05 Mapping Diesease

(05) Mapping Disease

Preamble

```
# Preamble

## Install Pacman
load.pac <- function() {

  if(require("pacman")){
    library(pacman)
  }else{
    install.packages("pacman")
    library(pacman)
  }

  pacman::p_load(xts, sp, gstat, ggplot2, rmarkdown, reshape2, ggmap,
                 parallel, dplyr, plotly, tidyverse, reticulate, UsingR, Rmpfr,
                 swirl, corrplot, gridExtra, mise, latex2exp, tree, rpart, lattice,
                 coin, primes)


   mise()
}

load.pac()
```

```
load(file = "~/Notes/DataSci/ThinkingAboutData/TAD.rdata")
load(file = "./TAD.rdata")
```

```
knitr::opts_chunk$set(
  fig.path = "./figure/"
)
```

Binomial Distribution

A random variable is Binomial if it represents the count of the number of successes from n trials and p is the probability of success.

Simulation and Probabilities

In order to simulate, for example, a coin toss:

```
library(tidyverse)
sample(c("H", "T"), size = 3, replace = TRUE, prob = c(1,1))
```

```
## [1] "T" "T" "H"
```

```r
## Count the number of heads
sum(sample(c("H", "T"), size = 3, replace = TRUE, prob = c(1,1))=="H")
```

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

```r
## This can be automated with `rbinom`
rbinom(n = 2,size = 3, prob = 0.5)
```

```
## [1] 2 1
```

where:

- The output is the number of Successes
- `size` is the size of teh experiment, i.e. the number of repetitions (i.e. coin flips)
- `n` is how many numbers you want to get back, each corresponding to a repeated simulation.

We can also use counting formulas like $\binom{m}{n}$ via `choose(m,n)`. See Counting Formulas generally.

Density

The density rather than the count of the binomial distribution can also be simulated.

So for example the probability of getting 0, 1, 2 or 3 heads out of 3 coin tosses is:

```r
dbinom(x = 1:3, size = 3, prob = 0.5)
```

```
## [1] 0.375 0.375 0.125
```

Mean and Variance

In theory the summary statistics of a binomial distribution are:

	Mean	Variance
Binomial Distribution	np	$np(1-p)$

Where:

- n is the number of repetitions
  - in ***R*** this is `size` because `n` is already being used for the output vector length.

This can be verified by doing:

```r
primes::generate_primes(min = 12, max = 99) # Two primes ensures p, n and (1-p)
```

```
##  [1] 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
```

```r
                                    # are relatively prime
n <- 29
p <- 0.43

(rbinom(n = 100000, size = n, prob = p) %>% mean() /p) %>% signif(2)
```

```
## [1] 29
```

```r
(rbinom(n = 10000, size = n, prob = p) %>% var()  / (n * (1-p)))  %>% signif(2)
```

```
## [1] 0.42
```

Hypothesis test for a Difference in Proportional

Let's say that we make the following observation:

- *Trump County* has 12/100 with a disease
- *Clinton County* has 188/1000 with a disease

A Chi-Square Distribution can be used to compare the proportions.

```
## Make a Table
dis_df <- data.frame("Trump" = c(12, 100-12), "Clinton" = c(188, 1000-188))
dis_mat <- as.matrix(dis_df)

dimnames(dis_mat) <- list(
    c("Disease", "No Disease"),
    c("Trump", "Clinton")
)
dis_mat
```

```
##            Trump Clinton
## Disease       12     188
## No Disease    88     812
```

```
## Perform the Chi Test
chisq.test(dis_mat)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  dis_mat
## X-squared = 2.3872, df = 1, p-value = 0.1223
```

```
chisq.test(t(dis_mat))
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  t(dis_mat)
## X-squared = 2.3872, df = 1, p-value = 0.1223
```

This shows the probability of rejecting the null hypothesis when it is true (i.e. asserting that there is a difference between counties when there is in fact not) is still too high of a risk at 12%, hence there is not enough evidence to reject the null hypothesis that there is no difference between counties.

Pesticide Question

If a type of pesticide:

- kills 13 out of 20 male budworms
- kills 10 out of 20 female budworks

1. The proportion of each gender killed?
2. Is there evidence the proportion killed by this pesticide at this dose is different for each gender?

```
male <- c(13, 20-13)
female <- c(10, 20-10)

bug <- matrix(c(male, female), ncol = 2)
```

```r
dimnames(bug) <- list(c("dead", "alive"), c("male", "female"))
bug
```

```
##       male female
## dead    13     10
## alive    7     10
```

```r
chisq.test(bug)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  bug
## X-squared = 0.40921, df = 1, p-value = 0.5224
```

1. The Proportion of Male Budworms killed is:

- $\frac{13}{20} = 0.65$ for Males
- $\frac{10}{20} = 0.5$ for Females

2. In this case, the probability of concluding that there is a difference between genders when there is in fact no difference is sufficiently small (p=5.2) to reject the null hypothesis and assert that there is indeed a difference between genders.
    - The sample size is quite small so a larger sample is justified.

Poisson Distribution

The *Poisson* distribution is appropriate where values are:

- integer values that may occur in interval of time
    - e.g. the number of call outs completed in one day
- Events are independent, i.e. the occurrence of one event does not effect the probability of another event
- The average rate of events occurring is independent from other occurences
- Events cannot overlap
    - So the number of callouts in one day is fine
    - The number of phone calls isn't because if two calls are recieved at the same time they either overlap or the other phone call is rejected in lieu of the first one meaning that the phone calls are not independent.

Horse Kicks

The number of deaths caused by a horse kick in a given regiment per year is (Bortkiewics, 1898):

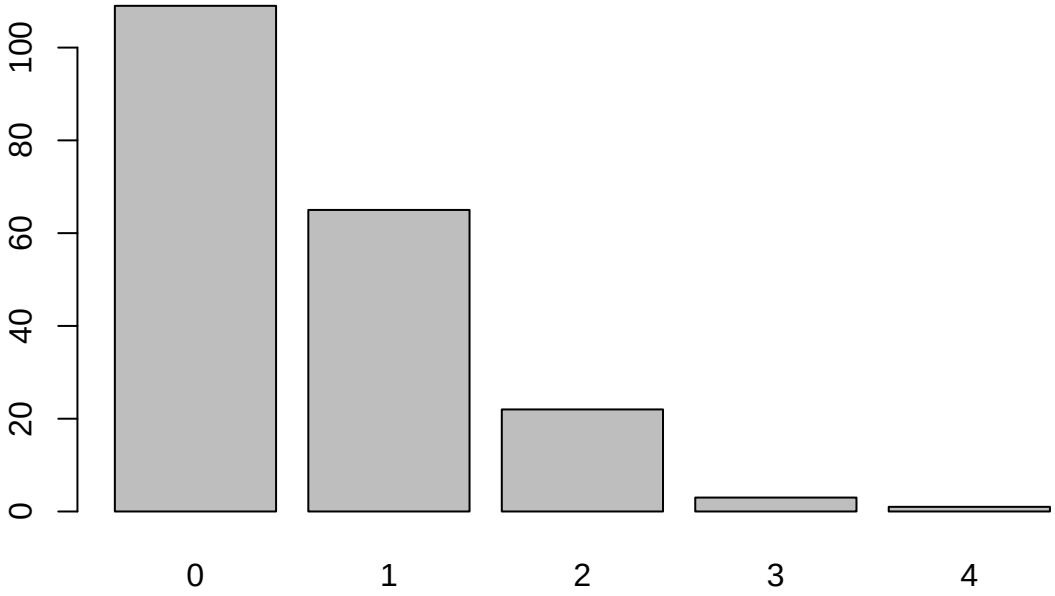
Deaths per Year Per Regiment	Deaths	Poisson Expectation ($\frac{\lambda^k d^{-\lambda}}{k!}$)
0	109	108.7
1	65	66.3
2	22	20.2
3	3	4.1
4	1	0.6
5+	0	0.1

This can be plotted in ***R***:

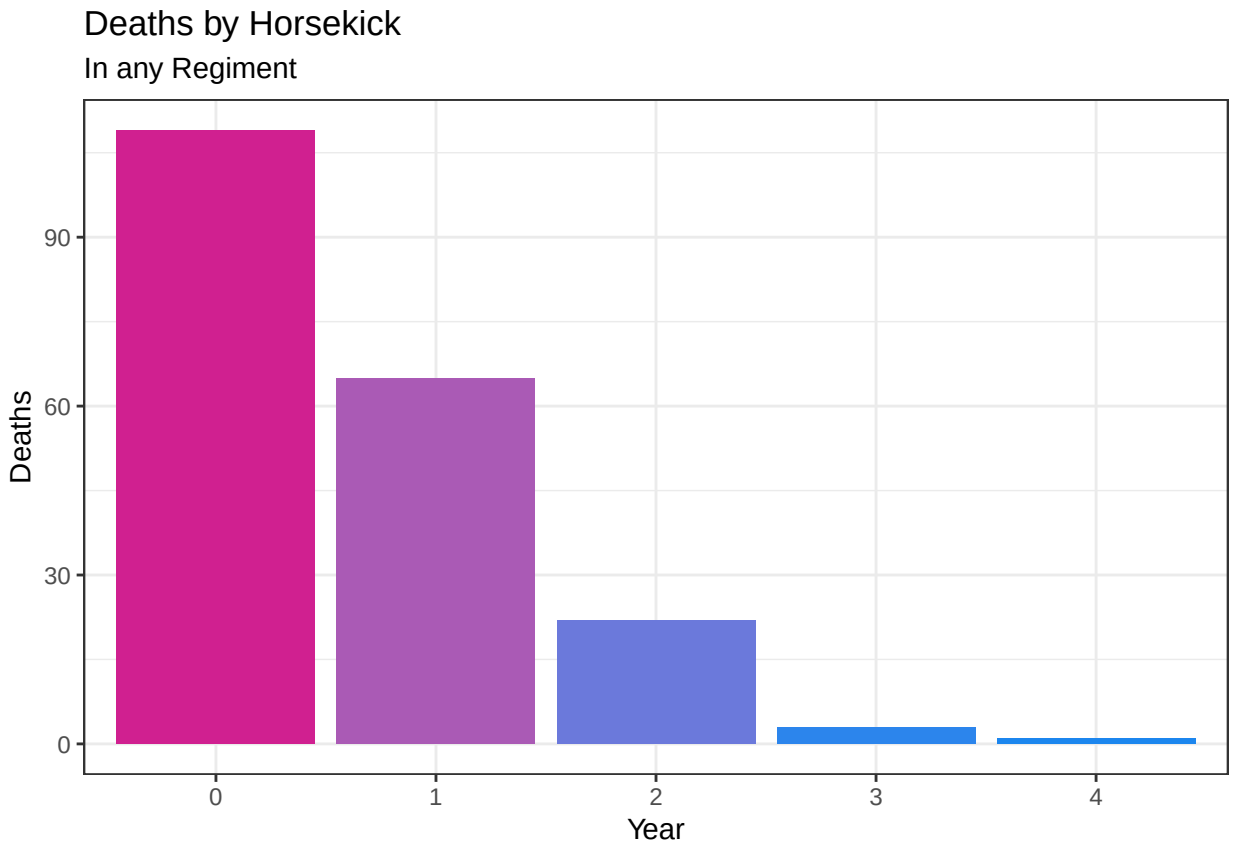
```r
horsekick        <- c(109, 65, 22, 3, 1)
names(horsekick) <- 0:4
print(horsekick)
```

```
##   0   1   2   3   4
## 109  65  22   3   1
```

```r
barplot(horsekick, col = "grey")
```




```r
hk_tb <- tibble::enframe(horsekick, name = "Year", value = "Deaths")

bp <- ggplot(hk_tb, aes(x = Year, y = Deaths, fill = Deaths)) +
    geom_col() +
    scale_fill_gradient(high = "#D02090", low = "#1c86ee") +
    theme_bw() +
    labs(title = "Deaths by Horsekick", subtitle = "In any Regiment") +
    guides(fill = FALSE)

bp
```



Mean Value

In order to determine the average number of deaths over the period of years (this is weird because the data set is weird, don't pay mind to it).

```
sum((0:4)*horsekick)/sum(horsekick)
```

```
## [1] 0.61
```

Simulation and Poisson Probabilities

Poisson Values can be simulated, If a delivery driver has 3 jobs every day, a month, in no particular order might look like this:

```
rpois(30, lambda = 3)
```

```
##  [1]  3  3  5  1  4  4  3  3  2  3  3  1  3  3  2  2  7  2  4  2  1  2  4  4  4
## [26]  4  2 11  3  6
```

but maybe we would have to consider different days as different poisson distributions?

Return Probability density

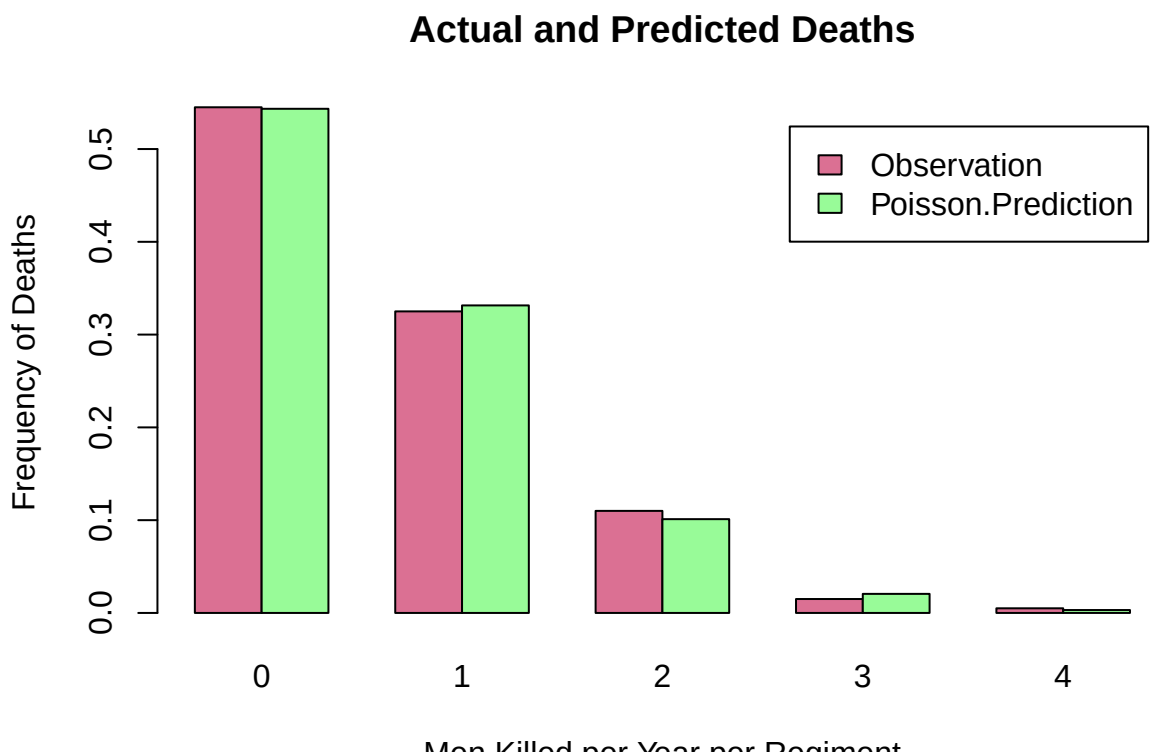
If we wanted to know the probability of a delivery driver getting a various number of jobs:

```
dpois(0:6, lambda = 3) %>% round(1)
```

```
## [1] 0.0 0.1 0.2 0.2 0.2 0.1 0.1
```

This means that the Horsekick data can be predicted and visualised:

```
lambda = sum(0:4*horsekick)/sum(horsekick)
obs <- horsekick/200
pred <- dpois(0:4, lambda)

horse_df  <- data.frame("Year" = 0:4, "Observation" = obs, "Poisson Prediction" = pred)
horse_mat <- as.matrix(horse_df)[,-1]
horse_mat <- t(horse_mat)

barplot(horse_mat, beside = TRUE,
        col = c("PaleVioletRed", "PaleGreen"),
        legend = TRUE,
        main = "Actual and Predicted Deaths",
        xlab = "Men Killed per Year per Regiment",
        ylab = "Frequency of Deaths")
```

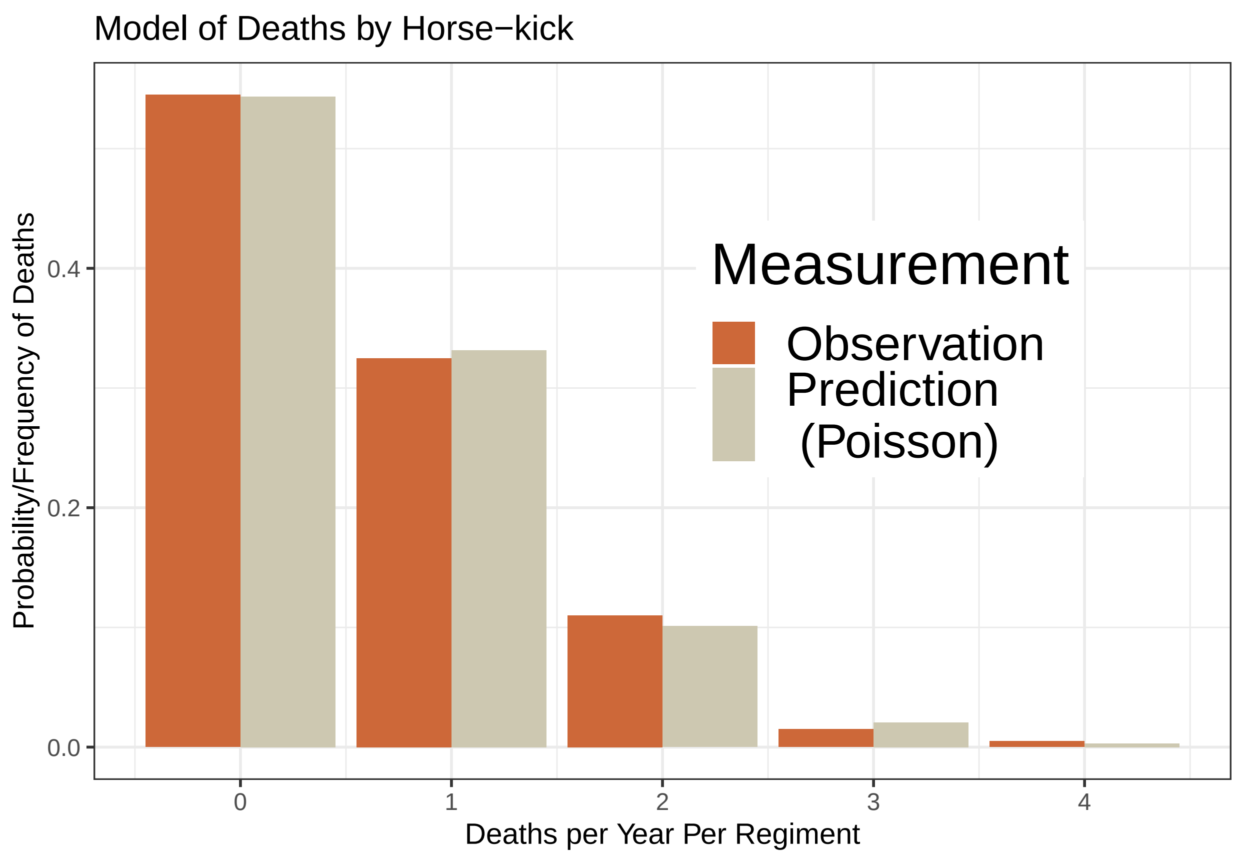



```
horse_tib <- pivot_longer(horse_df, cols = c(Observation, Poisson.Prediction), names_to = "Source")
## Using GGplot2
ggplot(horse_tib, aes(x = Year, y = value, fill = Source)) +
    geom_col(position = 'dodge') +
    labs(x = "Deaths per Year Per Regiment",
         y = "Probability/Frequency of Deaths",
         title = "Model of Deaths by Horse-kick") +
    guides(fill = guide_legend("Measurement")) +
    scale_fill_manual(values = c("Sienna3", "Cornsilk3"),
                      labels = c("Observation", "Prediction\n\t(Poisson)")) +
    theme_bw() +
    theme(legend.position = c(0.7, 0.6),
          legend.text = element_text(size = 18),
```

```
        legend.title = element_text(size = 22))
```

```
## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
```

```
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call(C_textBounds, as.graphicsAnnot(x$label), x$x, x$y, : font
## width unknown for character 0x9

## Warning in grid.Call.graphics(C_text, as.graphicsAnnot(x$label), x$x, x$y, :
## font width unknown for character 0x9
```



Confidence Intervals

Bootsrap Confidence Intervals

Binomial

Poisson

Approximate Confidence Intervals

Choropleth maps

Using ggmap