



Projeto Final

Grupo 4

Engenharia de
Dados



Custo de vida nas cidades de São Paulo x Rio de Janeiro

GRUPO



Allan Farias

LinkedIn:
<https://www.linkedin.com/in/allanfarias1>



Yago Sousa

LinkedIn:
www.linkedin.com/in/yago-sousa-ff



Alyne Berriel

LinkedIn:
www.linkedin.com/in/alyne-berriel



Enzo Lourenço

LinkedIn:
<https://www.linkedin.com/in/enzolourenco/>

Custo de vida nas cidades de São Paulo x Rio de Janeiro



Custo de vida nas cidades de São Paulo x Rio de Janeiro

TECNOLOGIAS E METODOLOGIAS UTILIZADAS

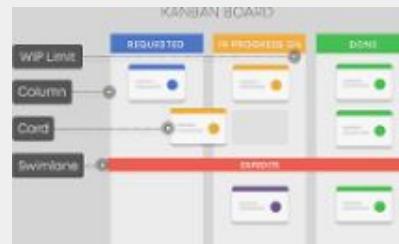
- SQL
- Python
- Pandas
- Pandera
- Pymongo
- Numpy
- PySpark



- Mongo Atlas

**Google Cloud**

- Cloud Storage
- Cloud SQL - MySQL
- VPC Network
- Dataproc
- BigQuery
- Billing
- Trello
- Meet
- Discord
- Metodologia Ágil Kanban



Custo de vida nas cidades de São Paulo x Rio de Janeiro

DATASETS

Combustível

ANP (Agência Nacional do Petróleo)**Quantidade: 22 datasets****Tamanho: 1,91GB****Linhas: 11.560.909****Formato: .CSV**

GLP P13

ANP (Agência Nacional do Petróleo)**Quantidade: 22 datasets****Tamanho: 500MB****Linhas: 3.263.485****Formato: .CSV**

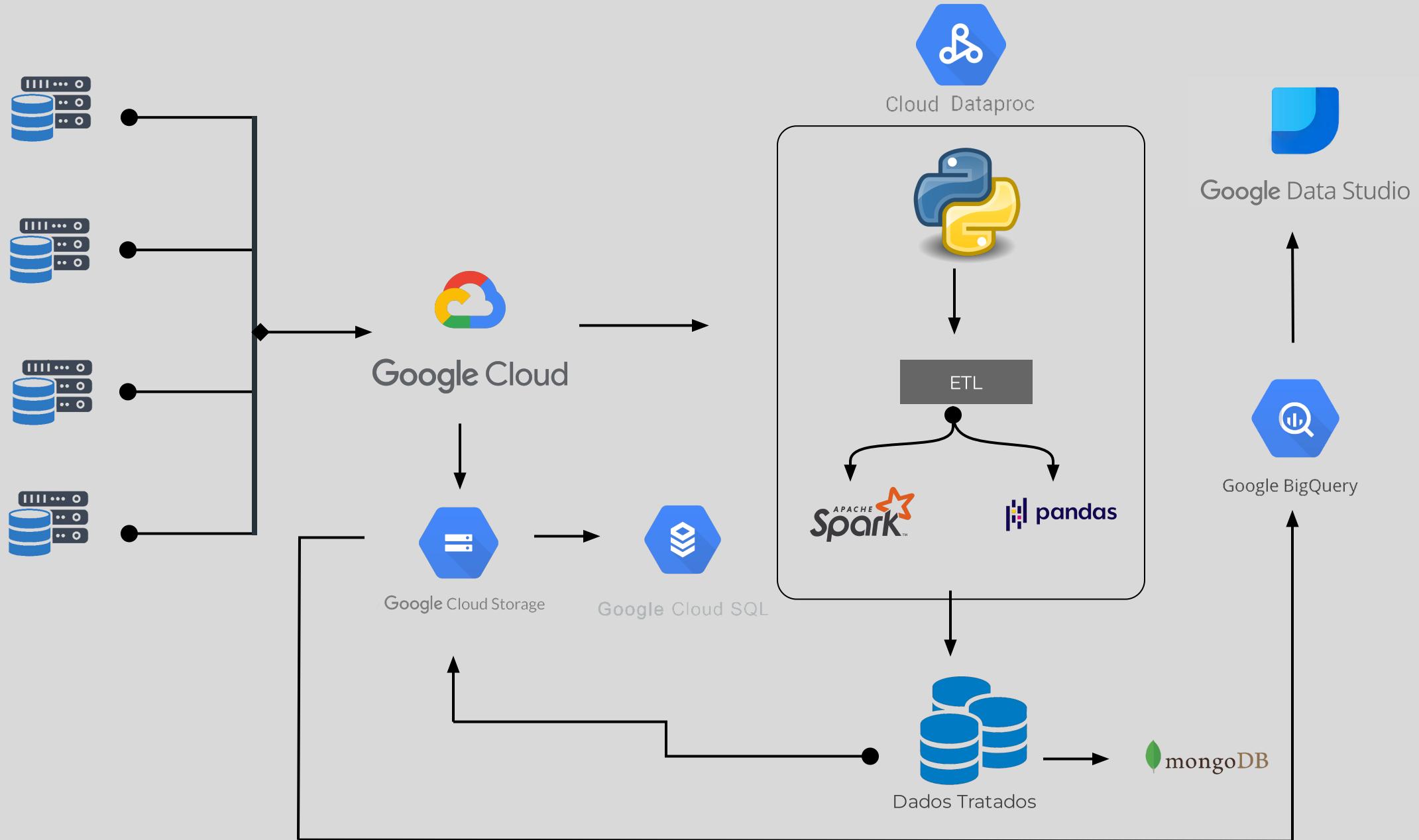
Salário Mínimo

IPEADATA**Tamanho: 24kb****Linhas: 991****Formato: .CSV**

Cesta Básica

DIEESE (Departamento Intersindical de Estatística e Estudos Socioeconômicos)**Tamanho: 64kb****Linhas: 138****Formato: .XLS**

WORKFLOW



Custo de vida nas cidades de São Paulo x Rio de Janeiro

CONCATENAR DATASETS - PANDAS

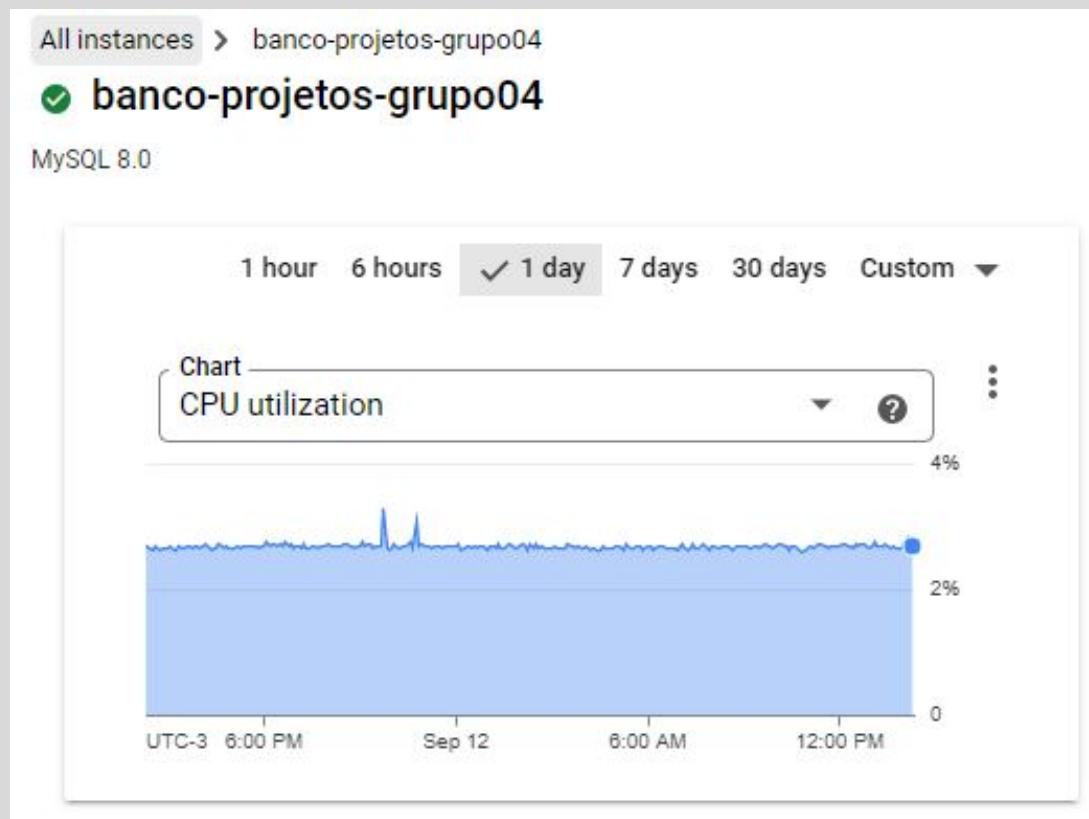
```
#EXTRAINDO OS DATASETS
df1 = pd.read_csv('gs://projeto_final_grupo4/Originais/combustivel/ca-2011-01.csv', sep=';')
df2 = pd.read_csv('gs://projeto_final_grupo4/Originais/combustivel/ca-2011-02.csv', sep=';')
df3 = pd.read_csv('gs://projeto_final_grupo4/Originais/combustivel/ca-2012-01.csv', sep=';')
df4 = pd.read_csv('gs://projeto_final_grupo4/Originais/combustivel/ca-2012-02.csv', sep=';')
df5 = pd.read_csv('gs://projeto_final_grupo4/Originais/combustivel/ca-2013-01.csv', sep=';')
df6 = pd.read_csv('gs://projeto_final_grupo4/Originais/combustivel/ca-2013-02.csv', sep=';')
df7 = pd.read_csv('gs://projeto_final_grupo4/Originais/combustivel/ca-2014-01.csv', sep=';')
df8 = pd.read_csv('gs://projeto_final_grupo4/Originais/combustivel/ca-2014-02.csv', sep=';')
df9 = pd.read_csv('gs://projeto_final_grupo4/Originais/combustivel/ca-2015-01.csv', sep=';')
df10 = pd.read_csv('gs://projeto_final_grupo4/Originais/combustivel/ca-2015-02.csv', sep=';')
df11 = pd.read_csv('gs://projeto_final_grupo4/Originais/combustivel/ca-2016-01.csv', sep=';')
df12 = pd.read_csv('gs://projeto_final_grupo4/Originais/combustivel/ca-2016-02.csv', sep=';')
df13 = pd.read_csv('gs://projeto_final_grupo4/Originais/combustivel/ca-2017-01.csv', sep=';')
df14 = pd.read_csv('gs://projeto_final_grupo4/Originais/combustivel/ca-2017-02.csv', sep=';')
df15 = pd.read_csv('gs://projeto_final_grupo4/Originais/combustivel/ca-2018-01.csv', sep=';')
df16 = pd.read_csv('gs://projeto_final_grupo4/Originais/combustivel/ca-2018-02.csv', sep=';')
df17 = pd.read_csv('gs://projeto_final_grupo4/Originais/combustivel/ca-2019-01.csv', sep=';')
df18 = pd.read_csv('gs://projeto_final_grupo4/Originais/combustivel/ca-2019-02.csv', sep=';')
df19 = pd.read_csv('gs://projeto_final_grupo4/Originais/combustivel/ca-2020-01.csv', sep=';')
df20 = pd.read_csv('gs://projeto_final_grupo4/Originais/combustivel/ca-2020-02.csv', sep=';')
df21 = pd.read_csv('gs://projeto_final_grupo4/Originais/combustivel/ca-2021-01.csv', sep=';')
df22 = pd.read_csv('gs://projeto_final_grupo4/Originais/combustivel/ca-2021-02.csv', sep=';', encoding='ISO-8859-1')

#CONCATENANDO TODOS OS DATAFRAMES EM UM SÓ
dfc = pd.concat([df1, df2, df3, df4, df5, df6, df7, df8, df9, df10, df11, df12, df13, df14, df15, df16, df17, df18, df19, df20, df21, df22])

#ENVIANDO ARQUIVO ORIGINAL PARA O BUCKET E CONVERTENDO O SEPARADOR ';' para ',' POR CAUSA DO MYSQL
dfc.to_csv('gs://projeto_final_grupo4/Mysql/combustivel.csv', sep=',', index=False)
```

Custo de vida nas cidades de São Paulo x Rio de Janeiro

DATASET ORIGINAL - MYSQL



CLOUD SHELL

Terminal (projeto-final-soulcode-grupo-4) +

```
mysql> show databases;
+-----+
| Database |
+-----+
| cestabasica
| combustivel
| gas
| information_schema
| mysql
| performance_schema
| salario
| sys
+-----+
8 rows in set (0.12 sec)

mysql> USE combustivel; show tables;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
+-----+
| Tables_in_combustivel |
+-----+
| original_combustivel |
+-----+
1 row in set (0.11 sec)

mysql> describe original_combustivel;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| regiao_sigla | varchar(10) | YES | | NULL |
| estado_sigla | varchar(10) | YES | | NULL |
| municipio | varchar(20) | YES | | NULL |
| revenda | varchar(30) | YES | | NULL |
| cnpj_da_Revenda | varchar(30) | YES | | NULL |
| nome_da_rua | varchar(30) | YES | | NULL |
| numero_rua | varchar(30) | YES | | NULL |
| complemento | varchar(30) | YES | | NULL |
| bairro | varchar(30) | YES | | NULL |
| cep | varchar(20) | YES | | NULL |
| produto | varchar(30) | YES | | NULL |
| data_da_coleta | varchar(10) | YES | | NULL |
| valor_de_venda | varchar(10) | YES | | NULL |
+-----+-----+-----+-----+-----+-----+
```

Custo de vida nas cidades de São Paulo x Rio de Janeiro

PROCEDIMENTOS DE ETL - DATAPROC

Google Cloud | Projeto-final-soulcode-grupo-4

Create a Dataproc cluster on Compute Engine

Jobs on clusters

- Clusters** (selected)
- Jobs
- Workflows
- Autoscaling policies

Serverless

- Batches

Metastore Services

- Metastore
- Federation

Utilities

- Component exchange
- Workbench

Release Notes

Set up cluster

Begin by providing basic information.

Name
Cluster Name * projeto-final-grupo4

Location
Region * southamerica-east1 Zone * southamerica-east1-a

Cluster type

- Standard (1 master, N workers)
- Single Node (1 master, 0 workers)
- High Availability (3 masters, N workers)

Autoscaling

Automates cluster resource management based on an autoscaling policy.

Policy None

Enhanced Flexibility Mode

Dataproc Enhanced Flexibility Mode (EFM) manages shuffle data to minimize job progress delays caused by the removal of nodes from a running cluster. EFM offloads shuffle data in one of two user-selectable modes, primary worker shuffle and Hadoop Compatible File System (HCFS) shuffle. [Learn more](#)

Versioning

An autoscaling policy must be selected to configure EFM.

CREATE CANCEL

EQUIVALENT COMMAND LINE

Google Cloud | Projeto-final-soulcode-grupo-4

Criar um cluster do Dataproc no Compute Engine

Configuração de rede

Estabelece a conectividade para as instâncias de VM neste cluster.

Redes neste projeto (selected)

Redes compartilhadas do projeto host: Choose a shared VPC network from project that is different from this cluster's project. [Learn more](#)

Rede principal vpc-projeto-final **Sub-rede** vpc-projeto-final

Tags de rede

Network tags are text attributes you can add to make firewall rules and routes applicable to specific VM instances.

Apenas IP interno

Configure todas as instâncias para que tenham apenas endereços IP internos. [Learn more](#)

Metastore do Dataproc

Configure o Dataproc para que use o respectivo metastore como seu metastore Hive. [Saiba mais](#)

Serviço de metastore Nenhum

Recomendamos esta opção para manter os metadados de tabela quando um cluster for

CRIAR CANCELAR

LINHA DE COMANDO EQUIVALENTE

PROCEDIMENTOS DE ETL - PYSPARK

CORREÇÃO DA COLUNA 'data' NO DATAFRAME DE SALÁRIO MÍNIMO



```
#CRIANDO UMA COLUNA DE 'dia' PARA COLOCAR 'data' E COLOCANDO O VALOR 1
df_salariomin = df_salariomin.withColumn('dia', lit(1))

#ALTERANDO O VALOR '1' POR '01'
df_salariomin = df_salariomin.withColumn('dia', regexp_replace('dia', '1', '01'))

#CONCATENANDO A COLUNA 'dia' COM A COLUNA 'data'
df_salariomin = df_salariomin.withColumn('data', F.concat_ws('.', F.col('data'), F.col('dia')))

#DROPANDO DADOS ANTERIORES A 2010
df_salariomin = df_salariomin.filter(F.col('data') > '2010.01')

#ALTERANDO AS DATAS COM MÊS 10 QUE CONSTAM COMO .1.
df_salariomin = df_salariomin.withColumn('data', regexp_replace('data', '2010.1.01', '2010.10.01'))
df_salariomin = df_salariomin.withColumn('data', regexp_replace('data', '2011.1.01', '2011.10.01'))
df_salariomin = df_salariomin.withColumn('data', regexp_replace('data', '2012.1.01', '2012.10.01'))
df_salariomin = df_salariomin.withColumn('data', regexp_replace('data', '2013.1.01', '2013.10.01'))
df_salariomin = df_salariomin.withColumn('data', regexp_replace('data', '2014.1.01', '2014.10.01'))
df_salariomin = df_salariomin.withColumn('data', regexp_replace('data', '2015.1.01', '2015.10.01'))
df_salariomin = df_salariomin.withColumn('data', regexp_replace('data', '2016.1.01', '2016.10.01'))
df_salariomin = df_salariomin.withColumn('data', regexp_replace('data', '2017.1.01', '2017.10.01'))
df_salariomin = df_salariomin.withColumn('data', regexp_replace('data', '2018.1.01', '2018.10.01'))
df_salariomin = df_salariomin.withColumn('data', regexp_replace('data', '2019.1.01', '2019.10.01'))
df_salariomin = df_salariomin.withColumn('data', regexp_replace('data', '2020.1.01', '2020.10.01'))
df_salariomin = df_salariomin.withColumn('data', regexp_replace('data', '2021.1.01', '2021.10.01'))
df_salariomin = df_salariomin.withColumn('data', regexp_replace('data', '2022.1.01', '2022.10.01'))
```

PROCEDIMENTOS DE ETL - PYSPARK

CORREÇÃO DA COLUNA 'data' NO DATAFRAME DE SALÁRIO MÍNIMO

```
#ALTERANDO O '.' POR '-' NA COLUNA 'data'  
df_salariomin = df_salariomin.withColumn('data', regexp_replace('data', '\\.', '-'))  
  
#DROPANDO A COLUNA 'dia'  
df_salariomin = df_salariomin.drop('dia')  
  
#CONVERTENDO O TIPO DE DADO DA COLUNA 'data' PARA date E COLUNA 'salario_minimo' PARA float  
df_salariomin = df_salariomin.withColumn('data', F.to_date('data', 'yyyy-MM-dd'))  
df_salariomin = df_salariomin.withColumn("salario_minimo", df_salariomin["salario_minimo"].cast('float'))  
  
#CONVERTENDO O DATASET PARA O PANDAS  
df_salariop = df_salariomin.toPandas()  
  
#ALTERANDO O TIPO DE COLUNA 'data' PARA date NOVAMENTE, POIS A CONVERSÃO ALTEROU PARA STRING  
df_salariop['data'] = pd.to_datetime(df_salariop['data'], format='%Y-%m-%d')
```

Custo de vida nas cidades de São Paulo x Rio de Janeiro

PROCEDIMENTOS DE ETL - PLOTAGEM PANDAS

Plot de combustível

```
dfplot = dfc.loc[(dfc['data'] > '2010-12-31')]
dfplot = dfplot.groupby('estado').agg('mean').reset_index()
```

```
dfplot = dfplot.sort_values('valor', ascending=False)
plt.figure(figsize=(15, 5))
plt.bar('estado', 'valor', data=dfplot)
plt.xlabel('Estado', size = 15)
plt.ylabel('Preço em R$', size = 15)
plt.title('Média do preço dos combustíveis nos últimos 10 anos por estado', size = 20)
```

```
Text(0.5, 1.0, 'Média do preço dos combustíveis nos últimos 10 anos por estado')
```



Custo de vida nas cidades de São Paulo x Rio de Janeiro

DATASET TRATADOS - MONGO

DATASET COMBUSTIVEL - dfc - client

```
[ ] #CRIAR AS COLEÇÕES DO DB
db = client ['Tratado']
combustivel = db.combustivel

[ ] #ENVIANDO PARA O MONGO
df_dict = dfc.to_dict('records')
combustivel.insert_many(df_dict)
```

DATASET GLPP13 - dfg - client2

```
[ ] #CRIAR AS COLEÇÕES DO DB2
db2 = client2 ['Tratados']
glpp13 = db2.glpp13

[ ] #ENVIANDO PARA O MONGO
df_dict2 = dfg.to_dict('records')
glpp13.insert_many(df_dict2)
```

The screenshot shows a user interface for a project management system. At the top, there is a header with a logo, the text "Projeto Final...", a gear icon, "Access Manager", and "Billing". Below the header is a sidebar on the left with the following menu items:

- ORGANIZATION
- Projects** (highlighted in green)
- Alerts (0)
- Activity Feed
- Settings
- Access Manager
- Billing
- Support

The main content area on the right is titled "PROJETO FINAL - GRUPO 4" and displays a list of projects:

- Projects
- Find a project... (with a search icon)
- Project Name
- Custo de Vida 2
- Custo de vida

Custo de vida nas cidades de São Paulo x Rio de Janeiro

DATASET TRATADOS - MONGO

PROJETO FINAL - GRUPO 4 > CUSTO DE VIDA > DATABASES

Cluster0

VERSION 5.0.12 REGION GCP Sao Paulo (southamerica-east1)

Overview Real Time Metrics Collections Search Profiler Performance Advisor Online Archive Cmd Line Tools

DATABASES: 1 COLLECTIONS: 1

+ Create Database

Search Namespaces

Tratado

combustivel

Tratado.combustivel

STORAGE SIZE: 114.25MB LOGICAL DATA SIZE: 408.23MB TOTAL DOCUMENTS: 3900000 INDEXES TOTAL SIZE: 115.71MB

Find Indexes Schema Anti-Patterns 0 Aggregation Search Indexes ●

INSERT DOCUMENT

FILTER { field: 'value' }

PROJETO FINAL - GRUPO 4 > CUSTO DE VIDA 2 > DATABASES

Cluster

VERSION 5.0.12 REGION AWS Sao Paulo (sa-east-1)

Overview Real Time Metrics Collections Search Profiler Performance Advisor Online Archive Cmd Line Tools

DATABASES: 1 COLLECTIONS: 3

+ Create Database

Search Namespaces

Tratados

cestabasica

glpp13

salario

Tratados.cestabasica

STORAGE SIZE: 40KB LOGICAL DATA SIZE: 73.9KB TOTAL DOCUMENTS: 133 INDEXES TOTAL SIZE: 20KB

Find Indexes Schema Anti-Patterns 0 Aggregation Search Indexes ●

INSERT DOCUMENT

FILTER { field: 'value' }

Custo de vida nas cidades de São Paulo x Rio de Janeiro

DATASET TRATADOS - BIGQUERY

The screenshot shows the BigQuery web interface. On the left, the Explorer panel displays a tree view of datasets and tables. A table named 'Gas-RJ-SP' is selected and highlighted in blue. The main area shows a query editor with the following SQL code:

```
1 --Cidades de SP e RJ preço do gas
2 select estado,
3 produto,
4 data,
5 VALOR
6 | from `projeto-final-soulcode-grupo-4.dataset_tratado.gas_tratado`
7 where estado in ('RJ', 'SP')
8 order by 1 desc
9 |
```

The query editor includes standard navigation and action buttons: RUN, SAVE, SHARE, SCHEDULE, and MORE.

BIGQUERY - PROCEDIMENTOS DE ETL

Análise de dados em SQL no BigQuery



```
SELECT
  Salario_minimo,
  extract(YEAR FROM data) AS ANO
  FROM projeto-final-soulcode-grupo-4.dataset_tratado.salario_tratado
  group by Salario_minimo, ANO
  order by 1 desc
```

BIGQUERY - PROCEDIMENTOS DE ETL

Análise de dados em SQL no BigQuery



```
--Cidades de SP e RJ media de PREÇO do combustivel
select
estado,
cidade,
produto,
count(cidade) AS TOTAL,
extract(MONTH FROM data) AS MES,
extract(YEAR FROM data) AS ANO,
(SUM(VALOR)/COUNT(*)) AS VALOR
    from projeto-final-soulcode-grupo-4.dataset_tratado.combustivel_tratado
where estado in ('RJ', 'SP')
group by estado, cidade, produto , ANO, MES, ANO
order by CIDADE,ANO desc
```

BIGQUERY - PROCEDIMENTOS DE ETL

Análise de dados em SQL no BigQuery



```
select mes_referencia, cidade, valor_cesta
  from
    (select mes_referencia, brasilia as Brasilia, campo_grande as Campo_Grande, cuiaba as
Cuiaba, goiania as Goiania, belo_horizonte as Belo_Horizonte, rio_de_janeiro as
Rio_de_Janeiro, sao_paulo as Sao_Paulo, vitoria as Vitoria, curitiba as Curitiba,
florianopolis as Florianopolis, porto_alegre as Porto_Alegre, belem as Belem,
boa_vista as Boa_Vista, macapa as Macapa, manaus as Manaus, palmas as Palmas,
porto_velho as Porto_Velho, rio_branco as Rio_Branco, aracaju as Aracaju, fortaleza as
Fortaleza, joao_pessoa as Joao_Pessoa, maceio as Maceio, natal as Natal, recife as Recife,
salvador as Salvador, sao_luis as Sao_Luis, teresina as Teresina, macae as Macae
    from projeto-final-soulcode-grupo-4.dataset_tratado.cestabasta_tratado) p
UNPIVOT
  (valor_cesta FOR cidade IN (Brasilia, Campo_Grande, Cuiaba, Goiania, Belo_Horizonte,
                               Rio_de_Janeiro, Sao_Paulo, Vitoria, Curitiba, Florianopolis,
                               Porto_Alegre, Belem, Boa_Vista, Macapa, Manaus, Palmas,
                               Porto_Velho, Rio_Branco, Aracaju, Fortaleza, Joao_Pessoa, Maceio,
                               Natal, Recife, Salvador, Sao_Luis, Teresina, Macae)) AS
UNPIVOT_TB_CESTA
```

Custo de vida nas cidades de São Paulo x Rio de Janeiro

COMBUSTÍVEL



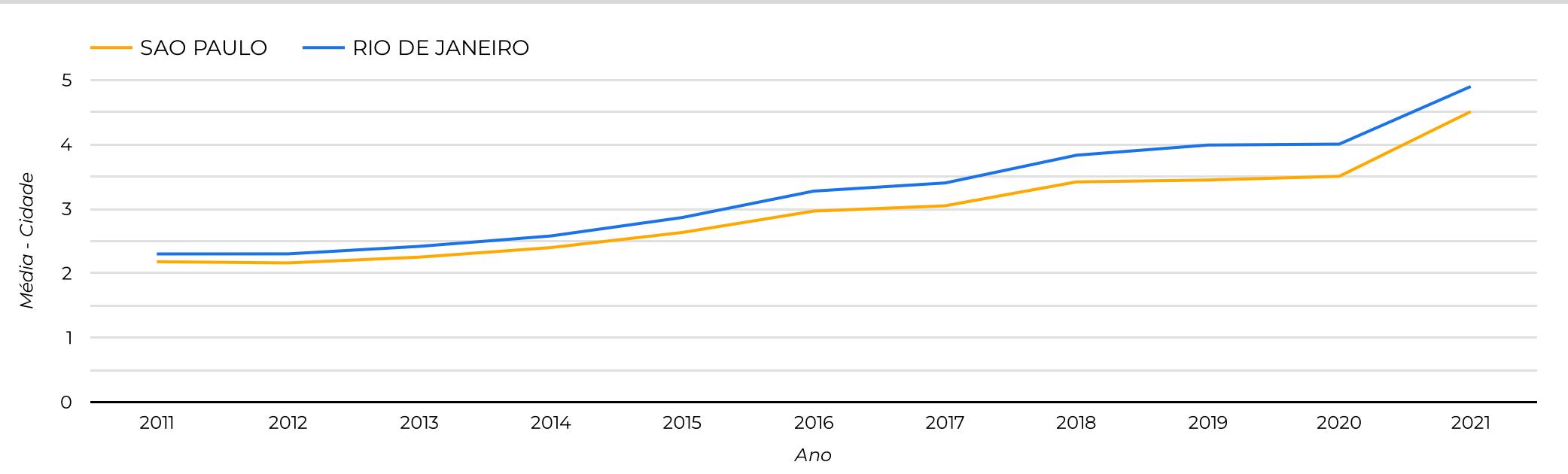
Custo de vida nas cidades de São Paulo x Rio de Janeiro

COMBUSTÍVEL - RJ x SP

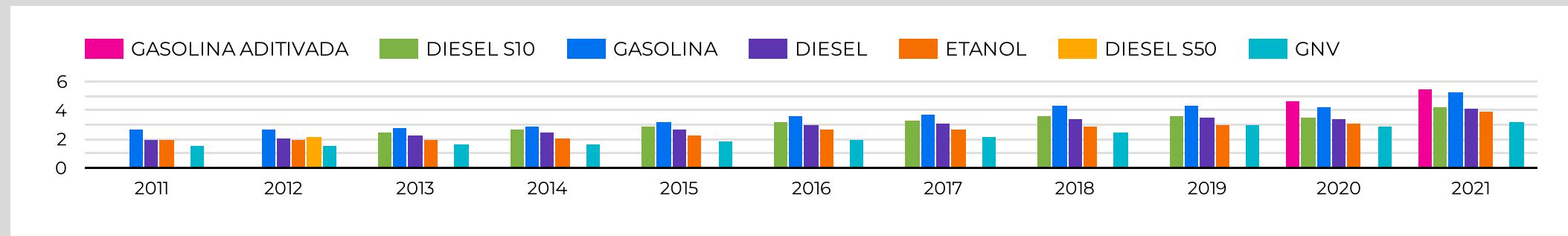
Produto

Cidade: RIO D... (2)

Comparação do valor do combustível entre as cidades de Rio de Janeiro e São Paulo



Média de cada tipo de combustível por ano



Custo de vida nas cidades de São Paulo x Rio de Janeiro

GLP P13



Custo de vida nas cidades de São Paulo x Rio de Janeiro

GLP P13

Estado:

Cidade:

Valor Mínimo

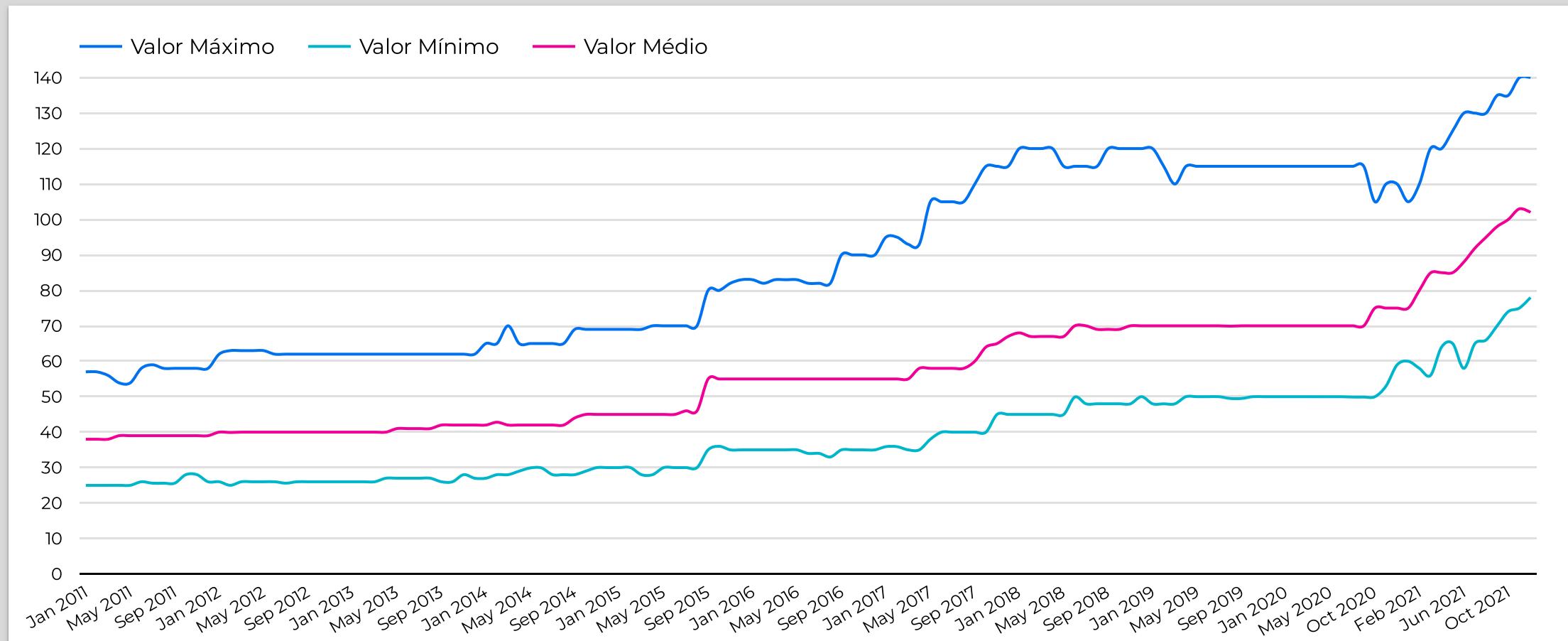
R\$ 24.99

Valor Médio

R\$ 46.00

Valor Máximo

R\$ 140.00

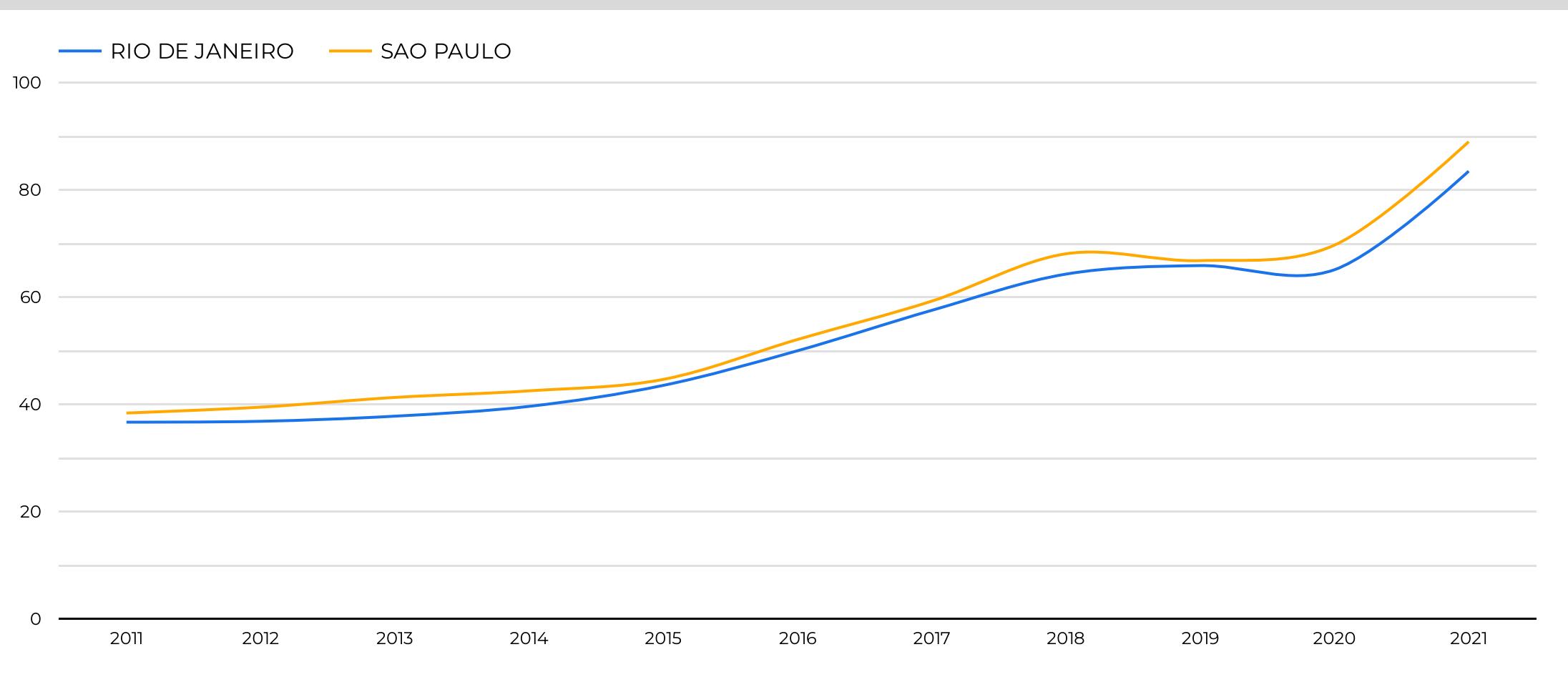
Fonte: <https://www.gov.br/anp/pt-br/centrais-de-conteudo/dados-abertos/serie-historica-de-precos-de-combustiveis>

Custo de vida nas cidades de São Paulo x Rio de Janeiro

GLP P13 - RJ x SP

cidade: RIO DE JANEIRO, SAO ... (2) ▾

Comparação do valor do gás de cozinha entre as cidades de Rio de Janeiro e São Paulo



Custo de vida nas cidades de São Paulo x Rio de Janeiro

CESTA BÁSICA E SALÁRIO MÍNIMO



Custo de vida nas cidades de São Paulo x Rio de Janeiro

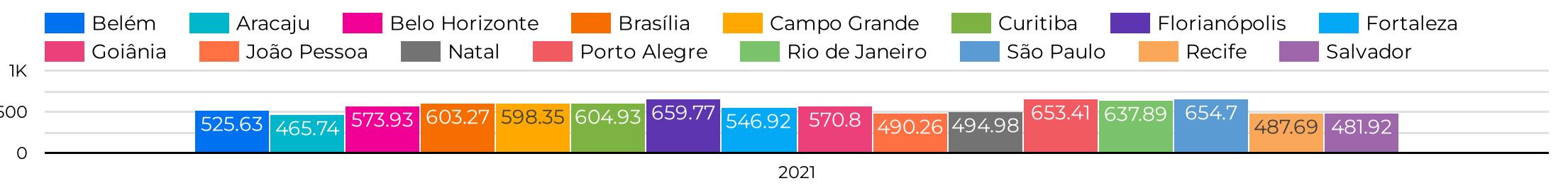
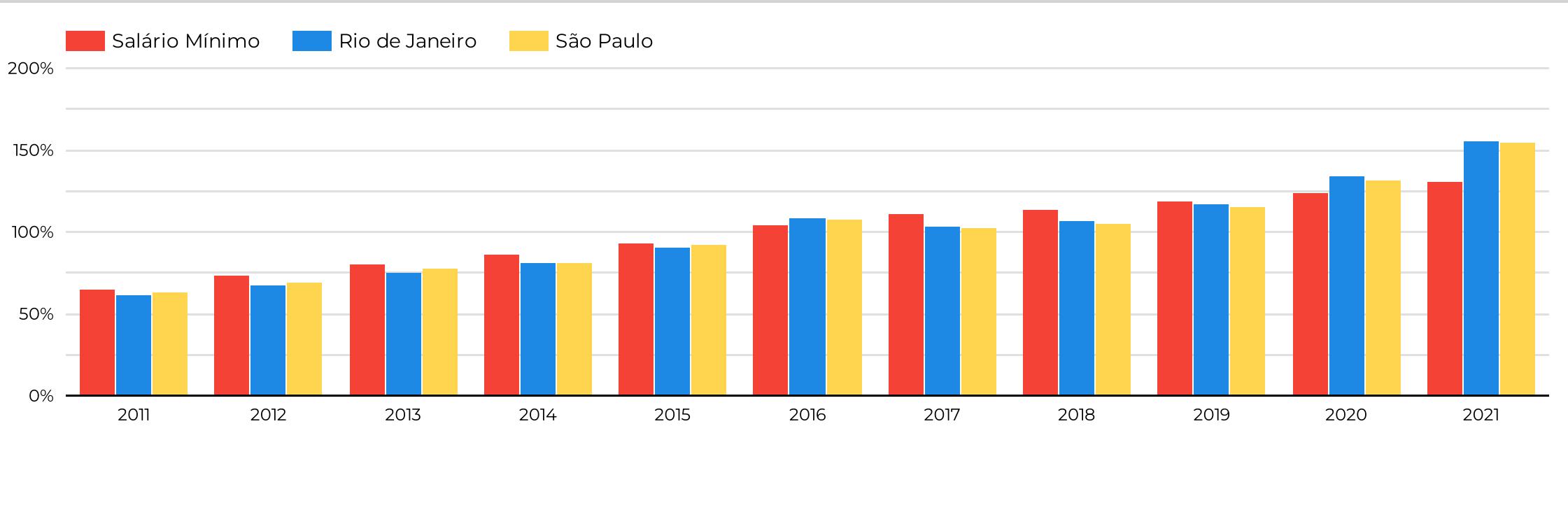
CORRELAÇÃO ENTRE SALÁRIO MÍNIMO / CESTA BÁSICA

Jan 1, 2011 - Dec 31, 2021

Salário Mínimo
R\$ 842.29
▲ 65.2%

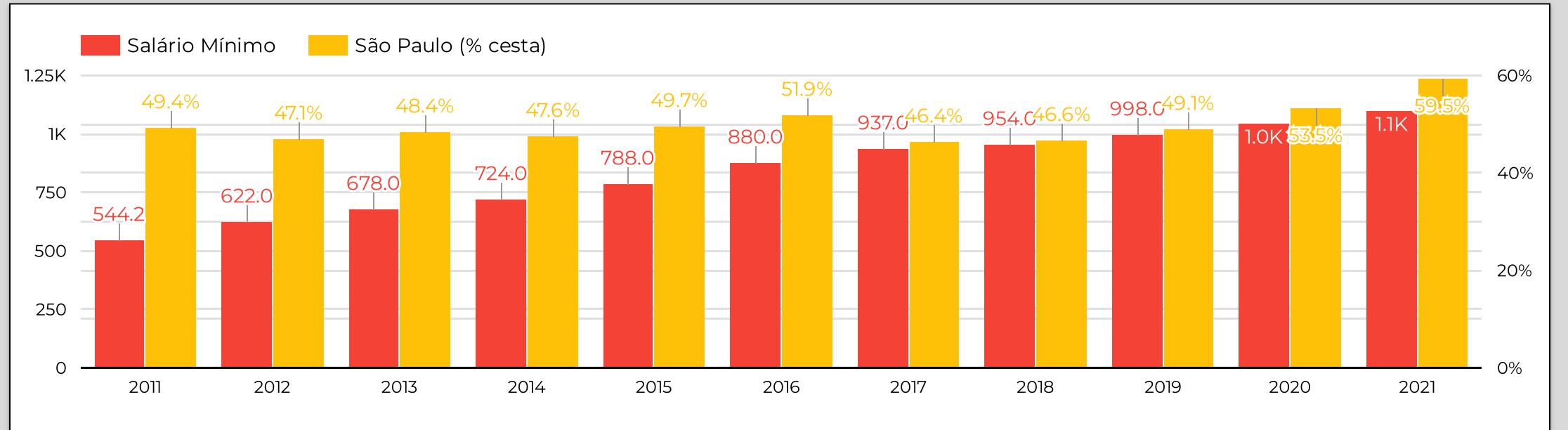
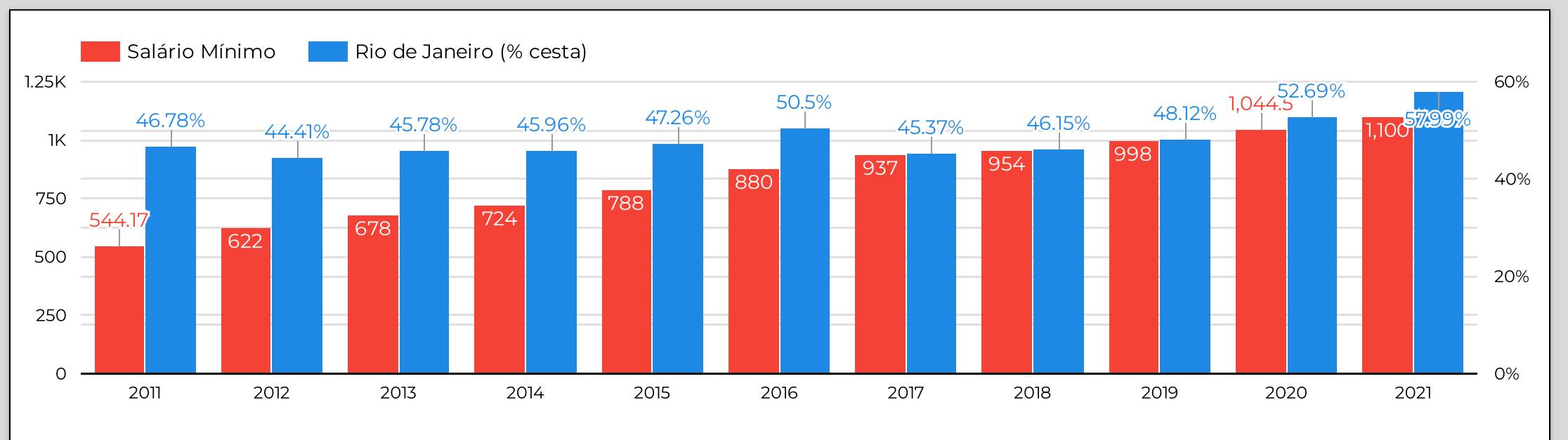
Rio de Janeiro
R\$ 411.03
No data

São Paulo
R\$ 423.88
No data



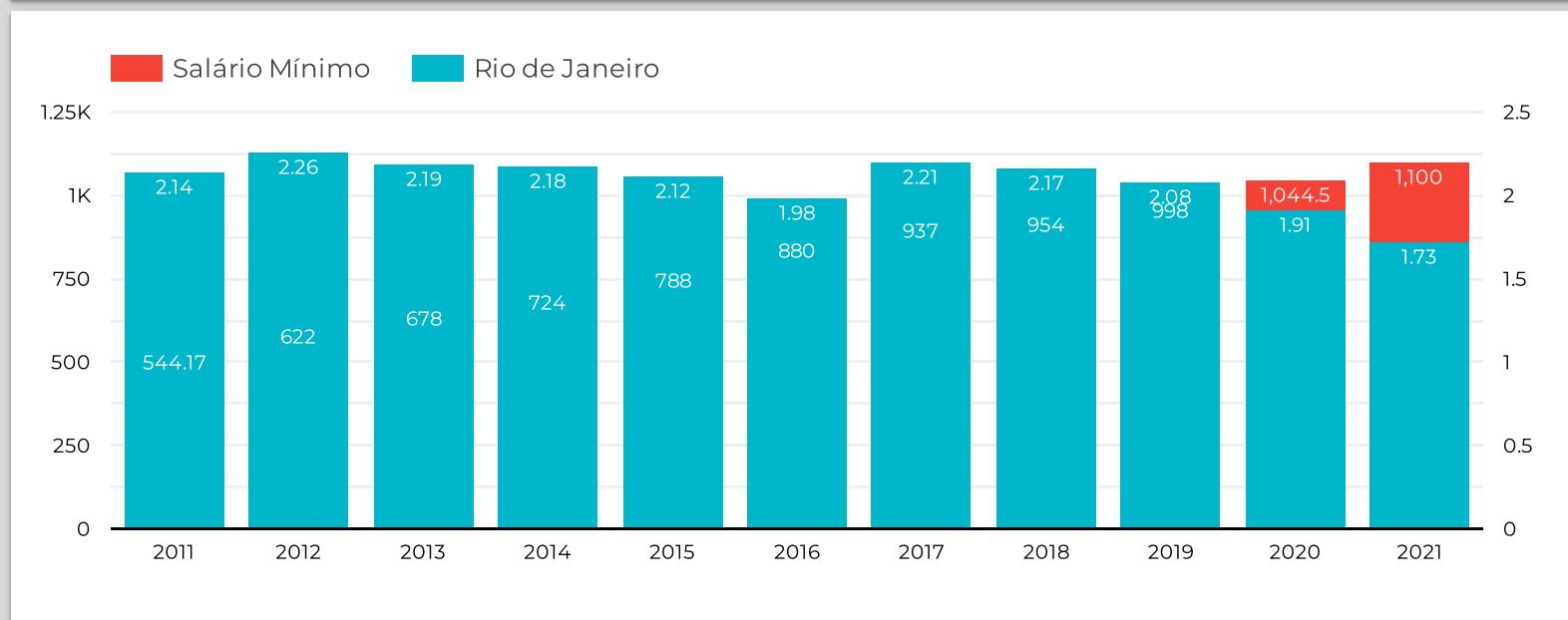
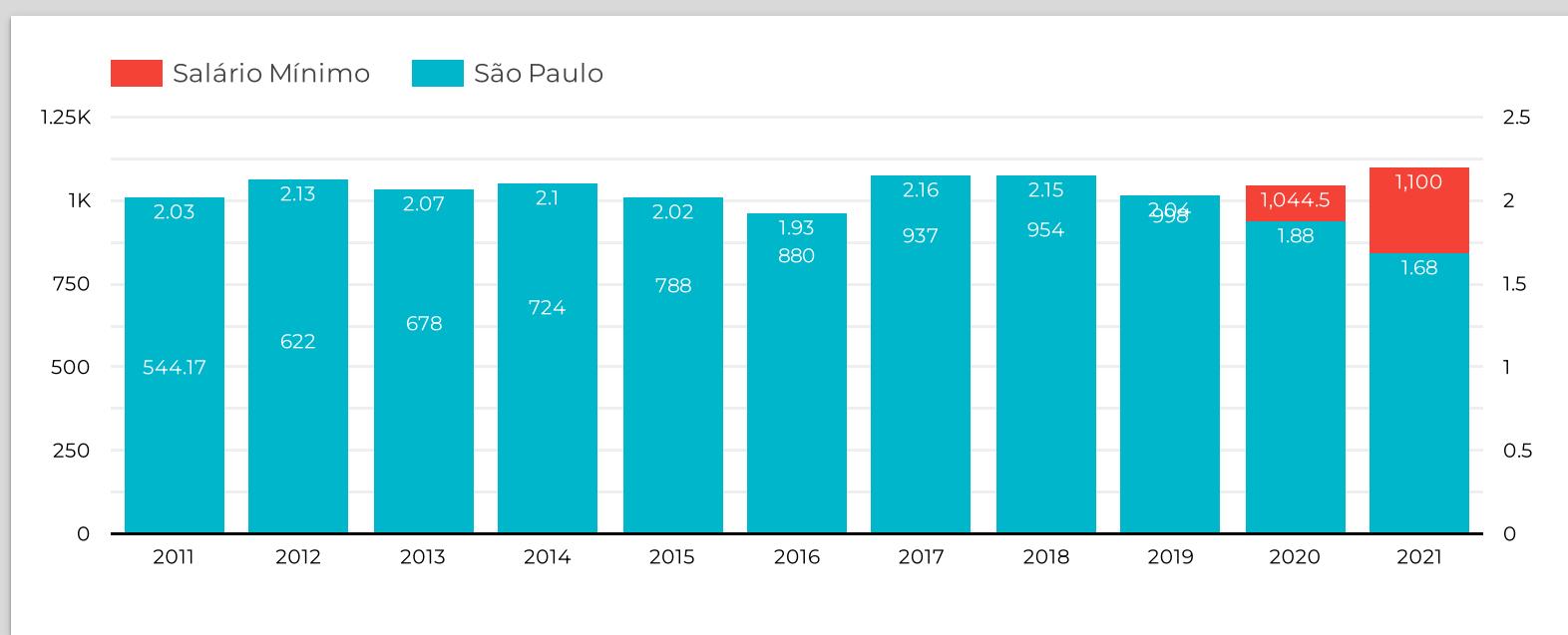
Custo de vida nas cidades de São Paulo x Rio de Janeiro

CRESCIMENTO PERCENTUAL DA CESTA BÁSICA PERANTE O VALOR DO SALÁRIO MÍNIMO



Custo de vida nas cidades de São Paulo x Rio de Janeiro

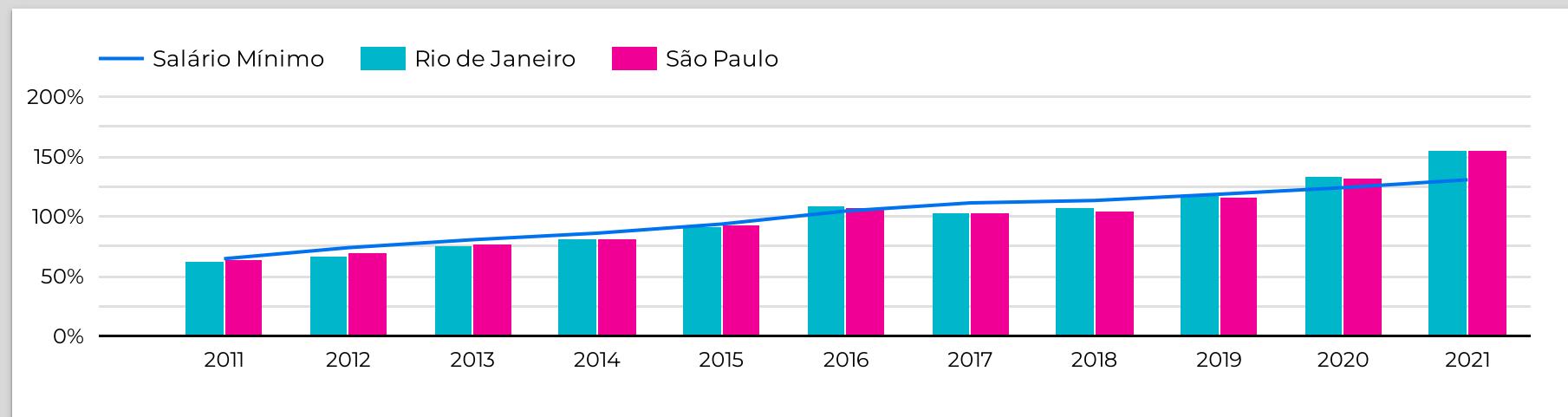
QUANTIDADE DE CESTA BÁSICA POR SALÁRIO MÍNIMO



Custo de vida nas cidades de São Paulo x Rio de Janeiro

CRESCIMENTO PERCENTUAL DA CESTA BÁSICA PERANTE O VALOR DO SALÁRIO MÍNIMO

Ano	Salário Mínimo	Rio de Janeiro %	Quantidade	São Paulo %	Quantidade
1. 2021	R\$ 1,100	57.99%	1.73	59.52%	1.68
2. 2020	R\$ 1,044.5	52.69%	1.91	53.45%	1.88
3. 2019	R\$ 998	48.12%	2.08	49.13%	2.04
4. 2018	R\$ 954	46.15%	2.17	46.59%	2.15
5. 2017	R\$ 937	45.37%	2.21	46.41%	2.16
6. 2016	R\$ 880	50.5%	1.98	51.87%	1.93
7. 2015	R\$ 788	47.26%	2.12	49.67%	2.02
8. 2014	R\$ 724	45.96%	2.18	47.64%	2.1
9. 2013	R\$ 678	45.78%	2.19	48.44%	2.07
10. 2012	R\$ 622	44.41%	2.26	47.08%	2.13
11. 2011	R\$ 544.17	46.78%	2.14	49.35%	2.03

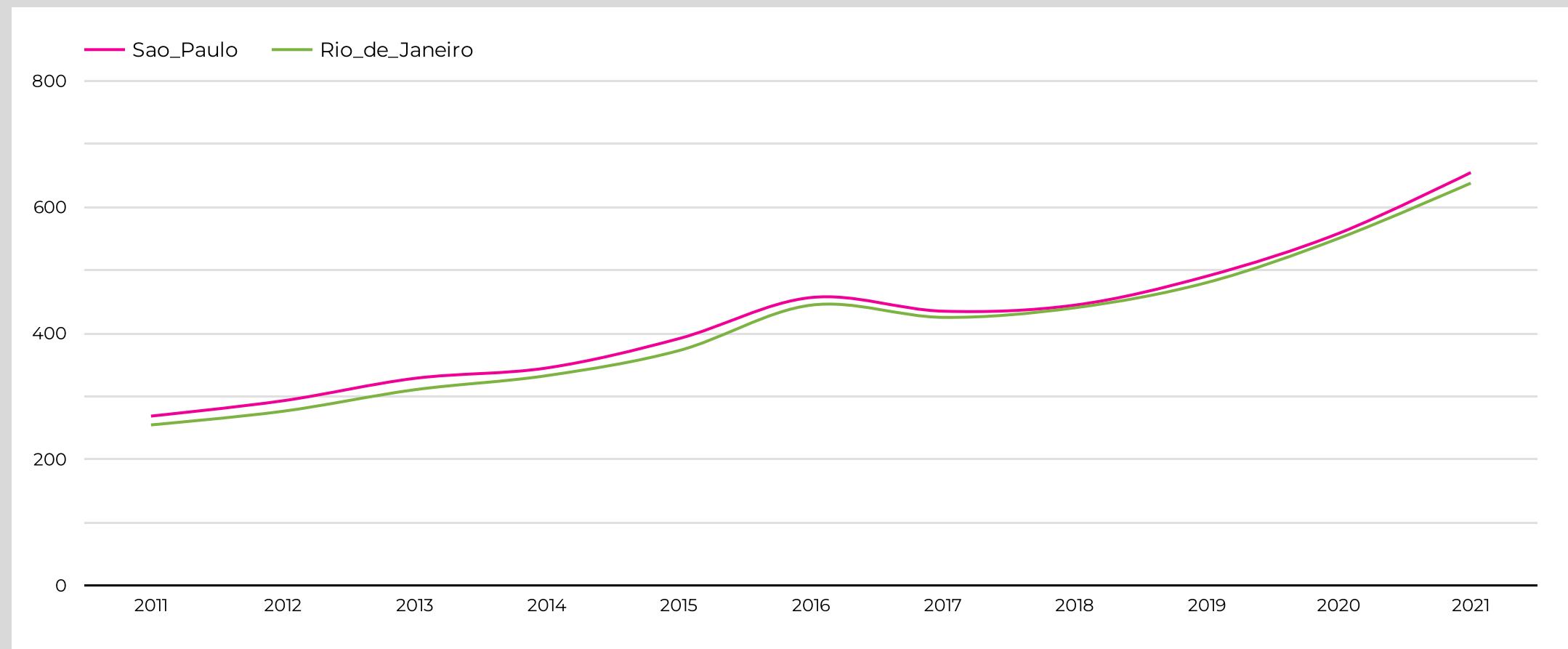


Custo de vida nas cidades de São Paulo x Rio de Janeiro

CESTA BÁSICA - RJ x SP

Cidade:: Rio_de_Janeiro, S... (2) ▾

Comparação do valor da cesta básica entre as cidades de Rio de Janeiro e São Paulo

FONTE: <https://www.dieese.org.br/cesta/>

INTEGRANTES

Allan Farias
Alyne Berriel
Enzo Lourenço
Yago Sousa

CONTATO

allanfarias1@gmail.com
alyneberriel@gmail.com
enzooliveira109@gmail.com
yago.devb@gmail.com

**AGRADECemos a ATENÇÃO
DE TODOS!**