Relatório Final ICE 2021

Douglas Moreira Mangini e Pedro Henrique Borges da Silva

02/09/2021

Table of Contents

# Introdução

Este relatório descreve as tarefas realizadas durante a apuração do Índice de Cidades Empreendedoras 2021 (ICE 2021), que contempla os maiores municípios brasileiros. Sete determinantes integram o índice: Ambiente Regulatório; Infraestrutura; Mercado; Acesso a Capital; Inovação; Capital Humano; Cultura. Cada determinante é composto por um grupo de indicadores, segmentados em diferentes subdeterminantes. OS 49 indicadores, seus respectivos subdeterminantes e determinantes estão elencados na tabela a seguir, juntamente com os códigos que são usados para referir-se a eles nos documentos e arquivos do projeto:

Determinantes, subdeterminantes e indicadores com compõem o ICE 2021 com seus respectivos códigos de referência

|  |  |  |
| --- | --- | --- |
| Determinante | Subdeterminante | Indicador |
| d1 Ambiente Regulatório | sd11 Tempo de Processos | i111 Tempo de Viabilidade de Localização |
| d1 Ambiente Regulatório | sd11 Tempo de Processos | i112 Tempo de Registro, Cadastro e Viabilidade de Nome |
| d1 Ambiente Regulatório | sd11 Tempo de Processos | i113 Taxa de Congestionamento em Tribunais |
| d1 Ambiente Regulatório | sd12 Tributação | i121 Alíquota Interna do ICMS |
| d1 Ambiente Regulatório | sd12 Tributação | i121 Alíquota Interna do IPTU |
| d1 Ambiente Regulatório | sd12 Tributação | i123 Alíquota Interna do ISS |
| d1 Ambiente Regulatório | sd12 Tributação | i124 Qualidade de Gestão Fiscal |
| d1 Ambiente Regulatório | sd13 Complexidade Burocrática | i131 Simplicidade Tributária |
| d1 Ambiente Regulatório | sd13 Complexidade Burocrática | i132 CNDs Municipais |
| d1 Ambiente Regulatório | sd13 Complexidade Burocrática | i133 Atualização de Zoneamento |
| d2 Infraestrutura | sd21 Transporte Interurbano | i211 Conectividade Via Rodovias |
| d2 Infraestrutura | sd21 Transporte Interurbano | i212 Número de Decolagens por Ano |
| d2 Infraestrutura | sd21 Transporte Interurbano | i213 Distância ao Porto mais Próximo |
| d2 Infraestrutura | sd22 Condições Urbanas | i221 Acesso à Internet Rápida |
| d2 Infraestrutura | sd22 Condições Urbanas | i222 Preço Médio do m² |
| d2 Infraestrutura | sd22 Condições Urbanas | i223 Custo da Energia Elétrica |
| d2 Infraestrutura | sd22 Condições Urbanas | i224 Taxa de Homicídios |
| d3 Mercado | sd31 Desenvolvimento Econômico | i311 Índice de Desenvolvimento Humano |
| d3 Mercado | sd31 Desenvolvimento Econômico | i312 Crescimento Real Médio do PIB |
| d3 Mercado | sd31 Desenvolvimento Econômico | i313 Número de Empresas Exportadoras com Sede na Cidade |
| d3 Mercado | sd32 Clientes Potenciais | i321 PIB per capita |
| d3 Mercado | sd32 Clientes Potenciais | i322 Proporção entre Grandes/Médias e Médias/Pequenas Empresas |
| d3 Mercado | sd32 Clientes Potenciais | i323 Compras Públicas |
| d4 Acesso a Capital | sd41 Capital Disponível | i411 Operações de Crédito por Município |
| d4 Acesso a Capital | sd41 Capital Disponível | i412 Proporção Relativa de Capital de Risco |
| d4 Acesso a Capital | sd41 Capital Disponível | i413 Capital Poupado per capita |
| d5 Inovação | sd51 Inputs | i511 Proporção de Mestres e Doutores em C&T |
| d5 Inovação | sd51 Inputs | i512 Proporção de Funcionários em C&T |
| d5 Inovação | sd51 Inputs | i513 Média de Investimentos do BNDES e FINEP |
| d5 Inovação | sd51 Inputs | i514 Infraestrutura Tecnológica |
| d5 Inovação | sd51 Inputs | i515 Contratos de Concessão |
| d5 Inovação | sd52 Outputs | i521 Patentes |
| d5 Inovação | sd52 Outputs | i522 Tamanho da Indústria Inovadora |
| d5 Inovação | sd52 Outputs | i523 Tamanho da Economia Criativa |
| d5 Inovação | sd52 Outputs | i524 Tamanho das Empresas TIC |
| d6 Capital Humano | sd61 Acesso e Qualidade da Mão de Obra Básica | i611 Nota do Ideb |
| d6 Capital Humano | sd61 Acesso e Qualidade da Mão de Obra Básica | i612 Proporção de Adultos com pelo menos o Ensino Médio Completo |
| d6 Capital Humano | sd61 Acesso e Qualidade da Mão de Obra Básica | i613 Taxa Líquida de Matrícula no Ensino Médio |
| d6 Capital Humano | sd61 Acesso e Qualidade da Mão de Obra Básica | i614 Nota Média no ENEM |
| d6 Capital Humano | sd61 Acesso e Qualidade da Mão de Obra Básica | i615 Proporção de Matriculados no Ensino Técnico e Profissionalizante |
| d6 Capital Humano | sd62 Acesso e Qualidade da Mão de Obra Qualificada | i621 Proporção de Adultos com pelo menos os Ensino Superior Completo |
| d6 Capital Humano | sd62 Acesso e Qualidade da Mão de Obra Qualificada | i622 Proporção de Alunos Concluintes em Cursos de Alta Qualidade |
| d6 Capital Humano | sd62 Acesso e Qualidade da Mão de Obra Qualificada | i623 Custo Médio de Salários de Dirigentes |
| d7 Cultura | sd71 Iniciativa | i711 Pesquisas pelo Termo Empreendedor |
| d7 Cultura | sd71 Iniciativa | i712 Pesquisas pelo Termo MEI |
| d7 Cultura | sd72 Instituições | i721 Pesquisas por Sebrae |
| d7 Cultura | sd72 Instituições | i722 Pesquisas por Franquia |
| d7 Cultura | sd72 Instituições | i723 Pesquisas por SIMPLES Nacional |
| d7 Cultura | sd72 Instituições | i724 Pesquisas por Senac |

Os trabalhos foram guiados principalmente pelo relatório final do ICE 2020 (disponível neste [link](https://drive.google.com/drive/u/1/folders/1uNwTvmKw0HNvWDWffLtCsPzE5n7-I_a7)) e pelo documento “Produto 3v3.docx”, fornecido pelo orientador, Prof. Arnaldo Mauerberg, que contém a proposta definitiva de metodologia do Índice de Cidades Empreendedoras. É importante frisar, contudo, que esta versão do ICE contém mudanças com relação à versão de 2020. São elas:

* O determinante Cultura foi reformulado, sendo composto nesta edição pelos seguintes subdeterminantes e indicadores:
  + Subdeterminante Iniciativa: busca medir o grau de confiança e intenção em empreender dentro do município. É composto por dois indicadores
    - Indicador Pesquisas pelo Termo Empreendedor (i711);
    - Indicador Pesquisas pelo Termo MEI (i712);
  + Subdeterminante Instituições: busca medir a cultura empreendedora local. É composto pelos seguintes indicadores:
    - Indicador Pesquisas pelo Termo Sebrae (i721);
    - Indicador Pesquisas pelo Termo Franquia (i722);
    - Indicador Pesquisas pelo Termo SIMPLES Nacional (i723);
    - Indicador Pesquisas pelo Termo Senac (i724) A mudança no determinante Cultura se deve a suspensão das pesquisas Mind Miners que o compunham;
* O índice passa a incluir, além dos 100 municípios mais populosos do Brasil, os municípios da edição anterior que eventualmente tenho saído do ranking. Em 2021, o município de Santa Maria (RS) saiu e Marabá (PA) entrou no ranking. O ICE 2021, portanto, contará com 101 municípios.

Nesta edição, a manipulação das bases de dados e o cálculo dos indicadores foram feitos exclusivamente no R. Os scripts estão disponíveis no [Apêndice](#apend) deste relatório. Além disso, todos arquivos de dados e scripts estão disponíveis no drive do projeto, disponível neste [link](https://1drv.ms/u/s!AqMHjJQ8NQwSgcFH6a7-tqbNjUz-LA?e=XJhGL2). Por fim, os códigos em R também estão disponíveis em repositório [GitHub](https://github.com/manginidouglas/ice2021).

Os indicadores foram calculados de maneira paralela pelos bolsistas. Alguns scripts fazem referência a um dataframe chamado municode, enquanto outros se referem a top100\_mun\_cod. Ambos possuem a mesma informação: a lista dos 101 municípios que compõem o ICE. Os scripts que geram esses dataframes estão disponíveis no [Apêndice](municode).

# Determinante Ambiente regulatório

## Subdeterminante Tempo de processos

clique [aqui](#sd11_script) para ver o script

### Tempo de viabilidade de locação

Fonte: <https://estatistica.redesim.gov.br/tempos-abertura>.

Período: 2020

Cidades faltantes: São José do Rio Preto (SP), Jundiaí (SP), Maringá (PR), Anápolis (GO)

Comentário: A coluna de interesse é *qtde. hh viabilidade end*. Calculamos o tempo médio por município e registramos como indicador o inverso desse número. Atribuimos nota zero às cidades faltantes. Detalhes estão comentados no script em apêndice.

### Tempo de registro, cadastro, e viabilidade de nome

fonte: <https://estatistica.redesim.gov.br/tempos-abertura>.

Período: 2020

Cidades faltantes: São José do Rio Preto (SP), Jundiaí (SP), Maringá (PR), Anápolis (GO)

Comentário: As colunas de interesse são *qtde. hh viabilidade de nome*, *qtde. hh liberação dbe*, *qtde. hh horas deferimento*. Agrupamos por estado e somamos as 3 colunas. O indicador é o inverso da média simples por estado. Municípios do mesmo estado terão notas iguais.

### Taxa de congestionamento em tribunais

Fonte: <https://paineis.cnj.jus.br/QvAJAXZfc/opendoc.htm?document=qvw_l%2FPainelCNJ.qvw&host=QVS%40neodimio03&anonymous=true&sheet=shPDPrincipal>

Período: 2020

Comentário: Usamos os parâmetros de pesquisa: justiça = Justiça Estadual, campos agrupados = Sede Município e tipo de variável = novos, pendentes e baixados. A taxa líquida de congestionamento é definida[[1]](#footnote-29) como

. O indicador é o inverso da taxa de congestionamento.

## Subdeterminante Tributação

clique [aqui](#sd12_script) para ver o script

### Alíquota interna do ICMS

Fonte: <https://siconfi.tesouro.gov.br/siconfi/index.jsf>

Período icms: 2020 Período pib: 2018

Comentário: Usamos a tabela Receitas Orçamentárias (Anexo I-C). Trabalhamos apenas com receitas brutas realizadas e a conta 1.1.1.8.02.0.0. O indicador é o inverso do icms por unidade de pib municipal. Detalhes estão comentados no script em apêndice.

### Alíquota interna do IPTU

Fonte: <https://siconfi.tesouro.gov.br/siconfi/index.jsf>

Período icms: 2020  
Período pib = 2018

Cidades faltantes: Brasília (DF), Carapicuíba (SP)

Comentário: Usamos a tabela Receitas Orçamentárias (Anexo I-C).Trabalhamos apenas com receitas brutas realizadas e as contas 1.1.1.8.01.1.0 e 1.1.1.8.02.3.0. Detalhes estão comentados no script em apêndice.

### Alíquota interna do ISS

Fonte: <https://siconfi.tesouro.gov.br/siconfi/index.jsf>

Período icms: 2020  
Período pib = 2018

Cidades faltantes: Brasília (DF), Carapicuíba (SP), Uberaba (MG), São João de Meriti (RJ) Belford Roxo (RJ)

Comentário: Usamos a tabela Receitas Orçamentárias (Anexo I-C). Detalhes estão comentados no script em apêndice.

### Qualidade da Gestão Fiscal

Fonte: <https://www.firjan.com.br/ifgf/>

Período: 2018

Comentário: Não há novos dados desde a última edição do ICE, então usamos os valores passados.

## Subdeterminante Complexidade burocrática

clique [aqui](#sd13_script) para ver o script

### Simplicidade tributária

Fonte: <https://siconfi.tesouro.gov.br/siconfi/index.jsf>

Período: 2020

Cidades faltantes: Brasília (DF)

Comentário: Para filtrar as contas orçamentárias corretas, lemos o relatório do ice passado[[2]](#footnote-40) no R. O indicador é o produto dos índices de Herfindahl-Hirshmann (ihh) e de visibilidade (iv). O ihh é a soma dos quadrados da participação relativa do tributo na arrecadação total. A visibilidade é a participação relativa de uma soma de de tributos[[3]](#footnote-41) na arrecadação total. Detalhes no script em apêndice.

### CND’s municipais

Fonte: <https://www.ibge.gov.br/estatisticas/sociais/saude/10586-pesquisa-de-informacoes-basicas-municipais.html?=&t=downloads>

Período: 2019

Comentário: variável binária igual a 1 se município emite certidão negativa de débitos. Utilizamos a quarta aba da planilha, coluna MTIC1211. Detalhes no script em apêndice.

### Atualização de zoneamento

fonte: <https://www.ibge.gov.br/estatisticas/sociais/saude/10586-pesquisa-de-informacoes-basicas-municipais.html?=&t=downloads>

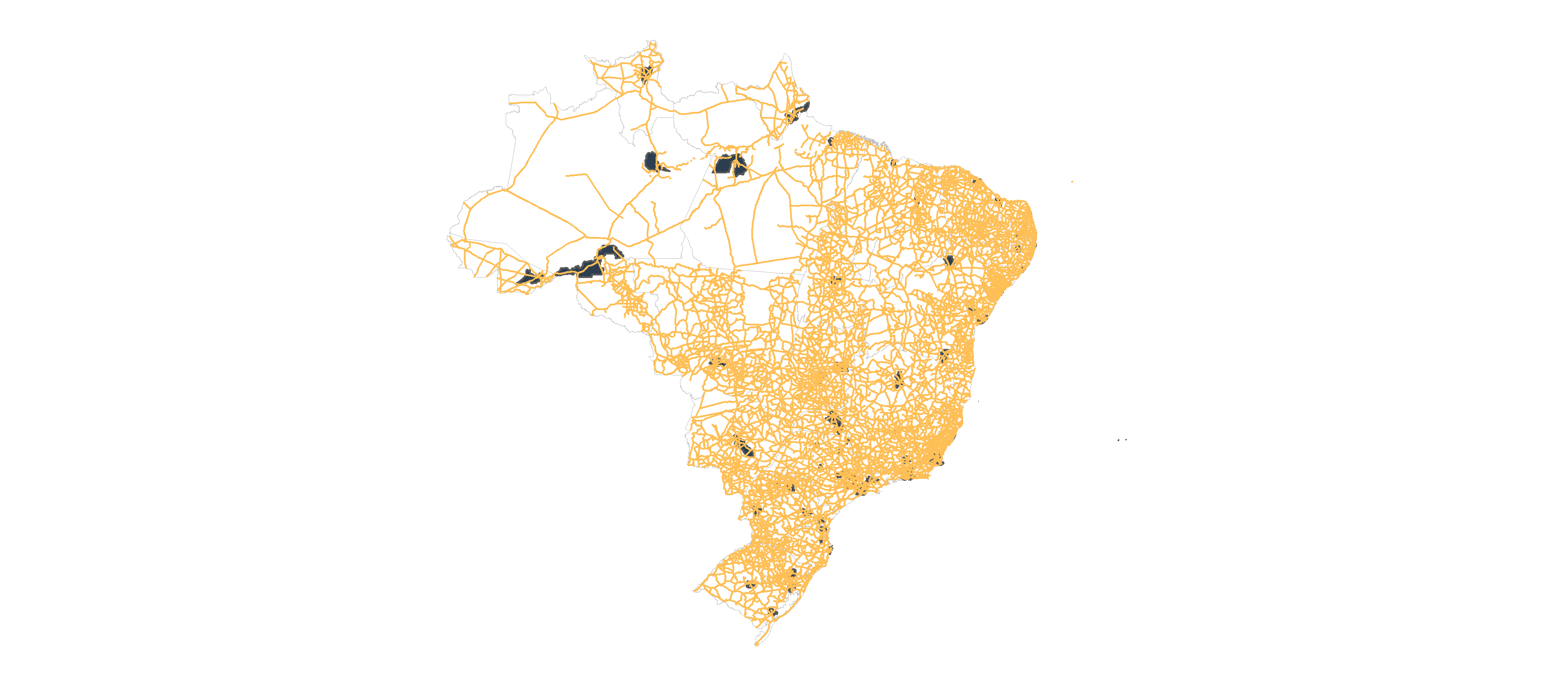
Período: 2018

Comentário: quantidade de anos desde que o município mudou a lei de zoneamento. O IBGE excluiu essa pergunta na pesquisa MUNIC 2019, então verificamos os sítios eletrônicos das prefeituras.

# Determinante Infraestrutura

## Subdeterminante Transporte Interurbano

Para os indicadores de rodovias, portos e aeroportos, importamos os dados georeferenciados de cada estrutura e fizemos operações geométricas para encontrar os indicadores. Por exemplo, encontrar as rodovias que passam por um município reduz-se a encontrar o número de interseções entre o polígono do município e a linha que descreve a estrada. Esses cálculos estão no script em apêndice. [ver script](#sd21_script)



Exemplo de rodovia

### Conectividade das rodovias

Fonte dos shape files das rodovias: <http://servicos.dnit.gov.br/vgeo/>

Período: 2021

Comentário: Consideramos apenas rodovias federais e estaduais. Construímos uma função que verifica se o município tem interseção com ao menos um trecho da rodovia. Nosso método implica incluir rodovias que estão próximas, mas não entram de fato no município. Consideramos este método melhor, pois reflete todas as opções de entrada e saída do território.

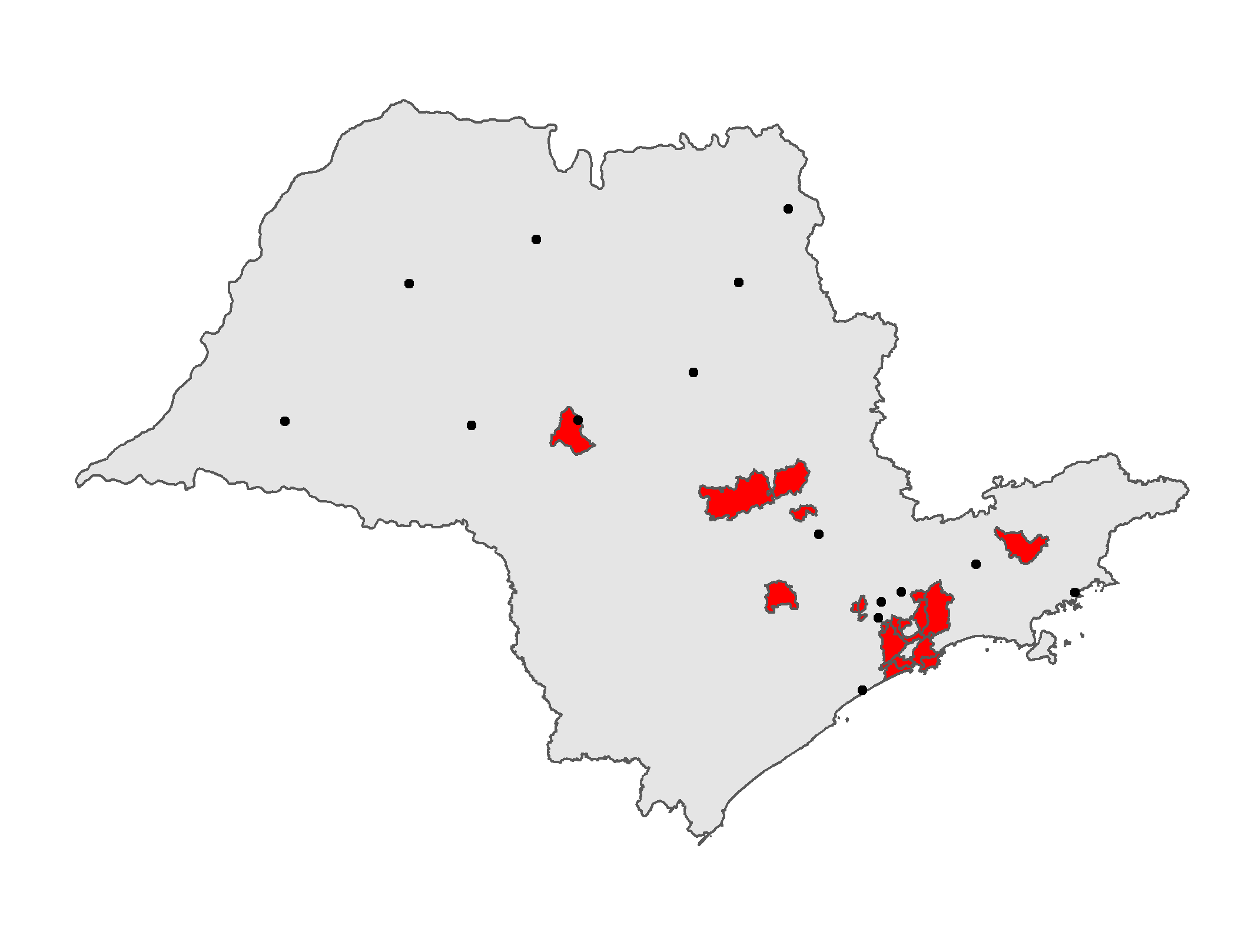
### Número de decolagens por ano

fonte dos dados de voos: <https://www.anac.gov.br/acesso-a-informacao/dados-abertos/areas-de-atuacao/voos-e-operacoes-aereas/dados-estatisticos-do-transporte-aereo>

Fonte dos Shape files dos aeroportos: <https://www.gov.br/infraestrutura/pt-br/assuntos/dados-de-transportes/bit/bitmodosmapas#mapaero>

Período: 2020

Comentário: Decolagens de voos regulares com origem no município. Faltam dados para metade dos municípios. Para completar os faltantes, atribuímos um número total de decolagens com base aeroporto mais próximo (distância euclidiana).



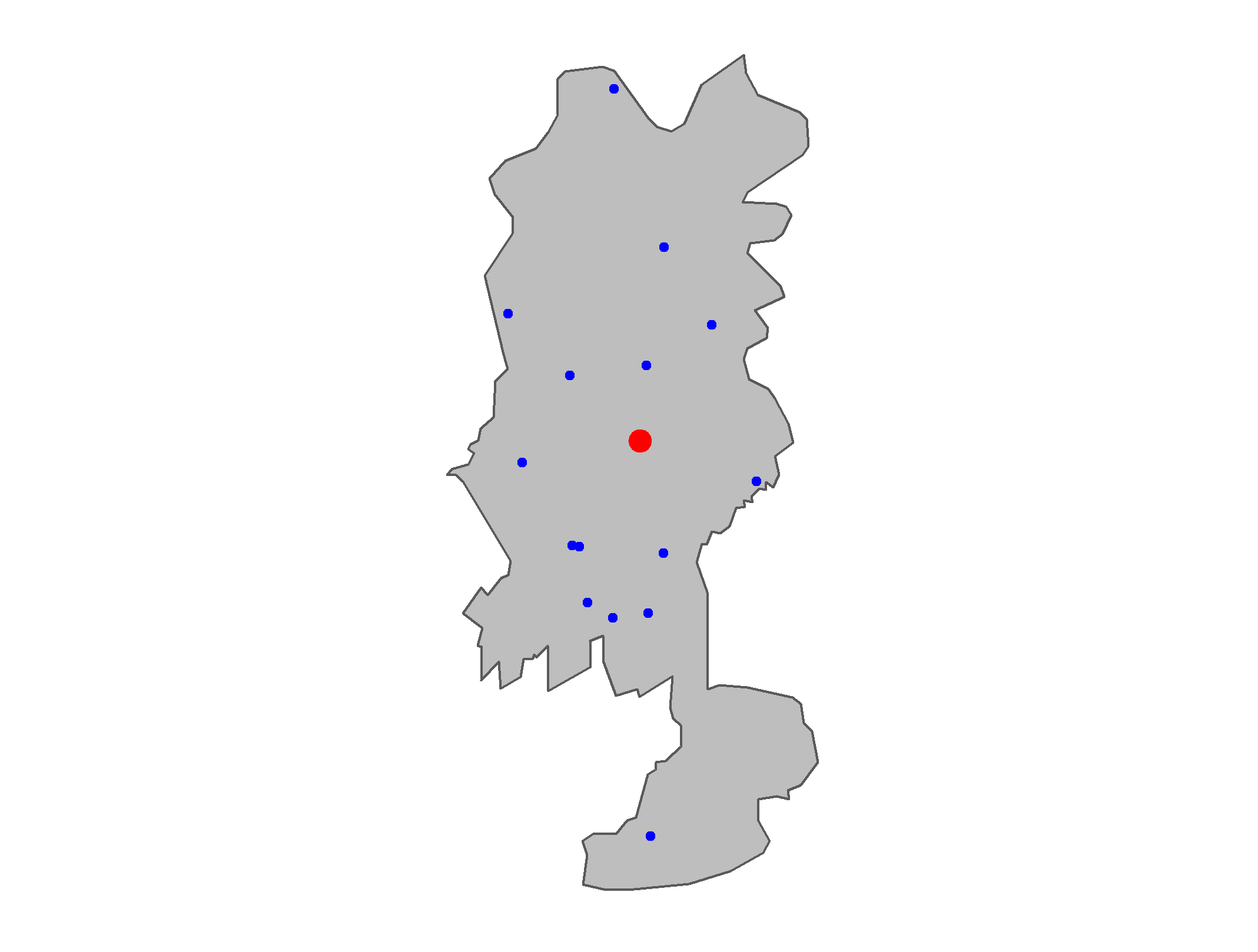
Exemplo: estado de SP e aeroportos mais próximos

### Distância ao porto mais próximo

Fonte dos shape files dos portos: <https://www.gov.br/infraestrutura/pt-br/assuntos/dados-de-transportes/bit/bitmodosmapas>

Período: 2020

Comentário: Distância do porto mais próximo ao município. Consideramos apenas portos públicos ou os fluvias do Amazonas. Tomamos o centro do polígono que representa o município como referência (ver figura).



Exemplo: centro de Ananindeua-PA e portos mais próximos

## Subdeterminante Condições Urbanas

clique [aqui](#sd22_script) para ver o script

### Acesso à internet rápida

Fonte: <https://dados.gov.br/dataset/dados-de-acessos-de-comunicacao-multimidia>

Período: 2020

Comentário: número de acessos à internet de alta velocidade (acima de 12Mbps) por habitante.

### Preço médio do m²

fonte: <https://www.zapimoveis.com.br/>

Período: 2021

Comentário: raspamos o site a procura de imóveis a venda. Extraímos preço e área útil e fizemos a média para cada município. Excluímos do cálculo os anúncios que implicavam um preço de metro quadrado menor do que 100 reais e maior do que 10.000 reais[[4]](#footnote-62). Para cada município, somamos o preço de todos os anúncios e dividimos pela soma de todas as áreas. Procuramos por 350 anúncios de cada município.

### Custo da energia elétrica

Fonte distribuidoras: <http://www2.aneel.gov.br/relatoriosrig/(S(fgy4psttnrfsam2x1s40fgib))/relatorio.aspx?folder=sfe&report=DistribuidoradecadaMunicipio>

Fonte das tarifas: <https://www.aneel.gov.br/ranking-das-tarifas>

Período: 2021

Comentário: Montamos manualmente a base de dados que indica qual distribuidora atende cada município.

### Taxa de Homicídios

fonte: <http://tabnet.datasus.gov.br/cgi/tabcgi.exe?sim/cnv/obt10br.def>

Período: 2019

Comentário: mortes causadas por agressão, ponderado pelo número estimado de habitantes em 2020. No site do datasus, selecionamos Conteúdo igual a *Óbitos p/ Ocorrência* e Grupo CID-10 igual a *agressões*. Os dados em formato excel estão na pasta de arquivos e o [script](#infra) em R para reproduzir o cálculo no apêndice deste documento.

# Determinante Mercados

## Subdeterminante Desenvolvimento Econômico

clique [aqui](#sd31_script) para ver o script

### índice de desenvolvimento humano

Fonte: <http://www.atlasbrasil.org.br/ranking>

Período: 2010

Comentário: Não há novos dados desde a última edição do índice. Apenas adicionamos as colunas código do município e sigla da UF.

### Crescimento médio real do PIB

Fonte: sidra-IBGE

Período: 2014 a 2018

Comentário: crescimento médio do pib municipal. Usamos os pacotes basedosdados e sidrar para encontrar o pib municipal e calcular o deflator do pib.

### Número de exportadoras sediadas no município

Fonte 1: Rais-IBGE  
Fonte 2: <https://www.gov.br/produtividade-e-comercio-exterior/pt-br/assuntos/comercio-exterior/estatisticas/empresas-brasileiras-exportadoras-e-importadoras>

Período: 2019

Comentário: Usamos o pacote basedosdados para acessar a RAIS e obter a quantidade de funcionarios por empresa. Dividimos o número de empresas exportadoras pelo total de empresas sediadas no município com pelo menos um funcionário.

## Subdeterminante Clientes Potenciais

clique [aqui](#sd32_script) para ver o script

### PIB per capita

Fonte: Sidra-IBGE

Período: 2018

Comentário: Acessamos os dados via base dos dados. Detalhes no script.

### Proporção de grandes empresas

Fonte: Rais-IBGE

Período: 2019

Comentário: Razão de duas proporções: empresas grandes por empresas médias e médias por pequenas. Acessamos os dados via base dos dados. O tamanho da empresa é dado pela variável qtde\_vinculos\_ativos. Empresas pequenas têm entre 10 e 49 funcionários; médias entre 50 e 249; grandes, acima de 250.

### Indicador Compras Públicas (i323)

Este indicador refere-se ao total de despesas e investimentos feitos pela prefeitura dividido pelo número de empresas com pelo menos um funcionário no município. Os seguintes passos descrevem o procedimento adotado para se chegar ao indicador:

1. Obter valor total de despesas e investimentos feitos pela prefeitura dos municípios do ICE

* Acessar site do [Sistema de Informações Contábeis e Fiscais do Setor Público Brasileiro (Siconfi) e acessar “contas anuais” na seção “consultar finbra” em “consultas”;
* Selecionar como “”exercício” o ano de 2020, ano mais recente disponível, como “escopo” selecionar Município, e selecionar a tabela “Despesas Orçamentárias (Anexo I-D)”;
* Observações: na data em de acesso à tabela (26/08/2021), ainda faltavam dados de Belém (PA). Foi escolhido manter os dados de 2020 uma vez que o documento “Produto 3v3” sugere selecionar o ano anterior somente quando os dados faltantes do ano mais recente forem “muitos”.
* Os dados de Brasília foram coletados à parte utilizando como escopo “Estados/DF” na seleção de tabela no site da Siconfi, já que, para os dados de despesas orçamentárias, Brasília é tratada como um estado e não um município;

1. Obter para cada município o número total de empresas com pelo menos um funcionário.

* Esse dado também é utilizado para o indicador Número de Empresas Exportadoras com Sede na Cidade (i313). O procedimento adotado para obtenção dos dados foi o mesmo .Foram usados os dados mais recentes disponíveis na RAIS, isto é, do ano de 2019

1. Calcular o indicador para cada município de acordo com a seguinte fórmula:
2. O tratamento das bases de dados e o cálculo do índice foram feitos usando o R. O script encontra-se no [Apêndice](#i323_script) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

# Determinante Acesso a Capital

## Subdeterminante Capital Disponível

### Indicador Operações de Crédito por Município (i411)

Este indicador refere-se ao valor, em reais, das operações de crédito no município para pessoas físicas e jurídicas, dos bancos múltiplos com carteira comercial, dividido pelo PIB total do município. Os seguintes passos descrevem o procedimento adotado para se chegar ao indicador:

1. Obter valor em reais das operações de crédito para pessoas físicas e jurídicas por município;

* Acessar site do [Banco Central](http://www4.bcb.gov.br/fis/cosif/estban.asp) e baixar arquivo “Estatística Bancária Mensal por município” referente ao mês de dezembro de 2020 (ano mais recente com dados disponíveis) Na planilha, é utilizada apenas a coluna chamada “VERBETE\_160\_OPERACOES\_DE\_CREDITO”;

1. Obter PIB dos municípios do ICE a preços correntes;

* Acessar o site do [SIDRA-IBGE](https://sidra.ibge.gov.br/tabela/5938) e baixar a planilha número 5938, escolhendo a opção “Produto Interno Bruto a Preços Correntes” e o nível territorial “município”. Os dados selecionados se referem ao ano mais recente com dados disponíveis, 2018;

1. Calcular o indicador para cada município de acordo com a seguinte fórmula:
2. O tratamento das bases de dados e o cálculo do índice foram feitos usando o R. O script encontra-se no [Apêndice](#i411_script) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

### Indicador Proporção Relativa de Capital de Risco (i412)

Este indicador consiste na soma de todos os investimentos de risco, em reais, recebidos por empresas do município no último ano dividido pelo PIB do município. Os seguintes passos descrevem o procedimento adotado para se chegar ao indicador:

1. Obter valor de capital de risco levantados por empresas dos municípios do ICE

* Acessar a base de dados [Crunchbase](https://www.crunchbase.com/) para obter os dados. Na área de busca, selecionar aba “Companies” e aplicar seguintes filtros:
  + Em “Financials” marcar “Past Year” na opção “Last Funding Date”;
  + Em “Overview”, digitar nome do município na área “Headquarters Location”. Muitos municípios possuem erros nessa base de dados quanto a sua localização. Brasília, por exemplo, está cadastrada como uma cidade do Rio Grande do Norte. Contudo, esses erros não comprometeram o levantamento dos dados;
* Buscar municípios manualmente um a um e obter o valor de capital de risco levantado pelas empresas do município no último ano. Foi montada uma tabela no excel com os dados de capital de risco por município;

1. Obter taxa de câmbio média do último ano de moedas diferente do real presentes na base de dados construída no passo 1 e converter valores para real;

* Obter dados de taxa de câmbio média no site [OFX](https://www.ofx.com/en-au/forex-news/historical-exchange-rates/yearly-average-rates/) ou outro similar. Transformar todos valores para real.

1. Obter PIB municipal;

* Esse dado foi obtido no indicador [PIB per capita](#sd32). Aqui, usamos o arquivo “pib\_mun.csv” que foi obtido exportando o PIB dos municípios usando o script em R do indicador PIB per capita;

1. Calcular o indicador para cada município de acordo com a seguinte fórmula:

O tratamento das bases de dados e o cálculo do índice foram feitos usando o R. O script encontra-se no [Apêndice](#i412_script) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

### Indicador Capital Poupado per capita (i413)

Este indicador refere-se ao valor, em reais, das operações de depósitos em poupança e depósitos a prazo feitos no município, de pessoas físicas e jurídicas, dividido pelo número de habitantes estimado do município. Os seguintes passos descrevem o procedimento adotado para se chegar ao indicador:

1. Obter dados de valor médio mensal de depósitos em poupança e de longo prazo por município;

* Acessar site do [Banco Central](https://www.bcb.gov.br/estatisticas/estatisticabancariamunicipios) e baixar arquivo “Estatística Bancária Mensal por município” referente ao mês de dezembro de 2020 (último disponível).Na planilha, utilizar as colunas “VERBETE\_420\_DEPOSITOS\_DE\_POUPANCA” e “VERBETE\_432\_DEPOSITOS\_A\_PRAZO”;

1. Obter estimativa populacional dos municípios do ICE;

* Os dados mais recentes são de 2021. Acessar site do [IBGE](https://www.ibge.gov.br/estatisticas/sociais/populacao/9103-estimativas-de-populacao.html?=&t=resultados) e baixar tabela;

1. Calcular indicador para cada município de acordo com a seguinte fórmula:
2. O tratamento das bases de dados e o cálculo do índice foram feitos usando o R. O script encontra-se no [Apêndice](#i413_script) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

# Determinante Inovação

## Subdeterminante Inputs

### Indicador Proporção de Mestres e Doutores em C&T (i511)

Este indicador refere-se ao número de mestres e doutores titulados nas cidades nas áreas de ciências, tecnologia, engenharias e matemática dividido pelo número total de empresas com pelo menos um funcionário no município. Os seguintes passos descrevem o procedimento adotado para se chegar ao indicador:

1. Obter para cada município o número de mestres e doutores titulados nas áreas de ciências, tecnologia, engenharias e matemática;

* Acessar site da [CAPES](https://dadosabertos.capes.gov.br/) e baixar conjunto de dados “[2017 a 2019] Discentes da Pós-Graduação stricto sensu do Brasil”.

1. Obter para cada município o número total de empresas com pelo menos um funcionário.

* Este dado também é utilizado para o indicador Número de Empresas Exportadoras com Sede na Cidade (i313). O procedimento adotado para obtenção dos dados foi o mesmo. Foram usados os dados mais recentes disponíveis na RAIS, isto é, do ano de 2019;

1. Calcular o indicador para cada município do ICE de acordo com a seguinte fórmula:
2. O tratamento das bases de dados e o cálculo do índice foram feitos usando o R. O script encontra-se no [Apêndice](#i511_script) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

### Indicador Proporção de Funcionários em C&T (i512)

Este indicador refere-se ao número de funcionários do município que trabalham nas áreas de ciência, tecnologia, engenharia, matemática (critério feito pela Classificação Brasileira de Ocupações - CBO) dividido pelo número total de trabalhadores do município. Os seguintes passos descrevem o procedimento adotado para se chegar ao indicador:

1. Obter para cada município o número de funcionários que trabalham nas áreas de ciência e tecnologia, engenharia, matemática;

* Fonte: RAIS, Ministério do Trabalho. As informações encontram-se na tabela de vínculos da RAIS. Os últimos dados disponíveis são do ano de 2019;
* Os dados são acessados por meio do *data lake* público [Base dos Dados](https://basedosdados.org/);
* As ocupações relacionadas à ciência e tecnologia segundo a classificação da CBO são as seguintes:
  + Subgrupo Principal 73: TRABALHADORES DA FABRICACAO E INSTALACAO ELETROELETRONICA;
  + Subgrupo Principal 20 “PESQUISADORES E PROFISSIONAIS POLICIENTIFICOS”;
  + Subgrupo Principal 21:“PROFISSIONAIS DAS CIÊNCIAS EXATAS, FÍSICAS E DA ENGENHARIA”
  + Subgrupo Principal 31 :“TECNICOS DE NIVEL MEDIO DAS CIENCIAS FISICAS, QUIMICAS, ENGENHARIA E AFINS”;
  + Subgrupo Principal 30: “TECNICOS POLIVALENTES”:
  + Subgrupo Principal 39: “OUTROS TECNICOS DE NIVEL MEDIO”; e,
  + Subgrupo Principal 72: “TRABALHADORES DA TRANSFORMAÇÃO DE METAIS E DE COMPOSITOS”;

1. Obter número total de trabalhadores no município;

* Fonte: RAIS, Ministério do Trabalho. As informações encontram-se na tabela de vínculos da RAIS. Os últimos dados disponíveis são do ano de 2019;
* Os dados são acessados por meio do *data lake* público [Base dos Dados](https://basedosdados.org/);

1. Calcular indicador de acordo com a seguinte fórmula:

O tratamento das bases de dados e o cálculo do índice foram feitos usando o R. O script encontra-se no [Apêndice](#i512_script) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

### Indicador Média de Investimentos do BNDES e da FINEP (i513)

Este indicador refere-se ao número total de investimentos do BNDES e da FINEP dividido pelo total de empresas com pelo menos um funcionário no ano corrente. Os seguintes passos descrevem o procedimento adotado para se chegar ao indicador:

1. Obter dados de investimentos do BNDES nos municípios do ICE

* Acessar a [central de downloads do site do BNDES](https://www.bndes.gov.br/wps/portal/site/home/transparencia/centraldedownloads) e selecionar “Operações contratadas na forma direta e indireta não automática (2002 a 30.06.2021)”;

1. Obter dados de investimentos da FINEP nos municípios do ICE;

* Acessar site da [FINEP](http://www.finep.gov.br/transparencia-finep/projetos-contratados-e-valores-liberados), na seção “Projetos contratados e valores liberados” e selecionar a planilha “Projetos Contratados”;

1. Obter para cada município o número total de empresas com pelo menos um funcionário;

* Este dado também é utilizado para o indicador Número de Empresas Exportadoras com Sede na Cidade (i313). O procedimento adotado para obtenção dos dados foi o mesmo. Foram usados os dados mais recentes disponíveis na RAIS, isto é, do ano de 2019;

1. Calcular o indicador para cada município de acordo com a fórmula:

O tratamento das bases de dados e o cálculo do índice foram feitos usando o R. O script encontra-se no [Apêndice](#i513_script) deste relatório. O script e os arquivos de dados com os resultados dos cálculos estão disponíveis no drive do projeto.

### Indicador Infraestrutura Tecnológica (i514)

A variável a medir infraestrutura tecnológica é uma dummy igual a um se o município foi listado no projeto Parques Tecnológicos como possuindo parque, e zero caso contrário. Como o documento usado como fonte para obter esses dados é o mesmo estudo utilizado na edição de 2020 do ICE e a única alteração na lista das 100 cidades mais populosas é a adição de Marabá (PA), o indicador será o mesmo do ano passado, com a adição do dado referente a Marabá. Então, o seguinte procedimento foi adotado para calcular o indicador:

1. Importar tabela com indicador i514 do ICE 2020;

* Baixar a planilha “Indicador Infraestrutura Tecnológica.xlsx” no [drive do projeto ICE 2020](https://drive.google.com/drive/u/0/folders/1zoR8VDP-yLox28dgplZSDFOfeXtGTrm1) ;

1. Acessar o estudo [“Indicadores de Parques Tecnológicos”](https://gestiona.com.br/wp-content/uploads/2019/10/MCTIC-UnB-ParquesTecnologicos-Portugues-final.pdf) que serve de base para o indicador e consultar se existem parques tecnológicos em operação em Marabá. Adicionar linha com a informação sobre o município Marabá na base de dados mencionada acima;

* Para este município a variável é igual a zero;

O tratamento da base de dados e o cálculo do índice foram feitos usando o R. O script encontra-se no [Apêndice](#i514_script) deste relatório. O script e os arquivos de dados com os resultados dos cálculos estão disponíveis no drive do projeto.

### Indicador Contratos de Concessão (i515)

Este indicador refere-se ao número total de contratos de Propriedade Intelectual depositados, dividido pelo total de empresas com pelo menos um funcionário, e multiplicado por mil (pois o dado é para cada mil empresas).

Para este indicador é importante ressaltar que os dados disponíveis de propriedade intelectual depositados a nível municipal não mudaram desde a edição passada do ICE. Os dados disponíveis são referentes aos anos de 2016 e 2017. Portanto, este indicador será o mesmo do ano passado, com a adição do dado referente ao município de Marabá (PA), que entrou no ranking dos municípios mais populosos. Para detalhes de como foi calculado, veja relatório final do ICE 2020. Ainda sim, utilizou-se o R criar um arquivo csv com os dados do indicador de 2020 adicionados ao dado de Marabá (PA). O script encontra-se no [Apêndice](#i515_script) deste relatório. O script e os arquivos de dados com os resultados dos cálculos estão disponíveis no drive do projeto.

## Subdeterminante Outputs

### Indicador Patentes (i521)

Para a apuração deste indicador, primeiro devem ser somados o total de patentes de inovação, de adição de inovação e de modelos de utilidades nos dois últimos anos disponíveis por município. Então, este total deve ser dividido pelo número de empresas com pelo menos um funcionário em cada um dos 100 municípios. Os seguintes passos descrevem o procedimento adotado para se chegar ao indicador:

1. Obter o total de patentes de inovação (PI), de adição de inovação (CA) e de modelos de utilidades (MU) nos últimos dois anos em cada município;

* Baixar os dados no site do [Instituto Nacional da Propriedade Industrial (INPI)](https://www.gov.br/inpi/pt-br/central-de-conteudo/estatisticas), na seção “estatísticas”, e “Indicadores de Propriedade Industrial”;

1. Obter para cada município o número total de empresas com pelo menos um funcionário.

* Este dado também é utilizado para o indicador Número de Empresas Exportadoras com Sede na Cidade (i313). O procedimento adotado para obtenção dos dados foi o mesmo. Foram usados os dados mais recentes disponíveis na RAIS, isto é, do ano de 2019;

1. Calcular o indicador para cada município do ICE de acordo com a seguinte fórmula:

O tratamento das bases de dados e o cálculo do índice foram feitos usando o R. O script encontra-se no [Apêndice](#i521_script) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

### Indicador Tamanho da Indústria Inovadora (i522)

Este indicador refere-se à proporção de empresas de indústria inovadora (classes da CNAE 2.0) em relação ao número total de empresas com ao menos um funcionário. Os seguintes passos descrevem o procedimento adotado para se chegar ao indicador:

1. Obter número de empresas de indústria inovadora;

* Obter, a partir da planilha “Indicador Tamanho da Indústria Inovadora.xlsx” no drive do projeto ICE 2020, a lista de classes associadas a Indústria Inovadora. São 86 classes que estão listadas no [Apêndice](#inovador) deste relatório. Usar a tabela “CNAE20\_EstruturaDetalhada.xls” baixada no site do IBGE para associar à cada denominação de classe seu código numérico;
* Obter dados de quantidade de empresas de indústria inovadora por meio da tabela de estabelecimentos da RAIS. Selecionar as classes CNAE 2.0 que compõem a indústria inovadora. A RAIS é acessada por meio do *data lake* público [Base dos Dados](https://basedosdados.org/);

1. Obter para cada município o número total de empresas com pelo menos um funcionário;

* Este dado também é utilizado para o indicador Número de Empresas Exportadoras com Sede na Cidade (i313). O procedimento adotado para obtenção dos dados foi o mesmo. Foram usados os dados mais recentes disponíveis na RAIS, isto é, do ano de 2019;

1. Calcular indicador de acordo com a seguinte fórmula:

O tratamento das bases de dados e o cálculo do índice foram feitos usando o R. O script encontra-se no [Apêndice](#i522_script) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

### Indicador Tamanho da Economia Criativa (i523)

Este indicador consiste na razão entre empresas de economia criativa selecionadas a partir das classes da CNAE 2.0 no município, e o número total de empresas com ao menos um funcionário no município. Os seguintes passos descrevem o procedimento adotado para se chegar ao indicador:

1. Obter número de empresas de economia criativa nos municípios do ICE

* Obter, a partir da planilha “Indicador Tamanho da Economia Criativa.xlsx” no drive do projeto ICE 2020, a lista de classes associadas a Economia Criativa. São 37 classes que estão listadas no Apêndice deste relatório.
* Usar a tabela “CNAE20\_EstruturaDetalhada.xls” baixada no site do IBGE para associar à cada denominação de classe seu código numérico.
* Obter dados de quantidade de empresas de economia criativa por meio da tabela de estabelecimentos da RAIS. Selecionar as classes CNAE 2.0 que compõem a indústria inovadora. A RAIS é acessada por meio do *data lake* público [Base dos Dados](https://basedosdados.org/);

1. Obter para cada município o número total de empresas com pelo menos um funcionário;

* Este dado também é utilizado para o indicador Número de Empresas Exportadoras com Sede na Cidade (i313). O procedimento adotado para obtenção dos dados foi o mesmo. Foram usados os dados mais recentes disponíveis na RAIS, isto é, do ano de 2019;

1. Calcular indicador para cada município de acordo com a seguinte fórmula:

O tratamento das bases de dados e o cálculo do índice foram feitos usando o R. O script encontra-se no [Apêndice](#i523_script) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

### Indicador Tamanho das Empresas TIC (i524)

Este indicador refere-se à razão entre o número empresas dos setores de tecnologia da informação e comunicação (TIC) no município, selecionadas a partir das classes da CNAE 2.0, e o número total de empresas com ao menos um funcionário no município. Os seguintes passos descrevem o procedimento adotado para se chegar ao indicador:

1. Obter número de empresas TIC nos municípios de interesse;

* Obter, a partir da planilha “Indicador Tamanho das Empresas TIC.xlsx” no drive do projeto ICE 2020, a lista de classes associadas a Economia Criativa. São 24 classes que estão listadas no Apêndice deste relatório;
* Usar a tabela “CNAE20\_EstruturaDetalhada.xls” baixada no site do IBGE para associar à cada denominação de classe seu código numérico;
* Obter dados de quantidade de empresas TIC por meio da tabela de estabelecimentos da RAIS. Selecionar as classes CNAE 2.0 que caracterizam empresas TIC. A RAIS é acessada por meio do data lake público Base dos Dados;

1. Obter para cada município o número total de empresas com pelo menos um funcionário;

* Este dado também é utilizado para o indicador Número de Empresas Exportadoras com Sede na Cidade (i313). O procedimento adotado para obtenção dos dados foi o mesmo. Foram usados os dados mais recentes disponíveis na RAIS, isto é, do ano de 2019;

1. Calcular indicador para cada município de acordo com a seguinte fórmula:

O tratamento das bases de dados e o cálculo do índice foram feitos usando o R. O script encontra-se no [Apêndice](#i524_script) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

# Determinante Capital Humano

## Subdeterminante Acesso e Qualidade da Mão de Obra Básica

### Indicador Nota do Ideb (i611)

Este indicador trata-se do Índice final do IDEB de cada município, que é calculado com base no desempenho escolar dos alunos dos anos finais do ensino fundamental nas escolas públicas dos municípios. Os seguintes passos descrevem o procedimento adotado para se chegar ao indicador:

1. Obter índice final Ideb de cada município do ICE;

* Os dados são fornecidos pelo INEP no site do [IDEB](http://ideb.inep.gov.br/). Seleciona-se “Município”, então, em Rede/Dependência Administrativa, é selecionada a opção “Pública (Federal, Estadual e Municipal). Em série, seleciona-se “8ª série / 9º ano”. A consulta é feita por cidade, selecionando-a nos campos UF e município;
* Os dados mais recentes, de 2019, foram, então, reunidos na planilha “i611 base de dados.xlxs”, disponível no drive de arquivos do projeto.

O tratamento das bases de dados e o cálculo do índice foram feitos usando o R. O script encontra-se no [Apêndice](#i611_script) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

### Indicador Proporção de adultos com pelo menos o Ensino Médio completo (i612)

Este indicador é a média simples de dois dados:

1. A razão entre o número de inscritos no ENEM no município que declararam ter pai com pelo menos ensino médio completo e total de inscritos no município;
2. A razão entre o número de inscritos no ENEM no município que declararam ter mãe com pelo menos ensino médio completo e total de inscritos no município.

A base de dados deste indicador são os microdados do ENEM fornecidos pelo [INEP](http://portal.inep.gov.br/microdados). No website, selecionou-se a base de dados mais recente, referente à edição de 2019. Essa base de dados contém os dados para este indicador e também para os indicadores Nota Média no Enem (i614) e Proporção de Adultos com pelo menos o Ensino Superior Completo (i621).

O tratamento das bases de dados e o cálculo do índice foram feitos usando o R. O script encontra-se no [Apêndice](#i612_script) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

### Indicador Taxa Líquida de Matrícula no Ensino Médio (i613)

Este indicador refere-se ao número de alunos entre 15 e 17 anos matriculados no ensino médio dividido pela população estimada de jovens com idade entre 15 e 17 anos. Então, a estimativa da população entre 15 e 17 anos é calculada com base nos dados do censo de 2010, aumentada pelo percentual de crescimento da população total daquele município até o ano em vigor. Por exemplo, se a população total de São Paulo cresceu 10% entre 2010 e 2021, a população de jovens entre 15 e 17 anos de 2010 é acrescida em 10%. Os passos tomados para chegar ao índice estão descritos a seguir:

1. Obter número de alunos entre 15 e 17 anos matriculados no EM em cada município;

* Os dados de alunos matriculados estão disponíveis na base de microdados do censo escolar do INEP;

1. Obter população entre 15 e 17 anos no município;

* Dados disponíveis no Censo 2010, obtidos no site do IBGE, baixando os dados do censo por município e idade;

1. Obter população total do município em 2010;

* Dados disponíveis no Censo 2010. Contudo, os dados foram obtidos por meio do *data lake* público Base dos Dados;

1. Obter estimativa populacional do município em 2021;

* Dados obtidos no site do IBGE;

1. Calcular fator de crescimento populacional do município entre 2010 e 2021 a partir dos dados obtidos;
2. Calcular população estimada do município em 2020 de pessoas entre 15 e 17;
3. Calcular indicador para cada município por meio da seguinte fórmula:

O tratamento das bases de dados e o cálculo do índice foram feitos usando o R. O script encontra-se no [Apêndice](#i613_script) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

### Indicador Nota média no ENEM (i614)

Este indicador consiste na nota média no Exame Nacional do Ensino Médio (ENEM) em todas as provas dos inscritos no município. Os passos tomados para chegar ao indicador estão descritos a seguir:

1. Obter microdados do ENEM para os 100 municípios mais populosos do Brasil;

* Acessar site do [INEP](http://portal.inep.gov.br/microdados) selecionar a base de dados mais recente, referente à edição de 2019. Essa base de dados contém os dados para este indicador e também para os indicadores Nota Média no Enem (i614) e Proporção de Adultos com pelo menos o Ensino Superior Completo (i621).

1. Calcular a nota média de cada aluno considerando as 5 provas do ENEM;
2. Calcular a nota média de cada município, considerando as notas médias de todos os inscritos do município.

O tratamento das bases de dados e o cálculo do índice foram feitos usando o R. O script encontra-se no [Apêndice](#i614_script) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

### Indicador Proporção de Matriculados no Ensino Técnico e Profissionalizante (i615)

Este indicador refere-se ao número total de alunos inscritos no ensino técnico, dividido pela população estimada com mais de 15 anos. Então, a estimativa da população acima de 15 anos é calculada com base nos dados do censo de 2010, aumentada pelo percentual de crescimento da população total daquele município até o ano em vigor. Por exemplo, se a população total de São Paulo cresceu 10% entre 2010 e 2021, a população acima de 15 anos de 2010 é acrescida em 10%. Os passos tomados para chegar ao índice estão descritos a seguir:

1. Obter número de alunos matriculados no ensino técnico em cada município;

* Os dados de alunos matriculados estão disponíveis na base de microdados do censo escolar do INEP;

1. Obter população acima de 15 anos no município;

* Dados disponíveis no Censo 2010, obtidos no site do IBGE, baixando os dados do censo por município e idade;

1. Obter população total do município em 2010;

* Dados disponíveis no Censo 2010. Contudo, os dados foram obtidos por meio do *data lake* público Base dos Dados;

1. Obter estimativa populacional do município em 2021;

* Dados obtidos no site do IBGE;

1. Calcular fator de crescimento populacional do município entre 2010 e 2021 a partir dos dados obtidos;
2. Calcular população estimada do município em 2021 de pessoas com mais de 15 anos;
3. Calcular indicador para cada município por meio da seguinte fórmula:

O tratamento das bases de dados e o cálculo do índice foram feitos usando o R. O script encontra-se no [Apêndice](#i615_script) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

## Subdeterminante Acesso e Qualidade da Mão de Obra Qualificada

### Indicador Proporção de Adultos com pelo Menos Ensino Superior Completo (i621)

Este indicador é a média simples de dois dados:

1. A razão entre o número de inscritos no ENEM no município que declararam ter pai com pelo menos ensino superior completo e total de inscritos no município;
2. A razão entre o número de inscritos no ENEM no município que declararam ter mãe com pelo menos ensino superior completo e total de inscritos no município.

A base de dados deste indicador são os microdados do ENEM fornecidos pelo [INEP](http://portal.inep.gov.br/microdados). No website, selecionou-se a base de dados mais recente, referente à edição de 2019. Essa base de dados contém os dados para este indicador e também para os indicadores Nota Média no Enem (i614) e Proporção de Adultos com pelo menos o Ensino Superior Completo (i621).

O tratamento das bases de dados e o cálculo do índice foram feitos usando o R. O script encontra-se no [Apêndice](#i621_script) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

### Indicador Proporção de Alunos Concluintes em Cursos de Alta Qualidade (i622)

Este indicador refere-se ao total de alunos concluintes em cursos de alta qualidade, que são reconhecidos com as notas 4 e 5 no ENADE, nos últimos três anos, dividido pelo total de alunos concluintes em cursos de graduação avaliados pelo ENADE. São coletadas as informações de três anos de ENADE, o que faz com que o indicador abranja todos os cursos da cidade. Para este indicador, foram obtidos os dados do Enade dos anos de 2017, 2018 e 2019 (os mais recentes até o momento da construção do indicador). Estes dados são fornecidos no site do [INEP](https://www.gov.br/inep/pt-br/acesso-a-informacao/dados-abertos/indicadores-educacionais/indicadores-de-qualidade-da-educacao-superior).Para calcular o índice, foi seguido o seguinte passo a passo: 1. Obter, para os cem municípios mais populosos, o número de concluintes em todos os cursos avaliados pelo ENADE; 1. Obter, para os cem municípios mais populosos, o número de concluintes em cursos com notas 4 e 5 no ENADE; 1. Calcular indicador por meio da seguinte fórmula:

O tratamento das bases de dados e o cálculo do índice foram feitos usando o R. O script encontra-se no [Apêndice](#i622_script) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

### Indicador Custo Médio de Salários de Dirigentes (i623)

Este indicador refere-se ao salário médio de funcionários em cargos de gerências e direção segundo classificação própria a partir da Classificação Brasileira de Ocupações (CBO). O cálculo do indicador consiste em dividir o salário dos funcionários atuantes nesses cargos pelo total de trabalhadores nos mesmos cargos. Os cargos utilizados nesse indicador segundo classificação da CBO são: 1) diretores e gerentes em empresa de serviços de saúde, da educação, ou de serviços cul, 2) dirigentes de empresas e organizações (exceto de interesse público), e 3) gerentes. Esse dado é coletado pela RAIS na seção de vínculos. Assim como os outros, deve-se selecionar os municípios do ICE e as ocupações da CBO 2002 na seção de “Princ Subgrupos” e selecionar “Vl Remuneração Média Nom”. Os cargos utilizados neste indicador e seus respectivos subgrupos principais segundo a CBO são: dirigentes de empresas e organizações (exceto de interesse público) (12), diretores e gerentes em empresa de serviços de saúde, da educação, ou de serviços culturais, sociais ou pessoais (13); e gerentes (14). Os passos tomados para chegar ao indicador estão descritos a seguir:

1. Obter total de trabalhadores em cargos de gerência e direção;

* Fonte: RAIS (Relação Anual de Informações Sociais). Ministério da Economia
* Para selecionar os devidos cargos no banco de dados da RAIS, foi consultado o arquivo “Estrutura CBO” disponível na [internet](http://www.mtecbo.gov.br/cbosite/pages/downloads.jsf);
* A RAIS foi acessado por meio do data lake público Base dos Dados;

1. Obter a soma dos salarios de todos dirigentes (cargos de interesse) em cada município.

* A fonte desses dados também é a RAIS (vínculo);

1. Calcular o indicador i623 de acordo com a seguinte fórmula:

O tratamento dos dados e cálculo do índice foram feitos no R. O script encontra-se no [Apêndice](#i623_script) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

# Determinante Cultura

## Subdeterminante Iniciativa

### Indicador Pesquisas com o Termo “Empreendedor” (i711)

Este indicador reflete a quantidade de buscas na internet pela palavra-chave “empreendedor” no município, reportado pelo Google Trends. Os seguintes passos foram tomados para se obter o indicador:

1. Obter informações a partir do site Google Trends;

* No site, pesquisar pela palavra-chave, selecionar “Brazil”, “2020”, “All Categories”, “Web Search”;
* Selecionar o nível territorial “City” e fazer o download da planilha com o resultado da pesquisa.

O tratamento dos dados e cálculo do indicador foram feitos no R. Como o procedimento é o mesmo para todos os indicadores do determinante 7, todos eles são calculados no mesmo script, que encontra-se no [Apêndice](#d7_script) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

### Indicador Pesquisas com o Termo “MEI” (i712)

Ver [Indicador Pesquisas com Termo “Empreendedor”](#i711), que apresenta o procedimento para se chegar aos indicadores do determinante Cultura. O script para os indicadores do determinante Cultura estão no [Apêndice](#d7_script) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

## Subdeterminante Instituições

### Indicador Pesquisas com o Termo “Sebrae” (i721)

Ver [Indicador Pesquisas com Termo “Empreendedor”](#i711), que apresenta o procedimento para se chegar aos indicadores do determinante Cultura. O script para os indicadores do determinante Cultura estão no [Apêndice](#d7_script) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

### Indicador Pesquisas com o Termo “Franquia” (i722)

Ver [Indicador Pesquisas com Termo “Empreendedor”](#i711), que apresenta o procedimento para se chegar aos indicadores do determinante Cultura. O script para os indicadores do determinante Cultura estão no [Apêndice](#d7_script) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

### Indicador Pesquisas com o Termo “SIMPLES Nacional” (i723)

Ver [Indicador Pesquisas com Termo “Empreendedor”](#i711), que apresenta o procedimento para se chegar aos indicadores do determinante Cultura. O script para os indicadores do determinante Cultura estão no [Apêndice](#d7_script) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

### Indicador Pesquisas com o Termo “Senac” (i724)

Ver [Indicador Pesquisas com Termo “Empreendedor”](#i711), que apresenta o procedimento para se chegar aos indicadores do determinante Cultura. O script para os indicadores do determinante Cultura estão no [Apêndice](#d7_script) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

# Análise de Componentes Principais

## Padronização

Os 49 indicadores são medidos em diferentes escalas: reais, anos, metros. A fim de impedir que a unidade de medida influencie o resultado, fizemos transformações estatísticas que alteram os valores dos indicadores mas não a posição relativa no ranking.

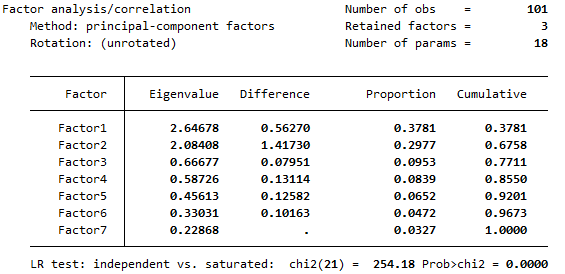
De cada indicador subtraímos sua média e dividimos pelo desvio padrão populacional. Assim, todos os indicadores terão valor entre zero e um. Em seguida, somamos os indicadores de cada sub-determinante. Padronizamos novamente cada sub-determinante, dessa vez somando 6 ao resultado. Somamos todos os sub-determinante de cada determinante. Padronizamos os determinantes uma última vez, repetindo a fórmula anterior. Se é o determinante, sua versão padronizada será

Clique [aqui](#pad_script) para ver o código em Stata que faz essas padronizações.

## Análise de Componentes Principais

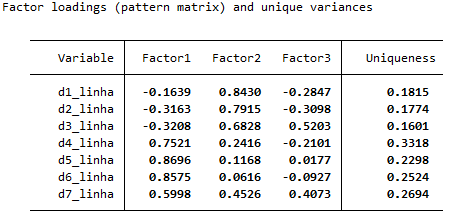
Clique [aqui](#pca_script) para ver o código Stata desta seção.

A análise de componentes principais (pca) é feita no Stata através do comando factor [variáveis], pcf, em que as variáveis são os determinantes padronizados (d1\_linha, …, d7\_linha). Geramos sete fatores, cujos autovalores (eigenvalues no Stata) mostram o quanto de variância é explicada por cada fator. Selecionamos tantos fatores quanto for necessário para explicar ao menos 70% da variância. Neste caso, usamos os três primeiros (Figura 1).



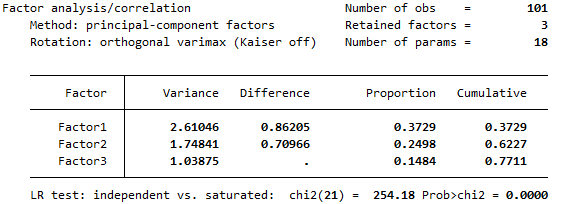
Fatores

As cargas fatoriais geradas pelo comando factor (Figura 2) apresentam as correlações entre as variáveis originais e os fatores. Assim, quanto maior o valor absoluto carga fatorial, maior será a correlação com o fator. Valores negativos representam impacto inverso no fator.

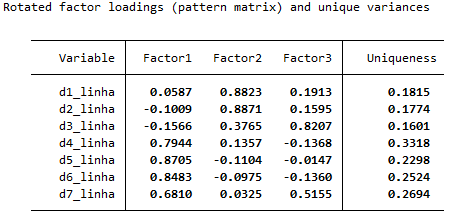


Cargas Fatoriais

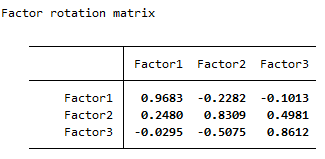
Em seguida, é feito o processo de rotação fatorial com o comando *rotate*. Trata-se de um ajuste aos eixos fatoriais para facilitar a sua interpretação. Os dois primeiros quadros (Figuras 4 e 5) gerados por este comando no Stata têm interpretação similar à dada acima. Já o terceiro quadro (Figura 6) mostra uma matriz de correlação entre os fatores relevantes.



fatores após rotação

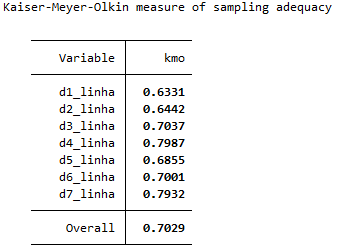


cargas após rotação



Matriz de rotação

Testamos se a nossa amostra é adequada com um teste de Kaiser-Meyer-Olklin (KMO). Valores mais altos, entre zero e um, representam uma melhor adequação da amostra.



Teste KMO

Em sequência, calculamos os scores para os fatores para cada cidade através do comando predict ice1 ice2, ice3, score, em que ice{i} corresponde ao i-ésimo fator. Os scores são calculados usando as cargas fatoriais como base para o cálculo de coeficientes de uma regressão padrão. Vejamos o coeficiente:

O numerador é representado pela covariância entre a variável independente e a dependente, que pode ser obtido pela carga fatorial. O denominador é dado pela variância do indicador, que é calculada pelo Stata. É possível calcular o valor dos scores multiplicando o valor dos determinantes padronizados (d1\_linha, …, d7\_linha) pelo coeficiente de regressão que o Stata calcula. Isto é feito pelo comando predict ice1 ice2,ice3, score.

Como resultado, são criadas três colunas (ice1, ice2 e ice3) contendo, para cada cidade, os scores para os fatores 1, 2 e 3. O Stata calcula estes scores de modo a cada um destes ter média zero e desvio padrão um.

Por fim, é criado o índice (variável ice), que soma os scores para os três fatores gerados pela análise fatorial. O último passo é padronizar a soma (ver [padronização](#pad)) Assim, o Índice de Cidades Empreendedoras 2021 é gerado pela variável ice\_final. O comando gsort -ice\_final deixa as observações em ordem decrescente.

# Correlação com ICE2020

## Teste de Correlação

Queremos saber em que medida o índice atual é diferente do passado. Para tanto, usamos um teste de correlação de Spearman, que mede a dependência estatística entre a classificação de duas variáveis . Excluindo o município de Marabá, que não aparece no ranking passado, encontramos correlação de 34%, e boa evidência estatística de que o cálculo atual e passado são relacionados (p valor = 0.00088).

# Ranking

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| id\_municipio | sigla\_uf | nome | ice1 | ice2 | ice3 | ice | ice\_final |
| 5108402 | MT | Varzea Grande | -1.134 | 5.279 | 0.938 | 5.084 | 8.950 |
| 5103403 | MT | Cuiaba | 0.534 | 3.256 | 0.963 | 4.753 | 8.758 |
| 2611606 | PE | Recife | 1.027 | 0.943 | 1.485 | 3.455 | 8.004 |
| 4106902 | PR | Curitiba | 2.610 | 1.309 | -1.007 | 2.911 | 7.689 |
| 2800308 | SE | Aracaju | -0.115 | 1.190 | 1.793 | 2.869 | 7.665 |
| 3106200 | MG | Belo Horizonte | 2.134 | 1.523 | -0.805 | 2.852 | 7.655 |
| 3550308 | SP | Sao Paulo | 3.534 | 0.101 | -0.908 | 2.727 | 7.582 |
| 3303302 | RJ | Niteroi | 0.739 | -0.526 | 2.316 | 2.530 | 7.468 |
| 3534401 | SP | Osasco | 2.190 | 0.055 | 0.068 | 2.313 | 7.342 |
| 3549904 | SP | Sao Jose Dos Campos | 1.542 | -0.223 | 0.979 | 2.297 | 7.333 |
| 3303500 | RJ | Nova Iguacu | -1.010 | 1.297 | 1.880 | 2.166 | 7.257 |
| 4314902 | RS | Porto Alegre | 2.384 | 0.668 | -0.902 | 2.151 | 7.248 |
| 3547809 | SP | Santo Andre | 1.071 | -0.217 | 1.245 | 2.098 | 7.217 |
| 3548708 | SP | Sao Bernardo Do Campo | 1.432 | 0.025 | 0.565 | 2.022 | 7.173 |
| 5300108 | DF | Brasilia | 0.952 | -0.472 | 1.405 | 1.885 | 7.093 |
| 4205407 | SC | Florianopolis | 2.011 | -0.170 | 0.040 | 1.881 | 7.091 |
| 2111300 | MA | Sao Luis | -0.196 | -0.781 | 2.820 | 1.843 | 7.069 |
| 5208707 | GO | Goiania | 0.731 | 0.829 | 0.227 | 1.788 | 7.037 |
| 3552205 | SP | Sorocaba | 1.026 | -1.172 | 1.892 | 1.746 | 7.013 |
| 3304557 | RJ | Rio De Janeiro | 1.688 | -0.197 | 0.000 | 1.492 | 6.866 |
| 2609600 | PE | Olinda | -1.116 | 1.989 | 0.617 | 1.489 | 6.864 |
| 4304606 | RS | Canoas | -0.229 | -0.098 | 1.699 | 1.372 | 6.796 |
| 2927408 | BA | Salvador | 0.315 | 0.729 | 0.290 | 1.333 | 6.774 |
| 5002704 | MS | Campo Grande | 0.070 | -0.882 | 2.101 | 1.289 | 6.748 |
| 3543402 | SP | Ribeirao Preto | 0.887 | -0.698 | 1.095 | 1.284 | 6.745 |
| 2704302 | AL | Maceio | -0.099 | 0.233 | 1.135 | 1.269 | 6.736 |
| 2507507 | PB | Joao Pessoa | 0.260 | 0.663 | 0.291 | 1.214 | 6.704 |
| 1501402 | PA | Belem | 0.447 | 0.896 | -0.258 | 1.085 | 6.630 |
| 2211001 | PI | Teresina | -0.159 | 0.438 | 0.769 | 1.048 | 6.608 |
| 3513801 | SP | Diadema | 0.740 | -1.010 | 1.249 | 0.978 | 6.567 |
| 3525904 | SP | Jundiai | 1.000 | -0.607 | 0.416 | 0.809 | 6.469 |
| 3506003 | SP | Bauru | 0.108 | 0.586 | 0.109 | 0.802 | 6.465 |
| 3509502 | SP | Campinas | 1.633 | -0.815 | -0.017 | 0.800 | 6.464 |
| 2304400 | CE | Fortaleza | -0.020 | 0.500 | 0.320 | 0.799 | 6.464 |
| 3205200 | ES | Vila Velha | 0.012 | -0.964 | 1.582 | 0.630 | 6.365 |
| 2607901 | PE | Jaboatao Dos Guararapes | -1.330 | 1.019 | 0.855 | 0.543 | 6.315 |
| 1100205 | RO | Porto Velho | -0.364 | 1.080 | -0.185 | 0.531 | 6.308 |
| 4209102 | SC | Joinville | 1.003 | -0.823 | 0.317 | 0.496 | 6.288 |
| 3205309 | ES | Vitoria | 1.325 | -0.156 | -0.693 | 0.477 | 6.277 |
| 2408102 | RN | Natal | 0.127 | 0.237 | 0.100 | 0.464 | 6.269 |
| 3118601 | MG | Contagem | -0.008 | -0.220 | 0.435 | 0.208 | 6.121 |
| 3301702 | RJ | Duque De Caxias | -0.554 | -0.253 | 0.995 | 0.188 | 6.109 |
| 2504009 | PB | Campina Grande | 0.127 | 0.854 | -0.872 | 0.108 | 6.063 |
| 3170206 | MG | Uberlandia | 0.368 | -0.452 | 0.164 | 0.080 | 6.047 |
| 1400100 | RR | Boa Vista | -0.869 | 0.307 | 0.638 | 0.077 | 6.044 |
| 4113700 | PR | Londrina | 0.680 | -0.111 | -0.575 | -0.005 | 5.997 |
| 1302603 | AM | Manaus | 0.030 | -1.005 | 0.967 | -0.008 | 5.995 |
| 4305108 | RS | Caxias Do Sul | 0.728 | -0.493 | -0.262 | -0.027 | 5.984 |
| 4314407 | RS | Pelotas | -0.090 | 0.230 | -0.224 | -0.083 | 5.952 |
| 2303709 | CE | Caucaia | -1.576 | 1.020 | 0.472 | -0.085 | 5.951 |
| 3205002 | ES | Serra | -0.434 | -0.087 | 0.396 | -0.124 | 5.928 |
| 4125506 | PR | Sao Jose Dos Pinhais | 0.001 | -0.431 | 0.258 | -0.171 | 5.901 |
| 2408003 | RN | Mossoro | -0.737 | 0.472 | 0.030 | -0.235 | 5.864 |
| 3526902 | SP | Limeira | 0.372 | 0.607 | -1.321 | -0.342 | 5.802 |
| 4316907 | RS | Santa Maria | 0.466 | 0.102 | -0.913 | -0.345 | 5.800 |
| 5201405 | GO | Aparecida De Goiania | -0.667 | 0.228 | 0.079 | -0.360 | 5.791 |
| 3303906 | RJ | Petropolis | -0.712 | 0.786 | -0.481 | -0.406 | 5.764 |
| 3136702 | MG | Juiz De Fora | 0.020 | 1.183 | -1.629 | -0.425 | 5.753 |
| 3548500 | SP | Santos | 0.717 | -0.693 | -0.497 | -0.473 | 5.725 |
| 1200401 | AC | Rio Branco | -0.755 | 0.843 | -0.604 | -0.516 | 5.701 |
| 3305109 | RJ | Sao Joao De Meriti | -1.307 | 0.973 | -0.265 | -0.599 | 5.652 |
| 2910800 | BA | Feira De Santana | -0.794 | -0.181 | 0.367 | -0.608 | 5.647 |
| 3300456 | RJ | Belford Roxo | -1.750 | -0.439 | 1.532 | -0.657 | 5.619 |
| 1600303 | AP | Macapa | -1.085 | 0.281 | 0.019 | -0.785 | 5.545 |
| 3170107 | MG | Uberaba | -0.236 | 0.288 | -0.850 | -0.798 | 5.537 |
| 4309209 | RS | Gravatai | -0.463 | 0.109 | -0.490 | -0.844 | 5.510 |
| 1721000 | TO | Palmas | 0.024 | -0.195 | -0.707 | -0.878 | 5.490 |
| 4119905 | PR | Ponta Grossa | -0.267 | -0.147 | -0.504 | -0.918 | 5.467 |
| 3549805 | SP | Sao Jose Do Rio Preto | 0.131 | -0.334 | -0.716 | -0.920 | 5.466 |
| 2611101 | PE | Petrolina | -0.878 | 1.944 | -2.019 | -0.953 | 5.447 |
| 3106705 | MG | Betim | -0.345 | -0.230 | -0.381 | -0.956 | 5.445 |
| 3529401 | SP | Maua | -0.473 | -0.486 | -0.135 | -1.094 | 5.365 |
| 2905701 | BA | Camacari | -0.481 | -0.818 | 0.192 | -1.106 | 5.358 |
| 3301009 | RJ | Campos Dos Goytacazes | -0.755 | 0.165 | -0.634 | -1.224 | 5.290 |
| 3304904 | RJ | Sao Goncalo | -0.907 | -1.227 | 0.910 | -1.225 | 5.289 |
| 4115200 | PR | Maringa | 0.538 | -0.027 | -1.765 | -1.254 | 5.272 |
| 4202404 | SC | Blumenau | 0.207 | -0.789 | -0.706 | -1.287 | 5.253 |
| 3143302 | MG | Montes Claros | -0.639 | 0.382 | -1.043 | -1.300 | 5.246 |
| 5201108 | GO | Anapolis | -0.421 | -0.150 | -0.775 | -1.346 | 5.219 |
| 3518800 | SP | Guarulhos | -0.306 | -0.526 | -0.552 | -1.384 | 5.197 |
| 3510609 | SP | Carapicuiba | -0.684 | -0.625 | -0.077 | -1.386 | 5.196 |
| 4104808 | PR | Cascavel | 0.077 | -0.814 | -0.689 | -1.426 | 5.172 |
| 3530607 | SP | Mogi Das Cruzes | -0.125 | -0.438 | -0.921 | -1.484 | 5.139 |
| 1504208 | PA | Maraba | -1.236 | -0.439 | 0.172 | -1.504 | 5.127 |
| 1500800 | PA | Ananindeua | -1.120 | -0.199 | -0.287 | -1.606 | 5.068 |
| 3552403 | SP | Sumare | -0.637 | -1.258 | 0.169 | -1.726 | 4.999 |
| 3538709 | SP | Piracicaba | 0.653 | -1.077 | -1.319 | -1.743 | 4.988 |
| 3516200 | SP | Franca | -0.201 | 0.270 | -1.874 | -1.804 | 4.953 |
| 3154606 | MG | Ribeirao Das Neves | -1.631 | -0.555 | 0.289 | -1.898 | 4.899 |
| 2604106 | PE | Caruaru | -1.111 | -0.687 | -0.395 | -2.193 | 4.728 |
| 3541000 | SP | Praia Grande | -0.870 | 0.464 | -1.801 | -2.207 | 4.719 |
| 3201308 | ES | Cariacica | -0.945 | -0.389 | -0.904 | -2.237 | 4.702 |
| 3523107 | SP | Itaquaquecetuba | -0.955 | -1.880 | 0.418 | -2.417 | 4.597 |
| 3552809 | SP | Taboao Da Serra | -0.088 | -1.343 | -1.102 | -2.532 | 4.531 |
| 3552502 | SP | Suzano | -0.345 | -0.766 | -1.423 | -2.534 | 4.530 |
| 1506807 | PA | Santarem | -1.093 | -0.663 | -0.796 | -2.552 | 4.519 |
| 3554102 | SP | Taubate | -0.074 | -1.174 | -1.344 | -2.593 | 4.496 |
| 2610707 | PE | Paulista | -1.274 | -1.344 | -0.095 | -2.713 | 4.426 |
| 3551009 | SP | Sao Vicente | -1.013 | -1.511 | -0.457 | -2.981 | 4.270 |
| 2933307 | BA | Vitoria Da Conquista | -0.919 | -0.851 | -1.593 | -3.363 | 4.049 |
| 3518701 | SP | Guaruja | -1.066 | -1.233 | -1.293 | -3.592 | 3.916 |

# Apêndice

### maiores municípios

voltar para [introdução](#intro)

library(tidyverse)  
library(readxl)  
library(basedosdados)  
  
# populacao estimada 2020  
  
# mp20 <- read\_xls("POP2020\_20210331.xls", sheet = "Municípios", skip = 1) %>%  
# select(sigla\_uf=1, nome = 4, pop =5 ) %>%  
# mutate(pop = pop %>% str\_remove\_all(padrao) %>% as.numeric) %>%  
# filter(rank(-pop) <= 100) %>%  
# arrange(-pop)  
  
# estao no ranking atual mas nao no antigo  
# setdiff(mp21$nome, mp20$nome)  
  
# estao no ranking antigo mas nao no atual  
# setdiff(mp20$nome,mp21$nome)  
  
# expressao regular que procura ponto ou digitos entre parenteses  
  
padrao <- "(\\.|\\(\\d+\\))"   
  
mp21 <- read\_xls("estimativa\_dou\_2021.xls", sheet = "Municípios", skip = 1)%>%  
 select(sigla\_uf=1, nome = 4, pop =5 ) %>%  
 mutate(pop = pop %>% str\_remove\_all(padrao) %>% as.numeric)  
  
mp21\_top100 <- mp21 %>%  
 filter(rank(-pop) <= 100 | (nome == "Santa Maria" & sigla\_uf == "RS")) %>%  
 arrange(-pop)  
  
  
  
# codigo dos municipios ---------------------------------------------------  
set\_billing\_id("ice2021")  
diretorios <- read\_sql(  
 "SELECT \* FROM `basedosdados.br\_bd\_diretorios\_brasil.municipio`")  
  
#juntar codigos e pop  
  
municode <- diretorios %>%   
 select(id\_municipio,id\_municipio\_6,sigla\_uf,nome) %>%  
 right\_join(mp21\_top100)  
  
# salvar no disco   
write\_csv(municode,"municode.csv")

### maiores municípios

voltar para [introdução](#intro)

library(tidyverse)  
library("basedosdados")  
library(readxl)  
  
# Ranking de 100 municipios mais populosos  
  
# ------------------------------------------------------------------  
# PASSO 0: definir projeto no google cloud  
# ------------------------------------------------------------------  
  
# definir projeto no Google Cloud  
set\_billing\_id("workshop-teste-322616")  
  
# baixar tabela com codigos e nomes de municipios  
query <- "SELECT \* FROM `basedosdados.br\_bd\_diretorios\_brasil.municipio`"  
cod\_mun <- read\_sql(query)  
  
# ------------------------------------------------------------------  
# PASSO 1: obter ranking de populacao de 2020  
# ------------------------------------------------------------------  
  
# baixar dados de populacao de 2020   
query <- "SELECT \* FROM `basedosdados.br\_ibge\_populacao.municipio`   
WHERE ano = 2020"  
pop\_mun20 <- read\_sql(query)  
  
# rankear municipios por populacao (maior para menor)  
top100\_mun20 <- pop\_mun20 %>%  
 top\_n(100, populacao) %>%  
 inner\_join(cod\_mun, by = 'id\_municipio') %>%   
 select(nome, sigla\_uf, populacao)  
  
# ------------------------------------------------------------------  
# PASSO 2: obter ranking de populacao de 2021  
# ------------------------------------------------------------------  
  
# importar tabela com estimatica populacional de 2021  
# algumas linhas possuiam observacoes no campo de populacao  
# essas observacoes foram removidas manualmente  
top100\_mun21 <- read\_excel('C:/Users/User/OneDrive/Documentos/ICE 2021/  
 estimativa\_dou\_2021.xls', sheet = 2, skip = 1) %>%  
 na.omit() %>%  
 select(nome = 'NOME DO MUNICÍPIO',  
 sigla\_uf = 'UF',  
 populacao = 'POPULAÇÃO ESTIMADA') %>%  
 top\_n(100, populacao)  
  
# juntar tabelas  
top100\_mun\_cod <- top100\_mun20 %>%  
 full\_join(top100\_mun21, by = c('nome', 'sigla\_uf'),  
 suffix = c('2020', '2021')) %>%  
 arrange(desc(populacao2021)) %>%  
 inner\_join(cod\_mun, by = c('nome', 'sigla\_uf')) %>%   
 select(id\_municipio, nome, sigla\_uf)  
  
# salvar dataframe com os 100 municipios mais populosos  
write.csv(top100\_mun\_cod, "top100\_mun\_cod.csv", row.names = FALSE)

voltar para [Tempo de Processos](#sd11)

#  
#  
# SD 11 - AMBIENTE REGULATÓRIO - TEMPO DE PROCESSOS  
  
library(tidyverse)  
library(readxl)  
library(abjutils)  
library(janitor)  
  
# maiores municipios  
municode <- read\_csv("municode.csv") %>%  
 select(id\_municipio, sigla\_uf, nome) %>%  
 mutate(nome = abjutils::rm\_accent(nome) %>% str\_to\_title())  
  
# pega colunas que interessam para os indicadores de endereco e nome  
col\_interesse <- read\_excel(  
 "ambiente\_regulatorio/dados/sd11\_localiza\_nome/sd11abr20.xlsx",  
 sheet = 1,  
 skip = 1  
) %>%  
 names() %>%  
 str\_subset(pattern = "QTDE")  
  
col\_interesse <- col\_interesse[c(1, 2, 5, 7)]  
  
# Le os dados de abertura  
aberturas <-  
 list.files("ambiente\_regulatorio/dados/sd11\_localiza\_nome/") %>%  
 paste0("ambiente\_regulatorio/dados/sd11\_localiza\_nome/", .) %>%  
 map\_dfr( ~ read\_excel(., sheet = 1, skip = 1))  
# limpa os dados  
aberturas\_clean <- aberturas %>%  
 select(sigla\_uf = UF,  
 nome = MUNICÍPIO,  
 all\_of(col\_interesse)) %>%  
 janitor::clean\_names() %>%  
 mutate(nome = str\_to\_title(nome))  
  
# filtra os maiores municipios  
aberturas\_top <- aberturas\_clean %>%  
 right\_join(municode)  
  
# Tempo Viabilidade Localizacao -------------------------------------------  
tempo\_localiza <- aberturas\_top %>%  
 filter(!is.na(qtde\_hh\_viabilidade\_end)) %>%  
 group\_by(id\_municipio, sigla\_uf, nome) %>%  
 summarise(  
 tempo\_medio = mean(qtde\_hh\_viabilidade\_end, na.rm = TRUE),  
 sd11\_localiza = 1 / tempo\_medio  
 )  
  
# encontra cidades faltantes  
id\_faltantes <-  
 setdiff(municode$id\_municipio, tempo\_localiza$id\_municipio)  
  
faltantes <- municode %>%  
 filter(id\_municipio %in% id\_faltantes) %>%  
 select(id\_municipio, sigla\_uf, nome)  
  
# completa com 0 as faltantes  
tempo\_localiza <- tempo\_localiza %>%  
 ungroup() %>%  
 add\_row(  
 id\_municipio = as.double(faltantes$id\_municipio),  
 sigla\_uf = faltantes$sigla\_uf,  
 nome = faltantes$nome,  
 tempo\_medio = 0,  
 sd11\_localiza = 0  
 ) %>%  
 arrange(-sd11\_localiza)  
  
# Tempo Viabilidade Nome --------------------------------------------------  
cols\_sd11\_nome <- c("nome", "dbe", "deferimento")  
  
tempo\_nome <- aberturas\_clean %>%  
 select(sigla\_uf, nome, contains(cols\_sd11\_nome)) %>%  
 group\_by(sigla\_uf) %>%  
 summarise(across(contains("qtde"), ~ sum(.x, na.rm = TRUE)),  
 ordens = n()) %>%  
 rowwise(sigla\_uf) %>%  
 mutate(  
 tempo\_sum = sum(c\_across(contains("qtde"))),  
 tempo\_nome\_medio = tempo\_sum / ordens,  
 sd11\_nome = 1 / tempo\_nome\_medio  
 )  
  
# monta numa planilha os indicadores endereco e nome  
sd11\_localiza\_nome <- tempo\_nome %>%  
 select(sigla\_uf, sd11\_nome) %>%  
 right\_join(tempo\_localiza) %>%  
 select(id\_municipio, sigla\_uf, nome, i111 = sd11\_localiza, i112 = sd11\_nome)  
  
# salva o arquivo  
write\_excel\_csv(sd11\_localiza\_nome, "dados\_finais/sd11\_localiza\_nome.xlsx")  
  
# congestionamento em tribunais -------------------------------------------  
# processos  
novos <-   
 read\_excel("ambiente\_regulatorio/dados/sd11\_processos\_novos.xlsx")  
  
baixados <-   
 read\_excel("ambiente\_regulatorio/dados/sd11\_processos\_baixados.xlsx")  
  
pendentes <-   
 read\_excel("ambiente\_regulatorio/dados/sd11\_processos\_pendentes.xlsx")  
  
  
sd11\_congestionamento\_tribunais <-  
 list.files("ambiente\_regulatorio/dados/", pattern = "sd11\_processos\_") %>%  
 paste0("ambiente\_regulatorio/dados/", .) %>%  
 map( ~ read\_excel(., col\_names = c("nome", .), skip = 1)) %>%  
 reduce(full\_join) %>%  
 rename(  
 nome = nome,  
 baixados = 2,  
 novos = 3,  
 pendentes = 4  
 ) %>%  
 mutate(nome = nome %>% str\_to\_title() %>% abjutils::rm\_accent()) %>%  
 right\_join(municode) %>%  
 mutate(congest = 1 - (baixados / (novos + pendentes)),  
 sd11\_congestionamento = 1 / congest) %>%  
 select(id\_municipio, sigla\_uf, nome, everything()) %>%  
 arrange(-sd11\_congestionamento)  
  
  
write\_excel\_csv(  
 sd11\_congestionamento\_tribunais,  
 "ambiente\_regulatorio/sd11\_congestionamento\_tribunais\_completo.xlsx"  
)  
  
  
sd11\_congestionamento\_tribunais %>%  
 select(1,2,3,8) %>%  
 write\_excel\_csv("dados\_finais/sd11\_congestionamento\_tribunais.xlsx")

### Script sd12

voltar para [Tributação](#sd12)

#  
#  
# sd 12 - AMBIENTE REGULATORIO - TRIBUTACAO  
  
library(tidyverse)  
library(basedosdados)  
  
# maiores municipios  
municode <-  
 read\_csv("municode.csv") %>% select(id\_municipio, sigla\_uf, nome)  
  
# ICMS --------------------------------------------------------------------  
  
icms <- read.delim(  
 "ambiente\_regulatorio/dados/finbra.csv",  
 dec = ",",  
 sep = ";",  
 skip = 3  
) %>%  
 filter(Coluna == "Receitas Brutas Realizadas",  
 str\_detect(Conta, "1.1.1.8.02.0.0")) %>%  
 select(sigla\_uf = UF, icms = Valor)  
  
# pib municipal  
basedosdados::set\_billing\_id("ice2021")  
  
pib\_mun <- read\_sql(  
 "SELECT \* FROM `basedosdados.br\_ibge\_pib.municipio` WHERE ano = 2018") %>%  
 select(1:3)  
  
# tabela auxliar para relacionar municipio e estado  
municode2 <- basedosdados::read\_sql(  
 "SELECT id\_municipio,sigla\_uf  
 FROM `basedosdados.br\_bd\_diretorios\_brasil.municipio`"  
)  
  
# pib estadual  
pib\_estadual <- pib\_mun %>%  
 left\_join(municode2) %>%  
 group\_by(sigla\_uf) %>%  
 summarise(pib\_estadual = sum(pib, na.rm = TRUE))  
  
# unindo pib e icms  
df <- full\_join(icms, pib\_estadual) %>%  
 mutate(icms\_pib = icms / pib\_estadual,  
 sd12\_icms = 1 / icms\_pib)  
  
# tabela final  
icms\_final <- left\_join(municode, select(df, sigla\_uf, i121 = sd12\_icms))  
  
write\_excel\_csv(icms\_final, "dados\_finais/sd12\_icms.xlsx")  
  
# IPTU e ISS --------------------------------------------------------------------  
# receitas dos municipios  
# dados brutos  
finbramun <- read.delim(  
 "ambiente\_regulatorio/dados/finbramun.csv",  
 dec = ",",  
 sep = ";",  
 skip = 3  
)  
# dados limpos  
finbramun\_clean <- finbramun %>%  
 select(  
 id\_municipio = Cod.IBGE,  
 coluna = Coluna,  
 conta = Conta,  
 valor = Valor  
 ) %>%  
 filter(id\_municipio %in% municode$id\_municipio,  
 coluna == "Receitas Brutas Realizadas") %>%  
 select(-coluna) %>%  
 as\_tibble() %>%  
 separate(conta,  
 c("conta\_num", "descricao"),  
 sep = " ",  
 extra = "merge") %>%  
 filter(conta\_num %>% str\_detect("(\\d\\.){4}(\\d){2}(\\.\\d){2}"))  
  
# somente iss e iptu  
munitax <- finbramun\_clean %>%  
 filter(conta\_num %in% c("1.1.1.8.01.1.0", "1.1.1.8.02.3.0")) %>%  
 pivot\_wider(names\_from = descricao, values\_from = valor) %>%  
 select(id\_municipio, iptu = 3, iss = 4) %>%  
 left\_join(municode)  
  
munitax2 <- munitax %>%  
 group\_by(id\_municipio, nome) %>%  
 summarise(across(c("iptu", "iss"), ~ sum(.x, na.rm = TRUE))) %>%  
 ungroup() %>%  
 mutate(id\_municipio = as.character(id\_municipio)) %>%  
 left\_join(pib\_mun) %>%  
 mutate(  
 iptu\_pib = iptu / pib,  
 iss\_pib = iss / pib,  
 sd12\_iptu = 1 / iptu\_pib,  
 sd12\_iss = 1 / iss\_pib  
 ) %>%  
 select(id\_municipio,  
 nome,  
 ano\_tax = ano,  
 iptu\_pib,  
 iss\_pib,  
 pib,  
 sd12\_iptu,  
 sd12\_iss) %>%  
 arrange(-sd12\_iptu,-sd12\_iss)  
  
munitax2[10, 8] <- 0 # atribui valor zero para belford roxo  
munitax2[22, 8] <- 0 # atribui valor zero para uberaba  
munitax2[30, 8] <- 0 # atribui valor zero para s.j meriti  
  
# econtra muni faltantes  
  
# maiores munis que nao estao nos dados  
m <- setdiff(municode$id\_municipio, munitax2$id\_municipio)  
  
faltantes <- municode %>%  
 filter(id\_municipio %in% m) %>%  
 mutate(id\_municipio = as.character(id\_municipio))  
  
# e lhes da nota zero  
munitax2 <- munitax2 %>%  
 ungroup() %>%  
 add\_row(  
 id\_municipio = faltantes$id\_municipio,  
 nome = faltantes$nome,  
 ano\_tax = rep("2018", 2),  
 iptu\_pib = rep("0", 2),  
 iss\_pib = rep("0", 2),  
 pib = rep("0", 2),  
 sd12\_iptu = rep("0", 2),  
 sd12\_iss = rep("0", 2)  
 ) %>%  
 filter(!(is.na(nome) | is.na(sd12\_iss))) %>%  
 distinct(nome, .keep\_all = TRUE)  
  
sd12 <- munitax2 %>%  
 select(1, 2, i122 = 7, i123 = 8)  
  
write\_excel\_csv(sd12, "dados\_finais/sd12\_munitax.xlsx")  
  
# indice de gestao fiscal -------------------------------------------------  
  
iqgf <-  
 read\_excel(  
 "dados\_finais/sd12\_qualidade\_gestao\_fiscal.xlsx",  
 sheet = "Indicador",  
 col\_names = c("sigla\_uf", "nome", "i124"),  
 skip = 1  
 ) %>%  
 full\_join(municode, keep = FALSE) %>%  
 select(id\_municipio, everything()) %>%  
 arrange(i124)  
  
write\_excel\_csv(iqgf, "dados\_finais/sd12\_qualidade\_gestao\_fiscal.xlsx")

### Script sd13

voltar para [Complexidade Tributária](#sd13)

#  
# sd13 - ambiente regulatorio - complexidade tributaria  
  
library(tidyverse)  
library(readxl)  
library(pdftools)  
  
# maiores municipios  
municode <-  
 read\_csv("municode.csv") %>% select(id\_municipio, sigla\_uf, nome)  
  
# conta das receitas para calcular os indices  
contas\_interessantes <- pdf\_text("ambiente\_regulatorio/pdf\_contas.pdf") %>%  
 str\_split("\n") %>%  
 unlist() %>%  
 str\_extract\_all("(\\d\\.){4}(\\d){2}(\\.\\d){2}") %>%  
 unlist()  
  
# receitas dos municipios  
df <- read.delim(  
 "ambiente\_regulatorio/dados/finbramun.csv",  
 dec = ",",  
 sep = ";",  
 skip = 3  
)  
  
df\_sep <- df %>%  
 separate(Conta,  
 c("conta\_num", "descricao"),  
 sep = " ",  
 extra = "merge")  
  
finbramun <- df\_sep %>%  
 select(  
 id\_municipio = Cod.IBGE,  
 coluna = Coluna,  
 conta = Conta,  
 valor = Valor  
 ) %>%  
 filter(id\_municipio %in% municode$id\_municipio,  
 coluna == "Receitas Brutas Realizadas") %>%  
 select(-coluna) %>%  
 as\_tibble() %>%  
 separate(conta,  
 c("conta\_num", "descricao"),  
 sep = " ",  
 extra = "merge") %>%  
 filter(conta\_num %>% str\_detect("(\\d\\.){4}(\\d){2}(\\.\\d){2}"))  
  
# HH index  
ihh <- finbramun %>%  
 group\_by(id\_municipio) %>%  
 mutate(receita\_total = sum(valor, na.rm = TRUE)) %>%  
 ungroup() %>%  
 filter(conta\_num %in% contas\_interessantes[1:52]) %>%  
 group\_by(id\_municipio, conta\_num) %>%  
 summarise(conta\_total\_quad = (valor / receita\_total) ^ 2) %>%  
 group\_by(id\_municipio) %>%  
 summarise(ihh = sum(conta\_total\_quad))  
  
# iv index  
iv <- finbramun %>%  
 group\_by(id\_municipio) %>%  
 mutate(receita\_total = sum(valor)) %>%  
 ungroup() %>%  
 filter(conta\_num %in% contas\_interessantes[53:56]) %>%  
 group\_by(id\_municipio, receita\_total) %>%  
 summarise(conta\_total = sum(valor)) %>%  
 mutate(iv = conta\_total / receita\_total) %>%  
 select(id\_municipio, iv)  
  
# tem um 'municipio' faltando: brasilia  
municode %>%  
 filter(id\_municipio == setdiff(municode$id\_municipio, finbramun$id\_municipio))  
  
#tudo junto  
df <- left\_join(ihh, iv) %>%  
 mutate(sd13\_complexidade = ihh \* iv) %>%  
 left\_join(municode) %>%  
 select(id\_municipio, sigla\_uf, nome, everything()) %>%  
 arrange(-sd13\_complexidade) %>%  
 add\_row(  
 id\_municipio = 5300108,  
 nome = "Brasília",  
 ihh = 0,  
 iv = 0,  
 sd13\_complexidade = 0  
 )  
  
# salvar  
df %>% select(1:3, i131 = 6) %>%  
 write\_excel\_csv(df, "dados\_finais/sd13\_simplicidade\_tributaria.xlsx")  
  
# CND ---------------------------------------------------------------------  
  
sd13\_cnd <-  
 read\_excel("ambiente\_regulatorio/dados/Base\_MUNIC\_2019\_20210817.xlsx",  
 sheet = 4) %>%  
 select(id\_municipio = 1, tem\_cnd = MTIC1211) %>%  
 mutate(tem\_cnd = if\_else(tem\_cnd == "Sim", 1, 0)) %>%  
 right\_join(municode, keep = FALSE) %>%  
 select(id\_municipio, sigla\_uf, nome, i132 = tem\_cnd)  
  
write\_excel\_csv(sd13\_cnd, "dados\_finais/sd13\_cnd.xlsx")  
  
# atualizacao de zoneamento -----------------------------------------------  
  
municode <- read\_csv("municode.csv") %>%  
 select(id\_municipio, sigla\_uf, nome)  
  
sd13\_zone <- read\_excel(  
 "ambiente\_regulatorio/dados/Base\_MUNIC\_2018\_xlsx\_20201103.xlsx",  
 sheet = 3,  
 na = "-"  
) %>%  
 select(id\_municipio = 1, update\_zone = MLEG061) %>%  
 mutate(dif = if\_else(is.na(update\_zone), Inf, 2019 - as.double(update\_zone)),  
 indice = 1 / dif) %>%  
 right\_join(municode, keep = FALSE) %>%  
 select(id\_municipio, sigla\_uf, nome, i133 = indice)  
  
write\_excel\_csv(sd13\_zone, "dados\_finais/sd13\_zone.xlsx")

### Script sd21

voltar para [Transporte Interurbano](#sd21)

#  
#  
# SD21 - INFRAESTRUTURA - TRANSPORTE INTERURBANO  
  
library(tidyverse)  
library(sf)  
library(geobr)  
  
# municipios --------------------------------------------------------------  
mcoords <- read\_municipality() %>% # coordenadas dos municipios  
 select(id\_municipio = 1,  
 nome = 2,  
 geom = 5)  
  
municode <- read\_csv("municode.csv") %>%  
 select(id\_municipio, sigla\_uf, nome) %>%  
 mutate(nome = str\_to\_title(nome)) %>%  
 left\_join(mcoords, keep = FALSE)  
  
# aeroportos br -----------------------------------------------------------  
  
#unzip("infraestrutura/aeroportos/aerodromos-zip.zip",  
# exdir = "infraestrutura/aeroportos/voos\_shapefiles",  
# junkpaths = TRUE, overwrite = TRUE)  
  
  
aeroshp <-  
 read\_sf("infraestrutura/aeroportos/voos\_shapefiles/Aerodromos.shp")  
  
siglas\_geo <- aeroshp %>%  
 mutate(nm\_municip = str\_to\_title(nm\_municip)) %>%  
 select(sigla\_aero = cod\_icao) # geometry is sticky  
  
# voos 2020 -------------------------------------------------------------------  
# unzip os dados apenas uma vez  
  
#unzip("infraestrutura/aeroportos/Dados\_Estatisticos.zip",  
# exdir = "infraestrutura/aeroportos/dados\_voos",  
# junkpaths = TRUE, overwrite = TRUE)  
  
# voos no mundo  
voos <- read.delim("infraestrutura/aeroportos/dados\_voos/Dados Estat¡sticos.csv",  
 sep = ";")  
  
# no br  
voos\_br <- voos %>%  
 as\_tibble() %>%  
 select(ANO,  
 MÊS,  
 contains("AEROPORTO.DE.ORIGEM"),  
 GRUPO.DE.VOO,  
 DECOLAGENS) %>%  
 filter(ANO == 2020,  
 AEROPORTO.DE.ORIGEM..PAÍS. == "BRASIL",  
 GRUPO.DE.VOO == "REGULAR") %>%  
 rename(sigla\_aero = AEROPORTO.DE.ORIGEM..SIGLA.,  
 nome = AEROPORTO.DE.ORIGEM..NOME.,  
 sigla\_uf = AEROPORTO.DE.ORIGEM..UF.) %>%  
 mutate(nome = str\_to\_title(nome),  
 sigla\_uf = if\_else(nome == "Guaíra", "PR", sigla\_uf)) %>% # dados errados  
 group\_by(sigla\_aero, sigla\_uf, nome) %>%  
 summarise(decolagens = sum(DECOLAGENS, na.rm = TRUE)) %>%  
 ungroup()  
  
# no br e nos 100 maiores munipios  
  
decolagens <- voos\_br %>%  
 group\_by(sigla\_uf, nome) %>%  
 summarise(decolagens = sum(decolagens, na.rm = TRUE)) %>%  
 ungroup() %>%  
 right\_join(municode) %>%  
 select(id\_municipio, everything()) %>%  
 arrange(id\_municipio)  
  
# nao estao na lista  
naotem <- decolagens %>% filter(is.na(decolagens))  
  
# geolocaliza os aeroportos br  
voos\_br <- left\_join(voos\_br, siglas\_geo)  
  
# data viz ----------------------------------------------------------------  
# br\_airports <- ggplot()+  
# geom\_sf(data = geobr::read\_state())+ # mapa base com shapes dos estados br  
# geom\_sf(data = naotem$geometry, fill = "red")+  
# geom\_sf(data = voos\_br$geometry)+  
# theme(panel.background = element\_blank(),  
# axis.text = element\_blank(),  
# axis.ticks = element\_blank())  
  
# idsp <- naotem %>% filter(sigla\_uf == "SP") %>% pull(id\_municipio)  
# sp <- filter(naotem, id\_municipio %in% idsp)  
# sp\_voos <- filter(voos\_br, sigla\_uf == "SP") %>% pull(geometry)  
  
# sp\_airports <- ggplot()+  
# geom\_sf(data = geobr::read\_state(code\_state = "SP"))+  
# geom\_sf(data = sp, fill = "red")+  
# geom\_sf(data = sp\_voos)+  
# theme(panel.background = element\_blank(),  
# axis.text = element\_blank(),  
# axis.ticks = element\_blank())  
  
# ggsave("infraestrutura/aeroportos/br\_airports.png", br\_airports)  
# ggsave("infraestrutura/aeroportos/sp\_airports.png", sp\_airports)  
  
# achar um aeroporto para os muni faltantes ------------------------------------  
  
aeroprox <- function(id) {  
 mun <- filter(naotem, id\_municipio == id) %>%  
 st\_as\_sf()  
   
 r <- st\_distance(mun, st\_as\_sf(voos\_br), by\_element = TRUE) %>%  
 which.min()  
   
 voos\_br %>% filter(row\_number() == r) %>%  
 as.data.frame() %>%  
 select(sigla\_aero)  
}  
  
# decolagens com base em aeroporto mais perto  
siglas\_faltantes <- naotem %>%  
 as.data.frame() %>%  
 pull(id\_municipio) %>%  
 map(aeroprox) %>%  
 unlist()  
  
naotem <- naotem %>%  
 as.data.frame() %>%  
 select(1:3) %>%  
 bind\_cols(siglas\_faltantes) %>%  
 rename(sigla\_aero = 4) %>%  
 left\_join(voos\_br, by = "sigla\_aero", suffix = c("", "\_ref")) %>%  
 select(!geometry)  
  
# colunas \_ref NA significam aeroporto na cidade  
sd21\_voos <- decolagens %>%  
 as.data.frame() %>%  
 select(!geom) %>%  
 filter(!is.na(decolagens)) %>%  
 bind\_rows(naotem) %>%  
 arrange(-decolagens) %>%  
 rename(i212 = decolagens)  
  
write\_excel\_csv(sd21\_voos, "infraestrutura/aeroportos/sd21\_voos\_completo.xlsx")  
  
sd21\_voos %>% select(1,2,3,4) %>% write\_excel\_csv("dados\_finais/sd21\_voos.xlsx")  
  
#  
#  
# sd22 - INFRAESTRUTURA - CONDICOES URBANAS - TARIFA ENERGIA ELETRICA  
  
library(readxl)  
library(tidyverse)  
  
# dados brutos  
df <- read\_excel("infraestrutura/energia\_eletrica/RankingB1-24-08-2021.xlsx")  
  
# selecionamos tarifa convencional e limpamos dos dados  
tarifa <- df %>%  
 select(dist = Distribuidora,  
 sigla\_uf = UF,  
 tarifa = `Tarifa Convencional`) %>%  
 mutate(dist = dist %>% str\_to\_lower())  
  
  
mudar\_dist <- c(  
 "boa vista" = "roraima energia",  
 "equatorial pa" = "celpa",  
 "equatorial al" = "ceal",  
 "equatorial pi" = "cepisa",  
 "eletropaulo" = "enel sp",  
 "cebdis" = "ceb-dis",  
 "celg-d" = "enel go",  
 "celesc dis" = "celesc-dis",  
 "copel dis" = "copel-dis",  
 "rge" = "rge (agrupada)"  
)  
  
  
dist <-  
 read\_csv("infraestrutura/energia\_eletrica/distribuidoras.csv") %>%  
 pivot\_longer(matches("^d\\d$"), names\_to = "isdist", values\_to = "dist") %>%  
 filter(!is.na(dist)) %>%  
 select(!c(id\_municipio\_6, isdist, pop, sigla\_uf)) %>%  
 mutate(dist = dist %>% str\_replace\_all(mudar\_dist)) %>%  
 left\_join(tarifa, by = "dist") %>%  
 group\_by(id\_municipio) %>%  
 mutate(tarifa\_media = mean(tarifa), i223 = 1/tarifa\_media) %>%  
 distinct(id\_municipio, .keep\_all = TRUE) %>%  
 arrange(-i223)  
  
write\_csv(dist,  
 "infraestrutura/energia\_eletrica/sd22\_energia\_completa.csv")  
  
  
dist %>% select(1, 4, 2, i223) %>%  
 write\_csv("dados\_finais/sd22\_energia\_eletrica.csv")  
#  
#  
# SD21 - INFRAESTRUTURA - TRANSPORTE INTERURBANO - PORTOS  
  
library(tidyverse)  
library(sf)  
  
# portos ------------------------------------------------------------------  
# unzip(zipfile = "infraestrutura/portos/portos-zip.zip",  
# exdir = "infraestrutura/portos/portos\_shapefile", junkpaths = TRUE,  
# overwrite = TRUE)  
  
portos <-  
 read\_sf("infraestrutura/portos/portos\_shapefile/Portos.shp",  
 as\_tibble = TRUE)  
  
# seleciona portos publicos e os fluviais do amazonas  
portos\_am <-  
 c("Itacoatiara", "Manaus", "Tabatinga", "Parintins", "Eirunepé")  
  
am <- portos %>% filter(MUNICIPIO %in% portos\_am)  
  
pub <- portos %>%  
 filter(!str\_detect(NOMEPORTO, "TUP"),  
 SITUACAOPO == "Operando",!is.na(MUNICIPIO)) %>%  
 bind\_rows(am) %>%  
 select(  
 nome\_porto = 3,  
 sigla\_uf = 21,  
 nome = 20,  
 geometry = 31  
 ) %>%  
 distinct(nome\_porto, .keep\_all = TRUE)  
  
# municipios --------------------------------------------------------------  
municode <- read\_csv("municode.csv") %>%  
 select(id\_municipio, sigla\_uf, nome)  
  
m <- geobr::read\_municipality(year = 2018) %>%  
 select(id\_municipio = 1, geom) %>%  
 right\_join(municode) %>%  
 select(id\_municipio, sigla\_uf, nome, geom) %>%  
 st\_as\_sf()  
  
# teste -------------------------------------------------------------------  
# exemplo ananindeua - PA, amostra e centro  
  
# anan <- m %>% filter(id\_municipio == 1500800)  
# s <-st\_sample(anan$geom, size = 15)  
  
# ananindeua\_ports <- ggplot()+  
# geom\_sf(data = anan, fill = "grey")+  
# geom\_sf(data = s, color = "blue")+  
# geom\_sf(data = st\_centroid(anan), color = "red", size=4)+  
# theme(panel.background = element\_blank(),  
# axis.text = element\_blank(),  
# axis.ticks = element\_blank())  
  
# ggsave("infraestrutura/portos/ananindeua\_ports.png", ananindeua\_ports)  
  
# funcao distancia portos ao centro do muni  
  
dist <- function(id) {  
 set.seed(1)  
 m %>%  
 filter(id\_municipio == id) %>%  
 st\_centroid() %>%  
 st\_distance(pub$geometry, by\_element = TRUE)  
}  
  
# distancia de cada muni a cada porto  
munidist <- m$id\_municipio %>%  
 map(dist) %>%  
 set\_names(nm = m$id\_municipio) %>%  
 as\_tibble()  
  
# transpoe munidist e nomeia as colunas  
# menor distancia em km e seu inverso (o indicador)  
  
munidistt <- cbind(id\_municipio = names(munidist), t(munidist)) %>%  
 as\_tibble %>%  
 set\_names(c("id\_municipio", pub$nome\_porto)) %>%  
 mutate(across(-id\_municipio, ~ round(as.numeric(.) / 1000)),  
 id\_municipio = as.numeric(id\_municipio)) %>%  
 rowwise(id\_municipio) %>%  
 mutate(menor\_dist = min(c\_across(-1)),  
 sd21\_portos = 1 / menor\_dist) %>%  
 select(id\_municipio, menor\_dist, sd21\_portos, everything())  
  
  
df <- left\_join(municode, munidistt, keep = FALSE) %>%  
 arrange(-sd21\_portos)  
  
write\_excel\_csv(df, "infraestrutura/portos/sd21\_portos\_completo.xlsx")  
  
df2 <- df %>% select(1,2,3,5) %>% rename(i213 = sd21\_portos)  
  
  
write\_excel\_csv(df2, "dados\_finais/sd21\_portos.xlsx")  
#  
#  
# SD 21 - INFRAESTRUTURA - TRANSPORTE INTERURBANO  
  
library(tidyverse)  
library(geobr)  
library(sf)  
# rodovias ----------------------------------------------------------------  
# federais  
unzip(  
 "infraestrutura/rodovias/vw\_snv\_rod.zip",  
 exdir = "infraestrutura/rodovias/fedroads\_shapefile",  
 junkpaths = TRUE,  
 overwrite = TRUE  
)  
  
br <-  
 sf::st\_read("infraestrutura/rodovias/fedroads\_shapefile/vw\_snv\_rod.shp",  
 as\_tibble = TRUE) %>%  
 select(codigo = Codigo\_BR, geometry) %>%  
 mutate(codigo = paste0("br\_", codigo))  
  
# estaduais  
unzip(  
 "infraestrutura/rodovias/vw\_cide\_rod.zip",  
 exdir = "infraestrutura/rodovias/estroads\_shapefile",  
 junkpaths = TRUE,  
 overwrite = TRUE  
)  
  
er <-  
 st\_read("infraestrutura/rodovias/estroads\_shapefile/vw\_cide\_rod.shp",  
 as\_tibble = T) %>%  
 select(codigo = Codigo\_Rod, geometry) %>%  
 mutate(codigo = paste0("es\_", codigo))  
# rodovias federais e estaduais  
roads <- bind\_rows(br, er)  
  
# localizao dos municipios ------------------------------------------------  
municode <-  
 read\_csv("municode.csv") %>% select(id\_municipio, sigla\_uf, nome)  
  
m <- geobr::read\_municipality() %>%  
 select(id\_municipio = 1, geom) %>%  
 right\_join(municode) %>%  
 select(id\_municipio, sigla\_uf, nome, geom) %>%  
 st\_as\_sf()  
  
p <- geobr::read\_state() %>%  
 ggplot() +  
 geom\_sf(  
 fill = "white",  
 color = "grey",  
 size = .15,  
 show.legend = F  
 ) +  
 geom\_sf(data = m, fill = "#2D3E50") +  
 geom\_sf(data = roads %>% sample(1), color = "#FEBF57") +  
 theme(  
 panel.background = element\_blank(),  
 axis.title = element\_blank(),  
 axis.text = element\_blank(),  
 axis.ticks = element\_blank()  
 )  
  
ggsave("infraestrutura/rodovias/exemplo\_rodovia.png", p)  
  
# funcoes auxiliares ------------------------------------------------------  
roads\_inteiras <- roads %>% split(.$codigo)  
  
passam <- function(r) {  
 i <- 1:100  
 muninterr <- function(i) {  
 # verifica se municipio m intersecta algum trecho da rodovia r  
 st\_intersects(m[i, ], r, sparse = FALSE) %>% any()  
 }  
 i %>% map(possibly(muninterr, "erro"))  
}  
  
recebem <- roads\_inteiras %>% map(passam)  
# intersecao rodovias e municipios ----------------------------------------  
  
df <- recebem %>% map\_dfc(unlist) %>%  
 filter(!br\_010 == "erro") %>% # exclui Rio Branco  
 mutate(across(where(is.character), as.logical)) %>%  
 rowwise() %>%  
 mutate(n\_estradas = sum(c\_across(everything()))) %>%  
 add\_column(id\_municipio = m$id\_municipio[-2], .before = 1)  
  
df\_final <- df %>% select(1, ncol(df)) %>%  
 left\_join(municode) %>%  
 select(id\_municipio, sigla\_uf, nome, i211 = n\_estradas) %>%  
 add\_row(  
 id\_municipio = 1200401,  
 sigla\_uf = "AC",  
 nome = "Rio Branco",  
 i211 = 4  
 )  
  
write\_excel\_csv(df\_final, "dados\_finais/sd21\_rodovias.xlsx")

### Script sd22

voltar para [Condições Urbanas](#sd22)

#  
#  
# sd22 INFRAESTRUTURA - CONDICOES URBANAS - HOMICIDIOS  
  
library(tidyverse)  
  
# dados brutos  
municode <- read\_csv("municode.csv") %>%  
 select(id\_municipio\_6, id\_municipio, sigla\_uf,pop)  
  
# le, separa e funde com os dados de municipios  
h <-  
 read\_delim(  
 "infraestrutura/homicidios/dados\_homicidios",  
 skip = 1,  
 delim = ";",  
 escape\_double = FALSE,  
 trim\_ws = TRUE,  
 col\_names = c("municipio", "obitos")  
 ) %>%  
 separate(  
 col = municipio,  
 into = c("id\_municipio\_6", "municipio"),  
 sep = " ",  
 extra = "merge",  
 convert = TRUE  
 )  
  
# calcula a taxa e o indicador  
homicidios <- h %>%  
 right\_join(municode) %>%  
 mutate(taxa\_homicidios = obitos / pop,  
 i224 = 1 / taxa\_homicidios)  
  
write\_excel\_csv(homicidios, "infraestrutura/homicidios/sd22\_homicidios.xlsx")  
  
  
homicidios %>%   
 select(4,5,2,8) %>%   
 write\_excel\_csv("dados\_finais/sd22\_homicidios.xlsx")   
#  
#  
# SD22 - INFRAESTRUTURA - Acesso a internet rapida-----------------------------  
  
library(tidyverse)  
  
municode <- read\_csv("municode.csv") %>% select(id\_municipio, pop)  
  
df <- read\_delim(  
 file = "infraestrutura/internet/Acessos\_Banda\_Larga\_Fixa\_2019-2020\_Colunas.csv",  
 delim = ";",  
 escape\_double = FALSE,  
 trim\_ws = TRUE  
)  
  
internet <- df %>%  
 select(UF,  
 id\_municipio = `Código IBGE Município`,  
 Município,  
 `Faixa de Velocidade`,  
 `2020-01`:`2020-12`) %>%  
 filter(`Faixa de Velocidade` %in% c("> 34Mbps", "12Mbps a 34Mbps")) %>%  
 gather(`2020-01`:`2020-12`, key = "mes", value = "acessos") %>%  
 group\_by(id\_municipio, Município) %>%  
 summarise(acessos = sum(acessos, na.rm = T)) %>%  
 right\_join(municode, keep = F) %>%  
 mutate(acessos\_hab = acessos / pop)  
  
write\_excel\_csv(internet,  
 "infraestrutura/internet/sd22\_internet\_completo.xlsx")  
  
  
internet %>%   
 select(1, 2, i221 = 5) %>%   
 write\_excel\_csv("dados\_finais/sd22\_internet.xlsx")  
#  
#  
# SD22 - PRECO DO METRO QUADRADO ------------------------------------------  
  
# bibliotecas -------------------------------------------------------------  
library(httr)  
library(tidyverse)  
library(basedosdados)  
library(abjutils)  
  
# funcao para baixar dados do site ----------------------------------------  
  
get\_muni <- function(uf, muni) {  
 # import data-----------------------------------------------------------------  
 u <- "https://glue-api.zapimoveis.com.br/v2/listings"  
 query <- list(  
 business = "SALE",  
 categoryPage = "RESULT",  
 includeFields = "search",  
 listingType = "USED",  
 addressState = uf,  
 addressCity = muni,  
 portal = "ZAP",  
 addressType = "city",  
 size = "350"  
 )  
 h <- httr::add\_headers("X-domain" = "www.zapimoveis.com.br")  
   
 resultado <- GET(u, query = query, h) %>%  
 content() %>%  
 pluck("search", "result", "listings")  
   
 # explore -----------------------------------------------------------------  
 # substitui NULL (ausencia de dado) por NA (indica que nao ha dado)  
 nullToNA <- function(x) {  
 x[unlist(map(x, is.null))] <- NA  
 return(x)  
 }  
   
 area <- resultado %>%  
 map( ~ pluck(., "listing", "usableAreas")) %>%  
 nullToNA() %>%  
 unlist() %>%  
 as.numeric()  
   
 endereco <- resultado %>%  
 map( ~ pluck(., "listing", "address", "locationId")) %>%  
 nullToNA() %>%  
 unlist()  
   
 # preços  
 get\_price <- function(x) {  
 unlist(price[[x]])['price']  
 }  
   
 price <- resultado %>%  
 map( ~ pluck(., "listing", "pricingInfos")) %>%  
 nullToNA()  
   
 pricef <- seq\_along(price) %>% map(get\_price) %>%  
 unlist() %>%  
 as.numeric()  
   
 # tudo junto  
 imoveis <-  
 tibble(endereco = endereco,  
 price = pricef,  
 area = area)  
 imoveis  
   
}  
  
# importa dados dos municipios --------------------------------------------  
  
set\_billing\_id("ice2021")  
  
nome\_uf <- read\_sql(  
 "SELECT id\_municipio, nome\_uf  
 FROM `basedosdados.br\_bd\_diretorios\_brasil.municipio`"  
)  
  
municode <- read\_csv("municode.csv") %>%  
 mutate(id\_municipio = as.character(id\_municipio)) %>%  
 left\_join(nome\_uf) %>%  
 select(id\_municipio, nome\_uf, nome)  
  
# raspa os dados para cada municipio --------------------------------------  
  
terrenos <- map2\_dfr(municode$nome\_uf, municode$nome, get\_muni)  
  
# organiza e exclui dados faltanes  
terrenos\_final <- terrenos %>%  
 filter(area != 0 & !is.na(area)) %>%  
 separate(  
 endereco,  
 sep = ">",  
 into = c("país", "nome\_uf", "NULL", "nome", "bairro"),  
 extra = "merge"  
 ) %>%  
 select(nome\_uf, nome, bairro, price, area)  
  
gabarito <- terrenos\_final %>%  
 mutate(`m^2` = price / area) %>%  
 arrange(-`m^2`)  
  
  
df <- terrenos\_final %>%  
 mutate(m2 = price / area) %>%  
 filter(between(m2, 100, 20000)) %>%  
 group\_by(nome\_uf, nome) %>%  
 summarise(  
 price\_total = sum(price, na.rm = TRUE),  
 area\_total = sum(area),  
 amostra = n(),  
 m2 = price\_total / area\_total  
 )  
  
  
df\_final <- municode %>%  
 mutate(across(everything(), rm\_accent)) %>%  
 left\_join(df) %>%  
 mutate(s22\_m2 = 1 / m2)  
  
write\_excel\_csv(df\_final, "infraestrutura/terrenos/sd22\_m2\_completo.xlsx")

### Script sd31

voltar para [Desenvolvimento Econômico](#sd31)

#  
#  
# sd31 - MERCADOS - DESENVOLVIMENTO ECONÔMICO  
  
library(tidyverse)  
library(readxl)  
library(basedosdados)  
library(sidrar)  
library(abjutils)  
  
municode <- read\_csv("municode.csv") %>%  
 select(id\_municipio, sigla\_uf, nome) %>%  
 mutate(id\_municipio = as.character(id\_municipio))  
  
# idh ---------------------------------------------------------------------  
idh <- read\_excel("dados\_finais/sd31\_idh.xlsx",  
 col\_names = c("id\_municipio", "nome", "i311")) %>%  
 right\_join(municode) %>%  
 select(id\_municipio, sigla\_uf, nome, i311)  
  
write\_excel\_csv(idh, "dados\_finais/sd31\_idh.xlsx")  
  
  
# crescimento medio real do pib -------------------------------------------  
# pib nominal municipal  
  
basedosdados::set\_billing\_id("ice2021")  
  
pib\_mun <- basedosdados::read\_sql(query = "SELECT \*  
 FROM `basedosdados.br\_ibge\_pib.municipio`  
 WHERE ano >= 2014 AND ano <= 2018") %>%  
 select(1:3) %>%  
 pivot\_wider(names\_from = ano,  
 values\_from = pib,  
 names\_prefix = "pib\_") %>%  
 right\_join(municode) %>%  
 select(id\_municipio, sigla\_uf, nome, everything())  
  
# pib\_corrente e passado em milhoes de reais  
d <-  
 get\_sidra(  
 x = 6784,  
 period = c("last" = 5),  
 variable = c(9808, 9809)  
 ) %>%  
 as\_tibble() %>%  
 select(var = Variável,  
 ano = Ano,  
 valor = Valor) %>%  
 pivot\_wider(names\_from = var, values\_from = valor) %>%  
 rename(ano = 1,  
 pib\_corrente = 2,  
 pib\_passado = 3) %>%  
 rowwise() %>%  
 mutate(relativo = pib\_corrente / pib\_passado) %>%  
 pull(relativo)  
  
d <-  
 as.list(c(1, d[2], d[2] \* d[3], d[2] \* d[3] \* d[4], d[2] \* d[3] \* d[4] \*  
 d[5]) \* 100)  
d  
# deflacionar  
names(d) <- pib\_mun %>% select(where(is.numeric)) %>% names(.)  
  
pib\_mun\_real <- pib\_mun %>%  
 mutate(  
 across(all\_of(names(d)), ~ .x \* 100 / d[[cur\_column()]]),  
 var\_1415 = (pib\_2015 / pib\_2014) - 1,  
 var\_1516 = (pib\_2016 / pib\_2015) - 1,  
 var\_1617 = (pib\_2017 / pib\_2016) - 1,  
 var\_1718 = (pib\_2018 / pib\_2017) - 1  
 ) %>%  
 rowwise() %>%  
 mutate(sd312 = mean(c\_across(contains("var\_")))) %>%  
 arrange(-sd312)  
  
write\_excel\_csv(pib\_mun\_real, "mercado/sd31\_pib\_var.xlsx")  
  
pib\_mun\_real %>% select(1,2,3,13) %>%  
 write\_excel\_csv(pib\_mun\_real, "dados\_finais/sd31\_pib\_var.xlsx")  
  
# exportadoras ------------------------------------------------------------  
  
vinculos <- basedosdados::read\_sql(  
 "SELECT ano, id\_municipio, qtde\_vinculos\_ativos,tamanho\_estabelecimento  
 FROM `basedosdados.br\_me\_rais.microdados\_estabelecimentos`  
 WHERE ano = 2019",  
 page\_size = 300000  
)  
  
temvinculo <- vinculos %>%  
 filter(qtde\_vinculos\_ativos != 0) %>%  
 group\_by(id\_municipio) %>%  
 summarise(n\_empresas = n()) %>%  
 right\_join(municode) %>%  
 mutate(nome = nome %>% str\_to\_title() %>% rm\_accent())  
  
exportadoras <-  
 read\_excel("mercado/EMPRESAS\_CADASTRO\_2019.xlsx", skip = 7) %>%  
 select(sigla\_uf = UF, nome = MUNICÍPIO) %>%  
 count(sigla\_uf, nome, name = "n\_exp") %>%  
 mutate(nome = nome %>% str\_to\_title() %>% rm\_accent())  
  
df <- left\_join(temvinculo, exportadoras) %>%  
 select(id\_municipio, sigla\_uf, nome, n\_empresas, n\_exp) %>%  
 mutate(i313 = n\_exp / n\_empresas)  
  
write\_excel\_csv(df, "mercado/sd31\_exportadoras\_completo.xlsx")  
  
df %>% select(1, 2, 3, 6) %>%  
 write\_excel\_csv(df, "dados\_finais/sd31\_exportadoras.xlsx")

### Script sd32

voltar para [Clientes Potenciais](#sd32)

#  
#  
# SD32 MERCADOS - CLIENTES POTENCIAS  
  
library(tidyverse)  
library(basedosdados)  
  
municode <- read\_csv("municode.csv") %>%  
 select(id\_municipio, sigla\_uf, nome) %>%  
 mutate(id\_municipio = as.character(id\_municipio))  
  
# pib per capita ----------------------------------------------------------  
  
basedosdados::set\_billing\_id("ice2021")  
  
# puxa populacao e pib municipal de todos os municipios  
df <- tibble(  
 query = c(  
 "SELECT \*  
 FROM `basedosdados.br\_ibge\_populacao.municipio`  
 WHERE ano = 2018",  
   
 "SELECT id\_municipio, pib  
 FROM `basedosdados.br\_ibge\_pib.municipio`  
 WHERE ano = 2018"  
 )  
) %>%  
 mutate(resultados = query %>% map(~ basedosdados::read\_sql(.x))) %>%  
 pull(resultados) %>%  
 reduce(full\_join)  
  
# calcula pib per capita  
df2 <- df %>%  
 right\_join(municode) %>%  
 select(ano, id\_municipio, sigla\_uf,nome, populacao, pib) %>%  
 mutate(i321 = pib / populacao)  
  
  
write\_excel\_csv(df2, "mercado/sd32\_pib\_capita\_completo.xlsx")  
  
df2 %>% select(2:4,7) %>% write\_excel\_csv("dados\_finais/sd32\_pib\_capita.xlsx")  
  
# proporcao empresas grandes e medias sobre medias e pequenas -------------  
  
vinculos <- basedosdados::read\_sql(  
 "SELECT ano, id\_municipio, qtde\_vinculos\_ativos,tamanho\_estabelecimento  
 FROM `basedosdados.br\_me\_rais.microdados\_estabelecimentos`  
 WHERE ano = 2019 AND qtde\_vinculos\_ativos != 0",  
 page\_size = 200000 # linhas por página na query  
)  
  
df <- vinculos %>%  
 select(id\_municipio, qtde\_vinculos\_ativos) %>%  
 mutate(  
 qtde\_vinculos\_ativos = as.integer(qtde\_vinculos\_ativos),  
 pequena = between(qtde\_vinculos\_ativos, 10, 49),  
 media = between(qtde\_vinculos\_ativos, 50, 249),  
 grande = qtde\_vinculos\_ativos > 250  
 ) %>%  
 group\_by(id\_municipio) %>%  
 summarise(across(where(is.logical), ~ sum(.x))) %>%  
 right\_join(municode) %>%  
 select(id\_municipio, sigla\_uf, nome, everything()) %>%  
 mutate(m\_p = media / pequena,  
 g\_m = grande / media,  
 i322 = g\_m / m\_p) %>%  
 arrange(-i322)  
  
write\_csv(df, "mercado/sd32\_prop\_empresas\_completo.csv")  
  
df %>%   
 select(1:3,9) %>%   
 write\_csv("dados\_finais/sd32\_prop\_empresas.csv")

## Script i323

Este script em R refere-se aos [Indicador Compras Públicas (i323)](#i323)

library(readxl)  
library(tidyverse)  
library("basedosdados")  
  
# Indicador Compras Publicas (i323)  
  
# ------------------------------------------------------------------  
# PASSO 0: preliminares  
# ------------------------------------------------------------------  
  
# definimos uma funcao para desvio padrao populacional  
sdp <- function(x) {  
 sd(x)\*(sqrt((length(x)-1)/length(x)))  
}  
  
# importamos tabela com dados dos municipios mais populosos  
top100\_mun\_cod <- read.csv("top100\_mun\_cod.csv", stringsAsFactors = FALSE)  
top100\_mun\_cod <- top100\_mun\_cod %>%   
 transform(id\_municipio = as.character(id\_municipio))  
  
# Definir projeto no Google Cloud  
set\_billing\_id("workshop-teste-322616")  
  
# ------------------------------------------------------------------  
# PASSO 1: obter despesas e investimentos das prefeituras  
# ------------------------------------------------------------------  
  
# importar planilha com dados de despesas orcamentarias dos municipios  
despesas <- read.csv2('C:/Users/User/OneDrive/Documentos/ICE 2021/  
 d3 Mercado/sd32/i323/finbra.csv', skip = 3)   
   
despesas\_mun <- despesas %>%   
 select(id\_municipio = 'Cod.IBGE',  
 sigla\_uf = UF,  
 Conta,  
 Valor) %>%  
 filter(Conta == '3.0.00.00.00 - Despesas Correntes' |   
 Conta == '4.4.00.00.00 - Investimentos') %>%  
 group\_by(id\_municipio, sigla\_uf) %>%  
 summarise(desp\_inv = sum(Valor)) %>%  
 transform(id\_municipio = as.character(id\_municipio)) %>%  
 inner\_join(top100\_mun\_cod, by = c('id\_municipio', 'sigla\_uf')) %>%  
 replace\_na(list(desp\_inv = 0))  
   
# importar planilha com dados de brasilia  
despesas\_df <- read.csv2('C:/Users/User/OneDrive/Documentos/ICE 2021/  
 d3 Mercado/sd32/i323/finbra-df.csv', skip = 3)   
  
despesas\_df <- despesas\_df %>%   
 select(id\_estado = 'Cod.IBGE',  
 sigla\_uf = UF,  
 Conta,  
 Valor) %>%  
 filter(Conta == '3.0.00.00.00 - Despesas Correntes' |  
 Conta == '4.4.00.00.00 - Investimentos',  
 sigla\_uf == 'DF') %>%  
 group\_by(sigla\_uf) %>%  
 summarise(desp\_inv = sum(Valor))   
  
despesas\_ice <- bind\_rows(despesas\_df, despesas\_mun)  
despesas\_ice[1,3] = '5300108'  
despesas\_ice[1,4] = 'Brasília'  
  
despesas\_ice <- despesas\_ice %>%  
 right\_join(top100\_mun\_cod, by = c('id\_municipio', 'sigla\_uf', 'nome')) %>%  
 replace\_na(list(desp\_inv = 0)) %>%  
 select(id\_municipio, nome, sigla\_uf, desp\_inv) %>%  
 arrange(desc(desp\_inv))  
  
# ------------------------------------------------------------------  
# PASSO 2: obter numero de empresas com pelo menos 1 funcionario  
# ------------------------------------------------------------------  
  
# acessar RAIS estabelecimento pelo Base dos Dados  
# obter numero de empresas com pelo menos um funcionario em cada municipio  
vinculos <- basedosdados::read\_sql(  
 "SELECT ano, id\_municipio, qtde\_vinculos\_ativos,tamanho\_estabelecimento   
 FROM `basedosdados.br\_me\_rais.microdados\_estabelecimentos`   
 WHERE ano = 2019",   
 page\_size = 300000)  
  
temvinculo <- vinculos %>%   
 filter(qtde\_vinculos\_ativos != 0) %>%  
 group\_by(id\_municipio) %>%  
 summarise(n\_empresas = n()) %>%  
 right\_join(top100\_mun\_cod, by = "id\_municipio") %>%  
 select(id\_municipio, nome, sigla\_uf,n\_empresas) %>%  
 arrange(desc(n\_empresas))  
  
# ------------------------------------------------------------------  
# PASSO 3: juntar tabelas e calcular indicador  
# ------------------------------------------------------------------  
  
i323 <- despesas\_ice %>%  
 inner\_join(temvinculo, by = c('id\_municipio', 'nome', 'sigla\_uf')) %>%  
 mutate(i323 = desp\_inv/n\_empresas,  
 i323\_pad = (i323 - mean(i323))/sdp(i323)) %>%  
 select(id\_municipio, nome, sigla\_uf, i323, i323\_pad) %>%  
 arrange(desc(i323\_pad))  
   
# exportar indicador  
write.csv(i323, "i323.csv", row.names = FALSE)

## Script i411

Este scrip em R refere-se ao [Indicador Operações de Crédito por Município (i411)](#i411)

library(readxl)  
library(tidyverse)  
library("basedosdados")  
  
# Indicador Operacoes de Credito por Municipio (i411)  
  
# ------------------------------------------------------------------  
# PASSO 0: preliminares  
# ------------------------------------------------------------------  
  
# definimos uma funcao para desvio padrao populacional  
sdp <- function(x) {  
 sd(x)\*(sqrt((length(x)-1)/length(x)))  
}  
  
# Definir projeto no Google Cloud  
set\_billing\_id("workshop-teste-322616")  
  
# obter codigos de municipio do banco central  
query <- "SELECT \* FROM `basedosdados.br\_bd\_diretorios\_brasil.municipio`"  
cod\_mun <- read\_sql(query) %>% select(id\_municipio, id\_municipio\_bcb)  
  
# importamos tabela com dados dos municipios mais populosos  
top100\_mun\_cod <- read.csv("top100\_mun\_cod.csv", stringsAsFactors = FALSE)  
top100\_mun\_cod <- top100\_mun\_cod %>%   
 transform(id\_municipio = as.character(id\_municipio)) %>%  
 inner\_join(cod\_mun, by = "id\_municipio")  
  
# ------------------------------------------------------------------  
# PASSO 1: obter valor em reais das operacoes de credito  
# ------------------------------------------------------------------  
  
# importar tabela com dados de operacoes de credito (dez. 2020)   
op\_cred <- read.csv2('C:/Users/User/OneDrive/Documentos/ICE 2021/  
 d4 Acesso a Capital/sd41/  
 i411/202012\_ESTBAN.csv', skip =2) %>%  
 as\_tibble() %>%  
 transform(CODMUN = as.character(CODMUN)) %>%  
 replace\_na(list(VERBETE\_160\_OPERACOES\_DE\_CREDITO = 0)) %>%  
 select(id\_municipio\_bcb = CODMUN,  
 op\_cred = VERBETE\_160\_OPERACOES\_DE\_CREDITO)  
  
op\_cred\_ice <- op\_cred %>%   
 inner\_join(top100\_mun\_cod, by = 'id\_municipio\_bcb') %>%  
 group\_by(id\_municipio, nome, sigla\_uf) %>%  
 summarise(valor\_total = sum(op\_cred))  
   
# ------------------------------------------------------------------  
# PASSO 2: obter PIB dos municipios  
# ------------------------------------------------------------------  
  
# importar tabela com PIB a nivel de municipios  
# os nomes das colunas foram previamente alterados no excel  
pib <- read.csv2('C:/Users/User/OneDrive/Documentos/ICE 2021/  
 d4 Acesso a Capital/sd41/i411/tabela5938.csv', skip = 3)  
  
pib\_ice <- pib %>%   
 inner\_join(top100\_mun\_cod, by = 'id\_municipio') %>%  
 mutate(pib = pib\_milreais \* 1000) %>%  
 select(id\_municipio, nome = nome.y, sigla\_uf, pib)  
  
# ------------------------------------------------------------------  
# PASSO 3: calcular indicador  
# ------------------------------------------------------------------  
  
i411 <- pib\_ice %>%   
 inner\_join(op\_cred\_ice, by = c('id\_municipio', 'nome', 'sigla\_uf')) %>%  
 mutate(i411 = valor\_total/pib,  
 i411\_pad = (i411 - mean(i411))/sdp(i411)) %>%  
 select(id\_municipio, nome, sigla\_uf, i411, i411\_pad) %>%  
 arrange(desc(i411\_pad))  
  
# exportar indicador   
write.csv(i411, "i411.csv", row.names = FALSE)

## Script i412

Este scrip em R refere-se ao [Indicador Proporção Relativa de Capital de Risco (i412)](#i412)

library(readxl)  
library(tidyverse)  
  
# Indicador Proporcao Relativa de Capital de Risco (i412)  
  
# ------------------------------------------------------------------  
# PASSO 0: preliminares  
# ------------------------------------------------------------------  
  
# definimos uma funcao para desvio padrao populacional  
sdp <- function(x) {  
 sd(x)\*(sqrt((length(x)-1)/length(x)))  
}  
  
# importamos tabela com dados dos municipios mais populosos  
top100\_mun\_cod <- read.csv("top100\_mun\_cod.csv", stringsAsFactors = FALSE)  
top100\_mun\_cod <- top100\_mun\_cod %>%   
 transform(id\_municipio = as.character(id\_municipio))  
  
# importar tabela pib municipal  
# dado levantado para o indicador pib per capita (i321)  
  
pib\_mun <- read.csv('pib\_mun.csv', stringsAsFactors = FALSE)  
pib\_mun$id\_municipio = as.character(pib\_mun$id\_municipio)  
  
# ------------------------------------------------------------------  
# PASSO 1: obter capital de risco por municipio no ultimo ano  
# ------------------------------------------------------------------  
  
# importar tabela construida manualmente a partir do site crunchbase  
# taxa media de cambio do ultimo ano: 5.360819 para dolar e 6.408495 para euro  
caprisco <- read\_excel('C:/Users/User/OneDrive/Documentos/ICE 2021/  
 d4 Acesso a Capital/sd41/i412/caprisco.xlsx', sheet = 2)  
  
caprisco\_dolar <- caprisco %>%   
 filter(dolar ==1) %>%  
 mutate(valor = valor\*5.360819) %>%  
 select(nome, sigla\_uf, empresa, valor)  
  
caprisco\_real <- caprisco %>%   
 filter(real ==1) %>%  
 select(nome, sigla\_uf, empresa, valor)  
  
caprisco\_euro <- caprisco %>%   
 filter(euro ==1) %>%  
 mutate(valor = valor\*6.408495) %>%  
 select(nome, sigla\_uf, empresa, valor)  
  
caprisco\_ice <- rbind(caprisco\_real, caprisco\_dolar, caprisco\_euro) %>%  
 group\_by(nome, sigla\_uf) %>%  
 summarise(valor\_total = sum(valor)) %>%  
 right\_join(top100\_mun\_cod, by = c('nome', 'sigla\_uf')) %>%  
 replace\_na(list(valor\_total = 0)) %>%  
 inner\_join(pib\_mun, by = c('id\_municipio', 'nome', 'sigla\_uf')) %>%  
 select(id\_municipio, nome, sigla\_uf, valor\_total, pib)  
  
# ------------------------------------------------------------------  
# PASSO 1: calcular e padronizar indicador  
# ------------------------------------------------------------------  
  
i412 <- caprisco\_ice %>%  
 transform(valor\_total = as.numeric(valor\_total)) %>%  
 mutate(i412 = valor\_total/pib,  
 i412\_pad = (i412 - mean(i412))/sdp(i412)) %>%  
 select(id\_municipio, nome, sigla\_uf, i412, i412\_pad) %>%  
 arrange(desc(i412\_pad))  
  
# exportar indicador  
write.csv(i412, "i412.csv", row.names = FALSE)

## Script i413

Este scrip em R refere-se ao [Indicador Capital Poupado per capita (i413)](#i413)

library(readxl)  
library(tidyverse)  
library("basedosdados")  
  
# Indicador Indicador Capital Poupado per capita (i413)  
  
# ------------------------------------------------------------------  
# PASSO 0: preliminares  
# ------------------------------------------------------------------  
  
# definimos uma funcao para desvio padrao populacional  
sdp <- function(x) {  
 sd(x)\*(sqrt((length(x)-1)/length(x)))  
}  
  
# Definir projeto no Google Cloud  
set\_billing\_id("workshop-teste-322616")  
  
# obter codigos de municipio do banco central  
query <- "SELECT \* FROM `basedosdados.br\_bd\_diretorios\_brasil.municipio`"  
cod\_mun <- read\_sql(query) %>% select(id\_municipio, id\_municipio\_bcb)  
  
# importamos tabela com dados dos municipios mais populosos  
top100\_mun\_cod <- read.csv("top100\_mun\_cod.csv", stringsAsFactors = FALSE)  
top100\_mun\_cod <- top100\_mun\_cod %>%   
 transform(id\_municipio = as.character(id\_municipio)) %>%  
 inner\_join(cod\_mun, by = "id\_municipio")  
  
# ------------------------------------------------------------------  
# PASSO 1: obter valor de depositos em poupanca e a prazo  
# ------------------------------------------------------------------  
  
# importar tabela com dados de operacoes de credito (dez. 2020)  
# notar que a tabela e a mesma utilizada em i411  
op\_dep <- read.csv2('C:/Users/User/OneDrive/Documentos/ICE 2021/  
 d4 Acesso a Capital/sd41/i411/  
 202012\_ESTBAN.csv', skip =2) %>%  
 as\_tibble() %>%  
 transform(CODMUN = as.character(CODMUN)) %>%  
 replace\_na(list(VERBETE\_420\_DEPOSITOS\_DE\_POUPANCA = 0,  
 VERBETE\_432\_DEPOSITOS\_A\_PRAZO = 0)) %>%  
 select(id\_municipio\_bcb = CODMUN,  
 poup = VERBETE\_420\_DEPOSITOS\_DE\_POUPANCA,  
 prazo = VERBETE\_432\_DEPOSITOS\_A\_PRAZO)  
  
op\_dep\_ice <- op\_dep %>%   
 inner\_join(top100\_mun\_cod, by = 'id\_municipio\_bcb') %>%  
 group\_by(id\_municipio, nome, sigla\_uf) %>%  
 summarise(soma\_poup = sum(poup),  
 soma\_prazo = sum(prazo)) %>%  
 mutate(poup\_prazo = soma\_poup + soma\_prazo) %>%  
 select(id\_municipio, nome, sigla\_uf, poup\_prazo)  
  
# ------------------------------------------------------------------  
# PASSO 2: obter populacao estimada dos municipios  
# ------------------------------------------------------------------  
  
# obter estimativa populacionao de 2020  
pop\_ice <- read\_excel('C:/Users/User/OneDrive/Documentos/ICE 2021/  
 estimativa\_dou\_2021.xls', sheet = 2, skip = 1) %>%  
 na.omit() %>%  
 select(nome = 'NOME DO MUNIC?PIO',  
 sigla\_uf = 'UF',  
 populacao = 'POPULA??O ESTIMADA') %>%  
 inner\_join(top100\_mun\_cod, by = c('nome', 'sigla\_uf')) %>%  
 select(id\_municipio, nome, sigla\_uf, populacao)  
  
# ------------------------------------------------------------------  
# PASSO 3: calcular indicador  
# ------------------------------------------------------------------  
  
i413 <- pop\_ice %>%   
 inner\_join(op\_dep\_ice, by = c('id\_municipio', 'nome', 'sigla\_uf')) %>%  
 mutate(i413 = poup\_prazo/populacao,  
 i413\_pad = (i413 - mean(i413))/sdp(i413)) %>%  
 select(id\_municipio, nome, sigla\_uf, i413, i413\_pad) %>%  
 arrange(desc(i413\_pad))  
  
# exportar indicador   
write.csv(i413, "i413.csv", row.names = FALSE)

## Script i511

Este script em R refere-se ao [Indicador Proporção de Mestres e Doutores em C&T (i511)](#i511)

library(readxl)  
library(tidyverse)  
library("basedosdados")  
  
# Indicador Proporcao de Mestres e Doutores em C&T (i511)  
  
# ------------------------------------------------------------------  
# PASSO 0: preliminares  
# ------------------------------------------------------------------  
  
# Definir projeto no Google Cloud  
set\_billing\_id("workshop-teste-322616")  
  
# definimos uma funcao para desvio padrao populacional  
sdp <- function(x) {  
 sd(x)\*(sqrt((length(x)-1)/length(x)))  
}  
  
# importamos tabela com dados dos municipios mais populosos  
top100\_mun\_cod <- read.csv("top100\_mun\_cod.csv", stringsAsFactors = FALSE)  
top100\_mun\_cod <- top100\_mun\_cod %>%   
 transform(id\_municipio = as.character(id\_municipio))  
  
# criar variavel com municipios ICE em letra maiuscula   
# (dados da CAPES tem municipios com letra maiuscula)  
top100\_mun\_cod$NOME = toupper(top100\_mun\_cod$nome)  
  
# ------------------------------------------------------------------  
# PASSO 1: obter dados sobre discentes  
# ------------------------------------------------------------------  
  
# obter dados sobre discentes da CAPES para o ano de referencia 2019  
discentes <- read.csv2("C:/Users/User/OneDrive/Documentos/ICE 2021/d5 Inovação/  
 sd51 Inputs/i511/  
 br-capes-colsucup-discentes-2019-2021-03-01.csv",   
 row.names=NULL)  
  
# definir vetor com areas de avaliacao de interesse  
alvo <- c('ASTRONOMIA / FÍSICA',  
 'BIOTECNOLOGIA',  
 'CIÊNCIA DA COMPUTAÇÃO',   
 'CIÊNCIA DE ALIMENTOS',   
 'CIÊNCIAS AGRÁRIAS I', 'CIÊNCIAS AMBIENTAIS',   
 'CIÊNCIAS BIOLÓGICAS I',   
 'CIÊNCIAS BIOLÓGICAS II',   
 'CIÊNCIAS BIOLÓGICAS III',   
 'ENGENHARIAS I',   
 'ENGENHARIAS II',   
 'ENGENHARIAS III',   
 'ENGENHARIAS IV',   
 'FARMÁCIA',   
 'GEOCIÊNCIAS',   
 'MATEMÁTICA / PROBABILIDADE E ESTATÍSTICA',   
 'MATERIAIS', 'QUÍMICA')  
  
# filtrar para obter apenas as situacoes "titulado"   
# e as areas de interesse e municipios de interesse  
disc\_titulados <- discentes %>% filter(NM\_SITUACAO\_DISCENTE == "TITULADO",  
 NM\_AREA\_AVALIACAO %in% alvo) %>%  
 rename(sigla\_uf = SG\_UF\_PROGRAMA,  
 NOME = NM\_MUNICIPIO\_PROGRAMA\_IES) %>%  
 inner\_join(top100\_mun\_cod, by = c("sigla\_uf", "NOME")) %>%  
 count(sigla\_uf, nome, name = "disc\_titulados") %>%  
 right\_join(top100\_mun\_cod, by = c("sigla\_uf","nome")) %>%  
 select(id\_municipio, nome, sigla\_uf, disc\_titulados) %>%  
 replace\_na(list(disc\_titulados = 0))  
  
# ------------------------------------------------------------------  
# PASSO 2: obter dados sobre empresas  
# ------------------------------------------------------------------  
  
# acessar RAIS estabelecimento pelo Base dos Dados  
# para obter numero de empresas com pelo menos um funcionario em cada municipio  
vinculos <- basedosdados::read\_sql(  
 "SELECT ano, id\_municipio, qtde\_vinculos\_ativos,tamanho\_estabelecimento   
 FROM `basedosdados.br\_me\_rais.microdados\_estabelecimentos`   
 WHERE ano = 2019",   
 page\_size = 300000  
)  
  
temvinculo <- vinculos %>%   
 filter(qtde\_vinculos\_ativos != 0) %>%  
 group\_by(id\_municipio) %>%  
 summarise(n\_empresas = n()) %>%  
 right\_join(top100\_mun\_cod) %>%  
 select(id\_municipio, nome, sigla\_uf,n\_empresas) %>%  
 arrange(desc(n\_empresas))  
  
# ------------------------------------------------------------------  
# PASSO 3: calcular indicador  
# ------------------------------------------------------------------  
  
# calcular indicador i511  
i511 <- temvinculo %>%   
 left\_join(disc\_titulados, by = c('id\_municipio', 'nome', 'sigla\_uf')) %>%  
 mutate(i511 = disc\_titulados/n\_empresas) %>%   
 mutate(i511\_pad = (i511 - mean(i511))/sdp(i511)) %>%  
 select(id\_municipio, nome, sigla\_uf, i511, i511\_pad) %>%  
 arrange(desc(i511\_pad))  
   
# salvamos o arquivo  
write.csv(i511, "i511.csv", row.names = FALSE)

## Script i512

Este script em R refere-se ao [Indicador Proporção de Funcionários em C&T (i512)](#i512)

library(readxl)  
library(tidyverse)  
library("basedosdados")  
  
# Indicador Proporcao de Funcionarios em C&T (i512)  
  
# ------------------------------------------------------------------  
# PASSO 0: preliminares  
# ------------------------------------------------------------------  
  
# Definir projeto no Google Cloud  
set\_billing\_id("workshop-teste-322616")  
  
# definimos uma função para desvio padrao populacional para ser usada mais a frente no código  
sdp <- function(x) {  
 sd(x)\*(sqrt((length(x)-1)/length(x)))  
}  
  
# importamos tabela com dados dos municipios mais populosos  
top100\_mun\_cod <- read.csv("top100\_mun\_cod.csv", stringsAsFactors = FALSE)  
top100\_mun\_cod <- top100\_mun\_cod %>%   
 transform(id\_municipio = as.character(id\_municipio))  
# ------------------------------------------------------------------  
# PASSO 1: obter trabalhadores em ocupacoes C&T  
# ------------------------------------------------------------------  
  
# importamos tabela com ocupacoes de interesse de acordo com CBO (2002)  
# essa tabela nao e usada diretamente neste codigo, mas e usada para auxiliar a criacao da query no Banco dos Dados  
ocupacoes\_cet <- read\_excel("C:/Users/User/OneDrive/Documentos/ICE 2021/d5 Inova??o/sd51 Inputs/i512/CBO2002 - Ocupa??o Subgrupo Principal 20 21 30 31 39 72 73.xlsx")  
ocupacoes\_cet$cod\_cet = as.character(ocupacoes\_cet$cod\_cet)  
  
# consultamos na RAIS os vinculos de trabalho do ano 2019  
query <- "SELECT valor\_remun\_media\_nominal, ano, id\_municipio, cbo\_2002   
FROM `basedosdados.br\_me\_rais.microdados\_vinculos`   
WHERE ano = 2019  
AND cbo\_2002 IN ('201105','201110','201115','201205','201210','201215','201220','201225','202105','202110'  
,'202115','202120','203005','203010','203015','203020','203025','203105','203110','203115'  
,'203120','203125','203205','203210','203215','203220','203225','203230','203305','203310'  
,'203315','203320','203405','203410','203415','203420','203505','203510','203515','203520'  
,'203525','204105','211105','211110','211115','211120','211205','211210','211215','212205'  
,'212210','212215','212305','212310','212315','212320','212405','212410','212415','212420'  
,'212425','212430','213105','213110','213115','213120','213125','213130','213135','213140'  
,'213145','213150','213155','213160','213165','213170','213175','213205','213210','213215'  
,'213305','213310','213315','213405','213410','213415','213420','213425','213430','213435'  
,'213440','214005','214010','214105','214110','214115','214120','214125','214130','214205'  
,'214210','214215','214220','214225','214230','214235','214240','214245','214250','214255'  
,'214260','214265','214270','214280','214305','214310','214315','214320','214325','214330'  
,'214335','214340','214345','214350','214360','214365','214370','214405','214410','214415'  
,'214420','214425','214430','214435','214505','214510','214515','214520','214525','214530'  
,'214535','214605','214610','214615','214705','214710','214715','214720','214725','214730'  
,'214735','214740','214745','214750','214805','214810','214905','214910','214915','214920'  
,'214925','214930','214935','214940','214945','215105','215110','215115','215120','215125'  
,'215130','215135','215140','215145','215150','215205','215210','215215','215220','215305'  
,'215310','215315','300105','300110','300305','301105','301110','301115','301205','311105'  
,'311110','311115','311205','311305','311405','311410','311505','311510','311515','311520'  
,'311605','311610','311615','311620','311625','311705','311710','311715','311720','311725'  
,'312105','312205','312210','312305','312310','312315','312320','313105','313110','313115'  
,'313120','313125','313130','313205','313210','313215','313220','313305','313310','313315'  
,'313320','313405','313410','313415','313505','314105','314110','314115','314120','314125'  
,'314205','314210','314305','314310','314315','314405','314410','314610','314615','314620'  
,'314625','314705','314710','314715','314720','314725','314730','314805','314810','314815'  
,'314825','314830','314835','314840','314845','316105','316110','316115','316120','316305'  
,'316310','316315','316320','316325','316330','316335','316340','317105','317110','317115'  
,'317120','317205','317210','318005','318010','318015','318105','318110','318115','318120'  
,'318205','318210','318215','318305','318310','318405','318410','318415','318420','318425'  
,'318430','318505','318510','318605','318610','318705','318710','318805','318810','318815'  
,'319105','319110','319205','391105','391110','391115','391120','391125','391130','391135'  
,'391140','391145','391205','391210','391215','391220','391225','391230','395105','395110'  
,'720105','720110','720115','720120','720125','720130','720135','720140','720145','720150'  
,'720155','720160','720205','720210','720215','720220','721105','721110','721115','721205'  
,'721210','721215','721220','721225','721305','721310','721315','721320','721325','721405'  
,'721410','721415','721420','721425','721430','722105','722110','722115','722205','722210'  
,'722215','722220','722225','722230','722235','722305','722310','722315','722320','722325'  
,'722330','722405','722410','722415','723105','723110','723115','723120','723125','723205'  
,'723210','723215','723220','723225','723230','723235','723240','723305','723310','723315'  
,'723320','723325','723330','724105','724110','724115','724120','724125','724130','724135'  
,'724205','724210','724215','724220','724225','724230','724305','724310','724315','724320'  
,'724325','724405','724410','724415','724420','724425','724430','724435','724440','724505'  
,'724510','724515','724605','724610','725005','725010','725015','725020','725025','725105'  
,'725205','725210','725215','725220','725225','725305','725310','725315','725320','725405'  
,'725410','725415','725420','725505','725510','725605','725610','725705','730105','731105'  
,'731110','731115','731120','731125','731130','731135','731140','731145','731150','731155'  
,'731160','731165','731170','731175','731180','731205','731305','731310','731315','731320'  
,'731325','731330','732105','732110','732115','732120','732125','732130','732135','732140')  
AND id\_municipio IN ('3550308','3304557','5300108','2927408','2304400','3106200','1302603','4106902','2611606','5208707','1501402','4314902','3518800'  
,'3509502','2111300','3304904','2704302','3301702','5002704','2408102','2211001','3548708','3303500','2507507','3549904','3547809'  
,'3543402','2607901','3534401','3170206','3552205','3118601','2800308','2910800','5103403','4209102','5201405','4113700','3136702'  
,'1100205','1500800','3205002','4305108','3303302','3300456','1600303','3301009','4205407','3205200','3529401','3305109','3549805'  
,'3530607','3106705','3548500','4115200','3513801','3525904','1400100','3143302','1200401','2504009','3538709','3510609','2609600'  
,'5201108','3201308','3506003','3523107','3551009','3205309','2604106','2303709','4202404','3516200','4119905','2611101','4304606'  
,'4314407','2933307','3154606','3170107','2610707','4104808','3541000','4125506','3518701','3554102','3526902','3303906','1506807'  
,'1721000','2905701','2408003','3552502','3552809','5108402','3552403','4316907','4309209', '1504208')"  
ocupacoes\_cet <- read\_sql(query)  
  
ocupacoes\_cet\_ice <- ocupacoes\_cet %>% count(id\_municipio, name = "n\_cet") %>%  
 inner\_join(top100\_mun\_cod, by = 'id\_municipio') %>%  
 select(id\_municipio, sigla\_uf, nome, n\_cet) %>%  
 arrange(desc(n\_cet))  
  
# ------------------------------------------------------------------  
# PASSO 2: obter total de trabalhadores  
# ------------------------------------------------------------------  
  
# obtemos na RAIS total de trabalhadores em cada municipio de interesse  
query <- "SELECT id\_municipio, count(\*)   
FROM `basedosdados.br\_me\_rais.microdados\_vinculos`   
WHERE ano = 2019  
AND id\_municipio IN ('3550308','3304557','5300108','2927408','2304400','3106200','1302603','4106902','2611606','5208707','1501402','4314902','3518800'  
,'3509502','2111300','3304904','2704302','3301702','5002704','2408102','2211001','3548708','3303500','2507507','3549904','3547809'  
,'3543402','2607901','3534401','3170206','3552205','3118601','2800308','2910800','5103403','4209102','5201405','4113700','3136702'  
,'1100205','1500800','3205002','4305108','3303302','3300456','1600303','3301009','4205407','3205200','3529401','3305109','3549805'  
,'3530607','3106705','3548500','4115200','3513801','3525904','1400100','3143302','1200401','2504009','3538709','3510609','2609600'  
,'5201108','3201308','3506003','3523107','3551009','3205309','2604106','2303709','4202404','3516200','4119905','2611101','4304606'  
,'4314407','2933307','3154606','3170107','2610707','4104808','3541000','4125506','3518701','3554102','3526902','3303906','1506807'  
,'1721000','2905701','2408003','3552502','3552809','5108402','3552403','4316907','4309209', '1504208')  
GROUP BY id\_municipio"  
trabalhadores <- read\_sql(query, page\_size = 300000)  
  
trabalhadores\_ice <- trabalhadores %>% inner\_join(top100\_mun\_cod, by = "id\_municipio") %>%  
 rename(trabalhadores = f0\_) %>%  
 select(id\_municipio, nome, sigla\_uf, trabalhadores) %>%  
 arrange(desc(trabalhadores))  
  
# ------------------------------------------------------------------  
# PASSO 3: calcular indice  
# ------------------------------------------------------------------  
  
# calcular e padronizar indice i512  
i512 <- trabalhadores\_ice %>%   
 inner\_join(ocupacoes\_cet\_ice, by = c('id\_municipio','nome', 'sigla\_uf')) %>%  
 mutate(i512 = n\_cet/trabalhadores) %>%  
 mutate(i512\_pad = (i512 - mean(i512))/sdp(i512)) %>%  
 arrange(desc(i512\_pad)) %>%  
 select(id\_municipio, nome, sigla\_uf, i512, i512\_pad)  
  
# exportar arquivo  
write.csv(i512, "i512.csv", row.names = FALSE)

## Script i513

Este script em R refere-se ao [Indicador Média de Investimentos do BNDES e da FINEP (i513)](#i513)

library(readxl)  
library(tidyverse)  
library("basedosdados")  
library('readODS')  
  
# Indicador Media de Investimentos do BNDES e da FINEP (i513)  
  
# ------------------------------------------------------------------  
# PASSO 0 : preliminares  
# ------------------------------------------------------------------  
  
# Definir projeto no Google Cloud  
set\_billing\_id("workshop-teste-322616")  
  
# definimos uma funÃ§Ã£o para desvio padrao populacional   
sdp <- function(x) {  
 sd(x)\*(sqrt((length(x)-1)/length(x)))  
}  
  
# importamos tabela com dados dos municipios mais populosos  
top100\_mun\_cod <- read.csv("top100\_mun\_cod.csv", stringsAsFactors = FALSE)  
top100\_mun\_cod <- top100\_mun\_cod %>%   
 transform(id\_municipio = as.character(id\_municipio))  
top100\_mun\_cod$NOME = toupper(top100\_mun\_cod$nome)  
  
# ------------------------------------------------------------------  
# PASSO 1 : obter dados do BNDES  
# ------------------------------------------------------------------  
  
# importar tabela do BNDES com informacoes sobre investimentos nos municipios  
bndes <- read\_excel("C:/Users/User/OneDrive/Documentos/ICE 2021/d5 Inovação/  
 sd51 Inputs/i513/naoautomaticas.xlsx", skip = 4)  
  
# selecionamos apenas os municipios do ICE   
# agrupamos e somamos por municipio os valores de 2020  
bndes\_ice <- top100\_mun\_cod %>%   
 left\_join(bndes, by = c('id\_municipio' = 'Município - código')) %>%  
 rename(data = "Data da contratação",  
 valor = "Valor contratado R$") %>%  
 filter(data >= as.POSIXct("2020-01-01"),  
 data <= as.POSIXct("2020-12-31")) %>%  
 group\_by(id\_municipio, nome, sigla\_uf) %>%  
 summarise(contratado = sum(valor)) %>%  
 arrange(desc(contratado)) %>%  
 right\_join(top100\_mun\_cod, by = c('id\_municipio',  
 'nome','sigla\_uf')) %>%  
 replace\_na(list(contratado = 0))  
  
# ------------------------------------------------------------------  
# PASSO 2: obter dados da FINEP  
# ------------------------------------------------------------------  
  
# importamos a tabela com informacoes sobre projetos contratados da FINEP  
finep <- read\_ods("C:/Users/User/OneDrive/Documentos/ICE 2021/d5 Inovação/  
 sd51 Inputs/i513/17\_08\_2021\_Contratacao.ods", skip = 5)  
  
# alterar nomes e tipo de colunas de interesse  
finep <- finep %>% rename(data = "Data Assinatura",  
 valor = "Valor Finep",   
 NOME = "Município",   
 sigla\_uf = "UF")  
finep$data = as.POSIXct(finep$data, format = "%d/%m/%Y")  
finep$NOME = toupper(finep$NOME)  
  
# filtrar para obter apenas contratos do ano de 2020   
# agrupar e somar valores de cada municipio  
finep\_2020 <- finep %>%   
 filter(data >= as.POSIXct("2020-01-01"),  
 data <= as.POSIXct("2020-12-31")) %>%  
 group\_by(NOME, sigla\_uf) %>%  
 summarise(contratado = sum(valor))  
  
# separar municipios de interesse  
finep\_ice <- top100\_mun\_cod %>%   
 left\_join(finep\_2020, by = c('NOME', 'sigla\_uf')) %>%  
 replace\_na(list(contratado = 0)) %>%  
 select(-NOME) %>%  
 arrange(desc(contratado))  
  
# ------------------------------------------------------------------  
# PASSO 3: obter numero de empresas com pelo menos um funcionario  
# ------------------------------------------------------------------  
  
# acessar RAIS estabelecimento pelo Base dos Dados   
# obter numero de empresas com pelo menos um funcionario em cada municipio  
vinculos <- basedosdados::read\_sql(  
 "SELECT ano, id\_municipio, qtde\_vinculos\_ativos,tamanho\_estabelecimento   
 FROM `basedosdados.br\_me\_rais.microdados\_estabelecimentos`   
 WHERE ano = 2019",   
 page\_size = 300000  
)  
  
temvinculo <- vinculos %>%   
 filter(qtde\_vinculos\_ativos != 0) %>%  
 group\_by(id\_municipio) %>%  
 summarise(n\_empresas = n()) %>%  
 right\_join(top100\_mun\_cod, by = "id\_municipio") %>%  
 select(id\_municipio, nome, sigla\_uf,n\_empresas) %>%  
 arrange(desc(n\_empresas))  
  
# ------------------------------------------------------------------  
# PASSO 4: calcular indicador i513 e salvar arquivo  
# ------------------------------------------------------------------  
  
i513 <- temvinculo %>%   
 inner\_join(finep\_ice, by = c('id\_municipio','nome','sigla\_uf')) %>%  
 inner\_join(bndes\_ice, by = c('id\_municipio','nome','sigla\_uf'),   
 suffix = c('\_finep', '\_bndes')) %>%  
 mutate(i513 = (contratado\_finep + contratado\_bndes)/n\_empresas) %>%  
 mutate(i513\_pad = (i513 - mean(i513))/sdp(i513)) %>%  
 select(id\_municipio, nome, sigla\_uf, i513, i513\_pad) %>%  
 arrange(desc(i513\_pad))  
  
# salvamos o arquivo com o indicador  
write.csv(i513, "i513.csv", row.names = FALSE)

## Script i514

Este script em R refere-se ao [Indicador Infraestrutura Tecnológica (i514)](#i514)

library(readxl)  
library(tidyverse)  
  
# Indicador Infraestrutura Tecnologica (i514)  
  
# ------------------------------------------------------------------  
# PASSO 0: preliminares  
# ------------------------------------------------------------------  
  
# definimos uma funcao para desvio padrao populacional  
sdp <- function(x) {  
 sd(x)\*(sqrt((length(x)-1)/length(x)))  
}  
  
# importamos tabela com dados dos municipios mais populosos  
top100\_mun\_cod <- read.csv("top100\_mun\_cod.csv", stringsAsFactors = FALSE)  
top100\_mun\_cod <- top100\_mun\_cod %>%   
 transform(id\_municipio = as.character(id\_municipio))  
  
# ------------------------------------------------------------------  
# PASSO 1 e 2: obter planilha com dados do indicador e padronizar  
# ------------------------------------------------------------------  
  
i514 <- read\_excel("C:/Users/User/OneDrive/Documentos/ICE 2021/d5 Inovação/  
 sd51 Inputs/i514/Indicador Infraestrutura Tecnológica.xlsx",   
 sheet = 2) %>%  
 inner\_join(top100\_mun\_cod, by = c('Cidade' = 'nome')) %>%  
 rename(i514 = Indicador, nome = Cidade) %>%  
 mutate(i514\_pad = (i514 - mean(i514))/sdp(i514)) %>%  
 select(id\_municipio, nome, sigla\_uf, i514, i514\_pad)   
   
# salvamos o arquivo com o indicador  
write.csv(i514, "i514.csv", row.names = FALSE)

## Script i515

Este script em R refere-se ao [Indicador Contratos de Concessão (i515)](#i515)

library(readxl)  
library(tidyverse)  
  
# Indicador Contratos de Concessao (i515)  
  
# ------------------------------------------------------------------  
# PASSO 0: preliminares  
# ------------------------------------------------------------------  
  
# definimos uma funcao para desvio padrao populacional  
sdp <- function(x) {  
 sd(x)\*(sqrt((length(x)-1)/length(x)))  
}  
  
# importamos tabela com dados dos municipios mais populosos  
top100\_mun\_cod <- read.csv("top100\_mun\_cod.csv", stringsAsFactors = FALSE)  
top100\_mun\_cod <- top100\_mun\_cod %>%   
 transform(id\_municipio = as.numeric(id\_municipio))  
  
# ------------------------------------------------------------------  
# PASSO 1: importar indicador do ano passado e padronizar  
# ------------------------------------------------------------------  
  
i515 <- read\_excel("C:/Users/User/OneDrive/Documentos/ICE 2021/d5 Inovação/  
 sd51 Inputs/i515/Indicador Contratos de Concessão.xlsx",   
 sheet = "Indicador") %>%  
 rename(nome = Cidade,  
 id\_municipio = Código,  
 i515 = Indicador) %>%  
 mutate(i515\_pad = (i515 - (mean(i515)))/sdp(i515)) %>%  
 inner\_join(top100\_mun\_cod, by = c('id\_municipio', 'nome')) %>%  
 select(id\_municipio, nome, sigla\_uf, i515, i515\_pad) %>%  
 arrange(desc(i515\_pad))  
  
# ------------------------------------------------------------------  
# PASSO 2: exportar indicador   
# ------------------------------------------------------------------  
  
# salvamos o arquivo com o indicador  
write.csv(i515, "i515.csv", row.names = FALSE)

## Script i521

Este script em R refere-se ao [Indicador Patentes (i521)](#i521)

library(readxl)  
library(tidyverse)  
library("basedosdados")  
  
# Indicador Patentes (i521)  
  
# ------------------------------------------------------------------  
# PASSO 0: preliminares  
# ------------------------------------------------------------------  
  
# Definir projeto no Google Cloud  
set\_billing\_id("workshop-teste-322616")  
  
# definimos uma funcao para desvio padrao populacional   
sdp <- function(x) {  
 sd(x)\*(sqrt((length(x)-1)/length(x)))  
}  
  
# importamos tabela com dados dos municipios mais populosos  
top100\_mun\_cod <- read.csv("top100\_mun\_cod.csv", stringsAsFactors = FALSE)  
top100\_mun\_cod <- top100\_mun\_cod %>%   
 transform(id\_municipio = as.numeric(id\_municipio))  
  
# ------------------------------------------------------------------  
# PASSO 1: obter dados de depositos de patentes  
# ------------------------------------------------------------------  
  
# importar planilhas com dados de depositos de patentes  
# nesta planilha, foram feitas pequenas alteracoes em celulas mescladas  
patentes\_pi <- read\_excel('C:/Users/User/OneDrive/Documentos/ICE 2021/  
 d5 Inovação/sd52 Outputs/i521/  
 5a - Depósitos de Patentes do Tipo PI por Cidade.xls',   
 skip = 7) %>%   
 na.omit() %>%   
 rename(ano2018 = '2018', ano2019 = '2019', id\_municipio = 'Cod IBGE',   
 nome = 'Cidade') %>%  
 mutate(pat\_pi = ano2018 + ano2019) %>%  
 select(id\_municipio, nome, pat\_pi)  
  
patentes\_mu <- read\_excel('C:/Users/User/OneDrive/Documentos/ICE 2021/  
 d5 Inovação/sd52 Outputs/i521/  
 5b - Depósitos de Patentes do Tipo MU por Cidade.xls',   
 skip = 7) %>%   
 na.omit() %>%   
 rename(ano2018 = '2018', ano2019 = '2019', id\_municipio = 'Cod IBGE',   
 nome = 'Cidade') %>%  
 mutate(pat\_mu = ano2018 + ano2019) %>%  
 select(id\_municipio, nome, pat\_mu)  
  
patentes\_ca <- read\_excel('C:/Users/User/OneDrive/Documentos/ICE 2021/  
 d5 Inovação/sd52 Outputs/i521/  
 5c - Depósitos de Patentes do Tipo CA por Cidade.xls',   
 skip = 7) %>%   
 na.omit() %>%   
 rename(ano2018 = '2018', ano2019 = '2019', id\_municipio = 'Cod IBGE',   
 nome = 'Cidade') %>%  
 mutate(pat\_ca = ano2018 + ano2019) %>%  
 select(id\_municipio, nome, pat\_ca) %>%  
 transform(id\_municipio = as.numeric(id\_municipio))  
  
# juntar as tres tabelas  
patentes <- patentes\_pi %>%  
 full\_join(patentes\_mu, by = c('id\_municipio', 'nome')) %>%  
 full\_join(patentes\_ca, by = c('id\_municipio', 'nome')) %>%  
 replace\_na(list(pat\_pi = 0,  
 pat\_mu = 0,  
 pat\_ca = 0)) %>%  
 mutate(pat = pat\_pi + pat\_mu + pat\_ca) %>%  
 select(id\_municipio, nome, pat)  
  
# ------------------------------------------------------------------  
# PASSO 2: obter numero de empresas com pelo menos 1 funcionario  
# ------------------------------------------------------------------  
  
# acessar RAIS estabelecimento pelo Base dos Dados  
# obter numero de empresas com pelo menos um funcionario em cada municipio  
vinculos <- basedosdados::read\_sql(  
 "SELECT ano, id\_municipio, qtde\_vinculos\_ativos,tamanho\_estabelecimento   
 FROM `basedosdados.br\_me\_rais.microdados\_estabelecimentos`   
 WHERE ano = 2019",   
 page\_size = 300000)  
  
top100\_mun\_cod$id\_municipio = as.character(top100\_mun\_cod$id\_municipio)  
  
temvinculo <- vinculos %>%   
 filter(qtde\_vinculos\_ativos != 0) %>%  
 group\_by(id\_municipio) %>%  
 summarise(n\_empresas = n()) %>%  
 right\_join(top100\_mun\_cod, by = "id\_municipio") %>%  
 select(id\_municipio, nome, sigla\_uf,n\_empresas) %>%  
 arrange(desc(n\_empresas))  
  
# ------------------------------------------------------------------  
# PASSO 3: juntar tabelas e calcular indicador  
# ------------------------------------------------------------------  
  
i521 <- temvinculo %>%   
 transform(id\_municipio = as.numeric(id\_municipio)) %>%  
 left\_join(patentes, by = c('id\_municipio', 'nome')) %>%  
 mutate(i521 = pat/n\_empresas,  
 i521\_pad = (i521 - mean(i521))/sdp(i521)) %>%  
 select(id\_municipio, nome, sigla\_uf, i521, i521\_pad) %>%  
 arrange(desc(i521\_pad))  
  
# exportar indicador  
write.csv(i521, "i521.csv", row.names = FALSE)

## Script i522

Este script em R refere-se ao [Indicador Tamanho da Indústria Inovadora (i522)](#i522)

library(readxl)  
library(tidyverse)  
library("basedosdados")  
  
# Indicador Tamanho da Industria Inovadora (i522)  
  
# ------------------------------------------------------------------  
# PASSO 0: preliminares  
# ------------------------------------------------------------------  
  
# Definir projeto no Google Cloud  
set\_billing\_id("workshop-teste-322616")  
  
# definimos uma funcao para desvio padrao populacional  
sdp <- function(x) {  
 sd(x)\*(sqrt((length(x)-1)/length(x)))  
}  
  
# importamos tabela com dados dos municipios mais populosos  
top100\_mun\_cod <- read.csv("top100\_mun\_cod.csv", stringsAsFactors = FALSE)  
top100\_mun\_cod <- top100\_mun\_cod %>%   
 transform(id\_municipio = as.character(id\_municipio))  
  
# ------------------------------------------------------------------  
# PASSO 1: obter numero de empresas de industria inovadora  
# ------------------------------------------------------------------  
  
# importar planilha com classificacao cnae de empresas  
cnae <- read\_excel('C:/Users/User/OneDrive/Documentos/ICE 2021/d5 Inovação/  
 sd52 Outputs/i522/CNAE20\_EstruturaDetalhada.xls',   
 skip = 2) %>%  
 select(classe = Classe,   
 denom = Denominação)  
  
# remover "-" e "." das classes CNAE  
cnae$classe = gsub('[[:punct:]]', '', cnae$classe)  
  
# importar planilha do ICE 2020 com classes de industria inovadora  
cnae\_ice <- read\_excel('C:/Users/User/OneDrive/Documentos/ICE 2021/  
 d5 Inovação/sd52 Outputs/i522/  
 Indicador Tamanho da Indústria Inovadora.xlsx',   
 sheet = 2) %>%  
 select(denom = "CNAE 2.0 Classe") %>%  
 inner\_join(cnae, by = 'denom') %>%  
 na.omit()  
  
# obter numero de empresas de industria inovadora a partir da RAIS estabelecimentos  
query <- "SELECT ano, id\_municipio, cnae\_2   
FROM `basedosdados.br\_me\_rais.microdados\_estabelecimentos`   
WHERE ano = 2019  
AND id\_municipio IN ('3550308','3304557','5300108','2927408','2304400','3106200','1302603','4106902','2611606','5208707','1501402','4314902','3518800'  
,'3509502','2111300','3304904','2704302','3301702','5002704','2408102','2211001','3548708','3303500','2507507','3549904','3547809'  
,'3543402','2607901','3534401','3170206','3552205','3118601','2800308','2910800','5103403','4209102','5201405','4113700','3136702'  
,'1100205','1500800','3205002','4305108','3303302','3300456','1600303','3301009','4205407','3205200','3529401','3305109','3549805'  
,'3530607','3106705','3548500','4115200','3513801','3525904','1400100','3143302','1200401','2504009','3538709','3510609','2609600'  
,'5201108','3201308','3506003','3523107','3551009','3205309','2604106','2303709','4202404','3516200','4119905','2611101','4304606'  
,'4314407','2933307','3154606','3170107','2610707','4104808','3541000','4125506','3518701','3554102','3526902','3303906','1506807'  
,'1721000','2905701','2408003','3552502','3552809','5108402','3552403','4316907','4309209','1504208')"  
  
estabelec <- read\_sql(query)  
  
estabelec\_ice <- estabelec %>%  
 group\_by(id\_municipio) %>%  
 count(cnae\_2, name = 'n\_empresas') %>%  
 inner\_join(top100\_mun\_cod, by = 'id\_municipio') %>%  
 inner\_join(cnae\_ice, by = c('cnae\_2' = 'classe')) %>%  
 group\_by(id\_municipio, nome, sigla\_uf) %>%  
 summarise(empresas\_inovadoras = sum(n\_empresas)) %>%  
 arrange(desc(empresas\_inovadoras))  
  
# ------------------------------------------------------------------  
# PASSO 2: obter numero de empresas com mais de um funcionario  
# ------------------------------------------------------------------  
  
# acessar RAIS estabelecimento pelo Base dos Dados  
# obter numero de empresas com pelo menos um funcionario em cada municipio  
vinculos <- basedosdados::read\_sql(  
 "SELECT ano, id\_municipio, qtde\_vinculos\_ativos,tamanho\_estabelecimento   
 FROM `basedosdados.br\_me\_rais.microdados\_estabelecimentos`   
 WHERE ano = 2019",   
 page\_size = 300000)  
  
temvinculo <- vinculos %>%   
 filter(qtde\_vinculos\_ativos != 0) %>%  
 group\_by(id\_municipio) %>%  
 summarise(n\_empresas = n()) %>%  
 right\_join(top100\_mun\_cod, by = "id\_municipio") %>%  
 select(id\_municipio, nome, sigla\_uf,n\_empresas) %>%  
 arrange(desc(n\_empresas))  
  
# ------------------------------------------------------------------  
# PASSO 3: calcular indicador  
# ------------------------------------------------------------------  
  
i522 <- temvinculo %>%   
 inner\_join(estabelec\_ice, by = c('id\_municipio', 'nome', 'sigla\_uf')) %>%  
 mutate(i522 = empresas\_inovadoras/n\_empresas,  
 i522\_pad = (i522 - mean(i522))/sdp(i522)) %>%  
 select(id\_municipio, nome, sigla\_uf, i522, i522\_pad) %>%  
 arrange(desc(i522\_pad))  
  
# exportar indicador  
write.csv(i522, "i522.csv", row.names = FALSE)

## Script i523

Este script em R refere-se ao [Indicador Tamanho da Economia Criativa (i523)](#i523)

library(readxl)  
library(tidyverse)  
library("basedosdados")  
  
# Indicador Tamanho da Economia Criativa (i523)  
  
# ------------------------------------------------------------------  
# PASSO 0: preliminares  
# ------------------------------------------------------------------  
  
# Definir projeto no Google Cloud  
set\_billing\_id("workshop-teste-322616")  
  
# definimos uma funcao para desvio padrao populacional   
sdp <- function(x) {  
 sd(x)\*(sqrt((length(x)-1)/length(x)))  
}  
  
# importamos tabela com dados dos municipios mais populosos  
top100\_mun\_cod <- read.csv("top100\_mun\_cod.csv", stringsAsFactors = FALSE)  
top100\_mun\_cod <- top100\_mun\_cod %>%   
 transform(id\_municipio = as.character(id\_municipio))  
  
# ------------------------------------------------------------------  
# PASSO 1: obter numero de empresas de economia criativa  
# ------------------------------------------------------------------  
  
# importar planilha com classificacao cnae de empresas  
cnae <- read\_excel('C:/Users/User/OneDrive/Documentos/ICE 2021/d5 Inovação/  
 sd52 Outputs/i522/CNAE20\_EstruturaDetalhada.xls',   
 skip = 2) %>%  
 select(classe = Classe,   
 denom = Denominação) %>%  
 na.omit()  
  
# remover "-" e "." das classes CNAE  
cnae$classe = gsub('[[:punct:]]', '', cnae$classe)  
  
# importar planilha do ICE 2020 com classes de industria inovadora  
cnae\_ice\_2 <- read\_excel('C:/Users/User/OneDrive/Documentos/ICE 2021/  
 d5 Inovação/sd52 Outputs/i523/  
 Indicador Tamanho da Economia Criativa.xlsx',   
 sheet = 2) %>%  
 select(denom = "CNAE 2.0 Classe") %>%  
 left\_join(cnae, by = 'denom') %>%  
 na.omit()  
  
# obter numero de empresas de industria inovadora a partir da RAIS estabelecimentos  
query <- "SELECT ano, id\_municipio, cnae\_2   
FROM `basedosdados.br\_me\_rais.microdados\_estabelecimentos`   
WHERE ano = 2019  
AND id\_municipio IN ('3550308','3304557','5300108','2927408','2304400','3106200','1302603','4106902','2611606','5208707','1501402','4314902','3518800'  
,'3509502','2111300','3304904','2704302','3301702','5002704','2408102','2211001','3548708','3303500','2507507','3549904','3547809'  
,'3543402','2607901','3534401','3170206','3552205','3118601','2800308','2910800','5103403','4209102','5201405','4113700','3136702'  
,'1100205','1500800','3205002','4305108','3303302','3300456','1600303','3301009','4205407','3205200','3529401','3305109','3549805'  
,'3530607','3106705','3548500','4115200','3513801','3525904','1400100','3143302','1200401','2504009','3538709','3510609','2609600'  
,'5201108','3201308','3506003','3523107','3551009','3205309','2604106','2303709','4202404','3516200','4119905','2611101','4304606'  
,'4314407','2933307','3154606','3170107','2610707','4104808','3541000','4125506','3518701','3554102','3526902','3303906','1506807'  
,'1721000','2905701','2408003','3552502','3552809','5108402','3552403','4316907','4309209','1504208')"  
  
estabelec <- read\_sql(query)  
  
estabelec\_ice\_2 <- estabelec %>%  
 group\_by(id\_municipio) %>%  
 count(cnae\_2, name = 'n\_empresas') %>%  
 inner\_join(top100\_mun\_cod, by = 'id\_municipio') %>%  
 inner\_join(cnae\_ice\_2, by = c('cnae\_2' = 'classe')) %>%  
 group\_by(id\_municipio, nome, sigla\_uf) %>%  
 summarise(empresas\_criativas = sum(n\_empresas)) %>%  
 arrange(desc(empresas\_criativas))  
  
# ------------------------------------------------------------------  
# PASSO 2: obter numero de empresas com mais de um funcionario  
# ------------------------------------------------------------------  
  
# acessar RAIS estabelecimento pelo Base dos Dados  
# obter numero de empresas com pelo menos um funcionario em cada municipio  
vinculos <- basedosdados::read\_sql(  
 "SELECT ano, id\_municipio, qtde\_vinculos\_ativos,tamanho\_estabelecimento   
 FROM `basedosdados.br\_me\_rais.microdados\_estabelecimentos`   
 WHERE ano = 2019",   
 page\_size = 300000)  
  
temvinculo <- vinculos %>%   
 filter(qtde\_vinculos\_ativos != 0) %>%  
 group\_by(id\_municipio) %>%  
 summarise(n\_empresas = n()) %>%  
 right\_join(top100\_mun\_cod, by = "id\_municipio") %>%  
 select(id\_municipio, nome, sigla\_uf,n\_empresas) %>%  
 arrange(desc(n\_empresas))  
  
# ------------------------------------------------------------------  
# PASSO 3: calcular indicador  
# ------------------------------------------------------------------  
  
i523 <- temvinculo %>%   
 inner\_join(estabelec\_ice\_2, by = c('id\_municipio', 'nome', 'sigla\_uf')) %>%  
 mutate(i523 = empresas\_criativas/n\_empresas,  
 i523\_pad = (i523 - mean(i523))/sdp(i523)) %>%  
 select(id\_municipio, nome, sigla\_uf, i523, i523\_pad) %>%  
 arrange(desc(i523\_pad))  
  
# exportar indicador  
write.csv(i523, "i523.csv", row.names = FALSE)

## Script i524

Este script em R refere-se ao [Indicador Tamanho das Empresas TIC (i514)](#i524)

library(readxl)  
library(tidyverse)  
library("basedosdados")  
  
# Indicador Tamanho das Empresas TIC (i524)  
  
# ------------------------------------------------------------------  
# PASSO 0: preliminares  
# ------------------------------------------------------------------  
  
# Definir projeto no Google Cloud  
set\_billing\_id("workshop-teste-322616")  
  
# definimos uma funÃ§Ã£o para desvio padrao populacional   
sdp <- function(x) {  
 sd(x)\*(sqrt((length(x)-1)/length(x)))  
}  
  
# importamos tabela com dados dos municipios mais populosos  
top100\_mun\_cod <- read.csv("top100\_mun\_cod.csv", stringsAsFactors = FALSE)  
top100\_mun\_cod <- top100\_mun\_cod %>%   
 transform(id\_municipio = as.character(id\_municipio))  
  
# ------------------------------------------------------------------  
# PASSO 1: obter numero de empresas TIC  
# ------------------------------------------------------------------  
  
# importar planilha com classificacao cnae de empresas  
cnae <- read\_excel('C:/Users/User/OneDrive/Documentos/ICE 2021/d5 Inovação/  
 sd52 Outputs/i522/CNAE20\_EstruturaDetalhada.xls',   
 skip = 2) %>%  
 select(classe = Classe,   
 denom = Denominação) %>%  
 na.omit()  
  
# remover "-" e "." das classes CNAE  
cnae$classe = gsub('[[:punct:]]', '', cnae$classe)  
  
# importar planilha do ICE 2020 com classes de industria inovadora  
cnae\_ice\_3 <- read\_excel('C:/Users/User/OneDrive/Documentos/ICE 2021/  
 d5 Inovação/sd52 Outputs/i524/  
 Indicador Tamanho das Empresas TIC.xlsx',   
 sheet = 2) %>%  
 select(denom = "CNAE 2.0 Classe") %>%  
 left\_join(cnae, by = 'denom') %>%  
 na.omit()  
  
# obter numero de empresas de industria inovadora a partir da RAIS estabelecimentos  
query <- "SELECT ano, id\_municipio, cnae\_2   
FROM `basedosdados.br\_me\_rais.microdados\_estabelecimentos`   
WHERE ano = 2019  
AND id\_municipio IN ('3550308','3304557','5300108','2927408','2304400','3106200','1302603','4106902','2611606','5208707','1501402','4314902','3518800'  
,'3509502','2111300','3304904','2704302','3301702','5002704','2408102','2211001','3548708','3303500','2507507','3549904','3547809'  
,'3543402','2607901','3534401','3170206','3552205','3118601','2800308','2910800','5103403','4209102','5201405','4113700','3136702'  
,'1100205','1500800','3205002','4305108','3303302','3300456','1600303','3301009','4205407','3205200','3529401','3305109','3549805'  
,'3530607','3106705','3548500','4115200','3513801','3525904','1400100','3143302','1200401','2504009','3538709','3510609','2609600'  
,'5201108','3201308','3506003','3523107','3551009','3205309','2604106','2303709','4202404','3516200','4119905','2611101','4304606'  
,'4314407','2933307','3154606','3170107','2610707','4104808','3541000','4125506','3518701','3554102','3526902','3303906','1506807'  
,'1721000','2905701','2408003','3552502','3552809','5108402','3552403','4316907','4309209','1504208')"  
  
estabelec <- read\_sql(query)  
  
estabelec\_ice\_3 <- estabelec %>%  
 group\_by(id\_municipio) %>%  
 count(cnae\_2, name = 'n\_empresas') %>%  
 inner\_join(top100\_mun\_cod, by = 'id\_municipio') %>%  
 inner\_join(cnae\_ice\_3, by = c('cnae\_2' = 'classe')) %>%  
 group\_by(id\_municipio, nome, sigla\_uf) %>%  
 summarise(empresas\_tic = sum(n\_empresas)) %>%  
 arrange(desc(empresas\_tic))  
  
# ------------------------------------------------------------------  
# PASSO 2: obter numero de empresas com mais de um funcionario  
# ------------------------------------------------------------------  
  
# acessar RAIS estabelecimento pelo Base dos Dados  
#obter numero de empresas com pelo menos um funcionario em cada municipio  
vinculos <- basedosdados::read\_sql(  
 "SELECT ano, id\_municipio, qtde\_vinculos\_ativos,tamanho\_estabelecimento   
 FROM `basedosdados.br\_me\_rais.microdados\_estabelecimentos`   
 WHERE ano = 2019",   
 page\_size = 300000)  
  
temvinculo <- vinculos %>%   
 filter(qtde\_vinculos\_ativos != 0) %>%  
 group\_by(id\_municipio) %>%  
 summarise(n\_empresas = n()) %>%  
 right\_join(top100\_mun\_cod, by = "id\_municipio") %>%  
 select(id\_municipio, nome, sigla\_uf,n\_empresas) %>%  
 arrange(desc(n\_empresas))  
  
# ------------------------------------------------------------------  
# PASSO 3: calcular indicador  
# ------------------------------------------------------------------  
  
i524 <- temvinculo %>%   
 inner\_join(estabelec\_ice\_3, by = c('id\_municipio', 'nome', 'sigla\_uf')) %>%  
 mutate(i524 = empresas\_tic/n\_empresas,  
 i524\_pad = (i524 - mean(i524))/sdp(i524)) %>%  
 select(id\_municipio, nome, sigla\_uf, i524, i524\_pad) %>%  
 arrange(desc(i524\_pad))  
  
# exportar indicador  
write.csv(i524, "i524.csv", row.names = FALSE)

## Script i611

Este script em R refere-se ao [Indicador Nota do Ideb (i611)](#i611)

library(tidyverse)  
library("basedosdados")  
library(readxl)  
  
# Indicador Nota do Ideb (i611)  
  
# ------------------------------------------------------------------  
# PASSO 0: preliminares  
# ------------------------------------------------------------------  
  
#definimos uma funcao para desvio padrao populacional  
sdp <- function(x) {  
 sd(x)\*(sqrt((length(x)-1)/length(x)))  
}  
  
# importamos tabela com dados dos municipios mais populosos  
top100\_mun\_cod <- read.csv("top100\_mun\_cod.csv", stringsAsFactors = FALSE)  
top100\_mun\_cod <- top100\_mun\_cod %>%   
 transform(id\_municipio = as.character(id\_municipio))  
  
top100\_mun\_cod$NOME <- toupper(top100\_mun\_cod$nome)  
  
# ------------------------------------------------------------------  
# PASSO 1: obter notas ideb 2019  
# ------------------------------------------------------------------  
  
#importamos a base de dados necessaria para calcular o indicador  
ideb <- read\_excel("C:/Users/User/OneDrive/Documentos/ICE 2021/  
 d6 Capital Humano/  
 sd61 Acesso e Qualidade da Mão de Obra Básica/  
 i611 base de dados.xlsx", range = "A2:J5570")  
ideb\_2019 <- ideb %>% select((1:2), '2019') %>%   
 rename(sigla\_uf = UF,   
 Ideb\_2019 = "2019",  
 NOME = 'Município')  
  
# juntar com tabela dos municipios mais populosos  
ideb\_100mun <- top100\_mun\_cod %>%   
 left\_join(ideb\_2019, by = c("NOME", "sigla\_uf")) %>%   
 select(-NOME)  
  
# ------------------------------------------------------------------  
# PASSO 2: calcular indicador e padroniza-lo  
# ------------------------------------------------------------------  
  
i611 <- ideb\_100mun %>%   
 rename(i611 = Ideb\_2019) %>%   
 transform(i611 = as.numeric(i611),   
 id\_municipio = as.character(id\_municipio)) %>%   
 mutate(i611\_pad = (i611 - mean(i611))/sdp(i611))  
  
# exportar tabela com indicador  
write.csv(i611, "i611.csv", row.names = FALSE)

## Script i612

Este script em R refere-se ao [Indicador Proporção de adultos com pelo menos o Ensino Médio completo (i612)](#i612)

library(tidyverse)  
  
# Indicador Proporcao de adultos com pelo menos o Ensino Medio completo (i612)  
  
# ------------------------------------------------------------------  
# PASSO 0: preliminares  
# ------------------------------------------------------------------  
  
# definimos uma funcao para desvio padrao populacional  
sdp <- function(x) {  
 sd(x)\*(sqrt((length(x)-1)/length(x)))  
}  
  
# importamos tabela com os municipios mais populosos  
top100\_mun\_cod <- read.csv("top100\_mun\_cod.csv", stringsAsFactors = FALSE)  
top100\_mun\_cod <- top100\_mun\_cod %>%   
 transform(id\_municipio = as.integer(id\_municipio))  
  
# ------------------------------------------------------------------  
# PASSO 1: obter dados do ENEM  
# ------------------------------------------------------------------  
  
#importamos a base de dados do ENEM  
ENEM\_2019 <- data.table::fread(input='C:/Users/User/OneDrive/Documentos/  
 ICE 2021/d6 Capital Humano/  
 sd61 Acesso e Qualidade da Mão de Obra Básica/  
 Downloads/DADOS/MICRODADOS\_ENEM\_2019.csv',  
 integer64='character',  
 skip=0, #Ler do inicio  
 nrow=-1, #Ler todos os registros  
 na.strings = "",   
 showProgress = TRUE)  
  
# selecionamos variaveis de interesse  
ENEM\_2019 <- ENEM\_2019 %>%   
 select(NU\_INSCRICAO,   
 NU\_ANO,   
 CO\_MUNICIPIO\_ESC,   
 CO\_MUNICIPIO\_RESIDENCIA,   
 CO\_ESCOLA,   
 TP\_ST\_CONCLUSAO,   
 TP\_ENSINO,   
 NU\_NOTA\_CH,   
 NU\_NOTA\_CN,   
 NU\_NOTA\_LC,   
 NU\_NOTA\_MT,   
 NU\_NOTA\_REDACAO,   
 Q001,   
 Q002)  
  
# unimos as duas tabelas para selecionar apenas os inscritos dos municipios ICE  
ENEM\_100mun <- top100\_mun\_cod %>%   
 left\_join(ENEM\_2019, by = c("id\_municipio" = "CO\_MUNICIPIO\_ESC"))  
  
# salvamos essa tabela para usar em outros indices  
write.csv(ENEM\_100mun, "ENEM\_100mun.csv", row.names = FALSE)  
  
# contamos a quantidade de inscritos por municipio  
inscritos\_mun <- ENEM\_100mun %>%   
 count(id\_municipio, nome, sigla\_uf, sort = TRUE) %>%   
 rename(inscritos = n)  
  
# criamos um vetor com as respostas do questionario que selecionam pai ou mae   
# com EM completo (ver dicionario dos microdados ENEM 2019)  
alvo\_EM <- c("E", "F", "G")  
  
# contamos a quantidade de inscritos que declaram ter pai com EM completo  
pai\_EM <- ENEM\_100mun %>%   
 filter(Q001 %in% alvo\_EM) %>%   
 count(id\_municipio, nome, sigla\_uf, sort = TRUE) %>%   
 rename(pai\_EM = n)  
  
# contamos a quantidade de inscritos que declaram ter mae com EM completo  
mae\_EM <- ENEM\_100mun %>%   
 filter(Q002 %in% alvo\_EM) %>%   
 count(id\_municipio, nome, sigla\_uf, sort = TRUE) %>%   
 rename(mae\_EM = n)  
  
# juntamos as tres tabelas:  
# inscritos totais, inscritos com pais com EM, incritos com mae com EM  
paimae\_EM <- pai\_EM %>%   
 inner\_join(mae\_EM, by = c('id\_municipio', 'nome', 'sigla\_uf')) %>%   
 inner\_join(inscritos\_mun, by = c('id\_municipio', 'nome', 'sigla\_uf')) %>%  
 mutate(prop\_paiEM = pai\_EM/inscritos,  
 prop\_maeEM = mae\_EM/inscritos) %>%  
 mutate(i612 = (prop\_paiEM + prop\_maeEM)/2)  
  
# ------------------------------------------------------------------  
# PASSO 2: calcular o indicador  
# ------------------------------------------------------------------  
  
# calculamos e padronizamos o indicador  
i612 <- paimae\_EM %>%   
 select(id\_municipio, nome, sigla\_uf, i612) %>%   
 transform(i612 = as.numeric(i612),   
 id\_municipio = as.character(id\_municipio)) %>%   
 mutate(i612\_pad = (i612 - mean(i612))/sdp(i612))   
  
# exportamos o indicador  
write.csv(i612, "i612.csv", row.names = FALSE)

## Script i613

Este script em R refere-se ao [Indicador Taxa Líquida de Matrícula no Ensino Médio (i613)](#i613)

library(tidyverse)  
library("basedosdados")  
library(readxl)  
  
# Indicador Taxa Liquida de Matricula no Ensino Medio (i613)  
  
# ------------------------------------------------------------------  
# PASSO 0: preliminares  
# ------------------------------------------------------------------  
  
# Definir projeto no Google Cloud  
set\_billing\_id("workshop-teste-322616")  
  
# definimos uma funcao para desvio padrao populacional  
sdp <- function(x) {  
 sd(x)\*(sqrt((length(x)-1)/length(x)))  
}  
  
# importamos tabela com os municipios mais populosos  
top100\_mun\_cod <- read.csv("top100\_mun\_cod.csv", stringsAsFactors = FALSE)  
top100\_mun\_cod <- top100\_mun\_cod %>%   
 transform(id\_municipio = as.character(id\_municipio))  
  
# ------------------------------------------------------------------  
# PASSO 1: obter numero de alunos entre 15 e 17 no EM  
# ------------------------------------------------------------------  
  
# importamos os dados do censo escolar de 2020 com alunos   
# de 15 a 17 anos matriculados no EM  
query <- "SELECT ano, id\_municipio, sigla\_uf, idade\_referencia,   
etapa\_ensino FROM `basedosdados.br\_inep\_censo\_escolar.matricula`   
WHERE ano = 2020  
AND (idade\_referencia >= 15 AND idade\_referencia <= 17)   
AND (etapa\_ensino IN ('25','26','27','28','29','30','31','32','33','34','35','36','37','38'))  
AND id\_municipio IN ('3550308','3304557','5300108','2927408','2304400','3106200','1302603','4106902','2611606','5208707','1501402','4314902','3518800'  
,'3509502','2111300','3304904','2704302','3301702','5002704','2408102','2211001','3548708','3303500','2507507','3549904','3547809'  
,'3543402','2607901','3534401','3170206','3552205','3118601','2800308','2910800','5103403','4209102','5201405','4113700','3136702'  
,'1100205','1500800','3205002','4305108','3303302','3300456','1600303','3301009','4205407','3205200','3529401','3305109','3549805'  
,'3530607','3106705','3548500','4115200','3513801','3525904','1400100','3143302','1200401','2504009','3538709','3510609','2609600'  
,'5201108','3201308','3506003','3523107','3551009','3205309','2604106','2303709','4202404','3516200','4119905','2611101','4304606'  
,'4314407','2933307','3154606','3170107','2610707','4104808','3541000','4125506','3518701','3554102','3526902','3303906','1506807'  
,'1721000','2905701','2408003','3552502','3552809','5108402','3552403','4316907','4309209','1504208')"  
censoescolar <- read\_sql(query, page\_size = 300000)  
  
# contamos a quantidade de alunos de 15 a 17 anos   
# matriculados no EM por municipio  
matriculados15a17\_EM <- censoescolar %>%   
 count(id\_municipio, sigla\_uf, sort = TRUE) %>%   
 rename(matriculados15a17\_EM = n)  
  
# ------------------------------------------------------------------  
# PASSO 2: obter populacao entre entre 15 e 17  
# ------------------------------------------------------------------  
  
# importamos tabela do censo 2010 com dados de populacao por municipio e idade  
censo2010 <- read\_excel("C:/Users/User/OneDrive/Documentos/ICE 2021/  
 d6 Capital Humano/  
 sd61 Acesso e Qualidade da Mão de Obra Básica/  
 Downloads/censo2010.xlsx", skip = 6)  
pop\_15\_17 <- top100\_mun\_cod %>%   
 left\_join(censo2010, by = c("id\_municipio" = "Cód.")) %>%   
 select(id\_municipio, nome, sigla\_uf,   
 "15 anos", "16 anos", "17 anos") %>%  
 rename(anos\_15 = "15 anos",   
 anos\_16 = "16 anos",   
 anos\_17 = "17 anos") %>%  
 mutate(pop\_15a17 = anos\_15 + anos\_16 + anos\_17) %>%   
 select(-anos\_15, -anos\_16, -anos\_17)  
  
# ------------------------------------------------------------------  
# PASSOS 3 e 4: obter populacao em 2010 e 2021  
# ------------------------------------------------------------------  
  
# importar tabela com populacao total dos municipios em 2010  
query <- "SELECT \* FROM `basedosdados.br\_ibge\_populacao.municipio`   
WHERE ano = 2010"  
pop\_2010 <- read\_sql(query)  
  
# obter populacao de 2021  
pop\_2021 <- read\_excel('C:/Users/User/OneDrive/Documentos/ICE 2021/  
 estimativa\_dou\_2021.xls', sheet = 2, skip = 1) %>%  
 na.omit() %>%  
 select(nome = 'NOME DO MUNICÍPIO',  
 sigla\_uf = 'UF',  
 populacao = 'POPULAÇÃO ESTIMADA') %>%  
 inner\_join(top100\_mun\_cod, by = c('nome', 'sigla\_uf'))  
  
# ------------------------------------------------------------------  
# PASSOS 5: obter fator de crescimento populacional  
# ------------------------------------------------------------------  
  
# calcular fator de crescimento populacional  
cresc\_pop <- pop\_2010 %>% inner\_join(pop\_2021, by = "id\_municipio",   
 suffix = c("\_2010", "\_2021")) %>%   
 mutate(cresc\_pop = populacao\_2021/populacao\_2010) %>%  
 select (-ano, -populacao\_2010, -populacao\_2021)  
  
# ------------------------------------------------------------------  
# PASSOS 6: obter populacao estimada entre 15 e 17 em 2021  
# ------------------------------------------------------------------  
  
# Calcular populacao estimada em cada municipio entre 15 e 17 anos  
pop\_15\_17\_atual <- pop\_15\_17 %>%   
 inner\_join(cresc\_pop, by = c("id\_municipio", "nome", "sigla\_uf")) %>%  
 mutate(pop\_15a17\_atual = pop\_15a17 \* cresc\_pop) %>%  
 select(-pop\_15a17, -cresc\_pop)  
# ------------------------------------------------------------------  
# PASSOS 7: calcular indicador  
# ------------------------------------------------------------------  
  
# calculamos indicador   
i613 <- pop\_15\_17\_atual %>% inner\_join(matriculados15a17\_EM,   
 by = c("id\_municipio", "sigla\_uf")) %>%   
 mutate(i613 = matriculados15a17\_EM/pop\_15a17\_atual) %>%  
 mutate(i613\_pad = (i613 - mean(i613))/sdp(i613))%>%  
 select(-pop\_15a17\_atual, -matriculados15a17\_EM) %>%  
 arrange(desc(i613\_pad))  
  
# salvamos o arquivo  
write.csv(i613, "i613.csv", row.names = FALSE)

## Script i614

Este script em R refere-se ao [Indicador Nota média no ENEM (i614)](#i614)

library(tidyverse)  
  
# Indicador Nota Media no ENEM (i614)  
  
# ------------------------------------------------------------------  
# PASSO 0: preliminares  
# ------------------------------------------------------------------  
  
# definimos uma funcao para desvio padrao populacional  
sdp <- function(x) {  
 sd(x)\*(sqrt((length(x)-1)/length(x)))  
}  
  
# importamos tabela com dados dos municipios mais populosos  
top100\_mun\_cod <- read.csv("top100\_mun\_cod.csv", stringsAsFactors = FALSE)  
top100\_mun\_cod <- top100\_mun\_cod %>%   
 transform(id\_municipio = as.integer(id\_municipio))  
  
# ------------------------------------------------------------------  
# PASSO 1: obter dados do ENEM  
# ------------------------------------------------------------------  
  
#importamos a base de dados do ENEM  
ENEM\_2019 <- data.table::fread(input='C:/Users/User/OneDrive/  
 Documentos/ICE 2021/d6 Capital Humano/  
 sd61 Acesso e Qualidade da Mão de Obra Básica/  
 Downloads/DADOS/MICRODADOS\_ENEM\_2019.csv',  
 integer64='character',  
 skip=0, #Ler do inicio  
 nrow=-1, #Ler todos os registros  
 na.strings = "",   
 showProgress = TRUE)  
  
# selecionamos variaveis de interesse  
ENEM\_2019 <- ENEM\_2019 %>%   
 select(NU\_INSCRICAO,   
 NU\_ANO,   
 CO\_MUNICIPIO\_ESC,   
 CO\_MUNICIPIO\_RESIDENCIA,   
 CO\_ESCOLA,   
 TP\_ST\_CONCLUSAO,   
 TP\_ENSINO,   
 NU\_NOTA\_CH,   
 NU\_NOTA\_CN,   
 NU\_NOTA\_LC,   
 NU\_NOTA\_MT,   
 NU\_NOTA\_REDACAO,   
 Q001,   
 Q002)  
  
# unimos as duas tabelas. selecionar apenas os inscritos dos municipios do ICE  
ENEM\_100mun <- top100\_mun\_cod %>%   
 left\_join(ENEM\_2019, by = c("id\_municipio" = "CO\_MUNICIPIO\_ESC"))  
  
# salvamos essa tabela para usar em outros indices  
write.csv(ENEM\_100mun, "ENEM\_100mun.csv", row.names = FALSE)  
  
# ------------------------------------------------------------------  
# PASSO 1: alternativo  
# ------------------------------------------------------------------  
  
# importar dadoss ENEM dos munipios do ICE  
ENEM\_100mun <- read.csv('ENEM\_100mun.csv')  
  
# ------------------------------------------------------------------  
# PASSO 2: obter notas medias no ENEM  
# ------------------------------------------------------------------  
  
# calculamos a nota media de cada aluno considerando as 5 provas   
# ciencias naturais, ciencas humanas, matematica, redacao, linguagens e codigos  
nota\_media\_ENEM <- ENEM\_100mun %>%   
 mutate(nota\_media = (NU\_NOTA\_CH + NU\_NOTA\_CN + NU\_NOTA\_LC +   
 NU\_NOTA\_MT + NU\_NOTA\_REDACAO)/5) %>%   
 select (-Q001, -Q002, -NU\_ANO, -NU\_NOTA\_CH,   
 -NU\_NOTA\_CN, -NU\_NOTA\_LC, -NU\_NOTA\_MT,   
 -NU\_NOTA\_REDACAO, -CO\_MUNICIPIO\_RESIDENCIA,   
 -CO\_ESCOLA, -TP\_ST\_CONCLUSAO, -TP\_ENSINO) %>%  
 drop\_na()  
# ------------------------------------------------------------------  
# PASSO 3: calcular indicador  
# ------------------------------------------------------------------  
  
# calculamos o indicador  
i614 <- nota\_media\_ENEM %>%   
 group\_by(id\_municipio, nome, sigla\_uf) %>%   
 summarise(i614 = mean(nota\_media)) %>%  
 ungroup() %>%  
 mutate(i614\_pad = (i614 - mean(i614))/sdp(i614)) %>%  
 arrange(desc(i614\_pad))  
  
# exportamos o indicador  
write.csv(i614, "i614.csv", row.names = FALSE)

## Script i615

Este script em R refere-se ao [Indicador Proporção de Matriculados no Ensino Técnico e Profissionalizante (i615)](#i615)

library(tidyverse)  
library("basedosdados")  
library(readxl)  
  
# Indicador Prop. de Matriculados no Ensino Tecnico e Profissionalizante (i615)  
  
# ------------------------------------------------------------------  
# PASSO 0: preliminares  
# ------------------------------------------------------------------  
  
# Definir projeto no Google Cloud  
set\_billing\_id("workshop-teste-322616")  
  
# definimos uma funcao para desvio padrao populacional  
sdp <- function(x) {  
 sd(x)\*(sqrt((length(x)-1)/length(x)))  
}  
  
# importamos tabela com dados dos municipios mais populosos  
top100\_mun\_cod <- read.csv("top100\_mun\_cod.csv", stringsAsFactors = FALSE)  
top100\_mun\_cod <- top100\_mun\_cod %>%   
 transform(id\_municipio = as.character(id\_municipio))  
  
# ------------------------------------------------------------------  
# PASSO 1: obter numero de inscritos no ensino tecnico  
# ------------------------------------------------------------------  
  
query <- "SELECT ano, id\_municipio, sigla\_uf, idade\_referencia, etapa\_ensino FROM `basedosdados.br\_inep\_censo\_escolar.matricula`   
WHERE ano = 2020  
AND (etapa\_ensino IN ('30','31','32','33','34','35','36','37','38','39','40','67','68','73','74'))  
AND id\_municipio IN ('3550308','3304557','5300108','2927408','2304400','3106200','1302603','4106902','2611606','5208707','1501402','4314902','3518800'  
,'3509502','2111300','3304904','2704302','3301702','5002704','2408102','2211001','3548708','3303500','2507507','3549904','3547809'  
,'3543402','2607901','3534401','3170206','3552205','3118601','2800308','2910800','5103403','4209102','5201405','4113700','3136702'  
,'1100205','1500800','3205002','4305108','3303302','3300456','1600303','3301009','4205407','3205200','3529401','3305109','3549805'  
,'3530607','3106705','3548500','4115200','3513801','3525904','1400100','3143302','1200401','2504009','3538709','3510609','2609600'  
,'5201108','3201308','3506003','3523107','3551009','3205309','2604106','2303709','4202404','3516200','4119905','2611101','4304606'  
,'4314407','2933307','3154606','3170107','2610707','4104808','3541000','4125506','3518701','3554102','3526902','3303906','1506807'  
,'1721000','2905701','2408003','3552502','3552809','5108402','3552403','4316907','4309209','1504208')"  
educ\_prof <- read\_sql(query)  
  
# contamos a quantidade de alunos matriculados na Educacao Profissional em cada municipio  
matriculados\_EP <- educ\_prof %>%   
 count(id\_municipio, sigla\_uf, sort = TRUE) %>%   
 rename(matriculados\_EP = n)  
  
# ------------------------------------------------------------------  
# PASSO 2: obter populacao de 15 ou mais em 2010  
# ------------------------------------------------------------------  
  
# importamos tabela do censo 2010 com dados de populacao por municipio e idade  
censo2010 <- read\_excel("C:/Users/User/OneDrive/Documentos/ICE 2021/  
 d6 Capital Humano/  
 sd61 Acesso e Qualidade da Mão de Obra Básica/Downloads/  
 censo2010.xlsx", skip = 6)  
pop\_mais15 <- top100\_mun\_cod %>%   
 left\_join(censo2010, by = c("id\_municipio" = "Cód.")) %>%   
 select(id\_municipio, nome, sigla\_uf, Total, "15 a 19 anos", "20 a 24 anos",   
 "25 a 29 anos", 38:55 ,-"Idade ignorada", -"80 anos ou mais",-Total) %>%  
 mutate(pop\_mais15 = rowSums(across(where(is.numeric))))  
  
# ------------------------------------------------------------------  
# PASSO 3: obter populacao dos municipios em 2010  
# ------------------------------------------------------------------  
  
# importar tabela com populacao total dos municipios em 2010  
query <- "SELECT \* FROM `basedosdados.br\_ibge\_populacao.municipio` WHERE ano = 2010"  
pop\_2010 <- read\_sql(query)  
  
# ------------------------------------------------------------------  
# PASSO 4: obter estimativa populacional de 2021  
# ------------------------------------------------------------------  
  
# obter populacao de 2021  
pop\_2021 <- read\_excel('C:/Users/User/OneDrive/Documentos/ICE 2021/  
 estimativa\_dou\_2021.xls', sheet = 2, skip = 1) %>%  
 na.omit() %>%  
 select(nome = 'NOME DO MUNICÍPIO',  
 sigla\_uf = 'UF',  
 populacao = 'POPULAÇÃO ESTIMADA') %>%  
 inner\_join(top100\_mun\_cod, by = c('nome', 'sigla\_uf'))  
  
# ------------------------------------------------------------------  
# PASSO 5: calcular fator de crescimento populacional   
# ------------------------------------------------------------------  
  
# calcular fator de crescimento populacional  
cresc\_pop <- pop\_2010 %>% inner\_join(pop\_2021, by = "id\_municipio",   
 suffix = c("\_2010", "\_2021")) %>%   
 mutate(cresc\_pop = populacao\_2021/populacao\_2010) %>%  
 select (-ano, -populacao\_2010, -populacao\_2021)  
  
# ------------------------------------------------------------------  
# PASSO 6: calcular populacao estimada com mais de 15 em 2021  
# ------------------------------------------------------------------  
  
# Calcular populacao estimada em cada municipio acima de 15 anos  
pop\_mais15\_atual <- pop\_mais15 %>%   
 inner\_join(cresc\_pop, by = c("id\_municipio", "nome", "sigla\_uf")) %>%  
 mutate(pop\_mais15\_atual = pop\_mais15 \* cresc\_pop) %>%  
 select(id\_municipio, nome, sigla\_uf, pop\_mais15\_atual)  
  
# ------------------------------------------------------------------  
# PASSO 7: calcular indicador  
# ------------------------------------------------------------------  
  
# por fim, calculamos o indice   
i615 <- pop\_mais15\_atual %>%   
 inner\_join(matriculados\_EP, by = c("id\_municipio", "sigla\_uf")) %>%   
 mutate(i615 = matriculados\_EP/pop\_mais15\_atual) %>%  
 mutate(i615\_pad = (i615 - mean(i615))/sdp(i615))%>%  
 select(id\_municipio, nome, sigla\_uf, i615, i615\_pad) %>%  
 arrange(desc(i615\_pad))  
  
# salvamos o arquivo  
write.csv(i615, "i615.csv", row.names = FALSE)

## Script i621

Este script em R refere-se ao [Indicador Proporção de Adultos com pelo Menos Ensino Superior Completo (i621)](#i621)

library(tidyverse)  
  
# Indicador Proporcao de Adultos com pelo Menos Ensino Superior Completo (i621)  
  
# ------------------------------------------------------------------  
# PASSO 0: preliminares  
# ------------------------------------------------------------------  
  
# definimos uma funcao para desvio padrao populacional  
sdp <- function(x) {  
 sd(x)\*(sqrt((length(x)-1)/length(x)))  
}  
  
# importamos tabela com dados dos municipios mais populosos  
top100\_mun\_cod <- read.csv("top100\_mun\_cod.csv", stringsAsFactors = FALSE)  
top100\_mun\_cod <- top100\_mun\_cod %>%   
 transform(id\_municipio = as.integer(id\_municipio))  
  
# ------------------------------------------------------------------  
# PASSO 1: obter dados do ENEM  
# ------------------------------------------------------------------  
  
#importamos a base de dados do ENEM  
ENEM\_2019 <- data.table::fread(input='C:/Users/User/OneDrive/Documentos/  
 ICE 2021/d6 Capital Humano/  
 sd61 Acesso e Qualidade da Mão de Obra Básica/  
 Downloads/DADOS/MICRODADOS\_ENEM\_2019.csv',  
 integer64='character',  
 skip=0, #Ler do inicio  
 nrow=-1, #Ler todos os registros  
 na.strings = "",   
 showProgress = TRUE)  
  
# selecionamos variaveis de interesse  
ENEM\_2019 <- ENEM\_2019 %>%   
 select(NU\_INSCRICAO,   
 NU\_ANO,   
 CO\_MUNICIPIO\_ESC,   
 CO\_MUNICIPIO\_RESIDENCIA,   
 CO\_ESCOLA,   
 TP\_ST\_CONCLUSAO,   
 TP\_ENSINO,   
 NU\_NOTA\_CH,   
 NU\_NOTA\_CN,   
 NU\_NOTA\_LC,   
 NU\_NOTA\_MT,   
 NU\_NOTA\_REDACAO,   
 Q001,   
 Q002)  
  
# unimos as duas tabelas para selecionar apenas os inscritos municipios ICE  
ENEM\_100mun <- top100\_mun\_cod %>%   
 left\_join(ENEM\_2019, by = c("id\_municipio" = "CO\_MUNICIPIO\_ESC"))  
  
# salvamos essa tabela para usar em outros indices  
write.csv(ENEM\_100mun, "ENEM\_100mun.csv", row.names = FALSE)  
  
# ------------------------------------------------------------------  
# PASSO 1: alternativo  
# ------------------------------------------------------------------  
  
# importar dadoss ENEM dos munipios do ICE  
ENEM\_100mun <- read.csv('ENEM\_100mun.csv')  
  
# ------------------------------------------------------------------  
# PASSO 2: dados do ENEM  
# ------------------------------------------------------------------  
  
# contamos a quantidade de inscritos por municipio  
inscritos\_mun <- ENEM\_100mun %>%   
 count(id\_municipio, nome, sigla\_uf, sort = TRUE) %>%   
 rename(inscritos = n)  
  
# criamos um vetor com as respostas do questionario que selecionam   
# pai ou mae com ES completo (ver dicionario dos microdados ENEM 2019)  
alvo\_ES <- c("F", "G")  
  
# contamos a quantidade de inscritos que declaram ter pai com ES completo  
pai\_ES <- ENEM\_100mun %>%   
 filter(Q001 %in% alvo\_ES) %>%   
 count(id\_municipio, nome, sigla\_uf, sort = TRUE) %>%   
 rename(pai\_ES = n)  
  
# contamos a quantidade de inscritos que declaram ter mae com ES completo  
mae\_ES <- ENEM\_100mun %>%   
 filter(Q002 %in% alvo\_ES) %>%   
 count(id\_municipio, nome, sigla\_uf, sort = TRUE) %>%   
 rename(mae\_ES = n)  
  
# juntamos as tres tabelas:  
# inscritos totais, inscritos com pais com EM, incritos com mae com EM)  
paimae\_ES <- pai\_ES %>%   
 inner\_join(mae\_ES, by = c('id\_municipio', 'nome', 'sigla\_uf')) %>%   
 inner\_join(inscritos\_mun, by = c('id\_municipio', 'nome', 'sigla\_uf')) %>%  
 mutate(prop\_paiES = pai\_ES/inscritos,  
 prop\_maeES = mae\_ES/inscritos) %>%  
 mutate(i621= (prop\_paiES + prop\_maeES)/2)  
  
# ------------------------------------------------------------------  
# PASSO 3: calcular indicador  
# ------------------------------------------------------------------  
  
# calculamos e padronizamos o indice  
i621 <- paimae\_ES %>%   
 select(id\_municipio, nome, sigla\_uf, i621) %>%   
 transform(id\_municipio = as.character(id\_municipio)) %>%   
 mutate(i621\_pad = (i621 - mean(i621))/sdp(i621))   
  
# exportamos o indicador  
write.csv(i621, "i621.csv", row.names = FALSE)

## Script i622

Este script em R refere-se ao [Indicador Proporção de Alunos Concluintes em Cursos de Alta Qualidade (i622)](#i622)

library(data.table)  
library(readxl)  
library(tidyverse)  
  
# Indicador Proporcao de Alunos Concluintes em Cursos de Alta Qualidade (i622)  
  
# ------------------------------------------------------------------  
# PASSO 0: preliminares  
# ------------------------------------------------------------------  
  
# definimos uma funÃ§Ã£o para desvio padrao populacional  
sdp <- function(x) {  
 sd(x)\*(sqrt((length(x)-1)/length(x)))  
}  
  
# importamos tabela com dados dos municipios mais populosos  
top100\_mun\_cod <- read.csv("top100\_mun\_cod.csv", stringsAsFactors = FALSE)  
top100\_mun\_cod <- top100\_mun\_cod %>%   
 transform(id\_municipio = as.numeric(id\_municipio))  
  
# ------------------------------------------------------------------  
# PASSO 1: obter concluintes em cursos avaliados pelo enade  
# ------------------------------------------------------------------  
  
# importando os dados. as planilhas sÃ£o baixadas no site do INEP  
conceito\_enade2017 = read\_excel("C:/Users/User/OneDrive/Documentos/ICE 2021/  
 d6 Capital Humano/sd62/  
 resultados\_conceito\_enade\_2017.xlsx")  
conceito\_enade2018 = read\_excel("C:/Users/User/OneDrive/Documentos/ICE 2021/  
 d6 Capital Humano/sd62/  
 resultados\_conceito\_enade\_2018.xlsx")  
conceito\_enade2019 = read\_excel("C:/Users/User/OneDrive/Documentos/ICE 2021/  
 d6 Capital Humano/sd62/Conceito\_Enade\_2019.xlsx")  
  
# juntando as bases de conceito  
conceito\_enade = bind\_rows(conceito\_enade2017, conceito\_enade2018, conceito\_enade2019)  
  
# selecionar apenas os cursos dos 100 municipios mais populosos  
enade\_ice <- top100\_mun\_cod %>%   
 left\_join(conceito\_enade, by = c("id\_municipio" = "Código do Município")) %>%   
 select(-"Município do Curso", - "Sigla da UF") %>%  
 rename(enade\_faixa = "Conceito Enade (Faixa)",  
 concl\_inscritos = "Nº de Concluintes Inscritos")  
  
enade\_ice\_resumo <- enade\_ice %>%   
 group\_by(id\_municipio, nome, sigla\_uf, enade\_faixa) %>%   
 summarise(concluintes = sum(concl\_inscritos))  
  
# Obter total de concluintes em todos cursos avaliados pelo ENADE  
concl\_grad <- enade\_ice\_resumo %>%   
 group\_by(id\_municipio, nome, sigla\_uf) %>%   
 summarise(tot\_concl = sum(concluintes))  
  
  
# obter total de concluintes em cursos com nota 4 ou 5 no ENADE  
concl\_alta\_quali <- enade\_ice\_resumo %>%   
 filter(enade\_faixa== 4 | enade\_faixa==5) %>%   
 group\_by(id\_municipio, nome, sigla\_uf) %>%   
 summarise(tot\_alta\_quali = sum(concluintes))  
  
# ------------------------------------------------------------------  
# PASSO 2: calcular indicador  
# ------------------------------------------------------------------  
  
#calcular indice   
i622 <- concl\_grad %>%   
 left\_join(concl\_alta\_quali, by = c("id\_municipio", "nome", "sigla\_uf")) %>%  
 replace\_na(list(tot\_alta\_quali = 0)) %>%  
 mutate(i622 = tot\_alta\_quali/tot\_concl) %>%  
 transform(i622 = as.numeric(i622)) %>%  
 mutate(i622\_pad = (i622 - mean(i622))/sdp(i622)) %>%  
 select(id\_municipio, nome, sigla\_uf, i622, i622\_pad) %>%  
 arrange(desc(i622\_pad))  
  
# salvamos o arquivo  
write.csv(i622, "i622.csv", row.names = FALSE)

## Script d7

Este script em R refere-se aos [Indicadores do determinante Cultura (d7)](#d7)

library(tidyverse)  
  
# Indicadores do Determinante Cultura  
  
# ------------------------------------------------------------------  
# PASSO 0: preliminares  
# ------------------------------------------------------------------  
  
#definimos uma funcao para desvio padrão amostral  
sdp <- function(x) {  
 sd(x) \* (sqrt((length(x) - 1) / length(x)))  
}  
  
# importamos tabela com dados dos municipios mais populosos  
top100\_mun\_cod <-  
 read.csv("top100\_mun\_cod.csv", stringsAsFactors = FALSE)  
top100\_mun\_cod <- top100\_mun\_cod %>%  
 transform(id\_municipio = as.numeric(id\_municipio))  
  
# ------------------------------------------------------------------  
# PASSO 1: importar dados sobre pesquisas as palavras-chave  
# ------------------------------------------------------------------  
  
# importar arquivos referentes ao subdeterminante iniciativa (sd71)  
termo\_empreendedor <-  
 read.csv(  
 "C:/Users/User/OneDrive/Documentos/ICE 2021/  
 d7 Determinante Cultura/empreendedor.csv",  
 stringsAsFactors = FALSE,  
 header = 1,  
 skip = 2,  
 fileEncoding = "UTF-8"  
 )  
termo\_mei <- read.csv(  
 "C:/Users/User/OneDrive/Documentos/ICE 2021/  
 d7 Determinante Cultura/mei.csv",  
 stringsAsFactors = FALSE,  
 header = 1,  
 skip = 2,  
 fileEncoding = "UTF-8"  
)  
  
# importar arquivos referentes ao subdeterminante instituicoes (sd72)  
termo\_sebrae <-  
 read.csv(  
 "C:/Users/User/OneDrive/Documentos/ICE 2021/  
 d7 Determinante Cultura/sebrae.csv",  
 stringsAsFactors = FALSE,  
 header = 1,  
 skip = 2,  
 fileEncoding = "UTF-8"  
 )  
termo\_franquia <-  
 read.csv(  
 "C:/Users/User/OneDrive/Documentos/ICE 2021/  
 d7 Determinante Cultura/franquia.csv",  
 stringsAsFactors = FALSE,  
 header = 1,  
 skip = 2,  
 fileEncoding = "UTF-8"  
 )  
termo\_simplesnacional <-  
 read.csv(  
 "C:/Users/User/OneDrive/Documentos/  
 ICE 2021/d7 Determinante Cultura/  
 simples nacional.csv",  
 stringsAsFactors = FALSE,  
 header = 1,  
 skip = 2,  
 fileEncoding = "UTF-8"  
 )  
termo\_senac <- read.csv(  
 "C:/Users/User/OneDrive/Documentos/ICE 2021/  
 d7 Determinante Cultura/senac.csv",  
 stringsAsFactors = FALSE,  
 header = 1,  
 skip = 2,  
 fileEncoding = "UTF-8"  
)  
  
  
# calcular indicador pesquisas com o termo 'empreendedor'  
i711 <- top100\_mun\_cod %>%  
 left\_join(termo\_empreendedor, by = c("nome" = "City")) %>%  
 replace\_na(list(empreendedor...2020. = 0)) %>%  
 rename(i711 = empreendedor...2020.) %>%  
 mutate(i711\_pad = (i711 - mean(i711)) / sdp(i711)) %>%  
 arrange(desc(i711\_pad))  
  
# calcular indicador pesquisas com o termo 'mei'  
i712 <- top100\_mun\_cod %>%  
 left\_join(termo\_mei, by = c("nome" = "City")) %>%  
 replace\_na(list(MEI...2020. = 0)) %>%  
 rename(i712 = MEI...2020.) %>%  
 mutate(i712\_pad = (i712 - mean(i712)) / sdp(i712)) %>%  
 arrange(desc(i712\_pad))  
  
  
# calcular indicador pesquisas com o termo 'sebrae'  
i721 <-  
 top100\_mun\_cod %>% left\_join(termo\_sebrae, by = c("nome" = "City")) %>%  
 replace\_na(list(Sebrae...2020. = 0)) %>%  
 rename(i721 = Sebrae...2020.) %>%  
 mutate(i721\_pad = (i721 - mean(i721)) / sdp(i721)) %>%  
 arrange(desc(i721\_pad))  
  
# calcular indicador pesquisas com o termo 'franquia'  
i722 <-  
 top100\_mun\_cod %>% left\_join(termo\_franquia, by = c("nome" = "City")) %>%  
 replace\_na(list(Franquia...2020. = 0)) %>%  
 rename(i722 = Franquia...2020.) %>%  
 mutate(i722\_pad = (i722 - mean(i722)) / sdp(i722)) %>%  
 arrange(desc(i722\_pad))  
  
# calcular indicador pesquisas com o termo 'simples nacional'  
i723 <-  
 top100\_mun\_cod %>%   
 left\_join(termo\_simplesnacional, by = c("nome" = "City")) %>%  
 replace\_na(list(SIMPLES.Nacional...2020. = 0)) %>%  
 rename(i723 = SIMPLES.Nacional...2020.) %>%  
 mutate(i723\_pad = (i723 - mean(i723)) / sdp(i723)) %>%  
 arrange(desc(i723\_pad))  
  
# calcular indicador pesquisas com o termo 'senac'  
i724 <-  
 top100\_mun\_cod %>% left\_join(termo\_senac, by = c("nome" = "City")) %>%  
 replace\_na(list(Senac...2020. = 0)) %>%  
 rename(i724 = Senac...2020.) %>%  
 mutate(i724\_pad = (i724 - mean(i724)) / sdp(i724)) %>%  
 arrange(desc(i724\_pad))  
  
# exportar indicadores do determinante cultura  
write.csv(i711, "i711.csv", row.names = FALSE)  
write.csv(i712, "i712.csv", row.names = FALSE)  
write.csv(i721, "i721.csv", row.names = FALSE)  
write.csv(i722, "i722.csv", row.names = FALSE)  
write.csv(i723, "i723.csv", row.names = FALSE)  
write.csv(i724, "i724.csv", row.names = FALSE)

## Padronização

Voltar para [Padronização](#pad)

\*\*\* ICE 2020 \*\*\*  
  
\*\*\* Análise Fatorial \*\*\*  
  
clear  
log using "C:\Users\User\OneDrive\Documentos\ICE 2021", replace   
set more off  
cd "C:\Users\User\OneDrive\Documentos\ICE 2021"  
pwd  
  
import excel "C:\Users\User\OneDrive\Documentos\ICE 2021\Análise Fatorial\indicadores.xlsx", firstrow  
  
\*\*Padronização  
  
gen N = \_N  
foreach var of varlist i111 i112 i113 i121 i122 i123 i124 i131 i132 i133 i211 i212 i213 i221 i222 i223 i224 i311 i312 i313 i321 i322 i323 i411 i412 i413 i511 i512 i513 i514 i515 i521 i522 i523 i524 i611 i612 i613 i614 i615 i621 i622 i623 i711 i712 i721 i722 i723 i724 {  
egen `var'\_media= mean(`var')  
egen `var'\_dp= sd(`var')  
gen `var'\_dpp= `var'\_dp \*sqrt((N-1)/N)  
gen `var'\_linha= (`var' - `var'\_media)/ `var'\_dpp  
drop `var'\_media `var'\_dp `var'\_dpp  
}  
  
gen s11=i111\_linha+i112\_linha+i113\_linha  
gen s12=i121\_linha+i122\_linha+i123\_linha+i124\_linha  
gen s13=i131\_linha+i132\_linha+i133\_linha  
gen s21=i211\_linha+i212\_linha+i213\_linha  
gen s22=i221\_linha+i222\_linha+i223\_linha+i224\_linha  
gen s31=i311\_linha+i312\_linha+i313\_linha  
gen s32=i321\_linha+i322\_linha+i323\_linha  
gen s41=i411\_linha+i412\_linha+i413\_linha  
gen s51=i511\_linha+i512\_linha+i513\_linha+i514\_linha+i515\_linha  
gen s52=i521\_linha+i522\_linha+i523\_linha+i524\_linha  
gen s61=i611\_linha+i612\_linha+i613\_linha+i614\_linha+i615\_linha  
gen s62=i621\_linha+i622\_linha+i623\_linha  
gen s71=i711\_linha+i712\_linha  
gen s72=i721\_linha+i722\_linha+i723\_linha+i724\_linha  
  
foreach var of varlist s11 s12 s13 s21 s22 s31 s32 s41 s51 s52 s61 s62 s71 s72{  
egen `var'\_media= mean(`var')  
egen `var'\_dp= sd(`var')  
gen `var'\_dpp= `var'\_dp \*sqrt((N-1)/N)  
gen `var'\_linha= ((`var' - `var'\_media)/ `var'\_dpp) + 6  
drop `var'\_media `var'\_dp `var'\_dpp  
}  
  
gen d1=s11\_linha+s12\_linha+s13\_linha  
gen d2=s21\_linha+s22\_linha  
gen d3=s31\_linha+s32\_linha  
gen d4=s41\_linha  
gen d5=s51\_linha+s52\_linha  
gen d6=s61\_linha+s62\_linha  
gen d7=s71\_linha+s72\_linha  
  
foreach var of varlist d1 d2 d3 d4 d5 d6 d7 {  
egen `var'\_media= mean(`var')  
egen `var'\_dp= sd(`var')  
gen `var'\_dpp= `var'\_dp \*sqrt((N-1)/N)  
gen `var'\_linha= ((`var' - `var'\_media)/ `var'\_dpp) + 6  
drop `var'\_media `var'\_dp `var'\_dpp  
}

## Análise de Componentes Principais

Voltar para [Análise de Componentes Principais](#pca)

\*\*Análise fatorial  
  
factor d1\_linha d2\_linha d3\_linha d4\_linha d5\_linha d6\_linha d7\_linha, pcf mineigen(0.6)  
  
rotate  
  
predict ice1 ice2 ice3  
  
estat kmo  
  
gen ice = ice1 + ice2 + ice3  
egen ice\_media= mean(ice)  
egen ice\_dp= sd(ice)  
gen ice\_dpp= ice\_dp \*sqrt((N-1)/N)  
gen ice\_final= ((ice - ice\_media)/ ice\_dpp) + 6  
drop ice\_media ice\_dp ice\_dpp  
gsort -ice\_final

1. <https://www.cnj.jus.br/gestao-estrategica-e-planejamento/estrategia-nacional-do-poder-judiciario-2009-2014/indicadores/03-taxa-de-congestionamento/> [↑](#footnote-ref-29)
2. pdf\_contas.pdf, pp.15-17 [↑](#footnote-ref-40)
3. IPTU, ITBI, ITR, IRRF [↑](#footnote-ref-41)
4. esses outliers aparecem em anúncios cujo preço refere-se a um prédio e a área a uma sala do prédio [↑](#footnote-ref-62)