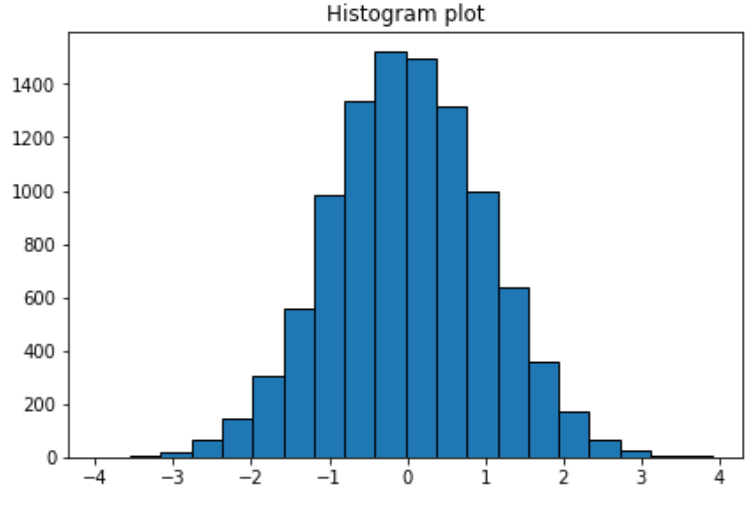
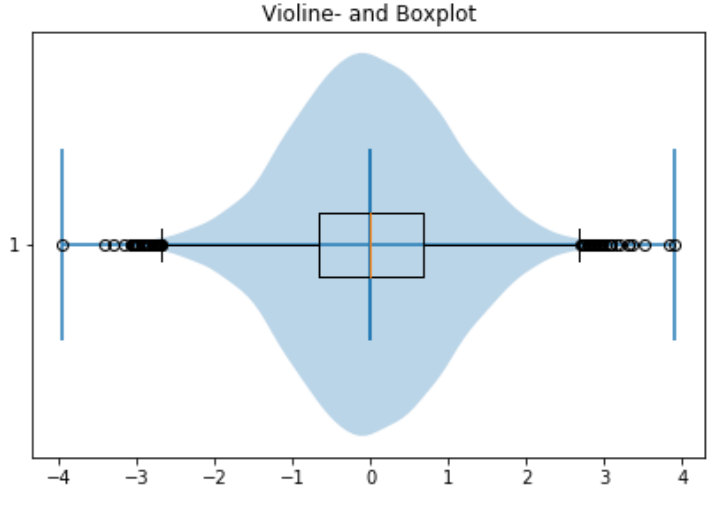
Why is knowledge of a distribution important?

1. Proper outliers handling
2. Correct hypothesis creation
3. Better model fit
4. Realisitic tests

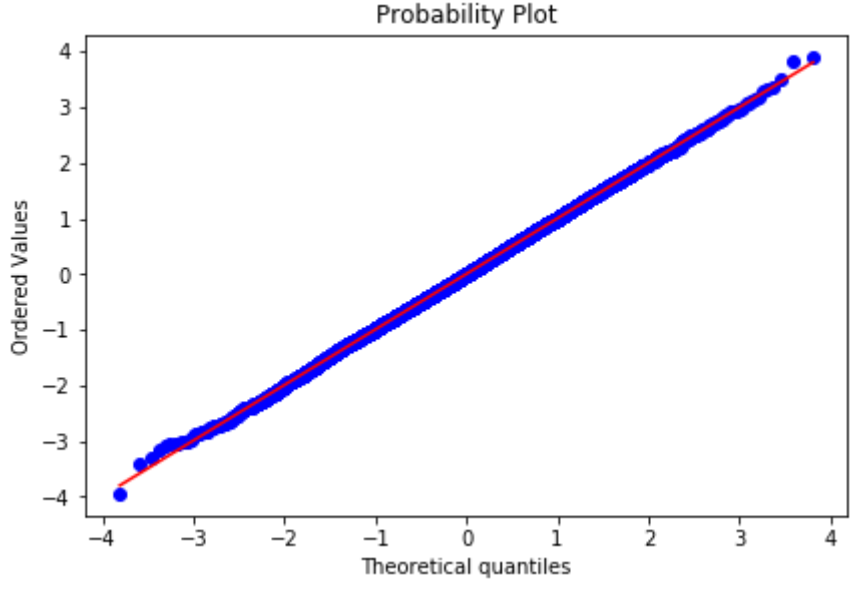
In [2]:
```
# Wouldn't it be great, if everyting was normally distributed?
normal = generate_distribution('normal')['observation']
fig = plt.figure(figsize=(15, 15))
plt.subplot(321)
plt.title("Histogram plot")
plt.hist(normal, bins=BINS, edgecolor='k')
plt.subplot(322)
plt.title("Violine- and Boxplot")
plt.violinplot(normal, vert=False, widths=0.9, showmeans=True, showextrema=
True, showmedians=True)
# plt.subplot(323)  # have them in different colors or differenbt plots?
plt.boxplot(normal,vert=False)
plt.show()
# TODO: add axis labels
```



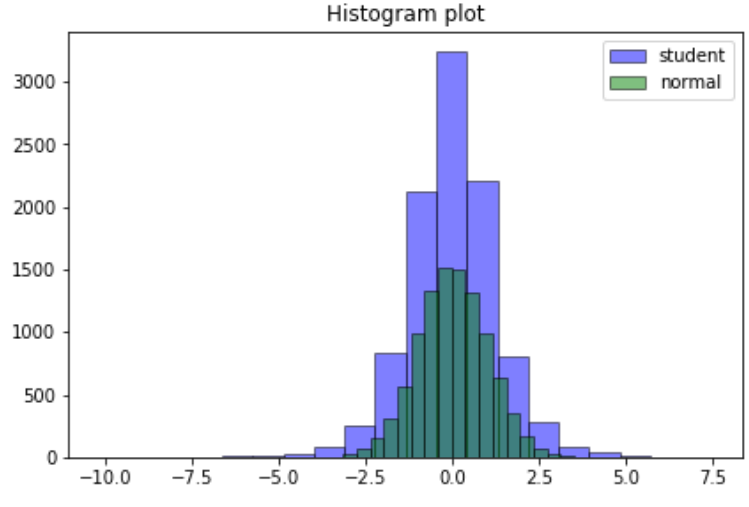


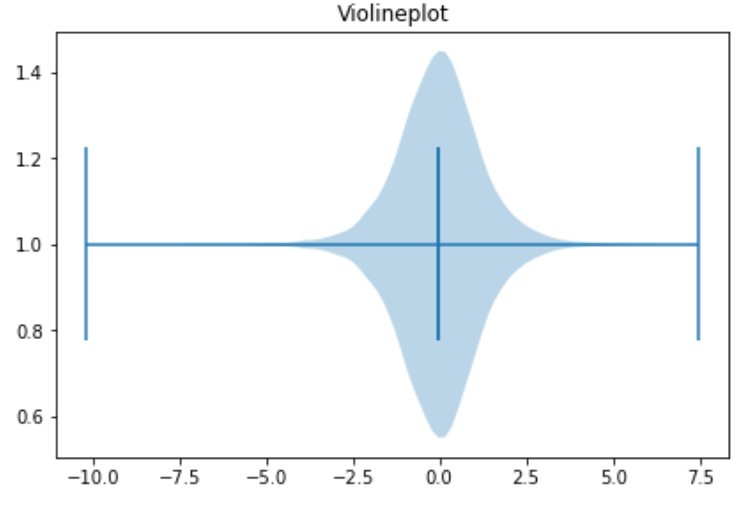
In [3]:
```
# A Q-Q plot stands for a "quantile-quantile plot".
# It is a plot where the axes are purposely transformed in order to make a
 normal (or Gaussian) distribution a
# ppear in a straight line.
# In other words, a perfectly normal distribution would exactly follow a li
ne with slope = 1 and intercept = 0.
# Therefore, if the plot does not appear to be - roughly - a straight line,

# then the underlying distribution is not normal.
# If it bends up, then there are more "high flyer" values than expected, fo
r instance.
fig = plt.figure(figsize=(15, 15))
plt.subplot(321)
qq = stats.probplot(normal, plot=plt)
```

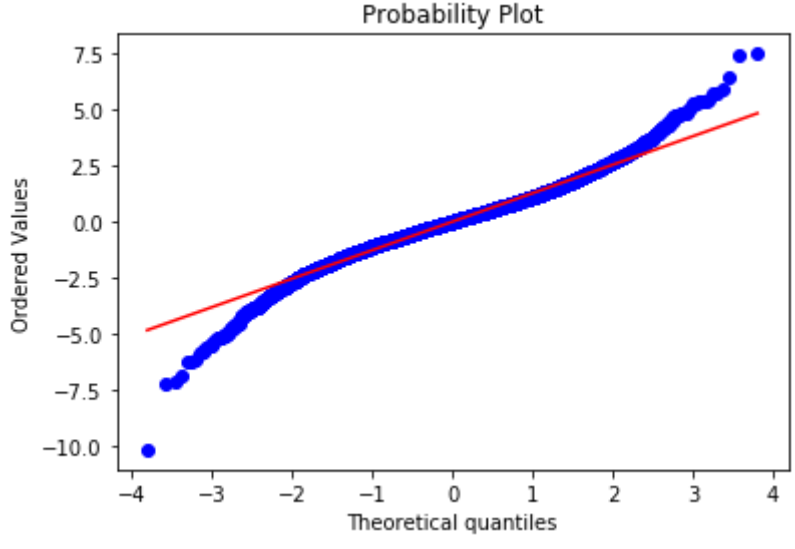


In [4]:
```
# Students distribution - approximation for
student = generate_distribution('student')['observation']
fig = plt.figure(figsize=(15, 15))
plt.subplot(321)
plt.title("Histogram plot")
plt.hist(student, bins=BINS, alpha=0.5, label='student', color='b', edgecol
or='k')
plt.hist(normal, bins=BINS, alpha=0.5, label='normal', color='g', edgecolor
='k')
plt.gca().legend(('student','normal'))
plt.subplot(322)
plt.title("Violineplot")
plt.violinplot(student, vert=False, widths=0.9, showmeans=True, showextrema
=True, showmedians=True)
plt.show()
```

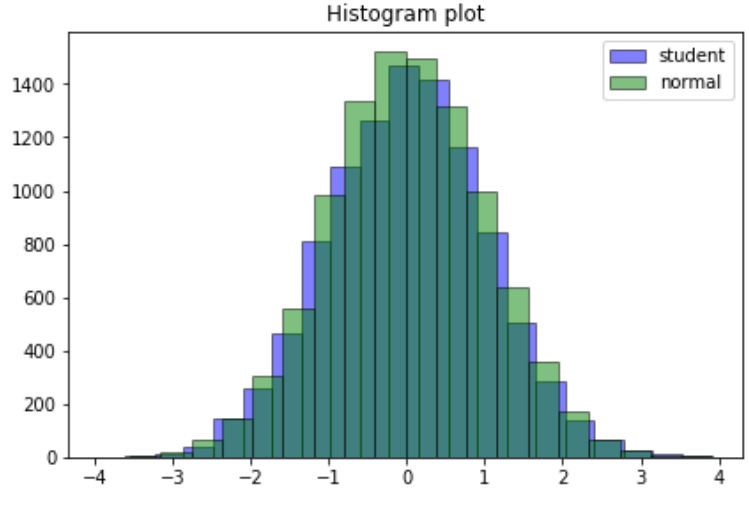


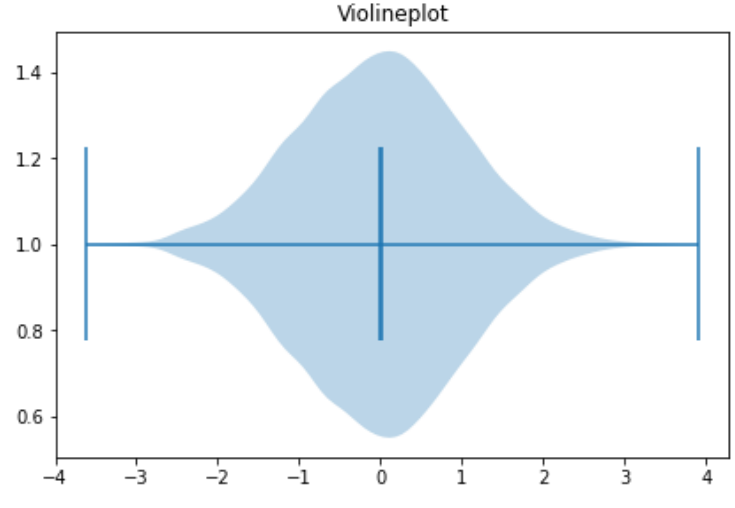


In [5]:
```
qq = stats.probplot(student, plot=plt)
```

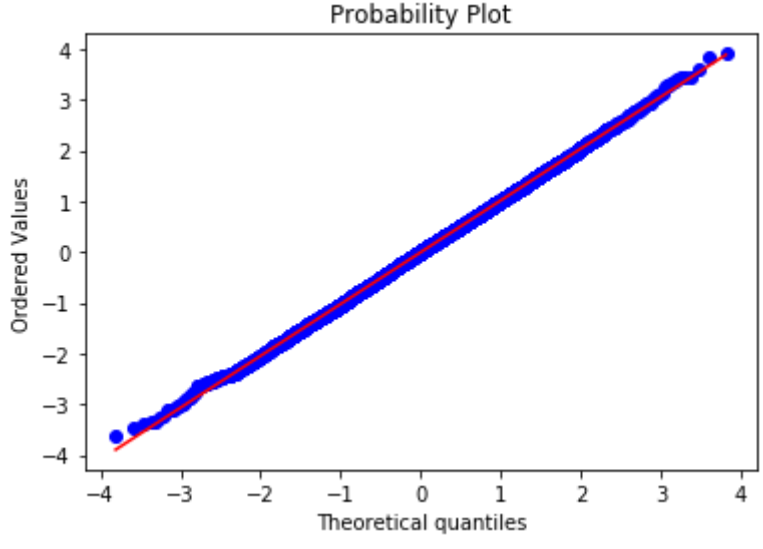


In [6]:
```
student_norm = gen_studet(100, 10000)['observation']
fig = plt.figure(figsize=(15, 15))
plt.subplot(321)
plt.title("Histogram plot")
plt.hist(student_norm, bins=BINS, alpha=0.5, label='student', color='b', ed
gecolor='k')
plt.hist(normal, bins=BINS, alpha=0.5, label='normal', color='g', edgecolor
='k')
plt.gca().legend(('student','normal'))
plt.subplot(322)
plt.title("Violineplot")
plt.violinplot(student_norm, vert=False, widths=0.9, showmeans=True, showex
trema=True, showmedians=True)
plt.show()
```



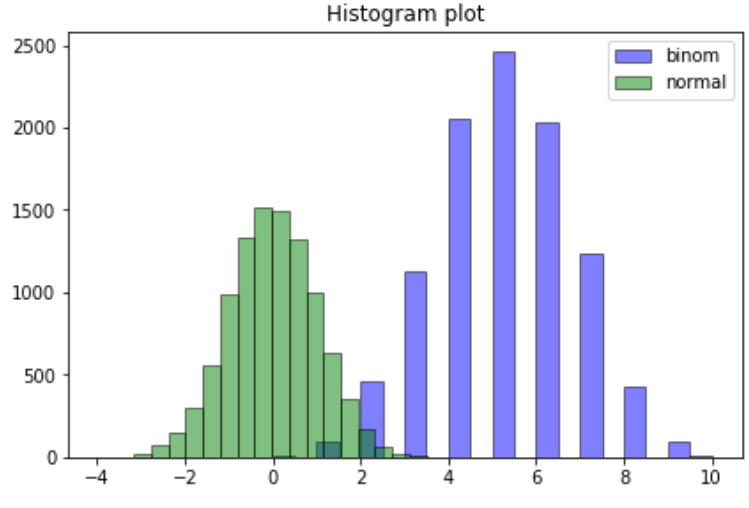


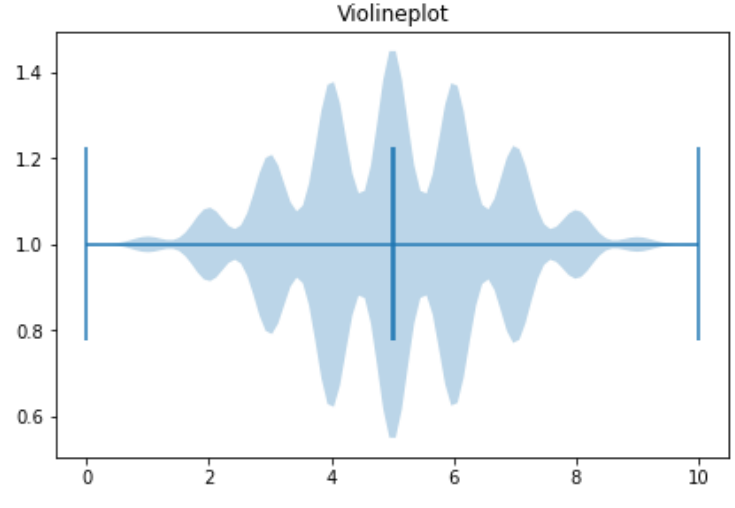
In [7]:
```
qq = stats.probplot(student_norm, plot=plt)
```



In [9]:

```
binom = generate_distribution('binom')['observation']
fig = plt.figure(figsize=(15, 15))
plt.subplot(321)
plt.title("Histogram plot")
plt.hist(binom, bins=BINS, alpha=0.5, label='binomial', color='b', edgecolo
r='k')
plt.hist(normal, bins=BINS, alpha=0.5, label='normal', color='g', edgecolor
='k')
plt.gca().legend(('binom','normal'))
plt.subplot(322)
plt.title("Violineplot")
plt.violinplot(binom, vert=False, widths=0.9, showmeans=True, showextrema=T
rue, showmedians=True)
plt.show()
```



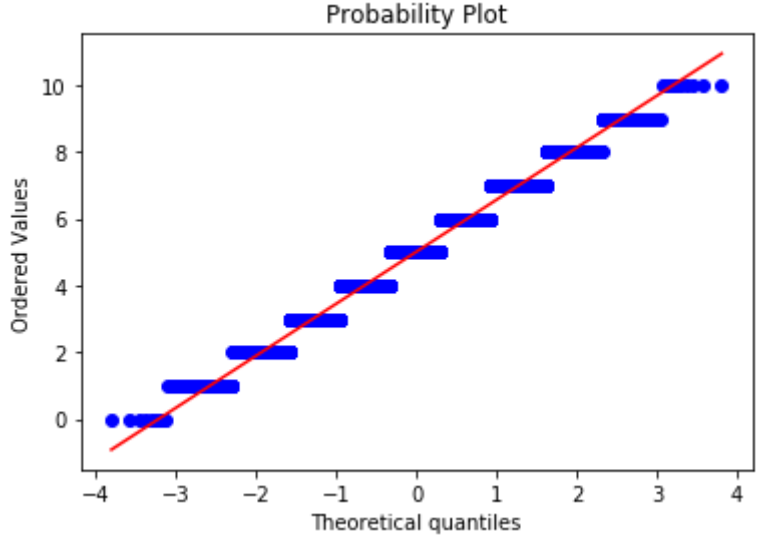


In [10]:

```
# more of a humanly understandable exaplanation
from collections import Counter
c = Counter(binom)
keys = sorted([i for i in c])
for k in keys:
    print("{}:{}".format(k, c[k]))
```

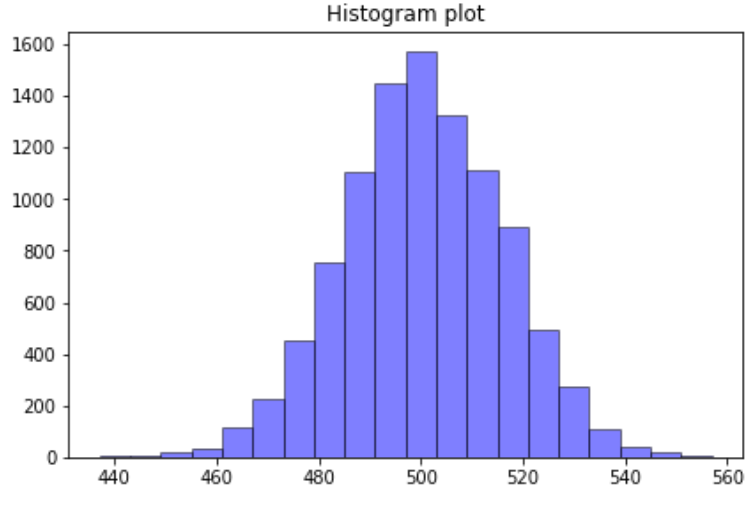
```
0:9
1:98
2:459
3:1124
4:2050
5:2459
6:2034
7:1236
8:428
9:92
10:11
```

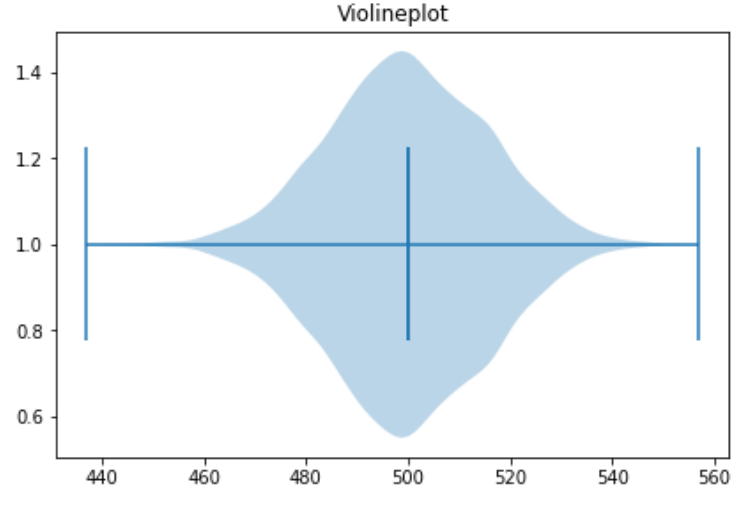
In [11]:
```
qq = stats.probplot(binom, plot=plt)
```



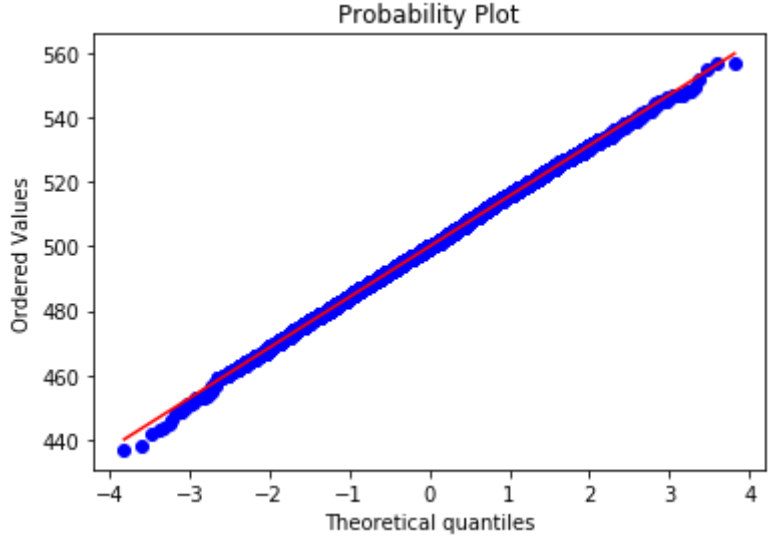
In [12]:

```
binom_normal = gen_binom(1000, 0.5)['observation']
fig = plt.figure(figsize=(15, 15))
plt.subplot(321)
plt.title("Histogram plot")
plt.hist(binom_normal, bins=BINS, alpha=0.5, label='binomial', color='b', e
dgecolor='k')
plt.subplot(322)
plt.title("Violineplot")
plt.violinplot(binom_normal, vert=False, widths=0.9, showmeans=True, showex
trema=True, showmedians=True)
plt.show()
```



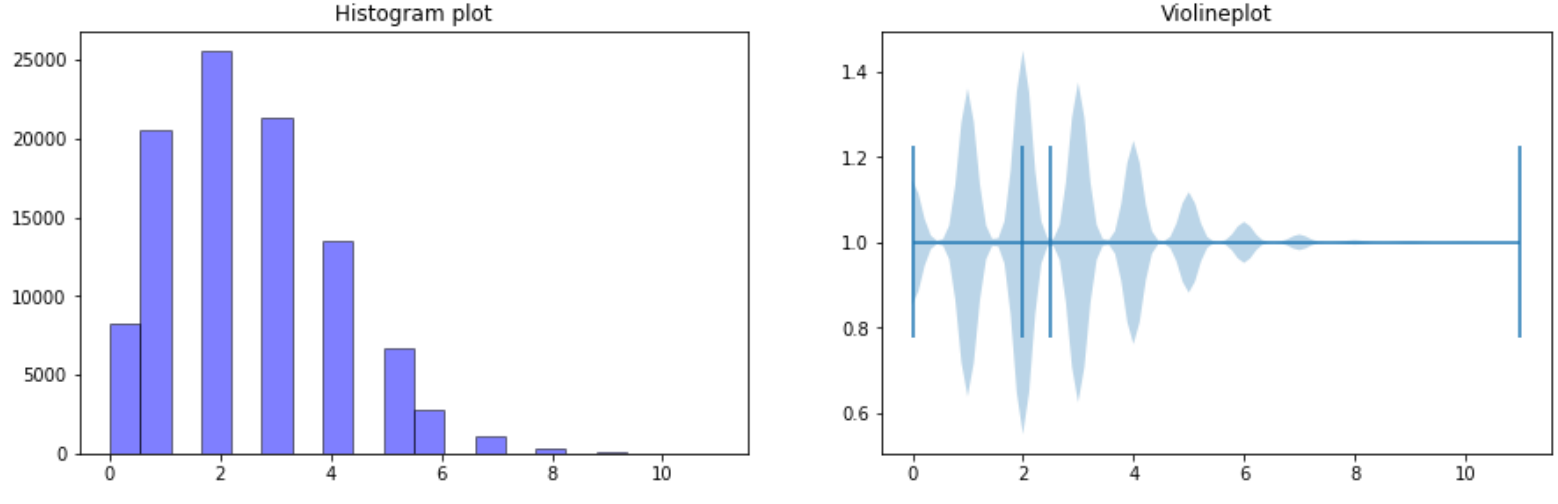


In [13]:
```
qq = stats.probplot(binom_normal, plot=plt)
```



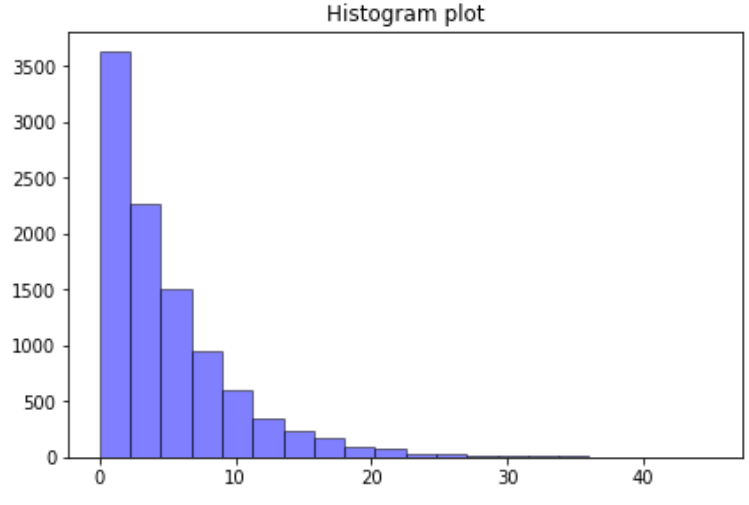
In [20]:

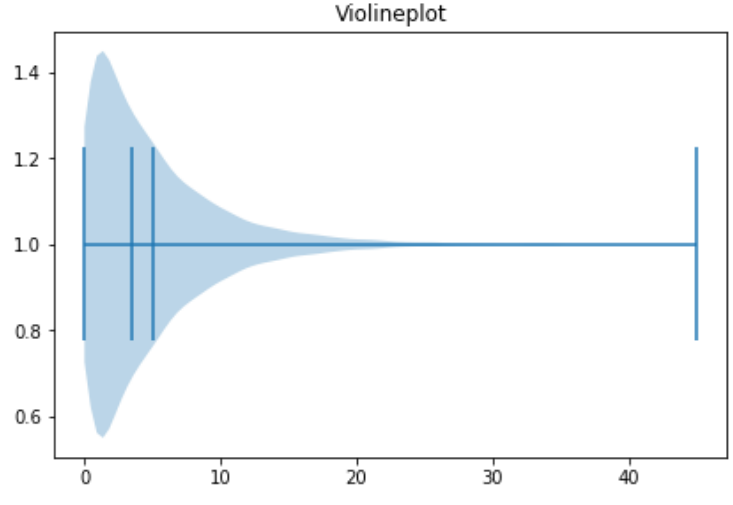
```
poisson_bigger = gen_poisson(2.5, 100000)['observation']
fig = plt.figure(figsize=(15, 15))
plt.subplot(321)
plt.title("Histogram plot")
plt.hist(poisson_bigger, bins=BINS, alpha=0.5, label='poisson', color='b',
edgecolor='k')
# plt.hist(normal, bins=BINS, alpha=0.5, label='normal', color='g', edgecol
or='k')
# plt.gca().legend(('poisson','normal'))
plt.subplot(322)
plt.title("Violineplot")
plt.violinplot(poisson_bigger, vert=False, widths=0.9, showmeans=True, show
extrema=True, showmedians=True)
plt.show()
```



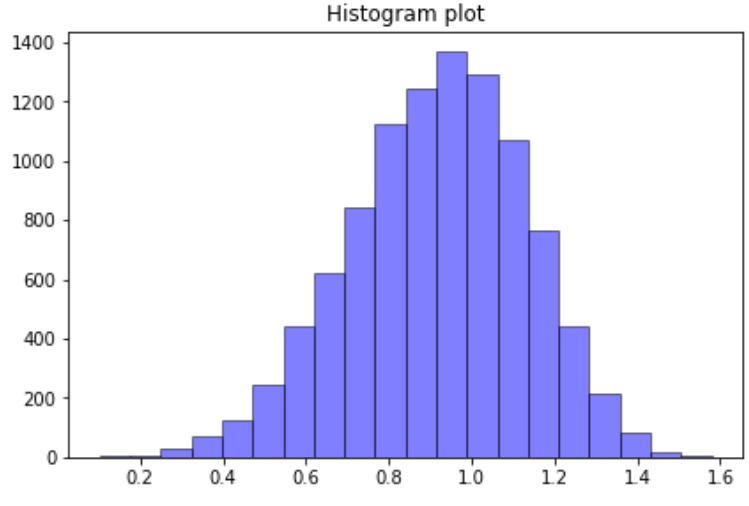
In [17]:

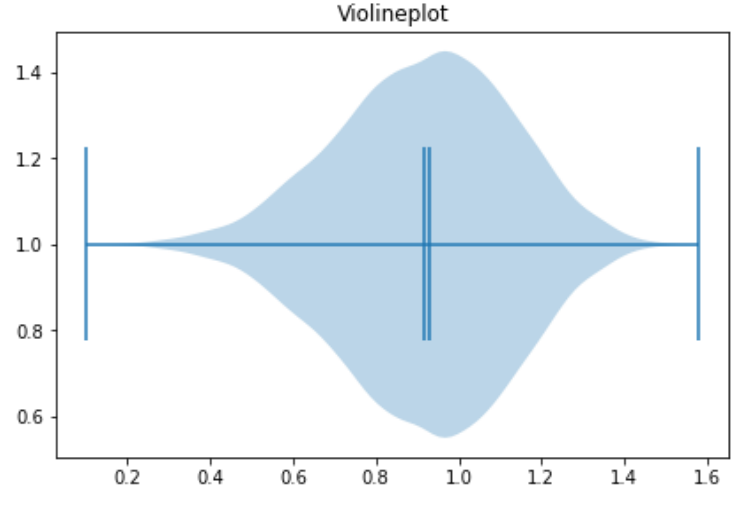
```
exp = generate_distribution('exp')['observation']
fig = plt.figure(figsize=(15, 15))
plt.subplot(321)
plt.title("Histogram plot")
plt.hist(exp, bins=BINS, alpha=0.5, label='exponential', color='b', edgecol
or='k')
plt.subplot(322)
plt.title("Violineplot")
plt.violinplot(exp, vert=False, widths=0.9, showmeans=True, showextrema=Tru
e, showmedians=True)
plt.show()
```





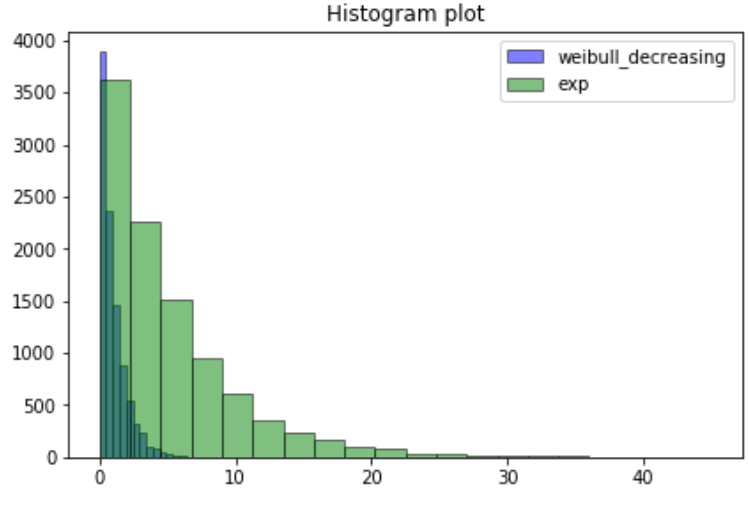
In [20]:
```
# Weibull distribution. Variant of Poisson/Exponential. The event frequency
 might change
weibull = generate_distribution('weibull')['observation']
fig = plt.figure(figsize=(15, 15))
plt.subplot(321)
plt.title("Histogram plot")
plt.hist(weibull, bins=BINS, alpha=0.5, label='weibull', color='b', edgecol
or='k')
plt.subplot(322)
plt.title("Violineplot")
plt.violinplot(weibull, vert=False, widths=0.9, showmeans=True, showextrema
=True, showmedians=True)
plt.show()
```

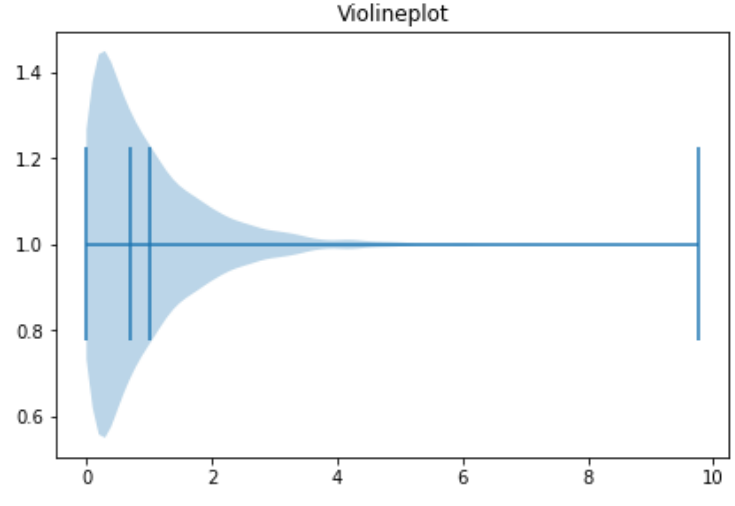




In [21]:

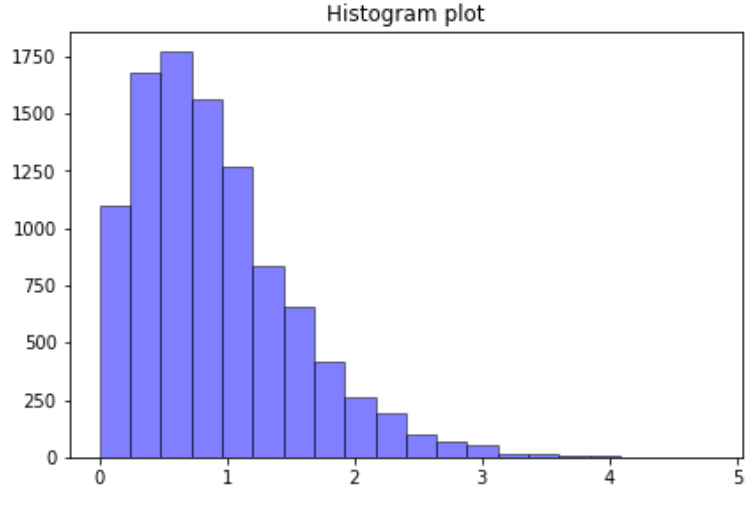
```
weibull_exp = gen_weibull(1)['observation']
fig = plt.figure(figsize=(15, 15))
plt.subplot(321)
plt.title("Histogram plot")
plt.hist(weibull_exp, bins=BINS, alpha=0.5, label='weibull', color='b', edg
ecolor='k')
plt.hist(exp, bins=BINS, alpha=0.5, label='normal', color='g', edgecolor=
'k')
plt.gca().legend(('weibull_decreasing','exp'))
plt.subplot(322)
plt.title("Violineplot")
plt.violinplot(weibull_exp, vert=False, widths=0.9, showmeans=True, showext
rema=True, showmedians=True)
plt.show()
```

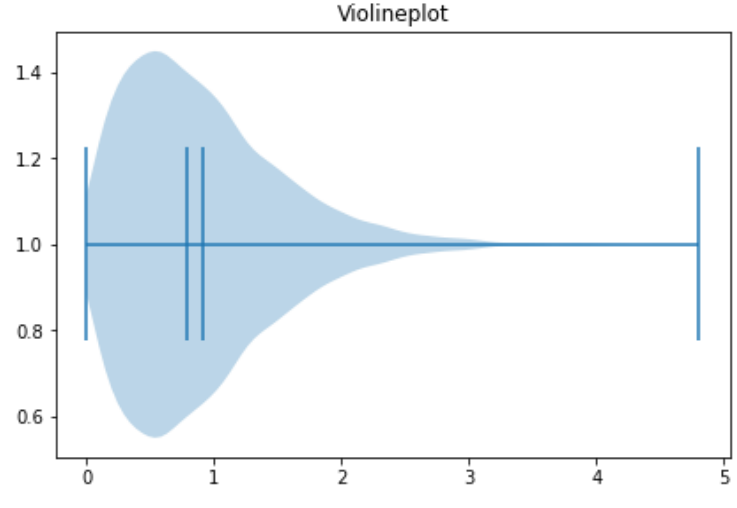




In [27]:

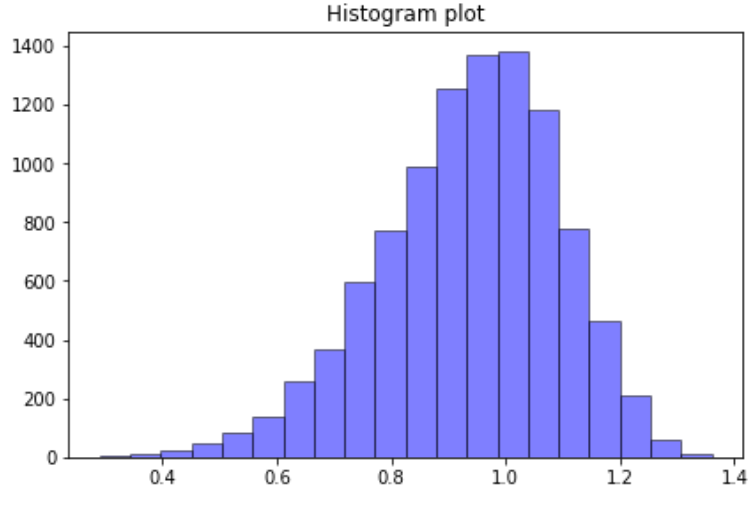
```
weibull_decreasing = gen_weibull(1.5)['observation']
fig = plt.figure(figsize=(15, 15))
plt.subplot(321)
plt.title("Histogram plot")
plt.hist(weibull_decreasing, bins=BINS, alpha=0.5, label='weibull', color=
'b', edgecolor='k')
# plt.hist(normal, bins=BINS, alpha=0.5, label='normal', color='g', edgecol
or='k')
# plt.gca().legend(('weibull_middle','normal'))
plt.subplot(322)
plt.title("Violineplot")
plt.violinplot(weibull_decreasing, vert=False, widths=0.9, showmeans=True,
showextrema=True, showmedians=True)
plt.show()
```

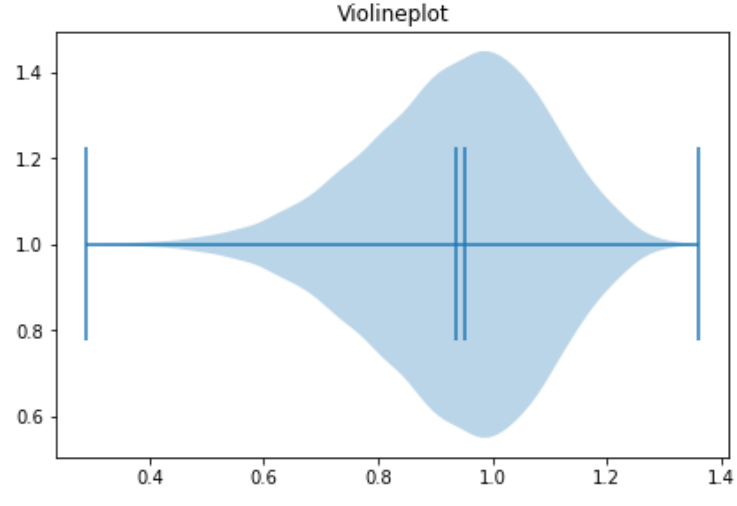




In [22]:

```
weibull_increasing = gen_weibull(7.1)['observation']
fig = plt.figure(figsize=(15, 15))
plt.subplot(321)
plt.title("Histogram plot")
plt.hist(weibull_increasing, bins=BINS, alpha=0.5, label='weibull', color=
'b', edgecolor='k')
plt.subplot(322)
plt.title("Violineplot")
plt.violinplot(weibull_increasing, vert=False, widths=0.9, showmeans=True,
showextrema=True, showmedians=True)
plt.show()
```





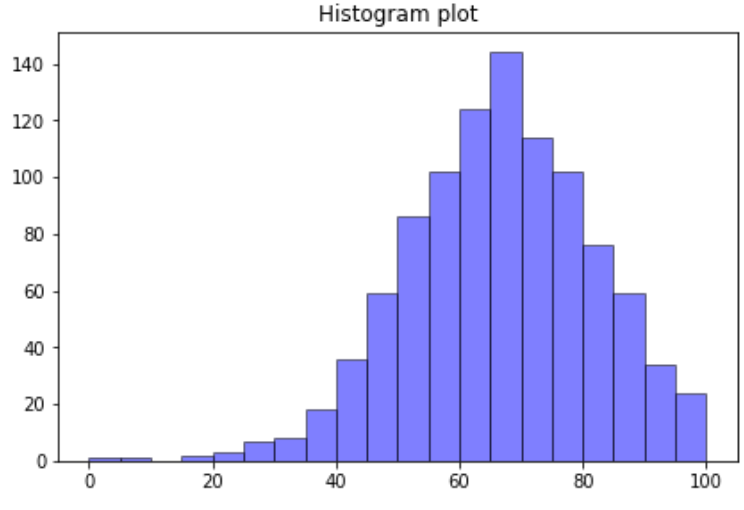
Useful links

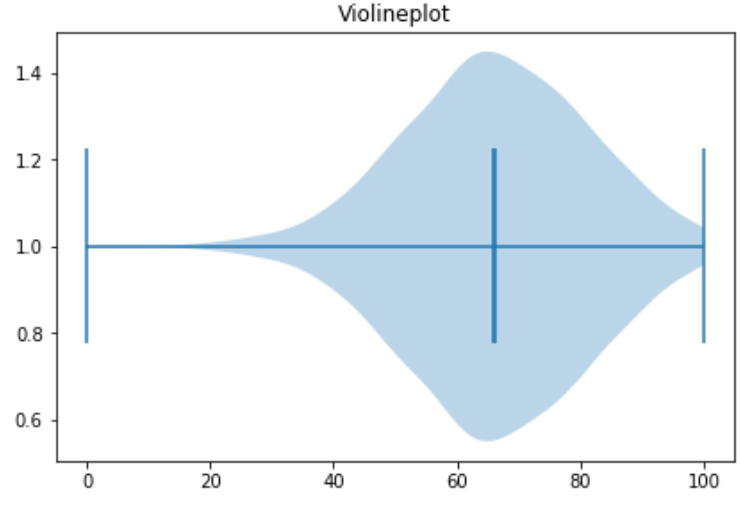
1. distribution list: https://en.wikipedia.org/wiki/List_of_probability_distributions (https://en.wikipedia.org/wiki/List_of_probability_distributions)
2. statistical moments overview: https://codeburst.io/2-important-statistics-terms-you-need-to-know-in-data-science-skewness-and-kurtosis-388fef94eeaa (https://codeburst.io/2-important-statistics-terms-you-need-to-know-in-data-science-skewness-and-kurtosis-388fef94eeaa)
3. short visualiozation: https://tekmarathon.com/2015/11/13/importance-of-data-distribution-in-training-machine-learning-models/ (https://tekmarathon.com/2015/11/13/importance-of-data-distribution-in-training-machine-learning-models/)
4. qqplot: https://en.wikipedia.org/wiki/Q%E2%80%93Q_plot (https://en.wikipedia.org/wiki/Q%E2%80%93Q_plot)
5. recap: https://www.analyticsvidhya.com/blog/2017/09/6-probability-distributions-data-science/ (https://www.analyticsvidhya.com/blog/2017/09/6-probability-distributions-data-science/)

In [33]:
```
# Tasks:
data1 = pd.read_csv('data/StudentsPerformance.csv')
what_is_this_distirution_1 = data1["math score"]
what_is_this_distirution_1.head(10)
```

Out[33]:
```
0    72
1    69
2    90
3    47
4    76
5    71
6    88
7    40
8    64
9    38
Name: math score, dtype: int64
```

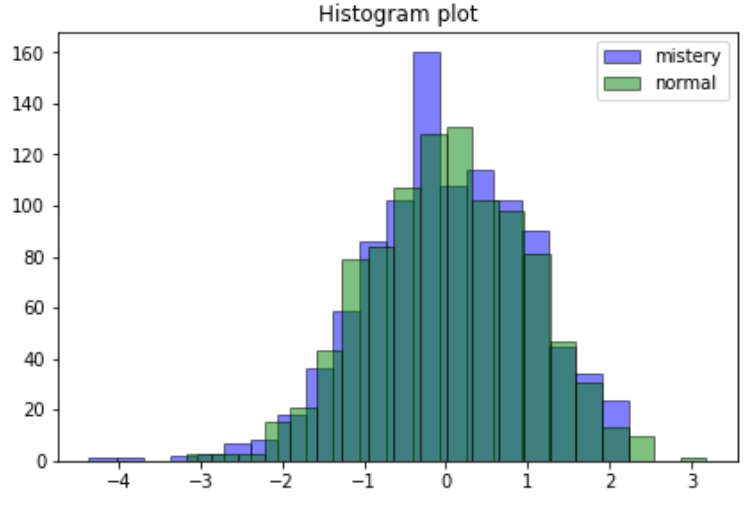
In [35]:
```
fig = plt.figure(figsize=(15, 15))
plt.subplot(321)
plt.title("Histogram plot")
plt.hist(what_is_this_distirution_1, bins=BINS, alpha=0.5, label='poisson',
 color='b', edgecolor='k')
plt.subplot(322)
plt.title("Violineplot")
plt.violinplot(what_is_this_distirution_1, vert=False, widths=0.9, showmean
s=True, showextrema=True, showmedians=True)
plt.show()
```

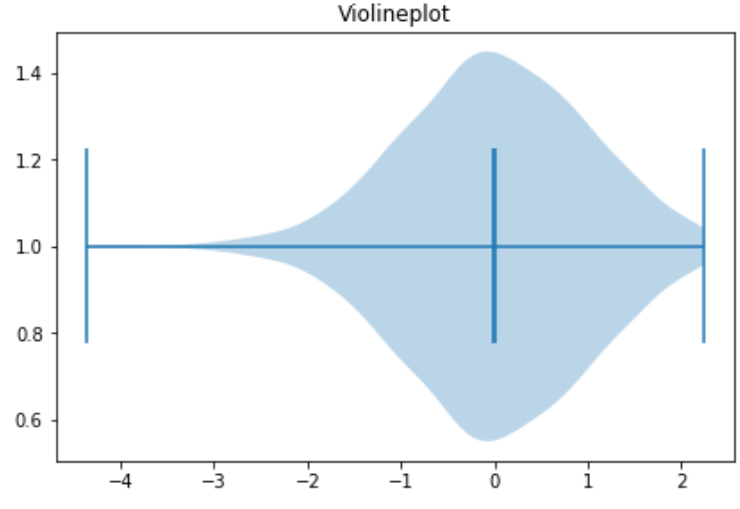




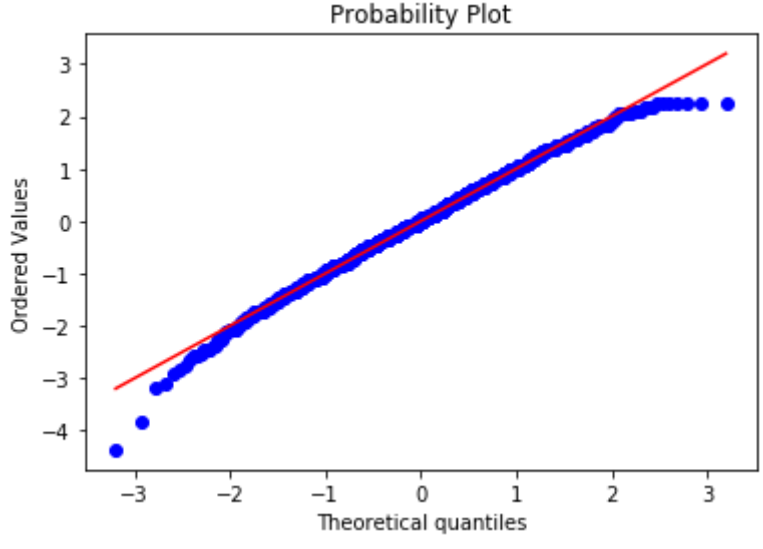
In [37]:

```
mu = np.mean(what_is_this_distirution_1)
sigma = np.std(what_is_this_distirution_1)
what_is_this_distirution_1_normalized = what_is_this_distirution_1.apply(la
mbda x: (x - mu)/sigma)
fig = plt.figure(figsize=(15, 15))
plt.subplot(321)
plt.title("Histogram plot")
plt.hist(what_is_this_distirution_1_normalized, bins=BINS, alpha=0.5, label
='poisson', color='b', edgecolor='k')
plt.hist(gen_normal(0, 1, 1000)['observation'], bins=BINS, alpha=0.5, label
='normal', color='g', edgecolor='k')
plt.gca().legend(('mistery','normal'))
plt.subplot(322)
plt.title("Violineplot")
plt.violinplot(what_is_this_distirution_1_normalized, vert=False, widths=0.
9, showmeans=True, showextrema=True, showmedians=True)
plt.show()
```





In [31]:
```
qq = stats.probplot(what_is_this_distirution_1_normalized, plot=plt)
```

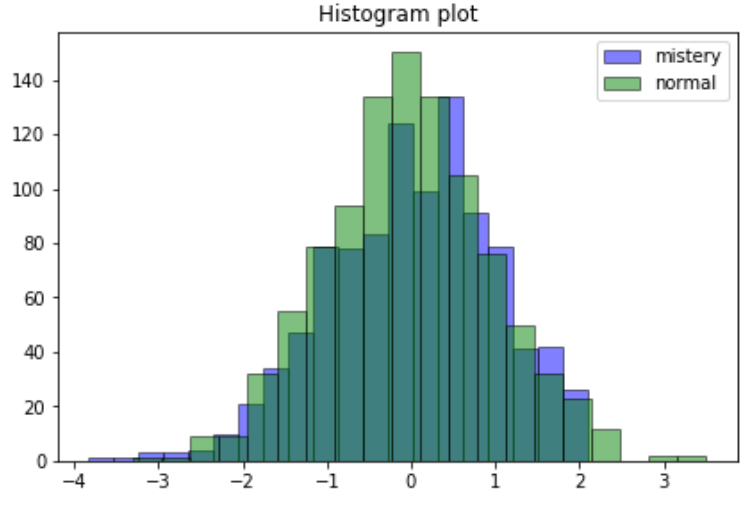


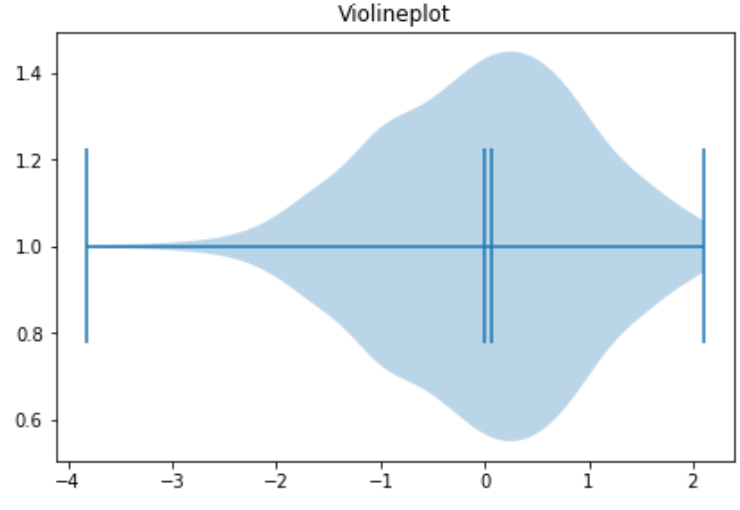
In [3]:

```
data2 = pd.read_csv('data/StudentsPerformance.csv')['writing score']
data3 = pd.read_csv('data/open-data-website-traffic.csv')['Socrata Session
s']
data4 = pd.read_csv('data/open-data-website-traffic.csv')['Socrata Bounce R
ate']
data5 = pd.read_csv('data/HorseKicksDeath.csv')['C1']
```

In [5]:

```
mu = np.mean(data3)
sigma = np.std(data2)
data2 = data3.apply(lambda x: (x - mu)/sigma)
fig = plt.figure(figsize=(15, 15))
plt.subplot(321)
plt.title("Histogram plot")
plt.hist(data2, bins=BINS, alpha=0.5, label='poisson', color='b', edgecolor
='k')
plt.hist(gen_normal(0, 1, 1000)['observation'], bins=BINS, alpha=0.5, label
='normal', color='g', edgecolor='k')
plt.gca().legend(('mistery','normal'))
plt.subplot(322)
plt.title("Violineplot")
plt.violinplot(data2, vert=False, widths=0.9, showmeans=True, showextrema=T
rue, showmedians=True)
plt.show()
```





used data links:

1. https://www.kaggle.com/venky73/predicting-student-percentage/data (https://www.kaggle.com/venky73/predicting-student-percentage/data)
2. https://www.kaggle.com/cityofLA/los-angeles-open-data-website-traffic (https://www.kaggle.com/cityofLA/los-angeles-open-data-website-traffic)