Data simulation and calculation of betas

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

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

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

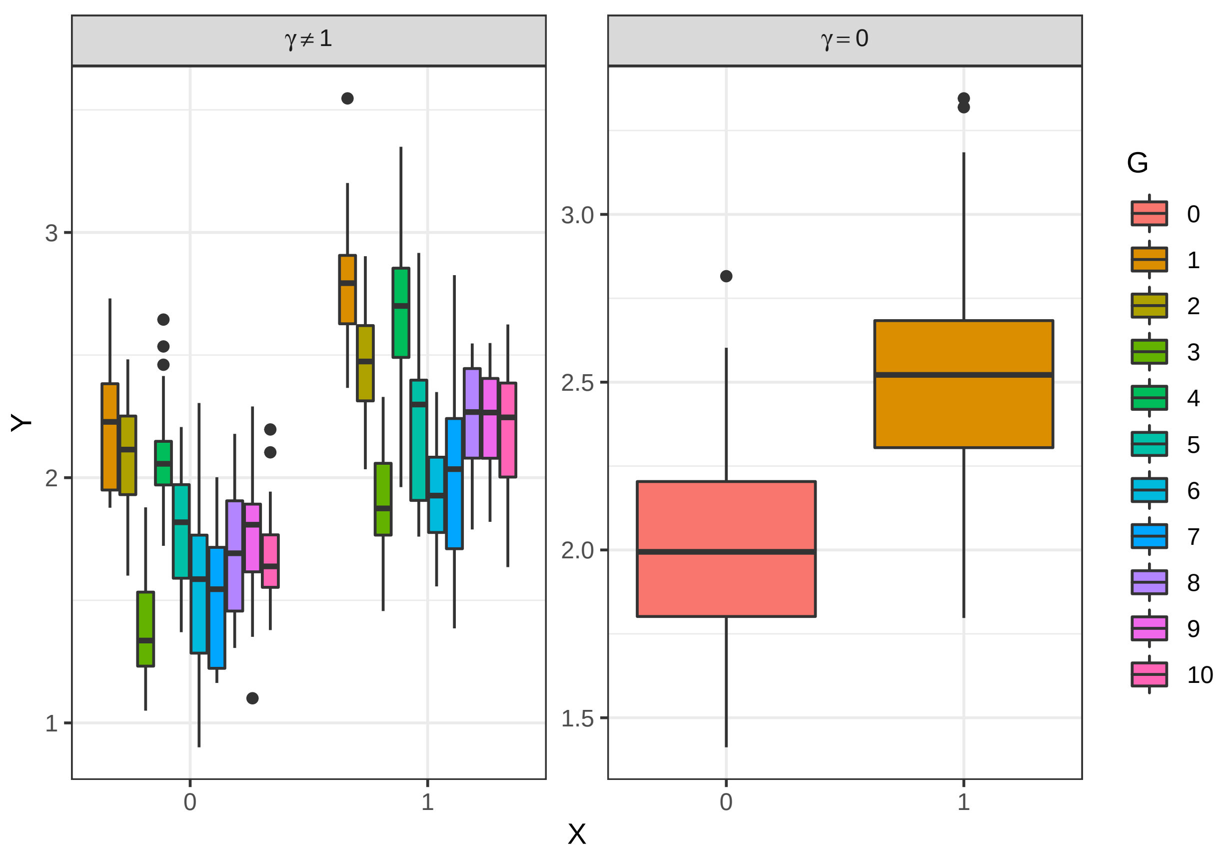
```
##
## Attaching package: 'dplyr'

## 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

```
seeds24_03_22 <- read.csv("seeds24.03.22.csv")$x
set.seed(seeds24_03_22[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)
```

```r
bias <- rnorm(N, sd = .25)
X <- rbinom(N,size=1,prob = .5)
# X_gauss <- rnorm(N,1,2)
Y_withoutAlea <- intercept + fixefEffect * X + bias
# Y_G <- intercept + fixefEffect * X_gauss + aleaEffect[G]  + bias
Y <- intercept + fixefEffect * 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",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) ) +
     geom_point() +
     geom_smooth(formula = as.formula(y ~ x), method = "lm")+
  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('',''))
```

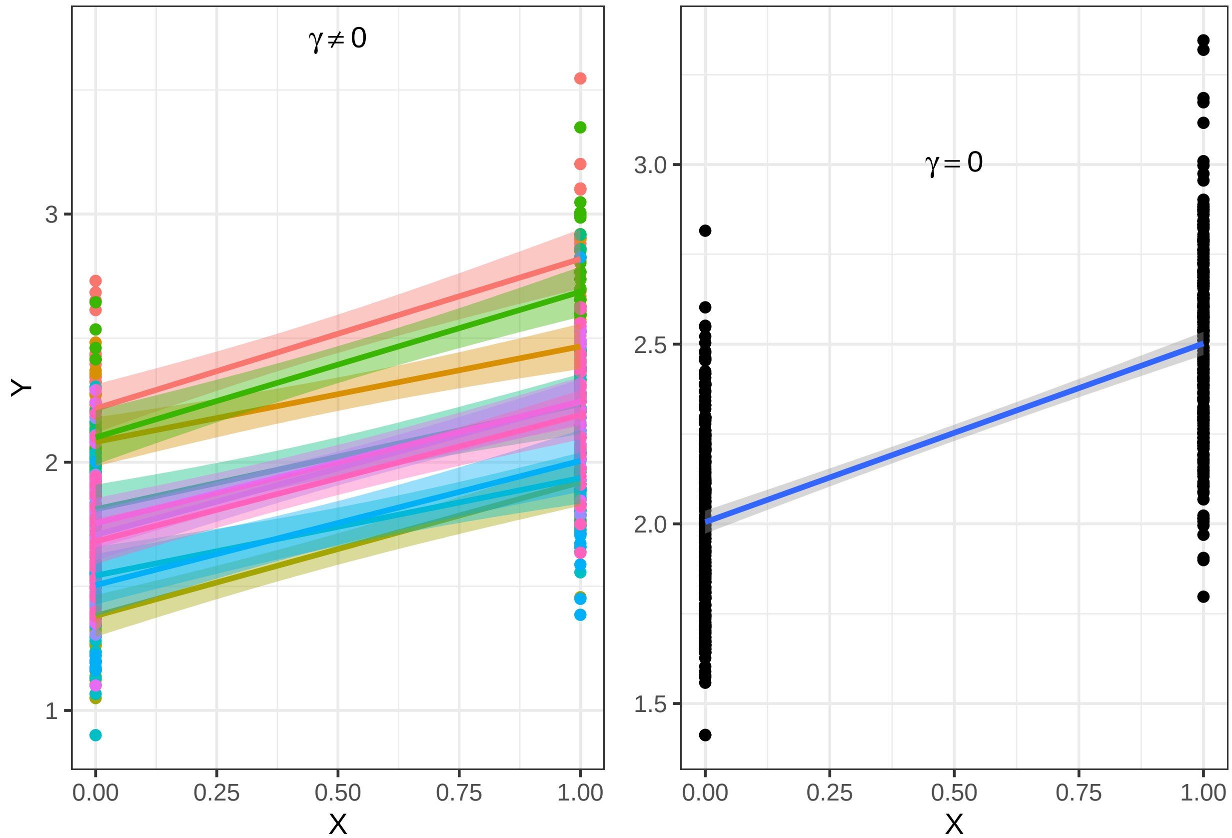


Table 1: Linear regression estimates with `lm()`

	$\hat{\beta}$	$SE(\hat{\beta})$	$\sigma(\hat{\beta})$
$\gamma \neq 0$	0.4895	0.0335	0.0238
$\gamma = 0$	0.4973	0.0231	0.0164

Mixed model with lmer function

```
# 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('',''))
```

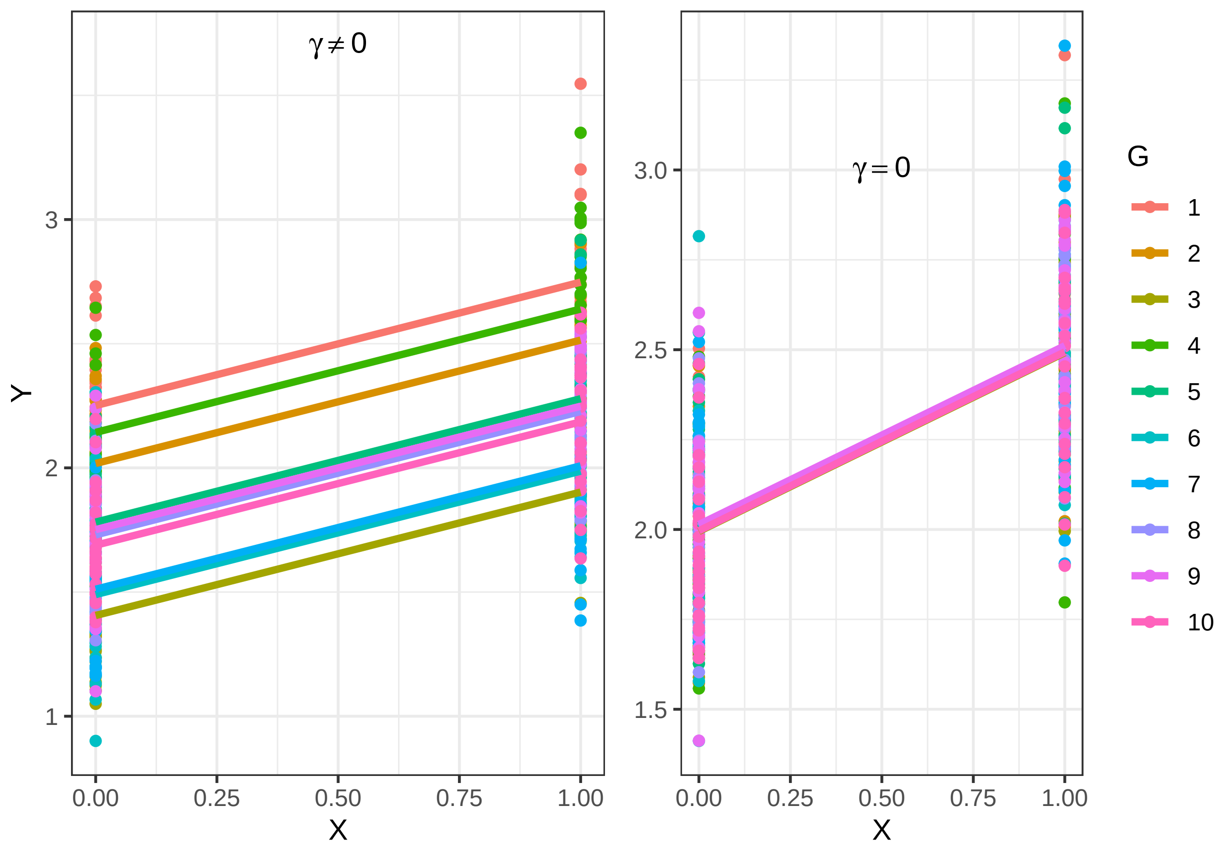


Table 2: Mixed Model estimates with `lmer()`

	$\hat{\beta}$	$SE(\hat{\beta})$	$\sigma(\hat{\beta})$
$\gamma \neq 0$	0.4968	0.0232	0.7247
$\gamma = 0$	0.4972	0.0231	0.8194

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

	$\hat{\beta}$	$SE(\hat{\beta})$	$\sigma(\hat{\beta})$
$\gamma \neq 0$	0.4895	0.0334	0.7247
$\gamma = 0$	0.4973	0.0231	0.8194

Monte Carlo

Simulating data

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

$$Y = \beta_0 + \beta X + \gamma G + \epsilon \tag{1}$$

with $\beta_0 = 2$, $\beta \in \{0.5, 2, 5, 10\}$ the fixed effect of $X \sim \mathcal{B}\binom{n}{0.5}$ 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 $G \in \{1, \ldots, K\}$ with $K = 10$.

Running the Models

```r
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) {
        set.seed(seeds24_03_22[kk])
        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)
          set.seed(seeds24_03_22[i])
          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)
```

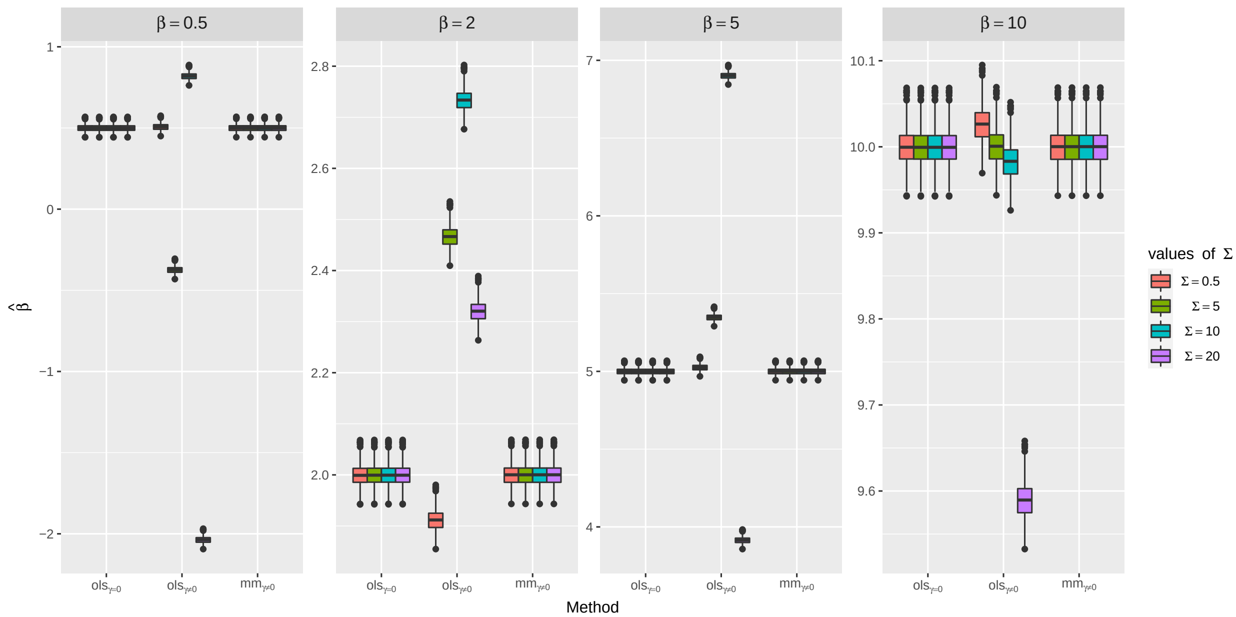
```
        dimnames(H_ols.1) <- dimnames(H_mm) <-
          dimnames(H_ols.2)  <- list( c('Beta','Standard Error', 'Sigm'), c('mean', 'se'))

        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

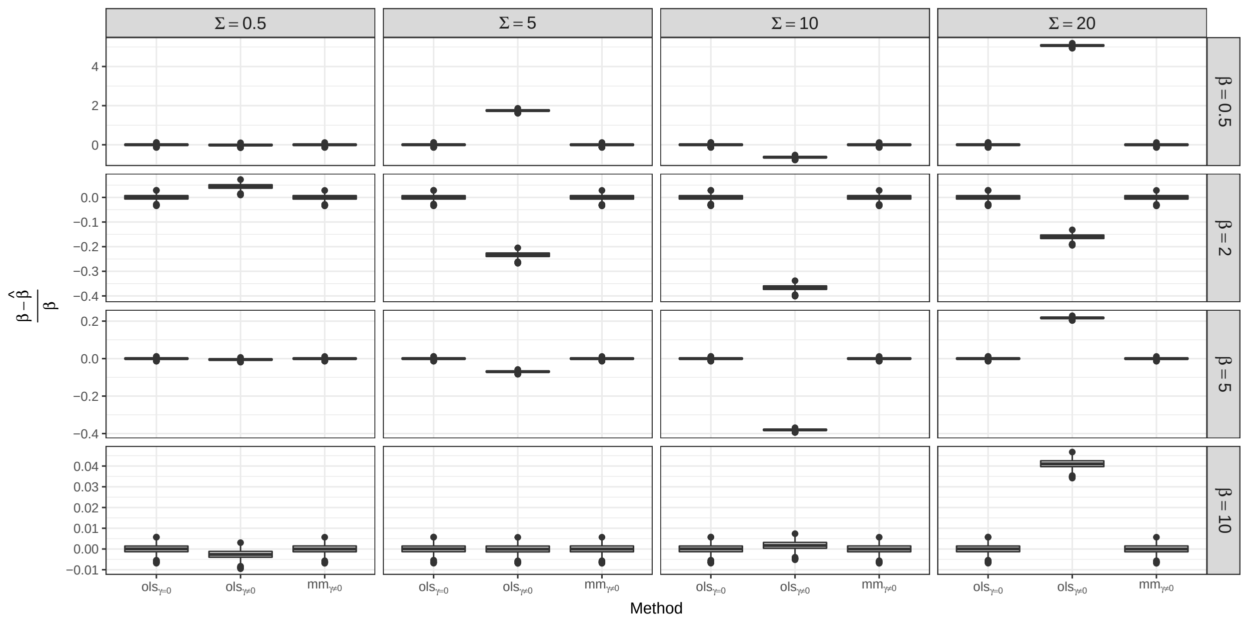
```
dataPLOT$Bias = (rep(resSim$estimateAll[[1]],3)-dataPLOT$betaEs)/rep(resSim$estimateAll[[1]],3)

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$Sigm <- gsub(0.5, "Sigma == 0.5",
           gsub(10, "Sigma == 10",
           gsub(20, "Sigma == 20",dataPLOT$Sigm)))
dataPLOT[dataPLOT$Sigm==5,'Sigm']="Sigma == 5"
dataPLOT$Sigm = factor(dataPLOT$Sigm,levels = c("Sigma == 0.5","Sigma == 5",
                                                "Sigma == 10","Sigma == 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 etimates 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("$\\frac{\\beta-\\hat{\\beta}}{\\beta}$"))+
  scale_x_discrete(labels=c(TeX("$ols_{\\gamma=0}$"),TeX("$ols_{\\gamma\\neq 0}$"),
                            TeX("$mm_{\\gamma\\neq 0}$")))
```



Running models on the average of the simulations

```r
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)),
```

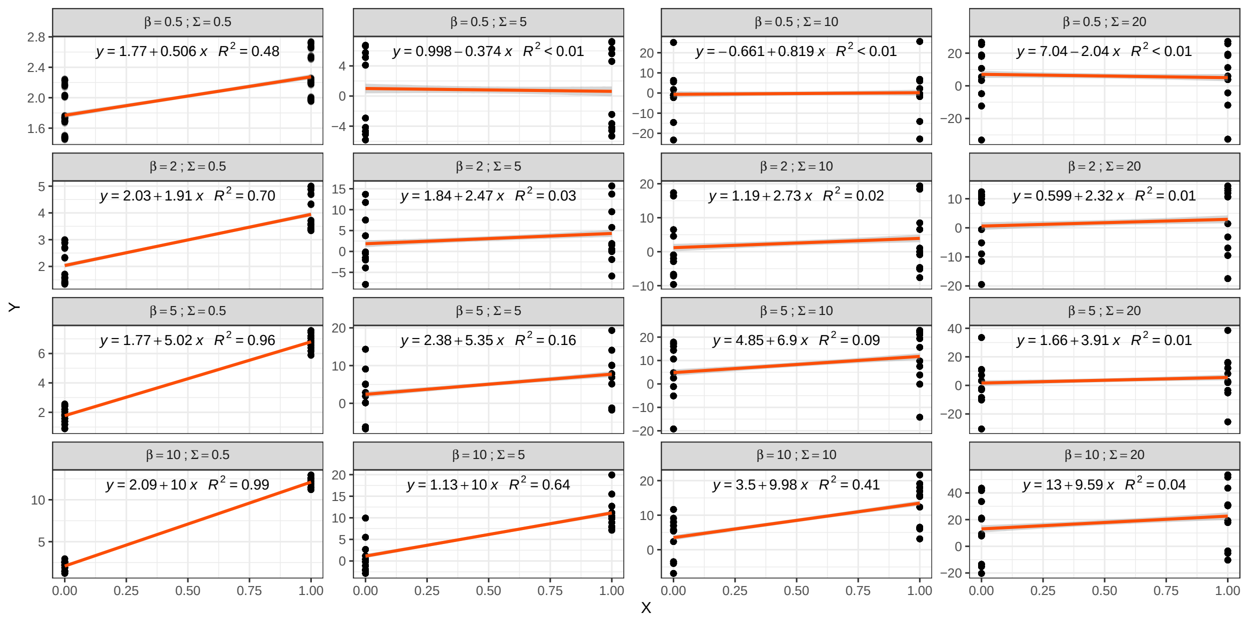
Table 4: OLS and Mixed Model estimates with 500 Monte-Carlo replicates

				$\boldsymbol{\Sigma = 0.5}$		$\boldsymbol{\Sigma = 5}$		$\boldsymbol{\Sigma = 10}$		$\boldsymbol{\Sigma = 20}$	
β	Model	γ	Estimate	μ	SE	μ	SE	μ	SE	μ	SE
$\boldsymbol{\beta = 0.5}$	OLS	$\gamma \neq 0$	$\hat{\boldsymbol{\beta}}$	0.5064	0.0217	-0.3745	0.0217	0.8187	0.0217	-2.0375	0.0217
			$\boldsymbol{SE}(\hat{\boldsymbol{\beta}})$	0.0326	9×10^{-4}	0.4647	0.001	1.0857	0.001	1.5799	9×10^{-4}
		$\gamma = 0$	$\hat{\boldsymbol{\beta}}$	0.4996	0.0217	0.4996	0.0217	0.4996	0.0217	0.4996	0.0217
			$\boldsymbol{SE}(\hat{\boldsymbol{\beta}})$	0.0223	7×10^{-4}	0.0223	7×10^{-4}	0.0223	7×10^{-4}	0.0223	7×10^{-4}
	MM	$\gamma \neq 0$	$\hat{\boldsymbol{\beta}}$	0.4997	0.0218	0.4995	0.0218	0.4996	0.0218	0.4996	0.0218
			$\boldsymbol{SE}(\hat{\boldsymbol{\beta}})$	0.0225	7×10^{-4}	0.0225	7×10^{-4}	0.0225	7×10^{-4}	0.0225	7×10^{-4}
$\boldsymbol{\beta = 2}$	OLS	$\gamma \neq 0$	$\hat{\boldsymbol{\beta}}$	1.9112	0.0217	2.466	0.0217	2.7331	0.0217	2.3199	0.0217
			$\boldsymbol{SE}(\hat{\boldsymbol{\beta}})$	0.0602	9×10^{-4}	0.5923	0.001	0.8061	0.001	0.9776	0.001
		$\gamma = 0$	$\hat{\boldsymbol{\beta}}$	1.9996	0.0217	1.9996	0.0217	1.9996	0.0217	1.9996	0.0217
			$\boldsymbol{SE}(\hat{\boldsymbol{\beta}})$	0.0223	7×10^{-4}	0.0223	7×10^{-4}	0.0223	7×10^{-4}	0.0223	7×10^{-4}
	MM	$\gamma \neq 0$	$\hat{\boldsymbol{\beta}}$	1.9993	0.0218	1.9996	0.0218	1.9996	0.0218	1.9996	0.0218
			$\boldsymbol{SE}(\hat{\boldsymbol{\beta}})$	0.0225	7×10^{-4}	0.0225	7×10^{-4}	0.0225	7×10^{-4}	0.0225	7×10^{-4}
$\boldsymbol{\beta = 5}$	OLS	$\gamma \neq 0$	$\hat{\boldsymbol{\beta}}$	5.0246	0.0217	5.3468	0.0217	6.9008	0.0217	3.914	0.0217
			$\boldsymbol{SE}(\hat{\boldsymbol{\beta}})$	0.051	0.001	0.5455	0.001	1.0104	9×10^{-4}	1.425	0.001
		$\gamma = 0$	$\hat{\boldsymbol{\beta}}$	4.9996	0.0217	4.9996	0.0217	4.9996	0.0217	4.9996	0.0217
			$\boldsymbol{SE}(\hat{\boldsymbol{\beta}})$	0.0223	7×10^{-4}	0.0223	7×10^{-4}	0.0223	7×10^{-4}	0.0223	7×10^{-4}
	MM	$\gamma \neq 0$	$\hat{\boldsymbol{\beta}}$	4.9997	0.0218	4.9996	0.0218	4.9996	0.0218	4.9996	0.0218
			$\boldsymbol{SE}(\hat{\boldsymbol{\beta}})$	0.0225	7×10^{-4}	0.0225	7×10^{-4}	0.0225	7×10^{-4}	0.0225	7×10^{-4}
$\boldsymbol{\beta = 10}$	OLS	$\gamma \neq 0$	$\hat{\boldsymbol{\beta}}$	10.0259	0.0217	10.0001	0.0217	9.9826	0.0217	9.589	0.0217
			$\boldsymbol{SE}(\hat{\boldsymbol{\beta}})$	0.0537	0.001	0.3396	0.001	0.5324	0.001	1.9954	0.001
		$\gamma = 0$	$\hat{\boldsymbol{\beta}}$	9.9996	0.0217	9.9996	0.0217	9.9996	0.0217	9.9996	0.0217
			$\boldsymbol{SE}(\hat{\boldsymbol{\beta}})$	0.0223	7×10^{-4}	0.0223	7×10^{-4}	0.0223	7×10^{-4}	0.0223	7×10^{-4}
	MM	$\gamma \neq 0$	$\hat{\boldsymbol{\beta}}$	9.9997	0.0218	9.9996	0.0218	9.9996	0.0218	9.9996	0.0218
			$\boldsymbol{SE}(\hat{\boldsymbol{\beta}})$	0.0225	7×10^{-4}	0.0225	7×10^{-4}	0.0225	7×10^{-4}	0.0225	7×10^{-4}

```
                      unlist(OLS(resSim$yWithoutSimMean[,i], X)),
                      beta,SE,Sigm)
  k <- k+1
}
resMeanSim <- as.data.frame(resMeanSim)
```

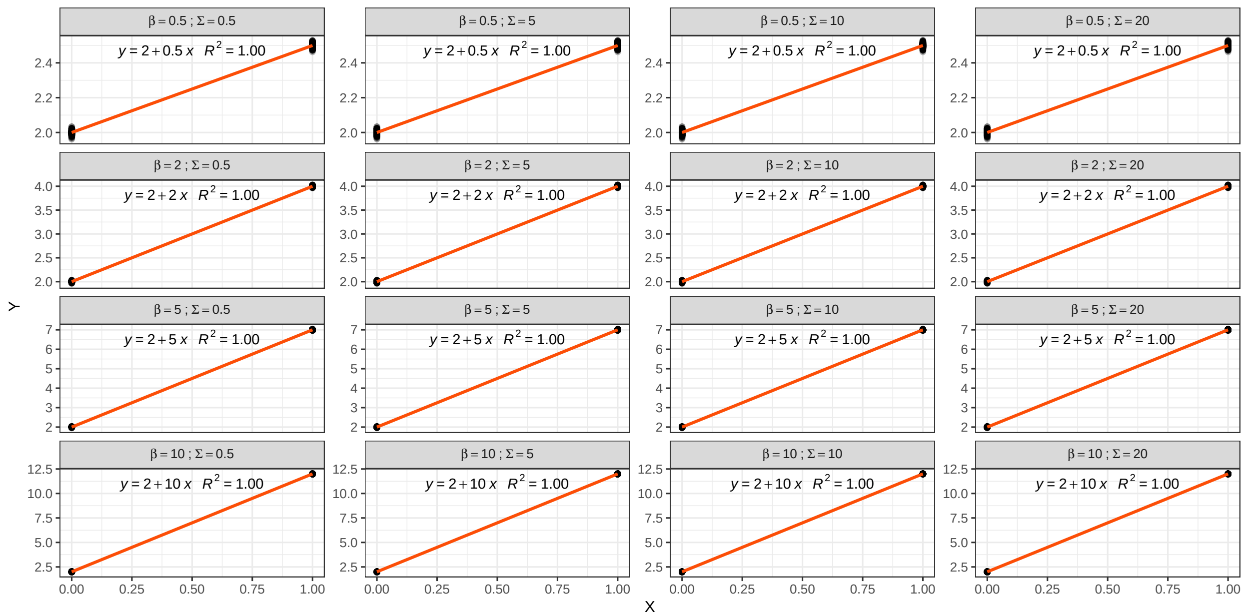
Visualization of the models

```
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)
```

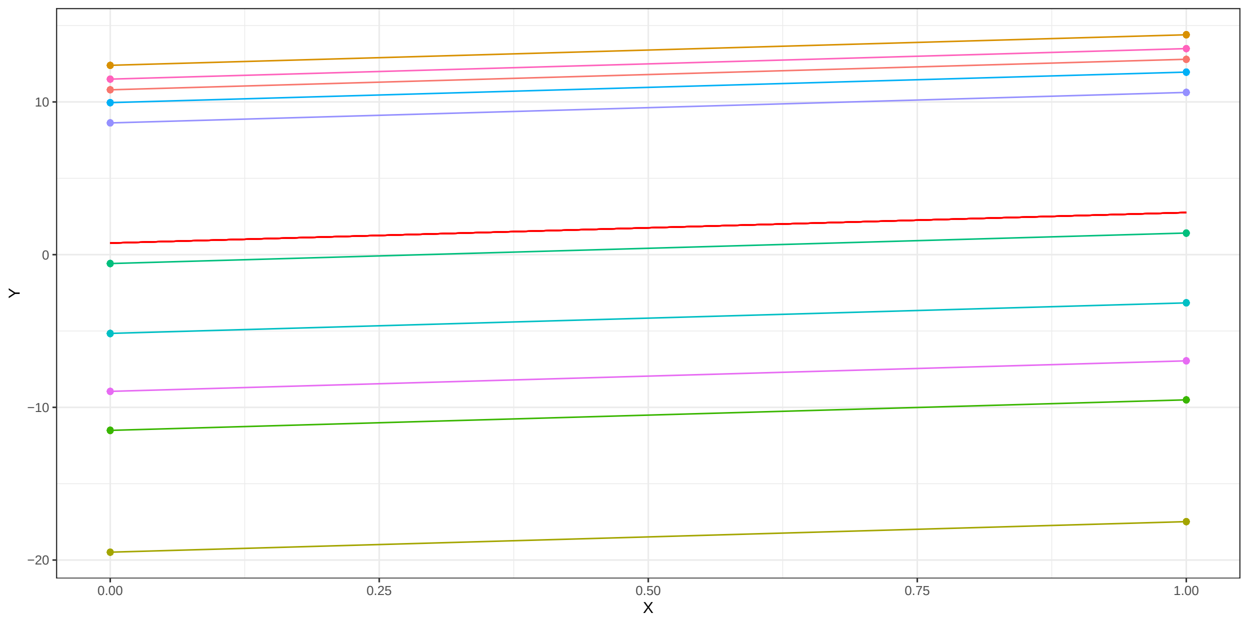



```
dataMod <- melt(resSim$yWithoutSimMean, id.vars=c("X","G"),value.name = "Y")
dataMod$variable <- recode_factor(dataMod$variable,
              "0.5_0.5"="beta == 0.5 ~ ';' ~ Sigma == 0.5",
              "0.5_5"="beta == 0.5 ~ ';' ~ Sigma == 5","0.5_10"="beta == 0.5 ~ ';' ~ Sigma == 10",
        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



CCDF application

Code

```
CCDF <- function(Y, X, Z = NULL, sample_group = NULL,
                  space_y = FALSE, number_y = length(unique(Y)), plt=FALSE){
  cdf <- list()
  if (is.null(Z)){
    colnames(X) <- sapply(seq_len(ncol(X)), function(i){paste0('X', i)})
    modelmat <- as.matrix(model.matrix(~.,data=X))
  }
  else{
    colnames(X) <- sapply(seq_len(ncol(X)), function(i){paste0('X', i)})
    colnames(Z) <- sapply(seq_len(ncol(Z)), function(i){paste0('Z', i)})
    modelmat <- as.matrix(model.matrix(~.,data=cbind(X,Z)))
  }
  indexes_X <- which(substring(colnames(modelmat), 1, 1) == "X")
  p_X <- length(indexes_X)
  sample_group <- sample_group[,1]
  modX_OLS <- modelmat[, c(1, indexes_X), drop = FALSE]
```

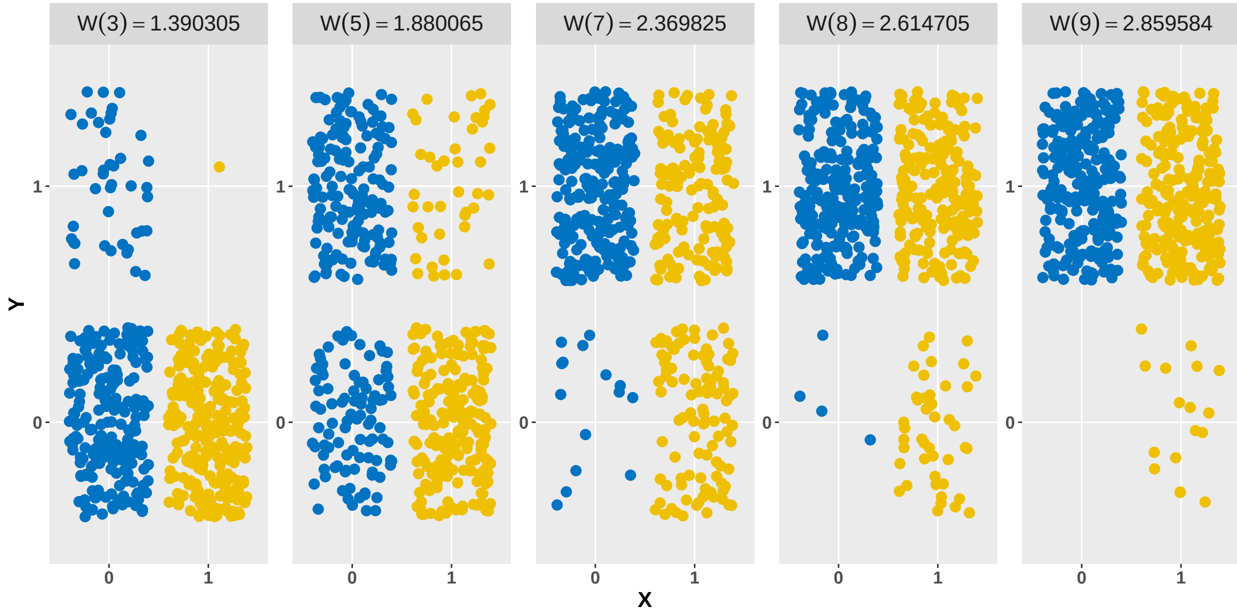
```
n_Y_all <- length(Y)
Y <- as.numeric(Y)

if (space_y){
  y <-  seq(ifelse(length(which(Y==0))==0,min(Y),min(Y[-which(Y==0)])),max(Y[-which.max(Y)]),length.ou
}
else{
  y <- sort(unique(Y))
}
n_y_unique <- length(y)

ind_X <- which(substring(colnames(modelmat),1,1)=="X")
betaOLS <- matrix(NA,10,p_X)
betaMM <- matrix(NA,10,p_X)
seuils <- matrix(NA,10,p_X)
SEsOLS <- matrix(NA,10,p_X)
SEsMM <- matrix(NA,10,p_X)
indi_pi <- matrix(NA,n_Y_all,(n_y_unique-1))
H1 <- as.formula("Y ~  X + (1|groups)")
dataPlots = list()

if (length(y)==11){
  for (i in 1:(n_y_unique-1)){
    seuils[i,] <- y[i]
    indi_Y <- 1*(Y<=y[i])
    indi_pi[,i] <- indi_Y
    reg <- OLS(indi_Y,modelmat[,-1])
    betaOLS[i,] <- round(reg$betaOLS,4)
    SEsOLS[i,] <- round(reg$SE,4)
    dataMix <- cbind.data.frame(X=X$X1,groups=sample_group)
    dataGlm <- cbind.data.frame(Y = indi_Y, dataMix)
    glm1 <- try({ lme4::lmer(H1, data = dataGlm,REML = TRUE)}, silent = T)
```

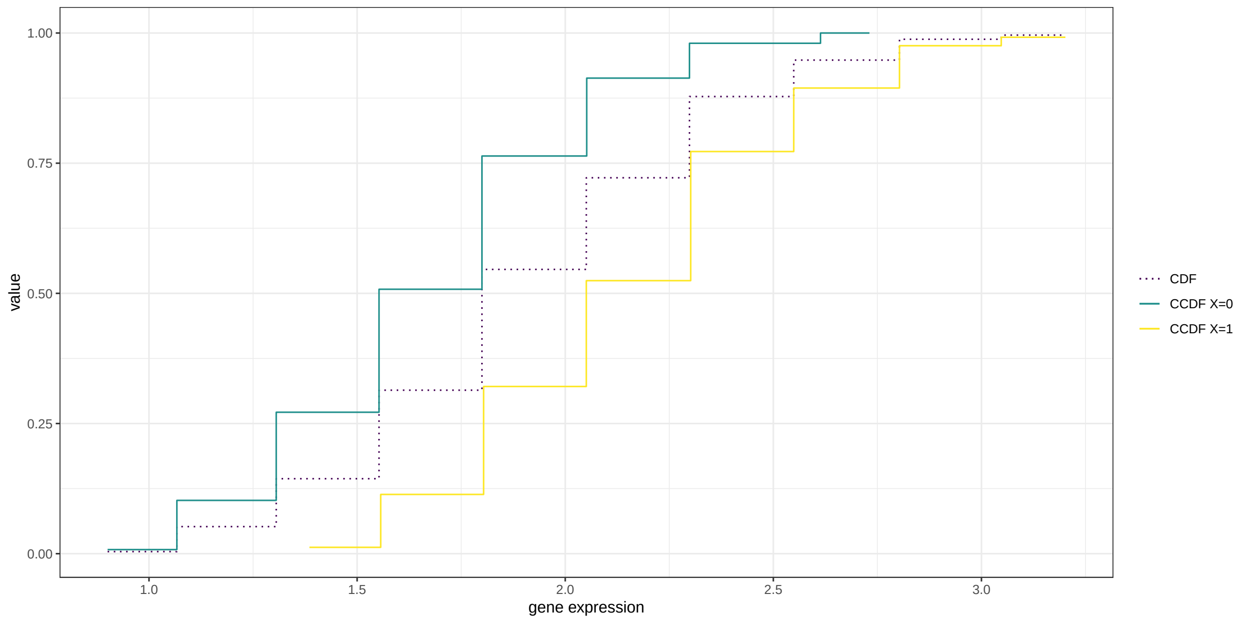
```
    if(class(glm1) != "try-error") {
      betaMM[i,] <- round(lme4::fixef(glm1)[ind_X],4)
      SEsMM[i,] <- round(sqrt(diag(as.matrix(vcov(glm1))))[-1],4)
    }
    if(plt){
      if(i%in%c(3,5,7,8,9)){
      dataPlots[[paste0("",y[i])]] <- dataGlm$Y
      }
    }
  }
  betaOLS <- as.vector(betaOLS)
  betaMM <- as.vector(betaMM)
  SEsOLS <- as.vector(SEsOLS)
  SEsMM <- as.vector(SEsMM)
  if(plt)
  {
    valsSeuils <- paste0("W(",c(3,5,7,8,9),") == ", names(dataPlots))
    dataPlots <- cbind.data.frame(as.data.frame(dataPlots),X=factor(X$X1), G=sample_group)
    dataPlots = melt(dataPlots, id.vars=c("X","G"),value.name = "Y")
    dataPlots$Y <- factor(dataPlots$Y)
    dataPlots$variable <- as.factor(rep(valsSeuils,each=nrow(X)))
    plts <- ggplot(dataPlots,aes(x=X, y=Y))+
        geom_jitter(aes(color=X),size=3)+
        ggpubr::color_palette("jco")+
        facet_wrap(~variable,scales = 'free_y',ncol = 5,labeller = label_parsed)+
         theme(
          text = element_text(size=15, face = "bold"),
          strip.text.x = element_text(
            size = 15, face = "bold"
            ),
          strip.text.y = element_text(
            size = 15, face = "bold"
```

```
              )
            )+ theme(legend.position="none")
       print(plts)

       cdf <- ccdf::CCDF(Y = Y, X = X, number_y = number_y, space_y = TRUE)
       df_plot <- data.frame("y" = cdf$y, "x" = cdf$x, "cdf" = cdf$cdf, "ccdf" = cdf$ccdf)
        levels(df_plot$x) <- unique(X[,1])
          df_plot$x <- ordered(df_plot$x, levels = levels(df_plot$x))
          l_X <- length(unique(X[,1]))

          plotCDF <- ggplot() + ggtitle(colnames(Y)) +
            geom_step(data = df_plot, aes_string(x = "y", y = "cdf",
                                                color = shQuote(viridisLite::viridis(n=(l_X+1))[1])),
                                                 size = 0.5,linetype="dotted") +
            geom_step(data = df_plot, aes_string(x = "y", y = "ccdf", color = "x"), size = 0.5) +
            scale_color_manual(name = "", labels=c("CDF", paste0("CCDF X=", levels(df_plot$x)[ordered(le
                                values = viridisLite::viridis(n=l_X+1),
                                guide = guide_legend(override.aes = list(linetype = c("dotted",rep("soli
            ylab("value") + theme_bw() + theme(plot.title = element_text(hjust = 0.5))
            print(plotCDF)
    }
    return(c(betaOLS,betaMM))
  }
}
```

With alea effect

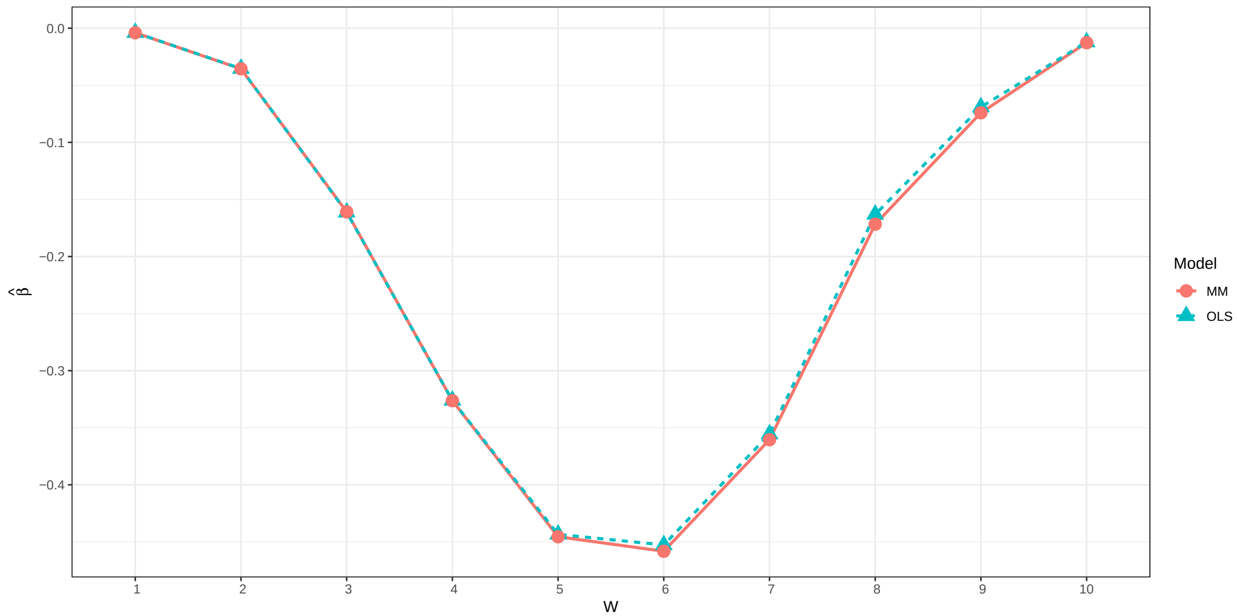
```
Y <- dfWith$Y
X <- data.frame(X=dfWith$X)
G <- data.frame(G = dfWith$G)
res <- CCDF(Y, X, sample_group = G,number_y = 11,space_y = TRUE,plt = TRUE)
```





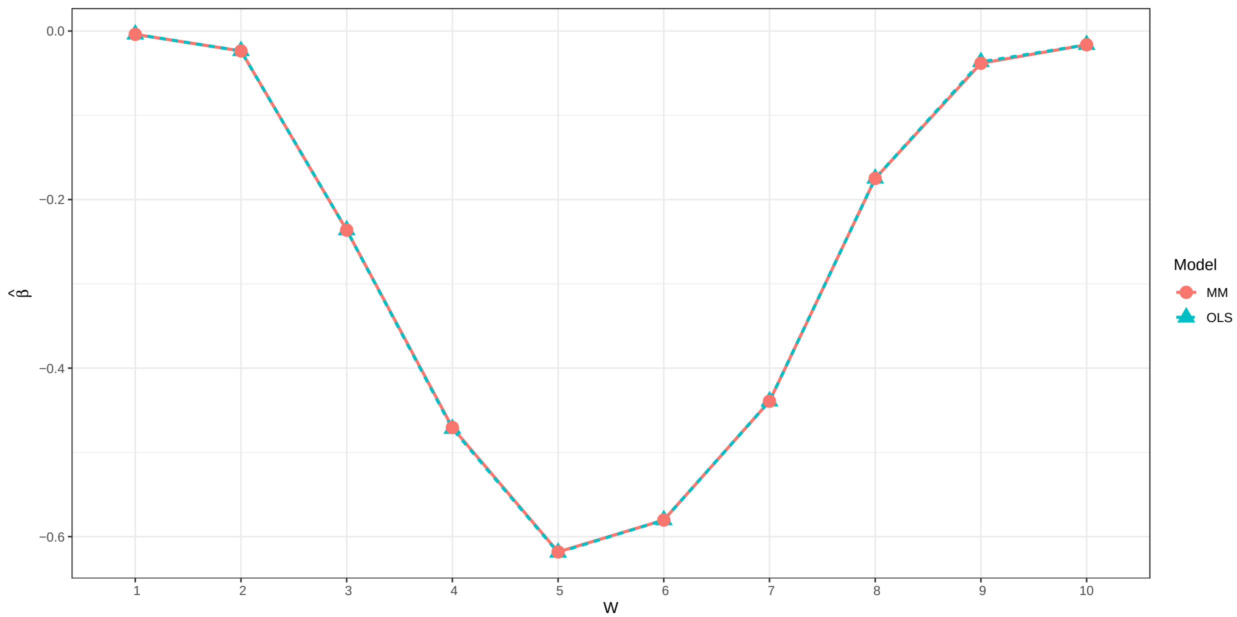

```
ccdfRes <- data.frame(indBeta=factor(rep(1:10,2)),Estimate=res[1:20],Model=factor(rep(c("OLS","MM"),eac
 ggplot(ccdfRes, aes(indBeta, Estimate, group=Model, colour=Model)) +
     geom_line(aes(linetype=Model), size=1) +
  geom_point(aes(shape=Model),size=4)+
   scale_x_discrete(limits=1:10)+
   labs(x = TeX("$W$"),y=TeX("$\\hat{\\beta}$"))+
  theme(strip.text.x = element_text(size=12, face="bold"),
        strip.text.y = element_text(size=12, face="bold"))+
```

```
  theme_bw()
```



Without alea effect

```
Y <- dfWithout$Y

X <- data.frame(X=dfWithout$X)

G <- data.frame(G = dfWithout$G)


res <- CCDF(Y, X, sample_group = G,number_y = 11,space_y = TRUE)

ccdfRes <- data.frame(indBeta=factor(rep(1:10,2)),Estimate=res[1:20],Model=factor(rep(c("OLS","MM"),each

 ggplot(ccdfRes, aes(indBeta, Estimate, group=Model, colour=Model)) +

     geom_line(aes(linetype=Model), size=1) +

  geom_point(aes(shape=Model),size=4)+

   scale_x_discrete(limits=1:10)+

   labs(x = TeX("$W$"),y=TeX("$\\hat{\\beta}$"))+

  theme(strip.text.x = element_text(size=12, face="bold"),

        strip.text.y = element_text(size=12, face="bold"))+

   theme_bw()
```



CCDF with Monte carlo with alea effect

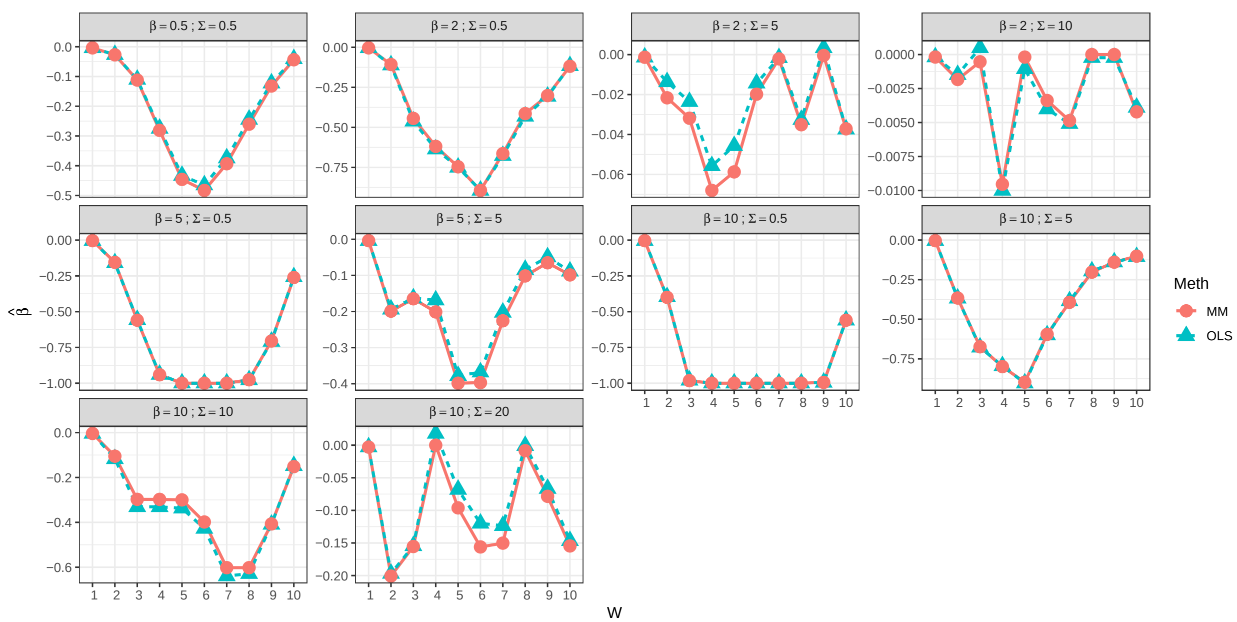
```
simCCDF <- function (intercept=2, fixefEffects = c(.5,2,5,10), sds = c(.5,5,10,20),
                    K=10, nK=50, sims=5000){
  estimateAll <- matrix(NA, length(fixefEffects)*length(sds)*sims, 22)


  resCCDFprime <- CCDFBIZ <- list()
  yWithSimMean <- yWithoutSimMean <- matrix(NA, 500, length(fixefEffects)*length(sds))
  n <- K * nK
  G <- rep(1:K, each = nK)
  X <- rbinom(n,size=1,prob = .5)
  k = 1
  kk = 1
  yNotConv <- list()
  for (fixefEffect in fixefEffects) {
    for (sdUnit in sds) {
      set.seed(seeds24_03_22[kk])
        aleaEffect <- rnorm(K, sd = sdUnit)
        resCCDF <- matrix(0, sims, 20)
```

```
      yOlsWith <- yOlsWithout <- matrix(0, 500, sims)
      for(i in 1:sims){
        set.seed(seeds24_03_22[i])
        options(warn=-1)
        bias <- rnorm(n, sd = .25)
        Y_with <- intercept + fixefEffect * X + aleaEffect[G] + bias
        Y_without <- intercept + fixefEffect * X + bias
        CCDFRES.1 <- CCDF(Y = Y_with, X = data.frame(X=factor(X,levels = 0:1)),
                          sample_group = data.frame(G=factor(G,levels = 1:10)),number_y = 11,space_y =
        CCDFBIZ[[as.character(paste0(fixefEffect,"_",sdUnit))]] <- data.frame(Y=Y_with,
                                                                               X=factor(X,levels = 0:1)
                                                                               G=factor(G,levels = 1:10
        if(any(is.na(CCDFRES.1[1:20]))){
          yNotConv[[as.character(paste0(fixefEffect,"_",sdUnit))]] <- Y_with
          next
        }

        resCCDF[i,] = CCDFRES.1[1:20]
        estimateAll[k,] <- c(CCDFRES.1,sdUnit,i)
        yOlsWith[,i] <- Y_with; yOlsWithout[,i] <- Y_without
        k <- k+1
      }
      resCCDF <- resCCDF[stats::complete.cases(resCCDF),]
      yWithSimMean[,kk] <- rowMeans(yOlsWith)
      yWithoutSimMean[,kk] <- rowMeans(yOlsWithout)
      kk <- kk + 1
      M <- apply(resCCDF, 2, mean)
      S <- apply(resCCDF, 2, sd)
      resCCDFprime[[as.character(paste0(fixefEffect,"_",sdUnit))]] <- M
  }
}
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)
  return (list("estimateAll"= estimateAll[stats::complete.cases(estimateAll),],
               "resCCDFprime"= resCCDFprime,
               "yWithSimMean"= yWithSimMean[stats::complete.cases(yWithSimMean),],
               'yWithoutSimMean'= yWithoutSimMean[stats::complete.cases(yWithoutSimMean),],
               'CCDFBIZ'=CCDFBIZ,
               'yNotConv'=yNotConv,
               'X'=X,'G'=G))
}
resSimCCDF <- simCCDF(sims = 500)
```

```
dataPlot <- data.frame(indBeta=factor(rep(rep(1:10,2),length(names(resSimCCDF$resCCDFprime)))),
                       Values=unlist(resSimCCDF$resCCDFprime,use.names =FALSE),
                       Parms=rep(names(resSimCCDF$resCCDFprime),each=20),
                       Meth = rep(rep(c("OLS","MM"),each=10),length(names(resSimCCDF$resCCDFprime))))

notCvg <- names(resSimCCDF$resCCDFprime[sapply(resSimCCDF$resCCDFprime,sum)==0])

dataPlot <- dataPlot[!(dataPlot$Parms %in% notCvg),]
dataPlot$Parms <- recode_factor(dataPlot$Parms,
              "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(dataPlot, aes(indBeta, Values, group=Meth, colour=Meth)) +
     geom_line(aes(linetype=Meth), size=1) +
  geom_point(aes(shape=Meth),size=4)+
   scale_x_discrete(limits=1:10)+
   labs(x = TeX("$W$"),y=TeX("$\\hat{\\beta}$"))+
  theme(strip.text.x = element_text(size=12, face="bold"),
        strip.text.y = element_text(size=12, face="bold"))+
   theme_bw()+
  facet_wrap(~Parms,scales = 'free_y',ncol = 4,labeller = label_parsed)
```




```
dataPlot <- data.frame(X=factor(rep(resSimCCDF$X,length(notCvg))),
                       G=factor(rep(resSimCCDF$G,length(notCvg))),
                       Values=unlist(resSimCCDF$yNotConv[notCvg],use.names =FALSE),Parms=rep(notCvg,eac
ggplot(dataPlot) +
  geom_boxplot(aes(x=X, y=Values, fill = G)) +
  scale_x_discrete(expand = c(0, 0.5)) +
  theme_bw()+
```

```
facet_wrap(~Parms, scales = 'free_y',ncol = 3)
```