Practical Data Analysis Assignment 3

Anthony Sisti

10/10/2019

Problem 1

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

zi= patient level factor (age, sex, race, income, occupation, working status, use of NSAIDs

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.

In the 'McAlindon_Big' data set, there are 300 instances when pain should have been measured for an individual, but no value is recorded. We omit these cases, and are left with 1135 pain measurements at different times, on 205 unique individuals. The between patient predictors we will examine include:

1. age - Years
2. treatment - 1 Yes, 0 No
3. sex - 1 Male, 0 Female
4. retire - 1 Retired, 0 Not
5. nsaid - 1 user, 0 not

Income was not considered because the variable is broken up into wide intervals, with over half of the the observations being missing values. Race was not considered because over 95% of cases were White Non-Hispanic. Occupation was not considered because there is no obvious grouping for the wide variety of careers listed. We use the 'retire' variable to indicate working status instead. We omit observations for which any of the variables we consider for prediction were not observed. We also remove any individuals who only have one pain observation, and any instances that temperature was not recorded.

This causes us to drop 48 total participants from the study, leaving us with 867 pain observations on 157 unique individuals. We compare a summary statistics for the dropped patients and the remaining patients to justify this removal.

```
## $`Dropped Observations`
##       Pain           avgtemp           age            treat
##  Min.   : 0.000   Min.   :12.00   Min.   :46.00   Min.   :0.0000
##  1st Qu.: 5.750   1st Qu.:43.25   1st Qu.:51.75   1st Qu.:0.0000
##  Median : 9.000   Median :67.00   Median :56.50   Median :0.0000
##  Mean   : 8.167   Mean   :57.96   Mean   :58.62   Mean   :0.4583
##  3rd Qu.:11.000   3rd Qu.:72.25   3rd Qu.:65.25   3rd Qu.:1.0000
##  Max.   :14.000   Max.   :83.00   Max.   :78.00   Max.   :1.0000
##
##       sex            retire       nsaid
##  Min.   :1.000   Min.   :1    Min.   :0.0000
##  1st Qu.:1.000   1st Qu.:1    1st Qu.:0.0000
##  Median :2.000   Median :1    Median :1.0000
##  Mean   :1.688   Mean   :1    Mean   :0.7292
##  3rd Qu.:2.000   3rd Qu.:1    3rd Qu.:1.0000
##  Max.   :2.000   Max.   :1    Max.   :1.0000
##                  NA's   :47
##
## $`Remaining Observations`
##       Pain           avgtemp           age            treat
##  Min.   : 0.000   Min.   :-5.00   Min.   :44.00   Min.   :0.0000
##  1st Qu.: 5.000   1st Qu.:48.00   1st Qu.:54.00   1st Qu.:0.0000
##  Median : 7.000   Median :62.00   Median :60.00   Median :1.0000
##  Mean   : 7.466   Mean   :58.85   Mean   :60.61   Mean   :0.5087
##  3rd Qu.:10.000   3rd Qu.:73.00   3rd Qu.:67.00   3rd Qu.:1.0000
##  Max.   :20.000   Max.   :95.00   Max.   :98.00   Max.   :1.0000
##       sex           retire         nsaid
##  Min.   :1.00   Min.   :1.00   Min.   :0.0000
##  1st Qu.:1.00   1st Qu.:1.00   1st Qu.:1.0000
##  Median :2.00   Median :1.00   Median :1.0000
##  Mean   :1.64   Mean   :1.46   Mean   :0.8235
##  3rd Qu.:2.00   3rd Qu.:2.00   3rd Qu.:1.0000
##  Max.   :2.00   Max.   :2.00   Max.   :1.0000
```

We see that 47 of the 48 participants were dropped because there was no information about whether or not they had retired, and the last participant was dropped because they only had one pain observation. It appears that each variable is centered around the same value in both groups. There are more extreme observations in the remaining group, but this is to be expected since it contains three times as many patients as the dropped group. We continue with the remaining patients and ignore the dropped observations, as it doesn't appear we lose very much information by dropping them. The exploratory data analysis that leads us to using multi-level models was conducted on Assignment 1, and we use it as a reference. We begin by fitting the unconditional means model, where intercept is made to vary by participant. The form of this model, and model output, are shown below.

$$\text{Pain}_{ij} = a_{0j} + e_{ij}$$

$$a_{0j} = \alpha_{00} + u_{0j}$$

Where $e_{ij} \sim N(0, \sigma^2)$ and $u_{0j} \sim N(0, \tau_{00})$

Table 1: Fixed Effects: Random Intercept Model

	Estimate	Std. Error	t value
(Intercept)	7.465	0.248	30.064

Table 2: Random Effects: Random Intercept Model

Group	Name	Variance	SD
ID	(Intercept)	8.627	2.937
Residual		5.386	2.321

We examine the effect of adding the within patient factors 'Time' and 'Average Temperature' to level one of our multi-level model as predictors of 'Pain'. Using analysis of variance, we compare these models to just the random intercept.

Table 3: ANOVA for Level 1 Predictors

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
Random_Intercept	3	4280.672	4294.967	-2137.336	4274.672	NA	NA	NA
Time_Only	4	4157.939	4176.999	-2074.969	4149.939	124.733	1	0

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
Random_Intercept	3	4280.672	4294.967	-2137.336	4274.672	NA	NA	NA
Temperature_Only	4	4279.242	4298.302	-2135.621	4271.242	3.43	1	0.064

Adding 'Time' to level one of the model is a significant improvement, while adding 'Temperature' is shown to be insignificant at the 5% level. Including interaction between them also does not add to the model wither, per ANOVA. We continue with 'Time' as our level one predictor of 'Pain'

We now examine how letting the coefficient for 'Time' vary between patient as a random effect influences the model. The ANOVA table comparing the fixed and random 'Time' models is shown below.

Table 4: Comparing Time as a Fixed Effect and Random Effect

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
Time_Only	4	4157.939	4176.999	-2074.969	4149.939	NA	NA	NA
Time_Random	6	4062.367	4090.957	-2025.183	4050.367	99.572	2	0

The model with 'Time' as a random effect is significantly better. We continue building our model with 'Time' as a predictor, while allowing both the intercept and 'Time' coefficient to vary between patients. The output of this model is shown in the tables below

Table 5: Fixed Effects: Random Intercept and Time Model

	Estimate	Std. Error	t value
(Intercept)	8.650	0.272	31.752
Time	-0.431	0.055	-7.852

Table 6: Random Effects: Random Intercept Model

Group	Variable 1	Variable 2	Var/Cov	SD/Corr
ID	(Intercept)		9.878	3.143
ID	Time		0.296	0.545
ID	(Intercept)	Time	-0.589	-0.344
Residual			3.136	1.771

In order to evaluate which patient level factors to include in the second level of our model, we first examine models of the form:

$$\begin{aligned}\text{Pain}_{ij} &= \beta_{0j} + \beta_{1j} * \text{Time}_{ij} + e_{ij} \\ \beta_{0j} &= \alpha_{00} + \alpha_{01} z_j + u_{0j} \\ \beta_{1j} &= \alpha_{10} + \alpha_{11} x_j + u_{1j}\end{aligned}$$

Where, $e_{ij} \sim N(0, \sigma^2)$ and

$$\begin{pmatrix} u_{0j} \\ u_{1j} \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & \tau_{10} \\ \tau_{10} & \tau_{11} \end{pmatrix} \right)$$

Note, z_j and x_j are between patient factors. We construct a loop which builds models containing every combination of patient level factors that could be included in a model, including the cases when there is only one patient level factor in each level 2 equation. We compare each nested model using ANOVA, and each non-nested models using BIC, keeping the model which remains significantly better when compared to each of the rest.

After comparing every possible combination of patient level factors, we determine that the model in which x_i and z_i indicate whether or not a patient uses NSAIDs is the most effective. We continue building our model with the NSAID indicator included in both of our level two regressions. Next, we run loops that test the effectiveness of adding extra patient level factors to level two ,adding interactions with the NSAID indicator at level two, or adding additional fixed or random effects at level two. The loops use ANOVA to determine if there is a significant benefit to adding these factors to the model.

After evaluating every possible combination and addition to level two of the model, we determine that in each case it is best to only include the NSAID indicator, with no interactions. Our final model is of the form:

$$\begin{aligned}\text{Pain}_{ij} &= \beta_{0j} + \beta_{1j} * \text{Time}_{ij} + e_{ij} \\ \beta_{0j} &= \alpha_{00} + \alpha_{01} NSAID_j + u_{0j} \\ \beta_{1j} &= \alpha_{10} + \alpha_{11} NSAID_j + u_{1j}\end{aligned}$$

Where, $e_{ij} \sim N(0, \sigma^2)$ and

$$\begin{pmatrix} u_{0j} \\ u_{1j} \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & \tau_{10} \\ \tau_{10} & \tau_{11} \end{pmatrix} \right)$$

The model output is shown in the tables below.

Table 7: Fixed Effects: Final Model

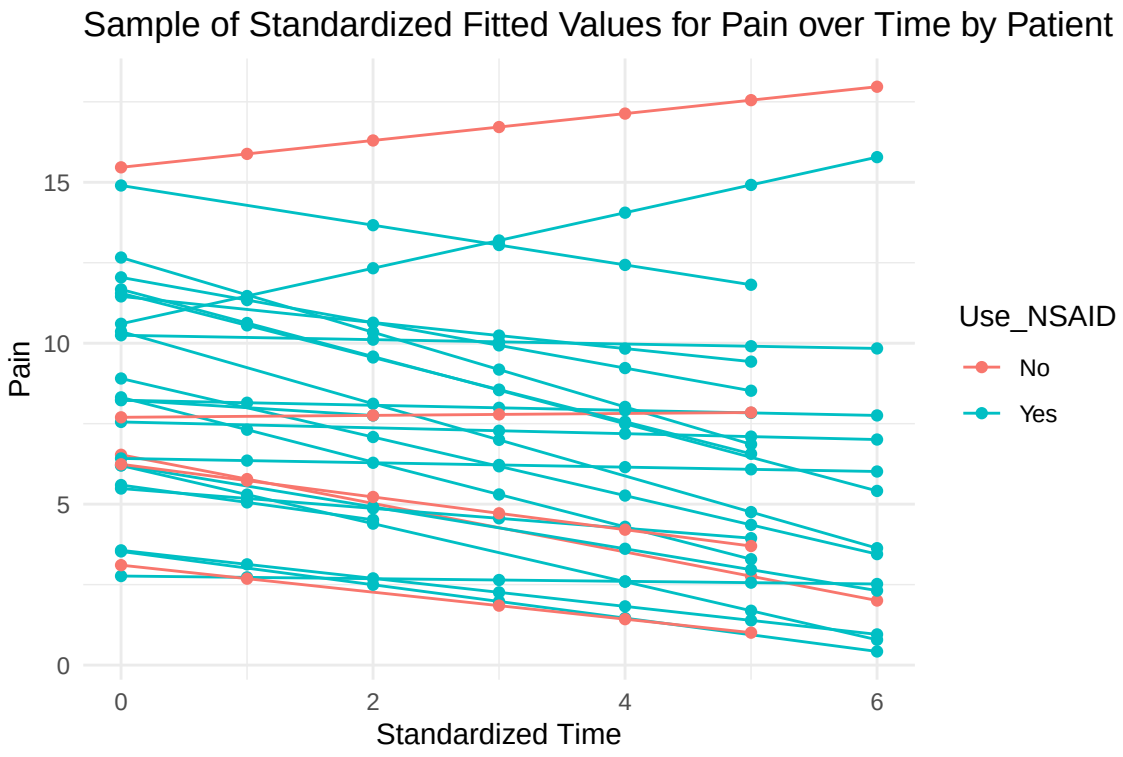
	Estimate	Std. Error	t value
(Intercept)	8.519	0.656	12.989
Time	-0.253	0.127	-1.984
nsaid	0.161	0.721	0.223
Time:nsaid	-0.218	0.141	-1.550

Table 8: Random Effects: Final Model

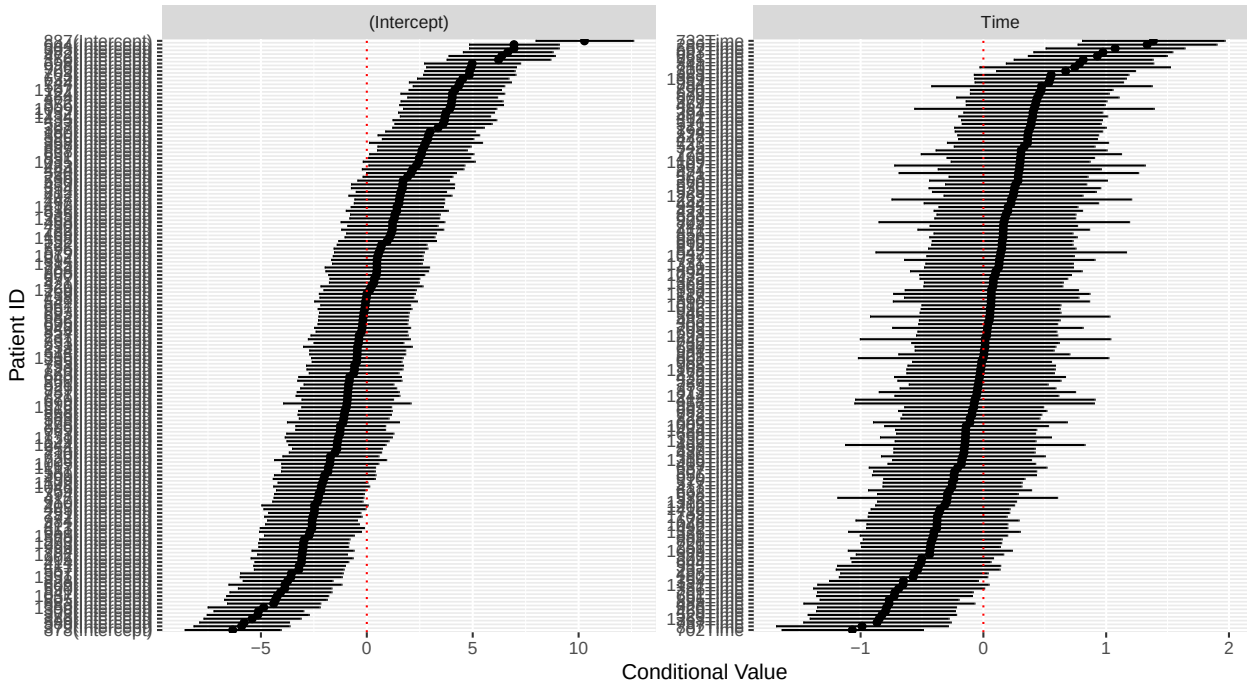
Group	Variable 1	Variable 2	Var/Cov	SD/Corr
ID	(Intercept)		9.871	3.142
ID	Time		0.289	0.537
ID	(Intercept)	Time	-0.580	-0.344
Residual			3.136	1.771

The random slope and intercept had a negative correlation, so higher initial pain tended to imply a smaller effect of time on the pain. The structure of the model tells us that patients who use NSAIDs have higher initial pain. The effect of 'Time' on 'Pain' was negative by itself, but for those who used NSAIDs, there was an even more dramatic expected decrease in pain over time.

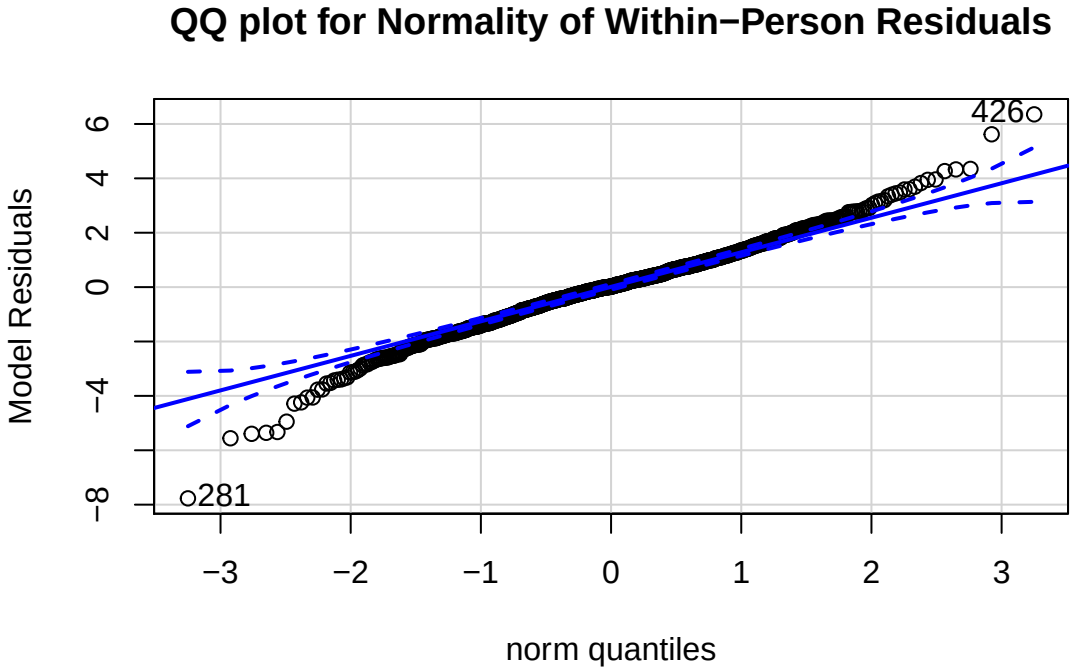
Below, we plot a sample of the within-patient regressions from this model.



As we expect, the slope and intercept coefficients vary across participant randomly, with use of NSAIDs as a between patient factor. Below, we plot the random effect values for the intercept and 'Time' by patient.



The random variation in the intercept is more dramatic than the random variation in the 'Time' effect, as can be seen by the range of the x-axis (and variance from the model output). There are also more Intercept coefficients significantly different from zero, and significantly different from each other. Despite this, there is still a visible pattern in the plot for 'Time', as the coefficients range from +1 to -1, which allows for substantial variation between patient 'Pain' prediction over the course of 6 measurements. We now analyze model residuals, first plotting overall model residuals below.

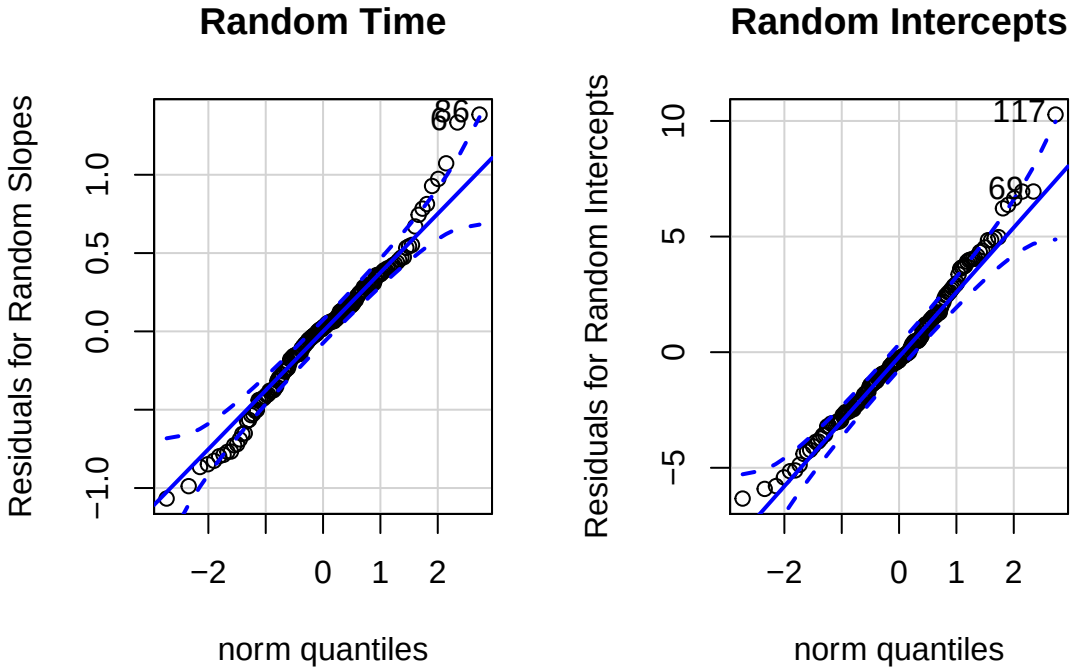


The residuals are fairly normal with slightly wider tails. Observations 426 and 281 appear to be outliers. Observation 281 has a pain score of 0, which is odd for this study, and observation 246 has an extremely high pain score for this study. These abnormal values are the likely cause for their outlier appearance.

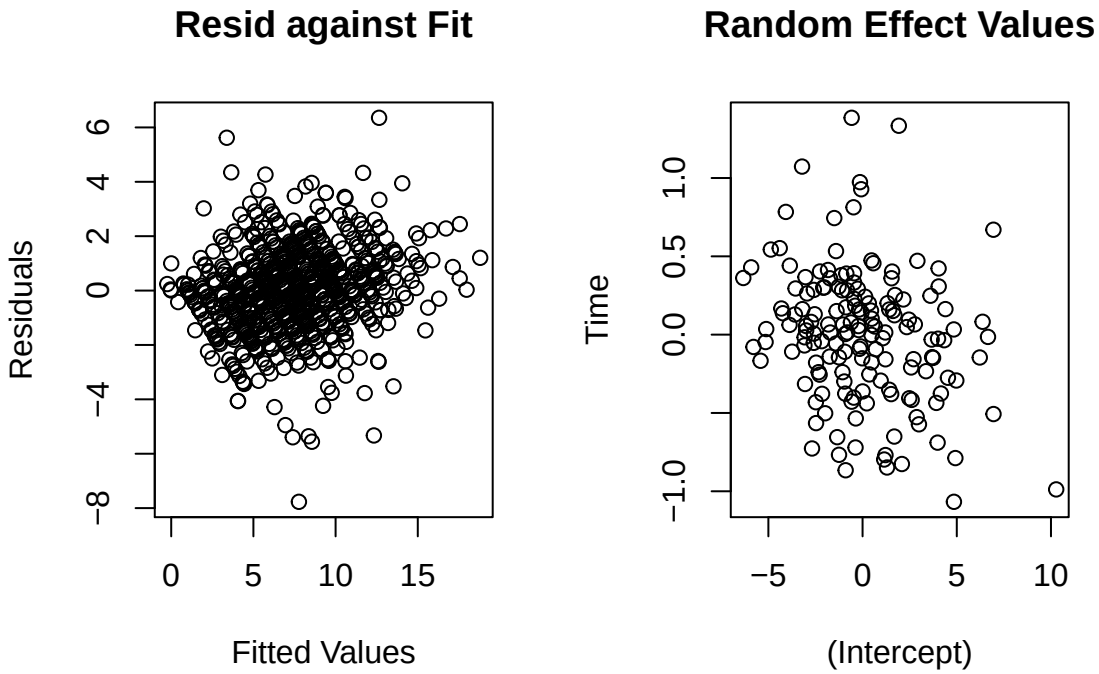
Table 9: Model Outliers

	ID	Pain	Time	avgtemp	age	treat	sex	retire	nsaid
281	557	0	3	57	0.5637857	1	1	0	0
426	703	19	2	49	-0.7957412	1	1	0	1

We now plot the between-person model residuals. For both random effects, the residuals are approximately normal with few deviations.



Finally, we plot the residuals against fitted values for the overall model, and the random slope values against random intercept values. There is no discernible pattern between the residual and fitted values, but there is a negative association between time and intercept. This was to be expected based on the negative correlation shown in the model output discussed earlier.



'Time' was found to be a more effective predictor of pain score than just an intercept or average temperature. The predictive ability was made even more effective when the coefficients for the intercept and 'Time' were made to vary across patient. The between patient factor that provided the most information for the intercept and 'Time' coefficient was the use of NSAID. The normality assumption was not quite satisfied, and possible variable transformations and additional factors should be investigated in a subsequent study.

Problem 2

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.

Srrs2.txt includes data from houses sampled within states and counties with one row per house. The key variables that you will need are:

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.

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.

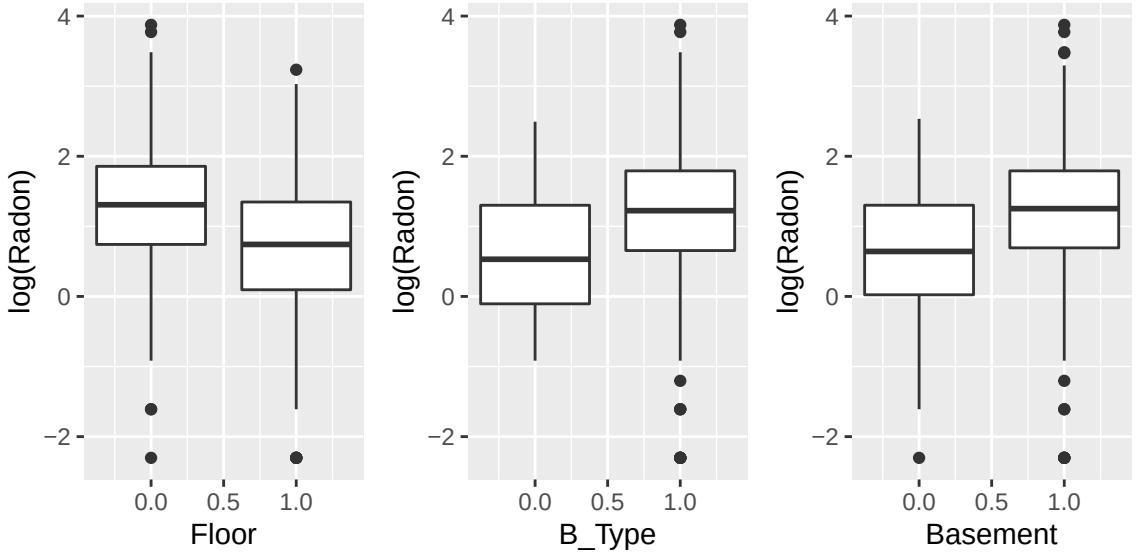
We begin by analyzing the data frames containing Uranium levels at the county-level and radon levels at the house level. In the house level data frame, we code any Radon measurement not conducted in the basement as a '1' in the floor variable, and any non single family home as a '0' in the building type variable. We then filter the data frame to only include counties in Minnesota, the state for which we will conduct the study, leaving us with 1188 observations. Taking the compete cases only, we lose 43 observations, less than 4% of the data, so we are not concerned with a loss in information. We then merge the Uranium and House data frames. Here, we lose 195 observations. While this is a large group, there is no information for the Uranium in their counties, making it impossible to evaluate its effectiveness as a county level predictor. We continue with our remaining 950 observations.

The variables we will consider, and their correlation matrix, can be found below. We consider the log transform of Uranium and Radon. For any Radon level of 0.0 we change it slightly, to 0.1, to allow for the logarithmic transformation.

1. log(Radon) - log of radon activity
2. log(Uranium) - log of the average Uranium reading in a county
3. B-Type - Building type (1 single family, 0 else)
4. Basement - Does the house have a basement (1 yes, 0 no)
5. Floor - Floor radon measurement was taken (1 first or above, 0 basement)

	B_Type	Basement	Floor	Radon	Uranium
B_Type	1.0000000	0.1367704	-0.1195650	0.0999214	-0.0404265
Basement	0.1367704	1.0000000	-0.5291863	0.1149904	0.0538511
Floor	-0.1195650	-0.5291863	1.0000000	-0.1370951	0.0349792
Radon	0.0999214	0.1149904	-0.1370951	1.0000000	0.2818879
Uranium	-0.0404265	0.0538511	0.0349792	0.2818879	1.0000000

Radon has the strongest linear association with Uranium, indicating that it should be useful at level two of the model. There aren't any extremely strong correlations between predictors, but 'floor' and 'basement' do appear to have some negative association. We now visualize the the effect of each house level predictor on radon level.



The plot shows that the floor on which the measurement was taken seems to have an opposite effect on Radon levels as 'B_Type' and 'Basement'. The effect of the type of building, and whether are not that building has a basement are nearly identical. After further inspection, they share the same value in over 90% of the

observations from the data set. With this in mind, we run regressions to determine which of the two variables to consider for the multi-level the model.

Table 10: Regresson of Radon at House Level with No Slope

	Dependent variable:	
	log(Radon)	
	(1)	(2)
Basement	1.225***	
	(0.030)	
B_Type		1.212***
		(0.029)
Observations	950	950
R^2	0.644	0.646
Adjusted R^2	0.643	0.645
Residual Std. Error (df = 949)	0.880	0.877
F Statistic (df = 1; 949)	1,714.117***	1,730.114***
Note:	*p<0.1; **p<0.05; ***p<0.01	

The results are extremely similar, but for the sake of this analysis, we use 'B_Type' because it has a slightly higher R^2 value. It is also worth noting that for some counties, the 'B_Type' variable is a column of 1's. As a result of this, we remove the fixed intercept in any model that includes building type as a predictor of Radon level, where the effect of 'B_Type' is varied by county. We begin model fitting below by testing models where the intercept, effect of 'Floor' and effect of 'B_Type' are varied by county. An ANOVA table for four initial models is shown below.

Table 11: Initial Model Screening

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
BType_Rand_Int	3	2366.562	2381.131	-1180.281	2360.562	NA	NA	NA
Floor_Rand_Int	4	2281.286	2300.712	-1136.643	2273.286	87.27534	1	0
Rand_BType	5	2364.420	2388.703	-1177.210	2354.420	0.00000	1	1
Rand_Floor_Int	6	2283.361	2312.500	-1135.681	2271.361	83.05901	1	0

Based on the table above, the model where the effect of 'Floor' and the intercept vary by county, and the model where intercept varies and 'Floor' is a fixed effect, are significantly better than the random intercept model where 'B_Type' is a fixed effect. When comparing the two 'Floor' models, we use BIC. The model in which 'Floor' is a fixed effect and the intercept is random has a lower value. This model takes the form:

$$\log(\text{Radon})_{ij} = \beta_{0j} + \beta_{1j} * \text{Floor}_{ij} + e_{ij}$$
$$\beta_{0j} = \alpha_{00} + u_{0j}$$

Where $e_{ij} \sim N(0, \sigma^2)$ and $u_{0j} \sim N(0, \tau_{00})$.

We continue building from this model. Next, we considered the addition of 'B_Type' as a fixed effect, the addition of 'B_Type' as a random effect, and using Uranium as a predictor of intercept, 'B_Type' and 'Floor' as random effects. An ANOVA table for these models is shown below.

Table 12: Second Model Screening

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
Base_Model	4	2281.286	2300.712	-1136.643	2273.286	NA	NA	NA
Floor_BType	4	2318.999	2338.425	-1155.499	2310.999	0.00000	0	1.00e+00
Floor_Rand_BType	6	2302.255	2331.394	-1145.127	2290.255	20.74404	2	3.13e-05
Floor_Rand_UBType	8	2272.695	2311.547	-1128.348	2256.695	33.55989	2	1.00e-07
Rand_UFloor	8	2251.013	2289.864	-1117.506	2235.013	21.68253	0	0.00e+00
BType_Rand_UFloor	9	2236.282	2279.991	-1109.141	2218.282	16.73009	1	4.31e-05

All of the models, except when 'Floor' and 'B_Type' are fixed effects with varying intercept, are better than the base model. In particular, it appears that the models where Uranium predicts the random effects of the intercept and 'Floor' or 'B_Type' are the best. We run a separate ANOVA looking at just these two models.

Table 13: Third Model Screening

	Df	AIC	BIC	logLik	deviance	Chisq	Chi Df	Pr(>Chisq)
Floor_Rand_UBType	8	2272.695	2311.547	-1128.348	2256.695	NA	NA	NA
BType_Rand_UFloor	9	2236.282	2279.991	-1109.141	2218.282	38.41262	1	0

The results of this ANOVA suggests (by BIC and p-value) the best model is when 'B_Type' is a fixed effect with random intercept and random 'Floor' effect being predicted by 'Uranium'. The form of this model, and its parameter estimates, are found below:

$$\log(\text{Radon})_{ij} = \beta_{0j} + \beta_{1j} * \text{Floor}_{ij} + \beta_{2j} * \text{BType}_{ij} + e_{ij}$$
$$\beta_{0j} = \alpha_{00} + \alpha_{01}\log(\text{Uranium})_j + u_{0j}$$
$$\beta_{1j} = \alpha_{10} + \alpha_{11}\log(\text{Uranium})_j + u_{1j}$$

Where, $e_{ij} \sim N(0, \sigma^2)$ and

$$\begin{pmatrix} u_{0j} \\ u_{1j} \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & \tau_{10} \\ \tau_{10} & \tau_{11} \end{pmatrix}\right)$$

Table 14: Fixed Effects: Final Model

	Estimate	Std. Error	t value
(Intercept)	0.967	0.127	7.640
B_Type	0.498	0.121	4.111
log(Uranium)	0.802	0.115	6.996
Floor	-0.637	0.083	-7.715
log(Uranium):Floor	-0.430	0.225	-1.908

Table 15: Random Effects: Final Model

Group	Variable 1	Variable 2	Var/Cov	SD/Corr
County	(Intercept)		0.054	0.233
County	Floor		0.066	0.257
County	(Intercept)	Floor	0.000	0.004
Residual			0.567	0.753

The intercept coefficient is 0.967, indicating that the average Radon level for buildings that are single family homes for which the measurement in the basement was $e^{0.967} = 2.63$. The 'B_type' coefficient is 0.498, implying that the geometric mean for radon levels increases by $e^{0.498} - 1 = 0.645 \approx 65\%$ when houses are single family homes. The random slope corresponding to 'Floor' and random intercept were nearly uncorrelated in this model. The standard deviation within the random 'Floor' slopes was 0.257 while the standard deviation within the random intercepts was 0.233. The fixed effect for log(Uranium) was positive while the fixed effects for Floor and the interaction between Floor and log(Uranium) were negative. This is difficult to interpret in general, so use one case from the study as an example. Consider the building with 'House ID' 5801.

Table 16: Example Case Factors

H_ID	State	County	B_Type	Basement	Floor	Radon	Uranium
5081	MN	AITKIN	1	0	1	2.2	0.502054

Table 17: Example Case Random Components

	(Intercept)	Floor
AITKIN	-0.0480729	0.0274808

Filling in our model with the above information, we have the following:

$$\widehat{\log(\text{Radon})}_{ij} = \widehat{\beta_{0j}} + \widehat{\beta_{1j}} * (1) + 0.498 * (1)$$

$$\widehat{\beta_{0j}} = 0.967 + (0.802)\log(0.502) + -0.048$$

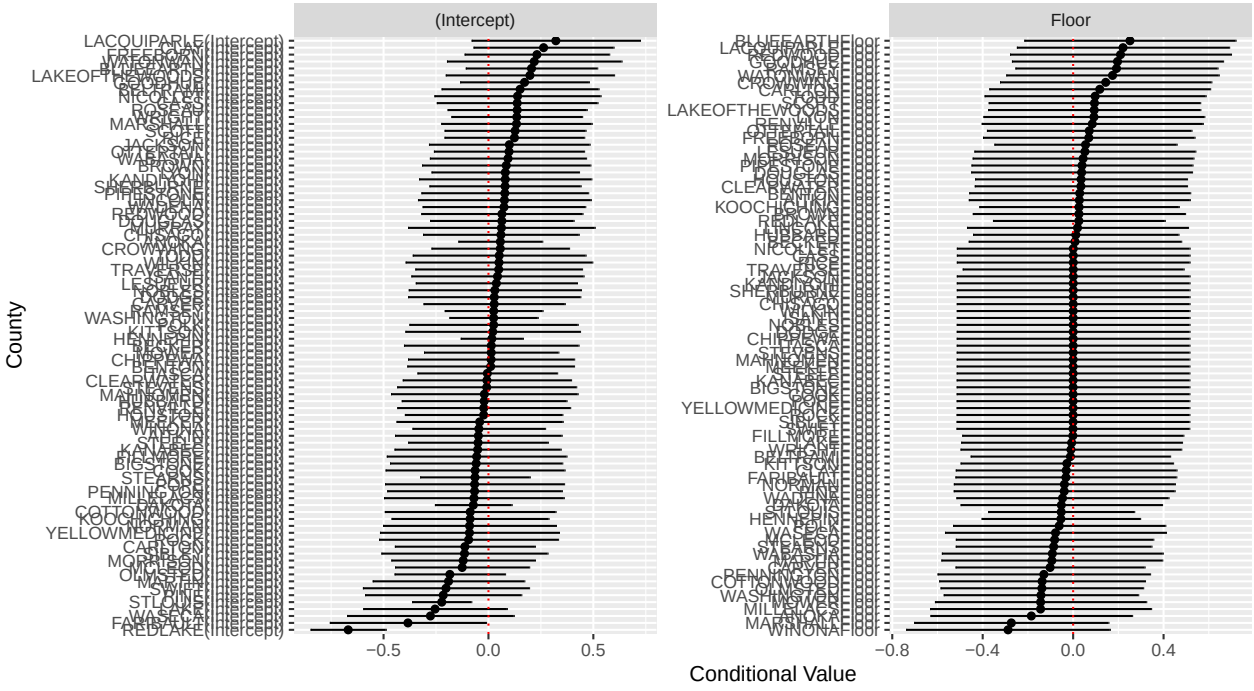
$$\widehat{\beta_{1j}} = -0.637 + (-0.430)\log(0.502)_j + 0.027$$

Solving, and plugging in the coefficients in the appropriate places we find:

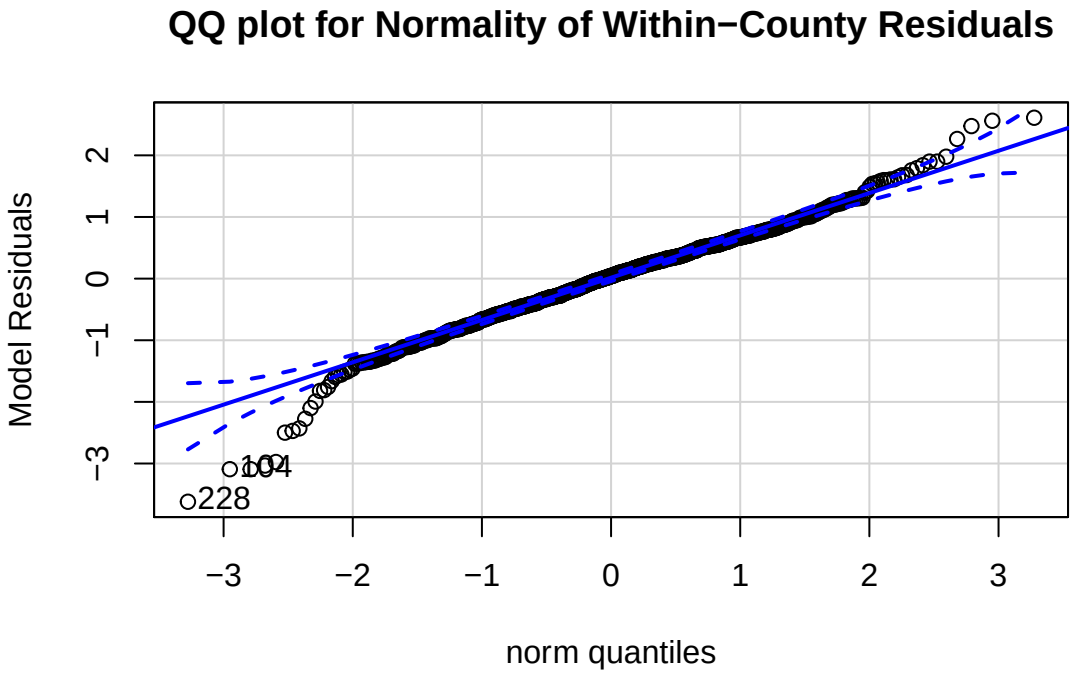
$$\widehat{\log(\text{Radon})}_{ij} = 0.366 + -0.314 * (1) + 0.498 * (1) = 0.55$$

This implies that Radon $= e^{0.55} = 1.73$. We notice that the Uranium level being below 1, implying that log(Uranium) was negative, increased the floor coefficient. While the coefficient still remained negative overall, it was less negative due to the lower levels of Uranium. The opposite effect was seen for the intercept. The lower Uranium level decreased the intercept, which makes sense, as we would expect counties with lower Uranium levels to have lower Radon readings in general.

Below, we plot the random effect values for the intercept and 'Floor' by county.



Based on the plots above, the random effects form both floor and intercept do not appear to show extreme deviation from zero or one another. This is particularly true of the "Floor" random effect, where none of the random effects appear to deviate from one another. 3We now conditioner model residuals, within county.



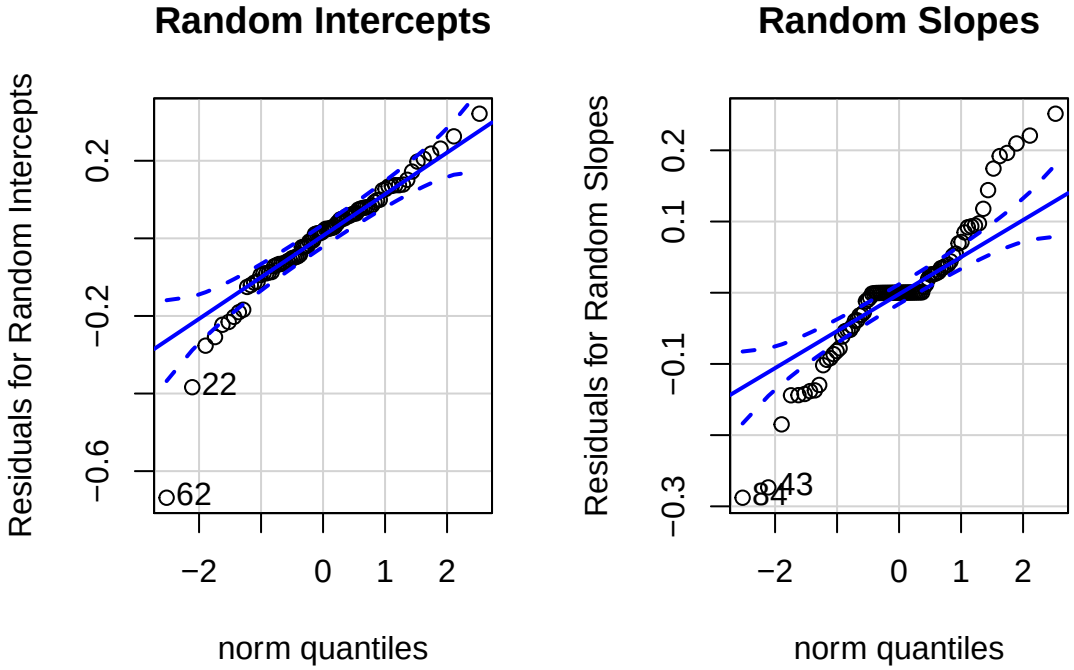
The residuals are approximately normal with a few outliers at the left tail. In particular, observations 228 and 104 deviate very far from normality. These counties had a Radon level of 0.1, likely corrected from a reading of 0.0 . The high Uranium levels would lead us to believe there would be a higher Radon levels, which explains the large deviation from the model.

We now plot the between-county model residuals. Outside of a few outliers, the random intercepts appears to be normal. The random slopes show an pattern that deviates far from normality. The line near zero is

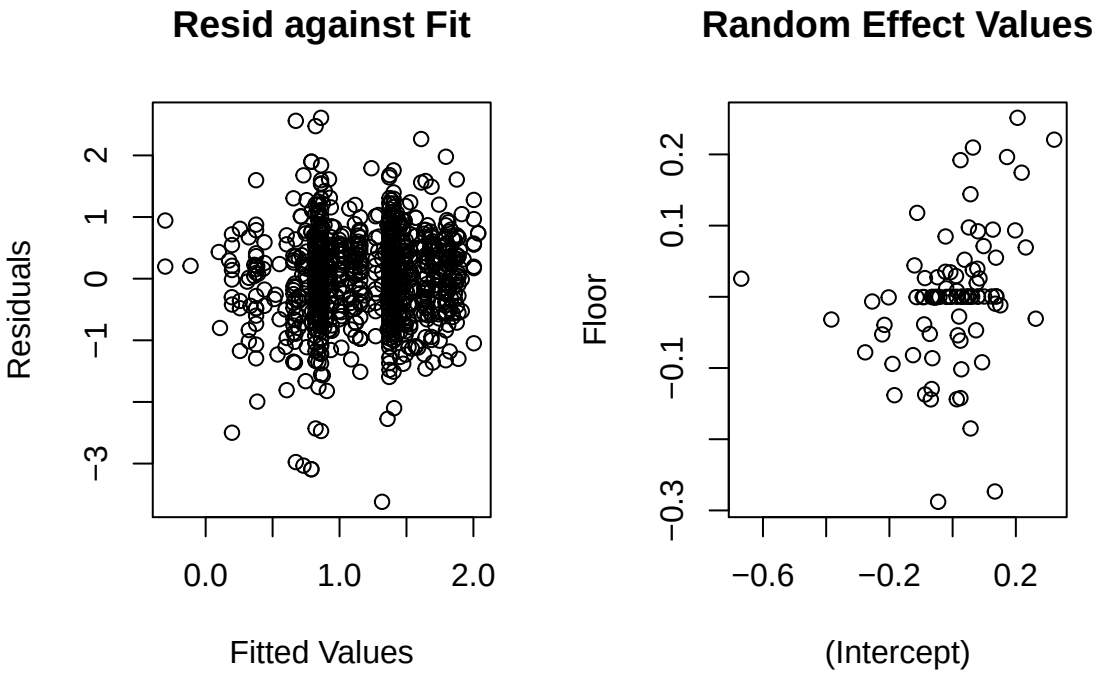
Table 18: Model Outliers

	H_ID	State	County	B_Type	Basement	Floor	Radon	Uranium
228	5319	MN	FARIBAULT	1	1	0	0.1	1.34316
104	5187	MN	CARVER	1	1	1	0.1	1.10061

likely a result of observations with the same values or close to the same values within a county, so given the Uranium level, you can exactly give the effect of 'Floor' solely from the fixed effects at the second level.



Finally, we plot the residuals against fitted values for the overall model and the random slope vs random intercept. There is no discernible trend between the residual and fitted values, but fitted values do seem to be concentrated just below 1 and just below 1.5 . There is no pattern between the random slopes and random intercept. This was to be expected based on the lack of correlation shown in the model output that was discussed earlier.



The best mulit-level model for predicting Radon levels for buildings in MN was when the type of building was a fixed effect, while the intercept and the floor on which a measurement was taken were random effects predicted by Uranium levels. The model residuals deviate slightly from normality, and the random slopes deviate far from normality. Further studies should investigate proper factor transformations that allow the model to abide closer to assumptions of normality.

Problem 3

Expand your model to 3 levels by considering all of the states. Level 1 is now house within county within state; level 2 is county within state and level 3 is state (there are no state level variables).

We now attempt to scale our model to the whole data set, while adding clusters at the state-level. We take all of the complete cases from the overall house level data set. This causes us to lose $769/12777 \approx 6\%$ of our data. We assume we are not losing too much information, and continue to merge the full Radon house data set with the county-level Uranium data set. When doing this we drop $27/386 \approx 7\%$ of counties. We again assume that this is few enough cases that we still retain a large majority of the information in the data set. We merge the state and county identifiers into one column to ensure that all unique combinations are considered, and counties with the same name and different state are not grouped together. In order to ensure model convergence, we remove the state of Michigan (15 cases) because it only consists of one county and any county that has 2 or fewer towns (55 cases). This reduces our total from 11538 to 11468, less than 1% of the data.

Since there are no state-level predictors, we decide to build off of our model from the previous question and add random slopes and intercepts at the second level. We force the correlations between these random effects to be zero to maintain some simplicity. We also remove the random slope of log(Uranium) corresponding to β_{1j}; this ensures non-singularity. This model takes the form:

$$\log(\text{Radon})_{ijt} = \beta_{0jt} + \beta_{1jt} * \text{Floor}_{ijt} + \beta_2 * \text{BType}_{ijt} + e_{ijt}$$

\

$$\beta_{0jt} = \alpha_{00t} + \alpha_{01t}\log(\text{Uranium})_{jt} + u_{0jt}$$
$$\beta_{1jt} = \alpha_{10t} + \alpha_{11t}\log(\text{Uranium})_{jt} + u_{1jt}$$

$$\alpha_{00t} = \eta_{000} + v_{0t}$$
$$\alpha_{10t} = \eta_{100} + v_{1t}$$
$$\alpha_{01t} = \eta_{010} + v_{2t}$$

Where, $e_{ijt} \sim N(0, \sigma^2)$ and

$$\begin{pmatrix} u_{0j} \\ u_{1j} \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \tau_{00} & \tau_{10} \\ \tau_{10} & \tau_{11} \end{pmatrix}\right)$$

$$\begin{pmatrix} v_{0t} \\ v_{1t} \\ v_{2t} \end{pmatrix} \sim \begin{pmatrix} \phi_{00} & 0 & 0 \\ 0 & \phi_{11} & 0 \\ 0 & 0 & \phi_{22} \end{pmatrix}$$

The output for this model is displayed below.

Table 19: Fixed Effects: Three-Level Model

	Estimate	Std. Error	t value
(Intercept)	0.684	0.170	4.030
B_Type	0.233	0.038	6.103
log(Uranium)	0.450	0.136	3.303
Floor	-0.780	0.052	-15.091
log(Uranium):Floor	0.013	0.071	0.182

Table 20: Random Effects: Three-Level Model

Group	Variable 1	Variable 2	Var/Cov	SD/Corr
State.County_State	(Intercept)		0.132	0.364
State.County_State	Floor		0.067	0.259
State.County_State	(Intercept)	Floor	-0.030	-0.322
State	(Intercept)		0.180	0.425
State.1	log(Uranium)		0.065	0.256
State.2	Floor		0.001	0.031
Residual			0.876	0.936

We find that all of the fixed effects keep the same sign except for the interaction between 'log(Uranium)' and 'Floor'. It is worth noting that the standard error of this coefficient is larger than the magnitude of the coefficient, so it is not significantly different from zero. Based on this, the overall fixed effect trends seen in MN appear to be generally scalable to the rest of the data set. The standard deviation of the intercept and 'Floor' random effect coefficients are 0.364 and 0.259 when grouping at the county level. The standard deviation of the intercept remains relatively large at the state level (0.425), but standard deviation of the effect of 'Floor' drops to 0.031. The random effect of log(Uranium) is clustered only at the state level, as it is a level two predictor, and has a standard deviation of 0.256. The correlation between the random intercept and random effect of floor at the county level becomes quite negative in this model (-0.332), a contrast from what we saw just in MN. In order to provide context to the structure of this model, we take an example from the data set and find the predicted Radon level.

Table 21: Example From Overall Radon Data Set

	H_ID	State	County	B_Type	Basement	Floor	Radon	Uranium	County_State
13	35	AZ	COCHISE	0	0	1	2.9	2.34707	COCHISE_AZ

Table 22: Example Random Effects: County Level

	(Intercept)	Floor
AZ:COCHISE_AZ	-0.0159999	0.1060253

Table 23: Example Random Effects: State Level

	(Intercept)	log(Uranium)	Floor
AZ	-0.519858	-0.2397015	0.0083282

Let $i = 13$, $j = 2$, and $k = 1$. Then, plugging in the appropriate fixed and random effects, we find:

$$\widehat{\log(\text{Radon})}_{ijt} = \widehat{\beta_{0jt}} + \widehat{\beta_{1jt}} * (1) + 0.233 * (0)$$

\

$$\widehat{\beta_{0jt}} = \hat{\alpha_{00t}} + \hat{\alpha_{01t}}\log(2.35) - 0.016$$

$$\widehat{\beta_{1jt}} = \hat{\alpha_{10t}} + (0.013) * \log(2.35) + 0.106$$

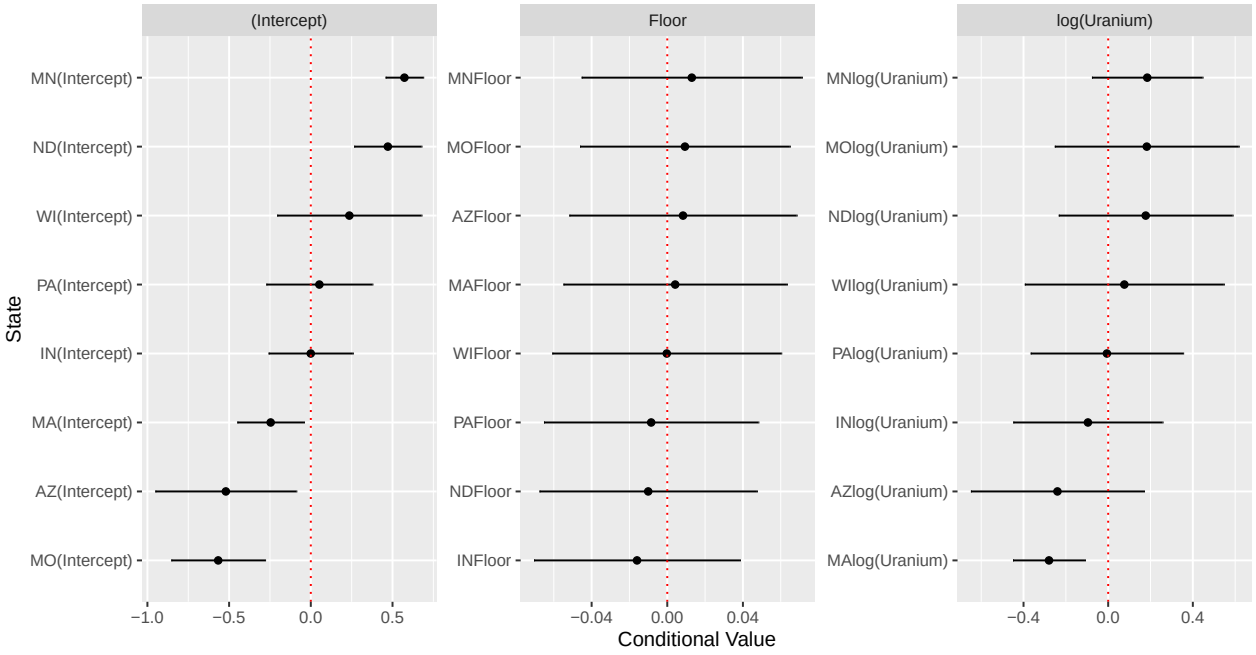
$$\hat{\alpha_{00t}} = 0.684 - 0.520$$
$$\hat{\alpha_{10t}} = -0.780 + 0.008$$
$$\hat{\alpha_{01t}} = 0.450 - 0.239$$

This implies that:

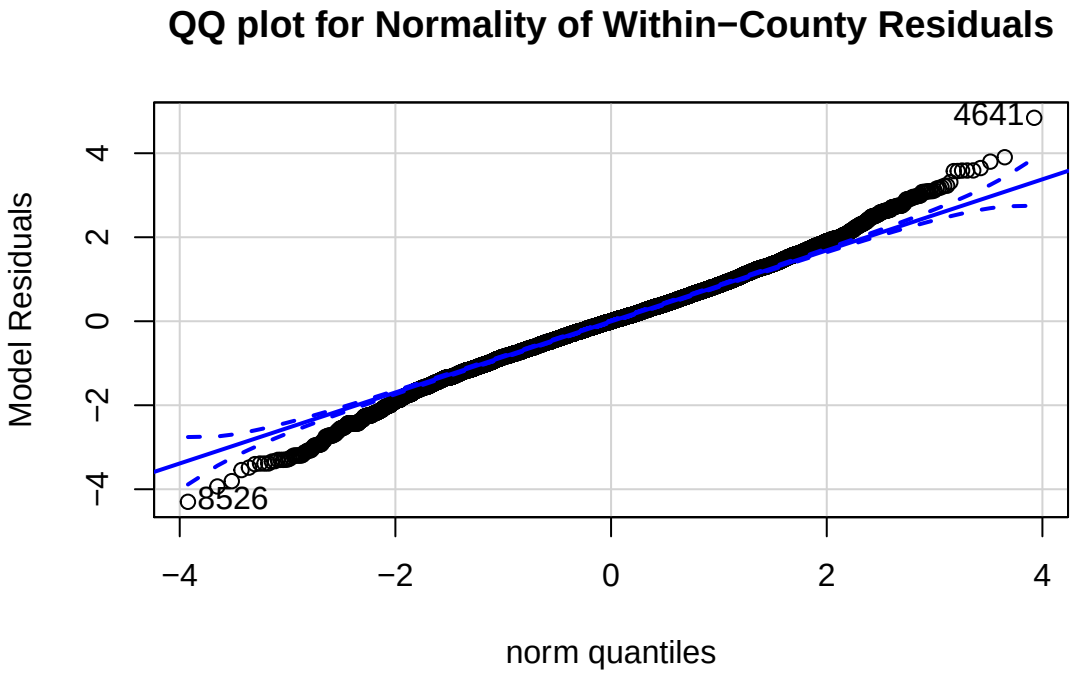
$$\widehat{\log(\text{Radon})}_{ijt} = -0.328 - 0.654 * (1) + 0.233 * (0) = -0.326$$

Finally, we have $\widehat{\log(\text{Radon})}_{ijt} = e^{-0.328} = 0.720$. This was an example of a poor prediction, since the actual Radon level is 2.9.

For a more comprehensive look at how adding clusters at the state level influenced the model, we plot the state-level random effects.



The intercept term appears to capture the most variation between states, while Floor and log(Uranium) only have one effect significantly different from zero combined. We now analyze model residuals, first by checking the overall residual normality.

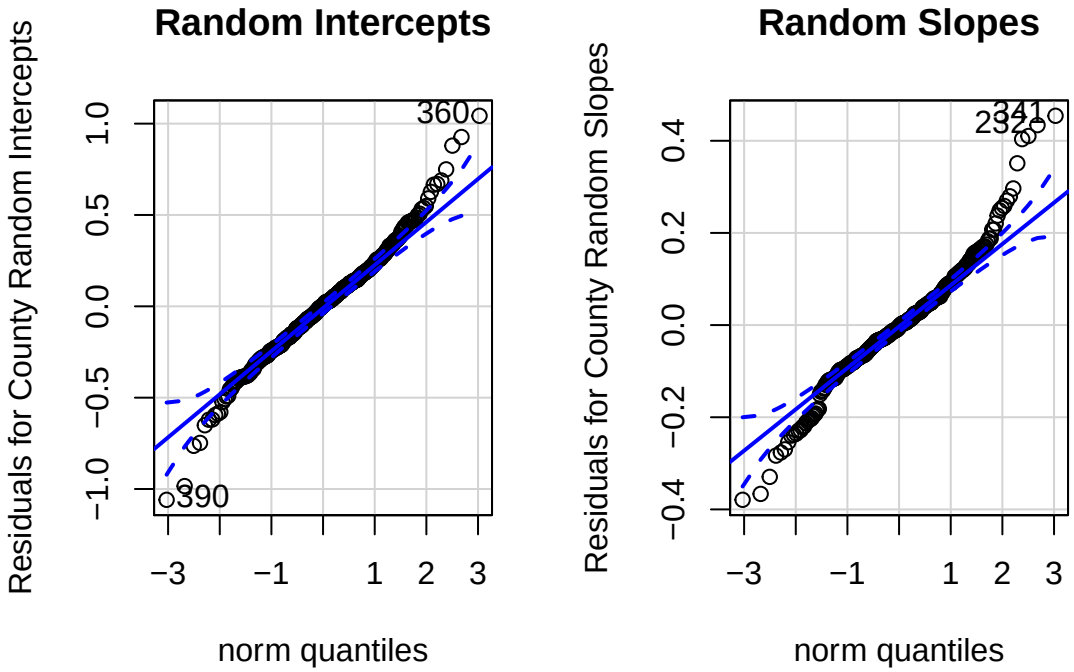


Based on the plot above, the residual distribution appears to have wide tails, indicating more extreme values. Observations 4641 and 8526 have residuals that are particularly far from normality.

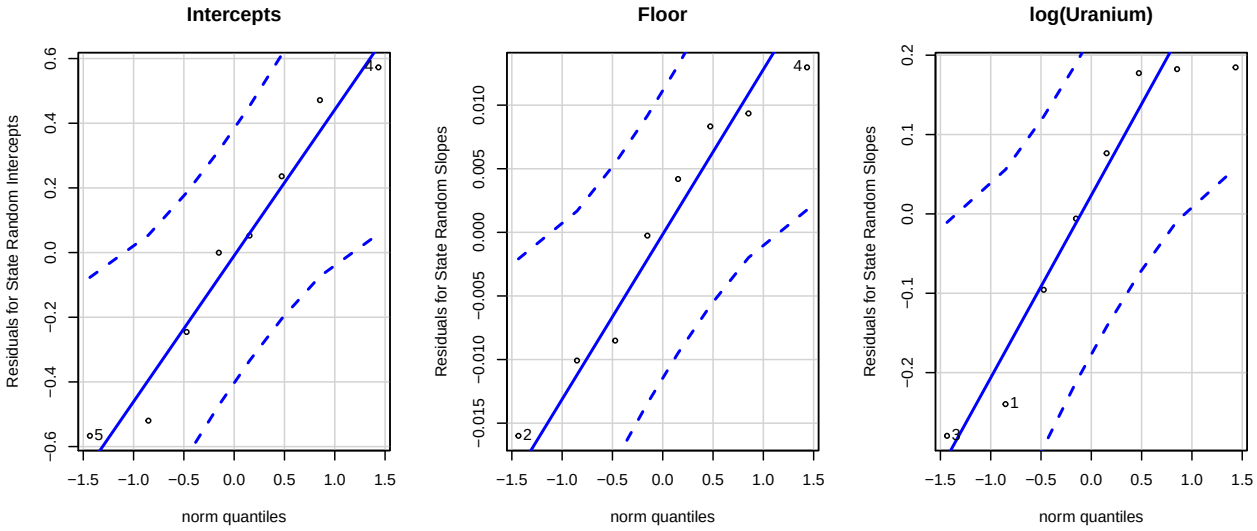
Table 24: Model Outliers

H_ID	State	County	B_Type	Basement	Floor	Radon	Uranium	County_State
4675	MA	PLYMOUTH	1	1	0	6.5	1.42956	PLYMOUTH_MA
9053	ND	RICHLAND	1	1	0	8.4	1.36463	RICHLAND_ND

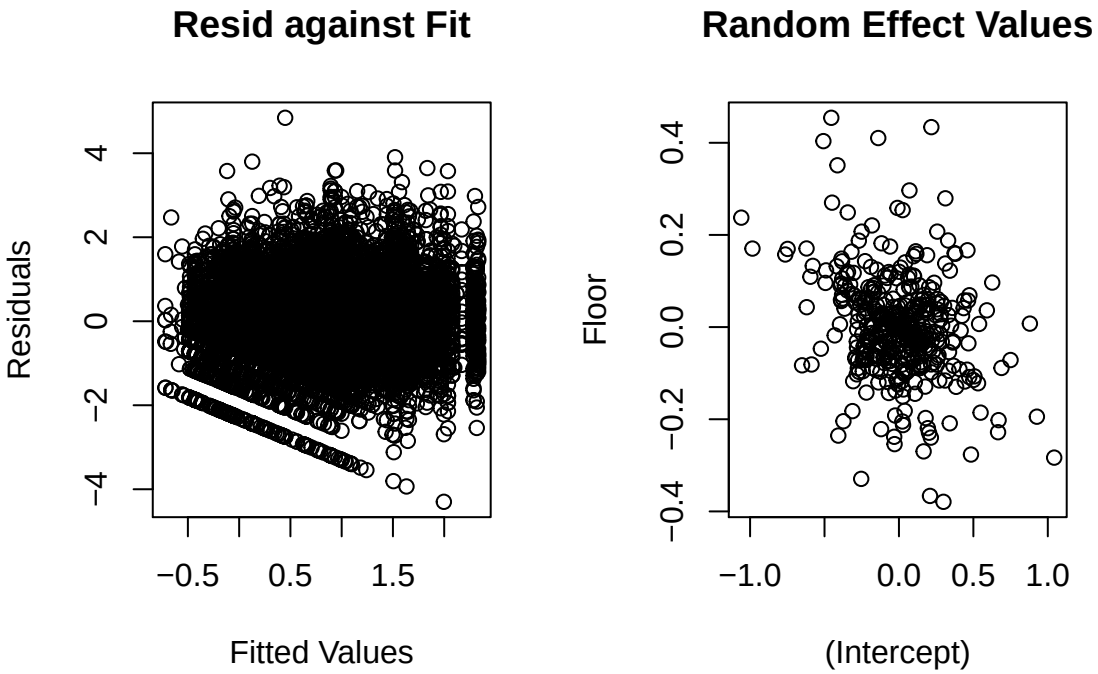
After further assessment, observation 4641 has a particularly high radon reading for that area of the country and observation 8526 has a particularly low reading for single family homes where the reading was taken in the basement in that area of the country. This is the likely cause for their labeling as outliers. We now plot between county model residuals.



Similar to the model residuals, the county random effects have wide tails. Outliers here are likely a result of extreme counties, or counties with too few data to establish an accurate regression. Next, we consider the state level random effects.



Since there are so few states in the data set, it is difficult to make an assessment on normality, there do not appear to be any egregious violations based on the eight states included in this analysis. Finally, we plot the residuals against fitted values for the overall model and the random slope vs random intercept at the county level.



The residuals against fitted values plot shows discernible patterns in the lower left of the graph, do to the high level of complexity of this model, it is difficult to expect that the pattern would have been completely random. There is a negative association between the 'Floor' random effect and the random intercept, as we would have expected based on the model output discussed earlier. Further analysis should investigate transformation of factors, or modifying the second and third levels of this model to ensure normality. Also possible state level predictors for the random effects at level two could be introduced.

Code Appendix

```
knitr::opts_chunk$set(echo = F)
knitr::opts_chunk$set(message = F)
knitr::opts_chunk$set(warning = F)
knitr::opts_chunk$set(fig.height = 4)
knitr::opts_chunk$set(fig.width = 6)
knitr::opts_chunk$set(fig.align="center")



# Read in Data and Clean it appropriately

library(tidyverse)
BigData <- read.csv("~/Documents/Brown 1/Practical Data Analysis/HW1/McAlindon_Big.csv")

`%notin%` <- Negate(`%in%`)

# Columns that long data will be created on
toExclude1 <- colnames(BigData)[colnames(BigData) %notin% c("pain.1","pain.2",
                                                            "pain.3", "pain.4","pain.5","pain.6","pain.
# Filter data so that each pain measurement has its own row by person ID
Cleaned_Data_Pre_Drop = BigData %>%
  filter(lastdt1 == WeatherDate |
   lastdt2 == WeatherDate |
   lastdt3 == WeatherDate |
   lastdt4 == WeatherDate |
   lastdt5 == WeatherDate |
   lastdt6 == WeatherDate |
   lastdt7 == WeatherDate) %>%
  gather(key = Time, value = Pain, -toExclude1) %>%
  distinct_at(vars(Time, ID, WeatherDate), .keep_all = TRUE) %>%
  mutate(Time = plyr::mapvalues(Time,
                                c("pain.1", "pain.2", "pain.3", "pain.4", "pain.5", "pain.6", "pain.
                                0:6)) %>%
#Remove Observations that include NA values for essential variables
  filter(!is.na(Pain) & !is.na(Time) & !is.na(avgtemp))%>%
  arrange(ID, Time)

#Next round of cleaning involves making sure no duplicate measurements are in the data set,
#and an NA values are removed
toExclude2 <- colnames(Cleaned_Data_Pre_Drop)[colnames(Cleaned_Data_Pre_Drop)
                                              %notin% c("lastdt1","lastdt2",
                                                        "lastdt3", "lastdt4","lastdt5","lastdt6","lastd

Cleaned_Data_Pre_Drop = Cleaned_Data_Pre_Drop %>%
  gather(key = Measure_Instace, value = Measure_Date, -toExclude2)%>%
  mutate(Measure_Instace = plyr::mapvalues(Measure_Instace,
                                c("lastdt1","lastdt2", "lastdt3", "lastdt4","lastdt5","lastdt6","las
                                0:6))%>%
  filter(Measure_Instace==Time& Measure_Date == WeatherDate)
```

```r
# summary(Cleaned_Data[complete.cases(Cleaned_Data), c("ID", "Pain", "Time",
#"avgtemp", "age", "treat"," sex", "retire", "nsaid")])

pre_removal = unique(Cleaned_Data_Pre_Drop$ID)


#drop NA observations
Cleaned_Data = Cleaned_Data_Pre_Drop%>%
  filter(!is.na(age) & !is.na(treat)& !is.na(sex)& !is.na(retire) & !is.na(nsaid))%>%
  arrange(ID, Time)%>%
  group_by(ID) %>%
  filter(n() >1) %>%
  select(ID, Pain, Time, avgtemp, age, treat, sex, retire, nsaid)

#After dropping NAs we want to determine if we lost unique information

post_removal = unique(Cleaned_Data$ID)

dropped = pre_removal[which(pre_removal %notin% post_removal)]

UniqueIDs = which(!duplicated(Cleaned_Data_Pre_Drop$ID))
UniqueRows = Cleaned_Data_Pre_Drop[UniqueIDs, c("ID", "Pain", "Time",
                                                "avgtemp", "age", "treat", "sex", "retire", "nsaid") ]
DroppedRows = UniqueRows[which(UniqueRows$ID %in% dropped),]


Dropped_v_remain = list(summary(DroppedRows[,-c(1,3)]), summary(Cleaned_Data[,-c(1,3)]))
names(Dropped_v_remain) = c("Dropped Observations", "Remaining Observations")

#Print Summary Data to determine difference between dropped and non dropped observations

print(Dropped_v_remain)

Cleaned_Data[,5] = apply(Cleaned_Data[,5], 2, scale)

#Recalibrate factors appropriately

Cleaned_Data$Pain = as.numeric(Cleaned_Data$Pain)
Cleaned_Data$Time = as.numeric(Cleaned_Data$Time)
Cleaned_Data$sex = Cleaned_Data$sex -1
Cleaned_Data$retire = Cleaned_Data$retire-1



#Show Model Output and create tables for display

library(lme4)
library(stargazer)
library(kableExtra)
# library("lmerTest")
#detach("package:lmerTest", unload=TRUE)
Random_Intercept = lmer(Pain ~ 1 +(1|ID),data=Cleaned_Data,REML=F) # random intercept model
rand_int_summary = summary(Random_Intercept)
```

```
kable_styling(kable(round(rand_int_summary$coefficients,3), caption = "Fixed Effects: Random Intercept M

Var_Random_Intercept = as.data.frame(VarCorr(Random_Intercept))
Var_Random_Intercept = Var_Random_Intercept[,c(1,2,4,5)]
colnames(Var_Random_Intercept) = c("Group", "Name", "Variance", "SD")
Var_Random_Intercept[,c(3,4)] = round(Var_Random_Intercept[,c(3,4)], 3)
Var_Random_Intercept[which(is.na(Var_Random_Intercept$Name)),"Name"] = ""

kable_styling(kable(Var_Random_Intercept, caption = "Random Effects: Random Intercept Model"),
              latex_options = "hold_position")


#Show model ANOVA when adding fixed effects

library(lme4)
Time_Only = lmer(Pain ~ 1 +(1|ID) + Time ,data=Cleaned_Data,REML=F)


Temperature_Only = lmer(Pain ~ 1 +(1|ID) + avgtemp ,data=Cleaned_Data,REML=F)


RandInt_Time = round(anova(Random_Intercept, Time_Only),3)
RandInt_Temp = round(anova(Random_Intercept, Temperature_Only),3)


kable_styling(kable(RandInt_Time, caption = "ANOVA for Level 1 Predictors"),
              latex_options = "hold_position")
kable(RandInt_Temp)

# test = lmer(Pain ~ 1 +(1 + Time |ID) + Time ,data=Cleaned_Data,REML=F)
# test2 = lmer(Pain ~ 1 +(1 + avgtemp*Time|ID) + avgtemp*Time ,data=Cleaned_Data,REML=F)

# Add Time as Random effect and view model output
Time_Random = lmer(Pain ~ 1 +(1 + Time|ID) + Time ,data=Cleaned_Data,REML=F)
RandFixed_Time = round(anova(Time_Random, Time_Only),3)
kable_styling(kable(RandFixed_Time, caption = "Comparing Time as a Fixed Effect and Random Effect"),
              latex_options = "hold_position")
#Show output for random Time model
rand_timeInt_summary = summary(Time_Random)
kable_styling(kable(round(rand_timeInt_summary$coefficients,3),
                    caption = "Fixed Effects: Random Intercept and Time Model"),
              latex_options = "hold_position")

Var_Random_Time = as.data.frame(VarCorr(Time_Random))
colnames(Var_Random_Time) = c("Group", "Variable 1", "Variable 2", "Var/Cov", "SD/Corr")
Var_Random_Time[,c(4,5)] = round(Var_Random_Time[,c(4,5)], 3)
Var_Random_Time[which(is.na(Var_Random_Time$`Variable 1`)),"Variable 1"] = ""
Var_Random_Time[which(is.na(Var_Random_Time$`Variable 2`)),"Variable 2"] = ""

kable_styling(kable(Var_Random_Time, caption = "Random Effects: Random Intercept Model"),
              latex_options = "hold_position")

#test possible model combinations
```

```
library(lme4)
Cleaned_Data = as.data.frame(Cleaned_Data)
Good_Model = lmer(Pain ~ 1 + Time + age*Time + age + (1 + Time | ID) ,data=Cleaned_Data,REML=F)
Good_i = NA
Good_j = NA

#Loop that tests ever possble addition of a patient level effect interacting with Time

for(i in ncol(Cleaned_Data):5){
  for(j in ncol(Cleaned_Data):5){
    New_Model = lmer(Pain ~ 1 + Time + eval(parse(text = colnames(Cleaned_Data)[i]))*Time +
                     eval(parse(text = colnames(Cleaned_Data)[j]))+ (1 + Time | ID) ,
                   data=Cleaned_Data,REML=F)
    check = anova(Good_Model, New_Model)
    #If model is significantly better by ANOVA, make it the "good model"
    if(rownames(check)[1] == "Good_Model" &
      check$`Pr(>Chisq)`[2] < 0.05){
      Good_Model = New_Model
      Good_i = i
      Good_j = j}
    if(rownames(check)[1] == "New_Model" &
      check$`Pr(>Chisq)`[2] >= 0.05){
      Good_Model = New_Model
      Good_i = i
      Good_j = j}
  }
}


Slope_Intercept_Fit = lmer(Pain ~ 1 + Time + nsaid*Time + nsaid + (1 + Time | ID) ,
                           data=Cleaned_Data,REML=F)

Good_Model = Slope_Intercept_Fit
Good_i = NA
Good_j = NA

#Check Interactions with NSAID and Time
  for(j in ncol(Cleaned_Data):5){
    New_Model = lmer(Pain ~ 1 + Time + eval(parse(text = colnames(Cleaned_Data)[j]))*nsaid*Time +
                       nsaid + (1 + Time | ID) ,data=Cleaned_Data,REML=F)
    check = anova(Good_Model, New_Model)
    if(rownames(check)[1] == "Good_Model" &
      check$`Pr(>Chisq)`[2] < 0.05){
      Good_Model = New_Model
      Good_i = i
      Good_j = j}
    if(rownames(check)[1] == "New_Model" &
      check$`Pr(>Chisq)`[2] >= 0.05){
      Good_Model = New_Model
      Good_i = i
      Good_j = j}
  }
```

```
Good_Model = Slope_Intercept_Fit
Good_i = NA
Good_j = NA

#Check interacton with NSAID alone
  for(j in ncol(Cleaned_Data):5){
    New_Model = lmer(Pain ~ 1 + Time + nsaid*Time + eval(parse(text = colnames(Cleaned_Data)[j]))*nsaid
                     (1 + Time | ID) ,data=Cleaned_Data,REML=F)
    check = anova(Good_Model, New_Model)
    if(rownames(check)[1] == "Good_Model" &
      check$`Pr(>Chisq)`[2] < 0.05){
      Good_Model = New_Model
      Good_j = j}
    if(rownames(check)[1] == "New_Model" &
      check$`Pr(>Chisq)`[2] >= 0.05){
      Good_Model = New_Model
      Good_j = j}
  }


#Check just adding a variable to the second level
Good_Model = Slope_Intercept_Fit
Good_i = NA
Good_j = NA

  for(j in ncol(Cleaned_Data):5){
    New_Model = lmer(Pain ~ 1 + Time + nsaid*Time + eval(parse(text = colnames(Cleaned_Data)[j]))+ nsai
                     (1 + Time | ID) ,data=Cleaned_Data,REML=F)
    check = anova(Good_Model, New_Model)
    if(rownames(check)[1] == "Good_Model" &
      check$`Pr(>Chisq)`[2] < 0.05){
      Good_Model = New_Model
      Good_j = j}
    if(rownames(check)[1] == "New_Model" &
      check$`Pr(>Chisq)`[2] >= 0.05){
      Good_Model = New_Model
      Good_j = j}
  }


#Check adding a second variable to interact with time
Good_Model = Slope_Intercept_Fit
Good_i = NA
Good_j = NA

  for(j in ncol(Cleaned_Data):5){
    New_Model = lmer(Pain ~ 1 + Time + nsaid*Time + eval(parse(text = colnames(Cleaned_Data)[j]))*Time+
                     nsaid + (1 + Time | ID) ,data=Cleaned_Data,REML=F)
    check = anova(Good_Model, New_Model)
    if(rownames(check)[1] == "Good_Model" &
      check$`Pr(>Chisq)`[2] < 0.05){
      Good_Model = New_Model
      Good_j = j}
```

```
    if(rownames(check)[1] == "New_Model" &
      check$`Pr(>Chisq)`[2] >= 0.05){
      Good_Model = New_Model
      Good_j = j}
  }



check = summary(Slope_Intercept_Fit)


# Show final model summary
kable_styling(kable(round(check$coefficients,3), caption = "Fixed Effects: Final Model"),
              latex_options = "hold_position")


Var_Random_NSAID = as.data.frame(VarCorr(Good_Model))
colnames(Var_Random_NSAID) = c("Group", "Variable 1", "Variable 2", "Var/Cov", "SD/Corr")
Var_Random_NSAID[,c(4,5)] = round(Var_Random_NSAID[,c(4,5)], 3)
Var_Random_NSAID[which(is.na(Var_Random_NSAID$`Variable 1`)),"Variable 1"] = ""
Var_Random_NSAID[which(is.na(Var_Random_NSAID$`Variable 2`)),"Variable 2"] = ""

kable_styling(kable(Var_Random_NSAID, caption = "Random Effects: Final Model"), latex_options = "hold_p
#Plot example of patient regressions

Fitted = Slope_Intercept_Fit@frame
fitted = fitted(Slope_Intercept_Fit)
Full_Time = Cleaned_Data$Time
Use_NSAID = rep("Yes" , nrow(Fitted))
Use_NSAID[which(Fitted$nsaid == 0)] = "No"
Fitted = cbind(Fitted,Use_NSAID, fitted, Full_Time)
Fitted$Use_NSAID = factor(Fitted$Use_NSAID)
Sample_Fitted = Fitted[1:143,]
ggplot(data =Sample_Fitted, aes(x =Time, y = fitted, group = ID, color = Use_NSAID))+
  geom_line() +geom_point() + xlab("Standardized Time") + ylab("Pain") +
  ggtitle("Sample of Standardized Fitted Values for Pain over Time by Patient") +
  theme_minimal()

#Plot Random intercept and time coefficient by patient
ranef_NSAID = ranef(Good_Model,condVar=T)
ranef.data = as.data.frame(ranef_NSAID)


reorder_within <- function(x, by, within, fun = mean, sep = "", ...) {
  new_x <- paste(x, within, sep = sep)
  stats::reorder(new_x, by, FUN = fun)
}


ggplot(ranef.data,aes(x=condval,y=reorder_within(grp, condval, term, identity))) + geom_point()+facet_w
  geom_errorbarh(aes(xmin=condval-2*condsd,xmax=condval+2*condsd),height=0) +ylab("Patient ID")+
  xlab("Conditional Value") +geom_vline(xintercept = 0, linetype = "dotted", color = "red")
```

```r
#Check residuals

#Do regression diagnostics
car::qqPlot(resid(Good_Model), ylab = "Model Residuals", main = "QQ plot for Normality of Within-Person
kable_styling(kable(Cleaned_Data[c(281,426),], caption = "Model Outliers"),
              latex_options = "hold_position")

resid.int = ranef(Good_Model)[[1]][,1]  #Residuals for random intercepts (between-person)
resid.time = ranef(Good_Model)[[1]][,2]  #Residuals for random intercepts (between-person)

opar = par(no.readonly = T)
par(mfrow=c(1,2))
car::qqPlot(resid.time,main="Random Time", ylab ="Residuals for Random Slopes" )
car::qqPlot(resid.int,main="Random Intercepts", ylab ="Residuals for Random Intercepts")
par(opar)
#Fitted vs. Residual Plots at within and between-person levels
par(mfrow=c(1,2))
plot(fitted(Good_Model),resid(Good_Model),xlab = "Fitted Values",ylab="Residuals",
     main = "Resid against Fit")
plot(ranef(Good_Model)[[1]], main = "Random Effect Values")
par(opar)



## PROBLEM 2

#Read in Data On Uranium Levels by County and Radon levels by house

Ulevels = read.csv("/Users/Anthony/Documents/Brown 1/Practical Data Analysis/HW3/cty.csv")
colnames(Ulevels) = c("S_FIPS", "C_FIPS", "State", "County", "Uranium")

HouseRadon = read.csv("/Users/Anthony/Documents/Brown 1/Practical Data Analysis/HW3/srrs2.csv")
colnames(HouseRadon) = c("H_ID", "State", "S_FIPS", "B_Type", "Floor", "Basement", "Radon",
                         "C_FIPS", "County" )


HouseRadon$County = as.character(HouseRadon$County)
Ulevels$County = as.character(Ulevels$County)

#Recode data so that We have 1s nd Os and interpetable designations
HouseRadon$Basement = as.character(HouseRadon$Basement)
HouseRadon$Basement[which(HouseRadon$Basement == " " |HouseRadon$Basement == "0")] = NA
HouseRadon$Basement[which(HouseRadon$Basement == "N")] = 0
HouseRadon$Basement[which(HouseRadon$Basement == "Y")] = 1

#Group Floor Variable so that measurments are either taken in the basement or not
HouseRadon$Floor = as.numeric(HouseRadon$Floor)
HouseRadon$Floor[which(HouseRadon$Floor == 9)] = NA
HouseRadon$Floor[which(HouseRadon$Floor == 1 | HouseRadon$Floor == 2|HouseRadon$Floor == 3 )] =1


HouseRadon$B_Type[which(HouseRadon$B_Type != 1)] = 0
```

```r
#89 obs
MNUlevels = Ulevels %>% filter(State == 'MN')
#1188 obs
MNHouseRadon = HouseRadon %>% filter(State == 'MN')
length(unique(MNHouseRadon$County))
# 1145 obs
MNHouseRadon = MNHouseRadon[complete.cases(MNHouseRadon$Basement),]
length(unique(MNHouseRadon$County))

library(stringr)

#Get rid of spaces in county names to match Uranium data set
for(i in 1:nrow(MNHouseRadon)){
  MNHouseRadon$County[i] = str_replace_all(string= MNHouseRadon$County[i], pattern=" ", repl="")
}

#950 obs
MNRadonData = merge(MNHouseRadon, MNUlevels[,c("County", "Uranium")], by = "County")

#Adjest Radon and Uranium dat so we can take the log
MNRadonData = MNRadonData[,c("H_ID", "State", "County", "B_Type",
                             "Basement", "Floor", "Radon", "Uranium")]
MNRadonData$Radon[which(MNRadonData$Radon == 0)] = 0.1
MNRadonData$Uranium[which(MNRadonData$Uranium == 0)] = 0.1


MNRadonData$County = factor(MNRadonData$County)
MNRadonData$Basement = as.numeric(MNRadonData$Basement)
kable_styling(kable(as.data.frame(cor(MNRadonData[,4:8]))), latex_options = "hold_position")

#Plot differences between levels of radon when looking at house factors

library(ggplot2)
p1 = ggplot(data = MNRadonData, aes(group = Floor,x =Floor, y = log(Radon)))+ geom_boxplot()
p2= ggplot(data = MNRadonData, aes(group = B_Type,x =B_Type, y = log(Radon)))+ geom_boxplot()
p3= ggplot(data = MNRadonData, aes(group = Basement,x =Basement, y = log(Radon)))+ geom_boxplot()

gridExtra::grid.arrange(p1,p2,p3, nrow=1)
#determine whether to use basement of bulding typoe
basefit = lm(log(Radon) ~ -1 + Basement, data = MNRadonData)
typefit = lm(log(Radon) ~ -1 + B_Type, data = MNRadonData)

#Regression output
library(stargazer)

stargazer(basefit, typefit, type = 'latex', header = F,
          title = "Regresson of Radon at House Level with No Slope")

#build and test base models with different predictors

Rand_Floor_Int =lmer(log(Radon) ~ 1 + Floor+ (1 + Floor | County), data = MNRadonData, REML = F)
```

```
Rand_BType =lmer(log(Radon) ~ -1 + B_Type+ (1 + B_Type | County), data = MNRadonData, REML = F)

Floor_Rand_Int =lmer(log(Radon) ~ 1 + Floor +(1|County), data = MNRadonData, REML = F)

BType_Rand_Int =lmer(log(Radon) ~ -1 + B_Type +(1|County), data = MNRadonData, REML = F)

Initial_ANOVA = anova(Rand_Floor_Int, Rand_BType, Floor_Rand_Int, BType_Rand_Int)
#Compare base models via ANOVA

kable_styling(kable(Initial_ANOVA, caption = "Initial Model Screening") ,
              latex_options = "hold_position")

#More model testing with B_Type and Floor as random and Fixed effects
#Uranium a second level predictor

Base_Model = lmer(log(Radon) ~ 1 + Floor+ (1| County), data = MNRadonData, REML = F)

Floor_Rand_BType = lmer(log(Radon) ~ -1 + B_Type + Floor + (1 + B_Type| County),
                        data = MNRadonData, REML = F)

Floor_BType = lmer(log(Radon) ~ -1 + B_Type + Floor + (1| County), data = MNRadonData, REML = F)


Floor_Rand_UBType= lmer(log(Radon) ~ -1 +  log(Uranium)*B_Type +  Floor + (1 + B_Type| County),
                        data = MNRadonData, REML = F)

Rand_UFloor = lmer(log(Radon) ~ 1 +  log(Uranium)*Floor  + (1 + Floor| County),
                   data = MNRadonData, REML = F)

BType_Rand_UFloor = lmer(log(Radon) ~ 1 + B_Type+ log(Uranium)*Floor  + (1 + Floor| County),
                         data = MNRadonData, REML = F)

Second_Screening = anova(Base_Model,Floor_Rand_BType,Floor_BType,
                         Floor_Rand_UBType,Rand_UFloor,BType_Rand_UFloor)
#Anova of all models
kable_styling(kable(Second_Screening,caption = "Second Model Screening"),
              latex_options = "hold_position")


#Anove for final comparison

Third_Screening = anova(Floor_Rand_UBType, BType_Rand_UFloor)
kable_styling(kable(Third_Screening, caption = 'Third Model Screening'),
              latex_options = "hold_position")
#Show final model output

final_sum = summary(BType_Rand_UFloor)
kable_styling(kable(round(final_sum$coefficients,3), caption = "Fixed Effects: Final Model"),
              latex_options = "hold_position")


Var_Random_Uranium = as.data.frame(VarCorr(BType_Rand_UFloor))
colnames(Var_Random_Uranium) = c("Group", "Variable 1", "Variable 2", "Var/Cov", "SD/Corr")
```

```
Var_Random_Uranium[,c(4,5)] = round(Var_Random_Uranium[,c(4,5)], 3)
Var_Random_Uranium[which(is.na(Var_Random_Uranium$`Variable 1`)),"Variable 1"] = ""
Var_Random_Uranium[which(is.na(Var_Random_Uranium$`Variable 2`)),"Variable 2"] = ""

kable_styling(kable(Var_Random_Uranium, caption = "Random Effects: Final Model"), latex_options = "hold

#Show example case for model comparison
kable_styling(kable(as.data.frame(MNRadonData[1,]), caption = "Example Case Factors"),
              latex_options = "hold_position")

kable_styling(kable(as.data.frame(ranef(BType_Rand_UFloor)$County[1,]),
                    caption = "Example Case Random Components"), latex_options = "hold_position")


#Plot model random effects

ranef_rad = ranef(BType_Rand_UFloor,condVar=T)
ranef.data = as.data.frame(ranef_rad)


reorder_within <- function(x, by, within, fun = mean, sep = "", ...) {
  new_x <- paste(x, within, sep = sep)
  stats::reorder(new_x, by, FUN = fun)
}


ggplot(ranef.data,aes(x=condval,y=reorder_within(grp, condval, term, identity))) + geom_point()+facet_w
  geom_errorbarh(aes(xmin=condval-2*condsd,xmax=condval+2*condsd),height=0) +ylab("County")+
  xlab("Conditional Value") +geom_vline(xintercept = 0, linetype = "dotted", color = "red")



#Do regression diagnostics
car::qqPlot(resid(BType_Rand_UFloor), ylab = "Model Residuals",
            main = "QQ plot for Normality of Within-County Residuals")
#QQ plot for normality of within-personresiduals for
kable_styling(kable(MNRadonData[c(228,104),], caption = "Model Outliers"),
              latex_options = "hold_position")

resid.int = ranef(BType_Rand_UFloor)[[1]][,1]  #Residuals for random intercepts (between-person)
resid.floor = ranef(BType_Rand_UFloor)[[1]][,2]  #Residuals for random intercepts (between-person)

opar = par(no.readonly = T)
par(mfrow=c(1,2))
car::qqPlot(resid.int,main="Random Intercepts", ylab ="Residuals for Random Intercepts")
car::qqPlot(resid.floor,main="Random Slopes", ylab ="Residuals for Random Slopes")
par(opar)

#Fitted vs. Residual Plots at within and between-person levels
par(mfrow=c(1,2))
plot(fitted(BType_Rand_UFloor),resid(BType_Rand_UFloor),xlab = "Fitted Values",
     ylab="Residuals", main = "Resid against Fit")
```

```r
plot(ranef(BType_Rand_UFloor)[[1]], main = "Random Effect Values")
par(opar)


# plot(fitted, scale(resid(Slope_Intercept_Fit)), xlab = "Fitted Values", ylab = "Scaled Residuals",
#      main = "Residual Plot")


## PROBLEM 3

#Take complete cases
# Observations: 12777 -> 12006
HouseRadon = HouseRadon[complete.cases(HouseRadon),]

library(stringr)

for(i in 1:nrow(HouseRadon)){
  HouseRadon$County[i] = str_replace_all(string= HouseRadon$County[i], pattern=" ", repl="")
}


#Merge data on county and state to obtain uranium levels for counties
#11538
RadonData = merge(HouseRadon, Ulevels[,c("State", "County", "Uranium")], by = c("State","County"))

RadonData = RadonData[,c("H_ID", "State", "County", "B_Type", "Basement", "Floor", "Radon", "Uranium")]

RadonData$Radon[which(RadonData$Radon == 0)] = 0.1
RadonData$Uranium[which(RadonData$Uranium == 0)] = 0.1
RadonData$Basement = as.numeric(RadonData$Basement)

# RadonData$Radon = log(RadonData$Radon)
# RadonData$Uranium = log(RadonData$Uranium)
RadonData$State = as.character(RadonData$State)


#11538 -> #11523 (One county in MI)
RadonData = RadonData[which(RadonData$State != "MI"),]

#Merge county and state names as one variable to ensure all unique combos are recognized
RadonData$County_State = as.character(paste(RadonData$County,RadonData$State, sep = '_'))




N_Counties = plyr::count(RadonData["County_State"])
N_Counties$x = as.character(N_Counties$x)


#34 Counties 55 cases
Small_Counties = N_Counties[which(N_Counties$freq <3), 1]
```

```r
#11523 -> 11468
RadonData = RadonData[-which(RadonData$County_State %in% Small_Counties),]

# plyr::count(RadonData[,"State"])

# summary(RadonData)

# Extend model from previous model to whole data set, but at 3 levels

State_Level1 =lmer(log(Radon)~1+B_Type+log(Uranium)*Floor+(1+Floor|State:County_State)+
                     (log(Uranium)+Floor||State), data= RadonData, REML = F)

summary(State_Level1)
#show model output

State_sum = summary(State_Level1)
kable_styling(kable(round(State_sum$coefficients,3), caption = "Fixed Effects: Three-Level Model"),
              latex_options = "hold_position")


Var_Random_State = as.data.frame(VarCorr(State_Level1))
colnames(Var_Random_State) = c("Group", "Variable 1", "Variable 2", "Var/Cov", "SD/Corr")
Var_Random_State[,c(4,5)] = round(Var_Random_State[,c(4,5)], 3)
Var_Random_State[which(is.na(Var_Random_State$`Variable 1`)),"Variable 1"] = ""
Var_Random_State[which(is.na(Var_Random_State$`Variable 2`)),"Variable 2"] = ""

kable_styling(kable(Var_Random_State, caption = "Random Effects: Three-Level Model"),
              latex_options = "HOLD_position")



#Example from model

kable_styling(kable(RadonData[13,], caption = "Example From Overall Radon Data Set"),
              latex_options = "HOLD_position")

kable_styling(kable(ranef(State_Level1)$`State:County_State`[2,],
                    caption = "Example Random Effects: County Level"), latex_options = "HOLD_position")


kable_styling(kable(ranef(State_Level1)$State[1,], caption = "Example Random Effects: State Level"),
              latex_options = "HOLD_position")
#Plot Random effects State

ranef_state = ranef(State_Level1,condVar=T)
ranef.data = as.data.frame(ranef_state)
ranef.data = ranef.data[which(ranef.data$grpvar == "State"),]

reorder_within <- function(x, by, within, fun = mean, sep = "", ...) {
  new_x <- paste(x, within, sep = sep)
  stats::reorder(new_x, by, FUN = fun)
}
```

```r
ggplot(ranef.data,aes(x=condval,y=reorder_within(grp, condval, term, identity))) + geom_point()+facet_wr
  geom_errorbarh(aes(xmin=condval-2*condsd,xmax=condval+2*condsd),height=0) +ylab("State")+
  xlab("Conditional Value") +geom_vline(xintercept = 0, linetype = "dotted", color = "red")



#Do regression diagnostics
car::qqPlot(resid(State_Level1), ylab = "Model Residuals",
            main = "QQ plot for Normality of Within-County Residuals")
#QQ plot for normality of within-personresiduals for
kable_styling(kable(RadonData[c(4641,8526),], caption = "Model Outliers" ,row.names = F),
              latex_options = "hold_position")


resid.int = ranef(State_Level1)[[1]][,1]  #Residuals for random intercepts (between-person)
resid.floor = ranef(State_Level1)[[1]][,2]  #Residuals for random intercepts (between-person)

opar = par(no.readonly = T)
par(mfrow=c(1,2))
car::qqPlot(resid.int,main="Random Intercepts", ylab ="Residuals for County Random Intercepts")
car::qqPlot(resid.floor,main="Random Slopes", ylab ="Residuals for County Random Slopes")
par(opar)

resid.int = ranef(State_Level1)[[2]][,1]  #Residuals for random intercepts (between-person)
resid.floor = ranef(State_Level1)[[2]][,3]#Residuals for random intercepts (between-person)
resid.ur = ranef(State_Level1)[[2]][,2]

opar = par(no.readonly = T)
par(mfrow=c(1,3))
car::qqPlot(resid.int,main="Intercepts", ylab ="Residuals for State Random Intercepts")
car::qqPlot(resid.floor,main="Floor", ylab ="Residuals for State Random Slopes")
car::qqPlot(resid.ur,main="log(Uranium)", ylab ="Residuals for State Random Slopes")
par(opar)

par(mfrow=c(1,2))
plot(fitted(State_Level1),resid(State_Level1),xlab = "Fitted Values",ylab="Residuals",
     main = "Resid against Fit")
plot(ranef(State_Level1)[[1]], main = "Random Effect Values")
par(opar)
```