UWEO StatR201 Lecture 1b

Multiple Regression Primer/Refresher

Assaf Oron, January 2013

A Toy Dataset about Earthquakes

Let's look at a simple R dataset, called `attenu`. It records the ground acceleration at various seismic stations during earthquake events.

```
dim(attenu)
```

```
## [1] 182   5
```

```
head(attenu)  # For help, let's do ?attenu
```

```
##   event mag station dist accel
## 1     1 7.0     117   12 0.359
## 2     2 7.4    1083  148 0.014
## 3     2 7.4    1095   42 0.196
## 4     2 7.4     283   85 0.135
## 5     2 7.4     135  107 0.062
## 6     2 7.4     475  109 0.054
```

We are interested to see how acceleration (how strongly the quake is felt) diminishes with distance.

At this point, we conveniently **ignore** the fact that each station and each event has multiple records (and as a first approximation it makes sense anyway).

Our first task in any modeling job, is to do extensive descriptive and data checking work.

```
table(is.na(attenu))
```

```
##
## FALSE  TRUE
##   894    16
```

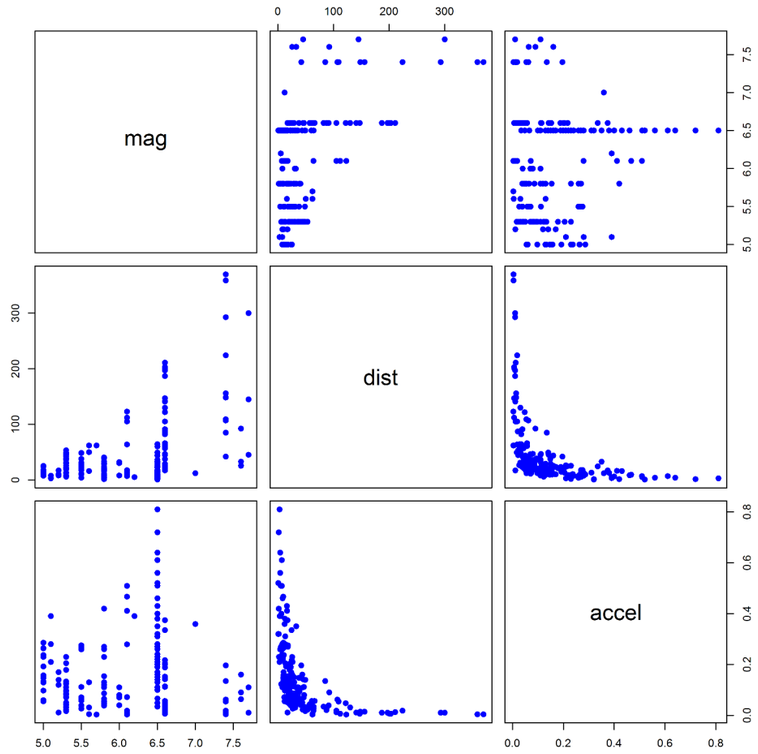
```
# Apparently there's missing data.... but where?
apply(attenu, 2, function(x) table(is.na(x)))
```

```
## $event
##
## FALSE
##   182
##
## $mag
##
## FALSE
##   182
##
## $station
##
## FALSE  TRUE
##   166    16
##
## $dist
##
## FALSE
##   182
##
## $accel
##
## FALSE
##   182
```

```
### Ok... only station ID is missing for some data; but we ignore that
### anyway.
```

Rule 0: ALWAYS Start with Descriptives

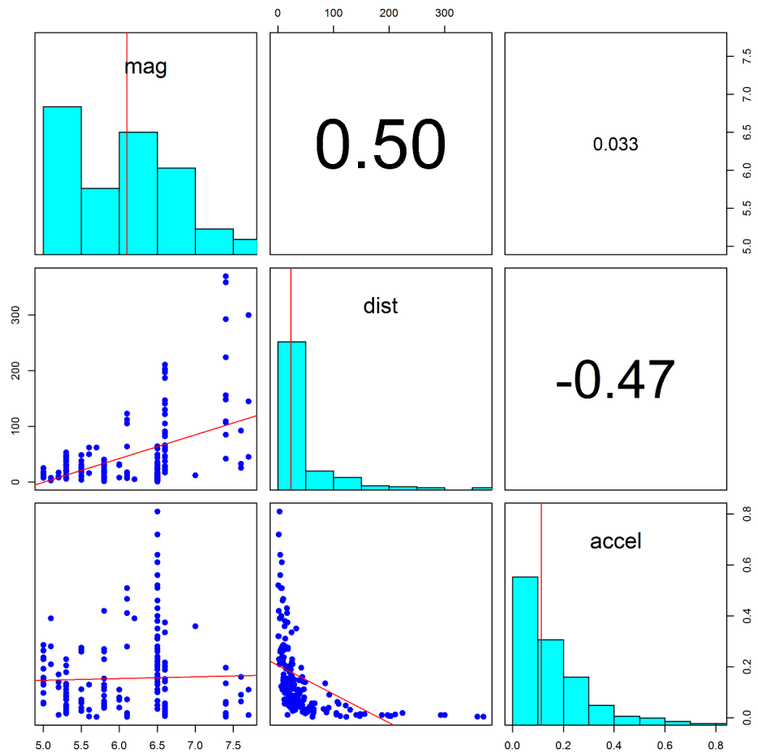
```
pairs(attenu[c(2, 4, 5)], pch = 19, col = 4)
```



Pairs plots are nice, but a bit wasteful. How so? And can we enhance them a bit?

An Improved Pairs Plot

```
source("../Code/enhancedGraphics.r")
pairsPlus(attenu[c(2, 4, 5)], pch = 19, col = 4)
```



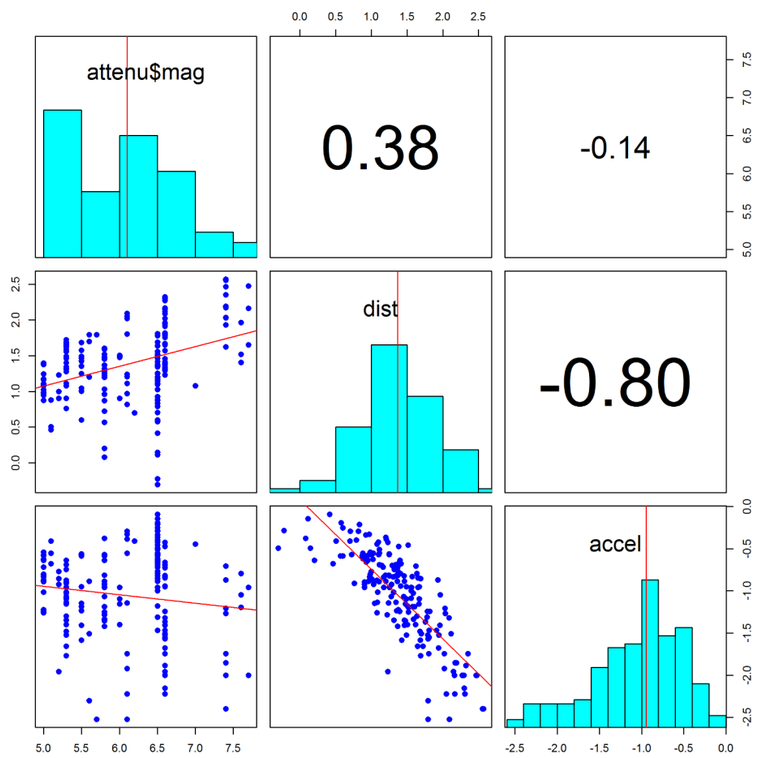
That's more like it:

- *On the upper triangle, we now get pairwise correlations (conveniently size-coded).*
- *On the diagonal, we get histograms of each variable, with the name still showing.*
- *On the bottom triangle, the scatters are overlaid with fit lines.*

Transformation Needed

Speaking of fit lines, the linear assumption seems broken for our main relation of interest (dist vs. accel). The distributions are also disturbingly skewed. Let's log-transform, using base 10 which is popular in science and esp. in geology (note that the magnitude is already log-transformed).

```
pairsPlus(cbind(attenu$mag, log10(attenu[c(4, 5)])), pch = 19, col = 4)
```



Now we're talking!

As we can see (and expect), the felt intensity ('accel') decreases with distance from the epicenter.

Micro-Break

Examine the pairs-plus function a bit…

(btw, this is our *"Cool R Trick of the Week"*. Pretty cool, huh?)

…and the data…

…Q's so far?

Our First R Regression Summary Together :)

```
summary(lm(log10(accel) ~ log10(dist), data = attenu))
```

```
##
## Call:
## lm(formula = log10(accel) ~ log10(dist), data = attenu)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.1249 -0.1880  0.0277  0.2039  0.7129
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.0802     0.0669     1.2     0.23
## log10(dist)  -0.8247     0.0456   -18.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.317 on 180 degrees of freedom
## Multiple R-squared: 0.645,   Adjusted R-squared: 0.643
## F-statistic:  327 on 1 and 180 DF,  p-value: <2e-16
```

Note: *Doing 'summary' on-the-fly, simultaneously as we call the regression command, is a quick-and-dirty way of examining the effects, without storing the (often bulky) regression object in memory.*

Also, 'summary' provides the regression dashboard in the way most statisticians like to view it - packing a lot of information in a compact and convenient layout, that is shared by nearly all regression-like R methods.

Interpreting Regression Results, Lesson 0

Here's how we could interpret the summary in a short paragraph:

"The acceleration felt at the station is significantly associated with the distance from the earthquake's epicenter. The relationship can be approximated as a **power law:**

$$a \sim d^{-\beta} .$$

The estimate of β, based on 182 records from California, is 0.825 ± 0.046 . This relationship alone explains 64% of the variability in acceleration."

Ok, what about the effect of magnitude?

```
summary(lm(log10(accel) ~ mag, data = attenu))$coef
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.4426    0.33255  -1.331  0.18489
## mag          -0.1004    0.05428  -1.849  0.06605
```

As the scatters above showed, acceleration **decreases** with increasing magnitude! WTH?

This is where **multiple regression** comes into play.

Univariate regression only gauges the overall effect of a single covariate, ignoring any context represented by other variables.

OTOH, multiple regression gauges the effect of each single covariate, while acknowledging the effects of all others, **and holding them constant.**

```
mod2 = lm(log10(accel) ~ log10(dist) + mag, data = attenu)
summary(mod2)
```

```
##
## Call:
## lm(formula = log10(accel) ~ log10(dist) + mag, data = attenu)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0343 -0.1852  0.0362  0.2153  0.6621
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.7161     0.1909   -3.75  0.00024 ***
## log10(dist)  -0.9047     0.0470  -19.24  < 2e-16 ***
## mag           0.1490     0.0337    4.42  1.7e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.302 on 179 degrees of freedom
## Multiple R-squared: 0.68,    Adjusted R-squared: 0.676
## F-statistic:  190 on 2 and 179 DF,  p-value: <2e-16
```

Now it makes sense, and here's how it can be interpreted:

"The acceleration a felt at the station is significantly associated with the distance d from the earthquake's epicenter, **while adjusting for earthquake magnitude.** The relationship can be approximated as a power law:

$$a \sim d^{-\beta_1} .$$

The estimate of β_1, based on 182 records from California, is 0.905 ± 0.047 . Magnitude m is also positively and significantly associated with acceleration:

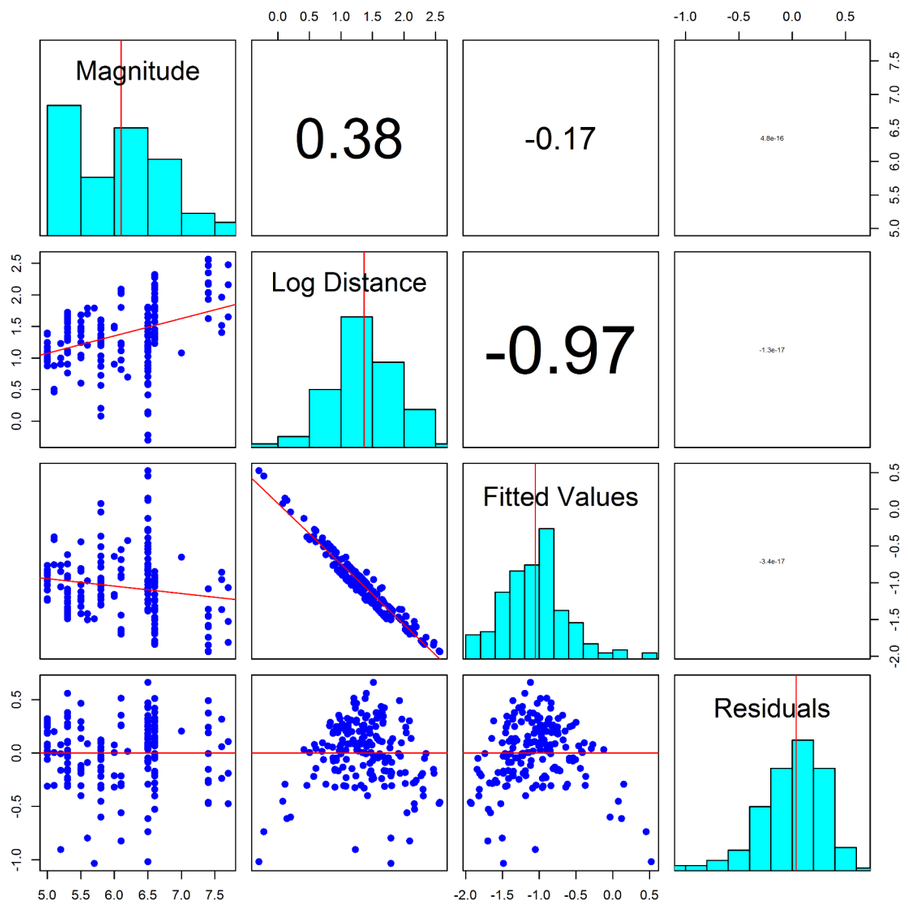
$$a \sim e^{\beta_2 m} .$$

The estimate for β_2 is 0.149 ± 0.034 . This two-variable model explains 68% of the variability in acceleration."

Wait! So why did magnitude flip its sign after we adjusted for distance?

In other words, why did it have the wrong side when not adjusted?

Let's look at some diagnostics:



*When it comes to diagnostics, We mostly really care about the **visual patterns** on **the bottom row.** (btw, the numerical correlation of the residuals with covariates and fitted values should be zero. why?)*

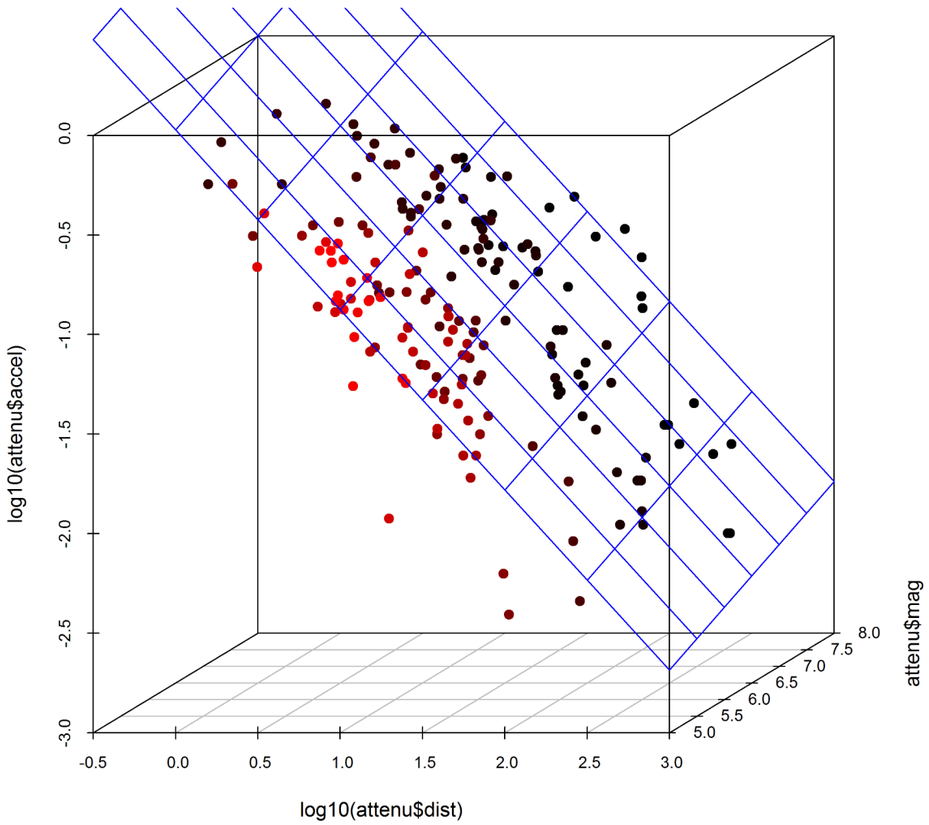
Look for nonlinearities, skewness, disturbing outliers, or anything else that might raise an eyebrow.

What do you think? Is the model good to go?

Do you remember what residuals are?

Visually, multiple regression is equivalent to passing a best-fit (hyper)plane through the cloud of points, plotted in p dimensions (p is the number of covariates in the model, plus 1). *(Note: this basic 3D scatter is rather lame; we will later learn nicer options, e.g. using the `lattice` package)*

```
library(scatterplot3d)
zz <- scatterplot3d(log10(attenu$dist), attenu$mag, log10(attenu$accel), type = "p",
    cex.sym = 1.5, pch = 20, highlight.3d = TRUE, angle = 30, scale.y = 0.5)
zz$plane3d(mod2$coef, lty = "solid", col = 4)
```



To be Continued…

As I said in the intro, we hit the ground running this quarter with what is - essentially - a compressed applied-regression course.

Next week we will dig deeper into it.

In a few days I will have Homework 1 online. It is due on Week 3.

There **will** be online office hours this coming Tuesday *(assuming I learn how to do it by then :)*

Meanwhile, before Lecture 2 it's a good idea to read a little background:

- Hastie et al., Sections 3.1-3.2
- Dobson and Barnett, Sections 2.4, 6.1-6.3

Oops! I almost forgot. R Annyoance/Workaround of the Week:

```
attenu[attenu$station == 1008, ]
```

```
##       event mag station dist accel
## 8         2 7.4    1008  224 0.018
## NA       NA  NA    <NA>   NA    NA
## NA.1     NA  NA    <NA>   NA    NA
## NA.2     NA  NA    <NA>   NA    NA
## NA.3     NA  NA    <NA>   NA    NA
## NA.4     NA  NA    <NA>   NA    NA
## NA.5     NA  NA    <NA>   NA    NA
## NA.6     NA  NA    <NA>   NA    NA
## NA.7     NA  NA    <NA>   NA    NA
## NA.8     NA  NA    <NA>   NA    NA
## NA.9     NA  NA    <NA>   NA    NA
## NA.10    NA  NA    <NA>   NA    NA
## NA.11    NA  NA    <NA>   NA    NA
## NA.12    NA  NA    <NA>   NA    NA
## NA.13    NA  NA    <NA>   NA    NA
## NA.14    NA  NA    <NA>   NA    NA
## NA.15    NA  NA    <NA>   NA    NA
```

```
### WTH ?!?!?!
```

If you try to subset data, using an (in)equality condition on a variable with missing values - **the indices with missing values will be included as if they were TRUE!**

Workarounds:

I have no idea why this behavior *(automatic extra credit to whoever researches this and brings back the answer)*.

One straightforward workaround is to add **an explicit non-missingness condition:**

```
attenu[attenu$station == 1008 & !is.na(attenu$station), ]
```

```
##   event mag station dist accel
## 8     2 7.4    1008  224 0.018
```

It's a bit awkward, esp. if you need to combine 2 conditions on 2 variables, that each has missing values!

Recently I have come across a more elegant workaround:

```
attenu[attenu$station %in% 1008, ]
```

```
##   event mag station dist accel
## 8     2 7.4    1008  224 0.018
```

Of course, if you don't need the records with missing values for any purpose, it's best to purge your dataset early in the session.

See you next week!