



Pontifícia Universidade de Minas Gerais(PUC Minas)

# APÊNDICE - Análise da Precipitação em Rondônia usando Python e R: Uma Abordagem Comparativa para Modelagem Climática

## Relatório de Pesquisa

(Síntese e documentação de projeto desenvolvido)

### Alunos:

Gabriel da Silva Cassino

Welbert Junio Afonso de Almeida

**Disciplina:** Linguagens de Programação

**Curso:** Graduação em Engenharia da Computação

Bacharelado em Ciências da Computação

**Professor:** Marco Rodrigo Costa

Baseado no trabalho de Luiz Augusto Ferreira Monteiro - XV ENANPEGE

Belo Horizonte - MG

27 de Novembro de 2025

# APÊNDICE - Análise da Precipitação em Rondônia usando Python e R: Uma Abordagem Comparativa para Modelagem Climática

## Relatório de Pesquisa

(Síntese e documentação de projeto desenvolvido)

Documentação do exemplo prático em Python e R apresentado à disciplina de Linguagens de Programação, do Instituto de Ciências Exatas e Informática da Pontifícia Universidade Católica de Minas Gerais, como requisito avaliativo do Trabalho de Pesquisa 1 (TP1).

Baseado no trabalho de Luiz Augusto Ferreira Monteiro - XV ENANPEGE

Professor: **Marco Rodrigo Costa**



QR Code do Repositório

Link do repositório:

<[https://github.com/kasshinokun/Q3\\_Q4\\_2025\\_Public/tree/main/7\\_Semestre/LP/TP\\_TP](https://github.com/kasshinokun/Q3_Q4_2025_Public/tree/main/7_Semestre/LP/TP_TP)>

Gabriel da Silva Cassino  
Welbert Junio Afonso de Almeida  
Belo Horizonte — 2025

# Resumo

Este relatório apresenta uma análise comparativa entre as linguagens de programação Python e R no contexto da modelagem estatística de dados climáticos, com foco na precipitação no estado de Rondônia, Brasil. A partir da replicação do modelo descrito no artigo original, foram desenvolvidas duas aplicações interativas: uma em Python utilizando o framework Streamlit e outra em R com Shiny. O trabalho detalha o processamento de dados do WorldClim, a implementação de modelos de regressão linear múltipla e a criação de visualizações para análise exploratória. São discutidas as vantagens, desvantagens e casos de uso ideais para cada linguagem, destacando a performance, a expressividade estatística e a integração com ecossistemas de ciência de dados. Conclui-se que ambas as ferramentas são poderosas e complementares, sendo a escolha dependente do perfil do projeto, da equipe e dos objetivos de análise.

**Palavras-chave:** Python, R, modelagem climática, análise estatística, dados de precipitação, Rondônia, WorldClim.

# Abstract

This report presents a comparative analysis between the programming languages Python and R in the context of statistical modeling of climate data, focusing on precipitation in the state of Rondônia, Brazil. Based on replicating the model described in the original article, two interactive applications were developed: one in Python using the Streamlit framework and another in R with Shiny. The work details the processing of WorldClim data, the implementation of multiple linear regression models, and the creation of visualizations for exploratory analysis. The advantages, disadvantages, and ideal use cases for each language are discussed, highlighting performance, statistical expressiveness, and integration with data science ecosystems. It is concluded that both tools are powerful and complementary, with the choice depending on the project profile, team, and analysis objectives.

**Keywords:** Python, R, climate modeling, statistical analysis, precipitation data, Rondônia, WorldClim.

# Sumário

<b>1 APÊNDICE: Análise Comparativa entre Python e R para Modelagem Estatística de Dados Climáticos</b>	<b>5</b>
1.1 O Uso de Python e R na Análise e Modelagem Estatística de Dados . . . . .	5
1.1.1 Contexto Histórico e Evolução . . . . .	6
1.1.2 Ecossistemas e Comunidades . . . . .	6
1.1.3 Adequação para Pesquisa Climática . . . . .	7
1.2 Abordagem em Python com Streamlit . . . . .	7
1.2.1 Arquitetura da Aplicação . . . . .	7
1.2.2 Processamento de Dados Climáticos . . . . .	7
1.2.3 Modelagem Estatística . . . . .	8
1.2.4 Visualização Interativa . . . . .	9
1.2.5 Vantagens da Abordagem Python . . . . .	9
1.3 Abordagem em R com Shiny . . . . .	9
1.3.1 Arquitetura da Aplicação Shiny . . . . .	9
1.3.2 Processamento de Dados com Tidyverse . . . . .	10
1.3.3 Modelagem Estatística em R . . . . .	11
1.3.4 Visualização com ggplot2 . . . . .	12
1.3.5 Vantagens da Abordagem R . . . . .	12
1.4 Diferenças, Vantagens e Desvantagens . . . . .	13
1.4.1 Análise Comparativa Detalhada . . . . .	13
1.4.2 Casos de Uso Recomendados . . . . .	13
1.4.3 Tendências e Futuro . . . . .	14

# 1 APÊNDICE: Análise Comparativa entre Python e R para Modelagem Estatística de Dados Climáticos

A análise e modelagem de dados climáticos constituem uma área de extrema relevância científica e socioambiental, especialmente em regiões sensíveis como a Amazônia. Neste contexto, a escolha da ferramenta computacional adequada pode influenciar significativamente a eficiência, a reproduzibilidade e a profundidade das análises realizadas. Este relatório tem como objetivo principal comparar duas das principais linguagens utilizadas na ciência de dados — Python e R — na implementação de um modelo estatístico para análise da precipitação em Rondônia, replicando a abordagem metodológica proposta no trabalho de referência.

A motivação para este estudo surge da crescente adoção de ambas as linguagens em pesquisas climáticas, cada uma com seus pontos fortes: Python, com seu ecossistema robusto para processamento de grandes volumes de dados e integração com sistemas de produção; e R, reconhecido por sua expressividade estatística e capacidade gráfica avançada. Por meio do desenvolvimento de aplicações interativas em Streamlit (Python) e Shiny (R), buscou-se não apenas reproduzir os resultados do artigo original, mas também avaliar de forma prática as diferenças nas etapas de manipulação de dados, modelagem e visualização.

A estrutura do relatório está organizada da seguinte forma: inicialmente, são apresentados os fundamentos teóricos e contextuais do uso de Python e R na análise de dados. Em seguida, detalha-se a implementação em Python, com ênfase na arquitetura modular, no processamento de dados do WorldClim e na criação de modelos e gráficos interativos. Na sequência, descreve-se a abordagem equivalente em R, utilizando o tidyverse para manipulação de dados, modelos nativos de regressão e visualizações com ggplot2. Por fim, uma análise comparativa sintetiza as vantagens, desvantagens e cenários de aplicação recomendados para cada linguagem, contribuindo para uma decisão informada no planejamento de futuros projetos na área de modelagem climática.

## 1.1 O Uso de Python e R na Análise e Modelagem Estatística de Dados

A escolha entre Python e R para análise de dados climáticos representa uma decisão estratégica que impacta desde o fluxo de trabalho até a interpretação dos resultados. Ambas as linguagens emergiram como ferramentas fundamentais na ciência de dados moderna, porém com filosofias de desenvolvimento e ecossistemas distintos (????).

### **1.1.1 Contexto Histórico e Evolução**

Python, desenvolvida por Guido van Rossum em 1991, foi concebida como uma linguagem de programação de propósito geral com ênfase na legibilidade do código. Sua adoção na ciência de dados cresceu exponencialmente com o desenvolvimento de bibliotecas especializadas como NumPy, Pandas e Scikit-learn. A filosofia "Zen do Python", que enfatiza simplicidade e legibilidade, tornou-a particularmente atraente para projetos que exigem colaboração entre equipes multidisciplinares.

R, por outro lado, foi criado por Ross Ihaka e Robert Gentleman em 1993 especificamente para análise estatística e computação gráfica. Derivado da linguagem S desenvolvida nos Laboratórios Bell, R foi adotado rapidamente pela comunidade acadêmica e de pesquisa. A filosofia do R é centrada na expressividade estatística e na capacidade de implementar métodos analíticos complexos de forma intuitiva.

### **1.1.2 Ecossistemas e Comunidades**

O ecossistema Python para ciência de dados é caracterizado por sua abrangência e integração com outras áreas da computação. Principais componentes incluem:

- **Manipulação de dados:** Pandas, NumPy
- **Visualização:** Matplotlib, Seaborn, Plotly
- **Modelagem estatística:** Statsmodels, Scikit-learn
- **Aplicações web:** Streamlit, Dash
- **Computação científica:** SciPy, Astropy

O ecossistema R, conhecido como "R Project", é notável por sua profundidade em métodos estatísticos:

- **Manipulação de dados:** dplyr, tidyr (Tidyverse)
- **Visualização:** ggplot2, lattice
- **Modelagem estatística:** Base R, nlme, survival
- **Aplicações web:** Shiny, Flexdashboard
- **Relatórios reproduzíveis:** R Markdown, knitr

### 1.1.3 Adequação para Pesquisa Climática

Na pesquisa climática, ambas as linguagens oferecem capacidades complementares. Python destaca-se no processamento de grandes volumes de dados espaciais e integração com sistemas de informação geográfica, enquanto R oferece métodos estatísticos mais sofisticados para análise de séries temporais e modelagem climática (????).

A tendência atual na comunidade científica é o uso híbrido, aproveitando os pontos fortes de cada linguagem conforme a etapa do fluxo de trabalho analítico.

## 1.2 Abordagem em Python com Streamlit

### 1.2.1 Arquitetura da Aplicação

A implementação em Python utilizou o framework Streamlit para desenvolvimento de uma aplicação web interativa. A arquitetura segue princípios de modularidade e reatividade:

```
app/  
  data_processing.py  
  modeling.py  
  visualization.py  
main.py
```

O código principal orquestra o fluxo de processamento através de funções especializadas:

```
def download_and_process_data(url, target_dir, zip_filename):  
def process_worldclim_data(data_dir):  
def train_regression_model(df_clean):  
def create_visualizations(df_clean, model):
```

### 1.2.2 Processamento de Dados Climáticos

O módulo de processamento implementa a leitura de arquivos ASCII Grid (.asc) do WorldClim:

```
def read_asc_file(filepath):  
    try:  
        # Ler cabeçalho (6 linhas)  
        with open(filepath, 'r') as con:  
            header_lines = [con.readline().strip() for _ in range(6)]
```

```

        header = {}
        for line in header_lines:
            parts = line.split()
            key = parts[0]
            value = int(parts[1]) if key in ['ncols', 'nrows'] else float(parts[1])
            header[key] = value

    # Ler dados matriciais
    data = np.loadtxt(filepath, skiprows=6)
    return {'header': header, 'data': data}
except Exception as e:
    print(f"Erro ao ler arquivo: {e}")
    return None

```

Esta função demonstra a capacidade do Python em manipular formatos de dados científicos complexos, combinando processamento de texto com análise numérica.

### 1.2.3 Modelagem Estatística

A implementação do modelo de regressão linear múltipla utiliza Scikit-learn:

```

from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score, mean_squared_error

def train_regression_model(df_clean):
    # Preparar dados
    model_df = df_clean[['bio1', 'bio12', 'bio4']].dropna()
    X = model_df[['bio12', 'bio4']]
    y = model_df['bio1']

    # Dividir em treino e teste
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.3, random_state=42
    )

    # Treinar modelo
    model = LinearRegression()
    model.fit(X_train, y_train)

    # Fazer previsões e calcular métricas

```

```

y_pred = model.predict(X_test)
r_squared = r2_score(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)

return model, {
    'r_squared': r_squared,
    'mse': mse,
    'predictions': y_pred,
    'actual': y_test
}

```

#### 1.2.4 Visualização Interativa

A criação de visualizações utiliza Matplotlib e Seaborn integrados ao Streamlit:

```

def create_correlation_heatmap(df_clean):
    bio_cols = [col for col in df_clean.columns if col.startswith('bio')]
    corr_matrix = df_clean[bio_cols].corr()

    fig, ax = plt.subplots(figsize=(12, 10))
    sns.heatmap(corr_matrix, annot=True, fmt='.2f',
                cmap='coolwarm', center=0, ax=ax)
    ax.set_title('Matriz de Correlação - Variáveis Bioclimáticas')
    return fig

```

#### 1.2.5 Vantagens da Abordagem Python

- **Performance:** Processamento eficiente de grandes datasets climáticos
- **Integração:** Conectividade com bancos de dados e APIs web
- **Deployment:** Facilidade em colocar aplicações em produção
- **Ecossistema ML:** Acesso a bibliotecas avançadas de machine learning
- **Comunidade:** Amplo suporte e documentação

### 1.3 Abordagem em R com Shiny

#### 1.3.1 Arquitetura da Aplicação Shiny

A implementação em R utiliza o framework Shiny, seguindo a arquitetura reativa característica da linguagem:

```

ui <- fluidPage(
  titlePanel("Análise Climática"),
  sidebarLayout(
    sidebarPanel(...),
    mainPanel(...)
  )
)

server <- function(input, output, session) {
  # Lógica reativa
  rv <- reactiveValues(
    df_clean = NULL,
    model = NULL
  )
}

```

### 1.3.2 Processamento de Dados com Tidyverse

O processamento dos dados climáticos utiliza o ecossistema Tidyverse para manipulação eficiente:

```

library(dplyr)
library(tidyr)
library(readr)

process_worldclim_data <- function(data_dir) {
  # Listar e ler arquivos ASC
  asc_files <- list.files(data_dir, pattern = "\\.asc$", full.names = TRUE)

  # Processar primeiro arquivo para estrutura
  first_result <- read_asc_file(asc_files[1])
  if (is.null(first_result)) return(NULL)

  # Coletar dados de todos os arquivos
  all_data <- list()
  for (filepath in asc_files) {
    result <- read_asc_file(filepath)
    if (!is.null(result)) {
      var_name <- tools::file_path_sans_ext(basename(filepath))
      all_data[[var_name]] <- as.vector(result$data)
    }
  }
}

# Teste
all_data <- process_worldclim_data("C:/Users/.../worldclim")

```

```

        }

    }

# Criar DataFrame e processar
df <- as.data.frame(all_data)
df_clean <- df %>%
  filter(!rowSums(. == nodata_value, na.rm = TRUE) > 0)

return(df_clean)
}

```

### 1.3.3 Modelagem Estatística em R

A implementação do modelo de regressão utiliza as funções estatísticas nativas do R:

```

train_regression_model <- function(df_clean) {
  # Preparar dados
  model_df <- df_clean %>%
    select(bio1, bio12, bio4) %>%
    na.omit()

  # Dividir em treino e teste
  set.seed(42)
  split_index <- sample(1:nrow(model_df), size = 0.7 * nrow(model_df))
  train_data <- model_df[split_index, ]
  test_data <- model_df[-split_index, ]

  # Treinar modelo
  model <- lm(bio1 ~ bio12 + bio4, data = train_data)

  # Fazer previsões
  predictions <- predict(model, newdata = test_data)

  # Calcular métricas
  r_squared <- summary(model)$r.squared
  mse <- mean((predictions - test_data$bio1)^2)

  return(list(
    model = model,
    metrics = list(

```

```

    r_squared = r_squared,
    mse = mse,
    predictions = predictions,
    actual = test_data$bio1
)
))
}

```

### 1.3.4 Visualização com ggplot2

A criação de visualizações utiliza o sistema de gramática de gráficos do ggplot2:

```

library(ggplot2)
library(reshape2)

create_correlation_heatmap <- function(df_clean) {
  bio_cols <- grep("^bio", names(df_clean), value = TRUE)
  corr_matrix <- cor(df_clean[bio_cols], use = "complete.obs")

  corr_melted <- melt(corr_matrix)

  ggplot(corr_melted, aes(x = Var1, y = Var2, fill = value)) +
    geom_tile() +
    scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                         midpoint = 0, limit = c(-1,1)) +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
    labs(title = "Matriz de Correlação - Variáveis Bioclimáticas",
         x = "", y = "", fill = "Correlação")
}

```

### 1.3.5 Vantagens da Abordagem R

- **Expressividade Estatística:** Sintaxe natural para modelos complexos
- **Visualização Avançada:** Gráficos de alta qualidade com ggplot2
- **Reprodutibilidade:** Integração nativa com R Markdown
- **Comunidade Acadêmica:** Amplo suporte em pesquisas científicas
- **Pacotes Especializados:** Acesso a métodos estatísticos avançados

## 1.4 Diferenças, Vantagens e Desvantagens

### 1.4.1 Análise Comparativa Detalhada

Tabela 1: Comparação Detalhada entre Python e R para Análise Climática

Aspecto	Python	R
Sintaxe e Aprendizado	Sintaxe clara e consistente, mais fácil para iniciantes em programação. Curva de aprendizagem suave.	Sintaxe específica para análise estatística, pode ser desafiadora para não-estatísticos. Maior curva de aprendizagem inicial.
Performance	Melhor performance com grandes volumes de dados. Processamento mais eficiente de arrays numéricos.	Performance otimizada para análises estatísticas em datasets de tamanho moderado. Pode ter limitações com dados muito grandes.
Visualização	Gráficos funcionais com Matplotlib/Seaborn. Mais customização requer código adicional.	Visualizações sofisticadas com ggplot2. Sistema de gramática de gráficos mais expressivo.
Ecossistema	Ecossistema amplo para machine learning e produção. Melhor para pipelines completos.	Ecossistema rico em métodos estatísticos. Superior para análise exploratória e modelagem estatística.
Comunidade	Comunidade diversificada com forte presença na indústria. Melhor para aplicações em produção.	Comunidade academicamente orientada. Melhor suporte para métodos estatísticos avançados.
Integração	Excelente integração com sistemas existentes, APIs web e bancos de dados.	Boa integração com formatos estatísticos, mas mais limitada com sistemas corporativos.

### 1.4.2 Casos de Uso Recomendados

**Python é recomendado para:**

- Projetos que exigem integração com sistemas de produção
- Aplicações de machine learning em larga escala
- Processamento de grandes volumes de dados climáticos

- Desenvolvimento de APIs e serviços web
- Projetos multidisciplinares com equipes diversificadas

**R é recomendado para:**

- Pesquisa acadêmica e análise estatística avançada
- Desenvolvimento de métodos estatísticos novos
- Criação de relatórios reproduzíveis com R Markdown
- Análise exploratória de dados e visualização científica
- Projetos com foco em publicação científica

#### 1.4.3 Tendências e Futuro

Ambas as linguagens continuam evoluindo para atender às demandas da ciência de dados moderna:

- **Python:** Fortalecimento do ecossistema estatístico com bibliotecas como Pingouin e Statsmodels
- **R:** Melhorias de performance com ALTREP e integração melhorada com outras linguagens
- **Convergência:** Desenvolvimento de ferramentas que permitem uso integrado (ex.: reticulate, rpy2)

A escolha ideal depende dos objetivos específicos do projeto, do perfil da equipe e do contexto de aplicação. Para a pesquisa climática, o uso complementar de ambas as linguagens frequentemente oferece os melhores resultados.

## Referências

- Fick, S. E., & Hijmans, R. J. (2017). WorldClim 2: new 1km spatial resolution climate surfaces for global land areas. *International Journal of Climatology*, 37(12), 4302–4315.
- Harrell, F. E. (2001). *Regression modeling strategies: with applications to linear models, logistic regression, and survival analysis*. Springer.
- Wickham, H., & Grolemund, G. (2016). *R for data science: import, tidy, transform, visualize, and model data*. O'Reilly Media, Inc.
- McKinney, W. (2017). *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. O'Reilly Media, Inc.
- Hijmans, R. J., Cameron, S. E., Parra, J. L., Jones, P. G., & Jarvis, A. (2005). Very high resolution interpolated climate surfaces for global land areas. *International journal of climatology*, 25(15), 1965–1978.
- Almeida, C. T., Oliveira-Júnior, J. F., Delgado, R. C., Cubo, P., & Ramos, M. C. (2017). Spatiotemporal rainfall and temperature trends throughout the Brazilian Legal Amazon, 1973–2013. *International Journal of Climatology*, 37, 2013–2026.
- De Souza, V. A. S. (2014). Eventos de precipitações extremas na Amazônia Ocidental: Rondônia-Brasil. *Revista Brasileira de Climatologia*, 14.
- De Oliveira, L. L. (2020). Influência do El Niño Oscilação Sul-ENOS-(2015/2016) no regime hidrológico e geração de energia na usina hidrelétrica de Curuá-Una, Amazônia central. *Revista Ibero-Americana de Ciências Ambientais*, 11(4), 136–144.