NYU Center for Data Science

DS-UA 112
Introduction to Data Science

Lecture 11

Text I – Working with Characters and Strings

Reminders

- Homework 3
  - Due Friday October 18
- Project 1
  - Due Sunday October 20
- Cluster
  - Maintenance on Monday October 14
  - Please download any materials for assignments

Visualization

- Finding the right visualization…
  - One Quantitative – Histogram / Boxplot
  - One Qualitative – Bar Chart
  - Multiple Quantitative – Line Plot, Scatter Plot
  - Multiple Qualitative – Overlay Bar Charts
  - Mix Qualitative / Quantitative
    - Side by side Boxplots
    - Overlay Histograms
    - Color and Size in Scatter Plot
    - Heat maps

Juxtapose vs Superimpose

Context

- Plot title
- Axes labels
- Reference lines and markers for important values
- Legend and Labels
- Captions that describe the data and its important features

Visualization

- Finding Understandable Visualizations…
  - Formatting
    - Remove extraneous colors, accents, effects
  - Scale
    - Count vs Frequency
  - Conditioning
    - Jiggling Baseline
    - Jittering Points
  - Transformations
    - Smoothing
    - Logarithm for Skewed Data
    - Dimension Reduction

Don’t stack bars!

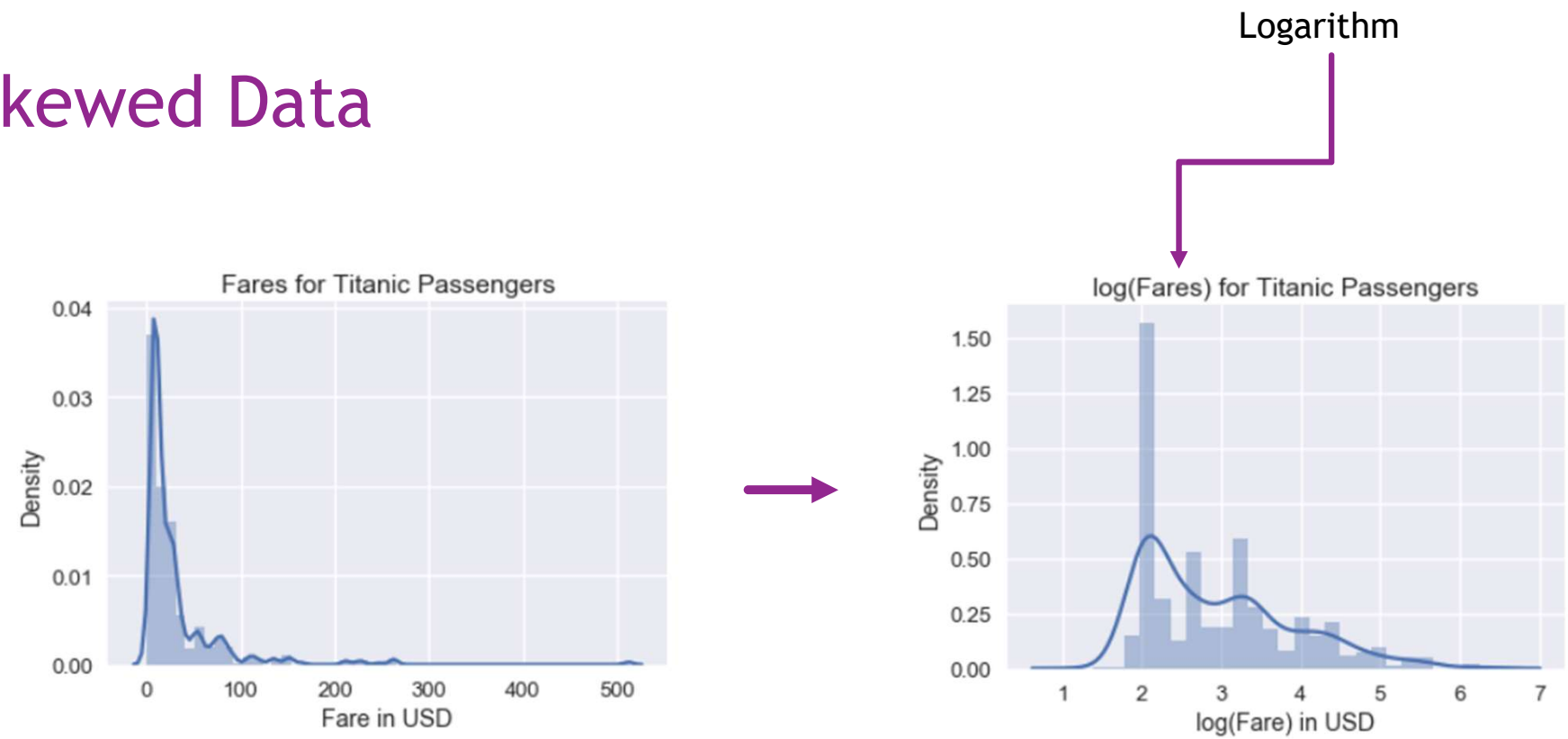
Objectives

- Applying transformations to data
  - Power Transformation
  - Smooth data
  - Reduce the size
- Manipulate strings
  - Operations such as strip and replace
  - Match patterns of characters in strings to locate or replace

Agenda

- Review
  - Transforming Data for Visualization
- Lesson
  - String Methods
  - Regular Expressions
- Demo
  - PCA
  - Police Reports

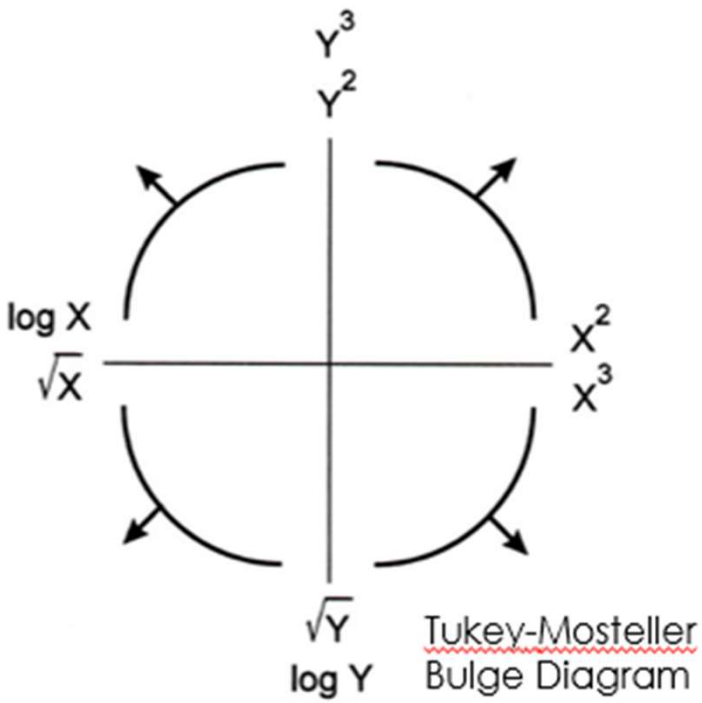
Skewed Data


Logarithm
Fares for Titanic Passengers
Density
0.04
0.03
0.02
0.01
0.00
0
100
200
300
400
500
Fare in USD
log(Fares) for Titanic Passengers
Density
1.50
1.25
1.00
0.75
0.50
0.25
0.00
1
2
3
4
5
6
7
log(Fare) in USD

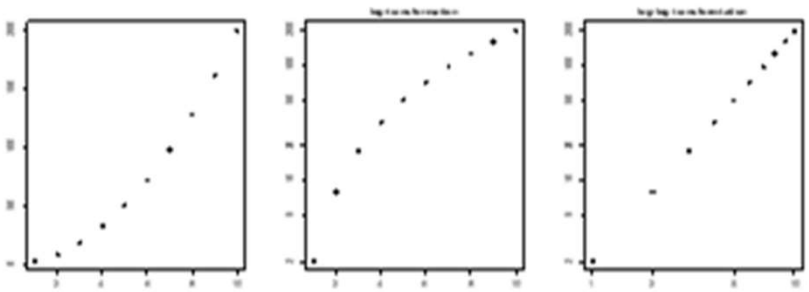
Generalization of Logarithmic Transformation

$$y_i^{(\lambda)} = \begin{cases} \dfrac{y_i^{\lambda} - 1}{\lambda} & \text{if } \lambda \neq 0, \\ \ln y_i & \text{if } \lambda = 0, \end{cases}$$

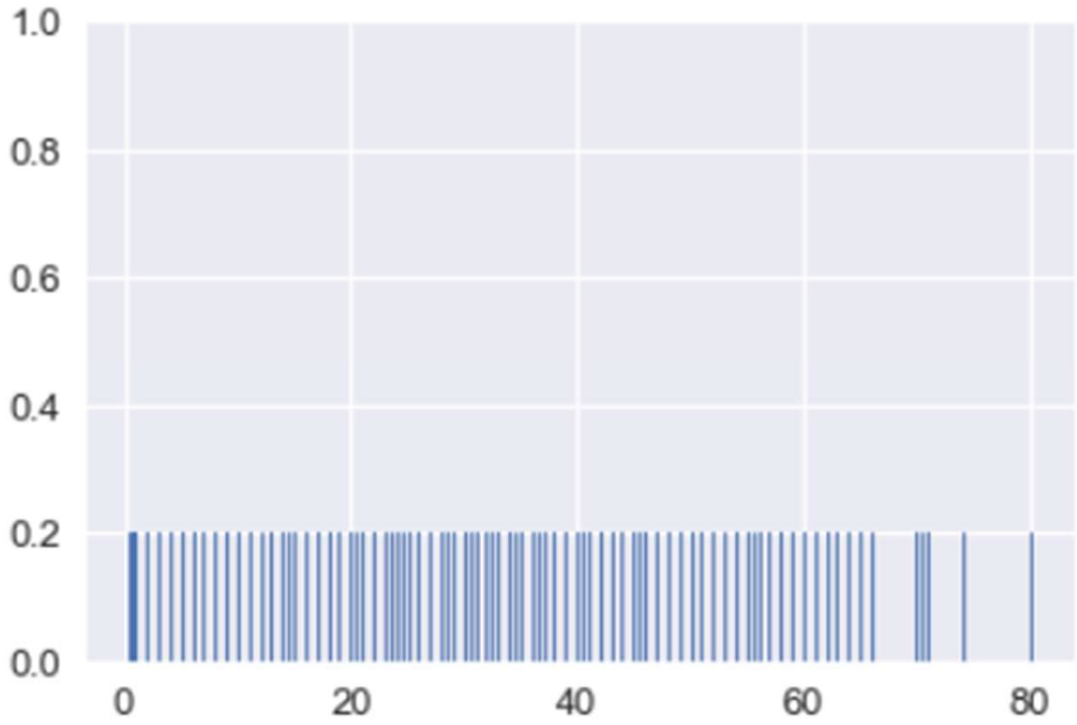
Generalization of Logarithmic Transformation



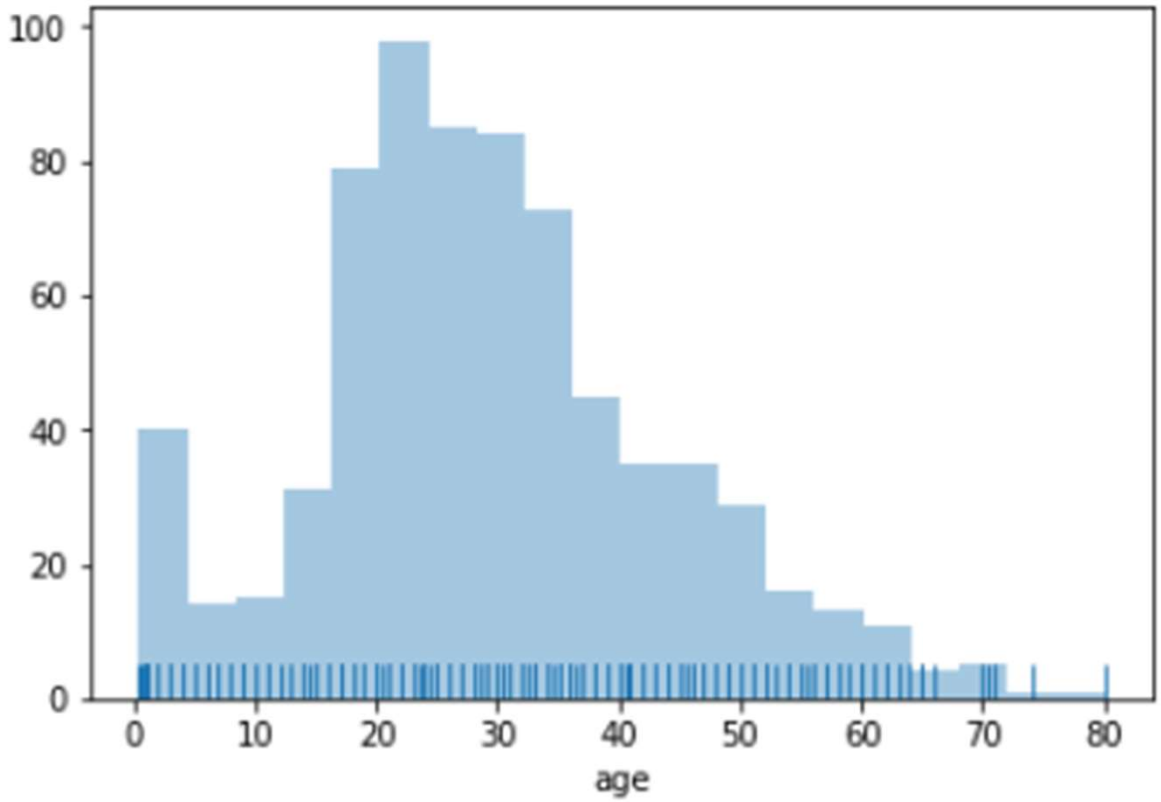
Tukey-Mosteller Bulge Diagram



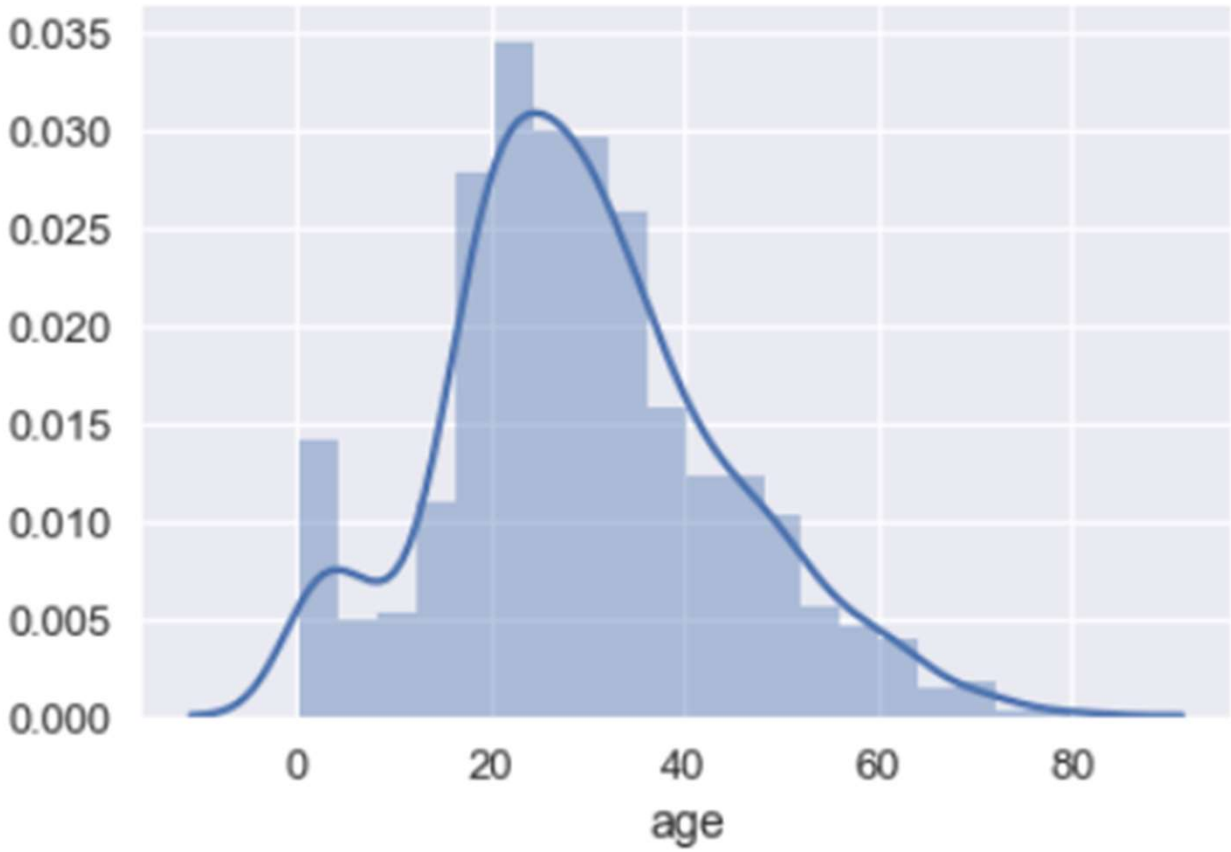
Smoothing



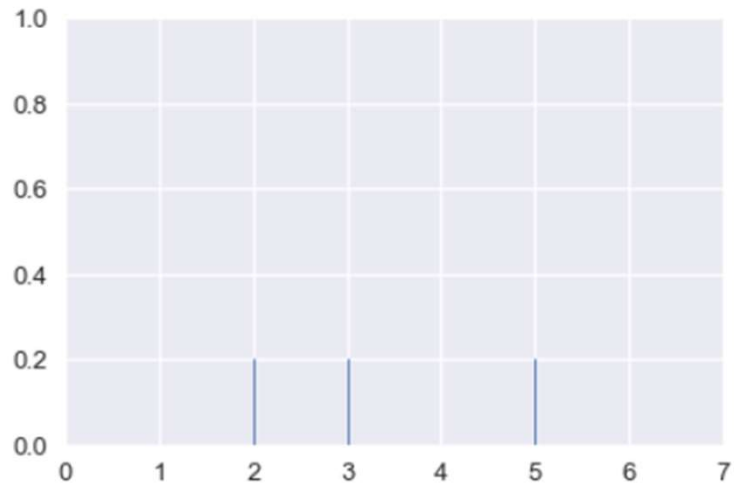
Smoothing



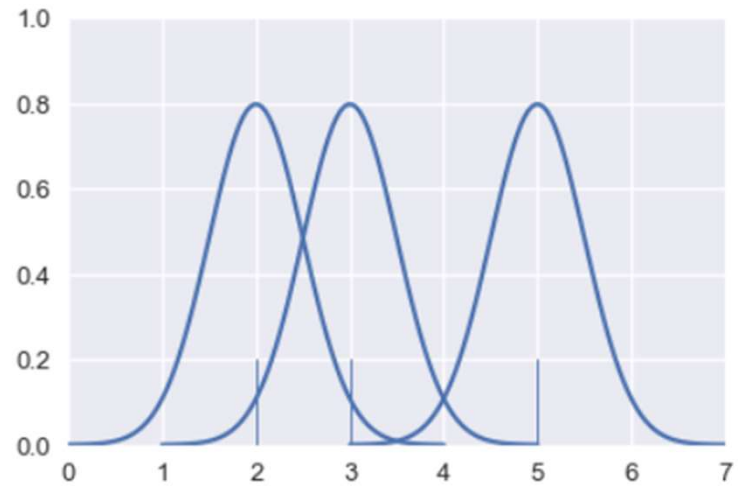
Smoothing



Density

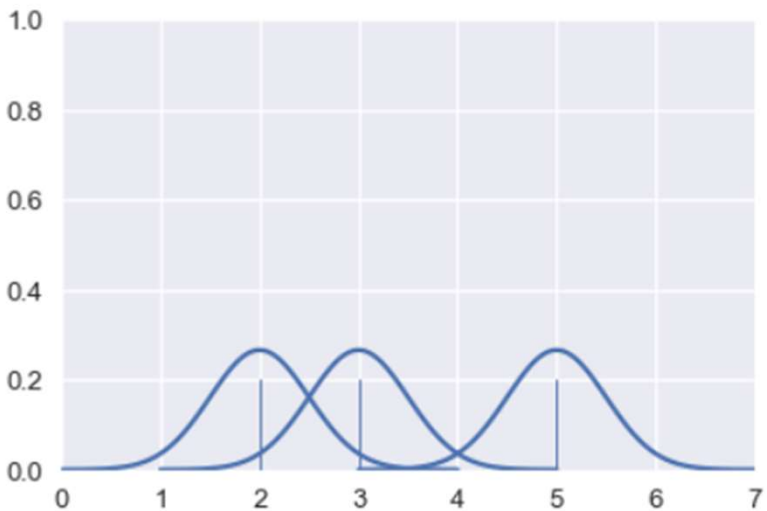


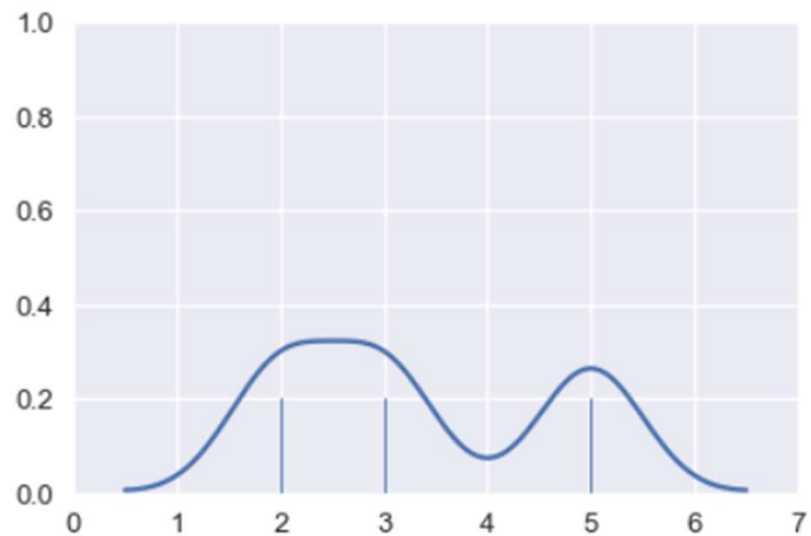
Bell Curves



Density

Set kde = True for density





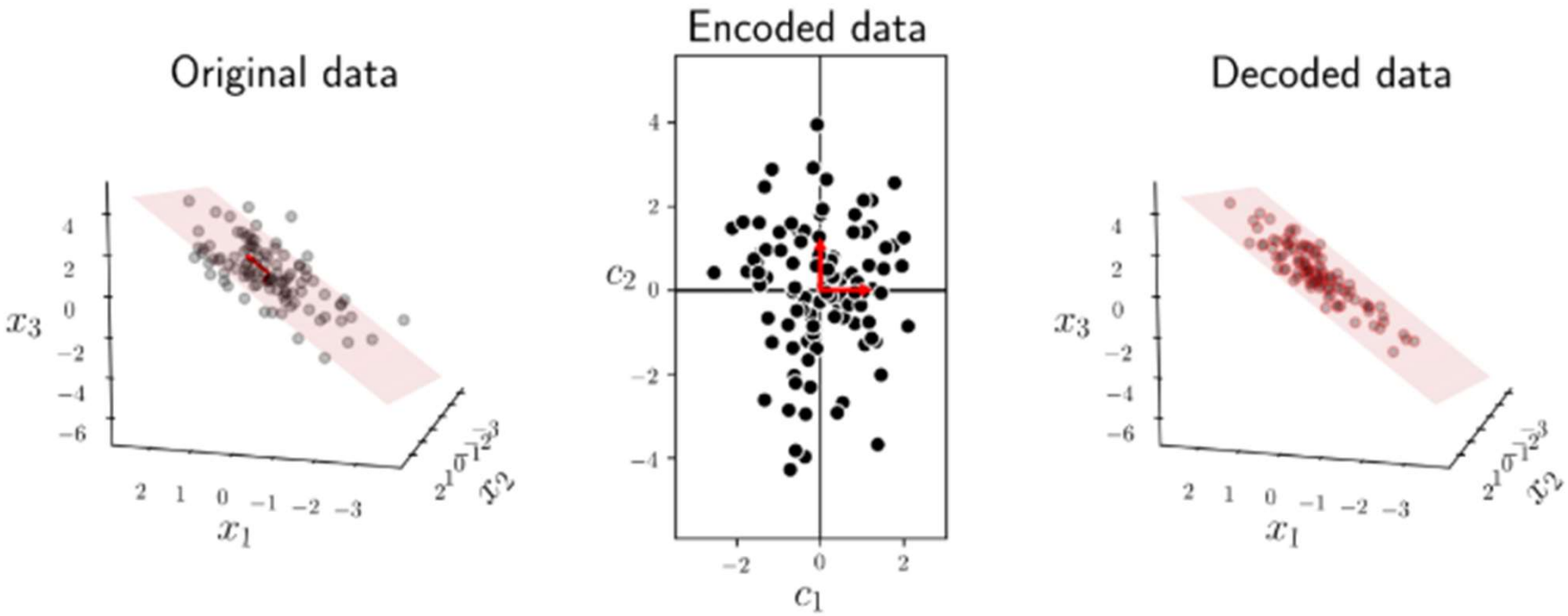
Reducing Size of Data

- With many features…
  - Overlapping data points
    - Data doesn’t fill the space
    - Some variable is a combination of other variables
  - There are more features than observations!

Reducing Size of Data

- Make couple new features that…
  - Reflect original features
  - Preserve as much information as possible
- With the new features we have…
  - Eliminated dimensions
  - Brought out strong patterns
- …so for the analysis will be more manageable

Reducing Size of Data



Reducing Size of Data

	England	N Ireland	Scotland	Wales
Alcoholic drinks	375	135	458	475
Beverages	57	47	53	73
Carcase meat	245	267	242	227
Cereals	1472	1494	1462	1582
Cheese	105	66	103	103
Confectionery	54	41	62	64
Fats and oils	193	209	184	235
Fish	147	93	122	160
Fresh fruit	1102	674	957	1137
Fresh potatoes	720	1033	566	874
Fresh Veg	253	143	171	265
Other meat	685	586	750	803
Other Veg	488	355	418	570
Processed potatoes	198	187	220	203
Processed Veg	360	334	337	365
Soft drinks	1374	1506	1572	1256
Sugars	156	139	147	175

Reducing Size of Data

This dimension is a combination of +potatoes – fresh fruit – fish

String Methods

	County	State
0	De Witt County	IL
1	Lac qui Parle County	MN
2	Lewis and Clark County	MT
3	St John the Baptist Parish	LA

	County	Population
0	DeWitt	16,798
1	Lac Qui Parle	8,067
2	Lewis & Clark	55,716
3	St. John the Baptist	43,044

String Methods

	County	State
0	dewitt	IL
1	lacquiparle	MN
2	lewisandclark	MT
3	stjohnthebaptist	LA

	County	Population
0	dewitt	16,798
1	lacquiparle	8,067
2	lewisandclark	55,716
3	stjohnthebaptist	43,044

	County	State	Population
0	dewitt	IL	16,798
1	lacquiparle	MN	8,067
2	lewisandclark	MT	55,716
3	stjohnthebaptist	LA	43,044

String Methods

```
def clean_county(county):
    return (county
            .lower()
            .strip()
            .replace(' county', '')
            .replace(' parish', '')
            .replace('&', 'and')
            .replace(' ', '')
            .replace('.', ''))
```

```
([clean_county(county) for county in state['County']],
 [clean_county(county) for county in population['County']]
)
```

Regular Expressions

Example: [0-9]{3}-[0-9]{2}-[0-9]{4}

3 of any digit, then a dash, then 2 of any digit, then a dash, then 4 of any digit.

```
text = "My social security number is 123-45-6789.";
pattern = "[0-9]{3}-[0-9]{2}-[0-9]{4}"
re.findall(pattern, text)
```

Regular Expressions

operation	order	example	matches	does not match
concatenation	3	AABAAB	AABAAB	any other string
or	4	AA|BAAB	AA BAAB	any other string
closure (zero or more)	2	AB*A	AA ABBBBBBA	AB ABABA
parentheses	1	A(A|B)AAB	AAAAB ABAAB	any other string
		(AB)*A	A ABABABABA	AA ABBA

Regular Expressions

operation	example	matches	does not match
any character (except newline)	`.U.U.U.`	`CUMULUS` `JUGULUM`	`SUCCUBUS` `TUMULTUOUS`
character class	`[A-Za-z][a-z]*`	`word` `Capitalized`	`camelCase` `4illegal`
at least one	`jo+hn`	`john` `jooooohn`	`jhn` `jjohn`
zero or one	`joh?n`	`jon` `john`	`any other string`
repeated exactly {a} times	`j[aeiou]{3}hn`	`jaoehn` `jooohn`	`jhn` `jaeiouhn`
repeated from a to b times: {a,b}	`j[ou]{1,2}hn`	`john` `juohn`	`jhn` `jooohn`

Regular Expressions

operation	example	matches	does not match
built-in character classes	\w+ \d+	fawef 231231	this person 423 people
character class negation	[^a-z]+	PEPPERS3982 17211!↑å	porch CLAmS
escape character	cow\.com	cow.com	cowscom

Regular Expressions

operation	example	matches	does not match
beginning of line	^ark	ark two ark o ark	dark
end of line	ark$	dark ark o ark	ark two
non-greedy qualifier	5.*?5	5005 55	5005005

Regular Expressions

Note that not wildcard!

Char	Description	Example	Matches	Doesn't Match
.	Any character except \n	`...`	abc	ab abcd
[ ]	Any character inside brackets	`[cb.]ar`	car .ar	jar
[^]	Any character *not* inside brackets	`[^b]ar`	car par	bar ar
*	≥ 0 or more of last symbol	`[pb]*ark`	bbark ark	dark
+	≥ 1 or more of last symbol	`[pb]+ark`	bbpark bark	dark ark
?	0 or 1 of last symbol	`s?he`	she he	the
{*n*}	Exactly *n* of last symbol	`hello{3}`	hellooo	hello
|	Pattern before or after bar	`we|[ui]s`	we us is	e s
\	Escapes next character	`\[hi\]`	[hi]	hi
^	Beginning of line	`^ark`	ark two	dark
$	End of line	`ark$`	noahs ark	noahs arks

Regular Expressions

Description	Bracket Form	Shorthand
Alphanumeric character	`[a-zA-Z0-9]`	`\w`
Not an alphanumeric character	`[^a-zA-Z0-9]`	`\W`
Digit	`[0-9]`	`\d`
Not a digit	`[^0-9]`	`\D`
Whitespace	`[\t\n\f\r\p{Z}]`	`\s`
Not whitespace	`[^\t\n\f\r\p{z}]`	`\S`

Take-Aways

- Transforming Data
  - Logarithm and Powers
  - Smoothing
  - Reducing Dimension
- Working with Text
  - String Methods
  - Regular Expressions