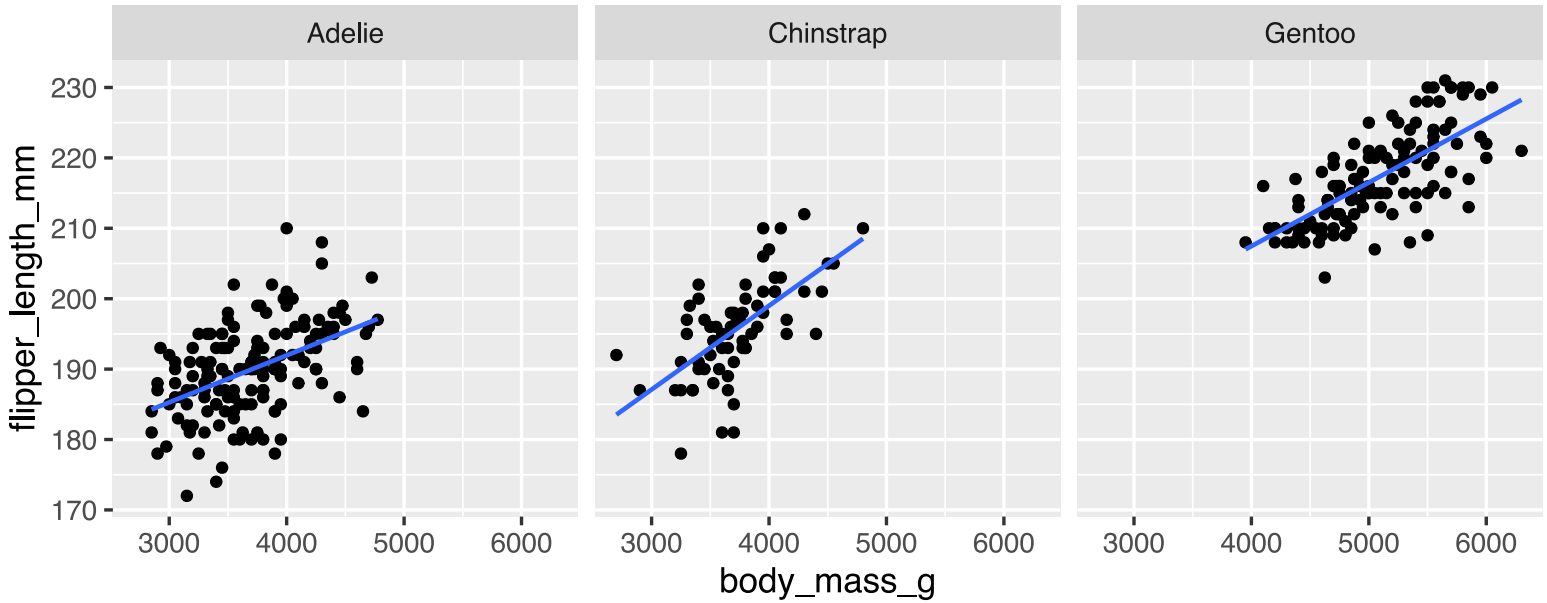
Working with models

Claus O. Wilke

last updated: 2021-03-19

How do we obtain information about model fits?

```
penguins %>%
  ggplot(aes(body_mass_g, flipper_length_mm)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  facet_wrap(vars(species))
```



We can fit a linear model with `lm( )`

```
penguins_adelie <- filter(penguins, species == "Adelie")

lm_out <- lm(flipper_length_mm ~ body_mass_g, data = penguins_adelie)
summary(lm_out)
```

against formula

y

predictor

```
Call:
lm(formula = flipper_length_mm ~ body_mass_g, data = penguins_adelie)

Residuals:
     Min       1Q   Median       3Q      Max
-14.2769  -3.6192   0.0569   3.4696  18.0477

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.652e+02  3.849e+00  42.929  < 2e-16 ***
body_mass_g 6.677e-03  1.032e-03   6.468 1.34e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.798 on 149 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.2192,    Adjusted R-squared:  0.214
F-statistic: 41.83 on 1 and 149 DF,  p-value: 1.343e-09
```

Use `map()` to fit models to groups of data

```
penguins %>%
  nest(data = -species) # nest all data except species
```

```
# A tibble: 3 x 2
  species   data
  <fct>     <list>
1 Adelie    <tibble [152 × 7]>
2 Gentoo    <tibble [124 × 7]>
3 Chinstrap <tibble [68 × 7]>
```

raw. column

Use `map()` to fit models to groups of data

```
penguins %>%
  nest(data = -species) %>%
  mutate(
    # apply linear model to each nested data frame
    fit = map(data, ~lm(flipper_length_mm ~ body_mass_g
  )
```

```
# A tibble: 3 x 3
  species   data               fit
  <fct>     <list>             <list>
1 Adelie    <tibble [152 × 7]> <lm>
2 Gentoo    <tibble [124 × 7]> <lm>
3 Chinstrap <tibble [68 × 7]>  <lm>
```

Use `map()` to fit models to groups of data

```
lm_data <- penguins %>%
  nest(data = -species) %>%
  mutate(
    # apply linear model to each nested data frame
    fit = map(data, ~lm(flipper_length_mm ~ body_mass_g
  )

lm_data$fit[[1]]  # first model fit, for Adelie species
```

```
Call:
lm(formula = flipper_length_mm ~ body_mass_g, data = .x)

Coefficients:
(Intercept)  body_mass_g
  1.652e+02    6.677e-03
```

```r
summary(lm_data$fit[[1]]) # summarize the first model, which is for Ade
```

```
Call:
lm(formula = flipper_length_mm ~ body_mass_g, data = .x)

Residuals:
     Min       1Q   Median       3Q      Max
-14.2769  -3.6192   0.0569   3.4696  18.0477

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.652e+02  3.849e+00  42.929  < 2e-16 ***
body_mass_g 6.677e-03  1.032e-03   6.468 1.34e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.798 on 149 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.2192,    Adjusted R-squared:  0.214
F-statistic: 41.83 on 1 and 149 DF,  p-value: 1.343e-09
```

```
summary(lm_out)
```

```
Call:
lm(formula = flipper_length_mm ~ body_mass_g, data = penguins_adelie)

Residuals:
     Min       1Q   Median       3Q      Max
-14.2769  -3.6192   0.0569   3.4696  18.0477

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.652e+02  3.849e+00  42.929  < 2e-16 ***
body_mass_g 6.677e-03  1.032e-03   6.468 1.34e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.798 on 149 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.2192,    Adjusted R-squared:  0.214
F-statistic: 41.83 on 1 and 149 DF,  p-value: 1.343e-09
```

```r
summary(lm_data$fit[[1]]) # summarize the first model, which is for Ade
```

```
Call:
lm(formula = flipper_length_mm ~ body_mass_g, data = .x)

Residuals:
     Min       1Q   Median       3Q      Max
-14.2769  -3.6192   0.0569   3.4696  18.0477

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.652e+02  3.849e+00  42.929  < 2e-16 ***
body_mass_g 6.677e-03  1.032e-03   6.468 1.34e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.798 on 149 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.2192,    Adjusted R-squared:  0.214
F-statistic: 41.83 on 1 and 149 DF,  p-value: 1.343e-09
```

```r
summary(lm_data$fit[[2]]) # second model, Chinstrap
```

```
Call:
lm(formula = flipper_length_mm ~ body_mass_g, data = .x)

Residuals:
     Min       1Q   Median       3Q      Max 
-12.0194  -2.7401   0.1781   2.9859   8.9806 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1.713e+02  4.244e+00   40.36   <2e-16 ***
body_mass_g 9.039e-03  8.321e-04   10.86   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.633 on 121 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.4937,    Adjusted R-squared:  0.4896 
F-statistic:   118 on 1 and 121 DF,  p-value: < 2.2e-16
```

```r
summary(lm_data$fit[[3]]) # third model, Gentoo
```

```
Call:
lm(formula = flipper_length_mm ~ body_mass_g, data = .x)

Residuals:
     Min       1Q   Median       3Q      Max 
-14.4296  -3.3315   0.4097   2.8889  11.5941 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1.514e+02  6.575e+00  23.024  < 2e-16 ***
body_mass_g 1.191e-02  1.752e-03   6.795 3.75e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.512 on 66 degrees of freedom
Multiple R-squared:  0.4116,    Adjusted R-squared:  0.4027 
F-statistic: 46.17 on 1 and 66 DF,  p-value: 3.748e-09
```

How do we get this information into a data table?

The **broom** package cleans model output: `glance()`

`glance()` provides model-wide summary estimates in tidy format

```
library(broom)

glance(lm_out)
```

```
# A tibble: 1 x 12
  r.squared adj.r.squared sigma statistic p.value    df logLik
      <dbl>         <dbl> <dbl>     <dbl>   <dbl> <dbl>  <dbl> <
1     0.219         0.214  5.80      41.8 1.34e-9     1  -479.
# … with 3 more variables: deviance <dbl>, df.residual <int>, nc
```

The **broom** package cleans model output: `tidy()`

`tidy()` provides information about regression coefficients in tidy format

```
library(broom)

tidy(lm_out)
```

```
# A tibble: 2 x 5
  term         estimate std.error statistic  p.value
  <chr>           <dbl>     <dbl>     <dbl>    <dbl>
1 (Intercept) 165.        3.85        42.9  8.68e-86
2 body_mass_g   0.00668   0.00103      6.47 1.34e- 9
```

Apply these functions to multiple models with `map()`

Reminder: This is the nested table with fitted models

```
lm_data
```

```
# A tibble: 3 x 3
  species   data               fit
  <fct>     <list>             <list>
1 Adelie    <tibble [152 × 7]> <lm>
2 Gentoo    <tibble [124 × 7]> <lm>
3 Chinstrap <tibble [68 × 7]>  <lm>
```

Apply these functions to multiple models with `map()`

```
lm_data %>%
  mutate(
    glance_out = map(fit, glance)
  )
```

```
# A tibble: 3 x 4
  species   data               fit    glance_out
  <fct>     <list>             <list> <list>
1 Adelie    <tibble [152 × 7]> <lm>   <tibble [1 × 12]>
2 Gentoo    <tibble [124 × 7]> <lm>   <tibble [1 × 12]>
3 Chinstrap <tibble [68 × 7]>  <lm>   <tibble [1 × 12]>
```

Apply these functions to multiple models with `map()`

```
lm_data %>%
  mutate(
    glance_out = map(fit, ~glance(.x)) # same as just `
  )
```

```
# A tibble: 3 x 4
  species   data               fit    glance_out
  <fct>     <list>             <list> <list>
1 Adelie    <tibble [152 × 7]> <lm>   <tibble [1 × 12]>
2 Gentoo    <tibble [124 × 7]> <lm>   <tibble [1 × 12]>
3 Chinstrap <tibble [68 × 7]>  <lm>   <tibble [1 × 12]>
```

Apply these functions to multiple models with `map()`

```
lm_data %>%
  mutate(
    glance_out = map(fit, glance)
  )
```

```
# A tibble: 3 x 4
  species   data               fit    glance_out
  <fct>     <list>             <list> <list>
1 Adelie    <tibble [152 × 7]> <lm>   <tibble [1 × 12]>
2 Gentoo    <tibble [124 × 7]> <lm>   <tibble [1 × 12]>
3 Chinstrap <tibble [68 × 7]>  <lm>   <tibble [1 × 12]>
```

And unnest

```
lm_data %>%
  mutate(
    glance_out = map(fit, glance)
  ) %>%
  select(species, glance_out)
```

```
# A tibble: 3 x 2
  species   glance_out
  <fct>     <list>
1 Adelie    <tibble [1 × 12]>
2 Gentoo    <tibble [1 × 12]>
3 Chinstrap <tibble [1 × 12]>
```

And unnest

```
lm_data %>%
  mutate(
    glance_out = map(fit, glance)
  ) %>%
  select(species, glance_out) %>%
  unnest(cols = glance_out)
```

```
# A tibble: 3 x 13
  species r.squared adj.r.squared sigma statistic  p.value    df
  <fct>       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>
1 Adelie      0.219         0.214  5.80      41.8 1.34e- 9     1
2 Gentoo      0.494         0.490  4.63     118.  1.33e-19     1
3 Chinst…     0.412         0.403  5.51      46.2 3.75e- 9     1
# … with 4 more variables: BIC <dbl>, deviance <dbl>, df.residua
#   nobs <int>
```

All in one pipeline

```
lm_summary <- penguins %>%
  nest(data = -species) %>%
  mutate(
    fit = map(data, ~lm(flipper_length_mm ~ body_mass_g, data = .x)),
    glance_out = map(fit, glance)
  ) %>%
  select(species, glance_out) %>%
  unnest(cols = glance_out)

lm_summary
```

```
# A tibble: 3 x 13
  species r.squared adj.r.squared sigma statistic  p.value    df logLik   AIC
  <fct>       <dbl>         <dbl> <dbl>     <dbl>    <dbl> <dbl>  <dbl> <dbl>
1 Adelie      0.219         0.214  5.80      41.8 1.34e- 9     1  -479.  963.
2 Gentoo      0.494         0.490  4.63     118.  1.33e-19     1  -362.  730.
3 Chinst…     0.412         0.403  5.51      46.2 3.75e- 9     1  -212.  429.
# … with 4 more variables: BIC <dbl>, deviance <dbl>, df.residual <int>,
#   nobs <int>
```

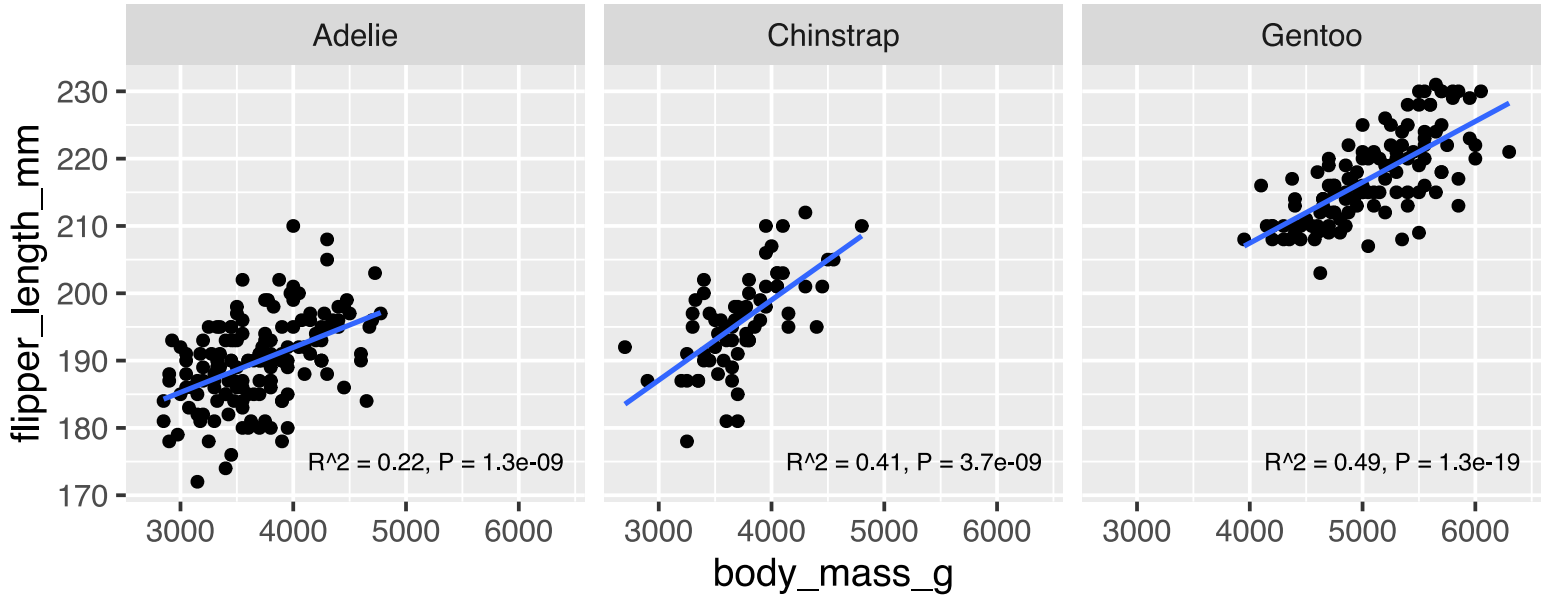
Make label data

```
library(glue)  # for easy text formatting

label_data <- lm_summary %>%
  mutate(
    rsqr = signif(r.squared, 2),  # round to 2 significant digits
    pval = signif(p.value, 2),
    label = glue("R^2 = {rsqr}, P = {pval}"),
    body_mass_g = 6400, flipper_length_mm = 175 # label position in plo
  ) %>%
  select(species, label, body_mass_g, flipper_length_mm)

label_data
```

```
# A tibble: 3 x 4
  species   label                    body_mass_g flipper_length_mm
  <fct>     <glue>                         <dbl>             <dbl>
1 Adelie    R^2 = 0.22, P = 1.3e-09        6400               175
2 Gentoo    R^2 = 0.49, P = 1.3e-19        6400               175
3 Chinstrap R^2 = 0.41, P = 3.7e-09        6400               175
```

And plot

```
ggplot(penguins, aes(body_mass_g, flipper_length_mm)) + geom_point() +
  geom_text(
    data = label_data, aes(label = label),
    size = 10/.pt, hjust = 1  # 10pt, right-justified
  ) +
  geom_smooth(method = "lm", se = FALSE) + facet_wrap(vars(species))
```



Further reading

- Data Visualization—A Practical Introduction: Chapter 6.5: Tidy model objects with broom
- **broom** reference documentation: https://broom.tidymodels.org/
- Article on using **broom** with **dplyr**: **broom** and **dplyr**