Jordan almond guessing contest analysis

Backgroud

The Law of Large Numbers (https://en.wikipedia.org/wiki/Law_of_large_numbers) (LLN) is an important underpinning of modern statistics. In general, the LLN states that given enough identical and independent measurements, the average measurement will tend towards the true value. One way to think about the LLN in practice is to consider the concept of the "wisdom of the crowd", which stipulates that a group of people will collectively come up with a better estimate of a quantity than a single person.

We can test the law of large numbers by answering the questions:

1) Will a crowd guess the correct almonds in a jar?

2) Will a crowd provide a closer guess on the number of almonds in a sealed jar than any single person?

We can perform this analysis and try to answer this question in pretty much any statistical package. Below is an example of one route using the python programming language in the form of a Jupyter Notebook (http://jupyter.org/). The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text and promotes open science and data analysis.

Experimental Design

- A jar was filled with a known number of almonds and sealed.
- The jar was placed in the coffee nook in the DES office for approximately one week.
- A note was placed with the jar asking people to guess how many almonds were in the jar and to send me their guess.

Data review and visualization

Setup the programming environment and import necessary python packages for the analysis

In [2]:
```
%matplotlib inline
import altair as alt
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from scipy.stats import shapiro, wilcoxon, ttest_1samp, probplot, t
pd.set_option('precision', 0)
```

Read in the guesses into a pandas dataframe for analysis.

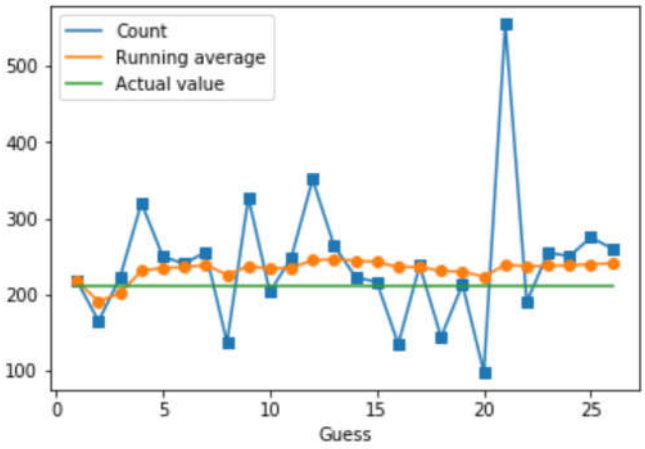
In [5]:
```
filename = 'https://raw.githubusercontent.com/OneGneissGuy/jordan_almond_guess/master/jordan_almonds_count.txt'
data = pd.read_csv(filename, delimiter='\t', index_col=[0])
# View the first few gusses
data.head()
```

Out[5]:

	Count	Running average	Actual value
Guess			
1	217	217	212
2	165	191	212
3	222	201	212
4	320	231	212
5	250	235	212

View the raw guesses and the running average of the counts.

In [6]:
```
plot = data['Count'].plot(marker='s',legend=True)
plot = data['Running average'].plot(marker='o',legend=True)
plot = data['Actual value'].plot(marker='None',legend=True)
```



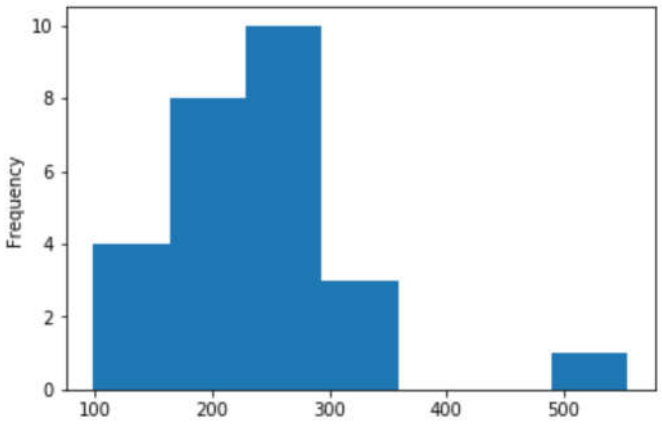
Summarize the raw data and identify outliers.

In [11]:
```
summary_stats = data['Count'].describe()
data_range = summary_stats['max'] -  summary_stats['min']
IQR = summary_stats['75%']- summary_stats['25%']
print("Number of samples =", len(data))
print("range \t", data_range)
print("IQR \t", IQR)
print("Extreme interval ", summary_stats['25%']-IQR*3, "-", IQR*3+summary_stats['75
%'],)
print("Outlier interval ", summary_stats['25%']-IQR*1.5, "-", IQR*1.5+summary_stats
['75%'],)
print(summary_stats)
```

```
Number of samples = 26
range    457.0
IQR      53.5
Extreme interval  45.0 - 419.5
Outlier interval  125.25 - 339.25
count     26
mean     240
std       88
min       98
25%      206
50%      239
75%      259
max      555
Name: Count, dtype: float64
```

Visualize the distribution of the guesses

In [7]:
```
plot = data['Count'].plot(kind='hist', bins=7)
```



Clearly the data is positively skewed (towards the highest guess of 555). We can filter out the most extreme value which is easily identified by both *parametric* (+/- 3 x standard deviation) and *non-parametric* methods (>3 * Inter-quartile range +/- 25/75th) and visualize the data again.

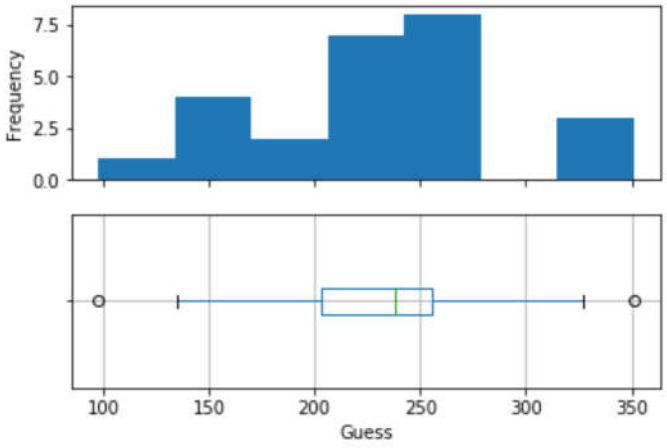
In [12]:
```
filtered_data = data['Count'][data['Count']<data['Count'].max()]
print("Number of samples =", len(filtered_data))
# Describe the filtered dataset
filtered_data.describe()
```

```
Number of samples = 25
```

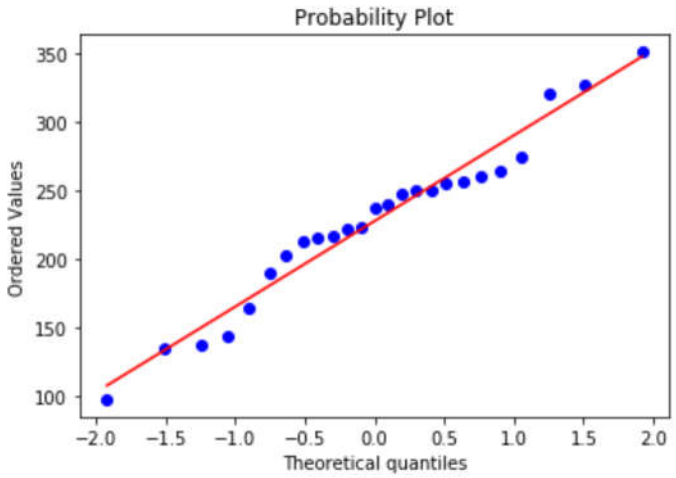
Out[12]:
```
count     25
mean     228
std       61
min       98
25%      203
50%      238
75%      256
max      351
Name: Count, dtype: float64
```

In [13]:
```
# Visualize the filtered data set
x_f = filtered_data.values
f, ax = plt.subplots(2, 1, sharey=False, sharex=True)
ax[0].hist(x_f, bins=7)
# generate hist and box plots to graphically inspect the samples distribution for n
ormality
df = pd.DataFrame(data=x_f, columns=[""])
df.boxplot(vert=False, ax=ax[1])
ylab = ax[0].set_ylabel('Frequency')
xlab = ax[1].set_xlabel('Guess')
print("Crowd mean difference =", np.mean(filtered_data-212))
```

```
Crowd mean difference = 15.88
```



In [14]:
```
import pylab
qq = probplot(x_f, dist="norm", plot=pylab)
```



With the exclusion of the extremely high guess, the average has moved towards the actual value, as one would expect. Visually the distribution is more mound-shaped and symmetrical about the mean and the q-q probability plot shows the data falls along the line, indicating normality of the dataset, even with two outliers remaining. So, vsually, the data appears as if it could have been sampled from a normal population, but is it? Let's run some tests for normality and find out!

Testing for normality

Perform the Shapiro-Wilk test for normality, which tests the null hypothesis that the sample (groups guesses) came from a normal distribution. This tests is especially appropriate for small sample sizes, as we have here. Assume significance level of `0.05` or `95%`, unless otherwise noted.

In [18]:
```
print("Shapiro Wilks test p-value =", shapiro(x_f)[1])
```

```
Shapiro Wilks test p-value = 0.5569809079170227
```

Because the *p*-value > `0.05`, we fail to reject the null hypothesis that the sample distribution was taken from a normal population so parametric methods for hypothesis testing are appropriate. In the event that we rejected the null hypothesis we would need to use a non-parametric hypothesis test, such as the Wilcoxon signed rank test.

Hypothesis testing

Parametric approach

Given the small number of guesses (N<30) and the assummption of normality, the one sample t-test can be used to test the null hypothesis that the crowd correctly guessed the actual number of almonds in the jar. The alternative hypothesis is that the crowd did not guess the actual number. First, we need to calculate the critical values. We can reject the null hypothesis if our test statistic is greater than or less than the critical values.

In [16]:
```
#upper limit test stat
t.ppf(q=0.975, df=len(x_f-1))
# The t-distribution is symmetrical aobut the mean, so the lower limit is equal but
opposite in sign of upper limit
```

Out[16]:
```
2.059538552753294
```

In [17]:
```
ttest_1samp(x_f, 212) # Return t-statistic and p-value
```

Out[17]:
```
Ttest_1sampResult(statistic=1.3073712930615178, pvalue=0.20347359183227254)
```

Based on the test statistic and *p*-value we fail reject the null hypothesis that the crowd guessed the correct amount of almonds in the jar and can say that the crowds guess was not significantly different than the actual amount of almonds in the jar.

Non-parametric approach

Use the Wilcoxon Signed Rank test,the non-parametric analogue to the one sample t-test and test the null hypotheisis that the crowds median guess is equal to the actual value.

In [18]:
```
#pass in the differences between the filtered guesses and the actual number of almo
nds and use the wilcox method to discard ties
wilcoxon(x_f-212, zero_method='wilcox')
```

Out[18]:
```
WilcoxonResult(statistic=109.0, pvalue=0.1499848685122849)
```

As in the parametric test, we fail reject the null hypothesis (indicated by high *p*-value) that the crowd guessed the correct amount of almonds in the jar and can say that the crowds guess was not significantly different than the actual amount of almonds in the jar.

Conclusions

Based on the statisitics, **we can conclude that the crowds guess is not significantly different from the actual amount.** Pretty cool that that a crowd could, statistically speaking, guess the right number of almonds in the jar (actually, the crowd didn't guess the wrong number of almonds in the jar), even with the paltry sample size.

But did the crowd do a better job than any one person? No, Elaine was able to guess within one almond of the correct amount, whereas the crowd's mean difference was 15 almonds.So, we can't really say the crowd was any wiser than an individual here. This might change if we had 100 or 1000 guesses.*

Just because a result is statistically signficant, it doesnt mean its practically significant (and there's a whole field of stats that deals with that, but I won't delve into here). No experiment is perfect, and often times there are many sources of bias. Lucky for us they dont seem to have played a large role this time. I've listed a few sources of bias for your consideration.

Potential sources of bias

- Non-independence of measurements (participant collusion!)
- Jar shape (tapered neck)
- Location (DES, coffee/microwave nook vs. an entry way)
- Volunteer participation
- Variable almond size and shape
- Measurement technique (random guess vs counting a layer and extrapolating)
- (Hidden) Tissue under the lid
- More?

Thanks again for participating and please reach out if you have any questions or would like to know more, or point out if I missed anything.