

UWEO StatR201 Lecture 4

De-CRAM 1 and Likelihood Ratio Tests

Assaf Oron, January 2013



De-CRAM 1

As our Advisory Board feared in December, this class has too much material. I'd wager an educated guess that the majority of students, as well as the instructor, feel at least somewhat disoriented right now.

Therefore, I devote today's first hour (give-or-take) to our first De-CRAM session. There will probably be one every 3-4 weeks. We will:

- Go over what we've learned and put it in perspective
- Re-practice some of the essential skills
- Q&A

After that, we will "resume" the original plan:

- Finish up Lecture 3 - likelihood in regression, quantile/robust regression
- Learn about Likelihood-Ratio Tests
- R trick/annoyance

Generalized Linear Models will have to wait for next week. But first, let's revisit that joke I sent you and no one responded...

De-CRAM: Regression

With so much effort spent warning you what **Not** to do, maybe I've de-emphasized what **Yes** to do.

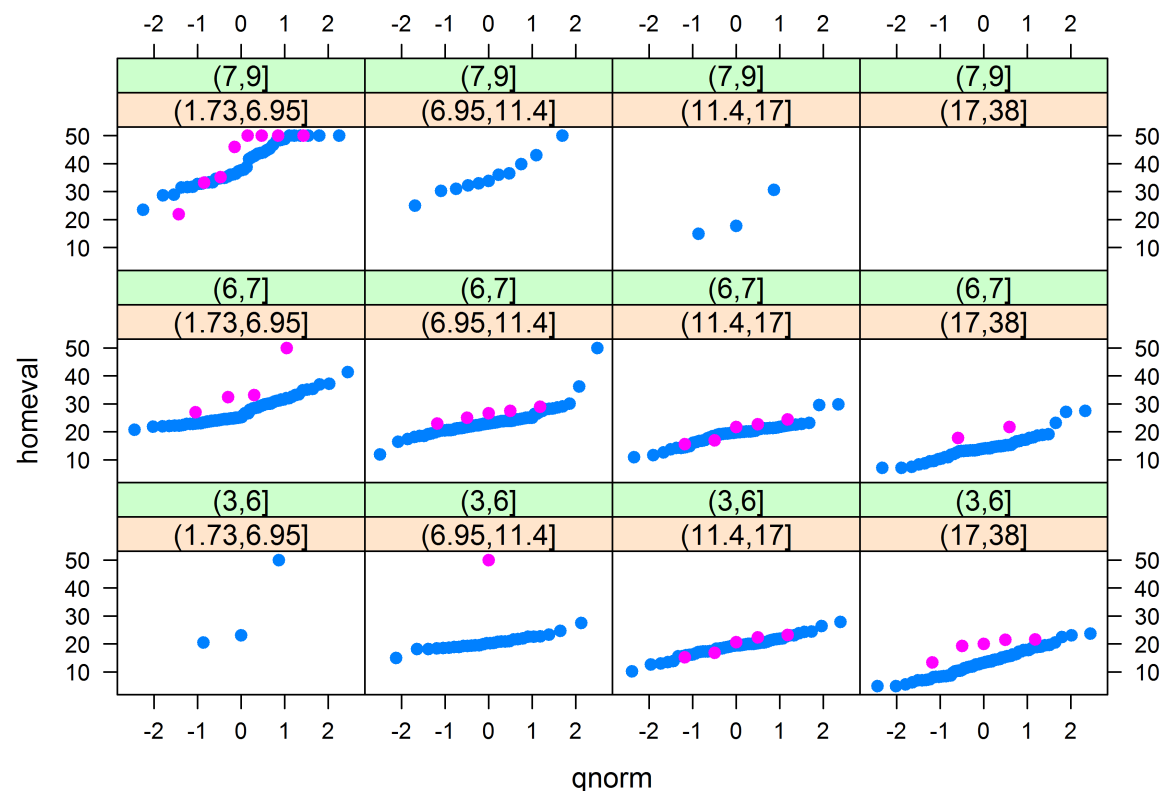
The biggest “Yes” is **Yes, do tons of descriptives.**

- Tables to find missing values. Looking at a spreadsheet view of the data: not a bad idea either.
- For continuous variables: visual univariate summaries (`hist`, `density`, `qqnorm`) to see the distribution.
- For categorical variables: tables.
- For continuous variables you can use tables as well, to find if there's too many instances of the same value (like in the `boston` dataset), you can use `table(table(x))`.
- Bivariate scatters (continuous), boxplots (continuous vs. categorical) and tables (categorical).
- Three-way and Four-way relationships using `lattice`.
- **Descriptives are not just “foreplay”. You should return to them again and again, in response questions arising from model output, or from your collaborators.** Keep in mind, that you will never find out everything in your first session with a dataset (if you ever do, consider yourself extremely lucky).
- **And don't afraid to be creative. Invent your own descriptives! That's one of the most fun sides of working with R.**

De-CRAM: Let's Learn some Cool lattice Tricks

The `lattice` version of `qqnorm`, and 4-variable plotting without breaking a sweat!

```
qqmath(~homeval | cut(lowSES, quantile(lowSES)) + cut(rooms, c(3, 6, 7, 9)),
      data = boston, group = river, pch = 19, scales = list(alternating = 3))
```



What can you learn about the relationships between the 4 variables?

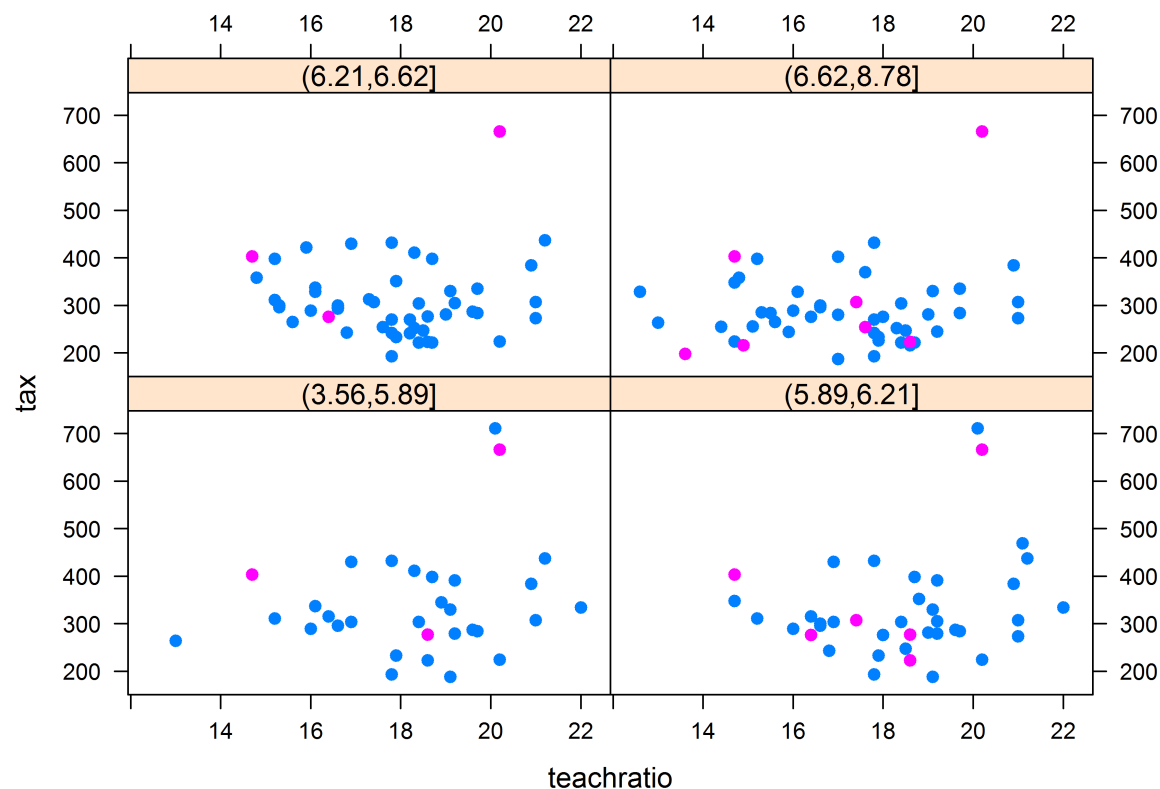
Note: `scales = list(alternating=3)` makes the scales show on *all* edges.

De-CRAM: Some More Cool lattice Tricks

Let's inspect those repeating values in `tax` and `teachratio`. Are they from the same places?

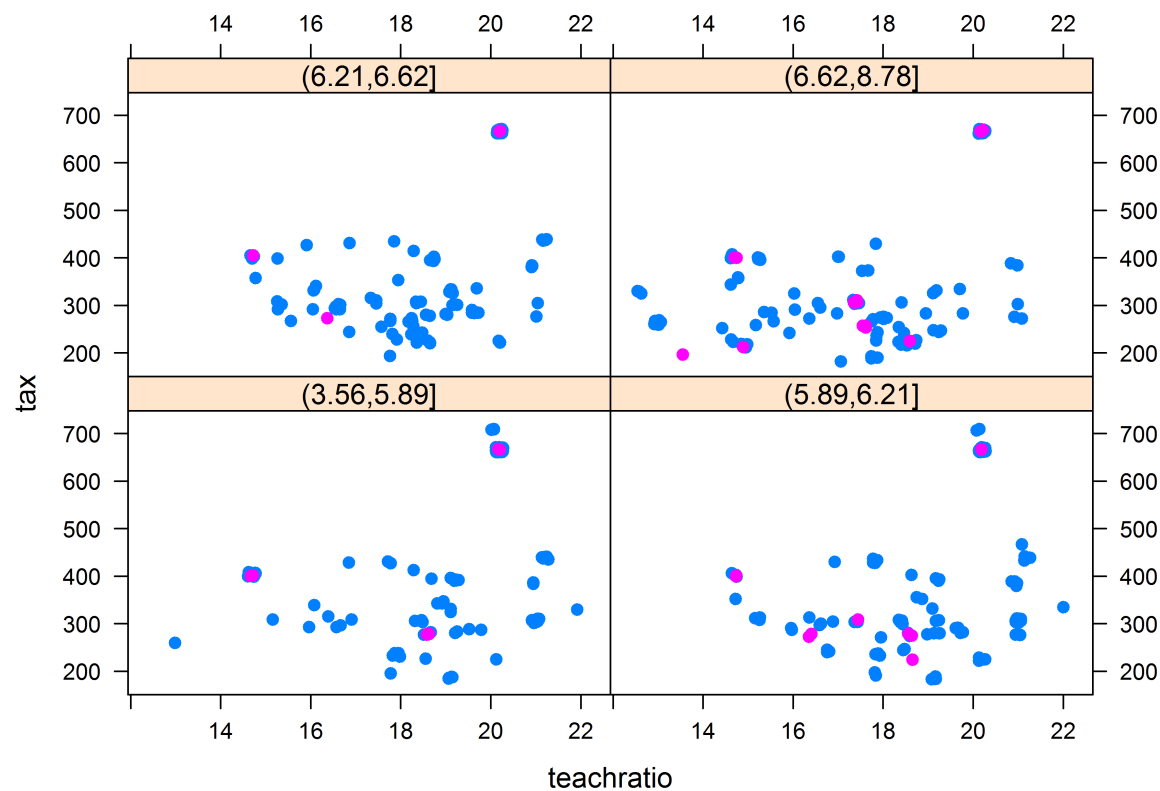
Where'd they go?

```
xyplot(tax ~ teachratio | cut(rooms, quantile(rooms)), data = boston, group = river,
       pch = 19, scales = list(alternating = 3))
```



De-CRAM: Some More Cool lattice Tricks

```
xyplot(tax ~ teachratio | cut(rooms, quantile(rooms)), data = boston, group = river,
       pch = 19, scales = list(alternating = 3), jitter.x = T, jitter.y = T, amount = 0)
```



Aha. See them now?

Note: setting `amount=0` actually lets lattice automatically determine the jitter (1/50 of the st.dev in each direction, I think).

Questions? (online/in-class)

De-CRAM: Ok, back to Regression

From the basic descriptives we move towards modeling by **identifying** things we need to do (partial list):

- Identifying variables to transform, and examining transformations.
- Identifying outliers (univariate/bivariate) and other suspect values/patterns, and deciding what to do about them for the time being (**consult the science**).
- Examining collinearity/VIF, and deciding what to do about them (**consult the science**).
- Similarly for influential points (leverage etc.).
- **Then we get to start building the model - which will be a major topic for the coming 2-3 weeks.**
- So we'll skip it now, and instead revisit how to **interpret the model output** after it's all done.

De-CRAM: Regression Interpretation

We did a bit on Lectures 1-2, but maybe didn't practice it enough - and certainly not for **categorical covariates**. Here goes:

```
summary(lm(Petal.Length ~ Species, data = iris))$coef
```

##	Estimate	Std. Error	t value	Pr(> t)
## (Intercept)	1.462	0.06086	24.02	9.303e-53
## Speciesversicolor	2.798	0.08607	32.51	5.255e-69
## Speciesvirginica	4.090	0.08607	47.52	4.106e-91

The average petal length of *Iris Setosa* flowers is 1.5 inches (95% CI: 1.3,1.6).

Iris versicolor petals are 2.8 inches longer on average (2.5,2.9; $p < 1E-6$). *Iris virginica* petals are, on average, 4.1 inches longer than *Setosa* (3.9,4.3; $p < 1E-6$).

De-CRAM: Regression Interpretation

```
summary(lm(Petal.Length ~ Species + Sepal.Length, data = iris))$coef
```

##		Estimate	Std. Error	t value	Pr(> t)
##	(Intercept)	-1.7023	0.23013	-7.397	1.005e-11
##	Speciesversicolor	2.2101	0.07047	31.362	9.646e-67
##	Speciesvirginica	3.0900	0.09123	33.870	4.918e-71
##	Sepal.Length	0.6321	0.04527	13.962	1.121e-28

The average petal length of *Iris Setosa* flowers with a sepal length of zero, is -1.7 inches (95% CI: -1.2,-2.2).

Iris versicolor petals **with a sepal length equal to *Setosa*'s**, are 2.2 inches longer on average (2.0,2.4; $p < 1E-6$). *Iris virginica* petals are, on average, 3.1 inches longer than *Setosa* flowers with similar-length sepals (2.9,3.3; $p < 1E-6$).

For all species, longer sepals are associated with longer petals: for each 1-inch increase in sepal length, petals are 0.63 (0.55,0.71; $p < 1E-6$) longer on average.

Note that just like this interpretation, regression has no problems delivering you scientifically meaningless numbers with a straight face. Statistical models, and certainly their output, don't check themselves for real-life plausibility.

De-CRAM: Regression Interpretation

```
summary(lm(Petal.Length ~ Species * Sepal.Length, data = iris))$coef
```

##	Estimate	Std. Error	t value	Pr(> t)
## (Intercept)	0.8031	0.5310	1.5122	1.327e-01
## Speciesversicolor	-0.6179	0.6837	-0.9039	3.676e-01
## Speciesvirginica	-0.1926	0.6578	-0.2928	7.701e-01
## Sepal.Length	0.1316	0.1058	1.2439	2.156e-01
## Speciesversicolor:Sepal.Length	0.5548	0.1281	4.3298	2.780e-05
## Speciesvirginica:Sepal.Length	0.6184	0.1210	5.1114	1.001e-06

The average petal length of *Iris Setosa* flowers with a sepal length of zero, is 0.8 inches (95% CI: -0.3, 1.9).

Iris versicolor petals **with a sepal length of zero**, are 0.6 inches **shorter** on average (-2.0, 0.8; $p = 0.37$).
Iris virginica petals with zero-length sepals are, on average, 0.2 inches shorter than *Setosa* flowers (-1.5, 1.1; $p = 0.77$).

For *Setosa*, longer sepals are associated with longer petals: for each 1-inch increase in sepal length, petals are 0.13 (-0.08, 0.34; $p = 0.22$) longer on average. The increase is stronger for *versicolor* and *virginica* - by 0.55 (0.30, 0.81; $p = 3E-5$) and 0.62 (0.38, 0.86; $p = 1E-6$) **more so, respectively, than for *Setosa*.**

...and now (after questions)... break, and then back to the future - to finish up Lecture 3.

Likelihood-Ratio Tests (LRTs)

We saw that MLEs are the “best” estimators in many ways. This is not the only bonus of the Likelihood approach. Another one is that hypothesis tests based on the likelihood are **the most powerful ones**, all other things being equal (...and, as usual, assuming we got the true model right).

A test's power, is the probability of its p-value falling below a pre-set threshold, assuming that indeed the Null is false.

$$Power = \Pr(\text{"Reject } H_0 \text{"} \mid H_a),$$

where H_a is a **specific** reference alternative distribution (specific enough to calculate probabilities with).

Bottom line: the more powerful the test, the more sensitive our detection method.

Likelihood-Ratio Tests (LRTs)

The test proven to be most powerful for a broad range of scenarios, is known as **the Likelihood-Ratio Test or LRT**. We usually prefer to work via the log of this likelihood ratio:

$$\lambda = \log\left(\frac{L_a}{L_0}\right) = l_a - l_0,$$

where L_a, L_0 are the best possible (i.e., maximum) likelihoods under the Alternative and Null, respectively, and l_a, l_0 their natural logarithms. The capital/lower-case L notation is fairly standard.

Important note: when H_0 fixes some parameter(s) at specific values/ranges, where as H_a does not - then L_a is evaluated by finding the MLE for these parameters over an *unrestricted range*, regardless of whether the MLEs eventually fall inside “Null territory” or not.

Likelihood-Ratio Tests (LRTs)

It turns out, that if the Alternative can be seen as an *extension* of the Null (i.e., if the Null can be seen as a simplification of the Alternative), then...

...assuming as usual that the Null is true, as $n \rightarrow \infty$, $2\lambda \rightarrow \chi^2$, with the Chi-square **degrees of freedom (d.f.)** equal to the different in d.f. between the two models.

Loosely speaking, the number of degrees of freedom in a model is the number of parameters estimated from the data. For example,

- With coin tosses, the model $p = 0.5$ has **zero d.f.**. A $p \neq 0.5$ Alternative would estimate \hat{p} from the data, and therefore it has 1 d.f.
- With regression, each continuous covariate adds 1 d.f.
- Each categorical variable with k levels, adds $k - 1$ d.f. **Indeed, LRT is the proper way to test for such covariates. It is easy to find examples, when all individual levels seem “Null” - but the entire set of additional covariates is in fact significant.**

Questions? (online/in-class)

Nested Models and Nested LRTs

In short, a standard linear regression model has as many d.f. as the number of rows in its coefficient-estimate summary. *(note that the model also estimates σ^2 , but as long as all compared models do it, it's a wash).*

In regression, when two models are comparable via LRT, the simpler one is said to be **nested** within the more complicated one. Examples for nesting:

- Adding any number of covariates to the simpler model
- For a categorical variable, ungrouping some levels or ranges that were lumped together in the simpler model

Nesting and hence LRT are **not** meaningful – in the asymptotic Chi-Squared sense – whenever the two models have the same d.f., or when the model with fewer d.f. has some covariates that the more complicated one does not. Tricky cases:

- Modeling a continuous covariate as linear, vs. cutting it into a 3-level factor. An LRT is not meaningful.
- Redividing a categorical variable from k levels to $m > k$ levels, but in a **non-nested manner** - so that some observations that were in different levels before are now in the same level. Again, an LRT is not meaningful.

Why are Non-Nested LRTs Not Meaningful?

In a nested-model LRT, the Null assumes that the simpler model is better: the additional covariates all have a true effect of 0.

When comparing two non-nested models, you cannot write down the Null, because each model involve estimating parameters which are (at best) set to zero in the other, or (at worst) are meaningless in the other (e.g., different manipulations of the same underlying variable).

This means, that you **should not** use or quote the p-values for 2λ between non-nested models.

That being said, comparing (penalized) log-likelihoods of non-nested models is a widely-accepted practice in **model selection**, with some theoretical justification. Just not as a formal hypothesis test.

Regression LRTs in R

```
library(lmtest)
mod1 = lm(log10(accel) ~ log10(dist), data = attenu)
lrtest(mod1) # vs. the Null of no regression
```

```
## Likelihood ratio test
##
## Model 1: log10(accel) ~ log10(dist)
## Model 2: log10(accel) ~ 1
##   #Df LogLik Df Chisq Pr(>Chisq)
## 1    3  -48.1
## 2    2 -142.3 -1   189    <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```


Regression LRTs in R

```
mod2 = lm(log10(accel) ~ log10(dist) + mag, data = attenu)
lrtest(mod1, mod2) # vs. each other
```

```
## Likelihood ratio test
##
## Model 1: log10(accel) ~ log10(dist)
## Model 2: log10(accel) ~ log10(dist) + mag
##   #Df LogLik Df Chisq Pr(>Chisq)
## 1    3  -48.1
## 2    4  -38.6  1  18.9    1.4e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Note that `lrtest` will **not** scream error at you if you compare non-nested models.

Questions? (online/in-class)

R Annoyance of the Week (also `lattice`)

As some of you have noticed, `lattice` doesn't make it easy to control panel names. Here's a trick from the package author, Deepayan Sarkar:

```
mylat = xyplot(homeval ~ rooms | factor(river), pch = 19, data = boston, scales = list(alternating = 3))  
dimnames(mylat)
```

```
## $`factor(river)`  
## [1] "0" "1"
```

Note that for this to work, you must capture the plot via an assignment to some variable on the LHS. You also want to condition on a factor, rather than a numeric variable (so: it's numeric like here, just `factor()` it).

R Annoyance of the Week (also lattice)

`dimnames` returns a list. You access a list's components using double square brackets `[[]]` (you've probably learned this in fall).

So now we can change the values:

```
dimnames(mylat)[[1]] = c("Not by River", "By Charles River")
print(mylat)
```

