R Examples from BIOST 514/517

Taught by Katie Kerr, Ph.D.

Brian D. Williamson

University of Washington
Department of Biostatistics

November 21, 2014

Difference in Means

FEV data

Default in R assumes variances unequal Can use `t.test()` (`stats`) or `ttest()` (`uwIntroStats`). You can set `var.eq=T` for equal variances
Make sure to load all of the required packages and data

```
smoke <- SMOKE
sex <- SEX

## ttest for difference in means
height <- HEIGHT
ttest(height, by=sex)

Call:
ttest.default(var1 = height, by = sex)

Two-sample t-test allowing for unequal variances :

Summary:
        Group Obs Missing  Mean Std. Err. Std. Dev.          95% CI
      sex = 2 336       0 62.03     0.345      6.33  [61.346, 62.7]
      sex = 1 318       0 60.21     0.269      4.79 [59.683, 60.74]
   Difference 654       0  1.81     0.438      <NA>   [0.954, 2.67]

 Ho: difference in  means = 0 ;
 Ha: difference in  means != 0
 t = 4.144 , df = 622
 Pr(|T| > t) =  3.88659e-05
```

Calculate the confidence interval by yourself! Use the `qt()` to get t quantiles, and `qbinom()` for binomial quantiles

`prop.test()` function
Gives us a two-sided test by default (change with `alternative` option). Tests equality of the proportions

```
## difference of proportions
is_smoker <- smoke==1
successMale <- smoke==1 & sex==1
table(successMale)
prop1 <- 26
## recall that we already know the observations from the t-test above
obs1 <- 336
successFemale <- smoke==1 & sex==2
prop2 <- 39
obs2 <- 318
prop.test(c(prop1, prop2), c(obs1, obs2), correct=FALSE)

2-sample test for equality of proportions without continuity correction

data:  c(prop1, prop2) out of c(obs1, obs2)
X-squared = 3.739, df = 1, p-value = 0.05316
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.0912611312  0.0007400171
sample estimates:
    prop 1     prop 2
0.07738095 0.12264151
```

Odds Ratios and Risk Ratios

Download the `epitools` package, or calculate by hand
The `riskratio` and `oddsratio` functions will calculate for entered counts

One-sample t-test

Given count data, use the `ttesti()` function from `uwIntroStats`

```
ttesti(25, 220, 46, null.hyp=211)

 One-sample t-test
  Obs Mean Std. Error SD 95 %CI
x 25  220  9.2        46 [201.01, 238.99]

 t-statistic = 0.978 , df = 24

 Ho: mean =  211

 Ha: mean != 211 , Pr(|T| > |t|) = 1.6623
```

WCGS Data

```
ttest(sbp, by=dibpat)

Call:
ttest.default(var1 = sbp, by = dibpat)

Two-sample t-test allowing for unequal variances :

Summary:
            Group  Obs Missing   Mean Std. Err. Std. Dev.             95% CI
   dibpat = A1,A2 1589       0 129.78     0.394      15.7 [129.01, 130.55]
   dibpat = B3,B4 1565       0 127.47     0.365      14.4 [126.75, 128.18]
       Difference 3154       0   2.32     0.537      <NA>     [1.26, 3.37]

 Ho: difference in  means = 0 ;
 Ha: difference in  means != 0
 t = 4.317 , df = 3137
 Pr(|T| > t) =  1.62861e-05
```

Shoulder Pain Data

Use the `matched=TRUE` option!
Can also run a one-sample t-test as above

```
ttest(pain[time==1], pain[time==6], matched=TRUE)

Call:
ttest.default(var1 = pain[time == 1], var2 = pain[time == 6],
    matched = TRUE)

Two-sample (matched) t-test  :

Summary:
             Group Obs Missing  Mean Std. Err. Std. Dev.          95% CI
   pain[time == 1]  41       0 1.098     0.206     1.319 [0.681, 1.514]
   pain[time == 6]  41       0 0.585     0.131     0.836 [0.322, 0.849]
        Difference  41       0 0.512     0.185      1.17 [0.138, 0.886]

 Ho: difference in  means = 0 ;
 Ha: difference in  means != 0
 t = 2.766 , df = 40
 Pr(|T| > t) =  0.00855172
```

PSA Data

Use the `tabulate` function

Use the same function in the PBC data set examples

```
tabulate(grade, bss)

Call:
tabulate.default(grade, bss)
             bss.1 bss.2 bss.3 bss.NA bss.NotNA bss.ALL
grade.1       1     3     6     0      10        10
grade.2       2     4     9     0      15        15
grade.3       2     4     9     1      15        16
grade.NA      0     2     6     1       8         9
grade.NotNA   5    11    24     1      40        41
grade.ALL     5    13    30     2      48        50
             Point Estimate Test Statistic df 95% CI p-value Warnings
Chi-squared                 4.3115         16        0.99825
```

Fisher's Exact Test

Use the `fisher.test()` function, or the `tabulate()` function with the `tests="fisher"` option

Survival

Create a `Surv` variable, plot the curve with confidence interval

To plot cumulative hazard function, use the `fun="cumhaz"` option

To compare survival times, use the `survdiff` function from the `survival` package

```
relapse <- ifelse(inrem=="no", 1, 0)
atrisk <- Surv(obstime, relapse)
survFit <- survfit(atrisk~1)
plot(survFit)
```