In [1]:

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
```

In [2]:

```
startups_data = pd.read_csv('50_Startups.csv')
startups_data
```

Out[2]:

	R&D Spend	Administration	Marketing Spend	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83
1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	407934.54	Florida	191050.39
3	144372.41	118671.85	383199.62	New York	182901.99
4	142107.34	91391.77	366168.42	Florida	166187.94
5	131876.90	99814.71	362861.36	New York	156991.12
6	134615.46	147198.87	127716.82	California	156122.51
7	130298.13	145530.06	323876.68	Florida	155752.60
8	120542.52	148718.95	311613.29	New York	152211.77
9	123334.88	108679.17	304981.62	California	149759.96
10	101913.08	110594.11	229160.95	Florida	146121.95
11	100671.96	91790.61	249744.55	California	144259.40
12	93863.75	127320.38	249839.44	Florida	141585.52
13	91992.39	135495.07	252664.93	California	134307.35
14	119943.24	156547.42	256512.92	Florida	132602.65
15	114523.61	122616.84	261776.23	New York	129917.04
16	78013.11	121597.55	264346.06	California	126992.93
17	94657.16	145077.58	282574.31	New York	125370.37
18	91749.16	114175.79	294919.57	Florida	124266.90
19	86419.70	153514.11	0.00	New York	122776.86
20	76253.86	113867.30	298664.47	California	118474.03
21	78389.47	153773.43	299737.29	New York	111313.02
22	73994.56	122782.75	303319.26	Florida	110352.25
23	67532.53	105751.03	304768.73	Florida	108733.99
24	77044.01	99281.34	140574.81	New York	108552.04
25	64664.71	139553.16	137962.62	California	107404.34
26	75328.87	144135.98	134050.07	Florida	105733.54
27	72107.60	127864.55	353183.81	New York	105008.31
28	66051.52	182645.56	118148.20	Florida	103282.38
29	65605.48	153032.06	107138.38	New York	101004.64
30	61994.48	115641.28	91131.24	Florida	99937.59
31	61136.38	152701.92	88218.23	New York	97483.56
32	63408.86	129219.61	46085.25	California	97427.84
33	55493.95	103057.49	214634.81	Florida	96778.92

	R&D Spend	Administration	Marketing Spend	State	Profit
34	46426.07	157693.92	210797.67	California	96712.80
35	46014.02	85047.44	205517.64	New York	96479.51
36	28663.76	127056.21	201126.82	Florida	90708.19
37	44069.95	51283.14	197029.42	California	89949.14
38	20229.59	65947.93	185265.10	New York	81229.06
39	38558.51	82982.09	174999.30	California	81005.76
40	28754.33	118546.05	172795.67	California	78239.91
41	27892.92	84710.77	164470.71	Florida	77798.83
42	23640.93	96189.63	148001.11	California	71498.49
43	15505.73	127382.30	35534.17	New York	69758.98
44	22177.74	154806.14	28334.72	California	65200.33
45	1000.23	124153.04	1903.93	New York	64926.08
46	1315.46	115816.21	297114.46	Florida	49490.75
47	0.00	135426.92	0.00	California	42559.73
48	542.05	51743.15	0.00	New York	35673.41
49	0.00	116983.80	45173.06	California	14681.40

In [3]:

```
# initial analysis
```

In [4]:

```
startups_data.dtypes
```

Out[4]:

```
R&D Spend          float64
Administration     float64
Marketing Spend    float64
State               object
Profit             float64
dtype: object
```

In [5]:

```
startups_data.isna().sum()
```

Out[5]:

```
R&D Spend          0
Administration     0
Marketing Spend    0
State              0
Profit             0
dtype: int64
```

In [6]:

```
startups_data.shape
```

Out[6]:

```
(50, 5)
```

In [7]:

```
#Renaming the columns
startups_data = startups_data.rename({'R&D Spend' :'RDS', 'Administration':'Admin' ,'Market
startups_data
```

Out[7]:

	RDS	Admin	Mkts	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83
1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	407934.54	Florida	191050.39
3	144372.41	118671.85	383199.62	New York	182901.99
4	142107.34	91391.77	366168.42	Florida	166187.94
5	131876.90	99814.71	362861.36	New York	156991.12
6	134615.46	147198.87	127716.82	California	156122.51
7	130298.13	145530.06	323876.68	Florida	155752.60
8	120542.52	148718.95	311613.29	New York	152211.77
9	123334.88	108679.17	304981.62	California	149759.96
10	101913.08	110594.11	229160.95	Florida	146121.95
11	100671.96	91790.61	249744.55	California	144259.40
12	93863.75	127320.38	249839.44	Florida	141585.52
13	91992.39	135495.07	252664.93	California	134307.35
14	119943.24	156547.42	256512.92	Florida	132602.65
15	114523.61	122616.84	261776.23	New York	129917.04
16	78013.11	121597.55	264346.06	California	126992.93
17	94657.16	145077.58	282574.31	New York	125370.37
18	91749.16	114175.79	294919.57	Florida	124266.90
19	86419.70	153514.11	0.00	New York	122776.86
20	76253.86	113867.30	298664.47	California	118474.03
21	78389.47	153773.43	299737.29	New York	111313.02
22	73994.56	122782.75	303319.26	Florida	110352.25
23	67532.53	105751.03	304768.73	Florida	108733.99
24	77044.01	99281.34	140574.81	New York	108552.04
25	64664.71	139553.16	137962.62	California	107404.34
26	75328.87	144135.98	134050.07	Florida	105733.54
27	72107.60	127864.55	353183.81	New York	105008.31
28	66051.52	182645.56	118148.20	Florida	103282.38
29	65605.48	153032.06	107138.38	New York	101004.64
30	61994.48	115641.28	91131.24	Florida	99937.59
31	61136.38	152701.92	88218.23	New York	97483.56
32	63408.86	129219.61	46085.25	California	97427.84
33	55493.95	103057.49	214634.81	Florida	96778.92

	RDS	Admin	Mkts	State	Profit
34	46426.07	157693.92	210797.67	California	96712.80
35	46014.02	85047.44	205517.64	New York	96479.51
36	28663.76	127056.21	201126.82	Florida	90708.19
37	44069.95	51283.14	197029.42	California	89949.14
38	20229.59	65947.93	185265.10	New York	81229.06
39	38558.51	82982.09	174999.30	California	81005.76
40	28754.33	118546.05	172795.67	California	78239.91
41	27892.92	84710.77	164470.71	Florida	77798.83
42	23640.93	96189.63	148001.11	California	71498.49
43	15505.73	127382.30	35534.17	New York	69758.98
44	22177.74	154806.14	28334.72	California	65200.33
45	1000.23	124153.04	1903.93	New York	64926.08
46	1315.46	115816.21	297114.46	Florida	49490.75
47	0.00	135426.92	0.00	California	42559.73
48	542.05	51743.15	0.00	New York	35673.41
49	0.00	116983.80	45173.06	California	14681.40

In [8]:

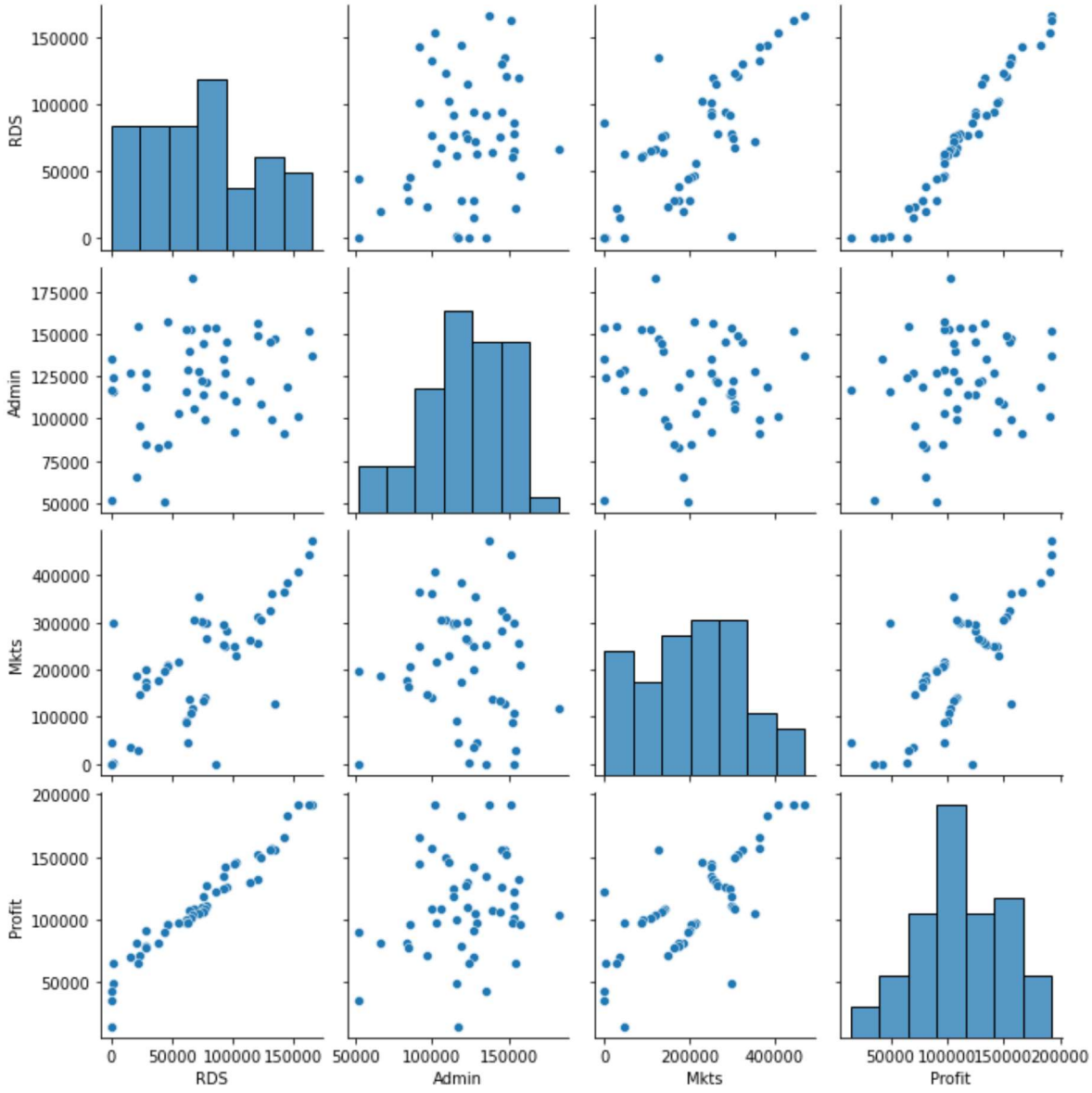
```
# Correlation check
startups_data.corr()
```

Out[8]:

	RDS	Admin	Mkts	Profit
RDS	1.000000	0.241955	0.724248	0.972900
Admin	0.241955	1.000000	-0.032154	0.200717
Mkts	0.724248	-0.032154	1.000000	0.747766
Profit	0.972900	0.200717	0.747766	1.000000

In [9]:

```
sns.pairplot(startups_data)
plt.show()
```



Model Building

In [10]:

```
import statsmodels.formula.api as smf
```

In [11]:

```
model=smf.ols('Profit~RDS+Admin+Mkts', data=startups_data).fit()
print('R2 score              :',model.rsquared)
print('Adjusted R2 score    : ',model.rsquared_adj)
print('AIC score             : ', model.aic)
print('BIC score             : ' , model.bic)
```

```
R2 score              : 0.9507459940683246
Adjusted R2 score    :  0.9475337762901719
AIC score             :  1058.7714985998055
BIC score             :  1066.419590621518
```

In [12]:

```
model.summary()
```

Out[12]:

OLS Regression Results

Dep. Variable:	Profit	**R-squared:**	0.951
Model:	OLS	**Adj. R-squared:**	0.948
Method:	Least Squares	**F-statistic:**	296.0
Date:	Sun, 24 Oct 2021	**Prob (F-statistic):**	4.53e-30
Time:	11:44:05	**Log-Likelihood:**	-525.39
No. Observations:	50	**AIC:**	1059.
Df Residuals:	46	**BIC:**	1066.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	5.012e+04	6572.353	7.626	0.000	3.69e+04	6.34e+04
RDS	0.8057	0.045	17.846	0.000	0.715	0.897
Admin	-0.0268	0.051	-0.526	0.602	-0.130	0.076
Mkts	0.0272	0.016	1.655	0.105	-0.006	0.060

Omnibus:	14.838	**Durbin-Watson:**	1.282
Prob(Omnibus):	0.001	**Jarque-Bera (JB):**	21.442
Skew:	-0.949	**Prob(JB):**	2.21e-05
Kurtosis:	5.586	**Cond. No.**	1.40e+06

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.4e+06. This might indicate that there are
strong multicollinearity or other numerical problems.

Model Testing

In [13]:

```
# finding Coefficient parameters
model.params
```

Out[13]:

```
Intercept    50122.192990
RDS              0.805715
Admin           -0.026816
Mkts             0.027228
dtype: float64
```

In [14]:

```
# finding tvalues and pvalues
model.tvalues , model.pvalues
```

Out[14]:

```
(Intercept     7.626218
 RDS          17.846374
 Admin        -0.525507
 Mkts          1.655077
 dtype: float64,
 Intercept    1.057379e-09
 RDS          2.634968e-22
 Admin        6.017551e-01
 Mkts         1.047168e-01
 dtype: float64)
```

In [15]:

```
model_1=smf.ols('Profit~Admin', data=startups_data).fit()
model_1.tvalues , model_1.pvalues # admin has in-significant pvalue
```

Out[15]:

```
(Intercept    3.040044
 Admin        1.419493
 dtype: float64,
 Intercept    0.003824
 Admin        0.162217
 dtype: float64)
```

In [16]:

```
model_2=smf.ols('Profit~Mkts', data=startups_data).fit()
model_2.tvalues , model_2.pvalues # mkts has significant pvalue
```

Out[16]:

```
(Intercept    7.808356
 Mkts         7.802657
 dtype: float64,
 Intercept    4.294735e-10
 Mkts         4.381073e-10
 dtype: float64)
```

In [17]:

```
model_3=smf.ols('Profit~Admin+Mkts', data=startups_data).fit()
model_3.tvalues , model_3.pvalues #admin+mkts has significant pvalues
```

Out[17]:

```
(Intercept    1.142741
 Admin        2.467779
 Mkts         8.281039
 dtype: float64,
 Intercept    2.589341e-01
 Admin        1.729198e-02
 Mkts         9.727245e-11
 dtype: float64)
```

Understanding R2 and Adjusted R2

In [18]:

```
model_4 =  smf.ols('Profit~RDS', data = startups_data).fit()
print('R2 score              :',model_4.rsquared)
print('Adjusted R2 score    : ',model_4.rsquared_adj)
print('AIC score             : ', model_4.aic)
print('BIC score             : ' , model_4.bic)
```

```
R2 score              : 0.9465353160804393
Adjusted R2 score    :  0.9454214684987817
AIC score             :  1058.8730295624773
BIC score             :  1062.6970755733337
```

In [19]:

```
model_5 =  smf.ols('Profit~RDS+Admin', data = startups_data).fit()
print('R2 score              :',model_5.rsquared)
print('Adjusted R2 score    : ',model_5.rsquared_adj)
print('AIC score             : ', model_5.aic)
print('BIC score             : ' , model_5.bic)
```

```
R2 score              : 0.9478129385009173
Adjusted R2 score    :  0.9455922124796797
AIC score             :  1059.6636934567293
BIC score             :  1065.3997624730139
```

In [20]:

```
model_6 =  smf.ols('Profit~Mkts+RDS+Admin', data = startups_data).fit()
print('R2 score              :',model_6.rsquared)
print('Adjusted R2 score    : ',model_6.rsquared_adj)
print('AIC score             : ', model_6.aic)
print('BIC score             : ' , model_6.bic)
```

```
R2 score              : 0.9507459940683246
Adjusted R2 score    :  0.9475337762901719
AIC score             :  1058.7714985998052
BIC score             :  1066.4195906215177
```

In [21]:

```
model_4.summary()
```

Out[21]:

OLS Regression Results

Dep. Variable:	Profit	**R-squared:**	0.947
Model:	OLS	**Adj. R-squared:**	0.945
Method:	Least Squares	**F-statistic:**	849.8
Date:	Sun, 24 Oct 2021	**Prob (F-statistic):**	3.50e-32
Time:	11:44:06	**Log-Likelihood:**	-527.44
No. Observations:	50	**AIC:**	1059.
Df Residuals:	48	**BIC:**	1063.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	4.903e+04	2537.897	19.320	0.000	4.39e+04	5.41e+04
RDS	0.8543	0.029	29.151	0.000	0.795	0.913

Omnibus:	13.727	**Durbin-Watson:**	1.116
Prob(Omnibus):	0.001	**Jarque-Bera (JB):**	18.536
Skew:	-0.911	**Prob(JB):**	9.44e-05
Kurtosis:	5.361	**Cond. No.**	1.65e+05

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.65e+05. This might indicate that there are
strong multicollinearity or other numerical problems.

In [22]:

```
model_5.summary()
```

Out[22]:

OLS Regression Results

Dep. Variable:	Profit	**R-squared:**	0.948
Model:	OLS	**Adj. R-squared:**	0.946
Method:	Least Squares	**F-statistic:**	426.8
Date:	Sun, 24 Oct 2021	**Prob (F-statistic):**	7.29e-31
Time:	11:44:06	**Log-Likelihood:**	-526.83
No. Observations:	50	**AIC:**	1060.
Df Residuals:	47	**BIC:**	1065.
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	5.489e+04	6016.718	9.122	0.000	4.28e+04	6.7e+04
RDS	0.8621	0.030	28.589	0.000	0.801	0.923
Admin	-0.0530	0.049	-1.073	0.289	-0.152	0.046

Omnibus:	14.678	**Durbin-Watson:**	1.189
Prob(Omnibus):	0.001	**Jarque-Bera (JB):**	20.449
Skew:	-0.961	**Prob(JB):**	3.63e-05
Kurtosis:	5.474	**Cond. No.**	6.65e+05

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 6.65e+05. This might indicate that there are
strong multicollinearity or other numerical problems.

In [23]:

```
model_6.summary()
```

Out[23]:

OLS Regression Results

Dep. Variable:	Profit	**R-squared:**	0.951
Model:	OLS	**Adj. R-squared:**	0.948
Method:	Least Squares	**F-statistic:**	296.0
Date:	Sun, 24 Oct 2021	**Prob (F-statistic):**	4.53e-30
Time:	11:44:07	**Log-Likelihood:**	-525.39
No. Observations:	50	**AIC:**	1059.
Df Residuals:	46	**BIC:**	1066.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	5.012e+04	6572.353	7.626	0.000	3.69e+04	6.34e+04
Mkts	0.0272	0.016	1.655	0.105	-0.006	0.060
RDS	0.8057	0.045	17.846	0.000	0.715	0.897
Admin	-0.0268	0.051	-0.526	0.602	-0.130	0.076

Omnibus:	14.838	**Durbin-Watson:**	1.282
Prob(Omnibus):	0.001	**Jarque-Bera (JB):**	21.442
Skew:	-0.949	**Prob(JB):**	2.21e-05
Kurtosis:	5.586	**Cond. No.**	1.40e+06

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.4e+06. This might indicate that there are
strong multicollinearity or other numerical problems.

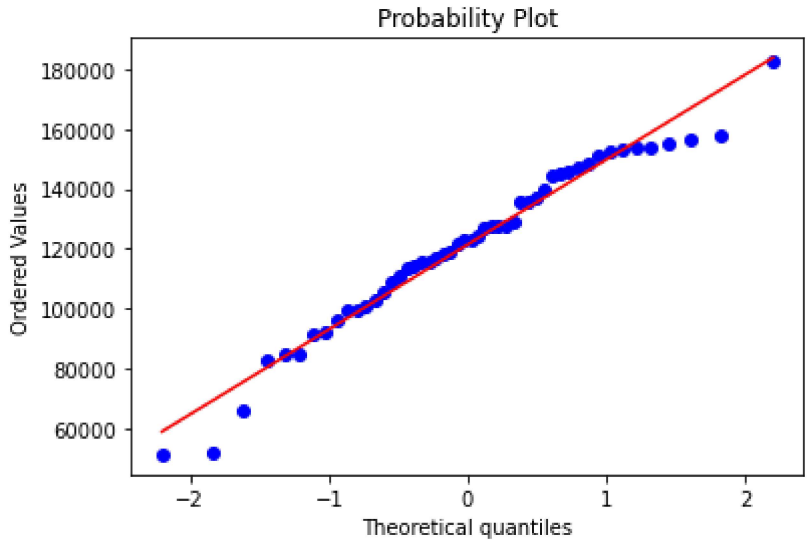
Normality test using QQPlot.

In [24]:

```
from scipy import stats
```

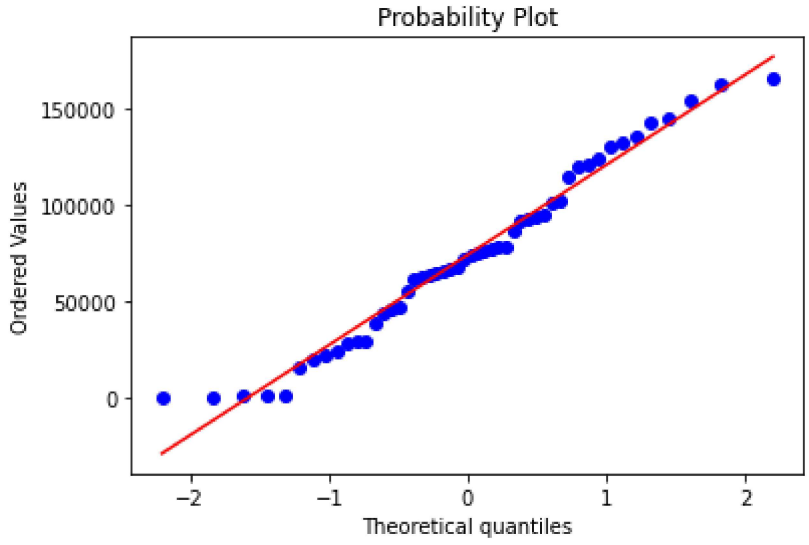
In [25]:

```
stats.probplot(x= startups_data['Admin'] , dist ='norm', plot=plt)
plt.show()
```



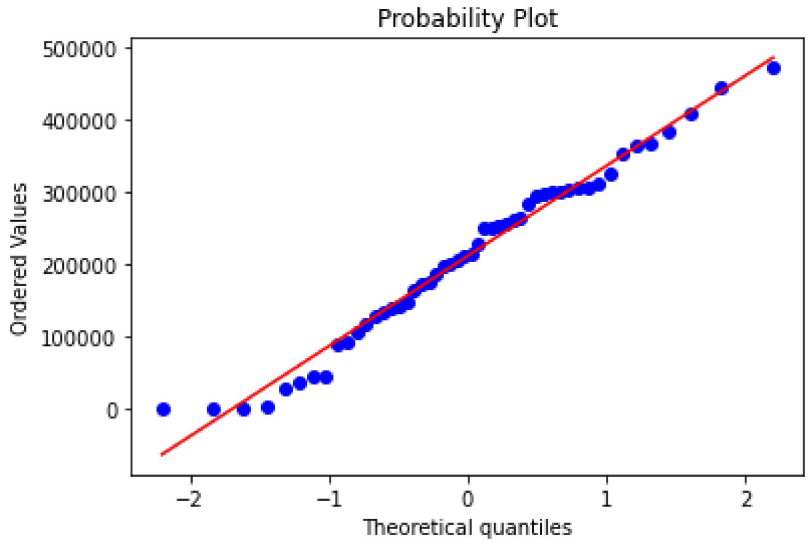
In [26]:

```
stats.probplot(x= startups_data['RDS'] , dist ='norm', plot=plt)
plt.show()
```



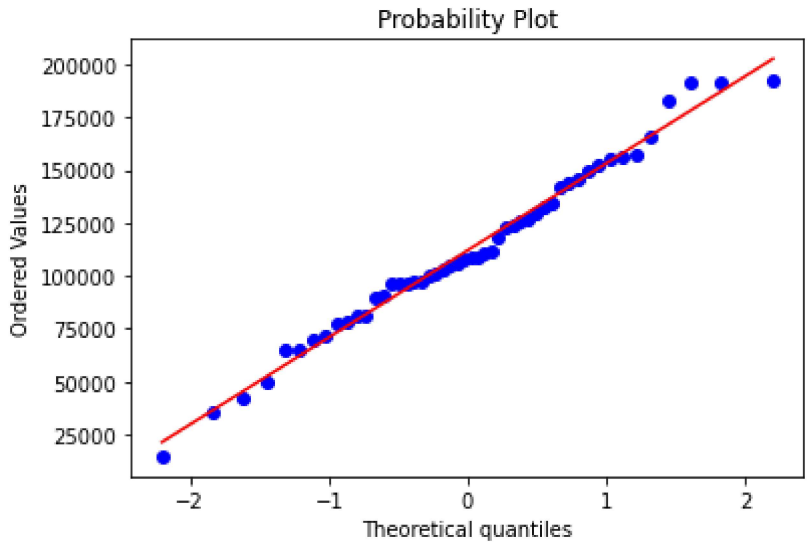
In [27]:

```
stats.probplot(x= startups_data['Mkts'] , dist ='norm', plot=plt)
plt.show()
```



In [28]:

```
stats.probplot(x= startups_data['Profit'] , dist ='norm', plot=plt)
plt.show()
```



Model Improvement Techniques

Log Transformation

In [29]:

```
import warnings
warnings.filterwarnings('ignore')
```

In [140]:

```
startups_data['log_RDS'] = np.log(startups_data['RDS'])
startups_data['log_Mkts'] = np.log(startups_data['Mkts'])
startups_data['log_Admin'] = np.log(startups_data['Admin'])
startups_data
```

Out[140]:

	RDS	Admin	Mkts	State	Profit	log_RDS	log_Mkts	log_Admin
0	165349.20	136897.80	471784.10	New York	192261.83	12.015815	13.064277	11.826990
1	162597.70	151377.59	443898.53	California	191792.06	11.999034	13.003351	11.927533
2	153441.51	101145.55	407934.54	Florida	191050.39	11.941075	12.918862	11.524316
3	144372.41	118671.85	383199.62	New York	182901.99	11.880151	12.856311	11.684117
4	142107.34	91391.77	366168.42	Florida	166187.94	11.864338	12.810849	11.422911
5	131876.90	99814.71	362861.36	New York	156991.12	11.789624	12.801776	11.511071
6	134615.46	147198.87	127716.82	California	156122.51	11.810178	11.757571	11.899540
7	130298.13	145530.06	323876.68	Florida	155752.60	11.777580	12.688118	11.888138
8	120542.52	148718.95	311613.29	New York	152211.77	11.699758	12.649518	11.909814
9	123334.88	108679.17	304981.62	California	149759.96	11.722659	12.628007	11.596155
10	101913.08	110594.11	229160.95	Florida	146121.95	11.531876	12.342180	11.613622

In [68]:

```
startups_data.drop(axis=0, index=[19,47,48,49], inplace= True)
```

In [69]:

```
startups_data
```

Out[69]:

	RDS	Admin	Mkts	State	Profit	log_RDS	log_Mkts	log_Admin
0	165349.20	136897.80	471784.10	New York	192261.83	12.015815	13.064277	11.826990
1	162597.70	151377.59	443898.53	California	191792.06	11.999034	13.003351	11.927533
2	153441.51	101145.55	407934.54	Florida	191050.39	11.941075	12.918862	11.524316
3	144372.41	118671.85	383199.62	New York	182901.99	11.880151	12.856311	11.684117
4	142107.34	91391.77	366168.42	Florida	166187.94	11.864338	12.810849	11.422911
5	131876.90	99814.71	362861.36	New York	156991.12	11.789624	12.801776	11.511071
6	134615.46	147198.87	127716.82	California	156122.51	11.810178	11.757571	11.899540
7	130298.13	145530.06	323876.68	Florida	155752.60	11.777580	12.688118	11.888138
8	120542.52	148718.95	311613.29	New York	152211.77	11.699758	12.649518	11.909814
9	123334.88	108679.17	304981.62	California	149759.96	11.722659	12.628007	11.596155
10	101913.08	110594.11	229160.95	Florida	146121.95	11.531876	12.342180	11.613622
11	100671.96	91790.61	249744.55	California	144259.40	11.519623	12.428194	11.427265
12	93863.75	127320.38	249839.44	Florida	141585.52	11.449600	12.428574	11.754462
13	91992.39	135495.07	252664.93	California	134307.35	11.429461	12.439820	11.816691
14	119943.24	156547.42	256512.92	Florida	132602.65	11.694774	12.454934	11.961114
15	114523.61	122616.84	261776.23	New York	129917.04	11.648536	12.475245	11.716820
16	78013.11	121597.55	264346.06	California	126992.93	11.264632	12.485014	11.708472
17	94657.16	145077.58	282574.31	New York	125370.37	11.458017	12.551697	11.885024
18	91749.16	114175.79	294919.57	Florida	124266.90	11.426814	12.594458	11.645495
20	76253.86	113867.30	298664.47	California	118474.03	11.241823	12.607076	11.642789
21	78389.47	153773.43	299737.29	New York	111313.02	11.269445	12.610662	11.943236
22	73994.56	122782.75	303319.26	Florida	110352.25	11.211747	12.622541	11.718172
23	67532.53	105751.03	304768.73	Florida	108733.99	11.120365	12.627309	11.568843
24	77044.01	99281.34	140574.81	New York	108552.04	11.252132	11.853495	11.505713
25	64664.71	139553.16	137962.62	California	107404.34	11.076971	11.834738	11.846201
26	75328.87	144135.98	134050.07	Florida	105733.54	11.229619	11.805969	11.878512
27	72107.60	127864.55	353183.81	New York	105008.31	11.185915	12.774744	11.758727
28	66051.52	182645.56	118148.20	Florida	103282.38	11.098190	11.679695	12.115303
29	65605.48	153032.06	107138.38	New York	101004.64	11.091415	11.581877	11.938403
30	61994.48	115641.28	91131.24	Florida	99937.59	11.034801	11.420056	11.658248
31	61136.38	152701.92	88218.23	New York	97483.56	11.020862	11.387569	11.936243
32	63408.86	129219.61	46085.25	California	97427.84	11.057359	10.738248	11.769269
33	55493.95	103057.49	214634.81	Florida	96778.92	10.924029	12.276693	11.543042
34	46426.07	157693.92	210797.67	California	96712.80	10.745616	12.258654	11.968411
35	46014.02	85047.44	205517.64	New York	96479.51	10.736701	12.233287	11.350964

	RDS	Admin	Mkts	State	Profit	log_RDS	log_Mkts	log_Admin
36	28663.76	127056.21	201126.82	Florida	90708.19	10.263389	12.211691	11.752385
37	44069.95	51283.14	197029.42	California	89949.14	10.693533	12.191108	10.845117
38	20229.59	65947.93	185265.10	New York	81229.06	9.914902	12.129543	11.096621
39	38558.51	82982.09	174999.30	California	81005.76	10.559932	12.072537	11.326380
40	28754.33	118546.05	172795.67	California	78239.91	10.266544	12.059865	11.683057
41	27892.92	84710.77	164470.71	Florida	77798.83	10.236128	12.010488	11.346998
42	23640.93	96189.63	148001.11	California	71498.49	10.070735	11.904975	11.474077
43	15505.73	127382.30	35534.17	New York	69758.98	9.648965	10.478250	11.754948
44	22177.74	154806.14	28334.72	California	65200.33	10.006844	10.251843	11.949929
45	1000.23	124153.04	1903.93	New York	64926.08	6.907985	7.551675	11.729270
46	1315.46	115816.21	297114.46	Florida	49490.75	7.181942	12.601873	11.659760

In [117]:

```
startups_data_log = startups_data.loc[:,['log_RDS','log_Admin','log_Mkts','Profit']]
startups_data_log
```

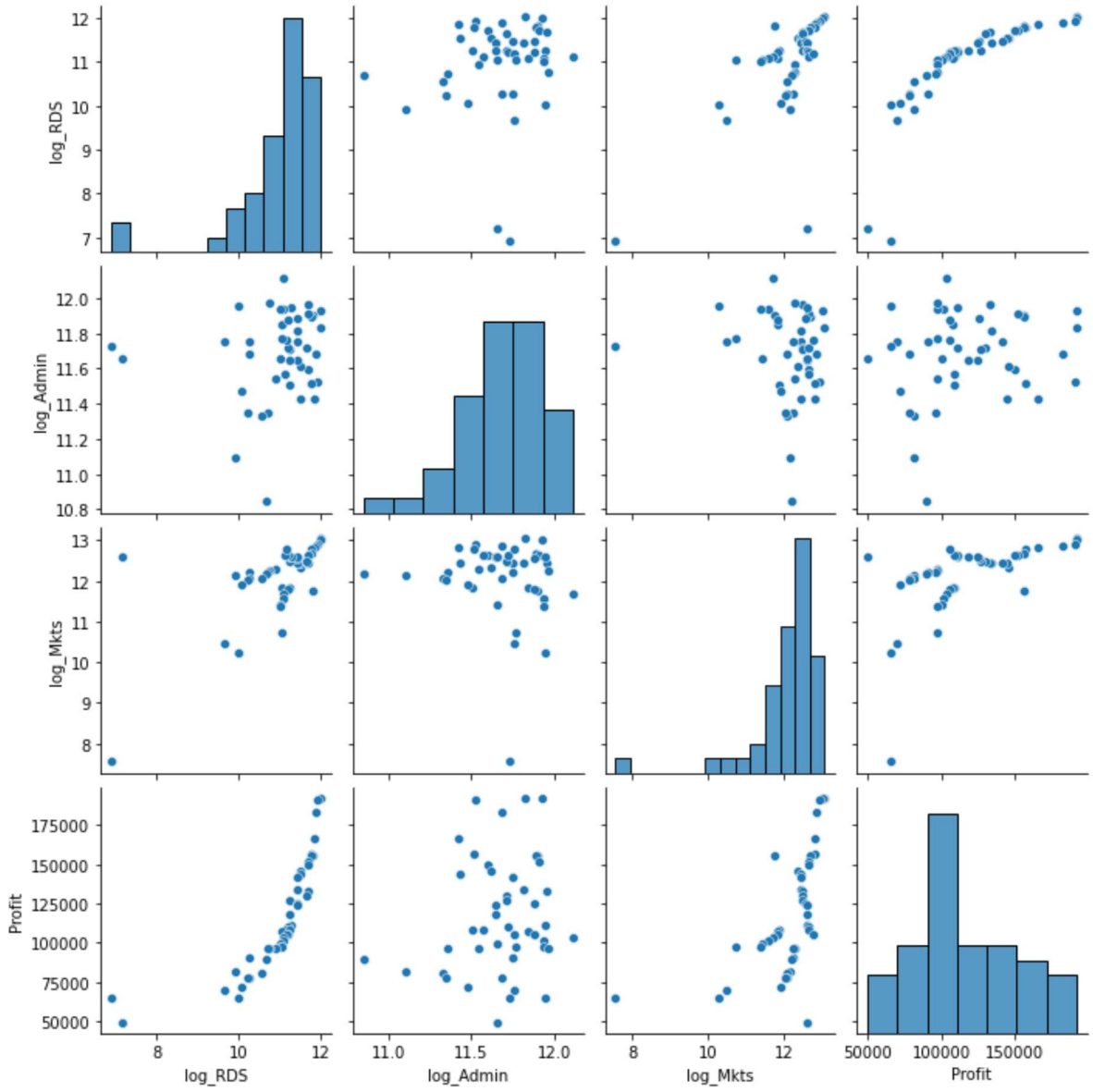
Out[117]:

	log_RDS	log_Admin	log_Mkts	Profit
0	12.015815	11.826990	13.064277	192261.83
1	11.999034	11.927533	13.003351	191792.06
2	11.941075	11.524316	12.918862	191050.39
3	11.880151	11.684117	12.856311	182901.99
4	11.864338	11.422911	12.810849	166187.94
5	11.789624	11.511071	12.801776	156991.12
6	11.810178	11.899540	11.757571	156122.51
7	11.777580	11.888138	12.688118	155752.60
8	11.699758	11.909814	12.649518	152211.77
9	11.722659	11.596155	12.628007	149759.96
10	11.531876	11.613622	12.342180	146121.95
11	11.519623	11.427265	12.428194	144259.40
12	11.449600	11.754462	12.428574	141585.52
13	11.429461	11.816691	12.439820	134307.35
14	11.694774	11.961114	12.454934	132602.65
15	11.648536	11.716820	12.475245	129917.04
16	11.264632	11.708472	12.485014	126992.93
17	11.458017	11.885024	12.551697	125370.37
18	11.426814	11.645495	12.594458	124266.90
20	11.241823	11.642789	12.607076	118474.03
21	11.269445	11.943236	12.610662	111313.02
22	11.211747	11.718172	12.622541	110352.25
23	11.120365	11.568843	12.627309	108733.99
24	11.252132	11.505713	11.853495	108552.04
25	11.076971	11.846201	11.834738	107404.34
26	11.229619	11.878512	11.805969	105733.54
27	11.185915	11.758727	12.774744	105008.31
28	11.098190	12.115303	11.679695	103282.38
29	11.091415	11.938403	11.581877	101004.64
30	11.034801	11.658248	11.420056	99937.59
31	11.020862	11.936243	11.387569	97483.56
32	11.057359	11.769269	10.738248	97427.84
33	10.924029	11.543042	12.276693	96778.92
34	10.745616	11.968411	12.258654	96712.80

	log_RDS	log_Admin	log_Mkts	Profit
35	10.736701	11.350964	12.233287	96479.51
36	10.263389	11.752385	12.211691	90708.19
37	10.693533	10.845117	12.191108	89949.14
38	9.914902	11.096621	12.129543	81229.06
39	10.559932	11.326380	12.072537	81005.76
40	10.266544	11.683057	12.059865	78239.91
41	10.236128	11.346998	12.010488	77798.83
42	10.070735	11.474077	11.904975	71498.49
43	9.648965	11.754948	10.478250	69758.98
44	10.006844	11.949929	10.251843	65200.33
45	6.907985	11.729270	7.551675	64926.08
46	7.181942	11.659760	12.601873	49490.75

In [118]:

```
sns.pairplot(startups_data_log)
plt.show()
```



Understanding R2 and Adjusted R2

In [119]:

```
model_7 =  smf.ols('Profit~log_RDS', data = startups_data_log).fit()
print('R2 score              :',model_7.rsquared)
print('Adjusted R2 score    : ',model_7.rsquared_adj)
print('AIC score             : ', model_7.aic)
print('BIC score             : ' , model_7.bic)
```

```
R2 score              : 0.6066133040418662
Adjusted R2 score    :  0.5976726973155451
AIC score             :  1055.7100136373704
BIC score             :  1059.3672964303485
```

In [120]:

```
model_8 =  smf.ols('Profit~log_RDS+log_Admin', data = startups_data_log).fit()
print('R2 score              :',model_8.rsquared)
print('Adjusted R2 score    : ',model_8.rsquared_adj)
print('AIC score             : ', model_8.aic)
print('BIC score             : ' , model_8.bic)
```

```
R2 score              : 0.6086660173345844
Adjusted R2 score    :  0.5904644367454952
AIC score             :  1057.4693546884955
BIC score             :  1062.9552788779629
```

In [121]:

```
model_9 =  smf.ols('Profit~log_RDS+log_Admin+log_Mkts', data = startups_data_log).fit()
print('R2 score              :',model_9.rsquared)
print('Adjusted R2 score    : ',model_9.rsquared_adj)
print('AIC score             : ', model_9.aic)
print('BIC score             : ' , model_9.bic)
```

```
R2 score              : 0.6167542135106918
Adjusted R2 score    :  0.5893795144757412
AIC score             :  1058.508651705086
BIC score             :  1065.8232172910425
```

In [76]:

```
model_7.summary()
```

Out[76]:

OLS Regression Results

Dep. Variable:	Profit	**R-squared:**	0.607
Model:	OLS	**Adj. R-squared:**	0.598
Method:	Least Squares	**F-statistic:**	67.85
Date:	Sun, 24 Oct 2021	**Prob (F-statistic):**	1.85e-10
Time:	12:16:42	**Log-Likelihood:**	-525.86
No. Observations:	46	**AIC:**	1056.
Df Residuals:	44	**BIC:**	1059.
Df Model:	1		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-1.799e+05	3.62e+04	-4.968	0.000	-2.53e+05	-1.07e+05
log_RDS	2.705e+04	3283.617	8.237	0.000	2.04e+04	3.37e+04

Omnibus:	8.269	**Durbin-Watson:**	0.405
Prob(Omnibus):	0.016	**Jarque-Bera (JB):**	8.304
Skew:	1.041	**Prob(JB):**	0.0157
Kurtosis:	3.026	**Cond. No.**	120.

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

In [77]:

```
model_8.summary()
```

Out[77]:

OLS Regression Results

Dep. Variable:	Profit	**R-squared:**	0.609
Model:	OLS	**Adj. R-squared:**	0.590
Method:	Least Squares	**F-statistic:**	33.44
Date:	Sun, 24 Oct 2021	**Prob (F-statistic):**	1.74e-09
Time:	12:16:50	**Log-Likelihood:**	-525.73
No. Observations:	46	**AIC:**	1057.
Df Residuals:	43	**BIC:**	1063.
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-2.558e+05	1.64e+05	-1.559	0.126	-5.87e+05	7.51e+04
log_RDS	2.681e+04	3350.287	8.002	0.000	2.01e+04	3.36e+04
log_Admin	6726.2460	1.42e+04	0.475	0.637	-2.18e+04	3.53e+04

Omnibus:	7.894	**Durbin-Watson:**	0.409
Prob(Omnibus):	0.019	**Jarque-Bera (JB):**	7.927
Skew:	1.017	**Prob(JB):**	0.0190
Kurtosis:	2.967	**Cond. No.**	781.

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

In [78]:

```
model_9.summary()
```

Out[78]:

OLS Regression Results

Dep. Variable:	Profit	**R-squared:**	0.617
Model:	OLS	**Adj. R-squared:**	0.589
Method:	Least Squares	**F-statistic:**	22.53
Date:	Sun, 24 Oct 2021	**Prob (F-statistic):**	7.50e-09
Time:	12:16:53	**Log-Likelihood:**	-525.25
No. Observations:	46	**AIC:**	1059.
Df Residuals:	42	**BIC:**	1066.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-3.28e+05	1.81e+05	-1.809	0.078	-6.94e+05	3.79e+04
log_RDS	2.388e+04	4578.403	5.215	0.000	1.46e+04	3.31e+04
log_Admin	1.067e+04	1.48e+04	0.722	0.474	-1.92e+04	4.05e+04
log_Mkts	4809.2857	5108.211	0.941	0.352	-5499.500	1.51e+04

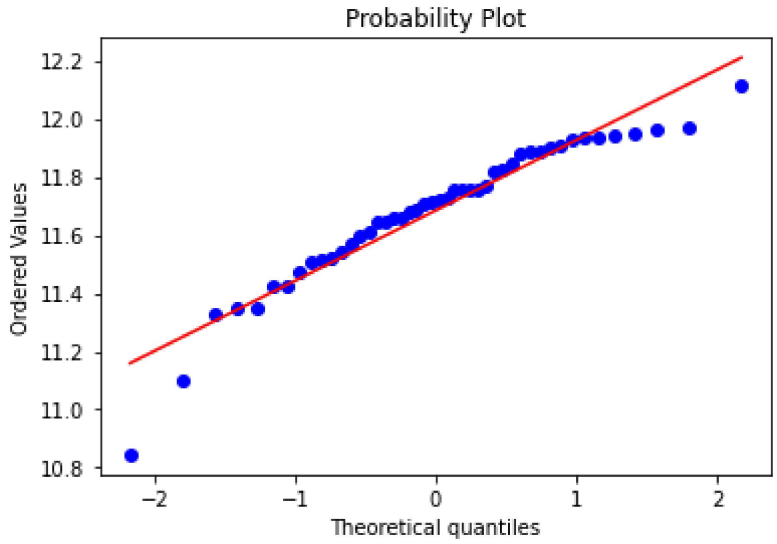
Omnibus:	11.369	**Durbin-Watson:**	0.524
Prob(Omnibus):	0.003	**Jarque-Bera (JB):**	11.469
Skew:	1.185	**Prob(JB):**	0.00323
Kurtosis:	3.604	**Cond. No.**	1.08e+03

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.08e+03. This might indicate that there are
strong multicollinearity or other numerical problems.

Normality Test using QQPlot

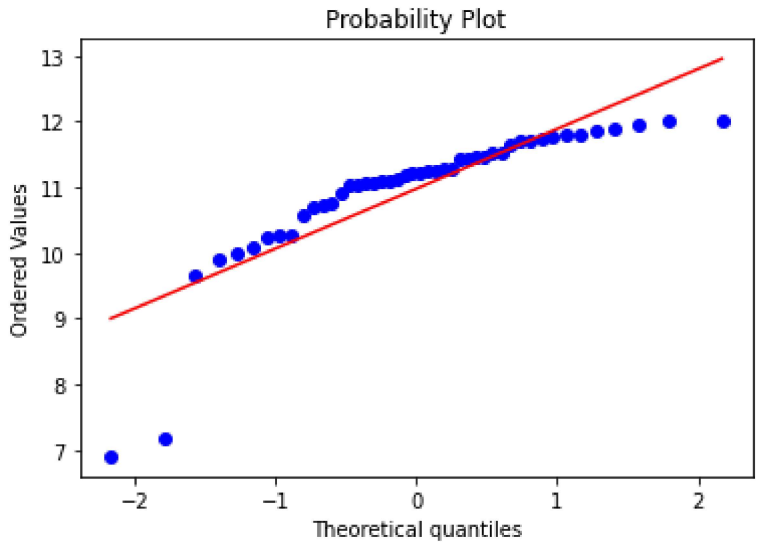
In [122]:

```
stats.probplot(x= startups_data_log['log_Admin'] , dist ='norm', plot=plt)
plt.show()
```



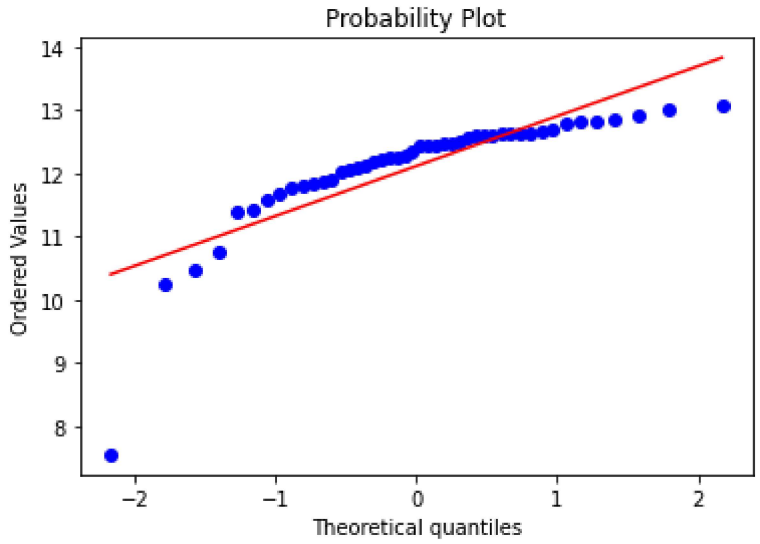
In [123]:

```
stats.probplot(x= startups_data_log['log_RDS'] , dist ='norm', plot=plt)
plt.show()
```



In [124]:

```
stats.probplot(x= startups_data['log_Mkts'] , dist ='norm', plot=plt)
plt.show()
```



Using sklearn library

In [82]:

```
startups_2 = startups_data.iloc[:,:5]
startups_2
```

Out[82]:

	RDS	Admin	Mkts	State	Profit
0	165349.20	136897.80	471784.10	New York	192261.83
1	162597.70	151377.59	443898.53	California	191792.06
2	153441.51	101145.55	407934.54	Florida	191050.39
3	144372.41	118671.85	383199.62	New York	182901.99
4	142107.34	91391.77	366168.42	Florida	166187.94
5	131876.90	99814.71	362861.36	New York	156991.12
6	134615.46	147198.87	127716.82	California	156122.51
7	130298.13	145530.06	323876.68	Florida	155752.60
8	120542.52	148718.95	311613.29	New York	152211.77
9	123334.88	108679.17	304981.62	California	149759.96
10	101913.08	110594.11	229160.95	Florida	146121.95
11	100671.96	91790.61	249744.55	California	144259.40
12	93863.75	127320.38	249839.44	Florida	141585.52
13	91992.39	135495.07	252664.93	California	134307.35
14	119943.24	156547.42	256512.92	Florida	132602.65
15	114523.61	122616.84	261776.23	New York	129917.04
16	78013.11	121597.55	264346.06	California	126992.93
17	94657.16	145077.58	282574.31	New York	125370.37
18	91749.16	114175.79	294919.57	Florida	124266.90
20	76253.86	113867.30	298664.47	California	118474.03
21	78389.47	153773.43	299737.29	New York	111313.02
22	73994.56	122782.75	303319.26	Florida	110352.25
23	67532.53	105751.03	304768.73	Florida	108733.99
24	77044.01	99281.34	140574.81	New York	108552.04
25	64664.71	139553.16	137962.62	California	107404.34
26	75328.87	144135.98	134050.07	Florida	105733.54
27	72107.60	127864.55	353183.81	New York	105008.31
28	66051.52	182645.56	118148.20	Florida	103282.38
29	65605.48	153032.06	107138.38	New York	101004.64
30	61994.48	115641.28	91131.24	Florida	99937.59
31	61136.38	152701.92	88218.23	New York	97483.56
32	63408.86	129219.61	46085.25	California	97427.84
33	55493.95	103057.49	214634.81	Florida	96778.92
34	46426.07	157693.92	210797.67	California	96712.80

	RDS	Admin	Mkts	State	Profit
35	46014.02	85047.44	205517.64	New York	96479.51
36	28663.76	127056.21	201126.82	Florida	90708.19
37	44069.95	51283.14	197029.42	California	89949.14
38	20229.59	65947.93	185265.10	New York	81229.06
39	38558.51	82982.09	174999.30	California	81005.76
40	28754.33	118546.05	172795.67	California	78239.91
41	27892.92	84710.77	164470.71	Florida	77798.83
42	23640.93	96189.63	148001.11	California	71498.49
43	15505.73	127382.30	35534.17	New York	69758.98
44	22177.74	154806.14	28334.72	California	65200.33
45	1000.23	124153.04	1903.93	New York	64926.08
46	1315.46	115816.21	297114.46	Florida	49490.75

In [88]:

```
startups_2 = startups_2.drop(labels='State', axis=1)
startups_2
```

Out[88]:

	RDS	Admin	Mkts	Profit
0	165349.20	136897.80	471784.10	192261.83
1	162597.70	151377.59	443898.53	191792.06
2	153441.51	101145.55	407934.54	191050.39
3	144372.41	118671.85	383199.62	182901.99
4	142107.34	91391.77	366168.42	166187.94
5	131876.90	99814.71	362861.36	156991.12
6	134615.46	147198.87	127716.82	156122.51
7	130298.13	145530.06	323876.68	155752.60
8	120542.52	148718.95	311613.29	152211.77
9	123334.88	108679.17	304981.62	149759.96
10	101913.08	110594.11	229160.95	146121.95
11	100671.96	91790.61	249744.55	144259.40
12	93863.75	127320.38	249839.44	141585.52
13	91992.39	135495.07	252664.93	134307.35
14	119943.24	156547.42	256512.92	132602.65
15	114523.61	122616.84	261776.23	129917.04
16	78013.11	121597.55	264346.06	126992.93
17	94657.16	145077.58	282574.31	125370.37
18	91749.16	114175.79	294919.57	124266.90
20	76253.86	113867.30	298664.47	118474.03
21	78389.47	153773.43	299737.29	111313.02
22	73994.56	122782.75	303319.26	110352.25
23	67532.53	105751.03	304768.73	108733.99
24	77044.01	99281.34	140574.81	108552.04
25	64664.71	139553.16	137962.62	107404.34
26	75328.87	144135.98	134050.07	105733.54
27	72107.60	127864.55	353183.81	105008.31
28	66051.52	182645.56	118148.20	103282.38
29	65605.48	153032.06	107138.38	101004.64
30	61994.48	115641.28	91131.24	99937.59
31	61136.38	152701.92	88218.23	97483.56
32	63408.86	129219.61	46085.25	97427.84
33	55493.95	103057.49	214634.81	96778.92
34	46426.07	157693.92	210797.67	96712.80

	RDS	Admin	Mkts	Profit
35	46014.02	85047.44	205517.64	96479.51
36	28663.76	127056.21	201126.82	90708.19
37	44069.95	51283.14	197029.42	89949.14
38	20229.59	65947.93	185265.10	81229.06
39	38558.51	82982.09	174999.30	81005.76
40	28754.33	118546.05	172795.67	78239.91
41	27892.92	84710.77	164470.71	77798.83
42	23640.93	96189.63	148001.11	71498.49
43	15505.73	127382.30	35534.17	69758.98
44	22177.74	154806.14	28334.72	65200.33
45	1000.23	124153.04	1903.93	64926.08
46	1315.46	115816.21	297114.46	49490.75

In [90]:

```
X = startups_2.drop(labels='Profit', axis=1)
y = startups_2[['Profit']]
```

In [91]:

```
X
```

Out[91]:

	RDS	Admin	Mkts
0	165349.20	136897.80	471784.10
1	162597.70	151377.59	443898.53
2	153441.51	101145.55	407934.54
3	144372.41	118671.85	383199.62
4	142107.34	91391.77	366168.42
5	131876.90	99814.71	362861.36
6	134615.46	147198.87	127716.82
7	130298.13	145530.06	323876.68
8	120542.52	148718.95	311613.29
9	123334.88	108679.17	304981.62
10	101913.08	110594.11	229160.95
11	100671.96	91790.61	249744.55
12	93863.75	127320.38	249839.44
13	91992.39	135495.07	252664.93
14	119943.24	156547.42	256512.92
15	114523.61	122616.84	261776.23
16	78013.11	121597.55	264346.06
17	94657.16	145077.58	282574.31
18	91749.16	114175.79	294919.57
20	76253.86	113867.30	298664.47
21	78389.47	153773.43	299737.29
22	73994.56	122782.75	303319.26
23	67532.53	105751.03	304768.73
24	77044.01	99281.34	140574.81
25	64664.71	139553.16	137962.62
26	75328.87	144135.98	134050.07
27	72107.60	127864.55	353183.81
28	66051.52	182645.56	118148.20
29	65605.48	153032.06	107138.38
30	61994.48	115641.28	91131.24
31	61136.38	152701.92	88218.23
32	63408.86	129219.61	46085.25
33	55493.95	103057.49	214634.81
34	46426.07	157693.92	210797.67
35	46014.02	85047.44	205517.64

	RDS	Admin	Mkts
36	28663.76	127056.21	201126.82
37	44069.95	51283.14	197029.42
38	20229.59	65947.93	185265.10
39	38558.51	82982.09	174999.30
40	28754.33	118546.05	172795.67
41	27892.92	84710.77	164470.71
42	23640.93	96189.63	148001.11
43	15505.73	127382.30	35534.17
44	22177.74	154806.14	28334.72
45	1000.23	124153.04	1903.93
46	1315.46	115816.21	297114.46

In [92]:

```
y
```

Out[92]:

	Profit
0	192261.83
1	191792.06
2	191050.39
3	182901.99
4	166187.94
5	156991.12
6	156122.51
7	155752.60
8	152211.77
9	149759.96
10	146121.95
11	144259.40
12	141585.52
13	134307.35
14	132602.65
15	129917.04
16	126992.93
17	125370.37
18	124266.90
20	118474.03
21	111313.02
22	110352.25
23	108733.99
24	108552.04
25	107404.34
26	105733.54
27	105008.31
28	103282.38
29	101004.64
30	99937.59
31	97483.56
32	97427.84
33	96778.92
34	96712.80
35	96479.51

	Profit
36	90708.19
37	89949.14
38	81229.06
39	81005.76
40	78239.91
41	77798.83
42	71498.49
43	69758.98
44	65200.33
45	64926.08
46	49490.75

Model Training

In [93]:

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

In [94]:

```
linear_model = LinearRegression()
linear_model.fit(X,y)
```

Out[94]:

```
LinearRegression()
```

Model Testing

In [95]:

```
y_pred = linear_model.predict(X)
y_pred
```

Out[95]:

```
array([[188542.8230628 ],
       [184994.62678374],
       [180214.90791276],
       [171634.1734915 ],
       [171190.59125946],
       [162668.81166146],
       [157412.09273868],
       [157935.13591697],
       [149925.42607178],
       [154381.59493502],
       [136158.68492535],
       [136724.17370809],
       [129293.21181254],
       [127400.52166174],
       [147926.66860343],
       [145861.87995931],
       [117599.69951904],
       [129469.65085564],
       [129310.4892184 ],
       [117359.46320106],
       [116633.9373571 ],
       [115155.86574365],
       [111188.78105914],
       [115808.45242181],
       [103710.05841339],
       [111645.78404533],
       [114343.36306294],
       [101808.29936342],
       [103034.95633614],
       [102174.68520376],
       [ 99217.46993946],
       [101587.7677164 ],
       [100259.97404795],
       [ 89845.37125579],
       [ 93803.10685529],
       [ 77702.7026391 ],
       [ 94164.40969467],
       [ 74526.89333457],
       [ 87546.07447869],
       [ 77740.55319307],
       [ 78950.63823043],
       [ 74637.13647202],
       [ 64268.76335085],
       [ 67661.79016718],
       [ 52543.26892927],
       [ 58975.82938982]])
```

In [96]:

```
error = y-y_pred
error
```

Out[96]:

	Profit
0	3719.006937
1	6797.433216
2	10835.482087
3	11267.816509
4	-5002.651259
5	-5677.691661
6	-1289.582739
7	-2182.535917
8	2286.343928
9	-4621.634935
10	9963.265075
11	7535.226292
12	12292.308187
13	6906.828338
14	-15324.018603
15	-15944.839959
16	9393.230481
17	-4099.280856
18	-5043.589218
20	1114.566799
21	-5320.917357
22	-4803.615744
23	-2454.791059
24	-7256.412422
25	3694.281587
26	-5912.244045
27	-9335.053063
28	1474.080637
29	-2030.316336
30	-2237.095204
31	-1733.909939
32	-4159.927716
33	-3481.054048
34	6867.428744

	Profit
35	2676.403145
36	13005.487361
37	-4215.269695
38	6702.166665
39	-6540.314479
40	499.356807
41	-1151.808230
42	-3138.646472
43	5490.216649
44	-2461.460167
45	12382.811071
46	-9485.079390

In [97]:

```
from sklearn.metrics import mean_squared_error
```

In [98]:

```
mean_squared_error(y,y_pred)
```

Out[98]:

```
49559086.4165616
```

Model Validation Techniques

In [99]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.20,random_state=12)
```

In [100]:

```
# Training data
X_train.shape , y_train.shape
```

Out[100]:

```
((36, 3), (36, 1))
```

In [101]:

```
# Testing data
X_test.shape , y_test.shape
```

Out[101]:

```
((10, 3), (10, 1))
```

In [102]:

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

In [103]:

```
linear_model_2 = LinearRegression()
linear_model_2.fit(X_train,y_train)
```

Out[103]:

```
LinearRegression()
```

In [104]:

```
# For Training data
y_train_pred = linear_model_2.predict(X_train)
y_train_pred
```

Out[104]:

```
array([[149412.17891458],
       [104163.48937205],
       [116728.34578824],
       [156062.92469178],
       [ 52965.14044032],
       [ 89681.16081035],
       [ 68173.59918106],
       [118237.79641862],
       [117779.30630835],
       [114133.54035033],
       [103315.1631629 ],
       [186336.4955927 ],
       [104721.0314186 ],
       [173100.95904789],
       [ 79306.14100003],
       [ 77420.53817714],
       [130167.96195047],
       [ 95034.67128465],
       [ 77742.07564572],
       [151108.92580504],
       [115398.53250611],
       [189896.04561659],
       [137788.59930678],
       [ 57386.60198349],
       [ 94347.81507195],
       [112926.37619849],
       [128364.49989359],
       [164250.28056245],
       [111408.51917671],
       [130404.04587672],
       [ 74888.26874712],
       [173227.76374518],
       [182039.61622867],
       [160292.60119493],
       [102592.33465112],
       [138351.29387929]])
```

In [105]:

```
mean_squared_error (y_train,y_train_pred)
```

Out[105]:

```
48947552.58758988
```

In [106]:

```
# For Testing data
y_test_pred = linear_model_2.predict(X_test)
y_test_pred
```

Out[106]:

```
array([[ 88127.89970423],
       [ 64808.79237251],
       [159322.03903722],
       [ 74706.16993713],
       [100823.94718182],
       [117509.83556217],
       [147483.94392229],
       [100364.17491382],
       [130202.25531576],
       [103671.74048956]])
```

In [107]:

```
mean_squared_error(y_test,y_test_pred)
```

Out[107]:

```
58130761.88421633
```

MSE for Log transformation inputs

In [128]:

```
X = startups_data_log.drop(labels = 'Profit', axis=1)
y = startups_data_log[['Profit']]
```

In [130]:

```
X
```

Out[130]:

	log_RDS	log_Admin	log_Mkts
0	12.015815	11.826990	13.064277
1	11.999034	11.927533	13.003351
2	11.941075	11.524316	12.918862
3	11.880151	11.684117	12.856311
4	11.864338	11.422911	12.810849
5	11.789624	11.511071	12.801776
6	11.810178	11.899540	11.757571
7	11.777580	11.888138	12.688118
8	11.699758	11.909814	12.649518
9	11.722659	11.596155	12.628007
10	11.531876	11.613622	12.342180
11	11.519623	11.427265	12.428194
12	11.449600	11.754462	12.428574
13	11.429461	11.816691	12.439820
14	11.694774	11.961114	12.454934
15	11.648536	11.716820	12.475245
16	11.264632	11.708472	12.485014
17	11.458017	11.885024	12.551697
18	11.426814	11.645495	12.594458
20	11.241823	11.642789	12.607076
21	11.269445	11.943236	12.610662
22	11.211747	11.718172	12.622541
23	11.120365	11.568843	12.627309
24	11.252132	11.505713	11.853495
25	11.076971	11.846201	11.834738
26	11.229619	11.878512	11.805969
27	11.185915	11.758727	12.774744
28	11.098190	12.115303	11.679695
29	11.091415	11.938403	11.581877
30	11.034801	11.658248	11.420056
31	11.020862	11.936243	11.387569
32	11.057359	11.769269	10.738248
33	10.924029	11.543042	12.276693
34	10.745616	11.968411	12.258654
35	10.736701	11.350964	12.233287

	log_RDS	log_Admin	log_Mkts
36	10.263389	11.752385	12.211691
37	10.693533	10.845117	12.191108
38	9.914902	11.096621	12.129543
39	10.559932	11.326380	12.072537
40	10.266544	11.683057	12.059865
41	10.236128	11.346998	12.010488
42	10.070735	11.474077	11.904975
43	9.648965	11.754948	10.478250
44	10.006844	11.949929	10.251843
45	6.907985	11.729270	7.551675
46	7.181942	11.659760	12.601873

In [129]:

```
y
```

Out[129]:

	Profit
0	192261.83
1	191792.06
2	191050.39
3	182901.99
4	166187.94
5	156991.12
6	156122.51
7	155752.60
8	152211.77
9	149759.96
10	146121.95
11	144259.40
12	141585.52
13	134307.35
14	132602.65
15	129917.04
16	126992.93
17	125370.37
18	124266.90
20	118474.03
21	111313.02
22	110352.25
23	108733.99
24	108552.04
25	107404.34
26	105733.54
27	105008.31
28	103282.38
29	101004.64
30	99937.59
31	97483.56
32	97427.84
33	96778.92
34	96712.80
35	96479.51

	Profit
36	90708.19
37	89949.14
38	81229.06
39	81005.76
40	78239.91
41	77798.83
42	71498.49
43	69758.98
44	65200.33
45	64926.08
46	49490.75

In [131]:

```
from sklearn.model_selection import train_test_split
X_train, X_test,y_train,y_test = train_test_split(X,y,test_size=0.20, random_state=12)
```

In [133]:

```
#Training data
X_train.shape, y_train.shape
```

Out[133]:

```
((36, 3), (36, 1))
```

In [134]:

```
#Test data
X_test.shape , y_test.shape
```

Out[134]:

```
((10, 3), (10, 1))
```

Model Training

In [135]:

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

In [136]:

```
linear_model_3 = LinearRegression()
linear_model_3.fit(X_train , y_train)
```

Out[136]:

```
LinearRegression()
```

Model Testing

In [137]:

```
y_train_pred = linear_model_3.predict(X_train)
y_train_pred
```

Out[137]:

```
array([[139336.73900055],
       [122167.28715211],
       [129171.21806834],
       [139465.31373385],
       [  8078.10211039],
       [115410.31633933],
       [ 92050.21665448],
       [128094.77282489],
       [127659.40245842],
       [127016.98157767],
       [118738.59091428],
       [148144.43143607],
       [122195.74252022],
       [142953.58500315],
       [100499.97764287],
       [102787.27837292],
       [132205.74855817],
       [110886.34993687],
       [102293.42572246],
       [139810.05680126],
       [127161.05213698],
       [148438.03278781],
       [134079.00676794],
       [ 27380.86708393],
       [113440.32371912],
       [125983.1108189 ],
       [132348.97792258],
       [141325.39845304],
       [124507.4761016 ],
       [132648.32338427],
       [ 96494.56035391],
       [144172.01510395],
       [145398.59852169],
       [140263.06301017],
       [123064.13980885],
       [133484.15719696]])
```

In [138]:

```
from sklearn.metrics import mean_squared_error
```

In [139]:

```
mean_squared_error(y_train, y_train_pred)
```

Out[139]:

```
552322171.564474
```

In [142]:

```
y_test_pred = linear_model_3.predict(X_test)
y_test_pred
```

Out[142]:

```
array([[108600.5780037 ],
       [ 83245.77476872],
       [141770.96991732],
       [ 92179.44523267],
       [118702.61744256],
       [125641.25998927],
       [137578.47401492],
       [119928.64071267],
       [133522.96994929],
       [119595.86942234]])
```

In [143]:

```
mean_squared_error(y_test,y_test_pred)
```

Out[143]:

```
304684653.4841407
```

In [ ]:

```

```