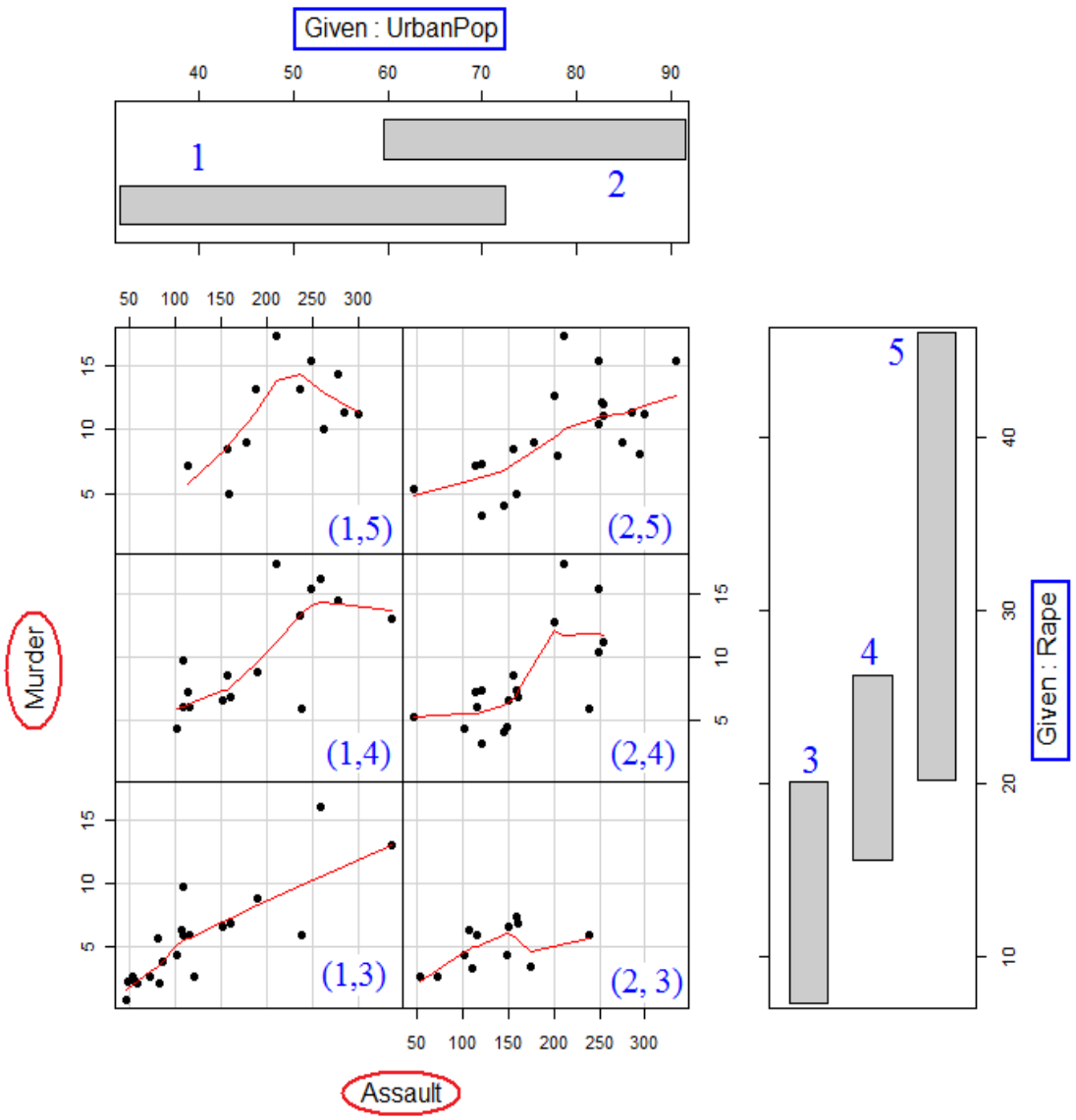
# 2. Gráficos

Duas variáveis condicionantes:

`UrbanPop` e `Rape`.

Número de intervalos (faixas) é diferente para cada variável condicionante.

```
> coplot(Murder ~ Assault
UrbanPop * Rape, number =
c(2, 3), pch = 20, cex =
1.5, panel = panel.smooth)
```



# 2. Gráficos

`UrbanPop` com três intervalos de igual comprimento.

```
> xyplot(Murder ~ Assault | cut(UrbanPop, 3))
```

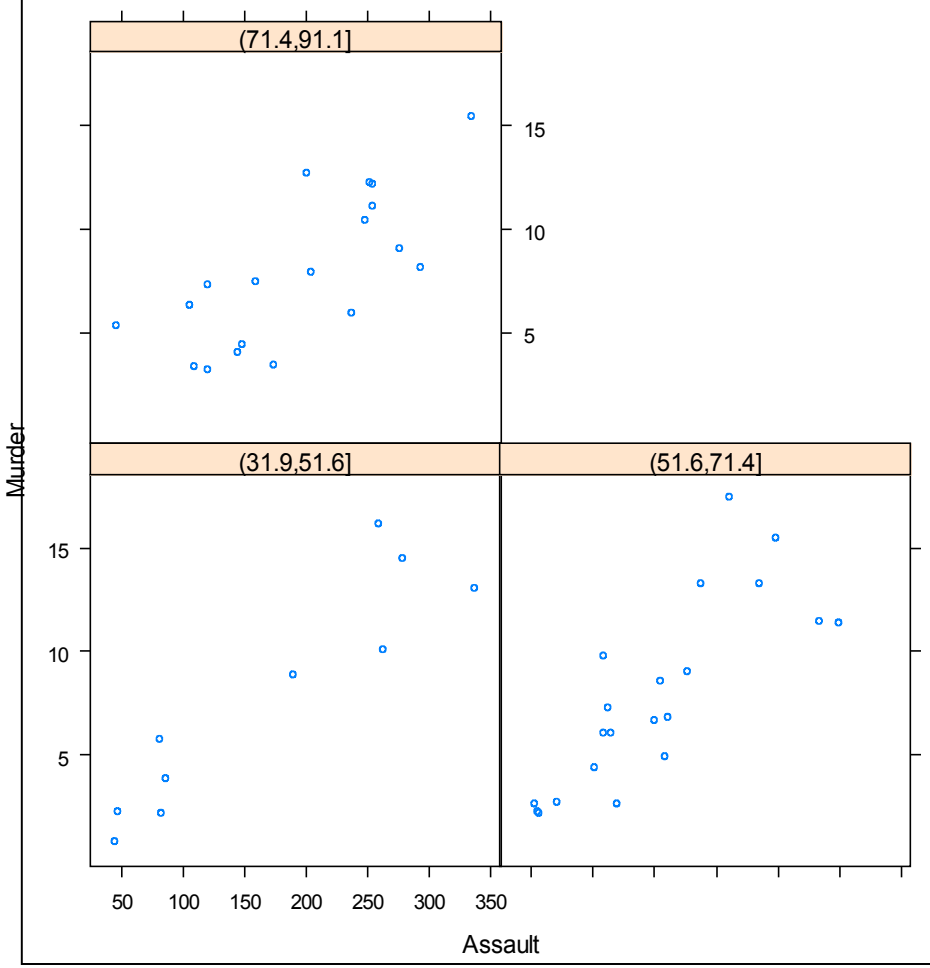
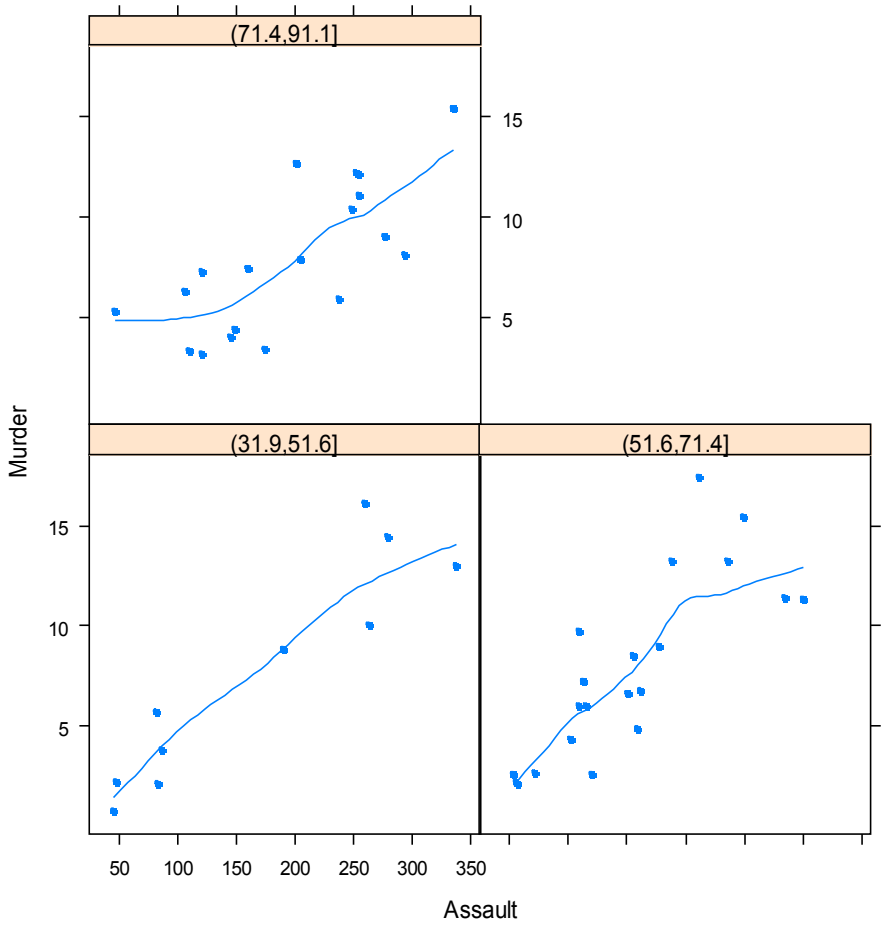



Gráfico com pontos (`p`) e linhas de tendência (`smooth`)

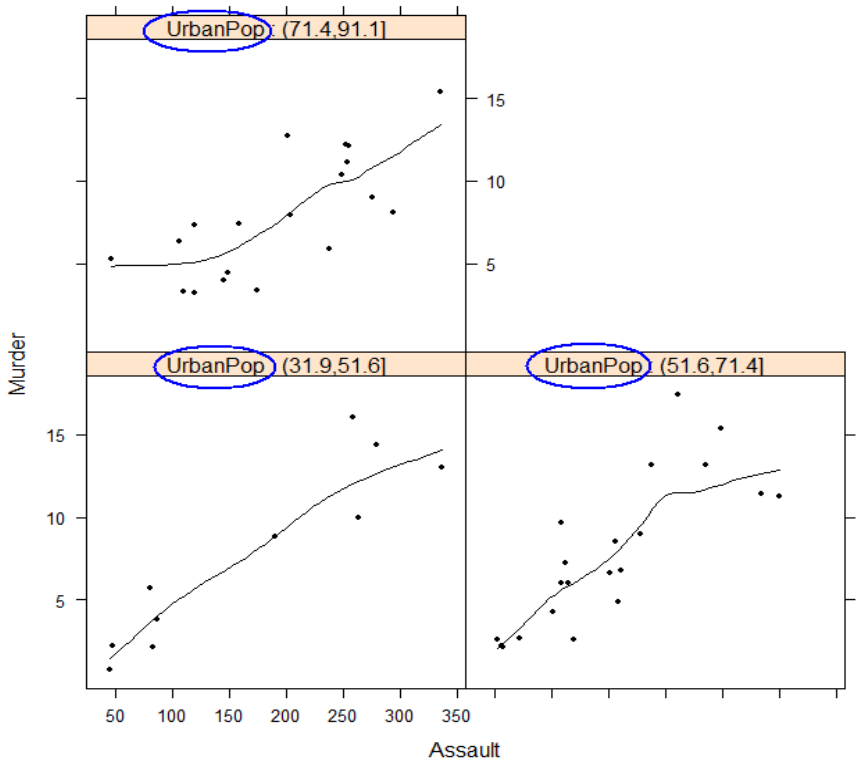
```
> xyplot(Murder ~ Assault | cut(UrbanPop, 3), type = c("p", "smooth"), pch = 20)
```



# 2. Gráficos

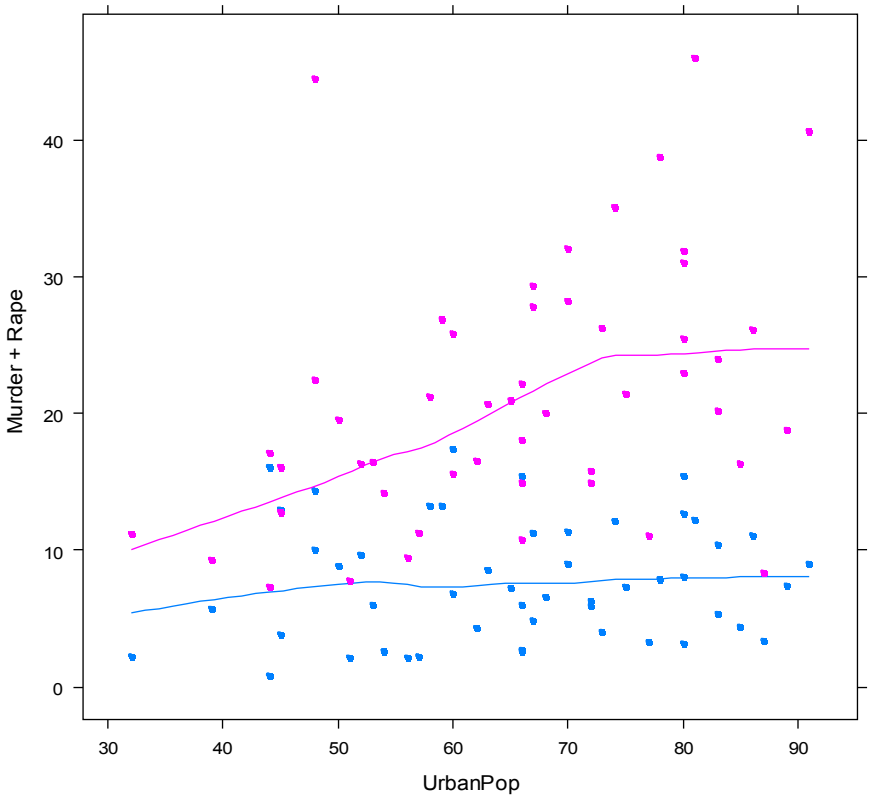
## Inclusão do nome da variável condicionante nos painéis

```
> xyplot(Murder ~ Assault | cut(UrbanPop, 3), type = c("p", "smooth"), pch = 20, strip.custom(strip.names = TRUE, var.name = "UrbanPop"))
```




## Duas variáveis dependentes, sem variável condicionante

```
> xyplot(Murder + Rape ~UrbanPop, type = c("p", "smooth"), pch =20)
```



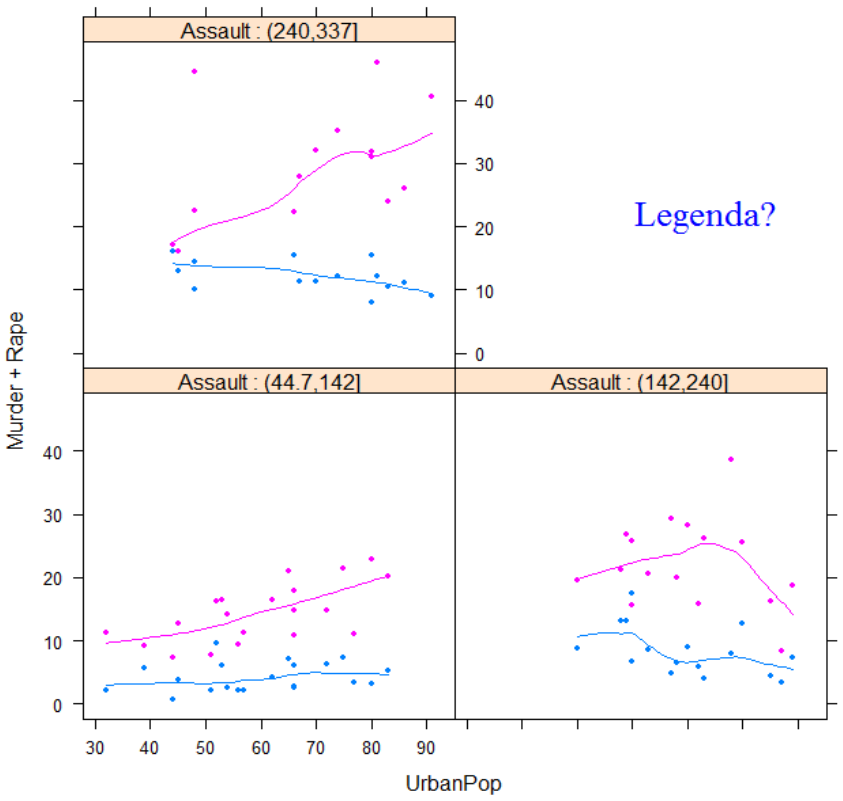

Obs. “+” não significa adição.

Exercício. Incluir uma legenda.

# 2. Gráficos

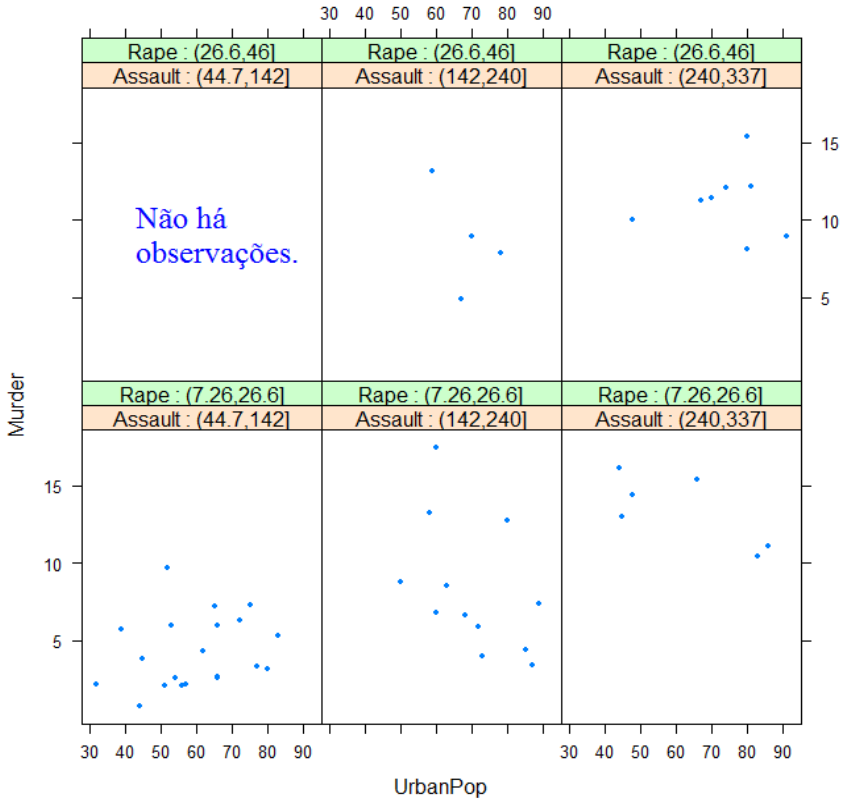
## Duas variáveis dependentes e uma variável condicionante

```
>  xyplot(Murder + Rape ~UrbanPop | cut(Assault, 3), type = c("p", "smooth"),   pch = 20, strip = strip.custom(strip.names = TRUE, var.name = "Assault"))
```



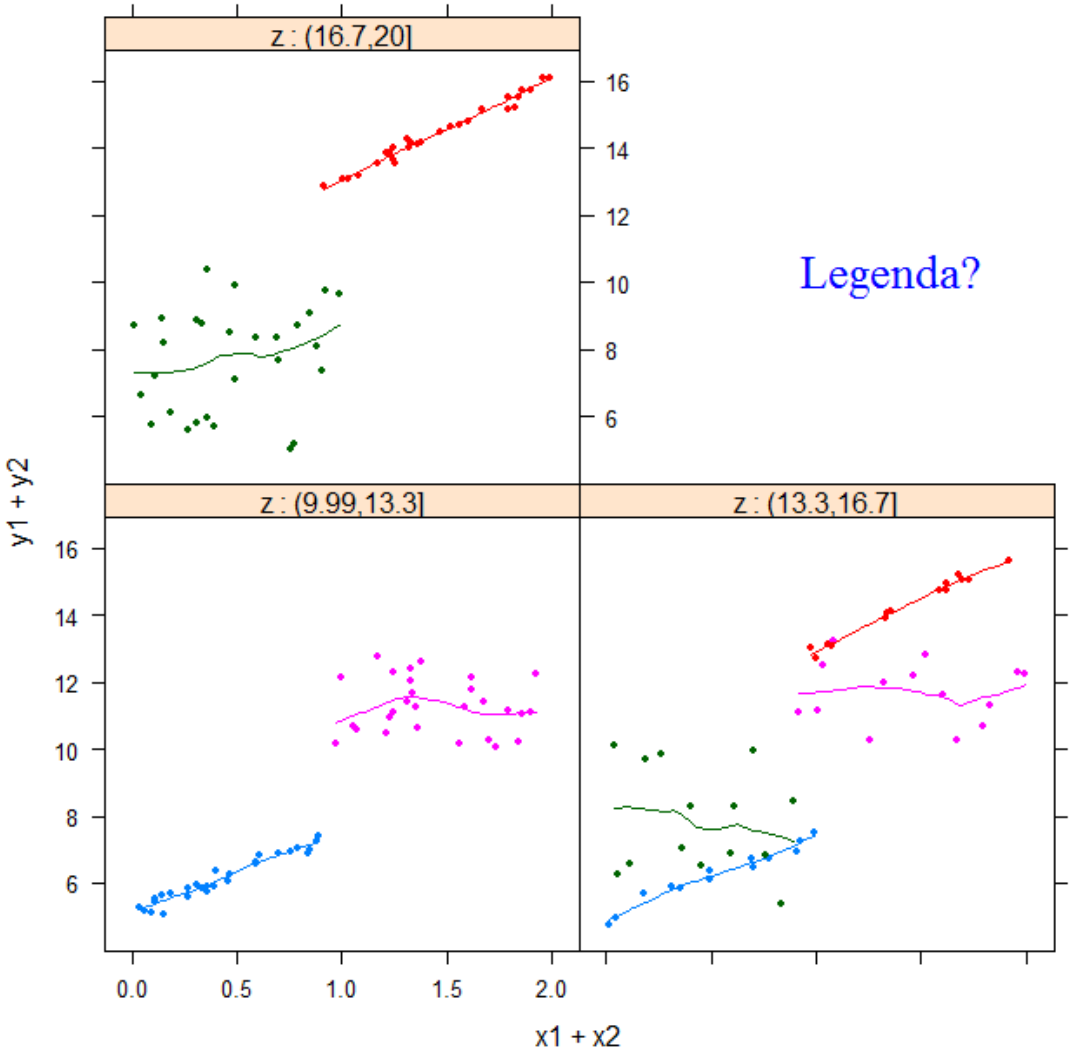

## Duas variáveis condicionantes

```
>  xyplot(Murder ~  UrbanPop | cut(Assault, 3) + cut(Rape, 2), pch = 20, strip = strip.custom(strip.names = TRUE, var.name = c("Assault", "Rape")))
```



# 2. Gráficos

Duas variáveis dependentes, duas variáveis independentes e uma variável condicionante (cinco variáveis)




Legenda?

Obs. (1) Quatro cores correspondem aos quatro pares de variáveis (x, y).

Neste exemplo, em cada painel podemos ter até quatro gráficos de dispersão.

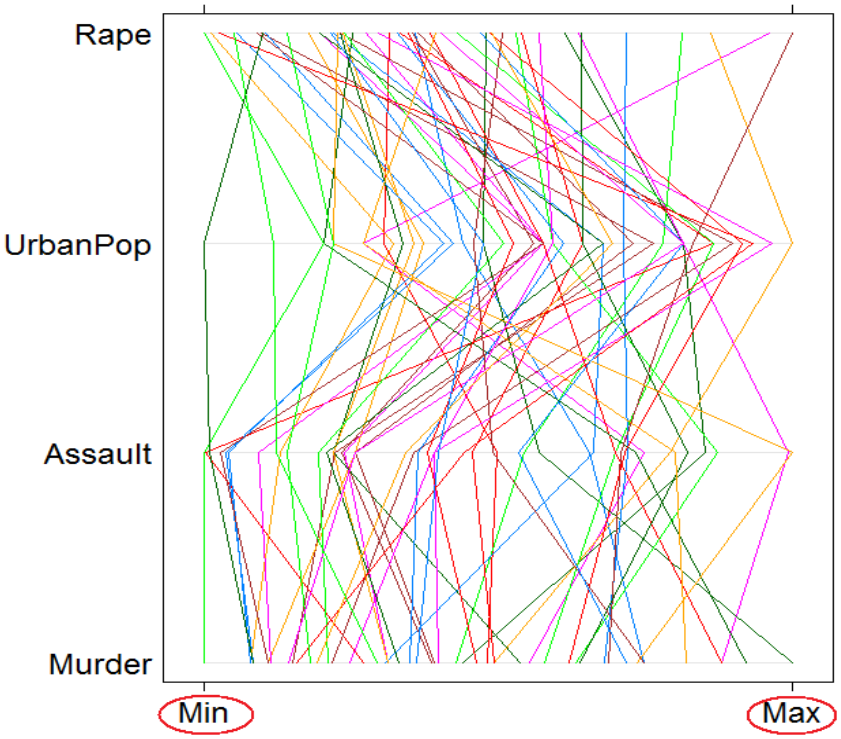
(2) Em uma fórmula, se quisermos somar variáveis (e se fizer sentido), utilizamos `I(x1 + x2)` e/ou `I(y1 + y2)`.

# 2. Gráficos

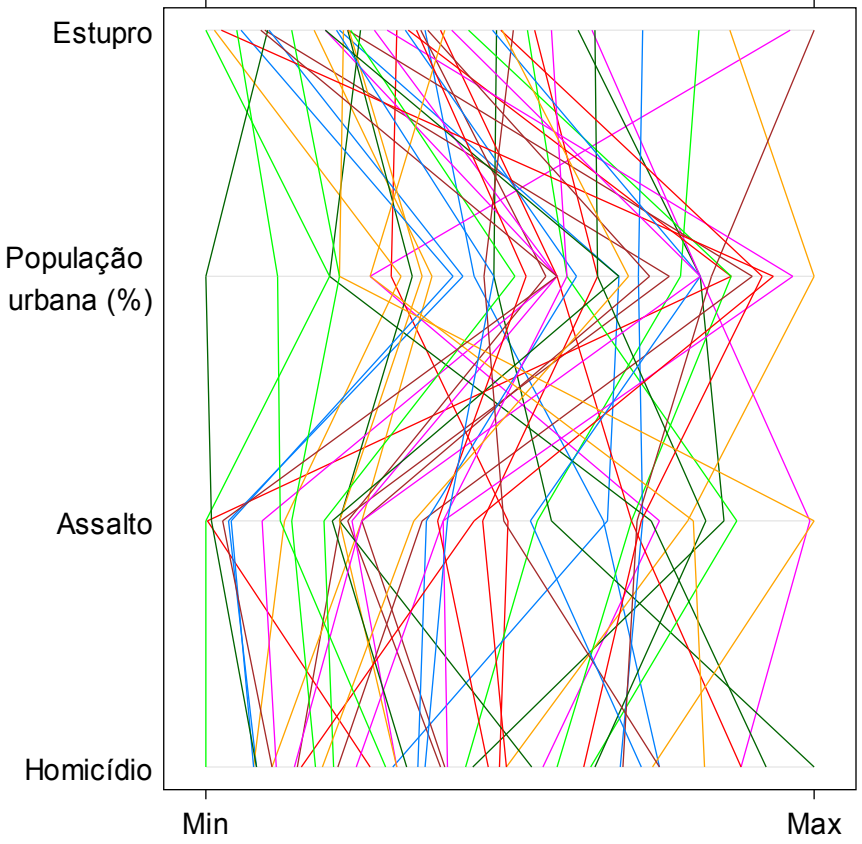
Função `parallel` (`lattice`): gráfico de coordenadas paralelas.

p – 1 segmentos de retas para cada observação unindo os valores escalonados em [Min, Max] para cada variável.

```
> parallel(USArrests)
```




```
> parallel(USArrests,
varnames = c("Homicídio",
"Assalto", "População \n
urbana (%)", "Estupro"))
```




Podem ser úteis para identificar grupos de observações (*cluster analysis*).

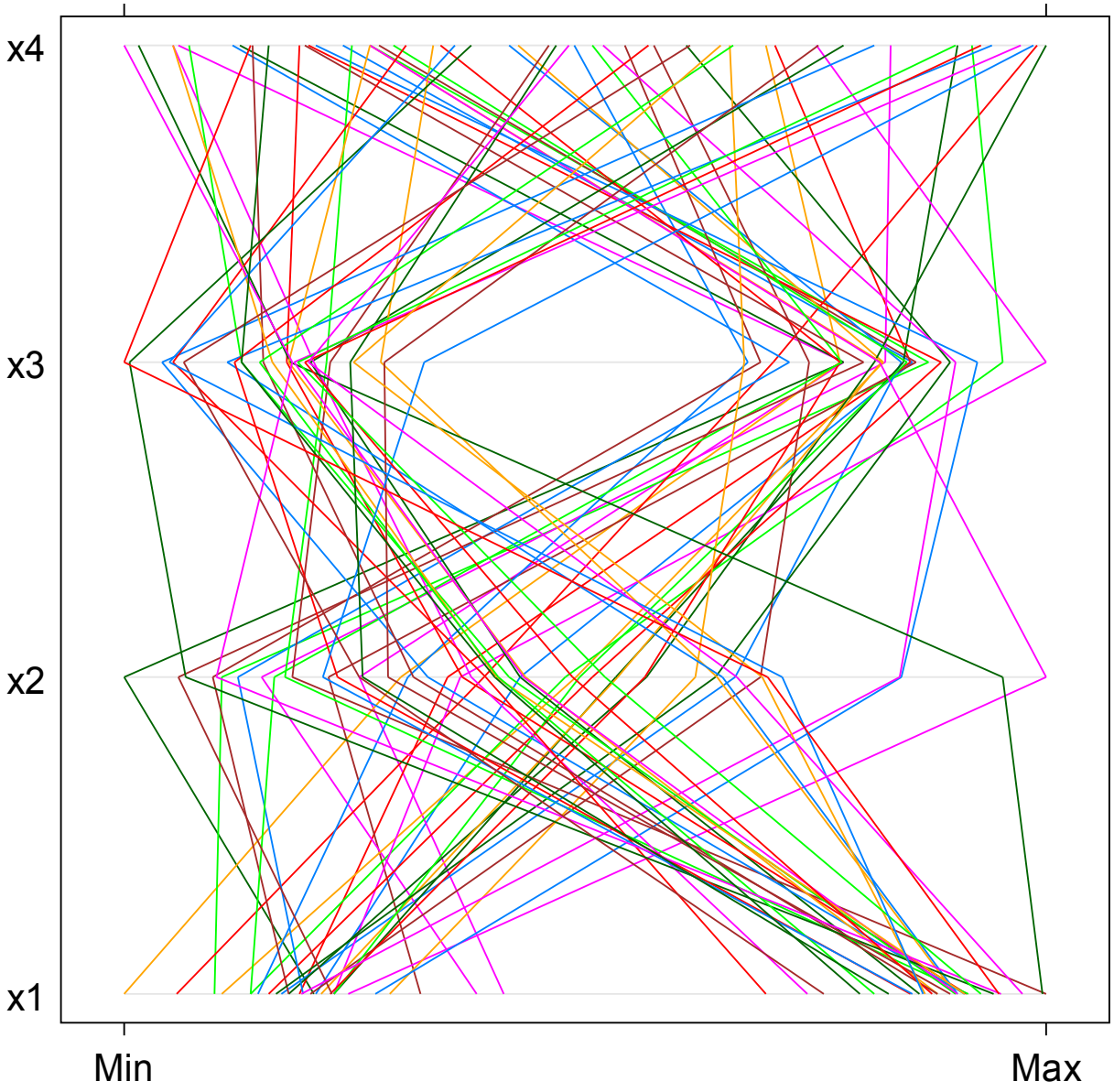
# 2. Gráficos

As variáveis $x_1$ e $x_3$ separam as observações em dois grupos.

Em um dos grupos os valores de $x_1$ são os menores e os valores de $x_3$ são os maiores.

No outro grupo há uma inversão.

As variáveis $x_2$ e $x_4$ não permitem uma separação tão nítida quanto $x_1$ e $x_3$.



# 2. Gráficos

```
> library(ineq)
> ?Ilocos
```

Ilocos {ineq} R Documentation

**Income Metadata from Ilocos, Philippines**

**Description**

Income metadata from surveys conducted by the Philippines' National Statistics Office.

**Usage**

data(Ilocos)

Dados coletados em domicílios nas Filipinas.

```
> data(Ilocos)
> dados = Ilocos
> dim(dados)  [1] 632   8
> names(dados)
"income" "sex" "family.size" "urbanity" "province" "AP.income"
"AP.family.size" "AP.weight"
```

n = 632 observações de 8 variáveis.

```
> summary(dados[, c("sex", "urbanity", "province")])
 sex           urbanity          province
 female:114    rural:301    Ilocos Norte: 65
 male  :518    urban:331    Ilocos Sur  : 68
                            La Union    :116
                            Pangasinan  :383
```

```
> class(dados$province)
[1] "factor"
```

Variável qualitativa: fator (*factor*).

# 2. Gráficos

Dados `Ilocos`.

```
> names(dados)
```

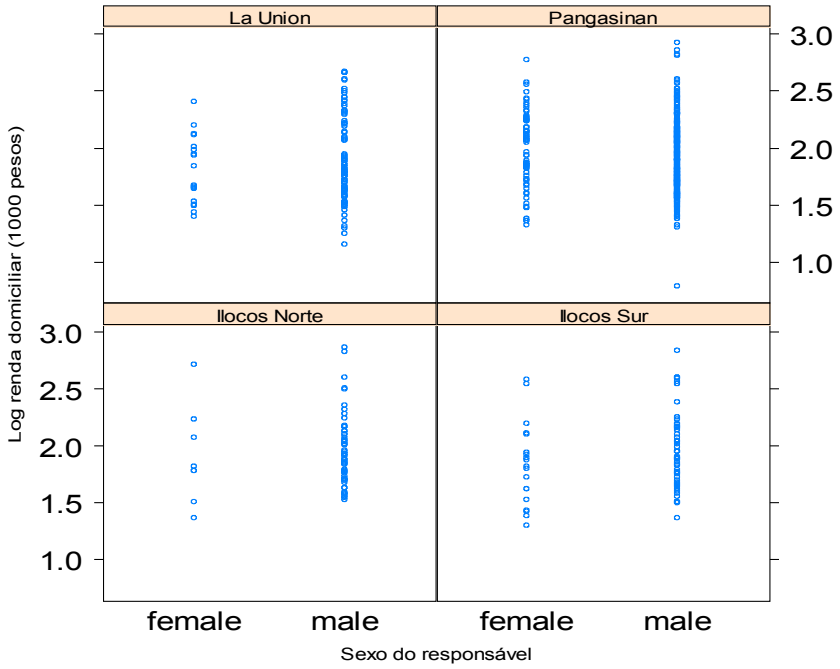
y x y x x

```
"income" "sex" "family.size" "urbanity" "province" "AP.income"
"AP.family.size" "AP.weight"
```

## Gráfico de pontos

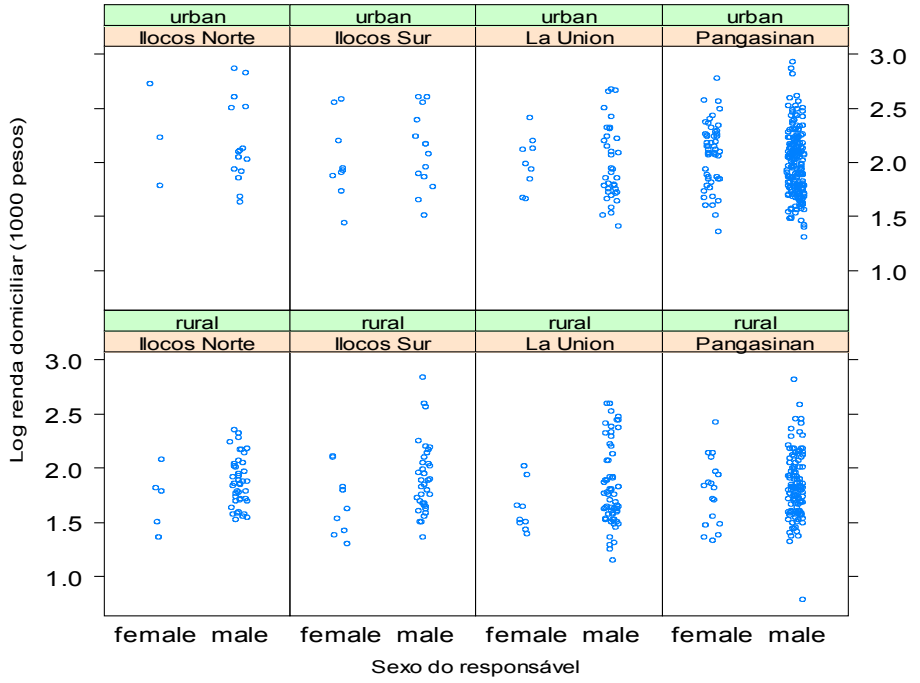
## Função `stripplot` (`lattice`)

```
> stripplot(log(income / 1000, 10) ~ sex | province, xlab = "Sexo do responsável", ylab = "Log renda domiciliar (1000 pesos)")
```




## Duas variáveis condicionantes e acréscimo de ruído

```
> stripplot(log(income / 1000, 10) ~ sex | province + urbanity, xlab = "Sexo do responsável", ylab = "Log renda domiciliar (1000 pesos)", jitter.data = TRUE)
```

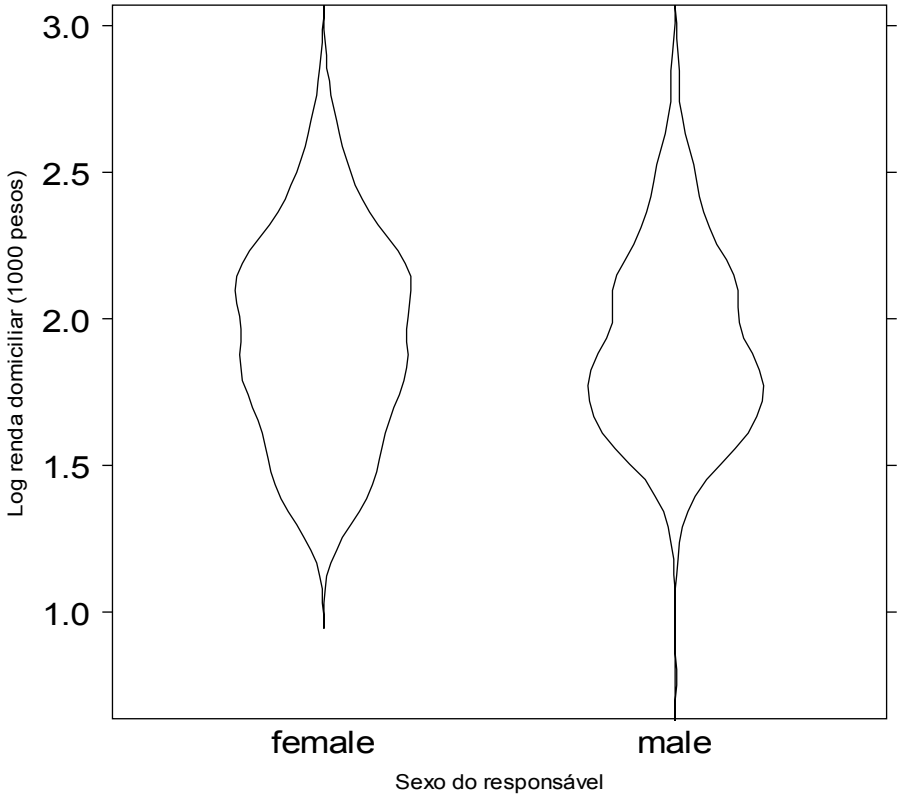


# 2. Gráficos

## Gráfico de violino

### Função `bwplot` (`lattice`)

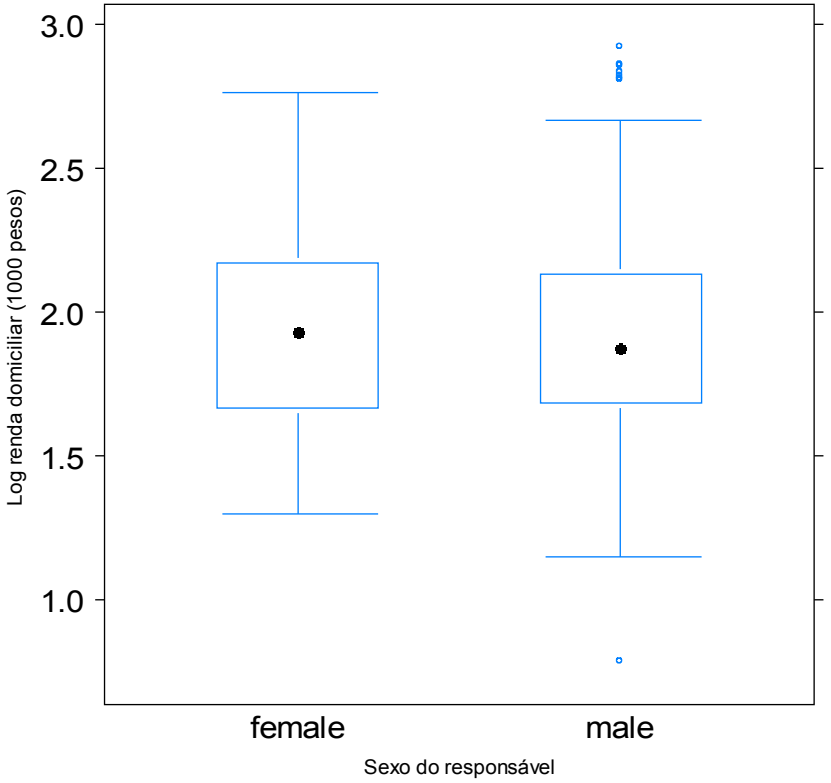
```
> bwplot(log(income / 1000, 10) ~ sex, panel = panel.violin, xlab = "Sexo do responsável", ylab = "Log renda domiciliar (1000 pesos)", col = "white")
```




## Gráfico de caixas

### Função `bwplot` (`lattice`)

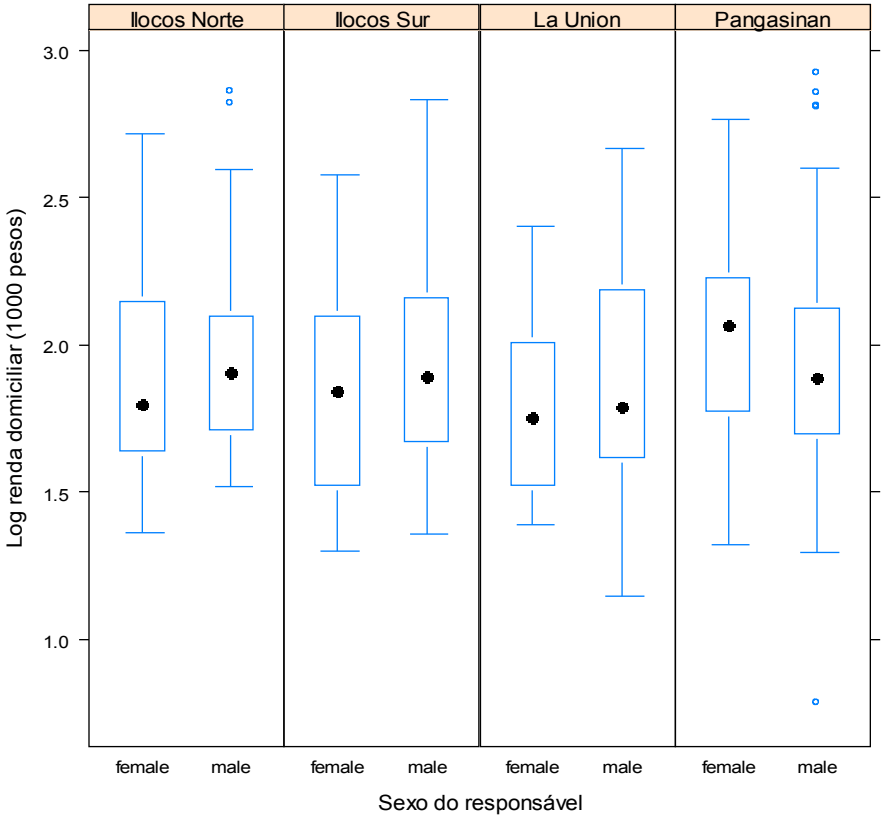
```
> bwplot(log(income / 1000, 10) ~ sex, xlab = "Sexo do responsável", ylab = "Log renda domiciliar (1000 pesos)")
```



# 2. Gráficos

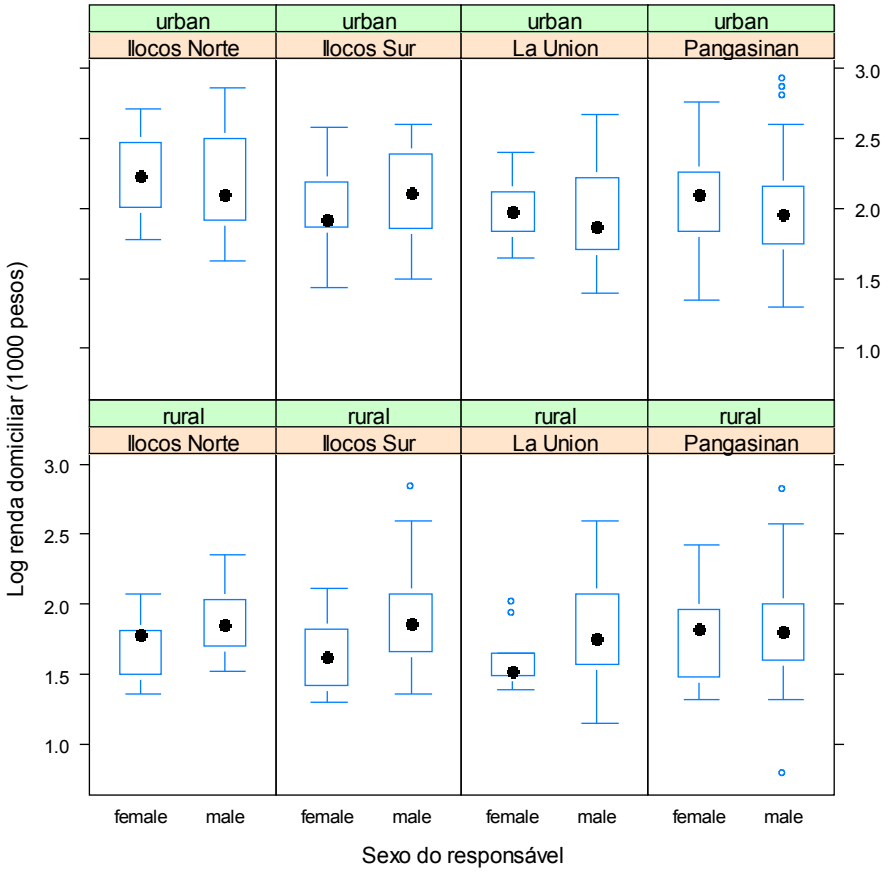
## Uma variável condicionante

```
> bwplot(log(income / 1000, 10)
~ sex | province, xlab = "Sexo
do responsável",   ylab = "Log
renda domiciliar (1000 pesos)",
layout = c(4, 1))
```




## Duas variáveis condicionantes

```
> bwplot(log(income / 1000, 10)
~ sex | province + urbanity,
xlab = "Sexo do responsável",
ylab = "Log renda domiciliar
(1000 pesos)")
```

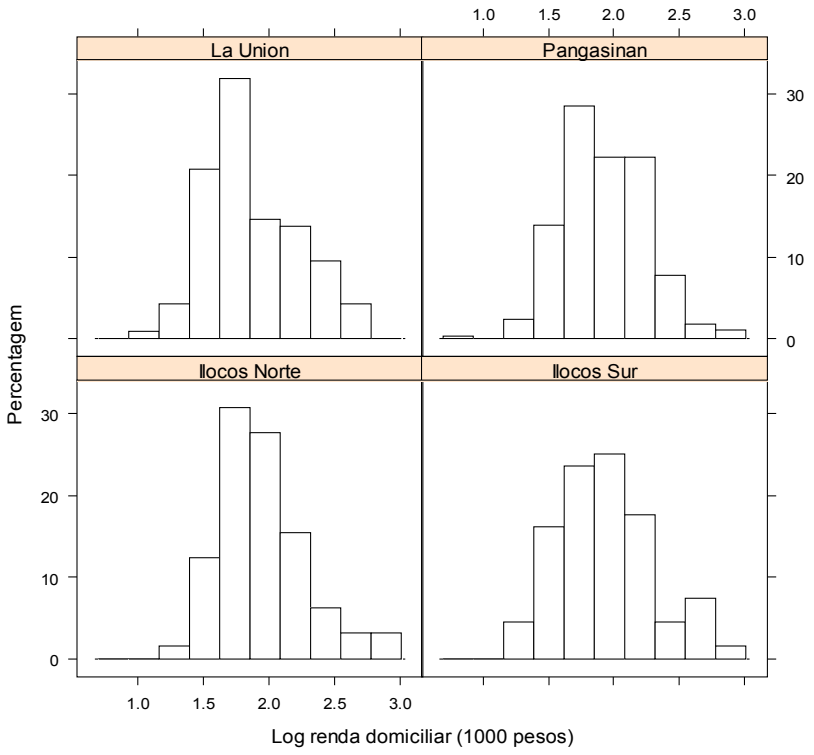


# 2. Gráficos

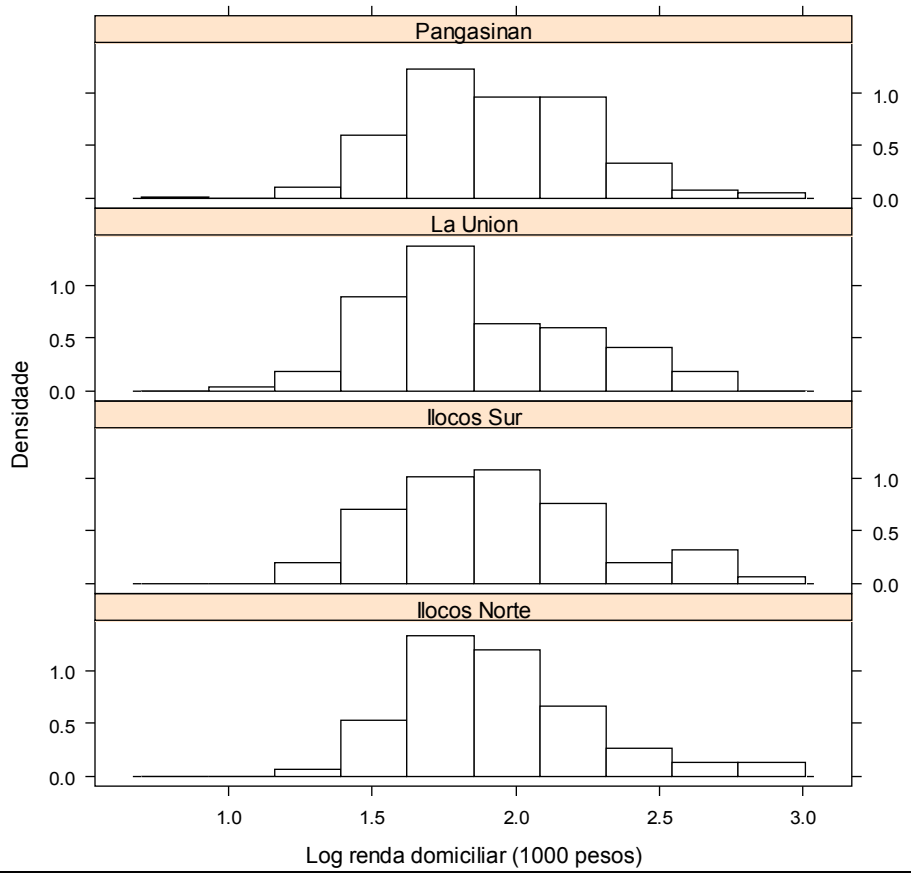
## Histograma

### Função `histogram` (`lattice`)

```
> histogram(~ log(income / 1000, 10) | province, type = "percent", ylab = "Percentagem", xlab = "Log renda domiciliar (1000 pesos)", col = "white")
```



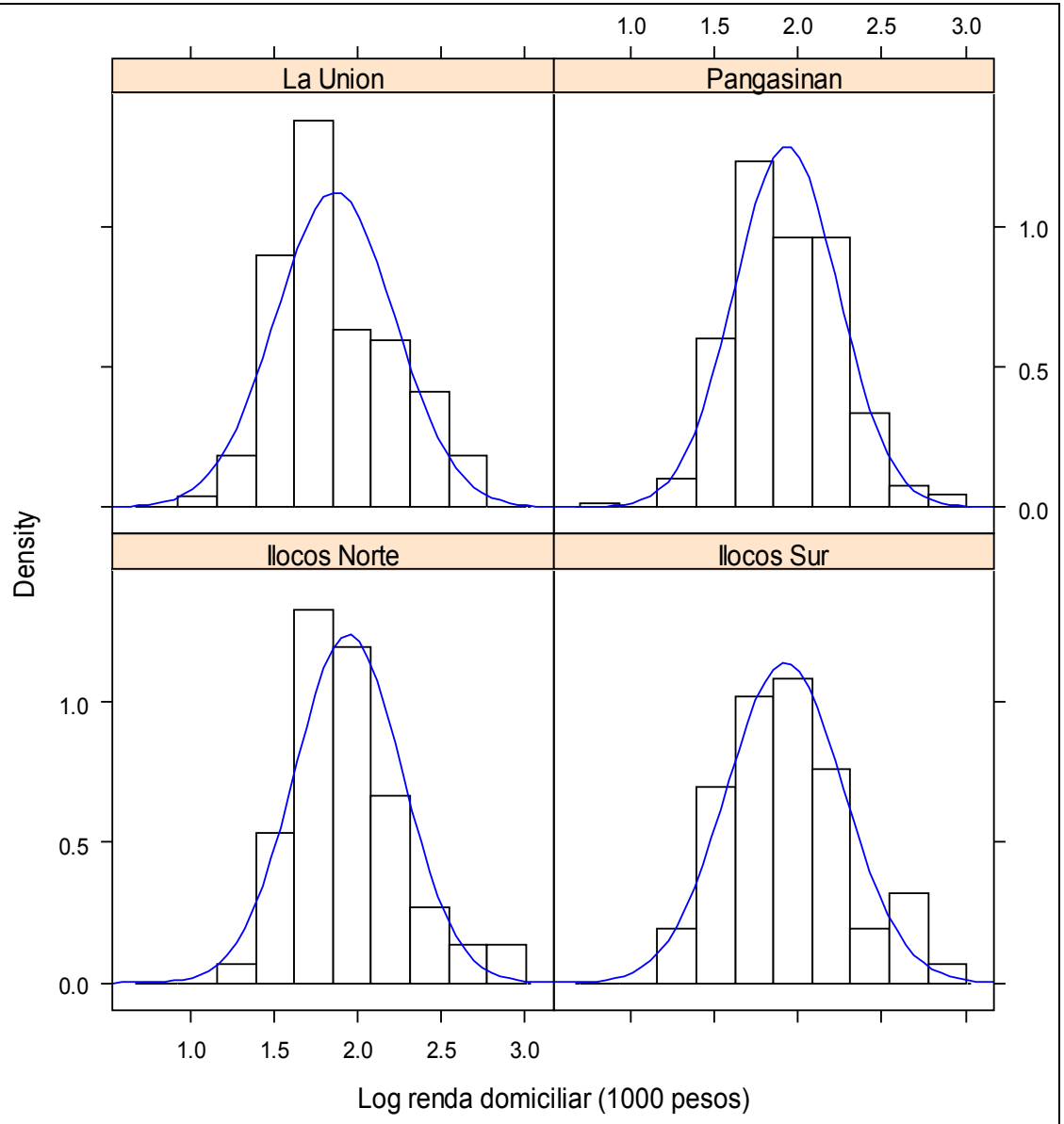

```
> histogram(~ log(income / 1000, 10) | province, type = "density", layout = c(1, length(levels(province))), ylab = "Densidade", xlab = "Log renda domiciliar (1000 pesos)", col = "white")
```



# 2. Gráficos

## Histograma e função densidade normal

```
> histogram(~ log(income / 1000, 10) | province, type = "density",
 ylab = "Densidade", xlab = "Log renda domiciliar (1000 pesos)", col = "white",
 panel = function(x, ...) {    panel.histogram(x, ...) panel.mathdensity(dmath = dnorm, col = "blue", args = list(mean  = mean(x),sd = sd(x))) })
```




## Exercícios.

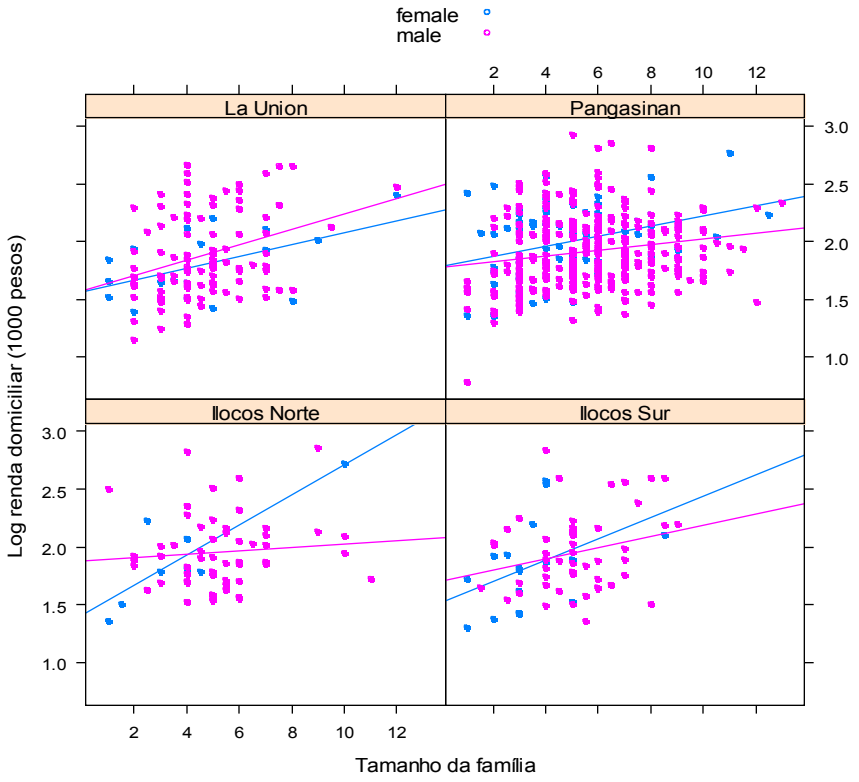
1. Substituir a função densidade normal pela densidade estimada.

2. Incluir os pontos no eixo horizontal.

# 2. Gráficos

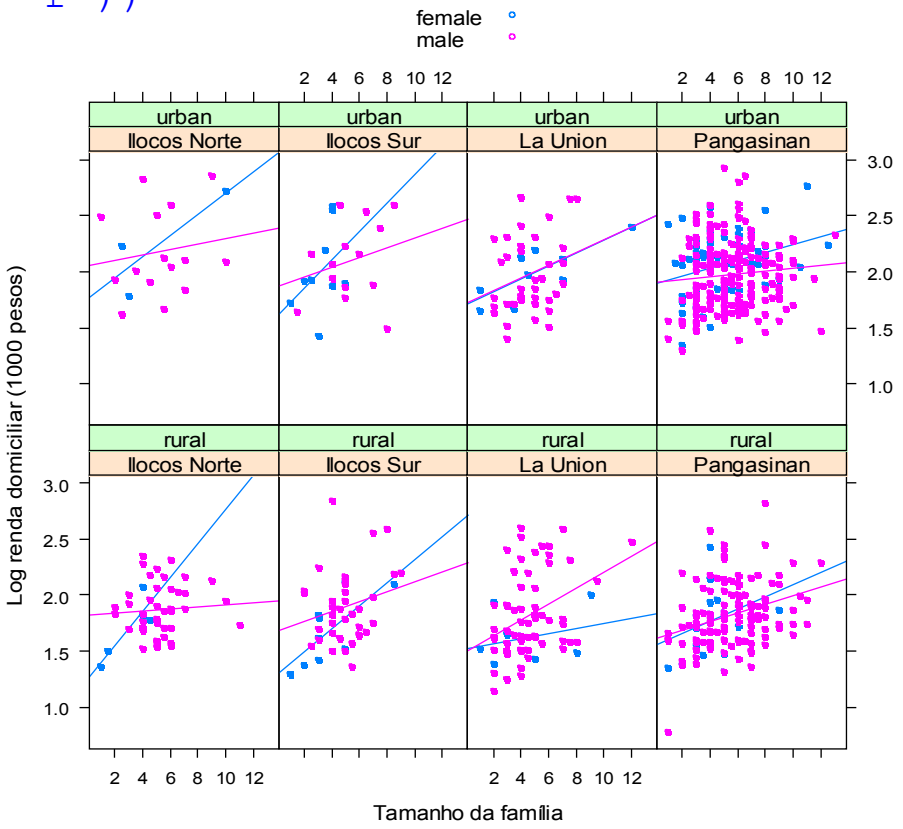
## Grupos de acordo com a variável `sex`

```
> xyplot(log(income / 1000, 10) ~ family.size | province, group = sex,  auto.key = TRUE, xlab = "Tamanho da família",  ylab = "Log renda domiciliar (1000 pesos)", pch = 20, type = c("p", "r"))
```




## Duas variáveis condicionantes

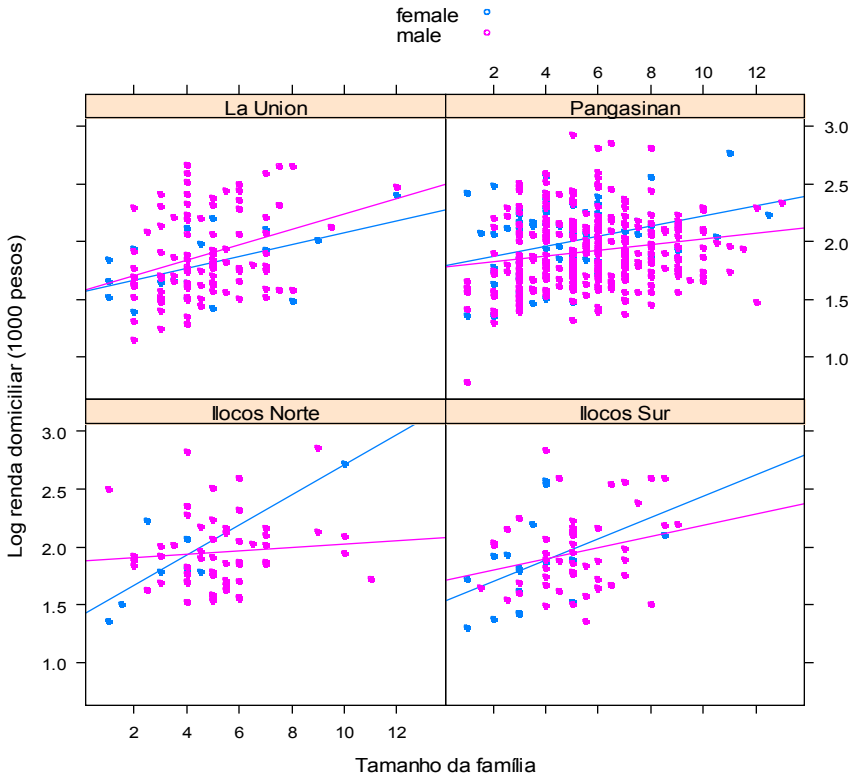
```
> xyplot(log(income / 1000, 10) ~ family.size | province + urbanity, group = sex,  auto.key = TRUE, xlab = "Tamanho da família",  ylab = "Log renda domiciliar (1000 pesos)", pch = 20, type = c("p", "r"))
```



# 2. Gráficos

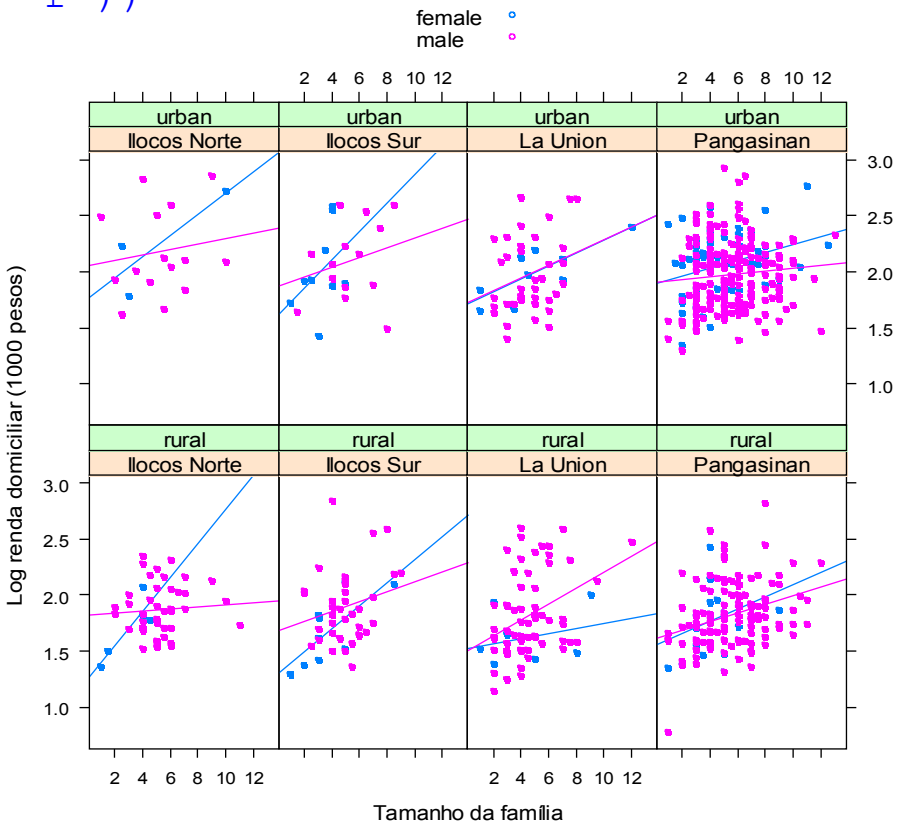
## Grupos de acordo com a variável `sex`

```
> xyplot(log(income / 1000, 10) ~ family.size | province, group = sex,  auto.key = TRUE, xlab = "Tamanho da família",  ylab = "Log renda domiciliar (1000 pesos)", pch = 20, type = c("p", "r"))
```



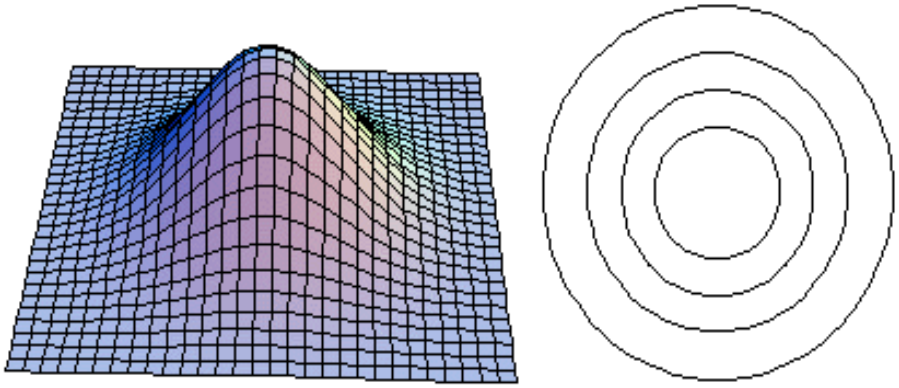

## Duas variáveis condicionantes

```
> xyplot(log(income / 1000, 10) ~ family.size | province + urbanity, group = sex,  auto.key = TRUE, xlab = "Tamanho da família",  ylab = "Log renda domiciliar (1000 pesos)", pch = 20, type = c("p", "r"))
```

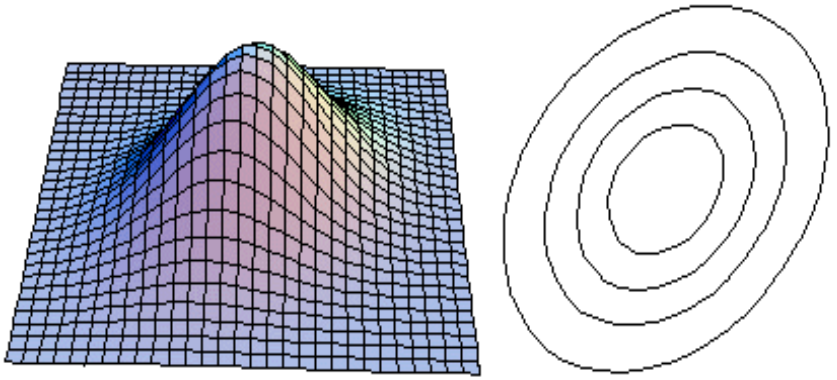


# 3.1. Distribuição normal bivariada

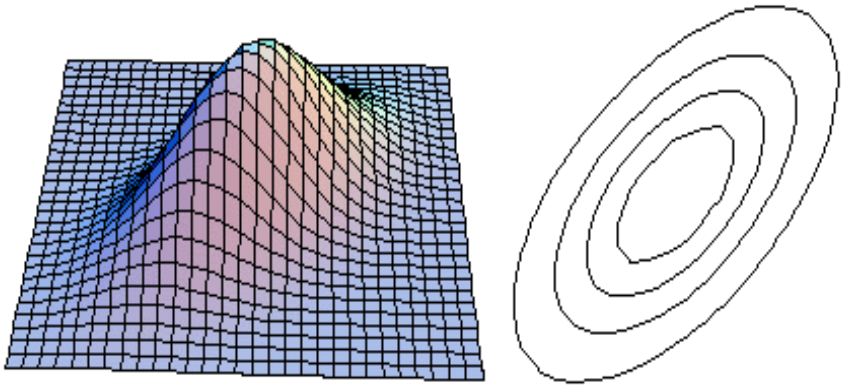
Correlação = 0 (independência)



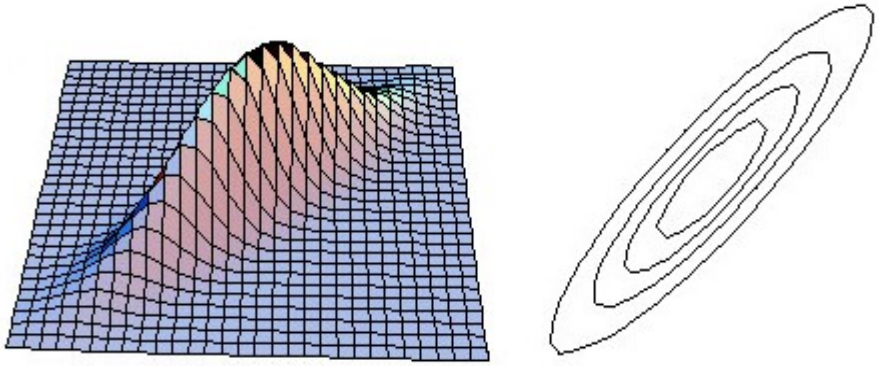
Correlação = 0,3



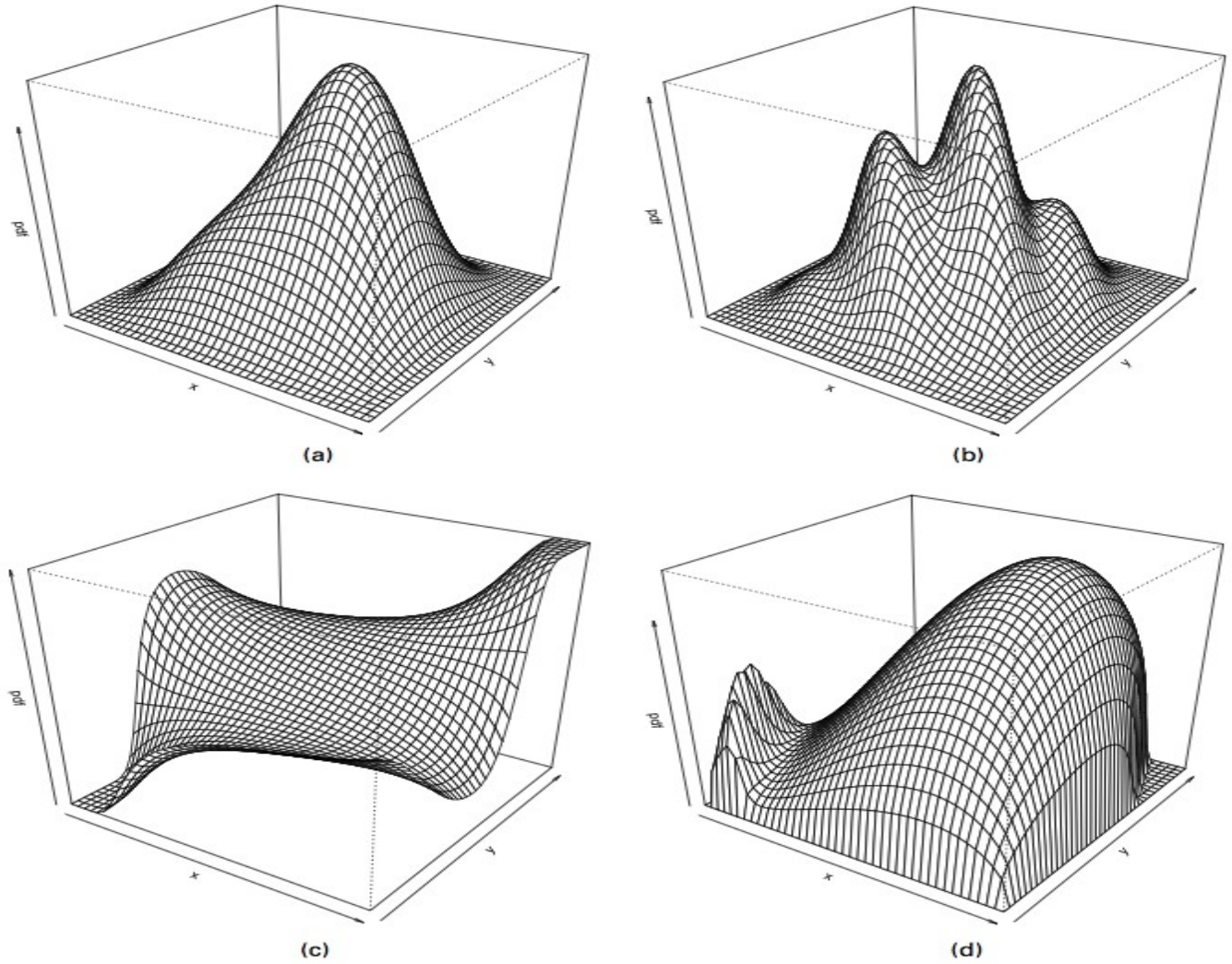
Correlação = 0,6



Correlação = 0,9



# 3.2. Distribuições bivariadas assimétricas

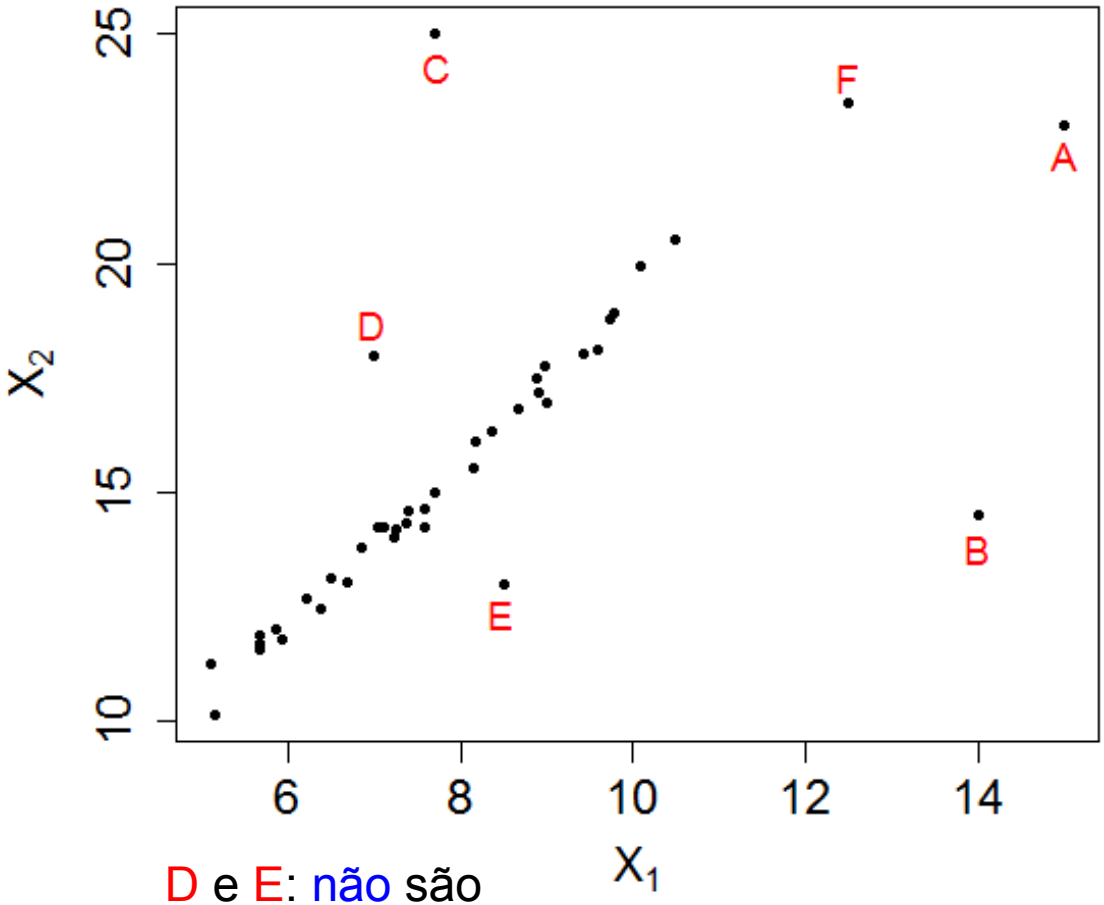



Distribuição normal assimétrica (*skew normal*):
http://azzalini.stat.unipd.it/SN/

# 4. Pontos aberrantes

C: aberrante para $X_2$.

F: aberrante unidimensional para $X_1$ e $X_2$.




A: aberrante unidimensional para $X_1$ e $X_2$.

B: aberrante para $X_1$.

D e E: não são aberrantes unidimensionais.

A, B, C, D, E e F: aberrantes bidimensionais.