Homework 3

Yue Peng

Problem 1

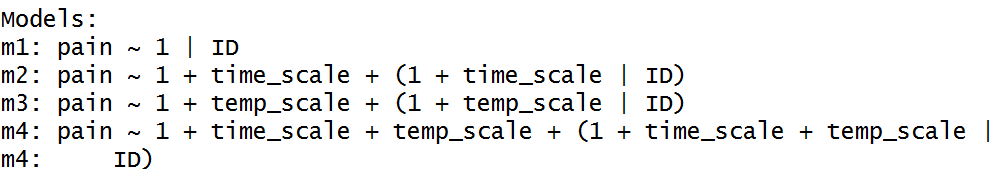
where,

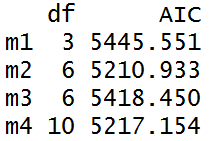
where,

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

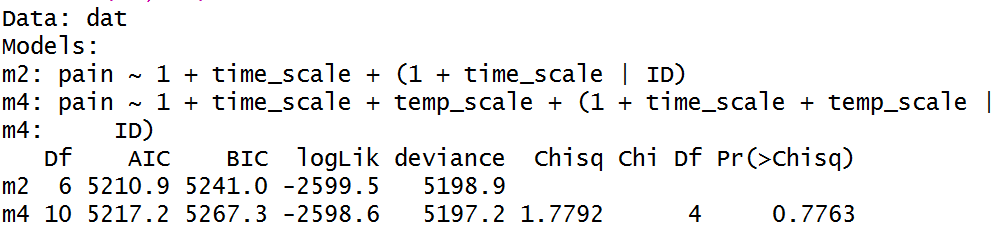
Hint: Occupation variable has been treated as “Retired” and “Not retired”.

First, we consider random effects.





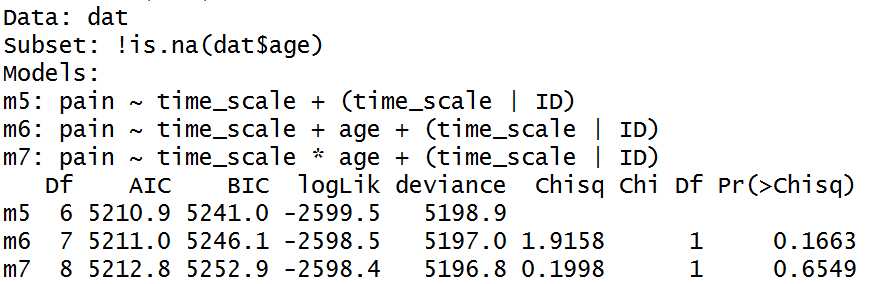
According to AIC table, we found out the model m2 with “time” and model m4 with “time” and “temperature” have relatively low AIC. Then we did a ANOVA test with these two models.



The p-value is larger than 0.05 which means that the extra term “temperature” can be not included in the model. Finally, we only kept variable “time” for further analysis.

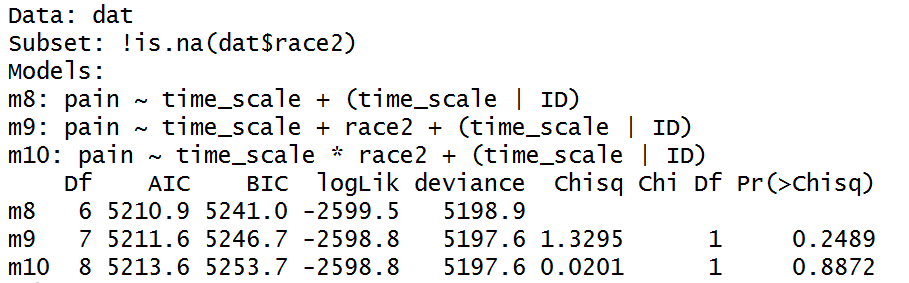
The next step was finding out the fixed effects.

1. Age



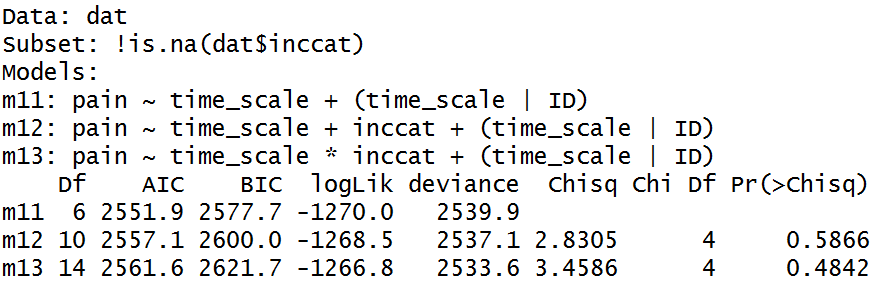
According to the ANOVA table, the “age” variable is not significant since all two p-values are larger than 0.05.

1. Race



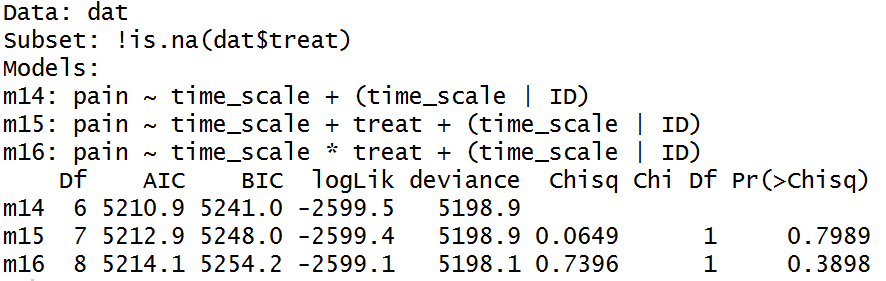
According to the ANOVA table, the “race2” variable is not significant since all two p-values are larger than 0.05.

1. Income



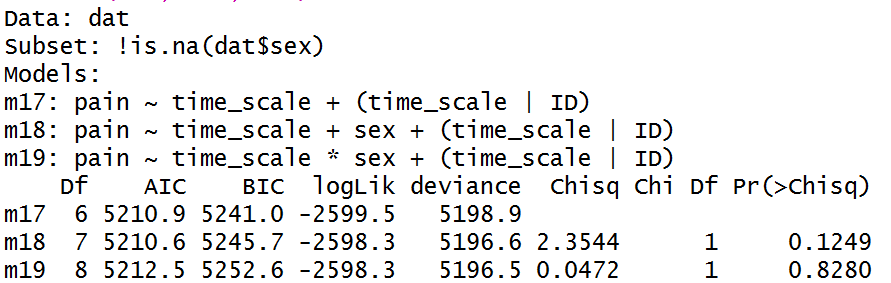
According to the ANOVA table, the “inccat” variable is not significant since all two p-values are larger than 0.05.

1. Treatment



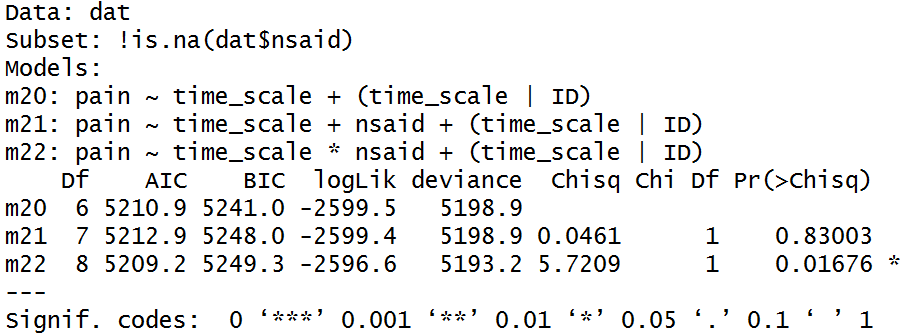
According to the ANOVA table, the “treat” variable is not significant since all two p-values are larger than 0.05.

1. Sex



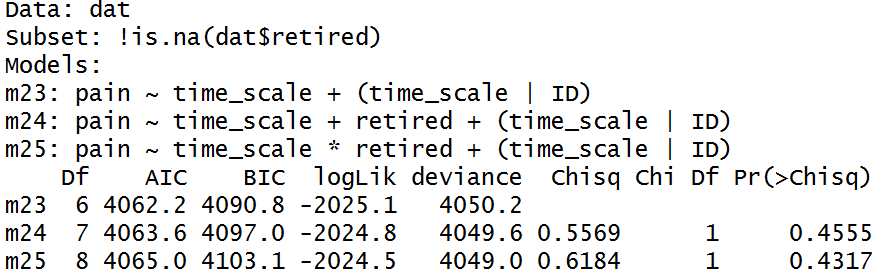
According to the ANOVA table, the “sex” variable is not significant since all two p-values are larger than 0.05.

1. Use of nsaid



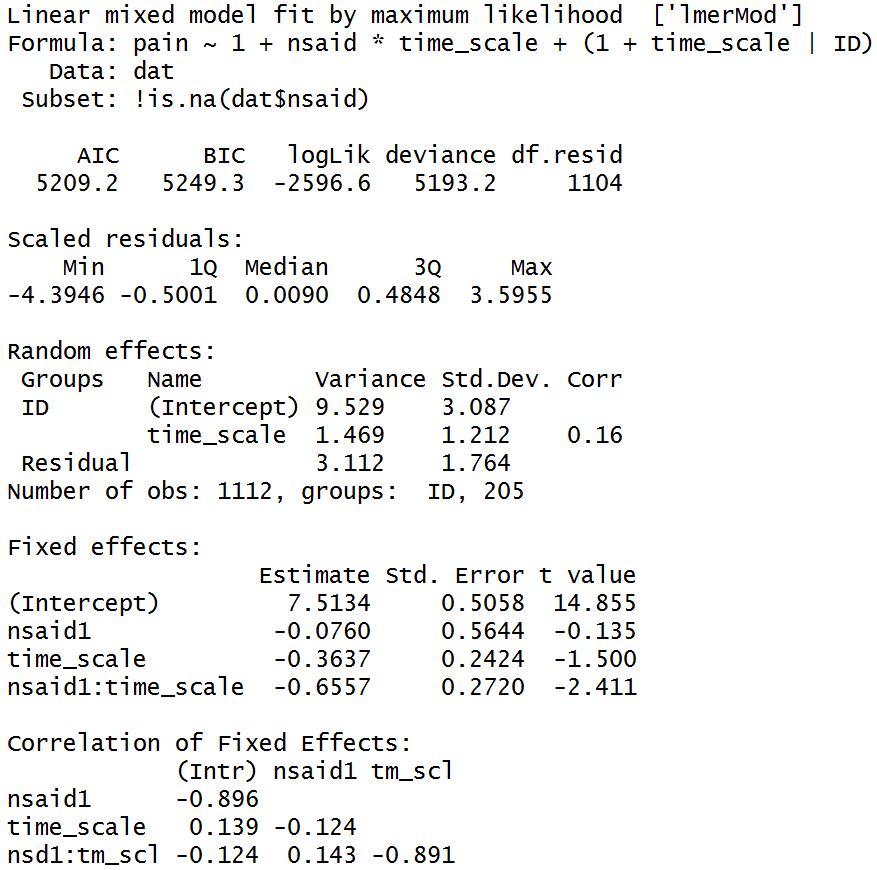
According to the ANOVA table, the “nsaid” variable in interaction model is significant since p-value is less than 0.05.

1. Occupation

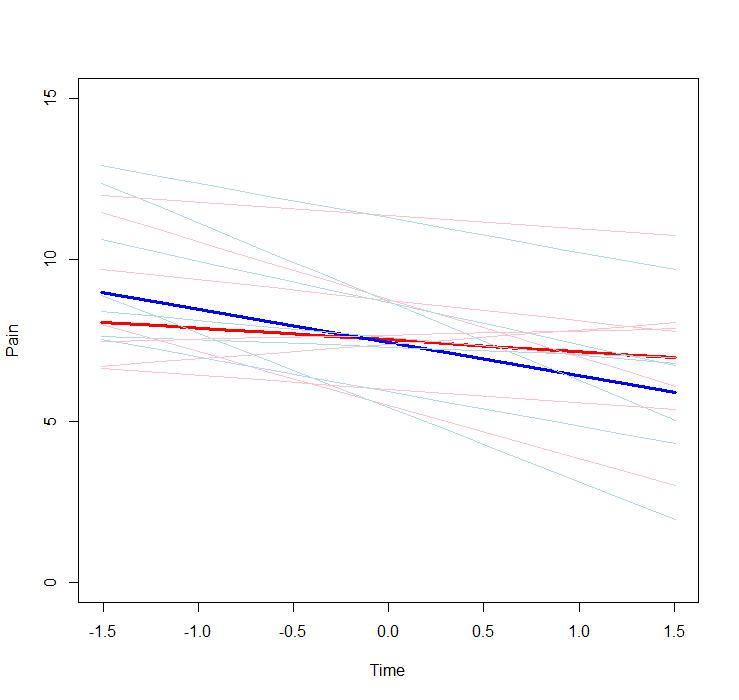


According to the ANOVA table, the “retired” variable is not significant since all two p-values are larger than 0.05.

Thus, the final model is with “time” as random effects and “time”, “nsaid” and “time\*nsaid” as fixed effects.



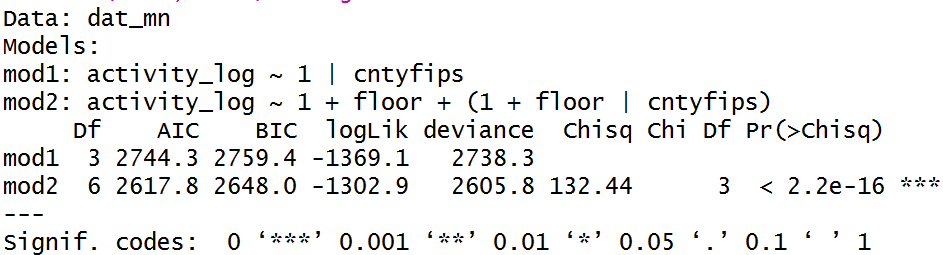
And,

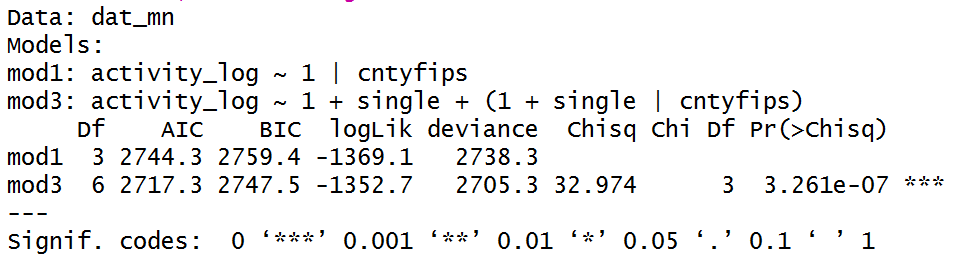


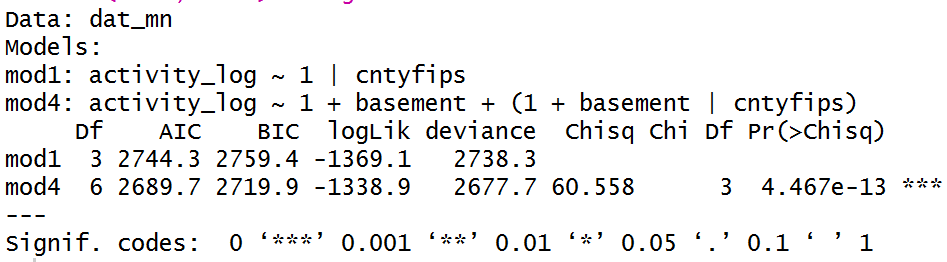
If we did not consider the Use of nsaid, the average pain decreases with time like the red line. If we did consider the Use of nsaid, the average pain decreases with time like the blue line. The light blue and pink lines show the trend of first 7 patients. And using nsaid can lead to a stage that the pain decreases in a fast speed.

Problem 2

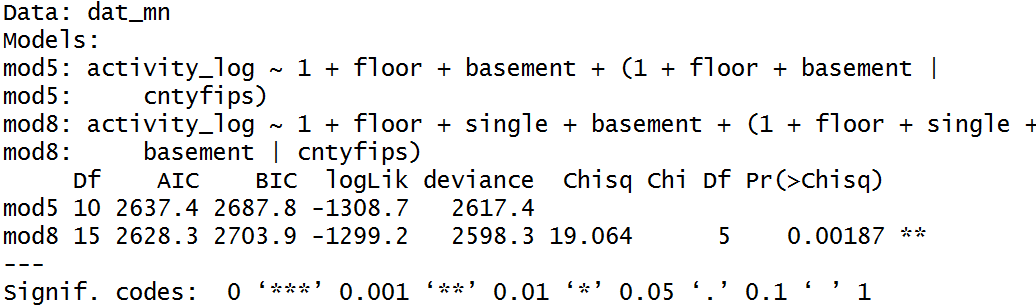
First, we consider random effects.

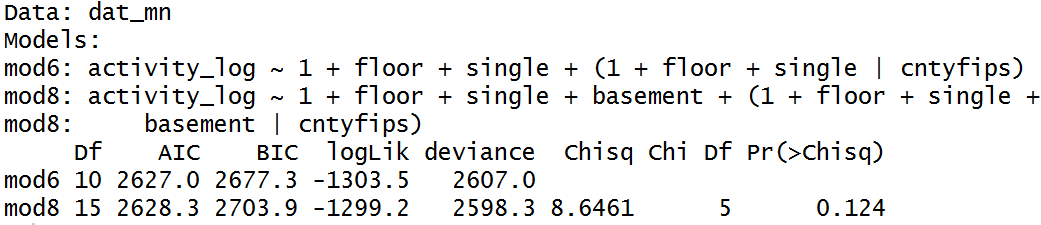


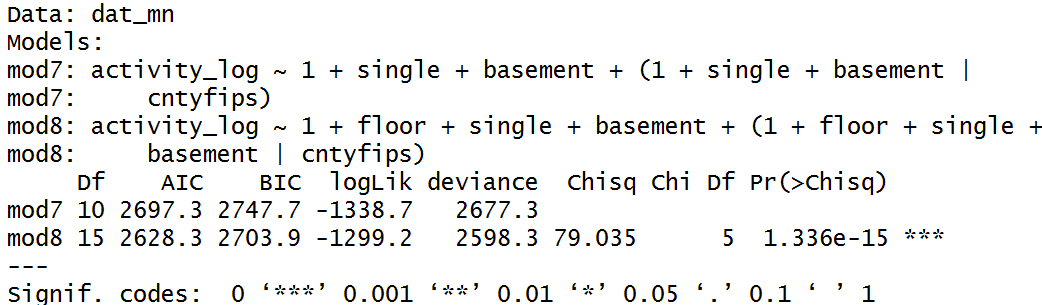




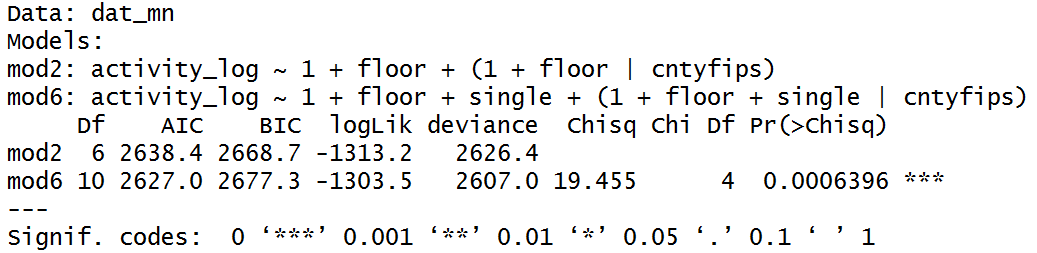
According to ANOVA table, we found out these three models with only “floor”, “single” or “basement” were all significant.

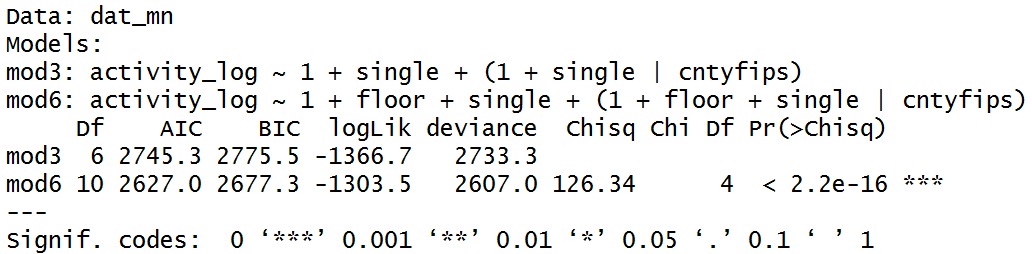






Then we tested the full model with three “drop one variable” models. The model with “floor” and “single” has the smallest AIC among all. Also, the ANOVA table showed the full model is unnecessary used in this case since the p-value is larger than 0.05 which means that the extra term “basement” can be not included in the model.

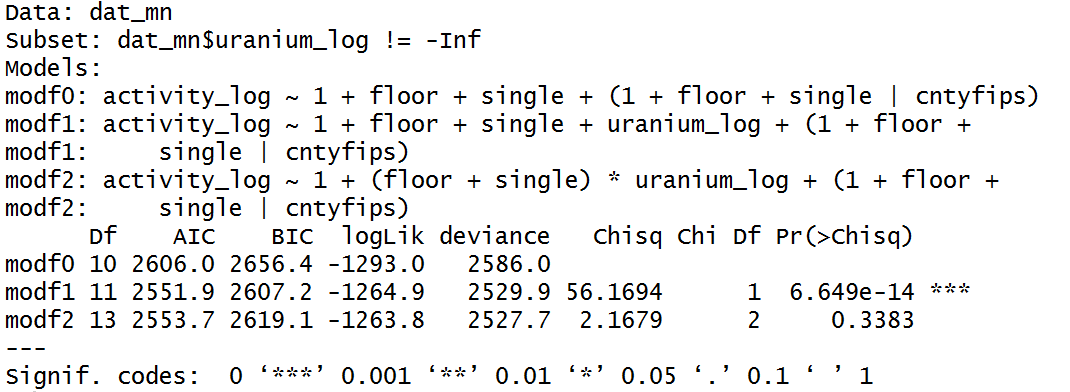




Looking back to the one-variable model, the model with both “single” and “floor” still got the smallest AIC and survived the ANOVA test.

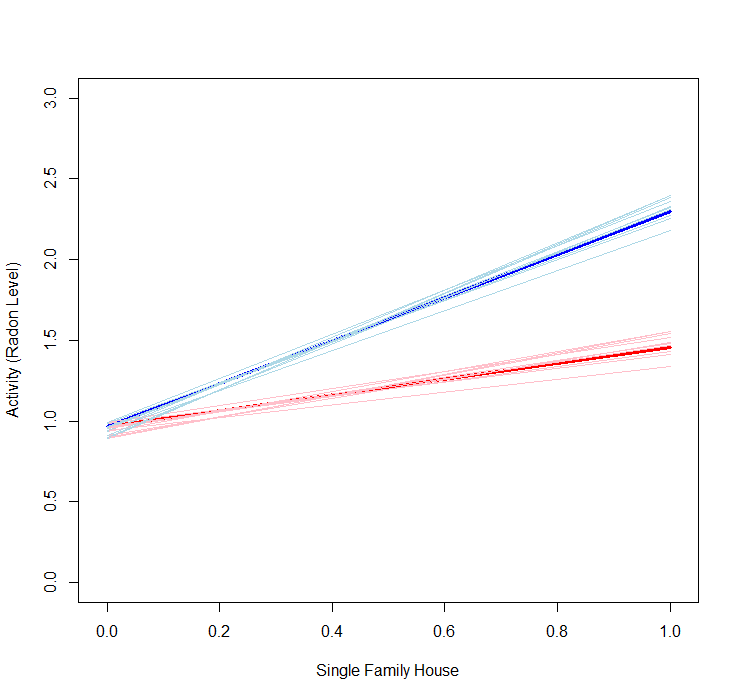
Finally, we only kept variable “floor” and “single” for further analysis.

The next step was finding out the fixed effects.



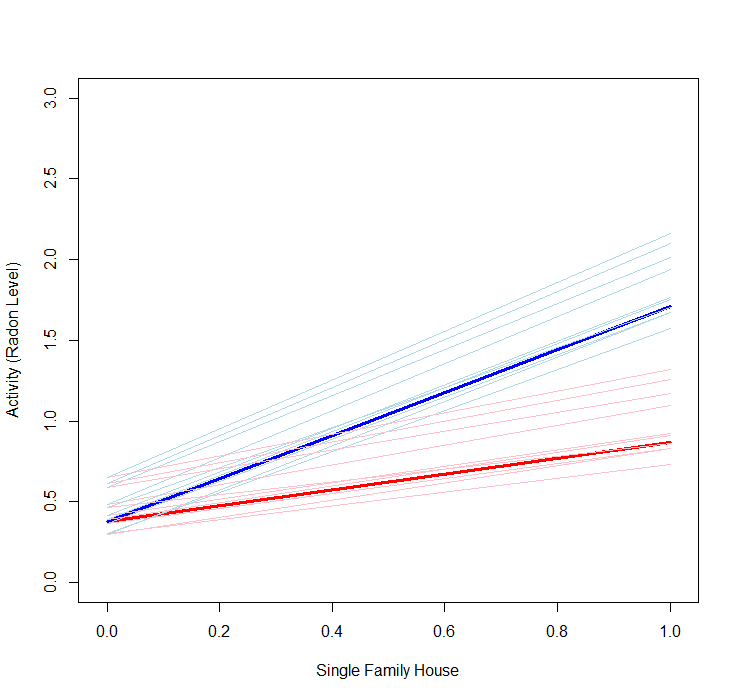
According to the ANOVA table, the “uranium” variable is significant in the model without interaction term since the p-value of it was less than 0.05.

Thus, the final model is with “floor” and “single” as random effects and “uranium” as fixed effects.



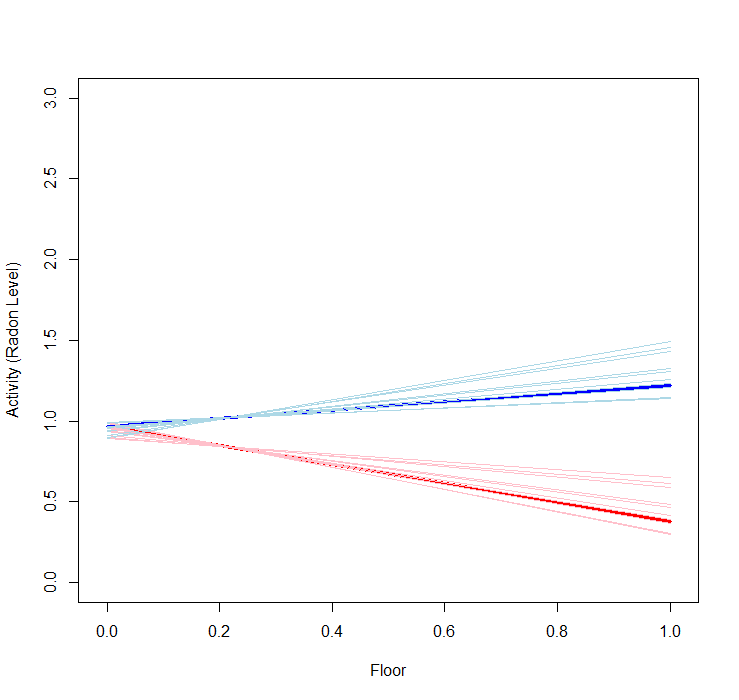
Controlled without floor

If we did not consider the uranium level, the radon level increases with single family house like the red line. If we did consider the uranium level, the radon level increases with single family house like the blue line. The light blue and pink lines show the trend of first 10 counties. And uranium level can lead to a stage that radon level increases.



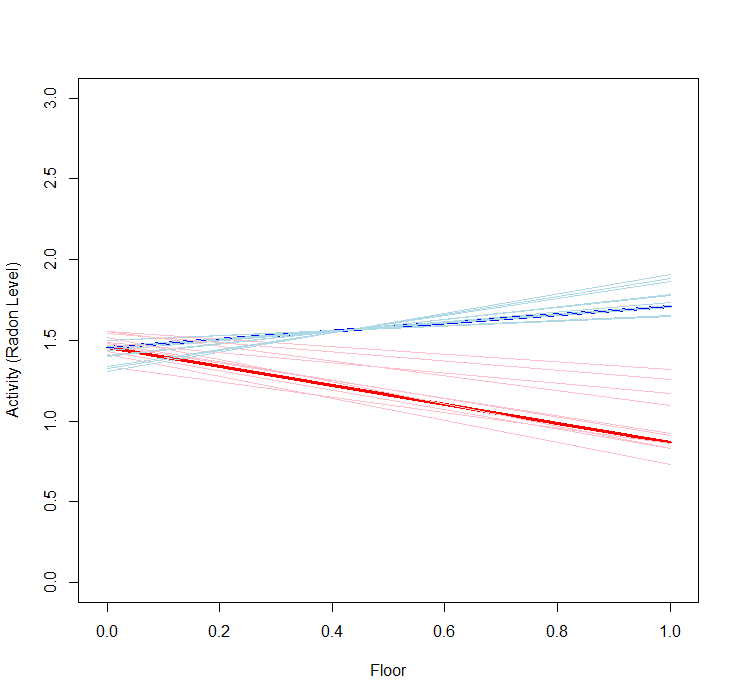
Controlled with floor

If we did not consider the uranium level, the radon level increases with single family house like the red line. If we did consider the uranium level, the radon level increases with single family house like the blue line. The light blue and pink lines show the trend of first 10 counties. And uranium level can lead to a stage that radon level increases.



Controlled without single family house

If we did not consider the uranium level, the radon level decreases with floor like the red line. If we did consider the uranium level, the radon level increases with floor like the blue line. The light blue and pink lines show the trend of first 10 counties. And uranium level can lead to a stage that radon level increases.



Controlled with single family house

If we did not consider the uranium level, the radon level decreases with floor like the red line. If we did consider the uranium level, the radon level increases with floor like the blue line. The light blue and pink lines show the trend of first 10 counties. And uranium level can lead to a stage that radon level increases.

Appendix

# coding: utf-8

# In[12]:

get\_ipython().run\_line\_magic('matplotlib', 'inline')

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

# In[13]:

data = pd.read\_csv("McAlindon\_Big.csv")

print(data.shape)

data.head()

# In[14]:

# Extract the first obs for each individual

ID\_idx = {}

for i, j in enumerate(data['ID']):

ID\_idx[j] = ID\_idx.get(j, i)

Big = data.iloc[[value for (key, value) in sorted(ID\_idx.items())],:]

Big.shape

Big.columns.values

# In[15]:

# Predictors

predictors = ['age', 'treat', 'sex', 'nsaid', 'race2', 'inccat', 'Occupation']

pain\_cols = np.array([16, 18, 20, 22, 24, 26, 28])

time\_cols = pain\_cols + 1

# In[16]:

ids = Big["ID"]

dat = pd.DataFrame()

# In[17]:

for i in range(len(ids)):

id0 = ids.values[i]

temp = data[data["ID"]==id0]

zi = temp[["ID"]+predictors]

pain = temp.iloc[0,pain\_cols]

pain = pain[pain.notnull()]

time = temp.iloc[0,time\_cols]

temperature = temp["avgtemp"][[d in [k for k in time] for d in temp["WeatherDate"]]]

time = time[time.notnull()]

zi = zi[[d in [k for k in time] for d in temp["WeatherDate"]]]

data\_t = pd.concat([pd.concat([pain.reset\_index(drop=True), time.reset\_index(drop=True)],axis=1), temperature.reset\_index(drop=True)], axis=1)

data\_t.columns = ["pain", "time", "temperature"]

data\_t = pd.concat([zi.reset\_index(drop=True),data\_t], axis=1)

dat = pd.concat([dat,data\_t], axis=0)

# In[18]:

dat = dat[dat.iloc[:,0].notnull()]

# In[19]:

dat.iloc[0:10,:]

# In[20]:

# Change some types for each columns

dat["ID"] = dat["ID"].astype(int)

dat["age"] = dat["age"].astype(int)

dat["treat"] = dat["treat"].astype(int)

dat["sex"] = dat["sex"].astype(int)

dat["race2"] = dat["race2"].astype(int)

dat["pain"] = dat["pain"].astype(float)

dat["time"] = dat["time"].astype(int)

# In[21]:

dat.iloc[0:10,:]

# In[22]:

j = 0

for i in ids:

if dat.iloc[j, 0] == i:

target = dat.iloc[j, 9]

while dat.iloc[j, 0] == i:

dat.iloc[j, 9] -= target

j += 1

if j + 1 == dat.shape[0]:

break

# In[23]:

dat.head()

# In[24]:

dat.to\_csv('dat.csv', sep=',')

# Use Pandas to finish the preprocessing part

dat <- read.csv("dat.csv")

dat <- dat[,c(2:12)]

dat$ID <- as.factor(dat$ID)

# Deal with the Occupation Column

idx <- c(grep("tired", dat$Occupation), grep("TIRED", dat$Occupation))

dat$retired <- rep(0,length(dat$Occupation))

idx <- idx[!(idx %in% grep("never retired", dat$Occupation))]

dat$retired[idx] <- 1

dat$retired[which(dat$Occupation=="")] <- NA

dat$retired <- as.factor(dat$retired)

dat <- dat[!is.na(dat$temperature),]

# Model

library(lme4)

library(nlme)

m1 <- lmer(pain ~ 1|ID,REML=F, data=dat)

m2 <- lmer(pain ~ 1 + time + (1+time|ID), REML = F, data = dat)

# Model is nearly unidentifiable: very large eigenvalue - Rescale variables?

dat$time\_scale <- scale(dat$time)

m2 <- lmer(pain ~ 1 + time\_scale+ (1+time\_scale|ID), REML = F, data = dat)

m3 <- lmer(pain ~ 1 + temperature+ (1+temperature|ID), REML = F, data = dat)

# Model is nearly unidentifiable: very large eigenvalue - Rescale variables?

dat$temp\_scale <- scale(dat$temperature)

m3 <- lmer(pain ~ 1 + temp\_scale+ (1+temp\_scale|ID), REML = F, data = dat)

m4 <- lmer(pain ~ 1 + time\_scale +temp\_scale+ (1+time\_scale+temp\_scale|ID), REML = F, data = dat)

AIC(m1, m2, m3, m4)

anova(m2, m4)

# age

m5 <- lmer(pain ~ time\_scale +(time\_scale|ID), REML = F, data = dat, subset = !is.na(dat$age))

m6 <- lmer(pain ~ time\_scale + age + (time\_scale|ID), REML = F, data = dat, subset = !is.na(dat$age))

m7 <- lmer(pain ~ time\_scale \* age + (time\_scale|ID), REML = F, data = dat, subset = !is.na(dat$age))

anova(m5, m6, m7)

# race2

dat$race2 <- as.factor(dat$race2)

m8 <- lmer(pain ~ time\_scale +(time\_scale|ID), REML = F, data = dat, subset = !is.na(dat$race2))

m9 <- lmer(pain ~ time\_scale + race2 + (time\_scale|ID), REML = F, data = dat, subset = !is.na(dat$race2))

m10 <- lmer(pain ~ time\_scale \* race2 + (time\_scale|ID), REML = F, data = dat, subset = !is.na(dat$race2))

anova(m8, m9, m10)

# inccat

dat$inccat <- as.factor(dat$inccat)

m11 <- lmer(pain ~ time\_scale +(time\_scale|ID), REML = F, data = dat, subset = !is.na(dat$inccat))

m12 <- lmer(pain ~ time\_scale + inccat + (time\_scale|ID), REML = F, data = dat, subset = !is.na(dat$inccat))

m13 <- lmer(pain ~ time\_scale \* inccat + (time\_scale|ID), REML = F, data = dat, subset = !is.na(dat$inccat))

anova(m11, m12, m13)

# treat

dat$treat <- as.factor(dat$treat)

m14 <- lmer(pain ~ time\_scale +(time\_scale|ID), REML = F, data = dat, subset = !is.na(dat$treat))

m15 <- lmer(pain ~ time\_scale + treat + (time\_scale|ID), REML = F, data = dat, subset = !is.na(dat$treat))

m16 <- lmer(pain ~ time\_scale \* treat + (time\_scale|ID), REML = F, data = dat, subset = !is.na(dat$treat))

anova(m14, m15, m16)

# sex

dat$sex <- as.factor(dat$sex)

m17 <- lmer(pain ~ time\_scale +(time\_scale|ID), REML = F, data = dat, subset = !is.na(dat$sex))

m18 <- lmer(pain ~ time\_scale + sex + (time\_scale|ID), REML = F, data = dat, subset = !is.na(dat$sex))

m19 <- lmer(pain ~ time\_scale \* sex + (time\_scale|ID), REML = F, data = dat, subset = !is.na(dat$sex))

anova(m17, m18, m19)

# nsaid

dat$nsaid <- as.factor(dat$nsaid)

m20 <- lmer(pain ~ time\_scale +(time\_scale|ID), REML = F, data = dat, subset = !is.na(dat$nsaid))

m21 <- lmer(pain ~ time\_scale + nsaid + (time\_scale|ID), REML = F, data = dat, subset = !is.na(dat$nsaid))

m22 <- lmer(pain ~ time\_scale \* nsaid + (time\_scale|ID), REML = F, data = dat, subset = !is.na(dat$nsaid))

anova(m20, m21, m22) # significant

# retired

m23 <- lmer(pain ~ time\_scale +(time\_scale|ID), REML = F, data = dat, subset = !is.na(dat$retired))

m24 <- lmer(pain ~ time\_scale + retired + (time\_scale|ID), REML = F, data = dat, subset = !is.na(dat$retired))

m25 <- lmer(pain ~ time\_scale \* retired + (time\_scale|ID), REML = F, data = dat, subset = !is.na(dat$retired))

anova(m23, m24, m25)

model <- lmer(pain ~ 1 + nsaid\*time\_scale + (1+time\_scale|ID), REML = F, data = dat, subset = !is.na(dat$nsaid))

summary(model)

#plot

x <- seq(0, 100, by=10)

x <- scale(x)

coefs\_f <- fixed.effects(model)

coefs\_r <- random.effects(model)

plot(x, rep(0, 11), type = "n", ylim = c(0,15), xlab = "Time", ylab = "Pain")

nsaid0 <- coefs\_f[1]+x\*coefs\_f[3]

nsaid1 <- coefs\_f[1] + coefs\_f[2] + x\*(coefs\_f[3] + coefs\_f[4])

points(x, nsaid0, type = "l", col='red', lwd=3)

points(x, nsaid1, type = "l", col='blue', lwd=3)

for (i in 1:7){

yi0 = coefs\_f[1] + x\*coefs\_f[3] +coefs\_r$ID[i,1] + coefs\_r$ID[i,2]\*x

yi1 = coefs\_f[1] + coefs\_f[2] + x\*(coefs\_f[3]+coefs\_f[4]) +coefs\_r$ID[i,1] + coefs\_r$ID[i,2]\*x

points(x, yi0,type="l",col="pink",lwd=0.5)

points(x, yi1,type="l",col="lightblue",lwd=0.5)

}

# Problem 2

srr <- read.table("srrs2.txt", sep = ",", header = T)

cty <- read.table("cty.txt", sep = ",", header = T)

col\_s <- c("idnum","state2","stfips","typebldg","floor","basement","activity","cntyfips")

col\_c <- cty\_col <- c("stfips","ctfips","Uppm")

dat\_s <- srr[col\_s]

dat\_c <- cty[col\_c]

names(dat\_c) <- c("stfips","cntyfips","uranium")

dat\_s$uranium <- rep(0, dim(dat\_s)[1])

for (i in 1:dim(dat\_s)[1]){

if (length(intersect(which(dat\_c$cntyfips==dat\_s$cntyfips[i]), which(dat\_c$stfips==dat\_s$stfips[i])))!=0){

idx <- intersect(which(dat\_c$cntyfips==dat\_s$cntyfips[i]), which(dat\_c$stfips==dat\_s$stfips[i]))

dat\_s$uranium[i] <- dat\_c$uranium[idx]

}

}

dat\_mn <- dat\_s[which(dat\_s$state2=="MN"),]

dat\_mn$floor[which(dat\_mn$floor==9)] <- NA

dat\_mn$single <- rep(0, dim(dat\_mn)[1])

dat\_mn$single[which(dat\_mn$typebldg == 1)] <- 1

dat\_mn$basement[which(dat\_mn$basement == 0)] <- NA

dat\_mn$basement[which(dat\_mn$basement == " ")] <- NA

dat\_mn$activity\_log <- log(dat\_mn$activity)

dat\_mn <- dat\_mn[which(dat\_mn$activity\_log != -Inf), ]

dat\_mn$uranium\_log <- log(dat\_mn$uranium)

dat\_mn <- dat\_mn[!is.na(dat\_mn$floor) & !is.na(dat\_mn$basement),]

dat\_mn$single <- as.factor(dat\_mn$single)

dat\_mn$basement <- as.factor(dat\_mn$basement)

dat\_mn$floor <- as.factor(dat\_mn$floor)

mod1 <- lmer(activity\_log ~ 1|cntyfips, REML = F, data = dat\_mn)

mod2 <- lmer(activity\_log ~ 1 + floor + (1 + floor|cntyfips), REML = F, data = dat\_mn)

mod3 <- lmer(activity\_log ~ 1 + single + (1 + single|cntyfips), REML = F, data = dat\_mn)

mod4 <- lmer(activity\_log ~ 1 + basement + (1 + basement|cntyfips), REML = F, data = dat\_mn)

anova(mod1, mod2) # significant

anova(mod1, mod3) # significant

anova(mod1, mod4) # significant

AIC(mod2, mod3, mod4)

mod5 <- lmer(activity\_log ~ 1 + floor + basement + (1+floor+basement|cntyfips), REML = F, data = dat\_mn)

mod6 <- lmer(activity\_log ~ 1 + floor + single + (1+floor+single|cntyfips), REML = F, data = dat\_mn)

mod7 <- lmer(activity\_log ~ 1 + single + basement + (1+single+basement|cntyfips), REML = F, data = dat\_mn)

mod8 <- lmer(activity\_log ~ 1 + floor + single + basement + (1 + floor+single+basement|cntyfips), REML = F, data = dat\_mn)

anova(mod5, mod8)

anova(mod6, mod8)

anova(mod7, mod8)

anova(mod2,mod6)

anova(mod3, mod6)

modf0 <- lmer(activity\_log ~ 1 + floor + single + (1+floor+single|cntyfips), REML = F, data = dat\_mn,

subset = dat\_mn$uranium\_log != -Inf)

modf1 <- lmer(activity\_log ~ 1 + floor + single + uranium\_log + (1+floor+single|cntyfips), REML = F, data = dat\_mn,

subset = dat\_mn$uranium\_log != -Inf)

modf2 <- lmer(activity\_log ~ 1 + (floor + single) \* uranium\_log + (1+floor+single|cntyfips), REML = F, data = dat\_mn,

subset = dat\_mn$uranium\_log != -Inf)

anova(modf0, modf1, modf2)

summary(modf1)

#plot

cnty <- unique(dat\_mn$cntyfips)[1:10]

ur\_cnty <- dat\_mn$uranium\_log[cnty]

x <- rep(0,10)

x1 <- c(0,1)

coefs\_f <- fixed.effects(modf1)

coefs\_r <- random.effects(modf1)

plot(x, rep(0, 10), type = "n", ylim = c(0,3), xlim=c(-0.01, 1.01), xlab = "Single Family House", ylab = "Activity (Radon Level)")

# floor == 0

uranium0 <- coefs\_f[1]+coefs\_f[3]\*x1

uranium1 <- coefs\_f[1] +(coefs\_f[3]+ coefs\_f[4])\*x1

points(x1, uranium0, type="l", col='red', lwd=3)

points(x1, uranium1, type="l", col='blue', lwd=3)

for (i in 1:10){

yi0 = coefs\_r$cntyfips[i,1] + coefs\_r$cntyfips[i,3]\*x1 + coefs\_f[1]+coefs\_f[3]\*x1

yi1 = coefs\_r$cntyfips[i,1] + (coefs\_r$cntyfips[i,3])\*x1 + coefs\_f[1] +(coefs\_f[3]+ coefs\_f[4])\*x1

points(x1, yi0,type="l",col="pink",lwd=0.5)

points(x1, yi1,type="l",col="lightblue",lwd=0.5)

}

# floor == 1

plot(x, rep(0, 10), type = "n", ylim = c(0,3), xlim=c(-0.01, 1.01), xlab = "Single Family House", ylab = "Activity (Radon Level)")

uranium0 <- coefs\_f[1]+coefs\_f[2]+coefs\_f[3]\*x1

uranium1 <- coefs\_f[1] + coefs\_f[2]+(coefs\_f[3]+ coefs\_f[4])\*x1

points(x1, uranium0, type="l", col='red', lwd=3)

points(x1, uranium1, type="l", col='blue', lwd=3)

for (i in 1:10){

yi0 = coefs\_r$cntyfips[i,1] +coefs\_r$cntyfips[i,2]+ coefs\_r$cntyfips[i,3]\*x1 + coefs\_f[1]+coefs\_f[2]+coefs\_f[3]\*x1

yi1 = coefs\_r$cntyfips[i,1] +coefs\_r$cntyfips[i,2]+ (coefs\_r$cntyfips[i,3])\*x1 + coefs\_f[1] +coefs\_f[2]+(coefs\_f[3]+ coefs\_f[4])\*x1

points(x1, yi0,type="l",col="pink",lwd=0.5)

points(x1, yi1,type="l",col="lightblue",lwd=0.5)

}

# single == 0

plot(x, rep(0, 10), type = "n", ylim = c(0,3), xlim=c(-0.01, 1.01), xlab = "Floor", ylab = "Activity (Radon Level)")

uranium0 <- coefs\_f[1]+coefs\_f[2]\*x1

uranium1 <- coefs\_f[1] +(coefs\_f[2]+ coefs\_f[4])\*x1

points(x1, uranium0, type="l", col='red', lwd=3)

points(x1, uranium1, type="l", col='blue', lwd=3)

for (i in 1:10){

yi0 = coefs\_r$cntyfips[i,1] + coefs\_r$cntyfips[i,2]\*x1 + coefs\_f[1]+coefs\_f[2]\*x1

yi1 = coefs\_r$cntyfips[i,1] + (coefs\_r$cntyfips[i,2])\*x1 + coefs\_f[1] +(coefs\_f[2]+ coefs\_f[4])\*x1

points(x1, yi0,type="l",col="pink",lwd=0.5)

points(x1, yi1,type="l",col="lightblue",lwd=0.5)

}

# single == 1

plot(x, rep(0, 10), type = "n", ylim = c(0,3), xlim=c(-0.01, 1.01), xlab = "Floor", ylab = "Activity (Radon Level)")

uranium0 <- coefs\_f[1]+coefs\_f[3]+coefs\_f[2]\*x1

uranium1 <- coefs\_f[1] +coefs\_f[3]+(coefs\_f[2]+ coefs\_f[4])\*x1

points(x1, uranium0, type="l", col='red', lwd=3)

points(x1, uranium1, type="l", col='blue', lwd=3)

for (i in 1:10){

yi0 = coefs\_r$cntyfips[i,1]+coefs\_r$cntyfips[i,3] + coefs\_r$cntyfips[i,2]\*x1 + coefs\_f[1]+coefs\_f[2]\*x1+coefs\_f[3]

yi1 = coefs\_r$cntyfips[i,1] +coefs\_r$cntyfips[i,1]+ (coefs\_r$cntyfips[i,2])\*x1 + coefs\_f[1] +(coefs\_f[2]+ coefs\_f[4])\*x1+coefs\_f[3]

points(x1, yi0,type="l",col="pink",lwd=0.5)

points(x1, yi1,type="l",col="lightblue",lwd=0.5)

}