STAT 207 Homework 4 [50 points] - Solutions

Quantitative Variable Descriptive Analytics and Probability

Due: Friday, February 10 by noon (11:59 am) CST

Package Imports

We'll import three Python packages that we've used so far and will need for this assignment. Those packages are pandas, matplotlib.pyplot, and seaborn.

Run the cell provided below to access the functions in these packages.

In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

Case Study: Groundhog Day

On Groundhog Day (February 2), a famous groundhog in Pennsylvania called Punxsutawney Phil emerges from his burrow. Spectators watch and observe whether he sees his shadow or not. The legend says that if he sees his shadow, winter will continue for six more weeks; if not, spring is coming early. In 2023, Phil saw his shadow. But, does the data support the legend? In this Case Study, we'll examine historical data to see what the data supports.

1. Read in the Data [3 points]

Read in the groundhog.csv data. Check the data for missing data, make any necessary adjustments, and print the first few rows of the data frame.

In [2]:
```
df = pd.read_csv('groundhog.csv')
```

In [3]:
```
df.dtypes
```

Out[3]:
```
Year                                          int64
Phil                                         object
February Average Temperature                float64
February Average Temperature (Northeast)    float64
February Average Temperature (Midwest)      float64
February Average Temperature (Pennsylvania) float64
March Average Temperature                   float64
March Average Temperature (Northeast)       float64
March Average Temperature (Midwest)         float64
March Average Temperature (Pennsylvania)    float64
dtype: object
```

In [4]:
```
df['Phil'].unique() # No Record is basically an NA, so...
```

Out[4]:
```
array(['No Record', 'Full Shadow', 'No Shadow', 'Partial Shadow'],
      dtype=object)
```

In [5]:
```
df = pd.read_csv('groundhog.csv', na_values = ['No Record'])
df.head()
```

Out[5]:

	Year	Phil	February Average Temperature	February Average Temperature (Northeast)	February Average Temperature (Midwest)	February Average Temperature (Pennsylvania)	March Average Temperature	March Average Temperature (Northeast)	Temp (M
0	1895	NaN	26.60	15.6	21.9	17.0	39.97	27.6	
1	1896	NaN	35.04	22.2	33.5	26.6	38.03	25.3	
2	1897	NaN	33.39	23.6	34.7	27.9	38.79	32.0	
3	1898	Full Shadow	35.37	24.8	33.3	26.7	41.05	38.0	
4	1899	NaN	25.50	18.1	22.2	20.0	37.63	29.3	

In [6]:
```
df = df.dropna()
```

In [7]:
```
df.head()
```

Out[7]:

	Year	Phil	February Average Temperature	February Average Temperature (Northeast)	February Average Temperature (Midwest)	February Average Temperature (Pennsylvania)	March Average Temperature	March Average Temperature (Northeast)	Temp (M
3	1898	Full Shadow	35.37	24.8	33.3	26.7	41.05	38.0	
5	1900	Full Shadow	30.76	21.4	27.5	24.1	41.27	26.0	
6	1901	Full Shadow	29.86	16.0	26.2	18.9	40.71	30.8	
8	1903	Full Shadow	28.42	24.5	31.3	28.0	42.21	40.4	
9	1904	Full Shadow	31.59	15.0	28.2	19.2	41.76	29.8	

In [8]:
```
df.shape
```

Out[8]:
```
(116, 10)
```

2. Phil's Outcomes [9 points]

For this problem, we will explore the results of Phil's experiment by focusing on the variable Phil.

a) First, what is the sample space of possible outcomes for Phil's experiment?

The sample space is the set of all possible outcomes. The result could be: {Full Shadow}, {Partial Shadow}, {No Shadow}.

b) Then, we'd like to summarize the observed outcomes for this experiment from the data. Calculate the

relative frequencies for each of the possible results when Phil emerges from his burrow.

In [9]:
```
df['Phil'].value_counts(normalize=True)
```

Out[9]:
```
Full Shadow       0.862069
No Shadow         0.129310
Partial Shadow    0.008621
Name: Phil, dtype: float64
```

c) Which result is most common? Least common?

Based on these results, choose whether you'd like to filter your data to remove uncommon outcomes. Explain your decision.

The Full Shadow is the most common, occuring 86% of the time. No Shadow is more rare, occuring 13% of the time. Partial Shadow occurs only 0.9% of the time (corresponding to only 1 year in my 116 years of complete data).

Because I only have one observation, which is not very much data, corresponding to a partial shadow, I am going to remove it from my dataset.

In [10]:
```
df = df[df['Phil'] != 'Partial Shadow']
df.shape
```

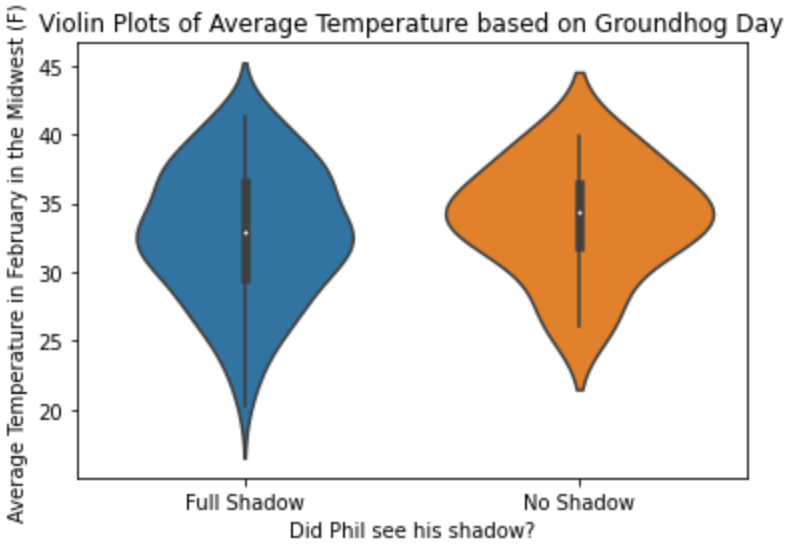
Out[10]:
```
(115, 10)
```

3. Midwest Temperatures [17 points]

Because we live in the Midwest, we'd like to observe whether Phil's prediction about winter seems to be accurate, based on the average temperatures recorded in the Midwest. The following steps will help you gather information to answer this question.

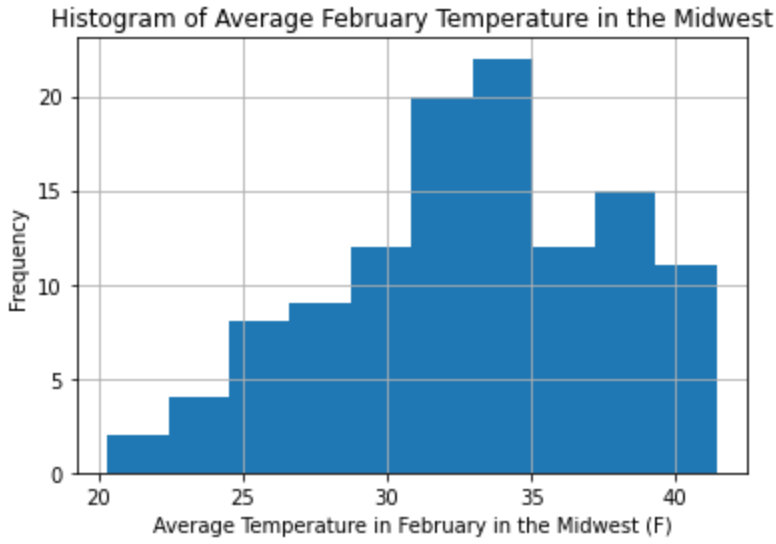
a) Generate side-by-side violin plots to display the average temperature in February for the Midwest, based on the result of Punxsutawney Phil's shadow.

In [11]:
```
sns.violinplot(x = 'Phil', y = 'February Average Temperature (Midwest)', data = df)
plt.xlabel('Did Phil see his shadow?')
plt.ylabel('Average Temperature in February in the Midwest (F)')
plt.title('Violin Plots of Average Temperature based on Groundhog Day')
plt.show()
```



In [12]:

```
df['February Average Temperature (Midwest)'].hist()
plt.xlabel('Average Temperature in February in the Midwest (F)')
plt.ylabel('Frequency')
plt.title('Histogram of Average February Temperature in the Midwest')
plt.show()
```



b) For each of the results of Phil's experiment, calculate one measure of center and one measure of spread.

In [13]:

```
def summary_med(series):
    s_median = series.median()
    s_Q1 = series.quantile(q=0.25)
    s_Q3 = series.quantile(q=0.75)
    s_iqr = s_Q3-s_Q1
    value = [s_Q1, s_median, s_Q3, s_iqr]
    index = ['Q1', 'Median', 'Q3', 'IQR']
    return pd.DataFrame({'value': value}, index = index)
def summary_mean(series):
    return pd.DataFrame({'value': [series.mean(), series.std()]}, index = ['mean', 'standa
```

In [14]:

```
summary_med(df[df['Phil'] == 'Full Shadow']['February Average Temperature (Midwest)'])
```

Out[14]:

	value
Q1	29.550

	value
Median	32.950
Q3	36.575
IQR	7.025

In [15]:
```
summary_med(df[df['Phil'] == 'No Shadow']['February Average Temperature (Midwest)'])
```

Out[15]:

	value
Q1	31.95
Median	34.30
Q3	36.40
IQR	4.45

In [16]:
```
summary_mean(df[df['Phil'] == 'Full Shadow']['February Average Temperature (Midwest)'])
```

Out[16]:

	value
mean	32.799000
standard deviation	4.807664

In [17]:
```
summary_mean(df[df['Phil'] == 'No Shadow']['February Average Temperature (Midwest)'])
```

Out[17]:

	value
mean	33.853333
standard deviation	3.987994

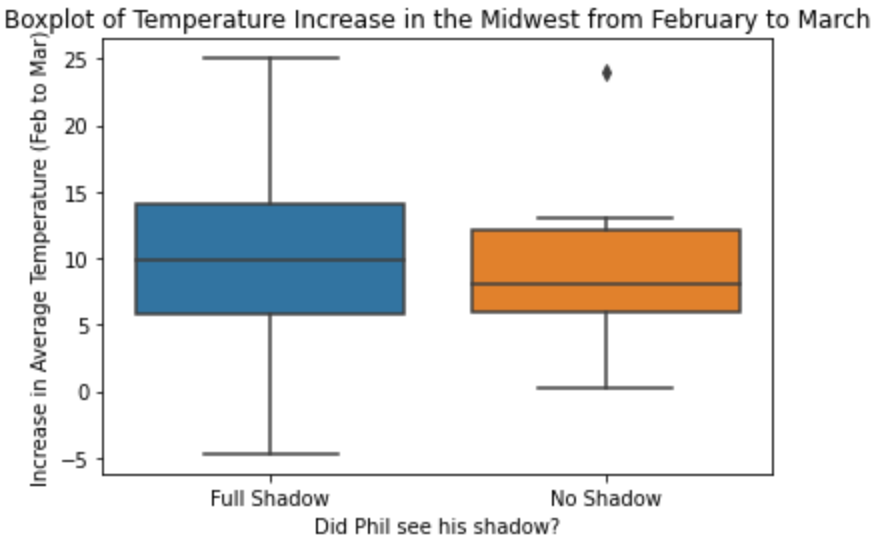
Based on the violin plots above, the distribution seems to be plausibly symmetric. From the histogram, it looks a little more bimodal and/or skewed left. However, both of these graphs are not definitive, so either set of measures of center or spread will be accepted. Note: you do need to select compatible measures of center and spread.

c) We've already examined the February temperatures, but a friend thinks that how quickly winter comes may actually relate to how much the average temperatures increase between February and March. Calculate a new variable for the increase in the average temperature in the Midwest from February to March, and include this variable in our data frame.

In [18]:
```
df['FebToMar'] = df['March Average Temperature (Midwest)'] - df['February Average Temperat
```

d) Generate side-by-side boxplots for the temperature increase between February and March calculated in part **c**.

In [19]:
```
sns.boxplot(x='Phil', y='FebToMar', data = df)
plt.xlabel('Did Phil see his shadow?')
plt.ylabel('Increase in Average Temperature (Feb to Mar)')
plt.title('Boxplot of Temperature Increase in the Midwest from February to March')
plt.show()
```



e) Finally, we'll consider what all of this information means. If you don't like the idea of winter being extended, would you want Phil to see his shadow on Groundhog Day? Explain.

Thinking critically about our data, do you have any questions about the data? Do you notice any limitations to results that you can draw from this data?

There are a number of ways to interpret the information from the graphs here, so there isn't one correct answer.

Some details to notice: the increase in average temperature from February to March seems to roughly be larger if Phil saw his shadow. The typical temperature in February seems to be somewhat larger in February if Phill does not see his shadow. At the end of the day, I would probably say that it really doesn't make too much of a difference, at least in the Midwest.

I might wonder if Phil's predictions are only limited to Pennsylvania. That is, how generalizable are Phil's predictions? Maybe they don't extend throughout the Midwest, which explains why there isn't a clear difference between the two outcomes. I also might want to do further analysis to determine if the differences in temperatures are different enough to be substantially different (in some way). Finally, I might wonder about how the average temperatures are calculated (both across time and across a geographic region), or how Phil's viewing of his shadow is recorded.

Case Study: Poker Hand

In poker, a player is dealt 5 cards from a standard deck of 52 cards. One type of hand that can be dealt is a three of a kind, where a player has three cards of the same rank (face), one card of a different rank, and the last card of a different rank from the other two. Note that this is distinct from a full house, where we have three of a kind (three cards of the same rank) and a pair (two cards of the same rank).

4. Actual Probability [6 points]

First, using our counting rules, calculate the actual probability if drawing a three of a kind poker hand.

Show your work and calculations.

DENOMINATOR: How many possible hands of size k=5 can we create from a

class of deck of size n=52?

Because our experiment involves randomly sampling k=5 cards from a deck of size $n = 52$ without replacement where the order/way/allocation of the k=5 cards in the deck doesn't matter, this means that we can use the combination function to help us calculate the all possible combinations of size $k = 5$ that we could make out of a deck of $n = 52$ cards.

$$\binom{n}{k} = \binom{52}{5} = \frac{52!}{5!(52-5)!} = \frac{52\times51\times50\times49\times48\times47...\times1}{(5\times4\times3\times2\times1)(47\times46\times...\times1)} = \frac{52\times51\times50\times49\times48}{5\times4\times3\times2\times1} = 2598960$$

In [20]:
```
# first the denominator, which we saw in class
total_number_of_possible_hands=(52*51*50*49*48)/(5*4*3*2*1)
total_number_of_possible_hands
```

Out[20]:
```
2598960.0
```

NUMERATOR: How many hands (ie. 5 cards) will be a 'three of a kind'?

There are a few decisions we would need to make in creating a hand that is a 'three of a kind', and a few equivalent ways of making those decisions.

- Decision A: Which rank should be designated to have the three cards in the hand (let's call it the "three card" face)?
- Decision B: Which two faces should be designated to have a single card in the hand (let's call them "single card" faces)?
- Decision C: Which three distinct suits should be designated to the "three card" face? (Hint: There are 4 options here as only one card can be left out for a given face).
- Decision D: Which suit should be designated for the first "single card" face?
- Decision E: Which suit should be designated for the second "single card" face?

So thus, we need to enumerate all possibilities for Decision A, B, C, D, and E and then multiply them together to enumerate all possible ways to create a 'three of a kind' poker hand.

One alternate way to separate this set of decisions:

- Decision A: What rank should be designated to have the three cards in the hand?
- Decision B: What three distinct suits should be designated to the "three card" face?
- Decision C: Of the remaining 48 cards (after removing the 3 selected cards & the one card to prevent it from being a four of a kind), what 2 cards should I select?
- Adjustment D: From Decision C, how do I ensure that those two cards are of distinct faces? That is, how do I prevent the hand from inadvertently being a full house?

We'll start with the first calculation, and then use the alternate way to perform this calculation.

Decision A: How many possibilities to assign a face to the "three card" face?

First, we'll look at how many ways we can select our "three card" portion of the "three of a kind" hand. There are 13 possible faces, and so we will use 13.

Decision B: How many possibilities to assign two faces to the single cards?

Now, we want to consider how many ways we could assign faces for the two single cards. The two single cards are selected without replacement and the order doesn't matter, so we can use the combinations rule:

$$\binom{n}{k} = \binom{12}{2} = \frac{12!}{2!(12-2)!} = \frac{12 \times 11}{2} = 66$$

Decision C: How many possibilities to assign three suits to the "three card" face?

This question involves counting the number of combinations of $k = 3$ suits out of $n = 4$ possible suits. This is selected without replacement and order doesn't matter, so we can use the combinations rule:

$$\binom{n}{k} = \binom{4}{3} = \frac{4!}{3!(4-3)!} = 4$$

Decision D: How many possibilities to assign the suit to the first "single card" face?

This question involves counting the number of combinations of $k = 1$ suits out of $n = 4$ possible suits. You may be able to answer this question intuitively (there are 4 possible suits), or you can use a counting rule. This single suit is selected without replacement and order doesn't matter, so we can use the combinations rule:

$$\binom{n}{k} = \binom{4}{1} = \frac{4!}{1!(4-1)!} = 4$$

Note: the result of this calculation is equivalent to the one from Decision C. You can think about Decision C as selecting which suit to leave out of the three of a kind, and this one as selecting which suit to keep in, but the results of the enumeration will be the same.

Decision E: How many possibilities to assign the suit to the second "single card" face?

This is the same question as Decision D. Since the results of the suit for this second face card don't depend on any of the previous decisions, we can repeat the calculation from Decision D.

$$\binom{n}{k} = \binom{4}{1} = \frac{4!}{1!(4-1)!} = 4$$

Now, we can multiply all of these options together to calculate the number of possible distinct three of a kind hands.

In [21]:
```
number_of_three_kind_hands = 13 * 66 * 4 ** 3
print(number_of_three_kind_hands)
actual_prob_three_of_kind = number_of_three_kind_hands / total_number_of_possible_hands
print(actual_prob_three_of_kind)
```

```
54912
0.02112845138055222
```

Decision A: How many possibilities to assign a face to the "three card" face?

First, we'll look at how many ways we can select our "three card" portion of the "three of a kind" hand. There are 13 possible faces, and so we will use 13. This is the same as Decision A above.

Decision B: How many possibilities to assign three suits to the "three card" face?

This question involves counting the number of combinations of $k = 3$ suits out of $n = 4$ possible suits. This is selected without replacement and order doesn't matter, so we can use the combinations rule:

$$\binom{n}{k} = \binom{4}{3} = \frac{4!}{3!(4-3)!} = 4$$

This is the same as Decision C above.

Decision C: How many possibilities to get two cards for our last two cards?

This calculation is a bit tricky. We've already picked out our "three of a kind" hand. We know that we can't pick the remaining card to complete the "three of a kind", because then our hand would be a "four of a kind" hand. That means we have $52 - 4 = 48$ possible cards to pick from. We'll start by calculating the number of possibilities for our two additional cards. These cards are selected without replacement and order doesn't matter.

$$\binom{n}{k} = \binom{48}{2} = \frac{48!}{2!(48-2)!} = \frac{48 \times 47}{2} = 1128$$

Adjustment D: How do we adjust to be sure we have two unique face cards?

In Decision C, we made sure that we didn't inadvertently create a "four of a kind" hand. Our hand has three cards of the same rank, and two additional cards. Other than these cards not matching the same rank as our "three card" rank, we have not put any additional restrictions on the cards. However, there are two possible poker hands resulting from a hand with three cards of the same rank: a full house and three of a kind. How do we adjust to ensure that our last two cards selected are distinct? In this case, we can subtract the number of full house possibilities (calculated in class as 3744) from the total number of "three card" hands to result in just the "three of a kind" hands being counted.

This works because there are only two possible card hands that have three cards of the same rank, so the total number of "three of a kind" and "full house" hands must add up to the three cards of the same rank options.

$$13 \times 4 \times 1128 - 3744 = 54912$$

In [22]:
```
# decisions a-c
total_three_cards = 13 * 4 * 1128
total_three_cards
```

Out[22]:
```
58656
```

In [23]:
```
# adjustment d
total_three_kind = total_three_cards - 3744
total_three_kind
```

Out[23]:
```
54912
```

PUTTING IT ALL TOGETHER

Thus, the probability of drawing a 'three of a kind' hand is:

$$\frac{\text{number of ways to create a three of a kind hand}}{\text{number of ways to create a hand with 5 cards}} = \frac{54912}{2598960} = 0.0211$$

In [24]:
```
probability = total_three_kind/total_number_of_possible_hands
probability
```

Out[24]:
```
0.02112845138055222
```

5. Probability Estimate [14 points]

We have created two test hands below in the form of data frames.

1. **hand1** is a 'three of a kind' hand
2. **hand2** is not a 'three of a kind' hand

In [25]:
```
import pandas as pd
hand1 = pd.DataFrame({'face': ['J','J','2','9','J'],
                      'suit': ['heart', 'club', 'spade', 'club', 'diamond']})
hand1
```

Out[25]:

	face	suit
0	J	heart
1	J	club
2	2	spade
3	9	club
4	J	diamond

In [26]:
```
hand2 = pd.DataFrame({'face': ['J','J','Q','Q','J'],
                      'suit': ['heart', 'club', 'spade', 'club', 'diamond']})
hand2
```

Out[26]:

	face	suit
0	J	heart
1	J	club
2	Q	spade
3	Q	club
4	J	diamond

a) Create a function that returns True if a hand is a three of a kind and False if a hand is not three of a kind.

In [27]:
```
def is_three_of_a_kind (df, var='face'):
    if df.shape[0] != 5:
        return "Not a valid poker hand"
    else:
        counts = df[var].value_counts()
        if counts.min() == 1 and counts.max() == 3:
            return True
        else:
            return False
```

b) Confirm this function works using **hand1** and **hand2**.

In [28]:
```
is_three_of_a_kind(hand1)
```

Out[28]:
```
True
```

In [29]:
```
is_three_of_a_kind(hand2)
```

Out[29]:
```
False
```

We see that our function does correctly identify our three of a kind hand.

c) Using the cards.csv file, conduct one trial that does the following:

1. Draws a poker hand.
2. Checks if the hand is three of a kind.

In [30]:
```
cards = pd.read_csv("cards.csv")
pokerhand = cards.sample(5, replace=False)
is_three_of_a_kind(pokerhand)
```

Out[30]:
```
False
```

d) Now, extend your work from part c to repeat this 20,000 times.

Specifically, for this simulation:

1. Simulate drawing 20,000 different hands of poker. Each hand is randomly sampled without replacement. You can assume after each trial the previous cards are returned to the deck, so that you have a fresh 52 card deck at the start of each trial.
2. Test whether the hand drawn is a three of a kind.
3. Calculate the proportion of trials that were a three of a kind.

Warning: using print statements may cause your notebook to crash, depending on where the statements are located in your code.

In [31]:
```
# This will keep a tally of the hands that are three of a kind hand
# We will update this as we iterate through the for-loop.
tk_count = 0

# Number of trials/poker hands in the simulation.
n_iterations = 20000

# This will iterate through n_iterations
for i in range(n_iterations):
    # Draws the sample of 5 cards from the 'cards' dataframe.
    new_hand = cards.sample(5)

    # Tests whether the randomly sampled hand is a three of a kind.
    # 1 is added to the tk_count tally if it is True (ie. a three of a kind).
    # 0 is added to the tk_count tally if it is False (ie. not a three of a kind).
    tk_count += is_three_of_a_kind(new_hand)

approx_prob_three_of_kind=tk_count/n_iterations
approx_prob_three_of_kind
```

Out[31]:
```
0.02165
```

6. Comparing Probabilities [1 point]

What is the difference between your theoretical and estimated probabilities above?

In [32]:
```
actual_prob_three_of_kind - approx_prob_three_of_kind
```

Out[32]:
```
-0.0005215486194477779
```

Remember to keep all your cells and hit the save icon above periodically to checkpoint (save) your results

on your local computer. Once you are satisified with your results restart the kernel and run all (Kernel -> Restart & Run All). **Make sure nothing has changed**. Checkpoint and exit (File -> Save and Checkpoint + File -> Close and Halt). Follow the instructions on the Homework 4 Canvas Assignment to submit your notebook to GitHub.