Session 4: How can I perform bivariate analysis?

A Slower Introduction to R

Yea-Hung Chen, PhD, MS
UCSF Library

Thursday, October 30, 2025

Continuous-versus-continuous analysis

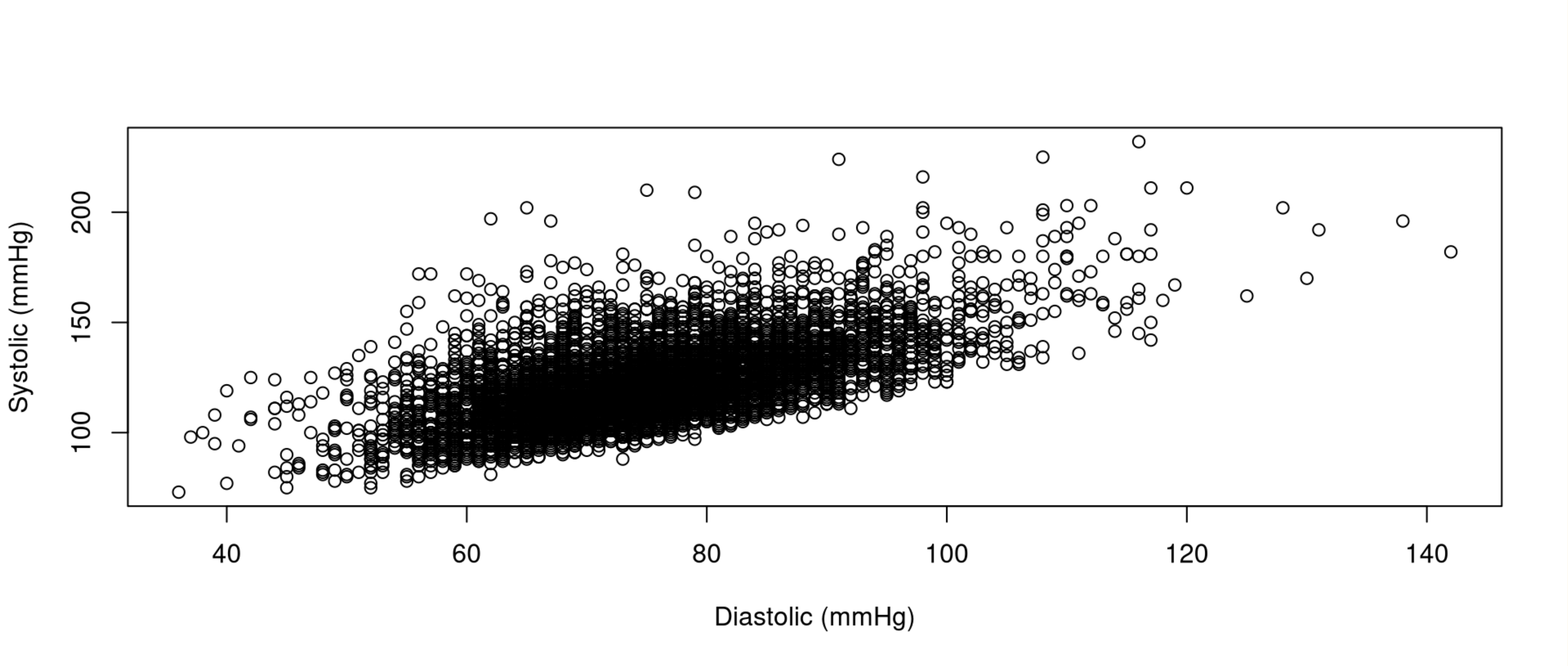
Systolic blood pressure versus diastolic blood pressure

- in this section, we'll examine the association between systolic blood pressure and diastolic blood pressure

```
class(nhanes$bpxosy1)
#|  [1] "integer"
head(nhanes$bpxosy1)
#|  [1] 135 121 111  NA  NA 110
class(nhanes$bpxodi1)
#|  [1] "integer"
head(nhanes$bpxodi1)
#|  [1] 98 84 79 NA NA 72
```

Scatter plots

```
plot(x=nhanes$bpxodi1,y=nhanes$bpxosy1,xlab='Diastolic (mmHg)',ylab='Systolic (mmHg)')
```



Formula notation

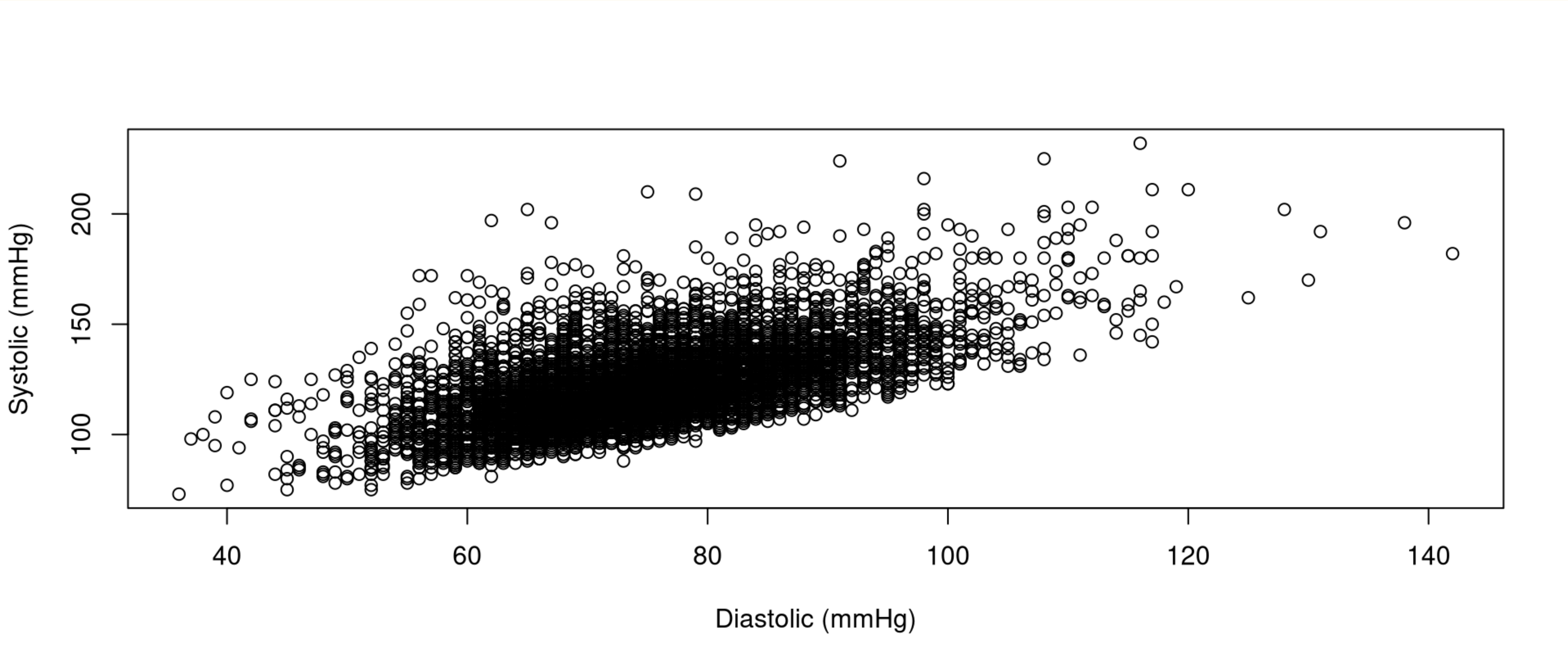
- we can equivalently use a `y~x` formula notation, like this:

```
plot(nhanes$bpxosy1~nhanes$bpxodi1,xlab='Diastolic (mmHg)',ylab='Systolic (mmHg)')
```

- formula notation is used in several R functions, including:
  - `lm()` for linear regression
  - `t.test()` for *t*-tests

Formula notation

```
plot(nhanes$bpxosy1~nhanes$bpxodi1,xlab='Diastolic (mmHg)',ylab='Systolic (mmHg)')
```



Pearson's correlation coefficient

```
cor(nhanes$bpxodi1,nhanes$bpxosy1)
#|  [1] NA
cor(nhanes$bpxodi1,nhanes$bpxosy1,use='complete.obs')
#|  [1] 0.6057034
```

- the `use=complete.obs` argument is similar to `na.rm=TRUE` in `mean()`, `sd()`, and other functions you saw last week
  - if either value in a pair is `NA`, then the pair is not used
- the order of the variables does not matter

Other measures of correlation

```
cor(nhanes$bpxodi1,nhanes$bpxosy1,use='complete.obs',method='kendall')
#|  [1] 0.4277319
cor(nhanes$bpxodi1,nhanes$bpxosy1,use='complete.obs',method='spearman')
#|  [1] 0.5832935
```

Simple linear regression

```
lm(nhanes$bpxosy1~nhanes$bpxodi1)
#|
#|  Call:
#|  lm(formula = nhanes$bpxosy1 ~ nhanes$bpxodi1)
#|
#|  Coefficients:
#|     (Intercept)  nhanes$bpxodi1
#|          49.165           0.979
```

Simple linear regression: `confint()`

```
my_model<-lm(nhanes$bpxosy1~nhanes$bpxodi1)
confint(my_model)
#|                     2.5 %    97.5 %
#|  (Intercept)    46.7210856 51.608631
#|  nhanes$bpxodi1  0.9467311  1.011178
```

Simple linear regression: `summary()`

```
summary(my_model)
#|
#|  Call:
#|  lm(formula = nhanes$bpxosy1 ~ nhanes$bpxodi1)
#|
#|  Residuals:
#|      Min       1Q   Median       3Q      Max
#|  -32.629 -10.323  -2.271   7.361  89.203
#|
#|  Coefficients:
#|                 Estimate Std. Error t value Pr(>|t|)
#|  (Intercept)    49.16486    1.24660   39.44   <2e-16 ***
#|  nhanes$bpxodi1  0.97895    0.01644   59.56   <2e-16 ***
#|  ---
#|  Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
#|
#|  Residual standard error: 14.68 on 6121 degrees of freedom
#|    (2030 observations deleted due to missingness)
```

Simple linear regression: `summary()`

- as with the `t.test()` function, it is possible to extract specific values, using dollar-sign notation

Simple linear regression: `summary()`

```
names(summary(my_model))
#|   [1] "call"          "terms"         "residuals"     "coefficients"
#|   [5] "aliased"       "sigma"         "df"            "r.squared"
#|   [9] "adj.r.squared" "fstatistic"    "cov.unscaled"  "na.action"
summary(my_model)$coefficients
#|                   Estimate Std. Error  t value      Pr(>|t|)
#|  (Intercept)    49.1648582 1.24659908 39.43919 2.462317e-303
#|  nhanes$bpxodi1  0.9789544 0.01643749 59.55620  0.000000e+00
summary(my_model)$r.squared
#|  [1] 0.3668766
```

Simple linear regression: the `data` argument

```
lm(bpxosy1~bpxodi1,data=nhanes)
#|
#|  Call:
#|  lm(formula = bpxosy1 ~ bpxodi1, data = nhanes)
#|
#|  Coefficients:
#|  (Intercept)      bpxodi1
#|       49.165        0.979
```

- several R functions support the `data` argument
- it is one way to omit dollar-sign notation

Exercise 1

Exercise 1

1. Create a script for today's exercises. Save the script. Include code at the top for importing the NHANES data.
2. Create a scatter plot for the relationship between the first two measurements of pulse (`bpxopls1` and `bpxopls2`), placing the second measurement on the *y* axis. Label both axes.

Exercise 1

3. Create a scatter plot for the relationship between diastolic blood pressure (`bpxodi1`) and pulse (`bpxopls1`), placing diastolic blood pressure on the *y* axis. Label both axes.
4. Fit a simple linear model for the relationship between diastolic blood pressure (`bpxodi1`) and pulse (`bpxopls1`), with diastolic blood pressure as the *y* variable. Use `summary()` to examine the model.

Continuous-versus-dichotomous analysis

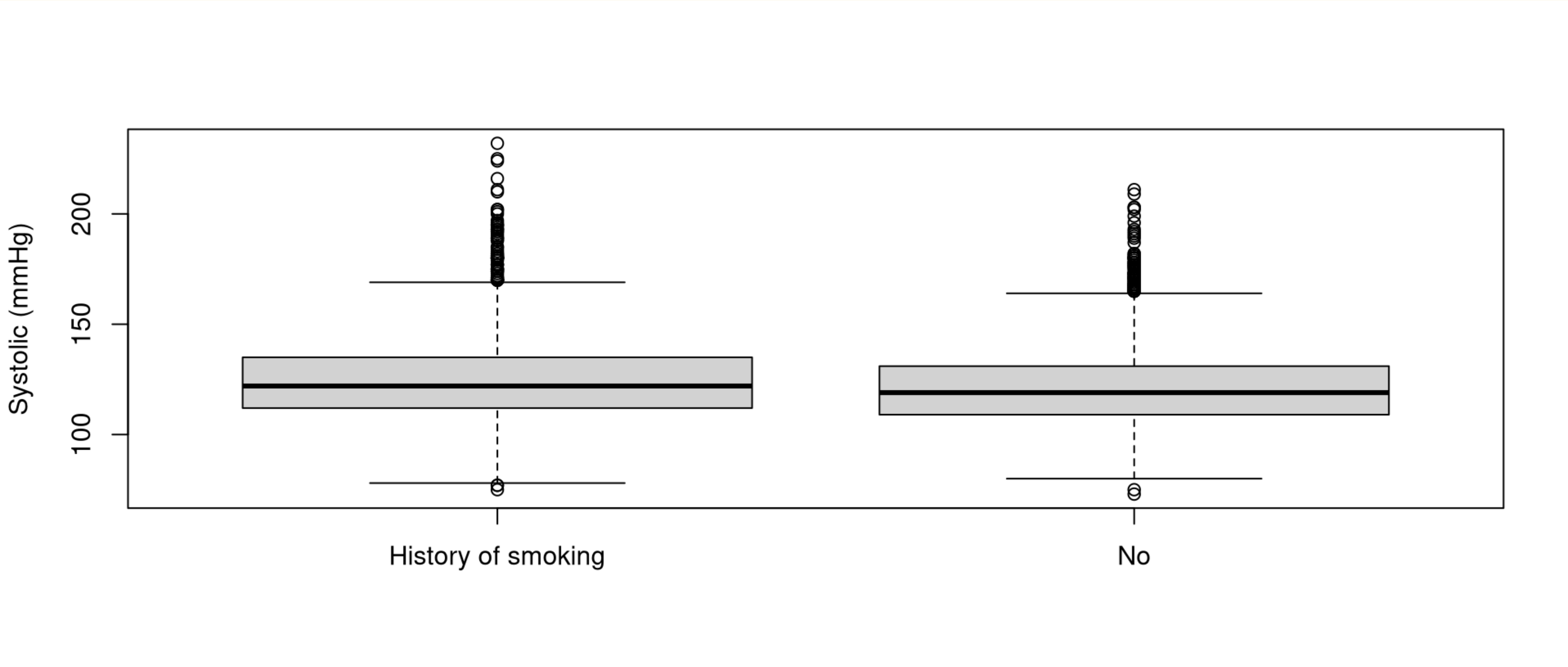
Systolic blood pressure versus smoking

- in this section, we'll examine the association between systolic blood pressure and smoking

```
class(nhanes$bpxosy1)
#|  [1] "integer"
head(nhanes$bpxosy1)
#|  [1] 135 121 111  NA  NA 110
class(nhanes$smoking)
#|  [1] "character"
head(nhanes$smoking)
#|  [1] "History of smoking" "History of smoking" "No"
#|  [4] "No"                 "No"                 "History of smoking"
```

Box plots

```
boxplot(nhanes$bpxosy1~nhanes$smoking,xlab='',ylab='Systolic (mmHg)')
```



t-tests

```
t.test(nhanes$bpxosy1~nhanes$smoking)
#|
#|      Welch Two Sample t-test
#|
#|  data:  nhanes$bpxosy1 by nhanes$smoking
#|  t = 7.4317, df = 4887.9, p-value = 1.258e-13
#|  alternative hypothesis: true difference in means between group History of smoking
#|  95 percent confidence interval:
#|   2.672803 4.588225
#|  sample estimates:
#|  mean in group History of smoking                 mean in group No
#|                          124.7512                         121.1207
```

t-tests

```
my_test<-t.test(nhanes$bpxosy1~nhanes$smoking)
names(my_test)
#|   [1] "statistic"   "parameter"   "p.value"     "conf.int"    "estimate"
#|   [6] "null.value"  "stderr"      "alternative" "method"      "data.name"
my_test$estimate
#|  mean in group History of smoking                mean in group No
#|                          124.7512                        121.1207
my_test$conf.int
#|  [1] 2.672803 4.588225
#|  attr(,"conf.level")
#|  [1] 0.95
```

Exercise 2

Exercise 2

1. Create a box plot for the relationship between pulse (`bpxopls1`) and asthma (`asthma`), adding a label for the *y* axis, and removing the label for the *x* axis.
2. Obtain a 95% confidence interval for the difference in mean pulse between the two asthma groups.

Exercise 2

3. Is the confidence interval for a yes-minus-no difference or for a no-minus-yes difference?

4. Obtain a 90% confidence interval for the difference in mean pulse between the two asthma groups, by adding the `conf.level=0.90` argument to `t.test()`.

Exercise 2

5. The `wilcox.test()` function implements the Wilcoxon rank-sum test (also known as the Mann-Whitney *U* test), which tests whether the "location" of a continuous variable's distribution differs between two groups. Use the function, and formula notation, to compare the distribution of pulse between the two asthma groups.

Categorical-versus-categorical analysis

Smoking versus asthma

- in this section, we'll examine the association between smoking and asthma

```
class(nhanes$smoking)
#|  [1] "character"
head(nhanes$smoking)
#|  [1] "History of smoking" "History of smoking" "No"
#|  [4] "No"                 "No"                 "History of smoking"
class(nhanes$asthma)
#|  [1] "character"
head(nhanes$asthma)
#|  [1] "No"                "No"                "No"
#|  [4] "No"                "History of asthma" "No"
```

Contingency tables

```
table(nhanes$smoking,nhanes$asthma)
#|
#|                          History of asthma   No
#|     History of smoking                 636 2603
#|     No                                 844 4029
```

- the first variable will be placed along the rows

Contingency tables

```
table(nhanes$smoking,nhanes$asthma,useNA='always')
#|
#|                          History of asthma   No <NA>
#|      History of smoking                 636 2603    4
#|      No                                 844 4029    5
#|      <NA>                                 6   25    1
```

- we can add the `useNA` argument to show missing values

Row-wise percents

```
prop.table(
  table(nhanes$smoking,nhanes$asthma),
  margin=1
)*100
```

- notice the two arguments inside `prop.table()`:
    - the `table()`
    - `margin=1`, which indicates we want row-wise percents

Row-wise percents

- equivalently, written all in one line:

```
prop.table(table(nhanes$smoking,nhanes$asthma),margin=1)*100
#|
#|                         History of asthma        No
#|     History of smoking           19.63569 80.36431
#|     No                           17.31993 82.68007
```

Column-wise percents

```
prop.table(table(nhanes$smoking,nhanes$asthma),margin=2)*100
#|
#|                         History of asthma       No
#|     History of smoking           42.97297 39.24910
#|     No                           57.02703 60.75090
```

- the `margin=2` argument indicates that we want column-wise percents

Fisher’s exact tests

```
fisher.test(table(nhanes$smoking,nhanes$asthma))
#|
#|      Fisher's Exact Test for Count Data
#|
#|  data:  table(nhanes$smoking, nhanes$asthma)
#|  p-value = 0.008981
#|  alternative hypothesis: true odds ratio is not equal to 1
#|  95 percent confidence interval:
#|   1.038673 1.309340
#|  sample estimates:
#|  odds ratio
#|    1.166346
```

- notice that this also returns the odds ratio

Chi-squared tests

```
chisq.test(table(nhanes$smoking,nhanes$asthma))
#|
#|      Pearson's Chi-squared test with Yates' continuity correction
#|
#|  data:  table(nhanes$smoking, nhanes$asthma)
#|  X-squared = 6.8411, df = 1, p-value = 0.008908
```

- `X-squared` is the χ^2 statistic
- `df` is the degrees of freedom

Exercise 3

Exercise 3

1. Produce a contingency table of asthma (`asthma`) and gender (`gender`), with gender along the rows.
2. Find the row-wise percents.
3. Conduct a Fisher's exact test for the association between asthma and gender.

Exercise 3

4. Store the test, using object assignment. Use `names()` to examine the values that can be extracted. Extract the odds ratio (`estimate`).

5. Extract the 95% confidence interval for the odds ratio (`conf.int`).

Exercise 3

6. Next week, we'll discuss the *ggplot2* package, which is a powerful and popular data-visualization package. Install the package, as follows:

   - Select Install in the Packages tab.
   - Type `ggplot2`.
   - Select the Install button at the bottom.

   Equivalently, type the following in the console and press the enter or return key on your keyboard:

```
install.packages('ggplot2')
```

Exercise 3

7. Verify that the package was successfully installed by checking the Packages tab.
    - In the search field at the top right, begin typing `ggplot2`.
    - Verify that the package shows up.