1. Continuing lecture 16, if $f$ is not normal, $T$ will be still a pivot but we do not know its distribution. So we generate 1000 bootstrap replicates of $T^* = \frac{\bar{X} - \bar{X}}{S^* / \sqrt{n}}$ of $T$ and get the bootstrap CI

$$t = \frac{\bar{x} - t_{1-\alpha/2} S / \sqrt{n}}{\bar{x} - t_{\alpha/2} S / \sqrt{n}}.$$

For $i = 1:1000$:
- $trep(i) = t(boot(arm,1), mean(arm))$
- $\text{quantile}(trep, [0.05 0.95]) = -19.8246, 22.0677$

2. For given bivariate data $(x_i, y_i), i = 1, \ldots, n$ suppose we have model

$$Y_i = g(x_i; \beta) + \epsilon_i.$$

Let $g(x; \beta) = \beta_0 + \beta_1 x$. Fitting a somewhat crazy model like $g(x; \beta) = \frac{\beta_0}{\beta_1 - \beta_2 x - \beta_3 x^2}$ is also possible. The least squares estimation (LSE) is given by

$$\hat{\beta} = \text{arg min}_{\beta} \sum_{i=1}^{n} (y_i - g(x_i; \beta))^2.$$

Based on LSE, the observed residual is

$$\hat{\epsilon}_i = y_i - g(x_i; \hat{\beta}).$$

A bootstrap dataset is then

$$Y_i^* = g(x_i; \hat{\beta}) + \epsilon_i^*.$$

The bootstrap estimate of $\beta$ is then

$$\hat{\beta}^* = \text{arg min}_{\beta} \sum_{i=1}^{n} (y_i^* - g(x_i; \beta))^2.$$

3. Let’s find CI for $\beta_1$ for model

$$grip = \beta_0 + \beta_1 \text{arm}.$$

The pivot for $\beta_1$ is

$$T = \frac{\hat{\beta}_1 - \beta_1}{SE\hat{\beta}_1} = \frac{\sqrt{\text{Var}\hat{\beta}_1}}{SE\hat{\beta}_1}.$$ If you don’t know the standard error $SE$, we estimate it, i.e. $T = (\hat{\beta}_1 - \beta_1)/SE\hat{\beta}_1$. Following eq.1,2 and 3 we generate 1000 bootstrap replicates of $T^* = \frac{\hat{\beta}_1^* - \beta_1}{SE\hat{\beta}_1^*}$. We need to estimate $SE\hat{\beta}_1^*$ via bootstrap which requires bootstrapping from bootstrapped residual $\epsilon_1^*$ (there is a formula for computing the standard error but well let’s forget that). Then 100(1 - $\alpha$)% CI for $\beta_1$ is

$$(\hat{\beta}_1 - t_{1-\alpha/2} S_{\beta_1}, \hat{\beta}_1 - t_{\alpha/2} S_{\beta_1}).$$

```matlab
beta=inline('pinv([ones(147,1) x])*y')
b=beta(arm,grip)
b=
54.7081
0.7050
e=grip-b(1)-b(2)*arm;
for j=1:1000
    bse=boot(e,1);
    bsgrip = bse + b(1) + b(2)*arm;
    bsb=beta(arm,bsgrip);
    for i=1:50
        bs2e=boot(bse,1);
        bs2grip = bs2e + b(1) + b(2)*arm;
        bs2b=beta(arm,bs2grip);
        bsbeta(i)=bs2b(2);
    end;
    trep(j)=(bsb(2)-0.7050)/std(bsbeta);end;
>>quantile(trep,[0.05 0.95])
-1.5308  1.6580
```