UWEO StatR201 Lecture 2

Regression - What Could Possibly Go Wrong? (1)

Assaf Oron, January 2013

Overview: It's a Busy Evening Tonight…

- Warm-Up: Interactions and other formula details
- The Generic Regression Framework
- Collinearity and what to do about it
- Self-Work and Break 1
- Residuals 1: Patterns
- Residuals 2: Outlier Triage, Studentized Residuals
- Outliers in X and Influence
- Self-Work and Break 2
- Non-Normality: really that bad?
- Q&A

From now on, any regression will be assumed to be multiple. For statisticians, univariate regression (i.e., a single x variable) is just a special case.

Interaction - and its Interpretation

First, a small correction regarding geometric visualization of the model's meaning. At the end of class, I showed the model fit as a (hyper-)plane fit through the points. Then we went to Hastie et al. (Section 3.2), and saw the model as a **projection.** That projection was done in a different space.

The observations are n points "living" in p-dimensional space. But the set of all observations can be seen as **a vector in n-dimensional space - as can the covariates. The covariates span a p-dimensional manifold in that space, onto which the regression projects the data vector.**

And now: Interaction.

Eli showed interaction in the context of ANOVA.

- Interaction can be used in regression between any pair of covariates, regardless of whether they are categorical or continuous.
- Mathematically, it is very simple: the model just adds a new covariate which is **the product of the original two.**
- Conceptually, it measures how one covariate **modifies** or **modulates** the effect of the other.
- The real trouble starts with interpretation.

Interaction Changes the Meaning of Participating Covariates

Recall: in regression, each covariate effect is evaluated while holding all others constant.

Now we added a product, say $a * b$. What does it mean, the effect of a while $a * b$ is held constant?

The only way this is guaranteed to happen, is by fixing $b = 0$.

So adding an interaction between X_j and X_k changes the meaning of $\hat{\beta}_j$ – from X_j's effect when all others are held constant, to X_j's effect when $X_k = 0$ *and* all others are held constant. And vice versa for $\hat{\beta}_k$.

This is one reason why statisticians are usually less eager than non-statisticians to add more and more interaction terms. We know how tricky it can become to even understand the meaning of the final model, and to make sure it is sensible.

How it Plays Out in R

Revisit our `attenu` model:

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

```
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept)  -0.7161    0.19094  -3.750 2.383e-04
## log10(dist)  -0.9047    0.04703 -19.236 1.973e-45
## mag           0.1490    0.03367   4.424 1.681e-05
```

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

```
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -1.44827    0.63794 -2.2702  0.02439
## log10(dist)     -0.38716    0.43290 -0.8943  0.37235
## mag              0.26552    0.10257  2.5886  0.01043
## log10(dist):mag -0.08093    0.06729 -1.2027  0.23068
```

Ok, the interaction is definitely not significant… but now distance, the dominant covariate, is also non-signficant! Its absolute strength, too, has been cut by 60%.

Recall that this is the estimate of distance's effect for an earthquake of magnitude…. zero. Can we do anything about it?

```
summary(attenu$mag)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5.00    5.30    6.10    6.08    6.60    7.70
```

```
# Let's mean-center the magnitude first...
```

How it Plays Out in R

```
summary(lm(log10(accel) ~ log10(dist) * I(mag - 6), data = attenu))$coef
```

```
##                         Estimate Std. Error t value  Pr(>|t|)
## (Intercept)              0.14484    0.07269   1.993 4.783e-02
## log10(dist)             -0.87274    0.05399 -16.165 9.318e-37
## I(mag - 6)               0.26552    0.10257   2.589 1.043e-02
## log10(dist):I(mag - 6) -0.08093    0.06729  -1.203 2.307e-01
```

Lesson: whenver you use interaction with continuous covariates, you must center those covariates around a meaningful and sensible reference value. Notes:

- Notice the use of `I()`. We do this whenever using some algebraic function of the covariate, including powers, e.g., `I(mag^2)`.
- When the interaction involves a categorical variable (=`factor` in R), a product term is added between each level combination beyond the reference levels. The meaning of a covariate when it interacts with a factor, is equivalent to setting the factor at its reference level.
- Ironically, **the interaction term itself is reference-invariant!** Compare its value and significance in this slide and the previous one.
- When specifying `a*b` in the formula, you automatically add `a`, `b` and their interaction. Technically, you can insert *only* the interaction by writing `a:b`. Most statistician would strongly advise against doing that, barring rare exceptions.
- Ok, so how *do* you interpret the interaction coefficient itself? (hint: there are always 2 valid ways)

Question? *Online* questions?

Black Box Statistics: Sermon 1

This was a great intro for today's sermon. Nowadays, performing regression - even a very complicated one - is easy. A single command line (in R), or even a drop-down click (in Stata/SPSS) does the trick.

All the complexity is hidden inside a numerical "black box" that spits out numbers, tables and plots.

As the little interaction demo shows, it is also very easy to spit results that are **crazy** wrong - and then take them and run. Uninformed and even reckless use of regression (and other similar) tools, is one of the plagues of 21st Century statistics.

My role is to help you become **informed, responsible, competent users.**

The 3 handles, or keywords, to keep in mind are:

Assumptions, Properties, Behavior

We now proceed to define what a **generic** regression model is and what it can do.

The Generic Regression Framework

Response i is related to predictor variables (=covariates) thus:

$$y_i = m(X_i, \beta) + \epsilon_i,$$

where:

- $m(X, \beta)$ is some **deterministic** function;
- X_i is a vector of **all relevant** covariate values for this specific response;
- β is a vector (of the same length as X_i), with fixed but unknown values which are the same for all responses;
- ϵ_i is random or (equivalently) unpredictable noise, whose randomness properties are shared across all responses.

Usually, we will write the regression equation in matrix-vector form:

$$y = m(\mathbf{X}, \beta) + \epsilon$$

*(note that in statistical equations, vectors are usually **not** marked differently from single variables)*

What Can Regression Do?

For the model above, a generic regression tool can:

1. Help 'find' $m()$, to a limited extent;
2. Estimate β, including its degree of uncertainty;
3. Characterize the behavior of ϵ_i and estimate the relevant parameters.

Which of the 3, do you think, is the most difficult task?

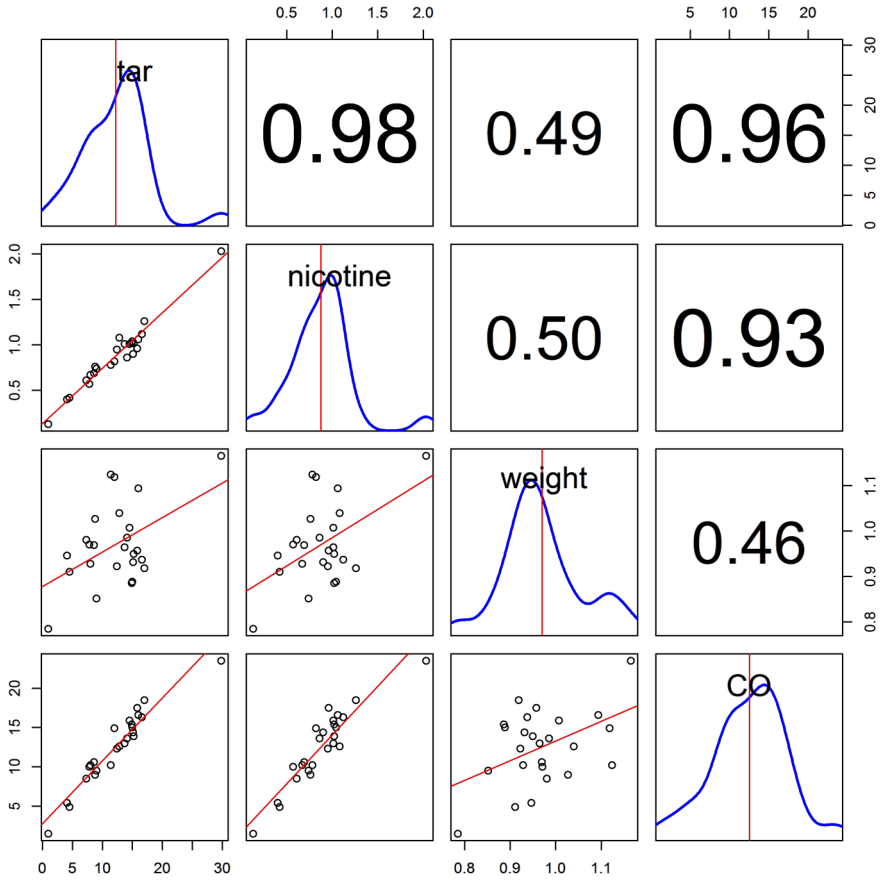
What Can Regression *Not* Do?

Regression cannot check your science or your logic.

You might get hints that something is wrong, but it is sometimes easy to miss those hints, especially when complacent.

Example: Carbon Monoxide from Cigarettes

```
cig = read.table("../Datasets/Amstat_cigarettes.csv", header = T, as.is = T,
    row.names = 1)
pairsPlus(cig)
```



In this dataset, researchers wanted to see what affects CO emissions during cigarette smoking.

All covariates seem correlated with the outcome and with each other.

On the next slide, we check univariate regression summaries.

Example: Carbon Monoxide from Cigarettes

```
summary(lm(CO ~ tar, data = cig))$coef
```

```
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept)    2.743    0.67521   4.063 4.812e-04
## tar            0.801    0.05032  15.918 6.552e-14
```

```
summary(lm(CO ~ nicotine, data = cig))$coef
```

```
##             Estimate Std. Error t value  Pr(>|t|)
## (Intercept)    1.665     0.9936   1.675 1.074e-01
## nicotine      12.395     1.0542  11.759 3.312e-11
```

```
summary(lm(CO ~ weight, data = cig))$coef
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -11.80      9.722  -1.213  0.23733
## weight         25.07      9.980   2.512  0.01948
```

All are significant separately. Tar and nicotine are very strongly significant. What happens when we throw them all into the model?

Example: Carbon Monoxide from Cigarettes

```
summary(lm(CO ~ tar + nicotine + weight, data = cig))$coef
```

```
##             Estimate Std. Error  t value  Pr(>|t|)
## (Intercept)   3.2022     3.4618  0.92502 0.3654643
## tar           0.9626     0.2422  3.97357 0.0006921
## nicotine     -2.6317     3.9006 -0.67469 0.5072343
## weight       -0.1305     3.8853 -0.03358 0.9735268
```

Now only tar is significant. The two others are completely null. And even tar's significance is greatly reduced. The t statistic indicating its signal-to-noise ratio, came down from nearly 16 when modeled alone, to < 4 in the multiple regression.

We can just take this at face value, or try to understand how it happened.

Looking back at the pairs plot, we see that tar and nicotine are *very* highly correlated. **It is almost as if we put two identical copies of the same variable in the model.**

In regression, using highly **collinear** covariates like these two is a no-no. Now we look at the math.

The Linear Regression Estimate in Matrix Form

Multiple linear regression is simply

$$y = \mathbf{X}\beta + \epsilon,$$

where $\mathbf{X}\beta$ is a matrix-vector product.

The least-square estimate of β is

$$\hat{\beta} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T y,$$

where

- ^{T} is the matrix transpose, and $^{-1}$ indicates its inverse;
- $\mathbf{X}$ is an $n \times p$ matrix – with p being the number of covariates $+1$ for the intercept;
- y and ϵ are n-length vectors.
- **What is the dimension of the matrix $\mathbf{X}^T\mathbf{X}$?**

Collinearity

We'll inspect the bowels of this formula later on, but right now recall your undergrad linear-algebra course. A matrix whose determinant is zero, has no inverse. It is essentially like trying to divide by zero. What matrices have a zero determinant? Degenerate matrices, i.e., with linear dependence, i.e., with perfect collinearity.

Suppose we artificially add a linear combination of two existing covariates. What will the model do?

```
cig$nicwt = cig$tar + 10 * cig$weight
summary(lm(CO ~ tar + nicotine + weight + nicwt, data = cig))
```

```
##
## Call:
## lm(formula = CO ~ tar + nicotine + weight + nicwt, data = cig)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8926 -0.7827  0.0043  0.9289  2.4508
##
## Coefficients: (1 not defined because of singularities)
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.202      3.462    0.93  0.36546
## tar            0.963      0.242    3.97  0.00069 ***
## nicotine      -2.632      3.901   -0.67  0.50723
## weight        -0.130      3.885   -0.03  0.97353
## nicwt             NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.45 on 21 degrees of freedom
## Multiple R-squared: 0.919,   Adjusted R-squared: 0.907
## F-statistic:   79 on 3 and 21 DF,  p-value: 1.33e-11
```

Collinearity

We seem protected: the redundant covariate is NA'ed. But consider this:

```
cig$nicwt = cig$tar + 10 * cig$weight + rnorm(25)
round(cor(cig), 2)
```

```
##           tar nicotine weight   CO nicwt
## tar      1.00     0.98   0.49 0.96  0.98
## nicotine 0.98     1.00   0.50 0.93  0.95
## weight   0.49     0.50   1.00 0.46  0.63
## CO       0.96     0.93   0.46 1.00  0.94
## nicwt    0.98     0.95   0.63 0.94  1.00
```

```
summary(lm(CO ~ tar + nicotine + weight + nicwt, data = cig))$coef
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.8376     3.6108  1.0628   0.3005
## tar           0.6742     0.4686  1.4386   0.1657
## nicotine     -2.2131     3.9882 -0.5549   0.5851
## weight       -3.8768     6.5098 -0.5955   0.5582
## nicwt         0.2808     0.3890  0.7219   0.4787
```

We added a wee bit of random noise to the bogus, linearly-dependent `nicwt`. The regression is now legit! And `tar`'s significance is dead.

If we received `nicwt` as a different-named, stand-alone variable, and if we are truly awful users of black-box regression, we might conclude there's nothing useful in this dataset. Generally speaking, there is no built-in protection - unless we ourselves check things. This is the recurring theme in regression.

What killed `tar`'s significance?

Bottom line: avoid collinearity.

The rules of thumb might vary by application, but 0.7, 0.8 or even less are typical thresholds for the maximum "allowable" correlation between covariate pairs. ($r = 0.7$ roughly translates to: the two covariates together have 1.5 times the useful information of one of them alone).

For collinearity between one covariate – say, covariate j – and all the rest combined, a common diagnostic is **the Variance Inflation Factor:**

$$VIF_j = \left(1 - R_j^2\right)^{-1},$$

where R_j^2 is the R^2 obtained by regressing j (as the outcome) on all other covariates.

Indeed, VIF_j is roughly the square of the factor by which everyone's standard errors get inflated, due to the addition of covariate j to the model. Montgomery et al.'s regression textbook suggests avoiding any covariates with a VIF over 5. That actually sounds a bit lenient to me.

Another way to look at it: $1 - R_j^2$ is the proportion of j's variance that might contain new information.

Micro-Break 1

Play with this a bit. See, e.g.,

- how big must the random noise be, to protect `tar`'s significance from being lost?
- Can you find the VIFs of our cigarette dataset? (hint: look online for an R function :)
- what happens if you add a square term, such as `tar^2` (how do you do this, anyway?)
- So… which variable is better to use in our model for CO, tar or nicotine?
- Questions? *Online* questions?

Now some Serious Self-Work, followed by a Real Break:

Download the dataset called `boston.csv`. *This seems to be a famous modeling (and machine-learning!) guinea-pig. Even better, it comes from the world of econometrics, that many students have indicated a wish to see more examples of during the class. I am not familiar with the dataset, even though it touches upon a major interest in my last UW job: the impact of air pollution.*

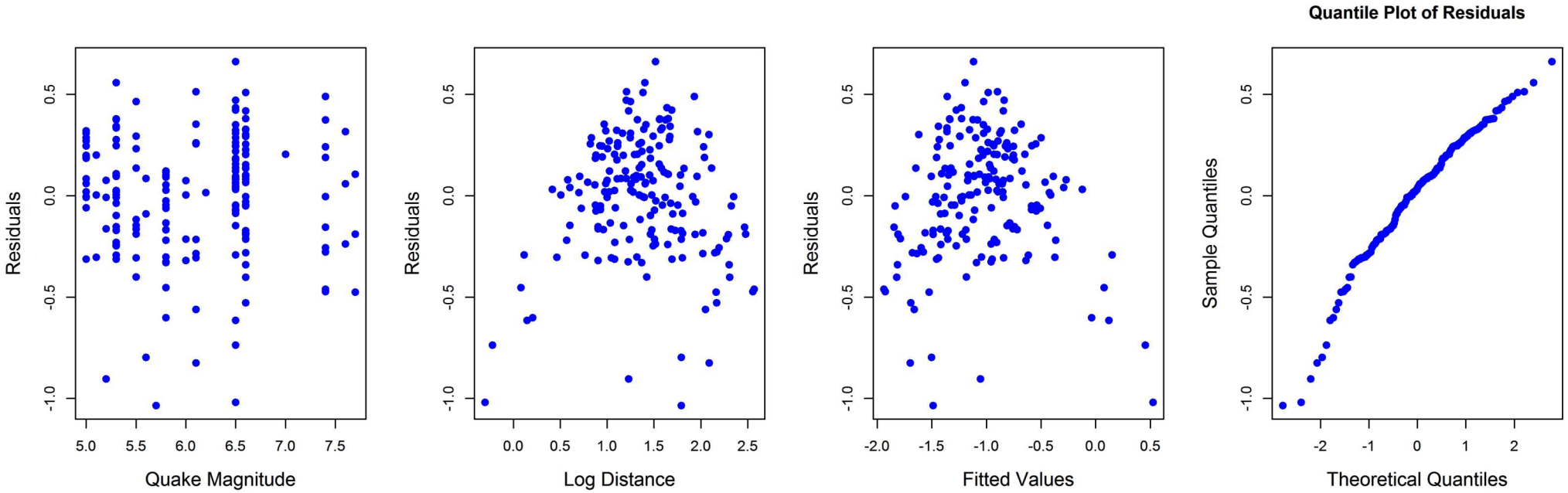
The version you downloaded is a bit simplified from the original. It contains Census-tract level data from greater Boston, mostly based on the 1970 Census. The outcome is the tracts' median home value (already given as such in the dataset). The original question of interest is whether average oxides-of-nitrogen (NOx) pollution levels affect home values. But we shall treat the problem as a free-form regression. The RHS shows the Data dictionary.

What I'd like you to do, is start hacking at it - descriptives, data quality/sanity, thinking about transformations, exploring collinearity, etc. etc. If all goes well, this will be an ongoing self-study dataset for several weeks.

- *crime: annual crime rate*
- *biglots: area with >25k sq.ft. lots (%)*
- *indust: non-retail business area (%)*
- *river: tract touching Charles River? (binary: 0 or 1)*
- *nox: estimated annual average ambient Nox (ppb)*
- *rooms: average number of rooms in homes*
- *jobdist: weighted-average distance from major job centers*
- *tax: property-tax rate per $10k*
- *teachratio: average teacher/student ratio*
- *lowSES: proportion of low-SES residents (%)*
- *homeval: believe it or not, it's in $k.*

Patterns in Residuals and their Meaning

We return to more widely-known, 'classic' regression diagnostics - and to last week's earthquake example.



A good, "finished" model should show no trends or other patterns in these plots. Meaning: all useful information about the outcome, is already in the model - and all that's left is **"White Noise"**.

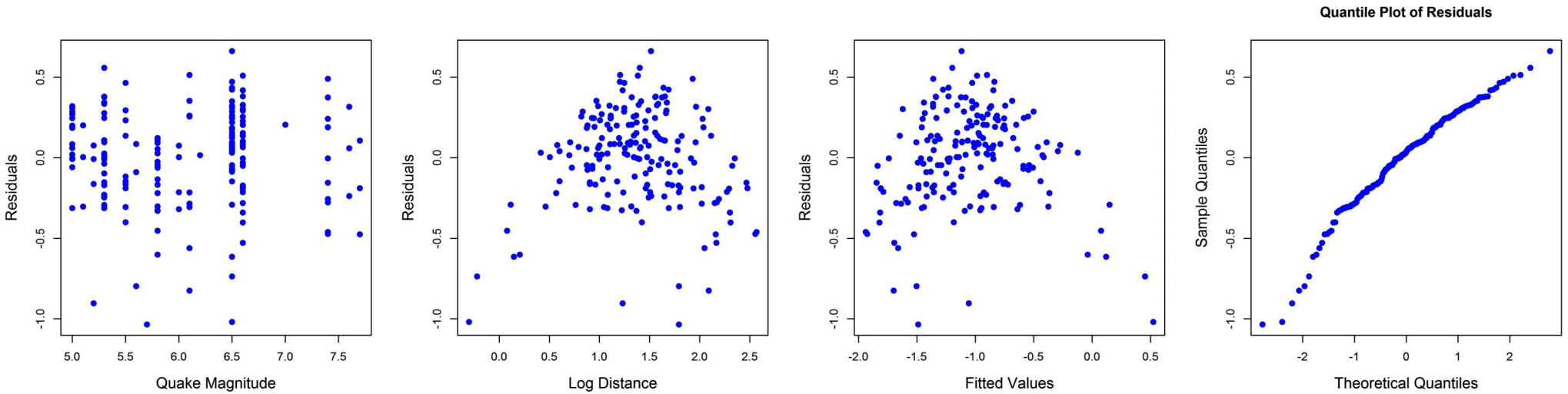
Do you see patterns? And what can we do about them?

Residuals Deviating from “White Noise”

In a perfect statistical-regression world, residuals are

- Normally distributed and without gross outliers
- **Pattern-free,** i.e., devoid of:
- visible dependence on any covariate in (or out of) the model
- ditto, w.r.t. the fitted values
- ditto, w.r.t. any temporal or order aspect of data acquisition
- Patterns can be:
- any trend, linear or not
- any expansion or contraction in the residual’s variability (“bandwidth”)
- anything funny that catches your eye. **Your eye is an amazing pattern-recognition device. Trust it.**

So…. Patterns Anyone?



The residuals have an asymmetric, long lower tail. These low outliers seem to have the strongest pattern **vs. distance.**

What do we see? The observations closest to the epicenter (=smallest distances) are **all** substantially over-predicted by the model. There are other such over-predictions scattered about, but this is the most clear-cut pattern.

Micro-Break 2

Any ideas what the patterns in the earthquake-dataset residuals might mean?

Also, look up **heteroskedasticity.**

Some solutions to residual patterns will be shown in the course. Right now, we first learn to identify them.

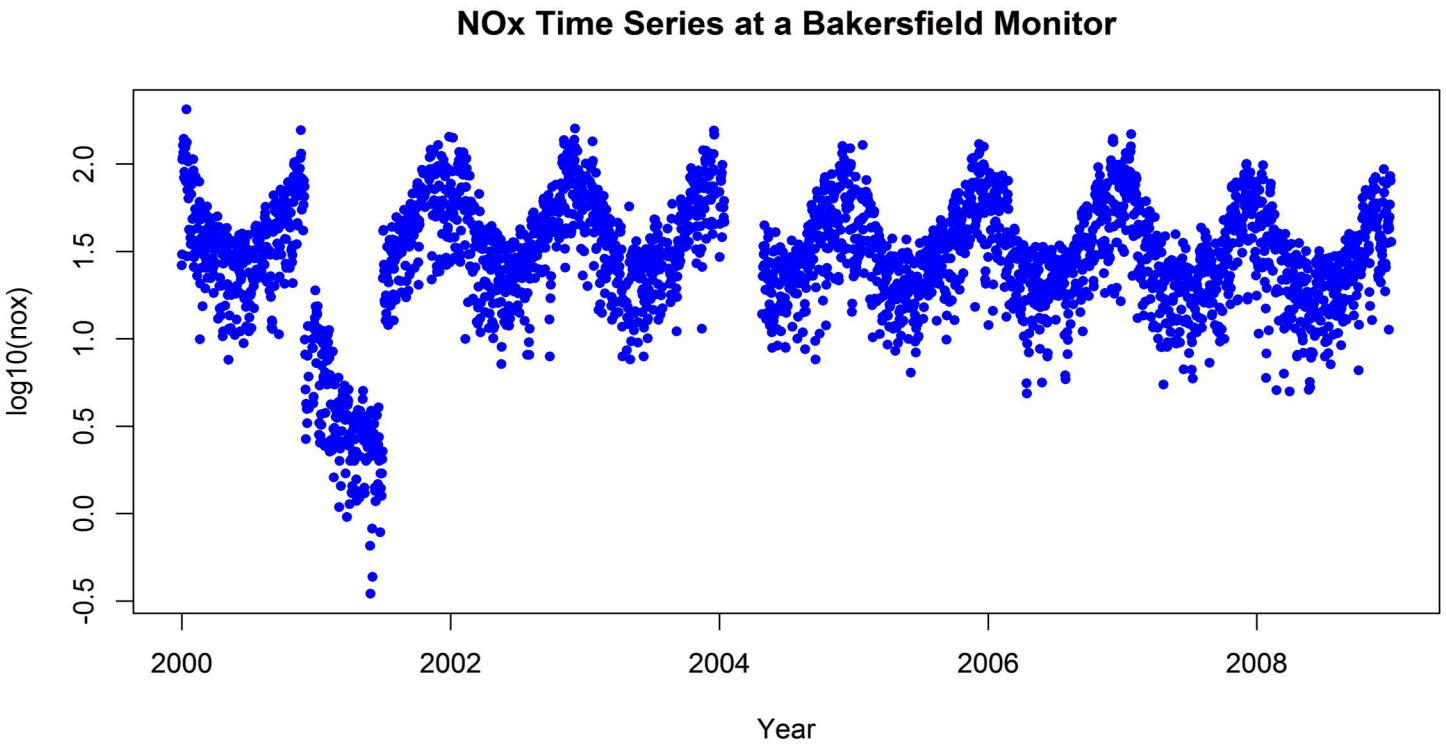
Questions? *Online* questions?

Outlier Triage: Category 1

Because regression estimates are sensitive to outliers (*why are they sensitive?), we often need to take care of outliers.

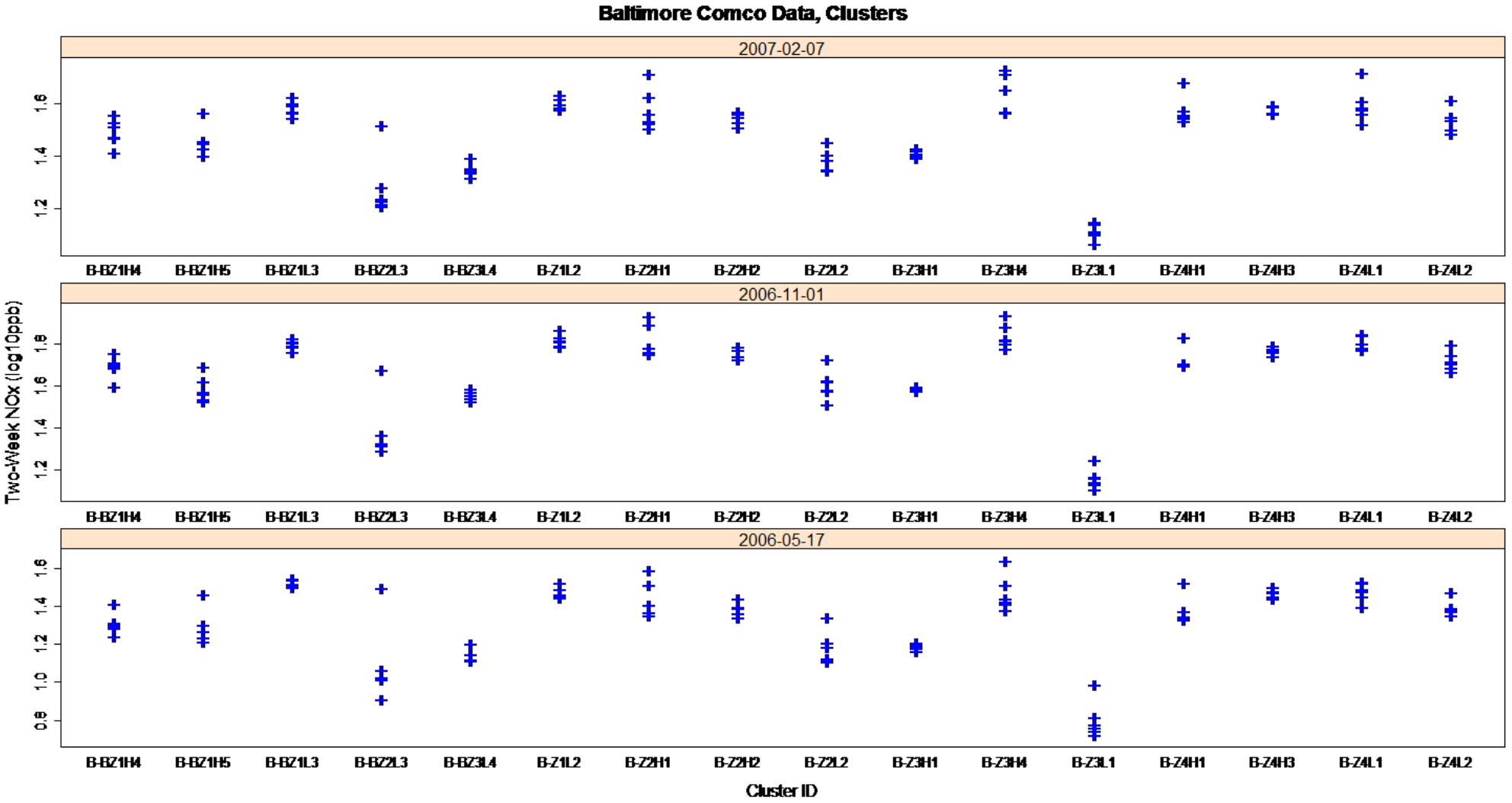
Conceptually, I divide outliers (whether in regression or elsewhere) into 3 groups:

1. Real data errors. Action: remove.



Outlier Triage: Category 2

2. Valid observations, but representing a qualitatively different context. Action: depending upon case, might seek a covariate to explain this difference, use robust methods, do a sensitivity analysis - or exclude due to irrelevance.



Outlier Triage: Category 3

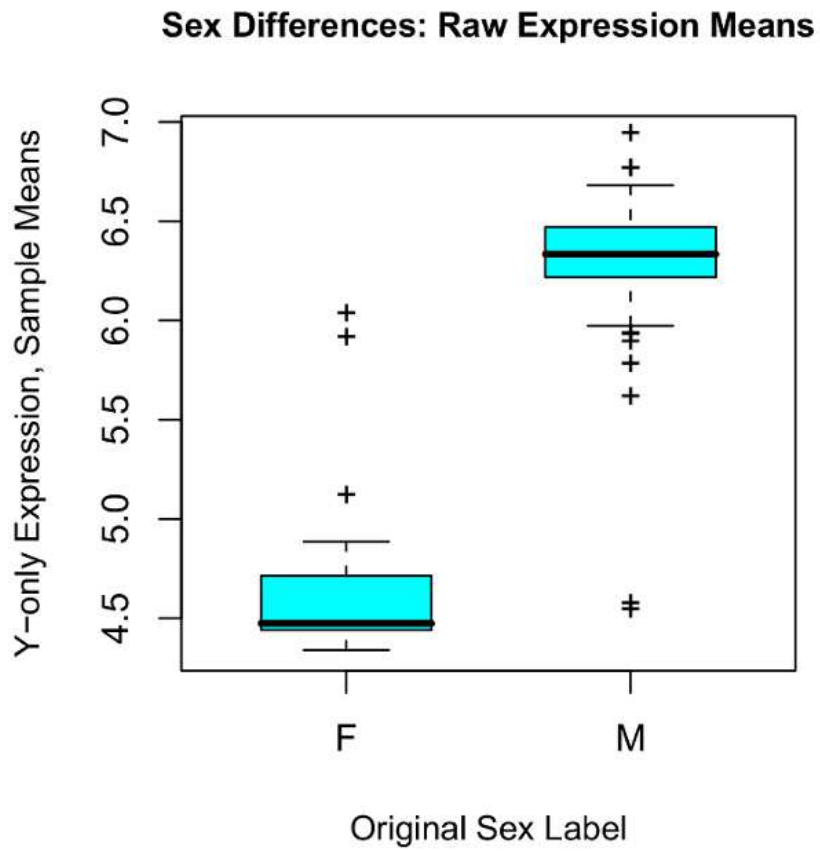
3. Valid observations that are a natural part of your population.

Naturally, this is the most difficult and most common case! There are no magic solutions, but here are some reasonable things you might do depending on context:

- Transform the data
- Sensitivity analysis: run the models both with and without the outliers
- Use more robust methods, either as your main tool or as a sensitivity analysis
- Carefully and methodically exclude, while documenting the exclusion (note: you should **always** document data exclusions)

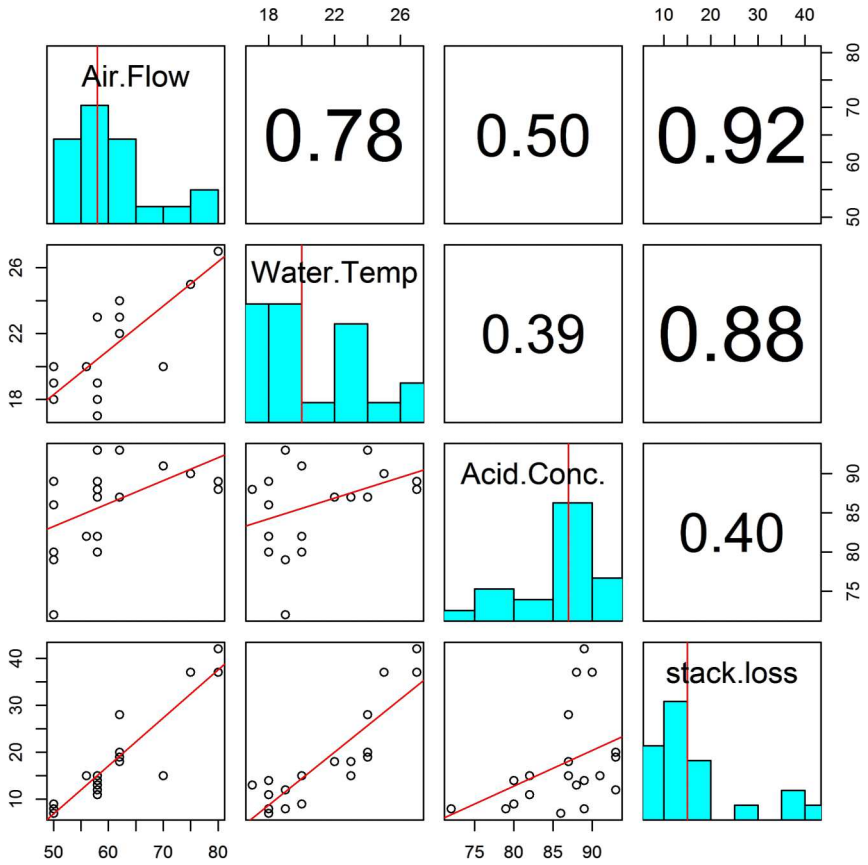
Outlier Triage

The chief rationale for excluding “Category 3” observations, is a well-justified suspicion that **they really are “Category 1/2”** – but lacking a formal paper trail.



Outlier Triage Example: “Stackloss”

How can this be done “carefully and methodically”? There are tons of methods, but here’s a very basic one. We’ll use a dataset that someone called *“The Guinea Pig of Multiple Regression”*. We will do it, assuming we know very little of the science and the context (which is true :)



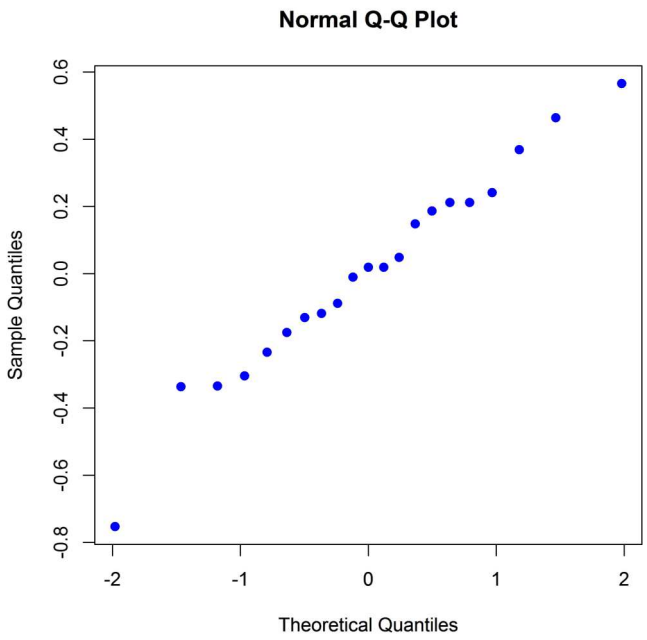
Outlier Triage Example: “Stackloss”

```
stackmod0 = lm(I(stack.loss/10) ~ Air.Flow + Water.Temp, data = stackloss)  # Acid conc. not significant
summary(stackmod0)  # Take a load of this fabulous R^2 !
```

```
##
## Call:
## lm(formula = I(stack.loss/10) ~ Air.Flow + Water.Temp, data = stackloss)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7529 -0.1750  0.0189  0.2116  0.5659
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -5.0359     0.5138   -9.80  1.2e-08 ***
## Air.Flow      0.0671     0.0127    5.30  4.9e-05 ***
## Water.Temp    0.1295     0.0367    3.52   0.0024 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.324 on 18 degrees of freedom
## Multiple R-squared: 0.909,   Adjusted R-squared: 0.899
## F-statistic: 89.6 on 2 and 18 DF,  p-value: 4.38e-10
```

Outlier Triage Example: “Stackloss”

```
qqnorm(stackmod0$resid, pch = 19, col = 4)
```



Ok, there’s a gross negative residual. But can we be methodical about it?

Introducing: Studentized Residuals

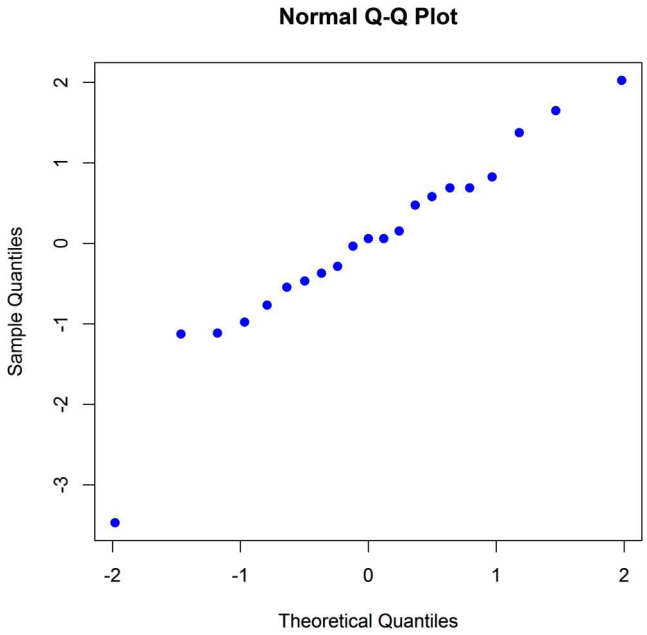
Ordinary residuals are correlated (how do we know?). However, **(Externally) Studentized residuals** are

- independent;
- Each one is t distributed, with $n-p-1$ degrees of freedom.
- **How are they calculated?**
- Run the regression model excluding the data point in question;
- **Predict** the value at said point;
- Subtract from the actual value, and divide by the theoretical std.dev. to normalize
- This is our first example of a **Leave-One-Out** scheme - which will become a standard tool in our work starting in a few weeks.

Testing with Studentized Residuals

Now we can perform a classical hypothesis test.

```
tees = rstudent(stackmod0)  # Told u it's simple...
qqnorm(tees, pch = 19, col = 4)
```




```
c(min(tees), 2 * pt(min(tees), df = length(tees) - 4))
```

```
## [1] -3.470732  0.002924
```

Testing with Studentized Residuals

The worst residual's p-value looks formidable ($p = 0.003$). **However, consider this.**

Even under the t-distribution Null, if we were to generate (say) 1000 numbers, about 3 of them would be as extreme as this – and nothing about it is unusual, these 3 numbers should **not** be excluded.

Here we have only 21 data points, but we still need to account for the fact that **we didn't flag in advance which one we're testing – rather, we took the worst one. In essence, we did 21 t-tests, not one. Hence the name of this practice: Multiple Testing.**

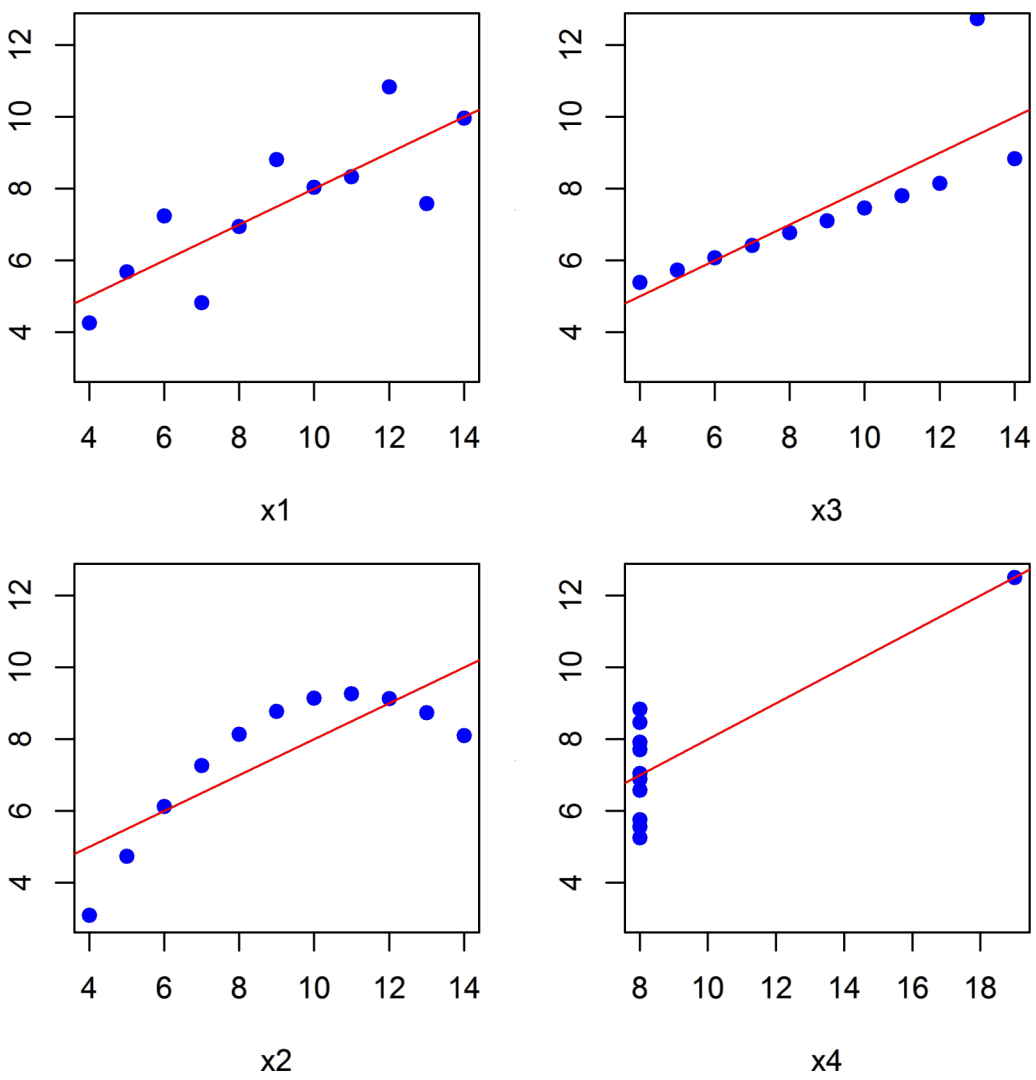
The simplest way to guard against multiple testing is known as **the Bonferroni correction.** It can be done by simply multiplying our p-value by the number of tests. This brings us to 0.06, which is only marginally significant – suggesting that the evidence for throwing the worst data point out the window, is not that overwhelming.

Hey, what is the Null anyway?

The Null for residuals says that all data points can be explained via the specified model, leaving only "White Noise".

Ok, now generate an artificial regression with a few covariates and a large sample - say, $n = 1000$. Look at outliers and see how many of them are found to be "significantly off". **And then we go on break again.**

…Outliers in X Matter Too!



In a prophetic warning againt "black-box" use of regression, Anscombe (1973) presented four artificial simple (univariate) regression examples.

The `summary` *dashboard of these regressions is nearly identical. However, the actual data patterns are quite different, and 'as-is' linear regression is adequate for only one of the four.*

What about the other 3?

Two of them we can already recognize. The third has to do with X's ***support****, and with what is known as* ***influence*** *or* ***leverage****.*

The Problem with Influential Points

**Any model is only valid along the range of covariate values for which we have observations. Extending beyond it is a risky affair.

When making predictions for single observations, the **prediction interval** will reflect the increasing uncertainty: it will expand as our X values become extreme. However, when we only look at effect estimates $\hat{\beta}$, there is no direct indication for their underlying X distribution, and how individual data points affected the estimates.

In the most extreme case, all X values are identical - and we cannot build a regression model at all for that variable. If you look back at the fourth Anscombe example, **it is only a single data point away from this situation. The entire regression model rests upon the shoulders of this single point.**

Not good.

Measures of Influence

Back to some matrix equations. We define **the Hat matrix** $\mathbf{H}$:

$$\hat{y} = \mathbf{X}\left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T y = \mathbf{H}y$$

It is called so, because it "puts the hat on y" (that's as far as statistician humor goes:). In words, $\mathbf{H}y$ is simply the vector of fitted values.

It turns out the the diagonal of $\mathbf{H}$ is a great measure for how much **leverage** the data points have.

The amount of impact that a single point i has on $\hat{\beta}$ is a function of its Studentized residual, and its Hat-diagonal value H_{ii}. There are several metrics trying to gauge this – the most famous of which is **Cook's D** for point i: the normalized square distance, in parameter space, by which i moves the $\hat{\beta}$ vector.

The sum $\mathbf{H}$'s diagonal is exactly p. Therefore, data points with Hat-diagonal values larger than about $2p/n$ are considered worthy of being closely watched. As n increases, the bar for caution can be raised somewhat.

Let us try and calculate the Hat-diagonals for the four anscombe examples - first "by hand", and then via a designated R function, which will lead us to further influence diagnostics.

Non-Normality: Why Regression Estimates are Fairly Robust

The most common mistake in regression diagnostics, is getting freaked out over non-Normality of residuals. Statisticians - even some who are extremely strict on most matters – enjoy this opportunity to switch roles for once, telling others it's not a big deal! (see, e.g., Lumley et al. 2002 on the class website) How come? Recall our parameter-estimate formula:

$$\hat{\beta} = \left(\mathbf{X} -^{T} \mathbf{X}\right)^{-1} \mathbf{X}^{T} y.$$

This is a matrix-vector product; let's simplify it by defining the matrix $\mathbf{G} = \left(\mathbf{X}^{T}\mathbf{X}\right)^{-1}\mathbf{X}^{T}$. The estimate of a single parameter is

$$\hat{\beta}_j = \left(\mathbf{G}y\right)_j = \sum_{i=1}^{n} G_{ji} y_i.$$

The estimate is just a linear combination of the observations! In fact, it is a weighted average of sorts, because the inverse-matrix part acts as a denominator, balancing out the magnitude of the terms in the numerator.

The Central Limit Theorem (CLT) which you've learned in StatR101, guarantees that arithmetic means of i.i.d. random observations (under fairly lenient conditions) are asymptotically normal. It turns out that the CLT also holds for a rather broad category of *weighted* averages, including regression estimates.

Bottom line: the $\hat{\beta}$s are asymptotically normal, even if the random part of the observation is not normal. The bare-minimum conditions for the CLT to hold are two finite moments (expectation and variance).

Using the Coefficients' t-statistics and p-values

…that being said, as you've seen in StatR101, it takes time for averages to become "close enough" to Normal. How much time depends on the original distribution from which the averages are calculated.

Now, the p-values in the regression `summary` are based on the t distribution (with $n-p$ degrees of freedom), which is in turn based on the Normality assumption. It is likely that if the true noise distribution has heavier tail(s), these p-values are **optimistic** - meaning that the reality is more Null (=less "significant") than they suggest.

If we have time today, we will try and simulate a large number of regressions to examine this - getting a head start on the **starred HW1 question 6.** In any case, consider this: - Unless n is huge or the noise is "really" Normal, the true $\hat{\beta}$ distribution has heavier tails than predicted by the t distribution; - The cutoffs 0.05, 0.01 or 0.1 are arbitrary to begin with; - *Rejecting the Null does not mean that the best alternative is precisely adding this particular covariate* - It all means we should **use the p-values with a grain of salt.**

It is more important to understand the t-statistic for what it is: **an estimate of the covariate's signal-to-noise ratio.** When the signal is roughly $> 3 \times$ the noise, it is quite likely that the covariate is useful/meaningful. And more so as the ratio increases. if signal $<$noise, it is essentially useless. In between - oh well…