# Mixed effects models 2

“Space is the place” - *Sun Ra*

Samuel Robinson, Ph.D.

December 17, 2020

# Motivation

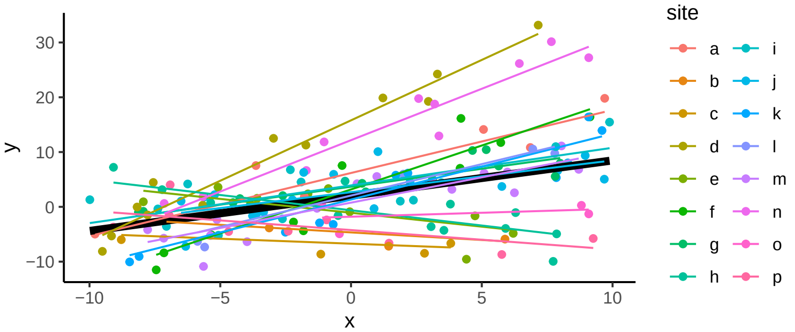
- How do I check if model results are valid?
    - Residual checks
    - Hypothesis testing
- What if my response variable is non-normal?
    - Generalized linear mixed models (GLMMs)
- Sampling over time or space
    - "Continuous" random effects
- Christmas-themed exercise!

# Mixed effect model example

Let's go back to our earlier example:

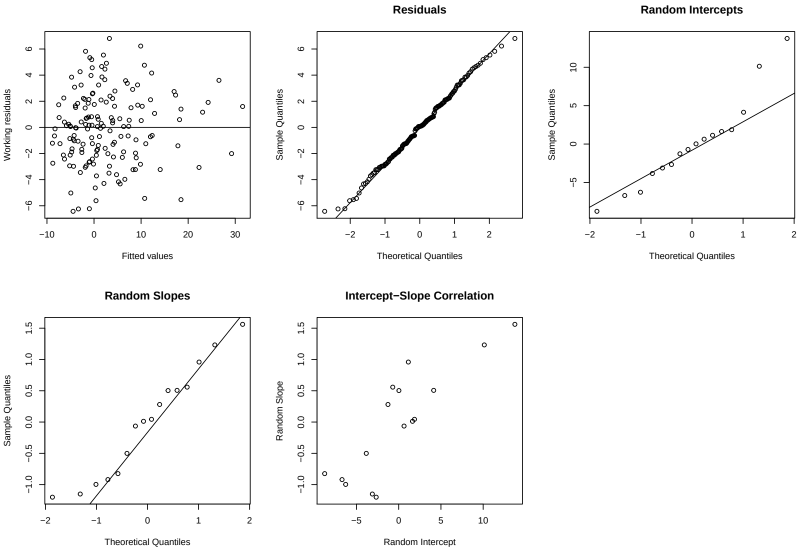
```
lmer(y ~ x + (x|site), data = dat)
```

- We're interested in predicting $y$ using $x$ (fixed effects)
- Data was collected at a number of *sites*, which may affect $y$
- Effect of each site is normally distributed (random intercept)
- Effect of site on slope of $x$ is normally distributed (random slope)



# Validation

- Similar to linear models, but we *also* check whether the random intercepts are normally distributed



# Hypothesis testing

Is this fixed effect important? (e.g. ANOVA)

- Use likelihood-based test via `drop1` (likelihood ratio test, AIC)
- Be careful to fit model with `REML = FALSE`!

```
lmm1 <- update(lmm1,REML=FALSE) #Refit model using ML rather than REML
drop1(lmm1,test='Chisq') #x has a strong effect
```

```
## Single term deletions
##
## Model:
## y ~ x + (x | site)
##        npar    AIC    LRT  Pr(Chi)
## <none>      913.08
## x         1 917.85 6.7697 0.009272 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

# Hypothesis testing (cont.)

How do I know this effect is different from $x$?

- Use Wald Z-test (2-sided p-value from Z-test)

```
lmm1 <- update(lmm1,REML=TRUE) #Reset to REML
meanEst <- fixef(lmm1)[2] #Get mean
seEst <- sqrt(vcov(lmm1)[2,2]) #Get standard error
(1-pnorm(meanEst/seEst,0,1))*2 #p-value from 2-sided Z-test
```

```
##           x
## 0.004659069
```

- `glht` from `library(multcomp)` works with `lmer` models if you are comparing between coefficients (e.g. "Is treatment A different from B and C?")

# What if my response variable is non-normal?

- Linear model (LM)

$$\hat{y} = X\beta$$
$$y \sim Normal(\hat{y}, \sigma)$$

- Generalized linear model (GLM)

$$logit(\hat{\phi}) = X\beta$$
$$y \sim Binomial(\hat{\phi})$$

- Linear mixed effects model (LMM)

$$\hat{y} = X\beta + U\zeta$$
$$y \sim Normal(\hat{y}, \sigma)$$
$$\zeta \sim Normal(0, \sigma_{site})$$

- Generalized linear mixed effects model (GLMM)

$$logit(\hat{\phi}) = X\beta + U\zeta$$
$$y \sim Binomial(\hat{\phi})$$
$$\zeta \sim Normal(0, \sigma_{site})$$

# How do I fit GLMMs?

- `glmer` and `glmer.nb` from `library(lme4)` work for Binomial, Poisson, and Negative Binomial data

```
library(lme4)
glmm1 <- glmer.nb(y2~x+(x|site),data=dat) #Negative binomial GLMM
summary(glmm1) #glmer.nb takes a LONG time to run
```

- `glmmTMB` from `library(glmmTMB)` works for those above, *plus* a bunch of others
  - e.g. Zero-inflation, Beta-binomial, Spatial Models

```
library(glmmTMB)
glmm2 <- glmmTMB(y2~x+(x|site),data=dat,family=nbinom2())
summary(glmm2) #Similar results, but quicker
```

# Fitting GLMMs - glmer.nb

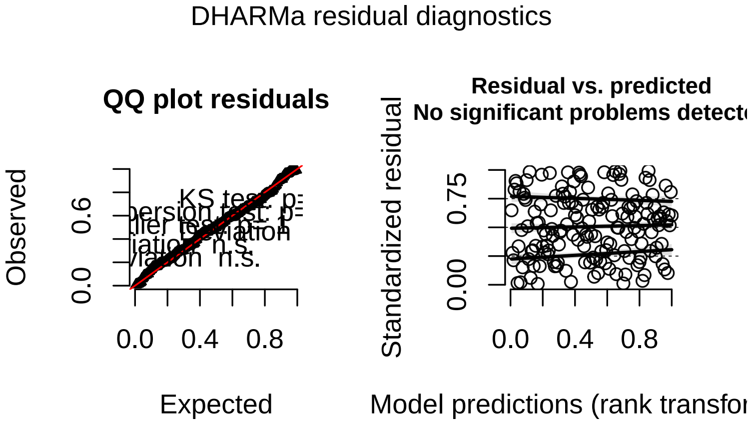
```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: Negative Binomial(5.1295)  ( log )
## Formula: y2 ~ x + (x | site)
##    Data: dat
##
##      AIC      BIC   logLik deviance df.resid
##    627.8    646.3   -307.9    615.8      154
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.3745 -0.7098 -0.3946  0.5108  2.5367
##
## Random effects:
##  Groups Name        Variance Std.Dev. Corr
##  site   (Intercept) 1.43502  1.1979
##         x           0.02878  0.1697   0.92
## Number of obs: 160, groups:  site, 16
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.32745    0.32166   1.018   0.3087
## x            0.10830    0.04681   2.314   0.0207 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##   (Intr)
## x 0.799
```

# Fitting GLMMs - glmmTMB

```
##  Family: nbinom2  ( log )
## Formula:          y2 ~ x + (x | site)
## Data: dat
##
##      AIC      BIC   logLik deviance df.resid
##    627.8    646.2   -307.9    615.8      154
##
## Random effects:
##
## Conditional model:
##  Groups Name        Variance Std.Dev. Corr
##  site   (Intercept) 1.43543  1.1981
##         x           0.02892  0.1701   0.92
## Number of obs: 160, groups:  site, 16
##
## Overdispersion parameter for nbinom2 family (): 5.12
##
## Conditional model:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.34132    0.32172   1.061   0.2887
## x            0.11026    0.04697   2.348   0.0189 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

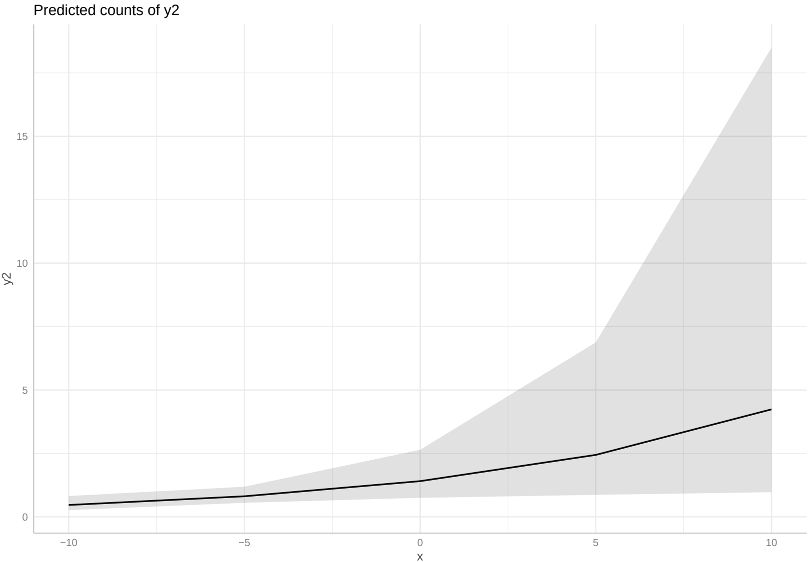
# Residual checks on `glmmTMB` models

- Extract residuals and make your own plots, or use `simulateResiduals` from `library(DHARMa)` (see here)
- `DHARMa` also has useful functions for checking overdispersion and zero-inflation (found here)



# Partial residual plots for `glmmTMB` models

- `ggpredict()` from `library(ggeffects)` works with `glmmTMB` models



# Spatial and Temporal Random Effects

"*My data were sampled over time or over a geographic area (or both). Can I just use day or site as a random effect?*"

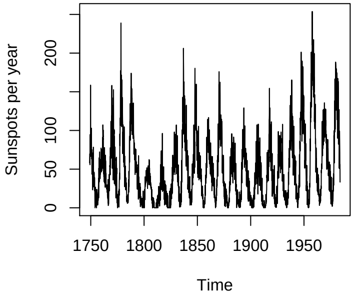
- Short answer: "Yes"
- Long answer: You might be able to do better, because of the **1st Law of Geography**:

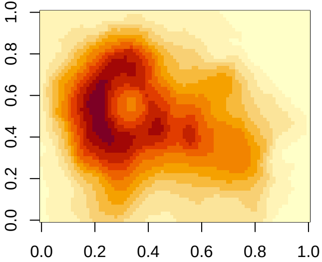
"*...everything is related to everything else, but near things are more related than distant things.*" Waldo Tobler

- If you have spatial or temporal information, this can help R to estimate random effects more accurately
  - Can improve prediction accuracy (smaller p-values)
  - Can give you hints about the underlying causal mechanisms

# Temporal or Spatial Data

- Correlation is often present in temporal data or spatial data; causes may be unknown or "uninteresting"
- Usually we are interested in accounting for these patterns, in order to better estimate the "interesting" patterns on top of them



# Covariance

- Normal distributions[1] don't just have a single $\sigma$, but a matrix of values
- If our data *y* are *independent*, then it looks like this:

$$y \sim Normal(M, \Sigma)$$

$$M = [\mu_1, \mu_2, \mu_3]$$

$$\Sigma = \begin{bmatrix} \sigma^2 & 0 & 0 \\ 0 & \sigma^2 & 0 \\ 0 & 0 & \sigma^2 \end{bmatrix}$$

- Zeros mean "$\mu_1$, $\mu_2$, & $\mu_3$ aren't related to each other"
- Diagonal elements = *variance*, off-diagonal = *covariance*

[1] Multivariate Normal

# Covariance and Correlation

In real life, things may not be independent from each other. For example:

- $\sigma = 2$ (variance $= \sigma^2 = 4$)
- $\mu_1$ and $\mu_2$ are strongly correlated (r=0.7), but $\mu_3$ is not related to anything (r=0). Shown here as a *correlation matrix* ($R$):

$$R = \begin{bmatrix} 1 & 0.7 & 0 \\ 0.7 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- When multiplied by the variance, this becomes the *covariance matrix* ($\Sigma$)

$$\Sigma = \begin{bmatrix} \sigma^2 \times 1 & \sigma^2 \times 0.7 & \sigma^2 \times 0 \\ \sigma^2 \times 0.7 & \sigma^2 \times 1 & \sigma^2 \times 0 \\ \sigma^2 \times 0 & \sigma^2 \times 0 & \sigma^2 \times 1 \end{bmatrix} = \begin{bmatrix} 4 & 2.8 & 0 \\ 2.8 & 4 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

# Gaussian Process Modelling

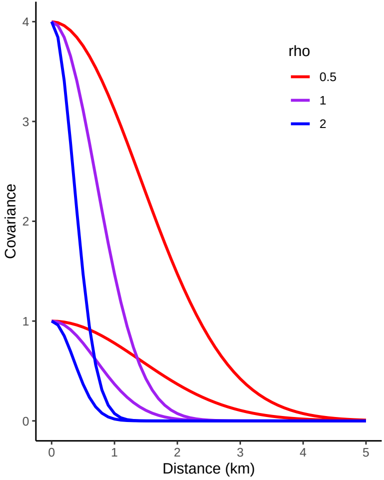
- We can model covariance between things as a function of *distance*, either in time or space
- Squared-exponential is fairly common[2]:

$$\Sigma = covariance$$

$$\Sigma = variance \times correlation$$

$$\Sigma = \sigma^2 \times e^{-\rho^2 Dist^2}$$

- Instead of finding a single $\sigma$ value, R now looks for $\sigma$ (maximum covariance) and $\rho$ (decay with distance)




[2] Also common: AR-1 (temporal processes), Matérn (spatial processes)

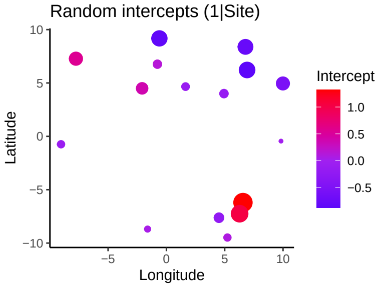
# Spatial random effects

- Say that we collected data at 16 sites, and we're interested in the effect of $y$ on $x$
- Let's first fit a model with a random intercept for site

```
lmm2 <- glmmTMB(y~x+(1|site),data=dat2)
```

- If we plot the intercepts for each site, we see that they are clustered:

```
dat2 %>% select(site,lat,lon) %>% distinct() %>% arran
  ggplot(aes(lon,lat,col=int))+geom_point(aes(size=abs
  labs(title='Random intercepts (1|Site)',x='Longitude
  scale_colour_gradient2(low='blue',mid='purple',high=
  guides(size=FALSE)
```



# Spatial random effects (cont.)

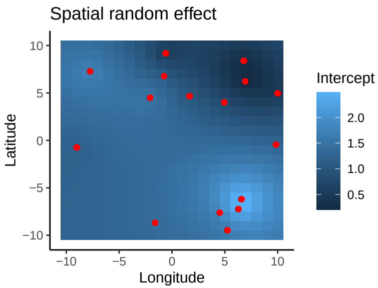
- Re-fit model with a spatial (exponential) random effect

```
dat2$coords <- numFactor(dat2$lon,dat2$lat) #Coordina
dat2$group <- factor(rep(1,nrow(dat2))) #Group factor
lmm3 <- glmmTMB(y~x+exp(coords+0|group),data=dat2)
```

- Clustering effect modeled as a spatial random effect

```
#Plot spatial random effect
spRanEff <- expand.grid(lon=-10:10,lat=-10:10) %>%
  mutate(coords=numFactor(lon,lat),group=factor(rep(1,
  mutate(pred=predict(object=lmm3,newdata=.,type='resp

ggplot(spRanEff,aes(lon,lat,fill=pred))+geom_raster()+
  geom_point(data=dat2,aes(lon,lat,fill=NULL),col='red
  labs(x='Longitude',y='Latitude',title='Spatial rando
```



# A challenger approaches

- Ho ho ho! Merry Christmas! In order to maximize the number of presents that you get from Santa Claus, you've decided to apply an analytic approach, and have collected data across Alberta on *number of Christmas presents received*
- You've also collected data across a wide geographic area on things that might influence Saint Nick's generosity (*naughtiness*, *presence of milk and cookies*, *chimney width*)
- Fit a GLMM to the present data, one using spatial random intercepts, and one using "regular" random intercepts