Solution to Lab 8

Univariate Statistics with R

Three different activities this week! These may require a bit of thinking; we've tried to include enough information to help you find solutions, but 'our' solutions aren't the only ones, obviously.

A brief note on `with()`

You may have noticed a really useful function in the solutions for last weeks lab - the `with()` function. The way in which this function works is to allow you to specify a dataframe that an expression uses and therefore not need to include it when specifying variables. See the following example:

```
cor.test(schools$iq, schools$exam)
```

```
##
##  Pearson's product-moment correlation
##
## data:  schools$iq and schools$exam
## t = 3.015, df = 132, p-value = 0.003083
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.08802564 0.40593935
## sample estimates:
##       cor
## 0.2538249
```

This command can be written as:

```
with(schools, cor.test(iq, exam))
```

This is a nice trick that can make your code more readable and also save time and effort when you have an expression where you are referring to a dataframe multiple times. Take this code which will print any values of the `iq` variable in a dataframe called `schools` that are above 2.5 standard deviations from the mean:

```
schools$iq[schools$iq > mean(schools$iq, na.rm = T) + 2.5 * sd(schools$iq, na.rm = T)]
```

This could be written as:

```
with(schools, iq[iq > mean(iq, na.rm = T) + 2.5 * sd(iq, na.rm = T)])
```

`with()` is essentially a convenience function. Make use of it if you feel it produces more readable code and saves you precious time and typing energy!

Last Week's Linear Model

Task 1: Start by opening last week's (Lab 7) project. Since we're continuing to work on the same data, it's easier to extend the same project.

Task 2: Re-run the code you ran last week to load and merge the school datasets. (Just select the relevant lines of code and hit Ctrl-Return, or Command-Return on Mac).

Don't have the relevant R code saved? That's a problem. It really is important to save code, in scripts, when you use R, so that you (and others) can replicate work you've already done. Statistical analysis is as much a *process* as an *outcome*. For now we can overcome this by loading a version of the data that has been merged already - create a new script file now and in the first line, type `load(url("http://is.gd/tsktsk"))`; put the cursor on that line and send it to the console (Ctrl-Return or Command-Return on Mac). Remember: Use the script editor from now on!

Task 3: Create a linear model and inspect it by typing the code below (my merged data is in `schools`; you may need to edit the code appropriately).

```
model <- lm(exam ~ iq, data = schools)
summary(model)
```

```
##
## Call:
## lm(formula = exam ~ iq, data = schools)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -52.742 -14.319  -1.098  12.805  49.294
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  21.7961    11.4757   1.899  0.05970 .
## iq            0.3423     0.1135   3.015  0.00308 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.82 on 132 degrees of freedom
##   (17 observations deleted due to missingness)
## Multiple R-squared:  0.06443,    Adjusted R-squared:  0.05734
## F-statistic:  9.09 on 1 and 132 DF,  p-value: 0.003083
```

Tip: Models may vary very slightly from person to person, depending which outliers you decided to remove, etc.

Task 4: Which is the intercept? Which is the slope? How can you interpret each of them?

Task 5: You may conclude that the intercept isn't very meaningful! Create a new linear model, `model2`, with a more meaningful intercept. *(Take a look at slides 9 and 15 from this weeks lecture for some guidance with this)*

Tip: Arithmetic operators such as `+` and `-` have different meanings within a *formula* (the part which represents the relationship between variables, such as `exam ~ iq`. To 'isolate' arithmetic from the formula

intepretation, use `I()`, so `exam ~ I(iq + 23)` means "`exam` is predicted by `iq + 23`"; `exam ~ iq + 23` means "`exam` is predicted by `iq` and the number 23" (whatever that means!).

```
### ANSWER ###

# possible solutions include
model2 <- lm(exam ~ I(iq - 100), data = schools)
# or model2 <- lm(exam ~ I(iq - mean(iq, na.rm = T)), data = schools)
```

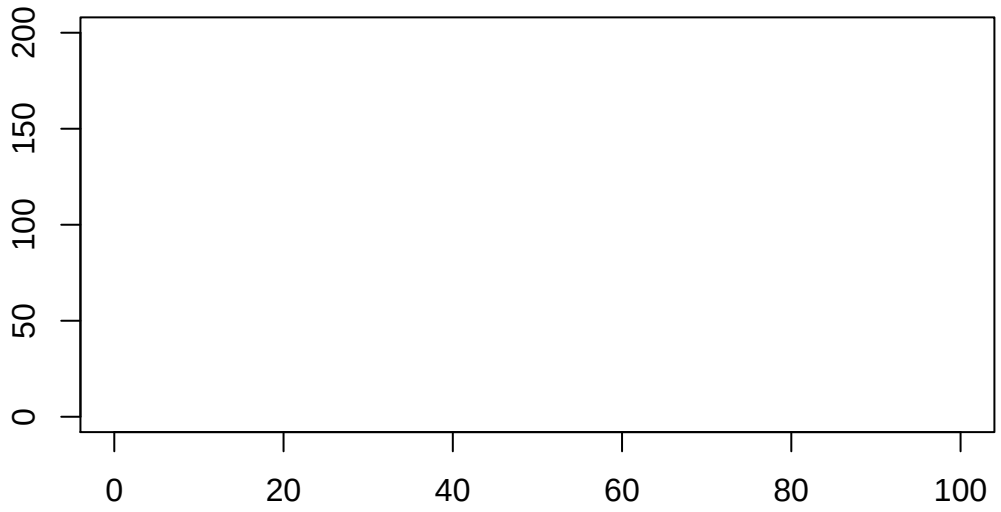
Task 6: What does the intercept of `model2` tell us? And the slope?

A Pretty Graph

Useful Stuff To Try First

Task 7: Create a blank canvas, like so:

```
plot(NULL, xlim = c(0, 100), ylim = c(0, 200), xlab = '', ylab = '')
```



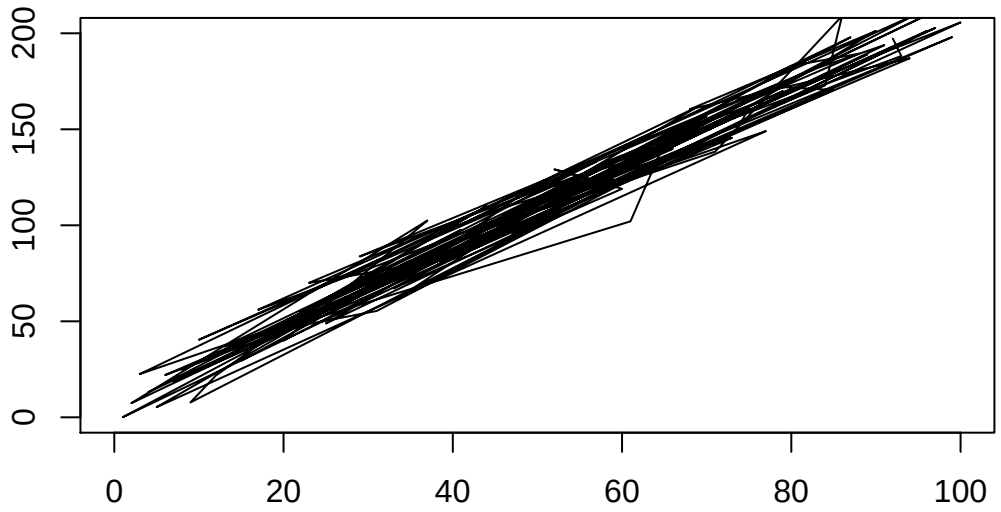
###

Task 8: Create a vector `x` of values between 0 and 100 in a random order, and another vector `y` which depends on `x`, like so:

```
x <- sample(100)
y <- 2 * x + rnorm(100, 10, 10)
```

Task 9: Now try drawing lines, connecting each point (x, y) together. What will you get?

```
lines(x, y)
```



Hopefully you'll have ended up with a mess, because the lines will have joined each point (x, y) *in the order they appear in the vector*. Take a look at the first 5 values in each vector (remember, your values will differ from mine):

```
head(x)
```

```
## [1] 92 93 29 81 62 50
```

```
head(y)
```

```
## [1] 197.21925 188.16161  83.75728 178.42899 134.89761 112.76551
```

The line will start at $(92, 197.2)$ and be drawn along to point two $(93, 188.2)$, and then to point three $(29, 83.8)$, and so forth. Given that these points are in a random order we get a *squiggly mess*. However, if the values for both vectors were ordered such that the x vector went from the lowest value to the highest value the line created would be much more interpretable, see fig. 1.

Task 10: Can you use `lines()` to recreate the plot in fig. 1 using the `x` and `y` vectors created above? *Use the tip below!*

```
### ANSWER ###

### NB., the trick is to use order(x) in each case
plot(NULL, xlim = c(0, 100), ylim = c(0, 200), xlab = '', ylab = '')
lines(x[order(x)], y[order(x)])
```

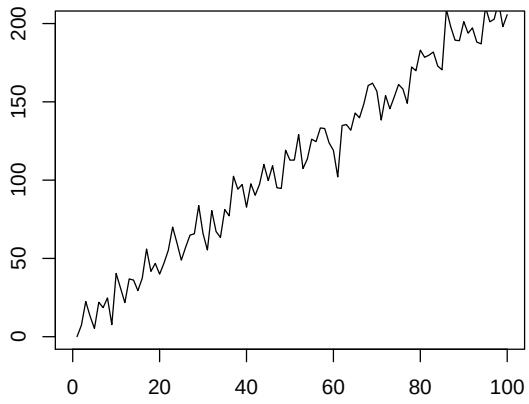


Figure 1: A simple line graph, made with 'lines()'

Tip: There are various ways of doing this, but the most ubiquitous is to use `order()`. Try `cbind(x, order(x))` to see what happens. Now try `x[order(x)]`. How does this work?

Task 11: Try adding two lines to your graph simultaneously. First create some more data (use the examples below, or your own), and bind the values together in a matrix:

```
### ANSWER ###

z <- 200 - 2 * x + rnorm(100, 10, 10)
t <- (1 + sin(pi * x / 50)) * 100
m <- cbind(z, t)
```

Task 12: Now look at the help for `matlines()`. Can you add lines to the graph from fig. 1 so it's like the one in fig. 2?

```
### ANSWER ###

# shorthand for plot() + lines() above
plot(x[order(x)], y[order(x)],
     xlim = c(0, 100), ylim = c(0, 200),
     xlab = '', ylab = '', type = 'l')
## new bit:  note the comma in m[] !
matlines(x[order(x)], m[order(x), ], col = c('red', 'blue'))
```

Task 13: One more thing to play with: Try the following (note, using `model` rather than `model2`):

```
p.exam <- predict(model, schools, interval = "confidence")
head(p.exam)
```

```
##        fit      lwr      upr
```

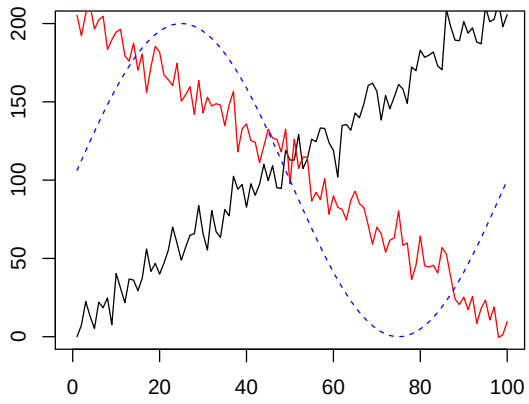


Figure 2: A simple line graph, made with 'lines()' and 'matlines()'

```
## 1 58.08451 54.43325 61.73578
## 2 56.37280 52.97690 59.76870
## 3 55.34577 51.93063 58.76091
## 4 59.79623 55.59458 63.99788
## 5 60.13857 55.80026 64.47688
## 6 55.34577 51.93063 58.76091
```

You'll find that `p.exam` is a matrix, consisting of three columns: `fit` is the fitted (predicted) value of `exam` based on each pupils `IQ` score; `lwr` and `upper` are the upper and lower bounds of the model confidence interval for each value of `schools$iq`.

Have a look at `?predict.lm` for much more on `predict()` (and, if you feel so inclined, try `?predict` to see where `?predict.lm` comes from!).

Task 14: Can you put everything in this section together and recreate the graph in fig. 3?

In this figure we have the basic scatterplot between actual `iq` and `exam` values along with the regression line that describes the linear relationship between the two (and the upper and lower estimates at each point).

```
### ANSWER ###

with(schools, plot(exam ~ iq))
# again, note the columns
with(schools, matlines(iq[order(iq)], p.exam[order(iq), ],
                       col = c('red', 'blue', 'blue'),
                       lwd = c(2, 1, 1), lty = c(1, 2, 2)))
```

More on Regression

Earlier on you estimated the effect of iq on exam performance, and found that there was a modest effect. However, that isn't the full story: After all of the students in the study had died, Professor Frank N. Stein measured all of their brain volumes, with the idea of discovering whether brain volume was a useful predictor of iq or of exam performance.

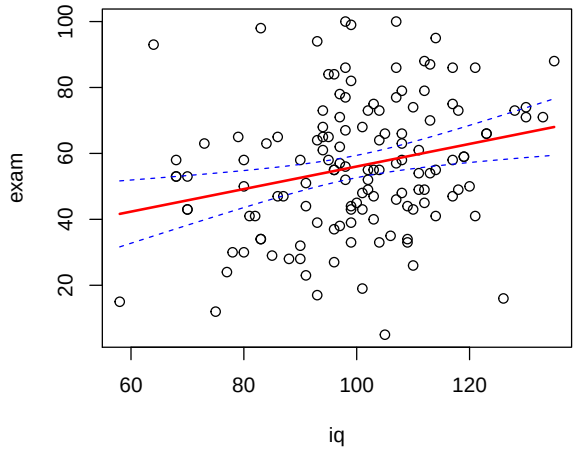


Figure 3: Best fit line and model confidence predicting ‘exam‘ from ‘iq‘

Task 15: Load the data for this week from the Learn website, and merge it with your existing schools dataframe.

Tip: As the data is already in an R format (.Rdata) you can use the `load()` function.

No tricks this week, the merge should go straightforwardly. Your merged dataframe will have an extra column in it now, called `bvol`, for brain volume in cm^3.

```
### ANSWER ###

schools <- merge(schools, bvdata)
```

Task 16: Does brain volume predict iq?

Tip: You need to build the relevant linear model, using `lm()`

```
### ANSWER ###

mod.b <- lm(iq ~ bvol, data = schools)
summary(mod.b)
```

```
##
## Call:
## lm(formula = iq ~ bvol, data = schools)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.716  -2.440   1.237   3.202   6.197
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -123.02111    5.73776  -21.44   <2e-16 ***
## bvol           0.18069    0.00465   38.86   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

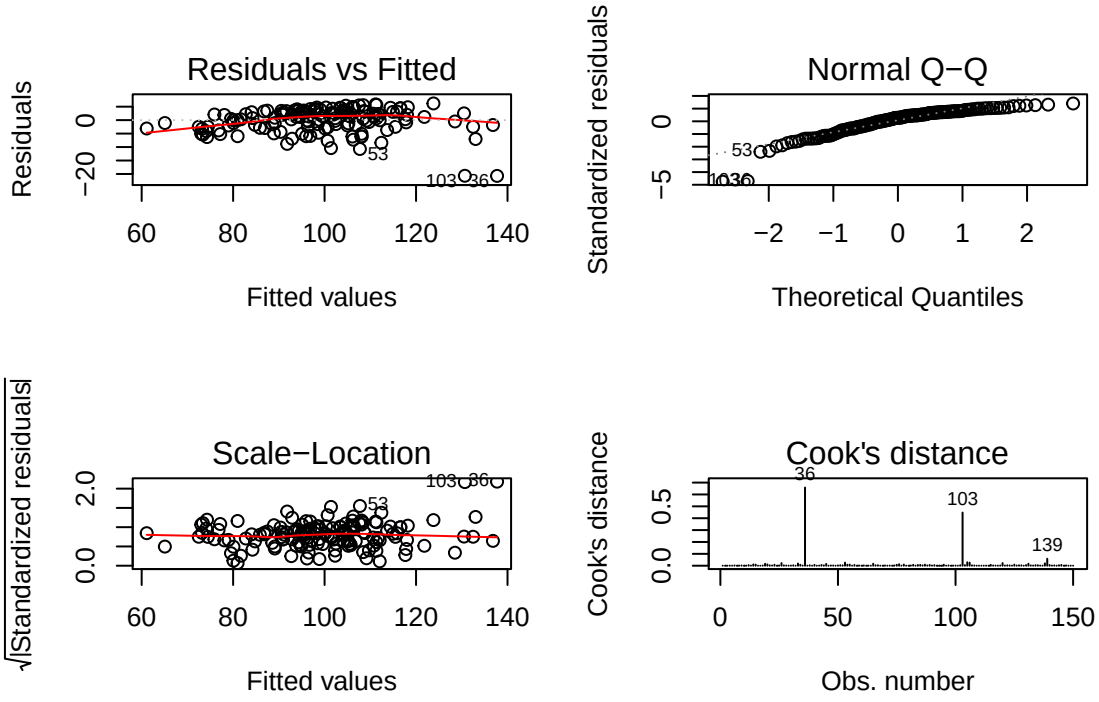
```
##
## Residual standard error: 4.479 on 148 degrees of freedom
## Multiple R-squared:  0.9107, Adjusted R-squared:  0.9101
## F-statistic:  1510 on 1 and 148 DF,  p-value: < 2.2e-16
```

Task 17: Are you sure? i.e. is the model appropriate?

Tip: Have you looked at your regression model carefully, using some of the model criticism tools you saw in the lecture? Functions you want to think about include `residuals()`, and `plot()` (where `plot()` is called on a linear model object)

```
### ANSWER ###

# Model Check (Visually):
par(mfrow = c(2, 2))
plot(mod.b, which = c(1:4))
```




```
# Some issues arise in checking our assumptions
  # QQ-Plot shows a systematic deviation at the lower end.
  # Homogeneity of Variance also problematic.
```

Task 18: We already know that `iq` predicts `exam` performance, because we tested and graphed the relationship earlier (and in the previous lab). However, does brain volume also predict exam performance? More importantly, does brain volume help predict exam performance *over and above* what `iq` already predicts?

Tip: You'll want to test whether adding a new predictor improves on chance (using `anova()`) as part of your answer... *Part 2 of the lecture will help you here.*

```
### ANSWER ###

### NO!
mod.c <- lm(exam ~ iq + bvol, data = schools)
anova(mod.c)
```

```
## Analysis of Variance Table
##
## Response: exam
##            Df Sum Sq Mean Sq F value   Pr(>F)
## iq          1   3572  3572.0  9.0708 0.003118 **
## bvol        1    284   283.8  0.7207 0.397463
## Residuals 131  51587   393.8
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(mod.c)
```

```
##
## Call:
## lm(formula = exam ~ iq + bvol, data = schools)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.986 -13.735  -0.961  13.491  48.623
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 65.72453   53.00477   1.240   0.2172
## iq           0.64730    0.37678   1.718   0.0882 .
## bvol        -0.06030    0.07103  -0.849   0.3975
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 19.84 on 131 degrees of freedom
##   (16 observations deleted due to missingness)
## Multiple R-squared:  0.06955,    Adjusted R-squared:  0.05534
## F-statistic: 4.896 on 2 and 131 DF,  p-value: 0.008903
```