Estimating power and sample size to detect trends in CPUE

Carl Schwarz

2017-05-13

Contents

1 Introduction

Often, it is of interest to know:

> How many years of sampling and how many sites need to be sampled in a watershed to detect a 50% increase in CPUE over 10 years?

This is known as a power analysis or a sample size analysis. This document assumes that the general concepts of a power analysis are known and will not go into great detail about these. Briefly, the power of a study is the probability that an effect of a certain magnitude (e.g., a 50% increase in CPUE over 10 years) is detected given the sample size (i.e., number of sites sampled each year and the number of years of sampling). Power is often computed using $\alpha = 0.05$ with a target power of at least 80% to detect the effect of interest.

2 Sources of variation

There are many sources of variation in the data when looking at trends, two of which have a direct impact on the ability to detect trends (the power). Consider a study where a wateshed is repeatedly measured over time with the same sites being measured each year. There are three levels of variation in this study:

- site-to-site variation. Some sites generally have higher densities than other sites and so the CPUE will tend to be consistently above/below the average CPUE whenever this site is measured.
- year-to-variation. Year-specific effects (also known as process effects) may push all the CPUE readings higher or lower in a particular year. For example, suppose that water temperature affects the CPUE with warmer water temperatures generally having higher CPUEs. Then in warm years, all sites will tend to have warmer water and so all sites will tend to have higher CPUEs compared to colder years.
- residual or random variation. If the exact same site was resurveyed, you will not get exactly the same CPUE despite your best efforts. This random variation could be due to site-year local effects such as the operator, the outside temperature when the sample was taken etc.

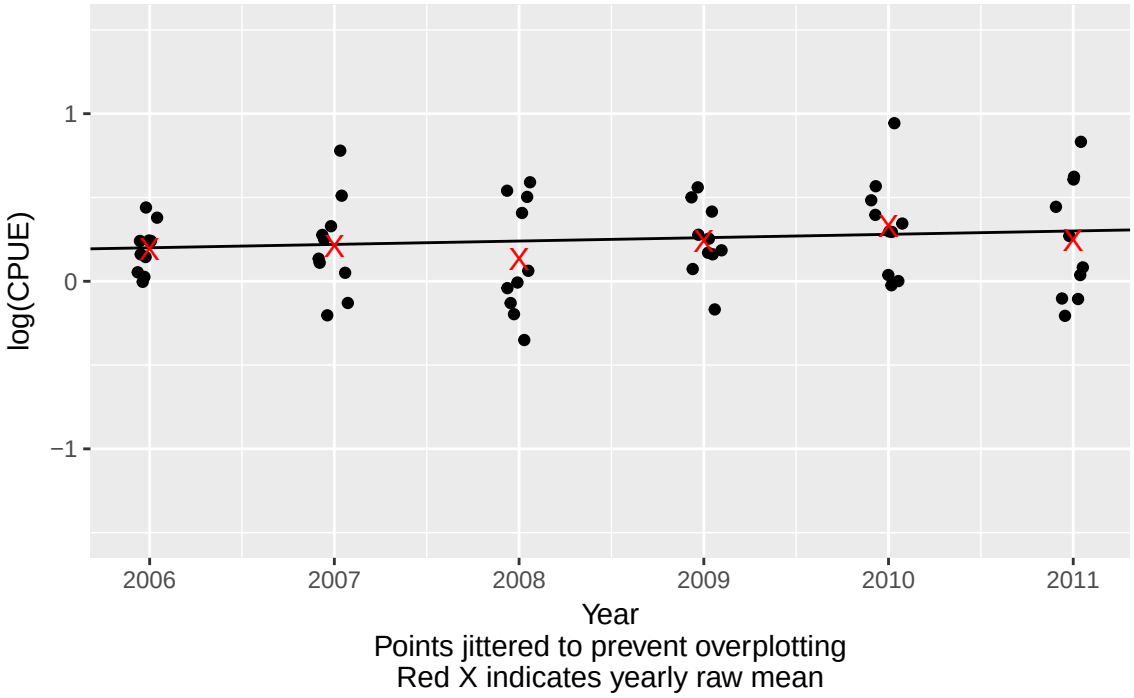
The magnitude of the year-to-year variation (a.k.a. process variation) is measured by the PROCESS standard deviation; and the magnitude of the variation of measurements among sites within a year (after accouting for site effects) is measured by the SAMPLING standard deviation. These two sources of variation, along with the number of sites sampled each year, and the number of years of sampling are the determinants of the power to detect a trend line.

3 Why is the process standard deviation important?

Whenever fitting trends over time, you need to be concerned about the year-specific effects (also known as process effects). Often, a naive analysis will ignore the process effects when fitting a trend over time which leads to erroneous inference.

In particular, year-specific effects (process effects) force the CPUE values in year to be higher or lower en masse, which is a violation of the key assumption of a regression analysis. For example, regression analysis assumes that the data in each year are always centered about the regression line as shown below:

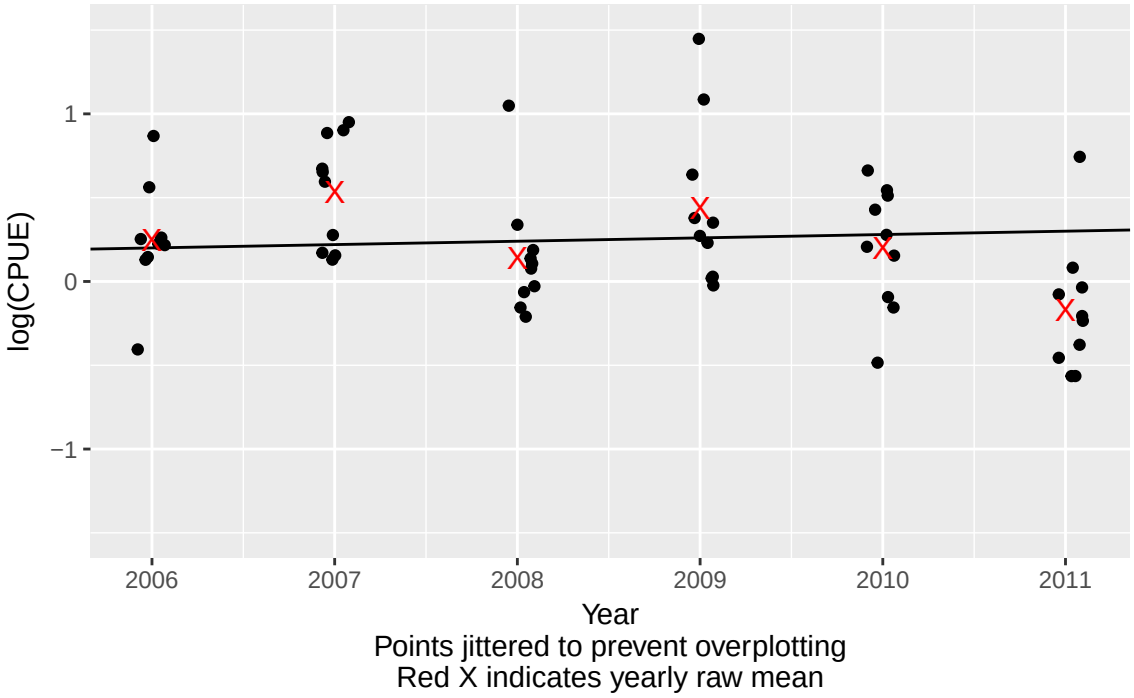
Figure 1: Trend with NO process effects



In the absence of process error, the yearly raw means will be close to the trend line because the data in each year is centered about the trend.

However, when process standard deviation is large, the points in each year will tend to be higher or lower in years with positive or negative year-specific effects as illustrated below:

Figure 2: Trend with large process effects



Now the yearly means may be far from the trend line and reflect the influence of the year-specific effects (the

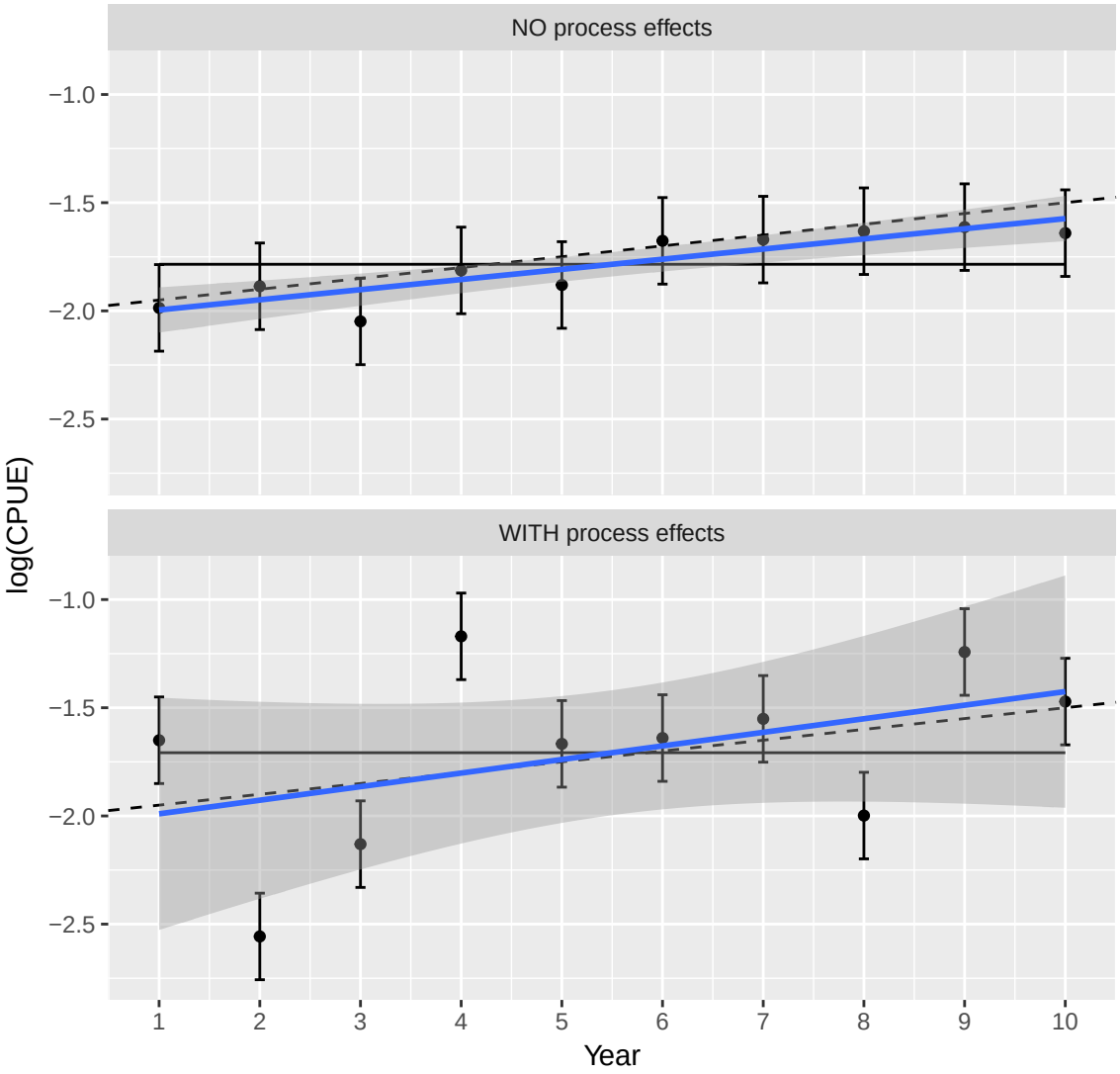
process effects).

Here the variation of the yearly means about the trend line represents the process standard deviation and the variation of the individual points about the yearly mean represents the sampling standard deviation.

In my experience, year-specific effects can often be quite large and, in some circumstances, be larger than sampling effects, i.e. the process standard deviation is larger than the sampling standard deviation.

As an illustration of the impact of process error, consider Figure 3.

Figure 3: Illustration of impacts of process effects



In the top panel of Figure 3, there are no year-specific effects (no process effects) and the response changes over time based on the underlying trend line shown as a dashed line. Only sampling variation is present so 95% of the confidence intervals for the mean response in a year will overlap with the underlying trend line. The fitted line will be close to the true underlying trend line (solid line). The uncertainty is small about the overall trend (the shaded region), and there is clear evidence that the trend is different from a 0 trend (which is shown by the horizontal line).

In the bottom panel of Figure 3, the year-specific effects (process effects) add extra variation to the underlying response each year due to effects such as weather, food etc. Now, the 95% confidence intervals for mean response still provide valid estimates for the yearly mean response values, but now may not overlap the true

underling trend (in dashed). The fitted line will still be unbiased for the true trend (solid line), but the extra variation makes the uncertainty in the fitted line (the shaded region) much larger and now there is no evidence that the trend line differs from 0.

If there is substantial year-specific effects (process effects), then sampling more sites in a year will NOT be helpful. Sampling more sites in a year will shrink the size of the confidence intervals for each year in the bottom panel of Figure 3, but has NO impact on the process effect and so the uncertainty about the fitted line will only be reduced slightly. In cases of substantial process effects, the limiting factor for detecting trends is likely to be the total years of sampling and not the number of sites/year that are sampled.

4 Why did we fit the regression line on the (natural) logarithmic scale?

Because we are interested in proportional changes (e.g. a 50% increase over 5 years), a linear mixed model is fit on the (natural) logarithmic scale. For example, suppose that a (mixed model) regression line was fit on the (natural) logarithmic scale and the fitted line had the form $\log(CPUE) = 0.3 + 0.02(Year)$. This implies that the $\log(CPUE)_{t+1} - \log(CPUE)_t = 0.02$ because a slope of 0.02 implies that the mean value of $\log(CPUE)$ changes by 0.02 for each additional year in the study. This can be reexpressed as:

$$\log(\frac{CPUE_{t+1}}{CPUE_t}) = 0.02$$

or

$$\frac{CPUE_{t+1}}{CPUE_t} = \exp(0.02) = 1.02$$

or a 2% increase per year in the CPUE.

Because the data was fit on the (natural) logarithmic scale, the process and sampling standard deviations have a simple interpretation. For example, if the sampling standard deviation was 0.3, then the standard deviation of the original CPUE is approximatately 30% of the mean, i.e. the sampling standard deviation on the (natural) logarithmic scale is an estimate of the coefficient of variation on the original scale.

The logarithm of the CPUE is not defined if the CPUE = 0. The usual convention is to add 1/2 of the smallest postive values of CPUE before taking the logarithm. For example, if the smallest positive CPUE for a system was 0.32, then $0.16 = 0.5 \times 0.32$ would be added to all of the CPUE values before taking the (natural) logarithm and fitting the trend line.

5 How to fit a trend line when process effects are present.

If process effects are ignored in the fitting process, then estimates are still unbiased, but the reported standard error are too small (so the reported uncertainty in the slope is too small), reported confidence intervals are too narrow, and the reported p-value is too small (leading to too many false positive results where you believe you have detected a trend, when in fact, there is no evidence of a trend).

For example, consider the data from Figure 2 where a new site is measured every year. A naive trend line was fit using the *lm()* function:

```
naive.fit <- lm(logCPUE ~ Year, data=plotdata.PE)
summary(naive.fit)$coefficients
```

```
##                 Estimate  Std. Error   t value   Pr(>|t|)
## (Intercept) 160.66702192 62.04116263  2.589684 0.01212651
## Year         -0.07987644  0.03088929 -2.585894 0.01224641
```

Here the (naive) fitted line has a slope of -0.08 (SE 0.03) and a 95% confidence interval for the slope between -0.14 and -0.02. The reported p-value is small ($p = 0.012$) and so you would believe that you detected an effect.

There are two ways to properly fit a trend line in the presence of process effects. Both methods are exactly equivalent if the design is balanced (i.e. the same number of observations taken each year).

In the first method, the trend line is fit to the AVERAGE yearly values. So the data in Figure 2 is reduced to one number per year for a total of 6 values (6 years of data) and the trend line is fit to the averages. The averaging process can be done using the *ddply()* function from the *plyr* package:

```
yearly.means.PE <- plyr::ddply(plotdata.PE, "Year", plyr::summarize, logCPUE.mean = mean(logCPUE))
yearly.means.PE
```

```
##   Year logCPUE.mean
## 1 2006    0.2493573
## 2 2007    0.5393409
## 3 2008    0.1437039
## 4 2009    0.4425834
## 5 2010    0.2052184
## 6 2011   -0.1690802
```

and then a regression fit is done on the means:

```
avg.fit <- lm(logCPUE.mean ~ Year, data=yearly.means.PE)
summary(avg.fit)$coefficients
```

```
##                 Estimate   Std. Error   t value  Pr(>|t|)
## (Intercept) 160.66702192 106.48806340  1.508780 0.2058536
## Year         -0.07987644   0.05301868 -1.506572 0.2063913
```

Here the slope of the fitted line on the averages has a value of -0.08 (SE 0.05) and a 95% confidence interval for the slope between -0.23 and 0.07. The estimated slope is the same as in the naive fit, but the (correct) reported standard error is much larger. The reported p-value from the fit on the average ($p = 0.206$) is quite large and there is no evidence of a trend, contrary to the conclusions from the naive fit.

In the second method of dealing with process effects, a mixed linear model is fit where a factor term (*YearC* below) represents the year-specific (process) effects:

```
plotdata.PE$YearC <- factor(plotdata.PE$Year)
mixed.fit <- lmerTest::lmer(logCPUE ~ Year + (1|YearC), data=plotdata.PE)
summary(mixed.fit)$coefficients
```

```
##                 Estimate   Std. Error       df   t value  Pr(>|t|)
## (Intercept) 160.66702202 106.48797177 4.009495  1.508781 0.2056885
## Year         -0.07987644   0.05301864 4.009495 -1.506573 0.2062262
```

The (correct) estimated slope, its SE, and p value are identical to the respective terms from the fit on the averages because the design was balanced. If the design were not balanced, e.g. unequal numbers of observations in each year, the fit on the averages will only be approximate and may differ (usually slightly) from the results of the mixed model.

6 Estimating the process and sampling standard deviations.

A key advantage of the using the second (mixed linear model) method is that estimates of the process and sampling standard deviation can be extracted which are then used in a power/sample size analysis. This are extracted from the fit object in *R* using:

```
vc <- data.frame(VarCorr(mixed.fit))
vc[,c("grp","sdcor")]
```

```
##        grp     sdcor
## 1    YearC 0.1868194
## 2 Residual 0.3780290
```

In this case the estimated sampling standard deviation (on the logarithmic scale) is 0.38 and the estimated process standard deviation (on the logarithmic scale) is 0.19. Because the analysis was done on the (natural) logarithmic scale, these have a simple interpretation as the coefficient of variation for the individual measurements and the yearly means respectively.

7 Extracting necessary information for a power analysis

In order to estimate the sampling and process standard deviation. at least two (and preferable 3+) years of data should be available which record the site name, the year, and the CPUE at that site in that year. This information is currently stored in a standard format for entry to FWIS – it should either be extracted from FWIS or read from spreadsheet with the information. I have developed an *R* function that extracts the information from the FWIS spreadsheets, and estimates that CPUE (fish captured per 300 m) for each species of interest.

There are three cases to consider

1. **Single year of data.** In this case, only an estimate of the sampling standard deviation is possible. The process standard deviation cannot be estimated because process standard deviation operates on the year-to-year level and there are no replicate years. The sampling standard deviation is simply the standard deviation of the (natural) logarithm of the CPUE.

2. **Two years of data.** In this case, it is assumed that there was NO trend over the two years. If new sites were measured in each year, then it is impossible to separate the site-to-site standard deviation and the sampling standard deviation. The mixed linear model does not have trend term, and is

   $$\log(CPUE) \sim (1|YearC)$$

   where *YearC* is an *R* factor variable for the year of the study.
   If some of the sites are measured in both years, then the above model is modified to capture the site-to-site variation:

   $$\log(CPUE) \sim (1|Year) + (1|SiteF)$$

   where *SiteF* is an *R* factor variable for the sites.

3. **Three or more years of data**. In this case, a trend is fit to the data to remove its effect before finding the variance components. The mixed linear model when new sites are measured each year is:

   $$\log(CPUE) \sim Year + (1|YearC)$$

   The mixed linear model when some sites are measured in more than one year is

   $$\log(CPUE) \sim Year + (1|YearC) + (1|SiteF)$$

In the second and third cases, the variance components can be extracted after the fit using the methods presented earlier.

As will be seen later, the process standard deviaton is often the limiting factor in determing the power to detect a trend. Some of the process effects may be attributable to other variables that also vary at the yearly level such as stream temperature, conductivity, etc. The additional covariates can be added to the models for cases 2 and 3 above to try and reduce the size of the process standard deviation.

8 (Finally) estimating the power and sample size

Once the sampling and process standard deviatons (after the analysis of the CPUE on the logarithmic scale) are available, we can now examine how the power to detect effects varies by the size of the effect (the trend), the number of sites sampled per year, and the number of years of sampling (and the pattern of sampling years).

We first translate the desired change over xx years into the yearly proportional change (the trend to be detected). For example, if we wish to detect a 50% increase over 5 years, this implies that

$$CPUE_{Year\ 5} = 1.50 \times CPUE_{Year\ 1}$$

or

$$\frac{CPUE_{Year\ 5}}{CPUE_{Year\ 1}} = 1.50 = (1+r)^{5-1} = (1+r)^4$$

where r is the proportional change per year. Notice that the exponent of $(1+r)$ is one less than the number of years because the first year's data establishes the baseline value.

We can solve the above equation to find that $r = 1.11$, i.e. corresponding to a 11% increase per year (compounded). For example, suppose that the mean CPUE had the value of 10 in year 1. Then in year 2, the mean CPUE will be 10× 1.11 or 11.07. Then in year 3, the mean CPUE will be 11.07 × 1.11 or 12.25. By year 5, the mean CPUE will have increased by 50% to 15.

We can create the following table showing the trend (on the logarithmic scale) needed for certain sized trends.

Table 1: Table 1. Slope on the (natural) logarithmic scale corresponding to certain change over a certain period.

Change	Over 5 Years	Over 10 Years
10%	0.024	0.011
30%	0.068	0.030
50%	0.107	0.046
100%	0.189	0.080
200%	0.316	0.130

I have written a *R* function (*slr.power.stroup()*) that estimates the power to detect a specified trend, given the process and sampling standard deviations, the α level (traditionally 0.05) and the sample sizes (number of sites measured each year and the number of years and pattern of sampling across years). This uses a methods developed by Stroup (1999). This function has arguments for the trend to be detected, the two standard deviations, the α level, and an argument that captures both the number of sites sampled each year, the number of years of sampling, and the pattern of the yearly samples (the *Xvalues* argument). For example, suppose that four sites are to be sampled in each of 5 years. Then the *Xvalues* argument would be specified as

$$Xvalues = c(1,1,1,1,2,2,2,2,3,3,3,3,4,4,4,4,5,5,5,5)$$

The individual values refer to the years sampled, and the replicated values specify how many sites are sampled each year. It is not necessary to worry if the same sites are sampled each year or new sites are sampled each year because any site-to-site variation will have been accounted for when the model was fit to the $\log(CPUE)$ data to estimate the variance components.

This can be specified in short hand way in *R* using

$$Xvalues = rep(1:5, each = 4)$$

Similarly, if sampling is only done in years 1, 3, and 5 with four sites sampled in each year, then the *Xvalues*

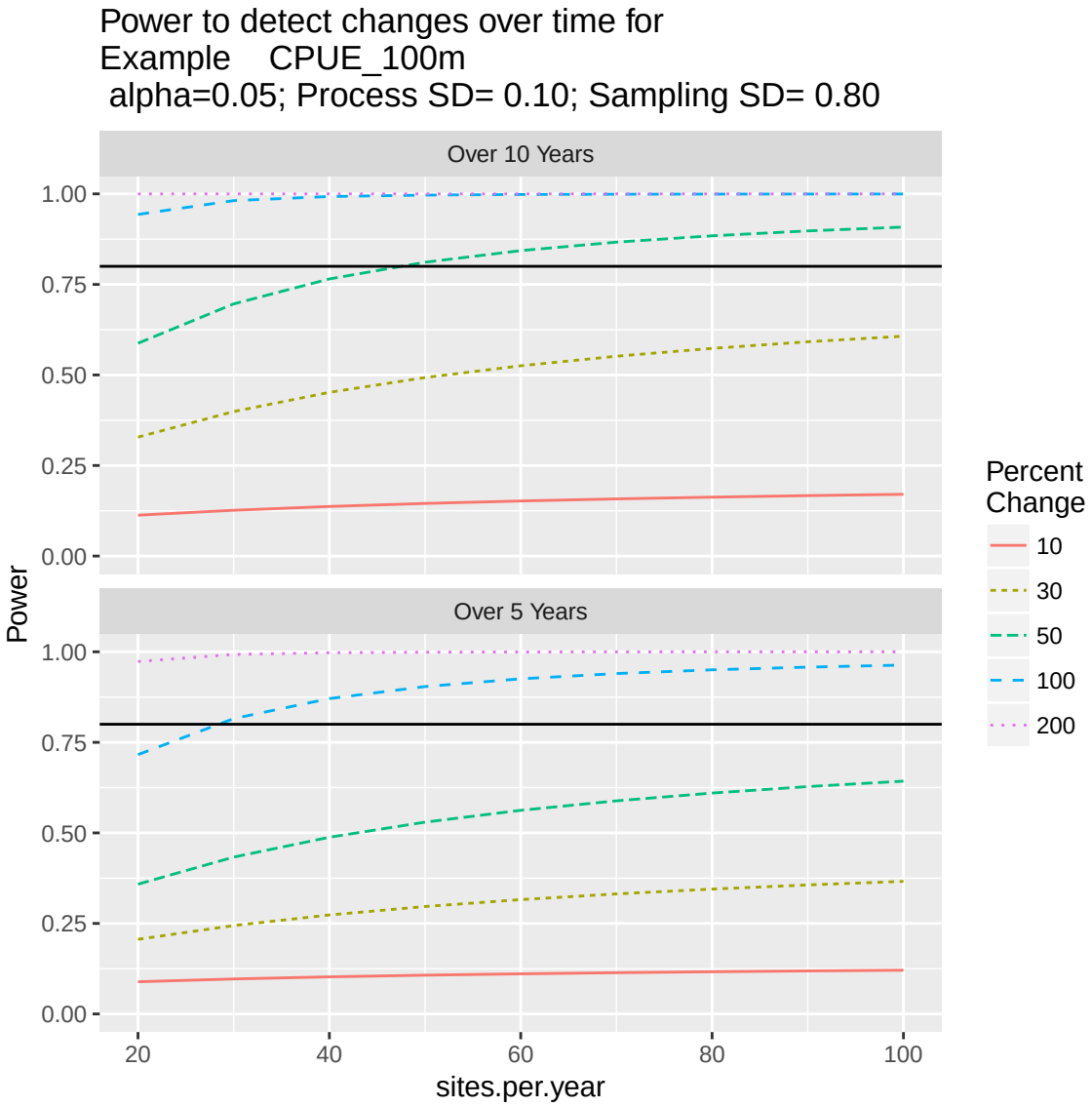
argument is specificed as:

$$Xvalues = c(1, 1, 1, 1, 3, 3, 3, 3, 5, 5, 5, 5)$$

This provides much flexibility to examine the impacts of different sampling effort and patterns on the power.

This function can be repeated called with different combinations of effort and trend and the results plotted as shown below.

```
source('functions.R')
  vc <- data.frame(SD.sampling=.8, SD.process=0.1, Watershed="Example ", Measure="CPUE_100m")
  alpha=0.05
  scenarios <- expand.grid(PerChange=c(10, 30, 50, 100, 200),
                           Years=c(5,10),
                           Sampling.SD=vc$SD.sampling,
                           Process.SD=vc$SD.process,
                           sites.per.year=seq(20,100,10),
                           Watershed=vc$Watershed,
                           Measure=vc$Measure,
                           alpha=alpha)
  scenarios$Scenario <- 1:nrow(scenarios)
  # estimate the power to detect a trend for each scenario
  power <- plyr::ddply(scenarios, "Scenario", function(x){
      # estimate compounded trend line on the log scale
      Trend <-  (x$PerChange/100+1)^(1/(x$Years-1))-1
      # sampling every year
      Xvalues <- rep(1:x$Year, each=x$sites.per.year)
      res <- slr.power.stroup(Trend=Trend,
                              Xvalues    =Xvalues   ,
                              Process.SD =x$Process.SD,
                              Sampling.SD=x$Sampling.SD,
                              alpha      =x$alpha)
      res <- cbind(res, x)
      res
   })
   # make a plot
   #browser()
   plotdata <- power
   plotdata$Years2 <- paste("Over ", plotdata$Year," Years",sep="")
   plotdata$PerChangeF <- factor(plotdata$PerChange)
   power.plot <- ggplot2::ggplot(data=plotdata, aes(x=sites.per.year, y=power.1sided.a, color=PerChangeF
      ggtitle(paste("Power to detect changes over time for \n", vc$Watershed, " ",vc$Type,"  ", vc$Measu
                    "; Process SD= ", format(round(power$Process.SD[1] ,2),nsmall=2),
                    "; Sampling SD= ",format(round(power$Sampling.SD[1],2),nsmall=2), sep=""))+
      geom_line(aes(group=PerChangeF, linetype=PerChangeF))+
      ylab("Power")+ylim(0,1)+geom_hline(yintercept=0.80)+
      facet_wrap(~Years2, ncol=1, scales='fixed')+
      scale_color_discrete(name="Percent\nChange")+
      scale_linetype_discrete(name="Percent\nChange")
   plot(power.plot)
```




```
# Find the sample size per year to detect 100% change over 5 years with 80% power
temp <- power[ power$Years==5 & power$PerChange==100,]
select <- which.max( temp$power.1sided.a >= 0.80)
ss <- temp[select, "sites.per.year"]
```

From this figure, we see that approximately 30 sites per year would be required to have a power of about 80% to detect a 100% increase over 5 years when the process standard deviation was 0.1, the sampling standard deviaton was 0.8 at $\alpha = 0.05$.

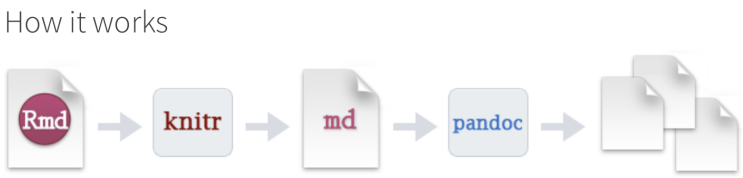
The power is relatively insensitive to the number of sites once it becomes large enough that the uncertainty in each year's mean CPUE is small so that process effects dominate changes from year to year.

9 Using the sample template.

9.1 Overview of process

Each power analysis report starts with the template file (*TrendPowerReport.Rmd*) that is *knitted* (in *R* parlance) together. The template file contain both text and *R* code as a living document so that as the data change, the report is updated when the the template are again knitted together.

The template is written using *RMarkdown* http://rmarkdown.rstudio.com which provides a simple set of commands to knit together *R* code and (formatted) text as shown below:



When you wish to render the document, *R Markdown* feeds the *.Rmd* file to *knitr*, which executes all of the code chunks and creates a new markdown (*.md*) document which includes the code and it's output in a general way.The *pandoc* program (included with *RStudio*) converts the markdown document to the final format (HTML, PDF, MSWord, etc.). While the above seems complicated, most of it is hidden from you and occurs "behind the scenes."

Creating the power analysis report will follow a multi-step process.

1. Read in the data from the FWIS system (the data is stored in the *Data* directory) and create some preliminary tables and plots to check the data, remove and outliers, or do any other pre-processing. This preprocessing is 'hidden' from the power report, but the code used to pre-process the data is stored with the document. Consequently, when you update the power analysis in a later year, the decisions made in pre-processing the data are evident.
2. Make some summary tables and plots for the wateshed as a whole. The current template makes some simple tables of counts of fish species by year. You are able to change the default figures and tables by modifying the corresponding *R* code.
3. Select the species of interest for which you want the power analysis. A power analysis is done for each species in turn.
4. Finally, you can enter summary text describing the conclusions from the power analysis.

A *RMarkdown* document has standard structure. For example, open the power analysis template file (*TrendPowerReport.Rmd.Rmd*) in *RStudio*. The script pane will look similar to:

```
---
title: "Your Favorite Watershed - Watershed Assessment Report"
author: "Carl Schwarz"
date: "`r format(Sys.time(), "%Y-%m-%d")`"
output:
  html_document:
    number_sections: yes
    toc: yes
  pdf_document:
    number_sections: yes
    toc: yes
  word_document:
    fig_caption: yes
    reference_docx: WatershedTemplate-STYLE.docx
    toc: yes
---

```{r setup, include=FALSE}
# Links for RMarkdown help

# Chunk options ..... https://yihui.name/knitr/options/

# Dealing with word templates and Rmarkdown
#   http://stackoverflow.com/questions/41982700/how-to-properly-number-headings-in-word-from-a-rmarkdown-document
#   http://rmarkdown.rstudio.com/articles_docx.html
#   http://rmarkdown.rstudio.com/word_document_format.html

# Information on using ggmap is available at
#   https://github.com/dkahle/ggmap/blob/master/README.md
```

The template file consists of 3 types of material:

- The headers (between the —'s) which identifies the power analysis name, author, date of creating, and list the options to be applied when the document is rendered. The date uses *R* to extract the date the code is run and formats it as YYYY-MM-DD. The options for a MSWord rendering indicate that the *TrendPowerReport.Rmd-STYLE.docx* file is to be used for styles when creating the document (this style document is where you specify that the sections are to be numbered). The headers only occur once in a template at the top of the primary file.
- Code chunks. These are sections of *R* code of the form delimited by triple BACK quotes with *R* code in between.

  ```
  ```{r chunkname, options}
     R code here
  ```
  ```

  Each chunk has a number of options specifying if the output is to be displayed in the report (e.g. a plot), or computations are proceed with nothing placed in the report. A list of chunk options is available at: https://yihui.name/knitr/options/. There is no limit to the number or types of chunks you can use.
- Text (such as):
  ```

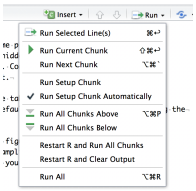
```
173
174 # Background
175 "How are the fish in my river and streams doing?"
176 We need this answer to set appropriate fishing regulations,
177 to understand and correct any problems with fish habitat and
178 to guard against invasive species.
179
180 A healthy fish population and fish community means we
181 can all enjoy the benefits of sustainable fisheries
182 and healthy ecosystems.
183 A standard method of assessing the status of
184 fish populations is necessary to allow comparisons of fish sustainability
```

which is written using *R Markdown* (refer to https://www.rstudio.com/wp-content/uploads/2015/03/rmarkdown-reference.pdf). There is no limit to the amount of text you can use.

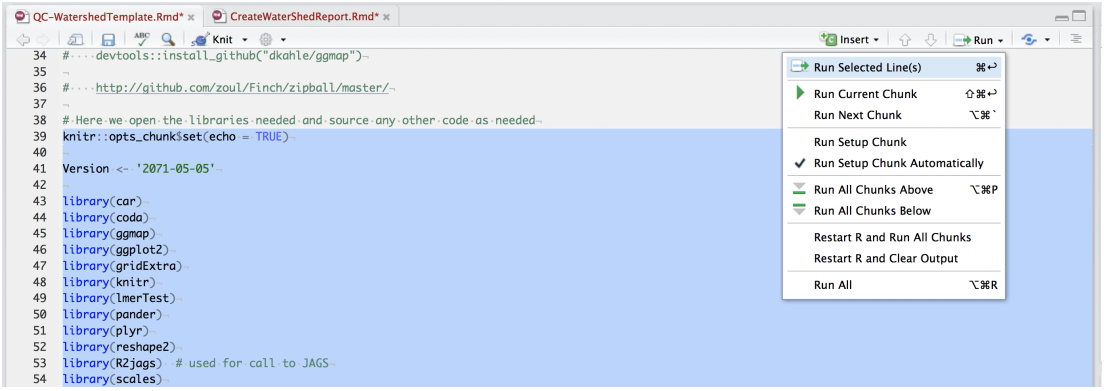
The text and code chunks can be intermixed at will.

The first chunk (labelled *setup*) is where we set up the *R* code by opening the libraries needed for the analyses, read in the data file, and do any data editing. The option *include=FALSE* implies that any of the output from this chunk will NOT be included in the draft report.

You can execute all of the *R* code in an entire chunk by clicking on the *Run* pop-down menu of *RStudio* and selecting the appropriate option.

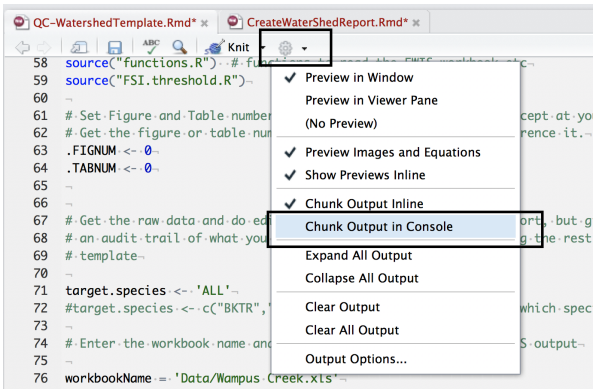


You can run one or several lines at a time by selecting, hightlighting, and then using the *Run* pop-down menu:



The keyboard short cuts (Command-Enter for Macintosh; Cntr-R for Windoze) can also be used.

You will find it useful to send all output to the console window during the template setup using:



I have created a sample template to help estimate the power sample size for a particular watershed Here are the steps to be taken to analyze a set of data for a system

9.2 Getting your computer ready

Your computer will need the following software. Instructions for installing the software can be found at http://people.stat.sfu.ca/~cschwarz/CourseNotes/HowGetSoftware.html.

- *R*. This is the statistical software used create summary table and to create plots. Don't forget to install the packages (add ons for *R*) as listed in item (6) of the installation instructions in the above link. You will also need to install the following packages:
  - *stringr* specialized functions for text processing
  - *pander* needed to create tables in *RMarkdown*
- *RStudio*. An integrated development environment that controls *R* and its output.

9.3 Getting the FWIS data

Information on each watershed is extracted from the FWIS database. Contact ????? for further details.

The current extraction program creates Excel 95 files. *R* requires workbooks to be in Excel 97 or later format, so you will have to open and save the file as Excel 97 or later format.

Many people simply find it easier to to save the FWIS file as a *csv* file and use that directly.

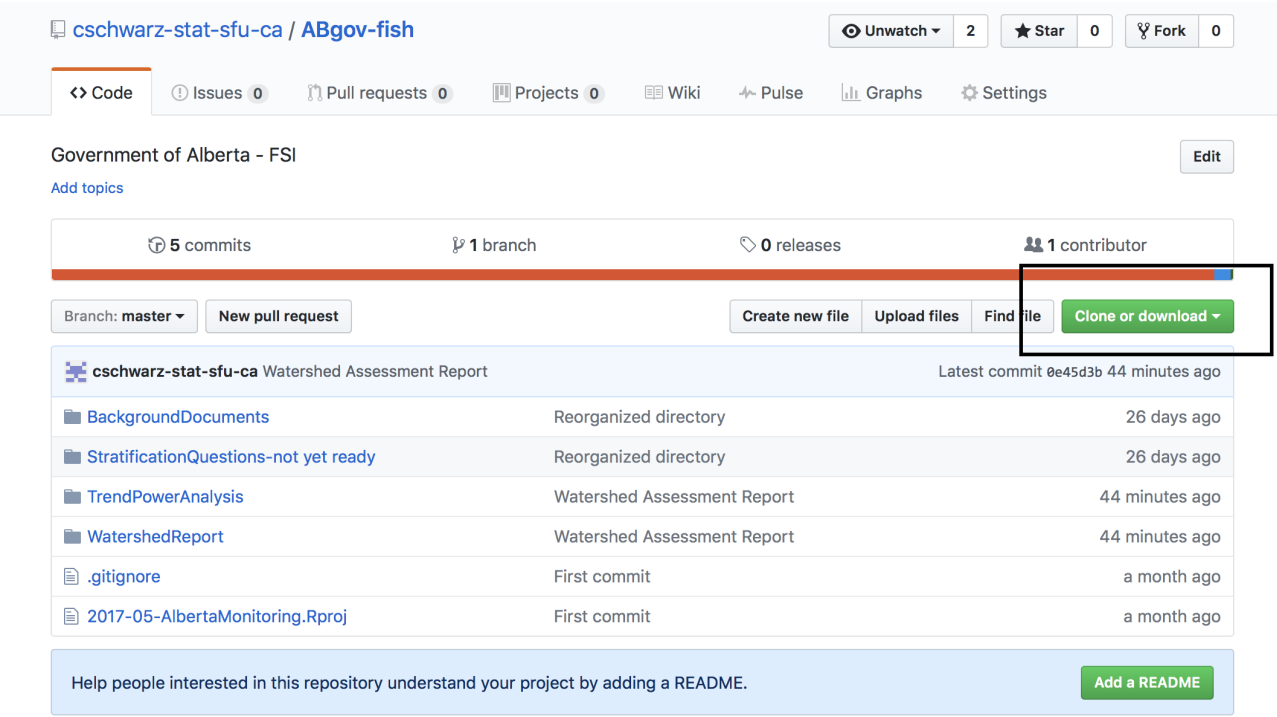
9.4 Create a working directory to hold the data files and the draft report template.

Create a working directory which will include the *RMarkdown* files used to create the draft watershed report and any data files, image files, etc. needed for the report.

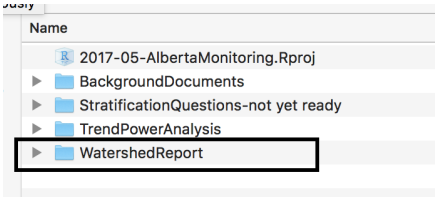
9.5 Copy the *RMarkdown* template files to the working directory.

The template files are located on the *GitHub* server and can be retreived as follows.

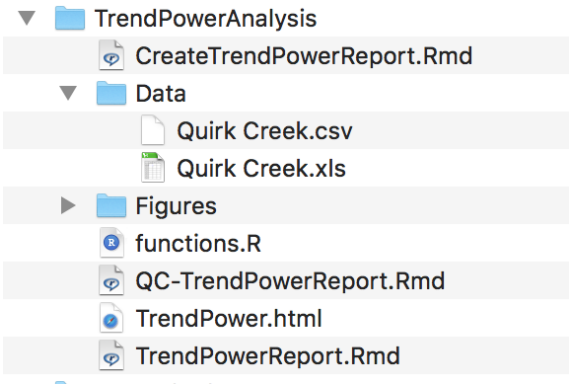
- Go to https://github.com/cschwarz-stat-sfu-ca/ABgov-fish The web page will look similar to:



- Click on the *Clone or Download* button to download a zip file of the entire contents.
- Move the zip file to your working directory.
- Unzip the zip file.
- Discard the zip file.
- Your working directory should now have several subdirectories:



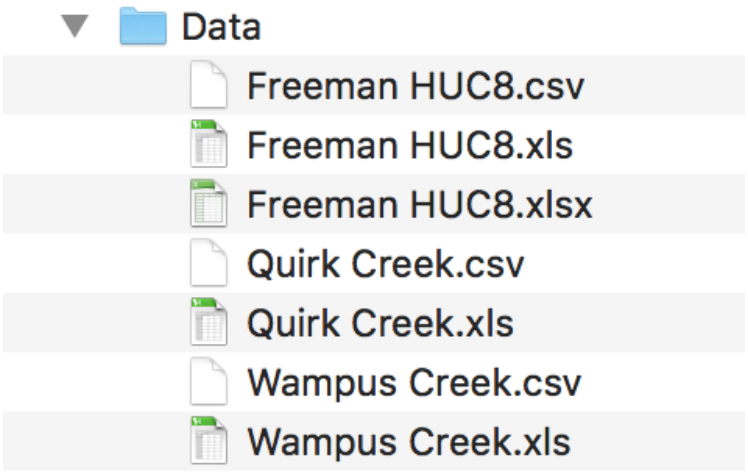
- Open the *TrendPowerAnalysis* directory



9.6 Move the FWIS data file to the *Data* directory.

Move the FWIS data file to the *Data* directory. **The data file MUST be in Excel 97 or later format.** The FWIS extraction system currently creates an Excel 95 file. You will have to open it and re-save it as an Excel 97 or later document (either *.xls* or *.xlsx* is fine).

The *Data* directory will look like:

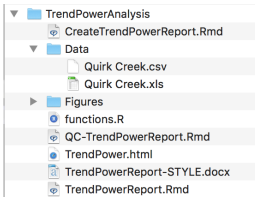


9.7 Duplicate and rename the *TrendPowerReport.Rmd* template file.

We plan to analyze the *Quirk Creek* data and so we will duplicate and rename the template file by adding a prefix (e.g. QC-) to the *TrendPowerReport.Rmd* file giving *QC-TrendPowerReport.Rmd*.

DO NOT MODIFY THE NAME OF THE TrendPowerReport-STYLE.docx file as this is a style template for the *MSWord* reports used by *RMarkdown* when creating the draft watershed report.

The directory show now look something like:



9.8 Edit the *xx-TrendPowerReport.Rmd* file.

9.8.1 Specify the workbook name or csv file name.

Specify the workbook name and/or the csv file name in the template file such as shown below:

```
# Enter the workbook name and worksheet name containing the FWIS output
# Or enter the name of the csv file.

workbookName = file.path('Data','Quirk Creek.xls')
sheetName    = 'F&W'
csvfilename  = file.path("Data","Quirk Creek.csv")
```

9.8.2 Select which method will be used to read the file

Select either the code fragment to read the workbook directly, or the code fragment to read the csv file. Here is the code fragment to read the csv file

```
# Read the csv file
# Check that  FWIS files exists
if(!file.exists(csvfilename))stop(paste("File ",csvfile,' not found '))

cpue <- read.FWIS.workbook.csv(
        csvfilename,
        target.species=target.species)
```

9.8.3 Do data editing.

I have included some basic data editing steps. Here you can remove outliers, select subset of the data, etc.

The final step of the data editing is to compute the catch summary. Here we need the estimated CPUE (fish / 300 m). Because of the way the data are structured, there are no records for a inventory for fish species not seen. Consequently, we need to impute a catch of 0 for species not seen during a particular inventory to make a complete set of CPUE for each Location in each Year of the study.

9.8.4 For which species do you want a power analysis?.

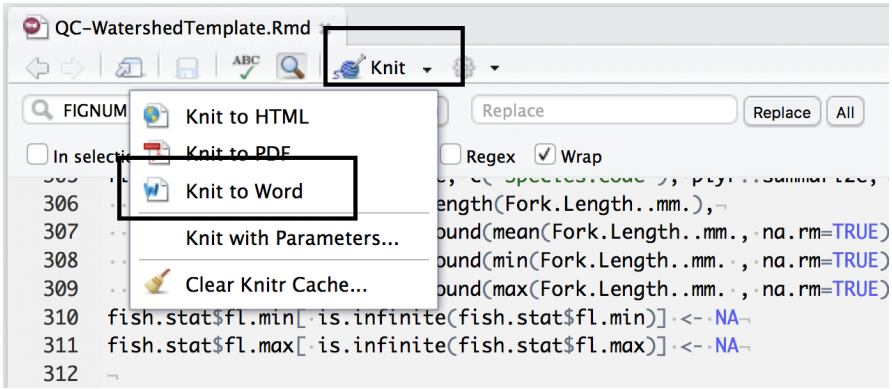
You may wish the power analysis only for a subset of the species. You can do this by modifying the *R* code fragement below. In this case a report will be made for ALL species found in the watershed.

```
target.species <- 'ALL'
#target.species <- c("BKTR","BLTR","MNWH") # or you can select which species to include
```

9.9 Obtain the power analysis.

Now that the data has been read and verified, the power analysis can be done.

Use the *Knit* button in *RStudio* to render the document in the format required.



If all goes well, a document in the format requested should be produced. Check the console for errors in either the rendering process or the the *R* code execution process for debugging. It is a bit of an art, but because all chunks are labelled, you can see in which chunk the *R* code failed.

I find it easier to create HTML documents during the debugging phase and leave creating the MSWord creation to the end.

CONGRATULATIONs! You have created your first draft of a watershed assessment!

The output will show the estimated variance components. Then these are used to estimate the power to detect a 10%, 30%, 50%, 100%, 200%, 300% change over 5 years by varying the number of sites given the sampling and process error estimated previously. A standard figure is created that can be modified as needed.

Sometimes, the mixed linear model fails – the variance components will be set to missing values and a message will appear in the list of variance components. The causes of these failures are difficult to predict, but often caused by singularities in the data (e.g. all the CPUE are identically equal to 0!). Please contact me for help in resolving the cases where the variance component analysis fails.

9.10 Write a summary statement.

Edit the template to include a summary statement about the dismal results.