

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)
library(pscl, 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

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

```
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

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

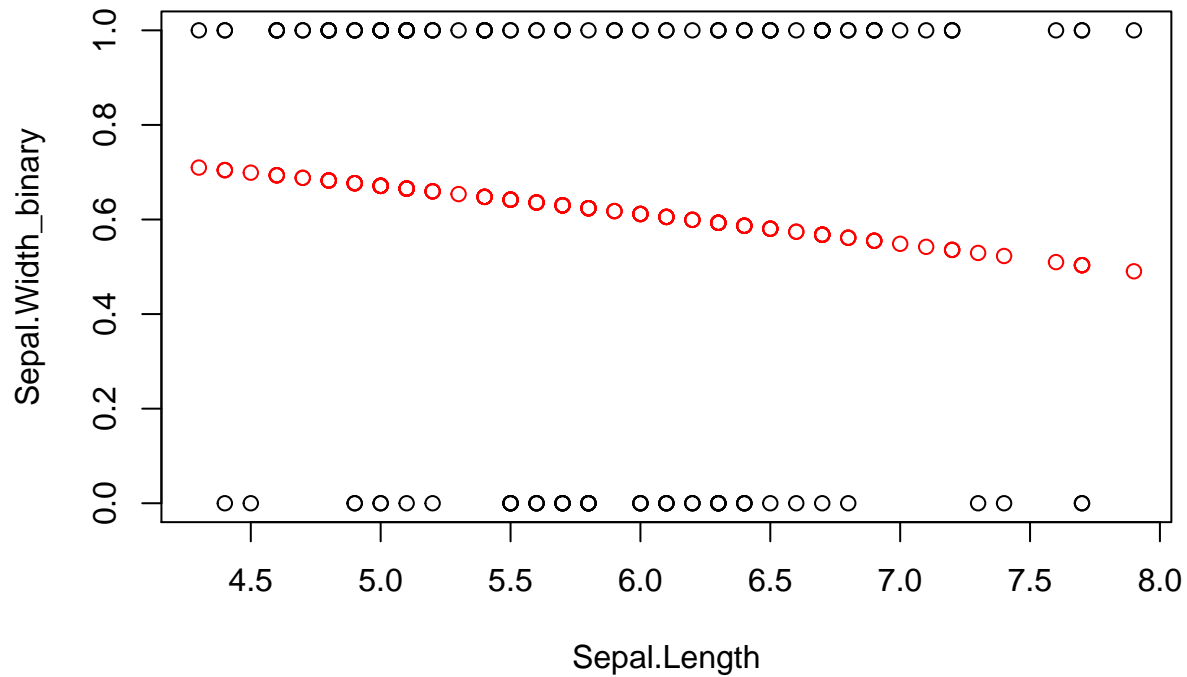
and X_{3i} = Sepal.Length of the i th data point.

```
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 *
```

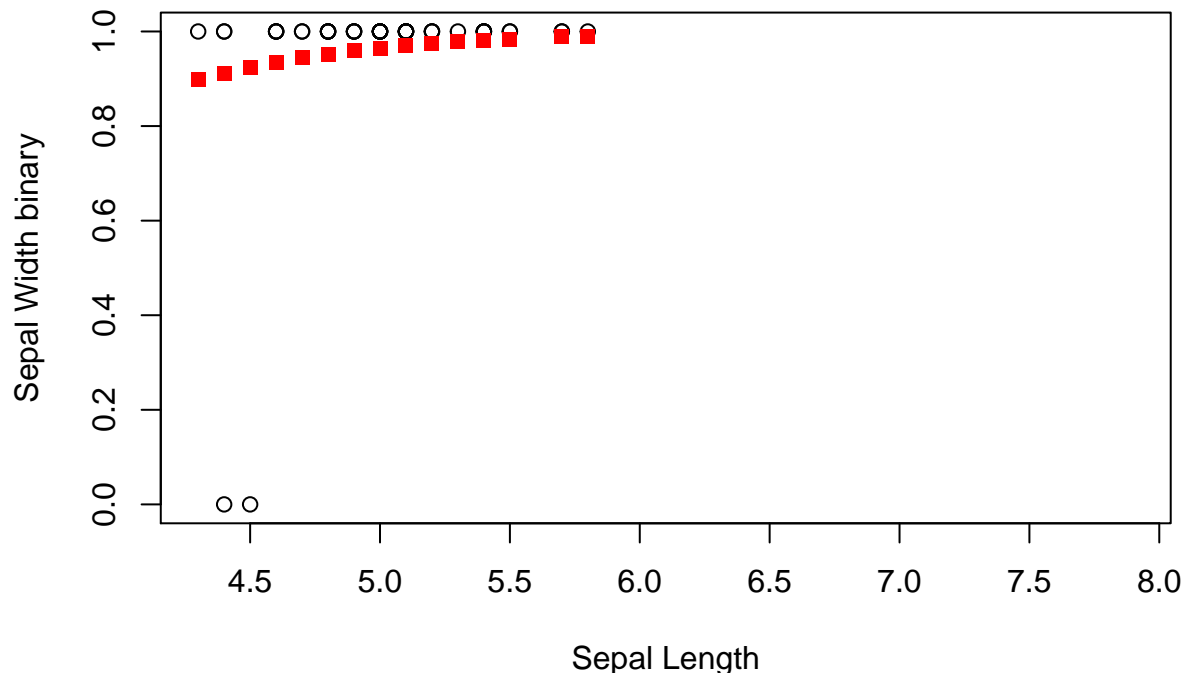
```
## Speciesversicolor -5.6936      0.9686 -5.878 4.16e-09 ***
## Speciesvirginica -5.4812      1.0879 -5.039 4.69e-07 ***
## Sepal.Length      1.6219      0.4510   3.596 0.000323 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family 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")
```

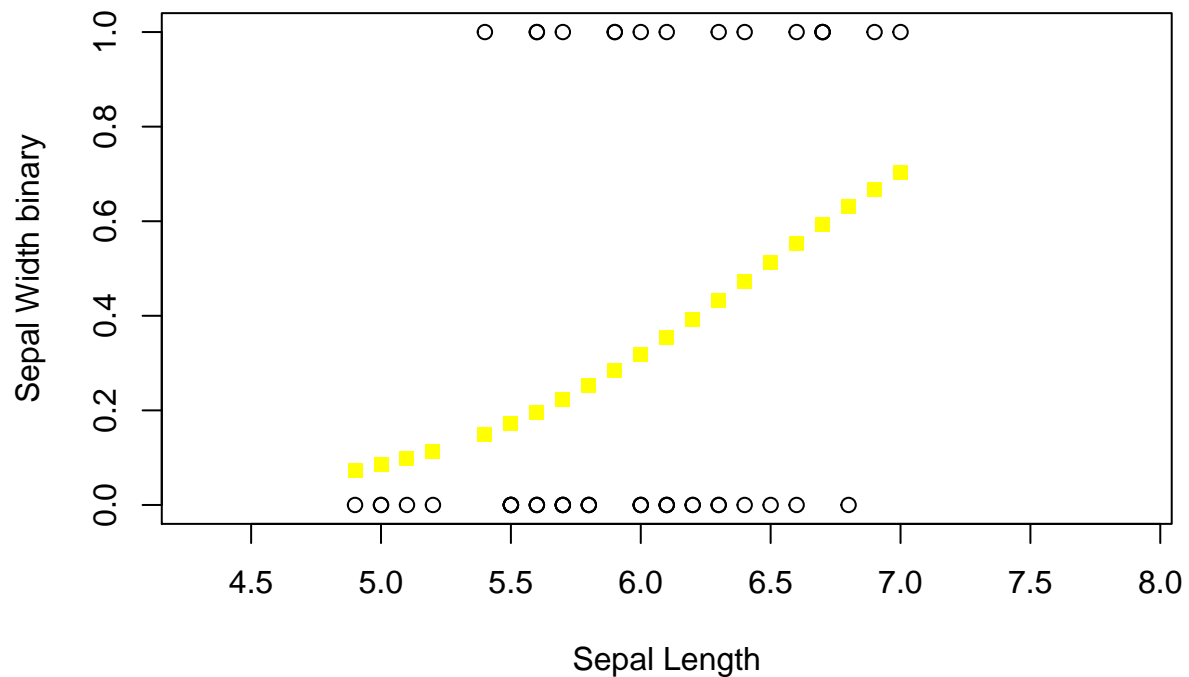
Scatter plot of sepal length vs sepal width



```
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")
```

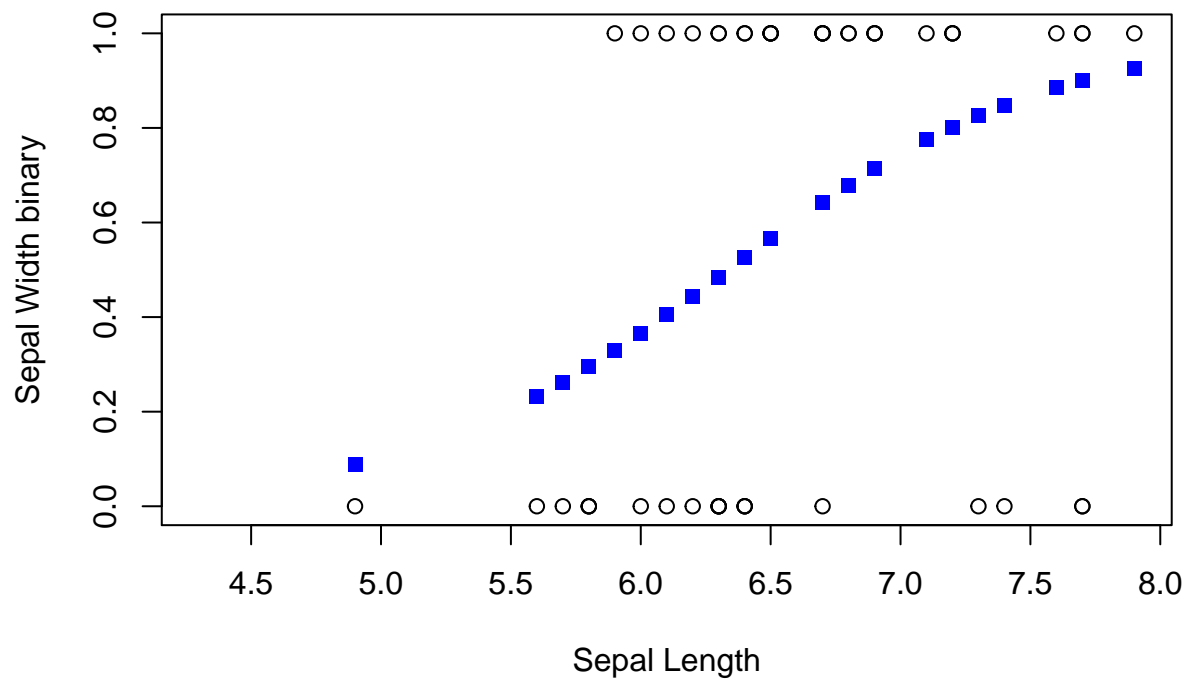
Scatter plot of sepal length vs sepal width



```
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 = the fitted model is equivalent to the null model

H_α = 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 General Linear Model

A poisson GLM is used to study *count* data (i.e. discrete numbers, 0, 1, 2, ...). *Count* data describes the number of events that occur within a given time frame.

insert plot of poisson distribution here

A poisson GLM is most useful when studying data in which the mean and variable are approximately equal. If they are not equal, the standard error of the model terms must be adjusted to account for the assumption violation.

Poisson Regression has the following properties: - a response binary variable, Y_i , that follows a Poisson distribution with mean μ_i - a systematic component, η_i , that relates the explanatory variables, $\eta_i = \sum_{j=1}^n \beta_j x_{ij}$ - a link function, $\log(\mu_i) = \sum_{j=1}^n \beta_j x_{ij}$

From Poisson regression, we learn the *mean* of each Y_i given the associated explanatory variables.

Data

We will consider the `bioChemists` data set in this section. This data set contains number of articles produced by PhD biochemistry student during the last 3 years of their PhD.

```
attach(bioChemists)
summary(bioChemists)
```

```
##      art      fem      mar      kid5
## Min.   : 0.000  Men   :494  Single :309  Min.   :0.0000
## 1st Qu.: 0.000  Women:421  Married:606  1st Qu.:0.0000
## Median : 1.000                                Median :0.0000
## Mean   : 1.693                                Mean   :0.4951
## 3rd Qu.: 2.000                                3rd Qu.:1.0000
## Max.   :19.000                                Max.   :3.0000
##      phd      ment
## Min.   :0.755  Min.   : 0.000
## 1st Qu.:2.260  1st Qu.: 3.000
## Median :3.150  Median : 6.000
## Mean   :3.103  Mean   : 8.767
## 3rd Qu.:3.920  3rd Qu.:12.000
## Max.   :4.620  Max.   :77.000
```

The data set also contains demographic data associated with each student. data of the flower of certain plant species. The data set has five variables:

- *art* - number of articles produced by the student in the last 3 years of their PhD
- *fem* - gender
- *mar* - marital status
- *kid5* - number of children less than 5
- *phd* - prestige of PhD program
- *ment* - number of articles of the mentor in the last 3 years

```
sapply(bioChemists, class)
```

```
##      art      fem      mar      kid5      phd      ment
## "integer" "factor" "factor" "numeric" "numeric" "integer"
```

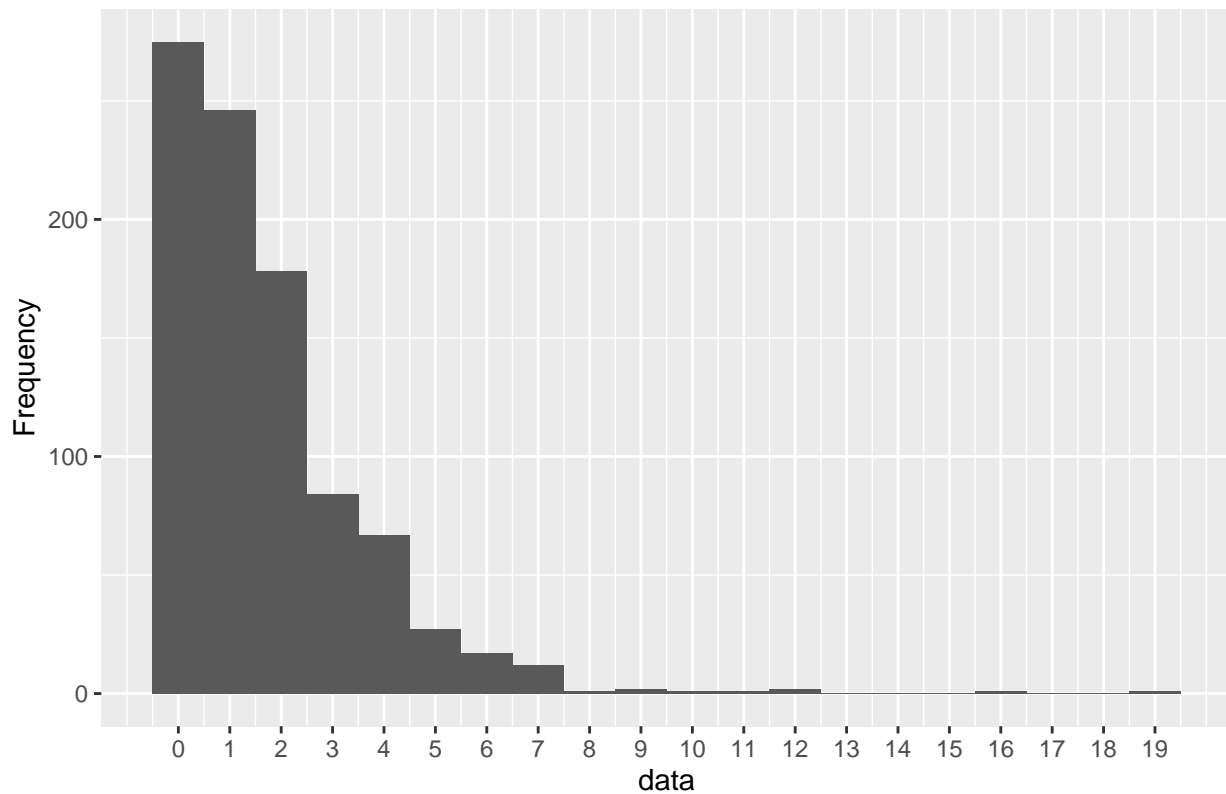
I convert `bioChemists$kid5` from numeric to factor. This will be used later.

```
bioChemists$kid5 <- factor(bioChemists$kid5,
                           levels= unique(bioChemists$kid5),
                           labels= unique(bioChemists$kid5))
```

Plotting the bar graph of `bioChemists$art`, we can see that the data looks Poisson-like since there is a large number of observations at 0.

```
ggplot(bioChemists,aes(x=bioChemists$art))+
  geom_histogram(binwidth = 1, center = 1) +
  scale_x_continuous(breaks=seq(0,max(bioChemists$art), by = 1))+
  ylab("Frequency")+ xlab("data")+
  ggtitle("Histogram plot of the number of articles published by biochemist phd students in last 3 years")
```

Histogram plot of the number of articles published by biochemist phd students



We can “quantify” the Poisson-ness by analyzing the mean and variance of the data.

```
mean(bioChemists$art)
```

```
## [1] 1.692896
```

```
var(bioChemists$art)
```

```
## [1] 3.709742
```

Although mean and variance are not equal, we will still fit it to Poisson distribution.

Possion Regression with constant term

```
poisson_model = glm(art ~ 1, family=poisson(link=log),data=bioChemists)
summary(poisson_model)
```

```
##
## Call:
## glm(formula = art ~ 1, family = poisson(link = log), data = bioChemists)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.8401  -1.8401  -0.5770   0.2294   7.5677
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.52644    0.02541   20.72  <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: 1817.4  on 914  degrees of freedom
## Residual deviance: 1817.4  on 914  degrees of freedom
## AIC: 3487.1
##
## Number of Fisher Scoring iterations: 5
```

The constant term is the mean number of counts.

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

```
## (Intercept)
##      0.5264408
```

```
print(log(mean(bioChemists$art)))
```

```
## [1] 0.5264408
```

Hypothesis test for goodness of fit

TODO should i reject the null in this case?

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

```
## [1] 3.304511e-62
```

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

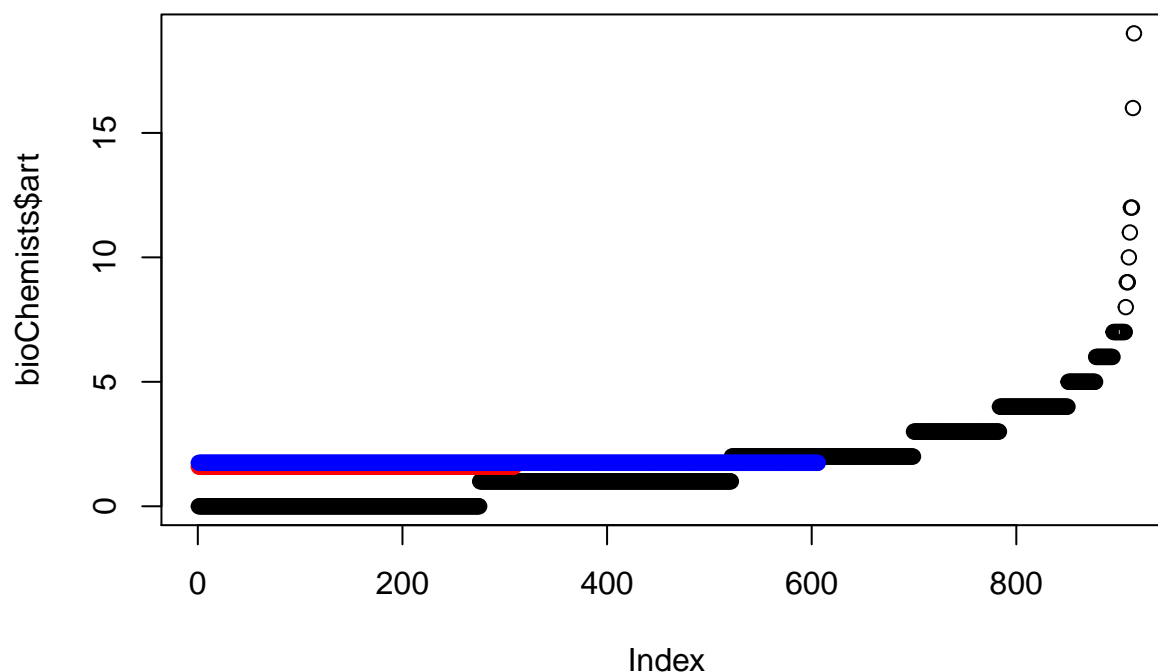
```
## [1] 1.016236e-70
```

Possion Regression with martial status covariate

```
poisson_model = glm(art~ 1 + mar , family=poisson(link=log),data=bioChemists)
summary(poisson_model)

##
## Call:
## glm(formula = art ~ 1 + mar, family = poisson(link = log), data = bioChemists)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.8677  -1.7845  -0.5042   0.3107   7.4992
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.46514    0.04508  10.317  <2e-16 ***
## marMarried   0.09117    0.05458   1.671   0.0948 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 1817.4  on 914  degrees of freedom
## Residual deviance: 1814.6  on 913  degrees of freedom
## AIC: 3486.3
##
## Number of Fisher Scoring iterations: 5

plot(bioChemists$art)
points(poisson_model$fitted[bioChemists$mar=='Single'],col="red")
points(poisson_model$fitted[bioChemists$mar=='Married'],col="blue")
```



Hypothesis test for goodness of fit

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

## [1] 4.731233e-62

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

## [1] 1.016236e-70

anova(poisson_model, test="Chisq")

## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: art
##
## Terms added sequentially (first to last)
##
##
##      Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                      914      1817.4
## mar    1    2.8211      913      1814.6 0.09304 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Possion Regression with martial status and children covariate

TODO why are interaction factors NA

```
poisson_model = glm(art ~ 1 + kid5 + mar,
                    family=poisson(link=log), data=bioChemists)
summary(poisson_model)

##
## Call:
## glm(formula = art ~ 1 + kid5 + mar, family = poisson(link = log),
##      data = bioChemists)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.9280  -1.7845  -0.5042   0.3518   7.3520
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.46514    0.04508  10.317  <2e-16 ***
## kid51         -0.05510    0.06907  -0.798   0.4250
## kid52         -0.18620    0.08960  -2.078   0.0377 *
## kid53         -0.82747    0.28067  -2.948   0.0032 **
## marMarried     0.15470    0.06235   2.481   0.0131 *
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 1817.4  on 914  degrees of freedom
## Residual deviance: 1799.9  on 910  degrees of freedom
## AIC: 3477.7
##
## 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] 6.462874e-61

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

## [1] 1.016236e-70

anova(poisson_model, test="Chisq")

## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: art
##
## Terms added sequentially (first to last)
##
##      Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                                914      1817.4
## kid5  3  11.3045             911      1806.1 0.01019 *
## mar   1   6.1638             910      1799.9 0.01304 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Poisson Regression with marital status, children and number of mentor articles

```
poisson_model = glm(art ~ 1 + kid5 + mar + ment,
                    family=poisson(link=log), data=bioChemists)
summary(poisson_model)

##
## Call:
## glm(formula = art ~ 1 + kid5 + mar + ment, family = poisson(link = log),
##      data = bioChemists)
```

```
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.6213  -1.5873  -0.3867   0.5089   5.6414
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.204373   0.050086   4.080  4.5e-05 ***
## kid51       -0.134128   0.069781  -1.922  0.05459 .
## kid52       -0.273998   0.090151  -3.039  0.00237 **
## kid53       -0.726113   0.280827  -2.586  0.00972 **
## marMarried   0.177081   0.062386   2.838  0.00453 **
## ment        0.026487   0.001954  13.556 < 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: 1817.4  on 914  degrees of freedom
## Residual deviance: 1650.6  on 909  degrees of freedom
## AIC: 3330.3
##
## 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] 1.194033e-45
```

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

```
## [1] 1.016236e-70
```

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

```
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: art
##
## Terms added sequentially (first to last)
##
##
```

	Df	Deviance	Resid. Df	Resid. Dev	Pr(>Chi)
## NULL			914	1817.4	
## kid5	3	11.304	911	1806.1	0.01019 *
## mar	1	6.164	910	1799.9	0.01304 *
## ment	1	149.384	909	1650.5	< 2e-16 ***


```
## ---  
## 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.

```
bioChemists$art_binary <- sapply(bioChemists$art,function(x) ifelse(x > 2, 1, 0))
bioChemists$ment_binary <- sapply(bioChemists$ment,function(x) ifelse(x > median(bioChemists$ment), 1, 0))
```

One-Way Contingency Table

```
table(art_relative=bioChemists$art_binary)
```

```
## art_relative
##    0    1
## 699 216
```

Two-Way Contingency Table

```
table(art_relative=bioChemists$art_binary,ment=bioChemists$ment_binary)
```

```
##           ment
## art_relative  0  1
##           0 408 291
##           1  84 132
```

Independent Model

```
contingency_table = table(art_relative=bioChemists$art_binary,ment=bioChemists$ment_binary)
```

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 ~ art_relative + ment,
                        data = contingency_table.df, family = poisson)
```

```
summary(log_linear_model)
```

```
##
## Call:
## glm(formula = Freq ~ art_relative + ment, family = poisson, data = contingency_table.df)
##
## Deviance Residuals:
##      1      2      3      4
## 1.635 -3.139 -1.819  3.064
```

```
##
## Coefficients:
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)   5.92921    0.04869 121.787  <2e-16 ***
## art_relative1 -1.17437    0.07785 -15.086  <2e-16 ***
## ment1         -0.15111    0.06631  -2.279   0.0227 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
## Null deviance: 298.793  on 3  degrees of freedom
## Residual deviance:  25.224  on 1  degrees of freedom
## AIC: 59.578
##
## Number of Fisher Scoring iterations: 4
```

Saturated Model

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

```
summary(log_linear_model)
```

```
##
## Call:
## glm(formula = Freq ~ art_relative * ment, family = poisson, data = contingency_table.df)
##
## Deviance Residuals:
## [1]  0  0  0  0
##
## Coefficients:
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)    6.01127    0.04951 121.422  < 2e-16 ***
## art_relative1 -1.58045    0.11982 -13.191  < 2e-16 ***
## ment1         -0.33794    0.07673  -4.404 1.06e-05 ***
## art_relative1:ment1  0.78993    0.15927   4.960 7.06e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
## Null deviance: 2.9879e+02  on 3  degrees of freedom
## Residual deviance: 1.3101e-14  on 0  degrees of freedom
## AIC: 36.354
##
## Number of Fisher Scoring iterations: 2
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                        3    298.793
## art_relative      1  268.360      2    30.432 < 2.2e-16 ***
## ment              1   5.208      1    25.224  0.02248 *
## art_relative:ment  1   25.224      0     0.000 5.104e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Three-Way Contingency Table

```
table(art_relative=bioChemists$art_binary,ment=bioChemists$ment_binary,
      kid5=bioChemists$kid5)
```

```
## , , kid5 = 0
##
##           ment
## art_relative  0  1
##           0 270 184
##           1  54  91
##
## , , kid5 = 1
##
##           ment
## art_relative  0  1
##           0  83  65
##           1  21  26
##
## , , kid5 = 2
##
##           ment
## art_relative  0  1
##           0  47  35
##           1   8  15
##
## , , kid5 = 3
##
##           ment
## art_relative  0  1
##           0   8   7
##           1   1   0
```

Independent Model

```
contingency_table = table(art_relative=bioChemists$art_binary,
                          ment=bioChemists$ment_binary,
                          kid5=bioChemists$kid5)
```

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

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

```
log_linear_model_int <- glm(Freq ~ art_relative + ment + kid5,  
                           data = contingency_table.df, family = poisson)
```

```
summary(log_linear_model_int)
```

```
##  
## Call:  
## glm(formula = Freq ~ art_relative + ment + kid5, family = poisson,  
##      data = contingency_table.df)  
##  
## Deviance Residuals:  
##      Min       1Q   Median       3Q      Max   
## -2.66652  -0.99671  -0.01157   0.68255   2.99079   
##  
## Coefficients:  
##              Estimate Std. Error z value Pr(>|z|)      
## (Intercept)   5.50554    0.05428  101.420  <2e-16 ***  
## art_relative1 -1.17437    0.07785  -15.086  <2e-16 ***  
## ment1         -0.15111    0.06631   -2.279   0.0227 *   
## kid51         -1.12226    0.08245  -13.612  <2e-16 ***  
## kid52         -1.74130    0.10580  -16.459  <2e-16 ***  
## kid53         -3.62267    0.25331  -14.301  <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: 1148.541  on 15  degrees of freedom  
## Residual deviance:   32.637  on 10  degrees of freedom  
## AIC: 122.9  
##  
## Number of Fisher Scoring iterations: 4
```

Saturated Model

```
log_linear_model_sat <- glm(Freq ~ art_relative*ment*kid5,  
                           data = contingency_table.df, family = poisson)
```

```
summary(log_linear_model_sat)
```

```
##  
## Call:  
## glm(formula = Freq ~ art_relative * ment * kid5, 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 Pr(>|z|)
```

```

## (Intercept)          5.598e+00  6.086e-02  91.991 < 2e-16 ***
## art_relative1        -1.609e+00  1.491e-01 -10.796 < 2e-16 ***
## ment1                -3.835e-01  9.560e-02  -4.012 6.03e-05 ***
## kid51                -1.180e+00  1.255e-01  -9.399 < 2e-16 ***
## kid52                -1.748e+00  1.580e-01 -11.061 < 2e-16 ***
## kid53                -3.519e+00  3.588e-01  -9.809 < 2e-16 ***
## art_relative1:ment1    9.054e-01  1.966e-01   4.605 4.12e-06 ***
## art_relative1:kid51    2.351e-01  2.862e-01   0.822  0.411
## art_relative1:kid52   -1.613e-01  4.105e-01  -0.393  0.694
## art_relative1:kid53   -4.700e-01  1.071e+00  -0.439  0.661
## ment1:kid51           1.390e-01  1.912e-01   0.727  0.467
## ment1:kid52           8.869e-02  2.429e-01   0.365  0.715
## ment1:kid53           2.500e-01  5.263e-01   0.475  0.635
## art_relative1:ment1:kid51 -4.473e-01  3.901e-01  -1.147  0.251
## art_relative1:ment1:kid52  1.805e-02  5.293e-01   0.034  0.973
## art_relative1:ment1:kid53 -2.307e+01  4.225e+04  -0.001  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: 1.1485e+03 on 15 degrees of freedom
## Residual deviance: 4.1221e-10 on 0 degrees of freedom
## AIC: 110.27
##
## Number of Fisher Scoring iterations: 20
anova(log_linear_model_int,log_linear_model_sat,test='Chisq')

## Analysis of Deviance Table
##
## Model 1: Freq ~ art_relative + ment + kid5
## Model 2: Freq ~ art_relative * ment * kid5
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1         10      32.637
## 2          0       0.000 10   32.637 0.0003134 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
anova(log_linear_model_sat,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      1148.54
## art_relative      1    268.36          14      880.18 < 2.2e-16 ***
## ment              1     5.21          13      874.97  0.02248 *
## kid5              3    842.34          10      32.64 < 2.2e-16 ***

```

```
## art_relative:ment      1    25.22      9      7.41 5.104e-07 ***
## art_relative:kid5      3     3.80      6      3.61 0.28374
## ment:kid5              3     0.19      3      3.42 0.97911
## art_relative:ment:kid5 3     3.42      0     0.00 0.33117
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

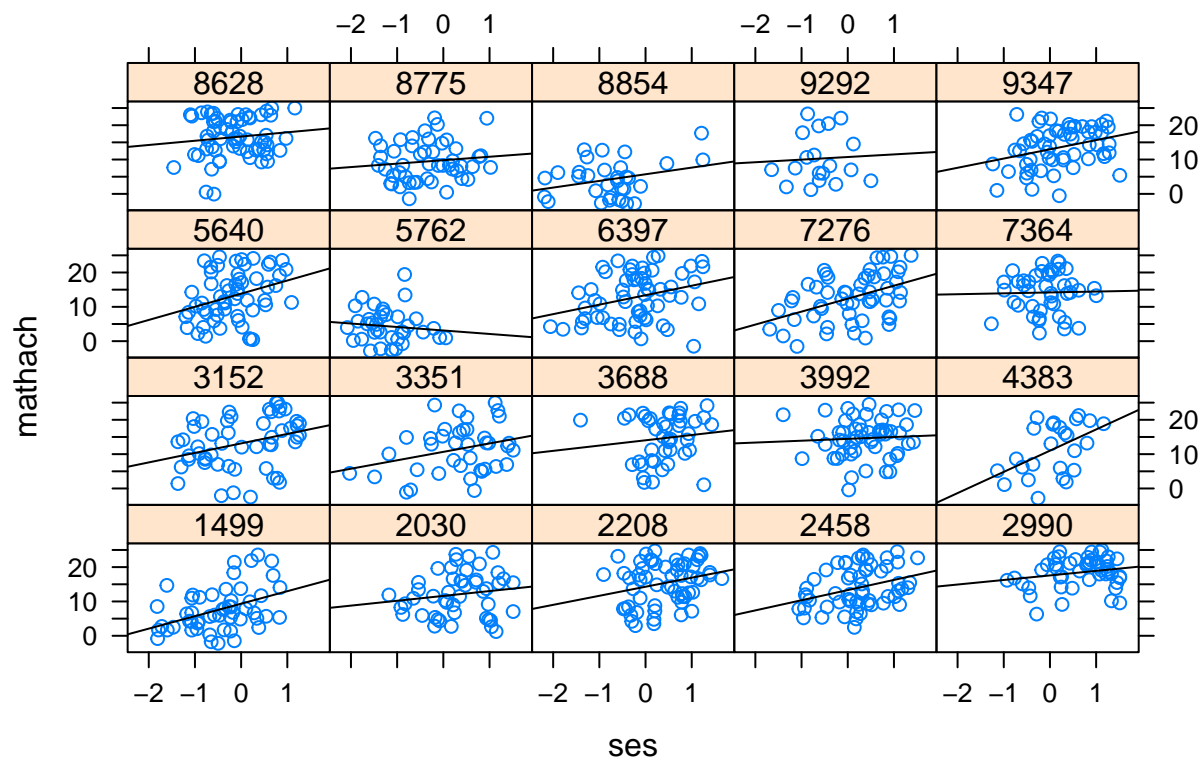
data <- merge(student_data, school_data, by = "id")

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, ]

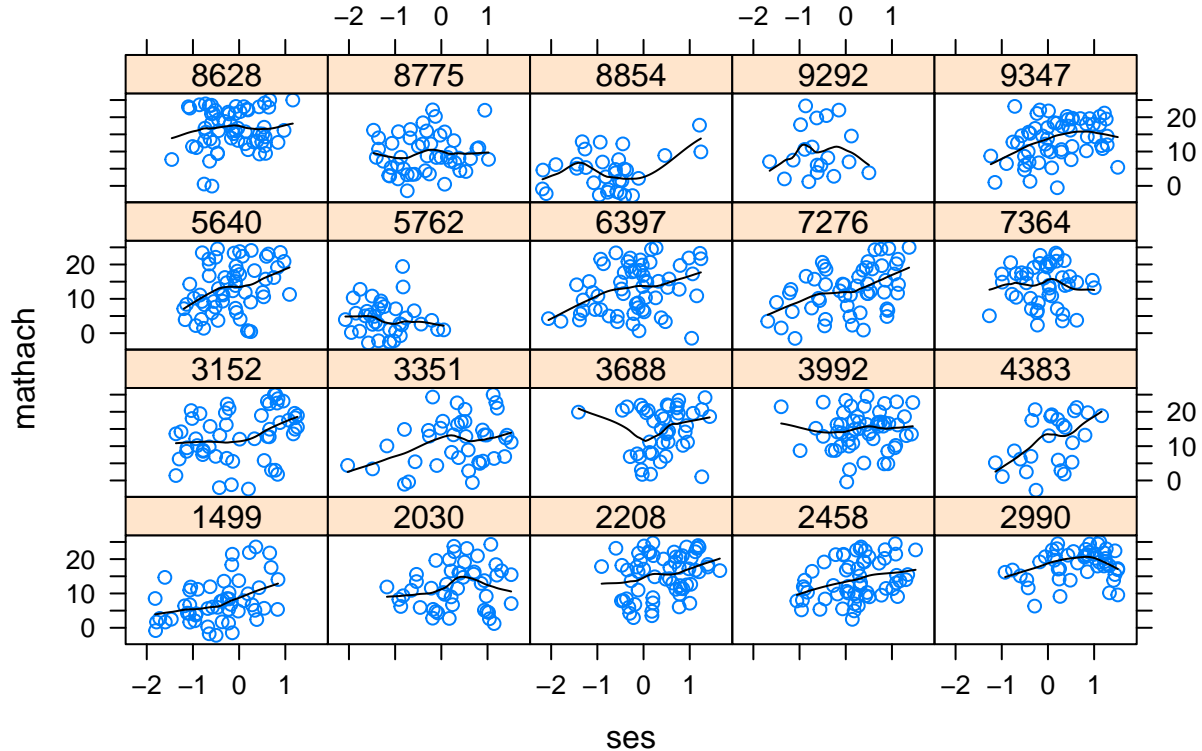
xyplot(mathach ~ ses | as.factor(id), subset,
       col.line = 'black',
       type = c("p", "r"),
       main = 'Variability in Math Achievement ~ SES Relationship')
```

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')
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## FALSE, : span too small. fewer data values than degrees of freedom.

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## FALSE, : pseudoinverse used at -0.2054

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## FALSE, : neighborhood radius 0.1974

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## FALSE, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## FALSE, : There are other near singularities as well. 0.99481

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## FALSE, : span too small. fewer data values than degrees of freedom.

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## FALSE, : pseudoinverse used at -0.2054

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## FALSE, : neighborhood radius 0.1974

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## FALSE, : reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## FALSE, : There are other near singularities as well. 0.99481

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## FALSE, : span too small. fewer data values than degrees of freedom.

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## FALSE, : pseudoinverse used at -0.2054

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## FALSE, : neighborhood radius 0.1974

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## FALSE, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## FALSE, : There are other near singularities as well. 0.99481

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## FALSE, : span too small. fewer data values than degrees of freedom.

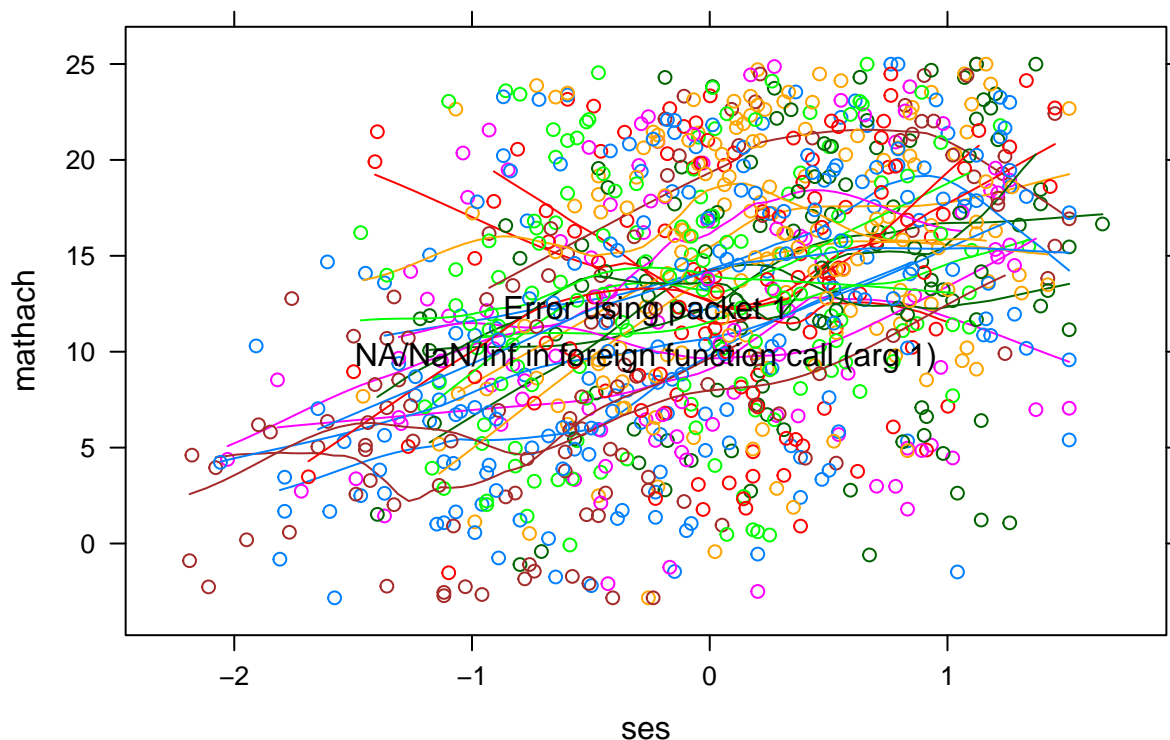
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## FALSE, : pseudoinverse used at -0.2054

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## FALSE, : neighborhood radius 0.1974

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## FALSE, : reciprocal condition number 0

## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## FALSE, : There are other near singularities as well. 0.99481
```

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 = d
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
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), data = data)
## 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

```