Stegen case study: Are you confused or ready to interact when eating Tiramisu and drinking beer?

# Copyright and License

**Source:**

This case study was first designed by Alain Moren and Gilles Desve, EPIET. It is based on an investigation conducted by Anja Hauri, RKI, Berlin, 1998

Minor revisions were brought to this case study by IntoEpi 2009,2010, Hong Kong 2016 It was then translated in to *R* by Alexander Spina in 2018.

**Revisions:**

*If you modify this case study, please indicate below your name and changes you made*

**You are free:**

* **to Share** — to copy, distribute and transmit the work
* **to Remix** — to adapt the work

**Under the following conditions:**

* **Attribution** — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work). The best way to do this is to keep as it is the list of contributors: sources, authors and reviewers.
* **Share Alike** — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one. Your changes must be documented. Under that condition, you are allowed to add your name to the list of contributors.
* You cannot sell this work alone but you can use it as part of a teaching.

**With the understanding that:**

* **Waiver** — Any of the above conditions can be waived if you get permission from the copyright holder.
* **Public Domain** — Where the work or any of its elements is in the public domain under applicable law, that status is in no way affected by the license.
* **Other Rights** — In no way are any of the following rights affected by the license:
  + Your fair dealing or fair use rights, or other applicable copyright exceptions and limitations;
  + The author’s moral rights;
  + Rights other persons may have either in the work itself or in how the work is used, such as publicity or privacy rights.
* **Notice** — For any reuse or distribution, you must make clear to others the license terms of this work by keeping together this work and the current license.

This licence is based on <http://creativecommons.org/licenses/by-sa/3.0/>

# Objectives

At the end of the case study, using *R* for stratified analysis and logistic regression, participants should be able to analyse data from a food borne outbreak investigation and to sort out the respective roles played by several food vehicles.

# Introduction

# Session 1

On 26 June 1998 the St Sebastian High School in Stegen, Germany, celebrated the graduation from school by organising a party to which 250 to 350 participants were expected. Attendants included graduates from St Sebastian High School, their families and friends, teachers, 12th grade students and some graduates from the nearby Marie-Curie school of Kirchzarten.

A self service party buffet was supplied by a commercial caterer from 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 and 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

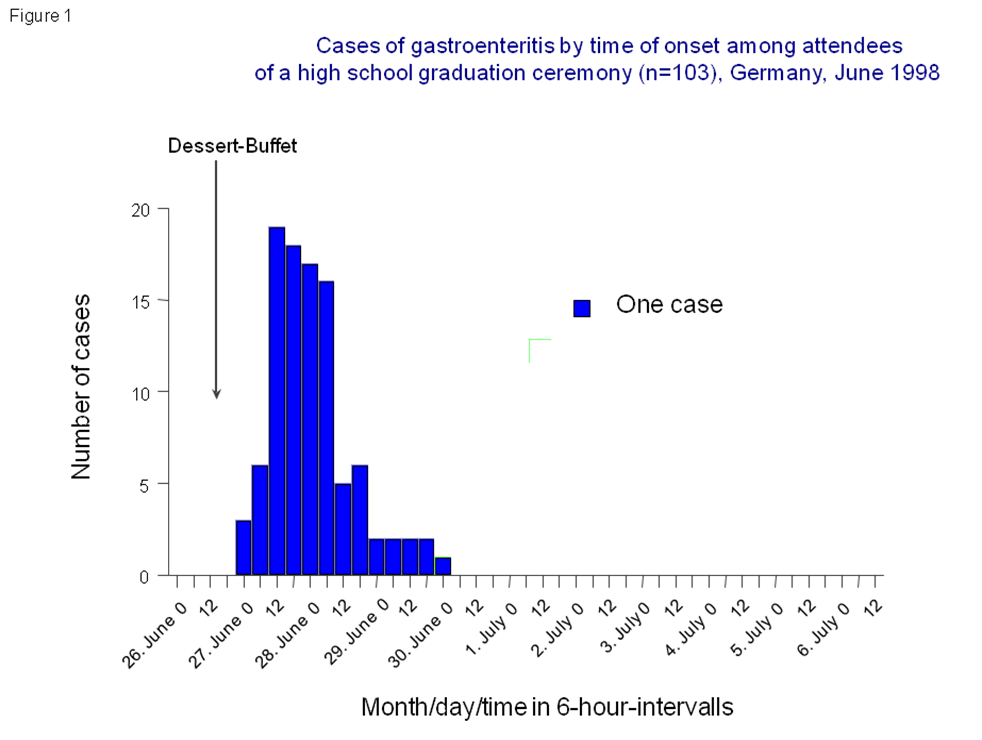
On 2nd July 1998, the Freiburg Health office of the Federal Council Office of Breisgau-Hochschwarzwald reported to the Robert Koch Institute (RKI) in Berlin the occurrence of many cases of gastroenteritis following the graduation party described above. More than 100 cases were suspected among participants 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.

Cases were defined as any person attending 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 (Attack rate = 35%). Among these cases, 84 received medical treatments and four were admitted to hospitals. Attack rates by age group were 36.6% for persons < 20 years, 32.1% for persons 20 to 29 years, and 36.8% for persons older than 29 years.



## Q1. In order to prepare your analysis, summarise the above information and formulate hypotheses to be tested. (5 min discussion)

The shape of the epidemic curve, the attendance to a single event (a buffet) pointed towards a food borne outbreak related to a point common source of infection.

Using the updated list of attendants, a retrospective cohort study including all attendants to the party (that could be reached) was conducted. All had received a standard questionnaire asking for demographic information, signs, symptoms and duration, admission to hospital, and food and beverages consumption at the party including amount consumed. Food specific attack rates were computed for more than 50 food items and beverages.

## Q2. Establish a plan of analysis. (10 min discussion)

## Help task 2

**Perform data cleaning**

* For each variable, look at range, unexpected values, missing values.
* Correct data using original forms if needed

**Describe each variable**

* For each variable, describe frequency distributions including missing values and if needed means, median, modes, quartiles, SD, outliers
* Make appropriate histograms and box plots
* Choose relevant characteristics to describe the population

**Identify the outbreak vehicle if any**

* Chose the appropriate measure of association (RR, RD or OR)
* Chose the appropriate statistical tests and appropriate level of confidence
* Compute food specific attack rates
* Look at proportion of cases exposed
* Compute attributable risk % among exposed
* Search for any dose response if appropriate
* Interpret the results

**Do a stratified analysis**

* Identify the variables that are potential effect modifiers (EM) and confounders
* Design appropriate stratification tables
* Stratify on each level taken by the EM and confounders
* Compute appropriate measurements to identify confounding and effect modification
* Apply appropriate statistical tests
* Interpret the results

**Do a multivariable analysis**

* Prepare the data set for the multivariable analysis
* Identify numerical, nominal, discrete, continuous variables and decide how to analyse them
* Create additional variables as needed (age groups, dummy variables, etc.)
* Recode as necessary
* Add variables one by one
* Check for confounding and interaction
* Select a final model

## Q3. Check and clean the dataset “stegen.dta”. Create labels where appropriate.

Verify your working directory and open stegen.dta. The data set “stegen.dta” includes the following variables:

|  |  |  |  |
| --- | --- | --- | --- |
| Variable.name | Type | Stata.code | Definition |
| ill | Dichotomous | 1=Yes, 0=No | Case Y/N |
| dateonset | Date |  | Date of onset of illness |
| sex | Dichotomous | 1=Man, 0=Woman | Gender |
| age (years) | Continued (can be transformed into categories) |  | Age in years |
| tira | Dichotomous | 1=Yes, 0=No | Consumption of tiramisu Y/N |
| tportion | Categorical (Four categories) | 4 categories 0: none ; 1=one portion, 2=two portions,3=three portions | Amount of portions of tiramisu consumed |
| wmousse | Dichotomous | 1=Yes, 0=No | Consumption of white chocolate mousse Y/N |
| dmousse | Dichotomous | 1=Yes, 0=No | Consumption of dark chocolate mousse |
| mousse | Dichotomous | 1=Yes, 0=No | Consumption of chocolate mousse |
| mportion | Categorical | 4 categories 0=none, 1=one 2=two portions, 3=three portions…. | Amount of portions of mousse consumed |
| beer | Dichotomous | 1=Yes, 0=No | Consumption of beer |
| redjelly | Dichotomous | 1=Yes, 0=No | Consumption of red jelly |
| fruitsalad | Dichotomous | 1=Yes, 0=No | Consumption of fruit salad |
| tomato | Dichotomous | 1=Yes, 0=No | Consumption of tomato |
| mince | Dichotomous | 1=Yes, 0=No | Consumption of minced meat |
| salmon | Dichotomous | 1=Yes, 0=No | Consumption of salmon |
| horseradish | Dichotomous | 1=Yes, 0=No | Consumption of horseradish |
| chickenwin | Dichotomous | 1=Yes, 0=No | Consumption of chicken wings |
| roastbeef | Dichotomous | 1=Yes, 0=No | Consumption of roastbeef |
| pork | Dichotomous | 1=Yes, 0=No | Consumption of pork |

## Help task 3

Describe your dataset: frequency distributions, means, median, modes, quartiles, SD, outliers. Make appropriate histograms and box plots.

### Setting your working directory

You can check the path for your current working directory using the *getwd* function.

#Check your current working directory  
getwd()

To set your working directory you can use the *setwd* function.

setwd("C:/Users/Username/Desktop/EpiconceptStegen")

### Reading in files

Import the dataset from a comma seperated value (.csv) file using the *read.csv* function, storing it as a dataframe within *R* called *tira.data*.For a CSV file the separator is normally a comma (“,”), however depending on the language of your operating system this can also be other values, for example a semi-colon (“;”). Here we also specify that we do not want to read in string (character or grouped variables as factors).

tira.data <- read.csv("stegen.csv", sep = ";", stringsAsFactors = FALSE )

### Describe your dataset

For example:

summary(tira.data)   
table(tira.data$sex)  
table(tira.data$beer, useNA = "always")   
summary(tira.data$age)   
aggregate(tira.data$age, by = list(tira.data$sex), FUN = summary)  
prop.table( table( tira.data$ill) )

*Example: ill*

#get counts of ill   
 #save table as "counts"  
counts <- table(tira.data$ill)   
  
#get proportions for counts table  
prop.table(counts)  
  
#you could also multiple by 100 and round to 2 digits  
round(prop.table(counts)\*100, digits = 2)

*Example: sex*

#get counts   
 #save table as "counts"  
counts <- table(tira.data$sex)   
  
#get proportions for counts table  
prop.table(counts)  
  
#you could also multiple by 100 and round to 2 digits  
round(prop.table(counts)\*100, digits = 2)

It is also possible to use a custom function to pull these various lines of code together in a custom function. You do not need to understand this code at current. You can run this code below which saves the *big.table* function in your environment; then you can use it the same way any other function works.

# Function to make tables with counts, proportions and cumulative sum  
big.table <- function(vars, data, useNA = "always") {  
 # 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  
 count <- table(data[ , var], useNA = useNA)  
 prop <- round(prop.table(count)\*100, digits = 2)  
 cumulative <- cumsum(prop)  
 total <- t(rbind(count,  
 prop,  
 cumulative))  
 # 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  
   
}

You can now use this function as any other function.

# specify the variable in quotations and the dataset to use  
big.table(var = "sex", data = tira.data)

## $sex  
## count prop cumulative  
## 0 139 47.77 47.77  
## 1 152 52.23 100.00  
## <NA> 0 0.00 100.00

*To show more than one table at a time:*

We could use the big.table function to show more than one table at a time.

# specify multiple vars using c()  
big.table(var = c("tira", "pork", "salmon"), data = tira.data)

*Describing continuous variable, example: age*

# use the aggregate function to group by sex  
 # sex must be as a list  
 # specify the function you would like to use (summary)  
aggregate(tira.data$age, by = list(tira.data$sex), FUN = summary)

## Group.1 x.Min. x.1st Qu. x.Median x.Mean x.3rd Qu. x.Max. x.NA's  
## 1 0 12.00000 17.50000 19.00000 26.37778 28.00000 80.00000 4.00000  
## 2 1 13.00000 18.00000 20.00000 26.92568 24.50000 65.00000 4.00000

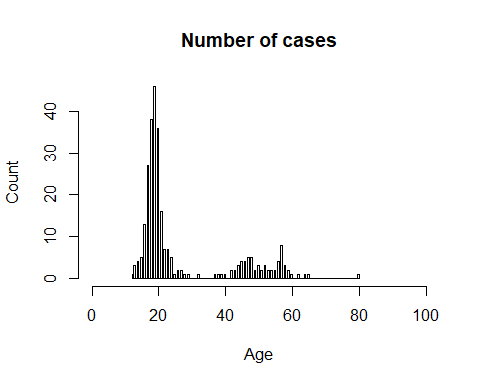
*Histograms*:

A frequency plot is the default for *hist*. In order to plot each unique value of age on the x-axis, specify number of breaks (in this case 100 years); set the x-axis to match this.

# Plot a histogram of age  
 # you can specify a bar for each age with "breaks"  
 # you can set your x axis from 0-100 using "xlim"  
hist(tira.data$age,   
 xlab = "Age",  
 ylab = "Count",   
 breaks = 100,  
 xlim = c(0, 100)  
)

To have a nicer title:

# Plot a histogram of age  
 # you can specify a bar for each age with "breaks"  
 # you can set your x axis from 0-100 using "xlim"  
 # main is where you set your title  
hist(tira.data$age,   
 xlab = "Age",  
 ylab = "Count",   
 breaks = 100,  
 xlim = c(0, 100),   
 main = "Number of cases"  
)



To save you can plot, then use *dev.copy* to choose a file type and name; *dev.off* closes the connection.

# save histogram of age as a png file  
dev.copy(png,'age.png')  
dev.off()

If you believe that two age groups are identifiable you may want to create a new variable with two age classes (< 30 years and above). The following shows one way to do it:

# create a binary variable for older than 30 years of age  
tira.data$agegroup <- ifelse(tira.data$age >= 30, 1, 0 )

# check the age grouping   
table(tira.data$agegroup)

The *str* function will provide an overview of which variable types are in your dataset. The *summary* function will give minimum, maximum, first and third quartiles as well as medians and means for variables which are not strings (characters). Each of these commands can be run for individual variables also. You can refer to an individual variable of a data set by using the **$**, for example, if you wanted to obtain a summary of the a numeric age variable, then you would write **summary(tira.data$age)**.

# str provides an overview of the number of observations and variable types  
str(tira.data$ill)  
  
# summary provides mean, median and max values of your variables (where applicable NAs)  
summary(tira.data$ill)

Codebook of the variables salmon, pork and horseradish show that a few records have the value 9. As in Q3 you are asked to clean these data, recode these values to missing.

#for the rows where salmon is 9, overwrite with NA  
tira.data$salmon[tira.data$salmon == 9] <- NA  
  
#same for horseradish and pork  
tira.data$horseradish[tira.data$horseradish == 9] <- NA  
tira.data$pork[tira.data$pork == 9] <- NA

### Creating labels:

In order to add labels in *R* you have to change variables in to factors. This allows you to specify levels (the order in which categories appear in output) and then label these levels.

#re-write the tira variable as a factor defining levels and labels  
tira.data$tira <- factor(tira.data$tira,   
 levels = c(0, 1),   
 labels = c("No", "Yes")  
 )  
  
  
#re-write the wmousse variable as a factor defining levels and labels  
tira.data$wmousse <- factor(tira.data$wmousse,   
 levels = c(0, 1),   
 labels = c("No", "Yes")  
 )  
  
  
#re-write the dmousse variable as a factor defining levels and labels  
tira.data$dmousse <- factor(tira.data$dmousse,   
 levels = c(0, 1),   
 labels = c("No", "Yes")  
 )

You can label more than one variable at a time using a for-loop

#define the variables you would like to recode  
vars <- c("mousse", "beer", "redjelly",   
 "fruitsalad", "tomato", "mince",   
 "salmon", "horseradish", "chickenwin",   
 "roastbeef", "pork")   
  
#for each var defined in vars above  
for (var in vars) {  
 #select the column of tira.data in square brackets  
 #overwrite with a factor as above  
 tira.data[ , var] <- factor(tira.data[ , var],   
 levels = c(0, 1),   
 labels = c("No", "Yes")  
 )  
}

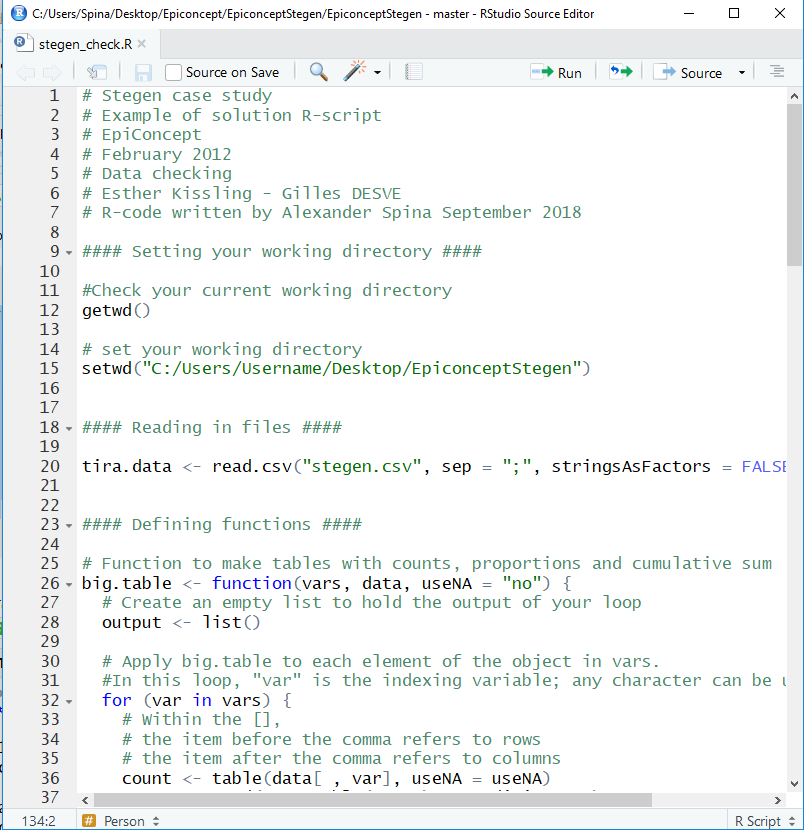
And define different categories.

#define the variables you would like to recode  
vars <- c("tportion", "mportion")   
  
#for each var defined in vars above  
for (var in vars) {  
 #select the column of tira.data in square brackets  
 #overwrite with a factor as above  
 tira.data[ , var] <- factor(tira.data[ , var],   
 levels = c(0, 1, 2, 3),   
 labels = c("None", "One portion",   
 "Two portions", "Three portions")  
 )  
}

### R-scripts

You may also want to develop an R-script in which you will keep all relevant commands and annotate / comment each step of your analysis.

You can select the “+” icon and select R-script from the dropdown (alternatively you could click File > New file > R-script) , insert your command and save with a specific “name.R”. An example is shown below:



You may want to create separate scripts for checking the dataset and for recoding the data (cleaning and creating labels).

## Q4. Describe the outbreak in terms of person and time

Note: A “Place” variable is not available. Use the cleaned dataset stegen1.dta dataset.

## Help task 4

load("stegen1.Rda")  
big.table("sex", tira.data)  
summary(tira.data$age)  
big.table("agegroup", tira.data)   
big.table("ill", tira.data)   
big.table("dateonset", tira.data[tira.data$ill == 1, ])

### Key tables for task 4:

You can create these tables in Word or Excel from the data from the Stata output.

**Table 1. Descriptive epidemiology: Study population by sex**

|  |  |  |
| --- | --- | --- |
| Sex | N | % |
| Male | 152 | 48 |
| Female | 139 | 52 |
| Total | 291 | 100 |

**Table 2. Descriptive epidemiology: Study population by age group**

|  |  |  |
| --- | --- | --- |
| Age group | N | % |
| <30 | 215 | 74 |
| 30+ | 68 | 23 |
| Missing | 8 | 3 |
| Total | 291 | 100 |

**Mean age:** 26 years **Median age:** 20 years **Range:** 12 – 80 years

**Table 3. Descriptive epidemiology: Attack rate**

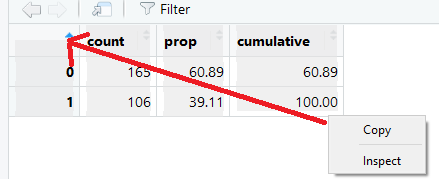
|  |  |  |
| --- | --- | --- |
| Ill | N | % |
| No | 188 | 65 |
| Yes | 103 | 35 |
| Total | 291 | 100 |

**Table 4. Descriptive epidemiology: Cases by date of onset of illness**

|  |  |  |
| --- | --- | --- |
| Ill | N | % |
| 27 June 1998 | 48 | 47 |
| 28 June 1998 | 46 | 45 |
| 29 June 1998 | 8 | 8 |
| Missing | 1 | 1 |
| Total | 103 | 100 |

### Copying tables in to Excel:

To copy tables in to Excel, save your output as an object and can use the *View* function on your output data frame. Then highlight this from the bottom right to the top left, then copy and paste to Excel.



Alternatively if you could use the *write.csv* function to creat a CSV file and open this with Excel.

# Session 2

## Q5. Understanding that investigators did a cohort study, test each of the relevant hypotheses.

Using *R* and the cleaned dataset “stegen1.Rda”:

**Q5a)** Compute food specific attack rates and attack rates by age and sex

**Q5b)** Choose the appropriate measure of association and the appropriate statistical tests and appropriate level of confidence

**Q5c)** Look at proportion of cases exposed

**Q5d)** Compute attributable risk % among exposed

**Q5e)** Search for any dose response if appropriate

**Q5f)** Interpret the results and identify the outbreak vehicle if any.

## Help task 5

### Help Q5a) Calculating attack rate:

There are several ways to do this. The first involves using base-R code, the second is with a user-written function and the third is to install a package.

#load your cleaned dataset  
load("stegen1.Rda")

# The first element will be rows and the 2nd will be columns  
count <- table(tira.data$sex, tira.data$ill, deparse.level = 2)  
  
# Here we select row % of count by including ,1 in the prop.table section  
prop <- round(prop.table(count, 1) \* 100, digits = 2)  
  
# We obtain the denominator using the rowSums function  
denominator <- rowSums(count)[2]  
  
# We combine all the elements together using cbind (binding by columns)  
output <- cbind(Ill = count[2, ], N = denominator, Proportions = prop[2, ])

You could alternatively combine the above code in to a user-written function called *attack.rate*. Here too, you do not need to understand this code at this point.

First run the code to save the function in your environment.

# Function to provide counts, denominator and proportions (equivalent of attack rate)  
attack.rate <- function(exposure, outcome, data, rowcol = "cols") {  
   
 #create an empty list to store results  
 output <- list()  
   
 #for each variable named in exposure  
 for (var in exposure) {  
   
 counts <- table(data[, var], data[, outcome] )  
   
 if (rowcol == "cols") {  
   
   
 #get column proportions  
 prop <- round(prop.table(counts, 1) \* 100, digits = 2)  
   
 #get row totals   
 denominator <- rowSums(counts)[2]  
  
   
 #pull counts together   
 intermediate <- cbind(Ill = counts[2, ], N = denominator, Proportions = prop[2, ])  
   
 }  
   
 if (rowcol == "rows") {  
   
 #get column proportions  
 prop <- round(prop.table(counts, 2) \* 100, digits = 2)  
   
 #get column totals  
 denominator <- colSums(counts)[2]  
   
 #pull counts together   
 intermediate <- cbind(Exposed = counts[ , 2],   
 N = denominator, Proportions = prop[ , 2])  
 }  
   
 if (nrow(counts) > 2) {  
   
 #get column proportions  
 prop <- round(prop.table(counts, 1) \* 100, digits = 2)  
   
 #get row totals   
 denominator <- rowSums(counts)  
  
   
 #pull counts together   
 intermediate <- cbind(Ill = counts[ , 2], N = denominator, Proportions = prop[ , 2])  
   
 }  
   
 #store your output table in the list  
 output[[var]] <- intermediate  
 }  
   
 return(output)  
}

You can now use your function to get attack rates.

# specify the exposure, the outcome and the dataset  
attack.rate( exposure = "sex", outcome = "ill", data = tira.data)

Alternatively you could use the *EpiStats* package written by Epiconcept.

# Install the package if you have not done this yet  
install.packages("EpiStats")

# Load the package to this session   
library(EpiStats)

## Warning: package 'EpiStats' was built under R version 3.5.1

# Use the CS function as this is a cohort study   
CS(tira.data, "ill", "sex")

## Help Q5b) Choose the appropriate measure of association and the appropriate statistical tests and appropriate level of confidence

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.

## Help Q5c) Look at proportion of cases exposed

Here you could use the *attack.rate* function and specify that you would like rows.

# specify rows to get proportion of exposed cases  
attack.rate(exposure = c("sex", "tira", "agegroup"),   
 outcome = "ill", data = tira.data,   
 rowcol = "rows")

Alternatively you could use the CS function and switch exposures and outcomes.

#reversed cs gives you one count  
CS(tira.data, "tira", "ill")

We can see that many cases were exposed to Tiramisu.

## $tira  
## Exposed N Proportions  
## No 7 101 6.93  
## Yes 94 101 93.07

## Help Q5d) Compute attributable risk % among exposed

To calculate the attributable risk % among the exposed, we can use the formula:

The attributable risk % is the proportion of the disease among the exposed, which can be attributed to the exposure (or could have been prevented by eliminating the exposure). We can also find the attributable risk % using the CS function:

#same command as above for CS  
CS(tira.data, "tira", "ill")

|  |  |
| --- | --- |
| Risk factor | Attributable risk % |
| Agegroupagegroup | 5.1 |
| Tiramisutiramisu | 94.5 |
| wmousse | 64.9 |
| dmousse | 77.8 |
| mousse | 79.9 |
| redjelly | 52 |
| fruitsalad | 60 |
| tomato | 22.5 |
| mince | 5.4 |
| salmon | 3.2 |
| horseradish | 20.4 |
| chickenwin | 13.9 |
| pork | 20.1 |

NB: It makes sense to calculate the attributable risk % among the exposed among those with a RR>1. If there is a protective effect (RR<1), then we can calculate the prevented fraction among the exposed:

The prevented fraction among the exposed is mainly used in vaccine studies (where the exposure, the vaccine, is protective). In outbreak investigation studies, we rarely have protective exposure. We need to have verified biological plausibility for an exposure to be a risk factor, otherwise the calculation of the prevented fraction does not make sense.

Here, there are three risk factors for getting ill that are associated with a RR<1: sex, beer and roastbeef. The epidemiologists in the outbreak investigation team decide to assess if drinking beer is protective in this outbreak. We can calculate the prevented fraction among the exposed (see below). For the variables sex and roastbeef, it may be harder to determine any plausible biological reasons for them to be protective factors.

|  |  |
| --- | --- |
| Risk factor | Prevented fraction % |
| Beer | 32.3 |

## Help Q5e) 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  
attack.rate(exposure = "tportion",   
 outcome = "ill",   
 data = tira.data)

## $tportion  
## Ill N Proportions  
## None 7 165 4.24  
## One portion 44 65 67.69  
## Two portions 38 42 90.48  
## Three portions 12 14 85.71

# 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 == "Three portions"] <- "Two portions"  
  
#drop the resulting NA factor level  
tira.data$tportion2 <- droplevels(tira.data$tportion2, NA)

# Tabulate tportion2 variable against illness using attack.rate function  
attack.rate(exposure = "tportion2",   
 outcome = "ill",   
 data = tira.data)

## $tportion2  
## Ill N Proportions  
## None 7 165 4.24  
## One portion 44 65 67.69  
## Two portions 50 56 89.29

Here you can 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 Q5f) Interpret the results and identify the outbreak vehicle if any.

Use the *CSTable* function from the *EpiStats* package. The output is automatically sorted by p-value however you can also get choose to sort by other values. This can be useful when you would like a summary table of attack rates and RRs.

#use the same commands as for CS  
CSTable(tira.data, cases = "ill", exposure = c("sex", "agegroup", "tira",  
 "beer", "mousse", "wmousse",  
 "dmousse", "redjelly", "fruitsalad",  
 "tomato", "mince", "salmon", "horseradish",  
 "chickenwin", "roastbeef", "pork"))

## $df  
## Tot.Exp. Exp.Cases AR% Tot.Unex. Unex.Cases AR% RR  
## tira 121 94 77.69 165 7 4.24 18.31  
## mousse 123 81 65.85 166 22 13.25 4.97  
## wmousse 72 49 68.06 205 49 23.90 2.85  
## dmousse 113 76 67.26 174 26 14.94 4.50  
## redjelly 79 45 56.96 212 58 27.36 2.08  
## fruitsalad 71 46 64.79 220 57 25.91 2.50  
## beer 106 30 28.30 165 69 41.82 0.68  
## tomato 83 35 42.17 208 68 32.69 1.29  
## pork 120 48 40.00 169 54 31.95 1.25  
## horseradish 72 30 41.67 217 72 33.18 1.26  
## sex 152 50 32.89 139 53 38.13 0.86  
## roastbeef 29 8 27.59 262 95 36.26 0.76  
## chickenwin 84 33 39.29 207 70 33.82 1.16  
## mince 87 32 36.78 204 71 34.80 1.06  
## agegroup 68 25 36.76 215 75 34.88 1.05  
## salmon 104 37 35.58 183 63 34.43 1.03  
## CI ll CI ul p(Chi2)  
## tira 8.81 38.04 0.000  
## mousse 3.30 7.48 0.000  
## wmousse 2.13 3.81 0.000  
## dmousse 3.09 6.56 0.000  
## redjelly 1.56 2.79 0.000  
## fruitsalad 1.89 3.31 0.000  
## beer 0.48 0.96 0.024  
## tomato 0.94 1.77 0.127  
## pork 0.92 1.71 0.158  
## horseradish 0.90 1.75 0.192  
## sex 0.63 1.18 0.351  
## roastbeef 0.41 1.40 0.354  
## chickenwin 0.84 1.61 0.377  
## mince 0.76 1.48 0.747  
## agegroup 0.73 1.51 0.777  
## salmon 0.75 1.43 0.844

# Session 3

**Ppt Presentation:**  **Summary of results for food specific attack rates (10 minutes)**

Several food items seem to play a role in the occurrence of illness; tiramisu, dark and white chocolate mousse, fruit salad, and red jelly. They can potentially explain up to respectively (94, 76, 49, 46, and 45 of the 103 cases). 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 occurrence of gastroenteritis.

## Q6. Conduct any relevant stratified analysis, using the dataset stegen1.dta.

**Q6a)** Identify the variables which are potential effect modifiers and confounders: Start by looking if beer is an effect modifier or a confunder, look at the 2X2 tables

**Q6b)** Use the cs command

**Q6c)** To summarize your results, use csinter command

**Q6d)** Then design all appropriate stratification tables

**Q6e)** Stratify on each level taken by the effect modifiers and confounders and compute appropriate measurements to identify confounding and effect modification.

**Q6f)** Interpret the results

## Help Task 6

## Help Q6a) Identify the variables which are potential effect modifiers and confounders: Start by looking if beer is an effect modifier or a confounder, look at the 2X2 tables.

Do a tabulation to show the 2 by 2 tables of each stratum.

# do a three way table with tira  
# drop NAs specifying useNA = "no"   
# put variable names using deparse.level = 2  
table(tira.data$beer, tira.data$ill,   
 tira.data$tira, useNA = "no",  
 deparse.level = 2)

## , , tira.data$tira = No  
##   
## tira.data$ill  
## tira.data$beer No Yes  
## No 83 4  
## Yes 60 3  
##   
## , , tira.data$tira = Yes  
##   
## tira.data$ill  
## tira.data$beer No Yes  
## No 12 63  
## Yes 14 27

## Help Q6c) To summarize your results, use csinter function:

Use the *CSInter* function to summarise. The *CSInter* function produces 2 x 2 tables with stratum specific risk ratios, attributable risk among exposed and population attributable risk.

As a summary it gives the Mantel Haenszel RR and the result of a Woolf test for homogeneity.

# specify CSInter as otherwise, but also choose a by group  
CSInter(tira.data, cases = "ill", exposure = "beer", by = "tira")

## Help Q6d) Then design all appropriate stratification tables

Other potential variables include: - Dark chocolate mousse - White chocolate mousse - Beer - Red jelly - Fruit salad - Age - Sex - Amount of Tiramisu eaten

## Help Q6e) Stratify on each level taken by the effect modifiers and confounders and compute appropriate measurements to identify confounding and effect modification.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | RR stratified for Tiramisu (Yes|No) | M-H Test for homogeneity (p-value) | Crude RR | Adjusted (MH-RR) | % change | Eff.mod or conf? |
| White mousse |  |  |  |  |  |  |
| Dark mousse |  |  |  |  |  |  |
| Beer |  |  |  |  |  |  |
| Red jelly |  |  |  |  |  |  |
| Fruit salad |  |  |  |  |  |  |
| Sex |  |  |  |  |  |  |
| Age |  |  |  |  |  |  |

By stratifying the analysis on tiramisu consumption we can measure the potential protective role of beer among those who ate tiramisu. The following tables show occurrence of gastroenteritis according to beer and tiramisu consumption.

**Table 5** Cases of gastroenteritis among attendants at a high school graduation ceremony by beer and tiramisu consumption, Stegen, Germany, 1998.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| CSInter ill - beer by(tira) | Total | Cases | Risk % | P.est. | Stats | 95%CI-ll | 95%CI-ul |
| tira = Yes | 116 | NA | NA | Risk difference | -0.18 | -0.35 | -0.01 |
| Exposed | 41 | 27 | 65.85 | Risk ratio | 0.78 | 0.62 | 1.00 |
| Unexposed | 75 | 63 | 84.00 | Prev. frac. ex. | 0.22 | 0.00 | 0.38 |
|  | NA | NA | NA | Prev. frac. pop | 0.08 | NA | NA |
| tira = No | 150 | NA | NA | Risk difference | 0.00 | -0.07 | 0.07 |
| Exposed | 63 | 3 | 4.76 | Risk Ratio | 1.04 | 0.24 | 4.47 |
| Unexposed | 87 | 4 | 4.60 | Attrib.risk.exp | 0.03 | -3.16 | 0.78 |
|  | NA | NA | NA | Attrib.risk.pop | 0.01 | NA | NA |
| Missing / Missing % | 25 | 8.6% | NA | NA | NA | NA | NA |

It seems that consumption of beer may reduce the relative effect of tiramisu consumption on occurrence of gastroenteritis. The RR does not significantly differ in the two strata (0.8 vs. 1.0 and confidence intervals overlap). But some effect modification may be present. This association between beer consumption and occurrence of gastroenteritis is however not confounded by Tiramisu consumption since the adjusted RR is 0.8 and the crude RR was 0.7. A similar stratification was conducted assessing dose response for tiramisu consumption among beer drinkers and not.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| CSInter ill - beer by(tportion2) | Total | Cases | Risk % | P.est. | Stats | 95%CI-ll | 95%CI-ul |
| tportion2 = Two portions | 53 | NA | NA | Risk difference | 0.01 | -0.16 | 0.19 |
| Exposed | 19 | 17 | 89.47 | Risk Ratio | 1.01 | 0.83 | 1.23 |
| Unexposed | 34 | 30 | 88.24 | Attrib.risk.exp | 0.01 | -0.20 | 0.19 |
|  | NA | NA | NA | Attrib.risk.pop | 0.01 | NA | NA |
| tportion2 = One portion | 63 | NA | NA | Risk difference | -0.35 | -0.59 | -0.11 |
| Exposed | 22 | 10 | 45.45 | Risk ratio | 0.56 | 0.35 | 0.91 |
| Unexposed | 41 | 33 | 80.49 | Prev. frac. ex. | 0.44 | 0.09 | 0.65 |
|  | NA | NA | NA | Prev. frac. pop | 0.15 | NA | NA |
| tportion2 = None | 150 | NA | NA | Risk difference | 0.00 | -0.07 | 0.07 |
| Exposed | 63 | 3 | 4.76 | Risk Ratio | 1.04 | 0.24 | 4.47 |
| Unexposed | 87 | 4 | 4.60 | Attrib.risk.exp | 0.03 | -3.16 | 0.78 |
|  | NA | NA | NA | Attrib.risk.pop | 0.01 | NA | NA |
| Missing / Missing % | 25 | 8.6% | NA | NA | NA | NA | NA |

# Session 4

**Case study continued** Results suggest that dark and white chocolate as well as fruit salad and red jelly consumption may have contributed to illness (since RR are high even among those who did not eat tiramisu). Such an association can be real (several contaminated food items, use of a single spoon to serve portions) or due to another unidentified confounding factor. Interpretation of results should also be cautious due to the small number of cases involved in this stratified analysis.

## Q7. Conduct a multivariable analysis without taking interaction into account.

**First explore the result of logistic regression**

* Start with the logit command
* Include one exposure variable (the most likely vehicle = tiramisu, outcome = ill)
* Look at the coefficients obtained from the logistical regression. With beta, compute the OR with a calculator and compare with output provided by *R*.

**Then add the other variables in the logistic regression**

## Help task 7

**Perform logistic regression**

* Include one exposure variable (the most likely vehicle = tiramisu, outcome = ill)
* Look at the coefficients obtained from the logistical regression. With beta, compute the OR with a calculator and compare with output provided by *R*
* Compare the univariable analysis results obtained from a 2 x 2 table and those obtained from the logistic regression with one exposure variable.
* Repeat the process for each variable of interest.

Start by running a logistic regression. Using the **generalised linear model** (glm) function with the **logit link** will provide output.

# Create logit regression model with tira as exposure variable  
  
model1 <- glm(ill~tira,  
 data = tira.data,  
 family = binomial(link = "logit"))  
  
# Gives an overview of key elements of the model  
summary(model1)

##   
## Call:  
## glm(formula = ill ~ tira, family = binomial(link = "logit"),   
## data = tira.data)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -1.7320 -0.2944 -0.2944 0.7106 2.5140   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -3.1167 0.3862 -8.071 7e-16 \*\*\*  
## tiraYes 4.3641 0.4436 9.837 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 371.44 on 285 degrees of freedom  
## Residual deviance: 186.41 on 284 degrees of freedom  
## (5 observations deleted due to missingness)  
## AIC: 190.41  
##   
## Number of Fisher Scoring iterations: 5

The odds ratio is equal to the exponential of the coeffecient. In order to get this output it is possible to use the *tidy* function from the *broom* package.

# install broom if you have not already done so  
install.packages("broom")

#load broom to this session   
library(broom)  
  
# Obtaining the key output of the regression model including ORs and CIs  
model1op <- tidy(model1, exponentiate = TRUE, conf.int = TRUE)  
  
# view your cleaned output  
model1op

## term estimate std.error statistic p.value conf.low  
## 1 (Intercept) 0.0443038 0.3861778 -8.070596 6.995604e-16 0.01880027  
## 2 tiraYes 78.5820052 0.4436312 9.837322 7.775296e-23 35.02913548  
## conf.high  
## 1 0.08744858  
## 2 203.48219359

**Then**

* Add a second variable

# We add beer to the model  
model2 <- glm(ill ~ tira + beer,  
 data = tira.data,  
 family = binomial(link = "logit"))  
  
# clean up your output and exponentiate  
model2op <- tidy(model2, exponentiate = TRUE, conf.int = TRUE)  
  
# view your cleaned output  
model2op

## term estimate std.error statistic p.value conf.low  
## 1 (Intercept) 0.0633450 0.4025692 -6.853876 7.187562e-12 0.02615275  
## 2 tiraYes 74.0274332 0.4547373 9.465763 2.914251e-21 32.32622013  
## 3 beerYes 0.4689017 0.4028665 -1.879933 6.011715e-02 0.20957970  
## conf.high  
## 1 0.129585  
## 2 195.934241  
## 3 1.026347

For categorical variables, *R* will automatically take the lowest value as the reference category. Try this with tportion.

# Here we use tportion a factor variable  
model3 <- glm(ill~tportion,   
 data = tira.data,  
 family = binomial(link = "logit"))  
  
# clean up your output and exponentiate  
model3op <- tidy(model3, exponentiate = TRUE, conf.int = TRUE)  
  
# view your cleaned output  
model3op

## term estimate std.error statistic p.value  
## 1 (Intercept) 0.0443038 0.3861778 -8.070596 6.995604e-16  
## 2 tportionOne portion 47.2925137 0.4684865 8.231511 1.848660e-16  
## 3 tportionTwo portions 214.4285567 0.6522646 8.229753 1.875991e-16  
## 4 tportionThree portions 135.4285621 0.8558426 5.735218 9.738688e-09  
## conf.low conf.high  
## 1 0.01880027 0.08744858  
## 2 19.96920519 127.75259273  
## 3 66.51598007 890.06235491  
## 4 30.21489385 992.29211228

We can however change the reference level (for example use 3 portions instead of 0).  
*NB*: reference designates the index and not the value.

tira.data$tportion <- relevel(tira.data$tportion, ref = "Three portions")

**Add more variables?**

We add mousse to the previous model. To simplify matters, we can use the **update** function. In this way, we retain the dataset and family from the previous model and just have to specify the variables to include in the formula.

# Update the previous model with a new formula  
model4 <- update(model2,  
 formula = ill ~ tira + beer + mousse)  
  
#clean and exponentiate  
model4op <- tidy(model4, exponentiate = TRUE, conf.int = TRUE)  
  
#view  
model4op

## term estimate std.error statistic p.value conf.low  
## 1 (Intercept) 0.0512656 0.4285911 -6.931398 4.167012e-12 0.02016546  
## 2 tiraYes 47.7558996 0.4879683 7.922856 2.321159e-15 19.51012482  
## 3 beerYes 0.5129572 0.4092824 -1.631057 1.028783e-01 0.22679653  
## 4 mousseYes 2.3395140 0.4275537 1.987922 4.682037e-02 0.99001885  
## conf.high  
## 1 0.1103337  
## 2 134.6657355  
## 3 1.1393003  
## 4 5.3547326

## Q8. optional: As we are carrying out a cohort study, our measure of effect of choice is the relative risk. Try binomial regression to directly estimate relative risk instead of odds ratio.

## Help Q8

To specify binomial regression use the *glm* command and choose log as a link rather than logit.

# Binomial regression with one independent variable  
 #specify link as log rather than logit  
bin1 <- glm(ill ~ tira,   
 data = tira.data,  
 family = binomial(link = "log"))  
  
#clean output and exponentiate  
bin1op <- tidy(bin1, exponentiate = TRUE, conf.int = TRUE)  
  
#view   
bin1op

## term estimate std.error statistic p.value conf.low  
## 1 (Intercept) 0.04242424 0.3698294 -8.544576 1.290091e-17 0.01845205  
## 2 tiraYes 18.31168808 0.3730249 7.794491 6.466871e-15 9.57966314  
## conf.high  
## 1 0.08042021  
## 2 42.29755568

Binomial regression estimates risk ratios. The risk ratio here for tiramisu is 18. The odds ratio is 79.

## term estimate std.error statistic p.value conf.low  
## 1 (Intercept) 0.0443038 0.3861778 -8.070596 6.995604e-16 0.01880027  
## 2 tiraYes 78.5820052 0.4436312 9.837322 7.775296e-23 35.02913548  
## conf.high  
## 1 0.08744858  
## 2 203.48219359

The odds ratio greatly overestimates the risk ratio in this study. In a cohort study, the odds ratio is close to the relative risk only when the incidence of the outcome of interest is low. In this study, the overall attack rate is 35.4%, which is a high incidence; therefore we see a very big difference between the odds ratio and the relative risk.

## Q9. optional: As it is not straightforward to use the lrtest with binomial regression, use logistic regression to carry out the likelihood ratio test to test between different models.

## Help task 9: Testing between two models

So far we have tested if a model with n variables significantly contributed to the probability of illness. Apart from the first variable we did not check if the addition or the drop of one variable significantly contributed to the outcome.

It is possible to compare two models. However they need to have the same number of observations. This is why in “stegen\_nomissing.Rda” we have deleted all observations with missing values. We now work with 239 observations instead of 291. (This could also be done from the original data set by deleting all observations with missing values). If we wanted to any missings from all variables we could use the *na.omit* funciton; however as we want to specify variables, we need to use the *complete.cases* function.

#drop rows with missing values  
tira.data <- tira.data[complete.cases(tira.data[ , c("ill", "tira", "age",   
 "dmousse", "wmousse", "beer",   
 "fruitsalad", "redjelly", "tportion",   
 "mportion", "salmon", "mince",   
 "tomato", "horseradish", "chickenwin",   
 "roastbeef", "pork")]  
 ) , ]

The following method can now be applied. Using “stegen\_nomissing.Rda”, do a first model with the outcome (ill) and one exposure (tira).

# Only one independent variable  
model5 <- glm(ill~tira,  
 data = tira.data,  
 family = binomial(link = "logit"))  
  
model5op <- tidy(model5, exponentiate = TRUE, conf.int = TRUE)  
model5op

## term estimate std.error statistic p.value conf.low  
## 1 (Intercept) 0.05185185 0.3876173 -7.634759 2.262439e-14 0.02195746  
## 2 tiraYes 74.24999945 0.4617729 9.328044 1.078426e-20 31.89479338  
## conf.high  
## 1 0.1027027  
## 2 198.5873895

Now do a second model with one additional variable (beer).

# As before, we can update the previous model and just write the new formula  
model6 <- update(model5,   
 formula = ill ~ tira + beer)  
  
model6op <- tidy(model6, exponentiate = TRUE, conf.int = TRUE)  
model6op

## term estimate std.error statistic p.value conf.low  
## 1 (Intercept) 0.06872088 0.4057751 -6.598981 4.139926e-11 0.02820815  
## 2 tiraYes 80.26304344 0.4758675 9.215399 3.101214e-20 33.67736191  
## 3 beerYes 0.44030520 0.4410228 -1.859965 6.289043e-02 0.18055390  
## conf.high  
## 1 0.1415055  
## 2 221.5079759  
## 3 1.0315458

To compare two models, we will use the **anova** test, which tests for the difference in the residual deviances between the models. This is equivalent to a likelihood ratio test.

anova(model5, model6, test = "Chisq")

## Analysis of Deviance Table  
##   
## Model 1: ill ~ tira  
## Model 2: ill ~ tira + beer  
## Resid. Df Resid. Dev Df Deviance Pr(>Chi)   
## 1 237 154.51   
## 2 236 150.94 1 3.5643 0.05903 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

If the anova test is statistically significant, this suggests that the addition of beer in the model significantly improves the residual deviance of this model.

The results of the anova (p = 0.0590) suggest a borderline significance (at the 0.05 level) for the addition of the variable beer. Remember this might be a confounder, so this may be a sufficient reason for which you may want to keep it in the model regardless of its p-value in the anova test. For the OR for beer we would therefore decide to keep beer in the model.

Then extend to other variables. Proceed similarly to extend or drop the model according to the anova results.

Keep or drop other variables as needed. Take anova, p-values, magnitude of OR, and the proportion of cases exposed into account in order to decide.

## Q10. Using the likelihood ratio test keep adding variables and testing if they contribute significantly to the model. You can use the table on the next page for this. (optional)

You can use a table similar to the following one to design your model.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Models | Const. | Tira (OR) | p-value | Beer (OR) | p-value | Mousse (OR) | p-value | Sex (OR) | p-value | Log likelihood | ANOVA (with previous model) |
| 1 |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  |
| 4 |  |  |  |  |  |  |  |  |  |  |  |
| 5 |  |  |  |  |  |  |  |  |  |  |  |
| 6 |  |  |  |  |  |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |  |  |  |  |

You can extract each term individually by using the “$” sign. For example you could extract the constant (or intercept) for the entire model. It reflects the odds of disease when all exposures are equal to 0 (note that here the OR is not the OR for the constant but the odds of disease when not exposed to any of the factors).

**Additional** In order to predict probability of outcome according to specific exposures you can use the following predict command:

model7 <- glm(ill ~ sex + tportion + mousse + beer + fruitsalad +   
 redjelly + tomato + mince + salmon + horseradish +   
 chickenwin + roastbeef + pork + age,  
 data = tira.data,  
 family = binomial(link = "logit"))  
  
#   
# newdata1 <- with(tira.data, data.frame(ill = mean(ill), gpa = mean(gpa), rank = factor(1:4)))  
#   
# ## view data frame  
# newdata1  
#   
# newdata1$rankP <- predict(mylogit, newdata = newdata1, type = "response")  
# newdata1  
#   
#   
# predict(model7)

## Q11. Repeat the multivariable analysis including the relevant interactions (optional). Use stegen\_nomissing.Rda

## Help task 11: Look for effect modification

To check for interaction *R* requires that an equal number of non-missing values are available for each variable included.

In “stegen\_nomissing.Rda” the data set was restricted to 239 observations for which the information on variable of interest is available.

Using “stegen\_nomissing.Rda” test for interaction between consumption of beer and tiramisu.

#load your non missing dataset   
load("stegen\_nomissing.Rda")

# Check for interaction between tira and beer  
tirabeer <- glm(ill ~ tira\*beer,   
 data = tira.data,  
 family = binomial(link = "logit"))  
  
tirabeerop <- tidy(tirabeer, exponentiate = TRUE, conf.int = TRUE)  
tirabeerop

## term estimate std.error statistic p.value  
## 1 (Intercept) 0.05194805 0.5127934 -5.767451767 8.047911e-09  
## 2 tiraYes 125.12499910 0.6381127 7.568119633 3.786652e-14  
## 3 beerYes 0.99568966 0.7832563 -0.005515003 9.955997e-01  
## 4 tiraYes:beerYes 0.32190032 0.9386451 -1.207605918 2.271989e-01  
## conf.low conf.high  
## 1 0.01584208 0.1248666  
## 2 39.82751065 504.8496932  
## 3 0.19004231 4.6834923  
## 4 0.05057842 2.1848867

With the star symbol between tira and beer, we are telling *R* to include a factorial interaction. We have to be very careful when interpreting output, when we include an interaction.

The odds ratio associated with “tira=yes” is no longer the comparison of those eating tiramisu and those not eating tiramisu. It is the effect of eating tiramisu when the effect of the other term in the interaction is at its reference value: beer=0. So among those who didn’t drink beer, the odds for eating tiramisu is 125 greater among ill than non-ill persons.

Similiarly, among those who didn’t eat tiramisu, the odds of drinking beer is the same (OR=1, or 0.9956897) among the ill and non-ill. This means that drinking beer was not protective of illness, if no tiramisu was consumed.

The interaction term is 0.3219. We can use this term to rebuild our stratification tables, as in Question 6.

To obtain the association between drinking beer and being ill among those who didn’t eat tiramisu, we can use the odds ratio associated with beer=yes (OR=1, or 0.9956897). To obtain the association between drinking beer and being ill among those who ate tiramisu, we multiply the OR for beer with the interaction coefficient: . We can use the *glht* function from the multcomp package for this, which makes obtaining the confidence intervals easier:

#install the multcomp function if you dont have it already   
install.packages("multcomp")

#load multcomp package to this session   
library(multcomp)

#We can use names to extract the coefficient names   
 # check with: names(coef(tirabeer))   
  
# linfct specifies the required combination:  
 #In this case we want beer and tira and beer:tira=0  
# The odds of illness among those who   
 #drank beer and consumed tiramisu compared to those  
 #who consumed neither tiramisu nor beer  
  
a <- summary(glht(tirabeer, linfct = c("beerYes + tiraYes:beerYes = 0")))  
  
ci <- confint(a)   
  
# Put together (cbind) a table with the exponent of the coefficients and CI, and p-value  
table\_interact <- round(cbind(OR = exp(coef(a)),  
 Interval = exp(ci$confint),  
 Pvalue = a$test$pvalues),  
 digits = 3)  
  
table\_interact

## OR Estimate lwr upr Pvalue  
## beerYes + tiraYes:beerYes 0.321 0.321 0.116 0.883 0.028

Table I

|  |  |  |  |
| --- | --- | --- | --- |
| Tiramisu = Yes | BeerYes BeerNo | OR 0.3205 Reference | 95%CI 0.1-0.9 |
| Tiramisu = No | BeerYes BeerNo | 0.3205 Reference | 0.1-0.9 |

In the above table, the interaction between eating tiramisu and drinking beer is equal to: 0.3205 / 0.9957 = 0.3219 This expresses by how much the OR for beer is multiplied when we go from Tira0 to Tira1. Note that these tables look different than the stratified analysis ones – we are using odds ratios, rather than risk ratios!

Alternatively we could have stratified on beer rather than tira. We can use the following commands:

a <- summary(glht(tirabeer, linfct = c("tiraYes + tiraYes:beerYes = 0")))  
  
ci <- confint(a)   
  
# Put together (cbind) a table with the exponent of the coefficients and CI, and p-value  
table\_interact <- round(cbind(OR = exp(coef(a)),  
 Interval = exp(ci$confint),  
 Pvalue = a$test$pvalues),  
 digits = 3)  
  
table\_interact

## OR Estimate lwr upr Pvalue  
## tiraYes + tiraYes:beerYes 40.278 40.278 10.45 155.244 0

Table II

|  |  |  |  |
| --- | --- | --- | --- |
| Beer = Yes | TiraYes TiraNo | OR 40.2778 Reference | 95%CI 10.4-155.3 |
| Beer = No | TiraYes TiraNo | 125.125 Reference | 35.8-437.0 |

In the above table, the multiplicative interaction between drinking beer and eating tiramisu is AGAIN equal to:

An alternative table would be:

Table III

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tira | Beer | Cases | Controls | OR |
| 1 | 1 | 25 | 12 | 40.1042 |
| 0 | 1 | 3 | 58 | 0.9957 |
| 1 | 0 | 52 | 8 | 125.125 |
| 0 | 0 | 4 | 77 | Reference |

**References**

Rodriguez L and Kirkwood B: Case-control designs in the study of common diseases: updates on the demise of the rare disease assumption and the choice of sampling scheme for controls. Int J Epidemiol 1990; 19: 205-13

Rothman KJ: Epidemiology. An introduction. Oxford University Press2002. P 74ff

Hosmer and Lesmeshow: Applied logistic regression. Wiley Inter Science, John Willey & Sons, 1989.

Svend Juul: An introduction to Stata for Health Researchers. Stata Press, Stata Corp LP.

Hauri A.M et al. An outbreak of Salmonella Enteritidis gastroenteritis following tiramisu consumption at a high school graduation ceremony, Germany, June 1998.

Nguyen TM, Ilef D, Jarraud S et al. A community-wide outbreak of legionnaires disease linked to industrial cooling towers–how far can contaminated aerosols spread? J Infect Dis. 2006; 193(1):102-111.

Nguyen TM, Ganiayre F, Lapidus N et al Enquête cas-témoins de cas groupés communautaires de légionellose, Pas-de-Calais, France, novembre 2003 - janvier 2004 BEH 2006 ;36-37 :181-182.

Lectures

* Study design: EPIET lecture notes
* Confounding and effect modification: EPIET lecture notes
* Logistic regression: EPIET lecture notes
* David Prieto lectures, EPIET, 2006
* Rachid Salmi lectures, University Victor Segalene, Bordeaux, EPIET course 2002-4.
* Jean Claude Desenclos lectures, EPIET, 2002-2004
* Thomas Grein Lectures EPIET, 2002-2004
* Alain Moren lectures, EPIET & Epiconcept, 2005-2007