2017-02-13

Rick O. Gilmore

2017-02-12

Contents

Today's topics

- Some observations about GitHub use and R Markdown
- Simulation as a tool for reproducible and transparent science
- Visualization tools in R

Observations about GitHub use

- You're in charge of what goes where.
- Public repos are public, but no one knows what you're doing unless you alert them.
  - If you create a repo in your own account, add `rogilmore` or other collaborators to use the `@rogilmore` type at-mentions feature.
  - Files you create in `https://github.com/psu-psych-511-2017-spring` I can already comment on.

Observations about GitHub use

- Pull requests are when you edit my code and want me to "pull"/adopt it.
  - If I'm a collaborator on the project with `write` privileges, I don't have to issue a pull request.

Observations about R Markdown

- Ok to make multiple R Markdown files
- Make sure to add spaces where they belong: `##Header` vs. `## Header`
- Comments! Add them. This is your record of what you did.
- Don't forget you can hide things

Be a risk-taker; be your own professor

- http://www.stat.cmu.edu/~cshalizi/rmarkdown
- http://stat545.com/bit006_github-browsability-wins.html

Simulation as a tool for reproducible and transparent science

- Why simulate
- What to simulate
- How to simulate

Why & what to simulate?

- Explore sample sizes, effect sizes, power
- Pre-plan/test, polish data-munging workflows
- Make hypotheses even more explicit
  - Simulation == Pregistration on steroids
  - '~~X affects Y~~' -> 'Mean(X) > Mean(Y)'
  - or 'Mean(X) >= 2*Mean(Y)'
- Simulate data analysis in advance

- Plan data visualizations in advance
- Avoid avoidable errors
- Plan your work, work your plan
- Super easy to run analyses when your data come in

How to simulate

- R functions
- R Markdown document(s)

Super-simple example

- Hypothesis 1: Height (inches) is correlated with weight (lbs)

```
# choose sample size
sample.n <- 200

# choose intercept and slope
beta0 <- 36   # inches
beta1 <- 0.33  # Rick's guess
```

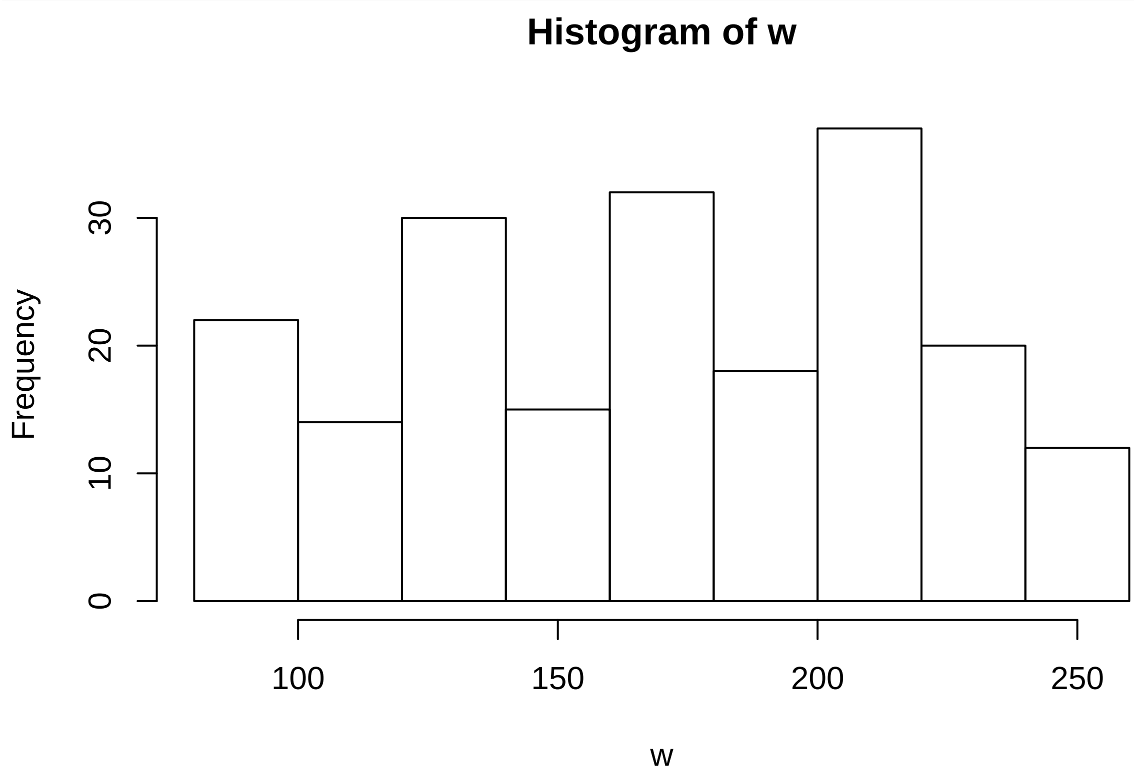
```r
# choose standard deviation for error
sigma <- 10 # Rick's guess
```

```r
# random weights between 80 lbs and 250 lbs (uniform sampling)
w <- runif(n = sample.n, min = 80, max = 250)

h.pred <- rep(x = beta0, n = sample.n) + beta1 * w
h <- h.pred + rnorm(n = sample.n, mean = 0, sd = sigma)
```

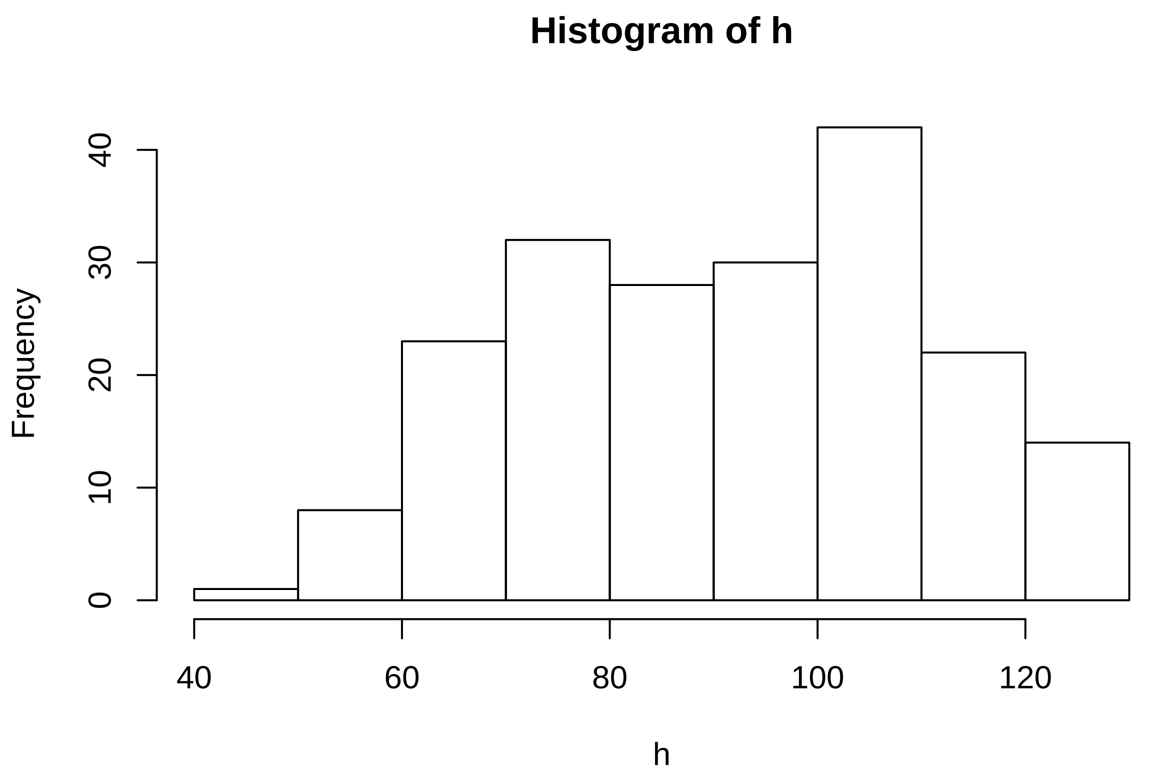
```r
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

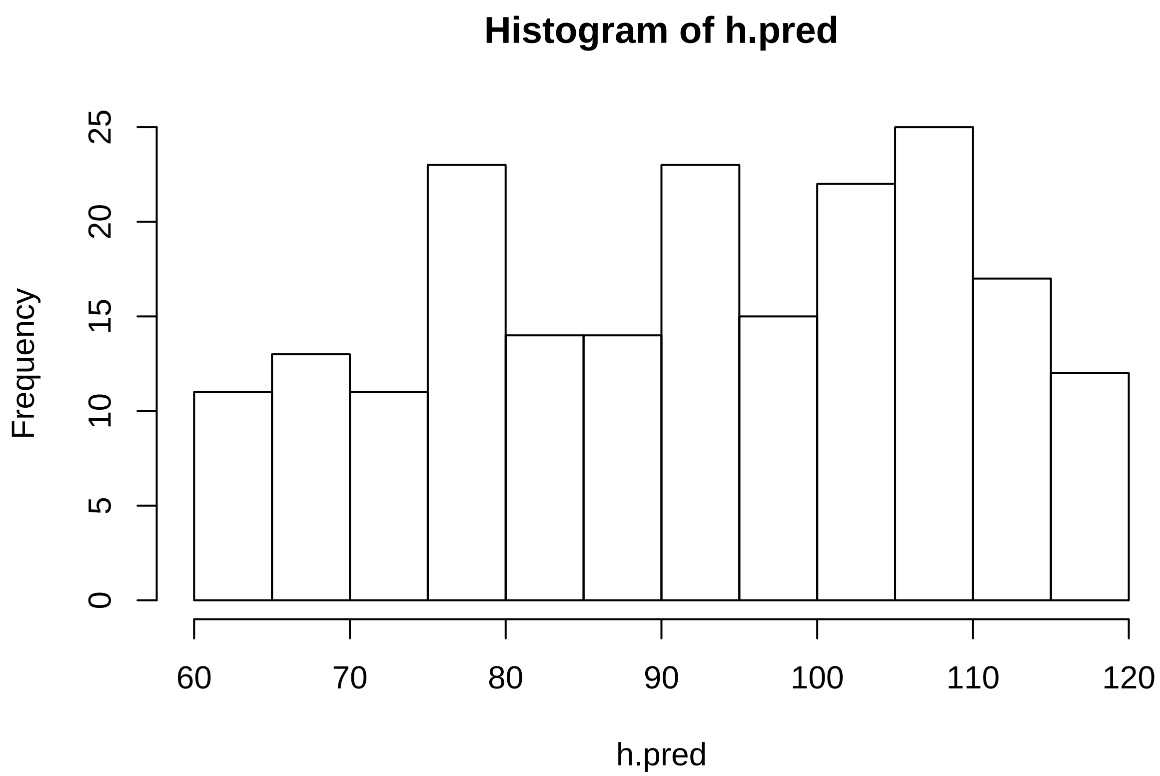
```r
hist(w)
```




```r
hist(h)
```

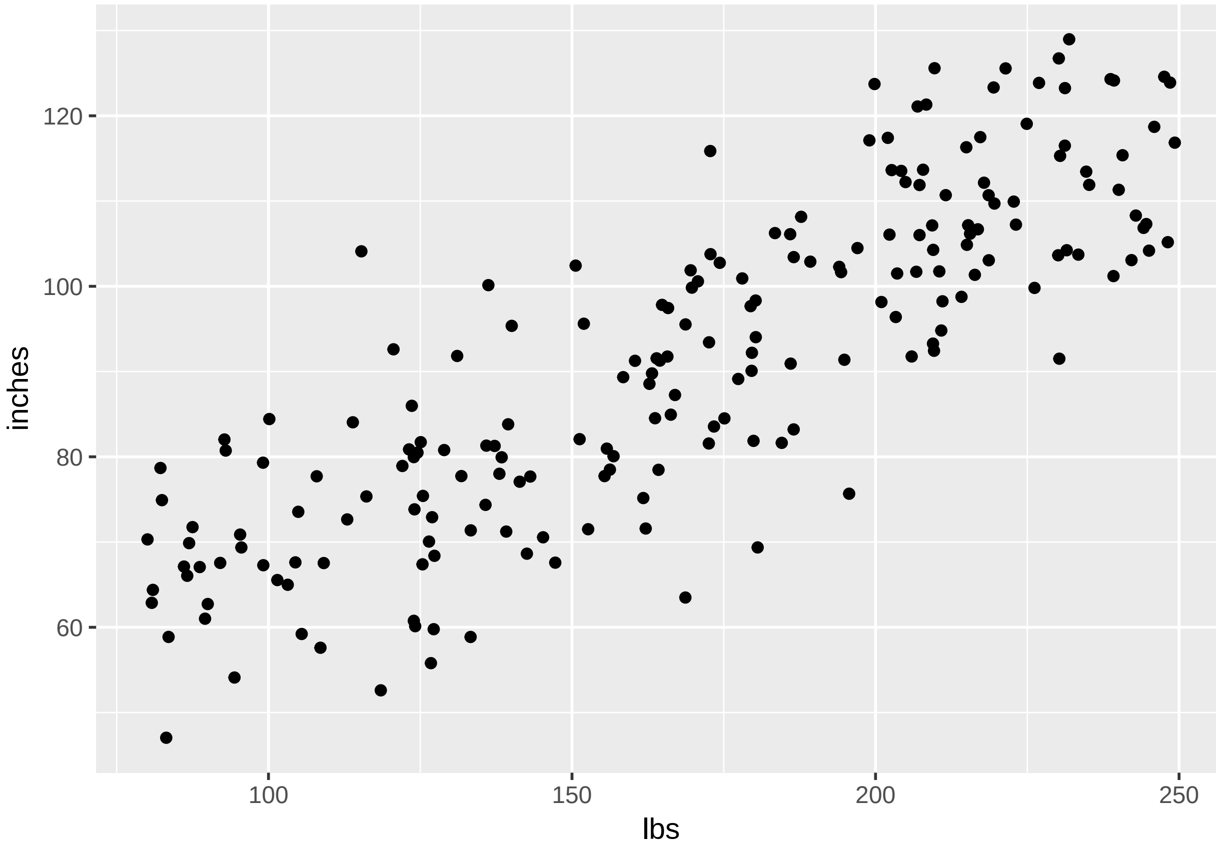



```r
hist(h.pred)
```



```r
# Put h and w into data frame for ggplot
height.weight <- data.frame(inches = h, lbs = w)
```

```r
# Plot
scatter.1 <- ggplot(data = height.weight) +
  aes(x = lbs, y = inches) +
  geom_point()
scatter.1
```



That's synthesis, now analysis

- Remember Hypothesis 1: Height (inches) is correlated with weight (lbs)?

```r
# Could use the raw data
# cor.test(x = w, y = h)
# Or, to use the values in the data frame, use with(...)

with(height.weight, cor.test(x = inches, y = lbs))
```

```
##
##  Pearson's product-moment correlation
##
## data:  inches and lbs
## t = 21.973, df = 198, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7964358 0.8782522
## sample estimates:
##       cor
```

```
## 0.8421236
```

Aside: extracting the statistics to make an interactive report

```r
# Save output as a variable
cor.test.inches.lbs <- with(height.weight, cor.test(x = inches, y = lbs))

# What sort of beast is this?
mode(cor.test.inches.lbs)
```

```
## [1] "list"
```

```r
# Aha, it's a list, this shows me all of the parts
unlist(cor.test.inches.lbs)
```

```
##                              statistic.t
##                       "21.9730525428121"
##                             parameter.df
##                                    "198"
##                                  p.value
##                   "5.33047884992903e-55"
##                             estimate.cor
##                      "0.842123572561128"
##                   null.value.correlation
##                                      "0"
##                              alternative
##                              "two.sided"
##                                   method
## "Pearson's product-moment correlation"
##                                data.name
##                        "inches and lbs"
##                                conf.int1
##                      "0.796435788839413"
##                                conf.int2
##                      "0.878252169338884"
```

```r
# Looks like the t value is the first element
cor.test.inches.lbs[[1]]
```

```
##        t
## 21.97305
```

The Pearson's product-moment correlation correlation between height and weight is .3f, $t(198)$=21.9730525, p=0.00000, with a 95% confidence interval of [0.7964358, 0.8782522].

Obviously, we should do some formatting before submitting this, but you get the idea.

Now back to analysis with our synthetic data

```r
fit <- lm(formula = inches ~ lbs, data = height.weight)
summary(fit)
```

```
##
## Call:
## lm(formula = inches ~ lbs, data = height.weight)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -27.6668  -7.2363   0.4818   7.2520  30.6205
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 35.29486    2.64766   13.33   <2e-16 ***
## lbs          0.33119    0.01507   21.97   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.31 on 198 degrees of freedom
## Multiple R-squared:  0.7092, Adjusted R-squared:  0.7077
## F-statistic: 482.8 on 1 and 198 DF,  p-value: < 2.2e-16
```

```
(ci <- confint(fit)) # saves in variable ci and prints
```

```
##                 2.5 %     97.5 %
## (Intercept) 30.073622 40.5160972
## lbs          0.301466  0.3609126
```

How'd we do?

Parameter	Actual	Low Estimate	High Estimate
$\beta 0$	36	30.0736221	40.5160972
$\beta 1$	0.33	0.301466	0.3609126

- Why off on the slope ($\beta 1$)
- Random error, probably. Could run again.

```
##
## Call:
## lm(formula = inches ~ lbs, data = height.weight)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -26.5352  -7.0397   0.2031   7.0904  26.9966
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 37.60306    2.32734   16.16   <2e-16 ***
## lbs          0.32373    0.01407   23.00   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.19 on 198 degrees of freedom
## Multiple R-squared:  0.7277, Adjusted R-squared:  0.7263
## F-statistic: 529.2 on 1 and 198 DF,  p-value: < 2.2e-16
```

```
##                  2.5 %     97.5 %
## (Intercept) 33.0135074 42.1926074
## lbs          0.2959732  0.3514776
```

Parameter	Actual	Low Estimate	High Estimate
$\beta0$	36	33.0135074	42.1926074
$\beta1$	0.33	0.2959732	0.3514776

Simulation of fMRI data

- Critical review: Welvaert, M., & Rosseel, Y. (2014). A Review of fMRI Simulation Studies. PLOS ONE, 9(7), e101953. https://doi.org/10.1371/journal.pone.0101953.
- Welvaert, M., Durnez, J., Moerkerke, B., Berdoolaege, G. & Rosseel, Y. (2011). neuRosim: An R Package for Generating fMRI Data. Journal of Statistical Software, 44(10). Retrieved from https://www.jstatsoft.org/article/view/v044i10
- AFNI's *AlphaSim*, https://afni.nimh.nih.gov/pub/dist/doc/program_help/AlphaSim.html

Visualization in R

Plot first, analyze last

- Why?
- Mike Meyer told me so
- Less biased
- Easier to be transparent and reproducible
- Want/need to plot eventually anyway
- If a picture's worth a thousand words…
- How?

How

- Base graphics
  - `plot(x,y)` `hist(x)`, `coplot()`
- `ggplot2`
  - Grammar of graphics

Base graphics

- Try it, maybe you'll like it
- `plot()` takes many types of input
- So does `summary()`
- A little harder to customize

Data visualization with ggplot2

Wickham, H. & Grolemund, G. (2017). *R for Data Science*. O'Reilly. http://r4ds.had.co.nz/

Let's just walk through the data visualiation chapter

http://r4ds.had.co.nz/data-visualisation.html

Other ggplot2 resources

- Wickham, H. (2010). *ggplot2: Elegant Graphics for Data Analysis (Use R!)* http://ggplot2.org/book/
- ggplot2 2.1.0 documentation: http://docs.ggplot2.org/current/