Projeto Final do Módulo: Linguagem R

true

1. Tarefas do projeto final

O projeto final do módulo "Linguagem R" determina que seja criado um dataset com pelo menos 5 colunas numéricas e 3 colunas categóricas, através do uso de funções de criação de distribuições aleatórias.

Depois que o dataset for criado ele será gravado em disco e algumas tarefas básicas de análise de dados devem ser realizadas:

1. Calcular somas e médias através do uso das funções `apply`, `lapply`, `sapply`, etc.
2. Usar a função `split`
3. Criar gráficos simples
4. Colocar o projeto no GitHub

Nenhuma outra especificação a respeito de como deve ser o dataset foi fornecida então o que vale aqui é a criatividade na definição das variáveis do dataset e a demonstração de que o aluno aprendeu pelo menos o básio da linguagem R.

2. Conteúdo do meu dataset fictício

Parece claro para mim que a primeira coisa é definir o assunto (o *subject*) do dataset para depois definir quais seriam as variáveis (colunas) a serem simuladas. Assim, defini que meu dataset fictício se trata de uma ***pesquisa a respeito de obesidade*** em adultos jovens na qual foram coletados dados antropométricos, demográficos e de alguns fatores de risco.

As variáveis coletadas estão detalhadas na tabela abaixo (incluindo o tipo da variável, sua representação em R e outras informações importantes):

Variável	Observações	Tipo	Representação no R
idade	Em anos completos	Dimensional de razão, discreta	Numeric
altura	Em metros (m)	Dimensional de razão, contínua	Numeric
peso	Em quilo (Kg)	Dimensional de razão, contínua	Numeric
imc	peso/altura^2	Índice dimensional de razão, contínuo	Numeric
sexo	1 = Masculino 2 = Feminino	Nominal	Unordered Factor
escolaridade	0 = Analfabeto 1 = 1º grau completo 2 = 2º grau completo 3 = 3º grau completo 4 = mestrado 5 = doutorado 6 = pós-doutorado	Ordinal	Ordered Factor
profissao	1 = Humanas 2 = Exatas 3 = Biológicas	Nominal	Unordered Factor
fumante	0 = Não 1 = Sim	Binária	Ordered Factor
salario	Em reais (R$)	Dimensional de razão, contínua	Numeric
carros	Número de carros	Dimensional de razão, discreta	Numeric
filhos	Número de filhos	Dimensional de razão, discreta	Numeric

3. Simulação das variáveis do dataset

Arbitrariamente decidi que o dataset conteria informações simuladas de 10.000 observações, retiradas aleatoriamente por algum processo de amostragem a partir de uma população de 100.000 indivídos. Esses parâmetros são definidos abaixo:

```
n <- 10000
p <- 100000
```

3.1. Variáveis dimensionais

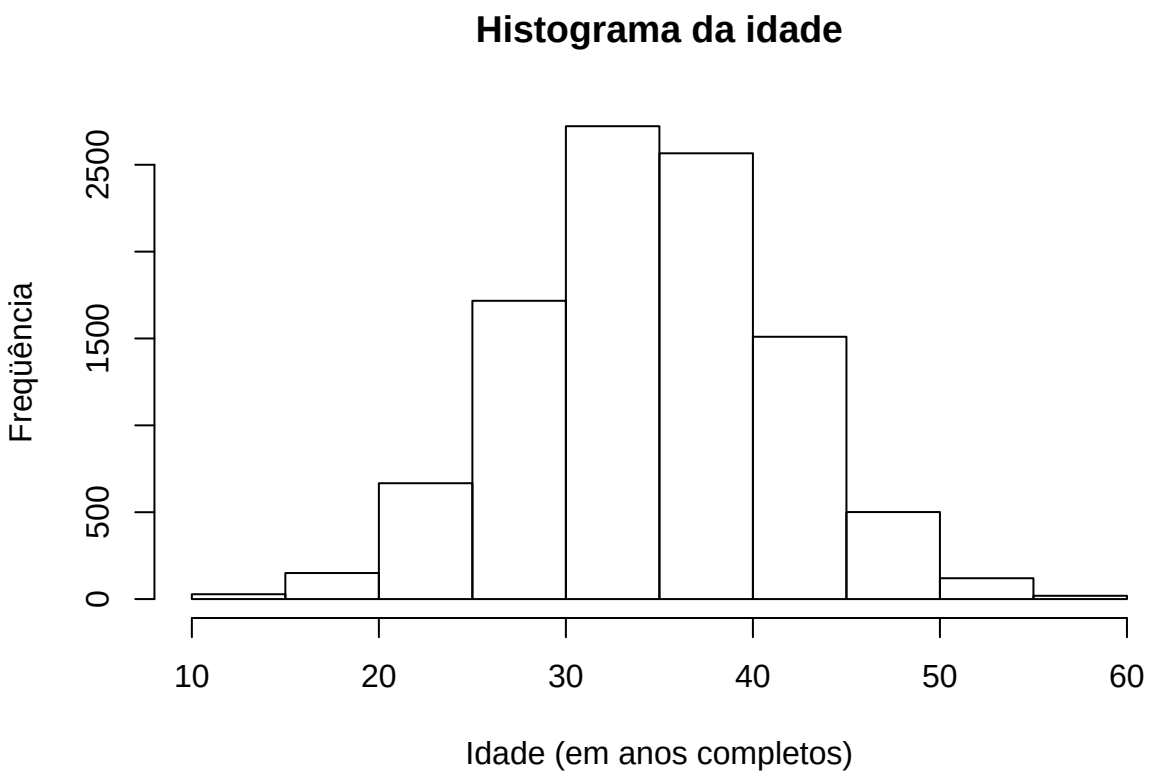
3.1.1. Idade

A variável idade (em anos completos) foi simulada a partir de uma distrituição normal, com o uso da função `rnorm` ajustada para uma média de idade de 37 anos com desvio padrão de 7 anos. Para evitar quaisquer números negativos foi utilizada a função `abs` e para manter a idade em anos completos o resultado foi arredondado para zero casas decimais com a função `round`. A função `set.seed` foi utilizada para tornar os resultados reprodutíveis.

```
set.seed(1234)
idade <- abs(round(rnorm(n, 35, 7),0))
summary(idade)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   11.00   30.00   35.00   35.04   40.00   60.00
```

```
hist(idade,
     main = "Histograma da idade",
     ylab = "Freqüência",
     xlab = "Idade (em anos completos)")
```



A simulação acima atingiu o objetivo de manter a média em 37 anos, mas o range de dados foi um pouco maior do que eu gostaria, de 11 a 60 anos, mas não comprometerá a simulação.

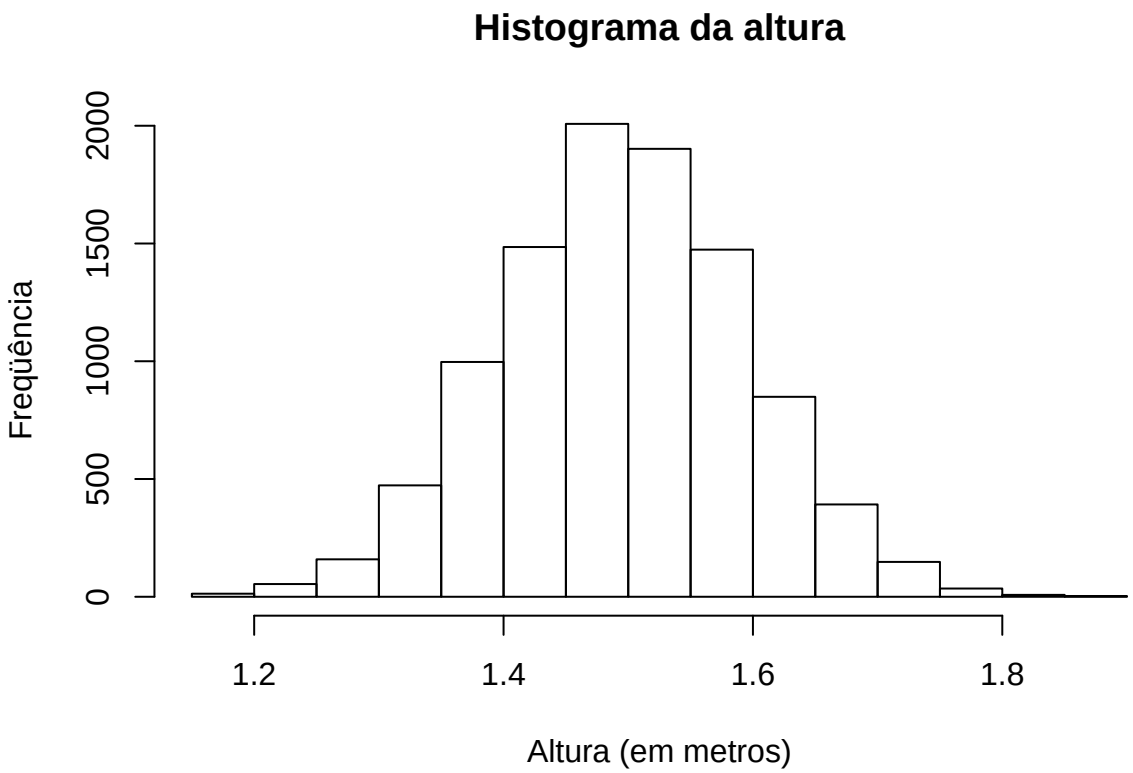
3.1.2. Altura

A altura (em metros) seguiu a mesma lógica da simulação da idade, utilizando-se uma distribuição normal com média 1,50 m e desvio padrão de 0,2 m. Entretanto, como a altura é uma variável dimensional de razão e contínua, utilizei duas casas decimais na simulação:

```
set.seed(1234)
altura <- abs(round(rnorm(n, 1.50, 0.1), 2))
summary(altura)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.160   1.430   1.500   1.501   1.570   1.860
```

```
hist(altura,
     main = "Histograma da altura",
     ylab = "Freqüência",
     xlab = "Altura (em metros)")
```



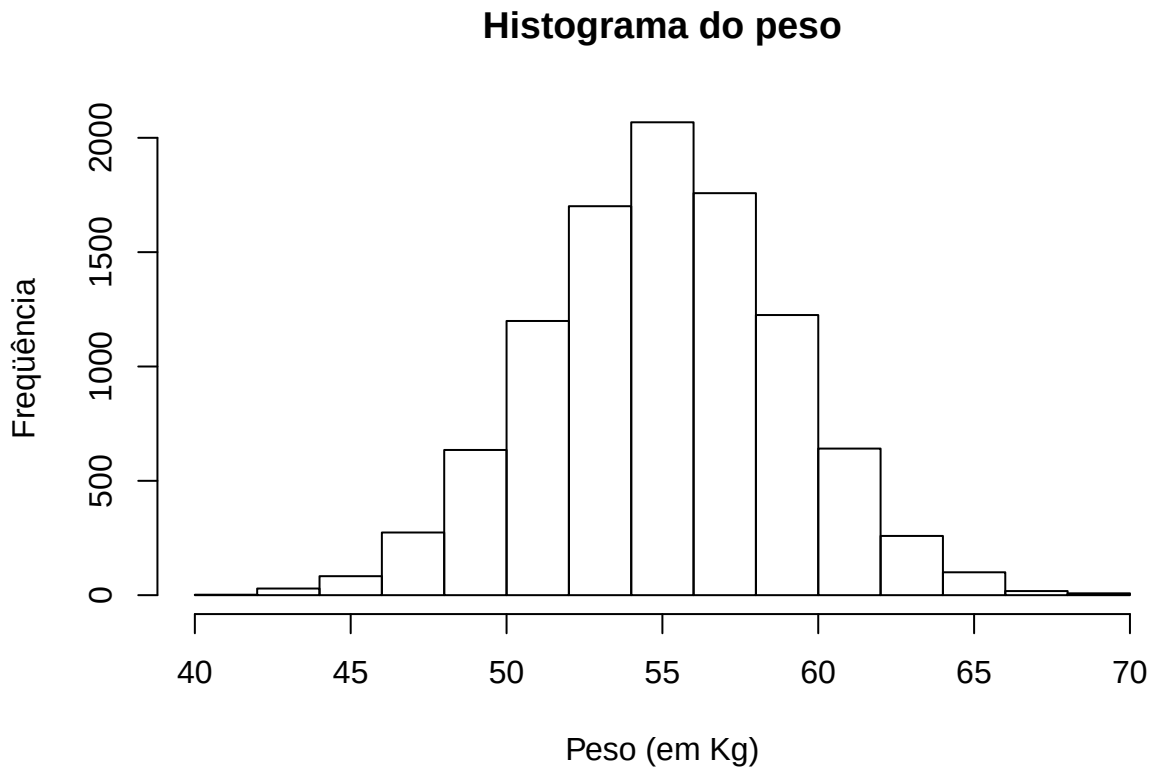
3.1.3. Peso

A variável peso (em Kg) seguiu a mesma lógica de simulação da altura, mas com média de 55 Kg e desvio padrão de 4 Kg (com duas casas decimais):

```
set.seed(1234)
peso <- abs(round(rnorm(n, 55, 4), 2))
summary(peso)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   41.42   52.36   55.02   55.02   57.68   69.47
```

```r
hist(peso,
     main = "Histograma do peso",
     ylab = "Freqüência",
     xlab = "Peso (em Kg)")
```



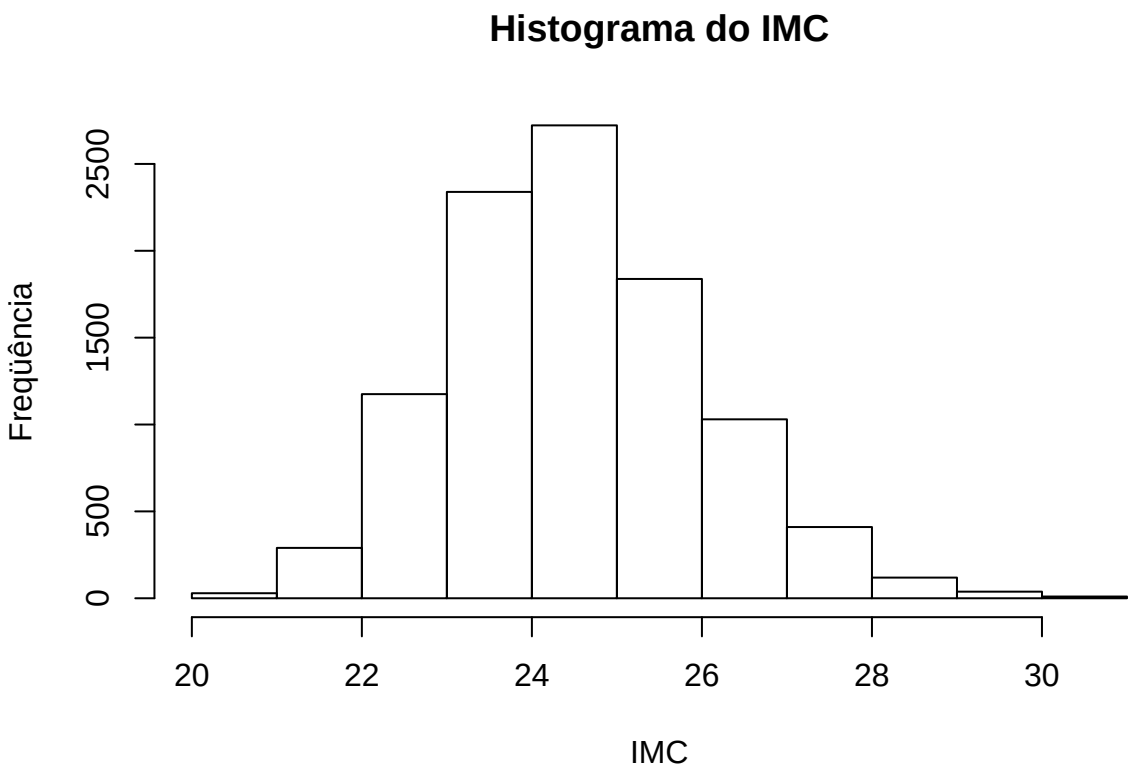
3.1.4. Índice de massa corpórea

O cálculo do índice de massa corpórea (IMC) foi feito utiliando-se os dados já simulados do peso e da altura, utilizando-se a fórmula padrão: peso/(altura)^2 (os dados foram arredondados para 2 casas decimais)

```r
imc <- round(peso/altura^2, 2)
summary(imc)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   20.01   23.49   24.44   24.52   25.46   30.78
```

```r
hist(imc,
     main = "Histograma do IMC",
     ylab = "Freqüência",
     xlab = "IMC")
```



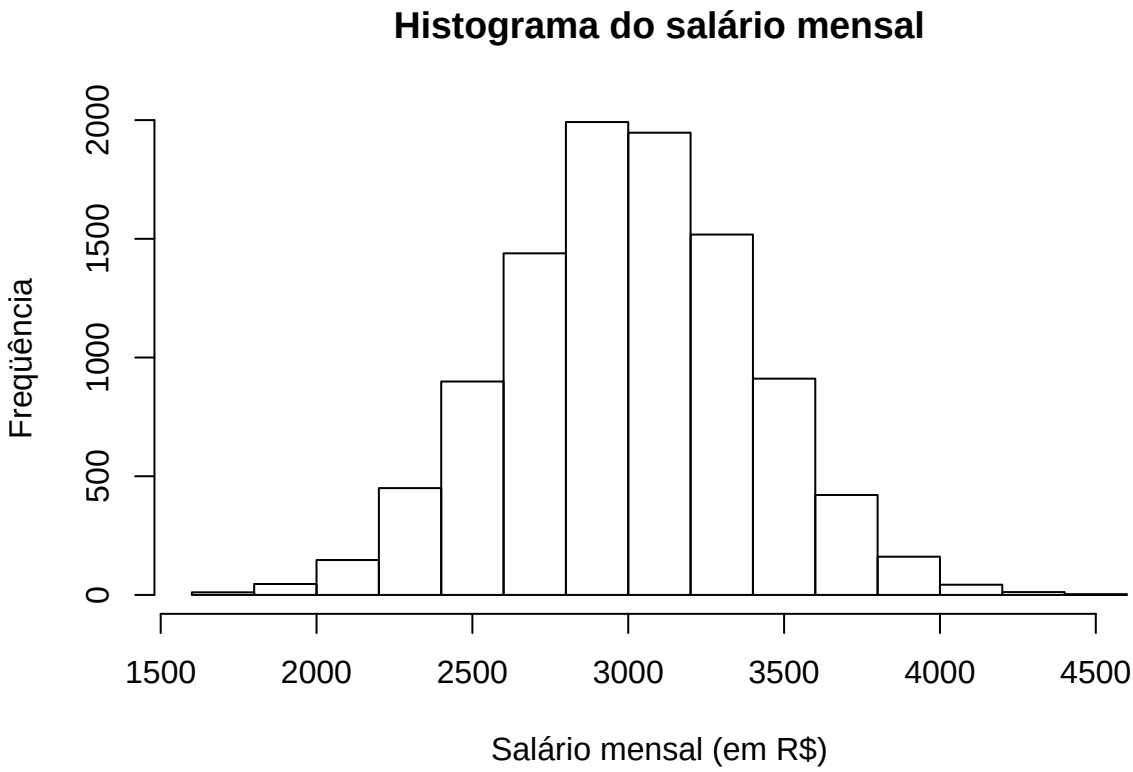
3.1.5. Salário

A variável salário (em reais) também foi simulada através de uma distribuição normal:

```
set.seed(1234)
salario <- abs(round(rnorm(n, 3000, 400), 2))
summary(salario)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1642    2736    3002    3002    3268    4447
```

```
hist(salario,
     main = "Histograma do salário mensal",
     ylab = "Freqüência",
     xlab = "Salário mensal (em R$)")
```



3.1.6. Carros

A variável número de carros foi simulada através de uma amostragem de valores de uma população de números (de 0 a 3). A população de valores foi criada com a função `rep` e a amostra foi retirada com a função `sample`.

```
pop.carros <- rep(c(0,1,2,3), p)
set.seed(1234)
carros <- sample(pop.carros, n)
rm(pop.carros)
summary(carros)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   1.000   1.492   2.000   3.000
```

```
table(carros)
```

```
## carros
##    0    1    2    3
## 2513 2512 2512 2463
```

3.1.7. Filhos

A variável número de filhos foi simulada com a mesma estratégia utilizada para simular o número de carros (uso das funções `rep` e `sample`):

```
pop.filhos <- rep(c(0, 1, 2), p)
set.seed(1234)
filhos <- sample(pop.filhos, n)
```

```r
rm(pop.filhos)
summary(filhos)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   1.000   1.002   2.000   2.000
```

```r
table(filhos)
```

```
## filhos
##    0    1    2
## 3331 3314 3355
```

3.2. Variáveis ordinais e binárias

3.2.1. Escolaridade

A variável escolaridade segue uma escala ordinal, do analfabeto até o pós-doutorado, e será representada no R por um fator ordenado. A estratégia de simulação adotada será o uso da função `rep` para criar uma população de números variando entre 0 a 6 (0 = analfabeto, 6 = pós-doutorado) e da função `sample` para selecionar uma amostra dessa população. Depois será criado um fator com a função `factor`, utiliando os labels adequados para cada level dos fatores:

```r
pop.escolaridade <- rep(c(0, 1, 2, 3, 4, 5, 6), p)
set.seed(1234)
escolaridade.temp <- sample(pop.escolaridade, n)
escolaridade <- factor(escolaridade.temp,
                       levels = c(0, 1, 2, 3, 4, 5, 6),
                       labels = c("Analfabeto", "1º Grau", "2º Grau", "3º Grau",
                                  "Mestrado", "Doutorado", "PósDoc"),
                       ordered = TRUE
                       )
rm(pop.escolaridade, escolaridade.temp)
str(escolaridade)
```

```
##  Ord.factor w/ 7 levels "Analfabeto"<"1º Grau"<..: 3 6 3 5 1 5 4 6 5 4 ...
```

```r
summary(escolaridade)
```

```
## Analfabeto    1º Grau    2º Grau    3º Grau   Mestrado  Doutorado
##       1459       1449       1399       1376       1398       1438
##     PósDoc
##       1481
```

```r
table(escolaridade)
```

```
## escolaridade
## Analfabeto    1º Grau    2º Grau    3º Grau   Mestrado  Doutorado
##       1459       1449       1399       1376       1398       1438
##     PósDoc
##       1481
```

3.2.2. Fumante

A variável fumante (binária) foi simulada através do uso de uma distribuição binomial para obter uma proprção de fumantes de aproximadamente 34% com o uso da função `rbinom`.

Notar que decidi usar um fator ordenado para a variável fumante já que uma variável binária como essa é, na verdade, um subconjunto de uma variável ordinal maior. A variável criada foi chamada de fumante.f (f de fator).

Notar também que mantive no dataset a variável fumante.n, que é a versão numérica da variável.

```
set.seed(1234)
fumante.n <- rbinom(n, 1, .34)
fumante.f <- factor(fumante.n,
                    levels = c(0, 1),
                    labels = c("não", "sim"),
                    ordered = TRUE)
str(fumante.f)
```

```
##  Ord.factor w/ 2 levels "não"<"sim": 1 1 1 1 2 1 1 1 2 1 ...
```

```
summary(fumante.f)
```

```
##  não  sim
## 6605 3395
```

```
str(fumante.n)
```

```
##  int [1:10000] 0 0 0 0 1 0 0 0 1 0 ...
```

```
mean(fumante.n)
```

```
## [1] 0.3395
```

3.3. Variáveis nominais

3.3.1. Sexo

A variável sexo foi simulada com o uso das funções `rep` e `sample` e depois transformada em um fator não ordenado:

```
pop.sexo <- rep(c(1, 2), p)
set.seed(1234)
sexo.temp <- sample(pop.sexo, n)
sexo <- factor(sexo.temp,
               levels = c(1, 2),
               labels = c("M", "F"),
               ordered = FALSE)
rm(pop.sexo, sexo.temp)
str(sexo)
```

```
##  Factor w/ 2 levels "M","F": 1 2 2 1 2 1 2 1 2 2 ...
```

```
summary(sexo)
```

```
##    M    F
## 5068 4932
```

3.3.3. Profissão

A variável profissão foi simulada com a mesma estratégia da variável sexo e depois transformada em um fator não ordenado:

```r
pop.profissao <- rep(0:2, p)
set.seed(1234)
profissao.temp <- sample(pop.profissao, n)
profissao <- factor(profissao.temp,
                    levels = c(0, 1, 2),
                    labels = c("Humanas", "Exatas", "Biológicas"),
                    ordered = FALSE
                    )
rm(pop.profissao, profissao.temp)
str(profissao)
```

```
##  Factor w/ 3 levels "Humanas","Exatas",..: 2 3 1 1 2 3 2 2 2 2 ...
```

```r
summary(profissao)
```

```
##    Humanas     Exatas Biológicas
##       3331       3314       3355
```

4. Criação do dataset

Com todas as variáveis já simuladas, para criar o dataset utilizamos a função `data.frame` para combinar todas as variáveis em um data frame do R. Também incluí aqui uma variável ID para identificar cada observação:

```r
df <- data.frame(id = 1:n,
                 idade,
                 altura,
                 peso,
                 imc,
                 sexo,
                 escolaridade,
                 profissao,
                 fumante.f,
                 fumante.n,
                 salario,
                 carros,
                 filhos
                 )
str(df)
```

```
## 'data.frame':    10000 obs. of  13 variables:
##  $ id          : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ idade       : num  27 37 43 19 38 39 31 31 31 29 ...
##  $ altura      : num  1.38 1.53 1.61 1.27 1.54 1.55 1.44 1.45 1.44 1.41 ...
##  $ peso        : num  50.2 56.1 59.3 45.6 56.7 ...
##  $ imc         : num  26.3 24 22.9 28.3 23.9 ...
##  $ sexo        : Factor w/ 2 levels "M","F": 1 2 2 1 2 1 2 1 2 2 ...
##  $ escolaridade: Ord.factor w/ 7 levels "Analfabeto"<"1º Grau"<..: 3 6 3 5 1 5 4 6 5 4 ...
##  $ profissao   : Factor w/ 3 levels "Humanas","Exatas",..: 2 3 1 1 2 3 2 2 2 2 ...
##  $ fumante.f   : Ord.factor w/ 2 levels "não"<"sim": 1 1 1 1 2 1 1 1 2 1 ...
##  $ fumante.n   : int  0 0 0 0 1 0 0 0 1 0 ...
##  $ salario     : num  2517 3111 3434 2062 3172 ...
##  $ carros      : num  1 3 0 1 2 1 2 2 0 3 ...
##  $ filhos      : num  1 2 0 0 1 2 1 1 1 1 ...
```

Agora que o dataset está criado, vamos salvar em um diretório específico, usando as funções `setwd` e `write.table`:

```
setwd("~/repositoriosGit/ApoemaTraining/abrantesasf/projeto01")
write.table(df, file = "projeto01.csv", sep = ",", col.names = TRUE, fileEncoding = "UTF-8")
```

5. Uso de algumas funções com o dataset

Vamos calcular o sumário de todas as variáveis dimensionais:

```
sapply(df[,c("idade", "altura", "peso", "imc", "salario", "carros", "filhos")],
       summary)
```

```
##         idade altura  peso   imc salario carros filhos
## Min.    11.00  1.160 41.42 20.01    1642  0.000  0.000
## 1st Qu. 30.00  1.430 52.36 23.49    2736  0.000  0.000
## Median  35.00  1.500 55.02 24.44    3002  1.000  1.000
## Mean    35.04  1.501 55.02 24.52    3002  1.492  1.002
## 3rd Qu. 40.00  1.570 57.68 25.46    3268  2.000  2.000
## Max.    60.00  1.860 69.47 30.78    4447  3.000  2.000
```

Agora vamos obter a soma de todas as variáveis dimensionais:

```
apply(df[,c("idade", "altura", "peso", "imc", "salario", "carros", "filhos")],
      2,
      sum)
```

```
##       idade      altura        peso         imc     salario      carros
##   350405.00    15005.75   550244.99   245243.83 30024463.54    14925.00
##      filhos
##    10024.00
```

Tabelas das variáveis ordinais e nominais:

```
sapply(df[,c("sexo", "escolaridade", "profissao", "fumante.f")],
       table)
```

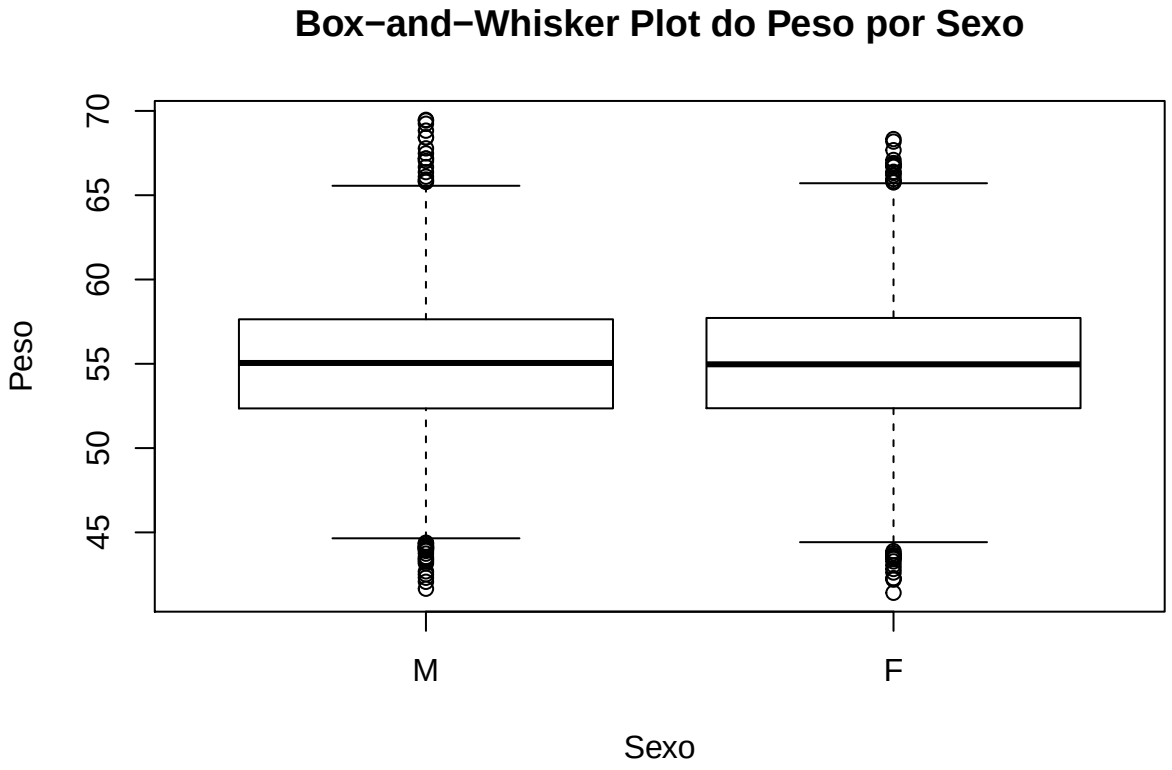
```
## $sexo
##
##    M    F
## 5068 4932
##
## $escolaridade
##
## Analfabeto    1º Grau    2º Grau    3º Grau   Mestrado  Doutorado
##       1459       1449       1399       1376       1398       1438
##     PósDoc
##       1481
##
## $profissao
##
##    Humanas     Exatas Biológicas
##       3331       3314       3355
##
## $fumante.f
##
```

```
##  não  sim
## 6605 3395
```

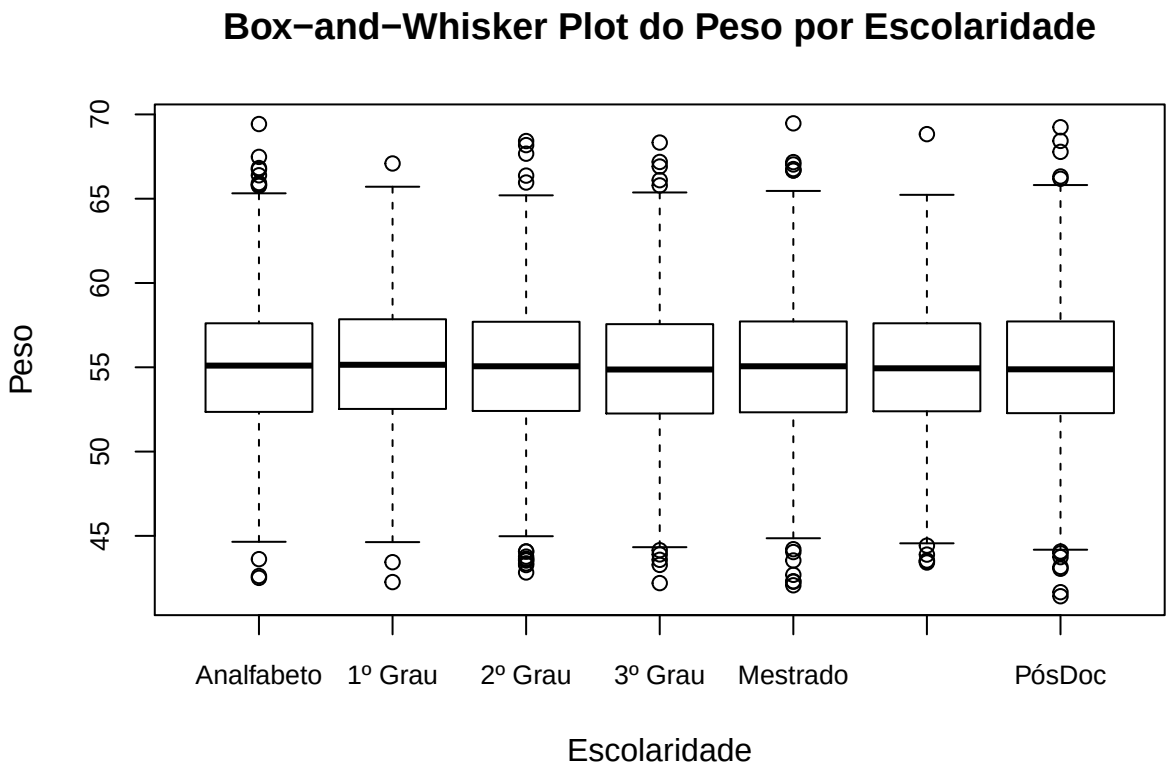
A tabela acima demonstra uma limitação do método de simulação utilizado: a distribuição dos valores em cada variável está muito uniforme, com praticamente a mesma quantidade de observações em cada categoria. Para fins de simulação neste projeto tudo bem mas, para fins de simulações mais compatíveis com a vida real este dataset não seria adequado.

Boxplot de algumas variáveis de acordo com sexo e salário:

```
boxplot(df$peso ~ df$sexo,
        main = "Box-and-Whisker Plot do Peso por Sexo",
        ylab = "Peso",
        xlab = "Sexo")
```

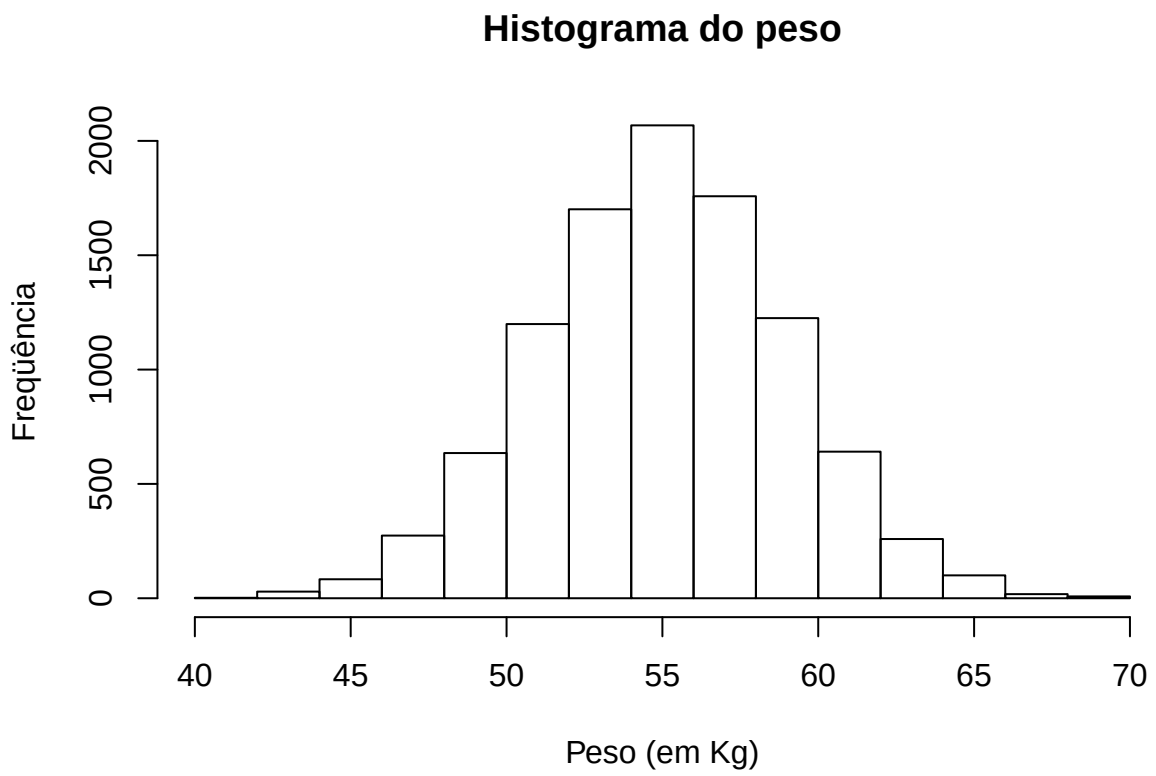



```
boxplot(df$peso ~ df$escolaridade,
        main = "Box-and-Whisker Plot do Peso por Escolaridade",
        ylab = "Peso",
        xlab = "Escolaridade",
        cex.axis = 0.8)
```

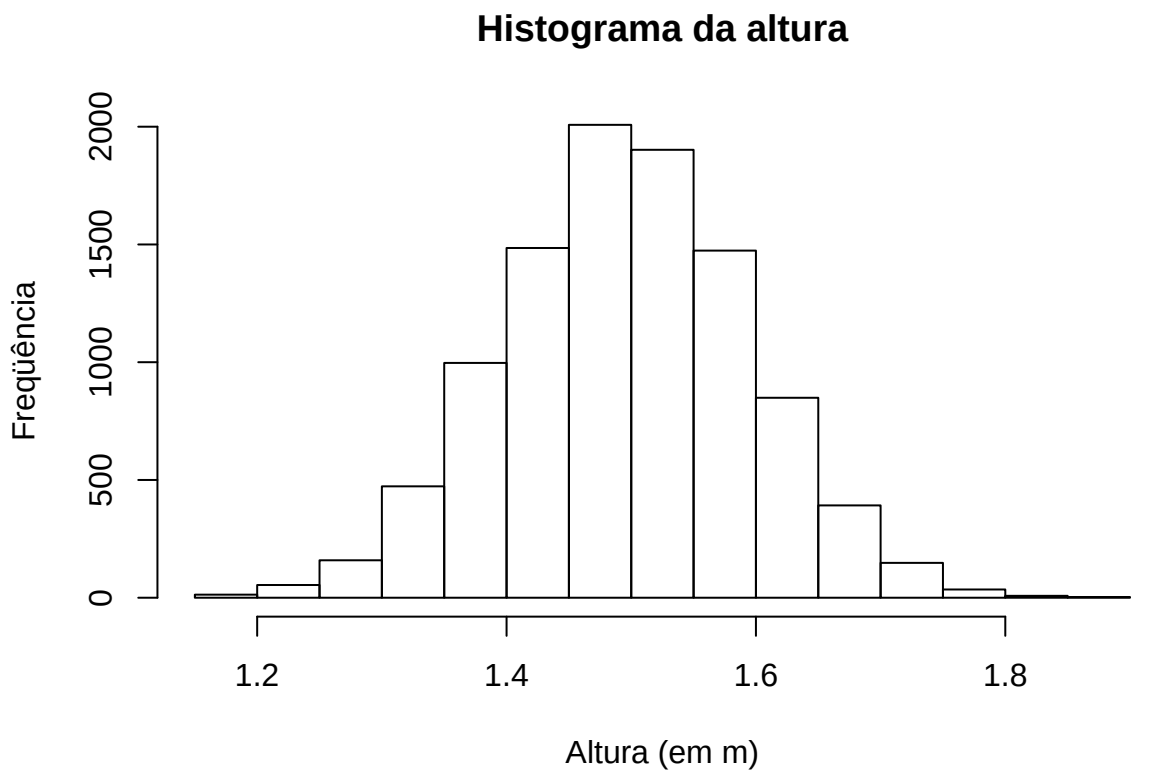


Distribuição do peso e altura (histogramas e scatter plot):

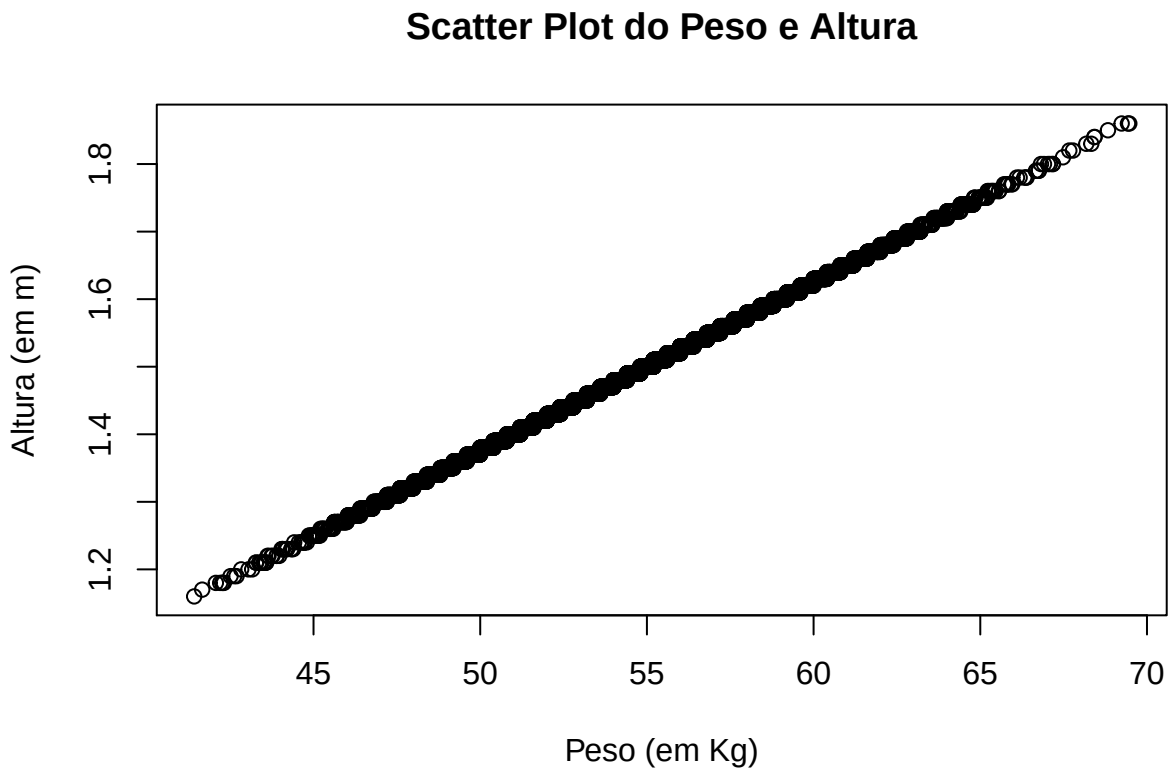
```
hist(df$peso,
     main = "Histograma do peso",
     xlab = "Peso (em Kg)",
     ylab = "Freqüência")
```




```
hist(df$altura,
     main = "Histograma da altura",
     xlab = "Altura (em m)",
     ylab = "Freqüência")
```



```r
plot(df$peso, df$altura,
     main = "Scatter Plot do Peso e Altura",
     xlab = "Peso (em Kg)",
     ylab = "Altura (em m)")
```



Devido ao método de simulação utilizado, a correlação entre o peso e a altura foi praticamente 1:

```r
cor(df$peso, df$altura)
```

```
## [1] 0.9995717
```