

# Lecture 33: Inference for Regression

## Chapter 23 (continued)

Corinne Riddell (Instructor: Alan Hubbard)

November 15, 2023

### Recap

- Last class we covered the assumptions necessary to perform linear regression
- Most of these assumptions can be investigated using plots of the residuals
- One of the assumptions could not be checked using plots. Which assumption was that?

### Recap on notation

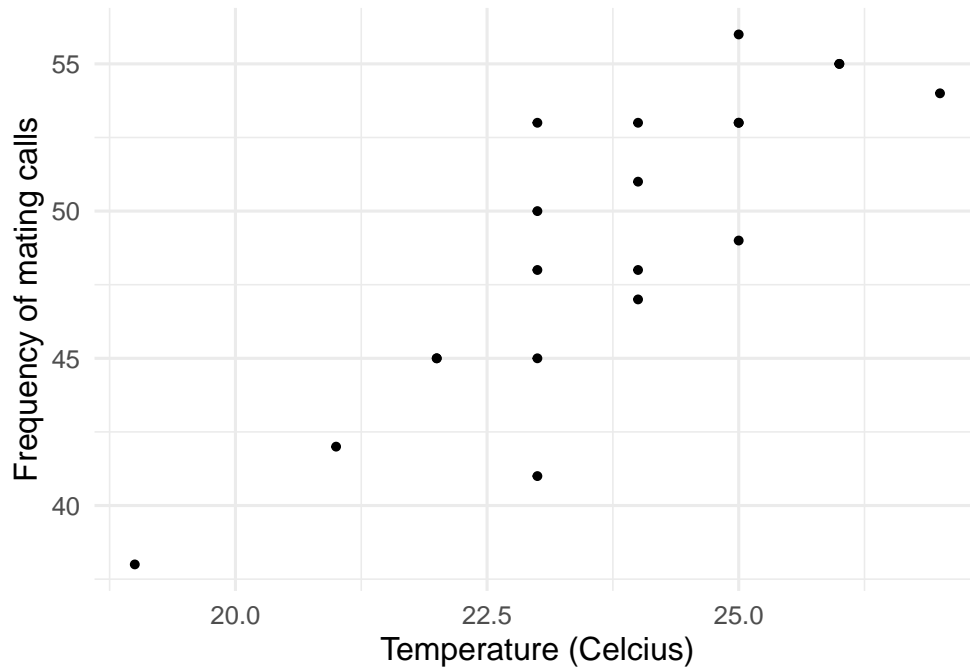
Term	Population	Sample
Intercept	$a$ or $\alpha$	$\hat{a}$
Slope	$b$ or $\beta$	$\hat{b}$
Residual	$e$	$\hat{e}$

### Learning objectives for today

- Conduct a hypothesis test for the slope parameter.
  - Define the test statistic
  - Know how to calculate the test using R output after running `lm()`
- Create a 95% **confidence interval** for the slope parameter
- Create a 95% **confidence interval** for the predicted value, and a 95% **prediction interval** for an individual value. Know how to explain the difference between the two
- Describe why the hypothesis test for correlation is the same (i.e., gives the same results) as the hypothesis test of the slope parameter

### Frog data

Recall the frog data from last class on temperature and the frequency of mating calls:



Use `lm()` + broom functions to look at your linear model

- `tidy(your_lm)`: Presents the output of the linear model in a tidy way
- `glance(your_lm)`: Takes a quick (one line) look at the fit statistics.
- `augment(your_lm)`: Creates an augmented data frame that contains a column for the fitted y-values ( $\hat{y}$ ) and the residuals ( $\hat{e} = y - \hat{y}$ ) among other columns (you don't need to worry about the other columns that are added)

Know these functions, what they do, and how to use them.

Use `lm()` + broom functions to look at your linear model

```
frog_lm <- lm(formula = freq ~ temp, data = frog_data)
tidy(frog_lm)
```

```
## # A tibble: 2 x 5
##   term      estimate std.error statistic    p.value
##   <chr>      <dbl>    <dbl>    <dbl>    <dbl>
## 1 (Intercept) -6.19      8.24    -0.751  0.462
## 2 temp         2.33     0.347     6.72  0.00000266
```

```
glance(frog_lm)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic    p.value    df logLik   AIC   BIC
##   <dbl>      <dbl> <dbl>    <dbl>    <dbl> <dbl> <dbl> <dbl> <dbl>
## 1   0.715      0.699   2.82    45.2  0.00000266     1  -48.1  102.  105.
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

```
frog_data_aug <- augment(frog_lm)
head(frog_data_aug)
```

```
## # A tibble: 6 x 8
##   freq temp .fitted .resid .hat .sigma .cooksd .std.resid
##   <dbl> <dbl>   <dbl>   <dbl> <dbl> <dbl>   <dbl>   <dbl>
## 1    38    19    38.1 -0.0952 0.384    2.90 0.000575 -0.0430
## 2    42    21    42.8 -0.757  0.160    2.90 0.00816  -0.293
## 3    45    22    45.1 -0.0876 0.0937    2.90 0.0000550 -0.0326
## 4    45    22    45.1 -0.0876 0.0937    2.90 0.0000550 -0.0326
## 5    41    23    47.4 -6.42   0.0574    2.42 0.167    -2.34
## 6    45    23    47.4 -2.42   0.0574    2.84 0.0237    -0.883
```

- Only need to pay attention to the added columns `.fitted` and `.resid`

### New terminology: SSE

Sum of squared estimates of error (**SSE**):  $SSE = \sum_i^n (y_i - \hat{y}_i)^2$

- The SSE is the summation of the squared distance between each individual's  $y$  value and the fitted (or predicted) value based on the line of best fit
- The higher the **SSE** the worse the model fits the data

We are interested in knowing the average spread of the squared residual distances. Because small spread would indicate a good fitting model. To measure this, we calculate the **regression standard error**

### New terminology: Regression standard error

- The regression standard error can be calculated as:  $s = \sqrt{\frac{1}{n-2} \times SSE}$
- This can also be written as:

$$s = \sqrt{\frac{1}{n-2} \sum_{i=1}^n \hat{e}^2}$$

or:

$$s = \sqrt{\frac{1}{n-2} \sum_{i=1}^n (y - \hat{y})^2}$$

- We divide by  $n - 2$  rather than  $n$  because this produces an unbiased estimate of  $s$ .
- A good-fitting model will have a low regression standard error because  $\hat{y}$  will be close to  $y$ .
- Look at  $s$  after running a linear model to assess the model's fit to the data.
- $s$  is on the same scale as  $y$  (i.e., they have the same units).
- `glance(your_lm)` prints  $s$ , which is denoted by `sigma`.

### `glance()` to view the regression standard error

```
glance(frog_lm)
```

```
## # A tibble: 1 x 12
##   r.squared adj.r.squared sigma statistic    p.value    df logLik   AIC   BIC
##   <dbl>      <dbl> <dbl>    <dbl>    <dbl> <dbl> <dbl> <dbl> <dbl>
## 1    0.715      0.699  2.82     45.2 0.00000266     1 -48.1  102.  105.
## # i 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>
```

- $\sigma = 2.82$ . This is the residual estimated standard deviation (or sometimes called residual standard error).

## Another way to contextualize the regression standard error

You can compute a five number summary on the residuals using the augmented data frame:

```
frog_data_aug %>% summarise(min_resid = min(.resid),
                             q25_resid = quantile(.resid, 0.25),
                             mean_resid = mean(.resid),
                             q75_resid = quantile(.resid, 0.75),
                             max_resid = max(.resid))
```

```
## # A tibble: 1 x 5
##   min_resid q25_resid mean_resid q75_resid max_resid
##   <dbl>      <dbl>      <dbl>    <dbl>    <dbl>
## 1    -6.42     -1.92   -8.88e-15     1.00     5.58
```

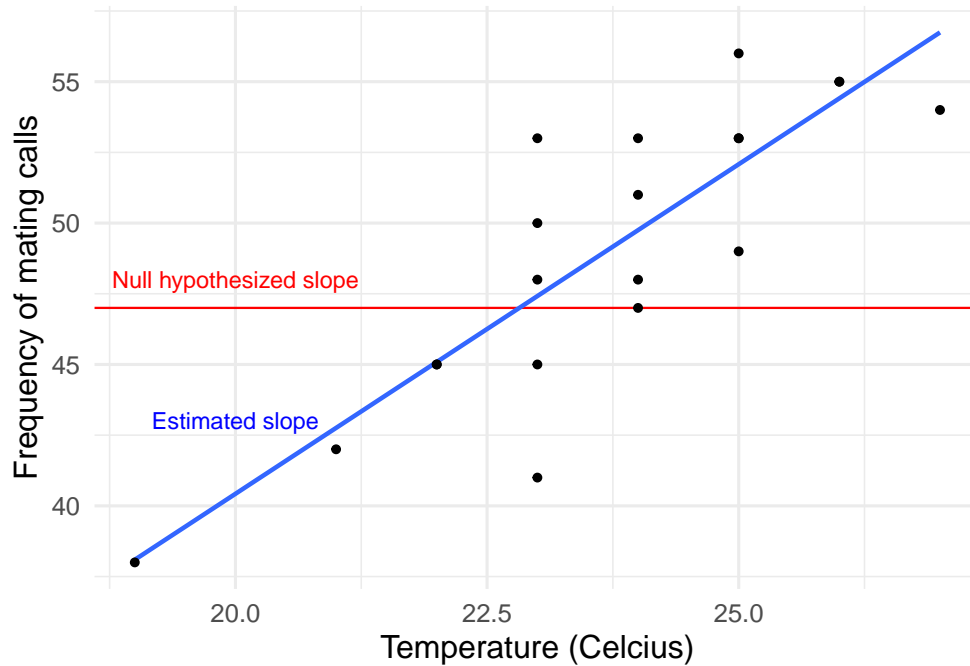
- The smallest residual is -6.42 and the largest is 5.58.
- The IQR for the residuals goes from -1.92 to 1.00.
- The mean residual is very close to 0.
- **The residual standard error (2.82) is capturing the standard deviation of this distribution of residuals.**

## Hypothesis testing for regression

- The regression standard error is used as part of the test statistic for the slope coefficient
- In this test, we'd like to know whether the slope is different from 0. That is  $H_0 : b = 0$  and  $H_A : b \neq 0$  for a two-sided test.

## Frog data showing the estimates slope vs. null hypothesis slope

```
## 'geom_smooth()' using formula = 'y ~ x'
```



### Hypothesis testing for regression

What are the null and alternative hypotheses?

### Hypothesis testing for regression

$H_0 : b = 0$  (i.e., There is no association between temperature and the frequency of mating calls)

$H_a : b \neq 0$  (i.e., There is an association between temperature and the frequency of mating calls)

### Hypothesis testing for regression

$H_0 : b = 0$  (i.e., There is no association between temperature and the frequency of mating calls)

$H_a : b \neq 0$  (i.e., There is an association between temperature and the frequency of mating calls)

To test the null hypothesis, the t-test statistic is:

$$t = \frac{\hat{b}}{SE_b}$$

where  $SE_b = \frac{s}{\sqrt{\sum (x - \bar{x})^2}}$  and  $s = \sqrt{\frac{1}{n-2} \sum_{i=1}^n (y - \hat{y})^2}$

We will use R to compute the test statistic,  $SE_b$  and  $s$ . Be sure you know where  $SE_b$ ,  $s$ , and  $\hat{b}$  can be found using the R output and which functions to use to find them.

### Two-sided hypothesis testing for regression using `tidy()`

```
tidy(frog_lm)
```

```
## # A tibble: 2 x 5
##   term          estimate std.error statistic    p.value
##   <chr>         <dbl>    <dbl>    <dbl>    <dbl>
## 1 (Intercept)   -6.19      8.24    -0.751  0.462
## 2 temp          2.33     0.347     6.72  0.00000266
```

Focus on the row of data for `temp`:

- `estimate` is the estimated slope coefficient  $\hat{b}$ : 2.33
- `std.error` is the standard error,  $SE_b = 0.347$
- `statistic` is the t-test statistic:  $\frac{\hat{b}}{SE_b} = 2.330816/0.3467893 = 6.72$
- The test has  $n - 2$  degrees of freedom, where  $n$  is the number of observations in the data frame.
- `p-value` is the p-value corresponding to the test

```
pt(q = 6.7211302, df = 18, lower.tail = F)*2
```

```
## [1] 2.663401e-06
```

### Confidence intervals for the regression coefficient

We can also use the output from `tidy(your_lm)` to create a 95% confidence interval for the slope coefficient.

estimate  $\pm$  margin of error

$$\hat{b} \pm t^* SE_b$$

Where  $t^*$  is the critical value for the t distribution with  $n - 2$  degrees of freedom with area C (e.g., 95%) between  $-t^*$  and  $t^*$ .

### Confidence intervals for the regression coefficient

```
tidy(frog_lm)
```

```
## # A tibble: 2 x 5
##   term          estimate std.error statistic    p.value
##   <chr>         <dbl>    <dbl>    <dbl>    <dbl>
## 1 (Intercept)   -6.19      8.24    -0.751  0.462
## 2 temp          2.33     0.347     6.72  0.00000266
```

First, find the critical value  $t^*$ , such that 95% of the area is between  $t^*$  and  $-t^*$ :

```
t_star <- qt(p = 0.975, df = 18)
t_star
```

```
## [1] 2.100922
```

95% CI:

$$2.330816 \pm t \cdot 0.3467893$$

$$2.330816 \pm 2.100922 \times 0.3467893$$

95% CI: 1.60 to 3.06

Interpretation: The estimate for the slope coefficient is 2.33 (95% CI: 1.60 to 3.06). If we had gathered 100 random samples and ran the same regression on each of them and used the same method to compute the 95% CI, 95 out of the 100 intervals would contain the true value of  $\beta$  in the interval.

### Code for the confidence interval

Alternatively you can use the following code to calculate the confidence interval in R:

```
confint(frog_lm, "temp")
```

```
##           2.5 %    97.5 %  
## temp 1.602239 3.059393
```

### Inference for prediction

- So far we've learned only about inference for the slope coefficient  $b$ .
- But what if you wanted to use the model to make a prediction?
- We already know how to predict the **average** number of mating calls corresponding to a specific  $x$  value, say of 21 degrees celsius:

$$\hat{y} = -6.190332 + 2.330816x$$

$$\hat{y} = -6.190332 + 2.330816(21) = 42.8$$

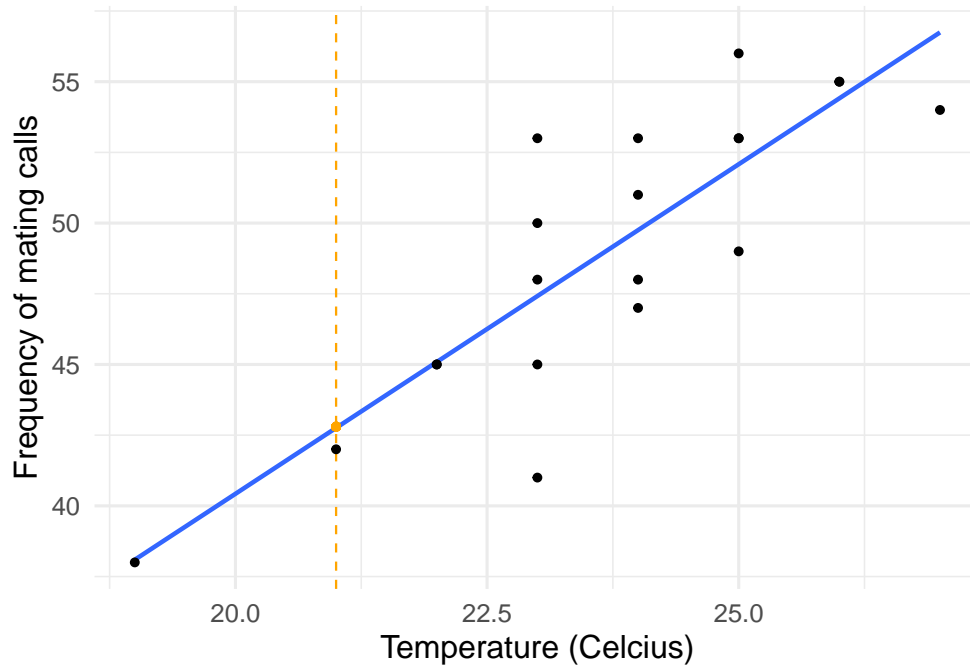
We expect 42.8 mating calls, so 43 mating calls (rounding because the outcome is a discrete variable) when the temperature is 21 degrees celsius.

### Inference for prediction

How do we make a confidence interval for this prediction?

- It depends on whether you want to make a CI for the **average response** or for an **individual's response**

```
## 'geom_smooth()' using formula = 'y ~ x'
```



### Inference for prediction of average vs. individual response, visualized

If you want to make inference for the **mean response**  $\mu_y$  when  $x$  takes the value  $x^*$  ( $x^*=21$  in our example):

$$\hat{y} \pm t * SE_{\hat{\mu}}, \text{ where } SE_{\hat{\mu}} = s \sqrt{\frac{1}{n} + \frac{(x^* - \bar{x})^2}{\sum (x - \bar{x})^2}}$$

If you want to make inference for a **single observation**  $y$  when  $x$  takes the value  $x^*$  ( $x^*=21$  in our example):

$$\hat{y} \pm t * SE_{\hat{y}}, \text{ where } SE_{\hat{y}} = s \sqrt{1 + \frac{1}{n} + \frac{(x^* - \bar{x})^2}{\sum (x - \bar{x})^2}}$$

Corresponding R code for prediction and confidence interval:

```
# specify the value of the explanatory variable for which you want the prediction:
newdata = data.frame(temp = 21)
```

```
# use `predict()` to make prediction and confidence intervals
prediction_interval <- predict(frog_lm, newdata, interval = "predict")
prediction_interval
```

```
##          fit          lwr          upr
## 1 42.7568 36.37187 49.14173
```

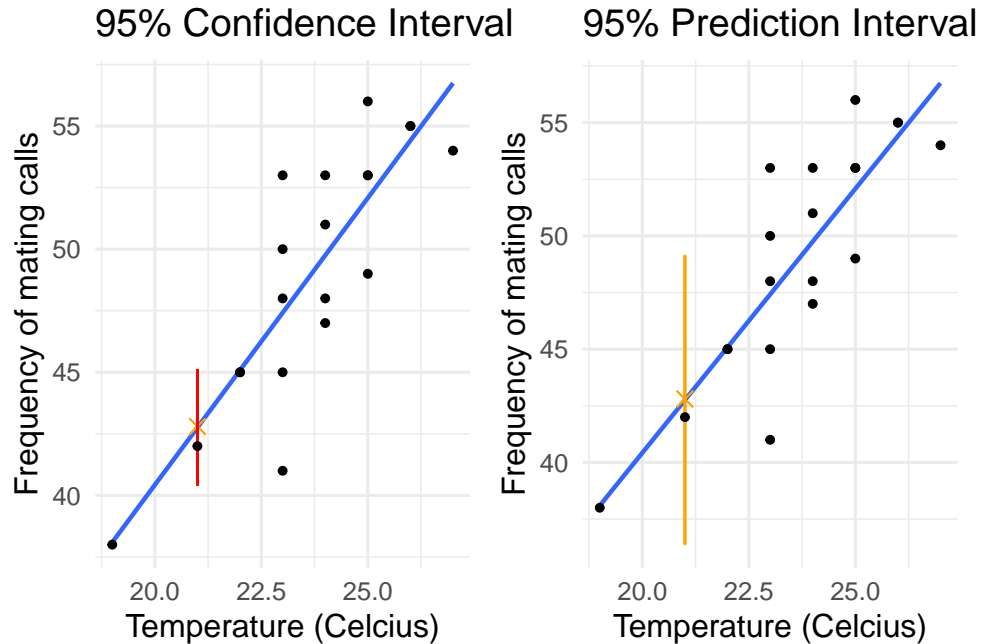
```
confidence_interval <- predict(frog_lm, newdata, interval = "confidence")
confidence_interval
```

```
##          fit          lwr          upr
## 1 42.7568 40.38472 45.12887
```



## Inference for prediction, visualized

```
## 'geom_smooth()' using formula = 'y ~ x'  
## 'geom_smooth()' using formula = 'y ~ x'
```



- Why is the prediction interval *wider* than the confidence interval?

## Test for a lack of correlation

- A lack of correlation occurs if and only if there is no association between the explanatory and response variables
- Thus, if your hypothesis test does not reject the null ( $b = 0$ ) then this also implies that you would not reject the hypothesis of no correlation between  $x$  and  $y$ .
- Can you describe the steps of a permutation test to test for a lack of correlation?
- Don't worry about the book section on this topic "Testing lack of correlation"