CSPB 3022 - Craven - Introduction to Data Science Algorithms

Started on	Friday, 23 February 2024, 7:03 PM
State	Finished
Completed on	Friday, 23 February 2024, 8:57 PM
Time taken	1 hour 53 mins
Grade	**85.00** out of 100.00

Question 1

Correct

Mark 4.00 out of 4.00

Reference Sheet

Pandas

In the entries below suppose `df` is a DataFrame, `s` is a Series and `pd` is the Pandas package

Attribute	Description
`df.index`	Returns the index labels of the DataFrame.
`df.columns`	Returns the column labels of the DataFrame.
`df.shape` or `s.shape`	Returns the shape of a DataFrame or Series in the form number of rows, number of columns
`df.size` or `s.size`	Returns the total number of entries in a DataFrame or Series (number of rows times number of columns)

Function	Description
`df[col]`	Returns the column labeled `col` from `df` as a Series.
`df[[col1, col2]]`	Returns a DataFrame containing the columns labeled `col1` and `col2`.
`s.loc[rows]` or `df.loc[rows, cols]`	Returns a Series/DataFrame with rows (and columns) selected by their index values.
`s.iloc[rows]` or `df.iloc[rows, cols]`	Returns a Series/DataFrame with rows (and columns) selected by their positions.
`s.isnull()` or `df.isnull()`	Returns boolean Series/DataFrame identifying missing values
`df.info()`	displays name and type of each column, the number of non-null entries, and size of dataframe
`s.fillna(value)` or `df.fillna(value)`	Returns a Series/DataFrame where missing values are replaced by `value`
`df.drop(labels, axis)`	Returns a DataFrame without the rows or columns named `labels` along `axis` (either 0 or 1)
`df.rename(index=None, columns=None)`	Returns a DataFrame with renamed columns from a dictionary `index` and/or `columns`
`df.sort_values(by, ascending=True)`	Returns a DataFrame where rows are sorted by the values in columns `by`
`s.sort_values(ascending=True)`	Returns a sorted Series.
`s.unique()`	Returns a NumPy array of the unique values
`s.value_counts()`	Returns the number of times each unique value appears in a Series
`pd.merge(left, right, how='inner', left_on='a', right_on='b')`	Returns a DataFrame joining DataFrames `left` and `right` on the columns labeled `a` in the left database and 'b' in the right database; the join is of type `inner`
`df.pivot_table(index, columns, values=None, aggfunc='mean')`	Returns a DataFrame pivot table where columns are unique values from `columns` (column name or list), and rows are unique values from `index` (column name or list); cells are collected `values` using `aggfunc`. If `values` is not provided, cells are collected for each remaining column with multi-level column indexing.
`df.set_index(col)`	Returns a DataFrame that uses the values in the column labeled `col` as the row index.
`df.reset_index()`	Returns a DataFrame that has row index 0, 1, etc., and adds the current index as a column.
`s.str.len()`	Returns a Series containing length of each string
`s.str.lower()` or `s.str.upper()`	Returns a Series containing lowercase/uppercase version of each string
`s.str.split(pat)`	Split strings around given separator/delimiter `pat`. If not specified, split on whitespace

Groupby

In the groupby entries below, `col` can be a column label or a list of column labels:

Function	Description
`df.groupby(col).count()`	Returns a Series/DataFrame with the counts of non-missing values in each column
`df.groupby(col).size()`	Returns a Series counting the number of rows in each group, including missing values
`df.groupby(col).mean()` or `df.groupby(col).min()` or `df.groupby(col).max()`	Returns a Series/DataFrame containing mean/min/max of each group for each column, excluding missing values
`df.groupby(col).first()` or `df.groupby(col).last()`	Returns a Series/DataFrame containing the first/last non-null entry of each group for each column
`df.groupby(col).agg(f)`	Returns a DataFrame with index `col`. Aggregates other columns using the given function f.

a. A card table

b. A Pandas Reference Sheet with some of what we have covered in class ✔

c. A duck

Your answer is correct.

What are these?

Question **2**

Correct

Mark 5.00 out of 5.00

$\sum_{n=1}^{4} n^2 = ?$

a. 1 + 2 + 3 + 4

b. $1^2 + 2^2 + 3^2 + 4^2$ ✔

c. $1^2 + 4^2$

d. $(1 + 2 + 3 + 4)^2$

Your answer is correct.

Question **3**

Correct

Mark 5.00 out of 5.00

Let $x_1, x_2, \ldots, x_n$ be a fixed list of numbers and let $\bar{x}$ be the mean of those numbers.

$\sum_{i=1}^{n} x_i = n\bar{x}$

Select one:

True ✔

False

Question **4**

Correct

Mark 5.00 out of 5.00

Which of the following are true statements about float values in Python?

Select ALL that apply for full credit.

Select one or more:

- [x] a.
  After arithmetic with float values, the final few decimal places can be wrong. ✔
- [ ] b.
  Float values don't have limited size.
- [ ] c.
  The boolean statement 0.1+ 0.1 + 0.1 == 0.3 will output True in Python
- [x] d.
  Float values can be used inside NumPy functions. ✔

Your answer is correct.

Question **5**

Correct

Mark 5.00 out of 5.00

```
weird = pd.DataFrame({1:["topdog","botdog"], "1":["topcat","botcat"]})
weird
```

	1	1
0	topdog	topcat
1	botdog	botcat

Predict the output of the following:

```
weird[1:]
```

a.

	1	1
1	botdog	botcat

b.

```
0    topcat
1    botcat
Name: 1, dtype: object
```

c.

```
0    topdog
1    botdog
Name: 1, dtype: object
```

d.

	1	1
0	topdog	topcat
1	botdog	botcat

Your answer is correct.

Question **6**

Incorrect

Mark 0.00 out of 5.00

Complete the definition of *the multiplication rule.*

P(C) ✖ P(A) = P(A) ✖ P(C)

P(A∩C) P(C|A) P(A|C) P(A∪C)

Your answer is incorrect.

You can condition on C -- $P(A \cap C) = P(A \mid C)P(C)$ -- or A -- $P(A \cap C) = P(C \mid A)P(A)$.

Question **7**

Correct

Mark 5.00 out of 5.00

Complete the definition of *conditional probability*.

$$P(A \mid C) = \frac{\text{P(C|A)} \checkmark \ * \text{P(A)}}{\text{P(C)} \checkmark}$$

P(A) P(A|C) P(A∪C) P(A∩C)

Your answer is correct.

The definition of bayes rule is $P(A \mid C) = \frac{P(C|A)*P(A)}{P(C)}$.

Question **8**

Correct

Mark 5.00 out of 5.00

To show that A and B are independent, it suffices to show (select all that apply, incorrect answers penalize you).

Select one or more:

- [ ] a. $P(B \cup A) = P(A)P(B)$
- [ ] b. $P(B \mid A) = \frac{P(A)}{P(B)}$
- [x] c. $P(A \mid B) = P(A)$ ✔
- [x] d. $P(B \mid A) = P(B)$ ✔
- [x] e. $P(B \cap A) = P(A)P(B)$ ✔

Your answer is correct.

Question **9**

Correct

Mark 4.00 out of 4.00

Answer type - Number: Please enter numerical values in decimal form.
Please format your numerical responses to include at least two decimal places.

Assume that $P(A \cup B)$ = 0.81, P(B) = 0.41 and $P(A \cap B) = 0.03$.
What is $P(A)$?

0. ✔

Correct!

Question **10**

Correct

Mark 3.00 out of 3.00

Let $x_1, x_2, \ldots, x_n$ be a fixed list of numbers and let $\bar{x}$ be the mean of those numbers.

$\sum_{i=1}^{n} (x_i - \bar{x}) = 0$

Select one:

- True ✔
- False

Question **11**

Correct

Mark 4.00 out of 4.00

The distribution of vehicle prices tends to be **right skewed**, with a few luxury and sports cars lingering out into the right tail. Assuming you care about price and are in the market **for a regular car, should you be more interested in the mean or median price of vehicles sold**?

- a. Median ✔
- b. Mean

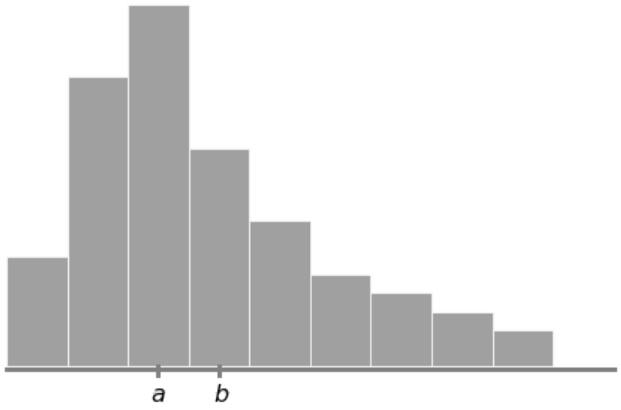
Your answer is correct.

Question **12**

Correct

Mark 5.00 out of 5.00

Which of the following statements is true about the data represented by the histogram below?



a. **a** and **b** are both technically the mean

b. **a** is the mean and **b** is the median

c. **a** is the mean and **b** is the mode

d. **b** is the mean and **a** is the median ✔

Your answer is correct.

Question **13**

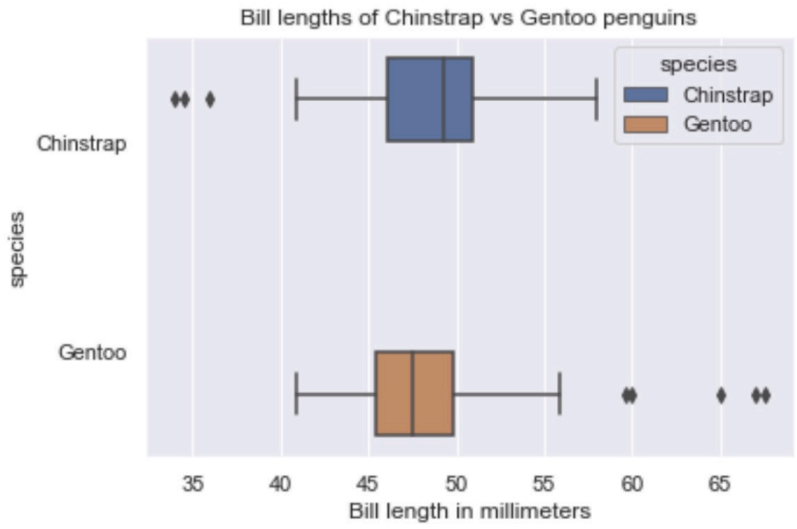
Correct

Mark 5.00 out of 5.00

Based on the boxplot, what is the 3rd quartile of the bill lengths of Gentoo penguins?

(Give your answer in mm)

For example, if answer is 3 mm, enter 3.



Answer: 50 ✔

Question **14**

Correct

Mark 5.00 out of 5.00

Suppose you flip a fair coin two times. Let A be the event "The first coin flip is a Heads". Let B be the event "The two flips match (i.e. both Heads or both Tails)". Are the events A and B independent?

- a. **Yes**, events A and B are independent. ✔
- b. **No**, events A and B are not independent.
- c. More information is needed.
- d. While on the surface the events appear independent, it is possible to show they are dependent.
- e. The events are neither independent or dependent.

Your answer is correct.

Question **15**

Correct

Mark 3.00 out of 3.00

You decide to study nutrition at the food locations on campus. You randomly sample 500 items served at the UMC in Sept 2023 and put the sampled items into a DataFrame **table** that contains one row for each item. Here are the first few rows.

item	category	priority	calories	plates
Herbed Pasta Salad.	Salad Bar	Low	150	10,421
Raspberry Sammie	Desserts	High	200	20,497
Lobster Bisque	Soups	Medium.	240	4,249

The **menu** table contains five columns:

- **item:** a string, the name of the item on the menu
- **category:** a string, the food category of the item
- **priority:** a string, the priority that Dining puts on making the item ('Low', 'Medium', or 'High')
- **calories:** an int, the number of calories per plate
- **plates:** an int, the number of plates served to students in the month

Identify the type for the **priority** variable.

a. Qualitative Nominal

b. Quantitative Discrete

c. Qualitative Ordinal ✔

Your answer is correct.

Question **16**

Correct

Mark 3.00 out of 3.00

You decide to study nutrition at the food locations on campus. You randomly sample 500 items served at the UMC in Sept 2023 and put the sampled items into a DataFrame **table** that contains one row for each item. Here are the first few rows.

item	category	priority	calories	plates
Herbed Pasta Salad.	Salad Bar	Low	150	10,421
Raspberry Sammie	Desserts	High	200	20,497
Lobster Bisque	Soups	Medium.	240	4,249

The **menu** table contains five columns:

- **item:** a string, the name of the item on the menu
- **category:** a string, the food category of the item
- **priority:** a string, the priority that Dining puts on making the item ('Low', 'Medium', or 'High')
- **calories:** an int, the number of calories per plate
- **plates:** an int, the number of plates served to students in the month

Identify the type for the **plates** variable.

a. Qualitative Ordinal

b. Qualitative Nominal

c. Quantitative ✔

Your answer is correct.

Question **17**

Incorrect

Mark 0.00 out of 3.00

You decide to study nutrition at the food locations on campus. You randomly sample 500 items served at the UMC in Sept 2023 and put the sampled items into a DataFrame **table** that contains one row for each item. Here are the first few rows.

item	category	priority	calories	plates
Herbed Pasta Salad.	Salad Bar	Low	150	10,421
Raspberry Sammie	Desserts	High	200	20,497
Lobster Bisque	Soups	Medium.	240	4,249

The **menu** table contains five columns:

- **item:** a string, the name of the item on the menu
- **category:** a string, the food category of the item
- **priority:** a string, the priority that Dining puts on making the item ('Low', 'Medium', or 'High')
- **calories:** an int, the number of calories per plate
- **plates:** an int, the number of plates served to students in the month

You decide to try to answer the following question. Choose which kind of visualization would be the **best choice out of the options given** to answer it. **Choose only one answer.**

Do dessert items tend to be of high priority or low priority?

- a. Histogram ✖
- b. Line Graph
- c. Bar Chart
- d. Violin Plot
- e. Scatter Plot

Your answer is incorrect.

Question **18**

Correct

Mark 3.00 out of 3.00

You decide to study nutrition at the food locations on campus. You randomly sample 500 items served at the UMC in Sept 2023 and put the sampled items into a DataFrame **table** that contains one row for each item. Here are the first few rows.

item	category	priority	calories	plates
Herbed Pasta Salad.	Salad Bar	Low	150	10,421
Raspberry Sammie	Desserts	High	200	20,497
Lobster Bisque	Soups	Medium.	240	4,249

The **menu** table contains five columns:

- **item:** a string, the name of the item on the menu
- **category:** a string, the food category of the item
- **priority:** a string, the priority that Dining puts on making the item ('Low', 'Medium', or 'High')
- **calories:** an int, the number of calories per plate
- **plates:** an int, the number of plates served to students in the month

You decide to try to answer the following question. Choose which kind of visualization would be the **best choice out of the options given** to answer it. **Choose only one answer.**

How do the distributions of calories per plate vary between the high priority and low priority desserts?

a. Scatter Plot

b. Side-by-side violin plots ✔

c. Line Plot

d. Bar Chart

Your answer is correct.

Question **19**

Correct

Mark 3.00 out of 3.00

There are 5 people in a room.

Assume a 365 day year (ignoring leap year)

To find the **probability P that at least 2 people share a birthday**, I use the following:

P(no shared birthdays) = 365/365 * 364/365 * 363/365 * 362/365 * 361/365

P(at least 2 people share a birthday) = 1 - P(no shared birthdays)

Is the above method to find P(at least 2 people share a birthday) correct? True or False

Select one:

- True ✔
- False

Question **20**

Complete

Mark 3.00 out of 3.00

This problem concerns conditional probabilities.

A **fair coin** is tossed three times. What is the probability that exactly two heads occur, given that:

(a) The first *two* outcomes were heads?

(b) The first *two* outcomes were tails?

Let F1, F2, F3 represent the different flips. You may use the notation *e.g.* F1=T to mean flip 1 is a tail or F2F3 = HH to mean that flips 2 & 3 result in a head.

For each problem, state the problem using conditional probabilities. You should **list the event space that is being addressed by the conditional probability and the final probability.**

(a) In short, the probability that **exactly** two heads occur given that the first two outcomes were heads is 50%. This is because in the conditional probability of

$$P(A|B_{2H}) = \frac{(A \cap B_{2H})}{P(B_{2H})}$$

$P(A \cap B_{2H}) = 1/8$ and $P(B_{2H}) = 1/4$. Using this result we find that this probability is indeed 50% by the aforementioned expression ((1 / 8) / (1/4) = 1 / 2).

(b) In short, the probability that **exactly** two heads occur given that the first two outcomes were tails is 0%. This is because in the conditional probability of

$$P(A|B_{2T}) = \frac{(A \cap B_{2T})}{P(B_{2T})}$$

$P(A \cap B_{2t}) = 0$ because it is impossible to get **exactly** two heads if the first two outcomes were tails. So this probability is zero.

(a) P(2 heads | F1F2 = HH). The subspace is HHH and HHT and there is one outcome, HHT, ergo the probability is 1/2.

(b) P(2 heads | F1F2 = TT). The subspace is TTH and TTT. The resulting probability is 0.

Comment:

Question **21**

Complete

Mark 0.00 out of 3.00

17. The `ti` and `fare` DataFrames contain data of the people aboard the Titanic when it crashed:

```
>>> ti.head()                          |  >>> fare.head()
   survived  class     sex    id       |        fare  alone    id
0         0  Third    male  1410       |  0  73.5000   True  1457
1         1  First  female  1522       |  1   9.2250   True  1645
2         1  Third  female  1864       |  2   8.6625   True  1716
3         1  First  female  1687       |  3  59.4000  False  1367
4         0  Third    male  1173       |  4  18.0000  False  1639
```

Both tables contain one row for each passenger, uniquely identified by the `id` column. Here's a description of the columns in each DataFrame:

DataFrame `ti`	**DataFrame `fare`**
`survived`: 1 if the person survived, else 0	`fare`: Price of ticket in USD
`class`: ticket class (First, Second, or Third)	`alone`: True if the person was alone at purchase.
`sex`: Sex of person (male or female)	

Fill in the blanks to compute the following statements. You may assume that the pandas module is imported as `pd`. **You may not use more lines than the ones provided.**

) The proportion of females who survived (a `float`).

```
ti.loc[_____,_____].mean()
```

Type your answer below.

We want to only include the rows that are first if they are female. Then we want to include the survived column to see if they indeed survived. And then take the mean after that

ti.loc['sex' == 'female', survived].mean()

This will only select rows where the column value for the sex is female, and then grab the survived column after to calculate the mean.

First blank: ti['sex'] == 'female'

Second blank: 'survived'

Comment:

```
ti.loc[ti['sex'] == 'female', 'survived'].mean()
```

Question **22**

Complete

Mark 2.00 out of 3.00

17. The `ti` and `fare` DataFrames contain data of the people aboard the Titanic when it crashed:

```
>>> ti.head()
   survived  class     sex    id
0         0  Third    male  1410
1         1  First  female  1522
2         1  Third  female  1864
3         1  First  female  1687
4         0  Third    male  1173
```

```
>>> fare.head()
      fare  alone    id
0  73.5000   True  1457
1   9.2250   True  1645
2   8.6625   True  1716
3  59.4000  False  1367
4  18.0000  False  1639
```

Both tables contain one row for each passenger, uniquely identified by the `id` column. Here's a description of the columns in each DataFrame:

DataFrame `ti`	**DataFrame `fare`**
`survived`: 1 if the person survived, else 0	`fare`: Price of ticket in USD
`class`: ticket class (First, Second, or Third)	`alone`: True if the person was alone at purchase.
`sex`: Sex of person (male or female)	

Fill in the blanks to compute the following statements. You may assume that the pandas module is imported as `pd`. **You may not use more lines than the ones provided.**

Write code to create a dataframe that combines a subset of the columns in the two tables, i.e. it should output a single table with columns id, survived, sex, and alone (in that order)

Type your answer below.

For this we want to use the merge function found in Pandas. But first, we need to extract the columns of the data frame for each data frame. From the 'ti' data frame, we need to extract the 'survived', 'sex', and 'id' columns, calling this ti_final:

ti_final = ti[['survived', 'sex', 'id']]

Doing the same for 'fare', but with just 'alone' and 'id', calling this fare_final:

fare_final = fare[['alone','id']]

We then combine these two to get the 'final' data frame:

final = pd.merge(ti_final, fare_final, how = 'inner', left_on = 'id', right_on = 'id')

```
pd.merge(ti, fare, on="id")
```

```
[["id", "survived","sex", "alone"]]
```

```
ti.merge(fare, on="id")
```

```
[["id", "survived","sex", "alone"]]
```

Comment:
columns id, survived, sex, and alone (in that order)

Question **23**

Complete

Mark 5.00 out of 5.00

You are tested for an uncommon disease. You know that there only is a 1% chance of getting it. Use the letter **D** for the event "**You have the disease**" and **T** for "**the test says so.**" Use **~D** for the event "**You don't have the disease,**" and **~T** for "**the test says you don't**." It is known that the test is imperfect:

P(T|D) = 0.98 and **P(~T|~D) = 0.95.**

a. Given that you test positive, what is the probability that you really have the disease?

Must show work to receive credit.

We essentially need to find the conditional probability of P(D|T) with Bayes' Theorem. We can calculate this using

$$P(D|T) = \frac{P(T|D)P(D)}{P(T)}$$

We know that the probability of catching the disease is a 1% chance (P(D)). So all we need to find to solve this problem is what P(T) is.

To answer this, we first need to figure out what P(T n D) is. We can find this with the conditional probability formula, after re-arranging we find:

$$P(T \cap D) = P(D) \cdot P(T|D) = 0.01 \cdot 0.98 = 0.0098$$

Next, we need to find P(T n D'). We know that P(~D) = 0.99 from the problem statement. It is true that P(T n ~D) + P(~T n ~D). So we first need to find P(~T n ~D). This is done by

$$P(T'|D') = P(D') \cdot P(T'|D') = 0.99 \cdot 0.95 = 0.9405.$$

We can use algebra to deduce P(T n ~D) = 0.99 - 0.9405 = 0.0495. It is also true that P(T) = P(T n D) + P(T n ~D). (This is all being calculated with a contingency table by the way). Adding these values together we find that P(T) = 0.0098 + 0.0495 = 0.0593. We can then plug this value in our Bayes' theorem formula to find:

$$P(D|T) = \frac{P(T|D) \cdot P(D)}{P(T)} = \frac{0.98 \cdot 0.01}{0.0593} \approx 0.1653$$

P(D|T) = 0.16526

Comment:

Question **24**

Complete

Mark 3.00 out of 6.00

We are trying to write code to return the table below:

Party	Year	Candidate	Popular vote	Result	%
American	1856	Millard Fillmore	873053	loss	21.554001
American Independent	1968	George Wallace	9901118	loss	13.571218
Anti-Masonic	1832	William Wirt	100715	loss	7.821583
Anti-Monopoly	1884	Benjamin Butler	134294	loss	1.335838
Citizens	1980	Barry Commoner	233052	loss	0.270182
Communist	1932	William Z. Foster	103307	loss	0.261069
Constitution	2008	Chuck Baldwin	199750	loss	0.152398
Constitutional Union	1860	John Bell	590901	loss	12.639283
Democratic	1964	Lyndon Johnson	43127041	win	61.344703

Each row shows the best result (in %) by each party.

For example: Best Democratic result ever was Johnson's 1964 win.

a) Given the following approach using elections.groupby("Party").agg(max).head(10), why does the table seem to claim that Woodrow Wilson won the presidency in 2020?

Why does the table seem to claim that Woodrow Wilson won the presidency in 2020?

```
elections.groupby("Party").agg(max).head(10)
```

Party	Year	Candidate	Popular vote	Result	%
American	1976	Thomas J. Anderson	873053	loss	21.554001
American Independent	1976	Lester Maddox	9901118	loss	13.571218
Anti-Masonic	1832	William Wirt	100715	loss	7.821583
Anti-Monopoly	1884	Benjamin Butler	134294	loss	1.335838
Citizens	1980	Barry Commoner	233052	loss	0.270182
Communist	1932	William Z. Foster	103307	loss	0.261069
Constitution	2016	Michael Peroutka	203091	loss	0.152398
Constitutional Union	1860	John Bell	590901	loss	12.639283
Democratic	2020	Woodrow Wilson	81268924	win	61.344703
Democratic-Republican	1824	John Quincy Adams	151271	win	57.210122

b) Describe how you would fix the issue. Provide an example of what code you could use.

Type answers to a and b below.

a) In this case, the aggregate function is finding the max of all values in all columns. So it is not essentially finding the max percentage of the candidate for each party in the election, it is finding the max value amongst all columns.

b) To solve this problem, one could first group the values by each party, sort the values in descending order, and then grab the first entry from each party to find the max value of each party. This would look something like this:

```
elections = elections.groupby("Party")
elections = elections.sort_values(by = "%", ascending = False)
elections = elections.groubpy("Party").first()
```

Comment:

```
AttributeError: 'DataFrameGroupBy' object has no attribute 'sort_values'
```

For b): Need to get the best result (in %) by each party

Approach would be something like:

elections_sorted_by_percent = elections.sort_values("%", ascending=False) elections_sorted_by_percent.groupby("Party").first()