

# Gráficos no R

Marcus Suassuna Santos

06/10/2020

# Estrutura do Curso

- Sistemas gráficos no R
- *Base R* ou R “Básico”
- Aspectos de ggplot2

Para os exemplos utilizados neste Módulo, vamos utilizar dados de cotas de Manaus e dados do índice climático NINO34.

# Leitura de dados

## Lendo os dados

```
library(readr)
library(dplyr)
Cotas <- read_csv2("dados/cotas_T_14990000.txt", skip = 12) %>%
  filter(complete.cases(Maxima, Media, Minima)) %>%
  filter(NivelConsistencia == 1 & MediaDiaria == 1) %>%
  select(c("Data", "Maxima")) %>%
  mutate(Data = as.Date(Data, "%d/%m/%Y"),
         Normalizada = as.numeric(scale(Maxima))) %>%
  arrange(Data)
```

```
head(Cotas)
```

```
## # A tibble: 6 x 3
##   Data      Maxima Normalizada
##   <date>    <dbl>     <dbl>
## 1 1902-10-01  1970     -1.36
## 2 1902-11-01  1774     -1.95
## 3 1902-12-01  1909     -1.54
## 4 1903-01-01  2094     -0.986
```

# Leitura de dados

## Lendo os dados

```
library(readr)
```

```
NINO <- read_delim("https://psl.noaa.gov/gcos_wgsp/Timeseries/Data/nino34.long.anom.data",  
                  delim = " ", skip = 1, n_max = 151,  
                  col_names = c("Anos", "Jan", "Fev", "Mar", "Abr", "Mai", "Jun", "Jul", "Ago", "Se  
NINO <- as.data.frame(apply(NINO, 2, function(x)gsub('\\s+', '',x)))  
NINO <- as.data.frame(apply(NINO, 2, function(x) as.numeric(x)))
```

```
head(NINO)
```

```
##   Anos  Jan  Fev  Mar  Abr  Mai  Jun  Jul  Ago  Set  Out  Nov  Dez  
## 1 1870 -1.00 -1.20 -0.83 -0.81 -1.27 -1.08 -1.04 -0.88 -0.53 -0.92 -0.79 -0.79  
## 2 1871 -0.25 -0.58 -0.43 -0.50 -0.70 -0.53 -0.60 -0.33 -0.24 -0.33 -0.31 -0.58  
## 3 1872 -0.72 -0.62 -0.50 -0.77 -0.62 -0.52 -0.32 -0.85 -1.02 -0.94 -0.79 -0.88  
## 4 1873 -0.78 -1.01 -1.31 -0.67 -0.53 -0.48 -0.58 -0.39 -0.34 -0.78 -0.77 -0.70  
## 5 1874 -0.93 -1.06 -1.40 -0.94 -0.86 -0.72 -1.00 -1.05 -1.13 -1.25 -1.33 -1.14  
## 6 1875 -0.71 -0.37 -0.59 -0.87 -1.09 -0.76 -0.85 -0.81 -0.91 -0.83 -0.64 -0.75
```

# Leitura de dados

Lendo os dados

```
library(tidyr)
```

```
NINO_long <- gather(NINO, Mes, Nino, Jan:Dez, factor_key=TRUE) %>%  
  mutate(Mes = paste(Anos, Mes, "01", sep = "-"),  
         Mes = as.Date(Mes, "%Y-%b-%d")) %>%  
  arrange(Mes)
```

```
head(NINO_long, 12)
```

```
##      Anos      Mes  Nino  
## 1  1870 1870-01-01 -1.00  
## 2  1870 1870-02-01 -1.20  
## 3  1870 1870-03-01 -0.83  
## 4  1870 1870-04-01 -0.81  
## 5  1870 1870-05-01 -1.27  
## 6  1870 1870-06-01 -1.08  
## 7  1870 1870-07-01 -1.04  
## 8  1870 1870-08-01 -0.88  
## 9  1870 1870-09-01 -0.53  
## 10 1870 1870-10-01 -0.92
```

# Leitura de dados

Lendo os dados

```
Tabela <- merge(Cotas, NINO_long,  
               by.x = "Data", by.y = "Mes")  
head(Tabela)
```

```
##           Data Maxima Normalizada Anos Nino  
## 1 1902-10-01   1970  -1.3601133 1902 1.40  
## 2 1902-11-01   1774  -1.9509731 1902 1.54  
## 3 1902-12-01   1909  -1.5440033 1902 1.39  
## 4 1903-01-01   2094  -0.9863040 1903 1.33  
## 5 1903-02-01   2172  -0.7511659 1903 0.93  
## 6 1903-03-01   2332  -0.2688314 1903 0.79
```

# Leitura de dados

Lendo os dados, aplicando as máximas anuais

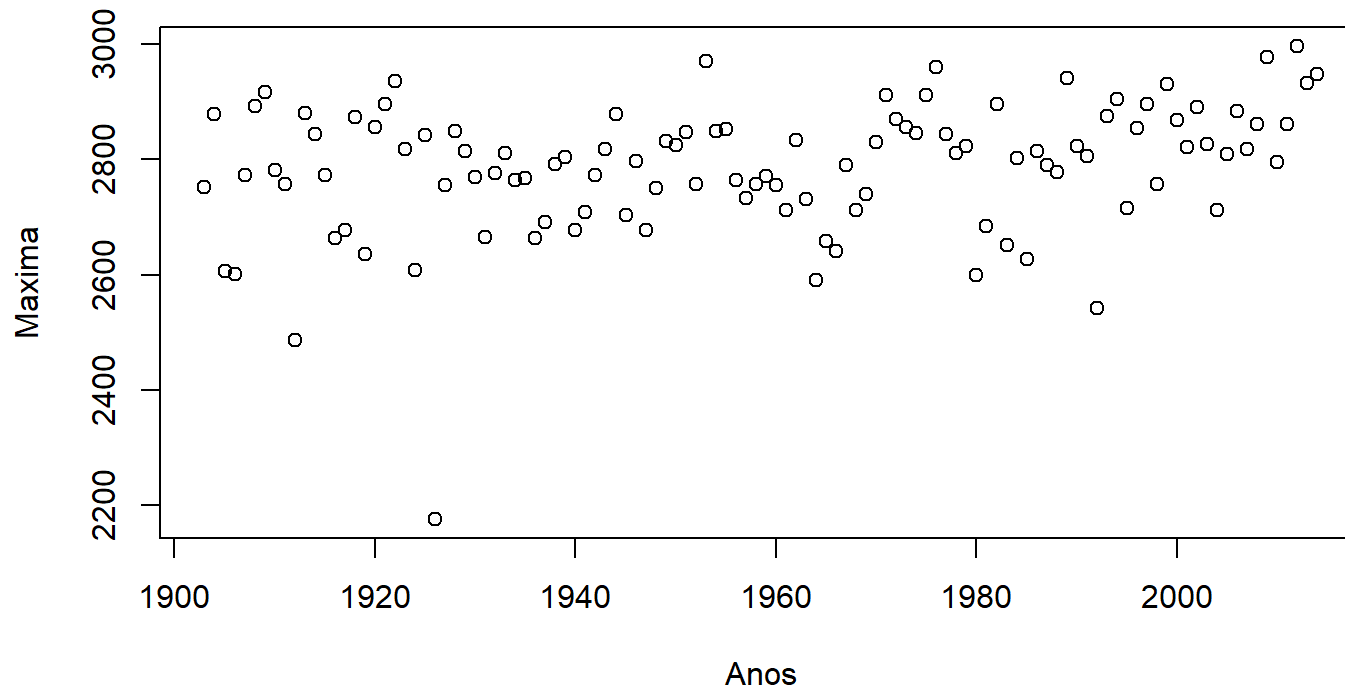
```
Tabela <- Tabela %>%
  group_by(Anos) %>%
  summarize(Maxima = max(Maxima),
            Nino = Nino[which.max(Maxima)]) %>%
  ungroup() %>%
  mutate(Normalizada = as.numeric(scale(Maxima)))
Tabela <- Tabela[-1,]
head(Tabela)
```

```
## # A tibble: 6 x 4
##   Anos Maxima   Nino Normalizada
##   <dbl> <dbl> <dbl>         <dbl>
## 1  1903   2752  1.33    -0.201
## 2  1904   2878 -0.73     0.712
## 3  1905   2607  0.68    -1.25
## 4  1906   2601  0.89    -1.30
## 5  1907   2773 -0.570  -0.0487
## 6  1908   2892 -0.27     0.814
```

# Primeiros gráficos

Diagrama de dispersão

```
with(Tabela, plot(Anos, Maxima))
```

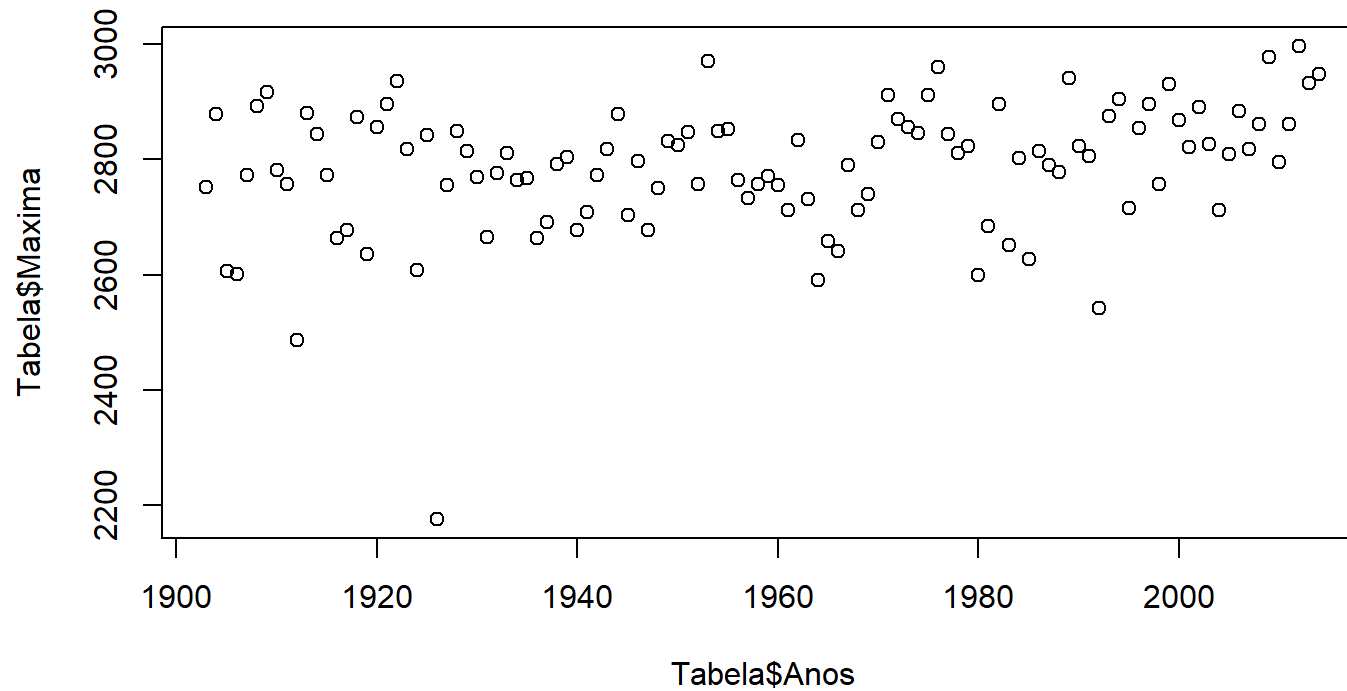




# Primeiros gráficos

Diagrama de dispersão

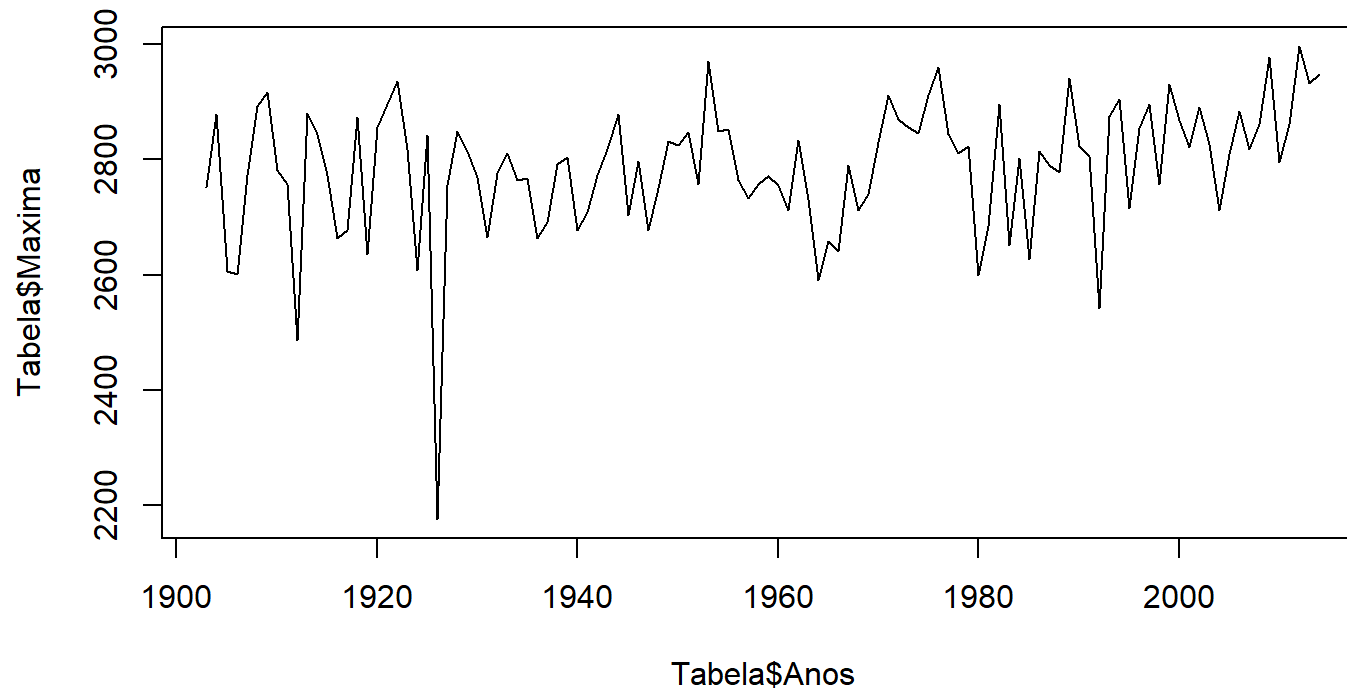
```
plot(Tabela$Anos, Tabela$Maxima)
```



# Primeiros gráficos

Série histórica

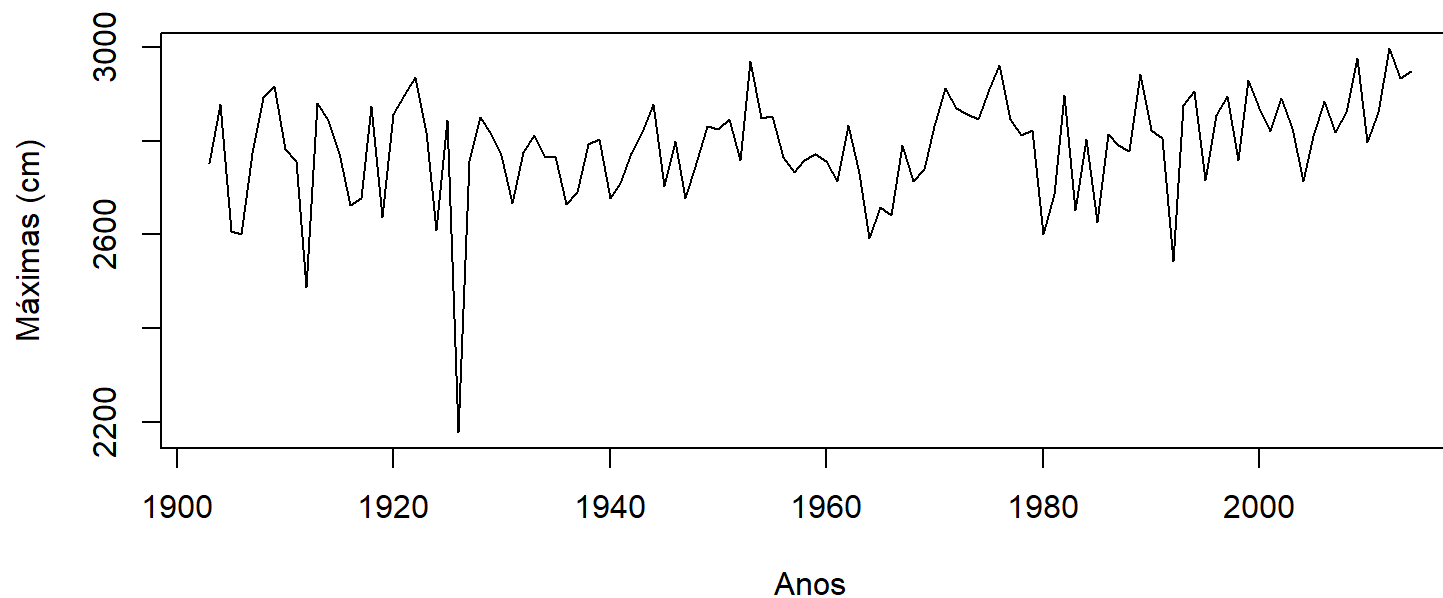
```
plot(Tabela$Anos, Tabela$Maxima, type = "l")
```



# Primeiros gráficos

Série histórica - alterando título dos eixos

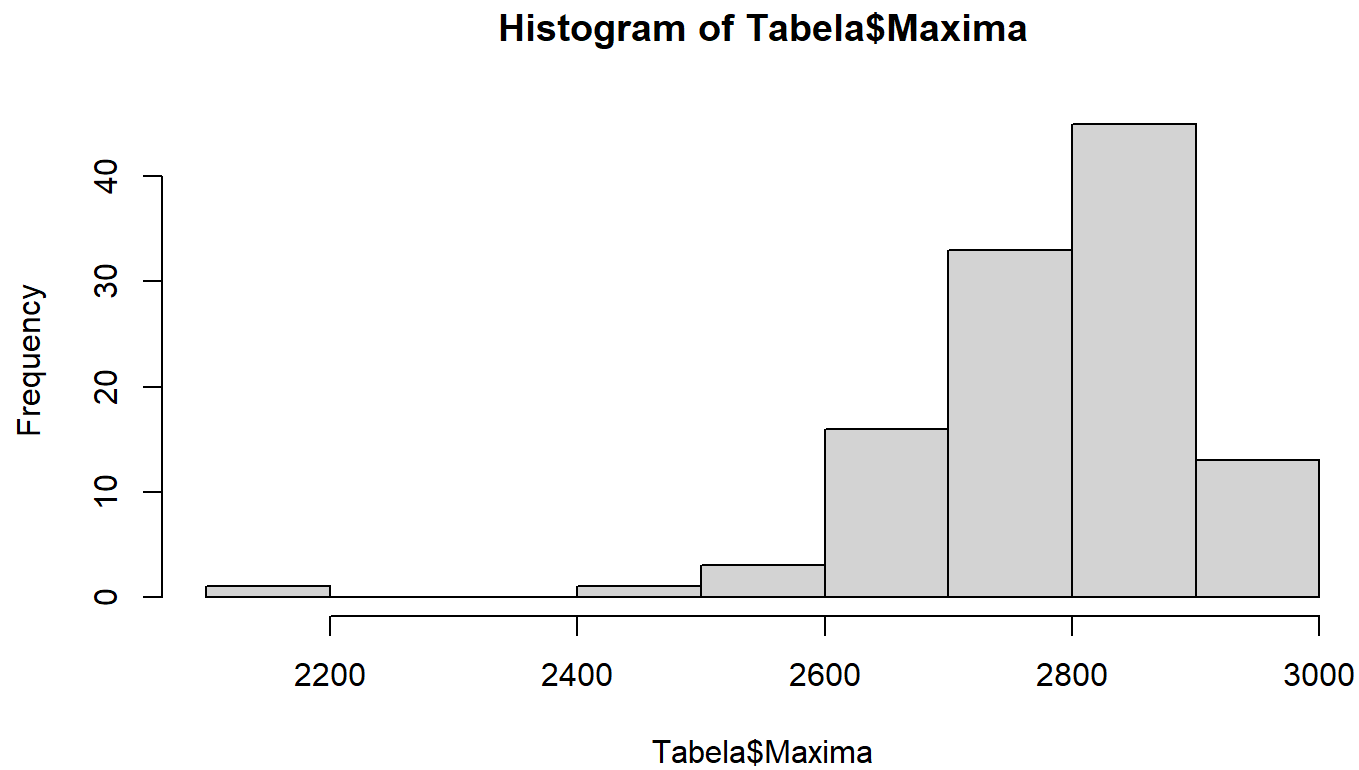
```
plot(Tabela$Anos, Tabela$Maxima, type = "l",  
      ylab = "Máximas (cm)", xlab = "Anos")
```



# Primeiros gráficos

## Histograma

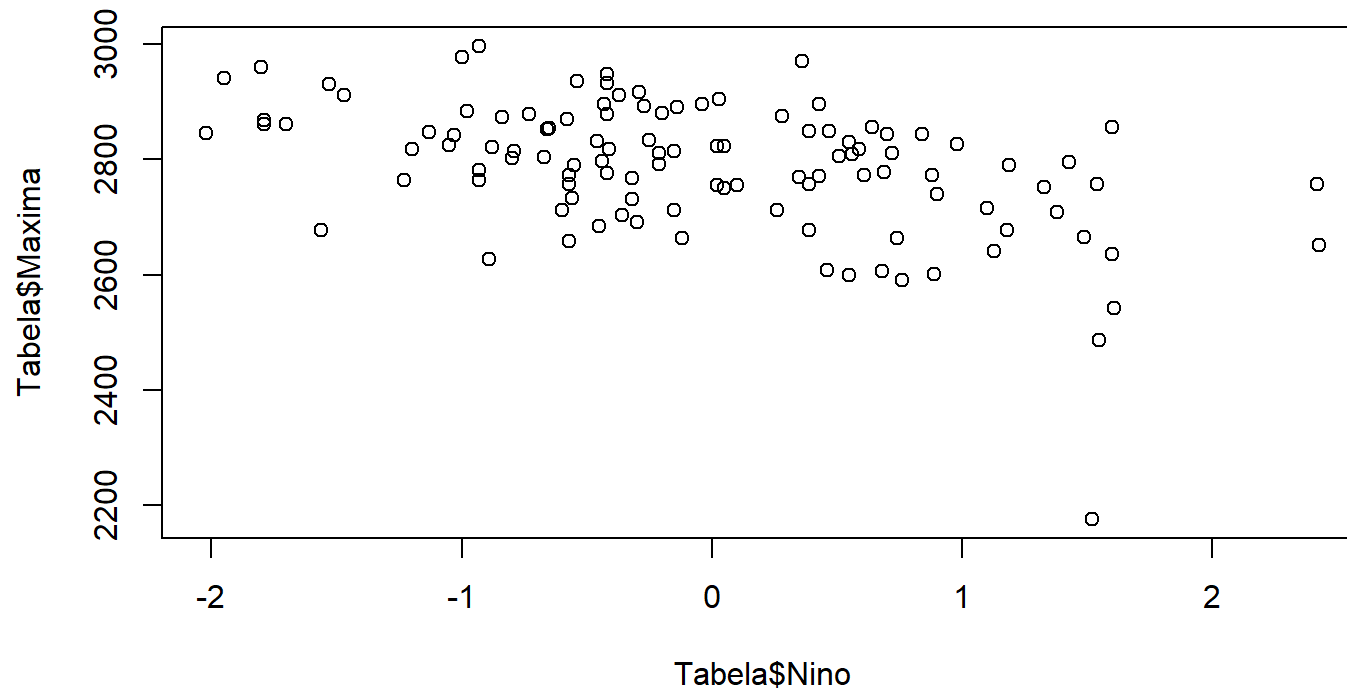
```
hist(Tabela$Maxima)
```



# Primeiros gráficos

Diagrama de dispersão

```
plot(Tabela$Nino, Tabela$Maxima)
```



# Primeiros gráficos

## Criando a classe de El Nino

```
Tabela$ClasseNino <- cut(Tabela$Nino,  
                        breaks = c(-Inf, -1, 1, Inf),  
                        labels = c("LaNina", "Neutro", "ElNino"))
```

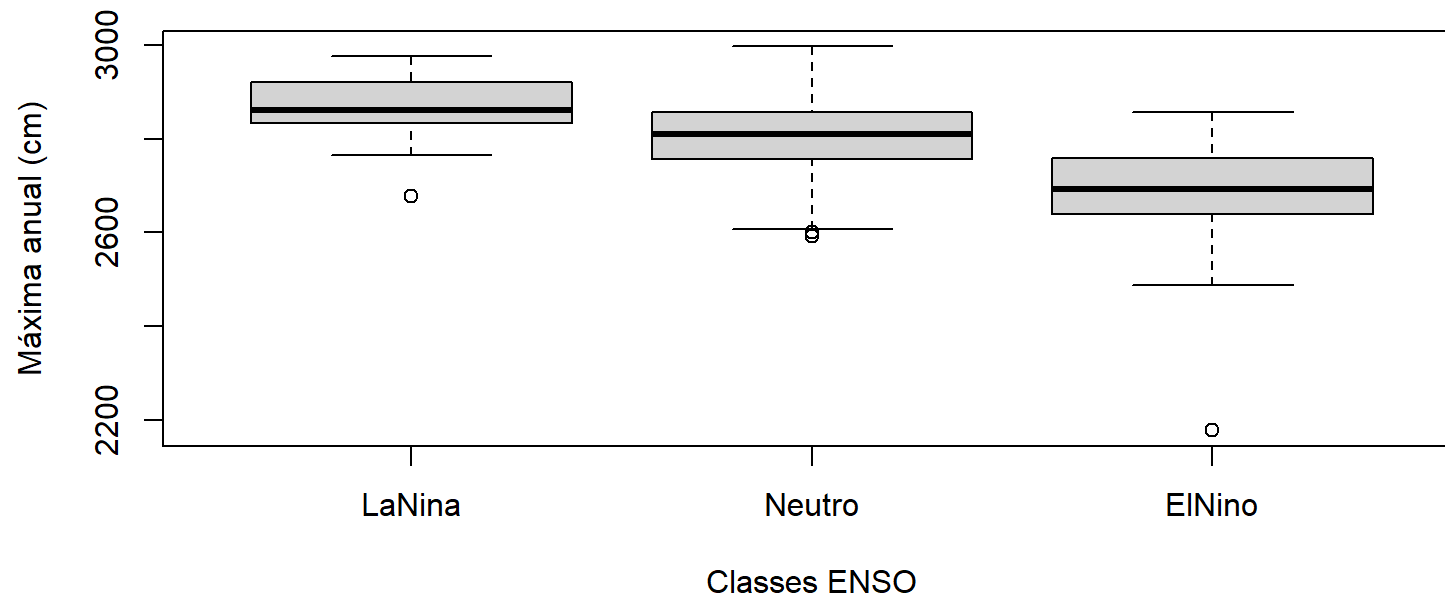
```
head(Tabela)
```

```
## # A tibble: 6 x 5  
##   Anos Maxima  Nino Normalizada ClasseNino  
##   <dbl> <dbl> <dbl>      <dbl> <fct>  
## 1  1903   2752  1.33     -0.201  ElNino  
## 2  1904   2878 -0.73      0.712  Neutro  
## 3  1905   2607  0.68     -1.25   Neutro  
## 4  1906   2601  0.89     -1.30   Neutro  
## 5  1907   2773 -0.570    -0.0487 Neutro  
## 6  1908   2892 -0.27      0.814  Neutro
```

# Primeiros gráficos

## Box Plot

```
boxplot(Maxima ~ ClasseNino, Tabela, xlab = "Classes ENSO",  
        ylab = "Máxima anual (cm)")
```



# Primeiros gráficos

## Box Plot

```
tapply(Tabela$Maxima, Tabela$ClasseNino, summary)
```

```
## $LaNina
```

```
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2677  2834    2862    2862  2921    2977
```

```
##
```

```
## $Neutro
```

```
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2591  2756    2810    2797  2857    2997
```

```
##
```

```
## $ElNino
```

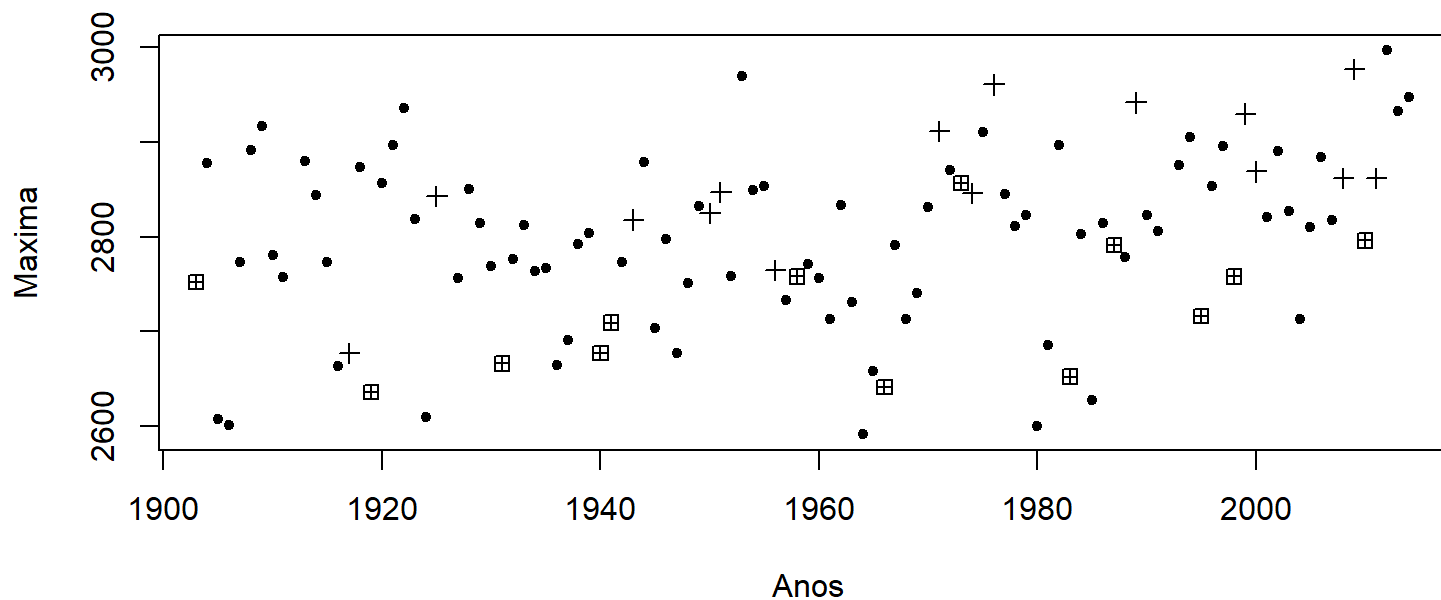
```
##   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2177  2640    2693    2663  2758    2857
```



# Opções gráficas

Inúmeras opções gráficas! - pch: símbolo gráfico

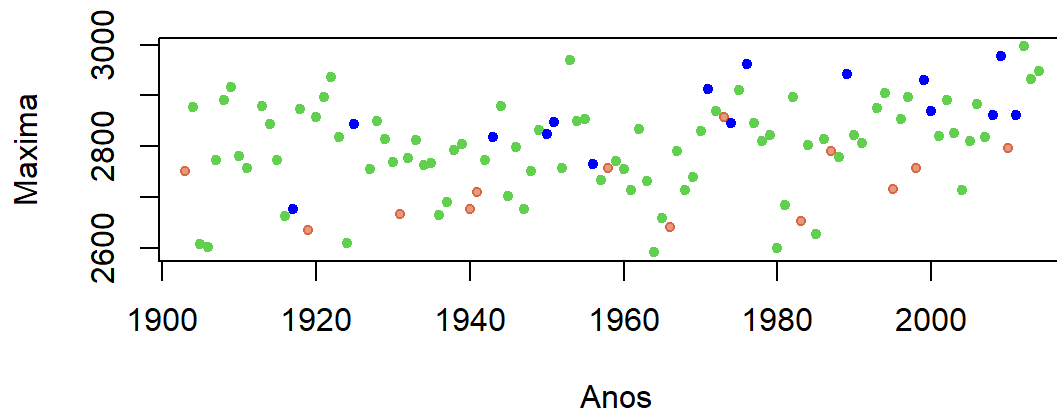
```
with(Tabela[Tabela$ClasseNino == "Neutro",], plot(Anos, Maxima, pch = 20))  
with(Tabela[Tabela$ClasseNino == "LaNina",], points(Anos, Maxima, pch = 3))  
with(Tabela[Tabela$ClasseNino == "ElNino",], points(Anos, Maxima, pch = 12))
```



# Opções gráficas

Inúmeras opções gráficas! - col: cor dos pontos

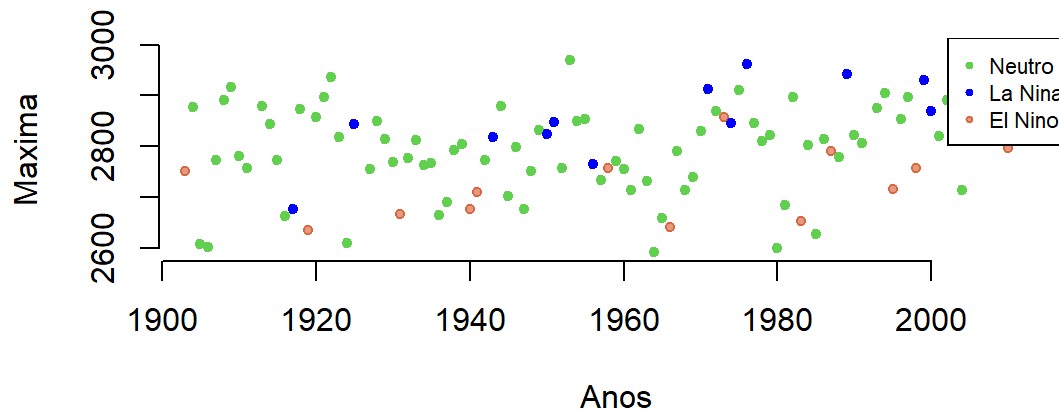
```
with(Tabela[Tabela$ClasseNino == "Neutro",],  
      plot(Anos, Maxima, pch = 20, col = 3))  
with(Tabela[Tabela$ClasseNino == "LaNina",],  
      points(Anos, Maxima, pch = 20, col = "blue"))  
with(Tabela[Tabela$ClasseNino == "ElNino",],  
      points(Anos, Maxima, pch = 20, col = rgb(0.8, 0.2, 0, 0.5)))
```



# Opções gráficas

Legendas:

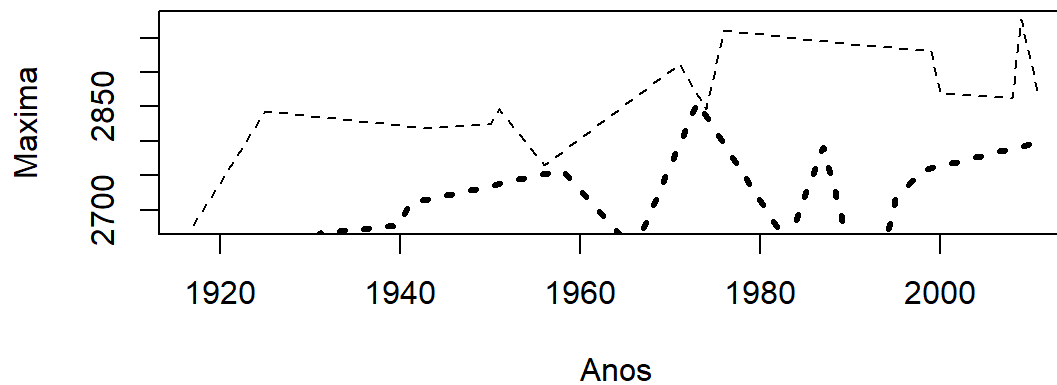
```
with(Tabela[Tabela$ClasseNino == "Neutro",], plot(Anos, Maxima, pch = 20, col = 3, bty = "n"))  
with(Tabela[Tabela$ClasseNino == "LaNina",], points(Anos, Maxima, pch = 20, col = "blue"))  
with(Tabela[Tabela$ClasseNino == "ElNino",], points(Anos, Maxima, pch = 20, col = rgb(0.8, 0.2, 0, 0.5)))  
legend("topright", pch = 20,  
      col = c(3, "blue", rgb(0.8,0.2,0, 0.5)),  
      legend = c("Neutro", "La Nina", "El Nino"), cex = 0.7)
```



# Opções gráficas

Inúmeras opções gráficas! - lty: tipo de linha - lwd: espessura da linha

```
with(Tabela[Tabela$ClasseNino == "LaNina",],  
      plot(Anos, Maxima, type = "l", lty = 2))  
with(Tabela[Tabela$ClasseNino == "ElNino",],  
      lines(Anos, Maxima, lty = 3, lwd = 3))
```



# Opções gráficas

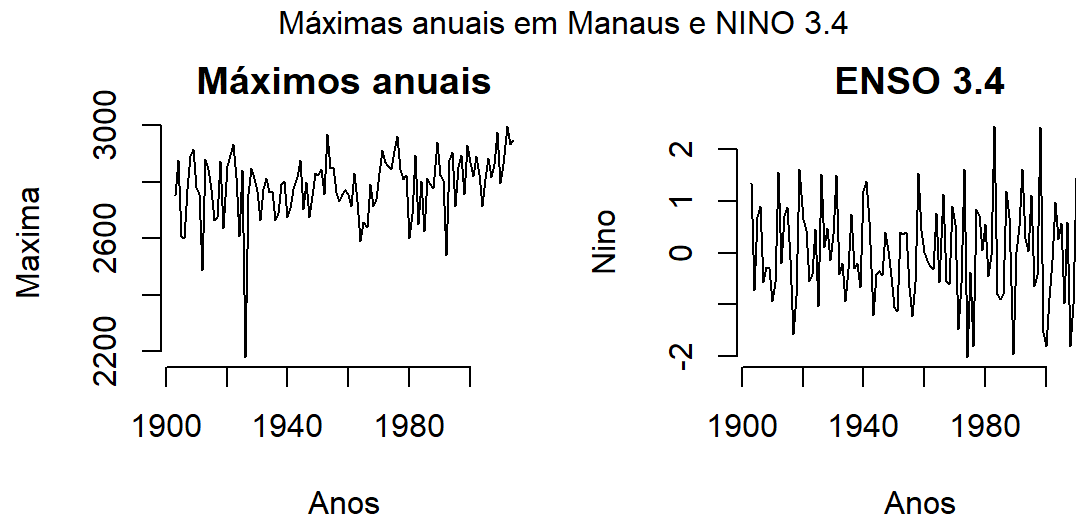
Outras opções gerais para gráficos, utilizando a função `par()`:

- `las`: orientação do texto das legendas
- `bg`: cor do fundo
- `mar`: tamanho das margens
- `oma`: margens externas
- `mfrow`: número de gráficos por linha
- `mfcop`: número de gráficos por coluna

# Opções gráficas

Exemplo:

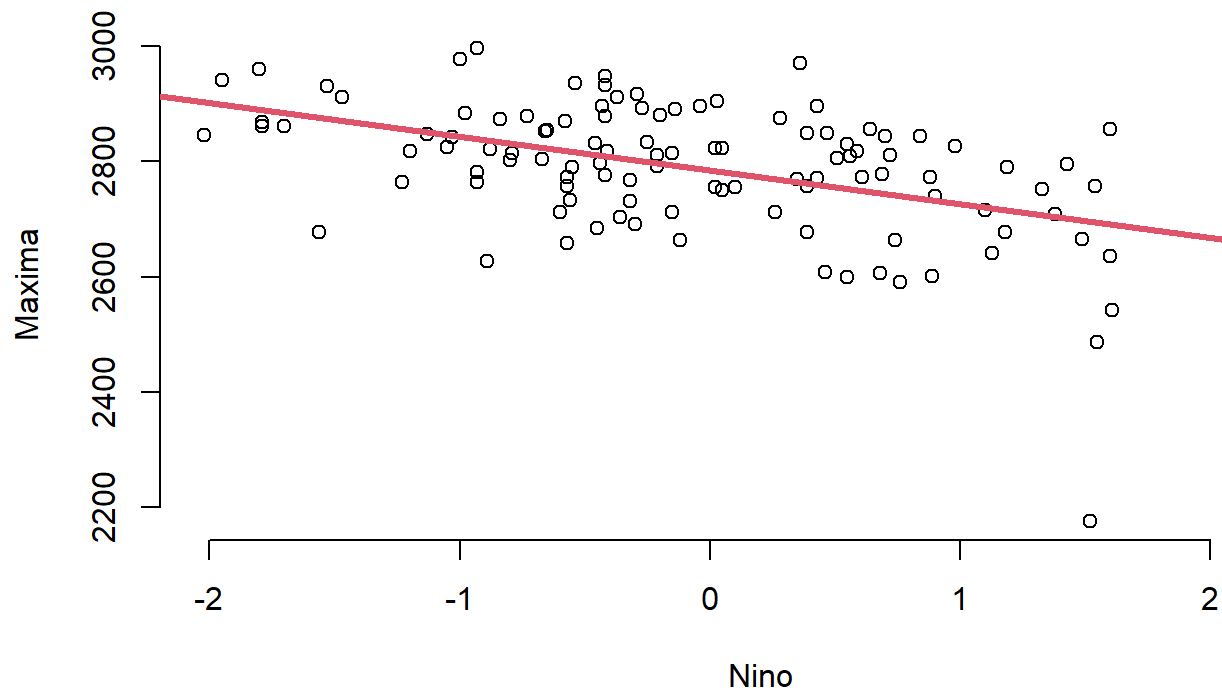
```
par(mar = c(4.5,4.5,2,1), mfrow = c(1, 2), oma = c(0,0,2,0))  
with(Tabela, plot(Anos, Maxima, type = "l", bty = "n", main = "Máximos anuais"))  
with(Tabela, plot(Anos, Nino, type = "l", bty = "n", main = "ENSO 3.4"))  
mtext("Máximas anuais em Manaus e NINO 3.4", outer = TRUE)
```



# Opções gráficas

Gráficos com linhas de regressão:

```
with(Tabela, plot(Nino, Maxima, bty = "n"))  
fit <- lm(Maxima ~ Nino, data = Tabela)  
abline(fit, col = 2, lwd = 3)
```



# Opções gráficas

```
fit
```

```
##
```

```
## Call:
```

```
## lm(formula = Maxima ~ Nino, data = Tabela)
```

```
##
```

```
## Coefficients:
```

```
## (Intercept)      Nino
```

```
##      2785.23      -58.72
```



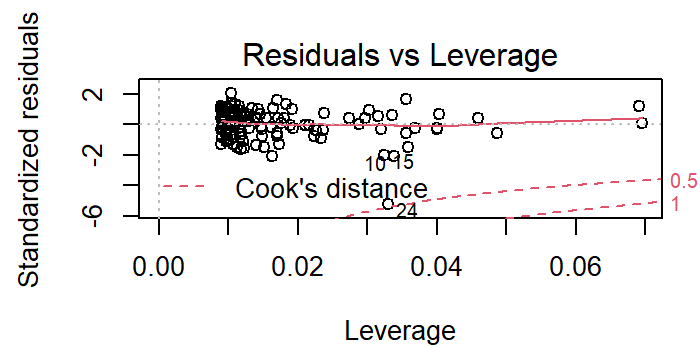
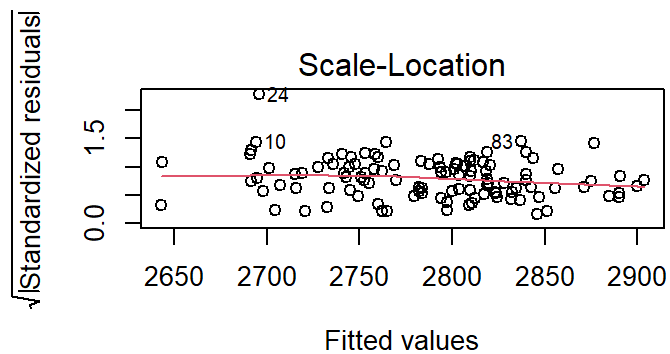
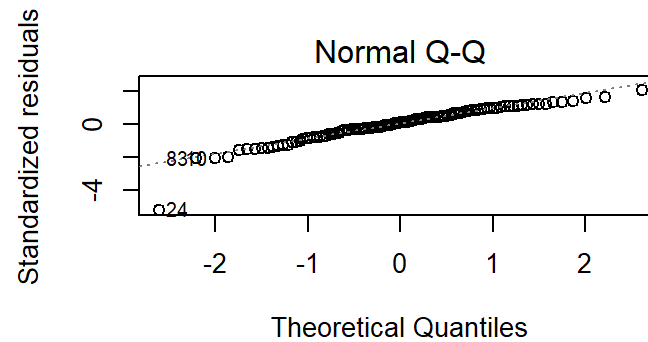
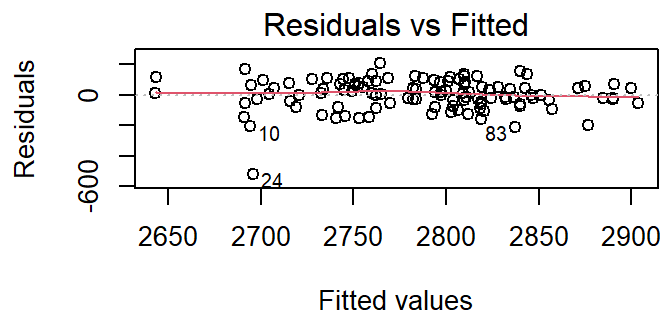
# Opções gráficas

```
summary(fit)
```

```
##
## Call:
## lm(formula = Maxima ~ Nino, data = Tabela)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -518.97  -55.70    8.54   69.35  205.91
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2785.232     9.564  291.224 < 2e-16 ***
## Nino         -58.722    10.131  -5.796 6.57e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 101.2 on 110 degrees of freedom
## Multiple R-squared:  0.234, Adjusted R-squared:  0.227
## F-statistic: 33.59 on 1 and 110 DF, p-value: 6.573e-08
```

# Opções gráficas

```
par(mfrow = c(2,2))  
plot(fit)
```



# ggplot2

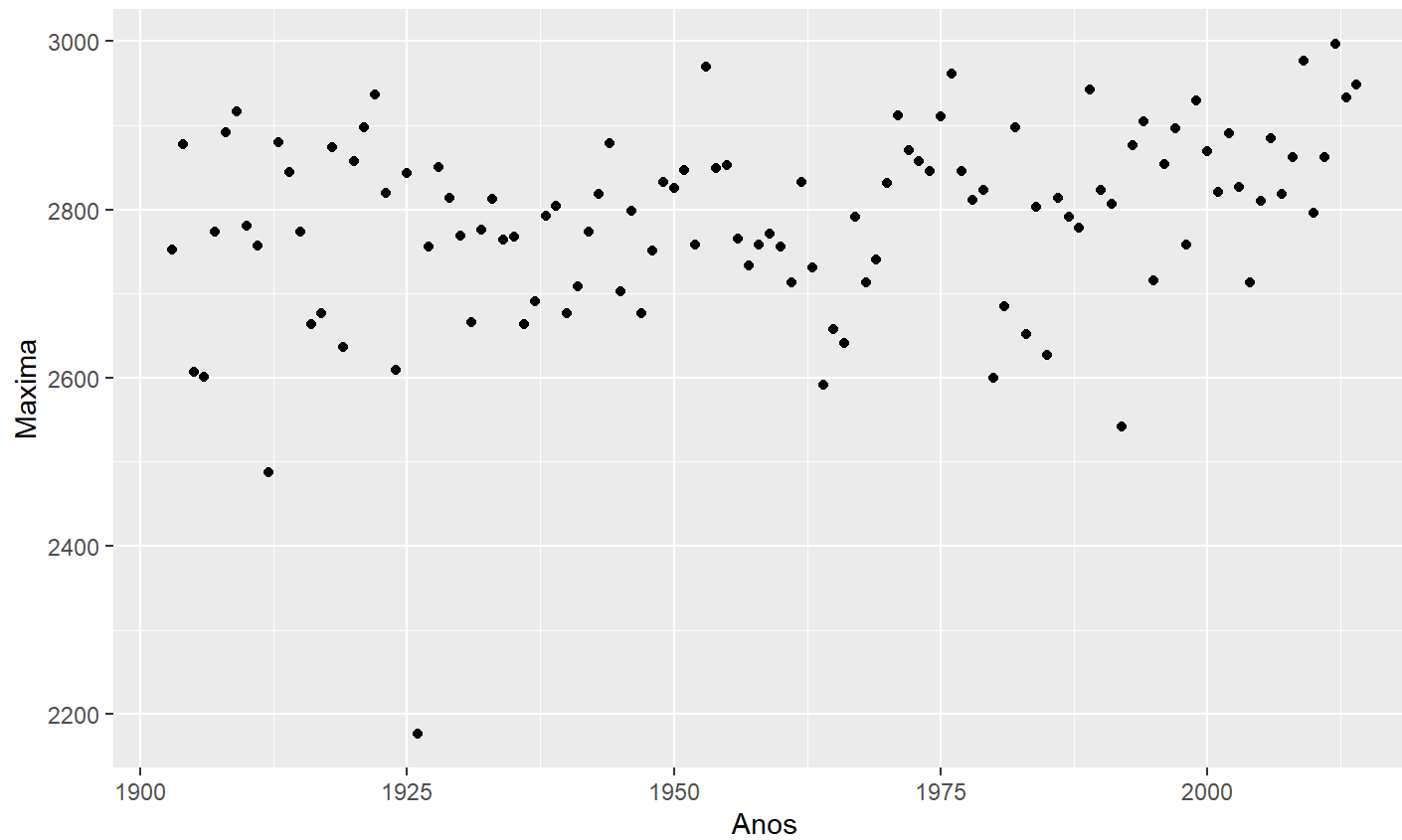
- Pacote bastante popular de gráficos em R
- Materializa o “Grammar of graphics”: abstração de ideias, verbos, substantivos, adjetivos, traduzidos pra linguagem gráfica
- Reduz distância entre pensamento e gráfico
- No popular: Os gráficos ficam mais bonitos com menos esforço

# qplot()

- Função semelhante ao plot, do R
- Contém aesthetics (size, shape, color) e geoms (points, lines)
- Execução simplificada do ggplot2, escondendo opções que serão explicitadas mais adiante
- Função `ggplot()` é o cerne do pacote ggplot2, mas é interessante apresentar o que se pode fazer com a função `qplot()`

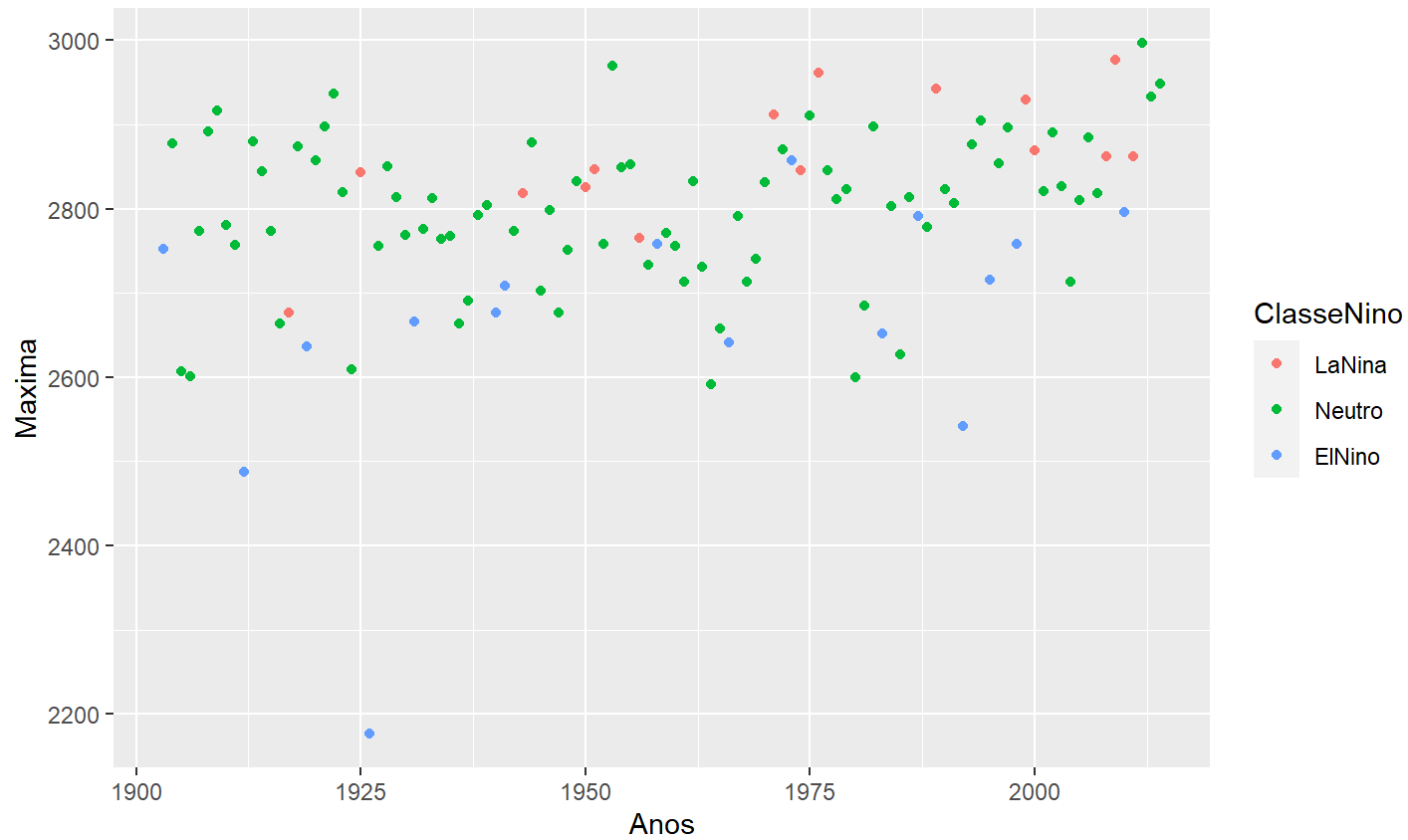
# qplot()

```
library(ggplot2)  
qplot(Anos, Maxima, data = Tabela)
```



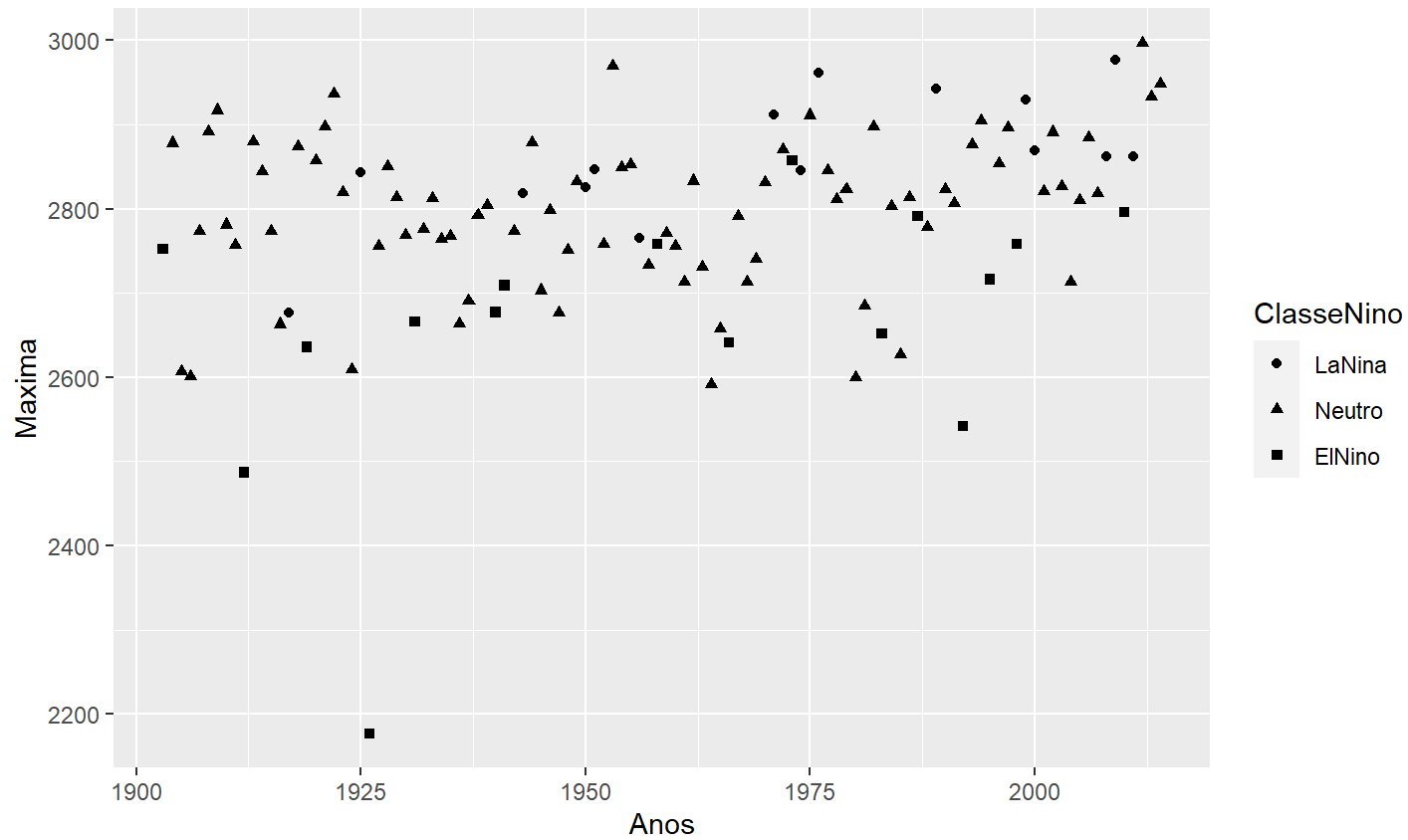
# qplot()

```
qplot(Anos, Maxima, data = Tabela, color = ClasseNino)
```



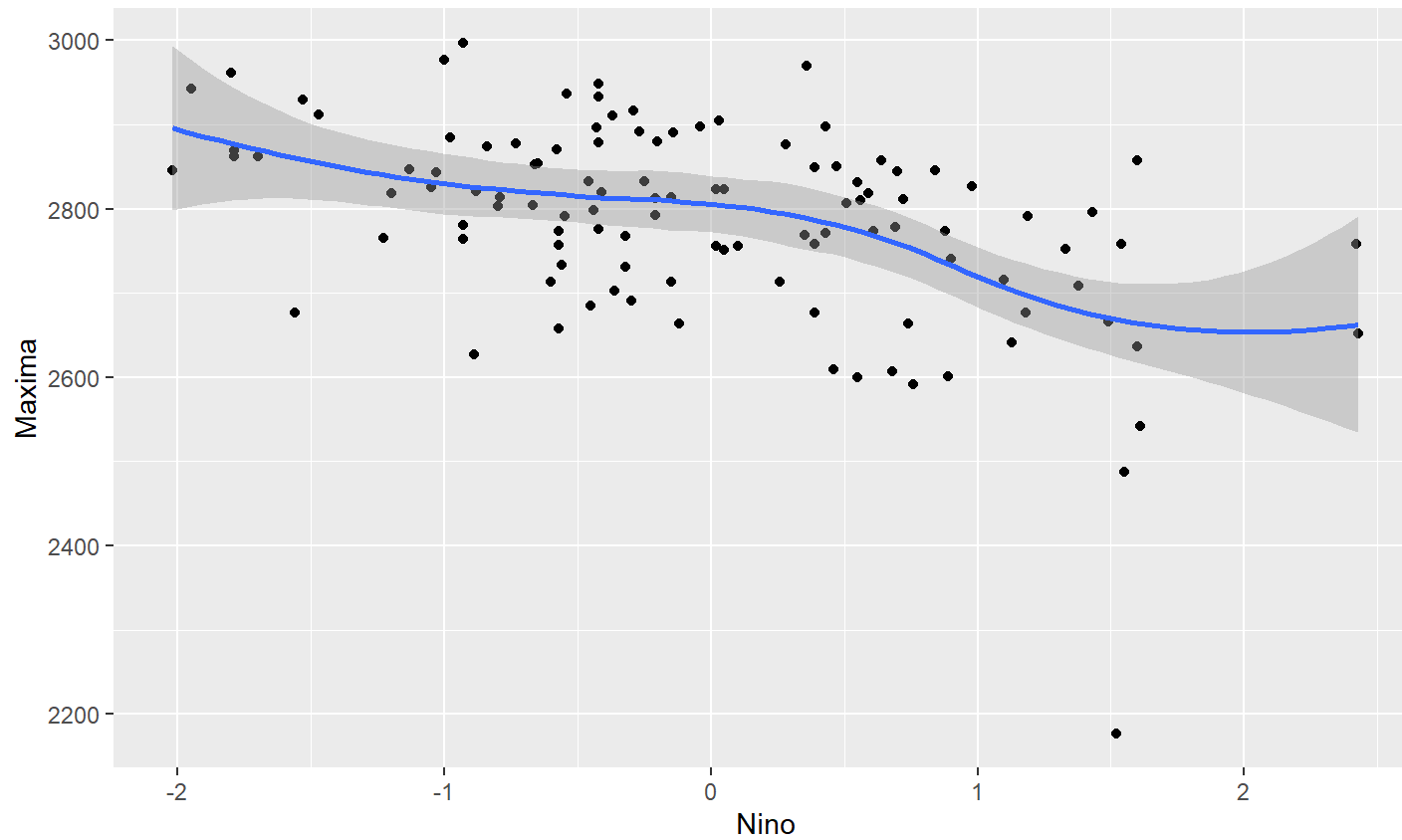
# qplot()

```
qplot(Anos, Maxima, data = Tabela, shape = ClasseNino)
```



# qplot()

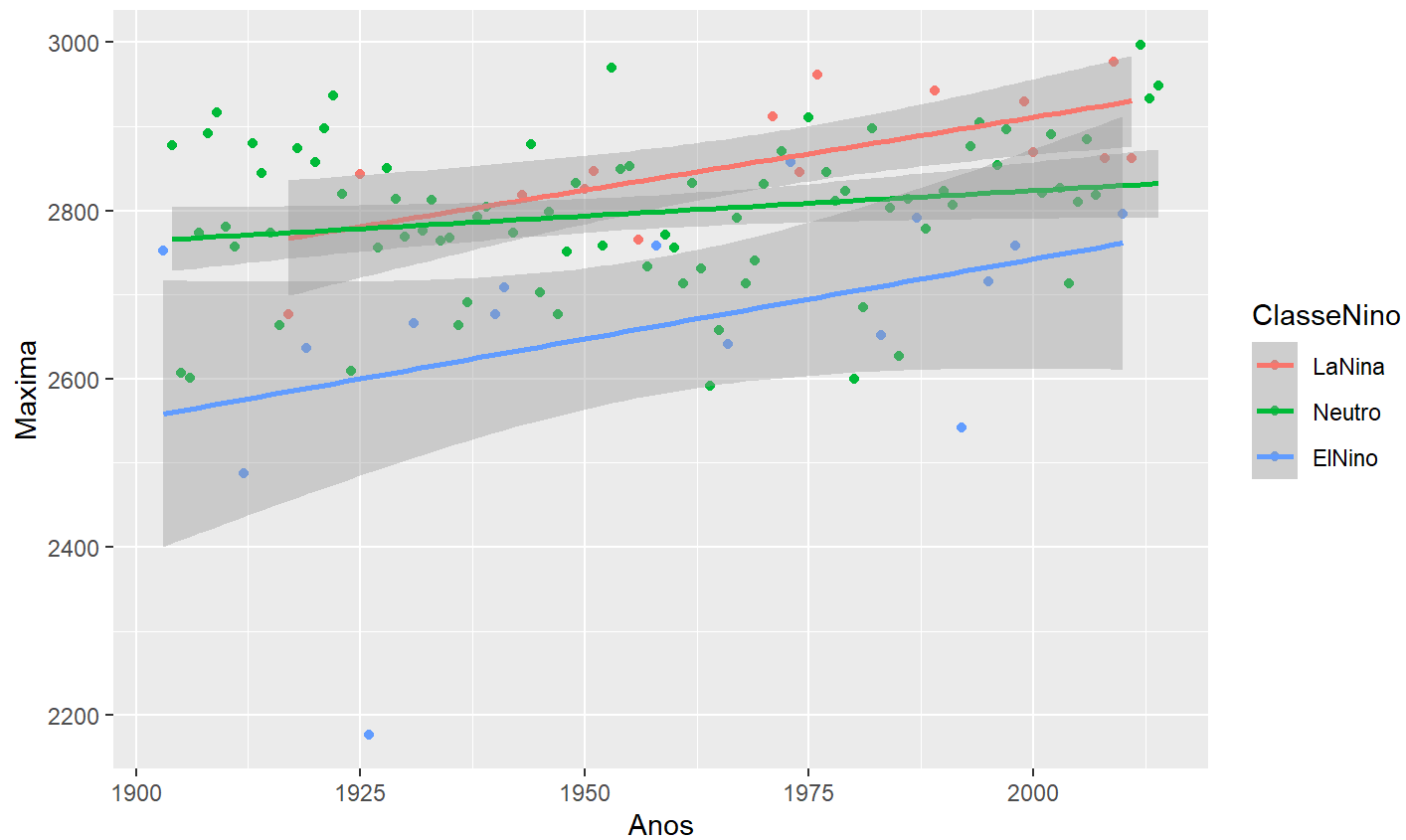
```
qplot(Nino, Maxima, data = Tabela, geom = c("point", "smooth"))
```





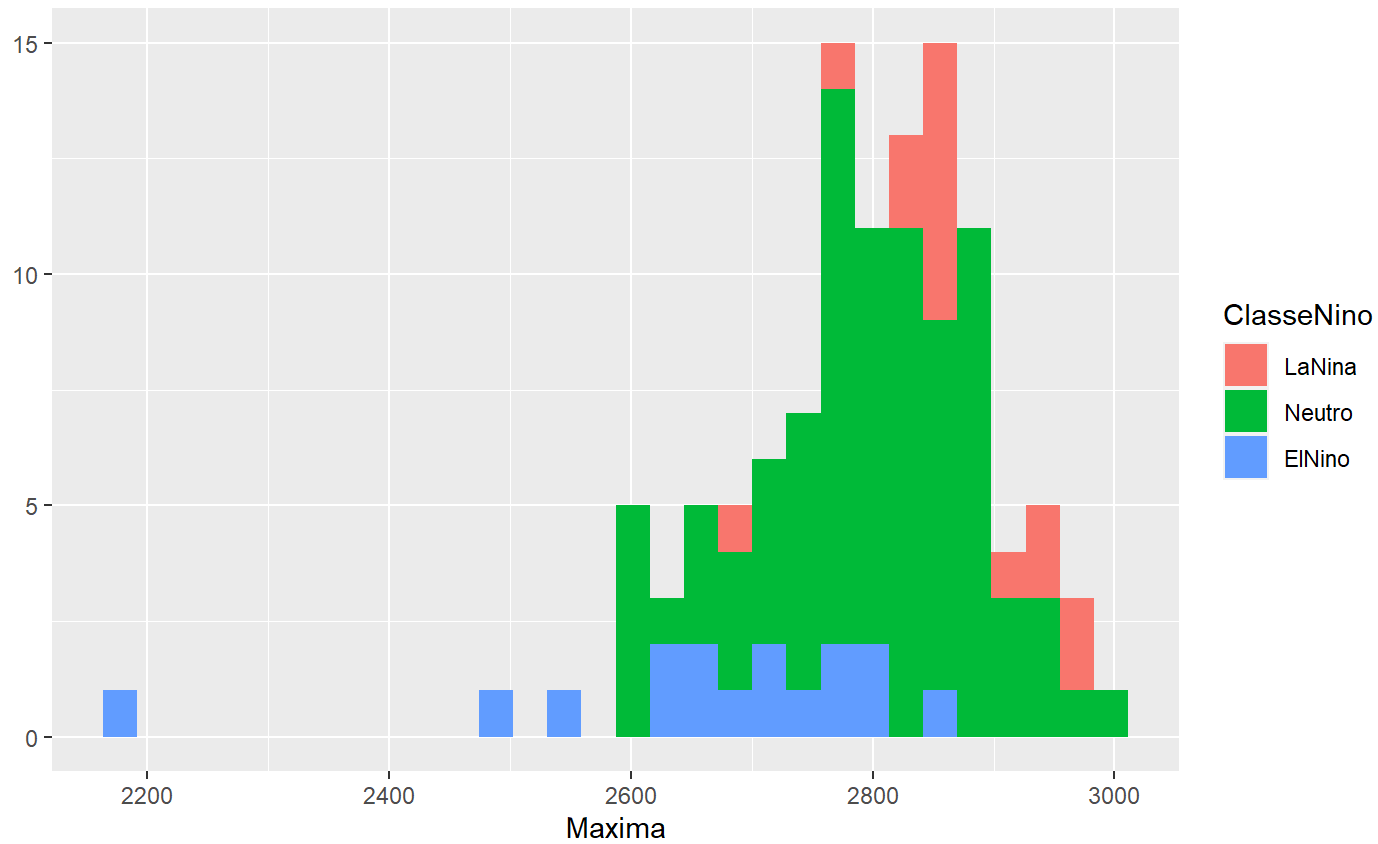
# qplot()

```
qplot(Anos, Maxima, data = Tabela, geom = c("point", "smooth"),  
      method = "lm",  
      color = ClasseNino)
```



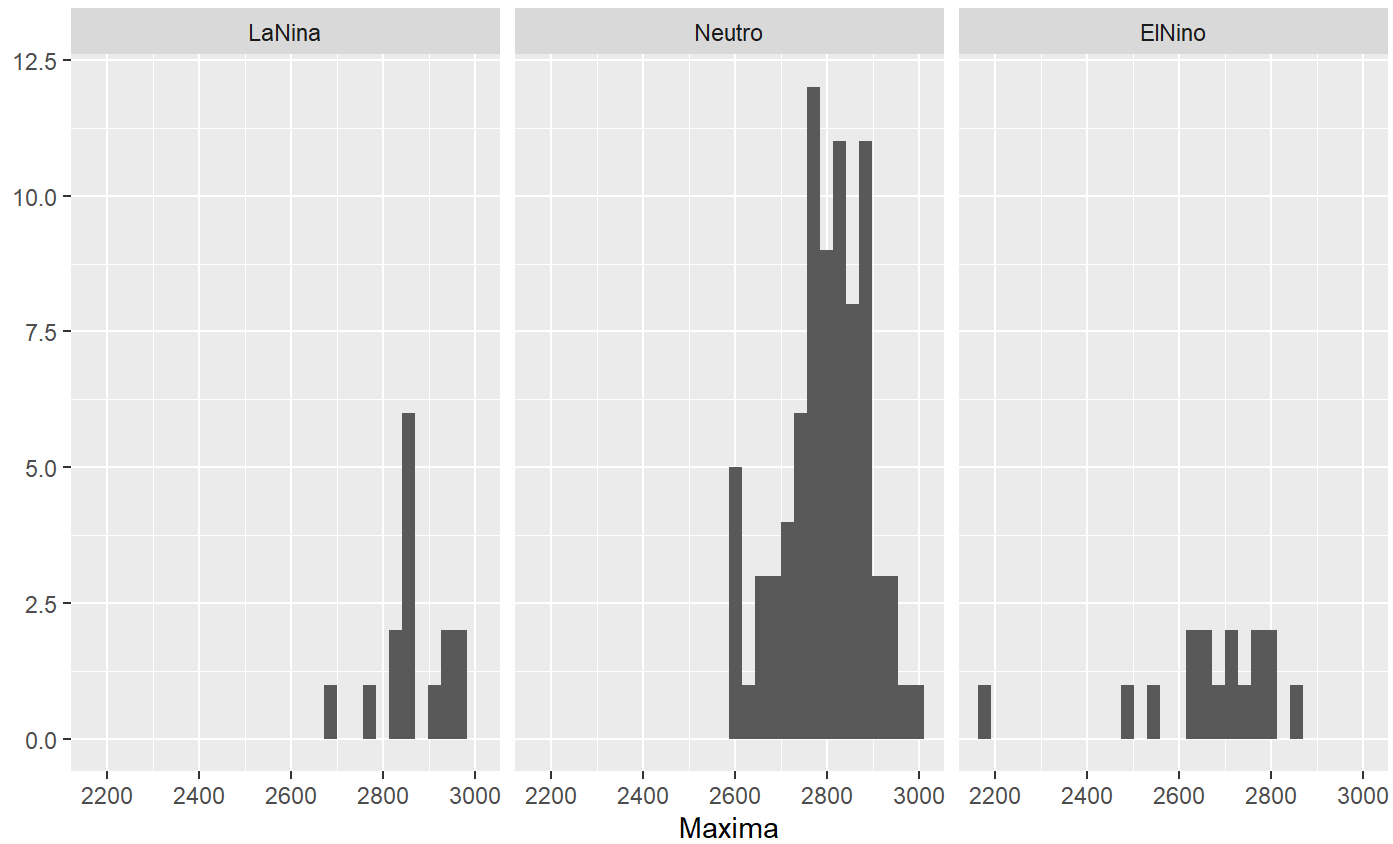
# qplot()

```
qplot(Maxima, data = Tabela, fill = ClasseNino)
```



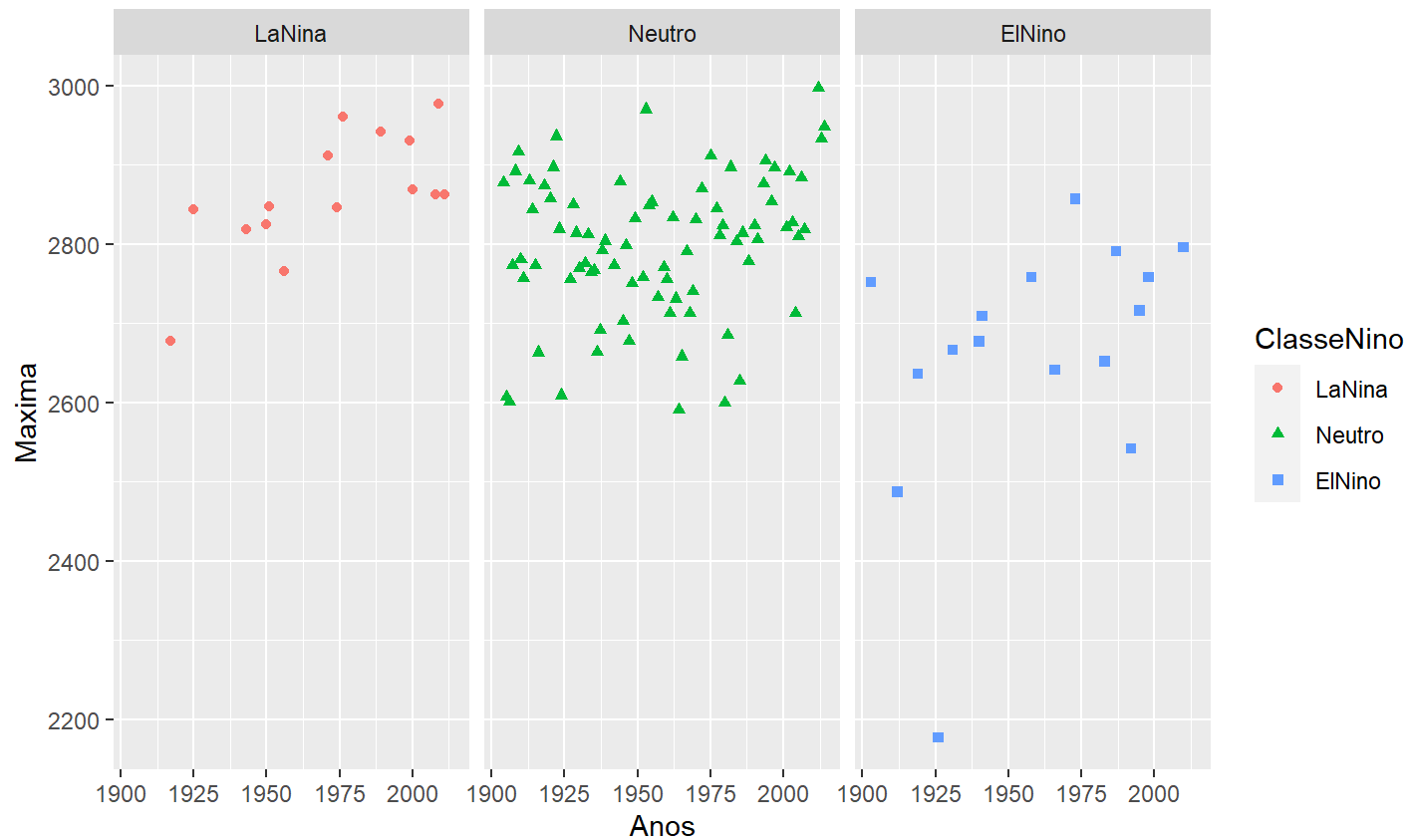
# qplot()

```
qplot(Maxima, data = Tabela, facets = .~ClasseNino, bandwidth = 3)
```



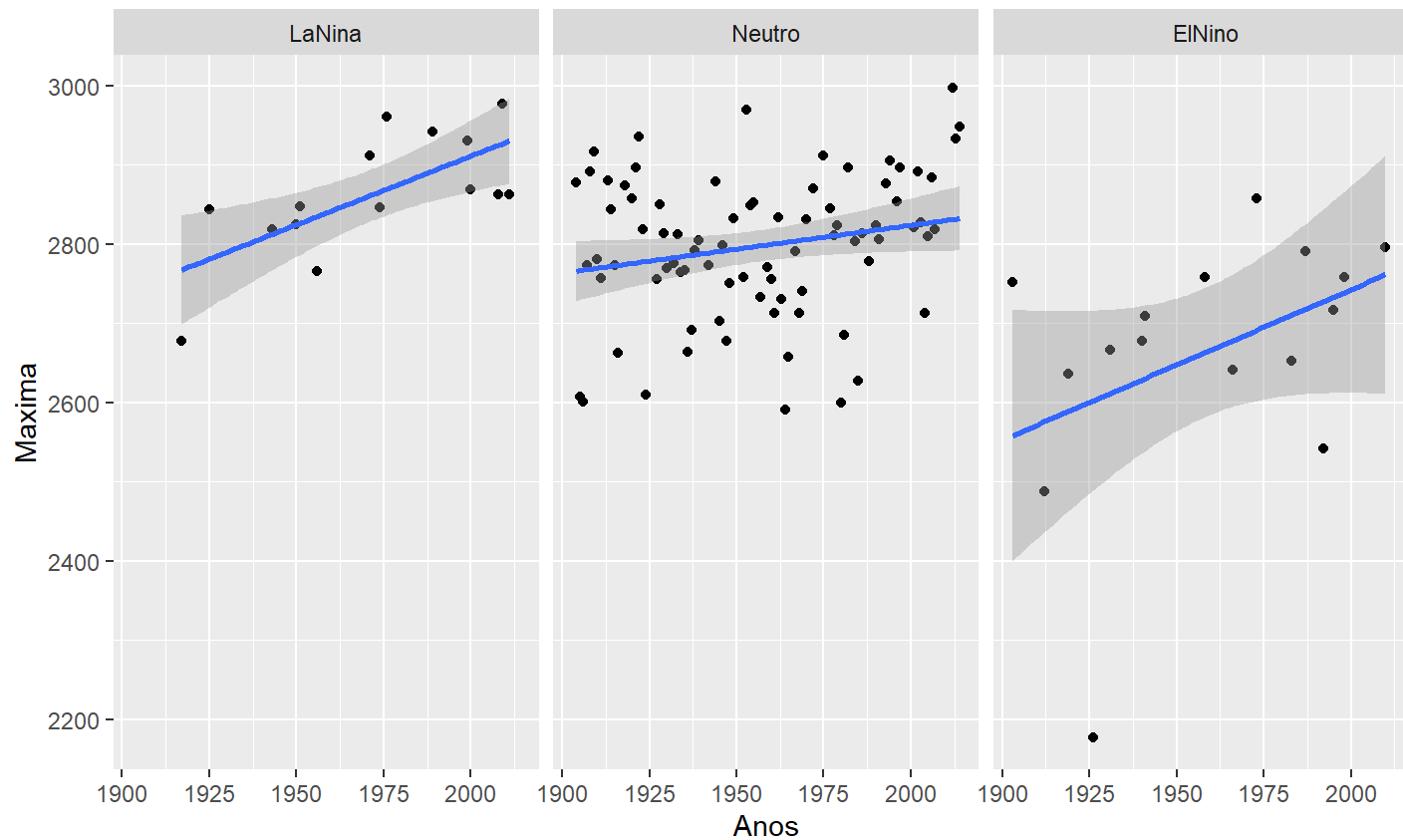
# qplot()

```
qplot(Anos, Maxima, data = Tabela, facets = .~ClasseNino,  
      color = ClasseNino, shape = ClasseNino)
```



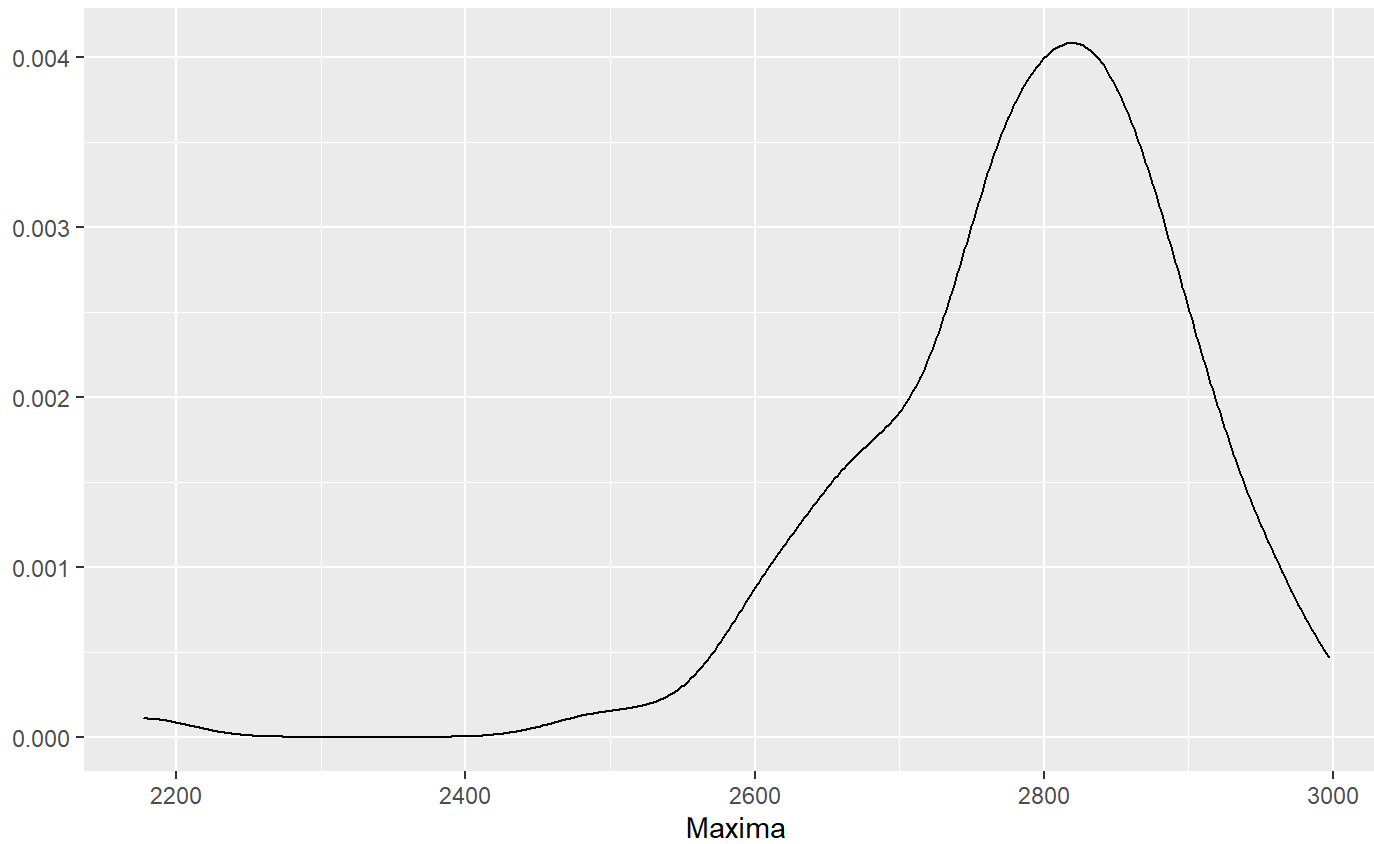
# qplot()

```
qplot(Anos, Maxima, data = Tabela, facets = .~ClasseNino,  
      geom = c("point", "smooth"), method = "lm")
```



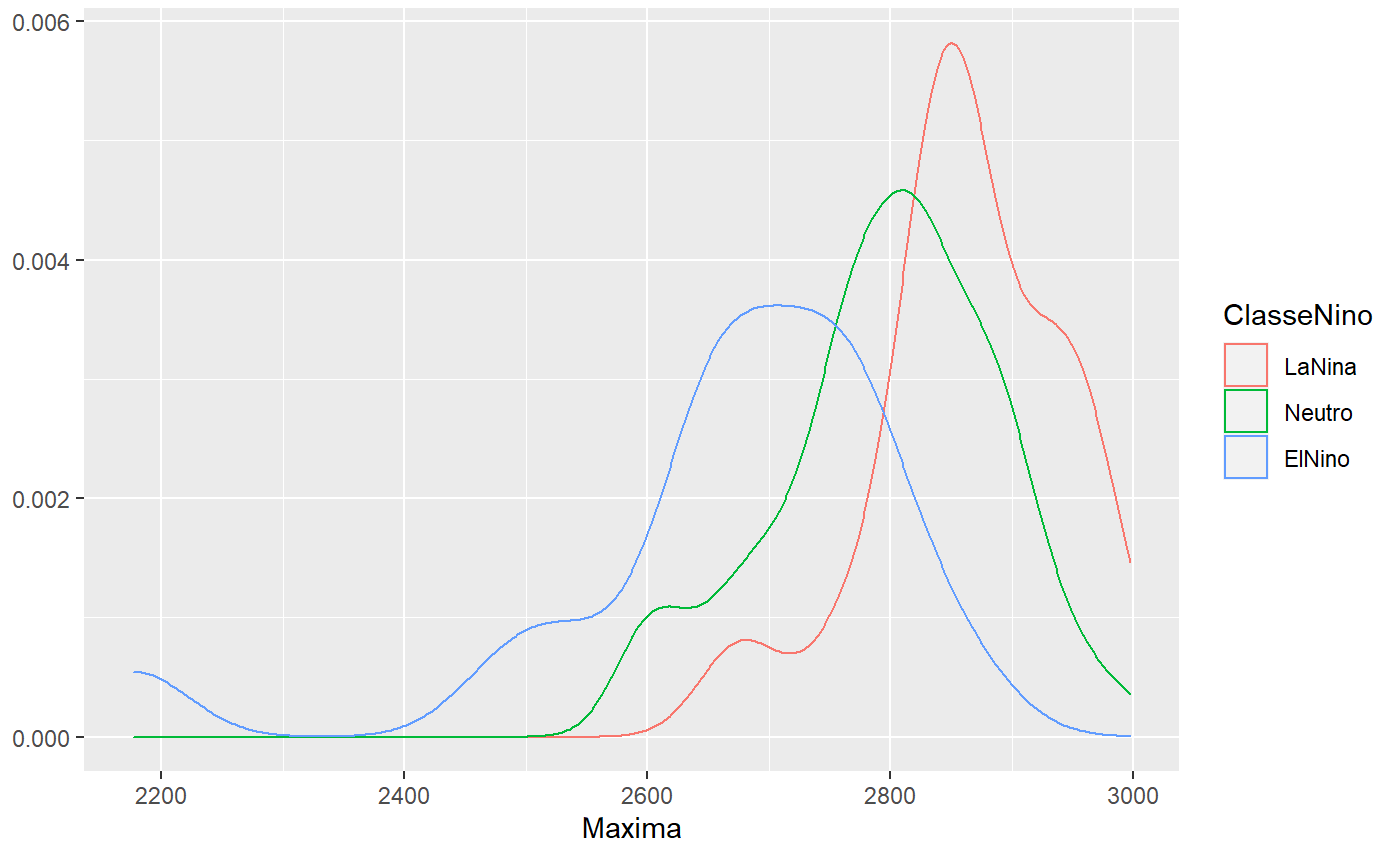
# qplot()

```
qplot(Maxima, data = Tabela, geom = "density")
```



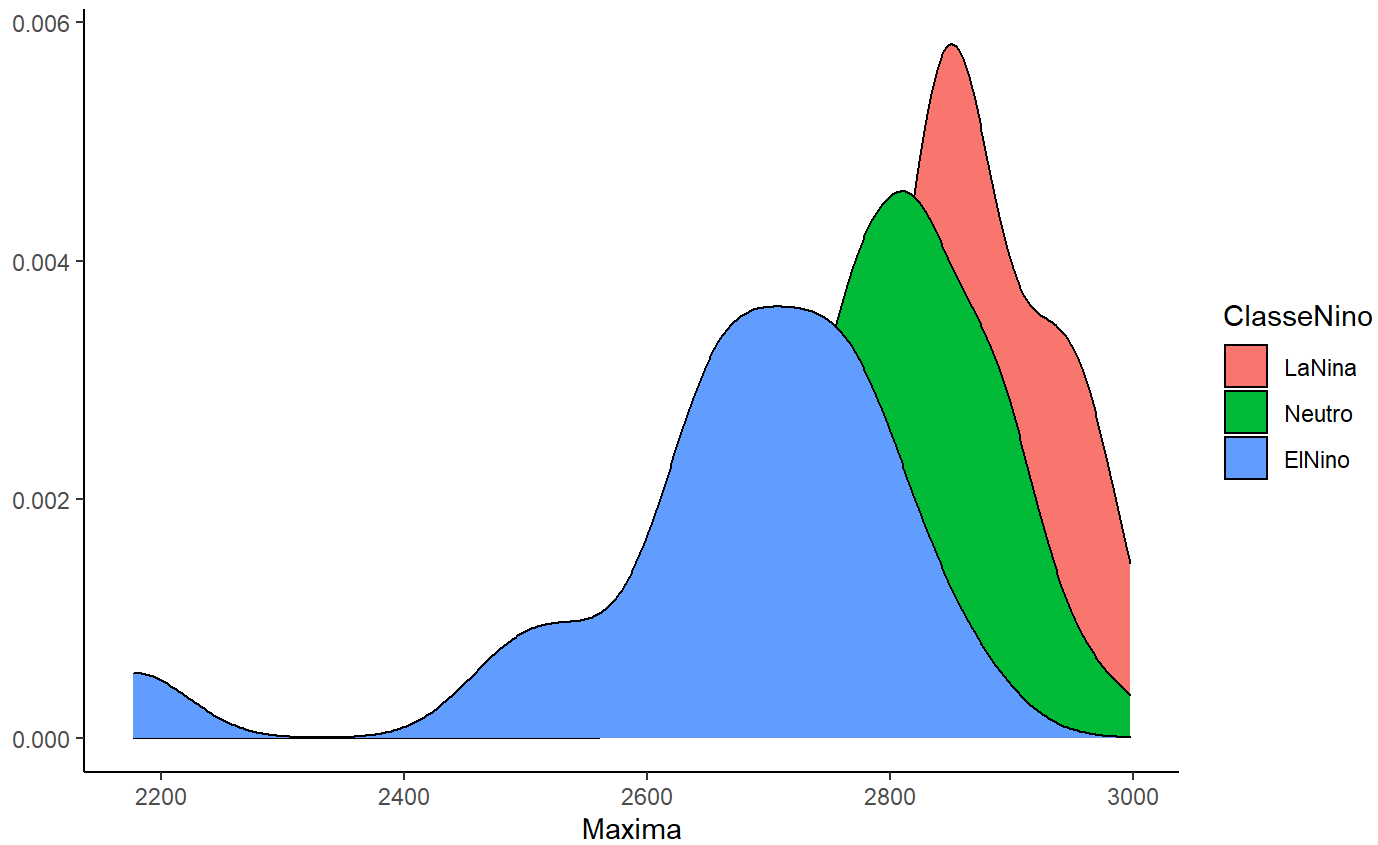
# qplot()

```
qplot(Maxima, data = Tabela, geom = "density", color = ClasseNino)
```



# qplot()

```
qplot(Maxima, data = Tabela, geom = "density", fill = ClasseNino) +  
  theme_classic()
```





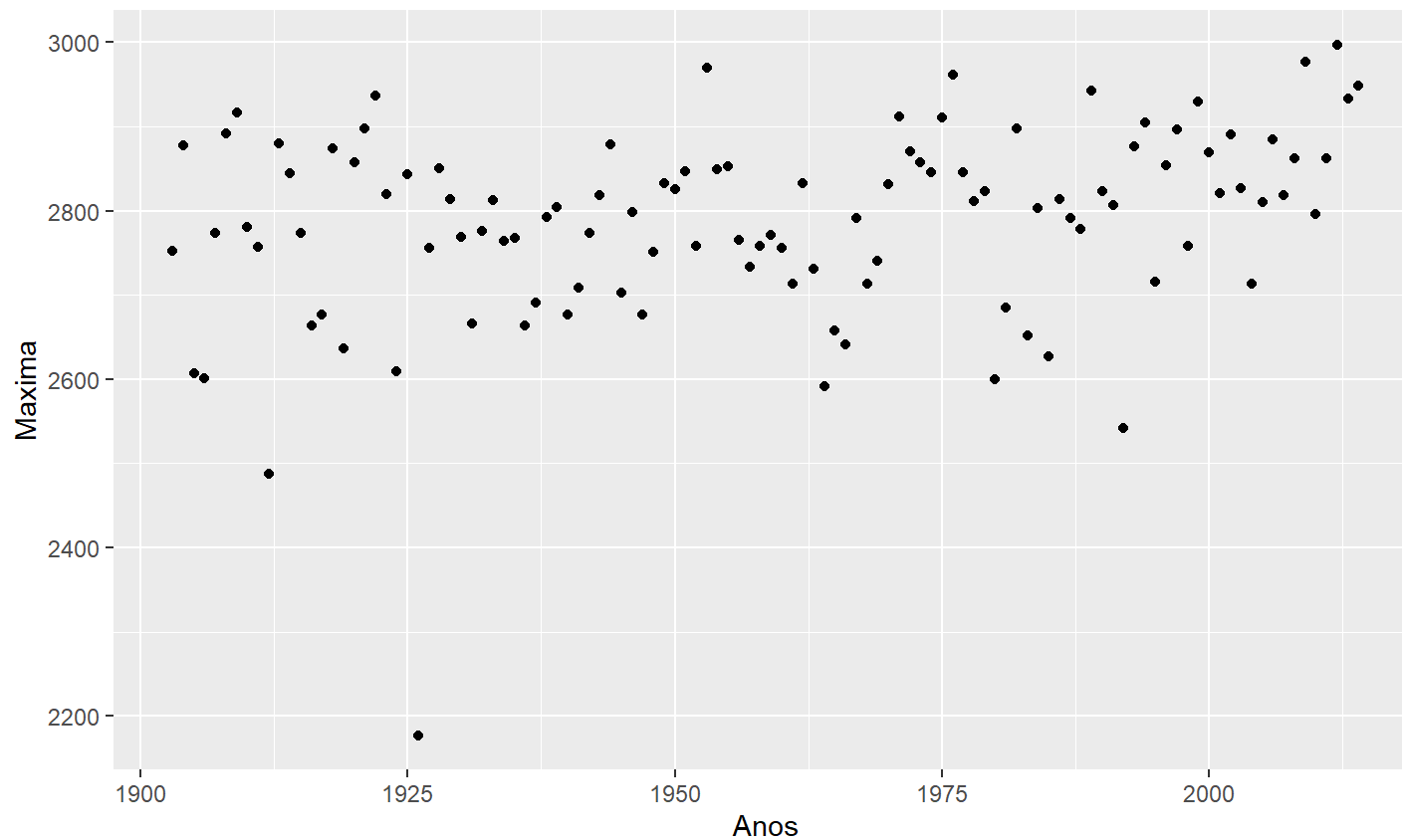
# qplot()

- Vantagens: com comandos simples, já é possível um visual interessante
- Desvantagens: várias opções padrão que nem sempre são customizáveis

# ggplot()

Vamos comparar com o `qplot()`

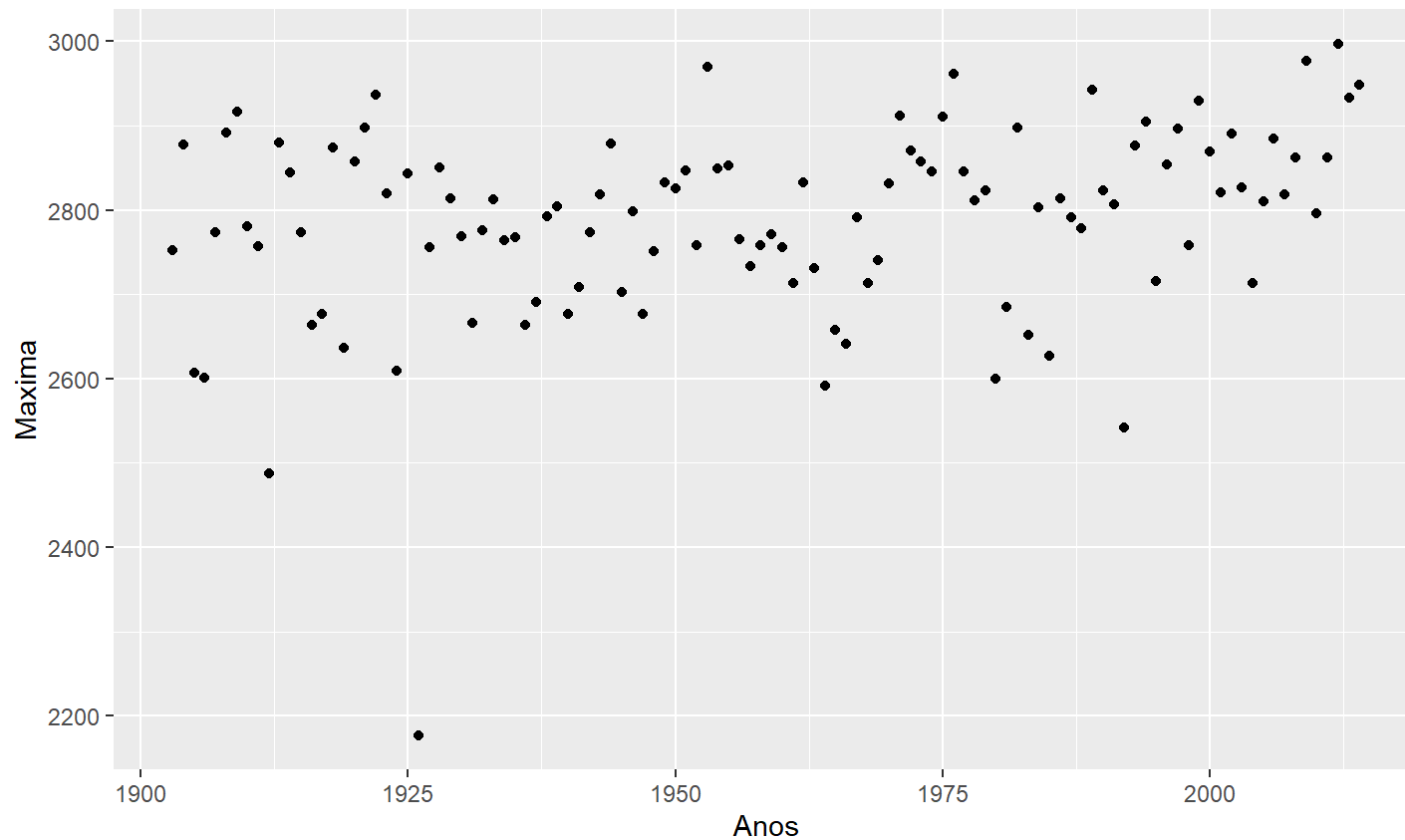
```
qplot(Anos, Maxima, data = Tabela)
```



# ggplot()

ggplot() produz o mesmo gráfico, mas vamos ver como as opções de customização são mais abundantes.

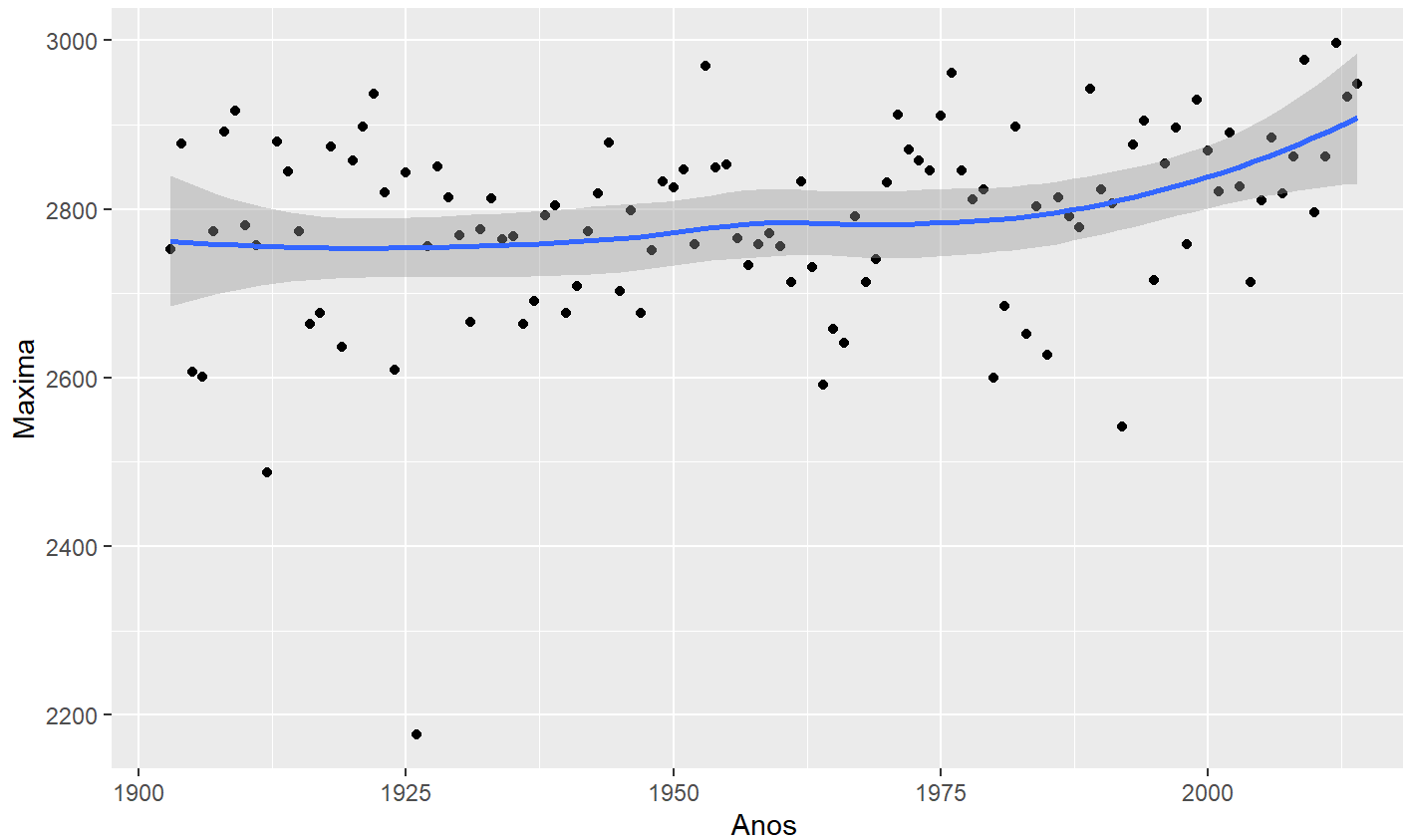
```
p <- ggplot(Tabela, aes(x = Anos, y = Maxima))  
p + geom_point()
```



# ggplot()

ggplot() produz o mesmo gráfico, mas vamos ver como as opções de customização são mais abundantes.

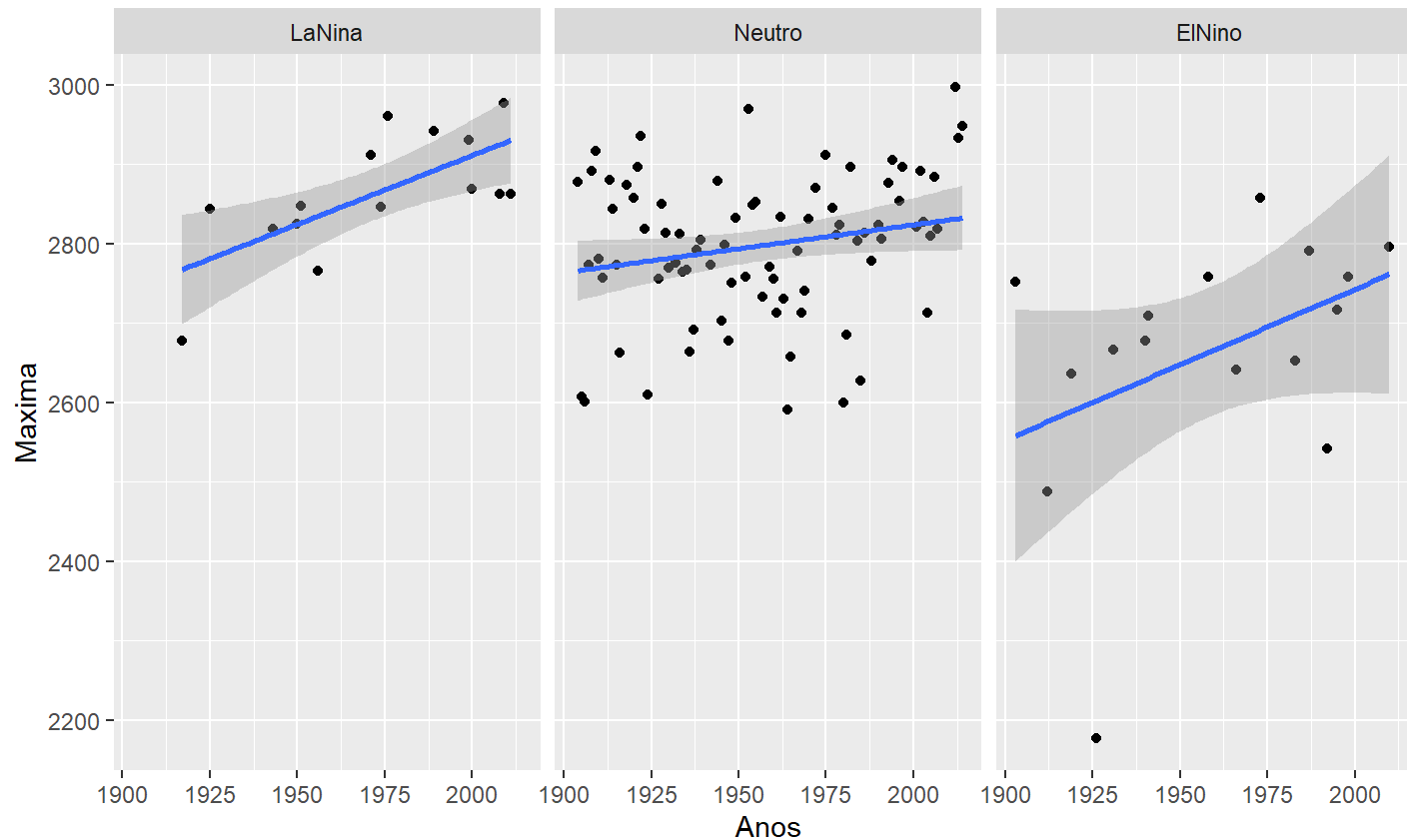
```
p + geom_point() + geom_smooth()
```



# ggplot()

ggplot() produz o mesmo gráfico, mas vamos ver como as opções de customização são mais abundantes.

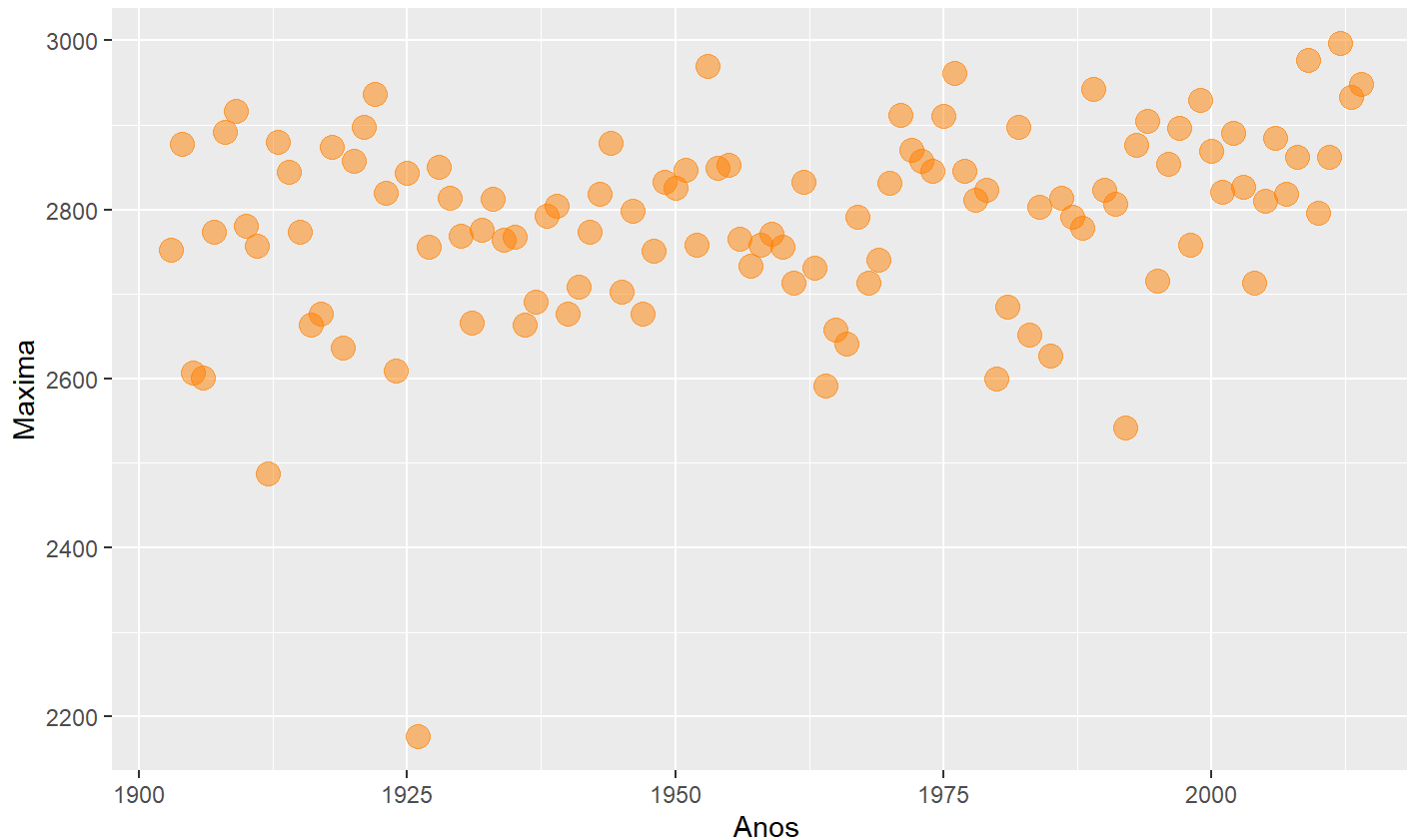
```
p + geom_point() + geom_smooth(method = "lm") + facet_grid(. ~ ClasseNino)
```



# ggplot()

ggplot() produz o mesmo gráfico, mas vamos ver como as opções de customização são mais abundantes.

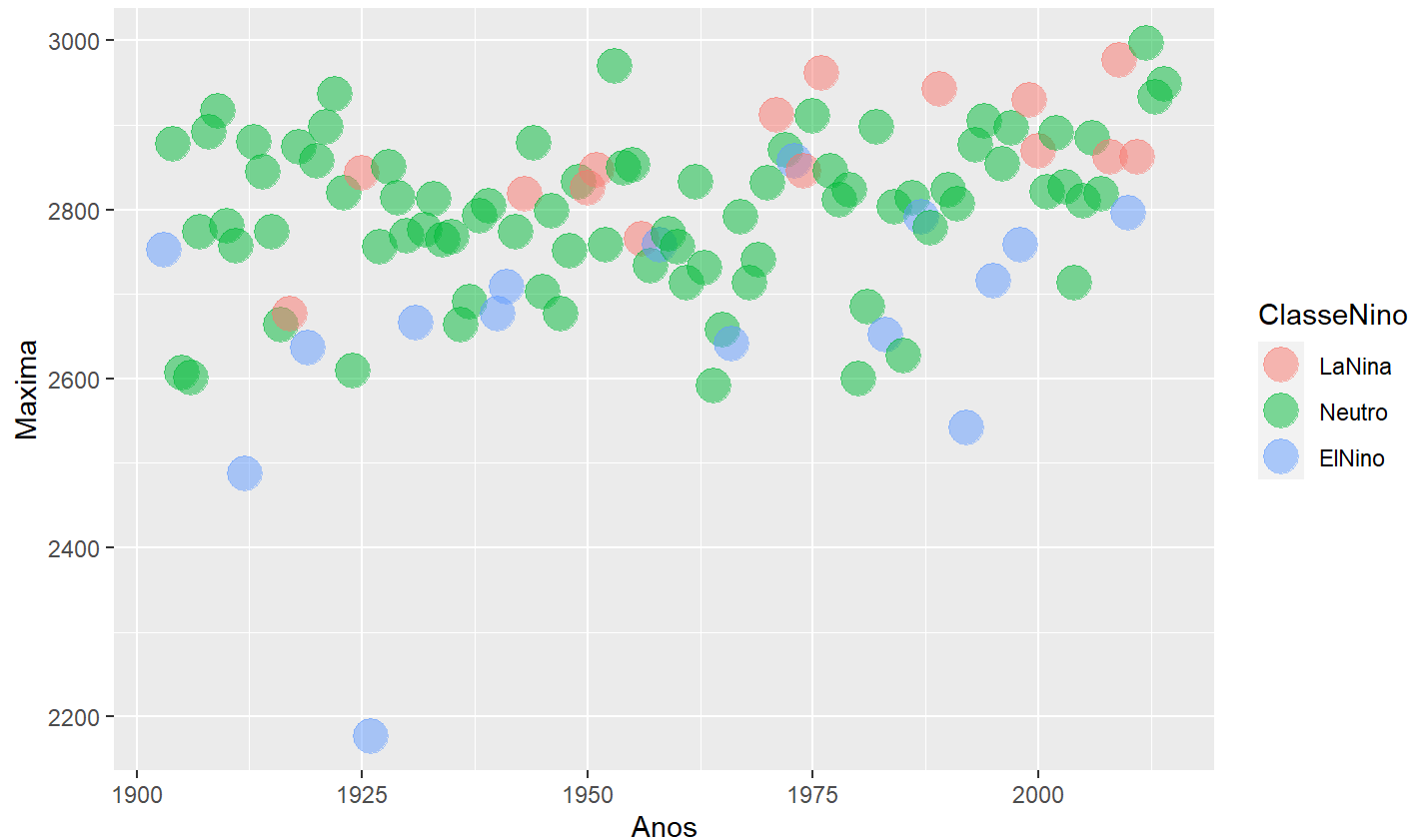
```
p + geom_point(color = rgb(1,0.5,0), size = 4, alpha = 0.5)
```



# ggplot()

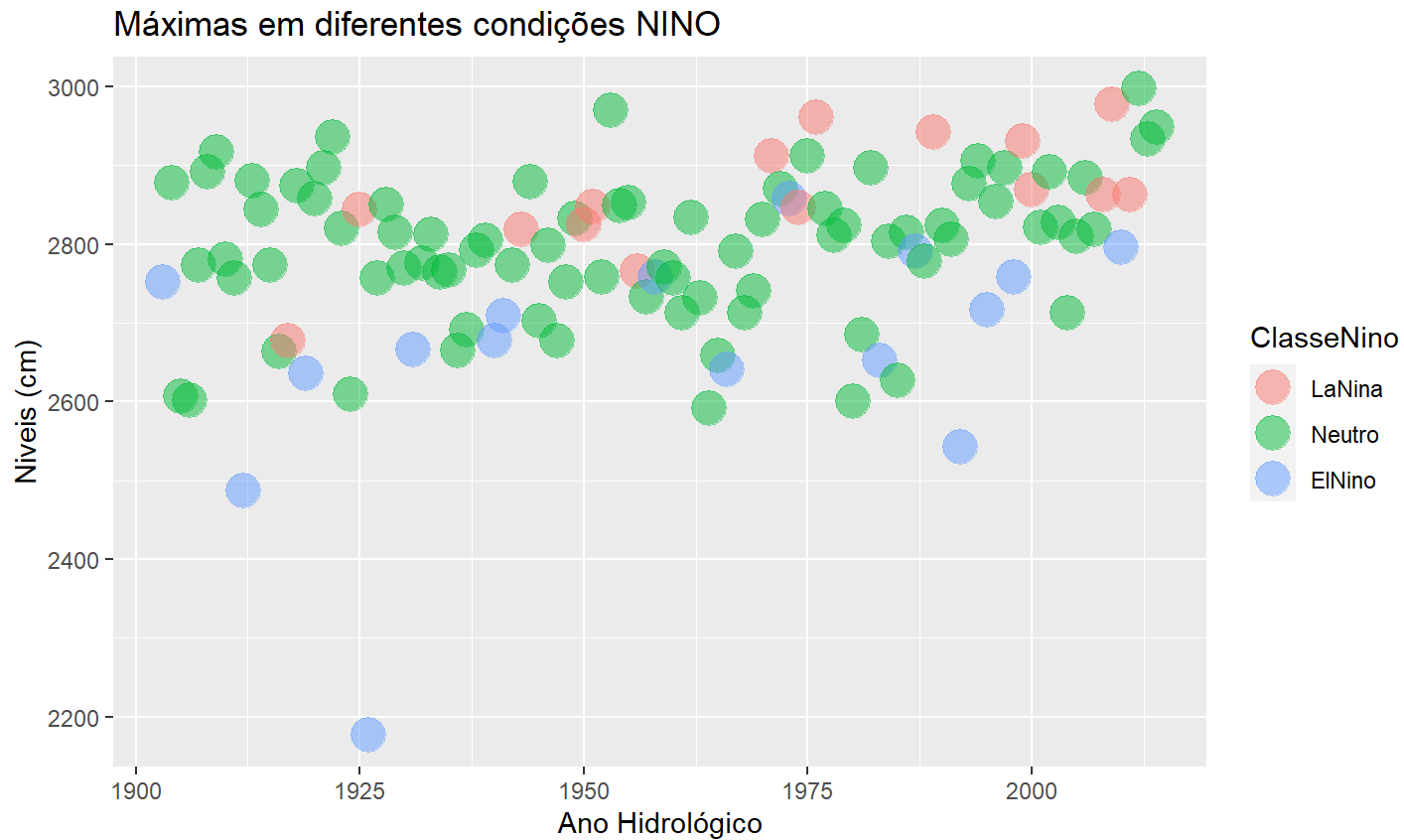
ggplot() produz o mesmo gráfico, mas vamos ver como as opções de customização são mais abundantes.

```
p + geom_point(aes(color = ClasseNino), size = 6, alpha = 0.5)
```



# ggplot()

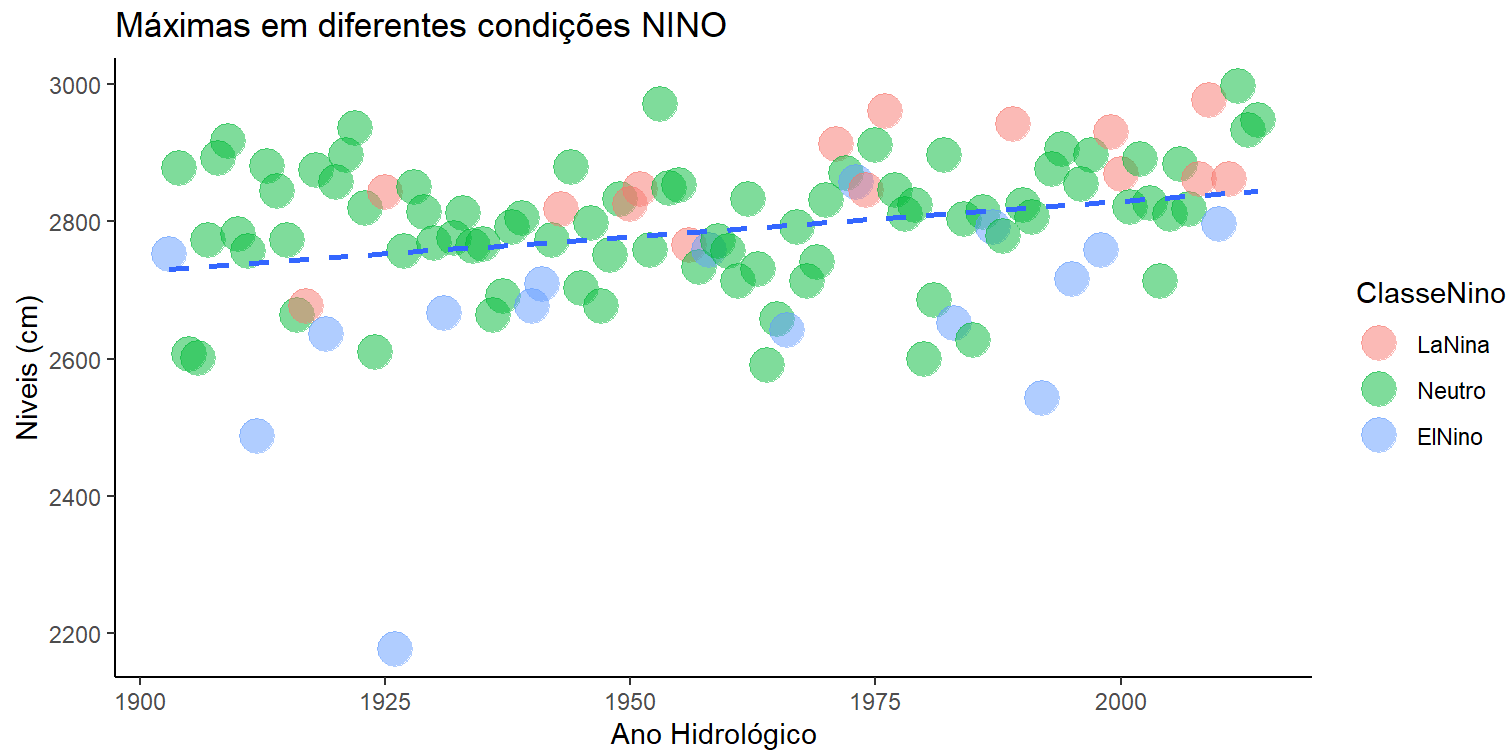
```
p + geom_point(aes(color = ClasseNino), size = 6, alpha = 0.5) +  
  labs(x = "Ano Hidrológico", y = "Niveis (cm)", title = "Máximas em diferentes condições NINO")
```





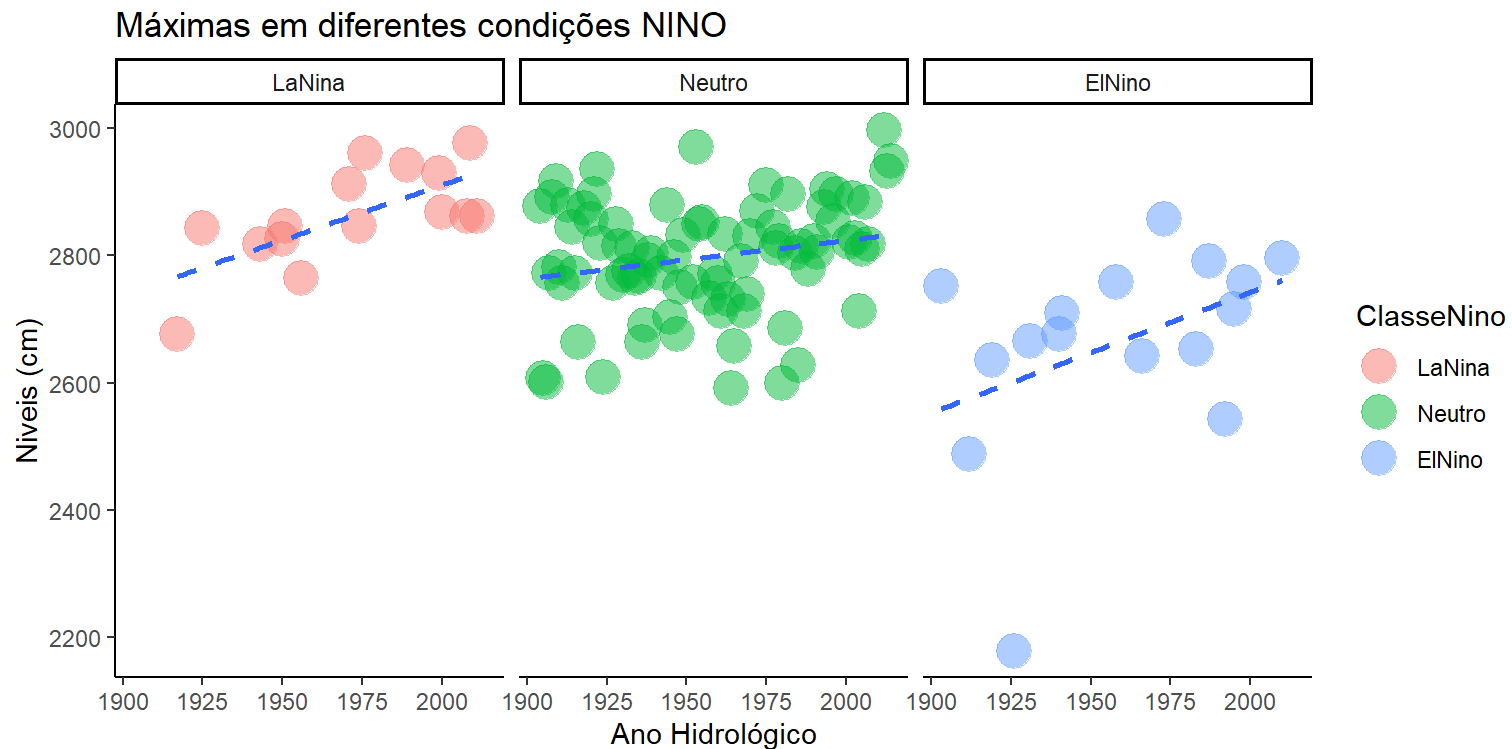
# ggplot()

```
p + geom_point(aes(color = ClasseNino), size = 6, alpha = 0.5) +  
  labs(x = "Ano Hidrológico", y = "Niveis (cm)", title = "Máximas em diferentes condições NINO") +  
  geom_smooth(linetype = 2, method = "lm", se = FALSE) +  
  theme_classic()
```



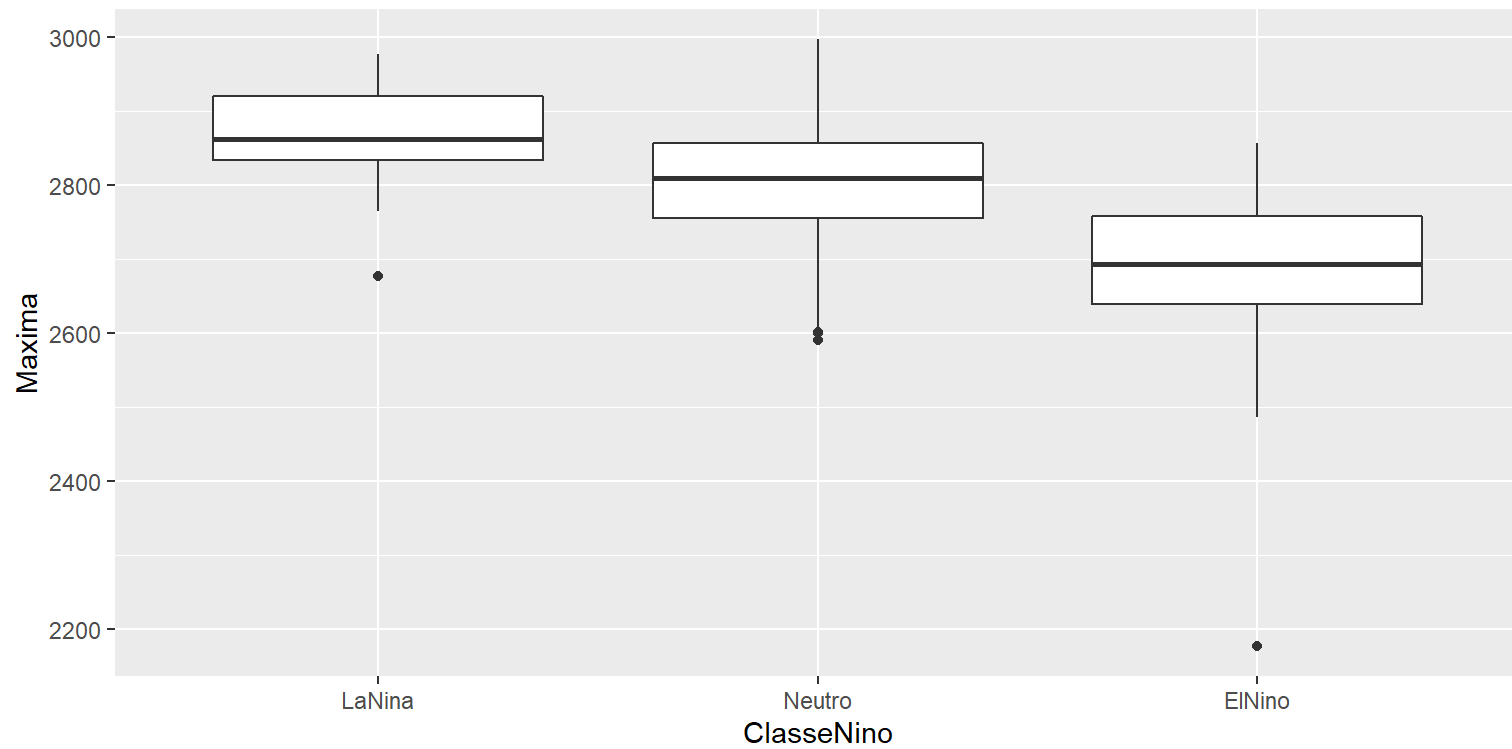
# ggplot()

```
p + geom_point(aes(color = ClasseNino), size = 6, alpha = 0.5) +  
  labs(x = "Ano Hidrológico", y = "Niveis (cm)", title = "Máximas em diferentes condições NINO") +  
  geom_smooth(linetype = 2, method = "lm", se = FALSE) +  
  facet_grid(. ~ ClasseNino) +  
  theme_classic()
```



# ggplot()

```
p <- ggplot(Tabela, aes(ClasseNino, Maxima))  
p + geom_boxplot()
```



# ggplot()

```
p <- ggplot(Tabela, aes(ClasseNino, Maxima))  
p + geom_boxplot(outlier.shape = NA, aes(fill = ClasseNino)) + geom_jitter(width = 0.5) +  
  labs(x = "Ano Hidrológico", y = "Niveis (cm)", title = "Máximas em diferentes condições NINO") +  
  facet_grid(. ~ ClasseNino) +  
  theme_dark()
```

