Pandas: Use for data manipulation, Data analysis, series and Dataframe

Pandas

```
SeriesL 1.index,1.column
Dataframe: combination of multiple series
```

In [1]:
```
import pandas as pd
```

In [2]:
```
import numpy as np
```

In [3]:
```
l=["lavesha","vaishanavi","gayatri","priyanka","ashwini"]
df=pd.Series(data=l)
print(df)
df1=pd.Series(data=l,index=["a","b","c","d","e"])
print(df1)

print(df1["c"])
```

```
0       lavesha
1    vaishanavi
2       gayatri
3      priyanka
4       ashwini
dtype: object
a       lavesha
b    vaishanavi
c       gayatri
d      priyanka
e       ashwini
dtype: object
gayatri
```

In [4]:
```
print(df[4])
```

```
ashwini
```

In [5]:
```
df3=pd.Series(data=l,index=["a","a","a","b","b"])
print(df3)
print(df3["a"])  #here duplicate indexing is also posssible
```

```
a       lavesha
a    vaishanavi
a       gayatri
b      priyanka
b       ashwini
dtype: object
a       lavesha
a    vaishanavi
a       gayatri
dtype: object
```

In [6]:
```
df[1:4]
```

Out[6]:
```
1    vaishanavi
2       gayatri
3      priyanka
dtype: object
```

dataframe

In [7]:
```
df=pd.DataFrame(data={"name":["shreyas","abhijeet","abhishek","tejas","atul
                      "age":[23,22,24,23,24,25,22],
                      "marks1":[89,67,78,89,90,88,78],
                      "address":["pune","mumbai","pune","akola","sambhajina
print(df)
```

```
       name  age  marks1        address
0   shreyas   23      89           pune
1  abhijeet   22      67         mumbai
2  abhishek   24      78           pune
3     tejas   23      89          akola
4      atul   24      90  sambhajinagar
5    jayesh   25      88        jalgaon
6    athang   22      78        jalgaon
```

In [8]:
```
df
```

Out[8]:

	name	age	marks1	address
0	shreyas	23	89	pune
1	abhijeet	22	67	mumbai
2	abhishek	24	78	pune
3	tejas	23	89	akola
4	atul	24	90	sambhajinagar
5	jayesh	25	88	jalgaon
6	athang	22	78	jalgaon

In [9]:
```
df["name"]
```

Out[9]:
```
0     shreyas
1    abhijeet
2    abhishek
3       tejas
4        atul
5      jayesh
6      athang
Name: name, dtype: object
```

In [10]:
```
df["name"][2]
```

Out[10]:
```
'abhishek'
```

In [13]:
```
df["marks2"]=[89,78,88,90,78,89,89]
```

In [14]:
```
df
```

Out[14]:

	name	age	marks1	address	marks2
0	shreyas	23	89	pune	89
1	abhijeet	22	67	mumbai	78
2	abhishek	24	78	pune	88
3	tejas	23	89	akola	90
4	atul	24	90	sambhajinagar	78
5	jayesh	25	88	jalgaon	89
6	athang	22	78	jalgaon	89

In [15]:
```
df["total"]=df["marks1"]+df["marks2"]
```

In [16]:
```
df
```

Out[16]:

	name	age	marks1	address	marks2	total
0	shreyas	23	89	pune	89	178
1	abhijeet	22	67	mumbai	78	145
2	abhishek	24	78	pune	88	166
3	tejas	23	89	akola	90	179
4	atul	24	90	sambhajinagar	78	168
5	jayesh	25	88	jalgaon	89	177
6	athang	22	78	jalgaon	89	167

In [22]:
```
df["percentage"]=df["total"]/2
```

In [23]:
```
df
```

Out[23]:

	name	age	marks1	address	marks2	total	percentage
0	shreyas	23	89	pune	89	178	89.0
1	abhijeet	22	67	mumbai	78	145	72.5
2	abhishek	24	78	pune	88	166	83.0
3	tejas	23	89	akola	90	179	89.5
4	atul	24	90	sambhajinagar	78	168	84.0
5	jayesh	25	88	jalgaon	89	177	88.5
6	athang	22	78	jalgaon	89	167	83.5

In [24]:
```
df[["name","age"]]
```

Out[24]:

	name	age
0	shreyas	23
1	abhijeet	22
2	abhishek	24
3	tejas	23
4	atul	24
5	jayesh	25
6	athang	22

In [25]:
```
df.drop(["percentage"],axis=1)
```

Out[25]:

	name	age	marks1	address	marks2	total
0	shreyas	23	89	pune	89	178
1	abhijeet	22	67	mumbai	78	145
2	abhishek	24	78	pune	88	166
3	tejas	23	89	akola	90	179
4	atul	24	90	sambhajinagar	78	168
5	jayesh	25	88	jalgaon	89	177
6	athang	22	78	jalgaon	89	167

In [29]:
```
df
```

Out[29]:

	name	age	marks1	address	marks2	total	percentage
0	shreyas	23	89	pune	89	178	89.0
1	abhijeet	22	67	mumbai	78	145	72.5
2	abhishek	24	78	pune	88	166	83.0
3	tejas	23	89	akola	90	179	89.5
4	atul	24	90	sambhajinagar	78	168	84.0
5	jayesh	25	88	jalgaon	89	177	88.5
6	athang	22	78	jalgaon	89	167	83.5

In [32]:
```
df1=df.drop(["percentage"],axis=1)
print(df1)
```

```
       name  age  marks1        address  marks2  total
0   shreyas   23      89           pune      89    178
1  abhijeet   22      67         mumbai      78    145
2  abhishek   24      78           pune      88    166
3     tejas   23      89          akola      90    179
4      atul   24      90  sambhajinagar      78    168
5    jayesh   25      88        jalgaon      89    177
6    athang   22      78        jalgaon      89    167
```

In [52]:
```
#df.drop(["percentage"],axis=1,inplace=True)
```

In [35]:
```
df
```

Out[35]:

	name	age	marks1	address	marks2	total
0	shreyas	23	89	pune	89	178
1	abhijeet	22	67	mumbai	78	145
2	abhishek	24	78	pune	88	166
3	tejas	23	89	akola	90	179
4	atul	24	90	sambhajinagar	78	168
5	jayesh	25	88	jalgaon	89	177
6	athang	22	78	jalgaon	89	167

In [ ]:
```

```

In [37]:
```
df.iloc[0:3]
```

Out[37]:

	name	age	marks1	address	marks2	total
0	shreyas	23	89	pune	89	178
1	abhijeet	22	67	mumbai	78	145
2	abhishek	24	78	pune	88	166

In [38]:
```
df[0:3]
```

Out[38]:

	name	age	marks1	address	marks2	total
0	shreyas	23	89	pune	89	178
1	abhijeet	22	67	mumbai	78	145
2	abhishek	24	78	pune	88	166

In [40]:
```
print(df.iloc[4:])
print(df[4:])
```

```
     name  age  marks1        address  marks2  total
4    atul   24      90  sambhajinagar      78    168
5  jayesh   25      88        jalgaon      89    177
6  athang   22      78        jalgaon      89    167
     name  age  marks1        address  marks2  total
4    atul   24      90  sambhajinagar      78    168
5  jayesh   25      88        jalgaon      89    177
6  athang   22      78        jalgaon      89    167
```

In [41]:
```
df.iloc[4:,0:2]
```

Out[41]:

	name	age
4	atul	24
5	jayesh	25
6	athang	22

In [43]:
```
df.iloc[1:4,3:]
```

Out[43]:

	address	marks2	total
1	mumbai	78	145
2	pune	88	166
3	akola	90	179

In [46]:
```
df.loc[df["address"]=="pune"]
```

Out[46]:

	name	age	marks1	address	marks2	total
0	shreyas	23	89	pune	89	178
2	abhishek	24	78	pune	88	166

In [47]:
```
df.loc[(df["address"]=="pune")&(df["age"]>23)]
```

Out[47]:

	name	age	marks1	address	marks2	total
2	abhishek	24	78	pune	88	166

In [48]:
```
df.loc[1:4,["name","total"]]
```

Out[48]:

	name	total
1	abhijeet	145
2	abhishek	166
3	tejas	179
4	atul	168

In [50]:
```
df2=df.drop([5],axis=0)
```

In [51]:
```
df2
```

Out[51]:

	name	age	marks1	address	marks2	total
0	shreyas	23	89	pune	89	178
1	abhijeet	22	67	mumbai	78	145
2	abhishek	24	78	pune	88	166
3	tejas	23	89	akola	90	179
4	atul	24	90	sambhajinagar	78	168
6	athang	22	78	jalgaon	89	167

In [56]:
```
df=pd.DataFrame(data={"id":[1,2,3,4,5,6,7],"product":["a","b","c","d","e","
                      "quantity":[2,3,10,12,14,20,6],"cp":[10,5,20,13,14,15
                     "sp":[12,6,22,14,16,20,23]})
```

In [57]:
```
df
```

Out[57]:

	id	product	quantity	cp	sp
0	1	a	2	10	12
1	2	b	3	5	6
2	3	c	10	20	22
3	4	d	12	13	14
4	5	e	14	14	16
5	6	f	20	15	20
6	7	g	6	16	23

In [ ]:
```
totalcp=quantity*cp
total sp=quantity*sp
profit/loss=cp<sp==profit
```

In [70]:
```
df["Total cp"]=df["quantity"]*df["cp"]
df["Total sp"]=df["quantity"]*df["sp"]
df["profit/loss"]= df["Total sp"]-df["Total cp"]
```

In [71]:
```
df
```

Out[71]:

	id	product	quantity	cp	sp	Total sp	Total cp	profit/loss
0	1	a	2	10	12	24	20	4
1	2	b	3	5	6	18	15	3
2	3	c	10	20	22	220	200	20
3	4	d	12	13	14	168	156	12
4	5	e	14	14	16	224	196	28
5	6	f	20	15	20	400	300	100
6	7	g	6	16	23	138	96	42

In [80]:
```
df[["id","product"]]
```

Out[80]:

	id	product
0	1	a
1	2	b
2	3	c
3	4	d
4	5	e
5	6	f
6	7	g

In [82]:
```
df.loc[2:5]
```

Out[82]:

	id	product	quantity	cp	sp	Total sp	Total cp	profit/loss
2	3	c	10	20	22	220	200	20
3	4	d	12	13	14	168	156	12
4	5	e	14	14	16	224	196	28
5	6	f	20	15	20	400	300	100

In [118]:
```
df.loc[(df["id"]<5)|(df["quantity"]>10)]
#df.loc[(df["address"]=="pune")&(df["age"]>23)]
#df.loc[1:4,["name","total"]]
#df.loc[df["address"]=="pune"]
#df.iloc[1:4,3:]
```

Out[118]:

	id	product	quantity	cp	sp	Total sp	Total cp	profit/loss
0	1	a	2	10	12	24	20	4
1	2	b	3	5	6	18	15	3
2	3	c	10	20	22	220	200	20
3	4	d	12	13	14	168	156	12
4	5	e	14	14	16	224	196	28
5	6	f	20	15	20	400	300	100

In [115]:
```
df[["id","product"]]
```

Out[115]:

	id	product
0	1	a
1	2	b
2	3	c
3	4	d
4	5	e
5	6	f
6	7	g

In [120]:
```
df[((df["product"]=="a")|(df["product"]=="b"))&(df["quantity"]>3)]
```

Out[120]:

id	product	quantity	cp	sp	Total sp	Total cp	profit/loss

In [127]:
```
df2=df.drop([4,5,6],axis=0)
#df2=df.drop([5],axis=0)
```

```
axis =1 for column
axis=0 for rows
```

In [128]:
```
df2
```

Out[128]:

	id	product	quantity	cp	sp	Total sp	Total cp	profit/loss
0	1	a	2	10	12	24	20	4
1	2	b	3	5	6	18	15	3
2	3	c	10	20	22	220	200	20
3	4	d	12	13	14	168	156	12

In [136]:
```
df3=df2.drop(["cp","sp"],axis=1)
```

In [137]:
```
df3
```

Out[137]:

	id	product	quantity	Total sp	Total cp	profit/loss
0	1	a	2	24	20	4
1	2	b	3	18	15	3
2	3	c	10	220	200	20
3	4	d	12	168	156	12

In [138]:
```
df
```

Out[138]:

	id	product	quantity	cp	sp	Total sp	Total cp	profit/loss
0	1	a	2	10	12	24	20	4
1	2	b	3	5	6	18	15	3
2	3	c	10	20	22	220	200	20
3	4	d	12	13	14	168	156	12
4	5	e	14	14	16	224	196	28
5	6	f	20	15	20	400	300	100
6	7	g	6	16	23	138	96	42

In [152]:
```
df.loc[3:,["quantity","cp"]]
```

Out[152]:

	quantity	cp
3	12	13
4	14	14
5	20	15
6	6	16

In [155]:
```
df3=df.drop([6],axis=0)
```

In [156]:
```
print(df3)
```

```
   id product  quantity  cp  sp  Total sp  Total cp  profit/loss
0   1       a         2  10  12        24        20            4
1   2       b         3   5   6        18        15            3
2   3       c        10  20  22       220       200           20
3   4       d        12  13  14       168       156           12
4   5       e        14  14  16       224       196           28
5   6       f        20  15  20       400       300          100
```

In [161]:
```
df3=df.drop(["profit/loss"],axis=1)
```

In [162]:
```
df3
```

Out[162]:

	id	product	quantity	cp	sp	Total sp	Total cp
0	1	a	2	10	12	24	20
1	2	b	3	5	6	18	15
2	3	c	10	20	22	220	200
3	4	d	12	13	14	168	156
4	5	e	14	14	16	224	196
5	6	f	20	15	20	400	300
6	7	g	6	16	23	138	96

In [164]:
```
df.loc[(df["id"]>3)&(df["quantity"]>10)]
```

Out[164]:

	id	product	quantity	cp	sp	Total sp	Total cp	profit/loss
3	4	d	12	13	14	168	156	12
4	5	e	14	14	16	224	196	28
5	6	f	20	15	20	400	300	100

In [168]:
```
df3=df2.set_index(["id"])
```

In [169]:
```
df3
```

Out[169]:

id	product	quantity	cp	sp	Total sp	Total cp	profit/loss
1	a	2	10	12	24	20	4
2	b	3	5	6	18	15	3
3	c	10	20	22	220	200	20
4	d	12	13	14	168	156	12

In [177]:
```
df5=pd.DataFrame(data={"name":[np.nan,"amit",np.nan,"dipa","pratap"],"age":
```

In [178]:
```
df5
```

Out[178]:

	name	age
0	NaN	45.0
1	amit	23.0
2	NaN	NaN
3	dipa	28.0
4	pratap	NaN

In [179]:
```
df5.isnull().sum()
```

Out[179]:
```
name    2
age     2
dtype: int64
```

In [186]:
```
data=pd.read_csv("C:/Users/rfpaw/Downloads/jupiter_employees.csv")
```

In [191]:
```
data
```

Out[191]:

	Id	Name	Age	Base_Pay	OverTime_Pay	Benefits	Total_Pay_benefits	Loan	De
0	910	Aaron Bennett	22	60000	30000	7500	97500	Yes	
1	679	Aaron Douglas	55	80000	40000	10000	130000	Yes	
2	225	Abigail Foster	22	40000	20000	5000	65000	Yes	
3	176	Adam Jensen	30	120000	60000	15000	195000	No	
4	705	Adam Johnson	22	120000	60000	15000	195000	Yes	
...	...	...	...	...	...	...	...	...	
995	405	William Shaw	22	96000	48000	12000	156000	No	
996	150	William	22	30000	15000	3750	48750	No	

In [196]:
```
data.head(10)
```

Out[196]:

	Id	Name	Age	Base_Pay	OverTime_Pay	Benefits	Total_Pay_benefits	Loan	Depa
0	910	Aaron Bennett	22	60000	30000	7500	97500	Yes	
1	679	Aaron Douglas	55	80000	40000	10000	130000	Yes	
2	225	Abigail Foster	22	40000	20000	5000	65000	Yes	
3	176	Adam Jensen	30	120000	60000	15000	195000	No	
4	705	Adam Johnson	22	120000	60000	15000	195000	Yes	
5	146	Adrian Pratt	37	120000	60000	15000	195000	No	
6	811	Adrian White	43	96000	48000	12000	156000	No	

In [197]:
```
data.head()
```

Out[197]:

	Id	Name	Age	Base_Pay	OverTime_Pay	Benefits	Total_Pay_benefits	Loan	Departm
0	910	Aaron Bennett	22	60000	30000	7500	97500	Yes	SD
1	679	Aaron Douglas	55	80000	40000	10000	130000	Yes	S
2	225	Abigail Foster	22	40000	20000	5000	65000	Yes	S
3	176	Adam Jensen	30	120000	60000	15000	195000	No	S
4	705	Adam Johnson	22	120000	60000	15000	195000	Yes	SD

In [198]:
```
data.head(2)
```

Out[198]:

	Id	Name	Age	Base_Pay	OverTime_Pay	Benefits	Total_Pay_benefits	Loan	Departm
0	910	Aaron Bennett	22	60000	30000	7500	97500	Yes	SD
1	679	Aaron Douglas	55	80000	40000	10000	130000	Yes	S

In [199]:
```
data.head(20)
```

Out[199]:

	Id	Name	Age	Base_Pay	OverTime_Pay	Benefits	Total_Pay_benefits	Loan	Depar
0	910	Aaron Bennett	22	60000	30000	7500	97500	Yes	
1	679	Aaron Douglas	55	80000	40000	10000	130000	Yes	
2	225	Abigail Foster	22	40000	20000	5000	65000	Yes	
3	176	Adam Jensen	30	120000	60000	15000	195000	No	
4	705	Adam Johnson	22	120000	60000	15000	195000	Yes	
5	146	Adrian Pratt	37	120000	60000	15000	195000	No	
6	811	Adrian White	43	96000	48000	12000	156000	No	
7	423	Aimee Wilson	32	110000	55000	13750	178750	Yes	
8	410	Alan Hardy	22	80000	40000	10000	130000	No	Lo
9	200	Alec Glenn	48	120000	60000	15000	195000	No	
10	837	Alejandro Pace	43	120000	60000	15000	195000	No	Ac
11	259	Alex Clayton	37	60000	30000	7500	97500	Yes	
12	177	Alexander Edwards	22	70000	35000	8750	113750	Yes	
13	616	Alexander Lutz	22	110000	55000	13750	178750	No	
14	569	Alexander Navarro	48	120000	60000	15000	195000	No	
15	190	Alexander Pope	27	80000	40000	10000	130000	Yes	Lo
16	926	Alexander Wiley	43	80000	40000	10000	130000	Yes	Lo
17	375	Alexandra Contreras MD	43	90000	45000	11250	146250	No	
18	700	Alexandra Noble	22	80000	40000	10000	130000	No	
19	741	Alexis Fleming	22	70000	35000	8750	113750	No	

In [200]:
```
data[6:11]
```

Out[200]:

	Id	Name	Age	Base_Pay	OverTime_Pay	Benefits	Total_Pay_benefits	Loan	Depar
6	811	Adrian White	43	96000	48000	12000	156000	No	
7	423	Aimee Wilson	32	110000	55000	13750	178750	Yes	
8	410	Alan Hardy	22	80000	40000	10000	130000	No	Lo
9	200	Alec Glenn	48	120000	60000	15000	195000	No	S
10	837	Alejandro Pace	43	120000	60000	15000	195000	No	Acc

In [201]:
```
data.iloc[6:11]
```

Out[201]:

	Id	Name	Age	Base_Pay	OverTime_Pay	Benefits	Total_Pay_benefits	Loan	Depar
6	811	Adrian White	43	96000	48000	12000	156000	No	
7	423	Aimee Wilson	32	110000	55000	13750	178750	Yes	
8	410	Alan Hardy	22	80000	40000	10000	130000	No	Lo
9	200	Alec Glenn	48	120000	60000	15000	195000	No	S
10	837	Alejandro Pace	43	120000	60000	15000	195000	No	Acc

In [204]:
```
data.iloc[100:121]
```

Out[204]:

	Id	Name	Age	Base_Pay	OverTime_Pay	Benefits	Total_Pay_benefits	Loan	Dep
100	889	Bill Smith	22	110000	55000	13750	178750	No	
101	25	Billy Moyer	22	120000	60000	15000	195000	No	
102	788	Billy Wilson	32	110000	55000	13750	178750	Yes	
103	416	Blake Choi	22	96000	48000	12000	156000	No	A
104	498	Blake Hudson	32	120000	60000	15000	195000	Yes	
105	959	Bob Castaneda	37	120000	60000	15000	195000	No	
106	821	Bonnie Garcia	22	110000	55000	13750	178750	Yes	
107	27	Bradley Olson	37	96000	48000	12000	156000	Yes	
108	830	Brandi Cordova	40	120000	60000	15000	195000	No	
109	455	Brandi Shields	50	120000	60000	15000	195000	Yes	
110	139	Brandon Brown	27	50000	25000	6250	81250	No	
111	346	Brandon Maxwell	50	120000	60000	15000	195000	No	
112	795	Brandon Navarro	40	80000	40000	10000	130000	Yes	
113	893	Brandon Simpson	32	110000	55000	13750	178750	No	
114	195	Brandy Anderson	22	70000	35000	8750	113750	No	
115	419	Brandy Jones	27	90000	45000	11250	146250	Yes	
116	755	Brandy Lambert	22	60000	30000	7500	97500	Yes	
117	678	Breanna Gutierrez	32	120000	60000	15000	195000	Yes	
118	787	Brenda Bell	27	40000	20000	5000	65000	No	
119	217	Brenda Friedman	27	120000	60000	15000	195000	No	
120	577	Brenda Howard	30	80000	40000	10000	130000	Yes	

In [210]:
```
data.loc[100:121,["Id","Name","Rating"]]
```

Out[210]:

	Id	Name	Rating
100	889	Bill Smith	8.2
101	25	Billy Moyer	8.9
102	788	Billy Wilson	9.3
103	416	Blake Choi	5.7
104	498	Blake Hudson	8.9
105	959	Bob Castaneda	7.4
106	821	Bonnie Garcia	3.3
107	27	Bradley Olson	8.9
108	830	Brandi Cordova	9.3
109	455	Brandi Shields	8.9
110	139	Brandon Brown	3.3
111	346	Brandon Maxwell	8.0
112	795	Brandon Navarro	5.0
113	893	Brandon Simpson	8.0
114	195	Brandy Anderson	5.0
115	419	Brandy Jones	8.0
116	755	Brandy Lambert	9.3
117	678	Breanna Gutierrez	7.4
118	787	Brenda Bell	9.3
119	217	Brenda Friedman	8.9
120	577	Brenda Howard	8.2
121	635	Brenda Martinez	8.0

In [212]:
```
data.loc[(data["Loan"]=="Yes")&(data["Department"]=="Admin")]
```

Out[212]:

	Id	Name	Age	Base_Pay	OverTime_Pay	Benefits	Total_Pay_benefits	Loan	Depa
47	115	Ana Rowland	22	120000	60000	15000	195000	Yes	
140	521	Brittany Sloan	22	110000	55000	13750	178750	Yes	
272	64	Derek Hahn	30	110000	55000	13750	178750	Yes	
309	901	Elizabeth Guzman	50	120000	60000	15000	195000	Yes	
401	508	Jasmine Mullen	22	60000	30000	7500	97500	Yes	
555	54	Kristen Cline	40	96000	48000	12000	156000	Yes	
559	988	Kristina Harris	22	30000	15000	3750	48750	Yes	
560	210	Kristina Wright	22	120000	60000	15000	195000	Yes	
688	94	Michael Wright	22	110000	55000	13750	178750	Yes	
746	913	Nicole Walter	22	120000	60000	15000	195000	Yes	
759	601	Patricia Gonzalez	32	80000	40000	10000	130000	Yes	
804	271	Richard House	40	40000	20000	5000	65000	Yes	
808	411	Rita James	22	80000	40000	10000	130000	Yes	
814	640	Robert Martin	22	70000	35000	8750	113750	Yes	
939	957	Theresa Jackson	37	96000	48000	12000	156000	Yes	
941	822	Thomas Meyer	27	120000	60000	15000	195000	Yes	
983	215	Walter James	32	120000	60000	15000	195000	Yes	

In [213]:
```
data.loc[data["Rating"]>9]
```

Out[213]:

	Id	Name	Age	Base_Pay	OverTime_Pay	Benefits	Total_Pay_benefits	Loan	Depa
1	679	Aaron Douglas	55	80000	40000	10000	130000	Yes	
4	705	Adam Johnson	22	120000	60000	15000	195000	Yes	
6	811	Adrian White	43	96000	48000	12000	156000	No	
8	410	Alan Hardy	22	80000	40000	10000	130000	No	L
10	837	Alejandro Pace	43	120000	60000	15000	195000	No	Ac
...	...	...	...	...	...	...	...	...	
991	23	William Lee	43	120000	60000	15000	195000	No	
993	928	William Ponce	37	120000	60000	15000	195000	No	
995	405	William Shaw	22	96000	48000	12000	156000	No	
998	178	Yvonne Baker	22	120000	60000	15000	195000	No	
999	8	Yvonne Johnson	22	120000	60000	15000	195000	No	

315 rows × 14 columns

In [214]:
```
data.tail()
```

Out[214]:

	Id	Name	Age	Base_Pay	OverTime_Pay	Benefits	Total_Pay_benefits	Loan	Depar
995	405	William Shaw	22	96000	48000	12000	156000	No	
996	150	William White	22	30000	15000	3750	48750	No	
997	482	Willie Walsh	22	120000	60000	15000	195000	No	
998	178	Yvonne Baker	22	120000	60000	15000	195000	No	
999	8	Yvonne Johnson	22	120000	60000	15000	195000	No	

In [215]:
```
data.tail(2)
```

Out[215]:

	Id	Name	Age	Base_Pay	OverTime_Pay	Benefits	Total_Pay_benefits	Loan	Depar
998	178	Yvonne Baker	22	120000	60000	15000	195000	No	
999	8	Yvonne Johnson	22	120000	60000	15000	195000	No	

In [217]:
```
data["Department"].unique()
```

Out[217]:
```
array(['SDE-T', 'SDE', 'Logistics', 'Accounts', 'Admin', 'Hr'],
      dtype=object)
```

In [218]:
```
data.columns
```

Out[218]:
```
Index(['Id', 'Name', 'Age', 'Base_Pay', 'OverTime_Pay', 'Benefits',
       'Total_Pay_benefits', 'Loan', 'Department', 'Duration', 'Rating',
       'Email', 'State', 'Country'],
      dtype='object')
```

In [219]:
```
data["Department"].value_counts()
```

Out[219]:
```
Department
SDE          497
SDE-T        175
Logistics    158
Accounts      65
Admin         53
Hr            52
Name: count, dtype: int64
```

In [221]:
```
data["Department"].value_counts(normalize=True)*100
```

Out[221]:
```
Department
SDE          49.7
SDE-T        17.5
Logistics    15.8
Accounts      6.5
Admin         5.3
Hr            5.2
Name: proportion, dtype: float64
```

In [ ]: