Measures of Variability

Nos últimos 3 exercícios focamos em calcular informações que nos ajudam a sumarizar a base em um único resultado, nos ajudando a entender a amostra/população, pois dependendo do volume de dados é humanamente impossível.

Durante esse estudo percebemos podem existir bases com valores muitos distintos resultando numa mesma média, ou seja existem casos com valores que giram em torno da média, ou que estão em extremos opostos muito discrepantes.

Então que tal a gente justamente buscar formas de avaliar como é o comportamento dessa distribuição das informações presentes na base?

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~
```

Range (Intervalo)

Comparando duas sequências de valores A=[4,4,4,4] e B=[0,8,0,8], A não varia e portanto o cálculo que representa essa variação deveria ser 0, enquanto que B varia, mas como calcular o quanto varia?

A média, moda e mediana não nos ajuda nesse caso, pois resulta em 4 para ambos casos, exceto a moda que é para o B não existe visto que 0 e 8 se repetem a mesma quantidade de vezes.

Sendo assim uma forma de medir a variedade é encontrando a diferença entre o valor mínimo e máximo.

$$max(A) - min(A) = 4 - 4 = 0$$

Então para a base A: 4 - 4 = 0, e para a base B: 8 - 0 = 8.

```
intervalo_preco <- base %>%
group_by(`Yr Sold`) %>%
summarize(intervalo_por_ano = max(SalePrice) - min(SalePrice))

intervalo_preco
```

```
## # A tibble: 5 x 2
##   `Yr Sold` intervalo_por_ano
##       <dbl>             <dbl>
## 1      2006            590000
## 2      2007            715700
## 3      2008            601900
## 4      2009            575100
## 5      2010            598868
```

O grande problema desse método é levar apenas em consideração o valor mínimo e máximo, ignorando os demais valores da base.

Nesse exemplo C=[1,1,1,1,1,1,1,1,21] o resultado é 20, mas não parece levar em consideração a quantidade de valores que não variam. Esse método é muito sensível à outliers.

Average Distance (Distância Média)

Uma forma de considerar todos os valores na hora de calcular a variabilidade de uma base e assim ter um resultado que represente melhor essa variação é com a distância média.

Aqui escolhemos um valor de referência, por exemplo a média da base e somamos a distância de todos valores presentes nessa base em relação à média, por último dividindo pelo número de ocorrências. Em outras palavras estamos calculando a distância média dos pontos em relação à media.

$$average\ distance = \frac{(x_1 - \mu) + (x_2 - \mu) + ...(x_N - \mu)}{N} = \frac{\sum_{i=1}^{n}(x_i - \mu)}{N}$$

```
C  <-  c(1,1,1,1,1,1,1,1,1,21)

distancia_media <- function(vetor){
    media <- mean(vetor)
    distancias <- vetor - media

    mean(distancias)
}

distancia_media(C)
```

```
## [1] 0
```

O resultado acima calculou que a média do vetor C é 3, então 1-3=-2, e como o número 1 se repete 9 vezes, acaba resultando em -18. Enquanto que o último número do vetor resulta em 21-2=18. Então os resultados se anulam resultando em zero, o que não parece representar muito bem o que aconteceu, afinal ocorreu algum grau de variação.

Mean Absolute Deviation (Desvio Absoluto Médio)

Pensando em resolver esse problema podemos pegar o valor absoluto usando o módulo, pois este retornará o valor positivo evitando que os valores se anulem.

$$mean\,absolute\,distance = \frac{|x_1 - \mu| + |x_2 - \mu| + ...|x_N - \mu|}{N} = \frac{\sum_{i=1}^{n}|x_i - \mu|}{N}$$

$$|-7| = +7$$
$$|+7| = +7$$

Esse cálculo se chama desvio absoluto médio (Mean Absolute Deviation), usamos a palavra desvio (deviation) quando nos referimos estatiticamente à distância da média.

Repetindo o cálculo para o Vetor chegamos num valor maior que zero (que nos faz entender que teve variação), no entanto muito menor que 20 comparado ao primeiro cálculo

```
C  <-  c(1,1,1,1,1,1,1,1,1,21)

desvio_absoluto_medio <- function(vetor) {
    media <- mean(vetor)
    distancias  <-  vetor - media
    abs_dev <- abs(distancias)
    mean(abs_dev)
}

desvio_absoluto_medio(C)
```

```
## [1] 3.6
```

Variance (Variância)

A Variância, também conhecida por Desvio Quadrático Médio (Mean Squared Deviation), traz uma segunda alternativa para resolver o problema com o cálculo da Distância Média. Aqui ao invés de somar o valor absoluto das distâncias, somamos o valor quadrado das distâncias, cálculo esse que por natureza já remove os sinais negativos assim evitando que os valores se anulem.

$$variance = \frac{(x_1 - \mu)^2 + (x_2 - \mu)^2 + ...(x_N - \mu)^2}{N} = \frac{\sum_{i=1}^{n}(x_i - \mu)^2}{N}$$

```
C  <-  c(1,1,1,1,1,1,1,1,1,21)

variancia <- function(vetor) {
  media <- mean(vetor)
  distancias  <-  (vetor - media)
```

```
    dist_quad <- distancias**2
    mean(dist_quad)
}

variancia(C)
```

```
## [1] 36
```

No entanto o valor ficou muito maior do que esperado, pior que o primeiro cálculo que resultou em 20.

Standard Deviation (Desvio Padrão)

Uma forma de solucionar o problema que a variância apresenta é extrair a raíz quadrada do resultado, assim chegando num valor que represente uma escala mais próxima da realidade e mais fácil de interpretar.

$$standard\,deviation = \sqrt{\frac{(x_1 - \mu)^2 + (x_2 - \mu)^2 + ...(x_N - \mu)^2}{N}} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{N}}$$

Em outras palavras o desvio padrão é a extração da raíz quadrada da variância

$$standard\,deviation = \sqrt{variance}$$

```
std_dev <- function(vetor) {
    media <- mean(vetor)
    distancias  <-  (vetor - media)
    squared_dist <- distancias**2
    sqrt(mean(squared_dist))
}

std_dev(C)
```

```
## [1] 6
```

Vamos entender melhor como interpretar esse resultado olhando o Preço das Casas na base.

```r
std_dev(base$SalePrice)
```

```
## [1] 79873.06
```

```r
mean(base$SalePrice)
```

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

Esse resultado nos mostra que o preço médio das casas giram em torno de 180 mil, mas não significa que a maioria delas ou cada uma delas custam esse valor. Uma casa poderia custar 120 e outra 240 mil, e até mesmo nenhuma casa ter o valor exato da média.

Com o desvio padrão temos uma noção melhor dessa variação, o resultado aproximado de 79 mil informa que a maioria das casas giram em torno de 79 mil abaixo ou acima da média. Vamos tentar visualizar isso num gráfico:

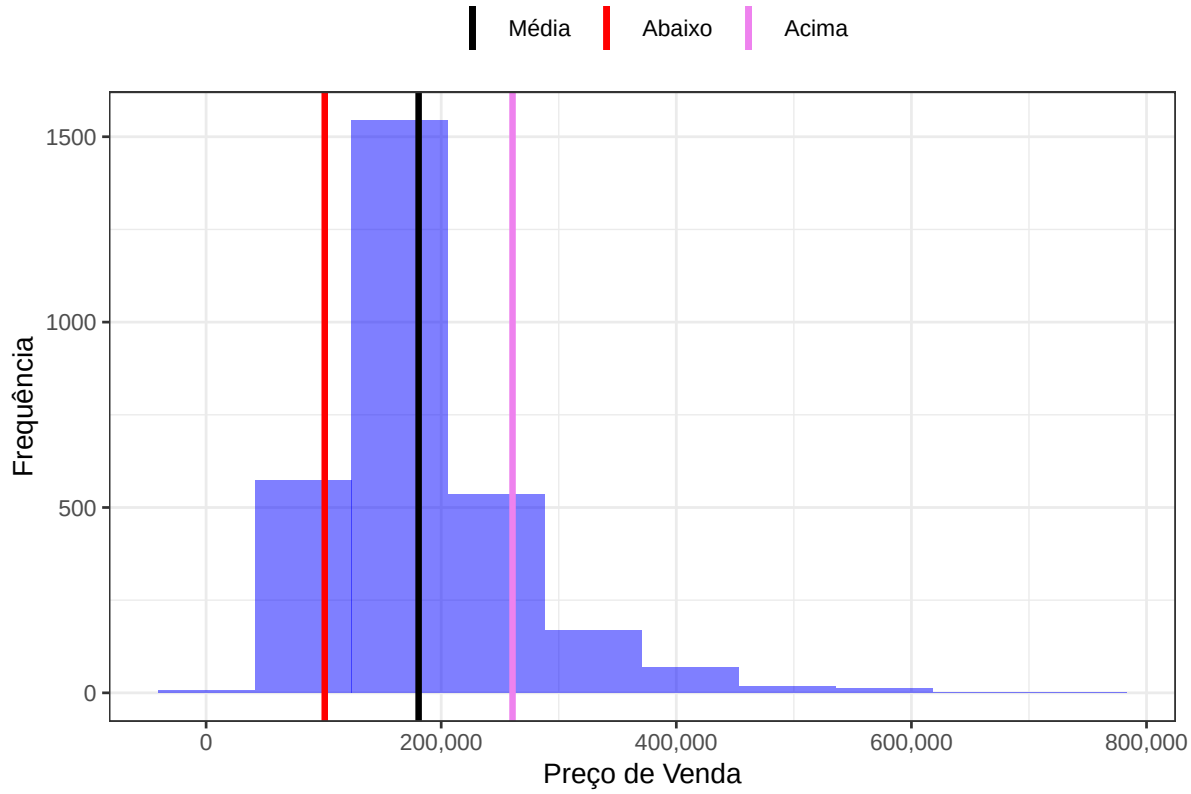
```r
library(ggplot2)

media <- mean(base$SalePrice)
desvio_padrao <- std_dev(base$SalePrice)

ggplot(data = base, aes(x = SalePrice)) +
    geom_histogram(bins = 10,
        position = "identity",
        alpha = 0.5,
        fill='blue') +
    geom_vline(aes(xintercept = media,
                   color = 'black'),
               size = 1.2 ) +
    geom_vline(aes(xintercept = media - desvio_padrao,
                   color = 'red'),
               size = 1.2 ) +
    geom_vline(aes(xintercept = media + desvio_padrao,
                   color = 'violet'),
               size = 1.2 ) +
    scale_x_continuous(labels = scales::comma) +
    scale_colour_manual(values = c("black", "red", "violet"),
                        name = "",
                        labels = c("Média", "Abaixo", "Acima")) +
    theme_bw() +
    theme(legend.position='top') +
```

```
    xlab("Preço de Venda") +
    ylab("Frequência")
```

```
## 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.
```



No gráfico vemos que os valores não param necessariamente entre 101 e 259 mil (1 desvio padrão abaixo ou acima da média), esses valores só indicam onde a maior parte das ocorrências se concentram, mas ainda assim podem existir outliers.

Dispersão dos dados

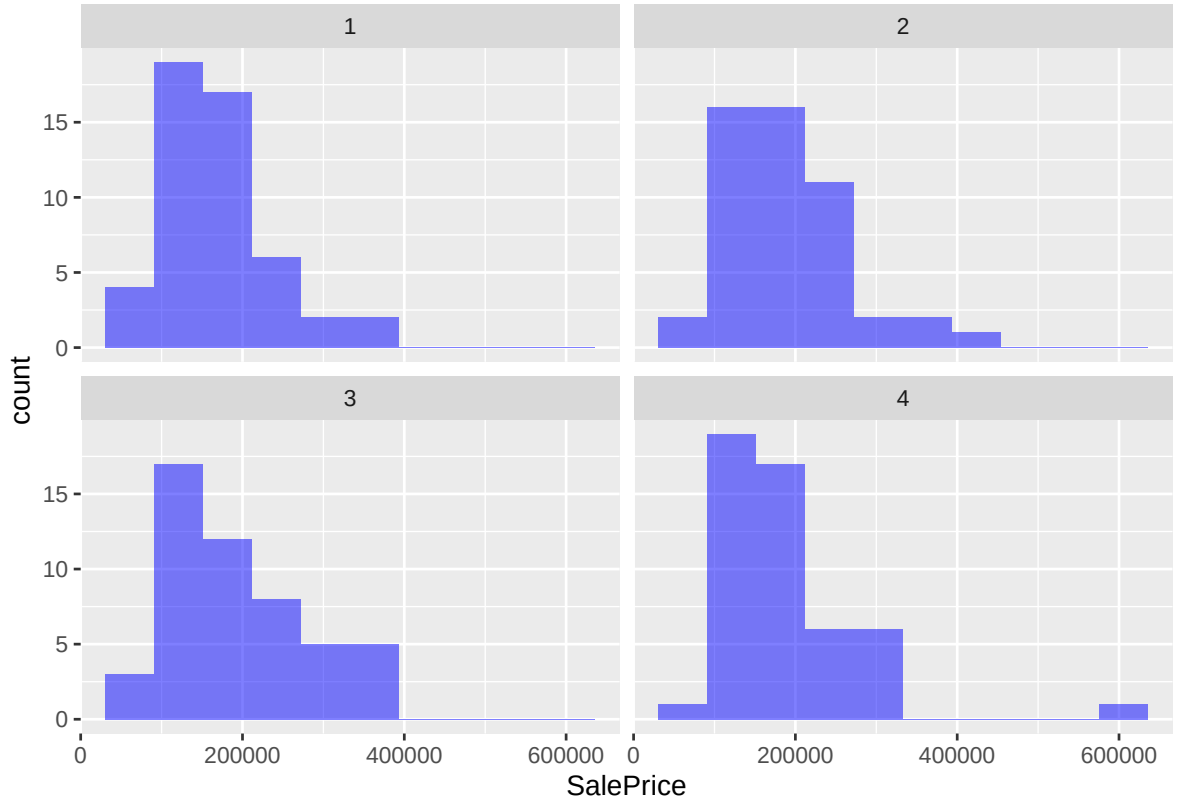
Vamos comparar o comportamento do desvio padrão em amostras aleatórias, propositalmente pequenas para notarmos as diferenças.

```r
set.seed(2)
amostras  <-  purrr::map(1:4, function(x) sample_n(base, size = 50))
dp_amostras <- purrr::map(1:4, function(i) std_dev(amostras[[i]]$SalePrice))
dp_amostras
```

```
## [[1]]
## [1] 66179.46
##
## [[2]]
## [1] 74755.68
##
## [[3]]
## [1] 81655.67
##
## [[4]]
## [1] 82162.53
```

```r
amostras_consolidado <- bind_rows(amostras, .id="No_Amostra")

ggplot(data = amostras_consolidado, aes(x = SalePrice)) +
    geom_histogram(bins = 10,
        position = "identity",
        alpha = 0.5,
        fill='blue')+
  facet_wrap(vars(No_Amostra))
```



Normalmente queremos tomar decisões sobre uma população inferindo a partir de uma amostra, mas o quanto o resultado baseado numa amostra é confiável?

Até o momento apresentamos a fórmula, com base na população, a representação matemática fica levemente diferente quando falamos em amostras, o símbolo da média e da quantidade de ocorrências muda:

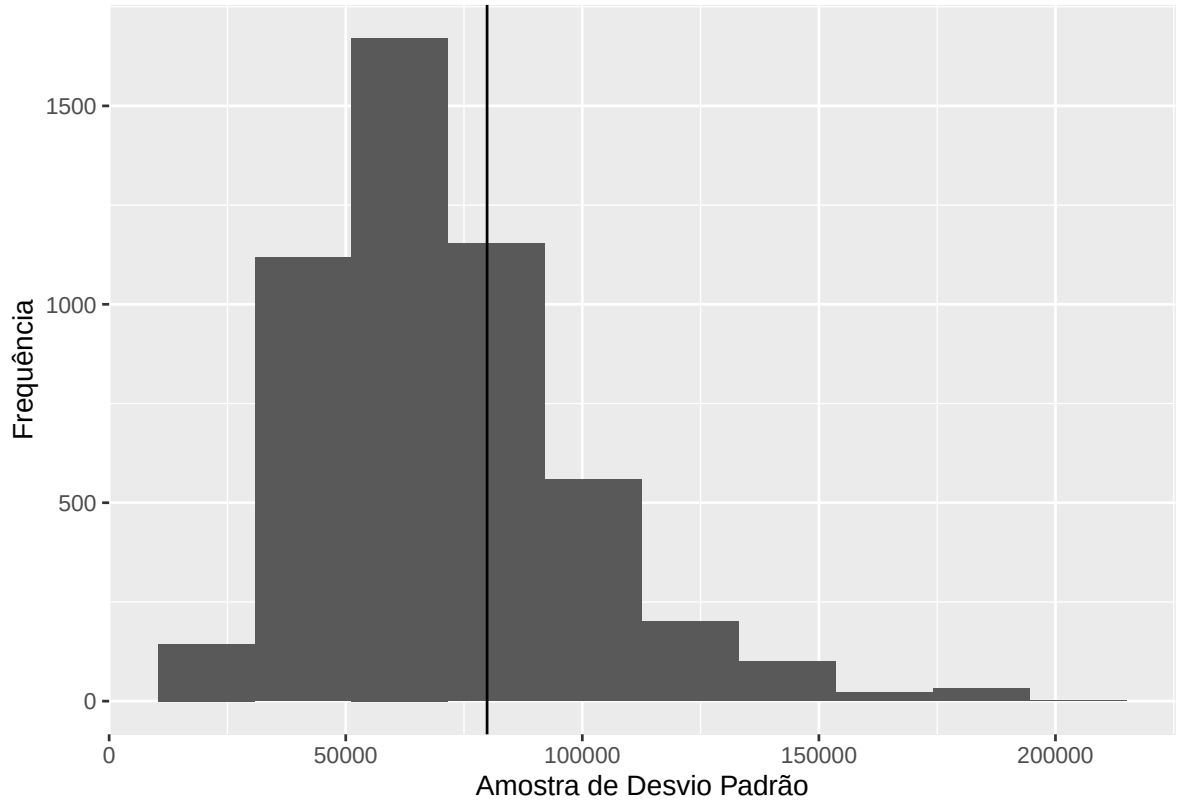
$$standard\,deviation = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + ...(x_n - \bar{x})^2}{n}} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n}}$$

Vamos criar repetidas amostras e calcular o desvio padrão de cada uma delas e plotar num histograma para observar o quanto essa informação muda dependendo da amostra comparado ao desvio padrão da população.

```
set.seed(2)
std_points <- replicate(n = 5000,
                        expr=std_dev(sample(x=base$SalePrice, size=10)))

std_points_tibble <- tibble::tibble(std_points)
```

```
ggplot(data=std_points_tibble, aes(std_points))+
geom_histogram(bins=10, position="identity")+
geom_vline(aes(xintercept=std_dev(base$SalePrice)))+
xlab("Amostra de Desvio Padrão")+
ylab("Frequência")
```



Notamos que em média o desvio padrão da amostra subestima o desvio padrão da população, em outras palavras, o **desvio padrão da amostra** normalmente será menor do que o **desvio padrão da população**. Vamos entender por que isso acontece e como corrigir.

Correção de Bessel

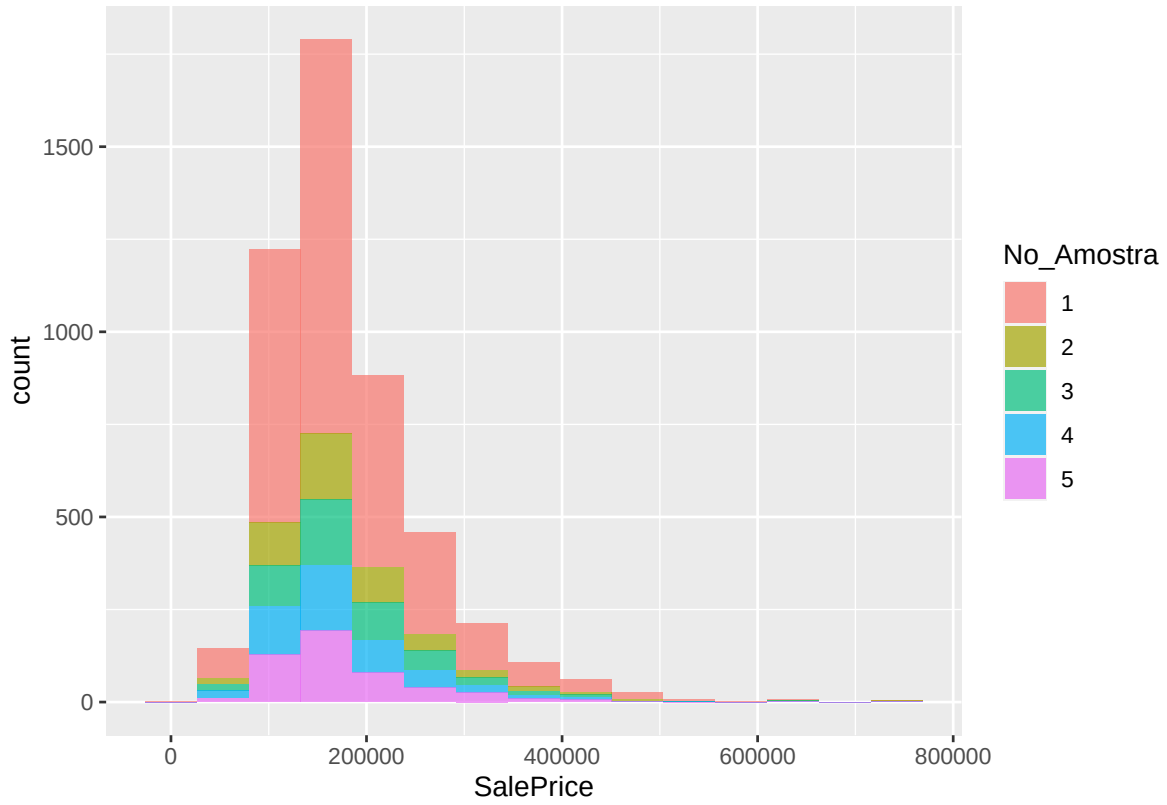
Abaixo veja a distribuição da base completa com 2930 linhas categorizado como No_Amostra 1, e as demais (2,3,4,5) são amostras de 500 linhas extraídas da base completa aleatoriamente.

A primeira coisa que temos que notar, que já de cara explica por que o desvio padrão da amostra é sempre menor que da população, é que a amostra nunca ultrapassa os limites da população. No desenho do gráfico as barras estão sempre contidas dentro

da imagem da base completa, porém como possui um volume menor a concentração dos dados vão ser mais estreitos. A probabilidade de uma amostra conseguir captar um registro pelo menos de cada variação de preço para representar toda a dispersão de 0 a 800 mil no preço da casa é muito raro.

```
amostras  <-  purrr::map(1:4, function(x) sample_n(base, size = 500))
amostras_consolidado <- bind_rows(base,
                                  amostras,
                                  .id="No_Amostra")

ggplot(data = amostras_consolidado, aes(x = SalePrice, fill= No_Amostra)) +
    geom_histogram(bins=15, alpha = 0.7)
```



Ainda assim existe uma forma de corrigir a fórmula do desvio padrão a fim de tentar captar um valor que represente melhor o comportamento da população. Fazemos isso diminuindo em 1 o valor do denominador, assim na divisão da fórmula vai aumentar o valor do resultado final.

$$standard\,deviation = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + ...(x_n - \bar{x})^2}{n-1}} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n-1}}$$

Vamos ajustar nossa função:

```
std_dev_bessel <- function(vetor) {
    distancias  <-  (vetor - mean(vetor))
    squared_dist <- distancias**2
    sqrt(sum(squared_dist)/(length(squared_dist)-1))
}
```

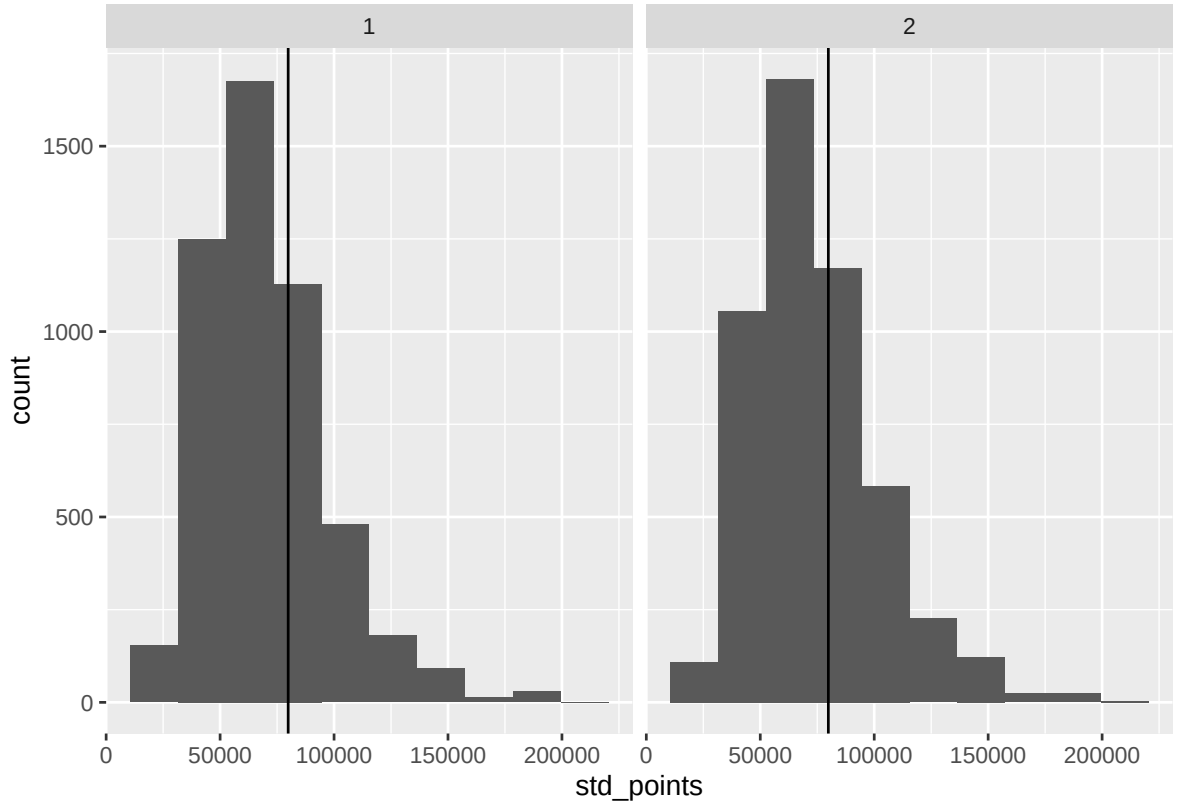
E agora o resultado do antes da correção (1) e depois (2). Notamos uma ligeira mudança, um deslocamento da concentração dos dados nas barras mais à direita, ou seja os valores do desvio padrão aumentaram um pouco.

```
set.seed(1)
std_points <- replicate(n = 5000,
                        expr=std_dev(sample(x=base$SalePrice, size=10)))
std_points_tibble <- tibble::tibble(std_points)

std_points_bessel <- replicate(n = 5000,
                               expr=std_dev_bessel(sample(x=base$SalePrice, size=10)))
std_points_bessel_tibble <- tibble::tibble(std_points_bessel) %>%
  rename("std_points" = "std_points_bessel")

std_points_tibble_full <-  bind_rows(std_points_tibble,
                                     std_points_bessel_tibble,
                                     .id="Bessel")

ggplot(data=std_points_tibble_full, aes(std_points))+
geom_histogram(bins=10, position="identity")+
geom_vline(aes(xintercept=std_dev(base$SalePrice)))+
facet_wrap(vars(Bessel))
```



Claro que poderíamos seguir diminuindo o denominador para tentar alcancar melhor o desvio padrão da população. Mas o que precisamos ter em mente é que aqui se trata de um exercício didático, no dia a dia com bases volumosas, não teremos certeza do valor da população e apenas trabalharemos com amostras. Além disso, usar a correção n-1 é um consenso entre os estatísticos de ser a melhor escolha para o cálculo.

Vamos por fim recapitular as notações matemáticas e suas fórmulas.

Lembrando que temos uma fórmula de desvio padrão para a população e outra para a amostra, e na amostra que aplicamos a correção de Bessel.

$$variancia\ (populacao) = \sigma^2$$
$$variancia\ (amostra) = s^2$$

$$desvio\ padrao\ (populacao) = \sqrt{}variancia = \sqrt{\sigma}^2 = \sigma$$

$$desvio\ padrao\ (amostra) = \sqrt{}variancia = \sqrt{s}^2 = s$$

Por fim as fórmulas:

Desvio Padrão População

$$\sigma = \sqrt{\frac{(x_1 - \mu)^2 + (x_2 - \mu)^2 + ...(x_N - \mu)^2}{N}} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \mu)^2}{N}}$$

Desvio Padrão Amostra (com correção de Bessel)

$$s = \sqrt{\frac{(x_1 - \bar{x})^2 + (x_2 - \bar{x})^2 + ...(x_n - \bar{x})^2}{n - 1}} = \sqrt{\frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n - 1}}$$

Para fins didáticos criamos as funções manualmente, mas vamos comparar o resultado da função criada com a função base existente no R?

```
amostra <- sample(x=base$SalePrice, size=10)

std_dev_bessel(amostra)
```

```
## [1] 66186.74
```

```
sd(amostra)
```

```
## [1] 66186.74
```