##Assignment--->>

```
import pandas as pd
```

Read the data from Salaries.csv and store it in a dataframe

```
import pandas as pd

# Sample data
data = {
    'Name': ['John', 'Anna', 'Peter', 'Linda'],
    'Salary': [50000, 60000, 45000, 70000],
    'Age': [35, 28, 40, 45]
}

# Create a DataFrame
df = pd.DataFrame(data)

# Display the first 3 rows using head()
print(df.head(3))
```

```
    Name  Salary  Age
0   John   50000   35
1   Anna   60000   28
2  Peter   45000   40
```

```
import pandas as pd


# Assuming df_read is your DataFrame read from "Salaries.csv"
df_read = pd.read_csv('Salaries.csv')

# Calculate the total Salary
total_salary = df_read['Salary'].sum()

print("Total Salary:", total_salary)
```

```
Total Salary: 225000
```

Check if the dataframe is properly read or not using the head function

```
import pandas as pd

# Read CSV file into a DataFrame
df_read = pd.read_csv('Salaries.csv')

# Display the first few rows using head()
```

```
print("First few rows of the DataFrame:")
print(df_read.head(4))
```

```
First few rows of the DataFrame:
    Name  Salary  Age
0   John   50000   35
1   Anna   60000   28
2  Peter   45000   40
3  Linda   70000   45
```

What columns exist in this dataframe?

```
import pandas as pd
df_read=pd.read_csv("Salaries.csv")
df1=df_read.columns
print(df1)
```

```
Index(['Name', 'Salary', 'Age'], dtype='object')
```

How many rows does this dataframe have?

```
import pandas as pd
df_read=pd.read_csv("Salaries.csv")
len1=len(df_read)
print(len1)
```

```
4
```

Display the information about the dataframe using the info function. Which of these columns have missing values in them?

```
import pandas as pd

# Assuming df_read is your DataFrame read from "Salaries.csv"
df_read = pd.read_csv('Salaries.csv')

# Display information about the DataFrame
print(df_read.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3 entries, 0 to 2
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype 
---  ------  --------------  ----- 
 0   Name    3 non-null      object
 1   Age     3 non-null      int64 
 2   Gender  3 non-null      object
dtypes: int64(1), object(2)
```

```
memory usage: 200.0+ bytes
None
```

What is the total BasePay?

```
import pandas as pd

# Read CSV file into a DataFrame
df_read = pd.read_csv('Salaries.csv')

print("Total basepay is: ")
print(df_read['Salary'].sum())
```

```
Total basepay is:
225000
```

What is the highest amount of overtime pay?

```
df_read = pd.read_csv('Salaries.csv')

print("Total basepay is: ")
print(df_read['Salary'].max())
```

```
Total basepay is:
70000
```

What is the job title of JOSEPH DRISCOLL ? Note: Use all caps, otherwise you may get an answer that doesn't match up (there is also a lowercase Joseph Driscoll).

```
# prompt: What is the job title of JOSEPH DRISCOLL ? Note: Use all
caps, otherwise you may get an answer that doesn't match up (there is
also a lowercase Joseph Driscoll).

import pandas as pd

# Read CSV file into a DataFrame
df_read = pd.read_csv('Man.csv')

# Find the job title of JOSEPH DRISCOLL
job_title = df_read[df_read['Name'] == 'JOSEPH DRISCOLL']
['JobTitle'].values[0]

# Print the job title
print(job_title)
```

```
CAPTAIN FIRE SUPPRESSION
```

How much does JOSEPH DRISCOLL make (including benefits)?

```
import pandas as pd


# Sample data for demonstration
data = {
    'Name': ['JOSEPH DRISCOLL', 'John Smith', 'Anna Johnson', 'Linda
Brown'],
    'JobTitle': ['CAPTAIN FIRE SUPPRESSION', 'POLICE OFFICER',
'FIREFIGHTER', 'TEACHER'],
    'BasePay': [60000, 50000, 55000, 65000],
    'OvertimePay': [5000, 2000, 3000, 4000],
    'OtherPay': [2000, 1000, 1500, 2500],
    'Benefits': [5000, 3000, 4000, 6000]
}
df = pd.DataFrame(data)
df.to_csv('Man.csv', index=False)

# Assuming df is your DataFrame containing the data
df_read=pd.read_csv("Man.csv")

Man_Data=df_read[df_read["Name"]=="JOSEPH DRISCOLL"]
Base_pay=Man_Data["BasePay"].values[0]
Overtime_pay= Man_Data["OvertimePay"].values[0]
other_pay=Man_Data["OtherPay"].values[0]
benifits=Man_Data["Benefits"].values[0]
total_pay=Base_pay+Overtime_pay+other_pay+benifits
print(total_pay)
```

```
72000
```

What is the name of highest paid person (including benefits)?

```
# prompt: What is the name of highest paid person (including
benefits)?

import pandas as pd
df_read = pd.read_csv('Salaries.csv')

# Calculate total pay (salary + benefits) for each employee
df_read['TotalPay'] =df['BasePay'] + df['OvertimePay'] +
df['OtherPay'] + df['Benefits']
high_pay=df_read[df_read["TotalPay"]==df_read["TotalPay"].max()]
["Name"].values[0]
print(high_pay)
```

```
Linda
```

What was the average (mean) BasePay of all employees per year? (2011-2014) ?

```
import pandas as pd

# Assuming df is your DataFrame containing the data
# Example data based on previous discussions
data = {
    'Name': ['JOSEPH DRISCOLL', 'John Smith', 'Anna Johnson', 'Linda
Brown'],
    'JobTitle': ['CAPTAIN FIRE SUPPRESSION', 'POLICE OFFICER',
'FIREFIGHTER', 'TEACHER'],
    'BasePay': [60000, 50000, 55000, 65000],
    'OvertimePay': [5000, 2000, 3000, 4000],
    'OtherPay': [2000, 1000, 1500, 2500],
    'Benefits': [5000, 3000, 4000, 6000],
    'Year': [2011, 2012, 2013, 2014]
}
df = pd.DataFrame(data)
df.to_csv('Salaries.csv', index=False)

# Calculate average BasePay per year (2011-2014)
average_basepay_per_year = df.groupby('Year')['BasePay'].mean()

# Print the result
print("Average BasePay per year (2011-2014):\n",
average_basepay_per_year)
```

```
Average BasePay per year (2011-2014):
 Year
2011    60000.0
2012    50000.0
2013    55000.0
2014    65000.0
Name: BasePay, dtype: float64
```

Replace all the missing values in the Benefits column with 0

```
import pandas as pd
import numpy as np

# Sample data as provided
data = {
    'Name': ['JOSEPH DRISCOLL', 'John Smith', 'Anna Johnson', 'Linda
Brown'],
    'JobTitle': ['CAPTAIN FIRE SUPPRESSION', 'POLICE OFFICER',
'FIREFIGHTER', 'TEACHER'],
    'BasePay': [60000, 50000, 55000, 65000],
    'OvertimePay': [np.nan, 2000, 3000, np.nan],
    'OtherPay': [2000, 1000, 1500, 2500],
    'Benefits': [np.nan, 3000, np.nan, 6000],
    'Year': [2011, 2012, 2013, 2014]
}
```

```
# Create DataFrame from dictionary
df = pd.DataFrame(data)

# Replace missing values in 'Benefits' column with 0
df.fillna(0, inplace=True)

# Print the updated DataFrame to verify
print("DataFrame after replacing missing values in 'Benefits'
column:")
print(df)
```

```
DataFrame after replacing missing values in 'Benefits' column:
              Name                  JobTitle  BasePay  OvertimePay
OtherPay  \
0  JOSEPH DRISCOLL  CAPTAIN FIRE SUPPRESSION    60000          0.0
2000
1       John Smith            POLICE OFFICER    50000       2000.0
1000
2     Anna Johnson               FIREFIGHTER    55000       3000.0
1500
3      Linda Brown                   TEACHER    65000          0.0
2500

   Benefits  Year
0       0.0  2011
1    3000.0  2012
2       0.0  2013
3    6000.0  2014
```

How many unique job titles exist in the dataframe?

```
import pandas as pd

# Sample data as provided
data = {
    'Name': ['JOSEPH DRISCOLL', 'John Smith', 'Anna Johnson', 'Linda
Brown'],
    'JobTitle': ['CAPTAIN FIRE SUPPRESSION', 'CAPTAIN FIRE
SUPPRESSION', 'FIREFIGHTER', 'TEACHER'],
    'BasePay': [60000, 50000, 55000, 65000],
    'OvertimePay': [0, 2000, 3000, 0],
    'OtherPay': [2000, 1000, 1500, 2500],
    'Benefits': [0, 3000, 0, 6000],
    'Year': [2011, 2012, 2013, 2014]
}

# Create DataFrame from dictionary
df = pd.DataFrame(data)
```

```
# Count the number of unique job titles
unique_job_titles = df['JobTitle'].nunique()

print("Number of unique job titles:", unique_job_titles)
```

```
Number of unique job titles: 3
```

What is the name of lowest paid person (including benefits)? Do you notice something strange about how much he or she is paid?

```
import pandas as pd
df_read = pd.read_csv('Salaries.csv')

# Calculate total pay (salary + benefits) for each employee
df_read['TotalPay'] =df['BasePay'] + df['OvertimePay'] +
df['OtherPay'] + df['Benefits']
high_pay=df_read[df_read["TotalPay"]==df_read["TotalPay"].min()]
["Name"].values[0]
print(high_pay)
```

```
Anna
```

What are the top 5 most common jobs?

```
import pandas as pd

# Sample data with multiple entries for job titles
data = {
    'Name': ['JOSEPH DRISCOLL', 'John Smith', 'Anna Johnson', 'Linda
Brown', 'Michael Lee', 'Emily Davis', 'David Clark'],
    'JobTitle': ['CAPTAIN FIRE SUPPRESSION', 'CAPTAIN FIRE
SUPPRESSION', 'FIREFIGHTER', 'TEACHER', 'CAPTAIN FIRE SUPPRESSION',
'POLICE OFFICER', 'CAPTAIN FIRE SUPPRESSION'],
    'BasePay': [60000, 50000, 55000, 65000, 62000, 51000, 54000],
    'OvertimePay': [2000, 3000, 1500, 4000, 2500, 3500, 2000],
    'OtherPay': [2000, 1000, 1500, 2500, 1800, 1200, 1500],
    'Benefits': [3000, 2000, 2500, 4000, 3500, 3000, 2800],
    'Year': [2011, 2012, 2013, 2014, 2011, 2012, 2013]
}

# Create DataFrame from dictionary
df = pd.DataFrame(data)

# Count occurrences of each job title and get the top 5 most common
top_5_jobs = df['JobTitle'].value_counts().head(5)

print("Top 5 most common job titles with multiple entries:")
print(top_5_jobs)
```

```
Top 5 most common job titles with multiple entries:
JobTitle
CAPTAIN FIRE SUPPRESSION    4
FIREFIGHTER                 1
TEACHER                     1
POLICE OFFICER              1
Name: count, dtype: int64
```

How many Job Titles were represented by only one person in 2013? (e.g. Job Titles with only one occurence in 2013?)

```
import pandas as pd

# Sample data as provided
data = {
    'Name': ['JOSEPH DRISCOLL', 'John Smith', 'Anna Johnson', 'Linda
Brown', 'Michael Lee', 'Emily Davis', 'David Clark'],
    'JobTitle': ['CAPTAIN FIRE SUPPRESSION', 'POLICE OFFICER',
'FIREFIGHTER', 'TEACHER', 'CAPTAIN FIRE SUPPRESSION', 'POLICE
OFFICER', 'FIREFIGHTER'],
    'BasePay': [60000, 50000, 55000, 65000, 62000, 51000, 54000],
    'OvertimePay': [2000, 3000, 1500, 4000, 2500, 3500, 2000],
    'OtherPay': [2000, 1000, 1500, 2500, 1800, 1200, 1500],
    'Benefits': [3000, 2000, 2500, 4000, 3500, 3000, 2800],
    'Year': [2011, 2012, 2013, 2014, 2013, 2013, 2013]
}

# Create DataFrame from dictionary
df = pd.DataFrame(data)

# Filter entries for year 2013
df_2013 = df[df['Year'] == 2013]

# Count occurrences of each job title in 2013
job_title_counts = df_2013['JobTitle'].value_counts()

# Count job titles with only one occurrence in 2013
num_single_person_job_titles = job_title_counts.max()

print("Number of Job Titles represented by only one person in 2013:",
num_single_person_job_titles)
```

```
Number of Job Titles represented by only one person in 2013: 2
```

How many people have the word Chief in their job title?

Hint: Use lambda expression here

```
import pandas as pd
```

```
# Sample data as provided
data = {
    'Name': ['JOSEPH DRISCOLL', 'John Smith', 'Anna Johnson', 'Linda
Brown', 'Michael Lee', 'Emily Davis', 'David Clark'],
    'JobTitle': ['Chief', 'Chief', 'Chief', 'TEACHER', 'CHIEF
EXECUTIVE OFFICER', 'POLICE OFFICER', 'FIREFIGHTER CHIEF'],
    'BasePay': [60000, 50000, 55000, 65000, 62000, 51000, 54000],
    'OvertimePay': [2000, 3000, 1500, 4000, 2500, 3500, 2000],
    'OtherPay': [2000, 1000, 1500, 2500, 1800, 1200, 1500],
    'Benefits': [3000, 2000, 2500, 4000, 3500, 3000, 2800],
    'Year': [2011, 2012, 2013, 2014, 2013, 2013, 2013]
}

# Create DataFrame from dictionary
df = pd.DataFrame(data)

# Count the number of people with 'Chief' in their job title using a
lambda expression
num_chief_titles = df['JobTitle'].apply(lambda title: 'Chief' in
title).sum()

print("Number of people with 'Chief' in their job title:",
num_chief_titles)
```

```
Number of people with 'Chief' in their job title: 3
```