Part 1 - Multiple Regression

2023-03-12

- How to do hypothesis testing?
- How to find a confidence interval?
- What is a confidence region?
- What does 95% confidence interval mean?
- What is a confidence band? Illustrate the confidence band using the 04cars dataset.

Notes are adapted from Yu Zhao's DSC241 lab sessions at UCSD.

How to do hypothesis testing?

Consider the Boston data and the linear model: $medv = \beta_0 + \beta_1 * crim + \beta_2 * nox + \beta_3 * rm + \beta_4 * age + \beta_5 * dis + \epsilon$

```
lm.medv = lm(medv ~ crim + nox + rm + age + dis,data = Boston)
summary(lm.medv)
```

```
##
## Call:
## lm(formula = medv ~ crim + nox + rm + age + dis, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.313  -2.917  -0.785   1.979  38.442
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -6.22734    4.01469  -1.551    0.122
## crim         -0.20808    0.03404  -6.113 1.97e-09 ***
## nox         -18.05089    3.94709  -4.573 6.06e-06 ***
## rm            7.73531    0.39542  19.562  < 2e-16 ***
## age          -0.06662    0.01514  -4.400 1.33e-05 ***
## dis          -1.19104    0.21675  -5.495 6.23e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.901 on 500 degrees of freedom
## Multiple R-squared:  0.5924, Adjusted R-squared:  0.5883
## F-statistic: 145.3 on 5 and 500 DF,  p-value: < 2.2e-16
```

Test the hypothesis that the slope $\beta_1 = 0$, against the alternative that $\beta_1 < 0$. Intuitively, higher crime rates will lead to lower house price. We want to test whether this intuition is true or not. β_1 is the coefficient of crime rate. We want to test whether $\beta_1 < 0$, which means when the level of crime rates goes up, the house price will go down. For example if $\beta_1 = -0.2$, it means that when the crime rate goes up by 1 unit, the house price will go down by 0.2 units.

How to do that?
–> Calculate t ratio:

- By hand using formula
  In general, we can test for linear combination of the coefficients. Indeed, if $\mathbf{c} = \left(c_0, c_1, \ldots, c_p\right) \in \mathbb{R}^{p+1}$, then

$$\frac{\mathbf{c}^\top \widehat{\boldsymbol{\beta}} - \mathbf{c}^\top \boldsymbol{\beta}}{\widehat{\mathrm{SE}}\left(\mathbf{c}^\top \widehat{\boldsymbol{\beta}}\right)} \sim \mathcal{T}_{n-p-1}$$

  where

$$\widehat{\mathrm{SE}}\left(\mathbf{c}^\top \widehat{\boldsymbol{\beta}}\right) = \hat{\sigma}\sqrt{\mathbf{c}^\top \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{c}}$$

  is the (estimated) standard error of $\mathbf{c}^\top \widehat{\boldsymbol{\beta}}$.

- By using the summary function

```
tval <- (coef(lm.medv)[2] - 0)/ sqrt(vcov(lm.medv)[2,2]) # use second coef, which is crim
df <- dim(Boston)[1] - 2
data.frame(tval = tval, df=df, pval = 1 - pt(abs(tval), df))
```

```
##           tval  df         pval
## crim -6.112813 504 9.812102e-10
```

The p-value is 9.8121022^{-10} < 0.05. Therefore we reject the null hypothesis, the alternative hypothesis is true, which means higher crime rates do lead to lower house price.

How to find a confidence interval?

Alternatively, we can use `confint` to calculate confidence interval.

```
confint(lm.medv, "crim",level = 0.95)
```

```
##           2.5 %     97.5 %
## crim -0.2749623 -0.1412027
```

We can also use confidence interval to test hypothesis. For example if we want to test wheter $\beta = 0$ with 95% confidence level, we can look at the 95% confidence level and see whether $\beta = 0$ is included in the interval. In this example, we can see that $\beta = 0$ is not included, so we reject the hypothesis.

If we want to test the hypothesis that $\beta_3 = \beta_4 = 0$, we can use F test.
We can: - calculate the F value by hand - use ANOVA

```
lm.medv.h0 = lm(medv ~ crim + nox + dis,data = Boston)  # if beta_3 = beta_4 = 0, we only have 3 predictors left
anova(lm.medv.h0, lm.medv) # then we can compare the full model with the reduced model
```

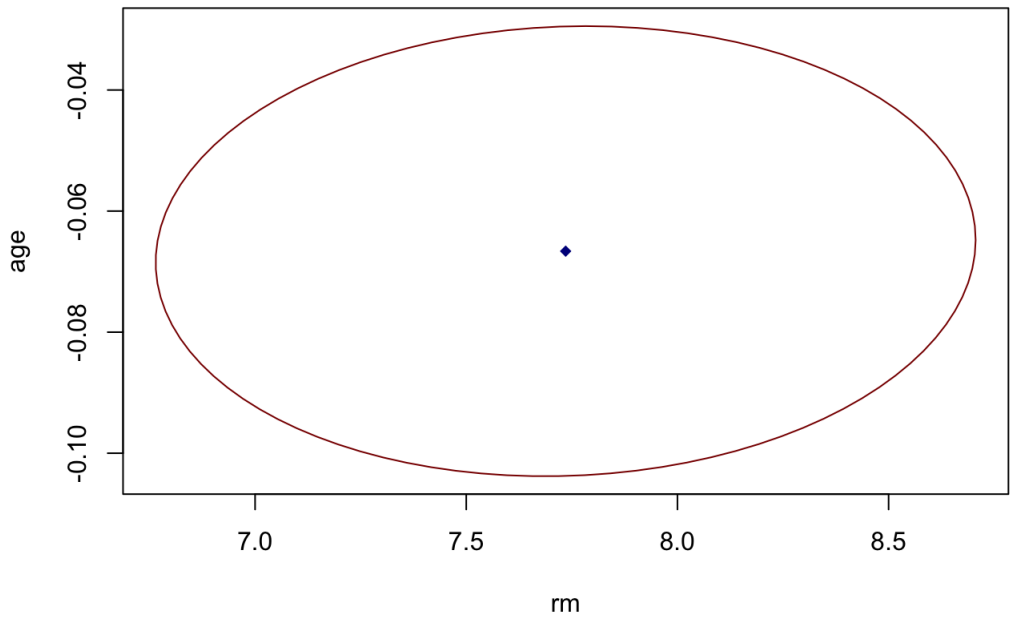
```
## Analysis of Variance Table
##
## Model 1: medv ~ crim + nox + dis
## Model 2: medv ~ crim + nox + rm + age + dis
##   Res.Df   RSS Df Sum of Sq      F    Pr(>F)
## 1    502 31749
## 2    500 17412  2     14337 205.85 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value is less than 0.05. We reject the null hypothesis, claiming at least one of the coefficient is not 0.

What is a confidence region?

When we have multiple coefficients, the confidence “interval” is no longer a 2d interval, it’s an ellipsoid. How do we represent this? –> Confidence Region, which also follows the F distribution.

```
plot(ellipse(lm.medv,which = c(4,5),level = 0.95),type = 'l',col = 'darkred') #confidence region
points(lm.medv$coefficients[4],lm.medv$coefficients[5],col = 'darkblue',pch = 18) #estimated value
```



What does 95% confidence interval mean?

The confidence level represents the long-run proportion of corresponding CIs that contain the true value of the parameter. For example, out of all intervals computed at the 95% level, 95% of them should contain the parameter's true value.

If we repeat our experiment multiple times, 95% the confidence intervals contain the true value. The point is that the confidence interval is random, not the true value.

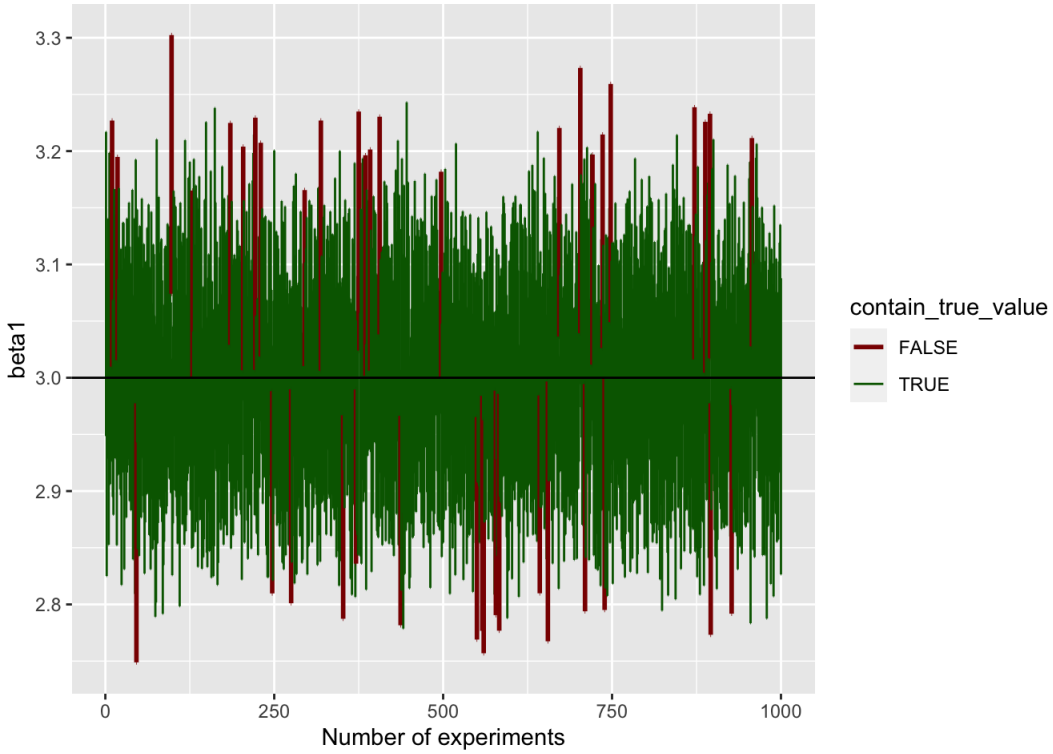
How do we validate this? Simulation

```
B = 1000 # number of simulations
beta0 = 1
beta1 = 3
n = 100 # sample size
CI = data.frame(lower = rep(NA,B),upper = rep(NA,B),
                contain_true_value = rep(NA,B))
              # create a data frame to store the lower bounds, upper bounds and indicator


for (i in 1:B) {
  x = rnorm(n)
  y = beta0 + beta1*x + 0.5*rnorm(n) # linear model with 1 predictor and added random residual errors
  lm.sim = lm(y~x)
  beta1_CI = confint(lm.sim, "x",level = 0.95)
  CI$lower[i] = beta1_CI[1] #lower bound
  CI$upper[i] = beta1_CI[2] #upper bound
  CI$contain_true_value[i] = (beta1_CI[1] <= beta1 & beta1_CI[2] >= beta1) # indicator, if the true value is wit
hin the confidence interval, return TRUE
}
```

```
cols = c('TRUE' = 'darkgreen','FALSE' = 'darkred')
size = c('TRUE' = 0.5,'FALSE' = 1)
ggplot(data = CI) +
  geom_errorbar(aes(x = 1:B,y = beta1,ymin=lower, ymax=upper,color = contain_true_value,size = contain_true_valu
e)) +
  geom_hline(yintercept = beta1) +
  scale_color_manual(values = cols) +
  scale_size_manual(values = size) +
  xlab('Number of experiments')
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
```



The green bars are the intervals that contain the true value. As we can see, sometimes, the confidence interval does not contain the true value, which is 3.

Validate coverage probability:

```
sum(CI$contain_true_value)/B
```

```
## [1] 0.958
```

This estimated coverage probability will coverge to 0.95 as $B \to \infty$.

What is a confidence band? Illustrate the confidence band using the 04cars dataset.

Write a function confBand(x, y, conf=0.95) taking in a predictor vector (x1; ...; xn) and a response vector y = (y1; ... ; yn) and return a plot with the points (x1; y1); ... ; (xn; yn), the least squares line, and the confidence band at level conf. Apply the function to hp and mpg from the 04cars dataset.

- Create the confBand function

```
confBand =  function(x, y, conf = 0.95) {
    fit = lm(y ~ x)
    plot(x, y, pch = 16)
    newx = seq(min(x), max(x), len = 100)
    model = predict(fit, newdata = data.frame(x = newx),
                    se.fit = TRUE) # predicted values and standard error of predicted means
    pred = model$fit
    pred.se = model$se.fit

    n = length(x)
    lower = pred - sqrt(2 * qf(conf, 2, n - 2)) * pred.se # lower bound of the confidence band
    upper = pred + sqrt(2 * qf(conf, 2, n - 2)) * pred.se # upper bound of the confidence band

    polygon(c(rev(newx), newx), c(rev(upper), lower),
            col = rgb(0.75, 0.75, 0.75, 0.5), border = NA) # draw the confidence band
    lines(newx, upper, lty = 'dashed', col = 'blue')
    lines(newx, lower, lty = 'dashed', col = 'blue')
    lines(newx, pred, col = "red", lwd = 2)
}
```

- Apply the confBand function to `hp` and `mpg` from the `04cars.rda` dataset

```
load("datasets/04cars.rda")
dat = dat[, c(13,15)] # extract target columns hp and mpg
dat = dat[complete.cases(dat), ]
names(dat) = c("hp", "mpg")
with(dat, confBand(hp, mpg)) # call the function confBand in the data environment
```