BIOS 6312: Modern Regression Analysis

Andrew J. Spieker, Ph.D.

Assistant Professor of Biostatistics
Vanderbilt University Medical Center

Set 1 supplementary slides for `R` enthusiasts

Version: 01/25/2021

Preface to supplementary slides

Examples for R enthusiasts:

- Based on your initial feedback, there seems to be a high degree of enthusiasm for opportunities to code in R.
  - R can be downloaded at `https://www.r-project.org`.
- In response to this feedback, many of the notes will have a corresponding supplementary set of slides in which one or more examples from the notes are worked out in R.
- These supplementary slides can serve as a resource for folks who have a strong preference for conducting analyses in R.

Preface to supplementary slides

Disclaimers:

- If your preference is to perform analyses in R, it will not be counted against you so long as your solution is correctly implemented. However, I make the following disclaimers:
  1. Many methods are more easily implemented in Stata (e.g., robust variance estimation, omnibus tests, survival analysis, penalized regression, Bayesian regression). Conducting analyses involving these methods in Stata will likely save you time.
  2. You will still be given Stata output on exams and will be expected to interpret salient results. The output is often (but not always) self explanatory (e.g., coefficient estimates are often labeled "`coef.`"). Familiarity with Stata output layout will therefore be to your benefit.
  3. There are often many ways to do the same thing in R. These notes will often only show one such way, and will try to minimize (where possible) dependencies on other packages.

Examples for Set 1

Examples for R enthusiasts:

- Analysis of log-transformed FEV (Slide 88)

Examples for Set 1

Examples for R enthusiasts:

- **Analysis of log-transformed FEV (Slide 88)**

Analysis of log-transformed FEV

Reading in data sets:

- Function: `read.csv()`.
- Stores data set as an object in environment.

```
fev.data <- read.csv("fev.csv",
                     header = TRUE,
                     stringsAsFactors = FALSE)
```

Analysis of log-transformed FEV

Log-transforming FEV:

- Create a new variable for log-transformed FEV (attaching it to the original data).

```
fev.data$logfev <- log(fev.data$fev)
```

Analysis of log-transformed FEV

Histograms:

- Create and plot histograms of log-transformed FEV.

```
h1 <- hist(fev.data$logfev[fev.data$sex == 1],
           breaks = seq(-0.5, 2, 0.1))
h2 <- hist(fev.data$logfev[fev.data$sex == 2],
           breaks = seq(-0.5, 2, 0.1))
```

- The histograms will be plotted one by one using this code, but we want to display them together.

Analysis of log-transformed FEV

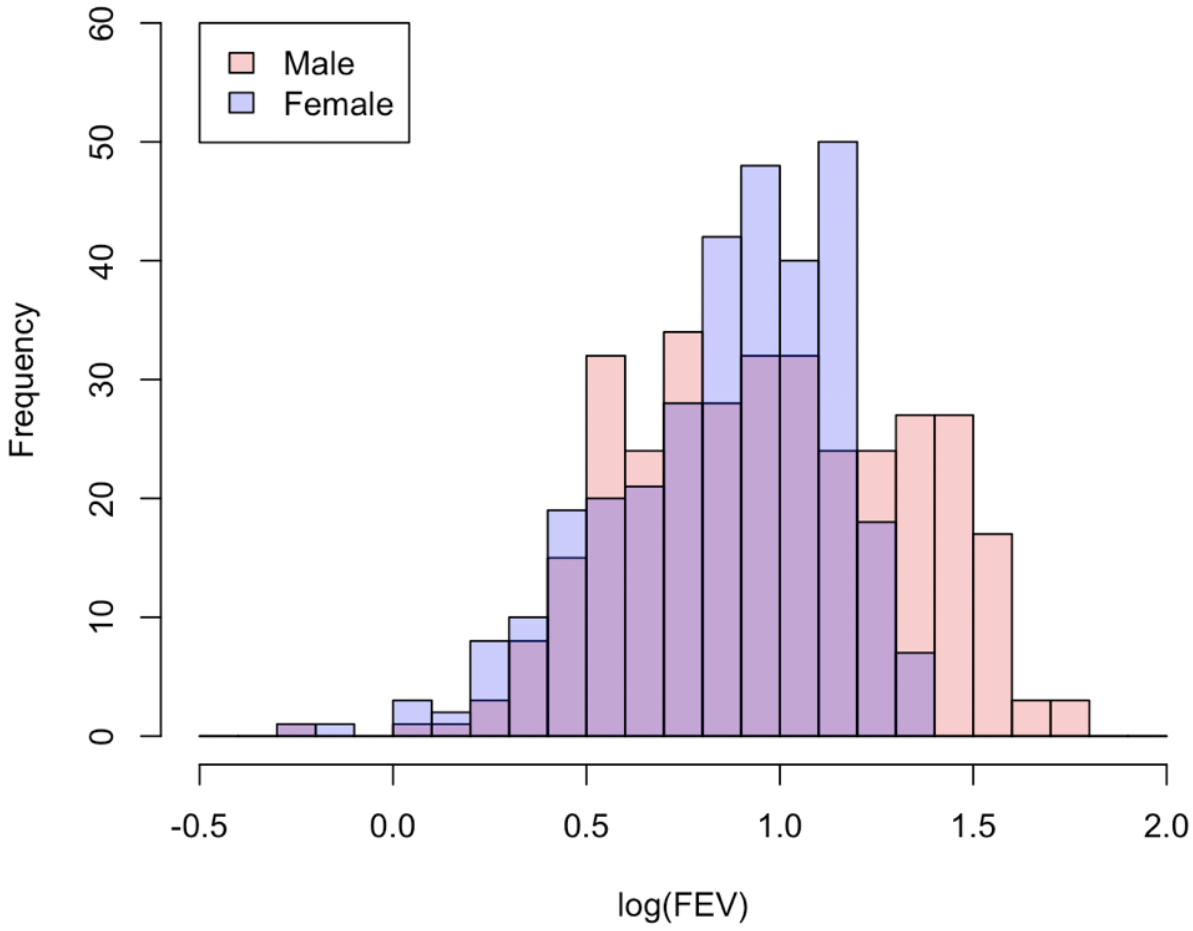
Histograms:

- Create histograms of log-transformed FEV.

```
plot(h1, col=rgb(1,0,0,1/4),
     xlim=c(-0.5,2),
     ylim = c(0,60),
     xlab = "FEV",
     main = "")
plot(h2, col=rgb(0,0,1,1/4),
     xlim=c(0,2), add=T)
legend(-0.5, 60,
       fill = c(rgb(1,0,0,1/4), rgb(0,0,1,1/4)),
       c("Male", "Female"))
```

Analysis of log-transformed FEV

Histograms:



Analysis of log-transformed FEV

Two-sample *t*-test: Three equivalent ways

- Method 1:

```
result <- t.test(x = fev.data$logfev[fev.data$sex == 1],
                 y = fev.data$logfev[fev.data$sex == 2],
                 alternative = "two.sided",
                 paired = FALSE,
                 var.equal = FALSE,
                 conf.level = 0.95)
```

Analysis of log-transformed FEV

Two-sample *t*-test: Three equivalent ways

- Method 2:

```
result <- t.test(x = fev.data$logfev[fev.data$sex == 1],
                 y = fev.data$logfev[fev.data$sex == 2])
```

- This is equivalent because all of the options on the previous slide were indeed the default options.

Analysis of log-transformed FEV

Two-sample *t*-test: Three equivalent ways

- Method 3:

```
result <- t.test(formula = logfev ~ sex, data = fev.data)
```

- Knows to compare variable on the left between groups defined by the variable on the right.

Analysis of log-transformed FEV

Two-sample *t*-test: Output

- Command "`print(result)` ällows you to read off results.
- You can also extract particular quantities of interest from the *t*-test object.

```
> result$estimate
mean in group 1 mean in group 2
         0.9704          0.8574
> result$conf.int
[1] 0.06283 0.16324
attr(,"conf.level")
[1] 0.95
> result$p.value
[1] 1.155e-05
```

Analysis of log-transformed FEV

Two-sample *t*-test: Output

- Better yet, you can exponentiate quantities of interest.

```
> as.numeric(exp(result$estimate[1] -
+                  result$estimate[2]))
[1] 1.12
> as.numeric(exp(result$conf.int))
[1] 1.065 1.177
```

Analysis of log-transformed FEV

Two-sample *t*-test: Output

- If you want more digits, all you have to do is ask nicely! :)

```
> options(digits = 5)
> as.numeric(exp(result$estimate[1] -
+                  result$estimate[2]))
[1] 1.1197
> as.numeric(exp(result$conf.int))
[1] 1.0648 1.1773
```

- The geoemtric mean ratio and corresponding confidence interval match the results we obtained in Stata (Slide 94).