Course Introduction, Descriptive Statistics and R

GENOME 560, Spring 2018

Doug Fowler, GS (dfowler@uw.edu)

Your Instructors

Doug Fowler

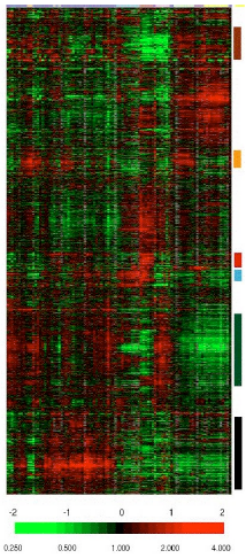
Assistant Professor of Genome Sciences
Adjunct Assistant Professor of Bioengineering

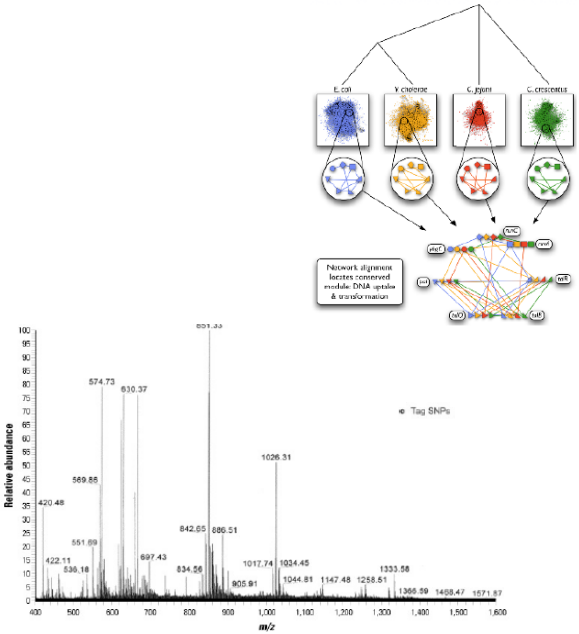
Su-in Lee

Associate Professor of Computer Science & Engineering
Associate Professor of Genome Sciences
Adjunct Associate Professor of Electrical Engineering

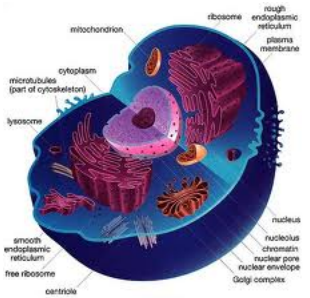
Why Take This Course?

- *Data* are interesting because they help us understand the world
- *Genomics*: massive amounts of data …
- Statistics is fundamental to the **design, analysis** and **interpretation** of experiments





What does it all mean?



Why Take This Course?

- *Data* are interesting because they help us understand the world
- *Genomics*: massive amounts of data …
- Statistics is fundamental in genomics because it is integral in the **design, analysis** and **interpretation** of experiments
- This course covers the key statistical concepts and methods necessary for extracting biological insights from experimental data

Learning Goals

- 10 weeks is too short to cover all of statistics or even every specific topic that might arise in the course of your research…

Learning Goals

- 10 weeks is too short to cover all of statistics or even every specific topic that might arise in the course of your research…

- Statistical and computational methods should never be treated as “recipes” to follow!

Learning Goals

- 10 weeks is too short to cover all of statistics or even every specific topic that might arise in the course of your research…

- Statistical and computational methods should never be treated as "recipes" to follow!

- Instead, we should focus on
  - rigorous understanding of fundamental concepts that will provide you with the tools necessary to address routine statistical analyses
  - foundation to understand and learn mode specific topics

Course Schedule

- **Syllabus:**

Date	Topic
Week 1	Introduction to probability, random variables and probability distributions, descriptive statistics, joint and conditional probability
Week 2	More probability distributions, introduction to hypothesis testing
Week 3	Parametric hypothesis testing; comparing means, comparing proportions
Week 4	Non-parametric hypothesis testing; comparing means, comparing proportions; rank-based tests; permutation testing
Week 5	More on permutation testing; resampling methods; sample size calculations
Week 6	Parameter estimation, maximum likelihood methods, Bayesian methods
Week 7	Single factor ANOVA, two-way ANOVA; multiple hypothesis testing
Week 8	FDR correction; linear regression; feature selection
Week 9	Cross-validation; logistic regression
Week 10	Model building techniques; Bayesian networks

- **Grading:** 10 problem sets (90%), participation (10%)

Books and Resources

- Course website (https://sites.google.com/a/cs.washington.edu/genome560-spr18/home)
- No required text
- Good on-line resources
  - https://www.khanacademy.org/math/probability
  - http://www.statsoft.com/textbook/stathome.html
  - http://www.stat.berkeley.edu/~stark/SticiGui/Text/toc.htm
- Some good books if you ever have some extra $$$:
  - Probability and Statistics for Engineering and the Scientists 6th Ed. Jay L. Devore (2004). Duxbury press, Thompson-Brooks/Cole.
  - All of Statistics Larry Wasserman (2004) Springer Science

Courses at UW

2014

GNOM 560 - Statistics for Genome Scientists

2015

GNOM 560 - Statistics for Genome Scientists

GNOM 373 - Genome Informatics

2016

GNOM 560 - Statistics for Genome Scientists

GNOM 373 - Genome Informatics

genome560-spr16

Home | Course Materials | Homework | People

Contents

Introduction to Probability and Statistics

Week 1: Introduction to probability, random variables and probability distributions, descriptive statistics, joint and conditional probability (lecture 1 [PDF, R], lecture 2 [PDF, R])

Week 2: More probability distributions; introduction to hypothesis testing (lecture 3 [PDF, R], lecture 4 [PDF, R])

Basic Hypothesis Testing

Week 3: Parametric hypothesis testing; comparing means, comparing proportions; rank-based tests; permutation testing (lecture 5 [PDF, R], lecture 6 [PDF, R])

Week 4: Non-parametric hypothesis testing; comparing means, comparing proportions; rank-based tests; permutation testing (lecture 7 [PDF, R], lecture 8 [PDF, R])

Advanced Hypothesis Testing

Week 5: More on permutation testing; resampling methods; sample size calculations (lecture 9 [PDF, R], lecture 10 [PDF, R])

Week 6: One-way ANOVA; basics on the degrees of freedom; two-way ANOVA (lecture 11 [PDF, R], lecture 12 [PDF, R])
- One way ANOVA [PDF]
- Two-way ANOVA [PDF] [video]

Week 7: Two-way ANOVA; interaction; multiple hypothesis testing (lecture 13 [PDF, R], lecture 14 [PDF, R])

Modeling

Week 8: Multiple hypothesis testing; parameter estimation; maximum likelihood estimation (MLE) (lecture 15 [PDF, R], lecture 16 [PDF, R])

Week 9: Bayesian estimation; linear regression; L2 regularization (lecture 17 [PDF, R], lecture 18 [PDF, R, R tutorial (Package 'MASS')])

Week 10: High-dimensionality; feature selection; L1 regularization; cross validation tests; model selection; clustering methods (lecture 19 [PDF, R], lecture 20 [PDF, R tutorial (Package 'glmnet')])

Class Meetings

- Class meets twice a week
  - Tue/Thur 9-10:20am @ Foege S110
- Each class will last for 80 minutes and be a mix of lecture and R tutorial/exploration
- Other forms of learning and interactions will be included
- We will work on problems in small groups as well as work through statistical analyses using R (please bring a laptop with R installed!)

Homework and grading

- Problem sets will be posted on Thursday and due the following Thursday before class. **No credit will be given for late problem sets.**

- Find problem sets and turn in answers on the course Canvas site (accessible from the course website)
  - https://sites.google.com/a/cs.washington.edu/genome560-spr18/home?pli=1

- 90% of your grade will be from the problem sets, the other 10% is based on in-class participation

Questions about the mechanics?

Lecture 1: Descriptive Statistics and Data Visualization

Outline

- What is descriptive statistics and exploratory data analysis?
- Basic numerical summaries of data
- Basic operations in R

Why do we collect data?

Why do we collect data?

- To describe characteristics of a population

- To make inferences about a population

Why Descriptive/Graphical Summary?

- Before making inferences from data, it is essential to understand its basic structure

- Why?

- To listen to the data:
  - to catch mistakes
  - to see patterns in the data
  - to find violations of statistical assumptions
  - to generate hypotheses
  - … and because if you don't, you will have trouble later!

Types of Data

- What kinds of data are there?

Types of Data

- Categorical
  - Binary: 2 categories (e.g. yes, no)
  - Nominal: more categories (e.g. red, white or blue)
  - Ordinal: order matters (e.g. low, medium, high)
  - E.g. gender, ethnicity, disease state, genotypes, etc

- Quantitative
  - Discrete (e.g. outcomes of a die roll)
  - Continuous (e.g. height)
  - E.g. fluorescence intensity, expression level, etc

Dimensionality of Data Sets

- **Univariate:** Measurement made on one variable per subject

- **Bivariate:** Measurement made on two variables per subject

- **Multivariate:** Measurement made on many variables per subject

Using statistics to describe data

Using statistics to describe data

- **What is a statistic?**

Using statistics to describe data

- **What is a statistic?** A quantity that is calculated from a sample of data.

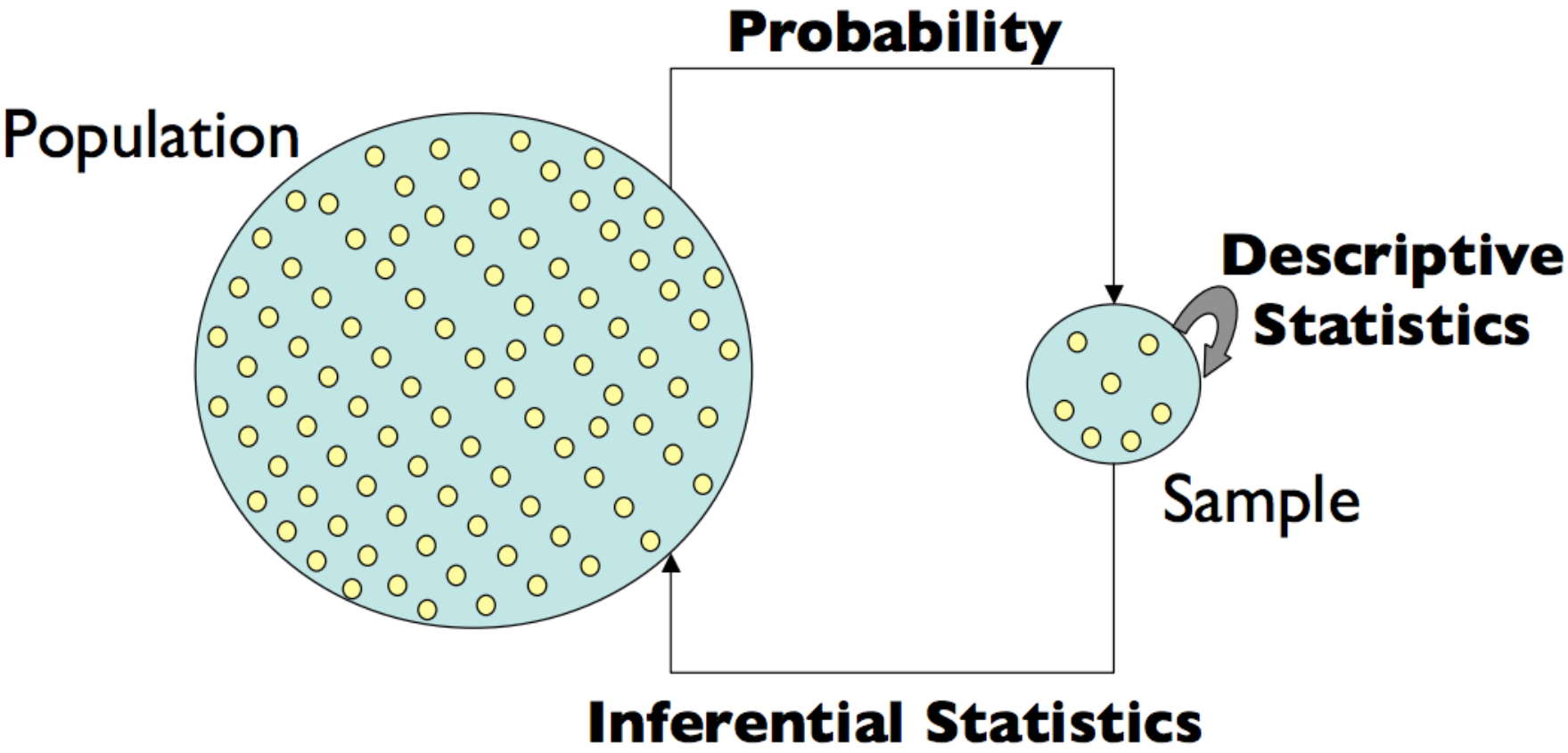
- More formally: a function of a sample that is independent of the sample's distribution (e.g. function can be written down before data is collected)

Using statistics to describe data

- **What is a statistic?** A quantity that is calculated from a sample of data.

- Useful statistics describe something of interest about the sample, and can allow us to estimate the something about the population.

Central Dogma of Statistics



Basic Descriptive Statistics

- ***Central tendency measures.*** They are computed to give a "center" around which the measurements in the data are distributed.

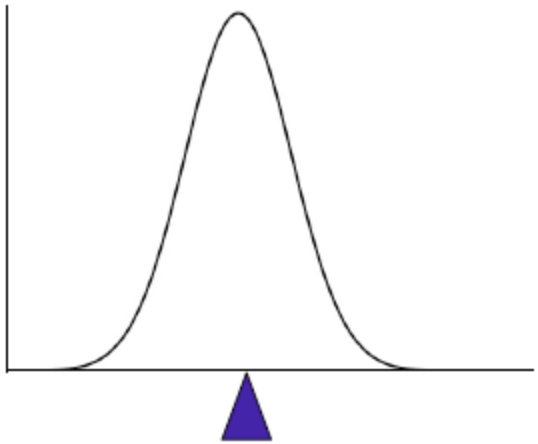
- ***Variation or variability measures.*** They describe "data spread" or how far away the measurements are from the center.

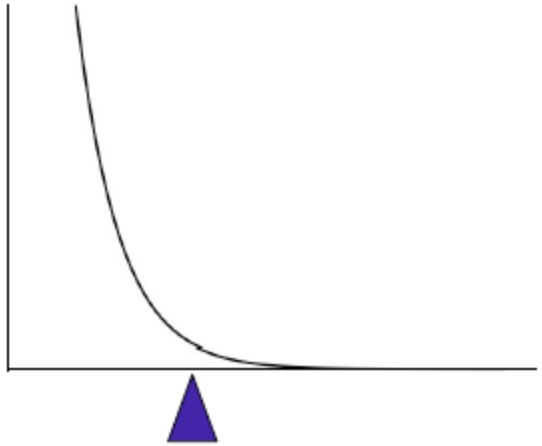
- ***Relative standing measures.*** They describe the relative position of specific measurements in the data

Central Tendency Measures: Mean

- To calculate the **mean** of a set of observations, add their value and divide by the number of observations:

$$\bar{x} = \frac{x_1 + x_2 + x_3 + \ldots + x_n}{n} = \frac{1}{n}\sum_{i=1}^{n} x_i$$





Central Tendency Measures: Mean

$$\bar{x} = \frac{x_1 + x_2 + x_3 + \ldots + x_n}{n} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$sample = (4, -3, 9, -7, 1)$$

Central Tendency Measures: Mean

$$\bar{x} = \frac{x_1 + x_2 + x_3 + \ldots + x_n}{n} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$sample = (4, -3, 9, -7, 1)$$

$$\bar{x} = \frac{4 + -3 + 9 + -7 + 1}{5}$$

$$\bar{x} = 0.8$$

Central Tendency Measures: Median

- **Median:** the geographic middle value

- **Calculation:**
  - If there are an odd number of observations, find the middle value
  - If there are an even number of observations, find the middle two values and take the mean

- **Example:**

  Some data:

  Age of participants: 17 19 21 22 23 23 23 38

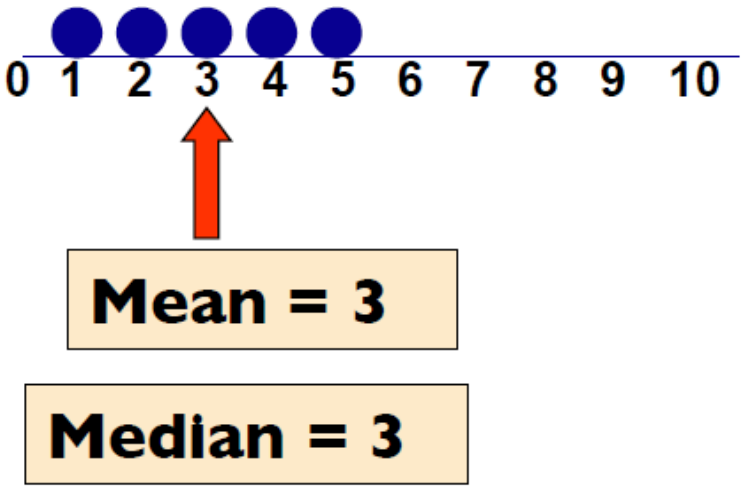
  Median = (22+23)/2 = 22.5

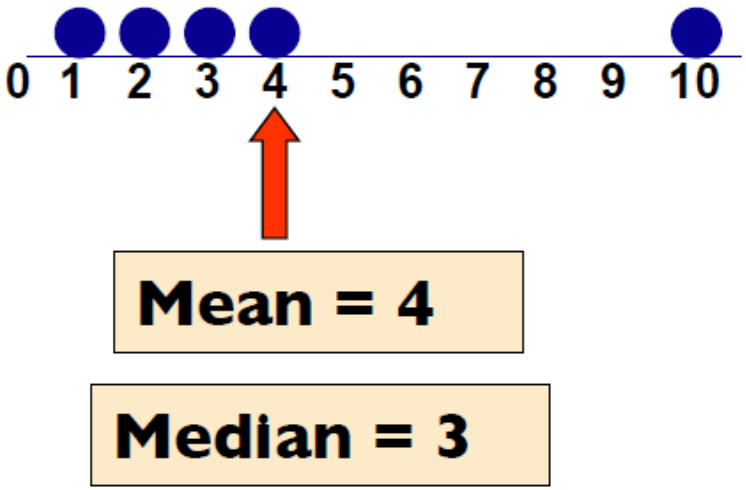
Which Measure Is Best?

Which Measure Is Best?

- **Mean** is useful for symmetric distributions without outliers

- **Median** is useful for skewed distributions or data with outliers





Variability Measures: Variance

- Average of squared deviation of values from the mean

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

Variability Measures: Variance

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

$$sample = (4, -3, 9, -7, 1)$$

Variability Measures: Variance

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

$$sample = (4, -3, 9, -7, 1)$$

$$s^2 = \frac{(4-0.8)^2 + (-3-0.8)^2 + (9-0.8)^2 + (-7-0.8)^2 + (1-0.8)^2}{n-1}$$

$$s^2 = \frac{152.8}{4}$$

$$s^2 = 38.2$$

Variability Measures: Variance

- Variance is somewhat arbitrary
- What does it mean to have a variance of 10.8? Or 2.2? Or 1459.092? Or 0.000001?
- Intuitively, not much. But if you transform the variance so that it has the same units as the samples, it takes on intuitive meaning

Variability Measures: Standard Deviation

- Most commonly used measure of variation
- Is the square root of the variance
- Has the same units as the original data

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

Variability Measures: Range

$$range = x_{maximum} - x_{minimum}$$

Variability Measures: Range

$$range = x_{maximum} - x_{minimum}$$

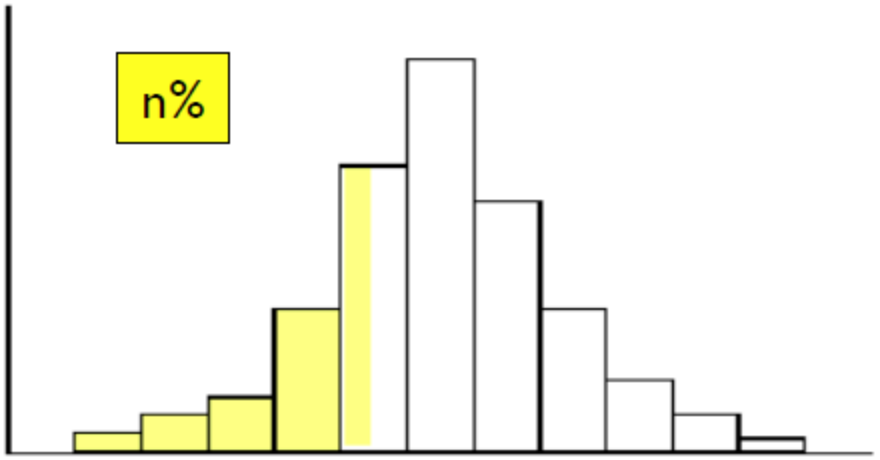
$$sample = (4, -3, 9, -7, 1)$$

$$range = 9 - -7$$

$$range = 16$$

Relative Standing: Percentiles (aka Quantiles)

- In general the n^{th} percentile is a value such that n% of the observations fall at or below of it

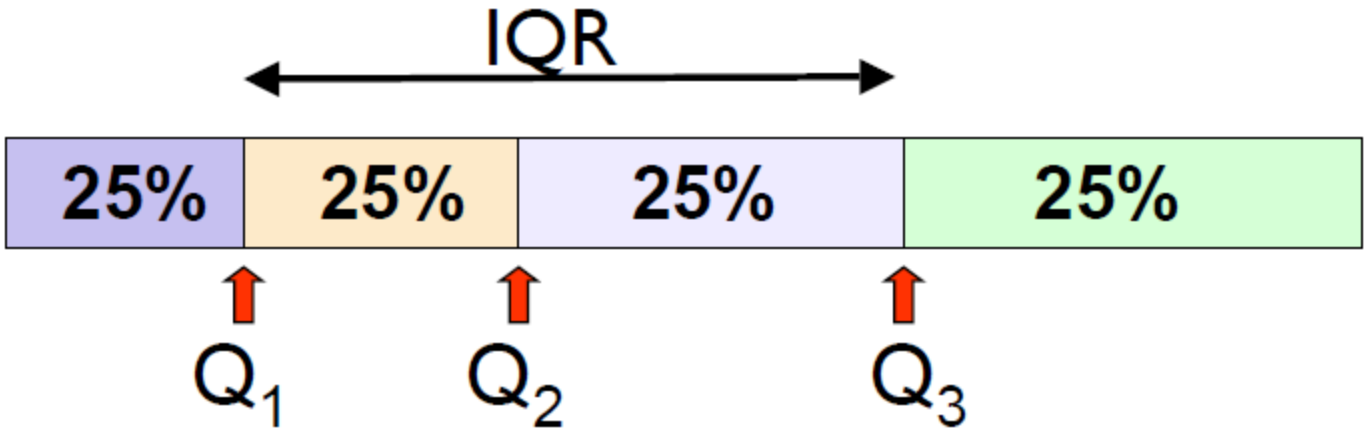


$Q_1 = 25^{th}$ percentile

Median = 50^{th} percentile

$Q_3 = 75^{th}$ percentile

Relative Standing: Quartiles and IQR



- The first quartile, Q_1, is the value for which 25% of the observations are smaller and 75% are larger
- Q_2 is the same as the median (50% are smaller, 50% are larger)
- IQR = “middle fifty”

What is R?

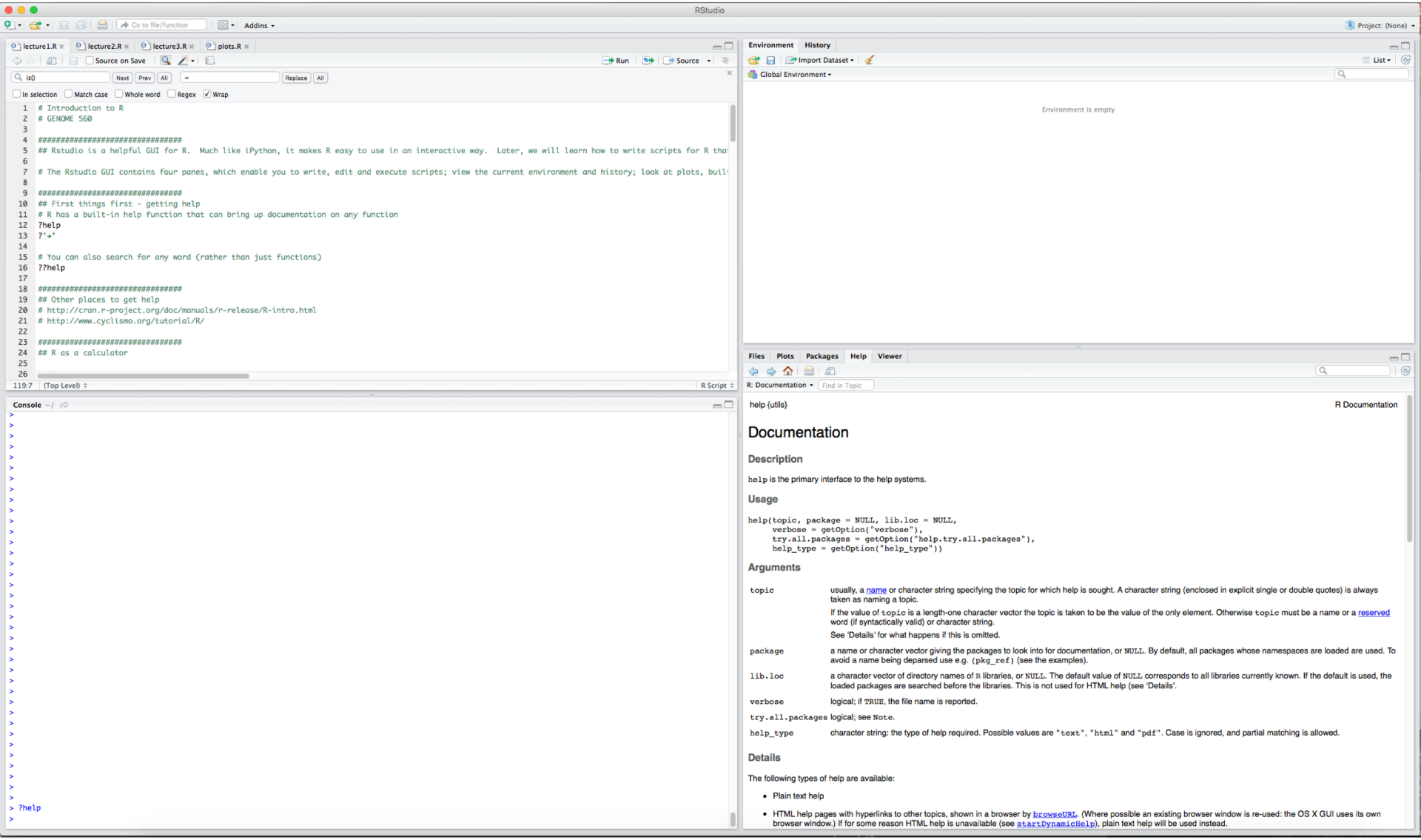
- The R statistical programming language is a free open source package based on the S language developed by Bell Labs
- Many statistical functions are already built in
- Contributed packages expand the functionality to cutting edge research
- Amazing graphics (ahem)
- Widely used in genetics, genomics, computational biology

What is R?

How is R different from C/Python?

- Getting stuff done in R
  - R is designed primarily to be interactive (though you can write scripts)
- Storing and manipulating data
  - R has some different data types
- Taking action
  - Functions in R work pretty much the same way as in other langs
- Repeating something
  - R is designed for parallel rather than iterative operations

My favorite R interactive tool: Rstudio



How is R different from C/Python?

- Getting stuff done in R
  - R is designed primarily to be interactive (though you can write scripts)
- Storing and manipulating data
  - R has some different data types
- Taking action
  - Functions in R work pretty much the same way as in other langs
- Repeating something
  - R is designed for parallel rather than iterative operations

One big difference: the data frame

- A data frame is a table, or two-dimensional array-like structure
- Each column contains measurements on one variable, and each row contains one case
- Consider an experiment where we've taken 12 individuals and randomized them to three meals, measuring blood sugar levels afterwards:

Doritos	Skittles	Kale
150	200	0
180	190	0
130	210	0

One big difference: the data frame

- A data frame is a table, or two-dimensional array-like structure

- Each column contains measurements on one variable, and each row contains one case

- Consider an experiment where we've taken 12 individuals and randomized them to three meals, measuring blood sugar levels afterwards:

Doritos	Skittles	Kale
150	200	0
180	190	0
130	210	0

One big difference: the data frame

- A data frame is a table, or two-dimensional array-like structure
- Each column contains measurements on one variable, and each row contains one case
- Consider an experiment where we've taken 12 individuals and randomized them to three meals, measuring blood sugar levels afterwards:

Doritos	Skittles	Kale
150	200	0
180	190	0
130	210	0

One big difference: the data frame

- A data frame is a table, or two-dimensional array-like structure
- Each column contains measurements on one variable, and each row contains one case
- Consider an experiment where we've taken 12 individuals and randomized them to eat either Doritos, Skittles, or Kale and measured their blood sugar levels afterwards:

Blood sugar	Meal
150	Doritos
180	Doritos
130	Doritos
200	Skittles
190	Skittles
210	Skittles
0	Kale
0	Kale
0	Kale

How is R different from C/Python?

- Getting stuff done in R
  - R is designed primarily to be interactive (though you can write scripts)
- Storing and manipulating data
  - R has some different data types
- Taking action
  - Functions in R work pretty much the same way as in other langs
- Repeating something
  - R is designed for parallel rather than iterative operations

How is R different from C/Python?

```
my_function = function(param1, param2, …) {
        do some stuff with param1
        do some stuff with param2
        return(stuff)
        }

> my_function(p1, p1)
> stuff
```

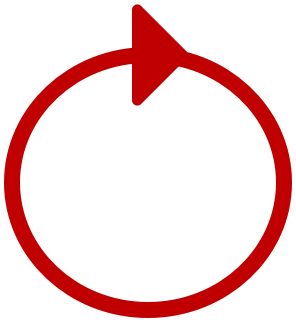
How is R different from C/Python?

- Getting stuff done in R
  - R is designed primarily to be interactive (though you can write scripts)
- Storing and manipulating data
  - R has some different data types
- Taking action
  - Functions in R work pretty much the same way as in other langs
- Repeating something
  - R is designed for parallel rather than iterative operations

How is R different from C/Python?

- Traditionally:

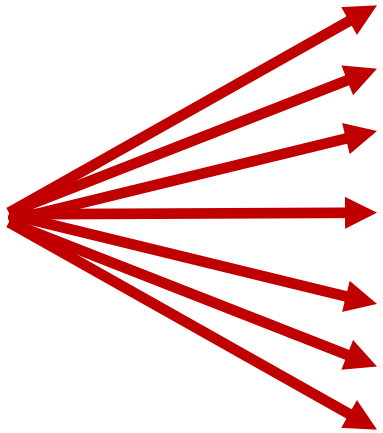
```
for(i in all_my_things){
    do something with i
    }
```



Perform a function i times, one at a time

- In R:

```
apply(all_my_things, do something)
```



Perform a function i times simultaneously

How is R different from C/Python?

R is built and maintained by volunteers. Packages are built and maintained(?) by anybody. Caveat emptor.

R Resources

- Windows, Mac and Linux binaries available at

http://www.r-project.org

- Rstudio is a nice GUI for R

https://www.rstudio.com/

- For a tutorials see:

https://cran.r-project.org/doc/contrib/Torfs+Brauer-Short-R-Intro.pdf

http://cran.r-project.org/doc/manuals/R-intro.html

- A nice R cheat sheet:

http://cran.r-project.org/doc/contrib/Short-refcard.pdf

R tutorial format

- We'll be working either individually or in groups
- Each day, I'll have a text document with information, problems and challenges, posted on the course website
- After class, I will post an updated document with my answers to the problems

Goals of Our R Tutorial Today

- Installing R
- Using R as a fancy calculator
- Data structures: scalars, vectors, data frames, matrices
- Reading in data from a file
- Subsetting and extracting data

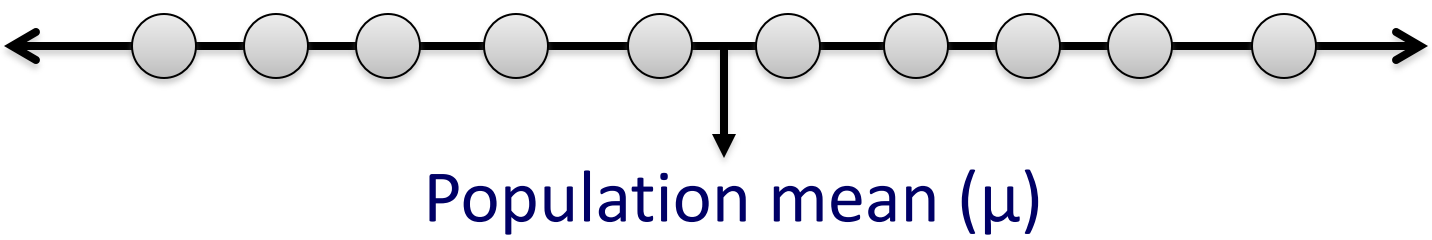
Extra slides

But Wait, Why n-1?

- It makes sense that n-1 yields a larger (more conservative) variance
- Intuitive proof that n underestimates variance:

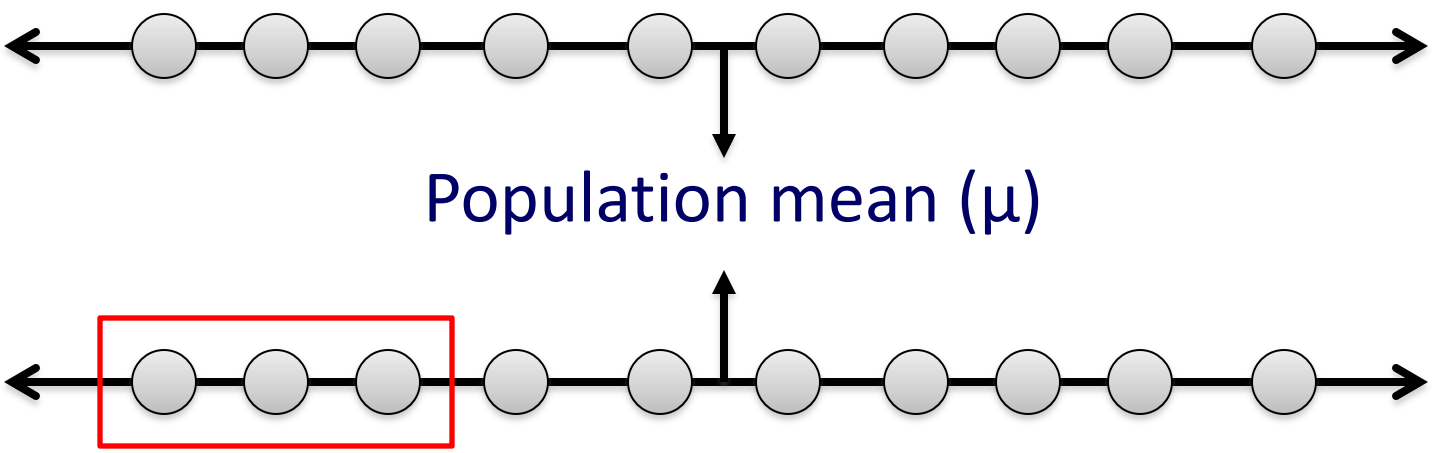
But Wait, Why n-1?

- It makes sense that n-1 yields a larger (more conservative) variance
- Intuitive proof that n underestimates variance:



But Wait, Why n-1?

- It makes sense that n-1 yields a larger (more conservative) variance
- Intuitive proof that n underestimates variance:



Some samples do not contain the population mean and the variance within these samples underestimates the population variance, hence n-1 is a better estimator

Graphical Summaries of Data

- Dimensionality of data matters when thinking about plots/graphs.
  - **Univariate:** Measurement made on one variable per subject
  - **Multivariate:** Measurement made on many variables per subject

Univariate Data

- Histograms, bar plots and box plots
- What is the difference between these?

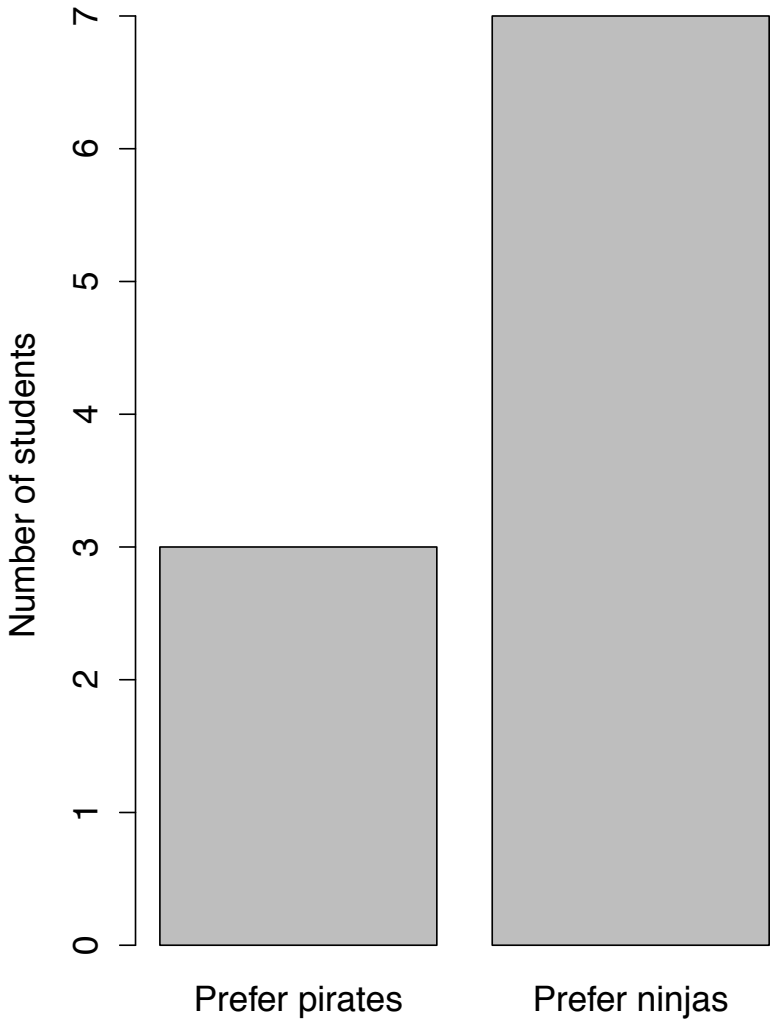
Bar plot:

- Used for categorical variables to show frequency or proportion in each category
- Translate the data from frequency tables into a pictorial presentation…

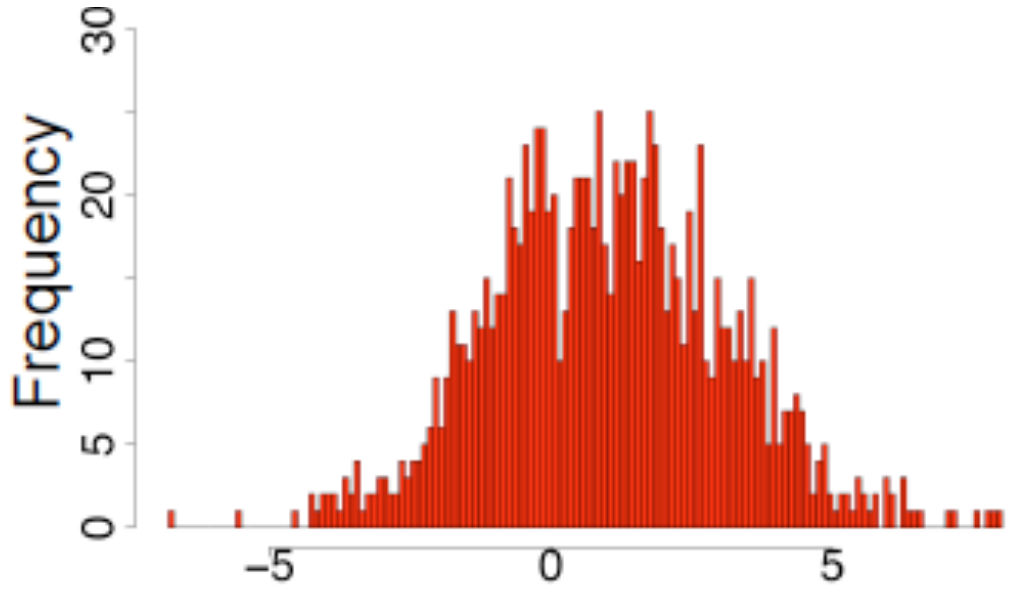
Histogram: Used to visualize distribution (shape, center, range, variation) of continuous variables

Box plot: Visual summary of a histogram

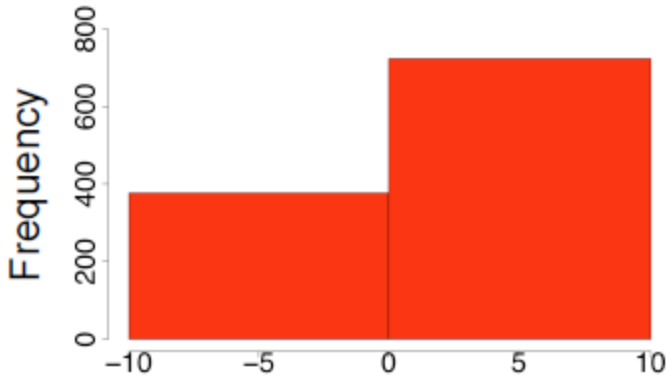
Bar plots

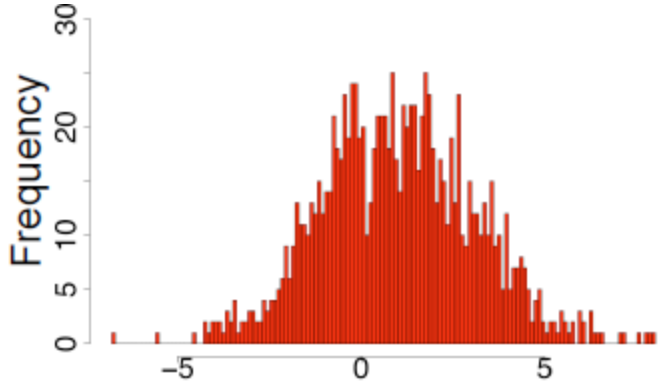


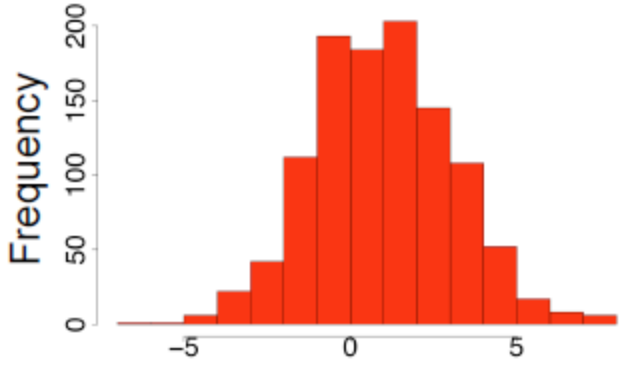
Histograms



Effect of Bin Size on Histogram

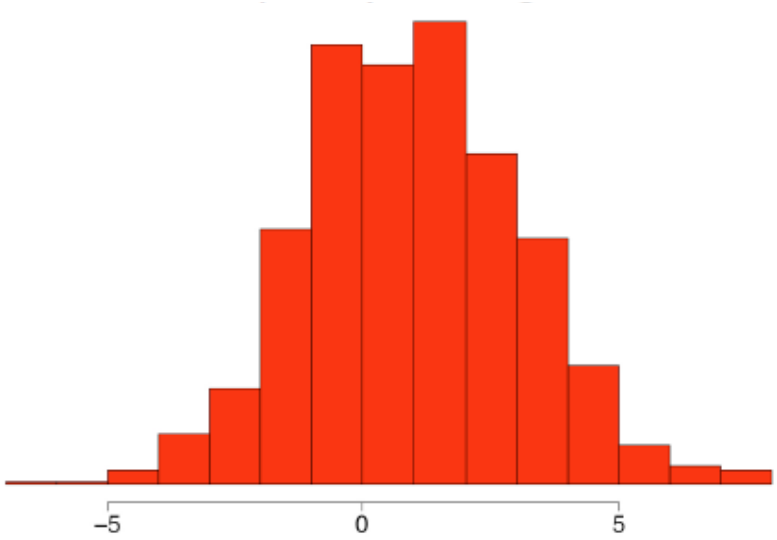


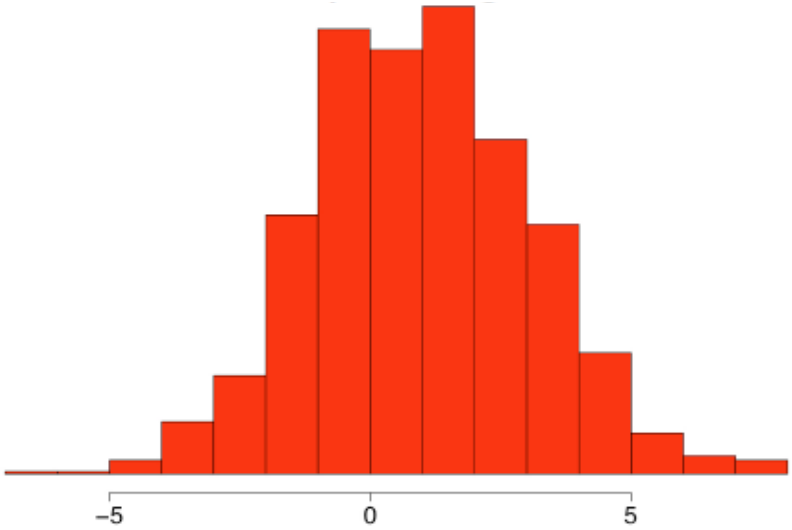




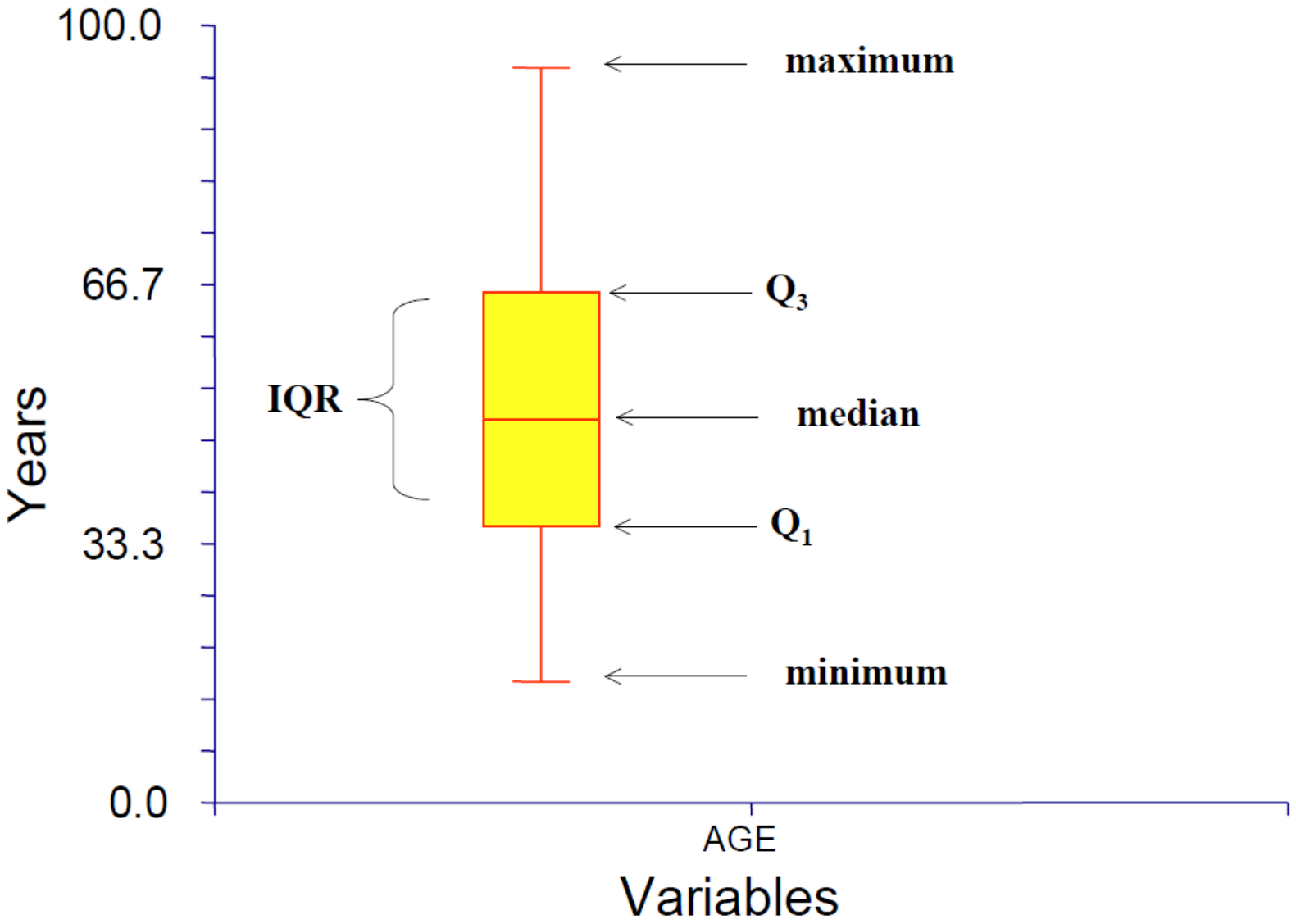
More on Histograms

- What's the difference between a frequency histogram and a density histogram?





Box Plots


100.0
66.7
33.3
0.0
Years
maximum
Q_3
IQR
median
Q_1
minimum
AGE
Variables

Multivariate Data

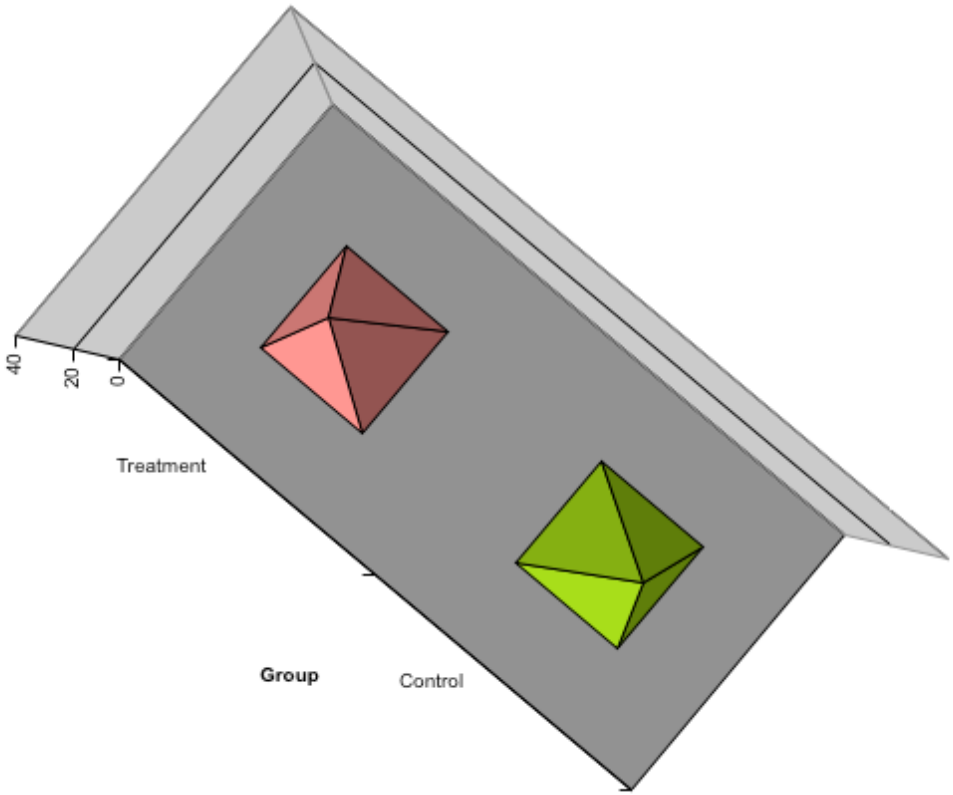
- **Clustering**
  - Organize variables into clusters
  - Descriptive, not inferential
  - Many approaches
  - “Clusters” always produced

- **Data reduction approaches**
  - Reduce n-dimensional dataset into much smaller number
  - Finds a new (smaller) set of variables that retains most of the information in the total sample
  - Effective way to visualize multivariate data

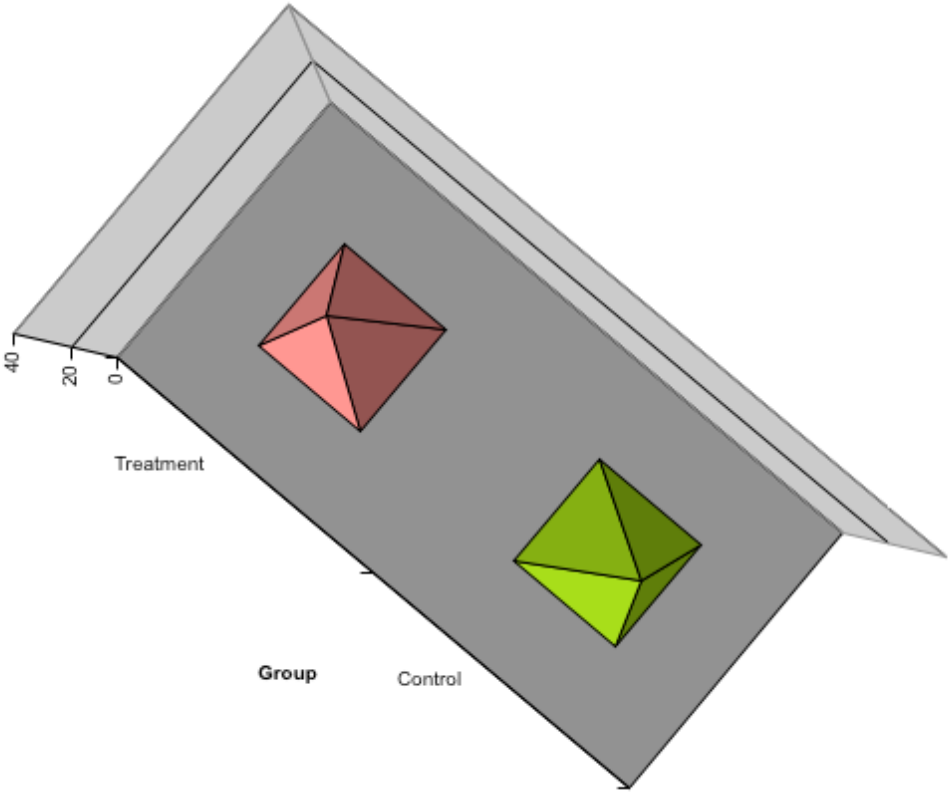
The Bad and Ugly

- **Let's go through some examples of bad graphs…**

Example 1



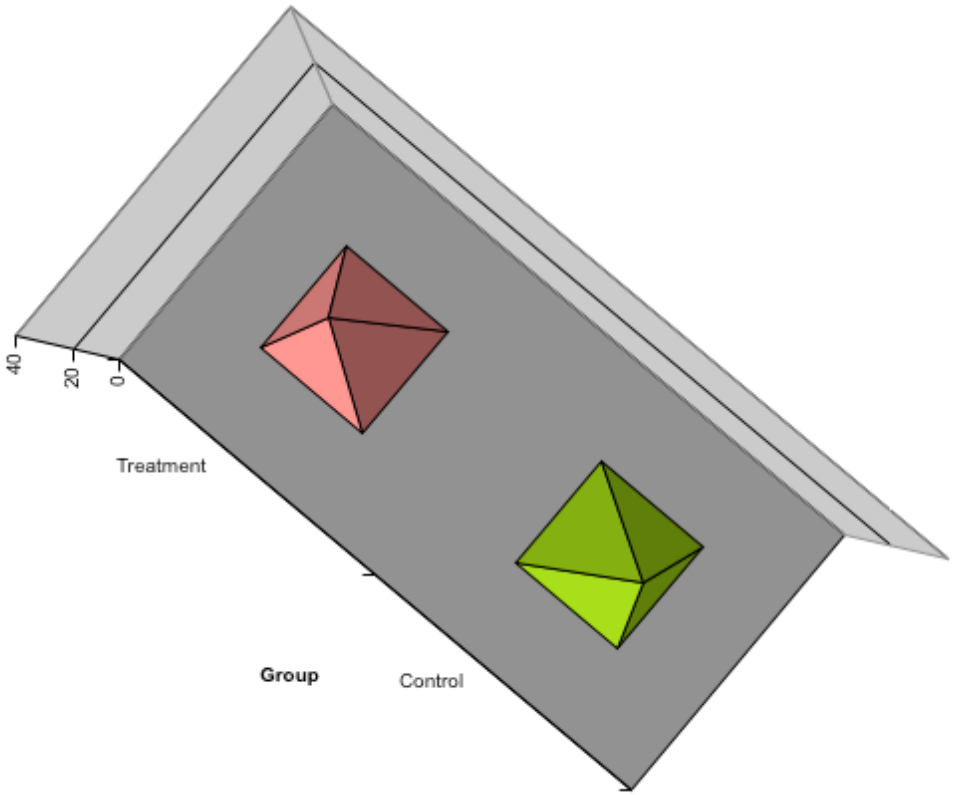
Example 1

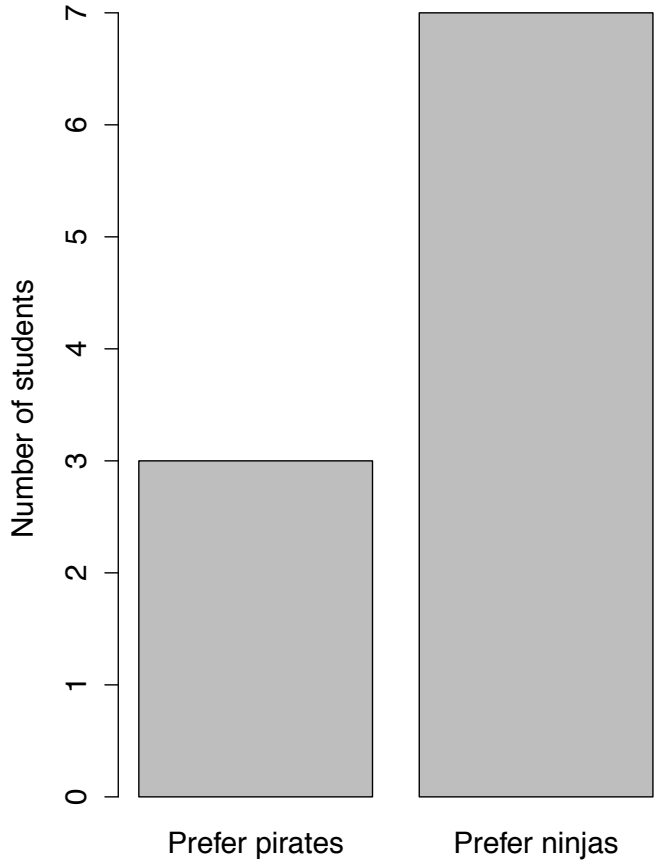


Many unnecessary elements (color, 3D, perspective) obscure a very simple comparison.

Solution:

Example 1

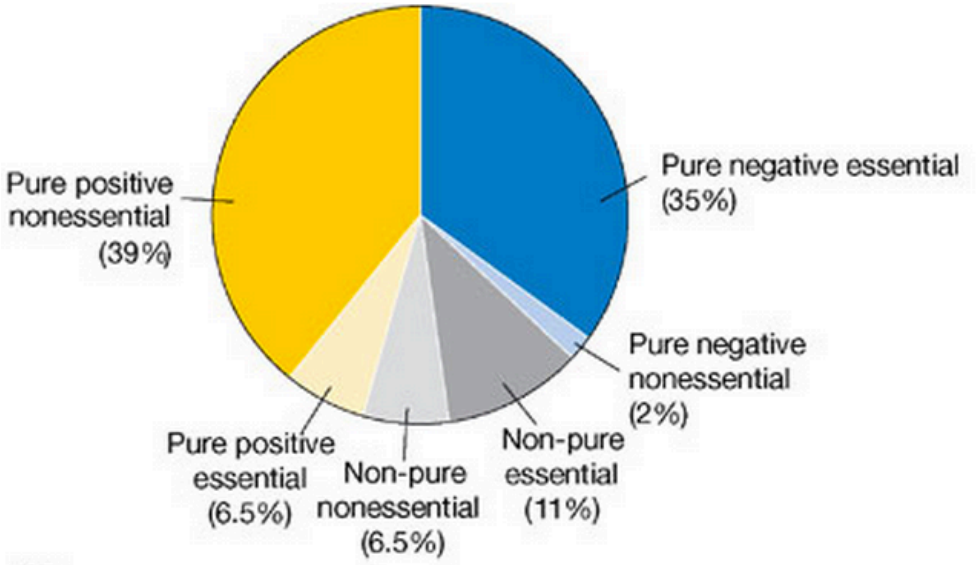




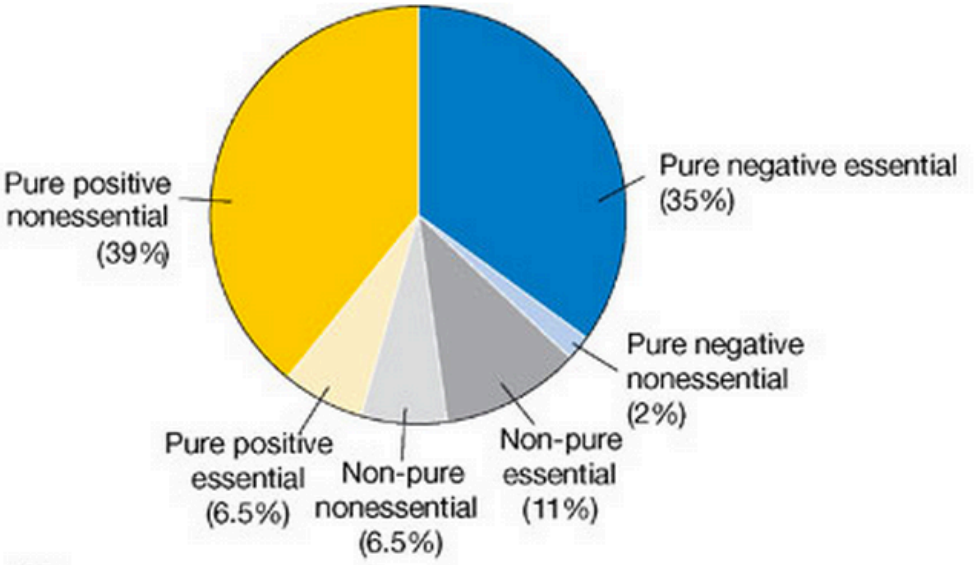
Many unnecessary elements (color, 3D, perspective) obscure a very simple comparison.

Solution: barplot

Example 2


Pure positive nonessential (39%)
Pure negative essential (35%)
Pure negative nonessential (2%)
Non-pure essential (11%)
Non-pure nonessential (6.5%)
Pure positive essential (6.5%)

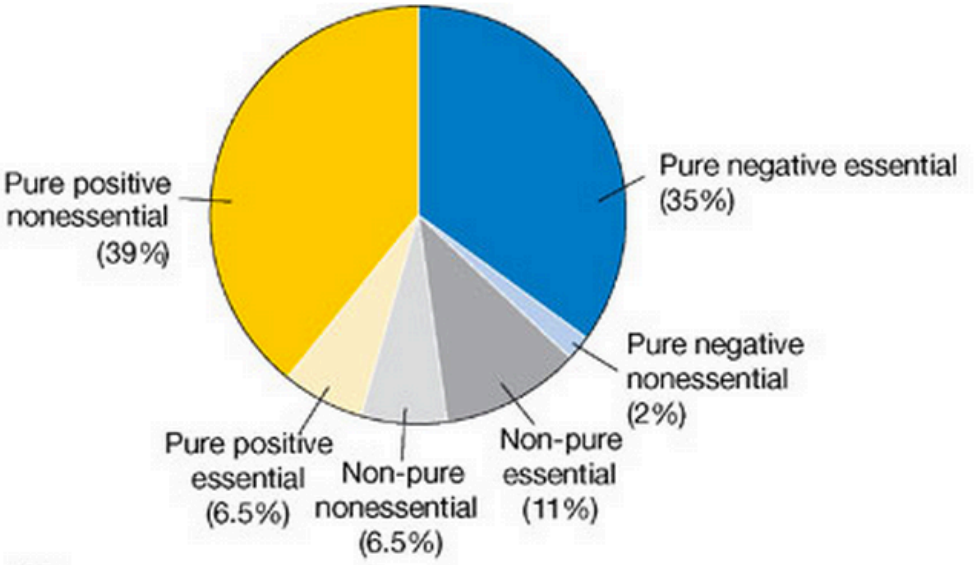
Example 2



Comparing areas/proportions is difficult, the pie chart is not ordered making things harder

Solution:

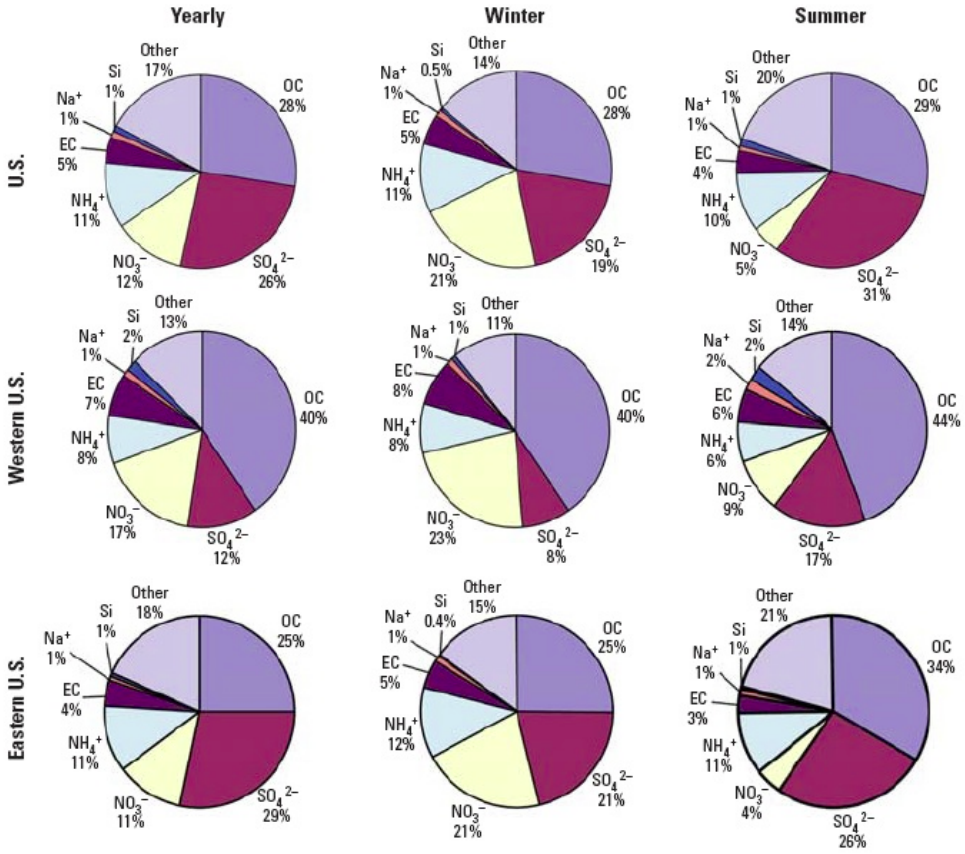
Example 2



Comparing areas/proportions is difficult, the pie chart is not ordered making things harder

Solution: barplot

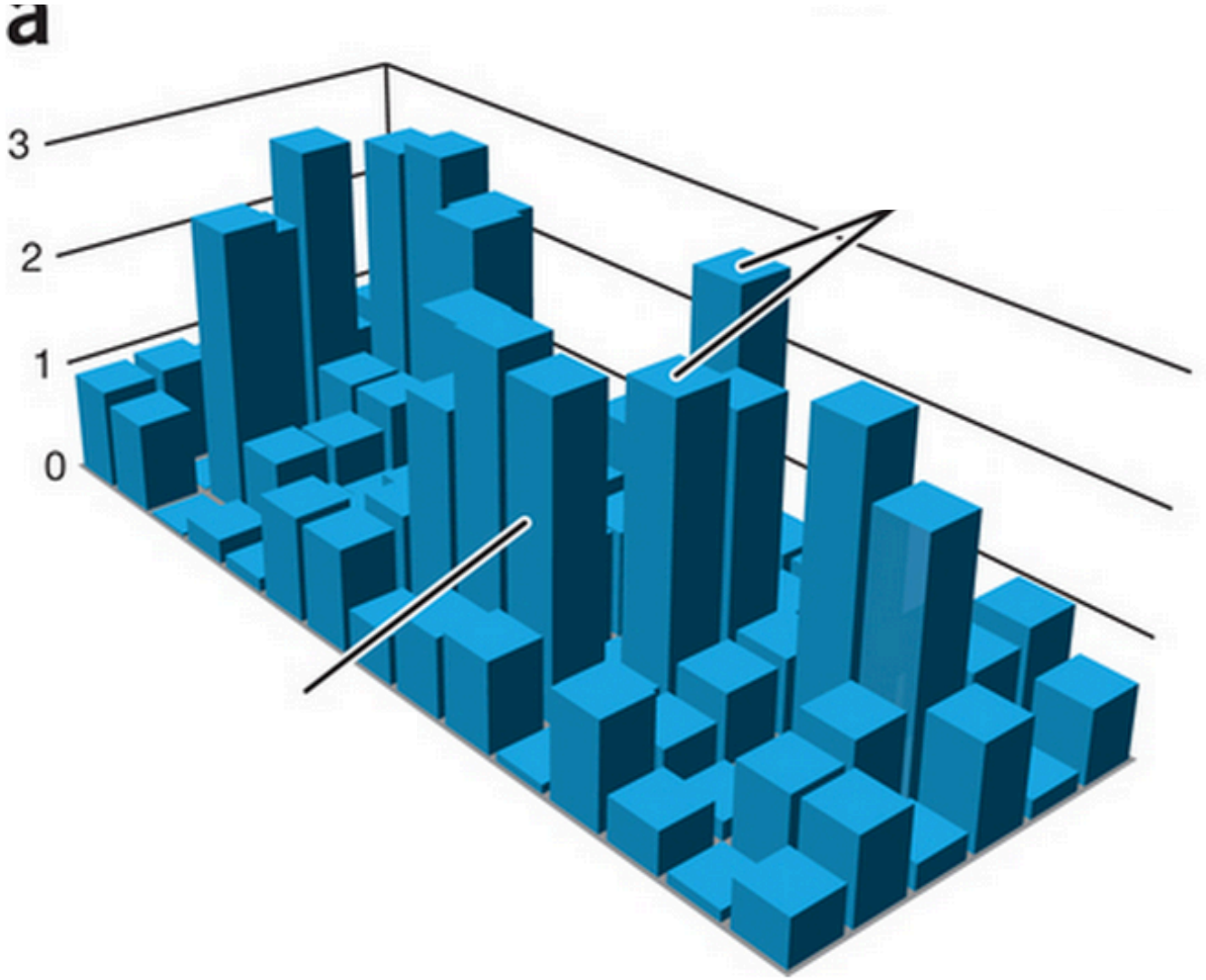
Example 3



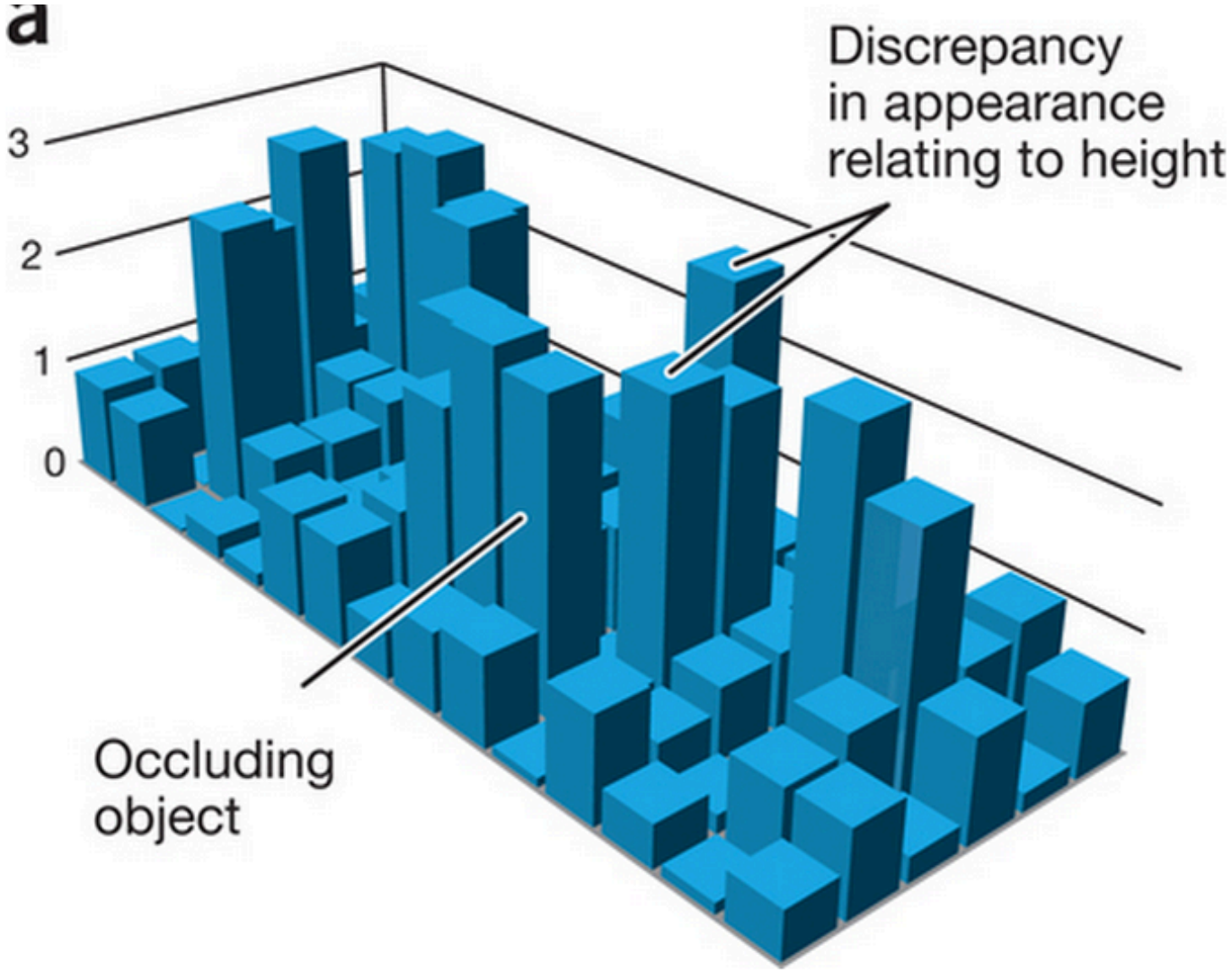
Multiple pie charts just confound the problem, and the labeling is hard to follow

Solution: This is a hard one; a set of heatmaps could work. Plotting relevant trends or doing formal dimensionality reduction are also workable

Example 4

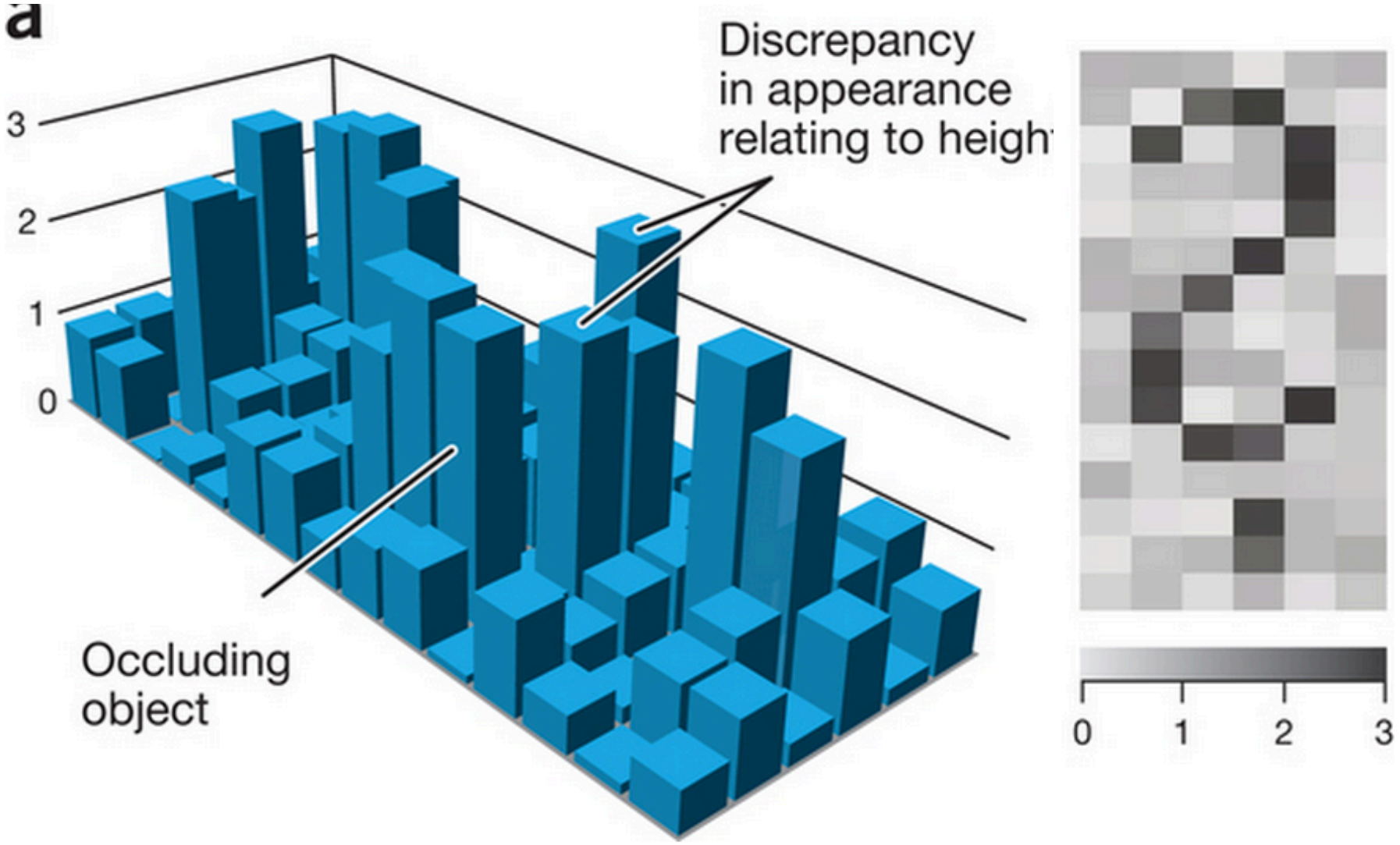


Example 4

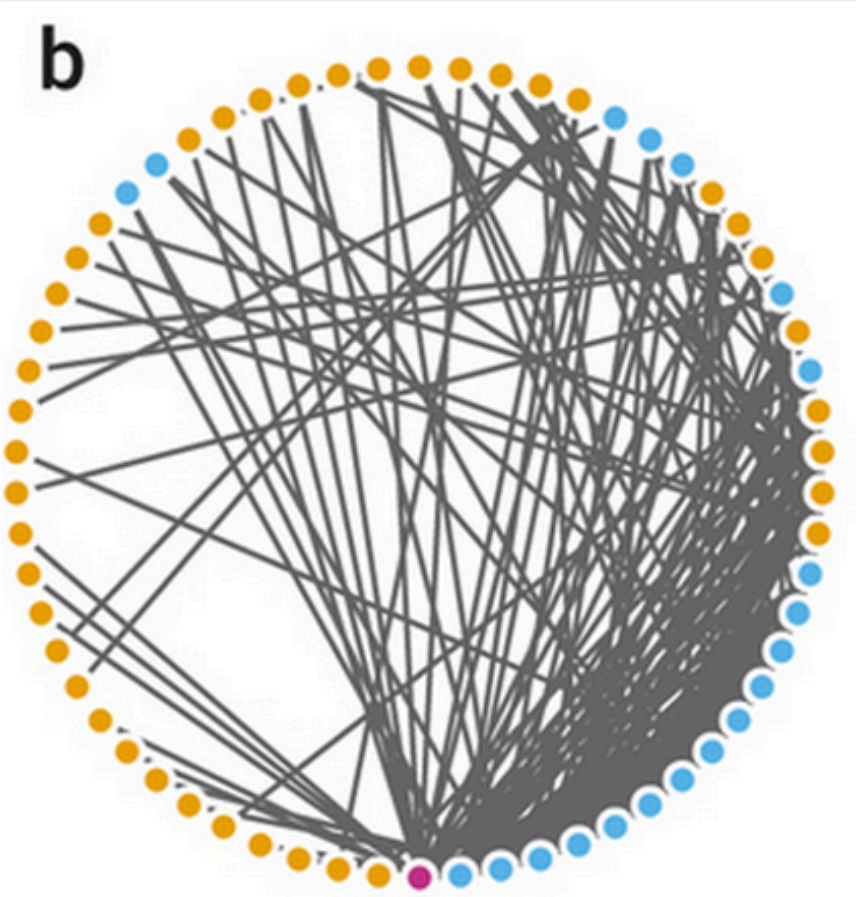


Solution?

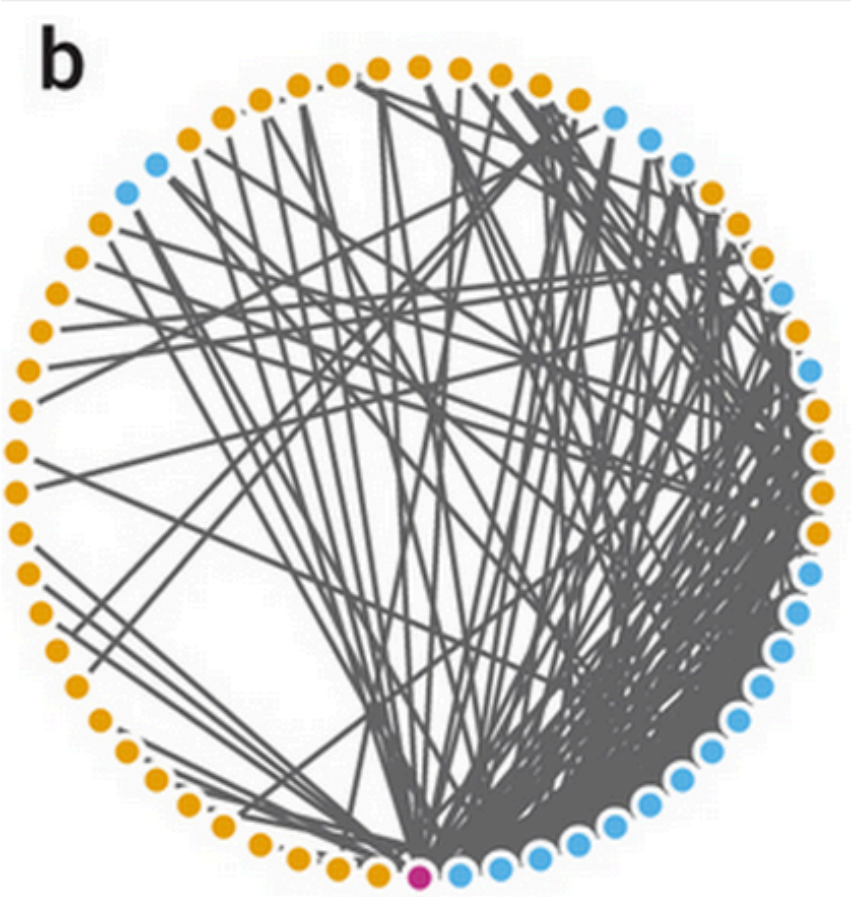
Example 4


a
Discrepancy
in appearance
relating to heigh
3
2
1
0
Occluding
object
0
1
2
3

Example 5



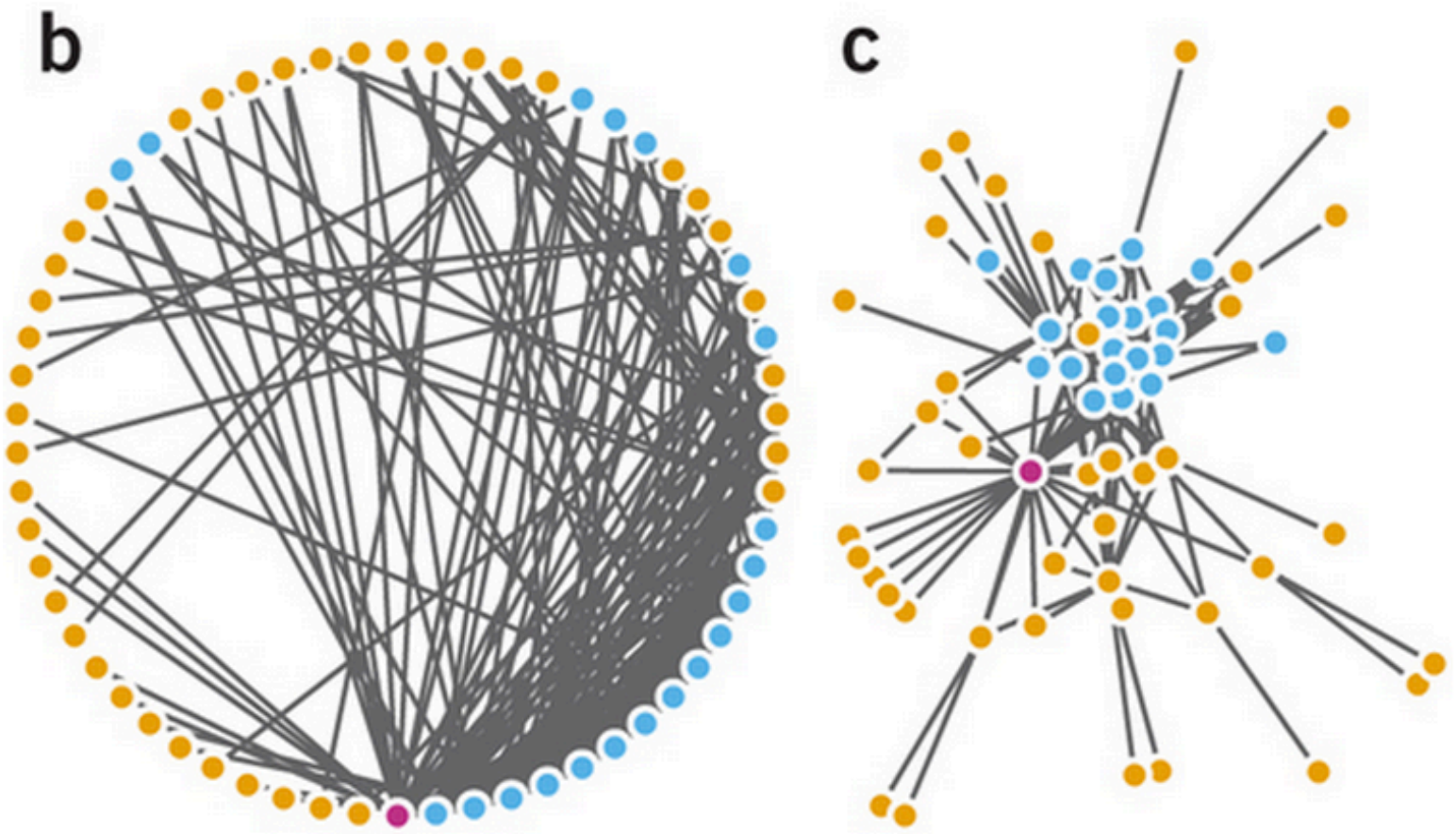
Example 5



An undirected network graph with nodes arranged in a circle can be hard to interpret because lines overlap

Solution:

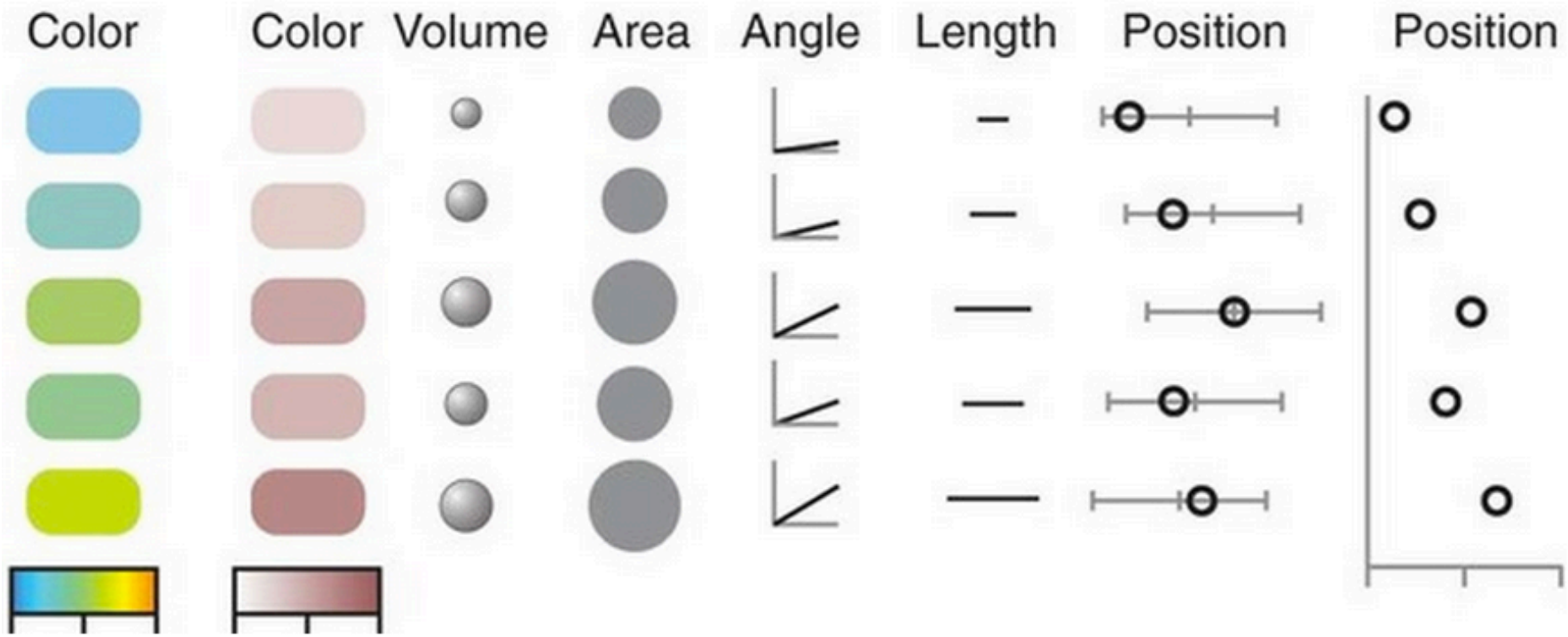
Example 5



An undirected network graph with nodes arranged in a circle can be hard to interpret because lines overlap

Solution: a spring-embedded (force-directed) layout makes it easier to spot nodes

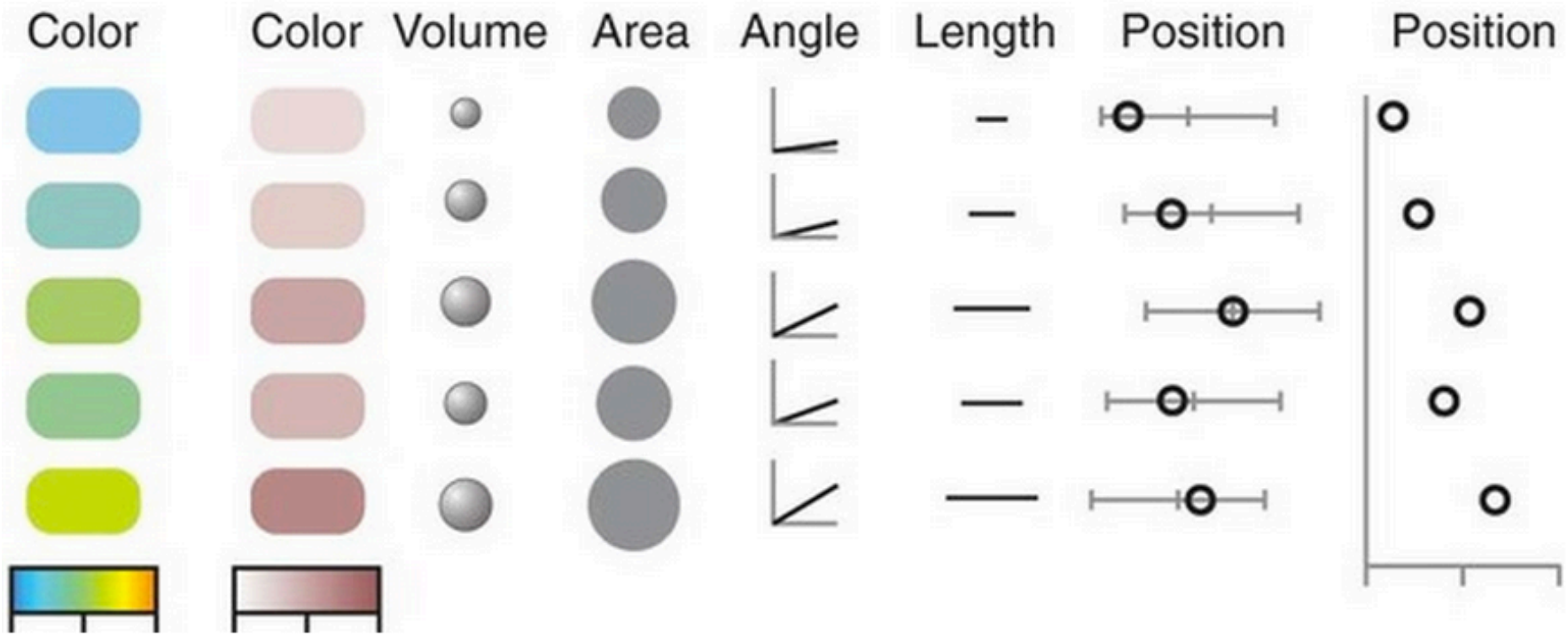
Different Ways of Visualizing Data



Color hue

Color saturation

Different Ways of Encoding Data



Color hue

Color saturation

Less accurately interpreted → More accurately interpreted

How to Make a Bad Graph

- The aim of **good** data graphics:
  - Display data accurately and clearly
  - Allow viewer to quantify differences

- Some rules for displaying data **badly**:
  - Display as little information as possible
  - Obscure what you do show (with chart junk)
  - Extraneous information: use pseudo-3d and color gratuitously
  - Use a difficult-to-decode scheme (e.g. color gradation or area)
  - Use a poorly chosen or deceptive scale

If you want more information

- “Points of View” column by Bang Wong in Nature Methods (http://clearscience.info/wp/?p=546)

- Designing Data Visualization by Noah Illinsky (or see https://www.youtube.com/watch?v=R-oiKt7bUU8)