

Part II: Generalized Linear Models

Load Packages

Again, we must load the packages that will be used in the first part of this workshop.

```
library(pastecs, quietly = TRUE)
library(lm.beta, quietly = TRUE)
library(lmtest, quietly = TRUE)
library(foreign, quietly = TRUE)
library(lattice, quietly = TRUE)
library(lme4, quietly = TRUE)
library(nlme, quietly = TRUE)
library(survival, quietly = TRUE)
library(dplyr, quietly = TRUE)
library(ggfortify, quietly = TRUE)
library(survminer, quietly = TRUE)
library(rms, quietly = TRUE)
library(MASS, quietly = TRUE)
```

Generalized linear models

A generalized linear model (GLM) has three:

- a random component with mean μ . Generally, the random component is the response variable Y_i .
- a systematic component, η_i , that relates the explanatory variables,

$$\eta_i = \sum_{j=1}^n \beta_j x_{ij}$$

- a link function that relates the mean of the random to the systematic component

$$g(\mu) = \eta_i$$

Logistic regression

Logistic regression is a GLM used the model binary (0 or 1) data. The response variable must be binary and is assumed to follow a bernoulli distribution.

That said, logistic regression has the following properties: - a response binary variable, Y_i , that follows a bernoulli distribution with mean π_i . - a systematic component, η_i , that relates the explanatory variables,

$$\eta_i = \sum_{j=1}^n \beta_j x_{ij}$$

- a link function that relates the mean of the random to the systematic component

$$\log \left(\frac{\pi_i}{1 - \pi_i} \right) = \sum_{j=1}^n \beta_j x_{ij}.$$

$\log \left(\frac{\pi_i}{1 - \pi_i} \right)$ is known as the log odds.

Data

Using the iris data, we create binary data. We add the column `Sepal.Width_binary` to `iris`. If the `Sepal.Width` is greater than the median then the associated value in `Sepal.Width_binary` is 1. Otherwise, `Sepal.Width_binary` is 0.

```
data <- iris
data$Sepal.Width_binary <- ifelse(data$Sepal.Width >= median(data$Sepal.Width), 1, 0)
```

Logistic Regression with only the constant term

Fitting only a constant term, the systematic component is

$$\eta_i = \beta_0$$

```
logit <- glm(Sepal.Width_binary ~ 1, data = data, family = "binomial")
summary(logit)

##
## Call:
## glm(formula = Sepal.Width_binary ~ 1, family = "binomial", data = data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.3911  -1.3911   0.9778   0.9778   0.9778
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.4895     0.1682    2.91  0.00361 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 199.22  on 149  degrees of freedom
## Residual deviance: 199.22  on 149  degrees of freedom
## AIC: 201.22
##
## Number of Fisher Scoring iterations: 4

p_avg <- mean(data$Sepal.Width_binary)
log_odds_avg <- log(p_avg/(1-p_avg))
print(log_odds_avg)

## [1] 0.4895482
```

Logistic Regression with Species

Fitting the species term, the systematic component is

$$\eta_i = 1 + \beta_2 X_{1i} + \beta_3 X_{2i}.$$

where

$$X_{1i} = \begin{cases} 1 & \text{if } i\text{th data point is versicolor} \\ 0 & \text{otherwise} \end{cases}, \quad X_{2i} = \begin{cases} 1 & \text{if } i\text{th data point is virginica} \\ 0 & \text{otherwise} \end{cases}$$

```
logit <- glm(Sepal.Width_binary ~ as.factor(Species), data = data, family = "binomial")
summary(logit)
```

```
##
## Call:
## glm(formula = Sepal.Width_binary ~ as.factor(Species), family = "binomial",
##      data = data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.5373  -0.8782   0.2857   1.0438   1.5096
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      3.1781     0.7215   4.405 1.06e-05 ***
## as.factor(Species)versicolor -3.9318     0.7826  -5.024 5.06e-07 ***
## as.factor(Species)virginica  -2.8553     0.7763  -3.678 0.000235 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 199.22  on 149  degrees of freedom
## Residual deviance: 147.51  on 147  degrees of freedom
## AIC: 153.51
##
## Number of Fisher Scoring iterations: 5
```

Let's compare the results to the average log odds of each Species group

```
log_odds_avg_fun <- function(data){
  p_avg <- mean(data)
  log_odds_avg <- log(p_avg/(1-p_avg))
  return(log_odds_avg)
}

tapply(data$Sepal.Width_binary,
        data$Species, log_odds_avg_fun)
```

```
##      setosa versicolor  virginica
## 3.1780538 -0.7537718  0.3227734
```

The intercept corresponds to the average log odds of setosa as we would expect. However, the other coefficients do not correspond to the average log odds of the other species. Why?

From the formula, $\eta_i = 1 + \beta_2 X_{2i} + \beta_3 X_{3i}$, the log odds of versicolor actually corresponds to $1 + \beta_2$. The log odds of virginica actually corresponds to $1 + \beta_3$.

```
coefficients<-unname(coef(logit))
print(c(coefficients[1],coefficients[1]+coefficients[2],
        coefficients[1]+coefficients[3]))
```

```
## [1] 3.1780537 -0.7537718  0.3227734
```

Logistic Regression with Sepal.Length

Fitting the species term, the systematic component is

$$\eta_i = \beta_3 X_{1i}.$$

where

$$X_{1i} = \begin{cases} 1 & \text{if } i\text{th data point is versicolor} \\ 0 & \text{otherwise} \end{cases}, \quad X_{2i} = \begin{cases} 1 & \text{if } i\text{th data point is virginica} \\ 0 & \text{otherwise} \end{cases}$$

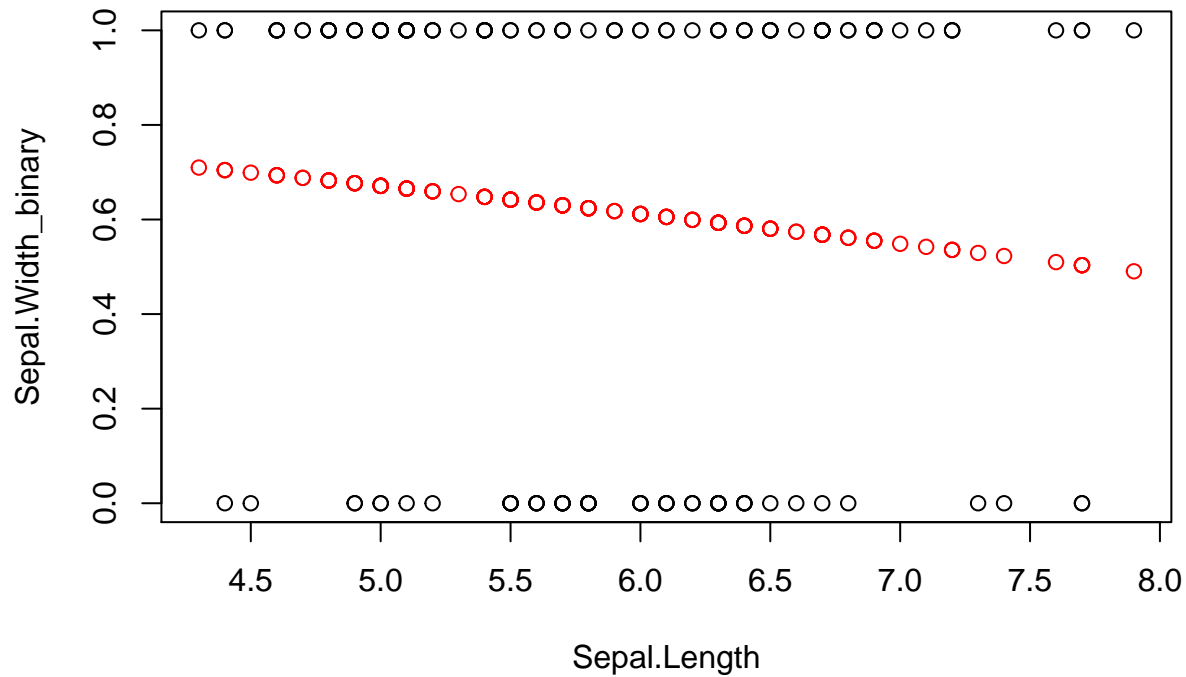
and X_{3i} = Sepal.Length of the i th data point.

```
logit <- glm(Sepal.Width_binary ~ Sepal.Length,
             data = data, family = "binomial")
summary(logit)

##
## Call:
## glm(formula = Sepal.Width_binary ~ Sepal.Length, family = "binomial",
##      data = data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.5614  -1.3524   0.8883   0.9890   1.1936
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    2.0088     1.2176   1.650   0.099 .
## Sepal.Length  -0.2591     0.2050  -1.264   0.206
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 199.22  on 149  degrees of freedom
## Residual deviance: 197.61  on 148  degrees of freedom
## AIC: 201.61
##
## Number of Fisher Scoring iterations: 4

plot(Sepal.Width_binary~Sepal.Length, data=data)
points(data$Sepal.Length[order(data$Sepal.Length)],
       logit$fitted[order(data$Sepal.Length)], col="red")
title(main="Data with Fitted Logistic Regression Line")
```

Data with Fitted Logistic Regression Line



Logistic Regression with Species and Sepal.Length

Fitting the species term, the systematic component is

$$\eta_i = 1 + \beta_2 X_{1i} + \beta_3 X_{2i} + \beta_3 X_{3i}.$$

where

$$X_{1i} = \begin{cases} 1 & \text{if } i\text{th data point is versicolor} \\ 0 & \text{otherwise} \end{cases}, \quad X_{2i} = \begin{cases} 1 & \text{if } i\text{th data point is virginica} \\ 0 & \text{otherwise} \end{cases}$$

and X_{3i} = Sepal.Length of the i th data point.

Fitting the logistic model accordingly,

```
logit <- glm(Sepal.Width_binary ~ Species + Sepal.Length,
             data = data, family = "binomial")
summary(logit)
```

```
##
## Call:
## glm(formula = Sepal.Width_binary ~ Species + Sepal.Length, family = "binomial",
##      data = data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.2710  -0.7538   0.2472   0.7020   1.9477
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -4.7988     2.2981  -2.088 0.036784 *
```

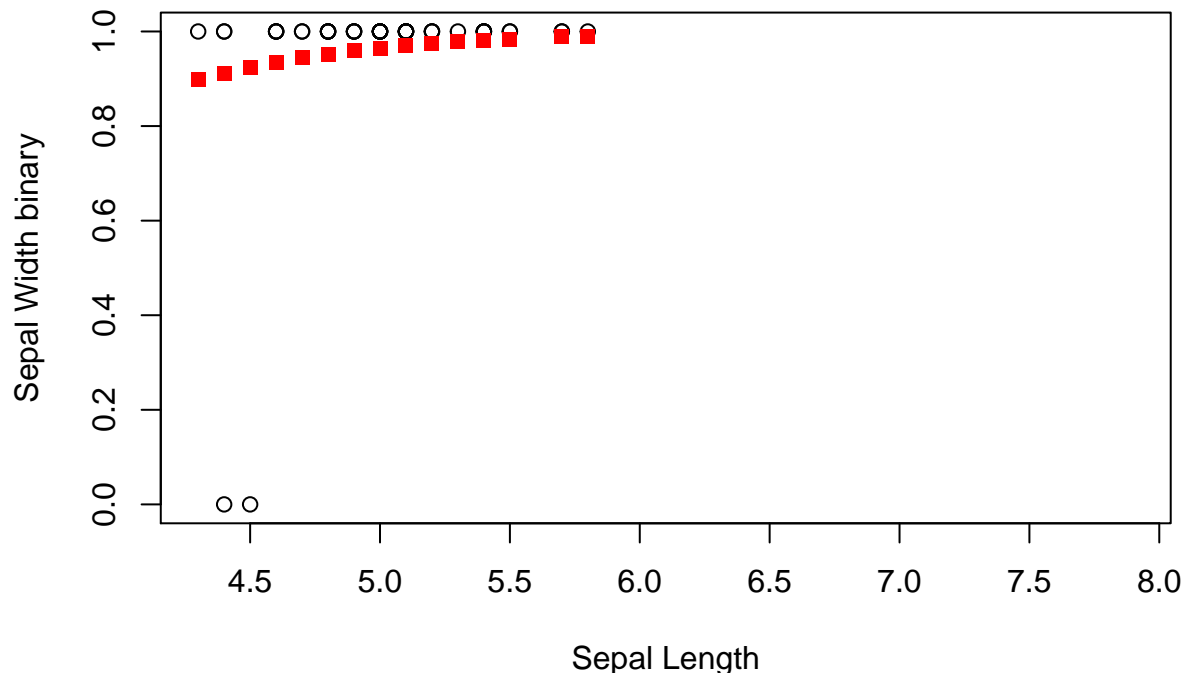
```
## Speciesversicolor -5.6936      0.9686 -5.878 4.16e-09 ***
## Speciesvirginica  -5.4812      1.0879 -5.039 4.69e-07 ***
## Sepal.Length      1.6219      0.4510   3.596 0.000323 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 199.22  on 149  degrees of freedom
## Residual deviance: 131.27  on 146  degrees of freedom
## AIC: 139.27
##
## Number of Fisher Scoring iterations: 6
```

Plot the results for each species, we get that

```
plot(data[data$Species == "setosa", ]$Sepal.Length,
      data[data$Species == "setosa", ]$Sepal.Width_binary,
      xlim=as.matrix(range(data$Sepal.Length)),
      xlab = 'Sepal Length', ylab= 'Sepal Width binary',
      main= 'Scatter plot of sepal length vs sepal width')

points(data$Sepal.Length[data$Species == "setosa"],
        logit$fitted[data$Species == "setosa"], pch=15,
        col="red")
```

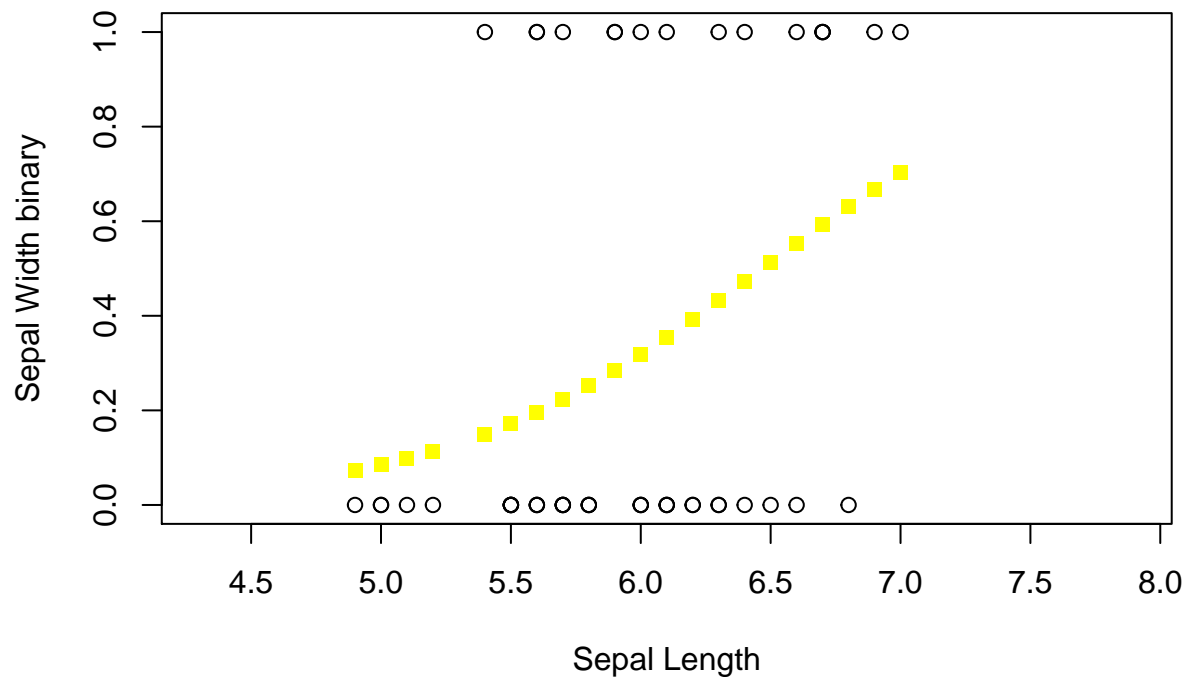
Scatter plot of sepal length vs sepal width



```
plot(data[data$Species == "versicolor", ]$Sepal.Length,
      data[data$Species == "versicolor", ]$Sepal.Width_binary,
      xlim=as.matrix(range(data$Sepal.Length)),
      xlab = 'Sepal Length', ylab= 'Sepal Width binary',
      main= 'Scatter plot of sepal length vs sepal width')
```

```
points(data$Sepal.Length[data$Species == "versicolor"],
       logit$fitted[data$Species == "versicolor"], pch=15,
       col="yellow")
```

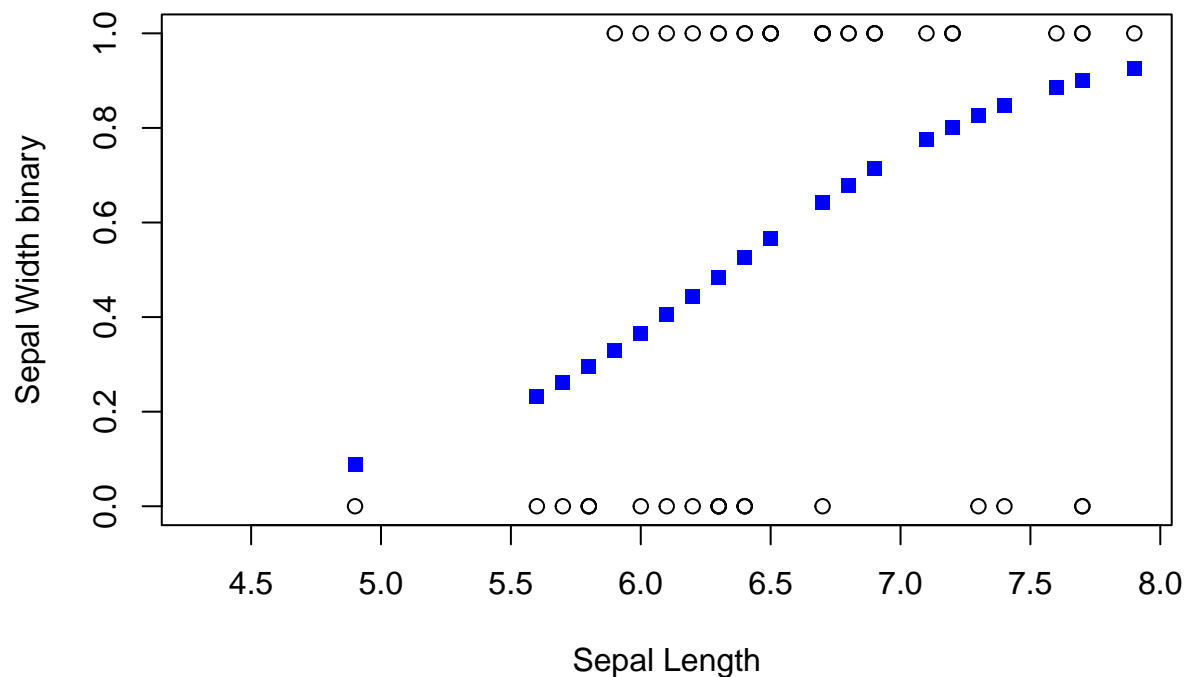
Scatter plot of sepal length vs sepal width



```
plot(data[data$Species == "virginica", ]$Sepal.Length,
     data[data$Species == "virginica", ]$Sepal.Width_binary,
     xlim=as.matrix(range(data$Sepal.Length)),
     xlab = 'Sepal Length', ylab= 'Sepal Width binary',
     main= 'Scatter plot of sepal length vs sepal width')

points(data$Sepal.Length[data$Species == "virginica"],
       logit$fitted[data$Species == "virginica"], pch=15,
       col="blue")
```

Scatter plot of sepal length vs sepal width



Deviance

For general linear models, we use *deviance* to compare two different models. Deviance is the difference in log likelihood of the models multiplied by 2.

Saturated Model

Let's consider a model in which each data point has its own mean and coefficients. This is called the saturated model. It basically replicates the data at hand.

Using deviance, we can compare our fitted model to a saturated model. If the fitted model behaves similarly to the saturated model, then the deviance can be well approximated by a chi-squared distribution with $m - n$ degrees of freedom. m is the number of data points and n is the number of coefficients in our fitted model.

This statistical property of the deviance allows us to perform a hypothesis test

H_0 : the fitted model is equivalent to the saturated model

H_a : the fitted model is not equivalent to the saturated model

`logit$deviance` is the deviance between the saturated model and the fitted model. `logit$df.residual` is equal to the number of observations minus the number of coefficients in the fitted model. Using this, we can calculate the p-value for the hypothesis test above.

```
p_value = pchisq(logit$deviance,
                  logit$df.residual, lower.tail = F)
print(p_value)
```

```
## [1] 0.8032738
```


Since the p value is less than 0.05, we fail to reject the null hypothesis. (This is a good thing.)

Null Model

We can also use deviance to determine if our fitted model is better than the null model. The null model is a model with only a linear term. Like above, we can design a hypothesis test comparing the null model to the fitted model.

$$H_0 = \text{the fitted model is equivalent to the null model}$$
$$H_\alpha = \text{the fitted model is not equivalent to the null model}$$

In the limit of large data, it is known that the deviance follows a chi-squared distribution with parameter $n - 1$.

`logit$deviance` is the deviance between saturated model and fitted model. `logit$df.residual` is equal to number of observations minus the number of coefficients in the fitted model.

`logit$null.deviance` is the deviance between saturated model and the null model. `logit$df.null` is the number of observations minus 1.

Using this information, we can calculate the p value for the hypothesis test above.

```
p_value = pchisq(logit$null.deviance-logit$deviance,
                 logit$df.null-logit$df.residual, lower.tail = F)
print(p_value)
```

```
## [1] 1.173879e-14
```

Since the p value is less than one, we reject our null hypothesis. (This is a good thing.)

Anova

Sequential comparison of model terms by deviance

```
anova(logit,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Sepal.Width_binary
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL              149      199.22
## Species           2    51.709      147    147.51 5.910e-12 ***
## Sepal.Length      1    16.239      146    131.27 5.583e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Poisson Regression

```
attach(colon)
summary(colon)
```

```
##          id          study          rx          sex          age
## Min.      : 1    Min.      :1    Obs       :630    Min.      :0.000    Min.      :18.00
## 1st Qu.:233    1st Qu.:1    Lev       :620    1st Qu.:0.000    1st Qu.:53.00
## Median :465    Median :1    Lev+5FU:608    Median :1.000    Median :61.00
## Mean      :465    Mean      :1                Mean      :0.521    Mean      :59.75
## 3rd Qu.:697    3rd Qu.:1                3rd Qu.:1.000    3rd Qu.:69.00
## Max.      :929    Max.      :1                Max.      :1.000    Max.      :85.00
##
##          obstruct          perfor          adhere          nodes
## Min.      :0.0000    Min.      :0.00000    Min.      :0.0000    Min.      : 0.00
## 1st Qu.:0.0000    1st Qu.:0.00000    1st Qu.:0.0000    1st Qu.: 1.00
## Median :0.0000    Median :0.00000    Median :0.0000    Median : 2.00
## Mean      :0.1938    Mean      :0.02906    Mean      :0.1453    Mean      : 3.66
## 3rd Qu.:0.0000    3rd Qu.:0.00000    3rd Qu.:0.0000    3rd Qu.: 5.00
## Max.      :1.0000    Max.      :1.00000    Max.      :1.0000    Max.      :33.00
##                                     NA's      :36
##          status          differ          extent          surg
## Min.      :0.0000    Min.      :1.000    Min.      :1.000    Min.      :0.0000
## 1st Qu.:0.0000    1st Qu.:2.000    1st Qu.:3.000    1st Qu.:0.0000
## Median :0.0000    Median :2.000    Median :3.000    Median :0.0000
## Mean      :0.4952    Mean      :2.063    Mean      :2.887    Mean      :0.2659
## 3rd Qu.:1.0000    3rd Qu.:2.000    3rd Qu.:3.000    3rd Qu.:1.0000
## Max.      :1.0000    Max.      :3.000    Max.      :4.000    Max.      :1.0000
##                                     NA's      :46
##          node4          time          etype
## Min.      :0.0000    Min.      : 8    Min.      :1.0
## 1st Qu.:0.0000    1st Qu.: 566    1st Qu.:1.0
## Median :0.0000    Median :1855    Median :1.5
## Mean      :0.2745    Mean      :1538    Mean      :1.5
## 3rd Qu.:1.0000    3rd Qu.:2331    3rd Qu.:2.0
## Max.      :1.0000    Max.      :3329    Max.      :2.0
##
```

```
colon_data <- na.omit(colon)
sapply(colon_data, class)
```

```
##          id          study          rx          sex          age obstruct perfor
## "numeric" "numeric" "factor" "numeric" "numeric" "numeric" "numeric"
##          adhere          nodes          status          differ          extent          surg          node4
## "numeric" "numeric" "numeric" "numeric" "numeric" "numeric" "numeric"
##          time          etype
## "numeric" "numeric"
```

```
colon_data$age.ds = sapply(colon_data$age,
                           function(x) ifelse(x > median(age), 1, 0))
colon_data$age.ds <- factor(colon_data$age.ds,
                           levels= c("0","1"),
                           labels=c("<median",">median"))
```

```

colon_data$node4 <- factor(colon_data$node4,
                           levels= c("0","1"),
                           labels=c("<4",">4"))

colon_data$sex <- factor(colon_data$sex,
                         levels= c("0","1"), labels=c("F","M"))

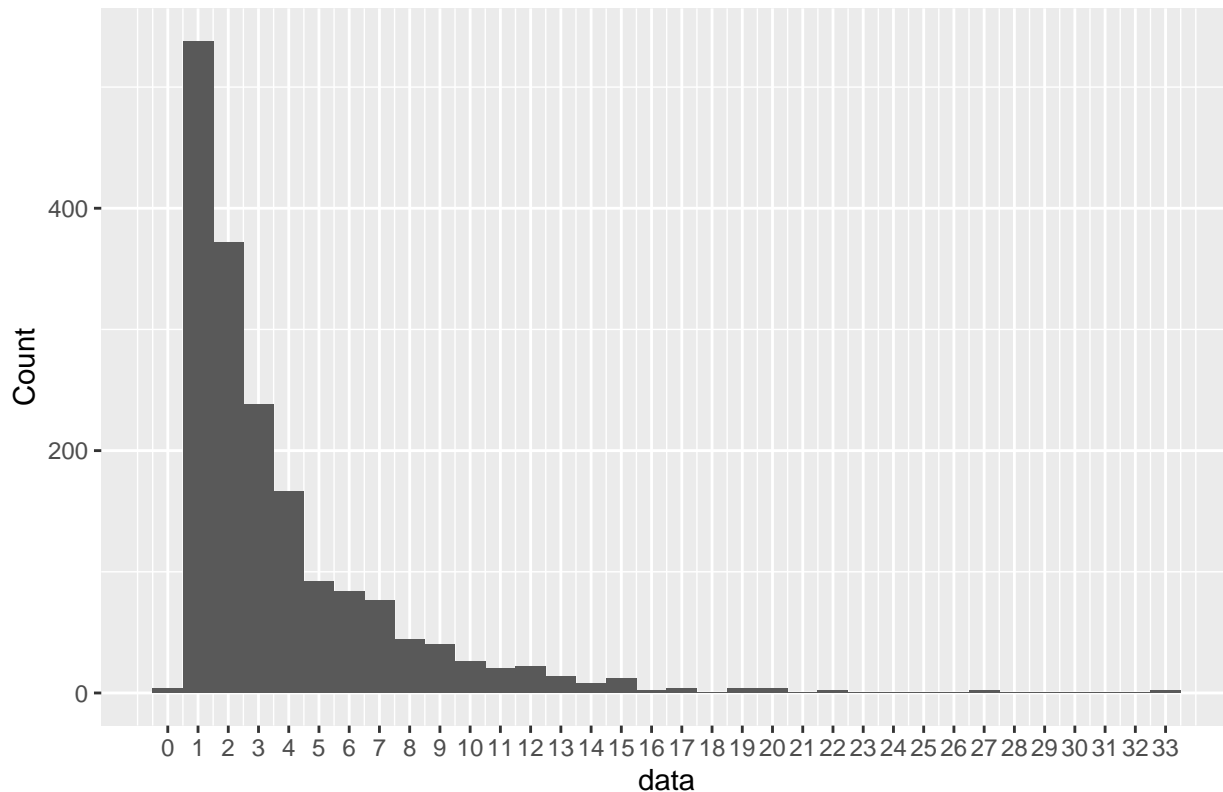
colon_data$obstruct <- factor(colon_data$obstruct,
                              levels= c("0","1"),
                              labels=c("no obstruct","obstruct"))
colon_data$adhere <- factor(colon_data$adhere,
                            levels= c("0","1"),
                            labels=c("no adhere","adhere"))
colon_data$perfor <- factor(colon_data$perfor,
                            levels= c("0","1"),
                            labels=c("no perfor","perfor"))

colon_data$differ <- factor(colon_data$differ,
                            levels= c("1","2","3"),
                            labels=c("well","mod","poor"))
colon_data$extent <- factor(colon_data$extent,
                            levels= c("1","2","3","4"),
                            labels=c("submucosa", "muscle", "serosa", "contiguous"))
colon_data$surg <- factor(colon_data$surg,
                          levels= c("0","1"),
                          labels=c("short","long"))

ggplot(colon_data,aes(x=colon_data$nodes))+
  geom_histogram(binwidth = 1, center = 1) +
  scale_x_continuous(breaks=seq(0,max(colon_data$nodes), by = 1))+
  ylab("Count")+ xlab("data")+
  ggtitle("Histogram plot of the number of epileptic seizures")

```

Histogram plot of the number of epileptic seizures



```
poisson_model = glm(nodes ~ 1, family=poisson(link=log),data=colon_data)
summary(poisson_model)
```

```
##
## Call:
## glm(formula = nodes ~ 1, family = poisson(link = log), data = colon_data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.7068  -1.6522  -0.9517   0.6613   9.2954
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.2984     0.0124   104.7  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 4593.4  on 1775  degrees of freedom
## Residual deviance: 4593.4  on 1775  degrees of freedom
## AIC: 9688.3
##
## Number of Fisher Scoring iterations: 5
```

```
print(coef(poisson_model))
```

```
## (Intercept)
```

```
##      1.298361
head(data.frame(colon_data$nodes,poisson_model$fitted))
```

```
##      colon_data.nodes poisson_model.fitted
## 1           5           3.663288
## 2           5           3.663288
## 3           1           3.663288
## 4           1           3.663288
## 5           7           3.663288
## 6           7           3.663288
```

```
head(poisson_model$linear.predictors)
```

```
##           1           2           3           4           5           6
## 1.298361 1.298361 1.298361 1.298361 1.298361 1.298361
```

```
head(exp(poisson_model$linear.predictors))
```

```
##           1           2           3           4           5           6
## 3.663288 3.663288 3.663288 3.663288 3.663288 3.663288
```

Hypothesis test for goodness of fit

```
p_value = pchisq(poisson_model$deviance,
                 poisson_model$df.residual, lower.tail = F)
print(p_value)
```

```
## [1] 2.524571e-248
```

```
p_value = pchisq(poisson_model$null.deviance-logit$deviance,
                 poisson_model$df.null-logit$df.residual, lower.tail = F)
print(p_value)
```

```
## [1] 1.395921e-261
```

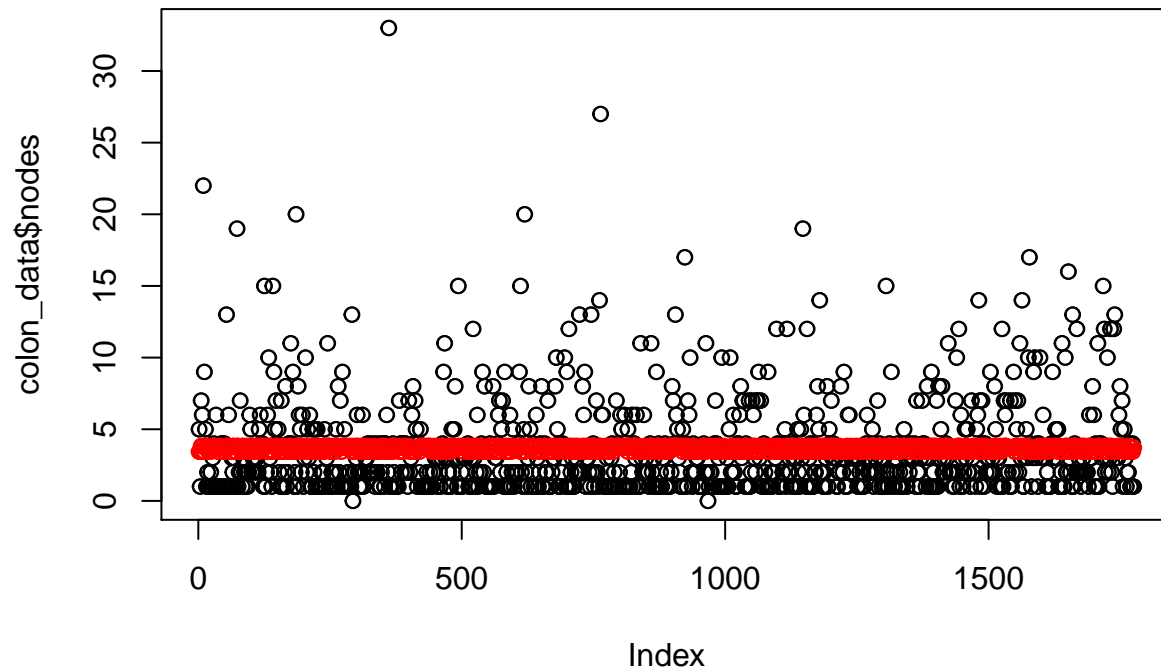
add a Covariate to the fit – treatment

```
poisson_model = glm(nodes~ 1 + rx, family=poisson(link=log),data=colon_data)
summary(poisson_model)
```

```
##
## Call:
## glm(formula = nodes ~ 1 + rx, family = poisson(link = log), data = colon_data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.7687  -1.5503  -0.8410   0.5692   9.2546
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.34359    0.02068  64.967 < 2e-16 ***
## rxLev       -0.03232    0.02977  -1.086  0.277536
## rxLev+5FU   -0.10929    0.03052  -3.581  0.000342 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 4593.4  on 1775  degrees of freedom
## Residual deviance: 4580.0  on 1773  degrees of freedom
## AIC: 9678.9
##
## Number of Fisher Scoring iterations: 5
```

```
plot(colon_data$nodes)
points(poisson_model$fitted,col="red")
```



Hypothesis test for goodness of fit

```
p_value = pchisq(poisson_model$deviance,
                  poisson_model$df.residual, lower.tail = F)
print(p_value)
```

```
## [1] 5.963521e-247
```

```
p_value = pchisq(poisson_model$null.deviance-logit$deviance,
                  poisson_model$df.null-logit$df.residual, lower.tail = F)
print(p_value)
```

```
## [1] 1.395921e-261
```

```
anova(poisson_model,test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
```

```
## Response: nodes
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                1775      4593.4
## rx      2      13.4      1773      4580.0 0.001231 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

add a Covariate to the fit – treatment, time

```
poisson_model = glm(nodes ~ 1 + rx + time,
                     family=poisson(link=log), data=colon_data)
summary(poisson_model)
```

```
##
## Call:
## glm(formula = nodes ~ 1 + rx + time, family = poisson(link = log),
##      data = colon_data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.2708  -1.2451  -0.5310   0.5057   9.0381
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.7292541  0.0260370  66.415  <2e-16 ***
## rxLev        -0.0235541  0.0297671  -0.791    0.429
## rxLev+5FU     -0.0287557  0.0307216  -0.936    0.349
## time         -0.0002937  0.0000133 -22.082  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 4593.4  on 1775  degrees of freedom
## Residual deviance: 4085.4  on 1772  degrees of freedom
## AIC: 9186.4
##
## Number of Fisher Scoring iterations: 5
```

Hypothesis test for goodness of fit

```
p_value = pchisq(poisson_model$deviance,
                  poisson_model$df.residual, lower.tail = F)
print(p_value)
```

```
## [1] 1.192842e-183
```

```
p_value = pchisq(poisson_model$null.deviance-logit$deviance,
                 poisson_model$df.null-logit$df.residual, lower.tail = F)
print(p_value)
```

```
## [1] 1.395921e-261
```

```
anova(poisson_model,test="Chisq")
```

```
## Analysis of Deviance Table
```

```
##
```

```
## Model: poisson, link: log
```

```
##
```

```
## Response: nodes
```

```
##
```

```
## Terms added sequentially (first to last)
```

```
##
```

```
##
```

```
##      Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
```

```
## NULL                1775      4593.4
```

```
## rx      2      13.40      1773      4580.0 0.001231 **
```

```
## time    1     494.57      1772      4085.4 < 2.2e-16 ***
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

add a Covariate to the fit – treatment, time interaction

```
poisson_model = glm(nodes ~ 1 + rx*time,
                    family=poisson(link=log),data=colon_data)
summary(poisson_model)
```

```
##
```

```
## Call:
```

```
## glm(formula = nodes ~ 1 + rx * time, family = poisson(link = log),
```

```
##      data = colon_data)
```

```
##
```

```
## Deviance Residuals:
```

```
##      Min       1Q   Median       3Q      Max
```

```
## -3.2059  -1.2393  -0.5101   0.5019   9.0511
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept)  1.834e+00  3.320e-02  55.237 < 2e-16 ***
```

```
## rxLev        -1.715e-01  4.844e-02  -3.540 0.000400 ***
```

```
## rxLev+5FU    -2.168e-01  5.360e-02  -4.046 5.22e-05 ***
```

```
## time         -3.842e-04  2.326e-05 -16.520 < 2e-16 ***
```

```
## rxLev:time    1.261e-04  3.242e-05   3.889 0.000101 ***
```

```
## rxLev+5FU:time 1.470e-04  3.315e-05   4.436 9.17e-06 ***
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## (Dispersion parameter for poisson family taken to be 1)
```

```
##
```

```
##      Null deviance: 4593.4  on 1775  degrees of freedom
```



```

## Residual deviance: 4061.8  on 1770  degrees of freedom
## AIC: 9166.8
##
## Number of Fisher Scoring iterations: 5
p_value = pchisq(poisson_model$deviance,
                 poisson_model$df.residual, lower.tail = F)
print(p_value)

## [1] 4.076866e-181
p_value = pchisq(poisson_model$null.deviance-logit$deviance,
                 poisson_model$df.null-logit$df.residual, lower.tail = F)
print(p_value)

## [1] 1.395921e-261
anova(poisson_model,test="Chisq")

## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: nodes
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                1775     4593.4
## rx          2      13.40      1773     4580.0 0.001231 **
## time        1     494.57      1772     4085.4 < 2.2e-16 ***
## rx:time     2      23.56      1770     4061.8 7.66e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

Log-Linear Regression

Log-linear models allow us to model association between two or more variables in contingency table. In a log-linear model, there are no well defined explanatory/response variables. This is because we are focused more on the *interaction* between two variables.

Contingency Table

Contingency table displays number of observations for a given combination of factors.

This definition is best represented by an example.

```
epil$y_binary <- sapply(epil$y,function(x) ifelse(x > median(epil$y), 1, 0))
```

One-Way Contingency Table

```
table(y_relative_median=epil$y_binary)
```

```
## y_relative_median
##      0      1
## 121 115
```

Two-Way Contingency Table

```
table(y_relative_median=epil$y_binary,period=epil$period)
```

```
##           period
## y_relative_median  1  2  3  4
##           0 30 29 32 30
##           1 29 30 27 29
```

Independent Model

```
contingency_table = table(y_relative_median=epil$y_binary,period=epil$period)
```

Need to convert contingency table in a form that is acceptable to glm

```
contingency_table.df = as.data.frame(contingency_table)
```

```
log_linear_model <- glm(Freq ~ y_relative_median + period,
                        data = contingency_table.df, family = poisson)
```

```
summary(log_linear_model)
```

```
##
## Call:
## glm(formula = Freq ~ y_relative_median + period, family = poisson,
##      data = contingency_table.df)
##
## Deviance Residuals:
##      1      2      3      4      5      6      7
## -0.04552  0.04656 -0.22887  0.23147  0.31519 -0.32977 -0.04552
```

```
##          8
## 0.04656
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    3.409e+00  1.448e-01  23.541  <2e-16 ***
## y_relative_median1 -5.086e-02  1.302e-01  -0.391   0.696
## period2         8.318e-14  1.841e-01   0.000   1.000
## period3         1.189e-12  1.841e-01   0.000   1.000
## period4         5.059e-16  1.841e-01   0.000   1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
## Null deviance: 0.47509  on 7  degrees of freedom
## Residual deviance: 0.32253  on 3  degrees of freedom
## AIC: 52.138
##
## Number of Fisher Scoring iterations: 3
```

Saturated Model

```
log_linear_model <- glm(Freq ~ period*y_relative_median,
  data = contingency_table.df, family = poisson)
```

```
summary(log_linear_model)
```

```
##
## Call:
## glm(formula = Freq ~ period * y_relative_median, family = poisson,
## data = contingency_table.df)
##
## Deviance Residuals:
## [1]  0  0  0  0  0  0  0  0  0
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    3.401e+00  1.826e-01  18.629  <2e-16 ***
## period2       -3.390e-02  2.604e-01  -0.130   0.896
## period3        6.454e-02  2.541e-01   0.254   0.800
## period4       -4.774e-16  2.582e-01   0.000   1.000
## y_relative_median1 -3.390e-02  2.604e-01  -0.130   0.896
## period2:y_relative_median1  6.780e-02  3.683e-01   0.184   0.854
## period3:y_relative_median1 -1.360e-01  3.689e-01  -0.369   0.712
## period4:y_relative_median1 -2.044e-16  3.683e-01   0.000   1.000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
## Null deviance: 4.7509e-01  on 7  degrees of freedom
## Residual deviance: 1.9540e-14  on 0  degrees of freedom
```

```
## AIC: 57.815
##
## Number of Fisher Scoring iterations: 3
anova(log_linear_model, test='Chisq')

## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: Freq
##
## Terms added sequentially (first to last)
##
##
##              Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL              7    0.47509
## period            3  0.00000      4    0.47509  1.0000
## y_relative_median  1  0.15256      3    0.32253  0.6961
## period:y_relative_median 3  0.32253      0    0.00000  0.9557
```

Three-Way Contingency Table

```
table(y_relative_median=epil$y_binary, period=epil$period, trt=epil$trt)

## , , trt = placebo
##
##           period
## y_relative_median  1  2  3  4
##           0 12 14 14 12
##           1 16 14 14 16
##
## , , trt = progabide
##
##           period
## y_relative_median  1  2  3  4
##           0 18 15 18 18
##           1 13 16 13 13
```

Independent Model

```
contingency_table = table(y_relative_median=epil$y_binary,
                          period=epil$period, trt=epil$trt)

Need to convert contingency table in a form that is acceptable to glm
contingency_table.df = as.data.frame(contingency_table)

log_linear_model <- glm(Freq ~ y_relative_median + period + trt,
                        data = contingency_table.df, family = poisson)

summary(log_linear_model)

##
```

```
## Call:
## glm(formula = Freq ~ y_relative_median + period + trt, family = poisson,
##      data = contingency_table.df)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.64008  -0.55522   0.00081   0.51717   0.62066
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    2.664e+00  1.602e-01  16.629  <2e-16 ***
## y_relative_median1 -5.086e-02  1.302e-01  -0.391   0.696
## period2         3.309e-17  1.841e-01   0.000   1.000
## period3        -2.685e-16  1.841e-01   0.000   1.000
## period4        -2.003e-16  1.841e-01   0.000   1.000
## trtprogabide     1.018e-01  1.304e-01   0.781   0.435
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 4.2194  on 15  degrees of freedom
## Residual deviance: 3.4564  on 10  degrees of freedom
## AIC: 87.964
##
## Number of Fisher Scoring iterations: 4
```

Saturated Model

```
log_linear_model <- glm(Freq ~ period*y_relative_median*trt,
                        data = contingency_table.df, family = poisson)
```

```
summary(log_linear_model)
```

```
##
## Call:
## glm(formula = Freq ~ period * y_relative_median * trt, family = poisson,
##      data = contingency_table.df)
##
## Deviance Residuals:
## [1]  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
##
## Coefficients:
##              Estimate Std. Error z value
## (Intercept)    2.485e+00  2.887e-01   8.608
## period2        1.542e-01  3.934e-01   0.392
## period3        1.542e-01  3.934e-01   0.392
## period4       -2.317e-15  4.082e-01   0.000
## y_relative_median1  2.877e-01  3.819e-01   0.753
## trtprogabide     4.055e-01  3.727e-01   1.088
## period2:y_relative_median1 -2.877e-01  5.373e-01  -0.535
## period3:y_relative_median1 -2.877e-01  5.373e-01  -0.535
## period4:y_relative_median1  2.385e-15  5.401e-01   0.000
```

```

## period2:trtprogabide -3.365e-01 5.263e-01 -0.639
## period3:trtprogabide -1.542e-01 5.156e-01 -0.299
## period4:trtprogabide 2.112e-15 5.270e-01 0.000
## y_relative_median1:trtprogabide -6.131e-01 5.276e-01 -1.162
## period2:y_relative_median1:trtprogabide 6.776e-01 7.418e-01 0.913
## period3:y_relative_median1:trtprogabide 2.877e-01 7.441e-01 0.387
## period4:y_relative_median1:trtprogabide -2.299e-15 7.461e-01 0.000
## Pr(>|z|)
## (Intercept) <2e-16 ***
## period2 0.695
## period3 0.695
## period4 1.000
## y_relative_median1 0.451
## trtprogabide 0.277
## period2:y_relative_median1 0.592
## period3:y_relative_median1 0.592
## period4:y_relative_median1 1.000
## period2:trtprogabide 0.523
## period3:trtprogabide 0.765
## period4:trtprogabide 1.000
## y_relative_median1:trtprogabide 0.245
## period2:y_relative_median1:trtprogabide 0.361
## period3:y_relative_median1:trtprogabide 0.699
## period4:y_relative_median1:trtprogabide 1.000
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
## Null deviance: 4.2194e+00 on 15 degrees of freedom
## Residual deviance: 7.9936e-15 on 0 degrees of freedom
## AIC: 104.51
##
## Number of Fisher Scoring iterations: 3
anova(log_linear_model,test='Chisq')

## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: Freq
##
## Terms added sequentially (first to last)
##
##
## Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL 15 4.2194
## period 3 0.00000 12 4.2194 1.0000
## y_relative_median 1 0.15256 11 4.0669 0.6961
## trt 1 0.61043 10 3.4564 0.4346
## period:y_relative_median 3 0.32253 7 3.1339 0.9557
## period:trt 3 0.00000 4 3.1339 1.0000
## y_relative_median:trt 1 2.00614 3 1.1278 0.1567
## period:y_relative_median:trt 3 1.12777 0 0.0000 0.7704

```

Hierarchical modeling

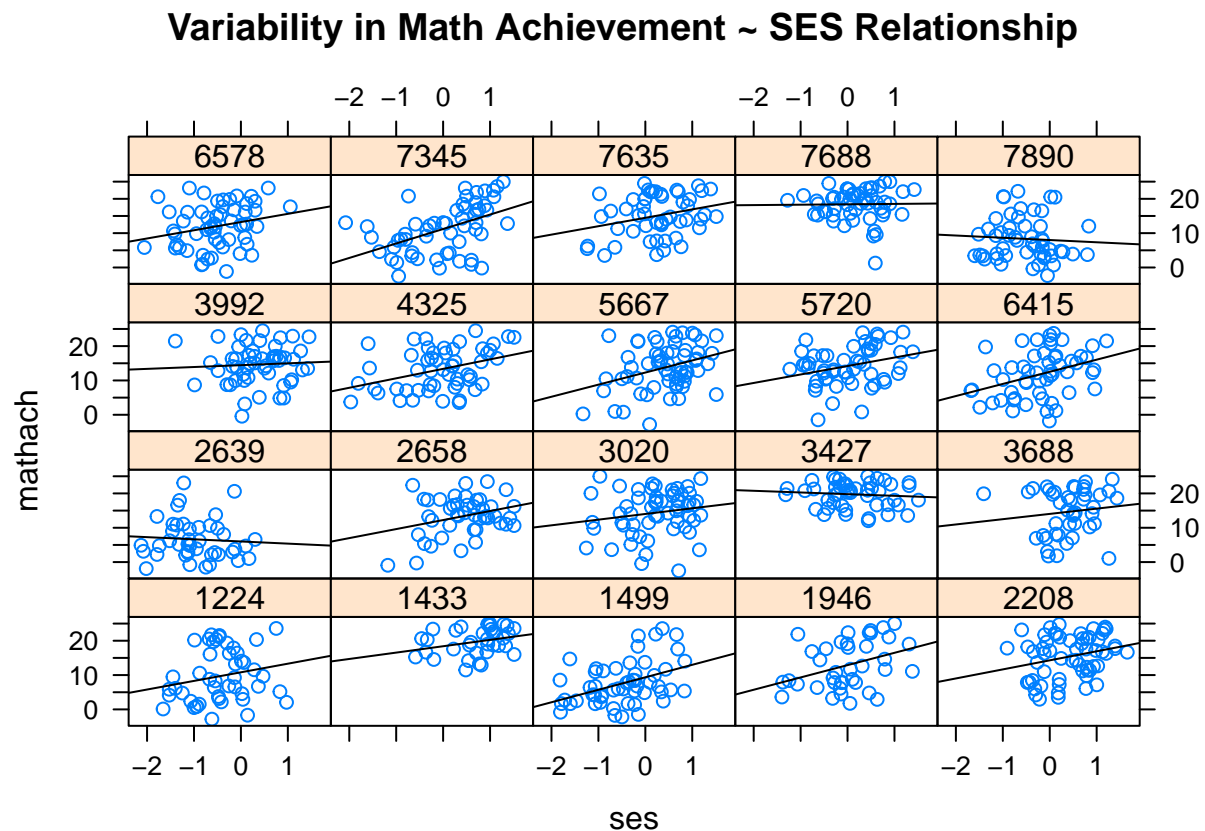
```
student_data <- read.csv("data/hsb1.csv")
school_data <- read.csv("data/hsb2.csv")
student_data$ses_grandmean <- student_data$ses - mean(student_data$ses) # Grand-mean centered student SES
school_data$sm_ses_grandmean <- school_data$meanses - mean(school_data$meanses) # Grand-mean centered school SES

data <- merge(student_data, school_data, by = "id")

ses_group_mean <- aggregate(data$ses, list(data$id), FUN = mean, data = data) # Group-mean centered student SES
names(ses_group_mean) <- c('id', 'groupmeanSES')
data <- merge(data, ses_group_mean, by = "id")

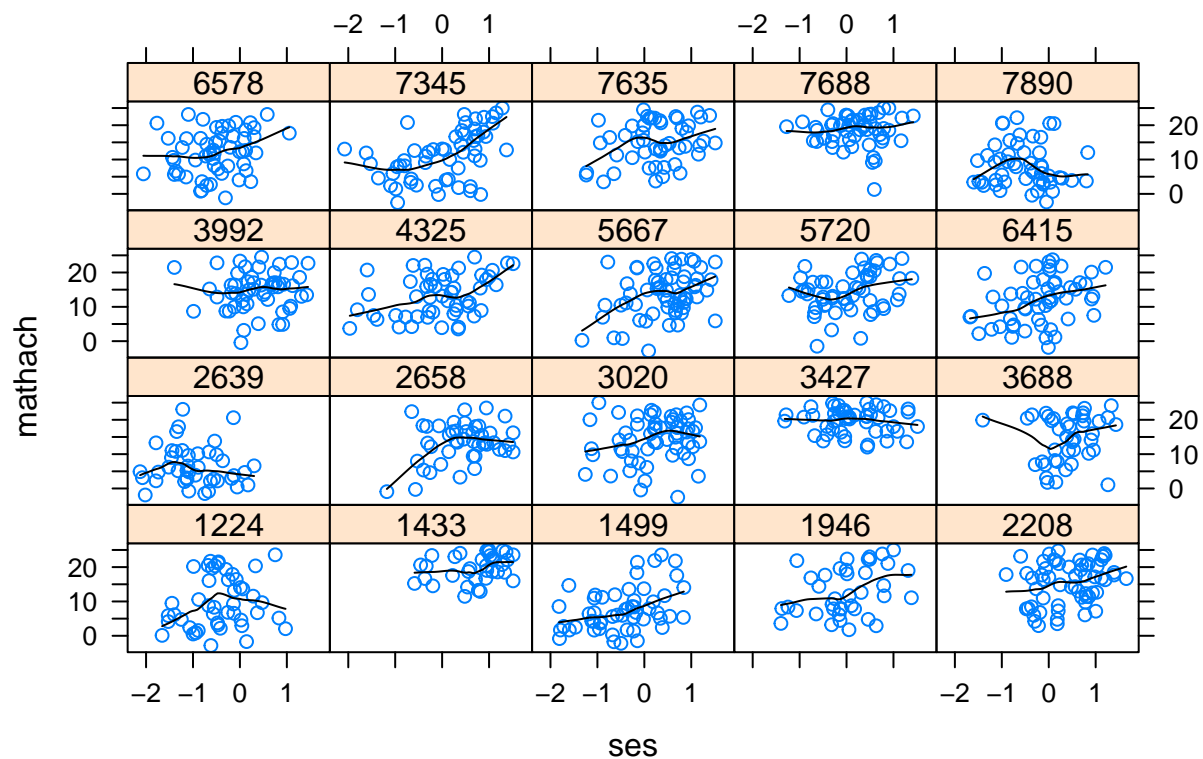
groups <- unique(data$id)[sample(1:160, 20)]
subset <- data[data$id %in% groups, ]

xyplot(mathach ~ ses | as.factor(id), subset,
       col.line = 'black',
       type = c("p", "r"),
       main = 'Variability in Math Achievement ~ SES Relationship')
```



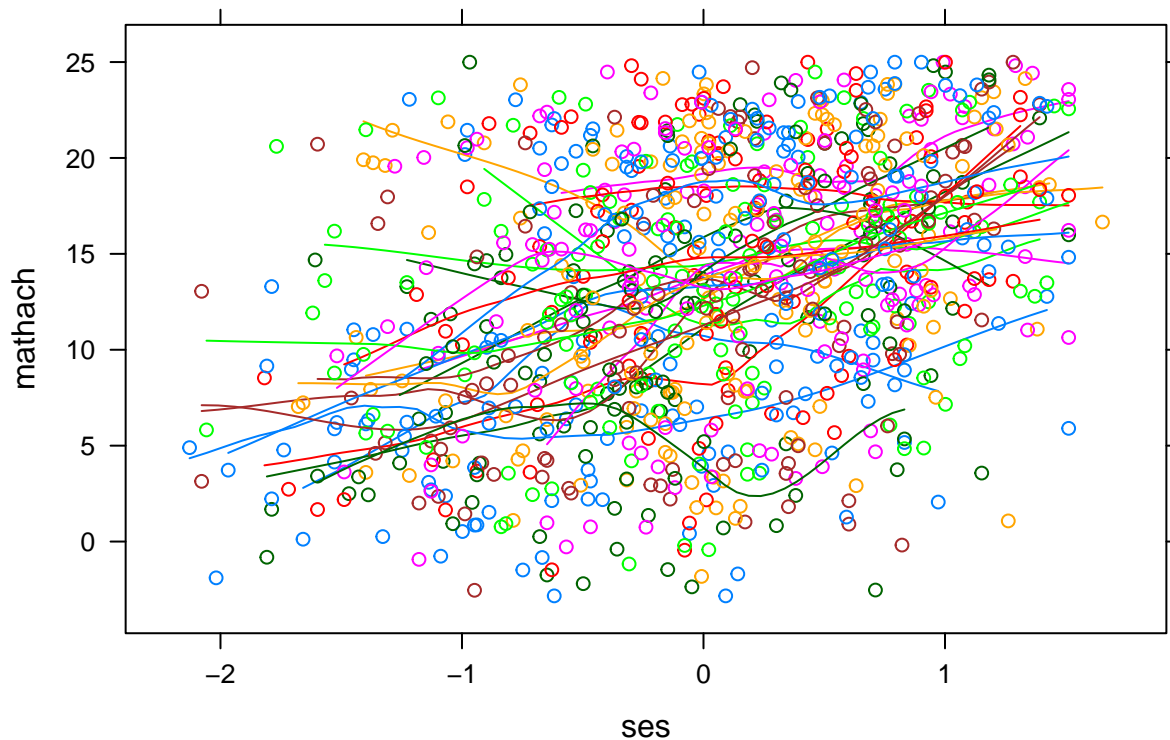
```
xyplot(mathach ~ ses | as.factor(id), subset,
       col.line = 'black',
       type = c("p", "smooth"),
       main = 'Variability in Math Achievement ~ SES Relationship')
```

Variability in Math Achievement ~ SES Relationship



```
xyplot(mathach ~ ses, subset,
       type = c("p", "smooth"),
       group = data$id,
       main = 'Variability in Math Achievement ~ SES Relationship')
```


Variability in Math Achievement ~ SES Relationship



```
unconditional <- lmer(mathach ~ 1 + (1|id), data = data)
summary(unconditional) # on p-values in nlme: https://stat.ethz.ch/pipermail/r-help/2006-May/094765.htm

## Linear mixed model fit by REML ['lmerMod']
## Formula: mathach ~ 1 + (1 | id)
## Data: data
##
## REML criterion at convergence: 47116.8
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.0631 -0.7539  0.0267  0.7606  2.7426
##
## Random effects:
## Groups Name Variance Std.Dev.
## id      (Intercept) 8.614  2.935
## Residual          39.148  6.257
## Number of obs: 7185, groups: id, 160
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept) 12.6370    0.2444   51.71
confint(unconditional) # you can also just calculate an approximate 95% confidence interval yourself: e

## Computing profile confidence intervals ...

##              2.5 %    97.5 %
## .sig01         2.594729  3.315880
```

```

## .sigma      6.154803  6.361786
## (Intercept) 12.156289 13.117121

unconditional_2 <- lme(mathach ~ 1, random = ~ 1 | id, data = data)
summary(unconditional_2)

## Linear mixed-effects model fit by REML
## Data: data
##      AIC      BIC    logLik
## 47122.79 47143.43 -23558.4
##
## Random effects:
## Formula: ~1 | id
##      (Intercept) Residual
## StdDev:      2.934966 6.256862
##
## Fixed effects: mathach ~ 1
##              Value Std.Error   DF  t-value p-value
## (Intercept) 12.63697 0.2443936 7025 51.70747      0
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -3.06312473 -0.75387398  0.02670132  0.76062171  2.74262579
##
## Number of Observations: 7185
## Number of Groups: 160

random_intercept_fixed_slope <- lmer(mathach ~ 1 + groupmeanSES + (1|id), data = data)
summary(random_intercept_fixed_slope)

## Linear mixed model fit by REML ['lmerMod']
## Formula: mathach ~ 1 + groupmeanSES + (1 | id)
## Data: data
##
## REML criterion at convergence: 46961.3
##
## Scaled residuals:
##      Min      1Q   Median      3Q      Max
## -3.13493 -0.75254  0.02413  0.76766  2.78515
##
## Random effects:
## Groups   Name      Variance Std.Dev.
## id      (Intercept)  2.639    1.624
## Residual                39.157    6.258
## Number of obs: 7185, groups: id, 160
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  12.6846    0.1493   84.97
## groupmeanSES  5.8635    0.3615   16.22
##
## Correlation of Fixed Effects:
##              (Intr)
## groupmenSES 0.010

```

```

confint(random_intercept_fixed_slope)

## Computing profile confidence intervals ...
##           2.5 %    97.5 %
## .sig01      1.385193  1.871127
## .sigma      6.155502  6.362511
## (Intercept) 12.391774 12.976903
## groupmeanSES 5.155743 6.572440

random_intercept_fixed_slope_2 <- lme(mathach ~ 1 + groupmeanSES, random = ~ 1 | id, data = data)
summary(random_intercept_fixed_slope_2)

## Linear mixed-effects model fit by REML
## Data: data
##      AIC      BIC    logLik
## 46969.29 46996.81 -23480.65
##
## Random effects:
## Formula: ~1 | id
##      (Intercept) Residual
## StdDev:      1.624462 6.257562
##
## Fixed effects: mathach ~ 1 + groupmeanSES
##              Value Std.Error   DF  t-value p-value
## (Intercept) 12.684609 0.1492900 7025 84.96624      0
## groupmeanSES 5.863539 0.3614712  158 16.22132      0
## Correlation:
##      (Intr)
## groupmeanSES 0.01
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -3.13493066 -0.75254260  0.02413095  0.76766113  2.78515398
##
## Number of Observations: 7185
## Number of Groups: 160

random_intercept_random_slope <- lmer(mathach ~ 1 + groupmeanSES + (1 + groupmeanSES|id), data = data)

## boundary (singular) fit: see ?isSingular

summary(random_intercept_random_slope)

## Linear mixed model fit by REML ['lmerMod']
## Formula: mathach ~ 1 + groupmeanSES + (1 + groupmeanSES | id)
## Data: data
##
## REML criterion at convergence: 46960.9
##
## Scaled residuals:
##      Min      1Q  Median      3Q      Max
## -3.13245 -0.75164  0.02212  0.76876  2.79449
##
## Random effects:
## Groups   Name              Variance Std.Dev. Corr

```

```

## id      (Intercept)  2.62707 1.6208
##      groupmeanSES  0.05417 0.2327  -1.00
## Residual              39.15798 6.2576
## Number of obs: 7185, groups: id, 160
##
## Fixed effects:
##      Estimate Std. Error t value
## (Intercept)  12.6832    0.1491  85.04
## groupmeanSES  5.8379    0.3644  16.02
##
## Correlation of Fixed Effects:
##      (Intr)
## groupmenSES -0.078
## convergence code: 0
## boundary (singular) fit: see ?isSingular
random_intercept_random_slope_2 <- lme(mathach ~ 1 + groupmeanSES, random = ~ 1 + groupmeanSES | id, da
summary(random_intercept_random_slope_2)

## Linear mixed-effects model fit by REML
## Data: data
##      AIC      BIC    logLik
## 46973.29 47014.57 -23480.65
##
## Random effects:
## Formula: ~1 + groupmeanSES | id
## Structure: General positive-definite, Log-Cholesky parametrization
##      StdDev      Corr
## (Intercept) 1.624460932 (Intr)
## groupmeanSES 0.008272356 -0.003
## Residual    6.257561467
##
## Fixed effects: mathach ~ 1 + groupmeanSES
##      Value Std.Error DF t-value p-value
## (Intercept) 12.684610 0.1492901 7025 84.96616 0
## groupmeanSES 5.863533 0.3614729 158 16.22122 0
## Correlation:
##      (Intr)
## groupmeanSES 0.01
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -3.13493049 -0.75254293 0.02413128 0.76766157 2.78515572
##
## Number of Observations: 7185
## Number of Groups: 160
fixed_intercept_random_slope <- lmer(mathach ~ 1 + groupmeanSES + (0 + groupmeanSES|id), data = data)
summary(fixed_intercept_random_slope)

## Linear mixed model fit by REML ['lmerMod']
## Formula: mathach ~ 1 + groupmeanSES + (0 + groupmeanSES | id)
## Data: data
##
## REML criterion at convergence: 47065

```

```

##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.1150 -0.7431  0.0317  0.7651  2.8202
##
## Random effects:
##   Groups   Name                Variance Std.Dev.
##    id      groupmeanSES 27.05      5.201
## Residual                39.75      6.304
## Number of obs: 7185, groups: id, 160
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)  12.7640     0.1226  104.07
## groupmeanSES  5.4202     0.5271   10.28
##
## Correlation of Fixed Effects:
##              (Intr)
## groupmenSES -0.045
fixed_intercept_random_slope_2 <- lme(mathach ~ 1 + groupmeanSES, random = ~ 0 + groupmeanSES | id, data = data,
summary(fixed_intercept_random_slope_2)

## Linear mixed-effects model fit by REML
## Data: data
##      AIC      BIC    logLik
## 47072.99 47100.51 -23532.5
##
## Random effects:
## Formula: ~0 + groupmeanSES | id
##      groupmeanSES Residual
## StdDev:      5.201045 6.304462
##
## Fixed effects: mathach ~ 1 + groupmeanSES
##              Value Std.Error   DF   t-value p-value
## (Intercept) 12.764014 0.1226493 7025 104.06918      0
## groupmeanSES 5.420157 0.5270957 158 10.28306      0
## Correlation:
##              (Intr)
## groupmeanSES -0.045
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -3.11504273 -0.74308714  0.03169931  0.76511017  2.82021818
##
## Number of Observations: 7185
## Number of Groups: 160
fixed_slope_level_two_variable <- lmer(mathach ~ 1 + groupmeanSES + sm_ses_grandmean + (1|id), data = data,
summary(fixed_slope_level_two_variable)

## Linear mixed model fit by REML ['lmerMod']
## Formula: mathach ~ 1 + groupmeanSES + sm_ses_grandmean + (1 | id)
## Data: data
##

```

```

## REML criterion at convergence: 46946.8
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.13127 -0.75215  0.02439  0.76700  2.78177
##
## Random effects:
##   Groups   Name                Variance Std.Dev.
##   id       (Intercept)    2.659     1.631
##   Residual                    39.157     6.258
## Number of obs: 7185, groups: id, 160
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)      11.675      3.299   3.539
## groupmeanSES    -157.361    532.675  -0.295
## sm_ses_grandmean  163.223    532.668   0.306
##
## Correlation of Fixed Effects:
##              (Intr) grpSES
## groupmenSES   0.999
## sm_ss_grndm  -0.999 -1.000
fixed_slope_level_two_variable_2 <- lme(mathach ~ 1 + groupmeanSES + sm_ses_grandmean, random = ~ 1 | id, data = data)
summary(fixed_slope_level_two_variable_2)

## Linear mixed-effects model fit by REML
## Data: data
##      AIC      BIC    logLik
## 46956.81 46991.2 -23473.4
##
## Random effects:
## Formula: ~1 | id
##      (Intercept) Residual
## StdDev:      1.630771 6.257562
##
## Fixed effects: mathach ~ 1 + groupmeanSES + sm_ses_grandmean
##              Value Std.Error   DF   t-value p-value
## (Intercept)    11.67469    3.2988 7025   3.539111  0.0004
## groupmeanSES   -157.36077   532.6748  157  -0.295416  0.7681
## sm_ses_grandmean 163.22262   532.6683  157   0.306425  0.7597
## Correlation:
##              (Intr) grpSES
## groupmeanSES    0.999
## sm_ses_grandmean -0.999 -1.000
##
## Standardized Within-Group Residuals:
##      Min       Q1       Med       Q3      Max
## -3.13126623 -0.75215319  0.02439264  0.76699775  2.78176653
##
## Number of Observations: 7185
## Number of Groups: 160

```

```

random_slope_level_two_variable <- lmer(mathach ~ 1 + groupmeanSES + sm_ses_grandmean + (1 + groupmeanSES |
## boundary (singular) fit: see ?isSingular
summary(random_slope_level_two_variable)

## Linear mixed model fit by REML ['lmerMod']
## Formula:
## mathach ~ 1 + groupmeanSES + sm_ses_grandmean + (1 + groupmeanSES |
## id)
## Data: data
##
## REML criterion at convergence: 46946.3
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -3.12727 -0.74930  0.02286  0.76841  2.79122
##
## Random effects:
##   Groups   Name                Variance Std.Dev. Corr
##   id       (Intercept)         2.64688  1.6269
##           groupmeanSES        0.05901  0.2429   -1.00
##   Residual                    39.15801  6.2576
## Number of obs: 7185, groups: id, 160
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)      11.493      3.292   3.491
## groupmeanSES     -186.500     531.608  -0.351
## sm_ses_grandmean  192.333     531.597   0.362
##
## Correlation of Fixed Effects:
##              (Intr) grpSES
## groupmenSES   0.999
## sm_ss_grndm  -0.999 -1.000
## convergence code: 0
## boundary (singular) fit: see ?isSingular
random_slope_level_two_variable_2 <- lme(mathach ~ 1 + groupmeanSES + sm_ses_grandmean, random = ~ 1 +
summary(random_slope_level_two_variable_2)

## Linear mixed-effects model fit by REML
## Data: data
##      AIC      BIC    logLik
## 46960.81 47008.96 -23473.4
##
## Random effects:
## Formula: ~1 + groupmeanSES | id
## Structure: General positive-definite, Log-Cholesky parametrization
##              StdDev    Corr
## (Intercept)  1.6307657 (Intr)
## groupmeanSES 0.0130297 -0.005
## Residual     6.2575620
##
## Fixed effects: mathach ~ 1 + groupmeanSES + sm_ses_grandmean

```

```

##              Value Std.Error   DF   t-value p-value
## (Intercept)    11.67466    3.2988 7025  3.539092  0.0004
## groupmeanSES   -157.36607   532.6762  157 -0.295425  0.7681
## sm_ses_grandmean 163.22790   532.6697  157  0.306434  0.7597
## Correlation:
##              (Intr) grpSES
## groupmeanSES    0.999
## sm_ses_grandmean -0.999 -1.000
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -3.13126549 -0.75215462  0.02439169  0.76699807  2.78177069
##
## Number of Observations: 7185
## Number of Groups: 160

fixed_slope_cl_interaction <- lmer(mathach ~ 1 + groupmeanSES*sm_ses_grandmean + (1|id), data = data)
summary(fixed_slope_cl_interaction)

## Linear mixed model fit by REML ['lmerMod']
## Formula: mathach ~ 1 + groupmeanSES * sm_ses_grandmean + (1 | id)
## Data: data
##
## REML criterion at convergence: 46945
##
## Scaled residuals:
##      Min      1Q   Median      3Q      Max
## -3.11930 -0.75112  0.02448  0.76597  2.78831
##
## Random effects:
## Groups Name Variance Std.Dev.
## id      (Intercept) 2.664  1.632
## Residual          39.158  6.258
## Number of obs: 7185, groups: id, 160
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept)      11.4252    3.3155  3.446
## groupmeanSES     -213.6963   537.6427 -0.397
## sm_ses_grandmean  219.4884   537.6248  0.408
## groupmeanSES:sm_ses_grandmean -0.5799    0.7253 -0.800
##
## Correlation of Fixed Effects:
##              (Intr) grpSES sm_ss_
## groupmenSES  0.998
## sm_ss_grndm -0.998 -1.000
## grpmnSES:__  0.094  0.131 -0.131

fixed_slope_cl_interaction_2 <- lme(mathach ~ 1 + groupmeanSES*sm_ses_grandmean, random = ~ 1 | id, data = data)
summary(fixed_slope_cl_interaction_2)

## Linear mixed-effects model fit by REML
## Data: data
##      AIC      BIC    logLik
## 46956.97 46998.25 -23472.49

```



```
##
## Random effects:
## Formula: ~1 | id
## (Intercept) Residual
## StdDev: 1.632105 6.257638
##
## Fixed effects: mathach ~ 1 + groupmeanSES * sm_ses_grandmean
## Value Std.Error DF t-value p-value
## (Intercept) 11.42519 3.3155 7025 3.445968 0.0006
## groupmeanSES -213.69625 537.6427 156 -0.397469 0.6916
## sm_ses_grandmean 219.48842 537.6248 156 0.408256 0.6836
## groupmeanSES:sm_ses_grandmean -0.57991 0.7253 156 -0.799543 0.4252
## Correlation:
## (Intr) grpSES sm_ss_
## groupmeanSES 0.998
## sm_ses_grandmean -0.998 -1.000
## groupmeanSES:sm_ses_grandmean 0.094 0.131 -0.131
##
## Standardized Within-Group Residuals:
## Min Q1 Med Q3 Max
## -3.11929841 -0.75112002 0.02448373 0.76596673 2.78831371
##
## Number of Observations: 7185
## Number of Groups: 160
random_slope_cl_interaction <- lmer(mathach ~ 1 + groupmeanSES*sm_ses_grandmean + (1 + groupmeanSES|id)

## boundary (singular) fit: see ?isSingular
summary(random_slope_cl_interaction)

## Linear mixed model fit by REML ['lmerMod']
## Formula:
## mathach ~ 1 + groupmeanSES * sm_ses_grandmean + (1 + groupmeanSES |
## id)
## Data: data
##
## REML criterion at convergence: 46944.6
##
## Scaled residuals:
## Min 1Q Median 3Q Max
## -3.11654 -0.75065 0.02247 0.76812 2.79659
##
## Random effects:
## Groups Name Variance Std.Dev. Corr
## id (Intercept) 2.65355 1.6290
## groupmeanSES 0.04692 0.2166 -1.00
## Residual 39.15898 6.2577
## Number of obs: 7185, groups: id, 160
##
## Fixed effects:
## Estimate Std. Error t value
## (Intercept) 11.3252 3.3070 3.425
## groupmeanSES -228.1145 536.0175 -0.426
## sm_ses_grandmean 233.9132 536.0005 0.436
```

```

## groupmeanSES:sm_ses_grandmean    -0.5251      0.7363   -0.713
##
## Correlation of Fixed Effects:
##          (Intr) grpSES sm_ss_
## groupmenSES    0.998
## sm_ss_grndm   -0.998 -1.000
## grpmnSES:__    0.080  0.118 -0.118
## convergence code: 0
## boundary (singular) fit: see ?isSingular

random_slope_cl_interaction_2 <- lme(mathach ~ 1 + groupmeanSES*sm_ses_grandmean, random = ~ 1 + groupm
summary(random_slope_cl_interaction_2)

## Linear mixed-effects model fit by REML
## Data: data
##      AIC      BIC    logLik
## 46960.97 47016.01 -23472.49
##
## Random effects:
## Formula: ~1 + groupmeanSES | id
## Structure: General positive-definite, Log-Cholesky parametrization
##              StdDev      Corr
## (Intercept)  1.632105137 (Intr)
## groupmeanSES 0.005745282 -0.002
## Residual      6.257637586
##
## Fixed effects: mathach ~ 1 + groupmeanSES * sm_ses_grandmean
##              Value Std.Error DF   t-value p-value
## (Intercept)    11.42519    3.3155 7025   3.445964 0.0006
## groupmeanSES   -213.69606   537.6432  156  -0.397468 0.6916
## sm_ses_grandmean    219.48823   537.6253  156   0.408255 0.6836
## groupmeanSES:sm_ses_grandmean  -0.57990    0.7253  156  -0.799534 0.4252
## Correlation:
##              (Intr) grpSES sm_ss_
## groupmeanSES      0.998
## sm_ses_grandmean  -0.998 -1.000
## groupmeanSES:sm_ses_grandmean  0.094  0.131 -0.131
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -3.11929848 -0.75111989  0.02448355  0.76596678  2.78831459
##
## Number of Observations: 7185
## Number of Groups: 160

logit_random_intercept_and_slope <- glmer(minority ~ groupmeanSES + (1 + groupmeanSES | id), data = data,
family = binomial(link="logit"))
summary(logit_random_intercept_and_slope)

## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: binomial ( logit )
## Formula: minority ~ groupmeanSES + (1 + groupmeanSES | id)
## Data: data
##

```

```

##      AIC      BIC   logLik deviance df.resid
##  5453.9   5488.3 -2721.9   5443.9     7180
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -6.2886 -0.3942 -0.2073  0.1590  6.1544
##
## Random effects:
##   Groups Name      Variance Std.Dev. Corr
##   id      (Intercept)  2.529   1.590
##   groupmeanSES 11.445   3.383   -0.32
## Number of obs: 7185, groups: id, 160
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.7382     0.1678 -10.359 < 2e-16 ***
## groupmeanSES -2.0523     0.5370  -3.822 0.000132 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##              (Intr)
## groupmenSES -0.230
specified_variance_covariance_matrix_for_random_effects <- lme(mathach ~ 1 + groupmeanSES*sm_ses_grandmean,
                                                                correlation = corAR1(), data = data) # j
summary(specified_variance_covariance_matrix_for_random_effects)

## Linear mixed-effects model fit by REML
## Data: data
##      AIC      BIC   logLik
##  46962.8 47024.71 -23472.4
##
## Random effects:
## Formula: ~1 + groupmeanSES | id
## Structure: General positive-definite, Log-Cholesky parametrization
##              StdDev      Corr
## (Intercept)  1.62876592 (Intr)
## groupmeanSES 0.07080922 -0.039
## Residual     6.25836130
##
## Correlation Structure: AR(1)
## Formula: ~1 | id
## Parameter estimate(s):
##      Phi
## 0.005104377
## Fixed effects: mathach ~ 1 + groupmeanSES * sm_ses_grandmean
##              Value Std.Error DF   t-value p-value
## (Intercept)    11.42446    3.3152 7025   3.446083  0.0006
## groupmeanSES   -213.84493   537.5895  156  -0.397785  0.6913
## sm_ses_grandmean  219.63651   537.5716  156   0.408572  0.6834
## groupmeanSES:sm_ses_grandmean  -0.57970    0.7256  156  -0.798894  0.4256
## Correlation:
##              (Intr) grpSES sm_ss_
## groupmeanSES    0.998

```

```

## sm_ses_grandmean          -0.998 -1.000
## groupmeanSES:sm_ses_grandmean  0.094  0.131 -0.131
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -3.11860390 -0.75163778  0.02452874  0.76588109  2.78744053
##
## Number of Observations: 7185
## Number of Groups: 160

```