Simple Random Sampling

Como a Estatística pode nos ajudar?

A estatística é extremamente necessária na análise de grande volume de dados. Com técnicas estatísticas podemos organizar, sumarizar e visualizar os dados nos ajudando a extrair insights para resolução de problemas.

Por exemplo, imagine que uma empresa deseja realizar uma pesquisa de satisfação com seus funcionários. Para uma empresa com menos de 10 funcionários é fácil olhar os dados individualmente e chegar em uma conclusão. Agora quando esse volume ultrapassa 100, 1000, ou 10000 funcionários, olhar individualmente se torna praticamente impossível. Sendo assim podemos fazer cálculos, tabelas frequência e utilizar gráficos para melhor interpretar esses dados.

Amostra vs População

Agora imagine uma situação onde queremos aplicar uma pesquisa com os moradores da cidade de São Paulo, por mais que através de técnicas estatísticas consigamos analisar milhões de dados, aplicar uma pesquisa para tamanha população pode ser muito custosa e levar tempo para concluir. É aqui que podemos usar a técnica da amostragem e selecionar uma parte dessa população, alguns milhares, para aplicar a pesquisa e conseguir extrapolar os resultados para toda a população.

Veja que nesse exemplo, os paulistanos é apenas uma parcela das pessoas moradoras do Estado de São Paulo, ou do Brasil, mas como o público de interesse é apenas Paulistanos, então os Paulistanos é a nossa **População** estatísticamente falando, e a parcela dessa população selecionada para aplicar a pesquisa é nossa **Amostra**. Sendo assim usamos a amostra para conseguir responder perguntas e chegar à conclusões que diz respeito ao todo (População).

Nesses exemplos utilizamos pessoas, mas o objeto de análise pode ser animais, países, vegetais e etc, então podemos chamar de *indivíduos*, *unidades*, *eventos*, ou *observações*. Então quando falamos de população podemos chamar de indivíduos da população, e quando falamos da amostra, indivíduos da amostra e assim por diante.

Explorando dados do WNBA

O dataset pode ser acessado através deste link. E o glossário dos termos neste link.

```
wnba <- read_csv("/home/marcella/Downloads/WNBA_Stats.csv")
```

```
## Rows: 143 Columns: 32
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): Name, Team, Pos, Birth_Place, Birthdate, College, Experience
## dbl (25): Height, Weight, BMI, Age, Games Played, MIN, FGM, FGA, FG%, 15:00,...
##
## 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(wnba)
```

```
## Rows: 143
## Columns: 32
## $ Name           <chr> "Aerial Powers", "Alana Beard", "Alex Bentley", "Alex M~
## $ Team           <chr> "DAL", "LA", "CON", "SAN", "MIN", "SEA", "PHO", "CHI", ~
## $ Pos            <chr> "F", "G/F", "G", "G/F", "G", "G", "G", "G", "G", "G", "~
## $ Height         <dbl> 183, 185, 170, 185, 175, 170, 188, 178, 185, 178, 180, ~
## $ Weight         <dbl> 71, 73, 69, 84, 78, 63, 81, 64, 76, 77, 76, 84, 113, 88~
## $ BMI            <dbl> 21.20099, 21.32944, 23.87543, 24.54346, 25.46939, 21.79~
## $ Birth_Place    <chr> "US", "US", "US", "US", "US", "US", "US", "US", "US", "~
## $ Birthdate      <chr> "January 17, 1994", "May 14, 1982", "October 27, 1990",~
## $ Age            <dbl> 23, 35, 26, 28, 23, 22, 23, 31, 24, 29, 30, 24, 24, 29,~
## $ College        <chr> "Michigan State", "Duke", "Penn State", "Georgia Tech",~
## $ Experience     <chr> "2", "12", "4", "6", "R", "R", "R", "8", "2", "5", "6",~
## $ `Games Played` <dbl> 8, 30, 26, 31, 24, 14, 16, 26, 30, 7, 30, 28, 25, 22, 4~
## $ MIN            <dbl> 173, 947, 617, 721, 137, 90, 112, 847, 834, 103, 843, 8~
## $ FGM            <dbl> 30, 90, 82, 75, 16, 9, 9, 166, 131, 14, 93, 154, 20, 18~
## $ FGA            <dbl> 85, 177, 218, 195, 50, 34, 34, 319, 346, 38, 183, 303, ~
## $ `FG%`          <dbl> 35.3, 50.8, 37.6, 38.5, 32.0, 26.5, 26.5, 52.0, 37.9, 3~
## $ `15:00`        <dbl> 12, 5, 19, 21, 7, 2, 4, 70, 29, 2, 20, 0, 2, 0, 0, 1, 0~
## $ `3PA`          <dbl> 32, 18, 64, 68, 20, 9, 15, 150, 103, 11, 62, 3, 8, 10, ~
## $ `3P%`          <dbl> 37.5, 27.8, 29.7, 30.9, 35.0, 22.2, 26.7, 46.7, 28.2, 1~
## $ FTM            <dbl> 21, 32, 35, 17, 11, 6, 2, 40, 104, 6, 38, 91, 9, 5, 0, ~
## $ FTA            <dbl> 26, 41, 42, 21, 12, 6, 2, 46, 129, 6, 51, 158, 12, 8, 0~
## $ `FT%`          <dbl> 80.8, 78.0, 83.3, 81.0, 91.7, 100.0, 100.0, 87.0, 80.6,~
```

```
## $ OREB           <dbl> 6, 19, 4, 35, 3, 3, 1, 9, 52, 3, 29, 34, 5, 12, 0, 16, ~
## $ DREB           <dbl> 22, 82, 36, 134, 9, 13, 14, 83, 75, 7, 97, 158, 18, 28,~
## $ REB            <dbl> 28, 101, 40, 169, 12, 16, 15, 92, 127, 10, 126, 192, 23~
## $ AST            <dbl> 12, 72, 78, 65, 12, 11, 5, 95, 40, 10, 50, 136, 7, 5, 1~
## $ STL            <dbl> 3, 63, 22, 20, 7, 5, 4, 20, 47, 5, 22, 48, 4, 3, 2, 1, ~
## $ BLK            <dbl> 6, 13, 3, 10, 0, 0, 3, 13, 19, 0, 4, 11, 5, 9, 0, 11, 2~
## $ TO             <dbl> 12, 40, 24, 38, 14, 11, 3, 59, 37, 2, 32, 87, 12, 6, 3,~
## $ PTS            <dbl> 93, 217, 218, 188, 50, 26, 24, 442, 395, 36, 244, 399, ~
## $ DD2            <dbl> 0, 0, 0, 2, 0, 0, 0, 0, 0, 0, 0, 4, 0, 0, 0, 0, 0, 0, 0~
## $ TD3            <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
```

Vamos calcular a maior quantidade de jogos que as jogadoras participaram. Quando fazemos esse cálculo para a população chamamos de **parâmetro**, enquanto que para a amostra chamamos de **estatística**. Na amostra é comum o cálculo ser diferente, afinal é uma parcela das informações, está imcompleta. Chamamos essa divergência de **erro amostral**, e pode ser calculada através da diferença entre o parâmetro e a estatística.

```
parameter <- max(wnba$`Games Played`, na.rm=TRUE)
glue("Parâmetro: {parameter}")
```

```
## Parâmetro: 32
```

```
set.seed(1)
sample <- sample(wnba$`Games Played`, size=30)
print("Amostra")
```

```
## [1] "Amostra"
```

```
sample
```

```
##  [1] 30 28 28 22 26  7 29 27 14 16 29 30 16 29 28 29 25 30 30 22 31 29 28 29 23
## [26] 29 17 18 22 26
```

```
statistic <- max(sample, na.rm=TRUE)
glue("Estatística: {statistic}")
```

```
## Estatística: 31
```

```r
sampling_error <- parameter-statistic
glue("Erro Amostral: {sampling_error}")
```

```
## Erro Amostral: 1
```

Queremos sempre alcançar o menor erro amostral possível, para ter resultados confiáveis que se pareçam com o resultado da população. Para isso a amostra precisa ser **representativa** o suficiente, com característica muito parecidas com a população.

Para garantir essa representatividade, usamos métodos que selecione os indivíduos de forma aleatória, assim dando chances iguais para um indivíduo ser selecionado na amostra. Chamamos de **Amostragem Aleatória Simples**

Usamos a função `sample` que já faz essa amostragem aleatória em vetores, mas para dataframes podemos utilizar a função `slice_sample`. Estas funções têm o parâmetro replacement que indica se cada observação pode ou não se repetir, ser selecionado mais de uma vez, o padrão é FALSE (sem repetição).

Como a amostragem é aleatória, usamos o código `set.seed(1)`, e escolhemos um número que será mantido em todas execuções, assim é possível repetir o mesmo cenário em cada execução. O que é muito útil para debugar, ou para dar andamento em estudos que não vamos concluir numa única execução, ou até para tornar o estudo reproduzível por outras pessoas também. Isso garante que o resultado será o mesmo, ainda que a amostra seja de fato aleatória.

Agora vamos ver como se comporta a média do número de pontos feitos por temporada por cada jogadora (PTS) em diversas amostras com tamanho de 10 observações.

```r
mean_points<- replicate(100, mean(sample(wnba$PTS,size=10)))
mean_points
```

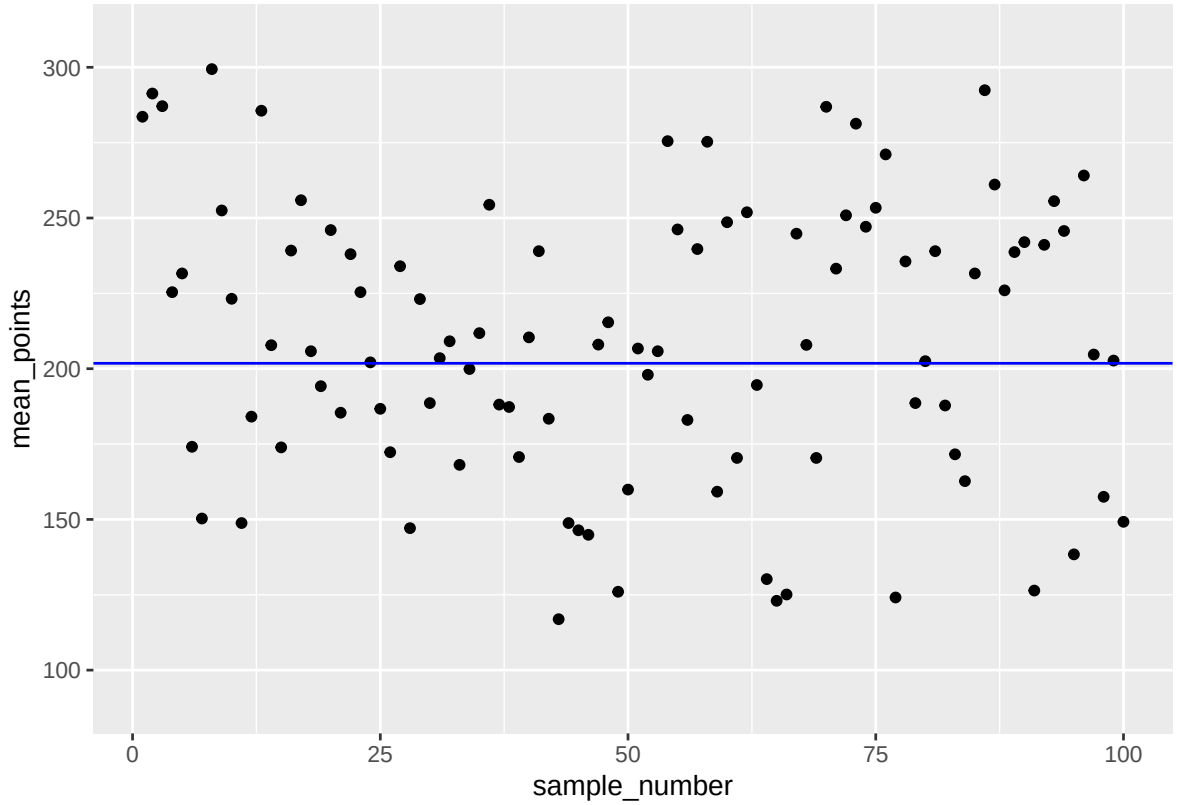
```
##   [1] 270.3 230.3 247.5 208.6 171.8 307.9 206.3 171.1 233.4 135.9 196.0  94.0
##  [13] 136.5 176.8 156.0 271.1 275.2 197.3 195.8 183.8 181.9 148.3 295.3 193.5
##  [25] 188.2 177.0 187.3 198.1 230.9 264.0 221.9 193.3 251.9 143.0 203.4 214.4
##  [37] 245.7 139.7 290.3 231.6 173.9 152.7 258.4 209.0 192.2 128.6 224.1 200.7
##  [49] 233.0 225.9 202.6 284.3 132.8 276.7 187.5 195.5 233.2 158.9 226.2 160.5
##  [61] 177.0 182.4 121.7 196.9 305.8 211.4 275.7 233.9 199.2  88.0 230.0 154.4
##  [73] 238.9 241.1 125.9 141.0 222.4 252.2 264.1 197.4 276.5 238.1 100.2 146.9
##  [85] 195.5 252.0 200.3 254.0 246.3 171.3 173.4 208.1 112.5 202.8 132.8 190.6
##  [97] 190.7 251.9 172.2 149.5
```

Percebemos que a média variou bastante, mas vamos visualizar melhor com a ajuda de um Scatter Plot

```
mean_points<- replicate(100, mean(sample(wnba$PTS,size=10)))
sample_number <- 1:100

df <- tibble(sample_number,mean_points)

ggplot(df, aes(sample_number,mean_points))+
  geom_point()+
  geom_hline(yintercept = mean(wnba$PTS),
    color = "blue") +
    ylim(90, 310)
```



O gráfico nos ajudou a notar que o resultado de cada amostra está muito aleatório e não se aproximando da média real da população. Uma forma de corrigir isso é aumentando o tamanho de observações de cada amostra, isso ajuda a diminuir o erro amostral também.

```
mean_points<- replicate(100, mean(sample(wnba$PTS,size=100)))
sample_number <- 1:100
```

```r
df <- tibble(sample_number,mean_points)

ggplot(df, aes(sample_number,mean_points))+
  geom_point()+
  geom_hline(yintercept = mean(wnba$PTS),
    color = "blue") +
    ylim(90, 310)
```