231	NaN	5000000.0
...
453	Shelvin Mack	Utah Jazz	8	PG	26	06-Mar	203	Butler	2433333.0
454	Raul Neto	Utah Jazz	25	PG	24	06-Jan	179	NaN	900000.0
455	Tibor Pleiss	Utah Jazz	21	C	26	07-Mar	256	NaN	2900000.0
456	Jeff Withey	Utah Jazz	24	C	26	7-0	231	Kansas	947276.0
457	Priyanka	Utah Jazz	34	C	25	07-Mar	231	Kansas	947276.0

458 rows × 9 columns
In [7]:

```
df.head()
```

Out[7]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	06-Feb	180	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99	SF	25	06-Jun	235	Marquette	6796117.0
2	John Holland	Boston Celtics	30	SG	27	06-May	205	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28	SG	22	06-May	185	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8	PF	29	06-Oct	231	NaN	5000000.0

In [8]:

```
df.tail()
```

Out[8]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
453	Shelvin Mack	Utah Jazz	8	PG	26	06-Mar	203	Butler	2433333.0
454	Raul Neto	Utah Jazz	25	PG	24	06-Jan	179	NaN	900000.0
455	Tibor Pleiss	Utah Jazz	21	C	26	07-Mar	256	NaN	2900000.0
456	Jeff Withey	Utah Jazz	24	C	26	7-0	231	Kansas	947276.0
457	Priyanka	Utah Jazz	34	C	25	07-Mar	231	Kansas	947276.0

In [9]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 458 entries, 0 to 457
Data columns (total 9 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Name        458 non-null   object
1   Team        458 non-null   object
2   Number      458 non-null   int64
3   Position    458 non-null   object
4   Age         458 non-null   int64
5   Height      458 non-null   object
6   Weight      458 non-null   int64
7   College     374 non-null   object
```

8 Salary 447 non-null float64
dtypes: float64(1), int64(3), object(5)
memory usage: 32.3+ KB
In [10]:

df.describe()
Out[10]:

	Number	Age	Weight	Salary
count	458.000000	458.000000	458.000000	4.470000e+02
mean	17.713974	26.934498	221.543668	4.833970e+06
std	15.966837	4.400128	26.343200	5.226620e+06
min	0.000000	19.000000	161.000000	3.088800e+04
25%	5.000000	24.000000	200.000000	1.025210e+06
50%	13.000000	26.000000	220.000000	2.836186e+06
75%	25.000000	30.000000	240.000000	6.500000e+06
max	99.000000	40.000000	307.000000	2.500000e+07

xxxxxxxxxx

Fill missing values in college

Fill missing values in college

In [11]:

from statistics import mode,mean
In [12]:

mode_college=df['College'].mode()[0]
df['College'].fillna(mode_college,inplace=True)

In [96]:

df.head()
Out[96]:

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	06-Feb	180	Texas	7730337.0
1	Jae Crowder	Boston Celtics	99	SF	25	06-Jun	235	Marquette	6796117.0
2	John Holland	Boston Celtics	30	SG	27	06-May	205	Boston University	NaN
3	R.J. Hunter	Boston Celtics	28	SG	22	06-May	185	Georgia State	1148640.0
4	Jonas Jerebko	Boston Celtics	8	PF	29	06-Oct	231	Kentucky	5000000.0

xxxxxxxxxx

Fill Missing Values In Salary

Fill Missing Values In Salary

In [13]:

```
mean_salary=df['Salary'].mean()  
df['Salary'].fillna(mean_salary,inplace=True)
```

In [14]:

```
df.head()  
Out[14]:
```

	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0	PG	25	06-Feb	180	Texas	7.730337e+06
1	Jae Crowder	Boston Celtics	99	SF	25	06-Jun	235	Marquette	6.796117e+06
2	John Holland	Boston Celtics	30	SG	27	06-May	205	Boston University	4.833970e+06
3	R.J. Hunter	Boston Celtics	28	SG	22	06-May	185	Georgia State	1.148640e+06
4	Jonas Jerebko	Boston Celtics	8	PF	29	06-Oct	231	Kentucky	5.000000e+06

xxxxxxxxxx

DROPPING UNWANTED COLUMNS

DROPPING UNWANTED COLUMNS

In [15]:

```
df=df.drop(df.columns[2],axis=1)
```

```
df  
Out[15]:
```

	Name	Team	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	PG	25	06-Feb	180	Texas	7.730337e+06
1	Jae Crowder	Boston Celtics	SF	25	06-Jun	235	Marquette	6.796117e+06
2	John Holland	Boston Celtics	SG	27	06-May	205	Boston University	4.833970e+06
3	R.J. Hunter	Boston Celtics	SG	22	06-May	185	Georgia State	1.148640e+06
4	Jonas Jerebko	Boston Celtics	PF	29	06-Oct	231	Kentucky	5.000000e+06
...
453	Shelvin Mack	Utah Jazz	PG	26	06-Mar	203	Butler	2.433333e+06
454	Raul Neto	Utah Jazz	PG	24	06-Jan	179	Kentucky	9.000000e+05
455	Tibor Pleiss	Utah Jazz	C	26	07-Mar	256	Kentucky	2.900000e+06
456	Jeff Withey	Utah Jazz	C	26	7-0	231	Kansas	9.472760e+05
457	Priyanka	Utah Jazz	C	25	07-Mar	231	Kansas	9.472760e+05

458 rows × 8 columns

xxxxxxxxxx

Data Preprocessing

Data Preprocessing

In [16]:

```
df['Height']=np.random.randint(160,195,458)
```

df
Out[16]:

	Name	Team	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	PG	25	177	180	Texas	7.730337e+06
1	Jae Crowder	Boston Celtics	SF	25	162	235	Marquette	6.796117e+06
2	John Holland	Boston Celtics	SG	27	180	205	Boston University	4.833970e+06
3	R.J. Hunter	Boston Celtics	SG	22	182	185	Georgia State	1.148640e+06
4	Jonas Jerebko	Boston Celtics	PF	29	166	231	Kentucky	5.000000e+06
...
453	Shelvin Mack	Utah Jazz	PG	26	166	203	Butler	2.433333e+06
454	Raul Neto	Utah Jazz	PG	24	185	179	Kentucky	9.000000e+05
455	Tibor Pleiss	Utah Jazz	C	26	194	256	Kentucky	2.900000e+06
456	Jeff Withey	Utah Jazz	C	26	178	231	Kansas	9.472760e+05
457	Priyanka	Utah Jazz	C	25	171	231	Kansas	9.472760e+05

458 rows × 8 columns

xxxxxxxxxx

2)Analysing The Dataset

2)Analysing The Dataset¶

xxxxxxxxxx

<P></p> 1.How many are there in each team and the percentage splitting with respect to the total employee?
1.How many are there in each team and the percentage splitting with respect to the total employee?
In [17]:

#Team count

```
Team_count=df['Team'].value_counts()
```

Team_count
Out[17]:

Team	
New Orleans Pelicans	19
Memphis Grizzlies	18
Utah Jazz	16
New York Knicks	16
Milwaukee Bucks	16
Brooklyn Nets	15
Portland Trail Blazers	15
Oklahoma City Thunder	15
Denver Nuggets	15
Washington Wizards	15
Miami Heat	15
Charlotte Hornets	15
Atlanta Hawks	15
San Antonio Spurs	15
Houston Rockets	15
Boston Celtics	15
Indiana Pacers	15

```
Detroit Pistons      15
Cleveland Cavaliers  15
Chicago Bulls        15
Sacramento Kings     15
Phoenix Suns         15
Los Angeles Lakers   15
Los Angeles Clippers 15
Golden State Warriors 15
Toronto Raptors      15
Philadelphia 76ers    15
Dallas Mavericks     15
Orlando Magic        14
Minnesota Timberwolves 14
Name: count, dtype: int64
In [18]:
```

```
#Total number of team
```

```
length=len(df)
In [19]:
```

```
#Team percentage
```

```
Team_percentage=(Team_count)/length*100
```

```
Team_percentage
Out[19]:
```

```
Team
New Orleans Pelicans      4.148472
Memphis Grizzlies         3.930131
Utah Jazz                 3.493450
New York Knicks           3.493450
Milwaukee Bucks           3.493450
Brooklyn Nets             3.275109
Portland Trail Blazers    3.275109
Oklahoma City Thunder     3.275109
Denver Nuggets            3.275109
Washington Wizards        3.275109
Miami Heat                3.275109
Charlotte Hornets         3.275109
Atlanta Hawks             3.275109
San Antonio Spurs         3.275109
Houston Rockets           3.275109
Boston Celtics            3.275109
Indiana Pacers            3.275109
Detroit Pistons           3.275109
Cleveland Cavaliers       3.275109
Chicago Bulls             3.275109
Sacramento Kings          3.275109
Phoenix Suns              3.275109
Los Angeles Lakers        3.275109
Los Angeles Clippers      3.275109
Golden State Warriors     3.275109
Toronto Raptors           3.275109
Philadelphia 76ers        3.275109
Dallas Mavericks          3.275109
Orlando Magic             3.056769
Minnesota Timberwolves    3.056769
Name: count, dtype: float64
```

```
xxxxxxxxxx
```

2.Segragate the employee with respect to different position

2.Segragate the employee with respect to different position¶

```
In [20]:
```



```
#position count
position_count=df.groupby(['Position'])['Position'].count()
position_count
```

```
Out[20]:
Position
C      79
PF    100
PG     92
SF     85
SG    102
Name: Position, dtype: int64

xxxxxxxxxx
```

3. FIND WHICH AGE GROUP MOST OF THE EMPLOYEE BELONGS TO

3. FIND WHICH AGE GROUP MOST OF THE EMPLOYEE BELONGS TO

```
In [21]:
age_value_count=df["Age"].value_counts()
age_value_count
Out[21]:
```

```
Age
24    47
25    46
27    41
23    41
26    36
28    31
30    31
29    28
22    26
31    22
20    19
21    19
33    14
32    13
34    10
36    10
35     9
37     4
38     4
40     3
39     2
19     2
Name: count, dtype: int64
```

```
In [22]:
age_most=age_value_count.head(1).index[0]

print("Most of the employee belongs to:",age_most)
Most of the employee belongs to: 24

xxxxxxxxxx
```

4.find out under which team and position spending in term of salary is high

4.find out under which team and position spending in term of salary is high

In [23]:

```
salary_sum=df.groupby(['Team', 'Salary'])['Position'].sum()
```

salary_sum
Out[23]:

Team	Salary	
Atlanta Hawks	525093.0	SG
	947276.0	PF
	1000000.0	PFC
	1304520.0	SG
	1763400.0	PG
...		
Washington Wizards	5694674.0	SG
	8000000.0	PF
	11217391.0	C
	13000000.0	C
	15851950.0	PG
Name: Position, Length: 433, dtype: object		

In [24]:

```
#highest salary spended
high_salary=df[df['Salary']==df['Salary'].max()]

print("highest salary spended:",high_salary)
highest salary spended:      Name      Team Position Age Height Weight College
109  Kobe Bryant  Los Angeles Lakers      SF    37    183    212  Kentucky  \

      Salary
109  25000000.0

xxxxxxxxxx
```

4.Find if there any correlation between age and salary,represent it visually
4.Find if there any correlation between age and salary,represent it visually

In [25]:

```
#Calculating the correlation
correlation=df['Age'].corr(df['Salary'])

print(correlation)

print('correlation between age and salary is possitive so the respective variable is in direct relationship')
0.2111695491768639
correlation between age and salary is possitive so the respective variable is in direct relationship
```

In [26]:

```
#creating a scatter plot
age=df['Age']
salary=df['Salary']

plt.scatter(x=age,y=salary)

plt.title('correlation between Age and Salary')

plt.show()
```

correlation between Age and Salary



In []: