



Pontifícia Universidade de Minas Gerais(PUC Minas)

# 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

# Análise da Precipitação em Rondônia usando Python e R:

## Uma Abordagem Comparativa para Modelagem Climática

### Relatório de Pesquisa

(Síntese e documentação de projeto desenvolvido)

Aluno: Gabriel da Silva Cassino

Aluno: Welbert Junio Afonso de Almeida

Disciplina: Linguagens de Programação

Curso: Graduação em Engenharia da Computação

Bacharelado em Ciências da Computação

Instituição: Pontifícia Universidade de Minas Gerais(PUC Minas)

27 de Novembro de 2025

## Resumo

O artigo *PYTHON E R COMO FERRAMENTAS DE ANÁLISE DA PRECIPITAÇÃO EM RONDÔNIA*, de Luiz Augusto Ferreira Monteiro, analisa a precipitação em Rondônia entre 1980 e 2020, utilizando dados da Agência Nacional de Águas (ANA). O estudo investiga a influência de fenômenos como El Niño, Oscilação Multidecadal do Atlântico e Oscilação Decadal do Pacífico no clima da região. A análise, feita com Python e R, revela padrões complexos, como o aumento das chuvas em Porto Velho durante o El Niño. O trabalho destaca a importância de entender essas interações para criar modelos climáticos precisos e estratégias de adaptação às mudanças climáticas.

**Palavras-chave:** Precipitação, Rondônia, Python, R, Modelagem Climática, ENOS

## Abstract

The article "PYTHON AND R AS TOOLS FOR ANALYZING PRECIPITATION IN RONDÔNIA," by Luiz Augusto Ferreira Monteiro, analyzes precipitation in Rondônia between 1980 and 2020, using data from the National Water Agency (ANA). The study investigates the influence of phenomena such as El Niño, the Atlantic Multidecadal Oscillation, and the Pacific Decadal Oscillation on the region's climate. The analysis, conducted using Python and R, reveals complex patterns, such as increased rainfall in Porto Velho during El Niño. The work highlights the importance of understanding these interactions to create accurate climate models and adaptation strategies to climate change.

**Keywords:** Precipitation, Rondônia, Python, R, Climate Modeling, ENSO

# Sumário

<b>1</b>	<b>Introdução</b>	<b>5</b>
<b>2</b>	<b>Metodologia</b>	<b>5</b>
<b>3</b>	<b>Resultados e Discussão</b>	<b>5</b>
<b>4</b>	<b>Considerações Finais</b>	<b>6</b>
<b>5</b>	<b>Conclusão</b>	<b>6</b>
<b>A</b>	<b>Análise Comparativa entre Python e R para Modelagem Estatística de Dados Climáticos</b>	<b>7</b>
A.1	O Uso de Python e R na Análise e Modelagem Estatística de Dados . . . .	7
A.1.1	Contexto Histórico e Evolução . . . . .	7
A.1.2	Ecossistemas e Comunidades . . . . .	7
A.1.3	Adequação para Pesquisa Climática . . . . .	8
A.2	Abordagem em Python com Streamlit . . . . .	8
A.2.1	Arquitetura da Aplicação . . . . .	8
A.2.2	Processamento de Dados Climáticos . . . . .	9
A.2.3	Modelagem Estatística . . . . .	9
A.2.4	Visualização Interativa . . . . .	10
A.2.5	Vantagens da Abordagem Python . . . . .	10
A.3	Abordagem em R com Shiny . . . . .	11
A.3.1	Arquitetura da Aplicação Shiny . . . . .	11
A.3.2	Processamento de Dados com Tidyverse . . . . .	11
A.3.3	Modelagem Estatística em R . . . . .	12
A.3.4	Visualização com ggplot2 . . . . .	13
A.3.5	Vantagens da Abordagem R . . . . .	14
A.4	Diferenças, Vantagens e Desvantagens . . . . .	15
A.4.1	Análise Comparativa Detalhada . . . . .	15
A.4.2	Casos de Uso Recomendados . . . . .	15
A.4.3	Tendências e Futuro . . . . .	16

# 1 Introdução

O presente trabalho tem como objetivo apresentar uma síntese do artigo científico intitulado *PYTHON E R COMO FERRAMENTAS DE ANÁLISE DA PRECIPITAÇÃO EM RONDÔNIA*, de autoria de Luiz Augusto Ferreira Monteiro. Este artigo aborda a análise da precipitação no estado de Rondônia, Brasil, no período de 1980 a 2020, com foco na influência de fenômenos oceano-atmosféricos como El Niño Oscilação Sul (ENOS), Oscilação Multidecadal do Atlântico (OMA) e Oscilação Decadal do Pacífico (ODP). A relevância do estudo reside na crescente preocupação com os extremos climáticos e seus impactos, especialmente na região amazônica, onde a compreensão dos padrões de chuva é crucial para a adaptação às mudanças climáticas.

## 2 Metodologia

A metodologia empregada no artigo baseou-se na coleta de dados de precipitação diária do Sistema de Informações Hidrológicas (HidroWeb) da Agência Nacional de Águas (ANA). Para lidar com dados faltantes, foi utilizada a técnica de imputação pela média da chuva. O estado de Rondônia foi dividido em mesorregiões, conforme dados do IBGE, para uma análise mais detalhada dos postos pluviométricos. As análises incluíram a construção da correlação de Pearson entre a chuva e os fenômenos meteorológicos (ENOS, OMA, ODP), visualização da série histórica diária, e identificação de dias acima do padrão pluviométrico, dias sem precipitação (verão amazônico) e dias chuvosos (inverno). As técnicas de análise foram implementadas através de algoritmos programados em Python, utilizando bibliotecas de ciência de dados como Pandas e Matplotlib.

## 3 Resultados e Discussão

Os resultados do estudo indicam que Rondônia registra uma quantidade significativa de chuvas, frequentemente ultrapassando 100 mm diários. A pesquisa corrobora estudos anteriores que apontam uma média anual de chuvas entre 1.400 e 2.600 mm na região. A análise específica de municípios como Porto Velho, Mirante da Serra e Cerejeiras revelou comportamentos distintos no regime de chuvas, com Porto Velho registrando maiores índices durante anos de El Niño. A complexidade dos padrões de chuva na Amazônia Legal é influenciada pela interação de diversos fenômenos oceano-atmosféricos, destacando a necessidade de modelos climáticos precisos e estratégias de adaptação.

## 4 Considerações Finais

O estudo conclui que há uma correlação clara entre os fenômenos oceano-atmosféricos estudados e a precipitação em Rondônia. Geralmente, as fases frias dos fenômenos oceânicos estão associadas a maiores índices pluviométricos na região.

Foi constatada uma mudança significativa no regime de chuvas, especialmente a partir de 2005, com uma tendência de redução na variabilidade durante o verão e o inverno, mas com aumento na frequência de eventos extremos. Esta mudança de padrão tem implicações importantes para o planejamento de recursos hídricos, agricultura e gestão de riscos climáticos na região.

A abordagem comparativa entre Python e R demonstrou a viabilidade de ambas as linguagens para pesquisa climática, cada uma oferecendo vantagens específicas conforme o contexto de aplicação. Para trabalhos futuros, sugere-se a exploração de técnicas de Inteligência Artificial e aprendizado de máquina para previsão de chuvas, bem como a expansão da análise para outras variáveis climáticas.

## 5 Conclusão

O artigo demonstra a complexidade dos padrões de precipitação em Rondônia e a forte influência de fenômenos oceano-atmosféricos. A utilização de Python e R como ferramentas de análise se mostrou eficaz para identificar padrões, anomalias e tendências. A compreensão dessas interações é fundamental para o desenvolvimento de estratégias de adaptação e mitigação dos impactos das mudanças climáticas na região amazônica.

# APÊNDICE

## A Análise Comparativa entre Python e R para Modelagem Estatística de Dados Climáticos

### A.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 (Wickham2016, McKinney2017).

#### A.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.

#### A.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 reprodutíveis:** R Markdown, knitr

### A.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 (**Hijmans2005; Fick2017**).

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.

## A.2 Abordagem em Python com Streamlit

### A.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):
```

### A.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.

### A.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
}

```

#### A.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

```

#### A.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
- **Ecosistema ML:** Acesso a bibliotecas avançadas de machine learning
- **Comunidade:** Amplo suporte e documentação

## A.3 Abordagem em R com Shiny

### A.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
  )
}
```

### A.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)
  }
}

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

return(df_clean)
}

```

### A.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
  )
))
}

```

### A.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")
}

```

}

#### A.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

## A.4 Diferenças, Vantagens e Desvantagens

### A.4.1 Análise Comparativa Detalhada

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

Aspecto	Python	R
<b>Sintaxe e Aprendizado</b>	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.
<b>Performance</b>	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.
<b>Visualização</b>	Gráficos funcionais com Matplotlib/Seaborn. Mais customização requer código adicional.	Visualizações sofisticadas com ggplot2. Sistema de gramática de gráficos mais expressivo.
<b>Ecossistema</b>	Ecossistema amplo para machine learning e produção. Melhor para pipelines completos.	Ecossistema rico em métodos estatísticos. Superior para análise exploratória e modelagem estatística.
<b>Comunidade</b>	Comunidade diversificada com forte presença na indústria. Melhor para aplicações em produção.	Comunidade academicamente orientada. Melhor suporte para métodos estatísticos avançados.
<b>Integração</b>	Excelente integração com sistemas existentes, APIs web e bancos de dados.	Boa integração com formatos estatísticos, mas mais limitada com sistemas corporativos.

### A.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 reprodutíveis com R Markdown
- Análise exploratória de dados e visualização científica
- Projetos com foco em publicação científica

#### A.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

- MONTEIRO, Luiz Augusto Ferreira. Python e R como ferramentas de análise da precipitação em Rondônia. In: ENCONTRO NACIONAL DE PÓS-GRADUAÇÃO E PESQUISA EM GEOGRAFIA, 15., 2023, Campina Grande. *Anais do XV ENANPEGE*. Campina Grande: Realize Editora, 2023. p. (Páginas não especificadas). ISSN 2175-8875. Disponível em: <<https://editorarealize.com.br/artigo/visualizar/94351>>. Acesso em: 05 out. 2025.
- 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.