Data Preprocessing

COSC 757: Spring 2016

Types of Data Sets

- Record
  - Relational records
  - Data matrix, e.g., numerical matrix, crosstabs
  - Document data: text documents: term-frequency vector
  - Transaction data
- Graph and network
  - World Wide Web
  - Social or information networks
  - Molecular Structures
  - Road network
- Ordered
  - Video data: sequence of images
  - Temporal data: time-series
  - Sequential Data: transaction sequences
  - Genetic sequence data
- Spatial, image and multimedia:
  - Spatial data: points, lines, polygons
  - Image data:
  - Video data:

Data Objects

- Data sets are made up of data objects.
- A **data object** represents an entity.
- Examples:
  - sales database: customers, store items, sales
  - medical database: patients, treatments
  - university database: students, professors, courses
- Also called *samples , examples, instances, data points, objects, tuples*.
- Data objects are described by **attributes**.
- Database rows -> data objects; columns ->attributes.

Attributes

- **Attribute (**or **dimensions, features, variables**): a data field, representing a characteristic or feature of a data object.
  - *E.g., customer _ID, name, address*

Categorical Attribute Types

- **Nominal:** categories, states, or “names of things”
  - *Hair_color =* {*auburn, black, blond, brown, grey, red, white*}
  - marital status, occupation, ID numbers, zip codes
- **Binary**
  - Nominal attribute with only 2 states (0 and 1)
  - <u>Symmetric binary</u>: both outcomes equally important
    - e.g., gender
  - <u>Asymmetric binary</u>: outcomes not equally important.
    - e.g., medical test (positive vs. negative)
    - Convention: assign 1 to most important outcome (e.g., HIV positive)
- **Ordinal**
  - Values have a meaningful order (ranking) but magnitude between successive values is not known.
  - *Size =* {*small, medium, large*}*,* grades, army rankings

Numeric Attribute Types

- Quantity (integer or real-valued)
- **Interval**
    - Measured on a scale of **equal-sized units**
    - Values have order
        - E.g., *temperature in C˚or F˚, calendar dates*
    - No true zero-point
- **Ratio**
    - Inherent **zero-point**
    - We can speak of values as being an order of magnitude larger than the unit of measurement (10 K˚ is twice as high as 5 K˚).
        - e.g., *temperature in Kelvin, length, counts, monetary quantities*

Discrete vs. Continuous Attributes

- **Discrete Attribute**
  - Has only a finite or countable infinite set of values
    - E.g., zip codes, profession, or the set of words in a collection of documents
  - Sometimes, represented as integer variables
  - Note: Binary attributes are a special case of discrete attributes
- **Continuous Attribute**
  - Has real numbers as attribute values
    - E.g., temperature, height, or weight
  - Practically, real values can only be measured and represented using a finite number of digits
  - Continuous attributes are typically represented as floating-point variables

Why do we preprocess data?

- Raw data is often unprocessed, incomplete, or noisy
- Likely to contain
  - Obsolete/redundant fields
  - Missing values
  - Outliers
  - Data in form not suitable for data mining
  - Values not consistent with policy or common sense

Why do we preprocess data?

- For data mining purposes, database values must undergo data cleaning and data transformation
- Data from legacy databases
  - Not looked at in years
  - Expired
  - No longer relevant
  - Missing
- Minimize GIGO
- Effort for data preparation = 10% to 60% of data mining process…

Can you find the problems in this dataset?

Customer ID	Zip	Gender	Income	Age	Marital Status	Transaction Amount
1001	10048	M	75000	C	M	5000
1002	J2S7K7	F	—40000	40	W	4000
1003	90210		10000000	45	S	7000
1004	6269	M	50000	0	S	1000
1005	55101	F	99999	30	D	3000

Handling Missing Data

- Missing values pose problems to data analysis methods
- More common in databases containing a large number of fields
- Absense of information rarely beneficial to task of analysis
- Having more data is always better
- Careful analysis is required to handle missing data

Consider the Following Dataset

	mpg	cubic inches	hp	brand
1	14.000	350	165	US
2	31.900		71	Eurpoe
3	17.000	302	140	US
4	15.000	400	150	
5	37.000	89	62	Japan

Which of the following is the best option?

1. Delete records containing missing values.
2. Replace missing values with a constant specified by the analyst.
3. Replace missing values with mode or mean.
4. Replace missing values with random values.

Data Imputation Methods

- Imputation of Missing Data – What is the likely value, given records other attribute values?
- Example: From two samples on the previous slide, American cars would be expected to have a higher horse power and cubic inches
- Tools like multiple regression and classification can be used for this purpose (more on that later).

Identifying Misclassifications

- Check classification labels, to verify values valid and consistent
- Example: Table below – Frequency distribution for origin of manufacture of automobiles
  - Frequency distribution shows five classes: USA, France, US, Europe, and Japan
  - Count for USA = 1 and France = 1?
  - Two records classified inconsistently with respect to origin of the manufacture
  - Maintain consistency by labeling USA → US, and France → Europe

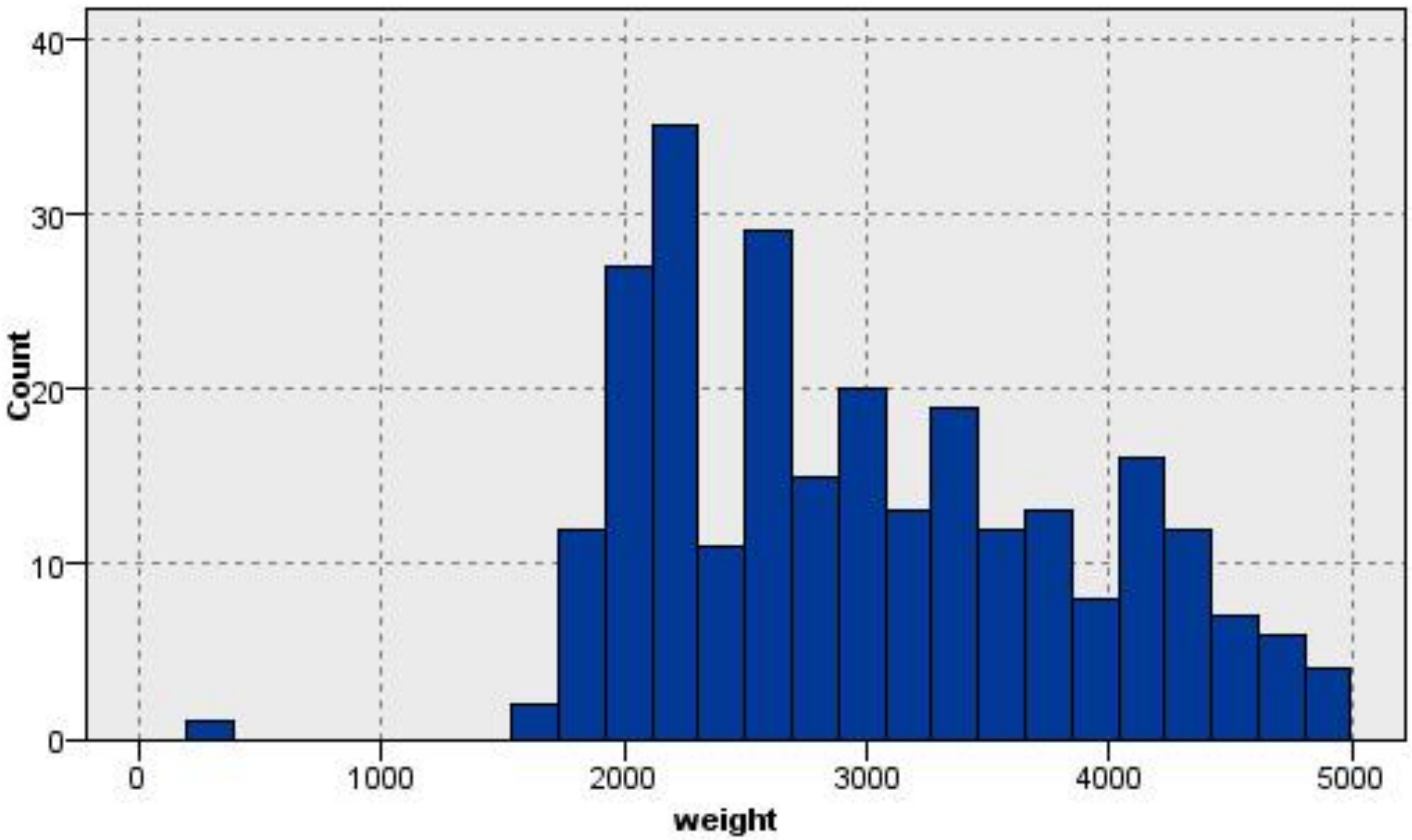
Brand	Frequency
USA	1
France	1
US	156
Europe	46

Identifying Outliers

- Outliers are extreme values that go against the trend of the remaining data
- Outliers may represent errors in data entry
- Even if valid data point, certain statistical methods are very sensitive to outliers and may produce unstable results

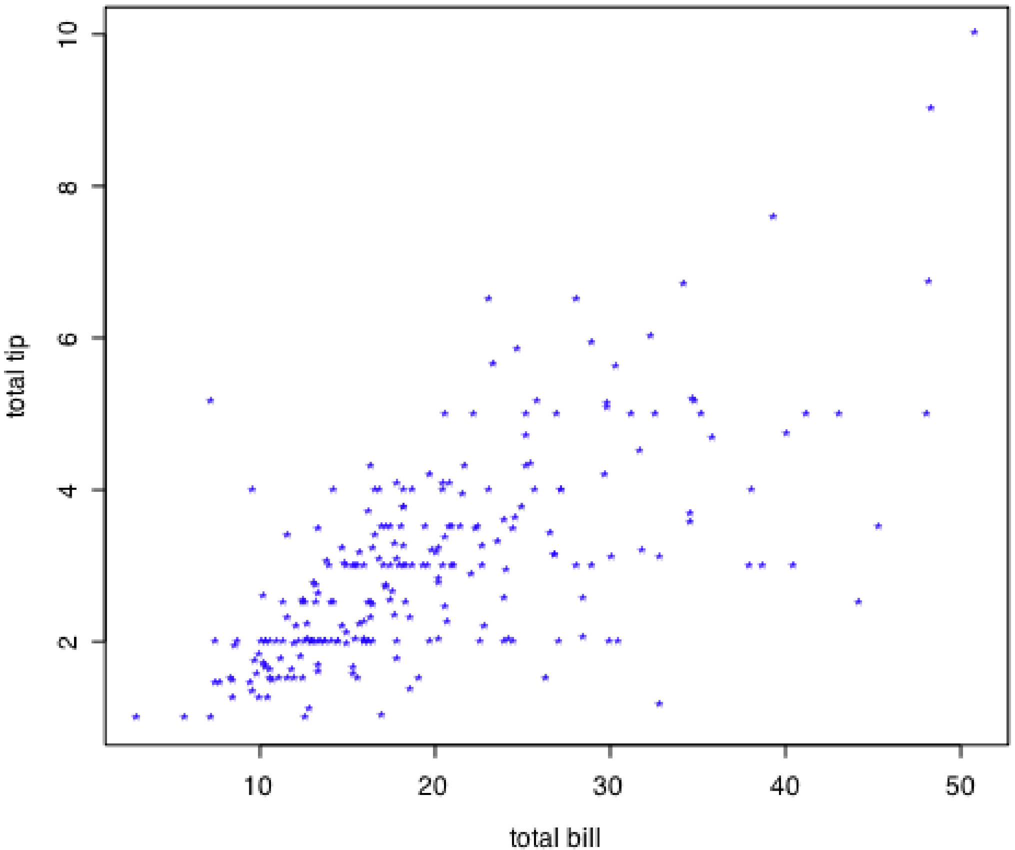
Outliers: Graphical Methods

- Method 1 - Histogram



Outliers: Graphical Methods

- Method 2 – 2D Scatter Plot



Basic Statistical Descriptions of Data

- Motivation
  - To better understand the data: central tendency, variation and spread
- Data dispersion characteristics
  - median, max, min, quantiles, outliers, variance, etc.
- Numerical dimensions correspond to sorted intervals
  - Data dispersion: analyzed with multiple granularities of precision
  - Boxplot or quantile analysis on sorted intervals
- Dispersion analysis on computed measures
  - Folding measures into numerical dimensions
  - Boxplot or quantile analysis on the transformed cube

Measuring the Central tendency

$$\mu = \frac{\sum x}{N}$$

- Mean (algebraic measure) (sample vs. population):

  Note: *n* is sample size and *N* is population size.

  - Weighted arithmetic mean:
  - Trimmed mean: chopping extreme values

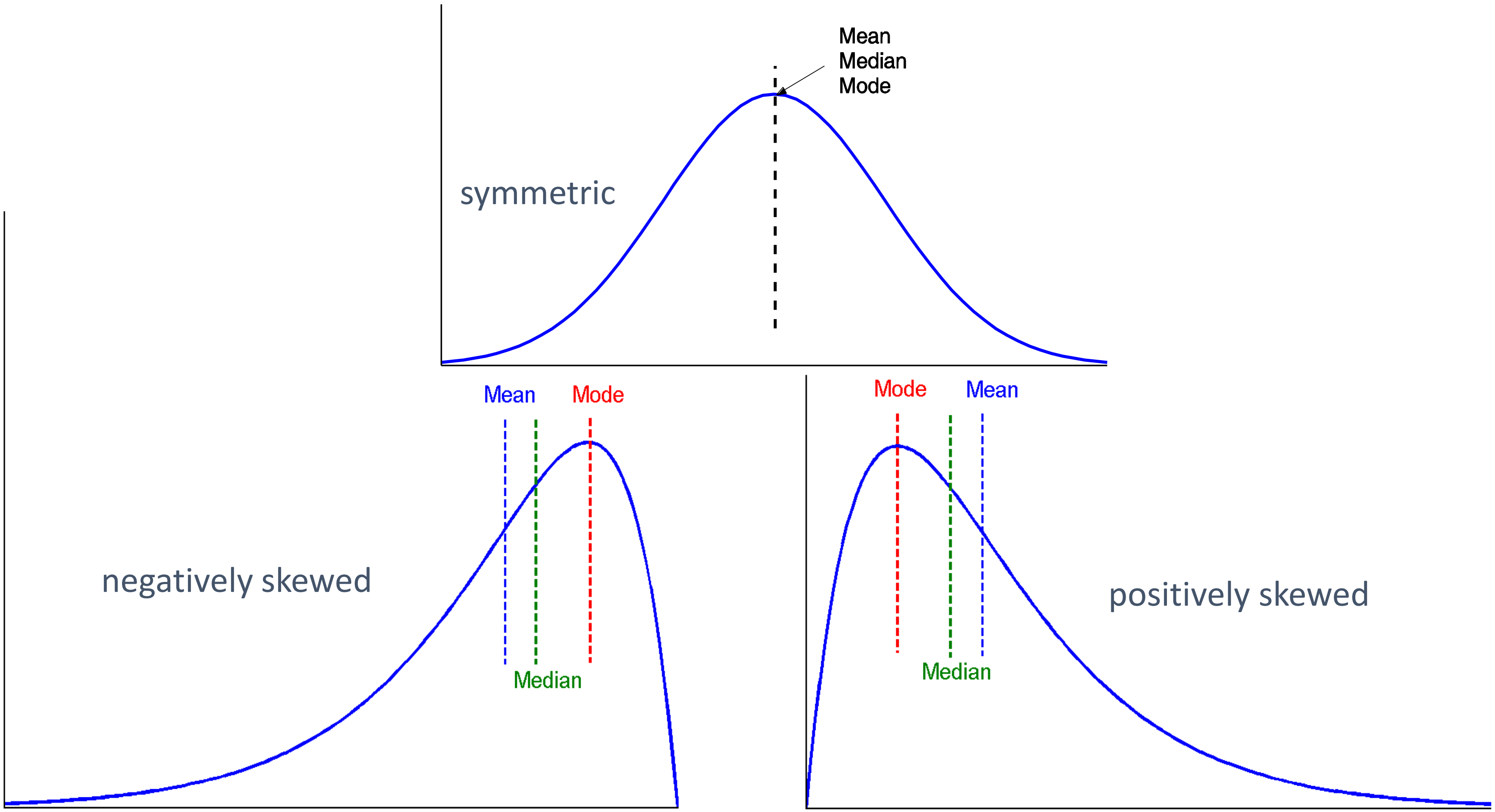
$$\overline{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i}$$

- Median:
  - Middle value if odd number of values, or average of the middle two values otherwise
- Mode
  - Value that occurs most frequently in the data
  - Unimodal, bimodal, trimodal
  - Empirical formula:

$$mean - mode = 3 \times (mean - median)$$

Symmetric vs. Skewed Data

- Median, mean and mode of symmetric, positively and negatively skewed data



Measuring the Dispersion of Data

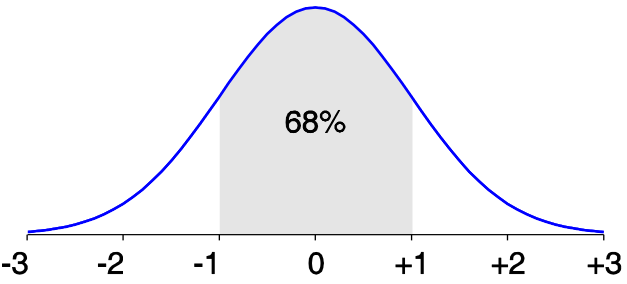
- Quartiles, outliers and boxplots
  - **Quartiles**: Q_1 (25th percentile), Q_3 (75th percentile)
  - **Inter-quartile range**: IQR = $Q_3 - Q_1$
  - **Five number summary**: min, Q_1, median, Q_3, max
  - **Boxplot**: ends of the box are the quartiles; median is marked; add whiskers, and plot outliers individually
  - **Outlier**: usually, a value higher/lower than 1.5 x IQR
- Variance and standard deviation – distance of observations from the mean
  - **Variance**: (algebraic, scalable computation)

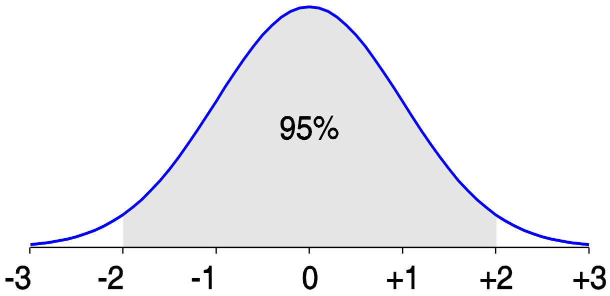
$$\sigma^2 = \frac{1}{N}\sum_{i=1}^{n}(x_i - \mu)^2 = \frac{1}{N}\sum_{i=1}^{n} x_i^{\ 2} - \mu^2$$

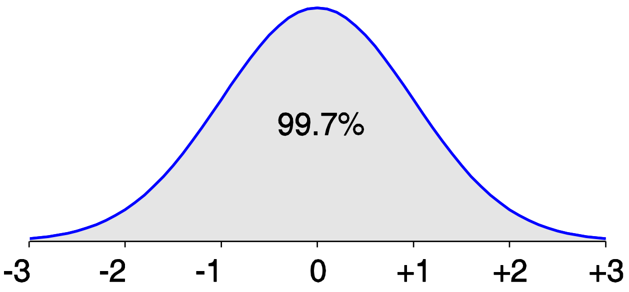
  - **Standard deviation** *s (or σ)* is the square root of variance s^2 (or σ^2)

Normal Distribution Curve

- The normal (distribution) curve
  - From μ–σ to μ+σ: contains about 68% of the measurements (μ: mean, σ: standard deviation)
  - From μ–2σ to μ+2σ: contains about 95% of it
  - From μ–3σ to μ+3σ: contains about 99.7% of it







Data Transformation

- Variables tend to have ranges different from each other
- For example:
  - Batting average [0.0,0.400]
  - Home runs [0,70]
- Some data mining algorithms are adversely affected by differences in variable ranges
- Variables with greater ranges tend to have larger influence on data model results
- Standardizing scales the effect each variable has on results
- Neural Networks and other algorithms that make use of distance measures benefit from normalization
- Two of the prevalent methods will be reviewed

Min-Max Normalization

- Determines how much greater field value is than minimum value for field
- Scales this difference by field's range

$$X^* = \frac{X - \min(X)}{\text{range}(X)} = \frac{X - \min(X)}{\max(X) - \min(X)}$$

Find Min-Max normalization for cars weighing 1613, 3384 and 4997 pounds, respectively

Where: min(X) = 1613 and max(X) = 4997

Car	Weightlbs	Formula	Result	Comments
Ultra-light vehicle	X = 1613	$X^* = \frac{1613 - 1613}{4997 - 1613}$	X* = 0	Represents the minimum value in this variable, and has min-max normalization of zero.
Mid-range vehicle	X = 3384	$X = \frac{3884 - 1613}{4997 - 1613}$	X* = 0.5	Weight exactly half-weight between the ligthest and the heaviest vehicle, and has min-max normalization of 0.5.
Heaviest vehicle	X = 4997	$X = \frac{4997 - 1613}{4997 - 1613}$	X* = 1	Heaviest vehicle of the dataset has min-max normalization of one.

Z-Score Standardization

- Widely used in statistical analysis
- Takes difference between field value and field value mean
- Scales this difference by field's standard deviation

$$X^* = \frac{X - \text{mean}(X)}{\text{SD}(X)}$$

Find Z-scode standardization for cars weighing 1613, 3384 and 4997 pounds, respectively

Where: mean(X) = 3005.49 and SD(X) = 852.65

Car	Weightlbs	Formula	Result	Comments
Ultra-light vehicle	X = 1613	$X^* = \frac{1613 - 3005.49}{852.646}$	X* ≈ -1.63	Data values below the mean will have negative Z-score standardization.
Mid-range vehicle	X = 3384	$X = \frac{3884 - 3005.49}{852.646}$	X* ≈ 0	Values falling exactly on the mean will have zero (0) Z-score
Heaviest vehicle	X = 4997	$X = \frac{4997 - 3005.49}{852.646}$	X* ≈ 2.34	Data values about the mean will have a negative Z-score standardization

Decimal Scaling

- Ensures that normalized values lies between -1 and 1
- Defined as:

$$X^* = \frac{X}{10^d}$$

where *d* represents the number of digits in the data value with the largest absolute value.

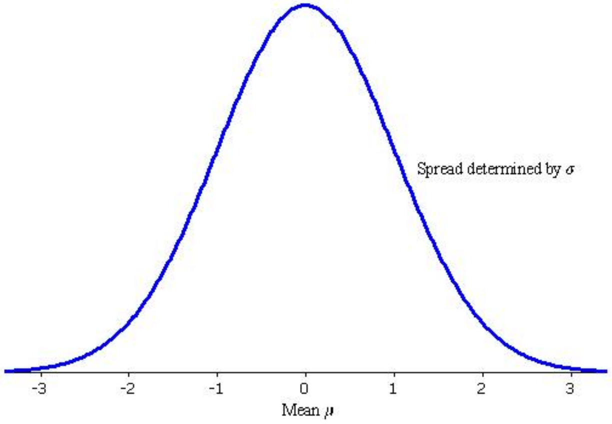
- For the weight data, the largest absolute value is |4997|=4997, with *d*=4 digits
- Decimal scaling for the minimum and maximum weights are:

$$Min: X^*_{decimal} = \frac{1613}{10^4} = 0.1613$$

$$Max: X^*_{decimal} = \frac{4997}{10^4} = 0.4997$$

Transformations to Achieve Normality

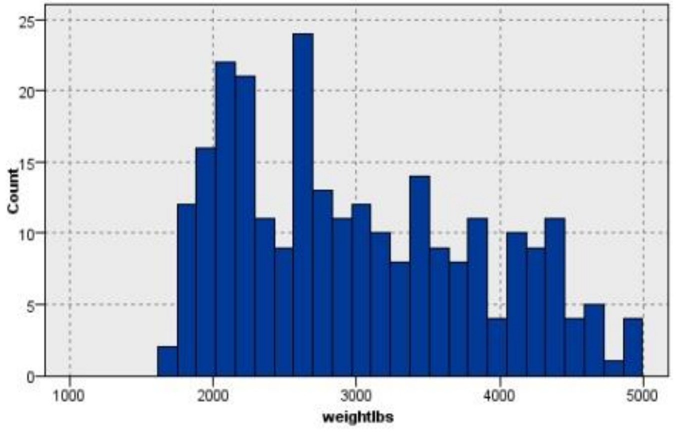
- Some data mining algorithms and statistical methods require *normally distributed* variables
- Normal distribution
  - Continuous probability distribution known as the ‘bell curve’ (symmetric)
  - Centered and mean μ (myu) and spread given by σ (sigma)

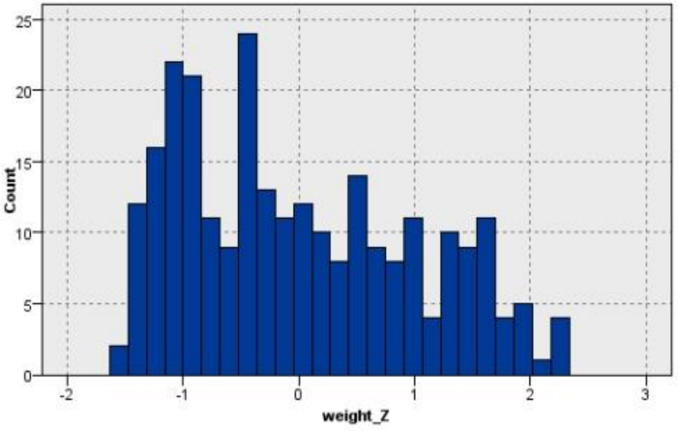


Standard normal Z-distribution
with μ=0 and σ=1

Transformations to Achieve Normality

- Misconception – Z-score standardization results in a normally distribution
- Z-score standardized variables do have μ=0 and σ=1, but the distribution may be skewed (not symmetric)



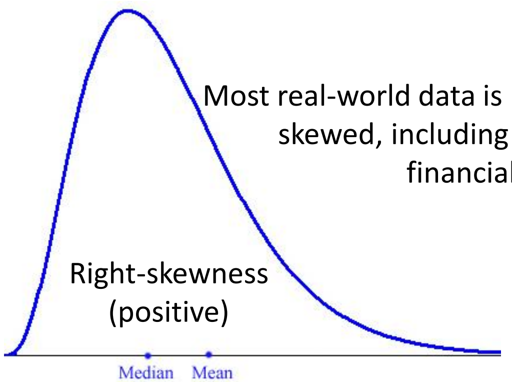


Measuring Skewness

- Statistics for measuring the skewness of a distribution:

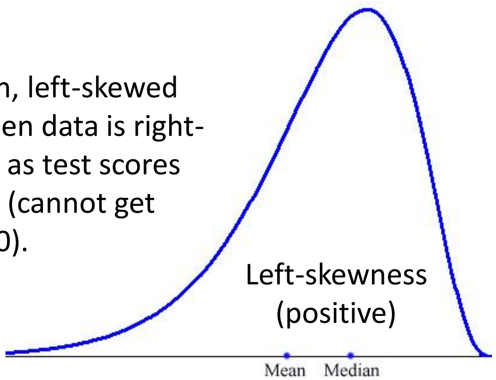
$$Skewness = \frac{3(mean\ - median)}{standard\ deviation}$$

  - Right-skewness data – Is positive, as mean is greater than the median
  - Left skewness data – Mean is smaller than the median, generating negative values
  - Perfectly symmetric data – mean, median and mode are equal, so skewness is zero



Most real-world data is right-skewed, including most financial data

Not as common, left-skewed data occurs when data is right-censored, such as test scores on an easy test (cannot get higher than 100).

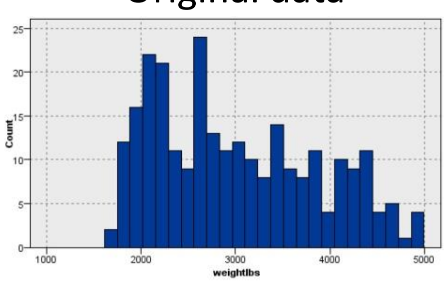


Transformations to Achieve Normality

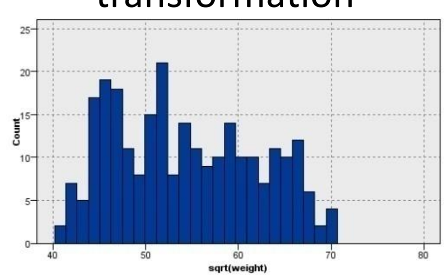
- To eliminate skewness, we must apply a transformation to the data
  - This makes the data symmetric and makes it "more normally distributed"
- Common transformations are:

Natural Log	Square Root	Inverse Square Root
$\ln(weight)$	$\sqrt{weight}$	$\dfrac{1}{\sqrt{weight}}$

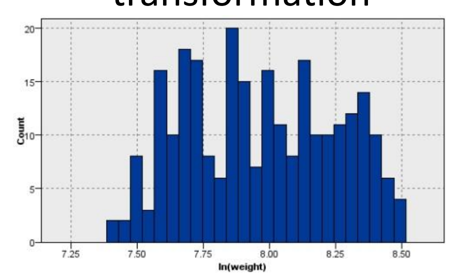
Original data



Square root transformation



Natural log transformation



Transformations to Achieve Normality

- Example #1: Apply SQRT and LN transformations to weight data

For SQRT(weight):

sqrt(weight)

Statistics

Mean	54.280
Standard Deviation	7.709
Median	53.245

$$Skewness\big(sqrt(weight)\big) = \frac{3(54.280 - 53.245)}{7.709} \approx 0.40$$

For LN(weight):

ln(weight)

Statistics

Mean	7.968
Standard Deviation	0.284
Median	7.950

$$Skewness\big(sqrt(weight)\big) = \frac{3(7.968 - 53.245)}{0.284} \approx 0.19$$

Transformations to Achieve Normality

- Example #2: Apply inverse square root transformation to weight data

For INVERSE_SQRT(weight):

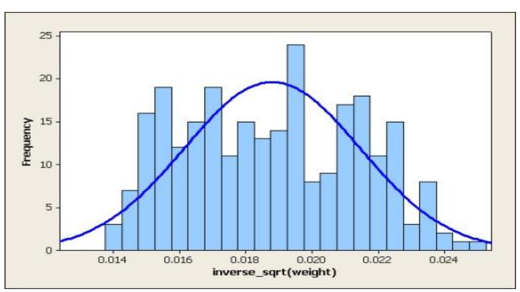
inverse_sqrt(weight)

Statistics

Mean	0.019
Standard Deviation	0.003
Median	0.019

$$Skewness\big(sqrt(weight)\big) = \frac{3(0.019 - 0.019)}{0.003} = 0$$

Important: There is nothing special about the inverse square root transformation. It just worked with the skewness in the weight data



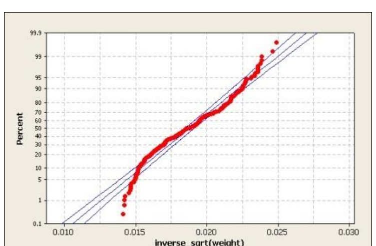
Histogram for inv_sqrte(weight) with normal distribution curve overlay

Notice that while we have achieved symmetry, we have not reached normality (the distribution does not match the normal curve)

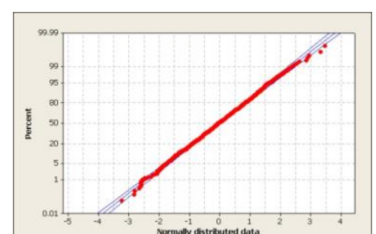
Checking for Normality

- After achieving symmetry, we must also check for normality
- The Normal Probability Plot
  - Plots the quantiles for a particular distribution against the quantiles of the standard normal distribution
  - Similar to percentile, pth quantile of a distribution is value xp, such that p% of the distribution values are less than or equal to xp
  - If the bulk of the points fall on a straight line, the distribution is normal; systematic deviations indicate nonnormality
- As expected, the normal probability plot for the inverse_sqrt(weigth) indicates nonnormality
- While normality was not achieved, algorithms requiring normality usually do fine when supplied with data that is symmetric and unimodal

Normal probability plots



Plot for inverse_sqrt(weight) has systematic deviations that indicate nonnormality



Plot for normally distributed data

Transformations to Achieve Normality

- Detransformation – After completing the analysis, it is required to “de-transform” the data
- Example for the Inverse Square Root:

Transformation $\rightarrow$ $y = \dfrac{1}{\sqrt{x}}$

De-transformation $\leftarrow$ $x = \dfrac{1}{y^2}$

- Results provided by algorithm in the transformed scale would have to be converted back using the de-transformation formula

Numerical Methods for Identifying Outliers

- Using Z-score Standardization to Identify Outliers
  - Outliers are Z-score Standardization values either less than -3, or greater than 3
  - Values much beyond range [-3, 3] require further investigation to determine their validity
    - Should not automatically omit outliers from analysis
  - For example, on the vehicle weight dataset:
    - Vehicle with minimum weight, 1613 pounds: Z-score = -1.63
    - Vehicle with maximum weight, 4997 pounds: Z-score = 2.34
    - Neither z-score is outside the [-3, 3] range, conclude no outliers among vehicle weights
  - However, Mean and Standard Deviation are both sensitive to the presence of outliers
    - μ and σ are both part of the formula for z-score standardization
    - If an outlier is added or deleted from the dataset, μ and σ will be affected
- When selecting a method for evaluating outliers, should not use measures which are themselves sensitive to outliers

Numerical Methods for Identifying Outliers

- Using Interquartile Range (IQR) to Identify Outliers
  - Robust statistical method and less sensitive to presence of outliers
  - Data divided into four quartiles, each containing 25% of data
    - First quartile (Q1) 25th percentile
    - Second quartile (Q2) 50th percentile (median)
    - Third quartile (Q3) 75th percentile
    - Fourth quartile (Q4) 100th percentile
  - IQR is measure of variability in data

Numerical Methods for Identifying Outliers

- IQR = Q3 – Q1 and represents spread of middle 50% of the data
- Data value defined as outlier if located:
  - 1.5 x (IQR) or more below Q1; or
  - 1.5 x (IQR) or more above Q3
- For example, set of test scores have 25th percentile (Q1) = 70, and 75th percentile (Q3) = 80
- 50% of test scores fall between 70 and 80 and Interquartile Range (IQR) = 80 – 70 = 10
- Test scores are identified as outliers if:
  - Lower than Q1 – 1.5 x (IQR) = 70 – 1.5(10) = 55; or
  - Higher than Q3 + 1.5 x (IQR) = 80 + 1.5(10) = 95

Flag Variables

- Some numerical methods require predictor to be numeric
  - Regression requires recoding categorical variable into one or more flag variables
- Flag variables (aka dummy or indicator variable) is a categorical variable with onle two values: 0 or 1
- Example: Categorical variable sex can be converted as:

  If sex = female, then sex_flag = 0;

  If sex = male, then sex_flag = 1
- If category has $k \geq 3$ possible values, then define $k - 1$ dummy variables
  - The unassigned category (the one for which no flag is created) is taken as the *reference category*

Flag Variables

- For example, for a variable region having $k = 4$ possible values {north, east, south, west
- Define the following $k - 1 = 3$ flag variables

Flag name	IF region=	then	otherwise
north_flag	north	north_flag=1	north_flag=0
east_flag	east	east_flag=1	east_flag=0
south_flag	south	south_flag=1	south_flag=0

- Variable for west is not needed, since $region = west$ is identified when all three flag variables are zero (0).
  - Inclusion of fourth flag variables will cause some algorithms to fail because of the singularity of the matrix regression, for instance.
  - Unassigned category becomes the reference category
  - For example: if in a regression the coefficient for north_flag equals $1000, then the estimated income for region = north is $1000 greater than for region = west when all other predictors are held constant

Transforming Categorical Variables into Numerical Variables

- Why not transforming the categorical variable region into a single numerical variable? For example:

Region	Region_num
North	1
East	2
South	3
West	4

- This is a common and hazardous error. The algorithm now assumes that:
  - The four regions are ordered
  - West > South > East > North
  - West is three times closer to South compared to north, etc.
- This practice should be avoided, except with categorical variables that are clearly ordered, such as with a variable *survey_response* with values *always, usually, sometimes, never*
- Still, careful consideration should be given to the actual values. Should *never, sometimes, usually, always* be numbered as:
  - 1, 2, 3 and 4; or 0, 1, 2 3, since 0 actually means never
  - But what if there relative distance between categorical values is not constant?

Binning Numerical Variables

- Some algorithms require categorical predictors
- Continuous predictors are partitioned as bins or bands
  - Example: *House value* numerical variable partitioned into: *low, medium or high*
- Four common methods:

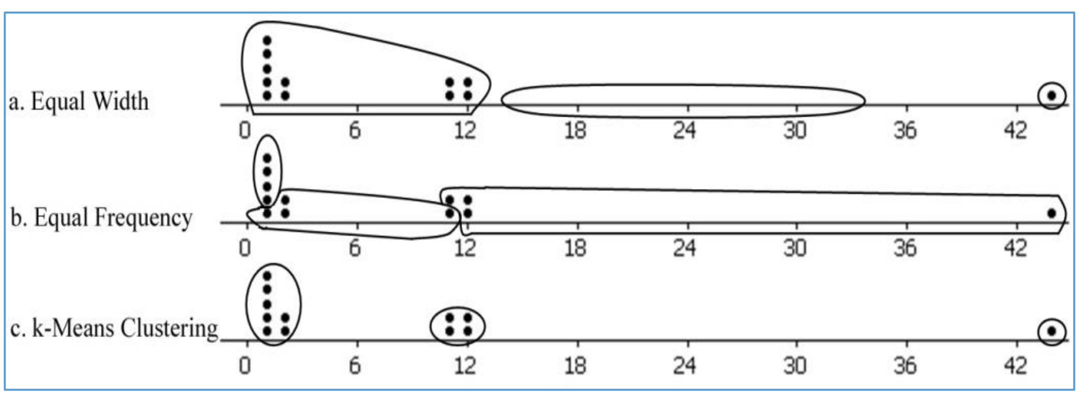
Method	Description	Notes
1. Equal width binning	Divides predictor into k categories of equal width, where k is chosen by client/analyst	Not recommended, since width of bins can be affected by presence of outliers
2. Equal frequency binning	Divides predictor into *k* categories, each having *k/n* records, where *n* is the total number of records	Assumes that each category is equally likely, which is not warranted
3. Binning by clustering	Uses clustering algorithm, like *k-means clustering* (Chapter 10) to automatically calculate “optimal” partitioning	Methods 3 and 4 are preferred
4. Binning based on predictive value	Methods 1 to 3 ignore the target variable; this method partitions numerical predictor based on the effect each partition has on the value of the target variable (see Chapter 3)	

Binning Numerical Variables

- Example: Discretize *X = {1,1,1,1,1,2,2,11,11,12,12,44}* into *k=3* categories

Method	Low	Medium	High
a. Equal Width	0 ≤ X < 15 Contains all values except one	15 ≤ X < 30 Contains no data	30 ≤ X < 45 Contains single outlier
b. Equal Frequency	First four data values {1,1,1,1}	Next four data values {1,2,2,11}	Last four data values {11,12,12,44}
c. k-means Clustering	{1,1,1,1,2,2}	11,11,12,12	{44}

- How is that in Equal Frequency, values {1,1,1,1,1} are split into two categories? Equal values should belong to the same category
- As illustrated in image below, k-means clustering identifies apparently intuitive partitions



Reclassifying categorical variables

- Equivalent of binning numerical variables
- Algorithms like Logistic Regression and C4.5 decision tree are suboptimal with too many categorical values
- Used to reduce the number of values in a categorical field
- Example:
  - Variable *state* {50 values} → Variable *region* {Northeast, Southeast, NorthCentral, Southwest, West}
  - Instead of 50 values, analyst/algorithm handle only 5 values
  - Alternatively, could convert *state* into *economic_level*, with values {richer states, midrange states, poorer states}
- Data analyst should select reclassification that fir business/research problem

Adding and index field

- Adding Index field is recommended
- Tracks the sort order of the records in the database
- Data mining data is partitioned at least once
  - Index helps to rebuild dataset in original order
- In IBM/SPSS Modeler, use @Index function in Derive node to create an index field

Removing variables that are not useful

- Some variables will not help the analysis
  - Unary variables – Take only a single value (a constant).
    - Example – In an all-girls private school, variable sex will always be femaly, thus not having any effect in the data mining algorithm
  - Variables which are very nearly unary – Some algorithms will treat these as unary. Analyst should consider whether removing.
    - Example - In a team with 99.9% females and 0.05% males, the variable sex is nearly unary.

Variables that should probably not be removed

Variables with 90% or more missing values

- Consider that there may be a pattern in missingness
- Inputation becomes challenging
- Example: Variable donation_dollars in self-reported survey
    - Top 10% donors might report donations, while others do not – the 10% is not representative
    - Preferable to construct a flag variable, *donation_flag*, since missingness might have predictive power
    - If there is reason to believe that 10% is representative, then proceed to imputation using regression or decision tree (chapter 13)

Variables that should probably not be removed

Strongly correlated variables

- Important information might be discarded when removing correlated variables
- Example: Variables *precipitation* and '*attendance at the beach*' are negatively correlated
    - This might double-count an aspect of the analysis or cause instability in model results – prompting analyst to remove one variable
    - Should perform Principal Component analysis instead, to convert into a set of uncorrelated principal components

Removal of duplicate records

- Records might have been inadvertly copied, creating duplicates
  - Duplicate records lead to overweighting of their data values – therefore, they should be removed
- Example – If ID field is duplicated, then remove it
- But, consider genuine duplicates
  - When the number of records is higher than all possible combination of field values, there will be genuine duplicates

A word about ID fields

- ID fields have different value for each record
- Might be hurtfull, with algorithm finding spurious relationships between ID field and target
- Recommendation: Filter ID fields from data mining algorithm, but do not remove them from the data, so that analyst can still differentiate the records