BIFX 553 - Confidence Intervals

Randy Johnson

March 2, 2017

Setup

```
library(gmodels)
library(tidyverse)
library(broom)
theme_set(theme_classic() +
            theme(axis.line.x = element_line(color = 'black'),
                  axis.line.y = element_line(color = 'black'),
                  text = element_text(size = 15)))
```

Confidence Intervals

Confidence Interals

Two results will help us understand the derivation of confidence intervals:

- Central Limit Theorem
- Law of Large Numbers

Central Limit Theorem

The mean, $\bar{x}$, of n independent, identically distributed random variables, X, with well defined expected value, $\mathrm{E}(X) = \mu$, and variance, $\mathrm{Var}(X) = \sigma$, will be approcimately normally distributed when n is sufficiently large:

$$\bar{x} \sim \mathrm{N}\left(\mu, \frac{\sigma}{\sqrt{n}}\right)$$

Central Limit Theorem: Simulation

Load the R function found at `http://tinyurl.com/zenq9q3`.

```
# Normal distribution, sample size of 10
clt.test(rnorm, 10)

# Chi-squared distribution sample size of 5
clt.test(rchisq, 5, df = 3)

# Bimodal mixture of Normals, sample size of 10
rbimodal <- function(n)
{
  m <- rbinom(n, 1, 0.5) %>%
       as.logical()

  return(ifelse(m, rnorm(n),
                   rnorm(n, 5, 2)))
}
clt.test(rbimodal, 10)
```

Law of Large Numbers

Given our sample mean, $\bar{x}$, the Law of Large Numbers states that $\bar{x}$ will converge to the true population mean as the sample size increases, assuming the sample, X, are independent, identically distributed random variables.

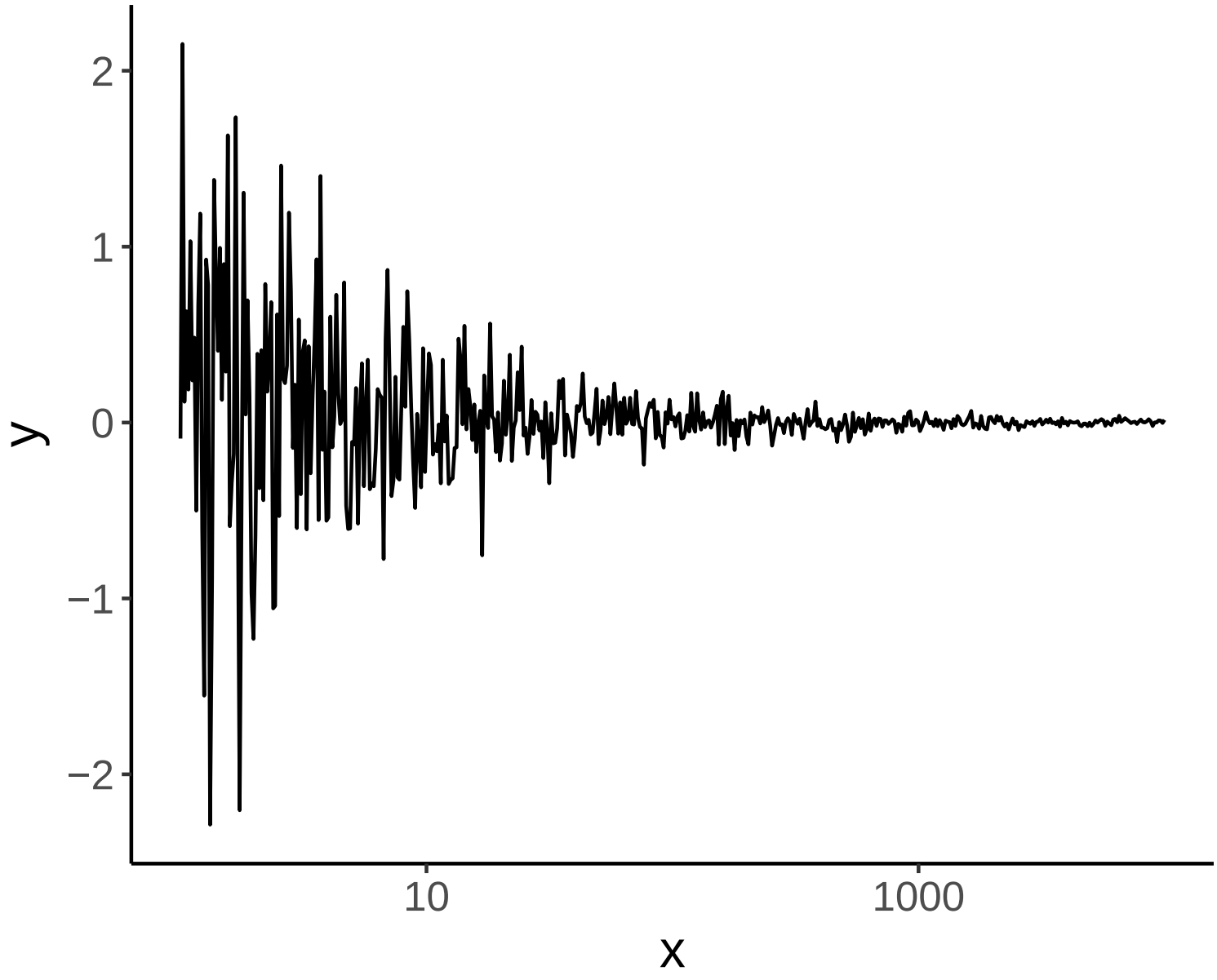
$$\bar{x} \xrightarrow{n \to \infty} \mu$$

Law of Large Numbers: Simulation

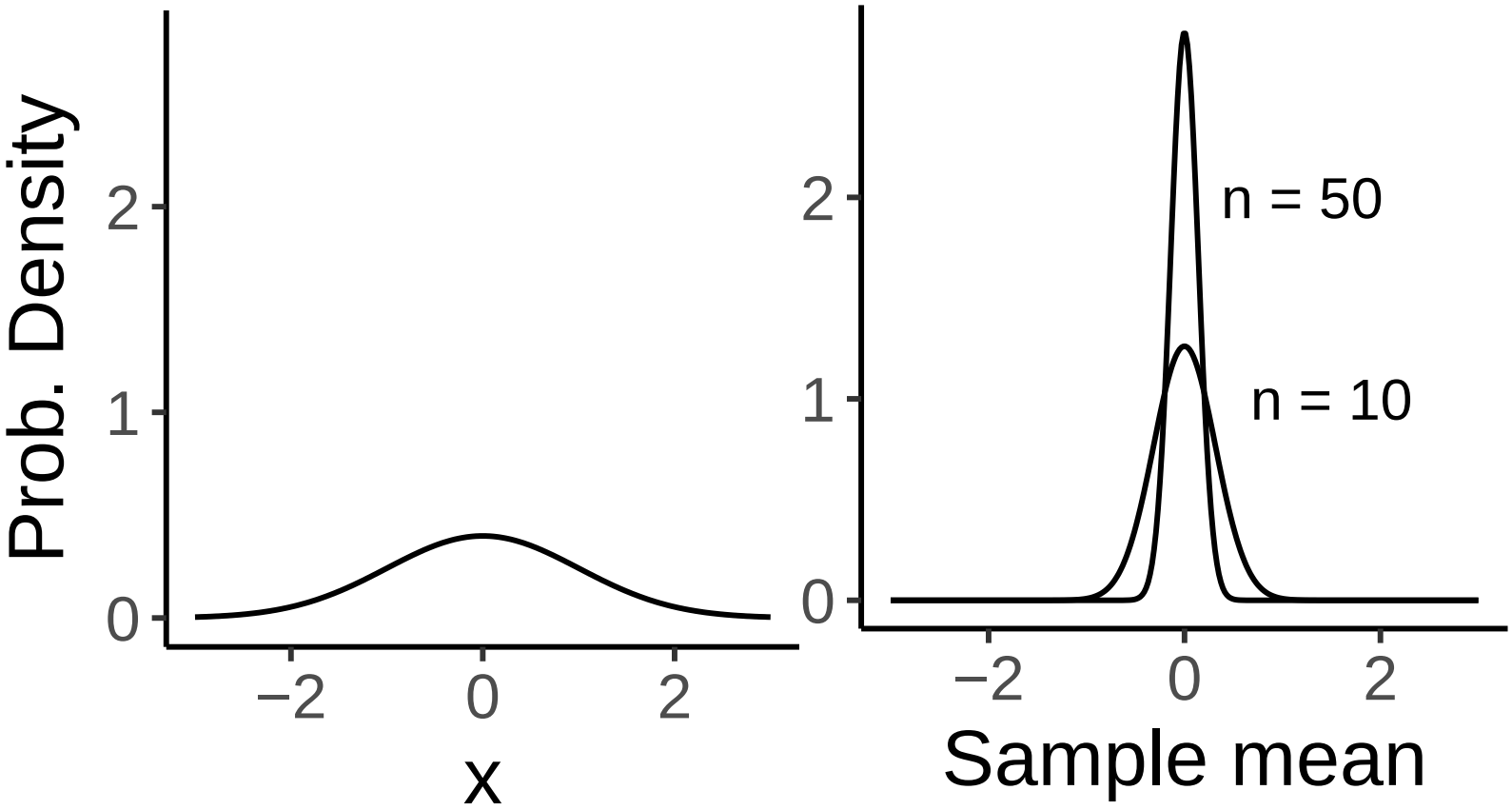
```
set.seed(293874)
lln <- data_frame(x = 10^seq(from = 0, to = 4,
                             length = 500),
                  y = {map(x, rnorm) %>%
                       map(mean) %>%
                       unlist()})

g <- ggplot(lln, aes(x, y)) +
     geom_line() +
     scale_x_log10()
```

Law of Large Numbers: Simulation

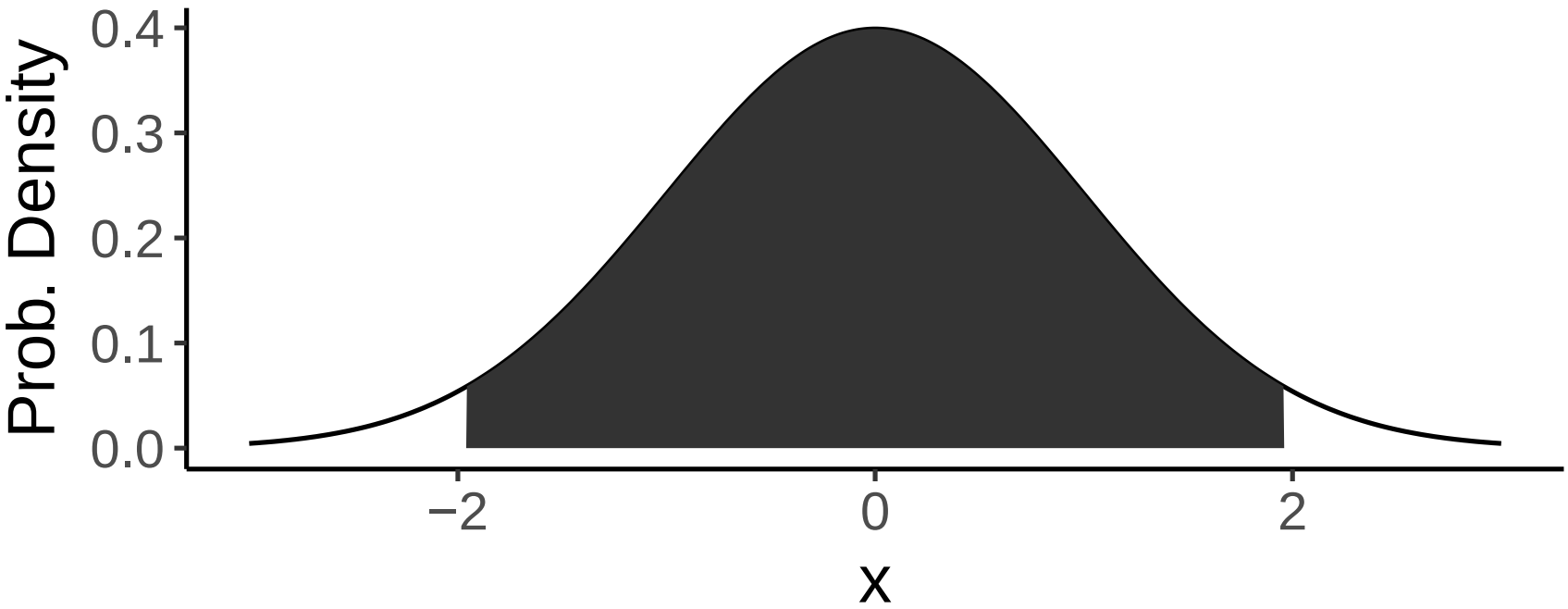


The distribution of $\bar{x}$



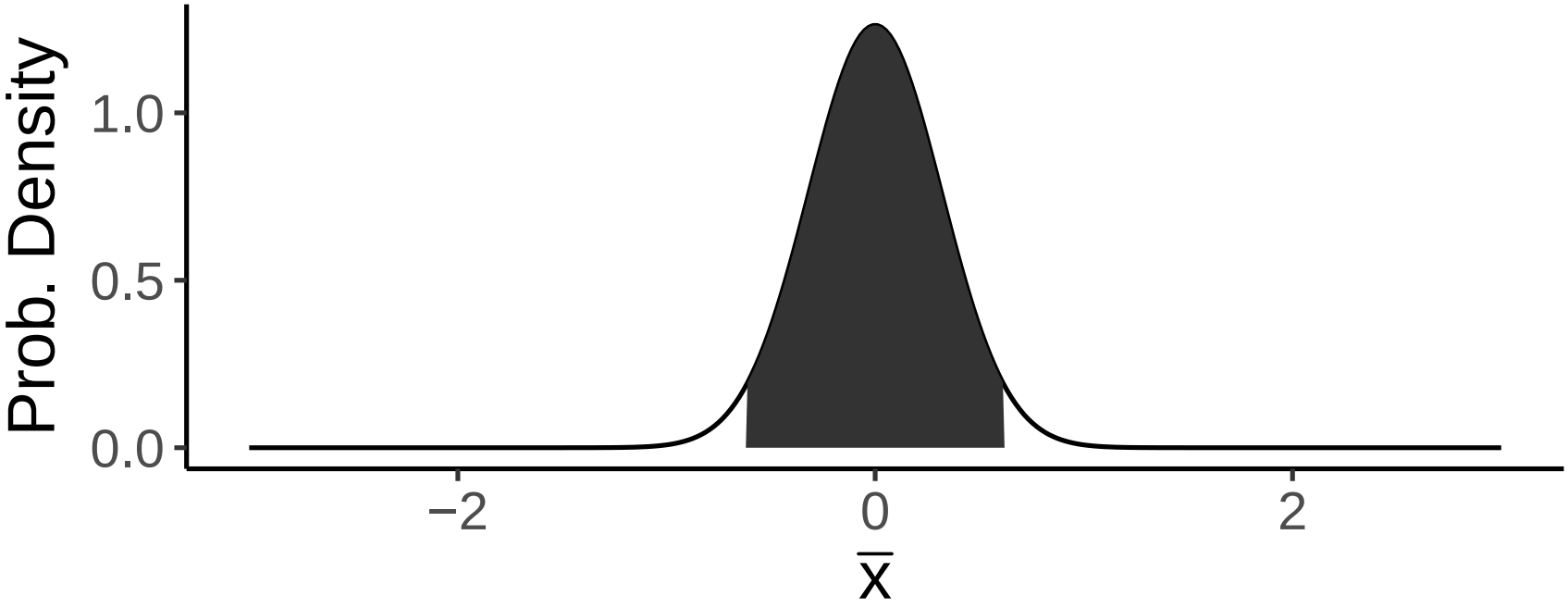
95% Confidence Region

95% of the samples of x we collect will fall in $\mu \pm 1.95\sigma$. This forms the basis of our confidence interval. Side note: the area under this curve over the range $(-\infty, \infty)$, and every probability distribution, is 1.



95% Confidence Interval construction

When the sample size is 10, the distribution of $\bar{x}$ looks like this. 95/% of the time we will expect $\bar{x}$ to fall in the region $\left(\mu \pm \frac{1.95\sigma}{\sqrt{n}}\right)$. From this, we infer that we are 95% confident that the true mean lies within the interval $\left(\bar{x} \pm \frac{1.95*sd}{\sqrt{n}}\right)$.



95% CI Example 1

Lets say that we are studying a population, and we have a sample of 100 blood systolic preasure measurements. The mean is 123 and the standard deviation is 12. What is our confidence interval?

$$\bar{x} = 123, sd(x) = 12, n = 100$$

$$95\%\ \mathrm{CI}(\mu) = 123 \pm \frac{1.95 * 12}{\sqrt{100}} = (120.66, 125.34)$$

95% CI with gmodels

Let's simulate a similar data set in R and use the gmodels package to calculate the CI.

```
set.seed(29874)
rnorm(100, 123, 12) %>%
  ci()
```

```
##   Estimate   CI lower   CI upper Std. Error
## 124.023961 121.599383 126.448540   1.221932
```

95% CI of lm object

A more practical use of `ci()` can be applied to the homework from a few weeks ago.

```
load('../Data/06_NonLinearVariables.RData')
lm(y ~ x1*x2, data = dat1) %>%
  ci()
```

```
##                Estimate   CI lower   CI upper Std. Error      p-value
## (Intercept)  0.08622145 -0.1120740  0.2845169 0.10054832 3.922090e-01
## x1          -0.83619322 -1.0211558 -0.6512306 0.09378771 3.391344e-16
## x2           1.85230919  1.5638783  2.1407401 0.14625265 2.995820e-27
## x1:x2        0.65742039  0.3376731  0.9771677 0.16213205 7.234576e-05
```

95% CI of lm object

Now it is your turn. What is the 95% CI for the $x1$ variable in the second dataset from a few weeks ago? The model is provided here for your convenience. What about the 90% CI?

```
lm(log(y) ~ x1, data = dat2)
```

Estimates with 95% CIs

Now, suppose we want to know the 95% CI of the expected number of nodes with detectable cancer in a woman with the following measurements:

- size = 23
- grade = 2
- pgr = 32.5
- hormon = "no tamoxifen"

As you may recall, the model we chose for this last week was

```
model <- lm(lnodes ~ size + grade + lpgr + hormon, data = gbsg)
coef(model)
```

```
##         (Intercept)                size               grade
##          0.43767500          0.01948081          0.13101418
##                lpgr hormonno tamoxifen
##         -0.02429904         -0.08218227
```

Estimates with 95% CIs

`ci()` will give us the confidence intervals for each of the betas, but won't get us very far with a specific prediction. We can get the prediction using `predict()`, but what is the standard error of the prediction?

```
predict(model, data.frame(size = 23, grade = 2,
                          lpgr = log(32.5),
                          hormon = 'no tamoxifen')) %>%
  exp() %>%
  signif(2)
```

```
##   1
## 2.7
```

Estimates with 95% CIs

We can use the `estimable()` function to give us CI's.

```
estimable(model, cm = c(1, 23, 2, signif(log(32.5), 1), 1),
          conf.int = 0.95)[c(1,6,7)]
```

```
##                 Estimate  Lower.CI Upper.CI
## (1 23 2 3 1) 0.9926826 0.9028564 1.082509
```

Estiates with 95% CIs

We can also use `estimable()` to explore more complicated questions. For example, we could ask, is there a difference in the expected number of expected nodes between a woman with a grade 2, 27 mm tumor and a woman with a grade 3, 20 mm tumor?