DSCI353-353m-453: Class 10b Contigency Tables & T-tests

Profs: R. H. French, L. S. Bruckman, P. Leu, K. Davis, S. Cirlos

TAs: W. Oltjen, K. Hernandez, M. Li, M. Li, D. Colvin

03 November, 2022

Contents

10.2.2.1 Class Readings, Assignments, Syllabus Topics . . . 1
10.2.2.1.1 Reading, Lab Exercises, SemProjects . . . 1
10.2.2.1.2 Textbooks . . . 2
10.2.2.1.3 Syllabus . . . 2
10.2.2.2 CrossTab/Contingency Tables and T-tests . . . 2
10.2.2.2.1 Frequency and Contingency Tables . . . 2
10.2.2.2.2 Frequency tables . . . 4
10.2.2.2.3 Tests of Independence . . . 9
10.2.2.2.4 Measures of Association . . . 12
10.2.2.3 Correlations . . . 14
10.2.2.3.1 Pearson, Spearman, and Kendall Correlations . . . 14
10.2.2.3.2 Covariances and correlations . . . 15
10.2.2.4 t-tests . . . 20
10.2.2.4.1 Independent t-test . . . 20
10.2.2.4.2 Dependent t-test . . . 21
10.2.2.5 Nonparametric tests of group differences . . . 22
10.2.2.5.1 Comparing two groups . . . 22
10.2.2.5.2 Comparing more than two groups . . . 24
10.2.2.5.3 Nonparametric multiple comparisons . . . 24
10.2.2.6 Summary . . . 27
10.2.3 Links . . . 28

10.2.2.1 Class Readings, Assignments, Syllabus Topics

10.2.2.1.1 Reading, Lab Exercises, SemProjects

- Readings:
  - For today:
  - For next class: OIS 8
- Laboratory Exercises:
  - LE 5: is due Thursday November 10th
  - LE 6: Will be posted Thursday November 10th
- Office Hours: (Class Canvas Calendar for Zoom Link)
  - Wednesday @ 4:00 PM to 5:00 PM, Will Oltjen
  - Saturday @ 3:00 PM to 4:00 PM, Kristen Hernandez
  - **Office Hours are on Zoom, and recorded**
- Semester Projects
  - DSCI 451 Students Biweekly Update #5 Due Friday Nov. 4th
  - DSCI 451 Students

        * Next **Update #5 is Due Friday November 4th**
    - All DSCI 351/351M/451 Students:
        * **Peer Grading of Report Out #2 is Due November 8th**
    - Exams
        * Final: Monday December 19, 2022, 12:00PM - 3:00PM, Nord 356 or remote

10.2.2.1.2 Textbooks

- Introduction to R and Data Science
    - For R, Coding, Inferential Statistics
        * Peng: R Programming for Data Science
        * Peng: Exploratory Data Analysis with R

Textbooks for this class

- OIS = Diez, Barr, Çetinkaya-Runde: Open Intro Stat v4
- R4DS = Wickham, Grolemund: R for Data Science

Textbooks for DSCI353/353M/453, And in your Repo now

- ISLR = James, Witten, Hastie, Tibshirani: Intro to Statistical Learning with R
- ESL = Trevor Hastie, Tibshirani, Friedman: Elements of Statistical Learning
- DLwR = Chollet, Allaire: Deep Learning with R

10.2.2.1.3 Syllabus

10.2.2.2 CrossTab/Contingency Tables and T-tests

- What are Contingency Tables
    - a contingency table
        * (also known as a cross tabulation or crosstab)
    - is a type of table in a matrix format
        * that displays the (multivariate) frequency distribution of the variables.

They are heavily used in

- survey research,
- business intelligence,
- engineering, and scientific research.

They provide a basic picture of

- the interrelation between two variables
    - and can help find interactions between them.

10.2.2.2.1 Frequency and Contingency Tables

- Lets look at frequency and contingency tables
    - from categorical variables,
    - along with
        * tests of independence,
        * measures of association, and
        * methods for graphically displaying results.

We'll be using functions in base R,

- along with functions from
    - the `vcd` package

Day:Date	Foundation	Practicum	Reading	Due
w01a:Tu:8/30/22	ODS Tool Chain	R, Rstudio, Git		
w01b:Th:9/1/22	Setup ODS Tool Chain	Bash, Git, Slack, Agile	PRP4-33	LE1
w02a:Tu:9/6/22	Bash-Git-Knuth-Lit.Prog.	RIntroR	PRP35-64	
w02b:Th:9/8/22	What is Data Science	OIS:Intro2R	OIS1,2	
w02Pr:Fr:9/9/22			PRP65-93	**451 Update1**
w03a:Tu:9/13/22	Data Intro	Data Analytic Style	PRP94-116	LE2 **LE1 Due**
w03b:Th:9/15/22	Rand. Var. Normal Dist.	Git, Rmds, Loops	OIS4	
w04a:Tu:9/20/22	Tidy Check Explore	Tidy GapMinder	EDA1-31	
w04b:Th:9/22/22	Inference, DSCI Process	Other Distrib. 7 ways	R4DS1-3	LE3 **LE2 Due**
w04Pr:Fr:9/23/22			EDA32-58	**451 Update2**
w05a:Tu:9/27/22	OIS4 Rand. Var.	EDA of PET Degr.	OIS5	
w05b:Th:9/29/22	OIS5 Found. of Infer.	Multivar Corr. Plot	R4DS4-6	
w05Pr:Fr:9/30/22				**451 RepOut1**
w06a:Tu:10/4/22	Pred., Algorithm, Model		R4DS7-8	
w06b:Th:10/6/22	Summ. Stats & Vis.	Anscombe's Quartets	R4DS9-16	LE4 **LE3 Due**
w06Pr:Fr:10/7/22				**451 Update3**
w07a:Tu:10/11/22	Midterm Rev. Tidy Data	Correl Plots Summ Stats	OIS6.1-2	**PeerRv1 Due**
w07b:Th:10/13/22	HypoTest, Infer. Recap	Penguin EDA, Sampling		
w08a:Tu:10/18/22	**MIDTERM**	**EXAM**		
w08b:Th:10/20/22	Programming & Coding	Code Packaging		**LE4 Due**
w08Pr:Fr:10/21/22				**451 Update4**
Tu:10/24,25	**CWRU**	**FALL BREAK**	R4DS17-21	
w09b:Th:10/27/22	Cat. Inf. 1 & 2 propor.	Indep. Test,2-way tables	OIS6.3-4	LE5
w09Pr:Fr:10/28/22				**451 RepOut2**
w10a:Tu:11/1/22	Goodness of Fit, χ^2 test	t-tests 1&2 means	OIS7.1-4	
w10b:Th:11/3/22	Num. Infer, Cont. Tables	Stat. Power		
w10Pr:Fr:11/4/22				**451 Update5**
w11a:Tu:11/8/22	Sample & Effect Size	Stat. Power GGmap	OIS8	**PeerRv2 Due**
w11b:Th:11/10/22	Inf. 4 Regr, Test & Train	Curse of Dimen.	ISLR1,2.1,2	LE6 **LE5 Due**
w12a:Tu:11/15/22	Lin. Regr. Part 1	Residuals	OIS9	
w12b:Th:11/17/22	Lin. Regr. Part 2	Regr. Diagnostics		
w12Pr:Fr:11/18/22				**451 Update6**
w13a:Tu:11/22/22	Mult. Lin. Regr.	Var. & Mod. Selec.,	ISLR3.1	LE7 **LE6 due**
w13b:Th:11/24/22	Log. Regr.	GIS Trends	ISLR3.2	
w13Pr:Fr:11/25/22				**451 RepOut3**
w14a:Tu:11/23/22	Classificat., Sup. Lrning	Caret, Broom 4 modeling	ISLR4.1-3	
Th,Fr:11/24,25	**THANKSGIVING**	**Vacation**		
w15a:Tu:11/29/22		Clustering		**PeerRv3 Due**
w15b:Th:12/1/22	Big Data Analytics	Dist. Comp., Hadoop		
w15SPr:Fr:12/2/22		Read Article by	Mirletz,2015	
w16a:Tu:12/6/22	Final Exam Review			
w15b:Th:12/8/22				**LE7 due**
Friday 12/12	**SemProj**	**Final Report**		**SemProj4 due**
Monday 12/19	**FINAL EXAM**	**12:00-3:00pm**	Nord 356	or remote

Figure 1: DSCI351-351M-451 Syllabus

- and the `gmodels` package

In the following examples,

- assume that A, B, and C
- represent categorical variables.

10.2.2.2.2 Frequency tables

- The data for this section come from the Arthritis dataset
  - included with the `vcd` package.

The data are from Kock & Edward (1988)

- and represent a double-blind clinical trial
- of new treatments for rheumatoid arthritis.

```
library(vcd)
```

```
## Loading required package: grid
```

```
head(Arthritis)
```

```
##   ID Treatment  Sex Age Improved
## 1 57   Treated Male  27     Some
## 2 46   Treated Male  29     None
## 3 77   Treated Male  30     None
## 4 17   Treated Male  32   Marked
## 5 36   Treated Male  46   Marked
## 6 23   Treated Male  58   Marked
```

```
str(Arthritis)
```

```
## 'data.frame':    84 obs. of  5 variables:
##  $ ID       : int  57 46 77 17 36 23 75 39 33 55 ...
##  $ Treatment: Factor w/ 2 levels "Placebo","Treated": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Sex      : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Age      : int  27 29 30 32 46 58 59 59 63 63 ...
##  $ Improved : Ord.factor w/ 3 levels "None"<"Some"<..: 2 1 1 3 3 3 1 3 1 1 ...
```

Above are the first few observations:

- Treatment (Placebo, Treated),
- Sex (Male, Female), and
- Improved (None, Some, Marked)
  - are all categorical factors.

Next we'll create

- frequency tables and
- contingency tables (cross-classifications)
  - from the data.

We'll begin with simple frequencies,

- followed by two-way contingency tables,
- and end with multiway contingency tables.

One way contigency table

- The first step is to create a table
  - using either the `table()`

  - or `xtabs()` function
    * and then manipulate it using the other functions.

You can generate simple frequency counts using the `table()` function.

You can turn these frequencies into proportions with `prop.table()`

- or into percentages using `prop.table()*100`:

```
mytable <- table(Arthritis$Improved)
mytable                       # counts
```

```
##
##   None   Some Marked
##     42     14     28
```

```
prop.table(mytable)           # proportions
```

```
##
##      None      Some    Marked
## 0.5000000 0.1666667 0.3333333
```

```
prop.table(mytable) * 100       # percents
```

```
##
##     None     Some   Marked
## 50.00000 16.66667 33.33333
```

Two way contingency table

- For two-way tables,
  - the format for the `table()` function is
    * `mytable <- table(A, B)`
  - where A is the row variable and
  - B is the column variable.

Alternatively,

- the `xtabs()` function allows you to create
  - a contingency table
  - using formula-style input.

```
mytable <- xtabs(~ Treatment + Improved, data = Arthritis)
mytable  # counts
```

```
##           Improved
## Treatment None Some Marked
##   Placebo   29    7      7
##   Treated   13    7     21
```

```
margin.table(mytable, 1)    # total counts for Treatment
```

```
## Treatment
## Placebo Treated
##      43      41
```

```
prop.table(mytable, 1)      # row proportions (rows add to 1)
```

```
##           Improved
## Treatment      None      Some    Marked
##   Placebo 0.6744186 0.1627907 0.1627907
```

```
##   Treated 0.3170732 0.1707317 0.5121951
```

```
margin.table(mytable, 2)    # total counts for Improved
```

```
## Improved
##   None   Some Marked
##     42     14     28
```

```
prop.table(mytable, 2)      # column proportions (columns add to 1)
```

```
##           Improved
## Treatment      None      Some    Marked
##   Placebo 0.6904762 0.5000000 0.2500000
##   Treated 0.3095238 0.5000000 0.7500000
```

```
prop.table(mytable)         # cell proportions (all cells add to 1)
```

```
##           Improved
## Treatment       None       Some     Marked
##   Placebo 0.34523810 0.08333333 0.08333333
##   Treated 0.15476190 0.08333333 0.25000000
```

```
addmargins(mytable)         # cell counts with row and column sums
```

```
##           Improved
## Treatment None Some Marked Sum
##   Placebo   29    7      7  43
##   Treated   13    7     21  41
##   Sum       42   14     28  84
```

```
addmargins(prop.table(mytable)) # cell proportions with row and column proportions
```

```
##           Improved
## Treatment       None       Some     Marked        Sum
##   Placebo 0.34523810 0.08333333 0.08333333 0.51190476
##   Treated 0.15476190 0.08333333 0.25000000 0.48809524
##   Sum     0.50000000 0.16666667 0.33333333 1.00000000
```

```
addmargins(prop.table(mytable, 1), 2) # row proportions with row sums
```

```
##           Improved
## Treatment      None      Some    Marked       Sum
##   Placebo 0.6744186 0.1627907 0.1627907 1.0000000
##   Treated 0.3170732 0.1707317 0.5121951 1.0000000
```

```
addmargins(prop.table(mytable, 2), 1) # column proportions with column sums
```

```
##           Improved
## Treatment      None      Some    Marked
##   Placebo 0.6904762 0.5000000 0.2500000
##   Treated 0.3095238 0.5000000 0.7500000
##   Sum     1.0000000 1.0000000 1.0000000
```

Two-way table using `gmodels::CrossTable`

```
library(gmodels)
CrossTable(Arthritis$Treatment, Arthritis$Improved)
```

```
##
##
```

```
##    Cell Contents
## |-------------------------|
## |                       N |
## | Chi-square contribution |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  84
##
##
##                      | Arthritis$Improved
## Arthritis$Treatment |      None |      Some |    Marked | Row Total |
## --------------------|-----------|-----------|-----------|-----------|
##             Placebo |        29 |         7 |         7 |        43 |
##                     |     2.616 |     0.004 |     3.752 |           |
##                     |     0.674 |     0.163 |     0.163 |     0.512 |
##                     |     0.690 |     0.500 |     0.250 |           |
##                     |     0.345 |     0.083 |     0.083 |           |
## --------------------|-----------|-----------|-----------|-----------|
##             Treated |        13 |         7 |        21 |        41 |
##                     |     2.744 |     0.004 |     3.935 |           |
##                     |     0.317 |     0.171 |     0.512 |     0.488 |
##                     |     0.310 |     0.500 |     0.750 |           |
##                     |     0.155 |     0.083 |     0.250 |           |
## --------------------|-----------|-----------|-----------|-----------|
##        Column Total |        42 |        14 |        28 |        84 |
##                     |     0.500 |     0.167 |     0.333 |           |
## --------------------|-----------|-----------|-----------|-----------|
##
##
```

Three-way contingency table

- Both `table()` and `xtabs()`
  - can be used to generate multidimensional tables
    - based on three or more categorical variables.
  - The `margin.table()`, `prop.table()`, and `addmargins()` functions
    - extend naturally to more than two dimensions.

Additionally, the `ftable()` function

- can be used to print multidimensional tables
- in a compact and attractive manner.

Here is an example.

```
mytable <- xtabs( ~ Treatment + Sex + Improved, data = Arthritis) # 1. Cell Frequencies
mytable
```

```
## , , Improved = None
##
##          Sex
## Treatment Female Male
##   Placebo     19   10
```

```
##   Treated      6    7
##
## , , Improved = Some
##
##           Sex
## Treatment Female Male
##   Placebo      7    0
##   Treated      5    2
##
## , , Improved = Marked
##
##           Sex
## Treatment Female Male
##   Placebo      6    1
##   Treated     16    5
```

```
margin.table(mytable, 1)  # totals for Treatment, 2. Marginal Frequencies
```

```
## Treatment
## Placebo Treated
##      43      41
```

```
margin.table(mytable, 2)  # totals for Sex
```

```
## Sex
## Female   Male
##     59     25
```

```
margin.table(mytable, 3)  # totals for Improved
```

```
## Improved
##   None   Some Marked
##     42     14     28
```

```
margin.table(mytable, c(1, 3)) # totals for Treatment by Improved
```

```
##           Improved
## Treatment None Some Marked
##   Placebo   29    7      7
##   Treated   13    7     21
```

```
                               # 3.  Treatment × Improved marginal frequencies
```

```
 ftable(prop.table(mytable, c(1, 2))) # 4.  Improved proportions for Treatment × Sex
```

```
##                  Improved       None       Some     Marked
## Treatment Sex
## Placebo   Female          0.59375000 0.21875000 0.18750000
##           Male            0.90909091 0.00000000 0.09090909
## Treated   Female          0.22222222 0.18518519 0.59259259
##           Male            0.50000000 0.14285714 0.35714286
```

```
ftable(addmargins(prop.table(mytable, c(1,  2)), 3))
```

```
##                  Improved       None       Some     Marked        Sum
## Treatment Sex
## Placebo   Female          0.59375000 0.21875000 0.18750000 1.00000000
##           Male            0.90909091 0.00000000 0.09090909 1.00000000
## Treated   Female          0.22222222 0.18518519 0.59259259 1.00000000
```

```
##           Male            0.50000000 0.14285714 0.35714286 1.00000000
```

In the code above

- The code at #1 produces cell frequencies for the three-way classification.
  - The code also demonstrates how the `ftable()` function can be used
  - to print a more compact and attractive version of the table.
- The code at #2 produces the marginal frequencies
  - for Treatment, Sex, and Improved.
- Because you created the table with
  - the formula ~Treatment+Sex + Improved,
- Treatment is referred to by index 1,
  - Sex is referred to by index 2,
  - and Improved is referred to by index 3.
- The code at #3 produces the marginal frequencies
  - for the Treatment x Improved classification,
  - summed over Sex.
- The proportion of patients with None, Some, and Marked improvement
  - for each Treatment × Sex combination
  - is provided in #4.
- Here you see that 36% of treated males had marked improvement,
  - compared to 59% of treated females.
- In general, the proportions will add to 1
  - over the indices not included in the prop.table() call
  - (the third index, or Improved in this case).
- You can see this in the last example,
  - where you add a sum margin over the third index.

Contingency tables tell you

- the frequency or proportions of cases
  - for each combination of the variables that make up the table,
- but you're probably also interested in whether the variables in the table
  - are related or independent.

Tests of independence are covered next.

10.2.2.2.3 Tests of Independence

- R provides several methods of testing the independence
  - of categorical variables.

The three tests described in here are

- the χ^2 (chi-square) test of independence,
- the Fisher exact test,
- and the Cochran-Mantel–Haenszel test.

Treatment by Sex for each Level of Improved

- We'll use this as our example

```
ftable(mytable)
```

```
##                  Improved None Some Marked
## Treatment Sex
## Placebo   Female            19    7      6
##           Male              10    0      1
## Treated   Female             6    5     16
```

```
##           Male               7    2      5
```

```
ftable(prop.table(mytable, c(1, 2))) # proportions sum to 1 over index omitted
```

```
##                  Improved       None       Some     Marked
## Treatment Sex
## Placebo   Female          0.59375000 0.21875000 0.18750000
##           Male            0.90909091 0.00000000 0.09090909
## Treated   Female          0.22222222 0.18518519 0.59259259
##           Male            0.50000000 0.14285714 0.35714286
```

```
ftable(addmargins(prop.table(mytable, c(1, 2)), 3))
```

```
##                  Improved       None       Some     Marked        Sum
## Treatment Sex
## Placebo   Female          0.59375000 0.21875000 0.18750000 1.00000000
##           Male            0.90909091 0.00000000 0.09090909 1.00000000
## Treated   Female          0.22222222 0.18518519 0.59259259 1.00000000
##           Male            0.50000000 0.14285714 0.35714286 1.00000000
```

```
ftable(addmargins(prop.table(mytable, c(1, 2)), 3)) * 100
```

```
##                  Improved       None       Some     Marked        Sum
## Treatment Sex
## Placebo   Female           59.375000  21.875000  18.750000 100.000000
##           Male             90.909091   0.000000   9.090909 100.000000
## Treated   Female           22.222222  18.518519  59.259259 100.000000
##           Male             50.000000  14.285714  35.714286 100.000000
```

χ^2 (Chi-square) test of independence

- You can apply the function `chisq.test()`
  - to a two-way table in order to produce a chi-square test of independence
  - of the row and column variables.

```
library(vcd)
mytable <- xtabs(~ Treatment + Improved, data = Arthritis)
chisq.test(mytable) # 1
```

```
##
##  Pearson's Chi-squared test
##
## data:  mytable
## X-squared = 13.055, df = 2, p-value = 0.001463
```

```
mytable <- xtabs(~ Improved + Sex, data = Arthritis)
chisq.test(mytable) #2
```

```
## Warning in chisq.test(mytable): Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  mytable
## X-squared = 4.8407, df = 2, p-value = 0.08889
```

1. Treatment and Improved aren't independent.
2. Gender and Improved are independent.

From the results #1, there appears to be

- a relationship between treatment received and level of improvement
  - ($p < .01$).

But there doesn't appear to be a relationship #2

- between patient sex and improvement ($p > .05$).

The p-values are the probability of obtaining the sampled results,

- assuming independence of the row and column variables in the population.

Because the probability is small for #1,

- you reject the hypothesis
  - that treatment type and outcome are independent.

Because the probability for #2 isn't small,

- it's not unreasonable to assume that outcome and gender are independent.

The warning message is produced because

- one of the six cells in the table (male-some improvement)
  - has an expected value less than five,
  - which may invalidate the chi-square approximation.

Fisher's exact test

- You can produce a Fisher's exact test
  - via the `fisher.test()` function.

Fisher's exact test evaluates

- the null hypothesis of independence of rows and columns
  - in a contingency table with fixed marginals.

Fisher's exact test is a statistical significance test

- used in the analysis of contingency tables.
- Although in practice it is employed
  - when sample sizes are small,
  - it is valid for all sample sizes.
- It is named after its inventor, Ronald Fisher,
  - and is one of a class of exact tests,
- so called because the significance of the deviation from a null hypothesis
  - (e.g., P-value)
  - can be calculated exactly,
- rather than relying on an approximation
  - that becomes exact in the limit
  - as the sample size grows to infinity,
  - as with many statistical tests.

The format is `fisher.test(mytable)`,

- where `mytable` is a two-way contingency table.

```
mytable <- xtabs( ~ Treatment + Improved, data = Arthritis)
fisher.test(mytable)
```

```
##
##  Fisher's Exact Test for Count Data
##
```

```
## data:  mytable
## p-value = 0.001393
## alternative hypothesis: two.sided
```

In contrast to many statistical packages,

- the `fisher.test()` function can be applied
  - to any two-way table with two or more rows and columns,
  - not just a 2 × 2 table.

Chochran-Mantel-Haenszel test

- The `mantelhaen.test()` function provides
  - a Cochran–Mantel–Haenszel chi-square test of the null hypothesis
    - that two nominal variables are conditionally independent
    - in each stratum of a third variable.

The following code tests the hypothesis

- that the Treatment and Improved variables
  - are independent within each level for Sex.

The test assumes

- that there's no three-way (Treatment × Improved × Sex) interaction:

```
mytable <- xtabs( ~ Treatment + Improved + Sex, data = Arthritis)
mantelhaen.test(mytable)
```

```
##
##  Cochran-Mantel-Haenszel test
##
## data:  mytable
## Cochran-Mantel-Haenszel M^2 = 14.632, df = 2, p-value = 0.0006647
```

The results suggest that

- the treatment received
- and the improvement reported
  - aren't independent within each level of Sex
  - (that is, treated individuals improved more
  - than those receiving placebos when controlling for sex).

10.2.2.2.4 Measures of Association

- The significance tests in the previous section
  - evaluate whether sufficient evidence exists
    - to reject a null hypothesis of independence between variables.

If you can reject the null hypothesis,

- your interest turns naturally to measures of association
  - in order to gauge the strength of the relationships present.

The `assocstats()` function in the `vcd` package

- can be used to calculate
  - the phi coefficient,
  - contingency coefficient,
  - and Cramer's V for a two-way table.

Measures of association for a two-way table

- The significance tests in the previous section
  - evaluate whether sufficient evidence exists
    - to reject a null hypothesis of independence between variables.

If you can reject the null hypothesis,

- your interest turns naturally to measures of association
  - in order to gauge the strength of the relationships present.

The `assocstats()` function in the `vcd` package

- can be used to calculate
  - the ϕ coefficient,
  - contingency coefficient,
  - and Cramer's V
- for a two-way table.

Here's an example.

```
library(vcd)
mytable <- xtabs(~ Treatment + Improved, data = Arthritis)
assocstats(mytable)
```

```
##                     X^2 df  P(> X^2)
## Likelihood Ratio 13.530  2 0.0011536
## Pearson          13.055  2 0.0014626
##
## Phi-Coefficient   : NA
## Contingency Coeff.: 0.367
## Cramer's V        : 0.394
```

In general, larger magnitudes indicate stronger associations.

The `vcd` package also provides a `kappa()` function

- that can calculate Cohen's kappa and weighted kappa
  - for a confusion matrix
- (for example, the degree of agreement between two judges
  - classifying a set of objects into categories).

We'll see more of confusion matrices in the spring.

Visualizing these types of results

- To visually exploring the relationships among categorical variables
  - You typically use bar charts to visualize
    - frequencies in one dimension.

The `vcd` package has excellent functions for visualizing relationships

- among categorical variables in multidimensional datasets
- using mosaic and association plots.

Finally, correspondence-analysis functions in the `ca` package

- allow you to visually explore relationships
  - between rows and columns in contingency tables
- using various geometric representations.

10.2.2.3 Correlations

- Correlation coefficients are used to describe relationships
  - among quantitative variables.

The sign (plus or minus) indicates

- the direction of the relationship (positive or inverse),
- and the magnitude indicates the strength of the relationship
  - (ranging from 0 for no relationship to 1
  - for a perfectly predictable relationship).

Here we'll look at a variety of correlation coefficients,

- as well as tests of significance.

We'll use the state.x77 dataset available in the base R installation.

- It provides data on
  - the population,
  - income,
  - illiteracy rate,
  - life expectancy,
  - murder rate, and
  - high school graduation rate
- for the 50 US states in 1977.

There are also temperature and land-area measures,

- but we'll drop them to save space.

Use help(state.x77) to learn more about the dataset.

In addition to the base installation,

- we'll be using the `psych` and `ggm` packages.

R can produce a variety of correlation coefficients, including

- Pearson,
- Spearman,
- Kendall,
- partial,
- polychoric, and
- polyserial.

10.2.2.3.1 Pearson, Spearman, and Kendall Correlations

- The Pearson product-moment correlation
  - assesses the degree of linear relationship
    - between two quantitative variables.

Spearman's rank-order correlation coefficient

- assesses the degree of relationship
  - between two rank-ordered variables.

Kendall's tau is also

- a nonparametric measure of rank correlation.

The `cor()` function produces all three correlation coefficients,

- whereas the `cov()` function provides covariances.

There are many options,

- but a simplified format for producing correlations is
  - cor(x, use= , method=)

The options are described in table 7.2.

Option	Description
`X`	Matrix or data frame.
`Use`	Specifies the handling of missing data. The options are `all.obs` (assumes no missing data—missing data will produce an error), `everything` (any correlation involving a case with missing values will be set to `missing`), `complete.obs` (listwise deletion), and `pairwise.complete.obs` (pairwise deletion).
`Method`	Specifies the type of correlation. The options are `pearson`, `spearman`, and `kendall`.

Figure 2: cov options

The default options are

- use = "everything" and
- method = "pearson".

```
states <- state.x77[, 1:6]
cov(states)
```

10.2.2.3.2 Covariances and correlations

```
##               Population      Income   Illiteracy     Life Exp      Murder
## Population 19931683.7588 571229.7796  292.8679592 -407.8424612 5663.523714
## Income       571229.7796 377573.3061 -163.7020408  280.6631837 -521.894286
## Illiteracy      292.8680   -163.7020    0.3715306   -0.4815122    1.581776
## Life Exp       -407.8425    280.6632   -0.4815122    1.8020204   -3.869480
## Murder         5663.5237   -521.8943    1.5817755   -3.8694804   13.627465
## HS Grad       -3551.5096   3076.7690   -3.2354694    6.3126849  -14.549616
##                 HS Grad
## Population -3551.509551
## Income      3076.768980
## Illiteracy    -3.235469
## Life Exp       6.312685
## Murder       -14.549616
## HS Grad       65.237894
```

```
cor(states)
```

```
##             Population     Income Illiteracy    Life Exp     Murder     HS Grad
## Population  1.00000000  0.2082276  0.1076224 -0.06805195  0.3436428 -0.09848975
## Income      0.20822756  1.0000000 -0.4370752  0.34025534 -0.2300776  0.61993232
## Illiteracy  0.10762237 -0.4370752  1.0000000 -0.58847793  0.7029752 -0.65718861
## Life Exp   -0.06805195  0.3402553 -0.5884779  1.00000000 -0.7808458  0.58221620
## Murder      0.34364275 -0.2300776  0.7029752 -0.78084575  1.0000000 -0.48797102
## HS Grad    -0.09848975  0.6199323 -0.6571886  0.58221620 -0.4879710  1.00000000
```

```
cor(states, method = "spearman")
```

```
##            Population     Income Illiteracy   Life Exp     Murder    HS Grad
## Population  1.0000000  0.1246098  0.3130496 -0.1040171  0.3457401 -0.3833649
## Income      0.1246098  1.0000000 -0.3145948  0.3241050 -0.2174623  0.5104809
## Illiteracy  0.3130496 -0.3145948  1.0000000 -0.5553735  0.6723592 -0.6545396
## Life Exp   -0.1040171  0.3241050 -0.5553735  1.0000000 -0.7802406  0.5239410
## Murder      0.3457401 -0.2174623  0.6723592 -0.7802406  1.0000000 -0.4367330
## HS Grad    -0.3833649  0.5104809 -0.6545396  0.5239410 -0.4367330  1.0000000
```

The first call

- produces the variances and covariances.

The second provides

- Pearson product-moment correlation coefficients,

And the third produces

- Spearman rank-order correlation coefficients.

You can see, for example,

- that a strong positive correlation exists
  - between income and high school graduation rate
- and that a strong negative correlation exists
  - between illiteracy rates and life expectancy.

Notice that you get square matrices by default

- (all variables crossed with all other variables).

You can also produce nonsquare matrices,

- as shown in the following example:

```
x <- states[, c("Population", "Income", "Illiteracy", "HS Grad")]
y <- states[, c("Life Exp", "Murder")]
cor(x, y)
```

```
##               Life Exp     Murder
## Population -0.06805195  0.3436428
## Income      0.34025534 -0.2300776
## Illiteracy -0.58847793  0.7029752
## HS Grad     0.58221620 -0.4879710
```

```
x <- states[, c("Population", "Income", "Illiteracy", "HS Grad")]
y <- states[, c("Life Exp", "Murder")]
cor(x, y)
```

```
##               Life Exp     Murder
## Population -0.06805195  0.3436428
## Income      0.34025534 -0.2300776
## Illiteracy -0.58847793  0.7029752
## HS Grad     0.58221620 -0.4879710
```

This version of the function is particularly useful

- when you're interested in the relationships
  - between one set of variables and another.

Notice that the results don't tell you if the correlations

- differ significantly from 0
  - that is, whether there's sufficient evidence based on the sample data
  - to conclude that the population correlations differ from 0.

Partial correlations

- A partial correlation is
  - a correlation between two quantitative variables,
    - controlling for one or more other quantitative variables.

You can use the `pcor()` function in the `ggm` package

- to provide partial correlation coefficients.
- The format is `pcor(u, S)`
  - where u is a vector of numbers,
    - with the first two numbers
    - being the indices of the variables to be correlated,
  - and the remaining numbers being the indices of the conditioning variables
    - (that is, the variables being partialed out).
  - S is the covariance matrix among the variables.

An example.

```
library(ggm)
colnames(states)
```

```
## [1] "Population" "Income"     "Illiteracy" "Life Exp"   "Murder"
## [6] "HS Grad"
```

```
pcor(c(1, 5, 2, 3, 6), cov(states))
```

```
## [1] 0.3462724
```

In this case, 0.346 is the correlation between

- population (variable 1) and
  - murder rate (variable 5),
- controlling for the influence of
  - income, illiteracy rate, and high school graduation rate
  - (variables 2, 3, and 6 respectively).

The use of partial correlations is common in the social sciences.

Other types of Correlations

The `hetcor()` function in the `polycor` package

- can compute a heterogeneous correlation matrix
  - containing Pearson product-moment correlations

  - between numeric variables,
- polyserial correlations between numeric and ordinal variables,
- polychoric correlations between ordinal variables,
- and tetrachoric correlations between two dichotomous variables.

Polyserial, polychoric, and tetrachoric correlations

- assume that the ordinal or dichotomous variables
- are derived from underlying normal distributions.

Testing a correlation coefficient for significance

- Once you've generated correlation coefficients,
  - how do you test them for statistical significance?

The typical null hypothesis is no relationship

- (that is, the correlation in the population is 0).

You can use the `cor.test()` function

- to test an individual Pearson, Spearman, & Kendall correlation coefficient.

A simplified format

- is `cor.test(x, y, alternative = , method = )`
  - where x and y are the variables to be correlated,
- alternative specifies a two-tailed or one-tailed test
  - ("two.side", "less", or "greater"),
- and method specifies the type of correlation
  - ("pearson", "kendall", or "spearman") to compute.

Use alternative ="less" when the research hypothesis

- is that the population correlation is less than 0.

Use alternative="greater" when the research hypothesis

- is that the population correlation is greater than 0.

By default, alternative="two.side"

- (population correlation isn't equal to 0)
- is assumed.

```
cor.test(states[, 3], states[, 5])
```

```
##
##  Pearson's product-moment correlation
##
## data:  states[, 3] and states[, 5]
## t = 6.8479, df = 48, p-value = 1.258e-08
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.5279280 0.8207295
## sample estimates:
##       cor
## 0.7029752
```

This code tests the null hypothesis

- that the Pearson correlation between life expectancy and murder rate is 0.
- Assuming that the population correlation is 0,

  - you'd expect to see a sample correlation as large as 0.703
  - less than 1 time out of 10 million (that is, p = 1.258e-08).
- Given how unlikely this is,
  - you reject the null hypothesis in favor of the research hypothesis,
  - that the population correlation
  - between life expectancy and murder rate is not 0.

Correlation matrix and tests of significance via `corr.test()`

- Unfortunately, you can
  - test only one correlation at a time
  - using `cor.test()`.

Luckily, the `corr.test()` function provided in the `psych` package

- allows you to go further.

The `corr.test()` function produces

- correlations and significance levels
- for matrices of Pearson, Spearman, and Kendall correlations.

An example

```
library(psych)
corr.test(states, use = "complete")
```

```
## Call:corr.test(x = states, use = "complete")
## Correlation matrix
##            Population Income Illiteracy Life Exp Murder HS Grad
## Population       1.00   0.21       0.11    -0.07   0.34   -0.10
## Income           0.21   1.00      -0.44     0.34  -0.23    0.62
## Illiteracy       0.11  -0.44       1.00    -0.59   0.70   -0.66
## Life Exp        -0.07   0.34      -0.59     1.00  -0.78    0.58
## Murder           0.34  -0.23       0.70    -0.78   1.00   -0.49
## HS Grad         -0.10   0.62      -0.66     0.58  -0.49    1.00
## Sample Size
## [1] 50
## Probability values (Entries above the diagonal are adjusted for multiple tests.)
##            Population Income Illiteracy Life Exp Murder HS Grad
## Population       0.00   0.59       1.00      1.0   0.10       1
## Income           0.15   0.00       0.01      0.1   0.54       0
## Illiteracy       0.46   0.00       0.00      0.0   0.00       0
## Life Exp         0.64   0.02       0.00      0.0   0.00       0
## Murder           0.01   0.11       0.00      0.0   0.00       0
## HS Grad          0.50   0.00       0.00      0.0   0.00       0
##
##  To see confidence intervals of the correlations, print with the short=FALSE option
```

Before leaving this topic, it should be mentioned that

- the `r.test()` function in the `psych` package
- also provides a number of useful significance tests.

The function can be used to test the following:

- The significance of a correlation coefficient
- The difference between two independent correlations
- The difference between two dependent correlations sharing a single variable
- The difference between two dependent correlations

  - based on completely different variables

10.2.2.4 t-tests

- The most common activity in research is the comparison of two groups.
  - Do patients receiving a new drug show greater improvement
    * than patients using an existing medication?
  - Does one manufacturing process produce fewer defects than another?
  - Which of two teaching methods is most cost-effective?

If your outcome variable is categorical,

- you can use the methods just described.

Here, we'll focus on group comparisons,

- where the outcome variable is continuous
- and assumed to be distributed -normally.

For this illustration, we'll use the UScrime dataset in the `MASS` package.

- It contains information about the effect of punishment regimes
  - on crime rates in 47 US states in 1960.
- The outcome variables of interest will be
  - Prob (the probability of imprisonment),
  - U1 (the unemployment rate for urban males ages 14–24),
  - and U2 (the unemployment rate for urban males ages 35–39).

The categorical variable So

- (an indicator variable for Southern states)
- will serve as the grouping variable.

The data have been rescaled by the original authors.

10.2.2.4.1 Independent t-test

- Are you more likely to be imprisoned
  - if you commit a crime in the South?

The comparison of interest is Southern versus non-Southern states,

- and the dependent variable is the probability of incarceration.

A two-group independent t-test can be used

- to test the hypothesis that the two population means are equal.

Here, you assume

- that the two groups are independent
- and that the data is sampled from normal populations.

The following code

- compares Southern (group 1)
  - and non-Southern (group 0) states
- on the probability of imprisonment
- using a two-tailed test without the assumption of equal variances:

```
library(MASS)
t.test(Prob ~ So, data = UScrime)
```

```
##
##  Welch Two Sample t-test
##
## data:  Prob by So
## t = -3.8954, df = 24.925, p-value = 0.0006506
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -0.03852569 -0.01187439
## sample estimates:
## mean in group 0 mean in group 1
##      0.03851265      0.06371269
```

You can reject the hypothesis

- that Southern states and non-Southern states
  - have equal probabilities of imprisonment
- ($p < .001$).

10.2.2.4.2 Dependent t-test

- As a second example, you might ask
  - if the unemployment rate for younger males (14–24)
    - is greater than for older males (35–39).

In this case, the two groups aren't independent.

You wouldn't expect the unemployment rate

- for younger and older males in Alabama to be unrelated.

When observations in the two groups are related,

- you have a dependent-groups design.

Pre-post or repeated-measures designs

- also produce dependent groups.

A dependent t-test assumes that

- the difference between groups is normally distributed.

```
sapply(UScrime[c("U1", "U2")], function(x)
  (c(mean = mean(x), sd = sd(x))))
```

```
##            U1       U2
## mean 95.46809 33.97872
## sd   18.02878  8.44545
```

```
with(UScrime, t.test(U1, U2, paired = TRUE))
```

```
##
##  Paired t-test
##
## data:  U1 and U2
## t = 32.407, df = 46, p-value < 2.2e-16
## alternative hypothesis: true mean difference is not equal to 0
## 95 percent confidence interval:
##  57.67003 65.30870
## sample estimates:
## mean difference
```

```
##         61.48936
```

The mean difference (61.5) is large enough

- to warrant rejection of the hypothesis
  - that the mean unemployment rate for older and younger males is the same.
  - Younger males have a unemployment higher rate.
- In fact, the probability of obtaining a sample difference this large
  - if the population means are equal
  - is less than 0.00000000000000022 (that is, 2.2e–16).

When there are more than two groups

- What do you do if you want to compare more than two groups?

If you can assume that the data are

- independently sampled from normal populations,
- you can use analysis of variance (ANOVA).

ANOVA is a comprehensive methodology

- that covers many experimental and quasi-experimental designs.
- As such it almost a topic of its own.

10.2.2.5 Nonparametric tests of group differences

- If you're unable to meet
  - the parametric assumptions of a t-test or ANOVA,
  - you can turn to nonparametric approaches.

For example, if the outcome variables

- are severely skewed or ordinal in nature,
- you may wish to use the techniques in this section.

10.2.2.5.1 Comparing two groups

- If the two groups are independent,
  - you can use the Wilcoxon rank sum test
    - (more popularly known as the Mann–Whitney U test)
  - to assess whether the observations are sampled
    - from the same probability distribution
    - (that is, whether the probability of obtaining higher scores
    - is greater in one population than the other).

Mann-Whitney U-test

- If you apply the Mann–Whitney U test
  - to the question of incarceration rates from the previous section,
  - you'll get these results:

```
with(UScrime, by(Prob, So, median))
```

```
## So: 0
## [1] 0.038201
## ------------------------------------------------------------
## So: 1
## [1] 0.055552
```

```
wilcox.test(Prob ~ So, data = UScrime)
```

```
##
##  Wilcoxon rank sum exact test
##
## data:  Prob by So
## W = 81, p-value = 8.488e-05
## alternative hypothesis: true location shift is not equal to 0
```

```
sapply(UScrime[c("U1", "U2")], median)
```

```
## U1 U2
## 92 34
```

```
with(UScrime, wilcox.test(U1, U2, paired = TRUE))
```

```
## Warning in wilcox.test.default(U1, U2, paired = TRUE): cannot compute exact p-
## value with ties
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  U1 and U2
## V = 1128, p-value = 2.464e-09
## alternative hypothesis: true location shift is not equal to 0
```

Again, you can reject the hypothesis

- that incarceration rates are the same in Southern and non-Southern states
  - ($p < .001$).

The Wilcoxon signed rank test provides

- a nonparametric alternative to the dependent sample t-test.

It's appropriate in situations where

- the groups are paired
- and the assumption of normality is unwarranted.

Let's apply it to the unemployment question from the previous section.

```
sapply(UScrime[c("U1", "U2")], median)
```

```
## U1 U2
## 92 34
```

```
with(UScrime, wilcox.test(U1, U2, paired = TRUE))
```

```
## Warning in wilcox.test.default(U1, U2, paired = TRUE): cannot compute exact p-
## value with ties
```

```
##
##  Wilcoxon signed rank test with continuity correction
##
## data:  U1 and U2
## V = 1128, p-value = 2.464e-09
## alternative hypothesis: true location shift is not equal to 0
```

Again, you reach the same conclusion reached with the paired t-test.

10.2.2.5.2 Comparing more than two groups

- When there are more than two groups to be compared,
  - you must turn to other methods.

Consider the state.x77 dataset we used earlier.

- It contains population, income, illiteracy rate, life expectancy,
- murder rate, and high school graduation rate data for US states.

What if you want to compare the illiteracy rates

- in four regions of the country
  - (Northeast, South, North Central, and West)?

This is called a one-way design,

- and there are both parametric and nonparametric approaches available
  - to address the question.

If you can't meet the assumptions of ANOVA designs,

- you can use nonparametric methods to evaluate group differences.

If the groups are independent, a Kruskal–Wallis test

- provides a useful approach.

If the groups are dependent

- (for example, repeated measures or randomized block design),
- the Friedman test is more appropriate.

Kruskal-Wallis test

- Let's apply the Kruskal–Wallis test to the illiteracy question.

First, you'll have to add the region designations to the dataset.

- These are contained in the dataset state.region
- distributed with the base installation of R:

```
states <- data.frame(state.region, state.x77)
kruskal.test(Illiteracy ~ state.region, data = states)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  Illiteracy by state.region
## Kruskal-Wallis chi-squared = 22.672, df = 3, p-value = 4.726e-05
```

The significance test suggests

- that the illiteracy rate isn't the same
  - in each of the four regions of the country ($p < .001$).

10.2.2.5.3 Nonparametric multiple comparisons

- Although you can reject the null hypothesis of no difference,
  - the test doesn't tell you which regions
    - differ significantly from each other.

To answer this question, you could

- compare groups two at a time using the Wilcoxon test.

A more elegant approach is to apply a multiple-comparisons procedure

- that computes all pairwise comparisons,
- while controlling the type I error rate
  - (the probability of finding a difference that isn't there).

Lets define a function `wmc()` that can be used for this purpose.

- It compares groups two at a time
  - using the Wilcoxon test
- and adjusts the probability values using the `p.adj()` function.

The function's format is `wmc(y ~ A, data, method)`,

- where y is a numeric outcome variable,
- A is a grouping variable,
- data is the data frame containing these variables,
- and method is the approach used to limit Type I errors.

It uses an adjustment method developed by Holm (1979).

- It provides strong control of the family-wise error rate
  - (the probability of making one or more Type I errors
  - in a set of comparisons).
- See help(p.adjust) for a description of the other methods available.

The `wmc()` function

- first provides the
  - sample sizes,
  - medians,
  - and median absolute deviations
  - for each group #2.

```
wmc <- function(formula,
                data,
                exact = FALSE,
                sort = TRUE,
                method = "holm") {
  # setup
  df <- model.frame(formula, data)
  y <- df[[1]]
  x <- as.factor(df[[2]])


  # reorder levels of x by median y
  if (sort) {
    medians <- aggregate(y, by = list(x), FUN = median)[2]
    index <- order(medians)
    x <- factor(x, levels(x)[index])
  }

  groups <- levels(x)
  k <- length(groups)

  # summary statistics
  stats <-
```

```
  function(z)
    (c(
      N = length(z),
      Median = median(z),
      MAD = mad(z)
    ))
sumstats <- t(aggregate(y, by = list(x), FUN = stats)[2])
rownames(sumstats) <- c("n", "median", "mad")
colnames(sumstats) <- groups
cat("Descriptive Statistics\n\n")
print(sumstats)

# multiple comparisons
mc <- data.frame(
  Group.1 = character(0),
  Group.2 = character(0),
  W = numeric(0),
  p.unadj = numeric(0),
  p = numeric(0),
  stars = character(0),
  stringsAsFactors = FALSE
)

# perform Wilcoxon test
row <- 0
for (i in 1:k) {
  for (j in 1:k) {
    if (j > i) {
      row <- row + 1
      y1 <- y[x == groups[i]]
      y2 <- y[x == groups[j]]
      test <- wilcox.test(y1, y2, exact = exact)
      mc[row, 1] <- groups[i]
      mc[row, 2] <- groups[j]
      mc[row, 3] <- test$statistic
      mc[row, 4] <- test$p.value
    }
  }
}
mc$p <- p.adjust(mc$p.unadj, method = method)

# add stars
mc$stars <- " "
mc$stars[mc$p <   .1] <- "."
mc$stars[mc$p <  .05] <- "*"
mc$stars[mc$p <  .01] <- "**"
mc$stars[mc$p < .001] <- "***"
names(mc)[6] <- " "

cat("\nMultiple Comparisons (Wilcoxon Rank Sum Tests)\n")
cat(paste("Probability Adjustment = ", method, "\n\n", sep = ""))
print(mc[-4], right = TRUE)
cat("---\nSignif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1\n")
```

```
  return(invisible(NULL))
}
```

```
states <- data.frame(state.region, state.x77)
wmc(Illiteracy ~ state.region, data = states, method = "holm")
```

```
## Warning in xtfrm.data.frame(x): cannot xtfrm data frames
```

```
## Descriptive Statistics
##
##            West North Central Northeast    South
## n      13.00000      12.00000   9.00000 16.00000
## median  0.60000       0.70000   1.10000  1.75000
## mad     0.14826       0.14826   0.29652  0.59304
##
## Multiple Comparisons (Wilcoxon Rank Sum Tests)
## Probability Adjustment = holm
##
##         Group.1       Group.2    W            p
## 1          West North Central 88.0 8.665618e-01
## 2          West     Northeast 46.5 8.665618e-01
## 3          West         South 39.0 1.788186e-02   *
## 4 North Central     Northeast 20.5 5.359707e-02   .
## 5 North Central         South  2.0 8.051509e-05 ***
## 6     Northeast         South 18.0 1.187644e-02   *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The West has the lowest illiteracy rate,

- and the South has the highest.

The function then generates six statistical comparisons

- West versus North Central,
- West versus Northeast,
- West versus South,
- North Central versus Northeast,
- North Central versus South,
- and Northeast versus South) #3.

You can see from the two-sided p-values (p) that

- the South differs significantly from the other three regions
- and that the other three regions don't differ from each other
- at a $p < .05$ level.

10.2.2.6 Summary

- Inferential Statistics and statistical Tests
  - Descriptive statistics are used
    - to describe the distribution a quantitative variable numerically.
    - Many packages in R provide descriptive statistics for data frames.
    - The choice among packages is primarily a matter of personal preference.
  - Frequency tables and cross tabulations are used
    - to summarize the distributions of categorical variables.
  - The t-tests and the Mann-Whitney U test can be used

    - to compare two groups on a quantitative outcome.
  - A chi-square test can be used
    - to evaluate the association between two categorical variables.
  - The correlation coefficient is used
    - to evaluate the association between two quantitative variables.
  - Numeric summaries and statistical tests
    - should usually be accompanied by data visualizations.
    - Otherwise, important features of the data may be missed.

10.2.3 Links

- R. I. Kabacoff, R in Action, 3rd Edition, Manning Publications