Day 2: Linear models and linear mixed models

Himanshu Yadav

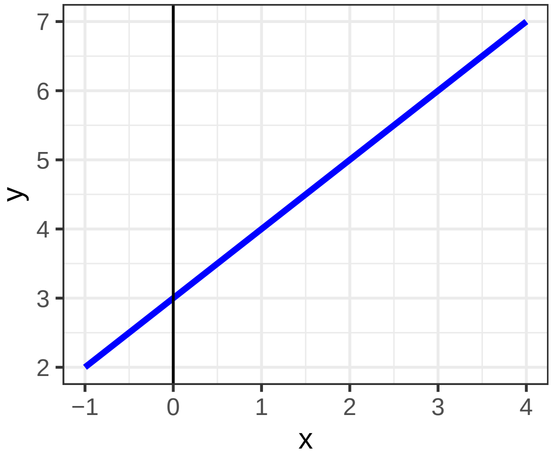
15.12.2020

Linear model in practice

Consider a linear equation,

$y = mx + c$

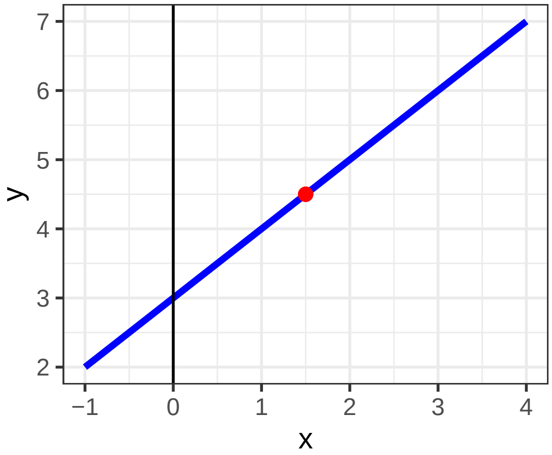
where m is the slope of the line, and c is the intercept.



This line has intercept 3 and slope 1. You can write equation for y as:

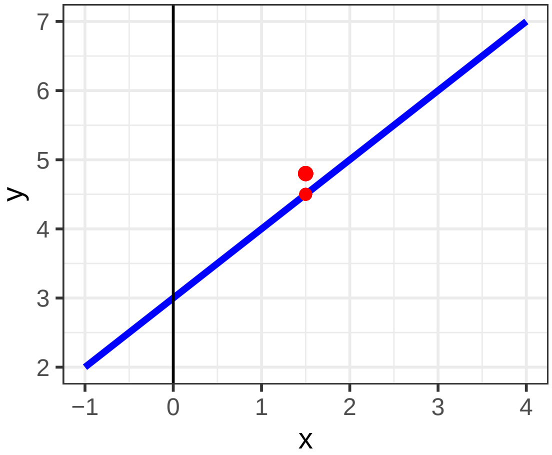
$y = 3 + x$

Consider a point on the line corresponding to x=1.5, what will be y?



$y = 3 + 1.5 = 4.5$

What if there is a point slightly above or below the line?



Say the point is (x,y')

I can write,

$y' = y + \epsilon$

$y' = mx + c + \epsilon$

This is basic idea behind *linear regression*: you can model the relationship between a dependent/outcome variable y and an independent/predictor variable x using an error variable ϵ such that every observation of y is a linear function of x plus error variable ϵ.

The linear mode to predict i^{th} observation of y using i^{th} observation of x:

$y_i = \alpha + \beta x_i + \epsilon_i$

If you have multiple predictors, say u, v, w and x

$y_i = \alpha + \beta_1 w_i + \beta_2 v_i + \beta_3 w_i + \beta_4 x_i + \epsilon_i$

Your goal is to estimate values of α, β_1,.. such that the error term ϵ is minimized.

Note: For computational convenience, errors $\epsilon_1, \epsilon_2, ..., \epsilon_n$ are assumed to be normally distributed, i.e., $\epsilon_i \sim Normal(0, \delta)$. While, in practice, most of linear modeling packages in R, python fit models under normality assumption, this assumption is often ignored during the data analysis and it may cause false interpretations. Other assumptions include independence of errors, Homoscedasticity and linear relationship.

Example. Linear model of relationship between weight and age

Suppose you are given data of weight and age for n people.

You assume that body weight increases linearly with increase in age. You can write your model like this.

$Weight_i = \alpha + \beta Age_i + \epsilon_i$

where $Weight_i$ and Age_i are weight and age of i^{th} person respectively. α and β are intercept and slope parameters.

You use some algorithm to estimate the values of α and β such that the errors are minimized. For example, we will use function $lm()$ in R to do this.

Let $\hat{\alpha}$ and $\hat{\beta}$ be the estimated coefficients. Now you can predict *weight* of a person given his/her *age* and estimated coefficients.

$Weight_{pred} = \hat{\alpha} + \hat{\beta} Age$

You can generate model predictions for each observation of *age* in your dataset,

$Weight_{pred,i} = \hat{\alpha} + \hat{\beta} Age_i$

where $Weight_{pred,i}$ is predicted weight of i^{th} person (predicted by a fitted linear model).

The difference between observed value and model predicted value of outcome variable is called residual.

$residual_i = Weight_i - Weight_{pred,i}$

Residuals are estimates of error, ϵ.

If you have assumed errors are normally distributed, then $residual_i \sim Normal(0, \delta)$.

```r
data <- read.table("Howell.csv", header = T,sep=",")
data <- data[,-1]
data$Gender <- ifelse(data$male==1,"Male","Female")
head(data)
```

```
##    height   weight age male Gender
## 1 151.765 47.82561  63    1   Male
## 2 139.700 36.48581  63    0 Female
## 3 136.525 31.86484  65    0 Female
## 4 156.845 53.04191  41    1   Male
## 5 145.415 41.27687  51    0 Female
## 6 163.830 62.99259  35    1   Male
```

```r
# Linear model to predict weight from age
# Subsetting for age <= 25
data.young <- subset(data,age<=25)
# For people <=25 years of age,
# I assume weight increases linearly with age
model1 <- lm(weight~age,data=data.young)
model1
```

```
##
## Call:
## lm(formula = weight ~ age, data = data.young)
##
## Coefficients:
## (Intercept)          age
##       5.522        1.740
```

```r
summary(model1)
```

```
##
## Call:
## lm(formula = weight ~ age, data = data.young)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.350  -3.028   0.155   2.242  18.003
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.52219    0.53194   10.38   <2e-16 ***
## age          1.73987    0.03933   44.24   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.807 on 253 degrees of freedom
## Multiple R-squared:  0.8855, Adjusted R-squared:  0.8851
## F-statistic:  1957 on 1 and 253 DF,  p-value: < 2.2e-16
```

- Residuals show distribution of estimate of errors, i.e., distribution of difference between observed weight and predicted weight.
- Coefficients summarize estimates for each parameter of the model.
- The estimate corresponding to *age* is estimate of slope term associated with *age*, you can also call it *main effect of age on weight*.
- The slope estimate tell us with what extent weight increases with increase in age
- I will explain Std. Error, t value and Pr value soon.
- In short, if Pr value (p-value) for a parameter estimate is less than 0.05, we infer that the true (population) value of the parameter is significantly different from zero.

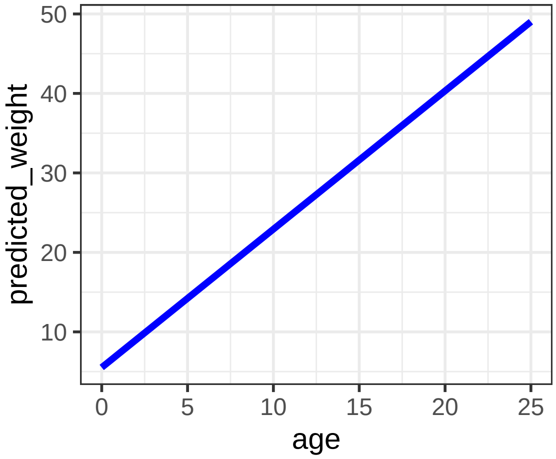
```
# Let us generate predictions from this model
head(predict(model1))
```

```
##        9       14       19       20       21       24
## 38.57968 40.31954 26.40060 19.44113 16.83133 28.14047
```

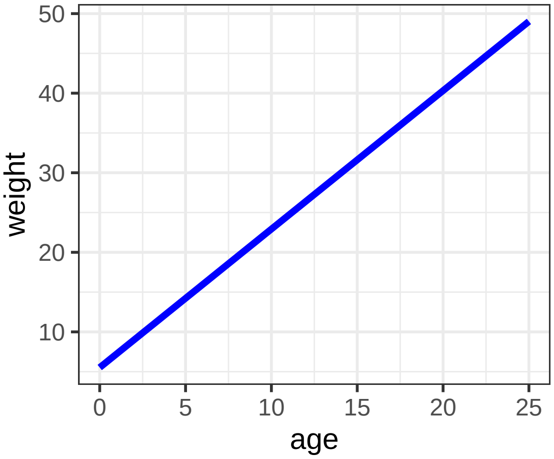
```
data.young$predicted_weight <- predict(model1)
head(data.young)
```

```
##     height   weight  age male Gender predicted_weight
## 9  147.955 34.86988 19.0    0 Female         38.57968
## 14 149.900 47.70000 20.0    0 Female         40.31954
## 19 121.920 19.61785 12.0    1   Male         26.40060
## 20 105.410 13.94795  8.0    0 Female         19.44113
## 21  86.360 10.48931  6.5    0 Female         16.83133
## 24 129.540 23.58678 13.0    1   Male         28.14047
```

```
# Let us plot the predicted weight
ggplot(data=data.young,aes(x=age,y=predicted_weight))+
  geom_line(size=1.2,color="blue")+
  theme_bw()
```



```
ggplot(data=data.young,aes(x=age,y=weight))+
  geom_smooth(method="lm",formula=y~x,size=1.2,color="blue",se=F)+
  theme_bw()
```



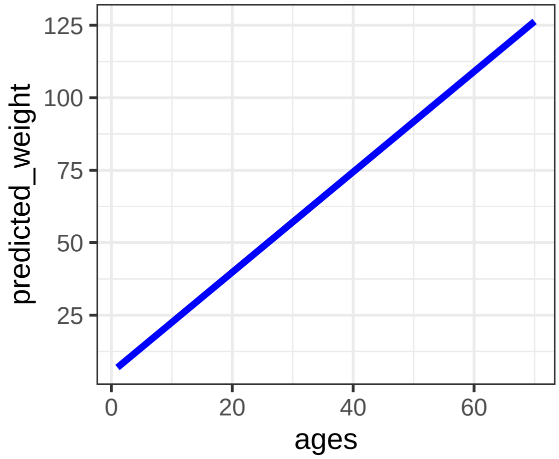
```r
# You can manually generate model predictions
summary(model1)$coefficients
```

```
##             Estimate Std. Error  t value      Pr(>|t|)
## (Intercept) 5.522192 0.53193811 10.38127  2.906648e-21
## age         1.739868 0.03933107 44.23646 4.556358e-121
```

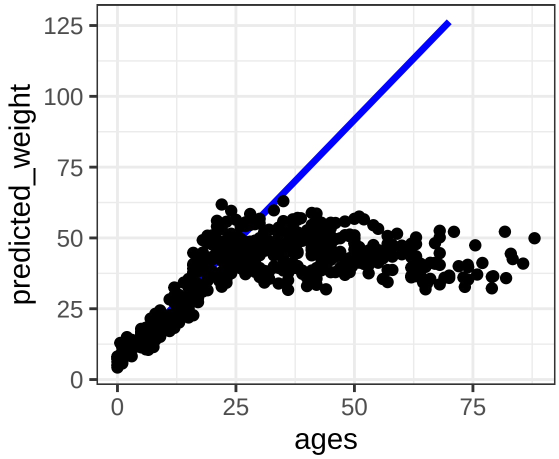
```r
ages <- 1:70
newdata <- data.frame(ages)
newdata$predicted_weight <- 5.22+1.73*ages
head(newdata)
```

```
##   ages predicted_weight
## 1    1             6.95
## 2    2             8.68
## 3    3            10.41
## 4    4            12.14
## 5    5            13.87
## 6    6            15.60
```

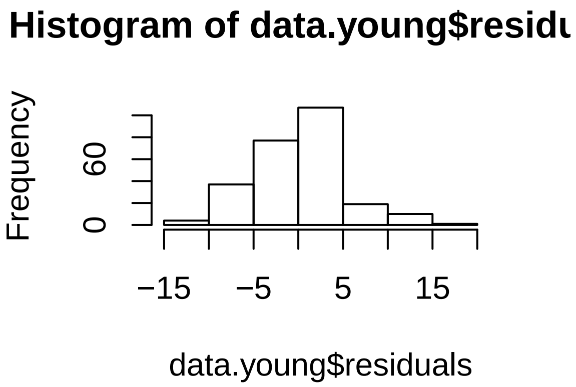
```r
ggplot(data=newdata,aes(x=ages,y=predicted_weight))+
  geom_line(size=1.2,color="blue")+
  theme_bw()
```



```r
ggplot(data=newdata,aes(x=ages,y=predicted_weight))+
  geom_line(size=1.2,color="blue")+
  theme_bw()+
  geom_point(data=data,aes(x=age,y=weight))
```



```r
#Check the distribution of residuals
data.young$residuals <- data.young$weight - data.young$predicted_weight
hist(data.young$residuals)
```



Example. Linear model of relationship between weight, age and height

Now suppose someone points out that the real reason behind increase in weight is body height. It is possible that people of same age have different weights due to different heights. So, **height would have an effect on weight over and above the effect of age**. If you want to model this situation, you can use both age and height as independent predictors in the linear model.

$Weight_i = \alpha + \beta_1 Age_i + \beta_2 Height_i + \epsilon_i$

After fitting this model, you will obtain the etimates for intercept and slopes i.e., $\hat{\alpha}$, $\hat{\beta}_1$ and $\hat{\beta}_2$.

- $\hat{\alpha}$ is weight of a person who is zero years old and have zero height.
- $\hat{\beta}_1$: for a person has zero height, by what extent his/her weight increases with increase in age
- $\hat{\beta}_2$: for a person who is zero year old, by what extent his/her weight increases with increase in height

Center the predictors for easier interpretation.

$cAge_i = Age_i - mean(Age)$

$cHeight_i = Height_i - mean(Height)$

$Weight_i = \alpha + \beta_1 cAge_i + \beta_2 cHeight_i + \epsilon_i$

- $\hat{\alpha}$ is weight of a person who is average old and have average height.
- $\hat{\beta}_1$: for a person has average height, by what extent his/her weight increases with increase in age
- $\hat{\beta}_2$: for a person who is average old, by what extent his/her weight increases with increase in height

```r
model2 <- lm(weight~age+height,data=data.young)
model2
```

```
##
## Call:
## lm(formula = weight ~ age + height, data = data.young)
##
## Coefficients:
## (Intercept)          age       height
##    -10.1164       1.0020       0.1995
```

```r
summary(model2)
```

```
##
## Call:
## lm(formula = weight ~ age + height, data = data.young)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.4769 -3.2279  0.0761  2.9275 15.4216
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -10.11638    2.18284  -4.634 5.74e-06 ***
## age           1.00197    0.10661   9.398  < 2e-16 ***
## height        0.19947    0.02715   7.347 2.81e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.371 on 252 degrees of freedom
## Multiple R-squared:  0.9057, Adjusted R-squared:  0.905
## F-statistic:  1210 on 2 and 252 DF,  p-value: < 2.2e-16
```

```r
# Center the predictors
data.young$c_age <- scale(data.young$age)
data.young$c_height <- scale(data.young$height)
head(data.young)
```

```
##     height   weight  age male Gender predicted_weight residuals      c_age
## 9  147.955 34.86988 19.0    0 Female         38.57968 -3.709790  1.0235255
## 14 149.900 47.70000 20.0    0 Female         40.31954  7.380457  1.1539142
## 19 121.920 19.61785 12.0    1   Male         26.40060 -6.782748  0.1108048
## 20 105.410 13.94795  8.0    0 Female         19.44113 -5.493178 -0.4107499
## 21  86.360 10.48931  6.5    0 Female         16.83133 -6.342016 -0.6063329
## 24 129.540 23.58678 13.0    1   Male         28.14047 -4.553686  0.2411935
##       c_height
## 9   0.94003654
## 14  1.00462126
## 19  0.07553096
## 20 -0.47269209
## 21 -1.10525715
## 24  0.32855698
```

```
model2 <- lm(weight~c_age+c_height,data=data.young)
model2
```

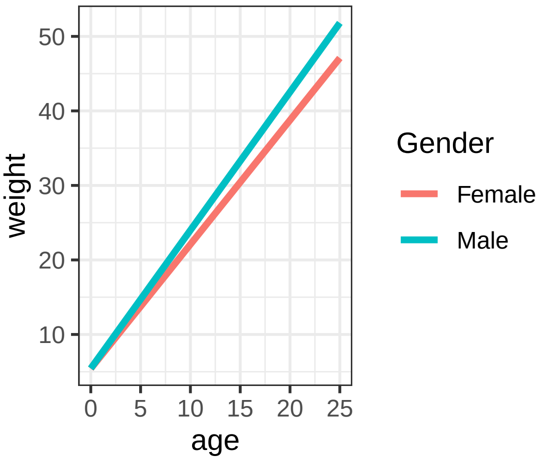
```
##
## Call:
## lm(formula = weight ~ c_age + c_height, data = data.young)
##
## Coefficients:
## (Intercept)        c_age     c_height
##      24.922        7.685        6.007
```

```
summary(model2)
```

```
##
## Call:
## lm(formula = weight ~ c_age + c_height, data = data.young)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.4769 -3.2279  0.0761  2.9275 15.4216
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  24.9221     0.2738  91.039  < 2e-16 ***
## c_age         7.6845     0.8177   9.398  < 2e-16 ***
## c_height      6.0073     0.8177   7.347 2.81e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.371 on 252 degrees of freedom
## Multiple R-squared:  0.9057, Adjusted R-squared:  0.905
## F-statistic:  1210 on 2 and 252 DF,  p-value: < 2.2e-16
```

Example. Weight increases faster with age for males compared to females

We assume that rate of increases in weight w.r.t. age is faster in males than females. Slope of line predicting weight from age will be larger for male population than females.



$Weight_i = \alpha + \beta_1' Age_i + \beta_2 Gender_i + \epsilon_i$

$\beta_1' = \beta_1 + \beta_3 Gender_i$

$Weight_i = \alpha + \beta_1 Age_i + \beta_2 Gender_i + \beta_3 Age_i * Gender_i + \epsilon_i$

But the variable *Gender* is a vector of string/factor. You will have to recode it as a numeric variable, otherwise R automatically converts it.

Interpretation:

Assuming that you have centered the continuous variable *age*, and you have coded *Gender* such that male is coded 0 and female is coded 1. - α is the weight of an average old male - β_1 is the effect of age on weight for male populations, to what extent weight of a male increases with age - β_2 is the effect of Gender on weight for average age population, i.e., for people with average age, to what extent females would have higher weight than males? - β_3 is called interaction effect. To what extent the effect of age on weight is larger for female population than male population? Slope for female population minus slope for male population.

We expect postive estimates for $alpha$ and β_1 and negative estimates for β_2 and β_3.

```
data.young$gender <- ifelse(data.young$Gender=="Male",0,1)
model3 <- lm(weight~c_age+gender+c_age*gender,data=data.young)
model3
```

```
##
## Call:
## lm(formula = weight ~ c_age + gender + c_age * gender, data = data.young)
##
## Coefficients:
##  (Intercept)         c_age        gender  c_age:gender
##       26.121        14.225        -2.114        -1.437
```

```
summary(model3)
```

```
##
```

```
## Call:
## lm(formula = weight ~ c_age + gender + c_age * gender, data = data.young)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.3471  -2.7141   0.0767   2.1963  15.5575
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   26.1210     0.4335  60.254  < 2e-16 ***
## c_age         14.2246     0.4442  32.022  < 2e-16 ***
## gender        -2.1141     0.5869  -3.602 0.000381 ***
## c_age:gender  -1.4374     0.5907  -2.433 0.015659 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.657 on 251 degrees of freedom
## Multiple R-squared:  0.8934, Adjusted R-squared:  0.8922
## F-statistic: 701.4 on 3 and 251 DF,  p-value: < 2.2e-16
```

Exercise 1

Load the data from a word recognition experiment: Participants had to recognize a word as quickly as possible; dataset lists the observed reaction time, frequency and length for each word.

1.1 We have a hypothesis that "the words which are more frequent in everyday use will be read faster". Fit a linear model to check this. Does the estimate support the prediction of the hypothesis?

1.2 What is the relationship between reaction time and word length?

1.3 Is there any interaction effect? What does the interaction imply?

```
hindi <- read.table("Hindi-word-recognition-data.csv",sep=",",header=T)
head(hindi)
```

```
##   X word   label frequency length reactionTime
## 1 1    1 hfshort  4.857125      2     749.3158
## 2 2    2 hfshort  4.905634      2     694.6500
## 3 3    3 lfshort  2.033606      3     779.1500
## 4 4    4 hfshort  4.640838      2     659.5263
## 5 5    5  hflong  3.785533      5     691.4583
## 6 6    6 hfshort  5.391810      2     614.1923
```

Linear mixed models

When data are drawn from a hierarchy of different populations!

Consider COVID-19 dataset:

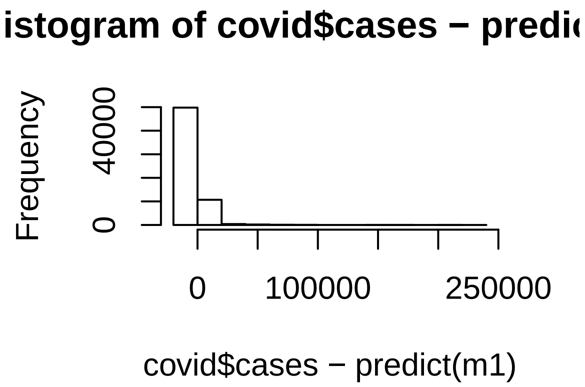
```
covid <- read.table("COVID19data.csv",sep=",",header=T)
covid <- subset(covid,year==2020)[,c(4,6,7,8,10)]
head(covid)
```

```
##   month cases deaths     country continentExp
## 1    12    63     10 Afghanistan         Asia
## 2    12   202     16 Afghanistan         Asia
## 3    12   135     13 Afghanistan         Asia
## 4    12   200      6 Afghanistan         Asia
## 5    12   210     26 Afghanistan         Asia
## 6    12   234     10 Afghanistan         Asia
```

```
m1 <- lm(cases~month,data=covid)
summary(m1)
```

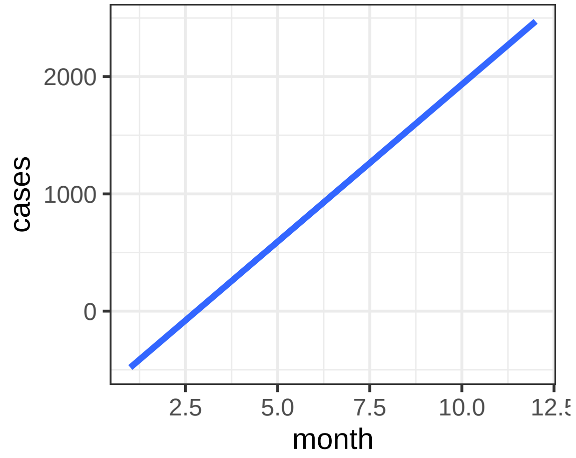
```
##
## Call:
## lm(formula = cases ~ month, data = covid)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -9927  -1398   -840   -145 229459
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -748.783     68.993  -10.85   <2e-16 ***
## month        268.316      9.084   29.54   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6569 on 61192 degrees of freedom
## Multiple R-squared:  0.01406,    Adjusted R-squared:  0.01404
## F-statistic: 872.5 on 1 and 61192 DF,  p-value: < 2.2e-16
```

```
hist(covid$cases-predict(m1))
```



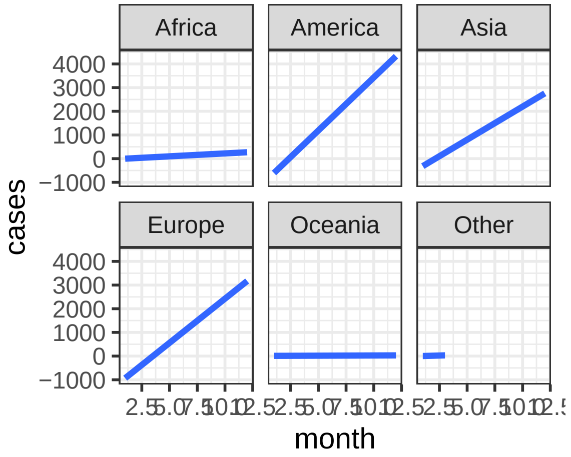
```
#However, residuals are not normally distributed, we will review this problem in gener

ggplot(covid,aes(x=month,y=cases))+
  geom_smooth(method="lm",size=1.1,se=F)+theme_bw()
```

```
## `geom_smooth()` using formula 'y ~ x'
```



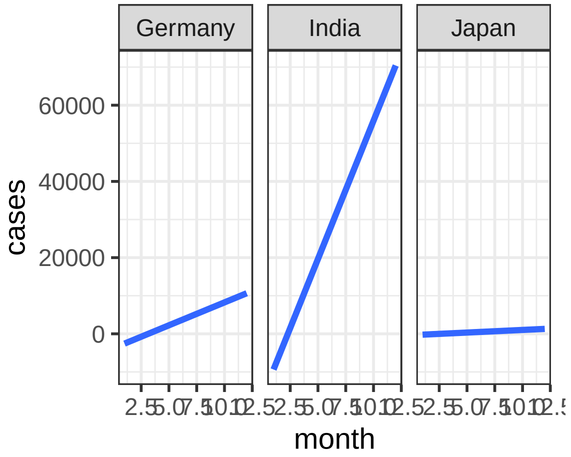
```
ggplot(covid,aes(x=month,y=cases))+
  geom_smooth(method="lm",size=1.1,se=F)+theme_bw()+
  facet_wrap(~continentExp)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



```
ggplot(subset(covid,country %in% c("India","Germany","Japan")),aes(x=month,y=cases))+
  geom_smooth(method="lm",size=1.1,se=F)+theme_bw()+
  facet_wrap(~country)
```

```
## `geom_smooth()` using formula 'y ~ x'
```



- the obsevations, i.e., number of cases per day, are not generated from the same (homogenous) population
- The observations come from a hierarchical of different populations
  - Number of cases per day comes from populations of different countries which come from a population of continents
- The observations are not independently drawn from a distribution

Let us build a linear model to predict number of cases from month number. We assume that number of cases increases from month 1 to month 12.

$Cases_i \ \alpha + \beta Month_i + \epsilon_i$

Let us rewrite this model assuming that the observations come from different subpopulations i.e., different countries.

Let i be the index for observations, j be the index for *countries*.

The model to predict the number of cases on i^{th} day for j^{th} country, i.e., $Cases_{i,j}$:

$Cases_{i,j} = \alpha_j + \beta_j Month_i$

Where α_j and β_j are intercept and slope for j^{th} country; $Month_i$ indicates to which month i^{th} day belongs to.

I assume that,

$\alpha_j = \alpha + u_j$

u_j is random intercept adjustment for j^{th} country.

$\beta_j = \beta + w_j$

w_j is slope intercept adjustment for j^{th} country.

Further, I assume that u_j and w_j come from a bivariate normal distribution,

$$\begin{pmatrix} u_j \\ w_j \end{pmatrix} \sim N_2 \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_u^2 & \rho\sigma_u\sigma_w \\ \rho\sigma_u\sigma_w & \sigma_w^2 \end{pmatrix} \right)$$

This is called random effect structure. The estimates of parameters σ_u, σ_w, and correlation ρ are called random effects.

$\hat{\alpha}$ and $\hat{\beta}$ are called fixed effects.

The linear mixed-effect models capture both fixed effects and random effects.

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
m2 <- lmer(cases ~ month+(1+month|country),data=covid)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
## Model failed to converge with max|grad| = 0.0095105 (tol = 0.002, component 1)
```

```
summary(m2)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: cases ~ month + (1 + month | country)
##    Data: covid
##
## REML criterion at convergence: 1163790
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -17.422  -0.026  -0.001   0.008  36.347
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  country  (Intercept)  6762374 2600
##           month        1080654 1040     -0.97
##  Residual             10346244 3217
## Number of obs: 61194, groups:  country, 214
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  -661.34     181.72  -3.639
## month         255.04      71.26   3.579
##
## Correlation of Fixed Effects:
##       (Intr)
## month -0.960
## convergence code: 0
## Model failed to converge with max|grad| = 0.0095105 (tol = 0.002, component 1)
```

Model was:

$Cases_{i,j} = (\alpha + u_j) + (\beta + w_j)Month_i + \epsilon_{i,j}$

$\begin{pmatrix} u_j \\ w_j \end{pmatrix} \sim N_2\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_u^2 & \rho\sigma_u\sigma_w \\ \rho\sigma_u\sigma_w & \sigma_w^2 \end{pmatrix}\right)$

This model is called correlated varying intercept varying slope model.

The parameter estimates are:

Fixed effects:

$\alpha = -661$

$\beta = 255$

Random effects:

$\sigma_u = 2600$

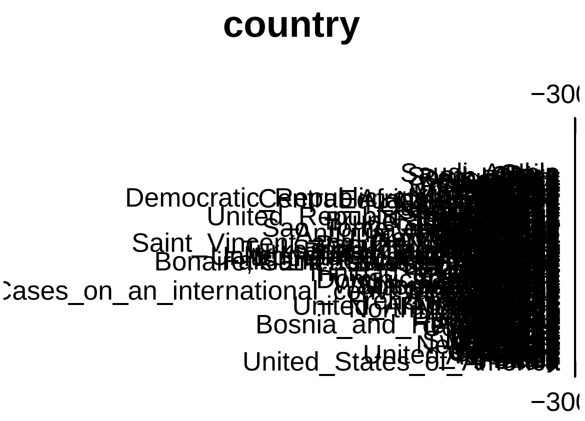
$\sigma_w = 1040$

$\rho = -0.97$

Visualize country-level random adjustments:

```r
library(lattice)
dotplot(ranef(m2,condVar=TRUE))
```

```
## $country
```



Let us make a slight change in the model:

$$\log(Cases_{i,j}) = (\alpha + u_j) + (\beta + w_j)Month_i + \epsilon_{i,j}$$

$$\begin{pmatrix} u_j \\ w_j \end{pmatrix} \sim N_2\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_u^2 & \rho\sigma_u\sigma_w \\ \rho\sigma_u\sigma_w & \sigma_w^2 \end{pmatrix}\right)$$

```r
m3 <- glmer(cases ~ month+(month|country),data=subset(covid,cases>0),family=poisson())
summary(m3)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: poisson  ( log )
## Formula: cases ~ month + (month | country)
##    Data: subset(covid, cases > 0)
##
##       AIC       BIC    logLik  deviance  df.resid
##  30632601  30632645 -15316296  30632591     41985
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -221.79   -8.48   -2.12    4.22  765.29
##
## Random effects:
##  Groups  Name        Variance Std.Dev. Corr
##  country (Intercept) 6.02434  2.4545
##          month       0.07799  0.2793   -0.49
## Number of obs: 41990, groups:  country, 214
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.85433    0.10753   26.55   <2e-16 ***
## month        0.19072    0.01775   10.74   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##       (Intr)
## month -0.342
```

This is a generalized linear model.

```
#m4 <- glmer(cases ~ month+(1|country)+(1|continentExp),data=subset(covid,cases>0))
#summary(m4)
```

Hypothesis testing

Frequentist perspective:

There exists a true population with fixed population parameters. If you repeatedly draw samples from the population, you can estimate the true parameters.

Consider the hypothesis that **average height of Indian people is greater than 5 feet**.

We are not going to directly target this hypothesis, we will first state a resonable null hypothesis.

Null hypothesis: **average height of Indian people is 5 feet** i.e., mu = 0

Alternative hypothesis: **average height of Indian people larger than 5 feet** i.e., mu > 0

The idea is to reject the null hypothesis. If we can prove that the null hypothesis is incorrect, we can say that there is evidence for the alternative hypothesis.

How to reject/accept the null?

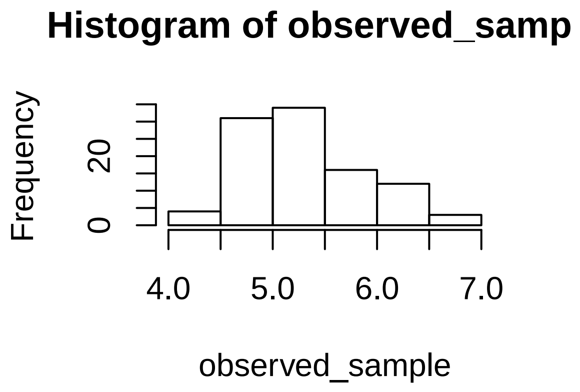
Assume that null hypothesis is TRUE. The true average height for the population is 5 feet. Draw repeated samples from this population. Calculate mean of each sample. If less than 5% of the samples have mean greater than observed sample mean, reject the null.

$p - value = Pr(T \geq t|H)$

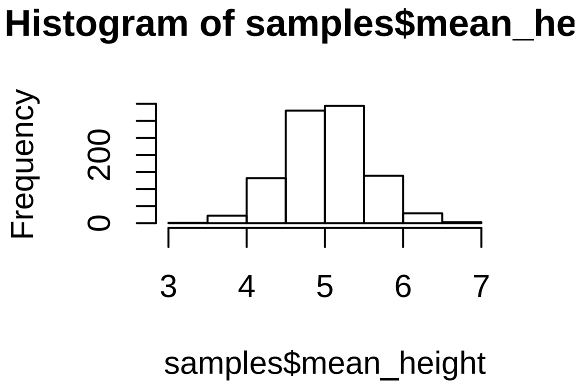
Where, T is the test statistic (mean in our example) of the samples drawn from the population, t is the mean of the observed sample.

Intuitively, $p - value$ tells you the probability of generating data at least as extreme as the observed data.

```
observed_sample <- rnorm(100,5.3,0.5)
hist(observed_sample)
```



```r
#Null hypothesis
mu_population <- 5
#repeated sampling
samples <- data.frame(matrix(ncol=2,nrow=1000))
colnames(samples) <- c("sample_id","mean_height")
for(i in 1:1000){
  h <- rnorm(100,mu_population,5)
  samples[i,] <- c(i,mean(h))
}
hist(samples$mean_height)
```



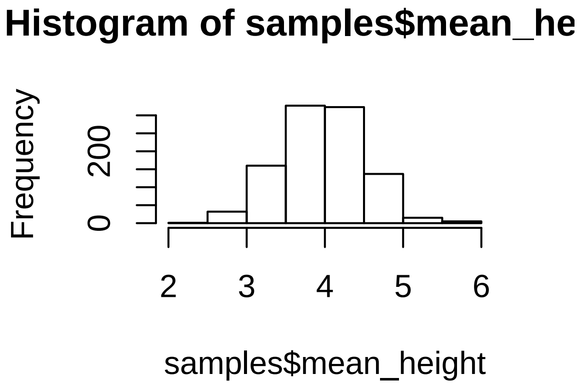
```r
sd(samples$mean_height)
```

```
## [1] 0.5205616
```

```r
pvalue <- length(which(samples$mean_height>=mean(observed_sample)))/1000
pvalue
```

```
## [1] 0.305
```

```
#We cannot reject the null hypothesis
#What if null hypothesis is that avergae height is 4
samples <- data.frame(matrix(ncol=2,nrow=1000))
colnames(samples) <- c("sample_id","mean_height")
for(i in 1:1000){
  h <- rnorm(100,4,5)
  samples[i,] <- c(i,mean(h))
}
hist(samples$mean_height)
```



```
sd(samples$mean_height)
```

```
## [1] 0.5197841
```

```
pvalue <- length(which(samples$mean_height>=mean(observed_sample)))/1000
pvalue
```

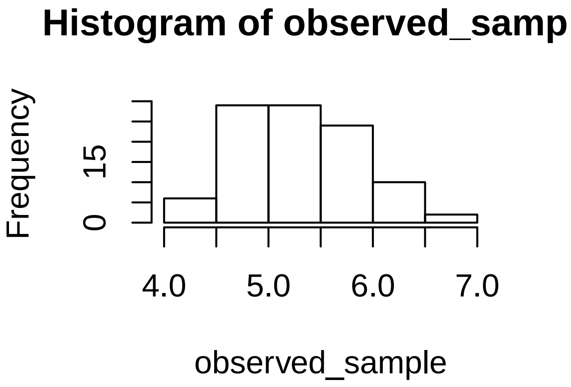
```
## [1] 0.007
```

```
# Reject the null
```

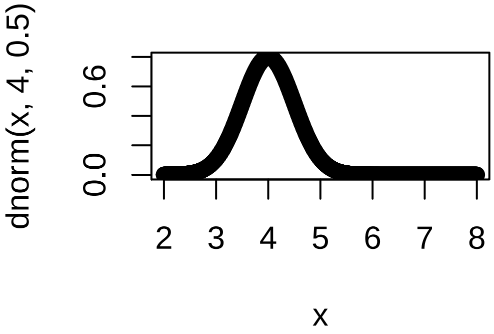
- the observed data is very unlikely to have occurred under the null hypothesis
- In other words, a p-value of 0.003 means if you draw repeated height samples from Indian population, 99.7% of the samples will have mean height larger than 5 feets.
- Problem: I cannot repeatdly sample from the population, this is practically impossible.
  - Central limit theorem!
  - Means of the samples drawn from a propulation are normally distributed.

- In practice,
  - First determine the distribution of test statistic of the samples drawn from null hypothesis
    - Usually, it is a student-t distribution or normal distribution
  - Determine the crtitical region for rejecting / accepting the null
    - For example, critical region to reject the null is $t > 1.8$ for t-distribution
  - Calculate p-value: you can simply calculate the area under the curve in the region $T > t$ where distribution has value greater than observed test statistic

```
observed_sample <- rnorm(100,5.3,0.5)
hist(observed_sample)
```



```
#Null hypothesis
mu_population <- 4
x <- seq(2,8,length=1000)
plot(x,dnorm(x,4,0.5))
```



```r
pvalue <- 1-pnorm(mean(observed_sample),mean=4,sd=0.5)
pvalue
```

```
## [1] 0.004741444
```

```r
# Reject the null
```

```r
summary(model3)
```

```
##
## Call:
## lm(formula = weight ~ c_age + gender + c_age * gender, data = data.young)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.3471  -2.7141   0.0767   2.1963  15.5575
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  26.1210     0.4335  60.254  < 2e-16 ***
## c_age        14.2246     0.4442  32.022  < 2e-16 ***
## gender       -2.1141     0.5869  -3.602 0.000381 ***
## c_age:gender -1.4374     0.5907  -2.433 0.015659 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.657 on 251 degrees of freedom
## Multiple R-squared:  0.8934, Adjusted R-squared:  0.8922
## F-statistic: 701.4 on 3 and 251 DF,  p-value: < 2.2e-16
```

Likelihood ratio test: model comparison

Which model has generated the observed data?

Quantify the likelihood that model1 has generated the observed data.

Model 1:

$\log(Cases_{i,j}) = (\alpha + u_j) + (\beta + w_j)Month_i + \epsilon_{i,j}$

$$\begin{pmatrix} u_j \\ w_j \end{pmatrix} \sim N_2\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_u^2 & \rho\sigma_u\sigma_w \\ \rho\sigma_u\sigma_w & \sigma_w^2 \end{pmatrix}\right)$$

Quantify the likelihood that model2 has generated the observed data.

Model 2:

$\log(Cases_{i,j}) = (\alpha + u_j) + \beta Month_i + \epsilon_{i,j}$

$u_j \sim N(0, \sigma_u^2)$

Quantify the likelihood that model3 has generated the observed data.

Model 3:

$\log(Cases_i) = \alpha + \beta Month_i + \epsilon_i$

```
m_full <- glmer(cases ~ month+(month|country),
                data=subset(covid,cases>0),
                family=poisson())
summary(m_full)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: poisson  ( log )
## Formula: cases ~ month + (month | country)
##    Data: subset(covid, cases > 0)
##
##      AIC       BIC    logLik  deviance  df.resid
##  30632601  30632645 -15316296  30632591     41985
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -221.79   -8.48   -2.12    4.22  765.29
##
## Random effects:
##  Groups  Name        Variance Std.Dev. Corr
##  country (Intercept) 6.02434  2.4545
##          month       0.07799  0.2793   -0.49
## Number of obs: 41990, groups:  country, 214
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.85433    0.10753   26.55   <2e-16 ***
## month        0.19072    0.01775   10.74   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##       (Intr)
## month -0.342
```

```
m_contrained <- glmer(cases ~ month+(1|country),
                      data=subset(covid,cases>0),
                      family=poisson())
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, :
```

```
## Model failed to converge with max|grad| = 0.00305735 (tol = 0.001, component 1)

## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model
##  - Rescale variables?;Model is nearly unidentifiable: large eigenvalue ratio
##  - Rescale variables?
```

```
summary(m_contrained)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: poisson  ( log )
## Formula: cases ~ month + (1 | country)
##    Data: subset(covid, cases > 0)
##
##       AIC       BIC    logLik  deviance  df.resid
##  38187355  38187381 -19093675  38187349     41987
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -231.88  -11.32   -3.02    4.66 1992.69
##
## Random effects:
##  Groups  Name        Variance Std.Dev.
##  country (Intercept) 6.022    2.454
## Number of obs: 41990, groups:  country, 214
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 2.242e+00  1.613e-01   13.91   <2e-16 ***
## month       2.784e-01  5.075e-05 5485.15   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##       (Intr)
## month -0.003
## convergence code: 0
## Model failed to converge with max|grad| = 0.00305735 (tol = 0.001, component 1)
## Model is nearly unidentifiable: very large eigenvalue
##  - Rescale variables?
## Model is nearly unidentifiable: large eigenvalue ratio
##  - Rescale variables?
```

```
m_null <- glm(cases ~ month,
                      data=subset(covid,cases>0),
                      family=poisson())
summary(m_null)
```

```
##
## Call:
## glm(formula = cases ~ month, family = poisson(), data = subset(covid,
##     cases > 0))
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
##  -91.86   -53.52   -36.19   -22.57  1184.12
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) 5.334e+00  4.806e-04   11098   <2e-16 ***
## month       2.513e-01  5.151e-05    4879   <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: 304738229  on 41989  degrees of freedom
## Residual deviance: 277581438  on 41988  degrees of freedom
## AIC: 277849337
##
## Number of Fisher Scoring iterations: 7
```

```
anova(m_full,m_contrained,m_null)
```

```
## Data: subset(covid, cases > 0)
## Models:
## m_null: cases ~ month
## m_contrained: cases ~ month + (1 | country)
## m_full: cases ~ month + (month | country)
##              Df       AIC       BIC     logLik  deviance     Chisq Chi Df
## m_null        2 277849337 277849354 -138924667 277849333
## m_contrained  3  38187355  38187381  -19093675  38187349 239661984      1
## m_full        5  30632601  30632644  -15316296  30632591   7554758      2
##              Pr(>Chisq)
## m_null
## m_contrained  < 2.2e-16 ***
## m_full        < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(m_full,m_contrained)
```

```
## Data: subset(covid, cases > 0)
## Models:
## m_contrained: cases ~ month + (1 | country)
## m_full: cases ~ month + (month | country)
##              Df      AIC      BIC    logLik deviance   Chisq Chi Df Pr(>Chisq)
## m_contrained  3 38187355 38187381 -19093675 38187349
## m_full        5 30632601 30632644 -15316296 30632591 7554758      2  < 2.2e-16
##
## m_contrained
## m_full       ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Next session: probability distributions, generalized linear models

Exercise 2

Fit a linear mixed model on the “fakedata” which has 50 groups and 100 observations for each group. Assume that group-level intercept come from a normal model with population-level intercept as the mean and group-level slopes are also normally distributed (but there is no correlation between the two). Find the estimates for each parameter.

```
sigma1 <- 2
sigma2 <- 0.4
intercept <- 3
slope <- 2
ngroup <- 50
u1 <- intercept + rnorm(ngroup,0,sigma1)
u2 <- slope + rnorm(ngroup,0,sigma2)
predictor <- rnorm(100,10,5)
fakedata <- data.frame(matrix(ncol=3,nrow=0))
colnames(fakedata) <- c("group","predictor","outcome")
for(i in 1:50){
  outcome <- u1[i]+u2[i]*predictor
  fakedata <- rbind(fakedata,data.frame(group=rep(i,100),predictor,outcome))
}
head(fakedata)
```

```
##   group predictor   outcome
## 1     1  9.489351  8.823066
## 2     1  6.458921  6.084204
## 3     1  4.947613  4.718303
## 4     1 20.270959 18.567340
## 5      1  4.690558  4.485981
## 6      1 10.032691  9.314129
```