Designing simulation studies using R

- R is a statistical programming environment
  - Free, open source
  - User-contributed code, including many cutting edge statistical methods (and some junk)
  - Good for designing simulations

- Takes some initial effort to learn the basics
  - Scripts rather than point & click
  - Objects
  - Functions

Simulation

1. Generate data from a probability model where you know the true parameters
2. Apply an estimation method to the data
3. See how close the estimate gets to the truth

Useful for

- Checking comprehension
- Building intuitions, testing hunches
- Formal research
  - Small-sample performance
  - Robustness, mis-specified models

Two-sample difference in means (The Behrens-Fisher problem)

- Control group data

$$Y_1^C, \ldots, Y_{n_C}^C \sim N\left(\mu_C, \sigma_C^2\right)$$

- Treatment group data

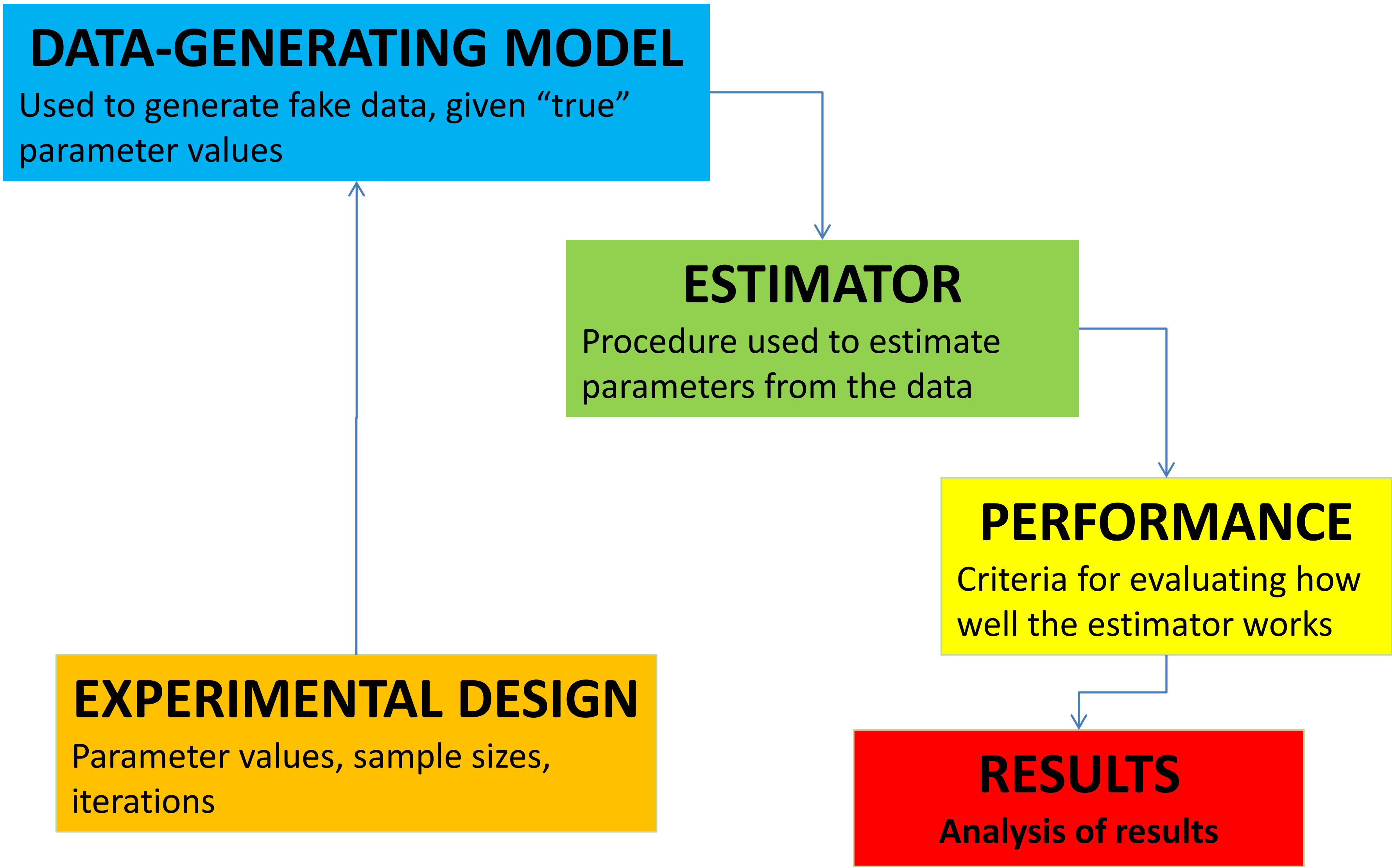
$$Y_1^T, \ldots, Y_{n_T}^T \sim N\left(\mu_T, \sigma_T^2\right)$$

- Want a 95% confidence interval for $\mu_T - \mu_C$

A very simple simulation

```
covered <- replicate(20000, {
  Y_C <- rnorm(n = 8, mean = 0, sd = 1)
  Y_T <- rnorm(n = 4, mean = 2, sd = 2.5)
  test <- t.test(x = Y_T, y = Y_C, var.equal = FALSE)
  (test$conf.int[1] < 2) & (2 < test$conf.int[2])
})

mean(covered)
```

Simulation design



Data-Generating Model

$$Y_1^C, \ldots, Y_{n_C}^C \sim N\left(0,1\right)$$

$$Y_1^T, \ldots, Y_{n_T}^T \sim N\left(\delta, R\right)$$

Parameters:

- δ = effect size
- R = variance ratio
- n = total sample size ($n_T + n_C$)
- p = proportion of sample in control group (n_C / n)

Data-Generating Model in R

```
two_group_data <- function(iterations, n, p, var_ratio, delta) {
   Group <- c(rep("C", n * p), rep("T", n * (1 - p)))
   Y_C <- matrix(rnorm(iterations * n * p, mean = 0, sd = 1),
                  n * p, iterations)
   Y_T <- matrix(rnorm(iterations * n * (1 - p),
                  mean = delta, sd = sqrt(var_ratio)),
                  n * (1 - p), iterations)
  dat <- data.frame(Group, rbind(Y_C, Y_T))
  return(dat)
}
```

Estimator

95% confidence interval

$$\left(\bar{y}_T - \bar{y}_C\right) \pm t\left(0.025, df\right) \times \sqrt{\frac{s_C^2}{n_C} + \frac{s_T^2}{n_T}}$$

Welch's degrees of freedom approximation

$$df = \frac{\left(\frac{s_C^2}{n_C} + \frac{s_T^2}{n_T}\right)^2}{\frac{s_C^4}{n_C^2\left(n_C - 1\right)} + \frac{s_T^4}{n_T^2\left(n_T - 1\right)}}$$

Estimator in R

```
CI_welch <- function(dat, alpha = 0.05) {
   CI <- apply(dat[,-1], 2, function(X)
              t.test(X ~ dat$Group, var.equal=FALSE,
                     conf.level = 1 - alpha)$conf.int)
   return(t(CI))
}
```

Performance Criteria

- Confidence interval coverage
  - What proportion of calculated confidence intervals contain the true value of δ?

Performance Criteria in R

```r
coverage <- function(CI, delta) {
  covered <- (CI[,1] < delta) & (delta < CI[,2])
  return(mean(covered))
}
```

Experimental Design

- Parameters:
  - δ (effect size) = 0
  - R (variance ratio) = ¼, ½, 1, 2, 4
  - n (total sample size) = 12, 24, 36, 48, 60
  - p (proportion in control group) = 1/2, 1/3, 1/4, 1/6
- Design: 5 × 5 × 4 full factorial
- Iterations = 1000

Experimental Design in R

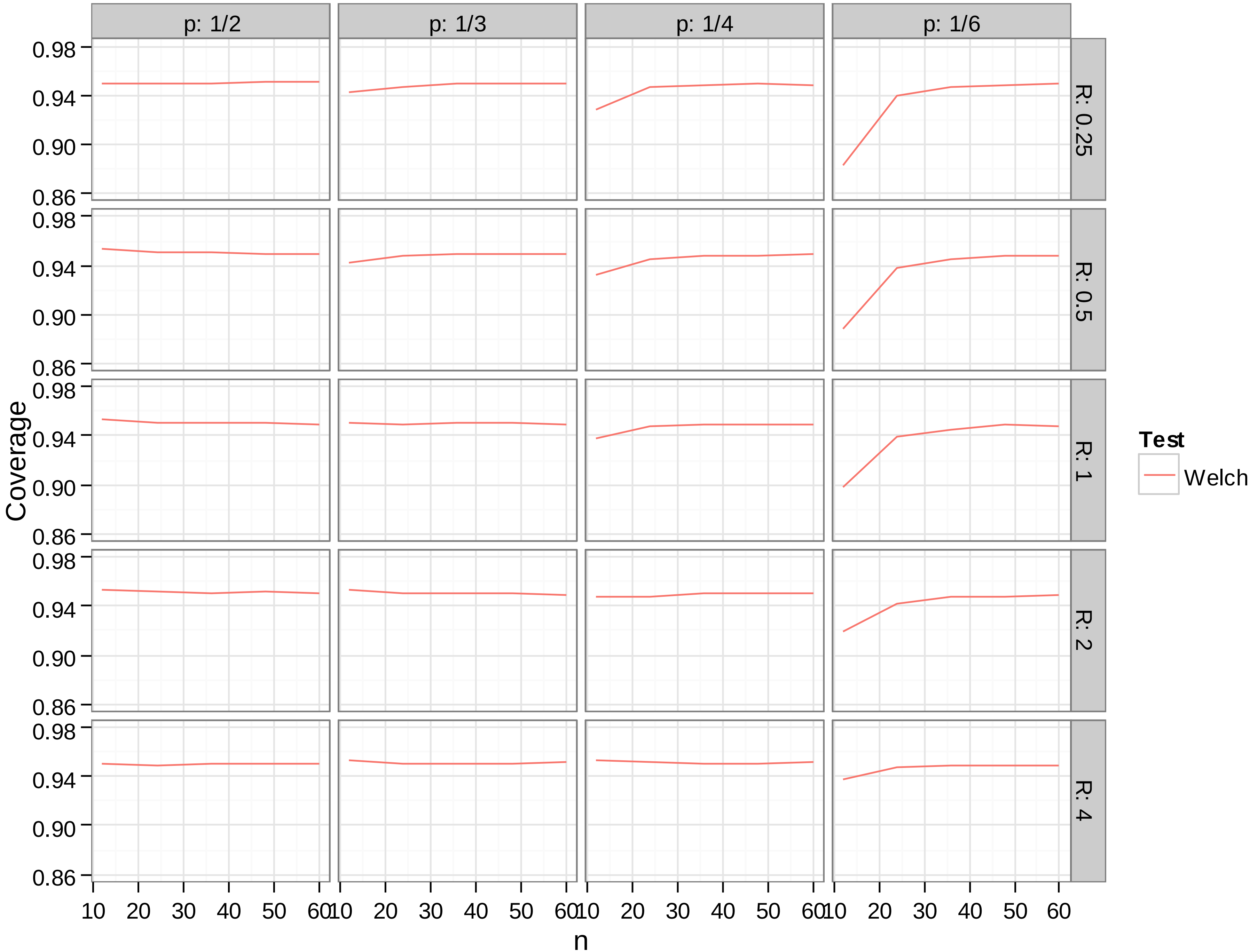
```
delta <- 0
R <- c(1/4, 1/2, 1, 2, 4)
n <- c(12,24,36,48,60)
p <- c(1/2, 1/3, 1/4, 1/6)
iterations <- 1000

parms <- expand.grid(iterations = iterations,
                     n = n, p = p, var_ratio = R, delta = delta)
```

Putting it all together

```
#-----------------------------------------------
# simulation driver
#-----------------------------------------------
run_sim <- function(iterations, n, p, var_ratio, delta) {
  dat <- two_group_data(iterations, n, p, var_ratio, delta)
  Welch <- coverage(CI_welch(dat), delta)
  return(c(Welch = Welch))
}

#---------------------------------------
# run simulations in serial
#---------------------------------------
library(plyr)
set.seed(20110325)
system.time(results_serial <- maply(parms, .fun = run_sim))
save(results_serial, file="BF results.Rdata")
```

Analyzing Results

```
#--------------------------------------
# plot results
#--------------------------------------
library(reshape)
library(ggplot2)

load("BF results.Rdata")
dimnames(results)$p <- c("1/6","1/4","1/3","1/2")
names(dimnames(results))[4] <- "R"
results_long <- melt(results)

qplot(data = results_long,
      x = n, y = value,
      geom = "line") +
  facet_grid(R ~ p, labeller = label_both) +
  labs(y = "Coverage") + theme_bw()
```


p: 1/2
p: 1/3
p: 1/4
p: 1/6
R: 0.25
R: 0.5
R: 1
R: 2
R: 4
Coverage
0.98
0.94
0.90
0.86
n
10
20
30
40
50
60
Test
Welch

Further development

- Improve efficiency of the code
- Add other estimators
  - Cochran-Cox (1950)
  - Banerjee (1961)
  - Patil (1965)
  - Bayesian credible interval
  - Equal-variance estimator
- Robustness to non-normality
- Running simulations in parallel
  - On your desktop
  - In the Visualization Lab
  - On the TACC

Learning more

- Full code posted on my blog
  - http://blogs.edb.utexas.edu/pusto/blog/

- “Computing for Data Analysis” on Coursera
  - https://www.coursera.org/course/compdata
  - 4 weeks, starting 1/6/2014

- EDP 384: Data Analysis and Simulation
  - Spring 2015

- Matloff (2011). The Art of R Programming.
  - Useful for more advanced projects

- Kim & Cohen (1998). On the Behrens-Fisher problem: A review. *Journal of Educational and Behavioral Statistics, 23*(4), 356-377.