

Night Tremors: There are more earthquakes at night

*David Hood*

*2017-05-06*



# Contents

<b>Front Matter</b>	<b>5</b>
0.1 Intended Audiences . . . . .	5
0.2 Acknowledgements . . . . .	7
0.3 Cover Picture . . . . .	7
0.4 Edition . . . . .	7
0.5 Statistics . . . . .	7
0.6 Using R . . . . .	7
0.7 Contacting me . . . . .	9
0.8 Links . . . . .	9
0.9 Chapter Code . . . . .	9
<b>1 Motivation</b>	<b>11</b>
1.1 Formal Statement . . . . .	12
1.2 Links . . . . .	12
1.3 Chapter Code . . . . .	13
<b>2 Data Processing</b>	<b>15</b>
2.1 Event types . . . . .	17
2.2 Gridded Latitude and Longitude . . . . .	19
2.3 Angle of the sun . . . . .	20
2.4 Formal statement . . . . .	20
2.5 Code . . . . .	21
2.6 Chapter Code . . . . .	21
<b>3 Key result</b>	<b>25</b>
3.1 Formal Statement . . . . .	25
3.2 Chapter Code . . . . .	26
<b>4 It is not dependence</b>	<b>29</b>
4.1 Formal Statement . . . . .	30
4.2 Chapter Code . . . . .	30
<b>5 It is not masking</b>	<b>33</b>
5.1 Formal Statement . . . . .	34
5.2 Links . . . . .	34
5.3 Chapter Code . . . . .	35
<b>6 It is not seasonal</b>	<b>39</b>
6.1 Formal statement . . . . .	40
6.2 Chapter Code . . . . .	40
<b>7 It is not the moon</b>	<b>43</b>
7.1 Absolute difference . . . . .	45

7.2 Relative . . . . .	48
7.3 Formal Statement . . . . .	50
7.4 Links . . . . .	51
7.5 Chapter Code . . . . .	51
<b>8 It is not uniform</b>	<b>63</b>
8.1 Formal Statement . . . . .	64
8.2 Chapter Code . . . . .	64
<b>9 Japanese earthquakes are the same</b>	<b>69</b>
9.1 Formal Statement . . . . .	71
9.2 Links . . . . .	71
9.3 Chapter Code . . . . .	71
<b>10 The continental United States is similar</b>	<b>81</b>
10.1 Formal Statement . . . . .	83
10.2 Links . . . . .	83
10.3 Chapter Code . . . . .	83
<b>11 Italian earthquakes are the same</b>	<b>93</b>
11.1 Formal Statement . . . . .	95
11.2 Links . . . . .	95
11.3 Chapter Code . . . . .	95
<b>12 Hawaii is similar</b>	<b>105</b>
12.1 Formal Statement . . . . .	107
12.2 Links . . . . .	107
12.3 Chapter Code . . . . .	107
<b>13 The Philippines is similar</b>	<b>117</b>
13.1 Formal Statement . . . . .	119
13.2 Links . . . . .	119
13.3 Chapter Code . . . . .	119
<b>14 Thailand is similar</b>	<b>129</b>
14.1 Formal Statement . . . . .	131
14.2 Links . . . . .	131
14.3 Chapter Code . . . . .	131
<b>15 Taiwan is not similar</b>	<b>141</b>
15.1 Formal Statement . . . . .	143
15.2 Links . . . . .	143
15.3 Chapter Code . . . . .	143
<b>16 Australia is similar</b>	<b>153</b>
16.1 Formal Statement . . . . .	155
16.2 Links . . . . .	155
16.3 Chapter Code . . . . .	155
<b>17 It is gravity</b>	<b>165</b>
17.1 Gravitational force differentials . . . . .	165
17.2 Formal Statement . . . . .	167
17.3 Chapter Code . . . . .	167

# Front Matter

There are times — possibly never, for most people — that you encounter a result so unlikely, so against common wisdom, that you need to write several thousand words and run many statistical tests to explain the result. This monograph is caused by such events.

## 0.1 Intended Audiences

Carl Sagan popularised the phrase “extraordinary claims require extraordinary evidence”, and in working through my analysis and discussing it with people, I have certainly been aware of a healthy scepticism about the results. One way of providing evidence is to make the analysis completely reproducible. I conducted the analysis in the data analysis language R, and anyone generally competent in the language should be able to repeat my analysis and verify my conclusions. By using R, I can provide a start to finish trail of everything I did in conducting the analysis, with accompanying text as to why. So, one audience is people who go “What!?” at the title, and want to look for mistakes.

In my experience of helping many people learn data analysis skills over the years, there is a common pattern. After acquiring some skills, people can stall because they are not sure of how to apply them. This gives my narrative of conducting the analysis, including what I noticed, why I chose particular techniques, and how I interpreted the results. For those people with some data analysis skills, this will hopefully give some ideas about how to use them and why.

I am making some strong, evidence-based, claims through the book about the sun and earthquakes, so the book might be of interest to those with a general interest in geophysics. My methods in analysing this data are general statistical techniques, rather than geophysics modelling. As a consequence, what I am identifying about earthquakes is different to a structural geological approach. The difference in ways of looking at the data means, at the time of writing, there is still work to be done unifying these conclusions with traditional geological understanding.

Finally, this is a case study in citizen science. I am not a geologist. However, the techniques applied are within the grasp of anyone prepared to put in the time to study books and online resources, and the data used is publicly available. The hard labour was done by the seismologists establishing the earthquake monitoring networks around the world and gathering the data. I am just benefitting from that data (and the knowledge of how to analyse the data) being publicly available. This kind of analysis is an example of something within reach of anyone taking an interest in improving their statistical skills. For me, I worked with data for many years, and over a year or so brushed up my statistical skills in evenings and weekends. I then did this analysis on a home computer in evenings and weekends. The resources for people to do this are out there; it is a matter of time and practice.



Figure 1: Inangahua earthquake

## 0.2 Acknowledgements

I would like to thank Simon Cox, Principal Scientist, GNS Science - Te Pu Ao, for asking challenging and well thought-out questions that led me to extend my analysis from “this is freaky” to “this is what is going on”.

I would not have had the skills to do this analysis if I hadn’t done a lot of online courses in analysing data in the past few years. I would particular like to note Mine Çetinkaya-Rundel, Roger Peng, Jeff Leek, and Brian Caffo for offering material that helped me up-skill my statistical and analysis understanding.

## 0.3 Cover Picture

The cover picture is of the 1968 Inangahua earthquake, that occurred at 5:24 am (local time) when the sun was 26 degrees below the horizon. The image comes from Archives New Zealand and is reproduced under a Creative Common Attribution 2.0 Generic license Link 1

## 0.4 Edition

This is the 1st edition, of 6th May 2017.

book site: <https://thoughtfulbloke.github.io/eqnz/>

github repo: <https://github.com/thoughtfulbloke/eqnz>

## 0.5 Statistics

While in the earlier drafts I used some complex statistical techniques that need a number of settings, I have trimmed it back to simple techniques that are sufficient to make the point (and deliver similar results to the more complex techniques).

One concept I make heavy use of is coin-toss (binomial) style statistical results. The idea that an earthquake could occur with one of two outcomes i.e. day or night; later in the analysis, there are binary choices like when the sun is 30 to 40 degrees below the horizon or the sun is at another angle. The statistics boil down to how likely the possible outcomes are given the chance of each option and the number of times I toss a coin (or have an earthquake occur in the data).

I also use confidence intervals in this study. These are ranges of outcomes that there is a probability of the “true” value being inside the range. In most of the study, I am using up to seven sigma intervals. Two sigma is considered worth publishing, and five sigma is used for confidence about fundamental physics experiments. With seven sigma, the true value is expected to be within the range 99.9999999744% of the time.

## 0.6 Using R

I include all the code for repeating the analysis in the course of the book. However, this book is not a full tutorial in learning R for data analysis. As there are many resources for learning R, I am writing this under the assumption that people who want to understand the detail of the code are familiar enough with R to understand data types, the dplyr analysis workflow, and the use of functions.

When I started the analysis I was using R version 3.2, by the time I finished this book I was using R version 3.3.2. Everything still works in the same way, so my only advice is to use a recent enough version that you

can use all the same packages (helper libraries) I used in the analysis. To specifically cite the work of others in the creation of R and supporting packages (in order of package name):

R Core Team (2016). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.

binom for making confidence intervals

Sundar Dorai-Raj (2014). binom: Binomial Confidence Intervals For Several Parameterizations. R package version 1.1-1. <https://CRAN.R-project.org/package=binom>

bookdown for making the book

Yihui Xie (2016). bookdown: Authoring Books and Technical Documents with R Markdown. R package version 0.3.

dplyr for general purpose data manipulations

Hadley Wickham and Romain Francois (2016). dplyr: A Grammar of Data Manipulation. R package version 0.5.0. <https://CRAN.R-project.org/package=dplyr>

geosphere for calculating distances

Robert J. Hijmans (2016). geosphere: Spherical Trigonometry. R package version 1.5-5. <https://CRAN.R-project.org/package=geosphere>

ggplot for some of the graphs

H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2009.

lubridate for date operations

Garrett Grolemund, Hadley Wickham (2011). Dates and Times Made Easy with lubridate. Journal of Statistical Software, 40(3), 1-25. URL <http://www.jstatsoft.org/v40/i03/>.

mapdata for a detailed map of New Zealand

Original S code by Richard A. Becker and Allan R. Wilks. R version by Ray Brownrigg. (2016). mapdata: Extra Map Databases. R package version 2.2-6. <https://CRAN.R-project.org/package=mapdata>

maps for drawing maps

Original S code by Richard A. Becker, Allan R. Wilks. R version by Ray Brownrigg. Enhancements by Thomas P Minka and Alex Deckmyn. (2016). maps: Draw Geographical Maps. R package version 3.1.1. <https://CRAN.R-project.org/package=maps>

maptools for the position of the sun

Roger Bivand and Nicholas Lewin-Koh (2016). maptools: Tools for Reading and Handling Spatial Objects. R package version 0.8-39. <https://CRAN.R-project.org/package=maptools>

plotrix for round graphs

Lemon, J. (2006) Plotrix: a package in the red light district of R. R-News, 6(4): 8-12.

readr for reading in some data

Hadley Wickham, Jim Hester and Romain Francois (2016). readr: Read Tabular Data. R package version 1.0.0. <https://CRAN.R-project.org/package=readr>

rvest for getting data from the web

Hadley Wickham (2016). rvest: Easily Harvest (Scrape) Web Pages. R package version 0.3.2. <https://CRAN.R-project.org/package=rvest>

tidyR for data rearrangement

Hadley Wickham (2016). `tidyR`: Easily Tidy Data with `spread()` and `gather()` Functions. R package version 0.6.0. <https://CRAN.R-project.org/package=tidyr>

## 0.7 Contacting me

Should you want to reach me, possible ways are Twitter Link 2, my blog Link 3, Github Link 4, or even email Link 5.

## 0.8 Links

For each section in the monograph, relevant links for the section are listed near the end so that it works in print as well as on a webpage. For example:

- 1 - Cover Image: <https://www.flickr.com/photos/archivesnz/8808854738>
- 2 - On Twitter: <https://twitter.com/Thoughtfulnz>
- 3 - My Blog: <https://thoughtfulbloke.wordpress.com>
- 4 - On Github: <https://github.com/thoughtfulbloke/>
- 5 - e-mail: [david@thoughtful.net.nz](mailto:david@thoughtful.net.nz)

## 0.9 Chapter Code

At the end of each section is any R code used in the creation of that section. In the case of `library()` commands, these libraries need to be pre-installed on the computer. This code includes a lot of commands found in other chapters as I have tried to make each section self contained so people can run just the parts that interest them. My own version of the code shares code between chapters, so is organised a little differently.

For most of chapters that draw on the New Zealand data, you will need to run Chapters 1 and 2 to prepare the data before running the other chapters.

```
library(lubridate)
library(maptools)
# Angles of the sun for the Inangahua earthquake
# As NZ Standard Time in May 1968, I am just subtracting 12 hours to make UTC
dt <- ymd_hm("1968-05-28 05:24", tz = "UTC") - hours(12)
solarpos(matrix(c(172.04, -41.76), nrow=1), dt)
```

As I technical note to Rmarkdown users, I purled the code in each chapter to create a separate file which I embeded at the end of the chapter.

I used a MIT licence for the code on Github, which I understand means people are free to use any of the code but should acknowledge me if you copy it directly.



# Chapter 1

## Motivation

**“Earthquakes are equally as likely to occur in the morning or the evening”** Link 1

In September of 2016, a scientific paper was published that reported a small relationship between the phase of the moon and a particular set of earthquakes. This report was vastly overblown by many commenting on the internet into a generalised “the moon causes earthquakes”. To cut through some of the hype, I thought I would write a satirical piece focusing on the sun (not expecting to find anything); I could then pare down the data until I got a result and thus demonstrate how conclusions can be overblown. This is because even given the small influence of the moon on selected fault lines, as of October 2016, there was no evidence for a general link between astronomical activity and earthquakes. Earthquakes were seen as a subsurface process of internal faults and pressures.

Even in satire, I like to be thorough. I downloaded 5 full solar years of earthquake data for New Zealand from the Geonet earthquake database Link 1. Geonet, the official home of New Zealand earthquake data, provides a publicly accessible database of earthquake events. Through a web form visitors can generate a database query with various possible added filters, and either download earthquake information through a web browser or use the query string with programs that can download files from the internet. For large amounts of data, the database separates the queries into blocks of 20,000 earthquakes to avoid system strain.

As the amount of day and night varies within a year, a full year avoids biases created by only downloading earthquakes in a narrow time range. Downloading five years was to avoid biases created by earthquake swarms occurring in a single year. As five years meant several queries of the database, I added a step to remove duplicates to avoid any confusion over whether the dates marking each range were inclusive or exclusive. Since I made my download in October 2016, the five-year range was from the start of October 2011 to the end of September 2016.

After downloading more than 100,000 earthquakes for this 5 year period for the New Zealand region, I calculated which hour of the day, in Coordinated Universal Time (UTC), each earthquake took place in. Then I made a basic graph to get a general feeling for the data. I expected a fairly flat line with a little random fluctuation, as there was no reason to expect any real differences.

The results shocked me. Not only was there a visible difference of thousands of earthquakes, the difference was consistent for both the night and day. Both of these things were radically different to the understanding of how earthquakes behaved. The size of the differences was more than would be expected, and the consistency of the differences between night and day showed that it was not random. Since night in New Zealand falls approximately after 7 and before 19 hours, this meant that around 56685 of the 101902 earthquakes occurred in approximate night hours; to put it another way, with 45217 earthquakes occurring in day hours, 25 % more earthquakes occurred at night than in the day.

If the occurrence of earthquakes is not influenced by the sun - as was generally believed to be the case - then the chance of getting such a large difference between night and day over so many earthquakes was too

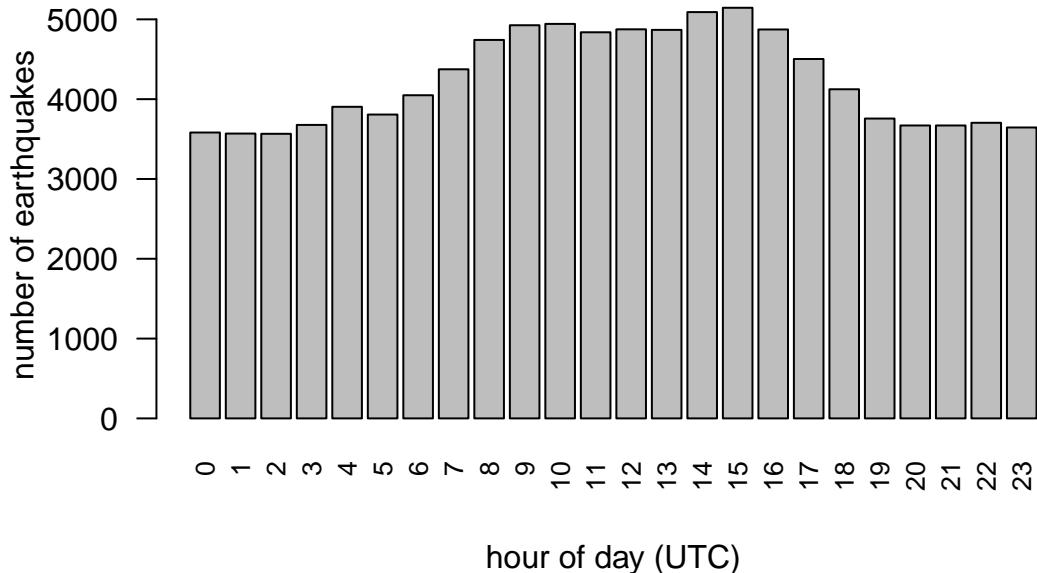


Figure 1.1: New Zealand Earthquakes by hour (UTC)

small for the computer to accurately calculate. However, for those happy with approximate figures, it is somewhere in the order of a p-value of:

I could also express the odds of seeing these results by chance given no true night/day effect as a bit less than 1 in 3 googols.

This result was really dramatic, like seeing something fall upwards instead of downwards- if my understanding of the area was correct this result should have been impossible. Once I recovered from the shock of encountering something so unlikely compared to established knowledge, instead of writing a short satirical piece, I had a significant mystery that needed explaining. This book is the account of working through an analysis of that mystery.

## 1.1 Formal Statement

New Zealand earthquake frequency shows features inconsistent with the established idea of earthquakes being equally likely in the day and night.

## 1.2 Links

1 - Can the position of the moon or the planets affect seismicity? Are there more earthquakes in the morning/in the evening/at a certain time of the month? (2016) U.S. Department of the Interior |U.S. Geological Survey |DOI Inspector General URL: [www2.usgs.gov/faq/categories/9827/3354](http://www2.usgs.gov/faq/categories/9827/3354)

2 - Geonet Earthquake Database quakesearch: <http://quakesearch.geonet.org.nz> Description of Geonet Database search output: <http://info.geonet.org.nz/display/appdata/Catalogue+Output>

## 1.3 Chapter Code

```
## ----setup, include=FALSE-----
knitr::opts_chunk$set(echo = FALSE)
knitr::opts_chunk$set(warnings=FALSE)
knitr::opts_chunk$set(errors=FALSE)
knitr::opts_chunk$set(message=FALSE)
knitr::opts_chunk$set(dpi = 150)
knitr::opts_chunk$set(fig.width =  6)
knitr::opts_chunk$set(fig.height =  4)

##
library(dplyr)
library(lubridate)

Sys.setenv(TZ = "UTC")

##

#if it doesn't exist, create an earthquake datasets folder
if(!dir.exists("eqdata")){
  dir.create("eqdata")
}

#if the data file doesn't exist, create it
if(!file.exists("eqdata/eqnz_raw.RData")){
  #bit1 & 2 are a hack to fit things on a single line
  bit1 <- "http://quakesearch.geonet.org.nz/csv?"
  bit2 <- "bbox=163.03711,-49.32512,182.41699,-32.47270&"
  e1 <-
    read.csv(paste0(bit1,bit2,"startdate=2015-09-01&enddate=2016-09-01"),
             stringsAsFactors = FALSE
    )
  e2 <-
    read.csv(paste0(bit1,bit2,"startdate=2014-08-01&enddate=2015-09-01"),
             stringsAsFactors = FALSE
    )
  e3 <-
    read.csv(paste0(bit1,bit2,"startdate=2013-09-01&enddate=2014-08-01"),
             stringsAsFactors = FALSE
    )
  e4 <-
    read.csv(paste0(bit1,bit2,"startdate=2012-10-01&enddate=2013-09-01"),
             stringsAsFactors = FALSE
    )
  e5 <-
    read.csv(paste0(bit1,bit2,"startdate=2011-09-01&enddate=2012-10-01"),
             stringsAsFactors = FALSE
    )
```

```

    )
eqnz <- rbind(e1, e2, e3, e4, e5)
eqnz$time_UTC <-
  as.POSIXct(eqnz$origintime, format = "%Y-%m-%dT%H:%M:%OSZ", tz = "UTC")

eqnz <-
  eqnz %>% filter(time_UTC >= as.POSIXct("2011-09-01T00:00:00", format =
                                             "%Y-%m-%dT%H:%M:%S", tz = "UTC") &
    time_UTC < as.POSIXct("2016-09-01T00:00:00", format =
                           "%Y-%m-%dT%H:%M:%S", tz = "UTC"))

) %>%
  distinct() %>% arrange(time_UTC)
eqnz$eq_hour <- hour(eqnz$time_UTC)
save(eqnz, file="eqdata/eqnz_raw.RData")
}

# clean down objects from environment for next stage
rm(list=ls())
load("eqdata/eqnz_raw.RData")

## ----fig.cap="New Zealand Earthquakes by hour (UTC)"-----
barplot(table(eqnz$eq_hour), cex.names=0.8, las=2, ylab="number of earthquakes",
        xlab="hour of day (UTC)")

## -----
# calculate summaries of how many in parts of day
n_night <- sum(eqnz$eq_hour > 7 & eqnz$eq_hour <=19)
n_total <- length(eqnz$eq_hour)
n_day <- n_total - n_night
n_perc_diff <- paste(round(100 *(n_night - n_day) / n_day,0), "%")
p_val = binom.test(n_night,n_total)$p.value

```

# Chapter 2

## Data Processing

Data processing involves finding, and ideally fixing, problems in the data that prevent an analysis being made. It also involves inventing transformations in the data to allow new insights. To me, the first of these is largely a matter of practice in becoming familiar with what goes wrong with data. The second is a more creative process of “what can be done to the data to get it to reveal things it otherwise would not, and is there a price to be paid in doing so?”

A good place to start is to make some general summaries looking for any odd biases in the data.

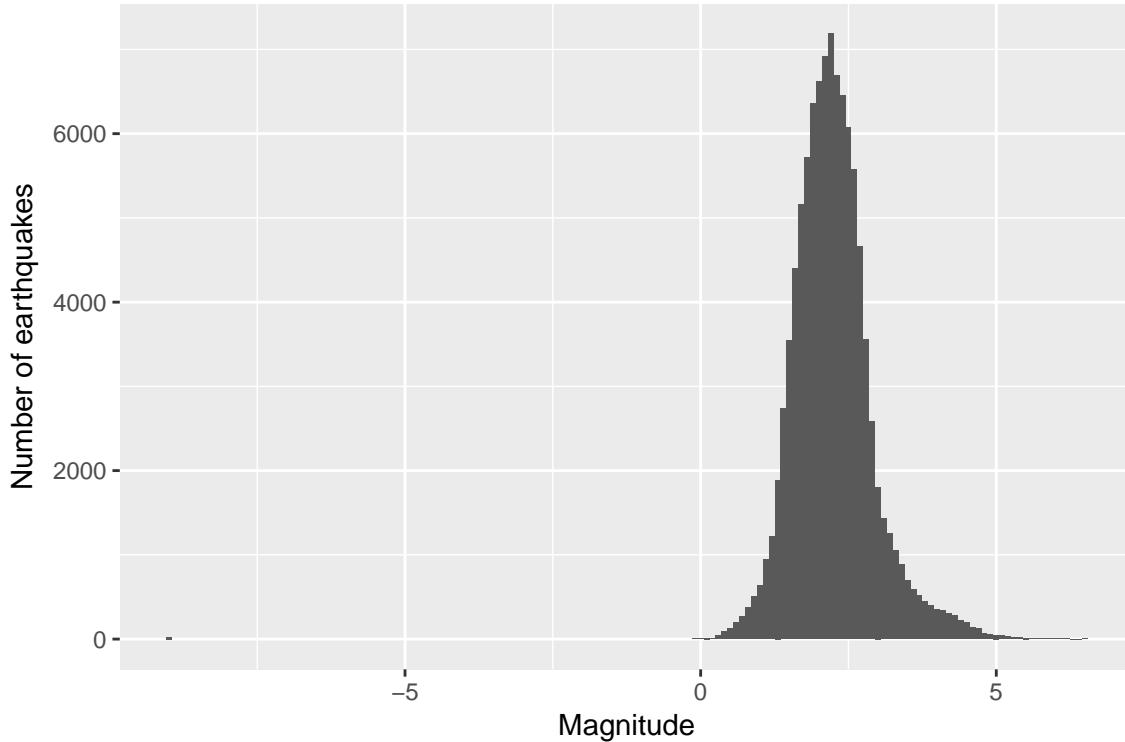


Figure 2.1: Distribution of Magnitude (steps of .1)

This is an artificially normal looking distribution. To the right of the peak of the histogram, at about magnitude 2.2, the number of earthquakes are decreasing as earthquakes become less frequent at higher magnitudes. To the left of the peak, the number is decreasing due to smaller earthquakes being more difficult to detect. This missing data might cause potential biases in the results if there is a connection

between an earthquake being detected at all and the time that it was detected.

In checking, there are a small number of events of -9 magnitude (negative magnitude is possible but rare). These seem to be extreme outliers and are likely to badly affect any model I create (as well as cause annoying amounts of empty space on graphs of magnitude).

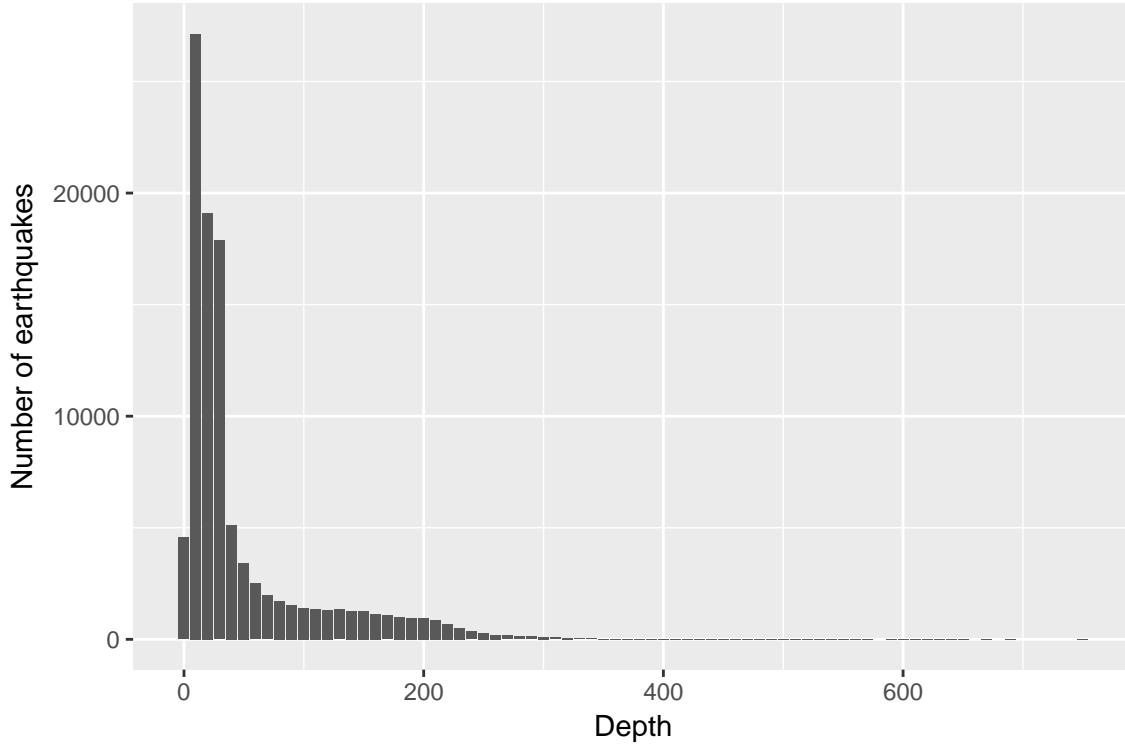


Figure 2.2: Distribution of Depth (steps of 10 km)

The graph of estimated earthquake depth does not have a smooth distribution; it is worth noting that the earth's crust is not smoothly distributed either, so this is unsurprising. There is an unusual reduction in events around depth 0. This is as much an excuse not to include events at this depth as a reason; the mechanics of bits of ground pushing and pulling against other bits of ground feels like it should be different at surface level, and on that basis alone I am happy to put surface events to one side

From the exploratory analysis this far I am inclined to:

- a) remove events of less than zero magnitude as being very few and suspiciously different to the rest of the data
- b) remove surface events as being different to subsurface ones.

From the graph of magnitude vs depth I see no reason to keep either the surface earthquakes or the negative magnitude earthquakes. In this I am technically not following best practice; since neither group of earthquakes affect the overall analysis, it would be better practice to keep them in (those comfortable in modifying the code I provide can easily discover the inclusion of these earthquakes makes no difference to the conclusions). Best practice would be to include the most data to make the broadest conclusions. However, I personally do not want to get bogged down in comparatively trivial debates about the quality of the data with the negative magnitude earthquakes included, or the interplay of forces with surface level earthquakes, so I am personally happy to discuss things in terms of positive magnitude subsurface earthquakes.

After removing the events at the surface and the events below zero magnitude, I did a quick check of what the magnitude distributions for under 3 magnitude earthquakes look like for each hour of the day. Magnitudes

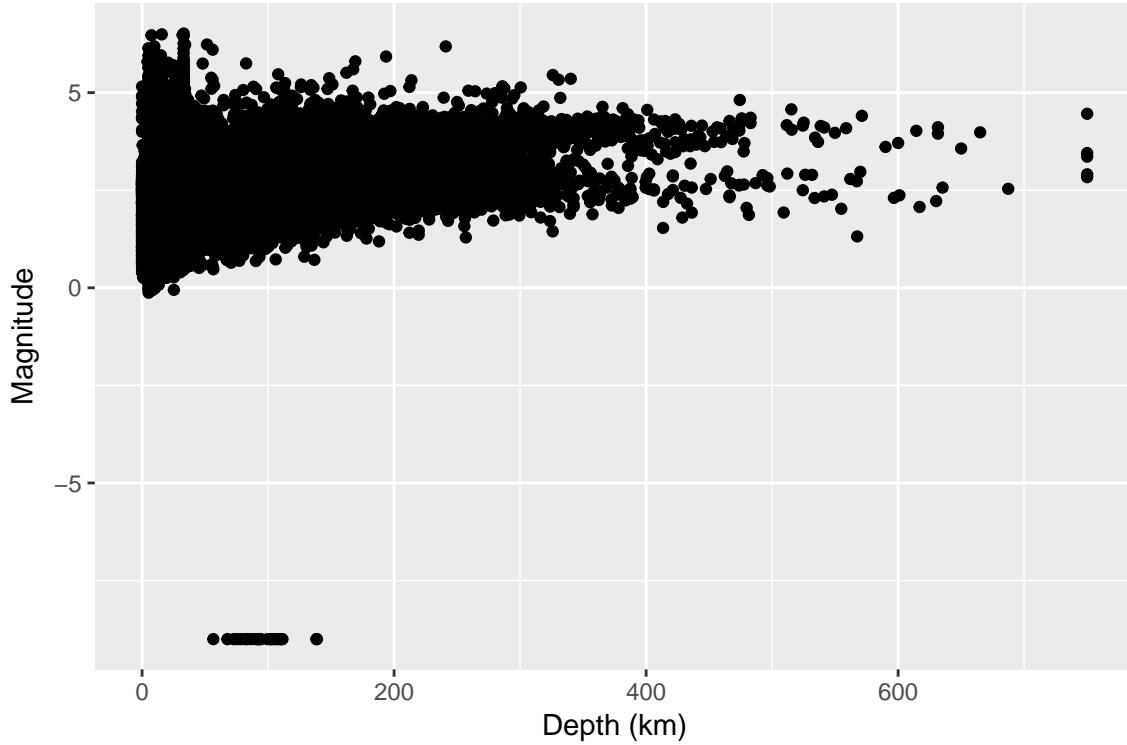


Figure 2.3: Depth and Magnitude of earthquakes

of under three were chosen because that is the level at which the number of detectable earthquakes begins to decrease, and therefore the missing data could cause biases.

All the individual hours seem to be showing a similar pattern, which leads me (in judging it by eye) to suspect it will be unlikely that there is any relationship between detected earthquakes and time of day that could be influencing results.

## 2.1 Event types

From the Earthquake Catalogue documentation, I am interested in two classes of events: earthquakes (reviewed by a human being) and likely earthquakes (indicated by a blank entry, but as that is annoying in tables I am replacing it with the term “**likely earthquake**” in the data)

This human review process could, in theory, mean that there are unknown confounders causing a large number of low magnitude events. However, there are no significant numbers of non-earthquake classified events with a magnitude of less than three, so there is no evidence for any alternative. As a result, I am in agreement with the Geonet documentation that likely earthquakes can be considered earthquakes.

### 2.1.1 Latitude and Longitude

Latitude and longitude are critical to the analysis, so it seems like a good idea to actually check them, particularly since New Zealand is near the international date line with a sudden longitude shift from 180 to -180.

Latitudes range from -49.3178902 to -32.4753952, while longitudes range from -179.9998322 to 180.63854 (thought it might be better to say longitudes from 164.2942963 to -177.59093).

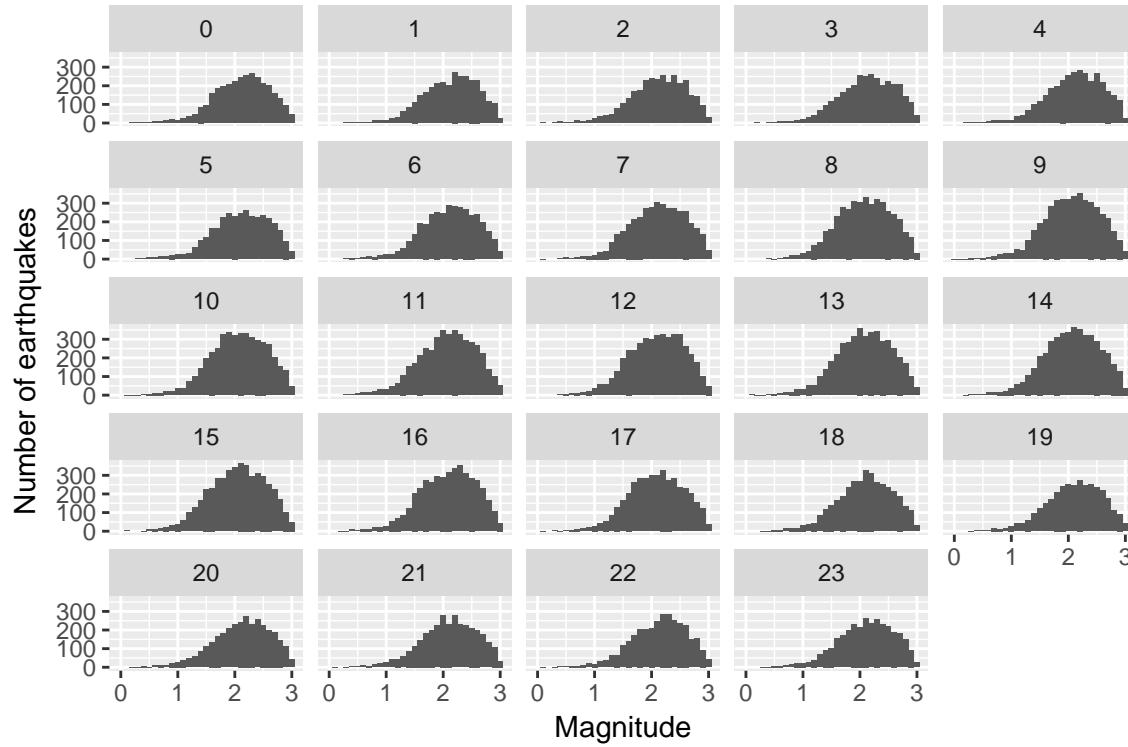


Figure 2.4: Number of earthquakes by magnitude, split by hour of day (UTC)

Table 2.1: Events types and number in the Geonet Database

Event type	Total
duplicate	1
earthquake	36424
explosion	1
induced earthquake	1
landslide	1
likely earthquake	63855
not locatable	19
nuclear explosion	1
other	1
outside of network interest	1367
quarry blast	21
volcanic eruption	1

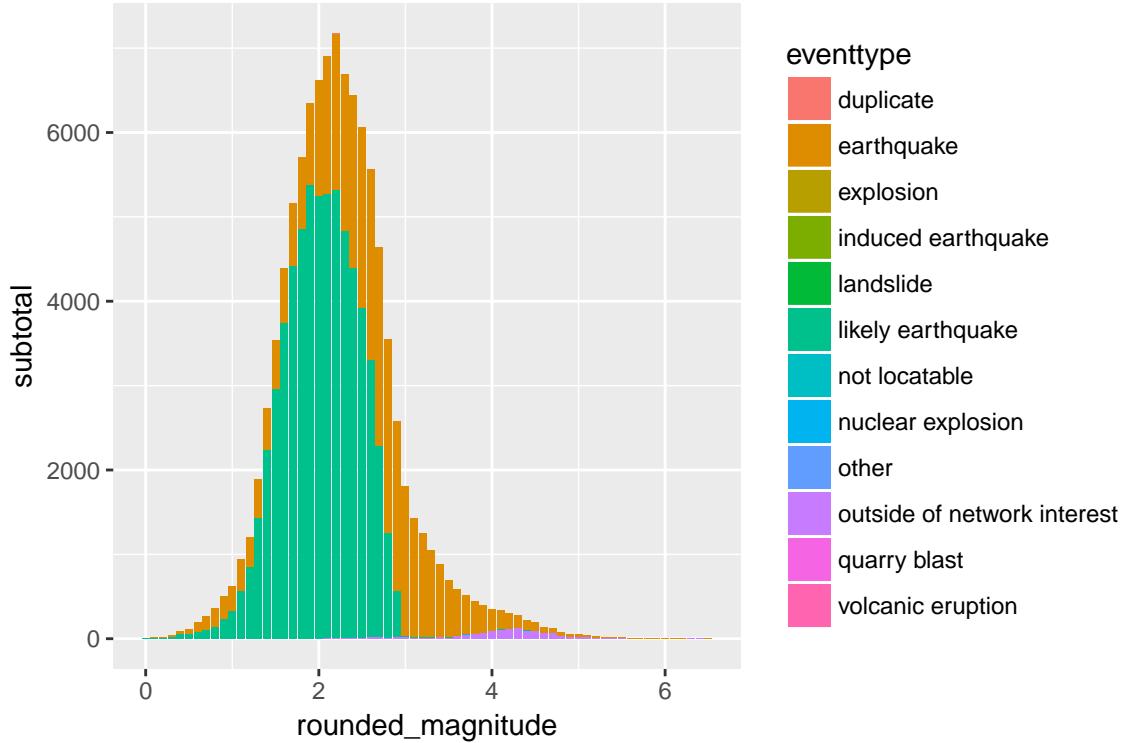


Figure 2.5: Number of earthquakes by Magnitude and Classification

The presence of negative longitudes represented as positive numbers greater than 180 causes problems for some R packages when conducting longitude related calculations, so I will correct the greater than 180 values to be negative values slightly greater than -180.

From the raw earthquake data I calculate a series of out variables, mostly following the principle of “it seemed like a good idea at the time”.

## 2.2 Gridded Latitude and Longitude

In the raw data, the earthquakes are individual events, but another way of looking at the data is as regions which have earthquakes. One straightforward approach is to take the entire area represented by the data, and divide that area into grids.

The size of the grids is a trade-off. Larger grids mean that local differences are homogenised into one block and variation between smaller in-block areas is lost. In this view, the analysis of the entire data is an analysis of one large grid square. On the other hand, smaller areas reduce the number of earthquakes in any single area, reducing the sample size and therefore weakening our confidence in our statistical results. An individual earthquake by itself is representative of a tiny area. An initial grid composed of squared 50km on a side seems reasonable here, given the size of the total area and the number of earthquakes within the 5 year period.

To make the grids, I found the southernmost of the earthquake events and calculate how far north of that base point each earthquake is. Then divided that distance by 50 and rounded to a whole number. I converted that distance back to a latitude for convenience of future plotting.

I take the same approach for longitude, starting from the easternmost earthquake and working westwards to calculate the grid centre closest to the earthquake event location.

While I might feel that 50 kilometres per side is a good grid size, I should also verify this intuition using tests:

104 (28%) of the 377 grid locations have more than 100 earthquakes, which I would expect to be enough to draw useful conclusions from, as each grid point contributes less than 1% to the overall figures. Each earthquake in each grid square also contributes less than 1% to the overall aggregate for the square. Among the data that we have, this minimises the effects of unusual variation.

## 2.3 Angle of the sun

Using the hour of the day is only an approximate measure of night and day, as the position of the sun at a particular hour changes with latitude, longitude, and time of year. Exactly declaring a particular place and time night or day requires knowing the position of the sun at that moment. This can be done with a function in the maptools package. The function returns both a compass and vertical angle for the sun's position, and I convert that to a 360 degree angle along an east-west arc. My reason for doing so is that the east-west arc of the sun sees the significant daily change, while the north-south component is a gradual seasonal change. This, in turn, allows me to simplify the problem from a three-dimensional one to a two-dimensional problem focused on the change within a day.

At the same time, I can use time of sunrise and sunset to calculate the length of night in the day the earthquake took place on, as this can be a better measure for seasonality than month, since the months cycle is slightly offset from the solar cycle of around 21st of December to 21st of June.

From the help files for the solar position function, the N.O.A.A. provided algorithms are accurate to approximately one minute for sunrise and sunset among latitudes with +/- 72 degrees.

Given there is a tiny potential uncertainty, I carry out quick test that it is giving sensible answers, by checking the range of sun angles per hour of UTC day.

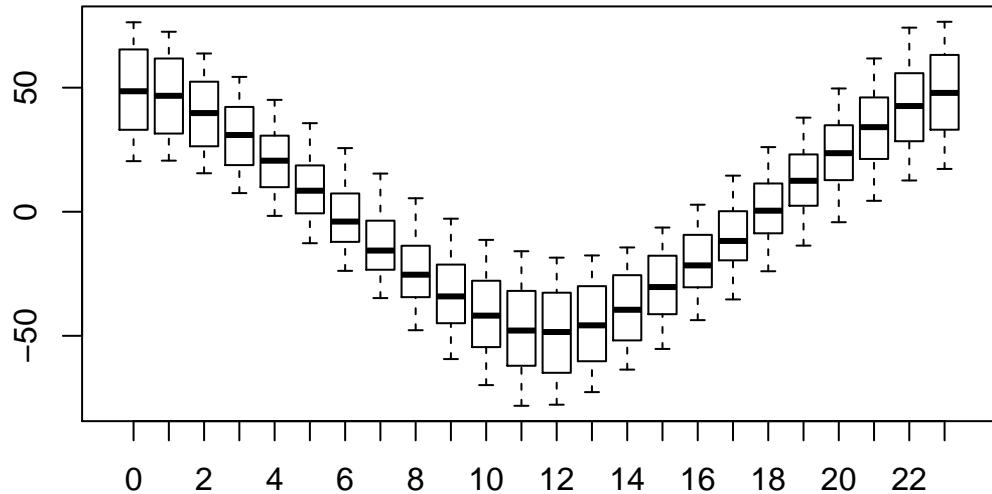


Figure 2.6: Variation in angle of sun by hour of day

The range of values seems reasonable, so I have no concerns about the calculated angles.

## 2.4 Formal statement

This analysis is restricted to subsurface earthquake and likely earthquake events of greater than zero magnitude.

## 2.5 Code

### 2.6 Chapter Code

```

## ----setup, include=FALSE-----
knitr::opts_chunk$set(echo = FALSE)
knitr::opts_chunk$set(warnings=FALSE)
knitr::opts_chunk$set(errors=FALSE)
knitr::opts_chunk$set(message=FALSE)
knitr::opts_chunk$set(dpi = 150)
knitr::opts_chunk$set(fig.width = 6)
knitr::opts_chunk$set(fig.height = 4)

## -----
library(dplyr)
library(lubridate)
library(ggplot2)
library(maptools)
library(geosphere)
library(solidearthtide)
Sys.setenv(TZ = "UTC")

# Assumes there is eqnz_raw data created in chapter 1
load("eqdata/eqnz_raw.RData")

## ----fig.cap="Distribution of Magnitude (steps of .1)"-----
eqnz %>% mutate(rounded_magnitude=round(magnitude,1)) %>%
  group_by(rounded_magnitude) %>%
  summarise(subtotal = n()) %>%
  ggplot(aes(x=rounded_magnitude, y=subtotal)) + geom_bar(stat="identity") +
  ylab("Number of earthquakes") + xlab("Magnitude")

## ----fig.cap="Distribution of Depth (steps of 10 km)"-----
eqnz %>% mutate(rounded_depth=round(depth,-1)) %>%
  group_by(rounded_depth) %>%
  summarise(subtotal = n()) %>%
  ggplot(aes(x=rounded_depth, y=subtotal)) + geom_bar(stat="identity") +
  ylab("Number of earthquakes") + xlab("Depth")

## ----fig.cap="Depth and Magnitude of earthquakes"-----
ggplot(eqnz, aes(x=depth, y=magnitude)) + geom_point() +
  ylab("Magnitude") + xlab("Depth (km)")

## ----fig.cap="Number of earthquakes by magnitude, split by hour of day (UTC)"-----
eqnz %>% filter(magnitude > 0, magnitude < 3) %>%
  mutate(rounded_magnitude=round(magnitude,1), eq_hour = hour(time_UTC)) %>%
  group_by(rounded_magnitude, eq_hour) %>%
  summarise(subtotal = n()) %>%
  ggplot(aes(x=rounded_magnitude, y=subtotal)) + geom_bar(stat="identity") +
  facet_wrap(~ as.factor(eq_hour)) + ylab("Number of earthquakes") +
  xlab("Magnitude")

```

```

## -----
eqnz$eventtype[eqnz$eventtype == ""] <- "likely earthquake"

## -----
eqnz %>% filter(magnitude > 0, depth > 0) %>%
  group_by(eventtype) %>% summarise(total=n()) %>%
  select(`Event type` = eventtype, Total = total) %>%
  knitr::kable(caption = "Events types and number in the Geonet Database")

## ----fig.cap="Number of earthquakes by Magnitude and Classification"----
eqnz %>% filter(magnitude > 0, depth > 0) %>%
  mutate(rounded_magnitude=round(magnitude,1)) %>%
  group_by(rounded_magnitude, eventtype) %>%
  summarise(subtotal = n()) %>%
  ggplot(aes(x=rounded_magnitude, y=subtotal, fill=eventtype)) +
  geom_bar(stat="identity")

## -----  

# only do all the work if work is not done
# to make calculated variables clear, I name them with an eq_prefix

if (!file.exists("eqdata/eqnz_processed.RData")){
  eqnz$eventtype[eqnz$eventtype == ""] <- "likely earthquake"
  eqnz$longitude[eqnz$longitude > 180] <- eqnz$longitude[eqnz$longitude > 180] - 360
  southmost <- min(eqnz$latitude)
  westmost <- min(eqnz$longitude[eqnz$longitude > 0])

  eqnz <- eqnz %>% filter(
    magnitude > 0, depth > 0, eventtype %in% c("earthquake", "likely earthquake")
  ) %>% rowwise() %>% mutate(
    eq_gridpoint_y = round(distVincentyEllipsoid(c(longitude, southmost),
                                                   c(longitude, latitude)) /50000,0),
    eq_gridpoint_x = round(distVincentyEllipsoid(c(westmost, latitude),
                                                   c(longitude, latitude)) /50000,0),
    eq_roundedlat = destPoint(p=c(longitude, southmost),
                               b=0, d=eq_gridpoint_y*50000)[2],
    eq_roundedlong = destPoint(p=c(westmost, eq_roundedlat),
                               b=90, d=eq_gridpoint_x*50000)[1]) %>% ungroup()
  # use maptools to calculate solar angles
  sun_angles <- solarpos(matrix(c(eqnz$longitude, eqnz$latitude), ncol=2), eqnz$time_UTC)
  colnames(sun_angles) <- c("eq_compass", "eq_vertical")
  eqnz <- cbind(eqnz, sun_angles)
  eqnz$eq_is_night <- eqnz$eq_vertical < 0
  # calculate 360 degree as well as vertical
  eqnz <- eqnz %>% mutate(
    eq_angle_360 = eq_vertical,
    eq_angle_360 = ifelse(eq_compass > 180, 180 - eq_angle_360, eq_angle_360),
    eq_angle_360 = ifelse(eq_vertical < 0 & eq_compass <= 180,
                         360 + eq_angle_360, eq_angle_360),
    eq_angle_by_10 = floor(eq_angle_360 /10) * 10)
  # get length of night earthquake in hours for UTC day occurred in
  sunr <- sunriset(matrix(c(eqnz$longitude, eqnz$latitude), ncol=2),

```

```

eqnz$time_UTC, direction="sunrise", POSIXct.out=TRUE)
suns <- sunriset(matrix(c(eqnz$longitude, eqnz$latitude), ncol=2),
                  eqnz$time_UTC, direction="sunset", POSIXct.out=TRUE)
eqnz$eq_nightlength = 24 - as.numeric(suns$time - sunr$time)
#calculating the solid earth tide vertical component
soletide_date <- Datenum(as.Date(eqnz$time_UTC)) +
  (hour(eqnz$time_UTC) * 3600 + minute(eqnz$time_UTC)*60 + second(eqnz$time_UTC))/86400
etv <- Earthtide(eqnz$latitude, eqnz$longitude, soletide_date)
eqnz$eq_solidearth_vertical <- (t(etv))[,1]
save(eqnz, file="eqdata/eqnz_processed.RData")

rm(list=ls())
}
load("eqdata/eqnz_processed.RData")

## -----
grid_summary <- eqnz %>% group_by(eq_roundedlat, eq_roundedlong) %>%
  summarise(subtotal=n()) %>% ungroup() %>%
  summarise(above_100 = sum(subtotal > 100), total=n())
n_above100 <- grid_summary$above_100[1]
total_grids <- grid_summary$total[1]

## ----fig.cap="Variation in angle of sun by hour of day"-----
boxplot(eqnz$eq_vertical ~ hour(eqnz$time_UTC))

```



# Chapter 3

## Key result

As I have now done all the initial processing of the data, I can make a formal statement about the amount of night and day

### 3.1 Formal Statement

The study data consists of all subsurface earthquake and likely earthquake events of positive magnitude detected within the study area and time period.

The study area is formed by the rectangular area encompassed by -32.47270 degrees latitude north, -49.32512 degrees latitude south, 182.41699 (-177.583) degrees longitude east, and 163.03711 degrees longitude west.

The study time period consists of the 5 full solar years from September 1st 2011 to September 1st 2015.

Of 100279 the events, 56246 occurred at night and 44033 occurred in the day. Night is defined as being when the sun is below the horizon at the time and surface location of the event.

This can be expressed as 56.09 percent of earthquakes occurred at night, or that there are 27.74 percent more earthquakes at night than in the day.

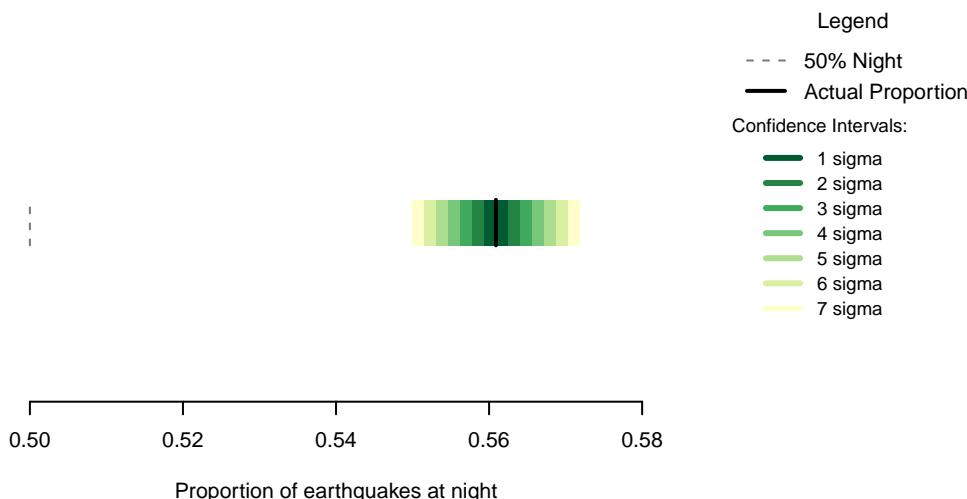


Figure 3.1: Proportion of New Zealand earthquakes occurring at night. n=100279

Given this number of samples and an equal amount of day and night for earthquakes to occur in, if earthquakes are independent events unrelated to the day/night cycle seeing this result by chance is less likely than 1 in a quadrillion (the specific number is too small to be calculated). A seven sigma confidence interval for the proportion of earthquakes occurring at night would be 0.5499 to 0.5718. This confidence interval in no way coincides with 0.5, and using one so large we can confidently say that if earthquakes occur randomly, and there are no biases in our data collection, we would never be seeing this result.

## 3.2 Chapter Code

```
## ----setup, include=FALSE-----
knitr::opts_chunk$set(echo = FALSE)
knitr::opts_chunk$set(warnings=FALSE)
knitr::opts_chunk$set(errors=FALSE)
knitr::opts_chunk$set(message=FALSE)
knitr::opts_chunk$set(dpi = 150)
knitr::opts_chunk$set(fig.width =  6)
knitr::opts_chunk$set(fig.height =  4)

## ----c002_libraries-----
Sys.setenv(TZ = "UTC")
library(binom)
library(dplyr)
# Assumes there is eqnz_processed data created in chapter 2
load("eqdata/eqnz_processed.RData")

## ----c4_002-----
eqtot <- nrow(eqnz)
eqnight <- sum(eqnz$eq_is_night)
eqday <- sum(!eqnz$eq_is_night)
eqdiff <- eqnight - eqday

##
bt <- binom.test(eqnight,eqtot, conf.level= .999999999997440)
eq_national <- eqnz

## ---- fig.cap="Proportion of New Zealand earthquakes occurring at night. n=100279"----
### making the basic proportion graph
eq_night = sum(eq_national$eq_is_night)
eq_total = nrow(eq_national)
bands <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
sigmas <- c(0.682689492137086,
           0.954499736103642,
           0.997300203936740,
           0.999936657516334,
           0.999999426696856,
           0.999999998026825,
           0.999999999997440)
lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
```

```

typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
old_par=par()
conf_steps <- function(x, sigmas=sigmas, night=eq_night, total=eq_total){
  ci_lower <- binom.confint(night, total, method=c("wilson"), conf.level = sigmas[x])[1,5]
  ci_upper <- binom.confint(night, total, method=c("wilson"), conf.level = sigmas[x])[1,6]
  ci_data <- data.frame(step = x, ci_lower, ci_upper)
}
ci_spacing <- lapply(7:1, conf_steps, sigmas=sigmas, night=eq_night, total=eq_total)
ci_steps <- bind_rows(ci_spacing)

layout(matrix(c(1,1,1,2), ncol=4))

par(mar=c(5,6,4,2))
plot(c(min(0.5,floor(100*ci_steps[1,2])/100), max(0.5,ceiling(100*ci_steps[1,3])/100)),
      y=c(-3,8), type="n", bty="n", yaxt="n", ylab="",
      xlab="Proportion of earthquakes at night")
a <- apply(ci_steps, 1, function(x){polygon(c(x[2], x[3], x[3], x[2]),
                                             c(0, 0, 1, 1), col=bands[x[1]], border=NA)})
lines(c(.5,.5), c(0,1), col="#FFFFFF")
lines(c(.5,.5), c(0,1), lty=2, col="#777777")
lines(c(eq_night/eq_total,eq_night/eq_total), c(0,1), lwd=2)

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE)
legend(0,5.5, legend=lbls, lty=typs, lwd=weights, col=bands, bty="n", xjust=0,
       title="Confidence Intervals:", y.intersp=1.1, cex=0.9)

lbls2=c("50% Night", "Actual Proportion")
typs2=c(2,1)
weights2=c(1,2)
cls2=c("#777777", "#000000")
legend(0,7, legend=lbls2, lty=typs2, lwd=weights2, col=cls2, bty="n",
       xjust=0,
       title="Legend", y.intersp=1.2)

par(mar=old_par$mar)
par(mfrow=c(1,1))

```



# Chapter 4

## It is not dependence

The formal statement establishes that, if there are no biases, then it is implausible that these observations are the result of chance. But it does not establish that it is a “True” effect. The observed unusual result could be caused by other things, with the truth being much less unusual. The analysis now moves to testing the data to prove that the observed effect is not other things, ruling out alternative hypotheses.

Earthquakes can occur in swarms- in particular where a large earthquake triggers aftershocks. In this sense, some of the earthquakes are dependent on other earthquakes, so the time the first earthquake occurs at potentially influences latter earthquakes.

In speaking with geologists, this was not generally a major concern as a single night (or day) is a very short amount of time in geological terms. If a large event happens the aftershocks soon start to cross back and forth between day and night as time goes on, and should balance out in the long run. So earthquakes would seem to be technically, but not practically, dependent.

A subtler version of this problem is that the data is dependent enough that a few big events occurring at day, or in the night, bias the data set enough to make what are only moderately unlikely events (a bunch of large earthquakes that occur in the day or night) seem to be much more unlikely (the numerous aftershocks occurring at times conditioned by those few big events).

To test this, we can find the earthquakes that are not part of a swarm, and check the time of day they occurred at. To define a swarm I do so calculations to work out how far apart earthquakes in the same region are.

Using the 50 kilometres per side gridded areas I defined earlier, I calculate the amount of time in hours between earthquakes. This is based on the assumption that I can define a swarm as earthquakes close in both time and space.

The median time between earthquake events in an area is 28 hours. This seems to offer a more useful measure than the mean, which is skewed by some very large gaps. 28 hours seems long enough to me that the events are not part of a swarm that is biasing results by occurring in a single day or night. Using this I can test the data based on the events that occurred in the non-swarm periods, being greater than or equal to 28 hours since the last earthquake.

Among earthquake events that show the least characteristics of being part of a swarm, 55.6% of the 50385 earthquakes occur at night, which is essentially identical to the overall earthquake catalogue result, and

Table 4.1: Time from previous earthquake (hours) per 50km sq. grid

median	mean	earthquakes
28	216	99902

nowhere near 50%. On this basis, the evidence is that the non-swarm earthquakes are the same as the overall population

Another way to consider potential swarm dependencies is to exclude those swarms that would have been generated by aftershocks. To do this I calculate the night time event percentage for events only for months that occurred in grid areas that did not have any magnitude 5+ earthquake events.

Using “not aftershocks in grid zones” as the definition of earthquakes not in swarms, 56.4% of 89279 earthquake events occurred at night. This is fractionally more (but essentially identical to) the general number of night time earthquakes.

Since restricting the data in two different ways to minimise dependence is not changing the results, I can conclude that dependence between earthquake events is not significantly influencing the observed percentage of earthquake events at night.

## 4.1 Formal Statement

Dependencies in the data caused by earthquakes triggering additional earthquakes are not influencing the results.

## 4.2 Chapter Code

```
## ----setup, include=FALSE-----
knitr::opts_chunk$set(echo = FALSE)
knitr::opts_chunk$set(warnings=FALSE)
knitr::opts_chunk$set(errors=FALSE)
knitr::opts_chunk$set(message=FALSE)
knitr::opts_chunk$set(dpi = 150)
knitr::opts_chunk$set(fig.width =  6)
knitr::opts_chunk$set(fig.height =  4)

## -----
Sys.setenv(TZ = "UTC")
library(dplyr)
library(lubridate)
# Assumes there is eqnz_processed data created in chapter 2
load("eqdata/eqnz_processed.RData")

## -
eqnz %>% group_by(eq_roundedlat, eq_roundedlong) %>%
  mutate(timediff = floor(as.numeric(c(NA,diff(time_UTC)))/360)) %>%
  ungroup() %>% filter(!is.na(timediff)) %>%
  summarise(`median` = median(timediff), `mean` = round(mean(timediff),0),
            `earthquakes`=n()) %>%
  knitr::kable(caption="Time from previous earthquake (hours) per 50km sq. grid")

## -
noswarm <- eqnz %>% group_by(eq_roundedlat, eq_roundedlong) %>%
  mutate(timediff = floor(as.numeric(c(NA,diff(time_UTC)))/360)) %>%
  ungroup() %>% filter(!is.na(timediff) & timediff >=28) %>%
  summarise(night = sum(eq_is_night), totalevents=n(),
            percent = round(100 * sum(eq_is_night) / n(),1))
```

```
## -----
tinymag <- eqnz %>%
  group_by(eq_roundedlat, eq_roundedlong, month(time_UTC), year(time_UTC)) %>%
  mutate(monthTopMag = max(magnitude)) %>%
  ungroup() %>% filter(monthTopMag < 5) %>%
  summarise(night = sum(eq_is_night), totalevents=n(),
            percent = round(100 * sum(eq_is_night) / n(),1))
```



# Chapter 5

## It is not masking

The proportion of earthquakes occurring at night changes as the magnitude of the earthquakes changes.

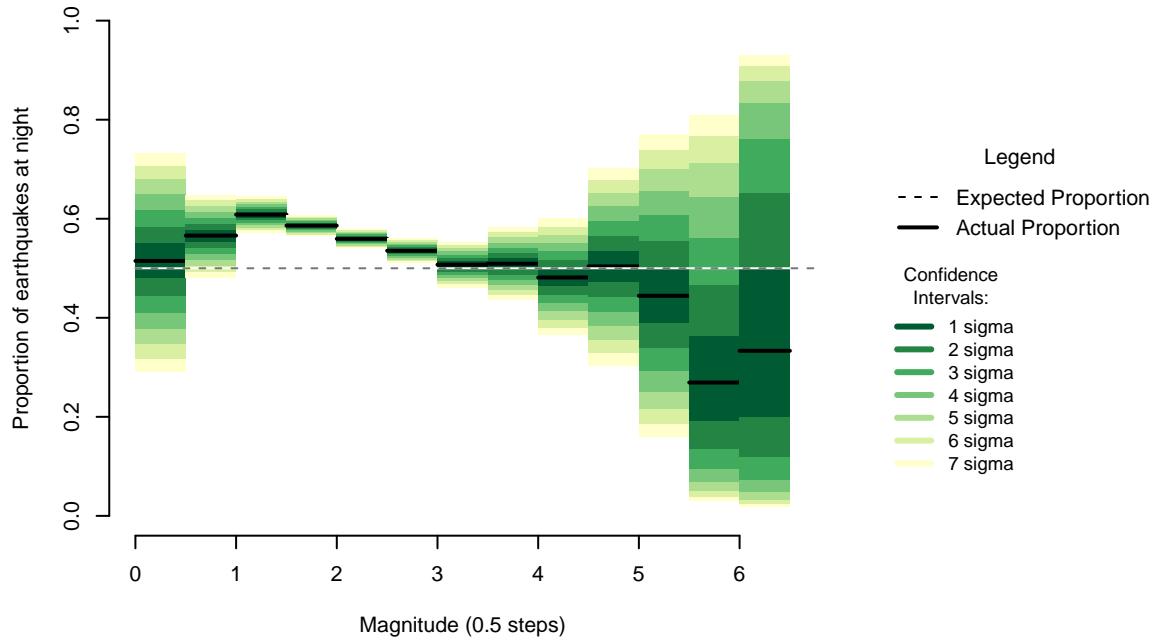


Figure 5.1: Proportion of earthquakes occurring at night by magnitude (0.5 steps)

The size of the confidence intervals around the proportions is determined by the number of earthquake observations within that magnitude. Among magnitude bands with a large number of earthquakes, so small confidence intervals, the proportion of earthquakes is clearly high then falling towards fifty percent.

As the proportion of earthquake events occurring at night decreased with earthquake magnitude, one possible explanation is an unknown mechanism that limits the ability to detect low magnitude earthquakes during the day, creating an illusory difference in overall earthquakes. Atef, Liu, and Gao report cultural noise in seismographic data matching freeway activity and other urban activity Link 1, leading to more earthquakes recorded at night. Similarly, Iwata attributes the masking effects to the decrease in earthquakes in the early and late part of the day and the recovery around noon Link 2.

To see if the masking interpretation is valid, I need to establish a test where the Null Hypothesis is that the difference is caused by masking, and if that test is failed then the effect cannot be caused by masking. After some thinking, the New Zealand data can be limited by sensor location to those seismographs in remote

wilderness areas and lightly populated offshore islands, all of which lack freeways and urban areas within sensor range for low magnitude earthquakes. If there is a masking pattern caused by human activity in the day, the sensors in the remote location would be expected to have a 50/50 balance of night and day earthquakes.

For New Zealand Earthquakes, the places an earthquake was detected from can be obtained from a second data source at Geonet Link 3 by making a separate query for each earthquake.

The two seismographs with the least cultural noise to potentially interfere with records are Raoul Island and Puyesgur Point.

The Raoul Island seismograph is located on an island group with no permanent inhabitants, no urban areas or roads, and around 1000km from both Tonga and the New Zealand mainland Link 4.

Puyesgur Point is an automated lighthouse in a remote corner of the Fiordland National park, approximately 150 kilometres from any city greater than 50,000 people and approximately 550 kilometres from any city with a population greater than 120,000 people. The 2013 census records less than 10 people within 60 kilometres of the seismograph in the roadless national park Link 5.

Selecting events detected by the Raoul Island (RIZ) Puyesgur Point (PYZ) yields a set of 2090 events, 1136 of which occurred at night. From this we can reject (95% (2 Sigma) confidence interval 0.522 to 0.565) the hypothesis that among seismographs in areas with no cultural activity there are as many day earthquakes as night. So, in turn, we reject the hypothesis that cultural activities are responsible for the difference between day and night activities. The 2 sigma level of confidence reflects that, though the result is not 0.5, I cannot be as confident with this amount of data (though I am still confident)

Though that rules out cultural effects masking earthquakes, there still might be mysterious, unknown daytime masking effect. While it is impossible to prove the existence of such a mechanism, after some thought it is possible to disprove its existence. To test if the difference between day and night can be explained by an unknown confounder affecting the range at which events are detected, I can limit the data set to only those events that must have been detectable during the day. The median distance to the closest seismograph an event was detected at during the day for events of magnitude 0.3 to 0.5 (inclusive) gives a distance that events, if they were of greater magnitude and occurred within that distance to a seismograph, should be equally detectable in day or night- If earthquakes are stronger and closer than typically detected daytime earthquakes, they can be assumed to not be masked.

By discarding all events outside that distance as potentially unable to be detected in the day, of the 9127 events that remained, 5059 occurred at night. From that we can strongly reject (5 Sigma confidence interval of the proportion is 0.544 to 0.565) the hypothesis that equal numbers of earthquakes occur in the day and at night among earthquakes at a range that can be detected in the day.

## 5.1 Formal Statement

The hypothesis that daytime masking is causing differences in the measured number of night and day earthquakes is rejected on two grounds. Seismographs in remote locations show no evidence of a lack of daytime masking.

## 5.2 Links

Link 1 - Atef, A. H., Liu, K. H. & Gao, S. S. Apparent Weekly and Daily Earthquake Periodicities in the Western United States. *Bulletin of the Seismological Society of America* 99, 2273–2279 (2009).

Link 2 - Iwata, T. in *Earthquake Research and Analysis - New Advances in Seismology* (ed. Damico, S.) (InTech, 2013).

Link 3 - Geonet Simple Queries <http://info.geonet.org.nz/display/appdata/Simple+Queries>

Link 4 - Raoul Island [https://en.wikipedia.org/wiki/Raoul\\_Island](https://en.wikipedia.org/wiki/Raoul_Island)

Link 5 - Puysegur Point [https://en.wikipedia.org/wiki/Puysegur\\_Point](https://en.wikipedia.org/wiki/Puysegur_Point) and the 2013 census results <http://www.stats.govt.nz/StatsMaps/Home/People%20and%20households/2013-census-quickstats-about-a-place-map.aspx>

## 5.3 Chapter Code

```
## ----setup, include=FALSE-----
knitr::opts_chunk$set(echo = FALSE)
knitr::opts_chunk$set(warnings=FALSE)
knitr::opts_chunk$set(errors=FALSE)
knitr::opts_chunk$set(message=FALSE)
knitr::opts_chunk$set(dpi = 150)
knitr::opts_chunk$set(fig.width = 6)
knitr::opts_chunk$set(fig.height = 4)

## ----c002_libraries-----
Sys.setenv(TZ = "UTC")
library(dplyr)
# Assumes there is eqnz_processed data created in chapter 2
load("eqdata/eqnz_processed.RData")
eq_national <- eqnz

## ----fig.cap="Proportion of earthquakes occurring at night by magnitude (0.5 steps)"---
old_par=par()
grf <- eq_national %>% mutate(floored_mag = floor(magnitude*2)/2) %>%
  group_by(floored_mag) %>% summarise(successes = sum(eq_is_night), trials=n())

poly_conf_int <- function(success, trials, aa, stepsize, sigma, colr){
  ci <- binom.confint(success, trials, method=c("wilson"), conf.level = sigma)
  lower <- ci[1,5]
  upper <- ci[1,6]
  a <- polygon(x=c(aa,aa+stepsize,aa+stepsize,aa), y=c(upper,upper,lower,lower),
                col=colr, border=NA)
}

plot7sig <- function(success, trials, aa, stepsize){
  library(binom)
  #bands <- c('#ffffb2', '#fed976', '#feb24c', '#fd8d3c', '#fc4e2a', '#e31a1c', '#b10026')
  bands <- rev(c('#ffffcc', '#d9f0a3', '#add8e6', '#78c679', '#41ab5d', '#238443', '#005a32'))
  sigmas <- c(0.682689492137086,
             0.954499736103642,
             0.997300203936740,
             0.999936657516334,
             0.999999426696856,
             0.99999998026825,
             0.9999999997440)
  sapply(7:1, function(x){
    poly_conf_int(success, trials, aa, stepsize, sigmas[x], bands[x])})
```

```

    a <- lines(c(aa, aa + stepsize), c(success/trials, success/trials), lwd=2)
}

lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
clrs <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
#clrs <- c('#fffffb2', '#fed976', '#feb24c', '#fd8d3c', '#fc4e2a', '#e31a1c', '#b10026')

layout(matrix(c(1,1,1,2), ncol=4))
plot(x=c(0,max(grf$floored_mag)+0.5), y=c(0,1), type="n", bty="n",
      xlab="Magnitude (0.5 steps)", ylab="Proportion of earthquakes at night")
a <- apply(grf, 1, function(x){plot7sig(x[2],x[3],x[1],0.5)})
lines(c(0,10), c(.5,.5), col="#FFFFFF")
lines(c(0,10), c(.5,.5), lty=2, col="#777777")

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE)
legend(0.5, legend=lbls, lty=typs, lwd=weights, col=clrs,
       bty="n", xjust=0,
       title="Confidence Intervals:", cex=0.9)
lbls=c("Expected Proportion", "Actual Proportion")
typs=c(2,1)
weights=c(1,2)
legend(0.7, legend=lbls, lty=typs, lwd=weights, bty="n", xjust=0,
       title="Legend", y.intersp=1.2)

par(mar=old_par$mar)
par(mfrow=c(1,1))

## -----
## Becuase this involved 100000 downloads (though very small ones)
## from geonet, I only wanted to do it once
## hence the only downloading if saved file of results does not exist.
## there are getting on for 2.368 million rows in the final data set
## and it takes many hours to assemble (though is only 3.9 MB in the end)
## if you want to replicate this and save time, and trust me,
## you can load the shortlists later instead of running this section.
if (!file.exists("eqdata/nz_detection_sites.RData")){
get_pick_data <- function(x){
  csv_url <- paste("http://quakeml.geonet.org.nz/csv/1.0.0/", x, "/picks", sep="")
  eq_data <- read.csv(csv_url, stringsAsFactors = FALSE)
  station_data <- data.frame(station= unique(eq_data$station), stringsAsFactors=FALSE)
  station_data$eq_id <- x
  return(station_data)
}

```

```
# when I checked, 4 earthquakes had no pick information
no_pick_info <- c("3571857", "2012p569703", "2012p813511", "2013p613798")
eq_ids <- eqnz$publicid[!(eqnz$publicid %in% no_pick_info)]
nz_station_info <- bind_rows(lapply(eq_ids, get_pick_data))
save(nz_station_info, file="eqdata/nz_detection_sites.RData")
}

rm(list=ls())
load("eqdata/eqnz_processed.RData")
load("eqdata/nz_detection_sites.RData")

RIZ_list <- nz_station_info$eq_id[nz_station_info$station == "RIZ"]
PYZ_list <- nz_station_info$eq_id[nz_station_info$station == "PYZ"]

no_cultural_influence <- unique(c(RIZ_list, PYZ_list))
tot_qt_nt <- sum(eqnz$eq_is_night[eqnz$publicid %in% no_cultural_influence])
tot_qt <- length(no_cultural_influence)
bt <- binom.test(tot_qt_nt, tot_qt)

## -----
threshold <- median(eqnz$minimumdistance[eqnz$magnitude >= 0.3 & eqnz$magnitude <= 0.5])
detectable <- eqnz %>% filter(magnitude > 0.5, minimumdistance <= threshold) %>%
  summarise(at_night = sum(eq_is_night), total=n(), proportion= at_night/total)
bt <- binom.test(detectable$at_night, detectable$total)
```



# Chapter 6

## It is not seasonal

The amount of night within a 24 hour period changes over the course of a year, so a possible explanation is that if earthquakes are more likely to occur in winter, and winter nights are longer, then there would overall be more earthquakes at night. This would be an interesting finding in itself, but it would make the daily pattern the result of a seasonal one, and in that case I would need to go back and consider the position of the sun on a seasonal north-south in addition to the daily east-west basis.

In calculating new variables from the data, I derived the length of night for each 24 hour period that an earthquake took place in. I can test this amount of night against the number of earthquakes that occurred in the available night.

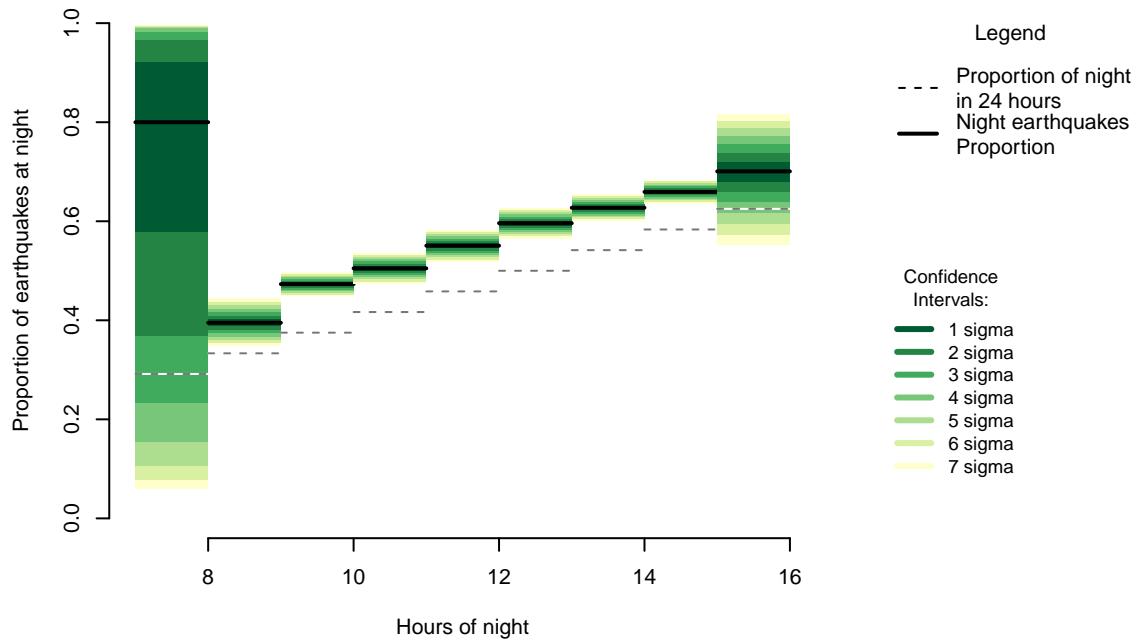


Figure 6.1: Proportion of night earthquakes by length of night. n = 100279

As (for all hours we can be confident about) there is a consistent level of nightly earthquake oversupply, I can rule out seasonal differences being the cause.

This is not a formal statistical test, because I don't think it needs it. The oversupply of nighttime earthquakes is neither rising or falling across the season's change of night lengths, so there is no evidence of an association with seasonal change.

Table 6.1: Length of night and number of earthquakes

hours of night	night quakes	total quakes	% hours of night	% quakes at night
7	4	5	29	80
8	1976	5003	33	39
9	10067	21281	38	47
10	6513	12899	42	50
11	6555	11904	46	55
12	7591	12734	50	60
13	10150	16181	54	63
14	13006	19724	58	66
15	384	548	62	70

## 6.1 Formal statement

There is no evidence that the occurrence of night earthquakes are influenced by seasonal changes. As a consequence this analysis focuses on change within a 24 hour period.

## 6.2 Chapter Code

```
## ----setup, include=FALSE-----
knitr:::opts_chunk$set(echo = FALSE)
knitr:::opts_chunk$set(warnings=FALSE)
knitr:::opts_chunk$set(errors=FALSE)
knitr:::opts_chunk$set(message=FALSE)
knitr:::opts_chunk$set(dpi = 150)
knitr:::opts_chunk$set(fig.width =  6)
knitr:::opts_chunk$set(fig.height =  4)

## ----c002_libraries-----
Sys.setenv(TZ = "UTC")
library(dplyr)
library(ggplot2)
library(lubridate)
library(mapproj)
library(binom)
library(parallel)
library(plotrix)
library(solidearthtide)
library(tidyr)
# Assumes there is eqnz_processed data created in chapter 2
load("eqdata/eqnz_processed.RData")
old_par <- par()

lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
typs <- c(1,1,1,1,1,1,1)
```

```

weights <- c(3,3,3,3,3,3,3)
clrs <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))

## ---- fig.cap="Proportion of night earthquakes by length of night. n = 100279"----
hrs <- eqnz %>% mutate(hour_band=floor(eq_nightlength)) %>%
  group_by(hour_band) %>%
  summarise(eq_nt = sum(eq_is_night), eq_all = n()) %>%
  mutate(nt_prop = hour_band/24, eq_prop = eq_nt/ eq_all)

old_par=par()
grf <- hrs
poly_conf_int <- function(success, trials, aa, stepsize, sigma, colr){
  ci <- binom.confint(success, trials, method=c("wilson"), conf.level = sigma)
  lower <- ci[1,5]
  upper <- ci[1,6]
  a <- polygon(x=c(aa,aa+stepsize,aa+stepsize,aa), y=c(upper,upper,lower,lower),
                col=colr, border=NA)
}
}

plot7sighr <- function(success, trials, aa, stepsize, expected){
  library(binom)
  #bands <- c('#ffffb2', '#fed976', '#feb24c', '#fd8d3c', '#fc4e2a', '#e31a1c', '#b10026')
  bands <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
  sigmas <- c(0.682689492137086,
             0.954499736103642,
             0.997300203936740,
             0.999936657516334,
             0.999999426696856,
             0.999999998026825,
             0.99999999997440)
  sapply(7:1, function(x){
    poly_conf_int(success, trials, aa, stepsize, sigmas[x], bands[x]))
  a <- lines(c(aa, aa + stepsize), c(success/trials, success/trials), lwd=2)
  a <- lines(c(aa, aa + stepsize), c(expected,expected), col="#FFFFFF")
  a <- lines(c(aa, aa + stepsize), c(expected,expected), lty=2, col="#777777")
})

lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
clrs <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
#clrs <- c('#ffffb2', '#fed976', '#feb24c', '#fd8d3c', '#fc4e2a', '#e31a1c', '#b10026')

layout(matrix(c(1,1,1,2), ncol=4))

```

```

plot(x=c(min(grf$hour_band),max(grf$hour_band)+1), y=c(0,1), type="n",
      bty="n", xlab="Hours of night", ylab="Proportion of earthquakes at night")
a <- apply(grf,1,function(x){plot7sighr(x[2],x[3],x[1],1,x[4])})

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE)
legend(0,5, legend=lbls, lty=typs, lwd=weights, col=clrs, bty="n", xjust=0,
       title="Confidence\nIntervals:", cex=0.9)
lbls=c("Proportion of night\nin 24 hours", "Night earthquakes\nProportion")
typs=c(2,1)
weights=c(1,2)
legend(0,9, legend=lbls, lty=typs, lwd=weights, bty="n", xjust=0,
       title="Legend", y.intersp=1.2)

par(mar=old_par$mar)
par(mfrow=c(1,1))

## -----
hrs %>% mutate(`% hours of night` = 100 * round(nt_prop,2),
               `% quakes at night` = 100 * round(eq_prop,2)) %>%
  select(`hours of night`=hour_band,
         `night quakes` = eq_nt,
         `total quakes` = eq_all,
         `% hours of night`,
         `% quakes at night`) %>%
knitr::kable(caption="Length of night and number of earthquakes")

```

# Chapter 7

## It is not the moon

The question “could it be the moon” tends to come from people who have heard a lot of commentary about the moon and earthquakes. But the moon’s daily rotation around the earth is not the same as the sun’s rotation, taking on average 24 hours and 50 minutes for full rotational cycle. This extended period compared to the solar day means that, over long periods of time, the moon is present in the day as much as the moon is present at night. So the effects of the moon are present in the day as much as the night. I have chosen to omit several hundred lines of code demonstrating empirically this is the case, on the grounds that it is not logically possible for it to be otherwise, so the code was a bit of a distraction.

The moon is the principle component of the solid earth tide, the deformation of the globe caused by tidal forces acting on the molten interior of the earth. This gravity-based effect causes displacement of the ground on an approximately 12.5 hour repeating cycle, with typical displacement of the ground around half a metre in that period. Though this is regular geological activity and stress, the pattern of earthquake frequency in no way resembles a 12.5 hour repeating cycle, so this is not solely the solid earth tide.

Up until now, I have been describing day and night as general categories, and asserting the moon does not make logical sense. Rather than looking at earthquakes in a general night/ day distinction we can focus on more specific 10 degree arcs of the sun.

By eye, there is a difference of thousands in the time a little either side of dawn and dusk with little difference (or earthquakes) at other times. As one form of day/ night comparison, I can compare each group with that 180 degrees apart.

Assuming it is valid to compare day against night with a 180 degree rotation (which I don’t actually know, but is a handy assumption), there does seem to more earthquakes at night than the day when the sun is in the early or late part of the orbit relative to the horizon rather than in the middle of the day or night.

But I haven’t established that rotating day is a reasonable basis for comparison, it is just useful.

A better approach is to figure out how many earthquakes are expected in each arc of the sun, and how many actually occurred. I can do this by calculating what angle the sun was at for a solar year for each earthquake site, and adding up how many occurrences at each angle. This gives the percentage of time the sun was in each arc of the sky. If the number of earthquakes (as a percentage of the total earthquakes) is greater than the percentage of time for the earthquakes to occur (in that arc of the sun) then there are more earthquakes than expected.

While the sun arcs for each individual site can be calculated, it takes many hours and the calculations get longer as the data grows. I can dramatically speed up the calculations by making a few simplifying assumptions

- 1) Given the size of the planet, the sun angles are very similar for sites with fifty kilometres. So rather than calculating the sun angles for each site, I can calculate the sun angles for the 50 km square grid points.

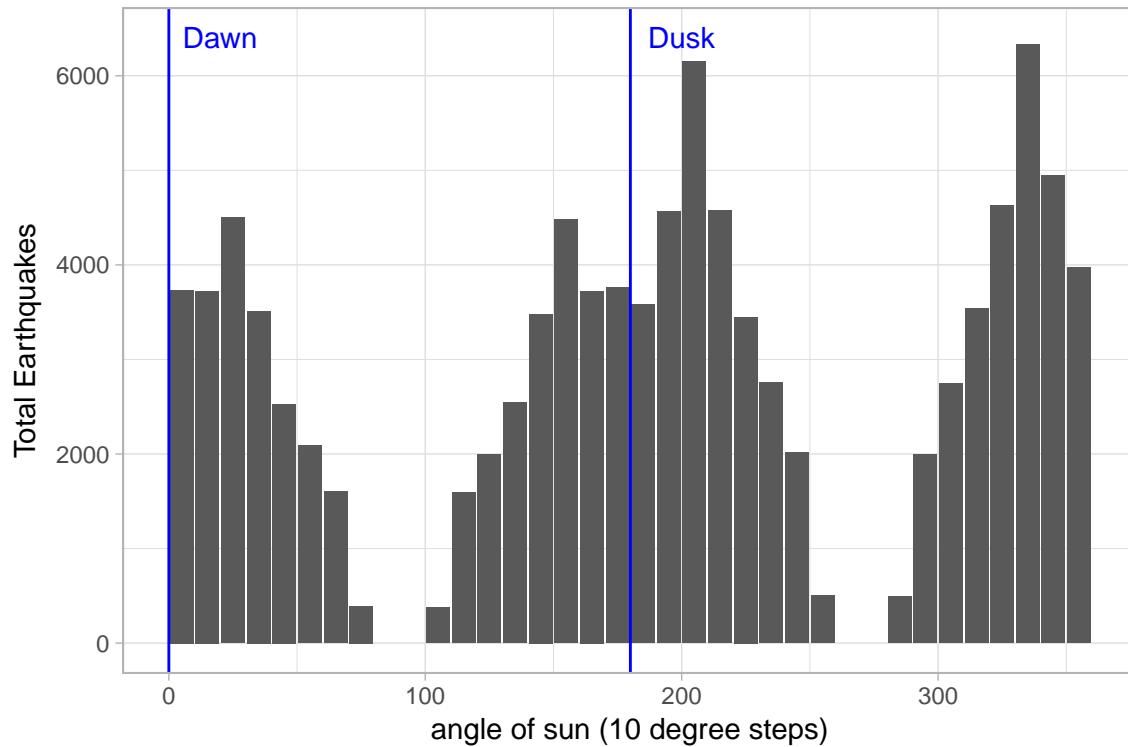


Figure 7.1: Number of earthquakes by 10 degree arcs of angle of the sun at time of earthquake

Table 7.1: 180 degree comparison of day and night

day angle	day_n	night_n	difference
0	3735	3580	-155
10	3723	4565	842
20	4501	6150	1649
30	3512	4574	1062
40	2529	3449	920
50	2091	2760	669
60	1599	2014	415
70	393	504	111
100	376	497	121
110	1595	1993	398
120	1993	2749	756
130	2542	3535	993
140	3481	4630	1149
150	4479	6329	1850
160	3724	4947	1223
170	3760	3970	210

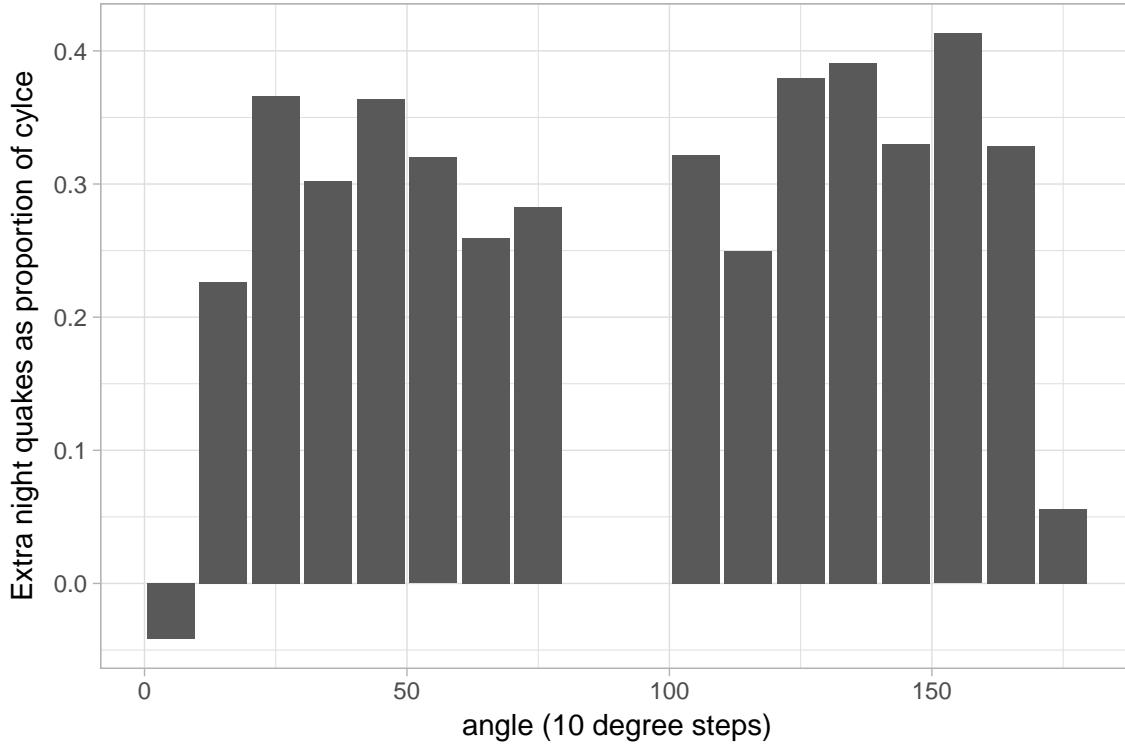


Figure 7.2: excess of night compared to 180 different sun position, as proportion of number of earthquakes in the day position

- 2) While potential sun angles change with latitude, over the long term longitude is displaced slightly in time but will add to the same total, so I only need to calculate the latitude steps.

I checked this, and found it gave me the same overall answers as doing all the individual calculations, so I am taking the gridded approach here. I am also using the parallel library in R to speed up the grid calculations over several cores on my computer. Your computer may not be the same in this respect.

By combining the gridded counts together with the original data I can get an observed versus expected numbers for each arc. There is a choice of ways of thinking about the problem here. We can think of the differences shown in absolute terms where there are a certain number of earthquakes above or below the expected amount, or in relative terms where there are more or less earthquakes in proportion to the number of earthquakes that happen.

## 7.1 Absolute difference

From comparing expected with observed, it is the case that there are both fewer earthquakes than expected during the day and more earthquakes than expected at night. To get a sense of how reliable these differences are, I can add confidence intervals to the observed data, to give a sense of how far away we might expect the “true” expected proportion to be.

Compared against an expected value based on all earthquakes and all available time, earthquakes are unusually low during the day and night. Except for around dawn and dusk (when it crosses from day to night) and near the middle of the night when there are very few earthquakes. This difference in actual versus expected earthquakes can be better seen by making a graph based on how far from the expected amount is each 10 degree sun arc.

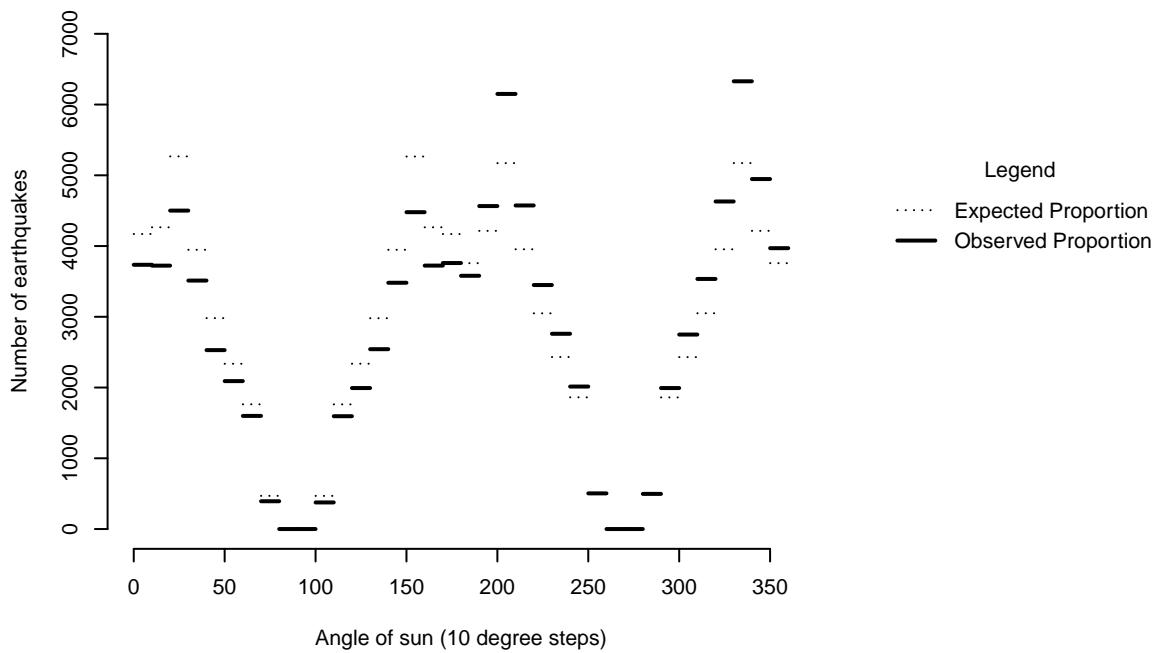


Figure 7.3: Comparison of expected to observed earthquakes by angle of the sun

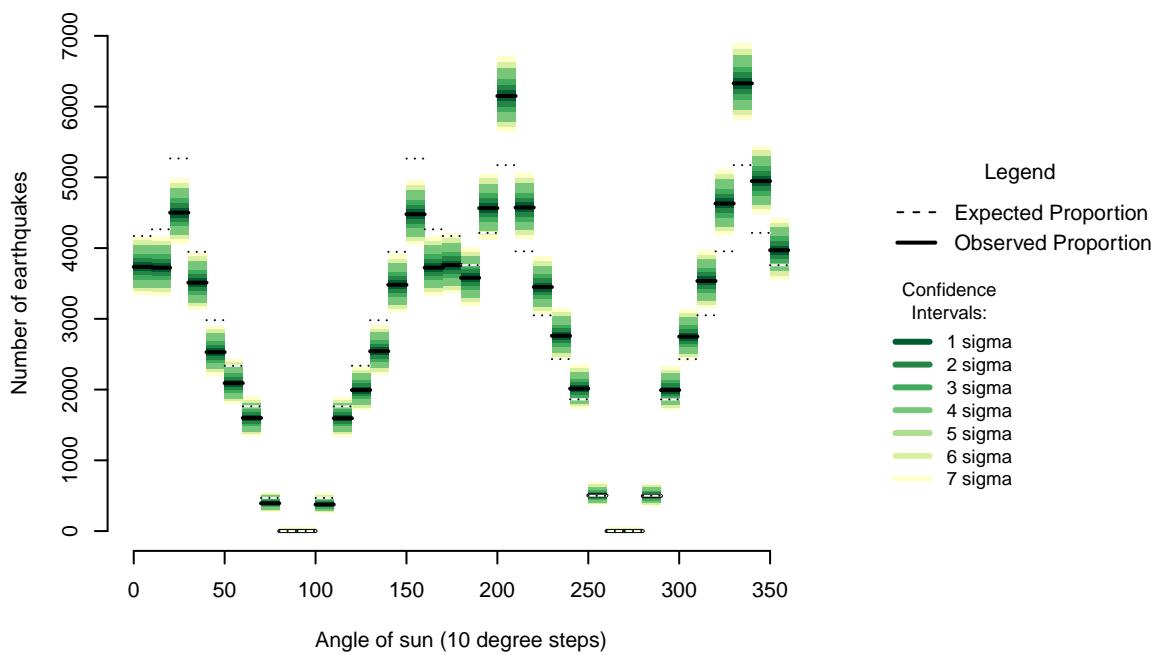


Figure 7.4: Comparison of expected to observed (with uncertainty) earthquakes by angle of the sun

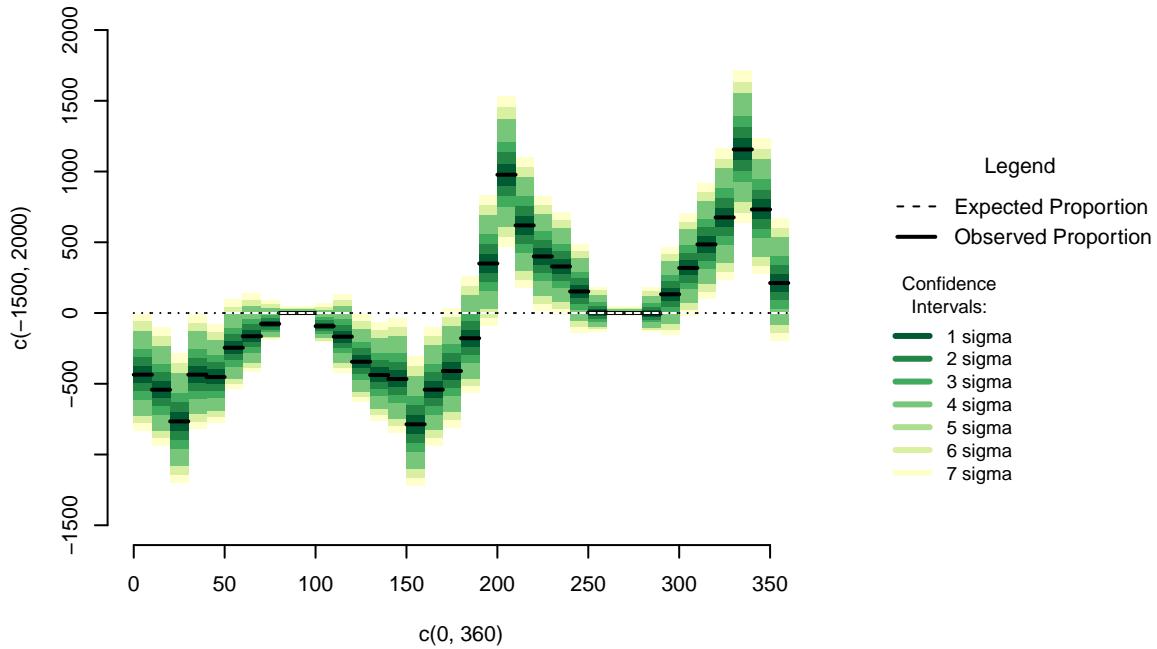


Figure 7.5: Difference of observed earthquake numbers from expected by 10 degree sun arc

On the graph the locations around 30 degrees away from the horizon seem to be particularly unusual points. However, the orbit of the sun is not linear, it is circular(-ish)

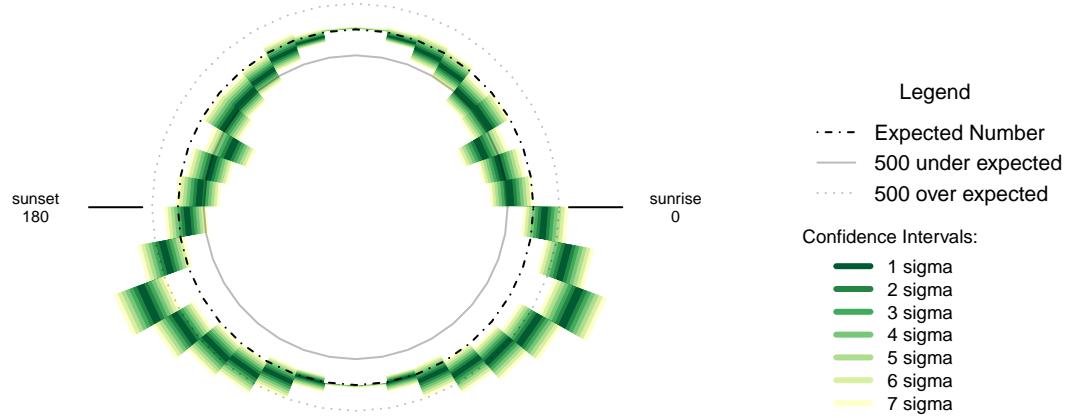


Figure 7.6: Oversupply of earthquakes compared to expected number in sun's orbit (10 degree steps)

While the areas 30 degrees above the horizon stand out as a little out of line with the overall pattern in their region, the overall shape is being most influenced by an overabundance of earthquakes focused on thirty degrees below the horizon.

This is not a lunar pattern. Even taking the solar component of the solid earth tide, it is not the same pattern. The solid earth tide contracts the local ground when the gravitational sources are near the horizon, and expands the earth when the gravitational sources are at midnight or midday.

## 7.2 Relative

A second way of looking at the difference of expected to actual earthquakes is in the relationships between rising counts of earthquakes.

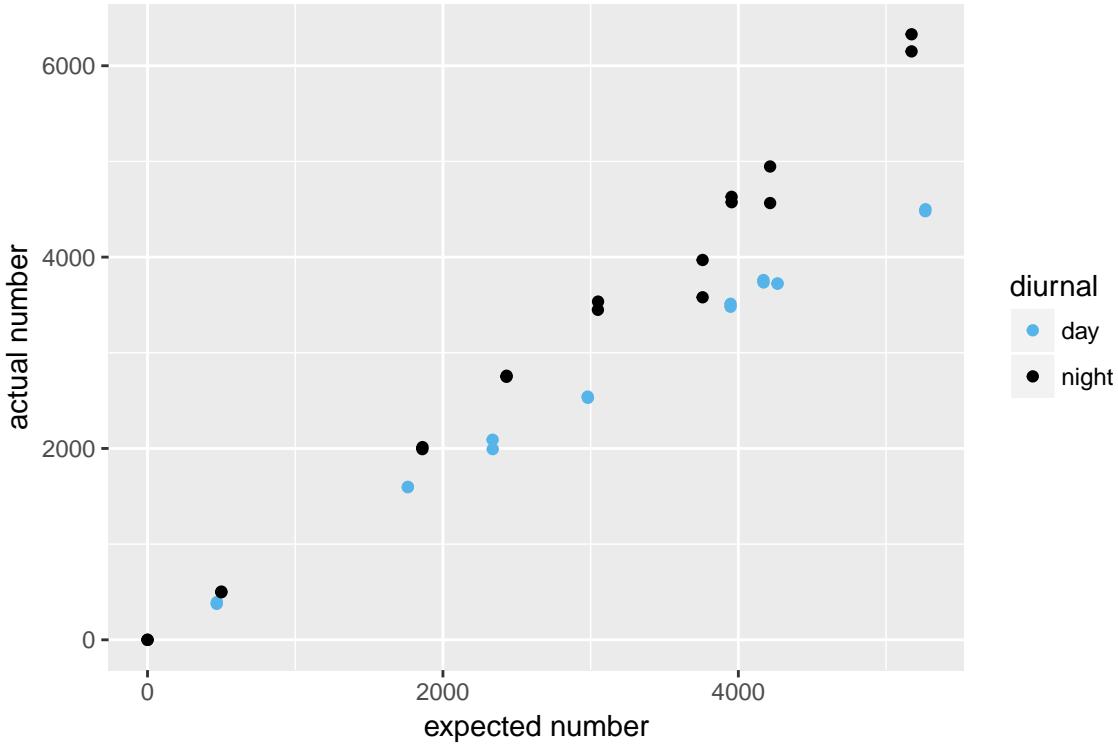


Figure 7.7: Actual vs expected number of earthquakes by diurnal cycle

The points do seem to be forming two lines that diverge as the number of earthquakes increase. For this reason, I think it makes sense to calculate a different line of best fit for both night and day.

Starting from zero, as with zero opportunities for earthquakes I am assuming there will be zero earthquakes, there are 0.872751 earthquakes in the day for “expected” quake. The Adjusted R-squared value of 0.9995 means that this line is an almost perfect fit to the data. For the night earthquakes there are 1.13761 earthquakes for each expected quake. The Adjusted R-squared value of 0.9959 is a line not quite as perfectly fitting as the day, but still very perfect.

By way of cross-checking the night earthquakes were 1.303476 as common as day ones, which accords with the overall values of night versus day. The good fit of the lines suggests that this is a good model for the data.

After fitting a linear model like this, it is standard practice to check the residuals- to see if there is a pattern to the amount of variation to the line of best fit.

Viewed in degree order from 0 to 360 degrees, there are peaks in the early hours of the night.

The over pattern is of unusual numbers of earthquakes peaking when the sun is around 30 degrees below the horizon, with daytime showing no unusual patterns. This result is broadly similar to that of the absolute value proportions for each arc. The pattern is reflectively symmetrical around midnight or midday based on unusual early/late night peaks.

The irregular residuals occurring at night, with the day being completely regular, is interesting by itself. It suggests that the unusual earthquake behaviour is nightly rather than daily.

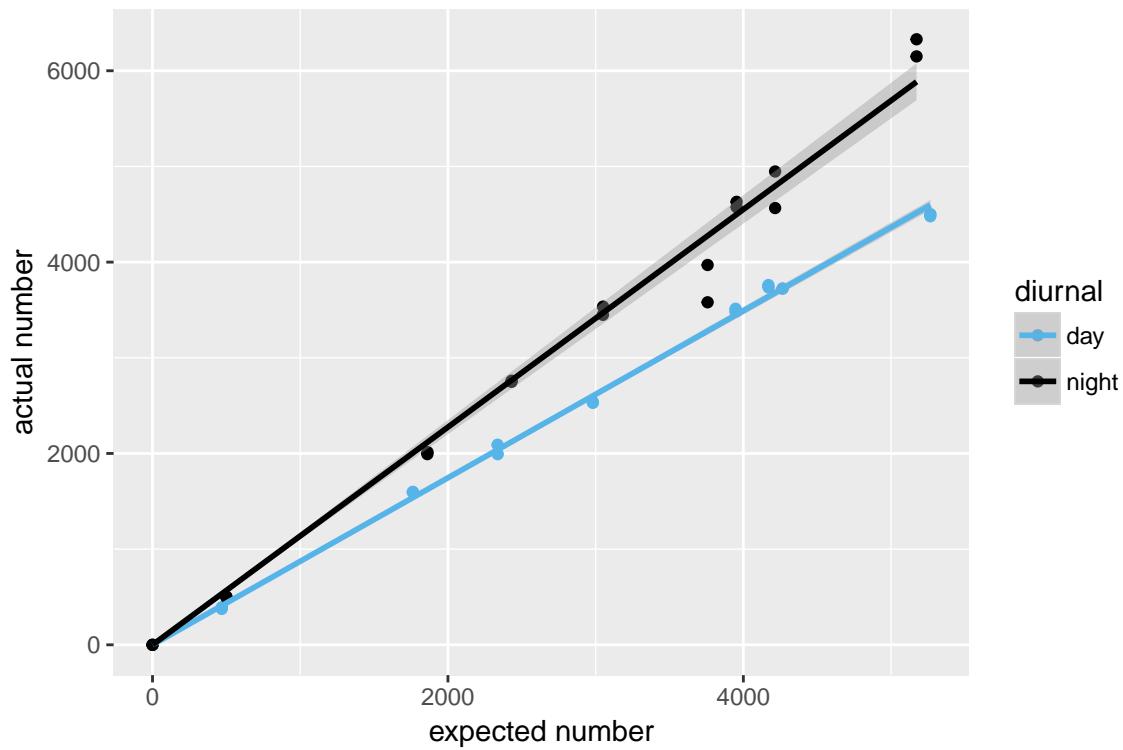


Figure 7.8: Best fit lines of actual vs expected number of earthquakes

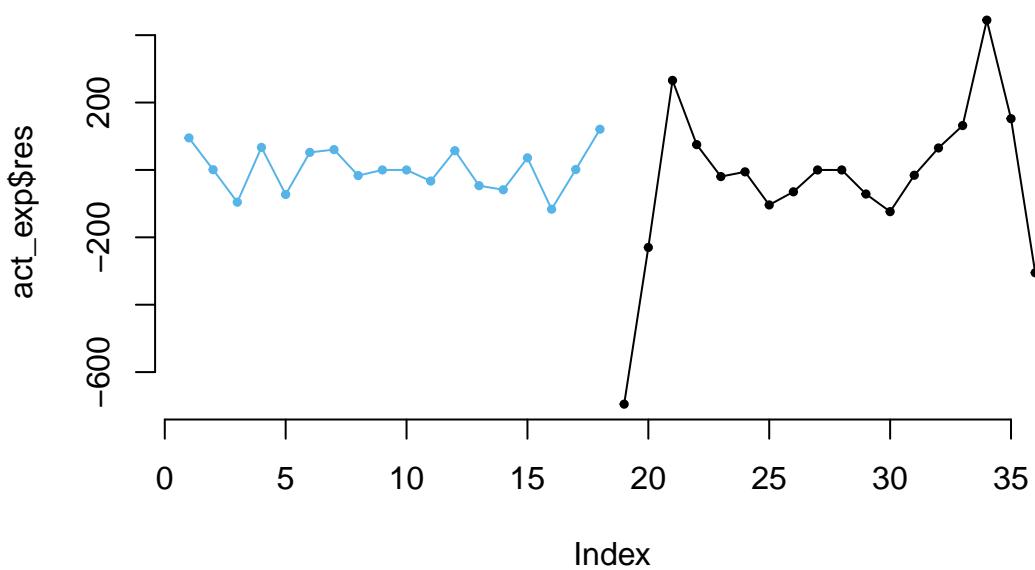


Figure 7.9: Variation in raw residuals

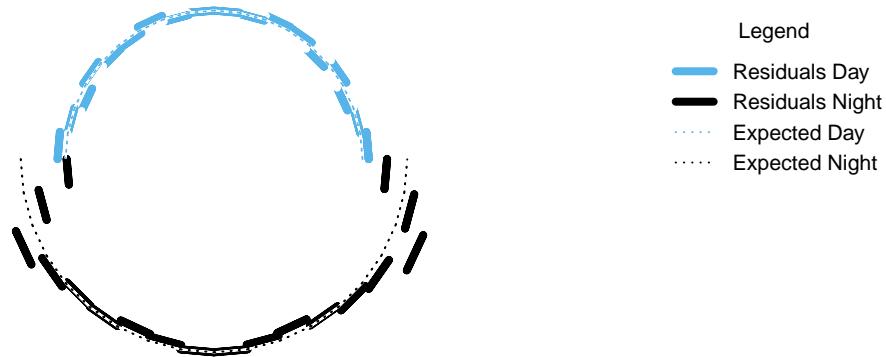


Figure 7.10: Rotational variation in residuals. The nightly residuals have been displaced by the mean difference between day and night, to reflect the different between the two lines

Table 7.2: Number of earthquakes by day/night cycle and condition of crust

solidearth	diurnal	total
contracted	day	25026
contracted	night	35951
expanded	day	19007
expanded	night	20295

This symmetry does not at all match the lunar component of the solid earth tide, nor does it match the much weaker solar component. The earth's crust is contracted when the source of gravity is near the horizon (for the sun, dusk and dawn), and expanded when the source of gravity is directly above or below (for the sun, midnight and midday). A peak only at night, about halfway between the maximum solid earth tide effects does not match this pattern.

But the solid earth tide is affecting the pattern of earthquakes. Categorising time periods into night and day for the sun and if the earth's crust is in a contracted or expanded state from the solid earth tide, if the events were independent there should be a similar number of earthquakes in each category.

Because the earth is not a perfect sphere, in New Zealand the crust is in a contracted state most of the time relative to the baseline of the hypothetical sphere, so the number of earthquakes when the solid earth is contracted is not expected to be the same as when it is expanded. However, the earth is contracted (and expanded) for the same amount of time in both night and day. The big difference in the number of earthquakes between night and day when the earth is contracted, compared to the small difference in the number of earthquakes between night and day when the earth is expanded, suggests that the solid earth tide does have some effect on the process, even though the tide is not the primary cause.

This monograph is not about the solid earth tide effects, so I am only noting that the solid earth tide has a significant effect on the pattern of earthquakes. To my knowledge no-one has found this effect before.

### 7.3 Formal Statement

The diurnal pattern of earthquakes is inconsistent with the moon being the direct cause. However, the solid earth tide, of which the moon is the major component, has a significant role in the frequency patterns of earthquakes.

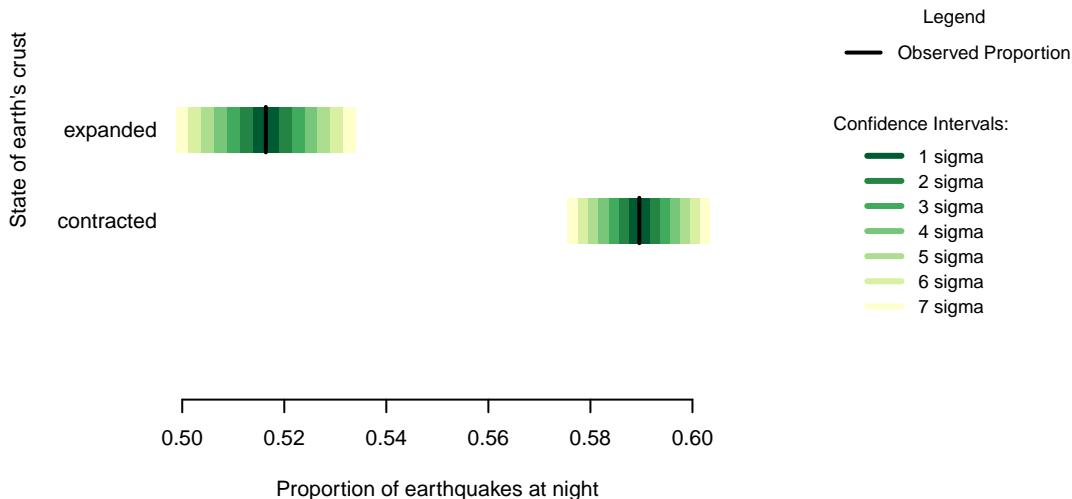


Figure 7.11: Solid earth tide and proportion of earthquakes at night

## 7.4 Links

1 - [http://oceanservice.noaa.gov/education/kits/tides/media/supp\\_tide05.html](http://oceanservice.noaa.gov/education/kits/tides/media/supp_tide05.html)

## 7.5 Chapter Code

```
## ----setup, include=FALSE-----
knitr:::opts_chunk$set(echo = FALSE)
knitr:::opts_chunk$set(warnings=FALSE)
knitr:::opts_chunk$set(errors=FALSE)
knitr:::opts_chunk$set(message=FALSE)
knitr:::opts_chunk$set(dpi = 150)
knitr:::opts_chunk$set(fig.width = 6)
knitr:::opts_chunk$set(fig.height = 4)

## ----c002_libraries-----
Sys.setenv(TZ = "UTC")
library(dplyr)
library(ggplot2)
library(lubridate)
library(mapproj)
library(binom)
library(parallel)
library(plotrix)
library(solidearthtide)
library(tidyr)
# Assumes there is eqnz_processed data created in chapter 2
load("eqdata/eqnz_processed.RData")
old_par <- par()

lbls <- c(
  "1 sigma", "2 sigma",
```

```

  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
clrs <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
circlesize=500

## ----fig.cap=
##"Number of earthquakes by 10 degree arcs of angle of the sun at time of earthquake"----
by_angle <- eqnz %>%
  group_by(eq_angle_by_10) %>% summarise(total= n()) %>%
  mutate(daynight=ifelse(eq_angle_by_10 < 180, "day", "night"))

ggplot(data=by_angle, aes(x=eq_angle_by_10+5, y=total)) +
  geom_bar(stat="identity") +
  xlab("angle of sun (10 degree steps)") + ylab("Total Earthquakes") +
  geom_vline(xintercept = c(0,180), colour="blue") +
  annotate("text", x = 20, y = 6400, label = "Dawn", color="blue") +
  annotate("text", x = 200, y = 6400, label = "Dusk", color="blue") + theme_light()

## -----results="asis"-----
only180 = by_angle[1:16,]
only180$night = by_angle$total[17:32]
only180$difference = only180$night - only180$total
only180$asprop = only180$difference / only180$total
only180 %>% select(`day angle`=eq_angle_by_10, day_n=total,
                     night_n=night, difference) %>% knitr::kable(
  caption="180 degree comparison of day and night")

## ---- fig.cap="excess of night compared to 180 different sun position, as proportion
## of number of earthquakes in the day position"----
ggplot(data=only180, aes(x=eq_angle_by_10+5, y=asprop)) +
  geom_bar(stat="identity") +
  xlab("angle (10 degree steps)") + ylab("Extra night quakes as proportion of cylce") +
  theme_light()

## -----
if (!file.exists("eqdata/eqnz_expected.RData")){
lat_range <- unique(eqnz$eq_roundedlat)
long_med <- median(eqnz$eq_roundedlong)

# 1 minute intervals for a full solar year
time1 <- ymd_hms("2015-01-01 00:00:00")
time2 <- ymd_hms("2015-12-31 23:59:00")
time_sq <- seq.POSIXt(from=time1, to=time2, by="min")

calc_angs <- function(x, longinput, timeinput){
  library(dplyr)
  sun_angles <- maptools::solarpos(matrix(c(longinput, x), ncol=2), timeinput)
  colnames(sun_angles) <- c("eq_compass", "eq_vertical")
}

```

```

# calculate 360 degree as well as vertical
site_summary <- as.data.frame(sun_angles) %>%
  mutate(eq_angle_360 = eq_vertical,
        eq_angle_360 = ifelse(eq_compass > 180, 180 - eq_angle_360, eq_angle_360),
        eq_angle_360 = ifelse(eq_vertical < 0 & eq_compass <= 180,
                              360 + eq_angle_360, eq_angle_360),
        eq_angle_by_10 = floor(eq_angle_360 /10) * 10) %>%
  group_by(eq_angle_by_10) %>% summarise(total= n())
  site_summary$lat <- x
  return(site_summary)
}

####
# Calculate the number of cores
no_cores <- detectCores() - 1
# Initiate cluster
cl <- makeCluster(no_cores)
clusterExport(cl, varlist=c("lat_range", "long_med", "time_sq", "calc_angs"))
list_angs <- parLapply(
  cl, lat_range, function(x){
    calc_angs(x=x, longinput=long_med, timeinput=time_sq)})
stopCluster(cl)
###

library(tidyr)
anglong <- bind_rows(list_angs)
angwide <- spread(anglong, key=eq_angle_by_10,value=total)
rm(anglong, list_angs, time_sq)
save(angwide, file="eqdata/eqnz_expected.RData")
}
load("eqdata/eqnz_expected.RData")

## -----
grand_total <- nrow(eqnz)
merged <- merge(eqnz, angwide, by.x="eq_roundedlat", by.y="lat")
agg_expected <- merged %>% select(`0`:`350`) %>% colSums(na.rm=TRUE)
expected_prop <- agg_expected / sum(agg_expected)
expected <- data.frame(eq_angle_by_10 = as.numeric(names(expected_prop)),
                       expected_prop = as.numeric(expected_prop))
expected$expected_number = expected_prop * grand_total
act_exp <- merge(expected, by_angle, by="eq_angle_by_10", all.x=TRUE)
act_exp$total[is.na(act_exp$total)] <- 0
act_exp$daynight <- NULL
act_exp$act_prop <- act_exp$total / sum(act_exp$total)

## ---- fig.cap="Comparison of expected to observed earthquakes by angle of the sun"----
old_par=par()
layout(matrix(c(1,1,1,2), ncol=4))
plot(c(0,360),c(0,7000), bty="n", type="n",
      ylab="Number of earthquakes", xlab="Angle of sun (10 degree steps)")
a <- apply(act_exp[,c(1,4)], 1, function(x){lines(c(x[1], x[1]+10), c(x[2],x[2]), lwd=2)})
a <- apply(act_exp[,c(1,3)], 1, function(x){lines(c(x[1], x[1]+10), c(x[2],x[2]), lty=3)})

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE)

```

```

lbls=c("Expected Proportion", "Observed Proportion")
typs=c(3,1)
weights=c(1,2)
legend(0,7, legend=lbls, lty=typs, lwd=weights, bty="n",
       xjust=0,
       title="Legend", y.intersp=1.2)

par(mar=old_par$mar)
par(mfrow=c(1,1))

## -----
sigmas <- c(0.682689492137086,
0.954499736103642,
0.997300203936740,
0.999936657516334,
0.999999426696856,
0.99999998026825,
0.999999999997440)

ci_brackets <- act_exp %>% ungroup() %>% mutate(grand_total=sum(total)) %>%
  rowwise() %>% mutate(
    ci_lower_7 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[7])[1,5] * grand_total,
    ci_upper_7 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[7])[1,6] * grand_total,
    ci_lower_6 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[6])[1,5] * grand_total,
    ci_upper_6 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[6])[1,6] * grand_total,
    ci_lower_5 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[5])[1,5] * grand_total,
    ci_upper_5 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[5])[1,6] * grand_total,
    ci_lower_4 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[4])[1,5] * grand_total,
    ci_upper_4 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[4])[1,6] * grand_total,
    ci_lower_3 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[3])[1,5] * grand_total,
    ci_upper_3 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[3])[1,6] * grand_total,
    ci_lower_2 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[2])[1,5] * grand_total,
    ci_upper_2 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[2])[1,6] * grand_total,
    ci_lower_1 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[1])[1,5] * grand_total,
    ci_upper_1 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[1])[1,6] * grand_total)

```

```

## ---- fig.cap="Comparison of expected to observed
## (with uncertainty) earthquakes by angle of the sun"----
bands <- rev(c('#ffffcc', '#d9f0a3', '#add8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
old_par=par()
layout(matrix(c(1,1,1,2), ncol=4))
plot(c(0,360),c(0,7000), bty="n",
      ylab="Number of earthquakes", xlab="Angle of sun (10 degree steps)")
a <- apply(ci_brackets, 1, function(x){polygon(c(x[1], x[1]+10, x[1]+10, x[1]),
                                              c(x[7],x[7], x[8],x[8]),
                                              col=bands[7], border=NA)})
a <- apply(ci_brackets, 1, function(x){polygon(c(x[1], x[1]+10, x[1]+10, x[1]),
                                              c(x[9],x[9], x[10],x[10]),
                                              col=bands[6], border=NA)})
a <- apply(ci_brackets, 1, function(x){polygon(c(x[1], x[1]+10, x[1]+10, x[1]),
                                              c(x[11],x[11], x[12],x[12]),
                                              col=bands[4], border=NA)})
a <- apply(ci_brackets, 1, function(x){polygon(c(x[1], x[1]+10, x[1]+10, x[1]),
                                              c(x[13],x[13], x[14],x[14]),
                                              col=bands[4], border=NA)})
a <- apply(ci_brackets, 1, function(x){polygon(c(x[1], x[1]+10, x[1]+10, x[1]),
                                              c(x[15],x[15], x[16],x[16]),
                                              col=bands[3], border=NA)})
a <- apply(ci_brackets, 1, function(x){polygon(c(x[1], x[1]+10, x[1]+10, x[1]),
                                              c(x[17],x[17], x[18],x[18]),
                                              col=bands[2], border=NA)})
a <- apply(ci_brackets, 1, function(x){polygon(c(x[1], x[1]+10, x[1]+10, x[1]),
                                              c(x[19],x[19], x[20],x[20]),
                                              col=bands[1], border=NA)})
a <- apply(ci_brackets, 1, function(x){lines(c(x[1], x[1]+10),
                                              c(x[4],x[4]), lwd=2)})
a <- apply(ci_brackets, 1, function(x){lines(c(x[1], x[1]+10),
                                              c(x[3],x[3]), col="white")})
a <- apply(ci_brackets, 1, function(x){lines(c(x[1], x[1]+10),
                                              c(x[3],x[3]), lty=3)})

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE)

lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
tys <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
clrs <- rev(c('#ffffcc', '#d9f0a3', '#add8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
legend(0,5, legend=lbls, lty=tys, lwd=weights, col=clrs, bty="n", xjust=0,
       title="Confidence\nIntervals:", cex=0.9)

lbls=c("Expected Proportion", "Observed Proportion")
tys=c(2,1)
weights=c(1,2)
legend(0,7, legend=lbls, lty=tys, lwd=weights, bty="n", xjust=0,

```

```

    title="Legend", y.intersp=1.2)

par(mar=old_par$mar)
par(mfrow=c(1,1))

## ---- fig.cap=
## "Difference of observed earthquake numbers from expected by 10 degree sun arc"----
#normalise on expected
norm_ci <- ci_brackets

for (i in c(4,7:20)){
norm_ci[,i] <- ci_brackets[,i]- ci_brackets[,3]
}

bands <- rev(c('#ffffcc','#d9f0a3','#addd8e','#78c679','#41ab5d','#238443','#005a32'))
old_par=par()
layout(matrix(c(1,1,1,2), ncol=4))
plot(c(0,360),c(-1500,2000), bty="n", type="n")
a <- apply(norm_ci, 1, function(x){polygon(c(x[1], x[1]+10, x[1]+10, x[1]),
                                             c(x[7],x[7], x[8],x[8]),
                                             col=bands[7], border=NA)})
a <- apply(norm_ci, 1, function(x){polygon(c(x[1], x[1]+10, x[1]+10, x[1]),
                                             c(x[9],x[9], x[10],x[10]),
                                             col=bands[6], border=NA)})
a <- apply(norm_ci, 1, function(x){polygon(c(x[1], x[1]+10, x[1]+10, x[1]),
                                             c(x[11],x[11], x[12],x[12]),
                                             col=bands[4], border=NA)})
a <- apply(norm_ci, 1, function(x){polygon(c(x[1], x[1]+10, x[1]+10, x[1]),
                                             c(x[13],x[13], x[14],x[14]),
                                             col=bands[4], border=NA)})
a <- apply(norm_ci, 1, function(x){polygon(c(x[1], x[1]+10, x[1]+10, x[1]),
                                             c(x[15],x[15], x[16],x[16]),
                                             col=bands[3], border=NA)})
a <- apply(norm_ci, 1, function(x){polygon(c(x[1], x[1]+10, x[1]+10, x[1]),
                                             c(x[17],x[17], x[18],x[18]),
                                             col=bands[2], border=NA)})
a <- apply(norm_ci, 1, function(x){polygon(c(x[1], x[1]+10, x[1]+10, x[1]),
                                             c(x[19],x[19], x[20],x[20]),
                                             col=bands[1], border=NA)})
a <- apply(norm_ci, 1, function(x){lines(c(x[1], x[1]+10), c(x[4],x[4]), lwd=2)})
a <- apply(norm_ci, 1, function(x){lines(c(x[1], x[1]+10), c(0,0), col="white")})
a <- apply(norm_ci, 1, function(x){lines(c(x[1], x[1]+10), c(0,0), lty=3)})

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE)
lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")

```

```

typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
clrs <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
legend(0,5, legend=lbls, lty=typs, lwd=weights, col=clrs, bty="n", xjust=0,
       title="Confidence\nIntervals:", cex=0.9)

lbls=c("Expected Proportion", "Observed Proportion")
typs=c(2,1)
weights=c(1,2)
legend(0,7, legend=lbls, lty=typs, lwd=weights, bty="n", xjust=0,
       title="Legend", y.intersp=1.2)

par(mar=old_par$mar)
par(mfrow=c(1,1))

## ---- "Oversupply of earthquakes compared to expected number in suns orbit"----
norm_ci$border = 2
# need to double entries with a displacement of 10 to make each side of the item on the
# graph
norm_ci2 <- norm_ci
norm_ci2$eq_angle_by_10 <- norm_ci2$eq_angle_by_10 + 10
norm_ci2$border = 1
graphdata <- bind_rows(norm_ci,norm_ci2) %>% arrange(eq_angle_by_10,border)

##### plot graph
bands <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
old_par=par()
layout(matrix(c(1,1,1,2), ncol=4))

# overall limits
limits=2 * max(abs(c(graphdata$ci_lower_7, graphdata$ci_upper_7)))
# plot upper confidence 7 interval using plotrix
polar.plot(graphdata$ci_upper_7, polar.pos=graphdata$eq_angle_by_10,
            radial.lim=c(-1*limits,limits),
            labels = "", main=NULL,lwd=0.5, rp.type="p",
            show.grid.labels=FALSE, show.grid=FALSE, mar=c(0,0,0,0),
            grid.col=bands[7], line.col=bands[7], poly.col=bands[7])
# plot upper 6 confidence interval
plot_ci_round <- function(upper_bound,x){
  polar.plot(upper_bound, polar.pos=graphdata$eq_angle_by_10, add=TRUE,
             radial.lim=c(-1*limits,limits),
             line.col=bands[x], lwd=0.5, rp.type="p", poly.col=bands[x])
}
plot_ci_round(graphdata$ci_upper_6, 6)
plot_ci_round(graphdata$ci_upper_5, 5)
plot_ci_round(graphdata$ci_upper_4, 4)
plot_ci_round(graphdata$ci_upper_3, 3)
plot_ci_round(graphdata$ci_upper_2, 2)
plot_ci_round(graphdata$ci_upper_1, 1)

```

```

plot_ci_round(graphdata$ci_lower_1, 2)
plot_ci_round(graphdata$ci_lower_2, 3)
plot_ci_round(graphdata$ci_lower_3, 4)
plot_ci_round(graphdata$ci_lower_4, 5)
plot_ci_round(graphdata$ci_lower_5, 6)
plot_ci_round(graphdata$ci_lower_6, 7)
polar.plot(graphdata$ci_lower_7, polar.pos=graphdata$eq_angle_by_10,
            add=TRUE, radial.lim=c(-1*limits,limits),
            line.col="white", lwd=0.5, rp.type="p", poly.col="white")

# plot expected guide line
polar.plot(rep(0,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10,
            add=TRUE,radial.lim=c(-1*limits,limits),
            rp.type="p", lty=4)
# plot 500 less than expected guide line
polar.plot(rep(-500,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10,
            add=TRUE,radial.lim=c(-1*limits,limits),
            rp.type="p", lty=1, line.col="#00000044")
# plot 500 more than expected guide line
polar.plot(rep(500,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10,
            add=TRUE,radial.lim=c(-1*limits,limits),
            rp.type="p", lty=3, line.col="#00000044")

lines(c(-1.5,-1.2)*limits, c(0,0))
lines(c(1.5,1.2)*limits, c(0,0))
text(-1.8*limits,0, label="sunset\n180", cex=0.7)
text(1.8*limits,0, label="sunrise\n0", cex=0.7)

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE, xlab="")
lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
clrs <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
legend(0,4.5, legend=lbls, lty=typs, lwd=weights, col=clrs, bty="n", xjust=0,
       title="Confidence Intervals:", cex=0.9)

lbls2=c("Expected Number", paste(circlesize,"under expected"),
       paste(circlesize,"over expected"))
typs2=c(4,1,3)
weights2=c(1,1,1)
clrs2=c("#000000", "#00000044", "#00000044")
legend(0,10, legend=lbls2, lty=typs2, lwd=weights2, bty="n", xjust=0,
       title="Legend", y.intersp=1.2, col=clrs2)

par(mar=old_par$mar)

```

```

par(mfrow=c(1,1))

## -----
act_exp$diurnal <- "day"
act_exp$diurnal[act_exp$eq_angle_by_10 >= 180] <- "night"
ggplot(act_exp, aes(x=expected_number, y=total, col=diurnal)) +
  geom_point() + scale_colour_manual(values=c("#56B4E9", "#000000"))

## -----
act_exp$is_night <- act_exp$eq_angle_by_10 >= 180
lin_data_day <- act_exp[!act_exp$is_night,]
lin_data_night <- act_exp[act_exp$is_night,]
lm_day <- lm(total ~ expected_number + 0, data=lin_data_day)
lm_night <- lm(total ~ expected_number + 0, data=lin_data_night)
ggplot(act_exp, aes(x=expected_number, y=total, col=diurnal)) +
  geom_point() + scale_colour_manual(values=c("#56B4E9", "#000000")) +
  stat_smooth(method="lm", formula= y ~ x + 0)
act_exp$res <- NA
act_exp$res[1:18] <- lm_day$residuals
act_exp$res[19:36] <- lm_night$residuals
act_exp$baseline <- 0
act_exp$baseline[19:36] <- mean(act_exp$total[19:36]) - mean(act_exp$total[1:18])
act_exp$adjust_res <- act_exp$res + act_exp$baseline

## ---- fig.cap="Variation in raw residuals"-----
plot(act_exp$res, type="n", bty="n")
points(1:18, act_exp$res[1:18], pch=19, cex=0.5, col="#56B4E9")
points(19:36, act_exp$res[19:36], pch=19, cex=0.5, col="black")
lines(1:18, act_exp$res[1:18], pch=19, col="#56B4E9")
lines(19:36, act_exp$res[19:36], pch=19, col="black")

## ---- fig.cap="Rotational variation in residuals.
## The nightly residuals have been displaced by the mean difference between day
## and night, to reflect the different between the two lines"-----
old_par=par()
layout(matrix(c(1,1,1,2), ncol=4))

act_exp$border = 1
# need to double entries with a displacement of 10
# to make each side of the item on the graph
limits=2 * max(abs(act_exp$adjust_res))
#white background
polar.plot(c(act_exp$adjust_res[1], act_exp$adjust_res[1]),
           polar.pos=c(graphdata$eq_angle_by_10[1],graphdata$eq_angle_by_10[1]+10),
           radial.lim=c(-1*limits,limits), labels = "", main=NULL,lwd=4, rp.type="p",
           show.grid.labels=FALSE, show.grid=FALSE,
           grid.col="white", line.col="blue", poly.col="white")

a <- apply(act_exp[1:18,c("adjust_res","eq_angle_by_10")],1, function(x){
  polar.plot(c(x[1], x[1]), polar.pos= c(x[2], x[2]+10),

```

```

    add=TRUE,radial.lim=c(-1*limits,limits), rp.type="p", lwd=4, line.col="#56B4E9")
})

a <- apply(act_exp[19:36,c("adjust_res","eq_angle_by_10")],1, function(x){
  polar.plot(c(x[1], x[1]), polar.pos= c(x[2], x[2]+10),
  add=TRUE,radial.lim=c(-1*limits,limits), rp.type="p", lwd=4, line.col="black")
})

a <- apply(act_exp[1:18,c("adjust_res","eq_angle_by_10")],1, function(x){
  polar.plot(c(0, 0), polar.pos= c(x[2], x[2]+10),
  add=TRUE,radial.lim=c(-1*limits,limits), rp.type="p", lwd=1, line.col="white")
})

a <- apply(act_exp[1:18,c("adjust_res","eq_angle_by_10")],1, function(x){
  polar.plot(c(0, 0), polar.pos= c(x[2], x[2]+10),
  add=TRUE,radial.lim=c(-1*limits,limits), rp.type="p", lwd=1, lty=3, line.col="#56B4E9")
})

a <- apply(act_exp[19:36,c("adjust_res","eq_angle_by_10")],1, function(x){
  polar.plot(c(678.5, 678.5), polar.pos= c(x[2], x[2]+10),
  add=TRUE,radial.lim=c(-1*limits,limits), rp.type="p", lwd=1, line.col="white")
})

a <- apply(act_exp[19:36,c("adjust_res","eq_angle_by_10")],1, function(x){
  polar.plot(c(678.5, 678.5), polar.pos= c(x[2], x[2]+10),
  add=TRUE,radial.lim=c(-1*limits,limits), rp.type="p", lwd=1, lty=3, line.col="black")
})

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE, xlab="")

```

lbls2=c("Residuals Day", "Residuals Night", "Expected Day", "Expected Night")  
 typs2=c(1,1,3,3)  
 weights2=c(4,4,1,1)  
 clrs2=c("#56B4E9","#000000","#56B4E9", "#000000")  
 legend(0,10, legend=lbls2, lty=typs2, lwd=weights2, bty="n", xjust=0,  
 title="Legend", y.intersp=1.2, col=clrs2)

```

par(mar=old_par$mar)
par(mfrow=c(1,1))

## -----
set_diurn <- eqnz %>%
  mutate(solidearth = ifelse(eq_solidearth_vertical < 0, "contracted", "expanded"),
         diurnal = ifelse(eq_is_night,"night","day")) %>%
  group_by(solidearth, diurnal) %>% summarise(total = n())

knitr::kable(set_diurn,
            caption="Number of earthquakes by day/night cycle and condition of crust")

## -----

```

```

ci_solearth <- set_diurn %>% ungroup() %>% spread(diurnal, total) %>% rowwise() %>%
  mutate(ci_lower_7 = binom.confint(night, (day + night),
                                    method=c("wilson"), conf.level = sigmas[7])[1,5],
         ci_upper_7 = binom.confint(night, (day + night),
                                    method=c("wilson"), conf.level = sigmas[7])[1,6],
         ci_lower_6 = binom.confint(night, (day + night),
                                    method=c("wilson"), conf.level = sigmas[6])[1,5],
         ci_upper_6 = binom.confint(night, (day + night),
                                    method=c("wilson"), conf.level = sigmas[6])[1,6],
         ci_lower_5 = binom.confint(night, (day + night),
                                    method=c("wilson"), conf.level = sigmas[5])[1,5],
         ci_upper_5 = binom.confint(night, (day + night),
                                    method=c("wilson"), conf.level = sigmas[5])[1,6],
         ci_lower_4 = binom.confint(night, (day + night),
                                    method=c("wilson"), conf.level = sigmas[4])[1,5],
         ci_upper_4 = binom.confint(night, (day + night),
                                    method=c("wilson"), conf.level = sigmas[4])[1,6],
         ci_lower_3 = binom.confint(night, (day + night),
                                    method=c("wilson"), conf.level = sigmas[3])[1,5],
         ci_upper_3 = binom.confint(night, (day + night),
                                    method=c("wilson"), conf.level = sigmas[3])[1,6],
         ci_lower_2 = binom.confint(night, (day + night),
                                    method=c("wilson"), conf.level = sigmas[2])[1,5],
         ci_upper_2 = binom.confint(night, (day + night),
                                    method=c("wilson"), conf.level = sigmas[2])[1,6],
         ci_lower_1 = binom.confint(night, (day + night),
                                    method=c("wilson"), conf.level = sigmas[1])[1,5],
         ci_upper_1 = binom.confint(night, (day + night),
                                    method=c("wilson"), conf.level = sigmas[1])[1,6])
ci_solearth$yheight <- c(1,3)
layout(matrix(c(1,1,1,2), ncol=4))

par(mar=c(5,6,4,2))
plot(c(0.49,0.61), y=c(-3,8), type="n", bty="n", yaxt="n", ylab="State of earth's crust",
      xlab="Proportion of earthquakes at night")
a <- apply(ci_solearth[,2:18], 1, function(x){
  polygon(c(x[3], x[4], x[4], x[3]), c(x[17]-1, x[17]-1, x[17], x[17]),
          col=bands[7], border=NA)})
a <- apply(ci_solearth[,2:18], 1, function(x){
  polygon(c(x[5], x[6], x[6], x[5]), c(x[17]-1, x[17]-1, x[17], x[17]),
          col=bands[6], border=NA)})
a <- apply(ci_solearth[,2:18], 1, function(x){
  polygon(c(x[7], x[8], x[8], x[7]), c(x[17]-1, x[17]-1, x[17], x[17]),
          col=bands[5], border=NA)})
a <- apply(ci_solearth[,2:18], 1, function(x){
  polygon(c(x[9], x[10], x[10], x[9]), c(x[17]-1, x[17]-1, x[17], x[17]),
          col=bands[4], border=NA)})
a <- apply(ci_solearth[,2:18], 1, function(x){
  polygon(c(x[11], x[12], x[12], x[11]), c(x[17]-1, x[17]-1, x[17], x[17]),
          col=bands[3], border=NA)})
a <- apply(ci_solearth[,2:18], 1, function(x){
  polygon(c(x[13], x[14], x[14], x[13]), c(x[17]-1, x[17]-1, x[17], x[17]),
          col=bands[2], border=NA)})

```

```

a <- apply(ci_solearth[,2:18], 1, function(x){
  polygon(c(x[15], x[16], x[16], x[15]), c(x[17]-1, x[17]-1, x[17], x[17]),
    col=bands[1], border=NA)})
a <- apply(ci_solearth[,2:18], 1, function(x){
  lines(c(x[2]/(x[1] + x[2]), x[2]/(x[1] + x[2])), c(x[17]-1, x[17]), lwd=2)})

axis(2, at=c(0.5,2.5), labels=c("contracted", "expanded"),
  tick=FALSE, lwd=0, las=2, pos=0.5)

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE)
legend(0,5.5, legend=lbls, lty=typs, lwd=weights, col=clrs, bty="n", xjust=0,
  title="Confidence Intervals:", y.intersp=1.1, cex=0.9)

lbls2=c("Observed Proportion")
tys2=1
weights2=2
legend(0,7, legend=lbls2, lty=tys2, lwd=weights2, bty="n", xjust=0,
  title="Legend", y.intersp=1.2, cex=0.9)

par(mar=old_par$mar)
par(mfrow=c(1,1))

```

# Chapter 8

## It is not uniform

The question “Is this a general effect” is closely related to the question “Why has no-one noticed this before, and been convinced enough to push the matter”. If it is a general effect, then you should observe a similar response in all subareas of a region. If the effect is regional, then local geological conditions are going to condition the response.

Using the gridded regions and with either of these assumptions being true:

- if the the region is not sensitive to a night/day effect
- or the sensitivity is confined only to that region

then a high proportion of night earthquakes in one region should not influence the number of night earthquakes in neighbouring regions. If the number of night earthquakes is both sensitive to local geological conditions, and those conditions extend beyond the 50 kilometre grid areas, then similar levels of night proportion earthquakes should be found in adjacent grids. This is a test of both the effect itself not being the result of chance, and the distribution of the effects results not being the result of chance

For the purposes of this question, I can rank the proportion of night earthquakes into ten steps (deciles) from highest to lowest. I define similar as a grid square being within one decile step of its neighbour, and I define neighbour as being the grid square to the north or east of the grid square in question. This is to stop double counting with the square to the south of a northern square being the same neighbour as the square to the north of the southern square.

	neighbour	
	region	neighbour

I also limit the grids used to those with more than 80 total earthquake events, so that the sample is large enough the proportion is likely to be a true reflection of the region. In reality, the threshold of 80 is a somewhat arbitrary one that balances having a good number of earthquakes with a good number of grid areas.

The current state is:



Only the very lowest decile contains any grid areas with less than 50% of the earthquakes at night. By eye there seems to be a lot of clustering, but the trick is to measure how unlikely that is.

Using the idea of similar neighbours being one decile apart or the same, 83 (45.86%) of the 181 neighbour pairs are similar.

To determine how likely this is to have occurred, I made a test by simulation. Taking the idea that there are 119 decile values that form the 181 pairs, I repeatedly randomly shuffle the locations of the deciles. This is done on the assumption of independence- that if the areas are unrelated and it is a random process, then the deciles are equally likely to be in any of the locations. Then for each shuffle result I count up how many neighbour pairs are similar. By comparing the 83 observed values with how often that value or more comes up in the simulation, I find out how unlikely the result would be to occur by chance.

As a code writing efficiency note, to be able to do a lot of simulations (100,000,000) quickly (less than 15 minutes), I work out the specific combinations of neighbours ahead of time and provide that, as the combinations are not going to change between simulations.

In the 100,000,000 replications of the simulation, the simulation produced a result as high as the observed one 13 times. As considering it as a two-sided test, where an extreme result in either direction from the mean is unusual (in this case a freakishly low score would also indicate unnatural arrangement of deciles), the result is 13 in 100,000,000 (as there were 0 that low). Without reading too much into the finer details of this, it does suggest that the observed result is very, extremely unlikely to occur by chance, being outside of 5 sigma.

## 8.1 Formal Statement

As the geographic clustering of night earthquake is not due to chance, we must conclude that not only is there an effect, that effect is influenced by local geological conditions. Those earthquakes influenced by the sun's position should thus be seen as a local geological response to solar gravity conditions.

## 8.2 Chapter Code

```

## ----setup, include=FALSE-----
knitr::opts_chunk$set(echo = FALSE)
knitr::opts_chunk$set(warnings=FALSE)
knitr::opts_chunk$set(errors=FALSE)
knitr::opts_chunk$set(message=FALSE)
knitr::opts_chunk$set(dpi = 150)
knitr::opts_chunk$set(fig.width =  6)
knitr::opts_chunk$set(fig.height =  4)

## -----
Sys.setenv(TZ = "UTC")
library(dplyr)
library(ggplot2)
library(lubridate)
library(maptools)
library(binom)
library(parallel)
library(plotrix)
library(solidearthtide)
library(tidyr)
library(maps)
library(mapdata)
# Assumes there is eqnz_processed data created in chapter 2
load("eqdata/eqnz_processed.RData")
old_par <- par()

lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
clrs <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))

## ---- fig.height=4, fig.width=4-----
plot(x=c(0,3), y=c(0,3), axes=FALSE, xlab="", ylab="", type="n")
abline(h=1)
abline(h=2)
abline(v=1)
abline(v=2)
text(1.5,1.5,"region", cex=0.8)
text(2.5,1.5,"neighbour", cex=0.8)
text(1.5,2.5,"neighbour", cex=0.8)

## -----
reg_eq <- eqnz %>% group_by(eq_roundedlat, eq_roundedlong,
  eq_gridpoint_y, eq_gridpoint_x) %>%
  summarise(region_tot = n(), night_prop = sum(eq_is_night) / region_tot) %>%
  filter(region_tot > 80) %>% ungroup() %>%
  mutate(decile = ntile(night_prop, 10))

```

```

ten_colours <- rev(c('#a50026', '#d73027', '#f46d43', '#fdbe61', '#fee090',
                      '#e0f3f8', '#abd9e9', '#74add1', '#4575b4', '#313695'))
with_alpha <- paste(ten_colours, "CC", sep="")
grid_min <- reg_eq %>% group_by(decile) %>%
  summarise(from= min(night_prop))

## -----
layout(matrix(c(1,1,1,2), ncol=4))
map("nzHires", ylim=c(-48.5, -33.5))
points(reg_eq$eq_roundedlong, reg_eq$eq_roundedlat, pch=15, col=with_alpha[reg_eq$decile],
       cex=1.5)

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE)
legend(0,9, legend=paste(1:10, ":", round(grid_min$from,2), "+", sep=""),
       fill=with_alpha, bty="n", xjust=0,
       title="Deciles:", cex=0.9, border=with_alpha)

par(mar=old_par$mar)
par(mfrow=c(1,1))

## -----
xgrids <- min(eqnz$eq_gridpoint_x):max(eqnz$eq_gridpoint_x)
ygrids <- min(eqnz$eq_gridpoint_y):max(eqnz$eq_gridpoint_y)
squares <- data.frame(xg = rep(xgrids, times=length(ygrids)),
                       yg = rep(ygrids, each=length(xgrids)))
squares$east_x <- squares$xg + 1
squares$north_y <- squares$yg + 1
gridx <- c(squares$xg, squares$xg)
gridy <- c(squares$yg, squares$yg)
neighbourx <- c(squares$east_x, squares$xg)
neighboury <- c(squares$yg, squares$north_y)
potential_neighbours <- data.frame(gridx, gridy, neighbourx, neighboury)
reg_eq$grid_index <- 1:nrow(reg_eq)
actual_grid <- merge(potential_neighbours,
                      reg_eq[,c("eq_gridpoint_x", "eq_gridpoint_y",
                               "grid_index", "decile")],
                      by.x= c("gridx", "gridy"),
                      by.y= c("eq_gridpoint_x", "eq_gridpoint_y"))
actual_neighbours <- merge(actual_grid,
                            reg_eq[,c("eq_gridpoint_x", "eq_gridpoint_y",
                                      "grid_index", "decile")],
                            by.x= c("neighbourx", "neighboury"),
                            by.y= c("eq_gridpoint_x", "eq_gridpoint_y"))
names(actual_neighbours) <- c("neighbourx", "neighboury", "gridx",
                               "gridy", "grid_index", "decile_grid",
                               "neighbour_index", "decile_neighbour")

```

```

grids <- actual_neighbours$grid_index
neighbours <- actual_neighbours$neighbour_index
deciles <- reg_eq$decile

similar <- sum(abs(
  actual_neighbours$decile_grid - actual_neighbours$decile_neighbour) < 2)

## -----
if(!file.exists("eqdata/simResult.RData")){
  repli <- function(decs, grids, neighbours){
    decile <- sample(decs, size=length(decs), replace=FALSE)
    grid_dec <- decile[grids]
    neighbour_dec <- decile[neighbours]
    sum(abs(grid_dec - neighbour_dec) < 2)
  }

  replication_size <- 100000000
  # can comfortably get up to 100000000 in one go (15 minutes) on 8 GB machine
  # if run in parallel on 7 cores
  # but another order of magnitude crashes things

  # Calculate the number of cores
  no_cores <- detectCores() - 1

  # Initiate cluster
  cl <- makeCluster(no_cores)
  # make R objects available to cluster
  clusterExport(cl, varlist=c("repli", "deciles", "grids", "neighbours"))
  # do function in parallel
  similar_rep <- parSapply(cl, 1:replication_size,
                           function(x){repli(deciles, grids, neighbours)})
  stopCluster(cl)
  save(similar_rep, file="eqdata/simResult.RData")
}

load("eqdata/simResult.RData")

sim_result <- sum(similar_rep >= similar)
sim_mean <- mean(similar_rep)
low_end <- sim_mean - (similar - sim_mean)
sim_extreme <- sum(similar_rep >= similar | similar_rep <= low_end)

```



# Chapter 9

## Japanese earthquakes are the same

One concern expressed to me is that even if it is a genuine effect, it might just be a New Zealand quirk, rather than a global effect.

A replication in science is when you conduct the same analysis with different data, and see if you reach a similar conclusion. Providing I have earthquake time, location, and magnitude it is easy to repeat the analysis as I just rerun the computer code, and there are many countries that have independently gathered their own earthquake data on their own seismographs. The main way I chose countries was that I could easily find the earthquake collections with an English language web search.

Japan is the source of the best, most detailed, earthquake data. To access the earthquake catalogue, you must register([Link 1](#)), and can then download earthquake information. If you do download 5 years of earthquake data by requesting seven days of earthquakes and saving the web page result, the code is an example of extracting and combining the earthquake data to form a five year set for all Japan.

For the same period as the New Zealand Data (from September 2011 to September 2016) Earthquakes data from the Japan Meteorological Agency, there are 719348 events of depth greater than 0 and magnitude greater than 0.

Of the 719348 in the data, 385156 occurred at night, a proportion of 0.5354. A seven sigma confidence interval for the proportion of earthquakes occurring at night would be 0.5313 to 0.5395. This confidence interval in no way coincides with 0.5, and using one so large we can confidently say that if earthquakes occur randomly, this result would never occur.

While the proportion is not as high as New Zealand's, the greater number of earthquakes means that the confidence intervals are smaller, so there is more certainty that the rate of night earthquakes is not 50%

Table 9.1: Data description

feature	value
Earliest (UTC)	2011-09-01 00:02:10
Latest (UTC)	2016-08-31 23:58:56
Northernmost	54.007
Southernmost	18.335
Westmost	120.1
Eastmost	158.506
Percent < Mag 3	94.47
total entries	719348
nighttime quakes	385156

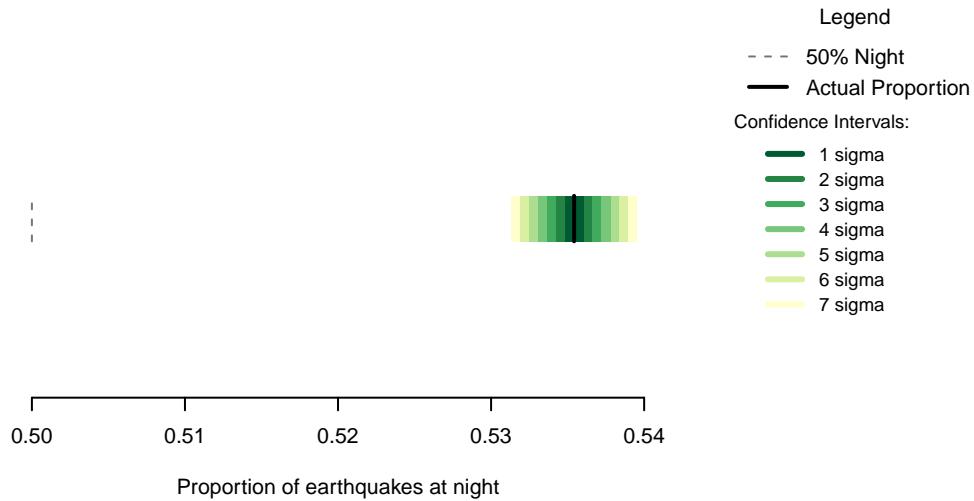


Figure 9.1: Proportion of earthquakes at night: Japan. n=719348

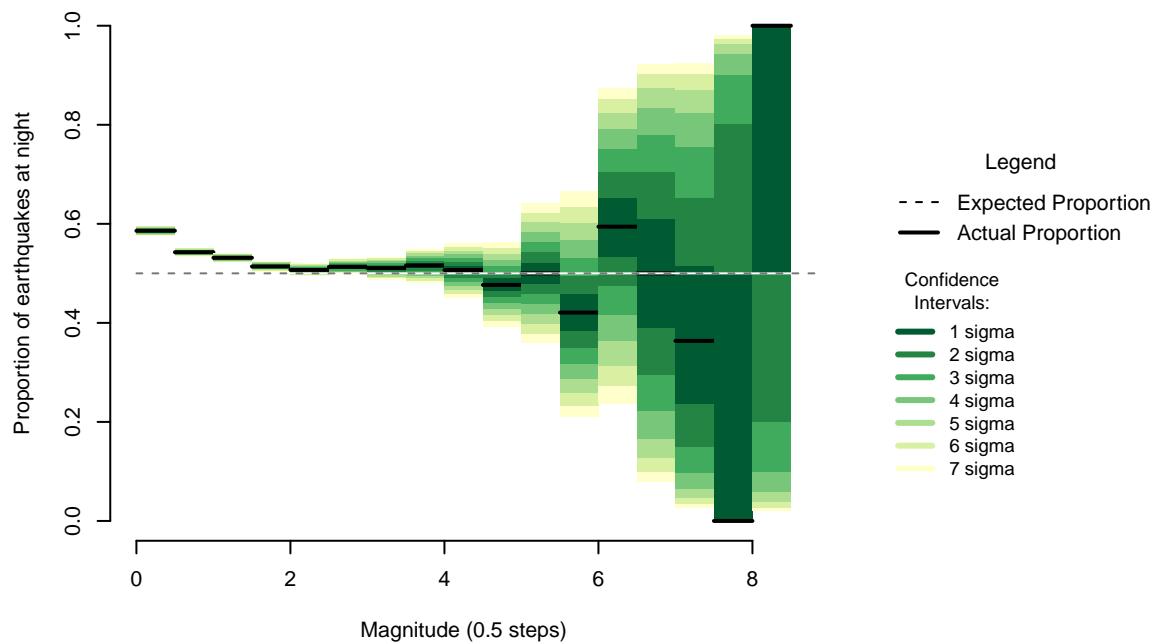


Figure 9.2: Proportion of night earthquakes by magnitude, Japan. n=719348

Examining magnitude, there is a similar pattern to New Zealand. Unequivocally high numbers of earthquakes in the low magnitudes falling towards 50% as magnitude increases, then becoming unclear as sample size decreases.

Taking the expected, and calculating the angle by 10s, and the day night slope.

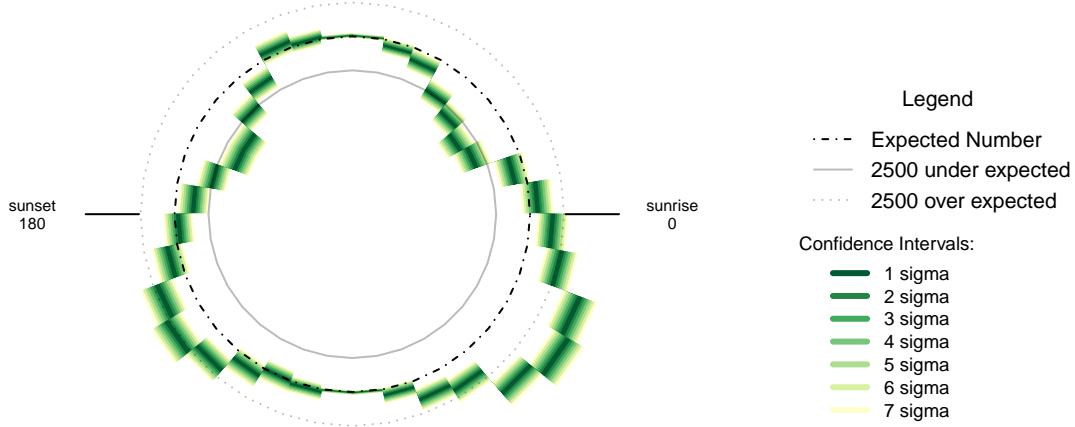


Figure 9.3: Over- and under-supply of earthquakes by angle of the sun (10 degree steps). Japan. n=719348

The trend for the number of earthquakes by 10 degree arc of the sun is similar to New Zealand- undersupplies at 30 degrees above the horizon and oversupplies 30-40 degrees below the horizon. The Japanese data is skewed slightly compared to New Zealand, as the oversupply is clearly at its greatest when the sun is 30 to 40 degrees below the horizon to the east.

## 9.1 Formal Statement

Earthquakes in Japan show the same pattern as New Zealand, displaying an oversupply of earthquakes at night that is not the result of chance. The magnitude pattern of the oversupply is similar to New Zealand's pattern, and the pattern with respect to the position of the sun is similar to that of New Zealand.

## 9.2 Links

1 - JMA Unified Hypocentre Catalogs <https://hinetwww11.bosai.go.jp/auth/?LANG=>

The earthquake catalog used in this study is produced by the Japan Meteorological Agency, in cooperation with the Ministry of Education, Culture, Sports, Science and Technology. The catalog is based on seismic data provided by the National Research Institute for Earth Science and Disaster Resilience, the Japan Meteorological Agency, Hokkaido University, Hirosaki University, Tohoku University, the University of Tokyo, Nagoya University, Kyoto University, Kochi University, Kyushu University, Kagoshima University, the National Institute of Advanced Industrial Science and Technology, the Geographical Survey Institute, Tokyo Metropolis, Shizuoka Prefecture, Hot Springs Research Institute of Kanagawa Prefecture, Yokohama City, and Japan Agency for Marine-Earth Science and Technology.

## 9.3 Chapter Code

```

## ----setup, include=FALSE-----
knitr::opts_chunk$set(echo = FALSE)

## ---- warnings=FALSE, errors=FALSE, message=FALSE-----
library(geosphere)
library(lubridate, quietly=TRUE)
library(dplyr)
library(binom)
library(ggplot2)
library(maps)
library(mapdata)
library(parallel)
library(readr)
library(plotrix)
library(tidyr)
library(mapproj)
Sys.setenv(TZ = "UTC")

## ----warnings=FALSE, errors=FALSE, message=FALSE-----
# Note:
# Data is from the Japan Meteorological Agency Unified Hypocenter Catalogs
# https://hinetwww11.bosai.go.jp/auth/?LANG=
# As this service requires registration to access the data, this script proceeds from the
# point that a registered user
# has used the web catalogue to save web pages of earthquake results from the catalog,
# with each saved file having the suffix .html
# the files are stored in a folder called JMA_web_results
# the files are fixed width entries embedded within html

if(!dir.exists("../othereqdata")){
  dir.create("../othereqdata")
}
if(!file.exists("../othereqdata/eq_japan_raw.RData")){

extract_eq <- function(x){
  wbtbl <- read_fwf(
    file=paste("../japan/JMA_web_results/",x, sep=""),
    col_positions=fwf_positions(c(1,32,46,61,74, 84),
                                c(19,37,52,66,78, NA),
                                col_names = c("date_raw", "latitude", "longitude",
                                             "depth", "magnitude", "description")),
    skip=304)
  #this does generate a lot of errors, but that is OK, it is having
  #trouble with the stuff we don't actually want
  return(wbtbl)
}

j_e <- lapply(list.files(path="../japan/JMA_web_results/", pattern="*.html"),extract_eq)
eqjp <- bind_rows(j_e)
rm(j_e)
eqjp$time_UTC <- ymd_hms(eqjp$date_raw, tz="UTC") - hours(9)

```

```
#time in Japanese standard so convert to UTC

eqjp <- eqjp[complete.cases(eqjp),]
eqjp$latitude <- as.numeric(eqjp$latitude)
eqjp$longitude <- as.numeric(eqjp$longitude)
eqjp$depth <- as.numeric(eqjp$depth)
eqjp$magnitude <- as.numeric(gsub("[A-Z]+", "", eqjp$magnitude, ignore.case = TRUE))
eq_national <- eqjp %>%
  filter(depth > 0 & magnitude >= 0 &
         time_UTC >=
           as.POSIXct("2011-09-01T00:00:00", format="%Y-%m-%dT%H:%M:%S", tz="UTC") &
         time_UTC <
           as.POSIXct("2016-09-01T00:00:00", format="%Y-%m-%dT%H:%M:%S", tz="UTC")) %>%
  distinct() %>% arrange(time_UTC)

save(eq_national, file="../othereqdata/eq_japan_raw.RData")
}
rm(list=ls())

## ----

if(!file.exists("../othereqdata/eq_japan_processed.RData")){
  load("../othereqdata/eq_japan_raw.RData")
  southmost <- min(eq_national$latitude)
  westmost <- min(eq_national$longitude)

  eq_national <- eq_national %>% filter(
    magnitude > 0, depth > 0) %>% rowwise() %>% mutate(
    eq_gridpoint_y = round(
      distVincentyEllipsoid(c(longitude, southmost), c(longitude, latitude)) /50000,0),
    eq_gridpoint_x = round(
      distVincentyEllipsoid(c(westmost, latitude), c(longitude, latitude)) /50000,0),
    eq_roundedlat = destPoint(
      p=c(longitude, southmost), b=0, d=eq_gridpoint_y*50000)[2],
    eq_roundedlong = destPoint(
      p=c(westmost, eq_roundedlat), b=90, d=eq_gridpoint_x*50000)[1]) %>% ungroup()
  # use maptools to calculate solar angles
  sun_angles <- solarpes(
    matrix(c(eq_national$longitude, eq_national$latitude), ncol=2), eq_national$time_UTC)
  colnames(sun_angles) <- c("eq_compass", "eq_vertical")
  eq_national <- cbind(eq_national, sun_angles)
  eq_national$eq_is_night <- eq_national$eq_vertical < 0
  # calculate 360 degree as well as vertical
  eq_national <- eq_national %>%
    mutate(eq_angle_360 = eq_vertical,
          eq_angle_360 = ifelse(eq_compass > 180, 180 - eq_angle_360, eq_angle_360),
          eq_angle_360 = ifelse(eq_vertical < 0 & eq_compass <= 180,
                                360 + eq_angle_360, eq_angle_360),
          eq_angle_by_10 = floor(eq_angle_360 /10) * 10)
  save(eq_national, file="../othereqdata/eq_japan_processed.RData")
}
rm(list=ls())
```

```

## -----
if(!file.exists("../othereqdata/eq_japan_expected.RData")){
load("../othereqdata/eq_japan_processed.RData")

lat_range <- unique(eq_national$eq_roundedlat)
long_med <- median(eq_national$eq_roundedlong)

# 1 minute intervals for a full solar year
time1 <- ymd_hms("2015-01-01 00:00:00")
time2 <- ymd_hms("2015-12-31 23:59:00")
time_sq <- seq.POSIXt(from=time1, to=time2, by="min")

calc_angs <- function(x, longinput, timeinput){
  library(dplyr)
  sun_angles <- maptools::solarpos(matrix(c(longinput, x), ncol=2), timeinput)
  colnames(sun_angles) <- c("eq_compass", "eq_vertical")
  # calculate 360 degree as well as vertical
  site_summary <- as.data.frame(sun_angles) %>%
    mutate(eq_angle_360 = eq_vertical,
          eq_angle_360 = ifelse(eq_compass > 180, 180 - eq_angle_360, eq_angle_360),
          eq_angle_360 = ifelse(eq_vertical < 0 & eq_compass <= 180,
                                360 + eq_angle_360, eq_angle_360),
          eq_angle_by_10 = floor(eq_angle_360 /10) * 10) %>%
    group_by(eq_angle_by_10) %>% summarise(total= n())
  site_summary$lat <- x
  return(site_summary)
}
####
# Calculate the number of cores
no_cores <- detectCores() - 1
# Initiate cluster
cl <- makeCluster(no_cores)
clusterExport(cl, varlist=c("lat_range", "long_med", "time_sq", "calc_angs"))
list_angs <- parLapply(cl, lat_range, function(x){
  calc_angs(x=x, longinput=long_med, timeinput=time_sq)})
stopCluster(cl)
####
library(tidyr)
anglong <- bind_rows(list_angs)
angwide <- spread(anglong, key=eq_angle_by_10,value=total)
rm(anglong, list_angs, time_sq)
save(angwide, file="../othereqdata/eq_japan_expected.RData")
}
rm(list=ls())

## -----
load("../othereqdata/eq_japan_processed.RData")
load("../othereqdata/eq_japan_expected.RData")
eq_night = sum(eq_national$eq_is_night)
eq_total = nrow(eq_national)
bands <- rev(c('#ffffcc', '#d9f0a3', '#add8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
sigmas <- c(0.682689492137086,

```

```

0.954499736103642,
0.997300203936740,
0.999936657516334,
0.999999426696856,
0.99999998026825,
0.99999999997440)
lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
old_par=par()

## -----
bt <- binom.test(eq_night ,eq_total, conf.level= .99999999997440)

## -----
feature <- c("Earliest (UTC)", "Latest (UTC)",
           "Northernmost", "Southernmost",
           "Westmost", "Eastmost",
           "Percent < Mag 3", "total entries",
           "nighttime quakes")
value <- c(as.character(min(eq_national$time_UTC)),
           as.character(max(eq_national$time_UTC)),
           as.character(max(eq_national$latitude)),
           as.character(min(eq_national$latitude)),
           as.character(min(eq_national$longitude)),
           as.character(max(eq_national$longitude)),
           as.character(round(100*sum(eq_national$magnitude < 3)/eq_total,2)),
           as.character(eq_total),
           as.character(eq_night))
data.frame(feature,value) %>% knitr::kable(caption="Data description")

## ---- fig.cap="Proportion of earthquakes at night: Japan. n=719348"-----
### making the basic proportion graph
eq_night = sum(eq_national$eq_is_night)
eq_total = nrow(eq_national)
bands <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
sigmas <- c(0.682689492137086,
           0.954499736103642,
           0.997300203936740,
           0.999936657516334,
           0.999999426696856,
           0.99999998026825,
           0.99999999997440)
lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
typs <- c(1,1,1,1,1,1,1)

```

```

weights <- c(3,3,3,3,3,3,3)
old_par=par()
conf_steps <- function(x, sigmas=sigmas, night=eq_night, total=eq_total){
  ci_lower <- binom.confint(night, total, method=c("wilson"), conf.level = sigmas[x])[1,5]
  ci_upper <- binom.confint(night, total, method=c("wilson"), conf.level = sigmas[x])[1,6]
  ci_data <- data.frame(step = x, ci_lower, ci_upper)
}
ci_spacing <- lapply(7:1, conf_steps, sigmas=sigmas, night=eq_night, total=eq_total)
ci_steps <- bind_rows(ci_spacing)

layout(matrix(c(1,1,1,2), ncol=4))

par(mar=c(5,6,4,2))
plot(c(min(0.5,floor(100*ci_steps[1,2])/100), max(0.5,ceiling(100*ci_steps[1,3])/100)),
     y=c(-3,8), type="n", bty="n", yaxt="n", ylab="",
     xlab="Proportion of earthquakes at night")
a <- apply(ci_steps, 1, function(x){
  polygon(c(x[2], x[3], x[3], x[2]), c(0, 0, 1, 1), col=bands[x[1]], border=NA)})
lines(c(.5,.5), c(0,1), col="#FFFFFF")
lines(c(.5,.5), c(0,1), lty=2, col="#777777")
lines(c(eq_night/eq_total,eq_night/eq_total), c(0,1), lwd=2)

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE)
legend(0.5, legend=lbls, lty=typs, lwd=weights, col=bands, bty="n", xjust=0,
       title="Confidence Intervals:", y.intersp=1.1, cex=0.9)

lbls2=c("50% Night", "Actual Proportion")
typs2=c(2,1)
weights2=c(1,2)
cls2=c("#777777", "#000000")
legend(0.7, legend=lbls2, lty=typs2, lwd=weights2, col=cls2, bty="n", xjust=0,
       title="Legend", y.intersp=1.2)

par(mar=old_par$mar)
par(mfrow=c(1,1))

## ---- fig.cap="Proportion of night earthquakes by magnitude, Japan. n=719348"----
old_par=par()
grf <- eq_national %>% mutate(floored_mag = floor(magnitude*2)/2) %>%
  group_by(floored_mag) %>% summarise(successes = sum(eq_is_night), trials=n())

poly_conf_int <- function(success, trials, aa, stepsize, sigma, colr){
  ci <- binom.confint(success, trials, method=c("wilson"), conf.level = sigma)
  lower <- ci[1,5]
  upper <- ci[1,6]
  a <- polygon(x=c(aa,aa+stepsize,aa+stepsize,aa), y=c(upper,upper,lower,lower),
                col=colr, border=NA)
}


```

```

plot7sig <- function(success, trials, aa, stepsize){
  library(binom)
  bands <- rev(c('#fffffcc', '#d9f0a3', '#add8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
  sigmas <- c(0.682689492137086,
             0.954499736103642,
             0.997300203936740,
             0.999936657516334,
             0.999999426696856,
             0.999999998026825,
             0.999999999997440)
  sapply(7:1, function(x){
    poly_conf_int(success, trials, aa, stepsize, sigmas[x], bands[x]))}
  a <- lines(c(aa, aa + stepsize), c(success/trials, success/trials), lwd=2)
}

lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
tys <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
clrs <- rev(c('#fffffcc', '#d9f0a3', '#add8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
#clrs <- c('#fffffb2', '#fed976', '#feb24c', '#fd8d3c', '#fc4e2a', '#e31a1c', '#b10026')

layout(matrix(c(1,1,1,2), ncol=4))
plot(x=c(0,max(grf$floored_mag)+0.5), y=c(0,1), type="n", bty="n",
      xlab="Magnitude (0.5 steps)", ylab="Proportion of earthquakes at night")
a <- apply(grf, 1, function(x){plot7sig(x[2],x[3],x[1],0.5)})
lines(c(0,10), c(.5,.5), col="#FFFFFF")
lines(c(0,10), c(.5,.5), lty=2, col="#777777")

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE)
legend(0.5, legend=lbls, lty=tys, lwd=weights, col=clrs, bty="n", xjust=0,
       title="Confidence Intervals:", cex=0.9)
lbls=c("Expected Proportion", "Actual Proportion")
tys=c(2,1)
weights=c(1,2)
legend(0.7, legend=lbls, lty=tys, lwd=weights, bty="n", xjust=0,
       title="Legend", y.intersp=1.2)

par(mar=old_par$mar)
par(mfrow=c(1,1))

## -----
by_angle <- eq_national %>%
  group_by(eq_angle_by_10) %>% summarise(total= n()) %>%
  mutate(daynight=ifelse(eq_angle_by_10 < 180, "day", "night"))

```

```

merged <- merge(eq_national, angwide, by.x="eq_roundedlat", by.y="lat")
agg_expected <- merged %>% select(`0`:`350`) %>% colSums(na.rm=TRUE)
expected_prop <- agg_expected / sum(agg_expected)
expected <- data.frame(eq_angle_by_10 = as.numeric(names(expected_prop)),
                       expected_prop = as.numeric(expected_prop))
expected$expected_number = expected_prop * eq_total
act_exp <- merge(expected, by_angle, by="eq_angle_by_10", all.x=TRUE)

act_exp$total[is.na(act_exp$total)] <- 0
act_exp$daynight <- NULL
act_exp$act_prop <- act_exp$total / sum(act_exp$total)

ci_brackets <- act_exp %>% ungroup() %>% mutate(grand_total=sum(total)) %>%
  rowwise() %>% mutate(
    ci_lower_7 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[7])[1,5] * grand_total,
    ci_upper_7 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[7])[1,6] * grand_total,
    ci_lower_6 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[6])[1,5] * grand_total,
    ci_upper_6 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[6])[1,6] * grand_total,
    ci_lower_5 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[5])[1,5] * grand_total,
    ci_upper_5 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[5])[1,6] * grand_total,
    ci_lower_4 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[4])[1,5] * grand_total,
    ci_upper_4 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[4])[1,6] * grand_total,
    ci_lower_3 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[3])[1,5] * grand_total,
    ci_upper_3 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[3])[1,6] * grand_total,
    ci_lower_2 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[2])[1,5] * grand_total,
    ci_upper_2 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[2])[1,6] * grand_total,
    ci_lower_1 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[1])[1,5] * grand_total,
    ci_upper_1 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[1])[1,6] * grand_total)

norm_ci <- ci_brackets

for (i in c(4,7:20)){
  norm_ci[,i] <- ci_brackets[,i] - ci_brackets[,3]
}
circlesize=2500

## ---- fig.cap="Over- and under- supply of earthquakes by angle of the sun (10 degree steps).
## Japan. n=719348"----

```

```

norm_ci$border = 2
# need to double entries with a displacement of 10 to make each side
# of the item on the graph
norm_ci2 <- norm_ci
norm_ci2$eq_angle_by_10 <- norm_ci2$eq_angle_by_10 + 10
norm_ci2$border = 1
graphdata <- bind_rows(norm_ci,norm_ci2) %>% arrange(eq_angle_by_10,border)

##### plot graph
bands <- rev(c('#ffffcc','#d9f0a3','#add8e','#78c679','#41ab5d','#238443','#005a32'))
old_par=par()
layout(matrix(c(1,1,1,2), ncol=4))

# overall limits
limits=2 * max(abs(c(graphdata$ci_lower_7, graphdata$ci_upper_7)))
# plot upper confidence 7 interval using plotrix
polar.plot(graphdata$ci_upper_7, polar.pos=graphdata$eq_angle_by_10,
           radial.lim=c(-1*limits,limits),
           labels = "", main=NULL,lwd=0.5, rp.type="p",
           show.grid.labels=FALSE, show.grid=FALSE, mar=c(0,0,0,0),
           grid.col=bands[7], line.col=bands[7], poly.col=bands[7])
# plot upper 6 confidence interval
plot_ci_round <- function(upper_bound,x){
  polar.plot(upper_bound, polar.pos=graphdata$eq_angle_by_10, add=TRUE,
             radial.lim=c(-1*limits,limits),
             line.col=bands[x], lwd=0.5, rp.type="p", poly.col=bands[x])
}
plot_ci_round(graphdata$ci_upper_6, 6)
plot_ci_round(graphdata$ci_upper_5, 5)
plot_ci_round(graphdata$ci_upper_4, 4)
plot_ci_round(graphdata$ci_upper_3, 3)
plot_ci_round(graphdata$ci_upper_2, 2)
plot_ci_round(graphdata$ci_upper_1, 1)
plot_ci_round(graphdata$ci_lower_1, 2)
plot_ci_round(graphdata$ci_lower_2, 3)
plot_ci_round(graphdata$ci_lower_3, 4)
plot_ci_round(graphdata$ci_lower_4, 5)
plot_ci_round(graphdata$ci_lower_5, 6)
plot_ci_round(graphdata$ci_lower_6, 7)
polar.plot(graphdata$ci_lower_7, polar.pos=graphdata$eq_angle_by_10, add=TRUE,
           radial.lim=c(-1*limits,limits),
           line.col="white", lwd=0.5, rp.type="p", poly.col="white")

# plot expected guide line
polar.plot(rep(0,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10, add=TRUE,
           radial.lim=c(-1*limits,limits),
           rp.type="p", lty=4)
# plot 500 less than expected guide line
polar.plot(rep(-1 * circlesize,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10,
           add=TRUE,radial.lim=c(-1*limits,limits),
           rp.type="p", lty=1, line.col="#00000044")
# plot 500 more than expected guide line
polar.plot(rep(circlesize,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10, add=TRUE,

```

```

radial.lim=c(-1*limits,limits),
rp.type="p", lty=3, line.col="#00000044")

lines(c(-1.5,-1.2)*limits, c(0,0))
lines(c(1.5,1.2)*limits, c(0,0))
text(-1.8*limits,0, label="sunset
180", cex=0.7)
text(1.8*limits,0, label="sunrise
0", cex=0.7)

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE, xlab="")
lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
clrs <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
legend(0,4.5, legend=lbls, lty=typs, lwd=weights, col=clrs, bty="n", xjust=0,
       title="Confidence Intervals:", cex=0.9)

lbls2=c("Expected Number", paste(circlesize,"under expected"),
       paste(circlesize,"over expected"))
typs2=c(4,1,3)
weights2=c(1,1,1)
clrs2=c("#000000", "#00000044", "#00000044")
legend(0,10, legend=lbls2, lty=typs2, lwd=weights2, bty="n", xjust=0,
       title="Legend", y.intersp=1.2, col=clrs2)

par(mar=old_par$mar)
par(mfrow=c(1,1))

```

# Chapter 10

## The continental United States is similar

As a second replication, earthquake data from the continental United States is publicly available from USGS (Link 1) and can be downloaded directly using R code.

Then repeating the calculations done with the New Zealand data.

For the same period as the New Zealand Data (from September 2011 to September 2016) Earthquakes data from the United States Geological Service there are 276458 events of depth greater than 0 and magnitude greater than 0.

Of the 276458 in the data, 141119 occurred at night, a proportion of 0.5105. A seven sigma confidence interval for the proportion of earthquakes occurring at night would be 0.5038 to 0.5171. This confidence interval in no way coincides with 0.5, and using one so large we can confidently say that if earthquakes occur randomly, this result would never occur.

While the proportion is not as high as New Zealand's, the greater number of earthquakes means that the confidence intervals are smaller, so there is more certainty that the rate of night earthquakes is not 50%. Based on New Zealand's response to earthquakes differing regionally, I would speculate that most of the United States is regions with a low (but still present) response to nighttime.

Examining magnitude, there is a similar pattern to New Zealand. Unequivocally high numbers of earthquakes in the low magnitudes falling towards 50% as magnitude increases, then becoming unclear as sample size decreases.

The trend for the number of earthquakes by 10 degree arc of the sun is similar to New Zealand- undersupplies

Table 10.1: Data description

feature	value
Earliest (UTC)	2011-09-01 00:01:27
Latest (UTC)	2016-08-31 23:51:15
Northernmost	49.9728333
Southernmost	24.607
Westmost	-124.9966667
Eastmost	-65.065
Percent < Mag 3	98.27
total entries	276458
nighttime quakes	141119

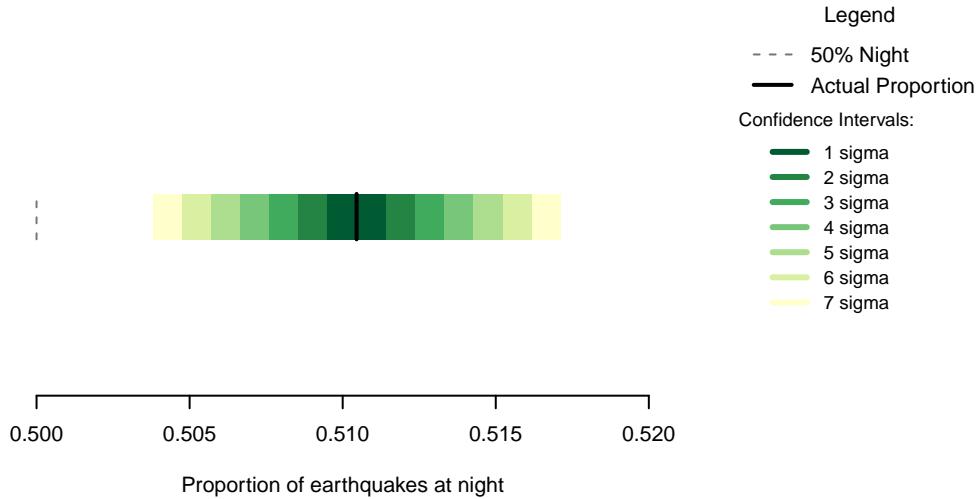


Figure 10.1: Proportion of earthquakes at night: U.S.A. n=278458

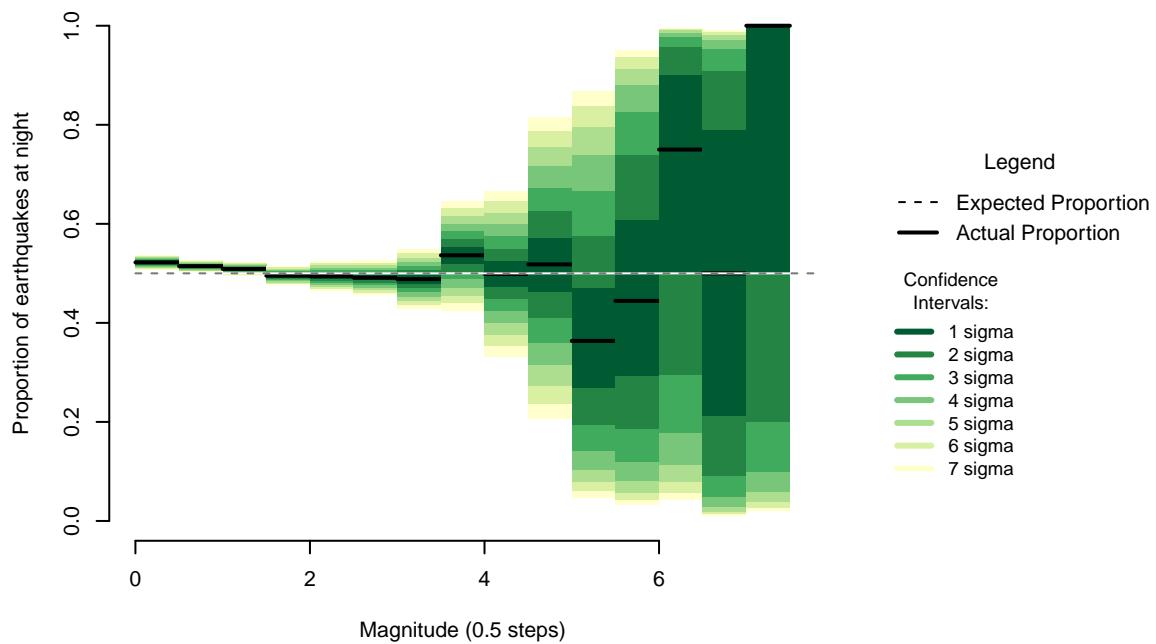


Figure 10.2: Proportion of night earthquakes by magnitude, U.S.A. n=278458

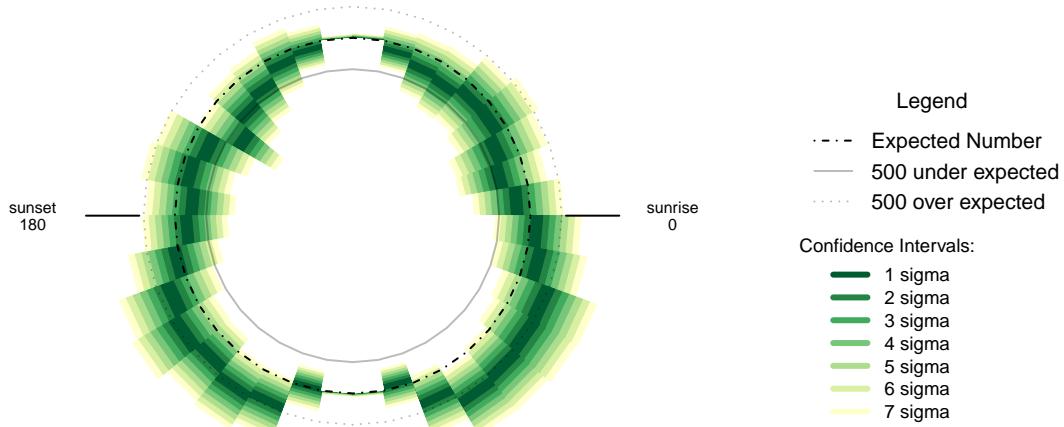


Figure 10.3: Over- and under-supply of earthquakes by angle of the sun (10 degree steps). U.S.A. n=276458

at 30 degrees above the horizon and oversupplies 30-50 degrees below the horizon. The U.S.A. data is skewed slightly compared to New Zealand, as the oversupply is clearly at its greatest when the sun is 30 to 40 degrees below the horizon to the east.

## 10.1 Formal Statement

Earthquakes in the United States of America show the same pattern as New Zealand, displaying an oversupply of earthquakes at night that is not the result of chance. The magnitude pattern of the oversupply is similar to New Zealand's pattern, and the pattern with respect to the position of the sun is similar to that of New Zealand.

## 10.2 Links

1 - USGS: <https://earthquake.usgs.gov>

## 10.3 Chapter Code

```
## ----setup, include=FALSE-----
knitr::opts_chunk$set(echo = FALSE)

## ----c10_001, warnings=FALSE, errors=FALSE, message=FALSE-----
library(geosphere)
library(lubridate, quietly=TRUE)
library(dplyr)
library(binom)
library(ggplot2)
library(maps)
library(mapdata)
library(parallel)
library(readr)
library(plotrix)
```

```

library(tidyr)
library(maptools)
Sys.setenv(TZ = "UTC")

## ----warnings=FALSE, errors=FALSE, message=FALSE-----

if(!dir.exists("../othereqdata")){
  dir.create("../othereqdata")
}
if(!file.exists("../othereqdata/eq_usa_raw.RData")){
  begin <- as.POSIXct("2011-09-01 00:00:00", tz="UTC")
  end <- as.POSIXct("2016-09-15 00:00:00", tz="UTC")
  first <- seq(from=begin, to=end, by="week")
  first <- first[1:(length(first)-1)]
  second <- first[2:length(first)]
  second <- second - seconds(1)
  mid_url <- paste("starttime=", format(first, "%Y-%m-%d"), "%20",
                    format(first, "%H:%M:%S"),
                    "&endtime=", format(second, "%Y-%m-%d"), "%20",
                    format(second, "%H:%M:%S"), sep="")
  secondlast <- "&maxlatitude=50&minlatitude=24.6&maxlongitude=-65&minlongitude=-125&"
  full_url <- paste("https://earthquake.usgs.gov/fdsnws/event/1/query.csv?",
                     mid_url,secondlast,
                     "minmagnitude=0&eventtype=earthquake&orderby=time", sep="")
  usgs <- lapply(full_url[1:(length(full_url)-2)],read.csv, stringsAsFactors=FALSE)
  equsa <- bind_rows(usgs)
  rm(usgs)
  equsa$time_UTC <- as.POSIXct(gsub("\\\\..+", "", as.character(equsa$time)),
                                format="%Y-%m-%dT%H:%M:%S", tz="UTC")
  eq_national <- equsa %>%
    filter(depth > 0 & mag >= 0 &
           time_UTC >=
             as.POSIXct("2011-09-01T00:00:00", format="%Y-%m-%dT%H:%M:%S", tz="UTC") &
           time_UTC <
             as.POSIXct("2016-09-01T00:00:00", format="%Y-%m-%dT%H:%M:%S", tz="UTC")) %>%
    distinct() %>% arrange(time_UTC)

  rm(equsa)
  names(eq_national)[5] <- "magnitude"
  save(eq_national, file="../othereqdata/eq_usa_raw.RData")
}

## ----

if(!file.exists("../othereqdata/eq_usa_processed.RData")){
  load("../othereqdata/eq_usa_raw.RData")
  southmost <- min(eq_national$latitude)
  westmost <- min(eq_national$longitude)
}

```

```

eq_national <- eq_national %>% filter(
  magnitude > 0, depth > 0) %>% rowwise() %>% mutate(
  eq_gridpoint_y = round(
    distVincentyEllipsoid(c(longitude, southmost), c(longitude, latitude)) /50000,0),
  eq_gridpoint_x = round(
    distVincentyEllipsoid(c(westmost, latitude), c(longitude, latitude)) /50000,0),
  eq_roundedlat = destPoint(
    p=c(longitude, southmost), b=0, d=eq_gridpoint_y*50000)[2],
  eq_roundedlong = destPoint(
    p=c(westmost, eq_roundedlat), b=90, d=eq_gridpoint_x*50000)[1]) %>% ungroup()
# use maptools to calculate solar angles
sun_angles <- solarpos(
  matrix(c(eq_national$longitude, eq_national$latitude), ncol=2), eq_national$time_UTC)
colnames(sun_angles) <- c("eq_compass", "eq_vertical")
eq_national <- cbind(eq_national, sun_angles)
eq_national$eq_is_night <- eq_national$eq_vertical < 0
# calculate 360 degree as well as vertical
eq_national <- eq_national %>%
  mutate(eq_angle_360 = eq_vertical,
        eq_angle_360 = ifelse(eq_compass > 180, 180 - eq_angle_360, eq_angle_360),
        eq_angle_360 = ifelse(eq_vertical < 0 & eq_compass <= 180,
                              360 + eq_angle_360, eq_angle_360),
        eq_angle_by_10 = floor(eq_angle_360 /10) * 10)
save(eq_national, file="../othereqdata/eq_usa_processed.RData")
}

## -----
if(!file.exists("../othereqdata/eq_usa_expected.RData")){
  load("../othereqdata/eq_usa_processed.RData")

  lat_range <- unique(eq_national$eq_roundedlat)
  long_med <- median(eq_national$eq_roundedlong)

  # 1 minute intervals for a full solar year
  time1 <- ymd_hms("2015-01-01 00:00:00")
  time2 <- ymd_hms("2015-12-31 23:59:00")
  time_sq <- seq.POSIXt(from=time1, to=time2, by="min")

  calc_angs <- function(x, longinput, timeinput){
    library(dplyr)
    sun_angles <- maptools::solarpos(matrix(c(longinput, x), ncol=2), timeinput)
    colnames(sun_angles) <- c("eq_compass", "eq_vertical")
    # calculate 360 degree as well as vertical
    site_summary <- as.data.frame(sun_angles) %>%
      mutate(eq_angle_360 = eq_vertical,
            eq_angle_360 = ifelse(eq_compass > 180, 180 - eq_angle_360, eq_angle_360),
            eq_angle_360 = ifelse(
              eq_vertical < 0 & eq_compass <= 180, 360 + eq_angle_360, eq_angle_360),
            eq_angle_by_10 = floor(eq_angle_360 /10) * 10) %>%
      group_by(eq_angle_by_10) %>% summarise(total= n())
    site_summary$lat <- x
    return(site_summary)
  }
}

```

```

}

### 
# Calculate the number of cores
no_cores <- detectCores() - 1
# Initiate cluster
cl <- makeCluster(no_cores)
clusterExport(cl, varlist=c("lat_range", "long_med", "time_sq", "calc_angs"))
list_angs <- parLapply(cl, lat_range,
                       function(x){
                           calc_angs(x=x, longinput=long_med, timeinput=time_sq)})
stopCluster(cl)
###

library(tidyr)
anglong <- bind_rows(list_angs)
angwide <- spread(anglong, key=eq_angle_by_10,value=total)
rm(anglong, list_angs, time_sq)
save(angwide, file="../othereqdata/eq_usa_expected.RData")
}

## -----
load("../othereqdata/eq_usa_processed.RData")
load("../othereqdata/eq_usa_expected.RData")
eq_night = sum(eq_national$eq_is_night)
eq_total = nrow(eq_national)
bands <- rev(c('#ffffcc', '#d9f0a3', '#add8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
sigmas <- c(0.682689492137086,
0.954499736103642,
0.997300203936740,
0.999936657516334,
0.999999426696856,
0.99999998026825,
0.99999999997440)
lbls <- c(
"1 sigma", "2 sigma",
"3 sigma", "4 sigma",
"5 sigma", "6 sigma",
"7 sigma")
typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
old_par=par()

## -----
bt <- binom.test(eq_night ,eq_total, conf.level= .999999999997440)

## -----
feature <- c("Earliest (UTC)", "Latest (UTC)",
"Northernmost", "Southernmost",
"Westmost", "Eastmost",
"Percent < Mag 3", "total entries",
"nighttime quakes")
value <- c(as.character(min(eq_national$time_UTC)),
as.character(max(eq_national$time_UTC)),
as.character(max(eq_national$latitude)),
as.character(max(eq_national$longitude)),
as.character(max(eq_national$depth)))

```

```

as.character(min(eq_national$latitude)),
as.character(min(eq_national$longitude)),
as.character(max(eq_national$longitude)),
as.character(round(100*sum(eq_national$magnitude < 3)/eq_total,2)),
as.character(eq_total),
as.character(eq_night))
data.frame(feature,value) %>% knitr::kable(caption="Data description")

## ---- fig.cap="Proportion of earthquakes at night: U.S.A. n=278458"-----
### making the basic proportion graph
eq_night = sum(eq_national$eq_is_night)
eq_total = nrow(eq_national)
bands <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
sigmas <- c(0.682689492137086,
           0.954499736103642,
           0.997300203936740,
           0.999936657516334,
           0.999999426696856,
           0.99999998026825,
           0.9999999997440)
lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
tys <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
old_par=par()
conf_steps <- function(x, sigmas=sigmas, night=eq_night, total=eq_total){
  ci_lower <- binom.confint(night, total, method=c("wilson"), conf.level = sigmas[x])[1,5]
  ci_upper <- binom.confint(night, total, method=c("wilson"), conf.level = sigmas[x])[1,6]
  ci_data <- data.frame(step = x, ci_lower, ci_upper)
}
ci_spacing <- lapply(7:1, conf_steps, sigmas=sigmas, night=eq_night, total=eq_total)
ci_steps <- bind_rows(ci_spacing)

layout(matrix(c(1,1,1,2), ncol=4))

par(mar=c(5,6,4,2))
plot(c(min(0.5,floor(100*ci_steps[1,2])/100), max(0.5,ceiling(100*ci_steps[1,3])/100)),
      y=c(-3,8), type="n", bty="n", yaxt="n", ylab="",
      xlab="Proportion of earthquakes at night")
a <- a <- apply(ci_steps, 1, function(x){
  polygon(c(x[2], x[3], x[3], x[2]), c(0, 0, 1, 1), col=bands[x[1]], border=NA)})
lines(c(.5,.5), c(0,1), col="#FFFFFF")
lines(c(.5,.5), c(0,1), lty=2, col="#777777")
lines(c(eq_night/eq_total,eq_night/eq_total), c(0,1), lwd=2)

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE)
legend(0,5.5, legend=lbls, lty=tys, lwd=weights, col=bands, bty="n", xjust=0,
       title="Confidence Intervals:", y.intersp=1.1, cex=0.9)

```

```

lbls2=c("50% Night", "Actual Proportion")
typs2=c(2,1)
weights2=c(1,2)
cls2=c("#777777", "#000000")
legend(0,7, legend=lbls2, lty=typs2, lwd=weights2, col=cls2, bty="n", xjust=0,
       title="Legend", y.intersp=1.2)

par(mar=old_par$mar)
par(mfrow=c(1,1))

## ---- fig.cap="Proportion of night earthquakes by magnitude, U.S.A. n=278458"----
old_par=par()
grf <- eq_national %>% mutate(floored_mag = floor(magnitude*2)/2) %>%
  group_by(floored_mag) %>% summarise(successes = sum(eq_is_night), trials=n())

poly_conf_int <- function(success, trials, aa, stepsize, sigma, colr){
  ci <- binom.confint(success, trials, method=c("wilson"), conf.level = sigma)
  lower <- ci[1,5]
  upper <- ci[1,6]
  a <- polygon(x=c(aa,aa+stepsize,aa+stepsize,aa), y=c(upper,upper,lower,lower),
                col=colr, border=NA)
}

plot7sig <- function(success, trials, aa, stepsize){
  library(binom)
  #bands <- c('#ffffb2', '#fed976', '#feb24c', '#fd8d3c', '#fc4e2a', '#e31a1c', '#b10026')
  bands <- rev(c('#ffffcc', '#d9f0a3', '#add8e6', '#78c679', '#41ab5d', '#238443', '#005a32'))
  sigmas <- c(0.682689492137086,
             0.954499736103642,
             0.997300203936740,
             0.999936657516334,
             0.999999426696856,
             0.99999998026825,
             0.99999999997440)
  sapply(7:1, function(x){
    poly_conf_int(success, trials, aa, stepsize, sigmas[x], bands[x]))}
  a <- lines(c(aa, aa + stepsize), c(success/trials, success/trials), lwd=2)
}

lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")

typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
clrs <- rev(c('#ffffcc', '#d9f0a3', '#add8e6', '#78c679', '#41ab5d', '#238443', '#005a32'))
#clrs <- c('#ffffb2', '#fed976', '#feb24c', '#fd8d3c', '#fc4e2a', '#e31a1c', '#b10026')

```

```

layout(matrix(c(1,1,1,2), ncol=4))
plot(x=c(0,max(grf$floored_mag)+0.5), y=c(0,1), type="n", bty="n",
      xlab="Magnitude (0.5 steps)", ylab="Proportion of earthquakes at night")
a <- apply(grf,1,function(x){plot7sig(x[2],x[3],x[1],0.5)})
lines(c(0,10), c(.5,.5), col="#FFFFFF")
lines(c(0,10), c(.5,.5), lty=2, col="#777777")

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE)
legend(0.5, legend=lbls, lty=typs, lwd=weights, col=clrs, bty="n", xjust=0,
       title="Confidence
Intervals:", cex=0.9)
lbls=c("Expected Proportion", "Actual Proportion")
typs=c(2,1)
weights=c(1,2)
legend(0.7, legend=lbls, lty=typs, lwd=weights, bty="n", xjust=0,
       title="Legend", y.intersp=1.2)

par(mar=old_par$mar)
par(mfrow=c(1,1))

## -----
by_angle <- eq_national %>%
  group_by(eq_angle_by_10) %>% summarise(total= n()) %>%
  mutate(daynight=ifelse(eq_angle_by_10 < 180, "day", "night"))

merged <- merge(eq_national, angwide, by.x="eq_roundedlat", by.y="lat")
agg_expected <- merged %>% select(`0`:`350`) %>% colSums(na.rm=TRUE)
expected_prop <- agg_expected / sum(agg_expected)
expected <- data.frame(eq_angle_by_10 = as.numeric(names(expected_prop)),
                       expected_prop = as.numeric(expected_prop))
expected$expected_number = expected_prop * eq_total
act_exp <- merge(expected, by_angle, by="eq_angle_by_10", all.x=TRUE)

act_exp$total[is.na(act_exp$total)] <- 0
act_exp$daynight <- NULL
act_exp$act_prop <- act_exp$total / sum(act_exp$total)

ci_brackets <- act_exp %>% ungroup() %>% mutate(grand_total=sum(total)) %>%
  rowwise() %>% mutate(
    ci_lower_7 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[7])[1,5] * grand_total,
    ci_upper_7 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[7])[1,6] * grand_total,
    ci_lower_6 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[6])[1,5] * grand_total,
    ci_upper_6 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[6])[1,6] * grand_total,
    ci_lower_5 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[5])[1,5] * grand_total,
    ci_upper_5 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[5])[1,6] * grand_total,
    ci_lower_4 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[4])[1,5] * grand_total,
    ci_upper_4 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[4])[1,6] * grand_total,
    ci_lower_3 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[3])[1,5] * grand_total,
    ci_upper_3 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[3])[1,6] * grand_total,
    ci_lower_2 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[2])[1,5] * grand_total,
    ci_upper_2 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[2])[1,6] * grand_total,
    ci_lower_1 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[1])[1,5] * grand_total,
    ci_upper_1 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[1])[1,6] * grand_total)
  
```

```

ci_upper_5 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[5])[1,6] * grand_total,
ci_lower_4 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[4])[1,5] * grand_total,
ci_upper_4 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[4])[1,6] * grand_total,
ci_lower_3 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[3])[1,5] * grand_total,
ci_upper_3 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[3])[1,6] * grand_total,
ci_lower_2 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[2])[1,5] * grand_total,
ci_upper_2 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[2])[1,6] * grand_total,
ci_lower_1 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[1])[1,5] * grand_total,
ci_upper_1 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[1])[1,6] * grand_total)

norm_ci <- ci_brackets

for (i in c(4,7:20)){
norm_ci[,i] <- ci_brackets[,i] - ci_brackets[,3]
}
circlesize=500

## ---- fig.cap="Over- and under- supply of earthquakes by angle of the sun
## (10 degree steps). U.S.A. n=276458"----
norm_ci$border = 2
# need to double entries with a displacement of 10
# to make each side of the item on the graph
norm_ci2 <- norm_ci
norm_ci2$eq_angle_by_10 <- norm_ci2$eq_angle_by_10 + 10
norm_ci2$border = 1
graphdata <- bind_rows(norm_ci,norm_ci2) %>% arrange(eq_angle_by_10,border)

#### plot graph
bands <- rev(c('#ffffcc', '#d9f0a3', '#add8e6', '#78c679', '#41ab5d', '#238443', '#005a32'))
old_par=par()
layout(matrix(c(1,1,1,2), ncol=4))

# overall limits
limits=2 * max(abs(c(graphdata$ci_lower_7, graphdata$ci_upper_7)))
# plot upper confidence 7 interval using plotrix
polar.plot(graphdata$ci_upper_7, polar.pos=graphdata$eq_angle_by_10,
            radial.lim=c(-1*limits,limits),
            labels = "", main=NULL, lwd=0.5, rp.type="p",
            show.grid.labels=FALSE, show.grid=FALSE, mar=c(0,0,0,0),
            grid.col=bands[7], line.col=bands[7], poly.col=bands[7])
# plot upper 6 confidence interval
plot_ci_round <- function(upper_bound,x){
  polar.plot(upper_bound, polar.pos=graphdata$eq_angle_by_10, add=TRUE,
             radial.lim=c(-1*limits,limits),

```

```

        line.col=bands[x], lwd=0.5, rp.type="p", poly.col=bands[x])
    }
plot_ci_round(graphdata$ci_upper_6, 6)
plot_ci_round(graphdata$ci_upper_5, 5)
plot_ci_round(graphdata$ci_upper_4, 4)
plot_ci_round(graphdata$ci_upper_3, 3)
plot_ci_round(graphdata$ci_upper_2, 2)
plot_ci_round(graphdata$ci_upper_1, 1)
plot_ci_round(graphdata$ci_lower_1, 2)
plot_ci_round(graphdata$ci_lower_2, 3)
plot_ci_round(graphdata$ci_lower_3, 4)
plot_ci_round(graphdata$ci_lower_4, 5)
plot_ci_round(graphdata$ci_lower_5, 6)
plot_ci_round(graphdata$ci_lower_6, 7)
polar.plot(graphdata$ci_lower_7, polar.pos=graphdata$eq_angle_by_10,
            add=TRUE, radial.lim=c(-1*limits,limits),
            line.col="white", lwd=0.5, rp.type="p", poly.col="white")

# plot expected guide line
polar.plot(rep(0,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10, add=TRUE,
            radial.lim=c(-1*limits,limits),
            rp.type="p", lty=4)
# plot 500 less than expected guide line
polar.plot(rep(-1 * circlesize,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10,
            add=TRUE,radial.lim=c(-1*limits,limits),
            rp.type="p", lty=1, line.col="#00000044")
# plot 500 more than expected guide line
polar.plot(rep(circlesize,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10,
            add=TRUE,radial.lim=c(-1*limits,limits),
            rp.type="p", lty=3, line.col="#00000044")

lines(c(-1.5,-1.2)*limits, c(0,0))
lines(c(1.5,1.2)*limits, c(0,0))
text(-1.8*limits,0, label="sunset
180", cex=0.7)
text(1.8*limits,0, label="sunrise
0", cex=0.7)

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE, xlab="")
lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
clrs <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
legend(0,4.5, legend=lbls, lty=typs, lwd=weights, col=clrs, bty="n", xjust=0,
       title="Confidence Intervals:", cex=0.9)

lbls2=c("Expected Number", paste(circlesize,"under expected"),

```

```
    paste(circlesize,"over expected"))
typs2=c(4,1,3)
weights2=c(1,1,1)
clrs2=c("#000000","#00000044","#00000044")
legend(0,10, legend=lbls2, lty=typs2, lwd=weights2, bty="n", xjust=0,
       title="Legend", y.intersp=1.2, col=clrs2)

par(mar=old_par$mar)
par(mfrow=c(1,1))
```

# Chapter 11

## Italian earthquakes are the same

Earthquake data for Italy is publicly available from Istituto Nazionale di Geofisica e Vulcanologia (Link 1), and can be downloaded directly from within R. As there is a 10000 record limit on individual downloads, I downloaded the records on a monthly basis.

For the same period as the New Zealand Data (from September 2011 to September 2016), in earthquakes data from the United Istituto Nazionale di Geofisica e Vulcanologia there are 103093 events of depth greater than 0 and magnitude greater than 0.

Of the 103093 in the data, 54886 occurred at night, a proportion of 0.5324. A seven sigma confidence interval for the proportion of earthquakes occurring at night would be 0.5215 to 0.5433. This confidence interval in no way coincides with 0.5, and using one so large we can confidently say that if earthquakes occur randomly, this result would never occur.

While the proportion is not as high as New Zealand's, it is still significant that the rate of night earthquakes is not 50%.

Examining magnitude, there is a similar pattern to New Zealand. Unequivocally high numbers of earthquakes in the low magnitudes falling towards 50% as magnitude increases, then becoming unclear as sample size decreases.

The trend for the number of earthquakes by 10 degree arc of the sun is similar to New Zealand- undersupplies at 30 degrees above the horizon and oversupplies 30-40 degrees below the horizon. Italy, like Japan and the United States, is clearly skewed to a peak oversupply when the sun is below the horizon to the east, and peak undersupply when the sun is 180 degrees away.

Table 11.1: Data description

feature	value
Earliest (UTC)	2011-09-01 01:14:52
Latest (UTC)	2016-08-31 23:54:09
Northernmost	48.4088
Southernmost	35.0485
Westmost	5.272
Eastmost	19.92
Percent < Mag 3	98.42
total entries	103093
nighttime quakes	54886

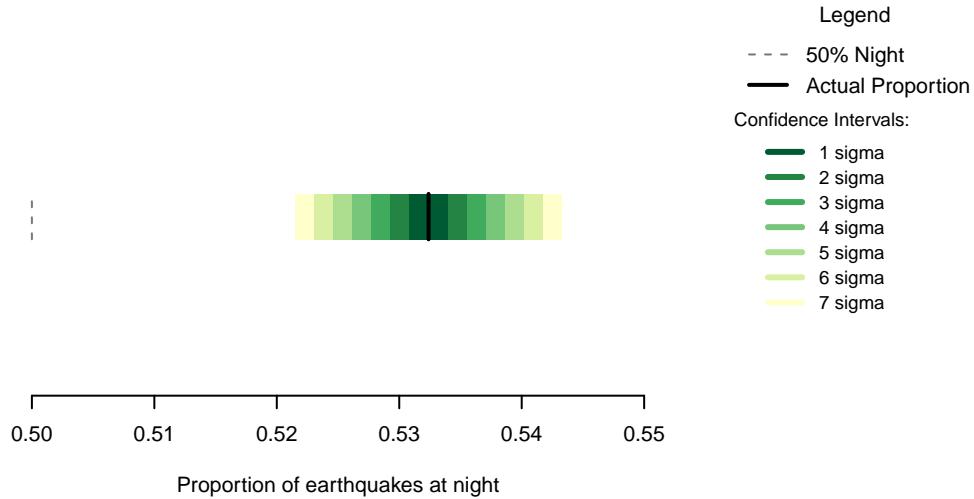


Figure 11.1: Proportion of earthquakes at night: Italy. n=103093

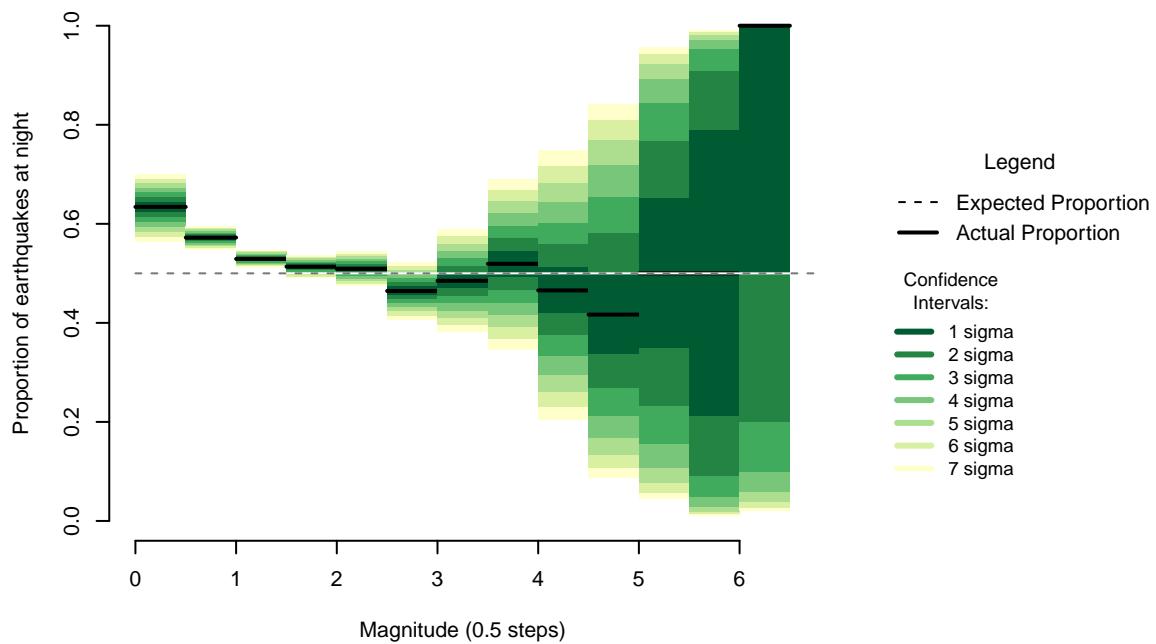


Figure 11.2: Proportion of night earthquakes by magnitude, Italy. n=103093

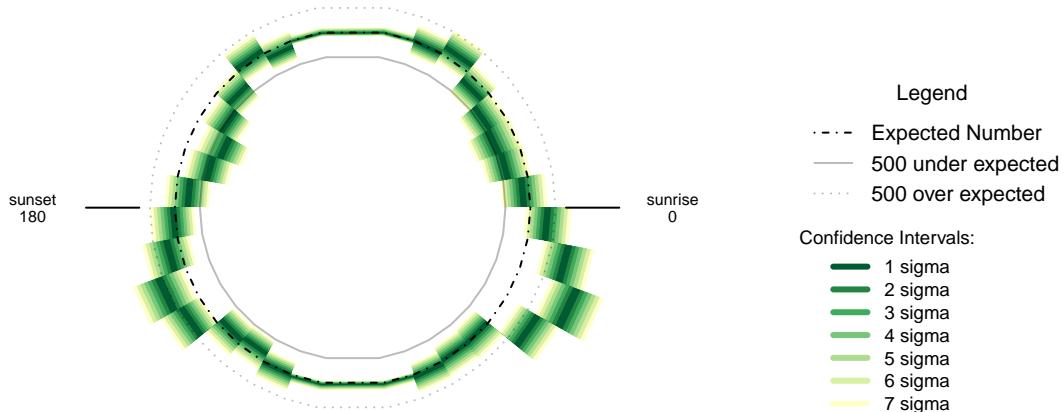


Figure 11.3: Over- and under-supply of earthquakes by angle of the sun (10 degree steps). Italy. n=103093

## 11.1 Formal Statement

Earthquakes in Italy show the same pattern as New Zealand, displaying an oversupply of earthquakes at night that is not the result of chance. The magnitude pattern of the oversupply is similar to New Zealand's pattern, and the pattern with respect to the position of the sun is similar to that of New Zealand.

## 11.2 Links

1 - Data and results published on this website by Istituto Nazionale di Geofisica e Vulcanologia are licensed under a Creative Commons Attribution 4.0 International License. Based on a work at National Earthquake Center. <http://cnt.rm.ingv.it/en>

## 11.3 Chapter Code

```
## ----setup, include=FALSE-----
knitr::opts_chunk$set(echo = FALSE)

## ---- warnings=FALSE, errors=FALSE, message=FALSE-----
library(geosphere)
library(lubridate, quietly=TRUE)
library(dplyr)
library(binom)
library(ggplot2)
library(maps)
library(mapdata)
library(parallel)
library(readr)
library(plotrix)
library(tidyr)
library(mapproj)
Sys.setenv(TZ = "UTC")
```

```
## ----warnings=FALSE, errors=FALSE, message=FALSE-----

if(!dir.exists("../othereqdata")){
  dir.create("../othereqdata")
}
if(!file.exists("../othereqdata/eq_italy_raw.RData")){

# Data and results published on this website by
# Istituto Nazionale di Geofisica e Vulcanologia are licensed under a Creative
# Commons Attribution 4.0 International License.
# Based on a work at National Earthquake Center.
# http://cnt.rm.ingv.it/en

begin <- as.Date("2011-09-01")
end <- as.Date("2016-09-01")
mnths <- as.character(seq(from=begin, to=end, by="month"))
bit1 <- "T23%3A59%3A59&minmag=0&maxmag=10&mindepth=-10&maxdepth=1000&"
bit2 <- "minlat=35&maxlat=49&minlon=5&maxlon=20&minversion=100&"
urls <- paste("http://webservices.ingv.it/fdsnws/event/1/query?starttime=",
             mnths[1:(length(mnths)-1)],
             "T00%3A00%3A00&endtime=", mnths[2:length(mnths)],
             bit1, bit2,
             "orderby=time-asc&format=text&limit=10000", sep="")

csvs <- lapply(urls, read.csv, sep="|", stringsAsFactors=FALSE)
eqit <- bind_rows(csvs)

names(eqit) <- c("event", "time", "latitude", "longitude", "depth",
                 "author", "Catalog", "Contributor", "ContributorID",
                 "MagType", "magnitude", "MagAuthor", "EventLocationName")

eqit$time_UTC <- as.POSIXct(as.character(eqit$time),
                             format="%Y-%m-%dT%H:%M:%OS", tz="UTC")

eq_national <- eqit %>% filter(depth > 0 & magnitude >= 0 &
                                    time_UTC >= as.POSIXct("2011-09-01T00:00:00",
                                                          format="%Y-%m-%dT%H:%M:%S", tz="UTC") &
                                    time_UTC < as.POSIXct("2016-09-01T00:00:00",
                                                          format="%Y-%m-%dT%H:%M:%S", tz="UTC")) %>%
  distinct() %>% arrange(time_UTC)

rm(csvs, eqit)
# calling the dataset eq_national so all the rest of the processing
# code is the same for all countries
# contains variables latitude, longitude, magnitude, and time_UTC
save(eq_national, file="../othereqdata/eq_italy_raw.RData")
}

## ----

if(!file.exists("../othereqdata/eq_italy_processed.RData")){
```

```

load("../othereqdata/eq_italy_raw.RData")
southmost <- min(eq_national$latitude)
westmost <- min(eq_national$longitude)

eq_national <- eq_national %>% filter(
  magnitude > 0, depth > 0) %>% rowwise() %>% mutate(
  eq_gridpoint_y = round(
    distVincentyEllipsoid(c(longitude, southmost), c(longitude, latitude)) /50000,0),
  eq_gridpoint_x = round(
    distVincentyEllipsoid(c(westmost, latitude), c(longitude, latitude)) /50000,0),
  eq_roundedlat = destPoint(
    p=c(longitude, southmost), b=0, d=eq_gridpoint_y*50000)[2],
  eq_roundedlong = destPoint(
    p=c(westmost, eq_roundedlat), b=90, d=eq_gridpoint_x*50000)[1]) %>% ungroup()
# use maptools to calculate solar angles
sun_angles <- solarpos(
  matrix(c(eq_national$longitude, eq_national$latitude), ncol=2), eq_national$time_UTC)
colnames(sun_angles) <- c("eq_compass", "eq_vertical")
eq_national <- cbind(eq_national, sun_angles)
eq_national$eq_is_night <- eq_national$eq_vertical < 0
# calculate 360 degree as well as vertical
eq_national <- eq_national %>%
  mutate(eq_angle_360 = eq_vertical,
        eq_angle_360 = ifelse(eq_compass > 180, 180 - eq_angle_360, eq_angle_360),
        eq_angle_360 = ifelse(eq_vertical < 0 & eq_compass <= 180,
                             360 + eq_angle_360, eq_angle_360),
        eq_angle_by_10 = floor(eq_angle_360 /10) * 10)
save(eq_national, file="../othereqdata/eq_italy_processed.RData")
}

## -----
if(!file.exists("../othereqdata/eq_italy_expected.RData")){
  load("../othereqdata/eq_italy_processed.RData")

  lat_range <- unique(eq_national$eq_roundedlat)
  long_med <- median(eq_national$eq_roundedlong)

  # 1 minute intervals for a full solar year
  time1 <- ymd_hms("2015-01-01 00:00:00")
  time2 <- ymd_hms("2015-12-31 23:59:00")
  time_sq <- seq.POSIXt(from=time1, to=time2, by="min")

  calc_angs <- function(x, longinput, timeinput){
    library(dplyr)
    sun_angles <- maptools::solarpos(matrix(c(longinput, x), ncol=2), timeinput)
    colnames(sun_angles) <- c("eq_compass", "eq_vertical")
    # calculate 360 degree as well as vertical
    site_summary <- as.data.frame(sun_angles) %>%
      mutate(eq_angle_360 = eq_vertical,
            eq_angle_360 = ifelse(eq_compass > 180, 180 - eq_angle_360, eq_angle_360),
            eq_angle_360 = ifelse(

```

```

        eq_vertical < 0 & eq_compass <= 180, 360 + eq_angle_360, eq_angle_360),
eq_angle_by_10 = floor(eq_angle_360 /10) * 10) %>%
group_by(eq_angle_by_10) %>% summarise(total= n())
site_summary$lat <- x
return(site_summary)
}

###  

# Calculate the number of cores
no_cores <- detectCores() - 1
# Initiate cluster
cl <- makeCluster(no_cores)
clusterExport(cl, varlist=c("lat_range", "long_med", "time_sq", "calc_angs"))
list_angs <- parLapply(cl, lat_range,
                       function(x){
                           calc_angs(x=x, longinput=long_med, timeinput=time_sq)})
stopCluster(cl)
###  

library(tidyr)
anglong <- bind_rows(list_angs)
angwide <- spread(anglong, key=eq_angle_by_10,value=total)
rm(anglong, list_angs, time_sq)
save(angwide, file="../othereqdata/eq_italy_expected.RData")
}

## -----
load("../othereqdata/eq_italy_processed.RData")
load("../othereqdata/eq_italy_expected.RData")
eq_night = sum(eq_national$eq_is_night)
eq_total = nrow(eq_national)
bands <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
sigmas <- c(0.682689492137086,
0.954499736103642,
0.997300203936740,
0.999936657516334,
0.999999426696856,
0.99999998026825,
0.999999999997440)
lbls <- c(
"1 sigma", "2 sigma",
"3 sigma", "4 sigma",
"5 sigma", "6 sigma",
"7 sigma")
typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
old_par=par()

## -----
bt <- binom.test(eq_night ,eq_total, conf.level= .999999999997440)

## -----
feature <- c("Earliest (UTC)", "Latest (UTC)",
"Northernmost", "Southernmost",
"Westmost", "Eastmost",

```

```

    "Percent < Mag 3", "total entries",
    "nighttime quakes")
value <- c(as.character(min(eq_national$time_UTC)),
           as.character(max(eq_national$time_UTC)),
           as.character(max(eq_national$latitude)),
           as.character(min(eq_national$latitude)),
           as.character(min(eq_national$longitude)),
           as.character(max(eq_national$longitude)),
           as.character(round(100*sum(eq_national$magnitude < 3)/eq_total,2)),
           as.character(eq_total),
           as.character(eq_night))
data.frame(feature,value) %>% knitr::kable(caption="Data description")

## ---- fig.cap="Proportion of earthquakes at night: Italy. n=103093"-----
### making the basic proportion graph
eq_night = sum(eq_national$eq_is_night)
eq_total = nrow(eq_national)
bands <- rev(c('#ffffcc','#d9f0a3','#addd8e','#78c679','#41ab5d','#238443','#005a32'))
sigmas <- c(0.682689492137086,
           0.954499736103642,
           0.997300203936740,
           0.999936657516334,
           0.999999426696856,
           0.99999998026825,
           0.9999999997440)
lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
old_par=par()
conf_steps <- function(x, sigmas=sigmas, night=eq_night, total=eq_total){
  ci_lower <- binom.confint(night, total, method=c("wilson"), conf.level = sigmas[x])[1,5]
  ci_upper <- binom.confint(night, total, method=c("wilson"), conf.level = sigmas[x])[1,6]
  ci_data <- data.frame(step = x, ci_lower, ci_upper)
}
ci_spacing <- lapply(7:1, conf_steps, sigmas=sigmas, night=eq_night, total=eq_total)
ci_steps <- bind_rows(ci_spacing)

layout(matrix(c(1,1,1,2), ncol=4))

par(mar=c(5,6,4,2))
plot(c(min(0.5,floor(100*ci_steps[1,2])/100), max(0.5,ceiling(100*ci_steps[1,3])/100)),
      y=c(-3,8), type="n", bty="n", yaxt="n", ylab="",
      xlab="Proportion of earthquakes at night")
a <- a <- apply(ci_steps, 1, function(x){
  polygon(c(x[2], x[3], x[3], x[2]), c(0, 0, 1, 1), col=bands[x[1]], border=NA))
  lines(c(.5,.5), c(0,1), col="#FFFFFF")
  lines(c(.5,.5), c(0,1), lty=2, col="#777777")
  lines(c(eq_night/eq_total,eq_night/eq_total), c(0,1), lwd=2)
})

```

```

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE)
legend(0,5.5, legend=lbls, lty=typs, lwd=weights, col=bands, bty="n", xjust=0,
       title="Confidence Intervals:", y.intersp=1.1, cex=0.9)

lbls2=c("50% Night", "Actual Proportion")
typs2=c(2,1)
weights2=c(1,2)
cls2=c("#777777", "#000000")
legend(0,7, legend=lbls2, lty=typs2, lwd=weights2, col=cls2, bty="n", xjust=0,
       title="Legend", y.intersp=1.2)

par(mar=old_par$mar)
par(mfrow=c(1,1))

## ---- fig.cap="Proportion of night earthquakes by magnitude, Italy. n=103093"----
old_par=par()
grf <- eq_national %>% mutate(floored_mag = floor(magnitude*2)/2) %>%
  group_by(floored_mag) %>% summarise(successes = sum(eq_is_night), trials=n())

poly_conf_int <- function(success, trials, aa, stepsize, sigma, colr){
  ci <- binom.confint(success, trials, method=c("wilson"), conf.level = sigma)
  lower <- ci[1,5]
  upper <- ci[1,6]
  a <- polygon(x=c(aa,aa+stepsize,aa+stepsize,aa), y=c(upper,upper,lower,lower),
                col=colr, border=NA)
}

plot7sig <- function(success, trials, aa, stepsize){
  library(binom)
  #bands <- c('#ffffb2', '#fed976', '#feb24c', '#fd8d3c', '#fc4e2a', '#e31a1c', '#b10026')
  bands <- rev(c('#ffffcc', '#d9f0a3', '#add8e6', '#78c679', '#41ab5d', '#238443', '#005a32'))
  sigmas <- c(0.682689492137086,
             0.954499736103642,
             0.997300203936740,
             0.999936657516334,
             0.999999426696856,
             0.999999998026825,
             0.99999999997440)
  sapply(7:1, function(x){
    poly_conf_int(success, trials, aa, stepsize, sigmas[x], bands[x])})
  a <- lines(c(aa, aa + stepsize), c(success/trials, success/trials), lwd=2)
}

lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",

```

```

"7 sigma")
typs <- c(1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3)
clrs <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
#clrs <- c('#ffffb2', '#fed976', '#feb24c', '#fd8d3c', '#fc4e2a', '#e31a1c', '#b10026')

layout(matrix(c(1,1,1,2), ncol=4))
plot(x=c(0,max(grf$floored_mag)+0.5), y=c(0,1), type="n", bty="n",
      xlab="Magnitude (0.5 steps)", ylab="Proportion of earthquakes at night")
a <- apply(grf, 1, function(x){plot7sig(x[2],x[3],x[1],0.5)})
lines(c(0,10), c(.5,.5), col="#FFFFFF")
lines(c(0,10), c(.5,.5), lty=2, col="#777777")

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE)
legend(0.5, legend=lbls, lty=typs, lwd=weights, col=clrs, bty="n", xjust=0,
       title="Confidence Intervals:", cex=0.9)
lbls=c("Expected Proportion", "Actual Proportion")
typs=c(2,1)
weights=c(1,2)
legend(0.7, legend=lbls, lty=typs, lwd=weights, bty="n", xjust=0,
       title="Legend", y.intersp=1.2)

par(mar=old_par$mar)
par(mfrow=c(1,1))

## -----
by_angle <- eq_national %>%
  group_by(eq_angle_by_10) %>% summarise(total= n()) %>%
  mutate(daynight=ifelse(eq_angle_by_10 < 180, "day", "night"))

merged <- merge(eq_national, angwide, by.x="eq_roundedlat", by.y="lat")
agg_expected <- merged %>% select(`0`:`350`) %>% colSums(na.rm=TRUE)
expected_prop <- agg_expected / sum(agg_expected)
expected <- data.frame(eq_angle_by_10 = as.numeric(names(expected_prop)),
                       expected_prop = as.numeric(expected_prop))
expected$expected_number = expected_prop * eq_total
act_exp <- merge(expected, by_angle, by="eq_angle_by_10", all.x=TRUE)

act_exp$total[is.na(act_exp$total)] <- 0
act_exp$daynight <- NULL
act_exp$act_prop <- act_exp$total / sum(act_exp$total)

ci_brackets <- act_exp %>% ungroup() %>% mutate(grand_total=sum(total)) %>%
  rowwise() %>% mutate(
    ci_lower_7 = binom.confint(total, grand_total, method=c("wilson"),
                               conf.level = sigmas[7])[1,5] * grand_total,
    ci_upper_7 = binom.confint(total, grand_total, method=c("wilson"),
                               conf.level = sigmas[7])[1,6] * grand_total,

```

```

ci_lower_6 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[6])[1,5] * grand_total,
ci_upper_6 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[6])[1,6] * grand_total,
ci_lower_5 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[5])[1,5] * grand_total,
ci_upper_5 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[5])[1,6] * grand_total,
ci_lower_4 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[4])[1,5] * grand_total,
ci_upper_4 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[4])[1,6] * grand_total,
ci_lower_3 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[3])[1,5] * grand_total,
ci_upper_3 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[3])[1,6] * grand_total,
ci_lower_2 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[2])[1,5] * grand_total,
ci_upper_2 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[2])[1,6] * grand_total,
ci_lower_1 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[1])[1,5] * grand_total,
ci_upper_1 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[1])[1,6] * grand_total)

norm_ci <- ci_brackets

for (i in c(4,7:20)){
norm_ci[,i] <- ci_brackets[,i] - ci_brackets[,3]
}
circlesize=500

## ---- fig.cap="Over- and under- supply of earthquakes by angle of the sun
## (10 degree steps). Italy. n=103093"----
norm_ci$border = 2
# need to double entries with a displacement of 10
# to make each side of the item on the graph
norm_ci2 <- norm_ci
norm_ci2$eq_angle_by_10 <- norm_ci2$eq_angle_by_10 + 10
norm_ci2$border = 1
graphdata <- bind_rows(norm_ci,norm_ci2) %>% arrange(eq_angle_by_10,border)

##### plot graph
bands <- rev(c('#ffffcc','#d9f0a3','#add8e','#78c679','#41ab5d','#238443','#005a32'))
old_par=par()
layout(matrix(c(1,1,1,2), ncol=4))

# overall limits
limits=2 * max(abs(c(graphdata$ci_lower_7, graphdata$ci_upper_7)))
# plot upper confidence 7 interval using plotrix
polar.plot(graphdata$ci_upper_7, polar.pos=graphdata$eq_angle_by_10,
            radial.lim=c(-1*limits,limits),
            labels = "", main=NULL,lwd=0.5, rp.type="p",

```

```

    show.grid.labels=FALSE, show.grid=FALSE, mar=c(0,0,0,0),
    grid.col=bands[7], line.col=bands[7], poly.col=bands[7])
# plot upper 6 confidence interval
plot_ci_round <- function(upper_bound,x){
  polar.plot(upper_bound, polar.pos=graphdata$eq_angle_by_10, add=TRUE,
             radial.lim=c(-1*limits,limits),
             line.col=bands[x], lwd=0.5, rp.type="p", poly.col=bands[x])
}
plot_ci_round(graphdata$ci_upper_6, 6)
plot_ci_round(graphdata$ci_upper_5, 5)
plot_ci_round(graphdata$ci_upper_4, 4)
plot_ci_round(graphdata$ci_upper_3, 3)
plot_ci_round(graphdata$ci_upper_2, 2)
plot_ci_round(graphdata$ci_upper_1, 1)
plot_ci_round(graphdata$ci_lower_1, 2)
plot_ci_round(graphdata$ci_lower_2, 3)
plot_ci_round(graphdata$ci_lower_3, 4)
plot_ci_round(graphdata$ci_lower_4, 5)
plot_ci_round(graphdata$ci_lower_5, 6)
plot_ci_round(graphdata$ci_lower_6, 7)
polar.plot(graphdata$ci_lower_7, polar.pos=graphdata$eq_angle_by_10,
           add=TRUE, radial.lim=c(-1*limits,limits),
           line.col="white", lwd=0.5, rp.type="p", poly.col="white")

# plot expected guide line
polar.plot(rep(0,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10, add=TRUE,
           radial.lim=c(-1*limits,limits),
           rp.type="p", lty=4)
# plot 500 less than expected guide line
polar.plot(rep(-1 * circlesize,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10,
           add=TRUE,radial.lim=c(-1*limits,limits),
           rp.type="p", lty=1, line.col="#00000044")
# plot 500 more than expected guide line
polar.plot(rep(circlesize,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10,
           add=TRUE,radial.lim=c(-1*limits,limits),
           rp.type="p", lty=3, line.col="#00000044")

lines(c(-1.5,-1.2)*limits, c(0,0))
lines(c(1.5,1.2)*limits, c(0,0))
text(-1.8*limits,0, label="sunset
180", cex=0.7)
text(1.8*limits,0, label="sunrise
0", cex=0.7)

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE, xlab="")
lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
typs <- c(1,1,1,1,1,1,1)

```

```
weights <- c(3,3,3,3,3,3,3)
clrs <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
legend(0,4.5, legend=lbls, lty=typs, lwd=weights, col=clrs, bty="n", xjust=0,
      title="Confidence Intervals:", cex=0.9)

lbls2=c("Expected Number", paste(circlesize,"under expected"),
       paste(circlesize,"over expected"))
typs2=c(4,1,3)
weights2=c(1,1,1)
clrs2=c("#000000", "#00000044", "#00000044")
legend(0,10, legend=lbls2, lty=typs2, lwd=weights2, bty="n", xjust=0,
       title="Legend", y.intersp=1.2, col=clrs2)

par(mar=old_par$mar)
par(mfrow=c(1,1))
```

# Chapter 12

## Hawaii is similar

As Hawaii is part of the United States, data for the Hawaiian region can be obtained from U.S.G.S. in a similar manner to the continental United States

For the same period as the New Zealand Data (from September 2011 to September 2016) Earthquakes data from the United States Geological Service there are 10181 events of depth greater than 0 and magnitude greater than 0.

Of the 10181 in the data, 5309 occurred at night, a proportion of 0.5215. A seven sigma confidence interval for the proportion of earthquakes occurring at night would be 0.4867 to 0.5561, however the proportion is between 4 and 5 confidence intervals.

While 4 confidence intervals would normally be a significant result in its own right, I can also observe Hawaii has a higher proportion of nighttime earthquakes than the continental United States, it is just that with around 5% the number of earthquakes we are less certain about the exact proportion.

Examining magnitude, there is a similar pattern to New Zealand. Even with the small number of earthquakes, it is clear that earthquakes of a small magnitude are occurring more often at night.

The trend for the number of earthquakes by 10 degree arc of the sun is similar to New Zealand, but much less pronounced. Peak oversupply is around 40 to 50 degrees below the horizon. Peak undersupply is 30 to 50 degrees above the horizon.

Table 12.1: Data description

feature	value
Earliest (UTC)	2011-09-07 04:13:11
Latest (UTC)	2016-08-31 15:28:29
Northernmost	28.9173
Southernmost	17.4758
Westmost	179.9979
Eastmost	-153.6606598
Percent < Mag 3	97.14
total entries	10181
nighttime quakes	5309

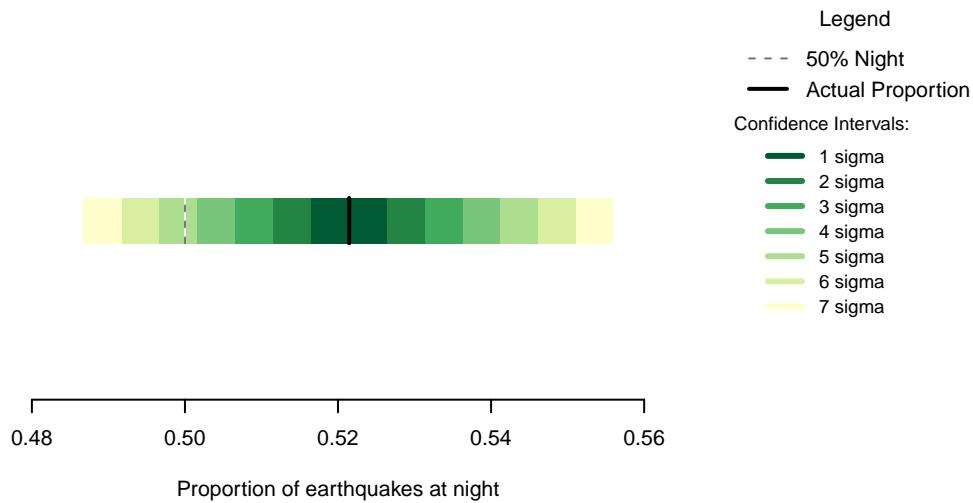


Figure 12.1: Proportion of earthquakes at night: Hawaii. n=10181

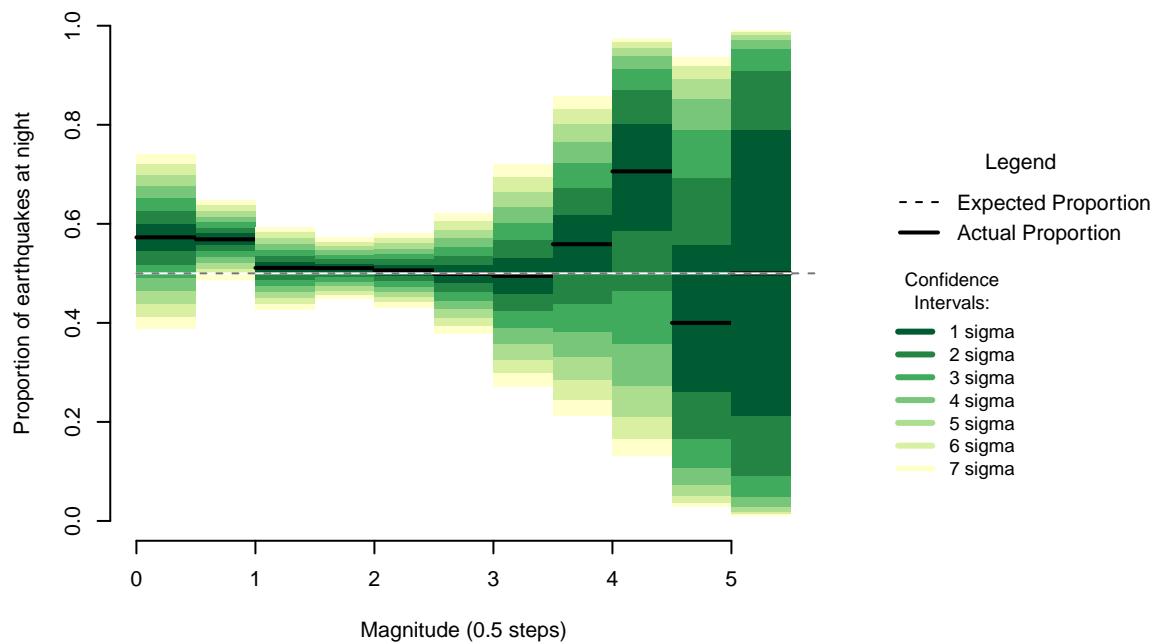


Figure 12.2: Proportion of night earthquakes by magnitude, Hawaii. n=10181

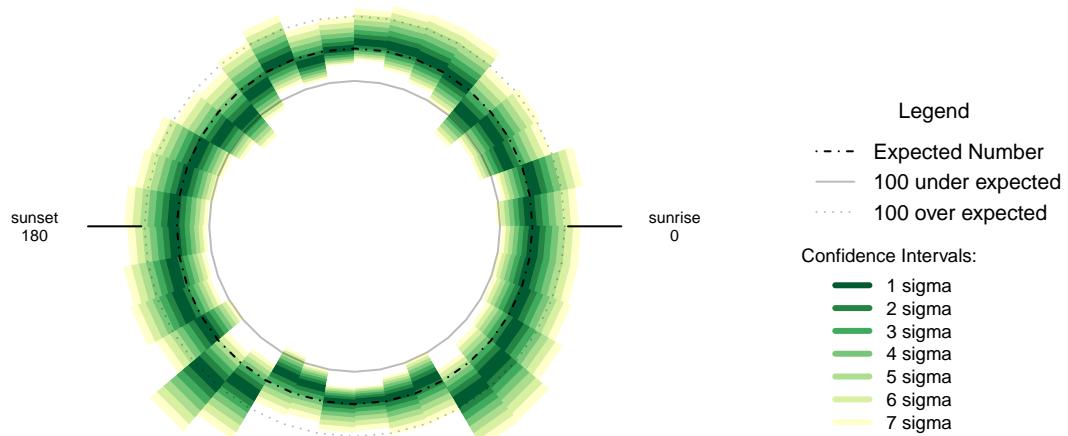


Figure 12.3: Over- and under-supply of earthquakes by angle of the sun (10 degree steps). Hawaii. n=10181

## 12.1 Formal Statement

Earthquakes in the region of Hawaii show a similar pattern to New Zealand, displaying an oversupply of earthquakes at night that is not the result of chance. The magnitude pattern of the oversupply is similar to New Zealand's pattern, and the pattern with respect to the position of the sun is similar to that of New Zealand. Because of the smaller sample size, Hawaii can be described as not inconsistent with New Zealand, Japan, Continental U.S.A., or Italy

## 12.2 Links

1 - USGS: <https://earthquake.usgs.gov>

## 12.3 Chapter Code

```
## ----setup, include=FALSE-----
knitr::opts_chunk$set(echo = FALSE)

## ----c10_001, warnings=FALSE, errors=FALSE, message=FALSE-----
library(geosphere)
library(lubridate, quietly=TRUE)
library(dplyr)
library(binom)
library(ggplot2)
library(maps)
library(mapdata)
library(parallel)
library(readr)
library(plotrix)
library(tidyr)
library(mapproj)
Sys.setenv(TZ = "UTC")
```

```

## ----warnings=FALSE, errors=FALSE, message=FALSE-----

if(!dir.exists("../othereqdata")){
  dir.create("../othereqdata")
}
if(!file.exists("../othereqdata/eq_hawaii_raw.RData")){
  bit1 <- "https://earthquake.usgs.gov/fdsnws/event/1/query.csv?"
  bit2 <- "starttime=2011-09-01%2000:00:00&endtime=2016-09-01%2000:00:00&"
  bit3 <- "maxlatitude=31.436&minlatitude=13.334&maxlongitude=213.882&"
  bit4 <- "minlongitude=177.407&minmagnitude=0&orderby=time"
  webaddress <- paste0(bit1, bit2, bit3, bit4)
  eqhawaii <- read.csv(webaddress, stringsAsFactors = FALSE)

  eqhawaii$time_UTC <- as.POSIXct(gsub("\\\\..+", "", as.character(eqhawaii$time)),
                                    format="%Y-%m-%dT%H:%M:%S", tz="UTC")
  eq_national <- eqhawaii %>%
    filter(depth > 0 & mag >= 0 &
           time_UTC >= as.POSIXct("2011-09-01T00:00:00",
                                    format="%Y-%m-%dT%H:%M:%S", tz="UTC") &
           time_UTC < as.POSIXct("2016-09-01T00:00:00",
                                    format="%Y-%m-%dT%H:%M:%S", tz="UTC")) %>%
    distinct() %>% arrange(time_UTC)

  rm(eqhawaii)
  names(eq_national)[5] <- "magnitude"
  save(eq_national, file="../othereqdata/eq_hawaii_raw.RData")
}

## ----

if(!file.exists("../othereqdata/eq_hawaii_processed.RData")){
  load("../othereqdata/eq_hawaii_raw.RData")
  southmost <- min(eq_national$latitude)
  westmost <- min(eq_national$longitude)

  eq_national <- eq_national %>% filter(
    magnitude > 0, depth > 0) %>% rowwise() %>% mutate(
    eq_gridpoint_y = round(
      distVincentyEllipsoid(c(longitude, southmost), c(longitude, latitude)) /50000,0),
    eq_gridpoint_x = round(
      distVincentyEllipsoid(c(westmost, latitude), c(longitude, latitude)) /50000,0),
    eq_roundedlat = destPoint(
      p=c(longitude, southmost), b=0, d=eq_gridpoint_y*50000)[2],
    eq_roundedlong = destPoint(
      p=c(westmost, eq_roundedlat), b=90, d=eq_gridpoint_x*50000)[1]) %>% ungroup()
  # use maptools to calculate solar angles
  sun_angles <- solarpos(
    matrix(c(eq_national$longitude, eq_national$latitude), ncol=2), eq_national$time_UTC)
  colnames(sun_angles) <- c("eq_compass", "eq_vertical")
  eq_national <- cbind(eq_national, sun_angles)
}

```

```

eq_national$eq_is_night <- eq_national$eq_vertical < 0
# calculate 360 degree as well as vertical
eq_national <- eq_national %>%
  mutate(eq_angle_360 = eq_vertical,
        eq_angle_360 = ifelse(eq_compass > 180, 180 - eq_angle_360, eq_angle_360),
        eq_angle_360 = ifelse(eq_vertical < 0 & eq_compass <= 180,
                              360 + eq_angle_360, eq_angle_360),
        eq_angle_by_10 = floor(eq_angle_360 /10) * 10)
save(eq_national, file="../othereqdata/eq_hawaii_processed.RData")
}

## -----
if(!file.exists("../othereqdata/eq_hawaii_expected.RData")){
  load("../othereqdata/eq_hawaii_processed.RData")

  lat_range <- unique(eq_national$eq_roundedlat)
  long_med <- median(eq_national$eq_roundedlong)

  # 1 minute intervals for a full solar year
  time1 <- ymd_hms("2015-01-01 00:00:00")
  time2 <- ymd_hms("2015-12-31 23:59:00")
  time_sq <- seq.POSIXt(from=time1, to=time2, by="min")

  calc_angs <- function(x, longinput, timeinput){
    library(dplyr)
    sun_angles <- maptools::solarpos(matrix(c(longinput, x), ncol=2), timeinput)
    colnames(sun_angles) <- c("eq_compass", "eq_vertical")
    # calculate 360 degree as well as vertical
    site_summary <- as.data.frame(sun_angles) %>%
      mutate(eq_angle_360 = eq_vertical,
            eq_angle_360 = ifelse(eq_compass > 180, 180 - eq_angle_360, eq_angle_360),
            eq_angle_360 = ifelse(
              eq_vertical < 0 & eq_compass <= 180, 360 + eq_angle_360, eq_angle_360),
            eq_angle_by_10 = floor(eq_angle_360 /10) * 10) %>%
      group_by(eq_angle_by_10) %>% summarise(total= n())
    site_summary$lat <- x
    return(site_summary)
  }
  ###
  # Calculate the number of cores
  no_cores <- detectCores() - 1
  # Initiate cluster
  cl <- makeCluster(no_cores)
  clusterExport(cl, varlist=c("lat_range", "long_med", "time_sq", "calc_angs"))
  list_angs <- parLapply(cl, lat_range,
                        function(x){
                          calc_angs(x=x, longinput=long_med, timeinput=time_sq)})
  stopCluster(cl)
  ###
  library(tidyr)
  anglong <- bind_rows(list_angs)
  angwide <- spread(anglong, key=eq_angle_by_10,value=total)
}

```

```

rm(anglong, list_angs, time_sq)
save(angwide, file="../othereqdata/eq_hawaii_expected.RData")
}

## -----
load("../othereqdata/eq_hawaii_processed.RData")
load("../othereqdata/eq_hawaii_expected.RData")
eq_night = sum(eq_national$eq_is_night)
eq_total = nrow(eq_national)
bands <- rev(c('#ffffcc', '#d9f0a3', '#add8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
sigmas <- c(0.682689492137086,
0.954499736103642,
0.997300203936740,
0.999936657516334,
0.999999426696856,
0.99999998026825,
0.999999999997440)
lbls <- c(
"1 sigma", "2 sigma",
"3 sigma", "4 sigma",
"5 sigma", "6 sigma",
"7 sigma")
typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
old_par=par()

## -----
bt <- binom.test(eq_night ,eq_total, conf.level= .999999999997440)

## -----
feature <- c("Earliest (UTC)", "Latest (UTC)",
"Northernmost", "Southernmost",
"Westmost", "Eastmost",
"Percent < Mag 3", "total entries",
"nighttime quakes")
value <- c(as.character(min(eq_national$time_UTC)),
as.character(max(eq_national$time_UTC)),
as.character(max(eq_national$latitude)),
as.character(min(eq_national$latitude)),
as.character(max(eq_national$longitude)),
as.character(max(eq_national$longitude[eq_national$longitude<0])),
as.character(round(100*sum(eq_national$magnitude < 3)/eq_total,2)),
as.character(eq_total),
as.character(eq_night))
data.frame(feature,value) %>% knitr::kable(caption="Data description")

## ---- fig.cap="Proportion of earthquakes at night: Hawaii. n=10181"-----
### making the basic proportion graph
eq_night = sum(eq_national$eq_is_night)
eq_total = nrow(eq_national)
bands <- rev(c('#ffffcc', '#d9f0a3', '#add8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
sigmas <- c(0.682689492137086,
0.954499736103642,

```

```

0.997300203936740,
0.999936657516334,
0.999999426696856,
0.999999998026825,
0.99999999997440)

lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")

typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
old_par=par()
conf_steps <- function(x, sigmas=sigmas, night=eq_night, total=eq_total){
  ci_lower <- binom.confint(night, total, method=c("wilson"), conf.level = sigmas[x])[1,5]
  ci_upper <- binom.confint(night, total, method=c("wilson"), conf.level = sigmas[x])[1,6]
  ci_data <- data.frame(step = x, ci_lower, ci_upper)
}

ci_spacing <- lapply(7:1, conf_steps, sigmas=sigmas, night=eq_night, total=eq_total)
ci_steps <- bind_rows(ci_spacing)

layout(matrix(c(1,1,1,2), ncol=4))

par(mar=c(5,6,4,2))
plot(c(min(0.5,floor(100*ci_steps[1,2])/100), max(0.5,ceiling(100*ci_steps[1,3])/100)),
     y=c(-3,8), type="n", bty="n", yaxt="n", ylab="",
     xlab="Proportion of earthquakes at night")
a <- a <- apply(ci_steps, 1, function(x){
  polygon(c(x[2], x[3], x[3], x[2]), c(0, 0, 1, 1), col=bands[x[1]], border=NA)})
lines(c(.5,.5), c(0,1), col="#FFFFFF")
lines(c(.5,.5), c(0,1), lty=2, col="#777777")
lines(c(eq_night/eq_total,eq_night/eq_total), c(0,1), lwd=2)

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE)
legend(0,5.5, legend=lbls, lty=typs, lwd=weights, col=bands, bty="n", xjust=0,
       title="Confidence Intervals:", y.intersp=1.1, cex=0.9)

lbls2=c("50% Night", "Actual Proportion")
typs2=c(2,1)
weights2=c(1,2)
cls2=c("#777777", "#000000")
legend(0,7, legend=lbls2, lty=typs2, lwd=weights2, col=cls2, bty="n", xjust=0,
       title="Legend", y.intersp=1.2)

par(mar=old_par$mar)
par(mfrow=c(1,1))

## ---- fig.cap="Proportion of night earthquakes by magnitude, Hawaii. n=10181"----

```

```

old_par=par()
grf <- eq_national %>% mutate(floored_mag = floor(magnitude*2)/2) %>%
  group_by(floored_mag) %>% summarise(successes = sum(eq_is_night), trials=n())

poly_conf_int <- function(success, trials, aa, stepsize, sigma, colr){
  ci <- binom.confint(success, trials, method=c("wilson"), conf.level = sigma)
  lower <- ci[1,5]
  upper <- ci[1,6]
  a <- polygon(x=c(aa,aa+stepsize,aa+stepsize,aa), y=c(upper,upper,lower,lower),
                col=colr, border=NA)
}

plot7sig <- function(success, trials, aa, stepsize){
  library(binom)
  #bands <- c('#fffffb2', '#fed976', '#feb24c', '#fd8d3c', '#fc4e2a', '#e31a1c', '#b10026')
  bands <- rev(c('#ffffcc', '#d9f0a3', '#add8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
  sigmas <- c(0.682689492137086,
             0.954499736103642,
             0.997300203936740,
             0.999936657516334,
             0.999999426696856,
             0.999999998026825,
             0.99999999997440)
  sapply(7:1, function(x){
    poly_conf_int(success, trials, aa, stepsize, sigmas[x], bands[x]))}
  a <- lines(c(aa, aa + stepsize), c(success/trials, success/trials), lwd=2)
}

lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
tys <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
clrs <- rev(c('#ffffcc', '#d9f0a3', '#add8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
#clrs <- c('#fffffb2', '#fed976', '#feb24c', '#fd8d3c', '#fc4e2a', '#e31a1c', '#b10026')

layout(matrix(c(1,1,1,2), ncol=4))
plot(x=c(0,max(grf$floored_mag)+0.5), y=c(0,1), type="n", bty="n",
      xlab="Magnitude (0.5 steps)", ylab="Proportion of earthquakes at night")
a <- apply(grf,1,function(x){plot7sig(x[2],x[3],x[1],0.5)})
lines(c(0,10), c(.5,.5), col="#FFFFFF")
lines(c(0,10), c(.5,.5), lty=2, col="#777777")

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,1), type="n", bty="n", axes=FALSE)
legend(0.5, legend=lbls, lty=tys, lwd=weights, col=clrs, bty="n", xjust=0,
       title="Confidence
Intervals:", cex=0.9)
lbls=c("Expected Proportion", "Actual Proportion")

```

```

typs=c(2,1)
weights=c(1,2)
legend(0,7, legend=lbls, lty=typs, lwd=weights, bty="n", xjust=0,
       title="Legend", y.intersp=1.2)

par(mar=old_par$mar)
par(mfrow=c(1,1))

## -----
by_angle <- eq_national %>%
  group_by(eq_angle_by_10) %>% summarise(total= n()) %>%
  mutate(daynight=ifelse(eq_angle_by_10 < 180, "day", "night"))

merged <- merge(eq_national, angwide, by.x="eq_roundedlat", by.y="lat")
agg_expected <- merged %>% select(`0`:`350`) %>% colSums(na.rm=TRUE)
expected_prop <- agg_expected / sum(agg_expected)
expected <- data.frame(eq_angle_by_10 = as.numeric(names(expected_prop)),
                       expected_prop = as.numeric(expected_prop))
expected$expected_number = expected_prop * eq_total
act_exp <- merge(expected, by_angle, by="eq_angle_by_10", all.x=TRUE)

act_exp$total[is.na(act_exp$total)] <- 0
act_exp$daynight <- NULL
act_exp$act_prop <- act_exp$total / sum(act_exp$total)

ci_brackets <- act_exp %>% ungroup() %>% mutate(grand_total=sum(total)) %>%
  rowwise() %>% mutate(
    ci_lower_7 = binom.confint(total, grand_total, method=c("wilson"),
                               conf.level = sigmas[7])[1,5] * grand_total,
    ci_upper_7 = binom.confint(total, grand_total, method=c("wilson"),
                               conf.level = sigmas[7])[1,6] * grand_total,
    ci_lower_6 = binom.confint(total, grand_total, method=c("wilson"),
                               conf.level = sigmas[6])[1,5] * grand_total,
    ci_upper_6 = binom.confint(total, grand_total, method=c("wilson"),
                               conf.level = sigmas[6])[1,6] * grand_total,
    ci_lower_5 = binom.confint(total, grand_total, method=c("wilson"),
                               conf.level = sigmas[5])[1,5] * grand_total,
    ci_upper_5 = binom.confint(total, grand_total, method=c("wilson"),
                               conf.level = sigmas[5])[1,6] * grand_total,
    ci_lower_4 = binom.confint(total, grand_total, method=c("wilson"),
                               conf.level = sigmas[4])[1,5] * grand_total,
    ci_upper_4 = binom.confint(total, grand_total, method=c("wilson"),
                               conf.level = sigmas[4])[1,6] * grand_total,
    ci_lower_3 = binom.confint(total, grand_total, method=c("wilson"),
                               conf.level = sigmas[3])[1,5] * grand_total,
    ci_upper_3 = binom.confint(total, grand_total, method=c("wilson"),
                               conf.level = sigmas[3])[1,6] * grand_total,
    ci_lower_2 = binom.confint(total, grand_total, method=c("wilson"),
                               conf.level = sigmas[2])[1,5] * grand_total,
    ci_upper_2 = binom.confint(total, grand_total, method=c("wilson"),

```

```

                conf.level = sigmas[2])[1,6] * grand_total,
ci_lower_1 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[1])[1,5] * grand_total,
ci_upper_1 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[1])[1,6] * grand_total)

norm_ci <- ci_brackets

for (i in c(4,7:20)){
norm_ci[,i] <- ci_brackets[,i] - ci_brackets[,3]
}
circlesize=100

## ---- fig.cap="Over- and under- supply of earthquakes by angle of the sun
## (10 degree steps). Hawaii. n=10181"----
norm_ci$border = 2
# need to double entries with a displacement of 10
# to make each side of the item on the graph
norm_ci2 <- norm_ci
norm_ci2$eq_angle_by_10 <- norm_ci2$eq_angle_by_10 + 10
norm_ci2$border = 1
graphdata <- bind_rows(norm_ci,norm_ci2) %>% arrange(eq_angle_by_10,border)

##### plot graph
bands <- rev(c('#ffffcc','#d9f0a3','#addd8e','#78c679','#41ab5d','#238443','#005a32'))
old_par=par()
layout(matrix(c(1,1,1,2), ncol=4))

# overall limits
limits=2 * max(abs(c(graphdata$ci_lower_7, graphdata$ci_upper_7)))
# plot upper confidence 7 interval using plotrix
polar.plot(graphdata$ci_upper_7, polar.pos=graphdata$eq_angle_by_10,
           radial.lim=c(-1*limits,limits),
           labels = "", main=NULL,lwd=0.5, rp.type="p",
           show.grid.labels=FALSE, show.grid=FALSE, mar=c(0,0,0,0),
           grid.col=bands[7], line.col=bands[7], poly.col=bands[7])
# plot upper 6 confidence interval
plot_ci_round <- function(upper_bound,x){
  polar.plot(upper_bound, polar.pos=graphdata$eq_angle_by_10, add=TRUE,
             radial.lim=c(-1*limits,limits),
             line.col=bands[x], lwd=0.5, rp.type="p", poly.col=bands[x])
}

plot_ci_round(graphdata$ci_upper_6, 6)
plot_ci_round(graphdata$ci_upper_5, 5)
plot_ci_round(graphdata$ci_upper_4, 4)
plot_ci_round(graphdata$ci_upper_3, 3)
plot_ci_round(graphdata$ci_upper_2, 2)
plot_ci_round(graphdata$ci_upper_1, 1)
plot_ci_round(graphdata$ci_lower_1, 2)
plot_ci_round(graphdata$ci_lower_2, 3)
plot_ci_round(graphdata$ci_lower_3, 4)
plot_ci_round(graphdata$ci_lower_4, 5)
plot_ci_round(graphdata$ci_lower_5, 6)

```

```

plot_ci_round(graphdata$ci_lower_6, 7)
polar.plot(graphdata$ci_lower_7, polar.pos=graphdata$eq_angle_by_10,
            add=TRUE, radial.lim=c(-1*limits,limits),
            line.col="white", lwd=0.5, rp.type="p", poly.col="white")

# plot expected guide line
polar.plot(rep(0,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10, add=TRUE,
            radial.lim=c(-1*limits,limits),
            rp.type="p", lty=4)
# plot 500 less than expected guide line
polar.plot(rep(-1 * circlesize,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10,
            add=TRUE,radial.lim=c(-1*limits,limits),
            rp.type="p", lty=1, line.col="#00000044")
# plot 500 more than expected guide line
polar.plot(rep(circlesize,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10,
            add=TRUE,radial.lim=c(-1*limits,limits),
            rp.type="p", lty=3, line.col="#00000044")

lines(c(-1.5,-1.2)*limits, c(0,0))
lines(c(1.5,1.2)*limits, c(0,0))
text(-1.8*limits,0, label="sunset
180", cex=0.7)
text(1.8*limits,0, label="sunrise
0", cex=0.7)

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE, xlab="")
lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
clrs <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
legend(0,4.5, legend=lbls, lty=typs, lwd=weights, col=clrs, bty="n", xjust=0,
       title="Confidence Intervals:", cex=0.9)

lbls2=c("Expected Number", paste(circlesize,"under expected"),
        paste(circlesize,"over expected"))
typs2=c(4,1,3)
weights2=c(1,1,1)
clrs2=c("#000000", "#00000044", "#00000044")
legend(0,10, legend=lbls2, lty=typs2, lwd=weights2, bty="n", xjust=0,
       title="Legend", y.intersp=1.2, col=clrs2)

par(mar=old_par$mar)
par(mfrow=c(1,1))

```



# Chapter 13

## The Philippines is similar

Philippines earthquakes are published by PHIVOLCS, and the web pages can be read directly into R. The approach used to gather this data automatically followed links on the page to capture the relevant information.

For the same period as the New Zealand Data (from September 2011 to September 2016), in earthquakes data from the Philippines there are 8054 events of depth greater than 0 and magnitude greater than 0.

Of the 8054 in the data, 4959 occurred at night, a proportion of 0.6157. A seven sigma confidence interval for the proportion of earthquakes occurring at night is 0.5773 to 0.6532, so I can be confident this is a non-random difference.

The Philippines has the highest rate of nighttime earthquakes in a data set analysed in this study, so is clearly not 50% despite the small sample size.

Unusually, while the earthquakes from the Philippines show high night proportions at low magnitudes like other data sets, that rate does not unequivocally drop before the sample sizes shrink and the confidence intervals expand. While the Philippines has many fewer small earthquakes than previous data sets, the high night proportions effect extends to far higher magnitudes than in other regions data sets.

Night earthquakes show a general discontinuity to day, with a raised incidence (relative to day) of earthquakes through the entire night. There is still a peak at night when the sun is in the east, but not until 60 degrees below the horizon.

Table 13.1: Data description

feature	value
Earliest (UTC)	2011-09-01
Latest (UTC)	2016-08-29 13:30:00
Northernmost	21.47
Southernmost	2.2
Westmost	116.99
Eastmost	128.68
Percent < Mag 3	40.81
total entries	8054
nighttime quakes	4959

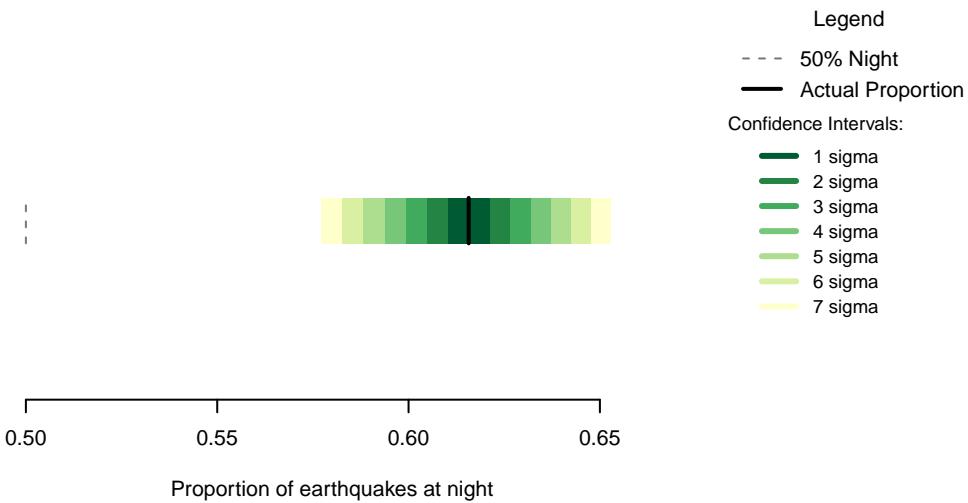


Figure 13.1: Proportion of earthquakes at night: Philippines. n=8054

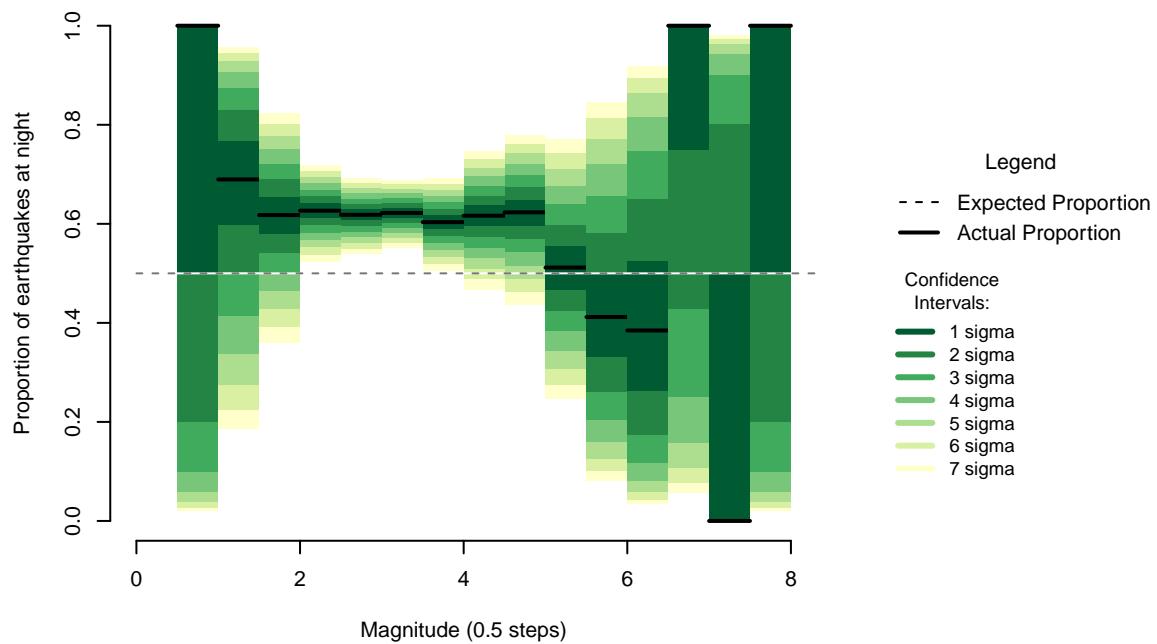


Figure 13.2: Proportion of night earthquakes by magnitude, Philippines. n=8054

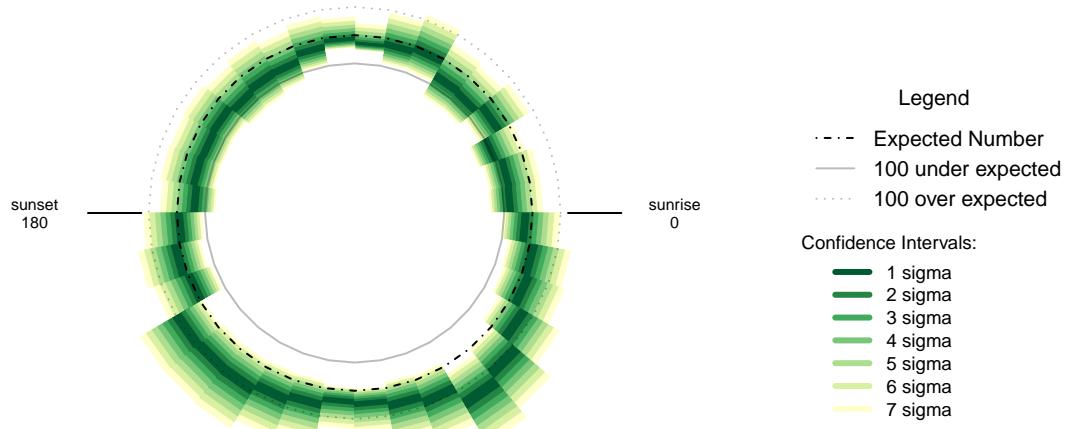


Figure 13.3: Over- and under-supply of earthquakes by angle of the sun (10 degree steps). , Philippines.  
n=8054

### 13.1 Formal Statement

Earthquakes in the Philippines show a similar pattern to New Zealand, displaying an oversupply of earthquakes at night that is not the result of chance. Despite a lack of small earthquakes, there is a clear bias towards night among small to medium magnitude earthquakes.

### 13.2 Links

1 - PHIVOLCS: [http://www.phivolcs.dost.gov.ph/html/update\\_SOEPD/EQLatest.htm](http://www.phivolcs.dost.gov.ph/html/update_SOEPD/EQLatest.htm)

### 13.3 Chapter Code

```
## ----setup, include=FALSE-----
knitr::opts_chunk$set(echo = FALSE)

## ----c10_001, warnings=FALSE, errors=FALSE, message=FALSE-----
library(geosphere)
library(lubridate, quietly=TRUE)
library(dplyr)
library(binom)
library(ggplot2)
library(maps)
library(mapdata)
library(parallel)
library(readr)
library(plotrix)
library(tidyr)
library(mapproj)
library(rvest)
Sys.setenv(TZ = "UTC")
```

```

## ----warnings=FALSE, errors=FALSE, message=FALSE-----

if(!dir.exists("../othereqdata")){
  dir.create("../othereqdata")
}
if(!file.exists("../othereqdata/eq_phippines_raw.RData")){

  pg <- read_html("http://www.phivolcs.dost.gov.ph/html/update_SOEPD/EQLatest.html")
  lnks <- pg %>% html_nodes("a") %>% html_attr("href") %>%
    grep(pattern = "(EQLatest-Monthly/)(2016|2015|2014|2013|2012|2011)", value=TRUE)
  lnks <- paste("http://www.phivolcs.dost.gov.ph/html/update_SOEPD/", lnks, sep="")

  assemble_eq <- function(x){
    h <- html_text(read_html(x))
    obs <- unlist(strsplit(gsub("(0-9)+ ([Jan|Feb|Mar|Apr|May|Jun|Jul|Aug|Sep|Oct|Nov|Dec)]",
      "~-\\1 \\2",h), "~"))
    obs <- obs[2:length(obs)]
    obs <- gsub("(AM|PM)", "\\1~",obs)

    obs <- gsub("_+", "", obs)
    obs <- gsub("[[:space:]]+", " ", obs)
    obs <- gsub("~ ~", "~",obs)
    obsdf <- data.frame(obs) %>%
      separate(obs, into=c("plpdate", "everything_else"), sep="~", extra="merge") %>%
      separate(everything_else,
        into=c("latitude", "longitude", "depth", "magnitude", "location"),
        sep=" ", extra="merge") %>%
      mutate(time_UTC = as.POSIXct(plpdate, format="%d %B %Y - %I:%M %p", tz="UTC") -
        hours(8),
        latitude = as.numeric(latitude),
        longitude = as.numeric(longitude),
        depth = as.numeric(depth),
        magnitude = as.numeric(magnitude))
    obsdf <- obsdf[!is.na(obsdf$latitude),]
    return(obsdf)
  }

  eq_list <- lapply(lnks, assemble_eq)
  eq_phippines <- bind_rows(eq_list)
  eq_national <- eq_phippines %>%
    filter(depth > 0 & magnitude >= 0 &
      time_UTC >=
        as.POSIXct("2011-09-01T00:00:00", format="%Y-%m-%dT%H:%M:%S", tz="UTC") &
        time_UTC < as.POSIXct("2016-09-01T00:00:00",
          format="%Y-%m-%dT%H:%M:%S", tz="UTC")) %>%
    distinct() %>% arrange(time_UTC)
  #one longitude was missing the decimal sign
  eq_phippines$longitude[eq_phippines$longitude > 9000] <-
    eq_phippines$longitude[eq_phippines$longitude > 9000] /100
  rm(eq_phippines)
  save(eq_national, file="..../othereqdata/eq_phippines_raw.RData")
}

```

```

}

## ----

if(!file.exists("../othereqdata/eq_phippines_processed.RData")){
load("../othereqdata/eq_phippines_raw.RData")
southmost <- min(eq_national$latitude)
westmost <- min(eq_national$longitude)

eq_national <- eq_national %>% filter(
  magnitude > 0, depth > 0) %>% rowwise() %>% mutate(
  eq_gridpoint_y = round(
    distVincentyEllipsoid(c(longitude, southmost), c(longitude, latitude)) /50000,0),
  eq_gridpoint_x = round(
    distVincentyEllipsoid(c(westmost, latitude), c(longitude, latitude)) /50000,0),
  eq_roundedlat = destPoint(
    p=c(longitude, southmost), b=0, d=eq_gridpoint_y*50000)[2],
  eq_roundedlong = destPoint(
    p=c(westmost, eq_roundedlat), b=90, d=eq_gridpoint_x*50000)[1]) %>% ungroup()
# use maptools to calculate solar angles
sun_angles <- solarpes(
  matrix(c(eq_national$longitude, eq_national$latitude), ncol=2), eq_national$time_UTC)
colnames(sun_angles) <- c("eq_compass", "eq_vertical")
eq_national <- cbind(eq_national, sun_angles)
eq_national$eq_is_night <- eq_national$eq_vertical < 0
# calculate 360 degree as well as vertical
eq_national <- eq_national %>%
  mutate(eq_angle_360 = eq_vertical,
        eq_angle_360 = ifelse(eq_compass > 180, 180 - eq_angle_360, eq_angle_360),
        eq_angle_360 = ifelse(eq_vertical < 0 & eq_compass <= 180,
                              360 + eq_angle_360, eq_angle_360),
        eq_angle_by_10 = floor(eq_angle_360 /10) * 10)
save(eq_national, file="../othereqdata/eq_phippines_processed.RData")
}

## ----

if(!file.exists("../othereqdata/eq_phippines_expected.RData")){
load("../othereqdata/eq_phippines_processed.RData")

lat_range <- unique(eq_national$eq_roundedlat)
long_med <- median(eq_national$eq_roundedlong)

# 1 minute intervals for a full solar year
time1 <- ymd_hms("2015-01-01 00:00:00")
time2 <- ymd_hms("2015-12-31 23:59:00")
time_sq <- seq.POSIXt(from=time1, to=time2, by="min")

calc_angs <- function(x, longinput, timeinput){
  library(dplyr)
  sun_angles <- maptools::solarpos(matrix(c(longinput, x), ncol=2), timeinput)
  colnames(sun_angles) <- c("eq_compass", "eq_vertical")
}

```

```

# calculate 360 degree as well as vertical
site_summary <- as.data.frame(sun_angles) %>%
  mutate(eq_angle_360 = eq_vertical,
        eq_angle_360 = ifelse(eq_compass > 180, 180 - eq_angle_360, eq_angle_360),
        eq_angle_360 = ifelse(
          eq_vertical < 0 & eq_compass <= 180, 360 + eq_angle_360, eq_angle_360),
        eq_angle_by_10 = floor(eq_angle_360 /10) * 10) %>%
  group_by(eq_angle_by_10) %>% summarise(total= n())
  site_summary$lat <- x
  return(site_summary)
}

#####
# Calculate the number of cores
no_cores <- detectCores() - 1
# Initiate cluster
cl <- makeCluster(no_cores)
clusterExport(cl, varlist=c("lat_range", "long_med", "time_sq", "calc_angs"))
list_angs <- parLapply(cl, lat_range,
                       function(x){
                         calc_angs(x=x, longinput=long_med, timeinput=time_sq)})
stopCluster(cl)
#####

library(tidyr)
anglong <- bind_rows(list_angs)
angwide <- spread(anglong, key=eq_angle_by_10,value=total)
rm(anglong, list_angs, time_sq)
save(angwide, file="../othereqdata/eq_philippines_expected.RData")
}

## -----
load("../othereqdata/eq_philippines_processed.RData")
load("../othereqdata/eq_philippines_expected.RData")
eq_night = sum(eq_national$eq_is_night)
eq_total = nrow(eq_national)
bands <- rev(c('#ffffcc', '#d9f0a3', '#add8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
sigmas <- c(0.682689492137086,
0.954499736103642,
0.997300203936740,
0.999936657516334,
0.999999426696856,
0.99999998026825,
0.999999999997440)
lbls <- c(
"1 sigma", "2 sigma",
"3 sigma", "4 sigma",
"5 sigma", "6 sigma",
"7 sigma")
typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
old_par=par()

## -----
bt <- binom.test(eq_night ,eq_total, conf.level= .999999999997440)

```

```

## -----
feature <- c("Earliest (UTC)", "Latest (UTC)",
           "Northernmost", "Southernmost",
           "Westmost", "Eastmost",
           "Percent < Mag 3", "total entries",
           "nighttime quakes")
value <- c(as.character(min(eq_national$time_UTC)),
           as.character(max(eq_national$time_UTC)),
           as.character(max(eq_national$latitude)),
           as.character(min(eq_national$latitude)),
           as.character(min(eq_national$longitude)),
           as.character(max(eq_national$longitude)),
           as.character(round(100*sum(eq_national$magnitude < 3)/eq_total,2)),
           as.character(eq_total),
           as.character(eq_night))
data.frame(feature,value) %>% knitr::kable(caption="Data description")

## ---- fig.cap="Proportion of earthquakes at night: Philippines. n=8054"----
### making the basic proportion graph
eq_night = sum(eq_national$eq_is_night)
eq_total = nrow(eq_national)
bands <- rev(c('#ffffcc','#d9f0a3','#addd8e','#78c679','#41ab5d','#238443','#005a32'))
sigmas <- c(0.682689492137086,
           0.954499736103642,
           0.997300203936740,
           0.999936657516334,
           0.999999426696856,
           0.99999998026825,
           0.999999999997440)
lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
old_par=par()
conf_steps <- function(x, sigmas=sigmas, night=eq_night, total=eq_total){
  ci_lower <- binom.confint(night, total, method=c("wilson"), conf.level = sigmas[x])[1,5]
  ci_upper <- binom.confint(night, total, method=c("wilson"), conf.level = sigmas[x])[1,6]
  ci_data <- data.frame(step = x, ci_lower, ci_upper)
}
ci_spacing <- lapply(7:1, conf_steps, sigmas=sigmas, night=eq_night, total=eq_total)
ci_steps <- bind_rows(ci_spacing)

layout(matrix(c(1,1,1,2), ncol=4))

par(mar=c(5,6,4,2))
plot(c(min(0.5,floor(100*ci_steps[1,2])/100), max(0.5,ceiling(100*ci_steps[1,3])/100)),
     y=c(-3,8), type="n", bty="n", yaxt="n", ylab="",
     xlab="Proportion of earthquakes at night")
a <- a <- apply(ci_steps, 1, function(x){

```

```

polygon(c(x[2], x[3], x[3], x[2]), c(0, 0, 1, 1), col=bands[x[1]], border=NA))
lines(c(.5,.5), c(0,1), col="#FFFFFF")
lines(c(.5,.5), c(0,1), lty=2, col="#777777")
lines(c(eq_night/eq_total,eq_night/eq_total), c(0,1), lwd=2)

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE)
legend(0,5.5, legend=lbls, lty=typs, lwd=weights, col=bands, bty="n", xjust=0,
       title="Confidence Intervals:", y.intersp=1.1, cex=0.9)

lbls2=c("50% Night", "Actual Proportion")
typs2=c(2,1)
weights2=c(1,2)
cls2=c("#777777", "#000000")
legend(0,7, legend=lbls2, lty=typs2, lwd=weights2, col=cls2, bty="n", xjust=0,
       title="Legend", y.intersp=1.2)

par(mar=old_par$mar)
par(mfrow=c(1,1))

## ---- fig.cap="Proportion of night earthquakes by magnitude, Philippines. n=8054"----
old_par=par()
grf <- eq_national %>% mutate(floored_mag = floor(magnitude*2)/2) %>%
  group_by(floored_mag) %>% summarise(successes = sum(eq_is_night), trials=n())

poly_conf_int <- function(success, trials, aa, stepsize, sigma, colr){
  ci <- binom.confint(success, trials, method=c("wilson"), conf.level = sigma)
  lower <- ci[1,5]
  upper <- ci[1,6]
  a <- polygon(x=c(aa,aa+stepsize,aa+stepsize,aa), y=c(upper,upper,lower,lower),
                col=colr, border=NA)
}

plot7sig <- function(success, trials, aa, stepsize){
  library(binom)
  #bands <- c('#fffffb2', '#fed976', '#feb24c', '#fd8d3c', '#fc4e2a', '#e31a1c', '#b10026')
  bands <- rev(c('#ffffcc', '#d9f0a3', '#add8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
  sigmas <- c(0.682689492137086,
             0.954499736103642,
             0.997300203936740,
             0.999936657516334,
             0.999999426696856,
             0.999999998026825,
             0.99999999997440)
  sapply(7:1, function(x){
    poly_conf_int(success, trials, aa, stepsize, sigmas[x], bands[x]))}
  a <- lines(c(aa, aa + stepsize), c(success/trials, success/trials), lwd=2)
}

```

```

lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
typs <- c(1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3)
clrs <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
#clrs <- c('#ffffb2', '#fed976', '#feb24c', '#fd8d3c', '#fc4e2a', '#e31a1c', '#b10026')

layout(matrix(c(1,1,1,2), ncol=4))
plot(x=c(0,max(grf$floored_mag)+0.5), y=c(0,1), type="n", bty="n",
      xlab="Magnitude (0.5 steps)", ylab="Proportion of earthquakes at night")
a <- apply(grf, 1, function(x){plot7sig(x[2],x[3],x[1],0.5)})
lines(c(0,10), c(.5,.5), col="#FFFFFF")
lines(c(0,10), c(.5,.5), lty=2, col="#777777")

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE)
legend(0.5, legend=lbls, lty=typs, lwd=weights, col=clrs, bty="n", xjust=0,
       title="Confidence
Intervals:", cex=0.9)
lbls=c("Expected Proportion", "Actual Proportion")
typs=c(2,1)
weights=c(1,2)
legend(0.7, legend=lbls, lty=typs, lwd=weights, bty="n", xjust=0,
       title="Legend", y.intersp=1.2)

par(mar=old_par$mar)
par(mfrow=c(1,1))

## -----
by_angle <- eq_national %>%
  group_by(eq_angle_by_10) %>% summarise(total= n()) %>%
  mutate(daynight=ifelse(eq_angle_by_10 < 180, "day", "night"))

merged <- merge(eq_national, angwide, by.x="eq_roundedlat", by.y="lat")
agg_expected <- merged %>% select(`0`:`350`) %>% colSums(na.rm=TRUE)
expected_prop <- agg_expected / sum(agg_expected)
expected <- data.frame(eq_angle_by_10 = as.numeric(names(expected_prop)),
                       expected_prop = as.numeric(expected_prop))
expected$expected_number = expected_prop * eq_total
act_exp <- merge(expected, by_angle, by="eq_angle_by_10", all.x=TRUE)

act_exp$total[is.na(act_exp$total)] <- 0
act_exp$daynight <- NULL
act_exp$act_prop <- act_exp$total / sum(act_exp$total)

ci_brackets <- act_exp %>% ungroup() %>% mutate(grand_total=sum(total)) %>%
  rowwise() %>% mutate(

```

```

ci_lower_7 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[7])[1,5] * grand_total,
ci_upper_7 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[7])[1,6] * grand_total,
ci_lower_6 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[6])[1,5] * grand_total,
ci_upper_6 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[6])[1,6] * grand_total,
ci_lower_5 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[5])[1,5] * grand_total,
ci_upper_5 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[5])[1,6] * grand_total,
ci_lower_4 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[4])[1,5] * grand_total,
ci_upper_4 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[4])[1,6] * grand_total,
ci_lower_3 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[3])[1,5] * grand_total,
ci_upper_3 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[3])[1,6] * grand_total,
ci_lower_2 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[2])[1,5] * grand_total,
ci_upper_2 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[2])[1,6] * grand_total,
ci_lower_1 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[1])[1,5] * grand_total,
ci_upper_1 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[1])[1,6] * grand_total)

norm_ci <- ci_brackets

for (i in c(4,7:20)){
norm_ci[,i] <- ci_brackets[,i] - ci_brackets[,3]
}
circlesize=100

## ---- fig.cap="Over- and under- supply of earthquakes by angle of the sun
## (10 degree steps). , Philippines. n=8054"----
norm_ci$border = 2
# need to double entries with a displacement of 10
# to make each side of the item on the graph
norm_ci2 <- norm_ci
norm_ci2$eq_angle_by_10 <- norm_ci2$eq_angle_by_10 + 10
norm_ci2$border = 1
graphdata <- bind_rows(norm_ci,norm_ci2) %>% arrange(eq_angle_by_10,border)

##### plot graph
bands <- rev(c('#ffffcc','#d9f0a3','#add8e','#78c679','#41ab5d','#238443','#005a32'))
old_par=par()
layout(matrix(c(1,1,1,2), ncol=4))

# overall limits
limits=2 * max(abs(c(graphdata$ci_lower_7, graphdata$ci_upper_7)))

```

```

# plot upper confidence 7 interval using plotrix
polar.plot(graphdata$ci_upper_7, polar.pos=graphdata$eq_angle_by_10,
            radial.lim=c(-1*limits,limits),
            labels = "", main=NULL, lwd=0.5, rp.type="p",
            show.grid.labels=FALSE, show.grid=FALSE, mar=c(0,0,0,0),
            grid.col=bands[7], line.col=bands[7], poly.col=bands[7])
# plot upper 6 confidence interval
plot_ci_round <- function(upper_bound,x){
  polar.plot(upper_bound, polar.pos=graphdata$eq_angle_by_10, add=TRUE,
             radial.lim=c(-1*limits,limits),
             line.col=bands[x], lwd=0.5, rp.type="p", poly.col=bands[x])
}
plot_ci_round(graphdata$ci_upper_6, 6)
plot_ci_round(graphdata$ci_upper_5, 5)
plot_ci_round(graphdata$ci_upper_4, 4)
plot_ci_round(graphdata$ci_upper_3, 3)
plot_ci_round(graphdata$ci_upper_2, 2)
plot_ci_round(graphdata$ci_upper_1, 1)
plot_ci_round(graphdata$ci_lower_1, 2)
plot_ci_round(graphdata$ci_lower_2, 3)
plot_ci_round(graphdata$ci_lower_3, 4)
plot_ci_round(graphdata$ci_lower_4, 5)
plot_ci_round(graphdata$ci_lower_5, 6)
plot_ci_round(graphdata$ci_lower_6, 7)
polar.plot(graphdata$ci_lower_7, polar.pos=graphdata$eq_angle_by_10,
           add=TRUE, radial.lim=c(-1*limits,limits),
           line.col="white", lwd=0.5, rp.type="p", poly.col="white")

# plot expected guide line
polar.plot(rep(0,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10, add=TRUE,
            radial.lim=c(-1*limits,limits),
            rp.type="p", lty=4)
# plot 500 less than expected guide line
polar.plot(rep(-1 * circlesize,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10,
           add=TRUE,radial.lim=c(-1*limits,limits),
           rp.type="p", lty=1, line.col="#00000044")
# plot 500 more than expected guide line
polar.plot(rep(circlesize,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10,
           add=TRUE,radial.lim=c(-1*limits,limits),
           rp.type="p", lty=3, line.col="#00000044")

lines(c(-1.5,-1.2)*limits, c(0,0))
lines(c(1.5,1.2)*limits, c(0,0))
text(-1.8*limits,0, label="sunset
180", cex=0.7)
text(1.8*limits,0, label="sunrise
0", cex=0.7)

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE, xlab="")
lbls <- c(
  "1 sigma", "2 sigma",

```

```
"3 sigma", "4 sigma",
"5 sigma", "6 sigma",
"7 sigma")
typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
clrs <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
legend(0,4.5, legend=lbls, lty=typs, lwd=weights, col=clrs, bty="n", xjust=0,
      title="Confidence Intervals:", cex=0.9)

lbls2=c("Expected Number", paste(circlesize,"under expected"),
       paste(circlesize,"over expected"))
typs2=c(4,1,3)
weights2=c(1,1,1)
clrs2=c("#000000", "#00000044", "#00000044")
legend(0,10, legend=lbls2, lty=typs2, lwd=weights2, bty="n", xjust=0,
      title="Legend", y.intersp=1.2, col=clrs2)

par(mar=old_par$mar)
par(mfrow=c(1,1))
```

# Chapter 14

## Thailand is similar

Thai earthquake data is provided publicly by the Seismology Bureau of the Thai Meteorological Department. The process to capture the data was to use R to cycle through the webpage URLs recording the information on each page.

For the same period as the New Zealand Data (from September 2011 to September 2016), in earthquakes data from the Thailand there are 2771 events of depth greater than 0 and magnitude greater than 0.

Of the 2771 events in the data, 1576 occurred at night, a proportion of 0.5687. A seven sigma confidence interval for the proportion of earthquakes occurring at night would be 0.5022 to 0.6337, making the observed proportion clearly non-random.

The smaller sample size of the Thai set of data makes wide confidence intervals, but the overall rate is beyond 7 sigma from the expected 0.5 proportion of earthquakes at night

While there are not enough earthquakes to be certain, the magnitude pattern seems to be similar to the nearby (in global terms) Philippines earthquake data. In the range from magnitude 2 to magnitude 4, where the confidence intervals are smallest, the proportion of night earthquakes seems to be consistently high.

The trend by angle of the sun also seems similar to the Philippines. There is a generally raised level of earthquake risk at night, with some heightened risk when the sun is 50 to 60 degrees below the horizon. A noticeable decrease in risk occurs as the sun is 30 to 60 degrees above the horizon in the west of the sky.

Table 14.1: Data description

feature	value
Earliest (UTC)	2011-09-02 21:14:05
Latest (UTC)	2016-08-31 18:22:43
Northernmost	37.75
Southernmost	0.77
Westmost	90.1
Eastmost	108.72
Percent < Mag 3	53.19
total entries	2771
nighttime quakes	1576

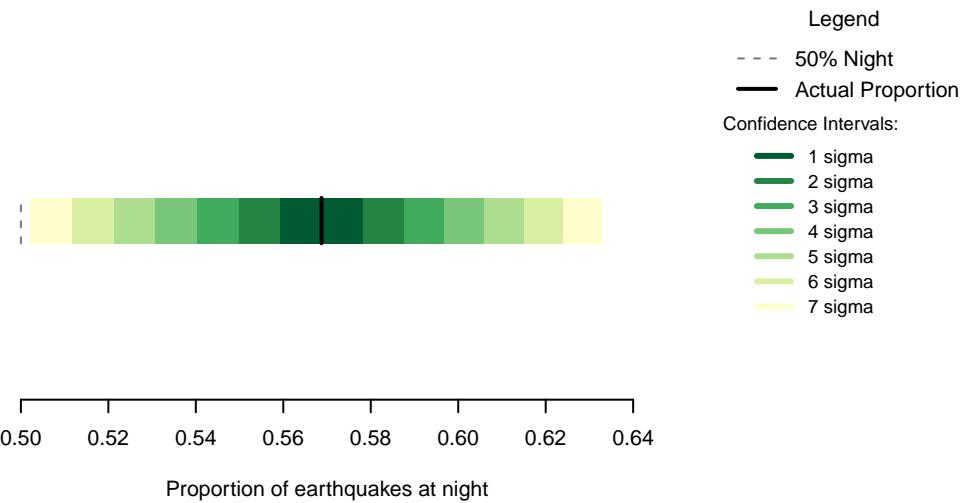


Figure 14.1: Proportion of earthquakes at night: Thailand n=2771

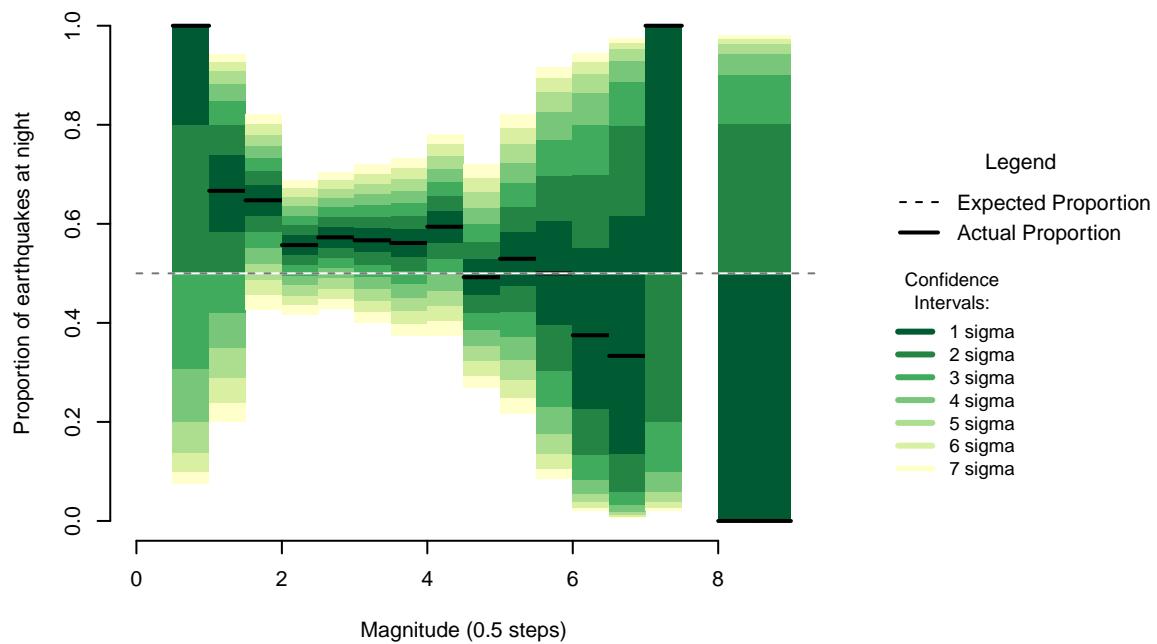


Figure 14.2: Proportion of night earthquakes by magnitude, Thailand n=2771

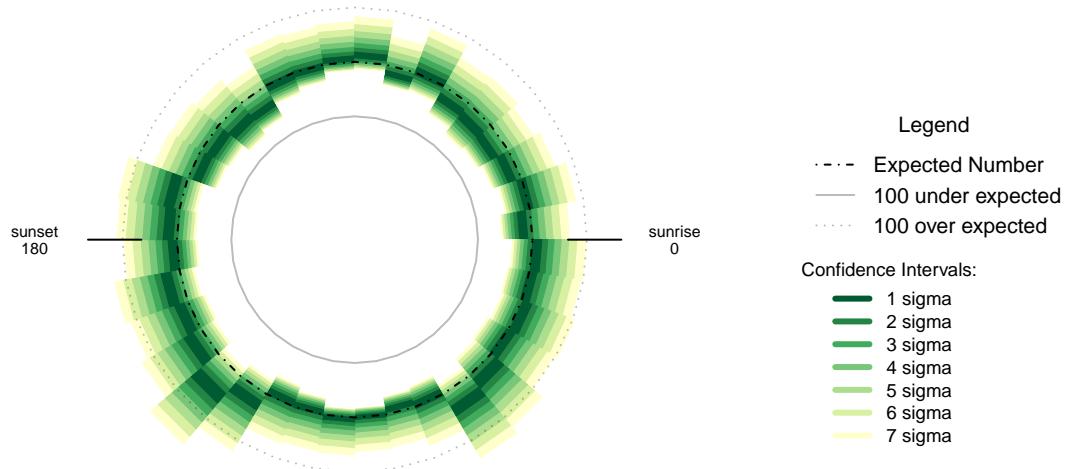


Figure 14.3: Over- and under-supply of earthquakes by angle of the sun (10 degree steps), Thailand n=2771

## 14.1 Formal Statement

Earthquakes in the region of Thailand show a similar pattern to New Zealand, displaying an oversupply of earthquakes at night that is not the result of chance. While the sample size is too small to make conclusions as confident as the other, larger, datasets, the magnitude patterns are not inconsistent with the general patterns observed in other data sets.

## 14.2 Links

1 - Thai Meteorological Department, Seismology Bureau [http://www.earthquake.tmd.go.th/en/local.php?pageNum\\_thaievent=106&totalRows\\_thaievent=4247](http://www.earthquake.tmd.go.th/en/local.php?pageNum_thaievent=106&totalRows_thaievent=4247)

## 14.3 Chapter Code

```
## ----setup, include=FALSE-----
knitr::opts_chunk$set(echo = FALSE)

## ----c10_001, warnings=FALSE, errors=FALSE, message=FALSE-----
library(geosphere)
library(lubridate, quietly=TRUE)
library(dplyr)
library(binom)
library(ggplot2)
library(maps)
library(mapdata)
library(parallel)
library(readr)
library(plotrix)
library(tidyr)
library(maptools)
library(rvest)
```

```

Sys.setenv(TZ = "UTC")

## ----warnings=FALSE, errors=FALSE, message=FALSE-----

if(!dir.exists("../othereqdata")){
  dir.create("../othereqdata")
}
if(!file.exists("../othereqdata/eq_thailand_raw.RData")){

assemble_eq <- function(x){
  wbgc <- paste("http://www.earthquake.tmd.go.th/en/local.php?pageNum_thaievent=",
                x, "&totalRows_thaievent=4247", sep="")
  th_eq <- read_html(wbgc) %>%
    html_table(header=TRUE)
  return(th_eq[[3]])
}

eq_list <- lapply(1:105, assemble_eq)
eq_thailand <- bind_rows(eq_list)
names(eq_thailand) <- c("origin_time", "magnitude", "latitude", "longitude", "region")
eq_thailand$time_UTC <- ymd_hms(eq_thailand$origin_time)

eq_national <- eq_thailand %>%
  filter(magnitude >= 0 &
         time_UTC >= as.POSIXct("2011-09-01T00:00:00",
                                  format="%Y-%m-%dT%H:%M:%S", tz="UTC") &
         time_UTC < as.POSIXct("2016-09-01T00:00:00",
                                  format="%Y-%m-%dT%H:%M:%S", tz="UTC")) %>%
  distinct() %>% arrange(time_UTC)
#one longitude was missing the decimal sign

rm(eq_thailand)
save(eq_national, file="../othereqdata/eq_thailand_raw.RData")
}

## ----

if(!file.exists("../othereqdata/eq_thailand_processed.RData")){
load("../othereqdata/eq_thailand_raw.RData")
southmost <- min(eq_national$latitude)
westmost <- min(eq_national$longitude)

eq_national <- eq_national %>% filter(
  magnitude > 0) %>% rowwise() %>% mutate(
  eq_gridpoint_y = round(distVincentyEllipsoid(c(longitude, southmost),
                                                 c(longitude, latitude)) /50000,0),
  eq_gridpoint_x = round(distVincentyEllipsoid(c(westmost, latitude),
                                                 c(longitude, latitude)) /50000,0),
  eq_roundedlat = destPoint(p=c(longitude, southmost),

```

```

        b=0, d=eq_gridpoint_y*50000)[2],
eq_roundedlong = destPoint(p=c(westmost, eq_roundedlat),
                           b=90, d=eq_gridpoint_x*50000)[1]) %>% ungroup()
# use maptools to calculate solar angles
sun_angles <- solarpes(matrix(c(eq_national$longitude, eq_national$latitude), ncol=2),
                        eq_national$time_UTC)
colnames(sun_angles) <- c("eq_compass", "eq_vertical")
eq_national <- cbind(eq_national,sun_angles)
eq_national$eq_is_night <- eq_national$eq_vertical < 0
# calculate 360 degree as well as vertical
eq_national <- eq_national %>%
  mutate(eq_angle_360 = eq_vertical,
        eq_angle_360 = ifelse(eq_compass > 180, 180 - eq_angle_360, eq_angle_360),
        eq_angle_360 = ifelse(eq_vertical < 0 & eq_compass <= 180,
                             360 + eq_angle_360, eq_angle_360),
        eq_angle_by_10 = floor(eq_angle_360 /10) * 10)
save(eq_national, file="../othereqdata/eq_thailand_processed.RData")
}

## -----
if(!file.exists("../othereqdata/eq_thailand_expected.RData")){
load("../othereqdata/eq_thailand_processed.RData")

lat_range <- unique(eq_national$eq_roundedlat)
long_med <- median(eq_national$eq_roundedlong)

# 1 minute intervals for a full solar year
time1 <- ymd_hms("2015-01-01 00:00:00")
time2 <- ymd_hms("2015-12-31 23:59:00")
time_sq <- seq.POSIXt(from=time1, to=time2, by="min")

calc_angs <- function(x, longinput, timeinput){
  library(dplyr)
  sun_angles <- maptools::solarpos(matrix(c(longinput, x), ncol=2), timeinput)
  colnames(sun_angles) <- c("eq_compass", "eq_vertical")
  # calculate 360 degree as well as vertical
  site_summary <- as.data.frame(sun_angles) %>%
    mutate(eq_angle_360 = eq_vertical,
          eq_angle_360 = ifelse(eq_compass > 180, 180 - eq_angle_360, eq_angle_360),
          eq_angle_360 = ifelse(
            eq_vertical < 0 & eq_compass <= 180, 360 + eq_angle_360, eq_angle_360),
          eq_angle_by_10 = floor(eq_angle_360 /10) * 10) %>%
    group_by(eq_angle_by_10) %>% summarise(total= n())
  site_summary$lat <- x
  return(site_summary)
}
####
# Calculate the number of cores
no_cores <- detectCores() - 1
# Initiate cluster
cl <- makeCluster(no_cores)
clusterExport(cl, varlist=c("lat_range", "long_med", "time_sq", "calc_angs"))

```

```

list_angs <- parLapply(cl, lat_range,
                       function(x){
                           calc_angs(x=x, longinput=long_med, timeinput=time_sq)})
stopCluster(cl)
##
library(tidyr)
anglong <- bind_rows(list_angs)
angwide <- spread(anglong, key=eq_angle_by_10,value=total)
rm(anglong, list_angs, time_sq)
save(angwide, file="../othereqdata/eq_thailand_expected.RData")
}

## -----
load("../othereqdata/eq_thailand_processed.RData")
load("../othereqdata/eq_thailand_expected.RData")
eq_night = sum(eq_national$eq_is_night)
eq_total = nrow(eq_national)
bands <- rev(c('#ffffcc', '#d9f0a3', '#add8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
sigmas <- c(0.682689492137086,
           0.954499736103642,
           0.997300203936740,
           0.999936657516334,
           0.999999426696856,
           0.99999998026825,
           0.99999999997440)
lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
old_par=par()

## -----
bt <- binom.test(eq_night ,eq_total, conf.level= .999999999997440)

## -----
feature <- c("Earliest (UTC)", "Latest (UTC)",
            "Northernmost", "Southernmost",
            "Westmost", "Eastmost",
            "Percent < Mag 3", "total entries",
            "nighttime quakes")
value <- c(as.character(min(eq_national$time_UTC)),
            as.character(max(eq_national$time_UTC)),
            as.character(max(eq_national$latitude)),
            as.character(min(eq_national$latitude)),
            as.character(min(eq_national$longitude)),
            as.character(max(eq_national$longitude)),
            as.character(round(100*sum(eq_national$magnitude < 3)/eq_total,2)),
            as.character(eq_total),
            as.character(eq_night))
data.frame(feature,value) %>% knitr::kable(caption="Data description")

```

```

## ---- fig.cap="Proportion of earthquakes at night: Thailand n=2771"-----
### making the basic proportion graph
eq_night = sum(eq_national$eq_is_night)
eq_total = nrow(eq_national)
bands <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
sigmas <- c(0.682689492137086,
           0.954499736103642,
           0.997300203936740,
           0.999936657516334,
           0.999999426696856,
           0.99999998026825,
           0.9999999997440)
lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
old_par=par()
conf_steps <- function(x, sigmas=sigmas, night=eq_night, total=eq_total){
  ci_lower <- binom.confint(night, total, method=c("wilson"), conf.level = sigmas[x])[1,5]
  ci_upper <- binom.confint(night, total, method=c("wilson"), conf.level = sigmas[x])[1,6]
  ci_data <- data.frame(step = x, ci_lower, ci_upper)
}
ci_spacing <- lapply(7:1, conf_steps, sigmas=sigmas, night=eq_night, total=eq_total)
ci_steps <- bind_rows(ci_spacing)

layout(matrix(c(1,1,1,2), ncol=4))

par(mar=c(5,6,4,2))
plot(c(min(0.5,floor(100*ci_steps[1,2])/100), max(0.5,ceiling(100*ci_steps[1,3])/100)),
      y=c(-3,8), type="n", bty="n", yaxt="n", ylab="",
      xlab="Proportion of earthquakes at night")
a <- a <- apply(ci_steps, 1, function(x){
  polygon(c(x[2], x[3], x[3], x[2]), c(0, 0, 1, 1), col=bands[x[1]], border=NA)})
lines(c(.5,.5), c(0,1), col="#FFFFFF")
lines(c(.5,.5), c(0,1), lty=2, col="#777777")
lines(c(eq_night/eq_total,eq_night/eq_total), c(0,1), lwd=2)

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE)
legend(0,5.5, legend=lbls, lty=typs, lwd=weights, col=bands, bty="n", xjust=0,
       title="Confidence Intervals:", y.intersp=1.1, cex=0.9)

lbls2=c("50% Night", "Actual Proportion")
typs2=c(2,1)
weights2=c(1,2)
cls2=c("#777777", "#000000")
legend(0,7, legend=lbls2, lty=typs2, lwd=weights2, col=cls2, bty="n", xjust=0,
       title="Legend", y.intersp=1.2)

```

```

par(mar=old_par$mar)
par(mfrow=c(1,1))

## ---- fig.cap="Proportion of night earthquakes by magnitude, Thailand n=2771"----
old_par=par()
grf <- eq_national %>% mutate(floored_mag = floor(magnitude*2)/2) %>%
  group_by(floored_mag) %>% summarise(successes = sum(eq_is_night), trials=n())

poly_conf_int <- function(success, trials, aa, stepsize, sigma, colr){
  ci <- binom.confint(success, trials, method=c("wilson"), conf.level = sigma)
  lower <- ci[1,5]
  upper <- ci[1,6]
  a <- polygon(x=c(aa,aa+stepsize,aa+stepsize,aa), y=c(upper,upper,lower,lower),
                col=colr, border=NA)
}

plot7sig <- function(success, trials, aa, stepsize){
  library(binom)
  #bands <- c('#fffffb2', '#fed976', '#feb24c', '#fd8d3c', '#fc4e2a', '#e31a1c', '#b10026')
  bands <- rev(c('#fffffcc', '#d9f0a3', '#add8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
  sigmas <- c(0.682689492137086,
             0.954499736103642,
             0.997300203936740,
             0.999936657516334,
             0.999999426696856,
             0.999999998026825,
             0.99999999997440)
  sapply(7:1, function(x){
    poly_conf_int(success, trials, aa, stepsize, sigmas[x], bands[x]))}
  a <- lines(c(aa, aa + stepsize), c(success/trials, success/trials), lwd=2)
}

lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")

typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
clrs <- rev(c('#fffffcc', '#d9f0a3', '#add8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
#clrs <- c('#fffffb2', '#fed976', '#feb24c', '#fd8d3c', '#fc4e2a', '#e31a1c', '#b10026')

layout(matrix(c(1,1,1,2), ncol=4))
plot(x=c(0,max(grf$floored_mag)+0.5), y=c(0,1), type="n", bty="n",
      xlab="Magnitude (0.5 steps)", ylab="Proportion of earthquakes at night")
a <- apply(grf,1,function(x){plot7sig(x[2],x[3],x[1],0.5)})
lines(c(0,10), c(.5,.5), col="#FFFFFF")
lines(c(0,10), c(.5,.5), lty=2, col="#777777")

```

```

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE)
legend(0,5, legend=lbls, lty=typs, lwd=weights, col=clrs, bty="n", xjust=0,
       title="Confidence
Intervals:", cex=0.9)
lbls=c("Expected Proportion", "Actual Proportion")
typs=c(2,1)
weights=c(1,2)
legend(0,7, legend=lbls, lty=typs, lwd=weights, bty="n", xjust=0,
       title="Legend", y.intersp=1.2)

par(mar=old_par$mar)
par(mfrow=c(1,1))

## -----
by_angle <- eq_national %>%
  group_by(eq_angle_by_10) %>% summarise(total= n()) %>%
  mutate(daynight=ifelse(eq_angle_by_10 < 180, "day", "night"))

merged <- merge(eq_national, angwide, by.x="eq_roundedlat", by.y="lat")
agg_expected <- merged %>% select(`0`:`350`) %>% colSums(na.rm=TRUE)
expected_prop <- agg_expected / sum(agg_expected)
expected <- data.frame(eq_angle_by_10 = as.numeric(names(expected_prop)),
                        expected_prop = as.numeric(expected_prop))
expected$expected_number = expected_prop * eq_total
act_exp <- merge(expected, by_angle, by="eq_angle_by_10", all.x=TRUE)

act_exp$total[is.na(act_exp$total)] <- 0
act_exp$daynight <- NULL
act_exp$act_prop <- act_exp$total / sum(act_exp$total)

ci_brackets <- act_exp %>% ungroup() %>% mutate(grand_total=sum(total)) %>%
  rowwise() %>% mutate(
    ci_lower_7 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[7])[1,5] * grand_total,
    ci_upper_7 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[7])[1,6] * grand_total,
    ci_lower_6 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[6])[1,5] * grand_total,
    ci_upper_6 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[6])[1,6] * grand_total,
    ci_lower_5 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[5])[1,5] * grand_total,
    ci_upper_5 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[5])[1,6] * grand_total,
    ci_lower_4 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[4])[1,5] * grand_total,
    ci_upper_4 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[4])[1,6] * grand_total,
    ci_lower_3 = binom.confint(total, grand_total, method=c("wilson"),

```

```

                conf.level = sigmas[3])[1,5] * grand_total,
ci_upper_3 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[3])[1,6] * grand_total,
ci_lower_2 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[2])[1,5] * grand_total,
ci_upper_2 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[2])[1,6] * grand_total,
ci_lower_1 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[1])[1,5] * grand_total,
ci_upper_1 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[1])[1,6] * grand_total)

norm_ci <- ci_brackets

for (i in c(4,7:20)){
norm_ci[,i] <- ci_brackets[,i] - ci_brackets[,3]
}
circlesize=100

## ---- fig.cap="Over- and under- supply of earthquakes by angle of the sun
## (10 degree steps), Thailand n=2771"----
norm_ci$border = 2
# need to double entries with a displacement of 10
# to make each side of the item on the graph
norm_ci2 <- norm_ci
norm_ci2$eq_angle_by_10 <- norm_ci2$eq_angle_by_10 + 10
norm_ci2$border = 1
graphdata <- bind_rows(norm_ci,norm_ci2) %>% arrange(eq_angle_by_10,border)

#### plot graph
bands <- rev(c('#ffffcc','#d9f0a3','#addd8e','#78c679','#41ab5d','#238443','#005a32'))
old_par=par()
layout(matrix(c(1,1,1,2), ncol=4))

# overall limits
limits=2 * max(abs(c(graphdata$ci_lower_7, graphdata$ci_upper_7)))
# plot upper confidence 7 interval using plotrix
polar.plot(graphdata$ci_upper_7, polar.pos=graphdata$eq_angle_by_10,
           radial.lim=c(-1*limits,limits),
           labels = "", main=NULL,lwd=0.5, rp.type="p",
           show.grid.labels=FALSE, show.grid=FALSE, mar=c(0,0,0,0),
           grid.col=bands[7], line.col=bands[7], poly.col=bands[7])
# plot upper 6 confidence interval
plot_ci_round <- function(upper_bound,x){
  polar.plot(upper_bound, polar.pos=graphdata$eq_angle_by_10, add=TRUE,
             radial.lim=c(-1*limits,limits),
             line.col=bands[x], lwd=0.5, rp.type="p", poly.col=bands[x])
}

plot_ci_round(graphdata$ci_upper_6, 6)
plot_ci_round(graphdata$ci_upper_5, 5)
plot_ci_round(graphdata$ci_upper_4, 4)
plot_ci_round(graphdata$ci_upper_3, 3)
plot_ci_round(graphdata$ci_upper_2, 2)

```

```

plot_ci_round(graphdata$ci_upper_1, 1)
plot_ci_round(graphdata$ci_lower_1, 2)
plot_ci_round(graphdata$ci_lower_2, 3)
plot_ci_round(graphdata$ci_lower_3, 4)
plot_ci_round(graphdata$ci_lower_4, 5)
plot_ci_round(graphdata$ci_lower_5, 6)
plot_ci_round(graphdata$ci_lower_6, 7)
polar.plot(graphdata$ci_lower_7, polar.pos=graphdata$eq_angle_by_10,
           add=TRUE, radial.lim=c(-1*limits,limits),
           line.col="white", lwd=0.5, rp.type="p", poly.col="white")

# plot expected guide line
polar.plot(rep(0,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10, add=TRUE,
           radial.lim=c(-1*limits,limits),
           rp.type="p", lty=4)
# plot 500 less than expected guide line
polar.plot(rep(-1 * circlesize,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10,
           add=TRUE,radial.lim=c(-1*limits,limits),
           rp.type="p", lty=1, line.col="#00000044")
# plot 500 more than expected guide line
polar.plot(rep(circlesize,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10,
           add=TRUE,radial.lim=c(-1*limits,limits),
           rp.type="p", lty=3, line.col="#00000044")

lines(c(-1.5,-1.2)*limits, c(0,0))
lines(c(1.5,1.2)*limits, c(0,0))
text(-1.8*limits,0, label="sunset
180", cex=0.7)
text(1.8*limits,0, label="sunrise
0", cex=0.7)

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE, xlab="")
lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
clrs <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
legend(0,4.5, legend=lbls, lty=typs, lwd=weights, col=clrs, bty="n", xjust=0,
       title="Confidence Intervals:", cex=0.9)

lbls2=c("Expected Number", paste(circlesize,"under expected"),
       paste(circlesize,"over expected"))
typs2=c(4,1,3)
weights2=c(1,1,1)
clrs2=c("#000000", "#00000044", "#00000044")
legend(0,10, legend=lbls2, lty=typs2, lwd=weights2, bty="n", xjust=0,
       title="Legend", y.intersp=1.2, col=clrs2)

```

```
par(mar=old_par$mar)
par(mfrow=c(1,1))
```

# Chapter 15

## Taiwan is not similar

Taiwan earthquake data is available from the Seismology Center, Central Weather Bureau. It was retrieved directly into R by automatically filling out the web form to supply each months worth of data.

For the same period as the New Zealand data (from September 2011 to September 2016), in earthquakes data from the Taiwan there are 3353 events of depth greater than 0 and magnitude greater than 0.

Of the 3353 in the data, 1700 occurred at night, a proportion of 0.507. A seven sigma confidence interval for the proportion of earthquakes occurring at night is 0.4466 to 0.5673, which 0.5 is well inside. A proportion of 0.5 is not even outside one confidence interval.

If this collection of 3353 earthquakes was all I had to test with, I would have no evidence that more earthquakes happen at night. Fortunately, I have more than 1.1 million earthquakes from other sources that indicate the general behaviour is otherwise, or I would not have much of a book.

While there is not enough data to be conclusive, it looks like for earthquakes in the 2.5 to 4.5 range (those we can be most exact about) the proportions are near to 50%. Part of the unusualness of this data set may stem from the absence of low magnitude earthquakes in the data (which often occur at night in other data sets). Below 3 magnitude earthquakes makeup under 25% of the recorded earthquakes, compared to over 90% for some of the larger collections.

Unsurprisingly, given the general result, the pattern with respect to the angle of the sun is the closest to a perfect circle of all the country data sets.

Table 15.1: Data description

feature	value
Earliest (UTC)	2011-09-02 01:17:00
Latest (UTC)	2016-08-31 16:47:00
Northernmost	25.97
Southernmost	20.05
Westmost	118.75
Eastmost	123.69
Percent < Mag 3	24.58
total entries	3353
nighttime quakes	1700

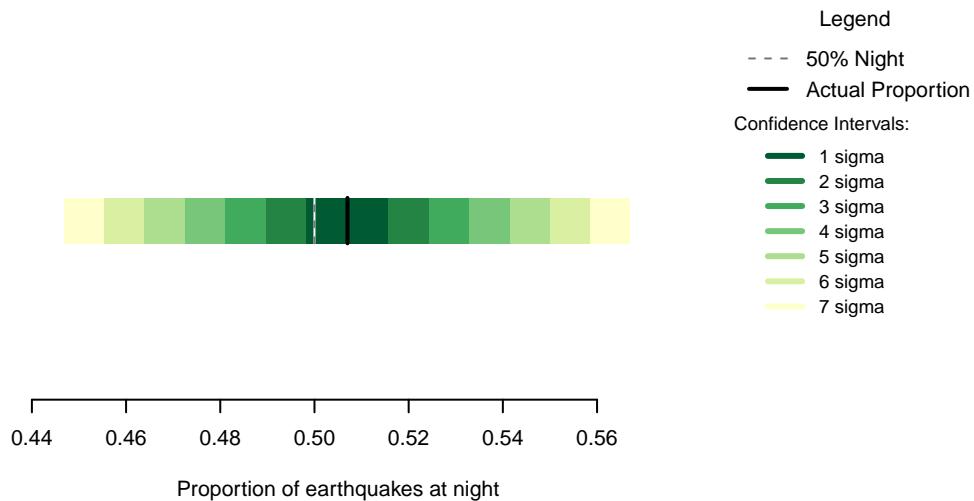


Figure 15.1: Proportion of earthquakes at night: Taiwan. n=3353

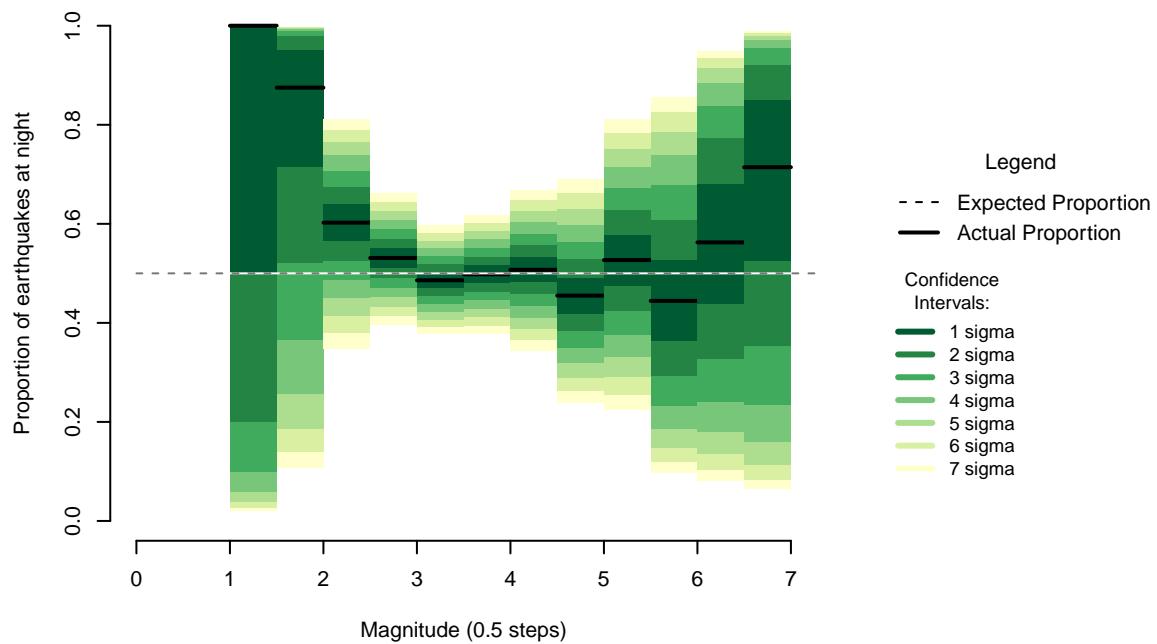


Figure 15.2: Proportion of night earthquakes by magnitude, Taiwan. n=3353

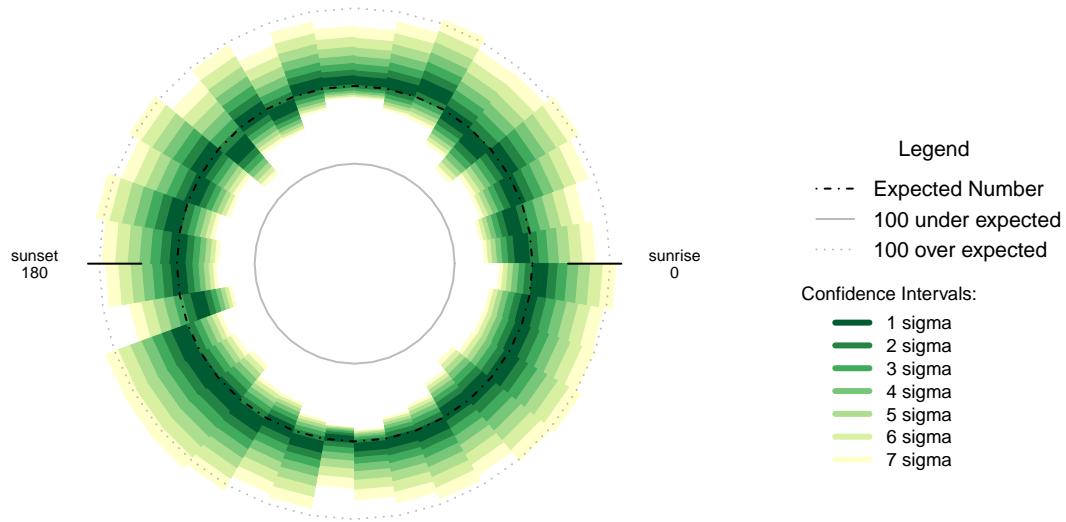


Figure 15.3: Over- and under-supply of earthquakes by angle of the sun (10 degree steps). Taiwan. n=3353

## 15.1 Formal Statement

Earthquakes in the Taiwan show no evidence of an increased risk of nighttime earthquakes. The Taiwanese data set does not contain the night-sensitive low magnitude earthquakes present in other data sets.

## 15.2 Links

1 - Seismology Center, Central Weather Bureau: <http://scweb.cwb.gov.tw/Default.aspx?loc=en>

## 15.3 Chapter Code

```
## ----setup, include=FALSE-----
knitr::opts_chunk$set(echo = FALSE)

## ----c10_001, warnings=FALSE, errors=FALSE, message=FALSE-----
library(geosphere)
library(lubridate, quietly=TRUE)
library(dplyr)
library(binom)
library(ggplot2)
library(maps)
library(mapdata)
library(parallel)
library(readr)
library(plotrix)
library(tidyr)
library(mapproj)
library(rvest)
Sys.setenv(TZ = "UTC")
```

```

## ----warnings=FALSE, errors=FALSE, message=FALSE-----

if(!dir.exists("../othereqdata")){
  dir.create("../othereqdata")
}
if(!file.exists("../othereqdata/eq_taiwan_raw.RData")){

  equrl <- 'http://scweb.cwb.gov.tw/Page.aspx?ItemId=26&loc=en&gis=n'
  eq_session <- html_session(equrl)
  eq_form <- html_form(eq_session)[[1]]

assemble_eq <- function(x, session_is, form_is){
  yr <- as.character(x[1])
  mnth <- formatC(x[2], width = 2, format = "d", flag = "0")
  filled_form <- set_values(form, `ctl03$ddlYear` = yr, `ctl03$ddlMonth` = mnth)
  out <- submit_form(session = eq_session, filled_form)
  eqtw <- out %>% html_nodes("#ctl03_gvEarthquake") %>%
    html_table() %>% .[[1]]
  return(eqtw)
}

frmfl1 <- data.frame(yrs = rep(2011:2016, each=12),mnths = rep(1:12, times=12))
eq_list <- apply(frmfl1, 1, assemble_eq, session_is=eq_session, form_is=eq_form)
eq_taiwan <- bind_rows(eq_list)
names(eq_taiwan) <- c("class", "origin_time", "longitude", "latitude",
  "magnitude", "depth", "location")
eq_taiwan$time_UTC <- ymd_hm(eq_taiwan$origin_time) - hours(8)
eq_taiwan$depth <- as.numeric(gsub(" km","",eq_taiwan$depth))
eq_national <- eq_taiwan %>%
  filter(magnitude >= 0 &
    time_UTC >= as.POSIXct("2011-09-01T00:00:00",
      format="%Y-%m-%dT%H:%M:%S", tz="UTC") &
    time_UTC < as.POSIXct("2016-09-01T00:00:00",
      format="%Y-%m-%dT%H:%M:%S", tz="UTC")) %>%
  distinct() %>% arrange(time_UTC)
#one longitude was missing the decimal sign

rm(eq_taiwan)
save(eq_national, file="../othereqdata/eq_taiwan_raw.RData")
}

## ----

if(!file.exists("../othereqdata/eq_taiwan_processed.RData")){
load("../othereqdata/eq_taiwan_raw.RData")
southmost <- min(eq_national$latitude)
westmost <- min(eq_national$longitude)

eq_national <- eq_national %>% filter(
  magnitude > 0) %>% rowwise() %>% mutate(

```

```

eq_gridpoint_y = round(distVincentyEllipsoid(c(longitude, southmost),
                                              c(longitude,latitude)) /50000,0),
eq_gridpoint_x = round(distVincentyEllipsoid(c(westmost, latitude),
                                              c(longitude,latitude)) /50000,0),
eq_roundedlat = destPoint(p=c(longitude, southmost),
                           b=0, d=eq_gridpoint_y*50000)[2],
eq_roundedlong = destPoint(p=c(westmost, eq_roundedlat),
                           b=90, d=eq_gridpoint_x*50000)[1]) %>% ungroup()
# use maptools to calculate solar angles
sun_angles <- solarpos(matrix(c(eq_national$longitude, eq_national$latitude), ncol=2),
                         eq_national$time_UTC)
colnames(sun_angles) <- c("eq_compass", "eq_vertical")
eq_national <- cbind(eq_national, sun_angles)
eq_national$eq_is_night <- eq_national$eq_vertical < 0
# calculate 360 degree as well as vertical
eq_national <- eq_national %>%
  mutate(eq_angle_360 = eq_vertical,
        eq_angle_360 = ifelse(eq_compass > 180, 180 - eq_angle_360, eq_angle_360),
        eq_angle_360 = ifelse(eq_vertical < 0 & eq_compass <= 180,
                             360 + eq_angle_360, eq_angle_360),
        eq_angle_by_10 = floor(eq_angle_360 /10) * 10)
save(eq_national, file="../othereqdata/eq_taiwan_processed.RData")
}

## -----
if(!file.exists("../othereqdata/eq_taiwan_expected.RData")){
load("../othereqdata/eq_taiwan_processed.RData")

lat_range <- unique(eq_national$eq_roundedlat)
long_med <- median(eq_national$eq_roundedlong)

# 1 minute intervals for a full solar year
time1 <- ymd_hms("2015-01-01 00:00:00")
time2 <- ymd_hms("2015-12-31 23:59:00")
time_sq <- seq.POSIXt(from=time1, to=time2, by="min")

calc_angs <- function(x, longinput, timeinput){
  library(dplyr)
  sun_angles <- maptools::solarpos(matrix(c(longinput, x), ncol=2), timeinput)
  colnames(sun_angles) <- c("eq_compass", "eq_vertical")
  # calculate 360 degree as well as vertical
  site_summary <- as.data.frame(sun_angles) %>%
    mutate(eq_angle_360 = eq_vertical,
          eq_angle_360 = ifelse(eq_compass > 180, 180 - eq_angle_360, eq_angle_360),
          eq_angle_360 = ifelse(
            eq_vertical < 0 & eq_compass <= 180, 360 + eq_angle_360, eq_angle_360),
          eq_angle_by_10 = floor(eq_angle_360 /10) * 10) %>%
    group_by(eq_angle_by_10) %>% summarise(total= n())
  site_summary$lat <- x
  return(site_summary)
}
###
```

```

# Calculate the number of cores
no_cores <- detectCores() - 1
# Initiate cluster
cl <- makeCluster(no_cores)
clusterExport(cl, varlist=c("lat_range", "long_med", "time_sq", "calc_angs"))
list_angs <- parLapply(cl, lat_range,
                       function(x){
                           calc_angs(x=x, longinput=long_med, timeinput=time_sq)})
stopCluster(cl)
###  

library(tidyr)
anglong <- bind_rows(list_angs)
angwide <- spread(anglong, key=eq_angle_by_10,value=total)
rm(anglong, list_angs, time_sq)
save(angwide, file="../othereqdata/eq_taiwan_expected.RData")
}  

## -----
load("../othereqdata/eq_taiwan_processed.RData")
load("../othereqdata/eq_taiwan_expected.RData")
eq_night = sum(eq_national$eq_is_night)
eq_total = nrow(eq_national)
bands <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
sigmas <- c(0.682689492137086,
0.954499736103642,
0.997300203936740,
0.999936657516334,
0.999999426696856,
0.99999998026825,
0.999999999997440)
lbls <- c(
"1 sigma", "2 sigma",
"3 sigma", "4 sigma",
"5 sigma", "6 sigma",
"7 sigma")
typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
old_par=par()  

## -----
bt <- binom.test(eq_night ,eq_total, conf.level= .999999999997440)  

## -----
feature <- c("Earliest (UTC)", "Latest (UTC)",
"Northernmost", "Southernmost",
"Westmost", "Eastmost",
"Percent < Mag 3", "total entries",
"nighttime quakes")
value <- c(as.character(min(eq_national$time_UTC)),
as.character(max(eq_national$time_UTC)),
as.character(max(eq_national$latitude)),
as.character(min(eq_national$latitude)),
as.character(min(eq_national$longitude)),
as.character(max(eq_national$longitude)),
as.character(min(eq_national$longitude)))

```

```

as.character(max(eq_national$longitude)),
as.character(round(100*sum(eq_national$magnitude < 3)/eq_total,2)),
as.character(eq_total),
as.character(eq_night))
data.frame(feature,value) %>% knitr::kable(caption="Data description")

## ---- fig.cap="Proportion of earthquakes at night: Taiwan. n=3353"-----
### making the basic proportion graph
eq_night = sum(eq_national$eq_is_night)
eq_total = nrow(eq_national)
bands <- rev(c('#ffffcc', '#d9f0a3', '#add8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
sigmas <- c(0.682689492137086,
           0.954499736103642,
           0.997300203936740,
           0.999936657516334,
           0.999999426696856,
           0.99999998026825,
           0.99999999997440)
lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
tys <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
old_par=par()
conf_steps <- function(x, sigmas=sigmas, night=eq_night, total=eq_total){
  ci_lower <- binom.confint(night, total, method=c("wilson"), conf.level = sigmas[x])[1,5]
  ci_upper <- binom.confint(night, total, method=c("wilson"), conf.level = sigmas[x])[1,6]
  ci_data <- data.frame(step = x, ci_lower, ci_upper)
}
ci_spacing <- lapply(7:1, conf_steps, sigmas=sigmas, night=eq_night, total=eq_total)
ci_steps <- bind_rows(ci_spacing)

layout(matrix(c(1,1,1,2), ncol=4))

par(mar=c(5,6,4,2))
plot(c(min(0.5,floor(100*ci_steps[1,2])/100), max(0.5,ceiling(100*ci_steps[1,3])/100)),
      y=c(-3,8), type="n", bty="n", yaxt="n", ylab="",
      xlab="Proportion of earthquakes at night")
a <- a <- apply(ci_steps, 1, function(x){
  polygon(c(x[2], x[3], x[3], x[2]), c(0, 0, 1, 1), col=bands[x[1]], border=NA))
  lines(c(.5,.5), c(0,1), col="#FFFFFF")
  lines(c(.5,.5), c(0,1), lty=2, col="#777777")
  lines(c(eq_night/eq_total,eq_night/eq_total), c(0,1), lwd=2)

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE)
legend(0,5.5, legend=lbls, lty=tys, lwd=weights, col=bands, bty="n", xjust=0,
       title="Confidence Intervals:", y.intersp=1.1, cex=0.9)

lbls2=c("50% Night", "Actual Proportion")

```

```

typs2=c(2,1)
weights2=c(1,2)
cls2=c("#777777", "#000000")
legend(0,7, legend=lbls2, lty=typs2, lwd=weights2, col=cls2, bty="n", xjust=0,
       title="Legend", y.intersp=1.2)

par(mar=old_par$mar)
par(mfrow=c(1,1))

## ---- fig.cap="Proportion of night earthquakes by magnitude, Taiwan. n=3353"----
old_par=par()
grf <- eq_national %>% mutate(floored_mag = floor(magnitude*2)/2) %>%
  group_by(floored_mag) %>% summarise(successes = sum(eq_is_night), trials=n())

poly_conf_int <- function(success, trials, aa, stepsize, sigma, colr){
  ci <- binom.confint(success, trials, method=c("wilson"), conf.level = sigma)
  lower <- ci[1,5]
  upper <- ci[1,6]
  a <- polygon(x=c(aa,aa+stepsize,aa+stepsize,aa), y=c(upper,upper,lower,lower),
                col=colr, border=NA)
}

plot7sig <- function(success, trials, aa, stepsize){
  library(binom)
  #bands <- c('#fffffb2', '#fed976', '#feb24c', '#fd8d3c', '#fc4e2a', '#e31a1c', '#b10026')
  bands <- rev(c('#fffffc', '#d9f0a3', '#add8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
  sigmas <- c(0.682689492137086,
             0.954499736103642,
             0.997300203936740,
             0.999936657516334,
             0.999999426696856,
             0.999999998026825,
             0.99999999997440)
  sapply(7:1, function(x){
    poly_conf_int(success, trials, aa, stepsize, sigmas[x], bands[x]))}
  a <- lines(c(aa, aa + stepsize), c(success/trials, success/trials), lwd=2)
}

lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
clrs <- rev(c('#fffffc', '#d9f0a3', '#add8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
#clrs <- c('#fffffb2', '#fed976', '#feb24c', '#fd8d3c', '#fc4e2a', '#e31a1c', '#b10026')

```

```

layout(matrix(c(1,1,1,2), ncol=4))
plot(x=c(0,max(grf$floored_mag)+0.5), y=c(0,1), type="n", bty="n",
      xlab="Magnitude (0.5 steps)", ylab="Proportion of earthquakes at night")
a <- apply(grf,1,function(x){plot7sig(x[2],x[3],x[1],0.5)})
lines(c(0,10), c(.5,.5), col="#FFFFFF")
lines(c(0,10), c(.5,.5), lty=2, col="#777777")

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE)
legend(0.5, legend=lbls, lty=typs, lwd=weights, col=clrs, bty="n", xjust=0,
       title="Confidence
Intervals:", cex=0.9)
lbls=c("Expected Proportion", "Actual Proportion")
typs=c(2,1)
weights=c(1,2)
legend(0.7, legend=lbls, lty=typs, lwd=weights, bty="n", xjust=0,
       title="Legend", y.intersp=1.2)

par(mar=old_par$mar)
par(mfrow=c(1,1))

## -----
by_angle <- eq_national %>%
  group_by(eq_angle_by_10) %>% summarise(total= n()) %>%
  mutate(daynight=ifelse(eq_angle_by_10 < 180, "day", "night"))

merged <- merge(eq_national, angwide, by.x="eq_roundedlat", by.y="lat")
agg_expected <- merged %>% select(`0`:`350`) %>% colSums(na.rm=TRUE)
expected_prop <- agg_expected / sum(agg_expected)
expected <- data.frame(eq_angle_by_10 = as.numeric(names(expected_prop)),
                       expected_prop = as.numeric(expected_prop))
expected$expected_number = expected_prop * eq_total
act_exp <- merge(expected, by_angle, by="eq_angle_by_10", all.x=TRUE)

act_exp$total[is.na(act_exp$total)] <- 0
act_exp$daynight <- NULL
act_exp$act_prop <- act_exp$total / sum(act_exp$total)

ci_brackets <- act_exp %>% ungroup() %>% mutate(grand_total=sum(total)) %>%
  rowwise() %>% mutate(
    ci_lower_7 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[7])[1,5] * grand_total,
    ci_upper_7 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[7])[1,6] * grand_total,
    ci_lower_6 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[6])[1,5] * grand_total,
    ci_upper_6 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[6])[1,6] * grand_total,
    ci_lower_5 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[5])[1,5] * grand_total,
    ci_upper_5 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[5])[1,6] * grand_total,
    ci_lower_4 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[4])[1,5] * grand_total,
    ci_upper_4 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[4])[1,6] * grand_total,
    ci_lower_3 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[3])[1,5] * grand_total,
    ci_upper_3 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[3])[1,6] * grand_total,
    ci_lower_2 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[2])[1,5] * grand_total,
    ci_upper_2 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[2])[1,6] * grand_total,
    ci_lower_1 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[1])[1,5] * grand_total,
    ci_upper_1 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[1])[1,6] * grand_total)
  
```

```

ci_upper_5 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[5])[1,6] * grand_total,
ci_lower_4 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[4])[1,5] * grand_total,
ci_upper_4 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[4])[1,6] * grand_total,
ci_lower_3 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[3])[1,5] * grand_total,
ci_upper_3 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[3])[1,6] * grand_total,
ci_lower_2 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[2])[1,5] * grand_total,
ci_upper_2 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[2])[1,6] * grand_total,
ci_lower_1 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[1])[1,5] * grand_total,
ci_upper_1 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[1])[1,6] * grand_total)

norm_ci <- ci_brackets

for (i in c(4,7:20)){
norm_ci[,i] <- ci_brackets[,i] - ci_brackets[,3]
}
circlesize=100

## ---- fig.cap="Over- and under- supply of earthquakes by angle of the sun
## (10 degree steps). Taiwan. n=3353"----
norm_ci$border = 2
# need to double entries with a displacement of 10
# to make each side of the item on the graph
norm_ci2 <- norm_ci
norm_ci2$eq_angle_by_10 <- norm_ci2$eq_angle_by_10 + 10
norm_ci2$border = 1
graphdata <- bind_rows(norm_ci,norm_ci2) %>% arrange(eq_angle_by_10,border)

#### plot graph
bands <- rev(c('#ffffcc','#d9f0a3','#addd8e','#78c679','#41ab5d','#238443','#005a32'))
old_par=par()
layout(matrix(c(1,1,1,2), ncol=4))

# overall limits
limits=2 * max(abs(c(graphdata$ci_lower_7, graphdata$ci_upper_7)))
# plot upper confidence 7 interval using plotrix
polar.plot(graphdata$ci_upper_7, polar.pos=graphdata$eq_angle_by_10,
            radial.lim=c(-1*limits,limits),
            labels = "", main=NULL,lwd=0.5, rp.type="p",
            show.grid.labels=FALSE, show.grid=FALSE, mar=c(0,0,0,0),
            grid.col=bands[7], line.col=bands[7], poly.col=bands[7])
# plot upper 6 confidence interval
plot_ci_round <- function(upper_bound,x){
  polar.plot(upper_bound, polar.pos=graphdata$eq_angle_by_10, add=TRUE,
             radial.lim=c(-1*limits,limits),

```

```

        line.col=bands[x], lwd=0.5, rp.type="p", poly.col=bands[x])
    }
plot_ci_round(graphdata$ci_upper_6, 6)
plot_ci_round(graphdata$ci_upper_5, 5)
plot_ci_round(graphdata$ci_upper_4, 4)
plot_ci_round(graphdata$ci_upper_3, 3)
plot_ci_round(graphdata$ci_upper_2, 2)
plot_ci_round(graphdata$ci_upper_1, 1)
plot_ci_round(graphdata$ci_lower_1, 2)
plot_ci_round(graphdata$ci_lower_2, 3)
plot_ci_round(graphdata$ci_lower_3, 4)
plot_ci_round(graphdata$ci_lower_4, 5)
plot_ci_round(graphdata$ci_lower_5, 6)
plot_ci_round(graphdata$ci_lower_6, 7)
polar.plot(graphdata$ci_lower_7, polar.pos=graphdata$eq_angle_by_10,
            add=TRUE, radial.lim=c(-1*limits,limits),
            line.col="white", lwd=0.5, rp.type="p", poly.col="white")

# plot expected guide line
polar.plot(rep(0,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10, add=TRUE,
            radial.lim=c(-1*limits,limits),
            rp.type="p", lty=4)
# plot 500 less than expected guide line
polar.plot(rep(-1 * circlesize,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10,
            add=TRUE,radial.lim=c(-1*limits,limits),
            rp.type="p", lty=1, line.col="#00000044")
# plot 500 more than expected guide line
polar.plot(rep(circlesize,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10,
            add=TRUE,radial.lim=c(-1*limits,limits),
            rp.type="p", lty=3, line.col="#00000044")

lines(c(-1.5,-1.2)*limits, c(0,0))
lines(c(1.5,1.2)*limits, c(0,0))
text(-1.8*limits,0, label="sunset
180", cex=0.7)
text(1.8*limits,0, label="sunrise
0", cex=0.7)

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE, xlab="")
lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
clrs <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
legend(0,4.5, legend=lbls, lty=typs, lwd=weights, col=clrs, bty="n", xjust=0,
       title="Confidence Intervals:", cex=0.9)

lbls2=c("Expected Number", paste(circlesize,"under expected"),

```

```
    paste(circlesize,"over expected"))
typs2=c(4,1,3)
weights2=c(1,1,1)
clrs2=c("#000000","#00000044","#00000044")
legend(0,10, legend=lbls2, lty=typs2, lwd=weights2, bty="n", xjust=0,
       title="Legend", y.intersp=1.2, col=clrs2)

par(mar=old_par$mar)
par(mfrow=c(1,1))
```

# Chapter 16

## Australia is similar

Australian earthquake data is publicly made available from Geoscience Australia. This data was obtained by searching the database for all Australia and download the result as csv, then reading the saved csv into R.

For the same period as the New Zealand Data (from September 2011 to September 2016), in earthquakes data from the Australia there are 3182 events of depth greater than 0 and magnitude greater than 0.

Of the 3182 in the data, 1911 occurred at night, a proportion of 0.6006. A seven sigma confidence interval for the proportion of earthquakes occurring at night is 0.5388 to 0.6602, so I can be confident this is a non-random difference. This very high rate is narrowly smaller than the rate observed in the Philippines.

Because of the strength of the effect in Australia, it is clearly not 50% despite the small sample size leading to large confidence intervals.

Australia seems to show a clearly high rate of night quakes in the 1 to 2 magnitude range, but due to sample size it is hard to form an opinion on if that rate is holding until 3.5 or dropping towards 50%.

By the time the earthquakes are divided into 10 degree arcs of the sun, sample sizes are a little small to say much useful about the Australian data. When the sun is in the east below the horizon seems to consistently have more earthquakes than other times, but there are not enough sample to form more detailed conclusions.

Table 16.1: Data description

feature	value
Earliest (UTC)	2011-09-01 01:56:51
Latest (UTC)	2016-08-31 16:25:18
Northernmost	-11.546
Southernmost	-43.429
Westmost	111.085
Eastmost	154.521
Percent < Mag 3	84.82
total entries	3182
nighttime quakes	1911

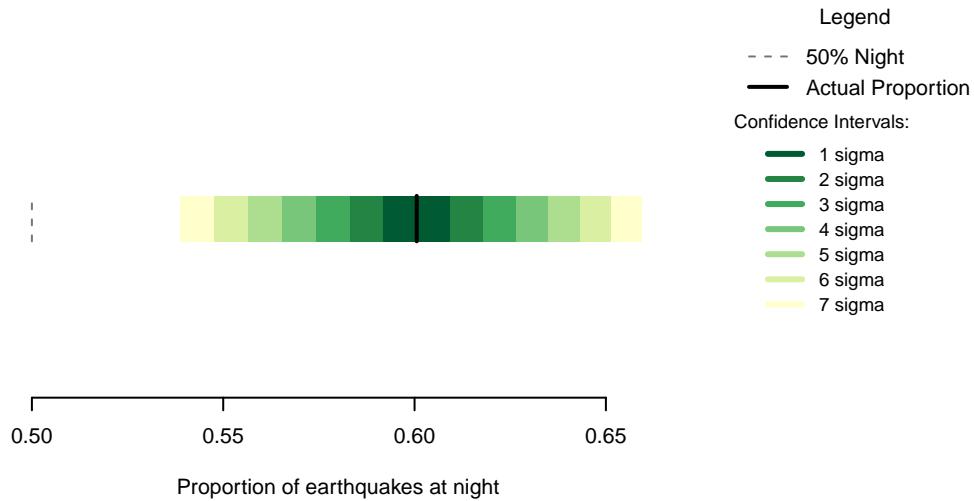


Figure 16.1: Proportion of earthquakes at night: Philippines. n=8054

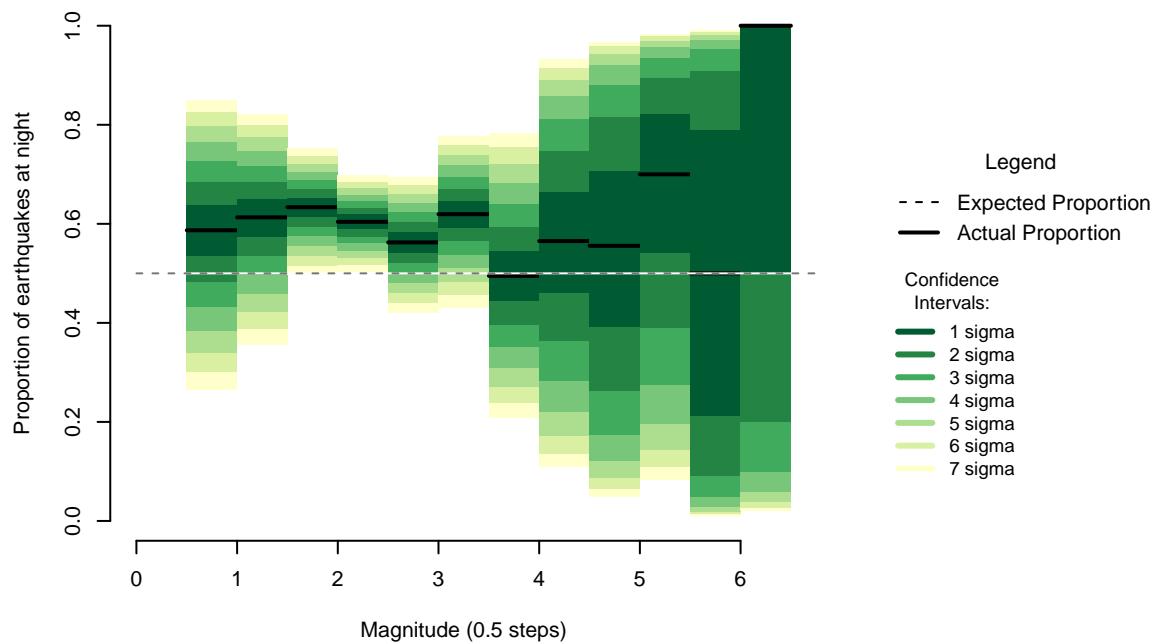


Figure 16.2: Proportion of night earthquakes by magnitude, Philippines. n=8054

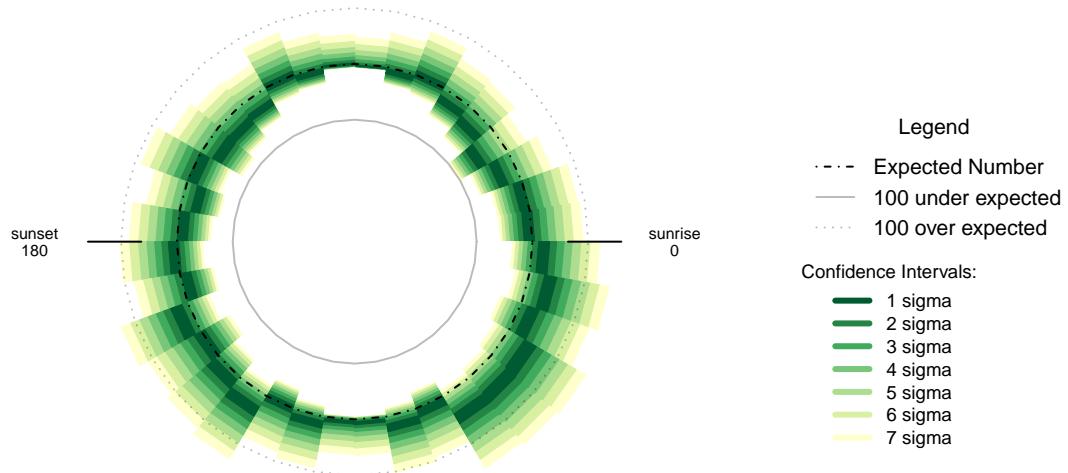


Figure 16.3: Over- and under-supply of earthquakes by angle of the sun (10 degree steps). Hawaii. n=10181

## 16.1 Formal Statement

Earthquakes in the Australia show a similar pattern to New Zealand, displaying an oversupply of earthquakes at night that is not the result of chance, and clear evidence of smaller magnitude earthquakes having a high rate of nighttime occurrence. The sample size does not permit detailed conclusions about the pattern with respect to the angle of the sun at time of earthquake.

## 16.2 Links

1 - Geoscience Australia Earthquakes: <http://www.ga.gov.au/earthquakes/searchQuake.do>

## 16.3 Chapter Code

```
## ----setup, include=FALSE-----
knitr::opts_chunk$set(echo = FALSE)

## ----c10_001, warnings=FALSE, errors=FALSE, message=FALSE-----
library(geosphere)
library(lubridate, quietly=TRUE)
library(dplyr)
library(binom)
library(ggplot2)
library(maps)
library(mapdata)
library(parallel)
library(readr)
library(plotrix)
library(tidyr)
library(maptools)
library(rvest)
Sys.setenv(TZ = "UTC")
```

```

## ----warnings=FALSE, errors=FALSE, message=FALSE-----

if(!dir.exists("../othereqdata")){
  dir.create("../othereqdata")
}
if(!file.exists("../othereqdata/eq_australia_raw.RData")){
  eq_aus <- read.csv("../australia/eq_103_1492071234067.csv", stringsAsFactors = FALSE)
  eq_aus$time_UTC <- ymd_hms(paste(eq_aus$UTC.Date, eq_aus$UTC.Time))
  names(eq_aus)[c(1,6,7,9)] <- c("magnitude",
                                    "latitude", "longitude", "depth")

  eq_national <- eq_aus %>%
    filter(magnitude >= 0 &
           time_UTC >= as.POSIXct("2011-09-01T00:00:00",
                                   format="%Y-%m-%dT%H:%M:%S", tz="UTC") &
           time_UTC < as.POSIXct("2016-09-01T00:00:00",
                                   format="%Y-%m-%dT%H:%M:%S", tz="UTC")) %>%
    distinct() %>% arrange(time_UTC)
#one longitude was missing the decimal sign

  rm(eq_aus)
  save(eq_national, file="../othereqdata/eq_australia_raw.RData")
}

## ----

if(!file.exists("../othereqdata/eq_australia_processed.RData")){
  load("../othereqdata/eq_australia_raw.RData")
  southmost <- min(eq_national$latitude)
  westmost <- min(eq_national$longitude)

  eq_national <- eq_national %>% filter(
    magnitude > 0) %>% rowwise() %>% mutate(
    eq_gridpoint_y = round(distVincentyEllipsoid(c(longitude, southmost),
                                                   c(longitude,latitude)) /50000,0),
    eq_gridpoint_x = round(distVincentyEllipsoid(c(westmost, latitude),
                                                   c(longitude,latitude)) /50000,0),
    eq_roundedlat = destPoint(p=c(longitude, southmost),
                               b=0, d=eq_gridpoint_y*50000)[2],
    eq_roundedlong = destPoint(p=c(westmost, eq_roundedlat),
                               b=90, d=eq_gridpoint_x*50000)[1]) %>% ungroup()
# use maptools to calculate solar angles
  sun_angles <- solarpes(matrix(c(eq_national$longitude, eq_national$latitude), ncol=2),
                           eq_national$time_UTC)
  colnames(sun_angles) <- c("eq_compass", "eq_vertical")
  eq_national <- cbind(eq_national,sun_angles)
  eq_national$eq_is_night <- eq_national$eq_vertical < 0
# calculate 360 degree as well as vertical

```

```

eq_national <- eq_national %>%
  mutate(eq_angle_360 = eq_vertical,
        eq_angle_360 = ifelse(eq_compass > 180, 180 - eq_angle_360, eq_angle_360),
        eq_angle_360 = ifelse(eq_vertical < 0 & eq_compass <= 180,
                               360 + eq_angle_360, eq_angle_360),
        eq_angle_by_10 = floor(eq_angle_360 /10) * 10)
save(eq_national, file="../othereqdata/eq_australia_processed.RData")
}

## -----
if(!file.exists("../othereqdata/eq_australia_expected.RData")){
  load("../othereqdata/eq_australia_processed.RData")

  lat_range <- unique(eq_national$eq_roundedlat)
  long_med <- median(eq_national$eq_roundedlong)

  # 1 minute intervals for a full solar year
  time1 <- ymd_hms("2015-01-01 00:00:00")
  time2 <- ymd_hms("2015-12-31 23:59:00")
  time_sq <- seq.POSIXt(from=time1, to=time2, by="min")

  calc_angs <- function(x, longinput, timeinput){
    library(dplyr)
    sun_angles <- maptools::solarpos(matrix(c(longinput, x), ncol=2), timeinput)
    colnames(sun_angles) <- c("eq_compass", "eq_vertical")
    # calculate 360 degree as well as vertical
    site_summary <- as.data.frame(sun_angles) %>%
      mutate(eq_angle_360 = eq_vertical,
            eq_angle_360 = ifelse(eq_compass > 180, 180 - eq_angle_360, eq_angle_360),
            eq_angle_360 = ifelse(
              eq_vertical < 0 & eq_compass <= 180, 360 + eq_angle_360, eq_angle_360),
            eq_angle_by_10 = floor(eq_angle_360 /10) * 10) %>%
      group_by(eq_angle_by_10) %>% summarise(total= n())
    site_summary$lat <- x
    return(site_summary)
  }
  #####
  # Calculate the number of cores
  no_cores <- detectCores() - 1
  # Initiate cluster
  cl <- makeCluster(no_cores)
  clusterExport(cl, varlist=c("lat_range", "long_med", "time_sq", "calc_angs"))
  list_angs <- parLapply(cl, lat_range,
                         function(x){
                           calc_angs(x=x, longinput=long_med, timeinput=time_sq)})
  stopCluster(cl)
  #####
  library(tidyr)
  anglong <- bind_rows(list_angs)
  angwide <- spread(anglong, key=eq_angle_by_10,value=total)
  rm(anglong, list_angs, time_sq)
  save(angwide, file="../othereqdata/eq_australia_expected.RData")
}

```

```

}

## -----
load("../othereqdata/eq_australia_processed.RData")
load("../othereqdata/eq_australia_expected.RData")

eq_night = sum(eq_national$eq_is_night)
eq_total = nrow(eq_national)
bands <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
sigmas <- c(0.682689492137086,
0.954499736103642,
0.997300203936740,
0.999936657516334,
0.999999426696856,
0.99999998026825,
0.99999999997440)
lbls <- c(
"1 sigma", "2 sigma",
"3 sigma", "4 sigma",
"5 sigma", "6 sigma",
"7 sigma")
typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
old_par=par()

## -----
bt <- binom.test(eq_night ,eq_total, conf.level= .99999999997440)

## -----
feature <- c("Earliest (UTC)", "Latest (UTC)",
"Northernmost", "Southernmost",
"Westmost", "Eastmost",
"Percent < Mag 3", "total entries",
"nighttime quakes")
value <- c(as.character(min(eq_national$time_UTC)),
as.character(max(eq_national$time_UTC)),
as.character(max(eq_national$latitude)),
as.character(min(eq_national$latitude)),
as.character(min(eq_national$longitude)),
as.character(max(eq_national$longitude)),
as.character(round(100*sum(eq_national$magnitude < 3)/eq_total,2)),
as.character(eq_total),
as.character(eq_night))
data.frame(feature,value) %>% knitr::kable(caption="Data description")

## ---- fig.cap="Proportion of earthquakes at night: Philippines. n=8054"----
### making the basic proportion graph
eq_night = sum(eq_national$eq_is_night)
eq_total = nrow(eq_national)
bands <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
sigmas <- c(0.682689492137086,
0.954499736103642,
0.997300203936740,

```

```

0.999936657516334,
0.999999426696856,
0.99999998026825,
0.999999999997440)

lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
typs <- c(1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3)
old_par=par()
conf_steps <- function(x, sigmas=sigmas, night=eq_night, total=eq_total){
  ci_lower <- binom.confint(night, total, method=c("wilson"), conf.level = sigmas[x])[1,5]
  ci_upper <- binom.confint(night, total, method=c("wilson"), conf.level = sigmas[x])[1,6]
  ci_data <- data.frame(step = x, ci_lower, ci_upper)
}
ci_spacing <- lapply(7:1, conf_steps, sigmas=sigmas, night=eq_night, total=eq_total)
ci_steps <- bind_rows(ci_spacing)

layout(matrix(c(1,1,1,2), ncol=4))

par(mar=c(5,6,4,2))
plot(c(min(0.5,floor(100*ci_steps[1,2])/100), max(0.5,ceiling(100*ci_steps[1,3])/100)),
     y=c(-3,8), type="n", bty="n", yaxt="n", ylab="",
     xlab="Proportion of earthquakes at night")
a <- a <- apply(ci_steps, 1, function(x){
  polygon(c(x[2], x[3], x[3], x[2]), c(0, 0, 1, 1), col=bands[x[1]], border=NA))
  lines(c(.5,.5), c(0,1), col="#FFFFFF")
  lines(c(.5,.5), c(0,1), lty=2, col="#777777")
  lines(c(eq_night/eq_total,eq_night/eq_total), c(0,1), lwd=2)

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE)
legend(0,5.5, legend=lbls, lty=typs, lwd=weights, col=bands, bty="n", xjust=0,
       title="Confidence Intervals:", y.intersp=1.1, cex=0.9)

lbls2=c("50% Night", "Actual Proportion")
typs2=c(2,1)
weights2=c(1,2)
cls2=c("#777777", "#000000")
legend(0,7, legend=lbls2, lty=typs2, lwd=weights2, col=cls2, bty="n", xjust=0,
       title="Legend", y.intersp=1.2)

par(mar=old_par$mar)
par(mfrow=c(1,1))

## ---- fig.cap="Proportion of night earthquakes by magnitude, Philippines. n=8054"----
old_par=par()

```

```

grf <- eq_national %>% mutate(floored_mag = floor(magnitude*2)/2) %>%
  group_by(floored_mag) %>% summarise(successes = sum(eq_is_night), trials=n())

poly_conf_int <- function(success, trials, aa, stepsize, sigma, colr){
  ci <- binom.confint(success, trials, method=c("wilson"), conf.level = sigma)
  lower <- ci[1,5]
  upper <- ci[1,6]
  a <- polygon(x=c(aa,aa+stepsize,aa+stepsize,aa), y=c(upper,upper,lower,lower),
                col=colr, border=NA)
}

plot7sig <- function(success, trials, aa, stepsize){
  library(binom)
  #bands <- c('#ffffb2', '#fed976', '#feb24c', '#fd8d3c', '#fc4e2a', '#e31a1c', '#b10026')
  bands <- rev(c('#ffffcc', '#d9f0a3', '#add8e6', '#78c679', '#41ab5d', '#238443', '#005a32'))
  sigmas <- c(0.682689492137086,
             0.954499736103642,
             0.997300203936740,
             0.999936657516334,
             0.999999426696856,
             0.99999998026825,
             0.99999999997440)
  sapply(7:1, function(x){
    poly_conf_int(success, trials, aa, stepsize, sigmas[x], bands[x]))}
  a <- lines(c(aa, aa + stepsize), c(success/trials, success/trials), lwd=2)
}

lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
tys <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
clrs <- rev(c('#ffffcc', '#d9f0a3', '#add8e6', '#78c679', '#41ab5d', '#238443', '#005a32'))
#clrs <- c('#ffffb2', '#fed976', '#feb24c', '#fd8d3c', '#fc4e2a', '#e31a1c', '#b10026')

layout(matrix(c(1,1,1,2), ncol=4))
plot(x=c(0,max(grf$floored_mag)+0.5), y=c(0,1), type="n", bty="n",
      xlab="Magnitude (0.5 steps)", ylab="Proportion of earthquakes at night")
a <- apply(grf,1,function(x){plot7sig(x[2],x[3],x[1],0.5)})
lines(c(0,10), c(.5,.5), col="#FFFFFF")
lines(c(0,10), c(.5,.5), lty=2, col="#777777")

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE)
legend(0.5, legend=lbls, lty=tys, lwd=weights, col=clrs, bty="n", xjust=0,
       title="Confidence
Intervals:", cex=0.9)
lbls=c("Expected Proportion", "Actual Proportion")
tys=c(2,1)

```

```

weights=c(1,2)
legend(0,7, legend=lbls, lty=typs, lwd=weights, bty="n", xjust=0,
       title="Legend", y.intersp=1.2)

par(mar=old_par$mar)
par(mfrow=c(1,1))

## -----
by_angle <- eq_national %>%
  group_by(eq_angle_by_10) %>% summarise(total= n()) %>%
  mutate(daynight=ifelse(eq_angle_by_10 < 180, "day", "night"))

merged <- merge(eq_national, angwide, by.x="eq_roundedlat", by.y="lat")
agg_expected <- merged %>% select(`0`:`350`) %>% colSums(na.rm=TRUE)
expected_prop <- agg_expected / sum(agg_expected)
expected <- data.frame(eq_angle_by_10 = as.numeric(names(expected_prop)),
                       expected_prop = as.numeric(expected_prop))
expected$expected_number = expected_prop * eq_total
act_exp <- merge(expected, by_angle, by="eq_angle_by_10", all.x=TRUE)

act_exp$total[is.na(act_exp$total)] <- 0
act_exp$daynight <- NULL
act_exp$act_prop <- act_exp$total / sum(act_exp$total)

ci_brackets <- act_exp %>% ungroup() %>% mutate(grand_total=sum(total)) %>%
  rowwise() %>% mutate(
    ci_lower_7 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[7])[1,5] * grand_total,
    ci_upper_7 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[7])[1,6] * grand_total,
    ci_lower_6 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[6])[1,5] * grand_total,
    ci_upper_6 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[6])[1,6] * grand_total,
    ci_lower_5 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[5])[1,5] * grand_total,
    ci_upper_5 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[5])[1,6] * grand_total,
    ci_lower_4 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[4])[1,5] * grand_total,
    ci_upper_4 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[4])[1,6] * grand_total,
    ci_lower_3 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[3])[1,5] * grand_total,
    ci_upper_3 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[3])[1,6] * grand_total,
    ci_lower_2 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[2])[1,5] * grand_total,
    ci_upper_2 = binom.confint(total, grand_total, method=c("wilson"),
                                conf.level = sigmas[2])[1,6] * grand_total,

```

```

ci_lower_1 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[1])[1,5] * grand_total,
ci_upper_1 = binom.confint(total, grand_total, method=c("wilson"),
                           conf.level = sigmas[1])[1,6] * grand_total)

norm_ci <- ci_brackets

for (i in c(4,7:20)){
norm_ci[,i] <- ci_brackets[,i] - ci_brackets[,3]
}
circlesize=100

## ---- fig.cap="Over- and under- supply of earthquakes by angle of the sun
## (10 degree steps). Hawaii. n=10181"----
norm_ci$border = 2
# need to double entries with a displacement of 10
# to make each side of the item on the graph
norm_ci2 <- norm_ci
norm_ci2$eq_angle_by_10 <- norm_ci2$eq_angle_by_10 + 10
norm_ci2$border = 1
graphdata <- bind_rows(norm_ci,norm_ci2) %>% arrange(eq_angle_by_10,border)

##### plot graph
bands <- rev(c('#ffffcc','#d9f0a3','#addd8e','#78c679','#41ab5d','#238443','#005a32'))
old_par=par()
layout(matrix(c(1,1,1,2), ncol=4))

# overall limits
limits=2 * max(abs(c(graphdata$ci_lower_7, graphdata$ci_upper_7)))
# plot upper confidence 7 interval using plotrix
polar.plot(graphdata$ci_upper_7, polar.pos=graphdata$eq_angle_by_10,
            radial.lim=c(-1*limits,limits),
            labels = "", main=NULL,lwd=0.5, rp.type="p",
            show.grid.labels=FALSE, show.grid=FALSE, mar=c(0,0,0,0),
            grid.col=bands[7], line.col=bands[7], poly.col=bands[7])
# plot upper 6 confidence interval
plot_ci_round <- function(upper_bound,x){
  polar.plot(upper_bound, polar.pos=graphdata$eq_angle_by_10, add=TRUE,
             radial.lim=c(-1*limits,limits),
             line.col=bands[x], lwd=0.5, rp.type="p", poly.col=bands[x])
}

plot_ci_round(graphdata$ci_upper_6, 6)
plot_ci_round(graphdata$ci_upper_5, 5)
plot_ci_round(graphdata$ci_upper_4, 4)
plot_ci_round(graphdata$ci_upper_3, 3)
plot_ci_round(graphdata$ci_upper_2, 2)
plot_ci_round(graphdata$ci_upper_1, 1)
plot_ci_round(graphdata$ci_lower_1, 2)
plot_ci_round(graphdata$ci_lower_2, 3)
plot_ci_round(graphdata$ci_lower_3, 4)
plot_ci_round(graphdata$ci_lower_4, 5)
plot_ci_round(graphdata$ci_lower_5, 6)
plot_ci_round(graphdata$ci_lower_6, 7)

```

```

polar.plot(graphdata$ci_lower_7, polar.pos=graphdata$eq_angle_by_10,
           add=TRUE, radial.lim=c(-1*limits,limits),
           line.col="white", lwd=0.5, rp.type="p", poly.col="white")

# plot expected guide line
polar.plot(rep(0,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10, add=TRUE,
           radial.lim=c(-1*limits,limits),
           rp.type="p", lty=4)
# plot 500 less than expected guide line
polar.plot(rep(-1 * circlesize,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10,
           add=TRUE,radial.lim=c(-1*limits,limits),
           rp.type="p", lty=1, line.col="#00000044")
# plot 500 more than expected guide line
polar.plot(rep(circlesize,nrow(graphdata)), polar.pos=graphdata$eq_angle_by_10,
           add=TRUE,radial.lim=c(-1*limits,limits),
           rp.type="p", lty=3, line.col="#00000044")

lines(c(-1.5,-1.2)*limits, c(0,0))
lines(c(1.5,1.2)*limits, c(0,0))
text(-1.8*limits,0, label="sunset
180", cex=0.7)
text(1.8*limits,0, label="sunrise
0", cex=0.7)

par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE, xlab="")
lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
clrs <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
legend(0,4.5, legend=lbls, lty=typs, lwd=weights, col=clrs, bty="n", xjust=0,
       title="Confidence Intervals:", cex=0.9)

lbls2=c("Expected Number", paste(circlesize,"under expected"),
       paste(circlesize,"over expected"))
typs2=c(4,1,3)
weights2=c(1,1,1)
clrs2=c("#000000", "#00000044", "#00000044")
legend(0,10, legend=lbls2, lty=typs2, lwd=weights2, bty="n", xjust=0,
       title="Legend", y.intersp=1.2, col=clrs2)

par(mar=old_par$mar)
par(mfrow=c(1,1))

```



# Chapter 17

## It is gravity

I can assert that no other mechanism than the sun's gravity (relative to the earth's gravity) makes sense given all the results I have gotten. When the sun is below the horizon, the gravity of the sun (though tiny compared to that of the earth) is added to the earth's gravity pulling bits of crust down into other bits of crust. When the sun is above the horizon, it is mitigating the gravity of the earth by pulling against it. The daily pulling in opposite directions is, in effect, a high-speed vibration in geological time.

I can point out that the forces involved when the sun is below the horizon but not directly in line with the centre of the earth bear a strong resemblance to the forces expressed geologically. When the sun is below the horizon at night, this is a general compression state similar to subduction zones, however the sun being to the side is creating a second vector of force. The angular imbalance is the same kind of effect as that observed in thrust earthquakes, where two sections of crust either side of a fault come into contact at different angles. I can also note that nighttime earthquakes seem to be more frequent in countries which have thrust faults in subduction zones, and suggest that the presence of such areas is a determining factor in the frequency of nighttime earthquakes.

I can establish that there are a number of different ways solar gravity can act- as daily compression, as an angular force relative to the earth's gravity, and as a component of the solid earth tide. I can then say any explanation should explain all the different highly statistically significant observations, and the response of local geological conditions to changing solar gravity does so.

But as a "last explanation standing" I am not proving this, I am saying this alternative hypothesis is consistent with the facts and disproving all the alternatives. So to strengthen the argument that gravity did it, I came up one more test. This test is for such an extremely specific set of circumstances that I can think of no possible explanation other than gravity as the alternative hypothesis.

### 17.1 Gravitational force differentials

The solar gravitational force creates a natural force differential across arcs of the earth's crust. At any point on the earth's crust, the sun is at an angle to that point relative to the crust. This angle to the gravitational source has a horizontal and vertical component of force. At the same time, the crust to the east or west of this point is at a different angle to the sun with different horizontal and vertical forces. The epicentre of an earthquake can be thought of as a weak point along an arc of the earth experiencing force differentials.

For a specific example, the New Zealand earthquake event with public ID 2012p571580, occurred on 2012-07-30 11:53:59 UTC at latitude -38.36902 and longitude 178.5677. At the moment of the earthquake the sun was at a vertical angle of -70.075 degrees relative to a point on the earth's surface at the epicentre. 500 kilometres to the east the angle to the sun was 1.652 degrees further from the horizon than at the epicentre. 500 kilometres to the west the angle to the sun was 2.613 degrees close to the horizon than at the epicentre.

Table 17.1: Number of earthquakes by sun angle differential decile

decile	night	day	combined
1	5775 (10.27%)	4253 (9.66%)	10010 (9.98%)
2	5883 (10.46%)	4160 (9.45%)	10045 (10.02%)
3	5895 (10.48%)	4201 (9.54%)	10087 (10.06%)
4	5997 (10.66%)	4280 (9.72%)	10255 (10.23%)
5	5854 (10.41%)	4416 (10.03%)	10146 (10.12%)
6	5434 (9.66%)	4472 (10.16%)	9967 (9.94%)
7	5167 (9.19%)	4511 (10.24%)	9650 (9.62%)
8	5263 (9.36%)	4636 (10.53%)	9748 (9.72%)
9	5541 (9.85%)	4488 (10.19%)	10023 (10%)
10	5437 (9.67%)	4616 (10.48%)	10348 (10.32%)

This gives an asymmetry of degrees between the crust either side of the epicentre. This asymmetry means that there is an imbalance in the horizontal and vertical components of the sun's gravity along the crust either side of the epicentre.

As the sun's orbit is not purely circular, for a given east-west arc of crust the difference in the angle to the sun fluctuates through time. If this change in gravity force is not associated with a difference in earthquake frequency, then that is evidence against the gravity hypothesis. If there is a difference in earthquake frequency on the basis of the asymmetry of the sun's angle to the earth's crust, then that is highly specific support for gravity as a mechanism.

To test this I am taking a similar approach to when I tested the angle of the sun at time of earthquake. For each minute of a solar year, for each 50 kilometre grid point in the data, I calculate the difference in size of the sun's angle between 500 kilometres east and 500 kilometres west of the point. For this angle (and thus force) differentials, due to the size of the data, I calculate the decile steps for force differentials in that area from least to most imbalanced. These are effectively the deciles of "opportunities to have an earthquake in, based on force differential"

I then calculate the force differential at the time and place of the earthquake, and see which decile the earthquake falls into for the area of the earthquake. If there is no relationship between earthquakes, then approximately 10% of earthquakes should fall into each decile. As there are more earthquakes at night, I am separating the tests into day earthquakes compared to potential daytime gravity imbalance and night earthquakes compared to potential nighttime gravity imbalance. Because other evidence has indicated different night and day patterns, I thought might learn more by separating the two.

A chi squared test of both the night and day together gives a p-value of  $8.4 \times 10^{-32}$ , a chi squared test of the night alone gives a p-value of  $3.7 \times 10^{-25}$ , and a chi squared test of just the day values gives a p-value of  $1.9 \times 10^{-9}$ . As a result we can say that there are overall difference and differences within the night and within the day.

While only a few of the deciles are overwhelming different to the expected amount by themselves, having so many deciles different to expected is very improbable to have occurred by chance. So I can conclude that gravity, as expressed through differential force along arcs of the earth, is related to earthquake frequency.

From the graph, the pattern between the deciles seems non-random. The effects of the gravitational differential are reversed between day and night, as the two sets of data are mirroring each other with respect to their relationships to the expected values. From the perspective of gravity having something to do with it, that is indicated by the non-random pattern. The complexity of the mirroring suggests this effect is also influenced by other factors.

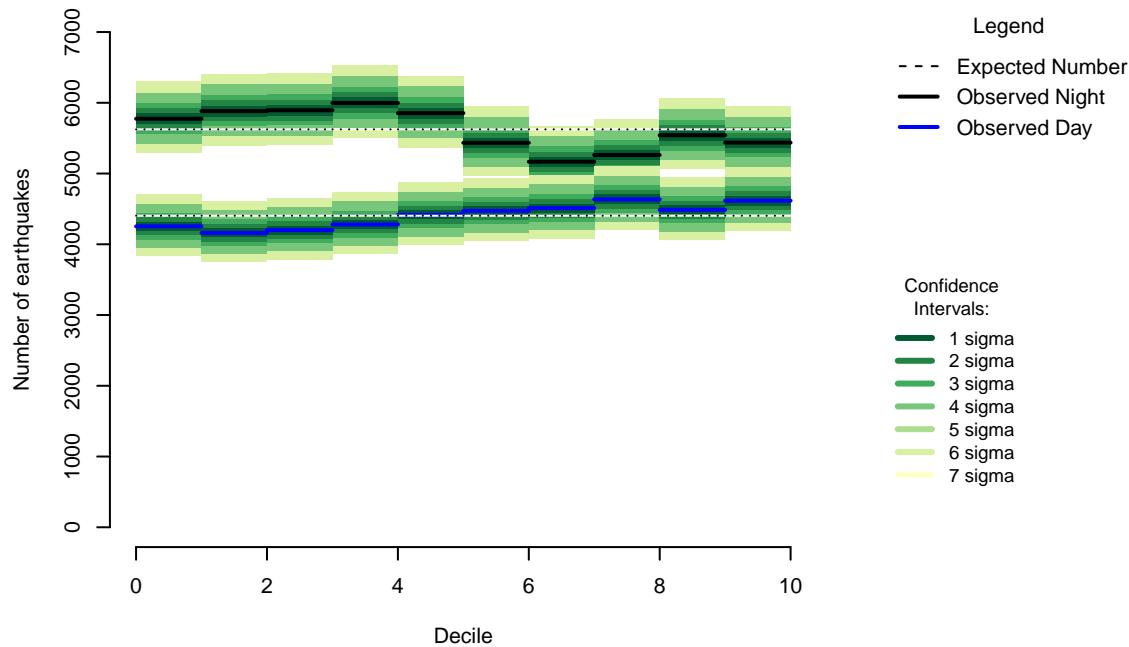


Figure 17.1: Distribution of Earthquakes by Gravitational Imbalance in east/west arc of crust

## 17.2 Formal Statement

Local geological response to solar gravitational effects is consistent with all the evidence.

## 17.3 Chapter Code

```
## ----setup, include=FALSE-----
knitr:::opts_chunk$set(echo = FALSE)
knitr:::opts_chunk$set(warnings=FALSE)
knitr:::opts_chunk$set(errors=FALSE)
knitr:::opts_chunk$set(message=FALSE)
opts_chunk$set(tidy.opts=list(width.cutoff=80),tidy=TRUE)
knitr:::opts_chunk$set(dpi = 150)
knitr:::opts_chunk$set(fig.width =  6)
knitr:::opts_chunk$set(fig.height =  4)

## ----c002_libraries-----
Sys.setenv(TZ = "UTC")
library(dplyr)
library(ggplot2)
library(lubridate)
library(maptools)
library(binom)
library(parallel)
library(plotrix)
library(solidearthtide)
library(geosphere)
library(tidyr)
```

```

# Assumes there is eqnz_processed data created in chapter 2
load("eqdata/eqnz_processed.RData")
old_par <- par()

lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
typs <- c(1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3)
clrs <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))

## -----
if (!file.exists("eqdata/eqnz_offsets.RData")){
  eqnz <- eqnz %>% mutate %>% rowwise() %>%
    mutate(eq_long_eastern = destPoint(p=c(longitude, latitude), b=90, d=500000)[1],
          eq_long_western = destPoint(p=c(longitude, latitude), b=-90, d=500000)[1]) %>%
    ungroup() %>%
    mutate(eq_e_angle = solarpos(matrix(c(eq_long_eastern,latitude), ncol=2), time_UTC)[,2],
          eq_w_angle = solarpos(matrix(c(eq_long_western,latitude), ncol=2), time_UTC)[,2],
          eq_vdiff = abs(eq_w_angle - eq_e_angle))

  # collective for country

  med_long <- median(eqnz$eq_roundedlong)
  range_lat <- unique(eqnz$eq_roundedlat)
  range_east <- sapply(range_lat, function(x){
    destPoint(p=c(med_long, x), b=90, d=500000)[1]})
  range_west <- sapply(range_lat, function(x){
    destPoint(p=c(med_long, x), b=-90, d=500000)[1]})

  # 1 minute intervals for a full solar year
  time1 <- ymd_hms("2015-01-01 00:00:00")
  time2 <- ymd_hms("2015-12-31 23:59:00")
  time_sq <- seq.POSIXt(from=time1, to=time2, by="min")

  # each combination of latitude point and time
  expected_sun <- data.frame(latitude = rep(range_lat, each=length(time_sq)),
                               long_east = rep(range_east, each=length(time_sq)),
                               long_west = rep(range_west, each=length(time_sq)),
                               time_UTC = rep(time_sq, times=length(range_lat)))
  expected_sun$longitude <- med_long

  #calc sun angles just like with earthquake data
  sun_angles <- solarpos(matrix(
    c(expected_sun$longitude,expected_sun$latitude), ncol=2), expected_sun$time_UTC)[,2]
  east_angles <- solarpos(matrix(
    c(expected_sun$long_east,expected_sun$latitude), ncol=2), expected_sun$time_UTC)[,2]
  west_angles <- solarpos(matrix(
    c(expected_sun$long_west,expected_sun$latitude), ncol=2), expected_sun$time_UTC)[,2]

```

```

expected_sun$eq_is_night <- sun_angles < 0
expected_sun$eq_v_offset_expected <- abs(west_angles - east_angles)

rm(sun_angles, east_angles, west_angles)

# To big to database merge and do deciles at once
# but we can ask the elegant question, for each location, what decile is the earthquake.
# So we calculate decile thresholds for each location, then merge and size is only * 10-ish

all_dec_thres <- expected_sun %>% group_by(latitude) %>%
  mutate(deciles = ntile(eq_v_offset_expected, 10)) %>%
  group_by(latitude, deciles) %>%
  summarise(thres_value = max(eq_v_offset_expected)) %>%
  ungroup() %>% mutate(thres_value = ifelse(deciles==10,700,thres_value))

day_dec_thres <- expected_sun %>% filter(!eq_is_night) %>% group_by(latitude) %>%
  mutate(deciles = ntile(eq_v_offset_expected, 10)) %>%
  group_by(latitude, deciles) %>%
  summarise(thres_value = max(eq_v_offset_expected)) %>%
  ungroup() %>% mutate(thres_value = ifelse(deciles==10,700,thres_value))

ngt_dec_thres <- expected_sun %>% filter(eq_is_night) %>% group_by(latitude) %>%
  mutate(deciles = ntile(eq_v_offset_expected, 10)) %>%
  group_by(latitude, deciles) %>%
  summarise(thres_value = max(eq_v_offset_expected)) %>%
  ungroup() %>% mutate(thres_value = ifelse(deciles==10,700,thres_value))

rm(expected_sun)

meq <- merge(eqnz,all_dec_thres, by.x="eq_roundedlat", by.y="latitude")
decall <- meq %>% filter(eq_vdiff < thres_value) %>% group_by(publicid) %>%
  arrange(deciles) %>% slice(1) %>% ungroup() %>% select(publicid, deciles)
rm(all_dec_thres, meq)
##
meq <- merge(eqnz,ngt_dec_thres, by.x="eq_roundedlat", by.y="latitude")
decngt<- meq %>% filter(eq_vdiff < thres_value) %>% group_by(publicid) %>%
  arrange(deciles) %>% slice(1) %>% ungroup() %>% select(publicid, deciles)
rm(ngt_dec_thres, meq)
##
meq <- merge(eqnz,day_dec_thres, by.x="eq_roundedlat", by.y="latitude")
decday<- meq %>% filter(eq_vdiff < thres_value) %>% group_by(publicid) %>%
  arrange(deciles) %>% slice(1) %>% ungroup() %>% select(publicid, deciles)
rm(day_dec_thres, meq)

names(decall)[2] <- "decile_overall"
names(decngt)[2] <- "decile_night"
names(decday)[2] <- "decile_day"

eqnz <- merge(eqnz,decall, by="publicid")
eqnz <- merge(eqnz,decngt, by="publicid")
eqnz <- merge(eqnz,decday, by="publicid")

```

```

save(eqnz, file="eqdata/eqnz_offsets.RData")
rm(list=ls())
}

load("eqdata/eqnz_offsets.RData")
examp <- eqnz %>% filter(publicid == "2012p571580") %>%
  select(eq_vertical, eq_e_angle, eq_w_angle)

## -----
all <- eqnz %>% group_by(decile_overall) %>%
  summarise(subtotal = n()) %>% ungroup() %>%
  mutate(percent = round(100* subtotal/sum(subtotal),2),
         combined = paste(subtotal, " (",percent,"%)", sep=""))

dy <- eqnz %>% filter(!eq_is_night) %>% group_by(decile_day) %>%
  summarise(subtotal = n()) %>% ungroup() %>%
  mutate(percent = round(100* subtotal/sum(subtotal),2),
         day = paste(subtotal, " (",percent,"%)", sep=""))

nt <- eqnz %>% filter(eq_is_night) %>% group_by(decile_night) %>%
  summarise(subtotal = n()) %>% ungroup() %>%
  mutate(percent = round(100* subtotal/sum(subtotal),2),
         night = paste(subtotal, " (",percent,"%)", sep=""))

deciles <- data.frame(decile= nt$decile_night, night=nt$night)
deciles$day <- dy$day
deciles$combined <- all$combined
knitr::kable(deciles, caption="Number of earthquakes by sun angle differential decile")

## -----
df <- data.frame(day=dy$subtotal, night=nt$subtotal)
ntvsdy_chi <- chisq.test(df)
nt_chi <- chisq.test(nt$subtotal)
dy_chi <- chisq.test(dy$subtotal)
dy_chi <- chisq.test(dy$subtotal)

## ---- fig.cap="Distribution of Earthquakes by Gravitational Imbalance in
##   east/west arc of crust"----

sigmas <- c(0.682689492137086,
0.954499736103642,
0.997300203936740,
0.999936657516334,
0.999999426696856,
0.99999998026825,
0.999999999997440)

ci_brackets_nt <- nt %>% ungroup() %>% mutate(grand_total=sum(subtotal)) %>%
  rowwise() %>%
  mutate(ci_lower_7 = binom.confint(subtotal, grand_total, method=c("wilson"),
                                    conf.level = sigmas[7])[1,5] * grand_total,
        ci_upper_7 = binom.confint(subtotal, grand_total, method=c("wilson"),
                                    conf.level = sigmas[7])[1,6] * grand_total,

```

```

ci_lower_6 = binom.confint(subtotal, grand_total, method=c("wilson"),
                           conf.level = sigmas[6])[1,5] * grand_total,
ci_upper_6 = binom.confint(subtotal, grand_total, method=c("wilson"),
                           conf.level = sigmas[6])[1,6] * grand_total,
ci_lower_5 = binom.confint(subtotal, grand_total, method=c("wilson"),
                           conf.level = sigmas[5])[1,5] * grand_total,
ci_upper_5 = binom.confint(subtotal, grand_total, method=c("wilson"),
                           conf.level = sigmas[5])[1,6] * grand_total,
ci_lower_4 = binom.confint(subtotal, grand_total, method=c("wilson"),
                           conf.level = sigmas[4])[1,5] * grand_total,
ci_upper_4 = binom.confint(subtotal, grand_total, method=c("wilson"),
                           conf.level = sigmas[4])[1,6] * grand_total,
ci_lower_3 = binom.confint(subtotal, grand_total, method=c("wilson"),
                           conf.level = sigmas[3])[1,5] * grand_total,
ci_upper_3 = binom.confint(subtotal, grand_total, method=c("wilson"),
                           conf.level = sigmas[3])[1,6] * grand_total,
ci_lower_2 = binom.confint(subtotal, grand_total, method=c("wilson"),
                           conf.level = sigmas[2])[1,5] * grand_total,
ci_upper_2 = binom.confint(subtotal, grand_total, method=c("wilson"),
                           conf.level = sigmas[2])[1,6] * grand_total,
ci_lower_1 = binom.confint(subtotal, grand_total, method=c("wilson"),
                           conf.level = sigmas[1])[1,5] * grand_total,
ci_upper_1 = binom.confint(subtotal, grand_total, method=c("wilson"),
                           conf.level = sigmas[1])[1,6] * grand_total)
ci_brackets_nt$night <- 0 #quick fix for being a character column

ci_brackets_dy <- dy %>% ungroup() %>% mutate(grand_total=sum(subtotal)) %>%
  rowwise() %>%
  mutate(ci_lower_7 = binom.confint(subtotal, grand_total, method=c("wilson"),
                                    conf.level = sigmas[7])[1,5] * grand_total,
         ci_upper_7 = binom.confint(subtotal, grand_total, method=c("wilson"),
                                    conf.level = sigmas[7])[1,6] * grand_total,
         ci_lower_6 = binom.confint(subtotal, grand_total, method=c("wilson"),
                                    conf.level = sigmas[6])[1,5] * grand_total,
         ci_upper_6 = binom.confint(subtotal, grand_total, method=c("wilson"),
                                    conf.level = sigmas[6])[1,6] * grand_total,
         ci_lower_5 = binom.confint(subtotal, grand_total, method=c("wilson"),
                                    conf.level = sigmas[5])[1,5] * grand_total,
         ci_upper_5 = binom.confint(subtotal, grand_total, method=c("wilson"),
                                    conf.level = sigmas[5])[1,6] * grand_total,
         ci_lower_4 = binom.confint(subtotal, grand_total, method=c("wilson"),
                                    conf.level = sigmas[4])[1,5] * grand_total,
         ci_upper_4 = binom.confint(subtotal, grand_total, method=c("wilson"),
                                    conf.level = sigmas[4])[1,6] * grand_total,
         ci_lower_3 = binom.confint(subtotal, grand_total, method=c("wilson"),
                                    conf.level = sigmas[3])[1,5] * grand_total,
         ci_upper_3 = binom.confint(subtotal, grand_total, method=c("wilson"),
                                    conf.level = sigmas[3])[1,6] * grand_total,
         ci_lower_2 = binom.confint(subtotal, grand_total, method=c("wilson"),
                                    conf.level = sigmas[2])[1,5] * grand_total,
         ci_upper_2 = binom.confint(subtotal, grand_total, method=c("wilson"),
                                    conf.level = sigmas[2])[1,6] * grand_total,
         ci_lower_1 = binom.confint(subtotal, grand_total, method=c("wilson"),
                                    conf.level = sigmas[1])[1,5] * grand_total)

```

```

conf.level = sigmas[1])[1,5] * grand_total,
ci_upper_1 = binom.confint(subtotal, grand_total, method=c("wilson"),
                           conf.level = sigmas[1])[1,6] * grand_total)
ci_brackets_dy$day <- 0 #quick fix for being a character column

bands <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
old_par=par()
layout(matrix(c(1,1,1,2), ncol=4))
plot(c(0,10),c(0,7000), bty="n", type="n",
     ylab="Number of earthquakes", xlab="Decile")
a <- apply(ci_brackets_nt, 1, function(x){
  polygon(c(x[1] -1, x[1], x[1], x[1]-1), c(x[6],x[6], x[7],x[7]),
          col=bands[6], border=NA)})
a <- apply(ci_brackets_nt, 1, function(x){
  polygon(c(x[1] -1, x[1], x[1], x[1]-1), c(x[8],x[8], x[9],x[9]),
          col=bands[6], border=NA)})
a <- apply(ci_brackets_nt, 1, function(x){
  polygon(c(x[1] -1, x[1], x[1], x[1]-1), c(x[10],x[10], x[11],x[11]),
          col=bands[4], border=NA)})
a <- apply(ci_brackets_nt, 1, function(x){
  polygon(c(x[1] -1, x[1], x[1], x[1]-1), c(x[12],x[12], x[13],x[13]),
          col=bands[4], border=NA)})
a <- apply(ci_brackets_nt, 1, function(x){
  polygon(c(x[1] -1, x[1], x[1], x[1]-1), c(x[14],x[14], x[15],x[15]),
          col=bands[3], border=NA)})
a <- apply(ci_brackets_nt, 1, function(x){
  polygon(c(x[1] -1, x[1], x[1], x[1]-1), c(x[16],x[16], x[17],x[17]),
          col=bands[2], border=NA)})
a <- apply(ci_brackets_nt, 1, function(x){
  polygon(c(x[1] -1, x[1], x[1], x[1]-1), c(x[18],x[18], x[19],x[19]),
          col=bands[1], border=NA)})
a <- apply(ci_brackets_nt, 1, function(x){
  lines(c(x[1]-1, x[1]), c(x[2],x[2]), lwd=2)})
expect <- sum(ci_brackets_nt$subtotal)/10
a <- apply(ci_brackets_nt, 1, function(x){
  lines(c(x[1]-1, x[1]), c(expect,expect), col="white")})
a <- apply(ci_brackets_nt, 1, function(x){
  lines(c(x[1]-1, x[1]), c(expect,expect), lty=3)})

a <- apply(ci_brackets_dy, 1, function(x){
  polygon(c(x[1] -1, x[1], x[1], x[1]-1), c(x[6],x[6], x[7],x[7]),
          col=bands[6], border=NA)})
a <- apply(ci_brackets_dy, 1, function(x){
  polygon(c(x[1] -1, x[1], x[1], x[1]-1), c(x[8],x[8], x[9],x[9]),
          col=bands[6], border=NA)})
a <- apply(ci_brackets_dy, 1, function(x){
  polygon(c(x[1] -1, x[1], x[1], x[1]-1), c(x[10],x[10], x[11],x[11]),
          col=bands[4], border=NA)})
a <- apply(ci_brackets_dy, 1, function(x){
  polygon(c(x[1] -1, x[1], x[1], x[1]-1), c(x[12],x[12], x[13],x[13]),
          col=bands[4], border=NA)})
a <- apply(ci_brackets_dy, 1, function(x){
  polygon(c(x[1] -1, x[1], x[1], x[1]-1), c(x[14],x[14], x[15],x[15]),
          col=bands[4], border=NA)})

```

```

        col=bands[3], border=NA)})}

a <- apply(ci_brackets_dy, 1, function(x){
  polygon(c(x[1]-1, x[1], x[1], x[1]-1), c(x[16],x[16], x[17],x[17]),
           col=bands[2], border=NA)})
a <- apply(ci_brackets_dy, 1, function(x){
  polygon(c(x[1]-1, x[1], x[1], x[1]-1), c(x[18],x[18], x[19],x[19]),
           col=bands[1], border=NA)})
a <- apply(ci_brackets_dy, 1, function(x){
  lines(c(x[1]-1, x[1]), c(x[2],x[2]), lwd=2, col="blue"))}
expect <- sum(ci_brackets_dy$subtotal)/10
a <- apply(ci_brackets_dy, 1, function(x){
  lines(c(x[1]-1, x[1]), c(expect,expect), col="white"))}
a <- apply(ci_brackets_dy, 1, function(x){
  lines(c(x[1]-1, x[1]), c(expect,expect), lty=3)})


par(mar=c(0,0,0,0))
plot(x=c(0,10), y=c(0,10), type="n", bty="n", axes=FALSE)

lbls <- c(
  "1 sigma", "2 sigma",
  "3 sigma", "4 sigma",
  "5 sigma", "6 sigma",
  "7 sigma")
typs <- c(1,1,1,1,1,1,1)
weights <- c(3,3,3,3,3,3,3)
clrs <- rev(c('#ffffcc', '#d9f0a3', '#addd8e', '#78c679', '#41ab5d', '#238443', '#005a32'))
legend(0,5, legend=lbls, lty=typs, lwd=weights, col=clrs, bty="n", xjust=0,
       title="Confidence\nIntervals:", cex=0.9)

lbls=c("Expected Number", "Observed Night", "Observed Day")
typs=c(2,1,1)
weights=c(1,2,2)
clrs=c("black","black","blue")
legend(0,9, legend=lbls, lty=typs, lwd=weights, col=clrs,bty="n", xjust=0,
       title="Legend", y.intersp=1.2)

par(mar=old_par$mar)
par(mfrow=c(1,1))

```