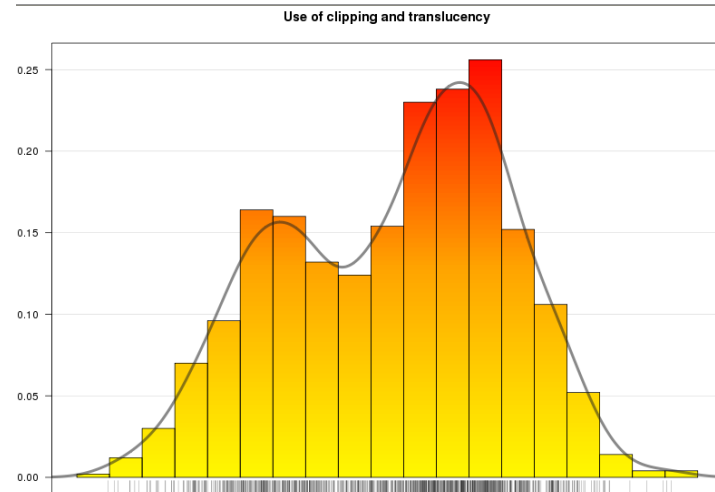
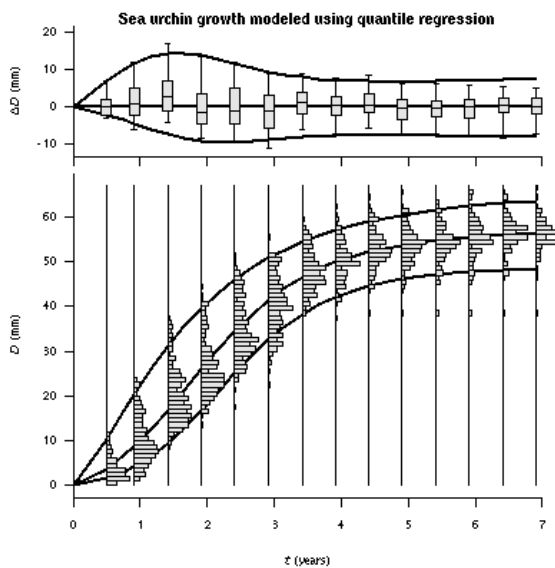
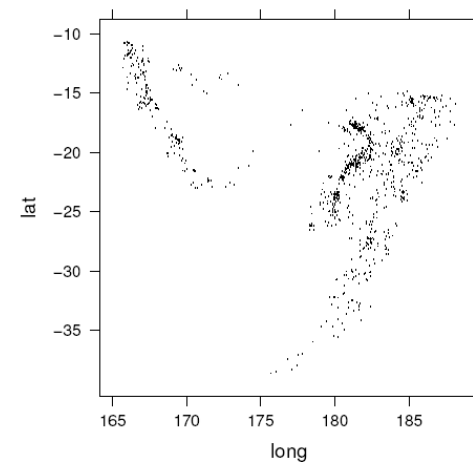
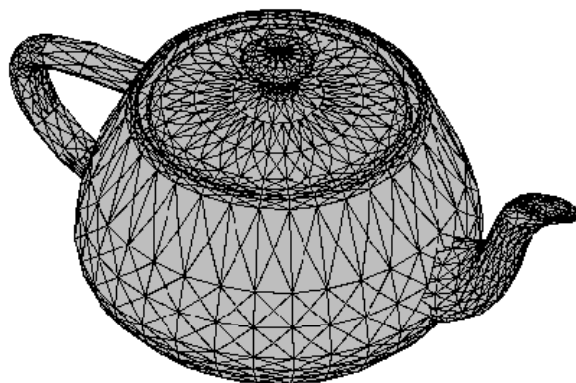
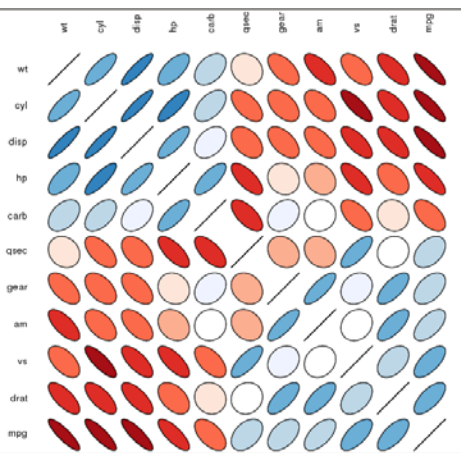


Criando e adaptando gráficos



1º passos – Noções básicas

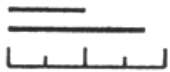
Tabela vs. gráfico

- ✓ Gráfico é melhor para **mostrar rapidamente** relações entre variáveis e dados multidimensionais de forma compreensível.
- ✓ Tabela mostra melhor valores exatos

Princípios básicos da apresentação gráfica:

- ✓ Ressaltar os padrões de interesse;
- ✓ Manter a estrutura dos dados, de forma que o leitor possa reconstruir os dados a partir da figura;
- ✓ A figura deve ter uma razão **dado:tinta** alta;
- ✓ As figuras **não** devem distorcer, exagerar ou apagar os dados.

BETTER



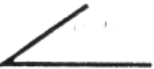
1. Position along a common scale



2. Position along identical scales



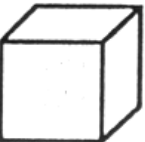
3. Length



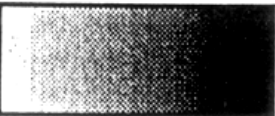
4. Angle/Slope



5. Area



6. Volume



7. Shading: color, saturation, density

WORSE



William S. Cleveland

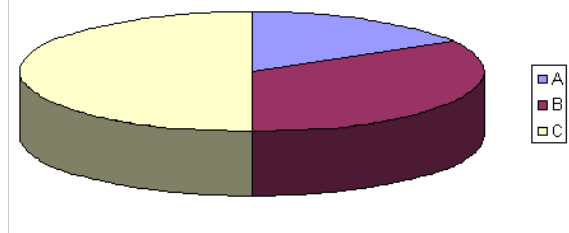
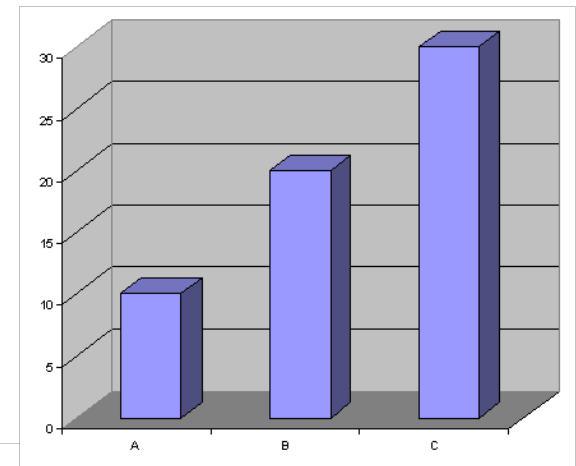
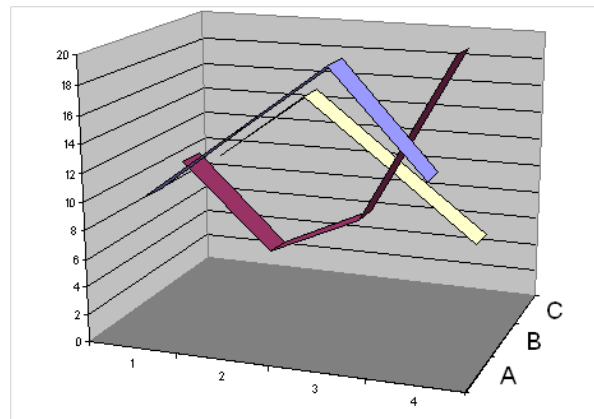
Mais dicas sobre apresentação gráfica:

- 1. linhas não podem obscurecer os dados**
- 2. dados não devem cair sobre os eixos**
- 3. evitar linhas de grade**
- 4. dados sobrepostos devem ser claramente distinguíveis (diminuir o tamanho dos pontos, separar os dados em gráficos diferentes, agitar os pontos etc)**
- 5. o gráfico deve ser legível após redução na publicação ou em projeções em seminários**

Diagramas 3D

- ✓ Usar gráficos tridimensionais **APENAS** se tiver 3 variáveis
- ✓ Nunca usar em trabalhos científicos:

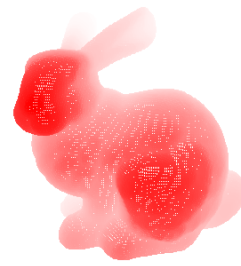
1. Barras com volume
2. Pizza com volume
3. Tiras para série temporais
4. Etc



Criando gráficos no R

- Algumas funções **criam** um gráfico (plot, barplot, boxplot, etc)
- Algumas funções **complementam** um gráfico já existente (text, lines, abline, etc)
- Algumas funções **apenas** podem ser chamadas pela função **par** (mfrow, mar, etc)
- Algumas funções **jamais** podem ser chamadas pela função **par** (axis, sqp, etc)
- A maioria das funções **podem ou não** ser chamadas pela função **par** (cex, family, etc)
- Algumas funções tem resultados diferentes se chamadas pela função **par**
ou no **plot**(cex, bty, etc)

2º passo – Criando gráficos **simples**



Variações sobre mesmo tema

```
plot(x=area ,y=riqueza, ...)
```

```
plot(area,riqueza, ...)
```

```
plot(riqueza~area, ...)
```

```
boxplot(riqueza~area,...)
```

```
plot(riqueza~area,...)
```

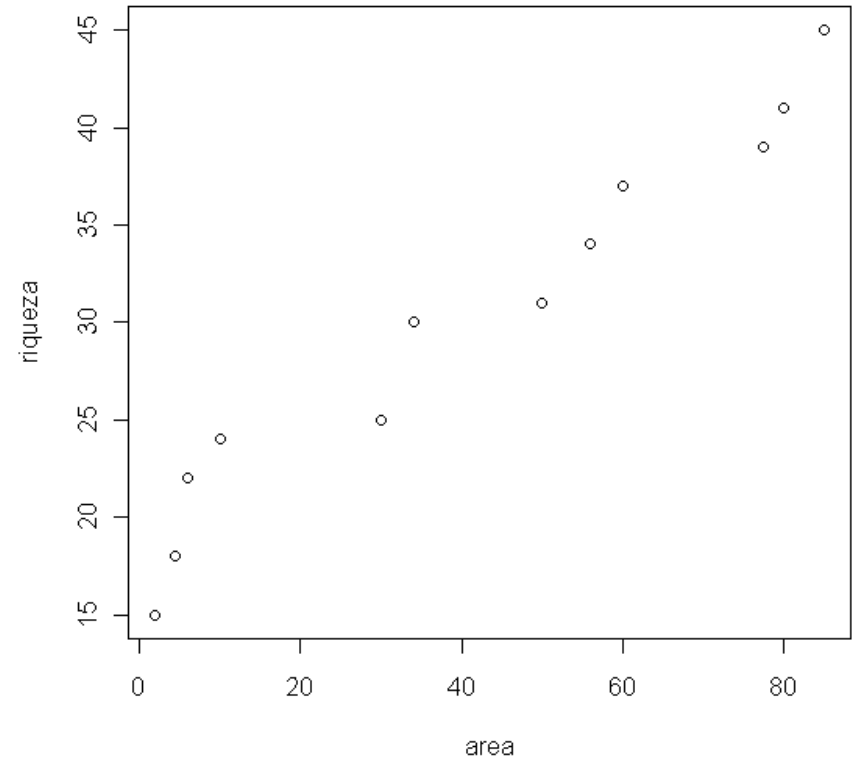
para os complicados ver
<http://addictedtor.free.fr/graphiques/>

algumas funções diferentes (notch, lwd)

se área for uma variável categórica

Criando um plot

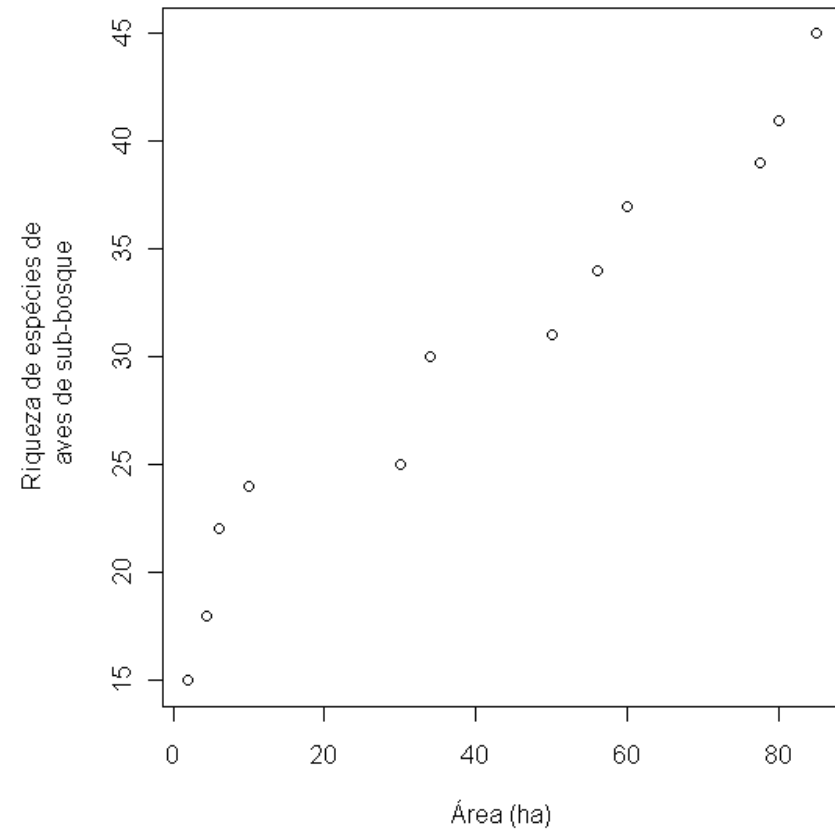
```
plot(riqueza~area)
```



```
riqueza <- c(15,18,22,24,25,30,31,34,37,39,41,45)
```

```
area <- c(2,4.5,6,10,30,34,50,56,60,77.5,80,85)
```

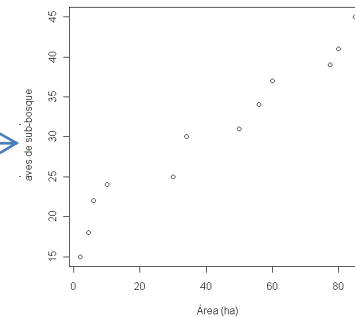

Mudando labels e margens



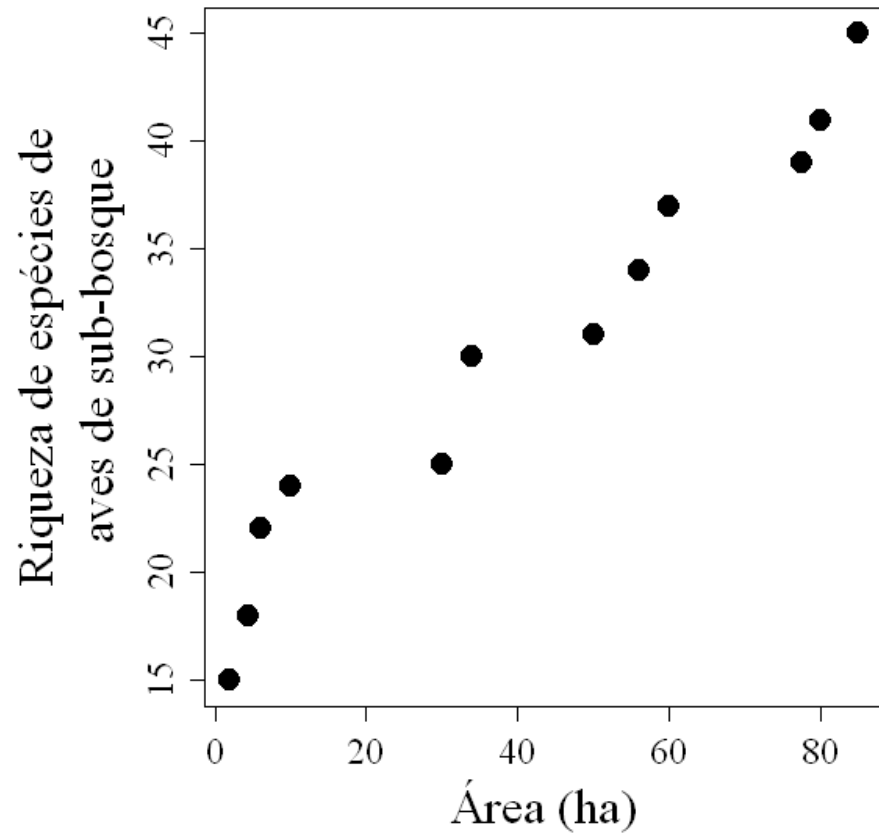
```
par(mar=c(5,6.7,2,1))
```

```
plot(riqueza~area, xlab="Área (ha)", ylab="Riqueza de espécies de\naves de sub-bosque")
```

Não se ve o label inteiro
mudar função **mar**



Mudando fontes



```
par(mar=c(5,6.7,2,1))
```

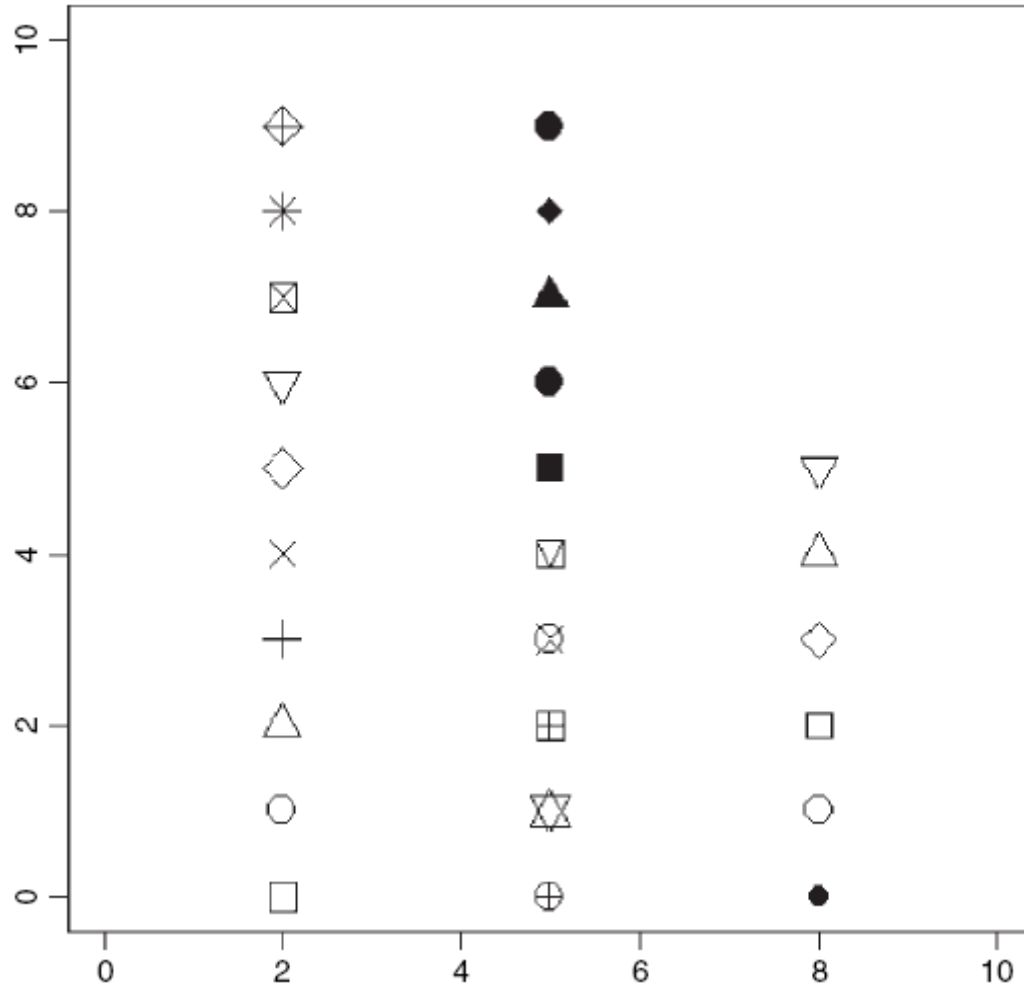
```
par(cex.axis=1.5)
```

```
par(cex.lab=2)
```

```
par(family="serif")
```

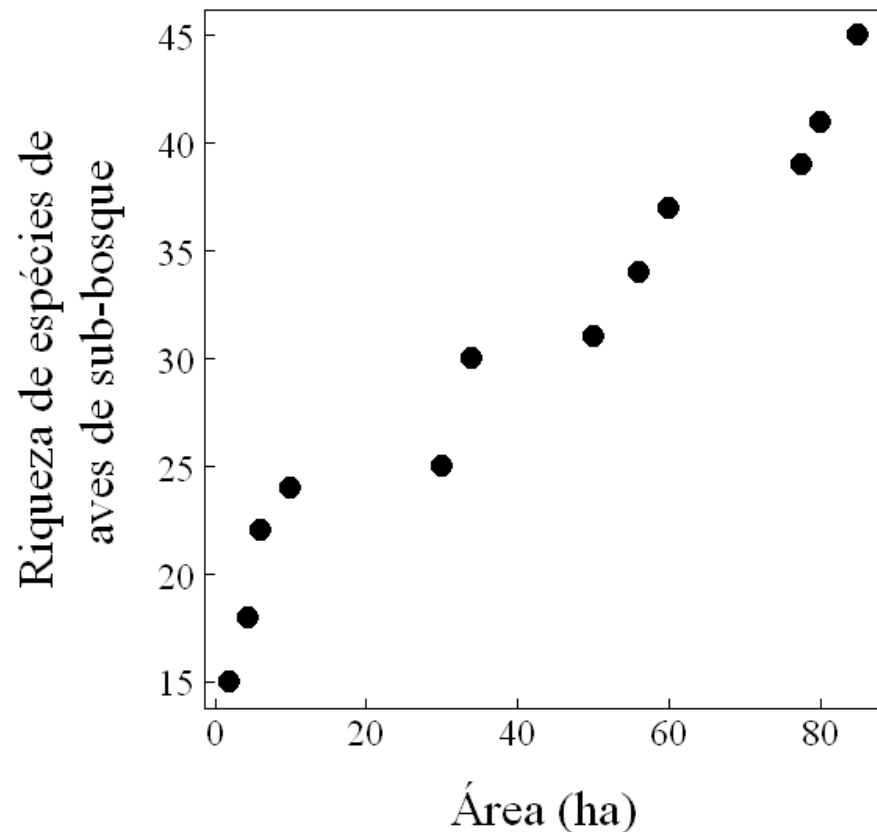
```
plot(riqueza~area, xlab="Área (ha)", ylab="Riqueza de espécies de  
aves de sub-bosque",  
cex=2, pch=16)
```

Tipo de fontes - pch

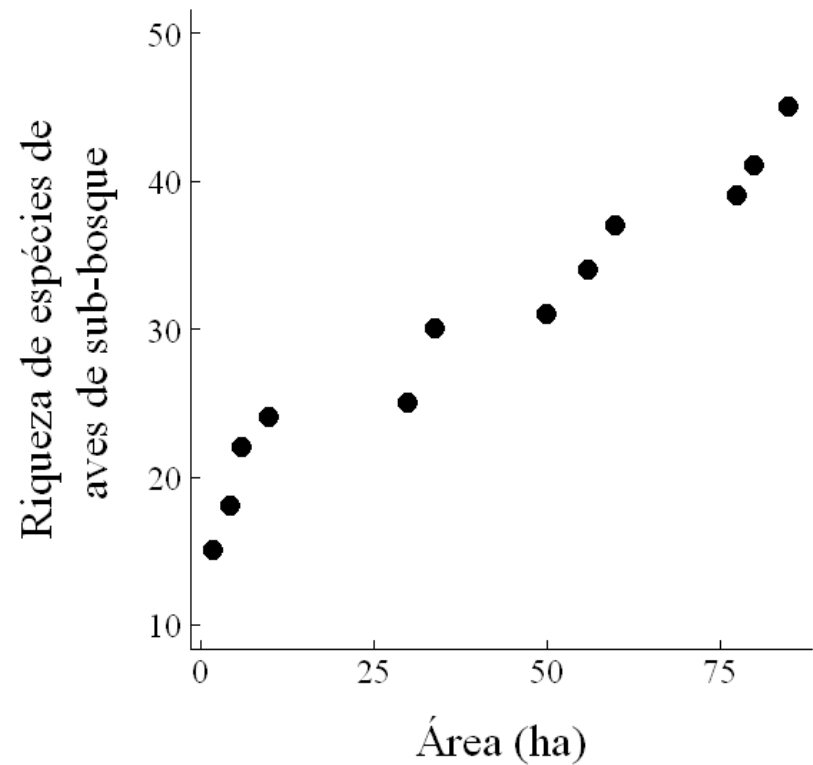


Mudando eixos - 1

```
par(mar=c(5,6.7,2,1))  
par(cex.axis=1.5)  
par(cex.lab=2)  
par(family="serif")  
par(las=1)  
par(tcl=0.3)  
par(mgp=c(3,0.3,0))  
plot(riqueza~area, xlab="Área (ha)", ylab="Riqueza de espécies de  
aves de sub-bosque",  
cex=2, pch=16)
```



Mudando eixos - 2



```
par(mar=c(5,6.7,2,1))
par(cex.axis=1.5)
par(cex.lab=2)
par(family="serif")
par(las=1)
par(tcl=0.3)
par(mgp=c(3,0.3,0))
plot(riqueza~area, xlab="Área (ha)", ylab="Riqueza de espécies de\naves de sub-bosque",
cex=2, pch=16, bty="l", xaxp=c(0,100,4), ylim= c(10,50))
```

→ não pode ir no par

Inserindo + info

```
z <- c(50,42,33,29,25,19,17,15,10,11,8,9)
```

```
par(mar=c(5,6.7,2,4))
```

```
par(cex.axis=1.5)
```

```
par(cex.lab=2)
```

```
par(family="serif")
```

```
par(las=1)
```

```
par(tcl=0.3)
```

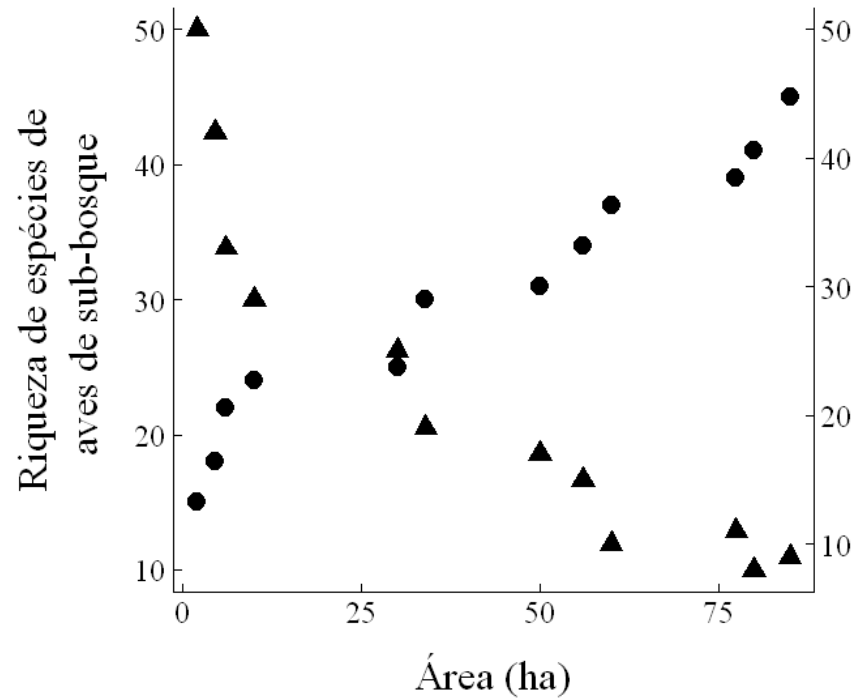
```
par(mgp=c(3,0.3,0))
```

```
plot(riqueza~area, xlab="Área (ha)", ylab="Riqueza de espécies de  
aves de sub-bosque",  
cex=2, pch=16, bty="u", xaxp=c(0,100,4), ylim= c(10,50))
```

```
par(new=TRUE)
```

```
plot(z~area, ann=F, axes=F, cex=2, pch=17)
```

```
axis(4)
```



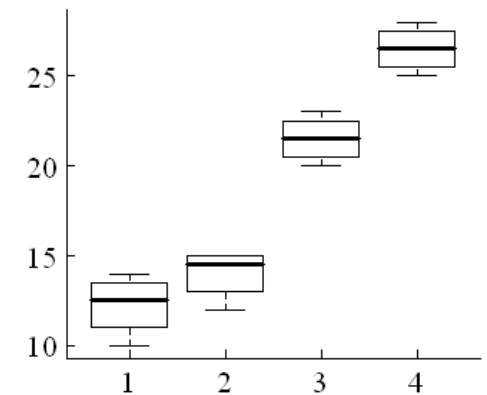
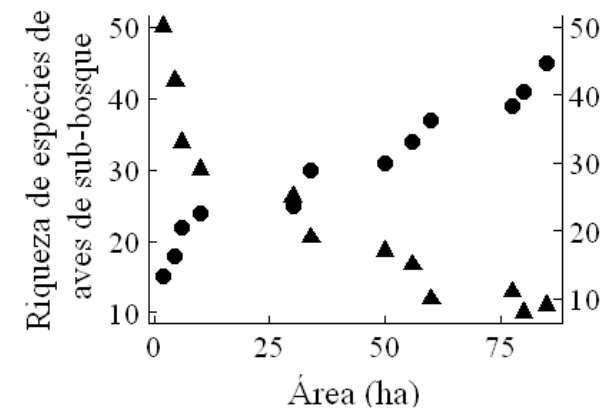
Inserindo ++ info

```
box <- c(10,13,12,14,15,12,14,15,20,23,22,21,26,27,28,25)
samples <- rep(1:4,each=4)
```

```
par(mfrow=c(2,1))
par(mar=c(3,5,2,4))
par(cex.axis=1.3)
par(cex.lab=1.5)
par(family="serif")
par(las=1)
par(tcl=0.3)
par(mgp=c(2,0.3,0))
plot(riqueza~area, xlab="Área (ha)", ylab="Riqueza de espécies de
aves de sub-bosque", cex=1.5,
pch=16, xaxp=c(0,100,4), ylim= c(10,50), bty="u")
```

```
par(new=TRUE)
plot(z~area, ann=F, axes=F, cex=1.5, pch=17)
axis(4)
```

```
par(mar=c(3,5,0.5,4))
par(bty="l")
boxplot(box~samples)
```

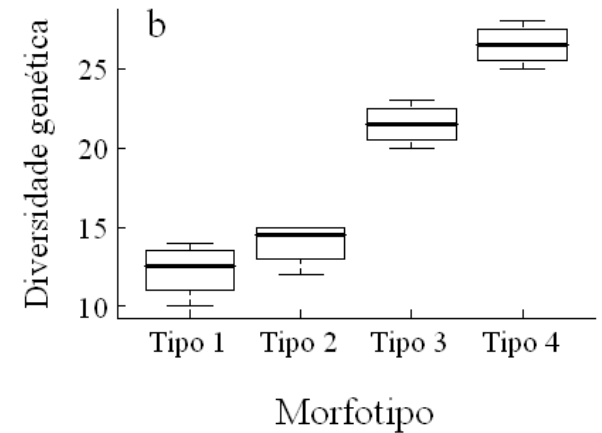
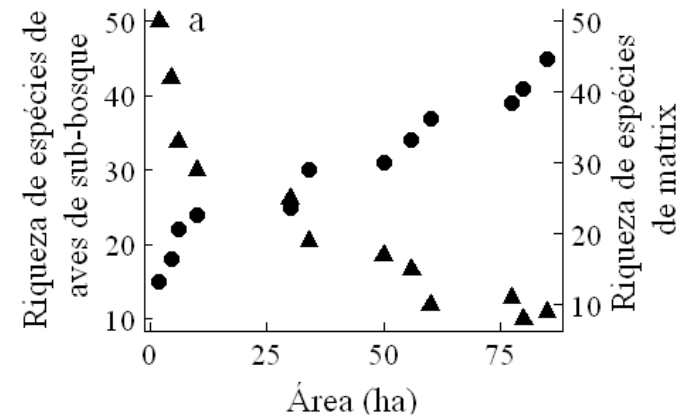


Inserindo texto

```
par(mfrow=c(2,1))  
par(...)  
plot(riqueza~area, ...)  
text(10,50, "a", cex=1.8)
```

```
par(new=TRUE)  
plot(z~area, ann=F, axes=F, cex=1.5, pch=17)  
axis(4)  
mtext("Riqueza de espécies \nde matrix", side=4, cex=1.5, line=3.2, las=0)
```

```
par(mar=c(5,5,0.5,2))  
par(bty="l")  
boxplot(box~samples, names= c("", "", "", ""))  
mtext(c("Tipo 1", "Tipo 2", "Tipo 3", "Tipo 4"),side= 1,cex=1.3,line=0.3,at=c(1,2,3,4))  
mtext("Diversidade genética", side=2, cex=1.5, line=2.5, las=0)  
mtext("Morfotipo", side=1, cex=1.7, line=3)  
text(0.7,28, "b", cex=1.8)
```



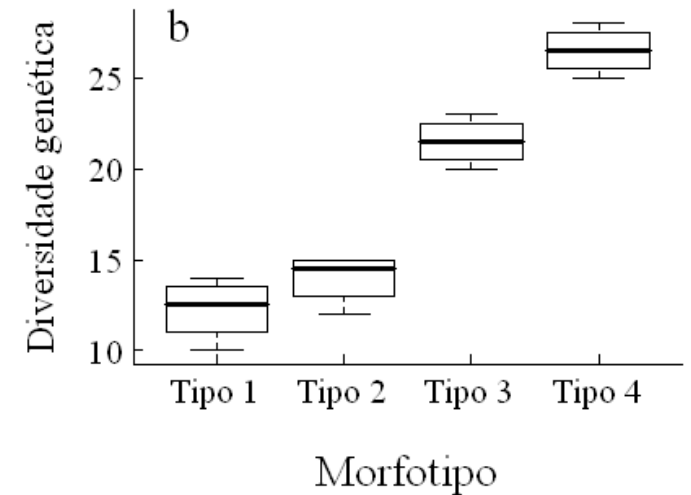
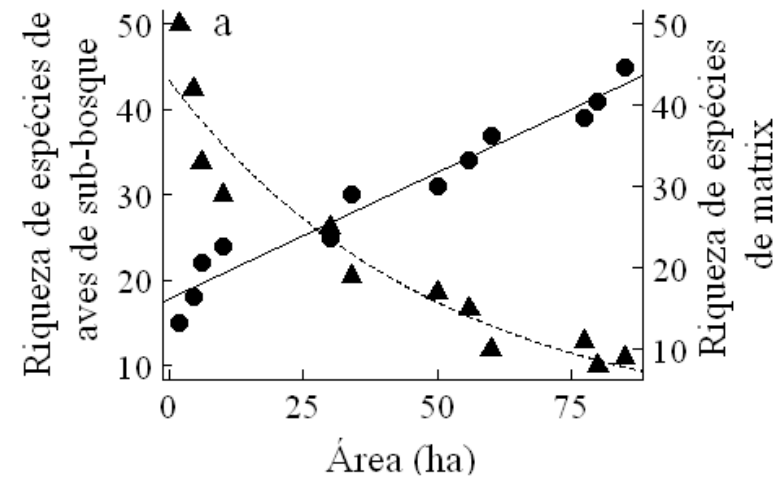
Inserindo linhas de tendência

Todos os códigos anteriores +:

```
model <- lm(riqueza~area)
model1 <- glm(z~area, poisson)
```

```
plot(riqueza~area, ...)
abline(model, lwd=1.5)
```

```
plot(z~area, ...)
xv<-seq(0,100,0.2)
yv<-exp(predict(model1,list(area=xv)))
lines(xv,yv, lwd=1.5, lty=2)
```



Cores

657 cores

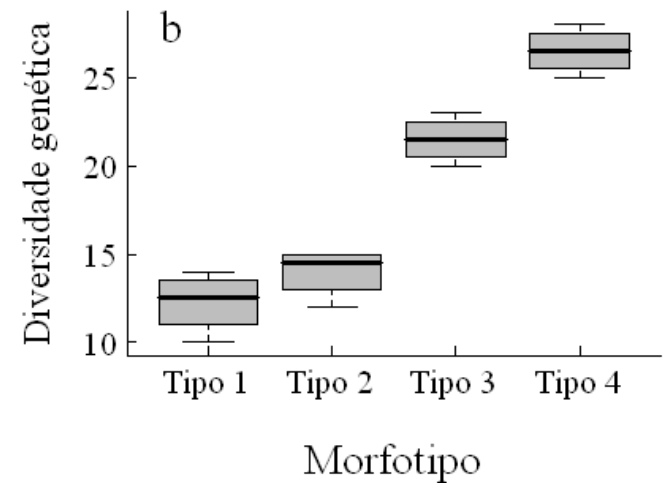
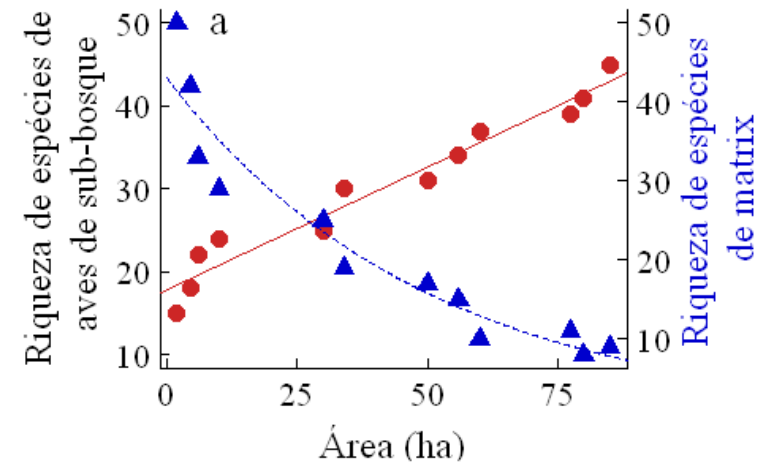
Cores podem ser indicadas por números ou nomes

Todos os códigos anteriores +:

```
plot(riqueza~area, ..., col="firebrick3")  
abline(model, lwd=1.5, col="firebrick3")
```

```
plot(z~area, ..., col="mediumblue")  
mtext("Riqueza ...matrix", col="mediumblue")  
lines(xv,yv, lwd=1.5, lty=2, col="mediumblue")
```

```
boxplot(box~samples, ..., col="grey")
```



só é possível mudar cor no mtext e text

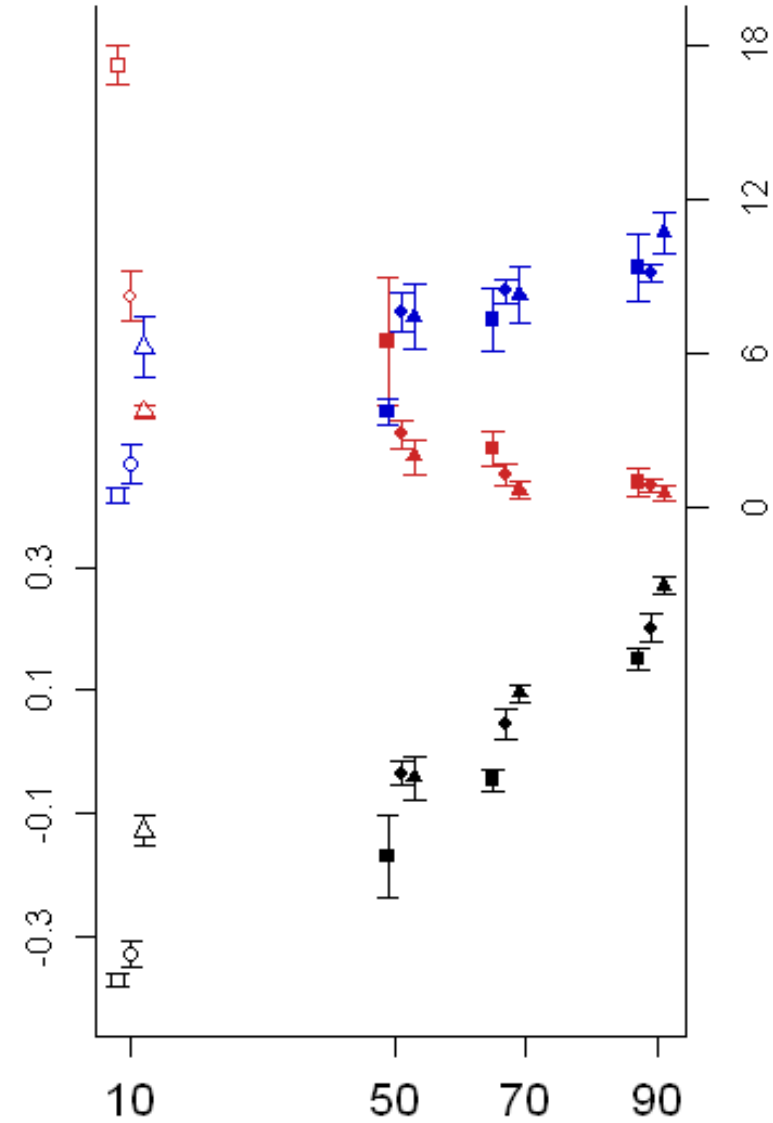
```
> colours()
```

```
[1] "white" "aliceblue" "antiquewhite"
[4] "antiquewhite1" "antiquewhite2" "antiquewhite3"
[7] "antiquewhite4" "aquamarine" "aquamarine1"
[10] "aquamarine2" "aquamarine3" "aquamarine4"
[13] "azure" "azure1" "azure2"
[16] "azure3" "azure4" "beige"
[19] "bisque" "bisque1" "bisque2"
[22] "bisque3" "bisque4" "black"
[25] "blanchedalmond" "blue" "blue1"
[28] "blue2" "blue3" "blue4"
[31] "blueviolet" "brown" "brown1"
[34] "brown2" "brown3" "brown4"
[37] "burlywood" "burlywood1" "burlywood2"
[40] "burlywood3" "burlywood4" "cadetblue"
[43] "cadetblue1" "cadetblue2" "cadetblue3"
[46] "cadetblue4" "chartreuse" "chartreuse1"
[49] "chartreuse2" "chartreuse3" "chartreuse4"
[52] "chocolate" "chocolate1" "chocolate2"
[55] "chocolate3" "chocolate4" "coral"
[58] "coral1" "coral2" "coral3"
[61] "coral4" "cornflowerblue" "cornsilk"
[64] "cornsilk1" "cornsilk2" "cornsilk3"
```

Dicas aleatórias – Error bars

```
x = cover.j  
uppy1 = rich.m+rich.sd  
lowy1 = rich.m-rich.sd
```

```
plot(rich.m~x, ...)  
arrows(x,uppy1,x,lowy1,length=.05,angle=90,code=3)
```

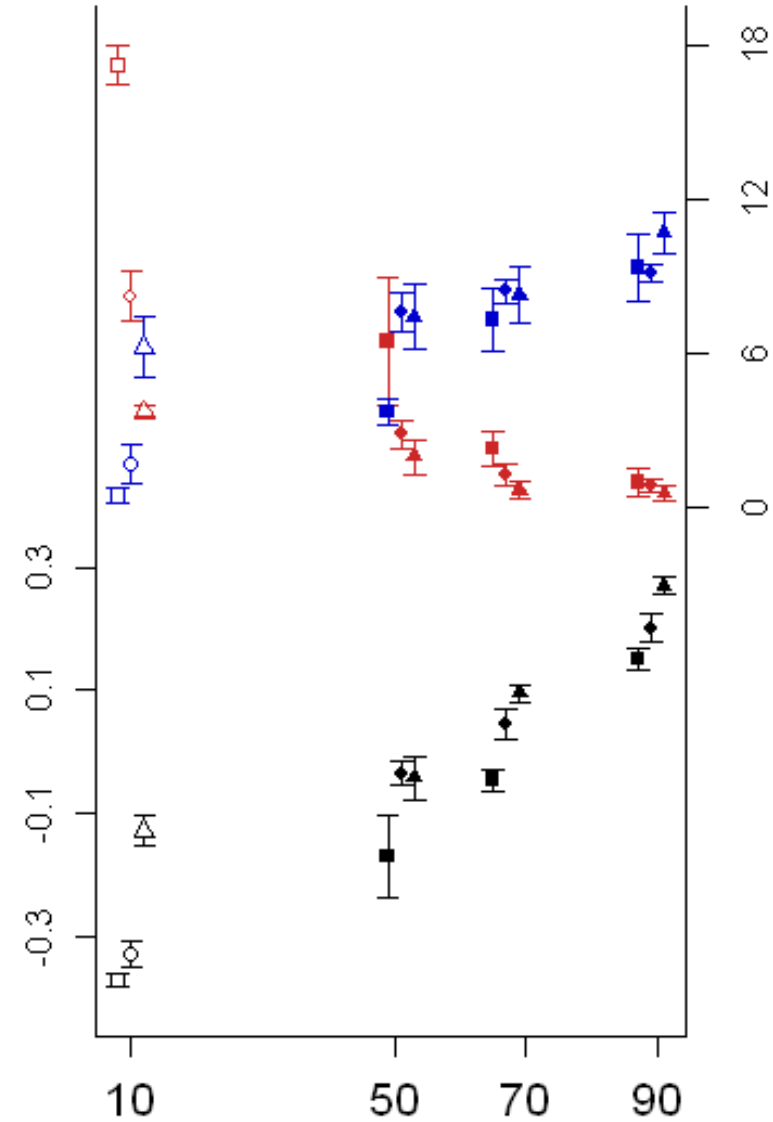


Dicas aleatórias – 1 figura, vários gráficos

```
plot(rich.1~x, ylim= c(-19,18), ann=FALSE,  
axes=FALSE, data=mat1,...)
```

```
par(new=TRUE)  
plot(rich.2~x, ylim= c(-19,18), ann=FALSE,  
axes=FALSE,  
axis(4,yaxp = c(0,18,3))
```

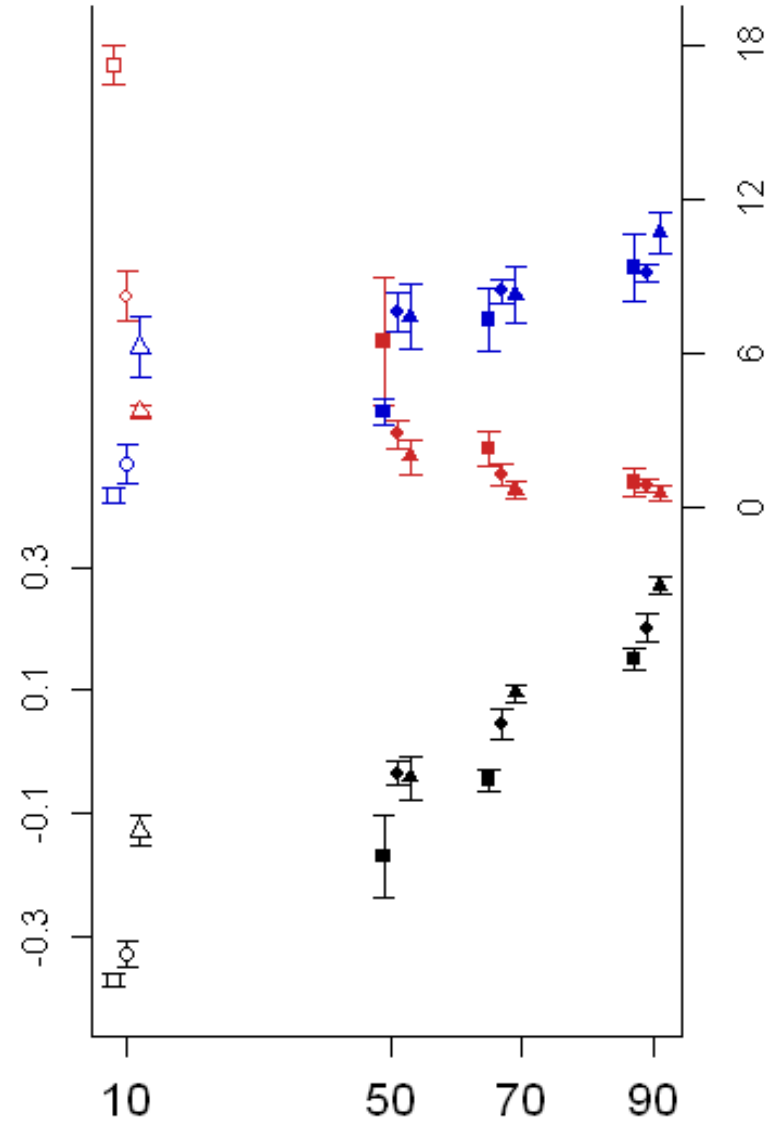
```
par(new=TRUE)  
plot(pcoa.m~x, ylim= c(-0.4,1.15), ann=FALSE,  
xaxt="n",
```



Dicas aleatórias – Mudando pontos

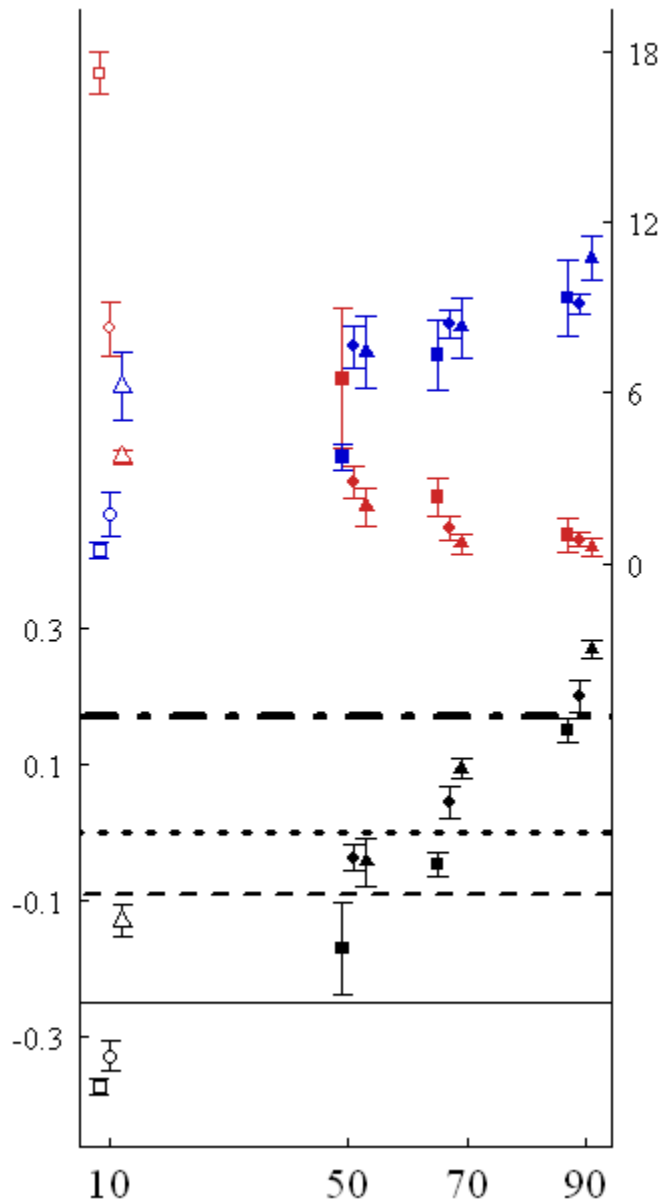
```
points(x[1],rich.sens.m[1], pch=22, bg="white",cex=0.9)  
points(x[2],rich.sens.m[2], pch=21, bg="white",cex=0.9)  
points(x[3],rich.sens.m[3], pch=24, bg="white",cex=0.9)
```

```
plot(pcoa.m~x,..., pch=c(0,1,2,15,16,17,15,16,17,15,16,17),  
cex=c(0.3,0.3,0.3,0.9,0.9,0.9,0.9,0.9,0.9,0.9,0.9,0.9))
```



Dicas aleatórias – Inserindo linhas divisórias

```
lines(c(0,100),c(-0.25, -0.25), lty=1, lwd=1)  
lines(c(0,100),c(-0.09, -0.09), lty=2 , lwd=2)  
lines(c(0,100),c(0, 0), lty=3 , lwd=3)  
lines(c(0,100),c(0.173, 0.173), lty=4 , lwd=4)
```



```

riqueza <- c(15,18,22,24,25,30,31,34,37,39,41,45)
area <- c(2,4.5,6,10,30,34,50,56,60,77.5,80,85)
box <- c(10,13,12,14,15,12,14,15,20,23,22,21,26,27,28,25)
samples <- rep(1:4,each=4)
model <- lm(riqueza~area)
model1 <- glm(z~area, poisson)
par(mfrow=c(2,1))
par(mar=c(3,5,2,4.2))
par(cex.axis=1.3)
par(cex.lab=1.5)
par(family="serif")
par(las=1)
par(tcl=0.3)
par(mgp=c(2,0.3,0))
par(bty="u")
plot(riqueza~area, xlab="Área (ha)", ylab="Riqueza de espécies de\n aves de sub-bosque", cex=1.5, pch=16, xaxp=c(0,100,4), ylim=c(10,50), col="firebrick3")
text(10,50, "a", cex=1.8)
abline(model, lwd=1.5,col="firebrick3")
par(new=TRUE)
plot(z~area, ann=F, axes=F, cex=1.5, pch=17, col="mediumblue")
axis(4)
mtext("Riqueza de espécies \nde matrix", side=4, cex=1.5, line=3.2, las=0, col="mediumblue")
xv<-seq(0,100,0.2)
yv<-exp(predict(model1,list(area=xv)))
lines(xv,yv, lwd=1.5, lty=2, col="mediumblue")
par(mar=c(5,5,0.5,2))
par(bty="l")
boxplot(box~samples,names= c("", "", "", ""),col="grey")
mtext(c("Tipo 1", "Tipo 2", "Tipo 3", "Tipo 4"),side= 1,cex=1.3,line=0.3,at=c(1,2,3,4))
mtext("Diversidade genética", side=2, cex=1.5, line=2.5, las=0)
mtext("Morfotipo", side=1, cex=1.7, line=3)
text(0.7,28, "b", cex=1.8)

```