Lab 5

Confidence Intervals

We will start by seeing how to calculate a confidence interval in R.

```
# number of observations in the sample
n <- 20
# number of covariates (must be less than n)
p <- 5

# Draw p covariates which are all independent of each other
X <- mvtnorm::rmvnorm(n, sigma = diag(p))
# The true coefficients are 1
beta <- rep(1, p)
# draw Y with no Y intercept and Gaussian errors
Y <- X %*% beta + rnorm(n)

# fit the model
mod1 <- lm(Y~X)

# The confidence interval for each coefficient can be calculated using confint
confint(mod1, level = .95)
```

```
##                  2.5 %    97.5 %
## (Intercept) -0.5398088 0.2333885
## X1           0.3483936 1.0636614
## X2           0.2103559 1.4801440
## X3           0.5081140 1.5118630
## X4           0.6875714 1.4748006
## X5           0.7965363 1.4905851
```

The confidence interval you calculate will be different from your neighbor's interval (it will even change each time you knit this document), and it either does or does not contain the true parameter (which happens to be 1). So it doesn't make sense to say "there's a 95% chance that $[.49, 1.5]$ (or whatever numbers you actually got) contains 1."

Questions

- Check around the room and see how many people got a confidence interval that does not contain 1. What proportion of the students have a confidence interval that contains the truth? Hopefully it's close to .95.

Instead of making a statement about a specific interval, the 95% chance describes the probability that the procedure will contain the truth when applied to a new set of data. Or thought another way, when repeating the procedure an infinite number of times with new data each time, 95% of the resulting confidence intervals will contain the true parameter. We use the phrase:

"We are 95% confident that the true coefficient is between $[.49, 1.5]$ (or whatever numbers you actually got)"

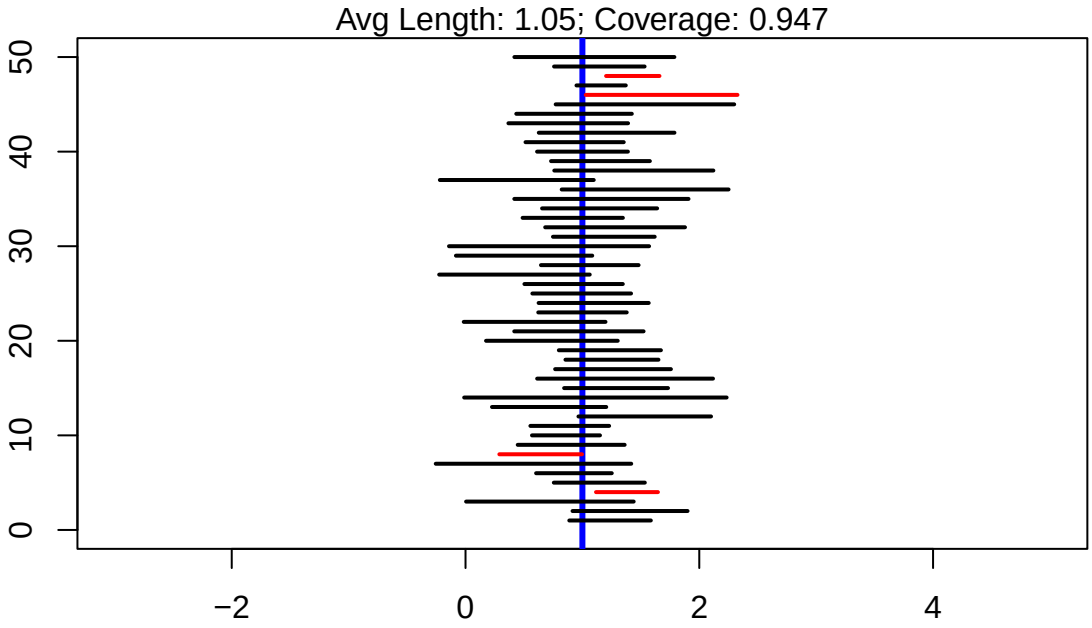
as a shorthand way of encoding that longer explanation.

CI simulations

We go back to the same code from last lab, where we looked at the sampling distribution of $\hat{b}_1$. But this time, we will look at confidence intervals and how the length of confidence intervals change depending on characteristics of the population and the confidence level of the CI.

```
#### Change the code below ####

# number of observations in the sample
n <- 20
# number of covariates (must be less than n)
p <- 3
# standard deviation of the X values
x.sd <- 1
# covariance of the X values (choose a positive value less than x.sd)
rho <- .2

# distribution of the errors (choose either "normal" or "gamma" or "T")
# gamma distribution is skewed, T distribution has outliers
errDist <- "gamma"

# Standard deviation of the epsilon terms
err.sd <- 1

# The target proportion of times that the confidence interval should contain the truth
ci_level <- .95


#### Don't change the code below ####
# Number of times we will simulate a new data set
sim.size <- 5000
# True coefficients
beta <- rep(1, p)
ci <- matrix(0, sim.size, 2)
# Covariance matrix of X
cov.mat <- matrix(rho, p,p); diag(cov.mat) <- x.sd^2

# Helper function to draw errors
drawErrors <- function(n, sd = 1, type = "normal"){
  if(type == "normal"){
    err <- rnorm(n)
  } else if (type == "gamma"){
    err <- rgamma(n, 1, 1) - 1
  } else if (type == "T"){
    err <- rt(n, df = 3)
  }
  return(err * sd)
}

for(i in 1:sim.size){
  X <- mvtnorm::rmvnorm(n, sigma = cov.mat)
  Y <- X %*% beta + drawErrors(n, sd = err.sd, type = errDist)
```

```r
  mod1 <- lm(Y~X)

  ### use confint
  confidence_interval <- confint(mod1, level = ci_level)
  ci[i, ] <- confidence_interval[2, ]
}


avgLength <- mean(ci[,2] - ci[,1])
coverage <- mean(ci[,1] < 1 & ci[,2] > 1)
plot(-5, 5, xlim = c(-3, 5), ylim = c(0, 50), xlab = "", ylab = "")
abline(v = 1, col = "blue", lwd = 3)
# only plot 50 of the 5000 replications
for(i in 1:50){
  segments(ci[i, 1], i, ci[i, 2], i, col = ifelse(ci[i,1] < 1 & ci[i,2] > 1, "black", "red"), lwd =2)
}
mtext(paste("Avg Length: ", round(avgLength, 2), "; Coverage: ", round(coverage,3), sep = ""), cex = 1)
```



Questions

- Change the population parameters and see how the average length of the confidence intervals change?
- Why might you choose a higher coverage probability? Why might you choose a lower coverage probability?

Hypothesis testing

In class we discussed a process for calculating a p-value when testing the null hypothesis $H_0 : b_k = \beta$ where $\beta = 0$. Fortunately, R does this all for you automatically.

```
# Sample errors from a normal distribution with mean 0 and sd = 1
errs <- rnorm(n, mean = 0, sd = 1)

# Sample errors from a gamma distribution with mean 0 and sd = 1
# errs <- rgamma(n, 1, 1) - 1

b <- rep(1, p)
# Form the dependent variable Y
Y <- X %*% b + errs

# Fit the regression
# we include the -1 term to tell R not to add in an intercept term
# since we've manually included the column of 1's in the matrix X
reg_norm <- lm(Y ~X - 1)

# In the summary function
# First column gives the estimated hat b_k
# Second column gives the estimated sd of each individual coordinate
# Third column gives the t statistic when testing the null hypothesis that b_k = 0
# Fourth column gives the p-value when testing the null hypothesis that b_k = 0
summary(reg_norm)
```

```
##
## Call:
## lm(formula = Y ~ X - 1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.58487 -0.61847  0.08735  0.65272  1.37525
##
## Coefficients:
##    Estimate Std. Error t value Pr(>|t|)
## X1   0.7638     0.2171   3.518  0.00264 **
## X2   1.1791     0.1837   6.418 6.37e-06 ***
## X3   1.3794     0.1881   7.335 1.17e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8891 on 17 degrees of freedom
## Multiple R-squared:  0.929,  Adjusted R-squared:  0.9165
## F-statistic: 74.13 on 3 and 17 DF,  p-value: 5.725e-10
```

```
# $coefficients pulls out the table
summary(reg_norm)$coefficients
```

```
##     Estimate Std. Error  t value     Pr(>|t|)
## X1 0.7637562  0.2170909 3.518140 2.638207e-03
## X2 1.1790552  0.1837075 6.418112 6.365168e-06
## X3 1.3794478  0.1880647 7.334965 1.165272e-06
```

```r
# estimated sd of hat b_1
summary(reg_norm)$coefficients[2,2]
```

```
## [1] 0.1837075
```

Questions

- How would you interpret the p-value (4th column) in the second row (which corresponds to b_1 since the first row corresponds to the intercept)?

Sampling distribution under null and alternative

Suppose we posit that $b_1 = \beta$. So far, we've described the distribution of

$$t = \frac{\hat{b}_1 - \beta}{\sqrt{\widehat{\text{var}}(\hat{b}_1)}}$$

when the null hypothesis is true. But what about when the null hypothesis is false? Turns out this is a little bit harder, but we can simulate the distribution. Suppose we are interested in the null hypothesis where $\beta = 0$; i.e.,:

$$H_0 : b_1 = 0 \qquad \text{vs} \qquad H_A : b_1 \neq 0.$$

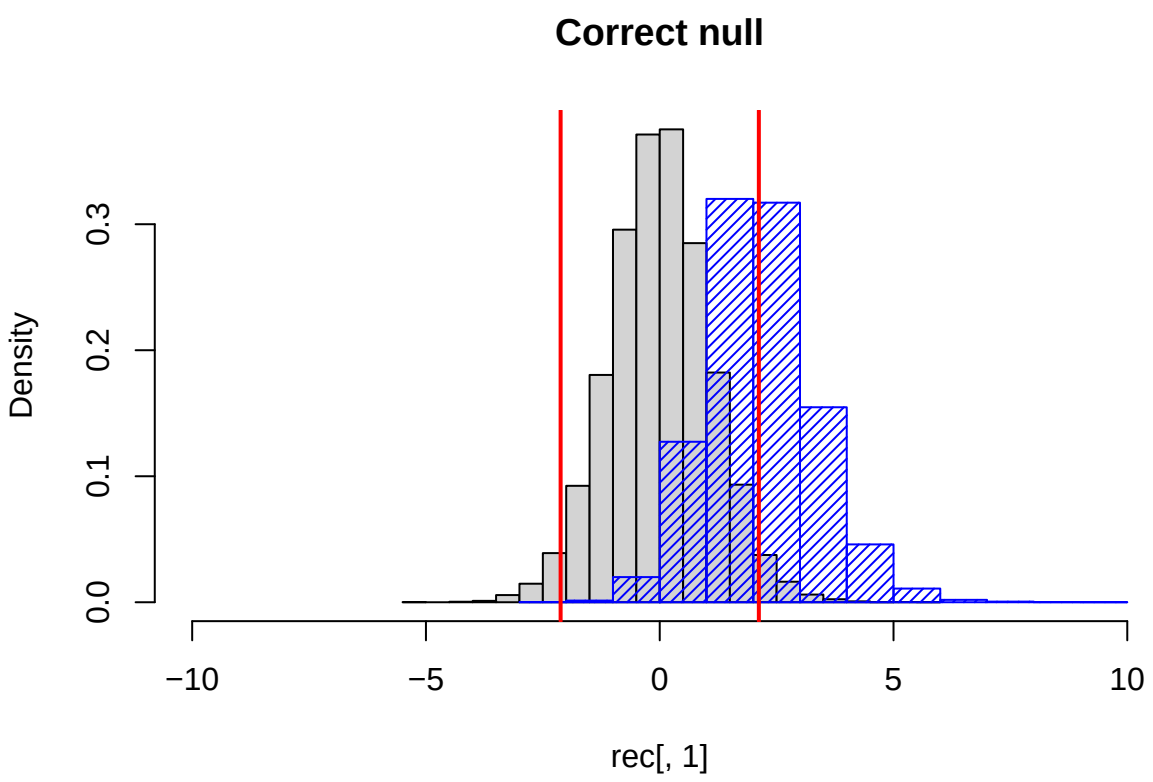
We will consider two different settings, one where the null hypothesis is true and $b_1 = 0$ and another where the null hypothesis is false and $b_1 = .5$

```
# Number of times we will simulate a new data set
sim.size <- 20000

# number of observations
n <- 20
# number of covariates
p <- 3
# standard deviation of the X values
x.sd <- 1

# drawing the covariates from a normal distribution
X <- matrix(rnorm(n * p, sd = x.sd), n, p)

# coefficients which fit the null hypothesis
b.trueNull <- rep(1, p)
# set the coefficient corresponding to b_1 to 0
b.trueNull[1] <- 0

# coefficients which do not fit the null hypothesis
b.falseNull <- rep(1, p)
# set the coefficient corresponding to b_1 to .5
b.falseNull[1] <- .5



# recording the estimated values for each simulated data set
rec <- matrix(0, sim.size, 4)

for(i in 1:sim.size){

  # Form the dependent variable Y where the null hypothesis is true
    Y.trueNull <- X %*% b.trueNull +  rnorm(n, mean = 0, sd = 1)
    # Y.trueNull <- X %*% b.trueNull +  (rgamma(n, 1, 1) - 1)

    # Form the dependent variable Y where the null hypothesis is false
    Y.falseNull <- X %*% b.falseNull +  rnorm(n, mean = 0, sd = 1)
    # Y.falseNull <- X %*% b.falseNull +  (rgamma(n, 1, 1) - 1)


    # Fit the regression
```

```
    # we include the -1 term to tell R not to add in an intercept term
    # since we've manually included the column of 1's in the matrix X
    reg.trueNull <- lm(Y.trueNull ~X)
    reg.falseNull <- lm(Y.falseNull ~X)

    # record the calculated t statistic and the resulting p-value in each case
    rec[i, ] <- c(summary(reg.trueNull)$coeff[2, 3], summary(reg.trueNull)$coeff[2, 4],
                  summary(reg.falseNull)$coeff[2, 3], summary(reg.falseNull)$coeff[2, 4])
}
```

We can plot the distribution of the t statistic when null hypothesis is correct in gray and the distribution of the statistic when the null hypothesis is false in blue. Since we know the theoretical distribution of the statistic when the null is correctly specified is a T distribution with $n - p - 1$ degrees of freedom, we can calculate cutoffs which would result in committing a Type I error only $\alpha = .05$ where α is the pre-determined level of our test.

```
hist(rec[, 1], main = "Correct null", freq = F, xlim = c(-10, 10))
hist(rec[, 3], main = "Incorrect null", add = T, density = 25, col = "blue", freq = F)

# qt(.025, n - p - 1) returns a number x such that an observation
# from a t distribution with n-p-1 degrees of freedom would only be smaller
# than x, .025 of the time

# qt(.975, n - p - 1) returns a number x such that an observation
# from a t distribution with n-p-1 degrees of freedom would be smaller
# than x, .975 of the time. Or put differently, a value would only be larger
# than x .025 of the time

# Thus, the total probability that a value would only be smaller or larger
# than the cut-offs is .05.
# The cutoffs are shown in red
alpha <- .05

lower_cutoff <- qt(alpha / 2, n - p - 1)
high_cutoff <- qt(1 - alpha / 2, n - p - 1)
abline(v = c(lower_cutoff, high_cutoff), lwd = 2, col = "red")
```



We can see that for the correctly specified hypothesis, we mistakenly reject the null hypothesis roughly .05 of the time. However, the blue distribution (when the null hypothesis is actually false) falls beyond the cut-offs roughly .45 of the time.

```
# Get the proportion of times we would incorrectly reject the hypothesis that's correctly specified
mean(rec[,1] < lower_cutoff | rec[,1] > high_cutoff)
```

```
## [1] 0.0502
```

```
# Get the proportion of times we would correctly reject the hypothesis that's incorrectly specified
mean(rec[,3] < lower_cutoff | rec[, 3] > high_cutoff)
```

```
## [1] 0.48665
```

Questions

- In the plot, what region corresponds to a type I error? What region corresponds to a Type II error?

The proportion of the times we reject the null hypothesis **when it is actually false** is known as the **power** of the test. The power of the test depends on many things including sample size and what the actual truth is (i.e., how wrong is the null hypothesis?).

Questions

- Suppose in the true model, $b_1 = 1$. Would we reject more/less often than when $b_1 = .5$? Why? How would the blue distribution look different?
- Suppose in the true model, $b_1 = .5$, but we have $n = 50$. Would we reject more/less often than when $n = 20$? Why? How would the blue distribution look different?
- What if we set $\alpha = .1$ so that we allowed a Type I error .1 of the time, but kept everything else the same? Would we reject more/less often than when $\alpha = .05$? What would change about the plot?

- Test out your conjectures in R by changing the code
- Why might you choose a larger α? Why might you choose a smaller α?

Testing multiple coefficients

Now let's consider using an F-test to test for the coefficients for multiple covariates at once. Below, we have 10 covariates, but only the first 5 actually are associated with Y.

```
# number of observations in the sample
n <- 200
# number of covariates (must be less than n)
p <- 10
# standard deviation of the X values
x.sd <- 1
# covariance of the X values (choose a positive value less than x.sd)
rho <- .2

cov.mat <- matrix(rho, p,p); diag(cov.mat) <- x.sd^2


beta <- c(rep(1, 5), rep (0, 5))
X <- mvtnorm::rmvnorm(n, sigma = cov.mat)
Y <- X %*% beta + drawErrors(n, sd = err.sd, type = errDist)

# Only include first 3 relevant variables
mod1 <- lm(Y~X[,1:3])
# Include all 5 relevant variables
mod2 <- lm(Y~X[,1:5])
# Include all 5 relevant variables plus 5 irrelevant variables
# (since the coefficients are 0 in the true model)
mod3 <- lm(Y~X)

### F test using anova command to compare two different linear models
anova(mod1, mod2)
```

```
## Analysis of Variance Table
##
## Model 1: Y ~ X[, 1:3]
## Model 2: Y ~ X[, 1:5]
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    196 638.82
## 2    194 153.28  2    485.54 307.27 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
### F test using anova command to compare two different linear models
anova(mod2, mod3)
```

```
## Analysis of Variance Table
##
## Model 1: Y ~ X[, 1:5]
## Model 2: Y ~ X
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    194 153.28
## 2    189 151.79  5    1.4863 0.3701 0.8687
```

Questions

- When comparing mod1 to mod2, write out the null hypothesis we are testing?
- When comparing mod2 to mod3, write out the null hypothesis we are testing?
- How does the RSS of mod1 compare to the RSS of mod2?
- How does the RSS of mod2 compare to the RSS of mod3?