

```

import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

# Caminho do arquivo
file_path =
"/content/drive/MyDrive/Python/TrensRJ/Supervia_PCA.xlsx"

# Carregar as planilhas
df_posto_empregos = pd.read_excel(file_path,
sheet_name='Posto_Empregos2021')
df_estat_supervia = pd.read_excel(file_path,
sheet_name='EstatSupervia')

# Selecionar as colunas relevantes
df_posto_empregos_selected = df_posto_empregos[['Renda2024',
'CNPJ2016', 'Soma de Empregos2016']]
df_estat_supervia_selected = df_estat_supervia.filter(like='Média
Mensal')

# Concatenar os dados para a análise
data_combined = pd.concat([df_posto_empregos_selected,
df_estat_supervia_selected], axis=1)

# Tratar valores ausentes
data_combined = data_combined.dropna()

# Padronizar os dados
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data_combined)

# Aplicar PCA
pca = PCA()
pca_results = pca.fit_transform(data_scaled)

# Criar DataFrame com os componentes principais (PCA)
pca_df = pd.DataFrame(
    data=pca_results,
    columns=[f'PC{i+1}' for i in range(pca_results.shape[1])]
)

# Criar DataFrame com as cargas (loadings) dos componentes
principais
components_df = pd.DataFrame(
    pca.components_,
    columns=data_combined.columns,
    index=[f'PC{i+1}' for i in range(pca.components_.shape[0])]
)

```

```
# Mostrar variância explicada
print("Variância explicada por componente principal:")
for i, var in enumerate(pca.explained_variance_ratio_, start=1):
    print(f"PC{i}: {var:.2%}")

print("\nCargas dos Componentes Principais:")
print(components_df)

# Caminhos de saída
output_pca_path =
"/content/drive/MyDrive/Python/TrensRJ/PCA_Resultados.parquet"
output_loadings_path =
"/content/drive/MyDrive/Python/TrensRJ/PCA_Loadings.parquet"

# Salvar os DataFrames em formato parquet
pca_df.to_parquet(output_pca_path, index=False)
components_df.to_parquet(output_loadings_path)

print(f"\nResultados PCA salvos como Parquet:")
print(f"Componentes principais: {output_pca_path}")
print(f"Cargas dos componentes: {output_loadings_path}")
```