descriptive_stats_project

February 15, 2015

Questions for Investigation

This experiment will require the use of a standard deck of playing cards. This is a deck of fifty-two cards divided into four suits (spades, hearts, diamonds, and clubs), each suit containing thirteen cards (Ace, numbers 2-10, and face cards Jack, Queen, and King). You can use either a physical deck of cards for this experiment or you may use a virtual deck of cards such as that found on random.org (http://www.random.org/playing-cards/). For the purposes of this task, assign each card a value: The Ace takes a value of 1, numbered cards take the value printed on the card, and the Jack, Queen, and King each take a value of 10.

1. First, create a histogram depicting the relative frequencies of the card values.
2. Now, we will get samples for a new distribution. Shuffle your deck of cards and draw three cards from it. (You will be sampling from the deck without replacement.) Record the cards that you have drawn and the sum of the three cards' values. Repeat this sampling procedure a total of at least thirty times.
3. Let's take a look at the distribution of the card sums. Report descriptive statistics for the samples you have drawn. Include at least two measures of central tendency and two measures of variability.
4. Create a histogram of the sampled card sums you have recorded. Compare its shape to that of the original distribution. How are they different, and can you explain why this is the case?
5. Make some estimates about values you will get on future draws. Within what range will you expect approximately 90% of your draw values to fall? What is the approximate probability that you will get a draw value of at least 20? Make sure you justify how you obtained your values.

1 The Rubric:

[1][2]

Criteria

Responses to Project Questions

Question 1: Plotting a histogram of card values

Question 2: Obtain samples from a deck of cards

Question 3: Report descriptive statistics regarding sample taken

Question 4: Plotting a histogram of sampled values

Question 5: Making estimates based on the sampled distribution

Does Not Meet Specifications

Meets Specifications

The histogram does not accurately reflect the card values' relative frequency distribution or no histogram is provided.

A histogram is provided that accurately reflects the card values' relative frequency distribution.

Sampled data is not provided, insufficient, or does not reflect the experiment being performed for the project.

At least thirty samples have been performed and the summed values from each sample have been reported in a submitted spreadsheet.

[1] https://docs.google.com/document/d/1gVCBsThTXBkUtMiccc4Rj5XT8I9h4ji0bpqTe2uUxRg/pub#

[2] https://docs.google.com/document/d/1gVCBsThTXBkUtMiccc4Rj5XT8I9h4ji0bpqTe2uUxRg/pub#

Two measures of central tendency and variability are not reported to describe the sample or are not computed correctly.

At least two measures of central tendency and two measures of variability are accurately reported to summarize and describe the samples taken for Question 2.

The histogram does not accurately reflect sampled values or no histogram is provided. No discussion of the shape of the distribution is provided or comparison is not well-justified.

A histogram accurately reflecting the sampled data is provided. Discussion of the shape is provided, including a comparison to that of the histogram of the original card values.

Estimates made for the prompted questions do not reflect the values obtained from the sample.

Estimates are made for the prompted questions that reflect the samples taken and their distribution.

Lets start with a rough draft of generating a deck of cards. Lets create 4 suites first. Lets create a dictionary of each suite in the deck named *deck*

```
In [32]: import matplotlib
         %matplotlib inline
         from matplotlib.pylab import hist

         import numpy as np
         from numpy import mean, std, var

         from pprint import pprint
```

```
In [13]: heart_suite = {"H": [str(i) + 'H' for i in ['A']+range(2,11)+['J', 'Q', 'K']]}

         diamond_suite = {"D": [str(i) + 'D' for i in ['A']+range(2,11)+['J', 'Q', 'K']]}

         club_suite = {"C": [str(i) + 'C' for i in ['A']+range(2,11)+['J', 'Q', 'K']]}

         spade_suite = {"S": [str(i) + 'S' for i in ['A']+range(2,11)+['J', 'Q', 'K']]}

         deck = (heart_suite, diamond_suite, club_suite, spade_suite)

         deck = {k:{value:num+1 for num, value in enumerate(cardict[k])} for cardict in deck for k in ca

         print deck
```

```
{'S': {'QS': 12, '2S': 2, '3S': 3, '4S': 4, '5S': 5, 'JS': 11, 'KS': 13, 'AS': 1, '6S': 6, '7S': 7, '10S
```

Lets create a function to pick a pseudo random card out of 4 randomly selected cards from the dictionary of 4 suites.

```
In [14]: def random_deck_card(deck):
             selection = ({t[0]:deck[t[0]][t[1]] \
             for t in \
             [(k, random.choice(deck[k].keys())) \
             for k in deck.iterkeys()]})
             res = random.choice(selection.keys())
             return res, selection[res]
```

Lets create a function to generate as asked in the question, a list of 30 samples with each of their sums, from 3 random selections using random_deck_card and *deck*

```
In [15]: def sum_shuffled_cards(deck, mini_range, grand_range):
             return [(lambda x: (x, sum([i[1] for i in x])))([random_deck_card(deck) \
             for _ in xrange(mini_range)]) for _ in xrange(grand_range)]
```

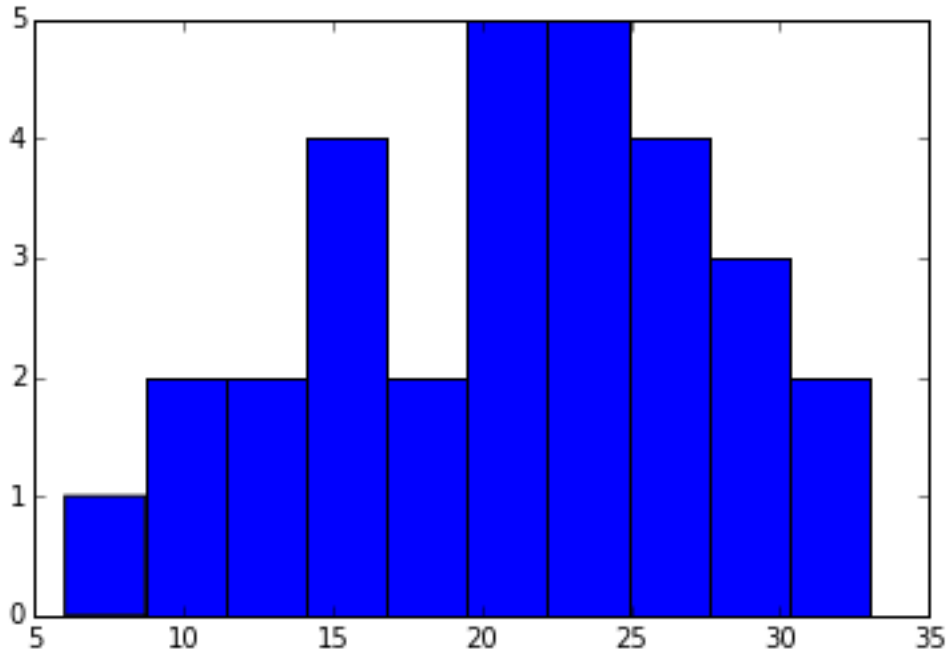
```
In [26]: #the historgram of this is almost uniform. for instance;
         all_cards = reduce(lambda x,y: (x+y), [[i for i in xrange(len(dct))] for dct in deck.itervalues

         print hist(all_cards)
```

```
(array([ 8.,  4.,  4.,  4.,  4.,  8.,  4.,  4.,  4.,  8.]), array([  0. ,   1.2,   2.4,   3.6,   4.8,
        10.8,  12. ]), <a list of 10 Patch objects>)
```

Lets create a histogram of 30 samples from 3 randomly picked cards each time.

```
In [33]: shuffled_cards = sum_shuffled_cards(deck, 3, 30)
         print hist(zip(*shuffled_cards)[1])
```

```
(array([ 1.,  2.,  2.,  4.,  2.,  5.,  5.,  4.,  3.,  2.]), array([  6. ,   8.7,  11.4,  14.1,  16.8,
        30.3,  33. ]), <a list of 10 Patch objects>)
```



Lets see the shuffled cards

```
In [34]: pprint(shuffled_cards)
```

```
[([('H', 8), ('C', 3), ('D', 5)], 16),
 ([('C', 10), ('H', 2), ('H', 11)], 23),
 ([('H', 3), ('S', 2), ('H', 4)], 9),
 ([('C', 10), ('D', 1), ('H', 10)], 21),
 ([('D', 8), ('H', 12), ('H', 12)], 32),
 ([('H', 4), ('S', 2), ('D', 9)], 15),
 ([('D', 1), ('H', 4), ('C', 1)], 6),
 ([('D', 2), ('S', 12), ('D', 12)], 26),
 ([('S', 3), ('D', 2), ('H', 8)], 13),
 ([('H', 10), ('H', 8), ('S', 8)], 26),
 ([('C', 6), ('S', 9), ('C', 3)], 18),
```

```
([('D', 7), ('C', 10), ('S', 12)], 29),
([('S', 1), ('C', 1), ('S', 8)], 10),
([('S', 2), ('D', 9), ('D', 11)], 22),
([('D', 3), ('C', 6), ('S', 7)], 16),
([('D', 11), ('S', 6), ('S', 11)], 28),
([('D', 11), ('D', 9), ('C', 6)], 26),
([('S', 7), ('D', 10), ('H', 12)], 29),
([('H', 8), ('D', 5), ('D', 6)], 19),
([('D', 6), ('S', 11), ('H', 6)], 23),
([('C', 5), ('D', 9), ('D', 2)], 16),
([('C', 11), ('H', 12), ('H', 10)], 33),
([('H', 2), ('C', 10), ('S', 8)], 20),
([('H', 7), ('S', 13), ('D', 4)], 24),
([('S', 11), ('D', 10), ('H', 3)], 24),
([('S', 3), ('H', 7), ('C', 3)], 13),
([('H', 8), ('C', 2), ('D', 11)], 21),
([('C', 13), ('S', 11), ('C', 3)], 27),
([('H', 5), ('D', 6), ('C', 9)], 20),
([('C', 8), ('S', 11), ('H', 5)], 24)]
```

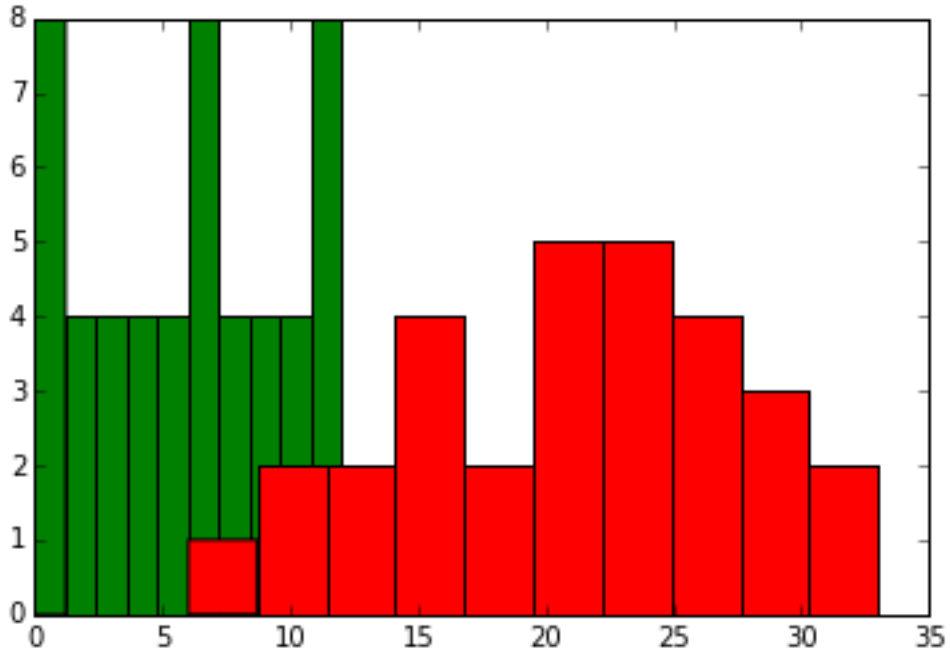
Lets analyze the sample distribution of samples.

In [47]:
```
sum_samples = zip(*shuffled_cards)[1]
```

```
        #standard error
        all_cards_std = np.std(all_cards)
        se = all_cards_std/np.sqrt(len(sum_samples))

        #mean
        sum_samples_mean = mean(sum_samples)
        print "Average of list of Sum of all samples: {}\n".format(sum_samples_mean)
        print hist(sum_samples)
        #std deviation
        sum_sample_std = sum([(x - sum_samples_mean)**2 for x in sum_samples])/float(len(sum_samples)-
        print "\nUnbiased STD DEV of list of sum of all samples std: {}\n".format(sum_sample_std)
        print "Standard Error of distribution of 30 sample size of 3 samples each: {}\n".format(se)

        print "Average of list of Sum of all cards: {}\n".format(mean(all_cards))
        print hist(all_cards)
        print "STD DEV Of all Cards in a suite: {}\n".format(all_cards_std)


        #the histrogram plot - averaging at around 20.9666666667
        print hist(zip(*shuffled_cards)[1])
```

```
Average of list of Sum of all samples: 20.9666666667

(array([ 1.,  2.,  2.,  4.,  2.,  5.,  5.,  4.,  3.,  2.]), array([  6. ,   8.7,  11.4,  14.1,  16.8,
        30.3,  33. ]), <a list of 10 Patch objects>)

Unbiased STD DEV of list of sum of all samples std: 45.9643678161

Standard Error of distribution of 30 sample size of 3 samples each: 0.683130051064
```

```
Average of list of Sum of all cards: 6.0

(array([ 8.,  4.,  4.,  4.,  4.,  8.,  4.,  4.,  4.,  8.]), array([  0. ,   1.2,   2.4,   3.6,   4.8,
        10.8,  12. ]), <a list of 10 Patch objects>)
STD DEV Of all Cards in a suite: 3.74165738677

(array([ 1.,  2.,  2.,  4.,  2.,  5.,  5.,  4.,  3.,  2.]), array([  6. ,   8.7,  11.4,  14.1,  16.8,
        30.3,  33. ]), <a list of 10 Patch objects>)
```



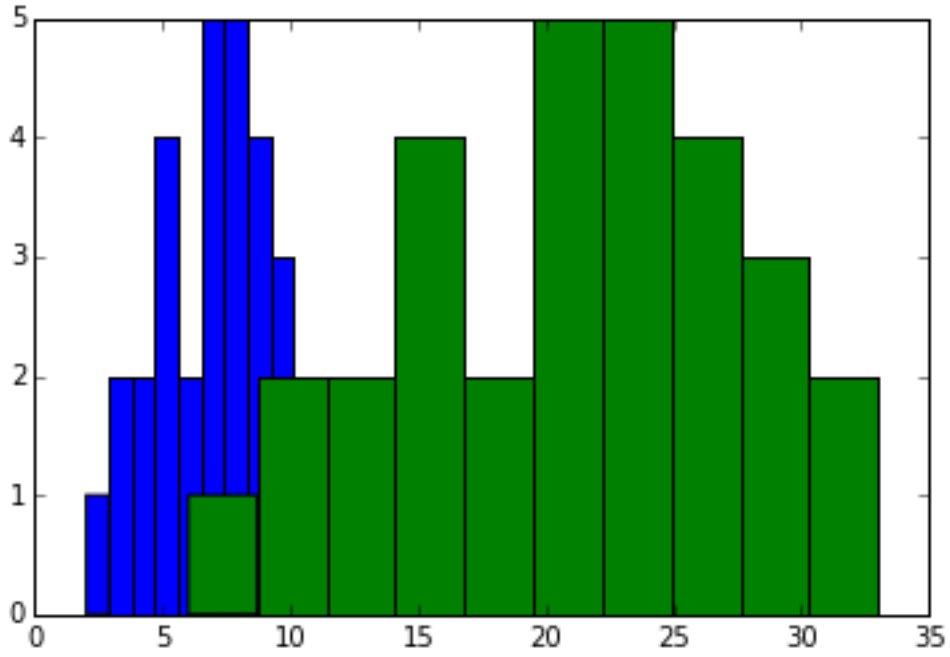
There are 2 histograms. Green: Distribution of all cards in a Deck- The distribution is uniform because each card is exactly 4 in number. Re: Distribution shows almost a normal curve. This is because we took 30 samples fo 3 random picks of cards per turn and it shows the most likely/frequent value it sums upto, averaging at around 20.96 ~ 21 - which mean on an average the 3 rounds we picked, summed up to around 21. Had we taken not **sum** but **MEAN** and then sampled them, it would have been more normal, smoother, skinnier.

Now lets, calculate the SE, Mean and Standard Deviation of this sample of mean plot against the SE, Mean and Standard Deviation of the plot for summed samples. It is shone in the graph below of what happens if we dont sum but take Mean of each times and then take 30 samples of each mean, what will be its mean.

```
In [48]: shuffled_cards_sample_mean = [mean([i[1] for i in each_stack[0]]) for each_stack in shuffled_ca
         print hist(shuffled_cards_sample_mean)
         print hist(sum_samples)

         print "\nAverage of list of Sum of all samples: {}\n".format(sum_samples_mean)
         print "\nUnbiased STD DEV of list of sum of all samples std: {}\n".format(sum_sample_std)
         print "Standard Error of distribution of 30 sample size of 3 samples each: {}\n".format(se)
```

```
        #mean and std deviation
        mean_shuffled_cards_sample_mean = mean(shuffled_cards_sample_mean)
        print "Average of list of Mean of all samples: {}\n".format(mean_shuffled_cards_sample_mean)
        shuffled_cards_sample_mean_std = sum([(x - mean_shuffled_cards_sample_mean)**2 for x in shuffle
        print "\nUnbiased STD DEV of list of sum of all samples std: {}\n".format(shuffled_cards_sample
        mean_se = std(shuffled_cards_sample_mean)/np.sqrt(len(shuffled_cards_sample_mean))
        print "Standard Error of distribution of 30 sample size of 3 samples each: {}\n".format(mean_se
```

```
(array([ 1.,  2.,  2.,  4.,  2.,  5.,  5.,  4.,  3.,  2.]), array([  2. ,   2.9,   3.8,   4.7,   5.6,
       10.1,  11. ]), <a list of 10 Patch objects>)
(array([ 1.,  2.,  2.,  4.,  2.,  5.,  5.,  4.,  3.,  2.]), array([  6. ,   8.7,  11.4,  14.1,  16.8,
       30.3,  33. ]), <a list of 10 Patch objects>)

Average of list of Sum of all samples: 20.9666666667


Unbiased STD DEV of list of sum of all samples std: 45.9643678161

Standard Error of distribution of 30 sample size of 3 samples each: 0.683130051064

Average of list of Mean of all samples: 6.98888888889


Unbiased STD DEV of list of sum of all samples std: 5.10715197957

Standard Error of distribution of 30 sample size of 3 samples each: 0.405664622574
```



As we can see from above, the difference between sampling:

1. Means and

2. Sums is the thickness of distribution.

We can note:

1. Standard Error of distribution of 30 sample size of mean of 3 samples each: 0.405664622574 while,
2. Standard Error of distribution of 30 sample size of sum of 3 samples each: 0.683130051064;

which means SE of Mean Samples is lower than of Sum Samples. and Average has come down from ˜21 to 6.98888888889 which is alot closer to Average of list of Sum of all cards: 6.0;

However, In the function **sum_shuffled_cards** If we raise the mini_range from 3 to say 30 and grand_range from 30 to 1000, the curve will be even smoother and skinnier with narrower tails meaning, the average will become more defined and precise.

In []: