In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as py
import seaborn as sns
```

In [2]:

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

Out[2]:

	Dato	Track	Race Number	Distance	Surface	Prize money	Starting position	Jockey	Jockey weight	Country	...
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	...
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 [4]:

```
d.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27008 entries, 0 to 27007
Data columns (total 21 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   Dato                27008 non-null  object
 1   Track               27008 non-null  object
 2   Race Number         27008 non-null  int64 
 3   Distance            27008 non-null  int64 
 4   Surface             27008 non-null  object
 5   Prize money        27008 non-null  int64 
 6   Starting position  27008 non-null  int64 
 7   Jockey             27008 non-null  object
 8   Jockey weight      27008 non-null  int64 
 9   Country            27008 non-null  object
 10  Horse age          27008 non-null  int64 
 11  TrainerName        27008 non-null  object
 12  Race time          27008 non-null  object
 13  Path               27008 non-null  int64 
 14  Final place        27008 non-null  int64 
 15  FGrating           27008 non-null  int64 
 16  Odds               27008 non-null  object
 17  RaceType           27008 non-null  object
 18  HorseId            27008 non-null  int64 
 19  JockeyId           27008 non-null  int64 
 20  TrainerID          27008 non-null  int64 
dtypes: int64(12), object(9)
memory usage: 4.3+ MB
```

In [3]:

```
d.isna()
```

Out[3]:

	Dato	Track	Race Number	Distance	Surface	Prize money	Starting position	Jockey	Jockey weight	Country	...	Trainer
0	False	False	False	False	False	False	False	False	False	False	...	
1	False	False	False	False	False	False	False	False	False	False	...	
2	False	False	False	False	False	False	False	False	False	False	...	
3	False	False	False	False	False	False	False	False	False	False	...	
4	False	False	False	False	False	False	False	False	False	False	...	
...	...	...	...	...	...	...	...	...	...	...	...	
27003	False	False	False	False	False	False	False	False	False	False	...	
27004	False	False	False	False	False	False	False	False	False	False	...	
27005	False	False	False	False	False	False	False	False	False	False	...	
27006	False	False	False	False	False	False	False	False	False	False	...	
27007	False	False	False	False	False	False	False	False	False	False	...	

27008 rows × 21 columns

In [5]:

```
d.describe()
```

Out[5]:

	Race Number	Distance	Prize money	Starting position	Jockey weight	Horse age	Pat
count	27008.000000	27008.000000	2.700800e+04	27008.000000	27008.000000	27008.000000	27008.00000
mean	5.268624	1401.666173	1.479445e+06	6.741447	55.867373	5.246408	1.67802
std	2.780088	276.065045	2.162109e+06	3.691071	2.737006	1.519880	1.63178
min	1.000000	1000.000000	6.600000e+05	1.000000	47.000000	2.000000	0.00000

	Race Number	Distance	Prize money	Starting position	Jockey weight	Horse age	Pat
25%	3.000000	1200.000000	9.200000e+05	4.000000	54.000000	4.000000	0.00000
50%	5.000000	1400.000000	9.670000e+05	7.000000	56.000000	5.000000	1.00000
75%	8.000000	1650.000000	1.450000e+06	10.000000	58.000000	6.000000	3.00000
max	11.000000	2400.000000	2.800000e+07	14.000000	63.000000	12.000000	11.00000

In [6]:

```
d.columns
```

Out[6]:

```
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 [7]:

```
d.index
```

Out[7]:

```
RangeIndex(start=0, stop=27008, step=1)
```

In [8]:

```
d=d.head(10)
d
```

Out[8]:

	Dato	Track	Race Number	Distance	Surface	Prize money	Starting position	Jockey	Jockey weight	Country	...	Trai
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	...	
5	10.12.2017	Sha Tin	1	1800	Gress	1310000	4	C Y Ho	52	Sverige	...	
6	01.01.2018	Sha Tin	9	1800	Gress	1310000	9	C Schofield	54	Sverige	...	
7	04.02.2018	Sha Tin	5	1800	Gress	1310000	6	Joao Moreira	57	Sverige	...	
8	03.03.2018	Sha Tin	8	1800	Gress	1310000	3	C Y Ho	56	Sverige	...	
9	11.03.2018	Sha Tin	10	1600	Gress	1310000	8	C Y Ho	57	Sverige	...	

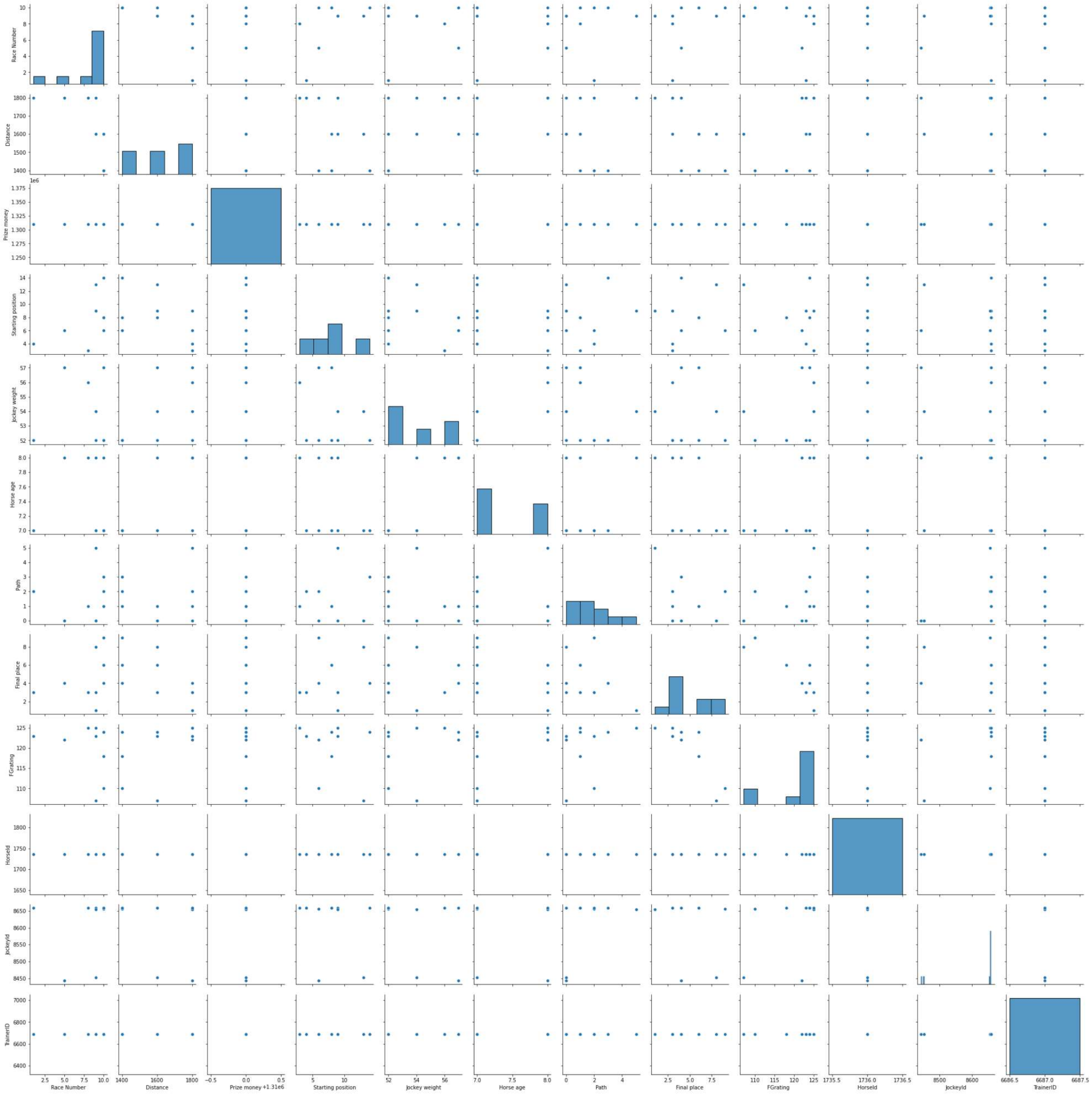
10 rows × 21 columns

In [9]:

```
sns.pairplot(d)
```

Out[9]:

```
<seaborn.axisgrid.PairGrid at 0x1f987180640>
```



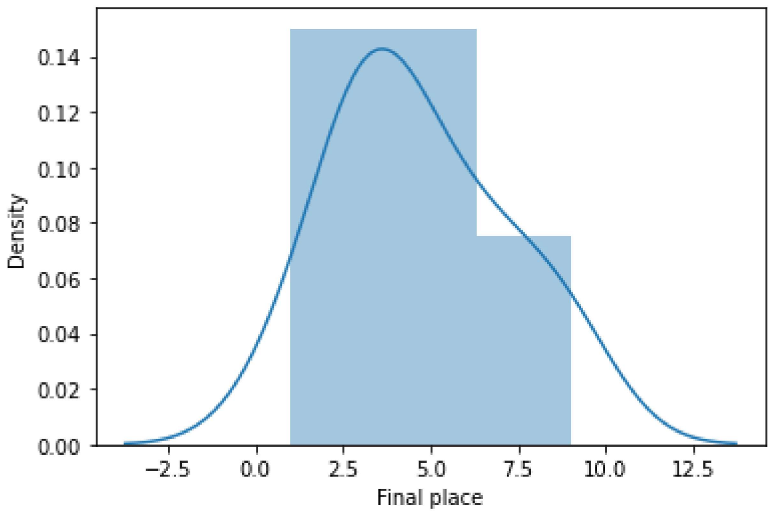
In [10]:

```
sns.distplot(d['Final place'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: FutureWarning:
`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

Out[10]:

```
<AxesSubplot:xlabel='Final place', ylabel='Density'>
```

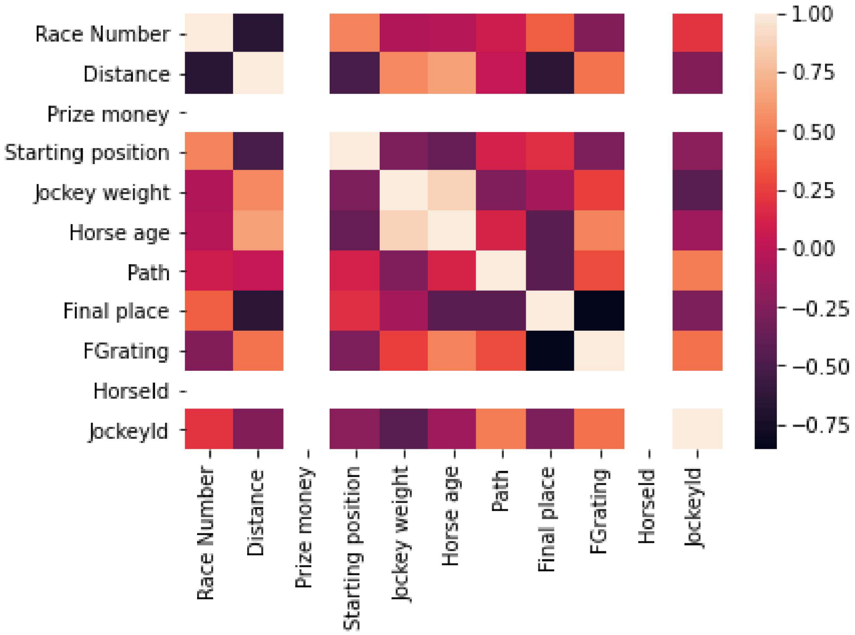


In [47]:

```
d1=d[['Race Number', 'Distance','Prize money',
       'Starting position','Jockey weight','Horse age','Race time', 'Path', 'Final plac
sns.heatmap(d1.corr())
```

Out[47]:

```
<AxesSubplot:>
```



In [48]:

```
x=d1[['Race Number', 'Distance','Starting position','Jockey weight','Horse age','HorseI
y=d1[ 'Final place']
```

In [49]:

```
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 [50]:

```
from sklearn.linear_model import LinearRegression
```

In [51]:

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

Out[51]:
```
LinearRegression()
```

In [52]:
```
print(lr.intercept_)
```

```
1573.0962711864383
```

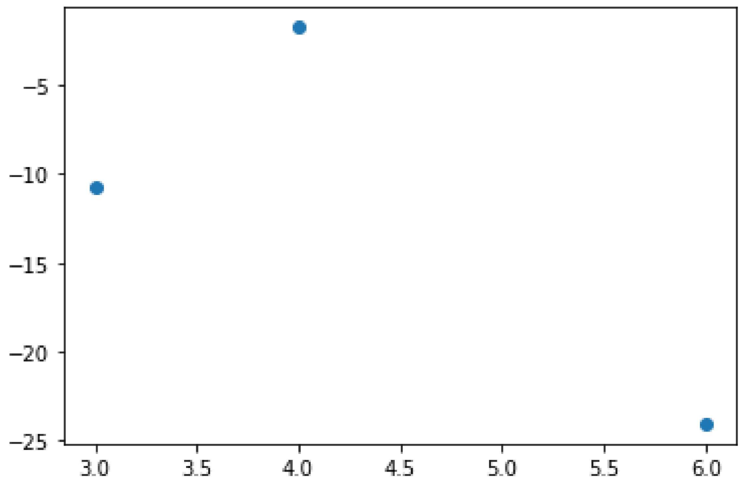
In [53]:
```
coeff =pd.DataFrame(lr.coef_,x.columns,columns=["Co-efficient"])
coeff
```

Out[53]:

	Co-efficient
Race Number	0.679322
Distance	-0.005141
Starting position	-1.292542
Jockey weight	-9.160339
Horse age	16.795593
HorseId	0.000000
JockeyId	-0.138305

In [54]:
```
prediction =lr.predict(x_test)
py.scatter(y_test,prediction)
```

Out[54]:
```
<matplotlib.collections.PathCollection at 0x1f990d15a00>
```



In [57]:
```
print(lr.score(x_test,y_test))
```

```
-240.19190629128846
```

In [58]:
```
print(lr.score(x_train,y_train))
```

```
1.0
```

In [59]:
```
from sklearn.linear_model import Ridge,Lasso
```

In [60]:

```
rr=Ridge(alpha=10)
rr.fit(x_train,y_train)
```

Out[60]:

```
Ridge(alpha=10)
```

In [61]:

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

Out[61]:

```
-3.3649491424459974
```

In [62]:

```
la=Lasso(alpha=10)
la.fit(x_train,y_train)
```

Out[62]:

```
Lasso(alpha=10)
```

In [63]:

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

Out[63]:

```
-2.1482100784784586
```

In [ ]: