# MIE237 March 1-2 Labs

*Neil Montgomery*

*February 29, 2016*

For this week and next you'll spend most of your computing time on the assignment, but there are also a few things to keep up with.

The new regression material from the past week included the $F$ distribution business, $R^2$, confidence intervals and prediction intervals. The $F$ and $R^2$ business was already being produced in the R code you've been using already, so there's nothing more to do.

That leaves us with estimating mean response and confidence intervals and predicting new values and prediction intervals. This can be done very quickly and easily with R.

## Worked example using exercise 11.57 data

Here's how I would go about obtaining the basic regression fit itself.

```
library(rio)
library(dplyr)

fitness <- import("Ex11.57.txt")

fitness %>%
  lm(O2 ~ Time, data = .) -> fitness_lm
```

Let's estimate the mean response at time 900 seconds, with a 95% confidence interval. Let's also predict the response at time 800 seconds, with a 95% prediction interval.

Everything we need is in single function `predict.lm`. The trickiest part is in how to tell it what the "new" x values are. This must be done using a `data.frame` with a variable name the same as in the original data, as follows:

```
new_times <- data.frame(Time = c(800, 900))
```

Now we can start using the `predict.lm` function. In its basic form it just returns the estimate/predictions:

```
predict.lm(object = fitness_lm, newdata = new_times)
```

```
##        1        2
## 49.82286 44.68942
```

We can tell it to add intervals as well:

```
predict.lm(object = fitness_lm, newdata = new_times, interval="prediction")
```

```
##        fit      lwr      upr
## 1 49.82286 42.41638 57.22934
## 2 44.68942 37.19917 52.17967
```

```r
predict.lm(object = fitness_lm, newdata = new_times, interval="confidence")
```

```
##        fit      lwr      upr
## 1 49.82286 48.28233 51.36340
## 2 44.68942 42.78647 46.59236
```

It's up to you to keep track of what you really want. Here's another way to accomplish the above perhaps more directly.

```r
predict.lm(fitness_lm, newdata = data.frame(Time = 900),
           interval = "confidence")
```

```
##        fit      lwr      upr
## 1 44.68942 42.78647 46.59236
```

```r
predict.lm(fitness_lm, newdata = data.frame(Time = 800),
           interval = "prediction", level = 0.95)
```

```
##        fit      lwr      upr
## 1 49.82286 42.41638 57.22934
```

Note that 0.95 is the default confidence level, but you can change it as suggested in the last line of code here.

## Your task

Use R to answer 11.22, 11.23, and 11.26 from the textbook.