INT 303 BIG DATA ANALYTICS

Lecture 2 Data

Jia WANG

Jia.wang02@xjtlu.edu.cn

Xi'an Jiaotong-Liverpool University

西交利物浦大學

LECTURE OUTLINE

What are Data

Data Exploration

Data Cleaning

Data Descriptive Statistics

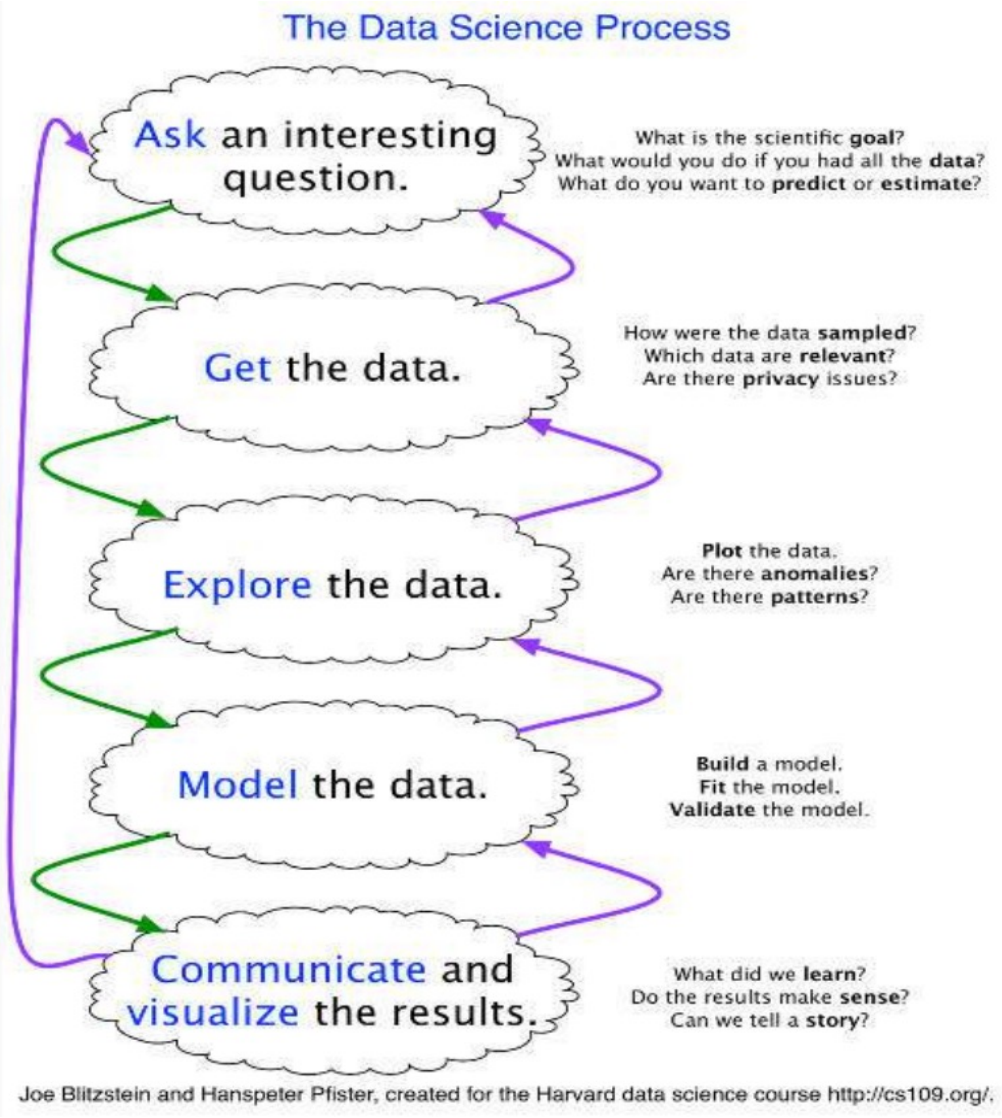
THE DATA SCIENCE PROCESS

Recall the data science process.

- Ask questions
- Data Collection
- Data Exploration
- Data Modeling
- Data Analysis
- Visualization and Presentation of Results

Today we will begin introducing data exploration steps.

THE DATA SCIENCE PROCESS (CONT.)

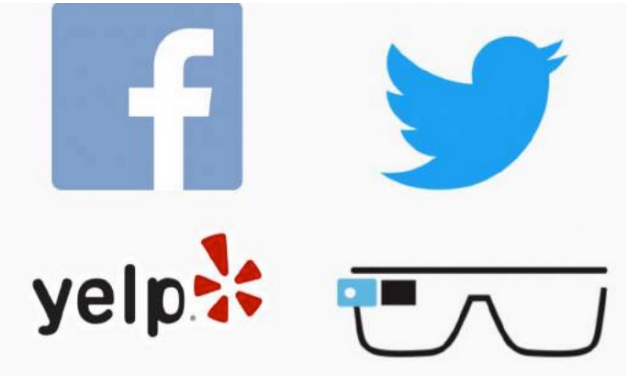


What are Data?

WHAT ARE DATA?

"A datum (a piece of information) is a single measurement of something on a scale that is understandable to both the recorder and the reader. Data are multiple such measurements."

Claim: everything is (can be) data!



WHERE DO DATA COME FROM?

- **Internal sources:** already collected by or is part of the overall data collection of you organization.
  For example: business-centric data that is available in the organization data base to record day to day operations; scientific or experimental data
- **Existing External Sources:** available in ready to read format from an outside source for free or for a fee.
  For example: public government databases, stock market data, Yelp reviews, [your favorite sport]-reference
- **External Sources Requiring Collection Efforts**: available from external source but acquisition requires special processing.
  For example: data appearing only in print form, or data on websites

WEB SCRAPING

- Why do it? Older government or smaller news sites might not have APIs for accessing data, or publish RSS feeds or have databases for download. Or, you don't want to pay to use the API or the database.
- How do you do it?
- Should you do it?
  - You just want to explore: Are you violating their terms of service? Privacy concerns for website and their clients?
  - You want to publish your analysis or product: Do they have an API or fee that you are bypassing? Are they willing to share this data? Are you violating their terms of service? Are there privacy concerns?

WAYS TO GATHER ONLINE DATA

How to get data generated, published or hosted online:

- **API (Application Programming Interface):** using a prebuilt set of functions developed by a company to access their services. Often pay to use. For example: Google Map API, Facebook API, Twitter API
- **RSS (Rich Site Summary):** summarizes frequently updated online content in standard format. Free to read if the site has one. For example: news-related sites, blogs
- **Web scraping:** using software, scripts or by-hand extracting data from what is displayed on a page or what is contained in the HTML file.

DATA STORAGE

How is your data represented and stored (data format)?

- **Tabular Data:** a dataset that is a two-dimensional table, where each row typically represents a single data record, and each column represents one type of measurement (csv, dat, xlsx, etc.).
- **Structured Data:** each data record is presented in a form of a [possibly complex and multi-tiered] dictionary (json, xml, etc.)
- **Semistructured Data:** not all records are represented by the same set of keys or some data records are not represented using the key-value pair structure.

DATA TYPES

What kind of values are in your data (data types)?

Simple or atomic types:

- **Numeric:** integers, floats
- **Boolean:** binary or true false values
- **Strings:** sequence of symbols

Compound, composed of a bunch of atomic types:

- **Date and time:** compound value with a specific structure
- **Lists:** a list is a sequence of values
- **Dictionaries:** A dictionary is a collection of key-value pairs, a pair of values *x* : *y* where *x* is usually a string called the key representing the "name" of the entry, and *y* is a value of any type.

QUESTION

Example: Student record: what are the types of these features?

- First: Kevin
- Last: Rader
- Classes: [CS-109A, STAT139]
- Record: Pass

DATA FORMAT

How is your data represented and stored (data format)?

- Textual Data
- Temporal Data
- Geolocation Data

TABULAR DATA

In tabular data, we expect each record or observation to represent a set of measurements of a single object or event. We've seen this already in Lecture 0:

First Look At The Data

```
In [27]: hubway_data = pd.read_csv('hubway_trips.csv', low_memory=False)
         hubway_data.head()
```

Out[27]:

	seq_id	hubway_id	status	duration	start_date	strt_statn	end_date	end_statn	bike_nr	subsc_type	zip_code	birth_d
0	1	8	Closed	9	7/28/2011 10:12:00	23.0	7/28/2011 10:12:00	23.0	B00468	Registered	'97217	1976.0
1	2	9	Closed	220	7/28/2011 10:21:00	23.0	7/28/2011 10:25:00	23.0	B00554	Registered	'02215	1966.0
2	3	10	Closed	56	7/28/2011 10:33:00	23.0	7/28/2011 10:34:00	23.0	B00456	Registered	'02108	1943.0
3	4	11	Closed	64	7/28/2011 10:35:00	23.0	7/28/2011 10:36:00	23.0	B00554	Registered	'02116	1981.0
4	5	12	Closed	12	7/28/2011 10:37:00	23.0	7/28/2011 10:37:00	23.0	B00554	Registered	'97214	1983.0

Each type of measurement is called a **variable** or an **attribute (features)** of the data (e.g. `seq_id`, `status` and `duration` are variables or attributes). The number of attributes is called the **dimension**.

We expect each table to contain a set of **Samples** (**records** or **observations)** of the same kind of object or event (e.g. our table above contains observations of rides/checkouts).

TABULAR ☺

Common causes of messiness are:

- Column headers are values, not variable names
- Variables are stored in both rows and columns
- Multiple variables are stored in one column/entry
- Multiple types of experimental units stored in same table

In general, we want each file to correspond to a dataset, each column to represent a single variable and each row to represent a single observation.

We want to **tabularize** the data. This makes Python happy.

EXAMPLE

```
In [8]: dfcand=pd.read_csv("../data/candidates.txt", sep='|')
        dfcand.head(10)
```

Out[8]:

	id	first_name	last_name	middle_name	party
0	33	Joseph	Biden	NaN	D
1	36	Samuel	Brownback	NaN	R
2	34	Hillary	Clinton	R.	D
3	39	Christopher	Dodd	J.	D
4	26	John	Edwards	NaN	D
5	22	Rudolph	Giuliani	NaN	R
6	24	Mike	Gravel	NaN	D
7	16	Mike	Huckabee	NaN	R
8	30	Duncan	Hunter	NaN	R
9	31	Dennis	Kucinich	NaN	D

WHY DO DATA TYPES MATTER A LOT?

We'll see later that it's important to distinguish between classes of variables or attributes based on the type of values they can take on.

- **Quantitative variable:** is numerical and can be either:
  - **discrete -** a finite number of values are possible in any bounded interval. For example: "Number of siblings" is a discrete variable
  - **continuous -** an infinite number of values are possible in any bounded interval. For example: "Height" is a continuous variable
- **Categorical variable:** no inherent order among the values For example: "What kind of pet you have" is a categorical variable

QUANTITATIVE VARIABLE (1)

In [21]:
```
bins = np.linspace(-3, 3, 11)
print("bins: {}".format(bins))
```

```
bins: [-3.  -2.4 -1.8 -1.2 -0.6  0.   0.6  1.2  1.8  2.4  3. ]
```

In [22]:
```
which_bin = np.digitize(X, bins=bins)
print("\nData points:\n", X[:5])
print("\nBin membership for data points:\n", which_bin[:5])
```

```
Data points:
 [[-0.753]
 [ 2.704]
 [ 1.392]
 [ 0.592]
 [-2.064]]

Bin membership for data points:
 [[ 4]
 [10]
 [ 8]
 [ 6]
 [ 2]]
```

QUANTITATIVE VARIABLE (2)

In [23]:
```
from sklearn.preprocessing import OneHotEncoder
# transform using the OneHotEncoder
encoder = OneHotEncoder(sparse=False)
# encoder.fit finds the unique values that appear in which_bin
encoder.fit(which_bin)
# transform creates the one-hot encoding
X_binned = encoder.transform(which_bin)
print(X_binned[:5])
```

```
[[0. 0. 0. 1. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 1.]
 [0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 0. 0. 0. 1. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]]
```

CATEGORICAL VARIABLE

In [8]:
```
# create a DataFrame with an integer feature and a categorical string feature
demo_df = pd.DataFrame({'Integer Feature': [0, 1, 2, 1],
                        'Categorical Feature': ['socks', 'fox', 'socks', 'box']})
display(demo_df)
```

	Categorical Feature	Integer Feature
0	socks	0
1	fox	1
2	socks	2
3	box	1

使用get_dummies 只会编码字符串特征，不会改变整数特征

In [9]:
```
display(pd.get_dummies(demo_df))
```

	Integer Feature	Categorical Feature_box	Categorical Feature_fox	Categorical Feature_socks
0	0	0	0	1
1	1	0	1	0
2	2	0	0	1
3	1	1	0	0

Data Cleaning

COMMON ISSUES

Common issues with data:

- Missing values: how do we fill in?
- Messy data format: how do we handle it?
- Not usable: the data cannot answer the question posed

MISSING DATA

Missing data could result from a human factor (for example, a person deliberately failing to respond to a survey question), a problem in electrical sensors, or other factors. And when this happens, you can lose significant information.

Among the categories are:

- Missing Completely at Random (MCAR).
- Missing at Random (MAR).
- Not Missing at Random (NMAR).

DATA THAT'S MISSING COMPLETELY AT RANDOM (MCAR)

- These are data that are missing completely at random. The missingness is independent from the data.
- There is no discernible pattern to this type of data missingness.
- This means that you cannot predict whether the value was missing due to specific circumstances or not.

DATA THAT'S MISSING AT RANDOM (MAR)

- These types of data are missing at random but not completely missing.

- The data's missingness is determined by the data you see.

MISSING DATA THAT'S NOT MISSING AT RANDOM (NMAR)

- These are data that are not missing at random and are also known as ignorable data.
- In other words, the missingness of the missing data is determined by the variable of interest.

How Should You Handle Missing Data?

- As we just learned, these techniques cannot be that precise in determining the missing value. They appear to have some biases.

- Handling missing values falls generally into two categories. We will look at the most common in each category.

- The two categories are as follows:
  - ✓ Deletion
  - ✓ Imputation

HANDLE MISSING DATA WITH DELETION

What is List-Wise Deletion?

- Remove a record or observation in the dataset if it contains some missing values.
- Perform list-wise deletion on any of the aforementioned missing value categories.
- Disadvantages is potential information loss.
- The general rule: **when the number of observations with missing values exceeds the number of observations without missing values.**

```
df.dropna(axis=1, inplace=True)
```

HOW TO HANDLE MISSING DATA WITH IMPUTATION?

- Another frequent general method for dealing with missing data is to fill in the missing value with a substituted value.

- This methodology encompasses various methods, but we will focus on the most prevalent ones here.

    - Prior knowledge of an ideal number
    - Regression imputation
    - Simple Imputation
    - KNN Imputation

FILL IN THE MISSING VALUE WITH PRIOR KNOWLEDGE

This method entails replacing the missing value with a specific value.
To use it, you need to have domain knowledge of the dataset.

You use this to populate the MAR and MCAR values.
To implement it in Python, you use the .fillna method in Pandas like this:

```
df.fillna(inplace=True)
```

SIMPLE IMPUTATION

This method involves utilizing a numerical summary of the variable where the missing value occurred (that is using the feature, such as mean, median, and mode).

When you use this strategy to fill in the missing values, you need to evaluate the variable's distribution to determine which central tendency summary to apply.

You use this method in the MCAR category. And you implement it in Python using the SimpleImputer transformer in the Scikit-learn library.

```
from sklearn.impute import SimpleImputer
#Specify the strategy to be the median class
fea_transformer = SimpleImputer(strategy="median")
values = fea_transformer.fit_transform(df[["Distance"]])
pd.DataFrame(values)
```

KNN IMPUTATION

KNN imputation is a fairer approach to the Simple Imputation method. It operates by replacing missing data with the average mean of the neighbors nearest to it.

You can use KNN imputation for the MCAR or MAR categories. And to implement it in Python you use the KNN imputation transformer in ScikitLearn, as seen below:

```
from sklearn.impute import KNNImputer
# I specify the nearest neighbor to be 3
fea_transformer = KNNImputer(n_neighbors=3)
values = fea_transformer.fit_transform(df[["Distance"]])
pd.DataFrame(values)
```

HOW TO USE LEARNING ALGORITHMS

- Some learning algorithms allow us to fit the dataset with missing values.
- The dataset algorithm then searches for patterns in the dataset and uses them to fill in the missing values.
- Such algorithms include XGboost, Gradient Boosting, and others.

But further discussion is out of the scope of Today…

MESSY DATA

We're measuring individual deliveries; the variables are Time, Day, Number of Produce.

	Friday	Saturday	Sunday
Morning	15	158	10
Afternoon	2	90	20
Evening	55	12	45

Problem: each column header represents a single value rather than a variable. Row headers are "hiding" the Day variable. The values of the variable, "Number of Produce", is not recorded in a single column.

FIXING MESSY DATA

We need to reorganize the information to make explicit the event we're observing and the variables associated to this event.

ID	Time	Day	Number
1	Morning	Friday	15
2	Morning	Saturday	158
3	Morning	Sunday	10
4	Afternoon	Friday	2
5	Afternoon	Saturday	9
6	Afternoon	Sunday	20
7	Evening	Friday	55
8	Evening	Saturday	12
9	Evening	Sunday	45

MORE MESSINESS

What object or event are we measuring?

What are the variables in this dataset?

How do we fix?

Delivery	Amount
On Sunday	
10:30	43
12:30	12
12:35	30
On Monday	
11:30	29
11:57	87
11.59	63
On Tuesday	
11:33	19
11:15	27
12.59	54

MORE MESSINESS

We’re measuring individual deliveries; the variables are Time, Day, Number of Produce:

Days	times	Amount
Sunday	10:30	43
Sunday	12:30	12
Sunday	12:35	30
Monday	11:30	29
Monday	11:57	87
Monday	11.59	63
Tuesday	11:33	19
Tuesday	11:15	27
Tuesday	12:59	54

FIXING MESSY DATA

We need to reorganize the information to make explicit the event we're observing and the variables associated to this event.

ID	Time	Day	Number
1	Morning	Friday	15
2	Morning	Saturday	158
3	Morning	Sunday	10
4	Afternoon	Friday	2
5	Afternoon	Saturday	9
6	Afternoon	Sunday	20
7	Evening	Friday	55
8	Evening	Saturday	12
9	Evening	Sunday	45

Data Descriptive Statistics

BASICS OF SAMPLING

Population versus sample:

- A **population** is the entire set of objects or events under study. Population can be hypothetical "all students" or all students in this class.
- A **sample** is a "representative" subset of the objects or events under study. Needed because it's impossible or intractable to obtain or compute with population data.

Biases in samples:

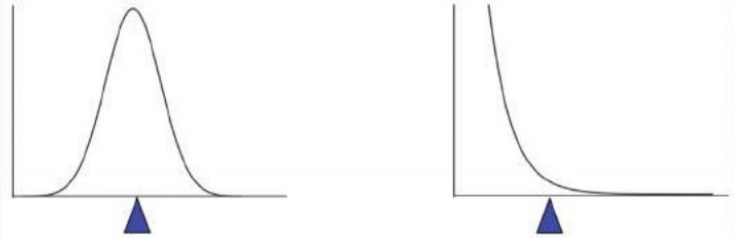
- **Selection bias**: some subjects or records are more likely to be selected
- Volunteer/**nonresponse bias**: subjects or records who are not easily available are not represented

Examples?

SAMPLE MEAN

The **mean** of a set of *n* observations of a variable is denoted $\bar{x}$ and is defined as:

$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_n}{n} = \frac{1}{n}\sum_{i=1}^{n} x_i$$



The mean describes what a "typical" sample value looks like, or where is the "center" of the distribution of the data.

Key theme: there is always uncertainty involved when calculating a sample mean to estimate a population mean.

SAMPLE MEDIAN

The **median** of a set of n number of observations in a sample, ordered by value, of a variable is is defined by

$$\text{Median} = \begin{cases} x_{(n+1)/2} & \text{if } n \text{ is odd} \\ \frac{x_{n/2} + x_{(n+1)/2}}{2} & \text{if } n \text{ is even} \end{cases}$$

Example (already in order):

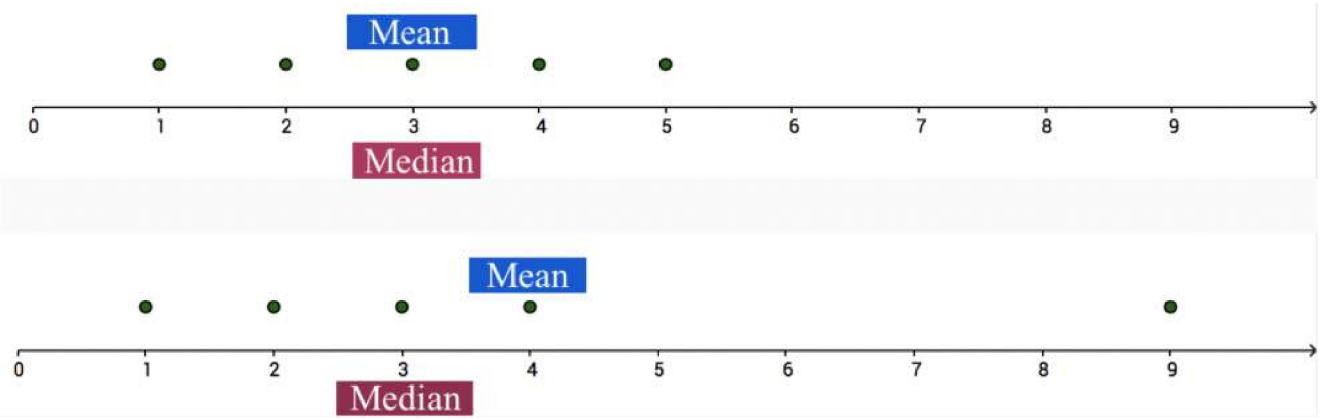
Ages: 17, 19, 21, 22, 23, 23, 23, 38

Median = (22+23)/2 = 22.5

The median also describes what a typical observation looks like, or where is the center of the distribution of the sample of observations.

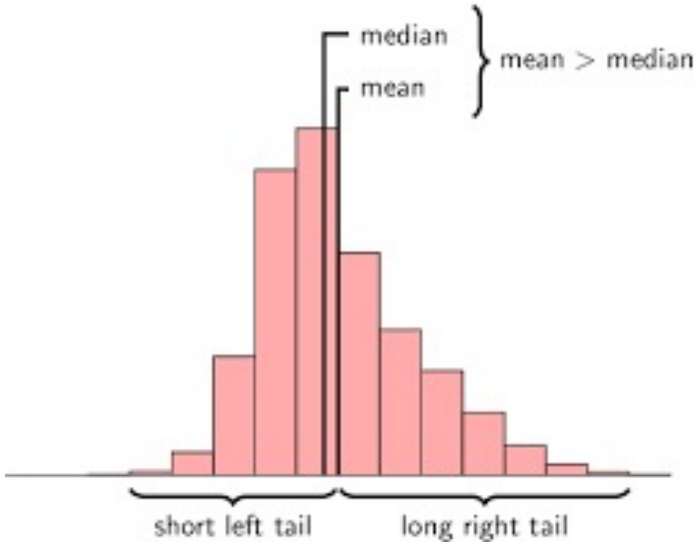
MEAN VS. MEDIAN

The mean is sensitive to extreme values (**outliers**)



MEAN, MEDIAN, AND SKEWNESS

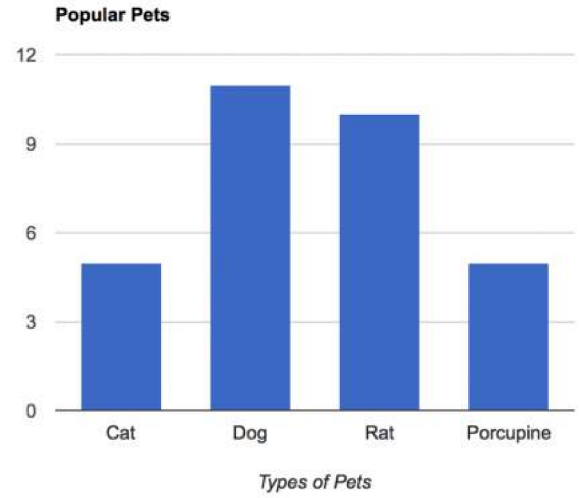
The mean is sensitive to outliers\.



The above distribution is called **right-skewed** since the mean is greater than the median. Note: **skewness** often "follows the longer tail".

REGARDING CATEGORICAL VARIABLES...

For categorical variables, neither mean or median make sense. Why?



The mode might be a better way to find the most “representative” value

MEASURES OF SPREAD: RANGE

The spread of a sample of observations measures how well the mean or median describes the sample.

One way to measure spread of a sample of observations is via the **range**.

Range = Maximum Value - Minimum Value

MEASURES OF SPREAD: VARIANCE

The (sample) **variance**, denoted s^2, measures how much on average the sample values deviate from the mean:

$$s^2 = \frac{1}{n-1}\sum_{i=1}^{n} |x_i - \bar{x}|^2$$

Note: the term $|x_i - \bar{x}|$ measures the amount by which each x_i deviates from the mean $\bar{x}$. Squaring these deviations means that s^2 is sensitive to extreme values (outliers).

Note: s^2 doesn't have the same units as the x_i :(

What does a variance of 1,008 mean? Or 0.0001?

MEASURES OF SPREAD: STANDARD DEVIATION

The (sample) **standard deviation**, denoted s, is the square root of the variance

$$s = \sqrt{s^2} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n} |x_i - \bar{x}|^2}$$

THANK YOU

VISIT US

WWW.XJTLU.EDU.CN

FOLLOW US

@XJTLU

Xi'an Jiaotong-Liverpool University
西交利物浦大學