Repeated Measures

NRES 710

Last compiled: 2024-09-19

## Repeated Measures Design

In the last few classes, we have been talking about mixed effects models and randomized block designs. Today we are going to talk about a particular kind of randomized block design model that is called a ‘repeated measures design’.

**Repeated Measures**

The original application of this involved giving a drug/treatment to an individual, and the response was measured *repeatedly* over time in that individual.

In truth, there is not that much differenc between a repeated measures design and a randomized block design. They are both mixed-effects models.

* Randomized Block Design: random effect is block.
* Repeated Measures: random effect is individual

There is one slight difference: **Autocorrelation**. Because of the particular nature of the repeated measures design, autocorrelation is a common attribute in those designs. One thing we have learned this semester is we often have to be pretty explicit in telling R what to do for us. This is no different when we fit a repeated measures model in R: we have to tell R very carefully to handle this autocorrelation. And we’ll learn how to do that today.

Originally designed for drug treatment trials where individuals are measured over time. But this can be used for any study where we have repeated measures of a subject – either repeated measures in *space* or repeated measures in *time* – that might lead to autocorrelation.

## Example: Pollution in a River

### Single River Example

Let’s go back to an example from earlier in the semester, where we considered how to measure whether a powerplant might be polluting a river. We want to know how pollution levels may dissipate as a function of distance from the source.

**Y – Pollution** **X – distance from source**

We previously discussed this example when we considered the assumption of linear regression; we said the assumption of autocorrelation was likely to be violated depending on how close together your samples are. Samples every 5 m, for example, might be extremely autocorrelated.

**Q:** This is actually not a repeated-measures design. Does anybody know why? You might think…

* Because the samples are taken at different distances? No, this is a good thing, it allows us to measure the distance effect on pollution levels.
* Because pollution might get introduced at different times? No, that is something else… that would be an interaction! Where pollution levels vary by distance and time.

The limitation is that we don’t have a variable here that we can include as a random-effect.

First, we only have *1 river*, and we therefore cannot include river as a random effect in a mixed-effects model. Our model will look something like this:

When we add in the random effect of river, R is going to try to measure what is the variation in pollution level due to river. Some rivers might have naturally high pollution, or naturally low pollution – but if we only have 1 river, we cannot include river as a random-effect like we were doing before.

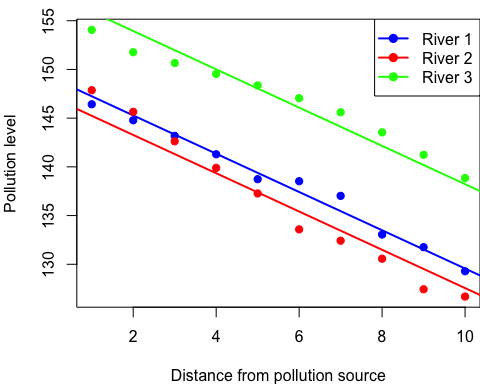
Second, we only sampled the river once, and we did not do repeated samples of each individual location (X), so we cannot include *Time* as a random effect.

### Multiple Rivers

Let’s assume that instead we now have ~3 rivers. Each river has a pollution source and we want to know how pollution dissipates as a function of distance from source among rivers in general. *e.g., Brian draw 3 of these on the board.*

We will run this model:

Graphically, this looks like: *draw graph on board with autocorrelated points for three rivers, each with same slope, but different y-intercepts*



Remember, the reason we are doing the mixed-effects model is because we have variation in pollution (Y) due to river – and we want to account for that river variation.

*We can visualie this with a normally-distributed bell curve over the graph to illustrate river variation*

This is the term in our model – ‘**between-river variation**’.

And the is the noise around our line – this is our ‘**within-river variation**’. By including the river-effect, we have greatly reduced our residual error, and increases our power.

**Q:** Any questions?

So far, this is exactly the same as what we have done previously; nothing new. Same as fields, plots, multiple plots per field, etc.

But the key difference here is that: our data have autocorrelation – that trailing pattern of data within each river. We have autocorrelation, and we want to deal with it. A repeated-measures design can deal with autocorrelation for us.

## Dealing with autocorrelation

In statistics, autocorrelation is dealt with in a **phenomenological** way, as opposed to a **mechanistic** way.

**Phenomenological – model that captures the ‘phenomena’ observed in the data**

**Mechanistic – model that tries to explain the mechanisms by which phenomena arise**

An example: we previously talked about how we might try to capture a non-linear relationship using a quadratic equation. This is a phenomenological model. It can explain curves, but it does explain how the curves arise. Alternatively, you can use a mechanistic model, such as Holling’s Disc equation or Michaelis-Menten equation for enzyme kinetics, that capture the mechanisms that produce non-linear phenomena.

* [Sidenote: Holling’s Disc equation was generated to explain how many prey a predator can kill as a function of prey density. As prey density goes up, predators are limited in how many prey they can capture, handle, and consume. The mechanistic equation generates non-linear dynamics.]
* [Sidenote: Michaelis-Menten equation for enzyme kinetics explains how reaction rate is influenced by substrate concentration; but as substrate concentration goes up, reaction rate increases plateau due to constraints in reactivity. The mechanistic equation generates non-linear dynamics.]

Most autocorrelation functions are not trying to capture the mechanisms by which autocorrelation arises. Rather, it just creates a curve that phenomenologically explains the autocorrelation.

There are two types of autocorrelation models that we work with in statistics.

* **Moving Average**
* **Autoregressive**

These are mechanistic-type models:

* **Moving Average – residuals are a function of previous residuals**
* **Autoregressive – Y is a function of the previous Y**

What’s the difference? It may feel like there isn’t much of a difference, but mathematically there is. We don’t have to dwell too much on it.

In environmental sciences, our autocorrelation is probably autoregressive in nature. For example, growth of individuals, height in 1 year is most likely a function of height in the previous year. This happens a lot, and causes autocorrelation in the system.

Alternatively, moving averages would say that if you were above average in 1 year, you are more likely to be above average in the next year. This is actually less likely in ecology; for example, when we have things like density dependence. If your population is above carrying capacity one year, it often responds and drops below the carrying capacity the next year. It doesn’t experience the moving average, instead it does the opposite, which causes there to be ~less autocorrelation in the system.

Don’t worry about this though – you don’t need to try to figure this out for your data. It doesn’t matter; we just need to account for the phenomena.

## Accounting for Autocorrelation in R

### Truth

Let’s examine the code used to simulate the dataset we will analyze here today. Now we have 10 rivers, distance (km) from a point source, we have some error due to each river being different from the average of rivers (error = 5), the residual error (error within rivers) is 1, beta0 is 150, and beta1 is -2.

As we expect, pollution decreases as you get further and further away from the point source in each river.

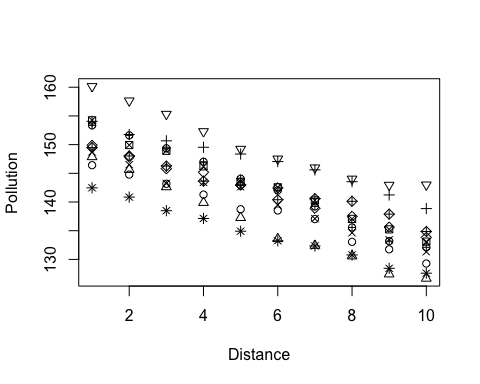
There is autocorrelation in the data! The residual error within each river is a moving average of itself.

You can examine how Truth was simulated at the end of the lecture’s R code.

**Note:** I have never understood statistics so well in my life as when I have had to make all of the data for this class. Simulating data with known parameters causes one to have to understand the processes by which patterns in nature occur, and subsequently how we then fit different statistical models to appropriately estimate those parameters.

**Note 2:** This is a procedure that I use for all of my science papers, particularly when I am fitting or building a model that I am not familiar with. I will make my own data, then I know truth, and then I can make sure the model I am using is measuring truth appropriately. Then I add this into my science paper and say: I know that my estimates are robust because I analyzed simulated data of known values to verify it. If I come up with a wrong estimate, then I may not understand what I am doing.

### Repeated measures model analysis in R  
  
# Read the data  
datum <- read.csv("lecture\_17\_dataset1.csv")  
  
# Examine it  
head(datum)  
  
# Plot it  
plot(Pollution ~ Distance, data = datum, pch = as.numeric(datum$River))



OK, so let’s look at the data. We notice some patterns in the data.

* Some rivers have relatively little pollution, while other rivers have much more.
* We can see the autocorrelation in error within each river, as the points tend to follow eachother.

Let’s jump straight into a linear mixed-effects model:

# Linear model  
library(nlme)  
results <- lme(Pollution ~ Distance, data = datum, random = ~1|River)  
summary(results)

## Linear mixed-effects model fit by REML  
## Data: datum   
##   
## Random effects:  
## Formula: ~1 | River  
## (Intercept) Residual  
## StdDev: 4.576 1.277  
##   
## Fixed effects: Pollution ~ Distance   
## Correlation:   
## (Intr)  
## Distance -0.166  
##   
## Standardized Within-Group Residuals:  
## Min Q1 Med Q3 Max   
## -2.13966 -0.76329 -0.03969 0.75392 2.07607   
##   
## Number of Observations: 100  
## Number of Groups: 10

Let’s first look at the error explained by the random effect term.

* Note that the random-effect formula is ~1|River, and below that we have error due to random variation in the intercept and also residual error. This means that we are specifying the Y-intercept to vary randomly by River; ‘river’ is the random intercept.
* The ‘Residual’ error is the error within rivers; unexplained noise.
* Both these values are similar to Truth.
* And our estimate is close to Truth, and its confidence intervals overlap Truth.
* P-value is highly significant.

Nothing new here. This is the same model we built last class. We have autocorrelation in our data, and we built a model that does not account for it – and we still get unbiased results. This is why I am often not bothered by autocorrelation.

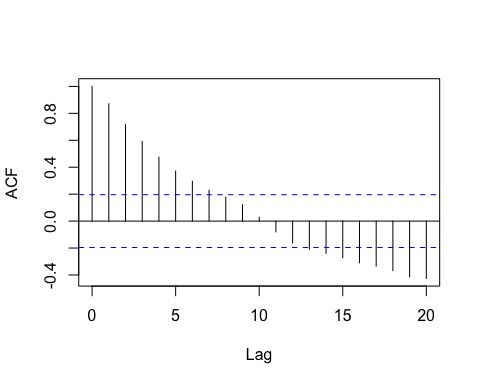
Let’s take a look at the LME function and see how to deal with autocorrelation

# Linear model  
help(lme)

Usage –> ‘fixed’, ‘data’, and ’random –> we have used these so far.

But the specific attribute we want to use is ‘correlation’. This is the argument that you use to introduce/account for autocorrelation in your data!

* ‘An optional ’corStruct’ object describing within-group correlation structure.
* When you have a random effect, it automatically doesn’t look for correlation within rivers. We have to give it something to do that.
* Click on the ‘corStruct’ objects link:
  + corAR1: autoregression process of order 1. The order is an indication of the ‘lag’. If we think back to the ACF plots, this would be a lag of 1. For example:



* In ecology, lags are at time=1. We generally don’t need to worry about higher-order lags, as they are uncommon.
* corARMA: allows you to include ‘moving average processes’ and ‘autoregressive processes’. You can do both, but you have to specify which process it is.
* Ignore the rest.

Let’s do this in R:

# Linear mixed-effects model that accounts for autocorrelation  
results2 <- lme(Pollution ~ Distance, data = datum, random = ~1|River, correlation = corAR1())  
summary(results2)

## Linear mixed-effects model fit by REML  
## Data: datum   
##   
## Random effects:  
## Formula: ~1 | River  
## (Intercept) Residual  
## StdDev: 0.236 4.934  
##   
## Correlation Structure: AR(1)  
## Formula: ~1 | River   
## Parameter estimate(s):  
## Phi   
## 0.9825   
## Fixed effects: Pollution ~ Distance   
## Correlation:   
## (Intr)  
## Distance -0.325  
##   
## Standardized Within-Group Residuals:  
## Min Q1 Med Q3 Max   
## -1.6684 -0.7054 -0.0675 0.5365 2.0115   
##   
## Number of Observations: 100  
## Number of Groups: 10

It doesn’t take any arguments, but it will run a function – so you have to have the two parentheses after ‘corAR1()’.

Our estimates look the same, but our random effects looks weird. It’s supposed to be an effect of 5 for River, but it gave us zero – and it instead said all the error is residual error.

It gives us a new parameter estimate, (phi). This may have meaning, but I honestly cannot explain it to you. That is for true statisticians. But I do know that it is part of the math that generates that autoregressive phenomenon.

It did not change our estimates much, but it did mess up our interpretation of the random effects. Which I don’t much like.

So, since it didn’t change our estimates, it’s for this reason that I often don’t deal with autocorrelation when I run these analyses. However, if a reviewer says you have to, then you can fit this model and do so.

We can use a partial likelihood test to see if this more complicated model better fits the data.

1. *Does the model that has autoregression in it explain the data better?*

# Partial likelihood test  
anova(results2, results)

* **~Yes – it fits the data better**

1. Did it give us a better results?

* Both produces the ~same estimates of the distance effect.
* But the SE for distance doubled! We lost precision…

I am not impressed by the results. Maybe it didn’t work that well because we used the wrong autocorrelation function. Let’s try another.

If we want to use a moving-average, we have to use ‘corARMA()’ function: ‘cor’ ‘AutoRegressive’ ‘Moving Average’. This requires two arguments:

1. p = auto-regressive order
2. q = moving-average order

corARMA(p = 1, q = 0) – same thing as ‘corAR1’ corARMA(p = 0, q = 1) – a moving-average autoregression

# Linear mixed-effects model that accounts for autocorrelation with a moving-average autocorrelation model  
results3 <- lme(Pollution ~ Distance, data = datum, random = ~1|River, correlation = corARMA(p = 0, q = 1))  
summary(results3)

## Linear mixed-effects model fit by REML  
## Data: datum   
##   
## Random effects:  
## Formula: ~1 | River  
## (Intercept) Residual  
## StdDev: 4.588 1.204  
##   
## Correlation Structure: ARMA(0,1)  
## Formula: ~1 | River   
## Parameter estimate(s):  
## Theta1   
## 0.6835   
## Fixed effects: Pollution ~ Distance   
## Correlation:   
## (Intr)  
## Distance -0.191  
##   
## Standardized Within-Group Residuals:  
## Min Q1 Med Q3 Max   
## -2.26868 -0.82138 -0.05013 0.72577 2.16720   
##   
## Number of Observations: 100  
## Number of Groups: 10

* Our random error terms are much better again! Effect of river and residual error terms are much closer to Truth.
* The estimates for the intercept and distance effect are close to Truth.
* Before it gave us ‘phi’ for the autoregressive parameter, but now it gives us ‘theta’ for the moving-average. I don’t know the meaning of this but you are welcome to look it up.

So now we have bent over backwards to account for autocorrelation – and it didn’t change our estimates of river error, residual error, or the distance effect compared to the model.

# Partial likelihood test  
anova(results3, results)

This model again fits our data better than our more simple model. A moving average is better than no autocorrelation.

# Compare the two models with autocorrelation  
anova(results3, results2)

**Q:** Does anybody know why we can’t get a p-value here? **They are both equally complex**. The p-value tests the hypothesis that the null hypothesis that the simpler model is adequate - against the alternative hypothesis that the more complex model is a better fit.

We can compare these models using AIC – we haven’t talked about AIC yet – AIC is a way with which you can compare all models – lower values are better. AIC says that the autoregressive model, results2, is a better model… even though I made the autocorrelation with a moving-average (???). I don’t know why.

**Questions?**

## Truth

Here is the code to create Truth for the two datasets analyzed here today.

################### 'Truth' ####################   
### Lecture 17: code to simulate data for class  
  
### Dataset 1  
# Pollution as a function of distance in three rivers  
# Set the seed for reproducibility  
set.seed(123)  
  
# Rivers  
n\_rivers <- 10  
n\_samples <- 10  
River <- sort(rep(1:n\_rivers, n\_samples))  
  
# Distance from pollution source  
Distance <- rep(seq(1, n\_samples, 1), n\_rivers)  
  
# Error due to river  
RiverError <- rep(rnorm(n\_rivers, 0, 5), each = n\_samples)  
  
# Error within river (residual error)  
# Simulate autocorrelated error - each value using the mean of the previous value  
Error <- matrix(NA, length(Distance), 1)  
Error[1,1] <- rnorm(1, mean = 0, sd = 1)  
for (i in 2:length(Distance)){  
 Error[i,1] <- rnorm(1, mean = Error[i-1, 1], sd = 1)  
}  
  
# Response variable - Pollution  
Pollution <- 150 - 2\*Distance + RiverError + Error  
  
# Save the data  
datum <- data.frame(River = River, Distance = Distance, RiverError = RiverError, Error = Error, Pollution = Pollution)  
  
# Save the data  
write.csv(datum, "lecture\_17\_dataset1.csv", row.names = FALSE)

[–go to next lecture–](lecture_18.html)