

Gráficos na linguagem R

Ivanildo Batista

8 de marco de 2021

Gráficos na linguagem R

Salvando graficos em png:

```
png("Grafico1.png", width=500, height = 500, res = 72)

plot(iris$Sepal.Length, iris$Petal.Length, col=iris$Species,
     main='Grafico gerado a partir do dataset Iris')
dev.off()
```

```
## png
## 2
```

Salvando graficos em pdf:

```
pdf("Grafico2.pdf")

plot(iris$Sepal.Length, iris$Petal.Length, col=iris$Species,
     main='Grafico gerado a partir do dataset Iris')
dev.off()
```

```
## png
## 2
```

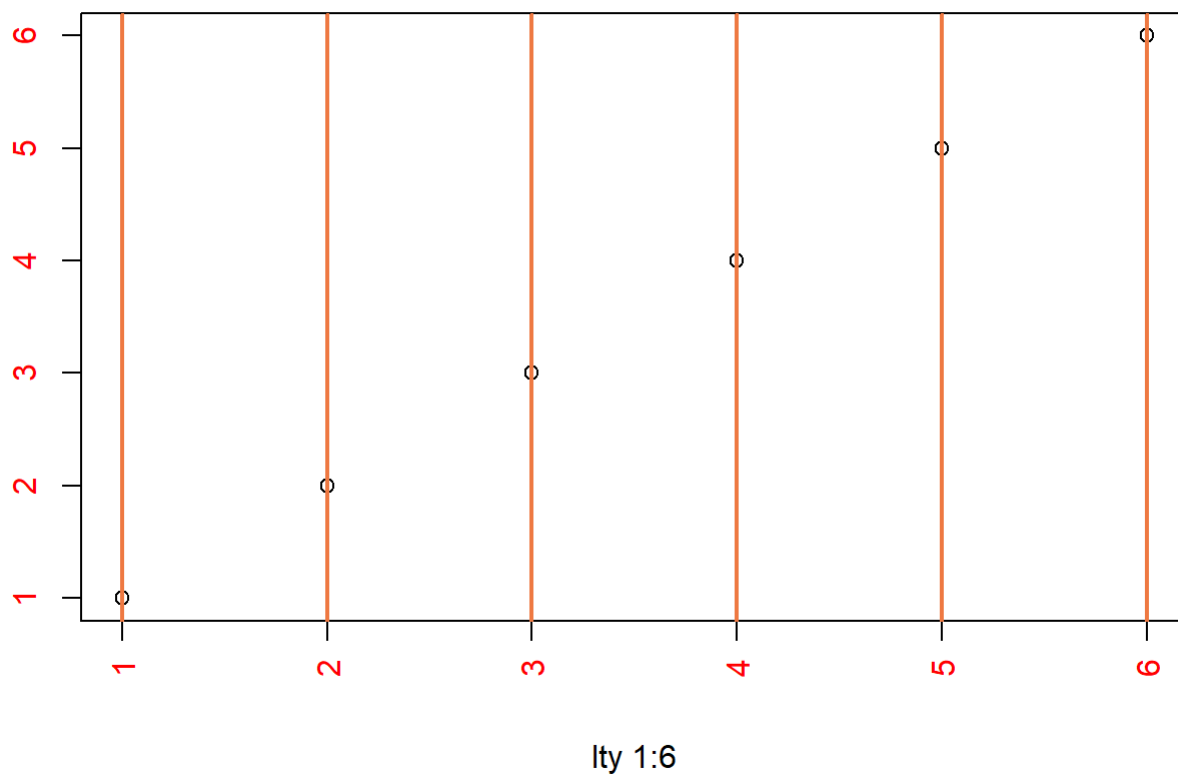
Estendendo a funcao *plot*:

```
library(plotrix)
```

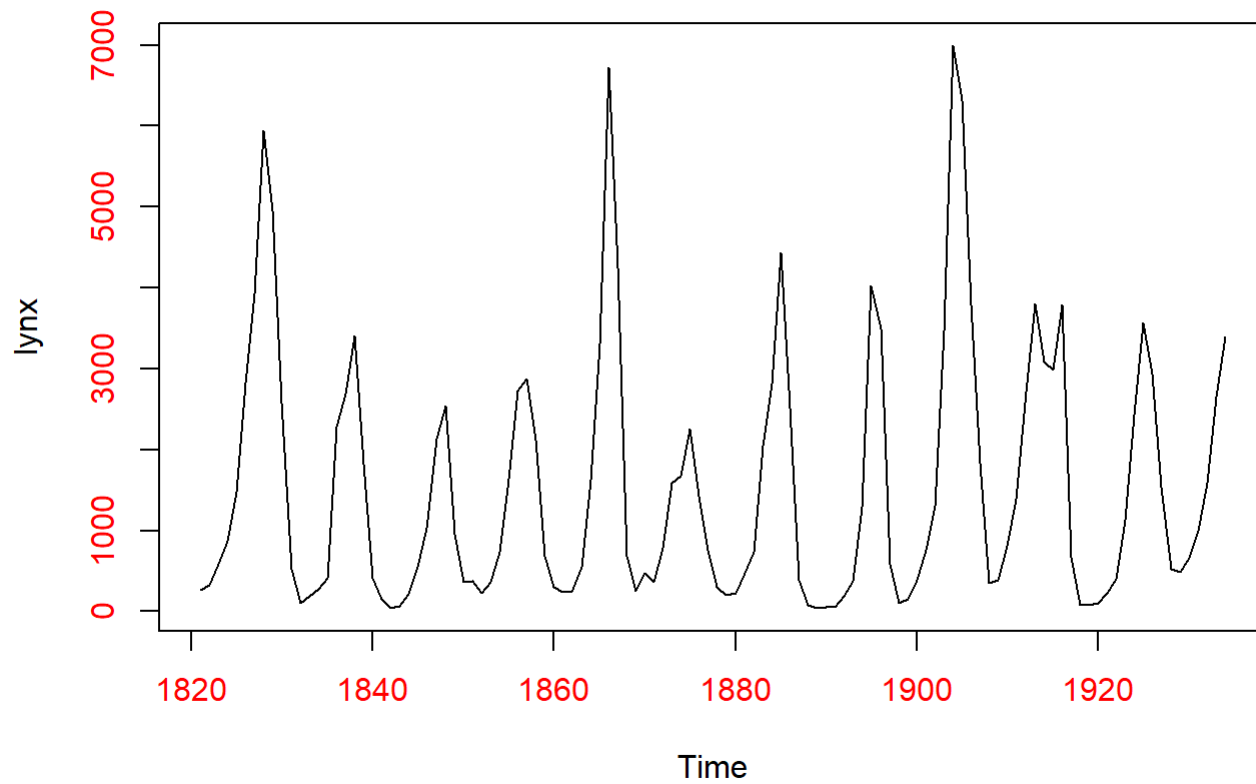
Criando um plot e adicionando linhas:

```
par(mfrow=c(1,1), col.axis='red')
plot(1:6, las=3, xlab = 'lty 1:6', ylab='', main = 'Mais opcoes ao plot')
ablineclip(v=1, lty=1, col='Sienna2', lwd=2)
ablineclip(v=2, lty=1, col='Sienna2', lwd=2)
ablineclip(v=3, lty=1, col='Sienna2', lwd=2)
ablineclip(v=4, lty=1, col='Sienna2', lwd=2)
ablineclip(v=5, lty=1, col='Sienna2', lwd=2)
ablineclip(v=6, lty=1, col='Sienna2', lwd=2)
ablineclip(v=7, lty=1, col='Sienna2', lwd=2)
```

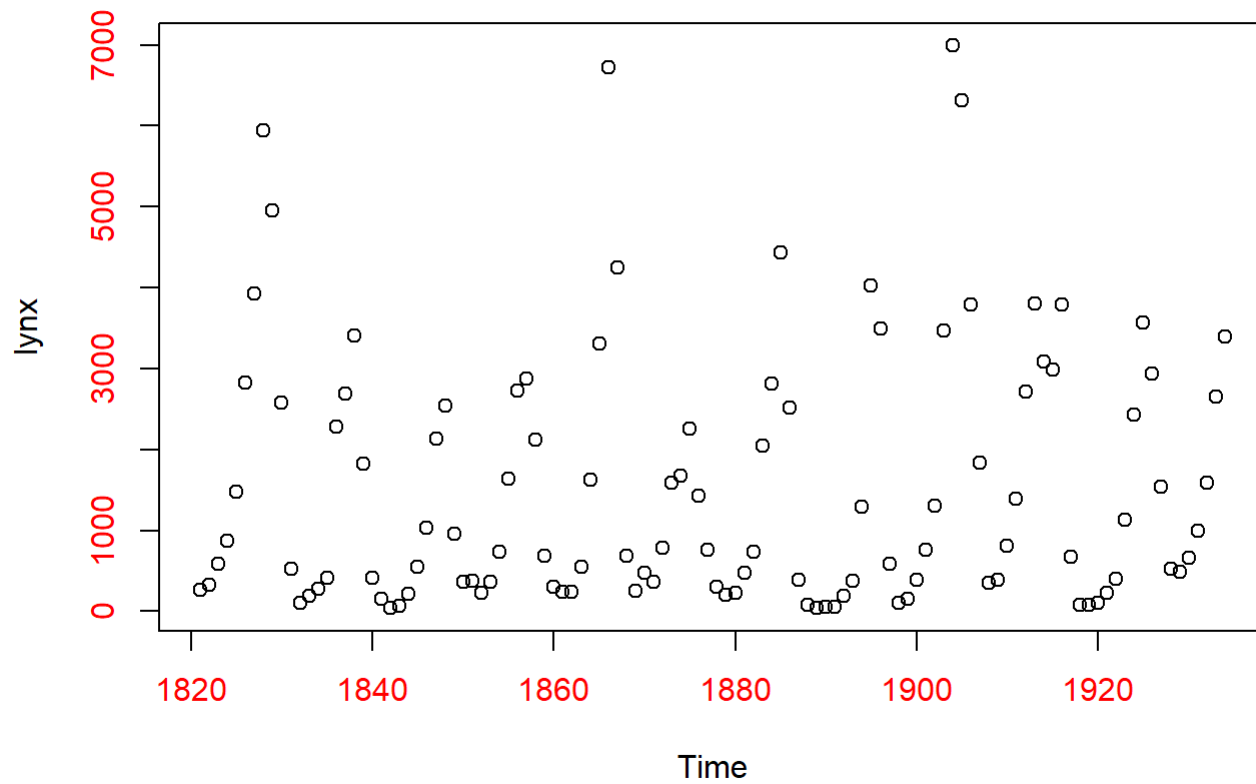
Mais opções ao plot



```
plot(lynx)
```

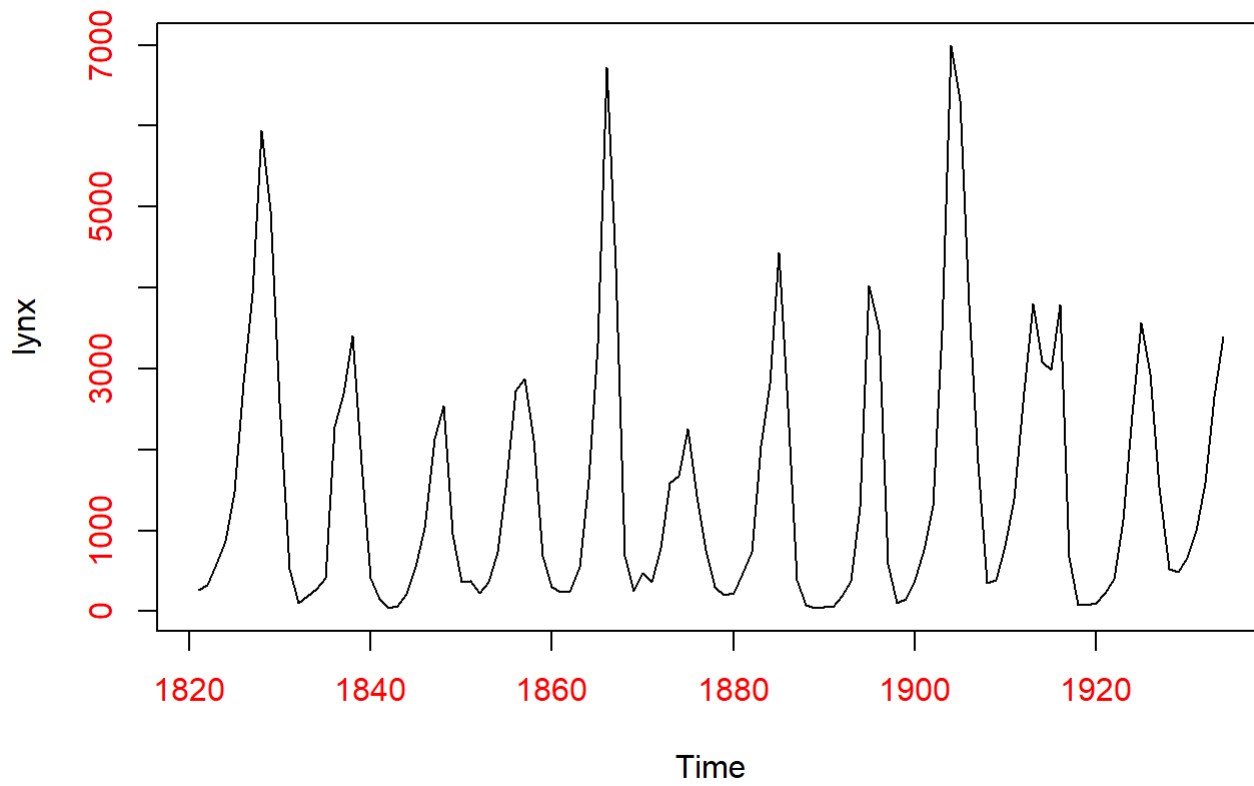


```
plot(lynx, type='p',main='type p') #pontos
```

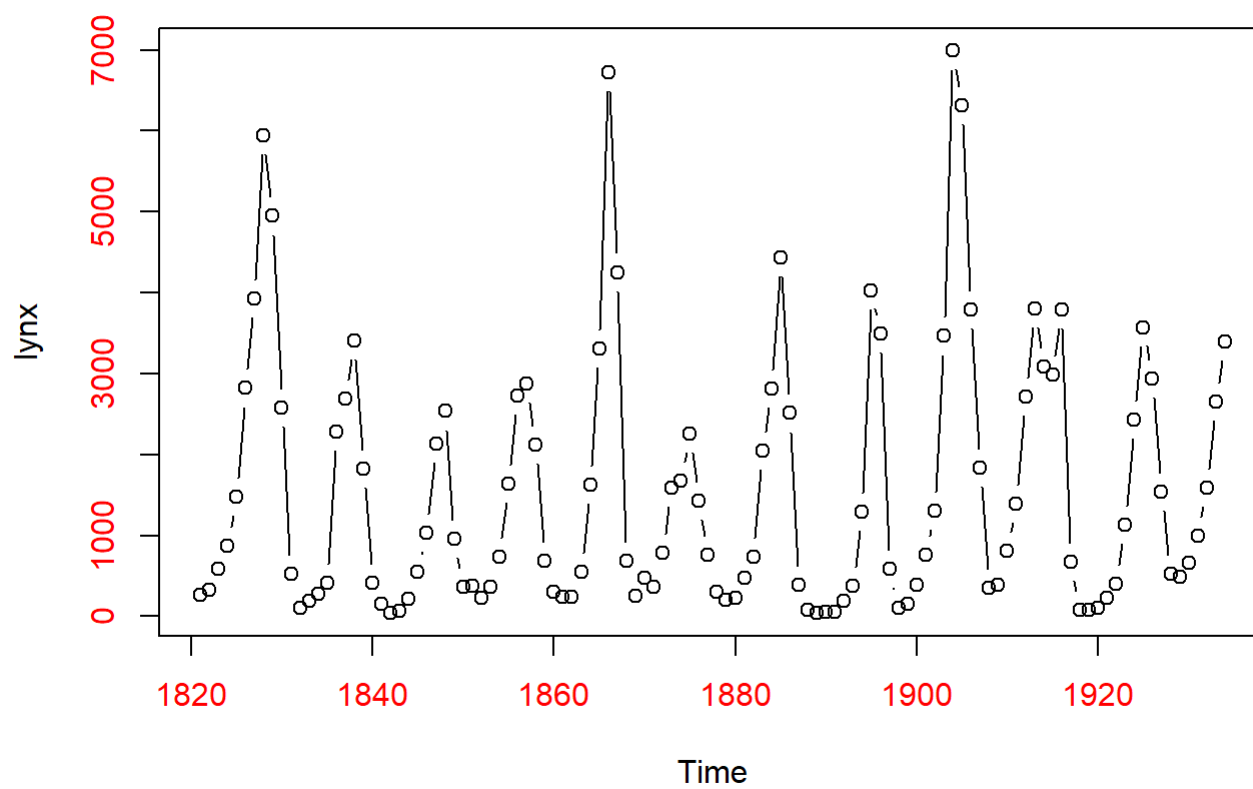
type p

```
plot(lynx, type='l',main='type l') #linhas
```

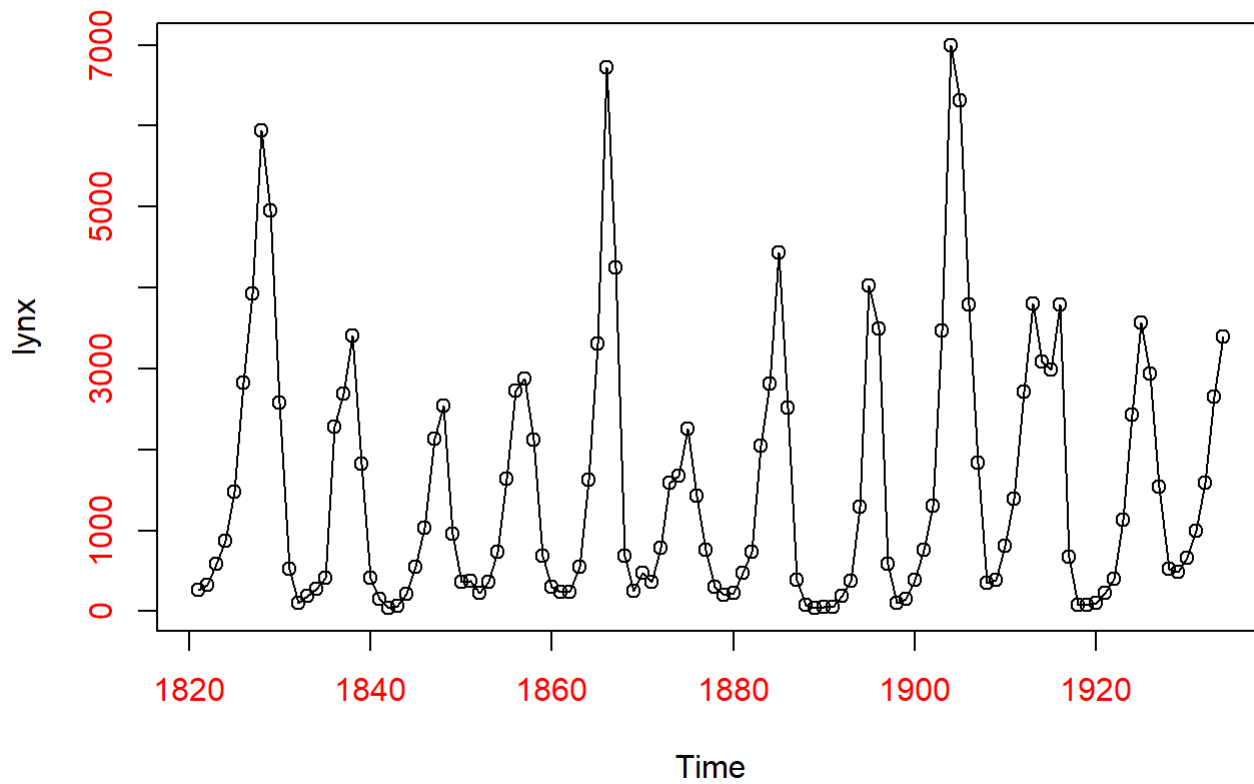
type l



```
plot(lynx, type='b', main='type b') #pontos e Linhas, mas nao se cruzam
```

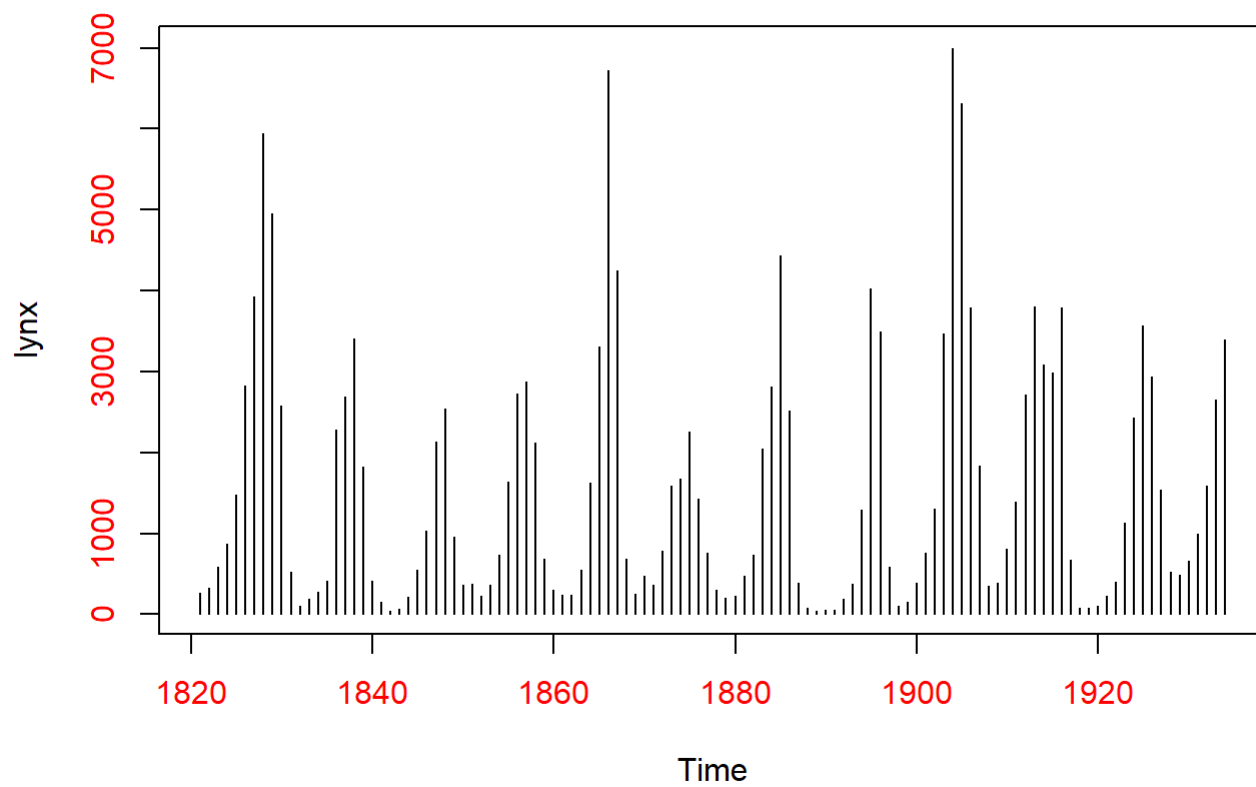
type b

```
plot(lynx, type='o', main='type o') #pontos e Linhas que se cruzam
```

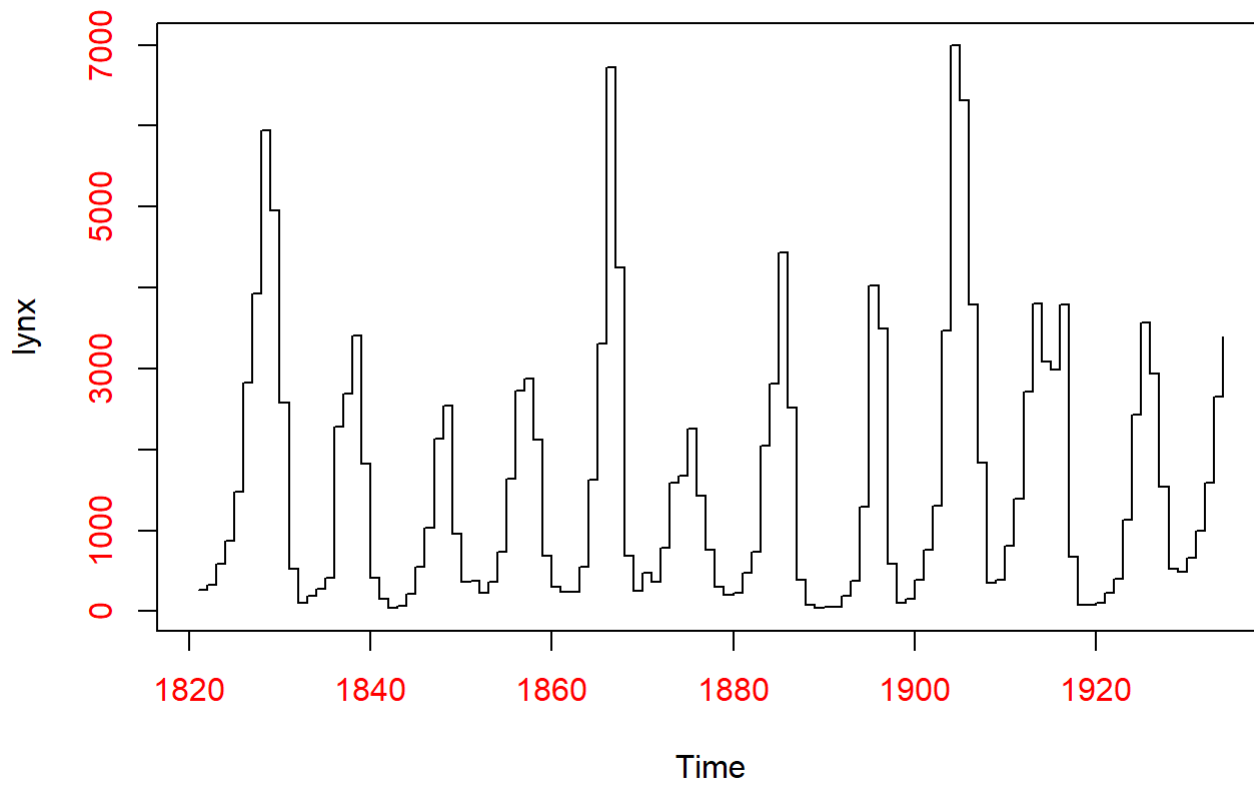
type o

```
plot(lynx, type='h',main='type h') #linhas verticais
```

type h

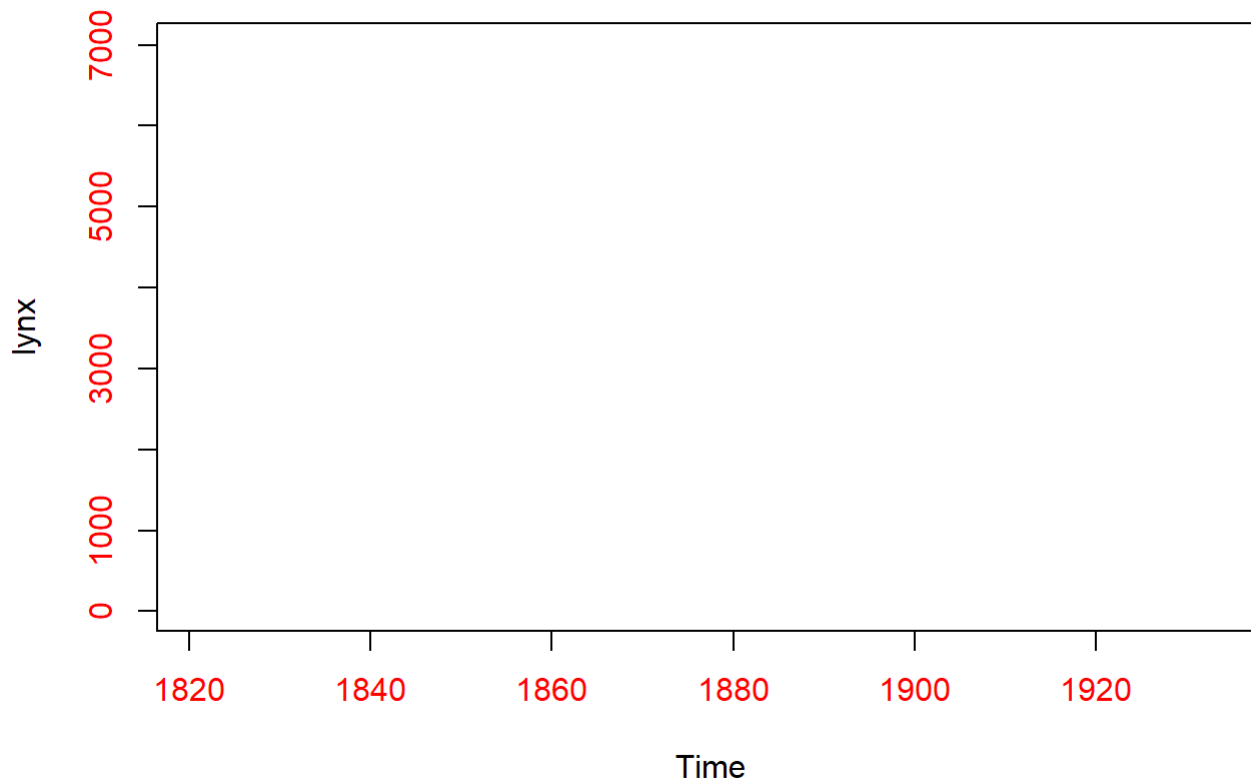


```
plot(lynx, type='s',main='type s') #retas
```


type s

```
plot(lynx, type='n',main='type n') #sem plotagem
```

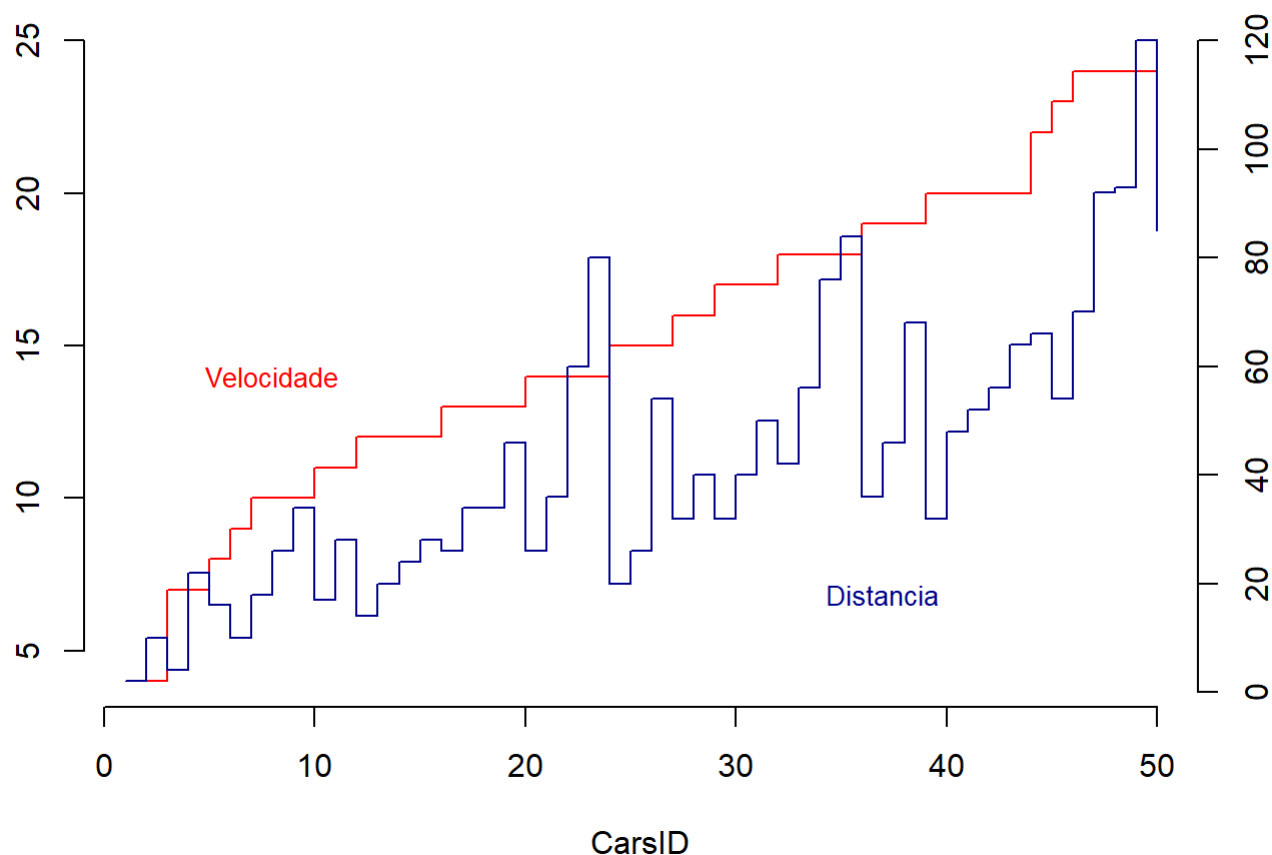
type n



Juntando dois *plots*:

```
par(mar=c(4,3,3,3), col.axis="black")
plot(cars$speed, type = "s", col="red", bty="n", xlab="CarsID", ylab = " ")
text(8,14, "Velocidade", cex=0.85, col="red") #dimensao 8x14
par(new=T)
plot(cars$dist, type = "s", col="darkblue", bty="n", ann=F, axes=F)
axis(side = 4)
text(37,18, "Distancia", cex=0.85, col="darkblue")
title(main = "Velocidade vs Distancia")
```

Velocidade vs Distancia

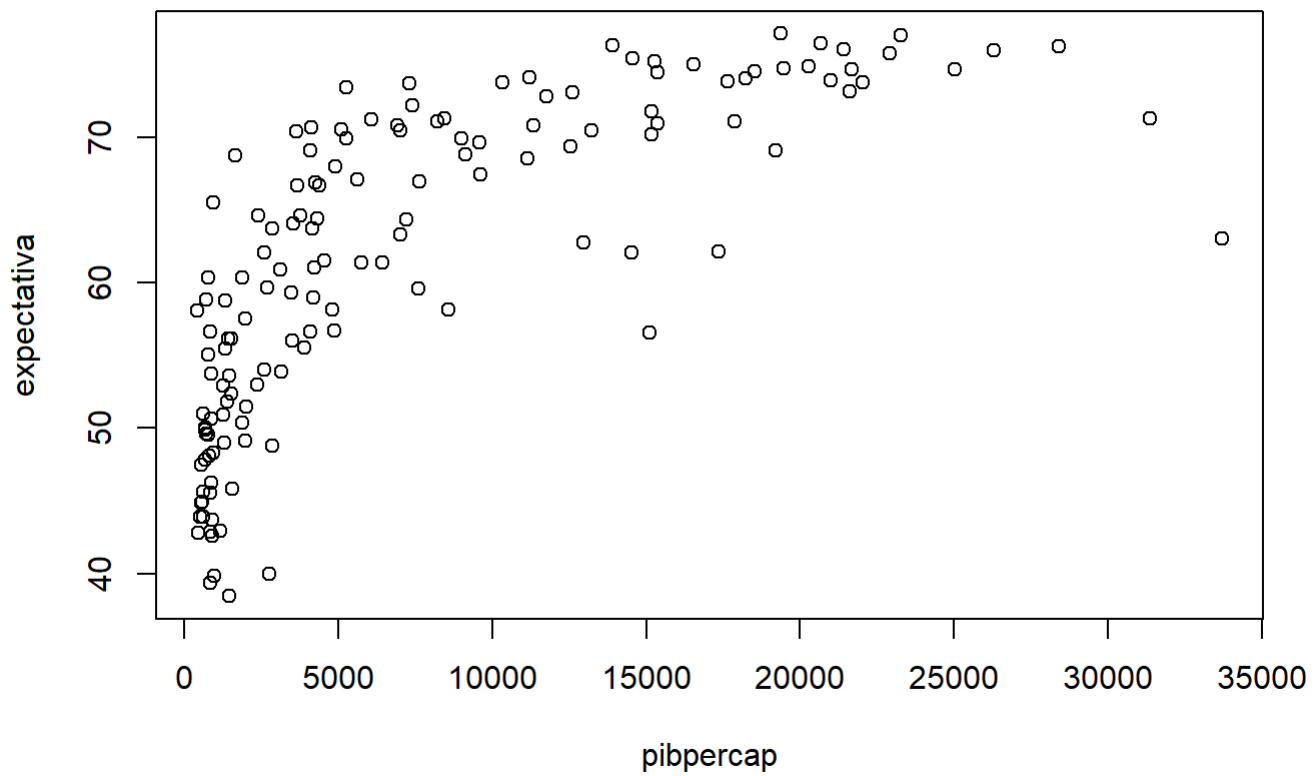


Plots a partir de datasets:

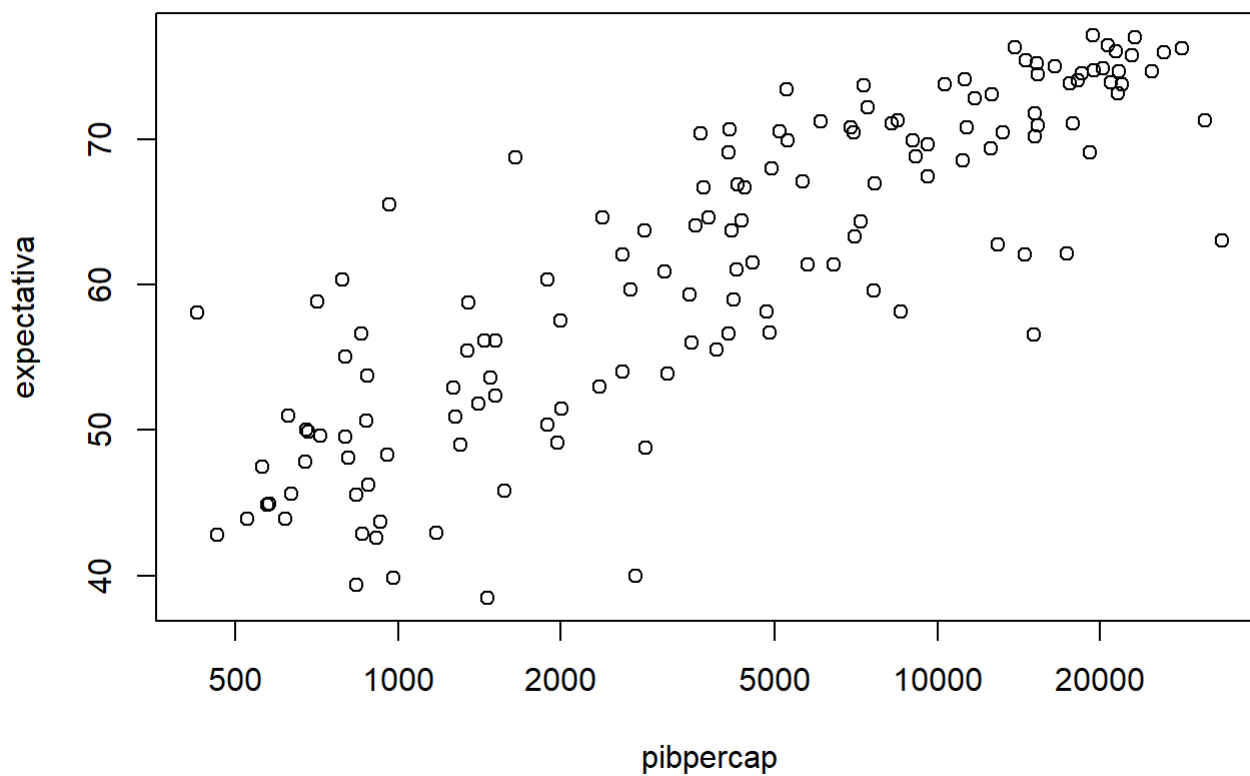
```
df = read.csv('C:/Users/junio/pibpercap.csv', stringsAsFactors = F)
```

```
df_1982 = subset(df, ano==1982)
View(df_1982)
```

```
plot(expectativa~pibpercap, data=df_1982)
```

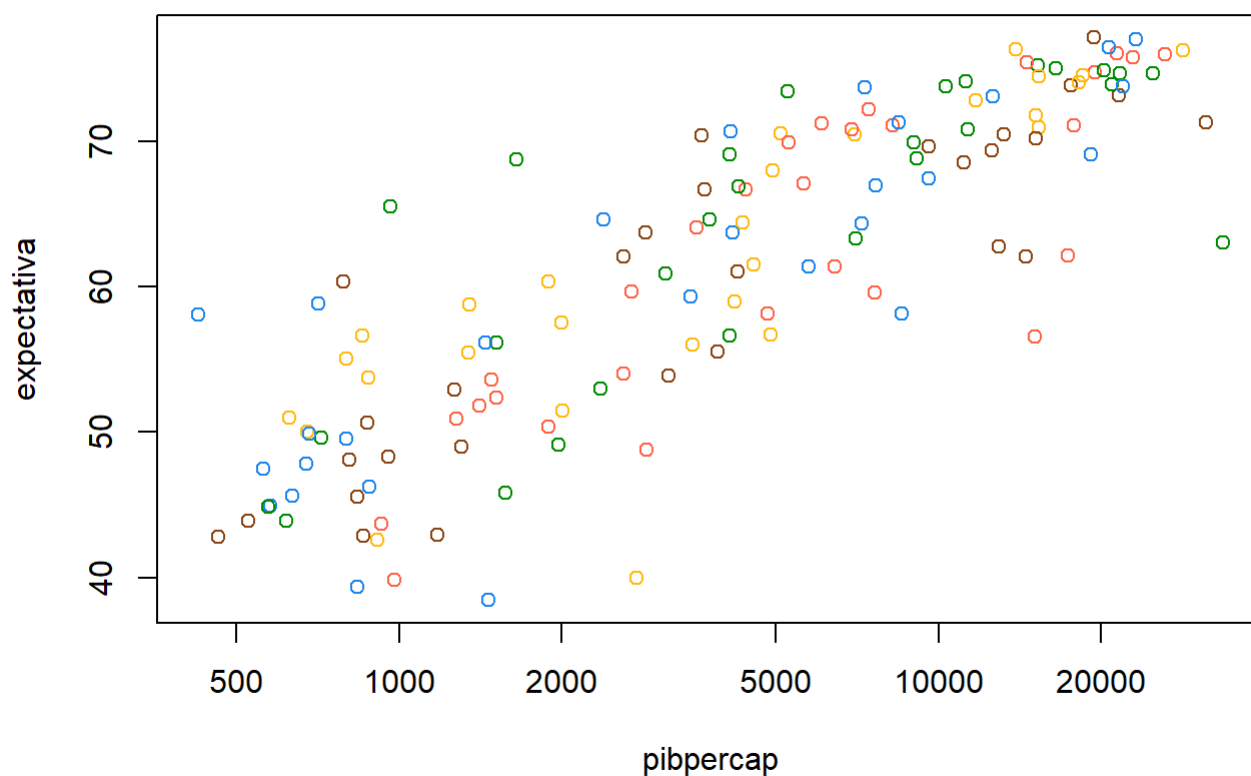


```
plot(expectativa~pibpercap, data=df_1982, log='x') #transformando a variavel x em log
```



```
mycol = c(Asia='tomato', Europe='chocolate4',Africa='dodgerblue2',  
          Americas='darkgoldenrod1',Oceania='green4') #gerando um vetor para destacar as cores d  
os continentes
```

```
plot(expectativa~pibpercap, data=df_1982, log='x', col=mycol)
```



Criando um funcao que ira redimensionar os valores dos dados

```
mycex = function(var,r,f=sqrt){
  x=f(var)
  x_scaled=(x-min(x))/(max(x)-min(x)) #colocando os dados na mesma dimensao
  r[1] + x_scaled*(r[2]-r[1])
}
```

Os tamanhos das bolinhas dos dados

```
plot(expectativa~pibpercap, data=df_1982, log='x', col=mycol[continente],
      cex=mycex(pop, r=c(0.2,10)))
```

