PSTAT 126

Lab 3

Roupen Khanjian

Spring 2021

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

Contents

Dataset: Adelie and Gentoo Penguins, with the same question as last section:

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

```
data("penguins")

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

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
```

Simple Linear Regression Model Assumptions

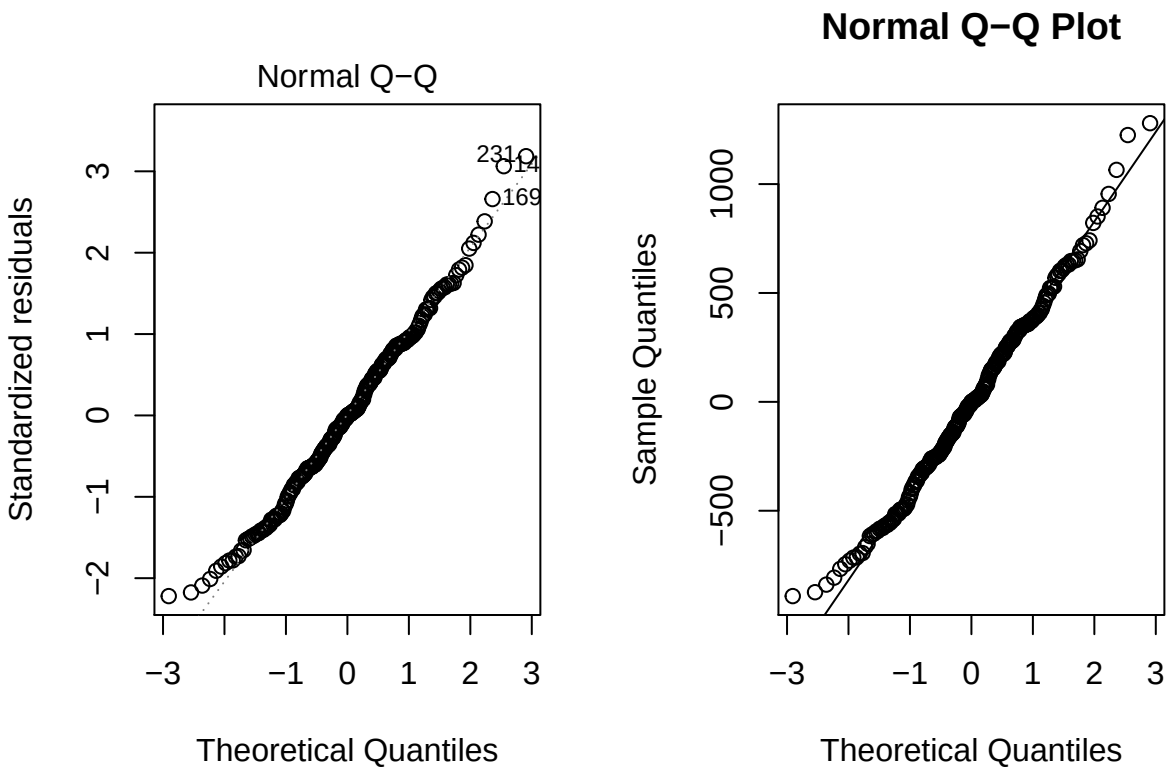
1) The relationship between each Y_n and each x_n, respectively, is linear. **L**inearity
2) Errors have **E**qual variance. $\text{Var}(Y_n) = \sigma^2$ for every n (homoscedasticity)
3) Errors are **N**ormally distributed
4) Errors are **I**ndependent

- Can use the acronym **L.I.N.E.** to help you remember.

Graphically checking the normality assumption

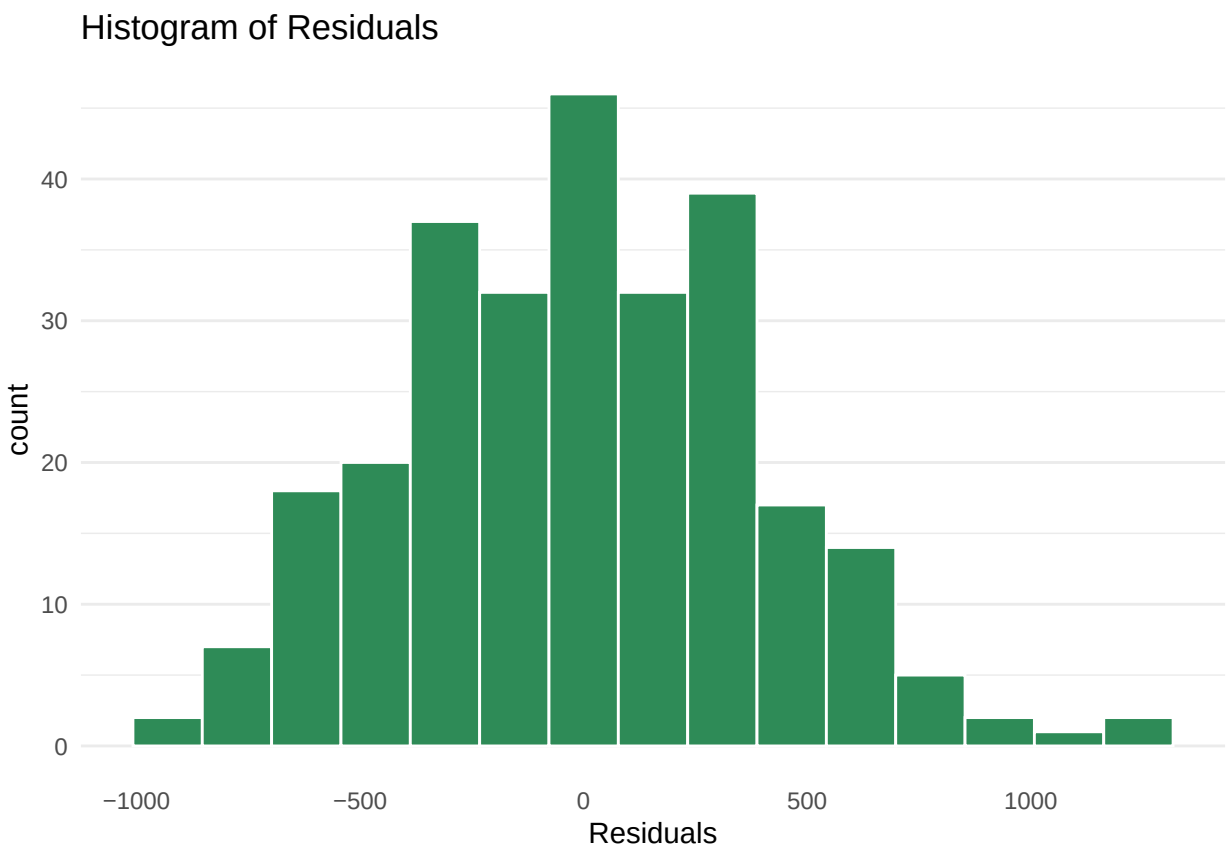
QQ - plot

```
par(mfrow = c(1, 2))

plot(model, which = 2) # QQ

e <- residuals(model) # Residuals

qqnorm(e) # QQ
qqline(e)
```



Histogram of residuals

```
par(mfrow = c(1, 1))
resid_model <- tibble(residuals = residuals(model))

ggplot(data = resid_model,
       aes(x = residuals)) +
  geom_histogram(color = "white",
                 fill = "seagreen",
                 bins = 15) +
  labs(x = "Residuals",
       title = "Histogram of Residuals") +
  theme_minimal() +
  theme(panel.grid.major.x = element_blank(),
        panel.grid.minor.x = element_blank())
```



Confidence interval for new observations

- Here we use $x_0 = 50$ bill length (mm)

95% Confidence Interval for new observation

```
n <- nrow(penguins_noChinstrap) # number of observations
x <- penguins_noChinstrap$bill_length_mm # predictor variable
y <- penguins_noChinstrap$body_mass_g # response variable
x_bar <- mean(x) # mean of bill_length_mm
y_bar <- mean(y) # mean of body_mass_g
Sxx <- sum((x - x_bar) ^ 2)
sigma_hat <- summary(model)$sigma # Residual Standard Error (RSE)
Yhat_50 <- # predicated body mass when bill length is 50 mm
  as.numeric(coef(model)[1] + coef(model)[2] * 50)
y_hat <- fitted(model) # fitted values

spe_50 <- sigma_hat*sqrt(1 + 1/n + (50 - x_bar)^2/Sxx) # se of y_hat(x_0)
t_pct <- qt(p = 0.975, df = n - 2) # t-statistic
CI_95 <- c(Yhat_50 - spe_50*t_pct, Yhat_50 + spe_50*t_pct)
CI_95
```

```
## [1] 4550.811 6144.387
```

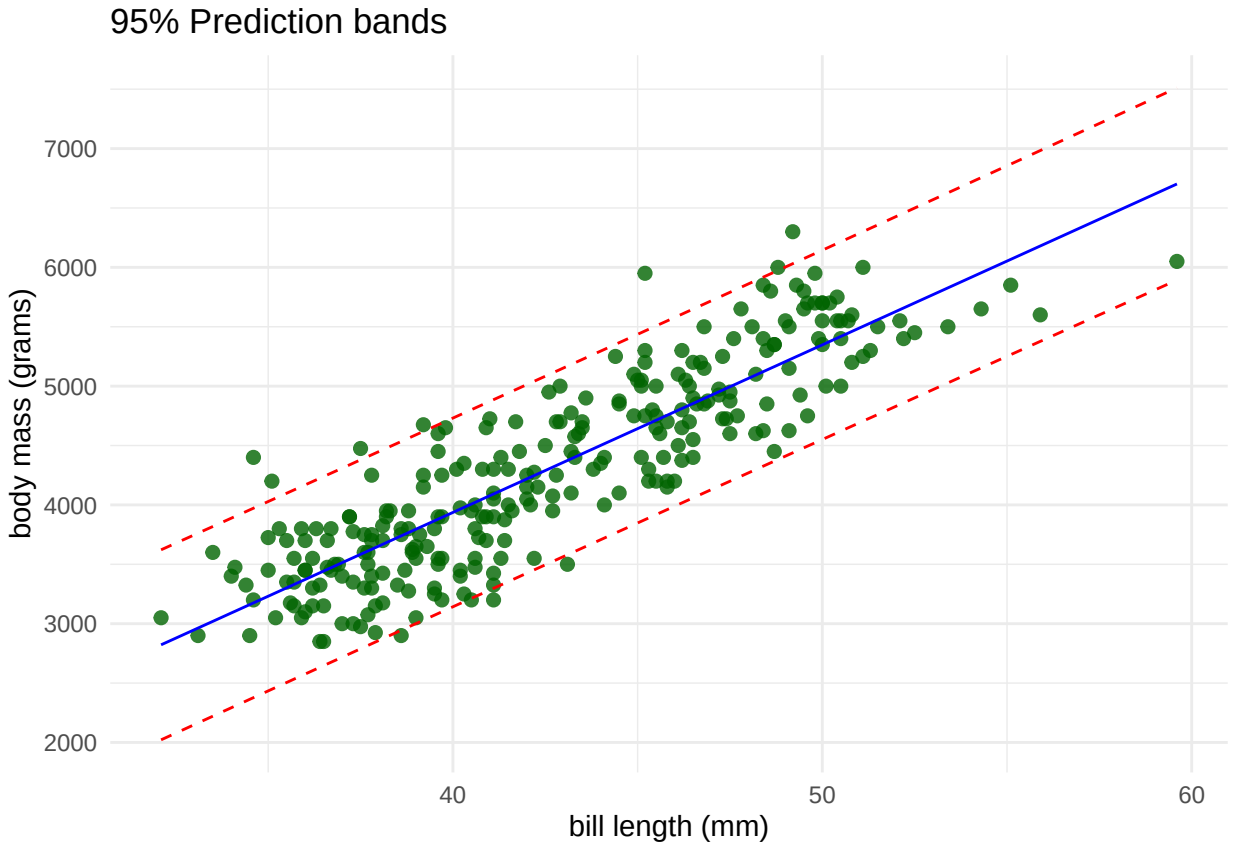
```
predict(model, newdata = data.frame(bill_length_mm = 50),
        level = 0.95, interval = 'prediction')
```

```
##        fit      lwr      upr
## 1 5347.599 4550.811 6144.387
```

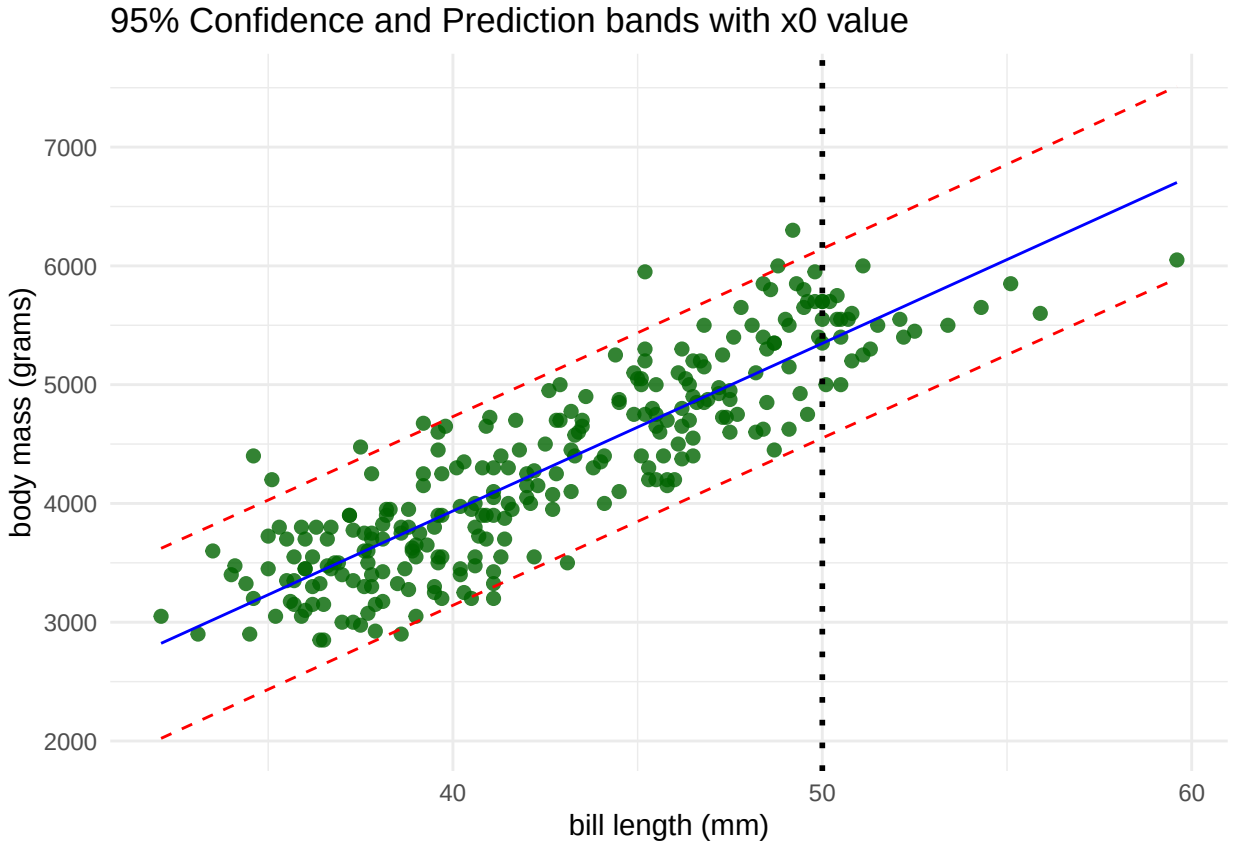
Visualizing confidence interval bands

```
ngrid <-  274
grid <-  seq(min(x), max(x), length = ngrid)
new <- data.frame(bill_length_mm = grid)
p2 <- predict(model, new, se.fit = TRUE, interval = "prediction", level = 0.95)

# tibble for plot
conf_pred_tib <- tibble(x,y, y_hat, new = new$bill_length_mm,
                        UL_p = p2$fit[,3], LL_p = p2$fit[,2])
```

```r
# Plot
ggplot(data = conf_pred_tib) +
  geom_point(aes(x = x, y = y), color = "darkgreen", alpha = 0.8, size = 2) + # data points
  geom_line(aes(x = x, y = y_hat), color = "blue") + # Fitted line
  geom_line(aes(x = new, y = UL_p), color = "red", linetype = "dashed") + # upper bound
  geom_line(aes(x = new, y = LL_p), color = "red", linetype = "dashed")+ # lower bound
  scale_x_continuous(breaks = seq(30, 60, by = 10)) +
  scale_y_continuous(breaks = seq(2000, 7000, by = 1000)) +
  labs(x = "bill length (mm)",
       y = "body mass (grams)",
       title = "95% Prediction bands") +
  theme_minimal() +
  theme()
```



```
ggplot(data = conf_pred_tib) +
  geom_point(aes(x = x, y = y), color = "darkgreen", alpha = 0.8, size = 2) + # data points
  geom_line(aes(x = x, y = y_hat), color = "blue") + # Fitted line
  geom_line(aes(x = new, y = UL_p), color = "red", linetype = "dashed") + # upper bound
  geom_line(aes(x = new, y = LL_p), color = "red", linetype = "dashed")+ # lower bound
  geom_vline(aes(xintercept = 50), size = 1, linetype = "dotted") + # x0 value
  scale_x_continuous(breaks = seq(30, 60, by = 10)) +
  scale_y_continuous(breaks = seq(2000, 7000, by = 1000)) +
  labs(x = "bill length (mm)",
       y = "body mass (grams)",
       title = "95% Confidence and Prediction bands with x0 value") +
  theme_minimal() +
  theme()
```




```
predict(model, newdata = data.frame(bill_length_mm = 50),
        level = 0.95, interval = 'predict')
```

```
##        fit      lwr      upr
## 1 5347.599 4550.811 6144.387
```

Coefficient of Determination R^2

- A goodness-of-fit measure

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

```
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
```

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.