I put in a lot of efforts into bringing practical challenges, examples from real life, and a lot of demonstrations of statistical theory with R. This post is an example for how I’ve been using R code (and specifically Shiny apps) to demonstrate statistical theory, concepts and provide intuition.

**What’s the difference between confidence and prediction intervals?**

Last week I taught multiple linear regression, and I noticed that students have a hard time comprehending the difference between confidence intervals and prediction intervals. The former being an interval for the model (i.e., interval for the underlying model), and the latter being an interval for a noval observation.

As the sample size increases, our uncertainty of the model’s parameters decreases, but the uncertainty in the value of a new observation, \(y\_0\) is associated with variance of \(Y\) (the random variable from which \(y\_0\) is drawn). Hence, it has a lower bound, based on that variance.

In R, you can get a prediction or a confidence interval by using either

predict(object, newdata, interval = "prediction")

Or

predict(object, newdata, interval = "confidence")

For a prediction or for a confidence interval, respectively.

To help me illustrate the differences between the two, I decided to build a small Shiny web app. It shows the differences between confidence intervals, prediction intervals, the regression fit, and the actual (original) model.

The app is available [here](https://sarid.shinyapps.io/intervals_demo/), and the source code is available on [github](https://github.com/adisarid/prediction_confidence_intervals_demo).

With this app you can choose three types of models to demonstrate. Simple linear regression, and regression with a twist (\(\log\) transformation on the \(y\) or \(\sin\) transformation on the \(x\):

* Linear model \(y = a + bx + \epsilon\)
* Log-linear model \(\log(y)=a+bx+\epsilon\)
* Sine \(y = a + b\sin(x) + \epsilon\)

All the models are based on simple linear regression (lm function), for the latter two models with either a log or sin transformation.

The app allows you to play around with various values such as the \(x\) range, the model’s parameters (\(a\) and \(b\)), the error’s standard deviation (\(\epsilon\)), and show or hide any of the following elements, on the chart:

* The original function (i.e., the original model)
* The sampled points
* The confidence interval
* The prediction interval
* The model’s fit

Feel free to share the app or the app’s code. As mentioned above, the **source code for the app** is available here: <https://github.com/adisarid/prediction_confidence_intervals_demo>.

Here’s an example for what the app’s generating code and output looks like, for a model of the type \(\log(y) = 1 + \frac{x}{2} + \epsilon\):

library(dplyr)

library(tidyr)

library(tibble)

library(ggplot2)

sample\_size <- 90

x\_range <- c(0, 1.5)

a <- 1

b <- 1.5

sigma <- 0.3

actual\_function <- tibble(x = seq(x\_range[1], x\_range[2], by = 0.01)) %>%

mutate(actual\_y = exp(a + b\*x))

random\_sample <- tibble(epsilon\_err = rnorm(n = sample\_size,

mean = 0,

sd = sigma),

x = runif(n = sample\_size,

min = x\_range[1],

max = x\_range[2])) %>%

mutate(sampled\_y = exp(a + b\*x + epsilon\_err))

linear\_model <- lm(formula = log(sampled\_y) ~ x, data = random\_sample)

prediction\_i <- predict(object = linear\_model,

newdata = actual\_function,

interval = "prediction") %>%

as\_tibble() %>%

rename\_at(vars(lwr,upr), ~paste0(., "\_pi")) %>%

mutate\_all(exp)

confidence\_i <- predict(object = linear\_model,

newdata = actual\_function,

interval = "confidence") %>%

as\_tibble() %>%

rename\_at(vars(lwr,upr), ~paste0(., "\_ci")) %>%

select(-fit) %>%

mutate\_all(exp)

intervals <- actual\_function %>%

bind\_cols(prediction\_i,

confidence\_i)

ggplot() +

geom\_line(data = actual\_function, aes(x, actual\_y, color = "Original Model"), size = 1) +

geom\_point(data = random\_sample, aes(x, sampled\_y), alpha = 0.5) +

geom\_line(data = intervals,

aes(x, fit, color = "Regression Fit"), size = 1) +

geom\_line(data = intervals,

aes(x, lwr\_pi, color = "Prediction Interval"),

linetype = 2, size = 1) +

geom\_line(data = intervals,

aes(x, upr\_pi, color = "Prediction Interval"),

linetype = 2, size = 1) +

geom\_line(data = intervals,

aes(x, lwr\_ci, color = "Confidence Interval"),

linetype = 2, size = 1) +

geom\_line(data = intervals,

aes(x, upr\_ci, color = "Confidence Interval"),

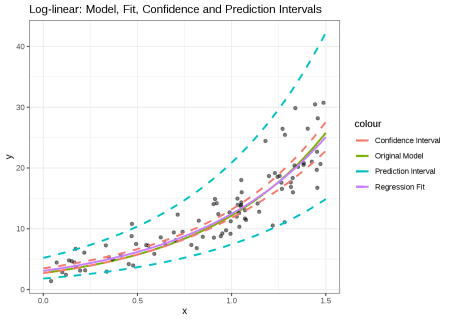
linetype = 2, size = 1) +

theme\_bw() +

xlab("x") +

ylab("y") +

ggtitle("Log-linear: Model, Fit, Confidence and Prediction Intervals")



**Conclusions**

Shiny apps are a great way to illustrate theoretical concepts, to provide intuition, and to let students experiment with parameters and see the outcomes. In this post I demonstrated how a Shiny app can be used to explain the concepts of a regression fit, confidence, and prediction intervals.

If you used Shiny for interesting educational demonstrations I’d love to hear about it! feel free to share in the comments or message me on twitter @SaridResearch.