Linear Regression - predictions

NRES 710

Last compiled: 2024-09-16

## Predictions

Using our statistical models (i.e., our statistical analyses) to make predictions is a really important component of science. Perhaps the ultimate goal. We are seeking to understand nature and make predictive models for how it works! If you can’t make predictions from your statistical model, then we might argue that your model needs some work.

**Quality of science**

* Level 1: is there a difference between groups? P-values. Bare minimum.
* Level 2: what is the difference between groups? Effect sizes. Next step up.
* Level 3: measure effects and make predictions from them.

For me personally, I am a conservation biologist who studies wildlife populations. I try to measure the demographic rates of populations: what are the survival and reproductive rate of animals in a population each year, and how does survival and reproduction vary by age? I then try to use those models for age-structured demographic rates to make predictions about how populations will grow or decline in size. This is often called ‘population viability analysis’. My ultimate goal is to generate predictive models for whether populations will persist or not.

### Simple predictions

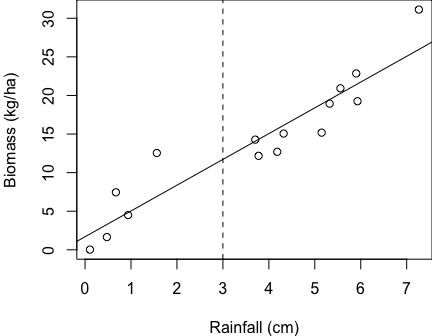
Ultimately, prediction comes back to our linear model:

We want to make predictions for what Y will be *for a given X*. To make predictions, we can:

* Fit our regression model
* Measure betas
* Get some new X-variable data
* Solve for Y! This is making predictions.

For example, let’s say we have results from a regression model explaining how biomass (kg/ha) changes as a consequence of rainfall (cm). Our results are:

* **beta0 = 1.55 and beta1 = 2.05**
* We want to use the model to predict what biomass might be when rainfall = 5 cm. We will ignore error, for now, because usually when we make predictions we want to know what the response will be *on average*.
* **Y = 1.55 + 2.05 \* 5**
* **= 1.55 + 10.25**
* **= 11.80 kg/ha –> 5 cm**

We would predict to observe 11.80 kg/ha biomass, on average, when an area gets 5 cm of rain.

### When can we make predictions?

Let’s examine the biomass example with a scatterplot:

We don’t have any data around when rainfall ~ 3 cm. Let’s say you want to know how much biomass you might expect at the vertical dashed line – 3 cm rainfall. Can we do that? **Yes!**

**Interpolation (good) – making predictions within the range of observed data.**

Let’s say you want to know how much biomass you might expect when there is 15 cm of rainfall. Can we do that? **Yes.**

**Extrapolation (be careful) – making predictions outside the range of observed data.**

* It’s not a bad practice, but we need to be **very clear to ourselves and our readers** that we are extrapolating. We don’t know if the relationship we observed *changes* outside of the observed data. Maybe it becomes nonlinear! We need to be honest about potential limitations.

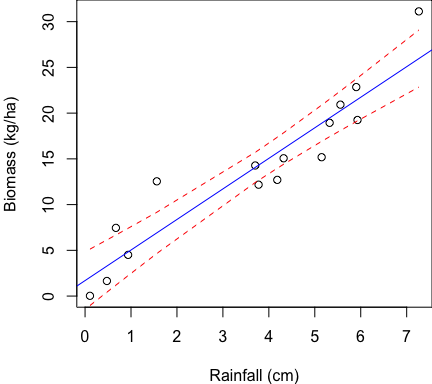
### Uncertainty

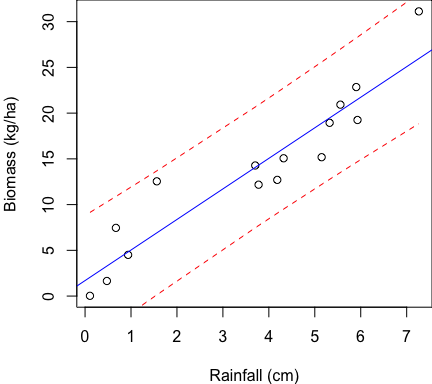
Anytime we provide an estimate of truth with out statistical model, we provide a measure of uncertainty. If we estimate a slope, difference between groups, whatever – we always provide estimates of confidence intervals. We always convey how certain we are of those estimates.

We must do the same for predictions. Predictions tell us what the predicted value is for the average regression line. *But how certain are we about that line*?

In the rainfall prediction example above, we aren’t going to provide confidence intervals, but will instead provide **prediction intervals**.

**Confidence intervals** – 95% of all such intervals contain the true value; **a measure of uncertainty in the *estimate***.

* The thing to know is that estimates deal with averages.
* For example, when we estimate biomass at 3 cm rainfall, we take an average.
* Individual observations can deviate above and below this average, depending on the noise in our system. But the mean estimate is the regression line.
* Confidence intervals usually fall inside the individual data.

**Prediction intervals – a measure of uncertainty in the individual outcomes**

* When we make predictions, we make making predictions about *individual outcomes*, and there will be more uncertainty.
* Prediction intervals are outside of most data. For example:

A prediction interval will capture most (95%) of the data.

**Q:** What do you notice about the shape of how I drew the confidence intervals and prediction intervals?

They are somewhat **pinched** in the middle and they **flare** at the edges. This is because we have more data in the middle, we gives us more certainty about in what the true estimate is and individual outcomes are.

However, at the tails we have less data, and thus we have less certainty about the estimate and individual outcomes. And if we **extrapolate**, the uncertainty will be larger.

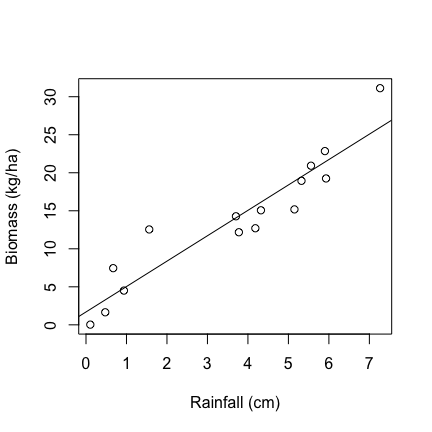
### Prediction summary

1. Predictions are basic math
2. Difference between interpolation and extrapolation, and differences in your confidence of these estimates
3. Difference between prediction and confidence intervals

## Making predictions

Let’s show you how to make predictions in R, including with **prediction intervals**. We will have R do this for us!

Before we make predictions, we first have to run our linear regression analysis. Let’s use the results from the biomass x rainfall data regression analysis.

# Read in the biomass data again, if necessary  
datum <- read.csv("lecture\_6\_biomass\_data.csv")  
  
# Plot  
plot(biomass ~ rainfall, data = datum, xlab = "Rainfall (cm)", ylab = "Biomass (kg/ha)")  
  
# Regression  
results <- lm(biomass ~ rainfall, data = datum)  
abline(results)

### Single-value predictions

Let’s say we want to know what the biomass is predicted to be when rainfall = 5 cm. We would simply do this using the linear model results:

# Examine the results   
summary(results)

##   
## Call:  
## lm(formula = biomass ~ rainfall, data = datum)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -3.712 -2.077 -0.534 1.054 5.634   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.693 1.458 1.16 0.27   
## rainfall 3.340 0.339 9.84 2.2e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.96 on 13 degrees of freedom  
## Multiple R-squared: 0.882, Adjusted R-squared: 0.873   
## F-statistic: 96.9 on 1 and 13 DF, p-value: 2.16e-07

# Make a prediction for when rainfall = 3 cm  
1.69 + 3.34 \* 3

## [1] 11.71

12.5 kg/ha biomass when rainfall is 3 cm!

The most common (and easy) way to make predictions in R is using the ‘predict()’ function. We will use ‘predict.lm()’.

# Examine the help file  
help(predict.lm)

Things that ‘predict.lm()’ needs:

* Your regression! Results from your regression.
* ‘newdata’ – these X-values that you want to make predictions for; interpolation, or extrapolation.
  + If you leave this out, it will make predictions for the X-values that were originally used. This may be fine. But if you want to do interpolation or extrapolation, your exisiting X-data won’t be able to do those things.
  + This ‘newdata’ has to have an X-column that matches the X-column from your original data.
* Interval – whether or not you want ‘confidence’ or ‘prediction’ intervals; you need to include *interval = “prediction”*.
* Other stuff you can ignore.

Let’s try to validate my prediction for biomass when rainfall = 5 cm.

# Create predict data  
datumPredict <- data.frame(rainfall = 3)  
datumPredict  
  
# Create a new object called predictions  
predictions <- predict(results, datumPredict, interval = "prediction")  
predictions

## fit lwr upr  
## 1 11.71 5.096 18.33

Three outputs: fit, lwr, and upr.

* fit – this is our prediction! And it matches pretty closely to what we calculated by ‘hand’ with our linear model.
* instead of intervals, we get limits:
  + lower prediction limit (lwr) –
  + upper prediction limit (upr) –
* if we want the prediction interval, we need to do some math. (Upper - Lower)/2 or Upper - Fit:

predictions[1,"upr"] - predictions[1,"fit"]

## [1] 6.619

The 95% prediction interval is +/-6.62.

### Predictions for data

Normally we will want to make predictions over a good interval range that encompasses our X-data. This might help us fill in some holes, like in the first example I did on the board.

# Let's look at information about our data  
summary(datum)

## X rainfall biomass   
## Min. : 1.0 Min. :0.107 Min. : 0.028   
## 1st Qu.: 4.5 1st Qu.:1.249 1st Qu.: 9.816   
## Median : 8.0 Median :4.183 Median :14.277   
## Mean : 8.0 Mean :3.658 Mean :13.911   
## 3rd Qu.:11.5 3rd Qu.:5.440 3rd Qu.:19.095   
## Max. :15.0 Max. :7.265 Max. :31.123

# Minimum rainfall was closer to 0, maximum was closer to 10.   
# Let's make predictions between 0 -- 10 cm of rainfall  
  
# And let's try to make 20 predictions across this interval:  
# 10 cm / 20 predictions = a prediction every 0.5 cm  
xSequence <- seq(from = 0, to = 10, by = 0.5)  
xSequence

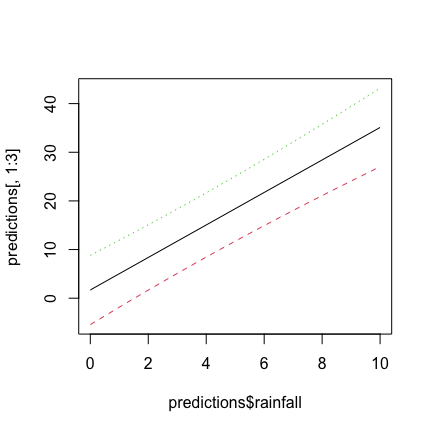
## [1] 0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0 5.5 6.0 6.5 7.0  
## [16] 7.5 8.0 8.5 9.0 9.5 10.0

# Turn this into a dataframe with a column called 'rainfall'  
datumPredict <- data.frame(rainfall = xSequence)  
datumPredict  
  
# Make the predictions  
predictions <- predict(results, datumPredict, interval = "prediction")  
  
# Examine predictions  
head(predictions)

## fit lwr upr  
## 1 1.693 -5.43182 8.818  
## 2 3.364 -3.63189 10.359  
## 3 5.034 -1.84906 11.917  
## 4 6.704 -0.08418 13.492  
## 5 8.374 1.66199 15.086  
## 6 10.044 3.38881 16.700

We now have a whole sequence of predictions, as well as lower and upper ‘prediction intervals’!

An annoying feature here is that we don’t really have X-value data in here; the ‘rainfall’ data.

# Make it a dataframe to make it easier to add data to it.  
predictions <- as.data.frame(predictions)  
  
# Add a new column called 'rainfall'  
predictions$rainfall <- xSequence  
  
# Examine  
head(predictions)

Pretty cool!?

Last thing I want to show you: plotting the prediction intervals over the range of X-values. I will show you two ways.

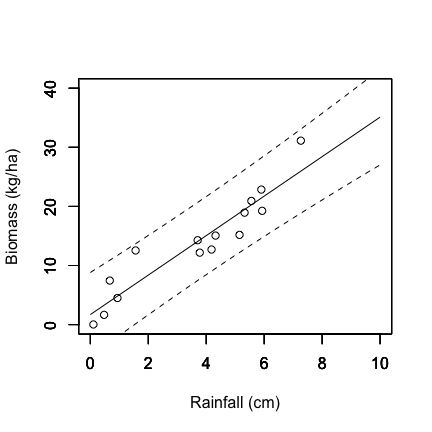
1. using ‘matplot’, which plots multiple ‘matrices’ of data at the same time. Simple
2. A more complicated way using ‘base R’ graphics

# Very simple way to plot the prediction intervals  
matplot(predictions$rainfall, predictions[,1:3], type = "l")

This graph ain’t too pretty.

**Q:** What’s this graph missing? The original data.

# Plot the data, but this time specify what the x- and y-limits are.  
# This is important to make sure everything will fit on here.  
plot(biomass ~ rainfall, data = datum, xlim = c(0, 10), ylim = c(0, 40), xlab = "Rainfall (cm)", ylab = "Biomass (kg/ha)")  
  
# But now we have to add other data onto here, without erasing the original data.  
# Specify the parameter: new = TRUE  
par(new = TRUE)  
plot(fit ~ rainfall, data = predictions, type = "l", xlim = c(0, 10), ylim = c(0, 40), ylab = "", xlab = "")  
par(new = TRUE)  
plot(lwr ~ rainfall, data = predictions, type = "l", xlim = c(0, 10), ylim = c(0, 40), ylab = "", xlab = "", lty=2)  
par(new = TRUE)  
plot(upr ~ rainfall, data = predictions, type = "l", xlim = c(0, 10), ylim = c(0, 40), ylab = "", xlab = "", lty=2)



This is more useful, I would say!

That being said, there are definitely better approaches to graphing predictions out there, likely using ‘ggplot2’ package.

Note: the prediction intervals are *slightly* curved, although it may be hard to see.

**Final thought**: This approach to prediction will be one that we can use for the rest of the semester with all of the more complicated analytical models we will learn!

[–go to next lecture–](lecture_7.html)