## 1. Introdução

### 1.1. Objetivo

Este capítulo tem como objetivo apresentar um projeto de análise de dados com o software R a partir de informações sobre estilo de vida e acidente vascular cerebral (AVC).

### 1.2. Habilidades a serem desenvolvidas

A atividade envolve a coleta de dados secundários, formulação de perguntas de pesquisa/hipóteses, análise exploratória dos dados, com a construção de tabelas e gráficos, aplicação de testes de hipóteses e interpretação dos resultados.

### 1.3. Público-alvo

Por se tratar de um banco de dados da área da saúde, alunos de cursos biomédicos, como Biomedicina, Enfermagem, Medicina, entre outras, poderão se beneficiar do projeto.

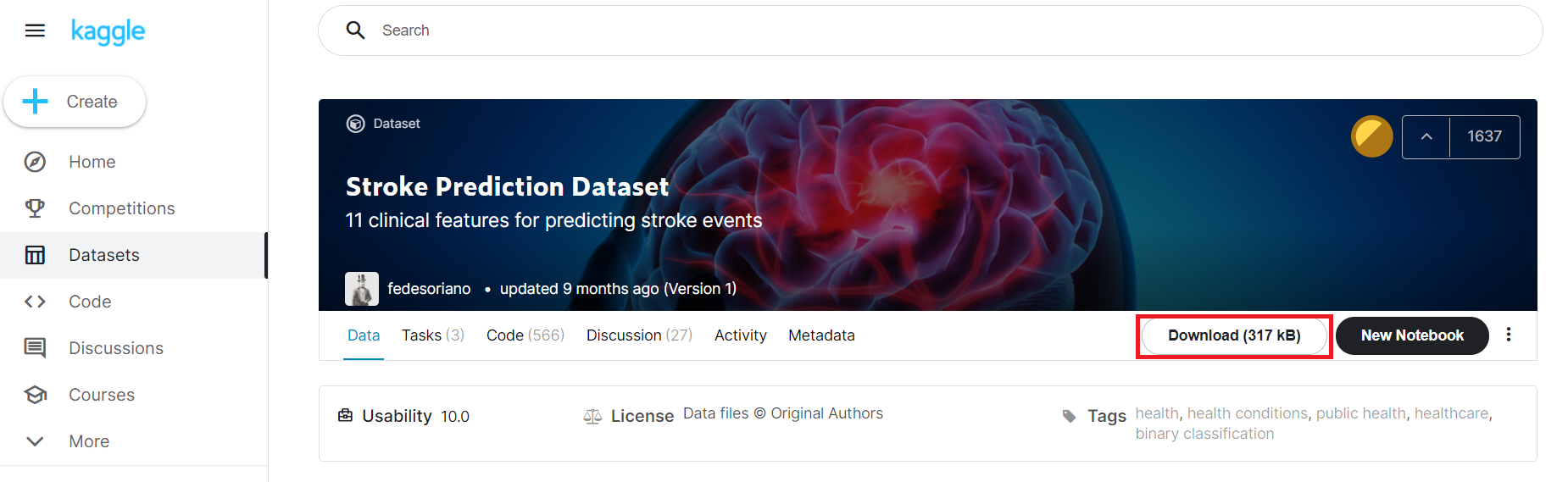
## 2. Etapas do projeto

### 2.1. Coleta dos dados

Os dados utilizados neste projeto são oriundos da plataforma [Kaggle](https://www.kaggle.com/), que fornece diversos conjuntos de dados, cursos e competições em ciência de dados.

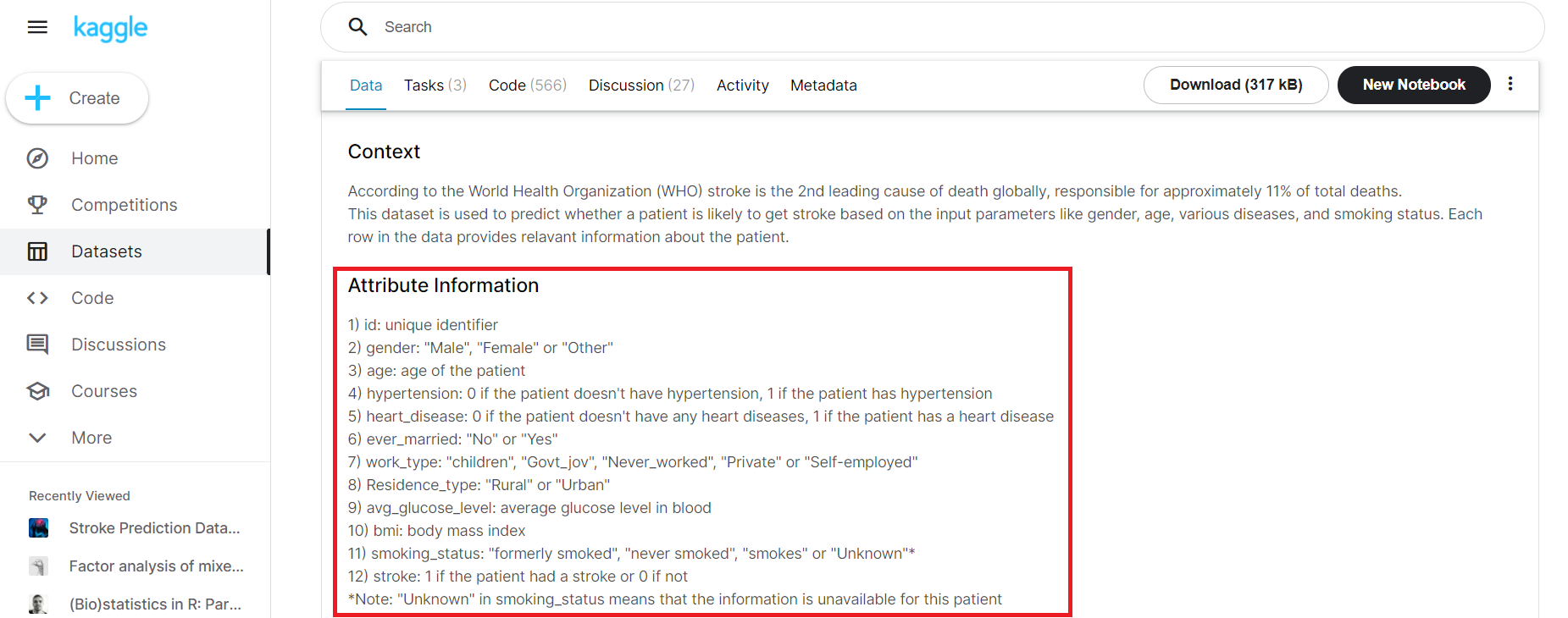
Dica: recomenda-se, nesta etapa, que o docente apresente a plataforma e suas potencialidades. É importante também informar que, para acessar os dados, é preciso estar cadastrado na plataforma.

Para a coleta do banco de dados, acesse o link <https://www.kaggle.com/fedesoriano/stroke-prediction-dataset> e, em seguida, clique no botão Download.



Escolha a pasta de destino e um arquivo .zip será baixado. Extraia os dados na pasta escolhida e o arquivo healthcare-dataset-stroke-data.csv estará disponível.

Ainda na mesma página do Kaggle, as informações sobre as variáveis estão presentes na seção Description, no item Attribute Information.



Traduzindo as informações, temos as seguintes variáveis:

1. **id**: identificador único
2. **gender**: gênero, “Male” (Masculino), “Female” (Feminino) ou “Other” (Outro)
3. **age**: idade do paciente
4. **hypertension**: 0 se o paciente não tem hipertensão, 1 se o paciente tem hipertensão
5. **heart\_disease**: 0 se o paciente não tem doença cardíaca, 1 se o paciente tem doença cardíaca
6. **ever\_married**: “No” (Se nunca casou) or “Yes” (Se já se casou)
7. **work\_type**: tipo de trabalho, “children” (Filhos), “Govt\_jov” (Governo), “Never\_worked” (nunca trabalhou), “Private” (Privado) or “Self-employed” (Autônomo)
8. **Residence\_type**: tipo de residência, “Rural” (Rural) or “Urban” (urbano)
9. **avg\_glucose\_level**: nível médio de glicose no sangue
10. **bmi**: índice de massa corporal
11. **smoking\_status**: “formerly smoked” (Ex-fumante), “never smoked” (Nunca fumou), “smokes” (Fumante) or “Unknown” (Informação não disponível)
12. **stroke**: 1 se o paciente teve um AVC ou 0 se não

### 2.2. Leitura dos dados no R

Após a etapa anterior, o aluno terá o primeiro contato com os dados no software R. Por se tratar de um arquivo .csv, recomenda-se a utilização da função [read.csv](https://www.rdocumentation.org/packages/utils/versions/3.6.2/topics/read.table). Os dados estão separados por vírgula (sep = ",") e a primeira linha traz o nome das variáveis (header = T).

Dica: por se tratar de um projeto extenso, é importante a definição de um diretório de trabalho para que todos os arquivos a serem lidos ou exportados sejam armazenados na mesma pasta. A função para isso é a [setwd](https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/getwd).

# Definindo o diretorio  
setwd("C:\\Users\\Leticia\\Google Drive\\UNIRIO\\Livro DMQ")  
  
# Lendo os dados  
avc <- read.csv("C:\\Users\\Leticia\\Google Drive\\UNIRIO\\Livro DMQ\\healthcare-dataset-stroke-data.csv",  
 header = T,  
 sep = ",")

### 2.3. Pré-processamento dos dados no R

Após a leitura dos dados, é a etapa de pré-processamento. Nesta fase o aluno colocará em prática conhecimentos teóricos, como a classificação de variáveis, e avaliará a qualidade dos dados.

A partir da função [str](https://www.rdocumentation.org/packages/utils/versions/3.6.2/topics/str), é possível ver o tamanho do banco e como as variáveis foram codificadas e estão sendo interpretadas pelo R.

# Vendo a estrutura dos dados  
str(avc)

## 'data.frame': 5110 obs. of 12 variables:  
## $ id : int 9046 51676 31112 60182 1665 56669 53882 10434 27419 60491 ...  
## $ gender : chr "Male" "Female" "Male" "Female" ...  
## $ age : num 67 61 80 49 79 81 74 69 59 78 ...  
## $ hypertension : int 0 0 0 0 1 0 1 0 0 0 ...  
## $ heart\_disease : int 1 0 1 0 0 0 1 0 0 0 ...  
## $ ever\_married : chr "Yes" "Yes" "Yes" "Yes" ...  
## $ work\_type : chr "Private" "Self-employed" "Private" "Private" ...  
## $ Residence\_type : chr "Urban" "Rural" "Rural" "Urban" ...  
## $ avg\_glucose\_level: num 229 202 106 171 174 ...  
## $ bmi : chr "36.6" "N/A" "32.5" "34.4" ...  
## $ smoking\_status : chr "formerly smoked" "never smoked" "never smoked" "smokes" ...  
## $ stroke : int 1 1 1 1 1 1 1 1 1 1 ...

A função mostra que o banco avc possui 5110 observações e 12 colunas. As variáveis foram interpretadas pelo R como:

1. **id**: int - número inteiro
2. **gender**: chr - caracter (cada elemento do vetor é uma sequência de um ou mais caracteres)
3. **age**: num - numérico (real ou decimal)
4. **hypertension**: int - número inteiro
5. **heart\_disease**: int - número inteiro
6. **ever\_married**: chr - caracter
7. **work\_type**: chr - caracter
8. **Residence\_type**: chr - caracter
9. **avg\_glucose\_level**: num - numérico (real ou decimal)
10. **bmi**: chr - caracter
11. **smoking\_status**: chr - caracter
12. **stroke**: int - número inteiro

Nesta etapa, o aluno deve-se questionar quais variáveis foram codificadas de forma errada e se alguma ou mais colunas podem ser eliminadas por não fazerem sentido ao projeto ou por fazerem apenas o papel de identificador das linhas. Neste caso, a coluna id desempenha este papel. Para isso, o aluno pode atribui-la como nome das linhas e, em seguida, exclui-la do banco de dados. A função rownames(<https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/row+colnames>) pode ser usada para este objetivo.

# Atribuindo a coluna id como nome das linhas  
rownames(avc) <- avc$id  
  
# Excluindo a coluna id  
avc$id <- NULL

Em seguida, o aluno deverá classificar as variáveis e identificar quais delas não foram interpretadas corretamente pelo R. A partir do conhecimento teórico adquirido, ele saberá que uma codificação errada desencadeará em erros consecutivos na análise de dados.

Assim, espera-se as seguintes classificações:

1. **gender**: variável qualitativa nominal binária
2. **age**: variável quantitativa razão contínua
3. **hypertension**: variável qualitativa nominal binária
4. **heart\_disease**: variável qualitativa nominal binária
5. **ever\_married**: variável qualitativa nominal binária
6. **work\_type**: variável qualitativa nominal policotômica
7. **Residence\_type**: variável qualitativa nominal binária
8. **avg\_glucose\_level**: num - numérico (real ou decimal)
9. **bmi**: variável quantitativa razão contínua
10. **smoking\_status**: variável qualitativa nominal policotômica
11. **stroke**: variável qualitativa nominal binária

Ao comparar com o resultado fornecido pelo R, vemos que as variáveis **hypertension**, **heart\_disease**, **bmi** e **stroke** foram codificadas de forma errada.

Por terem sido representadas pelos números 0 e 1, o R interpretou as variáveis **hypertension**, **heart\_disease**, e **stroke** como quantitativas. Logo, devemos transformá-las para qualitativa. A função a ser usada poderá ser a [as.factor](https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/factor). A codificação fator é a melhor opção para representar uma variável qualitativa e, por isso, também transformaremos as variáveis **gender**, **ever\_married**, **work\_type**, **Residence\_type** e **smoking\_status** de chr para factor.

Para facilitar a tarefa, a função [mutate\_all](https://www.rdocumentation.org/packages/dplyr/versions/0.7.8/topics/mutate) do pacote dplyr pode ser utilizada, pois ela transforma várias variáveis ao mesmo tempo. A função str pode ser novamente usada para ver as modificações.

# Transformando para factor  
library(dplyr)  
avc[,c(1,3:7,10,11)] <- avc[,c(1,3:7,10,11)] %>% mutate\_all(as.factor)  
  
# Vendo a estrutura dos dados  
str(avc)

## 'data.frame': 5110 obs. of 11 variables:  
## $ gender : Factor w/ 3 levels "Female","Male",..: 2 1 2 1 1 2 2 1 1 1 ...  
## $ age : num 67 61 80 49 79 81 74 69 59 78 ...  
## $ hypertension : Factor w/ 2 levels "0","1": 1 1 1 1 2 1 2 1 1 1 ...  
## $ heart\_disease : Factor w/ 2 levels "0","1": 2 1 2 1 1 1 2 1 1 1 ...  
## $ ever\_married : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 1 2 2 ...  
## $ work\_type : Factor w/ 5 levels "children","Govt\_job",..: 4 5 4 4 5 4 4 4 4 4 ...  
## $ Residence\_type : Factor w/ 2 levels "Rural","Urban": 2 1 1 2 1 2 1 2 1 2 ...  
## $ avg\_glucose\_level: num 229 202 106 171 174 ...  
## $ bmi : chr "36.6" "N/A" "32.5" "34.4" ...  
## $ smoking\_status : Factor w/ 4 levels "formerly smoked",..: 1 2 2 3 2 1 2 2 4 4 ...  
## $ stroke : Factor w/ 2 levels "0","1": 2 2 2 2 2 2 2 2 2 2 ...

Dica: para que as tabelas e gráficos a serem preparados posteriormente sejam autoinformativos, sugere-se a modificação das categorias das variáveis qualitativas para nomes mais compreensíveis. Essa etapa pode ser realizada pela função [levels](https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/levels). As mudanças devem seguir a ordem que o R apresenta as categorias. Por isso, recomenda-se rodar primeiramente a função levels para ver a ordem dos níveis antes de realizar a substituição.

# Mudando os nomes das categorias  
levels(avc$gender) <- c("Feminino", "Masculino", "Outro")  
levels(avc$hypertension) <- c("Não", "Sim")  
levels(avc$heart\_disease) <- c("Não", "Sim")  
levels(avc$ever\_married) <- c("Não", "Sim")  
levels(avc$work\_type) <- c("Filhos", "Governo", "Nunca trabalhou", "Privado", "Autônomo")  
levels(avc$Residence\_type) <- c("Rural", "Urbano")  
levels(avc$smoking\_status) <- c("Ex-fumante", "Nunca fumou", "Fuma", NA) # a informacao nao disponivel foi codificada como NA  
levels(avc$stroke) <- c("Não", "Sim")

Após lidar com as variáveis qualitativas, o aluno precisa ainda corrigir a variável **bmi**. Apesar de ser numéria, o R codificou-a como caracter. Isso se deve à presença de uma palavra entre os valores. Quando isso acontece, o R entende que aquela coluna contém caracteres e assim a codifica. Mas este não era o caso. Para verificar, vamos ver as 20 primeiras linhas da variável com a função [head](https://www.rdocumentation.org/packages/utils/versions/3.6.2/topics/head).

head(avc$bmi, 20)

## [1] "36.6" "N/A" "32.5" "34.4" "24" "29" "27.4" "22.8" "N/A" "24.2"  
## [11] "29.7" "36.8" "27.3" "N/A" "28.2" "30.9" "37.5" "25.8" "37.8" "N/A"

Podemos observar que por conta do “N/A”, o R interpretou a variável como caracter. Nesta etapa, o aluno deve então ser capaz de transformar esse dado ausente em NA e, em seguida, transformar a variável em numérica com a função [as.numeric](https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/numeric).

# Transformando os "N/A" da variavel bmi em NA  
avc$bmi[avc$bmi == "N/A"] <- NA  
  
# Transformando a variavel bmi em numerica  
avc$bmi <- as.numeric(avc$bmi)

Após o pré-processamento de dados, o aluno deverá ser capaz de chegar na seguinte estrutura:

str(avc)

## 'data.frame': 5110 obs. of 11 variables:  
## $ gender : Factor w/ 3 levels "Feminino","Masculino",..: 2 1 2 1 1 2 2 1 1 1 ...  
## $ age : num 67 61 80 49 79 81 74 69 59 78 ...  
## $ hypertension : Factor w/ 2 levels "Não","Sim": 1 1 1 1 2 1 2 1 1 1 ...  
## $ heart\_disease : Factor w/ 2 levels "Não","Sim": 2 1 2 1 1 1 2 1 1 1 ...  
## $ ever\_married : Factor w/ 2 levels "Não","Sim": 2 2 2 2 2 2 2 1 2 2 ...  
## $ work\_type : Factor w/ 5 levels "Filhos","Governo",..: 4 5 4 4 5 4 4 4 4 4 ...  
## $ Residence\_type : Factor w/ 2 levels "Rural","Urbano": 2 1 1 2 1 2 1 2 1 2 ...  
## $ avg\_glucose\_level: num 229 202 106 171 174 ...  
## $ bmi : num 36.6 NA 32.5 34.4 24 29 27.4 22.8 NA 24.2 ...  
## $ smoking\_status : Factor w/ 3 levels "Ex-fumante","Nunca fumou",..: 1 2 2 3 2 1 2 2 NA NA ...  
## $ stroke : Factor w/ 2 levels "Não","Sim": 2 2 2 2 2 2 2 2 2 2 ...