QQ-plot

[ ] ↳ *12 cells hidden*

Uber Data

```
id = "1NokZy4YzavFdTZlWcIUs47WW5M2A4ElE"
print("https://drive.google.com/uc?export=download&id=" + id)
```

```
https://drive.google.com/uc?export=download&id=1NokZy4YzavFdTZlWcIUs47WW5M2A4E
```

```
!wget "https://drive.google.com/uc?export=download&id=1NokZy4YzavFdTZlWcIUs47WW5M2A
```

```
--2022-05-12 16:03:56--  https://drive.google.com/uc?export=download&id=1NokZy
Resolving drive.google.com (drive.google.com)... 173.194.218.102, 173.194.218.
Connecting to drive.google.com (drive.google.com)|173.194.218.102|:443... conn
HTTP request sent, awaiting response... 303 See Other
Location: https://doc-0c-ag-docs.googleusercontent.com/docs/securesc/ha0ro937g
Warning: wildcards not supported in HTTP.
--2022-05-12 16:03:56--  https://doc-0c-ag-docs.googleusercontent.com/docs/sec
Resolving doc-0c-ag-docs.googleusercontent.com (doc-0c-ag-docs.googleuserconte
Connecting to doc-0c-ag-docs.googleusercontent.com (doc-0c-ag-docs.googleuserc
HTTP request sent, awaiting response... 200 OK
Length: 18251707 (17M) [application/zip]
Saving to: 'Uber_dataset.zip'

Uber_dataset.zip    100%[===================>]  17.41M  39.1MB/s    in 0.4s

2022-05-12 16:03:57 (39.1 MB/s) - 'Uber_dataset.zip' saved [18251707/18251707]
```

```
!unzip Uber_dataset.zip
```

```
Archive:  Uber_dataset.zip
  inflating: uber_travel_data.csv
  inflating: __MACOSX/._uber_travel_data.csv
```

```
!ls -lrt
```

```
total 525788
drwxr-xr-x 1 root root      4096 May  3 13:42 sample_data
-rw-r--r-- 1 root root 520141836 May 12 14:30 uber_travel_data.csv
-rw-r--r-- 1 root root  18251707 May 12 16:03 Uber_dataset.zip
drwxr-xr-x 2 root root      4096 May 12 16:04 __MACOSX
```

```
import pandas as pd
```

```
df = pd.read_csv("./uber_travel_data.csv")
df.sample(100).head()
```

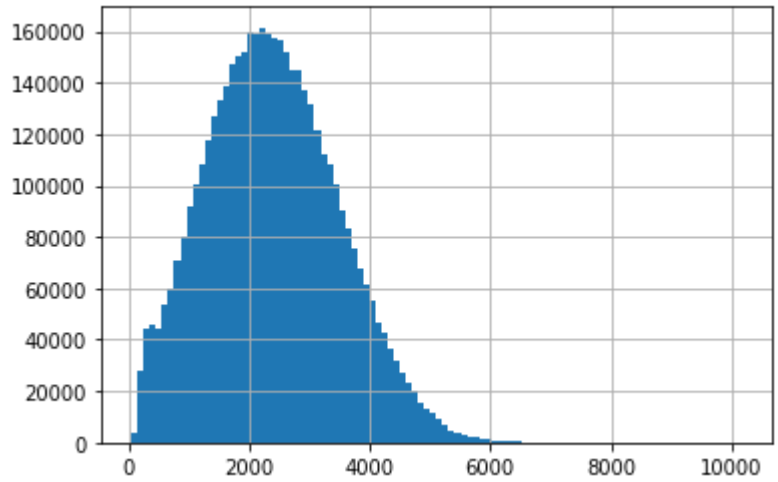
	sourceid	source	dstid	
2936366	187	60, Vijay Vihar Phase II, Pocket C, Sector 1, ...	77	0 Mehrauli - Ba
982706	63	Jawaharlal Nehru Stadium Marg, CGO Complex, Pr...	172	nullShiva Roa
3711028	234	113, Press Colony, Press Colony, Mayapuri, New...	259	
114690	7	P 13, Baird Place, Delhi Cantonment, New Delhi	131	4400 Gali Bal
2886810	183	NaN	243	

```
df.shape
```

```
(4542026, 5)
```

```
# histogram of travel_times
df["travel_time"].hist(bins = 100)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f05a867c150>
```




```
df.value_counts(['sourceid', 'dstid']).sort_values()
```

```
sourceid  dstid
69        4        50
167       107      50
          101      50
264       14       50
167       100      50
                   ..
83        88       79
244       32       79
202       201      79
```

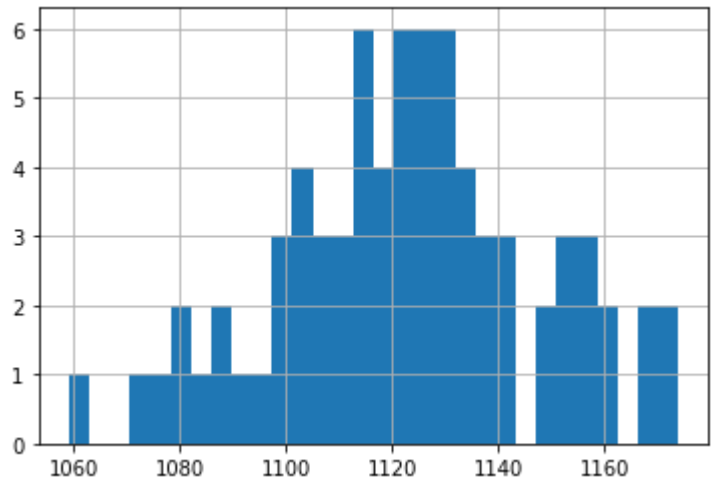
```
                135       79
45              170       79
Length: 70429, dtype: int64
```

```
data = df[(df["sourceid"] == 1) & (df["dstid"] == 5)] ["travel_time"]
data.shape
```

```
(75,)
```

```
data.hist(bins=30)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f05a5fd56d0>
```



CLT for C.I on mean of travel_time

[ ] ↳ *6 cells hidden*

95% C.I on 99th percentile value for travel_time via bootsrapping

```
# What if we want a C.I on the 99th precentile?
#Let's create r=10000 bootstrap samples, and let each bootstrap sample be of size=1
# bs_99p is a list of 'r' bootstrap sample's 99th percentiles
r = 10000
data = df[(df["sourceid"] == 1) & (df["dstid"] == 5)] ["travel_time"]
size = 75
bs_99p = np.empty(r)

for i in range(r):
    bs_sample = np.random.choice(data, size=size)
    bs_99p[i] = np.percentile(bs_sample,99)
```

```
len(bs_99p)
```

```
10000
```

```
bs_99p
```

```
array([1174.  , 1174.  , 1174.  , ..., 1167.  , 1174.  , 1168.82])
```

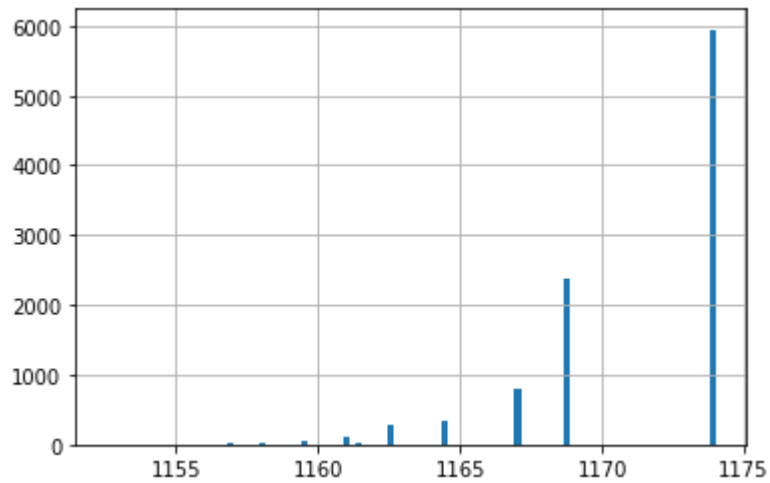
```
#bs_99p may or maynot be normally distributed.
print(np.percentile(bs_99p,2.5))
print(np.percentile(bs_99p,97.5))
```

```
1162.56
1174.0
```

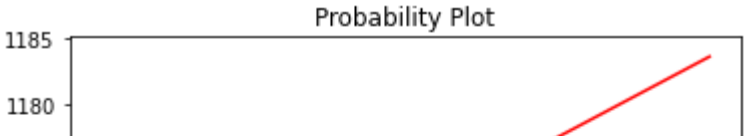
```
# Point estimate of the 99th percenitle of the 75 observed samples
print(np.percentile(data,99))
```

```
1174.0
```

```
# plot the pdf of bs_99p
import matplotlib.pyplot as plt
plt.figure()
plt.hist(bs_99p, bins=100)
plt.grid()
plt.show()
```




```
# QQ-plot with normal distribution
fig, ax1 = plt.subplots()
prob = stats.probplot(bs_99p, dist=stats.norm, plot=ax1)
```



box-cox transform

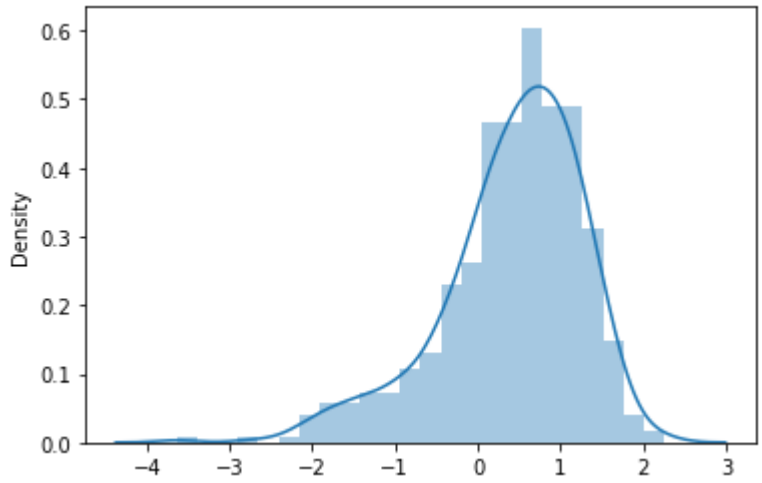

```
data =[ 0.04177737,  0.97977259,  1.19684675,  0.75969411,  0.2772351 ,
        1.20400739,  1.19512711, -1.33315966,  0.47241401,  0.58453053,
        0.21167461,  0.87106215, -0.56663286,  0.3702523 ,  0.72724427,
        0.41126015,  0.33358864,  0.72878097,  0.69929305,  0.72581333,
        1.67334826, -1.54572083, -1.22840893,  0.47103287,  0.895276  ,
        0.16538052, -0.43575904,  1.62784202,  0.98340417,  0.90482144,
       -0.47914975,  0.71812022,  1.14243   , -0.04393411,  1.24946471,
       -0.8699551 ,  1.60196517,  1.00140898,  1.48233878, -0.37088602,
       -0.0954339 ,  1.2969551 ,  0.0457524 , -0.06486335,  0.43257115,
       -0.18945797,  0.46525944,  0.12974487, -0.10501035,  0.94060547,
       -1.57714093,  0.24292938,  0.68759359,  0.24113398,  0.74353881,
        0.0129037 ,  0.47936105, -0.0596165 ,  0.3300311 , -0.19409805,
       -2.15213968, -0.9169724 ,  1.40476752,  0.74067023,  0.36119747,
        1.04507563, -0.54692221,  0.65000261,  0.5359208 ,  0.40091749,
        0.16959609,  0.43828974,  1.69191812, -0.40588725,  0.52772481,
        0.2410331 ,  1.8226663 , -1.36677194,  0.41745297,  0.94050797,
        1.15797033,  0.13883716,  0.9648131 ,  0.71495948,  1.73284151,
        0.9571359 ,  0.38785662,  0.41390929, -1.10391874, -0.41368798,
       -0.90497721,  1.37201217,  0.52934518,  0.45456489, -0.23302007,
        0.1206425 ,  1.43043074,  0.0599792 ,  0.39871742, -0.03524401,
       -1.59860382, -1.94105256,  1.22334603, -1.76544176, -0.80324714,
        1.16037195,  0.38303564, -0.44427508,  1.13694237,  0.58281873,
        1.01938666,  0.85409657,  0.32051415,  0.08834169,  0.15365941,
        1.68716621, -0.24197654,  1.2676363 ,  1.48518839, -0.47335603,
        1.15654111,  0.76654086, -0.11389136,  1.30586524,  0.32307392,
        0.54523295,  0.38590127,  0.50793605, -0.34701396,  0.74541391,
        0.79535705,  1.01896308, -0.22023158, -0.48871769,  0.05838767,
       -0.25024374,  0.69928181,  2.21454052,  0.20445216,  1.32931331,
        0.08653597,  0.07823139,  1.14485681,  0.91738973,  0.0543534 ,
       -1.45447157,  1.08313814, -0.27451755,  1.15577356,  1.15404113,
       -1.82969195,  0.17610396,  1.08855269,  0.67994842,  0.0750844 ,
       -0.30914221,  0.68824746, -2.02655603, -0.65056827,  0.03919982,
        0.06828509,  1.17926148,  0.86701368,  1.45238655,  1.63738079,
        0.63609739, -1.31232421,  0.98509236,  1.15594405,  0.20902709,
        0.96664264,  0.11769247,  0.48530914,  1.12505311,  0.60806881,
       -1.54771281,  0.92716597,  1.16839655, -0.06376581,  0.75839488,
        1.05027756,  1.41329557,  0.85657177, -0.2160035 ,  1.12248554,
        0.20020919,  1.1861288 ,  0.76429072, -1.83554409, -0.04585441,
        1.06873376,  0.10936729,  1.48407643,  0.52580339,  1.19815856,
        0.53797982, -0.42615522, -0.38198519,  0.53974062,  0.06254645,
        0.11724433,  0.67580552,  0.63406064,  1.03362043, -1.88639841,
        0.62474754,  0.89065659,  0.5328413 ,  0.92901562,  0.82901618,
       -1.40196713,  0.25330113, -0.11682618,  0.79230788, -1.37307874,
        0.37353503,  0.65753252,  0.61958929,  0.95358877, -0.63137426,
        0.73935171, -0.3392893 ,  0.90018122,  1.13697138,  1.07777798,
       -0.67428172,  1.20112044,  0.13277637,  0.88485663, -0.73037033,
```

```
 0.73817138,  0.1309939 ,  0.77936817,  1.16422402,  1.22697646,
 0.31219482,  0.6517649 ,  1.35374234, -0.11302125,  1.38551431,
 1.4890138 ,  0.75586738,  1.76803848,  0.56651688, -0.67678907,
 0.19554616,  0.46406559, -0.06019572,  1.53990381, -1.13432049,
-0.80700753, -2.31246741,  1.33986194,  1.38730476,  0.82881232,
 0.70062208, -0.12381894, -0.46690349, -1.57527874,  1.72209985,
 0.18183212,  0.8400035 ,  0.418469  ,  1.36179378, -0.92426075,
 0.22270703, -0.13774932,  0.93111539, -0.88921133,  1.0997085 ,
-0.9937949 ,  0.38843634,  1.01148004, -0.43816108, -0.95414947,
 1.32330751,  0.30404196,  0.88512404,  0.91372546,  0.32134319,
 0.14559158, -0.19978188,  0.88329649,  0.30335937,  0.76175674,
 0.57364537,  1.02631156,  1.66794999, -1.03792174, -0.31515864,
 1.07180383,  0.59720417, -0.32040037, -0.7674771 ,  0.25057312,
 0.27762623,  0.54672121, -1.34336276,  0.53814872,  1.24214509,
-1.12005068,  1.37171113,  0.0616415 , -0.74262483,  1.01415696,
 0.3901361 ,  0.70918134,  0.20065952,  0.94970448,  0.73886174,
 1.06909761,  0.86064287,  1.15752969,  0.82554495,  1.36024967,
 0.59598245,  0.89922565,  0.7362065 ,  1.00841732,  0.55340554,
 0.41274327,  0.50711349,  0.45157236, -0.2457261 ,  1.07731295,
-0.50619092,  0.77516586, -2.71031075,  0.69192707,  0.84959366,
 1.45540949, -0.44551638,  1.28008884,  0.61377305, -0.54839374,
 1.16915428,  1.1075064 ,  1.0229388 ,  1.22989514,  1.24266425,
 0.17096114,  1.00952836, -0.28128762, -0.31360414,  0.50315717,
 0.23675518, -0.15479312,  0.28744327,  0.66566966, -0.14055415,
 0.60945716, -1.45725682, -1.76229852,  0.10049782,  0.59945138,
 0.60902798,  0.92513724,  1.3161839 , -0.02831568,  0.53837944,
-3.63123097,  2.24728714,  0.14248232,  1.15824823,  0.1331667 ,
 1.30352524,  0.31862759,  1.48258693,  0.82365142,  1.22927344,
 0.65581787,  1.49120079,  1.26751206,  0.6596013 , -0.30466474,
 0.92502302,  1.05893148,  1.25006908,  1.51266005, -0.36946192,
 0.20367163,  1.54883376, -0.07722085,  0.29042734, -0.07913684,
 1.0009701 ,  0.66712984,  1.72579542,  1.81505526,  1.02742471,
 1.31574026, -1.10915715, -0.54120723,  0.51054351, -0.88139742,
-1.72785233,  1.9585019 ,  1.09644834, -1.27615429, -2.11919702,
 0.10586263,  0.70464499,  0.61638469,  0.30262653, -0.40630085,
-0.2274666 ,  1.20697563,  0.36656195,  0.57455917, -0.95850539,
 0.57487625,  0.09909038, -2.02132122,  0.79842403,  0.29482801,
-0.56063591,  1.22430722, -0.26074589, -0.61835677,  0.91307203,
 0.98181937,  1.60472708,  0.80975178,  0.57399004,  0.20730555,
 1.03604696,  1.99239206,  1.35579176, -1.06755095,  0.79769852,
-0.11886134,  0.92591275,  0.31100381,  1.45719763, -0.18650384,
 0.98158411,  0.38823413,  0.03501161,  1.3873394 ,  1.06988861,
-0.61101705,  0.64726664,  0.71829533,  0.37560761,  0.32028192,
 0.46012344,  0.92880202,  0.67717555,  0.04629136,  0.47546512,
 1.4513086 ,  1.45343272,  1.54991229,  0.62037232,  0.01407354,
 0.46979478,  0.05595689, -1.73249288,  0.23003225,  1.29352827,
 0.80189453,  1.61966331,  0.69681106,  1.03215339, -1.21549361,
 0.93475221,  1.30143537,  0.7254352 ,  0.22841529,  1.50249735,
-0.02415314, -0.18205881,  0.95388083,  0.66182587,  0.08282857,
 1.53432986,  1.07818559,  1.04804152,  0.62920033,  0.2221568 ,
 1.11689153,  0.70328342,  1.48907562, -0.85967934,  0.37330663,
 0.10042743,  0.43601618, -0.84872277, -0.18902961,  1.16872747,
 0.49445364,  0.97912906,  0.16970087,  1.43121388,  0.67825154,
 0.8233865 ,  1.20263091,  0.49206124,  0.34548617,  1.58287164]
```

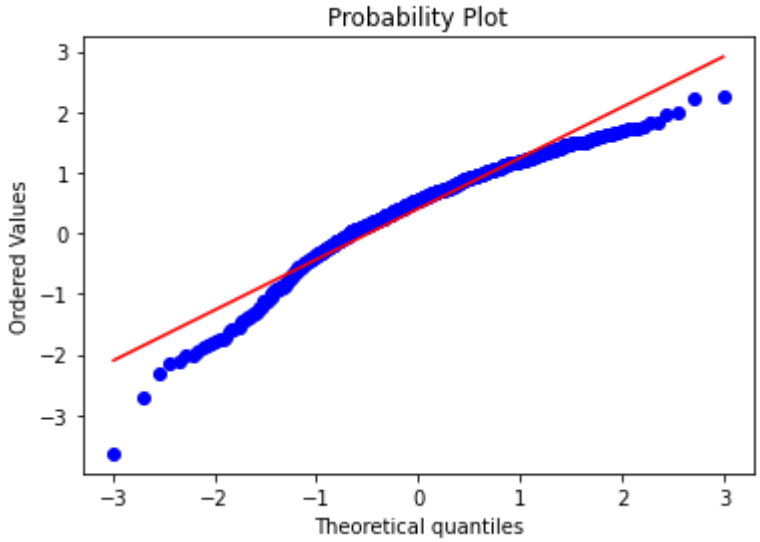
```
x = np.array(data)
```

```
sns.distplot(x)
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619: FutureWa
  warnings.warn(msg, FutureWarning)
<matplotlib.axes._subplots.AxesSubplot at 0x7f041121e250>
```



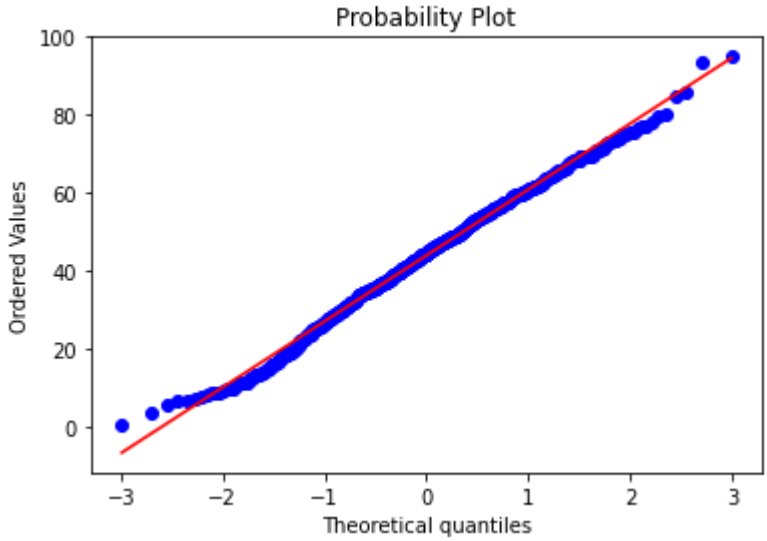

```
# qq plot of x vs normal
fig, ax1 = plt.subplots()
prob = stats.probplot(x, dist='norm', fit=False, plot=ax1)
```




```
# box cox transform
x1 = x + 5 # to avoid negative values of x
xt, l = stats.boxcox(x1,); # returns x_transfomred and lambda
print("lambda :" + str(l))

# check if xt is gaussian or not using QQ-Plot
fig, ax2 = plt.subplots()
prob = stats.probplot(xt, dist='norm', plot=ax2)
```

```
lambda :2.8250281815319838
```



✓ 0s completed at 22:12