Tutorial - Como fazer um Histograma no R usando ggplot2

Prof.DaviRocha

1 de novembro de 2018

Espero que ao fim desse artigo você seja capaz de fazer histogramas no R como o da figura abaixo.

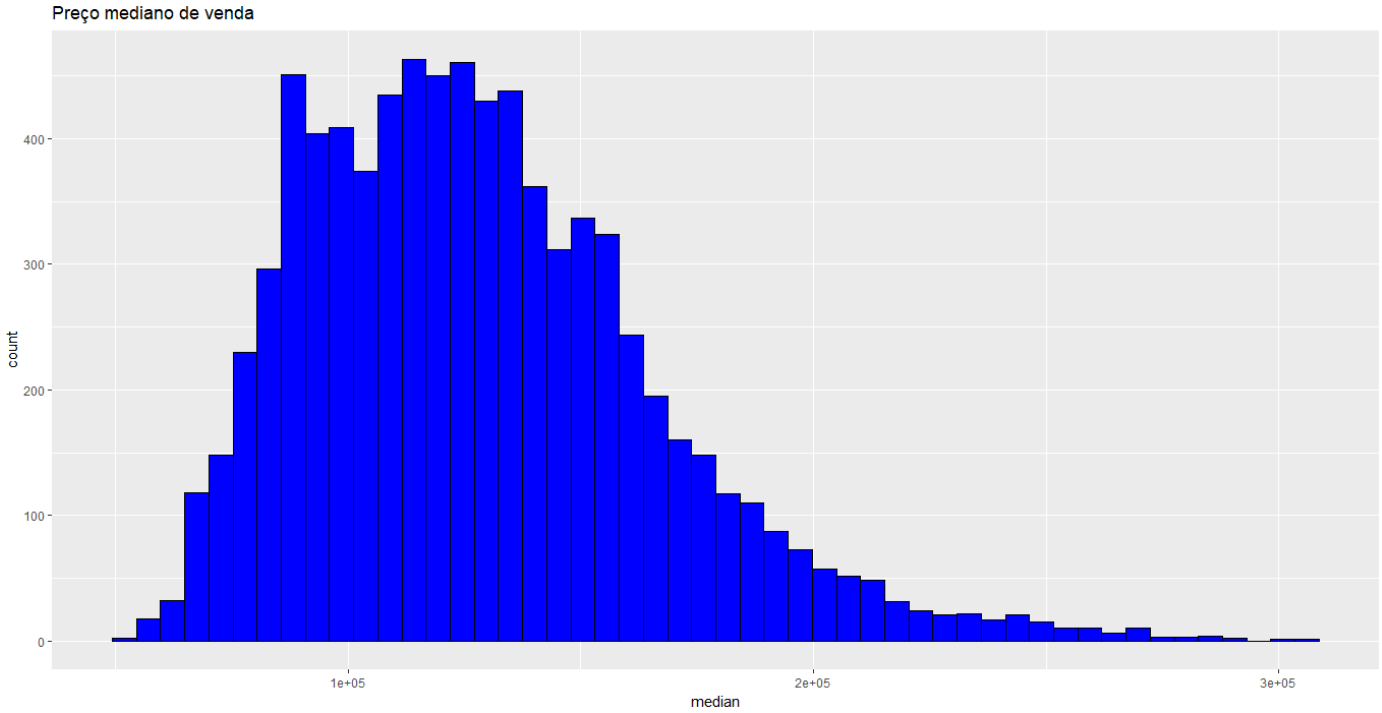


fig. Exemplo de histograma R

INTRODUÇÃO

Nesse breve tutorial Vou ensinar você a fazer um histograma no R usando a biblioteca *ggplot2*. Mas antes disso, deixe me mostrar alguns exemplos de utilização de um histograma para um analista ou cientista de dados:

- verificar se a variável tem distribuição normal para usar esse fato em um modelo de machine learning, um modelo de regressão por exemplo;
- verificar se a variável tem distribuição normal para aplicar um teste de hipóteses específico que necessita desse requisito;
- detecção de outliers, ou seja, dados que podem interferir nos resultados da análise;
- análise exploratória de dados;

Além disso, se você está fazendo sua tese, dissertação, TCC ou iniciação científica, os gráficos feitos no R via ggplot2 podem deixar seu trabalho bem mais profissional.

CONSTRUINDO UM HISTOGRAMA SIMPLES DE GGPLOT

Fazer qualquer tipo de visualização no ggplot2 pode parecer complicado de início. Porém, você perceberá que é bem fácil fazer um histograma e qualquer tipo de gráfico usando ggplot2!!

Nesse artigo, será mostrado como criar um histograma simples no ggplot2 e mostrar como editá-lo. Vamos lá!

Para construir um histograma usando o ggplot2, você precisa saber como o ggplot funciona. Não é muito difícil quando você pega o jeito, mas pode ser um pouco confuso no início.

Antes de começar, será instalado o ggplot2 e o pacote tidyverse. Também instalaremos o pacote txhousing, que contém o conjunto de dados de Vendas de Habitação no Texas em determinado período; usaremos esses dados nesse artigo.

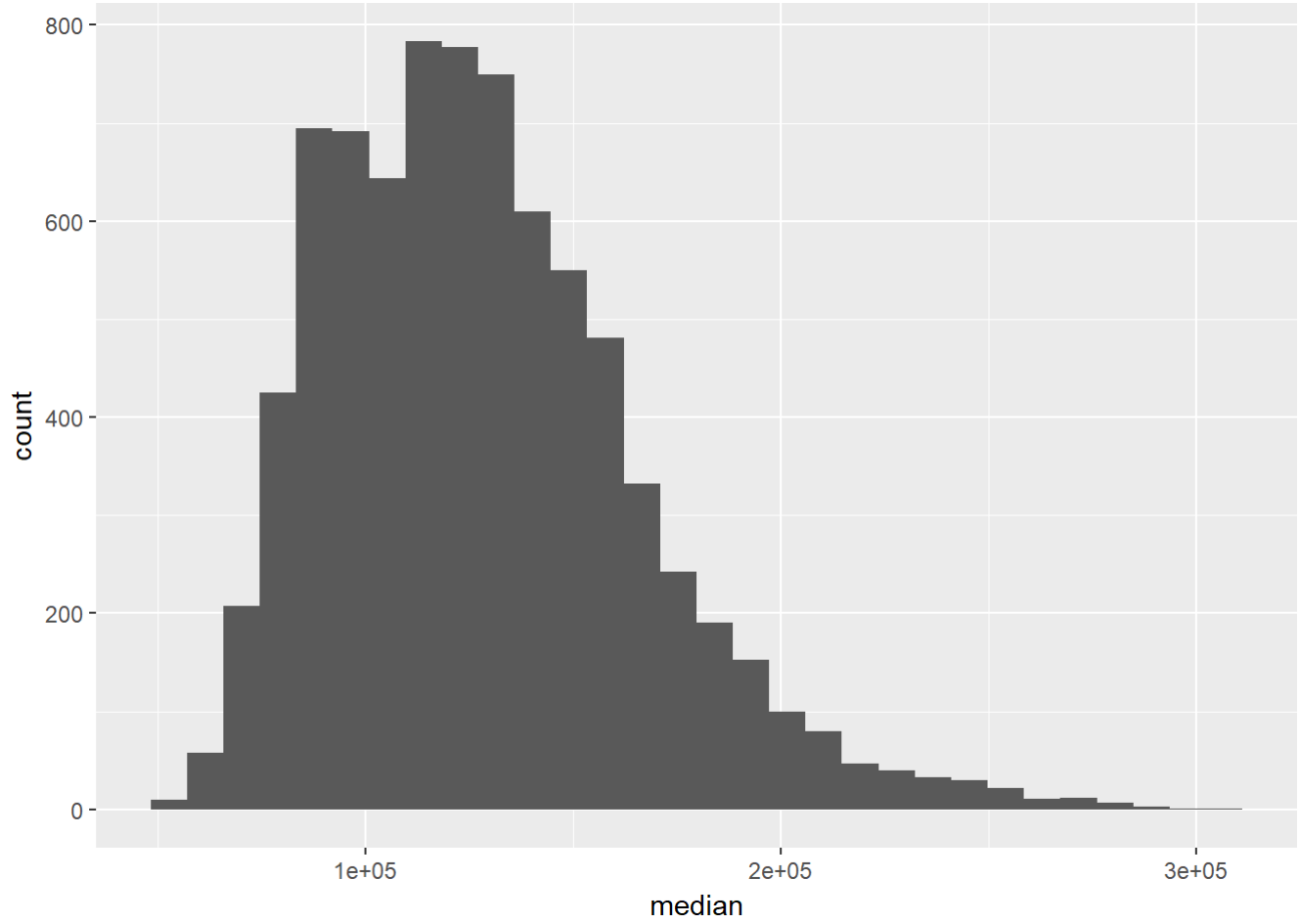
```
#chamar os pacotes:
library(tidyverse)
library(ggplot2)
#breve inspeção dos dados:
txhousing %>% glimpse
```

```
## Observations: 8,602
## Variables: 9
## $ city      <chr> "Abilene", "Abilene", "Abilene", "Abilene", "Abilene...
## $ year      <int> 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000, 2000...
## $ month     <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 1, 2, 3, 4, 5...
## $ sales     <dbl> 72, 98, 130, 98, 141, 156, 152, 131, 104, 101, 100, ...
## $ volume    <dbl> 5380000, 6505000, 9285000, 9730000, 10590000, 139100...
## $ median    <dbl> 71400, 58700, 58100, 68600, 67300, 66900, 73500, 750...
## $ listings  <dbl> 701, 746, 784, 785, 794, 780, 742, 765, 771, 764, 72...
## $ inventory <dbl> 6.3, 6.6, 6.8, 6.9, 6.8, 6.6, 6.2, 6.4, 6.5, 6.6, 6....
## $ date      <dbl> 2000.000, 2000.083, 2000.167, 2000.250, 2000.333, 20...
```

Agora, vamos fazer um histograma simples de ggplot:

```
#-----
# PLOT
#-----

# Histograma simples:
ggplot(data = txhousing, aes(x = median)) +
  geom_histogram()
```



Este histograma é bem simples de criar se você souber como funciona o ggplot2.

Mas, supondo que você não esteja familiarizado com o ggplot2, vamos rever rapidamente como ele funciona.

REVISÃO RÁPIDA: COMO FUNCIONA O SISTEMA GGPLOT2

Eu sou fã do ggplot2. Uso ele com muita frequência no meu trabalho. Isso se deve ao fato dele ser extremamente sistemático. Depois de saber como o sistema ggplot2 funciona, você pode criar praticamente qualquer visualização com relativa facilidade. Histogramas são apenas um exemplo e bem simples do poder dessa biblioteca do R.

A FUNÇÃO *GGPLOT ()*

A função ggplot() inicia a plotagem ggplot. Diz ao R que usaremos a biblioteca ggplot2 para construir uma visualização de dados.

FUNÇÃO *AES ()*

O aes () indica a escolha ou o mapeamento de variáveis que formarão a visualização.

Vamos dar uma olhada no nosso código de histograma novamente para tentar deixar isso mais claro.

```
# BASIC HISTOGRAM
#ggplot(data = txhousing, aes(x = median)) +
#  geom_histogram()
```

Observe que dentro do *aes()* está a expressão *x = median*. Ou seja, estamos escolhendo ou “mapeando” a variável mediana para o eixo x. Observe novamente que essa expressão aparece dentro da função aes (). Por quê? Porque é um mapeamento de variável. Todos os mapeamentos de conjuntos de dados para “atributos estéticos”, como o eixo x, ocorrem dentro da função aes ().

Isso pode ficar um pouco mais complicado se você quiser colocar mais atributos . Por exemplo, com um gráfico de dispersão, você mapeará uma variável para o eixo x e outra variável para o eixo y. Com técnicas avançadas de visualização, pode ficar ainda mais complicado. Mas fique tranquilo, aqui você aprenderá o básico.

Resumindo como aes() trabalha:*um conjunto de dados tem variáveis, uma visualização tem atributos estéticos como o eixo x, eixo y, cor, forma, etc. Precisamos "conectar" as variáveis aos atributos estéticos. Isso é feito no ggplot2 com a função aes ().*

PASSO A PASSO DE UM HISTOGRAMA NO GGPLOT2

A função *ggplot()* inicia a plotagem. A função aes () especifica como queremos "mapear" ou "conectar" variáveis em nosso conjunto de dados aos atributos estéticos.

Com esse conhecimento em mente, vamos revisitar nosso histograma ggplot e entender cada parte do código.

```
#-----
# PLOT
#-----

# BASIC HISTOGRAM
#ggplot(data = txhousing, aes(x = median)) +
#  geom_histogram()
```

o que está ocorrendo no código acima?

ggplot () indica que vamos traçar algo. O data = parâmetro indica que vamos plotar os dados do conjunto de dados de txhousing. Dentro da função aes (), estamos especificando que queremos colocar a variável "mediana" no eixo x. Finalmente, geom_histogram () indica que vamos traçar um histograma.

Obs: Nesse artigo não vou entrar em detalhes no que significa o "geom". Só adianto que existem muitos "geoms"" diferentes no ggplot2 que criam diferentes tipos de gráficos

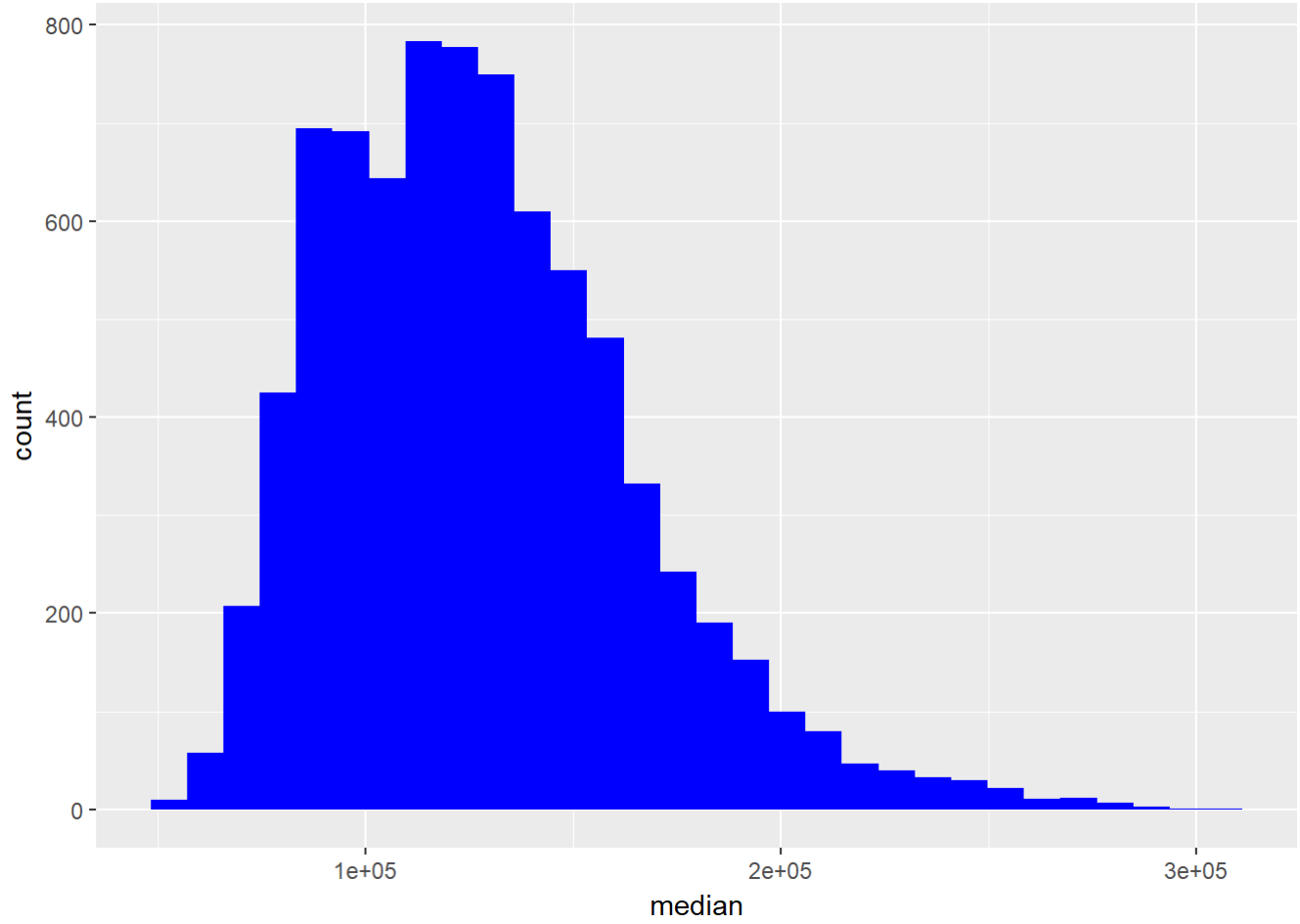
MODIFICANDO UM HISTOGRAMA NO GGPLOT2

Vamos agora fazer algumas modificações no histograma

MUDAR AS CORES DAS BARRAS

Vamos pegar o histograma que fizemos anteriormente, e vamos adicionar um pequeno trecho de código dentro de geom_histogram (), vamos adicionar o código fill = 'blue'. Isso mudará a cor de preenchimento interior de todas as barras do histograma para a cor azul. Veja:

```
# adicionando cor
# - preenchendo as barra
ggplot(data = txhousing, aes(x = median)) +
  geom_histogram(fill = 'blue')
```

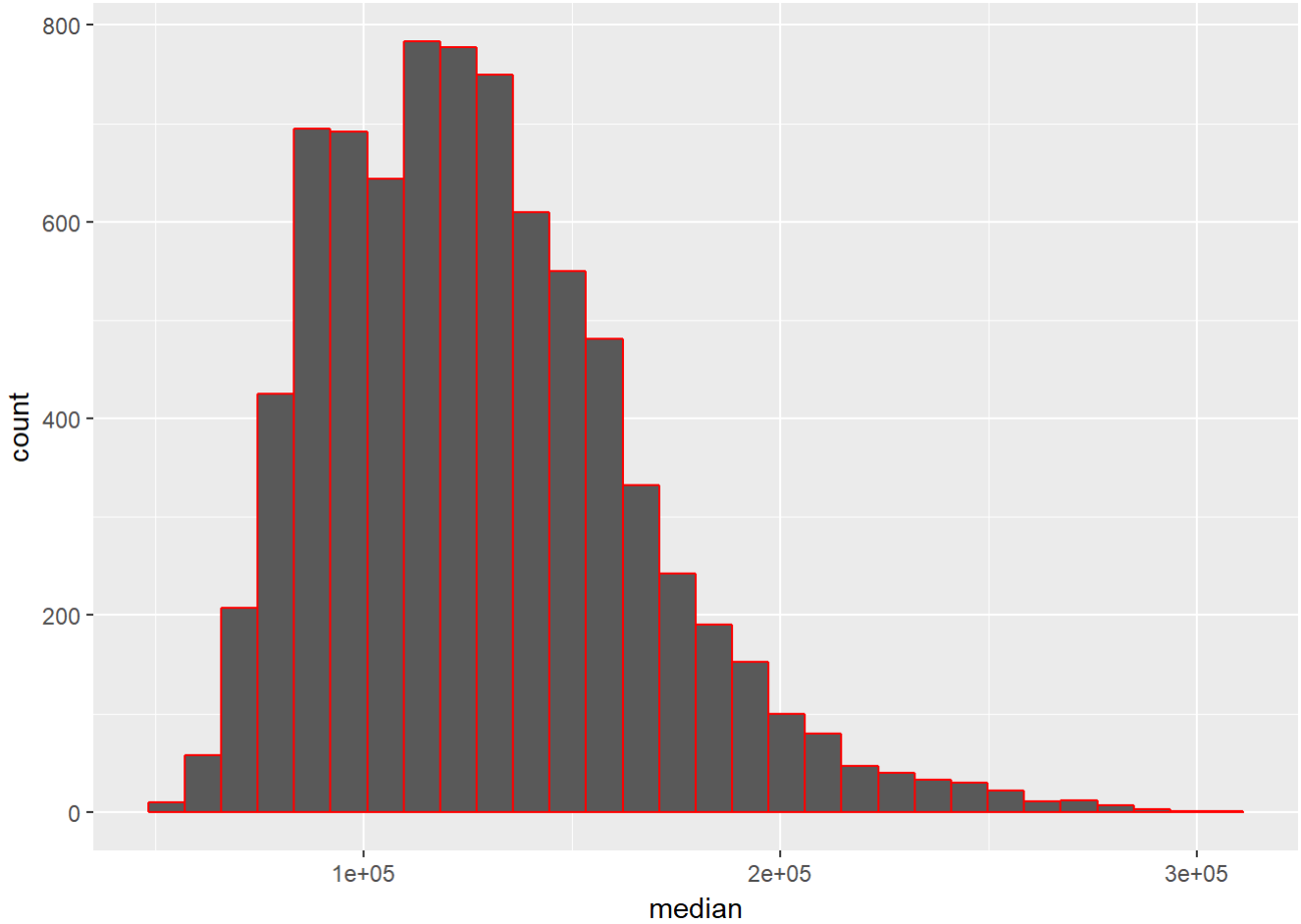


Não foi simples?! Como dica, eu recomendo que você aprenda ggplot e o programa R dessa forma, ou seja,comece com uma técnica simples. Aprenda. Domine a técnica. Então faça passo a passo pequenas mudanças (e domine como fazer essas mudanças). Comece simples e expanda sua habilidade em um assunto específico até dominá-lo por completo.

MUDAR AS CORES DA BORDAS

Isso é muito parecido com a alteração da cor da barra, mas em vez de usar o parâmetro fill =, usaremos o parâmetro color =.

```
# adicionando cor
# - color das bordas das barras
ggplot(data = txhousing, aes(x = median)) +
  geom_histogram(color = 'red')
```



Observe que o histograma é basicamente o mesmo, apenas cor das barras foi alterado.

ALTERAR O NÚMERO DE BARRAS NO HISTOGRAMA

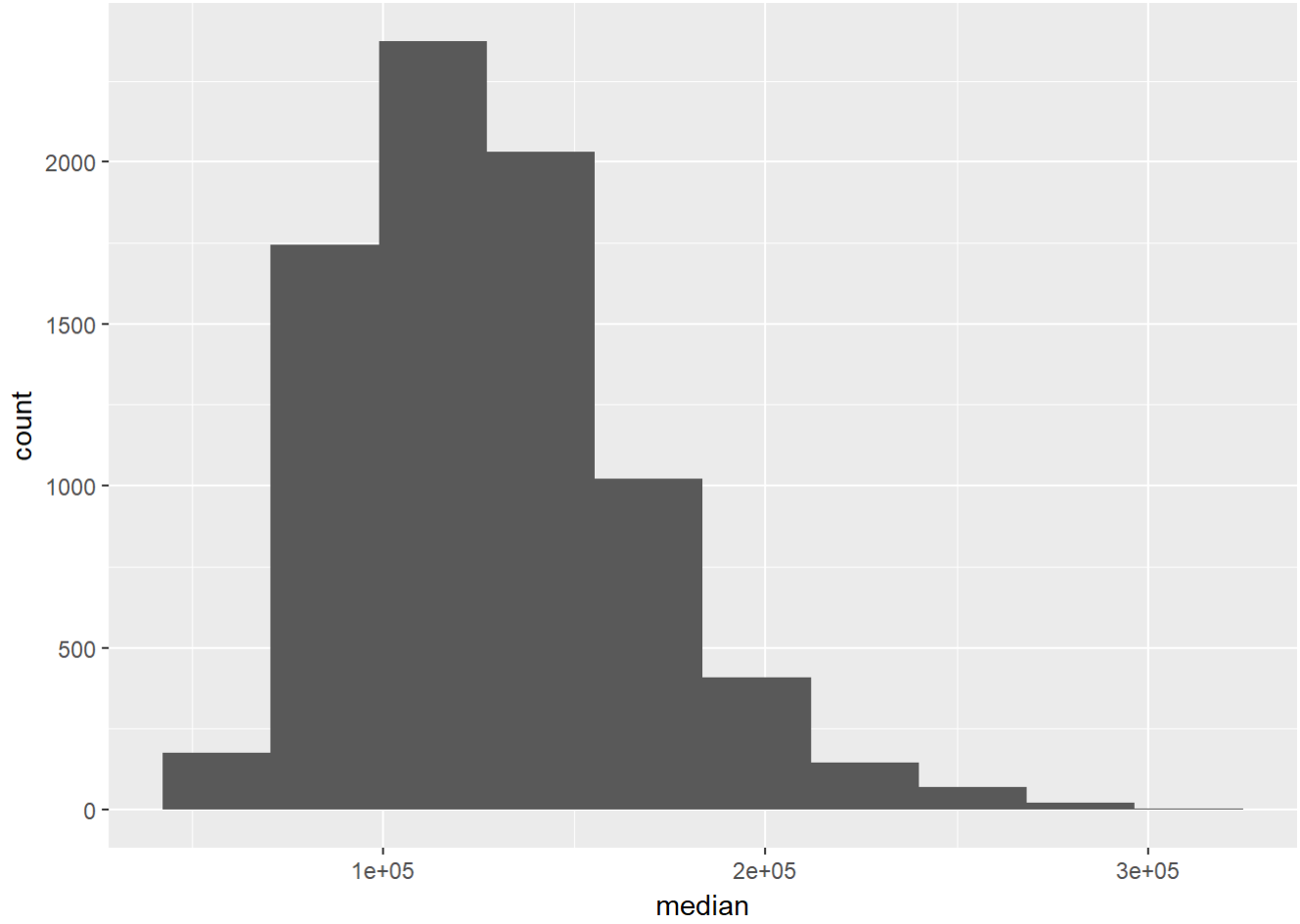
Por padrão, o ggplot2 usará 30 faixas ou intervalos no histograma. Porém, isso pode ser alterado pelo usuário. Isso pode ser útil dependendo de como os dados são distribuídos. Se houver muita variabilidade nos dados, podemos usar um número maior de compartimentos para ver algumas dessas variações. Ou podemos usar um número menor de intervalos para “suavizar” a variabilidade.

De qualquer forma, para alterar o número de barras basta usar o parâmetro bins =.

Vamos alterar de duas formas

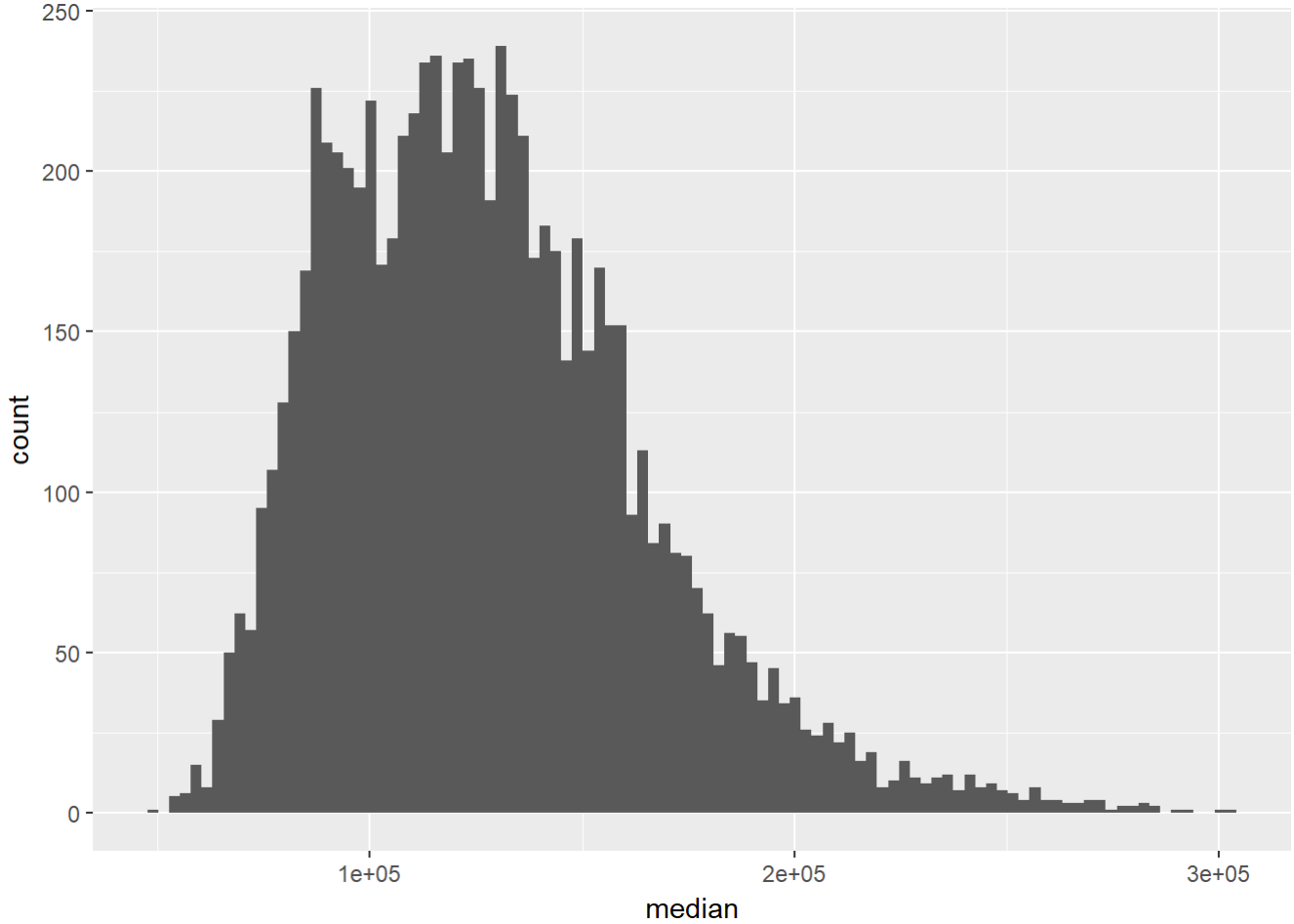
1) quantidade menor de barras

```
# USANDO NUMERO MENOR DE BARRAS
ggplot(data = txhousing, aes(x = median)) +
  geom_histogram(bins = 10)
```



2) quantidade maior de barras

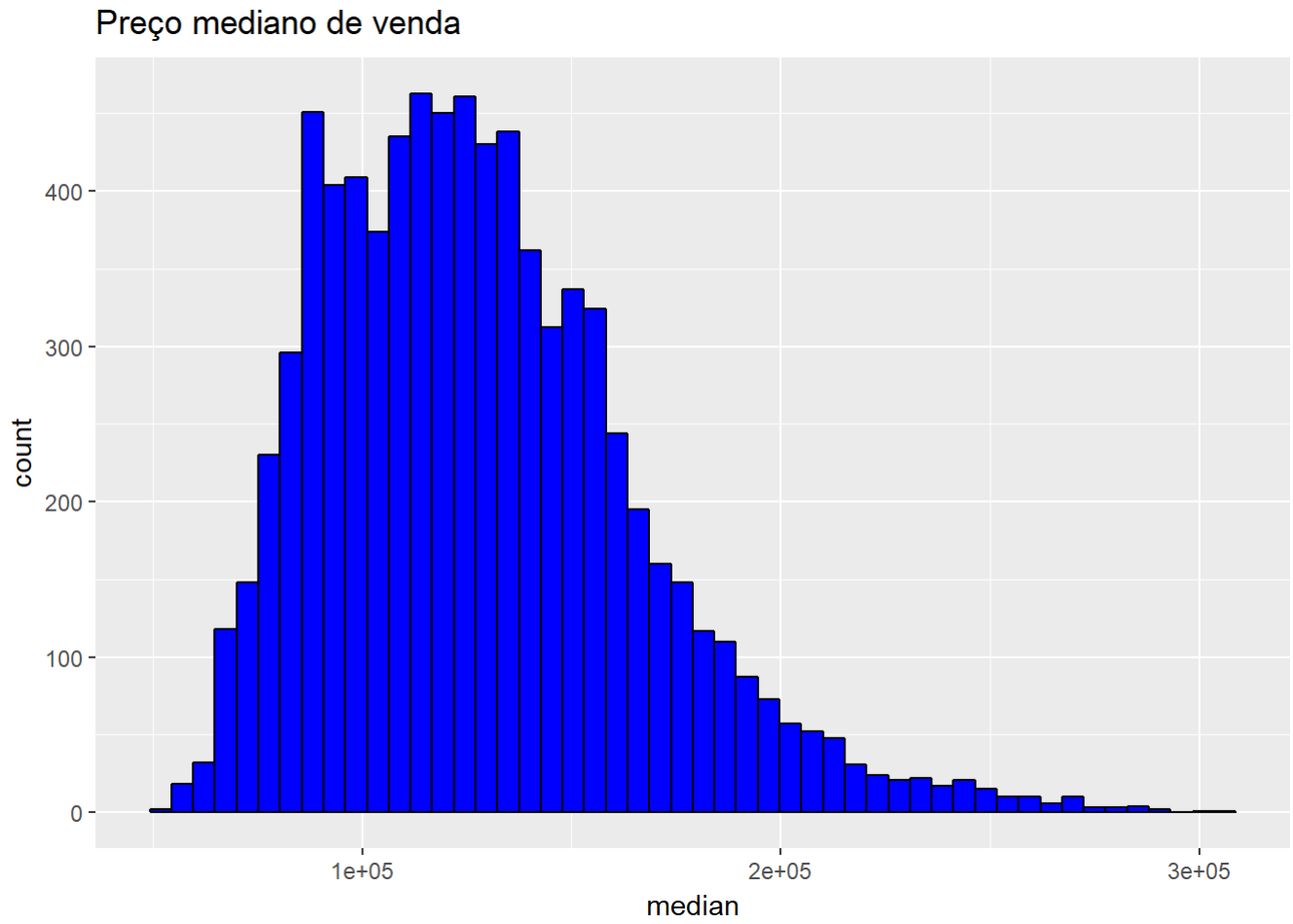
```
# USANDO NUMERO MAIOR DE BARRAS
ggplot(data = txhousing, aes(x = median)) +
  geom_histogram(bins = 100)
```



Você pode estar pensando, quantas barras usar? Isso depende de quais são seus objetivos. Pessoalmente, neste caso, 30 barras funcionam bem, mas, novamente, depende do seu objetivo.

Agora vou fazer um histograma misturando várias coisas que vimos:

```
# USANDO NUMERO MENOR DE BARRAS
ggplot(data = txhousing, aes(x = median)) +
  geom_histogram(bins=50, fill='blue', color ='black')+
  ggtitle("Preço mediano de venda")
```



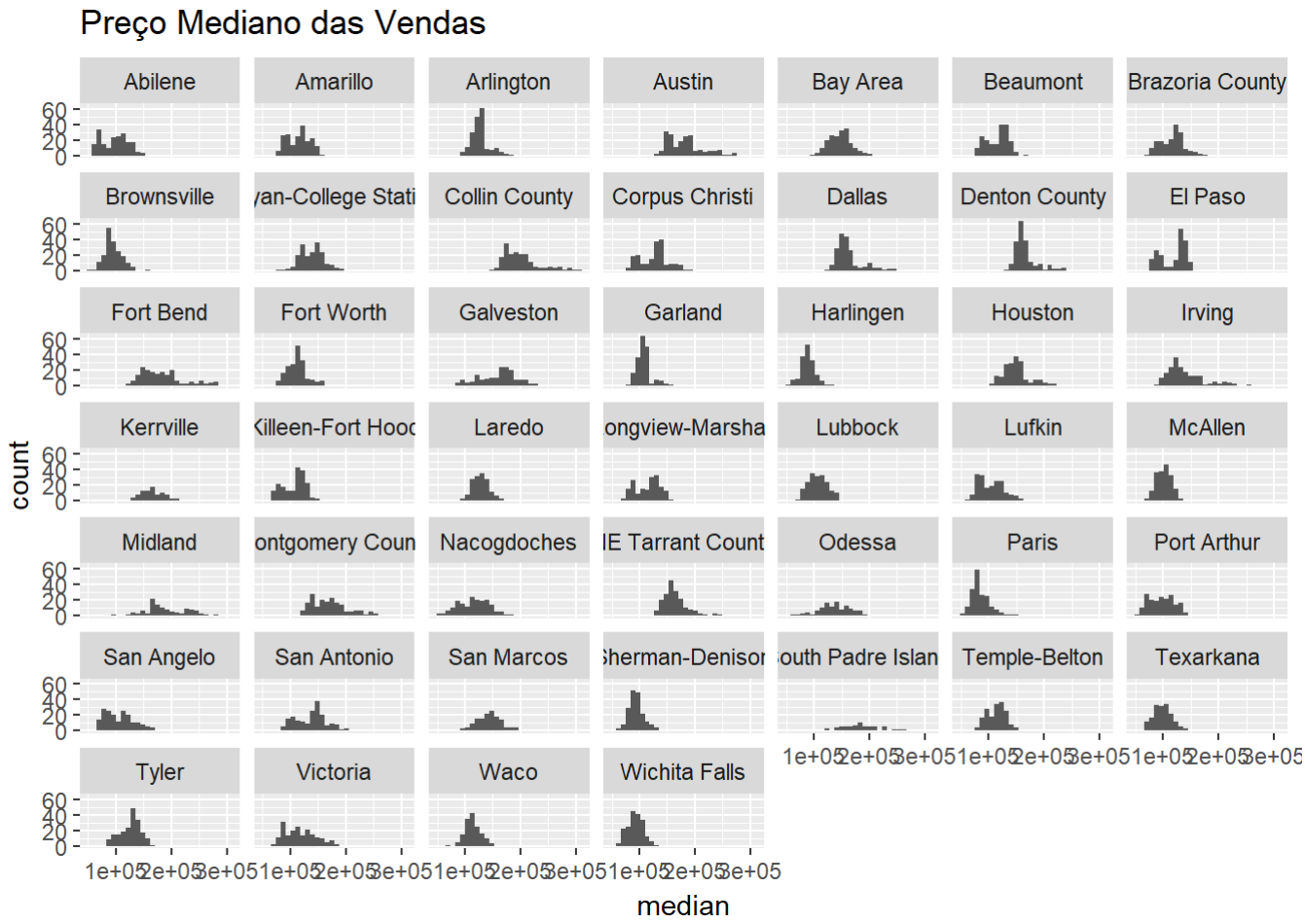
BÔNUS: COMO FAZER MÚLTIPLOS MINI HISTOGRAMAS COM BASE EM UMA VARIÁVEL CATEGÓRICA

O ggplot2 facilita muitas coisas quando o assunto é visualização. Um ótimo exemplo disso são os mini gráficos múltiplos. É extremamente útil para uma variedade de tarefas de análise de dados e ciência de dados. Mas você raramente os vê porque são difíceis de criar em outro software, mas no R é muito simples.

Para criar um gráfico desse tipo no ggplot2, adicionamos apenas um pedaço de código que “quebrará” o gráfico com base em uma variável categórica.

Aqui, usaremos o código facet_wrap (~ city) para fazer uma pequena versão do gráfico para cada valor da variável city.

```
# MULTIPLOS MINI HISTOGRAM
ggplot(data = txhousing, aes(x = median)) +
  geom_histogram() + ggtitle('Preço Mediano das Vendas')+
  facet_wrap(~city)
```



Não é interessante! Observe que há muitos dados e muitos detalhes. Será mais fácil ver se você executa o código em seu próprio computador e dar um zoom no gráfico. Faça isso!

O que é legal sobre o esses gráfico múltiplos é que você vê muitas informações em um espaço muito pequeno.

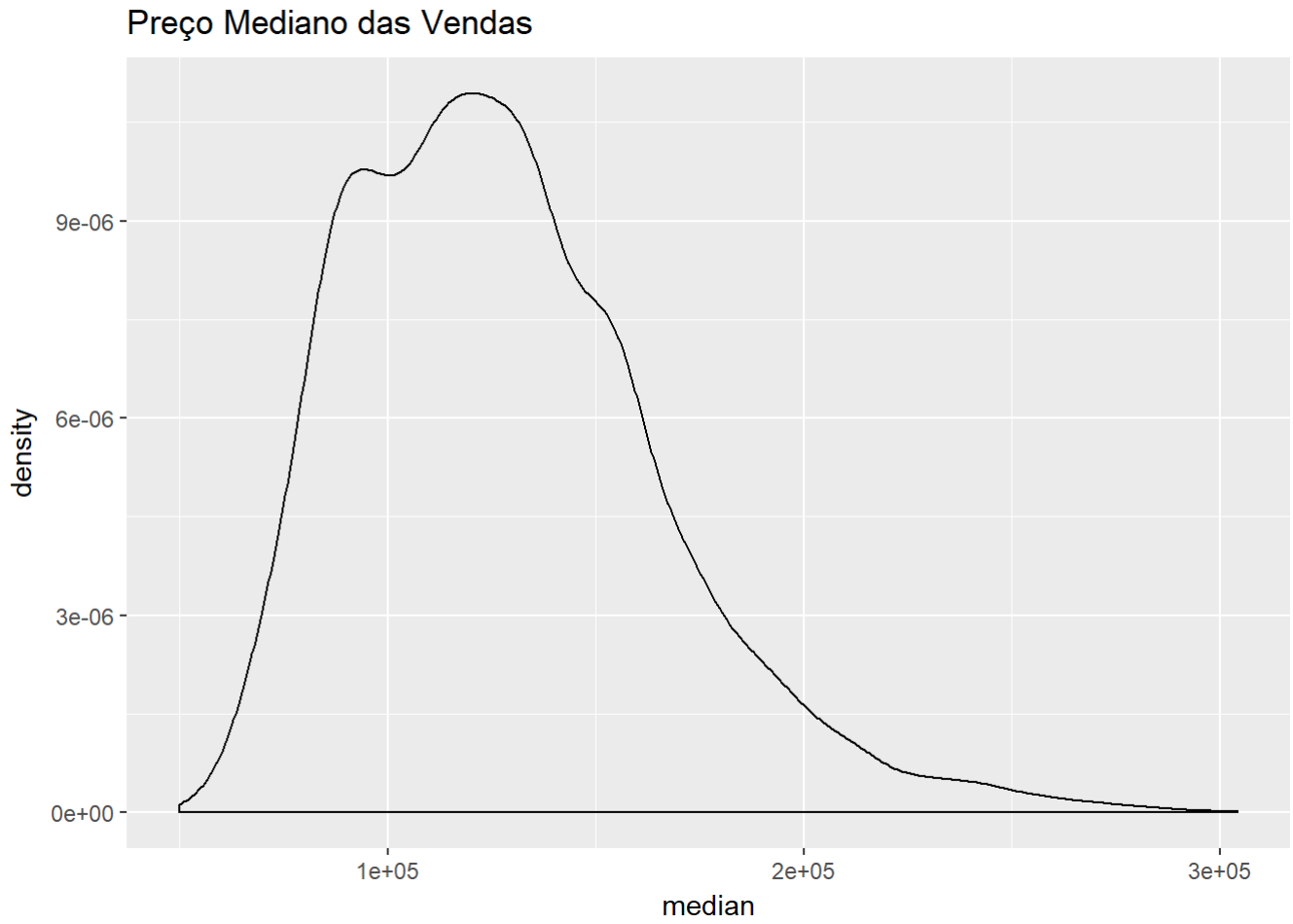
Neste gráfico, podemos ver histogramas individuais para cada cidade. Isso pode ser muito útil se você estivesse fazendo uma análise sobre as cidades e como elas são diferentes, fazer comparação de perfomances, etc.

BÔNUS: COMO FAZER UM GRÁFICO DE DENSIDADE

O gráfico de densidade é apenas uma variação do histograma, mas em vez do eixo y mostrar o número de observações, mostra a “densidade” dos dados.

No ggplot2, o gráfico de densidade é realmente muito fácil de criar. Apenas altere o código para o histograma básico que usamos acima e troque geom_histogram () por geom_density ().

```
# GRAFICO DE DENSIDADE
ggplot(data = txhousing, aes(x = median)) +
  geom_density() + ggtitle('Preço Mediano das Vendas')
```



Depois de saber o básico, alterar um histograma para um gráfico de densidade é tão fácil quanto alterar uma linha de código.

Espero que tenha gostado, até o próximo!

Keep calm and analysing data!