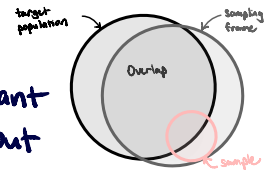
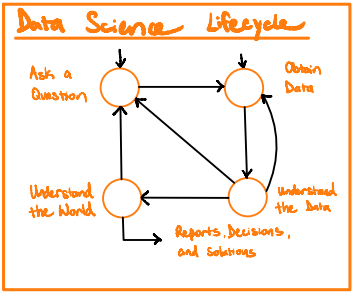
Data 100 Midterm 1 Study Guide — Jeffrey Shen

1 Sampling





Population: The group that you want to learn something about

Sampling Frame: List from which the sample is drawn

Sample: Who you actually end up sampling

Biases

Selection Bias: Systematically excluding particular groups
- ↳ **Avoid by**: sampling frame and method of sampling

Response Bias: People don't always respond truthfully
- ↳ **Avoid by**: nature of questions & method of surveying

Non-Response Bias: People don't always respond, people who don't aren't like ppl who do
- ↳ **Avoid by**: Keep surveys short, be persistant

Samples

Simple Random Sample (SRS): Sample drawn uniformly at random w/o replacement
Every individual, pair, triple... has same chance of being selected

---

2 Random Variables & Distributions

Random Variables: Variable that can take numerical values w/ particular probabilities

Expectation: weighted avg of values where weights are probabilities

$$\mathbb{E}(X) = \sum_{x \in X} x \, \mathbb{P}(X = x)$$

Linearity of Expectation: Linear transformations apply to expectations

$$\mathbb{E}(aX + bY) = a\mathbb{E}(X) + b\mathbb{E}(Y)$$

Distributions

Expectation

Bernoulli(p): takes on value 1 w/ probability p & 0 w/ prob $1-p$ — $E(X) = p$

Binomial(n, p): number of 1s in n independent Bernoulli(p) trials — $E(X) = np$

Binomial:

$$P(k \text{ successes}) = \binom{n}{k} p^k (1-p)^{n-k}$$

Multinomial n times, proportion p_i:

$$P(k_1, k_2 k_3) = \frac{n!}{k_1! k_2! k_3!} p_1^{k_1} p_2^{k_2} p_3^{k_3}$$

Uniform: probability of each value is $\frac{1}{\text{size of set}}$

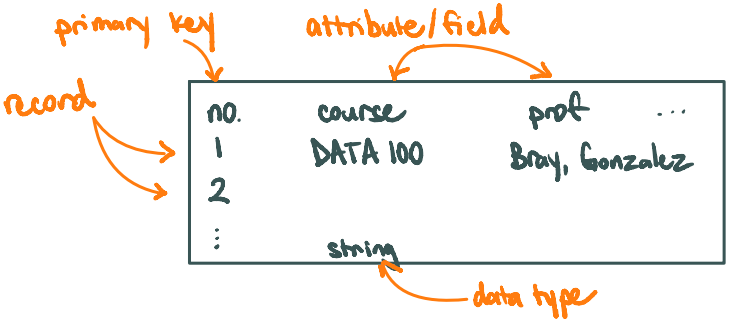
---

③ Databases & SQL

Database Management System (DBMS) vs CSV...

Pros: Reliable storage, optimized, performance
logically organized, safe concurrent operations

Database Terminology

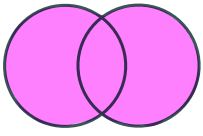


SQL

```
SELECT [DISTINT] <column expression list> FROM <list of tables>
[WHERE <predicate>]
[GROUP BY <column list>]
[HAVING <predicate>]
[ORDER BY <column list>]
[LIMIT <number of rows>]
```

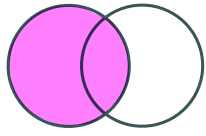
Joins

Outer Joins



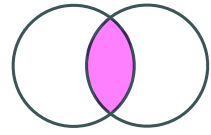
Every row from both

Left Joins



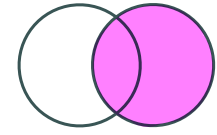
Every row from first

Inner Join



Every row both matching

Right Joins



Every row from right

Ex) SELECT * FROM s JOIN t ON s.id = t.id

---

④ Pandas

Data Structures

Index: row labels

0
1
2
⋮

Series: 1D data

0	Obama
1	McCain
⋮	

Name: Candidate, dtype: obj

Data Frame: 2D tabular data

	Candidate	Party	...
0	Obama	D	
1	McCain	R	
⋮			

[ ] operator

<dataframe>[<column name>] ⟶ Series (column selection)

<dataframe>[< list >] ⟶ Data Frame (column selection)

<dataframe>[<numeric slice>] ⟶ Data Frame (row selection)

Boolean Array Selection

Boolean mask with boolean array for each index in df

<dataframe>[<dataframe>[<column>] <condition>] ⟶ filtered Dataframe

loc / iloc

> Note: loc/iloc is inclusive on both ends, [start : stop]

loc: label based accessing

<dataframe>[< row indexes >, <column labels >]

iloc: index based accessing

<dataframe>[< row indexes >, <column indexes >]

Other helpful functions

function	description	return type
<dataframe>.index	range of indexes	Index
<dataframe>.columns	names of cols besides key	Index
<dataframe>[<col>].value_counts()	counts of each value	Series
<dataframe>.sort_values(<col>)	sort by <col>	DF
<dataframe>[<col>].unique()	list of unique items	np array

Groupby / Aggregate

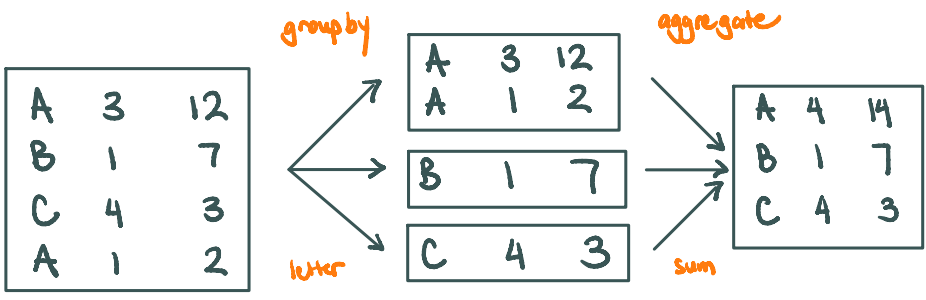
groupby: separates data by given column

<dataframe>.groupby(<col>)

aggregate: condenses every sub dataframe back into single row

<groupby>.agg(<function>)

<groupby>.sum()
<groupby>.max()
<groupby>.median()



Pivot Table

<dataframe>.pivot_table(index=<col>, columns=<col>, values=<col>, aggfunc=<f>)

- index: rows of new table
- columns: column values
- values: fields to process in each group
- aggfunc: group operation

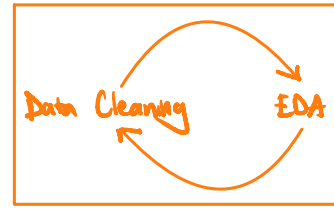
Pandas Joins

<dataframe>.merge(<df2>, how=<str>, left_on=<df_col>, right_on=<df2_col>)

- how: join type : 'inner' / 'outer' / 'left' / 'right'

---

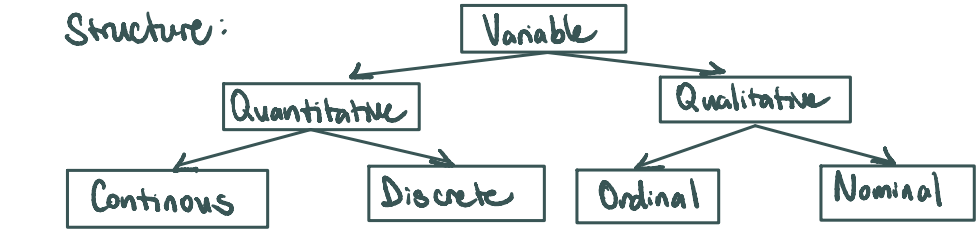
⑤ Exploratory Data Analysis (EDA)



Key Properties in EDA

1. Structure — "shape" of a data file (CSV, JSON)
2. Granularity — how fine/coarse is data (individual vs. group)
3. Scope — how (in)complete is data (cover area of interest, too expensive)
4. Temporality — how is data situated in time (data changes over time)
5. Faithfulness — how well does data capture reality (unrealistic, missing)

Structure:



---

⑥ Regular Expressions (RegEx) / Strings

Canonicalization

```
<str>.replace(<old_str>, <new_str>)
<str>.lower()
```

RegEx

Python `re` library for regex

`re.findall(<regex pattern>, <text>)`	return list of all instances
`re.sub(<regex pattern>, <text>`	substitutes pattern
`r"<regex>"`	reduces backslashes

Regex Reference

^	^ark	match beginning of string
$	ark$	match end of string
?	joh?n	match 0 or 1 times
+	jo+hn	match at least 1 time
*	AB*A	match 0 or more times
.	.u.u.	match any character
|	AA|BA	match left or right pattern
[ ]	[a-z]	match any of chars inside

( )	(AB)*A	group patterns
\b		boundary between words
\w		"word" (letters, digits, underscore)
\s		whitespace
\d		digits
[^]	[^a-z]	character class negation
\	cow\.com	escape character

---

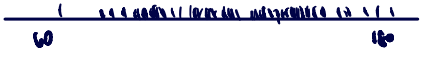
⑦ Visualizations

Accuracy of our judgements depend on type of marking

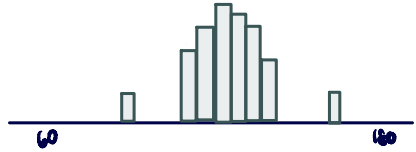
Generic ← color hue, color intensity, angle, volume, area, slope, length → Accurate

Graphs

Rug Plot: Shows every quantitative variable



Histograms: Smoothed rug plot, areas are proportions, x axis divided into bins

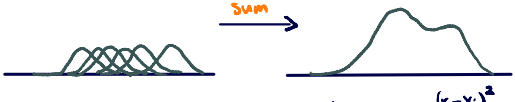


Bin width = $2\frac{IQR(x)}{\sqrt[3]{n}}$

Kernels: valid density function

1. Place kernels at each data point
2. Normalize kernels so area = 1
3. Sum all kernels

α: hyperparameter ↑ α ↑ smooth $f_\alpha(x) = \frac{1}{n}\sum_{i=1}^{n} K_\alpha(x, x_i)$

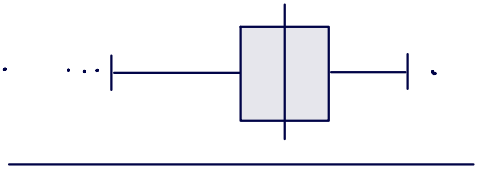


Gaussian: $K_\alpha(x, x_i) = \frac{1}{\sqrt{2\pi\alpha^2}} e^{-\frac{(x-x_i)^2}{2\alpha^2}}$

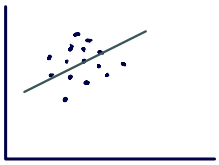
Box Plots: Summarize several characteristics

whiskers, lower quartile, median, upper quartile, outliers

↑ 1.5 * IQR = Q1/Q3

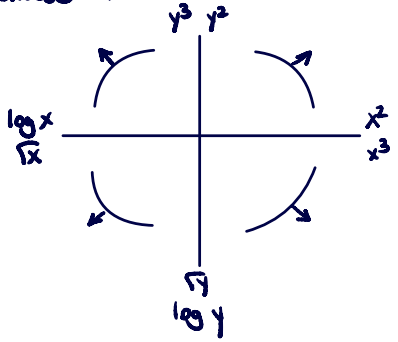


Scatter Plots: Plotting 2 quantitative variables on same plot



Tukey-Mosteller Bulge Diagram

Help choose transformations to linearize



---

8 Modeling

Model: useful simplification of reality
allows us to understand the world we live in & predict the value of unseen data

Modeling Process

1. Choose a Model (constant, linear, non-linear)
2. Choose an Objective Function (loss function)
3. Fit model by optimizing Objective Function (analytical, numerical)

Loss function

L_2 squared loss : $(y-\hat{y})^2$ ← mean of dataset minimizes

L_1 absolute loss : $|y-\hat{y}|$ ← median of dataset minimizes

Want to reduce average loss across all points

$$\frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{y}_i)$$

with L_2: Mean Squared Error (MSE) smooth

with L_1: Mean Absolute Error (MAE) robust to outliers

Model

Constant Model: $\hat{y} = \theta$

$\hat{\theta}_0 = \bar{y} - \hat{\theta}_1 \bar{x}$

Simple Linear Regression: $\hat{y} = \theta_0 + \theta_1 x$

Multiple Linear Regression: $\hat{y} = \theta_0 + \theta_1 x_1 + \ldots + \theta_p x_p = \theta_0 + \sum_{j=1}^{p} \theta_j x_j$

$\hat{\theta}$ is value that minimizes avg loss

1. Can take derivative of avg loss w/ respect to $\theta / \hat{y}$
2. set to 0, solve for $\theta / \hat{y}$

INTECEPT

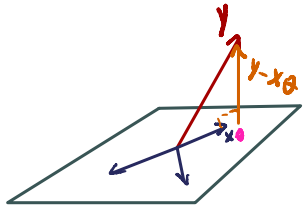
$$\hat{y} = X\theta$$

vector w/ dimensions $n \times 1$

matrix w/ rows as observations, cols as features

$$\begin{bmatrix} 1 & x_{11} & \cdots & x_{1p} \\ \vdots & & & \\ 1 & x_{n1} & \cdots & x_{np} \end{bmatrix}$$

vector w/ optimal $\hat{\theta}$ dimension $(p+1) \times 1$



Least Squares Regression: want θ where residual is orthogonal to span(X)

$$\hat{\theta} = (X^T X)^{-1} X^T y$$

if $X^T X$ is full rank