

```
# installing dependencies
```

```
import pandas as p
import seaborn as s
import numpy as n
```

```
data_Source = p.read_csv("/content/titanic.csv")
```

```
data_Source.head()
```

```
   PassengerId  Survived  Pclass \
0             892         0       3
1             893         1       3
2             894         0       2
3             895         0       3
4             896         1       3
```

```
   Parch \
0             Kelly, Mr. James      male  34.5      0
0
1             Wilkes, Mrs. James (Ellen Needs)  female  47.0      1
0
2             Myles, Mr. Thomas Francis      male  62.0      0
0
3             Wirz, Mr. Albert      male  27.0      0
0
4  Hirvonen, Mrs. Alexander (Helga E Lindqvist)  female  22.0      1
1
```

```
   Ticket    Fare  Cabin  Embarked
0  330911   7.8292   NaN      Q
1  363272   7.0000   NaN      S
2  240276   9.6875   NaN      Q
3  315154   8.6625   NaN      S
4  3101298  12.2875   NaN      S
```

```
data_Source.tail()
```

```
   PassengerId  Survived  Pclass
Sex \
413          1305         0       3      Spector, Mr. Woolf
male
414          1306         1       1  Oliva y Ocana, Dona. Fermina
female
415          1307         0       3  Saether, Mr. Simon Sivertsen
male
416          1308         0       3      Ware, Mr. Frederick
male
417          1309         0       3      Da vinci
female
```

	Age	SibSp	Parch		Ticket	Fare	Cabin	Embarked
413	NaN	0	0		A.5. 3236	8.05	NaN	S
414	39.0	0	0		PC 17758	108.90	C105	C
415	38.5	0	0	SOTON/0.Q.	3101262	7.25	NaN	S
416	NaN	0	0		359309	8.05	NaN	S
417	45.6	1	1		0	58.00	Nan	NaN

data_Source.columns

```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age',
      'SibSp',
      'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

data_Source.shape

(418, 12)

data_Source.describe()

	PassengerId	Survived	Pclass	Age	SibSp
count	418.000000	418.000000	418.000000	333.000000	418.000000
mean	1100.500000	0.363636	2.265550	30.318619	0.447368
std	120.810458	0.481622	0.841838	14.184726	0.896760
min	892.000000	0.000000	1.000000	0.170000	0.000000
25%	996.250000	0.000000	1.000000	21.000000	0.000000
50%	1100.500000	0.000000	3.000000	27.000000	0.000000
75%	1204.750000	1.000000	3.000000	39.000000	1.000000
max	1309.000000	1.000000	3.000000	76.000000	8.000000

	Parch	Fare
count	418.000000	417.000000
mean	0.392344	35.712660
std	0.981429	55.914486
min	0.000000	0.000000
25%	0.000000	7.895800
50%	0.000000	14.454200
75%	0.000000	31.500000
max	9.000000	512.329200

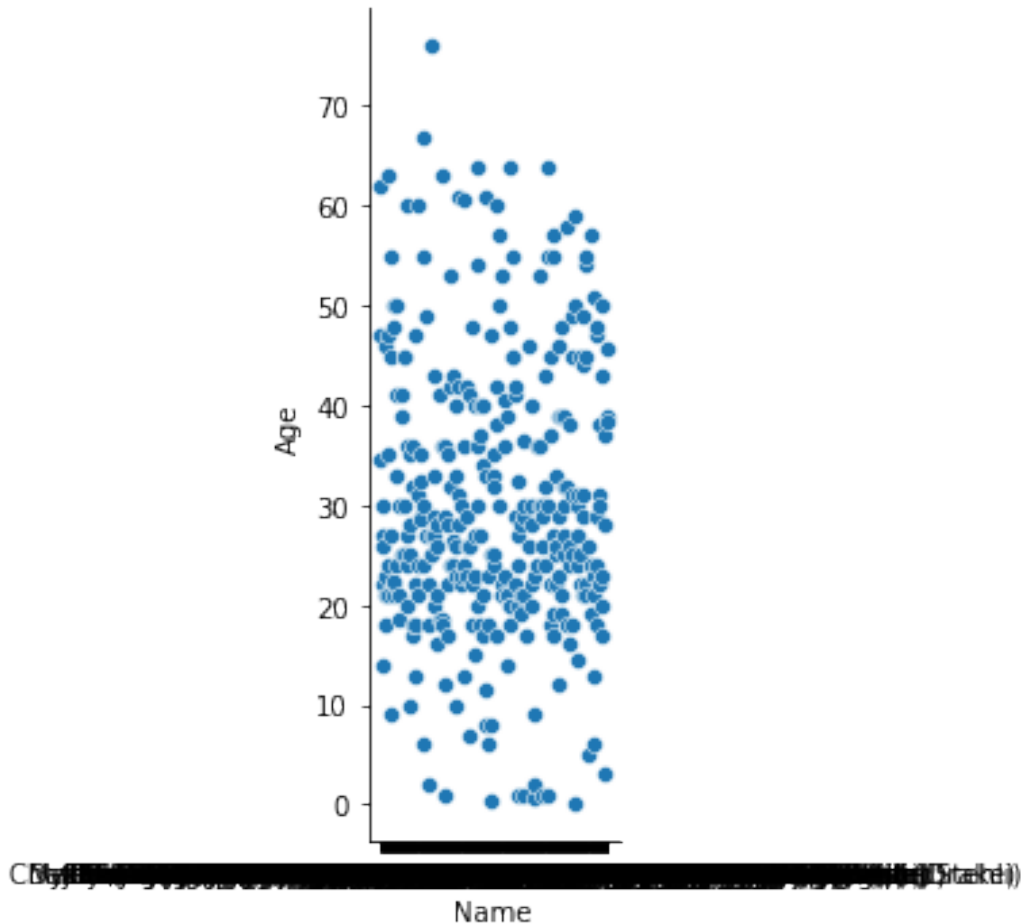
data_Source.isnull().sum()

PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	85
SibSp	0
Parch	0
Ticket	0
Fare	1

```
Cabin          326
Embarked       1
dtype: int64
```

```
s.relplot(x='Name', y='Age', data=data_Source)
```

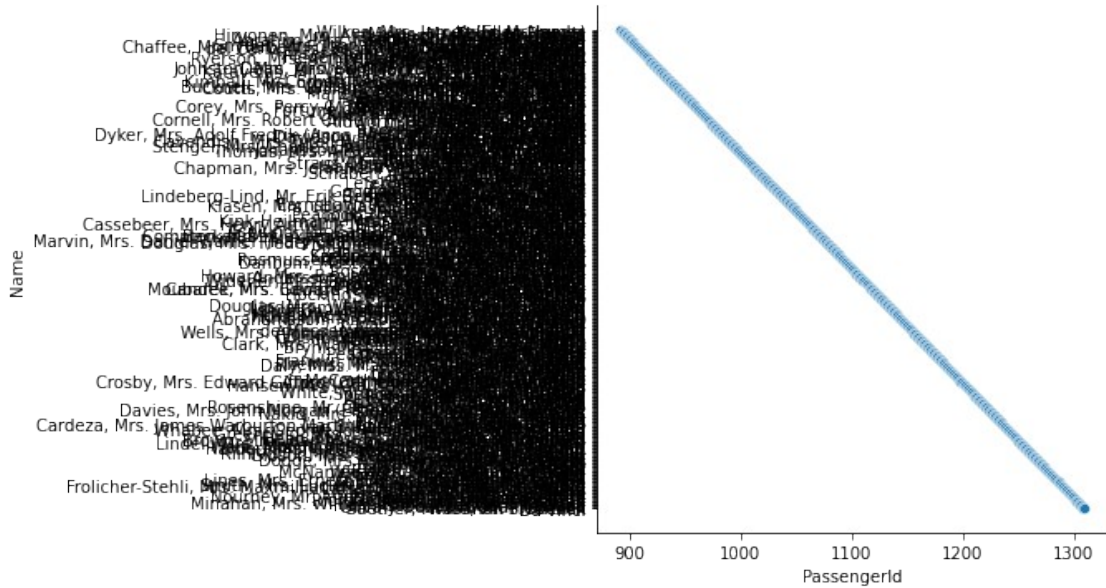
```
<seaborn.axisgrid.FacetGrid at 0x7face7a60050>
```



```
s.relplot(x='PassengerId', y='Name', data=data_Source)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:88:
UserWarning: Tight layout not applied. The left and right margins
cannot be made large enough to accommodate all axes decorations.
  self._figure.tight_layout(*args, **kwargs)
```

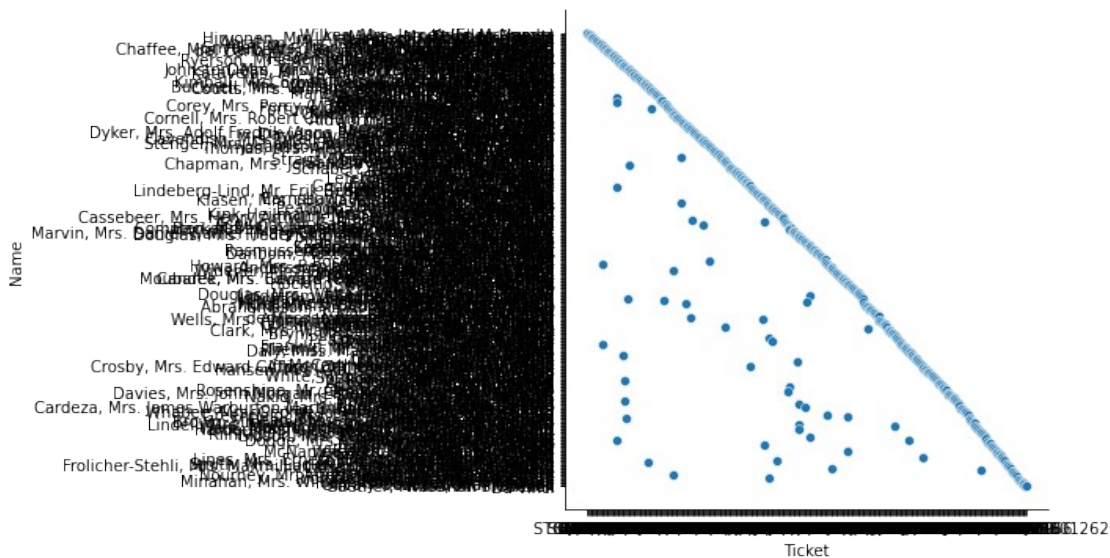
```
<seaborn.axisgrid.FacetGrid at 0x7faceca01090>
```



```
s.relplot(x='Ticket', y='Name', data=data_Source)
```

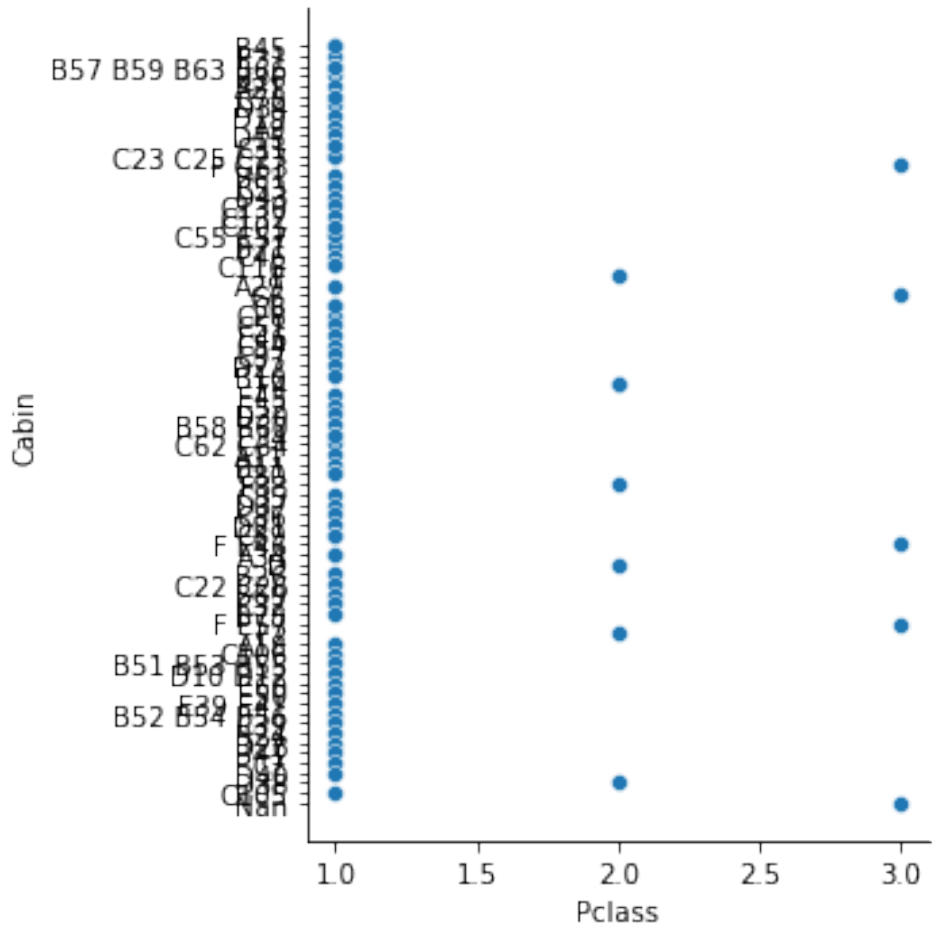
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:88:
 UserWarning: Tight layout not applied. The left and right margins
 cannot be made large enough to accommodate all axes decorations.
 self._figure.tight_layout(*args, **kwargs)

<seaborn.axisgrid.FacetGrid at 0x7face6c2f610>

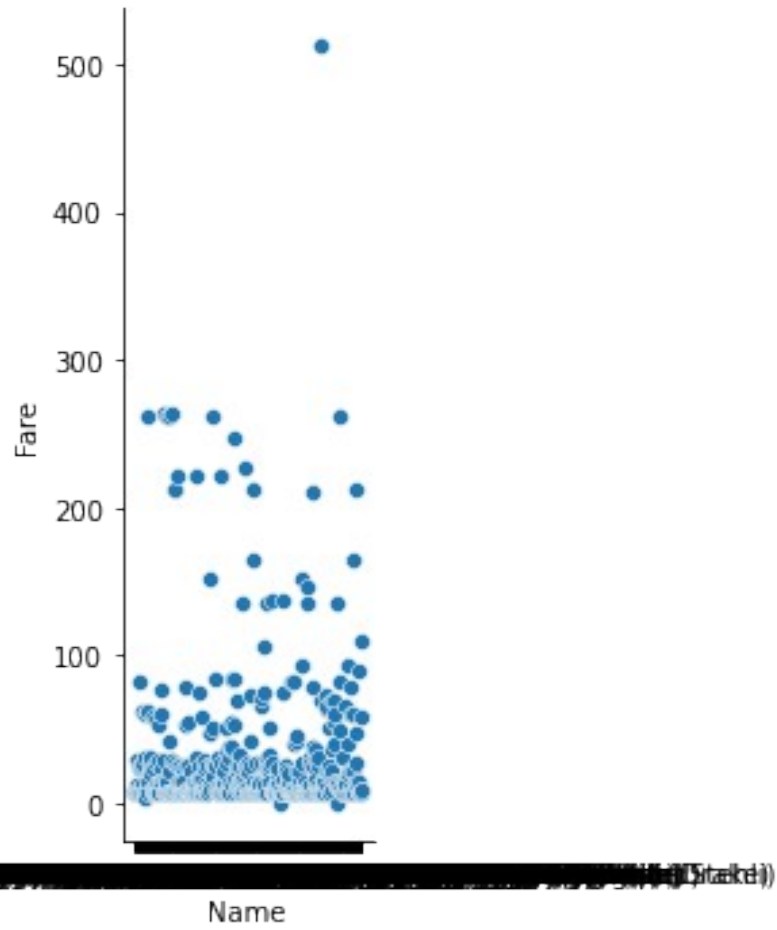


```
s.relplot(x='Pclass', y='Cabin', data=data_Source)
```

<seaborn.axisgrid.FacetGrid at 0x7face6bf5590>



```
s.relplot(x='Name', y='Fare', data=data_Source)
<seaborn.axisgrid.FacetGrid at 0x7face623fe10>
```



```
s.relplot(x='PassengerId', y='Name', hue='Ticket', data=data_Source)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/axisgrid.py:88:
UserWarning: Tight layout not applied. The left and right margins
cannot be made large enough to accommodate all axes decorations.
  self._figure.tight_layout(*args, **kwargs)
```

```
<seaborn.axisgrid.FacetGrid at 0x7face59fae10>
```

Ticket

- 330911
- 363272
- 240276
- 315154
- 3101298
- 7538
- 330972
- 248738
- 2657
- A/4 48871
- 349220
- 694
- 21228
- 24065
- W.E.P. 5734
- SC/PARIS 2167
- 233734
- 2692
- STON/O2. 3101270
- 2696
- PC 17603
- C 17368
- PC 17598
- PC 17597
- PC 17608
- A/5. 3337
- 113509
- 2698
- 113054
- 2662
- SC/AH 3085
- C.A. 31029
- C.A. 2315
- W/C. 6607
- 13236
- 2682
- 342712
- 315087
- 345768
- 1601
- 349256
- 113778
- SOTON/O.Q. 3101263
- 237249
- 11753
- STON/O 2. 3101291
- PC 17594
- 370374
- 11813
- C.A. 37671
- 13695
- SC/PARIS 2168
- 29105
- 19950
- SC/A.3 2861
- 382652
- 349230
- 348122
- 386525
- 349232
- 237216
- 347090
- 334914
- F.C.C. 13534
- 330963
- 113796
- 2543
- 382653
- 349211
- 3101297
- PC 17562
- 113503
- 359306
- 11770
- 248744
- 368702
- 2678
- PC 17483
- 19924
- 349238
- 240261
- 2660
- 330844

```

from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

train=data_Source.drop(['TotalOrgsWorked','Attrition','LastPromotionYears','AverageFeedback','LastPayIncrementBand'], axis=1)
test=data_Source['EmployeeID']

X_train, X_test,Y_train, Y_test= train_test_split(train, test,
test_size=0.4,random_state=2)

regre=LinearRegression()

regre.fit(X_train, Y_train)

LinearRegression()

predct=regre.predict(X_test)

predct

array([ 38., 727., 847., 296., 925., 659., 683., 287., 881., 273.,
138.,
      955., 516., 789., 861., 641., 738., 684., 413., 72., 214.,
68.,
      232., 857., 198., 640., 377., 193., 100., 679., 651., 224.,
206.,
      749., 463., 147., 21., 898., 547., 647., 688., 936., 822.,
554.,
      590., 371., 597., 887., 795., 914., 16., 758., 445., 331.,
872.,
      985., 544., 526., 165., 864., 183., 928., 102., 579., 186.,
69.,
      616., 550., 339., 323., 280., 289., 252., 909., 237., 509.,
717.,
      932., 231., 622., 728., 233., 833., 410., 798., 144., 907.,
600.,
      200., 66., 180., 281., 831., 743., 464., 159., 716., 882.,
825.,
      519., 654., 259., 952., 397., 558., 835., 433., 427., 523.,
505.,
      733., 699., 8., 913., 581., 194., 692., 557., 686., 841.,
412.,
      466., 975., 665., 268., 213., 888., 424., 294., 828., 816.,
813.,
      613., 870., 653., 90., 142., 419., 311., 734., 482., 240.,
71.,
      376., 313., 399., 456., 364., 829., 849., 615., 930., 398.,
446.,
      415., 380., 292., 75., 227., 487., 334., 85., 481., 695.,
285.,
      210., 143., 251., 926., 356., 577., 389., 536., 830., 465.,
2.,

```



```
184., 110., 161., 938., 812., 432., 392., 902., 173., 129.,
803.,
799., 780., 170., 485., 452., 578., 78., 576., 840., 811.,
931.,
617., 810., 176., 819., 678., 953., 201., 533., 312., 553.,
527.,
896., 400., 706., 310., 153., 834., 274., 820., 879., 375.,
92.,
958., 453., 934., 366., 974., 319., 63., 19., 977., 983.,
750.,
745., 348., 618., 676., 23., 661., 707., 511., 189., 236.,
420.,
740., 700., 543., 945., 316., 228., 242., 491., 674., 108.,
899.,
442., 440., 718., 437., 344., 474., 81., 697., 980., 947.,
31.,
11., 891., 525., 796., 158., 347., 753., 258., 318., 497.,
5.,
711., 199., 117., 205., 693., 175., 702., 203., 757., 921.,
120.,
178., 276., 863., 14., 249., 910., 346., 942., 385., 895.,
541.,
568., 642., 130., 91., 306., 127., 964., 164., 112., 703.,
860.,
869., 454., 720., 729., 328., 503., 73., 3., 574., 826.,
621.,
13., 804., 884., 689., 428., 109., 741., 970., 182., 924.,
949.,
336., 309., 754., 305., 782., 531., 43., 357., 634., 792.,
438.,
772., 472., 295., 601., 868., 559., 855., 594., 778., 439.,
341.,
56., 787., 875., 345., 217., 461., 41., 643., 990., 900.,
712.,
179., 177., 493., 59., 458., 477., 17., 459., 701., 441.,
687.,
890., 460., 823., 878., 638., 660., 520., 418., 30., 956.,
401.,
670., 784., 430., 489., 204., 372., 33., 514., 997., 968.,
157.,
573., 827., 335., 383.]])
```

```
regre.score(X_test, Y_test)
```

```
1.0
```

```
0
```