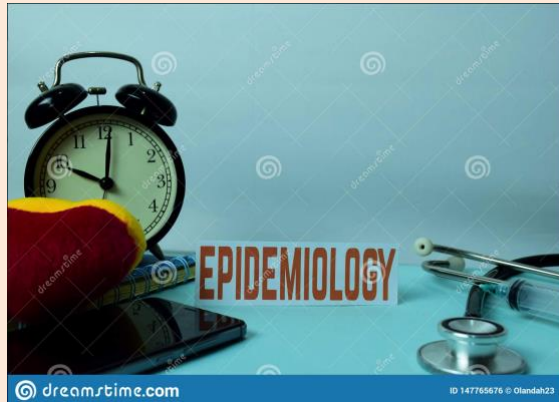


BIOSTATS 640 – Introduction to R

Fall 2023

<https://people.umass.edu/biep640w/webpages/demonstrations.html>



05  
Epidemiological Tables  
*October 6, 2023*

Dataset used  
*None!*

Packages Used:

tidyverse, DescTools, summarytools, ggplot2, epiR, vcd, epiDisplay

		Page
1	Highlights of Lesson 04– Introduction to R Markdown .....	2
2	<b>2x2 Table</b> a. Enter a table of counts ..... b. <code>Untable( )</code> to produce individual observations. .... c. Estimation, CI's, and Hypothesis Tests ..... d. Visualization .....	3 4 5 9
3	<b>Stratified Analysis of K 2x2 Tables</b> a. Enter K 2x2 tables of counts ..... b. <code>Untable( )</code> to produce individual observations ..... c. Estimation, CI's, and Hypothesis Tests . .... d. Visualization .....	12 13 14 16
4	Some Good Resources .....	20

## 1. Highlights of Lesson 04 Introduction to R Markdown

### *R Script v R Markdown*

Use **R script** to develop your R code.

Use **R Markdown** to perform analyses, generate output, and produce reports

### *R Markdown is comprised of narrative + chunks + utility to execute R commands + utilities to render/knit*

**Narrative is text.** Accomplishing things such as bold, bulleted lists, tables is done using the R Markup language (not MS WORD!!)

A **chunk is R code** that is sandwiched between a **beginning** and an **end** and the enclosure is shaded grey.

- A chunk begins with ```` {r }`
- A chunk ends with `````

**Rendering/knitting** an R Markdown refers to the execution of the R Markdown and produces (depending on your selections) a display of your code + the output + a pretty “marked up” narrative.

### *Suggestions for chunks*

**Keep it simple** – Each chunk does just one thing (e.g., produces a tabulation, produces one graph, etc.)

**Proceed chunk by chunk** – For each chunk, cycle: write code, execute, revise, re-execute.

## 2. 2x2 Table

### Illustration

Source: Woodward M. *Epidemiology Study Design and Data Analysis*, 2nd edition, Chapman & Hall/CRC, 2005. Example 2.15 on page 105

This was a study of the performance of the dipstick test (clinical screen) for the detection of a UTI in a sample of 229 patients. True status of UTI present is defined by a positive culture test.

Dipstick Screen Result		Culture	
		Positive	Negative
Positive		<b>a</b> = 84	<b>b</b> = 43
Negative		<b>c</b> = 10	<b>d</b> = 92

### 2a. Enter a table of counts

Use `rbind( )` to enter counts “row by row”

Use `as.table( )` to save object as a table

Use `dimnames( )` to make it readable!

```
table1 <- as.table(rbind(c(84,43),c(10,92)))
cat("\n2x2 Table of Counts - Unlabelled\n")

##
## 2x2 Table of Counts - Unlabelled

table1                                     # show, unlabelled
##      A  B
## A 84 43
## B 10 92

dimnames(table1)<- list(
  Dipstick=c("Dipstick positive","Dipstick negative"), # Rowvar = c("Label1", "Label2")
  Culture=c("True positive","True negative")           # Colvar = c("Label1", "Label2")
)
cat("\n2x2 Table of Counts - Labelled\n")
##
## 2x2 Table of Counts - Labelled

table1                                     # show, Labelled
##      Culture
## Dipstick  True positive True negative
## Dipstick positive      84         43
## Dipstick negative      10         92
```

## 2b. Untable() to produce individual observations

**Why bother?** Individual observations are required for data visualizations in ggplot.

Use Untable() in the package {DescTools}

Use ctable() in the package {summarytools}

```
library(DescTools) # Untable() in {DescTools}
library(summarytools) # ctable() in {summarytools}

df.table1 <- Untable(table1) # Untable() to convert to individual observations
str(df.table1) # str() to check dataset structure
## 'data.frame': 229 obs. of 2 variables:
## $ Dipstick: Factor w/ 2 levels "Dipstick positive",...: 1 1 1 1 1 1 1 1 1 ...
## $ Culture : Factor w/ 2 levels "True positive",...: 1 1 1 1 1 1 1 1 1 ...
## - attr(*, "out.attrs")=List of 2
## ..$ dim : Named int [1:2] 2 2
## ..$ attr(*, "names")= chr [1:2] "Dipstick" "Culture"
## ..$ dimnames:List of 2
## ..$ Dipstick: chr [1:2] "Dipstick=Dipstick positive" "Dipstick=Dipstick negative"
## ..$ Culture : chr [1:2] "Culture=True positive" "Culture=True negative"

ctable(df.table1$Culture,df.table1$Dipstick, # check that dataframe, table match
       prop='n',
       totals=FALSE)
## Cross-Tabulation
## Culture * Dipstick
## Data Frame: df.table1
##
## -----
##          Dipstick  Dipstick positive  Dipstick negative
## Culture
## True positive          84             10
## True negative          43             92
## -----
```

## 2c. Estimation, CI's, and Hypothesis Tests

### 2c.1 Estimation - Basic

Use `ctable()` in the package `{summarytools}`

```
library(summarytools) # ctable( )

with(df.table1,
      ctable(x = Culture,
             y = Dipstick,
             prop = "r")) # x = row variable
                        # y = column variable
                        # choices are "n", "r", "c", "t"

## Cross-Tabulation, Row Proportions
## Culture * Dipstick
## Data Frame: df.table1
##
## -----
##           Dipstick  Dipstick positive  Dipstick negative  Total
## Culture
## True positive      84 (89.4%)         10 (10.6%)      94 (100.0%)
## True negative      43 (31.9%)         92 (68.1%)     135 (100.0%)
## Total              127 (55.5%)        102 (44.5%)     229 (100.0%)
## -----
```

### 2c.2 Diagnostic Testing (sensitivity, specificity, etc.)

Use `epi.tests()` in the package `{epiR}`

```
library(epiR) # epi.tests( )

epi.tests(table1, conf.level=0.90)

##           Outcome +  Outcome -  Total
## Test +           84           43    127
## Test -           10           92    102
## Total            94          135    229
##
## Point estimates and 90% CIs:
## -----
## Apparent prevalence *      0.55 (0.50, 0.61)
## True prevalence *          0.41 (0.36, 0.47)
## Sensitivity *              0.89 (0.83, 0.94)
## Specificity *              0.68 (0.61, 0.75)
## Positive predictive value * 0.66 (0.59, 0.73)
## Negative predictive value * 0.90 (0.84, 0.95)
## Positive likelihood ratio    2.81 (2.26, 3.48)
## Negative likelihood ratio    0.16 (0.09, 0.26)
## False T+ proportion for true D- * 0.32 (0.25, 0.39)
## False T- proportion for true D+ * 0.11 (0.06, 0.17)
## False T+ proportion for T+ *   0.34 (0.27, 0.41)
## False T- proportion for T- *   0.10 (0.05, 0.16)
## Correctly classified proportion * 0.77 (0.72, 0.81)
## -----
## * Exact CIs
```

### 2c.3 Estimates and CI of Relative Risk and Odds Ratio

OddsRatio() in the package {DescTools}

RelRisk() in the package {DescTools}

```
library(DescTools)

OddsRatio(table1, conf.level=.95)

## odds ratio      lwr.ci      upr.ci
## 17.972093      8.498092     38.008076

RelRisk(table1, conf.level=.95)

## rel. risk      lwr.ci      upr.ci
## 6.746457      3.806903     12.362369
```

### 2c.4 Hypothesis Tests of No Association – Fisher Exact and Chi Square

chisq.test() that comes pre-installed

fisher.test() that comes pre-installed

```
# Fisher Exact Test
fisher.test(table1)                                # input is table of counts

##
## Fisher's Exact Test for Count Data
##
## data:  table1
## p-value < 0.00000000000000022
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  8.168663 42.123867
## sample estimates:
## odds ratio
## 17.69702

fisher.test(df.table1$Dipstick, df.table1$Culture)  # input is table from individual observations

##
## Fisher's Exact Test for Count Data
##
## data:  df.table1$Dipstick and df.table1$Culture
## p-value < 0.00000000000000022
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  8.168663 42.123867
## sample estimates:
## odds ratio
## 17.69702
```

```
# Approximate test using the Chi Square
chisq.test(table1) # default is continuity corrected

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: table1
## X-squared = 71.886, df = 1, p-value < 0.0000000000000022

chisq.test(table1, correct=FALSE) # turn OFF continuity correction

##
## Pearson's Chi-squared test
##
## data: table1
## X-squared = 74.196, df = 1, p-value < 0.0000000000000022

chisq.test(df.table1$Dipstick, df.table1$Culture) # input is table from individual observations

##
## Pearson's Chi-squared test with Yates' continuity correction
##
## data: df.table1$Dipstick and df.table1$Culture
## X-squared = 71.886, df = 1, p-value < 0.0000000000000022
```

## 2c.5 Hypothesis Tests of Association Using Relative Risk and Odds Ratio

epi.2by2() in the package {epiR}

### 2c.5. Hypothesis Tests of Association Using Relative Risk and Odds Ratio

```
library(epiR)

epi.2by2(table1, method = "cohort.count", conf.level = 0.95)

##           Outcome +      Outcome -      Total      Inc risk *      Odds
## Exposed +           84           43          127           66.1      1.953
## Exposed -           10           92          102           9.8      0.109
## Total              94          135          229          41.0      0.696
##
## Point estimates and 95% CIs:
## -----
## Inc risk ratio              6.75 (3.70, 12.31)
## Odds ratio                  17.97 (8.50, 38.01)
## Attrib risk in the exposed * 56.34 (46.29, 66.39)
## Attrib fraction in the exposed (%) 85.18 (72.95, 91.88)
## Attrib risk in the population * 31.24 (22.65, 39.84)
## Attrib fraction in the population (%) 76.12 (58.82, 86.15)
## -----
## Uncorrected chi2 test that OR = 1: chi2(1) = 74.196 Pr>chi2 = <0.001
## Fisher exact test that OR = 1: Pr>chi2 = <0.001
## Wald confidence limits
## CI: confidence interval
## * Outcomes per 100 population units
```

```

epi.2by2(table1, method = "case.control", conf.level = 0.95)

##           Outcome +      Outcome -      Total      Prevalence *      Odds
## Exposed +           84           43          127          66.1      1.953
## Exposed -           10           92          102           9.8      0.109
## Total              94          135          229          41.0      0.696
##
## Point estimates and 95% CIs:
## -----
## Odds ratio                                17.97 (8.50, 38.01)
## Attrib fraction (est) in the exposed (%)    94.35 (87.76, 97.63)
## Attrib fraction (est) in the population (%)  84.39 (71.64, 91.41)
## -----
## Uncorrected chi2 test that OR = 1: chi2(1) = 74.196 Pr>chi2 = <0.001
## Fisher exact test that OR = 1: Pr>chi2 = <0.001
## Wald confidence limits
## CI: confidence interval
## * Outcomes per 100 population units

```



## 2d. Visualization

### 2d.1 bar graph – basic: separate panels using facet\_wrap

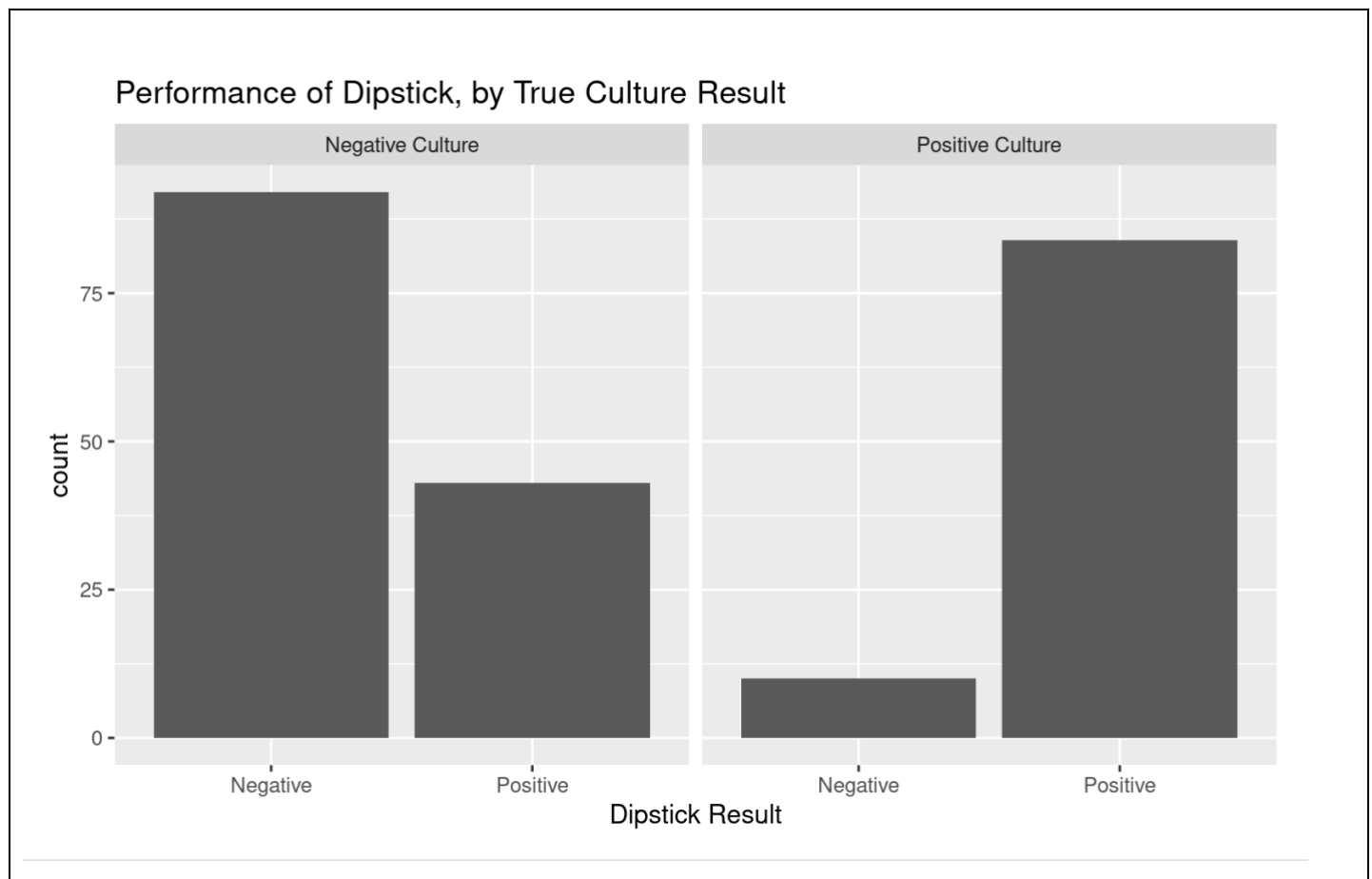
`geom_bar( )` in the package `{ggplot2}`

```
library(ggplot2)

# Preliminary: User sets order of factor levels explicitly (as desired for graph)
df.table1$Dipstickf <- factor(df.table1$Dipstick,
                             levels=c("Dipstick negative", "Dipstick positive"),
                             labels=c("Negative", "Positive"))
df.table1$Culturef <- factor(df.table1$Culture,
                             levels=c("True negative", "True positive"),
                             labels=c("Negative Culture", "Positive Culture"))

ggplot(data=df.table1) +
  aes(x=Dipstickf) + # x= OUTCOME
  geom_bar() +
  facet_wrap(~Culturef) + # facet_wrap(~PREDICTOR) for side-by-side

ggtitle("Performance of Dipstick, by True Culture Result") +
  xlab("Dipstick Result")
```

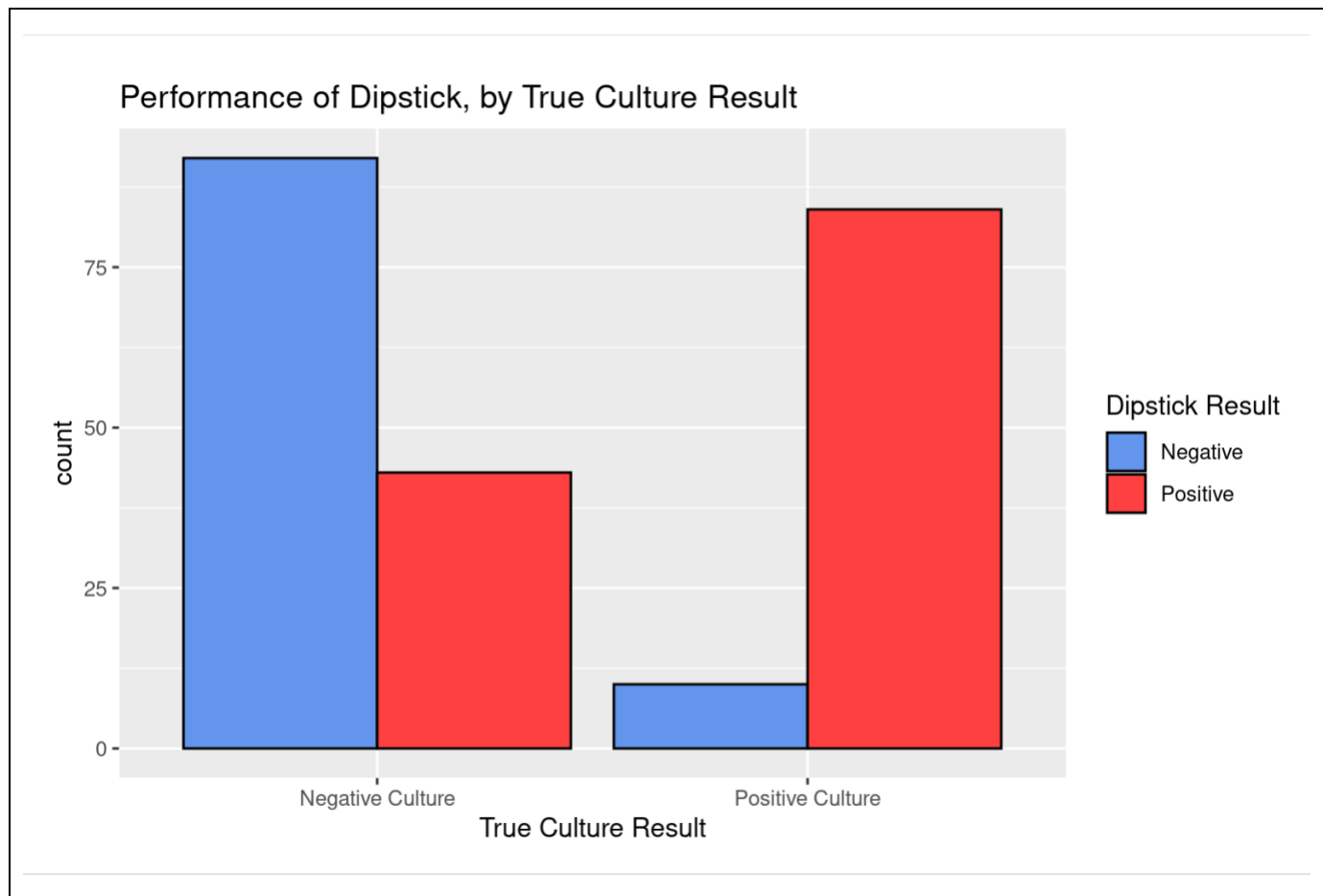


## 2d.2 bar graph – basic: One panel

`geom_bar()` in the package `{ggplot2}`

```
library(ggplot2)
```

```
ggplot(data=df.table1) +  
  aes(x=Culturef) + # x= Predictor  
  aes(fill=Dipstickf) + # fill= OUTCOME  
  geom_bar(position="dodge", # "dodge" for side-by-side or "stack"  
  color="black") + # bar borders  
  
  # User sets display of dipstick result  
  scale_fill_manual(values=c("cornflowerblue", "brown1"), # User sets colors of negative, positive  
  name="Dipstick Result") + # User sets legend title  
  
  ggtitle("Performance of Dipstick, by True Culture Result") +  
  xlab("True Culture Result")
```

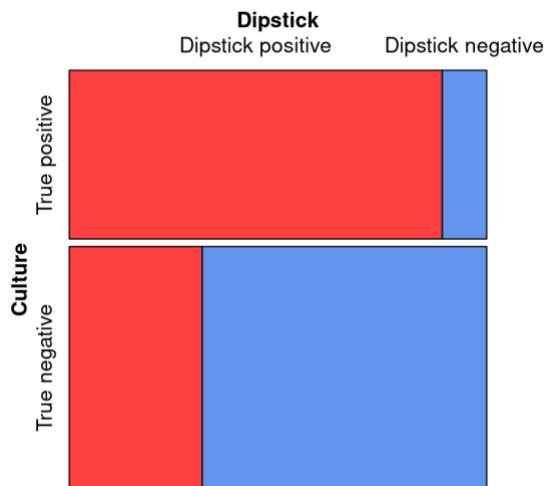


### 2d.3 mosaic – basic

mosaic() in the package {vcd}

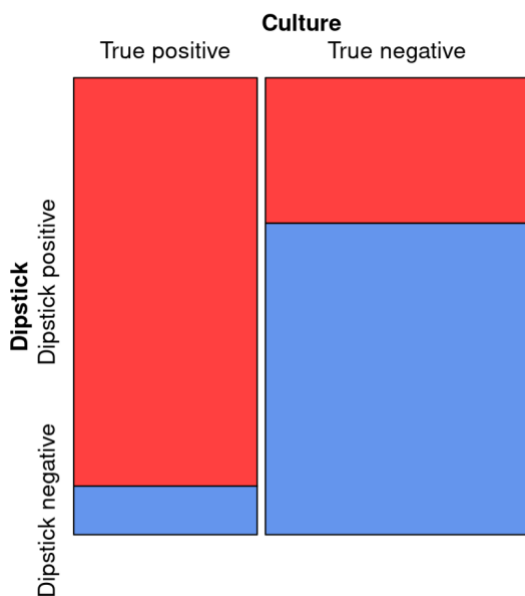
```
library(vcd)
# Row percents total 100%
mosaic(data=df.table1,
        Dipstick ~ Culture,
        highlighting_fill = c("brown1", "cornflowerblue"))
```

*# MODEL: Outcome ~ Predictor*  
*# User chooses colors for positive, negative*



```
# Column percents total 100%
mosaic(data=df.table1,
        Dipstick ~ Culture,
        highlighting_fill = c("brown1", "cornflowerblue"),
        direction=c("v", "h"))
```

*# MODEL: Outcome ~ Predictor*  
*# User chooses colors for positive, negative*



### #3. Stratified Analysis of K 2x2 Tables

#### Stratum: Smokers

		Myocardial Infarction (MI)	
		mi	NON-mi
Coffee	5+ cups	1011	390
	0 cups	81	77

*The down arrows are a reminder that counts are entered by columns!!!*

#### Stratum: NON-smokers

		Myocardial Infarction (MI)	
		mi	NON-mi
Coffee	5+ cups	383	365
	0 cups	66	123

### 3a. Enter K 2x2 tables of counts

Use array() to enter counts “column by column”

Use dimnames() to make it readable!

```
tablek2x2 <- array(c( 1011, 81, 390, 77,
                     383, 66, 365, 123),
                  dim = c(2, 2, 2))
# 2x2 table for smokers, by COLUMN (a,c,b,d)
# 2x2 table for non-smokers, by COLUMN
# KEY: c(#rows, # columns, # strata)

dimnames(tablek2x2) <- list(
  COFFEE = c("5+ cups", "0 cups"),
  MI = c("mi", "NON-mi"),
  STRATUM = c("Smokers", "NON-smokers"))
# row var name and value labels
# column var name and value labels
# stratum var name and strata labels

tablek2x2
# show

## , , STRATUM = Smokers
##
##      MI
## COFFEE  mi NON-mi
## 5+ cups 1011  390
## 0 cups   81   77
##
## , , STRATUM = NON-smokers
##
##      MI
## COFFEE  mi NON-mi
## 5+ cups 383  365
## 0 cups  66  123
```

### 3b. Untable() to produce individual observations

Use Untable() in the package {DescTools}

Check using stby() in the package {summarytools}

```
library(DescTools)
library(summarytools)

dataframek2x2 <- Untable(tablek2x2)
with(dataframek2x2,
      stby(data = list(x = COFFEE, y = MI),
           INDICES = STRATUM, FUN = ctable))
```

*# counts -> individual observations*  
*# CHECK*  
*# x=EXPOSURE var, y=DISEASE var*  
*# INDICES = STRATUM var*

```
## Cross-Tabulation, Row Proportions
## COFFEE * MI
## Data Frame: dataframek2x2
## Group: STRATUM = Smokers
##
```

	MI	mi	NON-mi	Total
COFFEE				
5+ cups	1011 (72.2%)	390 (27.8%)	1401 (100.0%)	
0 cups	81 (51.3%)	77 (48.7%)	158 (100.0%)	
Total	1092 (70.0%)	467 (30.0%)	1559 (100.0%)	

```
##
```

## Group: STRATUM = NON-smokers

```
##
```

	MI	mi	NON-mi	Total
COFFEE				
5+ cups	383 (51.2%)	365 (48.8%)	748 (100.0%)	
0 cups	66 (34.9%)	123 (65.1%)	189 (100.0%)	
Total	449 (47.9%)	488 (52.1%)	937 (100.0%)	

```
##
```

### 3c. Estimation, CI's, and Hypothesis Tests

#### 3c.1. Stratum Specific RR and CI

```
library(summarytools)
library(tidyverse)

nonsmokers <- dataframek2x2 %>%
  filter(STRATUM=="NON-smokers")
with(nonsmokers,
  ctable(COFFEE,MI,prop = 'n', OR=TRUE, RR=TRUE))

## Cross-Tabulation
## COFFEE * MI
## Data Frame: nonsmokers
##
##
## -----
##           MI      mi    NON-mi    Total
## COFFEE
## 5+ cups      383      365      748
## 0 cups        66      123      189
## Total       449      488      937
## -----
##
## -----
## Odds Ratio   Lo - 95%   Hi - 95%
## -----
##           1.96         1.40         2.72
## -----
##
## -----
## Risk Ratio   Lo - 95%   Hi - 95%
## -----
##           1.47         1.19         1.80
## -----

smokers <- dataframek2x2 %>%
  filter(STRATUM=="Smokers")
with(smokers,
  ctable(COFFEE,MI,prop = 'n', OR=TRUE, RR=TRUE))

## Cross-Tabulation
## COFFEE * MI
## Data Frame: smokers
##
##
## -----
##           MI      mi    NON-mi    Total
## COFFEE
## 5+ cups     1011      390     1401
## 0 cups        81       77      158
## Total     1092      467     1559
## -----
##
## -----
## Odds Ratio   Lo - 95%   Hi - 95%
## -----
##           2.46         1.77         3.44
## -----
##
```

```
## -----
## Risk Ratio   Lo - 95%   Hi - 95%
## -----
##      1.41      1.20      1.64
## -----
```

### 3c.2 Hypothesis Test of Null: Homogeneity of Association

```
library(epiDisplay)

epiDisplay::mhor(mhtable=tablek2x2,decimal=2,
  graph=FALSE,
  design="case control")

##
## Stratified analysis by STRATUM
##      OR lower lim. upper lim.      P value
## STRATUM Smokers      2.46      1.74      3.49 0.0000001970115
## STRATUM NON-smokers  1.95      1.39      2.77 0.0000626982391 # The CI's overlap
## M-H combined      2.18      1.72      2.76 0.0000000000407
##
## M-H Chi2(1) = 43.58 , P value = 0
## Homogeneity test, chi-squared 1 d.f. = 0.93 , P value = 0.334 # The null is NOT rejected
```

### 3c.3 Hypothesis Test of Null: Common OR = 1

```
library(epiDisplay)

epiDisplay::mhor(mhtable=tablek2x2,decimal=2,
  graph=FALSE,
  design="case control")

##
## Stratified analysis by STRATUM
##      OR lower lim. upper lim.      P value
## STRATUM Smokers      2.46      1.74      3.49 0.0000001970115
## STRATUM NON-smokers  1.95      1.39      2.77 0.0000626982391
## M-H combined      2.18      1.72      2.76 0.0000000000407
##
## M-H Chi2(1) = 43.58 , P value = 0 # The null IS rejected
## Homogeneity test, chi-squared 1 d.f. = 0.93 , P value = 0.334
```

### 3d. Visualization

#### 3d.1 Percent MI (95% CI) by coffee exposure, stratified by Smoking, separate panels

```
library(tidyverse) # mutate() in {tidyverse}
library(ggplot2)

# REQUIRED: Create 0/1 occurrence of MI to permit calculation of percent with mi
dataframek2x2 <- dataframek2x2 %>%
  mutate(mi01 = ifelse(MI=="mi",1,0),
         na.rm=TRUE)

# AS YOU LIKE: create new factor version of COFFEE with levels ordered as you want for graph
dataframek2x2$coffeef <- factor(dataframek2x2$COFFEE,
                                levels=c("0 cups","5+ cups"),
                                labels=c("0 cups","5+ cups"))

# CALCULATIONS needed for plot
plotdf <- dataframek2x2 %>%
  group_by(STRATUM, coffeef) %>%
  summarise(
    n_stratum = sum(!is.na(mi01)), # sample size in stratum
    mean_mi = mean(mi01, na.rm=TRUE), # percent with mi is mean of mi01
    sd_mi = sd(mi01, na.rm=TRUE), # standard deviation
    se_mi = sd_mi/sqrt(n_stratum), # standard error
    ci95_mi = 1.96*se_mi # 1.96*SE for a 95% CI using Z-score
  )

plotdf # Check
## # A tibble: 4 x 7
## # Groups:   STRATUM [2]
##   STRATUM    coffeef n_stratum mean_mi sd_mi se_mi ci95_mi
##   <fct>      <fct>    <int>    <dbl> <dbl> <dbl>    <dbl>
## 1 Smokers    0 cups      158    0.513 0.501 0.0399 0.0782
## 2 Smokers    5+ cups     1401    0.722 0.448 0.0120 0.0235
## 3 NON-smokers 0 cups      189    0.349 0.478 0.0348 0.0681
## 4 NON-smokers 5+ cups      748    0.512 0.500 0.0183 0.0358

# Plot
ggplot(data=plotdf) +
  aes(x=coffeef) + # x = Predictor (exposure)
  aes(y=mean_mi) + # y = Outcome (disease)

  geom_errorbar( # geom_errorbar() for CI limits
    aes(ymin=mean_mi - ci95_mi,
        ymax=mean_mi + ci95_mi),
    width=.05) +

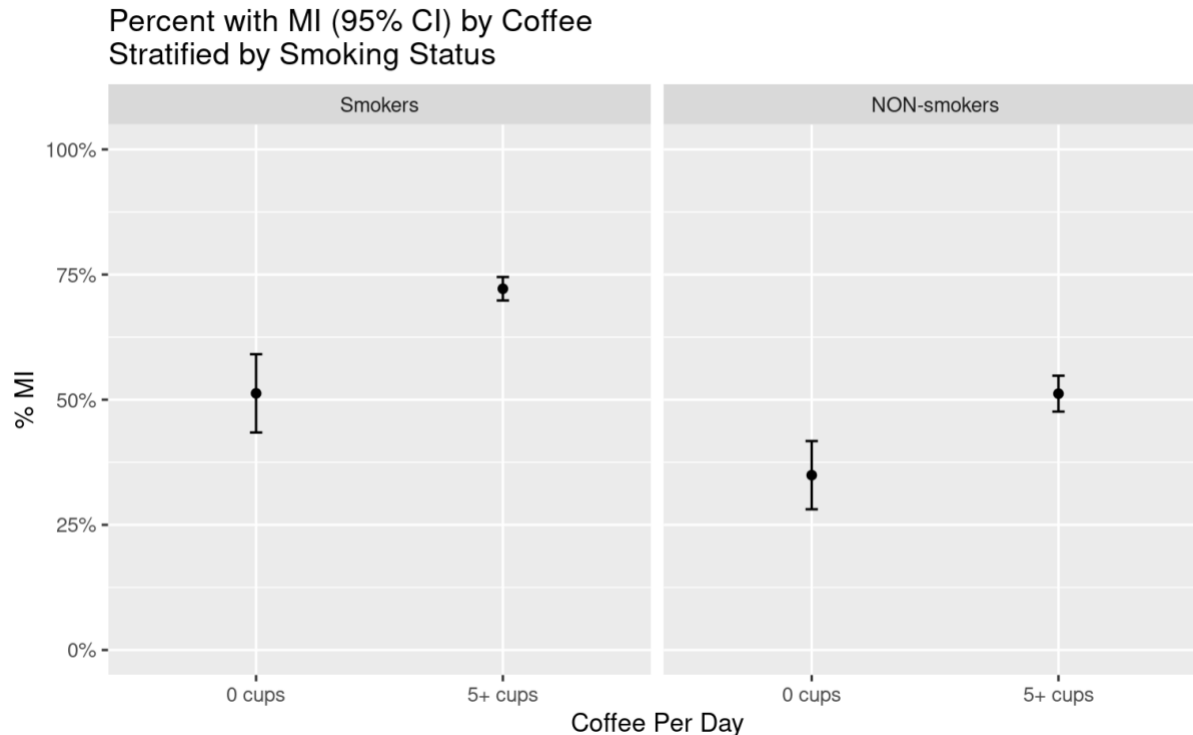
  geom_point() + # geom_point() for point estimate

  facet_wrap(~STRATUM) + # Separate plot for each stratum

  scale_y_continuous(limits=c(0,1), # TIP. User sets y-axis limits explicitly
                    labels=scales::percent) + # Required for display of percents

  ggtitle("Percent with MI (95% CI) by Coffee\nStratified by Smoking Status") +
  xlab("Coffee Per Day") +
  ylab("% MI")
```





### 3d.2 Percent MI (95% CI) by coffee exposure, stratified by Smoking, single panel

```
library(ggplot2)

pd = position_dodge(.2)                                     # RECOMMENDED to line things up

ggplot(data=plotdf) +
  aes(x=STRATUM) +
  aes(y=mean_mi) +
  aes(color=coffee) +

  geom_errorbar(aes(ymin = mean_mi - ci95_mi,
                    ymax = mean_mi + ci95_mi),
                width = 0.1,
                size = 0.5,
                position = pd) +

  geom_point(size = 2,
             position = pd) +

  # User sets display of exposure
  scale_color_manual(values=c("cornflowerblue", "brown1"),
                    name="Coffee Per Day") +

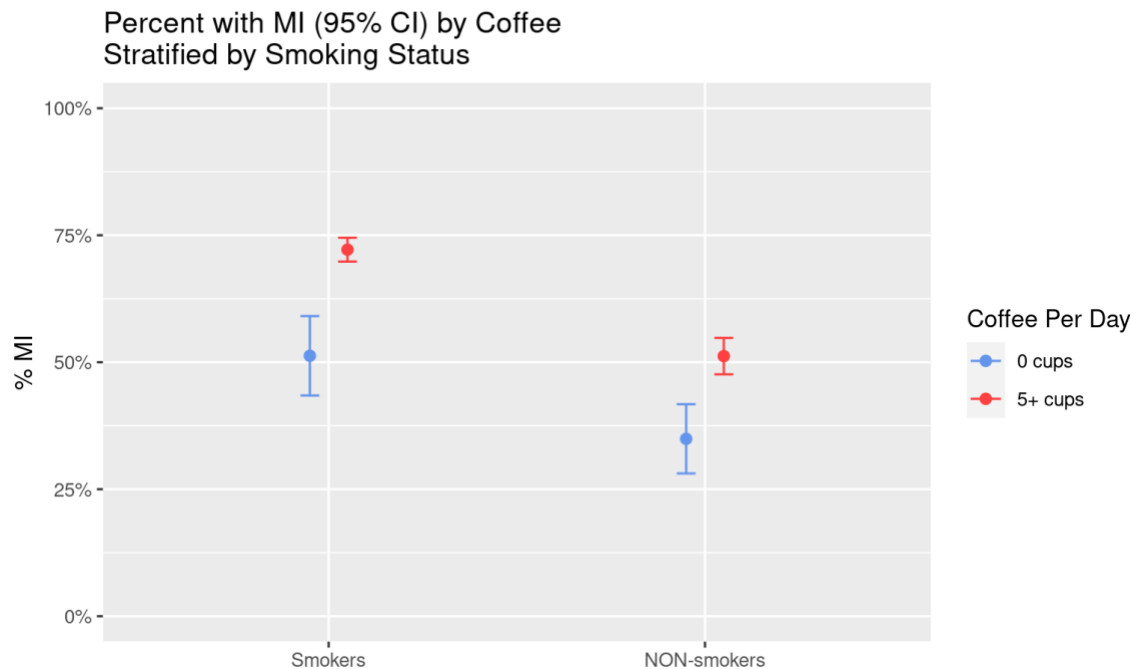
  # User defined y-axis
  scale_y_continuous(limits=c(0,1),
                    labels=scales::percent) +

  ggtitle("Percent with MI (95% CI) by Coffee\nStratified by Smoking Status") +
  xlab(" ") +
  ylab("% MI")
```

# x = Stratification variable  
# y = Outcome (disease)  
# color = Exposure (coffee)

# User sets legend colors of 0 cups, 5+ cups  
# User sets legend title

# TIP! Set y-axis limits explicitly  
# Required for display of percents



### 3d.3 Relative Odds (OR) with 95% CI, Crude and Stratified

```
library(tidyverse)
library(ggplot2)
library(DescTools)
```

```
# REQUIRED: Create 0/1 occurrence of COFFEE so that calculation of OR is correct
dataframek2x2 <- dataframek2x2 %>%
  mutate(coffee01 = ifelse(COFFEE=="5+ cups",1,0),
         na.rm=TRUE)
```

```
# Step 1: calculate point estimates and CI limits
```

```
calculations_all <- dataframek2x2 %>%
  mutate(
    group = "Crude",
    group = factor(group),
    point = (fisher.test(dataframek2x2$coffee01,dataframek2x2$mi01,
                        conf.level=.95))$estimate,
    lower = (fisher.test(dataframek2x2$coffee01,dataframek2x2$mi01,
                        conf.level=.95))$conf.int[1],
    upper = (fisher.test(dataframek2x2$coffee01,dataframek2x2$mi01,
                        conf.level=.95))$conf.int[2]
  ) %>%
  dplyr::select(group, point,lower,upper)
```

```
# Obtain crude OR w CI
```

```
calculations_strata <- dataframek2x2 %>%
  group_by(STRATUM) %>%
  mutate (
    group = STRATUM,
    point = (fisher.test(coffee01,mi01,conf.level=.95))$estimate,
    lower = (fisher.test(coffee01,mi01,conf.level=.95))$conf.int[1],
    upper = (fisher.test(coffee01,mi01,conf.level=.95))$conf.int[2],
  ) %>%
  dplyr::select(group, point,lower,upper)
```

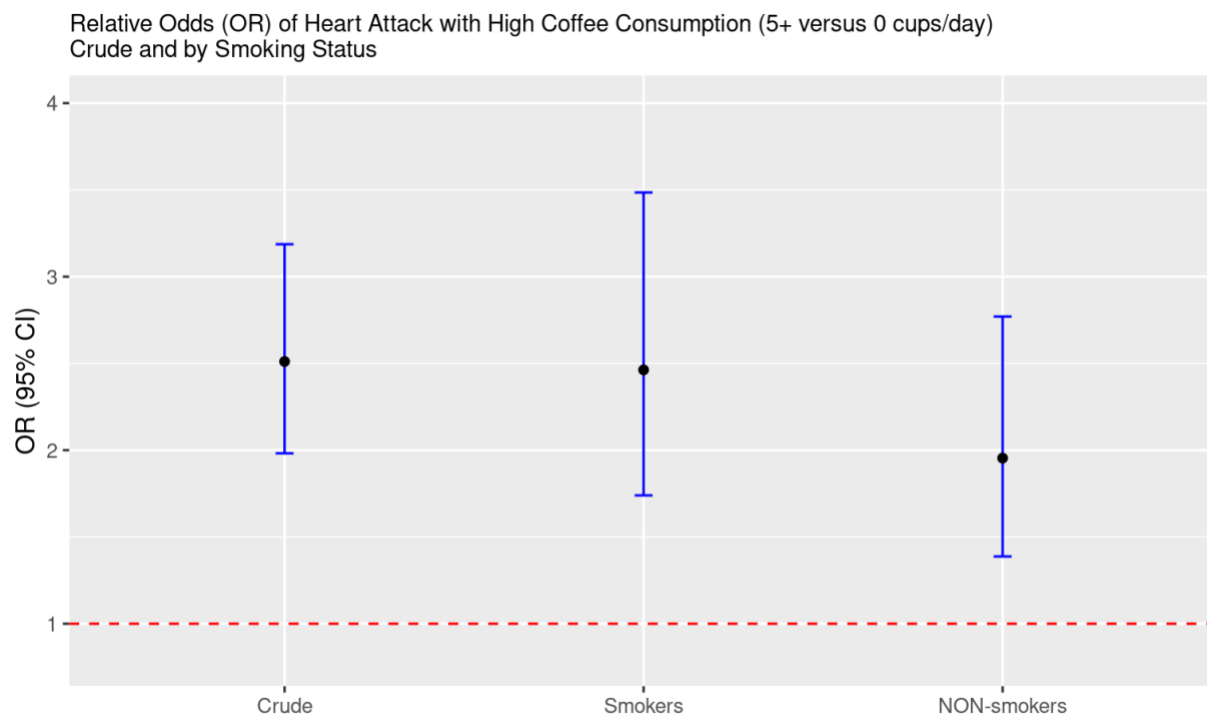
```
# Obtain stratified OR w CI
```

```
# Step 2: Combine crude OR w CI and stratified OR w CI in a single dataframe using bind_rows( )
plotting_df <- bind_rows(calculations_all, calculations_strata) # bind_rows( ) to append dataframes
plotting_df <- plotting_df[!duplicated(plotting_df), ] # unique obs only!!!
plotting_df <- plotting_df %>%
  dplyr::select(group, point, lower, upper) # keep ONLY the variables for plot

plotting_df # Check.

##           group   point   lower   upper
## 1           Crude 2.511074 1.981860 3.186916
## 2497        Smokers 2.462693 1.739417 3.485132
## 4056 NON-smokers 1.954177 1.387406 2.770220

# Step 3: plot
ggplot(data=plotting_df) +
  aes(x=group) +
  aes(y=point) +
  geom_errorbar(aes(ymin=lower,
                    ymax=upper),
               width=.05,
               color="blue") +
  geom_point(color="black") +
  geom_hline(yintercept = 1, linetype = 2, colour = "red") +
  scale_y_continuous(limits=c(0.8, 4),
                     breaks=c(1,2,3,4)) +
  ggtitle("Relative Odds (OR) of Heart Attack with High Coffee Consumption (5+ versus 0 cups/day)\nCrude and
by Smoking Status") +
  xlab(" ") +
  ylab("OR (95% CI)") +
  theme(plot.title = element_text(size=10)) # change size of title so it all fits!
```



#### 4. Some Good Resources

- 1. (Source: *Signorell HWZ University of Applied Sciences and Business Administration, Zurich*)  
Tables in R ([pdf, 29 pp](#))