In [76]:
```
# import libaries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [372]:
```
x=pd.read_csv(r"C:\Users\user\Downloads\15_Horse Racing Results - 15_Horse Rac
```

Out[372]:

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

27008 rows × 21 columns

In [375]:
```
x=x.head(10)
```

Out[375]:

	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
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 [376]:

```
<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 [377]:

Out[377]:

```
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 [378]:

```
d=x[['Dato', 'Track', 'Race Number', 'Distance', 'Surface', 'Prize money']]
```

Out[378]:

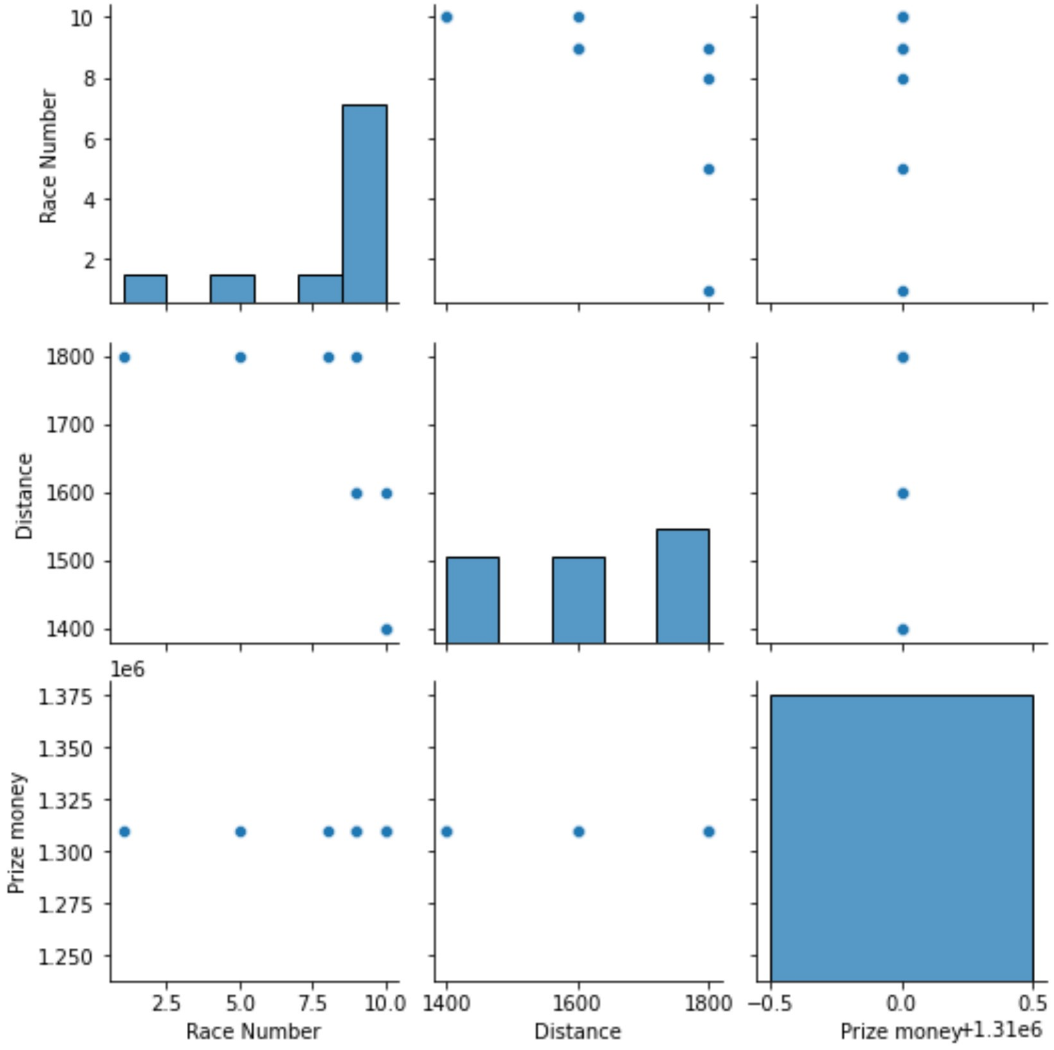
	Dato	Track	Race Number	Distance	Surface	Prize money
0	03.09.2017	Sha Tin	10	1400	Gress	1310000
1	16.09.2017	Sha Tin	10	1400	Gress	1310000
2	14.10.2017	Sha Tin	10	1400	Gress	1310000
3	11.11.2017	Sha Tin	9	1600	Gress	1310000
4	26.11.2017	Sha Tin	9	1600	Gress	1310000
5	10.12.2017	Sha Tin	1	1800	Gress	1310000
6	01.01.2018	Sha Tin	9	1800	Gress	1310000
7	04.02.2018	Sha Tin	5	1800	Gress	1310000
8	03.03.2018	Sha Tin	8	1800	Gress	1310000
9	11.03.2018	Sha Tin	10	1600	Gress	1310000

In [379]:

Out[379]:

	Race Number	Distance	Prize money	Starting position	Jockey weight	Horse age	Path	Final place
count	10.000000	10.000000	10.0	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
std	2.923088	175.119007	0.0	3.527668	2.149935	0.516398	1.581139	2.496664
min	1.000000	1400.000000	1310000.0	3.000000	52.000000	7.000000	0.000000	1.000000
25%	8.250000	1450.000000	1310000.0	6.000000	52.000000	7.000000	0.250000	3.000000
50%	9.000000	1600.000000	1310000.0	8.000000	53.000000	7.000000	1.000000	4.000000
75%	10.000000	1800.000000	1310000.0	9.000000	55.500000	8.000000	2.000000	6.000000
max	10.000000	1800.000000	1310000.0	14.000000	57.000000	8.000000	5.000000	9.000000

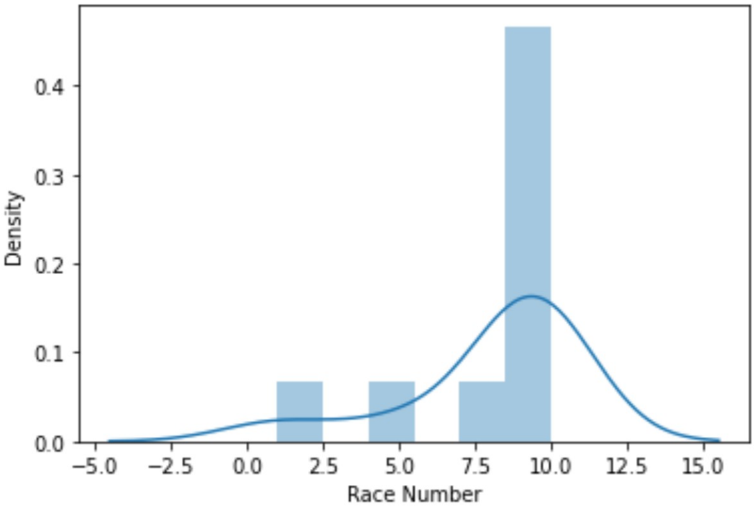
```
In [380]: sns.pairplot(d)

Out[380]: <seaborn.axisgrid.PairGrid at 0x190d13119d0>
```



In [381]:
```
sns.distplot(x1['Race Number'])
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\distributions.py:2557: Fut
ureWarning: `distplot` is a deprecated function and will be removed in a futu
re version. Please adapt your code to use either `displot` (a figure-level fu
nction with similar flexibility) or `histplot` (an axes-level function for hi
stograms).
  warnings.warn(msg, FutureWarning)
```

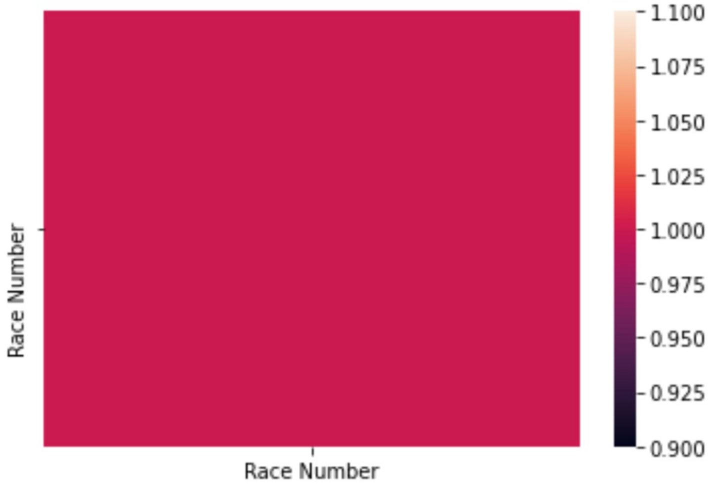
Out[381]:
```
<AxesSubplot:xlabel='Race Number', ylabel='Density'>
```



In [383]:
```
x1=x[['Race Number']]
```

In [384]:
```
sns.heatmap(x1.corr())
```

Out[384]:
```
<AxesSubplot:>
```



In [386]:
```
x=x1[['Race Number']]
y=x1['Race Number']
```

In [387]:

```
# to split my dataset into traning and test date

from sklearn.model_selection import train_test_split
```

In [388]:

```
from sklearn.linear_model import LinearRegression

lr=LinearRegression()
```

Out[388]:

```
LinearRegression()
```

In [389]:

```
-1.7763568394002505e-15
```

In [390]:

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

Out[390]:

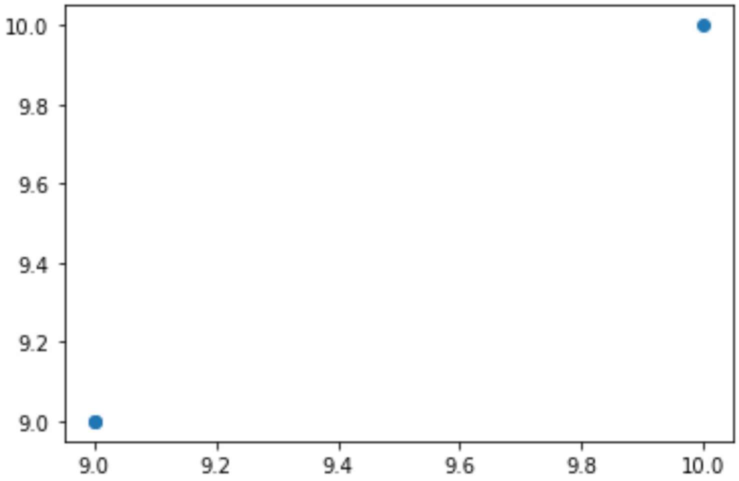
	Co-efficient
Race Number	1.0

In [391]:

```
prediction=lr.predict(x_test)
```

Out[391]:

```
<matplotlib.collections.PathCollection at 0x190d1ee5d90>
```



In [392]:

Out[392]:

```
1.0
```

In [393]:

Out[393]:

```
1.0
```

In [394]:

```
In [395]: rr=Ridge(alpha=10)
          rr.fit(x_train,y_train)

Out[395]: 0.7644236327899177
```

```
In [396]: la=Lasso(alpha=10)

Out[396]: Lasso(alpha=10)
```

```
In [397]:

Out[397]: -13.969387755102046
```

```
In [398]: from sklearn.linear_model import ElasticNet
          en=ElasticNet()

Out[398]: ElasticNet()
```

```
In [399]:

Out[399]: array([0.90439024])
```

```
In [400]:

Out[400]: array([8.86341463, 8.86341463, 9.76780488])
```

```
In [401]:

Out[401]: 0.72390243902439
```

```
In [402]:

Out[402]: 0.8631614515169552
```

```
In [403]:
```

```
In [404]:

          Mean Absolute Error 0.0
```

```
In [405]:

          Mean Squared Error 0.0
```

```
In [406]:

          Root Mean Squared Error 0.0
```

```
In [ ]:
```