+/- Message

```
## Loading required package: knitr
```

+/- R Code

+/- R Code

Quantitative Big Imaging

author: Kevin Mader date: 10 April 2014 width: 1440 height: 900 css: ../template.css transition: rotate

ETHZ: 227-0966-00L

Statistics and Reproducibility

Course Outline

- 20th February - Introductory Lecture
- 27th February - Filtering and Image Enhancement (A. Kaestner)
- 6th March - Basic Segmentation, Discrete Binary Structures
- 13th March - Advanced Segmentation
- 20th March - Analyzing Single Objects
- 27th March - Analyzing Complex Objects
- 3rd April - Many Objects and Distributions
- 10th April - **Statistics and Reproducibility**
- 17th April - Dynamic Experiments
- 8th May - Big Data
- 15th May - Guest Lecture - In-Operando Imaging of Batteries (V. Wood)
- 22th May - Project Presentations

Literature / Useful References

Books

- Jean Claude, Morphometry with R
  - Online (http://link.springer.com/book/10.1007%2F978-0-387-77789-4) through ETHZ
  - **Chapter 3**
  - Buy it (http://www.amazon.com/Morphometrics-R-Use-Julien-Claude/dp/038777789X)
- John C. Russ, “The Image Processing Handbook”,(Boca Raton, CRC Press)
  - Available online (http://dx.doi.org/10.1201/9780203881095) within domain ethz.ch (or proxy.ethz.ch / public VPN)
- Hypothesis Testing Chapter (http://www.sagepub.com/upm-data/40007_Chapter8.pdf)

Papers / Sites

- Google/Stanford Statistics Intro
  - https://www.youtube.com/watch?v=YFC2KUmEebc (https://www.youtube.com/watch?v=YFC2KUmEebc)
- MCB 140 P-value lecture at UC Berkeley (Audio)
  - https://itunes.apple.com/us/itunes-u/mcb-140-fall-2007-general/id461120088?mt=10 (https://itunes.apple.com/us/itunes-u/mcb-140-fall-2007-general/id461120088?mt=10)
- Correlation and Causation (Video)
  - https://www.youtube.com/watch?v=YFC2KUmEebc (https://www.youtube.com/watch?v=YFC2KUmEebc)
- Matlab Unit Testing Documentation (http://www.mathworks.ch/ch/help/matlab/matlab-unit-test-framework.html)
- Visualizing Genomic Data (http://circos.ca/documentation/course/visualizing-genomic-data.pdf) (General Visualization Techniques)
- NIMRod Parameter Studies (http://www.messagelab.monash.edu.au/nimrod)
- M.E. Wolak, D.J. Fairbairn, Y.R. Paulsen (2012) Guidelines for Estimating Repeatability. Methods in Ecology and Evolution 3(1):129-137.

Previously on QBI ...

- Image Enhancment
  - Highlighting the contrast of interest in images
  - Minimizing Noise
- Understanding image histograms
- Automatic Methods
- Component Labeling
- Single Shape Analysis
- Complicated Shapes

- Distribution Analysis

Quantitative "Big" Imaging

The course has covered imaging enough and there have been a few quantitative metrics, but "big" has not really entered.

What does **big** mean?

- Not just / even large
- it means being ready for *big data*
- volume, velocity, variety (3 V's)
- scalable, fast, easy to customize

So what is "big" imaging

- doing analyses in a disciplined manner
  - fixed steps
  - easy to regenerate results
  - no *magic*
- having everything automated
  - 100 samples is as easy as 1 sample
- being able to adapt and reuse analyses
  - one really well working script and modify parameters
  - different types of cells
  - different regions

Objectives

1. Scientific Studies all try to get to a single number
   - Make sure this number is describing the structure well (what we have covered before)
   - Making sure the number is meaningful (**today!**)
2. How do we compare the number from different samples and groups?
   - Within a sample or same type of samples
   - Between samples
3. How do we compare different processing steps like filter choice, minimum volume, resolution, etc?
4. How do we evaluate our parameter selection?
5. How can we ensure our techniques do what they are supposed to do?
6. How can we visualize so much data? Are there rules?

Outline

- Motivation (Why and How?)
- Scientific Goals
- Reproducibility
- Statistical metrics and results
- Parameterization
    - Parameter sweep
    - Sensitivity analysis
- Unit Testing
- Visualization

What do we start with?

Going back to our original cell image

1. We have been able to get rid of the noise in the image and find all the cells (lecture 2-4)
2. We have analyzed the shape of the cells using the shape tensor (lecture 5)
3. We even separated cells joined together using Watershed (lecture 6)
4. We have created even more metrics characterizing the distribution (lecture 7)

We have at least a few samples (or different regions), large number of metrics and an almost as large number of parameters to *tune*

How do we do something meaningful with it?

Correlation and Causation

One of the most repeated criticisms of scientific work is that correlation and causation are confused.

1. Correlation
    - means a statistical relationship
    - very easy to show (single calculation)
2. Causation
    - implies there is a mechanism between A and B
    - very difficult to show (impossible to prove)

Controlled and Observational

There are two broad classes of data and scientific studies.

Observational

- Exploring large datasets looking for trends
- Population is random
- Not always hypothesis driven
- Rarely leads to causation

We examined 100 people and the ones with blue eyes were on average 10cm taller

In 100 cake samples, we found a 0.9 correlation between cooking time and bubble size

Controlled

- Most scientific studies fall into this category
- Specifics of the groups are controlled
- Can lead to causation

We examined 50 mice with gene XYZ off and 50 gene XYZ on and as the foot size increased by 10%

We increased the temperature and the number of pores in the metal increased by 10%

Simple Model: Magic / Weighted Coin

incremental: true

Since most of the experiments in science are usually specific, noisy, and often very complicated and are not usually good teaching examples

- Magic / Biased Coin
  - You buy a *magic* coin at a shop
  - How many times do you need to flip it to *prove* it is not fair?
  - If I flip it 10 times and another person flips it 10 times, is that the same as 20 flips?
  - If I flip it 10 times and then multiple the results by 10 is that the same as 100 flips?
  - If I buy 10 coins and want to know which ones are fair what do I do?

Simple Model: Magic / Weighted Coin

1. Each coin represents a stochastic variable \(\mathcal{X}\) and each flip represents an observation \(\mathcal{X}_i\).
2. The act of performing a coin flip \(\mathcal{F}\) is an observation \(\mathcal{X}_i = \mathcal{F}(\mathcal{X})\)

We normally assume

1. A *fair* coin has an expected value of \(E(\mathcal{X})=0.5 \rightarrow\) 50% Heads, 50% Tails

2. An *unbiased* flip(er) means
    - each flip is independent of the others \[P(\mathcal{F}_1(\mathcal{X})*\mathcal{F}_2(\mathcal{X}))= P(\mathcal{F}_1(\mathcal{X}))*P(\mathcal{F}_2(\mathcal{X}))\]
    - the expected value of the flip is the same as that of the coin \[E(\prod_{i=0}^\infty \mathcal{F}_i(\mathcal{X})) = E(\mathcal{X}) \]

Simple Model to Reality

Coin Flip

1. Each flip gives us a small piece of information about the flipper and the coin
2. More flips provides more information
    - Random / Stochastic variations in coin and flipper cancel out
    - Systematic variations accumulate

Real Experiment

1. Each measurement tells us about our sample, out instrument, and our analysis
2. More measurements provide more information
    - Random / Stochasit variations in sample, instrument, and analysis cancel out
    - *Normally* the analysis has very little to no stochastic variation
    - Systematic variations accumulate

Reproducibility

A very broad topic with plenty of sub-areas and deeper meanings. We mean two things by reproducibility

Analysis

The process of going from images to numbers is detailed in a clear manner that *anyone*, *anywhere* could follow and get the exact (within some tolerance) same numbers from your samples

- No platform dependence
- No proprietary or "in house" algorithms
- No manual *clicking*, *tweaking*, or *copying*
- One script to go from image to result

Measurement

Everyhing for analysis + taking a measurement several times (noise and exact alignment vary each time) does not change the statistics *significantly*

- No sensitivity to mounting or rotation
- No sensitivity to noise
- No dependence on exact illumination

Reproducible Analysis

The basis for reproducible scripts and analysis are scripts and macros. Since we will need to perform the same analysis many times to understand how reproducible it is.

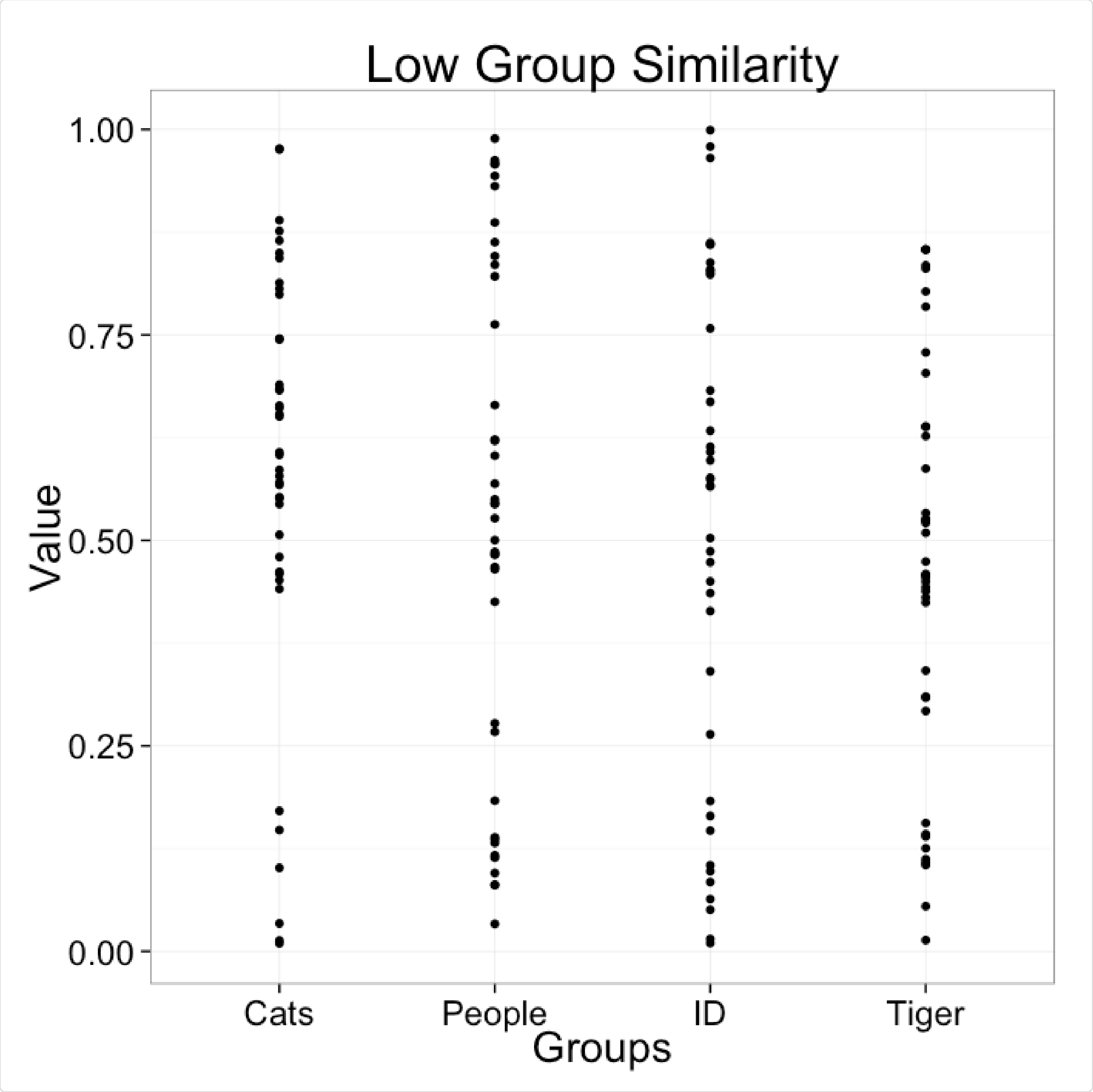
```
IMAGEFILE=$1
THRESHOLD=130
matlab -r "inImage=$IMAGEFILE; threshImage=inImage>$THRESHOLD; analysisScript;"
```

- **or** `java -jar ij.jar -macro TestMacro.ijm blobs.tif`
- **or** `Rscript -e "library(plyr);..."`

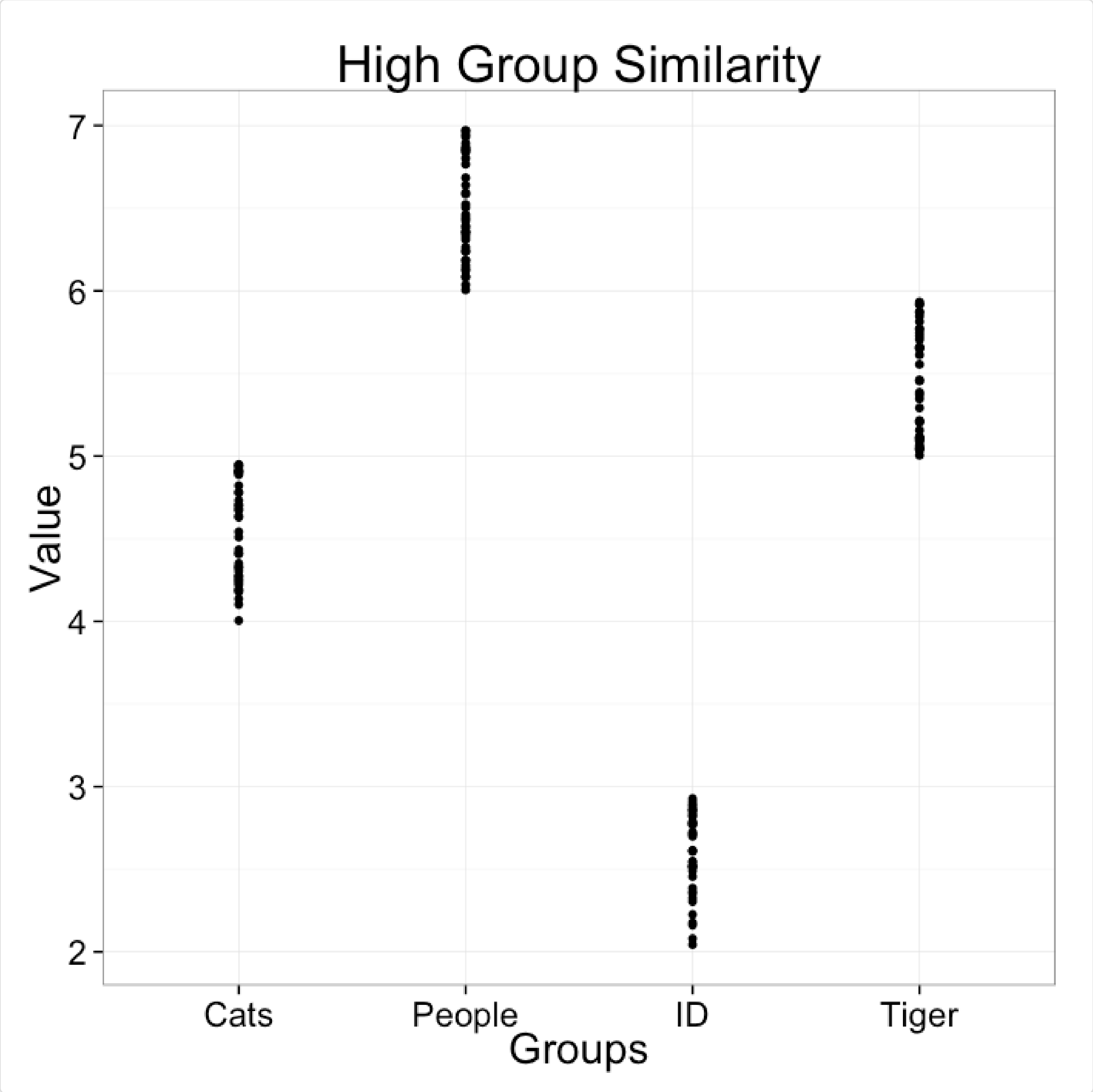
Comparing Groups: Intraclass Correlation Coefficient

The intraclass correlation coefficient basically looking at how similar objects within a group are compared to between groups

+/- R Code


Low Group Similarity
1.00
0.75
0.50
0.25
0.00
Value
Cats
People
ID
Tiger
Groups

+/- R Code



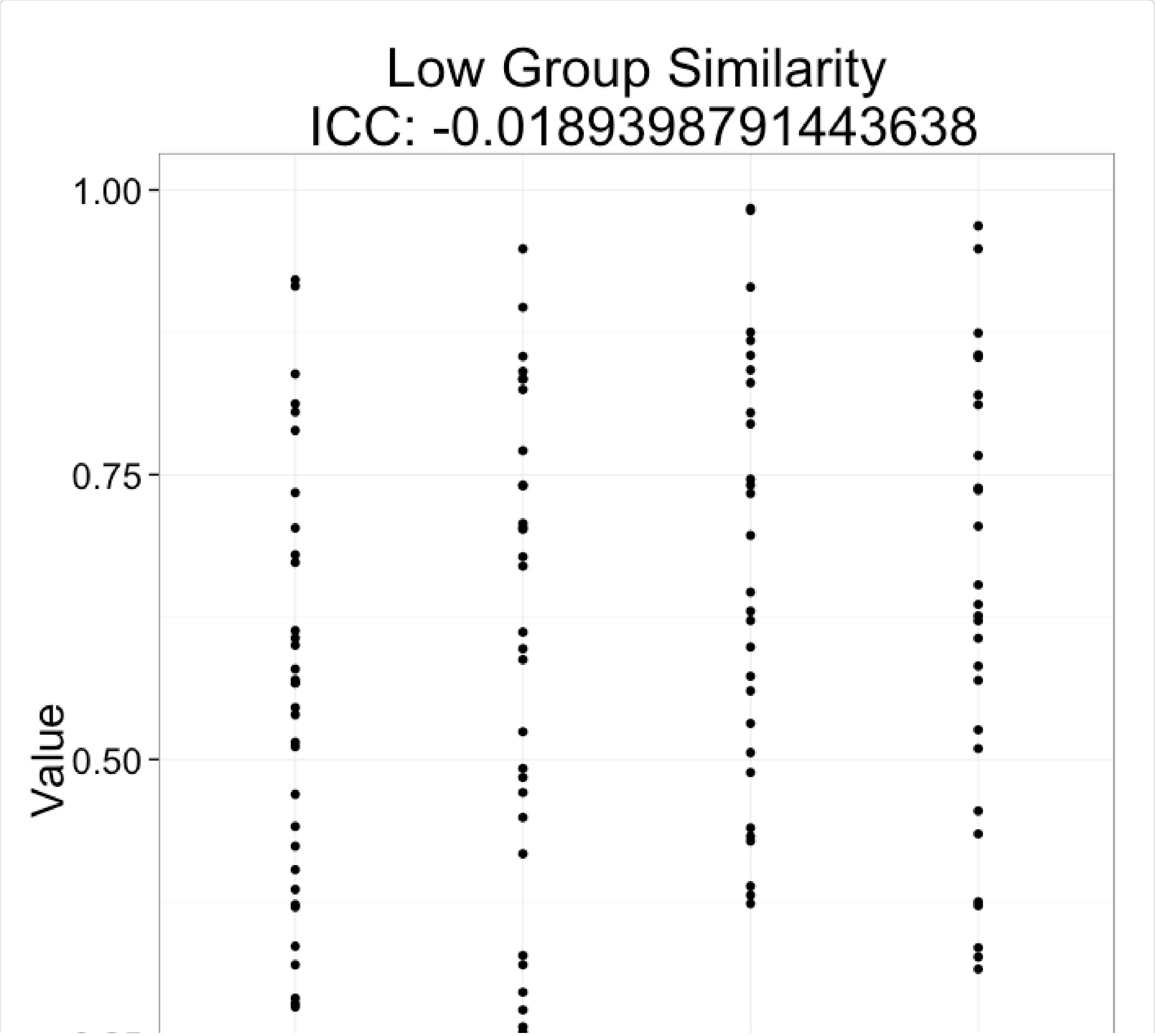
Intraclass Correlation Coefficient Definition

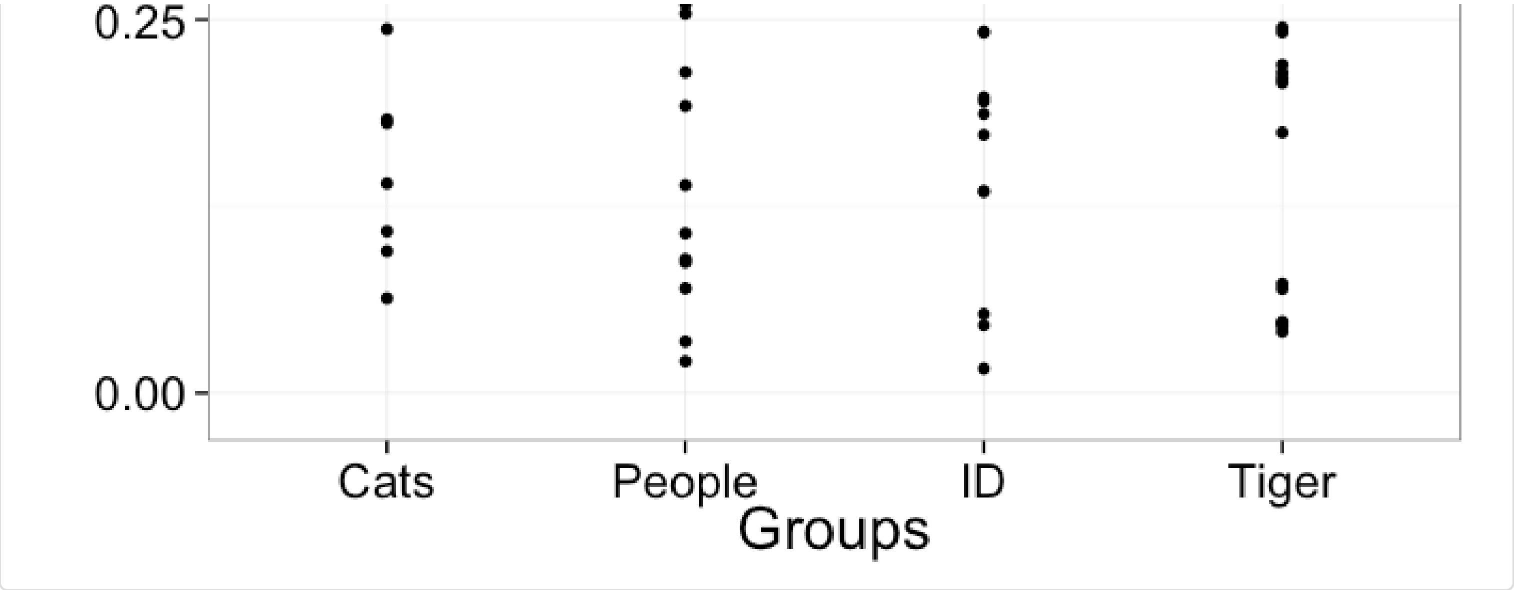
$$ ICC = \frac{S_A^2}{S_A^2+S_W^2} $$

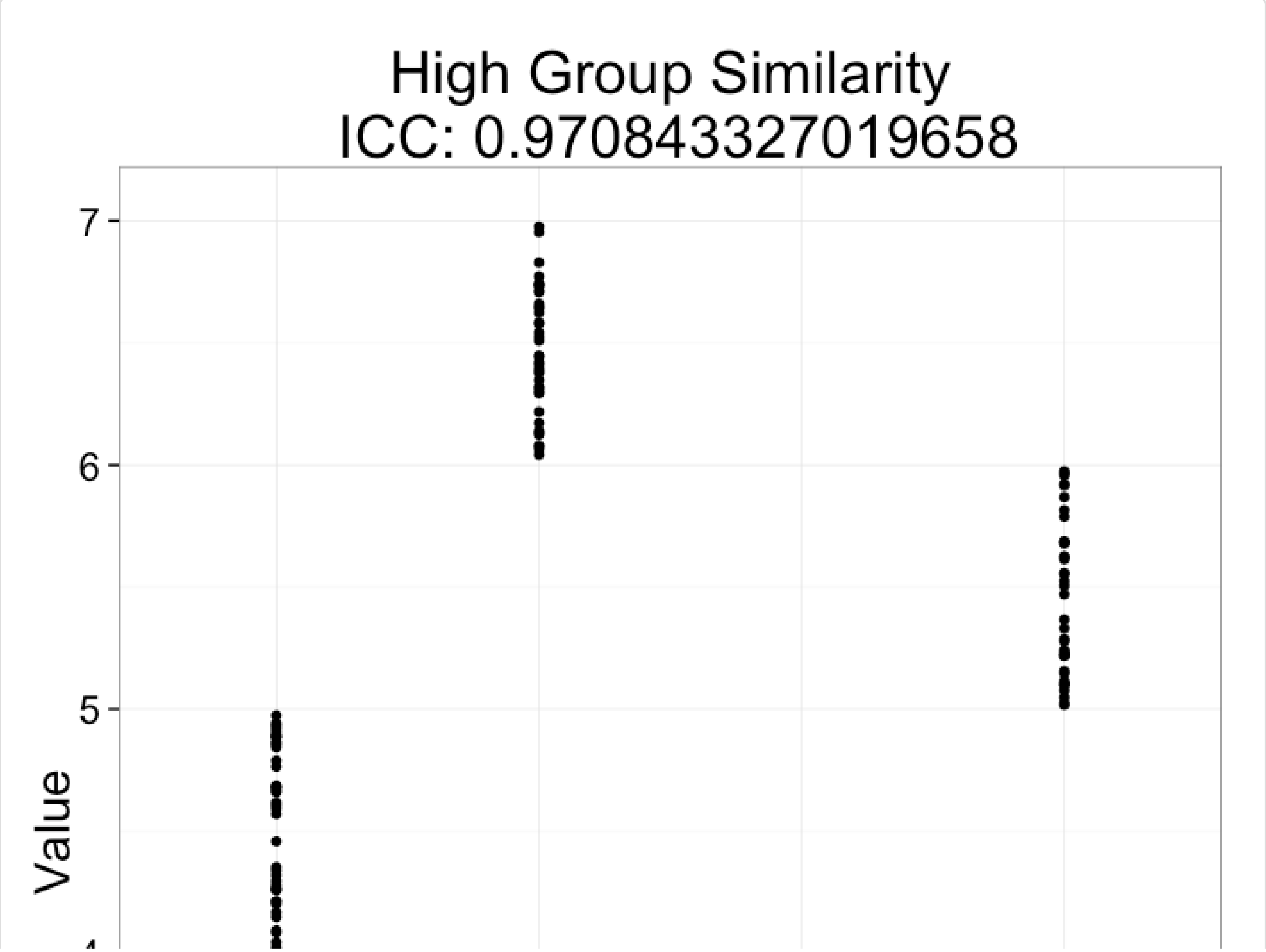
where

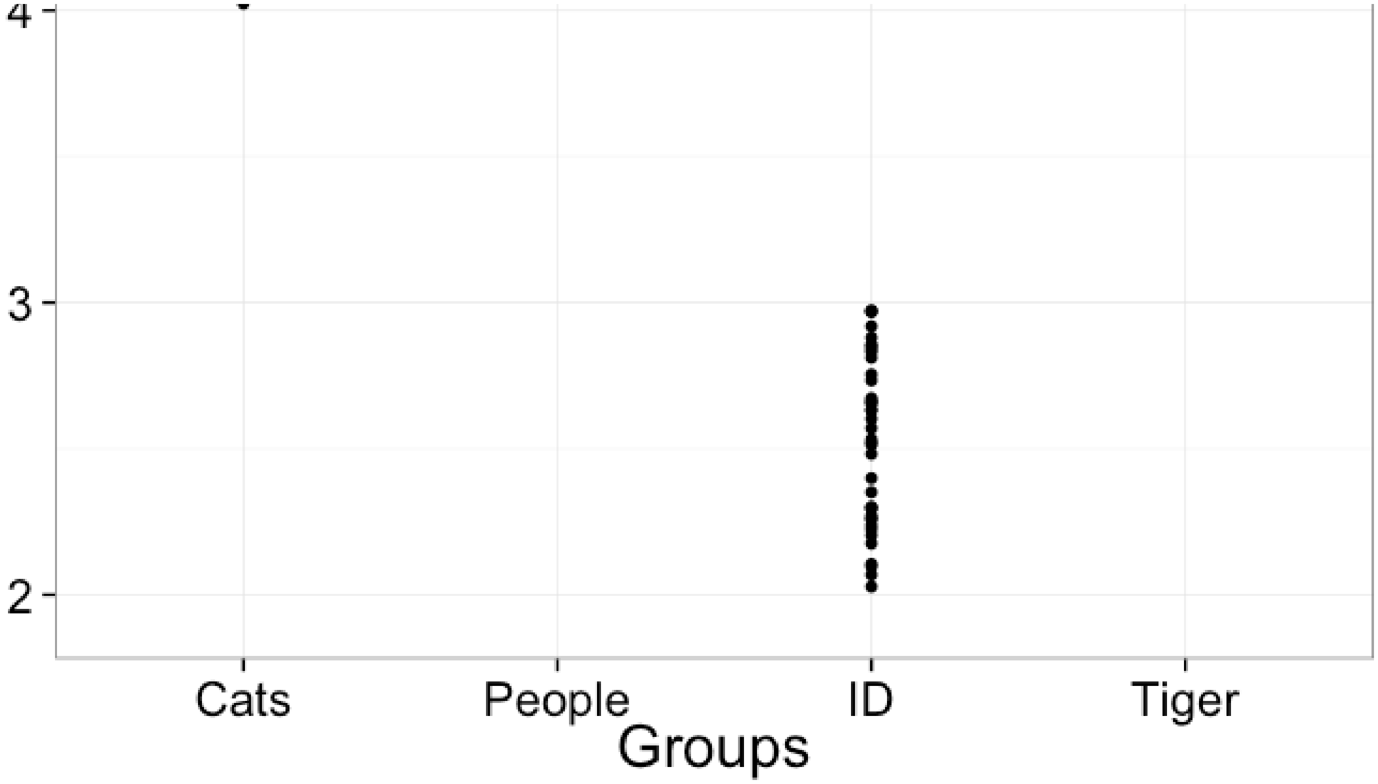
- S_A^2 is the variance among groups or classes
    - Estimate with the standard deviations of the mean values for each group
- S_W^2 is the variance within groups or classes.
    - Estimate with the average of standard deviations for each group
- 1 means 100% of the variance is between classes
- 0 means 0% of the variance is between classes

Intraclass Correlation Coefficient: Values





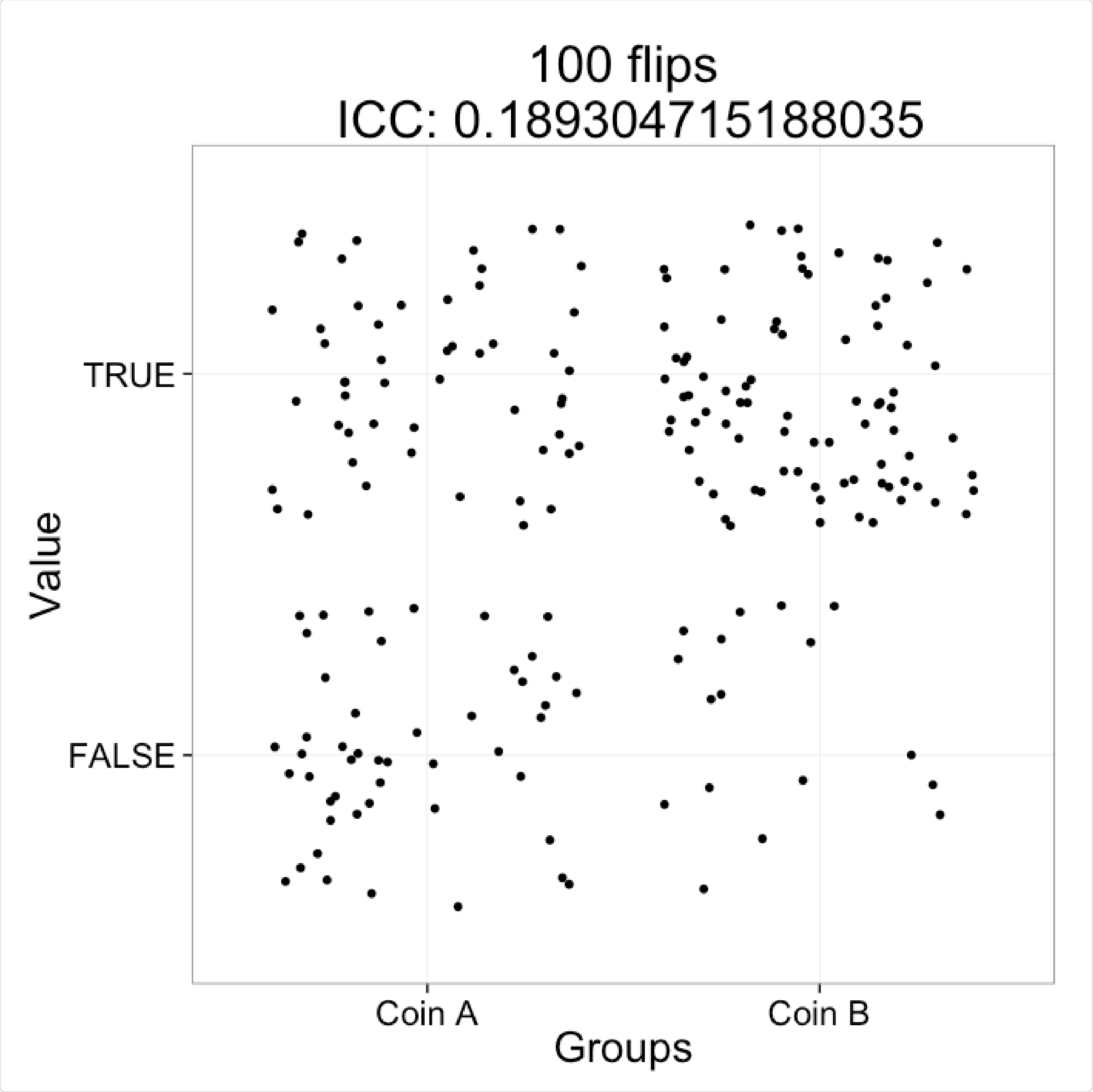

High Group Similarity
ICC: 0.970843327019658



Intraclass Correlation Coefficient: Values for Coin-Flips

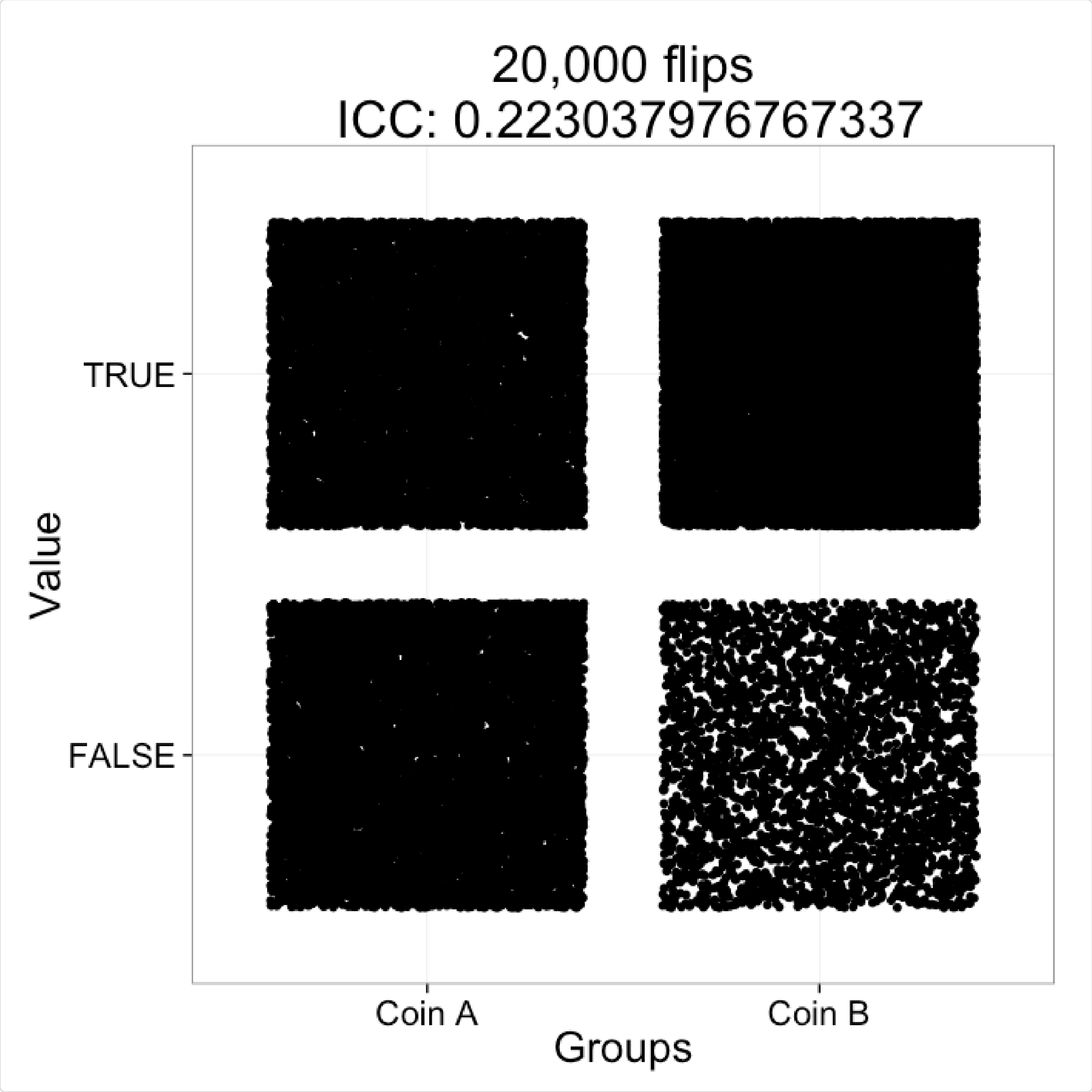
We have one biased coin and try to figure out how many flips we need for the ICC to tell the difference to the normal coin

+/- R Code



With many thousands of flips we eventually see a very strong difference but unless it is very strongly biased ICC is a poor indicator for the differences

+/- R Code



Comparing Groups: Tests

Once the reproducibility has been measured, it is possible to compare groups. The idea is to make a test to assess the likelihood that two groups are the same given the data

1. List assumptions

2. Establish a null hypothesis
    - Usually both groups are the same
3. Calculate the probability of the observations given the truth of the null hypothesis
    - Requires knowledge of probability distribution of the data
    - Modeling can be exceptionally complicated

Loaded Coin

We have 1 coin from a magic shop

- our assumptions are
    - we flip and observe flips of coins accurately and independently
    - the coin is invariant and always has the same expected value
- our null hypothesis is the coin is unbiased \(E(\mathcal{X})=0.5\)
- we can calculate the likelihood of a given observation given the number of flips (p-value)

+/- R Code

Number of Flips	Probability of All Heads Given Null Hypothesis (p-value)
1	50 %
5	3.1 %
10	0.1 %

How good is good enough?

Comparing Groups: Student's T Distribution

Since we do not usually know our distribution very well *or* have enough samples to create a sufficient probability model

Student T Distribution (http://en.wikipedia.org/wiki/Student's_t-distribution)

We assume the distribution of our stochastic variable is normal (Gaussian) and the t-distribution provides an estimate for the mean of the underyling distribution based on few observations.

- We estimate the likelihood of our observed values assuming they are coming from random observations of a normal process

Student T-Test

Incorporates this distribution and provides an easy method for assessing the likelihood that the two given set of observations are coming from the same underlying process (null hypothesis)

- Assume unbiased observations
- Assume normal distribution

Multiple Testing Bias

Back to the magic coin, let's assume we are trying to publish a paper, we heard a p-value of < 0.05 (5%) was good enough. That means if we get 5 heads we are good!

+/- R Code

Number of Flips	Probability of All Heads Given Null Hypothesis (p-value)
1	50 %
4	6.2 %
5	3.1 %

+/- R Code

Number of Friends Flipping	Probability Someone Flips 5 heads
1	3.1 %
10	27.2 %
20	47 %
40	71.9 %
80	92.1 %

Clearly this is not the case, otherwise we could keep flipping coins or ask all of our friends to flip until we got 5 heads and publish

The p-value is only meaningful when the experiment matches what we did.

- We didn't say the chance of getting 5 heads ever was < 5%
- We said if we have exactly 5 observations and all of them are heads the likelihood that a fair coin produced that result is <5%

Many methods (http://en.wikipedia.org/wiki/Multiple_comparisons_problem) to correct, most just involve scaling \(p\). The likelihood of a sequence of 5 heads in a row if you perform 10 flips is 5x higher.

Multiple Testing Bias: Experiments

This is very bad news for us. We have the ability to quantify all sorts of interesting metrics

- cell distance to other cells
- cell oblateness
- cell distribution oblateness

So lets throw them all into a magical statistics algorithm and push the **publish** button

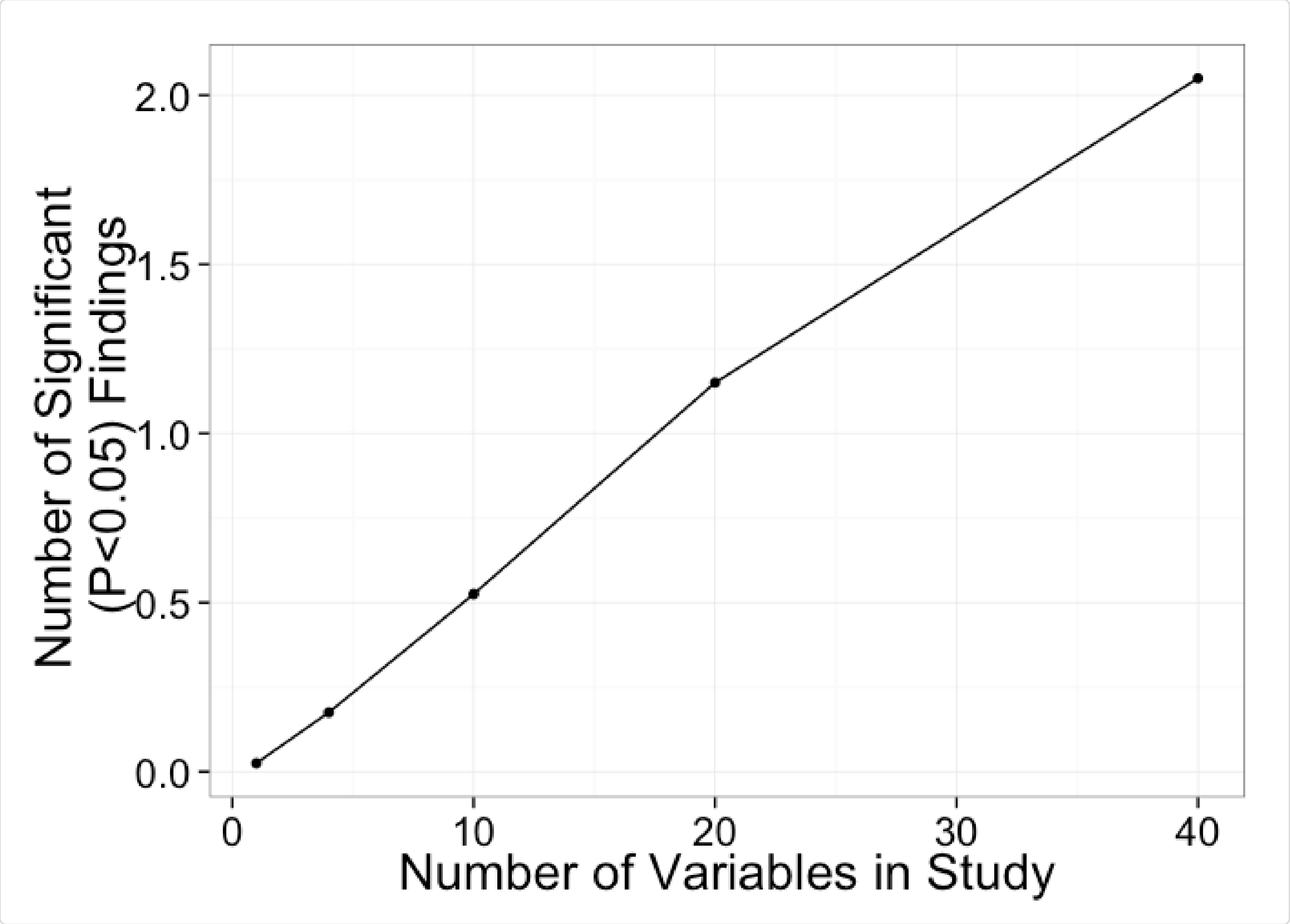
---

With our p value of less than 0.05 and a study with 10 samples in each group, how does increasing the number of variables affect our result

+/- R Code

+/- R Code

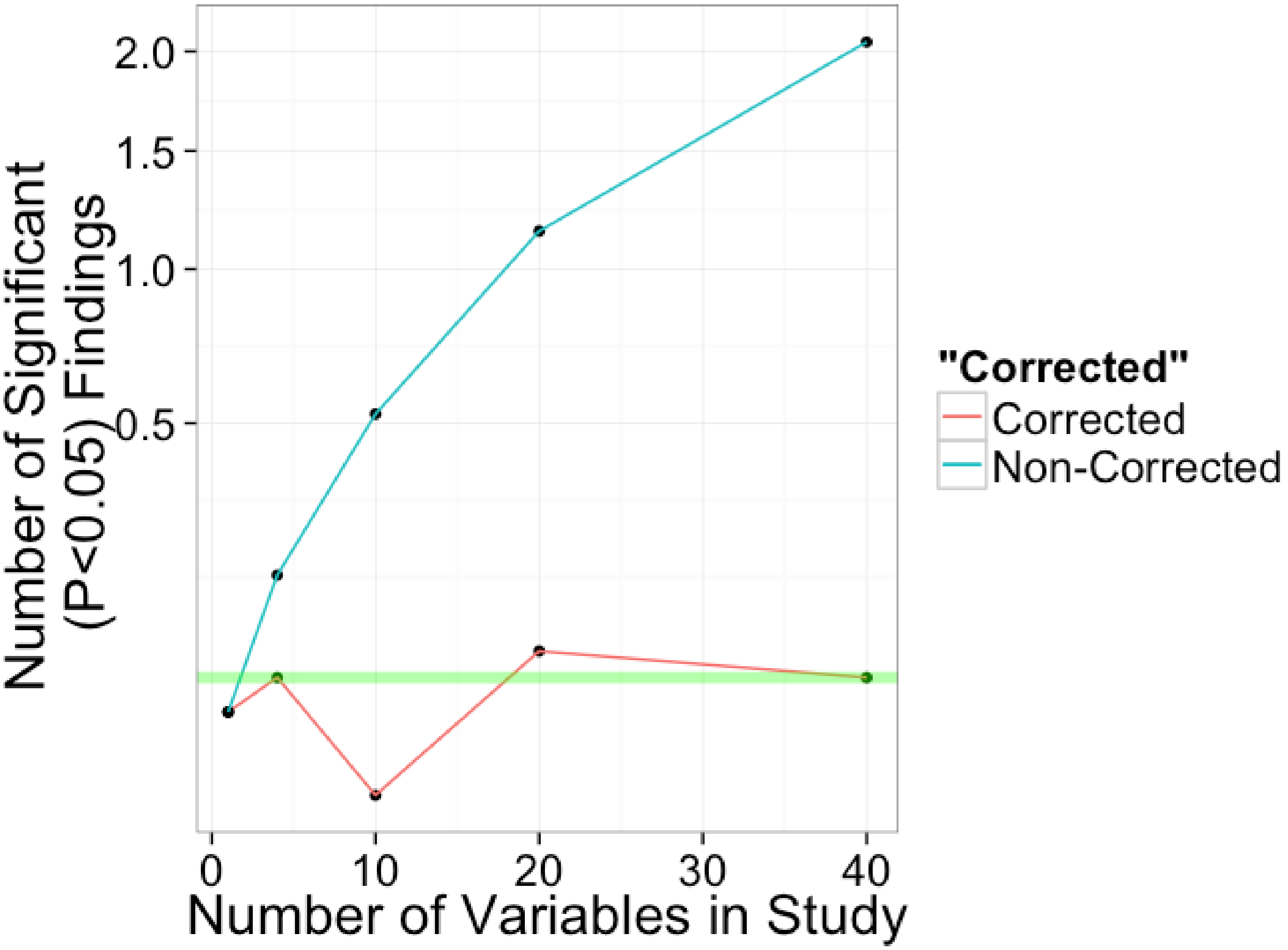
+/- R Code



Multiple Testing Bias: Correction

Using the simple correction factor (number of tests performed), we can make the significant findings constant again

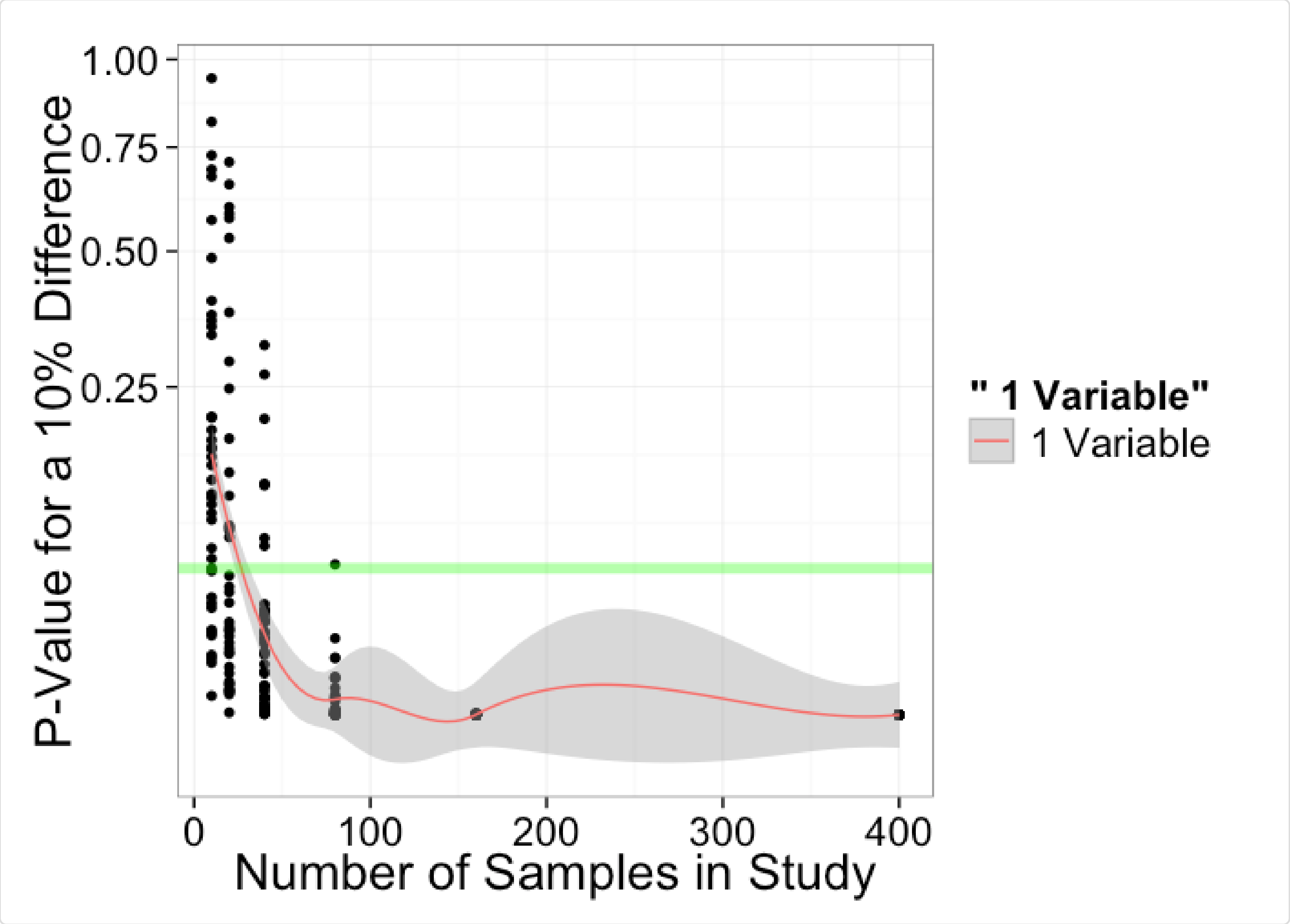
+/- R Code



So no harm done there we just add this correction factor right? Well what if we have exactly one variable with shift of 10% on top of 0-mean Gaussian distributions

+/- R Code

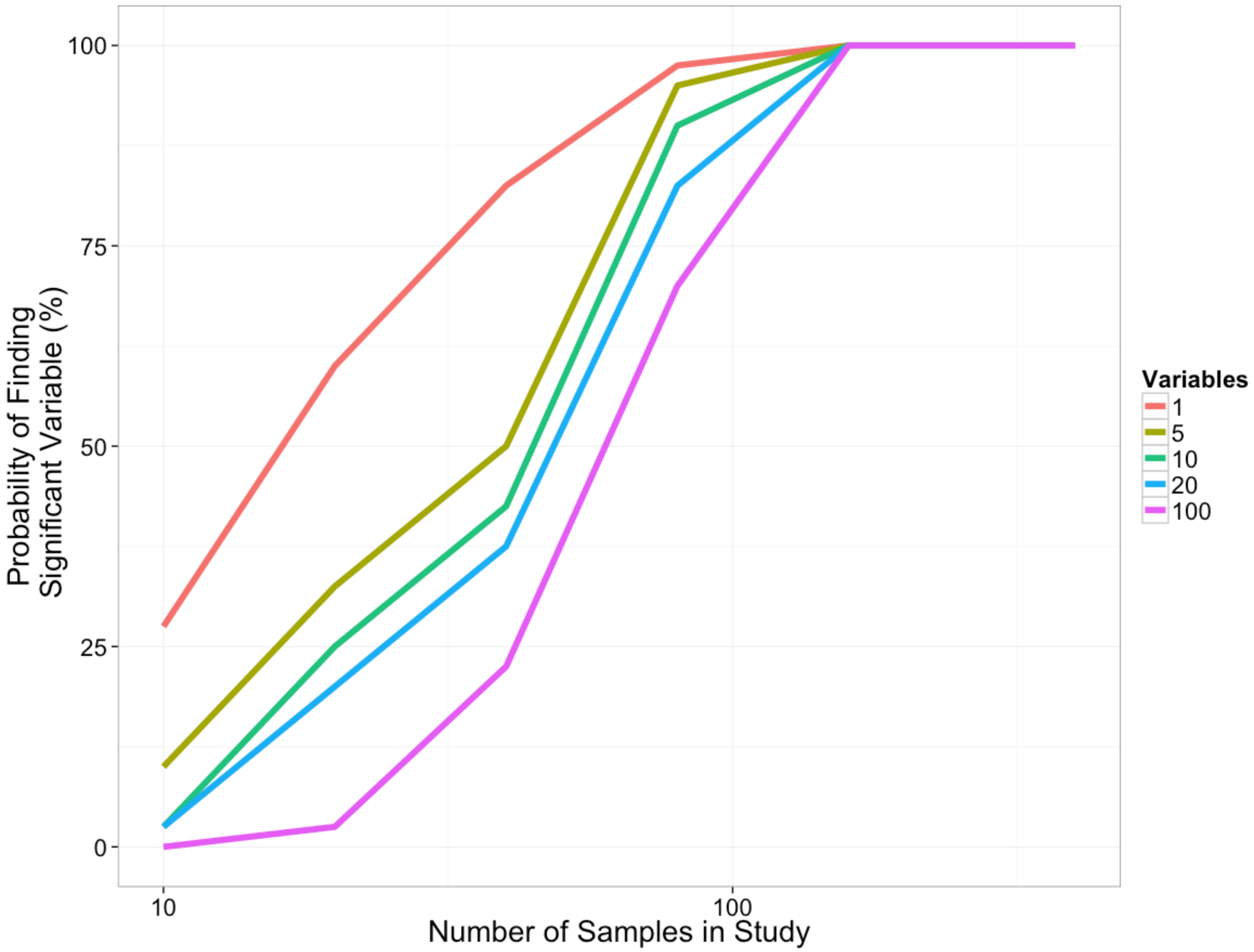
+/- R Code



Multiple Testing Bias: Sample Size

+/- R Code

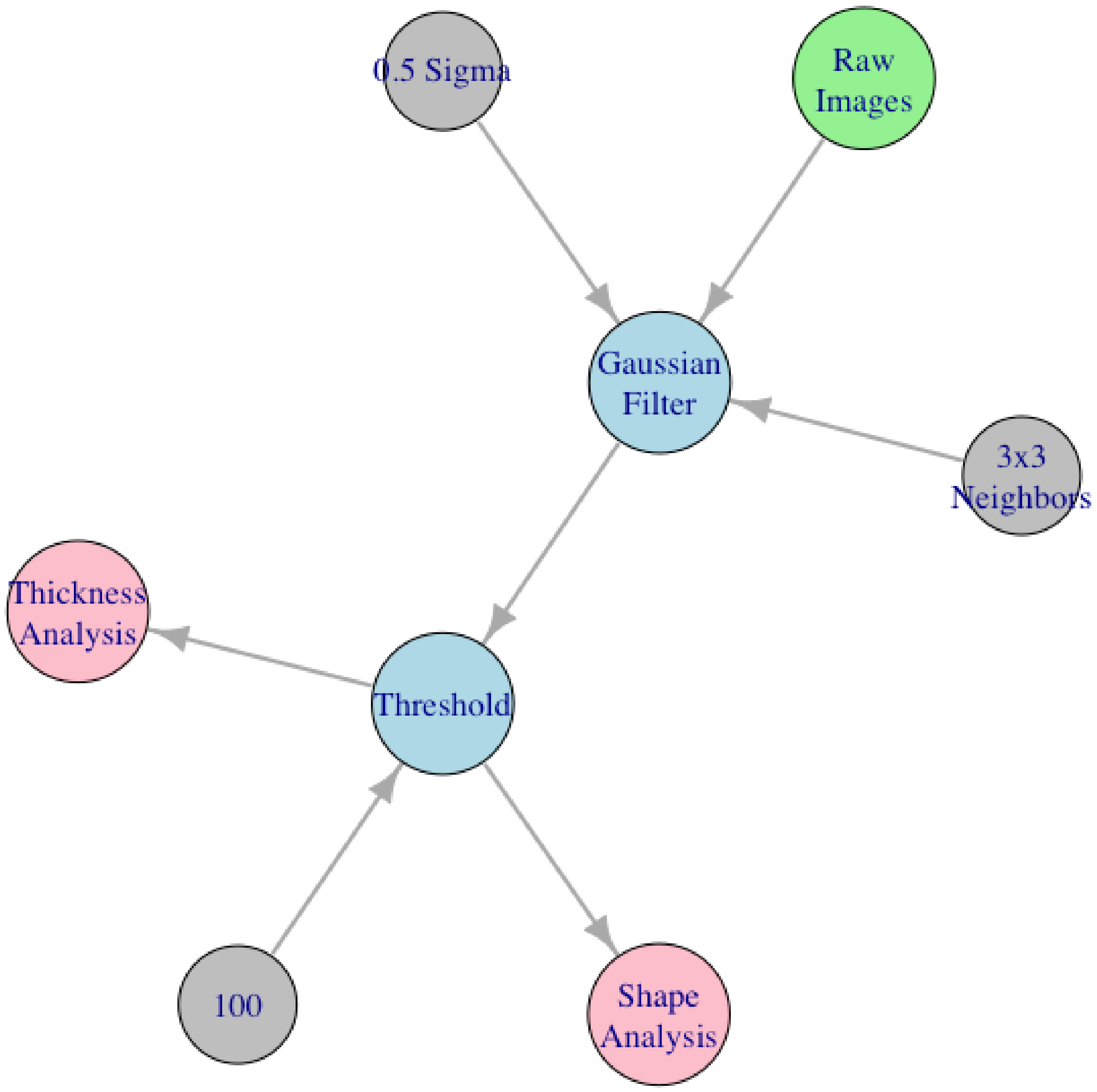
+/- R Code



Parameters

+/- R Code

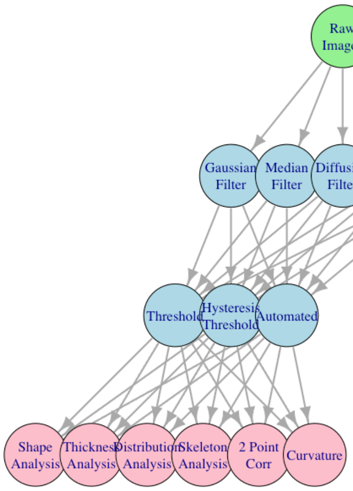
How does a standard image processing chain look?



- Green are the images we start with (measurements)

- Blue are processing steps
- Gray are use input parameters
- Pink are the outputs

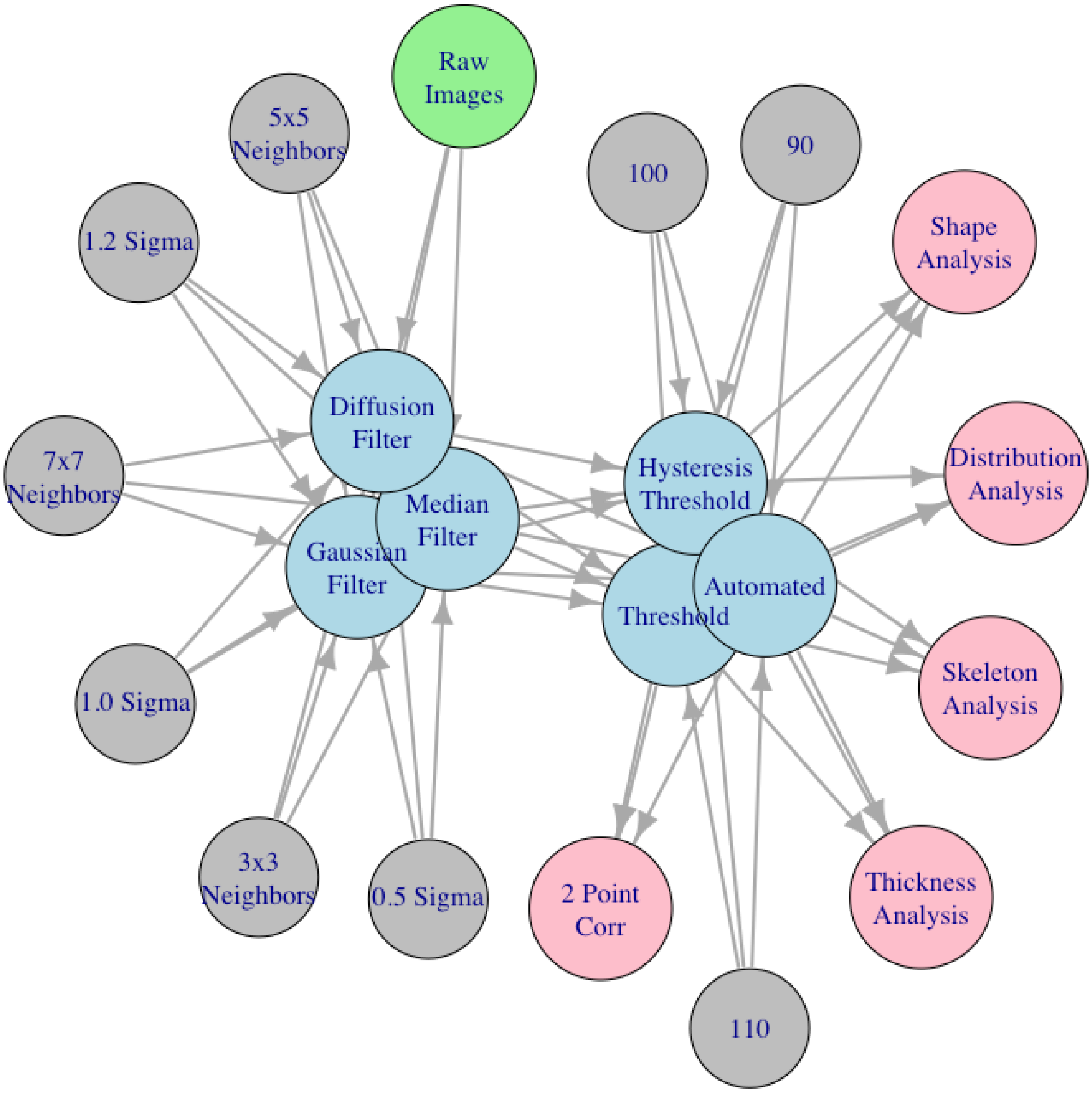
The Full Chain



+/- R Code

The Full Chain (with Parameters)

+/- R Code



- A **mess**, over 1080 combinations for just one sample (not even exploring a very large range of threshold values)
- To calculate this for even one sample can take days (weeks, years)
    - 512 x 512 x 512 foam sample \(\rightarrow\) 12 weeks of processing time

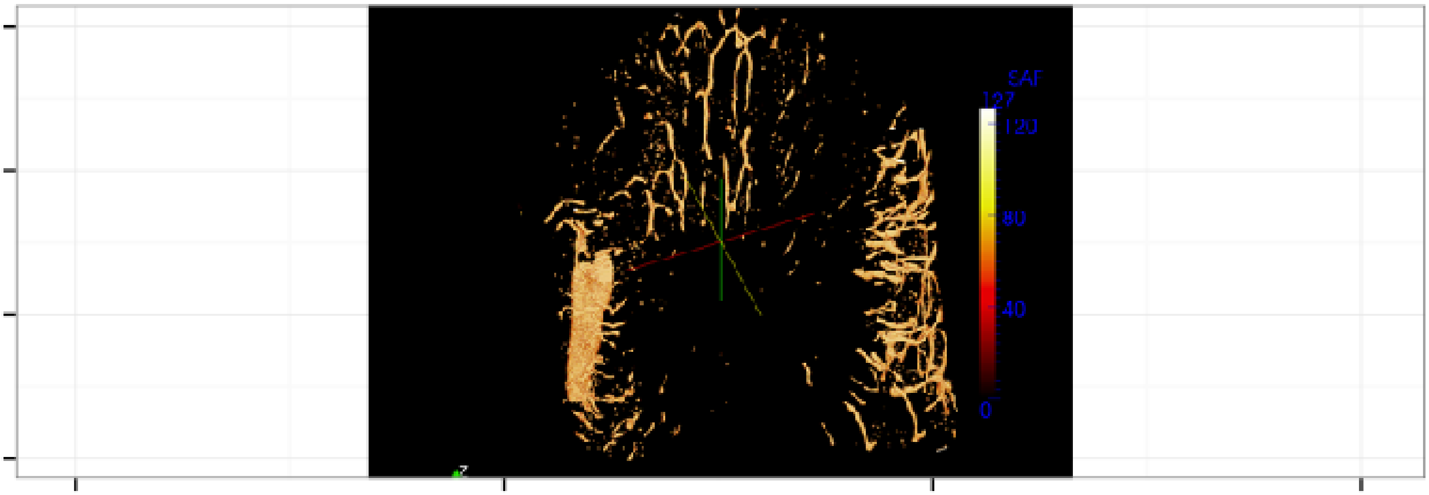
  - 1024 x 1024 x 1024 femur bone \(\rightarrow\) 1.9 years
- Not all samples are the same
- Once the analysis is run we have a ton of data
  - femur bone \(\rightarrow\) 60 million shapes analyzed
- What do we even want?
- How do we judge the different results?

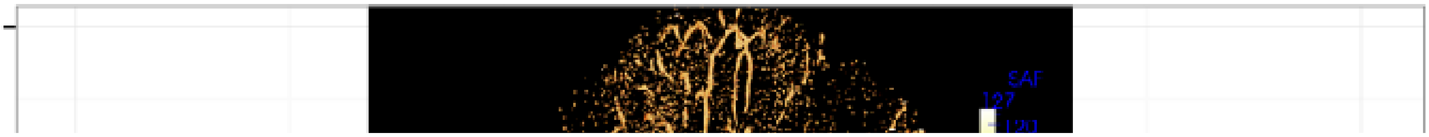
Qualitative vs Quantitative

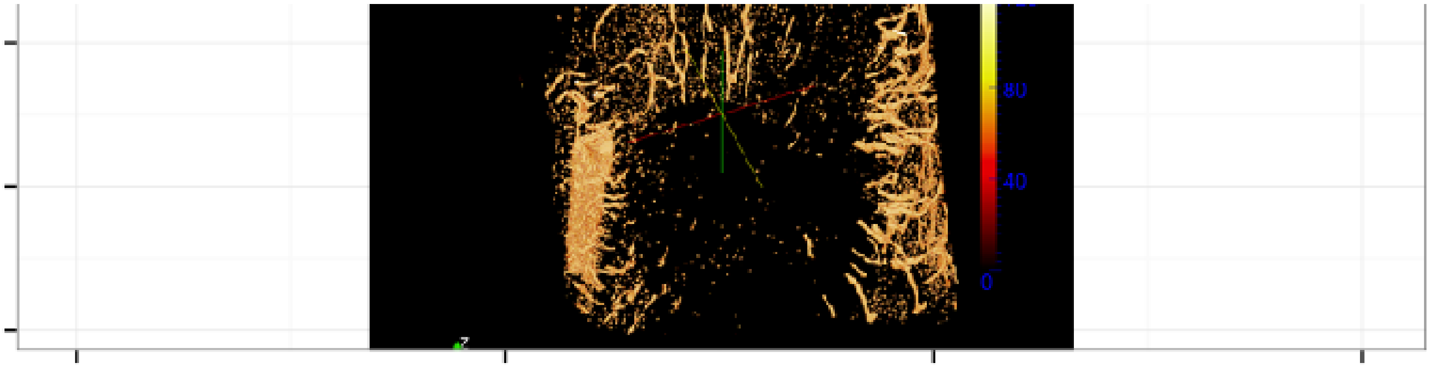
Given the complexity of the tree, we need to do some pruning

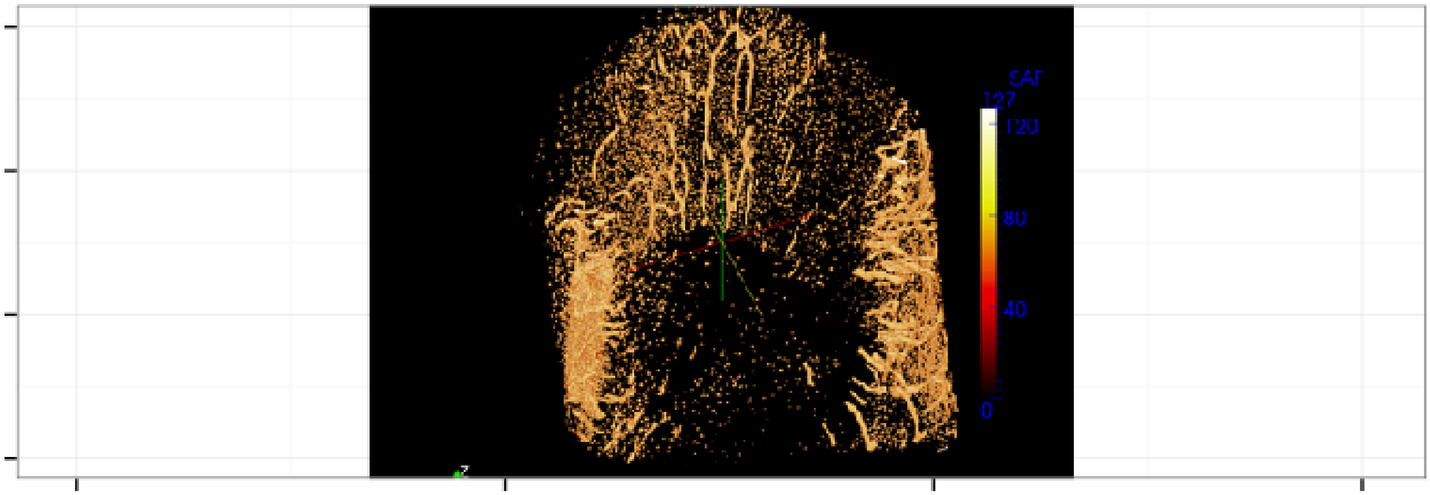
Qualitative Assessment

- Evaluating metrics using visual feedback
- Compare with expectations from other independent techniques or approach
- Are there artifacts which are included in the output?
- Do the shapes look correct?
- Are they distributed as expected?
- Is their orientation menaingful?









Quantitative Metrics

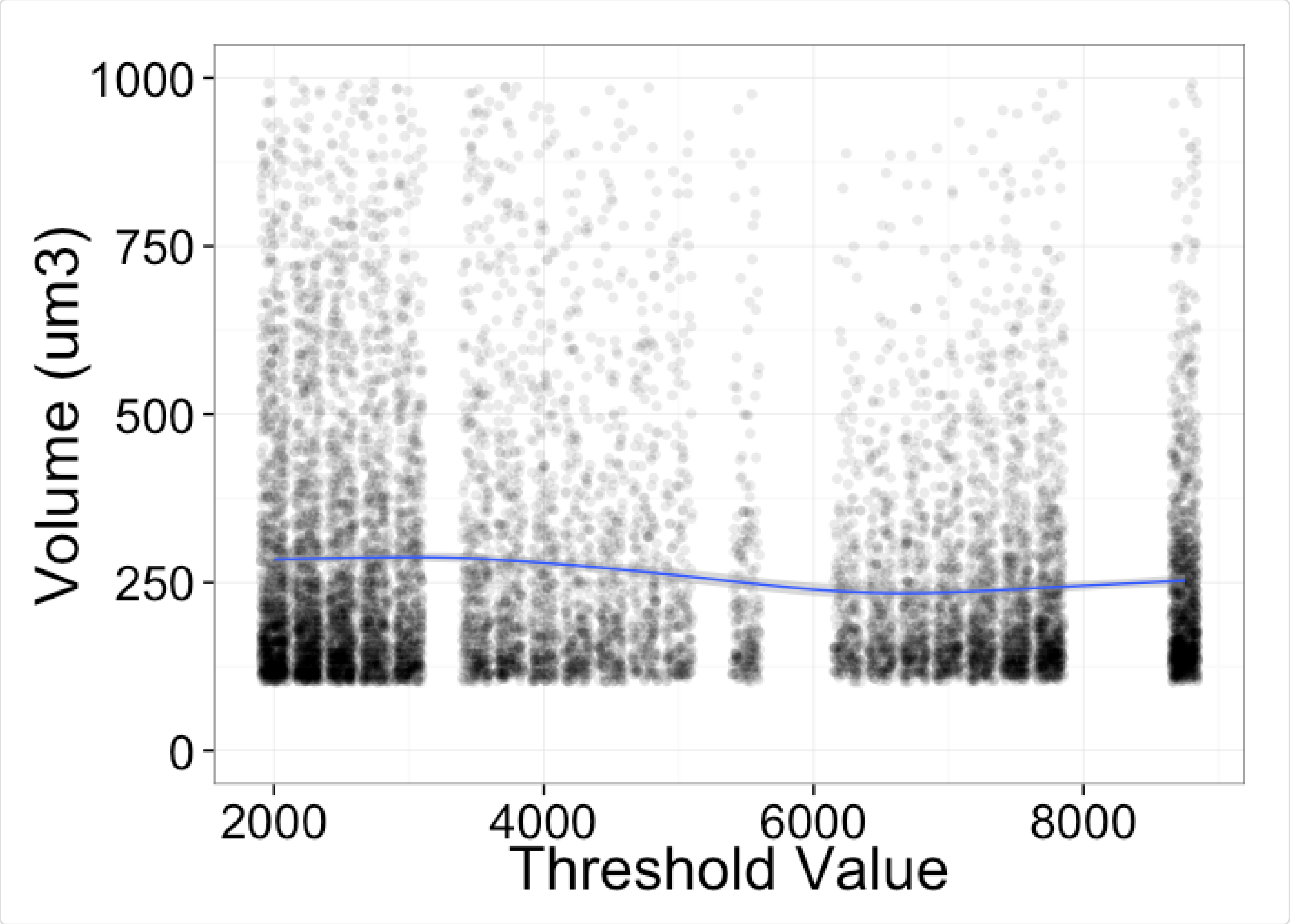
With a quantitative approach, we can calculate the specific shape or distribution metrics on the sample with each parameter and establish the relationship between parameter and metric.

Parameter Sweep

The way we do this is usually a parameter sweep which means taking one (or more) parameters and varying them between the reasonable bounds (judged qualitatively).

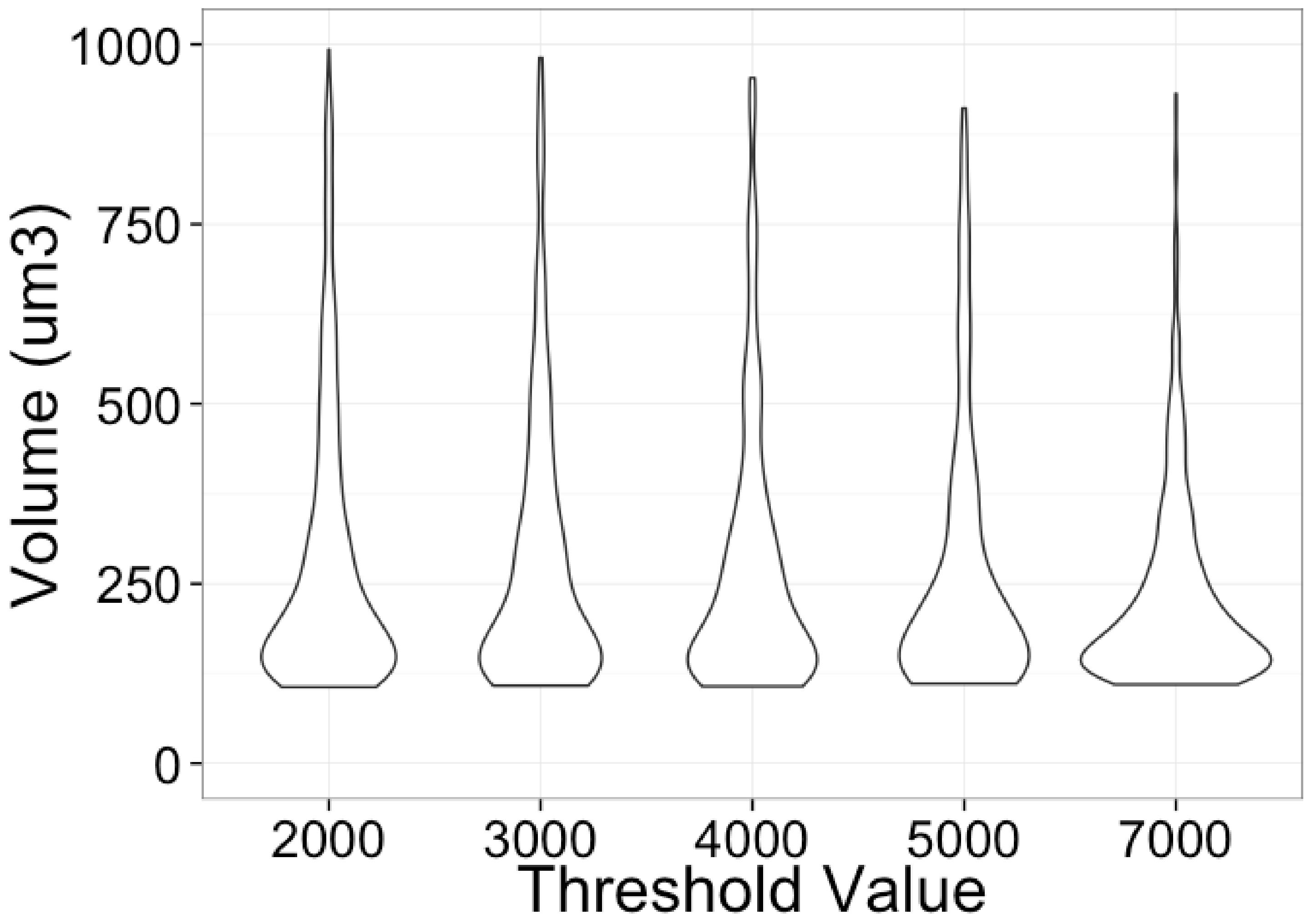
+/- R Code

+/- R Code

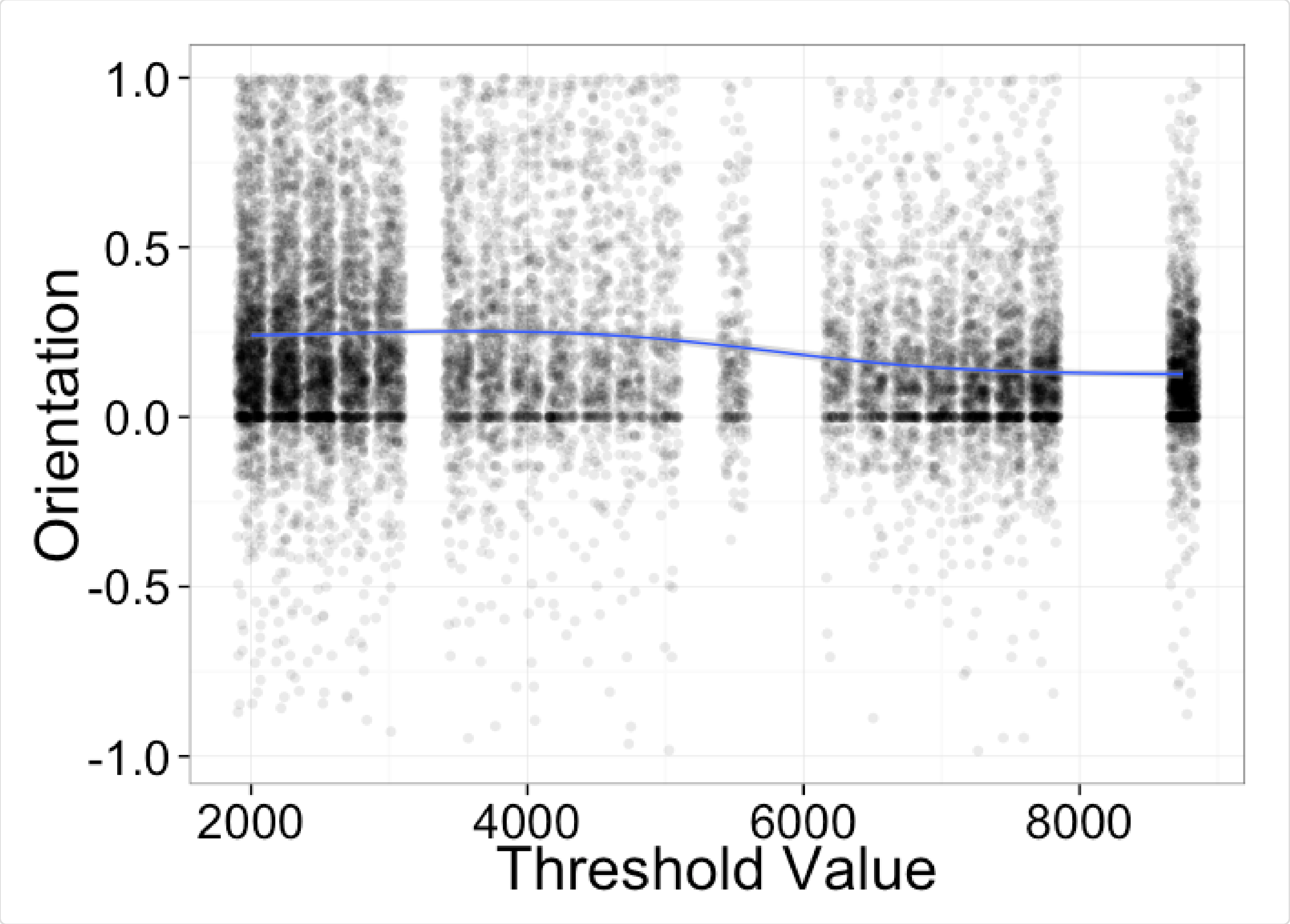


Is it always the same?

+/- R Code


Volume (um3)
1000
750
500
250
0
2000
3000
4000
5000
7000
Threshold Value

+/- R Code



Sensitivity

Sensitivity is defined in control system theory as the change in the value of an output against the change in the input. \[S = \frac{|\Delta \textrm{Metric}|}{|\Delta \textrm{Parameter}|} \]

Such a strict definition is not particularly useful for image processing since a threshold has a unit of intensity and a metric might be volume which has \(m^3\) so the sensitivity becomes volume per intensity.

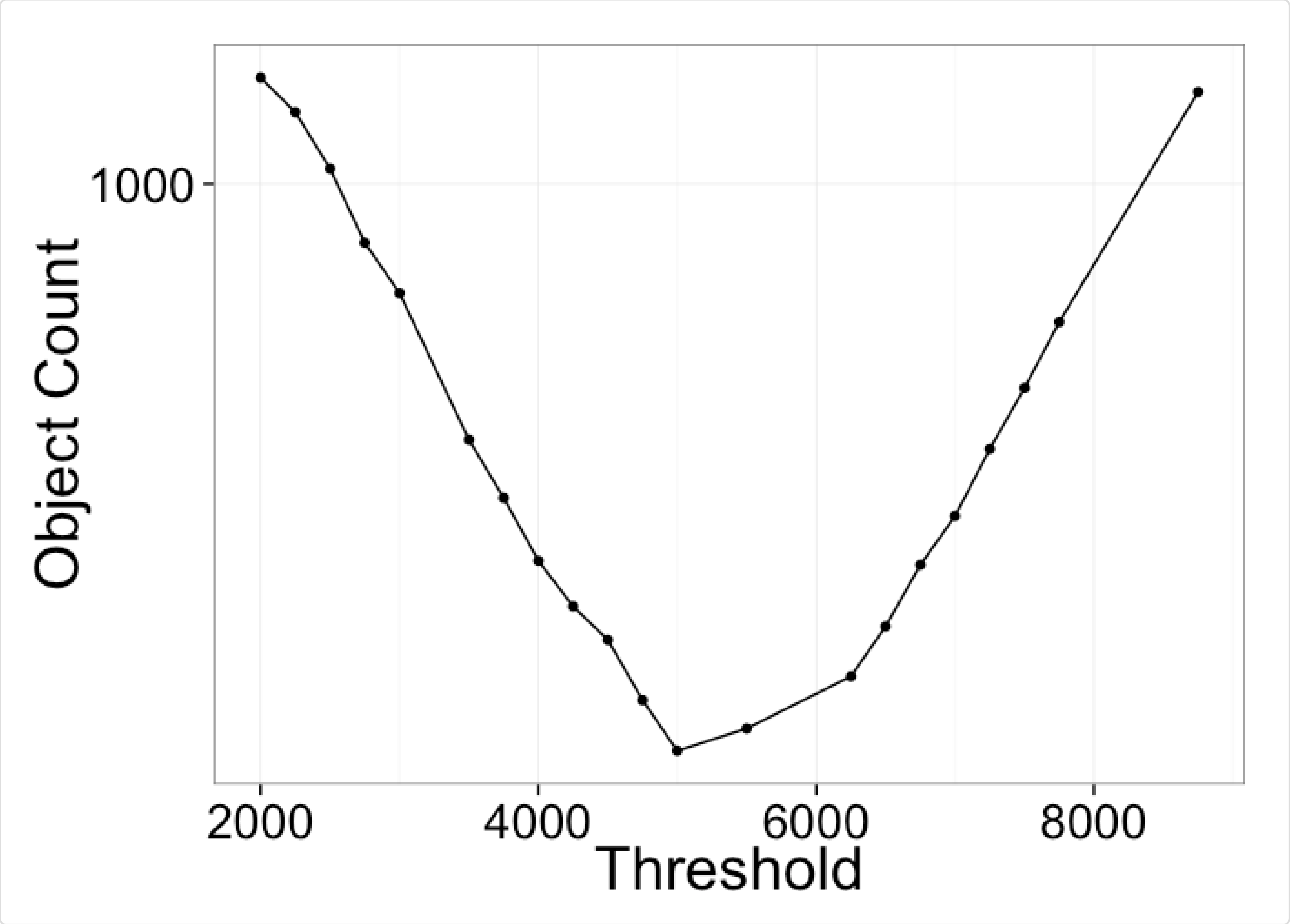
Practical Sensitivity

A more common approach is to estimate the variation in this parameter between images or within a single image (automatic threshold methods can be useful for this) and define the sensitivity based on this variation. It is also common to normalize it with the mean value so the result is a percentage.

\[S = \frac{max(\textrm{Metric})-min(\textrm{Metric})}{avg(\textrm{Metric})} \]

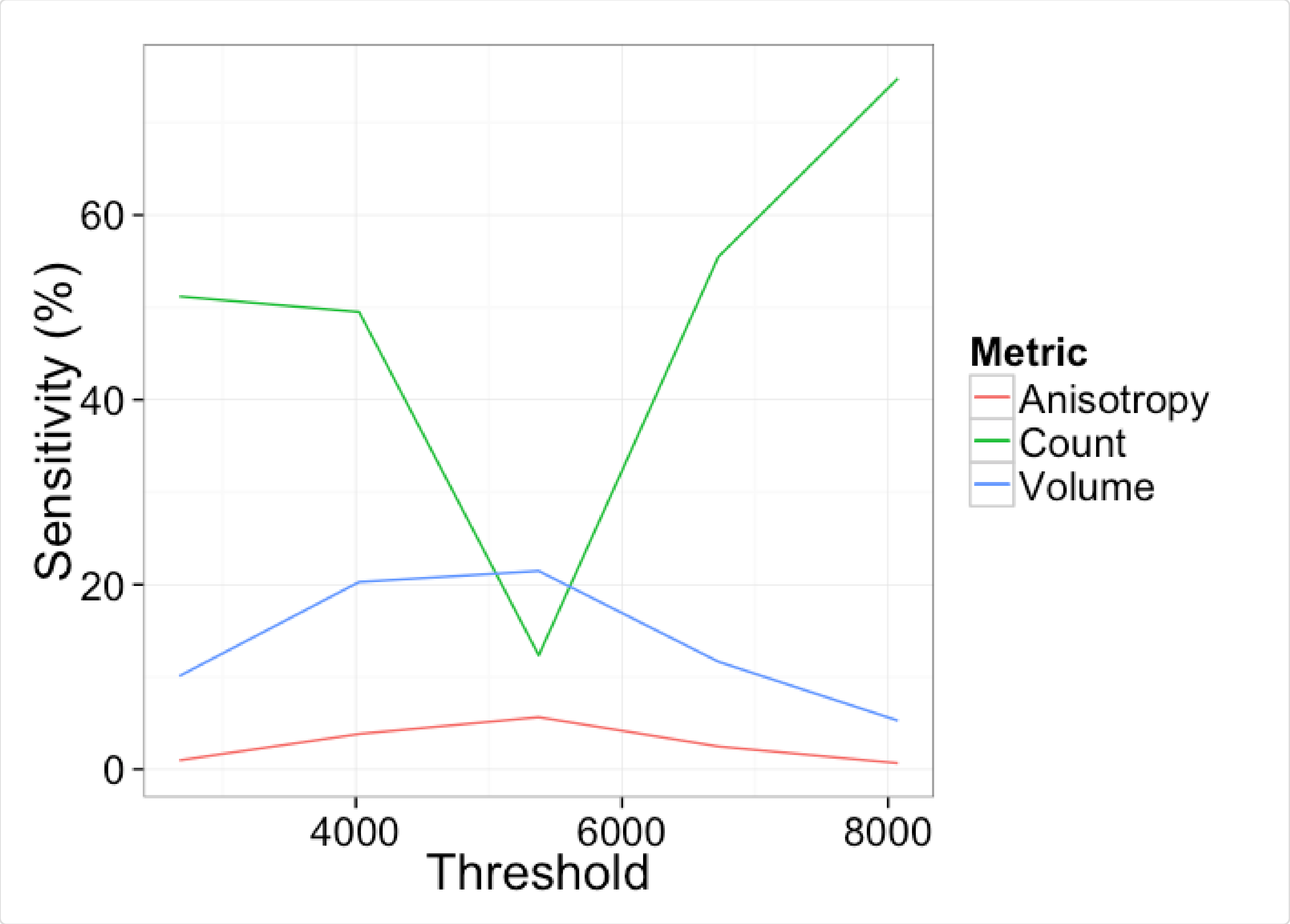
Sensitivity: Real Measurements

In this graph it is magnitude of the slope. The steeper the slope the more the metric changes given a small change in the parameter



Comparing Different Variables we see that the best (lowest) value for the count sensitivity is the highest for the volume and anisotropy.

+/- R Code



Which metric is more important?

Unit Testing

In computer programming, unit testing is a method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine if they are fit for use.

- Intuitively, one can view a unit as the smallest testable part of an application
- Unit testing is possible with every language
- Most (Java, C++, Matlab, R, Python) have built in support for automated testing and reporting

Unit Testing Continued

The first requirement for unit testing to work well is to have you tools divided up into small independent parts (functions)

- Each part can then be tested independently (unit testing)
  - If the tests are well done, units can be changed and tested independently
  - Makes upgrading or expanding tools *easy*
- The entire path can be tested (integration testing)
  - Catches mistakes in integration or *glue*

Ideally with realistic but simulated test data

- The utility of the testing is only as good as the tests you make

Example

- Given the following function `function vxCnt=countVoxs(inImage)`
- We can right the following tests
  - testEmpty2d

```
assert countVoxs(zeros(3,3) == 0
```

- testEmpty3d

```
assert countVoxs(zeros(3,3,3)) == 0
```

- testDiag3d

```
assert countVoxs(eye(3)) == 3
```

Unit Testing: Examples

- Given the following function `function shapeTable=shapeAnalysis(inImage)` We should decompose the task into sub-components
- `function clImage=componentLabel(inImage)`
- `function objInfo=analyzeObject(inObject)`
  - `function vxCnt=countVoxs(inObject)`
  - `function covMat=calculateCOV(inObject)`
  - `function shapeT=calcShapeT(covMat)`
  - `function angle=calcOrientation(shapeT)`
  - `function aniso=calcAnisotropy(shapeT)`

Unit Testing in ImageJ

Test Driven Programming

Test Driven programming is a style or approach to programming where the tests are written before the functional code. Like very concrete specifications. It is easy to estimate how much time is left since you can automatically see how many of the tests have been passed. You and your collaborators are clear on the utility of the system.

1. shapeAnalysis must give an anisotropy of 0 when we input a sphere
2. shapeAnalysis must give the center of volume within 0.5 pixels
3. shapeAnalysis must run on a 1000x1000 image in 30 seconds

Visualization

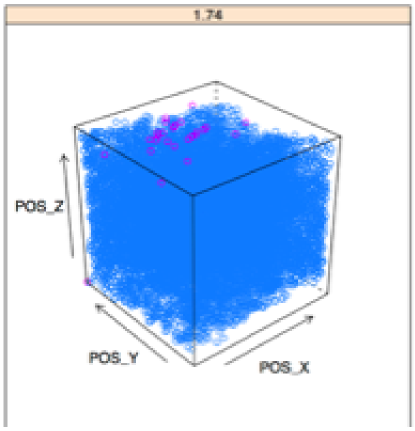
One of the biggest problems with *big* sciences is trying to visualize a lot of heterogeneous data.

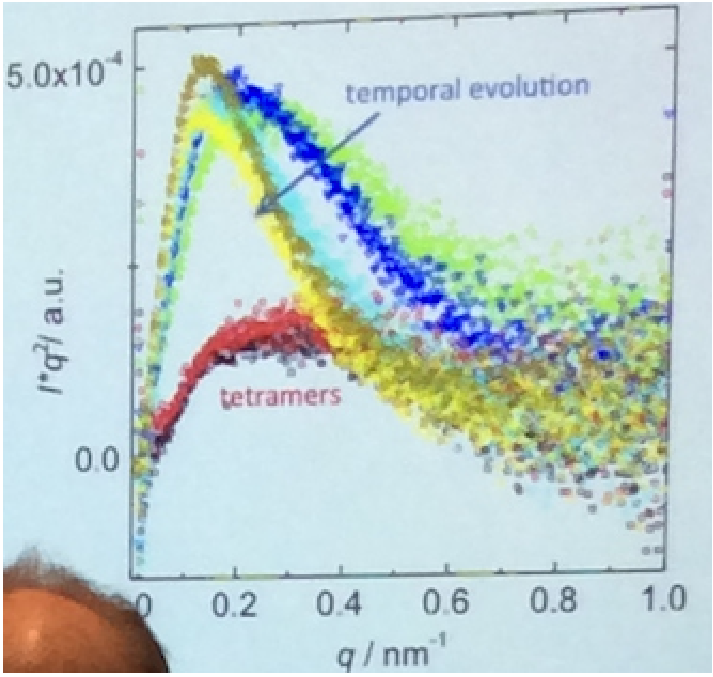
- Tables are difficult to interpret
- 3D Visualizations are very difficult to compare visually
- Contradictory necessity of simple single value results and all of the data to look for trends and find problems

Bad Graphs

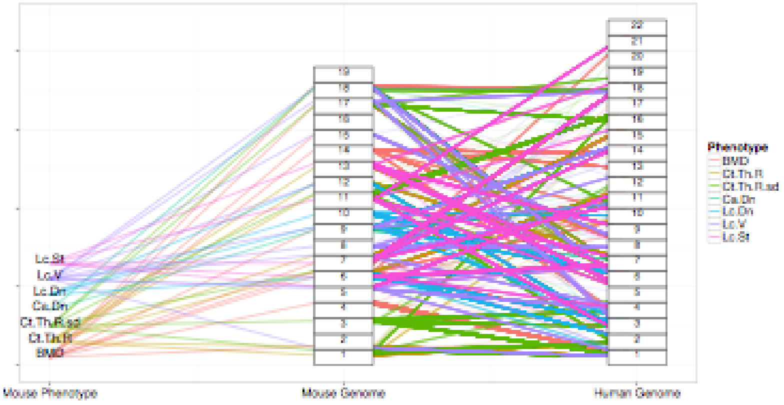
There are too many graphs which say

- ‘my data is very complicated’
- ‘I know how to use __ toolbox in Matlab/R/Mathematica’





- Most programs by default make poor plots
  - Good visualizations takes time

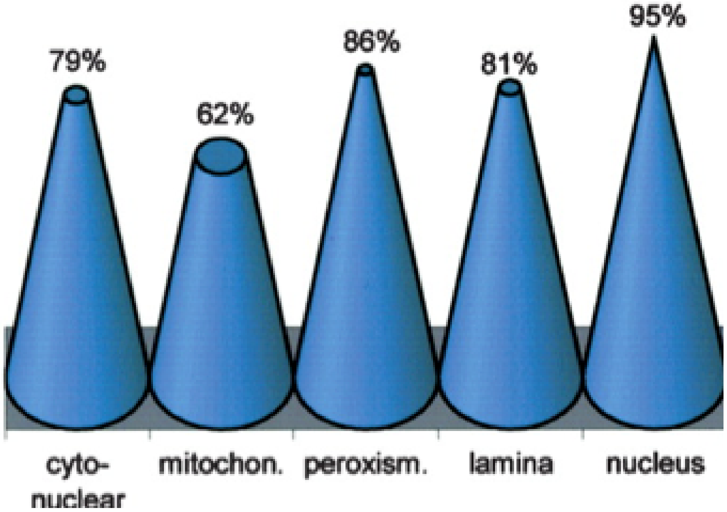


Key Ideas

1. What is my message?
2. Does the graphic communicate it clearly?
3. Is a graphic representation really necessary?
   -
4. Does every line / color serve a purpose?
   - Pretend ink is very expensive

Simple Rules

1. Never use 3D graphics when it can be avoided (unless you want to be deliberately misleading), our visual system is not well suited for comparing heights of different

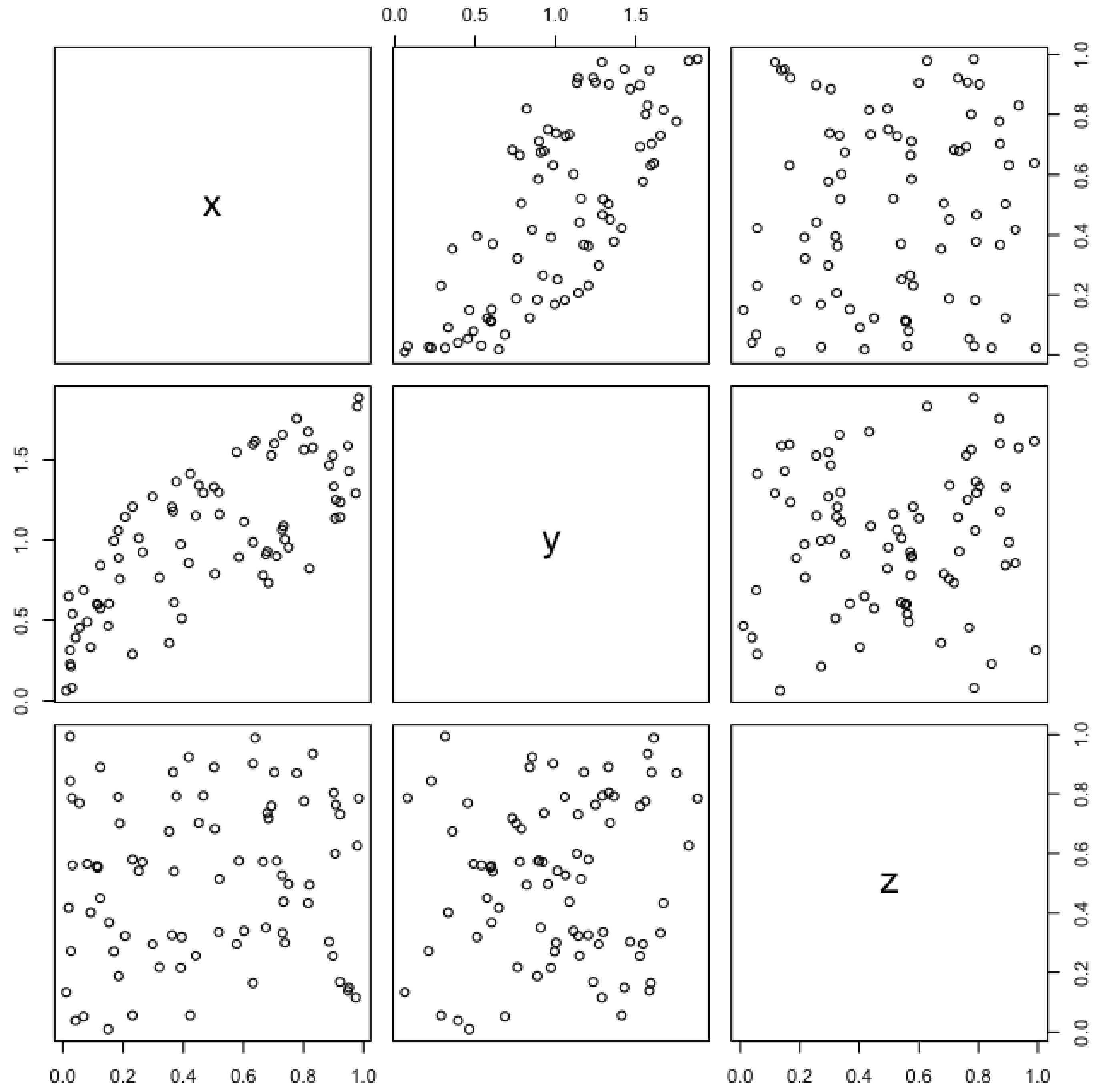


2. Pie charts can also be hard to interpret
3. Background color should almost always be white (not light gray)
4. Use color palettes adapted to human visual sensitivity

What is my message

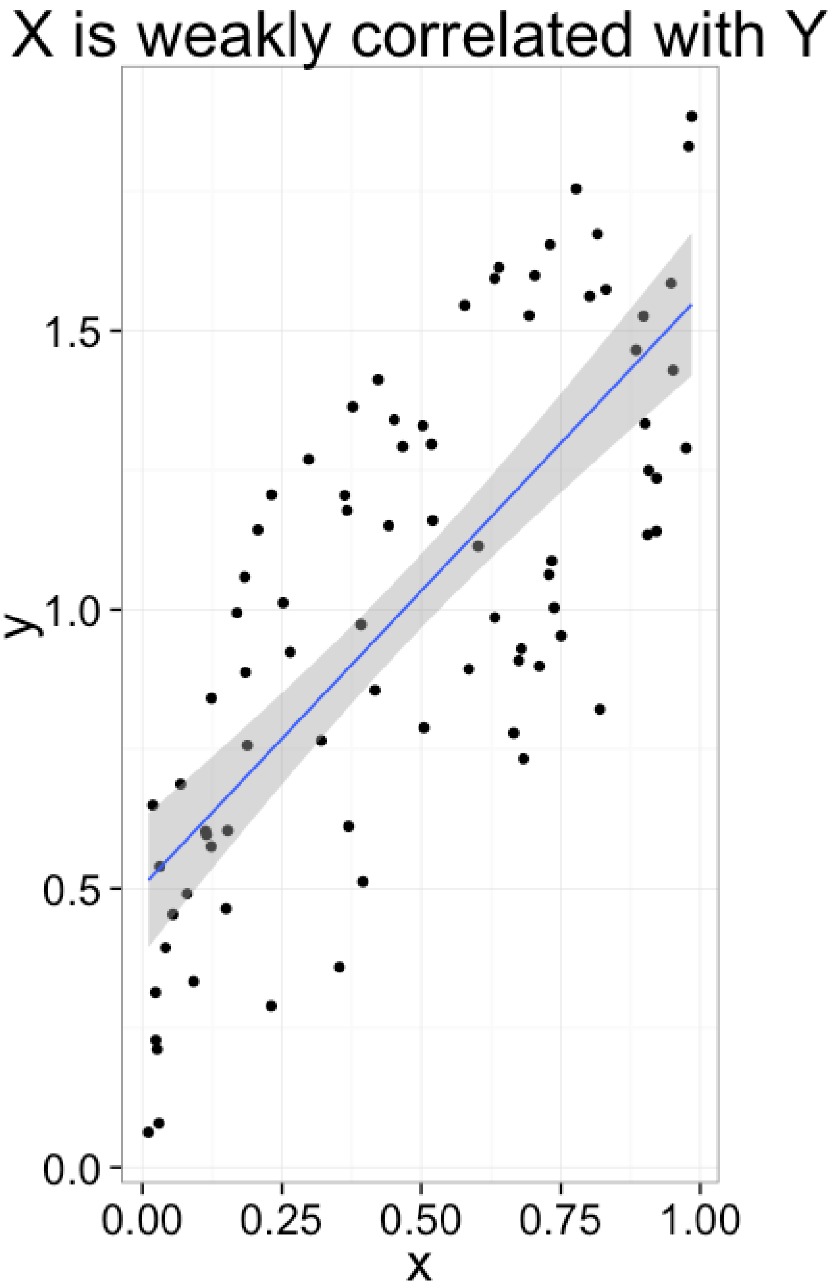
- Plots to "show the results" or "get a feeling" are usually not good

+/- R Code



- Focus on a single, simple message
    - X is a little bit correlated with Y

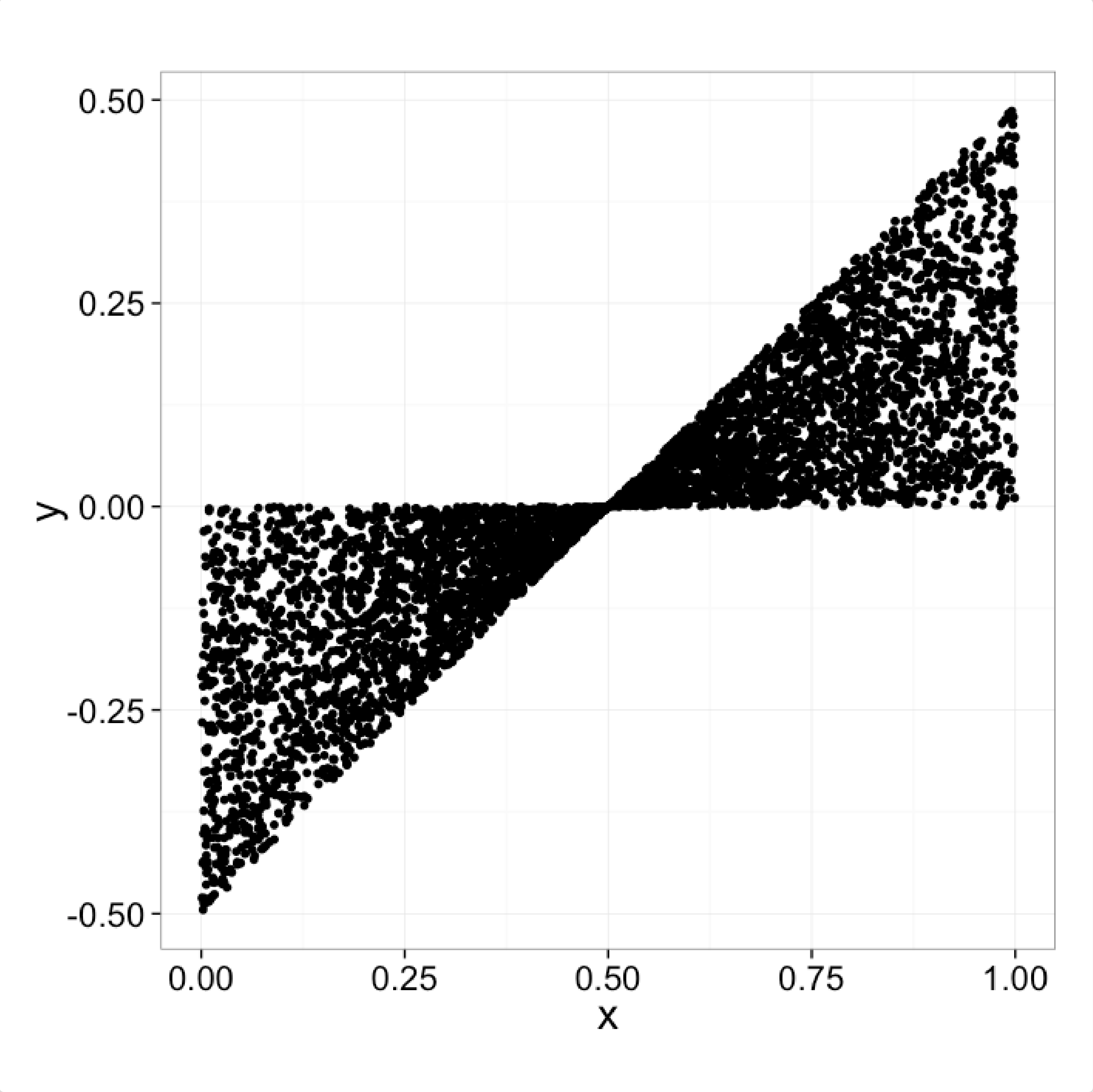
+/- R Code



Does my graphic communicate it clearly?

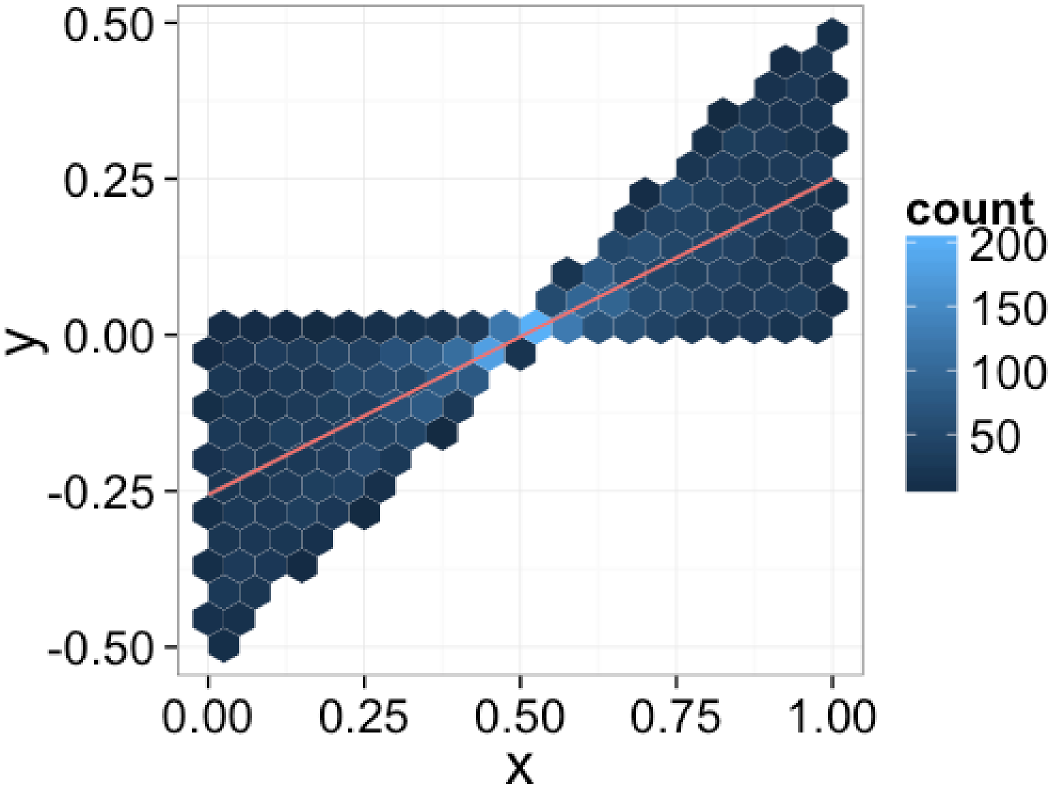
- Too much data makes it very difficult to derive a clear message

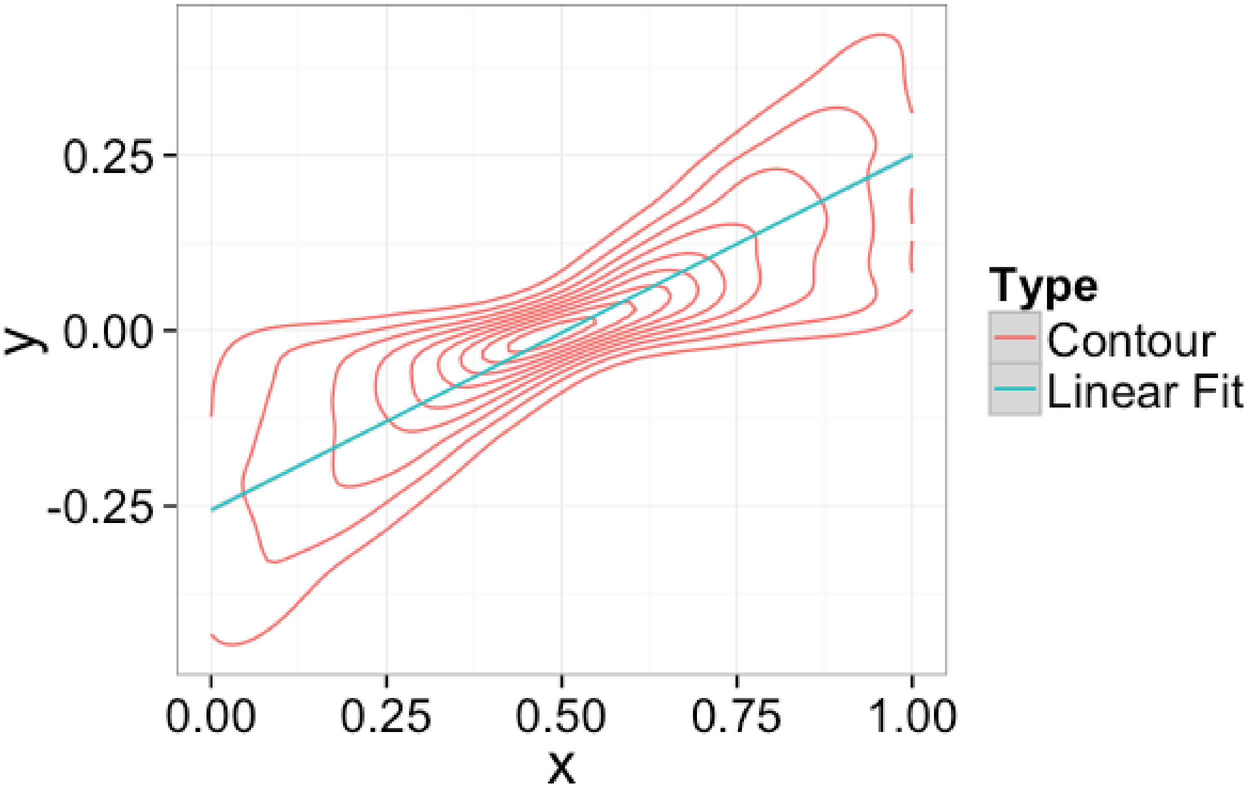
+/- R Code



- Filter and reduce information until it is extremely simple

+/- R Code


0.50
0.25
y 0.00
-0.25
-0.50
0.00
0.25
0.50
0.75
1.00
x
count
200
150
100
50


0.25
y 0.00
-0.25
0.00
0.25
0.50
0.75
1.00
x
Type
Contour
Linear Fit

Grammar of Graphics

- What is a grammar?
    - Set of rules for constructing and validating a sentence
    - Specifies the relationship and order between the words constituting the sentence
- How does this apply to graphics?
    - If we develop a consistent way of expressing graphics (sentences) in terms of elements (words) we can compose and decompose graphics easily

~~The most important modern work in graphical grammars is “The Grammar of Graphics” by Wilkinson, Anand, and Grossman (2005). This work built on earlier work by Bertin (1983) and proposed a grammar that can be used to describe and construct a wide range of statistical graphics.~~

H. Wickham. ggplot2: elegant graphics for data analysis. Springer New York, 2009.

Grammar Explained

Normally we think of plots in terms of some sort of data which is fed into a plot command that produces a picture

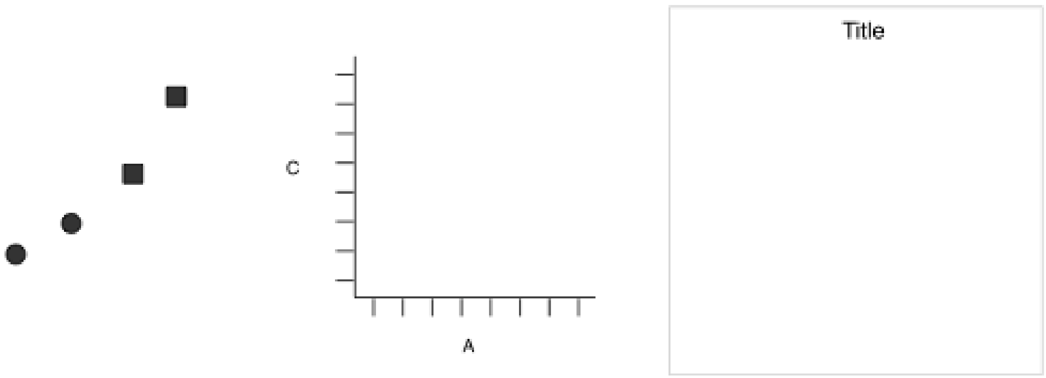
- In Excel you select a range and plot-type and click "Make"
- In Matlab you run `plot(xdata,ydata,color/shape)`

1. These produces entire graphics (sentences) or at least phrases in one go and thus abstract away from the idea of grammar.
2. If you spoke by finding entire sentences in a book it would be very ineffective, it is much better to build up word by word

Grammar

Separate the graph into its component parts

1. Data Mapping
    - $var1 \rightarrow x$, $var2 \rightarrow y$



1. Points
2. Axes / Coordinate System
3. Labels / Annotation

Construct graphics by focusing on each portion independently.

Wrapping up

- I am not a statistician
- This is not a statistics course
- If you have questions or concerns, Both ETHZ and Uni Zurich offer **free** consultation with real statisticians
    - They are rarely bearers of good news

- Simulations are very helpful
- Try and understand the tests you are performing