# Introduction to Data Cleaning

Dificilmente no dia a dia vamos receber dados prontos para trabalhar, sendo assim é importante saber trata-los para a análise e veremos algumas dicas aqui.

## Atividades na etapa de Data Cleaning

- Remover dados desnecessários para a análise em questão
- Remover dados duplicados
- Resolver inconsistências
- Lidar com dados nulos e *outliers*
- Criar novas variáveis quando necessário
- Combinar diferentes bases de dados

## Como saber se a base está pronta para análise?

O ideal é fazer algumas perguntas

1. Que dados precisamos para a análise a ser feita?
2. Precisamos de novas variáveis?
3. Os dados estão com o tipo correto?
4. Precisamos juntar diferentes bases?
5. Precisamos reestruturar os dados?
6. Existem dados nulos?

## Operações comuns durante o Data Cleaning

- Nas linhas
  - Filtrar
  - Agrupar as linhas
  - Remover linhas duplicadas

- Nas colunas
  - Selecionar colunas específicas
  - Criar novas colunas
  - Renomear colunas
  - Ajustar tipo de dados
- Valores
  - Resolver inconsistências
  - Combinar ou separar valores
  - Imputar valores no lugar de missings
- Dataframes
  - Combinar linhas
  - Combinar colunas
  - Reestruturar (transformar linhas em colunas vice-versa)

# Alguns Exemplos de código

## Alterar Tipo de Dados

### Transformar campo em numérico

```
base$coluna <- as.numeric(base$coluna)
```

### Transformar campo em texto

```
base$coluna <- as.character(base$coluna)
```

### Transformar várias colunas ao mesmo tempo de acordo com a sua nomenclatura ou índice

```
base <- base %>% mutate(across(contains("percent_"),as.numeric))
```

```
base <- base %>% mutate(across(3:5, as.numeric))
```

## Criar novas colunas

### Com operações matemáticas

```
base <- base %>% mutate(total = valor1 + valor2)
```

### Com expressões lógicas

```
base <- base %>% mutate(flag = ifelse(valor1 > 100, 1, 0))
```

```
base <- base %>% mutate(flag = case_when(valor1 > 100 ~ 1, valor1 < 100
~ 2, TRUE ~ 0))
```

## Filtrar linhas

### Utilizando operadores lógicos

```
base <- base %>% filter(flag == 1)
```

```
base <- base %>% filter(regiao != "SUL")
```

### Utilizando funções que têm retorno binário

```
base <- base %>% filter(!is.na(id))
```

## Agrupar linhas

```
base <- base %>% group_by(registro) %>% summarise(Qtd = n())
```

## Selecionar linhas

### Escolher colunas a manter

```
base <- base %>% select(coluna, `coluna com espaço`, coluna2, contains(valor))
```

### Escolher colunas a retirar

```
base <- base %>% select(-coluna)
```

## Renomear colunas

### Fazendo um de-para

```
base %>% rename(novo_nome = nome_antigo)
```

### Escolhendo a posição que deseja alterar

```
names(base)[0] <- "Novo_Nome"
```