EPID674 Epidemiologic Data Analysis using R

Hypothesis Testing in R

Kelly Bakulski, Lauren Middleton

Last compiled on December 15, 2021

Table of Contents

### Install new packages

### Load packages

##### Load these packages for the current session  
library(here)  
library(ggcorrplot)  
library(gtsummary)  
library(epiDisplay)  
library(sjlabelled)  
library(tidyverse)

# Load data

# Check the file path  
here("nhanes\_dataset.rda")

## [1] "/cloud/project/nhanes\_dataset.rda"

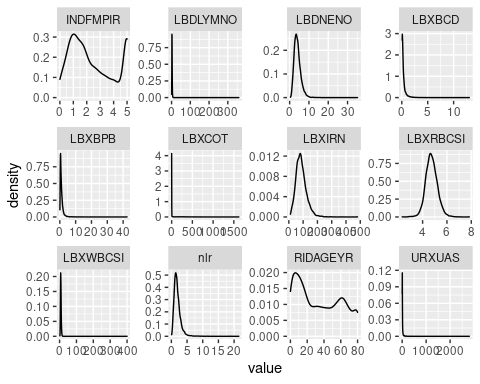
# Load the saved R data  
load(here("nhanes\_dataset.rda"))

# Hypothesis testing in R

## Before you begin: Remember to check your distributions/assumptions

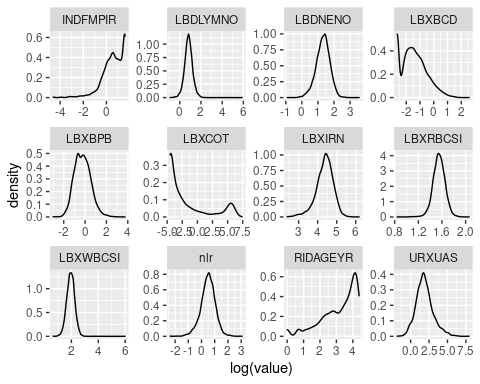
# Select the continuous variables to check and transform dataset to long form  
longer\_data <- nhanes\_dataset %>%  
 dplyr::select(RIDAGEYR,  
 INDFMPIR,  
 LBDLYMNO,  
 LBDNENO,  
 LBXRBCSI,  
 LBXWBCSI,  
 nlr,  
 LBXBCD,  
 LBXBPB,  
 LBXCOT,  
 LBXIRN,  
 URXUAS) %>%  
 remove\_all\_labels() %>%  
 pivot\_longer(everything(), names\_to = "variable", values\_to = "value")   
  
# Plot the distributions of the raw values  
longer\_data %>%  
 ggplot(aes(x = value)) +  
 geom\_density() +  
 facet\_wrap(~ variable, scales = "free")

## Warning: Removed 25934 rows containing non-finite values (stat\_density).



# Are any of these distributions normal?  
  
  
# Plot the distributions of the log transformed values  
longer\_data %>%  
 ggplot(aes(x = log(value))) +  
 geom\_density() +  
 facet\_wrap(~ variable, scales = "free")

## Warning: Removed 26382 rows containing non-finite values (stat\_density).



# Are any of the log transformed distributions normal?  
  
# Be sure to use your understanding of the distributions when deciding between parametric and non-parametric tests

## Correlation tests: Two continuous variables

# Correlation test between two continuous variables (default method is Pearson, a parametric test)  
cor.test(nhanes\_dataset$LBXIRN, nhanes\_dataset$LBXRBCSI)

##   
## Pearson's product-moment correlation  
##   
## data: nhanes\_dataset$LBXIRN and nhanes\_dataset$LBXRBCSI  
## t = 10.89, df = 5917, p-value < 2.2e-16  
## alternative hypothesis: true correlation is not equal to 0  
## 95 percent confidence interval:  
## 0.1151102 0.1650627  
## sample estimates:  
## cor   
## 0.1401757

# Can request spearman, a non-parametric test for non-normally distributed variables  
cor.test(nhanes\_dataset$RIDAGEYR, nhanes\_dataset$INDFMPIR, method = "spearman")

## Warning in cor.test.default(nhanes\_dataset$RIDAGEYR, nhanes\_dataset$INDFMPIR, :  
## Cannot compute exact p-value with ties

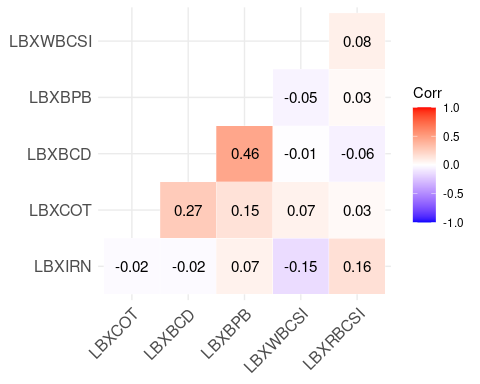
##   
## Spearman's rank correlation rho  
##   
## data: nhanes\_dataset$RIDAGEYR and nhanes\_dataset$INDFMPIR  
## S = 7.2895e+10, p-value < 2.2e-16  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## 0.1530888

## Correlation matrix

# Select the columns of interest  
nhanes\_chems <- nhanes\_dataset %>%  
 dplyr:: select(LBXIRN,  
 LBXCOT,  
 LBXBCD,  
 LBXBPB,  
 LBXWBCSI,  
 LBXRBCSI)  
  
  
# Calculate the correlations, use spearman for non-normally distributed variables  
chem\_correlations <- cor(nhanes\_chems,  
 use = "pairwise.complete.obs",  
 method = "spearman")  
  
# View the correlation matrix  
chem\_correlations

## LBXIRN LBXCOT LBXBCD LBXBPB LBXWBCSI  
## LBXIRN 1.00000000 -0.01926272 -0.02056100 0.06665300 -0.15056021  
## LBXCOT -0.01926272 1.00000000 0.27129009 0.14709065 0.07444648  
## LBXBCD -0.02056100 0.27129009 1.00000000 0.46261731 -0.01119554  
## LBXBPB 0.06665300 0.14709065 0.46261731 1.00000000 -0.05077459  
## LBXWBCSI -0.15056021 0.07444648 -0.01119554 -0.05077459 1.00000000  
## LBXRBCSI 0.15953053 0.02781844 -0.06204764 0.02625293 0.08120104  
## LBXRBCSI  
## LBXIRN 0.15953053  
## LBXCOT 0.02781844  
## LBXBCD -0.06204764  
## LBXBPB 0.02625293  
## LBXWBCSI 0.08120104  
## LBXRBCSI 1.00000000

# Plot the correlations with ggcorrplot  
ggcorrplot(chem\_correlations,  
 type = "lower",  
 outline.col = "white",  
 lab = TRUE)



cor.test(nhanes\_dataset$URXUAS, nhanes\_dataset$nlr, method="spearman")

## Warning in cor.test.default(nhanes\_dataset$URXUAS, nhanes\_dataset$nlr, method =  
## "spearman"): Cannot compute exact p-value with ties

##   
## Spearman's rank correlation rho  
##   
## data: nhanes\_dataset$URXUAS and nhanes\_dataset$nlr  
## S = 2676518001, p-value = 0.1255  
## alternative hypothesis: true rho is not equal to 0  
## sample estimates:  
## rho   
## 0.03035675

## Check your understanding

What is the relationship between urinary arsenic (LBXUAS) and neutrophil/lymphocyte ratio (nlr)? \* Make a new code chunk \* What type of variables are they? What are their distributions? \* What is the appropriate test to run? \* Run the test and interpret the output

## T-test: Independent variable dichotomous, dependent variable continuous, parametric test

# The relationship between sex and iron concentration:  
# What do you expect?  
  
# Calculate mean of iron concentration by sex  
nhanes\_dataset %>%  
 group\_by(sex) %>%  
 summarise(mean\_iron = mean(LBXIRN, na.rm = TRUE)) %>%  
 ungroup()

## # A tibble: 2 × 2  
## sex mean\_iron  
## <fct> <dbl>  
## 1 Male 93.8  
## 2 Female 79.1

# T-test of iron and sex: Parametric test with normally distributed dependent variable  
t.test(nhanes\_dataset$LBXIRN ~ nhanes\_dataset$sex) #What do you get?

##   
## Welch Two Sample t-test  
##   
## data: nhanes\_dataset$LBXIRN by nhanes\_dataset$sex  
## t = 15.678, df = 5901.7, p-value < 2.2e-16  
## alternative hypothesis: true difference in means between group Male and group Female is not equal to 0  
## 95 percent confidence interval:  
## 12.83508 16.50353  
## sample estimates:  
## mean in group Male mean in group Female   
## 93.8048 79.1355

# Do they match?  
  
# Calculated within a bivariate table  
nhanes\_dataset %>%  
 dplyr::select(sex,  
 LBXIRN) %>%  
 tbl\_summary(by = sex, #stratify by sex  
 statistic = all\_continuous() ~ "{mean} ({sd})",  
 missing\_text = "Missing (n)"  
 ) %>%  
 add\_p(LBXIRN ~ "t.test") %>%  
 modify\_header(label ~ "\*\*Variable\*\*") %>%  
 bold\_labels()

## Table printed with {flextable}, not {gt}. Learn why at  
## http://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html  
## To suppress this message, include `message = FALSE` in the code chunk header.

| Variable | Male, N = 4,5571 | Female, N = 4,6971 | p-value2 |
| --- | --- | --- | --- |
| **Iron frozen, Serum (ug/dL)** | 94 (36) | 79 (36) | <0.001 |
| Missing (n) | 1,683 | 1,649 |  |
| 1Mean (SD) | | | |
| 2Welch Two Sample t-test | | | |

## Working with a non-normally distributed variable

## T-test and Wilcoxon test: Independent variable dichotomous, dependent variable continuous

# The relationship between sex and blood Pb  
nhanes\_dataset %>%  
 group\_by(sex) %>%  
 summarise(mean\_lead = mean(LBXBPB, na.rm = TRUE),  
 mean\_log\_lead = mean(log(LBXBPB), na.rm = TRUE)) %>%  
 ungroup()

## # A tibble: 2 × 3  
## sex mean\_lead mean\_log\_lead  
## <fct> <dbl> <dbl>  
## 1 Male 1.25 -0.100  
## 2 Female 0.924 -0.374

# T-test: Be sure to log transform blood Pb variable before calculating parametric test statistics  
t.test(log(nhanes\_dataset$LBXBPB) ~ nhanes\_dataset$sex)

##   
## Welch Two Sample t-test  
##   
## data: log(nhanes\_dataset$LBXBPB) by nhanes\_dataset$sex  
## t = 15.17, df = 6837, p-value < 2.2e-16  
## alternative hypothesis: true difference in means between group Male and group Female is not equal to 0  
## 95 percent confidence interval:  
## 0.2383423 0.3090824  
## sample estimates:  
## mean in group Male mean in group Female   
## -0.1004491 -0.3741614

# Wilcoxon test: Non parametric test, no need to transform variables  
wilcox.test(nhanes\_dataset$LBXBPB ~ nhanes\_dataset$sex)

##   
## Wilcoxon rank sum test with continuity correction  
##   
## data: nhanes\_dataset$LBXBPB by nhanes\_dataset$sex  
## W = 7127720, p-value < 2.2e-16  
## alternative hypothesis: true location shift is not equal to 0

## ANOVA: Independent variable 3+ categories, dependent variable continuous, parametric test

# The relationship between age and blood Pb:  
nhanes\_dataset %>%  
 group\_by(age\_groups) %>%  
 summarise(mean\_lead = mean(LBXBPB, na.rm = TRUE),  
 mean\_log\_lead = mean(log(LBXBPB), na.rm = TRUE)) %>%  
 ungroup()

## # A tibble: 5 × 3  
## age\_groups mean\_lead mean\_log\_lead  
## <fct> <dbl> <dbl>  
## 1 [0,16] 0.566 -0.760  
## 2 (16,32] 0.738 -0.601  
## 3 (32,48] 1.03 -0.277  
## 4 (48,64] 1.45 0.143  
## 5 (64,80] 1.64 0.292

# ANOVA test  
aov(log(nhanes\_dataset$LBXBPB) ~ nhanes\_dataset$age\_groups)

## Call:  
## aov(formula = log(nhanes\_dataset$LBXBPB) ~ nhanes\_dataset$age\_groups)  
##   
## Terms:  
## nhanes\_dataset$age\_groups Residuals  
## Sum of Squares 1177.830 2798.378  
## Deg. of Freedom 4 6879  
##   
## Residual standard error: 0.6378089  
## Estimated effects may be unbalanced  
## 2370 observations deleted due to missingness

anova(aov(log(nhanes\_dataset$LBXBPB) ~ nhanes\_dataset$age\_groups))

## Analysis of Variance Table  
##   
## Response: log(nhanes\_dataset$LBXBPB)  
## Df Sum Sq Mean Sq F value Pr(>F)   
## nhanes\_dataset$age\_groups 4 1177.8 294.457 723.84 < 2.2e-16 \*\*\*  
## Residuals 6879 2798.4 0.407   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

# How do we interpret the p-value from an ANOVA test?  
  
# Pairwise t-test  
pairwise.t.test(log(nhanes\_dataset$LBXBPB), nhanes\_dataset$age\_groups)

##   
## Pairwise comparisons using t tests with pooled SD   
##   
## data: log(nhanes\_dataset$LBXBPB) and nhanes\_dataset$age\_groups   
##   
## [0,16] (16,32] (32,48] (48,64]  
## (16,32] 9e-11 - - -   
## (32,48] <2e-16 <2e-16 - -   
## (48,64] <2e-16 <2e-16 <2e-16 -   
## (64,80] <2e-16 <2e-16 <2e-16 5e-10   
##   
## P value adjustment method: holm

# Pairwise t-test with bonferroni adjustment  
pairwise.t.test(log(nhanes\_dataset$LBXBPB), nhanes\_dataset$age\_groups,  
 p.adj = "bonferroni")

##   
## Pairwise comparisons using t tests with pooled SD   
##   
## data: log(nhanes\_dataset$LBXBPB) and nhanes\_dataset$age\_groups   
##   
## [0,16] (16,32] (32,48] (48,64]  
## (16,32] 4.5e-10 - - -   
## (32,48] < 2e-16 < 2e-16 - -   
## (48,64] < 2e-16 < 2e-16 < 2e-16 -   
## (64,80] < 2e-16 < 2e-16 < 2e-16 5.0e-09  
##   
## P value adjustment method: bonferroni

# Comparing two categorical variables

## Chi-square test: Parametric test

# The relationship between sex and education:  
table(nhanes\_dataset$sex, nhanes\_dataset$education)

##   
## Less than high school Less than 5th grade High school or GED  
## Male 1050 570 745  
## Female 1005 572 735  
##   
## More than high school  
## Male 1480  
## Female 1705

# Chi-square test  
chisq.test(nhanes\_dataset$sex, nhanes\_dataset$education)

##   
## Pearson's Chi-squared test  
##   
## data: nhanes\_dataset$sex and nhanes\_dataset$education  
## X-squared = 13.195, df = 3, p-value = 0.004234

# Calculated within a bivariate table  
nhanes\_dataset %>%  
 dplyr::select(sex,  
 education) %>%  
 tbl\_summary(by = sex, #stratify by sex  
 digits = list(all\_categorical() ~ c(0, 1)),  
 missing\_text = "Missing (n)"  
 ) %>%  
 add\_p() %>%  
 modify\_header(label ~ "\*\*Variable\*\*") %>%  
 bold\_labels()

## Table printed with {flextable}, not {gt}. Learn why at  
## http://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html  
## To suppress this message, include `message = FALSE` in the code chunk header.

| Variable | Male, N = 4,5571 | Female, N = 4,6971 | p-value2 |
| --- | --- | --- | --- |
| **education** |  |  | 0.004 |
| Less than high school | 1,050 (27.3%) | 1,005 (25.0%) |  |
| Less than 5th grade | 570 (14.8%) | 572 (14.2%) |  |
| High school or GED | 745 (19.4%) | 735 (18.3%) |  |
| More than high school | 1,480 (38.5%) | 1,705 (42.4%) |  |
| Missing (n) | 712 | 680 |  |
| 1n (%) | | | |
| 2Pearson's Chi-squared test | | | |

## Fisher’s test: Non-parametric test

# The relationship between sex and iron status:  
table(nhanes\_dataset$sex, nhanes\_dataset$iron\_status)

##   
## Deficient Excessive Normal  
## Male 427 91 2356  
## Female 949 51 2048

# Fisher exact test  
fisher.test(nhanes\_dataset$sex, nhanes\_dataset$iron\_status)

##   
## Fisher's Exact Test for Count Data  
##   
## data: nhanes\_dataset$sex and nhanes\_dataset$iron\_status  
## p-value < 2.2e-16  
## alternative hypothesis: two.sided

# Calculated within a bivariate table  
nhanes\_dataset %>%  
 dplyr::select(sex,  
 iron\_status) %>%  
 tbl\_summary(by = sex, #stratify by sex  
 statistic = list(all\_categorical() ~ "{n} ({p}%)"),   
 digits = all\_categorical() ~ c(0,1),  
 missing\_text = "Missing (n)"  
 ) %>%  
 add\_p(iron\_status ~ "fisher.test") %>%  
 modify\_header(label ~ "\*\*Variable\*\*") %>%  
 bold\_labels()

## Table printed with {flextable}, not {gt}. Learn why at  
## http://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html  
## To suppress this message, include `message = FALSE` in the code chunk header.

| Variable | Male, N = 4,5571 | Female, N = 4,6971 | p-value2 |
| --- | --- | --- | --- |
| **iron\_status** |  |  | <0.001 |
| Deficient | 427 (14.9%) | 949 (31.1%) |  |
| Excessive | 91 (3.2%) | 51 (1.7%) |  |
| Normal | 2,356 (82.0%) | 2,048 (67.2%) |  |
| Missing (n) | 1,683 | 1,649 |  |
| 1n (%) | | | |
| 2Fisher's exact test | | | |

# Calculate odds ratios from 2x2 table

# Simplify iron into two categories  
nhanes\_dataset\_iron <- nhanes\_dataset %>%  
 filter(!iron\_status == "Excessive") %>%  
 droplevels()  
table(nhanes\_dataset\_iron$iron\_status)

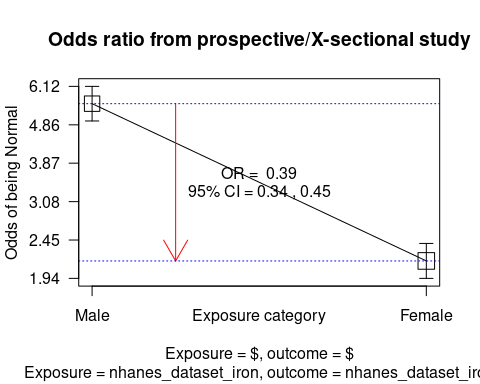
##   
## Deficient Normal   
## 1376 4404

# The relationship between iron status (normal and deficient) and sex  
  
# Make 2x2 table  
nhanes\_dataset\_iron %>%  
 dplyr::select(sex,  
 iron\_status) %>%  
 tbl\_summary(by = sex,  
 digits = all\_categorical() ~ c(0,1)  
 ) %>%  
 modify\_header(label ~ "\*\*Variable\*\*") %>%  
 bold\_labels()

## Table printed with {flextable}, not {gt}. Learn why at  
## http://www.danieldsjoberg.com/gtsummary/articles/rmarkdown.html  
## To suppress this message, include `message = FALSE` in the code chunk header.

| Variable | Male, N = 2,7831 | Female, N = 2,9971 |
| --- | --- | --- |
| **iron\_status** |  |  |
| Deficient | 427 (15.3%) | 949 (31.7%) |
| Normal | 2,356 (84.7%) | 2,048 (68.3%) |
| 1n (%) | | |

# Calculate odds ratio  
cc(outcome = nhanes\_dataset\_iron$iron\_status,  
 exposure = nhanes\_dataset\_iron$sex)



##   
## nhanes\_dataset\_iron$sex  
## nhanes\_dataset\_iron$iron\_status Male Female Total  
## Deficient 427 949 1376  
## Normal 2356 2048 4404  
## Total 2783 2997 5780  
##   
## OR = 0.39   
## 95% CI = 0.34, 0.44   
## Chi-squared = 211.93, 1 d.f., P value = 0  
## Fisher's exact test (2-sided) P value = 0

# Create fake data for graphing odds ratio results

## Assuming that you obtained ORs for all and by sex and race.  
## Let's create a data frame for these ORs and 95% CIs.  
  
x\_num <- c(1, 3, 4, 6, 7)  
x1 <- c("Overall", "Male", "Female", "White", "Black")  
or <- c(1.5, 1.1, 2.0, 1.4, 1.6)  
or\_lower\_lim <- c(1.2, 0.9, 1.65, 0.95, 1.3)  
or\_upper\_lim <- c(1.85, 1.35, 2.4, 2.05, 2.0)  
results <- data.frame(x\_num, x1, or, or\_lower\_lim, or\_upper\_lim)  
results$x1 <- factor(results$x1,  
 levels = results$x1)

# Plot odds ratios with a forest plot

# Forest plot, ggplot2 package  
ggplot(results,  
 aes(x = or,  
 y = x1,  
 xmin = or\_lower\_lim,  
 xmax = or\_upper\_lim)) +  
 geom\_pointrange() +  
 labs(x = "Odds Ratio", y = "Group") +  
 geom\_vline(xintercept = 1,  
 linetype = 2)

