Computing OLS, CIs, Hypothesis Testing, Plots

PSTAT 126

Lab 2

```r
library(tidyverse) # Easily Install and Load the 'Tidyverse'
library(palmerpenguins) # Palmer Archipelago (Antarctica) Penguin Data
```

Contents

Computing OLS estimators

Dataset: Adelie and Gentoo Penguins

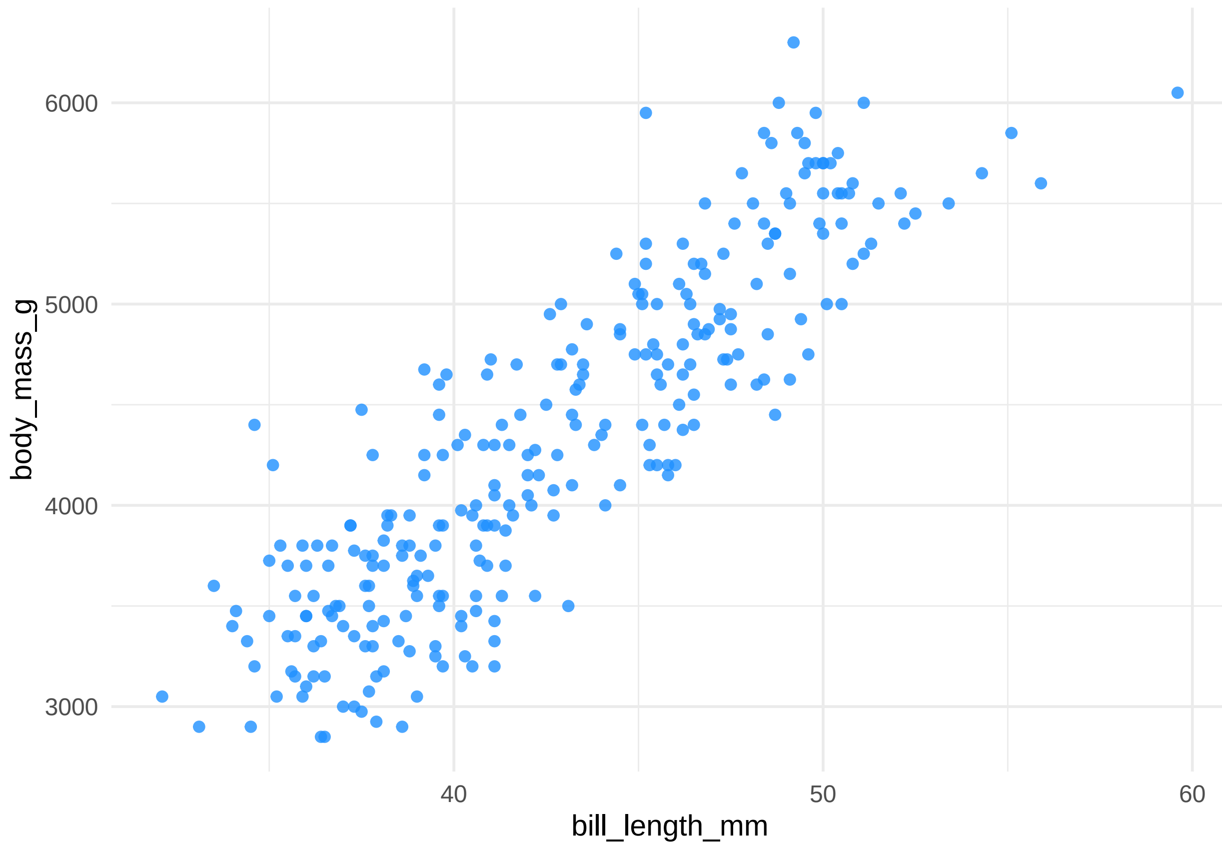
- **Question: Can we predict body mass in grams by a penguins bill length in mm?**

```r
data("penguins")

penguins_noChinstrap <- penguins %>%
  filter(species != "Chinstrap") %>%
  drop_na(bill_length_mm, body_mass_g)

summary(penguins_noChinstrap)
```

```
##       species          island    bill_length_mm  bill_depth_mm
##  Adelie   :151   Biscoe   :167   Min.   :32.10   Min.   :13.10
##  Chinstrap:  0   Dream    : 56   1st Qu.:38.35   1st Qu.:15.00
##  Gentoo   :123   Torgersen: 51   Median :42.00   Median :17.00
##                                  Mean   :42.70   Mean   :16.84
##                                  3rd Qu.:46.67   3rd Qu.:18.50
##                                  Max.   :59.60   Max.   :21.50
##  flipper_length_mm  body_mass_g       sex           year
##  Min.   :172.0     Min.   :2850   female:131   Min.   :2007
##  1st Qu.:190.0     1st Qu.:3600   male  :134   1st Qu.:2007
##  Median :198.0     Median :4262   NA's  :  9   Median :2008
##  Mean   :202.2     Mean   :4318                Mean   :2008
##  3rd Qu.:215.0     3rd Qu.:4950                3rd Qu.:2009
##  Max.   :231.0     Max.   :6300                Max.   :2009
```

```r
# plot of data
ggplot(data = penguins_noChinstrap,
```

```r
    aes(x = bill_length_mm, y = body_mass_g)) +
  geom_point(color = "dodgerblue", alpha = 0.8, size = 1.5) +
  theme_minimal()
```




```r
x <- penguins_noChinstrap$bill_length_mm
y <- penguins_noChinstrap$body_mass_g
```

First obtain means of x and y

```r
x_bar <- mean(x)
y_bar <- mean(y)
```

$$S_{xx} : \Sigma_{i=1}^{n}(x_i - \bar{x})^2$$

```r
Sxx <- sum((x - x_bar)^2)
Sxx
```

```
## [1] 7369.338
```

$$S_{yy} : \Sigma_{i=1}^{n}(y_i - \bar{y})^2$$

```r
Syy <- sum((y - y_bar)^2)
Syy
```

```
## [1] 190768075
```

$$S_{xy} : \Sigma_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$$

```r
Sxy <- sum((x - x_bar)*(y - y_bar))
Sxy
```

```
## [1] 1039728
```

$$\hat{\beta}_1 = S_{xy}/S_{xx}$$

```r
b1 <- Sxy / Sxx
b1
```

```
## [1] 141.0884
```

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x}$$

```r
b0 <- y_bar - b1*x_bar
b0
```

```
## [1] -1706.821
```

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

```r
y_hat <- b0 + b1*x
```

Estimation of Residuals

$$e_i = y_i - \hat{y}$$

```r
e <- y - y_hat
```

$$\hat{\sigma}^2 = \frac{1}{n-2}\Sigma_{i=1}^{n} e_i^2$$

```r
n <- length(y)
sigma_2_hat <- sum(e^2) / (n-2)
sigma_2_hat
```

```
## [1] 162038.6
```

```r
sqrt(sigma_2_hat) # Residual Standard Error (RSE)
```

```
## [1] 402.5402
```

The lm() function

```r
model <- lm(body_mass_g ~ bill_length_mm , data = penguins_noChinstrap)
summary(model)
```

```
##
## Call:
## lm(formula = body_mass_g ~ bill_length_mm, data = penguins_noChinstrap)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -891.91 -272.91   -0.82  282.47 1279.63
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1706.821    201.712  -8.462 1.65e-15 ***
## bill_length_mm   141.088      4.689  30.088  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 402.5 on 272 degrees of freedom
## Multiple R-squared:  0.769,  Adjusted R-squared:  0.7681
## F-statistic: 905.3 on 1 and 272 DF,  p-value: < 2.2e-16
```

```r
coef(model) # estimates for beta0 and beta1
```

```
##    (Intercept) bill_length_mm
##     -1706.8209       141.0884
```

```r
model$coefficients
```

```
##    (Intercept) bill_length_mm
##     -1706.8209       141.0884
```

```r
head(model$residuals) # residuals
```

```
##          1          2          3          4          5          6
##  -59.73552  -66.17088 -729.04160  -21.12337 -187.95320 -156.51784
```

```r
head(model$fitted.values) # y_hat values
```

```
##        1        2        3        4        5        6
## 3809.736 3866.171 3979.042 3471.123 3837.953 3781.518
```

```r
summary(model$residuals) # first line in summary output.
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -891.9123 -272.9122   -0.8239    0.0000  282.4722 1279.6252
```

$\hat{se}(\hat{\beta}_1) = \sqrt{\frac{\hat{\sigma}^2}{S_{xx}}}$ $\hat{se}(\hat{\beta}_0) = \sqrt{\hat{\sigma}^2(\frac{1}{n} + \frac{\bar{x}^2}{S_{xx}})}$

```r
# Standard errors
summary(model)$coef
```

```
##                 Estimate Std. Error   t value     Pr(>|t|)
## (Intercept)    -1706.8209  201.71210 -8.461668 1.648813e-15
## bill_length_mm   141.0884    4.68916 30.088202 1.590571e-88
```

```r
coef(summary(model))[, "Std. Error"]
```

```
##    (Intercept) bill_length_mm
##      201.71210        4.68916
```

```r
sqrt(sigma_2_hat/Sxx)
```

```
## [1] 4.68916
```

```r
sqrt(sigma_2_hat*(1/n + (x_bar^2)/Sxx))
```

```
## [1] 201.7121
```

```r
# p-values for intercept and slope
summary(model)$coef[,4]
```

```
##    (Intercept) bill_length_mm
##   1.648813e-15   1.590571e-88
```

p-values for t-test and F-test in simple linear regression are identical.

```r
summary(model)$sigma
```

```
## [1] 402.5402
```

Confidence Intervals for intercept and slope estimates

Can calculate a 90% confidence interval by entering values into formula:

- **Intercept**

$$\hat{\beta}_0 \pm (t_{\alpha/2,n-2} se(\hat{\beta}_0))$$

- **Slope**

$$\hat{\beta}_1 \pm (t_{\alpha/2,n-2} se(\hat{\beta}_1))$$

```r
se_b0 <- sqrt(sigma_2_hat*(1/n + (x_bar^2)/Sxx)) # se of intercept
se_b1 <- sqrt(sigma_2_hat/Sxx) # se of slope
t <- qt(p = 0.95, df = n - 2) # t-statistic
```

```r
CI_b0_90 <-  c(b0 - t*se_b0, b0 + t*se_b0) # 90% CI for b0
CI_b1_90 <-  c(b1 - t*se_b1, b1 + t*se_b1) # 90% CI for b1
CI_b0_90
```

```
## [1] -2039.742 -1373.900
```

```r
CI_b1_90
```

```
## [1] 133.3491 148.8277
```

Can also use the `confint` function

```r
#?confint
confint(model, level = 0.95) # 95% CI
```

```
##                     2.5 %     97.5 %
## (Intercept)    -2103.9363 -1309.7054
## bill_length_mm   131.8567   150.3201
```

```r
confint(model, level = 0.90) # 90% CI
```

```
##                       5 %       95 %
## (Intercept)    -2039.7416 -1373.9001
## bill_length_mm   133.3491   148.8277
```

Hypothesis Testing

Hypothesis testing of $\hat{\beta}_0, \hat{\beta}_1$

Want to test:

$H_0 : \hat{\beta}_0 = 0$ vs. $H_1 : \hat{\beta}_0 \neq 0$
$H_0 : \hat{\beta}_1 = 0$ vs. $H_1 : \hat{\beta}_1 \neq 0$
Let $\alpha = 0.05$

```
t_b0 <- (b0-0)/se_b0
t_b1 <- (b1-0)/se_b1
t_b0
```

```
## [1] -8.461668
```

```
t_b1
```

```
## [1] 30.0882
```

- For distributions in R, p stands for "probability", the cumulative distribution function (c.d.f.).

```
p0 <- 2*(1 - pt(abs(t_b0), df = n-2))
p1 <- 2*(1 - pt(abs(t_b1), df = n-2))

p0
```

```
## [1] 1.776357e-15
```

```
p1
```

```
## [1] 0
```

Reject null hypothesis for both $\hat{\beta}_0, \hat{\beta}_1$

Coefficient of Determination R^2

- A goodness-of-fit measure

$$R^2 = 1 - \frac{RSS}{S_{yy}}$$

$$R^2_{adj} = 1 - \frac{RSS/df}{S_{yy}/(n-1)}$$

```
b0 <- summary(model)$coef[1,1] # Intercept
b1 <- summary(model)$coef[2,1] # Slope
y_hat <- b0 + b1*x # Fitted values
e <- y - y_hat # Residuals

Syy <- sum((y - y_bar)^2)

r_2 <- 1 - (sum(e^2)/Syy)
r_2
```

```
## [1] 0.7689629
```

```
summary(model)$r.squared
```

```
## [1] 0.7689629
```

```
r <- cor(x,y)
r^2
```

```
## [1] 0.7689629
```

```r
adj_r2 <- 1 - (sum(e^2)/(n-2))/(Syy/(n-1))
adj_r2
```

```
## [1] 0.7681135
```

```r
summary(model)$adj.r.squared
```

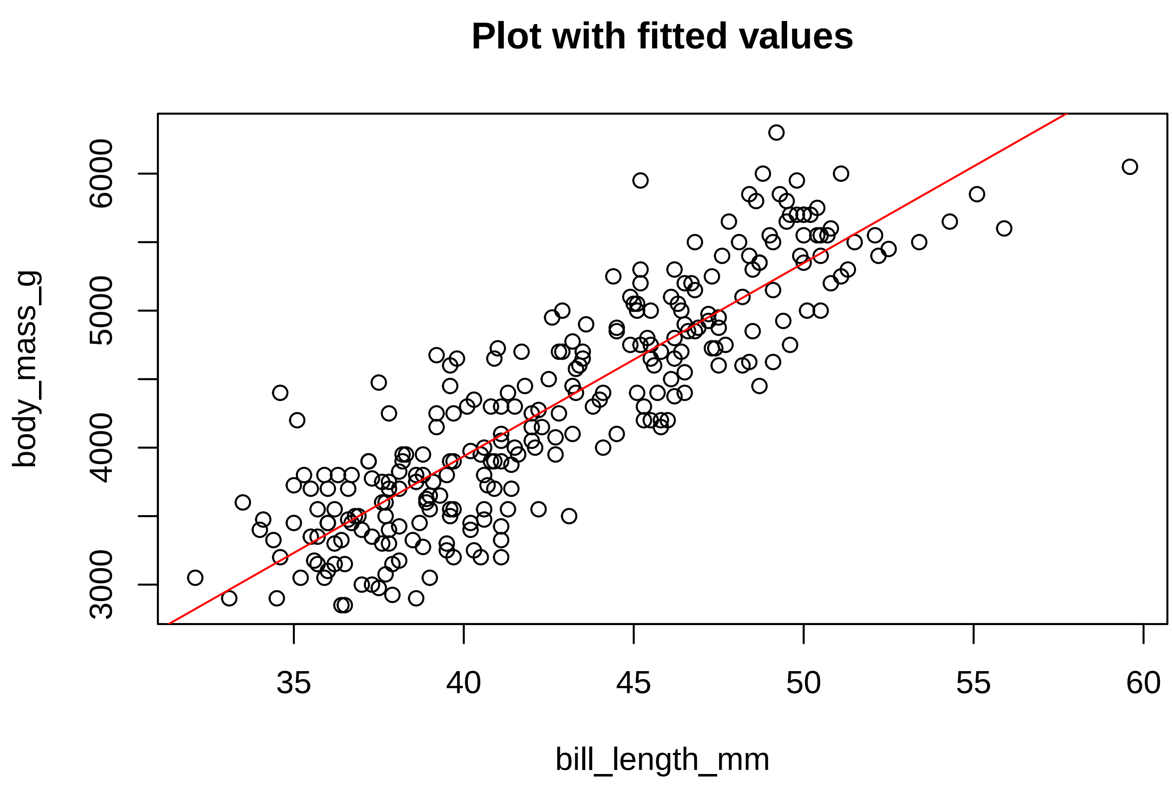
```
## [1] 0.7681135
```

Notes on R^2

- Always between 0 and 1
- Can interpret as $R^2 \times 100$ percent of the variation in Y is explained by the variation in the predictor x.

Plots

```r
plot(body_mass_g ~ bill_length_mm , data = penguins_noChinstrap,
     main = "Plot with fitted values")
abline(model, col = "Red")
```




```r
ggplot(data = penguins_noChinstrap) +
  geom_point(aes(x = bill_length_mm, y = body_mass_g), color = "dodgerblue", alpha = 0.95) +
  geom_abline(aes(intercept = model$coefficients[[1]],
                  slope = model$coefficients[[2]]),
                  color = "red") +
  labs(x = "bill length (mm)",
       y = "body mass (grams)",
       title = "Plot with fitted values")
```

Plot with fitted values