Pirâmides etárias e outros gráficos de população

Tópicos Especiais em Métodos Quantitativos - UFF Campos

26 de abril de 2022

Table of Contents

**Observações:** (1). Este documento foi desenvolvido como uma nota de aula e é parte integrante do curso de Tópicos Especiais em Métodos Quantitativos, ofertado para o curso de geografia da Universidade Federal Fluminense, *campus* de Campos dos Goytacazes, no 1º semestre de 2022. Sinta-se livre para divulga-lo desde que referencie o documento original.

(2). Este conteúdo foi preparado para ser desenvolvido no [RStudio](rstudio.com), que é uma interface gráfica do [R](r-project.org).

*Sugestão de referência:*

GIVISIEZ, Gustavo Henrique Naves. Preparando e Limpando Banco de Dados [R Markdown Document]. . Campos dos Goytacazes (RJ): RPubs. Disponível em: <https://rpubs.com/ghnaves/PreparandoDB>. Acesso em: 26 abr. 2022. , 2022

# Introdução

# Fonte de dados

As bases de dados usadas nas atividades são baseadas em dados históricos e projetados da população mundial, organizada por países e regiões (Population Division 2019). Os dados originais podem ser consultados no site da [World Population Division, vinculado à ONU](https://population.un.org/wpp/)

Os dados usados nesse exercício podem ser baixados no arquivo compactado DB.zip disponível nesse [link](https://drive.google.com/open?id=1UGnA2OzXyYwb_XWjh15fj759nb8k7h0w&authuser=gh_naves%40id.uff.br&usp=drive_fs). Os arquivos são:

* [WPP2019\_POP\_F07\_2\_POPULATION\_BY\_AGE\_MALE.XLSX](https://population.un.org/wpp/Download/Files/1_Indicators%20(Standard)/EXCEL_FILES/1_Population/WPP2019_POP_F07_2_POPULATION_BY_AGE_MALE.xlsx): Quinquennial Population by Five-Year Age Groups - Male. De facto population as of 1 July of the year indicated classified by five-year age groups (0-4, 5-9, 10-14, …, 95-99, 100+). Data are presented in thousands.
* [WPP2019\_POP\_F07\_3\_POPULATION\_BY\_AGE\_FEMALE.XLSX](https://population.un.org/wpp/Download/Files/1_Indicators%20(Standard)/EXCEL_FILES/1_Population/WPP2019_POP_F07_3_POPULATION_BY_AGE_FEMALE.xlsx): Quinquennial Population by Five-Year Age Groups - Female. De facto population as of 1 July of the year indicated classified by five-year age groups (0-4, 5-9, 10-14, …, 95-99, 100+). Data are presented in thousands.
* Arquivos adicionais com a documentação foram incluidos no arquivo DB.zip, sendo eles:
* *WPP2019\_Data-Sources* : Descrição das fontes de dados das projeções, país a país.
* *WPP2019\_F01\_LOCATIONS.xlsx* : Arquivo que auxilia o entendimento das subdivisões regionais, países e territórios.
* *WPP2019\_F02\_METADATA.xlsx* : Descrição de todos os campos de cada uma das tabelas geradas pela projeção

Inicialmente, criem um novo *projeto*, um novo *script* e salvem os arquivos acima referenciados na pasta do projeto.

# Leitura das Bases

A importação dos dados não nos traz tabelas utilizáveis a princípio. Mas, deem uma investigada nos arquivos excel e vamos tentar entender o banco de dados da ONU.

library(tidyverse)  
library(readxl)  
inicio<-read\_xlsx('./DB/WPP2019\_POP\_F07\_2\_POPULATION\_BY\_AGE\_MALE.xlsx')  
str(inicio)

## tibble [3,837 × 29] (S3: tbl\_df/tbl/data.frame)  
## $ United Nations: chr [1:3837] "Population Division" "Department of Economic and Social Affairs" NA "World Population Prospects 2019" ...  
## $ ...2 : chr [1:3837] NA NA NA NA ...  
## $ ...3 : chr [1:3837] NA NA NA NA ...  
## $ ...4 : chr [1:3837] NA NA NA NA ...  
## $ ...5 : chr [1:3837] NA NA NA NA ...  
## $ ...6 : chr [1:3837] NA NA NA NA ...  
## $ ...7 : chr [1:3837] NA NA NA NA ...  
## $ ...8 : chr [1:3837] NA NA NA NA ...  
## $ ...9 : chr [1:3837] NA NA NA NA ...  
## $ ...10 : chr [1:3837] NA NA NA NA ...  
## $ ...11 : chr [1:3837] NA NA NA NA ...  
## $ ...12 : chr [1:3837] NA NA NA NA ...  
## $ ...13 : chr [1:3837] NA NA NA NA ...  
## $ ...14 : chr [1:3837] NA NA NA NA ...  
## $ ...15 : chr [1:3837] NA NA NA NA ...  
## $ ...16 : chr [1:3837] NA NA NA NA ...  
## $ ...17 : chr [1:3837] NA NA NA NA ...  
## $ ...18 : chr [1:3837] NA NA NA NA ...  
## $ ...19 : chr [1:3837] NA NA NA NA ...  
## $ ...20 : chr [1:3837] NA NA NA NA ...  
## $ ...21 : chr [1:3837] NA NA NA NA ...  
## $ ...22 : chr [1:3837] NA NA NA NA ...  
## $ ...23 : chr [1:3837] NA NA NA NA ...  
## $ ...24 : chr [1:3837] NA NA NA NA ...  
## $ ...25 : chr [1:3837] NA NA NA NA ...  
## $ ...26 : chr [1:3837] NA NA NA NA ...  
## $ ...27 : chr [1:3837] NA NA NA NA ...  
## $ ...28 : chr [1:3837] NA NA NA NA ...  
## $ ...29 : chr [1:3837] NA NA NA NA ...

O processamento incial deve considerar organização dessas bases de dados. A intenção é criar um banco de dados com as seguintes variáveis:

* *variant*: Tipo do dado (Estimativa da População: Medium Variant, Low Variant e High Variant)
* *region*: Nome da região, subregion, país ou área.
* *cntycode*: Código a região, subregion, país ou área.
* *type*: identificação do tipo (World Label/Separator, Development Group, Special other, Income Group, Region, SDG region, Subregion, Country/Area ou SDG subregion)
* *parentcode*: Código da regiao em que o *cntycode* está contido.
* *sex*: Sexo
* *idgr*: grupos quinquenais de idade
* *year*: Ano de referencia

## Estimativas populacionais entre 1950 e 2020

est.male<-read\_xlsx('./DB/WPP2019\_POP\_F07\_2\_POPULATION\_BY\_AGE\_MALE.xlsx',  
 range='B17:AC3842',sheet='ESTIMATES',col\_names = T)  
  
str(est.male)

## tibble [3,825 × 28] (S3: tbl\_df/tbl/data.frame)  
## $ Variant : chr [1:3825] "Estimates" "Estimates" "Estimates" "Estimates" ...  
## $ Region, subregion, country or area \*: chr [1:3825] "WORLD" "WORLD" "WORLD" "WORLD" ...  
## $ Notes : chr [1:3825] NA NA NA NA ...  
## $ Country code : num [1:3825] 900 900 900 900 900 900 900 900 900 900 ...  
## $ Type : chr [1:3825] "World" "World" "World" "World" ...  
## $ Parent code : num [1:3825] 0 0 0 0 0 0 0 0 0 0 ...  
## $ Reference date (as of 1 July) : num [1:3825] 1950 1955 1960 1965 1970 ...  
## $ 0-4 : chr [1:3825] "172419.83199999999" "207941.39199999999" "221606.23" "244985.413" ...  
## $ 5-9 : chr [1:3825] "138298.389" "161087.46799999999" "196202.91500000001" "210103.74299999999" ...  
## $ 10-14 : chr [1:3825] "133685.70199999999" "134816.62100000001" "158034.84599999999" "192262.07800000001" ...  
## $ 15-19 : chr [1:3825] "122155.285" "130542.26700000001" "131917.18400000001" "155046.204" ...  
## $ 20-24 : chr [1:3825] "113206.51700000001" "118354.001" "126960.74099999999" "128691.856" ...  
## $ 25-29 : chr [1:3825] "97625.680999999997" "109514.656" "115443.47900000001" "123991.745" ...  
## $ 30-34 : chr [1:3825] "84569.559000000096" "94210.615999999995" "106320.534" "112472.30100000001" ...  
## $ 35-39 : chr [1:3825] "81447.199999999997" "81153.535999999993" "90924.086999999898" "103170.762" ...  
## $ 40-44 : chr [1:3825] "73162.817999999999" "77551.460999999996" "77485.353000000003" "87399.431000000099" ...  
## $ 45-49 : chr [1:3825] "63547.845000000001" "68433.11" "73198.497000000003" "73242.922999999995" ...  
## $ 50-54 : chr [1:3825] "52643.999000000003" "58721.707999999999" "63487.697" "67940.417000000001" ...  
## $ 55-59 : chr [1:3825] "42559.548000000003" "46807.777000000002" "52410.457000000002" "57212.928999999996" ...  
## $ 60-64 : chr [1:3825] "34381.408000000003" "36007.392999999996" "39687.949999999997" "44788.076999999997" ...  
## $ 65-69 : chr [1:3825] "25077.246999999999" "26899.246999999999" "28289.214" "31640.648000000001" ...  
## $ 70-74 : chr [1:3825] "16576.97" "17544.451000000001" "18948.032999999999" "20284.195" ...  
## $ 75-79 : chr [1:3825] "9343.6820000000007" "9942.3320000000094" "10560.571" "11822.536" ...  
## $ 80-84 : chr [1:3825] "3876.7190000000001" "4466.3819999999996" "4791.4930000000004" "5344.8940000000002" ...  
## $ 85-89 : chr [1:3825] "1296.646" "1313.7460000000001" "1558.046" "1774.7370000000001" ...  
## $ 90-94 : chr [1:3825] "316.51" "296.80099999999999" "308.46100000000001" "389.38299999999998" ...  
## $ 95-99 : chr [1:3825] "58.606000000000002" "47.271999999999998" "45.901000000000003" "49.616999999999997" ...  
## $ 100+ : chr [1:3825] "9.3930000000000007" "6.0619999999999896" "5.0129999999999999" "4.99" ...

Importou tudo como character e coluna *Notes* não nos interessa. Percebam que o comando importou os nomes das colunas que estão originalmente na linha 17 do Excel: range='B17:AC3842'

col\_types <- c("text","text","skip","numeric","text", rep("numeric",23))  
  
est.male<-read\_xlsx('./DB/WPP2019\_POP\_F07\_2\_POPULATION\_BY\_AGE\_MALE.xlsx',  
 range='B17:AC3842',  
 sheet='ESTIMATES', col\_names = T,  
 col\_types = col\_types)  
  
str(est.male)

## tibble [3,825 × 27] (S3: tbl\_df/tbl/data.frame)  
## $ Variant : chr [1:3825] "Estimates" "Estimates" "Estimates" "Estimates" ...  
## $ Region, subregion, country or area \*: chr [1:3825] "WORLD" "WORLD" "WORLD" "WORLD" ...  
## $ Country code : num [1:3825] 900 900 900 900 900 900 900 900 900 900 ...  
## $ Type : chr [1:3825] "World" "World" "World" "World" ...  
## $ Parent code : num [1:3825] 0 0 0 0 0 0 0 0 0 0 ...  
## $ Reference date (as of 1 July) : num [1:3825] 1950 1955 1960 1965 1970 ...  
## $ 0-4 : num [1:3825] 172420 207941 221606 244985 267293 ...  
## $ 5-9 : num [1:3825] 138298 161087 196203 210104 234848 ...  
## $ 10-14 : num [1:3825] 133686 134817 158035 192262 207446 ...  
## $ 15-19 : num [1:3825] 122155 130542 131917 155046 189471 ...  
## $ 20-24 : num [1:3825] 113207 118354 126961 128692 151513 ...  
## $ 25-29 : num [1:3825] 97626 109515 115443 123992 126114 ...  
## $ 30-34 : num [1:3825] 84570 94211 106321 112472 121401 ...  
## $ 35-39 : num [1:3825] 81447 81154 90924 103171 109517 ...  
## $ 40-44 : num [1:3825] 73163 77551 77485 87399 100025 ...  
## $ 45-49 : num [1:3825] 63548 68433 73198 73243 83743 ...  
## $ 50-54 : num [1:3825] 52644 58722 63488 67940 68800 ...  
## $ 55-59 : num [1:3825] 42560 46808 52410 57213 62367 ...  
## $ 60-64 : num [1:3825] 34381 36007 39688 44788 50239 ...  
## $ 65-69 : num [1:3825] 25077 26899 28289 31641 36906 ...  
## $ 70-74 : num [1:3825] 16577 17544 18948 20284 23612 ...  
## $ 75-79 : num [1:3825] 9344 9942 10561 11823 13196 ...  
## $ 80-84 : num [1:3825] 3877 4466 4791 5345 6288 ...  
## $ 85-89 : num [1:3825] 1297 1314 1558 1775 2079 ...  
## $ 90-94 : num [1:3825] 317 297 308 389 461 ...  
## $ 95-99 : num [1:3825] 58.6 47.3 45.9 49.6 65 ...  
## $ 100+ : num [1:3825] 9.39 6.06 5.01 4.99 5.6 ...

Posso até importar com os nomes das colunas, usando a opção col\_nacmes = T, como feito anteriormente, mas os nomes não serão queles que nós queremos.

Para definir o nome das colunas, existem duas formas:

1). Como o comando colnames é possível renomear as colunas do data.frame.

colnames<-c('variant','region','cntycode','type','parentcode','year',  
 '0-4','5-9','10-14','15-19', '20-24',  
 '25-29','30-34','35-39','40-44','45-49',  
 '50-54','55-59','60-64','65-69','70-74',  
 '75-79','80-84','85-89','90-94','95-99','100+')  
  
colnames(est.male) <-colnames  
str(est.male)

## tibble [3,825 × 27] (S3: tbl\_df/tbl/data.frame)  
## $ variant : chr [1:3825] "Estimates" "Estimates" "Estimates" "Estimates" ...  
## $ region : chr [1:3825] "WORLD" "WORLD" "WORLD" "WORLD" ...  
## $ cntycode : num [1:3825] 900 900 900 900 900 900 900 900 900 900 ...  
## $ type : chr [1:3825] "World" "World" "World" "World" ...  
## $ parentcode: num [1:3825] 0 0 0 0 0 0 0 0 0 0 ...  
## $ year : num [1:3825] 1950 1955 1960 1965 1970 ...  
## $ 0-4 : num [1:3825] 172420 207941 221606 244985 267293 ...  
## $ 5-9 : num [1:3825] 138298 161087 196203 210104 234848 ...  
## $ 10-14 : num [1:3825] 133686 134817 158035 192262 207446 ...  
## $ 15-19 : num [1:3825] 122155 130542 131917 155046 189471 ...  
## $ 20-24 : num [1:3825] 113207 118354 126961 128692 151513 ...  
## $ 25-29 : num [1:3825] 97626 109515 115443 123992 126114 ...  
## $ 30-34 : num [1:3825] 84570 94211 106321 112472 121401 ...  
## $ 35-39 : num [1:3825] 81447 81154 90924 103171 109517 ...  
## $ 40-44 : num [1:3825] 73163 77551 77485 87399 100025 ...  
## $ 45-49 : num [1:3825] 63548 68433 73198 73243 83743 ...  
## $ 50-54 : num [1:3825] 52644 58722 63488 67940 68800 ...  
## $ 55-59 : num [1:3825] 42560 46808 52410 57213 62367 ...  
## $ 60-64 : num [1:3825] 34381 36007 39688 44788 50239 ...  
## $ 65-69 : num [1:3825] 25077 26899 28289 31641 36906 ...  
## $ 70-74 : num [1:3825] 16577 17544 18948 20284 23612 ...  
## $ 75-79 : num [1:3825] 9344 9942 10561 11823 13196 ...  
## $ 80-84 : num [1:3825] 3877 4466 4791 5345 6288 ...  
## $ 85-89 : num [1:3825] 1297 1314 1558 1775 2079 ...  
## $ 90-94 : num [1:3825] 317 297 308 389 461 ...  
## $ 95-99 : num [1:3825] 58.6 47.3 45.9 49.6 65 ...  
## $ 100+ : num [1:3825] 9.39 6.06 5.01 4.99 5.6 ...

2). Ou importar e, no meso comando, informar os nomes para o comando read\_xlsx no argumento colnames (col\_names = colnames). Percebam que para importar informando os nomes das colunas, o argumento range não pode incluir a linha que contém os nomes (Linha 17 no Excel) - range agora é range='B18:AC3842'

est.male<-read\_xlsx('./DB/WPP2019\_POP\_F07\_2\_POPULATION\_BY\_AGE\_MALE.xlsx',  
 range='B18:AC3842',  
 sheet='ESTIMATES',  
 col\_names = colnames,  
 col\_types = col\_types)  
str(est.male)

## tibble [3,825 × 27] (S3: tbl\_df/tbl/data.frame)  
## $ variant : chr [1:3825] "Estimates" "Estimates" "Estimates" "Estimates" ...  
## $ region : chr [1:3825] "WORLD" "WORLD" "WORLD" "WORLD" ...  
## $ cntycode : num [1:3825] 900 900 900 900 900 900 900 900 900 900 ...  
## $ type : chr [1:3825] "World" "World" "World" "World" ...  
## $ parentcode: num [1:3825] 0 0 0 0 0 0 0 0 0 0 ...  
## $ year : num [1:3825] 1950 1955 1960 1965 1970 ...  
## $ 0-4 : num [1:3825] 172420 207941 221606 244985 267293 ...  
## $ 5-9 : num [1:3825] 138298 161087 196203 210104 234848 ...  
## $ 10-14 : num [1:3825] 133686 134817 158035 192262 207446 ...  
## $ 15-19 : num [1:3825] 122155 130542 131917 155046 189471 ...  
## $ 20-24 : num [1:3825] 113207 118354 126961 128692 151513 ...  
## $ 25-29 : num [1:3825] 97626 109515 115443 123992 126114 ...  
## $ 30-34 : num [1:3825] 84570 94211 106321 112472 121401 ...  
## $ 35-39 : num [1:3825] 81447 81154 90924 103171 109517 ...  
## $ 40-44 : num [1:3825] 73163 77551 77485 87399 100025 ...  
## $ 45-49 : num [1:3825] 63548 68433 73198 73243 83743 ...  
## $ 50-54 : num [1:3825] 52644 58722 63488 67940 68800 ...  
## $ 55-59 : num [1:3825] 42560 46808 52410 57213 62367 ...  
## $ 60-64 : num [1:3825] 34381 36007 39688 44788 50239 ...  
## $ 65-69 : num [1:3825] 25077 26899 28289 31641 36906 ...  
## $ 70-74 : num [1:3825] 16577 17544 18948 20284 23612 ...  
## $ 75-79 : num [1:3825] 9344 9942 10561 11823 13196 ...  
## $ 80-84 : num [1:3825] 3877 4466 4791 5345 6288 ...  
## $ 85-89 : num [1:3825] 1297 1314 1558 1775 2079 ...  
## $ 90-94 : num [1:3825] 317 297 308 389 461 ...  
## $ 95-99 : num [1:3825] 58.6 47.3 45.9 49.6 65 ...  
## $ 100+ : num [1:3825] 9.39 6.06 5.01 4.99 5.6 ...

Repetir o mesmo comando para as mulheres

est.female<-read\_xlsx('./DB/WPP2019\_POP\_F07\_3\_POPULATION\_BY\_AGE\_FEMALE.xlsx',  
 range='B18:AC3842',  
 sheet='ESTIMATES',col\_names = colnames,  
 col\_types = col\_types)

## Projeção da população de 2020 a 2100

Seria basicamente replicar os comandos anteriores para importar as demais planilhas contidos nos arquivos (percebam que o orgumento range mudou).

medium.male<-read\_xlsx('./DB/WPP2019\_POP\_F07\_2\_POPULATION\_BY\_AGE\_MALE.xlsx',  
 range='B18:AC4352',  
 sheet='MEDIUM VARIANT',col\_names = colnames,  
 col\_types = col\_types)  
  
medium.female<-read\_xlsx('./DB/WPP2019\_POP\_F07\_3\_POPULATION\_BY\_AGE\_FEMALE.xlsx',  
 range='B18:AC4352',  
 sheet='MEDIUM VARIANT',col\_names = colnames,  
 col\_types = col\_types)  
  
high.male<-read\_xlsx('./DB/WPP2019\_POP\_F07\_2\_POPULATION\_BY\_AGE\_MALE.xlsx',  
 range='B18:AC4352',  
 sheet='HIGH VARIANT',col\_names = colnames,  
 col\_types = col\_types)  
  
high.female<-read\_xlsx('./DB/WPP2019\_POP\_F07\_3\_POPULATION\_BY\_AGE\_FEMALE.xlsx',  
 range='B18:AC4352',  
 sheet='HIGH VARIANT',col\_names = colnames,  
 col\_types = col\_types)  
  
low.male<-read\_xlsx('./DB/WPP2019\_POP\_F07\_2\_POPULATION\_BY\_AGE\_MALE.xlsx',  
 range='B18:AC4352',  
 sheet='LOW VARIANT',col\_names = colnames,  
 col\_types = col\_types)  
  
low.female<-read\_xlsx('./DB/WPP2019\_POP\_F07\_3\_POPULATION\_BY\_AGE\_FEMALE.xlsx',  
 range='B18:AC4352',  
 sheet='LOW VARIANT',col\_names = colnames,  
 col\_types = col\_types)

Na sequencia o importante é criar uma variável que identique o sexo da população estimada e projetada.

est.male <- est.male %>%  
 mutate(sex='Male')  
  
est.female <- est.female %>%  
 mutate(sex='Female')  
  
medium.male <- medium.male %>%  
 mutate(sex='Male')  
  
medium.female <- medium.female %>%  
 mutate(sex='Female')  
  
low.male <- low.male %>%  
 mutate(sex='Male')  
  
low.female <- low.female %>%  
 mutate(sex='Female')  
  
high.male <- high.male %>%  
 mutate(sex='Male')  
  
high.female <- high.female %>%  
 mutate(sex='Female')  
  
#### OU  
  
#high.female$sex<-'Female'  
# ...

O próximo passo é agregar os banco de dados com comandos rbind que pode ser traduzido como *ligar* ou *vincular* por linhas ( *r* de *row*). Para isso as colunas devem ter sempre os mesmos nomes e o mesmo tipo.

**Dica**: O uso de um vetor do tipo *colnames* e *coltypes* evita erros de dgitação.

pop<-est.male %>%  
 rbind(est.female)%>%  
 rbind(medium.male)%>%  
 rbind(medium.female)%>%  
 rbind(low.male)%>%  
 rbind(low.female)%>%  
 rbind(high.female)%>%  
 rbind(high.male)

O próxima passo é o mais complicado: transformar as colunas dos grupos de idades em uma variável. Esse comando altera os dados de *wide* ( *largo* em ingles) para *long* ( *comprido* em ingles). Para ver como esse comando funciona nesse [link do youtube](https://youtu.be/2KF7ovJZUYw). Os comandos a seguir replicam os comandos gather ( *reunir* em inglês) e spread desse vídeo do Youtube. Essas funções fazem parte do pacote tidyr que é caregado conjuntamente no comando library(tidyverse) rodado anteriormete.

library(tidyverse)  
dados\_wide<-tibble(pais=c('Brasil','Alemanha'),`2002`=c(2,0),`2014`=c(1,7))  
dados\_wide

## # A tibble: 2 × 3  
## pais `2002` `2014`  
## <chr> <dbl> <dbl>  
## 1 Brasil 2 1  
## 2 Alemanha 0 7

dados\_long<-dados\_wide%>%  
 gather(key=ano,value=gols,-pais)  
  
dados\_long

## # A tibble: 4 × 3  
## pais ano gols  
## <chr> <chr> <dbl>  
## 1 Brasil 2002 2  
## 2 Alemanha 2002 0  
## 3 Brasil 2014 1  
## 4 Alemanha 2014 7

dados\_wide2 <- dados\_long %>%  
 spread(key=ano, value=gols)  
dados\_wide2

## # A tibble: 2 × 3  
## pais `2002` `2014`  
## <chr> <dbl> <dbl>  
## 1 Alemanha 0 7  
## 2 Brasil 2 1

O nosso banco de dados pop está no formato *wide*. A ideia é tranformar no formato *long* com a variável *idgr* identificando o grupo quinquenal de idade. Para isso usamos a função gather no banco de dados de população. Precisamos manter várias colunas (e não apenas a coluna país). Para isso, veja o código a seguir:

pop<-pop %>%  
 gather(key=idgr,value=pop,  
 -c('variant','region','cntycode','type','parentcode','year','sex'))

Outro ponto válido é transformar algumas variáveis para o tipo fator. Isso sempre gera gráficos mais interessantes.

pop<-pop %>%  
 mutate(variant=factor(variant),  
 type=factor(type),  
 year=factor(year,ordered=T,  
 levels=seq(1950,2100,5)),  
 sex=factor(sex),  
 idgr=factor(idgr,ordered=T,  
 levels=c('0-4','5-9','10-14','15-19', '20-24', '25-29',  
 '30-34','35-39','40-44','45-49','50-54','55-59',  
 '60-64','65-69','70-74','75-79','80-84','85-89',  
 '90-94','95-99','100+')))

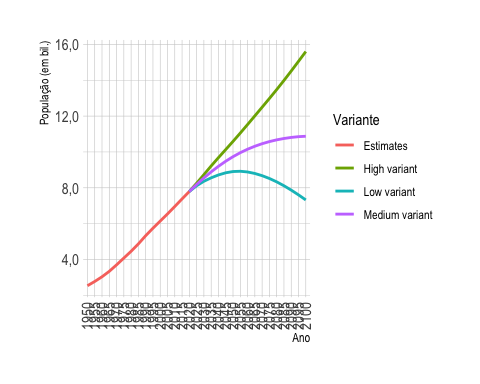
# Gráficos associados a esse banco de dados.

Agora o banco de dados está quase pronto no formato *long*, o próximo passo é fazer pirâmides etárias e outros gráficos de interesse. Mas, a organização deste banco de ados da ONU considera várias agregaç!oes espaciais. Sendo assim, o banco de dados ainda precisa de alguns filtros a partir do data.frame *pop*

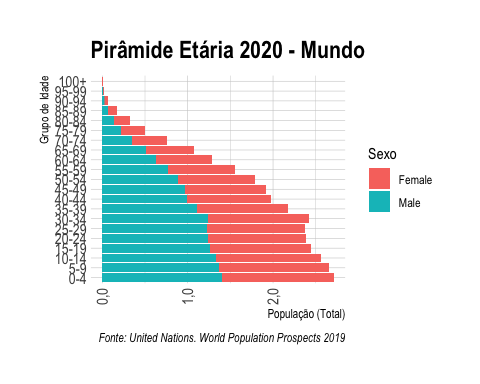
## População do Mundo x Ano

A variavel region e a cntycode identificam a região em análise. Elas podem ser regiões (Ásia, por exemplo) ou apenas um país (Brazil, por exemplo). Para entender como os bancos de ados da ONU se organizam consultem o arquivo WPP2019\_F01\_LOCATIONS.XLSX.

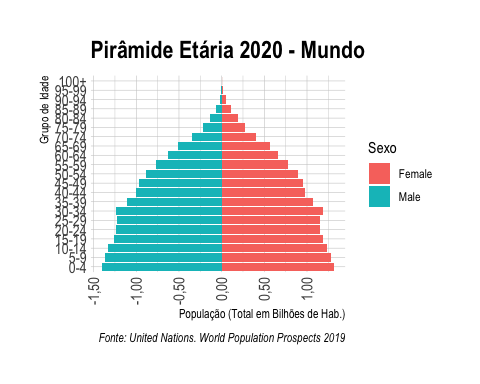
#World  
  
pop\_World<-pop%>%  
 filter(region=='WORLD')  
  
library(scales)  
library(hrbrthemes)  
  
#th<-theme\_ipsum\_pub()  
th<-theme\_ipsum() +  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust=1))  
  
temp<-pop\_World%>%  
 group\_by(year,variant)%>%  
 summarise(pop=sum(pop))  
  
ggplot(data=temp,aes(x=year,y=pop,group=variant,col=variant))+  
 geom\_line(size=1)+  
 scale\_y\_continuous(name='População (em bil.)',  
 labels=label\_comma(big.mark = '.',  
 decimal.mark = ',',scale=10^-6))+  
 scale\_x\_discrete(name='Ano')+  
 scale\_color\_discrete(name='Variante')+  
 th



temp<-pop\_World%>%  
 filter(year=='2020')  
  
ggplot(data=temp,aes(x=idgr,y=pop,group=sex,fill=sex))+  
 geom\_bar(stat='identity')+  
 coord\_flip()+  
 labs(caption = "Fonte: United Nations. World Population Prospects 2019")+  
 ggtitle('Pirâmide Etária 2020 - Mundo')+  
 scale\_y\_continuous(name='População (Total)',  
 labels=label\_number(big.mark = '.',decimal.mark = ',',  
 scale=10^-6))+  
 scale\_x\_discrete(name='Grupo de Idade')+  
 scale\_fill\_discrete(name='Sexo')+  
 th



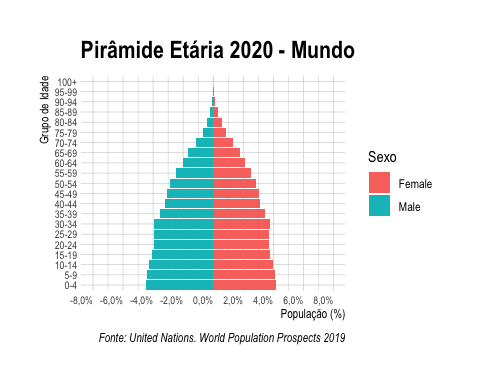
temp<-pop\_World%>%  
 filter(year=='2020')%>%  
 mutate(poppir=if\_else(sex=='Male',-pop,pop))  
  
ggplot(data=temp,aes(x=idgr,y=poppir,group=sex,fill=sex))+  
 geom\_bar(stat='identity')+  
 coord\_flip()+  
 labs(caption = "Fonte: United Nations. World Population Prospects 2019")+  
 ggtitle('Pirâmide Etária 2020 - Mundo')+  
scale\_y\_continuous(name='População (Total em Bilhões de Hab.)',  
 labels=label\_number(big.mark = '.',decimal.mark = ',',  
 scale=10^-6))+  
 scale\_x\_discrete(name='Grupo de Idade')+  
 scale\_fill\_discrete(name='Sexo')+  
 th



temp<-pop\_World%>%  
 filter(year=='2020')%>%  
 group\_by(sex,idgr)%>%  
 summarise(pop=sum(pop))%>%  
 ungroup()%>%  
 mutate(poprel=pop/(sum(pop)))%>%  
 mutate(poprel=if\_else(sex=='Male',-poprel,poprel))

## `summarise()` has grouped output by 'sex'. You can override using the `.groups` argument.

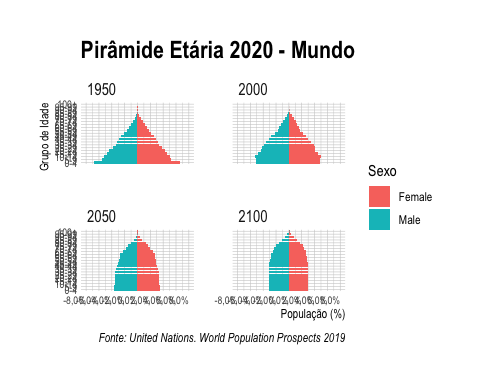
th<-theme\_ipsum()+  
 theme(axis.text.x = element\_text(angle = 0, vjust = 0.5, hjust=1,size=8),  
 axis.text.y = element\_text(angle = 0, vjust = 0.5, hjust=1,size=8))  
  
ggplot(data=temp,aes(x=idgr,y=poprel,group=sex,fill=sex))+  
 geom\_bar(stat='identity')+  
 coord\_flip()+  
 labs(caption = "Fonte: United Nations. World Population Prospects 2019")+  
 ggtitle('Pirâmide Etária 2020 - Mundo')+  
 scale\_y\_continuous(name='População (%)',  
 labels=label\_percent(big.mark = '.',decimal.mark = ','),  
 breaks = seq(-.08,.08,.02),limits=c(-.08,.08))+  
 scale\_x\_discrete(name='Grupo de Idade')+  
 scale\_fill\_discrete(name='Sexo')+  
 th



temp<-pop\_World%>%  
 filter(year %in% seq(1950,2100,50))%>%  
 group\_by(sex,idgr,year)%>%  
 summarise(pop=sum(pop))%>%  
 ungroup()%>%  
 group\_by(year)%>%  
 mutate(poprel=pop/(sum(pop)))%>%  
 mutate(poprel=if\_else(sex=='Male',-poprel,poprel))%>%  
 ungroup()

## `summarise()` has grouped output by 'sex', 'idgr'. You can override using the `.groups` argument.

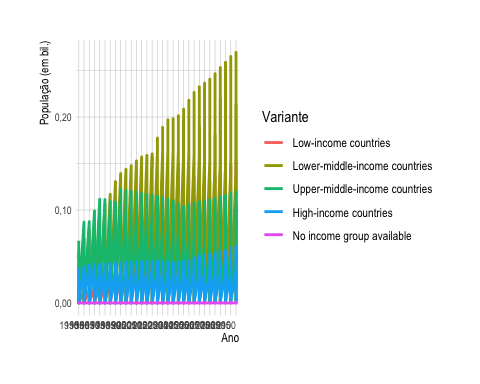
th<-theme\_ipsum()+  
 theme(axis.text.x = element\_text(angle = 0, vjust = 0.5, hjust=1,size=8),  
 axis.text.y = element\_text(angle = 0, vjust = 0.5, hjust=1,size=8))  
  
ggplot(data=temp,aes(x=idgr,y=poprel,group=sex,fill=sex))+  
 geom\_bar(stat='identity')+  
 coord\_flip()+  
 facet\_wrap(~year)+  
 labs(caption = "Fonte: United Nations. World Population Prospects 2019")+  
 ggtitle('Pirâmide Etária 2020 - Mundo')+  
 scale\_y\_continuous(name='População (%)',  
 labels=label\_percent(big.mark = '.',decimal.mark = ','),  
 breaks = seq(-.08,.08,.02),limits=c(-.08,.08))+  
 scale\_x\_discrete(name='Grupo de Idade')+  
 scale\_fill\_discrete(name='Sexo')+  
 th



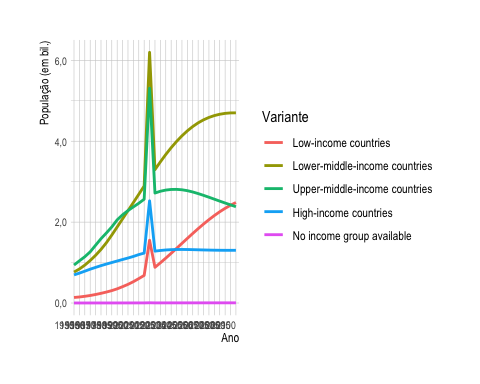
levels(pop$type)

## [1] "Country/Area" "Development Group" "Income Group"   
## [4] "Label/Separator" "Region" "SDG region"   
## [7] "SDG subregion" "Special other" "Subregion"   
## [10] "World"

# World Bank income groups  
 # 1. High-income countries  
 # 2. Middle-income countries  
 # 2.2 Upper-middle-income countries  
 # 2.2 Lower-middle-income countries  
 # 3. Low-income countries  
 # 4. No income group available  
  
pop\_Income<-pop%>%  
 filter(type=="Income Group" & region != 'Middle-income countries')%>%  
 mutate(region=factor(region,ordered=T,  
 levels=c("Low-income countries",  
 "Lower-middle-income countries",  
 "Upper-middle-income countries",  
 "High-income countries",  
 "No income group available"  
 )))  
  
ggplot(data=pop\_Income,aes(x=year,y=pop,group=region,col=region))+  
 geom\_line(size=1)+  
 scale\_y\_continuous(name='População (em bil.)',  
 labels=label\_comma(big.mark = '.',  
 decimal.mark = ',',scale=10^-6))+  
 scale\_x\_discrete(name='Ano')+  
 scale\_color\_discrete(name='Variante')+  
 th

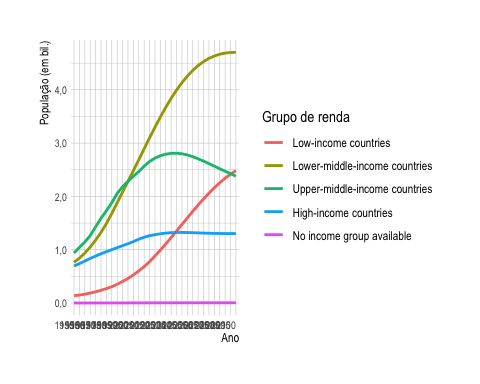


temp<-pop\_Income%>%  
 filter(type=='Income Group',variant %in% c('Estimates','Medium variant'))%>%  
 group\_by(year,region)%>%  
 summarise(pop=sum(pop))  
  
ggplot(data=temp,aes(x=year,y=pop,group=region,col=region))+  
 geom\_line(size=1)+  
 scale\_y\_continuous(name='População (em bil.)',  
 labels=label\_comma(big.mark = '.',  
 decimal.mark = ',',scale=10^-6))+  
 scale\_x\_discrete(name='Ano')+  
 scale\_color\_discrete(name='Variante')+  
 th



Nota-se que a linha para o ano 2020 aparece nas estimativas e também nas projeções. E, por isso, elea é somada duas vezes oacasionando esse “soluço” no gráfico. Uma solução é, por exemplo, excluir as linhas em que a variavel year==2020 e que variant=='Estimates'.

temp<-pop\_Income%>%  
 filter(type=='Income Group',variant %in% c('Estimates','Medium variant'))%>%  
 filter(!(year==2020 & variant=='Estimates'))%>%  
 group\_by(year,region)%>%  
 summarise(pop=sum(pop))  
  
  
ggplot(data=temp,aes(x=year,y=pop,group=region,col=region))+  
 geom\_line(size=1)+  
 scale\_y\_continuous(name='População (em bil.)',  
 labels=label\_comma(big.mark = '.',  
 decimal.mark = ',',scale=10^-6))+  
 scale\_x\_discrete(name='Ano')+  
 scale\_color\_discrete(name='Grupo de renda')+  
 th



##knitr::purl('PreparandoDB.Rmd')

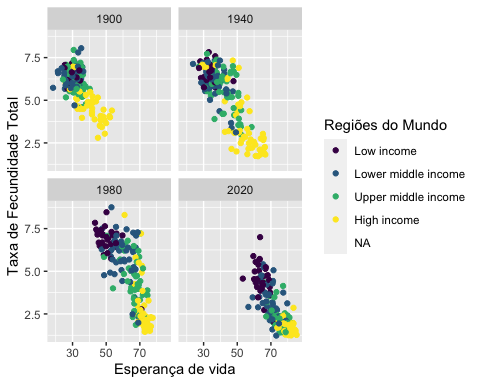
library(ggsci)  
library(scales)  
library(hrbrthemes)  
library(gridExtra)

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

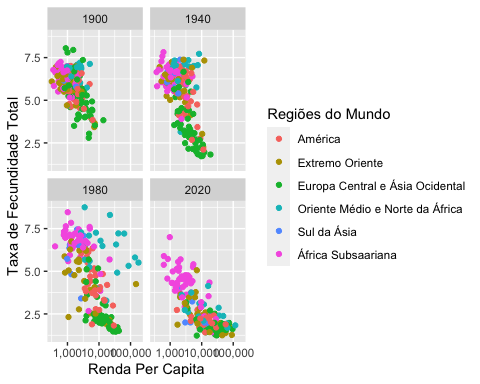
Gapminder<-readRDS('Gapminder.rds')  
  
Gapminder\_1880a2020<- Gapminder %>%  
 filter(time %in% seq(1900,2040,40))  
  
ggplot(data=Gapminder\_1880a2020,aes(y=tfr,x=le,color=WB\_income2017))+  
 geom\_point()+  
 scale\_x\_continuous(labels=comma)+  
 labs(x='Esperança de vida', y = 'Taxa de Fecundidade Total',color='Regiões do Mundo')+  
 facet\_wrap(~time)

## Warning: Removed 44 rows containing missing values (geom\_point).



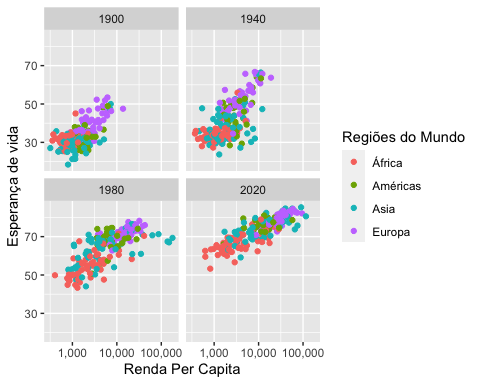
ggplot(data=Gapminder\_1880a2020,aes(y=tfr,x=ipp,color=six\_regions))+  
 geom\_point()+  
 scale\_x\_log10(labels=comma)+  
 labs(x='Renda Per Capita', y = 'Taxa de Fecundidade Total',color='Regiões do Mundo')+  
 facet\_wrap(~time)

## Warning: Removed 44 rows containing missing values (geom\_point).

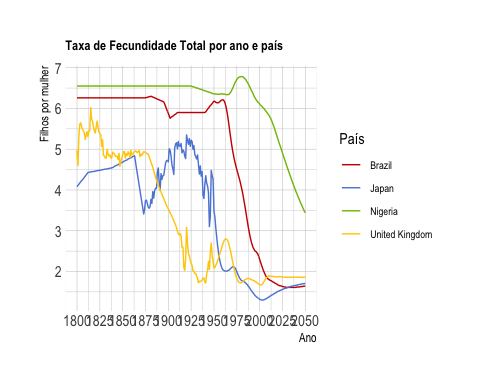


ggplot(data=Gapminder\_1880a2020,aes(y=le,x=ipp,color=four\_regions))+  
 geom\_point()+  
 scale\_x\_log10(labels=comma)+  
 labs(x='Renda Per Capita', y = 'Esperança de vida',color='Regiões do Mundo')+  
 facet\_wrap(~time)

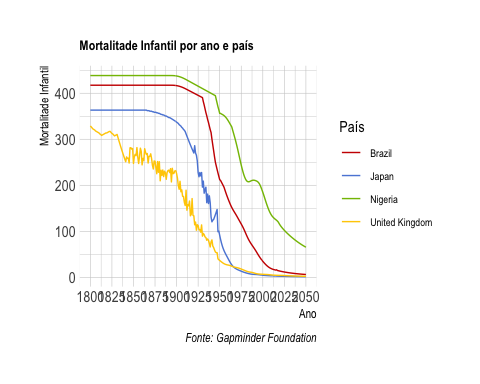
## Warning: Removed 41 rows containing missing values (geom\_point).



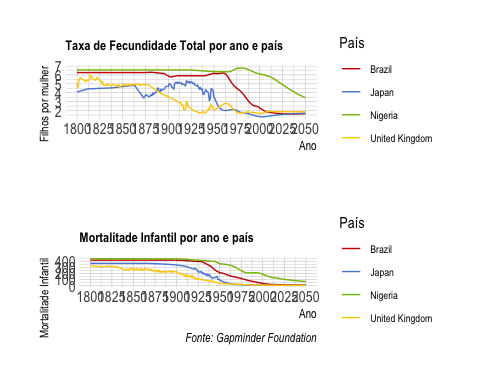
Gapminder\_Varios<- Gapminder %>%  
 filter((name == 'Brazil'|name=='Japan'  
 |name=='Nigeria'|name == 'United Kingdom')&  
 time<=2050)  
  
th<-theme\_ipsum()+  
 theme(plot.title = element\_text(size = 10),  
 legend.text = element\_text(size = 8),  
 axis.text=element\_text(size = 8))  
  
ggtfr<-ggplot(data=Gapminder\_Varios,aes(y=tfr,x=time,color=name))+  
 geom\_line()+  
 ggtitle('Taxa de Fecundidade Total por ano e país')+  
 scale\_x\_continuous(breaks = seq(1800,2100,25))+  
 scale\_color\_startrek()+  
 labs(x='Ano', y = 'Filhos por mulher',color='País')+  
 th  
  
ggmi<-ggplot(data=Gapminder\_Varios,aes(y=cm,x=time,color=name))+  
 geom\_line()+  
 ggtitle('Mortalitade Infantil por ano e país')+  
 scale\_color\_startrek()+  
 scale\_x\_continuous(breaks = seq(1800,2100,25))+  
 labs(x='Ano', y = 'Mortalitade Infantil',color='País',  
 caption = "Fonte: Gapminder Foundation")+  
 th  
  
ggtfr



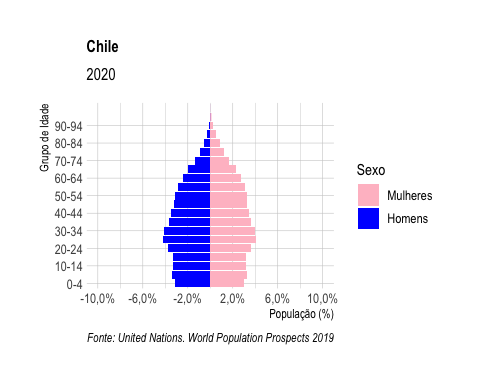
ggmi



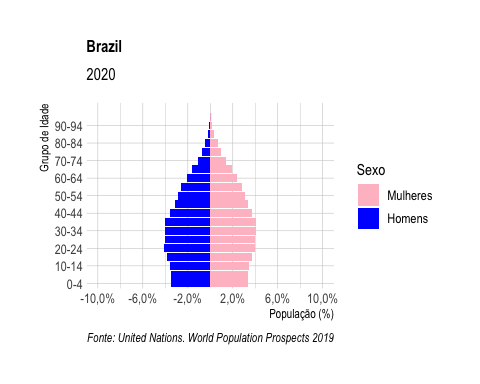
ggarr<-grid.arrange(ggtfr, ggmi,  
 ncol = 1, nrow = 2)



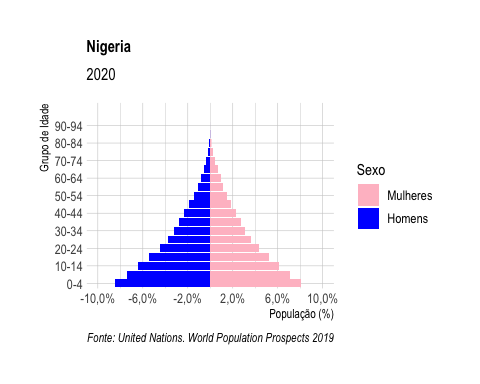
brazil<-pop%>%  
 filter(year==2020 &  
 region=='Brazil' &  
 variant %in% c('Medium variant','Estimates'))%>%  
 group\_by(year)%>%  
 mutate(poptotal=sum(pop))%>%  
 ungroup()%>%  
 mutate(poprel=pop/poptotal)%>%  
 mutate(poprel=if\_else(sex=='Male',-poprel,poprel))  
  
usa<-pop%>%  
 filter(year==2020 &  
 region=='United States of America' &  
 variant %in% c('Medium variant','Estimates'))%>%  
 group\_by(year)%>%  
 mutate(poptotal=sum(pop))%>%  
 ungroup()%>%  
 mutate(poprel=pop/poptotal)%>%  
 mutate(poprel=if\_else(sex=='Male',-poprel,poprel))  
  
nigeria<-pop%>%  
 filter(year==2020 &  
 region=='Nigeria' &  
 variant %in% c('Medium variant','Estimates'))%>%  
 group\_by(year)%>%  
 mutate(poptotal=sum(pop))%>%  
 ungroup()%>%  
 mutate(poprel=pop/poptotal)%>%  
 mutate(poprel=if\_else(sex=='Male',-poprel,poprel))  
  
chile<-pop%>%  
 filter(year==2020 &  
 region=='Chile' &  
 variant %in% c('Medium variant','Estimates'))%>%  
 group\_by(year)%>%  
 mutate(poptotal=sum(pop))%>%  
 ungroup()%>%  
 mutate(poprel=pop/poptotal)%>%  
 mutate(poprel=if\_else(sex=='Male',-poprel,poprel))  
  
th<-theme\_ipsum()+  
 theme(plot.title = element\_text(size = 12),  
 legend.text = element\_text(size = 10),  
 axis.text=element\_text(size = 10),  
 axis.text.x=element\_text(size = 10),  
 axis.text.y=element\_text(size = 10))  
  
ggplot(data=chile,aes(x=idgr,y=poprel,group=sex,fill=sex))+  
 geom\_bar(stat='identity')+  
 coord\_flip()+  
 labs(caption = "Fonte: United Nations. World Population Prospects 2019",  
 title='Chile',  
 subtitle='2020')+  
 scale\_y\_continuous(name='População (%)',  
 labels=label\_percent(big.mark = '.',decimal.mark = ','),  
 breaks = seq(-.1,.1,.04),limits=c(-.1,.1))+  
 scale\_x\_discrete(name='Grupo de Idade',breaks=levels(pop$idgr)[seq(1,19,2)])+  
 scale\_fill\_manual(name='Sexo',label=c('Mulheres','Homens'),  
 values=c('pink','blue'))+  
 th



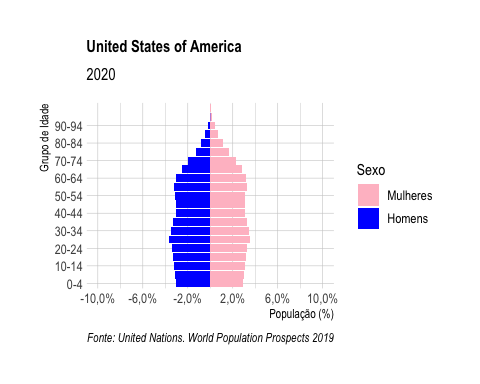
ggplot(data=brazil,aes(x=idgr,y=poprel,group=sex,fill=sex))+  
 geom\_bar(stat='identity')+  
 coord\_flip()+  
 labs(caption = "Fonte: United Nations. World Population Prospects 2019",  
 title='Brazil',  
 subtitle='2020')+  
 scale\_y\_continuous(name='População (%)',  
 labels=label\_percent(big.mark = '.',decimal.mark = ','),  
 breaks = seq(-.1,.1,.04),limits=c(-.1,.1))+  
 scale\_x\_discrete(name='Grupo de Idade',breaks=levels(pop$idgr)[seq(1,19,2)])+  
 scale\_fill\_manual(name='Sexo',label=c('Mulheres','Homens'),  
 values=c('pink','blue'))+  
 th



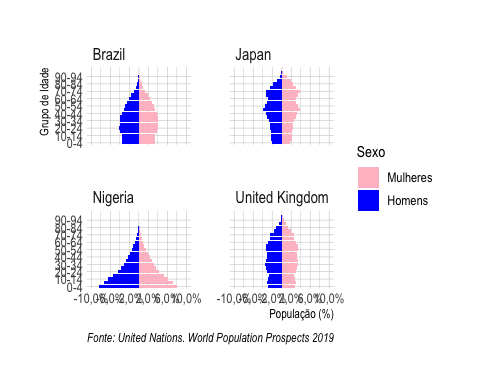
ggplot(data=nigeria,aes(x=idgr,y=poprel,group=sex,fill=sex))+  
 geom\_bar(stat='identity')+  
 coord\_flip()+  
 labs(caption = "Fonte: United Nations. World Population Prospects 2019",  
 title='Nigeria',  
 subtitle='2020')+  
 scale\_y\_continuous(name='População (%)',  
 labels=label\_percent(big.mark = '.',decimal.mark = ','),  
 breaks = seq(-.1,.1,.04),limits=c(-.1,.1))+  
 scale\_x\_discrete(name='Grupo de Idade',breaks=levels(pop$idgr)[seq(1,19,2)])+  
 scale\_fill\_manual(name='Sexo',label=c('Mulheres','Homens'),  
 values=c('pink','blue'))+  
 th



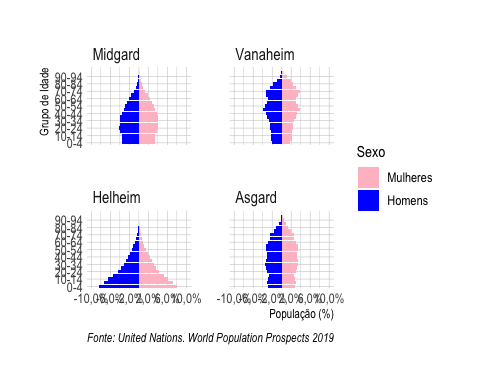
ggplot(data=usa,aes(x=idgr,y=poprel,group=sex,fill=sex))+  
 geom\_bar(stat='identity')+  
 coord\_flip()+  
 labs(caption = "Fonte: United Nations. World Population Prospects 2019",  
 title='United States of America',  
 subtitle='2020')+  
 scale\_y\_continuous(name='População (%)',  
 labels=label\_percent(big.mark = '.',decimal.mark = ','),  
 breaks = seq(-.1,.1,.04),limits=c(-.1,.1))+  
 scale\_x\_discrete(name='Grupo de Idade',breaks=levels(pop$idgr)[seq(1,19,2)])+  
 scale\_fill\_manual(name='Sexo',label=c('Mulheres','Homens'),  
 values=c('pink','blue'))+  
 th



paises<-pop%>%  
 filter(year==2020 &  
 (region=='Nigeria'| region=='Brazil'|  
 region=='Japan'| region == 'United Kingdom'))%>%  
 mutate(region=factor(region,  
 levels=c("Brazil","Japan","Nigeria","United Kingdom")))%>%#,  
 #labels=c('Midgard','Vanaheim','Helheim','Asgard')))%>%  
 group\_by(year,region)%>%  
 mutate(poptotal=sum(pop))%>%  
 ungroup()%>%  
 mutate(poprel=pop/poptotal)%>%  
 mutate(poprel=if\_else(sex=='Male',-poprel,poprel))  
  
th<-theme\_ipsum()+  
 theme(plot.title = element\_text(size = 12),  
 legend.text = element\_text(size = 10),  
 axis.text=element\_text(size = 10),  
 axis.text.x=element\_text(size = 10),  
 axis.text.y=element\_text(size = 10))  
  
ggplot(data=paises,aes(x=idgr,y=poprel,group=sex,fill=sex))+  
 geom\_bar(stat='identity')+  
 coord\_flip()+  
 facet\_wrap(~region)+  
 labs(caption = "Fonte: United Nations. World Population Prospects 2019")+  
 scale\_y\_continuous(name='População (%)',  
 labels=label\_percent(big.mark = '.',decimal.mark = ','),  
 breaks = seq(-.1,.1,.04),limits=c(-.1,.1))+  
 scale\_x\_discrete(name='Grupo de Idade',breaks=levels(pop$idgr)[seq(1,19,2)])+  
 scale\_fill\_manual(name='Sexo',label=c('Mulheres','Homens'),  
 values=c('pink','blue'))+  
 th

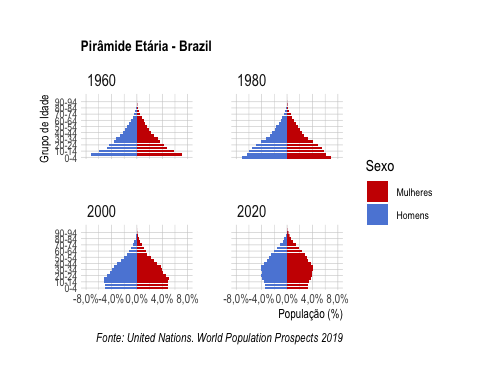


paises<-pop%>%  
 filter(year==2020 &  
 (region=='Nigeria'| region=='Brazil'|  
 region=='Japan'| region == 'United Kingdom'))%>%  
 mutate(region=factor(region,  
 levels=c("Brazil","Japan","Nigeria","United Kingdom"),  
 labels=c('Midgard','Vanaheim','Helheim','Asgard')))%>%  
 group\_by(year,region)%>%  
 mutate(poptotal=sum(pop))%>%  
 ungroup()%>%  
 mutate(poprel=pop/poptotal)%>%  
 mutate(poprel=if\_else(sex=='Male',-poprel,poprel))  
  
ggplot(data=paises,aes(x=idgr,y=poprel,group=sex,fill=sex))+  
 geom\_bar(stat='identity')+  
 coord\_flip()+  
 facet\_wrap(~region)+  
 labs(caption = "Fonte: United Nations. World Population Prospects 2019")+  
 scale\_y\_continuous(name='População (%)',  
 labels=label\_percent(big.mark = '.',decimal.mark = ','),  
 breaks = seq(-.1,.1,.04),limits=c(-.1,.1))+  
 scale\_x\_discrete(name='Grupo de Idade',breaks=levels(pop$idgr)[seq(1,19,2)])+  
 scale\_fill\_manual(name='Sexo',label=c('Mulheres','Homens'),  
 values=c('pink','blue'))+  
 th



th<-theme\_ipsum()+  
 theme(plot.title = element\_text(size = 11),  
 plot.subtitle = element\_text(size=10),  
 legend.text = element\_text(size = 8),  
 axis.text.x=element\_text(size = 9),  
 axis.text.y=element\_text(size = 8))  
  
temp<-pop%>%  
 filter(year %in% seq(1960,2020,20) & region=='Brazil')%>%  
 group\_by(year,region)%>%  
 mutate(poptotal=sum(pop))%>%  
 ungroup()%>%  
 mutate(poprel=pop/poptotal)%>%  
 mutate(poprel=if\_else(sex=='Male',-poprel,poprel))  
  
ggplot(data=temp,aes(x=idgr,y=poprel,group=sex,fill=sex))+  
 geom\_bar(stat='identity')+  
 coord\_flip()+  
 facet\_wrap(~year)+  
 labs(caption = "Fonte: United Nations. World Population Prospects 2019")+  
 ggtitle('Pirâmide Etária - Brazil')+  
 scale\_y\_continuous(name='População (%)',  
 labels=label\_percent(big.mark = '.',decimal.mark = ','),  
 breaks = seq(-.08,.08,.04),limits=c(-.08,.08))+  
 scale\_x\_discrete(name='Grupo de Idade',breaks=levels(pop$idgr)[seq(1,19,2)])+  
 scale\_fill\_startrek(name='Sexo',label=c('Mulheres','Homens'))+  
 th

## Warning: Removed 2 rows containing missing values (position\_stack).



th<-theme\_ipsum()+  
 theme(plot.title = element\_text(size = 24),  
 plot.subtitle = element\_text(size=18),  
 legend.text = element\_text(size = 20),  
 axis.text.x=element\_text(size = 12),  
 axis.text.y=element\_text(size = 12))  
  
  
for(p in c("Brazil","Japan","Nigeria","United Kingdom")){  
   
 df<-pop%>%  
 filter(year==2020 & region==p)%>%  
 group\_by(year)%>%  
 mutate(poptotal=sum(pop))%>%  
 ungroup()%>%  
 mutate(poprel=pop/poptotal)%>%  
 mutate(poprel=if\_else(sex=='Male',-poprel,poprel))  
   
 ggp<-ggplot(data=df,aes(x=idgr,y=poprel,group=sex,fill=sex))+  
 geom\_bar(stat='identity')+  
 coord\_flip()+  
 labs(caption = "Fonte: United Nations. World Population Prospects 2019")+  
 ggtitle('Pirâmide Etária',subtitle=paste(p,'- 2020'))+  
 scale\_y\_continuous(name='População (%)',  
 labels=label\_percent(big.mark = '.',decimal.mark = ','),  
 breaks = seq(-.1,.1,.04),limits=c(-.1,.1))+  
 scale\_x\_discrete(name='Grupo de Idade')+  
 scale\_fill\_startrek(name='Sexo',label=c('Mulheres','Homens'))+  
 th  
   
 pf<-paste0('./piramides/',p,'-2020.png')  
 ggsave(pf,plot=ggp,width = 20, height = 14, dpi = 300,   
 units = "cm", device='png')  
}

th<-theme\_ipsum()+  
 theme(plot.title = element\_text(size = 24),  
 plot.subtitle = element\_text(size=18),  
 legend.text = element\_text(size = 12),  
 axis.text.x=element\_text(size = 10),  
 axis.text.y=element\_text(size = 10))  
  
for(p in c("Brazil","Japan","Nigeria","United Kingdom")){  
   
 df<-pop%>%  
 filter(year %in% seq(1960,2020,20) & region==p)%>%  
 group\_by(year,region)%>%  
 mutate(poptotal=sum(pop))%>%  
 ungroup()%>%  
 mutate(poprel=pop/poptotal)%>%  
 mutate(poprel=if\_else(sex=='Male',-poprel,poprel))  
   
 ggp<-ggplot(data=df,aes(x=idgr,y=poprel,group=sex,fill=sex))+  
 geom\_bar(stat='identity')+  
 facet\_wrap(~year)+  
 coord\_flip()+  
 labs(caption = "Fonte: United Nations. World Population Prospects 2019")+  
 ggtitle('Pirâmide Etária',subtitle=paste(p,'- 1960 a 2020'))+  
 scale\_y\_continuous(name='População (%)',  
 labels=label\_percent(big.mark = '.',decimal.mark = ','),  
 breaks = seq(-.1,.1,.04),limits=c(-.1,.1))+  
 scale\_x\_discrete(name='Grupo de Idade',breaks=levels(pop$idgr)[seq(1,19,2)])+  
 scale\_fill\_startrek(name='Sexo',label=c('Mulheres','Homens'))+  
 th  
   
 pf<-paste0('./piramides/',p,'-1960a2020.png')  
 ggsave(pf,plot=ggp,width = 20, height = 14, dpi = 300, units = "cm", device='png')  
}

Population Division, United Nations. 2019. “2019 Revision of World Population Prospects.” New York. <https://population.un.org/wpp/>.