Wk 13 - Regression

Nicholas Kortessis

2025-04-16

Linear Regression

The basics: simple linear regression

Time to do some basic linear regression. To do so, we are going to use data collected on three species of penguins studied in Antarctica. We don't need to load this data from a file on our computer, it's saved in a package. The package is named **palmerpenguins**(Horst, Hill, and Gorman 2020). To load it, let's install the package.

```
install.packages('palmerpenguins')
```

That installs the package. To access it, we have to load it into R for use and then we can look in the object **penguins**, like this

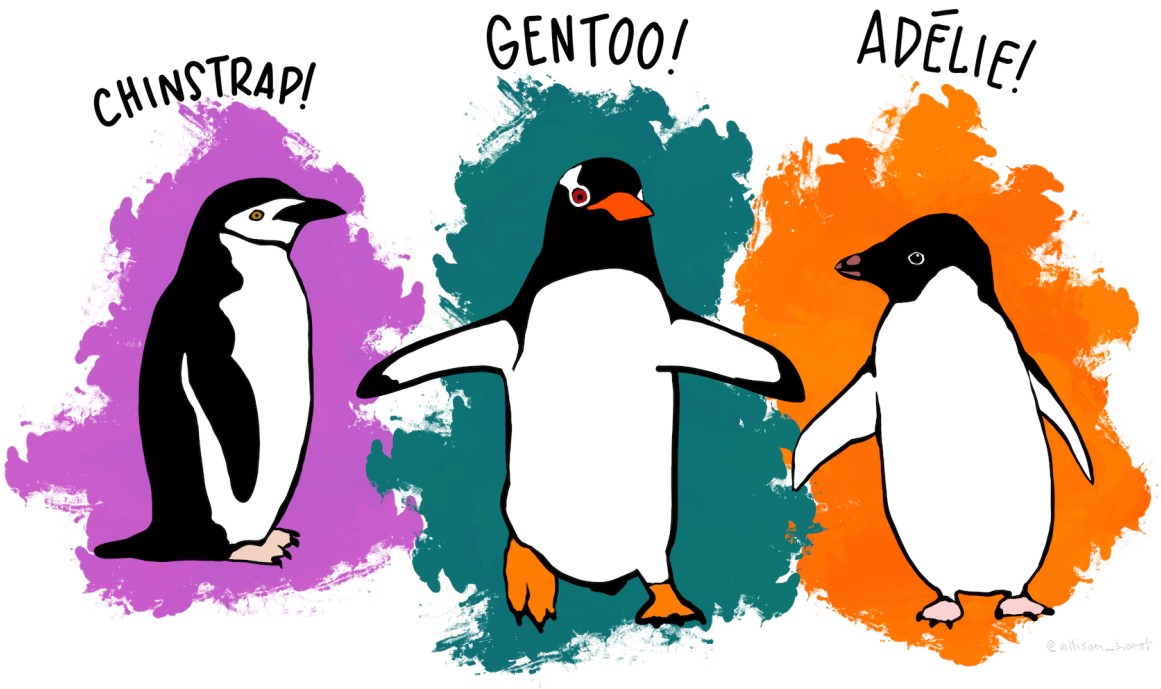
```
library(palmerpenguins)
head(penguins) # Just the first few rows of the data set
```

```
## # A tibble: 6 x 8
##   species island    bill_length_mm bill_depth_mm flipper_length_mm body_mass_g
##   <fct>   <fct>              <dbl>         <dbl>             <int>       <int>
## 1 Adelie  Torgersen           39.1          18.7               181        3750
## 2 Adelie  Torgersen           39.5          17.4               186        3800
## 3 Adelie  Torgersen           40.3          18                 195        3250
## 4 Adelie  Torgersen           NA            NA                  NA          NA
## 5 Adelie  Torgersen           36.7          19.3               193        3450
## 6 Adelie  Torgersen           39.3          20.6               190        3650
## # i 2 more variables: sex <fct>, year <int>
```

```
str(penguins) # A summary of the data set
```

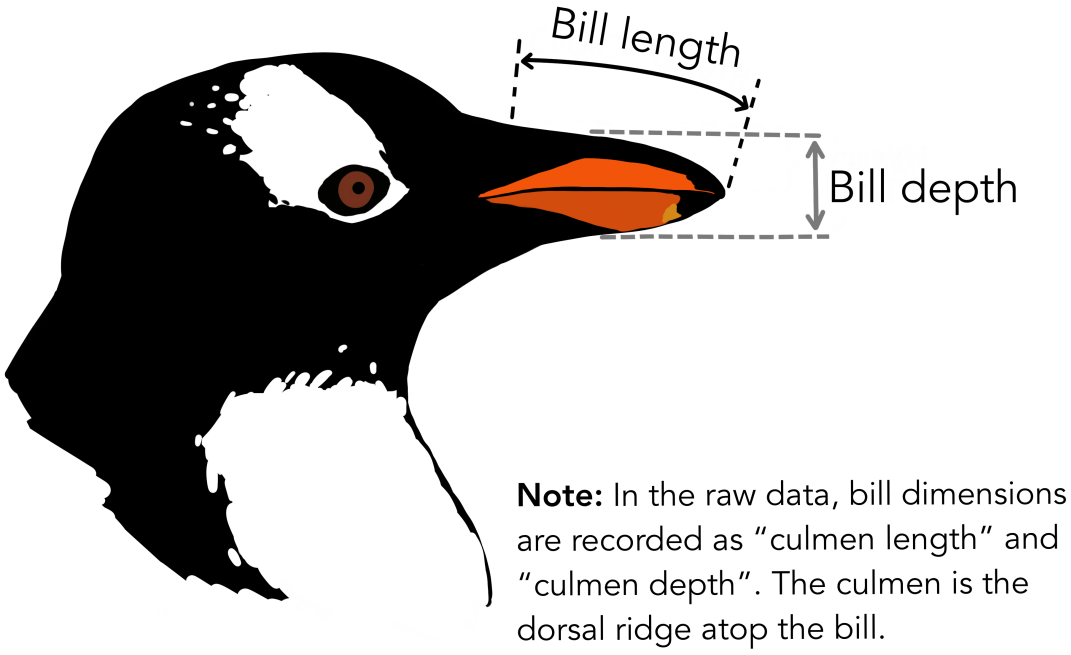
```
## tibble [344 x 8] (S3: tbl_df/tbl/data.frame)
##  $ species          : Factor w/ 3 levels "Adelie","Chinstrap",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ island           : Factor w/ 3 levels "Biscoe","Dream",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ bill_length_mm   : num [1:344] 39.1 39.5 40.3 NA 36.7 39.3 38.9 39.2 34.1 42 ...
##  $ bill_depth_mm    : num [1:344] 18.7 17.4 18 NA 19.3 20.6 17.8 19.6 18.1 20.2 ...
##  $ flipper_length_mm: int [1:344] 181 186 195 NA 193 190 181 195 193 190 ...
##  $ body_mass_g      : int [1:344] 3750 3800 3250 NA 3450 3650 3625 4675 3475 4250 ...
##  $ sex              : Factor w/ 2 levels "female","male": 2 1 1 NA 1 2 1 2 NA NA ...
##  $ year             : int [1:344] 2007 2007 2007 2007 2007 2007 2007 2007 2007 2007 ...
```

This data set contains information about individual penguins from three species illustrated below.

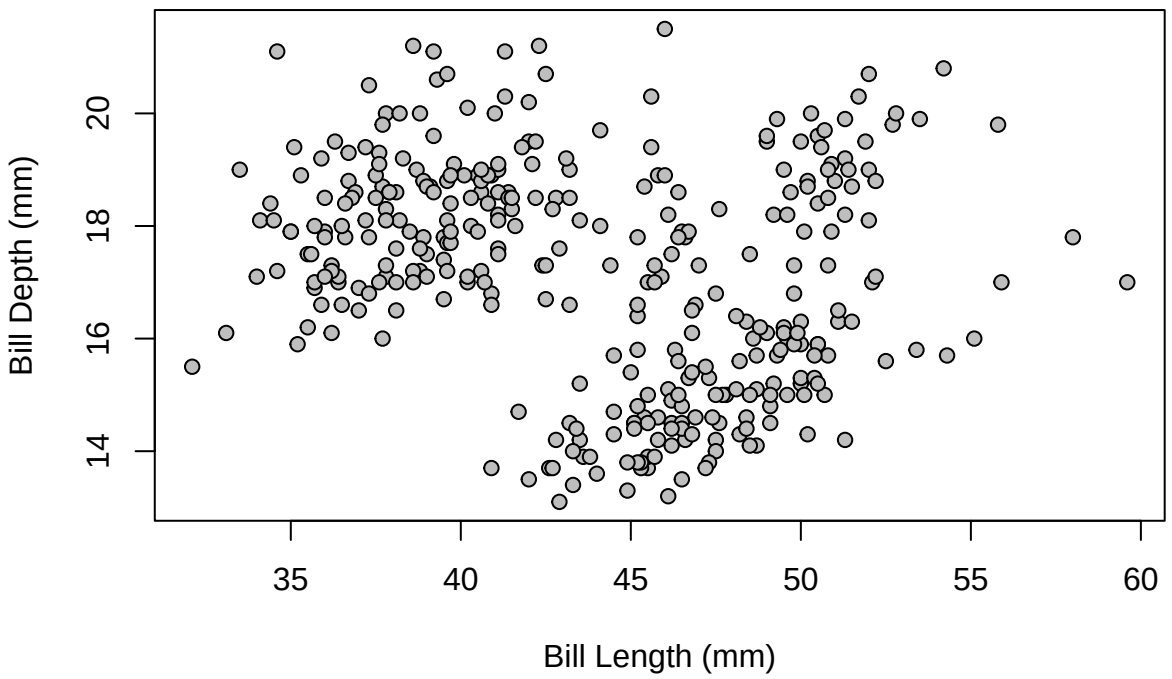


Plotting Associations Between Characters

Let's look at the relationship between bill length and bill depth. These measurements are in the data set and are illustrated in the figure below.



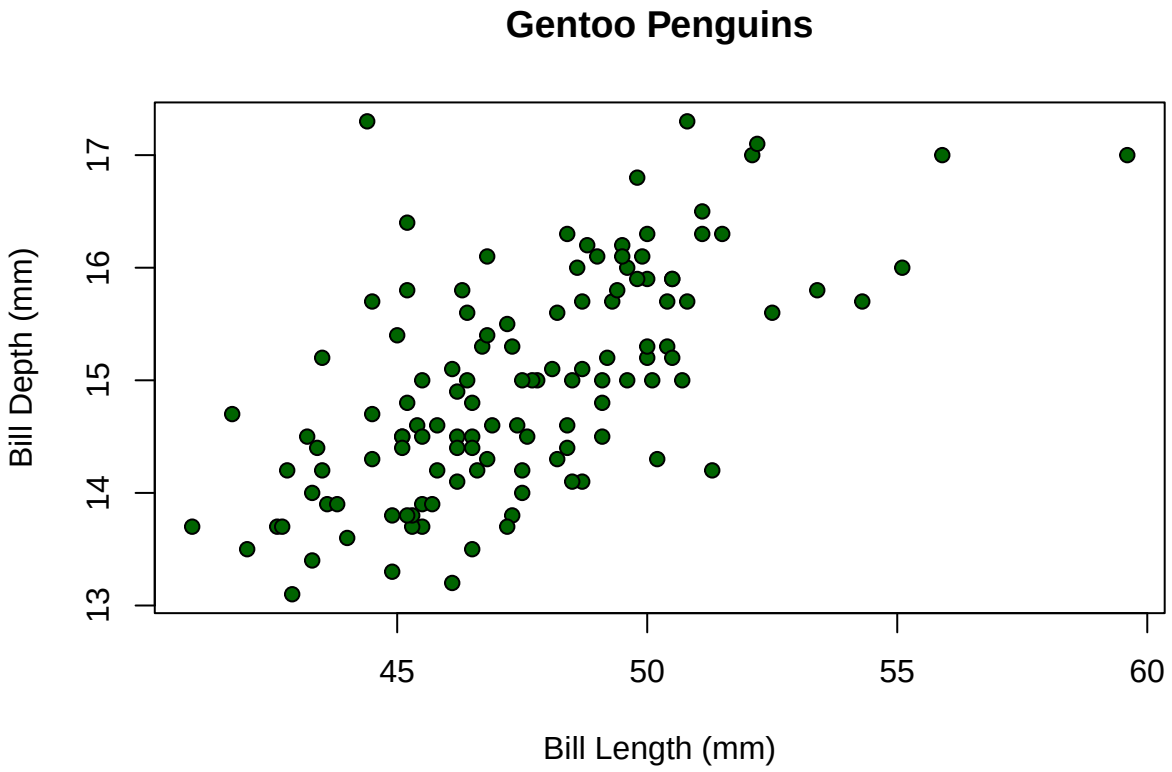
```
plot(penguins$bill_length_mm, penguins$bill_depth_mm,
     pch = 21, bg = 'gray', xlab = 'Bill Length (mm)',
     ylab = 'Bill Depth (mm)')
```



Hmm. This is the relationship when all species are lumped together. Species differences here are obscuring relationships between bill length and depth. Let's do this for one species. I think the Gentoo penguins are pretty cute. Let's use them.

```
gentoos <- subset(penguins, subset = (species == 'Gentoo'))

plot(gentoos$bill_length_mm, gentoos$bill_depth_mm,
     pch = 21, bg = 'darkgreen', xlab = 'Bill Length (mm)',
     ylab = 'Bill Depth (mm)', main = 'Gentoo Penguins')
```



This looks like this data could be reasonably modeled with linear regression. As a reminder, a linear model looks like this

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i.$$

For this data, we have the following interpretations:

- Y_i is a penguins bill depth in mm
- X_i is a penguin's bill length in mm
- $\epsilon_i \sim \text{Normal}(\mu = 0, \sigma^2)$ is a penguin's residual, i.e., the difference between the actual bill depth and that predicted solely by bill length. This is assumed to be normally distributed with mean 0 and variance σ^2.
- β_0 is the intercept of the model. This gives the value of bill depth when the penguin has a bill length of 0 mm.
- β_1 is the slope of the model. This is how much bill depth changes on average (in mm) for every mm increase in the length of a penguin's bill.

Let's first fit a linear regression. To make this a bit easier, the `lm` (linear model) function includes its own `subset` command that means we don't need to subset the data first.

```
gentoo.mdl <- lm(bill_depth_mm ~ bill_length_mm,
                 subset = (species == 'Gentoo'),
                 data = penguins)
```

Ok, model is fit. Let's check to see what model we fit using the `summary` command.

```
summary(gentoo.mdl)
```

```
##
## Call:
## lm(formula = bill_depth_mm ~ bill_length_mm, data = penguins,
```

```
##     subset = (species == "Gentoo"))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.55952 -0.52572 -0.06658  0.46041  2.95390
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     5.25101    1.05481   4.978 2.15e-06 ***
## bill_length_mm  0.20484    0.02216   9.245 1.02e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7543 on 121 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.4139, Adjusted R-squared:  0.4091
## F-statistic: 85.46 on 1 and 121 DF,  p-value: 1.016e-15
```

Here are the components of the model summary.

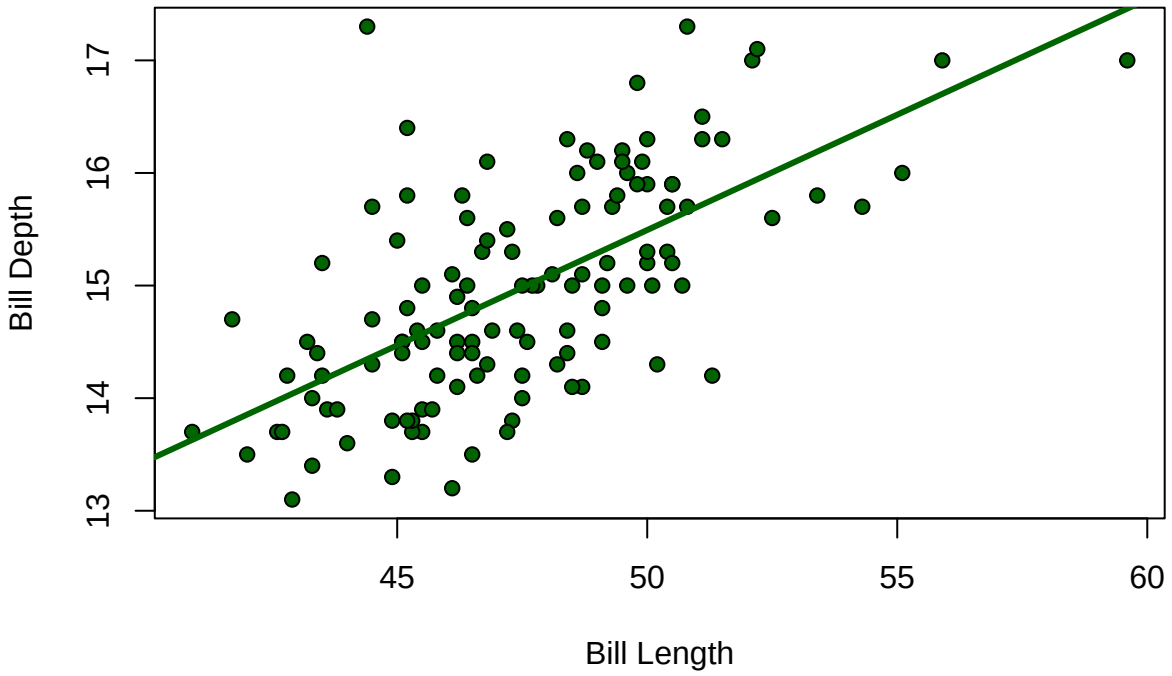
- **Call:** This is a restatement of the model that is fit; depth as a function of length for the penguins data set subsetted by Gentoo species.
- **Residuals:** This section gives a summary statistic of the distribution of residuals. These values are what is used to create a boxplot of residuals. Remember that linear regression assumes that residuals (everything not explained by the line) should be normally distributed. So these residual summary statistics should largely look symmetric (more on this in a minute).
- **Coefficients:** Here are the estimated parameters of the model. It includes estimates of the intercept and the slope, indicated by the row `bill_length_mm`. In addition, there are estimates of the standard error of each estimate. For hypothesis testing, there is also a t-stat and p-value for each estimate under the hypotheses that the intercept is zero and the slope is zero.
- **Residual standard error:** This is the estimate of the standard deviation of the residuals. Remember how $\epsilon \sim \text{Normal}(\mu = 0, \sigma^2)$? This is the estimate for σ. We write the residual standard error as $\hat{\sigma}$ reflecting the fact that the standard deviation of the residuals is also estimated from the data. The degrees of freedom represents how many data points were used to estimate the residual standard error.
- **R-squared:** This row has information about how much of the variation in the data is explained by the model. There is the raw value (Multiple R-squared) and a value that adjusts for small sample sizes and the number of parameters in the model(Adjusted R-squared). The adjusted value is more helpful here because it helps account for overfitting like we discussed in lecture. Models with more parameters always fit better, even if at the cost of overfitting.
- **F-statistic:** The F-value from ANOVA. This is the ratio of Mean Squares from the regression (the regression analogue to MS_{Groups}) compared to the Mean Squares from the residuals (the regression analogue to MS_{error}). The p-value is for the hypothesis that bill length has no effect on bill depth. In the case of a single predictor variable, the F-statistic and p-value gives nearly the same information as the slope p-value. This can be different for **multiple regression**, where multiple predictors are used.

It looks like there is a clear effect of bill length on bill depth. Should we trust this model? To do that, let's check the fit of our model. First, let's plot the model on the data. First, plot the data and then use `abline` to put in the estimated coefficients of the model. To grab the estimates, we can ask for them from the fitted model using `gentoo.mdl$coefficients`.

```
gentoo.mdl$coefficients #Intercept and slope estimates
```

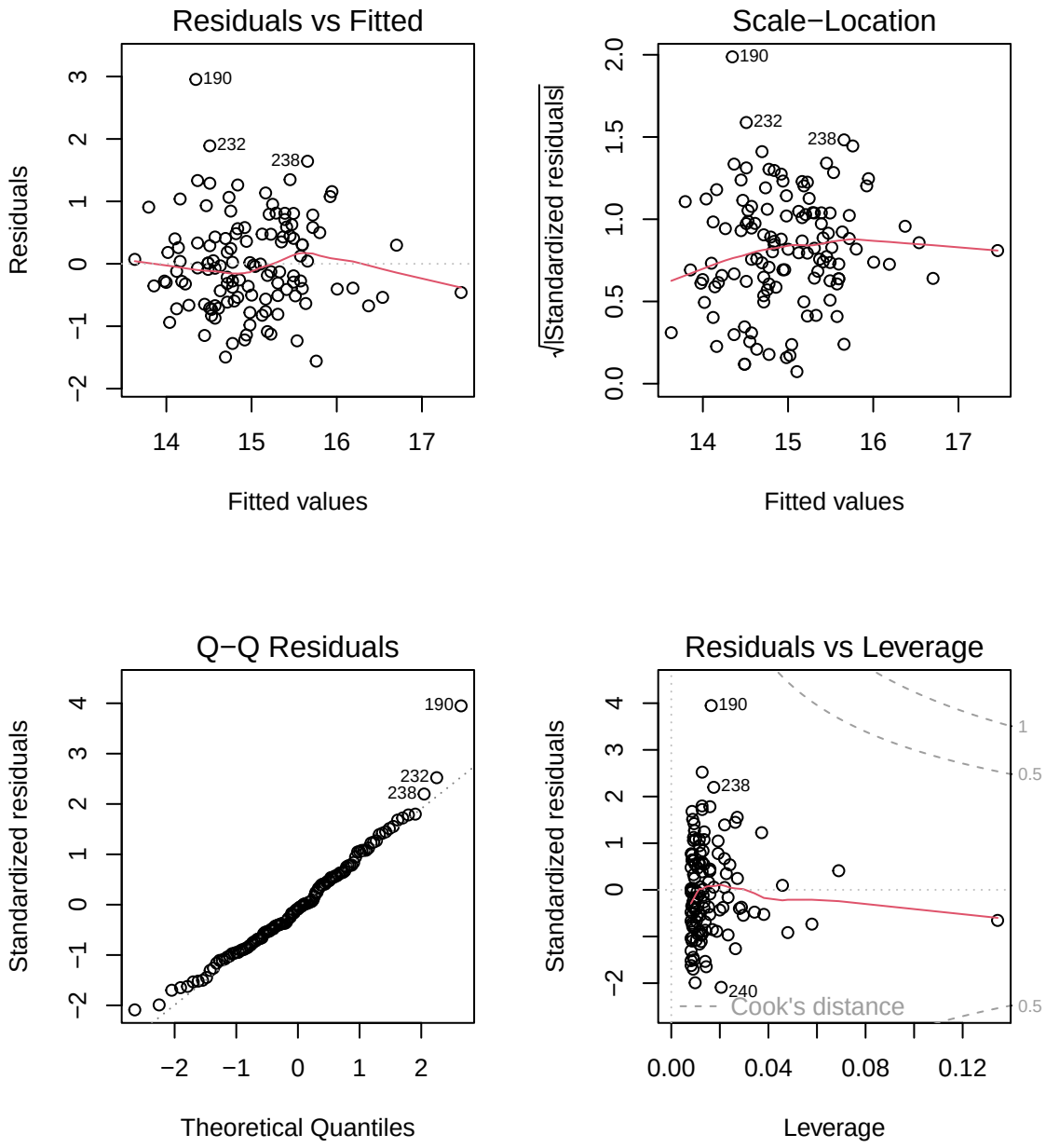
```
##    (Intercept) bill_length_mm
##      5.2510084      0.2048443
```

```r
plot(gentoos$bill_length_mm, gentoos$bill_depth_mm,
     pch = 21, bg = 'darkgreen', # Make gray filled circles
     xlab = 'Bill Length',
     ylab = 'Bill Depth')
abline(gentoo.mdl$coefficients,
       col = 'darkgreen', lwd = 3)
```



That looks pretty good. It looks like the residuals are pretty evenly spread above and below the line. But we should check the model fit using the regression diagnostics. The easiest way to do this is by plotting the model fit.

```r
par(mfcol = c(2,2))
plot(gentoo.mdl)
```



I see no obvious patterning in the residuals. The topleft plot shows no consistent pattern in the **average** residual. The topright plot shows no consistent change in the **magnitude** of the residuals. The bottomleft plot shows decent fit to the normal distribution of the residuals. And the bottomright plot shows that the individuals with highest leverage have moderate residuals and so don't seem to have much influence on the regression. These are all good signs that the model fits well.

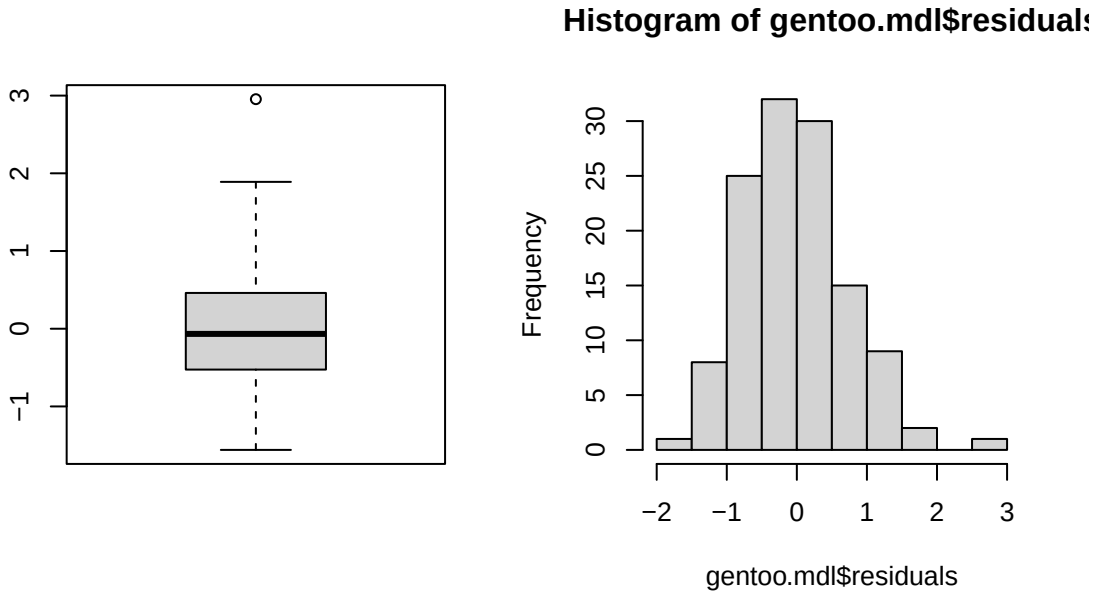
An important consideration is what the residuals look like. It's easy to ask for the residuals in R. They are calculated as part of the model fitting process. Let's look at the distribution of residuals. We can plot residuals in multiple ways. We can use boxplots, stripcharts, histograms, and density plots. Here they are

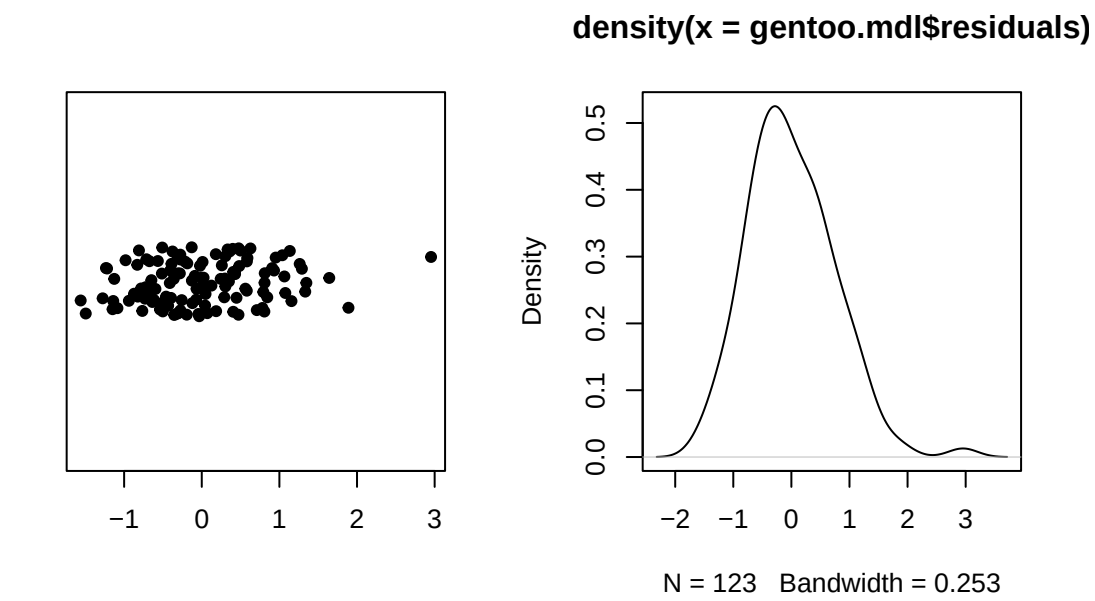
```
head(gentoo.mdl$residuals) # The residual values
```

```
##         153         154         155         156         157         158
## -1.4943325  0.8067746 -1.1269278 -0.2932254 -0.5015990 -1.2762702
```

```
par(mfcol = c(2,2))
boxplot(gentoo.mdl$residuals)
```

```
stripchart(gentoo.mdl$residuals, method = 'jitter',
           pch = 19)
hist(gentoo.mdl$residuals)
plot(density(gentoo.mdl$residuals))
```

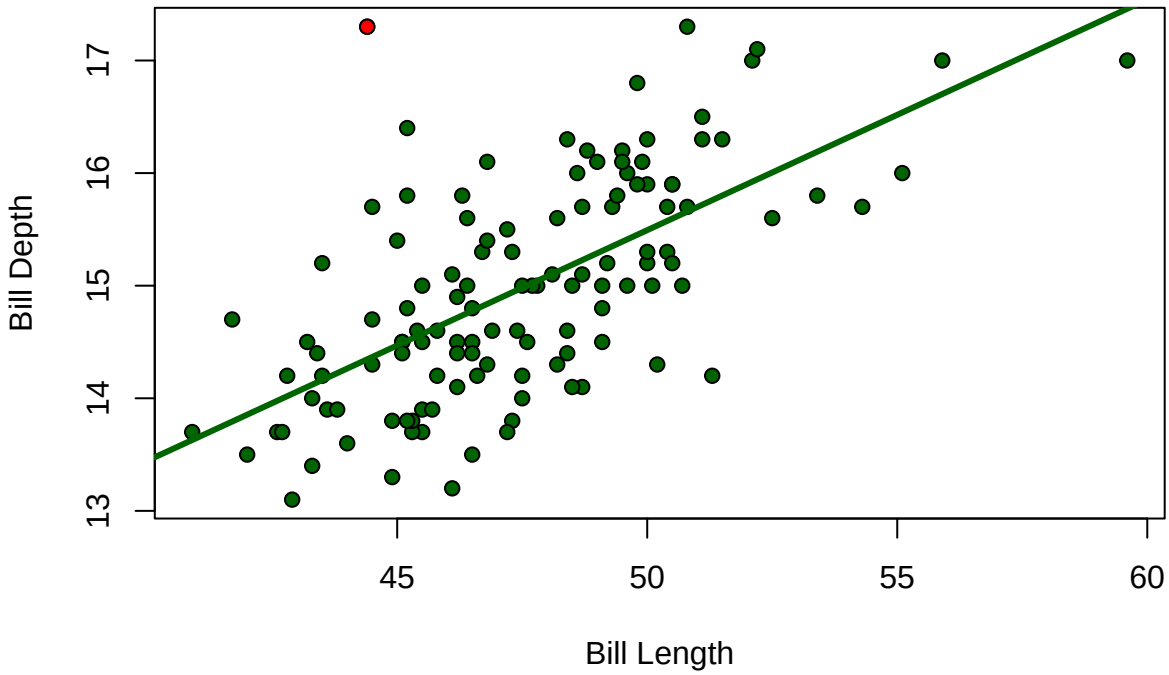




You can see that these are centered on zero, as should be the case for the residuals. They look normally distributed except for one point with really large residuals. This point is identified in the regression diagnostics as point 190. This means it is in row 190 of the `penguins` data set. Let's find it on the original plot and make it red.

```
plot(gentoos$bill_length_mm, gentoos$bill_depth_mm,
     pch = 21, bg = 'darkgreen', # Make gray filled circles
     xlab = 'Bill Length',
     ylab = 'Bill Depth')
```

```r
abline(gentoo.mdl$coefficients,
       col = 'darkgreen', lwd = 3)
points(penguins$bill_length_mm[190], # Plot point 190 only
       penguins$bill_depth_mm[190],
       pch = 21, bg = 'red')
```



There is something special about this individual with an exceptionally deep bill given it's bill length. Sometimes, this is a sign that there is an error in transcribing the data. Sometimes, there are just big beefy penguins. We can't tell, but it's good to be able to identify these individuals in the data set. You'll notice that point 190 is marked in nearly all the regression diagnostic plots. This point is identified because it's residual value is so much different from all the others. That's just a sign that it is worthy of a second look. **It's not a sign that we should just remove the data point**.

Interpreting the Line

So we have a model. Now we need to interpret what it means. Let's pull up the model summary again and interpret the values of the parameter estimates.

```r
summary(gentoo.mdl)
```

```
##
## Call:
## lm(formula = bill_depth_mm ~ bill_length_mm, data = penguins,
##     subset = (species == "Gentoo"))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.55952 -0.52572 -0.06658  0.46041  2.95390
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     5.25101    1.05481   4.978 2.15e-06 ***
## bill_length_mm  0.20484    0.02216   9.245 1.02e-15 ***
```

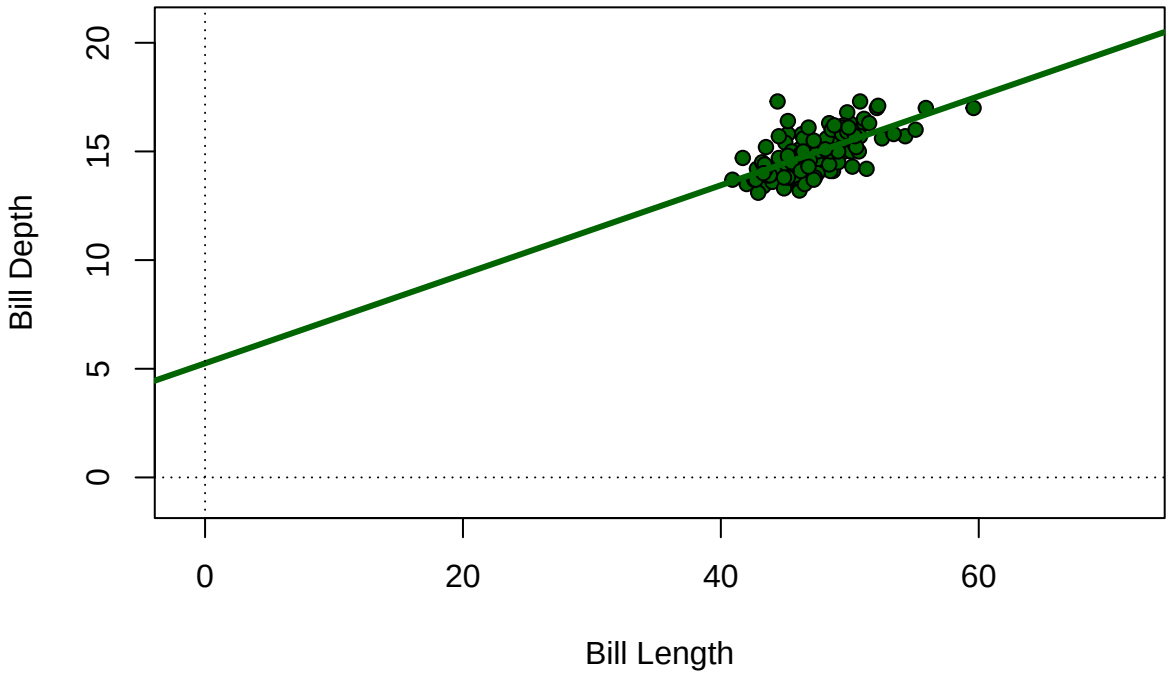
```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7543 on 121 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.4139, Adjusted R-squared:  0.4091
## F-statistic: 85.46 on 1 and 121 DF,  p-value: 1.016e-15
```

Let's start with the slope. It says that the slope intercept is $\beta_1 = 0.205$. This means that, on average across penguins, birds that differ by 1mm of bill length differ by 0.205 mm of bill depth. Another way to think about this is that bills get longer faster than they get deeper.

The intercept is estimated as $\beta_0 = 5.25$.

Checkpoint 1: What is the biological interpretation of this intercept? What do you think it should be? What does the hypothesis test say about this intercept? Here is a fair warning. Interpreting the intercept by itself is a practice in **extrapolation**. This is extrapolation because we have no birds anywhere near the intercept. As such, the intercept interpreted literally is far outside the bounds of what we have evidence for. Here is a picture of what I mean.

```
plot(gentoos$bill_length_mm, gentoos$bill_depth_mm,
     pch = 21, bg = 'darkgreen', # Make gray filled circles
     xlab = 'Bill Length',
     ylab = 'Bill Depth',
     xlim = c(-1, 1.2*max(na.omit(gentoos$bill_length_mm))),
     ylim = c(-1, 1.2*max(na.omit(gentoos$bill_depth_mm))))
abline(gentoo.mdl$coefficients,
       col = 'darkgreen', lwd = 3)
abline(h = 0, lty = 3)
abline(v = 0, lty = 3)
```



We shouldn't trust this value anymore than we should trust a prediction about the bill depth of a bird with a bill length of 5 meters! That's not going to happen either. The model makes predictions about that, but it's so far outside the realm of evidence that we have that it's not even worth considering.

A better approach is to use the intercept and slope to make interpretations about the bill depth of a bird with average bill length.

```
gentoo.mdl$coefficients[1] + gentoo.mdl$coefficients[2]*mean(gentoos$bill_length_mm, na.rm = T)
```

```
## (Intercept)
##    14.98211
```

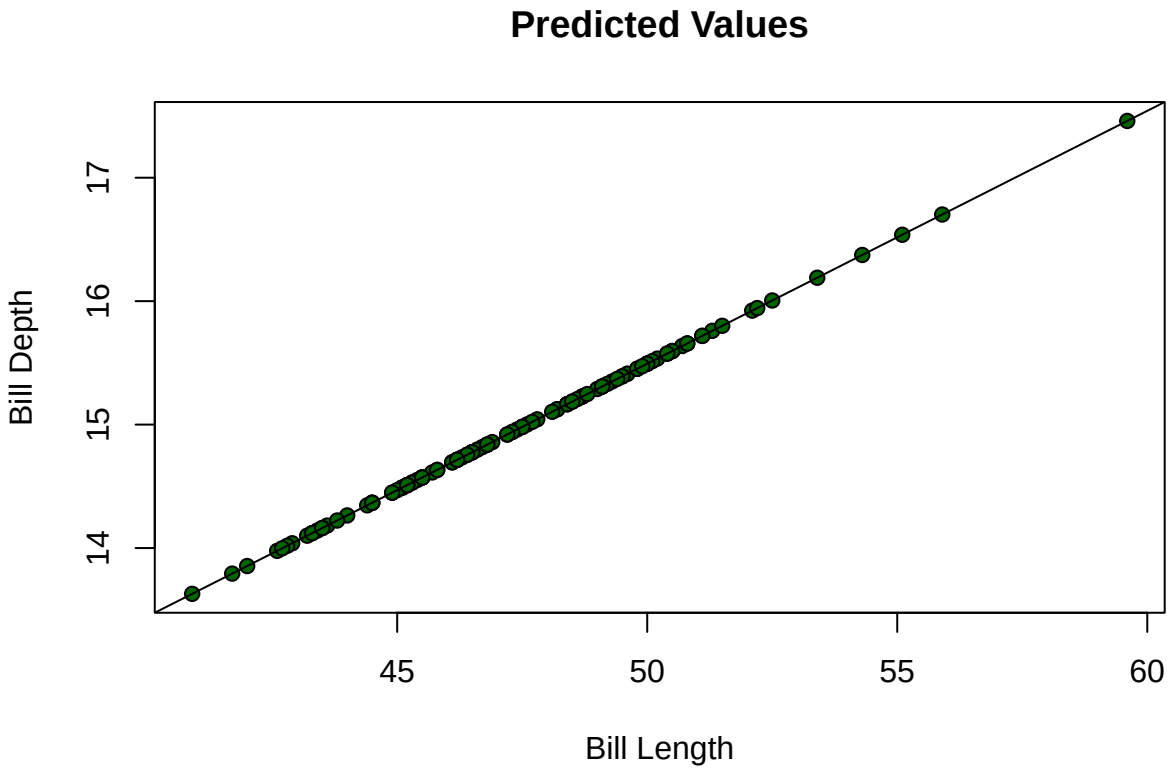
This says we predict a bird with average bill length to have bill depth of about 15mm.

Making Predictions From the Line

Maybe we want to make predictions about birds for different bill lengths from the line. The fitted model actually gives predicted bill depths for every bird given their bill length. We can call them up from `gentoo.mdl$fitted.values`.

Let's plot them.

```
plot(gentoo.mdl$model[,'bill_length_mm'], # Bill lengths
     gentoo.mdl$fitted.values, # Predicted bill depths
     pch = 21, bg = 'darkgreen',
     xlab = 'Bill Length', ylab = 'Bill Depth',
     main = 'Predicted Values')
abline(gentoo.mdl$coefficients)
```



But what if we've got new birds and we just measure their length? What is their bill depth? To do that, we can use the `predict` function of R. We just need to pass the fitted model to the function as well as the data points we want to predict. To make this work, the new data points have to be in a data frame where the column name is the same as the column name for the predictor in the original data frame. Here, the column name for the predictor is "bill_length_mm" so we make a new data frame with that column name.

```
new.bill.lengths <- data.frame(bill_length_mm = c(40,50,60))
predict.lm(gentoo.mdl, newdata = new.bill.lengths)
```

```
##        1        2        3
## 13.44478 15.49323 17.54167
```

This says that birds with bill lengths of 40mm, 50mm, and 60mm should have bill depths of 13.44mm, 15.5mm, and 17.54mm, respectively.

Checkpoint 2: What is the predicted bill depth for a bird with bill length of 43.256 mm?

Confidence Intervals of the Line

So we've got our best fit lines and we can make predictions from it. What about including uncertainty? To do that, we need can show the uncertainty in our estimate of the line by making confidence intervals of the line. We can do this again by using the function `predict` but now specifying that we want a confidence interval. The argument `level` specifies how much of the sampling distribution of lines you want to encompass. The default is 95%.

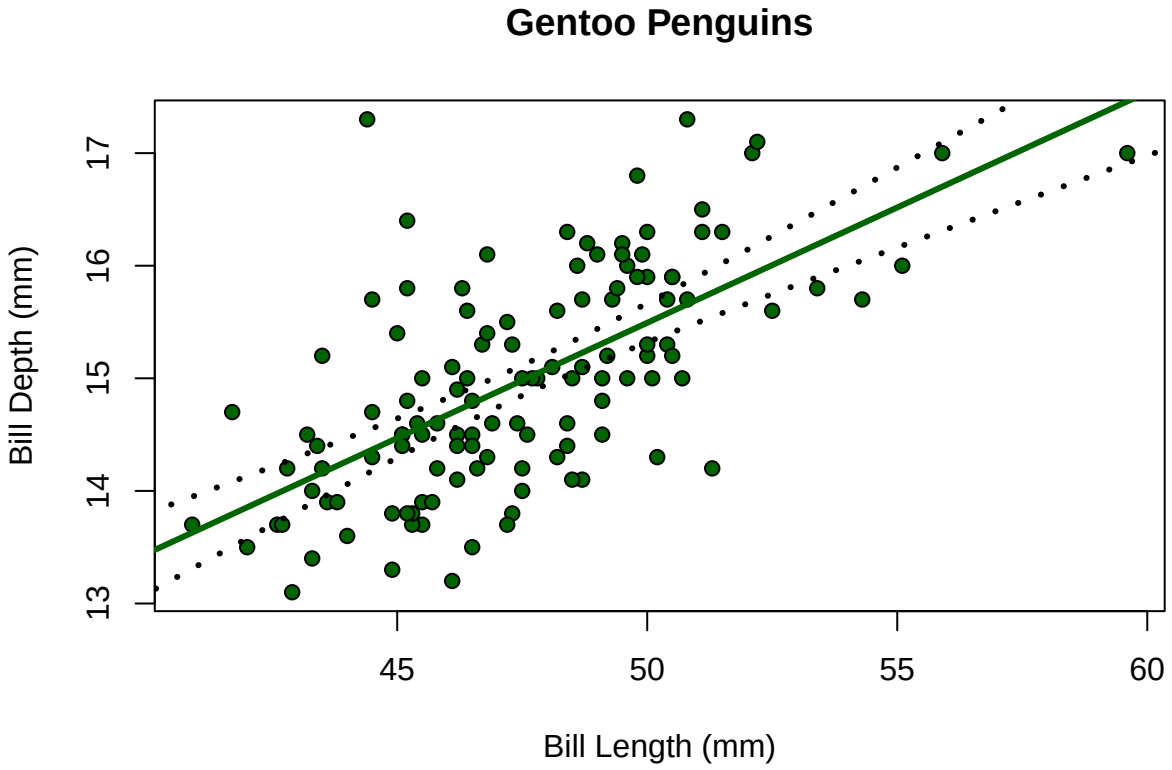
```
# Find 95% confidence intervals of the line
new.bill.lengths <- data.frame(
  bill_length_mm = seq(from = 30, to = 70, by = 0.01))
line.ci <- predict(gentoo.mdl, newdata = new.bill.lengths,
        interval = 'confidence',
        level = 0.95)
head(line.ci)
```

```
##        fit      lwr      upr
## 1 11.39634 10.61673 12.17595
## 2 11.39839 10.61921 12.17757
## 3 11.40044 10.62169 12.17918
## 4 11.40248 10.62417 12.18080
## 5 11.40453 10.62665 12.18242
## 6 11.40658 10.62913 12.18403
```

You can see that it gives `lwr` (lower) and `upr` (upper) bounds on the confidence interval for each value of new data we gave it. (It also gives the fitted values on the line). Now we can just plot these lines on the original data plot. To make things easier, let's throw the bill lengths we used to create the confidence interval in the same object as a new column.

```
line.ci <- data.frame(line.ci,
                      bill_length_mm = new.bill.lengths$bill_length_mm)
# First plot the data
plot(gentoos$bill_length_mm, gentoos$bill_depth_mm,
     pch = 21, bg = 'darkgreen', # Make gray filled circles
     xlab = 'Bill Length (mm)',
     ylab = 'Bill Depth (mm)',
     main = 'Gentoo Penguins')
abline(gentoo.mdl$coefficients,
       col = 'darkgreen', lwd = 3)

# Lower bound of the CI
lines(line.ci$bill_length_mm, line.ci[,'lwr'],
      lty = 3, lwd = 3) # Make them dashed and wider
# Upper bound of the CI
lines(line.ci$bill_length_mm, line.ci[,'upr'],
      lty = 3, lwd = 3)
```



Not bad looking.

Making predictions using the residuals

We can make predictions from the line, but you can tell from the plot that penguins' bill depths don't fall right along the line. Bill depths are determined by other factors besides bill length. We can see this from the estimate of the R^2 value, which for this model is about 41%. This means more than half of the variation in bill depth is from other factors besides bill length.

We can include that other, unknown variability by levering estimates of the variation in the distribution of residuals. We can visualize this with the **prediction interval**.

Prediction intervals don't tell us about 95% of the lines, but about 95% of the penguins' bill depths for a given bill length. Again, we can use the function `predict.lm` to find these, but now we have to ask for a prediction rather than confidence interval.

```
pred.int <- predict.lm(gentoo.mdl, newdata = new.bill.lengths,
                       interval = 'prediction', level = 0.95)
pred.int <- data.frame(pred.int,
                       bill_length_mm = new.bill.lengths$bill_length_mm)
```

This is really nice. It tells us, if I know the bill length of a Gentoo penguin, I can tell you what the bill depths of 95% of penguins looks like. Here is an example assuming a penguin has bill length of 40mm.

```
subset(pred.int, subset = (bill_length_mm == 40.00))
```

```
##           fit      lwr      upr bill_length_mm
## 1001 13.44478 11.90974 14.97982             40
```

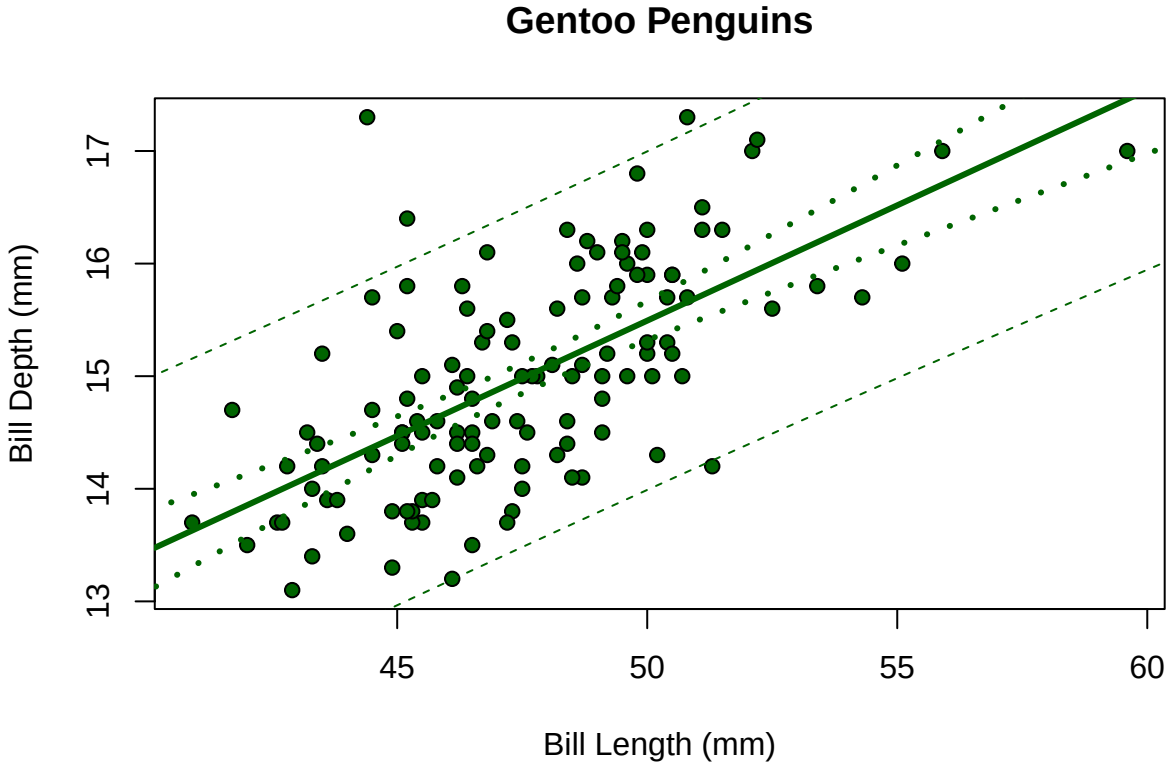
There you go. It says that 95% of Gentoo penguins have bill depths between 11.90 mm and 14.97 mm and that the average Gentoo penguin has bill depth of 13.44 mm.

Let's plot this interval on our plot of the data points and the confidence interval.

```r
# First plot the data
plot(gentoos$bill_length_mm, gentoos$bill_depth_mm,
     pch = 21, bg = 'darkgreen', # Make gray filled circles
     xlab = 'Bill Length (mm)',
     ylab = 'Bill Depth (mm)',
     main = 'Gentoo Penguins')
# Best fit estimate for the line
abline(gentoo.mdl$coefficients,
       col = 'darkgreen', lwd = 3)

# Lower bound of the CI
lines(line.ci$bill_length_mm, line.ci[,'lwr'],
      lty = 3, lwd = 3,
      col = 'darkgreen') # Make them dashed and wider
# Upper bound of the CI
lines(line.ci$bill_length_mm, line.ci[,'upr'],
      lty = 3, lwd = 3,
      col = 'darkgreen')

# Lower bound of the PI
lines(pred.int$bill_length_mm, pred.int$lwr,
      lty = 2, lwd = 1,
      col = 'darkgreen') # Make them dashed and wider
# Upper bound of the PI
lines(pred.int$bill_length_mm, pred.int$upr,
      lty = 2, lwd = 1,
      col = 'darkgreen')
```



You can clearly see that most birds (but not all) fit within the bounds of the 95% prediction interval. In fact,

about 95% of birds should be in the bounds. Only about 5% should be outside. By my count, there are 5 out of 124 that don't fit in the prediction interval, which is 4%.

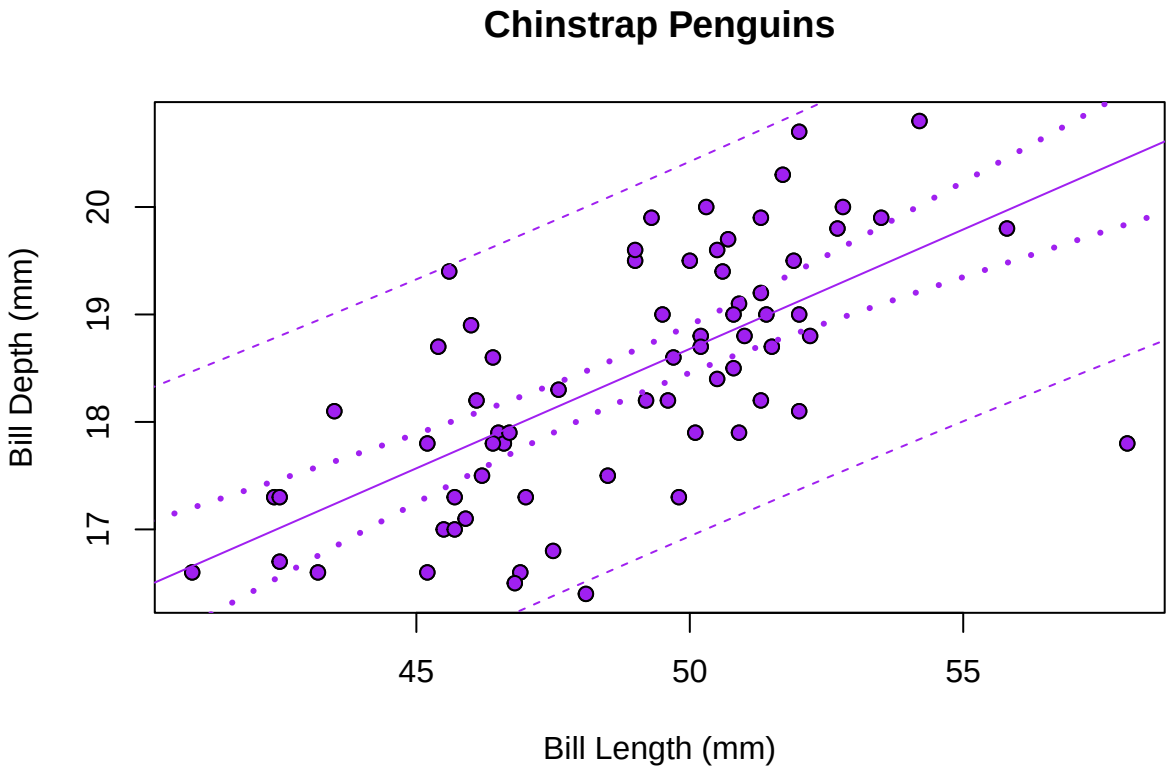
Checkpoint 3: What is the 95% prediction interval for a Gentoo penguin with bill length of 50.35 mm?

Your turn!

Checkpoint 4: Run a linear regression on bill depth as a function of bill length for Chinstrap penguins. Make sure to include the following.

1. A summary of the model and its fits.
2. Regression diagnostics. Does the data fit the model assumptions? Are there any features that might influence the fit?
3. A statement of whether the relationship between bill length and depth is the same or different for Chinstrap penguins and Gentoo penguins.
4. A plot showing the data, the best fit line, the 95% confidence interval of the line, and the 95% prediction interval of the line.
5. A comment on whether the relationship between bill depth and length is more precise for Chinstrap or Gentoo penguins.

Here's my final figure.



Some regression weirdness

So now we have estimates of how much bill depth changes as a function of bill depth, like this

$$\text{bill depth} = \beta_0 + \beta_1 \times \text{bill length}.$$

We have estimates to that the estimated relationship for Gentoo penguins is

$$\text{bill depth} = 5.251 + 0.205 \times \text{bill length}.$$

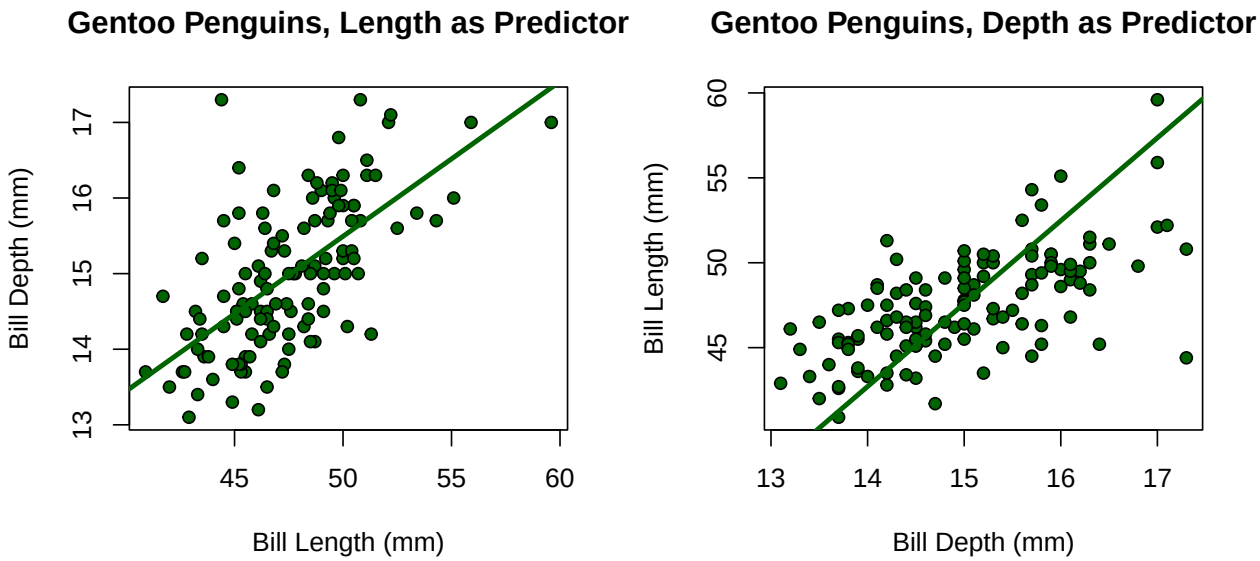
You might be asking, why are we assuming bill length controls bill depth? What if it works the other way around? We could just do a rewriting of this equation by solving for bill length as a function of bill depth. Doing so gives us

$$\text{bill length} = \frac{-5.251}{0.205} + \frac{1}{0.205} \times \text{bill depth} = -25.43 + 4.88 \times \text{bill depth}$$

Okay, great. Here's a picture of the line where we can switch whether bill length or bill depth is the predictor variable.

```
par(mfcol = c(1,2))
plot(gentoos$bill_length_mm, gentoos$bill_depth_mm,
     pch = 21, bg = 'darkgreen',
     xlab = 'Bill Length (mm)',
     ylab = 'Bill Depth (mm)',
     main = 'Gentoo Penguins, Length as Predictor')
# Best fit estimate for the line
abline(gentoo.mdl$coefficients,
       col = 'darkgreen', lwd = 3)

plot(gentoos$bill_depth_mm, gentoos$bill_length_mm,
     pch = 21, bg = 'darkgreen',
     ylab = 'Bill Length (mm)',
     xlab = 'Bill Depth (mm)',
     main = 'Gentoo Penguins, Depth as Predictor')
# Best fit estimate for the line
abline(a = -gentoo.mdl$coefficients[1]/gentoo.mdl$coefficients[2],
       b = 1/gentoo.mdl$coefficients[2],
       col = 'darkgreen', lwd = 3)
```

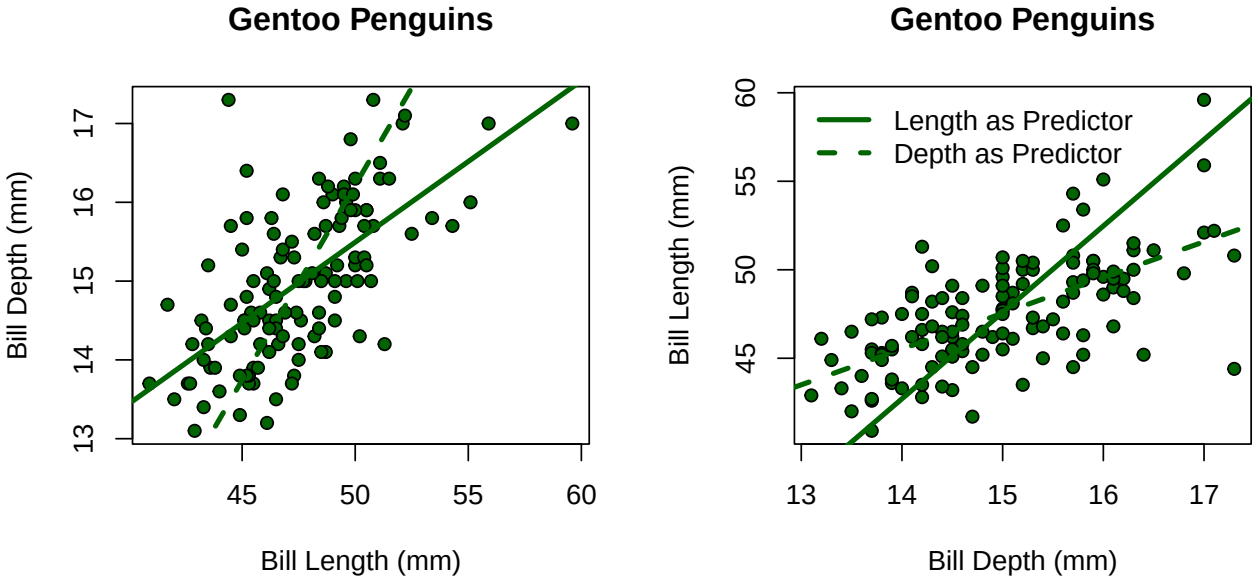


This says that two birds that differ in bill depth by 1 mm should differ in bill length by 4.88mm. Let's do this specific regression to see what the model fits.

```
gentoo.mdl2 <- lm(bill_length_mm ~ bill_depth_mm,
                  data = penguins,
                  subset = (species == 'Gentoo'))
gentoo.mdl2$coefficients
```

```
##   (Intercept) bill_depth_mm
##     17.229501      2.020768
```

Hmm. That's odd. That doesn't match what we calculated when trying to make depth a predictor. Did we make a mistake? Let's plot the lines for bill depth as predictor and bill length as predictor (I've suppressed the code to save you time and to save reading space).



This illustrates an important point. **The best fit line changes depending on which predictor you choose!** This is because each regression makes different assumptions about where the "error" in the model comes from. When we assume length is the predictor, we assume all the errors in the model come from things that are NOT bill length. When we assume depth is the predictor, we assume all the errors in the model come from things that are NOT bill depth. Thus, these models are actually very different, even though the data is the same.

A solution - covariances and correlations

In this case, there is no obvious choice of what determines one bill dimension. Without a specific question or scientific need, we have no way to pick one over the other. Oftentimes with this data, we just want to know how strong the association is between each of the variables. If we just want to do that, we can instead measure covariances and correlations.

Covariances are how much two factors vary together. They can be both positive or negative. Here is the equation for a covariance

$$\text{Cov}(X, Y) = \frac{1}{n-1} \sum_{i}^{n} (X_i - \bar{X})(Y_i - \bar{Y}).$$

This is just the numerator of the slope estimate from ordinary least squares regression. As a reminder, the slope estimator from ordinary least squares regression is

$$\hat{\beta}_1 = \frac{\sum_i^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sum_i^n (X_i - \bar{X})^2} = \frac{\frac{1}{n-1}\sum_i^n (X_i - \bar{X})(Y_i - \bar{Y})}{\frac{1}{n-1}\sum_i^n (X_i - \bar{X})^2} = \frac{\text{Cov}(X,Y)}{\text{Var}(X)},$$

where $\text{Var}(X) = \sigma_X^2$ is the variance of X. You can see that the slope is the covariance standardized by the variation in X. When we arbitrarily flip the predictor from bill length to bill depth, our estimate of the slope changes because the covariance is standardized by a different variance. Importantly, the covariance doesn't change when we flip the predictors. Let's calculate the covariance for bill length and bill depth.

```
cov(gentoos$bill_length_mm, gentoos$bill_depth_mm)
```

```
## [1] NA
```

```
# Looks like there is a missing value in the data set, likely because somem measurements are
# missing for some penguins. We can ignore these with the argument "use".
cov(gentoos$bill_length_mm, gentoos$bill_depth_mm,
    use = 'pairwise.complete.obs')
```

```
## [1] 1.94558
```

This tells us about the magnitude of covariation. If we want to standardize this to know how much the two vary in relation to their own variability, we can use a linear correlation coefficient, typically written as r.

$$\text{Correlation Coefficient} = r = \frac{\text{Cov}(X,Y)}{SD(X)SD(Y)}$$

This measures variation in units of how much both X and Y vary. Correlation coefficients always go between -1 and 1. Here is the correlation coefficient for the Gentoo penguin beak depth and length.

```
cor(gentoos$bill_length_mm, gentoos$bill_depth_mm,
    use = 'pairwise.complete.obs')
```

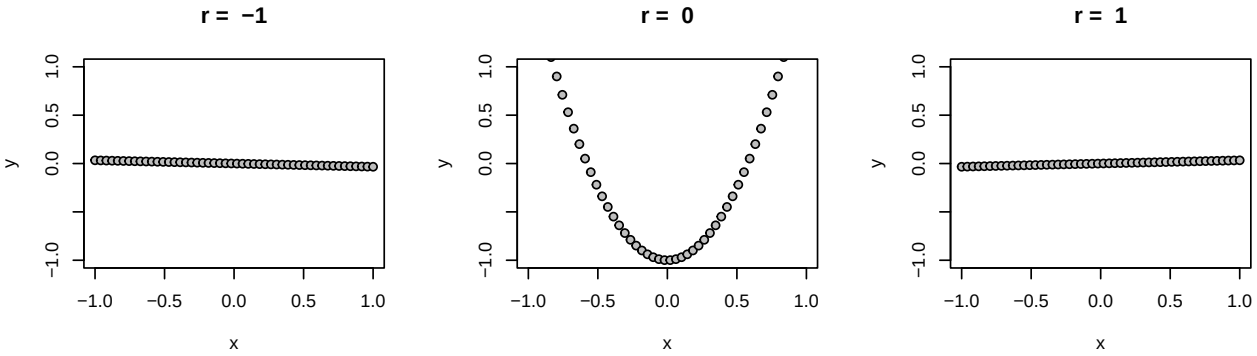
```
## [1] 0.6433839
```

This number is positive, meaning that bill depth and bill length increase together. That is is 0.64 means that they covary together quite strongly.

- X and Y points fall exaclty in a straight line with positive slope if $r = 1$.
- X and Y points fall exactly in a straight line with negative slope if $r = -1$.
- X and Y show no linear relationship if $r = 0$.

The square of the correlation coefficient, r^2, called the **coefficient of determination** goes from 0 to 1 and gives information about how close the point fall along a straight line relationship. It is best for understanding how much variation is encapsulated by Y when you know the value of X (or vice versa).

Wikipedia has some great examples of correlation coefficients and what they can and cannot tell you. Most importantly, **correlation coefficients do not tell you whether two values are associated with one another, nor how strong their relationship is.** Correlation coefficients can only identify straight line associations and measure how close close the linear association is, even if it is a subtle one. Below are three examples. They key here is to LOOK AT YOUR DATA! Humans are much better at picking up patterns (associations) than math!



Using Linear Regression to Fit Nonlinear Curves

Linear regression can also be used to fit nonlinear curves. Many are possible, but two are very common in biology: power law relationships and exponential relationships.

- Power laws look like the following

$$Y = aX^b,$$

  where a and b are parameters to be fit. They have the following interpretation: a is the value of Y when $X = 1$ and b is the proportional rate at which increases in X lead to increases in Y. When $b = 1$ X and Y change proportional to one another. When $b > 1$, increases in X cause *proportionately larger* increases in Y. When $0 < b < 1$, increases in X cause *proportionately smaller* increases in Y. Power laws are common in allometric relationships, i.e., those related to changes in body size components.

- Exponential relationships look like the following

$$Y = ae^{bX},$$

  where a is the value of Y when $X = 0$ and b is the exponential rate of growth (or decline). When $b > 0$, Y grows exponentially with X. When $b < 0$, Y declines exponentially with X. Exponential relationship show up all the time in population growth processes.

Power laws with metabolic body scaling

To fit a power law, we will use data from estimates of body mass and metabolic rate across a wide range of organisms. This is in the data set `Body_Size_Observations.csv`. Download it and looks at it.

```
metab.df <- read.csv('Body_Size_Observations.csv')
str(metab.df)
```

```
## 'data.frame':    1214 obs. of  15 variables:
##  $ phylum                              : chr  "Arthropoda" "Arthropoda" "Arthropoda" "Arthropoda" ..
##  $ class                               : chr  "Arachnida" "Arachnida" "Arachnida" "Arachnida" ...
##  $ order                               : chr  "Araneae" "Araneae" "Araneae" "Araneae" ...
##  $ family                              : chr  "Salticidae" "Salticidae" "Salticidae" "Salticidae" ..
##  $ genus                               : chr  "Marpissa" "Phidippus" "Sarinda" "Zygoballus" ...
##  $ species                             : chr  "Marpissa bina" "Phidippus clarus" "Sarinda hentzi" "Zy
##  $ specificEpithet                     : chr  "bina" "clarus" "hentzi" "rufipes" ...
##  $ body.mass                           : num  1.68e-04 2.60e-04 4.60e-06 3.00e-06 4.94e-09 8.59e-09 1
##  $ body.mass...units                   : chr  "kg" "kg" "kg" "kg" ...
##  $ original.body.mass                  : num  168 260 4.6 3 4.94 8.59 22.4 21.4 39.1 27.5 ...
```

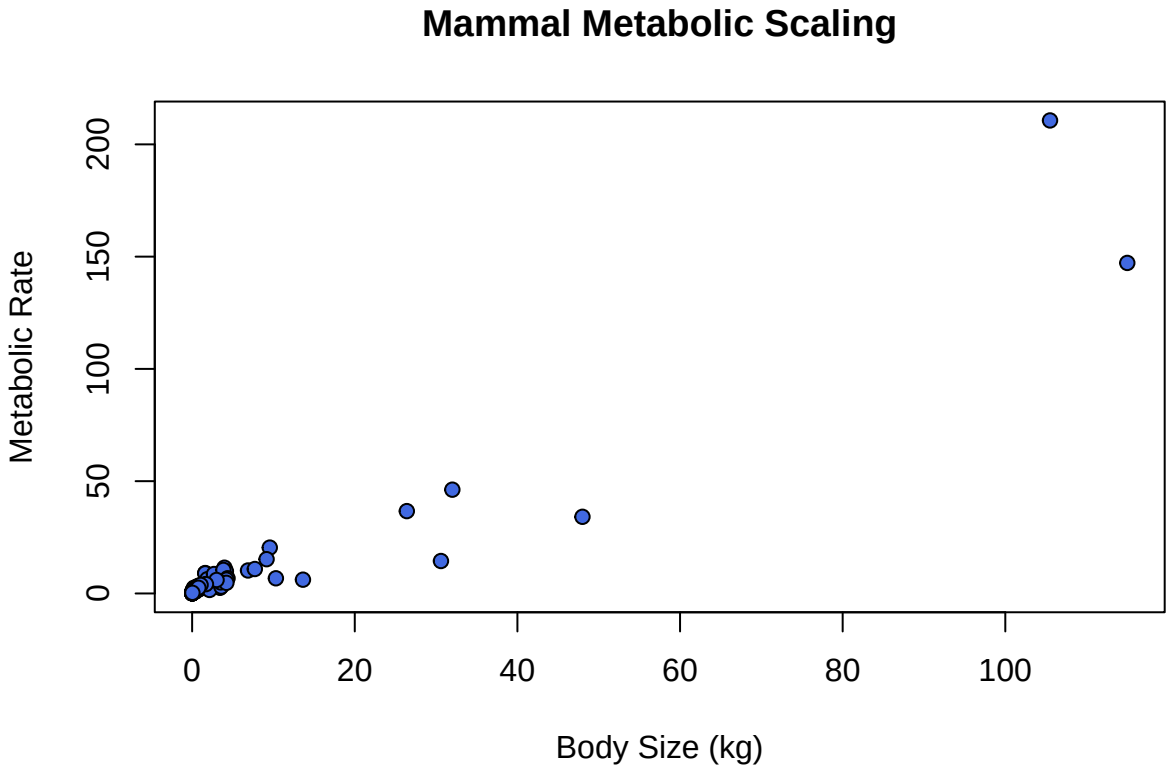
```
##  $ original.body.mass...units          : chr  "mg" "mg" "mg" "mg" ...
##  $ metabolic.rate                      : num  2.04e-04 3.61e-04 1.57e-05 4.71e-06 2.61e-08 ...
##  $ original.metabolic.rate             : num  26 46 2 0.6 0.95 2.07 1.77 1.69 1.41 1.45 ...
##  $ mass.specific.metabolic.rate        : num  1.22 1.39 3.42 1.57 5.28 ...
##  $ mass.specific.metabolic.rate...units: chr  "W/kg" "W/kg" "W/kg" "W/kg" ...
```

There is a lot in this data frame. Let's focus on mammals.

```
mammals.df <- subset(metab.df, subset = (class == "Mammalia"))
```

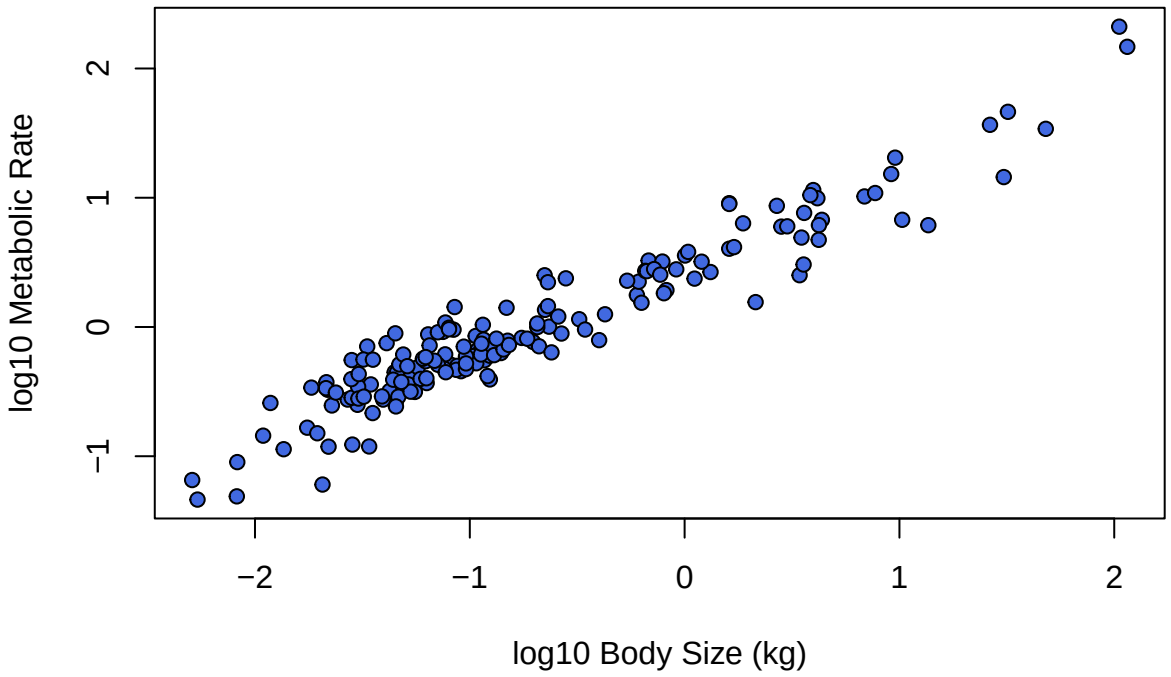
Now let's plot metabolic rate as a function of body size.

```
plot(mammals.df$body.mass, mammals.df$metabolic.rate,
     pch = 21, bg = 'royalblue',
     xlab = 'Body Size (kg)',
     ylab = 'Metabolic Rate',
     main = 'Mammal Metabolic Scaling')
```



Checkpoint 5: Make a linear regression of this data and identify a problem with this regression using regression diagnostics. To solve this problem, let's try log-transforming both the x- and y-axes. I'll use log base 10 because that one is relatively easy to interpret.

```
plot(log10(mammals.df$body.mass), log10(mammals.df$metabolic.rate),
     pch = 21, bg = 'royalblue',
     xlab = 'log10 Body Size (kg)',
     ylab = 'log10 Metabolic Rate')
```

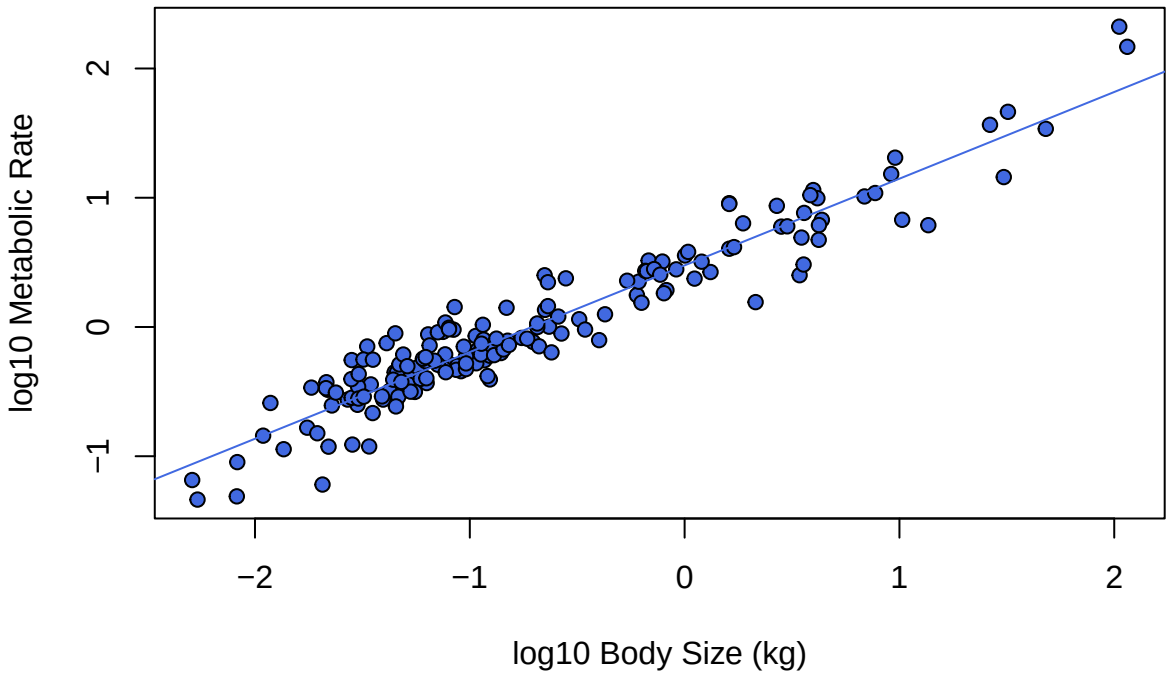


Hey that looks like a straight line! Since this is a straight line on a log-log scale, we have a power law relationship here. Let's fit a regression through this now.

```
powlaw.mdl <- lm(log10(metabolic.rate) ~ log10(body.mass),
                 data = mammals.df)
summary(powlaw.mdl)
```

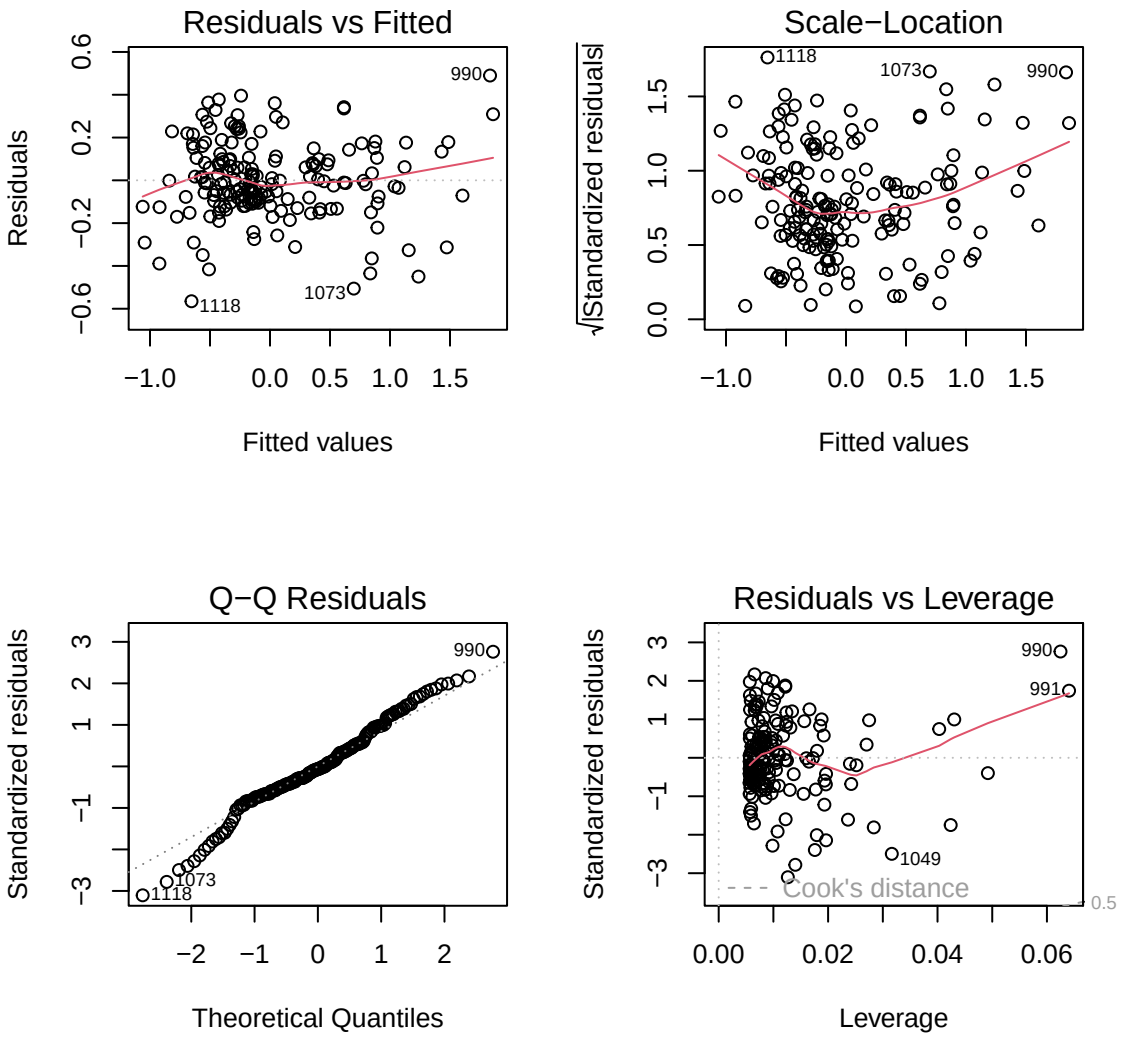
```
##
## Call:
## lm(formula = log10(metabolic.rate) ~ log10(body.mass), data = mammals.df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.56509 -0.10495 -0.01188  0.10455  0.48980
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.47670    0.01793   26.58   <2e-16 ***
## log10(body.mass)  0.67073    0.01590   42.18   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1832 on 175 degrees of freedom
## Multiple R-squared:  0.9104, Adjusted R-squared:  0.9099
## F-statistic:  1779 on 1 and 175 DF,  p-value: < 2.2e-16
```

```
plot(log10(mammals.df$body.mass), log10(mammals.df$metabolic.rate),
     pch = 21, bg = 'royalblue',
     xlab = 'log10 Body Size (kg)',
     ylab = 'log10 Metabolic Rate')
abline(powlaw.mdl, col = 'royalblue')
```



Let's look at the regression diagnostic plots.

```
par(mfcol = c(2,2))
plot(powlaw.mdl)
```



That looks much better. So what are we to make of this fit? Let's look at the coefficients.

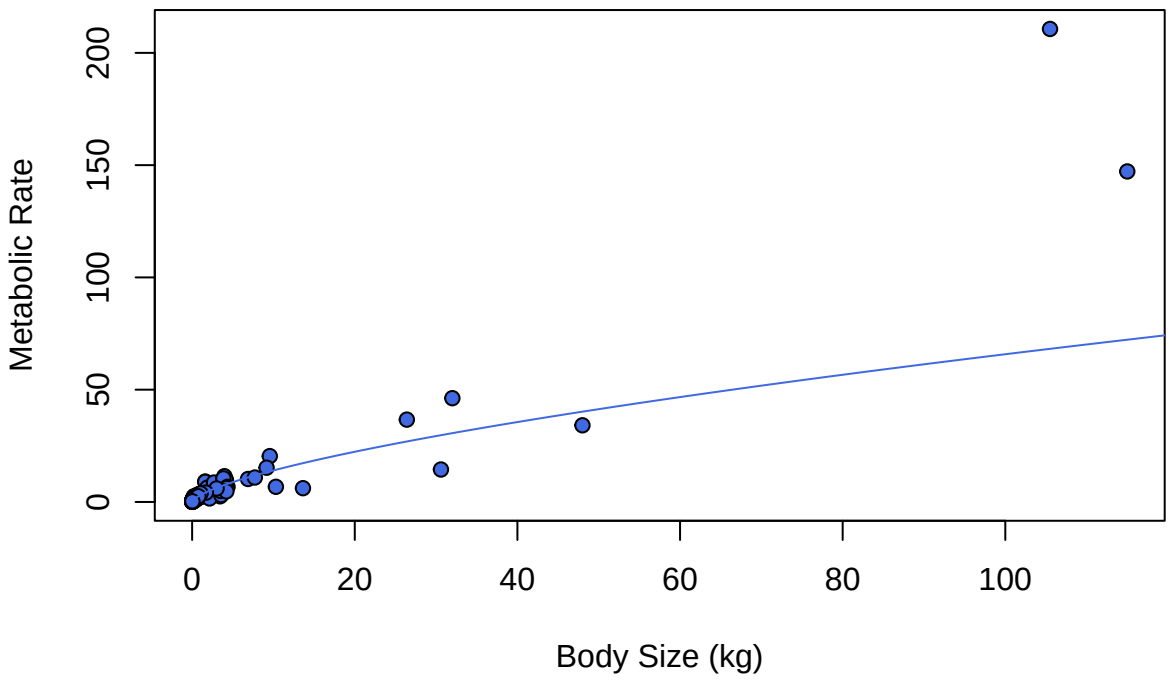
- The intercept is 0.478. The intercept for a power law represents the logarithm of a. That is $\beta_0 = \log_{10}(a)$. Hence, $a = 10^{\beta_0}$. This is the metabolic rate of a mammal with body mass of 1kg.
- The slope is 0.67. The slope for a power law is the parameter b. As you can see from this value, the slope is less than 1 meaning that for a proportional increase in body size of mammals, there is less than a proportional increase in metabolic rate. Stated differently, if you double the size of a mammal by mass, it's metabolic rate less than doubles.

Let's put these on the original graph so you can see.

```
a <- 10^powlaw.mdl$coefficients[1]
b <- powlaw.mdl$coefficients[2]

# Make some fake body sizes to draw the power law curve
body.sizes <- seq(from = 0, to = 150, length = 1000)
powlaw.fit <- a*body.sizes^b
# Plot data
plot(mammals.df$body.mass, mammals.df$metabolic.rate,
     pch = 21, bg = 'royalblue',
     xlab = 'Body Size (kg)',
     ylab = 'Metabolic Rate',
```

```
    )
# Add power law
lines(body.sizes, powlaw.fit, col = 'royalblue')
```



Checkpoint 6: Try this now with birds (Class: Aves). Find the power-law scaling relationship between body size and metabolic rate for birds. Do they show the same scaling relationship as mammals?

Exponential growth of measles cases in the US

Exponential relationships show up in population processes. There is lots of data about population growth of infectious diseases. Infectious diseases grow just like normal populations at the host level by counting how many hosts are infected.

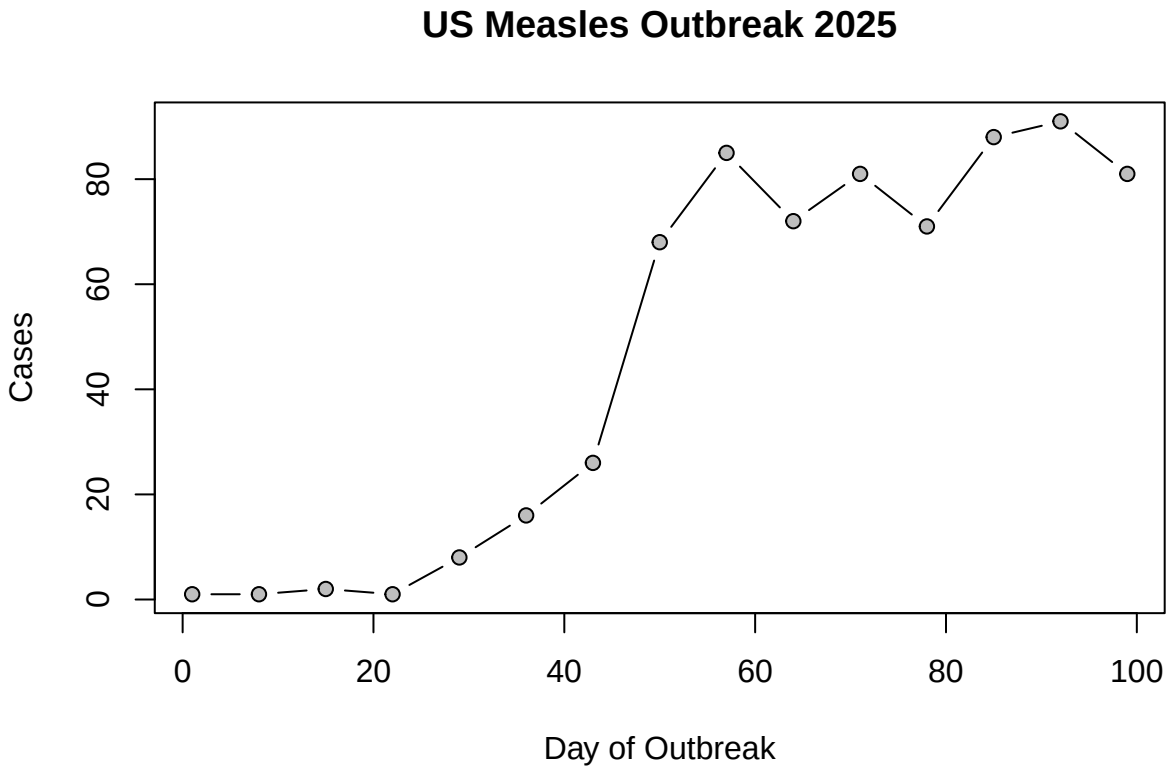
The US is in the middle of an outbreak of measles cases. Let's evaluate whether the population is growing exponentially. Load the data frame `US_Measles_Cases.csv`.

```
measles.df <- read.csv('US_Measles_Cases.csv')
str(measles.df)
```

```
## 'data.frame':    15 obs. of  4 variables:
##  $ week_start   : chr  "12/22/24" "12/29/24" "1/5/25" "1/12/25" ...
##  $ week_end     : chr  "12/28/24" "1/4/25" "1/11/25" "1/18/25" ...
##  $ outbreak_days: int  1 8 15 22 29 36 43 50 57 64 ...
##  $ cases        : int  1 1 2 1 8 16 26 68 85 72 ...
```

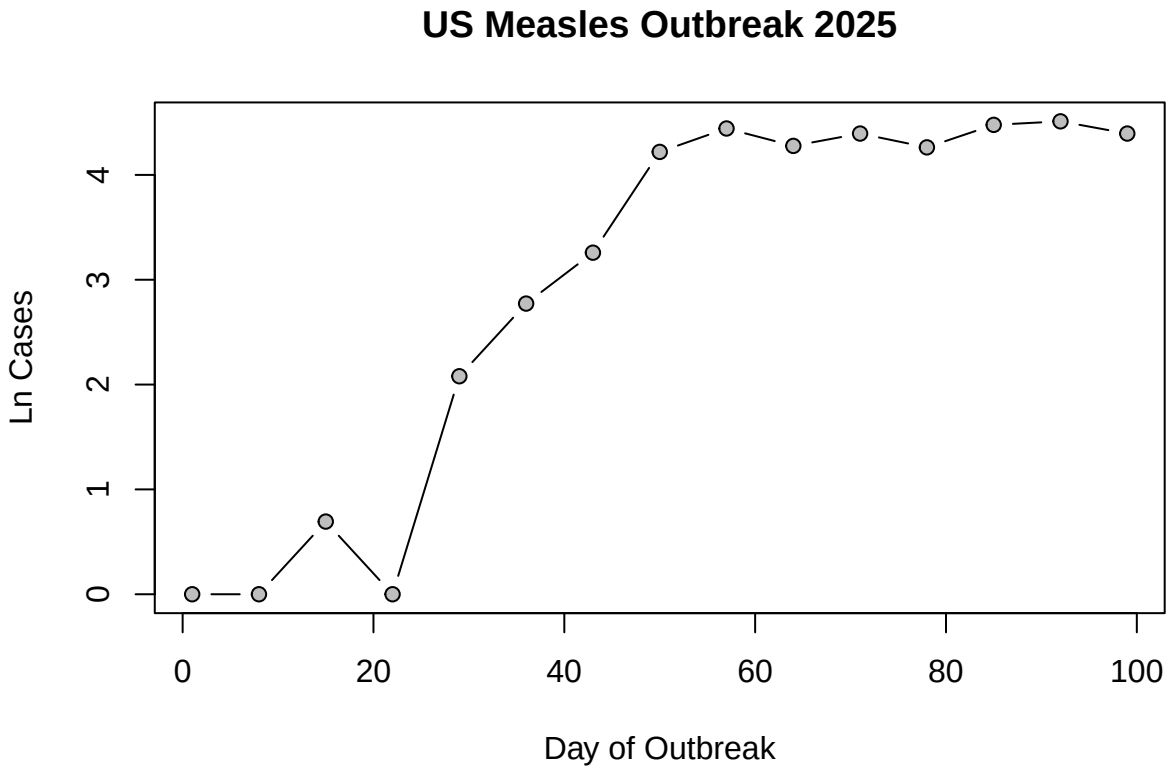
This shows reported measles cases by week starting in late December of last year. Let's take a look at the data.

```
plot(measles.df$outbreak_days, measles.df$cases, typ = 'b',
     pch = 21, bg = 'gray',
     xlab = 'Day of Outbreak', ylab = 'Cases',
     main = 'US Measles Outbreak 2025')
```



Clearly there is no straight line to fit through there. But if the number of cases grows exponentially, then we should see a straight line when we plot ln(Cases) against time.

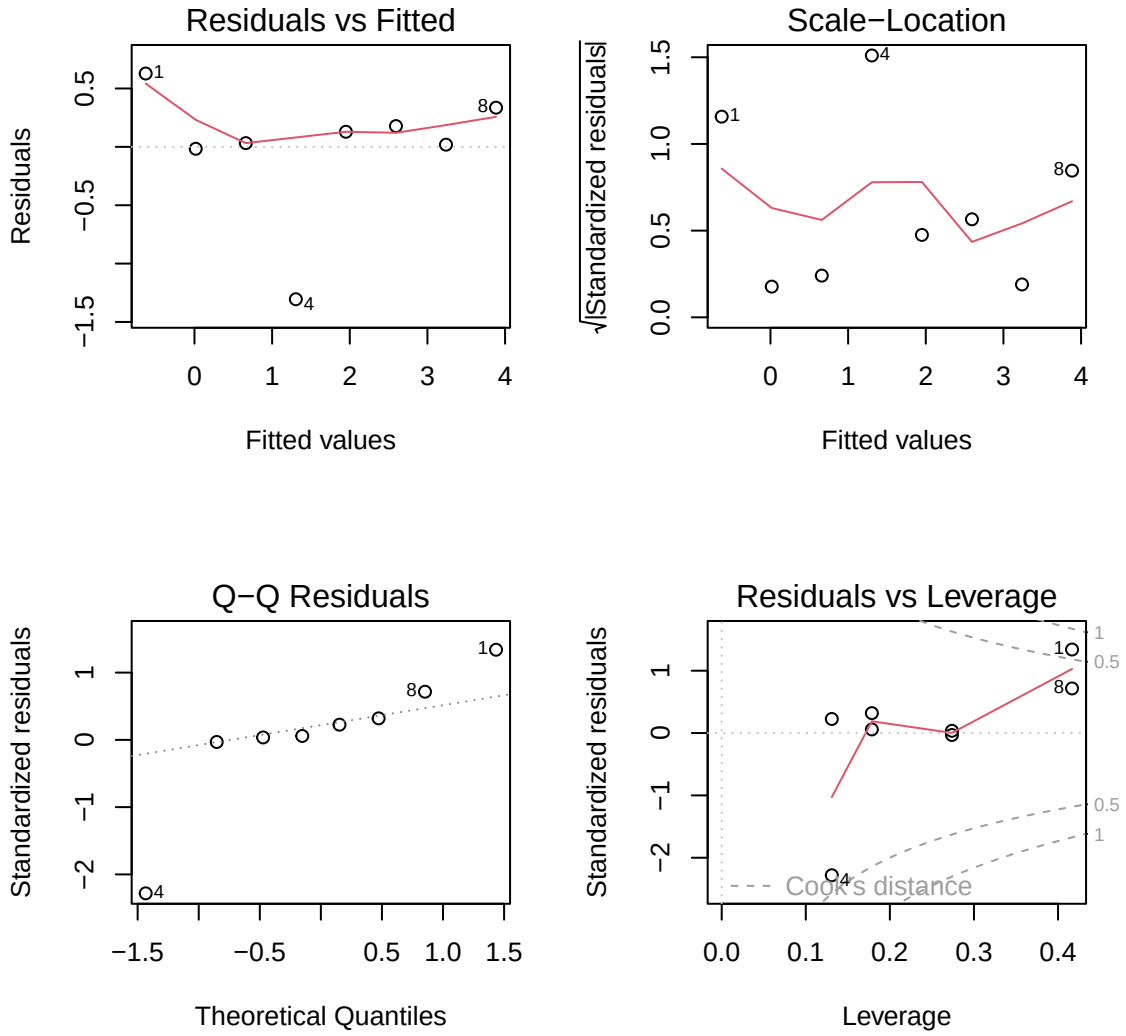
```
plot(measles.df$outbreak_days, log(measles.df$cases), typ = 'b',
     pch = 21, bg = 'gray',
     xlab = 'Day of Outbreak', ylab = 'Ln Cases',
     main = 'US Measles Outbreak 2025')
```



From this figure, I see a segment where cases rise close to a straight line and then taper off. This is a suggestion that there is exponential growth followed by something different around day 50. This is expected for measles, as it won't continue to expand exponentially because it runs out of susceptible hosts for the disease because many people are vaccinated or had a prior infection. Measles causes lifetime immunity once infected, so only people (almost always children) who have not yet been infected or haven't received a vaccine can get measles.

Let's look at the first 50 days to see whether we can fit an exponential growth model.

```
exp.measles.mdl <- lm(log(cases) ~ outbreak_days,
                      data = measles.df,
                      subset = (outbreak_days <=50))
par(mfcol = c(2,2))
plot(exp.measles.mdl)
```



Data point 4 looks problematic, but it has low leverage, so I think we are okay. This is a common problem with public health data sets. There are strange reporting errors, and I suspect this is one.

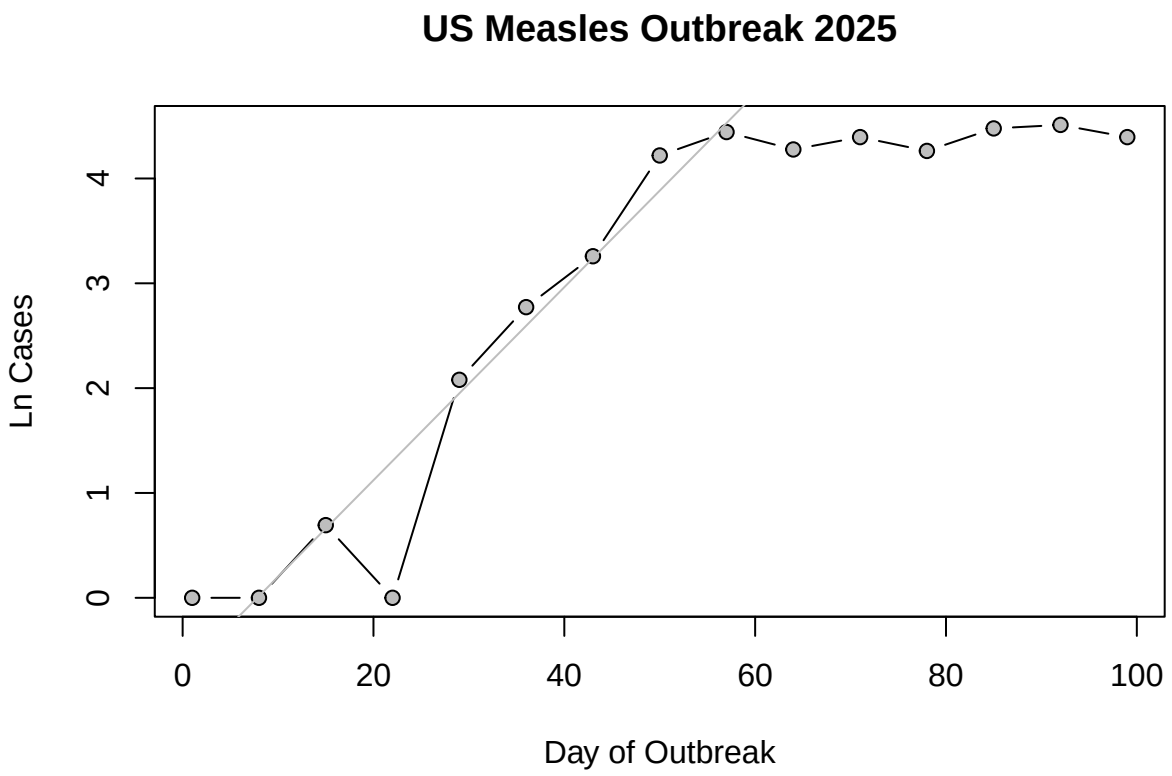
Let's see what this model predicts for the rate of growth of measles cases.

```
summary(exp.measles.mdl)
```

```
##
## Call:
## lm(formula = log(cases) ~ outbreak_days, data = measles.df, subset = (outbreak_days <=
##     50))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.30556  0.01000  0.08074  0.21731  0.62819
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.72027    0.40771  -1.767 0.127717
## outbreak_days  0.09208    0.01353   6.804 0.000494 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.614 on 6 degrees of freedom
## Multiple R-squared:  0.8853, Adjusted R-squared:  0.8661
## F-statistic: 46.29 on 1 and 6 DF,  p-value: 0.0004939
```

```
plot(measles.df$outbreak_days, log(measles.df$cases), typ = 'b',
     pch = 21, bg = 'gray',
     xlab = 'Day of Outbreak', ylab = 'Ln Cases',
     main = 'US Measles Outbreak 2025')
abline(exp.measles.mdl$coefficients, col = 'gray')
```



The slope is 0.092 which represents the exponential rate of growth in units of cases per day. But we also have a standard error and can get the confidence interval for this exponential rate of growth.

```
confint(exp.measles.mdl, parm = 'outbreak_days')
```

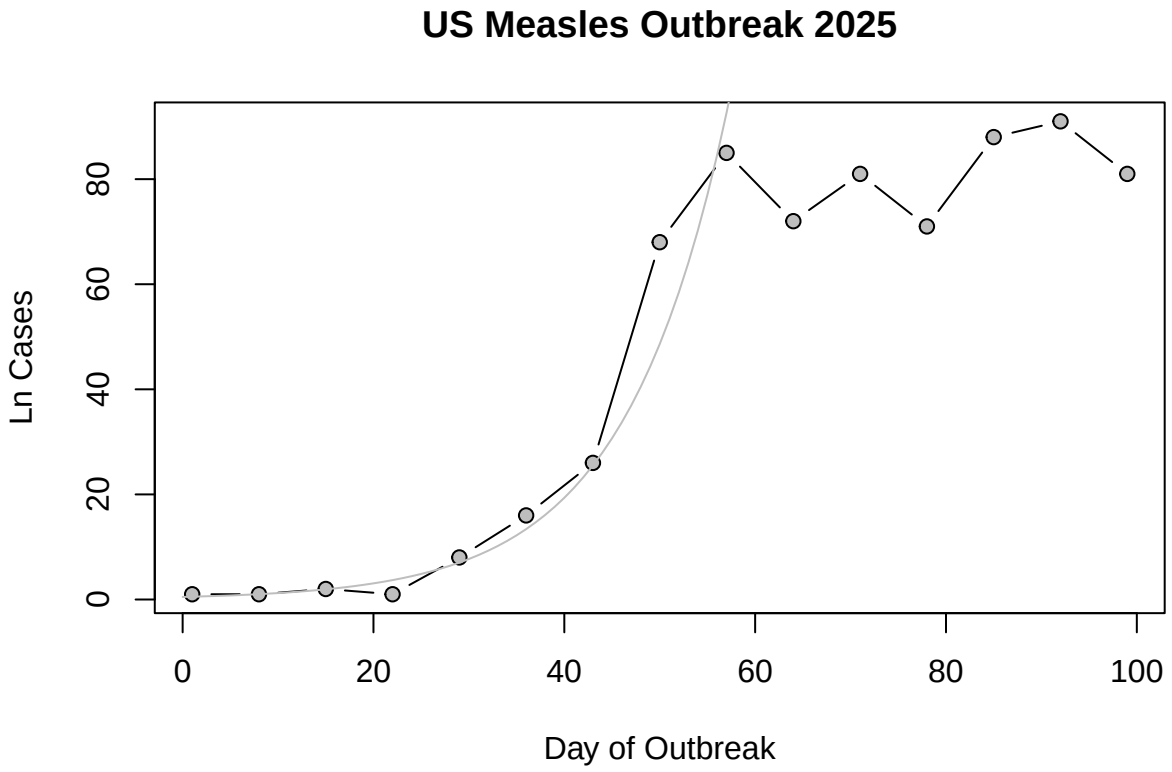
```
##                    2.5 %    97.5 %
## outbreak_days 0.05896661 0.1251994
```

This says that there is a 95% chance that the growth rate is in the interval [0.059, 0.125].

Let's plot this on the normal scale.

```
days <- 0:70
pred.cases <- exp(exp.measles.mdl$coefficients[1] + exp.measles.mdl$coefficients[2]*days)

plot(measles.df$outbreak_days, measles.df$cases, typ = 'b',
     pch = 21, bg = 'gray',
     xlab = 'Day of Outbreak', ylab = 'Ln Cases',
     main = 'US Measles Outbreak 2025')
lines(days, pred.cases,
      typ = 'l', col = 'gray')
```



If we want to do more and include the rest of the data, we need a more sophisticated model. But this works for now.

Checkpoint 7: What questions do you have about fitting exponential growth or power law models?

Multiple Regression

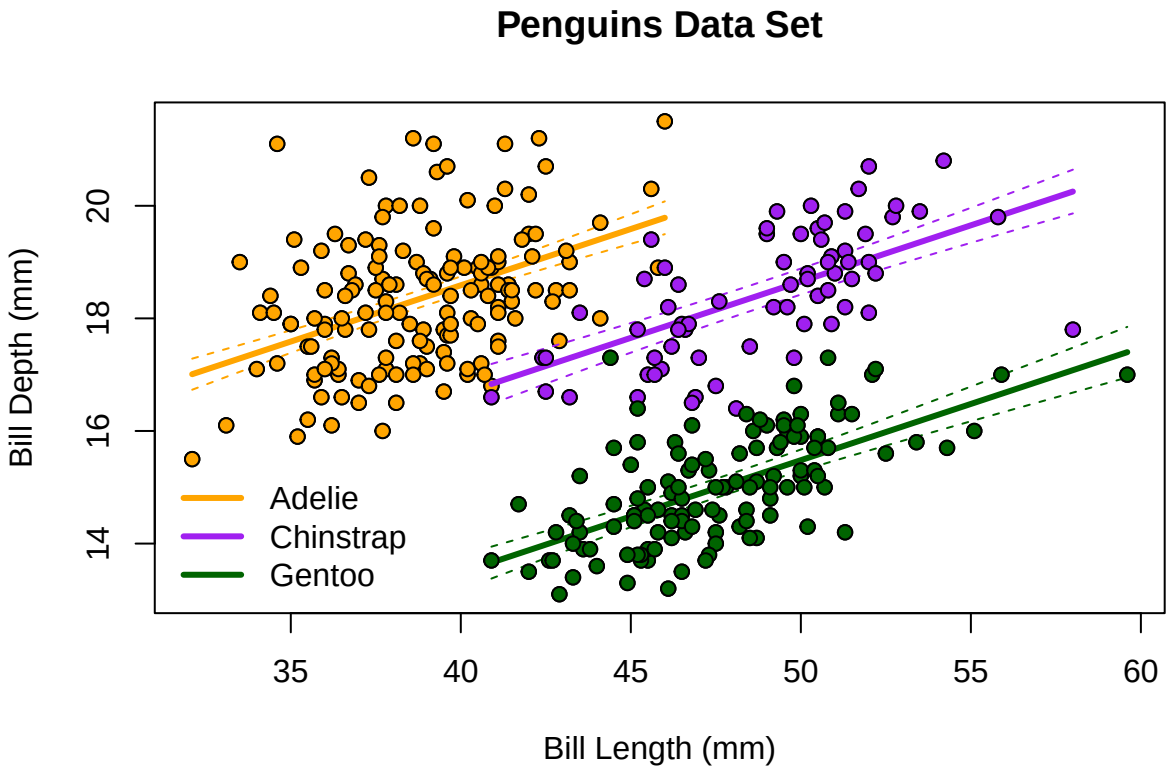
Last thing. Maybe we have questions about how bill shape differs for these three penguin species or how metabolic scaling relationships change by groups of organisms or about how growth rates change for different diseases. In all these cases, we need to compare the slopes and intercepts of fits of different groups. We can do that with multiple regression. This works just like 2-way ANOVA, but it's a 2-way regression. Here we predict, say, bill depth as a function of bill length AND the species. If we allow for interactions between these two factors, then we allow for the slopes of the lines to change for each species. We'll do this in class on Friday. Here is a preview.

```
penguin.mdl <- lm(bill_depth_mm ~ bill_length_mm*species, data = penguins)
summary(penguin.mdl)
```

```
##
## Call:
## lm(formula = bill_depth_mm ~ bill_length_mm * species, data = penguins)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.6574 -0.6675 -0.0524  0.5383  3.5032
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      11.40912    1.13812  10.025  < 2e-16 ***
```

```
## bill_length_mm                   0.17883    0.02927   6.110 2.76e-09 ***
## speciesChinstrap                -3.83998    2.05398  -1.870 0.062419 .
## speciesGentoo                   -6.15812    1.75451  -3.510 0.000509 ***
## bill_length_mm:speciesChinstrap  0.04338    0.04558   0.952 0.341895
## bill_length_mm:speciesGentoo     0.02601    0.04054   0.642 0.521590
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9548 on 336 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.7697, Adjusted R-squared:  0.7662
## F-statistic: 224.5 on 5 and 336 DF,  p-value: < 2.2e-16
```

```
##
## Call:
## lm(formula = bill_depth_mm ~ bill_length_mm + species, data = penguins)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4529 -0.6864 -0.0508  0.5519  3.5915
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      10.59218    0.68302  15.508  < 2e-16 ***
## bill_length_mm    0.19989    0.01749  11.427  < 2e-16 ***
## speciesChinstrap -1.93319    0.22416  -8.624 2.55e-16 ***
## speciesGentoo    -5.10602    0.19142 -26.674  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9533 on 338 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.769,  Adjusted R-squared:  0.7669
## F-statistic: 375.1 on 3 and 338 DF,  p-value: < 2.2e-16
```



References

Horst, Allison M, Alison Presmanes Hill, and Kristen B Gorman. 2020. *Allisonhorst/Palmerpenguins: V0.1.0.* Zenodo. https://doi.org/10.5281/ZENODO.3960218.