# The Weighted Mean and the Median

No exercício passado aprendemos sobre a média aritmética, e seu comportamento dependendo do tamanho/representatividade da amostra em relação à população. Aqui vamos expandir os horizontes da Estatítica e abordar a média ponderada e mediana.

## Base Casas Vendidas em Ames entre 2006 e 2010

Vamos usar essa base com 2930 linhas com 82 colunas contendo informações de características de casas vendidas entre 2006 e 2010 na cidade Ames (estado de Iowa nos EUA).

Esse foi um trabalho feito pelo professor Dean DeCock, publicado neste artigo e os detalhes sobre as informações presentes na base estão neste link

O separador da base são tabs, é um arquivo do tipo TSV (tab-separated value), são basicamente espaços. Poderíamos usar a função read.csv e informar o parâmetro sep= “\t” que funcionaria da mesma forma.

```
base <- read_tsv("https://s3.amazonaws.com/dq-content/444/AmesHousing.txt")
```

```
## Rows: 2930 Columns: 82
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (45): PID, MS SubClass, MS Zoning, Street, Alley, Lot Shape, Land Contou...
## dbl (37): Order, Lot Frontage, Lot Area, Overall Qual, Overall Cond, Year Bu...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
glimpse(base)
```

```
## Rows: 2,930
## Columns: 82
## $ Order             <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 1~
## $ PID               <chr> "0526301100", "0526350040", "0526351010", "052635303~
## $ `MS SubClass`     <chr> "020", "020", "020", "020", "060", "060", "120", "12~
## $ `MS Zoning`       <chr> "RL", "RH", "RL", "RL", "RL", "RL", "RL", "RL", "RL"~
## $ `Lot Frontage`    <dbl> 141, 80, 81, 93, 74, 78, 41, 43, 39, 60, 75, NA, 63,~
## $ `Lot Area`        <dbl> 31770, 11622, 14267, 11160, 13830, 9978, 4920, 5005,~
## $ Street            <chr> "Pave", "Pave", "Pave", "Pave", "Pave", "Pave", "Pav~
## $ Alley             <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ `Lot Shape`       <chr> "IR1", "Reg", "IR1", "Reg", "IR1", "IR1", "Reg", "IR~
## $ `Land Contour`    <chr> "Lvl", "Lvl", "Lvl", "Lvl", "Lvl", "Lvl", "Lvl", "HL~
## $ Utilities         <chr> "AllPub", "AllPub", "AllPub", "AllPub", "AllPub", "A~
## $ `Lot Config`      <chr> "Corner", "Inside", "Corner", "Corner", "Inside", "I~
## $ `Land Slope`      <chr> "Gtl", "Gtl", "Gtl", "Gtl", "Gtl", "Gtl", "Gtl", "Gt~
## $ Neighborhood      <chr> "NAmes", "NAmes", "NAmes", "NAmes", "Gilbert", "Gilb~
## $ `Condition 1`     <chr> "Norm", "Feedr", "Norm", "Norm", "Norm", "Norm", "No~
## $ `Condition 2`     <chr> "Norm", "Norm", "Norm", "Norm", "Norm", "Norm", "Nor~
## $ `Bldg Type`       <chr> "1Fam", "1Fam", "1Fam", "1Fam", "1Fam", "1Fam", "Twn~
## $ `House Style`     <chr> "1Story", "1Story", "1Story", "1Story", "2Story", "2~
## $ `Overall Qual`    <dbl> 6, 5, 6, 7, 5, 6, 8, 8, 8, 7, 6, 6, 6, 7, 8, 8, 8, 9~
## $ `Overall Cond`    <dbl> 5, 6, 6, 5, 5, 6, 5, 5, 5, 5, 5, 7, 5, 5, 5, 5, 7, 2~
## $ `Year Built`      <dbl> 1960, 1961, 1958, 1968, 1997, 1998, 2001, 1992, 1995~
## $ `Year Remod/Add`  <dbl> 1960, 1961, 1958, 1968, 1998, 1998, 2001, 1992, 1996~
## $ `Roof Style`      <chr> "Hip", "Gable", "Hip", "Hip", "Gable", "Gable", "Gab~
## $ `Roof Matl`       <chr> "CompShg", "CompShg", "CompShg", "CompShg", "CompShg~
## $ `Exterior 1st`    <chr> "BrkFace", "VinylSd", "Wd Sdng", "BrkFace", "VinylSd~
## $ `Exterior 2nd`    <chr> "Plywood", "VinylSd", "Wd Sdng", "BrkFace", "VinylSd~
## $ `Mas Vnr Type`    <chr> "Stone", "None", "BrkFace", "None", "None", "BrkFace~
## $ `Mas Vnr Area`    <dbl> 112, 0, 108, 0, 0, 20, 0, 0, 0, 0, 0, 0, 0, 0, 0, 60~
## $ `Exter Qual`      <chr> "TA", "TA", "TA", "Gd", "TA", "TA", "Gd", "Gd", "Gd"~
## $ `Exter Cond`      <chr> "TA", "TA", "TA", "TA", "TA", "TA", "TA", "TA", "TA"~
## $ Foundation        <chr> "CBlock", "CBlock", "CBlock", "CBlock", "PConc", "PC~
## $ `Bsmt Qual`       <chr> "TA", "TA", "TA", "TA", "Gd", "TA", "Gd", "Gd", "Gd"~
## $ `Bsmt Cond`       <chr> "Gd", "TA", "TA", "TA", "TA", "TA", "TA", "TA", "TA"~
## $ `Bsmt Exposure`   <chr> "Gd", "No", "No", "No", "No", "No", "Mn", "No", "No"~
## $ `BsmtFin Type 1`  <chr> "BLQ", "Rec", "ALQ", "ALQ", "GLQ", "GLQ", "GLQ", "AL~
## $ `BsmtFin SF 1`    <dbl> 639, 468, 923, 1065, 791, 602, 616, 263, 1180, 0, 0,~
## $ `BsmtFin Type 2`  <chr> "Unf", "LwQ", "Unf", "Unf", "Unf", "Unf", "Unf", "Un~
## $ `BsmtFin SF 2`    <dbl> 0, 144, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1120, 0,~
## $ `Bsmt Unf SF`     <dbl> 441, 270, 406, 1045, 137, 324, 722, 1017, 415, 994, ~
## $ `Total Bsmt SF`   <dbl> 1080, 882, 1329, 2110, 928, 926, 1338, 1280, 1595, 9~
## $ Heating           <chr> "GasA", "GasA", "GasA", "GasA", "GasA", "GasA", "Gas~
## $ `Heating QC`      <chr> "Fa", "TA", "TA", "Ex", "Gd", "Ex", "Ex", "Ex", "Ex"~
```

```
## $ `Central Air`     <chr> "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y~
## $ Electrical        <chr> "SBrkr", "SBrkr", "SBrkr", "SBrkr", "SBrkr", "SBrkr"~
## $ `1st Flr SF`      <dbl> 1656, 896, 1329, 2110, 928, 926, 1338, 1280, 1616, 1~
## $ `2nd Flr SF`      <dbl> 0, 0, 0, 0, 701, 678, 0, 0, 0, 776, 892, 0, 676, 0, ~
## $ `Low Qual Fin SF` <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ `Gr Liv Area`     <dbl> 1656, 896, 1329, 2110, 1629, 1604, 1338, 1280, 1616,~
## $ `Bsmt Full Bath`  <dbl> 1, 0, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1~
## $ `Bsmt Half Bath`  <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ `Full Bath`       <dbl> 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 3, 2, 1~
## $ `Half Bath`       <dbl> 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1~
## $ `Bedroom AbvGr`   <dbl> 3, 2, 3, 3, 3, 3, 2, 2, 2, 3, 3, 3, 3, 2, 1, 4, 4, 1~
## $ `Kitchen AbvGr`   <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1~
## $ `Kitchen Qual`    <chr> "TA", "TA", "Gd", "Ex", "TA", "Gd", "Gd", "Gd", "Gd"~
## $ `TotRms AbvGrd`   <dbl> 7, 5, 6, 8, 6, 7, 6, 5, 5, 7, 7, 6, 7, 5, 4, 12, 8, ~
## $ Functional        <chr> "Typ", "Typ", "Typ", "Typ", "Typ", "Typ", "Typ", "Ty~
## $ Fireplaces        <dbl> 2, 0, 0, 2, 1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1~
## $ `Fireplace Qu`    <chr> "Gd", NA, NA, "TA", "TA", "Gd", NA, NA, "TA", "TA", ~
## $ `Garage Type`     <chr> "Attchd", "Attchd", "Attchd", "Attchd", "Attchd", "A~
## $ `Garage Yr Blt`   <dbl> 1960, 1961, 1958, 1968, 1997, 1998, 2001, 1992, 1995~
## $ `Garage Finish`   <chr> "Fin", "Unf", "Unf", "Fin", "Fin", "Fin", "Fin", "RF~
## $ `Garage Cars`     <dbl> 2, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 2, 3~
## $ `Garage Area`     <dbl> 528, 730, 312, 522, 482, 470, 582, 506, 608, 442, 44~
## $ `Garage Qual`     <chr> "TA", "TA", "TA", "TA", "TA", "TA", "TA", "TA", "TA"~
## $ `Garage Cond`     <chr> "TA", "TA", "TA", "TA", "TA", "TA", "TA", "TA", "TA"~
## $ `Paved Drive`     <chr> "P", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y", "Y~
## $ `Wood Deck SF`    <dbl> 210, 140, 393, 0, 212, 360, 0, 0, 237, 140, 157, 483~
## $ `Open Porch SF`   <dbl> 62, 0, 36, 0, 34, 36, 0, 82, 152, 60, 84, 21, 75, 0,~
## $ `Enclosed Porch`  <dbl> 0, 0, 0, 0, 0, 0, 170, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,~
## $ `3Ssn Porch`      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ `Screen Porch`    <dbl> 0, 120, 0, 0, 0, 0, 0, 144, 0, 0, 0, 0, 0, 0, 140, 2~
## $ `Pool Area`       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ `Pool QC`         <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, ~
## $ Fence             <chr> NA, "MnPrv", NA, NA, "MnPrv", NA, NA, NA, NA, NA, NA~
## $ `Misc Feature`    <chr> NA, NA, "Gar2", NA, NA, NA, NA, NA, NA, NA, NA, "She~
## $ `Misc Val`        <dbl> 0, 0, 12500, 0, 0, 0, 0, 0, 0, 0, 0, 500, 0, 0, 0, 0~
## $ `Mo Sold`         <dbl> 5, 6, 6, 4, 3, 6, 4, 1, 3, 6, 4, 3, 5, 2, 6, 6, 6, 6~
## $ `Yr Sold`         <dbl> 2010, 2010, 2010, 2010, 2010, 2010, 2010, 2010, 2010~
## $ `Sale Type`       <chr> "WD", "WD", "WD", "WD", "WD", "WD", "WD", "WD", "WD"~
## $ `Sale Condition`  <chr> "Normal", "Normal", "Normal", "Normal", "Normal", "N~
## $ SalePrice         <dbl> 215000, 105000, 172000, 244000, 189900, 195500, 2135~
```

Agora vamos dizer que estamos interessados em entender a distribuição do preço de venda das casas, coluna SalePrice. Com a função summary já conseguimos ter uma boa noção do comportamento.

```r
base$SalePrice %>% summary(digits = 8)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   12789  129500  160000  180796  213500  755000
```

Vamos imaginar que ao invés do dataset acima, nós temos dados summarizados por ano

```r
base_consolidada <- base %>%
  group_by(`Yr Sold`) %>%
  summarise(MediaPreco = mean(SalePrice),
            CasasVendidas = n()) %>%
  arrange(CasasVendidas)

base_consolidada
```

```
## # A tibble: 5 x 3
##   `Yr Sold` MediaPreco CasasVendidas
##       <dbl>      <dbl>         <int>
## 1      2010    172598.           341
## 2      2008    178842.           622
## 3      2006    181762.           625
## 4      2009    181405.           648
## 5      2007    185138.           694
```

Nosso objetivo é calcular a média, vamos calcular a média do preço médio calculado na base consolidada e comparar se chegamos no mesmo resultado da coluna SalePrice original.

```r
mean(base_consolidada$MediaPreco)
```

```
## [1] 179948.8
```

A média de fato ficou diferente, e isso se dá pois a média foi calculada separadamente para cada ano, e a quantidade de casas em cada ano é diferente colocando um peso diferente na média. No entanto o cálculo fica errado, pois estamos somando os 5 preços médios e dividindo por 5, em outras palavras estamos nesse cálculo dando peso igual para os 5 preços, sendo que cada ano tem quantidades diferentes de casas vendidas.

Uma forma de corrigir isso, imaginando que só tivéssemos a base consolidada para trabalhar, é multiplicar os preços médios pela quantidade de casas, somar os preços e dividir pela quantidade de casas.

```r
base_consolidada <- base_consolidada %>%
  mutate(resultado = MediaPreco * CasasVendidas)

media_consolidada <- sum(base_consolidada$resultado) /
                     sum(base_consolidada$CasasVendidas)
media_consolidada
```

```
## [1] 180796.1
```

```r
media <- mean(base$SalePrice)
media
```

```
## [1] 180796.1
```

Agora sim chegamos na mesma média que calculamos inicialmente, pois levamos em consideração o peso de cada valor.

## Média Ponderada (com pesos)

Em situações vamos usar ou não a média ponderada?

Quando temos uma base sem agregação, sem agrupamento, onde cada linha representa um caso único, seja ele único na base ou até se repetindo em outras linhas, podemos usar a média aritmética, pois o peso de cada linha é um.

Agora quando na base cada linha representa um grupo, tem um campo com quantidade ou probabilidade em que aquele evento acontece, nesse caso precisamos da média ponderada para levar o peso correto em consideração no cálculo.

Vamos representar isso através de uma fórmula.

Basicamente o que fizemos é pegar o preço *x* e multiplicar pelo peso *w*, e repetimos isso para cada valor da base, então *x1* multiplicado por *w1*, *x2* por *w2*, *x3w3* e assim por diante. O importante é ter a mesma quantidade de *x* e *w*.

$$weighted\,mean = \frac{x_1w_1 + x_2w_2 + ...x_nw_n}{w_1 + w_2 + w_n}$$

De forma simplificada:

$$weighted\,mean = \frac{\sum_{i=1}^{n} x_iw_i}{\sum_{i=1}^{n} w_i}$$

## Namespaces

Fazendo um rápido parênteses antes de seguir no assunto da Média, você sabe o que são os namespaces?

É basicamente uma forma que o R permite de identificar melhor as funções pelo nome. É comum existir mais de uma função em pacotes diferentes, porém com o mesmo nome. Sendo assim usando namespace conseguimos definir com mais precisão de qual pacote queremos usar. Também pode ser o caso se usarmos apenas uma função de um pacote sem a necessidade de carregar na memória do R o pacote inteiro, e nesse caso o namespace ajuda carregando apenas a função em específico na memória.

Como funciona o namespace? Basta seguir o seguinte padrão:

```
nome_do_pacote::nome_da_funcao(parametros)
```

Por exemplo o código abaixo:

```
stringr::str_c("Hello", "World")
```

```
## [1] "HelloWorld"
```

Também daria para carregar o pacote por completo

```
library(stringr)

str_c("Hello", "World")
```

```
## [1] "HelloWorld"
```

E mesmo carregando a biblioteca inteira, pode ser interessante usar o namespace para evitar ambiguidade no código.

## Mediana

Nós vimos até aqui situações onde é possível calcular a média, apesar de o cálculo sair errado (e a média ponderada foi a solução). Mas e quando não é nem possível realizar um cálculo?

Falando ainda de casas, vamos imaginar que na base existisse um campo com a quantidade de cômodos nessa situação:

```r
comodos <- c(5,1,2,"10 ou mais", 4, 5,2)
```

Não é possível calcular a média pois o dado “10 ou mais” é um texto e representa um valor que pode variar, podendo ser o próprio 10, ou 11, 12 e assim por diante. Chamamos isso de distribuição aberta, pois o limite não é bem definido.

Nesse caso poderíamos considerar a mediana. Basicamente precisamos ordenar os valores e considerar o valor do meio.

```r
comodos <- c(1,2,2,4,5,5,"10 ou mais")
```

O resultado seria 4 para esse caso. e a quantidade de posições abaixo e acima desse valor são iguais, 3 para cada lado.

E quando a amostra tem uma quantidade par de valores, o caso acima tinha 7 (ímpar) e foi fácil encontrar o meio.

Se a amostra acima tivesse 6 posições, por exemplo, pegamos as 2 posições do meio e calculamos a média para saber o resultado.

```r
comodos <- c(1,2,2,4,5,"10 ou mais")
#meio possui 2 e 4, então soma e divide pelas 2 posições
(2+4)/2
```

```
## [1] 3
```

Nesse segundo cenário o resultado é 3. É comum em inglês utilizar o termo *average*, e muitas vezes se confunde com o termo *mean*, mas *average* diz respeito a o valor estatístico mais representativo da amostra, que pode ser média como pode ser mediana ou moda dependendo do contexto.

Para a média vimos símbolos utilizados para representar essas métricas, já para mediana, costumam usar o próprio nome, não existe um consenso de um símbolo.

Agora vamos ilustrar como ficaria o exemplo acima via código.

```r
comodos <- c(5,1,2,"10 ou mais", 4,2)

comodos_ajustado <- stringr::str_replace(comodos, "10 ou mais", "10") %>%
  as.numeric() %>% sort()
comodos_ajustado
```

```
## [1]  1  2  2  4  5 10
```

```r
indice_meio <- c(length(comodos_ajustado)/2,
                 (length(comodos_ajustado)/2)+1)

mediana <- mean(comodos_ajustado[indice_meio])
mediana
```

```
## [1] 3
```

Agora comparando a mediana com a média, a média considera um peso igualmente distribuído em todos valores da amostra. Enquanto que a mediana olha apenas os valores do meio. Isso significa que a média resiste muito mais à mudanças de valores comparado à média. Por exemplo se o maior valor da amostra acima fosse 1000 ao invés de 10, isso mudaria significativamente a média, enquanto que a mediana permaneceria a mesma.

Tendo isso em mente, é uma medida estatística muito boa para ser utilizada em bases onde temos conhecimento da existência de outliers que distorceria o cálculo da média. Um bom exemplo é o cálculo da média da renda de uma população. Sabemos que a maior parte dos brasileiros recebem poucos salários mínimos, mas temos uma minúscula parcela de super ricos com renda na casa de bilhões, que distorceriam completamente o resultado da média.

Para simplificar veja o exemplo abaixo. A mediana ficou em 5k, enquanto que a média foi muito influenciada pelo valor de 100k ficando em 23.020 reais.

```r
df <- tibble::tibble(empregado = purrr::map_chr(1:5,
                                 function(x) stringr::str_c("empregado ", x)),
                     salario = c(1300,1300,5000,7500,100000))

mean(df$salario)
```

```
## [1] 23020
```

```r
median(df$salario)
```

```
## [1] 5000
```

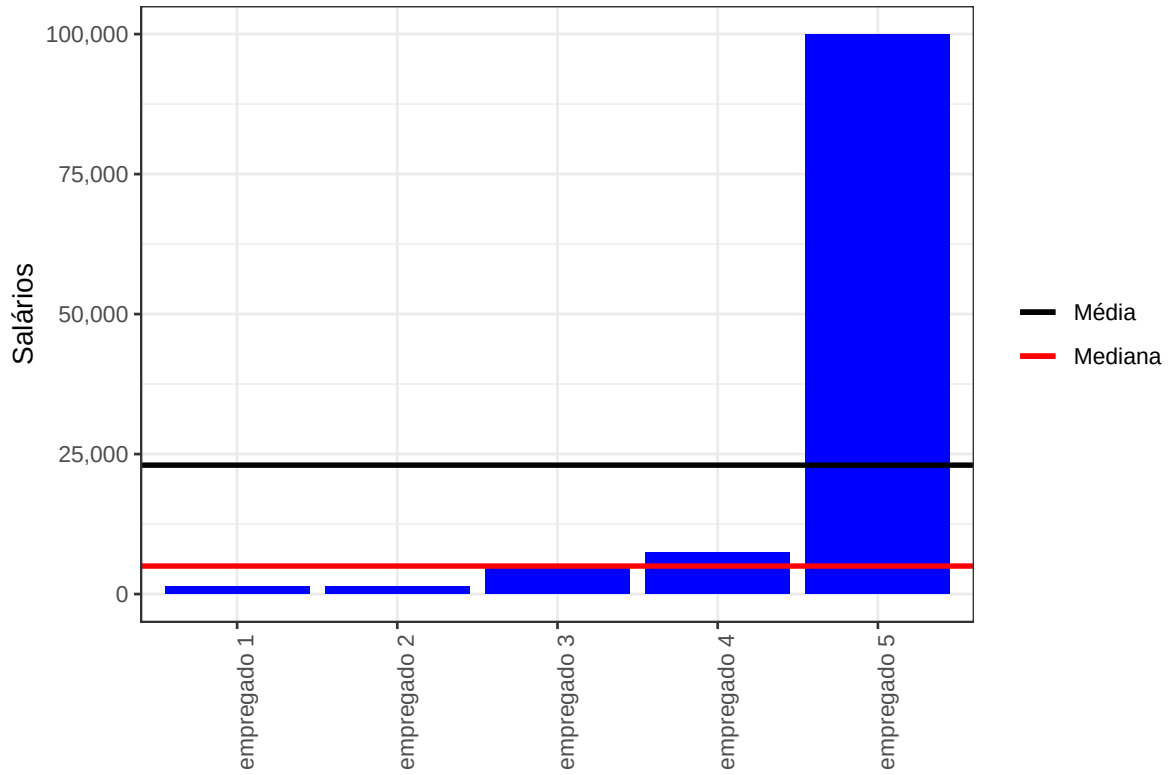
```r
library(ggplot2)

df %>%
  ggplot(aes(x = empregado, y = salario)) +
    geom_bar(stat = "identity",
```

```
            fill = "blue") +
  geom_hline(aes(yintercept = mean(df$salario),
                 color = "black"),
             size = 1) +
  geom_hline(aes(yintercept = median(df$salario),
                  color = "red"),
             size = 1) +
  scale_y_continuous(labels = scales::comma) +
  scale_colour_manual(values = c("black", "red"),
                      name = "",
                      labels = c("Média", "Mediana")) +
  theme_bw() + theme(axis.text.x = element_text(angle = 90)) +
  xlab("") +
  ylab("Salários")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: Use of `df$salario` is discouraged.
## i Use `salario` instead.
## Use of `df$salario` is discouraged.
## i Use `salario` instead.
```




## Identificando outliers com boxplot

Vamos agora utilizar o mesmo exemplo do gráfico de barras para construir um boxplot e observar o comportamento

```
df <- tibble::tibble(empregado = purrr::map_chr(1:5,
                                    function(x) stringr::str_c("empregado ", x)),
                     salario = c(1300,1300,5000,7500,100000))

mean(df$salario)
```

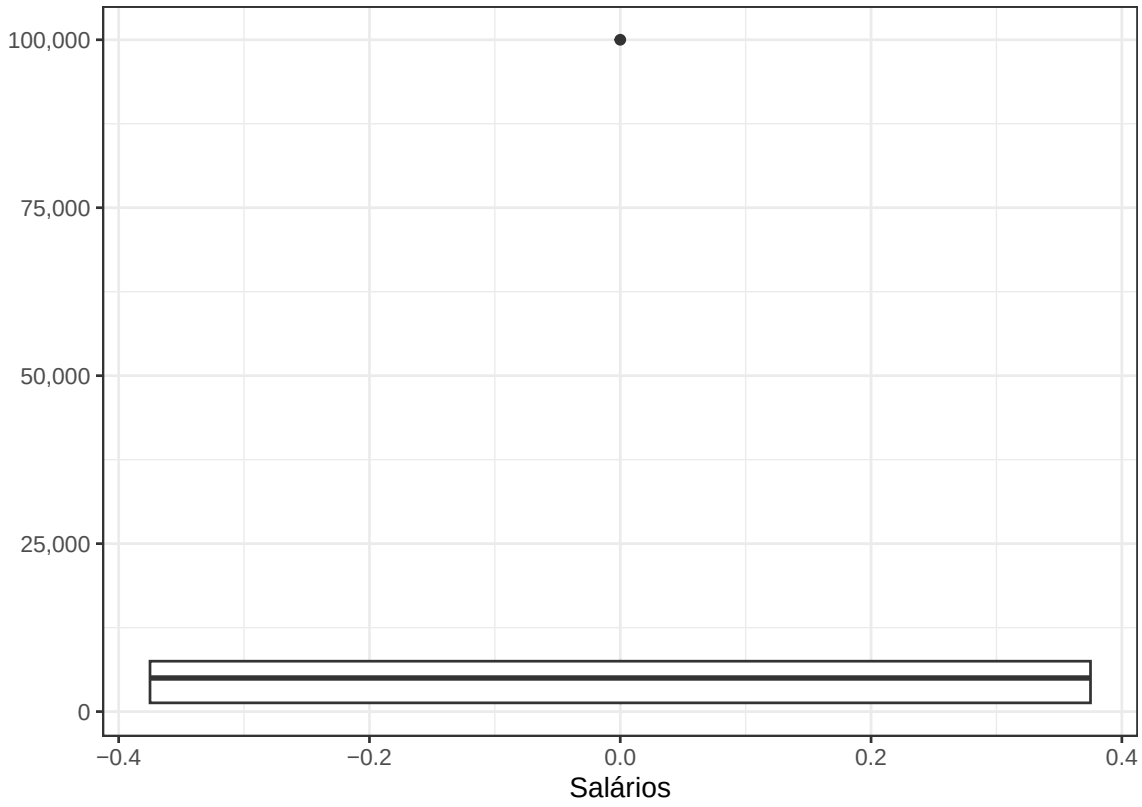
```
## [1] 23020
```

```
median(df$salario)
```

```
## [1] 5000
```

```r
library(ggplot2)

df %>%
  ggplot(aes(y = salario)) +
    geom_boxplot()+
    scale_y_continuous(labels = scales::comma) +
    theme_bw() +
    ylab("") +
    xlab("Salários")
```




Até aqui mostramos o quanto a média pode não ser o melhor valor mais representativo da amostra. Mas em qual caso ela é? Vejamos o seguinte gráfico que traz uma "nota" das condições do imóvel de 1 a 9

A média é 5.56 enquanto que a mediana é 5. Mas pelo histograma vemos uma concentração muito maior para notas maiores que 5, então a média nesse caso representa mais as características dessa amostra.

Mas vale ressaltar que nesse exemplo estamos falando de uma informação ordinal, e a nota vai de 1 a 9 mas poderia ser de 1 a 100, e não temos o valor exato da distância de uma nota para outra, e a interpretação do resultado pode levar a problemas teóricos.

```r
base$`Overall Cond` %>% summary()
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   5.000   5.000   5.563   6.000   9.000
```

```r
base %>%
  ggplot(aes(x=`Overall Cond`))+
       geom_histogram(bins=10)
```