Statistics-lecture-2024-01-10-confounds-multilevel-statistics

Adam Tallman

2024-01-09

library(ggdag)

##   
## Attaching package: 'ggdag'

## The following object is masked from 'package:stats':  
##   
## filter

library(ggplot2)  
library(VGAM)

## Loading required package: stats4

## Loading required package: splines

library(ggExtra)  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.3 ✔ readr 2.1.4  
## ✔ forcats 1.0.0 ✔ stringr 1.5.0  
## ✔ lubridate 1.9.3 ✔ tibble 3.2.1  
## ✔ purrr 1.0.2 ✔ tidyr 1.3.0

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks ggdag::filter(), stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(gridExtra)

##   
## Attaching package: 'gridExtra'  
##   
## The following object is masked from 'package:dplyr':  
##   
## combine

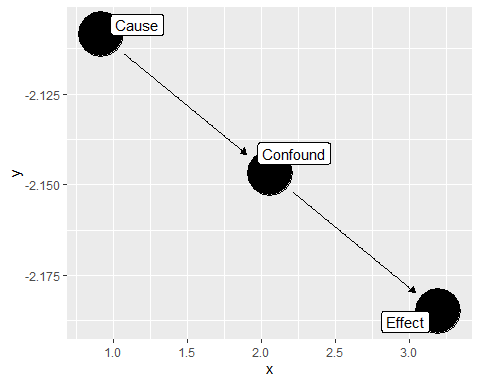
library(lme4)

## Loading required package: Matrix  
##   
## Attaching package: 'Matrix'  
##   
## The following objects are masked from 'package:tidyr':  
##   
## expand, pack, unpack

library(lmerTest)

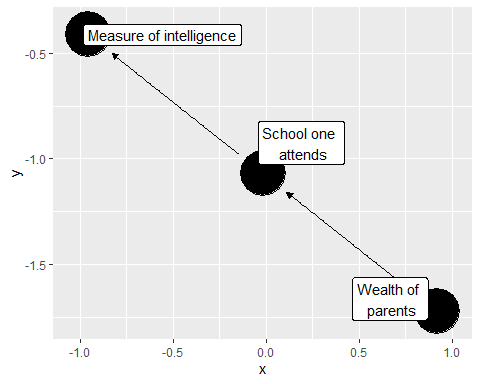
##   
## Attaching package: 'lmerTest'  
##   
## The following object is masked from 'package:lme4':  
##   
## lmer  
##   
## The following object is masked from 'package:stats':  
##   
## step

pipe.example <- dagify(Y~Z,  
 Z~X,  
 labels = c(  
 "X" = "Cause",  
 "Y" = "Effect",  
 "Z" = "Confound"  
 )  
 )  
ggdag(pipe.example, text=FALSE, use\_labels = "label")



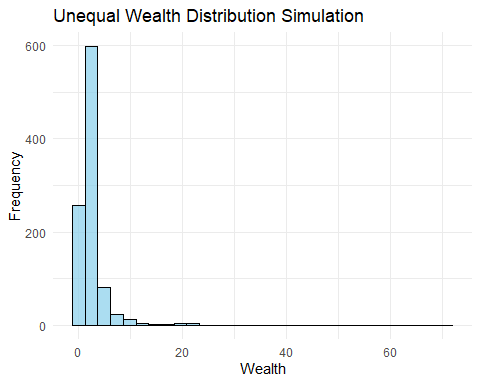
Let’s imagine we have the following situation.

pipe.example <- dagify(Y~Z,  
 Z~X,  
 labels = c(  
 "X" = "Wealth of \n parents",  
 "Y" = "Measure of intelligence",  
 "Z" = "School one \n attends"  
 )  
 )  
ggdag(pipe.example, text=FALSE, use\_labels = "label")



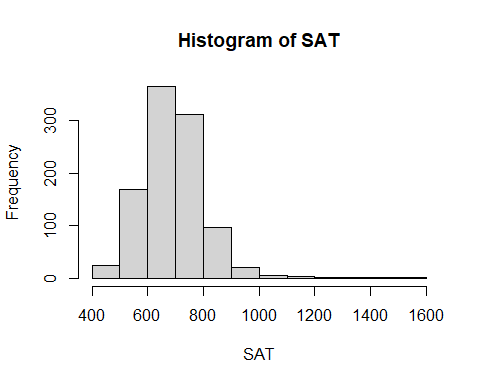
We can get something like a wealth distribution by simulating a pareto distribution. We can use the rpareto function in the VGAM package.

num\_individuals <- 1000  
alpha <- 1.5 #how unequal they are going to be  
wealth\_distribution <- rpareto(num\_individuals, shape=alpha)  
df <- data.frame(Wealth = wealth\_distribution)  
ggplot(df, aes(x = Wealth)) +  
 geom\_histogram(bins = 30, fill = "skyblue", color = "black", alpha = 0.7) +  
 labs(title = "Unequal Wealth Distribution Simulation",  
 x = "Wealth",  
 y = "Frequency") +  
 theme\_minimal()



x1 <- 0.7  
x2 <- 0.2  
yearly\_income\_of\_parents <- wealth\_distribution\*40000  
school\_attended\_quality <- yearly\_income\_of\_parents\*x1 + rnorm(n=1000, m=0, sd=4000)  
test\_taking\_ability <- school\_attended\_quality\*x2 + rnorm(n=1000, m=0, sd=30000)

test\_score <- (test\_taking\_ability - min(test\_taking\_ability)) / (max(test\_taking\_ability) - min(test\_taking\_ability))   
test\_score <- (test\_score \* 1600)+400  
SAT <- numeric(length(test\_score))  
# Loop through each element in the original vector  
for (i in seq\_along(test\_score)) {  
 # Check if the value is above 1600  
 if (test\_score[i] > 1600) {  
 # If yes, set it to 1600  
 SAT[i] <- 1600  
 } else {  
 # If no, keep the original value  
 SAT[i] <- test\_score[i]  
 }  
}  
hist(SAT)



df <- data.frame(wealth = yearly\_income\_of\_parents,  
 school = school\_attended\_quality,  
 SAT = SAT)  
  
summary(lm(SAT~wealth, data = df))

##   
## Call:  
## lm(formula = SAT ~ wealth, data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -266.753 -62.155 4.677 58.224 260.412   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6.474e+02 3.353e+00 193.10 <2e-16 \*\*\*  
## wealth 4.143e-04 1.578e-05 26.25 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 90.38 on 998 degrees of freedom  
## Multiple R-squared: 0.4085, Adjusted R-squared: 0.4079   
## F-statistic: 689.1 on 1 and 998 DF, p-value: < 2.2e-16

summary(lm(SAT~wealth+school, data = df))

##   
## Call:  
## lm(formula = SAT ~ wealth + school, data = df)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -266.759 -62.145 4.679 58.144 260.311   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6.474e+02 3.355e+00 192.983 <2e-16 \*\*\*  
## wealth 4.276e-04 5.139e-04 0.832 0.406   
## school -1.899e-05 7.337e-04 -0.026 0.979   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 90.42 on 997 degrees of freedom  
## Multiple R-squared: 0.4085, Adjusted R-squared: 0.4073   
## F-statistic: 344.2 on 2 and 997 DF, p-value: < 2.2e-16

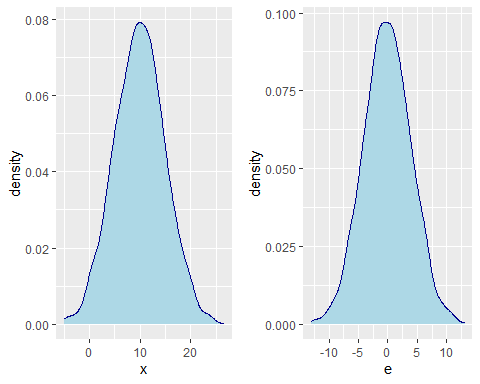
Okay, do we conclude from this that wealth is not a causal factor in our SAT scores? More appropriately, do we conclude that changing the wealth distribution will have an impact on general SAT scores? We couldn’t conclude that, because we blocked the causal chain from wealth to SAT scores.

## Multilevel models

### Conceptual introduction

Here is out linear model

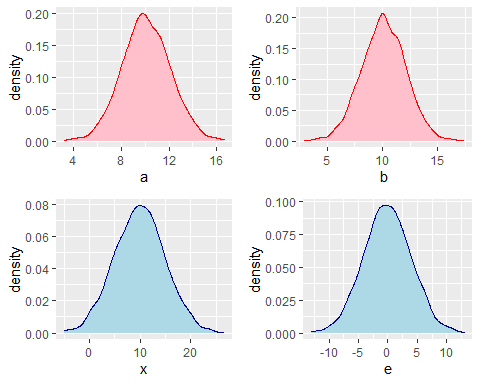
e <- rnorm(m=0, sd=4, n=2000)  
x <- rnorm(m=10, sd=5, n= 2000)  
a <- 10  
b <- 5  
y <- a + b \* x + e  
d <- data.frame(x, e)  
  
plot\_x <- ggplot(d, aes(x=x))+  
 geom\_density(linetype="solid", fill="lightblue", color="darkblue")  
  
plot\_e <- ggplot(d, aes(x=e))+  
 geom\_density(linetype="solid",fill="lightblue", color="darkblue")  
  
grid.arrange(plot\_x, plot\_e, ncol=2)



In this model we have two random variables.

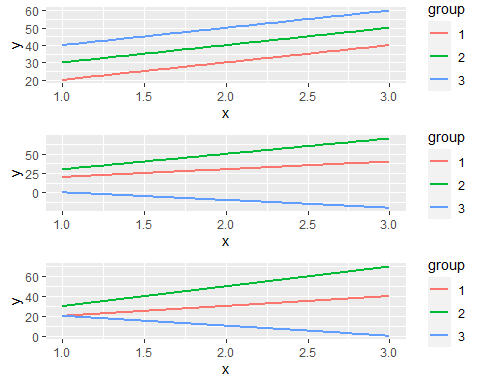
### Multilevel model

a <- rnorm(m=10, sd=2, n=2000)  
b <- rnorm(m=10, sd=2, n= 2000)  
plot\_a <- ggplot(d, aes(x=a))+  
 geom\_density(linetype="solid", fill="pink", color="red")  
  
plot\_b <- ggplot(d, aes(x=b))+  
 geom\_density(linetype="solid", fill="pink", color="red")  
  
grid.arrange(plot\_a, plot\_b, plot\_x, plot\_e, ncol=2)



x <- seq(1, 3, by=0.1 )  
a1 <- 10  
a2 <- 20  
a3 <- 30  
b1 <- 10  
b2 <- 10  
b3 <- 10  
y1 <- a1 + x\*b1  
d1 <- data.frame(y=y1,x)  
y2 <- a2 + x\*b2  
d2 <- data.frame(y=y2,x)  
y3 <- a3 + x\*b3  
d3 <- data.frame(y=y3,x)  
d1$group = "1"  
d2$group = "2"  
d3$group = "3"  
d <- rbind(d1,d2,d3)   
plot\_01 <- ggplot(d, aes(x=x, y=y, shape=group))+  
 geom\_smooth(method=lm, se=FALSE, fullrange=TRUE,  
 aes(color=group))  
  
x <- seq(1, 3, by=0.1 )  
a1 <- 10  
a2 <- 10  
a3 <- 10  
b1 <- 10  
b2 <- 20  
b3 <- -10  
y1 <- a1 + x\*b1  
d1 <- data.frame(y=y1,x)  
y2 <- a2 + x\*b2  
d2 <- data.frame(y=y2,x)  
y3 <- a3 + x\*b3  
d3 <- data.frame(y=y3,x)  
d1$group = "1"  
d2$group = "2"  
d3$group = "3"  
d <- rbind(d1,d2,d3)   
plot\_02 <- ggplot(d, aes(x=x, y=y, shape=group))+  
 geom\_smooth(method=lm, se=FALSE, fullrange=TRUE,  
 aes(color=group))  
  
x <- seq(1, 3, by=0.1 )  
a1 <- 10  
a2 <- 10  
a3 <- 30  
b1 <- 10  
b2 <- 20  
b3 <- -10  
y1 <- a1 + x\*b1  
d1 <- data.frame(y=y1,x)  
y2 <- a2 + x\*b2  
d2 <- data.frame(y=y2,x)  
y3 <- a3 + x\*b3  
d3 <- data.frame(y=y3,x)  
d1$group = "1"  
d2$group = "2"  
d3$group = "3"  
d <- rbind(d1,d2,d3)   
plot\_03 <- ggplot(d, aes(x=x, y=y, shape=group))+  
 geom\_smooth(method=lm, se=FALSE, fullrange=TRUE,  
 aes(color=group))  
  
grid.arrange(plot\_01, plot\_02, plot\_03, nrow=3)

## `geom\_smooth()` using formula = 'y ~ x'  
## `geom\_smooth()` using formula = 'y ~ x'  
## `geom\_smooth()` using formula = 'y ~ x'



## Simulated example

set.seed(200)  
  
## School 1  
N1 <- round(runif(1,10,120))  
N <- N1   
grades <- runif(N,0,10)  
training <- rbinom(N,1,(0.2+grades/20))  
gender <- rbinom(N,1,0.5)  
b1 <- rnorm(1,2,0.5)  
b2 <- rnorm(1,2,0.5)  
b3 <- rnorm(1,0,2)  
a <- rnorm(1,5,2)  
aptitude <- a + b1\*grades + b2\*training +b3\*gender+rnorm(N, sd=1)  
school1 <- data.frame(Grades =grades,  
 Training = training,   
 Aptitude = aptitude,  
 Gender = factor(gender, levels=c(0,1),  
 labels = c("M", "F")))

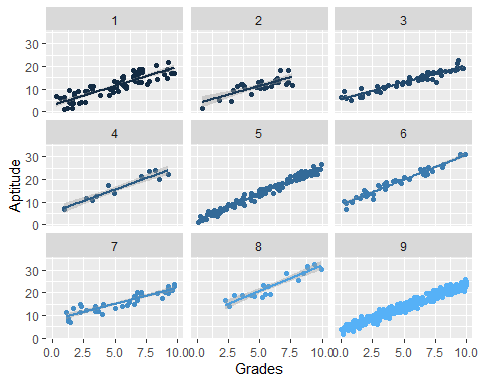
# School 2  
N2 <- round(runif(1,10,120))   
N <- N2  
grades <- runif(N, 0, 10)  
training <- rbinom(N, 1, (0.2+grades/20))  
gender <- rbinom(N, 1, 0.5)  
b1 <- rnorm(1,2,0.5)  
b2 <- rnorm(1,2,0.5)  
b3 <- rnorm(1,0,2)  
a <- rnorm(1,5,2)  
aptitude <- a + b1\*grades + b2\*training + b3\*gender + rnorm(N, sd= 1)   
school2 <- data.frame(Grades = grades,  
 Training = training,  
 Aptitude = aptitude,  
 Gender = factor(gender, levels=c(0, 1),  
 labels=c("M", "F")))  
  
N3 <- round(runif(1,10,120))  
N <- N3  
grades <- runif(N, 0, 10)  
training <- rbinom(N, 1, (0.2+grades/20))  
gender <- rbinom(N, 1, 0.5)  
b1 <- rnorm(1,2,0.5)  
b2 <- rnorm(1,2,0.5)  
b3 <- rnorm(1,0,2)  
a <- rnorm(1,5,2)  
aptitude <- a + b1\*grades + b2\*training + b3\*gender + rnorm(N, sd= 1)   
school3 <- data.frame(Grades = grades,  
 Training = training,  
 Aptitude = aptitude,  
 Gender = factor(gender, levels=c(0, 1),  
 labels=c("M", "F")))  
  
  
N4 <- round(runif(1,10,120))  
N <- N4  
grades <- runif(N, 0, 10)  
training <- rbinom(N, 1, (0.2+grades/20))  
gender <- rbinom(N, 1, 0.5)  
b1 <- rnorm(1,2,0.5)  
b2 <- rnorm(1,2,0.5)  
b3 <- rnorm(1,0,2)  
a <- rnorm(1,5,2)  
aptitude <- a + b1\*grades + b2\*training + b3\*gender + rnorm(N, sd= 1)   
school4 <- data.frame(Grades = grades,  
 Training = training,  
 Aptitude = aptitude,  
 Gender = factor(gender, levels=c(0, 1),  
 labels=c("M", "F")))  
  
  
  
N5 <- round(runif(1,10,120))  
N <- N5  
grades <- runif(N, 0, 10)  
training <- rbinom(N, 1, (0.2+grades/20))  
gender <- rbinom(N, 1, 0.5)  
b1 <- rnorm(1,2,0.5)  
b2 <- rnorm(1,2,0.5)  
b3 <- rnorm(1,0,2)  
a <- rnorm(1,5,2)  
aptitude <- a + b1\*grades + b2\*training + b3\*gender + rnorm(N, sd= 1)   
school5 <- data.frame(Grades = grades,  
 Training = training,  
 Aptitude = aptitude,  
 Gender = factor(gender, levels=c(0, 1),  
 labels=c("M", "F")))  
  
  
N6 <- round(runif(1,10,120))  
N <- N6  
grades <- runif(N, 0, 10)  
training <- rbinom(N, 1, (0.2+grades/20))  
gender <- rbinom(N, 1, 0.5)  
b1 <- rnorm(1,2,0.5)  
b2 <- rnorm(1,2,0.5)  
b3 <- rnorm(1,0,2)  
a <- rnorm(1,5,2)  
aptitude <- a + b1\*grades + b2\*training + b3\*gender + rnorm(N, sd= 1)   
school6 <- data.frame(Grades = grades,  
 Training = training,  
 Aptitude = aptitude,  
 Gender = factor(gender, levels=c(0, 1),  
 labels=c("M", "F")))  
  
N7 <- round(runif(1,10,120))  
N <- N7  
grades <- runif(N, 0, 10)  
training <- rbinom(N, 1, (0.2+grades/20))  
gender <- rbinom(N, 1, 0.5)  
b1 <- rnorm(1,2,0.5)  
b2 <- rnorm(1,2,0.5)  
b3 <- rnorm(1,0,2)  
a <- rnorm(1,5,2)  
aptitude <- a + b1\*grades + b2\*training + b3\*gender + rnorm(N, sd= 1)   
school7 <- data.frame(Grades = grades,  
 Training = training,  
 Aptitude = aptitude,  
 Gender = factor(gender, levels=c(0, 1),  
 labels=c("M", "F")))  
  
N8 <- round(runif(1,10,120))  
N <- N8  
grades <- runif(N, 0, 10)  
training <- rbinom(N, 1, (0.2+grades/20))  
gender <- rbinom(N, 1, 0.5)  
b1 <- rnorm(1,2,0.5)  
b2 <- rnorm(1,2,0.5)  
b3 <- rnorm(1,0,2)  
a <- rnorm(1,5,2)  
aptitude <- a + b1\*grades + b2\*training + b3\*gender + rnorm(N, sd= 1)   
school8 <- data.frame(Grades = grades,  
 Training = training,  
 Aptitude = aptitude,  
 Gender = factor(gender, levels=c(0, 1),  
 labels=c("M", "F")))  
  
N9 <- 300  
N <- N9  
grades <- runif(N, 0, 10)  
training <- rbinom(N, 1, (0.2+grades/20))  
gender <- rbinom(N, 1, 0.5)  
b1 <- rnorm(1,2,0.5)  
b2 <- -1.5  
b3 <- rnorm(1,0,2)  
a <- rnorm(1,5,2)  
aptitude <- a + b1\*grades + b2\*training + b3\*gender + rnorm(N, sd= 1)   
school9 <- data.frame(Grades = grades,  
 Training = training,  
 Aptitude = aptitude,  
 Gender = factor(gender, levels=c(0, 1),  
 labels=c("M", "F")))  
  
data <- rbind(school1,school2,school3,school4,school5,school6,school7,school8,school9)

data <- rbind(school1,school2,school3,school4,school5,school6,school7,school8,school9)  
  
data$School <- c(rep(1, N1), rep(2, N2), rep(3, N3), rep(4, N4), rep(5, N5), rep(6, N6), rep(7, N7), rep(8, N8), rep(9, N9))

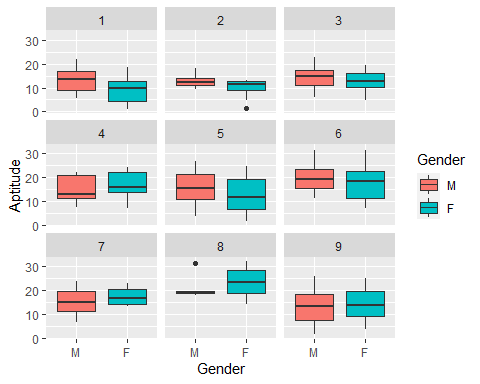
##Plotting  
library(ggplot2)

plot.grades.apt <-ggplot(data = data, aes(x = Grades, y = Aptitude, group=School))+   
 facet\_wrap( ~ School, ncol=3)+   
 geom\_point(aes(colour = School))+   
 geom\_smooth(method = "lm", se = TRUE, aes(colour = School))+   
 xlab("Grades")+ylab("Aptitude")+   
 theme(legend.position = "none")   
plot.grades.apt

## `geom\_smooth()` using formula = 'y ~ x'



plot.gender.apt <- ggplot(data, aes(x=Gender, y=Aptitude, group=Gender)) +   
 facet\_wrap( ~ School, ncol=3)+  
 geom\_boxplot(aes(fill=Gender))  
  
plot.gender.apt



mod.9 <- lm(Aptitude~Gender, data=school9)  
summary(mod.9)

##   
## Call:  
## lm(formula = Aptitude ~ Gender, data = school9)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -11.6096 -5.6993 -0.1237 5.2906 12.6406   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 13.3573 0.4853 27.52 <2e-16 \*\*\*  
## GenderF 0.6887 0.7029 0.98 0.328   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 6.081 on 298 degrees of freedom  
## Multiple R-squared: 0.003211, Adjusted R-squared: -0.0001338   
## F-statistic: 0.96 on 1 and 298 DF, p-value: 0.328

## all pooled  
M1.pooling <- lm(Aptitude~Training+Grades+Gender, data=data)  
summary(M1.pooling)

##   
## Call:  
## lm(formula = Aptitude ~ Training + Grades + Gender, data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -7.0726 -1.5955 -0.2555 1.0650 10.5648   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.31022 0.23985 17.971 < 2e-16 \*\*\*  
## Training 0.38766 0.22023 1.760 0.078813 .   
## Grades 1.96354 0.03889 50.483 < 2e-16 \*\*\*  
## GenderF -0.78851 0.20878 -3.777 0.000173 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.7 on 669 degrees of freedom  
## Multiple R-squared: 0.8134, Adjusted R-squared: 0.8126   
## F-statistic: 972 on 3 and 669 DF, p-value: < 2.2e-16

M1.nopooling <- lm(Aptitude~Training\*as.factor(School)+Grades+Gender, data=data)  
summary(M1.nopooling)

##   
## Call:  
## lm(formula = Aptitude ~ Training \* as.factor(School) + Grades +   
## Gender, data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -4.8581 -0.9808 -0.0501 0.9373 5.4153   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.60371 0.26213 6.118 1.63e-09 \*\*\*  
## Training 0.32985 0.37789 0.873 0.38306   
## as.factor(School)2 0.12195 0.47372 0.257 0.79693   
## as.factor(School)3 1.65296 0.36754 4.497 8.14e-06 \*\*\*  
## as.factor(School)4 3.00195 0.59074 5.082 4.89e-07 \*\*\*  
## as.factor(School)5 1.92047 0.31065 6.182 1.11e-09 \*\*\*  
## as.factor(School)6 7.50979 0.45193 16.617 < 2e-16 \*\*\*  
## as.factor(School)7 2.06768 0.44210 4.677 3.54e-06 \*\*\*  
## as.factor(School)8 7.42753 0.62648 11.856 < 2e-16 \*\*\*  
## as.factor(School)9 3.07538 0.26889 11.437 < 2e-16 \*\*\*  
## Grades 2.00206 0.02254 88.806 < 2e-16 \*\*\*  
## GenderF -0.71158 0.11996 -5.932 4.85e-09 \*\*\*  
## Training:as.factor(School)2 -0.45907 0.69988 -0.656 0.51210   
## Training:as.factor(School)3 -0.76998 0.55545 -1.386 0.16615   
## Training:as.factor(School)4 2.83892 0.94810 2.994 0.00285 \*\*   
## Training:as.factor(School)5 0.84539 0.47182 1.792 0.07364 .   
## Training:as.factor(School)6 1.26892 0.62503 2.030 0.04274 \*   
## Training:as.factor(School)7 1.86873 0.63304 2.952 0.00327 \*\*   
## Training:as.factor(School)8 3.59232 0.82955 4.330 1.72e-05 \*\*\*  
## Training:as.factor(School)9 -1.63250 0.41403 -3.943 8.92e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.528 on 653 degrees of freedom  
## Multiple R-squared: 0.9417, Adjusted R-squared: 0.94   
## F-statistic: 555 on 19 and 653 DF, p-value: < 2.2e-16

M0 <- lmer(Aptitude~1 +(1|School)+Grades, data=data)  
  
M1 <- lmer(Aptitude~Training + (1|School)+Grades, data=data )   
M2 <- lmer(Aptitude ~Training + (1+Training|School)+Grades, data=data)  
  
summary(M2)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [  
## lmerModLmerTest]  
## Formula: Aptitude ~ Training + (1 + Training | School) + Grades  
## Data: data  
##   
## REML criterion at convergence: 2580.4  
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -3.4110 -0.5672 -0.0466 0.5983 3.7479   
##   
## Random effects:  
## Groups Name Variance Std.Dev. Corr  
## School (Intercept) 7.494 2.737   
## Training 2.727 1.651 0.61  
## Residual 2.454 1.567   
## Number of obs: 673, groups: School, 9  
##   
## Fixed effects:  
## Estimate Std. Error df t value Pr(>|t|)   
## (Intercept) 4.29763 0.92697 8.09347 4.636 0.00162 \*\*   
## Training 0.96485 0.57773 7.69060 1.670 0.13497   
## Grades 2.00529 0.02306 659.68284 86.976 < 2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr) Tranng  
## Training 0.548   
## Grades -0.115 -0.049

anova(M1,M2,M0)

## refitting model(s) with ML (instead of REML)

## Data: data  
## Models:  
## M0: Aptitude ~ 1 + (1 | School) + Grades  
## M1: Aptitude ~ Training + (1 | School) + Grades  
## M2: Aptitude ~ Training + (1 + Training | School) + Grades  
## npar AIC BIC logLik deviance Chisq Df Pr(>Chisq)   
## M0 4 2684.6 2702.6 -1338.3 2676.6   
## M1 5 2686.5 2709.1 -1338.2 2676.5 0.0551 1 0.8145   
## M2 7 2590.7 2622.2 -1288.3 2576.7 99.8455 2 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

M0 <- lmer(Aptitude~1+(1|School), data=data)  
M1.nopooling <- lm(Aptitude~Gender\*Training+as.factor(School), data=data)  
summary(M1.nopooling)

##   
## Call:  
## lm(formula = Aptitude ~ Gender \* Training + as.factor(School),   
## data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -13.6615 -4.0825 -0.1717 4.2698 13.9034   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 10.5227 0.7483 14.063 < 2e-16 \*\*\*  
## GenderF -2.0561 0.6016 -3.418 0.000670 \*\*\*  
## Training 2.5255 0.6032 4.187 3.21e-05 \*\*\*  
## as.factor(School)2 -0.1428 1.2579 -0.114 0.909649   
## as.factor(School)3 2.0709 0.9965 2.078 0.038080 \*   
## as.factor(School)4 4.6962 1.6728 2.807 0.005141 \*\*   
## as.factor(School)5 2.9466 0.8452 3.487 0.000522 \*\*\*  
## as.factor(School)6 7.0749 1.1045 6.405 2.85e-10 \*\*\*  
## as.factor(School)7 4.3805 1.1414 3.838 0.000136 \*\*\*  
## as.factor(School)8 11.3559 1.4690 7.730 4.01e-14 \*\*\*  
## as.factor(School)9 2.3610 0.7402 3.190 0.001491 \*\*   
## GenderF:Training 2.2031 0.8626 2.554 0.010871 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 5.53 on 661 degrees of freedom  
## Multiple R-squared: 0.2264, Adjusted R-squared: 0.2135   
## F-statistic: 17.59 on 11 and 661 DF, p-value: < 2.2e-16

M1 <- lmer(Aptitude~Gender+(1+Gender|School), data=data)  
summary(M1)

## Linear mixed model fit by REML. t-tests use Satterthwaite's method [  
## lmerModLmerTest]  
## Formula: Aptitude ~ Gender + (1 + Gender | School)  
## Data: data  
##   
## REML criterion at convergence: 4298.2  
##   
## Scaled residuals:   
## Min 1Q Median 3Q Max   
## -2.04878 -0.79974 0.00018 0.78144 2.21685   
##   
## Random effects:  
## Groups Name Variance Std.Dev. Corr  
## School (Intercept) 7.310 2.704   
## GenderF 4.385 2.094 0.56  
## Residual 33.443 5.783   
## Number of obs: 673, groups: School, 9  
##   
## Fixed effects:  
## Estimate Std. Error df t value Pr(>|t|)   
## (Intercept) 15.4677 0.9909 6.3088 15.61 2.81e-06 \*\*\*  
## GenderF -0.6881 0.8939 6.3630 -0.77 0.469   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Correlation of Fixed Effects:  
## (Intr)  
## GenderF 0.236