

Point count data analysis: How to violate assumptions and get away with it

Peter Solymos

2019-06-04

Contents

1 Foreword	5
1.1 About the book and the course	5
1.2 About the author	5
1.3 Summary of course objectives	6
1.4 Installing	6
1.5 These are just reminders, to be deleted later	7
2 Introduction	9
2.1 Apples and oranges	9
2.2 Apples to apples	9
2.3 Effects can be significant	10
2.4 So what is a point count?	10
2.5 Questions we want to answer using point counts	10
2.6 Standardization by design	10
2.7 Protocols do vary	11
2.8 Pop quiz	11
2.9 Moving away from standards	12
2.10 Model based approaches	12
2.11 Models come with assumptions	12
2.12 Assumptions are everywhere	12
2.13 Pop quiz	12
2.14 The hard truth	12
2.15 Our approach	13
3 Organizing and Processing Point Count Data	15
3.1 JOSM (Joint Oil Sands Monitoring) data	15
3.2 Cross tabulating species counts	18
3.3 Joining species data with predictors	22
3.4 Explore predictor variables	22
4 A Primer in Regression Techniques	25
5 Behavioral Complexities	27
6 The Detection Process	29

7 Dealing with Recordings	31
8 A Closer Look at Assumptions	33
9 Understanding Roadside Surveys	35
10 Miscellaneous Topics	37

Chapter 1

Foreword

This book provides material for the workshop *Analysis of point-count data in the presence of variable survey methodologies and detection error* at the AOS 2019 conference by Peter Solymos.

The book and related materials in this repository is the basis of a full day workshop (8 hours long with 3 breaks).

Prior exposure to R language is necessary (i.e. basic R object types and their manipulation, such as arrays, data frames, indexing) because this is not covered as part of the course. Check this intro.

1.1 About the book and the course

You'll learn

- how to analyze your point count data when it combines different methodologies/protocols/technologies,
- how to violate assumptions and get away with it.

1.2 About the author

- Ecologist (molluscs, birds),
- pretty good at stats (modeling, detectability, data cloning, multivariate),
- R programmer (vegan, detect, ResourceSelection, pbapply),
- sometimes I teach (like today).

1.3 Summary of course objectives

This course is aimed towards ornithologists analyzing field observations, who are often faced by data heterogeneities due to field sampling protocols changing from one project to another, or through time over the lifespan of projects, or trying to combine ‘legacy’ data sets with new data collected by recording units. Such heterogeneities can bias analyses when data sets are integrated inadequately, or can lead to information loss when filtered and standardized to common standards. Accounting for these issues is important for better inference regarding status and trend of bird species and communities.

Analysts of such ‘messy’ data sets need to feel comfortable with manipulating the data, need a full understanding the mechanics of the models being used (i.e. critically interpreting the results and acknowledging assumptions and limitations), and should be able to make informed choices when faced with methodological challenges.

The course emphasizes critical thinking and active learning. Participants will be asked to take part in the analysis: first hand analytics experience from start to finish. We will use publicly available data sets to demonstrate the data manipulation and analysis. We will use freely available and open-source R packages.

The expected outcome of the course is a solid foundation for further professional development via increased confidence in applying these methods for field observations.

1.4 Installing

The **bookdown** package can be installed from CRAN or Github:

```
install.packages("bookdown")
# or the development version
# devtools::install_github("rstudio/bookdown")

## clean up
bookdown::clean_book(TRUE)
## rendering the book
bookdown::render_book('index.Rmd', 'bookdown::pdf_book')
bookdown::render_book('index.Rmd', 'bookdown::gitbook')
bookdown::render_book('index.Rmd', 'bookdown::epub_book')
```

To compile this example to PDF, you need XeLaTeX. You are recommended to install TinyTeX (which includes XeLaTeX): <https://yihui.name/tinytex/>.

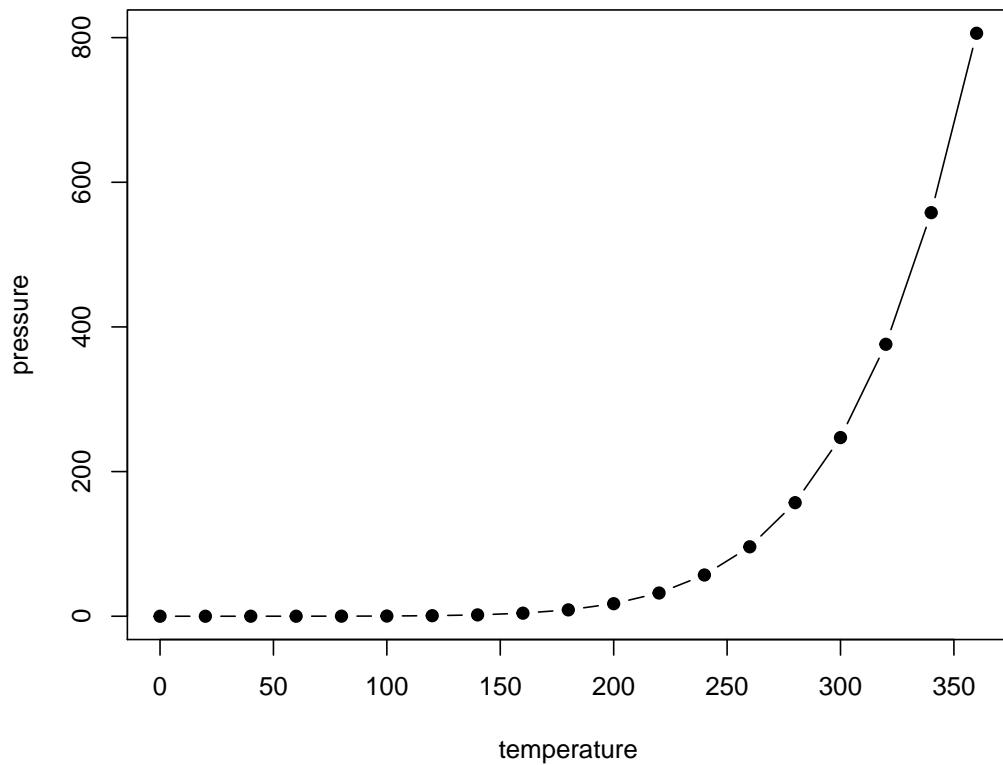


Figure 1.1: Here is a nice figure!

1.5 These are just reminders, to be deleted later

You can label chapter and section titles using `{#label}` after them, e.g., we can reference Chapter 2. If you do not manually label them, there will be automatic labels anyway, e.g., Chapter ??.

Figures and tables with captions will be placed in `figure` and `table` environments, respectively.

```
par(mar = c(4, 4, .1, .1))
plot(pressure, type = 'b', pch = 19)
```

Reference a figure by its code chunk label with the `fig:` prefix, e.g., see Figure 1.1. Similarly, you can reference tables generated from `knitr::kable()`, e.g., see Table 1.1.

```
knitr::kable(
  head(iris, 20), caption = 'Here is a nice table!',
  booktabs = TRUE
)
```

You can write citations, too. For example, we are using the `bookdown` package (Xie, 2018) in this sample book, which was built on top of R Markdown and `knitr` (Xie, 2015).

Table 1.1: Here is a nice table!

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.4	3.9	1.7	0.4	setosa
4.6	3.4	1.4	0.3	setosa
5.0	3.4	1.5	0.2	setosa
4.4	2.9	1.4	0.2	setosa
4.9	3.1	1.5	0.1	setosa
5.4	3.7	1.5	0.2	setosa
4.8	3.4	1.6	0.2	setosa
4.8	3.0	1.4	0.1	setosa
4.3	3.0	1.1	0.1	setosa
5.8	4.0	1.2	0.2	setosa
5.7	4.4	1.5	0.4	setosa
5.4	3.9	1.3	0.4	setosa
5.1	3.5	1.4	0.3	setosa
5.7	3.8	1.7	0.3	setosa
5.1	3.8	1.5	0.3	setosa

Chapter 2

Introduction

All assumptions are violated, but some are more than others

2.1 Apples and oranges

“A comparison of apples and oranges occurs when two items or groups of items are compared that cannot be practically compared.” [Wikipedia]

How we measure things can have big impact on our results.

- You might say: I saw 5 robins (walking down the road),
- I might say: I only saw one (sitting on my porch)

2.2 Apples to apples

Effort:

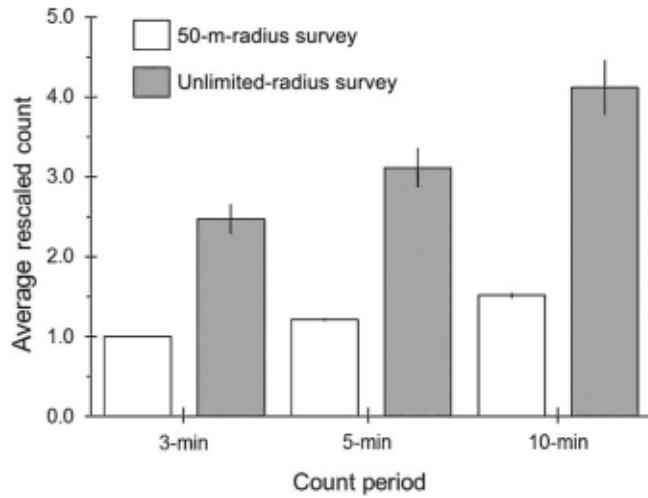
- area of the physical space searched,
- amount of time spent,
- number of individuals identified.

Experience, skill, “sensitivity”:

- number of years in field work,
- eye sight, hearing ability,
- mic sensitivity.

The goal is to make our measurements comparable.

2.3 Effects can be significant



10-min unlimited count ~300% increase over 3-min 50-m count. Average across 54 species of boreal songbirds.

2.4 So what is a point count?

- A trained observer
- records all the birds
- seen and heard
- from a point count station
- for a set period of time
- within a defined distance radius.

2.5 Questions we want to answer using point counts

- How many? (Abundance, density, population size)
- Is this location part of the range? (0/1)
- How is abundance changing in space? (Distribution)
- How is abundance changing in time? (Trend)
- What is the effect of a treatment on abundance?

2.6 Standardization by design

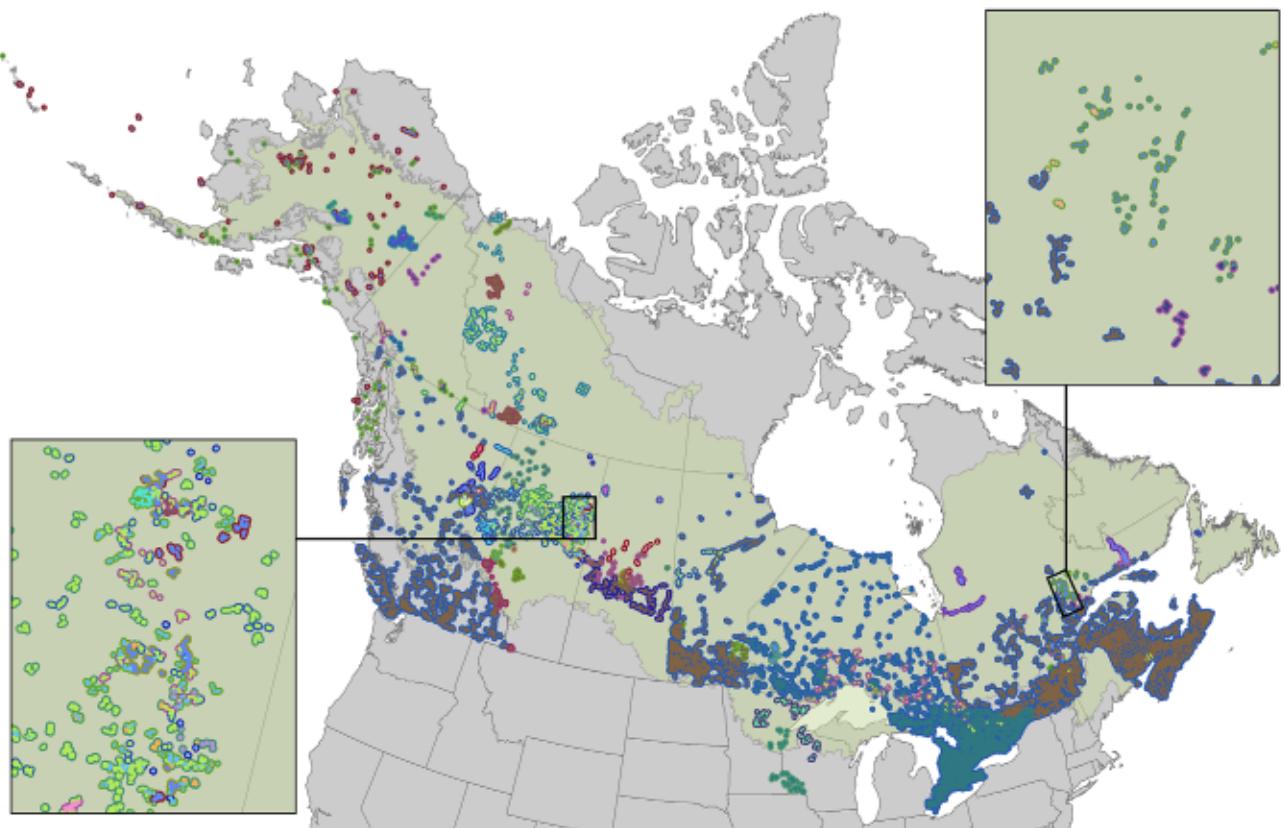
Have a set of standards/recommendations that people will follow to

- maximize efficiency in the numbers of birds and species counted,
- minimize extraneous variability in the counts.

But programs started to deviate from standards:

“For example, only 3% of 196,000 point counts conducted during the period 1992–2011 across Alaska and Canada followed the standards recommended for the count period and count radius.”

2.7 Protocols do vary



Survey methodology variation (colors) among contributed projects in the Boreal Avian Modelling (BAM) data base as of 2014.

2.8 Pop quiz

- In what regard can protocols differ?
- What drives protocol variation among projects?
- Why have we abandoned following protocols?

2.9 Moving away from standards

- Detection probabilities might vary even with fixed effort (we'll cover this more later),
- programs might have their own goals and constraints (access, training, etc).

2.10 Model based approaches

Less labour intensive methods for unmarked populations has come to the forefront:

- double observer (Nichols et al. 2000),
- distance sampling (Buckland et al. 2001),
- removal sampling (Farnsworth et al. 2002),
- multiple visit occupancy (MacKenzie et al. 2002),
- multiple visit abundance (Royle 2004).

2.11 Models come with assumptions

- Population is closed during multiple visits,
- observers are independent,
- all individuals emit cues with identical rates,
- spatial distribution of individuals is uniform,
- etc. (we will investigate this further in depth).

2.12 Assumptions are everywhere

Although assumptions are everywhere, we are really good at ignoring them:

- Relativistic time dilation is negligible (as long as we are not on a space station),
- samples are independent.

2.13 Pop quiz

- Can you mention some other common assumptions?
- Can you explain why we neglect/violate assumptions?

2.14 The hard truth

Assumptions are violated in many ways, because we seek simplicity.

The main question we have to ask: **does it matter in practice?**

2.15 Our approach

1. We will introduce a concept,
2. understand how we can infer it from data,
3. then we recreate the situation *in silico*,
4. and see how the outcome changes as we make different assumptions.

It is guaranteed that we violate **every** assumption we make.

To get away with it, we need to understand **how much is too much**.

“All assumptions are violated, but some are more than others.”

Chapter 3

Organizing and Processing Point Count Data

All data are messy, but some are missing

It is often called *data processing*, *data munging*, *data wrangling*, *data cleaning*. None of these expressions capture the dread associated with the actual activity.

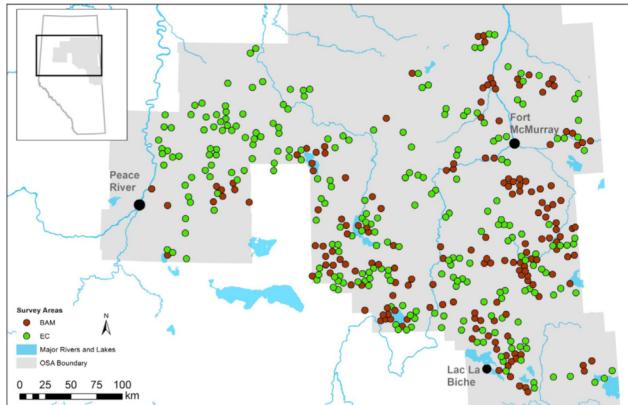
Luckily, there are only 4 things that can get messed up:

1. space (e.g. wrong UTM zones),
2. time (ISO format please),
3. taxonomy (UNK, mis-ID),
4. something else (if there were no errors, check again).

3.1 JOSM (Joint Oil Sands Monitoring) data

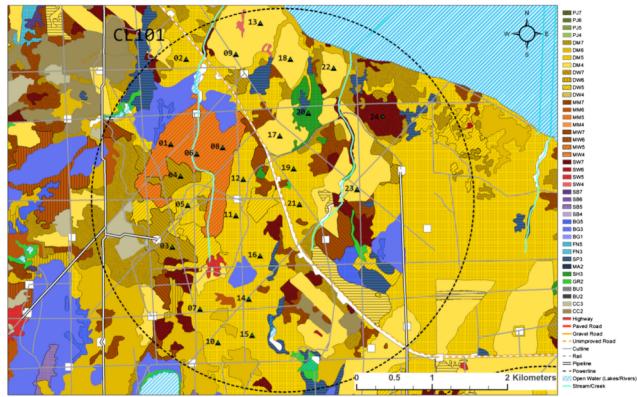
Look at the source code in the `_data/josm` directory of the book if you are interested in data processing details. We skip that for now.

```
include_graphics("./images/mahon-2016-fig-1.png")
```



Cause-Effect Monitoring Migratory Landbirds at Regional Scales: understand how boreal songbirds are affected by human activity in the oil sands area.

```
include_graphics("./images/mahon-2016-fig-2.png")
```



Survey area boundary ($r=2.5$ km circle), habitat type and human footprint mapping, and clustered point count site locations.

Surveys were spatially replicated because:

- we want to make inferences about a population,
- full census is out of reach,
- thus we take a sample of the population
- that is representative and random.
- Ideally, sample size should be as large as possible,
- it reduces variability and
- increases statistical power.

Survey locations were picked based on various criteria:

- stratification (land cover),
- gradients (disturbance levels),
- random location (control for unmeasured effects),
- take into account historical surveys (avoid, or revisit),
- access, cost (clusters).

The `josm` object is a list with 3 elements:

- `surveys`: data frame with survey specific information,
- `species`: lookup table for species,
- `counts`: individual counts by survey and species.

```
library(mefa4)
load("./_data/josm/josm.rda")
names(josm)
```

```
## [1] "surveys" "species" "counts"
```

Species info: species codes, common and scientific names. The table could also contain taxonomic, trait, etc. information as well.

```
head(josm$species)
```

At the survey level, we have coordinates, date/time info, variables capturing survey conditions, and land cover info extracted from 1 km² resolution rasters.

```
colnames(josm$surveys)
```

```
## [1] "SiteID"          "SurveyArea"      "Longitude"
## [4] "Latitude"        "Date"           "StationID"
## [7] "ObserverID"      "TimeStart"       "VisitID"
## [10] "WindStart"        "PrecipStart"     "TempStart"
## [13] "CloudStart"       "WindEnd"         "PrecipEnd"
## [16] "TempEnd"          "CloudEnd"        "TimeFin"
## [19] "Noise"            "OvernightRain"   "DateTime"
## [22] "SunRiseTime"     "SunRiseFrac"    "TSSR"
## [25] "OrdinalDay"      "DAY"             "Open"
## [28] "Water"            "Agr"             "UrbInd"
## [31] "SoftLin"          "Roads"           "Decid"
## [34] "OpenWet"          "Conif"           "ConifWet"
```

The count table contains one row for each unique individual of a species (`SpeciesID` links to the species lookup table) observed during a survey (`StationID` links to the survey attribute table). Check the data dictionary in `_data/josm` folder for a detailed explanation of each column.

```
str(josm$counts)
```

```
## 'data.frame': 52372 obs. of 18 variables:
## $ ObservationID: Factor w/ 57024 levels "CL10102-130622-001",...: 1 2 3 4 5 6 8 9 10 11 ...
## $ SiteID       : Factor w/ 4569 levels "CL10102","CL10106",...: 1 1 1 1 1 1 1 1 2 2 ...
## $ StationID    : Factor w/ 4569 levels "CL10102-1","CL10106-1",...: 1 1 1 1 1 1 1 1 2 2 ...
## $ TimeInterval : int 1 1 1 1 5 5 1 1 1 1 ...
## $ Direction    : int 1 2 2 2 1 4 4 4 1 1 ...
## $ Distance     : int 1 2 2 1 3 3 2 1 1 1 ...
```

```
## $ DetectType1 : Factor w/ 3 levels "C","S","V": 2 2 2 2 1 1 2 2 2 2 ...
## $ DetectType2 : Factor w/ 3 levels "C","S","V": NA NA NA NA NA NA NA NA NA ...
## $ DetectType3 : Factor w/ 3 levels "C","S","V": NA NA NA NA NA NA NA NA NA ...
## $ Sex         : Factor w/ 4 levels "F","M","P","U": 2 2 2 2 4 4 2 2 2 2 ...
## $ Age          : Factor w/ 6 levels "A","F","J","JUV",...: 1 1 1 1 1 1 1 1 1 ...
## $ Activity1   : Factor w/ 17 levels "BE","CF","CH",...: 5 5 5 5 NA NA NA 5 5 NA ...
## $ Activity2   : Factor w/ 17 levels "48","BE","CF",...: NA NA NA NA NA NA NA NA NA ...
## $ Activity3   : Factor w/ 7 levels "CF","DC","DR",...: NA NA NA NA NA NA NA ...
## $ ActivityNote: Factor w/ 959 levels "AGITATED","AGITATED CALLING",...: NA NA NA NA NA ...
## $ Dur          : Factor w/ 3 levels "0-3min","3-5min",...: 1 1 1 1 3 3 1 1 1 ...
## $ Dis          : Factor w/ 3 levels "0-50m","50-100min",...: 1 2 2 1 3 3 2 1 1 ...
## $ SpeciesID   : Factor w/ 150 levels "ALFL","AMBI",...: 107 95 95 107 46 43 140 95 125 38 ...
```

3.2 Cross tabulating species counts

Take the following dummy data frame (long format):

```
(d <- data.frame(
  sample=factor(paste0("S", c(1,1,1,2,2)), paste0("S", 1:3)),
  species=c("BTNW", "OVEN", "CANG", "AMRO", "CANG"),
  abundance=c(1, 1, 2, 1, 1),
  behavior=rep(c("heard", "seen"), c(4, 1)))
str(d)

## 'data.frame': 5 obs. of 4 variables:
## $ sample : Factor w/ 3 levels "S1","S2","S3": 1 1 1 2 2
## $ species : Factor w/ 4 levels "AMRO","BTNW",...: 2 4 3 1 3
## $ abundance: num 1 1 2 1 1
## $ behavior : Factor w/ 2 levels "heard","seen": 1 1 1 1 2
```

We want to add up the abundances for each sample (rows) and species (column):

```
(y <- Xtab(abundance ~ sample + species, d))
```

```
## 3 x 4 sparse Matrix of class "dgCMatrix"
##   AMRO BTNW CANG OVEN
## S1    .    1    2    1
## S2    1    .    1    .
## S3    .    .    .    .
```

y is a sparse matrix, that is a very compact representation:

```
object.size(d[,1:3])
```

```
## 2328 bytes
```

```
object.size(y)
```

```
## 2160 bytes
```

Notice that we have 3 rows, but `d$sample` did not have an `S3` value, but it was a level. We can drop such unused levels, but it is generally not recommended, and we need to be careful not to drop samples where no species was detected (this can happen quite often depending on timing of surveys)

```
Xtab(abundance ~ sample + species, d, drop.unused.levels = TRUE)
```

```
## 2 x 4 sparse Matrix of class "dgCMatrix"
##   AMRO BTNW CANG OVEN
## S1   .    1    2    1
## S2   1    .    1    .
## S3   0    0    0    0
```

A sparse matrix can be converted to ordinary matrix

```
as.matrix(y)
```

```
##   AMRO BTNW CANG OVEN
## S1   0    1    2    1
## S2   1    0    1    0
## S3   0    0    0    0
```

The nice thing about this cross tabulation is that we can filter the records without changing the structure (rows, columns) of the table:

```
Xtab(abundance ~ sample + species, d[d$behavior == "heard",])
```

```
## 3 x 4 sparse Matrix of class "dgCMatrix"
##   AMRO BTNW CANG OVEN
## S1   .    1    2    1
## S2   1    .    .    .
## S3   .    .    .    .
```

```
Xtab(abundance ~ sample + species, d[d$behavior == "seen",])
```

```
## 3 x 4 sparse Matrix of class "dgCMatrix"
##   AMRO BTNW CANG OVEN
## S1   .    .    .    .
## S2   .    .    1    .
## S3   .    .    .    .
```

Now let's do this for the real data. We have no abundance column, because each row stands for exactly one individual. We can add a column with 1's, or we can just count the number of rows by using only the right-hand-side of the formula in `Xtab`. `ytot` will be our total count matrix for now.

We also want to filter the records to contain only Songs and Calls, without Vvisual detections:

```
table(josm$counts$DetectType1, useNA="always")
```

```
##  
##      C      S      V  <NA>  
##  9180 41808  1384      0
```

We use SiteID for row names, because only 1 station and visit was done at each site:

```
ytot <- Xtab(~ SiteID + SpeciesID , josm$counts[josm$counts$DetectType1 != "V",])
```

See how not storing 0's affect size compared to the long formar and an ordinary wide matrix

```
## 2-column data frame as reference  
tmp <- as.numeric(object.size(  
  josm$counts[josm$counts$DetectType1 != "V", c("StationID", "SpeciesID")]))  
## spare matrix  
as.numeric(object.size(ytot)) / tmp
```

```
## [1] 0.1366087  
## dense matrix  
as.numeric(object.size(as.matrix(ytot))) / tmp
```

```
## [1] 1.106463  
## matrix fill  
sum(ytot > 0) / prod(dim(ytot))
```

```
## [1] 0.0491063
```

Check if counts are as expected:

```
max(ytot) # this is interesting  
  
## [1] 200  
sort(apply(as.matrix(ytot), 2, max)) # it is CANG
```

```
## BUFF BWTE COGO COHA DCCO GWTE HOLA NHOW NSHO RTHU WWSA CANV NOPI  
## 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## AMBI AMCO AMGO BAEA BAOR BEKI BOWA CONI CSWA EAPH GBHE GCTH GGOW  
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## GHOW HOWR LEOW MERL NESP NOGO NOHA NSWO PBGR RBGU RTHA SAVS SPSA  
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1  
## WBNU BRBL CAGU MYWA SNBU VEER AMKE AMWI BADO BARS BBWO BHCO BLBW  
## 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2  
## BLPW BLTE BWHA COGR DOWO EAKI HAWO KILL LEYE NAWA NOPO OSFL OSPR  
## 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2  
## PIWO PUFI RNDU SORA SSHA COSN AMCR AMRO ATTW BHVI BOCH BRCR BTNW  
## 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3
```

```

## CMWA FOSP FRGU GCKI MAWR MOWA NOFL PHVI SACR SOSA SOSP SPGR TRES
##   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
## WETA WIWA WIWR YBSA FOTE BAWW BBWA BCCH BLJA CAWA CONW COTE GRYE
##   3   3   3   3   3   4   4   4   4   4   4   4   4   4   4   4
## NOWA NRWS OCWA REVI RNGR RUBL RWBL WAVI WEWP WISN YBFL YWAR ALFL
##   4   4   4   4   4   4   4   4   4   4   4   4   4   4   4   5
## AMRE CHSP CORA EVGR HETH LCSP RBGR RBNU RCKI SWSP CCSP COYE DEJU
##   5   5   5   5   5   5   5   5   5   5   5   6   6   6   6
## LEFL LISPL MAWA OVEN RUGR SWTH BOGU MALL GRAJ PAWA WTSP YRWA COLO
##   6   6   6   6   6   6   7   7   7   8   8   8   8   8   9
## TEWA AMPI WWCR CEDW PISI RECR CANG
##  12  12  20  23  50  51 200

## lyover (FO) flock (FL) beyond 100m distance
head(josm$counts[
  josm$counts$SiteID == rownames(ytot)[which(ytot[, "CANG"] == 200)] &
  josm$counts$SpeciesID == "CANG",])

```

We can check overall mean counts

```
round(sort(colMeans(ytot)), 4)
```

```

## BUFF BWTE COGO COHA DCCO GWTE HOLA NHOW NSHO
## 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000
## RTHU WWSC CANV NOPI GBHE GCTH GHOW LEOW NOHA
## 0.0000 0.0000 0.0000 0.0000 0.0002 0.0002 0.0002 0.0002 0.0002
## RBGU BRBL CAGU AMCO BAEA BARS NESP NOGO NOPO
## 0.0002 0.0002 0.0002 0.0004 0.0004 0.0004 0.0004 0.0004 0.0004
## NSWO RNDU SNBU VEER BEKI CSWA MERL SAVS SSHA
## 0.0004 0.0004 0.0004 0.0004 0.0007 0.0007 0.0007 0.0007 0.0007
## MYWA AMKE BAOR OSPR SPGR WBNU AMGO AMWI BOWA
## 0.0007 0.0009 0.0009 0.0009 0.0009 0.0009 0.0011 0.0011 0.0011
## CONI EAPH HOWR NRWS BLTE COGR EAKI GGOW NAWA
## 0.0011 0.0011 0.0011 0.0011 0.0013 0.0013 0.0013 0.0013 0.0013
## COSN COTE FRGU MAWR FOTE KILL RTHA BADO BLBW
## 0.0013 0.0015 0.0015 0.0015 0.0015 0.0018 0.0020 0.0024 0.0024
## AMBI PBGR SPSA AMPI BHCO BWHA SOSP RUBL MALL
## 0.0028 0.0028 0.0028 0.0028 0.0031 0.0037 0.0042 0.0044 0.0046
## PUFI DOWO SORA LEYE ATTW HAWO RNGR BBWO BLJA
## 0.0048 0.0059 0.0068 0.0094 0.0096 0.0101 0.0101 0.0107 0.0134
## BOGU AMCR EVGR RWBL OSFL LCSP TRES FOSP WEWP
## 0.0140 0.0166 0.0169 0.0169 0.0186 0.0193 0.0201 0.0217 0.0232
## WIWA PIWO RECR SOSA YWAR GCKI BLPW CAWA SACR
## 0.0236 0.0256 0.0269 0.0269 0.0291 0.0304 0.0306 0.0315 0.0322
## BTNW NOWA OCWA BRCR CCSP COLO PHVI CONW CEDW
## 0.0335 0.0341 0.0359 0.0381 0.0385 0.0387 0.0394 0.0429 0.0449

```

```

##   RUGR    MOWA    WAVI    BCCH    BOCH    NOFL    SWSP    GRYE    WWCR
## 0.0475 0.0477 0.0582 0.0593 0.0593 0.0622 0.0659 0.0685 0.0751
##   AMRO    RBNU    BBWA    CMWA    BHVI    COYE    YBFL    YBSA    AMRE
## 0.0757 0.0766 0.0810 0.0812 0.0814 0.0814 0.0873 0.0878 0.0889
##   BAWW    LEFL    WETA    WISN    CORA    WIWR    ALFL    MAWA    PISI
## 0.0963 0.0974 0.1086 0.1280 0.1401 0.1466 0.1582 0.1727 0.1775
##   RBGR    LISP    DEJU    GRAJ    CANG    PAWA    REVI    RCKI    HETH
## 0.1832 0.2169 0.2725 0.2898 0.3018 0.3053 0.3344 0.3898 0.4344
##   CHSP    SWTH    WTSP    OVEN    YRWA    TEWA
## 0.4460 0.7402 0.8091 0.8831 0.8934 1.2221

```

3.3 Joining species data with predictors

Let's join the species counts with the survey attributes. This is how we can prepare the input data for regression analysis.

```

spp <- "OVEN" # which species
josm$species[spp,]

compare_sets(rownames(josm$surveys), rownames(ytot))

##           xlength ylength intersect union xbutnoty ybutnotx
## labels      4569      4569      4569  4569        0        0
## unique      4569      4569      4569  4569        0        0

x <- josm$surveys
x$y <- as.numeric(ytot[rownames(x), spp])

```

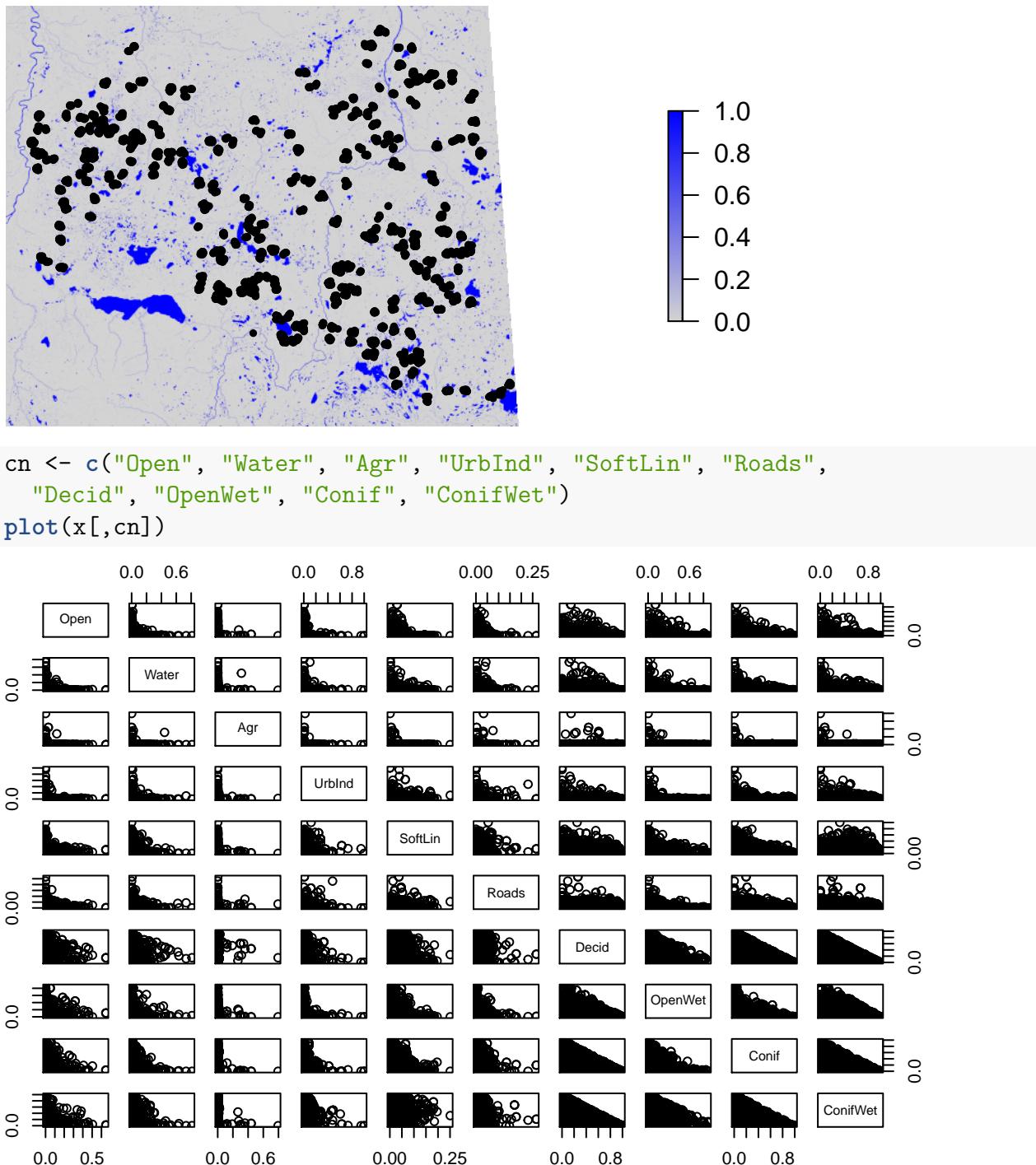
3.4 Explore predictor variables

Locations

```

library(raster)
library(sp)
rr <- stack("./_data/josm/landcover-hfi2016.grd")
#' Define CRS NAD83 for our sites
xy <- x[,c("Longitude", "Latitude")]
coordinates(xy) <- ~ Longitude + Latitude
proj4string(xy) <- "+proj=longlat +ellps=GRS80 +datum=NAD83 +no_defs"
xy <- spTransform(xy, proj4string(rr))
col <- colorRampPalette(c("lightgrey", "blue"))(100)
plot(rr[["Water"]], col=col, axes=FALSE, box=FALSE)
plot(xy, add=TRUE, pch=19, cex=0.5)

```



Add here:

- those kinds of transformations that are needed for regression
- need to add absolute links to figures???

Exercise:

- play with the data to understand the distributions
- use `summary`, `table`, `hist`, `plot`

Chapter 4

A Primer in Regression Techniques

All models are wrong, but some are useful – Box

lm, glm

main effects, interactions, offsets

lasso, brt, boot/bagging, glmm

conditional and marginal effects

maybe opticut

cloglog motivation

Chapter 5

Behavioral Complexities

Behaviour related stuff

constant p (time as covariate)

time varying p

finite mix

time varying p/c

rate, count, time-to-event

Chapter 6

The Detection Process

EDR, tau constant

truncated, unlimited

variable tau: habitat effect (continuous case?)

discrete: land cover, observer effects

contrast fixed effects with offsets – motivation for ARU

Chapter 7

Dealing with Recordings

integration challenges

calibration (exponential/cloglog approximation)

fixed effects

paired

sensor sensitivity - EDR

Chapter 8

A Closer Look at Assumptions

break thos assumptions

Chapter 9

Understanding Roadside Surveys

directional diff in signal transmission

Chapter 10

Miscellaneous Topics

model selection and conditional likelihood

variance/bias trade off

error propagation

MCMC?

N-mixture ideas

phylogenetic and life history/trait stuff

PIF methods

Bibliography

- Xie, Y. (2015). *Dynamic Documents with R and knitr*. Chapman and Hall/CRC, Boca Raton, Florida, 2nd edition. ISBN 978-1498716963.
- Xie, Y. (2018). *bookdown: Authoring Books and Technical Documents with R Markdown*. R package version 0.9.