Learning to Adapt:

Monitoring Design and Data Analysis for Adaptive Management

Lindsay Mico CEP, GISP

Bright Eye Analytics

SER 2016

Agenda

- Intro 1:00 – 1:30
- Fetcho 1:30 – 2:00
- Design 2:00 – 3:00
- Analysis 3:00 – 4:00
- Q&A and/or project discussions 4:00 – 5:00

About me…

- Jedi Knight
- Master of Science
- Data Nerd
- Etc

Workshop Philosophy

- I’m going to walk through the key stages I use in designing and analyzing a monitoring program from a very <u>pragmatic</u> statistical perspective.
- By this I mean that although everything we discuss will be about generating quantitative results, there will be a minimum of discussion about math itself.
- We can do this because of the technology tools available to us
- With this approach we need to understand how and when to use something, but don’t need to know what goes under the hood.
- We will **not** talk about specific field protocols except as examples

Monitoring Philosophy

- Think deeply about what you **need** to know first
- Add in what you **want** only after
- If you can't be confident in answering what you **need** to know, consider changing the question or skipping the project
- Don't monitor just because you have a budget – its really hard to back calculate from budget to question (not that its easy the other way)
- Getting to the end of a monitoring project with no answers is VERY frustrating

(more) Monitoring Philosophy

- Translate the questions that motivate the monitoring into a quantifiable variables
- Identify a good protocol for generating estimates of those variables
- Focus on sources of error estimates in the protocol when selecting
- Consideration of protocol error, resources, and quantified questions makes this an optimization problem.
- Develop the key analytical steps before you collect the data
- This **includes writing the code** that will QA and

About the R programming language

- R is an open sourced software package designed for statistical computing
- Smarter people than us have built libraries for almost anything we can imagine
- The R ‘ecosystem’ is awesome and is why I prefer it – Rstudio, Markdown, Rpubs, Shiny, and CRAN
- It’s also a little bit ‘funky’ compared to other languages
- Python is another EXCELLENT alternative that I use regularly (albeit for other purposes)

Introducing Ken Fetcho

- Why did I invite Ken?
- A monitoring funder needs to know….
  - The study design is appropriate to the question
  - The study design can be extended or aggregated in the future
  - The results are calculated accurately
  - The results can be reproduced
  - Additional analysis can be applied to the original data
- This list also applies to a manager, property owner, regulator, or other stakeholder

Design

Design Section Goals

- Decide what to measure.
- Decide how much data to collect.
- Incorporate temporal considerations in study design to maximize power and satisfy multiple objectives.
- Build stratified, randomized spatial sampling plans with GRTS.
- Select appropriate reference conditions.
- Select appropriate protocols and sampling approach to leverage existing data.

First questions

- Make a list of the specific questions that you want to answer, in order of importance.
- Be clear what must be answered for the project to be a success
- Make a realistic estimate of the resources available to do monitoring
- Allocate resources to the questions, in order of importance, until they are exhausted
  - Be conservative in estimates of what can be accomplished
- This is an iterative process and should be done with professional design expertise as part of the team

Types of monitoring (in order of complexity and not mutually exclusive)

- Implementation - Did something happen? When and how?
  - Requires good geospatial data management
- Status - What does something look like today?
  - Requires a one time spatially balanced sample design
- Trend - How is something changing in space and time?
  - Requires a spatially balanced, multiyear sample design
- Effectiveness - Did my project result in the physical changes I intended?
  - Requires a paired sample, multi year design
- Did those physical changes effect the ecosystem in the way I intended?
  - Requires a paired sample, spatially balanced, multiyear sample design

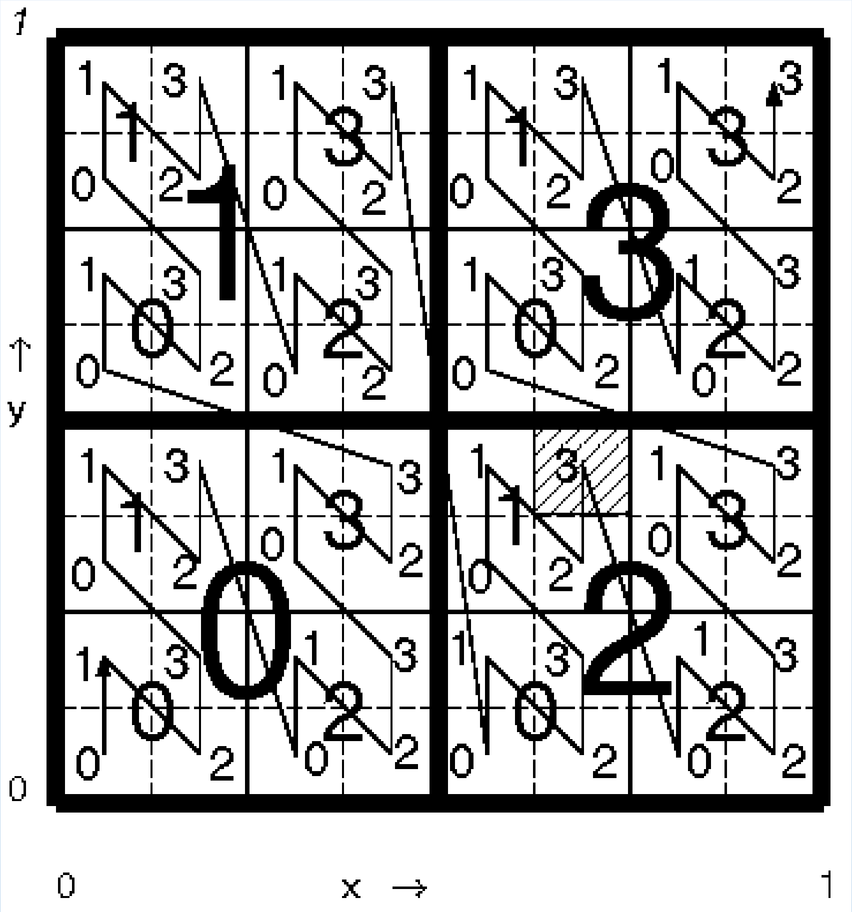
Spatially balanced samples – why?

- A spatially balanced sample helps us to accurately characterize a population that has substantial spatial auto-correlation
- By balancing the sample spatially we maximize the chance that we get a representative sample for a given n
- This is especially important because ecological monitoring can be very expensive on a per n basis
- I.e. we work in a data poor environment
- Note that the next few slides are drawn from a document Dr Tony Olsen prepared some time ago…

Spatially balanced samples – how?

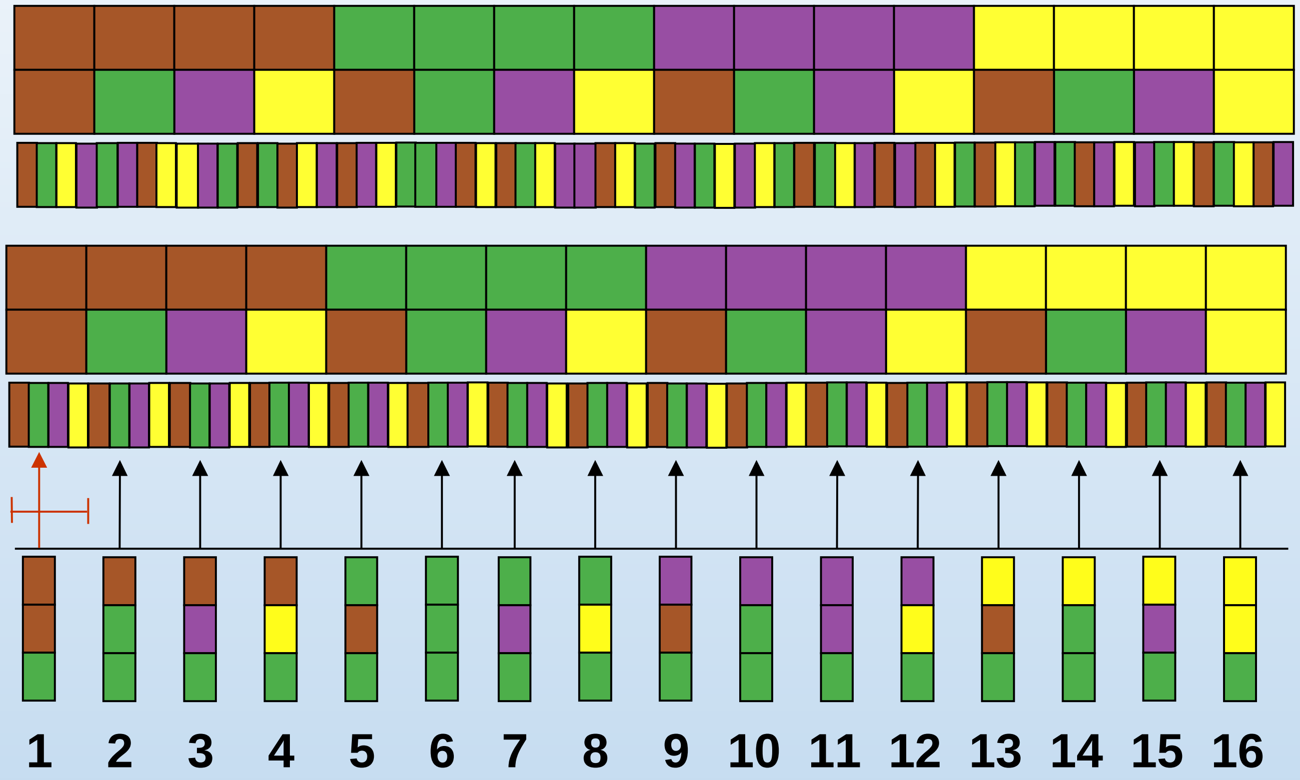
- GRTS – General Random Tessellation Stratified
- Every replication of the sample 'mimics' the spatial balance of the resource in question
- GRTS generates a hierarchical addressed grid, randomizes it, and then seeds a random start point before selecting sites in order.
- Input for this is a GIS file that is then imported into R.
- Most of the work is 1) theoretical (determining what your n is for each sub pop), and 2) GIS based to build your '***sample frame'***
- A sample frame is the GIS file that defines what can be

(more) spatially balanced random sampling



- GRTS breaks a spatial area into a grid, numbers it, and then assigns semi-random numbers to each part of the grid (addresses it)
- When all is said and done, you have a balanced sample where any contiguous sample remains random and balanced

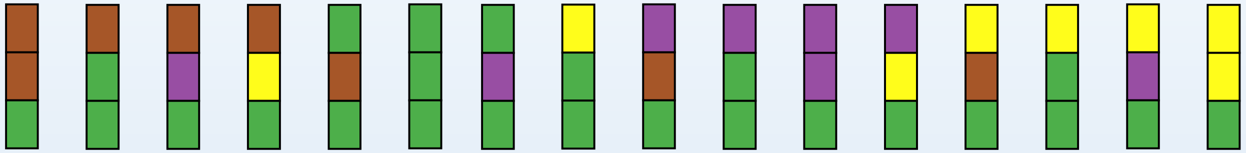
Selecting 16 Sample Points



Subdivide, Create Random Sequence, Assign Address & Colors
Sort Addresses
Random Starting point, Uniformly Sample Line
Assign Sequence Number to Each Point

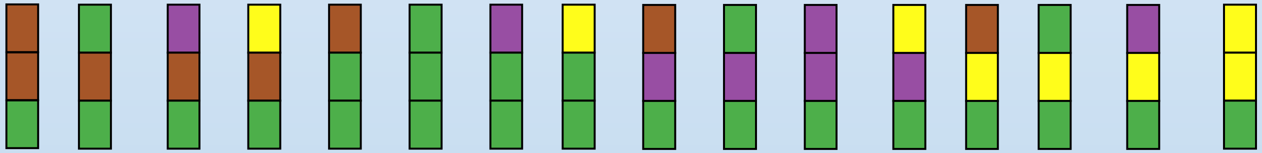
Reverse Hierarchical Order

Original Order



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Base4	00	01	02	03	10	11	12	13	20	21	22	23	30	31	32	33
Reverse Base4	00	10	20	30	01	11	21	31	02	12	22	32	03	13	23	33
Sort	00	01	02	03	10	11	12	13	20	21	22	23	30	31	32	33

RHO Site Number



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Create Base4 Addresses
Reverse Address Digits

Sort
Assign RHO Site Nos.

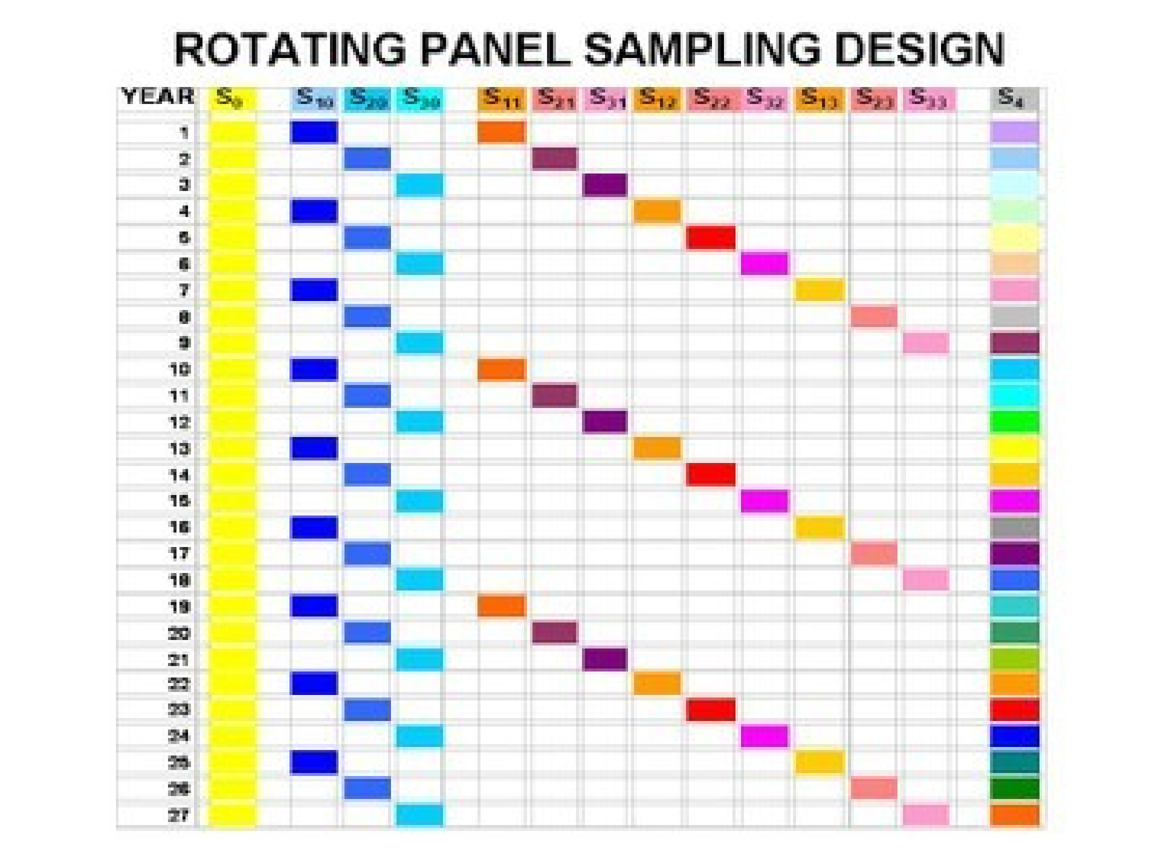
Other cool things about GRTS

- We can emphasize sub-populations in two ways –
  - We can require that the overall ‘n’ is allocated in a precise way (i.e. stratification)
  - We can also tweak the inclusion probabilities to get close to what we want, but still leave some randomness (I like this one better but both are ok)
- Remember that any contiguous sample is random and balanced
- Sites can be dropped, or the sample plan adjusted on the fly (by an expert only please!) so long as there is no bias and analysis takes it into account
- You can add two GRTS samples together and they remain random and balanced (mostly)
- Built for aquatic resources but good for any spatial sample

Sampling for trend

- The simplest approach is to simply sample the restoration project across multiple years
  - But this doesn’t tell us if the change is due to restoration or other unrelated factors
- So we add in a control, also sampled across time
  - But this doesn’t let us keep track of changes across the landscape
- Panel designs allow us to do a lot with a little
  - It’s a compromise between trend and status

Partially augmented serially alternating



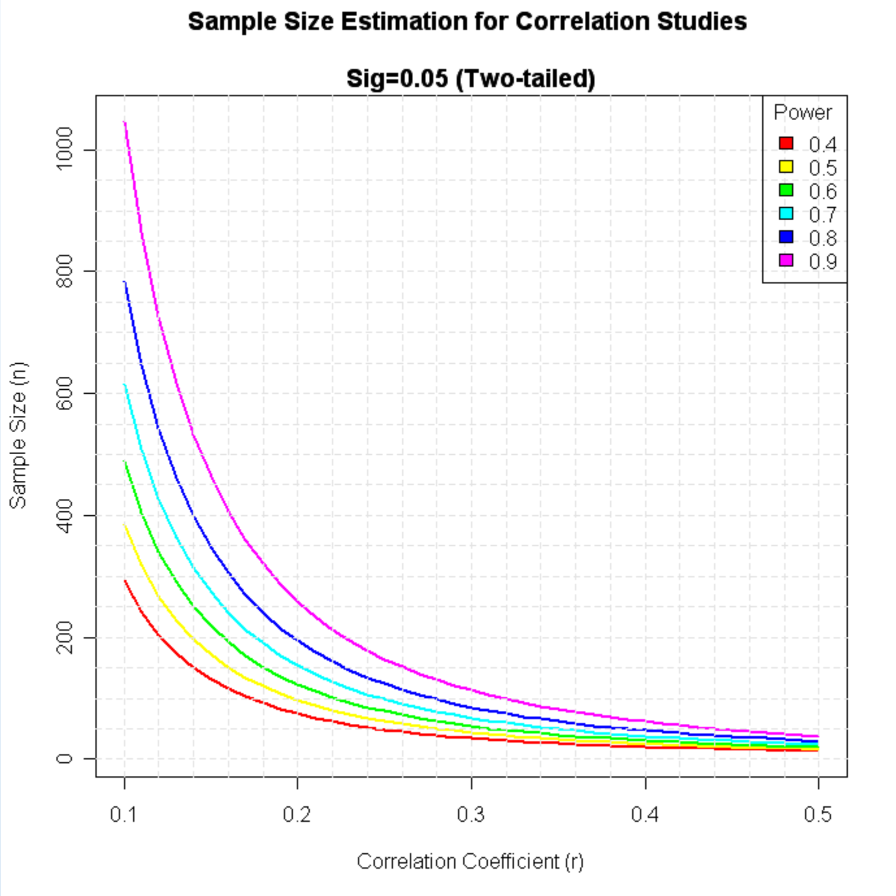
- The more yearly revisits the better the site specific trend estimate (for those sites)
- The more we spread out across the landscape, the better the landscape scale trend estimate (but we lose power and introduce noise)
- Its <u>really</u> easy to build these with GRTS if you get

How much to sample – Power Analysis

- Allows us to determine the sample size required to detect an effect of a given size with a given degree of confidence. Conversely, it allows us to determine the probability of detecting an effect of a given size with a given level of confidence, under sample size constraints.
- The following **four quantities** have an intimate relationship:
  - sample size (This is what we can control)
  - effect size (We can’t control this)
  - significance level = P(Type I error) = probability of finding an effect that is not there (This is a decision we make during design)
  - power = 1 - P(Type II error) = probability of finding an effect that is there (another decision we make)
  - Given any three, we can determine the fourth.

The pain of a data poor environment…

- As you can see to the right, its clear that unless we have a big effect, it is hard to find for the type of sample sizes common in restoration monitoring
- The take home message is to focus on relatively large changes in your design



A brief discussion of reference conditions

- Restoration vs reference is a specialized case of a ‘paired sample’
- Paired sample designs are intended to isolate the effects of an intervention from other factors that may influence the variable of interest
- If we sample restoration and reference across time, we effectively have a double paired sample
- A control site is one that is close to identical to our restoration site, except that no restoration has been done
  - Usually this is accomplished by selecting a site in physical proximity
- A reference site reflects the characteristics that we ***want*** to see in our restoration sites
  - This can be hand selected or based on a GIS disturbance metric (e.g. road crossings)

Design questions?

Analysis

Analysis Section Goals

- Build a reproducible analytical pipeline with the R software package.
- Utilize basic QA/QC and data management processes and considerations.
- Move beyond t-testing with Bayesian analysis, relative risk, and other indicators of success.
- Analyze trend and control for landscape variables.
- Make your data available and accessible with ESRI geo-databases and Github.
- Analyze and express uncertainty in your results.

From the field to the computer

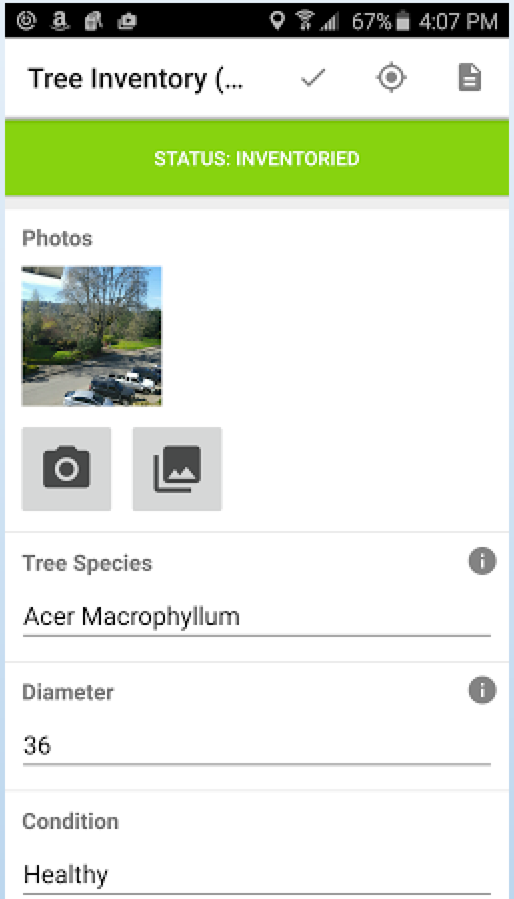
- Getting the data from the field to the computer is a non trivial task, can be a substantial constraint in the analytical pipeline
- Building a good technology interface at this stage pays big dividends –
  - QA/QC “on the fly” by alerting field staff to unreasonable or missing values
  - Data can be transferred digitally, often times immediately following data collection
  - Data import and pre-processing can happen almost instantaneously, often before the crew even gets back to the office
- With modern smart phone and GPS technology there is no reason not to build a data entry interface

Things to consider in building your interface

- Although in some cases you may need an expensive and integrated GPS unit (e.g. Trimble and the like) for most purposes as a monitoring professional this level of detail is unnecessary.
- Ideally every interface should include the following –
  - Flexible constraints on what is allowed for each field
  - Export as a csv, txt, or other machine readable table
  - Geospatial integration (at least be able to drop a pin)
- There are MANY options out there – I picked Fulcrum as an example
- Beta test with your crew – and be prepared for complaints

Fulcrum Example

- Took less than 15 minutes to sign up and deploy a spatially enabled form
- Lots of great templates
- Exports as a csv
- Even if you don’t have good cellular coverage, you should still take as much digital data as possible
- http://www.fulcrumapp.com/



Building the analytical pipeline

- One of the most immediate advantages to 1) taking good digital data and 2) using a code based analytical pipeline (e.g. R) is that data can be processed almost instantaneously.
- This is the upside of being in a data ‘poor’ environment
- **You should never need to open Excel**. Spreadsheets are not your friend!!!
- Time required to analyze a given dataset increases ~exponentially in a spreadsheet as the dataset grows - Big O(c^n)
- Time required to analyze data with code is effectively constant - Big O(c)

Automate and test your pipeline early

- This is a two part, iterative process
- First simulate data in R
  - This gives you a ‘stub’ to test the pipeline with
  - It allows you to specify the signal and the noise based on the power analysis you did in the Design phase
  - This helps to verify that the code does what you think it does
  - This also helps to validate that you have the power you think you do
- Second write code to process and analyze the data
- Process the data, review results, adjust and repeat as needed

Data preprocessing

- Import the data into R from your txt or csv file
- Preprocess the data –
  - Change names
  - Reorder the fields
  - Extract just what you need
  - Format into a ‘dataframe’
- Although you can write out data along the way as you preprocess, this is not necessary (and in fact is less than ideal)
- QA/QC your data….

QA/QC your data and deal with problems

- Ideally you have caught most of the data entry problems at the front end by building a good interface but something **always** seems to slip through the cracks
- If you do end up with **a lot** of bad or questionable data (e.g. because of lab problems) I suggest you go forward with the analysis as planned but be very cautious when presenting the results
- Think **very carefully** about using volunteers for monitoring. I strongly discourage this for serious monitoring work (outreach and education is a different story)
- QA/QC is why you need to avoid spreadsheets at all

Dealing with missing data

- Random missing values can be imputed
- Non-detects can be imputed based on a distribution
- R can often do this automatically for you
  - NA.RM = TRUE
- Imputing missing data should be done with care – it can give you a false sense of security
- Be sure to adjust your n value when testing later on

Calculate your metrics

- This is what I call ‘processing’, and is probably the part most people are comfortable with
- Although I’m not going to spend much time on this today, this is possibly the most useful part of using a code based pipeline
- This is also an area that needs careful unit testing ahead of time
- This is another HUGE advantage to using code – it is very easy to test and hard to break once testing is done

Weighted data and NBV

- At the stage of analysis, it is essential to reweight the data points to reflect the extent of the resource (based on the sample frame)
  - For example, if you have ten evenly spaced sites drawn from a 100 acre sample frame, each site has a weight of 10
- Weighting ensures that the estimates reflect the population
- Spatially balanced sample designs with unequal design probabilities are tricky to generate variance estimates for with normal techniques (e.g. Horvitz Thompson)
- Neighborhood based variance (NBV) is another tool from the EPA to account for this, and generates smaller **and** more accurate variance estimates (as shown with simulation)
- The math for NBV gets pretty intense pretty fast here…

Compare populations

- Depending on what you are interested in we can start with the standard classical statistical tests
  - T-testing, chi square, and the like for comparing two samples
  - ANOVA for more than two groups (and for interactions)
- Data (especially with a small n) rarely clearly conforms to the assumptions of parametric methods so you should first try to transform the data, then look at non-parametric tests
- Despite their prevalence, I find these to be useful primarily for exploratory purposes only
- Hypothesis testing over-emphasizes sample size and under emphasizes effect. This is a big problem in a data poor environment
- Always put the results of significance testing on Occam’s razor

Look at (and compare) trend

- Linear regression is always the right place to start
- Even in cases where the response is highly non-linear, if the trend is strong enough you will still see a linear trend
- When including multiple regressor there is a fine line between too few and too many
  - Too few and you may bias your results
  - Too many and you may introduce unneeded variance
- Dimension reduction may help identify what to include as a regressor

Relative Risk

- Relative risk or risk ratio (RR) is the ratio of the probability of an event occurring (for example, developing a disease, being injured) in an exposed group to the probability of the event occurring in a comparison, non-exposed group.
- Relative risk includes two important features: (i) a comparison of risk between two "exposures" puts risks in context, and (ii) "exposure" is ensured by having proper denominators for each group representing the exposure

	Outcome	No Outcome
Exposed	a	b
Unexposed	c	d

$$\text{Relative risk} = \frac{\text{Incidence in exposed}}{\text{Incidence in unexposed}} = \frac{a/a+b}{c/c+d}$$

Bayesian estimates

- Bayesian estimation is a fundamentally different way of looking at data
- Bayes' theorem is a way to work out the likelihood of something in the face of some particular piece, or pieces, of evidence
- The theorem is usually written like this:
  - $p(A|B) = p(B|A)\ p(A) / p(B)$
- Instead of telling us about the probability of sampling error (the frequentist p value) Bayesian estimators tells us about the probability that one thing can predict another (which is close but not equal to cause and effect)
- Bayesian math also gets pretty intense pretty fast but its also really useful

Simple Bayesian Example

- How likely is it that a stream is impaired if sediment levels are high Ie p(impairment|high sediment) or p(A|B)
  - We need the other half of the equation from the previous slide
  - p(impairment) = 10% of all streams are impaired
  - p(sediment) = 50% of streams have high sediment
  - p(sediment|impairment) = 90% of streams that are impaired have high sediment
- So p(impairment|high sediment) = (0.9*0.1)/0.5 = 0.18 = 18%
- This is a much better way to do inference than t-testing the sediment values for impaired vs non-impaired streams

Open data & reproducible research

- Data should not be stuck in a report. No one reads them anyways
- Geodatabases are an excellent way to store raw field data in association with its geospatial component
- Building your analytical pipeline in R makes it possible to reproduce the analysis exactly from the raw data
  - As such you do not need to store process and output files, just figures and tables
- R Markdown is an easy way to blend code and text
- GitHub is a free and easy to use code sharing and VCS web tool/site to which you can add your code (and data if its not too huge)

A story about error, dirt, and t - testing

- Once upon a time a young man wanted to validate a 303(d) listing….

Fin