SimOb Inject 14 - Univariable analysis

## 1. Install packages and load libraries

# Load the required libraries into the current R session:  
pacman::p\_load(rio,   
 here,   
 tidyverse,   
 skimr,  
 plyr,  
 janitor,  
 lubridate,  
 gtsummary,   
 flextable,  
 officer,  
 epikit,   
 apyramid,   
 scales,  
 EpiStats)

## 2. Import your data

# Import the raw data set:  
linelist <- rio::import(here::here("data", "Copenhagen\_clean2.rds"))

## 3. Risk Ratios

### a) Calculate 95% CI Risk Ratios

# You could use the EpiStats package for each food item  
CS(linelist, "case", "tuna")

$df1  
 Cases Non Cases Total Risk  
Exposed 156 115 271 0.58  
Unexposed 60 42 102 0.59  
Total 216 157 373 0.58  
  
$df2  
 Point estimate 95%CI ll 95%CI ul  
Risk difference -0.01 -0.12 0.10  
Risk ratio 0.98 0.81 1.19  
Prev. frac. ex. 0.02 -0.19 0.19  
Prev. frac. pop 0.02 NA NA  
chi2(1) 0.05 NA NA  
Pr>chi2 0.826 NA NA

CS(linelist, "case", "shrimps")

$df1  
 Cases Non Cases Total Risk  
Exposed 150 105 255 0.59  
Unexposed 65 52 117 0.56  
Total 215 157 372 0.58  
  
$df2  
 Point estimate 95%CI ll 95%CI ul  
Risk difference 0.03 -0.08 0.14  
Risk ratio 1.06 0.87 1.28  
Attr. frac. ex. 0.06 -0.14 0.22  
Attr. frac. pop 0.04 NA NA  
chi2(1) 0.35 NA NA  
Pr>chi2 0.553 NA NA

CS(linelist, "case", "green")

$df1  
 Cases Non Cases Total Risk  
Exposed 123 93 216 0.57  
Unexposed 83 60 143 0.58  
Total 206 153 359 0.57  
  
$df2  
 Point estimate 95%CI ll 95%CI ul  
Risk difference -0.01 -0.12 0.09  
Risk ratio 0.98 0.82 1.18  
Prev. frac. ex. 0.02 -0.18 0.18  
Prev. frac. pop 0.01 NA NA  
chi2(1) 0.04 NA NA  
Pr>chi2 0.837 NA NA

CS(linelist, "case", "veal")

$df1  
 Cases Non Cases Total Risk  
Exposed 201 137 338 0.59  
Unexposed 14 22 36 0.39  
Total 215 159 374 0.57  
  
$df2  
 Point estimate 95%CI ll 95%CI ul  
Risk difference 0.21 0.04 0.37  
Risk ratio 1.53 1.01 2.32  
Attr. frac. ex. 0.35 0.01 0.57  
Attr. frac. pop 0.32 NA NA  
chi2(1) 5.64 NA NA  
Pr>chi2 0.018 NA NA

# And so one

# You can save time (and probably typos!) by creating a vector for food variables...  
food\_vars <- c("tuna", "shrimps", "green", "veal",   
 "pasta", "rocket", "sauce", "bread",  
 "champagne", "beer", "redwine", "whitewine")  
  
# ...and using EpiStats::CSTable() to run all variables together!  
CSTable(linelist, "case", c(food\_vars))

$df  
 Tot.Exp. Exp.Cases AR% Tot.Unex. Unex.Cases AR% RR CI ll CI ul  
pasta 338 202 59.76 36 13 36.11 1.65 1.06 2.58  
veal 338 201 59.47 36 14 38.89 1.53 1.01 2.32  
champagne 316 187 59.18 48 21 43.75 1.35 0.97 1.89  
rocket 211 114 54.03 154 95 61.69 0.88 0.73 1.04  
sauce 149 90 60.40 198 106 53.54 1.13 0.94 1.36  
beer 281 166 59.07 78 41 52.56 1.12 0.89 1.42  
redwine 80 42 52.50 259 150 57.92 0.91 0.72 1.14  
shrimps 255 150 58.82 117 65 55.56 1.06 0.87 1.28  
whitewine 260 150 57.69 98 54 55.10 1.05 0.85 1.29  
bread 342 196 57.31 29 16 55.17 1.04 0.74 1.46  
tuna 271 156 57.56 102 60 58.82 0.98 0.81 1.19  
green 216 123 56.94 143 83 58.04 0.98 0.82 1.18  
 p(Chi2)  
pasta 0.006  
veal 0.018  
champagne 0.044  
rocket 0.144  
sauce 0.202  
beer 0.303  
redwine 0.393  
shrimps 0.553  
whitewine 0.659  
bread 0.823  
tuna 0.826  
green 0.837

### b) Prepare the RR table for publication

rr\_tbl <- CSTable(linelist, "case", c(food\_vars)) %>%   
 as.data.frame() %>%   
 flextable() %>%   
 set\_header\_labels(  
 values = c("Total exposed",   
 "Cases exposed",   
 "AR among exposed",   
 "Total unexposed",  
 "Cases unexposed",  
 "AR among unexposed",  
 "RR",   
 "95% lower CI",   
 "95% upper CI",  
 "p-value"))

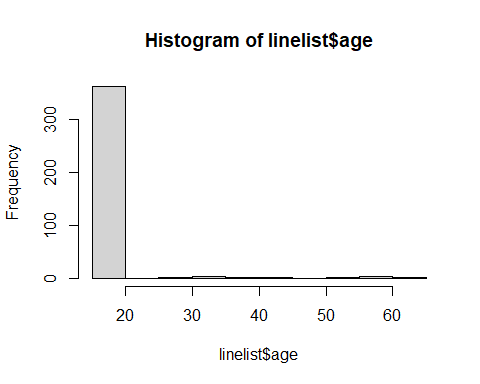
## 4. Hypothesis tests for other variables

#### a) age

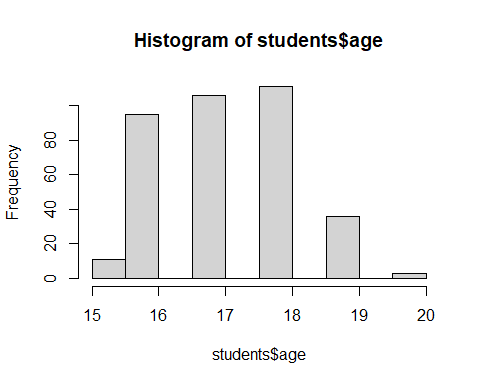
# Check if age overall follows a normal distribution:  
shapiro.test(linelist$age)

Shapiro-Wilk normality test  
  
data: linelist$age  
W = 0.31302, p-value < 2.2e-16

# Can simply have a look at  
hist(linelist$age)



# Looking only at the students:  
students <- linelist %>%   
 filter(group == "student")  
hist(students$age)



Age overall (nor within the students’ group) is not normally distributed. We could use

# Perform Wilcoxon rank sum test on age and sex:  
wilcox.test(age ~ case, data = linelist)

Wilcoxon rank sum test with continuity correction  
  
data: age by case  
W = 15934, p-value = 0.1512  
alternative hypothesis: true location shift is not equal to 0

#### b) sex

CS(linelist, "case", "sex")

$df1  
 Cases Non Cases Total Risk  
Exposed 99 65 164 0.60  
Unexposed 117 96 213 0.55  
Total 216 161 377 0.57  
  
$df2  
 Point estimate 95%CI ll 95%CI ul  
Risk difference 0.05 -0.05 0.15  
Risk ratio 1.10 0.92 1.31  
Attr. frac. ex. 0.09 -0.08 0.24  
Attr. frac. pop 0.04 NA NA  
chi2(1) 1.12 NA NA  
Pr>chi2 0.290 NA NA

#### c) class

CS(linelist, "case", "class")

$df1  
 Cases Non Cases Total Risk  
Exposed 57 44 101 0.56  
Unexposed 68 63 131 0.52  
Total 125 107 232 0.54  
  
$df2  
 Point estimate 95%CI ll 95%CI ul  
Risk difference 0.05 -0.08 0.17  
Risk ratio 1.09 0.86 1.38  
Attr. frac. ex. 0.08 -0.17 0.27  
Attr. frac. pop 0.04 NA NA  
chi2(1) 0.47 NA NA  
Pr>chi2 0.493 NA NA

#### d) group

CS(linelist, "case", "group")

$df1  
 Cases Non Cases Total Risk  
Exposed 210 152 362 0.58  
Unexposed 6 9 15 0.40  
Total 216 161 377 0.57  
  
$df2  
 Point estimate 95%CI ll 95%CI ul  
Risk difference 0.18 -0.07 0.43  
Risk ratio 1.45 0.78 2.71  
Attr. frac. ex. 0.31 -0.29 0.63  
Attr. frac. pop 0.30 NA NA  
chi2(1) 1.91 NA NA  
Pr>chi2 0.167 NA NA

## 5. Dose Response

#### a) Veal

# Perform the Wilcoxon rank sum test on number of veal portions  
wilcox.test(as.numeric(vealD) ~ case, data = linelist)

Wilcoxon rank sum test with continuity correction  
  
data: as.numeric(vealD) by case  
W = 15820, p-value = 0.1526  
alternative hypothesis: true location shift is not equal to 0

#### b) Pasta

# Perform the Wilcoxon rank sum test on number of pasta portions  
wilcox.test(as.numeric(pastaD) ~ case, data = linelist)

Wilcoxon rank sum test with continuity correction  
  
data: as.numeric(pastaD) by case  
W = 14338, p-value = 0.004135  
alternative hypothesis: true location shift is not equal to 0