

Data simulation and calculation of betas

```
library(knitr)
library(kableExtra)
library(cowplot)
library(ggforce)
```

```
## Loading required package: ggplot2
```

```
library(latex2exp)
library(reshape2)
library(dplyr)
```

```
##
```

```
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:kableExtra':
```

```
##
```

```
##   group_rows
```

```
## The following objects are masked from 'package:stats':
```

```
##
```

```
##   filter, lag
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##   intersect, setdiff, setequal, union
```

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

Simulation data

```
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)
# X_gauss <- rnorm(N, 1, 2)
```

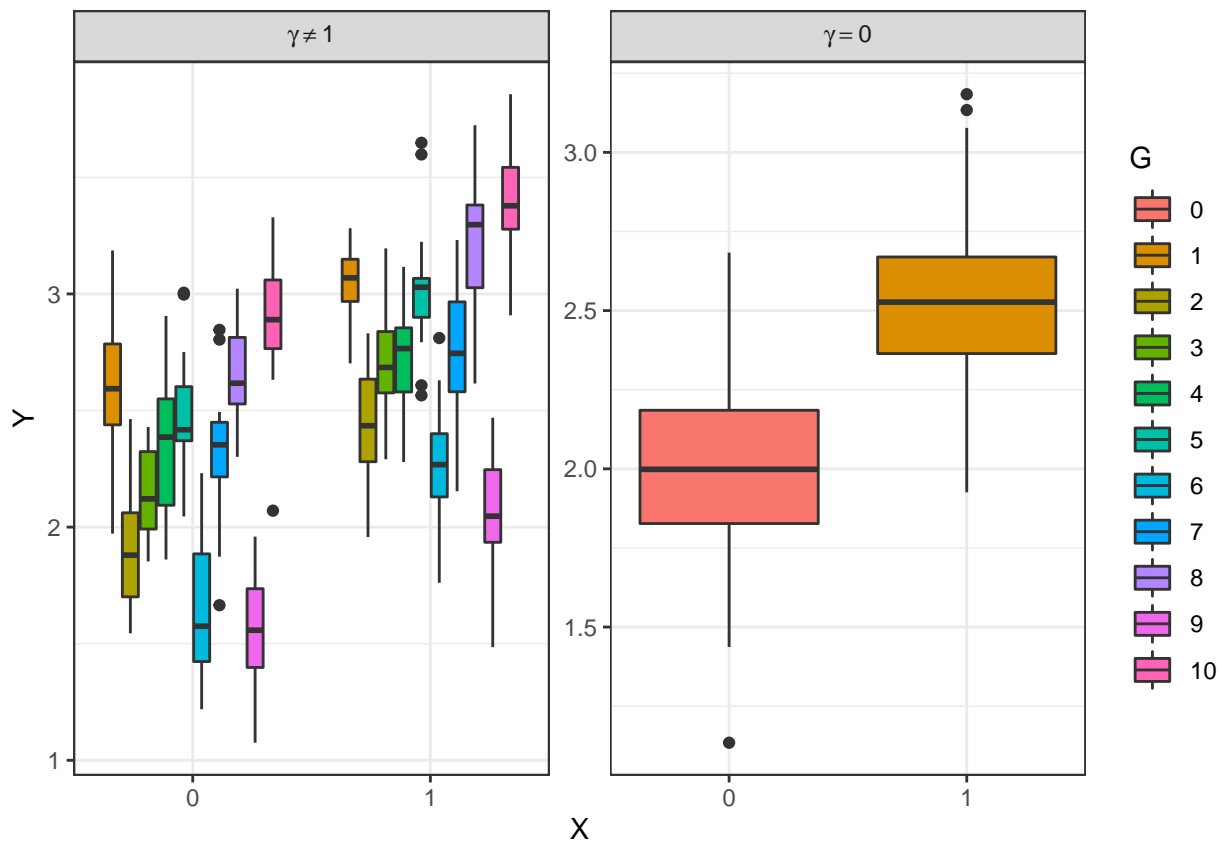
```

Y_withoutAlea <- intercept + fixeEffect * X + bias
# Y_G <- intercept + fixeEffect * X_gauss + aleaEffect[G] + bias
Y <- intercept + fixeEffect * X + aleaEffect[G] + bias
dfWith <- data.frame(X, Y, G)
dfWithout <- data.frame(X, Y=Y_withoutAlea, G)
H0 <- formula(Y ~ X)
H1 <- formula(Y ~ X + (1|G))

dataPlot = cbind.data.frame(X=rep(factor(X),2), Y = c(Y, Y_withoutAlea),
                             G = factor(c(G,X),levels = 0:10),
                             Effect = factor(rep(c("gamma != 1", "gamma == 0"),
                                                  each=length(X))))

ggplot(dataPlot) +
  geom_boxplot(aes(x=X, y=Y, fill = G)) +
  scale_x_discrete(expand = c(0, 0.5)) +
  theme_bw()+
  ggforce::facet_row(vars(Effect), scales = 'free', space = 'free',
                     labeller = "label_parsed")

```



Linear model with lm function

```

options(warn=-1)
X_prime <- cbind(1, X)

```

```

# With Random Effect
lmModel <- lm(H0,data = dfWith)
Sigm <- sqrt(((summary(lmModel)$sigma)**2)*solve(t(X) %*% X))
betaLm <- lmModel$coefficients
SE <- (betaLm[-1] - confint(lmModel)[-1,][1])/1.96

A <- ggplot(dfWith, aes(x = X, y = Y, color = G) ) +
  geom_point() +
  geom_smooth(formula = as.formula(y ~ x), method = "lm", se=.3,aes(fill = G))+
  theme_bw()+theme(legend.position="none")+
  annotate(geom='text', label=TeX("$\\gamma \\neq 0$"), y=3.7, x=.5)

# Without Random Effect
lmModel <- lm(H0,data = dfWithout)
Sigm.1 <- sqrt(((summary(lmModel)$sigma)**2)*solve(t(X) %*% X))
betaLm.1 <- lmModel$coefficients
SE.1 <- (betaLm.1[-1] - confint(lmModel)[-1,][1])/1.96

B <- ggplot(dfWithout, aes(x = X, y = Y, color = G) ) +
  geom_point() +
  geom_smooth(formula = as.formula(y ~ x), method = "lm",se = .3, aes(fill = G))+
  theme_bw()+theme(axis.title.y = element_blank())+
  annotate(geom='text', label=TeX("$\\gamma=0$"), y=3, x=.5)
cowplot::plot_grid(A, B, labels = c('',''))

```

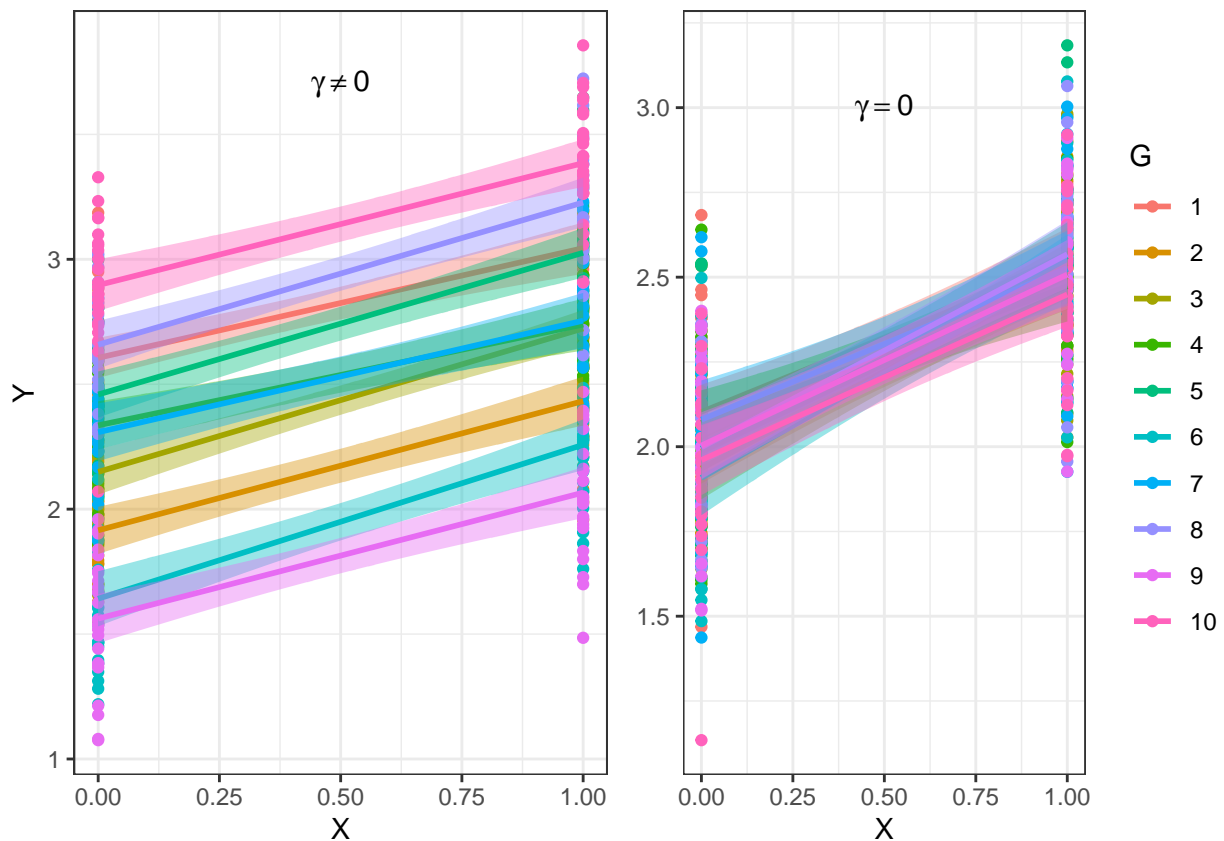


Table 1: Linear regression estimates with `lm()`

	$\hat{\beta}$	$SE(\hat{\beta})$	$\sigma(\hat{\beta})$
$\gamma \neq 0$	0.4964	0.0425	0.0301
$\gamma = 0$	0.5063	0.0219	0.0156

Mixed model with `lmer` function

```
options(warn=-1)
# With Random Effect
lmerModel <- lme4::lmer(H1, data=dfWith)
betaLmer <- lme4::fixef(lmerModel)
SE <- sqrt(diag(as.matrix(vcov(lmerModel))))[-1]
Sigm <- sqrt(stats::var(as.vector(betaLmer %*% t(X))))
A <- ggplot(dfWith, aes(x=X, y=Y, colour=G)) +
  geom_point(size=1.5) +
  geom_line(aes(y=predict(lmerModel), group=G), size=1.3) +
  theme_bw()+theme(legend.position="none")+
  annotate(geom='text', label=TeX("$\\gamma \\neq 0$"), y=3.7, x=.5)
# Without Random Effect
lmerModel.1 <- lme4::lmer(H1, data=dfWithout)
betaLmer.1 <- lme4::fixef(lmerModel.1)
SE.1 <- sqrt(diag(as.matrix(vcov(lmerModel.1))))[-1]
Sigm.1 <- sqrt(stats::var(as.vector(betaLmer.1 %*% t(X))))

B <- ggplot(dfWithout, aes(x=X, y=Y, colour=G)) +
  geom_point(size=1.5) +
  geom_line(aes(y=predict(lmerModel.1), group=G), size=1.3) +
  theme_bw()+theme(axis.title.y = element_blank())+
  annotate(geom='text', label=TeX("$\\gamma=0$"), y=3, x=.5)
cowplot::plot_grid(A, B, labels = c('', ''))
```

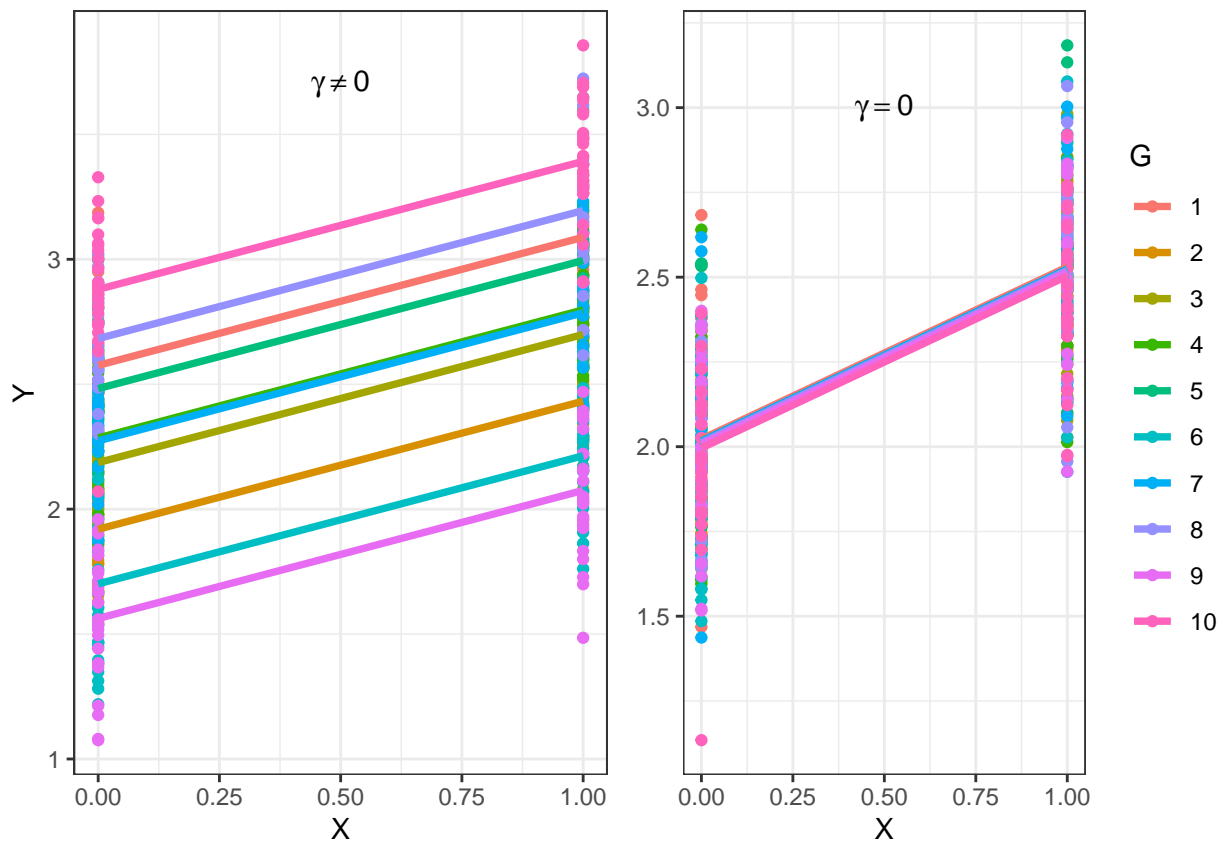


Table 2: Mixed Model estimates with `lmer()`

	$\hat{\beta}$	$SE(\hat{\beta})$	$\sigma(\hat{\beta})$
$\gamma \neq 0$	0.5126	0.022	0.9243
$\gamma = 0$	0.5077	0.0219	0.8211

OLS

```

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]
  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)
Sigm <- sqrt(Vb)
OLSCOV <- 1/(n-k) * as.numeric(t(residuals)%*%residuals) * solve(t(modX_OLS)%*%modX_OLS)
SE <- sqrt(diag(OLSCOV))[-1]
return(list('betaOLS'=betaOLS[indexes_X, ,drop=FALSE], 'SE'=SE, 'Sigm'=Sigm))
}
options(warn=-1)
# With Random Effect
res <- OLS(dfWith$Y,X)
# Without Random Effect
res.1 <- OLS(dfWithout$Y,X)

```

Table 3: OLS estimates

	$\hat{\beta}$	$SE(\hat{\beta})$	$\sigma(\hat{\beta})$
$\gamma \neq 0$	0.4964	0.0424	0.9243
$\gamma = 0$	0.5063	0.0219	0.8211

Monte Carlo

Simulating data

The following represents a simulate a random intercepts model obtained 500 Monte-Carlo replicates. The method follows the following simulation framework from gaussian distributions with or without the presence of random effects:

$$Y = \beta_0 + \beta X + \gamma G + \epsilon \quad (1)$$

with $\beta_0 = 2$, $\beta \in \{0.5, 2, 5, 10\}$ the fixed effect of $X \sim \mathcal{B}_{(0.5)}^n$ with $n = 500$ the number total of samples, $\epsilon \sim \mathcal{N}(0, 0.25)$ the bias associated, the random effect of group $\gamma \sim \mathcal{N}(0, \sigma_\gamma)$ if simulated with random effect with $\sigma_\gamma \in \{0.5, 5, 10, 20\}$ and $\gamma = 0$ if not random effect and the group $G \in \{1, \dots, K\}$ with $K = 10$.

Running the Models

```

simOLS <- function (intercept=2, fixefEffects = c(.5,2,5,10), sds = c(.5,5,10,20),
                    K=10, nK=50, sims=5000){
  resAllOLS.1 <- resAllOLS.2 <- resAllMM.1 <- list()
  estimateAll <- matrix(0, length(fixefEffects)*length(sds)*sims, 12)
  yWithSimMean <- yWithoutSimMean <- matrix(0, 500, length(fixefEffects)*length(sds))
  n <- K * nK
  G <- factor(rep(1:K, each = nK))
  X <- rbinom(n,size=1,prob = .5)
  k = 1
  kk = 1
  for (fixefEffect in fixefEffects) {
    for (sdUnit in sds) {
      aleaEffect <- rnorm(K, sd = sdUnit)
    }
  }
}

```

```

resOLS.1 <- resOLS.2 <- resMM <- matrix(0, sims, 3)
yOlsWith <- yOlsWithout <- matrix(0, 500, sims)
for(i in 1:sims){
  options(warn=-1)
  bias <- rnorm(n, sd = .25)
  Y_with <- intercept + fixefEffect * X + aleaEffect[G] + bias
  Y_without <- intercept + fixefEffect * X + bias
  modOLS.1 <- OLS(Y_with,X)
  modOLS.2 <- OLS(Y_without,X)
  resOLS.1[i,] <- c(modOLS.1$betaOLS, modOLS.1$SE, modOLS.1$Sigm)
  resOLS.2[i,] <- c(modOLS.2$betaOLS, modOLS.2$SE, modOLS.2$Sigm)
  lmerModel.1 <- try({ lme4::lmer(Y_with ~ X + (1|G), REML = TRUE)}, silent = T)
  beta <- lme4::fixef(lmerModel.1)[-1]
  SE_b <- sqrt(diag(as.matrix(vcov(lmerModel.1))))[-1]
  Sigm <- sqrt(stats::var(as.vector(beta %*% t(X))))
  resMM[i,] <- c(beta, SE_b,Sigm)
  estimateAll[k,] <- c(fixefEffect, modOLS.1$betaOLS,modOLS.2$betaOLS,beta,
    modOLS.1$SE,modOLS.2$SE,SE_b,
    modOLS.1$Sigm,modOLS.2$Sigm,Sigm,sdUnit,i)
  yOlsWith[,i] <- Y_with; yOlsWithout[,i] <- Y_without
  k <- k+1
}
yWithSimMean[,kk] <- rowMeans(yOlsWith)
yWithoutSimMean[,kk] <- rowMeans(yOlsWithout)
kk <- kk + 1

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

M <- apply(resOLS.2, 2, mean)
S <- apply(resOLS.2, 2, sd)
H_ols.2 <- 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.1) <- dimnames(H_mm) <-
  dimnames(H_ols.2) <- list( c('Beta','Standard Error', 'Sigm'), c('mean', 'se'))

resAllOLS.1[[as.character(paste0(fixefEffect,"_",sdUnit))]] <- H_ols.1
resAllOLS.2[[as.character(paste0(fixefEffect,"_",sdUnit))]] <- H_ols.2
resAllMM.1[[as.character(paste0(fixefEffect,"_",sdUnit))]] <- H_mm
}
}
estimateAll <- as.data.frame(estimateAll)
yWithSimMean <- as.data.frame(yWithSimMean)
yWithoutSimMean <- as.data.frame(yWithoutSimMean)
yWithSimMean <- cbind.data.frame(X,G,yWithSimMean)
yWithoutSimMean <- cbind.data.frame(X,G,yWithoutSimMean)
colnames(estimateAll) = c('fixefEffect','betaOLS.1','betaOLS.2','betaMM',
  'seOLS.1','seOLS.2','seMM','sigmOLS.1',

```

```

      'sigmOLS.2','sigmMM','sdUnit','simId')
return (list('resAllOLS'=resAllOLS.1, 'resAllMM'=resAllMM.1,
      'resAllOLSWithoutAleaEffect'=resAllOLS.2,
      "estimateAll"=estimateAll,"yWithSimMean"=yWithSimMean,
      'yWithoutSimMean'=yWithoutSimMean))
}

resSim <- simOLS(sims = 500)

dataPLOT <- cbind.data.frame(betaEs = c(resSim$estimateAll[[2]],resSim$estimateAll[[3]],
      resSim$estimateAll[[4]]),
      betas = factor(rep(resSim$estimateAll[[1]],3)),
      Sigm = factor(rep(resSim$estimateAll[[11]],3)),
      Method = as.factor(rep(c('OLS.1','OLS.2','MM'),
      each=length(resSim$estimateAll[[1]]))))

```

Results

Plot estimation of β

```

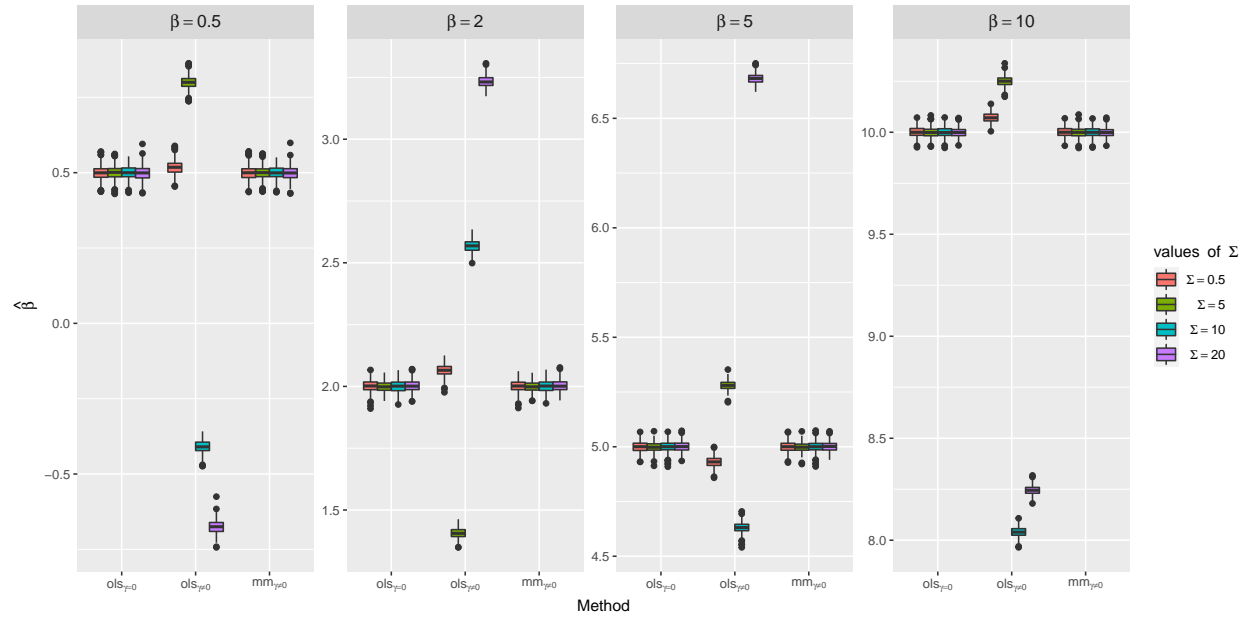
dataPLOT$Bias = rep(resSim$estimateAll[[1]],3)-dataPLOT$betaEs

dataPLOT$betas <- gsub(0.5, "beta == 0.5",
      gsub(2, "beta == 2",
      gsub(10, "beta == 10",dataPLOT$betas)))
dataPLOT[dataPLOT$betas==5,'betas']="beta == 5"
dataPLOT$betas = factor(dataPLOT$betas,levels = c("beta == 0.5","beta == 2",
      "beta == 5","beta == 10"))

dataPLOT$Sigm <- gsub(0.5, "gamma == 0.5",
      gsub(10, "gamma == 10",
      gsub(20, "gamma == 20",dataPLOT$Sigm)))
dataPLOT[dataPLOT$Sigm==5,'Sigm']="gamma == 5"
dataPLOT$Sigm = factor(dataPLOT$Sigm,levels = c("gamma == 0.5","gamma == 5",
      "gamma == 10","gamma == 20"))

ggplot(dataPLOT, aes(Method,betaEs))+
geom_boxplot(position="dodge",aes(fill=Sigm))+
facet_wrap(~betas, scales = "free_y",labeller = label_parsed,ncol = 4)+
ylab(TeX("$\\hat{\\beta}$"))+
scale_x_discrete(labels=c(TeX("$ols_{\\gamma=0}$"),TeX("$ols_{\\gamma\\neq 0}$"),
      TeX("$mm_{\\gamma\\neq 0}$")))+
theme(strip.text.x = element_text(size=12, face="bold"),
      strip.text.y = element_text(size=12, face="bold",)) +
scale_fill_discrete(name=TeX("$values\\ of\\ \\Sigma$"),
      labels=c(TeX("$\\Sigma=.5$"),TeX("$\\Sigma=5$"),
      TeX("$\\Sigma=10$"),TeX("$\\Sigma=20$")))

```

Plot bias of estimates β

```
ggplot(data=dataPLOT, aes(x=Method, y=Bias))+
  geom_boxplot()+
  theme_bw()+
  facet_grid(betas~Sigm, scales = "free_y", labeller = label_parsed)+
  theme(strip.text.x = element_text(size=12, face="bold"),
        strip.text.y = element_text(size=12, face="bold",)) +
  ylab(TeX("$\\beta-\\hat{\\beta}$"))+
  scale_x_discrete(labels=c(TeX("$ols_{\\gamma=0}$"), TeX("$ols_{\\gamma \\neq 0}$"),
                           TeX("$mm_{\\gamma \\neq 0}$")))
```

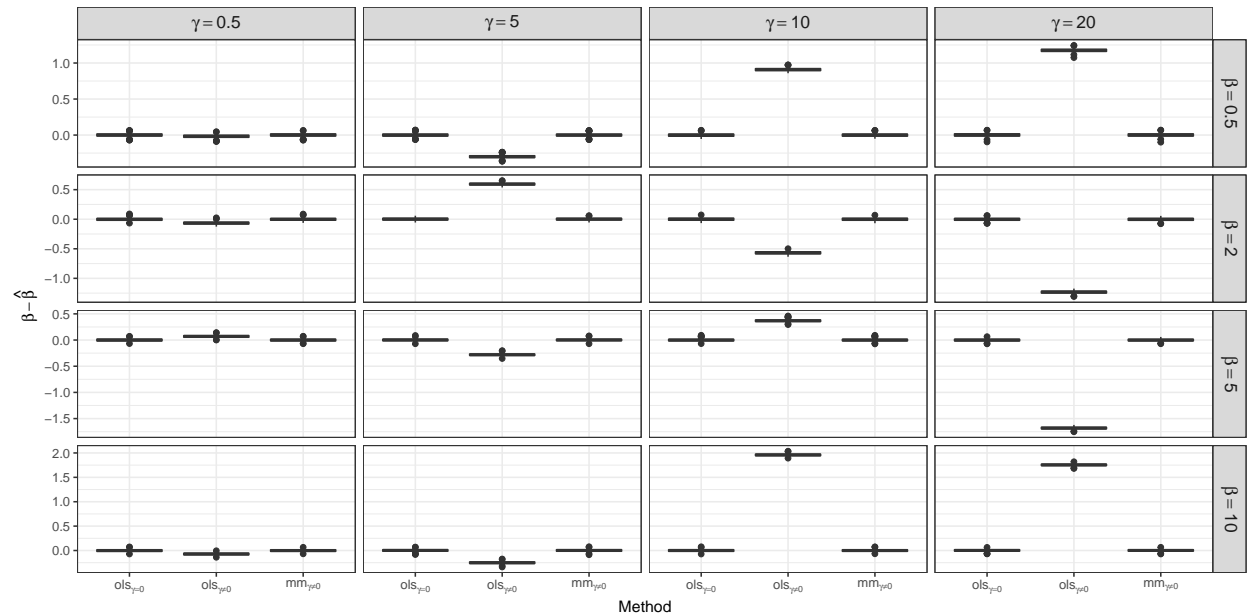


Table 4: OLS and Mixed Model estimates with 500 Monte-Carlo replicates

β	Model	γ	Estimate	$\Sigma = 0.5$		$\Sigma = 5$		$\Sigma = 10$		$\Sigma = 20$	
				μ	SE	μ	SE	μ	SE	μ	SE
$\beta = 0.5$	OLS	$\gamma \neq 0$	$\hat{\beta}$	0.5173	0.0228	0.7996	0.0213	-0.409	0.0205	-0.6748	0.0229
			$SE(\hat{\beta})$	0.0584	0.001	0.28	0.001	0.6239	0.001	2.0317	0.001
			$\sigma(\hat{\beta})$	0.043	7×10^{-4}	0.2061	8×10^{-4}	0.4593	8×10^{-4}	1.4957	7×10^{-4}
		$\gamma = 0$	$\hat{\beta}$	0.4992	0.0228	0.4998	0.0213	0.5002	0.0205	0.4993	0.0229
			$SE(\hat{\beta})$	0.0225	7×10^{-4}	0.0224	7×10^{-4}	0.0224	7×10^{-4}	0.0224	7×10^{-4}
			$\sigma(\hat{\beta})$	0.0165	5×10^{-4}	0.0165	5×10^{-4}	0.0165	5×10^{-4}	0.0165	5×10^{-4}
	MM	$\gamma \neq 0$	$\hat{\beta}$	0.4992	0.0233	0.5	0.0217	0.5001	0.0207	0.4993	0.0231
			$SE(\hat{\beta})$	0.0228	7×10^{-4}	0.0227	7×10^{-4}	0.0227	7×10^{-4}	0.0227	7×10^{-4}
			$\sigma(\hat{\beta})$	0.249	0.0116	0.2494	0.0108	0.2494	0.0103	0.249	0.0115
$\beta = 2$	OLS	$\gamma \neq 0$	$\hat{\beta}$	2.065	0.0228	1.4059	0.0214	2.5674	0.024	3.2332	0.022
			$SE(\hat{\beta})$	0.042	0.001	0.3179	0.001	0.6252	0.001	1.5803	0.001
			$\sigma(\hat{\beta})$	0.0309	7×10^{-4}	0.234	7×10^{-4}	0.4602	7×10^{-4}	1.1634	7×10^{-4}
		$\gamma = 0$	$\hat{\beta}$	2.0014	0.0228	1.9986	0.0214	2.0002	0.024	2.0027	0.022
			$SE(\hat{\beta})$	0.0224	7×10^{-4}	0.0225	7×10^{-4}	0.0224	7×10^{-4}	0.0224	7×10^{-4}
			$\sigma(\hat{\beta})$	0.0165	5×10^{-4}	0.0166	5×10^{-4}	0.0165	5×10^{-4}	0.0165	5×10^{-4}
	MM	$\gamma \neq 0$	$\hat{\beta}$	2.0018	0.0234	1.9986	0.0215	2.0003	0.0243	2.0023	0.0222
			$SE(\hat{\beta})$	0.0227	8×10^{-4}	0.0228	7×10^{-4}	0.0227	7×10^{-4}	0.0227	7×10^{-4}
			$\sigma(\hat{\beta})$	0.9984	0.0117	0.9968	0.0107	0.9976	0.0121	0.9986	0.0111
$\beta = 5$	OLS	$\gamma \neq 0$	$\hat{\beta}$	4.9305	0.0233	5.2808	0.0213	4.631	0.0229	6.6814	0.0215
			$SE(\hat{\beta})$	0.0717	0.001	0.3144	0.001	1.1203	0.001	1.7431	0.001
			$\sigma(\hat{\beta})$	0.0528	7×10^{-4}	0.2315	7×10^{-4}	0.8248	8×10^{-4}	1.2833	7×10^{-4}
		$\gamma = 0$	$\hat{\beta}$	4.9999	0.0233	4.9986	0.0213	5.0003	0.0229	5.0007	0.0215
			$SE(\hat{\beta})$	0.0224	7×10^{-4}	0.0225	7×10^{-4}	0.0224	7×10^{-4}	0.0225	7×10^{-4}
			$\sigma(\hat{\beta})$	0.0165	5×10^{-4}	0.0166	5×10^{-4}	0.0165	5×10^{-4}	0.0165	5×10^{-4}
	MM	$\gamma \neq 0$	$\hat{\beta}$	4.9996	0.0234	4.9987	0.0216	5.0001	0.0233	5.0006	0.0215
			$SE(\hat{\beta})$	0.0227	7×10^{-4}	0.0228	7×10^{-4}	0.0227	7×10^{-4}	0.0228	7×10^{-4}
			$\sigma(\hat{\beta})$	2.4935	0.0117	2.493	0.0108	2.4937	0.0116	2.4939	0.0107
$\beta = 10$	OLS	$\gamma \neq 0$	$\hat{\beta}$	10.0716	0.0232	10.2497	0.0243	8.0396	0.0224	8.2451	0.0218
			$SE(\hat{\beta})$	0.0454	9×10^{-4}	0.4424	0.001	0.8905	0.001	1.8213	9×10^{-4}
			$\sigma(\hat{\beta})$	0.0334	7×10^{-4}	0.3257	7×10^{-4}	0.6556	7×10^{-4}	1.3409	7×10^{-4}
		$\gamma = 0$	$\hat{\beta}$	10.0008	0.0232	9.9985	0.0243	9.9996	0.0224	9.9989	0.0218
			$SE(\hat{\beta})$	0.0224	7×10^{-4}	0.0224	7×10^{-4}	0.0225	7×10^{-4}	0.0224	7×10^{-4}
			$\sigma(\hat{\beta})$	0.0165	5×10^{-4}	0.0165	5×10^{-4}	0.0165	5×10^{-4}	0.0165	5×10^{-4}
	MM	$\gamma \neq 0$	$\hat{\beta}$	10.0013	0.0235	9.9987	0.0248	9.9996	0.0226	9.999	0.0222
			$SE(\hat{\beta})$	0.0227	7×10^{-4}	0.0227	7×10^{-4}	0.0228	7×10^{-4}	0.0228	7×10^{-4}
			$\sigma(\hat{\beta})$	4.9879	0.0117	4.9867	0.0124	4.9871	0.0113	4.9868	0.0111

Running models on the average of the simulations

```
options(warn=-1)
colnames(resSim$yWithSimMean) <- colnames(resSim$yWithoutSimMean) <- c('X', 'G',
                                                                           names(resSim$resAllOLS))

resMeanSim <- matrix(0, nrow = length(colnames(resSim$yWithSimMean[, -c(1:2)])), 9)
X <- resSim$yWithSimMean$X
G <- resSim$yWithSimMean$G
k=1
for (i in colnames(resSim$yWithSimMean[, -c(1:2)])) {
  lmerModel <- try({ lme4::lmer(H1,
                              data = cbind.data.frame(Y=resSim$yWithSimMean[, i],
                                                         X=X,
                                                         G=G),
                              REML = TRUE)}, silent = T)

  beta <- lme4::fixef(lmerModel)[-1]
  SE <- sqrt(diag(as.matrix(vcov(lmerModel))))[-1]
  Sigm <- sqrt(stats::var(as.vector(beta %*% t(X))))
  resMeanSim[k,] <- c(unlist(OLS(resSim$yWithSimMean[, i], X)),
                     unlist(OLS(resSim$yWithoutSimMean[, i], X)),
                     beta, SE, Sigm)

  k <- k+1
}
resMeanSim <- as.data.frame(resMeanSim)
```

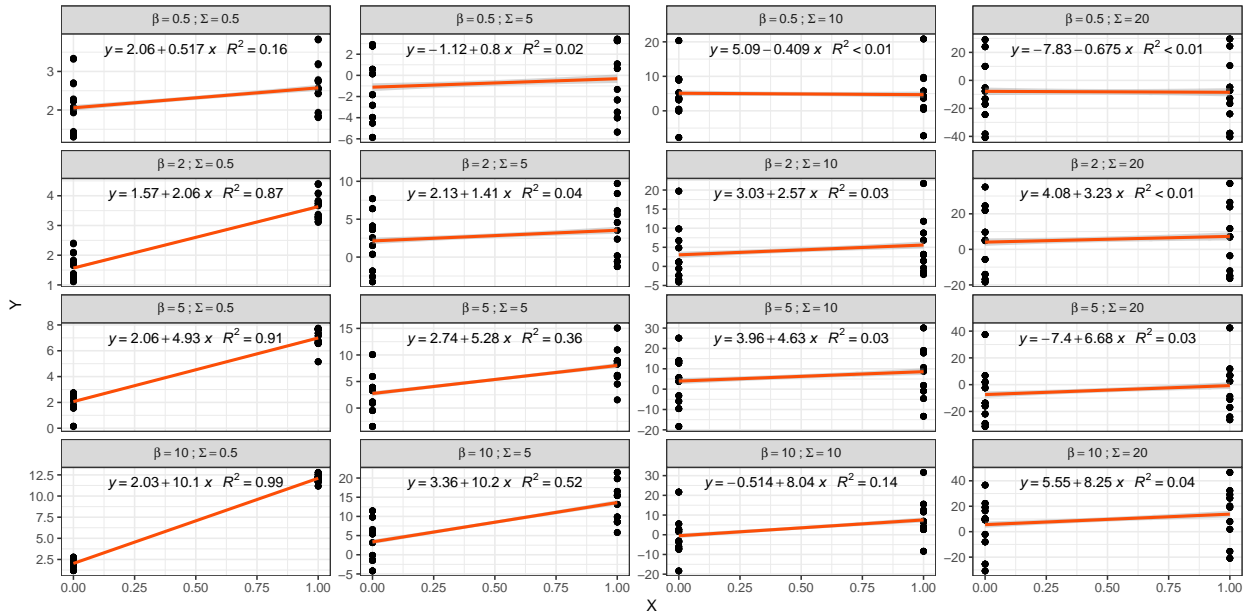
Table 5: OLS and Mixed Model estimates with 500 Monte-Carlo replicates

β	Σ	$OLS_{\gamma \neq 0}$			$OLS_{\gamma=0}$			$MM_{\gamma \neq 0}$		
		$\hat{\beta}$	$SE(\hat{\beta})$	$\sigma(\hat{\beta})$	$\hat{\beta}$	$SE(\hat{\beta})$	$\sigma(\hat{\beta})$	$\hat{\beta}$	$SE(\hat{\beta})$	$\sigma(\hat{\beta})$
$\beta = 0.5$	$\Sigma = 0.5$	0.5173	0.0539	0.0397	0.4992	0.001	7×10^{-4}	0.4992	0.001	0.249
	$\Sigma = 5$	0.7996	0.2791	0.2055	0.4998	0.001	7×10^{-4}	0.5	0.001	0.2494
	$\Sigma = 10$	-0.409	0.6235	0.459	0.5002	0.001	7×10^{-4}	0.5002	0.001	0.2494
	$\Sigma = 20$	-0.6748	2.0316	1.4957	0.4993	0.001	8×10^{-4}	0.4993	0.001	0.249
$\beta = 2$	$\Sigma = 0.5$	2.065	0.0355	0.0261	2.0014	0.001	7×10^{-4}	2.0014	0.001	0.9981
	$\Sigma = 5$	1.4059	0.3171	0.2335	1.9986	0.0011	8×10^{-4}	1.9987	0.0011	0.9968
	$\Sigma = 10$	2.5674	0.6248	0.4599	2.0002	0.001	7×10^{-4}	2.0002	0.001	0.9976
	$\Sigma = 20$	3.2332	1.5801	1.1633	2.0027	0.001	7×10^{-4}	2.0023	0.001	0.9986
$\beta = 5$	$\Sigma = 0.5$	4.9305	0.0682	0.0502	4.9999	0.001	8×10^{-4}	4.9998	0.0011	2.4935
	$\Sigma = 5$	5.2808	0.3136	0.2309	4.9986	0.0011	8×10^{-4}	4.9987	0.0011	2.493
	$\Sigma = 10$	4.631	1.1201	0.8246	5.0003	0.001	7×10^{-4}	5.0001	0.001	2.4937
	$\Sigma = 20$	6.6814	1.7429	1.2832	5.0007	0.001	8×10^{-4}	5.0006	0.001	2.4939
$\beta = 10$	$\Sigma = 0.5$	10.0716	0.0395	0.0291	10.0008	0.001	8×10^{-4}	10.0008	0.001	4.9877
	$\Sigma = 5$	10.2497	0.4419	0.3253	9.9985	0.001	8×10^{-4}	9.9987	0.001	4.9867
	$\Sigma = 10$	8.0396	0.8902	0.6553	9.9996	0.001	8×10^{-4}	9.9997	0.001	4.9871
	$\Sigma = 20$	8.2451	1.8212	1.3408	9.9989	0.001	7×10^{-4}	9.999	0.001	4.9868

Visualization of the models

```
dataMod <- melt(resSim$yWithSimMean, id.vars=c("X","G"),value.name = "Y")
dataMod$variable = recode_factor(dataMod$variable,
  "0.5_0.5"="beta==0.5 ~ ';' ~ Sigma == 0.5",
  "0.5_5"="beta == 0.5 ~ ';' ~ Sigma == 5","0.5_10"="beta == 0.5 ~ ';' ~ Sigma == 10",
  "0.5_20"="beta == 0.5 ~ ';' ~ Sigma == 20",
  "2_0.5"="beta == 2 ~ ';' ~ Sigma == 0.5", "2_5"="beta == 2 ~ ';' ~ Sigma == 5",
  "2_10"="beta == 2 ~ ';' ~ Sigma == 10",
  "2_20"="beta == 2 ~ ';' ~ Sigma == 20","5_0.5"="beta == 5 ~ ';' ~ Sigma == 0.5",
  "5_5"="beta == 5 ~ ';' ~ Sigma == 5","5_10"="beta == 5 ~ ';' ~ Sigma == 10",
  "5_20"="beta == 5 ~ ';' ~ Sigma == 20",
  "10_0.5"="beta == 10 ~ ';' ~ Sigma == 0.5",
  "10_5"="beta == 10 ~ ';' ~ Sigma == 5","10_10"="beta == 10 ~ ';' ~ Sigma == 10",
  "10_20"="beta == 10 ~ ';' ~ Sigma == 20")

ggplot(data = dataMod, aes(x = X, y = Y)) +
  geom_point(aes(X, Y), alpha = 0.3)+
  geom_smooth(formula = as.formula(y~x),aes(x = X, y = Y),
    method = "lm", colour="#FC4E07", fullrange = TRUE, se = TRUE)+
  ggpmisc::stat_poly_eq(formula = as.formula(y~x),
    aes(label=paste(..eq.label.., ..rr.label.., sep = "~~~")),
    parse = TRUE, label.x.npc = "center", size = 3.45)+
  theme(strip.text.x = element_text(size=12, face="bold"),
strip.text.y = element_text(size=12, face="bold",))+theme_bw()+
  facet_wrap(~ variable, ncol=4, scales = "free_y",labeller = label_parsed)
```



```
dataMod <- melt(resSim$yWithoutSimMean, id.vars=c("X","G"),value.name = "Y")
dataMod$variable = recode_factor(dataMod$variable,

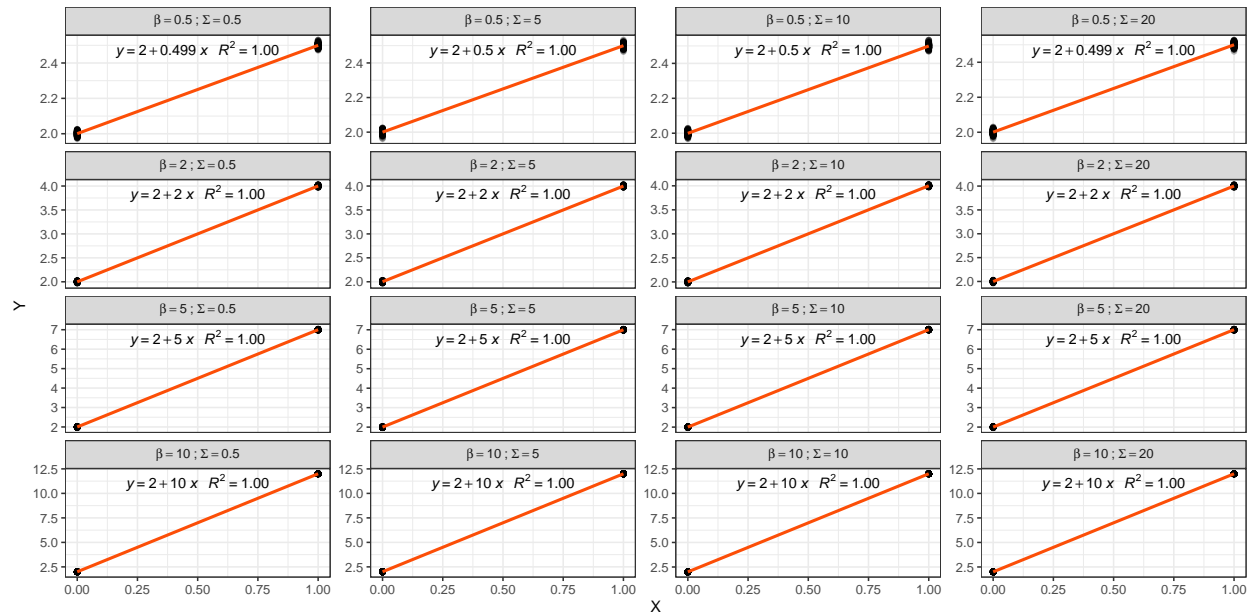
  "0.5_0.5"="beta == 0.5 ~ ';' ~ Sigma == 0.5",
  "0.5_5"="beta == 0.5 ~ ';' ~ Sigma == 5","0.5_10"="beta == 0.5 ~ ';' ~ Sigma == 10",
  "0.5_20"="beta == 0.5 ~ ';' ~ Sigma == 20",
  "2_0.5"="beta == 2 ~ ';' ~ Sigma == 0.5", "2_5"="beta == 2 ~ ';' ~ Sigma == 5",
  "2_10"="beta == 2 ~ ';' ~ Sigma == 10",
  "2_20"="beta == 2 ~ ';' ~ Sigma == 20","5_0.5"="beta == 5 ~ ';' ~ Sigma == 0.5",
  "5_5"="beta == 5 ~ ';' ~ Sigma == 5","5_10"="beta == 5 ~ ';' ~ Sigma == 10",
  "5_20"="beta == 5 ~ ';' ~ Sigma == 20",
  "10_0.5"="beta == 10 ~ ';' ~ Sigma == 0.5",
  "10_5"="beta == 10 ~ ';' ~ Sigma == 5","10_10"="beta == 10 ~ ';' ~ Sigma == 10",
  "10_20"="beta == 10 ~ ';' ~ Sigma == 20")

ggplot(data = dataMod, aes(x = X, y = Y)) +
  geom_point(aes(X, Y), alpha = 0.3)+
  geom_smooth(formula = as.formula(y~x),aes(x = X, y = Y),
    method = "lm", colour="#FC4E07", fullrange = TRUE, se = TRUE)+
  ggpmisc::stat_poly_eq(formula = as.formula(y~x),
    aes(label=paste(..eq.label.., ..rr.label.., sep = "~~~")),
```

```

    parse = TRUE, label.x.npc = "center", size = 3.45)+
  theme(strip.text.x = element_text(size=12, face="bold"),
strip.text.y = element_text(size=12, face="bold",))+theme_bw()+
  facet_wrap(~ variable, ncol=4, scales = "free_y",labeller = label_parsed)

```



```

colnames(resSim$yWithSimMean)[10]="Y"

lmerModel <- lme4::lmer(H1, data=resSim$yWithSimMean)

ab_lines <- coef(lmerModel)[["G"]] %>%
  tibble::rownames_to_column("G") %>%
  rename(intercept = `(Intercept)`)

ab_lines$G <- factor(ab_lines$G, levels=1:K)

ggplot(resSim$yWithSimMean,aes(x = X, y = Y, colour=G)) +
  geom_point() +
  geom_line(aes(y=predict(lmerModel), group=G)) +
  geom_line(colour="red",aes(y=predict(lmerModel,re.form=NA),group=G))+
  theme_bw()+theme(legend.position="none")

```

