

Relatório Final ICE 2021

Douglas Moreira Mangini e Pedro Henrique Borges da Silva

02/09/2021

Contents

1	Introdução	3
2	Determinante Ambiente regulatório	6
2.1	Subdeterminante Tempo de processos	6
2.2	Subdeterminante Tributação	6
2.3	Subdeterminante Complexidade burocrática	7
3	Determinante Infraestrutura	9
3.1	Subdeterminante Transporte Interurbano	9
3.2	Subdeterminante Condições Urbanas	11
4	Determinante Mercados	13
4.1	Subdeterminante Desenvolvimento Econômico	13
4.2	Subdeterminante Clientes Potenciais	13
5	Determinante Acesso a Capital	14
5.1	Subdeterminante Capital Disponível	14
6	Determinante Inovação	17
6.1	Subdeterminante Inputs	17
6.2	Subdeterminante Outputs	19
7	Determinante Capital Humano	22
7.1	Subdeterminante Acesso e Qualidade da Mão de Obra Básica	22
7.2	Subdeterminante Acesso e Qualidade da Mão de Obra Qualificada	24
8	Determinante Cultura	26
8.1	Subdeterminante Iniciativa	26
8.2	Subdeterminante Instituições	26
9	Análise de Componentes Principais	27
9.1	Padronização	27
9.2	Análise de Componentes Principais	27
10	Apêndice	30
10.1	Script i323	57
10.2	Script i411	59
10.3	Script i412	60
10.4	Script i413	62
10.5	Script i511	63
10.6	Script i512	65

10.7 Script i513	68
10.8 Script i514	70
10.9 Script i515	71
10.10Script i521	72
10.11Script i522	74
10.12Script i523	76
10.13Script i524	78
10.14Script i611	80
10.15Script i612	81
10.16Script i613	83
10.17Script i614	85
10.18Script i615	87
10.19Script i621	89
10.20Script i622	92
10.21Script d7	93
10.22Padronização	96
10.23Análise de Componentes Principais	97

1 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:

Table 1: Determinantes, subdeterminantes e indicadores com com-
põem 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 m ²
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 Poucado per capita

Determinante	Subdeterminante	Indicador
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](#)) 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](#) deste relatório. Além disso, todos arquivos de dados e scripts estão disponíveis no drive do projeto, disponível neste [link](#). Por fim, os códigos em R também estão disponíveis em repositório [GitHub](#).

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](#).

2 Determinante Ambiente regulatório

2.1 Subdeterminante Tempo de processos

clique [aqui](#) para ver o script

2.1.1 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. Atribuímos nota zero às cidades faltantes. Detalhes estão comentados no script em apêndice.

2.1.2 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 db*, *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.

2.1.3 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 definida¹ como

$$1 - \frac{\text{baixados}}{\text{novos} + \text{pendentes}}$$

. O indicador é o inverso da taxa de congestionamento.

2.2 Subdeterminante Tributação

clique [aqui](#) para ver o script

2.2.1 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.

¹<https://www.cnj.jus.br/gestao-estrategica-e-planejamento/estrategia-nacional-do-poder-judiciario-2009-2014/indicadores/03-taxa-de-congestionamento/>

2.2.2 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.

2.2.3 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.

2.2.4 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.

2.3 Subdeterminante Complexidade burocrática

clique [aqui](#) para ver o script

2.3.1 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 passado² 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 tributos³ na arrecadação total. Detalhes no script em apêndice.

2.3.2 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.

²pdf_contas.pdf, pp.15-17

³IPTU, ITBI, ITR, IRRF

2.3.3 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.

3 Determinante Infraestrutura

3.1 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](#)

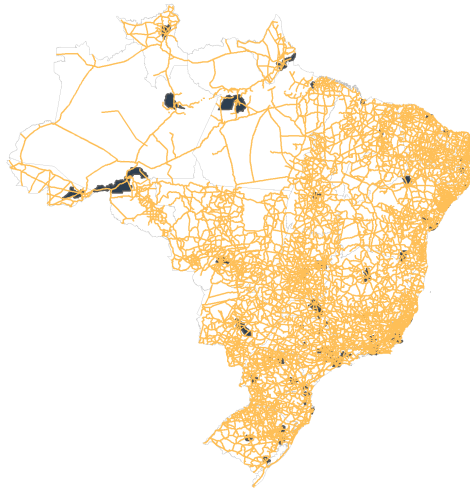


Figure 1: Exemplo de rodovia

3.1.1 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.

3.1.2 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).

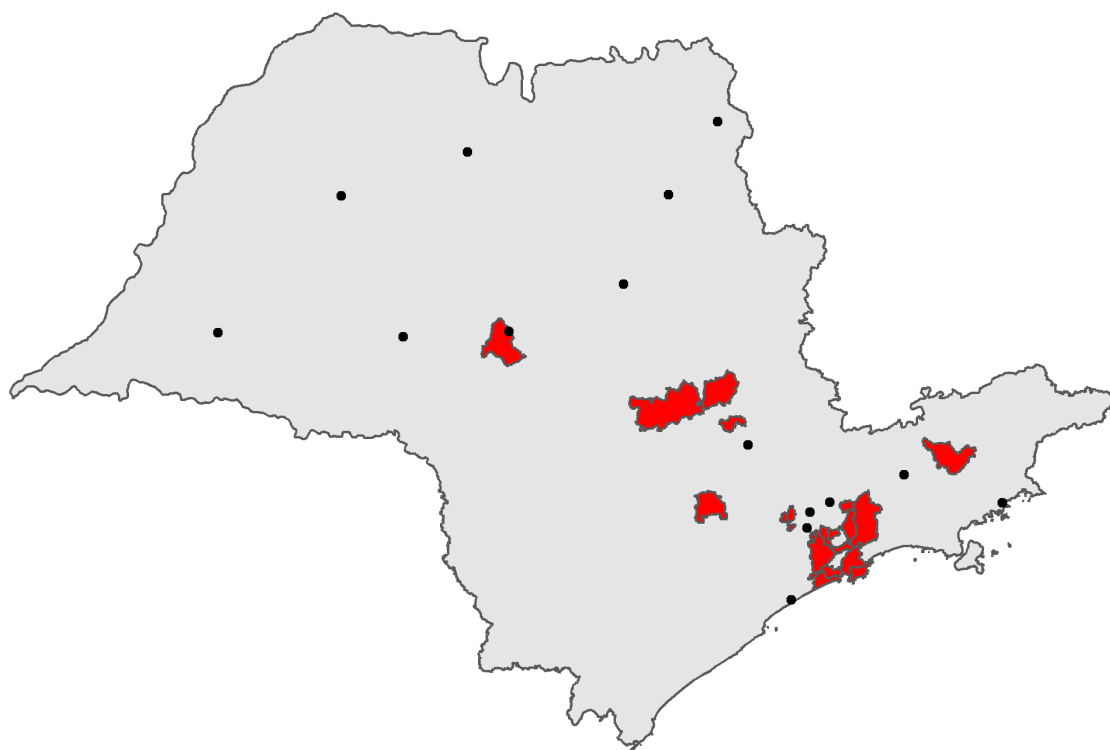


Figure 2: Exemplo: estado de SP e aeroportos mais próximos

3.1.3 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 flúvios do Amazonas. Tomamos o centro do polígono que representa o município como referência (ver figura).

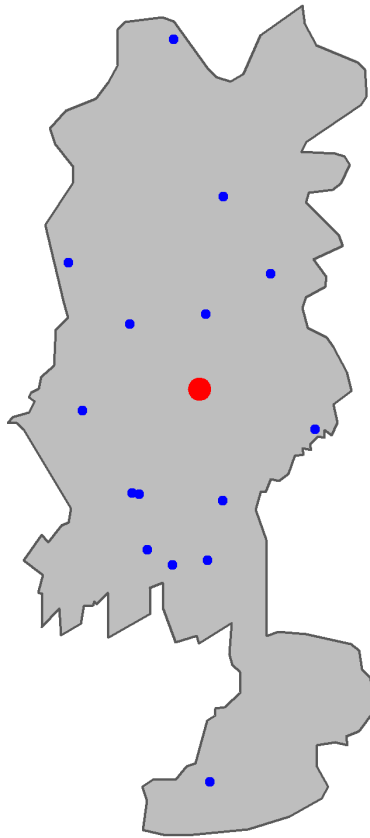


Figure 3: Exemplo: centro de Ananindeua-PA e portos mais próximos

3.2 Subdeterminante Condições Urbanas

clique [aqui](#) para ver o script

3.2.1 Acesso à internet rápida

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

Período: 2020

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

3.2.2 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. Excluimos do cálculo os anúncios que implicavam um preço de metro quadrado menor do que 100 reais e maior do que 10.000 reais⁴. 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.

3.2.3 Custo da energia elétrica

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

3.2.4 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** em R para reproduzir o cálculo no apêndice deste documento.

⁴esses outliers aparecem em anúncios cujo preço refere-se a um prédio e a área a uma sala do prédio

4 Determinante Mercados

4.1 Subdeterminante Desenvolvimento Econômico

clique [aqui](#) para ver o script

4.1.1 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.

4.1.2 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.

4.1.3 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.

4.2 Subdeterminante Clientes Potenciais

clique [aqui](#) para ver o script

4.2.1 PIB per capita

Fonte: Sidra-IBGE

Período: 2018

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

4.2.2 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.

4.2.3 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;
2. 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
3. Calcular o indicador para cada município de acordo com a seguinte fórmula:

$$i323 = \frac{\text{despesas + investimentos}}{\text{quantidade de empresas com pelo menos um funcionário}}$$

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

5 Determinante Acesso a Capital

5.1 Subdeterminante Capital Disponível

5.1.1 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](#) 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”;
2. Obter PIB dos municípios do ICE a preços correntes;
 - Acessar o site do [SIDRA-IBGE](#) 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;
3. Calcular o indicador para cada município de acordo com a seguinte fórmula:

$$i411 = \frac{\text{total de operações de crédito}}{\text{PIB}}$$

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

5.1.2 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](#) 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;
2. 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](#) ou outro similar. Transformar todos valores para real.
3. Obter PIB municipal;
 - Esse dado foi obtido no indicador [PIB per capita](#). 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;
4. Calcular o indicador para cada município de acordo com a seguinte fórmula:

$$i412 = \frac{\text{total de investimentos de risco}}{\text{PIB do município}}$$

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

5.1.3 Indicador Capital Poucado 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](#) 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”;
2. Obter estimativa populacional dos municípios do ICE;
 - Os dados mais recentes são de 2021. Acessar site do [IBGE](#) e baixar tabela;

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

$$i413 = \frac{\text{valor de depósitos em poupança e a prazo}}{\text{estimativa populacional}}$$

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

6 Determinante Inovação

6.1 Subdeterminante Inputs

6.1.1 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](#) e baixar conjunto de dados “[2017 a 2019] Discentes da Pós-Graduação stricto sensu do Brasil”.
2. 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;
3. Calcular o indicador para cada município do ICE de acordo com a seguinte fórmula:

$$i511 = \frac{\text{número de mestres e doutores em C\&T}}{\text{número de empresas com pelo menos um funcionário}}$$

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

6.1.2 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](#);
 - 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 FÍSICAS, 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”;
2. 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](#);

3. Calcular indicador de acordo com a seguinte fórmula:

$$i512 = \frac{\text{número de funcionários de C\&T}}{\text{número total de trabalhadores}}$$

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

6.1.3 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](#) e selecionar “Operações contratadas na forma direta e indireta não automática (2002 a 30.06.2021)”;
2. Obter dados de investimentos da FINEP nos municípios do ICE;
 - Acessar site da [FINEP](#), na seção “Projetos contratados e valores liberados” e selecionar a planilha “Projetos Contratados”;
3. 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;
4. Calcular o indicador para cada município de acordo com a fórmula:

$$i313 = \frac{\text{valores contratados BNDES} + \text{valores contratados FINEP}}{\text{nº de empresas com pelo menos um funcionário}}$$

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

6.1.4 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](#) ;
2. Acessar o estudo “[Indicadores de Parques Tecnológicos](#)” 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](#) deste relatório. O script e os arquivos de dados com os resultados dos cálculos estão disponíveis no drive do projeto.

6.1.5 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](#) deste relatório. O script e os arquivos de dados com os resultados dos cálculos estão disponíveis no drive do projeto.

6.2 Subdeterminante Outputs

6.2.1 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 \(INPI\)](#), na seção “estatísticas”, e “Indicadores de Propriedade Industrial”;
2. 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;
3. Calcular o indicador para cada município do ICE de acordo com a seguinte fórmula:

$$i521 = \frac{\text{nº de patentes PI ou CA ou MU}}{\text{nº de empresas com pelo menos um funcionário}}$$

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

6.2.2 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](#) 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](#);
2. 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;
3. Calcular indicador de acordo com a seguinte fórmula:

$$i522 = \frac{\text{n}^\circ \text{ de empresas de indústria inovadora}}{\text{n}^\circ \text{ de empresas com pelo menos um funcionário}}$$

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

6.2.3 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](#);
2. 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;
3. Calcular indicador para cada município de acordo com a seguinte fórmula:

$$\frac{\text{n}^\circ \text{ de empresas de economia criativa}}{\text{n}^\circ \text{ de empresas com pelo menos um funcionário}}$$

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

6.2.4 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;
2. 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;
 3. Calcular indicador para cada município de acordo com a seguinte fórmula:

$$i524 = \frac{\text{n}^{\circ} \text{ de empresas TIC}}{\text{n}^{\circ} \text{ de empresas com pelo menos um funcionário}}$$

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

7 Determinante Capital Humano

7.1 Subdeterminante Acesso e Qualidade da Mão de Obra Básica

7.1.1 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](#). 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.xlsx”, 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](#) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

7.1.2 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](#). 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](#) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

7.1.3 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;
2. 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;
3. 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;
4. Obter estimativa populacional do município em 2021;
- Dados obtidos no site do IBGE;
5. Calcular fator de crescimento populacional do município entre 2010 e 2021 a partir dos dados obtidos;
 6. Calcular população estimada do município em 2020 de pessoas entre 15 e 17;
 7. Calcular indicador para cada município por meio da seguinte fórmula:

$$i613 = \frac{\text{matriculados no Ensino Médio de 15 e 17 anos em 2020}}{\text{estimativa populacional de pessoas entre 15 a 17 em 2020}}$$

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

7.1.4 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](#) 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).
2. Calcular a nota média de cada aluno considerando as 5 provas do ENEM;
3. 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](#) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

7.1.5 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;
2. 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;
3. 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;
4. Obter estimativa populacional do município em 2021;
 - Dados obtidos no site do IBGE;

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

$$i615 = \frac{\text{matriculados no ensino técnico e profissionalizante}}{\text{estimativa populacional de pessoas com mais de 15 anos}}$$

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

7.2 Subdeterminante Acesso e Qualidade da Mão de Obra Qualificada

7.2.1 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](#). 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](#) deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

7.2.2 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](#). 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:

$$i622 = \frac{\text{total de alunos concluintes em cursos com notas 4 e 5 no ENADE}}{\text{total de alunos concluintes em cursos avaliados pelo ENADE}}$$

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

7.2.3 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 “VI 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](#);
 - A RAIS foi acessado por meio do data lake público Base dos Dados;
2. Obter a soma dos salários de todos dirigentes (cargos de interesse) em cada município.
 - A fonte desses dados também é a RAIS (vínculo);
3. Calcular o indicador i623 de acordo com a seguinte fórmula:

$$i623 = \frac{\text{soma dos salários de dirigentes}}{\text{nº de dirigentes}}$$

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

8 Determinante Cultura

8.1 Subdeterminante Iniciativa

8.1.1 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** deste relatório. O script e os arquivos de dados com resultados dos cálculos estão disponíveis no drive do projeto.

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

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

8.2 Subdeterminante Instituições

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

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

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

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

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

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

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

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

9 Análise de Componentes Principais

9.1 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 X é o determinante, sua versão padronizada será

$$\frac{X - \text{média}(X)}{\text{desvio padrao populacional}(X)}$$

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

9.2 Análise de Componentes Principais

Clique [aqui](#) 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).

```
Factor analysis/correlation          Number of obs   =      101
Method: principal-component factors   Retained factors =       3
Rotation: (unrotated)                Number of params =      18
```

Factor	Eigenvalue	Difference	Proportion	Cumulative
Factor1	2.64678	0.56270	0.3781	0.3781
Factor2	2.08408	1.41730	0.2977	0.6758
Factor3	0.66677	0.07951	0.0953	0.7711
Factor4	0.58726	0.13114	0.0839	0.8550
Factor5	0.45613	0.12582	0.0652	0.9201
Factor6	0.33031	0.10163	0.0472	0.9673
Factor7	0.22868	.	0.0327	1.0000

```
LR test: independent vs. saturated:  chi2(21) = 254.18 Prob>chi2 = 0.0000
```

Figure 4: 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.

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.

Factor loadings (pattern matrix) and unique variances

Variable	Factor1	Factor2	Factor3	Uniqueness
d1_linha	-0.1639	0.8430	-0.2847	0.1815
d2_linha	-0.3163	0.7915	-0.3098	0.1774
d3_linha	-0.3208	0.6828	0.5203	0.1601
d4_linha	0.7521	0.2416	-0.2101	0.3318
d5_linha	0.8696	0.1168	0.0177	0.2298
d6_linha	0.8575	0.0616	-0.0927	0.2524
d7_linha	0.5998	0.4526	0.4073	0.2694

Figure 5: Cargas Fatoriais

Factor analysis/correlation	Number of obs	=	101
Method: principal-component factors	Retained factors	=	3
Rotation: orthogonal varimax (Kaiser off)	Number of params	=	18

Factor	Variance	Difference	Proportion	Cumulative
Factor1	2.61046	0.86205	0.3729	0.3729
Factor2	1.74841	0.70966	0.2498	0.6227
Factor3	1.03875	.	0.1484	0.7711

LR test: independent vs. saturated: $\chi^2(21) = 254.18$ Prob> $\chi^2 = 0.0000$

Figure 6: fatores após rotação

Rotated factor loadings (pattern matrix) and unique variances

Variable	Factor1	Factor2	Factor3	Uniqueness
d1_linha	0.0587	0.8823	0.1913	0.1815
d2_linha	-0.1009	0.8871	0.1595	0.1774
d3_linha	-0.1566	0.3765	0.8207	0.1601
d4_linha	0.7944	0.1357	-0.1368	0.3318
d5_linha	0.8705	-0.1104	-0.0147	0.2298
d6_linha	0.8483	-0.0975	-0.1360	0.2524
d7_linha	0.6810	0.0325	0.5155	0.2694

Figure 7: cargas após rotação

Factor rotation matrix

	Factor1	Factor2	Factor3
Factor1	0.9683	-0.2282	-0.1013
Factor2	0.2480	0.8309	0.4981
Factor3	-0.0295	-0.5075	0.8612

Figure 8: Matriz de rotação

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

Kaiser-Meyer-Olkin measure of sampling adequacy

Variable	kmo
d1_linha	0.6331
d2_linha	0.6442
d3_linha	0.7037
d4_linha	0.7987
d5_linha	0.6855
d6_linha	0.7001
d7_linha	0.7932
Overall	0.7029

Figure 9: 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:

$$\hat{\beta}_1 = \frac{\sum (X_i - \bar{X}) (Y_i - \bar{Y})}{\sum (X_i - \bar{X})^2}$$

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](#)). Assim, o Índice de Cidades Empreendedoras 2021 é gerado pela variável `ice_final`. O comando `gsort -ice_final` deixa as observações em ordem decrescente.

10 Apêndice

10.0.1 maiores municípios

voltar para [introdução](#)

```
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")
```

10.0.2 maiores municípios

voltar para [introdução](#)

```
library(tidyverse)
library("basedosdados")
library(readxl)

# Ranking de 100 municípios 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 estimativa 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](#)

```

#
#
# 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")

```

10.0.3 Script sd12

voltar para [Tributação](#)

```
#
#
# 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 auxiliar 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_ipitu,-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_ipitu = 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")

```

10.0.4 Script sd13

voltar para [Complexidade Tributária](#)

```
#
# 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")

```


10.0.5 Script sd21

voltar para [Transporte Interurbano](#)

```
#
#
# 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_municipip = str_to_title(nm_municipip)) %>%
  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 Estatísticos.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 munípios

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)

# localizacao 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")

```


10.0.6 Script sd22

voltar para [Condições Urbanas](#)

```
#
#
# sd22 INFRAESTRUTURA - CONDICÕES URBANAS - HOMICÍDIOS

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 municípios
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 - PREÇO 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 faltantes
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")

```

10.0.7 Script sd31

voltar para [Desenvolvimento Econômico](#)

```
#
#
# 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")
```

10.0.8 Script sd32

voltar para [Clientes Potenciais](#)

```
#
#
# 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")

```

10.1 Script i323

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

```

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 orçamentarias 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)

```

10.2 Script i411

Este scrip em R refere-se ao **Indicador Operações de Crédito por Município (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)

```

10.3 Script i412

Este scrip em R refere-se ao **Indicador Proporção Relativa de Capital de Risco (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)

```

10.4 Script i413

Este scrip em R refere-se ao **Indicador Capital Poucado per capita (i413)**

```

library(readxl)
library(tidyverse)
library("basedosdados")

# Indicador Indicador Capital Poucado 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)

```

10.5 Script i511

Este script em R refere-se ao **Indicador Proporção de Mestres e Doutores em C&T (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)

```

10.6 Script i512

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

```

library(readxl)
library(tidyverse)
library("basedosdados")

# Indicador Proporcão 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 CBT
# -----

# 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 Ban
ocupacoes_cet <- read_excel("C:/Users/User/OneDrive/Documentos/ICE 2021/d5 Inova??o/sd51 Inputs/i512/CB
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','2021
,'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', '2
, '3509502', '2111300', '3304904', '2704302', '3301702', '5002704', '2408102', '2211001', '3548708', '3303500', '2
, '3543402', '2607901', '3534401', '3170206', '3552205', '3118601', '2800308', '2910800', '5103403', '4209102', '5
, '1100205', '1500800', '3205002', '4305108', '3303302', '3300456', '1600303', '3301009', '4205407', '3205200', '3
, '3530607', '3106705', '3548500', '4115200', '3513801', '3525904', '1400100', '3143302', '1200401', '2504009', '3
, '5201108', '3201308', '3506003', '3523107', '3551009', '3205309', '2604106', '2303709', '4202404', '3516200', '4
, '4314407', '2933307', '3154606', '3170107', '2610707', '4104808', '3541000', '4125506', '3518701', '3554102', '3
, '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', '2
, '3509502', '2111300', '3304904', '2704302', '3301702', '5002704', '2408102', '2211001', '3548708', '3303500', '2
, '3543402', '2607901', '3534401', '3170206', '3552205', '3118601', '2800308', '2910800', '5103403', '4209102', '5
, '1100205', '1500800', '3205002', '4305108', '3303302', '3300456', '1600303', '3301009', '4205407', '3205200', '3
, '3530607', '3106705', '3548500', '4115200', '3513801', '3525904', '1400100', '3143302', '1200401', '2504009', '3

```

```

, '5201108', '3201308', '3506003', '3523107', '3551009', '3205309', '2604106', '2303709', '4202404', '3516200', '4
, '4314407', '2933307', '3154606', '3170107', '2610707', '4104808', '3541000', '4125506', '3518701', '3554102', '3
, '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)

```

10.7 Script i513

Este script em R refere-se ao **Indicador Média de Investimentos do BNDES e da FINEP (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 função 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)

```

10.8 Script i514

Este script em R refere-se ao **Indicador Infraestrutura Tecnológica (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)

```

10.9 Script i515

Este script em R refere-se ao **Indicador Contratos de Concessão (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)

```

10.10 Script i521

Este script em R refere-se ao **Indicador Patentes (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)

```

10.11 Script i522

Este script em R refere-se ao **Indicador Tamanho da Indústria Inovadora (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','2
','3509502','2111300','3304904','2704302','3301702','5002704','2408102','2211001','3548708','3303500','2
','3543402','2607901','3534401','3170206','3552205','3118601','2800308','2910800','5103403','4209102','5
','1100205','1500800','3205002','4305108','3303302','3300456','1600303','3301009','4205407','3205200','3
','3530607','3106705','3548500','4115200','3513801','3525904','1400100','3143302','1200401','2504009','3
','5201108','3201308','3506003','3523107','3551009','3205309','2604106','2303709','4202404','3516200','4
','4314407','2933307','3154606','3170107','2610707','4104808','3541000','4125506','3518701','3554102','3
','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)

```

10.12 Script i523

Este script em R refere-se ao **Indicador Tamanho da Economia Criativa (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','2
','3509502','2111300','3304904','2704302','3301702','5002704','2408102','2211001','3548708','3303500','2
','3543402','2607901','3534401','3170206','3552205','3118601','2800308','2910800','5103403','4209102','5
','1100205','1500800','3205002','4305108','3303302','3300456','1600303','3301009','4205407','3205200','3
','3530607','3106705','3548500','4115200','3513801','3525904','1400100','3143302','1200401','2504009','3
','5201108','3201308','3506003','3523107','3551009','3205309','2604106','2303709','4202404','3516200','4
','4314407','2933307','3154606','3170107','2610707','4104808','3541000','4125506','3518701','3554102','3
','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)

```

10.13 Script i524

Este script em R refere-se ao **Indicador Tamanho das Empresas TIC (i514)**

```

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 função 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','2
','3509502','2111300','3304904','2704302','3301702','5002704','2408102','2211001','3548708','3303500','2
','3543402','2607901','3534401','3170206','3552205','3118601','2800308','2910800','5103403','4209102','5
','1100205','1500800','3205002','4305108','3303302','3300456','1600303','3301009','4205407','3205200','3
','3530607','3106705','3548500','4115200','3513801','3525904','1400100','3143302','1200401','2504009','3
','5201108','3201308','3506003','3523107','3551009','3205309','2604106','2303709','4202404','3516200','4
','4314407','2933307','3154606','3170107','2610707','4104808','3541000','4125506','3518701','3554102','3
','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)

```

10.14 Script i611

Este script em R refere-se ao **Indicador Nota do Ideb (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)

```

10.15 Script i612

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

```

library(tidyverse)

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

# -----
# PASSO 0: preliminares
# -----

# definimos uma função para desvio padrão 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)

```

10.16 Script i613

Este script em R refere-se ao **Indicador Taxa Líquida de Matrícula no Ensino Médio (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','2
,'3509502','2111300','3304904','2704302','3301702','5002704','2408102','2211001','3548708','3303500','2
,'3543402','2607901','3534401','3170206','3552205','3118601','2800308','2910800','5103403','4209102','5
,'1100205','1500800','3205002','4305108','3303302','3300456','1600303','3301009','4205407','3205200','3
,'3530607','3106705','3548500','4115200','3513801','3525904','1400100','3143302','1200401','2504009','3
,'5201108','3201308','3506003','3523107','3551009','3205309','2604106','2303709','4202404','3516200','4
,'4314407','2933307','3154606','3170107','2610707','4104808','3541000','4125506','3518701','3554102','3
,'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 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)

```

10.17 Script i614

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

```

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 dados ENEM dos municipios 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, ciencias 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)

```

10.18 Script i615

Este script em R refere-se ao **Indicador Proporção de Matriculados no Ensino Técnico e Profissionalizante (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`
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','2
,'3509502','2111300','3304904','2704302','3301702','5002704','2408102','2211001','3548708','3303500','2
,'3543402','2607901','3534401','3170206','3552205','3118601','2800308','2910800','5103403','4209102','5
,'1100205','1500800','3205002','4305108','3303302','3300456','1600303','3301009','4205407','3205200','3
,'3530607','3106705','3548500','4115200','3513801','3525904','1400100','3143302','1200401','2504009','3
,'5201108','3201308','3506003','3523107','3551009','3205309','2604106','2303709','4202404','3516200','4
,'4314407','2933307','3154606','3170107','2610707','4104808','3541000','4125506','3518701','3554102','3
,'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)

```

10.19 Script i621

Este script em R refere-se ao **Indicador Proporção de Adultos com pelo Menos Ensino Superior Completo (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 dados ENEM dos municipios 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, inscritos 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))/sd(i621))

# exportamos o indicador
write.csv(i621, "i621.csv", row.names = FALSE)

```

10.20 Script i622

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

```
library(data.table)
library(readxl)
library(tidyverse)

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

# -----
# PASSO 0: preliminares
# -----

# definimos uma função 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 são 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)

```

10.21 Script d7

Este script em R refere-se aos **Indicadores do determinante Cultura (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_empresa <-
  read.csv(
    "C:/Users/User/OneDrive/Documentos/ICE 2021/
      d7 Determinante Cultura/empresa.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)

```

10.22 Padronização

Voltar para [Padronização](#)

```

*** 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 {
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
}

```

10.23 Análise de Componentes Principais

Voltar para [Análise de Componentes Principais](#)

```

**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

```