

TASK #1 and TASK #3

>Rename the given data frame as df_Edata
>Display the data frame present in the given url

```
import pandas as pd
```

#Specify the file path

```
file_path = r"H:\My Drive\Data Analytics Course\Source Files\employees.csv"
```

#Read the CSV File

```
df_Edata=pd.read_csv(file_path)
```

#Print the content in the file

```
print(df_Edata)
```

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	
HIRE_DATE \						
0	198	Donald	OConnell	DOCONNEL	650.507.9833	21-Jun-07
1	199	Douglas	Grant	DGRANT	650.507.9844	13-Jan-08
2	200	Jennifer	Whalen	JWHALEN	515.123.4444	17-Sep-03
3	201	Michael	Hartstein	MHARTSTE	515.123.5555	17-Feb-04
4	202	Pat	Fay	PFAY	603.123.6666	17-Aug-05
5	203	Susan	Mavris	SMAVRIS	515.123.7777	07-Jun-02
6	204	Hermann	Baer	HBAER	515.123.8888	07-Jun-02
7	205	Shelley	Higgins	SHIGGINS	515.123.8080	07-Jun-02
8	206	William	Gietz	WGIETZ	515.123.8181	07-Jun-02
9	100	Steven	King	SKING	515.123.4567	17-Jun-03
10	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-Sep-05
11	102	Lex	De Haan	LDEHAAN	515.123.4569	13-Jan-01
12	103	Alexander	Hunold	AHUNOLD	590.423.4567	03-Jan-06
13	104	Bruce	Ernst	BERNST	590.423.4568	21-May-07
14	105	David	Austin	DAUSTIN	590.423.4569	25-Jun-05
15	106	Valli	Pataballa	VPATABAL	590.423.4560	05-Jun-06

Feb-06						
16	107	Diana	Lorentz	DLORENTZ	590.423.5567	07-
Feb-07						
17	108	Nancy	Greenberg	NGREENBE	515.124.4569	17-
Aug-02						
18	109	Daniel	Faviet	DFAVIET	515.124.4169	16-
Aug-02						
19	110	John	Chen	JCHEN	515.124.4269	28-
Sep-05						
20	111	Ismael	Sciarra	ISCIARRA	515.124.4369	30-
Sep-05						
21	112	Jose Manuel	Urman	JMURMAN	515.124.4469	07-
Mar-06						
22	113	Luis	Popp	LPOPP	515.124.4567	07-
Dec-07						
23	114	Den	Raphaely	DRAPHEAL	515.127.4561	07-
Dec-02						
24	115	Alexander	Khoo	AKH00	515.127.4562	18-
May-03						
25	116	Shelli	Baida	SBAIDA	515.127.4563	24-
Dec-05						
26	117	Sigal	Tobias	STOBIAS	515.127.4564	24-
Jul-05						
27	118	Guy	Himuro	GHIMURO	515.127.4565	15-
Nov-06						
28	119	Karen	Colmenares	KCOLMENA	515.127.4566	10-
Aug-07						
29	120	Matthew	Weiss	MWEISS	650.123.1234	18-
Jul-04						
30	121	Adam	Fripp	AFRIPP	650.123.2234	10-
Apr-05						
31	122	Payam	Kaufling	PKAUFLIN	650.123.3234	01-
May-03						
32	123	Shanta	Vollman	SVOLLMAN	650.123.4234	10-
Oct-05						
33	124	Kevin	Mourgos	KMOURGOS	650.123.5234	16-
Nov-07						
34	125	Julia	Nayer	JNAYER	650.124.1214	16-
Jul-05						
35	126	Irene	Mikkilineni	IMIKKILI	650.124.1224	28-
Sep-06						
36	127	James	Landry	JLANDRY	650.124.1334	14-
Jan-07						
37	128	Steven	Markle	SMARKLE	650.124.1434	08-
Mar-08						
38	129	Laura	Bissot	LBISSOT	650.124.5234	20-
Aug-05						
39	130	Mozhe	Atkinson	MATKINS0	650.124.6234	30-
Oct-05						

40	131	James	Marlow	JAMRLOW	650.124.7234	16-
Feb-05						
41	132	TJ	Olson	TJOLSON	650.124.8234	10-
Apr-07						
42	133	Jason	Mallin	JMALLIN	650.127.1934	14-
Jun-04						
43	134	Michael	Rogers	MROGERS	650.127.1834	26-
Aug-06						
44	135	Ki	Gee	KGEE	650.127.1734	12-
Dec-07						
45	136	Hazel	Philtanker	HPHILTAN	650.127.1634	06-
Feb-08						
46	137	Renske	Ladwig	RLADWIG	650.121.1234	14-
Jul-03						
47	138	Stephen	Stiles	SSTILES	650.121.2034	26-
Oct-05						
48	139	John	Seo	JSE0	650.121.2019	12-
Feb-06						
49	140	Joshua	Patel	JPATEL	650.121.1834	06-
Apr-06						

	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
0	SH_CLERK	2600	-	124	50
1	SH_CLERK	2600	-	124	50
2	AD_ASST	4400	-	101	10
3	MK_MAN	13000	-	100	20
4	MK_REP	6000	-	201	20
5	HR_REP	6500	-	101	40
6	PR_REP	10000	-	101	70
7	AC_MGR	12008	-	101	110
8	AC_ACCOUNT	8300	-	205	110
9	AD_PRES	24000	-	-	90
10	AD_VP	17000	-	100	90
11	AD_VP	17000	-	100	90
12	IT_PROG	9000	-	102	60
13	IT_PROG	6000	-	103	60
14	IT_PROG	4800	-	103	60
15	IT_PROG	4800	-	103	60
16	IT_PROG	4200	-	103	60
17	FI_MGR	12008	-	101	100
18	FI_ACCOUNT	9000	-	108	100
19	FI_ACCOUNT	8200	-	108	100
20	FI_ACCOUNT	7700	-	108	100
21	FI_ACCOUNT	7800	-	108	100
22	FI_ACCOUNT	6900	-	108	100
23	PU_MAN	11000	-	100	30
24	PU_CLERK	3100	-	114	30
25	PU_CLERK	2900	-	114	30
26	PU_CLERK	2800	-	114	30

27	PU_CLERK	2600	-	114	30
28	PU_CLERK	2500	-	114	30
29	ST_MAN	8000	-	100	50
30	ST_MAN	8200	-	100	50
31	ST_MAN	7900	-	100	50
32	ST_MAN	6500	-	100	50
33	ST_MAN	5800	-	100	50
34	ST_CLERK	3200	-	120	50
35	ST_CLERK	2700	-	120	50
36	ST_CLERK	2400	-	120	50
37	ST_CLERK	2200	-	120	50
38	ST_CLERK	3300	-	121	50
39	ST_CLERK	2800	-	121	50
40	ST_CLERK	2500	-	121	50
41	ST_CLERK	2100	-	121	50
42	ST_CLERK	3300	-	122	50
43	ST_CLERK	2900	-	122	50
44	ST_CLERK	2400	-	122	50
45	ST_CLERK	2200	-	122	50
46	ST_CLERK	3600	-	123	50
47	ST_CLERK	3200	-	123	50
48	ST_CLERK	2700	-	123	50
49	ST_CLERK	2500	-	123	50

TASK #3

>Rename the given data frame as df_Sdata

```
import pandas as pd
```

```
# Specify the file path
```

```
file_path = r"H:\My Drive\Data Analytics Course\Source Files\Data-analytic-course-assignment-sheet1.csv"
```

```
# Read the CSV file
```

```
df_Sdata = pd.read_csv(file_path)
```

```
# Display the Content in the file
```

```
print(df_Sdata)
```

	S.No	Name	Gender	Unnamed: 3	Height
0	1	A CHARAN	Male	NaN	165.0
1	2	A LOLITHA SREE	Female	NaN	165.0
2	3	A ROHINI	NaN	NaN	NaN
3	4	AKKAYYAGARI VENKATESWARLU	NaN	NaN	NaN

4	5	AYODHYAPURAM THARUN KUMAR REDDY	male	NaN	165.0
..
66	67	KOTAKONDA HARIKA	Female	NaN	153.0
67	68	KURMAI TRIBHUVAN KUMAR REDDY	male	NaN	180.0
68	69	MEKALA HARSHITHA	Female	NaN	153.0
69	70	MUMMADI AMRUTHA	Female	NaN	153.0
70	71	NARA DIVYABHARATHI	Female	NaN	154.0

	favourete	color	Cell phone brand	Laptop(Y/N)
0		Blue	Redmi	Y
1		Blue	Vivo	N
2		NaN	NaN	NaN
3		NaN	NaN	NaN
4		black	realme	N
..	
66		Black	samsung	N
67		black	OnePlus	N
68		Black	Redmi	N
69		white	Vivo	N
70		Black	VIVO	N

[71 rows x 8 columns]

TASK #4

Find the initial information and characterization of the df_Edata data frame

```

i)Data Type Information
ii)Data Types
iii)Column names
iv)Size of Data Frame

import pandas as pd

#Specify the file path
file_path = r"H:\My Drive\Data Analytics Course\Source Files\
employees.csv"

#Read the CSV File
df_Edata=pd.read_csv(file_path)

#df_Edata.info() - Used to display Column names, Datatype, No.of non-
null values in each column
print("Information about df_Edata:")

```

```

print(df_Edata.info())

#df_Edata.dtypes() - Check data types of all columns
print("\nData types in df_Edata:")
print(df_Edata.dtypes)

#df_Edata.columns() - Used to identify the column names of a given
Data set
print("\nColumn names in df_Edata:")
print(df_Edata.columns)

#df_Edata.shape() - Gives no of rows and columns
print("\nSize of df_Edata:")
print(df_Edata.shape)

```

```

Information about df_Edata:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50 entries, 0 to 49
Data columns (total 11 columns):
#   Column                Non-Null Count  Dtype
---  -
0   EMPLOYEE_ID           50 non-null    int64
1   FIRST_NAME            50 non-null    object
2   LAST_NAME             50 non-null    object
3   EMAIL                 50 non-null    object
4   PHONE_NUMBER          50 non-null    object
5   HIRE_DATE             50 non-null    object
6   JOB_ID               50 non-null    object
7   SALARY                50 non-null    int64
8   COMMISSION_PCT        50 non-null    object
9   MANAGER_ID            50 non-null    object
10  DEPARTMENT_ID         50 non-null    int64
dtypes: int64(3), object(8)
memory usage: 4.4+ KB
None

```

```

Data types in df_Edata:
EMPLOYEE_ID      int64
FIRST_NAME       object
LAST_NAME        object
EMAIL            object
PHONE_NUMBER     object
HIRE_DATE        object
JOB_ID           object
SALARY           int64
COMMISSION_PCT   object
MANAGER_ID       object
DEPARTMENT_ID    int64
dtype: object

```

```

Column names in df_Edata:
Index(['EMPLOYEE_ID', 'FIRST_NAME', 'LAST_NAME', 'EMAIL',
      'PHONE_NUMBER',
      'HIRE_DATE', 'JOB_ID', 'SALARY', 'COMMISSION_PCT',
      'MANAGER_ID',
      'DEPARTMENT_ID'],
      dtype='object')

Size of df_Edata:
(50, 11)

```

Find the initial information and characterization of the df_Sdata data frame

```

i)Data Type Information
ii)Data Types
iii)Column names
iv)Size of Data Frame

import pandas as pd

file_path = r"H:\My Drive\Data Analytics Course\Source Files\Data-
analytic-course-assignment-sheet1.csv"

#Read the CSV File
df_Sdata=pd.read_csv(file_path)

#df_Sdata.info() - Used to display Column names, Datatype, No.of non-
null values in each column
print("\nInformation about df_Sdata:")
print(df_Sdata.info())

#df_Sdata.dtypes() - Check data types of all columns
print("\nData types in df_Sdata:")
print(df_Sdata.dtypes)

#df_Sdata.columns() - Used to identify the column names of a given
Data set
print("\nColumn names in df_Sdata:")
print(df_Sdata.columns)

#df_Sdata.shape() - Gives no of rows and columns
print("\nSize of df_Sdata:")
print(df_Sdata.shape)

Information about df_Sdata:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 71 entries, 0 to 70
Data columns (total 8 columns):
#   Column                Non-Null Count  Dtype
---  -

```

```

0    S.No          71 non-null    int64
1    Name          71 non-null    object
2    Gender        53 non-null    object
3    Unnamed: 3     0 non-null     float64
4    Height        53 non-null    float64
5    favourete color 53 non-null    object
6    Cell phone brand 53 non-null    object
7    Laptop(Y/N)    53 non-null    object
dtypes: float64(2), int64(1), object(5)
memory usage: 4.6+ KB
None

Data types in df_Sdata:
S.No          int64
Name          object
Gender        object
Unnamed: 3     float64
Height        float64
favourete color object
Cell phone brand object
Laptop(Y/N)    object
dtype: object

Column names in df_Sdata:
Index(['S.No', 'Name', 'Gender', 'Unnamed: 3', 'Height', 'favourete color',
       'Cell phone brand', 'Laptop(Y/N)'],
      dtype='object')

Size of df_Sdata:
(71, 8)

```

TASK #4

Role of df.describe() with an Example

df.describe() - The df.describe() function in pandas provides a statistical summary of the numeric columns in a DataFrame. It gives an overview of central tendency, dispersion, and shape of the dataset's distribution.

Example is explained by the code given below

```

import pandas as pd

#Specify the file path
file_path = r"H:\My Drive\Data Analytics Course\Source Files\employees.csv"

#Read the CSV File

```



```
df_Edata=pd.read_csv(file_path)
```

```
df_Edata.describe()
```

	EMPLOYEE_ID	SALARY	DEPARTMENT_ID
count	50.000000	50.000000	50.000000
mean	134.760000	6182.320000	57.600000
std	33.631594	4586.181772	25.11687
min	100.000000	2100.000000	10.000000
25%	112.250000	2725.000000	50.000000
50%	124.500000	4600.000000	50.000000
75%	136.750000	8150.000000	60.000000
max	206.000000	24000.000000	110.000000

TASK #5

Consider the df_Sdata data frame and do the following

1. Find the total number of students
2. How many girls in the class
3. How many Boys in the class
4. what is the maximum height of students and find that student S.No or name?

```
import pandas as pd
```

```
# Specify the file path
```

```
file_path = r"H:\My Drive\Data Analytics Course\Source Files\Data-analytic-course-assignment-sheet1.csv"
```

```
# Read the CSV file
```

```
df_Sdata = pd.read_csv(file_path)
```

```
# 1. Total number of students
```

```
total_students = len(df_Sdata)
```

```
# 2. Number of girls in the class
```

```
girls_count = len(df_Sdata[df_Sdata["Gender"] == "Girl"])
```

```
# 3. Number of boys in the class
```

```
boys_count = len(df_Sdata[df_Sdata["Gender"] == "Boy"])
```

```
# 4. Maximum height and the corresponding student
```

```
max_height = df_Sdata["Height"].max()
```

```
student_with_max_height = df_Sdata[df_Sdata["Height"] == max_height]
```

```
# Output results
```

```
print("1. Total number of students:", total_students)
```

```
print("2. Number of girls:", girls_count)
```

```
print("3. Number of boys:", boys_count)
```

```
print("4. Maximum height:", max_height)
print("Student with max height:", student_with_max_height)
```

1. Total number of students: 71
 2. Number of girls: 0
 3. Number of boys: 0
 4. Maximum height: 180.0

Student with max height:	S.No	Name	Gender
Unnamed: 3	Height \		
67	68	KURMAI TRIBHUVAN KUMAR REDDY	male
			NaN
			180.0

	favourete	color	Cell phone brand	Laptop(Y/N)
67		black	OnePlus	N

Consider the df_Sdata data frame and do the following

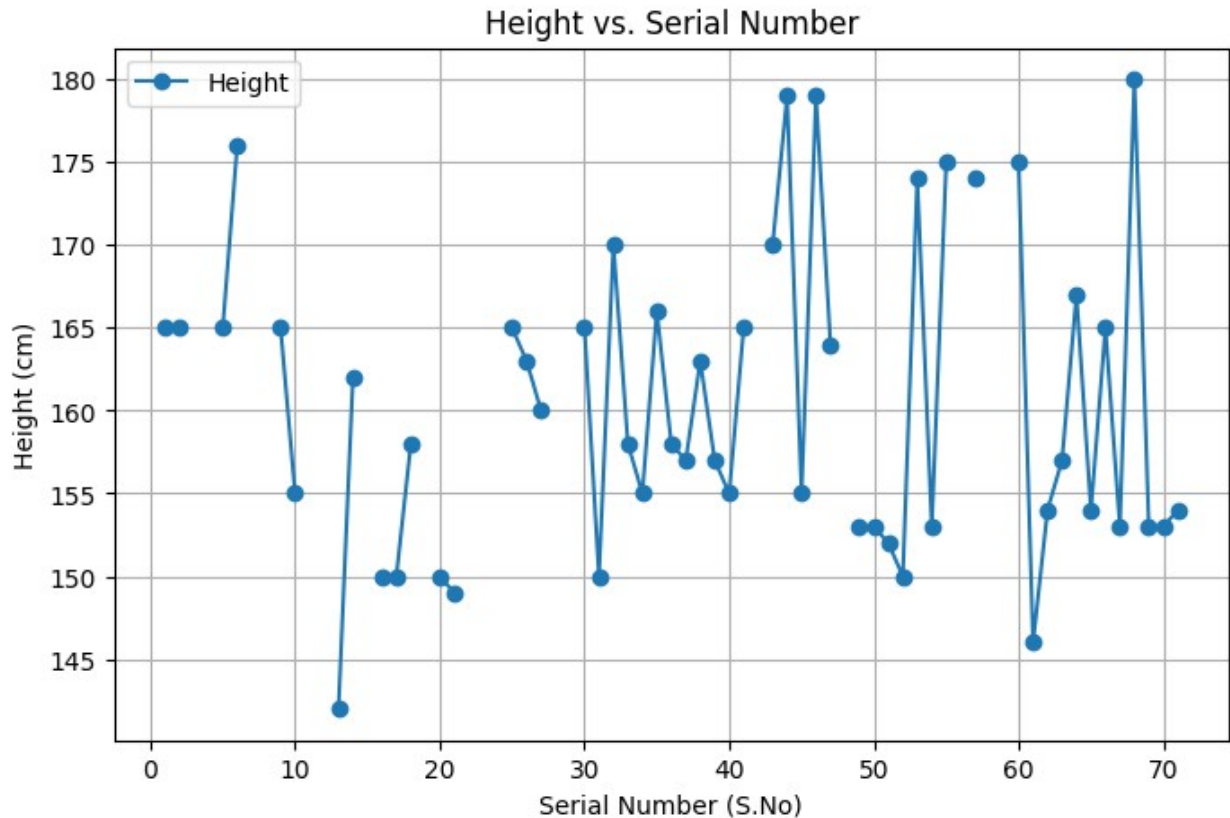
1. Plot ** height(y) Vs Sr.No (x)**

```
import matplotlib.pyplot as plt
import pandas as pd

# Specify the file path
file_path = r"H:\My Drive\Data Analytics Course\Source Files\Data-
analytic-course-assignment-sheet1.csv"

# Read the CSV file
df_Sdata = pd.read_csv(file_path)

# Plot height vs. serial number
plt.figure(figsize=(8, 5))
plt.plot(df_Sdata["S.No"], df_Sdata["Height"], marker="o",
label="Height")
plt.xlabel("Serial Number (S.No)")
plt.ylabel("Height (cm)")
plt.title("Height vs. Serial Number")
plt.grid()
plt.legend()
plt.show()
```



TASK #6

1. Convert the gender column of the df_Sdata into upper case
2. Find the Body Mass Index

```
import pandas as pd

#Specify the file path
file = r"H:\My Drive\Data Analytics Course\Source Files\Data-analytic-
course-assignment-sheet1.csv"

#Read CSV File
df_Sdata = pd.read_csv(file)

#Transform Gender column to uppercase
df_Sdata['Gender'] = df_Sdata['Gender'].str.upper()
print(df_Sdata)
```

	S.No	Name	Gender	Unnamed: 3	Height
0	1	A CHARAN	MALE	NaN	165.0
1	2	A LOLITHA SREE	FEMALE	NaN	165.0
2	3	A ROHINI	NaN	NaN	NaN

3	4	AKKAYYAGARI VENKATESWARLU	NaN	NaN	NaN
4	5	AYODHYAPURAM THARUN KUMAR REDDY	MALE	NaN	165.0
..
66	67	KOTAKONDA HARIKA	FEMALE	NaN	153.0
67	68	KURMAI TRIBHUVAN KUMAR REDDY	MALE	NaN	180.0
68	69	MEKALA HARSHITHA	FEMALE	NaN	153.0
69	70	MUMMADI AMRUTHA	FEMALE	NaN	153.0
70	71	NARA DIVYABHARATHI	FEMALE	NaN	154.0

	favourete	color	Cell phone brand	Laptop(Y/N)
0		Blue	Redmi	Y
1		Blue	Vivo	N
2		NaN	NaN	NaN
3		NaN	NaN	NaN
4		black	realme	N
..	
66		Black	samsung	N
67		black	OnePlus	N
68		Black	Redmi	N
69		white	Vivo	N
70		Black	VIVO	N

[71 rows x 8 columns]

```
import pandas as pd

file_path = r"H:\My Drive\Data Analytics Course\Source Files\BMI-
calculation-assignment.xlsx"

df_Sdata = pd.read_excel(file_path)

df_Sdata['BMI'] = df_Sdata['Weight'] / (df_Sdata['Height'] ** 2)

print(df_Sdata)
```

TASK #7

Consider the df_Edata data frame

1. Find the Average salary of the df_Edata data frame
2. Find the Average salary of the ST_CLREK
3. If ST_CLREK getting 3% commission of the salary, then find the Total.Income by creating the new column in the existing data frame

```

import pandas as pd

file_path = r"H:\My Drive\Data Analytics Course\Source Files\
employees.csv"

df_Edata = pd.read_csv(file_path)

# 1. Find the Average Salary
avg_salary = df_Edata['SALARY'].mean()
print("Average Salary of an Employee is : ",avg_salary)

# 2. Find the Average Salary of ST_CLERK
st_clerk_avg_salary = df_Edata[df_Edata['JOB_ID'] == 'ST_CLERK']
['SALARY'].mean()
print("Average Salary for ST CLERK is : ",st_clerk_avg_salary)

# 3. Add Total Income Column (3% commission) for ST_CLERK in a new
DataFrame
df_new = df_Edata.copy()
df_new['Total_Income'] = df_new.apply(lambda row: row['SALARY'] * 1.03
if row['JOB_ID'] == 'ST_CLERK' else row['SALARY'], axis=1)
print("New Data Set is :")
print()
print(df_new)

```

Average Salary of an Employee is : 6182.32
 Average Salary for ST CLERK is : 2750.0
 New Data Set is :

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER
HIRE_DATE \				
0	198	Donald	OConnell	DOCONNEL 650.507.9833 21-
1	199	Douglas	Grant	DGRANT 650.507.9844 13-
2	200	Jennifer	Whalen	JWHALEN 515.123.4444 17-
3	201	Michael	Hartstein	MHARTSTE 515.123.5555 17-
4	202	Pat	Fay	PFAY 603.123.6666 17-
5	203	Susan	Mavris	SMAVRIS 515.123.7777 07-
6	204	Hermann	Baer	HBAER 515.123.8888 07-
7	205	Shelley	Higgins	SHIGGINS 515.123.8080 07-
8	206	William	Gietz	WGIETZ 515.123.8181 07-

9 Jun-03	100	Steven	King	SKING	515.123.4567	17-
10 Sep-05	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-
11 Jan-01	102	Lex	De Haan	LDEHAAN	515.123.4569	13-
12 Jan-06	103	Alexander	Hunold	AHUNOLD	590.423.4567	03-
13 May-07	104	Bruce	Ernst	BERNST	590.423.4568	21-
14 Jun-05	105	David	Austin	DAUSTIN	590.423.4569	25-
15 Feb-06	106	Valli	Pataballa	VPATABAL	590.423.4560	05-
16 Feb-07	107	Diana	Lorentz	DLORENTZ	590.423.5567	07-
17 Aug-02	108	Nancy	Greenberg	NGREENBE	515.124.4569	17-
18 Aug-02	109	Daniel	Faviet	DFAVIET	515.124.4169	16-
19 Sep-05	110	John	Chen	JCHEN	515.124.4269	28-
20 Sep-05	111	Ismael	Sciarra	ISCIARRA	515.124.4369	30-
21 Mar-06	112	Jose Manuel	Urman	JMURMAN	515.124.4469	07-
22 Dec-07	113	Luis	Popp	LPOPP	515.124.4567	07-
23 Dec-02	114	Den	Raphaely	DRAPHEAL	515.127.4561	07-
24 May-03	115	Alexander	Khoo	AKH00	515.127.4562	18-
25 Dec-05	116	Shelli	Baida	SBAIDA	515.127.4563	24-
26 Jul-05	117	Sigal	Tobias	STOBIAS	515.127.4564	24-
27 Nov-06	118	Guy	Himuro	GHIMURO	515.127.4565	15-
28 Aug-07	119	Karen	Colmenares	KCOLMENA	515.127.4566	10-
29 Jul-04	120	Matthew	Weiss	MWEISS	650.123.1234	18-
30 Apr-05	121	Adam	Fripp	AFRIPP	650.123.2234	10-
31 May-03	122	Payam	Kaufling	PKAUFLIN	650.123.3234	01-
32 Oct-05	123	Shanta	Vollman	SVOLLMAN	650.123.4234	10-
33	124	Kevin	Mourgos	KMOURGOS	650.123.5234	16-

Nov-07						
34	125	Julia	Nayer	JNAYER	650.124.1214	16-
Jul-05						
35	126	Irene	Mikkilineni	IMIKKILI	650.124.1224	28-
Sep-06						
36	127	James	Landry	JLANDRY	650.124.1334	14-
Jan-07						
37	128	Steven	Markle	SMARKLE	650.124.1434	08-
Mar-08						
38	129	Laura	Bissot	LBISSOT	650.124.5234	20-
Aug-05						
39	130	Mozhe	Atkinson	MATKINS0	650.124.6234	30-
Oct-05						
40	131	James	Marlow	JAMRLOW	650.124.7234	16-
Feb-05						
41	132	TJ	Olson	TJOLSON	650.124.8234	10-
Apr-07						
42	133	Jason	Mallin	JMALLIN	650.127.1934	14-
Jun-04						
43	134	Michael	Rogers	MROGERS	650.127.1834	26-
Aug-06						
44	135	Ki	Gee	KGEE	650.127.1734	12-
Dec-07						
45	136	Hazel	Philtanker	HPHILTAN	650.127.1634	06-
Feb-08						
46	137	Renske	Ladwig	RLADWIG	650.121.1234	14-
Jul-03						
47	138	Stephen	Stiles	SSTILES	650.121.2034	26-
Oct-05						
48	139	John	Seo	JSE0	650.121.2019	12-
Feb-06						
49	140	Joshua	Patel	JPATEL	650.121.1834	06-
Apr-06						

	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
Total_Income					
0	SH_CLERK	2600	-	124	50
2600					
1	SH_CLERK	2600	-	124	50
2600					
2	AD_ASST	4400	-	101	10
4400					
3	MK_MAN	13000	-	100	20
13000					
4	MK_REP	6000	-	201	20
6000					
5	HR_REP	6500	-	101	40
6500					
6	PR_REP	10000	-	101	70

10000					
7	AC_MGR	12008	-	101	110
12008					
8	AC_ACCOUNT	8300	-	205	110
8300					
9	AD_PRES	24000	-	-	90
24000					
10	AD_VP	17000	-	100	90
17000					
11	AD_VP	17000	-	100	90
17000					
12	IT_PROG	9000	-	102	60
9000					
13	IT_PROG	6000	-	103	60
6000					
14	IT_PROG	4800	-	103	60
4800					
15	IT_PROG	4800	-	103	60
4800					
16	IT_PROG	4200	-	103	60
4200					
17	FI_MGR	12008	-	101	100
12008					
18	FI_ACCOUNT	9000	-	108	100
9000					
19	FI_ACCOUNT	8200	-	108	100
8200					
20	FI_ACCOUNT	7700	-	108	100
7700					
21	FI_ACCOUNT	7800	-	108	100
7800					
22	FI_ACCOUNT	6900	-	108	100
6900					
23	PU_MAN	11000	-	100	30
11000					
24	PU_CLERK	3100	-	114	30
3100					
25	PU_CLERK	2900	-	114	30
2900					
26	PU_CLERK	2800	-	114	30
2800					
27	PU_CLERK	2600	-	114	30
2600					
28	PU_CLERK	2500	-	114	30
2500					
29	ST_MAN	8000	-	100	50
8000					
30	ST_MAN	8200	-	100	50
8200					

31	ST_MAN	7900	-	100	50
7900					
32	ST_MAN	6500	-	100	50
6500					
33	ST_MAN	5800	-	100	50
5800					
34	ST_CLERK	3200	-	120	50
3200					
35	ST_CLERK	2700	-	120	50
2700					
36	ST_CLERK	2400	-	120	50
2400					
37	ST_CLERK	2200	-	120	50
2200					
38	ST_CLERK	3300	-	121	50
3300					
39	ST_CLERK	2800	-	121	50
2800					
40	ST_CLERK	2500	-	121	50
2500					
41	ST_CLERK	2100	-	121	50
2100					
42	ST_CLERK	3300	-	122	50
3300					
43	ST_CLERK	2900	-	122	50
2900					
44	ST_CLERK	2400	-	122	50
2400					
45	ST_CLERK	2200	-	122	50
2200					
46	ST_CLERK	3600	-	123	50
3600					
47	ST_CLERK	3200	-	123	50
3200					
48	ST_CLERK	2700	-	123	50
2700					
49	ST_CLERK	2500	-	123	50
2500					

TASK #8

Using Employee data frame set to find the following:

1. Find the how many different JOB_ID and DEPARTMENT_ID available in the df_Edata data frame? you may use .value_counts() method
2. Find the highest salary of the df_Edata data frame?
3. Plot the pie chart for DEPARTMENT_ID
4. Plot the Salary in Y Vs EMPLOYEE_ID in x

```

import pandas as pd
import matplotlib.pyplot as plt

# Load the dataset
file_path = r"H:\My Drive\Data Analytics Course\Source Files\
employees.csv"
df_Edata = pd.read_csv(file_path)

# 1. Count unique JOB_ID and DEPARTMENT_ID
job_id_counts = df_Edata['JOB_ID'].value_counts()
department_id_counts = df_Edata['DEPARTMENT_ID'].value_counts()

# Display the counts
print("Job ID Counts:")
print(job_id_counts)
print("\nDepartment ID Counts:")
print(department_id_counts)

# 2. Find the highest salary
highest_salary = df_Edata['SALARY'].max()
print("\nHighest Salary:", highest_salary)

# 3. Plot pie chart for DEPARTMENT_ID
plt.figure(figsize=(8, 8))
department_id_counts.plot.pie(autopct='%1.1f%%', startangle=90,
cmap="viridis", labels=department_id_counts.index)
plt.title('Distribution of DEPARTMENT_ID')
plt.ylabel('') # Hide the y-label for aesthetics
plt.show()

# 4. Plot Salary (Y) vs. EMPLOYEE_ID (X)
plt.figure(figsize=(10, 6))
plt.plot(df_Edata['EMPLOYEE_ID'], df_Edata['SALARY'], marker='o',
linestyle='-', color='b')
plt.title('Salary vs. Employee ID')
plt.xlabel('Employee ID')
plt.ylabel('Salary')
plt.grid(True)
plt.show()

```

Job ID Counts:

JOB_ID	Count
ST_CLERK	16
ST_MAN	5
FI_ACCOUNT	5
IT_PROG	5
PU_CLERK	5
SH_CLERK	2
AD_VP	2
AD_ASST	1

MK_MAN	1
AD_PRES	1
AC_ACCOUNT	1
AC_MGR	1
PR_REP	1
HR_REP	1
MK_REP	1
FI_MGR	1
PU_MAN	1

Name: count, dtype: int64

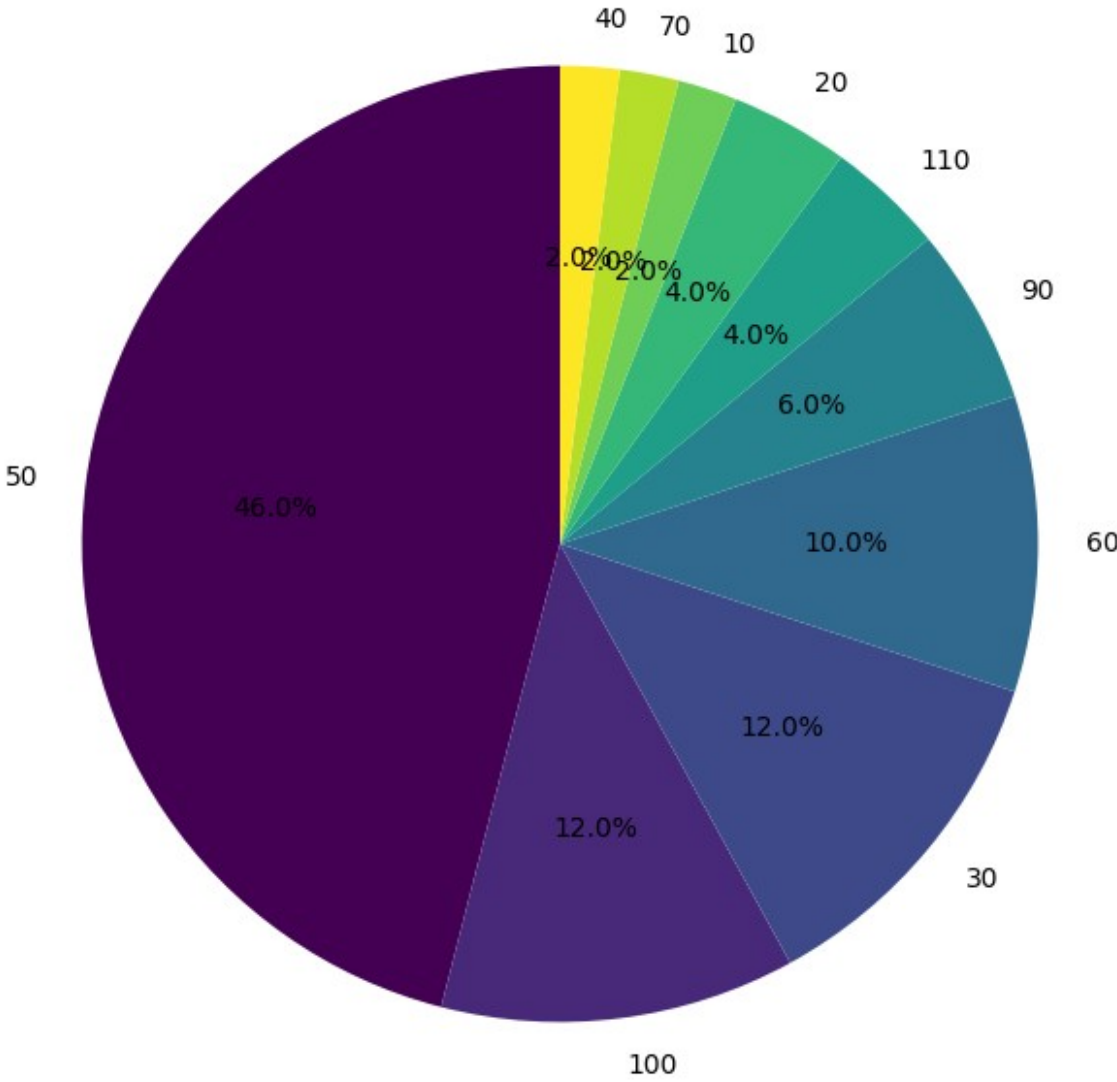
Department ID Counts:

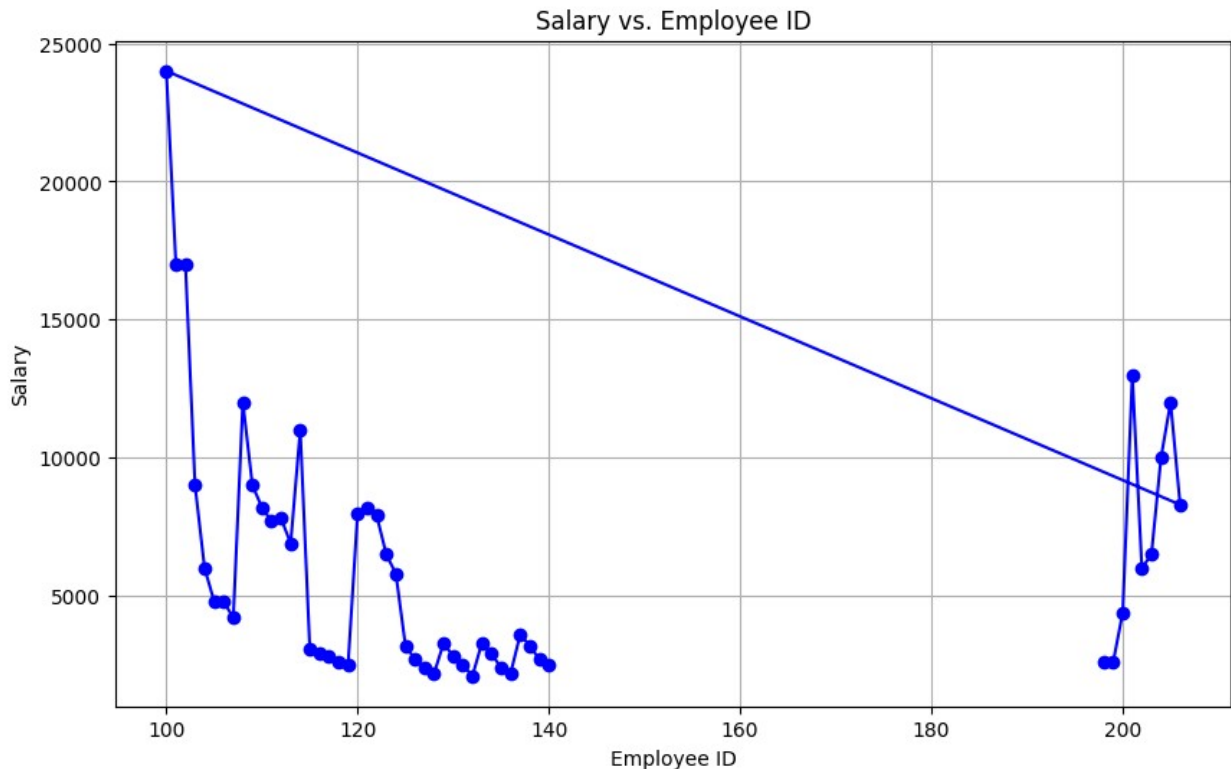
DEPARTMENT_ID	
50	23
100	6
30	6
60	5
90	3
110	2
20	2
10	1
70	1
40	1

Name: count, dtype: int64

Highest Salary: 24000

Distribution of DEPARTMENT_ID





TASK #9

1. Define What is a Pandas Series

A Pandas Series is a one-dimensional labeled array capable of holding data of any type (integer, string, float, etc.). It is similar to a column in a table or a 1D array. Each element in a Pandas Series has an associated index, allowing easy access to its elements.

1. Create a simple pandas series using `pd.Series()`

2. Key Difference Between Pandas DataFrame and Pandas Series

Pandas Series: A one-dimensional structure, similar to a single column or a 1D array.

Pandas DataFrame: A two-dimensional structure, analogous to a table, consisting of rows and columns, where each column is a Pandas Series.

3. How `df.iloc()` and `df.loc()` used to access the data frame elements and give example

#2 Creating a simple Pandas Series

```
import pandas as pd
```

```
#Assign data values
```

```
data = [10, 20, 30, 40]
```

```
#Create series using pd.series() function
```

```
series = pd.Series(data, index=["A", "B", "C", "D"])
```

```
#print the series
```

```
print(series)
```

```
A    10
```

```
B    20
```

```
C    30
```

```
D    40
```

```
dtype: int64
```

```
#4 Difference between df.iloc() and df.loc()
```

```
import pandas as pd
```

```
# Creating a DataFrame
```

```
data = {
```

```
    "Name": ["Alice", "Bob", "Charlie"],
```

```
    "Age": [25, 30, 35],
```

```
    "City": ["New York", "Los Angeles", "Chicago"]
```

```
}
```

```
df = pd.DataFrame(data, index=["R1", "R2", "R3"])
```

```
# Using .iloc[] to access data by position
```

```
print("Using iloc:")
```

```
print(df.iloc[1]) # Access the second row by index position
```

```
print(df.iloc[:, 1]) # Access the second column (Age)
```

```
# Using .loc[] to access data by labels
```

```
print("\nUsing loc:")
```

```
print(df.loc["R2"]) # Access the row with label 'R2'
```

```
print(df.loc[:, "City"]) # Access the 'City' column
```

```
Using iloc:
```

```
Name          Bob
```

```
Age           30
```

```
City    Los Angeles
```

```
Name: R2, dtype: object
```

```
R1    25
```

```
R2    30
```

```
R3    35
```

```
Name: Age, dtype: int64
```

```
Using loc:
```

```
Name          Bob
```

```
Age           30
```

```
City    Los Angeles
```

```
Name: R2, dtype: object
```

```
R1    New York
```

```
R2    Los Angeles
```

```
R3    Chicago
```

```
Name: City, dtype: object
```