MAD-CB

Visualização dos Dados – Entender as Variáveis

John Tukey

The simple graph has brought more information to the data analyst's mind than any other device.

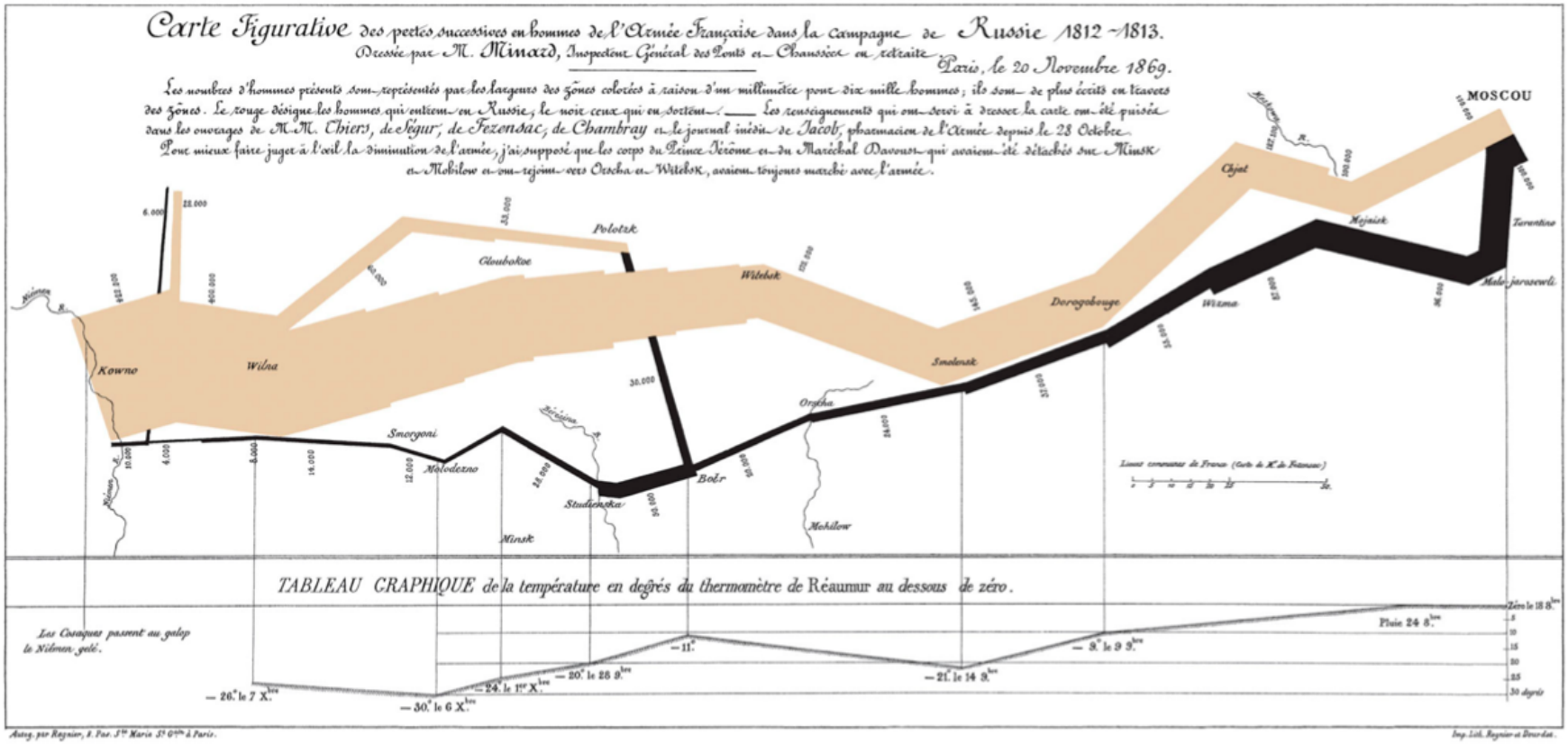
O gráfico básico tem trazido mais informações para a mente do analista de dados que qualquer outro dispositivo.

O Que Fazem os Gráficos

- Mostra comparação entre variáveis quantitativas em termos da distribuição delas
  - Ponto central
  - Dispersão
- Quer transmitir a quantia certa das informações
  - Suficiente para comunicar padrões e tendências nos dados
  - Não tanto que interpretação é difícil
- Mostrar distribuições
  - Dispersão de uma variável
  - Como os valores da variável são distribuídos
  - Tb, em termos dos vários níveis das variáveis categóricas

Gráfico Historico - Marcha de Napoleão para Moscovo

- Charles Joseph Minard, engenheiro francês, 1869



- Mostra seis tipos de informação em um gráfico

Gráficos Estatísticos com `ggplot2`

- *gg* - Grammar of Graphics
- 5 Tipos de Gráficos
  - Gráfico de dispersão
  - Gráfico de linha
  - Histograma
  - Boxplot
  - Gráfico de barra

Gráficos de Dispersão - "*Scatter Plots*"

```
testes <- read_csv("pac_demo.csv") %>%
          mutate(logcv = log10(copias_cv)) %>%
          select(c(codepac, logcv, contagem_cd4, contagem_cd8))
```

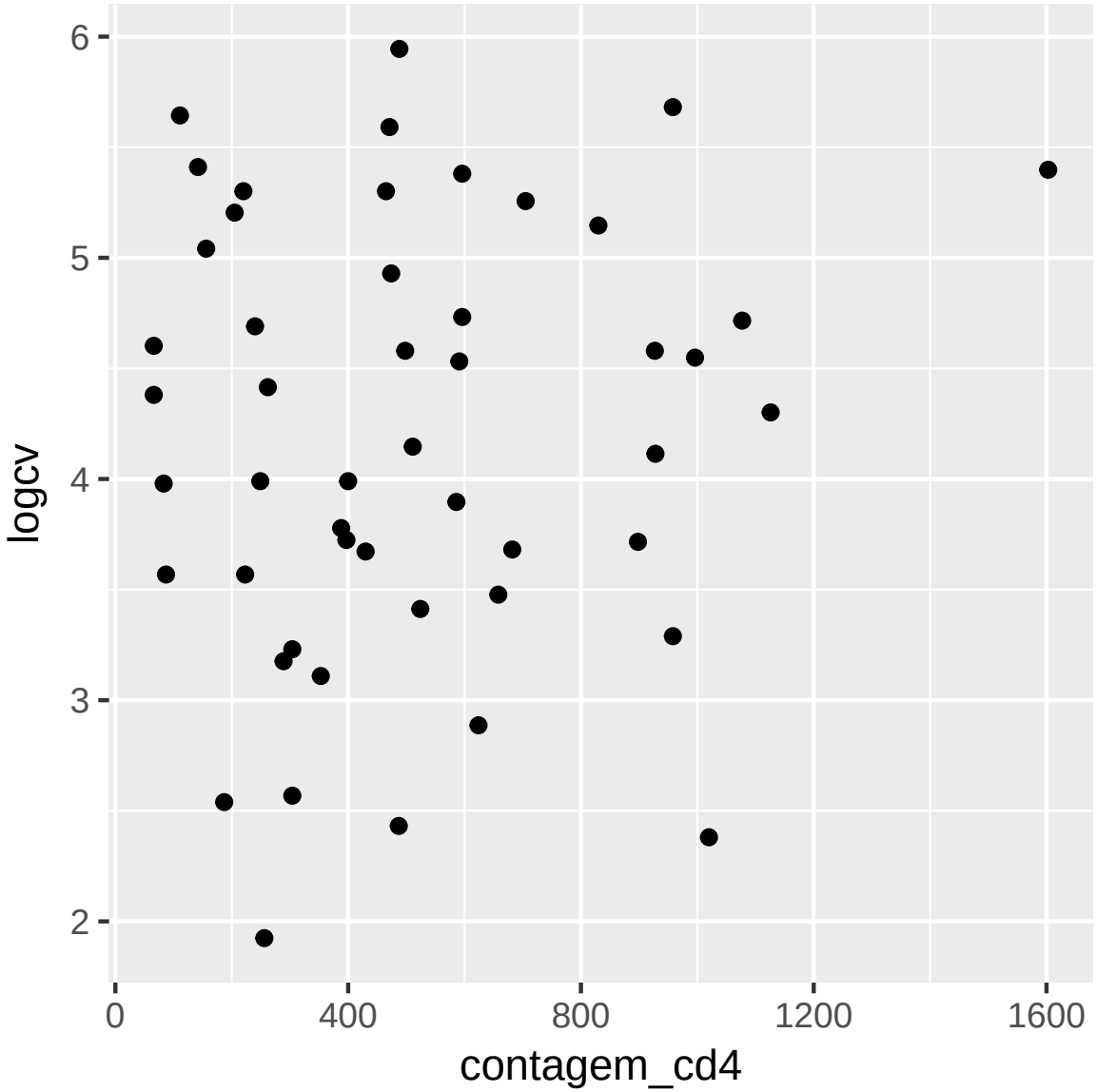
- Relação entre 2 variáveis continuas
- Usar os dados de testes dos pacientes HIV
- Comparar carga viral (`copias_cv`) e contagem de células CD4+ T (`contagem_cd4`)
  - Carga viral transformada em log_{10}
- Dados armazenados em `testes`

Construindo um Gráfico Simples

- Dar um nome para gráfico
- Indicar no `ggplot` qual tibble ou data.frame vai usar
- Indicar quais variáveis você quer nos eixos x e y (no bloco `aes`)
- Indicar qual tipo de gráfico você quer
  - Chamado em `ggplot` um `geom`
  - No caso de um gráfico de dispersão – `geom_point`

nome <- ggplot(data = tibble ou df, aes(x = var, y = var)) + geom_point()

```r
cvcd4 <- ggplot(data = testes, aes(x = contagem_cd4, y = logcv )) + geom_point()
cvcd4  # Só precisar chamar o gráfico para mostrar ele
```



Outro Exemplo – CD4+ vs. CD8+

- Vocês me mostram como escrever o código

Outro Exemplo – CD4+ vs. CD8+

- Vocês me mostram como escrever o código
- Nome: cd4cd8

Outro Exemplo – CD4+ vs. CD8+

- Vocês me mostram como escrever o código
- Nome: cd4cd8
- *cd4cd8 <- ggplot()*

Outro Exemplo – CD4+ vs. CD8+

- Vocês me mostram como escrever o código
- Nome: cd4cd8
- *cd4cd8 <- ggplot()*
- Data = testes

Outro Exemplo – CD4+ vs. CD8+

- Vocês me mostram como escrever o código
- Nome: cd4cd8
- *cd4cd8 <- ggplot()*
- Data = testes
- *cd4cd8 <- ggplot(data = testes)*

Outro Exemplo – CD4+ vs. CD8+

- Vocês me mostram como escrever o código
- Nome: cd4cd8
- *cd4cd8 <- ggplot()*
- Data = testes
- *cd4cd8 <- ggplot(data = testes)*
- bloco `aes()`

Outro Exemplo – CD4+ vs. CD8+

- Vocês me mostram como escrever o código
- Nome: cd4cd8
- *cd4cd8 <- ggplot()*
- Data = testes
- *cd4cd8 <- ggplot(data = testes)*
- bloco `aes()`
- *cd4cd8 <- ggplot(data = testes, aes())*

- x = contagem_cd4

- x = contagem_cd4
- *cd4cd8 <- ggplot(data = testes, aes(x = contagem_cd4))*

- x = contagem_cd4
- *cd4cd8 <- ggplot(data = testes, aes(x = contagem_cd4))*
- y = contagem_cd8

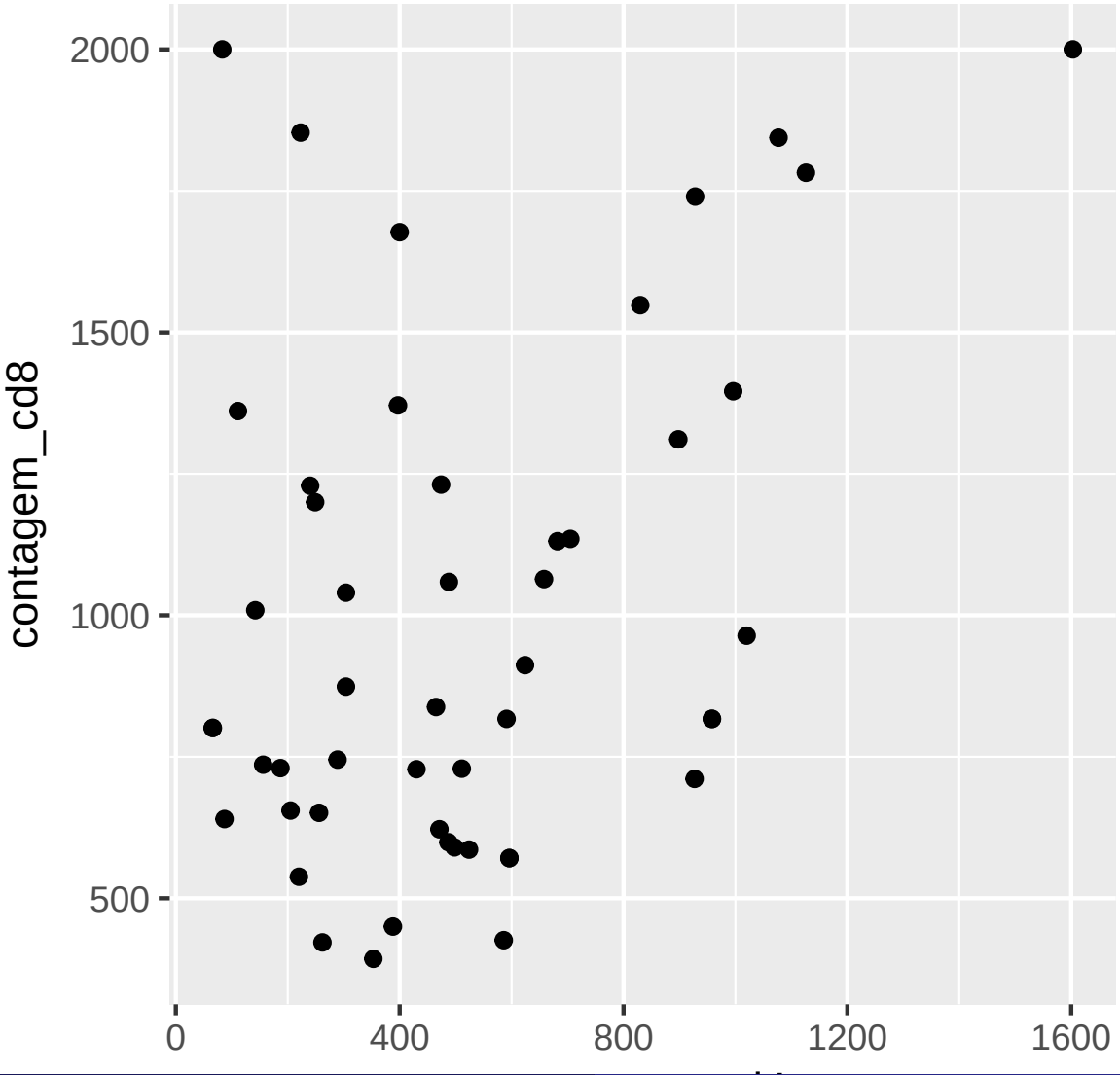
- x = contagem_cd4
- *cd4cd8 <- ggplot(data = testes, aes(x = contagem_cd4))*
- y = contagem_cd8
- *cd4cd8 <- ggplot(data = testes, aes(x = contagem_cd4, y = contagem_cd8))*

- x = contagem_cd4
- *cd4cd8 <- ggplot(data = testes, aes(x = contagem_cd4))*
- y = contagem_cd8
- *cd4cd8 <- ggplot(data = testes, aes(x = contagem_cd4, y = contagem_cd8))*
- geom_point()

- x = contagem_cd4
- *cd4cd8 <- ggplot(data = testes, aes(x = contagem_cd4))*
- y = contagem_cd8
- *cd4cd8 <- ggplot(data = testes, aes(x = contagem_cd4, y = contagem_cd8))*
- geom_point()
- *cd4cd8 <- ggplot(data = testes, aes(x = contagem_cd4, y = contagem_cd8)) + geom_point()*

Resultado

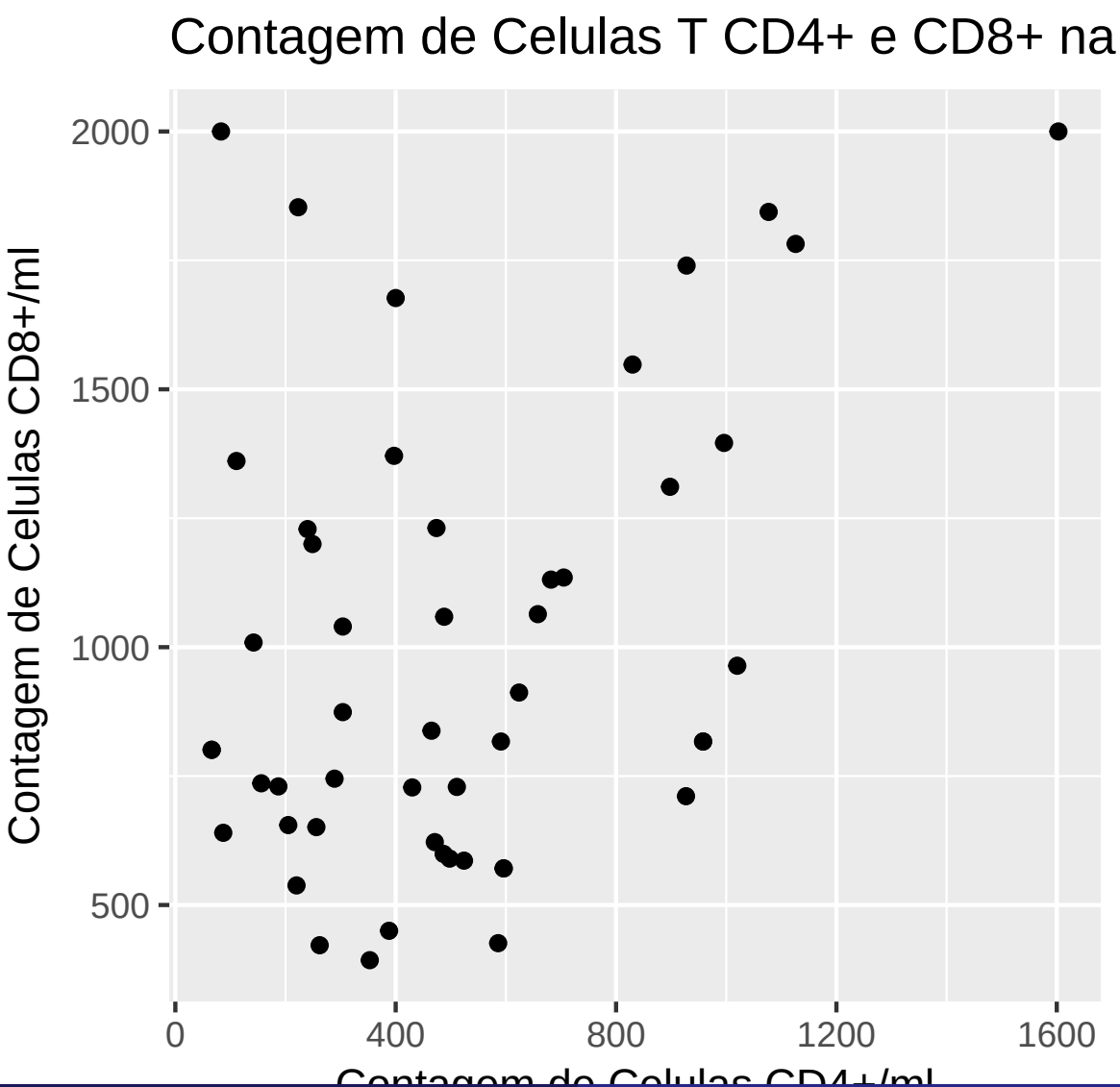
```
cd4cd8 <- ggplot(data = testes, aes(x = contagem_cd4, y = contagem_cd8)) + geom_poi
cd4cd8
```



Podemos Fazer Este Gráfico Mais Bonito

- Pôr títulos
  - No gráfico como todo
  - Nos eixos
- Usar o argumento `labs()`
  - Mais os subargumentos: x, y, title, subtitle
- `+ labs(x = "Contagem de Celulas CD4+/ml", y = "Contagem de Celulas CD8+/ml", title = "Contagem de Celulas T CD4+ e CD8+ na Amostra")`

```r
cd4cd8 <- ggplot(data = testes, aes(x = contagem_cd4, y = contagem_cd8)) + geom_poi
cd4cd8 <- cd4cd8 + labs(x = "Contagem de Celulas CD4+/ml",
                        y = "Contagem de Celulas CD8+/ml",
                        title = "Contagem de Celulas T CD4+ e CD8+ na Amostra")
cd4cd8
```

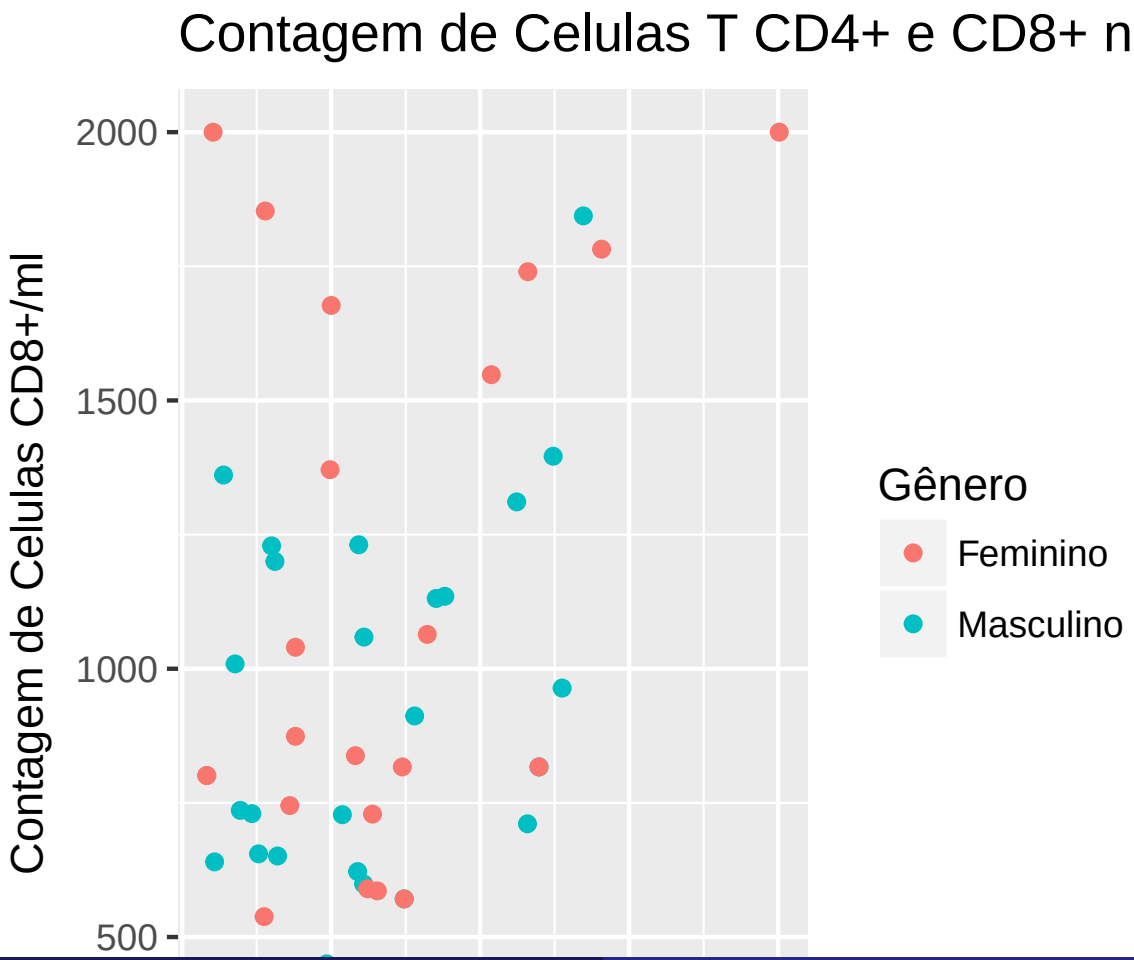


“+” Indica que Está Colocando Nova Camada no Gráfico

Qual é a Relação entre Gênero e os Celulas T?

- Podemos usar cores para indicar uma outra variável
- Use o argumento de `colour`
  - Coloque uma variável no argumento
  - `ggplot` selecionará as cores
- Nova versão do tibble com sexo (`testessexo`)

```r
grcdsex <- ggplot(data = testessexo, aes(x = contagem_cd4,
                                         y = contagem_cd8,
                                         colour = sexo)) + geom_point()
grcdsex <- grcdsex + labs(x = "Contagem de Celulas CD4+/ml",
                          y = "Contagem de Celulas CD8+/ml",
                          title = "Contagem de Celulas T CD4+ e CD8+ na Amostra",
                          colour = "Gênero")
grcdsex
```

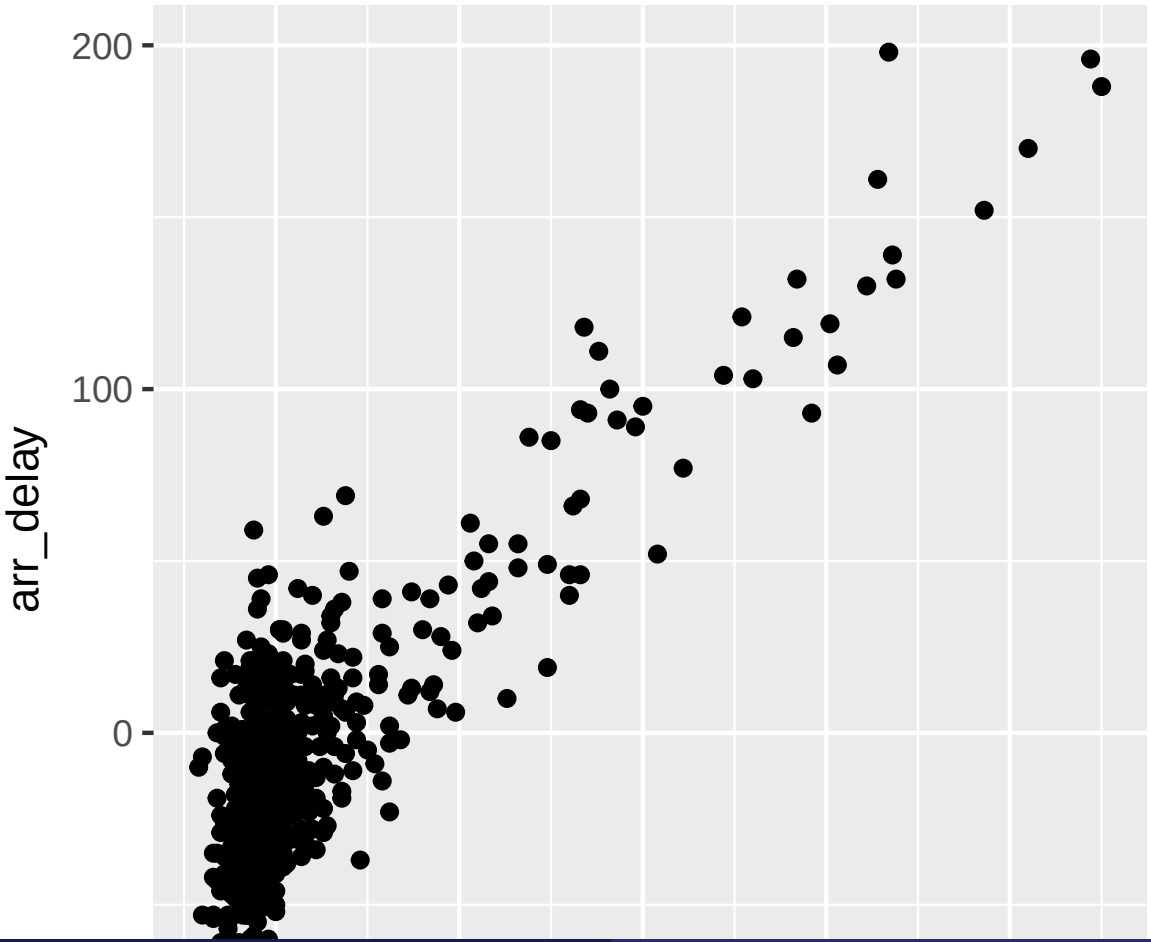


Overplotting

- Um exemplo não do mundo de biociências (sorry)
- Atrasos nos vôos de Alaska Airlines saindo de NYC em 2013
- Criar um tibble `AKvoos` com estes vôos
- Mostrar os atrasos na decolagem contra os de chegada
- Verá que muitos pontos são agrupados perto de 0,0
- Pode criar confusão em interpretação

Overplot Exemplo

```
data(flights)
AKvoos <- flights %>% filter(carrier == "AS" & !is.na(dep_delay) &
                                !is.na(arr_delay))
grAK <- ggplot(data = AKvoos, aes(x = dep_delay, y = arr_delay)) + geom_point()
grAK
```

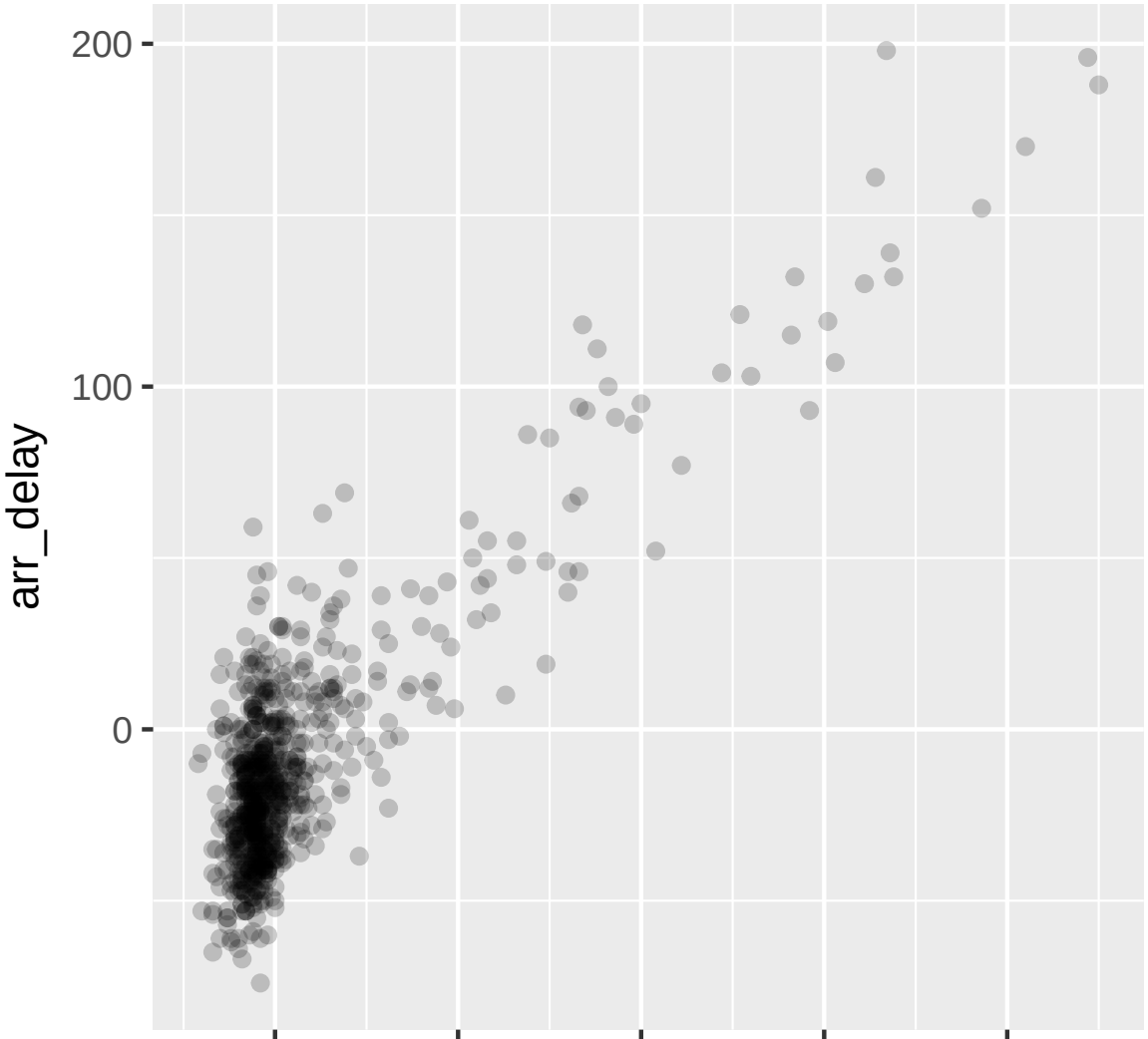


2 Maneiras de Tirar Excesso de Pontos perto de 0,0

- Ajustar a transparência dos pontos com o argumento `alpha`
  - Pode aceitar valores entre 0 e 1
- Criar espaço entre os pontos com o `geom_jitter()`

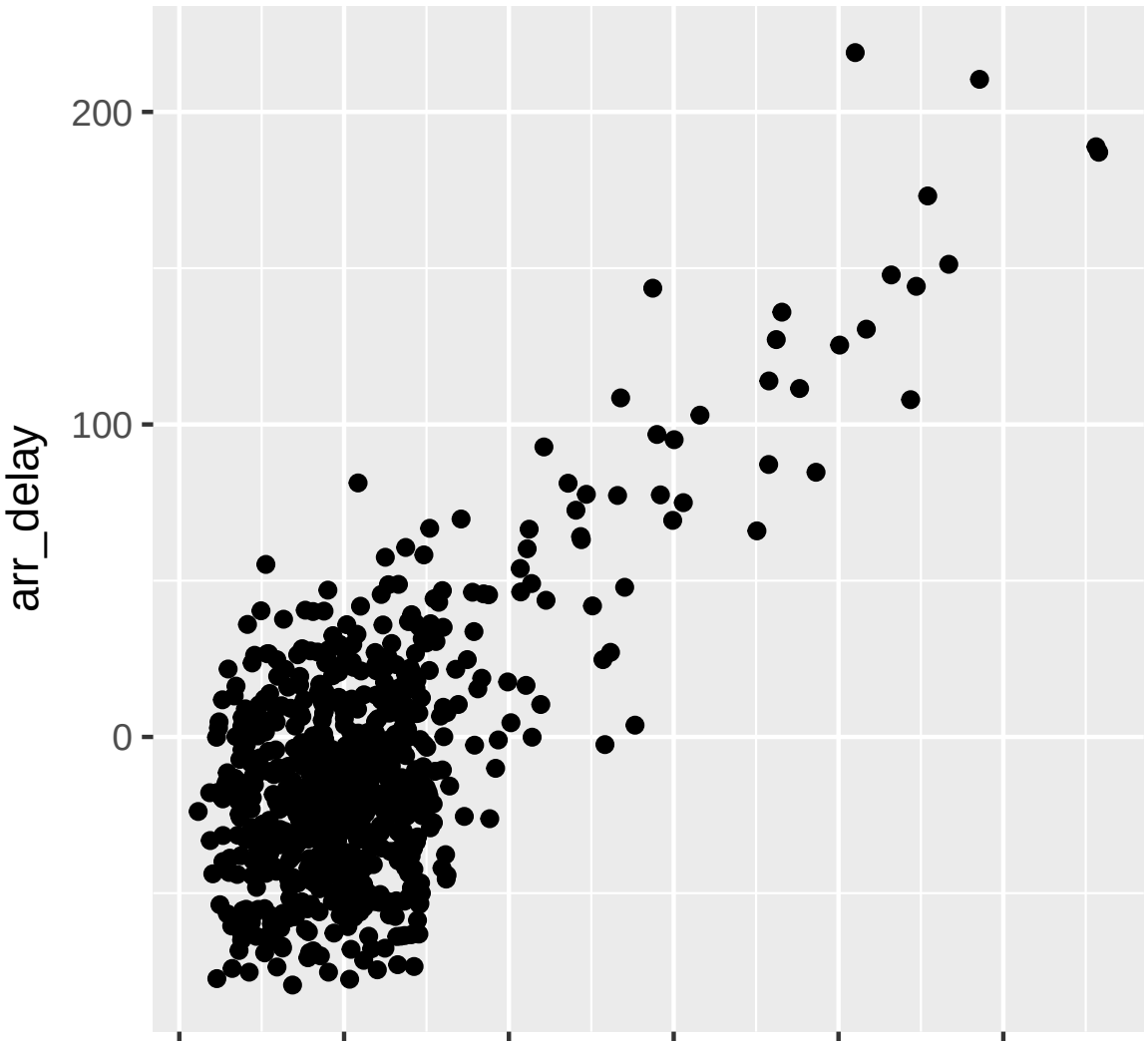
Ajustar Alphas para 0.2

```
grAK2 <- ggplot(data = AKvoos, aes(x = dep_delay, y = arr_delay)) +
  geom_point(alpha = 0.2)
grAK2
```



Ajustar com Jitter; Especificação de height and width

```
grAK3<- ggplot(data = AKvoos, aes(x = dep_delay, y = arr_delay)) +
  geom_jitter(width = 30, height = 30)
grAK3
```

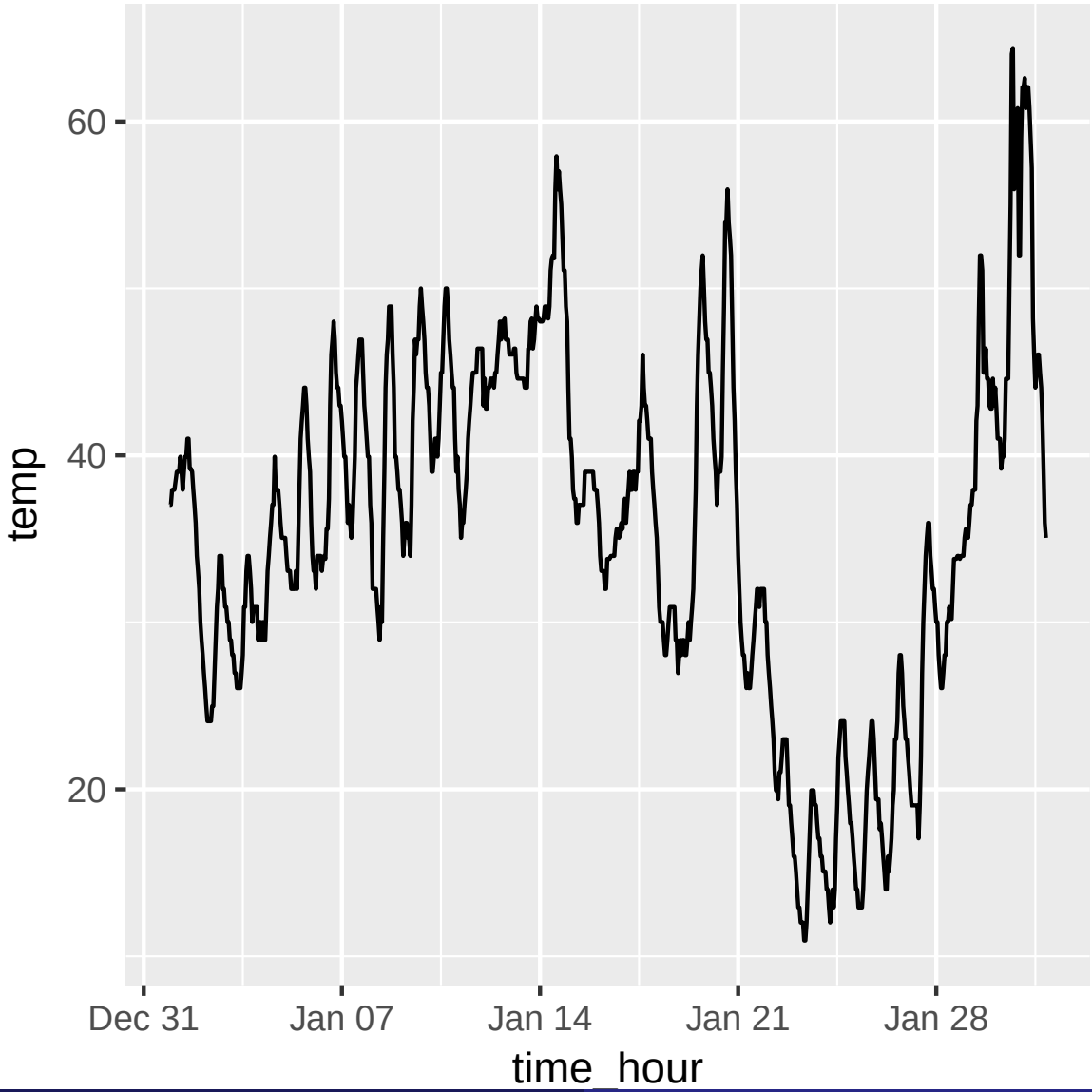


Gráficos de Linha

- Como dispersão, linhas mostra relações entre 2 variáveis continuas
- Linhas sugerem que existe realmente uma conexão entre os pontos
- Se este não existe, melhor usar gráfico de barra
- Tempo é um eixo x típico
- Janeiro é a temporada de influenza na região de New York
- Vamos olhar nas temperaturas no mês de Janeiro em um ponto: EWR (Aeroporto de Newark)
- Usamos `geom_line()`

Temperatura em Janeiro EWR

```
ggplot(data = jantemp, aes(x = time_hour, y = temp)) + geom_line()
```

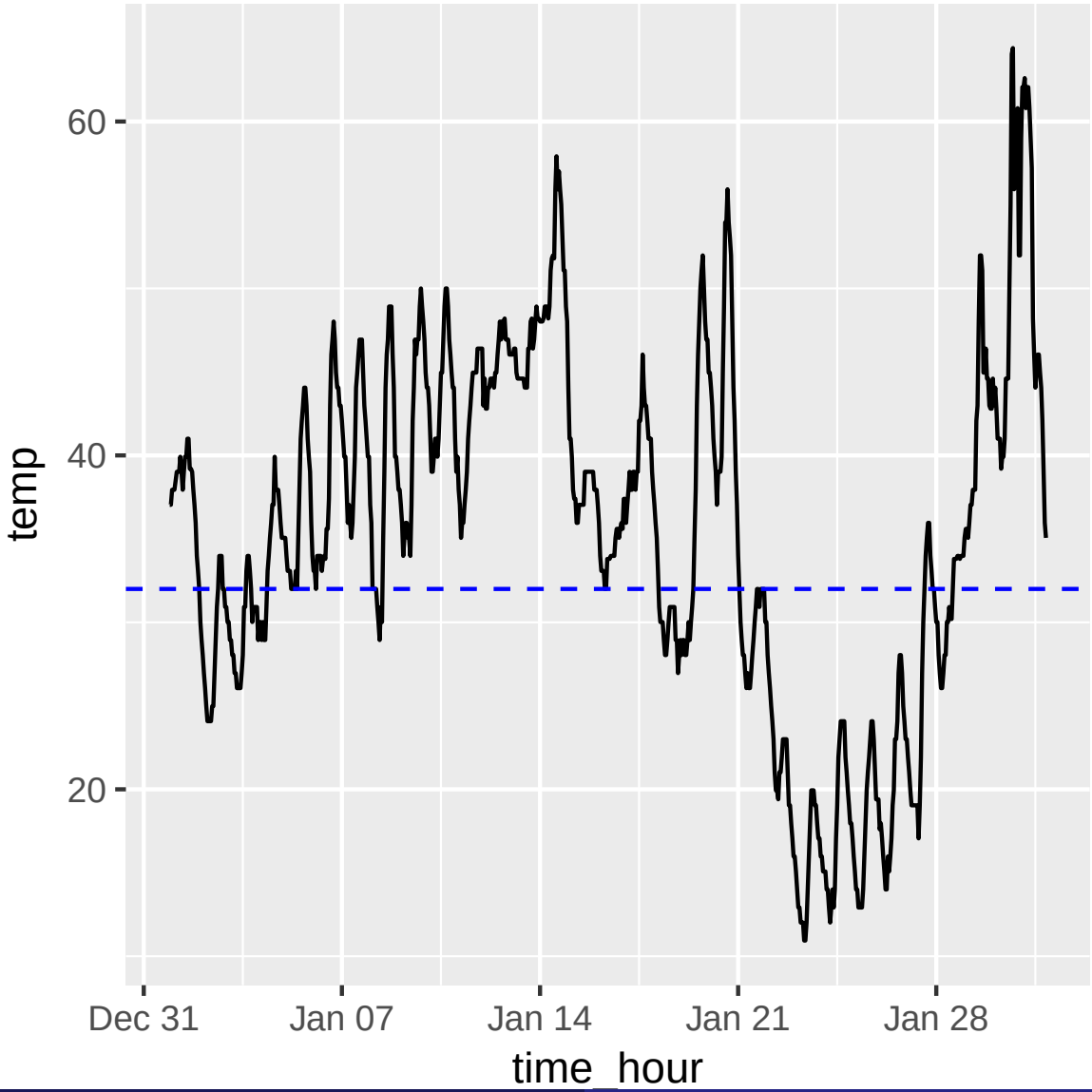


Melhorar Este Gráfico

- Pode indicar o ponto de congelamento (0ºC = 32ºF)
- Criar uma nova camada com `hline()`
  - Pode colocar uma linha horizontal em qualquer ponto do eixo y
  - Existe também `vline()` (vertical) e `abline()` (linha calculada)
  - Todos podem ter cor e format diferente da linha básica do gráfico

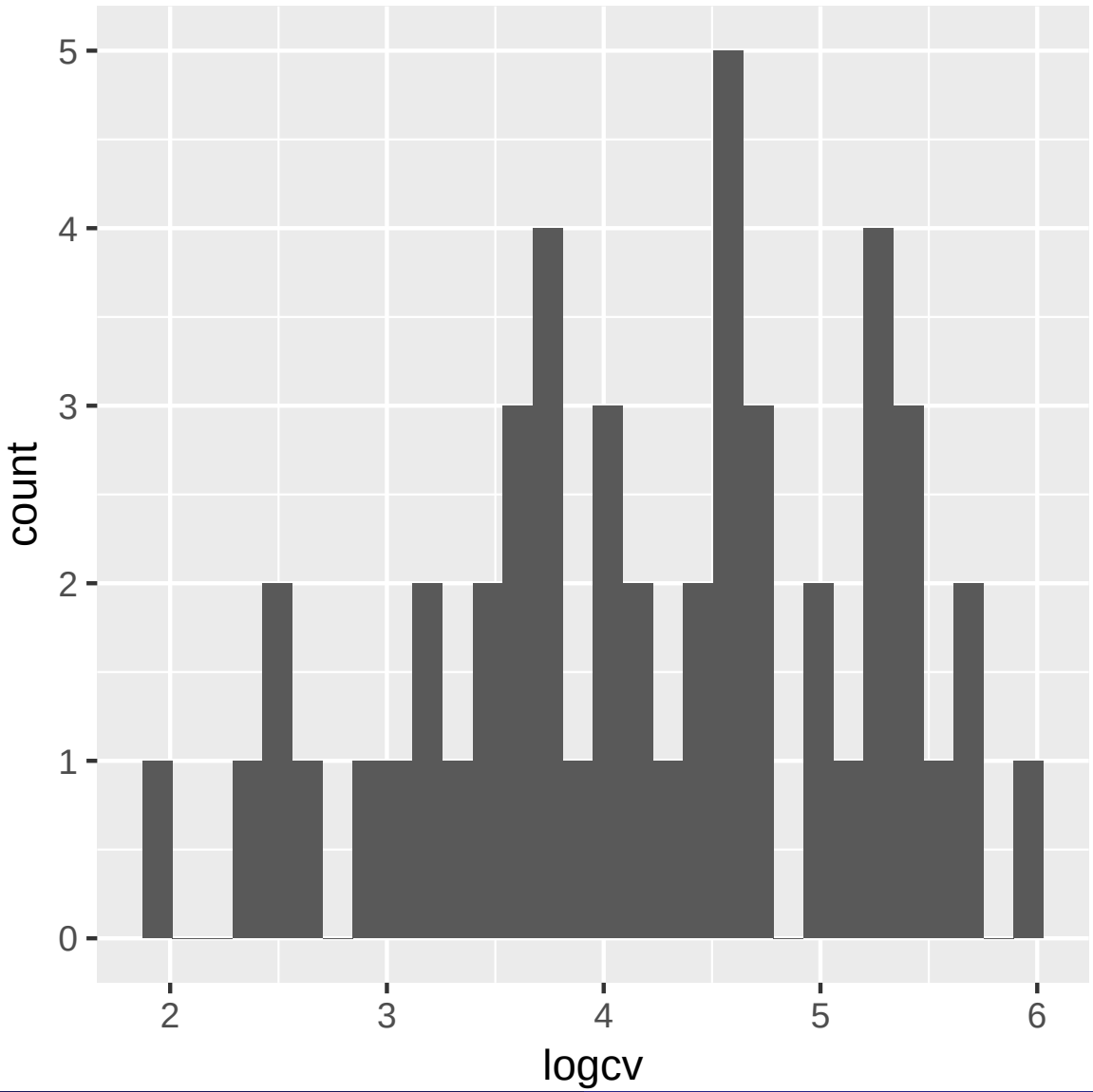
Demonstração de `hline`

```
ggplot(data = jantemp, aes(x = time_hour, y = temp)) + geom_line() + geom_hline(aes(
```



Histograma de `logcv`

```
ggplot(data = testes, mapping = aes(x = logcv)) + geom_histogram()
```

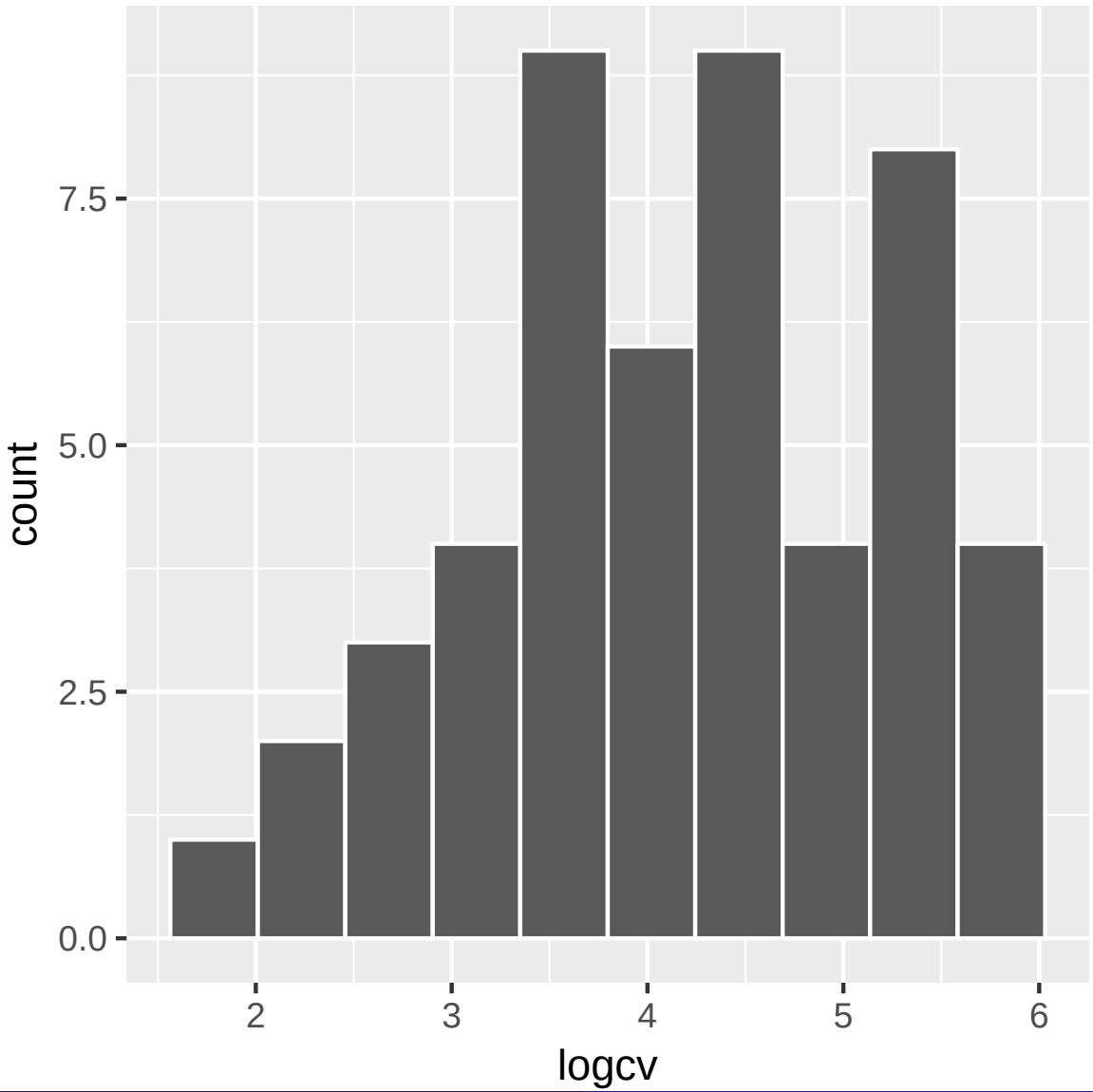


Ajustando o Número de Bins

- Pode aumentar ou reduzir o número de bins utilizando 1 de 2 argumentos
- "`binwidth =`" ajuste o intervalo dos bins
  - Se você quiser um intervalo especifico (ex: 10 - 19, 20 - 29, etc.)
- "`bins =`" ajuste o número de bins (padrão é 30 bins)

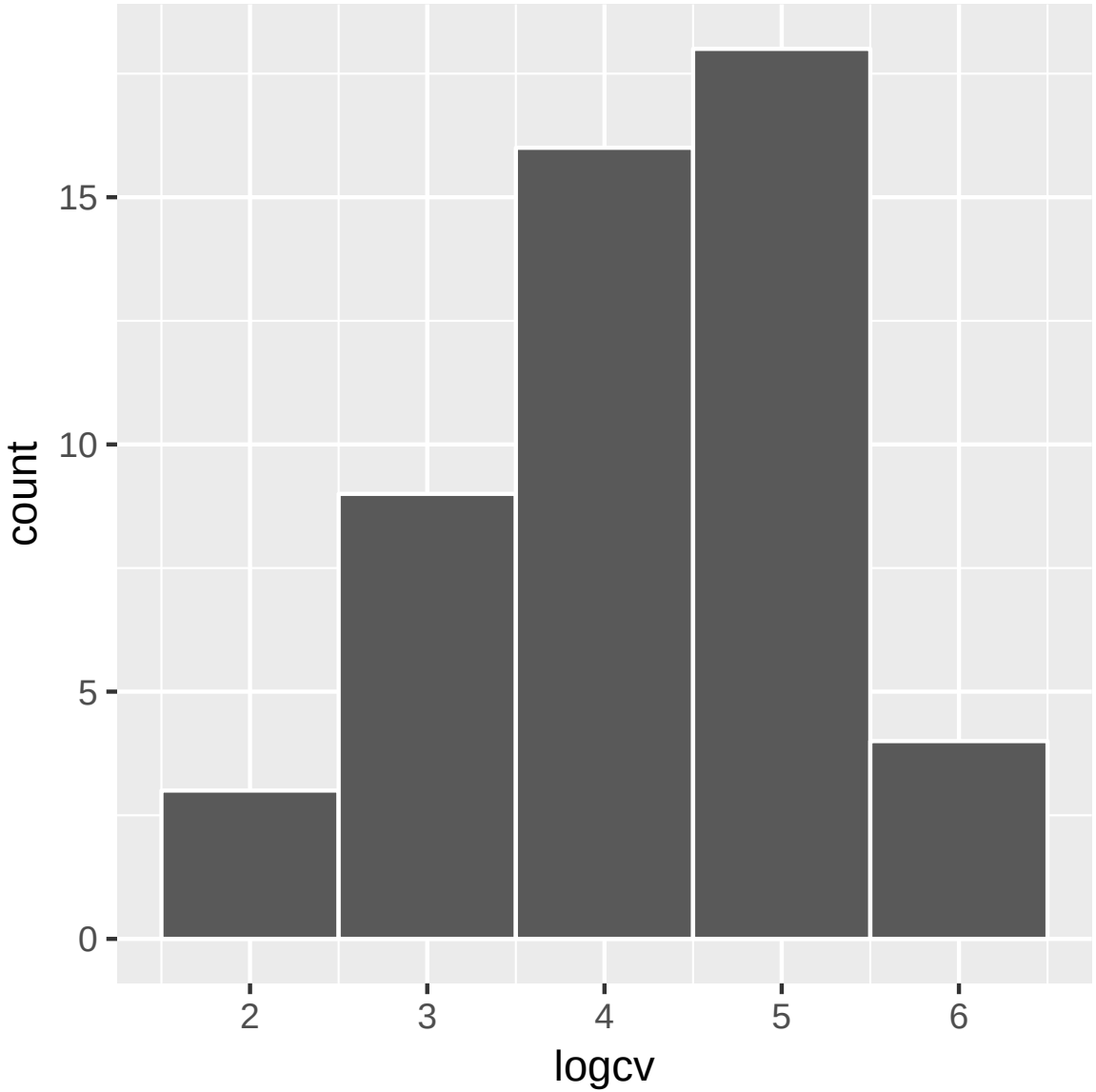
Reduzir o Número de Bins para 10

```
ggplot(data = testes, mapping = aes(x = logcv)) + geom_histogram(bins = 10, color =
```



Ajustar o Intervalo de binwidth para 1

```
ggplot(data = testes, mapping = aes(x = logcv)) + geom_histogram(binwidth = 1, color
```

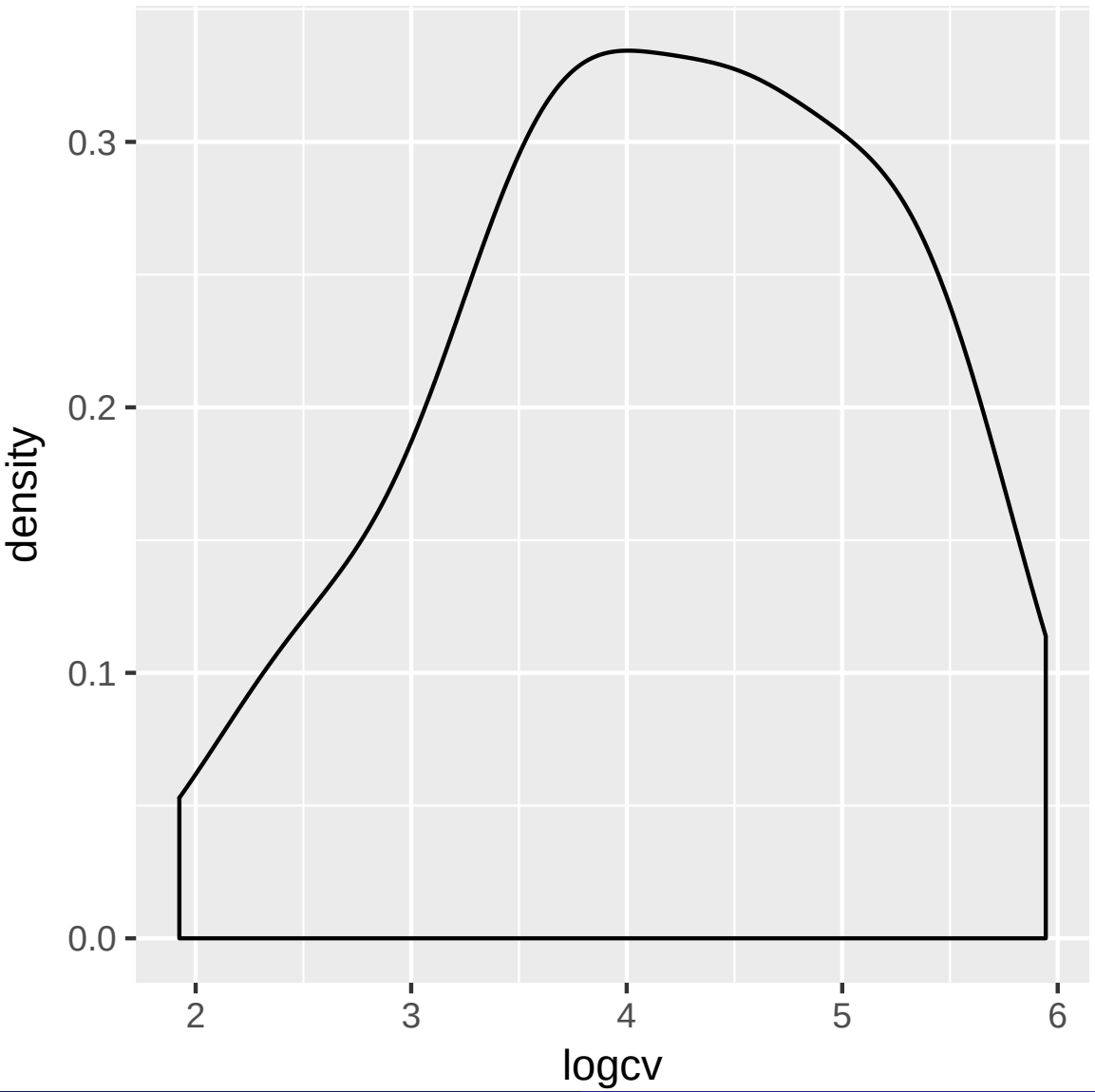


Aumentar o Histograma com uma Curva de Densidade

- Também pode só mostrar a curve sem o histograma
  - É um `geom` separada
- Precisa o argumento `stat = "density"` (padrão para `geom_density`)
- Quanto suave você quer a curva, precisa ajustar o "*kernel bandwidth*"
- Maior o *kernel bandwidth*, mais suave a curva
- Argumento para ajustar: `adjust =`; padrão é `adjust = 1`

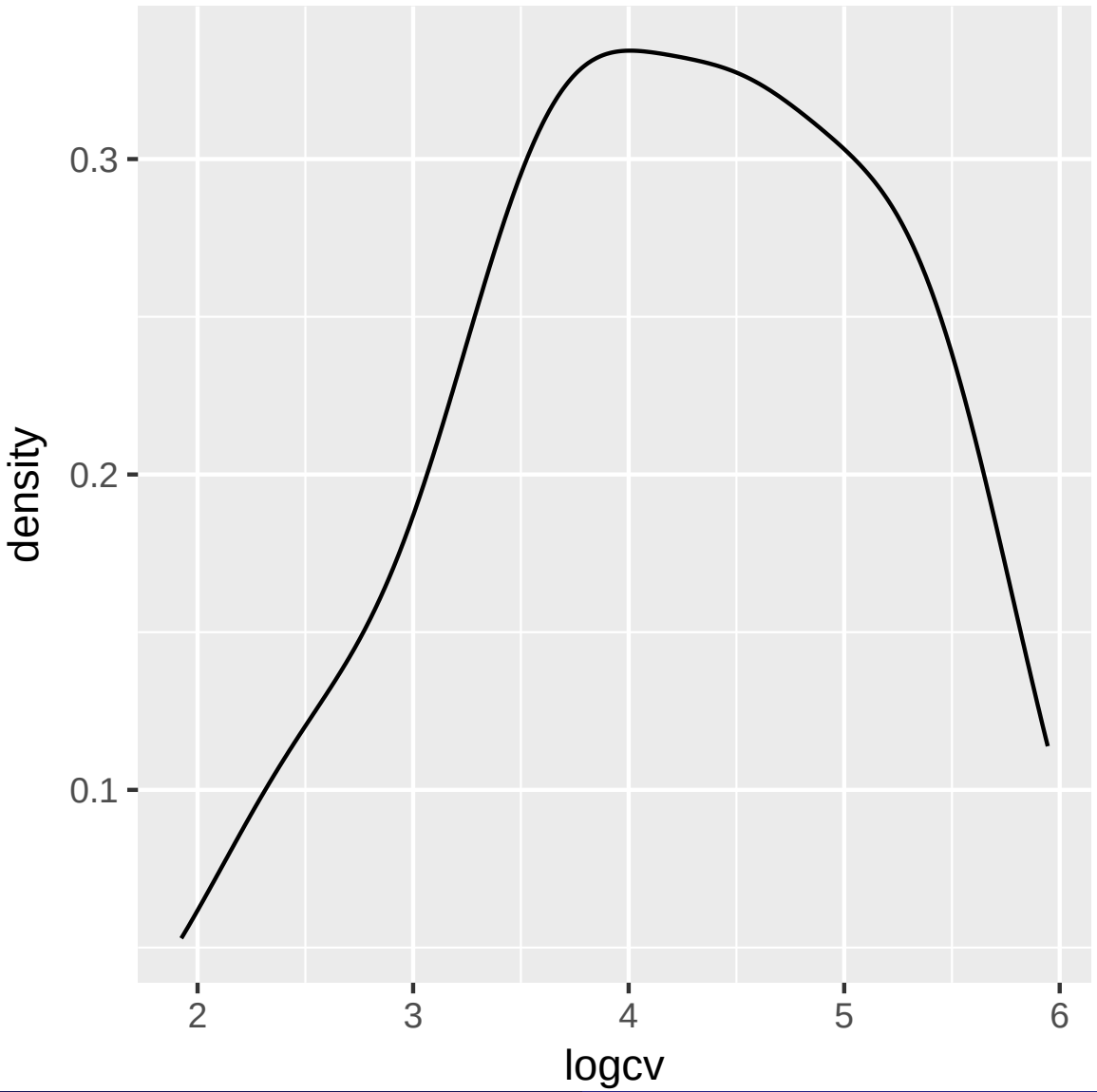
Curva de Densidade Sozinho

```
ggplot(data = testes, mapping = aes(x = logcv)) + geom_density()
```



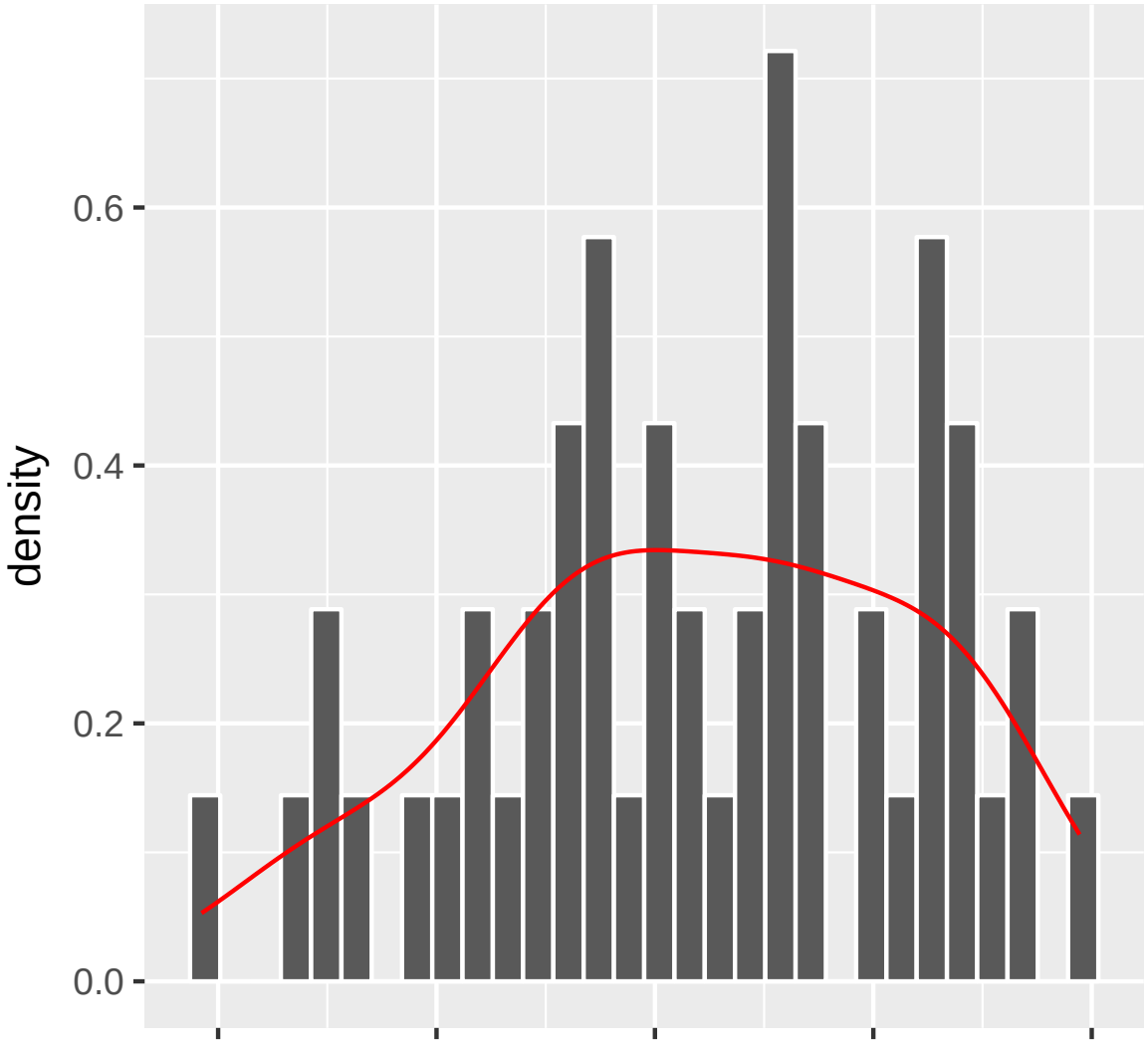
Para Tirar a Linha ao Fundo, Use geom_linha

```
ggplot(data = testes, mapping = aes(x = logcv)) + geom_line(stat = "density")
```



Para Combinar com Histograma, Use Versão `geom_line(stat = "density")`

```
ggplot(data = testes, mapping = aes(x = logcv, y = ..density..)) + geom_histogram(co
```



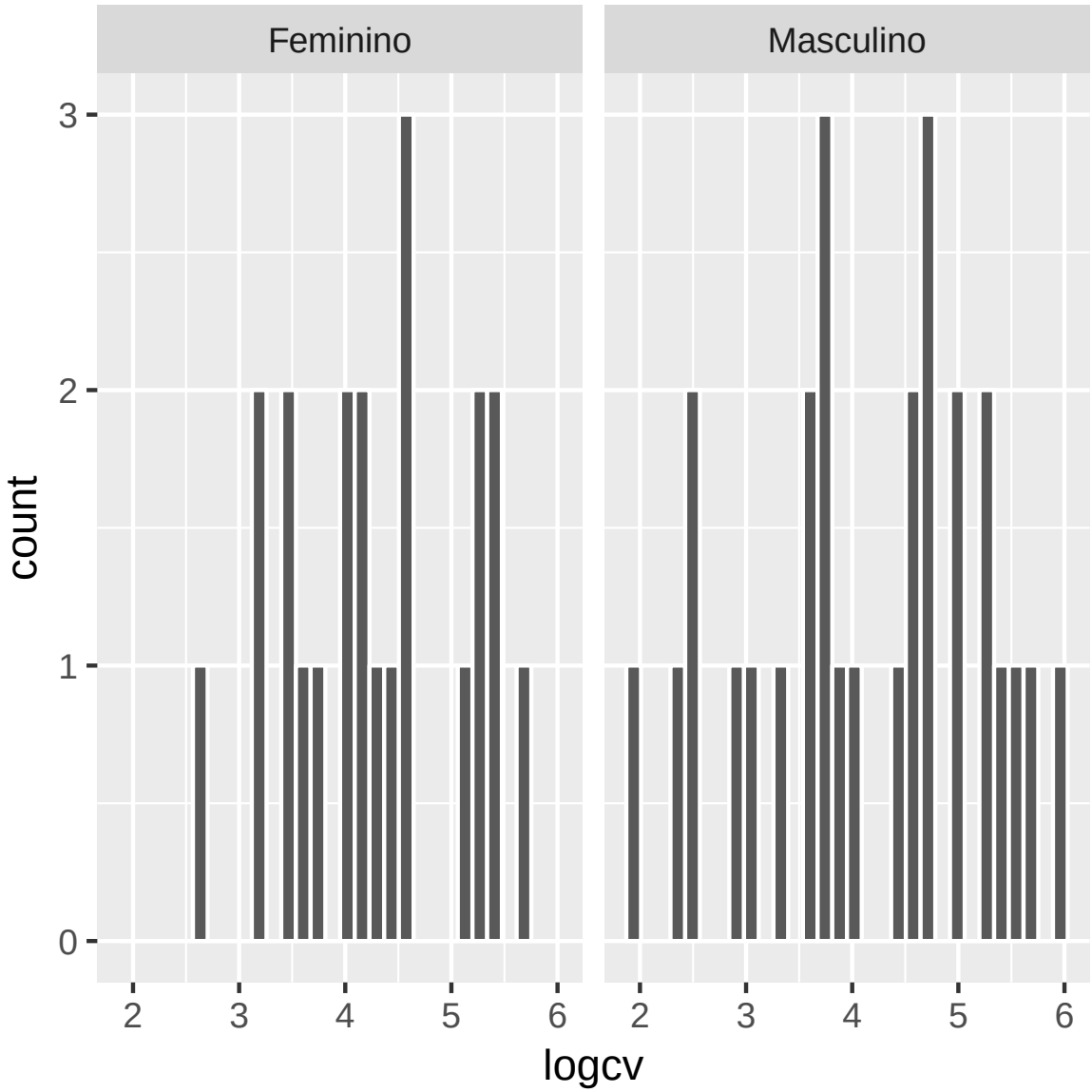
Kernel Bandwidth

```
ggplot(data = testes, mapping = aes(x = logcv)) +
  geom_line(stat = "density", adjust = 0.5, color = "red") +
  geom_line(stat = "density", adjust = 1.0) +
  geom_line(stat = "density", adjust = 2.0, color = "blue")
```

Olhar nos Histogramas dos Gêneros - Tem Diferença entre Eles?

- Podemos combinar histogramas dos níveis de uma variável categórica em 1 gráfico
- Camada: 'facet_wrap() (em geral, "*facets*")
- Facets precisam:
  - Uma formula usando til ("~")
  - Se for complicado, especificação de número de linhas (`nrow`) e colunas (`ncol`)

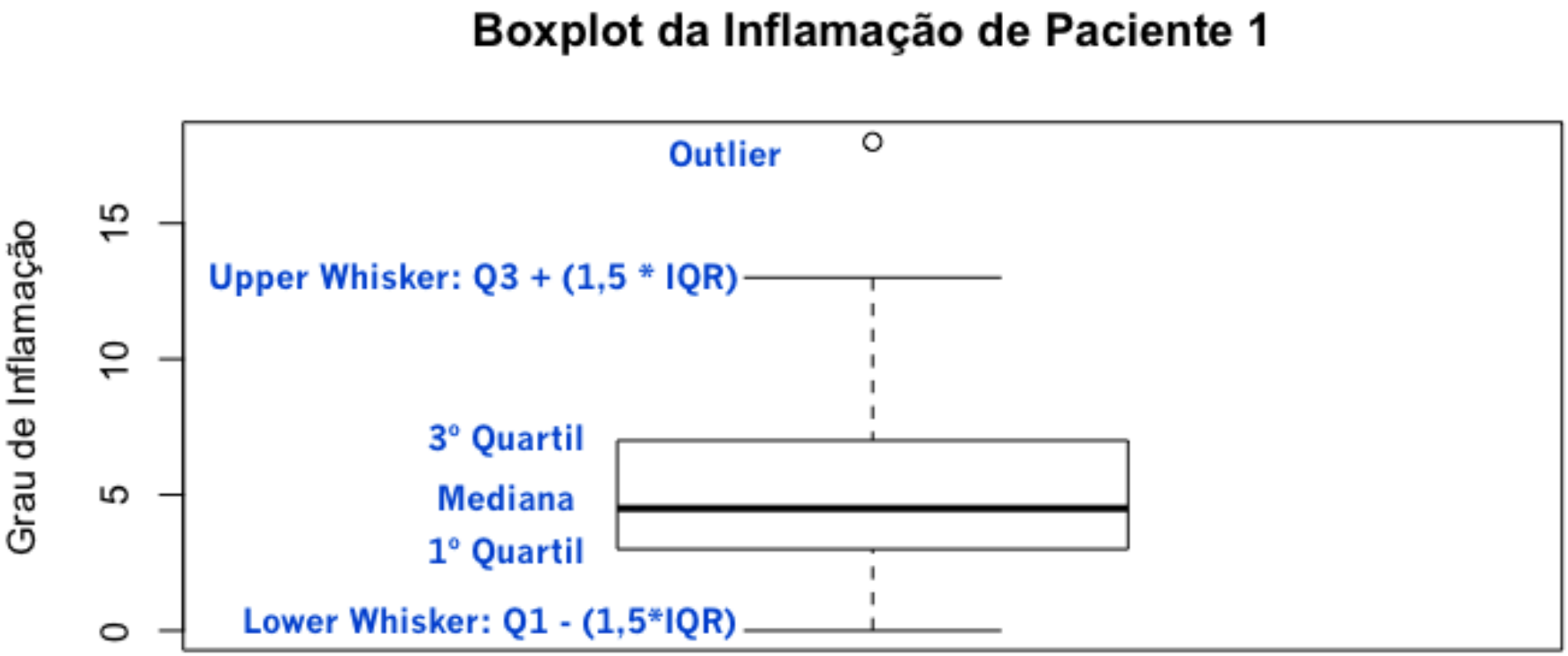
```
ggplot(data = testessexo, mapping = aes(x = logcv)) + geom_histogram(color = "white"
```



Boxplots

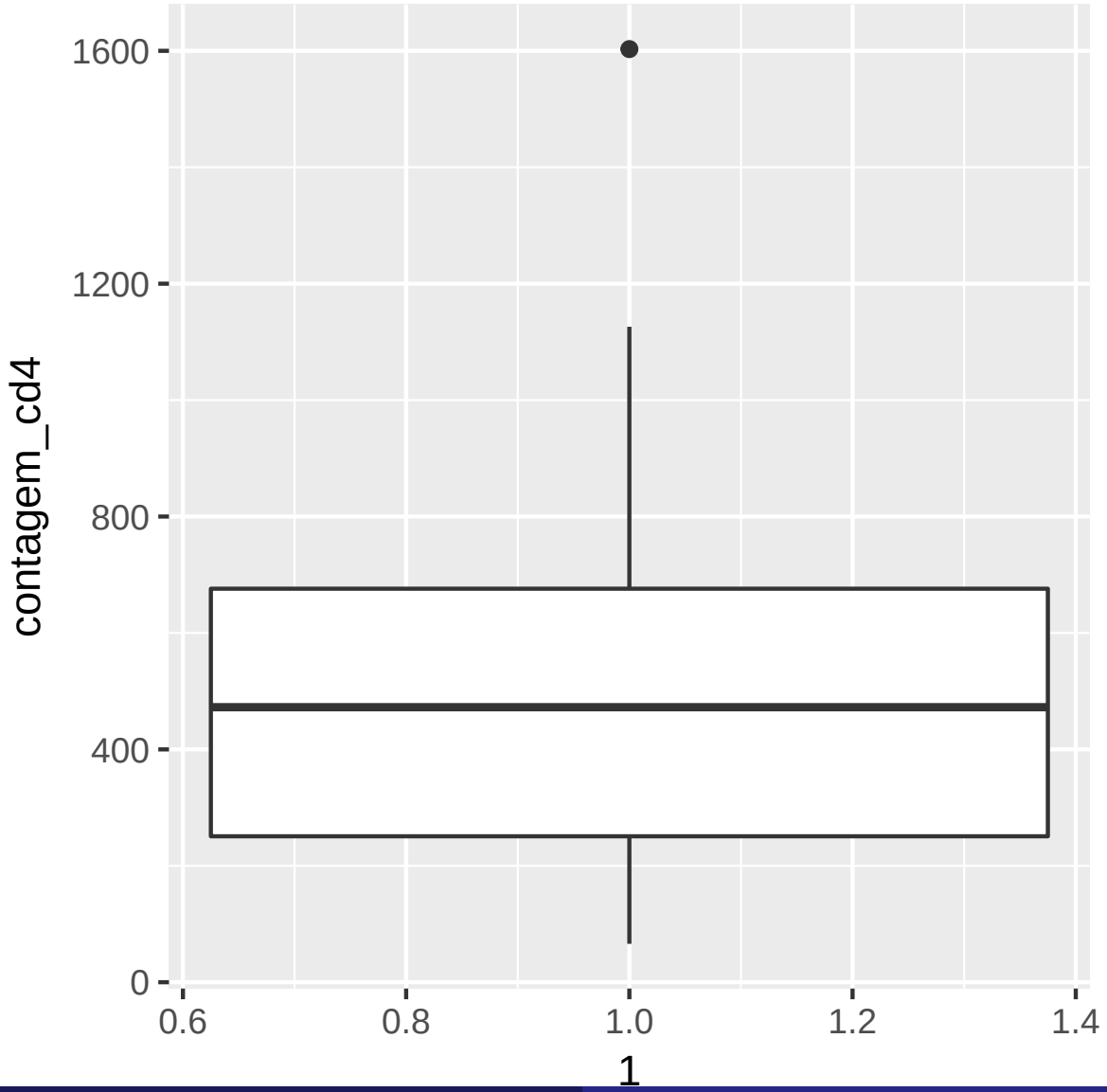
- Segunda maneira de mostrar uma distribuição
- Mais difícil para entender que histogramas
  - Só precisa prática
  - Mas, eu acho o mais útil para demonstrar qualidades da distribuição

Estrutura de um Boxplot



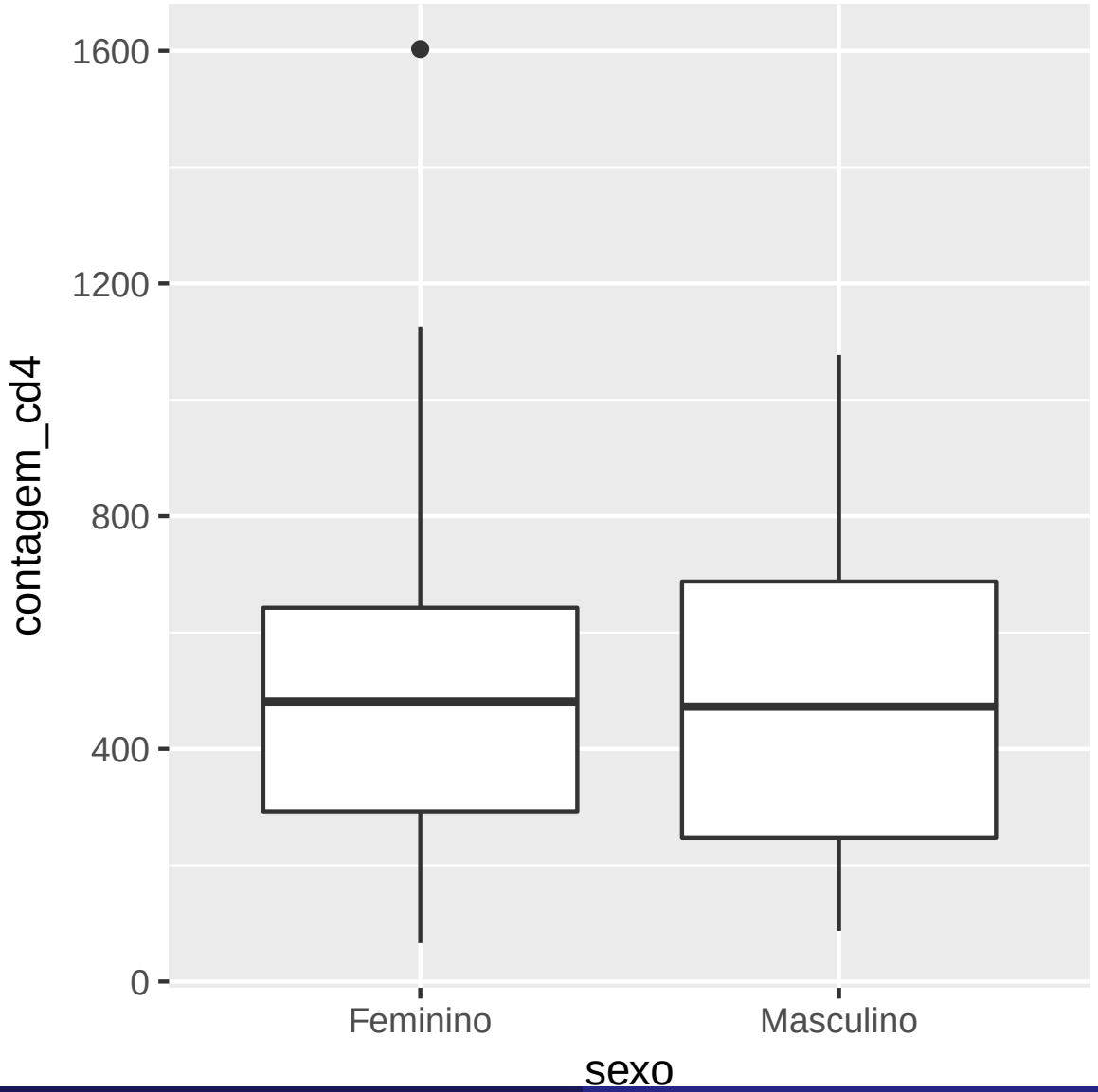
Boxplot de Nossos Dados de CD4 – Versão Mais Simples

```
ggplot(data = testes, mapping = aes(x = 1, y = contagem_cd4)) + geom_boxplot()
```



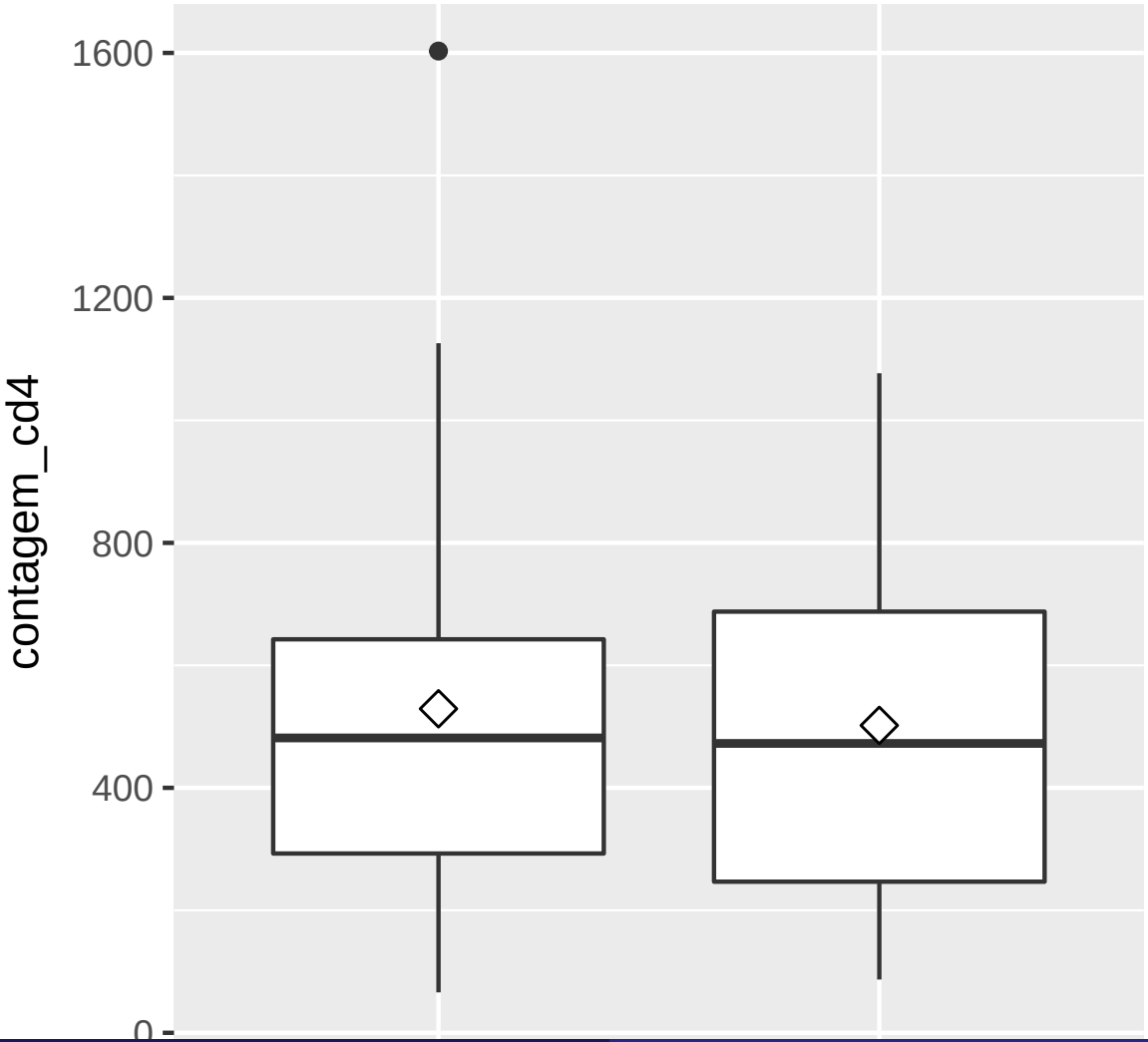
Pode Mostrar Diferenças entre 2 Nivéis de uma Variável

```
ggplot(data = testessexo, mapping = aes(x = sexo, y = contagem_cd4)) + geom_boxplot(
```



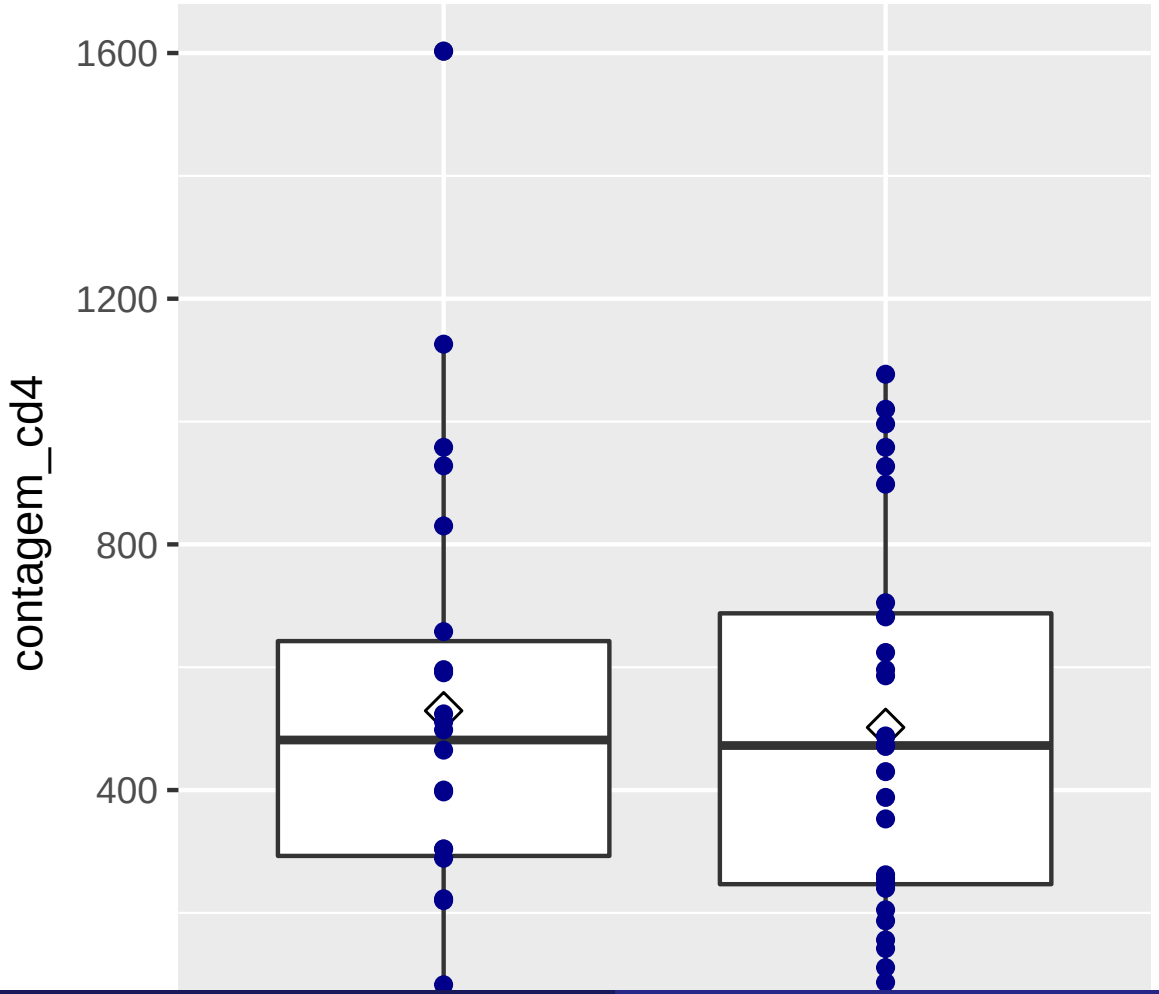
Pode Mostrar Também a Média no Boxplot

```
ggplot(data = testessexo, mapping = aes(x = sexo, y = contagem_cd4)) +
  geom_boxplot() + stat_summary(fun.y="mean",
  geom="point", shape=23, size=3, fill="white")
```



Incluir os Dados no Boxplot em Forma de Pontos

```
ggplot(data = testessexo, mapping = aes(x = sexo, y = contagem_cd4)) +
  geom_boxplot() + stat_summary(fun.y="mean",
  geom="point", shape=23, size=3, fill="white") +
  geom_point(color = "darkblue")
```



Exemplo de Um Boxplot mais Elegante

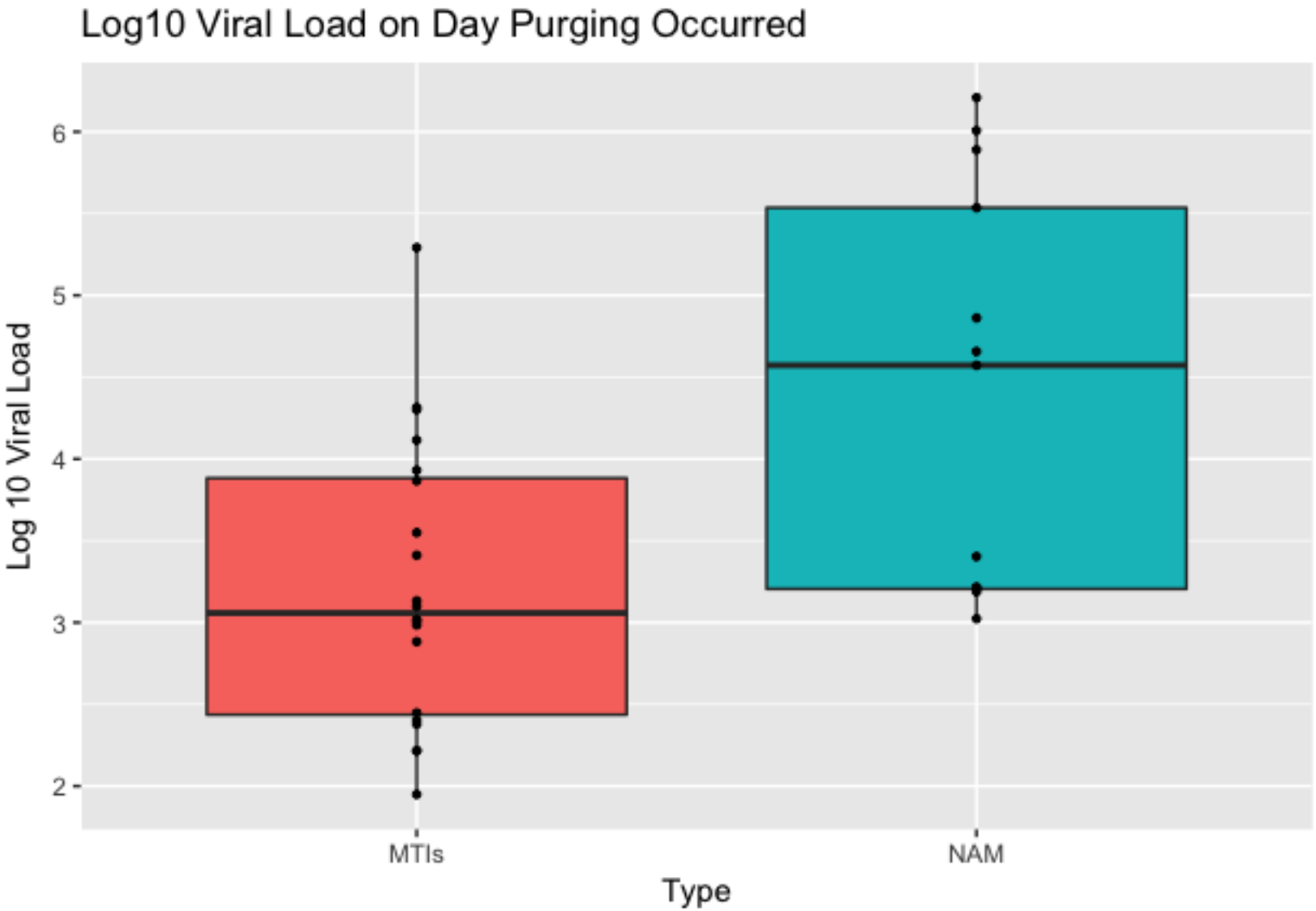
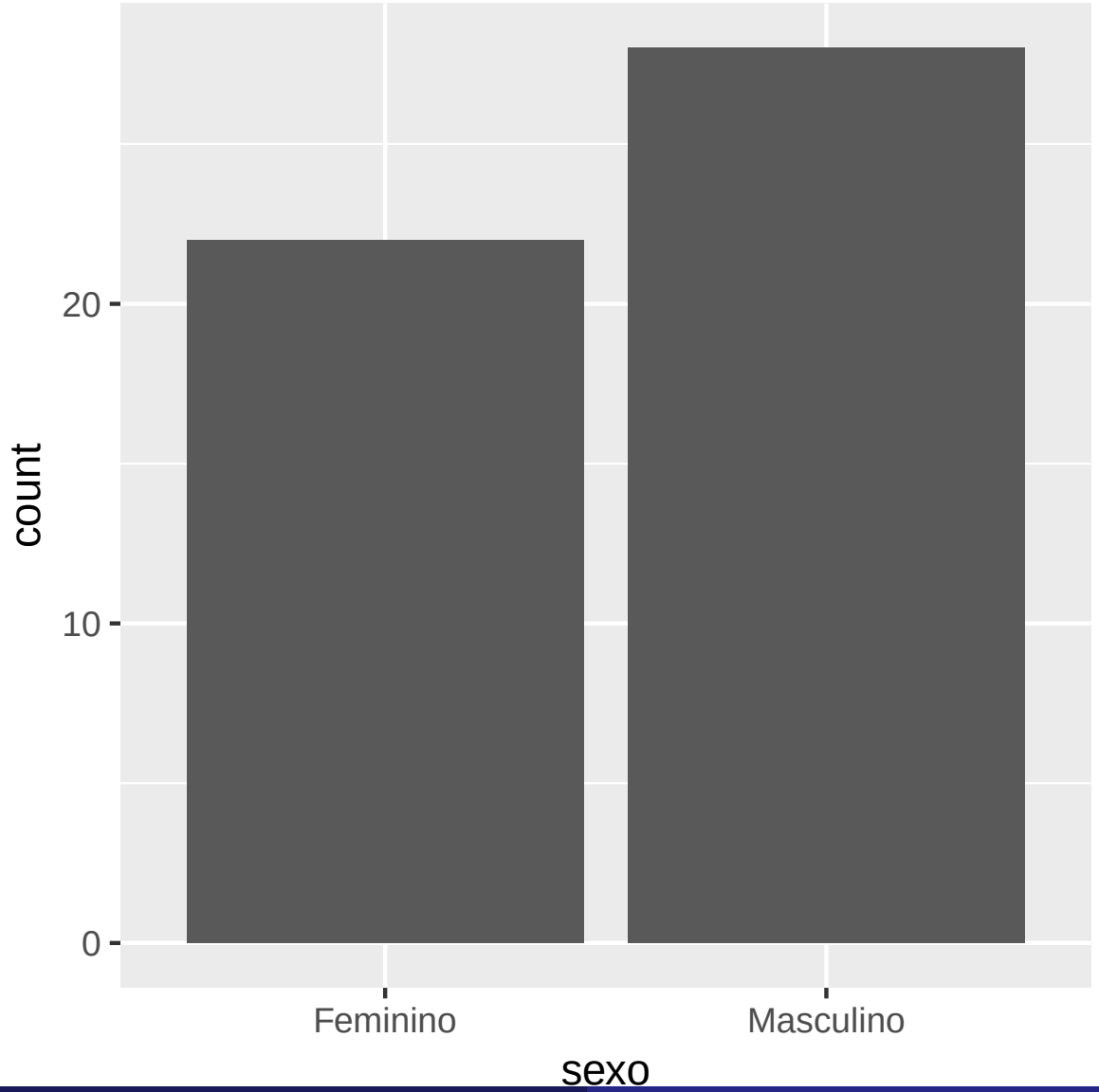


Gráfico de Barra

- Outra maneira de visualizar frequências das variáveis quantitativos
- Também funciona bem com variáveis categóricas
- `geom_bar()`

Applicação Simples – Gráfico de Distribuição dos Gêneros

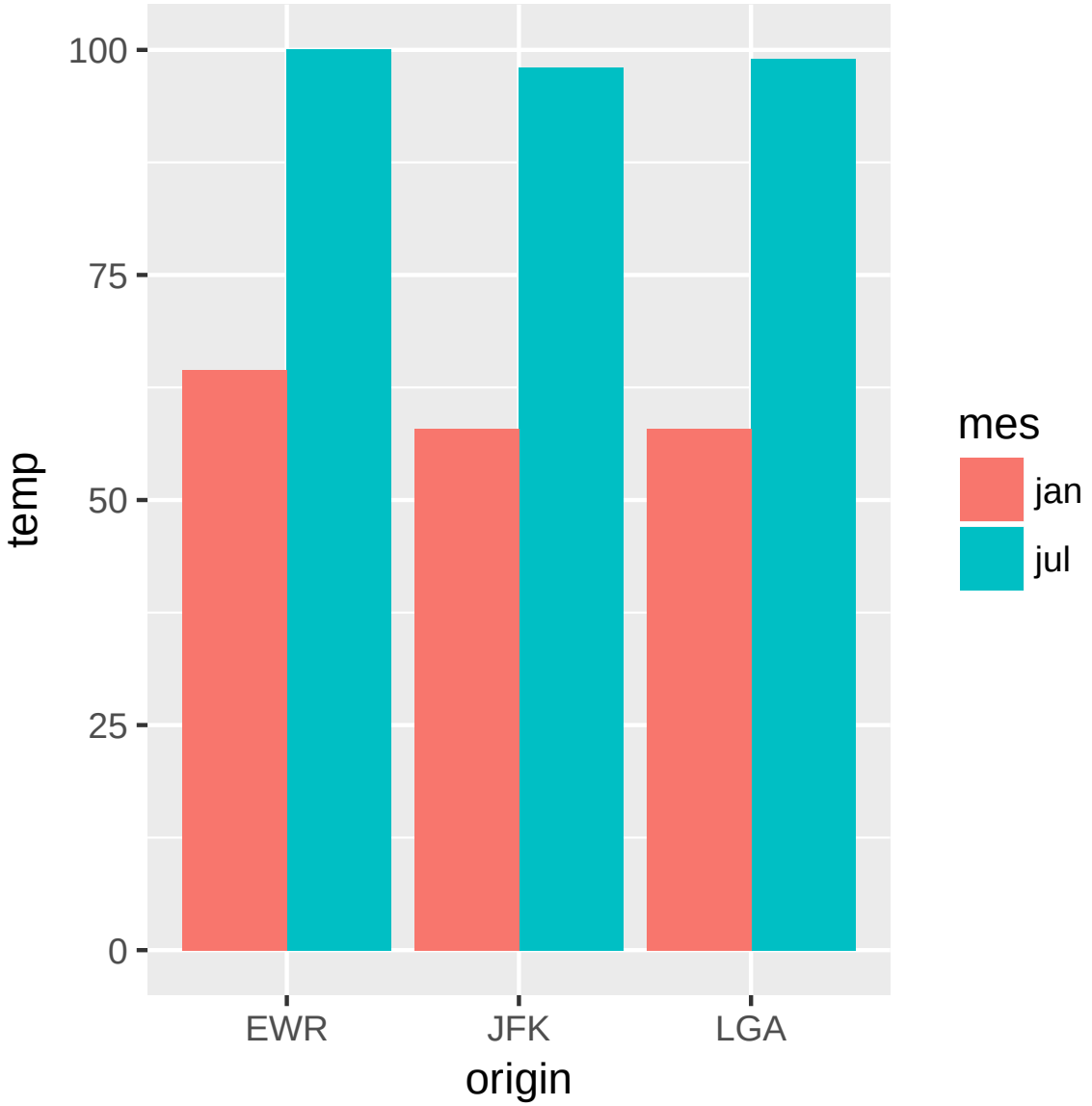
```
ggplot(data = testessexo, mapping = aes(x = sexo)) + geom_bar()
```



Pode Aumentar uma Variável e Grupar as Barras

- Voltamos às temperaturas na região de New York
- Queremos comparar temperatura em Janeiro e Julho aos 3 aeroportos
- Precisa especificar uma variável no `aes` `fill =`
  - `fill` = colocaria uma cor para cada nível da variável
- Precisa contar para `geom_bar()` que você quer as barras uma ao lado do outra
  - `position = "dodge"`

```
ggplot(data = janjul, mapping = aes(x = origin, y = temp, fill = mes)) +
  geom_bar(position = "dodge", stat = "identity")
```



Se Você Omitir `position = "dodge"` – Stacked Bar

```
ggplot(data = janjul, mapping = aes(x = origin, y = temp, fill = mes)) +
  geom_bar(stat = "identity")
```

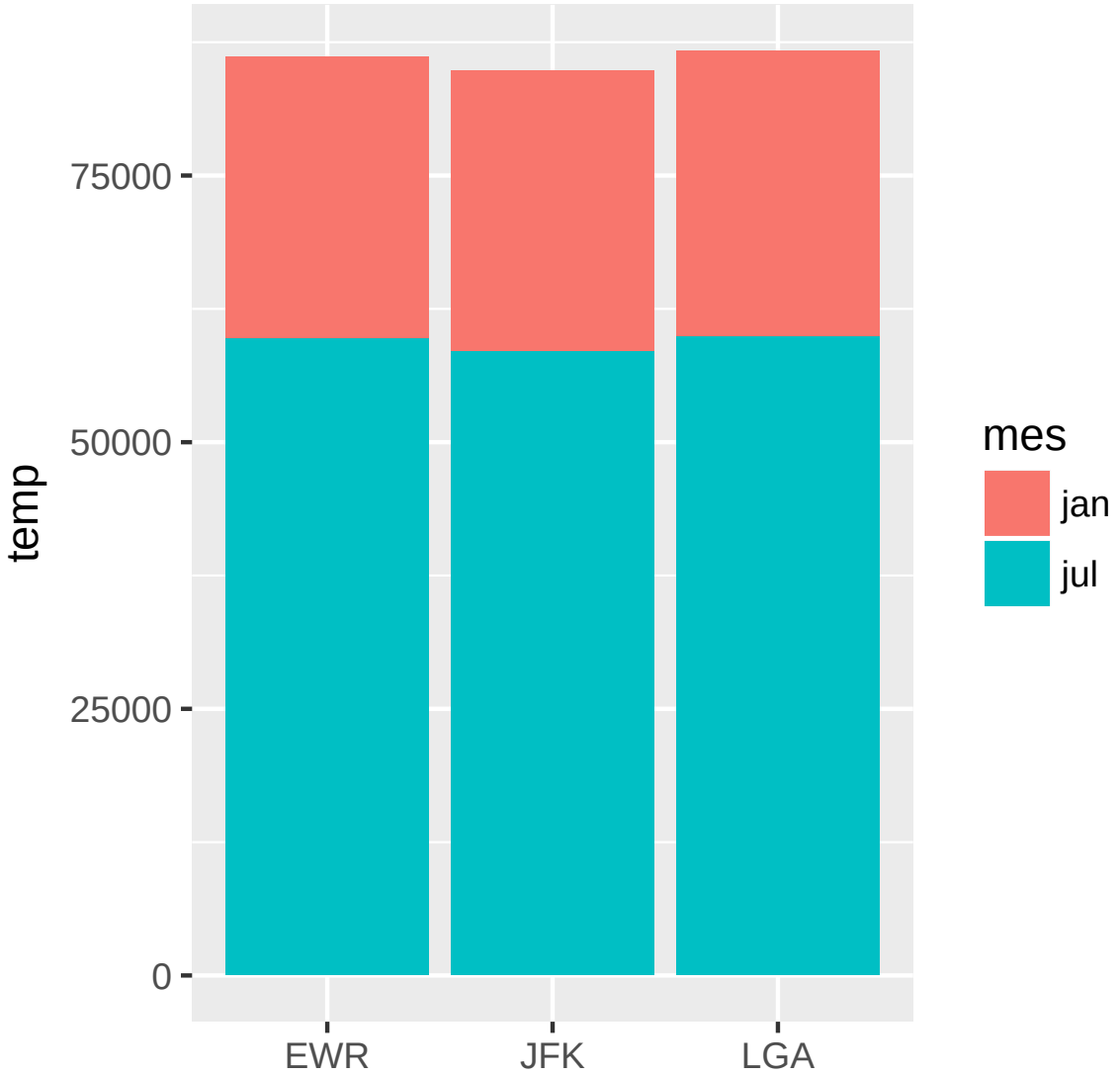
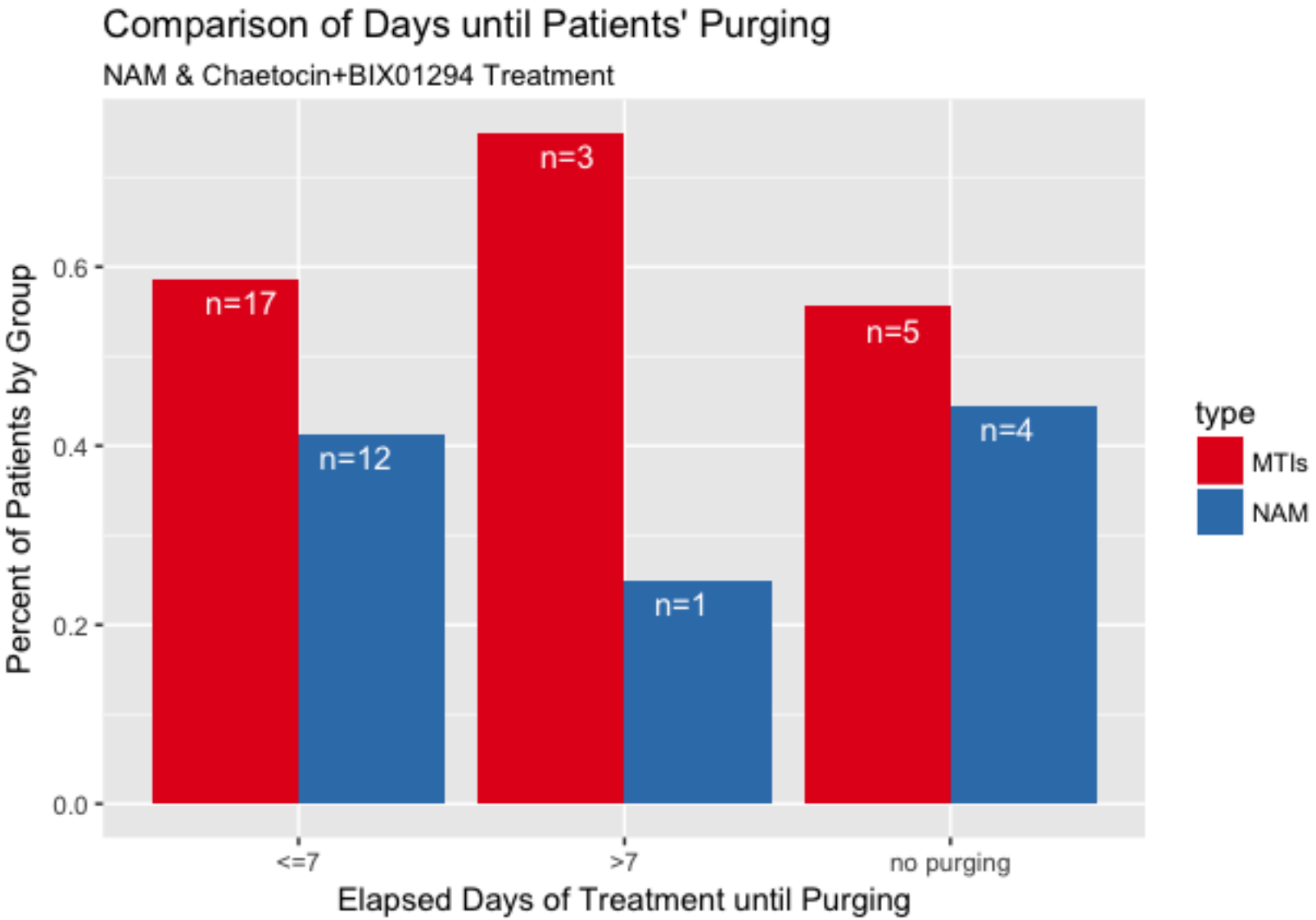


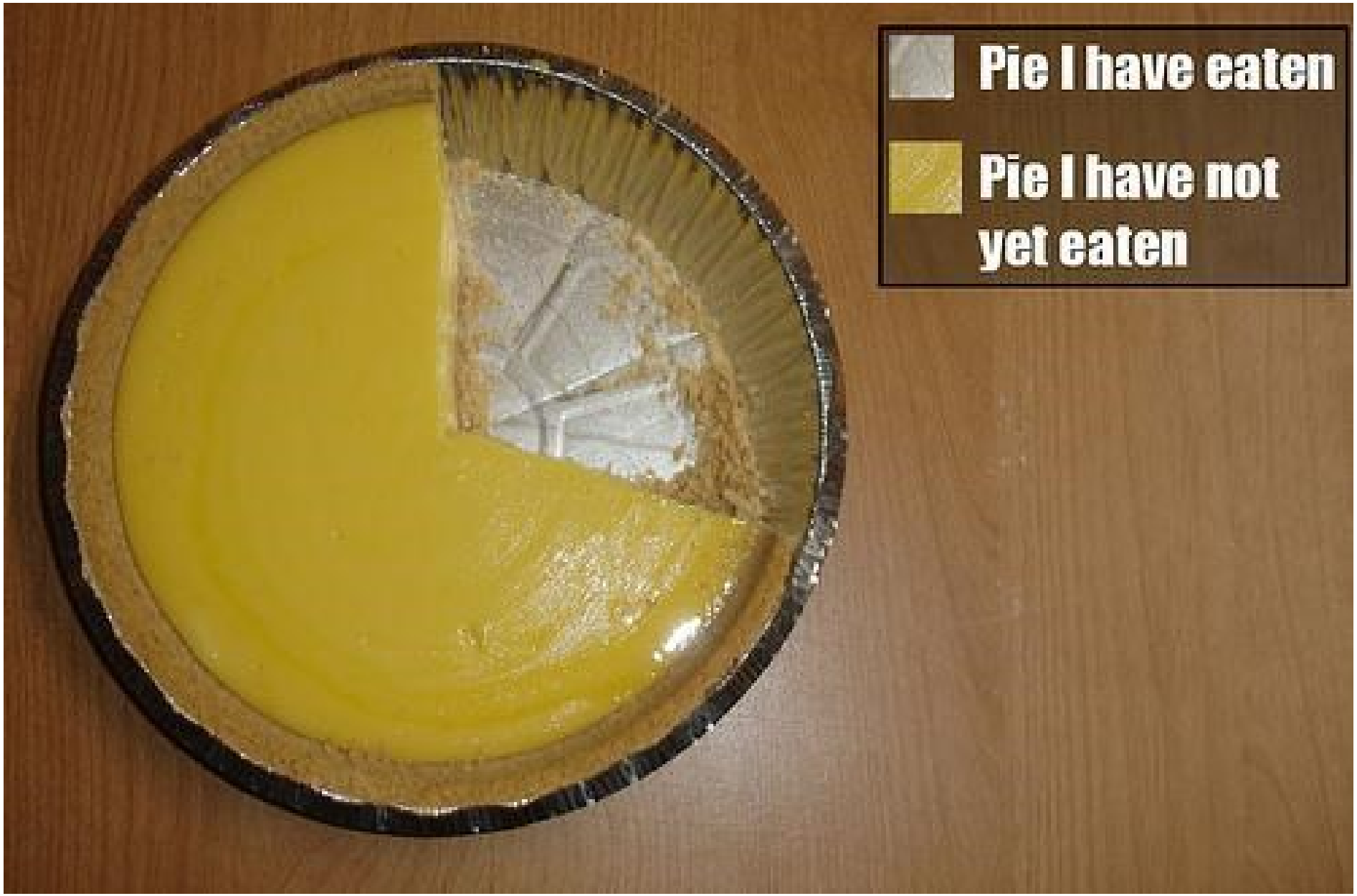
Gráfico de Barras Completo



Gráficos de Pizza (pie charts) – Porque Não Falei Deles??????

- Porque o olho de um ser humano não pode ver distinções finas entre as fatias
  - Especialmente se fosse muitas
- Use gráficos de barra
  - Olho interpreta altura ou largura melhor que a área
- Resumo de Gráficos de Pizza – **Nunca, jamais use eles**

Um tipo de Pie Chart Permitido – Graças a Nathan Yau



Fontes para Esta Aula

1. Chester Ismay and Albert Y. Kim, ModernDive, **An Introduction to Statistical and Data Sciences via R** (ismayc.github.io/moderndiver-book/index.html)
2. Wickham & Grolemund, **R for Data Science**, (`http://r4ds.had.co.nz` or O'Reilly)
3. Data Visualization with ggplot2: Cheat Sheet (RStudio)
4. Winston Chang, **R Graphics Cookbook** (O'Reilly)