An outbreak of gastroenteritis in Stegen, Germany

Patrick Keating and Alexander Spina

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

The following code has been adapted to *R* for learning purposes. The initial contributors are listed below. All copyrights and licenses of the original document apply here as well.

Authors: :Alain Moren and Gilles Desve

Reviewers: :Marta Valenciano, Alain Moren.

Adapted for previous modules: Alicia Barrasa and Ioannis Karagiannis

Prerequisites

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

Copyright and license

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 (??? 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 ???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

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.

```r
# Installing required packages for the week
required_packages <- c("foreign", "Hmisc", "epiDisplay", "epiR", "xlsx", "epitools", "ISOweek")
install.packages(required_packages)
```

Run the following code at the beginning of each of the training days to make sure that you have made available all the packages and functions that you need. Be sure to include it in any scripts too.

```r
# Loading required packages for the week
required_packages <- c("foreign", "Hmisc", "epiDisplay", "epiR", "xlsx", "epitools", "ISOweek")

for (i in seq(along = required_packages)) {
  library(required_packages[i], character.only = TRUE)
}
```

```r
# 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)
}

 # Function to provide counts, denominator and proportions (equivalent of attack rate)
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)
}

# Adds a function to create epicurves
source("C:/Users/Spina/Desktop/MSF training/Stegen/Session2/epicurve.v.1.8.R")

# Adds a function to create output similar to cctable or cstable in Stata
source("C:/Users/Spina/Desktop/MSF training/Stegen/Session2/single.variable.analysis.v0.2.R")
```

R and Stata have minor differences in default settings and methods. In this document we will follow the Stata analysis as closely as possible, but small and usually unimportant differences may be noted between the statistical findings in R and those in Stata. At some points additional steps (which would usually be optional in R) will be taken to produce output which is comparable to that of Stata.

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 **epicurve** function allows creation of easily formatted epicurves. To find out more about the function, first load it as above and then click on function in the **Global Environment** tab on the right of the R Studio window.

The **single variable analysis** function allows calculation of attack rates of multiple variables at one time and provides similar output to the cctable and cstable commands in Stata.

You will work with Stata.dta data sets which can be loaded into R with the “foreign” or “readstata13” packages. The appropriate functions to use will be indicated.

R can hold one or many data sets in memory simultaneously, so there is usually no need to save intermediate files or close and re-open datasets.

Dataset is now available for analysis

Setting your working directory

```
setwd("C:/Users/Spina/Desktop/MSF Training/Stegen/Session2/")
```

Reading in your dataset

```
tira.data <- read.dta("tirav12.dta", convert.factors = FALSE)
```

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

1a: Describe the dataset

- frequency distributions, means, medians, modes, quartiles, SD, quartiles, outliers.

Hint: Use the **str**, **summary*, and **describe** functions

You can find out more about those functions by typing ? followed by the name of the function in the console e.g. ?str

1b: Recode any missing values that are improperly coded as missing (i.e. NA as opposed to "9").

Hint: use the [] to subset your dataset when making changes

1c: Create summary tables with counts and proportions for the following variables: ill, sex, tira, beer, pork and salmon

Hint: Use the function **big.table** on page X e.g. big.table(data$sex). Combine with a for loop to save time

1d: Make appropriate histograms and box plots.

Hint: use the **hist** and **boxplot** functions

1e: Create an age group variable with 2 groups (<30 and >=30).

Hint: use one of theh following **cut**, **ifelse** or **findInterval** functions

Help Task 1

1a: Describe the 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)

# summary provides mean, median and max values of your variables
summary(tira.data)

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

1b: 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”

Below is code to correct the errors in the pork variable. You will need to do the same for salmon and horseradish.

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

1c: Create summary tables with counts and proportions

You 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
```

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

```
big.table(tira.data$sex)
```

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

You could use the big.table function on each of the variables, or you could use a **for loop** to loop through the variables (similar to Stata) with the big.table function.

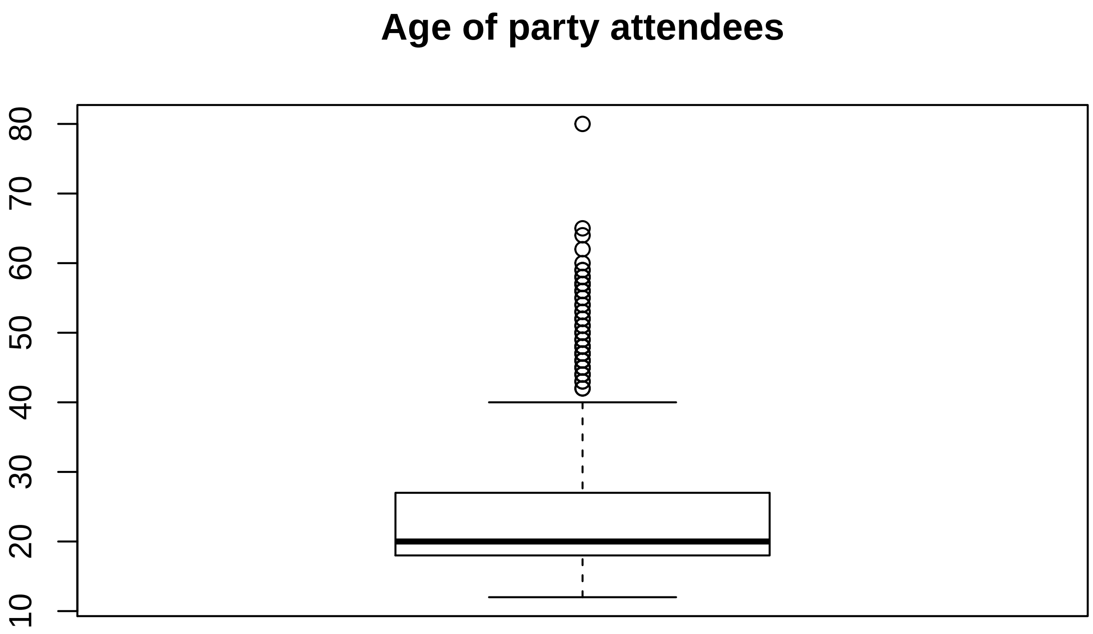
```
# List the variables of interest and use c() to combine the elements into a
# vector
vars <- c("ill", "tira", "beer", "pork", "salmon")

# 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 character can be used e.g. 'i'
for (var in vars) {

    # within the [], the item before the comma refers to rows 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 to subset elements of a list)
    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      9
## count      183.00 104.00   4.00
## prop        62.89  35.74   1.37
## cumulative  62.89  98.63 100.00
```

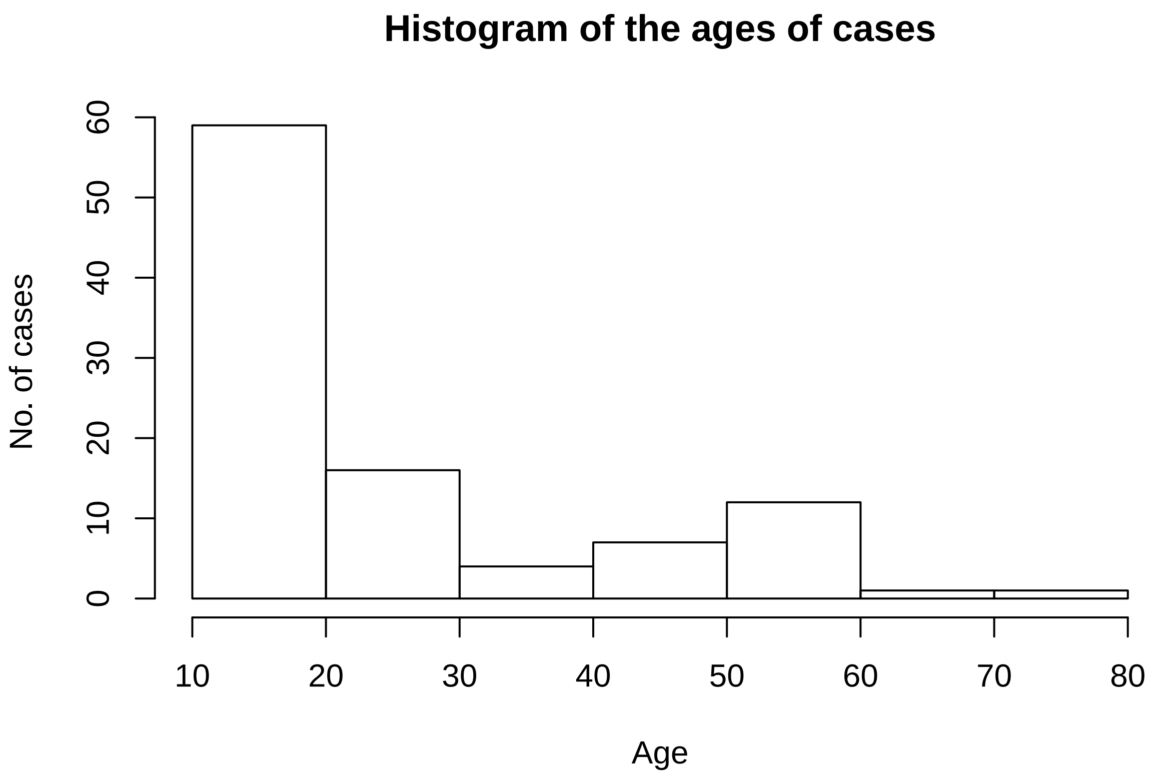
1d: Make a box plot and histogram of age

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.

```
# Boxplot of the age of all who attended the party
boxplot(tira.data$age, main = "Age of party attendees")
```



```
# Histogram of the ages of those who attended the party and who fell ill
age_hist_all <- hist(tira.data$age[tira.data$ill == 1],
                     xlab = "Age",
                     ylab = "No. of cases",
                   main = "Histogram of the ages of cases")
```



1e: Create a new age group variable with 2 groups (<30 and >=30)

You 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 ( levels start at 1, so we have to subtract 1)
tira.data$agegroup <- findInterval(tira.data$age, c(0,30,150)) - 1
```

Task 2: Describe the outbreak in terms of person and time

2a: Describe the outbreak by person (age and sex)

Hint: use the **big.table** function

2b: Describe the outbreak by time

Hint: use the **epicurve** function. You can find out more about this function by clicking on it in the global environment.

The epicurve function requires specifying a number of elements including:

- dataset
- date.col =date variable
- time period = day/week/month
- start.date = first date to show on epicurve
- end.date = final date to show on epicurve
- x and y labels
- epi.squares = TRUE, allows you to see individual boxes for each case
- na.rm = TRUE, means any missing values are excluded

e.g. epicurve(data, date.col= "dateofsymptoms", time.period ="day", start.at = "2003-08-09", stop.at = "2003-08-14", xlab = "Date of symptom onset", ylab = "No. of cases", epi.squares = TRUE, na.rm = TRUE)

Help Task 2

2a: Describe the outbreak by sex and age

You can produce summary tables by person (no place variable provided) using the big.table function.

```
# 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
```

```
# 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
```

```
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
```

```
# Attack rate
big.table(tira.data$ill)
```

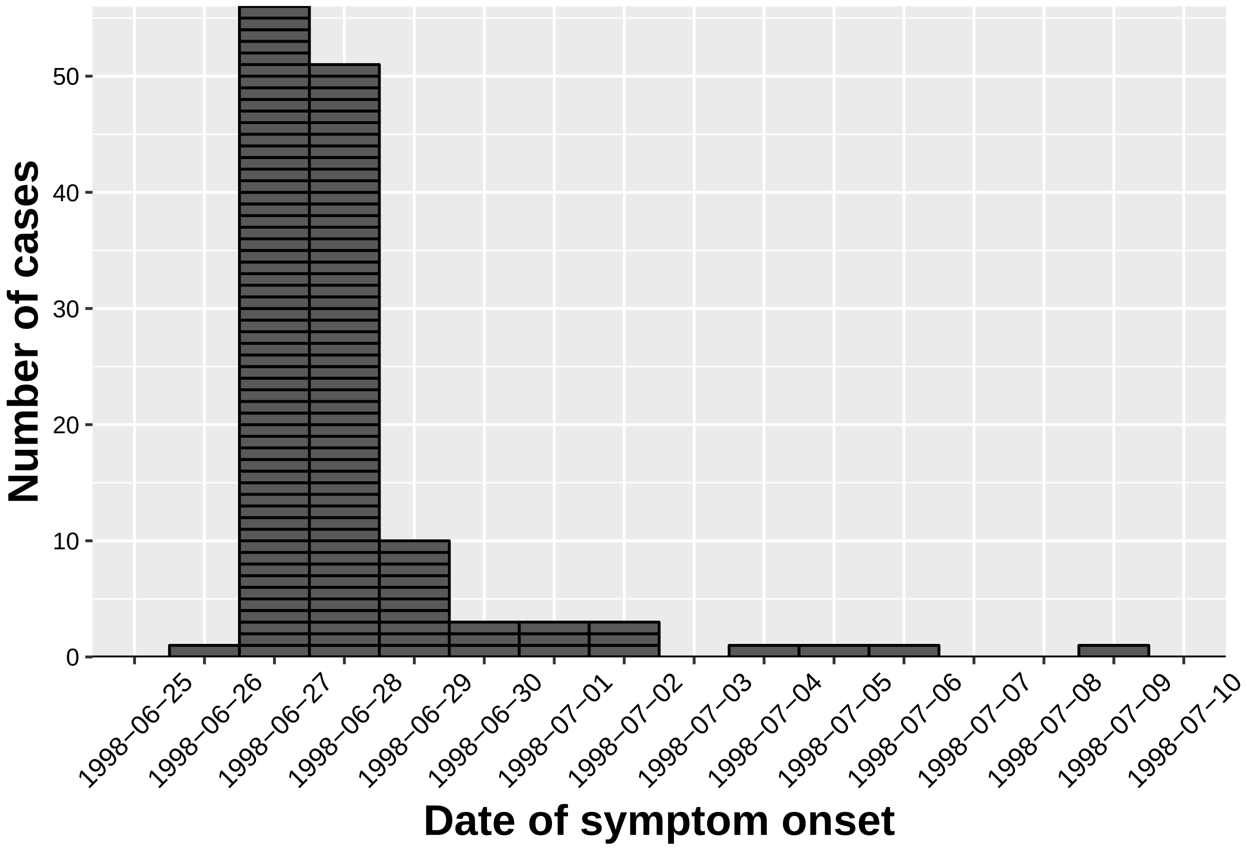
```
##                0     1
## count      188.0 103.0
## prop        64.6  35.4
## cumulative  64.6 100.0
```

2b: Describe the outbreak by time using the epicurve function

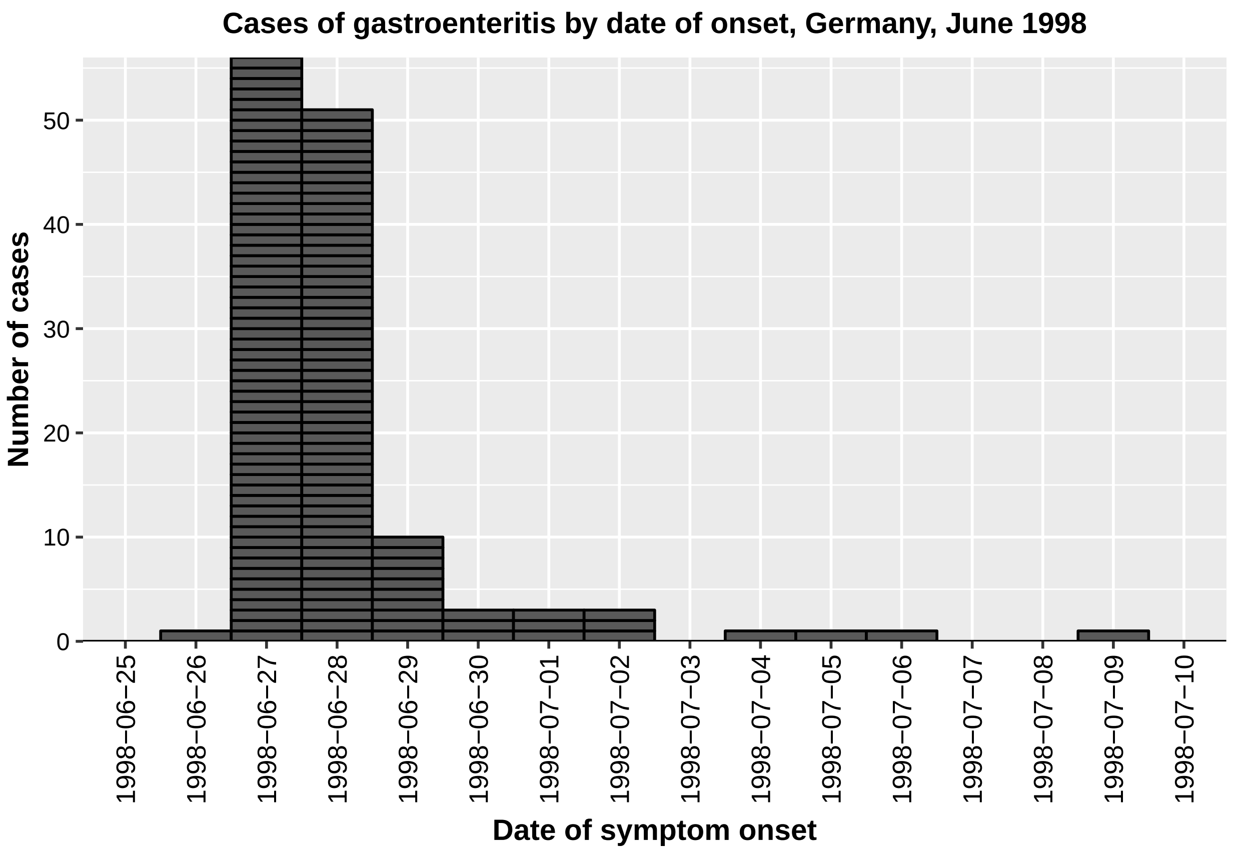
```
epicurve.tira <- epicurve(tira.data, date.col = "dateonset", time.period = "day",
                start.at = "1998-06-25", stop.at = "1998-07-10",
                xlab = "Date of symptom onset", ylab = "Number of cases",
                col.pal = 4, label.breaks = 0, epi.squares = TRUE, na.rm = TRUE)
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:epiDisplay':
##
##     alpha

## 160 rows have missing dates OR dates outside of the start/stop period
```

```
epicurve.tira
```



```
#This can be further improved by some of the following:

epicurve.tira <- epicurve.tira +

              # rotating the x axis label by 90
                  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
              # resizing the x-axis label
                  theme(axis.title.x = element_text(size = 11)) +
              # resizing the y-axis label
                  theme(axis.title.y = element_text(size = 11)) +
              # adding a title
                  ggtitle("Cases of gastroenteritis by date of onset, Germany, June 1998") +
              # centring the title and reducing its size
                  theme(plot.title = element_text(hjust = 0.5, size = 11))


epicurve.tira
```



```
# You could save the epicurve as follows
##ggsave(filename = "C:/Users/Patrick/Documents/MSF R training/output/epicurve.png")
```

Task 3: Identify the outbreak vehicle if any

3a Compute food-specific attack rates

You will need to crosstabulate your variable of interest with the outcome variable. Please do this for **at least one exposure variable** e.g. tira

Hint: Use the **table** function for counts and **prop.table** of your table for proportions. The **attack.rate** function from page 2 could be used for attack rates by age and sex. You will need to crosstabulate the exposure and outcome variables and then use this table as the object of the attack.rate function

3b Compute the proportion of cases exposed for each exposure

Hint: Use the **sva** function on page 2 to obtain results for all variables of interest at one time (click on the **sva** function in the global environment to find out more about how to use it).

3c: Search for any dose response if appropriate

Hint: Use the variable **tportion** and tabulate it. Consider whether you would recode this variable so it has fewer categories, and then actually do it. Recoding tportion can be done in a similar way as for salmon/horseradish in the previous task.

3d Interpret the results and identify the outbreak vehicle if any.

Help 3a/b

The outputs required for a and b 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
# You can label the table with variable names using deparse.level
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 FETP cohort 2015). This function was included at the start of the script. It gives similar output to the cstable command in Stata.

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

```
#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
```

Help 3c 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
```

```
# 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
```

```
# 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.

Help 3d 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.

Task 4: Identify the variables which are potential effect modifiers and confounders.

4a Stratify an exposure variable by tiramisu

Hint: use the **epi.2by2** function. Don't forget that outcome and exposure variables need to be factor variables and relevelled (from 0,1 to 1,0)

4b Stratify key exposure variables of interest by tira using a for loop

Help 4a

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 variable
# names could equally be used
vars <- colnames(tira.data[, c(2, 6, 8:10, 12:21)])




# convert variables to factors levels of the variable are now (1,0) instead
# of (0,1)
for (var in vars) {
    tira.data[, var] <- factor(tira.data[, var], levels = c(1, 0))
}
```

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
```

```
# 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
```

```
# 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
```

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

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

```
# You 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
```

Help 4b

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

```
# Select wmousse, dmousse, mousse and beer to pork as variables of interest
vars <- colnames(tira.data[,c(8:10,12:21)])

# 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 <- 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
}

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.