Simulating the data and calculating the beta

```
library(tidyverse)
```

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

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1

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

```
library(knitr)
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
knitr::opts_chunk$set(cache = FALSE, warning = FALSE, message = FALSE, cache.lazy = FALSE)
```

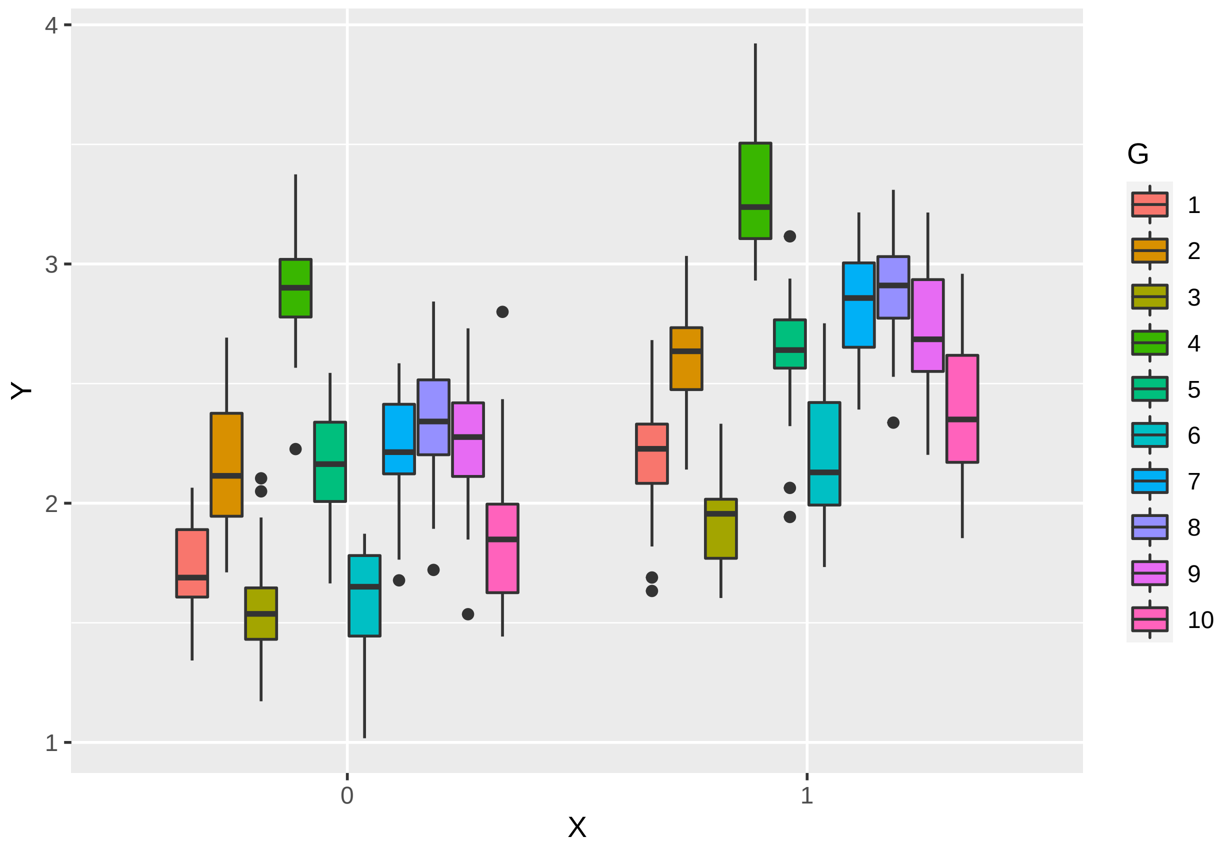
Initialize

```
set.seed(1)
K <- 10 # Nombre de groupe
nK <- 50 # Nombre d'observations par groupe
N <- K * nK # Nombre total d'observation
G <- factor(rep(1:K, each = nK))
intercept <- 2
fixefEffect <- .5
aleaEffect <- rnorm(K, sd = .5)
bias <- rnorm(N, sd = .25)
X <- rbinom(N,size=1,prob = .5)
Y <- intercept + fixefEffect * X + aleaEffect[G] + bias

df <- data.frame(X, Y, G)
```

```r
H0 <- formula(Y ~ X)
H1 <- formula(Y ~ X + (1|G))


dataPlot = cbind.data.frame(X=factor(X),Y, G = factor(G))
ggplot(dataPlot)+geom_boxplot(aes(x=X, y=Y, fill = G))
```



linear model

```r
X_prime <- cbind(1, X)
lmModel <- lm(H0)
betaLm <- lmModel$coefficients
betaLm
```

```
## (Intercept)           X
##   2.0710855   0.4996166
```

```r
vcov(lmModel)
```

```
##               (Intercept)             X
## (Intercept)  0.0007626236 -0.0007626236
## X           -0.0007626236  0.0016651170
```

```
Sig2 <- summary(lmModel)$sigma^2; Sig2
```

```
## [1] 0.206671
```

```
Sigma <- diag(Sig2,N)
SD_beta <- solve(crossprod(X_prime)) %*% crossprod(X_prime, Sigma) %*% X_prime %*% solve(crossprod(X_pr
sqrt(diag(SD_beta))[2]
```

```
##          X
## 0.04080585
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

mixed model

```
lmerModel <- lme4::lmer(H1, data=df)
betaLmer <- lme4::fixef(lmerModel)
betaLmer
```

```
## (Intercept)           X
##   2.0770172   0.4866651
```

```
vcov(lmerModel) %>% as.matrix() %>% diag() %>% sqrt()
```

```
## (Intercept)           X
##  0.12666446  0.02275628
```

```
sqrt(var(as.vector(lme4::fixef(lmerModel) %*% t(lmerModel@pp$X))))
```

```
## [1] 0.2427154
```

OLS

```
options(warn=-1)
OLS <- function(Y, X){
  modelmat <- model.matrix(~.,cbind.data.frame(X=X))
  indexes_X <- which(substring(colnames(modelmat), 1, 1) == "X")
  modX_OLS <- modelmat[, c(1, indexes_X), drop = FALSE]
  H <- (modX_OLS%*%solve(crossprod(modX_OLS))%*%t(modX_OLS))[indexes_X, , drop=FALSE]
  Y <- as.numeric(Y)
  betaOLS <- solve(crossprod(modX_OLS))%*%(t(modX_OLS)%*%Y)
  k <- ncol(modX_OLS)
  n <- nrow(modX_OLS)
```

```
  residuals <- as.matrix(Y - (betaOLS[1, , drop=FALSE]) - X * betaOLS[indexes_X, , drop=FALSE])
  RSS <- as.numeric(t(residuals)%*%residuals)
  Sigma2 <- as.numeric(RSS/(n-k))
  Vb <- Sigma2*solve(t(X)%*% X)
  SEb <- sqrt(diag(Vb))
  OLSCOV <- 1/(n-k) * as.numeric(t(residuals)%*%residuals) * solve(t(modX_OLS)%*%modX_OLS)
  SD <- sqrt(diag(OLSCOV))
  return(list('betaOLS'=betaOLS, 'SE'=SEb, 'OLSCOV'=OLSCOV, "SD"=SD))
}

OLS(Y,X)
```

```
## $betaOLS
##                  [,1]
## (Intercept) 2.0710855
## X           0.4996166
##
## $SE
## [1] 0.03004153
##
## $OLSCOV
##               (Intercept)             X
## (Intercept)  0.0007626236 -0.0007626236
## X           -0.0007626236  0.0016651170
##
## $SD
## (Intercept)           X
##  0.02761564  0.04080585
```

Monte Carlo

```
simOLS <- function (intercept=2, fixefEffects = c(.5,2,5,10), sds = c(.5,4,8,10),
                    K=10, nK=50, sims=5000){
  resAllOLS<- list()
  resAllMM<- list()
  n <- K * nK
  for (fixefEffect in fixefEffects) {
    for (sdUnit in sds) {
        aleaEffect <- rnorm(K, sd = sdUnit)
        G <- factor(rep(1:K, each = nK))
        resOLS <- matrix(0, sims, 3)
        resMM <- matrix(0, sims, 3)
        X <- rbinom(n,size=1,prob = .5)

        for(i in 1:sims){
          options(warn=-1)
          bias <- rnorm(n, sd = .25)
          Y <- intercept + fixefEffect * X + aleaEffect[G] + bias

          modOLS <- OLS(Y,X)
```

```r
          resOLS[i,] <- c(modOLS$betaOLS[2,], modOLS$SE, modOLS$SD[2])


          lmerModel <- lme4::lmer(Y ~ X + (1|G),REML = TRUE)
          beta <- lme4::fixef(lmerModel)[-1]
          SE_b <- sqrt(diag(as.matrix(vcov(lmerModel))))[-1]
          SD_b <- sqrt(stats::var(as.vector(beta %*% t(X))))
          resMM[i,] <- c(beta, SE_b, SD_b)
        }

        M <- apply(resOLS, 2, mean)
        S <- apply(resOLS, 2, sd)

        H_ols <- matrix(c(M[1], S[1], M[2], S[2], M[3], S[3]), ncol = 2, byrow = TRUE)


        M <- apply(resMM, 2, mean)
        S <- apply(resMM, 2, sd)
        H_mm <- matrix(c(M[1], S[1], M[2], S[2], M[3], S[3]), ncol = 2, byrow = TRUE)

        dimnames(H_ols) <- dimnames(H_mm) <- list( c('Beta','Standard Error',
                                                     'Standard Deviation'), c('mean', 'se'))


        resAllOLS[[as.character(paste0(fixefEffect,"_",sdUnit))]] <- H_ols
        resAllMM[[as.character(paste0(fixefEffect,"_",sdUnit))]] <- H_mm
    }

  }

  return (list('resAllOLS'=resAllOLS, 'resAllMM'=resAllMM))
}

resSim <- simOLS(sims = 100)
```

			$\boldsymbol{\Sigma = 0.5}$		$\boldsymbol{\Sigma = 4}$		$\boldsymbol{\Sigma = 8}$		$\boldsymbol{\Sigma = 10}$	
Model	$\boldsymbol{\beta}$	Estimate	μ	SE	μ	SE	μ	SE	μ	SE
$\boldsymbol{\beta = 0.5}$	OLS	$\hat{\boldsymbol{\beta}}$	0.515	0.022	0.627	0.023	0.719	0.021	0.711	0.024
		$\boldsymbol{SE}(\hat{\boldsymbol{\beta}})$	0.033	0.001	0.264	0.001	0.401	0.001	0.586	0.001
		$\boldsymbol{SD}(\hat{\boldsymbol{\beta}})$	0.046	0.001	0.366	0.001	0.579	0.001	0.824	0.001
	MM	$\hat{\boldsymbol{\beta}}$	0.5	0.023	0.5	0.023	0.5	0.021	0.498	0.024
		$\boldsymbol{SE}(\hat{\boldsymbol{\beta}})$	0.023	0.001	0.022	0.001	0.023	0.001	0.022	0.001
		$\boldsymbol{SD}(\hat{\boldsymbol{\beta}})$	0.25	0.011	0.25	0.011	0.25	0.011	0.249	0.012
$\boldsymbol{\beta = 2}$	OLS	$\hat{\boldsymbol{\beta}}$	2.02	0.022	1.737	0.022	2.764	0.023	2.816	0.023
		$\boldsymbol{SE}(\hat{\boldsymbol{\beta}})$	0.031	0.001	0.253	0.001	0.457	0.001	0.796	0.001
		$\boldsymbol{SD}(\hat{\boldsymbol{\beta}})$	0.045	0.001	0.368	0.001	0.661	0.001	1.137	0.001
	MM	$\hat{\boldsymbol{\beta}}$	1.998	0.023	1.998	0.023	2.001	0.023	2	0.023
		$\boldsymbol{SE}(\hat{\boldsymbol{\beta}})$	0.023	0.001	0.022	0.001	0.023	0.001	0.022	0.001
		$\boldsymbol{SD}(\hat{\boldsymbol{\beta}})$	0.999	0.011	0.998	0.011	1	0.011	1.001	0.012
$\boldsymbol{\beta = 5}$	OLS	$\hat{\boldsymbol{\beta}}$	5.008	0.022	4.556	0.022	4.949	0.024	4.554	0.019
		$\boldsymbol{SE}(\hat{\boldsymbol{\beta}})$	0.03	0.001	0.275	0.001	0.333	0.001	0.435	0.001
		$\boldsymbol{SD}(\hat{\boldsymbol{\beta}})$	0.041	0.001	0.382	0.001	0.476	0.001	0.63	0.001
	MM	$\hat{\boldsymbol{\beta}}$	4.996	0.023	4.998	0.023	4.998	0.025	5.003	0.021
		$\boldsymbol{SE}(\hat{\boldsymbol{\beta}})$	0.023	0.001	0.022	0.001	0.022	0.001	0.023	0.001
		$\boldsymbol{SD}(\hat{\boldsymbol{\beta}})$	2.498	0.011	2.5	0.011	2.501	0.012	2.502	0.01
$\boldsymbol{\beta = 10}$	OLS	$\hat{\boldsymbol{\beta}}$	10.066	0.021	10.169	0.023	9.471	0.02	10.132	0.021
		$\boldsymbol{SE}(\hat{\boldsymbol{\beta}})$	0.027	0.001	0.133	0.001	0.535	0.001	0.532	0.001
		$\boldsymbol{SD}(\hat{\boldsymbol{\beta}})$	0.04	0.001	0.183	0.001	0.747	0.001	0.771	0.001
	MM	$\hat{\boldsymbol{\beta}}$	10.001	0.021	10	0.023	10.001	0.021	9.998	0.021
		$\boldsymbol{SE}(\hat{\boldsymbol{\beta}})$	0.023	0.001	0.023	0.001	0.023	0.001	0.023	0.001
		$\boldsymbol{SD}(\hat{\boldsymbol{\beta}})$	5	0.011	4.995	0.012	5.003	0.01	4.998	0.011

Table 1: OLS and Mixed Model estimates with 100 Monte-Carlo replicates