Exam 1 Review

Format of the Exam

Read a dataset and be able to answer the following questions involving this dataset. See the attached documents for specific question examples of this nature.

1. Formulating a research question and collecting data (a sample)

1.1. Definitions:

1.1.1. Population vs. a sample vs. observation

1.2. Questions about populations

1.2.1. When can we use a sample from a population to answer a research question about the population?

1.2.2. What is the difference between saying "two variable in a population are **associated"** vs. "there is a **causal relationship** between two variables in a population?"

1.2.3. When can we use a sample from a population to determine if there is a **causal relationship** between two variables in the population?

2. Data management

2.1. How do we **read** a csv file into a dataframe.

2.2. What does a **row** represent in a dataframe?

2.3. What does a **column** represent in a dataframe?

3. Data cleaning

3.1. Basic Ensuring read data is of the **correct 'data type'.**

3.1.1. What are some different types of objects in Python?

3.1.2. How do we determine what "type" of data an object is in Python?

3.1.3. What happens if we try to add two string objects in Python?

3.1.4. How do we visualize all types of values for a column in a dataframe?

3.2. Basic Dealing with **Missing values**

3.2.1. How do we read a csv into a dataframe and indicate, which types of values we would like Python to convert into a "NaN" object?

3.2.2. How do you detect if there are "NaN" objects in a dataframe?

3.2.3. How do we drop all rows from a dataframe that contain "NaN" values?

3.2.4. What will most functions in Python automatically do if they encounter a dataframe that has rows with missing values?

4. Dataframe Manipulation

4.1. **Imports**

4.1.1. What Python package have we used in this class that contains all the dataframe manipulation functions?

4.2. Basic Dataframe **Description**

4.2.1. How to show the first k rows of a dataframe.

4.2.2. How to get the shape of a dataframe.

4.2.3. How to get the columns of a dataframe.

4.2.4. How to get the index of a dataframe.

4.3. **Creating** a dataframe

4.3.1. How do you create a dataframe from scratch in Python?

4.4. How to **isolate a column** from a dataframe.

4.4.1. How do you isolate a column from a dataframe using: 1.) brackets and 2.) the name of the column?

4.4.2. How do you isolate a column from a dataframe using : the iloc function?

4.5. How to **isolate ranges of values** in a dataframe?

4.5.1. How do you isolate a row in the dataframe?

4.5.2. How do you isolate a range of rows in a dataframe?

4.5.3. How do you isolate a set of rows in a dataframe?

4.5.4. How do you isolate a column in the dataframe?

4.5.5. How do you isolate a range of column in a dataframe?

4.5.6. How do you isolate a set of columns in a dataframe?

4.5.7. How do you isolate a range/set of rows and columns in a dataframe?

4.6. **Filtering** a dataframe

4.6.1. How do you set up a condition for a column in a dataframe?

4.6.2. How do you a filter a dataframe for a certain column condition in Python?

4.6.3. When do you use '==' vs '=' in Python?

4.7. What are two ways to **vertically stack** two dataframes?

4.7.1. How do you do this so the index of the resulting dataframe goes from 0 to n-1 (where n is the length of the dataframe)?

4.8. How do you **merge** two dataframes that share similar values in a given column?

4.9. How do you **sort** a dataframe by values in a given column?

4.9.1. In ascending order?

4.9.2. In descending order?

5. Descriptive analytics

5.1. **What type of analysis?**

5.1.1. What the difference between a numerical variable and a categorical variable?

5.2. **Imports**

5.2.1. What three Python packages that have we used in this class that contains functions that allow us to calculate **summary statistics** and create **visualizations** for dataframes?

5.3. **Describing** the distribution of a **single categorical variable** *in a dataset (sample).*

5.3.1. Summary Statistics

5.3.1.1. How do you **count** the **number** of each level of a categorical variable in Python?

5.3.2. Visualizations

5.3.2.1. How do you **visualize** the **number** of each level of a categorical variable in Python?

5.4. Determining if there is an **association** between **two categorical variables** *in a dataset (sample).*

5.4.1. Summary Statistics

5.4.1.1. How do you **count** the **number** of each combination of levels of two categorical variables in Python?

5.4.1.2. How do you **count** the **proportion** of each level of a categorical variable **for a given level of *another* categorical variable?**

5.4.2. Visualizations

5.4.2.1. How do you **visualize** the **proportion** of each level of a categorical variable **for a given level of *another* categorical variable?**

5.4.3. Putting it all together

5.4.3.1. How do you use these summary statistics and/or visualization to determine if there is an association between the two categorical variables?

5.5. **Describing** the distribution of a **single numerical variable** *in a dataset (sample).*

5.5.1. Summary Statistics

5.5.1.1. Measures of Center:

5.5.1.1.1. How do you calculate the mean of a numerical variable in Python?

5.5.1.1.2. How do you calculate the median of a numerical variable in Python?

5.5.1.1.3. Where will the mean and the median be located in a:

5.5.1.1.3.1. Symmetric distribution?

5.5.1.1.3.2. Right-skewed distribution?

5.5.1.1.3.3. Left-skewed distribution?

5.5.1.2. **Measures of Spread:**

5.5.1.2.1. How do you calculate the standard deviation of a numerical variable in Python and by hand?

5.5.1.2.2. How do you calculate the variance of a numerical variable in Python and by hand?

5.5.1.2.3. How do you calculate the Q1 of a numerical variable in Python?

5.5.1.2.4. How do you calculate the Q3 of a numerical variable in Python?

5.5.1.2.5. How do you calculate the IQR of a numerical variable in Python and by hand?

5.5.1.2.6. How do you calculate the range of a numerical variable in Python and by hand?

5.5.2. Visualizations

5.5.2.1. Histograms

5.5.2.1.1. How do you **create a histogram** in Python for a single numerical variable?

5.5.2.1.1.1. With "counts" on the y-axis?

5.5.2.1.1.2. With "frequency" on the y-axis?

5.5.2.1.1.3. With a "density curve" overlaid?

5.5.2.1.2. How do you **use a histogram** to:

5.5.2.1.2.1. Determine **shape** of the distribution

5.5.2.1.2.2. Determine **skew** of the distribution.

5.5.2.1.2.3. Calculate the proportion of observations in a dataset that are within a certain range?

5.5.2.2. **Boxplots**

5.5.2.2.1. How do you **create a boxplot** in Python (and by hand) for a single numerical variable?

5.5.2.2.2. How do you **use a boxplot** to determine the Q1, Q3, IQR, median, outliers, and range of a single numerical variable?

5.5.2.3. **Violin Plots**

5.5.2.3.1. How to create a violin plot in Python for a single numerical variable?

5.6. Determining if there is an **association** between a **categorical variable** and **numerical variable** *in a dataset (sample).*

5.6.1. How do you **create side-by-side boxplots** in Python (and by hand) for a numerical variable and categorical variable?

5.6.2. How do you **create side-by-side violin plots** in Python (and by hand) for a numerical variable and categorical variable?

5.6.3. How do you use these plots to determine if there is an **association** between the numerical and categorical variable?

6. Probability Theory

6.1. Calculate probabilities using the uniform distribution rule, combinatorics, and permutation equations.

6.1.1. How do you collect a sample of size n from a dataframe in Python with/without replacement?

6.1.2. Drawing k observations from a population **with replacement** sampling (where **order/sequence/assignment matters)**

6.1.2.1. Generate the sample space.

6.1.2.2. Use combinatorics equations to calculate the number of simple events in the sample space.

6.1.3. Drawing k observations from a population **without replacement** sampling (where **order/sequence/assignment matters)**

6.1.3.1. Generate the sample space.

6.1.3.2. Use combinatorics equations to calculate the number of simple events in the sample space.

6.1.4. Drawing k observations a population **without replacement** sampling (where **order/sequence/assignment DOESN'T matter)**

6.1.4.1. Generate the sample space.

6.1.4.2. Use combinatorics equations to calculate the number of simple events in the sample space.

6.2. For a discrete random variable X, *that is NOT of a specific family of random variables.*

6.2.1. Calculate the mean (ie. E[X]) of X.

6.2.1.1. Given a table of probabilities.

6.2.2. Calculate the variance (ie. Var[X]) of X.

6.2.2.1. Given a table of probabilities.

6.2.3. Calculate the standard deviation (ie. SD[X]) of X.

6.2.3.1. Given a table of probabilities.

6.2.4. Calculate the P(X=#) given a probability mass function.

6.3. For any random variable X.

6.3.1. What does it mean to calculate a summary statistic of a random variable?

6.4. For a Bernoulli random variable X.

6.4.1. Determine when a random variable X is a *Bernoulli random variable.* What parameter values do we need to know to calculate probabilities for a Bernoulli random variable?

6.4.2. Calculate the mean (ie. E[X]) of X.

6.4.3. Calculate the variance (ie. Var[X]) of X.

6.4.4. Calculate the standard deviation (ie. SD[X]) of X.

6.4.5. Calculate the P(X=#).

6.5. For a normal random variable X.

6.5.1. What parameter values do we need to know to calculate probabilities for a normal random variable?

6.5.2. Calculate the mean (ie. E[X]) of X.

6.5.2.1. Using Python.

6.5.3. Calculate the variance (ie. Var[X]) of X.

6.5.3.1. Using Python.

6.5.4. Calculate the standard deviation (ie. SD[X]) of X.

6.5.4.1. Using Python.

6.5.5. For a given #, calculate P(X≤#), P(X>=#), P(a<=X<=b).

6.5.5.1. Using Python.

6.6. For a geometric random variable

6.6.1. Determine when a random variable X is a *Geometric random variable.* What parameter values do we need to know to calculate probabilities for a Bernoulli random variable?

6.6.2. Calculate the mean (ie. E[X]) of X in Python.

6.6.3. Calculate the variance (ie. Var[X]) of X in Python.

6.6.4. Calculate the standard deviation (ie. SD[X]) of X in Python.

6.6.5. Calculate the P(X=#).

6.7. Calculate probabilities using the rules for combining probabilities

6.7.1. For events A and B

6.7.1.1. Determine if A and B are mutually exclusive or not.

6.7.1.2. Determine if A and B are independent or dependent.

6.7.2. Given P(A) and P(B).

6.7.2.1. How to calculate P(A or B).

6.7.2.2. How to calculate P(A and B).

6.7.2.3. How to calculate P(A|B).

7. Inference basics

7.1. Create a Sampling Distribution of Sample Means

7.1.1. How to create one.

7.1.2. Calculate the mean, standard deviation, and find the shape of this sampling distribution.

7.2. Create a Sampling Distribution of Sample Proportions

7.2.1. How to create one.

7.2.2. Calculate the mean, standard deviation, and find the shape of this sampling distribution.

8. Additional Coding

8.1. For loops

8.1.1. How to iterate a for-loop over a **range()** of values in Python.

8.1.2. How to use a for-loop over a **list** of values in Python.

8.1.3. How to use the **append()** function in Python.

8.1.4. How to use the '+=' operator in Python.

8.2. while loops

8.2.1. How to use a while loop.

8.3. Functions

8.3.1. How to define a function in Python.

8.3.2. How do use a function you defined in Python.

8.4. If/then operators

8.4.1. How to write an if/then statement in Python.