In [37]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [38]:
```
data=pd.read_csv(r"C:\Users\user\Downloads\15_Horse Racing Results - 15_Horse Racing Results.csv")
data
```

3	11.11.2017	Sha Tin	9	1600	Gress	1310000	13	Brett Prebble	54	Sverige	...	CH Yip
4	26.11.2017	Sha Tin	9	1600	Gress	1310000	9	C Y Ho	52	Sverige	...	CH Yip
...	...	...	...	...	...	...	...	...	...	...	...	...
27003	14.06.2020	Sha Tin	11	1200	Gress	1450000	6	A Hamelin	59	Australia	...	WY So
27004	21.06.2020	Sha Tin	2	1200	Gress	967000	7	K C Leung	57	Australia	...	KL Man
27005	21.06.2020	Sha Tin	4	1200	Gress	967000	6	Blake Shinn	57	Australia	...	P O'Sullivan
27006	21.06.2020	Sha Tin	5	1200	Gress	967000	14	Joao Moreira	57	New Zealand	...	AS Cruz
27007	21.06.2020	Sha Tin	11	1200	Gress	1450000	7	C Schofield	55	New Zealand	...	WY So

In [39]:
```
df=data.head(100)
df
```

Out[39]:

	Dato	Track	Race Number	Distance	Surface	Prize money	Starting position	Jockey	Jockey weight	Country	...	TrainerName	Race time
0	03.09.2017	Sha Tin	10	1400	Gress	1310000	6	K C Leung	52	Sverige	...	CH Yip	83,38
1	16.09.2017	Sha Tin	10	1400	Gress	1310000	14	C Y Ho	52	Sverige	...	CH Yip	81,56
2	14.10.2017	Sha Tin	10	1400	Gress	1310000	8	C Y Ho	52	Sverige	...	CH Yip	82,36
3	11.11.2017	Sha Tin	9	1600	Gress	1310000	13	Brett Prebble	54	Sverige	...	CH Yip	96,53
4	26.11.2017	Sha Tin	9	1600	Gress	1310000	9	C Y Ho	52	Sverige	...	CH Yip	94,17
...	...	...	...	...	...	...	...	...	...	...	...	...	...
95	10.12.2017	Sha Tin	5	1200	Gress	18500000	13	Francois-Xavier Bertras	57	Great Britain	...	D Guillemin	69,22
96	10.12.2017	Sha Tin	7	1600	Gress	23000000	11	Ryan Moore	57	USA	...	A O'Brien	94,24
97	01.10.2017	Sha Tin	7	1000	Gress	3000000	10	Brett Prebble	59	New Zealand	...	J Size	56,99
98	22.10.2017	Sha Tin	7	1200	Gress	4000000	9	Brett Prebble	59	New Zealand	...	J Size	68,27
99	19.11.2017	Sha Tin	7	1200	Gress	4000000	3	Brett Prebble	56	New Zealand	...	J Size	69,57

100 rows × 21 columns

In [40]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 21 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   Dato               100 non-null    object
 1   Track              100 non-null    object
 2   Race Number        100 non-null    int64 
 3   Distance           100 non-null    int64 
 4   Surface            100 non-null    object
 5   Prize money        100 non-null    int64 
 6   Starting position  100 non-null    int64 
 7   Jockey             100 non-null    object
 8   Jockey weight      100 non-null    int64 
 9   Country            100 non-null    object
 10  Horse age          100 non-null    int64 
 11  TrainerName        100 non-null    object
 12  Race time          100 non-null    object
 13  Path               100 non-null    int64 
 14  Final place        100 non-null    int64 
 15  FGrating           100 non-null    int64 
 16  Odds               100 non-null    object
 17  RaceType           100 non-null    object
 18  HorseId            100 non-null    int64 
 19  JockeyId           100 non-null    int64 
 20  TrainerID          100 non-null    int64 
dtypes: int64(12), object(9)
memory usage: 16.5+ KB
```

In [41]:
```
df.describe()
```

Out[41]:

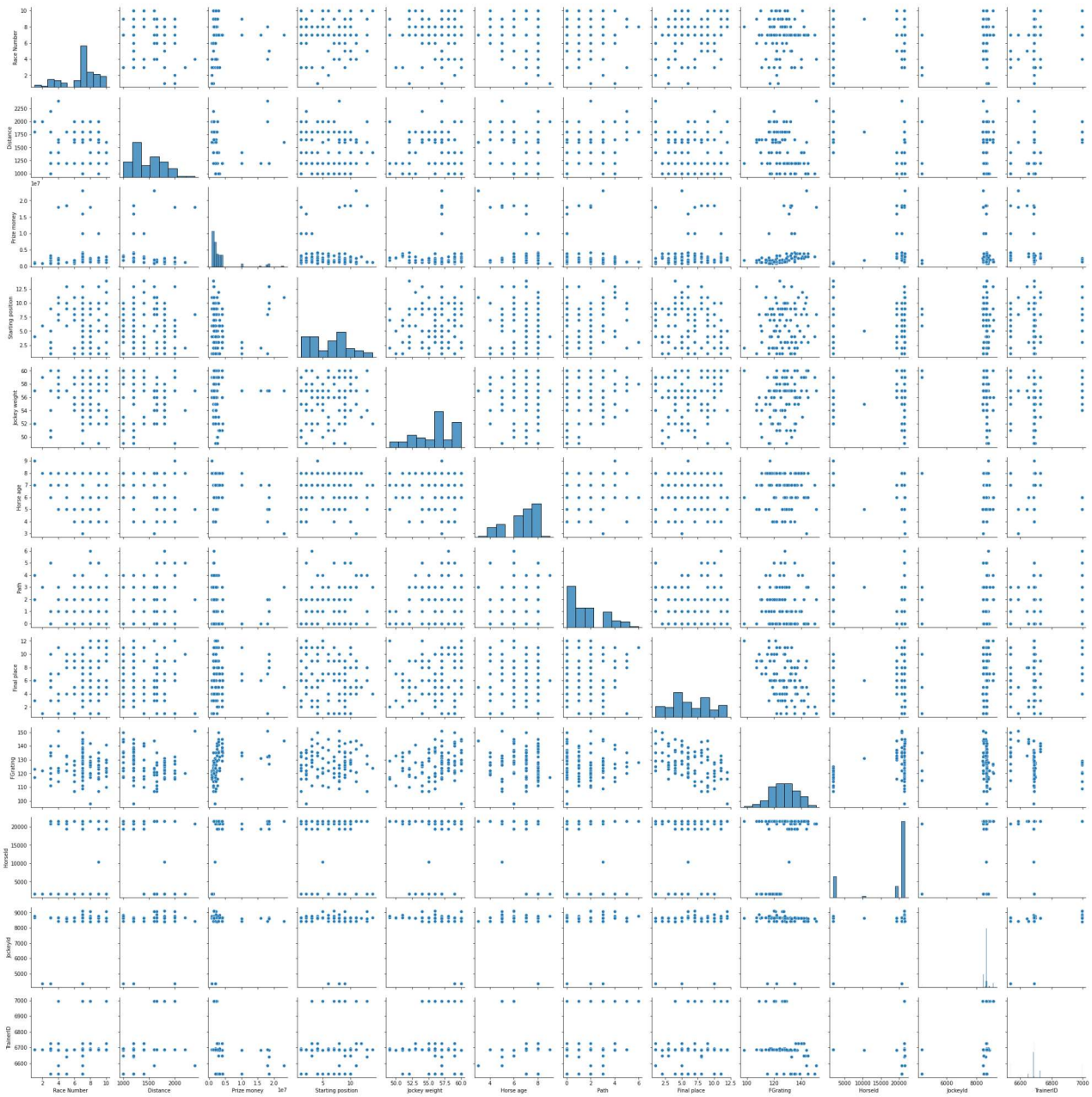
ce	Prize money	Starting position	Jockey weight	Horse age	Path	Final place	FGrating	HorseId	JockeyId	Traine
00	1.000000e+02	100.000000	100.000000	100.00000	100.000000	100.000000	100.000000	100.000000	100.000000	100.00
00	3.562200e+06	6.170000	55.870000	6.58000	1.510000	6.330000	127.120000	17336.270000	8508.620000	6702.63
23	4.486259e+06	3.440857	2.942736	1.35721	1.573101	3.011946	10.047644	7695.941777	748.998102	121.37
00	9.200000e+05	1.000000	49.000000	3.00000	0.000000	1.000000	98.000000	1736.000000	4340.000000	6535.00
00	1.380000e+06	3.000000	54.000000	6.00000	0.000000	4.000000	120.000000	19373.000000	8609.000000	6683.00
00	1.950000e+06	6.000000	56.000000	7.00000	1.000000	6.000000	127.000000	21512.500000	8655.000000	6687.00
00	3.000000e+06	9.000000	58.000000	8.00000	3.000000	9.000000	135.000000	21523.000000	8660.500000	6687.00
00	2.300000e+07	14.000000	60.000000	9.00000	6.000000	12.000000	151.000000	21587.000000	9111.000000	6996.00

In [42]:
```
df.columns
```

Out[42]:
```
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 [43]:
```
sns.pairplot(df)
```

Out[43]: <seaborn.axisgrid.PairGrid at 0x1a20af21eb0>



In [44]:
```
'Distance', 'Prize money',
ion', 'Jockey weight','Horse age', 'Path', 'Final place', 'FGrating', 'Odds', 'JockeyId', 'TrainerID
```

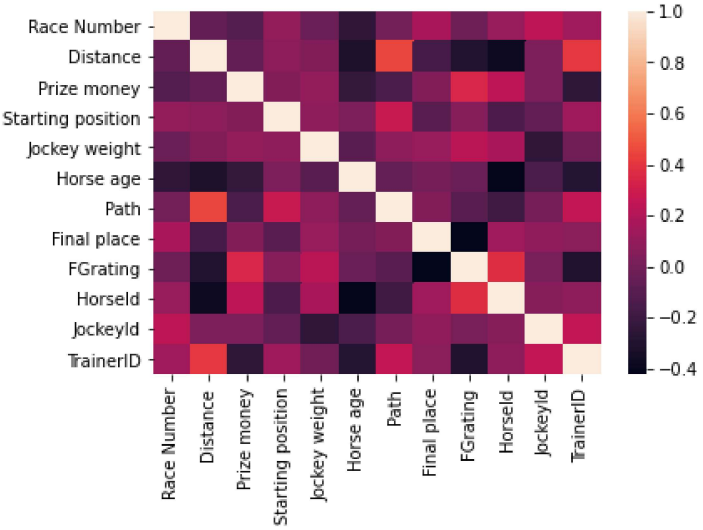
Out[44]:

	Dato	Track	Race Number	Distance	Surface	Prize money	Starting position	Jockey	Jockey weight	Country	...	TrainerName	Race time
0	03.09.2017	Sha Tin	10	1400	Gress	1310000	6	K C Leung	52	Sverige	...	CH Yip	83,38
1	16.09.2017	Sha Tin	10	1400	Gress	1310000	14	C Y Ho	52	Sverige	...	CH Yip	81,56
2	14.10.2017	Sha Tin	10	1400	Gress	1310000	8	C Y Ho	52	Sverige	...	CH Yip	82,36
3	11.11.2017	Sha Tin	9	1600	Gress	1310000	13	Brett Prebble	54	Sverige	...	CH Yip	96,53
4	26.11.2017	Sha Tin	9	1600	Gress	1310000	9	C Y Ho	52	Sverige	...	CH Yip	94,17
...	...	...	...	...	...	...	...	...	...	...	...	...	...
95	10.12.2017	Sha Tin	5	1200	Gress	18500000	13	Francois-Xavier Bertras	57	Great Britain	...	D Guillemin	69,22
96	10.12.2017	Sha Tin	7	1600	Gress	23000000	11	Ryan Moore	57	USA	...	A O'Brien	94,24
97	01.10.2017	Sha Tin	7	1000	Gress	3000000	10	Brett Prebble	59	New Zealand	...	J Size	56,99
98	22.10.2017	Sha Tin	7	1200	Gress	4000000	9	Brett Prebble	59	New Zealand	...	J Size	68,27
99	19.11.2017	Sha Tin	7	1200	Gress	4000000	3	Brett Prebble	56	New Zealand	...	J Size	69,57

100 rows × 21 columns

In [45]:
```
sns.heatmap(da.corr())
```

Out[45]: <AxesSubplot:>



In [53]:
```
x=da[['Prize money','Jockey weight','Horse age', 'Path', 'Final place','JockeyId', 'TrainerID']]
y=da['Horse age']
```

In [54]:
```
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 [55]:
```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
lr.fit(x_train,y_train)
```

Out[55]:
```
LinearRegression()
```

In [56]:
```
print(lr.intercept_)
```

```
1.5791812302268227e-12
```

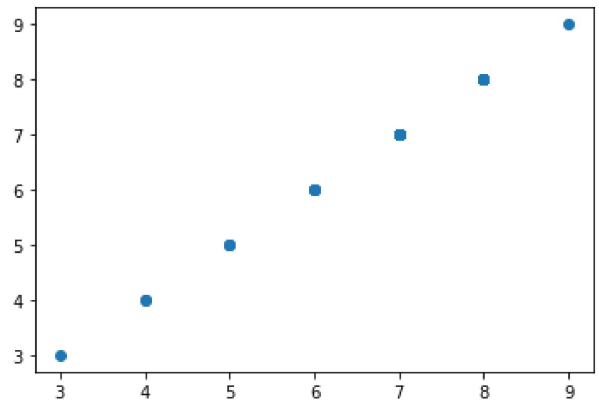
In [57]:
```
coeff = pd.DataFrame(lr.coef_,x.columns,columns=['Co-efficient'])
coeff
```

Out[57]:

	Co-efficient
Prize money	3.068981e-21
Jockey weight	7.233797e-16
Horse age	1.000000e+00
Path	-1.716448e-15
Final place	-2.169150e-16
JockeyId	-3.009745e-16
TrainerID	1.400789e-16

In [58]:
```
prediction=lr.predict(x_test)
plt.scatter(y_test,prediction)
```

Out[58]:
```
<matplotlib.collections.PathCollection at 0x1a212e0a4c0>
```



In [59]:
```
print(lr.score(x_test,y_test))
```

```
1.0
```

In [60]:
```
print(lr.score(x_train,y_train))
```

```
1.0
```

In [61]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [62]:
```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[62]:
```
Ridge(alpha=10)
```

In [63]:
```
rr.score(x_test,y_test)
```

Out[63]:
```
0.9943482570518789
```

In [64]:
```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[64]:
```
Lasso(alpha=10)
```

In [65]:
```
la.score(x_test,y_test)
```

Out[65]:
```
0.2426747543033907
```

In [ ]:
```

```