Stratified Sampling and Cluster Sampling

No último exercício aprendemos a amostrar os dados de uma população usando a estratégia da Amostragem Aleatória Simples. No entanto essa abordagem funciona bem em dados uniformes. Quando temos uma população com características em maior abundância que outras, as características menos representadas podem acabar sendo omitidas da amostragem. Uma amostragem que não é representativa, que não se assemelha a população original, pode acabar enviesando os resultados e por consequência chegarmos a conclusões equivocadas.

Com o objetivo de criar uma amostra representativa, podemos realizar uma Amostragem Estratificada, na qual tentamos replicar a característica e proporção dessas características da população na amostra.

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 por exemplo calcular a média de pontos por temporada da população e comparar com o resultado da amostra simples.

```
mean(wnba$PTS)
```

```
## [1] 201.7902
```

```r
set.seed(1)
amostra <- sample_n(wnba, 10)

mean(amostra$PTS)
```

```
## [1] 171.4
```

O resultado da média da amostra se distanciou muito da média da população. Provavelmente isso se dá, pois no dataset temos informações de jogadoras do basquete de diferentes posições que portanto pontuam de forma diferente. Se a amostra não garantir a mesma proporção de posições, a média vai distoar da população.

Vamos então ver a média por posição para conferir se essa ideia faz sentido.

```r
wnba %>%
group_by(Pos) %>%
sample_n(10) %>%
summarise(mean_pts_season = mean(PTS)) %>%
arrange()
```

```
## # A tibble: 5 x 2
##   Pos   mean_pts_season
##   <chr>           <dbl>
## 1 C                109.
## 2 F                271.
## 3 F/C              220.
## 4 G                247.
## 5 G/F              185.
```

Um outro fator que pode estar influenciando na pontuação é a quantidade de jogos que cada jogadora participou, afinal quem participou de mais jogos tem mais chances de pontuar mais.

```r
min(wnba$`Games Played`)
```

```
## [1] 2
```

```r
max(wnba$`Games Played`)
```

```
## [1] 32
```

De fato tem muita variação no dataset, tem jogadoras que partiiciparam de apenas 2 jogos, enquanto que outras participaram de até 32 partidas.

E se a gente separar em por exemplo 3 grupos de acordo com a quantidade de partidas que as jogaras participaram, como influencia essa média?

```
wnba <- wnba %>% mutate(games_cut = cut(`Games Played`, breaks = 3))

wnba%>%
  group_by(games_cut) %>%
  summarise(qtd = n()) %>%
  mutate(Perc = qtd/sum(qtd) * 100) %>%
  arrange(desc(Perc))
```

```
## # A tibble: 3 x 3
##   games_cut   qtd  Perc
##   <fct>     <int> <dbl>
## 1 (22,32]     104 72.7
## 2 (12,22]      26 18.2
## 3 (1.97,12]    13  9.09
```

Com o cálculo acima vemos que +70% da população tem participação entre 23 a 32 partidas, enquanto que 9% participa de até 12 partidas.

Vamos então criar uma função para gerar uma amostra que respeite essa proporção e ver como isso afeta a média. Assim como nos exercício anterior, vamos ver isso num gráfico e gerando dezenas de amostras para perceber o efeito desse cálculo.

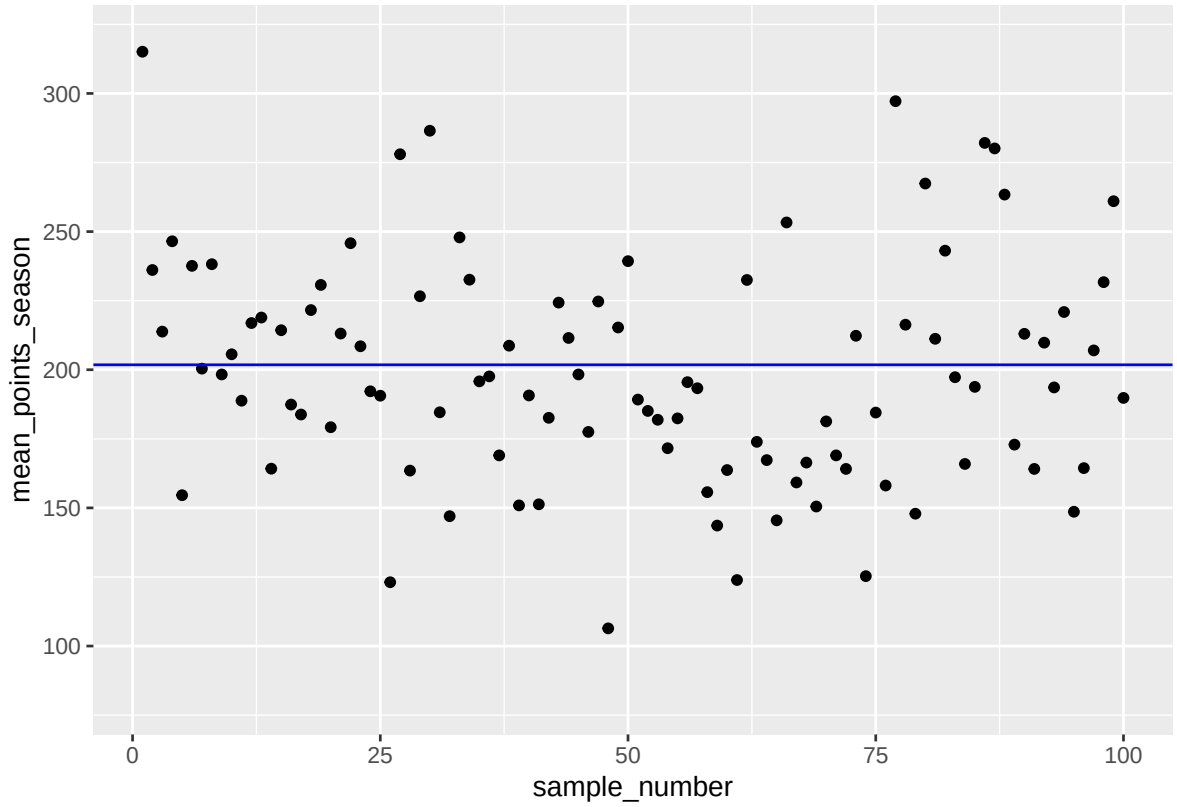
```
sample_mean <- function(x){
  under_12 <- wnba %>%
    filter(`Games Played` <= 12) %>%
    sample_n(1)
  btw_13_22 <- wnba %>%
    filter(`Games Played` > 12 & `Games Played` <= 22) %>%
    sample_n(2)
  over_22 <- wnba %>%
    filter(`Games Played` > 22) %>%
    sample_n(7)

  combined <- bind_rows(under_12, btw_13_22, over_22)
  mean(combined$PTS)
}
```

```
sample_number <- 1:100
mean_points_season <- map_dbl(sample_number, sample_mean)

df <- tibble(sample_number, mean_points_season)

ggplot(data = df,
       aes(x = sample_number, y = mean_points_season)) +
    geom_point() +
    geom_hline(yintercept = mean(wnba$PTS),
               color = "blue") +
    ylim(80, 320)
```



Existe uma forma mais eficiente de criar uma amostra estratificada usando a função `sample_frac`. Ao invés de informar a quantidade absoluta de linhas, podemos passar o valor percentual desejado. Nesse exemplo de 10 registros em uma base de 143 observações, temos 7% da base, portando passamos 0.07 como parâmetro da função. O uso do group_by fará com que a função respeite a proporção da variável de interesse.

```r
sample <- wnba %>%
  group_by(games_cut) %>%
  sample_frac(0.07)

sample %>%
  group_by(games_cut) %>%
  summarise(qtd = n()) %>%
  mutate(Perc = qtd/sum(qtd) * 100) %>%
  arrange(desc(Perc))
```

```
## # A tibble: 3 x 3
##   games_cut   qtd  Perc
##   <fct>     <int> <dbl>
## 1 (22,32]       7    70
## 2 (12,22]       2    20
## 3 (1.97,12]     1    10
```

No entanto percebemos no gráfico que essa abordagem de amostrar considerando a quantidade de partidas em que cada jogadora participou não foi eficiente. Pensando no caso de uma jogadora ter jogado apenas 10 minutos, enquanto outra jogou 40, ainda assim ambas jogaram uma partida, mas as oportunidades de marcar pontos são diferentes. Sendo assim faz mais sentido estratificar a amostra pensando no tempo jogado por cada jogadora.

```r
wnba <- wnba %>% mutate(min_cut = cut(MIN, breaks = 3,
                                      dig.lab = max(nchar(MIN),na.rm = T)))

wnba %>% group_by(min_cut) %>%
  summarise(qtd=n()) %>%
  mutate(Perc = qtd/sum(qtd))
```

```
## # A tibble: 3 x 3
##   min_cut         qtd  Perc
##   <fct>         <int> <dbl>
## 1 (10.99,347.3]    48 0.336
## 2 (347.3,682.7]    50 0.350
## 3 (682.7,1019]     45 0.315
```

Reproduzindo o mesmo gráfico agora com a amostra baseada nos minutos jogados, podemos enxergar uma melhor performance na média das amostras se aproximando mais da média da população do que na tentativa anterior.

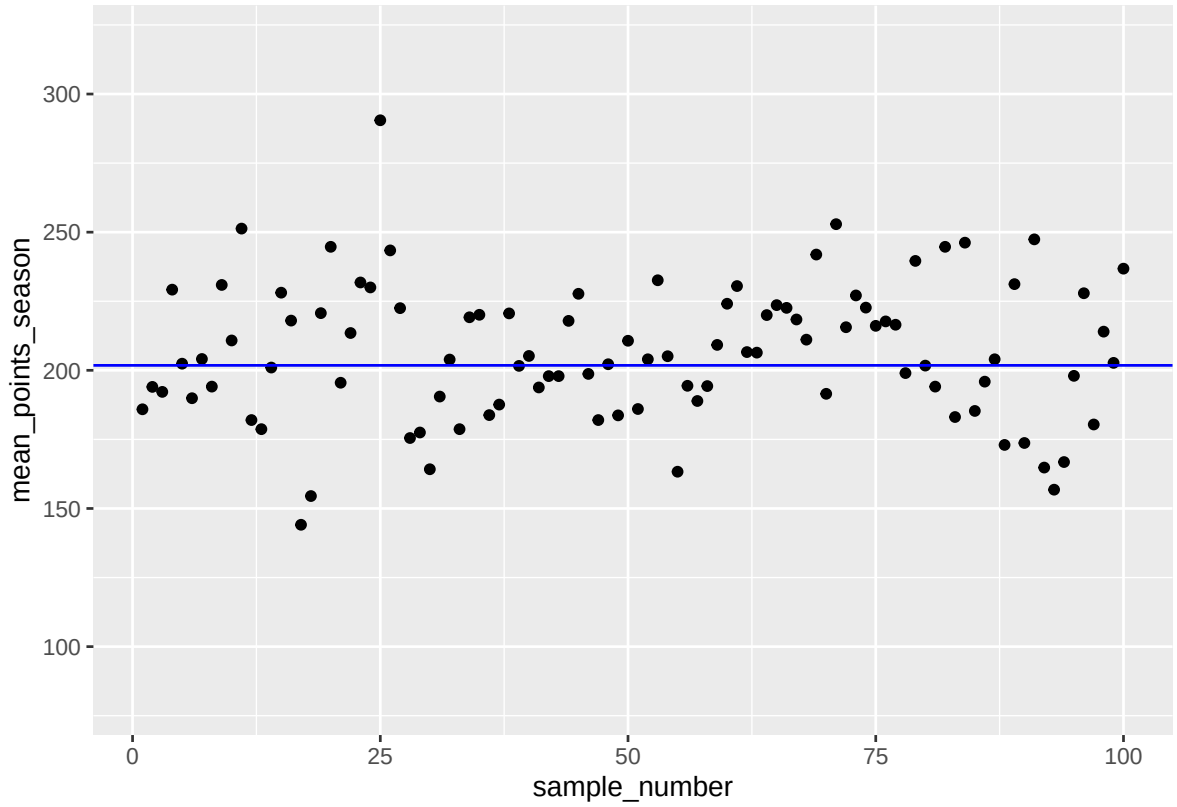
```r
sample_mean <- function(x){
  sample <- wnba %>%
  group_by(min_cut) %>%
  sample_frac(0.07)

  mean(sample$PTS)
}

sample_number <- 1:100
mean_points_season <- map_dbl(sample_number, sample_mean)

df <- tibble(sample_number, mean_points_season)

ggplot(data = df,
       aes(x = sample_number, y = mean_points_season)) +
    geom_point() +
    geom_hline(yintercept = mean(wnba$PTS),
               color = "blue") +
    ylim(80, 320)
```



Clustering Sample

Imagine que coletar as informações de jogos de basquete do WNBA não fosse tão simples e para cada time houvesse um site diferente para essa coleta. Coletar de todos os lugares pode consumir muito tempo ou dependendo da situação ser impossível (pode ser uma consulta com um custo que não estamos dispostos a pagar por todos registros por exemplo). Nesse caso poderíamos escolher alguns sites/times para fazer essa coleta, imaginando que cada time é um cluster, estaríamos selecionando/amostrando alguns clusters para a análise

```
clusters <- unique(wnba$Team) %>% sample(4)
clusters
```

```
## [1] "MIN" "SAN" "LA"  "DAL"
```

```
sample <- wnba %>%
filter(Team %in% clusters)

smp_height <- mean(sample$Height)
smp_height
```

```
## [1] 184.75
```

```
wnba_height <- mean(wnba$Height)
wnba_height
```

```
## [1] 184.5664
```

```
sampling_error_height <- wnba_height - smp_height
sampling_error_height
```

```
## [1] -0.1835664
```