MVA Case study in R: An outbreak of gastroenteritis in Stegen, Germany

Niklas Willrich (RKI), Patrick Keating (AGES), and Alexander Spina (AGES)

27 February 2017

Contributors to *R* code:
Daniel Gardiner (PHE) and Lukas Richter (AGES)

Authors:
Alain Moren and Gilles Desve

Reviewers:
Marta Valenciano, Alain Moren.

Adapted for the EPIET MVA module December 2015: Alicia Barrasa (EPIET), Ioannis Karagiannis (UK-FETP)

Prerequisites

Participants are expected to be familiar with data management and basic analysis in R

Introduction

On 26 June 1998, the St Sebastian High School in Stegen (school A), Germany, celebrated a graduation party, where 250 to 350 participants were expected. Attendants included graduates from that school, their families and friends, teachers, 12th grade students and some graduates from a nearby school (school B).

A self-service party buffet was supplied by a commercial caterer in Freiburg. Food was prepared the day of the party and transported in a refrigerated van to the school.

Festivities started with a dinner buffet open from 8.30 pm onwards and were followed by a dessert buffet offered from 10 pm. The party and the buffet extended late during the night and alcoholic beverages were quite popular. All agreed it was a party to be remembered.

The alert

On 2nd July 1998, the Freiburg local health office reported to the Robert Koch Institute (RKI) in Berlin the occurrence of many cases of gastroenteritis following the graduation party described above. More than 100 cases were suspected among participants and some of them were admitted to nearby hospitals. Sick people suffered from fever, nausea, diarrhoea and vomiting lasting for several days. Most believed that the tiramisu consumed at dinner was responsible for their illness. *Salmonella enteritidis* was isolated from 19 stool samples.

The Freiburg health office sent a team to investigate the kitchen of the caterer. Food preparation procedures were reviewed. Food samples, except tiramisu (none was left over), were sent to the laboratory of Freiburg University. Microbiological analyses were performed on samples of the following: brown chocolate mousse, caramel cream, remoulade sauce, yoghurt dill sauce, and 10 raw eggs.

The Freiburg health office requested help from the RKI in the investigation to assess the magnitude of the outbreak and identify potential vehicle(s) and risk factors for transmission in order to better control the outbreak

The study

Cases were defined as any person who had attended the party at St Sebastian High School who suffered from diarrhoea (min. 3 loose stool for 24 hours) between 27 June and 29 June 1998; or who suffered from at least three of the following symptoms: vomiting, fever over 38.5° C, nausea, abdominal pain, headache.

Students from both schools attending the party were asked through phone interviews to provide names of persons who attended the party.

Overall, 291 responded to enquiries and 103 cases were identified.

An introduction to the R companion

This text was adapted from the introduction used at the 2016 TSA module.

R packages are bundles of functions which extend the capability of R. Thousands of add-on packages are available in the main online repository (known as CRAN) and many more packages in development can be found on GitHub. They may be installed and updated over the Internet.

We will mainly use packages which come ready installed with R (base code), but where it makes things easier we will use add-on packages. In addition, we have included a few extra functions to simplify the code required. All the R packages you need for the exercises can be installed over the Internet.

Setting up

Required R packages and functions

Install and load the required R packages for this practical.

> **n.b.** you should only need to do this once.

Run the following code to make sure that you have all the functions that you need for this practical.

```
# Function to make tables with counts, proportions and cumulative sum
big_table <- function(data, useNA = "no") {
  count <- table(data, useNA = useNA)
  prop <- round(prop.table(count)*100, digits = 2)
  cumulative <- cumsum(prop)
  rbind(count,
        prop,
        cumulative)
}

attack_rate <- function(table) {
  prop <- round(prop.table(table,1),digits = 2)
  denominator <- rowSums(table)
  output <- cbind(Ill = table[,2], N = denominator, Proportions = prop[,2])
  return(output)
}
```

The big_table function uses data directly and allows combining of counts, proportions and cumulative sums, thus reducing the number of lines of code required for descriptive analyses. The attack_rate function makes tables that combine counts, proportions and row sums.

The dataset

In this practical, we will be using the tirav12.csv file located in the data folder.

Read in the dataset for this practical.

```
tira_data <- read.csv(here::here("data", "tirav12.csv"), stringsAsFactors = FALSE)
```

Question 1. What are the main characteristics of the study population?

Describe your dataset:

- frequency distributions, means, medians, modes, quartiles, SD, quartiles, outliers
- make appropriate histograms and box plots
- make sure that your missing values are properly coded as missing (i.e. as opposed to “9”)

a) Browse your dataset.

What variables does your dataset contain?

b) Describe your dataset.

Look at:

- the number of observations and variable types
- mean, median, and maximum values for each variable

c) Recode data.

Identify variables with missing values (variables that have a records with a value of 9). Recode these to NA.

d) Create summary tables with counts and proportions.

For each variable in the dataset.

e) Make a box plot and histogram of age.

f) Number of cases by date of onset.

Use the incidence package to create an epicurve for this outbreak, providing information on daily incidence. Also create an epicurve stratified by sex.

Help Q1

Browsing your dataset

RStudio has the nice feature that everything is in one browser window, so you can browse your dataset and your code without having to switch between browser windows.

```
# to browse your data, use the View command
View(tira_data)
```

Alternatively, you can also view your dataset by clicking on **tira_data** in the top right "global environment" panel of your *RStudio* browser. Your global environment is where you can see all the datasets, functions and other things you have loaded in the current session.

Describing your dataset

You can view the structure of your data set using the following commands:

```
# str provides an overview of the number of observations and variable types
str(tira_data)
```

```
## 'data.frame':    291 obs. of  21 variables:
##  $ uniquekey  : int  210 12 288 186 20 148 201 106 272 50 ...
##  $ ill        : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ dateonset  : chr  "1998-06-27" "1998-06-27" "1998-06-27" "1998-06-27" ...
##  $ sex        : int  1 0 1 0 1 0 0 0 1 0 ...
##  $ age        : int  18 57 56 17 19 16 19 19 40 53 ...
```

```
##  $ tira       : int  1 1 0 1 1 1 1 1 1 1 ...
##  $ tportion   : int  3 1 0 1 2 2 3 2 2 1 ...
##  $ wmousse    : int  0 0 0 1 0 1 0 1 1 1 ...
##  $ dmousse    : int  1 1 0 0 0 1 1 1 1 1 ...
##  $ mousse     : int  1 1 0 1 0 1 1 1 1 1 ...
##  $ mportion   : int  1 1 0 NA 0 1 1 1 2 1 ...
##  $ beer       : int  0 0 0 0 1 0 0 0 1 0 ...
##  $ redjelly   : int  0 0 0 1 0 0 0 1 0 1 ...
##  $ fruitsalad : int  0 1 0 0 0 1 1 1 0 0 ...
##  $ tomato     : int  0 0 1 0 0 0 0 0 1 0 ...
##  $ mince      : int  0 1 1 0 0 1 0 0 0 0 ...
##  $ salmon     : int  0 1 1 9 0 1 0 0 1 1 ...
##  $ horseradish: int  0 1 0 0 0 0 0 1 0 1 ...
##  $ chickenwin : int  0 0 0 0 0 1 0 1 0 1 ...
##  $ roastbeef  : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ pork       : int  1 0 0 9 0 0 0 0 0 0 ...
```

```r
# summary provides mean, median and max values of your variables
summary(tira_data)
```

```
##    uniquekey          ill          dateonset              sex        
##  Min.   :  1.0   Min.   :0.000   Length:291         Min.   :0.0000  
##  1st Qu.: 73.5   1st Qu.:0.000   Class :character   1st Qu.:0.0000  
##  Median :146.0   Median :0.000   Mode  :character   Median :1.0000  
##  Mean   :146.0   Mean   :0.354                      Mean   :0.5223  
##  3rd Qu.:218.5   3rd Qu.:1.000                      3rd Qu.:1.0000  
##  Max.   :291.0   Max.   :1.000                      Max.   :1.0000  
##                                                                     
##       age             tira           tportion         wmousse      
##  Min.   :12.00   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000  
##  1st Qu.:18.00   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000  
##  Median :20.00   Median :0.0000   Median :0.0000   Median :0.0000  
##  Mean   :26.66   Mean   :0.4231   Mean   :0.6678   Mean   :0.2599  
##  3rd Qu.:27.00   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000  
##  Max.   :80.00   Max.   :1.0000   Max.   :3.0000   Max.   :1.0000  
##  NA's   :8       NA's   :5        NA's   :5        NA's   :14      
##     dmousse           mousse          mportion           beer       
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000  
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000  
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000  
##  Mean   :0.3937   Mean   :0.4256   Mean   :0.6523   Mean   :0.3911  
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000  
##  Max.   :1.0000   Max.   :1.0000   Max.   :3.0000   Max.   :1.0000  
##  NA's   :4        NA's   :2        NA's   :12       NA's   :20      
##     redjelly        fruitsalad        tomato           mince      
##  Min.   :0.0000   Min.   :0.000   Min.   :0.0000   Min.   :0.000  
##  1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:0.0000   1st Qu.:0.000  
##  Median :0.0000   Median :0.000   Median :0.0000   Median :0.000  
##  Mean   :0.2715   Mean   :0.244   Mean   :0.2852   Mean   :0.299  
##  3rd Qu.:1.0000   3rd Qu.:0.000   3rd Qu.:1.0000   3rd Qu.:1.000  
##  Max.   :1.0000   Max.   :1.000   Max.   :1.0000   Max.   :1.000  
##                                                                   
##      salmon        horseradish       chickenwin       roastbeef      
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000  
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000  
##  Median :0.0000   Median :0.0000   Median :0.0000   Median :0.00000  
##  Mean   :0.4811   Mean   :0.3093   Mean   :0.2887   Mean   :0.09966  
##  3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:1.0000   3rd Qu.:0.00000  
##  Max.   :9.0000   Max.   :9.0000   Max.   :1.0000   Max.   :1.00000  
```

```
##
##       pork
##  Min.   :0.0000
##  1st Qu.:0.0000
##  Median :0.0000
##  Mean   :0.4742
##  3rd Qu.:1.0000
##  Max.   :9.0000
##
```

```
# describe (from Hmisc package) provides no. of observations, missing values, unique levels of each variable
Hmisc::describe(tira_data)
```

```
## tira_data
##
##  21  Variables      291  Observations
## --------------------------------------------------------------------------------
## uniquekey
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      291        0      291        1      146    97.33     15.5     30.0
##      .25      .50      .75      .90      .95
##     73.5    146.0    218.5    262.0    276.5
##
## lowest :   1   2   3   4   5, highest: 287 288 289 290 291
## --------------------------------------------------------------------------------
## ill
##        n  missing distinct     Info      Sum     Mean      Gmd
##      291        0        2    0.686      103    0.354   0.4589
##
## --------------------------------------------------------------------------------
## dateonset
##        n  missing distinct
##      131      160       11
##
## 1998-06-26 (1, 0.008), 1998-06-27 (56, 0.427), 1998-06-28 (51, 0.389),
## 1998-06-29 (10, 0.076), 1998-06-30 (3, 0.023), 1998-07-01 (3, 0.023),
## 1998-07-02 (3, 0.023), 1998-07-04 (1, 0.008), 1998-07-05 (1, 0.008),
## 1998-07-06 (1, 0.008), 1998-07-09 (1, 0.008)
## --------------------------------------------------------------------------------
## sex
##        n  missing distinct     Info      Sum     Mean      Gmd
##      291        0        2    0.749      152   0.5223   0.5007
##
## --------------------------------------------------------------------------------
## age
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##      283        8       46     0.99    26.66    13.75       16       17
##      .25      .50      .75      .90      .95
##       18       20       27       52       57
##
## lowest : 12 13 14 15 16, highest: 60 62 64 65 80
## --------------------------------------------------------------------------------
## tira
##        n  missing distinct     Info      Sum     Mean      Gmd
##      286        5        2    0.732      121   0.4231   0.4899
##
## --------------------------------------------------------------------------------
## tportion
```

```
##        n  missing distinct     Info     Mean      Gmd
##      286        5        4    0.793   0.6678   0.8993
##
## Value          0     1     2     3
## Frequency    165    65    42    14
## Proportion 0.577 0.227 0.147 0.049
## ---------------------------------------------------------------------------
## wmousse
##        n  missing distinct     Info      Sum     Mean      Gmd
##      277       14        2    0.577       72   0.2599   0.3861
##
## ---------------------------------------------------------------------------
## dmousse
##        n  missing distinct     Info      Sum     Mean      Gmd
##      287        4        2    0.716      113   0.3937   0.4791
##
## ---------------------------------------------------------------------------
## mousse
##        n  missing distinct     Info      Sum     Mean      Gmd
##      289        2        2    0.733      123   0.4256   0.4906
##
## ---------------------------------------------------------------------------
## mportion
##        n  missing distinct     Info     Mean      Gmd
##      279       12        4    0.777   0.6523   0.8902
##
## Value          0     1     2     3
## Frequency    166    55    47    11
## Proportion 0.595 0.197 0.168 0.039
## ---------------------------------------------------------------------------
## beer
##        n  missing distinct     Info      Sum     Mean      Gmd
##      271       20        2    0.714      106   0.3911   0.4781
##
## ---------------------------------------------------------------------------
## redjelly
##        n  missing distinct     Info      Sum     Mean      Gmd
##      291        0        2    0.593       79   0.2715   0.3969
##
## ---------------------------------------------------------------------------
## fruitsalad
##        n  missing distinct     Info      Sum     Mean      Gmd
##      291        0        2    0.553       71    0.244   0.3702
##
## ---------------------------------------------------------------------------
## tomato
##        n  missing distinct     Info      Sum     Mean      Gmd
##      291        0        2    0.612       83   0.2852   0.4091
##
## ---------------------------------------------------------------------------
## mince
##        n  missing distinct     Info      Sum     Mean      Gmd
##      291        0        2    0.629       87    0.299   0.4206
##
## ---------------------------------------------------------------------------
## salmon
##        n  missing distinct     Info     Mean      Gmd
##      291        0        3    0.706   0.4811   0.6861
##
```

```
## Value          0     1     9
## Frequency    183   104     4
## Proportion 0.629 0.357 0.014
## ---------------------------------------------------------------------------
## horseradish
##        n  missing distinct     Info     Mean      Gmd
##      291        0        3     0.57   0.3093   0.4902
##
## Value          0     1     9
## Frequency    217    72     2
## Proportion 0.746 0.247 0.007
## ---------------------------------------------------------------------------
## chickenwin
##        n  missing distinct     Info      Sum     Mean      Gmd
##      291        0        2    0.616       84   0.2887   0.4121
##
## ---------------------------------------------------------------------------
## roastbeef
##        n  missing distinct     Info      Sum     Mean      Gmd
##      291        0        2    0.269       29  0.09966   0.1801
##
## ---------------------------------------------------------------------------
## pork
##        n  missing distinct     Info     Mean      Gmd
##      291        0        3    0.734   0.4742   0.5982
##
## Value          0     1     9
## Frequency    169   120     2
## Proportion 0.581 0.412 0.007
## ---------------------------------------------------------------------------
```

"table", "summary", and "describe" functions provide similar output to the "tabulate", "summarize", and "codebook" commands in Stata.

"Summary" and "describe" can be applied to:

- the whole dataset
- specific variables of interest

In the example below we look at sex, age and pork in the **tira_data** dataset. You can examine a variable within a dataset using the '$' sign followed by the variable name.

```r
# table will give a very basic frequency table (counts),
table(tira_data$sex)
```

```
##
##   0   1
## 139 152
```

```r
# summary gives the mean, median and max values of the specified variable
summary(tira_data$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   12.00   18.00   20.00   26.66   27.00   80.00       8
```

```
# describe gives the number of data points, missing values and number of categories
describe(tira_data$pork)
```

```
## tira_data$pork
##        n  missing distinct     Info     Mean      Gmd
##      291        0        3    0.734   0.4742   0.5982
##
## Value          0     1     9
## Frequency    169   120     2
## Proportion 0.581 0.412 0.007
```

Recode the data

Use the "describe" command to assess your data and identify variables with missing values. The describe command showed that the variables salmon, pork and horseradish have a few records with a value of 9. These need to be recoded to NA

- Using the square brackets "[. . .]" after a variable allows you to subset for certain observations. To recode values of 9 to NA for the pork variable, select observations where pork **(tira_data$pork)** is equal to 9 [**tira_data$pork == 9**] and set these observations equal to NA
- Always use the double equals "==" within square brackets; this a logical (Boolean) operator
- Use "! =" when you want to write "not equal to"

```
# The first line below is read as follows:  assign a value of NA to tira_data$pork WHERE tira_data$pork is eq
tira_data$pork[tira_data$pork == 9] <- NA

tira_data$salmon[tira_data$salmon == 9] <- NA

tira_data$horseradish[tira_data$horseradish == 9] <- NA
```

Create summary tables with counts and proportions

We can create individual tables for each variable with the following steps:

```
# Assign the counts of tira_data$sex to the object "sex"
sex <- table(tira_data$sex)

# Assign the proportion of tira_data$sex to the object "prop" and round the values to 2 decimal places
prop <- round(prop.table(sex)*100, digits = 2)

# Assign the cumulative sum of tira_data$sex to the object "cumul"
cumul <- cumsum(prop)

# Append/row bind the results of the three objects together and assign to the object table1
table1 <- rbind(sex,prop,cumul)
```

```
table1
```

```
##            0      1
## sex   139.00 152.00
## prop   47.77  52.23
## cumul  47.77 100.00
```

We could also use the big_table function (on page 2), which does all of the above steps in one line.

```r
big_table(tira_data$sex)
```

```
##                 0      1
## count      139.00 152.00
## prop        47.77  52.23
## cumulative  47.77 100.00
```

```r
big_table(tira_data$beer)
```

```
##                 0      1
## count      165.00 106.00
## prop        60.89  39.11
## cumulative  60.89 100.00
```

We could use the big_table function on each of our variables, or we could use a **for loop** to loop through our variables (similar to Stata) with the big_table function.

```r
# List the variables of interest and use c() to combine the elements into a vector
vars <- c("ill", "tira", "beer", "pork", "salmon")

# Create an empty list to hold the output of your loop
output <- list()

# Apply big_table to each element of the object in vars. In this loop, "var" is the indexing variable; any ch
for (var in vars) {
  # Within the [], the item before the comma refers to rows and the item after the comma refers to columns
  total <- big_table(tira_data[,var])
  # assign the value of your tables (total) to the output list (note: double square brackets "[[]]" are used
  output[[var]] <- total
}

output
```

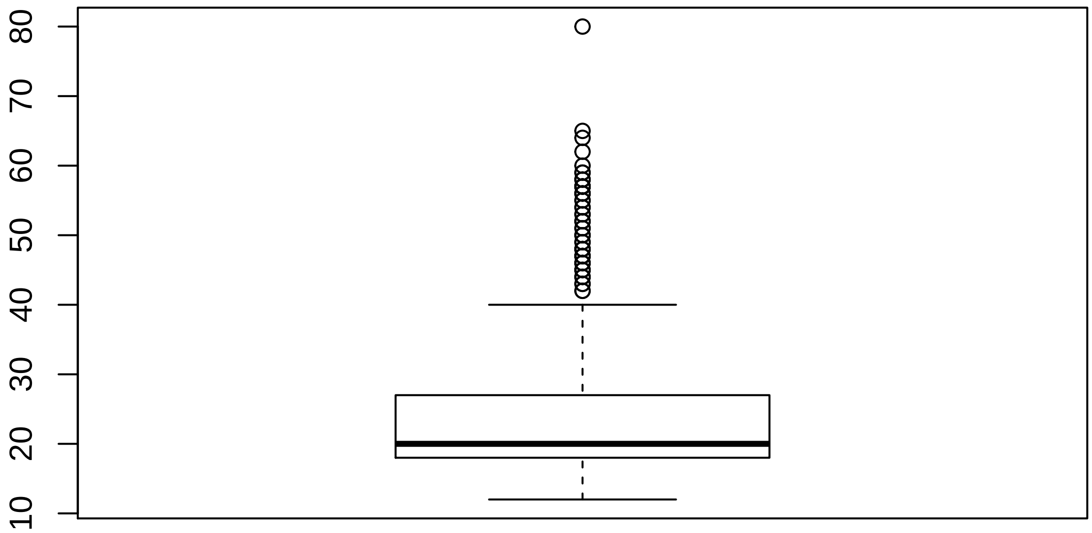
```
## $ill
##                0     1
## count      188.0 103.0
## prop        64.6  35.4
## cumulative  64.6 100.0
##
## $tira
##                 0      1
## count      165.00 121.00
## prop        57.69  42.31
## cumulative  57.69 100.00
##
## $beer
##                 0      1
## count      165.00 106.00
## prop        60.89  39.11
## cumulative  60.89 100.00
##
## $pork
##                 0      1
## count      169.00 120.00
## prop        58.48  41.52
## cumulative  58.48 100.00
```

```
##
## $salmon
##                 0      1
## count      183.00 104.00
## prop        63.76  36.24
## cumulative  63.76 100.00
```

Make a box plot and histogram of age

You can use the following to examine the age distribution among people who attended the party, as well as only those and who fell ill and additionally to save the chart.

```
# Boxplot of the age of all who attended the party
boxplot(tira_data$age)
```




```
# Histogram of the ages of those who attended the party and who fell ill

# To save the histogram, the file path and filename must be specified prior to running the histogram code

# This function changes the "graphics device" to jpeg. You can view the current graphics device using dev.cur

jpeg(filename = "N:/MED/IMED-VIE/INFE/Public/CC-INFE-Schmid/EPIET/Learning R/R Case studies/MVA module 2016/H

# Here we use the hist function to plot the age of cases only (ill == 1)
# You will see that RStudio creates a jpeg file in your working directory with the above path and filename.
age_hist_all <- hist(tira_data$age[tira_data$ill == 1],
                     xlab = "Age",
                     ylab = "No. of cases",
                   main = "Histogram of the ages of cases")

# This function closes the graphics device and returns to the default
dev.off()
```

If we believe that there are two identifiable age groups, then we can create a new age group variable using **one** of the following approaches:

```
# by using ifelse (similar to Excel if statements)
tira_data$agegroup <- ifelse(tira_data$age >= 30, 1, 0)
```

```
# Two alternative approaches
# The below are particularly useful when you want to create more than 2 categories
# by using cut
tira_data$agegroup <- cut(tira_data$age, c(0,30,150), labels = FALSE) - 1
# by using findInterval
tira_data$agegroup <- findInterval(tira_data$age, c(30,150))
```

Describe the outbreak in terms of person and time

You can produce summary tables by person and time (no place variable provided) using the big_table function.

```
# Table 1: Descriptive epidemiology: Study population by sex
big_table(tira_data$sex)
```

```
##                 0      1
## count      139.00 152.00
## prop        47.77  52.23
## cumulative  47.77 100.00
```

```
# Table 2: Descriptive epidemiology: Study population by age group
# useNA ="always" here allows you to see the proportion of NAs for this variable
big_table(tira_data$agegroup, useNA = "always")
```

```
##                 0     1   <NA>
## count      215.00 68.00   8.00
## prop        73.88 23.37   2.75
## cumulative  73.88 97.25 100.00
```

```r
summary(tira_data$age)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   12.00   18.00   20.00   26.66   27.00   80.00       8
```

```r
# Table 3: Descriptive epidemiology: Attack rate
big_table(tira_data$ill)
```

```
##                0     1
## count      188.0 103.0
## prop        64.6  35.4
## cumulative  64.6 100.0
```

```r
# Table 4: Descriptive epidemiology: Cases by date of onset of illness
big_table(tira_data$dateonset)
```

```
##            1998-06-26 1998-06-27 1998-06-28 1998-06-29 1998-06-30
## count            1.00      56.00      51.00      10.00       3.00
## prop             0.76      42.75      38.93       7.63       2.29
## cumulative       0.76      43.51      82.44      90.07      92.36
##            1998-07-01 1998-07-02 1998-07-04 1998-07-05 1998-07-06
## count            3.00       3.00       1.00       1.00       1.00
## prop             2.29       2.29       0.76       0.76       0.76
## cumulative      94.65      96.94      97.70      98.46      99.22
##            1998-07-09
## count            1.00
## prop             0.76
## cumulative      99.98
```

Question 2: What is/are the vehicle/s for this outbreak?

a) Compute food-specific attack rates and % of cases exposed

b) Choose the appropriate measure of association and the appropriate statistical tests and appropriate level of confidence:

c) Look at the proportion of cases exposed. What would be your suspected food item at this point?

d) Compute the proportion of cases exposed for each exposure

Help questions 2a to d

As we are carrying out a cohort study, the appropriate measure of association is relative risk. The appropriate statistical test for determining a p-value is a Chi2 test of comparison of proportions. For our analyses we will use a 95% confidence level, as this is the standard used in public health.

The outputs required for a, c and d are provided by the same function as described below. In Stata, we would normally use the **cstable** and **csinter** commands to calculate food-specific attack rates and the proportion of cases exposed to specific exposures. There are a number of ways of doing this in R. Below you will see two approaches. The first approach gives us the % of cases exposed to tiramisu.

```
# The first element will be rows and the 2nd will be columns
count <- table(tira_data$tira,tira_data$ill, deparse.level = 2)

# Here we select row % of count by including ,1 in the prop.table section
prop <- round(prop.table(count,1),digits = 2)

# We obtain the denominator using the rowSums function
denominator <- rowSums(count)

# We combine all the elements together using cbind (binding by columns)
tira <- cbind(Ill = count[,2], N = denominator, Proportions = prop[,2])
tira
```

```
##   Ill   N Proportions
## 0   7 165        0.04
## 1  94 121        0.78
```

Alternatively, we can use a user-written command called single variable analysis.v.02 (developed by Daniel Gardiner Cohort 2015). This gives similar output to the cstable command in Stata.

```
# This function needs to be saved in the same folder as the working directory
source(here::here("scripts/single.variable.analysis.v0.2.R"))
```

```
# specify your exposures of interest i.e. tira-pork
vars <- c("tira", "wmousse", "dmousse", "mousse", "beer", "redjelly", "fruitsalad", "tomato", "mince", "salmo
```

```
#NB. click on "sva" in your global environment to view Daniel's source code and read his explanations
a <- sva(tira_data, outcome = "ill", exposures = c(vars), measure = "rr", verbose = TRUE)
a
```

```
##       exposure exp exp.cases exp.AR unexp unexp.cases unexp.AR     rr
## 1         tira 121        94   77.7   165           7      4.2 18.312
## 2      wmousse  72        49   68.1   205          49     23.9  2.847
## 3      dmousse 113        76   67.3   174          26     14.9  4.501
## 4       mousse 123        81   65.9   166          22     13.3  4.969
```

```
## 5         beer 106        30   28.3   165           69     41.8  0.677
## 6     redjelly  79        45   57.0   212           58     27.4  2.082
## 7   fruitsalad  71        46   64.8   220           57     25.9  2.501
## 8       tomato  83        35   42.2   208           68     32.7  1.290
## 9        mince  87        32   36.8   204           71     34.8  1.057
## 10      salmon 104        37   35.6   183           63     34.4  1.033
## 11 horseradish  72        30   41.7   217           72     33.2  1.256
## 12  chickenwin  84        33   39.3   207           70     33.8  1.162
## 13   roastbeef  29         8   27.6   262           95     36.3  0.761
## 14        pork 120        48   40.0   169           54     32.0  1.252
##    lower  upper  p.value
## 1  8.814 38.043 0.000000
## 2  2.128  3.809 0.000000
## 3  3.087  6.563 0.000000
## 4  3.299  7.483 0.000000
## 5  0.476  0.963 0.028064
## 6  1.556  2.786 0.000004
## 7  1.887  3.314 0.000000
## 8  0.938  1.774 0.136893
## 9  0.757  1.475 0.789388
## 10 0.745  1.433 0.897642
## 11 0.901  1.751 0.202601
## 12 0.838  1.611 0.417660
## 13 0.413  1.402 0.417293
## 14 0.918  1.708 0.170878
```

To calculate attack rates for age and sex, you can use the attack_rate function.

```
# the attack_rate function acts on tables and not data (as in the big_table function)
counts_sex <- table(tira_data$sex, tira_data$ill)
attack_rate(counts_sex)
```

```
##   Ill   N Proportions
## 0  53 139        0.38
## 1  50 152        0.33
```

```
counts_age <- table(tira_data$agegroup, tira_data$ill)
attack_rate(counts_age)
```

```
##   Ill   N Proportions
## 0  75 215        0.35
## 1  25  68        0.37
```

e) Search for any dose response if appropriate

Use the variable tportion and tabulate it. Consider whether you would recode this variable so it has fewer categories, and actually do it.

```
# Tabulate tportion variable against illness using attack_rate function
counts_tportion <- table(tira_data$tportion, tira_data$ill)
attack_rate(counts_tportion)
```

```
##   Ill   N Proportions
## 0   7 165        0.04
## 1  44  65        0.68
## 2  38  42        0.90
## 3  12  14        0.86
```

```r
# Recode 3 portions of tportion as 2 portions
# Make a new variable called tportion2 that has the same values as tportion
tira_data$tportion2 <- tira_data$tportion
tira_data$tportion2[tira_data$tportion2 == 3] <- 2
```

```r
# Calculate counts, proportions and sum of recoded tportion2
counts_tportion2 <- table(tira_data$tportion2,tira_data$ill)
attack_rate(counts_tportion2)
```

```
##   Ill   N Proportions
## 0   7 165        0.04
## 1  44  65        0.68
## 2  50  56        0.89
```

Here you should be able to see that those who ate 2 or more portions of tiramisu have a higher attack rate than those that ate only 1 portion of tiramisu. Those who ate 1 portion of tiramisu have a higher attack rate than those who ate no tiramisu.

f) Interpret the results and identify the outbreak vehicle if any.

Refer to the results of the **sva** output and identify likely vehicles.

Several food items seemed to be associated with the occurrence of illness; tiramisu, dark and white chocolate mousse, fruit salad, and red jelly. They can potentially explain up to 94, 76, 49, 46, and 45 of the 103 cases respectively. Investigators decided to identify their respective role in the occurrence of illness.

From the crude analysis, epidemiologists noticed that the occurrence of gastroenteritis was lower among those attendants who had drunk beer. They also decided to assess if beer had a protective effect on the occurrence of gastroenteritis.

Question 3: How would you assess if the chocolate mousses were the vehicles of the illness?

Question 4. How would you assess if beer had a protective effect on the occurrence of illness?

Help questions 3 and 4

Identify the variables which are potential effect modifiers and confounders.

Stata users could use the **csinter** function to identify effect modifiers/confounders. The **epi.2by2** function in the epiR package provides similar functionality. Outcome and exposure variables of interest need to be **factor/categorical variables** prior to performing stratified analysis with this function and also need to be **relevelled from (0,1) to (1,0)** so that they can be correctly organised in a 2 by 2 table.

```
# Convert outcome/exposure variables to factor variables and reorder them
# The variables of interest are identified by their column number but variable names could equally be used
vars <- colnames(tira_data[,c(2,6,8:10,12:21)])

for (var in vars) {
  tira_data[,var] <- factor(tira_data[,var],levels = c(1,0)) # levels of the variable are now (1,0) instead o
}
```

Stratify key exposure variables by exposure to tiramisu. We will use exposure to **wmousse** stratified by tiramisu as an example of the steps required and then run a loop over all variables of interest.

```
# Make a 3-way table with exposure of interest, the outcome and the stratifying variable in that order
a <- table(tira_data$wmousse, tira_data$ill, tira_data$tira)

# Use the epi.2by2 function to calculate RRs (by stating method = "cohort.count")
mh1 <- epi.2by2(a, method = "cohort.count")

# View the output of mh1
mh1
```

```
##              Outcome +    Outcome -      Total        Inc risk *
## Exposed +           47           22         69              68.1
## Exposed -           49          155        204              24.0
## Total               96          177        273              35.2
##                   Odds
## Exposed +        2.136
## Exposed -        0.316
## Total            0.542
##
##
## Point estimates and 95 % CIs:
## -------------------------------------------------------------------
## Inc risk ratio (crude)                       2.84 (2.12, 3.80)
## Inc risk ratio (M-H)                         1.23 (1.02, 1.48)
## Inc risk ratio (crude:M-H)                   2.31
## Odds ratio (crude)                           6.76 (3.71, 12.31)
## Odds ratio (M-H)                             2.25 (1.01, 5.05)
## Odds ratio (crude:M-H)                       3.00
## Attrib risk (crude) *                        44.10 (31.64, 56.56)
## Attrib risk (M-H) *                          11.47 (-14.72, 37.66)
## Attrib risk (crude:M-H)                      3.84
## -------------------------------------------------------------------
##  Test of homogeneity of IRR: X2 test statistic: 13.477 p-value: < 0.001
##  Test of homogeneity of  OR: X2 test statistic: 7.233 p-value: 0.007
##  Wald confidence limits
##  M-H: Mantel-Haenszel
##  * Outcomes per 100 population units
```

```r
# We can select specific elements of mh1 using the $ twice as below
# Crude RR
mh1$massoc$RR.crude.wald
```

```
##        est   lower    upper
## 1 2.835847 2.11641 3.799846
```

```r
# Stratum-specific RR
mh1$massoc$RR.strata.wald
```

```
##         est     lower    upper
## 1  1.078595 0.8946851  1.30031
## 2 11.294118 2.7572793 46.26194
```

```r
# Adjusted RR
mh1$massoc$RR.mh.wald
```

```
##        est    lower    upper
## 1 1.227898 1.021927 1.475382
```

```r
# We can combine all of those elements in to a single table using rbind
results <- rbind(mh1$massoc$RR.crude.wald,
                          mh1$massoc$RR.strata.wald,
                          mh1$massoc$RR.mh.wald)


# We can label the rows of this table as below
rownames(results) <- c("Crude", "Strata 1", "Strata 0", "Adjusted")

results
```

```
##                est     lower     upper
## Crude     2.835847 2.1164097  3.799846
## Strata 1  1.078595 0.8946851  1.300310
## Strata 0 11.294118 2.7572793 46.261941
## Adjusted  1.227898 1.0219273  1.475382
```

We can now put all of the above steps in a for loop and apply it to all of the variables of interest.

```r
# Select wmousse, dmousse, mousse and beer to pork as variables of interest
vars <- c("wmousse", "dmousse", "mousse", "beer", "redjelly", "fruitsalad", "tomato", "mince", "salmon", "hor

# Create an empty list to save the output of the loop
output3 <- list()


for (var in vars) {
  b <- table(tira_data[,var], tira_data$ill, tira_data$tira)
  mh <- epiR::epi.2by2(b, method = "cohort.count")
  resultstable <- rbind(mh$massoc$RR.crude.wald,
                        mh$massoc$RR.strata.wald,
                        mh$massoc$RR.mh.wald)
  rownames(resultstable) <- c("Crude", "Strata 1", "Strata 0", "Adjusted")
  output3[[var]] <- resultstable
}
```

```
## Warning in qf(1 - N., 2 * a, 2 * c + 2): NaNs produced
```

```
output3 # Gives crude, stratum-specific and adjusted RRs
```

```
## $wmousse
##                est     lower     upper
## Crude     2.835847 2.1164097  3.799846
## Strata 1  1.078595 0.8946851  1.300310
## Strata 0 11.294118 2.7572793 46.261941
## Adjusted  1.227898 1.0219273  1.475382
##
## $dmousse
##                est     lower     upper
## Crude     4.476224 3.0686715  6.529398
## Strata 1  1.045455 0.8322176  1.313329
## Strata 0 16.022727 3.3099508 77.562418
## Adjusted  1.271697 1.0199911  1.585518
##
## $mousse
##                est     lower     upper
## Crude     4.937500 3.2773963  7.438498
## Strata 1  1.062766 0.8304511  1.360070
## Strata 0 13.269231 2.7184609 64.769181
## Adjusted  1.306886 1.0277660  1.661809
##
## $beer
##                est     lower     upper
## Crude    0.6974742 0.4899427 0.9929125
## Strata 1 0.7839721 0.6157496 0.9981530
## Strata 0 1.0357143 0.2401914 4.4660388
## Adjusted 0.8016329 0.6238023 1.0301585
##
## $redjelly
##                est     lower    upper
## Crude    2.1329640 1.5912038 2.859178
## Strata 1 0.9786415 0.8074733 1.186094
## Strata 0 1.2083333 0.1532792 9.525557
## Adjusted 0.9855072 0.8084612 1.201325
##
## $fruitsalad
##                est     lower     upper
## Crude     2.576155 1.9420052  3.417384
## Strata 1  1.026488 0.8476913  1.242996
## Strata 0 12.416667 3.0456078 50.621624
## Adjusted  1.170130 0.9713650  1.409567
##
## $tomato
##                est     lower     upper
## Crude    1.3192905 0.9581780  1.816497
## Strata 1 0.9708141 0.7910830  1.191379
## Strata 0 2.3437500 0.5475857 10.031607
## Adjusted 1.0300098 0.8329043  1.273760
##
## $mince
##                est     lower    upper
## Crude    1.0785305 0.7719538 1.506862
## Strata 1 0.9034615 0.7255375 1.125018
## Strata 0 1.9402174 0.4516000 8.335792
## Adjusted 0.9546119 0.7605370 1.198211
```

```
##
## $salmon
##                est     lower    upper
## Crude    1.0111111 0.7243853 1.411329
## Strata 1 0.9439655 0.7677181 1.160675
## Strata 0 0.7785714 0.1559618 3.886680
## Adjusted 0.9320267 0.7478143 1.161617
##
## $horseradish
##               est     lower    upper
## Crude    1.272709 0.9093277 1.781302
## Strata 1 1.132812 0.9412713 1.363331
## Strata 0 0.000000 0.0000000      NaN
## Adjusted 1.047734 0.8521226 1.288249
##
## $chickenwin
##                est     lower    upper
## Crude    1.1670168 0.8399503 1.621439
## Strata 1 0.9492188 0.7684916 1.172448
## Strata 0 2.0625000 0.4805998 8.851243
## Adjusted 1.0026527 0.8047753 1.249184
##
## $roastbeef
##                est     lower    upper
## Crude    0.7623285 0.4135421 1.405286
## Strata 1 1.1364943 0.8583971 1.504687
## Strata 0 1.1428571 0.1446593 9.028958
## Adjusted 1.1372400 0.8039242 1.608752
##
## $pork
##               est     lower    upper
## Crude    1.298566 0.9492131 1.776496
## Strata 1 1.010204 0.8353316 1.221685
## Strata 0 2.215054 0.5126599 9.570601
## Adjusted 1.066360 0.8713803 1.304968
```

Have a look at the association between beer and the illness. By stratifying the analysis on tiramisu consumption we can measure the potential protective effect of beer among those who ate tiramisu. It seems that consumption of beer may reduce the effect of tiramisu consumption on the occurrence of gastroenteritis. The RR does not significantly differ between the two strata (0.8 vs. 1.0 and confidence intervals overlap). But, effect modification may be present. A similar stratification was conducted assessing dose response for tiramisu consumption among beer drinkers and no-beer drinkers.

After stratifying beer consumption by the amount of tiramisu consumed, it appeared that beer consumption reduced the effect of tiramisu on the occurrence of gastroenteritis only among those who had eaten an average amount of tiramisu. This is suggesting that, if the amount of tiramisu was large, consumption of beer no longer reduced the risk of illness when eating tiramisu.

How would you proceed with your analysis?