

In [ ]:

In [173]:

```
# IMPORT LIBRARIES
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [177]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\15_Horse Racing Results.CSV - 15_Horse Racing Results.CSV.csv")
a
```

Out[177]:

	Dato	Track	Race Number	Distance	Surface	Prize money	Starting position	Jockey	Jockey weight	Country	...	Træ
0	03.09.2017	Sha Tin	10	1400	Gress	1310000	6	K C Leung	52	Sverige	...	
1	16.09.2017	Sha Tin	10	1400	Gress	1310000	14	C Y Ho	52	Sverige	...	
2	14.10.2017	Sha Tin	10	1400	Gress	1310000	8	C Y Ho	52	Sverige	...	
3	11.11.2017	Sha Tin	9	1600	Gress	1310000	13	Brett Prebble	54	Sverige	...	
4	26.11.2017	Sha Tin	9	1600	Gress	1310000	9	C Y Ho	52	Sverige	...	
...	...	...	...	...	...	...	...	...	...	...	...	
27003	14.06.2020	Sha Tin	11	1200	Gress	1450000	6	A Hamelin	59	Australia	...	
27004	21.06.2020	Sha Tin	2	1200	Gress	967000	7	K C Leung	57	Australia	...	
27005	21.06.2020	Sha Tin	4	1200	Gress	967000	6	Blake Shinn	57	Australia	...	P
27006	21.06.2020	Sha Tin	5	1200	Gress	967000	14	Joao Moreira	57	New Zealand	...	
27007	21.06.2020	Sha Tin	11	1200	Gress	1450000	7	C Schofield	55	New Zealand	...	

27008 rows × 21 columns

In [178]:

```
a=a.head(10)
a
```

Out[178]:

	Dato	Track	Race Number	Distance	Surface	Prize money	Starting position	Jockey	Jockey weight	Country	...	Trainerl
0	03.09.2017	Sha Tin	10	1400	Gress	1310000	6	K C Leung	52	Sverige	...	C
1	16.09.2017	Sha Tin	10	1400	Gress	1310000	14	C Y Ho	52	Sverige	...	C
2	14.10.2017	Sha Tin	10	1400	Gress	1310000	8	C Y Ho	52	Sverige	...	C
3	11.11.2017	Sha Tin	9	1600	Gress	1310000	13	Brett Prebble	54	Sverige	...	C
4	26.11.2017	Sha Tin	9	1600	Gress	1310000	9	C Y Ho	52	Sverige	...	C
5	10.12.2017	Sha Tin	1	1800	Gress	1310000	4	C Y Ho	52	Sverige	...	C
6	01.01.2018	Sha Tin	9	1800	Gress	1310000	9	C Schofield	54	Sverige	...	C
7	04.02.2018	Sha Tin	5	1800	Gress	1310000	6	Joao Moreira	57	Sverige	...	C
8	03.03.2018	Sha Tin	8	1800	Gress	1310000	3	C Y Ho	56	Sverige	...	C
9	11.03.2018	Sha Tin	10	1600	Gress	1310000	8	C Y Ho	57	Sverige	...	C

10 rows × 21 columns



In [179]:

```
# to find  
a.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 10 entries, 0 to 9  
Data columns (total 21 columns):  
#   Column                Non-Null Count  Dtype    
---  ---  
0   Dato                   10 non-null     object   
1   Track                  10 non-null     object   
2   Race Number           10 non-null     int64    
3   Distance               10 non-null     int64    
4   Surface                10 non-null     object   
5   Prize money            10 non-null     int64    
6   Starting position      10 non-null     int64    
7   Jockey                 10 non-null     object   
8   Jockey weight          10 non-null     int64    
9   Country                10 non-null     object   
10  Horse age              10 non-null     int64    
11  TrainerName            10 non-null     object   
12  Race time              10 non-null     object   
13  Path                   10 non-null     int64    
14  Final place            10 non-null     int64    
15  FGrating               10 non-null     int64    
16  Odds                   10 non-null     object   
17  RaceType               10 non-null     object   
18  HorseId                10 non-null     int64    
19  JockeyId                10 non-null     int64    
20  TrainerID              10 non-null     int64    
dtypes: int64(12), object(9)  
memory usage: 1.8+ KB
```

In [180]:

```
# to display summary of statistic  
a.describe()
```

Out[180]:

	Race Number	Distance	Prize money	Starting position	Jockey weight	Horse age	Path	Final place	FGratin
count	10.000000	10.000000	10.0	10.000000	10.000000	10.000000	10.000000	10.000000	10.000000
mean	8.100000	1620.000000	1310000.0	8.000000	53.800000	7.400000	1.500000	4.700000	120.10000
std	2.923088	175.119007	0.0	3.527668	2.149935	0.516398	1.581139	2.496664	6.47130
min	1.000000	1400.000000	1310000.0	3.000000	52.000000	7.000000	0.000000	1.000000	107.00000
25%	8.250000	1450.000000	1310000.0	6.000000	52.000000	7.000000	0.250000	3.000000	119.00000
50%	9.000000	1600.000000	1310000.0	8.000000	53.000000	7.000000	1.000000	4.000000	123.00000
75%	10.000000	1800.000000	1310000.0	9.000000	55.500000	8.000000	2.000000	6.000000	124.00000
max	10.000000	1800.000000	1310000.0	14.000000	57.000000	8.000000	5.000000	9.000000	125.00000

In [181]:

```
# to display colum heading
a.columns
```

Out[181]:

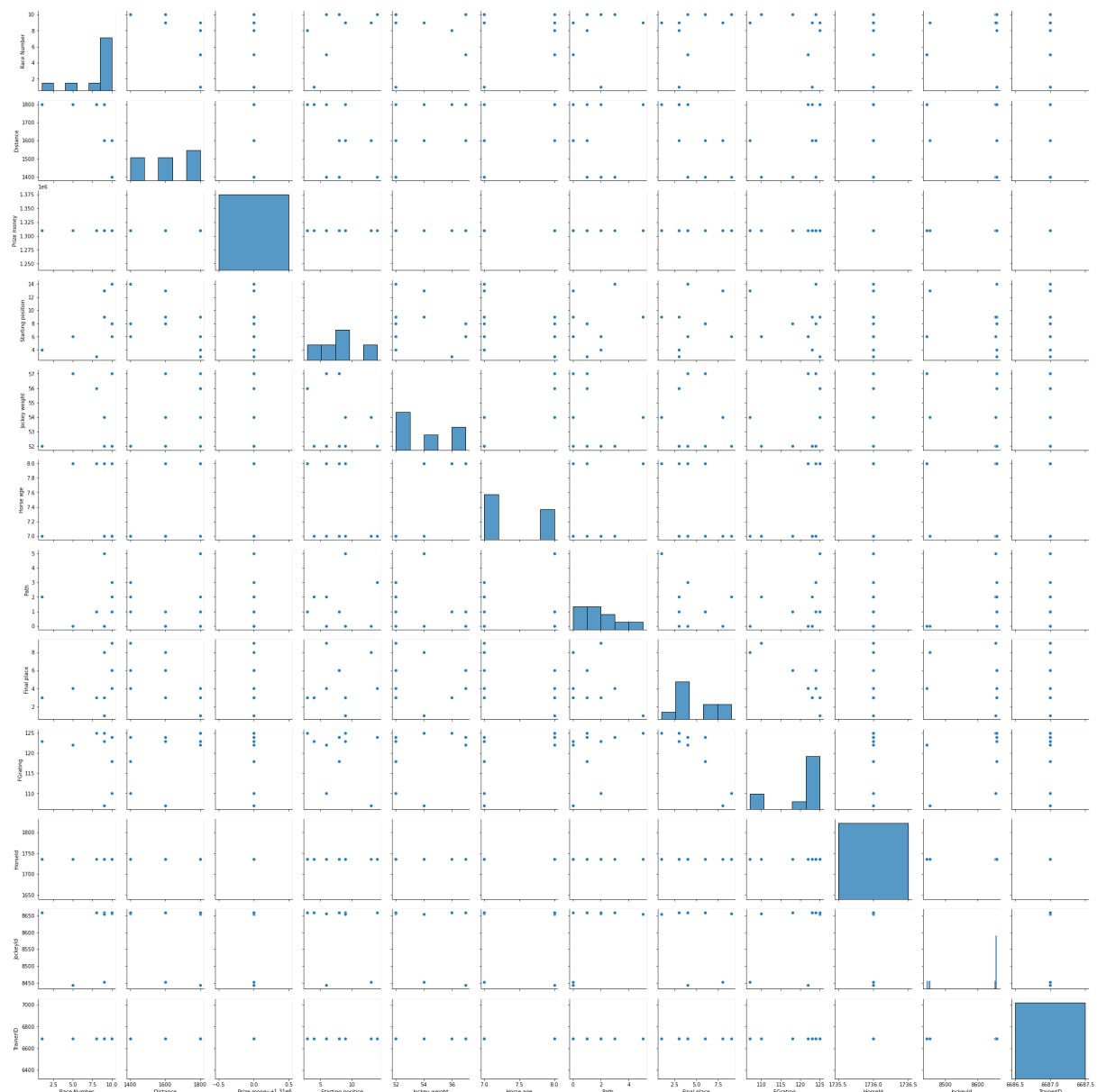
```
Index(['Dato', 'Track', 'Race Number', 'Distance', 'Surface', 'Prize money',
       'Starting position', 'Jockey', 'Jockey weight', 'Country', 'Horse age',
       'TrainerName', 'Race time', 'Path', 'Final place', 'FGrating', 'Odds',
       'RaceType', 'HorseId', 'JockeyId', 'TrainerID'],
      dtype='object')
```

In [182]:

```
sns.pairplot(a)
```

Out[182]:

<seaborn.axisgrid.PairGrid at 0x203e258b190>

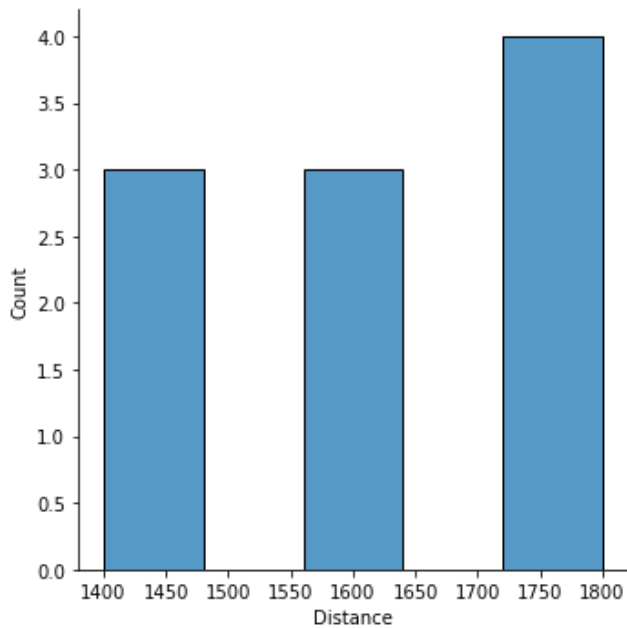


In [183]:

```
sns.displot(a["Distance"])
```

Out[183]:

<seaborn.axisgrid.FacetGrid at 0x203e72741c0>



In [184]:

```
b=a[['Dato', 'Track', 'Race Number', 'Distance', 'Surface', 'Prize money',  
     'Starting position', 'Jockey', 'Jockey weight', 'Country', 'Horse age',  
     'TrainerName', 'Race time', 'Path', ]]  
b
```

Out[184]:

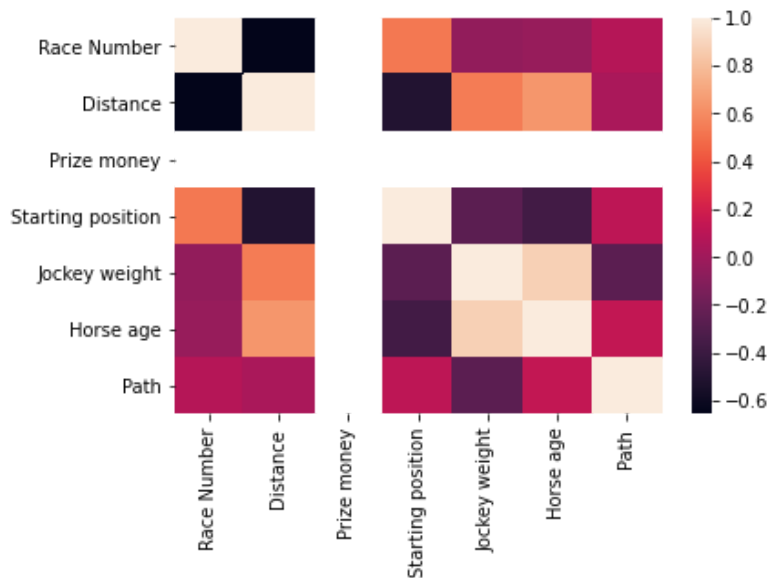
	Dato	Track	Race Number	Distance	Surface	Prize money	Starting position	Jockey	Jockey weight	Country	Horse age	Tra
0	03.09.2017	Sha Tin	10	1400	Gress	1310000	6	K C Leung	52	Sverige	7	
1	16.09.2017	Sha Tin	10	1400	Gress	1310000	14	C Y Ho	52	Sverige	7	
2	14.10.2017	Sha Tin	10	1400	Gress	1310000	8	C Y Ho	52	Sverige	7	
3	11.11.2017	Sha Tin	9	1600	Gress	1310000	13	Brett Prebble	54	Sverige	7	
4	26.11.2017	Sha Tin	9	1600	Gress	1310000	9	C Y Ho	52	Sverige	7	
5	10.12.2017	Sha Tin	1	1800	Gress	1310000	4	C Y Ho	52	Sverige	7	
6	01.01.2018	Sha Tin	9	1800	Gress	1310000	9	C Schofield	54	Sverige	8	
7	04.02.2018	Sha Tin	5	1800	Gress	1310000	6	Joao Moreira	57	Sverige	8	
8	03.03.2018	Sha Tin	8	1800	Gress	1310000	3	C Y Ho	56	Sverige	8	
9	11.03.2018	Sha Tin	10	1600	Gress	1310000	8	C Y Ho	57	Sverige	8	

In [185]:

```
sns.heatmap(b.corr())
```

Out[185]:

<AxesSubplot:>



In [194]:

```
x=a[['Race Number', 'Distance', 'Prize money']]  
y=a['Distance']
```

In [195]:

```
from sklearn.model_selection import train_test_split  
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3)
```

In [196]:

```
from sklearn.linear_model import LinearRegression  
lr=LinearRegression()  
lr.fit(x_train,y_train)
```

Out[196]:

LinearRegression()

In [197]:

```
lr.intercept_
```

Out[197]:

-1.1368683772161603e-12

In [198]:

```
coeff=pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])  
coeff
```

Out[198]:

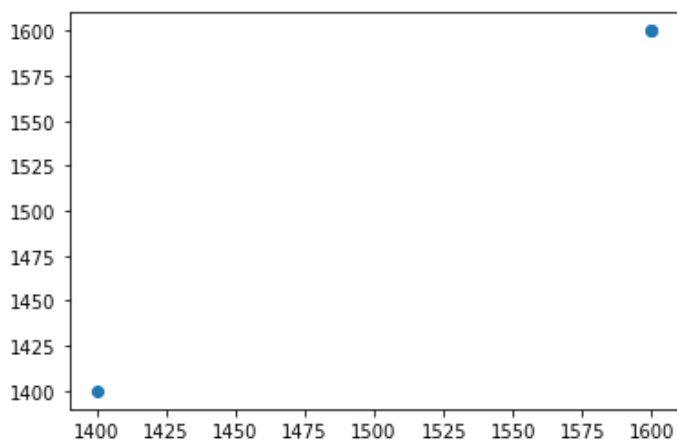
	Co-efficient
Race Number	-5.123237e-17
Distance	1.000000e+00
Prize money	0.000000e+00

In [199]:

```
prediction = lr.predict(x_test)  
plt.scatter(y_test,prediction)
```

Out[199]:

<matplotlib.collections.PathCollection at 0x203e90eb7c0>



In [200]:

```
lr.score(x_test,y_test)
```

Out[200]:

1.0

In [201]:

```
lr.score(x_train,y_train)
```

Out[201]:

1.0

In [202]:

```
from sklearn.linear_model import Ridge,Lasso
```

In [203]:

```
rr=Ridge(alpha=10)  
rr.fit(x_test,y_test)
```

Out[203]:

Ridge(alpha=10)

In [204]:

```
rr.score(x_test,y_test)
```

Out[204]:

0.9999998594874325

In [205]:

```
la=Lasso(alpha=10)  
la.fit(x_test,y_test)
```

Out[205]:

Lasso(alpha=10)

In [206]:

```
la.score(x_test,y_test)
```

Out[206]:

0.999998734375

In [ ]: