Aula 06. Testes Qui-Quadrado e ANOVA

Estatística Inferencial

MBA CDIA

ENAP - Escola Nacional de Administração Pública

2025

Agenda da Aula

1. **Teste Qui-Quadrado (χ^2)**
   - Teste de Aderência (Goodness of Fit)
   - Teste de Independência (Tabelas de Contingência)
2. **ANOVA de Um Fator**
   - Comparação de múltiplas médias
   - Decomposição da variabilidade
3. **Testes de Acompanhamento**
   - Comparações múltiplas de Tukey
   - Contrastes planejados

Teste Qui-Quadrado: Introdução

Objetivo: Testar hipóteses sobre variáveis categóricas

Dois tipos principais:

- **Teste de Aderência:** Verifica se uma distribuição observada segue um padrão específico
- **Teste de Independência:** Verifica se duas variáveis categóricas são independentes

Estatística de teste:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

onde O_i = frequência observada, E_i = frequência esperada

Exemplo 1: Teste de Aderência - Programa Auxílio Emergencial

A distribuição regional do Auxílio Emergencial deveria seguir a população: 40% Sudeste, 30% Nordeste, 30% demais regiões.
Em uma amostra de 1000 beneficiários, observamos:

Região	Sudeste	Nordeste	Demais	Total
Observado	380	350	270	1000
Esperado	400	300	300	1000

Hipóteses:

- H_0: As proporções seguem 0.4:0.3:0.3
- H_1: As proporções não seguem essa distribuição

Cálculo da Estatística Qui-Quadrado

Passo 1: Calcular $(O_i - E_i)^2/E_i$ para cada categoria:

Região	Sudeste	Nordeste	Demais
Observado (O)	380	350	270
Esperado (E)	400	300	300
$O - E$	-20	50	-30
$(O - E)^2$	400	2500	900
$(O - E)^2/E$	1.0	8.33	3.0

Passo 2: Somar todos os componentes:

$$\chi^2 = 1.0 + 8.33 + 3.0 = 12.33$$

Passo 3: Comparar com valor crítico ($gl = k - 1 = 2$):

- $\chi^2_{0.05,2} = 5.991$
- Como $12.33 > 5.991$, rejeitamos H_0
- **Conclusão:** A distribuição não segue o padrão populacional esperado

Exemplo 2: Atendimentos em CRAS por Dia da Semana

Os atendimentos nos Centros de Referência de Assistência Social são igualmente distribuídos durante a semana?
Amostra de 700 atendimentos:

Dia	Seg	Ter	Qua	Qui	Sex
Observado	120	155	180	140	105
Esperado	140	140	140	140	140

Hipóteses:

- H_0: $p_1 = p_2 = p_3 = p_4 = p_5 = 1/5$
- H_1: Pelo menos uma proporção é diferente

$$\chi^2 = 18.21, \quad gl = 4, \quad p\text{-valor} < 0.05$$

Conclusão: Rejeitamos H_0 - os atendimentos não são uniformemente distribuídos

Teste de Independência: Programas Sociais e Escolaridade

Objetivo: Testar se a participação em programas sociais é independente do nível de escolaridade
Exemplo: Beneficiários do Bolsa Família

Escolaridade/Programa	Participa	Não participa	Total
Fundamental	180	120	300
Médio	90	210	300
Superior	30	170	200
Total	300	500	800

Hipóteses:

- H_0: Participação e escolaridade são independentes
- H_1: Existe relação entre participação e escolaridade

Valor esperado: $E_{ij} = \frac{\text{Total linha}_i \times \text{Total coluna}_j}{\text{Total geral}}$

Cálculo do Qui-Quadrado para Tabelas de Contingência

Valores esperados:

Escolaridade/Programa	Participa	Não participa
Fundamental	$\frac{300\times300}{800} = 112.5$	$\frac{300\times500}{800} = 187.5$
Médio	$\frac{300\times300}{800} = 112.5$	$\frac{300\times500}{800} = 187.5$
Superior	$\frac{200\times300}{800} = 75$	$\frac{200\times500}{800} = 125$

Cálculo:

$$\chi^2 = \frac{(180 - 112.5)^2}{112.5} + \frac{(120 - 187.5)^2}{187.5} + \ldots = 102.4$$

Graus de liberdade: $gl = (r - 1)(c - 1) = (3 - 1)(2 - 1) = 2$

Conclusão: $\chi^2 = 102.4 > \chi^2_{0.05,2} = 5.991 \rightarrow$ Há forte associação entre escolaridade e participação no programa

Objetivo: Comparar médias de 3 ou mais grupos

Perguntas fundamentais:

1. Existem diferenças significativas entre os grupos?
2. Quais são essas diferenças?

Hipóteses:

- $H_0 : \mu_1 = \mu_2 = ... = \mu_k$ (todas as médias são iguais)
- H_1: Pelo menos uma média é diferente

Princípio: Decompor a variabilidade total em:

- Variabilidade **entre grupos** (SSG)
- Variabilidade **dentro dos grupos** (SSE)

Decomposição da Variabilidade

Soma de Quadrados Total (SST): $SST = \sum_{i,j}(y_{ij} - \bar{y})^2$

Soma de Quadrados Entre Grupos (SSG): $SSG = \sum_i n_i(\bar{y}_i - \bar{y})^2$

Soma de Quadrados Dentro dos Grupos (SSE): $SSE = \sum_{i,j}(y_{ij} - \bar{y}_i)^2$

Relação fundamental: $SST = SSG + SSE$

Estatística F: $F = \frac{MSG}{MSE} = \frac{SSG/(k-1)}{SSE/(N-k)}$

Exemplo ANOVA: Preferências de Bebidas

Comparação do gasto ($) com diferentes tipos de bebidas:

Pessoa	Bebida (Gasto)
Zane	Cerveja (20)
Justin	Cerveja (15)
Blake	Dr. Pepper (10)
America	Limonada (8)
Paige	Água (2)
Addie	Sprite (5)

Grupos:

- G1 (Cerveja): 20, 15 $\rightarrow \bar{x}_1 = 17.5$
- G2 (Refrigerante): 10, 5 $\rightarrow \bar{x}_2 = 7.5$
- G3 (Outros): 8, 2 $\rightarrow \bar{x}_3 = 5$

Média geral: $\bar{y} = 10$

Tabela ANOVA

Fonte	**GL**	**SQ**	**QM**	**F**	**p-valor**
Entre grupos	$k-1$	SSG	MSG = SSG/(k-1)	F = MSG/MSE	$P(F > F_{obs})$
Dentro dos grupos	$N-k$	SSE	MSE = SSE/(N-k)		
Total	$N-1$	SST			

Decisão:

- Se p-valor $< \alpha \to$ Rejeitar H_0
- Se p-valor $\geq \alpha \to$ Não rejeitar H_0

Interpretação:

- Rejeitar H_0: Existe diferença significativa entre pelo menos duas médias
- Não rejeitar H_0: Não há evidência de diferença entre as médias

Problema: ANOVA significativa indica diferenças, mas não especifica quais grupos diferem
Solução: Testes de acompanhamento (post-hoc)
Método de Tukey (HSD - Honestly Significant Difference):

- Controla o erro tipo I global
- Compara todos os pares de médias simultaneamente
- Intervalo de confiança simultâneo:

$$(\bar{x}_i - \bar{x}_j) \pm q_\alpha \cdot s_p \sqrt{\frac{1}{n_i} + \frac{1}{n_j}}$$

onde q_α é o valor crítico da distribuição Studentized Range

Exemplo: Comparações Múltiplas de Tukey

Dados: Tempo de espera (dias) para atendimento médico em diferentes modalidades do SUS

Modalidade	n	Média	DP
1. UBS Tradicional	25	45.2	12.3
2. UBS com Telemedicina	22	32.1	8.7
3. Clínica da Família	28	18.5	6.2

ANOVA: $F = 85.4$, $p < 0.001$ (significativo)
Comparações de Tukey (95% confiança):

- UBS Trad. vs UBS Telem.: $(7.2, 19.0) \rightarrow$ Significativo
- UBS Trad. vs Clínica Fam.: $(21.1, 32.3) \rightarrow$ Significativo
- UBS Telem. vs Clínica Fam.: $(8.3, 19.1) \rightarrow$ Significativo

Contrastes Planejados

Definição: Comparações específicas definidas antes da coleta de dados
Contraste populacional:

$$L = c_1\mu_1 + c_2\mu_2 + \ldots + c_k\mu_k$$

onde $\sum c_i = 0$
Exemplo: Cores que atraem besouros

- Cores frias (azul, branco) vs. cores quentes (verde, amarelo)
- $L = \frac{\mu_{verde}+\mu_{amarelo}}{2} - \frac{\mu_{azul}+\mu_{branco}}{2}$
- Coeficientes: $c_{azul} = -1/2$, $c_{branco} = -1/2$, $c_{verde} = 1/2$, $c_{amarelo} = 1/2$

Teste:

$$t = \frac{\hat{L}}{SE_{\hat{L}}}$$

Condições e Pressupostos

Para o Teste Qui-Quadrado:

- Todas as frequências esperadas ≥ 1
- No máximo 20% das frequências esperadas < 5
- Para tabelas 2×2: todas as frequências esperadas ≥ 5

Para ANOVA:

- Amostras aleatórias independentes
- Populações normalmente distribuídas
- Variâncias populacionais iguais (homocedasticidade)
- Regra prática: maior DP / menor DP < 2

Quando os pressupostos são violados:

- Qui-quadrado: Teste exato de Fisher (tabelas pequenas)
- ANOVA: Teste de Kruskal-Wallis (não-paramétrico)

Implementação em Python: Teste Qui-Quadrado

```
import numpy as np
import pandas as pd
from scipy import stats
import matplotlib.pyplot as plt

# Teste de Aderencia - Distribuicao do Auxilio Emergencial
observado = np.array([380, 350, 270])
esperado = np.array([400, 300, 300])

# Teste qui-quadrado
chi2, p_valor = stats.chisquare(observado, esperado)
print(f"Estatistica qui-quadrado: {chi2:.3f}")
print(f"p-valor: {p_valor:.3f}")
print(f"Graus de liberdade: {len(observado) - 1}")
```

Implementação em Python: Teste de Independência

```python
# Teste de Independencia - Bolsa Familia e Escolaridade
tabela = np.array([[180, 120],
                   [90, 210],
                   [30, 170]])

# Teste qui-quadrado para independencia
chi2, p_valor, gl, esperado = stats.chi2_contingency(tabela)

print("Tabela de Contingencia Observada:")
print(pd.DataFrame(tabela,
                   index=['Fundamental', 'Medio', 'Superior'],
                   columns=['Participa', 'Nao Participa']))
print("\nTabela de Valores Esperados:")
print(pd.DataFrame(esperado,
                   index=['Fundamental', 'Medio', 'Superior'],
                   columns=['Participa', 'Nao Participa']))
print(f"\nEstatistica qui-quadrado: {chi2:.3f}")
print(f"Graus de liberdade: {gl}")
print(f"p-valor: {p_valor:.3f}")
```

Implementação em Python: ANOVA de Um Fator

```python
# ANOVA de um fator
# Exemplo: Dieta de cafeteria em ratos
grupo1 = np.random.normal(605.6, 49.64, 19)  # Racao apenas
grupo2 = np.random.normal(657.3, 50.68, 16)  # Racao + cafeteria restrita
grupo3 = np.random.normal(691.1, 63.41, 15)  # Racao + cafeteria livre

# Teste ANOVA
f_stat, p_valor = stats.f_oneway(grupo1, grupo2, grupo3)
print(f"Estatistica F: {f_stat:.3f}")
print(f"p-valor: {p_valor:.4f}")

# Criando DataFrame para visualizacao
dados = pd.DataFrame({
    'Peso': np.concatenate([grupo1, grupo2, grupo3]),
    'Grupo': ['Racao']*19 + ['Restrita']*16 + ['Livre']*15
})

# Boxplot
plt.figure(figsize=(10, 6))
dados.boxplot(column='Peso', by='Grupo', grid=False)
plt.suptitle('Peso dos Ratos por Tipo de Dieta')
plt.xlabel('Grupo de Dieta')
plt.ylabel('Peso (g)')
plt.title('')
plt.show()

# Medias e intervalos de confianca
print("\nEstatisticas por grupo:")
print(dados.groupby('Grupo')['Peso'].agg(['mean', 'std', 'count']))
```

Implementação em Python: Comparações Múltiplas de Tukey

```python
from statsmodels.stats.multicomp import pairwise_tukeyhsd

# Teste de Tukey HSD
tukey = pairwise_tukeyhsd(endog=dados['Peso'],
                          groups=dados['Grupo'], alpha=0.05)
print(tukey)

# Visualizacao das comparacoes
fig = tukey.plot_simultaneous(figsize=(10, 6))
plt.title('Intervalos de Confianca Simultaneos de Tukey (95%)')
plt.show()

# Criar matriz de comparacoes
import itertools
grupos = dados['Grupo'].unique()
comparacoes = list(itertools.combinations(grupos, 2))

# Calcular diferencas entre medias
for g1, g2 in comparacoes:
    media1 = dados[dados['Grupo'] == g1]['Peso'].mean()
    media2 = dados[dados['Grupo'] == g2]['Peso'].mean()
    diff = media2 - media1
    print(f"{g2} - {g1}: {diff:.2f}")
```

Exercício Prático 1: Teste Qui-Quadrado

Contexto: Uma pesquisa sobre efetividade de campanhas de saúde pública em três regiões do Brasil

Região/Resultado	Efetiva	Parcial	Ineficaz	Total
Norte	45	35	20	100
Sudeste	60	55	35	150
Sul	30	40	30	100
Total	135	130	85	350

Tarefas:

1. Teste se há associação entre região e efetividade da campanha ($\alpha = 0.05$)
2. Calcule os valores esperados
3. Interprete os resultados no contexto de políticas públicas

Exercício Prático 2: ANOVA e Comparações Múltiplas

Contexto: Comparação de três métodos de ensino de estatística

Arquivo: `metodos_ensino.csv`

- Método A: Tradicional (n=25)
- Método B: Com tecnologia (n=25)
- Método C: Aprendizagem ativa (n=25)

Tarefas:

1. Realize ANOVA para testar diferenças entre os métodos
2. Se significativo, aplique teste de Tukey
3. Crie visualizações apropriadas
4. Escreva um relatório com conclusões práticas

Solução Completa: Análise de Métodos de Ensino

```
import pandas as pd
import numpy as np
from scipy import stats
from statsmodels.stats.multicomp import pairwise_tukeyhsd
import matplotlib.pyplot as plt
import seaborn as sns

# Simulando dados (em pratica, voce carregaria o CSV)
np.random.seed(42)
metodo_a = np.random.normal(70, 10, 25)  # Tradicional
metodo_b = np.random.normal(78, 9, 25)   # Com tecnologia
metodo_c = np.random.normal(82, 8, 25)   # Aprendizagem ativa

# Criar DataFrame
dados = pd.DataFrame({
    'Nota': np.concatenate([metodo_a, metodo_b, metodo_c]),
    'Metodo': ['A']*25 + ['B']*25 + ['C']*25
})

# 1. ANOVA
f_stat, p_valor = stats.f_oneway(metodo_a, metodo_b, metodo_c)
print(f"ANOVA - F: {f_stat:.3f}, p-valor: {p_valor:.4f}")

# 2. Teste de Tukey
tukey = pairwise_tukeyhsd(dados['Nota'], dados['Metodo'])
print("\n", tukey)
```

Exemplo: Contrastes Planejados em Python

```
import numpy as np
import pandas as pd
from scipy import stats
from statsmodels.formula.api import ols
from statsmodels.stats.anova import anova_lm

# Exemplo: Cores que atraem besouros
azul = [14.8, 12.5, 16.2, 13.8, 15.1, 17.3]
verde = [31.2, 28.5, 33.2, 30.8, 32.1, 31.3]
branco = [16.2, 14.5, 17.8, 15.3, 16.9, 15.5]
amarelo = [47.2, 45.2, 49.1, 46.3, 48.2, 46.1]

# Criar DataFrame
dados_cores = pd.DataFrame({
    'besouros': azul + verde + branco + amarelo,
    'cor': ['azul']*6 + ['verde']*6 + ['branco']*6 + ['amarelo']*6
})

# ANOVA
modelo = ols('besouros ~ C(cor)', data=dados_cores).fit()
anova_resultado = anova_lm(modelo)
print("ANOVA:")
print(anova_resultado)

# Extrair MSE e graus de liberdade para os contrastes
mse = anova_resultado['mean_sq'][1]  # MSE (erro)
df_erro = anova_resultado['df'][1]   # Graus de liberdade do erro

print(f"\nMSE: {mse:.3f}")
print(f"GL erro: {df_erro}")
```

Exemplo: Contrastes Planejados em Python

```
# Definir contrastes
def calcular_contraste(dados, contraste, mse, df_erro):
    medias = dados.groupby('cor')['besouros'].mean()
    tamanhos = dados.groupby('cor')['besouros'].count()

    L_hat = sum(contraste[cor] * medias[cor] for cor in contraste)
    se_L = np.sqrt(mse * sum(contraste[cor]**2 / tamanhos[cor]
                        for cor in contraste))
    t_stat = L_hat / se_L
    p_valor = 2 * (1 - stats.t.cdf(abs(t_stat), df_erro))

    return L_hat, se_L, t_stat, p_valor

# Contraste 1: Cores frias vs quentes
contraste_frias_quentes = {'azul': -0.5, 'branco': -0.5,
                           'verde': 0.5, 'amarelo': 0.5}

L1, se1, t1, p1 = calcular_contraste(dados_cores, contraste_frias_quentes, mse, df_erro)

print("\n" + "="*50)
print("CONTRASTE 1: Cores frias vs quentes")
print("="*50)
print(f"Coeficientes: {contraste_frias_quentes}")
print(f"L_hat: {L1:.3f}")
print(f"Erro padrão: {se1:.3f}")
print(f"Estatística t: {t1:.3f}")
print(f"p-valor: {p1:.4f}")
print(f"Resultado: {'Significativo' if p1 < 0.05 else 'Não significativo'} ( = 0.05)")
print(f"Interpretação: Cores quentes atraem {'mais' if L1 > 0 else 'menos'} besouros que cores frias")
```

Exemplo: Contrastes Planejados em Python

```
# Contraste 2: Azul vs Branco (dentro das frias)
contraste_azul_branco = {'azul': 1, 'branco': -1, 'verde': 0, 'amarelo': 0}

L2, se2, t2, p2 = calcular_contraste(dados_cores, contraste_azul_branco, mse, df_erro)

print("\n" + "="*50)
print("CONTRASTE 2: Azul vs Branco")
print("="*50)
print(f"Coeficientes: {contraste_azul_branco}")
print(f"L_hat: {L2:.3f}")
print(f"Erro padrão: {se2:.3f}")
print(f"Estatística t: {t2:.3f}")
print(f"p-valor: {p2:.4f}")
print(f"Resultado: {'Significativo' if p2 < 0.05 else 'Não significativo'} ( = 0.05)")

# Contraste 3: Verde vs Amarelo (dentro das quentes)
contraste_verde_amarelo = {'azul': 0, 'branco': 0, 'verde': 1, 'amarelo': -1}

L3, se3, t3, p3 = calcular_contraste(dados_cores, contraste_verde_amarelo, mse, df_erro)

print("\n" + "="*50)
print("CONTRASTE 3: Verde vs Amarelo")
print("="*50)
print(f"Coeficientes: {contraste_verde_amarelo}")
print(f"L_hat: {L3:.3f}")
print(f"Erro padrão: {se3:.3f}")
print(f"Estatística t: {t3:.3f}")
print(f"p-valor: {p3:.4f}")
print(f"Resultado: {'Significativo' if p3 < 0.05 else 'Não significativo'} ( = 0.05)")
```

Exemplo: Contrastes Planejados em Python

```
# Resumo dos resultados
print("\n" + "="*50)
print("RESUMO DOS CONTRASTES")
print("="*50)
contrastes_resumo = [
    ("Frias vs Quentes", L1, t1, p1),
    ("Azul vs Branco", L2, t2, p2),
    ("Verde vs Amarelo", L3, t3, p3)
]

for nome, L, t, p in contrastes_resumo:
    sig = "Sim" if p < 0.05 else "Não"
    print(f"{nome:15} | L_hat: {L:6.2f} | t: {t:6.2f} | p: {p:.4f} | Sig: {sig}")

# Correção para múltiplos testes (Bonferroni)
alpha_bonferroni = 0.05 / 3
print(f"\nCom correção de Bonferroni ( = {alpha_bonferroni:.3f}):")
for nome, L, t, p in contrastes_resumo:
    sig = "Sim" if p < alpha_bonferroni else "Não"
    print(f"{nome:15} | Significativo: {sig}")
```

Resumo: Testes para Dados Categóricos e Múltiplos Grupos

Teste Qui-Quadrado:

- Aderência: $\chi^2 = \sum \frac{(O-E)^2}{E}$
- GL: $k-1$ (aderência)
- GL: $(r-1)(c-1)$ (independência)
- Python: `stats.chi2_contingency()`

ANOVA e Acompanhamento:

- Estatística: $F = \frac{MSG}{MSE}$
- GL: $(k-1, N-k)$
- Tukey: Comparações múltiplas
- Python: `stats.f_oneway()`

Pontos-chave:

- Verificar pressupostos antes de aplicar os testes
- ANOVA detecta diferenças, Tukey identifica quais
- Qui-quadrado para variáveis categóricas, ANOVA para quantitativas
- Sempre interpretar resultados no contexto do problema

Próxima Aula

Aula 07: Regressão Linear

- Regressão linear simples
- Regressão linear múltipla
- Diagnóstico de modelos
- Predição e interpretação

Obrigado!

Dúvidas?