R: Models for Clustered and Correlated Data

Mary Ryan
@marym_ryan

University of California, Irvine
Presented at NICAR 2019

March 7-10

What's wrong with linear regression?

"Essentially, all models are wrong, but some are useful"

— George Box, *Empirical Model-Building and Response Surfaces*, pg. 424

What's wrong with linear regression?

"Essentially, all models are wrong, but some are useful"

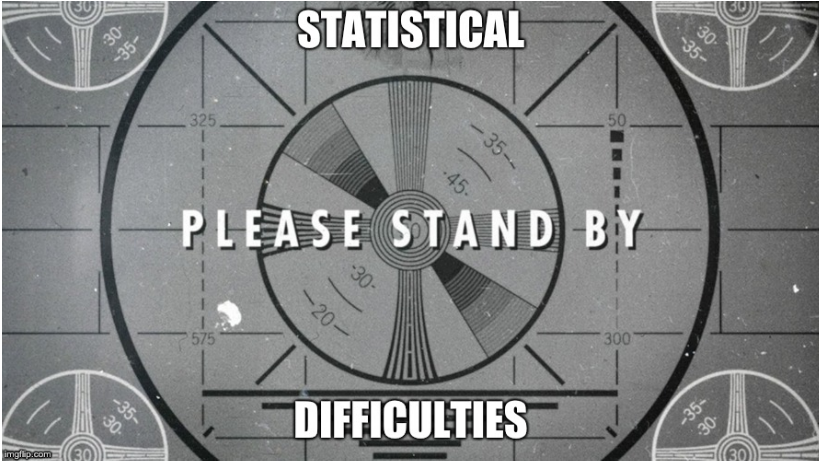
— George Box, *Empirical Model-Building and Response Surfaces*, pg. 424

Other iterations:

- "Remember that all models are wrong; the practical question is **how wrong do they have to be to not be useful**" (*Empirical Model-Building and Response Surfaces*, pg. 74)
- "The most that can be expected from any model is that it can supply **a useful approximation to reality**: All models are wrong; some models are useful" (*Statistics for Experimenters*, pg. 440)

What's wrong with linear regression?

- Regular linear regression assumes each datapoint is *independent* of each other
- What if we have multiple datapoints for each person/school/hospital/location we're measuring?


STATISTICAL
PLEASE STAND BY
DIFFICULTIES
imgflip.com

Ordinary Least Squares (OLS)

- If we have data $\boldsymbol{X}$ and response $\vec{Y}$, we can find our regression coefficients through:

$$\boldsymbol{X}^T(\vec{Y} - \boldsymbol{X}\vec{\beta}) = 0$$

- Requires all observations are independent of one another
- Implement this using the `lm()` function

Iteratively Reweighted Least Squares (IRLS)

- Put a weight on the equation that estimates our coefficients
  - Accounts for the fact that we are no longer dealing with completely independent data points

$$\boldsymbol{X}^T \boldsymbol{W}(\vec{Y} - \boldsymbol{X}\vec{\beta}) = 0$$

- Implement this using the `gee()` function from the gee package
  - Can also use `lme()` from the `nlme` package, and `lmer()` from the `lme4` package


AND NOW BACK TO
OUR REGULARLY
SCHEDULED
PROGRAMMING

The gee() Function

- gee stands for Generalized Estimating Equation

Like with the lm() function, there are several arguments we need to fill in:

- **Formula**: this is the same formula you would plug in for lm(), of the form response $\sim$ variable1 $+$ variable2 $+ \dots$
- **id**: this is a variable in your dataframe that identifies your clusters. If I have 12 patients with 3 datapoints each, each datapoint needs to have something that tells us which patient it is coming from. Usually this is done as the very first column of your dataframe, where the id can be a number or a string.
- **data**: like with lm(), this is the name of your dataframe

The gee() Function

- **family**: the default for this argument is "gaussian"", which just means Normal. We generally won't put anything in for this argument unless we're dealing with binary data (we'll see this later).
- **corstr**: this tells the function how we want to do our weights.
  - There are 3 main options for this:

  1. "independence": this is the default and will get us lm()
  2. "exchangeable": this gives us random intercepts
  3. "AR-M": this gives us random slopes and random intercepts. With this though, we also need to specify a "Mv" argument, which will be 1.

Side note: two other popular functions for modeling longitudinal data are lme() from the nlme package, and lmer() from the lme4 package. These work similarly to the gee() function but have slightly different synatx and technically require stronger statistical assumptions to use. I generally stick to gee().

Robust Variance

- Sometimes different clusters/groups will have different variances
- Robust variance is a post-model fix-em-up to account for different cluster variances
  - If the variances really don't differ by cluster, you'll get something pretty close to the regular varaince
  - If they do differ, then this will help fix the variance
- Robust variance doesn't perform well when there are fewer than 50 clusters
  - If you have fewer than 50 clusters, it's safer to go with the regular variance because at least we know why it's wrong

Example 1: Median Housing Value in Texas

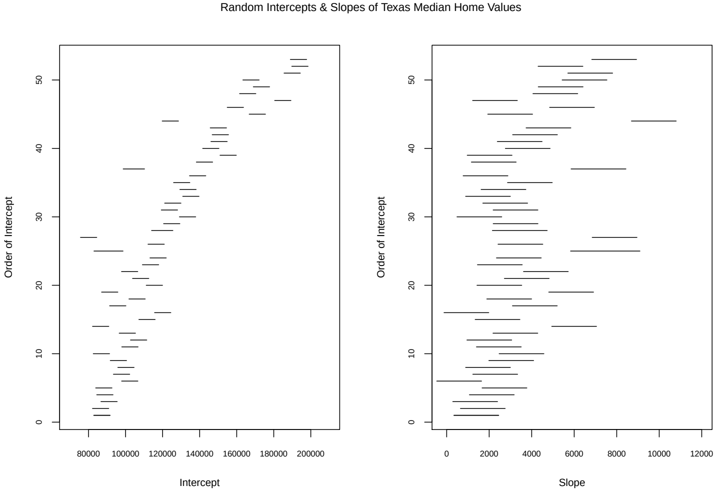
- Housing dataset (select variables)
  - `countyID`: ID number for each county
  - `countyName`: Name of county
  - `yrsSince2009`: Number of years since 2009
  - `Median`: Median housing value in county
  - `totPop18plus`: Population 18 years or older
  - `BAtotPop18plus`: Population 18 years or older with at least a Bachelors
  - `BApctTotPop18plus`: Percentage of population 18 years or older with at least a Bachelors
- 53 unique counties
- 420 unique observaations
- 8 years of data

Question: how does the percentage of Bachelors-holders in a distrct affect mean housing value?

Example 1: Random Slopes and Intercepts

- Forest plot: represents what the intercepts and slopes would be if we performed individual lm()s on the data from each subject separately
- If the data were truly independent, all the lines would be overlapping
  - If plot on left shows non-overlapping lines: case for random intercepts
  - If plot on right shows non-overlapping lines: case for random slopes

Example 1: Random Slopes and Intercepts



Example 1: Random Slopes and Intercepts

- Helps us determine what kind of *correlation structure* we want to use in our model
  - No overlapping on either plot ⇒ Independence structure (`corstr="independence"`)
  - Overlapping on left but not on right ⇒ Exchangeable correlation structure (`corstr="exchangeable`)
  - Overlapping on both left and right ⇒ AR-1 correlation structure (`corstr = "AR-M; Mv = 1`)

Example 1: Modeling with Independent Structure

```
house.indep <- gee( Median ~ BApctTotPop18plus + yrsSince20
                    id = countyID,
                    data = housing,
                    corstr = "independence" )
```

```
summary( house.indep )$coef
```

```
##                    Estimate Naive S.E.   Naive z Robust
## (Intercept)       45973.153  2601.1356 17.674262   4757.
## BApctTotPop18plus  5606.520   151.7380 36.948698    325.
## yrsSince2009       2399.319   355.2078  6.754691    253.
```

Example 1: gee() vs lm()

```
summary( house.indep )$coef
```

```
##                        Estimate Naive S.E.   Naive z Robust
## (Intercept)           45973.153  2601.1356 17.674262   4757
## BApctTotPop18plus      5606.520   151.7380 36.948698    325
## yrsSince2009           2399.319   355.2078  6.754691    253
```

```
summary( lm(Median ~ BApctTotPop18plus + yrsSince2009, data
```

```
##                        Estimate Std. Error   t value      Pr
## (Intercept)           45973.153  2601.1356 17.674262  1.4025
## BApctTotPop18plus      5606.520   151.7380 36.948698 1.32680
## yrsSince2009           2399.319   355.2078  6.754691  4.8255
```

Example 1: Modeling with Exchangeable Structure

```
house.exch <- gee (Median ~ BApctTotPop18plus + yrsSince20(
                   id = countyID,
                   data = housing,
                   corstr = "exchangeable" )
```

```
summary( house.exch )$coef
```

```
##                         Estimate Naive S.E.    Naive z Robust
## (Intercept)           109779.313  5215.7652 21.047595    545(
## BApctTotPop18plus       1053.461   240.0832  4.387899     355
## yrsSince2009            3399.258   131.3704 25.875381     233
```

Example 1: Modeling with AR-1 Structure

```
house.ar1 <- gee( Median ~ BApctTotPop18plus + yrsSince2009
                  id = countyID,
                  data = housing,
                  corstr = "AR-M",
                  Mv = 1 )
```

```
summary( house.ar1 )$coef
```

```
##                      Estimate Naive S.E.   Naive z Robus
## (Intercept)       116922.8807  5411.9154 21.604713    381
## BApctTotPop18plus    473.5751   226.6894  2.089092     18
## yrsSince2009        3982.7697   375.6238 10.603080     27
```

Example 2: California API Scores

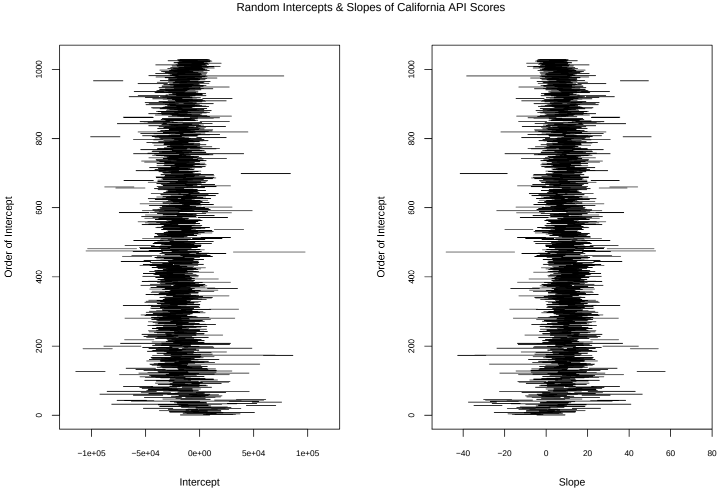
- API dataset (select variables)
  - `CDS`: County/District/School code
  - `RTYPE`: Record Type: (D=District, S=School, X=State)
  - `DNAME`: District name
  - `API`: Base API score
  - `PCT_AA`, `PCT_AS`, `PCT_HI`: Percentage of African American, Asian, and Hispanic students
  - `P_EL`: Percent English learners
  - `MEALS`: Percentage of Students Tested that are eligible for Free or Reduced Price Lunch Program

Example 2: California API Scores

- 1047 unique school districts
- 7178 total observations
- 7 years of data

Question: how does charter status affect district API score?

Example 2: Forest Plot



Example 2: Modeling with Exchangeable Structure

```r
summary( api.exch )$coef
```

```
##                  Estimate   Naive S.E.    Naive z  Robus
## (Intercept) -2.365369e+04 793.17629852 -29.821484 811.70
## PCT_AA      -2.759615e+00   0.12720559 -21.694131   0.22
## PCT_AS       1.453824e+00   0.09589456  15.160647   0.07
## PCT_HI      -7.118761e-01   0.05567068 -12.787273   0.06
## MEALS       -1.767373e+00   0.03945641 -44.793051   0.05
## P_EL         8.109702e-01   0.08788002   9.228152   0.10
## year         1.220614e+01   0.39493962  30.906353   0.40
```

Example 3: Childhood Wheezing and Maternal Smoking

- Ohio dataset
  - `resp`: an indicator of wheeze status (1=yes, 0=no)
  - `id`: a numeric vector for subject id
  - `age`: a numeric vector of age, 0 is 9 years old
  - `smoke`: an indicator of maternal smoking at the first year of the study
- 537 unique subjects
- 2148 total observations

Question: how does maternal smoking affect wheezing?

Example 3: Method

- Binary response, so using logistic regression
  - binary response transformed using logit function:
  - regression tells us about variables changing the *log-odds* of response, instead of the mean

$$logit(\mu) = ln(\frac{\mu}{1-\mu})$$

Example 3: Modeling with Exchangeable Structure

```
ohio.exch <- gee( resp ~ age + smoke,
                  id = id,
                  data = ohio,
                  family=binomial(link='logit'),
                  corstr="exchangeable",
                 silent = T )
```

```
summary( ohio.exch )$coef
```

```
##               Estimate Naive S.E.    Naive z Robust S.E.
## (Intercept) -1.8804277 0.11483941 -16.374411  0.1138929
## age         -0.1133850 0.04354142  -2.604073  0.0438553
## smoke        0.2650809 0.17700086   1.497625  0.1777465
```

Example 3: Modeling with Exchangeable Structure

```r
glmCI.long( ohio.exch, robust = T )
```

```
##               exp( Est ) robust ci95.lo robust ci95.hi rob
## (Intercept)       0.1525         0.1220         0.1907
## age               0.8928         0.8193         0.9729
## smoke             1.3035         0.9201         1.8468
##               robust Pr(>|z|)
## (Intercept)            0.0000
## age                    0.0097
## smoke                  0.1359
```

Example 3: Modeling with AR-1 Structure

```
ohio.ar1 <- gee( resp ~ age + smoke,
                  id = id,
                  data = ohio,
                  family=binomial(link='logit'),
                  corstr="AR-M",
                  Mv = 1,
                 silent = T )
```

```
summary( ohio.ar1 )$coef
```

```
##                Estimate Naive S.E.    Naive z Robust S.E.
## (Intercept) -1.8981575 0.10961955 -17.315867  0.1146781
## age         -0.1147505 0.05586065  -2.054229  0.0449352
## smoke        0.2438312 0.16620395   1.467060  0.1798310
```

Example 3: Modeling with AR-1 Structure

```
glmCI.long( ohio.ar1, robust = T )
```

```
##             exp( Est ) robust ci95.lo robust ci95.hi rob
## (Intercept)     0.1498         0.1197         0.1876
## age             0.8916         0.8164         0.9737
## smoke           1.2761         0.8971         1.8154
##             robust Pr(>|z|)
## (Intercept)          0.0000
## age                  0.0107
## smoke                0.1751
```