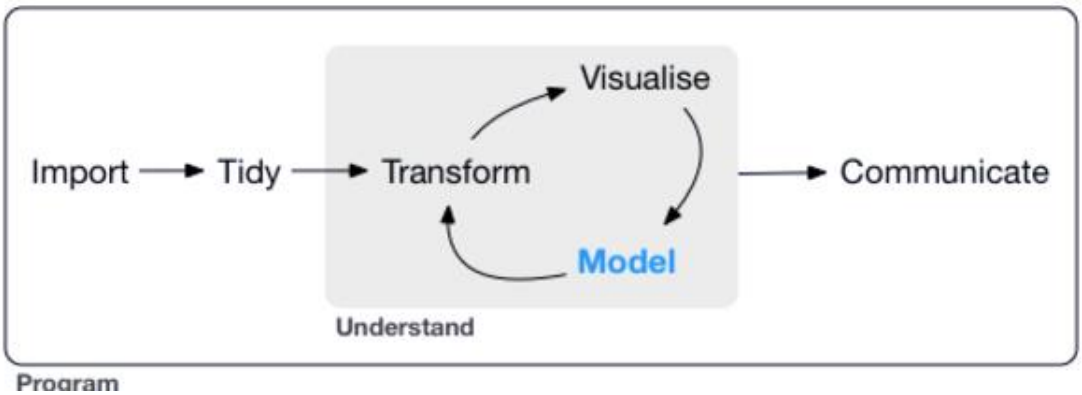
Tidyverse

Material desenvolvido por Dr. Clarice Braúna Mendes

30-04-2024

O Tidyverse é uma coleção de pacotes para a manipulação, exploração e visualização de dados. Foi desenvolvido por Hadley Wickham, mas está em constante evolução! Os pacotes do Tidyverse foram desenhados especificamente para aumentar a produtividade de estatísticos e cientistas de dados, guiando-os através de fluxos de trabalho que facilitam a comunicação e reprodutibilidade das análises.

A sintaxe do Tidyverse é feita de forma coerente entre seus diferentes pacotes. Embora possa causar um pouco de estranhamento no início, ela será destrinchada passo a passo para que você possa compreendê-la e usá-la em seu dia a dia na pesquisa! Essa sintaxe foi desenvolvida especialmente para que o seguinte fluxo de trabalho fosse facilitado:

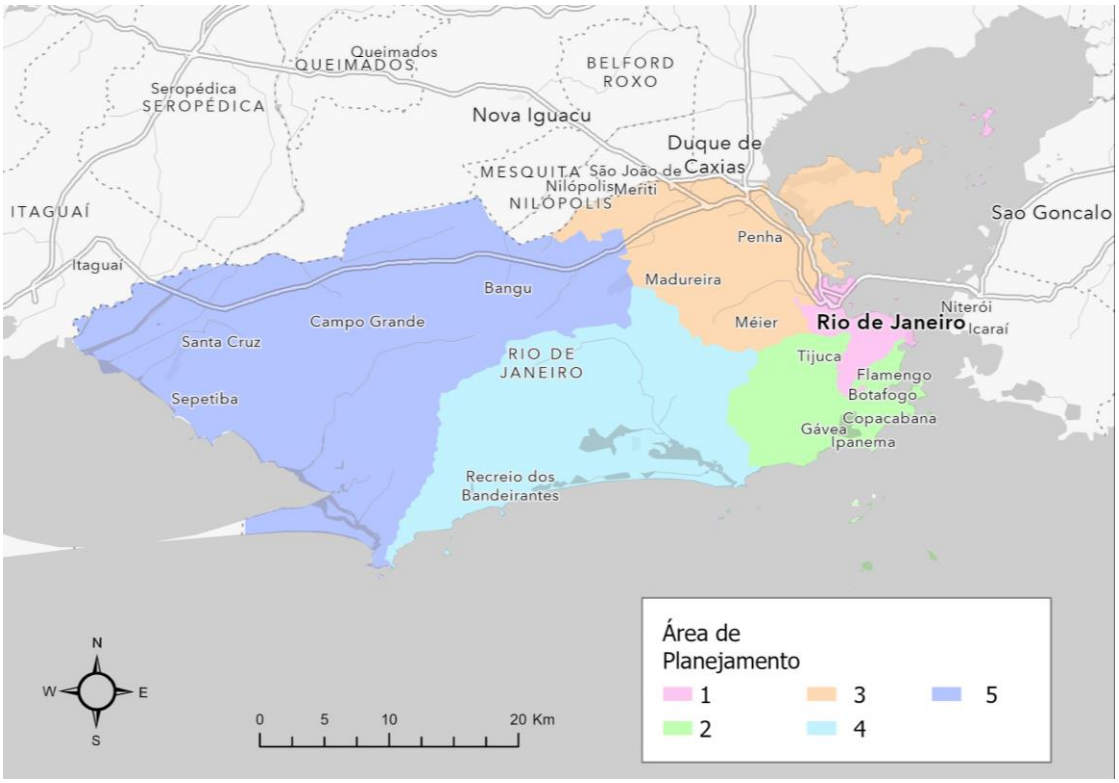


O primeiro passo deste fluxo é a importação de dados para o ambiente do R, que pode se dar por pacotes como o **readr**, **readxl**, entre outros. Você já executou essa etapa em aulas anteriores! Logo em seguida, vem a etapa de organização e limpeza dos seus dados (*tidying,* em inglês significa arrumar, limpar). Ela é feita pelo pacote **tidyr**, destinado à filtragem de dados em nossas tabelas que não fazem sentido (os famosos NAs!), ou para a reorganização de colunas e linhas nos *dataframes*. Começamos então a transformar e relacionar os nossos dados de acordo com a pergunta que queremos responder: pode ser pelo cálculo de estatísticas descritivas (média, contagem) ou pela redução de variáveis a partir de várias medidas iniciais (nº espécies em uma mancha de habitat, contaminação por metais pesados em um lago, índice de Shannon entre outros). Isso pode ser feito pelo pacote **dplyr**.

As etapas de organização, limpeza e transformação dos dados são comumente chamadas em conjunto de *data wragling* - ou "brigando com os dados", já que tornar os dados mais palatáveis para análises e modelagens pode ser tão duro quanto uma briga de bar (adaptação livre das palavras do Hadley Wickham). Finalmente, é necessário modelar e criar visualizações para comunicar nossas descobertas ao público. O **ggplot** fornece uma gama incrível de ferramentas com as quais podemos criar gráficos e visualizações para os nossos dados! Será dado um enfoque especial em nossas aulas para esse último pacote.

Missão do dia

Imagine que você é cientista de dados ambientais, e que a prefeitura do Rio de Janeiro pediu para você fazer um estudo de priorização de criação de Áreas de Proteção (APs) na cidade. A prefeitura quer saber em qual Área de Planejamento (abaixo) é mais estratégico alocar esforços para a criação de novas Áreas de Proteção.



Para isso, você tem em suas mãos dados de uso do solo e de cobertura vegetal (Classes de Uso do solo e Cobertura Vegetal - RJ.xlsx), além de dados referentes às Áreas de Proteção do Rio de Janeiro (Areas_Protegidas_Rio.xlsx), provenientes do data.rio. Você vai utilizar o fluxo de trabalho do Tidyverse para tentar resolver esse problema! É comum propor novas Áreas de Proteção em locais com poucos parques e unidades de conservação presentes, e/ou em locais em que ainda haja grande cobertura florestal ou de habitat para ser protegida.

Carregando pacotes

Uma das vantagens do Tidyverse é que, assim que você o instala, não é necessário carregar seus pacotes individualmente. Basta aplicar **library(tidyverse)** que todas as funções de todos os pacotes podem ser então utilizadas! O pacote **readxl**, entretanto, deve ser carregado separadamente.

```
library(readxl)
library(tidyverse)
```

Importando os dados

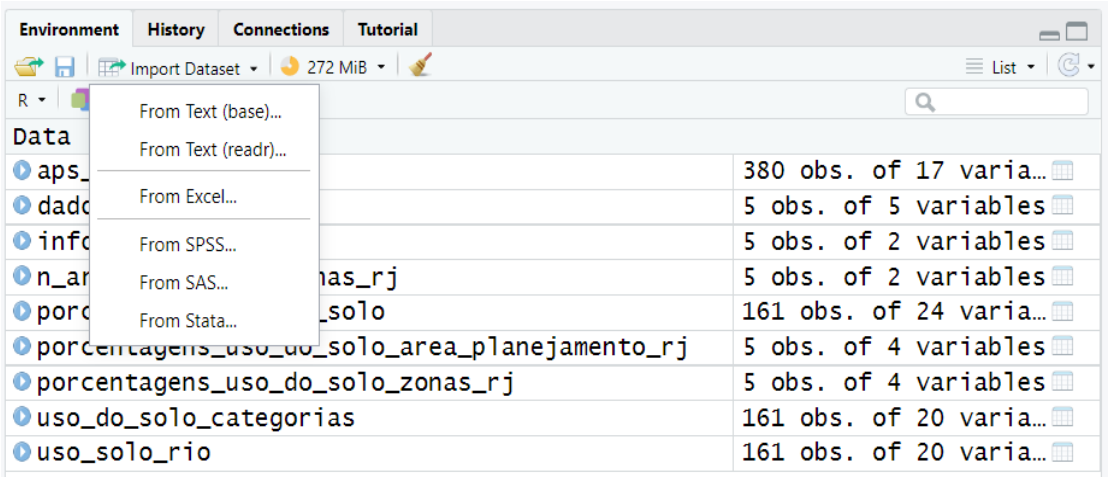
Antes de importar os dados, abra as tabelas no Excel. Dê uma olhada inicial nas suas colunas e linhas, e tente interpretar o que cada uma delas pode informar a você. Repare que a tabela de uso do solo parece ser mais organizada e intuitiva que a das Áreas de Proteção, e que possui duas abas - a que temos interesse de importar é de "Dados".

```r
#dados do uso do solo da cidade do Rio de Janeiro
uso_solo_rio<-read_excel("Classes de Uso do solo e Cobertura Vegetal - RJ.xlsx",sheet = "Dados")

#dados das áreas protegidas da cidade do Rio de Janeiro
aps_rio<-read_excel("Areas_Protegidas_Rio.xlsx")
```

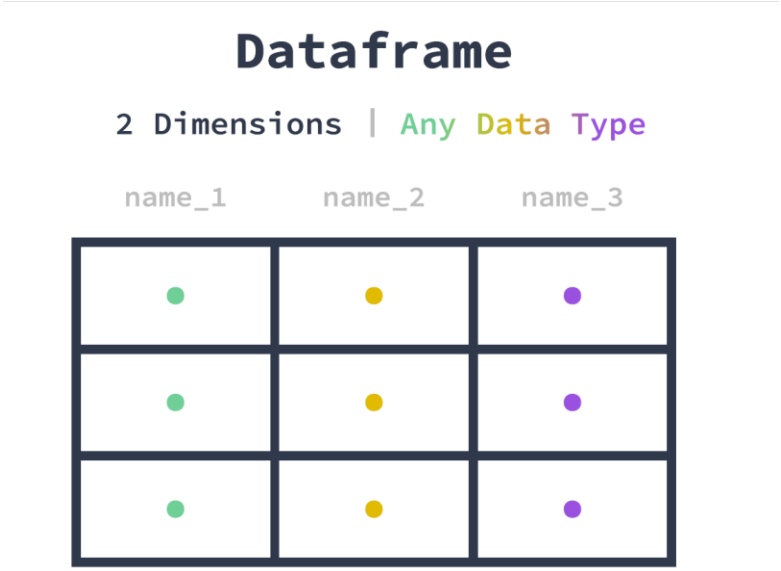
Na tabela de uso do solo, cada linha corresponde individualmente a um bairro do Rio de Janeiro, enquanto que na tabela de APs um bairro pode corresponder a mais de uma linha, já que o mesmo bairro pode ter mais de uma Área de Proteção.

Lembre-se que também é possível importar dados diretamente pelo console do R, na área "Environment". Em nosso caso, iríamos assinalar "From Excel" (abaixo), por se tratar de um arquivo xlsx.



Exploração inicial dos dados

É bom lembrar que os *dataframes* são compreendidos pelo R como estruturas bidimensionais, as quais armazenam valores de quaisquer tipos de dado ou objeto. Ou seja, cada coluna pode armazenar dados/objetos do tipo caractere, numérico, entre outros (abaixo).



Esquema fornecido pelo Dataquest

Uma forma de verificar os tipos de dados do seu *dataframe* é por meio da função **glimpse**, do pacote **tibble** do Tidyverse.

```
glimpse(uso_solo_rio)
glimpse(aps_rio)
```

Dados de uso do solo no Rio de Janeiro

Organizando os dados de acordo com o uso

Na aba "Definições das classes" do arquivo Excel de dados de uso do solo, vemos que a prefeitura do Rio classificou os diferentes usos do solo em três grandes grupos: "Áreas naturais", "Antropismos" e "Corpos d'água continentais". Esses grupos não estão disponíveis na aba "Dados", e o primeiro passo é organizar nossas colunas de uso do solo para serem compatíveis com essa classificação. Podemos fazer isso com a função **relocate**.

```
uso_do_solo_categorias<-relocate(uso_solo_rio,Reflorestamento,.after =
`Afloramento Rochoso`)
```

Com essa nova organização, é possível ver que os usos do solo compatíveis com os três grandes grupos agora estão próximos uns dos outros. Suponha, agora, que logo após ter feito isso, você gostaria de realocar a primeira coluna para o final do *dataframe*. Sua sequência de códigos ficaria mais ou menos assim:

```
#realocando apenas a classe "Reflorestamento", como feito anteriormente
uso_do_solo_categorias<-relocate(uso_solo_rio,Reflorestamento,.after =
`Afloramento Rochoso`)

#realocando primeira coluna para o final do dataframe
```

```
uso_do_solo_categorias_realocado<-
relocate(uso_do_solo_categorias,`Bairros Rio de Janeiro`,.after =
`Praia`)
```

Perceba que sempre criar um objeto novo (uso_do_solo_categorias, uso_do_solo_categorias_realocado e assim por diante) pode ser extremamente exaustivo e confuso quando você precisa fazer uma sequência de operações em um único *dataframe*. Por isso, o Tidyverse criou o operador %>%, também conhecido como *pipeline*, traduzido do inglês como "canalizar". Ou seja, o operador %>% age como um encanamento, conectando as diferentes funções, sequencialmente, para modificar seu *dataframe* original. Daqui por diante, iremos aplicar os *pipelines* aos nossos códigos dentro do escopo do Tidyverse. Veja como o código anterior se apresenta de acordo com essa nova sintaxe:

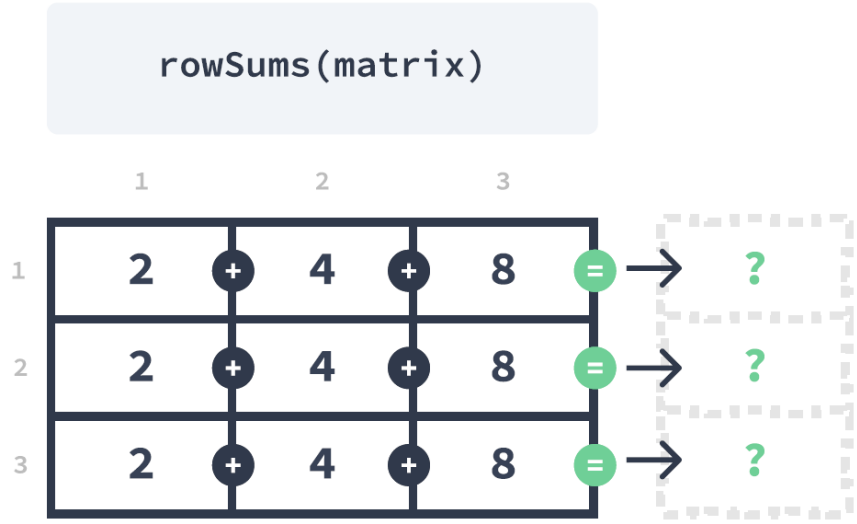
```
uso_do_solo_categorias_realocado<-uso_solo_rio%>%
  relocate(Reflorestamento,.after = `Afloramento Rochoso`)%>%
  relocate(`Bairros Rio de Janeiro`,.after = `Praia`)
```

Transformando os dados de uso do solo

Agora, vamos criar três novas colunas no nosso *dataframe*, cada uma delas sendo a soma das classes de uso do solo compatíveis aos três grandes grupos mencionados anteriormente. Também criaremos uma coluna que será a soma de todos os tipos de uso do solo. Iremos fazer isso atráves da função **mutate**. Essa função permite não apenas criar uma nova coluna, mas também modificar uma coluna pré-existente.

```
estimativa_uso_do_solo<-uso_do_solo_categorias%>%
  mutate(Vegetacao_natural = rowSums(pick(`Floresta Ombrófila
Densa`:Brejo)),.before = `Floresta Ombrófila Densa`)%>%
  mutate(Antropismos =  rowSums(pick(`Área
Urbana`:Reflorestamento)),.before = `Área Urbana`)%>%
  mutate(Corpos_dagua_continental = rowSums(pick(`Corpo d'água
continental`:Praia)),.before = `Corpo d'água continental`)%>%
  mutate(Uso_do_solo =
rowSums(pick(Vegetacao_natural,Antropismos,Corpos_dagua_continental)),.be
fore = Vegetacao_natural)
```

Lembre-se que, para somar os valores de colunas distintas, devemos aplicar a função **rowSums**, que apresenta a seguinte lógica:



Esquema fornecido pelo Dataquest

Agora que temos a área total de todos os usos do solo (coluna "Uso_do_solo"), assim como de cada grande grupo (colunas "Vegetacao_natural", "Antropismos" e "Corpos_dagua_continental") em cada bairro do Rio de Janeiro, queremos saber qual a porcentagem de uso que cada um desses grandes grupos em cada Área de Planejamento da cidade. Por exemplo, será que Áreas Antropizadas são mais comuns na Zona Sul (Área de Planejamento 2) ou na Zona Norte (Área de Planejamento 3)? Mas primeiro, vamos "limpar" nosso *dataframe* e tentar encontrar alguma inconsistência em seus dados por meio das funções **filter** e **select**.

```
uso_do_solo_area_planejamento_rj_tidy<-estimativa_uso_do_solo%>%
  filter(`Bairros Rio de Janeiro` != "Lapa")%>%
  select(`Área de Planejamento
(AP)`,Uso_do_solo,Vegetacao_natural,Antropismos,Corpos_dagua_continental)
```

Veja que temos agora um *dataframe* com cinco colunas. Para descobrirmos as porcentagens dos grandes grupos, primeiro precisamos dizer que iremos agrupar nossos dados de acordo com cada Área de Planejamento (função **group_by**) e que, em sequência, iremos somar cada uma de suas linhas (que correspondiam, originalmente, aos bairros que compunham aquela Área de Planejamento) por meio da função **summarise**.

```
soma_uso_do_solo_area_planejamento_rj<-
uso_do_solo_area_planejamento_rj_tidy%>%
  group_by(`Área de Planejamento (AP)`)%>%
  summarise(across(everything(),list(sum)))
```

Finalmente, temos a relação da área em hectares de cada uso do solo em grandes grupos, assim como o uso total do solo, em cada Área de Planejamento.

Iremos calcular e arredondar as porcentagens de cada grande grupo em relação ao uso total do solo com a função **mutate** e selecionar as colunas que nos interessam.

```
porcentagens_uso_do_solo_area_planejamento_rj<-
soma_uso_do_solo_area_planejamento_rj%>%
  mutate(Vegetacao_natural_Porcentagem =
(Vegetacao_natural_1/Uso_do_solo_1)*100)%>%
  mutate(Antropismos_Porcentagem = (Antropismos_1/Uso_do_solo_1)*100)%>%
  mutate(Corpos_dagua_Porcentagem =
(Corpos_dagua_continental_1/Uso_do_solo_1)*100)%>%
  mutate(across(where(is.numeric), round, 2))%>%
  select(`Área de Planejamento
(AP)`,Vegetacao_natural_Porcentagem:Corpos_dagua_Porcentagem)
```

Organização e transformação dos dados de Áreas Protegidas do Rio de Janeiro

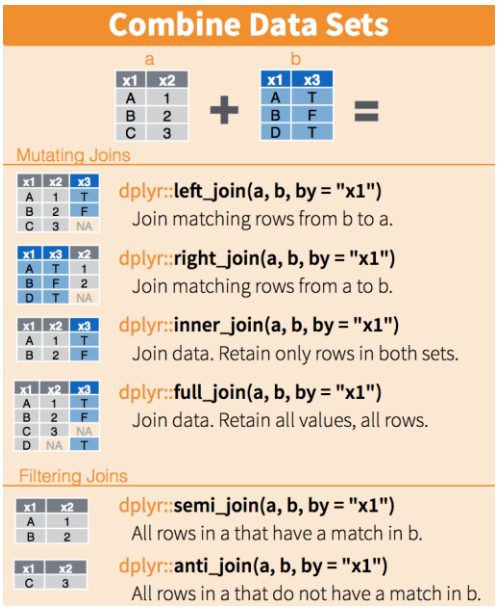
Em uma inspeção inicial, você deve ter percebido que o *dataframe* de dados de Áreas de Proteção possui colunas denominadas "nome", "area_plane" e "nome_1". Essas colunas possuem informações referentes, respectivamente, aos bairros, às Àreas de Planejamento e aos nomes das Áreas de Proteção da cidade. Repare que uma mesma Área de Proteção pode estar em mais de um bairro e em mais de uma Área de Planejamento (e.g. APA da Orla da Baía de Sepetiba).

Temos que remover essas duplicatas, já que estamos interessados em saber quantas Áreas de Proteção individuais existem em cada uma das Áreas de Planejamento da cidade. Para isso, usamos a função **distinct**. Em seguida, usamos a função **group_by** e as funções **count** e **summarise**, para a estimativa final do número de Áreas de Proteção por Área de Planejamento no Rio de Janeiro. Por último, empregamos a função **rename** para atribuir às nossas colunas nomes que façam mais sentido, facilitando nossas análises de integração posteriores.

```
n_areas_protecao_zonas_rj<-aps_rio%>%
  select(area_plane,nome_1)%>%
  distinct(area_plane,nome_1)%>%
  group_by(area_plane)%>%
  count(nome_1)%>%
  summarise(n = sum(n))%>%
  rename("Área de Planejamento (AP)" = "area_plane",
         "Nº de Áreas de Proteção" = "n")
```

Integrando dados de uso do solo aos de Áreas Protegidas

Para integrar nossos *dataframes* finais, iremos utilizar as Áreas de Planejamento como ponto de referência. Essa "relação" entre dados de diferentes fontes é construída por meio de um conjunto de funções chamadas **joins**. Abaixo, estão ilustradas as diferentes possibilidades de combinações possíveis.



Cortesia do RStudio

Iremos utilizar a **left_join** para os nossos dados. Após isso, iremos criar um arquivo Excel no nosso diretório de trabalho, por meio da função **write_xlsx**. Repare que uma das maneiras de fazer isso sem utilizar a função **library** é aplicando o par de dois pontos (::), que conecta o pacote à função de interesse.

```
dados_integrados<-n_areas_protecao_zonas_rj%>%
  left_join(porcentagens_uso_do_solo_area_planejamento_rj,by="Área de
Planejamento (AP)")

writexl::write_xlsx(dados_integrados,"Relação Nº Áreas Protegidas X Uso
do solo - RJ.xlsx")
```

Quais conclusões que você pode chegar com os dados finais? Será possível definir com esses dados uma Área de Planejamento da cidade que seja, de fato, ideal para a alocação de novas Áreas de Proteção?

Um grande bloco único de código

Perceba que nesta aula quebramos o fluxo de trabalho do Tidyverse para tornar a sua compreensão mais fácil. No entanto, todas as funções aplicadas aos dados de uso do solo poderiam ser integradas por meio de *pipelines*, formando um grande bloco único de código. De primeira, essa visão pode nos assustar, mas conforme você for ganhando experiência, irá começar a programar cada vez mais frequentemente desta forma! Todas as funções abaixo utilizadas pertencem ao pacote **dplyr** do Tidyverse.

```
#trabalhando dados das Áreas de Proteção
n_areas_protecao_zonas_rj<-aps_rio%>%
  select(nome,area_plane,nome_1)%>%
  distinct(area_plane,nome_1)%>%
```

```
  group_by(area_plane)%>%
  count(nome_1)%>%
  summarise(n = sum(n))%>%
  rename("Área de Planejamento (AP)" = "area_plane",
         "Nº de Áreas de Proteção" = "n")

#trabalhando dados do uso do solo e integrando aos dados de Áreas de
Proteçao
dados_integrados_script_completo<-uso_solo_rio%>%
  relocate(Reflorestamento,.after = `Afloramento Rochoso`)%>%
  mutate(Vegetacao_natural = rowSums(pick(`Floresta Ombrófila
Densa`:Brejo)),.before = `Floresta Ombrófila Densa`)%>%
  mutate(Antropismos =  rowSums(pick(`Área
Urbana`:Reflorestamento)),.before = `Área Urbana`)%>%
  mutate(Corpos_dagua_continental = rowSums(pick(`Corpo d'água
continental`:Praia)),.before = `Corpo d'água continental`)%>%
  mutate(Uso_do_solo =
rowSums(pick(Vegetacao_natural,Antropismos,Corpos_dagua_continental)),.be
fore = Vegetacao_natural)%>%
  filter(`Bairros Rio de Janeiro` != "Lapa")%>%
  select(`Área de Planejamento
(AP)`,Uso_do_solo,Vegetacao_natural,Antropismos,Corpos_dagua_continental)
%>%
  group_by(`Área de Planejamento (AP)`)%>%
  summarise(across(everything(),list(sum)))%>%
  mutate(Vegetacao_natural_Porcentagem =
(Vegetacao_natural_1/Uso_do_solo_1)*100)%>%
  mutate(Antropismos_Porcentagem = (Antropismos_1/Uso_do_solo_1)*100)%>%
  mutate(Corpos_dagua_Porcentagem =
(Corpos_dagua_continental_1/Uso_do_solo_1)*100)%>%
  mutate(across(where(is.numeric), round, 2))%>%
  select(`Área de Planejamento
(AP)`,Vegetacao_natural_Porcentagem:Corpos_dagua_Porcentagem)%>%
  left_join(n_areas_protecao_zonas_rj,by="Área de Planejamento (AP)")
```

Exercício - Porcentagens do uso do solo nos bairros do Rio de Janeiro

Sua missão agora é fornecer a porcentagem dos grandes grupos de classes de uso do solo (“Vegetação natural”, “Antropismos” e “Corpos d’água continentais”) para cada bairro do Rio de Janeiro. Aplique os conhecimentos que aprendeu anteriormente para criar o seu script!