Section 3.3.4 - Unbalanced Linear model

Marouane IL IDRISSI

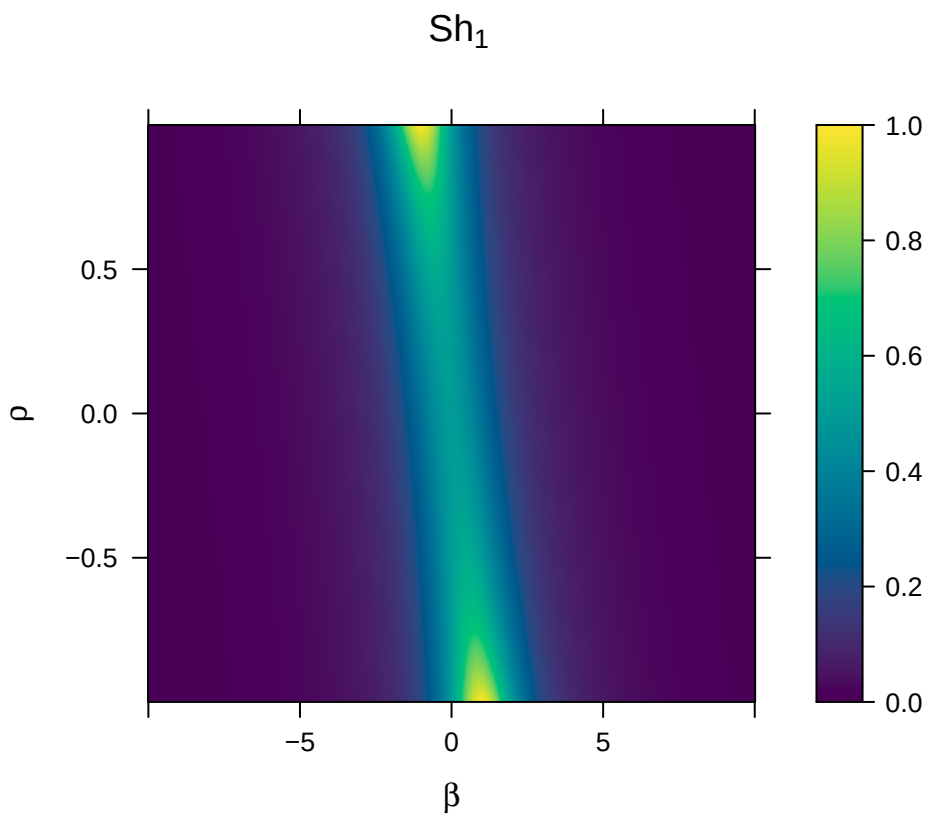
03/12/2023

Computing the analytical allocations

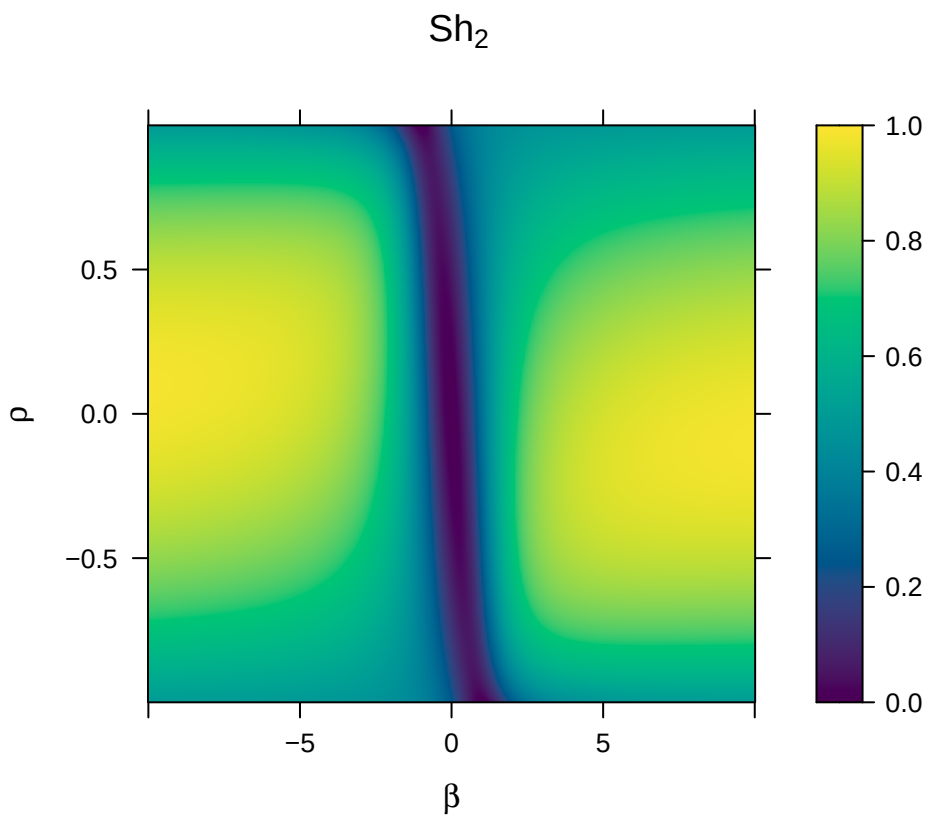
```
# Model: G(X) = X1 + bX2 + X3

################################################################################
# Function to compute the analytical Shapley and PME Values

computeAllocs<-function(b, rho){

  #Output variance
  vy=2 + b^2 + 2*b*rho

  #(Normalized) Shapley effects
  sh1=1/vy
  sh2=(b^2 + b*rho + 0.5*(rho^2*(1-b^2)))/vy
  sh3=(1+b*rho - (0.5*rho^2*(1-b^2)))/vy

  #(Normalized) PME
  pme1=1/vy
  pme2=(b^2*(1+b^2 +2*rho*b))/(vy*(1+b^2))
  pme3=(1+b^2 +2*rho*b)/(vy*(1+b^2))

  res=cbind(sh1, sh2, sh3, pme1,pme2,pme3)
  return(res)
}
```

```
################################################################################
# Sequence of correlation values
rhos<-seq(-0.999,0.999,0.001)

################################################################################
# Sequence of beta values
betas<-seq(-10,10,0.025)

################################################################################
# Grid of crossed rho/beta values to compute
grid.rhobeta<-expand.grid(betas, rhos)

resAllocs=cbind(grid.rhobeta, computeAllocs(grid.rhobeta[,1], grid.rhobeta[,2]))
colnames(resAllocs)[c(1,2)]=c("beta", "rho")
```

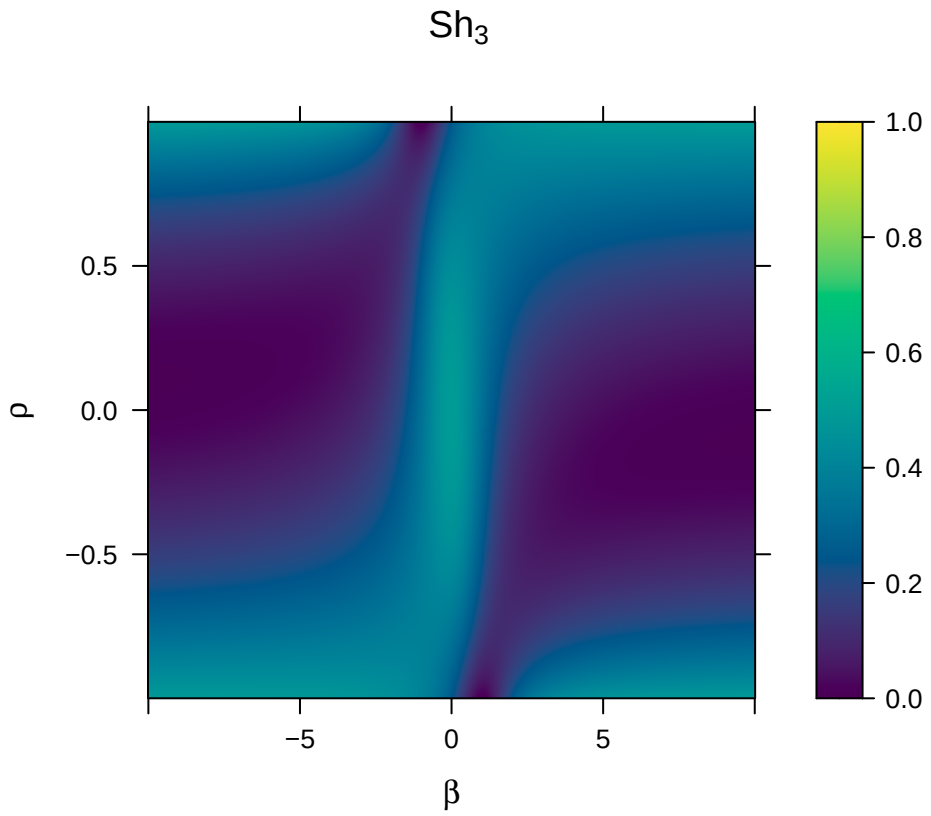
```
par(mfrow=c(3,1),
    cex=2)
levelplot(sh1~beta*rho,
          data=resAllocs,
          col.regions=hcl.colors(10000),
          cuts=10000,
          main=expression(Sh[1]),
          ylab=expression(rho),
          xlab=expression(beta),
          at=seq(0,1,0.0001),
          cex=2)
```



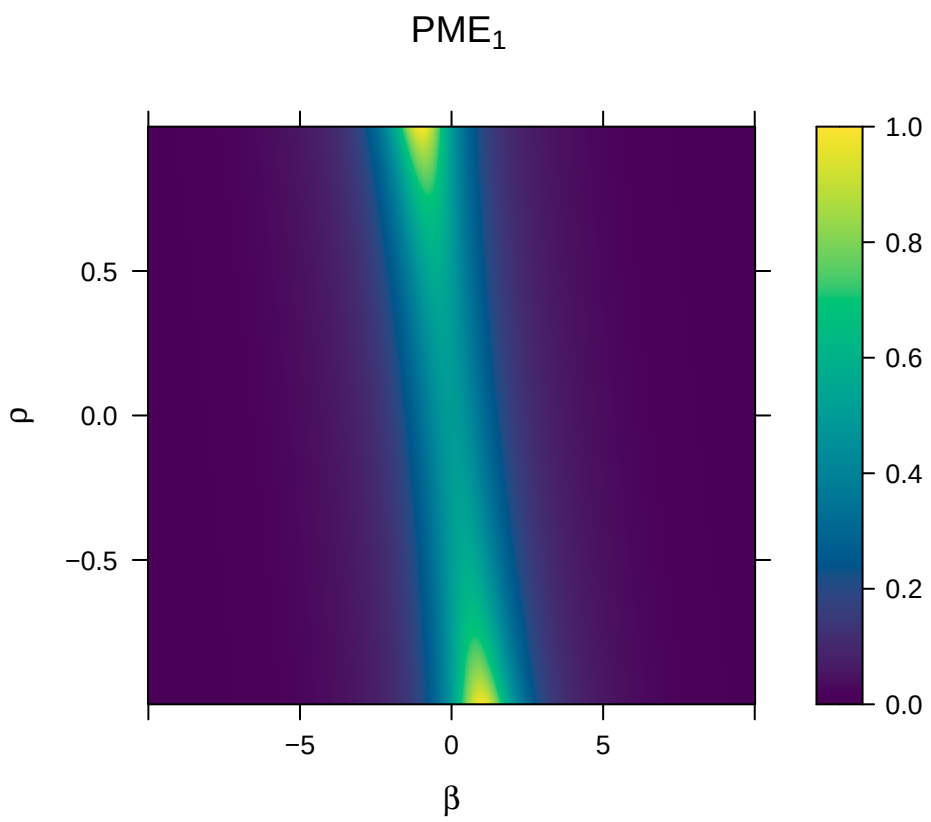

```
levelplot(sh2~beta*rho,
          data=resAllocs,
          col.regions=hcl.colors(10000),
          cuts=10000,
          main=expression(Sh[2]),
          ylab=expression(rho),
          xlab=expression(beta),
          at=seq(0,1,0.0001),
          cex=2)
```



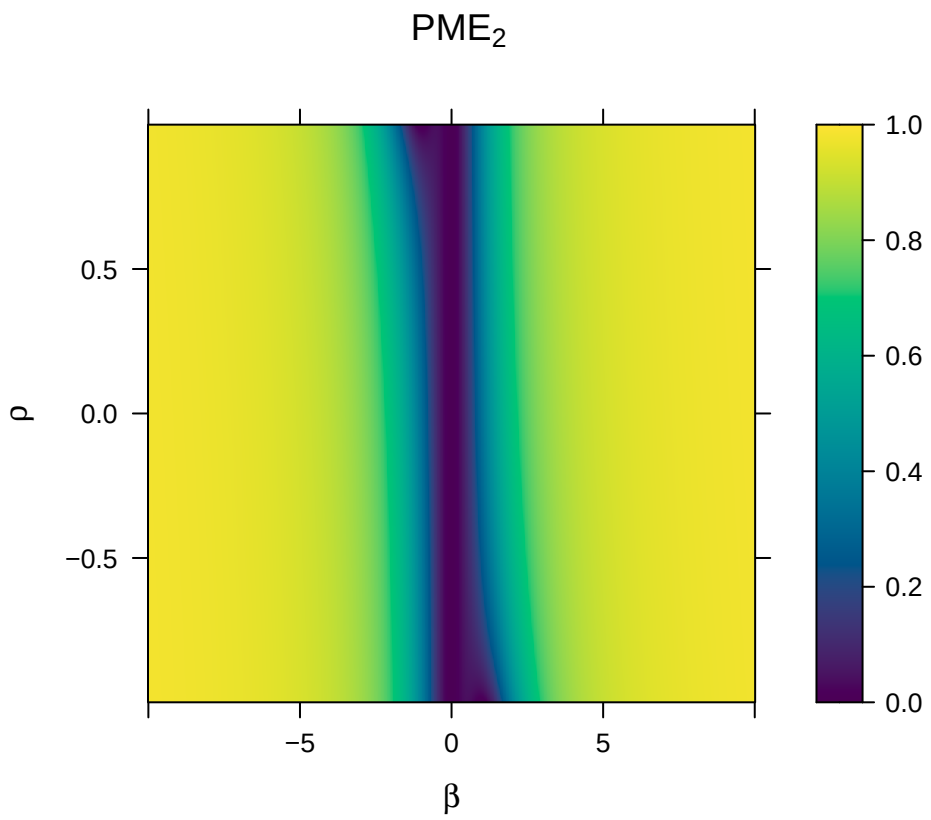

```
levelplot(sh3~beta*rho,
          data=resAllocs,
          col.regions=hcl.colors(10000),
          cuts=10000,
          main=expression(Sh[3]),
          ylab=expression(rho),
          xlab=expression(beta),
          at=seq(0,1,0.0001),
          cex=2)
```



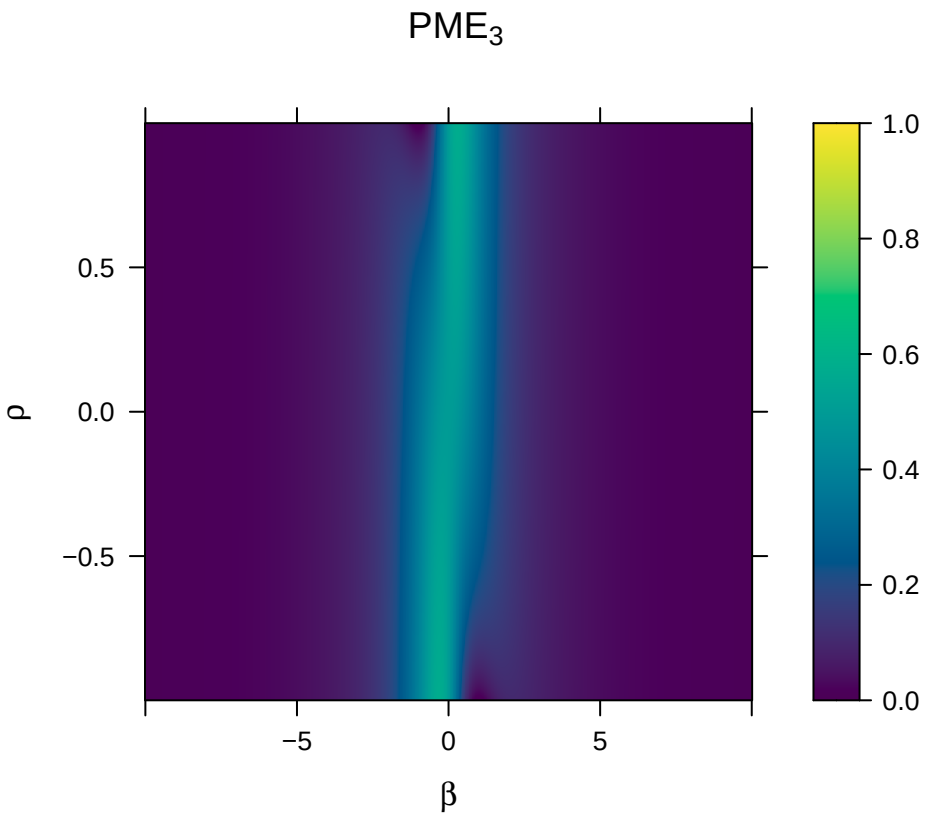

```
par(mfrow=c(3,1),
    cex=2)
levelplot(pme1~beta*rho,
          data=resAllocs,
          col.regions=hcl.colors(10000),
          cuts=10000,
          main=expression(PME[1]),
          ylab=expression(rho),
          xlab=expression(beta),
          at=seq(0,1,0.0001),
          cex=2)
```




```
levelplot(pme2~beta*rho,
          data=resAllocs,
          col.regions=hcl.colors(10000),
          cuts=10000,
          main=expression(PME[2]),
          ylab=expression(rho),
          xlab=expression(beta),
          at=seq(0,1,0.0001),
          cex=2)
```



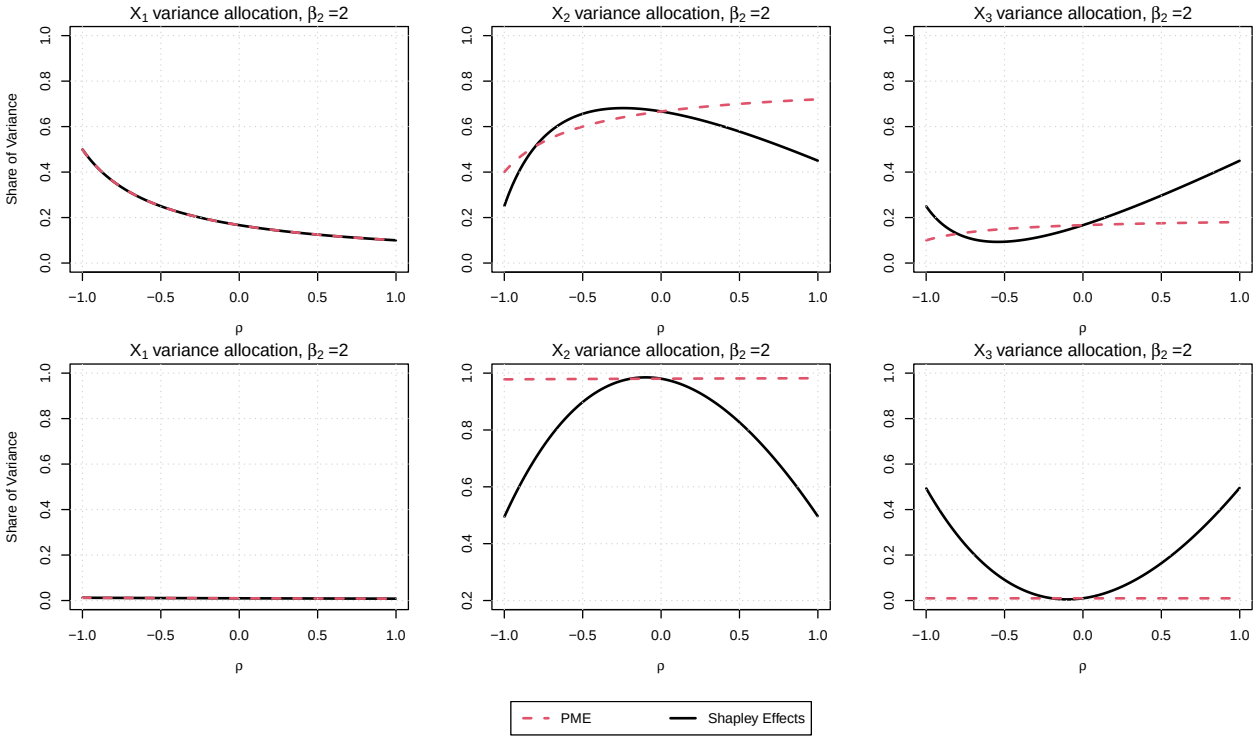

```
levelplot(pme3~beta*rho,
          data=resAllocs,
          col.regions=hcl.colors(10000),
          cuts=10000,
          main=expression(PME[3]),
          ylab=expression(rho),
          xlab=expression(beta),
          at=seq(0,1,0.0001),
          cex=2)
```



#Punctual evaluation

```
par(mar=c(4,4,1.5,1))

layout(matrix(c(1,2,3,
                4,5,6,
                7,7,7), nrow=3, ncol=3, byrow=T))

# Beta=2
plot(rhos, resAllocs[resAllocs[,1]==2,3],type='l',
     ylim=c(0,1),
     xlab=expression(rho),
     main=expression(X[1]~'variance allocation,'~beta[2]~'=2'),
     ylab="Share of Variance",
     lwd=2)
lines(rhos, resAllocs[resAllocs[,1]==2,6], lty=2, col=2,
     lwd=2)
grid()

plot(rhos, resAllocs[resAllocs[,1]==2,4],type='l',
     ylim=c(0,1),
     xlab=expression(rho),
     main=expression(X[2]~'variance allocation,'~beta[2]~'=2'),
     ylab="",
     lwd=2)
lines(rhos, resAllocs[resAllocs[,1]==2,7], lty=2, col=2,
     lwd=2)
grid()
```

```
plot(rhos, resAllocs[resAllocs[,1]==2,5],type='l',
     ylim=c(0,1),
     xlab=expression(rho),
     main=expression(X[3]~'variance allocation,'~beta[2]~'=2'),
     ylab="",
     lwd=2)
lines(rhos,resAllocs[resAllocs[,1]==2,8], lty=2, col=2,
     lwd=2)
grid()


# Beta=10
plot(rhos, resAllocs[resAllocs[,1]==10,3],type='l',
     ylim=c(0,1),
     xlab=expression(rho),
     main=expression(X[1]~'variance allocation,'~beta[2]~'=2'),
     ylab="Share of Variance",
     lwd=2)
lines(rhos, resAllocs[resAllocs[,1]==10,6], lty=2, col=2,
     lwd=2)
grid()

plot(rhos, resAllocs[resAllocs[,1]==10,4],type='l',
     ylim=c(0.2,1),
     xlab=expression(rho),
     main=expression(X[2]~'variance allocation,'~beta[2]~'=2'),
     ylab="",
     lwd=2)
lines(rhos, resAllocs[resAllocs[,1]==10,7], lty=2, col=2,
     lwd=2)
grid()

plot(rhos, resAllocs[resAllocs[,1]==10,5],type='l',
     ylim=c(0,1),
     xlab=expression(rho),
     main=expression(X[3]~'variance allocation,'~beta[2]~'=2'),
     ylab="",
     lwd=2)
lines(rhos,resAllocs[resAllocs[,1]==10,8], lty=2, col=2,
     lwd=2)
grid()

plot.new()
legend("top",
       lty=c(2,1),
       col=c(2,1),
       lwd=c(2,2),
       legend=c("PME", "Shapley Effects"),
       horiz=T)
```


X1 variance allocation, β2 =2
X2 variance allocation, β2 =2
X3 variance allocation, β2 =2
X1 variance allocation, β2 =2
X2 variance allocation, β2 =2
X3 variance allocation, β2 =2
Share of Variance
Share of Variance
ρ
ρ
ρ
ρ
ρ
ρ
−1.0
−0.5
0.0
0.5
1.0
0.0
0.2
0.4
0.6
0.8
1.0
PME
Shapley Effects