2550 Problem Set 3

Alyson Singleton (collaborated with Katherine Webb)

10/18/2019

Question 1

Given background

In the first homework, you examined the relationship between pain and time for 200 individuals in the weather dataset. You also examined the correlation between pain and temperature. You then looked at the relationships between the intercepts and slopes from the regression and factors that varied by patient such as age and sex. This forms the basis of a multilevel model in which pain varies within patient according to time and temperature and between patients these relationships vary with patient-level factors. In this exercise, you will put these together and fit a multilevel model in which within patient, pain is a function of time and temperature and between patients these relationships may depend on age, race, income, treatment, sex, occupation, working status and use of NSAIDs. Use the lme or lmer function to fit a multilevel model in which

Yit = ai + biXit + eit

ai = g0 + g1zi + ui bi = h0 + h1zi + wi

where Yit = pain at time t for individual i, Xit = time t or temperature at time t for individual i and zi= patient level factor (age, sex, race, income, occupation, working status, use of NSAIDs).

Problem Statement

Build a multivariable regression model treating ai and bi as random effects. Make sure to give a thorough writeup of your results with appropriate graphs and tables. Code should be put at the end of the assignment.

The syntax for lmer is lmer(y~1+x+(1+x|Cluster), data = ...) where y is the response variable, x are the fixed effects (predictors) and (1+x|Cluster) describes the random effects nested within cluster.

Data Exploration

We look to investigate the relationship of pain with time and temperature within patients and how between patients these relationships may depend on age, race, income, treatment, sex, occupation, working status and use of NSAIDs.

Data Formatting

I begin my analysis by cleaning the data and reformatting it so I may use it in a multilevel model through the lmer function. I began by making a "mother" data set where each patient was represented by one row that included all of their time and temperature information (a reduced wide format). Next, I changed this into a long format using the melt function on time, temperature, and pain.

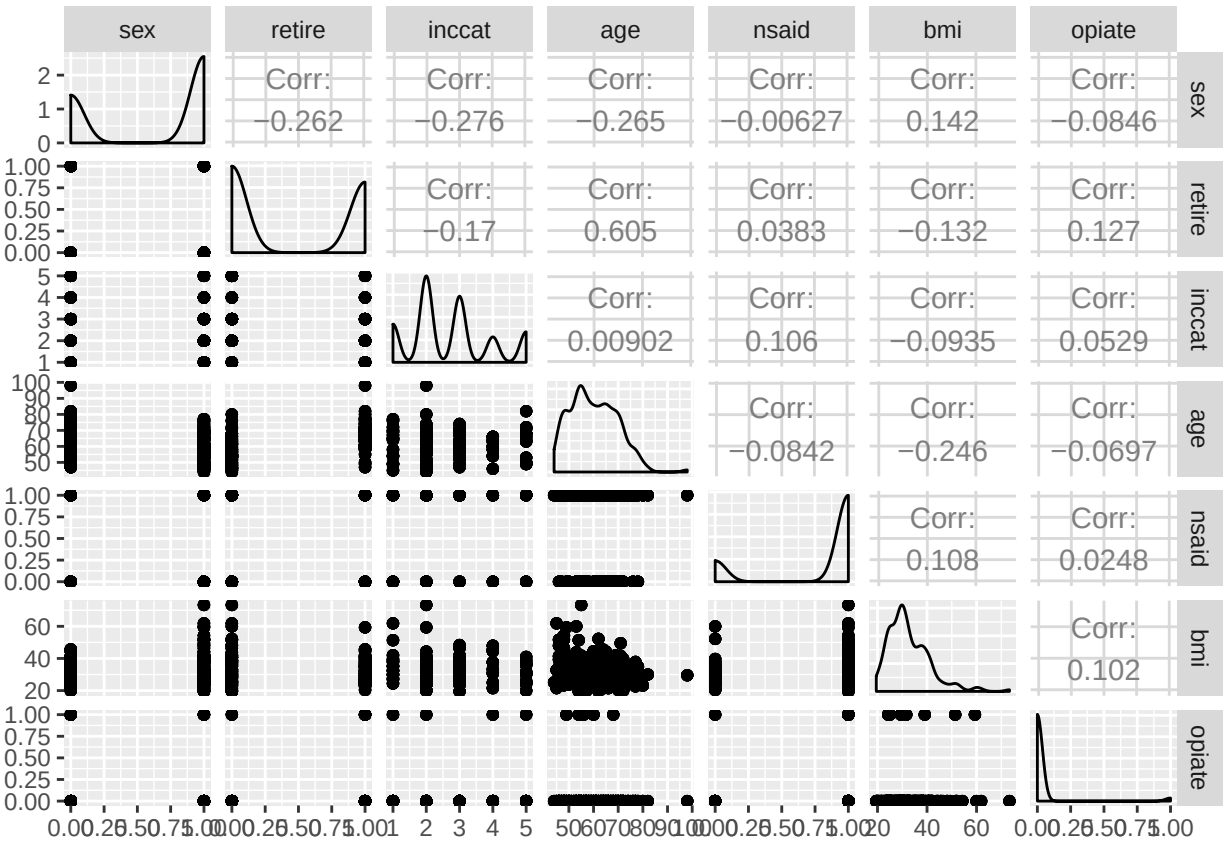
Demographic Variables

I also removed variables not mentioned in the problem statement as we were directed to focus our efforts specifically on the referenced demographic variables. Then, using and building on the investigation of these variables in Problem Set #1, I further editted our list of demographic variables to investigate.

1. Race: Nearly 89% of the patients are white. Therefore, it is almost a constant variable throughout the patients, and we ignore it from this investigation as we did in Problem Set #1.
2. Income: Income has too many categories and weakens the regression result. Therefore, we replace it with inccat as we did in Problem Set #1. Note inccat also has a reasonable number of observations in each category (1 - 105, 2 - 238, 3 - 182, 4 - 70, 5 - 84).
3. Sex: Restructure so 0 - male (511 observations), 1 - female (924 observations).
4. Occupation: I believe spending the time recoding the many categories into a usable data format is not worth the marignal information it would provide in addition to the retirement information–we remove occupation and use retire instead.
5. Retire: Restructure so 1 - retired (497 observations), 0 - not retired (609 observations).
6. Age: Subject age (widespread number of observations from 44 to 98).
7. NSAID: NSAID use, 1 - yes (1155 observations), 0 - no (280 observations). Although it is not very even, there is a reasonable number of observations in each so we keep it.

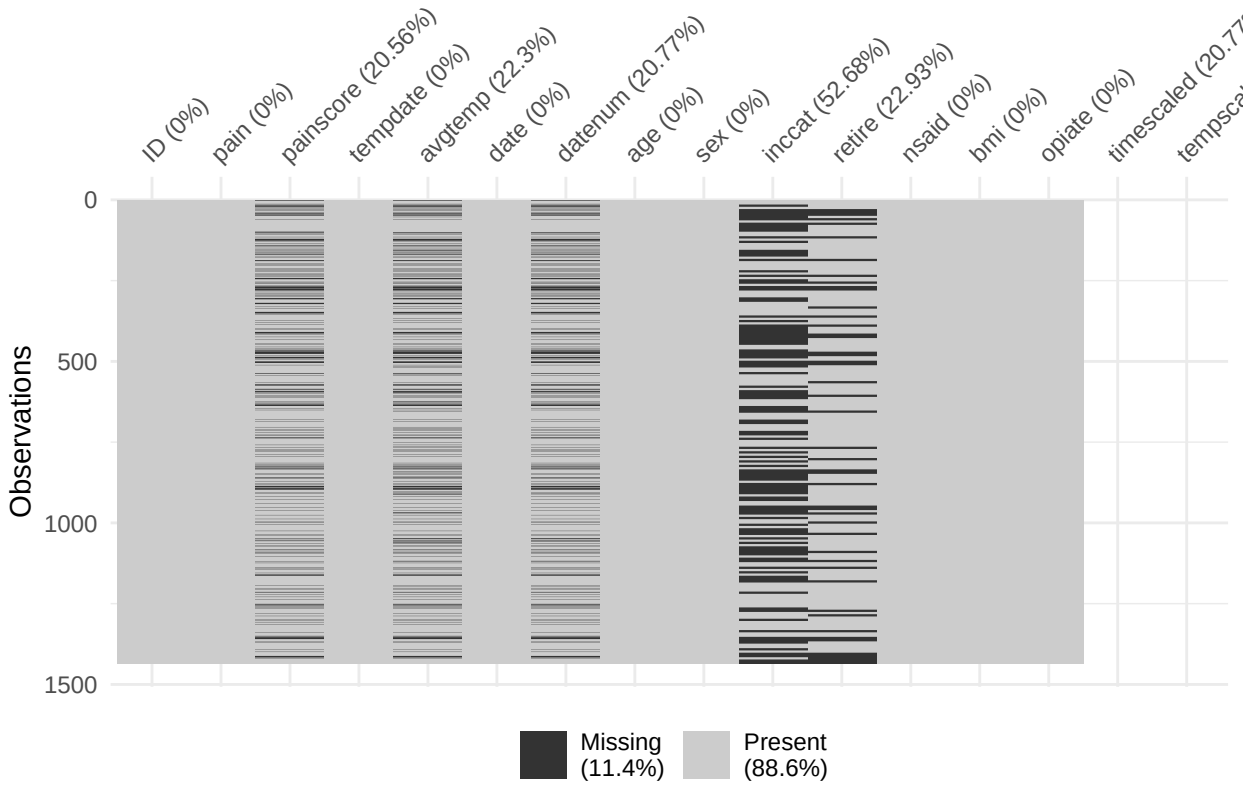
Additional variables added / changed:

8. I opted to include BMI (body mass index) and opiate (1 - uses opiate, 49 observations, ; 0 - no opiate use, 1386 observations) as addition variables given their effect in Problem Set #1. Opiate is borderline with the number of samples with opiate use, so our result will be conditioned on the fact there are few observations of opiate use.
9. I standardized the time columns of each person so that each patient entered the study effectively on day zero. I did this by subtracting their "lastdt1" value from all time values. This way, each patient is entering the study at their personal "time zero" and their time variables are comparable.
10. I also added two new columns : timescaled and tempscaled. I used the scale function to scale time and temperature, respectively, to help the models' converge.
11. Lastly, note from the ggpairs plot below that there appears to be no substantial correlation between any of the demographic variables.

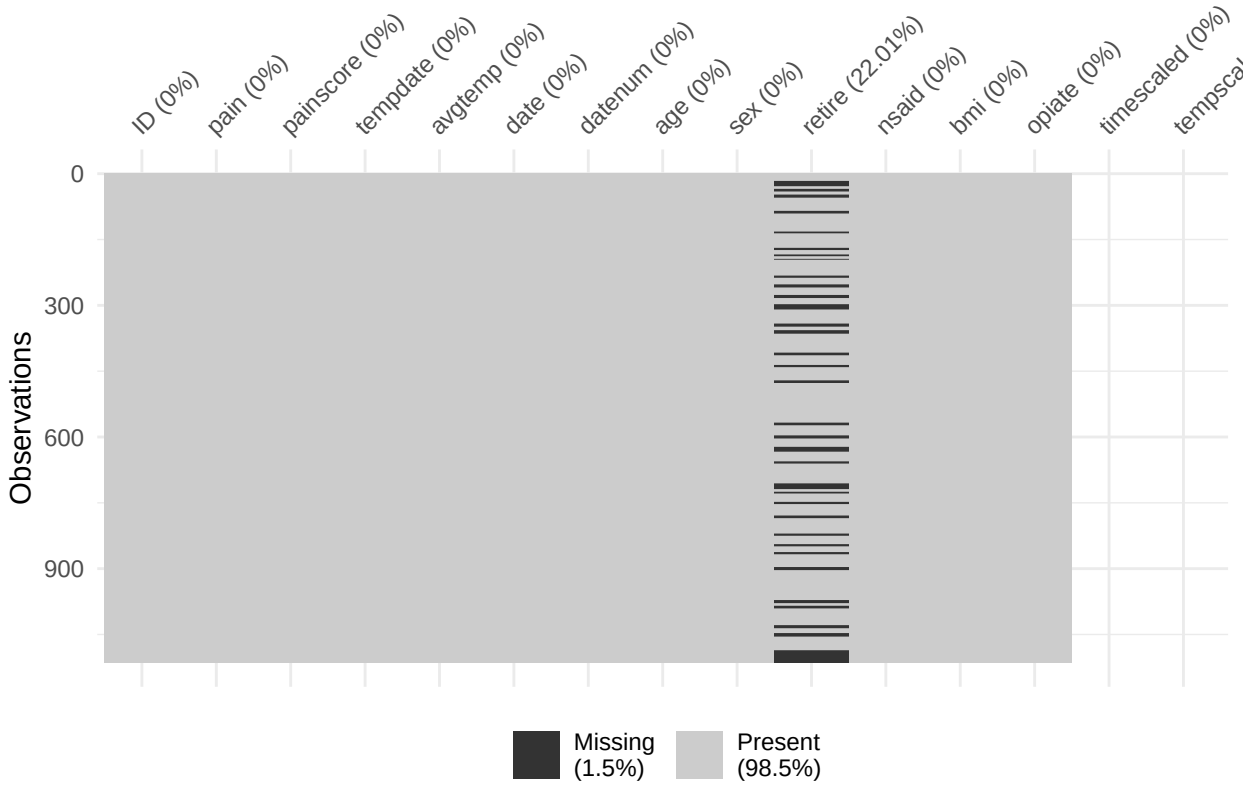


Missing Values

Next, I began to look at missing values. See below for a visual representation of missing values in my newly constructed data set.



We note that over 50% of the inccat values are missing, therefore I decide to remove this variable from the data set. We decide to leave retire as is since it is only at 21.97%. We also note that ~20% of pain values, temperature values, and date values are missing. As these are key variables in the analysis, I decided to remove the observations where any of these are missing they will not contribute to building a useful model. Note that this will only remove the one row representing a pain measurement visit, not an entire patient (i.e. if one person misses coming in for a third visit, only that third visit will be removed). I recognize that there are other ways to deal with missing data that help you prevent bias and allow you to use the other data included in the row with the missing value. I, however, do not know how to employ these techniques and I opted for explicitly naming their removal and conditioning my results upon this decision. See below for the final missing values in our final data set.



Model Construction and Testing

Now I begin testing models and making edits / removals / additions one by one, testing their performance as I go. I used ANOVA (AIC, BIC, logLik) and diagnostic plots as model tests. I begin with just the random intercept model of outcome of painscore and cluster of ID number. Here are my decisions and my rationale :

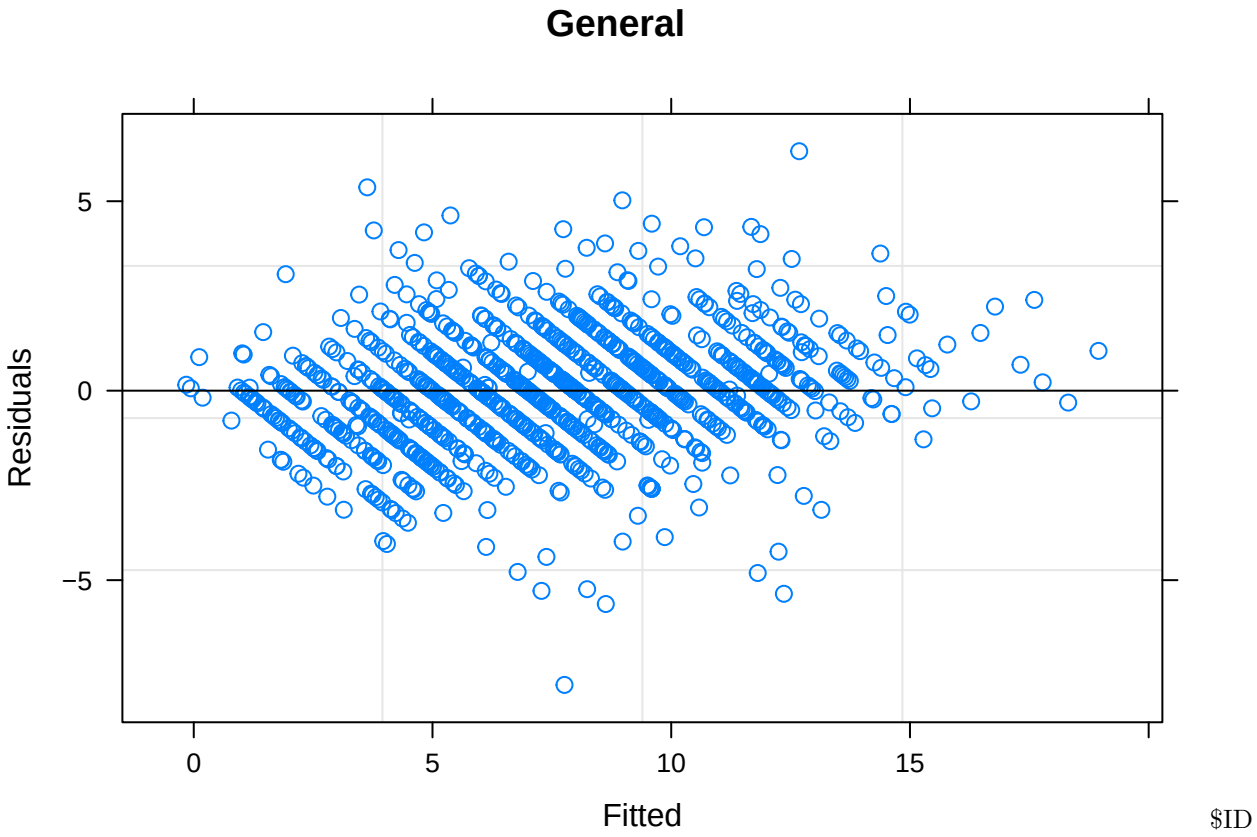
(1) Next we look at whether to include time, temperature, or both. Note that I use time scaled and temperature scaled because otherwise the model does not converge. When I add time and temperature each seperately as random effects, both result in a significantly different model. However, if I add temperature to the time model there is no significant difference where as there is if I add time to temperature. This leads me to believe that time is worthwhile but we can exclude temperature for simplicity's sake. See table below for confirmation with AIC and BIC levels.

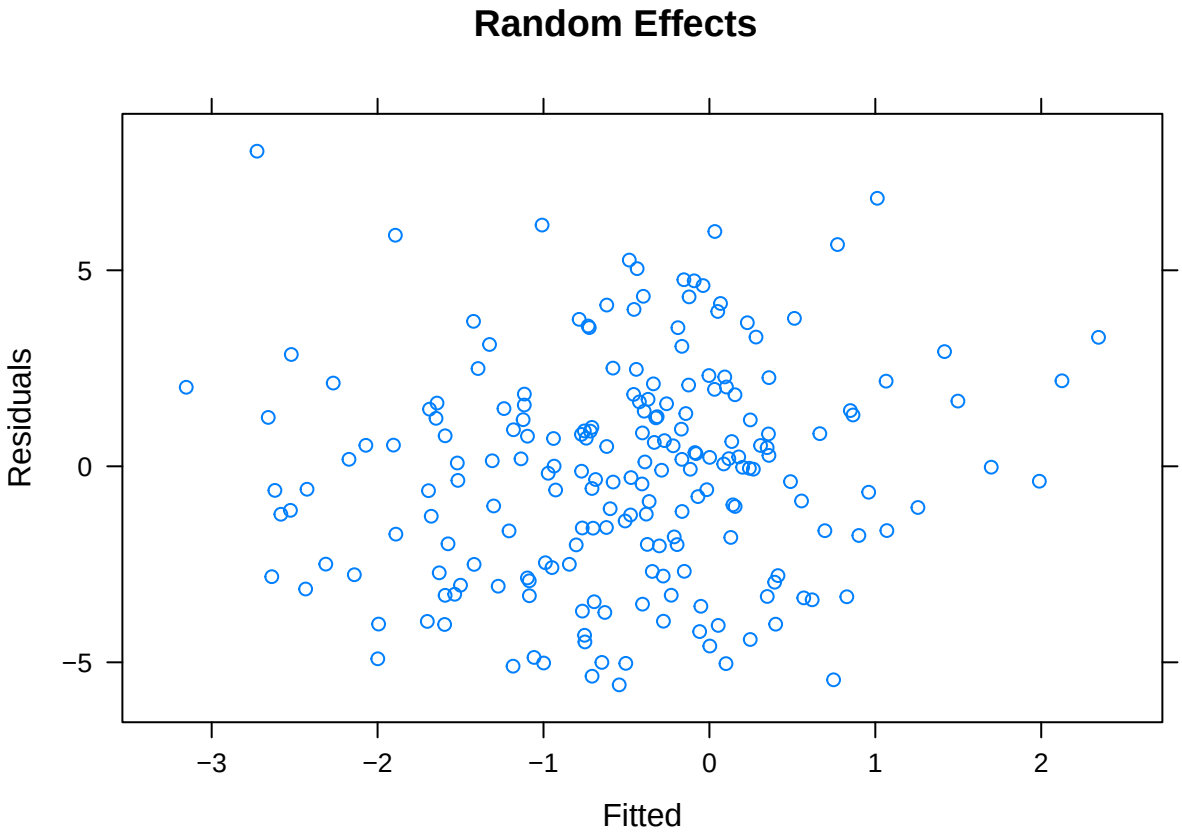
	AIC	BIC
Random Intercept Model	5450.5	5465.5
Time Model	5264.3	5289.4
Temp Model	5427.0	5452.1
Time and Temp Model	5270.0	5310.1

(2) Now we look at the signifcance of demographic variables as random effects. The only two variables that change the model signifcantly (according to ANOVA) are BMI and opioid use. You can see the AIC, BIC and the p-value of the comparison to the time model constructed in part (1). Lastly, you see additionally that having both BMI and opioid use is a change in AIC and BIC from the models with only one of them (p-values not listed for these comparisons but both are signifacant, adding bmi to time and opiate and adding opiate to time and bmi). The time + bmi + opiate model also remains significantly different than the time only model.

	AIC	BIC	P-Val Compared to Time Model
Time Model	5264.3	5289.4	NA
Time + Age Model	5264.40	5294.50	0.17
Time + Sex Model	5264.10	5294.20	0.14
Time + Retire Model (Retire NA's Removed)	4117.60	4146.20	0.39
Time + NSAID	5266.20	5296.30	0.82
Time + BMI Model	5253.8	5283.9	0.0004
Time + Opiate Model	5255.2	5285.3	0.0008
Time + BMI + Opiate	5246.3	5281.4	1.691e-05

(3) Lastly, we plot the fitted values against the residuals to confirm there is no pattern in the residuals both for the overall model and for the random effects specifically. We see below that there is no pattern in either and this encourages our use of this model.





Final Model

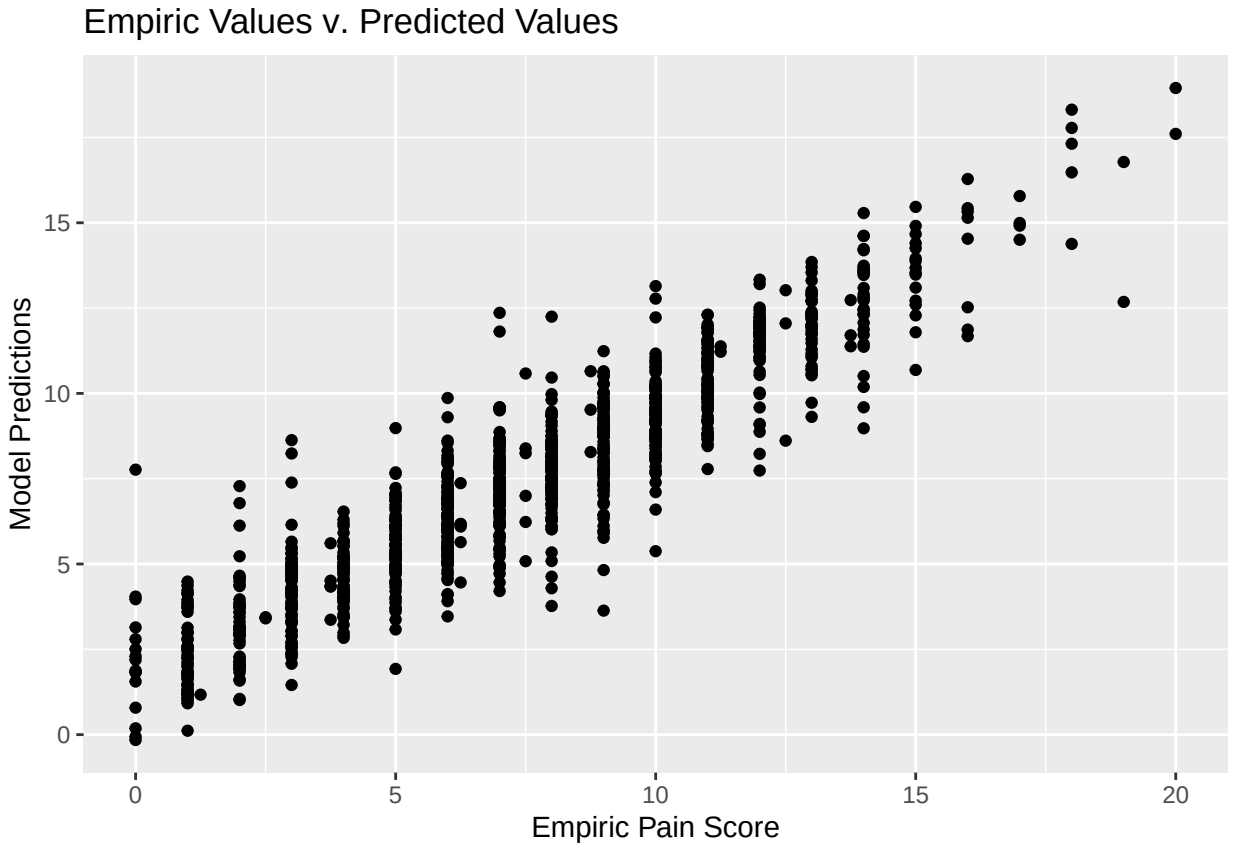
Therefore, my final model proposal is as follows :

painscore ~ 1 + bmi + opiate + (1 + timescaled | ID)

Model Output:

	Variance	Std. Dev.
Random Intercept	8.416	2.901
Random Time Scaled	1.721	1.312

	Estimate	Std. Error	T-Value
Fixed Intercept	4.82645	0.84031	5.744
Fixed BMI	0.08402	0.02506	3.353
Fixed Opiate	3.61013	1.15240	3.133



Interpretation of Coefficients

First, we conclude that a patient's BMI and opioid user status is a predictive factor of their pain. I would interpret the results of this model by saying that the average pain score when all other variables are zero is 4.83. Additionally, 0.084 is the average change in pain level for an increase in BMI by one. Now to interpret the random effects. We say that the variance of time, 1.721, is the variance of the distribution of the slopes within the patients depending on their own personal timing. The variance of the intercept, 8.416, is the variance of the distribution of intercept values between patients.

Question 2

Given Background

Use the text files srrs2.txt and cty.txt found in the Datasets folder to analyze the radon data in order to determine factors that affect the amount of radon measured in houses randomly sampled in various counties in Minnesota as a function of the floor on which the measurement was taken, the uranium level of the county, whether or not the house has a basement and whether or not the home is single family. Constuct a 2 level model for 1) houses within counties with predictors that correspond to variables measured on houses and 2) counties with the county level predictor uranium. Use a logarithmic transformation of both radon and uranium. The file radon text files.rtf contains a codebook but key variables are described below also.

- idnum: House number
- state2: State (don't use state) Arizona (AZ), Indiana (IN), Massachusetts (MA), Michigan (MI), Minnesota (MN), Missouri (MO), North Dakota (ND), Pennsylvania (PA) and Wisconsin (WI)

- stfips: state number (used in setting up variables)
- typebldg: = 1 if single family; otherwise another type of home (lump all others together
- floor: floor of house on which radon measurement taken (0 = basement, 1 = 1st, etc.; 9 indicates missing)
- basement: Y if house has basement; N if house does not have basement; else unknown
- activity: radon level
- county: name of county
- cntyfips: county number (used in setting up variables)

Cty.txt includes data on county uranium levels. You will need the following variables:

- stfips: state number code
- ctfips: county number code
- st: state
- cty: county
- uppm: average uranium level in county

Note that there are some additional variables you can ignore and that some of these variables might have a few additional categories that you will have to determine what to do with. For instance, there are a few homes where readings were not taken on the basement or first floors but on the second or third. You could choose to treat these as non-basement (grouping with first floor for example).

You will first need to create a dataset that has the following variables: House, State(MN only), County, Single family house (1/0), Basement (1/0), Floor of house. Activity. Uranium level. Note that not all counties and states in the cty.txt data are in the srrs2.txt dataset.

Problem Statement

Once the dataset is created (for hints on how to do this check out the Gelman/Hill R file 12.2 saved in the Rscripts folder) Explain which factors are related to the radon levels and construct some useful tables and figures to explain your model. Constuct a 2 level model for 1) houses within counties with predictors that correspond to variables measured on houses and 2) counties with the county level predictor uranium. Use a logarithmic transformation of both radon and uranium.

Data Exploration

We aim to analyze the radon data in order to determine factors that affect the amount of radon measured in houses randomly sampled in various counties in Minnesota as a function of the floor on which the measurement was taken, the uranium level of the county, whether or not the house has a basement and whether or not the home is single family. We begin by investigating our data set.

Data Formatting

As I did in Question #1, I begin my analysis by cleaning the data and reformatting it so I may use it in a multilevel model through the lmer function. I load the datasets and add the variables that I am directed to. I join the two subsetted data sets to create the long data set that we are looking for. Each line is a house within our counties, our groups.

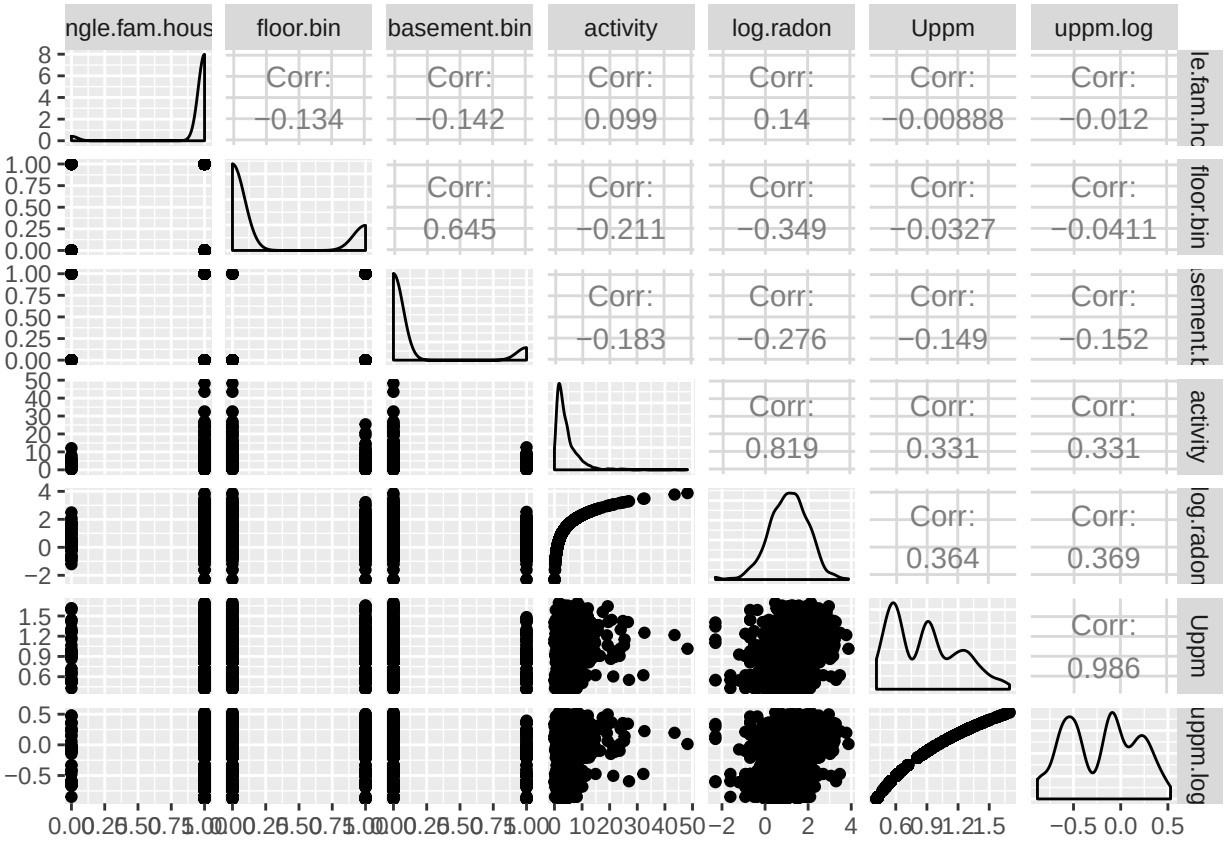
Demographic Variables

Now we will investigate the categories and number of observations per category in each of our house level variables.

1. single.fam.house: Restructed so 1 if single family; 0 if otherwise another type of home (lumped all others together from typebldg). 1103 single family observations, 57 other observations. Not ideal only 57 observations in other category, make note.
2. basement.bin: Restructed so 0 if house has basement (1000 observations); 1 if house does not have basement (145 observations).
3. floor.bin: Floor of house on which radon measurement taken. Restructed so 0 = basement (920 observations), 1 = first floor (267 observations).
4. activity: Radon level of the house.

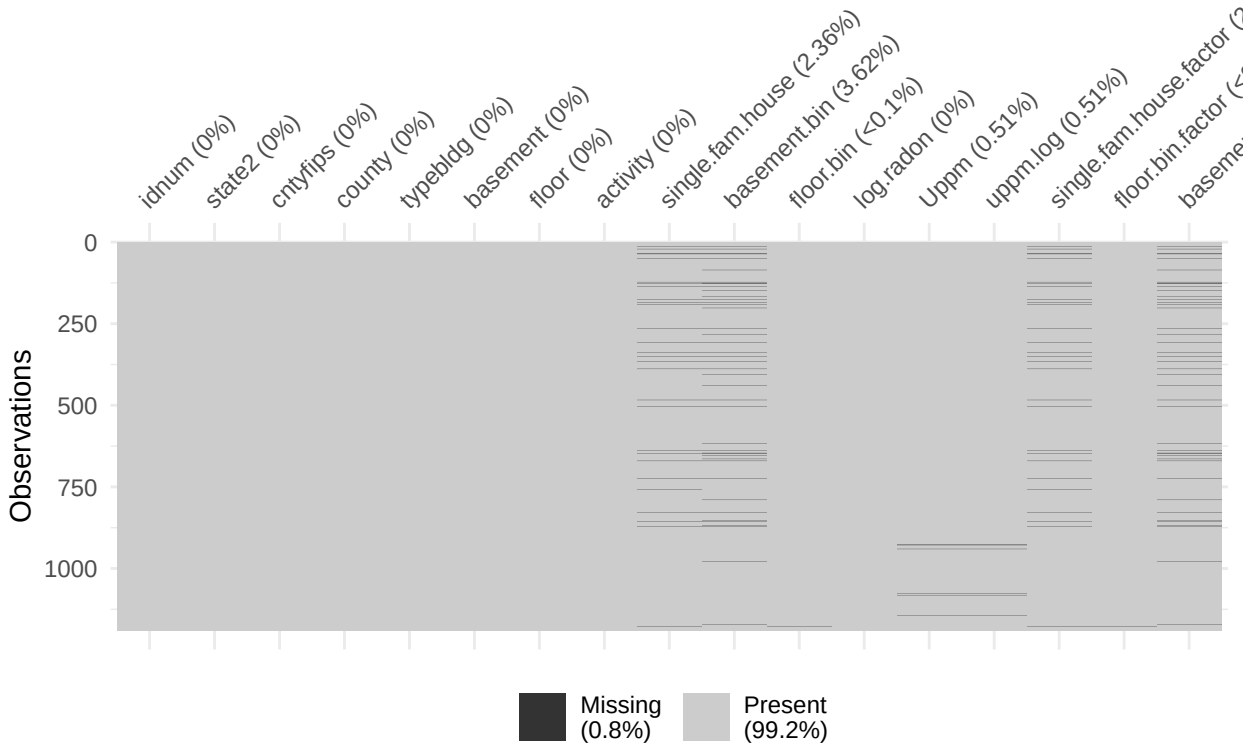
Additional variables added / changed:

5. I subset so we are only looking at the state of Minnesota as directed (used the state2 variable).
6. Built log(Uranium) variable and log(radon) variable as directed. To avoid infinite values I change any radon values of 0 to 0.1 prior to the transformation. The pairs plots below of their intitial and then transformed distributions demonstrates helpfulness of these two transformations.
7. Notice the correlation of 0.645 between basement.bin and floor.bin. This indicates to me that we should remove one. Noted for model selection process.

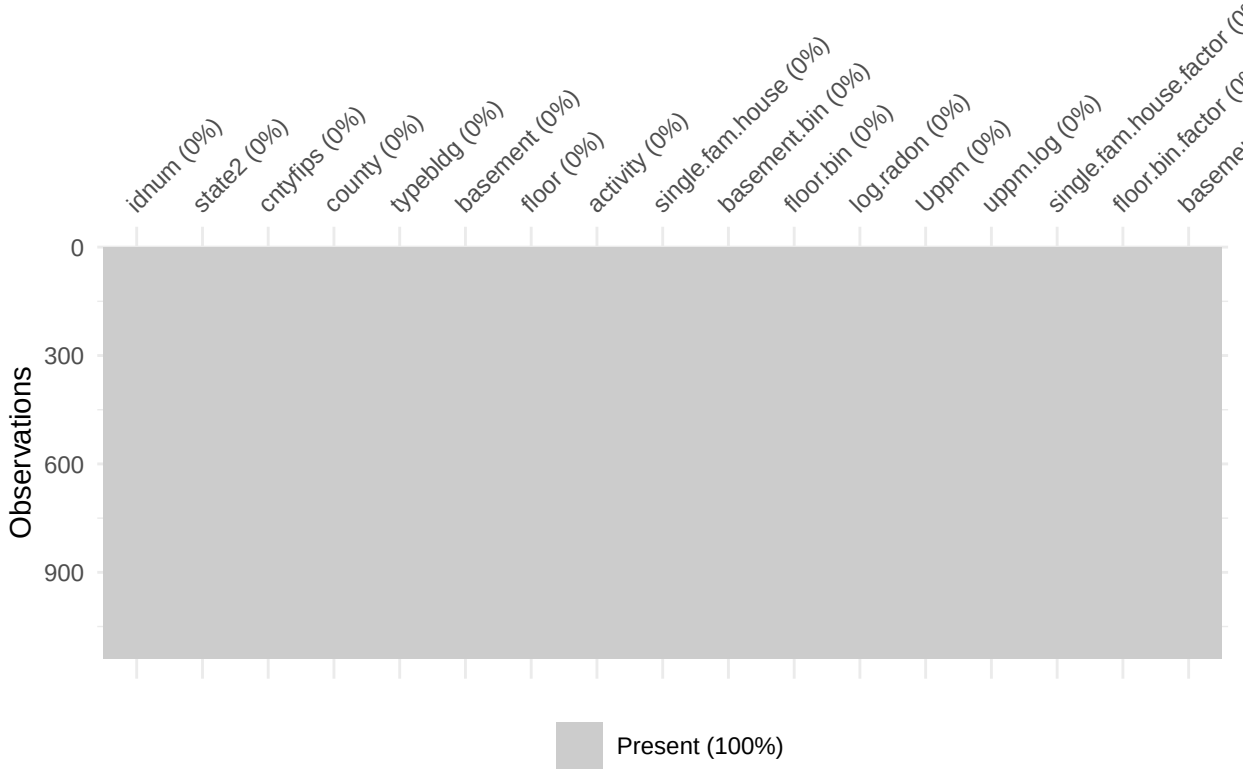


Missing Values

Next we must identify where, if any, missing values lie and how to handle them. See below for initial missing values:



As there are very few missing values (<3% of single.fam.house and <4% of basement.bin), we opt to remove these rows for simplicity's sake. We make note that our output will be conditioned on this decision.



Model Construction and Testing HEART

Now I begin testing models and making edits / removals / additions one by one, testing their performance as I go. I used ANOVA (AIC, BIC, logLik) and diagnostic plots as model tests. I begin with just the random intercept model of outcome of log.radon and cluster of county. Here are my decisions and my rationale :

(1) Next I looked at whether I should add a random effect of log transformed uranium. This addition significantly reduces the AIC and BIC so I decide to keep it. See table bleow for values.

(2) Next I investigated which random effects to include. The only two variables that change the model signifcantly (according to ANOVA) are BMI and opioid use. You can see the AIC and BIC of my progression of attempted models below. I note that adding Basement does not add much more on top of adding Floor, and actually seems to be causing AIC and BIC to rise (likely beacuse it's added specificity is not worth the added model complexity). This makes sense since I noted the high correlation between Floor and Basement. I now test which has a larger impact: Floor or Basement, since we only need one. I see that Floor lowers AIC and BIC more significantly. Single.Fam does seem to make a significant difference. However, we note that while it lowers AIC it does raise BIC. I decide to keep it, but this decision could go either way. Therefore, I conclude that we should include Floor and Single.Fam in our model.

	AIC	BIC
Random Intercept Model	2823.8	2838.9
log(Ur) Model	2777.9	2798.0
log(Ur) + Floor Model	2690.0	2720.2
log(Ur) + Basement Model	2757.7	2787.9
log(Ur) + Floor + Basement Model	2693.1	2738.4
log(Ur) + Floor + Single.Fam Model	2685.8	2731.1
log(Ur) + Floor + Basement + Single.Fam Model	2690.1	2755.5

(3) Lastly, we plot the fitted values against the residuals to confirm there is no pattern in the residuals for the overall model. We see below that there is no pattern and this encourages our use of this model. I tried to do this plot for the random effects specifically and had a weird output that I did not know how to interpret correctly so I left it out.

Final Model

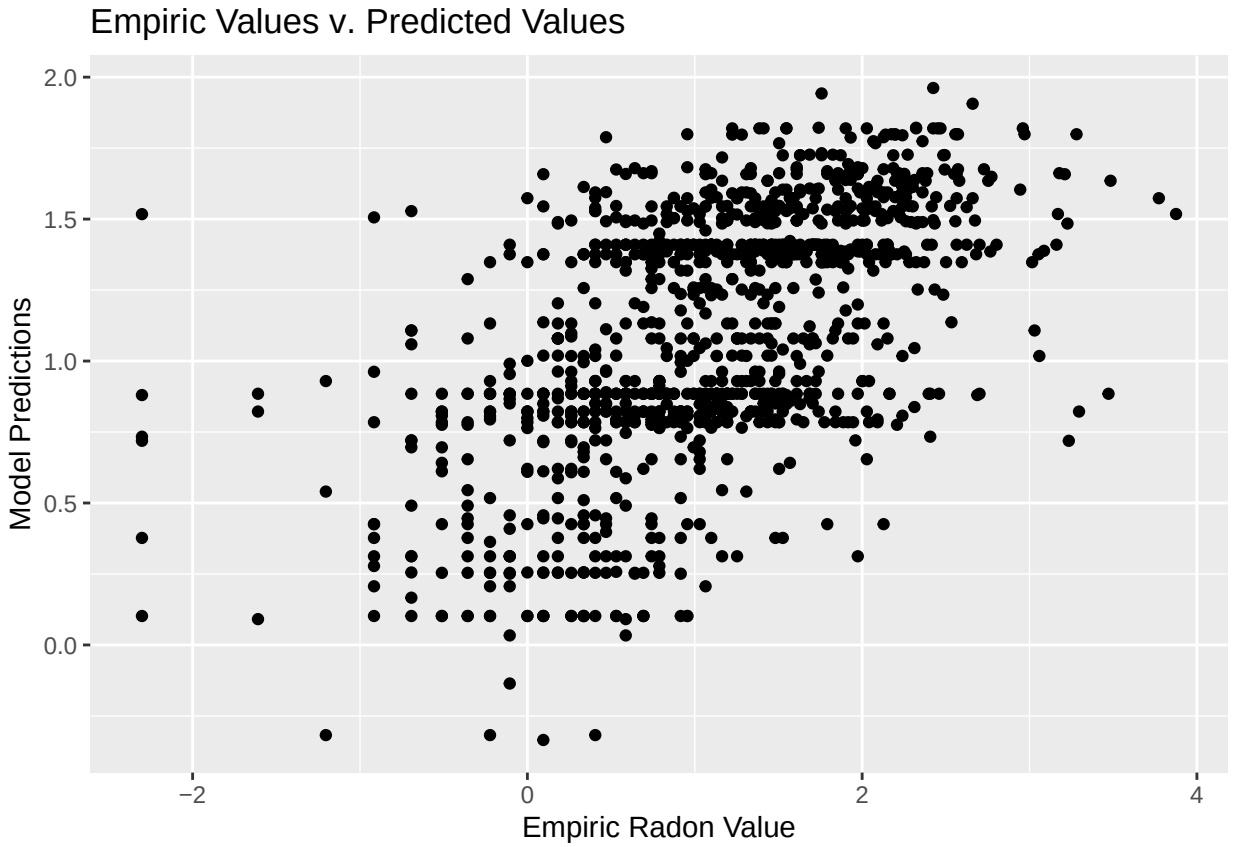
Therefore, my final model proposal is as follows :

log.radon ~ 1 + uppm.log + (1 + floor.bin + single.fam.house | cntyfips)

Model Output:

	Variance	Std. Dev.
Random Intercept	0.08669	0.2944
Random Floor	0.48165	0.6940
Random Single.Fam	0.12969	0.3601

	Estimate	Std. Error	T-Value
Fixed Intercept	1.37182	0.03158	43.43
Fixed log(Uppm)	0.83730	0.07882	10.62



Interpretation of Coefficients

First, we conclude that the county uranium levels in which a house resides is a predictive factor in that house's radon levels. I would interpret the results of this model by saying that the average radon level when all house level variables are zero is eˆ1.37 or 3.94. Additionally, 0.837 is the average change in radon level for an increase in county level uranium by one. We must only exponentiate the intercept and not the coefficient, as the uranium coefficient was also log transformed so exponentiation keeps the relationship the same. In terms of interpretting the random effects. We say that the variance of whether it was a single family home, for example, (0.12969) is the variance of the distribution of the slopes within the counties depending on this distinction. The other random effects variables follow in similar fashion (the variance of the intercept is the variance of the distribution of intercept values between counties).

Code Appendix

```
knitr::opts_chunk$set(echo = TRUE)

## Organize and Clean Data / Build Mother Dataset

setwd("~/Desktop/masters/PHP2550/")
library(dplyr)
library(tidyr)
library(reshape2)
```

```r
#read in data set
mcalindon.big <- read.csv("McAlindon_Big.csv")

#confirm the data in order of id
mcalindon.big <- mcalindon.big %>%
  arrange(ID)

#only keep columns of mention in problem statement for cleanliness (and add bmi and opiate)
mcalindon.cleaned <- mcalindon.big %>%
  select(c(ID, age, agecat, sex, bmi, opiate, Race, racecat, inccat, retire, nsaid, pain.1, pain.2, pai

# get row numbers of rows that will match w WeatherDate
mcalindon.mother <- mcalindon.cleaned %>%
 group_by(ID) %>%
 mutate(rownum1 = ifelse(is.na(lastdt1), NA, which(WeatherDate == lastdt1[1])),
        rownum2 = ifelse(is.na(lastdt2), NA, which(WeatherDate == lastdt2[1])),
        rownum3 = ifelse(is.na(lastdt3), NA, which(WeatherDate == lastdt3[1])),
        rownum4 = ifelse(is.na(lastdt4), NA, which(WeatherDate == lastdt4[1])),
        rownum5 = ifelse(is.na(lastdt5), NA, which(WeatherDate == lastdt5[1])),
        rownum6 = ifelse(is.na(lastdt6), NA, which(WeatherDate == lastdt6[1])),
        rownum7 = ifelse(is.na(lastdt7), NA, which(WeatherDate == lastdt7[1])))

#pull the avgtemp at the identified row number of Weather Date
mcalindon.mother <- mcalindon.mother %>%
 group_by(ID) %>%
 mutate(avgtemp1 = avgtemp[rownum1[1]],
        avgtemp2 = avgtemp[rownum2[1]],
        avgtemp3 = avgtemp[rownum3[1]],
        avgtemp4 = avgtemp[rownum4[1]],
        avgtemp5 = avgtemp[rownum5[1]],
        avgtemp6 = avgtemp[rownum6[1]],
        avgtemp7 = avgtemp[rownum7[1]])

# to standardize times
mcalindon.mother <- mcalindon.mother %>%
  group_by(ID) %>%
  mutate(lastdt2 = ifelse(is.na(lastdt2), NA, lastdt2-lastdt1),
         lastdt3 = ifelse(is.na(lastdt3), NA, lastdt3-lastdt1),
         lastdt4 = ifelse(is.na(lastdt4), NA, lastdt4-lastdt1),
         lastdt5 = ifelse(is.na(lastdt5), NA, lastdt5-lastdt1),
         lastdt6 = ifelse(is.na(lastdt6), NA, lastdt6-lastdt1),
         lastdt7 = ifelse(is.na(lastdt7), NA, lastdt7-lastdt1),
         lastdt1 = ifelse(is.na(lastdt1), NA, 0))


#remove all duplicates, only the first row off each id (person) will remain
mcalindon.mother <- mcalindon.mother[!duplicated(mcalindon.mother$ID),]

## Build Long Dataset

# put data (mcalindon.mother) in long data format
pain.melt <- melt(mcalindon.mother,
                  na.rm = FALSE,
```

```
                  id.vars = c("ID",
                              "age",
                              "sex",
                              "inccat",
                              "retire",
                              "nsaid",
                              "bmi",
                              "opiate"),
                  measure.vars = c("pain.1",
                                   "pain.2",
                                   "pain.3",
                                   "pain.4",
                                   "pain.5",
                                   "pain.6",
                                   "pain.7"),
                  variable.name = "pain",
                  value.name="painscore")

temp.melt <- melt(mcalindon.mother,
                  na.rm = FALSE,
                  id.vars = c("ID"),
                  measure.vars = c("avgtemp1",
                                   "avgtemp2",
                                   "avgtemp3",
                                   "avgtemp4",
                                   "avgtemp5",
                                   "avgtemp6",
                                   "avgtemp7"),
                  variable.name = "tempdate",
                  value.name="avgtemp")

date.melt <- melt(mcalindon.mother,
                  na.rm = FALSE,
                  id.vars = c("ID"),
                  measure.vars = c("lastdt1",
                                   "lastdt2",
                                   "lastdt3",
                                   "lastdt4",
                                   "lastdt5",
                                   "lastdt6",
                                   "lastdt7"),
                  variable.name = "date",
                  value.name="datenum")

# build long dataset out of melted columns
mcalindon.long <- cbind(temp.melt, pain.melt, date.melt)

# get rid of extra ID columns to prep for analysis
mcalindon.long[4] <- NULL
mcalindon.long[13] <- NULL

# build cleaned long dataset and add scaled time and temperature
mcalindon.long <- mcalindon.long %>%
```

```
  arrange(ID) %>%
  select(c(ID,
           pain,
           painscore,
           tempdate,
           avgtemp,
           date,
           datenum,
           age,
           sex,
           inccat,
           retire,
           nsaid,
           bmi,
           opiate)) %>%
  # scale time and temperature to help models converge
  mutate(timescaled = scale(datenum)) %>%
  mutate(tempscaled = scale(avgtemp))
# restructure variables to match above (better binaries)
mcalindon.long$sex = ifelse(mcalindon.long$sex==1, 0, 1)
mcalindon.long$retire = ifelse(mcalindon.long$retire==2, 1, 0)


# build factor versions of discrete variables for graphing ease

#mcalindon.long$sex.factor = as.factor(ifelse(mcalindon.long$sex==0, "male", "female"))
#mcalindon.long$retire.factor = as.factor(ifelse(mcalindon.long$retire==1, "retired", "working"))
#mcalindon.long$nsaid.factor = as.factor(ifelse(mcalindon.long$nsaid==1, "nsaid", "non-nsaid"))
#mcalindon.long$opiate.factor = as.factor(ifelse(mcalindon.long$opiate==1, "opiate-use", "no-use"))

# Which variables to include, linear relationships? or transformations necessary?
library(GGally)

ggpairs(mcalindon.long, columns=c("sex", "retire", "inccat", "age", "nsaid", "bmi", "opiate"),
    diag=list(continuous="density", discrete="bar"), axisLabels="show")
library(naniar)

#investigate missing values
vis_miss(mcalindon.long)

# remove missing variables in outcome and random effects and remove inccat
mcalindon.long.nona <- mcalindon.long[!is.na(mcalindon.long$painscore),]
mcalindon.long.nona <- mcalindon.long.nona[!is.na(mcalindon.long.nona$tempscaled),]
mcalindon.long.nona <- mcalindon.long.nona[!is.na(mcalindon.long.nona$timescaled),]

mcalindon.long.nona$inccat <- NULL
mcalindon.long.nonaretire <-  mcalindon.long.nona[!is.na(mcalindon.long.nona$retire),]
vis_miss(mcalindon.long.nona)
library(knitr)
library(kableExtra)

aic.bic.table <- data.frame(c(5450.5, 5465.5), c(5264.3, 5289.4), c(5427, 5452.1), c(5270, 5310.1))
aic.bic.table <- t(aic.bic.table)
```

```
colnames(aic.bic.table) <- c("AIC", "BIC")
rownames(aic.bic.table) <- c("Random Intercept Model",
                             "Time Model",
                             "Temp Model",
                             "Time and Temp Model")

#build table
kable(aic.bic.table) %>%
  kable_styling(bootstrap_options = "striped", full_width = F, position = "center")
library(knitr)
library(kableExtra)

aic.bic.table2 <- data.frame(c(5264.3, 5289.4, NA),
                             c(5264.4, 5294.5, 0.17),
                             c(5264.1, 5294.2, 0.14),
                             c(4117.6, 4146.2, 0.39),
                             c(5266.2, 5296.3, 0.82),
                             c(5253.8, 5283.9, "0.0004"),
                             c(5255.2, 5285.3, "0.0008"),
                             c(5246.3, 5281.4, "1.691e-05"))
aic.bic.table2 <- t(aic.bic.table2)
colnames(aic.bic.table2) <- c("AIC", "BIC", "P-Val Compared to Time Model")
rownames(aic.bic.table2) <- c("Time Model",
                              "Time + Age Model",
                              "Time + Sex Model",
                              "Time + Retire Model (Retire NA's Removed)",
                              "Time + NSAID",
                              "Time + BMI Model",
                              "Time + Opiate Model",
                              "Time + BMI + Opiate")

#build table
kable(aic.bic.table2) %>%
  kable_styling(bootstrap_options = "striped", full_width = F, position = "center")
library("lme4")
# fitting multilevel model
multi.mod.mcalindon = lmer(painscore~1+(1|ID),data=mcalindon.long.nona,REML=F) # random intercept model
#summary(multi.mod.mcalindon)
#fixef(multi.mod.mcalindon)  # Get fixed

multi.mod.mcalindon.time = lmer(painscore~1+(1+timescaled|ID),data=mcalindon.long.nona,REML=F)
#summary(multi.mod.mcalindon.time)
#plot(multi.mod.mcalindon.time)
#plot(ranef(multi.mod.mcalindon.time), resid.int, pty="s", xlab = "Fitted",ylab="Residuals")

multi.mod.mcalindon.temp = lmer(painscore~1+(1+tempscaled|ID),data=mcalindon.long.nona,REML=F)
#summary(multi.mod.mcalindon.temp)

multi.mod.mcalindon.time.temp = lmer(painscore~1+(1+tempscaled+timescaled|ID),data=mcalindon.long.nona,
#summary(multi.mod.mcalindon.time.temp)

#anova(multi.mod.mcalindon,multi.mod.mcalindon.time)
# time is significant
```

```r
#anova(multi.mod.mcalindon,multi.mod.mcalindon.temp)
# temperature is not significant
#anova(multi.mod.mcalindon.time,multi.mod.mcalindon.time.temp)
#anova(multi.mod.mcalindon.temp,multi.mod.mcalindon.time.temp)
# confirmation that we should keep time and ditch temperature

# to test : age, sex, retire, nsaid, bmi, opiate
multi.mod.mcalindon.time.bmi = lmer(painscore~1+bmi+(1+timescaled|ID),data=mcalindon.long.nona,REML=F)
multi.mod.mcalindon.time.opiate = lmer(painscore~1+opiate+(1+timescaled|ID),data=mcalindon.long.nona,RE
multi.mod.mcalindon.time.bmi.opiate = lmer(painscore~1+bmi+opiate+(1+timescaled|ID),data=mcalindon.long
#summary(multi.mod.mcalindon.time.bmi.opiate)
multi.mod.mcalindon.full = lmer(painscore~1+bmi+opiate+age+sex+nsaid+(1+timescaled|ID),data=mcalindon.l
#summary(multi.mod.mcalindon.full)

#anova(multi.mod.mcalindon.time.bmi,multi.mod.mcalindon.time.bmi.opiate)
#anova(multi.mod.mcalindon.time.opiate,multi.mod.mcalindon.time.bmi.opiate)
#anova(multi.mod.mcalindon.time,multi.mod.mcalindon.time.bmi.opiate)

plot(multi.mod.mcalindon.time.bmi.opiate, main="General", xlab = "Fitted",ylab="Residuals")
plot(ranef(multi.mod.mcalindon.time.bmi.opiate), resid.int, pty="s", main="Random Effects", xlab = "Fit

library(knitr)
library(kableExtra)
library(ggplot2)

random.effects <- data.frame(c(8.416, 2.901),
                             c(1.721, 1.312))
random.effects <- t(random.effects)
colnames(random.effects) <- c("Variance", "Std. Dev.")
rownames(random.effects) <- c("Random Intercept",
                              "Random Time Scaled")


fixed.effects <- data.frame(c(4.82645, 0.84031, 5.744),
                            c(0.08402, 0.02506, 3.353),
                            c(3.61013, 1.15240, 3.133))
fixed.effects <- t(fixed.effects)
colnames(fixed.effects) <- c("Estimate", "Std. Error", "T-Value")
rownames(fixed.effects) <- c("Fixed Intercept",
                             "Fixed BMI",
                             "Fixed Opiate")

#build table
kable(random.effects) %>%
  kable_styling(bootstrap_options = "striped", full_width = F, position = "center")

kable(fixed.effects) %>%
  kable_styling(bootstrap_options = "striped", full_width = F, position = "center")

mcalindon.long.nona$predictions <- predict(multi.mod.mcalindon.time.bmi.opiate)

ggplot(mcalindon.long.nona) +
  geom_point(aes(x=painscore, y=predictions)) +
```

```
  labs(title="Empiric Values v. Predicted Values", x="Empiric Pain Score", y="Model Predictions")
## Organize and Clean Data / Build Mother Dataset
setwd("~/Desktop/masters/PHP2550/")
library(dplyr)
library(tidyr)
library(reshape2)

srrs.dat <- read.csv("srrs2.txt")
cty.dat <- read.csv("cty.txt")

srrs.clean <- srrs.dat %>%
  select(c(idnum, state2, cntyfips, county, typebldg, basement, floor, activity)) %>%
  filter(state2 == "MN") %>%
  mutate(single.fam.house = recode(typebldg,
                                   "0" = NULL,
                                   "1" = 1,
                                   "2" = 0,
                                   "3" = 0,
                                   "4" = 0,
                                   "5" = 0,
                                   .missing = NULL)) %>%
  mutate(basement.bin = recode(basement,
                               Y = 0,
                               N = 1,
                               .missing = NULL)) %>%
  mutate(floor.bin = recode(floor,
                            "0" = 0,
                            "1" = 1,
                            "2" = 2,
                            "9" = NULL,
                            .missing = NULL)) %>%
  mutate(log.radon = log (ifelse (activity==0, .1, activity)))

cty.clean <- cty.dat %>%
  filter(st == "MN") %>%
  select(c(ctfips, Uppm)) %>%
  mutate(uppm.log = log(Uppm))
## Build Long Dataset

# join sets for full long data set
radon.dat <- srrs.clean %>%
  left_join(unique(cty.clean), by=c("cntyfips" = "ctfips"))
# Which variables to include, linear relationships? or transformations necessary?
library(GGally)

# build factor versions of discrete variables for graphing ease

radon.dat$single.fam.house.factor = as.factor(ifelse(radon.dat$single.fam.house==1, "single-family", "o
radon.dat$floor.bin.factor = as.factor(ifelse(radon.dat$floor.bin==0, "basement", "no-basement"))
radon.dat$basement.bin.factor = as.factor(ifelse(radon.dat$basement.bin==0, "basement", "first-floor"))

ggpairs(radon.dat, columns=c("single.fam.house", "floor.bin", "basement.bin", "activity", "log.radon",
    diag=list(continuous="density", discrete="bar"), axisLabels="show")
```

```r
#ggpairs(radon.dat, columns=c("single.fam.house.factor", "floor.bin.factor", "basement.bin.factor", "ac
library(naniar)

#investigate missing values
vis_miss(radon.dat)

# remove missing variables
radon.dat.rm <- na.omit(radon.dat)
vis_miss(radon.dat.rm)
library(knitr)
library(kableExtra)

aic.bic.table.yay <- data.frame(c(2823.8,  2838.9),
                            c(2777.9, 2798.0),
                            c(2690.0, 2720.2),
                            c(2757.7, 2787.9),
                            c(2693.1, 2738.4),
                            c(2685.8, 2731.1),
                            c(2690.1, 2755.5))
aic.bic.table.yay <- t(aic.bic.table.yay)
colnames(aic.bic.table.yay) <- c("AIC", "BIC")
rownames(aic.bic.table.yay) <- c("Random Intercept Model",
                             "log(Ur) Model",
                             "log(Ur) + Floor Model",
                             "log(Ur) + Basement Model",
                             "log(Ur) + Floor + Basement Model",
                             "log(Ur) + Floor + Single.Fam Model",
                             "log(Ur) + Floor + Basement + Single.Fam Model")

#build table
kable(aic.bic.table.yay) %>%
  kable_styling(bootstrap_options = "striped", full_width = F, position = "center")
# fitting multilevel model
library("lme4")
library("ggplot2")
# random intercept model
multi.mod.radon = lmer(log.radon~1+(1|cntyfips),data=radon.dat.rm,REML=F)
#plot(multi.mod.radon)

#summary(multi.mod.radon)
#fixef(multi.mod.radon)  # Get fixed

multi.mod.radon.two = lmer(log.radon~1+uppm.log+(1|cntyfips),data=radon.dat.rm,REML=F)
#anova(multi.mod.radon,multi.mod.radon.two)

multi.mod.radon.three = lmer(log.radon~1+uppm.log+(1+floor.bin|cntyfips),data=radon.dat.rm,REML=F)
#anova(multi.mod.radon.two,multi.mod.radon.three)
#plot(multi.mod.radon.three)

multi.mod.radon.three.half = lmer(log.radon~1+uppm.log+(1+basement.bin|cntyfips),data=radon.dat.rm,REML
#anova(multi.mod.radon.two,multi.mod.radon.three.half)

multi.mod.radon.four = lmer(log.radon~1+uppm.log+(1+single.fam.house+floor.bin|cntyfips),data=radon.dat
```

```
#anova(multi.mod.radon.three,multi.mod.radon.four)

multi.mod.radon.five = lmer(log.radon~1+uppm.log+(1+basement.bin+floor.bin|cntyfips),data=radon.dat.rm,
#anova(multi.mod.radon.three,multi.mod.radon.five)
#plot(multi.mod.radon.five)

multi.mod.radon.full = lmer(log.radon~1+uppm.log+(1+basement.bin+floor.bin+single.fam.house|cntyfips),da
#plot(multi.mod.radon.full)

#anova(multi.mod.radon.three,multi.mod.radon.full)


# everything except basement.bin. build table of anova's output for each model step through

multi.mod.radon.final = lmer(log.radon~1+uppm.log+(1+floor.bin+single.fam.house|cntyfips),data=radon.da
#summary(multi.mod.radon.final)
plot(multi.mod.radon.final, main="General", xlab = "Fitted",ylab="Residuals")
#plot(ranef(multi.mod.radon.final), resid.int, pty="s", main="Random Effects", xlab = "Fitted",ylab="Re

#anova(multi.mod.radon.final,multi.mod.radon.full)
library(knitr)
library(kableExtra)
library(ggplot2)

random.effects2 <- data.frame(c(0.08669, 0.2944),
                              c(0.48165, 0.6940),
                              c(0.12969, 0.3601))
random.effects2 <- t(random.effects2)
colnames(random.effects2) <- c("Variance", "Std. Dev.")
rownames(random.effects2) <- c("Random Intercept",
                               "Random Floor",
                               "Random Single.Fam")


fixed.effects2 <- data.frame(c(1.37182, 0.03158, 43.43),
                             c(0.83730, 0.07882, 10.62))
fixed.effects2 <- t(fixed.effects2)
colnames(fixed.effects2) <- c("Estimate", "Std. Error", "T-Value")
rownames(fixed.effects2) <- c("Fixed Intercept",
                              "Fixed log(Uppm)")

#build table
kable(random.effects2) %>%
  kable_styling(bootstrap_options = "striped", full_width = F, position = "center")

kable(fixed.effects2) %>%
  kable_styling(bootstrap_options = "striped", full_width = F, position = "center")

radon.dat.rm$predictions <- predict(multi.mod.radon.final)

ggplot(radon.dat.rm) +
  geom_point(aes(x=log.radon, y=predictions)) +
  labs(title="Empiric Values v. Predicted Values", x="Empiric Radon Value", y="Model Predictions")
```