

Atividade 2 - Respostas

Questão 1: Na tabela abaixo são dadas as notas de dois alunos em três provas que tinham pesos 2, 3 e 5, respectivamente. Calcule as medianas e as modas dos dois alunos.

Aluno	1ª Prova	2ª Prova	3ª Prova
Marcos	7	6	5
Pedro	1	2	9

- Script

 R

```
# Atividade 2 - Questão 1

# carregando os pacotes utilizados
library(dplyr)
library(tibble)
library(tidyr)  # <-- ESTE AQUI!
library(modeest)

# inserindo dados dos alunos
notas <- tribble(~Aluno, ~Prova1, ~Prova2, ~Prova3,
                 "Marcos", 7, 6, 5,
                 "Pedro", 1, 2, 9)

# transformando os dados para formato longo
notas_long <- notas %>%
  pivot_longer(cols = starts_with("Prova"), names_to = "Prova",
               values_to = "Nota")

# calculando a mediana e moda por aluno
resultados <- notas_long %>%
  group_by(Aluno) %>%
  summarise(
    Mediana = median(Nota),
    Moda = mfv(Nota)[1] # mfv retorna todas as modas, sendo retornado o
                        # primeiro
  )

# imprimindo os objetos criados
print(notas)
print(notas_long)
print(resultados)
```

- Explicação do Script

- Etapa 1: carregando os pacotes

 R

```
library(dplyr)
library(tibble)
library(tidyr)
library(modeest)
```

- dplyr: manipulação e agrupamento dos dados como a função `group_by()`.
- tibble: criar data frames modernos (tipo `tribble()`).
- tidyr: para "pivotar" os dados de formato largo para longo como a função `pivot_longer()`.
- modeest: fornece a função `mfv()` que é uma função robusta para cálculo da moda.

- **Etapa 2: carregando os dados dos alunos**

 R

```
notas <- tribble(~Aluno, ~Prova1, ~Prova2, ~Prova3,
                 "Marcos", 7, 6, 5,
                 "Pedro", 1, 2, 9)
```

- Cria uma tabela (tibble) com as notas por prova dos dois alunos.
- Está no formato largo, ou seja, uma linha por aluno, uma coluna por prova.

- **Etapa 3: transformando em formato longo**

 R

```
notas_long <- notas %>%
  pivot_longer(cols = starts_with("Prova"), names_to = "Prova",
               values_to = "Nota")
```

- Agrupa por aluno e aplica as funções estatísticas.

- **Etapa 4: calculando a mediana e a moda**

 R

```
resultados <- notas_long %>%
  group_by(Aluno) %>%
  summarise(
    Mediana = median(Nota),
    Moda = mfv(Nota)[1]
  )
```

- `group_by(Aluno)`: agrupa os dados por aluno.
- `median(Nota)`: calcula a mediana das três notas.
- `mfv(Nota)`: calcula a(s) moda(s) (*Most Frequent Value*).
- `[1]`: em caso de empate, retorna apenas a primeira moda para manter o resultado unívoco.

- **Etapa 5: imprimindo os resultados**

 R

```
print(notas)
print(notas_long)
print(resultados)
```

- notas: para ver a estrutura original.
 - notas_long: para confirmar a transformação.
 - resultados: que contém as estatísticas finais.
-
- **Saída no console do RStudio**

Shell

```
> # Atividade 2 - Questão 1
>
> # importando os pacotes utilizados
> library(dplyr)
> library(tibble)
> library(tidyr) # <-- ESTE AQUI!
> library(modeest)
>
> # inserindo dados dos alunos
> notas <- tribble(~Aluno, ~Prova1, ~Prova2, ~Prova3,
+                 "Marcos", 7, 6, 5,
+                 "Pedro", 1, 2, 9)
> # transformando os dados para formato longo
> notas_long <- notas %>%
+   pivot_longer(cols = starts_with("Prova"), names_to = "Prova",
+               values_to = "Nota")
> # calculando a mediana e moda por aluno
> resultados <- notas_long %>%
+   group_by(Aluno) %>%
+   summarise(
+     Mediana = median(Nota),
+     Moda = mfv(Nota)[1] # mfv retorna todas as modas, sendo retornado o
+                       # primeiro
+   )
>
> # imprimindo os objetos criados
> print(notas)
# A tibble: 2 × 4
  Aluno Prova1 Prova2 Prova3
<chr>   <dbl> <dbl> <dbl>
1 Marcos     7     6     5
2 Pedro     1     2     9
> print(notas_long)
# A tibble: 6 × 3
  Aluno Prova  Nota
<chr> <chr> <dbl>
1 Marcos Prova1     7
2 Marcos Prova2     6
3 Marcos Prova3     5
4 Pedro  Prova1     1
5 Pedro  Prova2     2
6 Pedro  Prova3     9
> print(resultados)
# A tibble: 2 × 3
  Aluno Mediana Moda
<chr>   <dbl> <dbl>
1 Marcos     6     5
2 Pedro     2     1
```

Questão 2: Considere os seguintes tempos (em minutos) que um grupo de estudantes levou para completar uma prova: 22, 24, 25, 27, 28, 29, 30, 31, 33, 35, 90

- **Script**

```
R
# carregando os pacotes utilizados
library("ggplot2")
library("dplyr")

# inserindo os dados dos tempos em minutos
tempos <- c(22, 24, 25, 27, 28, 29, 30, 31, 33, 35, 90)

# realizando os cálculos estatísticos com os dados amostrais
minimo <- min(tempos)
maximo <- max(tempos)
mediana <- median(tempos)
q1 <- quantile(tempos, 0.25)
q2 <- quantile(tempos, 0.50)
q3 <- quantile(tempos, 0.75)
iqr <- IQR(tempos)
limite_inferior <- q1 - 1.5 * iqr
limite_superior <- q3 + 1.5 * iqr
outliers <- tempos[tempos < limite_inferior | tempos > limite_superior]

# imprimindo os objetos
cat("Mediana = ", mediana, "\n")
cat("Quartil q1 = ", q1, "\n")
cat("Quartil q2 = ", q2, "\n")
cat("Quartil q3 = ", q3, "\n")
cat("Limite inferior = ", limite_inferior, "\n")
cat("Limite superior = ", limite_superior, "\n")
cat("Outliers = ", outliers, "\n")

# criando um dataframe
df <- tibble::tibble(Tempos=tempos)

# plotando o gráfico boxplot
ggplot(df, aes(x = "", y = Tempos)) +
  geom_boxplot(fill = "skyblue", outlier.color = "red", outlier.shape = 8) +
  labs(title = "Boxplot dos Tempos", y = "Minutos") +
  coord_cartesian(ylim = c(minimo, maximo)) +
  theme_minimal()
```

- **Explicação do Script**

- **Etapa 1: carregando os pacotes**

R

```
library("ggplot2")  
library("dplyr")
```

- ggplot2: utilizado para criar gráficos elegantes e personalizáveis — neste caso, um boxplot.
- dplyr: carregado, mas não é utilizado diretamente no script atual (está preparado para manipulações de dados se necessário).

- **Etapa 2: carregando os dados**

R

```
tempos <- c(22, 24, 25, 27, 28, 29, 30, 31, 33, 35, 90)
```

- Vetor tempos armazena os tempos (em minutos) que os estudantes levaram para completar uma prova.
- Contém um valor discrepante: 90, que será identificado como outlier.

- **Etapa 3: calculando os valores estatísticos descritivos**

R

```
minimo <- min(tempos)  
maximo <- max(tempos)  
mediana <- median(tempos)  
q1 <- quantile(tempos, 0.25)  
q2 <- quantile(tempos, 0.50)  
q3 <- quantile(tempos, 0.75)  
iqr <- IQR(tempos)
```

- min, max: definem o intervalo [mínimo, máximo] dos dados.
- median, quantile: capturam os valores da mediana e quartis.
- iqr: calcula o intervalo interquartil = $Q3 - Q1$. Usado para detecção de outliers.

- **Etapa 4: calculando os limites para detecção de outliers**

R

```
limite_inferior <- q1 - 1.5 * iqr  
limite_superior <- q3 + 1.5 * iqr  
outliers <- tempos[tempos < limite_inferior | tempos > limite_superior]
```

- Limites teóricos definidos pela fórmula padrão:
 - Inferior = $Q1 - 1.5 \times IQR$
 - Superior = $Q3 + 1.5 \times IQR$
- Qualquer valor fora desses limites é classificado como outlier.
- No caso em questão, o valor 90 será detectado como fora do limite superior, e isolado no vetor denominado outliers.

- **Etapa 5: imprimindo os valores estatísticos**

 R

```
cat("Mediana = ", mediana, "\n")
cat("Quartil q1 = ", q1, "\n")
cat("Quartil q2 = ", q2, "\n")
cat("Quartil q3 = ", q3, "\n")
cat("Limite inferior = ", limite_inferior, "\n")
cat("Limite superior = ", limite_superior, "\n")
cat("Outliers = ", outliers, "\n")
```

- Uso a função `cat()` para exibir no console os valores estatísticos obtidos.

- **Etapa 6: preparando os dados para a visualização gráfica**

 R

```
df <- tibble::tibble(Tempos = tempos)
```

- Converte o vetor denominado `tempos` em um data frame do tipo `tibble`, com uma coluna chamada `Tempos`.
- Isso foi necessário para que a função `ggplot()` possa mapear os dados corretamente no gráfico.

- **Etapa 7: construindo o boxplot dos dados**

 R

```
ggplot(df, aes(x = "", y = Tempos)) +
  geom_boxplot(fill = "skyblue", outlier.color = "red", outlier.shape = 8) +
  labs(title = "Boxplot dos Tempos", y = "Minutos") +
  coord_cartesian(ylim = c(minimo, maximo)) +
  theme_minimal()
```

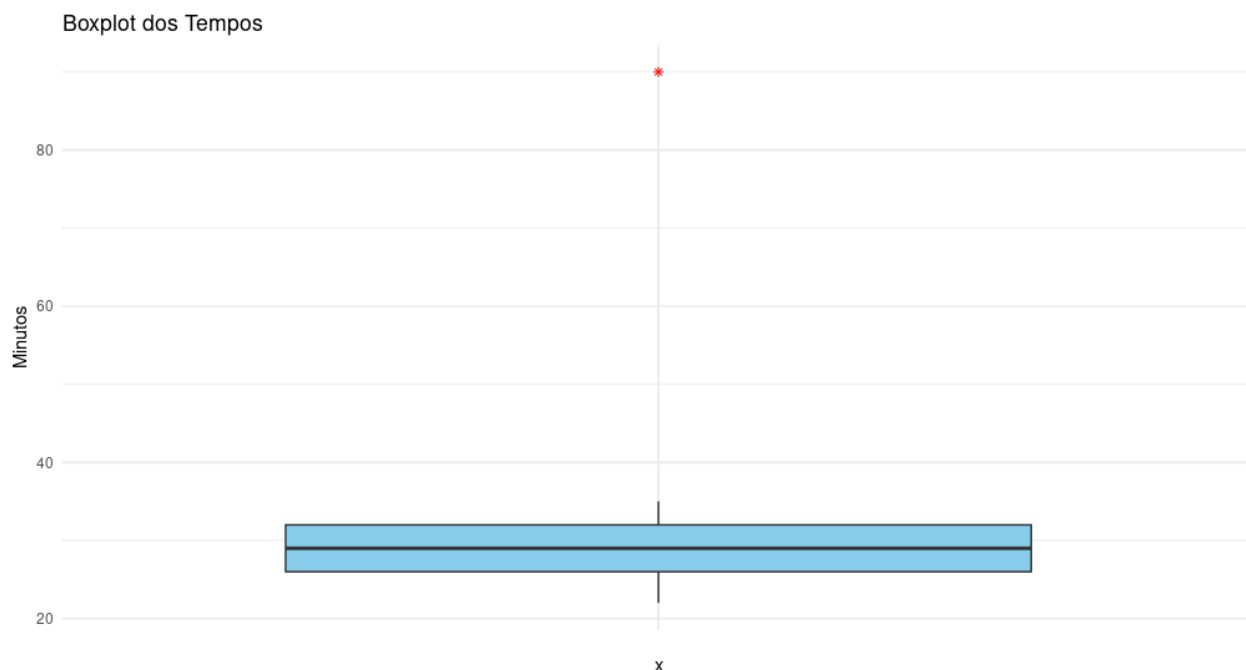
- `aes(x = "", y = Tempos)`: a variável `Tempos` será o eixo vertical, e um valor fixo `""` no eixo x cria um único boxplot.
- `geom_boxplot()`: gera o boxplot:
 - `fill`: cor da caixa.
 - `outlier.color = "red"`: destaca outliers em vermelho.
 - `outlier.shape = 8`: usa o símbolo de asterisco (*).
- `labs()`: define título e rótulo do eixo y.
- `coord_cartesian(ylim = c(minimo, maximo))`: define o intervalo visível do eixo y, preservando os dados fora do intervalo (como o outlier 90).
- `theme_minimal()`: aplica um tema visual limpo ao gráfico.

- **Saída no Console do RStudio**

Shell

```
> # carregando os pacotes utilizados
> library("ggplot2")
> library("dplyr")
>
> # inserindo os dados dos tempos em minutos
> tempos <- c(22, 24, 25, 27, 28, 29, 30, 31, 33, 35, 90)
>
> # realizando os cálculos estatísticos com os dados amostrais
> minimo <- min(tempos)
> maximo <- max(tempos)
> mediana <- median(tempos)
> q1 <- quantile(tempos, 0.25)
> q2 <- quantile(tempos, 0.50)
> q3 <- quantile(tempos, 0.75)
> iqr <- IQR(tempos)
> limite_inferior <- q1 - 1.5 * iqr
> limite_superior <- q3 + 1.5 * iqr
> outliers <- tempos[tempos < limite_inferior | tempos > limite_superior]
>
> # imprimindo os objetos
> cat("Mediana = ", mediana, "\n")
Mediana = 29
> cat("Quartil q1 = ", q1, "\n")
Quartil q1 = 26
> cat("Quartil q2 = ", q2, "\n")
Quartil q2 = 29
> cat("Quartil q3 = ", q3, "\n")
Quartil q3 = 32
> cat("Limite inferior = ", limite_inferior, "\n")
Limite inferior = 17
> cat("Limite superior = ", limite_superior, "\n")
Limite superior = 41
> cat("Outliers = ", outliers, "\n")
Outliers = 90
>
> # criando um dataframe
> df <- tibble::tibble(Tempos=tempos)
>
> # plotando o gráfico boxplot
> ggplot(df, aes(x = "", y = Tempos)) +
+   geom_boxplot(fill = "skyblue", outlier.color = "red", outlier.shape = 8) +
+   labs(title = "Boxplot dos Tempos", y = "Minutos") +
+   coord_cartesian(ylim = c(minimo, maximo)) +
+   theme_minimal()
>
```

- Saída na aba Plots do RStudio



a) Identifique a mediana, os quartis, os limites inferior e superior e os possíveis outliers.

- mediana (q2) = 29
- q1 = 26
- q3 = 32
- limite inferior = 17
- limite superior = 41
- outliers: 90

b) Explique como o boxplot representa graficamente a presença de *outliers* nesse caso.

O *boxplot* comunica visualmente que quase todos os estudantes realizaram a prova em um tempo relativamente próximo (distribuição concentrada entre ~25 e ~35 minutos), com uma única exceção extrema (o estudante que levou 90 minutos). O uso de um ponto vermelho em formato de um asterisco (*outlier.color* e *outlier.shape*) para marcar o *outlier* é uma convenção gráfica clara, objetiva e eficaz para destacar anomalias nos dados.

Questão 3: Um órgão do governo do estado está interessado em determinar padrões sobre o investimento em educação, por habitante, realizado pelas prefeituras. De um levantamento amostral de dez cidades, foram obtidos os valores (codificados) do quadro abaixo:

Cidade	A	B	C	D	E	F	G	H	I	J
Investimento	20	16	14	8	19	15	14	16	19	18

Nesse caso será considerado como investimento básico a média final das observações calculada da seguinte maneira:

1. Obtém-se uma média inicial;
2. Eliminam-se do conjunto aquelas observações que forem superiores à média inicial mais duas vezes o desvio padrão, ou inferiores à média inicial menos duas vezes o desvio padrão;
3. Calcula-se a média final com o novo conjunto de observações.

- Script em R

R

```
# carregando os pacotes utilizados
library(dplyr)
library(ggplot2)
library(tibble)

# carregando os dados
investimento <- tibble::tibble(
  Cidade = LETTERS[1:10],
  Valor = c(20, 16, 14, 8, 19, 15, 14, 16, 19, 18)
)

# calculando a média inicial
media_inicial <- mean(investimento$Valor)

# calculando o desvio-padrão
desvio <- sd(investimento$Valor)

# calculando o limite inferior
lim_inf <- media_inicial - 2 * desvio

# calculando o limite superior
lim_sup <- media_inicial + 2 * desvio

# filtrando os dados
filtrado <- investimento %>%
  filter(Valor >= lim_inf & Valor <= lim_sup)

filtrado

# calculando a média final
media_final <- mean(filtrado$Valor)

# imprimindo os resultados dos objetos criados
cat("Média inicial das amostras: ", media_inicial, "\n")
cat("Desvio-padrão das amostras: ", desvio, "\n")
cat("Limite inferior: ", lim_inf, "\n")
cat("Limite superior: ", lim_sup, "\n")
cat("Média final: ", media_final, "\n")

# visualizando os dados
ggplot(investimento, aes(x = reorder(Cidade, Valor), y = Valor)) +
  geom_col(fill = ifelse(investimento$Valor >= lim_inf
    & investimento$Valor <= lim_sup, "steelblue", "tomato")) +
  geom_hline(yintercept = media_final, linetype = "dashed", color = "black") +
  labs(title = "Investimento por Cidade", y = "Valor",
    x = "Cidade") +
  theme_minimal()
```

- **Explicação do Script**

- **Etapa 1: Carregamento dos pacotes utilizados**

```
R  
library(dplyr)  
library(ggplot2)  
library(tibble)
```

Esses pacotes são usados para:

- dplyr: manipulação de dados (como filter()).
- ggplot2: construção de visualizações (gráfico de colunas).
- tibble: criação moderna e legível de tabelas (tibble()).

- **Etapa 2: carregando os dados**

```
R  
investimento <- tibble::tibble(  
  Cidade = LETTERS[1:10],  
  Valor = c(20, 16, 14, 8, 19, 15, 14, 16, 19, 18)  
)
```

- Cria um data frame com:
 - Cidades de A a J.
 - Investimentos correspondentes (valores codificados).

- **Etapa 3: calculando as estatísticas descritivas iniciais**

```
R  
media_inicial <- mean(investimento$Valor)  
desvio <- sd(investimento$Valor)
```

- média_inicial: calcula a média aritmética dos investimentos.
- desvio: calcula o desvio padrão, que mede a dispersão dos valores.

- **Etapa 4: calculando os limites de corte**

```
R  
lim_inf <- media_inicial - 2 * desvio  
lim_sup <- media_inicial + 2 * desvio
```

- Esses limites definem um intervalo [lim_inf, lim_sup].
- Valores fora desse intervalo serão considerados atípicos (*outliers*).

- **Etapa 5: filtrando as amostras iniciais**

R

```
filtrado <- investimento %>%  
  filter(Valor >= lim_inf & Valor <= lim_sup)
```

- Remove valores menores que o limite inferior ou maiores que o limite superior.
- Gera uma nova tabela denominado filtrado apenas com os valores "aceitáveis".

- **Etapa 6: calculando a média final**

R

```
media_final <- mean(filtrado$Valor)
```

- A média é recalculada com base somente nos dados filtrados.
- A nova média é o investimento básico.

- **Etapa 7: Imprimindo os resultados obtidos**

R

```
cat("Média inicial das amostras: ", media_inicial, "\n")  
cat("Desvio-padrão das amostras: ", desvio, "\n")  
cat("Limite inferior: ", lim_inf, "\n")  
cat("Limite superior: ", lim_sup, "\n")  
cat("Média final: ", media_final, "\n")
```

- Exibe os principais parâmetros estatísticos no console para inspeção.

- **Etapa 8: visualizando os dados em um gráfico de colunas**

R

```
ggplot(investimento, aes(x = reorder(Cidade, Valor), y = Valor)) +  
  geom_col(fill = ifelse(investimento$Valor >= lim_inf  
    & investimento$Valor <= lim_sup, "steelblue", "tomato")) +  
  geom_hline(yintercept = media_final, linetype = "dashed", color = "black") +  
  labs(title = "Investimento por Cidade", y = "Valor",  
    x = "Cidade") +  
  theme_minimal()
```

- geom_col(): gráfico de barras verticais por cidade.
- ifelse: cores diferentes:
 - Azul → dentro dos limites.
 - Alaranjado → valor fora dos limites.
- geom_hline(): linha horizontal pontilhada indicando a **média final**.
- labs() e theme_minimal(): adicionam título e estilo limpo ao gráfico.

- **Saída no Console do RStudio**

 Shell

```

> # carregando os pacotes utilizados
> library(dplyr)
> library(ggplot2)
> library(tibble)
>
> # carregando os dados
> investimento <- tibble::tibble(
+   Cidade = LETTERS[1:10],
+   Valor = c(20, 16, 14, 8, 19, 15, 14, 16, 19, 18)
+ )
>
> # calculando a média inicial
> media_inicial <- mean(investimento$Valor)
>
> # calculando o desvio-padrão
> desvio <- sd(investimento$Valor)
>
> # calculando o limite inferior
> lim_inf <- media_inicial - 2 * desvio
>
> # calculando o limite superior
> lim_sup <- media_inicial + 2 * desvio
>
> # filtrando os dados
> filtrado <- investimento %>%
+   filter(Valor >= lim_inf & Valor <= lim_sup)
>
> filtrado
# A tibble: 9 × 2
  Cidade Valor
  <chr>   <dbl>
1 A         20
2 B         16
3 C         14
4 E         19
5 F         15
6 G         14
7 H         16
8 I         19
9 J         18
>
> # calculando a média final
> media_final <- mean(filtrado$Valor)
>
> # imprimindo os resultados dos objetos criados
> cat("Média inicial das amostras: ", media_inicial, "\n")
Média inicial das amostras: 15.9
> cat("Desvio-padrão das amostras: ", desvio, "\n")
Desvio-padrão das amostras: 3.510302
> cat("Limite inferior: ", lim_inf, "\n")
Limite inferior: 8.879395

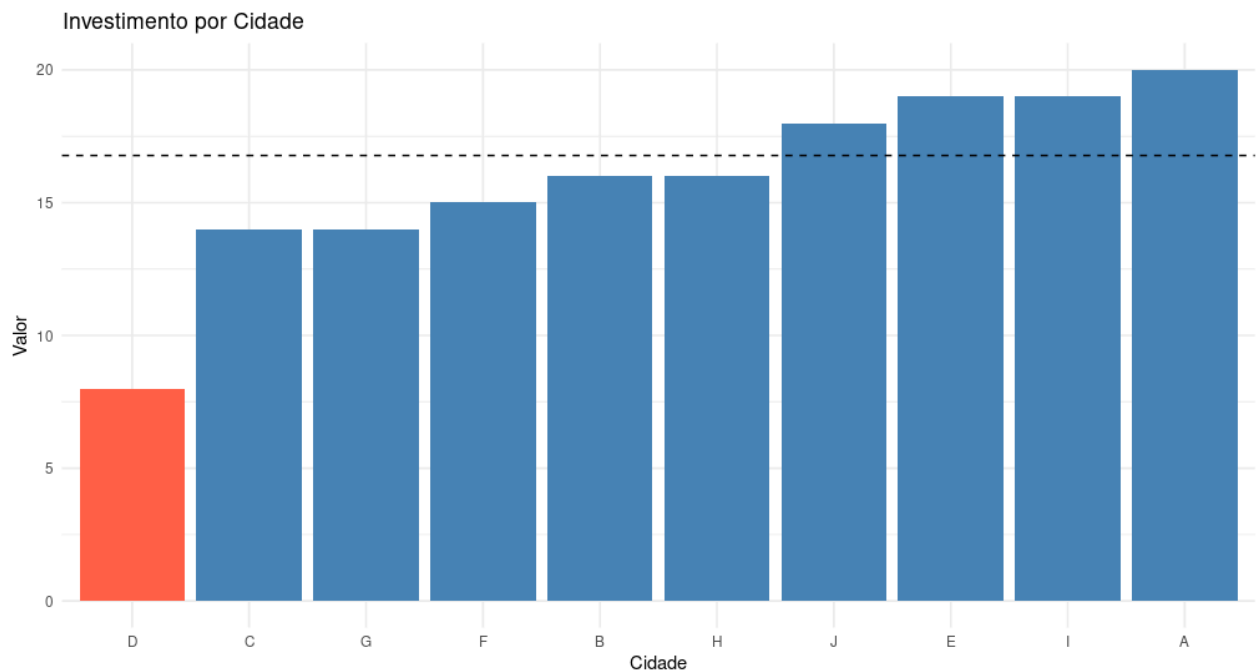
```

```

> cat("Limite superior: ", lim_sup, "\n")
Limite superior: 22.9206
> cat("Média final: ", media_final, "\n")
Média final: 16.77778
>
> # visualizando os dados
> ggplot(investimento, aes(x = reorder(Cidade, Valor), y = Valor)) +
+   geom_col(fill = ifelse(investimento$Valor >= lim_inf
+   & investimento$Valor <= lim_sup, "steelblue", "tomato")) +
+   geom_hline(yintercept = media_final, linetype = "dashed", color = "black") +
+   labs(title = "Investimento por Cidade", y = "Valor",
+   x = "Cidade") +
+   theme_minimal()
>

```

- **Saída na aba Plots do RStudio**



Qual o investimento básico que você daria como resposta?

O investimento básico a ser considerado como resposta é 16,78 unidades. Esse valor representa a média final dos investimentos após excluir o valor considerado atípico (8) referente a cidade D, conforme exibido e destacado na cor laranja da primeira coluna. Essa exclusão e recálculo ajudam a suavizar o impacto de valores extremos e refletem melhor o investimento típico entre as cidades.

Questão 4: Um órgão de planejamento governamental está avaliando a distribuição da renda per capita entre os municípios de um determinado estado, com o objetivo de direcionar políticas públicas de combate à desigualdade. A equipe de análise produziu o seguinte resumo estatístico com base nos dados coletados:

- Média: R\$ 850,00
- Mediana: R\$ 700,00
- Moda: R\$ 600,00
- Coeficiente de curtose: 5,2
- Script

 R

```
# carregando os pacotes
library(ggplot2)
library(tibble)

# simulando os dados baseados nas medidas fornecidas
set.seed(123)
renda <- c(rnorm(95, mean = 700, sd = 100), runif(5, 1200, 3000))

# inserindo as estatísticas fornecidas
media <- 850
mediana <- 700
moda <- 600
curtose <- 5.2

# criando o dataframe
df <- tibble::tibble(Renda = renda)

# visualizando os dados simulados
ggplot(df, aes(x = Renda)) +
  geom_histogram(bins = 30, fill = "skyblue", color = "white") +

  # Linhas verticais para medidas
  geom_vline(xintercept = media, color = "red",
             linetype = "dashed", size = 1.2) +
  geom_vline(xintercept = mediana, color = "blue",
             linetype = "dotted", size = 1.2) +
  geom_vline(xintercept = moda, color = "darkgreen",
             linetype = "dotdash", size = 1.2) +

  # Legenda manual com annotate
  annotate("text", x = media + 20, y = Inf, label = "Média",
          color = "red", hjust = 0, vjust = 2, size = 4) +
  annotate("text", x = mediana + 20, y = Inf, label = "Mediana",
          color = "blue", hjust = 0, vjust = 3.5, size = 4) +
  annotate("text", x = moda + 20, y = Inf, label = "Moda",
          color = "darkgreen", hjust = 0, vjust = 5, size = 4) +

  labs(
    title = "Distribuição Simulada da Renda per Capita",
    x = "Renda (R$)", y = "Frequência"
  ) +
  theme_minimal()
```

- Explicação do Script
- Etapa 1: Carregamento dos pacotes

 R

```
library(ggplot2)
library(tibble)
```

- ggplot2: usado para visualização de dados (gráfico de histograma com anotações).
- tibble: usado para criar um data frame moderno, que é mais legível e eficiente que o data.frame base.

- **Etapa 2: simulando com os dados informados**

 R

```
set.seed(123)
renda <- c(rnorm(95, mean = 700, sd = 100), runif(5, 1200, 3000))
```

- set.seed(123): garante que a simulação será reproduzível (mesmos números aleatórios sempre).
- rnorm(95, mean = 700, sd = 100): gera 95 valores de renda a partir de uma distribuição normal, centrada em R\$ 700.
- runif(5, 1200, 3000): adiciona 5 valores aleatórios altos, simulando outliers — municípios muito ricos.
- O vetor nomeado de renda simula uma distribuição assimétrica à direita, como descrito no enunciado da questão.

- **Etapa 3: inserido as estatísticas fornecidas**

 R

```
media <- 850
mediana <- 700
moda <- 600
curtose <- 5.2
```

Esses valores são fornecidos pela equipe analítica do órgão de planejamento e são usados para comparação visual no gráfico. Eles representam:

- Média: R\$850
- Mediana: R\$700
- Moda: R\$600
- Curtose: apenas informativa; não é visualizada diretamente no gráfico.

- **Etapa 4: criando um dataframe**

 R

```
df <- tibble::tibble(Renda = renda)
```

- Converte o vetor de nome renda em uma coluna Renda dentro de um tibble, preparando os dados para o ggplot.

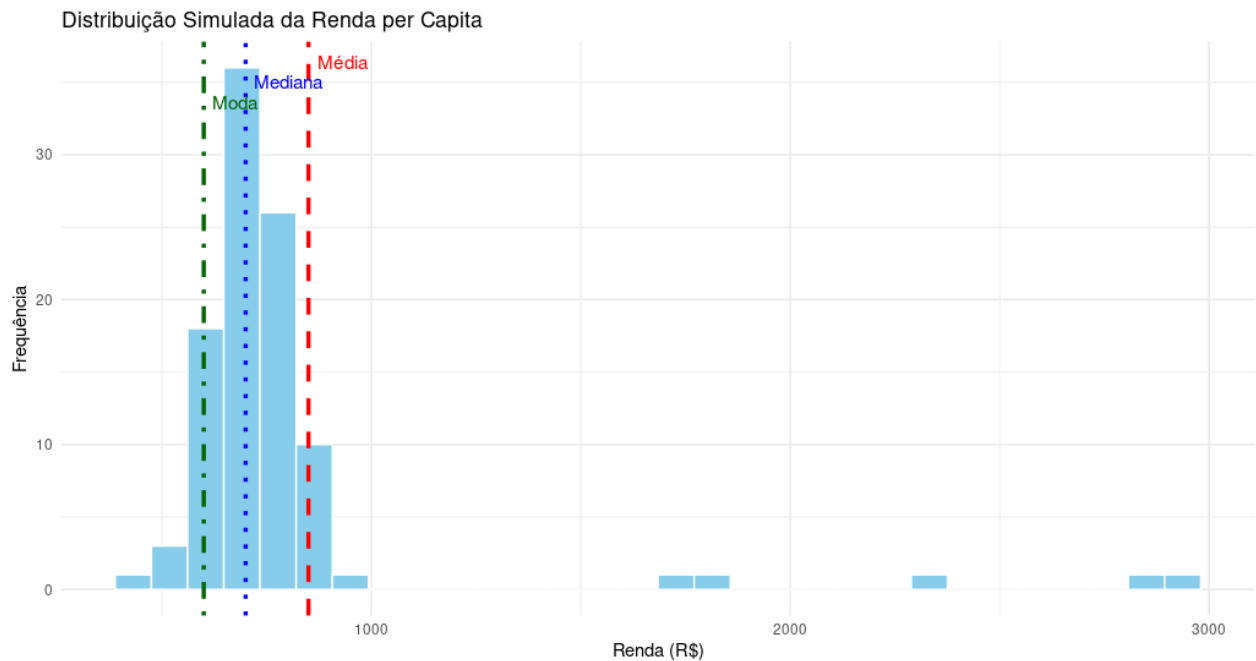
- **Etapa 5: construindo o gráfico com os dados simulados**

R

```
ggplot(df, aes(x = Renda)) +  
  geom_histogram(bins = 30, fill = "skyblue", color = "white") +  
  
  # Linhas verticais para medidas  
  geom_vline(xintercept = media, color = "red",  
             linetype = "dashed", size = 1.2) +  
  geom_vline(xintercept = mediana, color = "blue",  
             linetype = "dotted", size = 1.2) +  
  geom_vline(xintercept = moda, color = "darkgreen",  
             linetype = "dotdash", size = 1.2) +  
  
  # Legenda manual com annotate  
  annotate("text", x = media + 20, y = Inf, label = "Média",  
           color = "red", hjust = 0, vjust = 2, size = 4) +  
  annotate("text", x = mediana + 20, y = Inf, label = "Mediana",  
           color = "blue", hjust = 0, vjust = 3.5, size = 4) +  
  annotate("text", x = moda + 20, y = Inf, label = "Moda",  
           color = "darkgreen", hjust = 0, vjust = 5, size = 4) +  
  
  labs(  
    title = "Distribuição Simulada da Renda per Capita",  
    x = "Renda (R$)", y = "Frequência"  
  ) +  
  theme_minimal()
```

- Inicia o gráfico usando a variável Renda como eixo x (renda per capita). O ggplot() serve de base para os elementos adicionados a seguir.
- Cria um histograma da variável Renda com 30 classes (bins). Cor azul para preenchimento e contorno branco das barras.
- geom_vline(): desenha linhas verticais no histograma para destacar:
 - Média: linha vermelha tracejada.
 - Mediana: linha azul pontilhada.
 - Moda: linha verde com traço-ponto.
- Isso permite comparar visualmente a posição de cada medida e observar a assimetria da distribuição.
- annotate("text", ...): adiciona texto flutuante no gráfico.
- x = media + 20: posiciona o texto levemente à direita da linha correspondente.
- y = Inf: posiciona o texto no topo do gráfico.
- hjust e vjust: controlam o alinhamento do texto.
- As anotações exibem os rótulos "Média", "Mediana" e "Moda" no canto superior direito do gráfico, de forma clara e colorida.
- labs(): define o título do gráfico e os rótulos dos eixos.
- theme_minimal(): aplica um estilo limpo e moderno, sem grades pesadas, ideal para apresentações e relatórios técnicos.

- **Saída na aba Plots do RStudio**



Com base nessas informações, responda às perguntas abaixo:

a) Interprete a assimetria da distribuição da renda per capita. Que tipo de política pública pode ser sugerida com base nessa informação?

Ao observar a distribuição da renda per capita, percebe-se que a média é superior à mediana, que, por sua vez, é maior que a moda.

Essa configuração revela uma distribuição assimétrica à direita, também conhecida como assimetria positiva.

Nesse tipo de distribuição, a maior parte dos dados está concentrada em valores mais baixos, enquanto poucos municípios apresentam rendas extremamente elevadas, distorcendo a média para cima.

Essa assimetria tem implicações significativas na análise da desigualdade regional. A média deixa de representar o padrão de vida da maioria da população, pois está inflacionada por poucos municípios com alta renda per capita.

Com base nessa interpretação, é possível sugerir políticas públicas redistributivas, voltadas especialmente para os municípios com rendas abaixo da mediana.

Programas de transferência de renda, incentivos fiscais e investimentos sociais em saúde, educação e infraestrutura poderiam ser direcionados de forma mais equitativa, buscando reduzir a desigualdade estrutural entre regiões.

b) O valor do coeficiente de curtose indica uma distribuição mais ou menos concentrada em torno da média? O que isso pode representar em termos de variação de renda entre os municípios?

O coeficiente de curtose informado é de 5,2, valor consideravelmente superior ao coeficiente da distribuição normal (que é 3). Isso indica que a distribuição é leptocúrtica, isto é, apresenta um pico mais acentuado ao redor da mediana e caudas mais longas.

Na prática, isso significa que há uma forte concentração de municípios com rendas próximas à mediana, mas também há a presença significativa de valores extremos, principalmente no topo da distribuição.

A alta curtose sugere que, embora muitos municípios estejam em uma faixa de renda semelhante, há uma variação substancial nos extremos, com alguns municípios muito mais ricos.

Essa configuração pode indicar desigualdade regional intensa, onde poucos municípios concentram grande parte da riqueza, enquanto a maioria permanece em condições econômicas mais modestas.

c) Explique, em termos práticos, por que conhecer medidas de formato (assimetria e curtose) é importante para o desenvolvimento de políticas públicas mais eficazes.

Compreender medidas de forma como assimetria e curtose é essencial para o desenvolvimento de políticas públicas eficazes.

A assimetria permite identificar se a média é uma boa representação da realidade.

Em distribuições assimétricas, como a analisada, o uso da média como referência pode mascarar desigualdades e induzir a políticas inadequadas.

Já a curtose informa sobre a concentração e dispersão dos dados, permitindo identificar se existem muitos valores extremos que demandam atenção especial.

Essas medidas ajudam a mapear com precisão os perfis socioeconômicos dos municípios, orientando decisões mais justas, equitativas e baseadas em evidências.

Elas permitem que o gestor público não apenas identifique desigualdades, mas também compreenda sua natureza e extensão, promovendo a alocação mais eficiente de recursos e o planejamento de ações que promovam maior justiça social.