Python Practice 4

Probability and Statistics Programming (Sejong University)¶

Date: 2019.05.22(By: S.M. Riazul Islam)

Data Frame, Normality Test, and t Test

Pandas Data Frame

In [2]:
```
# Read zoo.csv file saved in home directory
import numpy as np
import pandas as pd
pd.read_csv('zoo.csv',delimiter=',')
```

Out[2]:

	animal	uniq_id	water_need
0	elephant	1001	500
1	elephant	1002	600
2	elephant	1003	550
3	tiger	1004	300
4	tiger	1005	320
5	tiger	1006	330
6	tiger	1007	290
7	tiger	1008	310
8	zebra	1009	200
9	zebra	1010	220
10	zebra	1011	240
11	zebra	1012	230
12	zebra	1013	220
13	zebra	1014	100
14	zebra	1015	80
15	lion	1016	420
16	lion	1017	600
17	lion	1018	500
18	lion	1019	390
19	kangaroo	1020	410
20	kangaroo	1021	430
21	kangaroo	1022	410

Read data file from online location

In [4]:
```
import wget
url='http://46.101.230.157/dilan/pandas_tutorial_read.csv'
wget.download(url)
```

Out[4]: 'pandas_tutorial_read.csv'

In [6]:
```
col_names=['my_datetime','event','country','user_id','source','topic']
mydata=pd.read_csv('pandas_tutorial_read.csv',delimiter=';',names=col_names)
```

In [7]:
```
# check type and size of the mydata
type(mydata)
```

Out[7]:
```
pandas.core.frame.DataFrame
```

In [9]:
```
mydata.shape
```

Out[9]:
```
(1795, 6)
```

In [10]:
```
# Retrieve some sampple records from mydata
mydata.head()
```

Out[10]:

	my_datetime	event	country	user_id	source	topic
0	2018-01-01 00:01:01	read	country_7	2458151261	SEO	North America
1	2018-01-01 00:03:20	read	country_7	2458151262	SEO	South America
2	2018-01-01 00:04:01	read	country_7	2458151263	AdWords	Africa
3	2018-01-01 00:04:02	read	country_7	2458151264	AdWords	Europe
4	2018-01-01 00:05:03	read	country_8	2458151265	Reddit	North America

In [11]:
```
mydata.tail()
```

Out[11]:

	my_datetime	event	country	user_id	source	topic
1790	2018-01-01 23:57:14	read	country_2	2458153051	AdWords	North America
1791	2018-01-01 23:58:33	read	country_8	2458153052	SEO	Asia
1792	2018-01-01 23:59:36	read	country_6	2458153053	Reddit	Asia
1793	2018-01-01 23:59:36	read	country_7	2458153054	AdWords	Europe
1794	2018-01-01 23:59:38	read	country_5	2458153055	Reddit	Asia

In [12]:
```
mydata.sample(5)
```

Out[12]:

	my_datetime	event	country	user_id	source	topic
1774	2018-01-01 23:46:22	read	country_2	2458153035	AdWords	Asia
224	2018-01-01 03:10:20	read	country_2	2458151485	Reddit	Asia
574	2018-01-01 07:51:31	read	country_3	2458151835	Reddit	Asia
12	2018-01-01 00:09:11	read	country_5	2458151273	Reddit	Asia
1259	2018-01-01 17:02:05	read	country_7	2458152520	SEO	North America

Filter Data

In [13]:
```
mydata[['country','source']]      # outerbrace indicates column selections; inner
```

Out[13]:

	country	source
0	country_7	SEO
1	country_7	SEO
2	country_7	AdWords
3	country_7	AdWords
4	country_8	Reddit
5	country_6	Reddit
6	country_2	Reddit
7	country_6	AdWords
8	country_7	AdWords
9	country_5	Reddit
10	country_5	AdWords
11	country_7	SEO
12	country_5	Reddit
13	country_2	Reddit
14	country_7	Reddit
15	country_7	SEO
16	country_8	SEO
17	country_2	Reddit
18	country_4	SEO
19	country_2	Reddit
20	country_2	Reddit
21	country_7	Reddit
22	country_2	AdWords
23	country_2	Reddit
24	country_7	Reddit
25	country_2	AdWords
26	country_5	SEO
27	country_2	SEO
28	country_2	Reddit
29	country_7	Reddit
...	...	...
1765	country_2	Reddit
1766	country_5	AdWords
1767	country_5	AdWords

	country	source
1768	country_8	Reddit
1769	country_5	AdWords
1770	country_6	Reddit
1771	country_5	AdWords
1772	country_7	SEO
1773	country_8	Reddit
1774	country_2	AdWords
1775	country_2	AdWords
1776	country_6	Reddit
1777	country_5	SEO
1778	country_2	Reddit
1779	country_4	SEO
1780	country_8	AdWords
1781	country_2	Reddit
1782	country_2	AdWords
1783	country_3	Reddit
1784	country_2	SEO
1785	country_2	AdWords
1786	country_6	Reddit
1787	country_2	Reddit
1788	country_7	AdWords
1789	country_4	AdWords
1790	country_2	AdWords
1791	country_8	SEO
1792	country_6	Reddit
1793	country_7	AdWords
1794	country_5	Reddit

1795 rows × 2 columns

In [15]:
```
### Records with a specific value of a particular column
mydata[mydata.source=='SEO']
```

Out[15]:

	my_datetime	event	country	user_id	source	topic
0	2018-01-01 00:01:01	read	country_7	2458151261	SEO	North America
1	2018-01-01 00:03:20	read	country_7	2458151262	SEO	South America
11	2018-01-01 00:08:57	read	country_7	2458151272	SEO	Australia
15	2018-01-01 00:11:22	read	country_7	2458151276	SEO	North America
16	2018-01-01 00:13:05	read	country_8	2458151277	SEO	North America
18	2018-01-01 00:13:39	read	country_4	2458151279	SEO	North America
26	2018-01-01 00:20:18	read	country_5	2458151287	SEO	North America
27	2018-01-01 00:20:44	read	country_2	2458151288	SEO	North America
30	2018-01-01 00:24:52	read	country_6	2458151291	SEO	North America
41	2018-01-01 00:35:23	read	country_5	2458151302	SEO	Asia
54	2018-01-01 00:46:42	read	country_5	2458151315	SEO	North America
56	2018-01-01 00:49:37	read	country_7	2458151317	SEO	North America
57	2018-01-01 00:49:58	read	country_2	2458151318	SEO	Europe
60	2018-01-01 00:55:38	read	country_7	2458151321	SEO	North America
62	2018-01-01 00:56:27	read	country_6	2458151323	SEO	Asia
69	2018-01-01 01:03:19	read	country_5	2458151330	SEO	Europe
72	2018-01-01 01:04:22	read	country_8	2458151333	SEO	North America
73	2018-01-01 01:05:51	read	country_2	2458151334	SEO	North America
77	2018-01-01 01:12:51	read	country_4	2458151338	SEO	Africa
80	2018-01-01 01:14:24	read	country_4	2458151341	SEO	Africa
81	2018-01-01 01:15:53	read	country_7	2458151342	SEO	North America
83	2018-01-01 01:17:14	read	country_2	2458151344	SEO	South America
87	2018-01-01 01:18:12	read	country_6	2458151348	SEO	Australia
91	2018-01-01 01:20:37	read	country_7	2458151352	SEO	North America
97	2018-01-01 01:24:49	read	country_2	2458151358	SEO	North America
98	2018-01-01 01:28:30	read	country_4	2458151359	SEO	Asia
106	2018-01-01 01:35:40	read	country_5	2458151367	SEO	North America
110	2018-01-01 01:37:56	read	country_2	2458151371	SEO	Africa
114	2018-01-01 01:41:15	read	country_7	2458151375	SEO	North America
125	2018-01-01 01:51:51	read	country_2	2458151386	SEO	South America
...	...	...	...	...	...	...
1601	2018-01-01 21:41:50	read	country_2	2458152862	SEO	Asia
1602	2018-01-01 21:44:00	read	country_5	2458152863	SEO	North America

	my_datetime	event	country	user_id	source	topic
1603	2018-01-01 21:44:10	read	country_5	2458152864	SEO	North America
1615	2018-01-01 21:52:25	read	country_8	2458152876	SEO	North America
1616	2018-01-01 21:52:31	read	country_4	2458152877	SEO	Europe
1619	2018-01-01 21:56:27	read	country_7	2458152880	SEO	Europe
1624	2018-01-01 21:59:03	read	country_8	2458152885	SEO	North America
1637	2018-01-01 22:07:57	read	country_2	2458152898	SEO	Australia
1641	2018-01-01 22:09:53	read	country_6	2458152902	SEO	North America
1662	2018-01-01 22:28:41	read	country_7	2458152923	SEO	North America
1664	2018-01-01 22:30:26	read	country_4	2458152925	SEO	North America
1669	2018-01-01 22:33:19	read	country_7	2458152930	SEO	Australia
1675	2018-01-01 22:43:25	read	country_5	2458152936	SEO	Asia
1688	2018-01-01 22:49:18	read	country_1	2458152949	SEO	Africa
1692	2018-01-01 22:52:44	read	country_2	2458152953	SEO	South America
1694	2018-01-01 22:53:19	read	country_2	2458152955	SEO	North America
1706	2018-01-01 23:01:15	read	country_7	2458152967	SEO	North America
1708	2018-01-01 23:03:12	read	country_6	2458152969	SEO	Europe
1723	2018-01-01 23:14:18	read	country_3	2458152984	SEO	North America
1736	2018-01-01 23:21:08	read	country_5	2458152997	SEO	Asia
1740	2018-01-01 23:23:20	read	country_7	2458153001	SEO	Asia
1748	2018-01-01 23:26:34	read	country_7	2458153009	SEO	South America
1756	2018-01-01 23:32:29	read	country_7	2458153017	SEO	North America
1757	2018-01-01 23:32:36	read	country_7	2458153018	SEO	Europe
1762	2018-01-01 23:36:09	read	country_5	2458153023	SEO	North America
1772	2018-01-01 23:45:58	read	country_7	2458153033	SEO	South America
1777	2018-01-01 23:49:52	read	country_5	2458153038	SEO	North America
1779	2018-01-01 23:51:25	read	country_4	2458153040	SEO	South America
1784	2018-01-01 23:54:03	read	country_2	2458153045	SEO	North America
1791	2018-01-01 23:58:33	read	country_8	2458153052	SEO	Asia

346 rows × 6 columns

In [16]:
```
mydata.source=='SEO'
```

Out[16]:
```
0        True
1        True
2       False
3       False
4       False
5       False
6       False
7       False
8       False
9       False
10      False
11       True
12      False
13      False
14      False
15       True
16       True
17      False
18       True
19      False
20      False
21      False
22      False
23      False
24      False
25      False
26       True
27       True
28      False
29      False
        ...
1765    False
1766    False
1767    False
1768    False
1769    False
1770    False
1771    False
1772     True
1773    False
1774    False
1775    False
1776    False
1777     True
1778    False
1779     True
1780    False
1781    False
1782    False
1783    False
1784     True
1785    False
1786    False
1787    False
```

```
1788    False
1789    False
1790    False
1791     True
1792    False
1793    False
1794    False
Name: source, Length: 1795, dtype: bool
```

In [18]:
```
mydata[(mydata.source=='SEO')&(mydata.topic=='Asia')]        # Bitwise and opera
```

Out[18]:

	my_datetime	event	country	user_id	source	topic
41	2018-01-01 00:35:23	read	country_5	2458151302	SEO	Asia
62	2018-01-01 00:56:27	read	country_6	2458151323	SEO	Asia
98	2018-01-01 01:28:30	read	country_4	2458151359	SEO	Asia
207	2018-01-01 02:55:28	read	country_2	2458151468	SEO	Asia
228	2018-01-01 03:10:57	read	country_6	2458151489	SEO	Asia
295	2018-01-01 03:56:21	read	country_8	2458151556	SEO	Asia
324	2018-01-01 04:19:15	read	country_5	2458151585	SEO	Asia
358	2018-01-01 04:46:36	read	country_2	2458151619	SEO	Asia
421	2018-01-01 05:47:50	read	country_2	2458151682	SEO	Asia
491	2018-01-01 06:36:34	read	country_2	2458151752	SEO	Asia
648	2018-01-01 08:51:51	read	country_5	2458151909	SEO	Asia
650	2018-01-01 08:52:12	read	country_2	2458151911	SEO	Asia
692	2018-01-01 09:25:31	read	country_2	2458151953	SEO	Asia
709	2018-01-01 09:38:47	read	country_2	2458151970	SEO	Asia
746	2018-01-01 10:04:57	read	country_8	2458152007	SEO	Asia
827	2018-01-01 11:15:43	read	country_5	2458152088	SEO	Asia
897	2018-01-01 12:03:46	read	country_7	2458152158	SEO	Asia
1054	2018-01-01 14:10:19	read	country_5	2458152315	SEO	Asia
1140	2018-01-01 15:18:23	read	country_6	2458152401	SEO	Asia
1154	2018-01-01 15:34:33	read	country_5	2458152415	SEO	Asia
1168	2018-01-01 15:45:08	read	country_5	2458152429	SEO	Asia
1169	2018-01-01 15:45:48	read	country_8	2458152430	SEO	Asia
1232	2018-01-01 16:42:47	read	country_2	2458152493	SEO	Asia
1257	2018-01-01 16:59:56	read	country_1	2458152518	SEO	Asia
1317	2018-01-01 17:55:56	read	country_8	2458152578	SEO	Asia
1333	2018-01-01 18:05:47	read	country_6	2458152594	SEO	Asia
1357	2018-01-01 18:23:19	read	country_8	2458152618	SEO	Asia
1527	2018-01-01 20:43:33	read	country_7	2458152788	SEO	Asia
1528	2018-01-01 20:44:10	read	country_4	2458152789	SEO	Asia
1601	2018-01-01 21:41:50	read	country_2	2458152862	SEO	Asia
1675	2018-01-01 22:43:25	read	country_5	2458152936	SEO	Asia
1736	2018-01-01 23:21:08	read	country_5	2458152997	SEO	Asia
1740	2018-01-01 23:23:20	read	country_7	2458153001	SEO	Asia
1791	2018-01-01 23:58:33	read	country_8	2458153052	SEO	Asia

Data download from github and perform Statistical Analysis

In [19]:
```
import numpy as np
import scipy as sp
import stats
import wget
import matplotlib.pyplot as plt
import pandas as pd


# File load from github
# The original link: https://github.com/Opensourcefordatascience/Data-sets/blob/m
# domain should be changed to raw.github for only file download; not the content
# Also, blob should be omitted.

url='https://raw.github.com/Opensourcefordatascience/Data-sets/master/blood_press
wget.download(url)
```

Out[19]: 'blood_pressure.csv'

In [20]:
```
bpdata=pd.read_csv('blood_pressure.csv',delimiter=',')
```

In [21]:
```
bpdata.head()
```

Out[21]:

	patient	sex	agegrp	bp_before	bp_after
0	1	Male	30-45	143	153
1	2	Male	30-45	163	170
2	3	Male	30-45	153	168
3	4	Male	30-45	153	142
4	5	Male	30-45	146	141

In [22]:
```
bpdata.tail()
```

Out[22]:

	patient	sex	agegrp	bp_before	bp_after
115	116	Female	60+	152	152
116	117	Female	60+	161	152
117	118	Female	60+	165	174
118	119	Female	60+	149	151
119	120	Female	60+	185	163

In [24]:
```
bpdata.sample(5)
```

Out[24]:

	patient	sex	agegrp	bp_before	bp_after
56	57	Male	60+	147	176
50	51	Male	60+	175	146
38	39	Male	46-59	185	140
92	93	Female	46-59	144	157
67	68	Female	30-45	151	135

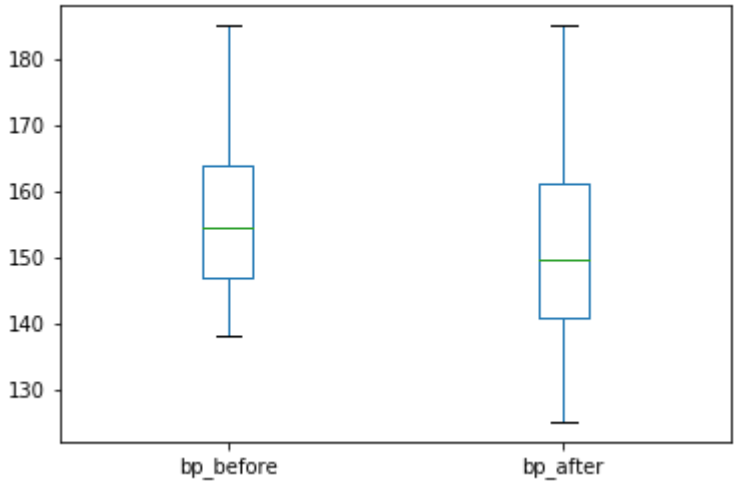
In [26]:
```
bpdata[['bp_before','bp_after']].describe()
```

Out[26]:

	bp_before	bp_after
count	120.000000	120.000000
mean	156.450000	151.358333
std	11.389845	14.177622
min	138.000000	125.000000
25%	147.000000	140.750000
50%	154.500000	149.500000
75%	164.000000	161.000000
max	185.000000	185.000000

Check outlier via boxplot

In [27]:
```
bpdata[['bp_before','bp_after']].plot(kind='box')
plt.show()
```



In [28]:
```
bpdata['difference']=bpdata['bp_before']-bpdata['bp_after']
```

In [29]:
```
type(bpdata['difference'])
```

Out[29]:
```
pandas.core.series.Series
```

In [30]:
```
bpdata.head()
```

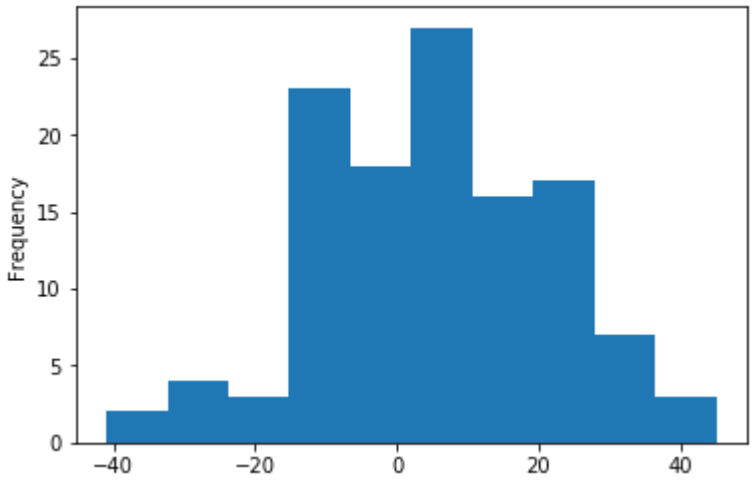
Out[30]:

	patient	sex	agegrp	bp_before	bp_after	difference
0	1	Male	30-45	143	153	-10
1	2	Male	30-45	163	170	-7
2	3	Male	30-45	153	168	-15
3	4	Male	30-45	153	142	11
4	5	Male	30-45	146	141	5

Histogram Check

In [31]:
```
bpdata['difference'].plot(kind='hist')
```

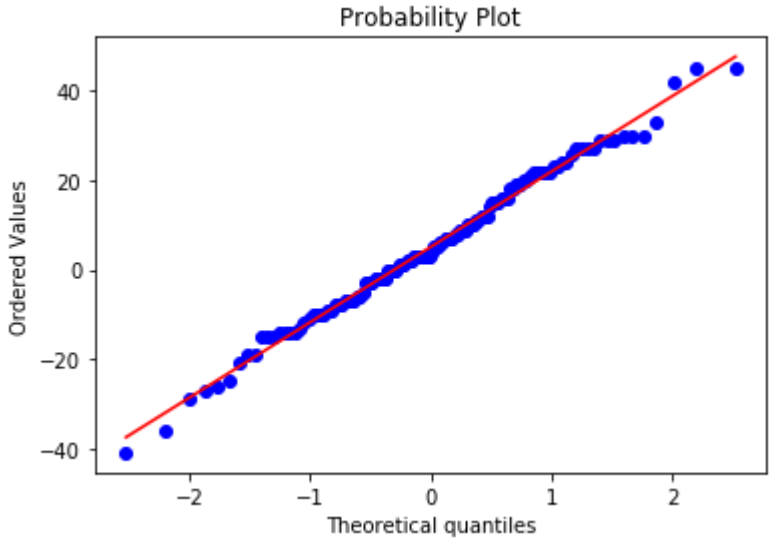
Out[31]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x20de03f9b70>
```



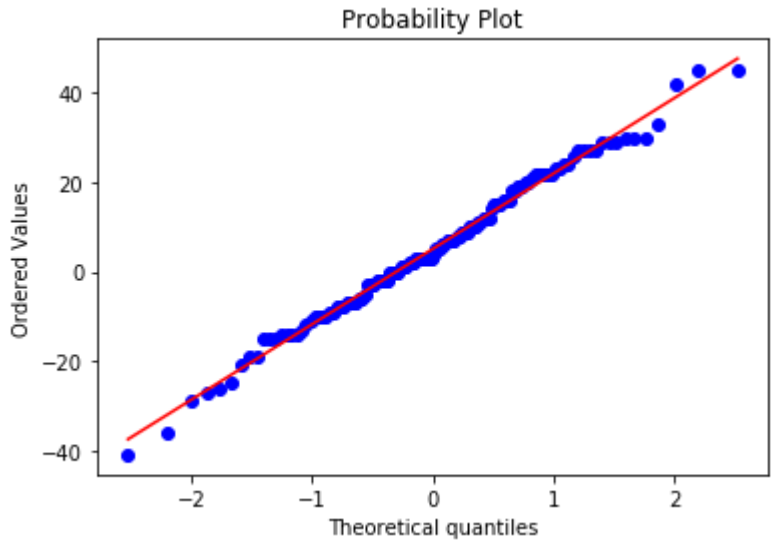
QQ Plot

In [35]:
```
import scipy.stats as mystat
mystat.probplot(bpdata['difference'],plot=plt)
```

Out[35]:
```
((array([-2.52654228, -2.1978944 , -2.0086642 , -1.8721281 , -1.76356639,
         -1.67252351, -1.59354821, -1.5234211 , -1.46007481, -1.40209915,
         -1.3484871 , -1.29849326, -1.25154963, -1.20721295, -1.16513026,
         -1.12501567, -1.08663416, -1.04979006, -1.01431855, -0.98007946,
         -0.94695242, -0.9148333 , -0.8836313 , -0.85326673, -0.82366923,
         -0.79477627, -0.76653206, -0.73888652, -0.71179451, -0.68521516,
         -0.65911132, -0.6334491 , -0.60819743, -0.58332778, -0.55881382,
         -0.53463119, -0.51075726, -0.48717098, -0.46385269, -0.44078394,
         -0.41794744, -0.39532687, -0.37290682, -0.35067268, -0.32861058,
         -0.3067073 , -0.28495019, -0.26332716, -0.24182657, -0.2204372 ,
         -0.19914822, -0.17794913, -0.15682971, -0.13578003, -0.11479034,
         -0.09385111, -0.07295295, -0.05208661, -0.03124292, -0.0104128 ,
          0.0104128 ,  0.03124292,  0.05208661,  0.07295295,  0.09385111,
          0.11479034,  0.13578003,  0.15682971,  0.17794913,  0.19914822,
          0.2204372 ,  0.24182657,  0.26332716,  0.28495019,  0.3067073 ,
          0.32861058,  0.35067268,  0.37290682,  0.39532687,  0.41794744,
          0.44078394,  0.46385269,  0.48717098,  0.51075726,  0.53463119,
          0.55881382,  0.58332778,  0.60819743,  0.6334491 ,  0.65911132,
          0.68521516,  0.71179451,  0.73888652,  0.76653206,  0.79477627,
          0.82366923,  0.85326673,  0.8836313 ,  0.9148333 ,  0.94695242,
          0.98007946,  1.01431855,  1.04979006,  1.08663416,  1.12501567,
          1.16513026,  1.20721295,  1.25154963,  1.29849326,  1.3484871 ,
          1.40209915,  1.46007481,  1.5234211 ,  1.59354821,  1.67252351,
          1.76356639,  1.8721281 ,  2.0086642 ,  2.1978944 ,  2.52654228]),
  array([-41, -36, -29, -27, -26, -25, -21, -19, -19, -15, -15, -15, -14,
         -14, -14, -14, -13, -12, -11, -10, -10, -10, -10,  -9,  -9,  -8,
          -8,  -8,  -7,  -7,  -7,  -7,  -6,  -6,  -5,  -3,  -3,  -3,  -2,
          -2,  -2,  -2,  -2,   0,   0,   0,   0,   1,   1,   1,   2,   2,
           2,   3,   3,   3,   3,   3,   3,   3,   4,   5,   5,   6,   6,
           7,   7,   7,   7,   8,   8,   9,   9,   9,  10,  10,  10,  11,
          11,  12,  12,  12,  14,  15,  15,  15,  16,  16,  16,  18,  18,
          19,  19,  20,  20,  21,  22,  22,  22,  22,  22,  23,  23,  24,
          24,  26,  27,  27,  27,  27,  29,  29,  29,  30,  30,  30,  33,
          42,  45,  45], dtype=int64)),
 (16.87714049175083, 5.091666666666663, 0.9965872150745229))
```



In [37]:
```
import scipy.stats as mystat
mystat.probplot(bpdata['difference'], dist=mystat.norm, sparams=(0,1), plot=plt)
```



Shapiro Test

In [44]:
```
# Shapiro-Wilk test
# H_0: Samples are drawn from Null Value
# First one is Test Value w; Second value is the p-value

mystat.shapiro(bpdata['difference'])
```

Out[44]:
```
(0.9926842451095581, 0.7841846942901611)
```

paired t Test manually

In [39]:
```
dbar=bpdata['difference'].mean()
sigmabar=bpdata['difference'].std()
```

In [40]:
```
t=(dbar-0)/(sigmabar/np.sqrt(120))      # n=120 as we saw from tail
```

In [41]:
```
t
```

Out[41]:
```
3.337187051083365
```

In [42]:
```
### Calculate p-value
2*(1-mystat.t.cdf(t,df=119))
```

Out[42]:
```
0.001129791464484109
```

The above results show that we reject Null Hypothesis

t Test by calling function

In [43]:
```
mystat.ttest_rel(bpdata['bp_before'],bpdata['bp_after'])
```

Out[43]:
```
Ttest_relResult(statistic=3.3371870510833657, pvalue=0.0011297914644840823)
```