📚 Documentação Técnica: Ferramentas Estatísticas para Modelo SARIMA

📋 Índice

🎯 Visão Geral

Este documento explica **todas as ferramentas estatísticas** implementadas no projeto de previsão de estoque usando modelos SARIMA. Cada ferramenta é justificada teoricamente e explicada em detalhes para que você possa defender sua escolha em apresentações, defesas de TCC ou discussões técnicas.

Objetivo do Projeto: Prever demanda futura de estoque para produtos (SKUs) em um e-commerce de brinquedos, utilizando essas previsões como uma das métricas na ferramenta de elencação (ranking) para reposição de estoque.

📊 Metodologia Box-Jenkins

O que é?

A **Metodologia Box-Jenkins** é um processo iterativo e sistemático para construir modelos de séries temporais. Foi desenvolvida por George Box e Gwilym Jenkins em 1970 e é considerada o **padrão-ouro** para modelagem de séries temporais.

Por que usar?

1. **Rigor Estatístico**: Garante que o modelo seja adequado aos dados
2. **Validação Completa**: Testa todas as suposições do modelo
3. **Reprodutibilidade**: Processo sistemático e documentado
4. **Aceitação Acadêmica**: Metodologia amplamente aceita na literatura

Como funciona?

A metodologia tem **4 etapas principais**:

1. **Identificação**: Determina se a série é adequada para SARIMA
2. **Estimação**: Encontra os melhores parâmetros do modelo
3. **Diagnóstico**: Verifica se o modelo é adequado
4. **Previsão**: Gera previsões futuras

Referências

- **Box, G. E. P., & Jenkins, G. M.** (1976). *Time Series Analysis: Forecasting and Control*. Holden-Day.
- **Hyndman, R. J., & Athanasopoulos, G.** (2021). *Forecasting: Principles and Practice* (3rd ed.). OTexts.

🔍 Ferramentas de Identificação

1. Teste de Estacionariedade (Augmented Dickey-Fuller - ADF)

O que é?

O **Teste ADF** verifica se uma série temporal é **estacionária**. Uma série é estacionária quando:

- **Média constante** ao longo do tempo
- **Variância constante** ao longo do tempo
- **Autocovariância** depende apenas do lag, não do tempo

Por que é necessário?

Modelos SARIMA requerem séries estacionárias (ou estacionárias após diferenciação). Se a série não for estacionária:

- ❌ O modelo não captura corretamente os padrões
- ❌ Previsões podem ser enviesadas
- ❌ Intervalos de confiança podem ser incorretos

Como funciona?

Hipóteses:

- **H_0**: A série possui raiz unitária (não estacionária)
- **H_1**: A série é estacionária

Interpretação:

- Se **p-value < 0.05**: Rejeita H_0 → Série é estacionária ✅
- Se **p-value ≥ 0.05**: Não rejeita H_0 → Série não é estacionária ❌

Ação:

- Se não estacionária: Aplicar **diferenciação** (parâmetro `d` no ARIMA)
- O `auto_arima` faz isso automaticamente

Implementação no Projeto

```
# Em analise_box_jenkins_sarima.py
def teste_estacionariedade_adf(self):
    resultado = adfuller(self.serie.dropna(), autolag='AIC')
    p_value = resultado[1]
    is_stationary = p_value < 0.05
    return is_stationary
```

Referências

- **Dickey, D. A., & Fuller, W. A.** (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74(366), 427-431.
- **Said, S. E., & Dickey, D. A.** (1984). Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika*, 71(3), 599-607.

---

2. Análise de Autocorrelação (ACF e PACF)

O que é?

- **ACF (Autocorrelation Function)**: Mede correlação entre valores da série em diferentes lags
- **PACF (Partial Autocorrelation Function)**: Mede correlação direta entre valores, removendo efeitos de lags intermediários

Por que é necessário?

ACF e PACF ajudam a identificar os parâmetros do modelo SARIMA:

- **PACF**: Identifica ordem **p** (AutoRegressivo)
    - Se PACF corta abruptamente no lag `k`, então `p = k`
- **ACF**: Identifica ordem **q** (Média Móvel)
    - Se ACF corta abruptamente no lag `k`, então `q = k`
- **Padrões Sazonais**: Picos em lags múltiplos do período sazonal (ex: lags 7, 14, 21 para sazonalidade semanal)

Como funciona?

Interpretação Visual:

- **Corte abrupto**: Lag onde a autocorrelação cai para dentro do intervalo de confiança
- **Decaimento gradual**: Indica necessidade de diferenciação
- **Picos periódicos**: Indica sazonalidade

Exemplo:

- PACF corta no lag 2 → `p = 2`
- ACF corta no lag 1 → `q = 1`
- Picos em lags 7, 14, 21 → Sazonalidade semanal (`m = 7`)

Implementação no Projeto

```
# Em analise_box_jenkins_sarima.py
def analise_acf_pacf(self, lags=40):
    acf_values, acf_confint = acf(self.serie.dropna(), nlags=lags, alpha=
    pacf_values, pacf_confint = pacf(self.serie.dropna(), nlags=lags, alp
    # Identifica lags significativos
    return acf_values, pacf_values
```

Referências

- **Box, G. E. P., & Jenkins, G. M.** (1976). *Time Series Analysis: Forecasting and Control*. Holden-Day. (Cap. 2)
- **Chatfield, C.** (2016). *The Analysis of Time Series: An Introduction* (7th ed.). CRC Press.

---

3. Decomposição Sazonal

O que é?

A **Decomposição Sazonal** separa uma série temporal em componentes:

- **Tendência (Trend)**: Movimento de longo prazo
- **Sazonalidade (Seasonal)**: Padrões que se repetem em intervalos regulares
- **Resíduo (Residual)**: Componente aleatória (ruído)

Por que é necessário?

1. **Identificar Sazonalidade**: Confirma se há padrões sazonais e qual o período
2. **Ajustar Modelo**: Define o parâmetro `m` (período sazonal) do SARIMA
3. **Entender Dados**: Visualiza componentes separadamente
4. **Calcular Força da Sazonalidade**: Mede quão forte é o padrão sazonal

Como funciona?

Modelo Aditivo:

```
Série = Tendência + Sazonalidade + Resíduo
```

Força da Sazonalidade:

```
Força = Var(Sazonalidade) / [Var(Sazonalidade) + Var(Resíduo)]
```

- **Força > 0.5**: Sazonalidade forte (importante modelar)
- **Força < 0.5**: Sazonalidade fraca (pode ser ignorada)

Implementação no Projeto

```python
# Em analise_box_jenkins_sarima.py
def decomposicao_sazonal(self, periodo=30):
    decomposicao = seasonal_decompose(
        self.serie.dropna(),
        model='additive',
        period=periodo
    )
    # Calcula força da sazonalidade
    var_sazonal = np.var(decomposicao.seasonal.dropna())
    var_residuo = np.var(decomposicao.resid.dropna())
    forca_sazonal = var_sazonal / (var_sazonal + var_residuo)
    return decomposicao, forca_sazonal
```

Referências

- **Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I.** (1990). STL: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6(1), 3-73.
- **Hyndman, R. J., & Athanasopoulos, G.** (2021). *Forecasting: Principles and Practice* (3rd ed.). OTexts. (Cap. 6)

⚙️ Ferramentas de Estimação

4. Auto-ARIMA (Stepwise Search)

O que é?

Auto-ARIMA é um algoritmo que **automaticamente encontra os melhores parâmetros** (p, d, q) x (P, D, Q, s) para um modelo SARIMA, testando múltiplas combinações e escolhendo a melhor baseado em critérios estatísticos.

Por que é necessário?

Problema Manual:

- Testar todas as combinações manualmente é **impraticável**
- Para um SKU: ~1000+ combinações possíveis
- Para 1000 SKUs: ~1 milhão de combinações

Solução Auto-ARIMA:

- ✅ **Automatizado**: Testa combinações automaticamente
- ✅ **Eficiente**: Algoritmo stepwise reduz tempo de busca
- ✅ **Escalável**: Funciona para centenas/milhares de produtos
- ✅ **Objetivo**: Usa critérios estatísticos (AIC) para escolher

Como funciona?

Algoritmo Stepwise:

1. Começa com modelo simples (p=0, d=0, q=0)
2. Testa adicionar/remover parâmetros
3. Escolhe combinação com menor **AIC** (Akaike Information Criterion)
4. Para quando não há melhoria

Critério AIC:

```
AIC = -2 × log(Likelihood) + 2 × k
```

- **k**: Número de parâmetros
- **Menor AIC = Melhor modelo** (equilibra ajuste e complexidade)

Implementação no Projeto

```
# Em sarima_estoque.py
modelo = auto_arima(
    serie,
    seasonal=True,
    m=30,                    # Período sazonal
    stepwise=True,           # Busca eficiente
    information_criterion='aic',  # Critério de seleção
    max_p=5, max_d=2, max_q=5,
    max_P=2, max_D=1, max_Q=2
)
```

Referências

- **Hyndman, R. J., & Khandakar, Y.** (2008). Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software*, 27(3), 1-22.
- **Akaike, H.** (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6), 716-723.

---

5. Critério de Informação de Akaike (AIC)

O que é?

O **AIC** é um critério para comparar modelos, equilibrando:

- **Qualidade do Ajuste**: Quão bem o modelo se ajusta aos dados
- **Complexidade**: Número de parâmetros (penaliza modelos muito complexos)

Por que é necessário?

Problema do Overfitting:

- Modelos muito complexos podem se ajustar perfeitamente aos dados de treino
- Mas **falham em prever dados novos** (overfitting)
- AIC **penaliza complexidade excessiva**

Vantagens do AIC:

- ✅ Compara modelos objetivamente

- ✅ Previne overfitting
- ✅ Amplamente aceito na literatura

Como funciona?

Fórmula:

```
AIC = -2 × log(Likelihood) + 2 × k
```

Interpretação:

- **Menor AIC = Melhor modelo**
- Diferença > 2: Modelo significativamente melhor
- Diferença < 2: Modelos equivalentes

Alternativas:

- **BIC**: Penaliza mais a complexidade (melhor para amostras grandes)
- **AICc**: Versão corrigida para amostras pequenas

Referências

- **Akaike, H.** (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6), 716-723.
- **Burnham, K. P., & Anderson, D. R.** (2002). *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach* (2nd ed.). Springer.

🔬 Ferramentas de Diagnóstico

6. Teste de Ljung-Box (Resíduos)

O que é?

O **Teste de Ljung-Box** verifica se os **resíduos do modelo são não correlacionados** (ruído branco). Se os resíduos têm padrão, significa que o modelo não capturou toda a informação disponível.

Por que é necessário?

Suposição do Modelo SARIMA:

- Os resíduos devem ser **ruído branco** (aleatórios, não correlacionados)
- Se resíduos são correlacionados:
  - ❌ Modelo não capturou todos os padrões

    - ❌ Pode melhorar aumentando ordem do modelo
    - ❌ Previsões podem ser subótimas

Como funciona?

Hipóteses:

- **H_0**: Resíduos são não correlacionados (ruído branco) ✅
- **H_1**: Resíduos são correlacionados ❌

Estatística:

```
Q = n(n+2) × Σ(ρ²ₖ/(n-k))
```

- **n**: Tamanho da amostra
- **ρ_k**: Autocorrelação no lag k
- **k**: Número de lags testados

Interpretação:

- Se **p-value > 0.05**: Resíduos são ruído branco ✅ (modelo adequado)
- Se **p-value ≤ 0.05**: Resíduos são correlacionados ❌ (modelo pode melhorar)

Implementação no Projeto

```python
# Em analise_box_jenkins_sarima.py
def teste_ljung_box(self, lags=10):
    resultado = acorr_ljungbox(self.residuos.dropna(), lags=lags, return_
    p_value = resultado['lb_pvalue'].iloc[-1]
    residuos_ok = p_value > 0.05
    return residuos_ok
```

Referências

- **Ljung, G. M., & Box, G. E. P.** (1978). On a measure of lack of fit in time series models. *Biometrika*, 65(2), 297-303.
- **Box, G. E. P., & Pierce, D. A.** (1970). Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American Statistical Association*, 65(332), 1509-1526.

---

7. Testes de Normalidade dos Resíduos

O que é?

Testes estatísticos que verificam se os **resíduos seguem distribuição normal**. Implementamos 3 testes para robustez:

1. **Shapiro-Wilk**: Para amostras pequenas/médias
2. **Jarque-Bera**: Testa assimetria e curtose
3. **Anderson-Darling**: Teste robusto

Por que é necessário?

Suposição do Modelo SARIMA:

- Resíduos devem ser **normalmente distribuídos** para:
    - Intervalos de confiança serem válidos
    - Testes estatísticos funcionarem corretamente
    - Previsões serem confiáveis

Por que 3 testes?

- ✅ **Robustez**: Se todos concordam, conclusão é forte
- ✅ **Diferentes amostras**: Cada teste funciona melhor em diferentes tamanhos
- ✅ **Diferentes aspectos**: Testam normalidade de formas diferentes

Como funciona?

1. Shapiro-Wilk:

- **H_0**: Resíduos são normais
- **H_1**: Resíduos não são normais
- **Melhor para**: Amostras pequenas/médias ($n \leq 5000$)

2. Jarque-Bera:

- Testa **assimetria** (skewness) e **curtose** (kurtosis)
- Normal: Assimetria = 0, Curtose = 3
- **Melhor para**: Amostras grandes

3. Anderson-Darling:

- Teste robusto baseado na função de distribuição empírica
- **Melhor para**: Detectar desvios nas caudas da distribuição

Interpretação:

- Se **todos p-values > 0.05**: Resíduos são normais ✅
- Se **algum p-value ≤ 0.05**: Resíduos podem não ser normais ⚠️

Nota Importante:

- Resíduos não normais **não invalidam** o modelo
- Mas podem afetar intervalos de confiança
- Em muitos casos práticos, é aceitável

Implementação no Projeto

```
# Em analise_box_jenkins_sarima.py
def teste_normalidade_residuos(self):
    # Shapiro-Wilk
    shapiro_stat, shapiro_p = stats.shapiro(residuos_clean)

    # Jarque-Bera
    jb_stat, jb_p = stats.jarque_bera(residuos_clean)

    # Anderson-Darling
    anderson_result = stats.anderson(residuos_clean, dist='norm')

    return {
        'shapiro': shapiro_p > 0.05,
        'jarque_bera': jb_p > 0.05,
        'anderson': anderson_ok
    }
```

Referências

- **Shapiro, S. S., & Wilk, M. B.** (1965). An analysis of variance test for normality. *Biometrika*, 52(3/4), 591-611.
- **Jarque, C. M., & Bera, A. K.** (1987). A test for normality of observations and regression residuals. *International Statistical Review*, 55(2), 163-172.
- **Anderson, T. W., & Darling, D. A.** (1954). A test of goodness of fit. *Journal of the American Statistical Association*, 49(268), 765-769.

---

8. Teste de Heterocedasticidade (ARCH)

O que é?

O **Teste ARCH** (Autoregressive Conditional Heteroscedasticity) verifica se a **variância dos resíduos é constante** ao longo do tempo. Se a variância muda, dizemos que há heterocedasticidade.

Por que é necessário?

Suposição do Modelo SARIMA:

- Resíduos devem ser **homocedásticos** (variância constante)
- Se há heterocedasticidade:
  - ❌ Intervalos de confiança podem ser incorretos
  - ❌ Previsões podem ser menos confiáveis
  - ❌ Pode indicar necessidade de modelos GARCH

No Contexto de Estoque:

- Variância não constante pode indicar:
  - Períodos de maior volatilidade (ex: Black Friday)
  - Mudanças estruturais na série
  - Necessidade de modelagem adicional

Como funciona?

Hipóteses:

- **H_0**: Resíduos são homocedásticos (variância constante) ✅
- **H_1**: Resíduos são heterocedásticos (variância não constante) ❌

Teste LM (Lagrange Multiplier):

- Regride quadrados dos resíduos em seus valores defasados
- Se há correlação, há heterocedasticidade

Interpretação:

- Se **p-value > 0.05**: Homocedástico ✅
- Se **p-value ≤ 0.05**: Heterocedástico ❌

Solução se Heterocedástico:

- Usar modelos **GARCH** (Generalized ARCH)
- Transformar a série (log, diferença)
- Usar intervalos de confiança robustos

Implementação no Projeto

```python
# Em analise_box_jenkins_sarima.py
def teste_heterocedasticidade(self):
    resultado = het_arch(self.residuos.dropna(), maxlag=5)
    lm_pvalue = resultado[1]
    is_homocedastico = lm_pvalue > 0.05
    return is_homocedastico
```

Referências

- **Engle, R. F.** (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica*, 50(4), 987-1007.
- **Bollerslev, T.** (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3), 307-327.

✅ Ferramentas de Validação

9. Validação Cruzada Walk-Forward

O que é?

Walk-Forward é um método de validação específico para séries temporais que **respeita a ordem temporal** dos dados. Diferente de validação cruzada tradicional (que embaralha dados), walk-forward:

- Treina com dados do passado
- Testa em dados futuros
- Expande a janela de treino progressivamente

Por que é necessário?

Problema da Validação Tradicional:

- ❌ Embaralhar dados quebra a ordem temporal
- ❌ Usar dados futuros para prever o passado (data leakage)
- ❌ Não reflete como o modelo será usado na prática

Vantagens do Walk-Forward:

- ✅ Respeita ordem temporal
- ✅ Simula uso real (treina com passado, prevê futuro)
- ✅ Testa estabilidade do modelo ao longo do tempo
- ✅ Método correto para séries temporais

Como funciona?

Exemplo Prático:

```
Dados: 12 meses (M1 a M12)
```

```
Fold 1:
  Treino: M1-M6 (6 meses)
  Teste: M7 (1 mês)

Fold 2:
  Treino: M1-M7 (7 meses) ← Expandiu!
  Teste: M8 (1 mês)

Fold 3:
  Treino: M1-M8 (8 meses) ← Expandiu!
  Teste: M9 (1 mês)

... e assim por diante
```

Métricas Calculadas:

- **MAE** (Mean Absolute Error) por fold
- **RMSE** (Root Mean Squared Error) por fold
- **MAPE** (Mean Absolute Percentage Error) por fold

Análise de Estabilidade:

- Se métricas variam muito entre folds → Modelo instável
- Se métricas são consistentes → Modelo estável ✅

Implementação no Projeto

```python
# Em validacao_walk_forward_sarima.py
class ValidacaoWalkForward:
    def executar_validacao(self):
        n_treino_inicial = int(len(serie) * 0.7)
        n_teste = int(len(serie) * 0.1)

        pos_treino_fim = n_treino_inicial
        while pos_treino_fim + n_teste <= len(serie):
            # Treina com dados até pos_treino_fim
            serie_treino = serie.iloc[:pos_treino_fim]
            # Testa nos próximos n_teste períodos
            serie_teste = serie.iloc[pos_treino_fim:pos_treino_fim + n_te

            # Treina modelo e calcula métricas
            modelo = auto_arima(serie_treino)
            previsao = modelo.predict(n_periods=len(serie_teste))
            mae = mean_absolute_error(serie_teste, previsao)
```

```
        # Avança para próximo fold
        pos_treino_fim += passo
```

Referências

- **Bergmeir, C., & Benítez, J. M.** (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191, 192-213.
- **Hyndman, R. J., & Athanasopoulos, G.** (2021). *Forecasting: Principles and Practice* (3rd ed.). OTexts. (Cap. 3.4)

🛠️ Ferramentas de Tratamento de Dados

10. Tratamento de Outliers

O que é?

Outliers são valores que se desviam significativamente do padrão normal da série. Em e-commerce de brinquedos, podem ocorrer devido a:

- **Eventos especiais**: Dia das Crianças, Black Friday, Natal
- **Promoções**: Descontos que aumentam demanda
- **Erros de dados**: Registros incorretos

Por que é necessário?

Problemas causados por outliers:

- ❌ Distorcem estimativas dos parâmetros do modelo
- ❌ Afetam previsões futuras
- ❌ Podem fazer modelo "aprender" padrões incorretos
- ❌ Aumentam erro de previsão

No Contexto de Estoque:

- Picos de demanda em eventos especiais podem:
    - Fazer modelo superestimar demanda futura
    - Ou subestimar se não tratados corretamente

Como funciona?

Métodos Implementados:

1. Método IQR (Interquartile Range):

```
Q1 = Percentil 25
Q3 = Percentil 75
IQR = Q3 - Q1

Outlier se: valor < Q1 - 1.5×IQR  OU  valor > Q3 + 1.5×IQR
```

2. Método Z-Score:

```
z = (valor - média) / desvio_padrão

Outlier se: |z| > 3 (3 desvios padrão)
```

Tratamento:

- **Remover**: Substitui por NaN (perde informação)
- **Substituir por Mediana**: Preserva estrutura, remove pico
- **Suavizar**: Substitui por média móvel (preserva informação, suaviza pico)

Recomendação:

- **Suavização** é preferível para séries temporais
- Preserva informação temporal
- Não cria gaps na série

Implementação no Projeto

```
# Em tratamento_outliers_sarima.py
class TratamentoOutliers:
    def identificar_outliers_iqr(self, fator=1.5):
        Q1 = self.serie.quantile(0.25)
        Q3 = self.serie.quantile(0.75)
        IQR = Q3 - Q1
        outliers = (self.serie < Q1 - fator*IQR) | (self.serie > Q3 + fat
        return outliers

    def substituir_outliers_suavizacao(self, janela=5):
        media_movel = self.serie.rolling(window=janela, center=True).mear
        serie_tratada = self.serie.copy()
        serie_tratada[outliers] = media_movel[outliers]
        return serie_tratada
```

Referências

- **Tukey, J. W.** (1977). *Exploratory Data Analysis*. Addison-Wesley.
- **Barnett, V., & Lewis, T.** (1994). *Outliers in Statistical Data* (3rd ed.). Wiley.

📊 Métricas de Avaliação

11. MAE (Mean Absolute Error)

O que é?

MAE mede o **erro médio absoluto** entre valores reais e previstos.

Fórmula:

```
MAE = (1/n) × Σ |y_real - y_previsto|
```

Por que usar?

- ✅ **Fácil de interpretar**: Erro médio em unidades da variável
- ✅ **Robusto a outliers**: Não é muito afetado por valores extremos
- ✅ **Escala natural**: Mesma unidade dos dados (ex: unidades de estoque)

Interpretação

- **MAE = 5**: Erro médio de 5 unidades
- **Menor MAE = Melhor modelo**

---

12. RMSE (Root Mean Squared Error)

O que é?

RMSE mede o **erro quadrático médio**, dando mais peso a erros grandes.

Fórmula:

```
RMSE = √[(1/n) × Σ(y_real - y_previsto)²]
```

Por que usar?

- ✅ **Penaliza erros grandes**: Mais sensível a outliers
- ✅ **Amplamente usado**: Padrão na literatura
- ✅ **Propriedades matemáticas**: Facilita otimização

Interpretação

- **RMSE ≥ MAE**: Sempre (por propriedade matemática)
- **Diferença grande**: Indica presença de erros grandes (outliers)
- **Menor RMSE = Melhor modelo**

---

13. MAPE (Mean Absolute Percentage Error)

O que é?

MAPE mede o **erro percentual médio**, útil para comparar modelos em diferentes escalas.

Fórmula:

```
MAPE = (1/n) × Σ |y_real - y_previsto| / |y_real| × 100
```

Por que usar?

- ✅ **Comparável entre SKUs**: Normalizado por escala
- ✅ **Fácil de comunicar**: "Erro de 10%" é mais intuitivo
- ✅ **Útil para negócio**: Stakeholders entendem percentuais

Interpretação

- **MAPE < 10%**: Excelente
- **MAPE 10-20%**: Bom
- **MAPE 20-50%**: Razoável
- **MAPE > 50%**: Precisa melhorar

Limitação:

- Problemas quando valores reais são próximos de zero (divisão por zero)

📚 Referências Bibliográficas Completas

Livros Fundamentais

1. **Box, G. E. P., & Jenkins, G. M.** (1976). *Time Series Analysis: Forecasting and Control*. Holden-Day.
    - **Por que**: Livro clássico que estabeleceu a metodologia Box-Jenkins

2. **Hyndman, R. J., & Athanasopoulos, G.** (2021). *Forecasting: Principles and Practice* (3rd ed.). OTexts.
    - **Por que**: Livro moderno, gratuito, com exemplos práticos em R e Python
    - **Disponível em**: https://otexts.com/fpp3/
3. **Chatfield, C.** (2016). *The Analysis of Time Series: An Introduction* (7th ed.). CRC Press.
    - **Por que**: Introdução acessível a séries temporais

Artigos Científicos

4. **Dickey, D. A., & Fuller, W. A.** (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American Statistical Association*, 74(366), 427-431.
    - **Sobre**: Teste de estacionariedade ADF
5. **Ljung, G. M., & Box, G. E. P.** (1978). On a measure of lack of fit in time series models. *Biometrika*, 65(2), 297-303.
    - **Sobre**: Teste de Ljung-Box para resíduos
6. **Engle, R. F.** (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica*, 50(4), 987-1007.
    - **Sobre**: Teste de heterocedasticidade ARCH
7. **Akaike, H.** (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6), 716-723.
    - **Sobre**: Critério de Informação de Akaike (AIC)
8. **Hyndman, R. J., & Khandakar, Y.** (2008). Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software*, 27(3), 1-22.
    - **Sobre**: Auto-ARIMA e seleção automática de parâmetros
9. **Bergmeir, C., & Benítez, J. M.** (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191, 192-213.
    - **Sobre**: Validação cruzada para séries temporais

Testes Estatísticos

10. **Shapiro, S. S., & Wilk, M. B.** (1965). An analysis of variance test for normality. *Biometrika*, 52(3/4), 591-611.

- **Sobre**: Teste de normalidade Shapiro-Wilk

11. **Jarque, C. M., & Bera, A. K.** (1987). A test for normality of observations and regression residuals. *International Statistical Review*, 55(2), 163-172.

    - **Sobre**: Teste de normalidade Jarque-Bera

12. **Anderson, T. W., & Darling, D. A.** (1954). A test of goodness of fit. *Journal of the American Statistical Association*, 49(268), 765-769.

    - **Sobre**: Teste de normalidade Anderson-Darling

Tratamento de Dados

13. **Tukey, J. W.** (1977). *Exploratory Data Analysis*. Addison-Wesley.

    - **Sobre**: Método IQR para detecção de outliers

14. **Barnett, V., & Lewis, T.** (1994). *Outliers in Statistical Data* (3rd ed.). Wiley.

    - **Sobre**: Tratamento de outliers em dados estatísticos

---

🎓 Como Explicar em Apresentações

Estrutura de Explicação

Para cada ferramenta, explique seguindo esta estrutura:

1. **O QUE É**: Definição simples e clara
2. **POR QUE USAR**: Justificativa teórica e prática
3. **COMO FUNCIONA**: Mecanismo básico (sem entrar em detalhes matemáticos demais)
4. **RESULTADO**: O que esperamos obter
5. **INTERPRETAÇÃO**: Como interpretar os resultados

Exemplo: Teste ADF

"O que é?"
"O Teste ADF verifica se nossa série de estoque é estacionária, ou seja, se a média e variância são constantes ao longo do tempo."

"Por que usar?"
"Modelos SARIMA requerem séries estacionárias. Se a série não for estacionária, o modelo não captura corretamente os padrões e as previsões podem ser enviesadas."

"Como funciona?"

"O teste compara duas hipóteses: série é estacionária ou não. Se o p-value for menor que 0.05, concluímos que a série é estacionária. Caso contrário, aplicamos diferenciação, que o auto_arima faz automaticamente."

"Resultado"

"Obtemos um p-value que nos diz se precisamos diferenciar a série ou não."

"Interpretação"

"P-value < 0.05: série estacionária, podemos prosseguir. P-value ≥ 0.05: série não estacionária, diferenciação necessária."

✅ Checklist de Defesa

Use este checklist para garantir que você pode explicar cada ferramenta:

- [ ] Entendo o que cada ferramenta faz
- [ ] Sei explicar por que é necessária
- [ ] Consigo interpretar os resultados
- [ ] Sei quando usar cada ferramenta
- [ ] Conheço as limitações de cada uma
- [ ] Tenho referências para justificar

Documento criado para TCC MBA Data Science & Analytics - 2024