qbs121_hw5_gibran

Gibran Erlangga

2/15/2022

Questions

Choose two online datasets that are suitable for use demonstrating (1) normal linear mixed and (2) binary/poisson mixed models. These can be either longitudinal or simply clustered, but should include covariates as well as cluster indicators. 1. For both the linear and nonlinear analyses, describe and justify the longitudinal/clustered outcomes and covariates and the plan for fitting and interpreting mixed random and fixed effects models. 2. Fit the models using lmer and glmer and provide summary statistics and graphs for summarizing the results and assessment of modeling assumptions.

Dataset Justifications

For normal linear mixed models, I am using the "House Prices in the City of Windsor, Canada" dataset, which contains these following variables:
price = sale price of a house
lotsize = the lot size of a property in square feet
bedrooms = the number of bedrooms
bathrooms = the number of full bathrooms
stories = the number of stories excluding basement
driveway = 1 if the house has a driveway
recreation = 1 if the house has a recreational room
fullbase = 1 if the house has a full finished basement
gasheat = 1 if the house uses gas for hot water heating
aircon = 1 if there is central air conditioning
garage = the number of garage places
prefer = 1 if the house is located in the preferred neighbourhood of the city

Cluster indicator in this data set is the prefer variable, which separates between houses that reside in preferred neighborhood in the city and not. The dependent variable is price, and the rest of the variables stated above are the independent variables.

For binary/poisson mixed models, I am using "lung cancer" data set. This data set is sourced from a large Health Maintenance Organization (HMO) wants to know what patient and physician factors are most related to whether a patient's lung cancer goes into remission after treatment as part of a larger study of treatment outcomes and quality of life in patients with lung cancer.

Cluster indicator in this data set is the Doctor ID (DID) variable, which indicates from which doctor unique identifier. The dependent variable is remission, and I plan to use IL6, CRP, CancerStage as patient level categorical independent variables, and Experience as doctor level continuous independent variable.

Normal Linear Mixed Model

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5          v purrr   0.3.4
## v tibble  3.1.5          v dplyr   1.0.7.9000
## v tidyr   1.1.4          v stringr 1.4.0
## v readr   2.0.2          v forcats 0.5.1

## Warning: package 'ggplot2' was built under R version 4.1.1

## Warning: package 'tibble' was built under R version 4.1.1

## Warning: package 'tidyr' was built under R version 4.1.1

## Warning: package 'readr' was built under R version 4.1.1

## Warning: package 'stringr' was built under R version 4.1.1

## Warning: package 'forcats' was built under R version 4.1.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(ggplot2)
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.1.1

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
library(lme4)
```

```
## Loading required package: Matrix

## Warning: package 'Matrix' was built under R version 4.1.1

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
data <- read.csv('HousePrices.csv')
```

The dataset describes house prices in the city of Windsor, Canada (546 rows and 13 columns). The dependent variable is house price, which signifies by the "price" column. The rest of the variables signify all the details about each house presented in the dataset (house size in square feet, number of bedrooms, bathrooms, garages, as well as other factors such as whether or not the house is located in the preferred neighborhood of the city). We can see some sample data from the dataset along with the distribution of the dependent variable below:

```
paste('# of rows/# of columns:', dim(data)[1] ,'/', dim(data)[2])
```

```
## [1] "# of rows/# of columns: 546 / 13"
```

```
print('list of columns: ')
```

```
## [1] "list of columns: "
```

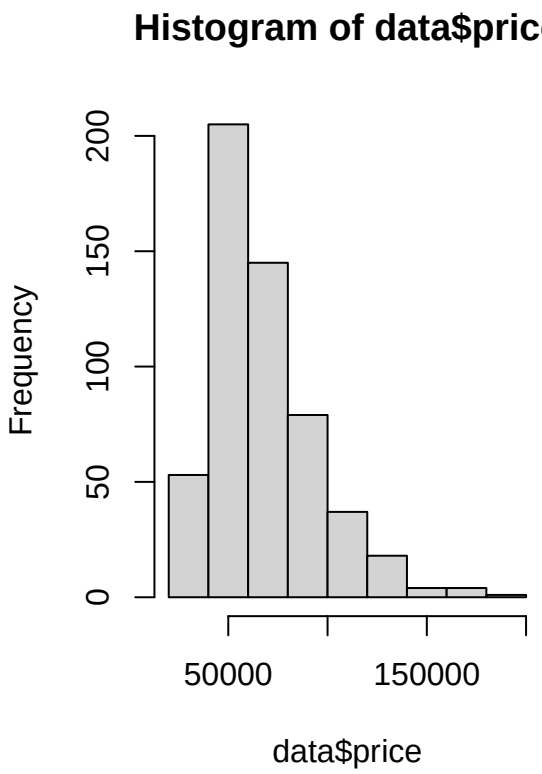
```
names(data)
```

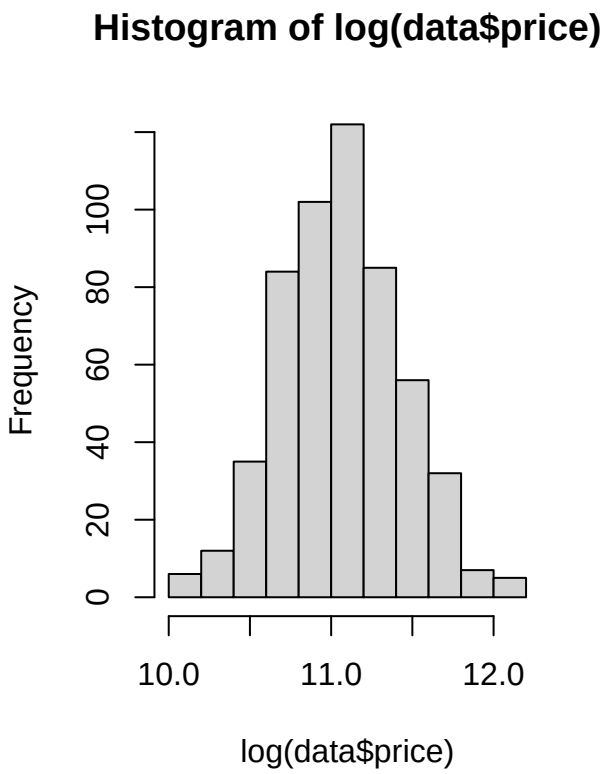
```
##  [1] "X"          "price"      "lotsize"    "bedrooms"   "bathrooms"
##  [6] "stories"    "driveway"   "recreation" "fullbase"   "gasheat"
## [11] "aircon"     "garage"     "prefer"
```

```
# see some sample data
print(head(data, 3))
```

```
##   X price lotsize bedrooms bathrooms stories driveway recreation fullbase
## 1 1 42000    5850        3         1       2      yes         no      yes
## 2 2 38500    4000        2         1       1      yes         no       no
## 3 3 49500    3060        3         1       1      yes         no       no
##   gasheat aircon garage prefer
## 1      no     no      1     no
## 2      no     no      0     no
## 3      no     no      0     no
```

```
# plot dependent and independent variables
par(mfrow=c(1,2))
hist(data$price)
hist(log(data$price))
```

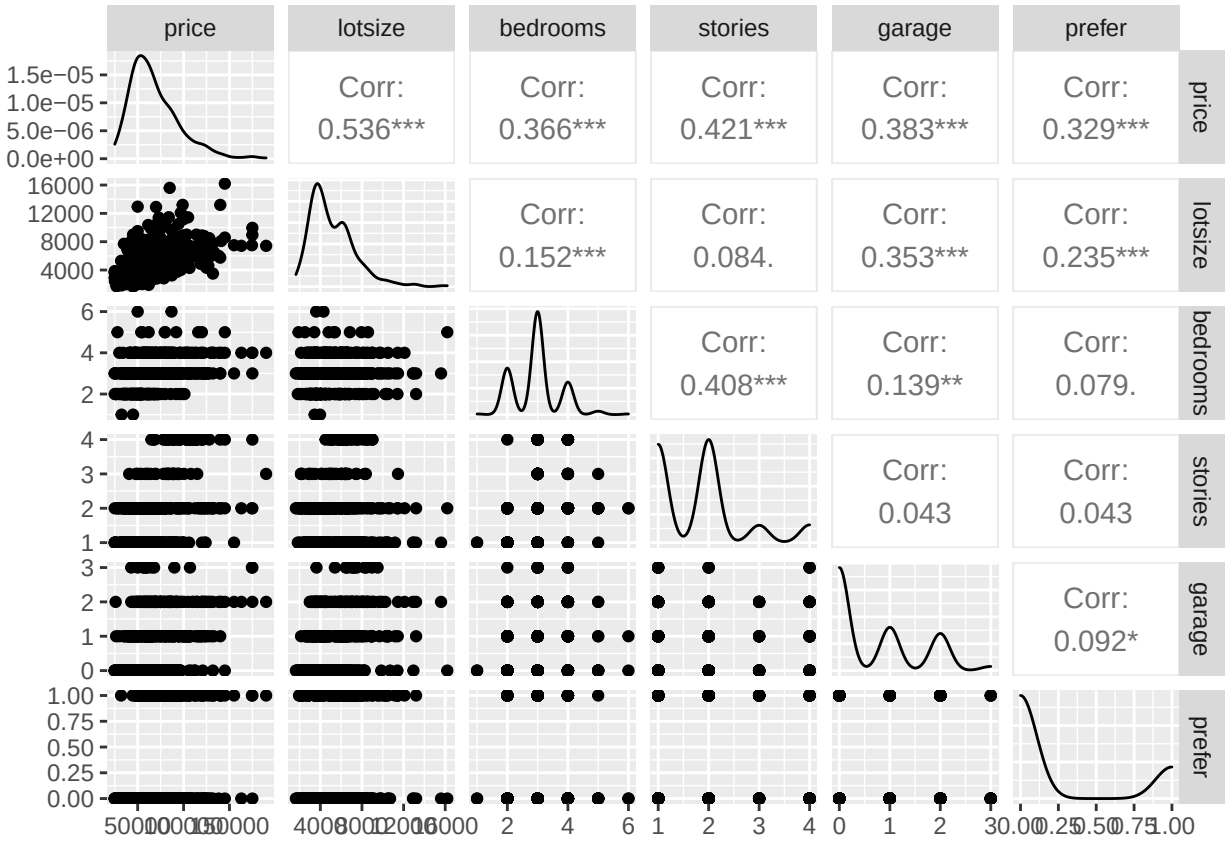




Above figures show the distribution of the dependent variable, the original one (left) and the one after applying a log transformation to the data (right). We can observe that the house price distribution is skewed to the right, meaning that it has a long right tail and the variable mean to the right of the median. To comply with one of the assumptions of linear regression, I applied a log-transform to the house price variable to make it more normally distributed.

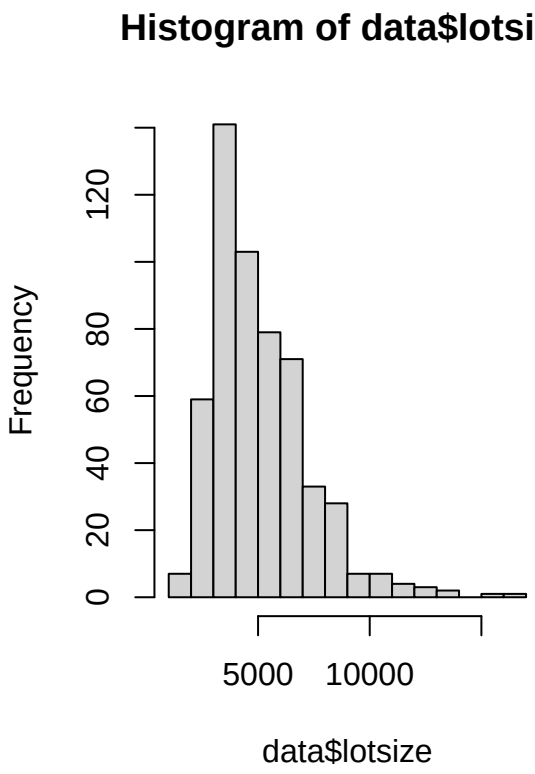
I selected a handful of independent variables as potential predictors for the house price. The variables are: lotsize -> lot size of a property in square feet bedrooms -> number of bedrooms stories -> number of stories excluding basement garage -> number of garages prefer -> a flag that shows whether the house located in the preferred neighborhood of the city

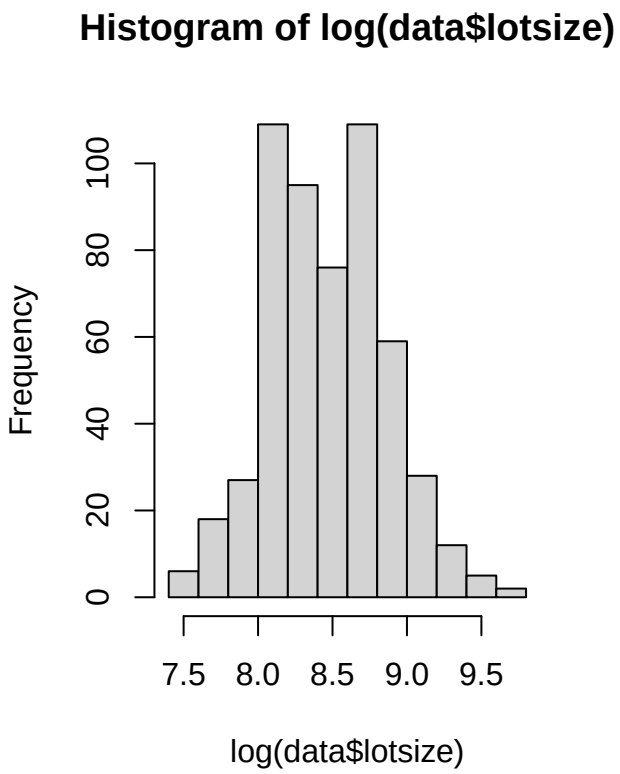
```
boolean_convert <- function(data) {
  if (data == "yes") {
    return(1)
  } else {
    return(0)
  }
}

data$prefer <- sapply(data$prefer, boolean_convert)

df <- data %>%
      select('price', 'lotsize', 'bedrooms', 'stories', 'garage', 'prefer')

ggpairs(df)
```



Aside from price column, from above graph we can see that lotsize column also shows a right-skewed distribution, so I applied a log transformation to make it more normally distributed. Here's what it looks like before-after (graph below, left-right):

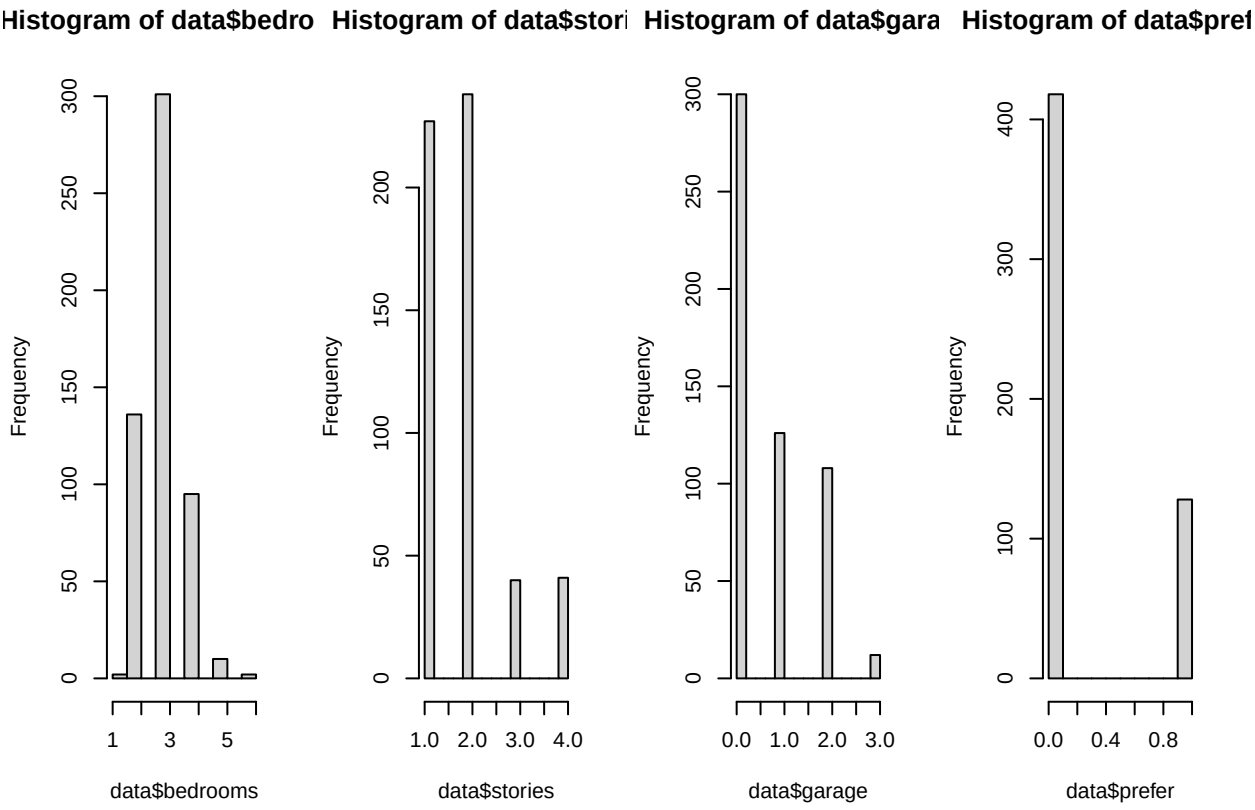
```
par(mfrow=c(1,2))
hist(data$lotsize)
hist(log(data$lotsize))
```





And I also plotted the distribution of other independent variables (bedrooms, stories, garage, prefer) I chose earlier.

```
par(mfrow=c(1, 4))
hist(data$bedrooms)
hist(data$stories)
hist(data$garage)
hist(data$prefer)
```

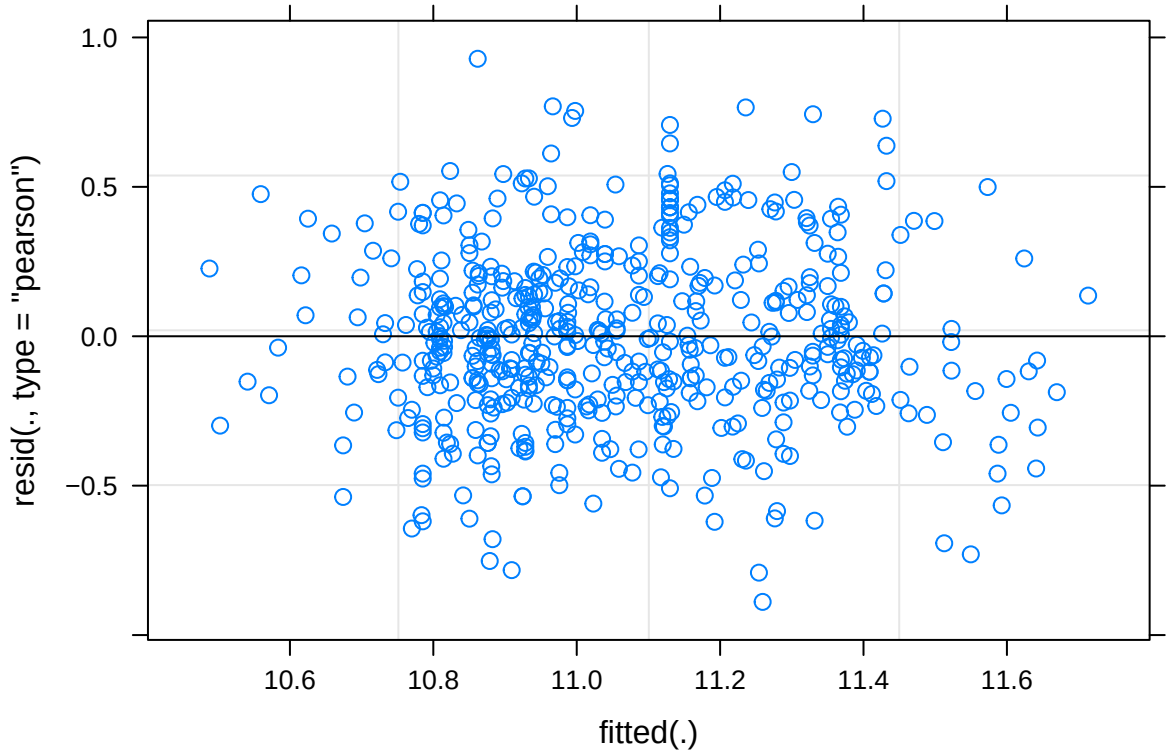



```
df$log_price <- log(df$price)

log_price_mixed <- lmer(log_price ~ log(lotsize) + (1 | prefer), data = df)
summary(log_price_mixed)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: log_price ~ log(lotsize) + (1 | prefer)
##    Data: df
##
## REML criterion at convergence: 221
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.0386 -0.6238 -0.0449  0.6367  3.1743
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  prefer   (Intercept) 0.01868  0.1367
##  Residual             0.08559  0.2926
## Number of obs: 546, groups:  prefer, 2
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)   6.89748    0.29174   23.64
## log(lotsize)  0.49751    0.03226   15.42
```

```
##
## Correlation of Fixed Effects:
##             (Intr)
## log(lotsiz) -0.942
```

```
plot(log_price_mixed)
```



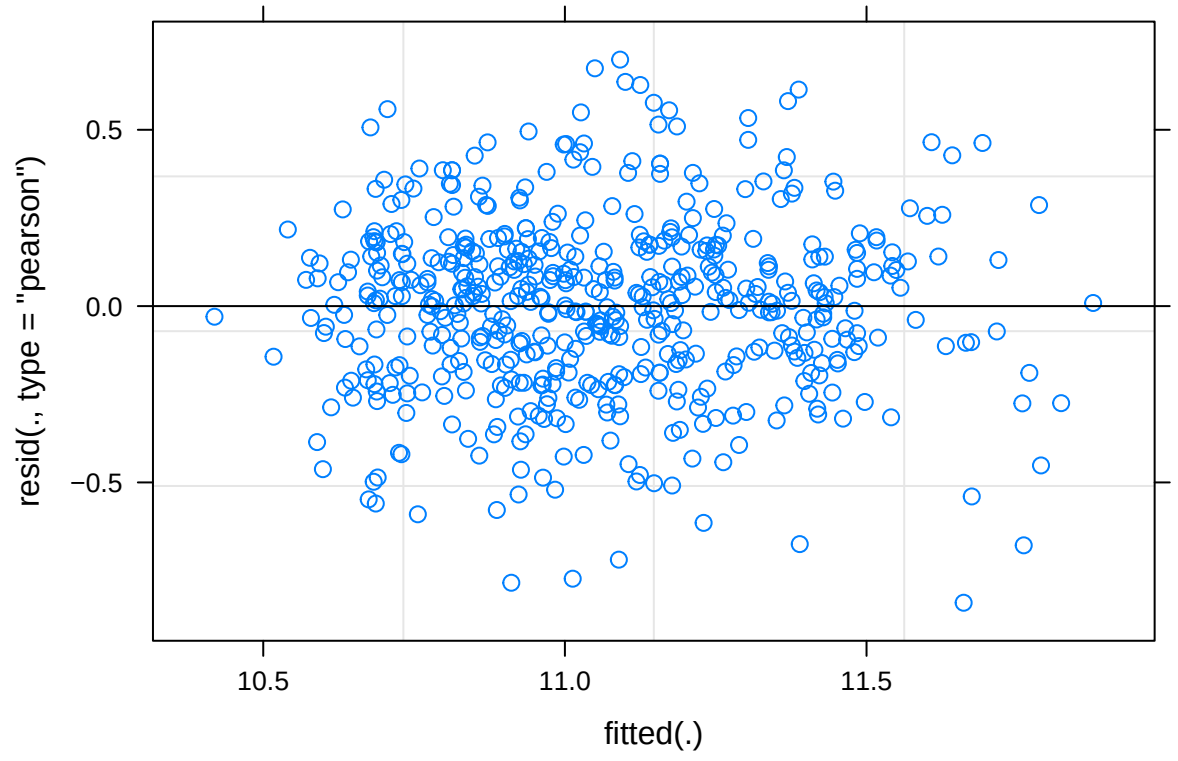
In above modeling attempt, I set prefer as a random effect to the model, while setting up log(lotsize) as a fixed effect. We can see the details of the coefficient on prefer variable in the random effect section and coefficient on log(lotsize) on the fixed effect section, along with its consecutive standard error. Additionally, we can also see the number of observations (546) and the level 2 observations (2) in the details of random effect.

```
log_price_mixed_cluster <- lmer(log_price ~ log(lotsize) + bedrooms + garage + stories +
                                  (1 | prefer), data = df)
summary(log_price_mixed_cluster)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: log_price ~ log(lotsize) + bedrooms + garage + stories + (1 |
##     prefer)
##    Data: df
##
## REML criterion at convergence: 66.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -3.3665 -0.6408  0.0572  0.6145  2.7973
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  prefer   (Intercept) 0.01627  0.1275
##  Residual             0.06244  0.2499
## Number of obs: 546, groups:  prefer, 2
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)   7.27369    0.26446  27.504
## log(lotsize)  0.39387    0.02962  13.299
## bedrooms      0.07422    0.01610   4.609
## garage        0.07185    0.01339   5.366
## stories       0.12618    0.01353   9.326
##
## Correlation of Fixed Effects:
##             (Intr) lg(lt) bedrms garage
## log(lotsiz) -0.924
## bedrooms    -0.079 -0.066
## garage       0.301 -0.340 -0.098
## stories      0.035 -0.059 -0.399  0.035
```

```
plot(log_price_mixed_cluster)
```



The results above show the similar modeling framework, only with an additional variables included as the predictors for fixed effects (bedrooms, garage, stories). We can compare this with the result on previous

model where we only use log(lotsize) as fixed effect. We can see that the coefficients of both random and fixed effects got adjusted with the presence of additional variables. The changes on random effect coefficient is not that much while the changes on fixed effect coefficient (log(lotsize)) seems to be more apparent.

```
# mixed random effects
log_price_mixed_random <- lmer(log_price ~ log(lotsize) + garage + stories +
                               (1 + log(lotsize) | prefer), data = df)
```

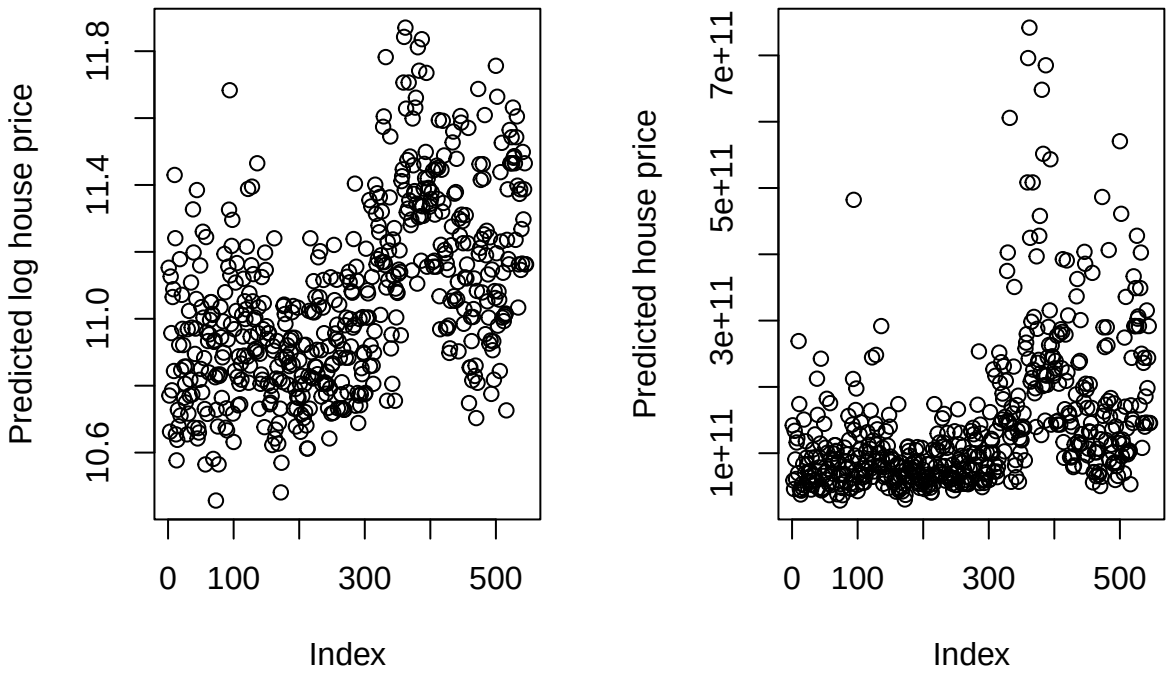
```
## boundary (singular) fit: see ?isSingular
```

```
summary(log_price_mixed_random)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: log_price ~ log(lotsize) + garage + stories + (1 + log(lotsize) |
##     prefer)
##    Data: df
##
## REML criterion at convergence: 81.3
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.4076 -0.6017  0.0369  0.5616  3.0139
##
## Random effects:
##  Groups   Name         Variance  Std.Dev.  Corr
##  prefer   (Intercept)  1.768e-02 0.1329799
##           log(lotsize) 4.933e-08 0.0002221 -1.00
##  Residual              6.477e-02 0.2545061
## Number of obs: 546, groups:  prefer, 2
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)   7.37081    0.26928  27.372
## log(lotsize)  0.40274    0.03010  13.380
## garage        0.07788    0.01357   5.739
## stories       0.15109    0.01263  11.959
##
## Correlation of Fixed Effects:
##             (Intr) lg(lt) garage
## log(lotsiz) -0.933
## garage       0.295 -0.348
## stories      0.003 -0.094 -0.004
## optimizer (nloptwrap) convergence code: 0 (OK)
## boundary (singular) fit: see ?isSingular
```

```
predicted_log_price <- predict(log_price_mixed_random, type="response")

# back-transform from log into original unit
back_transform <- function(data) {
  return(10**data)
}
```

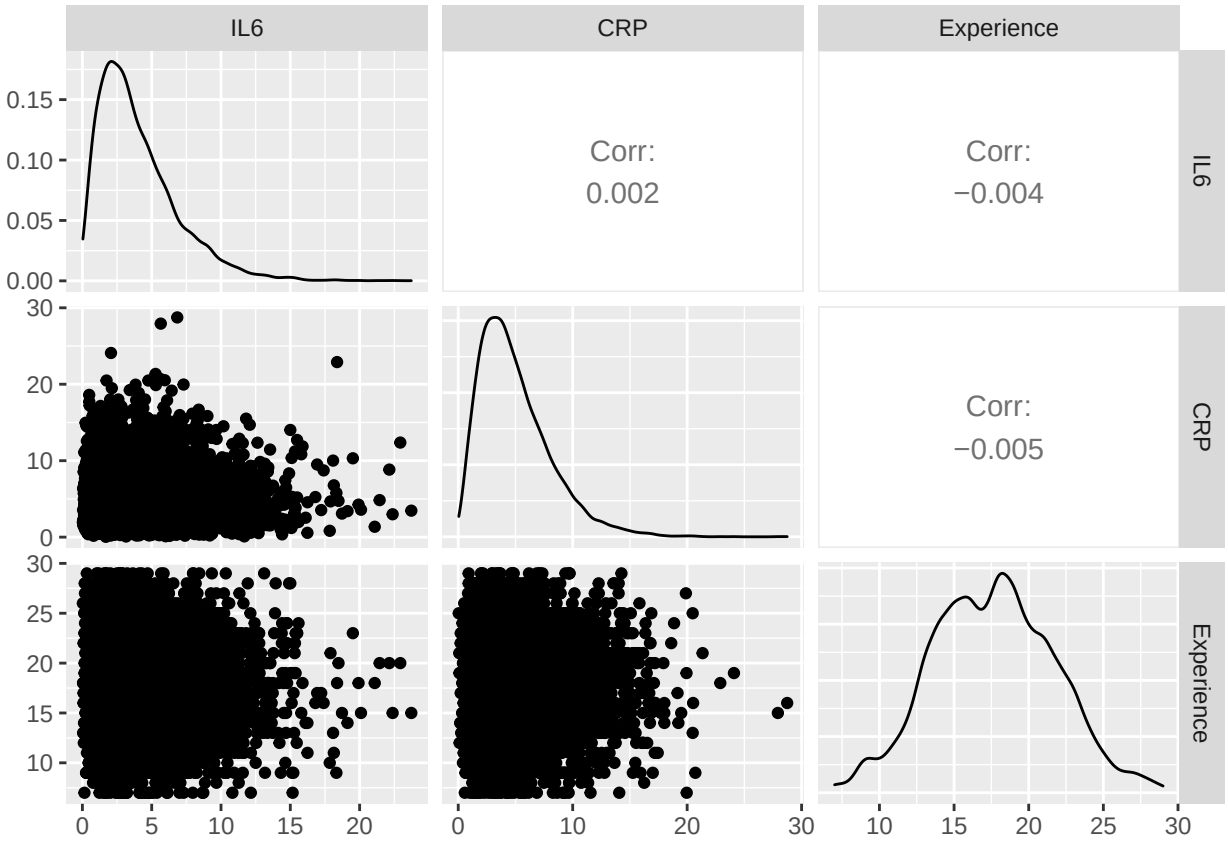
```
predicted_price <- sapply(predicted_log_price, back_transform)

par(mfrow=c(1,2))
plot(predicted_log_price, ylab="Predicted log house price")
plot(predicted_price, ylab="Predicted house price")
```



Binary/Poisson Mixed Model

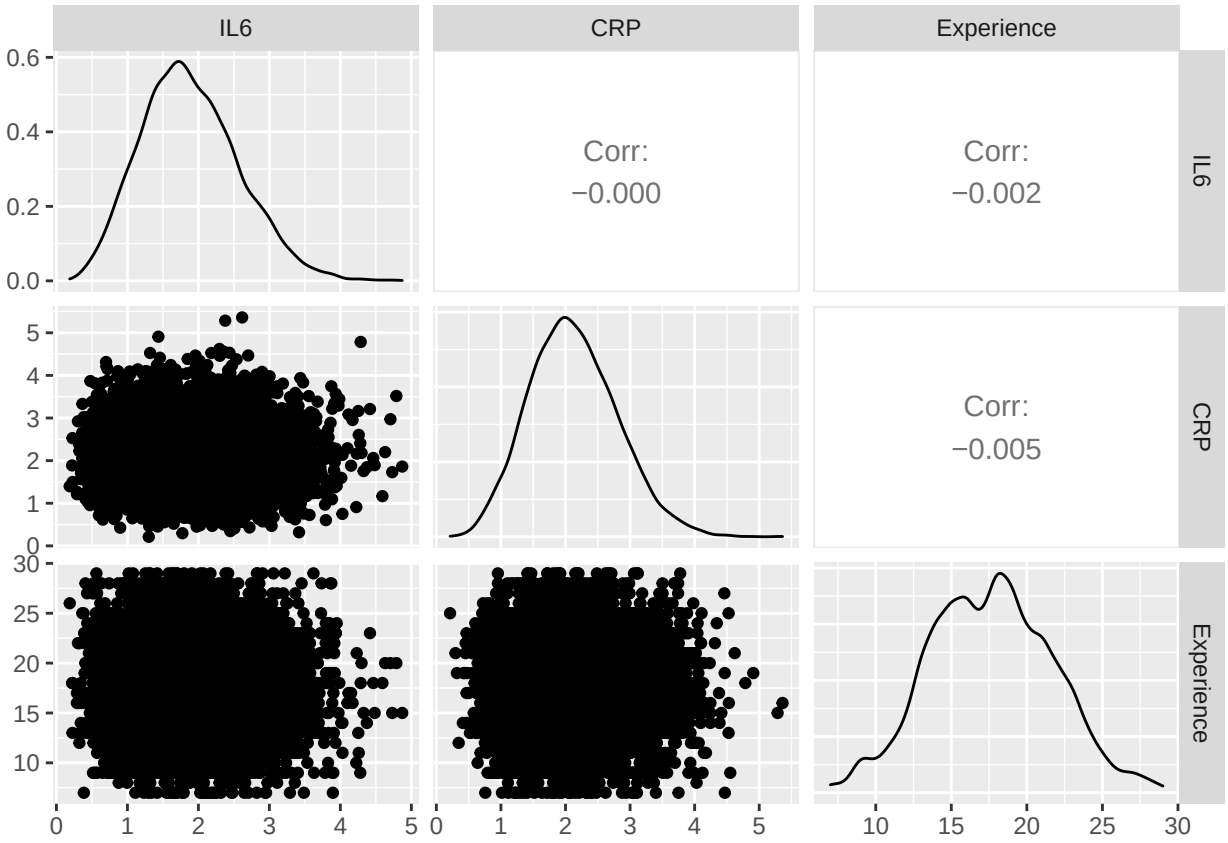
```
data_binary <- read.csv('https://stats.idre.ucla.edu/stat/data/hdp.csv')
```

```
data_binary <- within(data_binary, {
  Married <- factor(Married, levels = 0:1, labels = c("no", "yes"))
  DID <- factor(DID)
  HID <- factor(HID)
  CancerStage <- factor(CancerStage)
})

# plotting numerical independent variables
ggpairs(data_binary[, c("IL6", "CRP", "Experience")])
```



It looks like IL6 and CRP variable have a right-skewed distribution, so I applied the square-root transformation on IL6 and CRP columns to make it more normally distributed. On the other side, Experience, tumorsize and co2 have bell-shaped, normal distribution. We can see that there are no strong correlation between our continuous independent variables, so each variable brings a different information to the model.

```
data_binary$IL6 <- sqrt(data_binary$IL6)
data_binary$CRP <- sqrt(data_binary$CRP)

ggpairs(data_binary[, c("IL6", "CRP", "Experience")])
```




```
m <- glmer(remission ~ IL6 + CRP + CancerStage + Experience +
    (1 | DID), data = data_binary, family = binomial, nAGQ = 10)

summary(m)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 10) [glmerMod]
##  Family: binomial  ( logit )
## Formula: remission ~ IL6 + CRP + CancerStage + Experience + (1 | DID)
##    Data: data_binary
##
##      AIC      BIC   logLik deviance df.resid
##   7409.8   7466.2  -3696.9   7393.8     8517
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.5262 -0.4452 -0.1992  0.4006  7.1522
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  DID    (Intercept) 4.044    2.011
## Number of obs: 8525, groups:  DID, 407
##
## Fixed effects:
##                Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)    -2.32678    0.51875  -4.485 7.28e-06 ***
## IL6            -0.22469    0.04673  -4.808 1.52e-06 ***
## CRP            -0.10036    0.04648  -2.159   0.0309 *
## CancerStageII  -0.48647    0.07296  -6.668 2.60e-11 ***
## CancerStageIII -1.12434    0.09238 -12.171  < 2e-16 ***
## CancerStageIV  -2.50947    0.15083 -16.638  < 2e-16 ***
## Experience      0.11919    0.02740   4.350 1.36e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) IL6    CRP    CncSII CnSIII CncSIV
## IL6         -0.162
## CRP         -0.189  0.004
## CancerStgII -0.067  0.009 -0.005
## CancrStgIII -0.047  0.011  0.006  0.446
## CancerStgIV -0.025  0.035  0.014  0.281  0.249
## Experience  -0.940 -0.006 -0.001 -0.008 -0.013 -0.019
```

In code above, I used the glmer command to run a mixed effects logistic regression model with Il6 and CRP as patient level continuous predictors, CancerStage as a patient level categorical predictor (I, II, III, or IV), Experience as a doctor level continuous predictor, a random intercept by DID (doctor ID) and remission as the dependent variable.

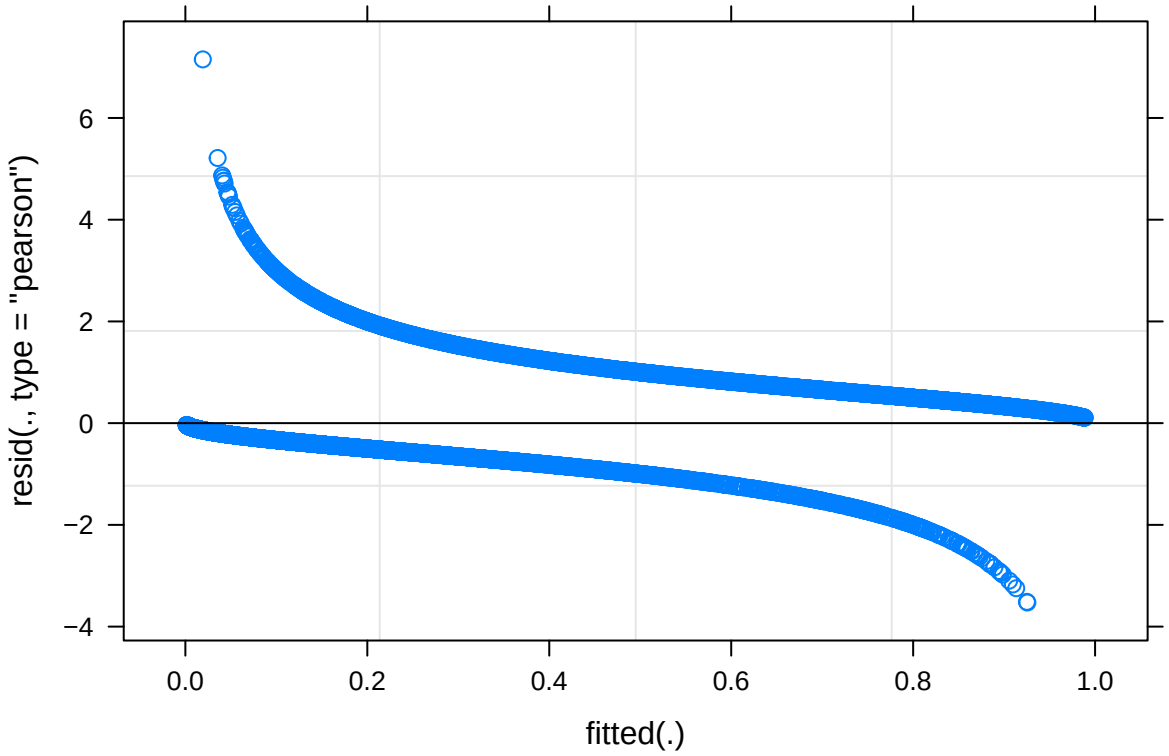
The first part tells us the estimates are based on an adaptive Gaussian Hermite approximation of the likelihood. In particular we used 10 integration points. I specified 10 integration points (nAGQ) to ensure convergence and not taking too much computational power (increase in integration points will improve probability of convergence, but also increase computational power needed to run the model).

The next section gives us basic information that can be used to compare models, followed by the random effect estimates. This represents the estimated variability in the intercept on the logit scale. Had there been other random effects, such as random slopes, they would also appear here. It also shows the total number of observations, and the number of level 2 observations. In our case, this includes the total number of patients (8,525) and doctors (407).

The next section is a table of the fixed effects estimates. The estimates represent the regression coefficients. These are unstandardized and are on the logit scale. The estimates are followed by their standard errors (SEs). The SE values here are approximations because it is more likely to stabilize faster than actual SEs. Therefore, if you are using fewer integration points, the estimates may be reasonable, but the approximation of the SEs may be less accurate. The Wald tests, (frac{Estimate}{SE}), rely on asymptotic theory, here referring to as the highest level unit size converges to infinity, these tests will be normally distributed, and from that, p values (the probability of obtaining the observed estimate or more extreme, given the true estimate is 0).

The last section shows us the correlation values between fixed effect variables. I think the correlation values shown here are pretty great, as most of the values are not correlated except for CancrStgII-CancrStgIII and CancrStgIII-CancrStgIV.

```
plot(m)
```


resid(., type = "pearson")
6
4
2
0
−2
−4
0.0
0.2
0.4
0.6
0.8
1.0
fitted(.)