Análise de Dispêndios e Quadrantes de Decisão - Lei do Bem

## 1. Configuração e Conexão com Banco de Dados

import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
from sklearn.linear\_model import LogisticRegression  
from sklearn.preprocessing import StandardScaler  
from sklearn.metrics import classification\_report, confusion\_matrix  
from scipy import stats  
import warnings  
warnings.filterwarnings('ignore')  
  
# Configurações de visualização  
plt.rcParams['figure.figsize'] = (14, 8)  
plt.rcParams['font.size'] = 11  
sns.set\_style("whitegrid")  
  
# Carregar dados do CSV  
print("Carregando dados do CSV...")  
df = pd.read\_csv('./csv\_longo/perguntas2508.csv', sep=';')  
  
print(f"\n{'='\*73}")  
print("DADOS CARREGADOS")  
print(f"{'='\*73}")  
print(f"Total de registros: {len(df):,}")  
print(f"Total de colunas: {len(df.columns)}")  
print(f"\nColunas disponíveis:")  
print(df.columns.tolist())

Carregando dados do CSV...  
  
=========================================================================  
DADOS CARREGADOS  
=========================================================================  
Total de registros: 2,831,160  
Total de colunas: 29  
  
Colunas disponíveis:  
['id\_empresa\_ano', 'ano\_referencia', 'setor', 'nrcnpj', 'norazaosocial', 'iddadoanaliseprojeto', 'do\_resultado\_analise', 'p\_resultado\_analise', 'vlreceitaliquida', 'valortotalproj', 'valordoutorproj', 'quantdoutorproj', 'valormestreproj', 'quantmestreproj', 'valorrhproj', 'quantrhproj', 'valorictproj', 'valoruniversidadesproj', 'valorservterceirosproj', 'percservterceirosproj', 'quantpessoalempresa', 'percpessoalalocadoempresa', 'quantdoutorempresa', 'quantmestreempresa', 'valortotalempresa', 'valorserterceirosempresa', 'percservterceirosempresa', 'valorbensequipempresa', 'percbensequipempresa']

## 2. Pré-processamento e Criação dos Quadrantes

Classificação dos quadrantes:

Q1: Ministério Recomenda, Pesquisador Recomenda  
Q2: Ministério Não Recomenda, Pesquisador Recomenda  
Q3: Ministério Recomenda, Pesquisador Não Recomenda  
Q4: Ministério Não Recomenda, Pesquisador Não Recomenda

# Agregar por projeto (pegar primeira ocorrência já que os dados estão agregados)  
df\_projetos = df.groupby('iddadoanaliseprojeto').first().reset\_index()  
  
# Filtrar anos relevantes (2018-2023)  
df\_projetos = df\_projetos[df\_projetos['ano\_referencia'].isin([2018, 2019, 2020, 2021, 2022, 2023])]  
  
def classificar\_quadrante(row):  
 """  
 Classifica projetos nos 4 quadrantes:  
 S/S: Ministério Recomenda, Pesquisador Recomenda  
 S/N: Ministério Não Recomenda, Pesquisador Recomenda  
 N/S: Ministério Recomenda, Pesquisador Não Recomenda  
 N/N: Ministério Não Recomenda, Pesquisador Não Recomenda  
 """  
 do\_result = str(row['do\_resultado\_analise']).strip().lower() if pd.notna(row['do\_resultado\_analise']) else ''  
 p\_result = str(row['p\_resultado\_analise']).strip().lower() if pd.notna(row['p\_resultado\_analise']) else ''  
   
 if not do\_result or not p\_result:  
 return np.nan  
   
 # DO = Pesquisador, P = Ministério  
 do\_rec = 'recomendado' in do\_result and 'não' not in do\_result  
 p\_rec = 'recomendado' in p\_result and 'não' not in p\_result  
   
 if p\_rec and do\_rec:  
 return 'S/S' # Ambos recomendam  
 elif not p\_rec and do\_rec:  
 return 'S/N' # Ministério não, Pesquisador sim  
 elif p\_rec and not do\_rec:  
 return 'N/S' # Ministério sim, Pesquisador não  
 else:  
 return 'N/N' # Ambos não recomendam  
  
# Criar variáveis binárias para análise  
df\_projetos['quadrante'] = df\_projetos.apply(classificar\_quadrante, axis=1)  
df\_projetos['ministerio\_aprova'] = df\_projetos['p\_resultado\_analise'].str.contains('Recomendado', case=False, na=False) & \  
 ~df\_projetos['p\_resultado\_analise'].str.contains('Não', case=False, na=False)  
df\_projetos['pesquisador\_aprova'] = df\_projetos['do\_resultado\_analise'].str.contains('Recomendado', case=False, na=False) & \  
 ~df\_projetos['do\_resultado\_analise'].str.contains('Não', case=False, na=False)  
  
# Remover NaN dos quadrantes  
df\_analise = df\_projetos.dropna(subset=['quadrante']).copy()  
  
print(f"\n{'='\*73}")  
print("DISTRIBUIÇÃO DOS QUADRANTES")  
print(f"{'='\*73}")  
dist = df\_analise['quadrante'].value\_counts(normalize=True) \* 100  
for q in ['S/S', 'S/N', 'N/S', 'N/N']:  
 if q in dist.index:  
 print(f"{q}: {dist[q]:.1f}% ({df\_analise['quadrante'].value\_counts()[q]} projetos)")

=========================================================================  
DISTRIBUIÇÃO DOS QUADRANTES  
=========================================================================  
S/S: 53.8% (36471 projetos)  
S/N: 7.5% (5104 projetos)  
N/S: 8.7% (5914 projetos)  
N/N: 29.9% (20294 projetos)

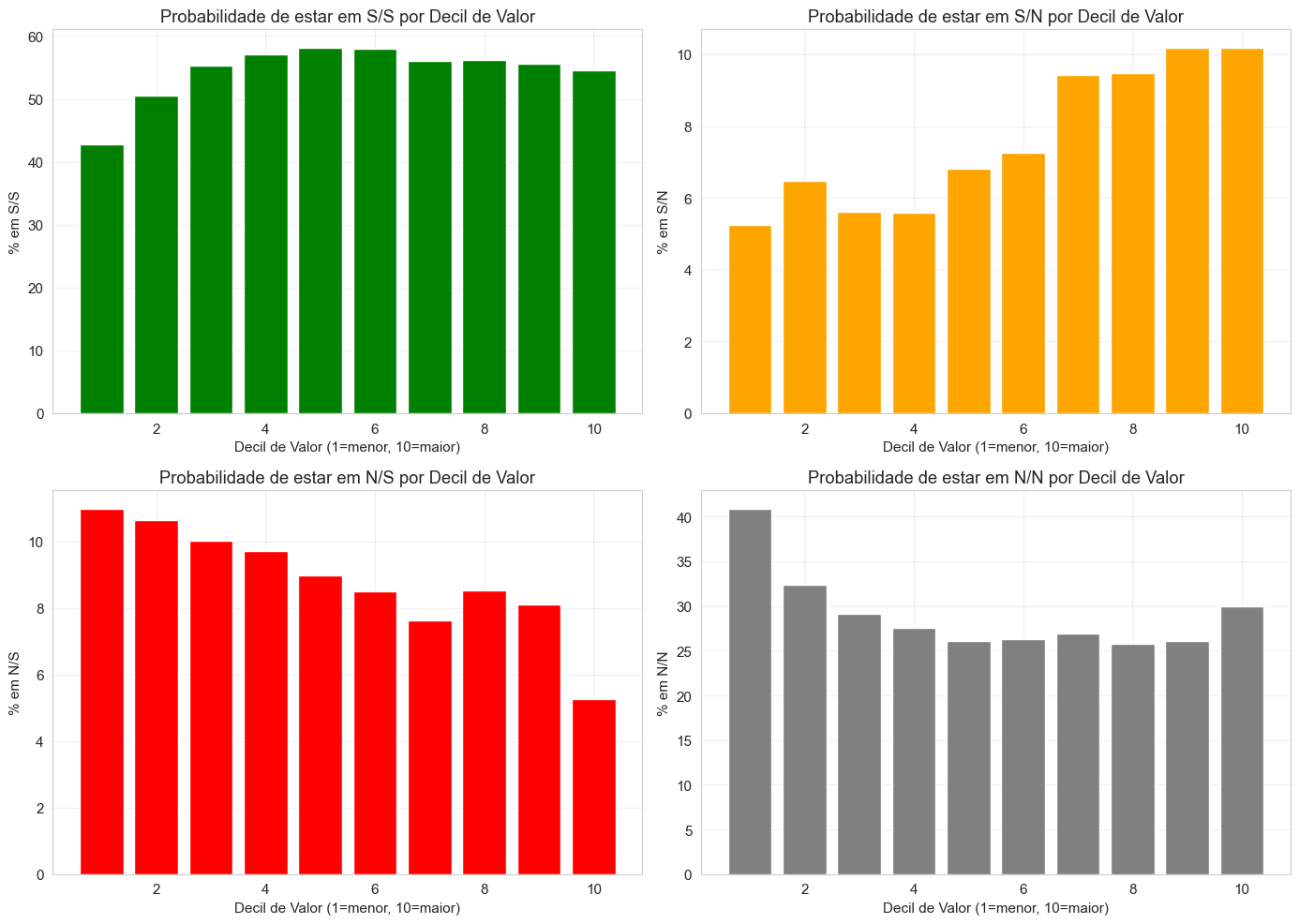
## 3. Análise 1: Regressão Logística - Valor do Projeto vs Quadrantes

Esta análise investiga se o valor financeiro total de um projeto (valortotalproj) influencia sua probabilidade de ser classificado em cada um dos quatro quadrantes de decisão. A regressão logística foi escolhida porque é a técnica estatística apropriada quando queremos predizer uma variável categórica binária (estar ou não em determinado quadrante) a partir de uma variável contínua (valor do projeto).

O **coeficiente** indica a direção e magnitude da relação - valores positivos significam que aumentos no valor do projeto aumentam a probabilidade de estar naquele quadrante. O **Odds Ratio** quantifica esse efeito de forma interpretável: um OR de 1.02 significa que cada aumento de um desvio padrão no valor do projeto aumenta em 2% as chances de estar naquele quadrante. O **Score** representa a acurácia do modelo, ou seja, o percentual de classificações corretas. O **p-value** testa a significância estatística - valores abaixo de 0.05 indicam que a relação encontrada provavelmente não ocorreu por acaso, sendo estatisticamente confiável.

print(f"\n{'='\*73}")  
print("ANÁLISE 1: REGRESSÃO LOGÍSTICA - VALOR DO PROJETO VS QUADRANTES")  
print(f"{'='\*73}")  
  
# Preparar dados para regressão  
X = df\_analise[['valortotalproj']].fillna(0)  
X\_scaled = StandardScaler().fit\_transform(X)  
  
# Criar variáveis dummy para cada quadrante  
for q in ['S/S', 'S/N', 'N/S', 'N/N']:  
 df\_analise[f'is\_{q.replace("/", "\_")}'] = (df\_analise['quadrante'] == q).astype(int)  
  
# Realizar regressão logística para cada quadrante  
resultados\_regressao = {}  
  
for q in ['S/S', 'S/N', 'N/S', 'N/N']:  
 print(f"\n{'-'\*73}")  
 print(f"REGRESSÃO LOGÍSTICA PARA {q}")  
 print(f"{'-'\*73}")  
   
 y = df\_analise[f'is\_{q.replace("/", "\_")}']  
   
 # Treinar modelo  
 model = LogisticRegression(random\_state=42, max\_iter=1000)  
 model.fit(X\_scaled, y)  
   
 # Coeficiente e odds ratio  
 coef = model.coef\_[0][0]  
 odds\_ratio = np.exp(coef)  
   
 # Score do modelo  
 score = model.score(X\_scaled, y)  
   
 # Calcular p-value  
 from scipy.stats import norm  
 n = len(X\_scaled)  
 predictions = model.predict\_proba(X\_scaled)[:, 1]  
 V = np.sum((predictions \* (1 - predictions))[:, np.newaxis] \* (X\_scaled \*\* 2))  
 se = np.sqrt(1 / V) if V > 0 else 1  
 z\_score = coef / se  
 p\_value = 2 \* (1 - norm.cdf(np.abs(z\_score)))  
   
 resultados\_regressao[q] = {  
 'coeficiente': coef,  
 'odds\_ratio': odds\_ratio,  
 'score': score,  
 'p\_value': float(p\_value)  
 }  
   
 print(f"Coeficiente: {coef:.4f}")  
 print(f"Odds Ratio: {odds\_ratio:.4f}")  
 print(f"Score (acurácia): {score:.4f}")  
 print(f"P-value: {float(p\_value):.4e}")  
  
# Visualização  
fig, axes = plt.subplots(2, 2, figsize=(14, 10))  
axes = axes.ravel()  
  
for i, q in enumerate(['S/S', 'S/N', 'N/S', 'N/N']):  
 df\_vis = df\_analise.copy()  
 df\_vis['valor\_bin'] = pd.qcut(df\_vis['valortotalproj'], q=10, labels=range(1, 11), duplicates='drop')  
 prop\_by\_bin = df\_vis.groupby('valor\_bin')[f'is\_{q.replace("/", "\_")}'].mean() \* 100  
   
 axes[i].bar(prop\_by\_bin.index, prop\_by\_bin.values,   
 color=['green' if q=='S/S' else 'orange' if q=='S/N' else 'red' if q=='N/S' else 'gray'][0])  
 axes[i].set\_title(f'Probabilidade de estar em {q} por Decil de Valor')  
 axes[i].set\_xlabel('Decil de Valor (1=menor, 10=maior)')  
 axes[i].set\_ylabel(f'% em {q}')  
 axes[i].grid(True, alpha=0.3)  
  
plt.tight\_layout()  
plt.show()  
  
  
# Resumo  
print(f"\n{'='\*73}")  
print("RESUMO DA ANÁLISE 1")  
print(f"{'='\*73}")  
for q, res in resultados\_regressao.items():  
 sig = "SIM" if res['p\_value'] < 0.05 else "NÃO"  
 direção = "POSITIVA" if res['coeficiente'] > 0 else "NEGATIVA"  
 print(f"{q}: Significativo? {sig} | Direção: {direção} | Odds Ratio: {res['odds\_ratio']:.3f}")

=========================================================================  
ANÁLISE 1: REGRESSÃO LOGÍSTICA - VALOR DO PROJETO VS QUADRANTES  
=========================================================================  
  
-------------------------------------------------------------------------  
REGRESSÃO LOGÍSTICA PARA S/S  
-------------------------------------------------------------------------  
Coeficiente: 0.0198  
Odds Ratio: 1.0200  
Score (acurácia): 0.5381  
P-value: 1.5059e-02  
  
-------------------------------------------------------------------------  
REGRESSÃO LOGÍSTICA PARA S/N  
-------------------------------------------------------------------------  
Coeficiente: 0.0240  
Odds Ratio: 1.0243  
Score (acurácia): 0.9247  
P-value: 3.7396e-02  
  
-------------------------------------------------------------------------  
REGRESSÃO LOGÍSTICA PARA N/S  
-------------------------------------------------------------------------  
Coeficiente: -0.2020  
Odds Ratio: 0.8171  
Score (acurácia): 0.9128  
P-value: 5.8089e-12  
  
-------------------------------------------------------------------------  
REGRESSÃO LOGÍSTICA PARA N/N  
-------------------------------------------------------------------------  
Coeficiente: -0.0007  
Odds Ratio: 0.9993  
Score (acurácia): 0.7006  
P-value: 9.3113e-01



=========================================================================  
RESUMO DA ANÁLISE 1  
=========================================================================  
S/S: Significativo? SIM | Direção: POSITIVA | Odds Ratio: 1.020  
S/N: Significativo? SIM | Direção: POSITIVA | Odds Ratio: 1.024  
N/S: Significativo? SIM | Direção: NEGATIVA | Odds Ratio: 0.817  
N/N: Significativo? NÃO | Direção: NEGATIVA | Odds Ratio: 0.999

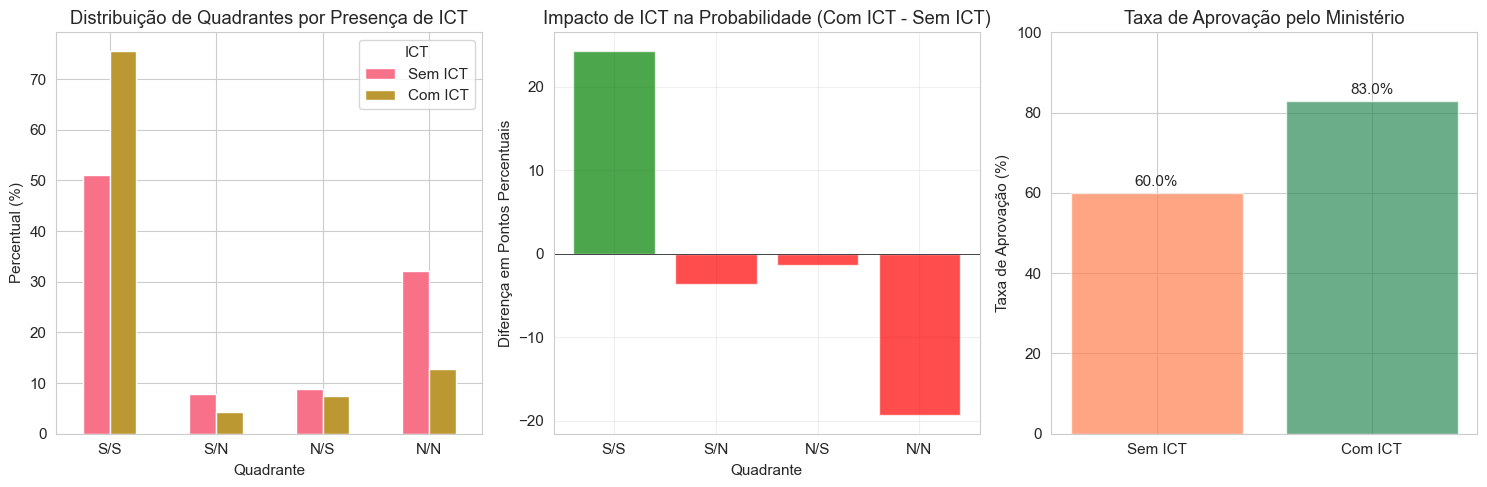
## 4. Análise 2: ICT e Probabilidade por Quadrante

Esta análise examina como a presença de investimento em Instituições de Ciência e Tecnologia (ICTs) afeta a distribuição dos projetos entre os quadrantes. Utiliza-se o teste qui-quadrado para verificar se existe associação significativa entre ter ICT (variável categórica binária) e o quadrante resultante (variável categórica).

O **Risco Relativo** compara diretamente as probabilidades - um RR de 1.48 para S/S significa que projetos com ICT têm 48% mais chance de receber dupla aprovação comparado aos projetos sem ICT. A **diferença em pontos percentuais** mostra o impacto absoluto, enquanto o **p-value do teste qui-quadrado** confirma se essa diferença é estatisticamente significativa ou poderia ter ocorrido por acaso.

print(f"\n{'='\*73}")  
print("ANÁLISE 2: IMPACTO DE ICT NA PROBABILIDADE POR QUADRANTE")  
print(f"{'='\*73}")  
  
# Criar variável binária para ICT  
df\_analise['tem\_ict'] = (df\_analise['valorictproj'] > 0).astype(int)  
  
# Estatísticas gerais  
total\_com\_ict = df\_analise['tem\_ict'].sum()  
pct\_com\_ict = df\_analise['tem\_ict'].mean() \* 100  
  
print(f"Projetos com ICT: {total\_com\_ict:,} ({pct\_com\_ict:.1f}%)")  
print(f"Projetos sem ICT: {len(df\_analise) - total\_com\_ict:,} ({100-pct\_com\_ict:.1f}%)")  
  
# Análise de probabilidade por quadrante  
print(f"\n{'-'\*60}")  
print("PROBABILIDADE DE CADA QUADRANTE COM/SEM ICT")  
print(f"{'-'\*60}")  
  
# Tabela de contingência  
for q in ['S/S', 'S/N', 'N/S', 'N/N']:  
 prob\_com\_ict = df\_analise[df\_analise['tem\_ict'] == 1]['quadrante'].value\_counts(normalize=True).get(q, 0) \* 100  
 prob\_sem\_ict = df\_analise[df\_analise['tem\_ict'] == 0]['quadrante'].value\_counts(normalize=True).get(q, 0) \* 100  
   
 contingency = pd.crosstab(df\_analise['tem\_ict'], df\_analise['quadrante'] == q)  
 chi2, p\_value, dof, expected = stats.chi2\_contingency(contingency)  
   
 risco\_relativo = prob\_com\_ict / prob\_sem\_ict if prob\_sem\_ict > 0 else np.inf  
   
 print(f"\n{q}:")  
 print(f" Com ICT: {prob\_com\_ict:.1f}%")  
 print(f" Sem ICT: {prob\_sem\_ict:.1f}%")  
 print(f" Diferença: {prob\_com\_ict - prob\_sem\_ict:+.1f} pontos percentuais")  
 print(f" Risco Relativo: {risco\_relativo:.2f}")  
 print(f" P-value (χ²): {p\_value:.4f}")  
   
 if p\_value < 0.05:  
 if prob\_com\_ict > prob\_sem\_ict:  
 print(f" ✓ Ter ICT AUMENTA significativamente a probabilidade de {q}")  
 else:  
 print(f" ✓ Ter ICT DIMINUI significativamente a probabilidade de {q}")  
 else:  
 print(f" ✗ ICT não tem efeito significativo em {q}")  
  
# Visualização  
fig, axes = plt.subplots(1, 3, figsize=(15, 5))  
  
# Gráfico 1: Distribuição geral  
quadrante\_ict = pd.crosstab(df\_analise['quadrante'], df\_analise['tem\_ict'], normalize='columns') \* 100  
quadrante\_ict = quadrante\_ict.sort\_index(ascending=False)  
quadrante\_ict.plot(kind='bar', ax=axes[0])  
axes[0].set\_title('Distribuição de Quadrantes por Presença de ICT')  
axes[0].set\_xlabel('Quadrante')  
axes[0].set\_ylabel('Percentual (%)')  
axes[0].legend(title='ICT', labels=['Sem ICT', 'Com ICT'])  
axes[0].set\_xticklabels(axes[0].get\_xticklabels(), rotation=0)  
  
# Gráfico 2: Diferença absoluta  
diffs = []  
for q in ['S/S', 'S/N', 'N/S', 'N/N']:  
 prob\_com = df\_analise[df\_analise['tem\_ict'] == 1]['quadrante'].value\_counts(normalize=True).get(q, 0) \* 100  
 prob\_sem = df\_analise[df\_analise['tem\_ict'] == 0]['quadrante'].value\_counts(normalize=True).get(q, 0) \* 100  
 diffs.append(prob\_com - prob\_sem)  
  
colors = ['green' if d > 0 else 'red' for d in diffs]  
axes[1].bar(['S/S', 'S/N', 'N/S', 'N/N'], diffs, color=colors, alpha=0.7)  
axes[1].set\_title('Impacto de ICT na Probabilidade (Com ICT - Sem ICT)')  
axes[1].set\_xlabel('Quadrante')  
axes[1].set\_ylabel('Diferença em Pontos Percentuais')  
axes[1].axhline(y=0, color='black', linestyle='-', linewidth=0.5)  
axes[1].grid(True, alpha=0.3)  
  
# Gráfico 3: Taxa de aprovação  
aprovacao\_com\_ict = df\_analise[df\_analise['tem\_ict'] == 1]['ministerio\_aprova'].mean() \* 100  
aprovacao\_sem\_ict = df\_analise[df\_analise['tem\_ict'] == 0]['ministerio\_aprova'].mean() \* 100  
  
bars = axes[2].bar(['Sem ICT', 'Com ICT'], [aprovacao\_sem\_ict, aprovacao\_com\_ict],   
 color=['coral', 'seagreen'], alpha=0.7)  
axes[2].set\_title('Taxa de Aprovação pelo Ministério')  
axes[2].set\_ylabel('Taxa de Aprovação (%)')  
axes[2].set\_ylim(0, 100)  
  
for bar, val in zip(bars, [aprovacao\_sem\_ict, aprovacao\_com\_ict]):  
 axes[2].text(bar.get\_x() + bar.get\_width()/2, bar.get\_height() + 1,  
 f'{val:.1f}%', ha='center', va='bottom')  
  
plt.tight\_layout()  
plt.show()

=========================================================================  
ANÁLISE 2: IMPACTO DE ICT NA PROBABILIDADE POR QUADRANTE  
=========================================================================  
Projetos com ICT: 7,472 (11.0%)  
Projetos sem ICT: 60,311 (89.0%)  
  
------------------------------------------------------------  
PROBABILIDADE DE CADA QUADRANTE COM/SEM ICT  
------------------------------------------------------------  
  
S/S:  
 Com ICT: 75.4%  
 Sem ICT: 51.1%  
 Diferença: +24.3 pontos percentuais  
 Risco Relativo: 1.48  
 P-value (χ²): 0.0000  
 ✓ Ter ICT AUMENTA significativamente a probabilidade de S/S  
  
S/N:  
 Com ICT: 4.3%  
 Sem ICT: 7.9%  
 Diferença: -3.6 pontos percentuais  
 Risco Relativo: 0.55  
 P-value (χ²): 0.0000  
 ✓ Ter ICT DIMINUI significativamente a probabilidade de S/N  
  
N/S:  
 Com ICT: 7.5%  
 Sem ICT: 8.9%  
 Diferença: -1.3 pontos percentuais  
 Risco Relativo: 0.85  
 P-value (χ²): 0.0001  
 ✓ Ter ICT DIMINUI significativamente a probabilidade de N/S  
  
N/N:  
 Com ICT: 12.7%  
 Sem ICT: 32.1%  
 Diferença: -19.4 pontos percentuais  
 Risco Relativo: 0.40  
 P-value (χ²): 0.0000  
 ✓ Ter ICT DIMINUI significativamente a probabilidade de N/N



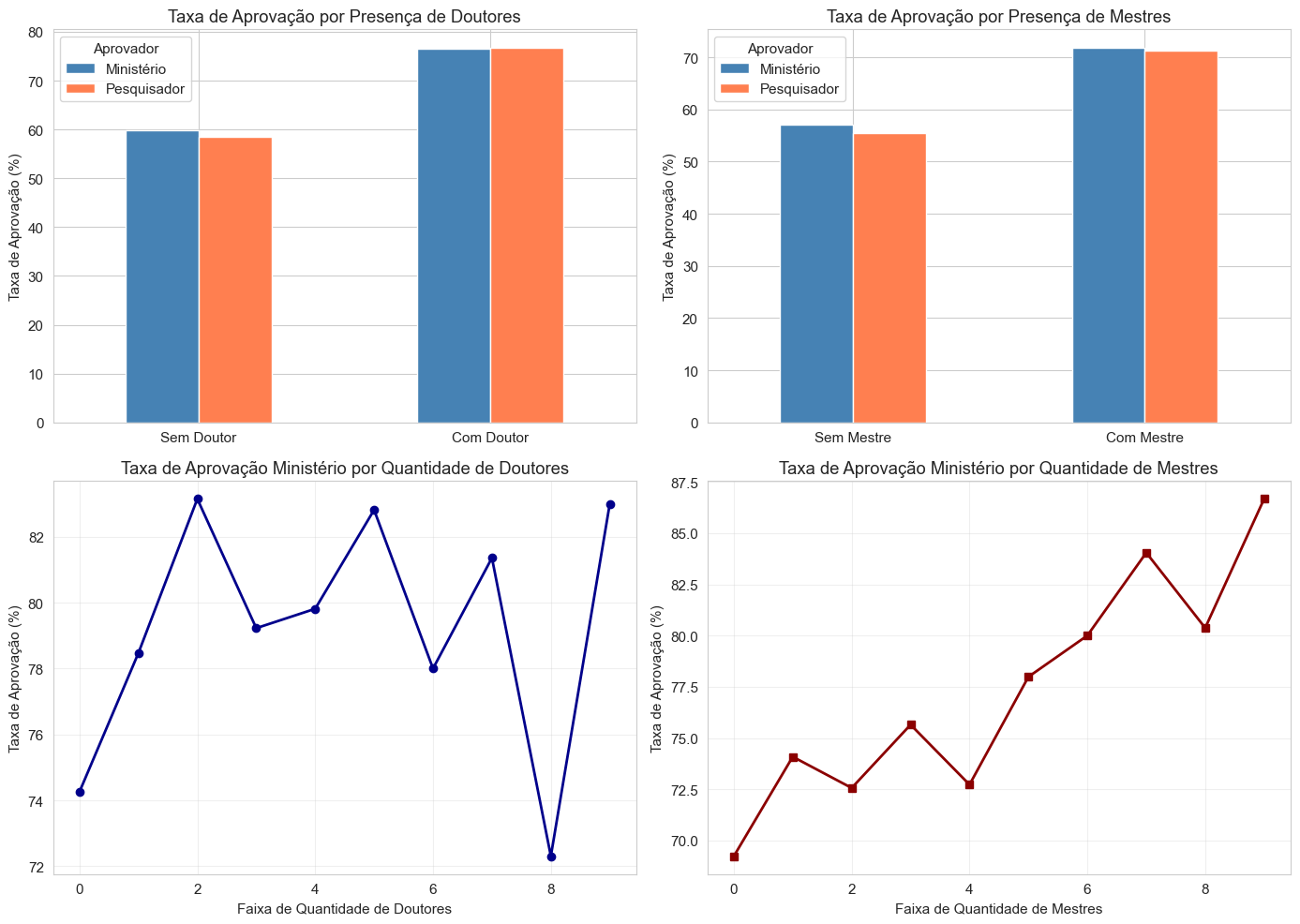
## 5. Análise 3: Doutores/Mestres e Aprovação

Esta análise investiga se a presença de profissionais altamente qualificados (doutores e mestres) aumenta as chances de aprovação tanto pelo Ministério quanto pelo pesquisador DO. Combina duas abordagens: análise categórica (tem/não tem) usando teste qui-quadrado, e análise contínua (quantidade) usando regressão logística.

O **coeficiente de regressão** quantifica o impacto de cada doutor ou mestre adicional na probabilidade de aprovação. A comparação entre **taxas de aprovação com e sem qualificação** fornece uma visão intuitiva do impacto, enquanto o **p-value** confirma a significância estatística dessa diferença.

print(f"\n{'='\*73}")  
print("ANÁLISE 3: IMPACTO DE DOUTORES E MESTRES NA APROVAÇÃO")  
print(f"{'='\*73}")  
  
# Preparar dados  
df\_analise['tem\_doutor'] = (df\_analise['quantdoutorproj'] > 0).astype(int)  
df\_analise['tem\_mestre'] = (df\_analise['quantmestreproj'] > 0).astype(int)  
  
# Análise para DOUTORES  
print("\nANÁLISE PARA DOUTORES")  
print("-"\*60)  
  
for aprovador, coluna in [('Ministério', 'ministerio\_aprova'), ('Pesquisador DO', 'pesquisador\_aprova')]:  
 print(f"\nAprovação pelo {aprovador}:")  
   
 # Taxa de aprovação com/sem doutor  
 taxa\_com\_doutor = df\_analise[df\_analise['tem\_doutor'] == 1][coluna].mean() \* 100  
 taxa\_sem\_doutor = df\_analise[df\_analise['tem\_doutor'] == 0][coluna].mean() \* 100  
   
 # Teste estatístico  
 contingency = pd.crosstab(df\_analise['tem\_doutor'], df\_analise[coluna])  
 chi2, p\_value, dof, expected = stats.chi2\_contingency(contingency)  
   
 # Regressão logística  
 X\_doutor = df\_analise[['quantdoutorproj']].fillna(0)  
 y = df\_analise[coluna]  
 model = LogisticRegression(random\_state=42, max\_iter=1000)  
 model.fit(StandardScaler().fit\_transform(X\_doutor), y)  
 coef\_doutor = model.coef\_[0][0]  
   
 print(f" Taxa aprovação COM doutor: {taxa\_com\_doutor:.1f}%")  
 print(f" Taxa aprovação SEM doutor: {taxa\_sem\_doutor:.1f}%")  
 print(f" Diferença: {taxa\_com\_doutor - taxa\_sem\_doutor:+.1f} pp")  
 print(f" P-value: {p\_value:.4f}")  
 print(f" Coeficiente regressão (por doutor adicional): {coef\_doutor:.4f}")  
   
 if p\_value < 0.05:  
 if taxa\_com\_doutor > taxa\_sem\_doutor:  
 print(f" ✓ Ter doutores AUMENTA significativamente a aprovação pelo {aprovador}")  
 else:  
 print(f" ✓ Ter doutores DIMINUI significativamente a aprovação pelo {aprovador}")  
 else:  
 print(f" ✗ Doutores não têm efeito significativo na aprovação pelo {aprovador}")  
  
# Análise para MESTRES  
print("\n" + "="\*60)  
print("ANÁLISE PARA MESTRES")  
print("-"\*60)  
  
for aprovador, coluna in [('Ministério', 'ministerio\_aprova'), ('Pesquisador DO', 'pesquisador\_aprova')]:  
 print(f"\nAprovação pelo {aprovador}:")  
   
 # Taxa de aprovação com/sem mestre  
 taxa\_com\_mestre = df\_analise[df\_analise['tem\_mestre'] == 1][coluna].mean() \* 100  
 taxa\_sem\_mestre = df\_analise[df\_analise['tem\_mestre'] == 0][coluna].mean() \* 100  
   
 # Teste estatístico  
 contingency = pd.crosstab(df\_analise['tem\_mestre'], df\_analise[coluna])  
 chi2, p\_value, dof, expected = stats.chi2\_contingency(contingency)  
   
 # Regressão logística  
 X\_mestre = df\_analise[['quantmestreproj']].fillna(0)  
 y = df\_analise[coluna]  
 model = LogisticRegression(random\_state=42, max\_iter=1000)  
 model.fit(StandardScaler().fit\_transform(X\_mestre), y)  
 coef\_mestre = model.coef\_[0][0]  
   
 print(f" Taxa aprovação COM mestre: {taxa\_com\_mestre:.1f}%")  
 print(f" Taxa aprovação SEM mestre: {taxa\_sem\_mestre:.1f}%")  
 print(f" Diferença: {taxa\_com\_mestre - taxa\_sem\_mestre:+.1f} pp")  
 print(f" P-value: {p\_value:.4f}")  
 print(f" Coeficiente regressão (por mestre adicional): {coef\_mestre:.4f}")  
   
 if p\_value < 0.05:  
 if taxa\_com\_mestre > taxa\_sem\_mestre:  
 print(f" ✓ Ter mestres AUMENTA significativamente a aprovação pelo {aprovador}")  
 else:  
 print(f" ✓ Ter mestres DIMINUI significativamente a aprovação pelo {aprovador}")  
 else:  
 print(f" ✗ Mestres não têm efeito significativo na aprovação pelo {aprovador}")  
  
# Visualização  
fig, axes = plt.subplots(2, 2, figsize=(14, 10))  
  
# Gráfico 1: Taxa de aprovação com/sem doutor  
taxa\_doutor\_data = pd.DataFrame({  
 'Ministério': [df\_analise[df\_analise['tem\_doutor'] == 0]['ministerio\_aprova'].mean() \* 100,  
 df\_analise[df\_analise['tem\_doutor'] == 1]['ministerio\_aprova'].mean() \* 100],  
 'Pesquisador': [df\_analise[df\_analise['tem\_doutor'] == 0]['pesquisador\_aprova'].mean() \* 100,  
 df\_analise[df\_analise['tem\_doutor'] == 1]['pesquisador\_aprova'].mean() \* 100]  
}, index=['Sem Doutor', 'Com Doutor'])  
  
taxa\_doutor\_data.plot(kind='bar', ax=axes[0,0], color=['steelblue', 'coral'])  
axes[0,0].set\_title('Taxa de Aprovação por Presença de Doutores')  
axes[0,0].set\_ylabel('Taxa de Aprovação (%)')  
axes[0,0].set\_xticklabels(axes[0,0].get\_xticklabels(), rotation=0)  
axes[0,0].legend(title='Aprovador')  
  
# Gráfico 2: Taxa de aprovação com/sem mestre  
taxa\_mestre\_data = pd.DataFrame({  
 'Ministério': [df\_analise[df\_analise['tem\_mestre'] == 0]['ministerio\_aprova'].mean() \* 100,  
 df\_analise[df\_analise['tem\_mestre'] == 1]['ministerio\_aprova'].mean() \* 100],  
 'Pesquisador': [df\_analise[df\_analise['tem\_mestre'] == 0]['pesquisador\_aprova'].mean() \* 100,  
 df\_analise[df\_analise['tem\_mestre'] == 1]['pesquisador\_aprova'].mean() \* 100]  
}, index=['Sem Mestre', 'Com Mestre'])  
  
taxa\_mestre\_data.plot(kind='bar', ax=axes[0,1], color=['steelblue', 'coral'])  
axes[0,1].set\_title('Taxa de Aprovação por Presença de Mestres')  
axes[0,1].set\_ylabel('Taxa de Aprovação (%)')  
axes[0,1].set\_xticklabels(axes[0,1].get\_xticklabels(), rotation=0)  
axes[0,1].legend(title='Aprovador')  
  
# Gráfico 3: Relação quantidade de doutores vs aprovação  
df\_plot = df\_analise[df\_analise['quantdoutorproj'] <= df\_analise['quantdoutorproj'].quantile(0.95)]  
df\_grouped = df\_plot.groupby(pd.cut(df\_plot['quantdoutorproj'], bins=10))['ministerio\_aprova'].mean() \* 100  
  
axes[1,0].plot(range(len(df\_grouped)), df\_grouped.values, marker='o', linewidth=2, color='darkblue')  
axes[1,0].set\_title('Taxa de Aprovação Ministério por Quantidade de Doutores')  
axes[1,0].set\_xlabel('Faixa de Quantidade de Doutores')  
axes[1,0].set\_ylabel('Taxa de Aprovação (%)')  
axes[1,0].grid(True, alpha=0.3)  
  
# Gráfico 4: Relação quantidade de mestres vs aprovação  
df\_plot = df\_analise[df\_analise['quantmestreproj'] <= df\_analise['quantmestreproj'].quantile(0.95)]  
df\_grouped = df\_plot.groupby(pd.cut(df\_plot['quantmestreproj'], bins=10))['ministerio\_aprova'].mean() \* 100  
  
axes[1,1].plot(range(len(df\_grouped)), df\_grouped.values, marker='s', linewidth=2, color='darkred')  
axes[1,1].set\_title('Taxa de Aprovação Ministério por Quantidade de Mestres')  
axes[1,1].set\_xlabel('Faixa de Quantidade de Mestres')  
axes[1,1].set\_ylabel('Taxa de Aprovação (%)')  
axes[1,1].grid(True, alpha=0.3)  
  
plt.tight\_layout()  
plt.show()

=========================================================================  
ANÁLISE 3: IMPACTO DE DOUTORES E MESTRES NA APROVAÇÃO  
=========================================================================  
  
ANÁLISE PARA DOUTORES  
------------------------------------------------------------  
  
Aprovação pelo Ministério:  
 Taxa aprovação COM doutor: 76.6%  
 Taxa aprovação SEM doutor: 59.9%  
 Diferença: +16.7 pp  
 P-value: 0.0000  
 Coeficiente regressão (por doutor adicional): 0.4098  
 ✓ Ter doutores AUMENTA significativamente a aprovação pelo Ministério  
  
Aprovação pelo Pesquisador DO:  
 Taxa aprovação COM doutor: 76.7%  
 Taxa aprovação SEM doutor: 58.4%  
 Diferença: +18.2 pp  
 P-value: 0.0000  
 Coeficiente regressão (por doutor adicional): 0.4394  
 ✓ Ter doutores AUMENTA significativamente a aprovação pelo Pesquisador DO  
  
============================================================  
ANÁLISE PARA MESTRES  
------------------------------------------------------------  
  
Aprovação pelo Ministério:  
 Taxa aprovação COM mestre: 71.8%  
 Taxa aprovação SEM mestre: 57.1%  
 Diferença: +14.7 pp  
 P-value: 0.0000  
 Coeficiente regressão (por mestre adicional): 0.3018  
 ✓ Ter mestres AUMENTA significativamente a aprovação pelo Ministério  
  
Aprovação pelo Pesquisador DO:  
 Taxa aprovação COM mestre: 71.2%  
 Taxa aprovação SEM mestre: 55.5%  
 Diferença: +15.7 pp  
 P-value: 0.0000  
 Coeficiente regressão (por mestre adicional): 0.3176  
 ✓ Ter mestres AUMENTA significativamente a aprovação pelo Pesquisador DO



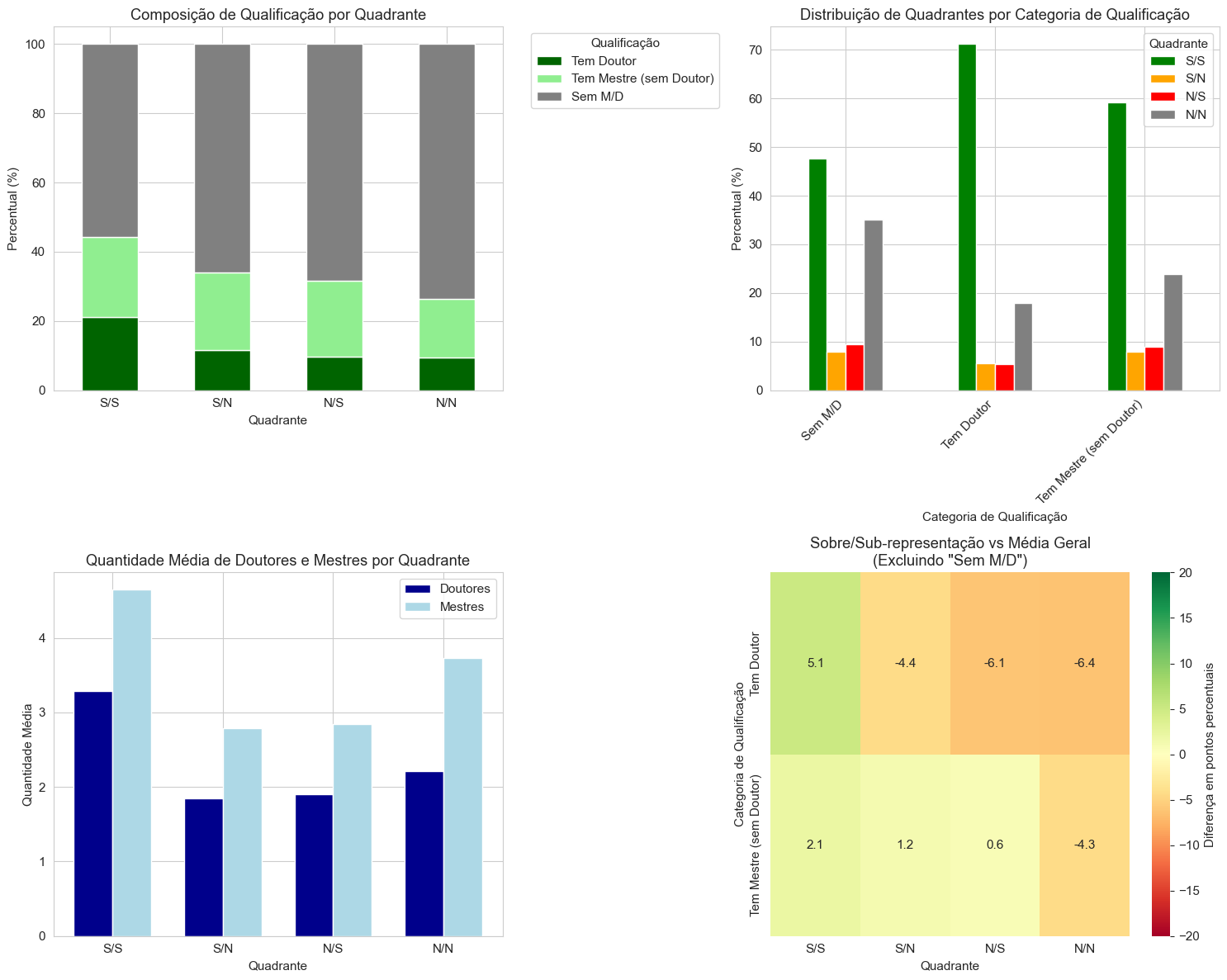
### 5.1. Análise 3.1: Importância de Mestres e Doutores por Quadrante

Esta análise aprofunda a investigação anterior, examinando como diferentes combinações de qualificação (apenas doutor, apenas mestre, ambos, ou nenhum) se distribuem entre os quadrantes. O teste qui-quadrado de independência verifica se existe associação significativa entre o perfil de qualificação da equipe e o quadrante resultante.

A análise de **sobre/sub-representação** compara a proporção de cada categoria de qualificação em cada quadrante com sua proporção na população geral, identificando padrões. Por exemplo, se “Tem Doutor” representa 16% dos projetos gerais mas 21% dos projetos em S/S, há uma sobre-representação de 5 pontos percentuais, indicando que doutores são especialmente importantes para obter dupla aprovação.

print(f"\n{'='\*73}")  
print("ANÁLISE 3.1: IMPORTÂNCIA DE MESTRES E DOUTORES POR QUADRANTE")  
print(f"{'='\*73}")  
  
# Criar as novas categorias conforme solicitado  
df\_analise['categoria\_qualificacao'] = 'Sem M/D'  
df\_analise.loc[df\_analise['tem\_doutor'] == 1, 'categoria\_qualificacao'] = 'Tem Doutor'  
df\_analise.loc[(df\_analise['tem\_doutor'] == 0) & (df\_analise['tem\_mestre'] == 1), 'categoria\_qualificacao'] = 'Tem Mestre (sem Doutor)'  
  
# Verificar distribuição das categorias  
print("Distribuição das categorias de qualificação:")  
print(df\_analise['categoria\_qualificacao'].value\_counts())  
print(f"\nPercentual:")  
print(df\_analise['categoria\_qualificacao'].value\_counts(normalize=True) \* 100)  
  
categorias\_existentes = ['Tem Doutor', 'Tem Mestre (sem Doutor)', 'Sem M/D']  
  
# Análise por quadrante  
print("\n" + "="\*73)  
print("DISTRIBUIÇÃO DE QUALIFICAÇÃO POR QUADRANTE")  
print("-"\*73)  
  
resultados\_qualificacao = {}  
  
for q in ['S/S', 'S/N', 'N/S', 'N/N']:  
 print(f"\n{q}:")  
 df\_q = df\_analise[df\_analise['quadrante'] == q]  
   
 # Distribuição de categorias neste quadrante  
 dist = df\_q['categoria\_qualificacao'].value\_counts(normalize=True) \* 100  
   
 print(" Composição do quadrante:")  
 for cat in categorias\_existentes:  
 if cat in dist.index:  
 print(f" {cat:25}: {dist[cat]:.1f}%")  
 else:  
 print(f" {cat:25}: 0.0%")  
   
 # Comparar com distribuição geral  
 dist\_geral = df\_analise['categoria\_qualificacao'].value\_counts(normalize=True) \* 100  
   
 # Calcular sobre-representação  
 print("\n Sobre/sub-representação (vs média geral):")  
 for cat in categorias\_existentes:  
 if cat in dist.index and cat in dist\_geral.index:  
 diff = dist[cat] - dist\_geral[cat]  
 ratio = dist[cat] / dist\_geral[cat]  
 resultados\_qualificacao[f"{q}\_{cat}"] = {'diff': diff, 'ratio': ratio}  
   
 if diff > 2:  
 print(f" {cat:25}: +{diff:.1f}pp (↑ {ratio:.2f}x)")  
 elif diff < -2:  
 print(f" {cat:25}: {diff:.1f}pp (↓ {ratio:.2f}x)")  
 else:  
 print(f" {cat:25}: {diff:+.1f}pp (≈ média)")  
  
# Análise inversa: dado uma categoria, qual a probabilidade de cada quadrante  
print(f"\n{'='\*73}")  
print("PROBABILIDADE DE QUADRANTE DADA A QUALIFICAÇÃO")  
print("-"\*73)  
  
for cat in categorias\_existentes:  
 df\_cat = df\_analise[df\_analise['categoria\_qualificacao'] == cat]  
 if len(df\_cat) > 0:  
 print(f"\n{cat} (n={len(df\_cat):,}):")  
 dist\_quad = df\_cat['quadrante'].value\_counts(normalize=True) \* 100  
   
 for q in ['S/S', 'S/N', 'N/S', 'N/N']:  
 if q in dist\_quad.index:  
 # Comparar com média geral  
 media\_geral = (df\_analise['quadrante'] == q).mean() \* 100  
 diff = dist\_quad[q] - media\_geral  
   
 if abs(diff) > 3:  
 sinal = "↑" if diff > 0 else "↓"  
 print(f" {q}: {dist\_quad[q]:.1f}% ({sinal} {abs(diff):.1f}pp vs média)")  
 else:  
 print(f" {q}: {dist\_quad[q]:.1f}% (≈ média)")  
  
# Teste estatístico de independência  
from scipy.stats import chi2\_contingency  
  
contingency\_table = pd.crosstab(df\_analise['categoria\_qualificacao'], df\_analise['quadrante'])  
chi2, p\_value, dof, expected = chi2\_contingency(contingency\_table)  
  
print(f"\n{'='\*73}")  
print("TESTE DE INDEPENDÊNCIA")  
print("-"\*73)  
print(f"Chi-quadrado: {chi2:.2f}")  
print(f"P-value: {p\_value:.2e}")  
print(f"Graus de liberdade: {dof}")  
  
if p\_value < 0.001:  
 print("\n✓ Há uma associação MUITO FORTE entre qualificação e quadrantes (p < 0.001)")  
elif p\_value < 0.05:  
 print("\n✓ Há uma associação significativa entre qualificação e quadrantes (p < 0.05)")  
else:  
 print("\n✗ Não há associação significativa entre qualificação e quadrantes")  
  
# Visualização  
fig, axes = plt.subplots(2, 2, figsize=(15, 12))  
  
# Gráfico 1: Distribuição de categorias por quadrante  
cat\_by\_quad = pd.crosstab(df\_analise['quadrante'],   
 df\_analise['categoria\_qualificacao'],   
 normalize='index') \* 100  
  
cat\_by\_quad = cat\_by\_quad.sort\_index(ascending=False)  
  
# Ordenar colunas  
cols\_ordered = ['Tem Doutor', 'Tem Mestre (sem Doutor)', 'Sem M/D']  
cols\_to\_plot = [col for col in cols\_ordered if col in cat\_by\_quad.columns]  
  
colors = ['darkgreen', 'lightgreen', 'gray']  
  
cat\_by\_quad[cols\_to\_plot].plot(  
 kind='bar', stacked=True, ax=axes[0,0],   
 color=colors[:len(cols\_to\_plot)]  
)  
axes[0,0].set\_title('Composição de Qualificação por Quadrante')  
axes[0,0].set\_xlabel('Quadrante')  
axes[0,0].set\_ylabel('Percentual (%)')  
axes[0,0].set\_xticklabels(axes[0,0].get\_xticklabels(), rotation=0)  
axes[0,0].legend(title='Qualificação', bbox\_to\_anchor=(1.05, 1))  
  
# Gráfico 2: Probabilidade de quadrante por categoria  
quad\_by\_cat = pd.crosstab(df\_analise['categoria\_qualificacao'],   
 df\_analise['quadrante'],   
 normalize='index') \* 100  
  
quad\_by\_cat[['S/S', 'S/N', 'N/S', 'N/N']].plot(  
 kind='bar', ax=axes[0,1],  
 color=['green', 'orange', 'red', 'gray']  
)  
axes[0,1].set\_title('Distribuição de Quadrantes por Categoria de Qualificação')  
axes[0,1].set\_xlabel('Categoria de Qualificação')  
axes[0,1].set\_ylabel('Percentual (%)')  
axes[0,1].set\_xticklabels(axes[0,1].get\_xticklabels(), rotation=45, ha='right')  
axes[0,1].legend(title='Quadrante')  
  
# Gráfico 3: Comparação de médias de doutores e mestres  
quad\_means = df\_analise.groupby('quadrante').agg({  
 'quantdoutorproj': 'mean',  
 'quantmestreproj': 'mean'  
})  
  
quad\_means = quad\_means.sort\_index(ascending=False)  
  
x = np.arange(len(quad\_means.index))  
width = 0.35  
  
axes[1,0].bar(x - width/2, quad\_means['quantdoutorproj'], width, label='Doutores', color='darkblue')  
axes[1,0].bar(x + width/2, quad\_means['quantmestreproj'], width, label='Mestres', color='lightblue')  
axes[1,0].set\_xlabel('Quadrante')  
axes[1,0].set\_ylabel('Quantidade Média')  
axes[1,0].set\_title('Quantidade Média de Doutores e Mestres por Quadrante')  
axes[1,0].set\_xticks(x)  
axes[1,0].set\_xticklabels(quad\_means.index)  
axes[1,0].legend()  
  
# Gráfico 4: Heatmap de sobre-representação (SEM "Sem M/D")  
# Criar matriz apenas com categorias relevantes  
categorias\_heatmap = ['Tem Doutor', 'Tem Mestre (sem Doutor)']  
matriz\_sobre = pd.DataFrame(index=categorias\_heatmap,  
 columns=['S/S', 'S/N', 'N/S', 'N/N'])  
  
for q in ['S/S', 'S/N', 'N/S', 'N/N']:  
 for cat in categorias\_heatmap:  
 key = f"{q}\_{cat}"  
 if key in resultados\_qualificacao:  
 matriz\_sobre.loc[cat, q] = resultados\_qualificacao[key]['diff']  
  
matriz\_sobre = matriz\_sobre.astype(float).fillna(0)  
sns.heatmap(matriz\_sobre, annot=True, fmt='.1f', cmap='RdYlGn', center=0, ax=axes[1,1],  
 cbar\_kws={'label': 'Diferença em pontos percentuais'},  
 vmin=-20, vmax=20)  
axes[1,1].set\_title('Sobre/Sub-representação vs Média Geral\n(Excluindo "Sem M/D")')  
axes[1,1].set\_xlabel('Quadrante')  
axes[1,1].set\_ylabel('Categoria de Qualificação')  
  
plt.tight\_layout()  
plt.show()  
  
# Insights principais  
print(f"\n{'='\*73}")  
print("INSIGHTS PRINCIPAIS DA ANÁLISE 3.1")  
print("-"\*73)  
  
# Taxa de sucesso por categoria  
for cat in categorias\_existentes:  
 df\_cat = df\_analise[df\_analise['categoria\_qualificacao'] == cat]  
 if len(df\_cat) > 0:  
 taxa\_q1 = (df\_cat['quadrante'] == 'Q1').mean() \* 100  
 taxa\_aprovacao = df\_cat['quadrante'].isin(['Q1', 'Q3']).mean() \* 100  
 print(f"\n{cat}:")  
 print(f" Taxa Q1 (ambos aprovam): {taxa\_q1:.1f}%")  
 print(f" Taxa aprovação Ministério: {taxa\_aprovacao:.1f}%")

=========================================================================  
ANÁLISE 3.1: IMPORTÂNCIA DE MESTRES E DOUTORES POR QUADRANTE  
=========================================================================  
Distribuição das categorias de qualificação:  
categoria\_qualificacao  
Sem M/D 42642  
Tem Mestre (sem Doutor) 14356  
Tem Doutor 10785  
Name: count, dtype: int64  
  
Percentual:  
categoria\_qualificacao  
Sem M/D 62.909579  
Tem Mestre (sem Doutor) 21.179352  
Tem Doutor 15.911069  
Name: proportion, dtype: float64  
  
=========================================================================  
DISTRIBUIÇÃO DE QUALIFICAÇÃO POR QUADRANTE  
-------------------------------------------------------------------------  
  
S/S:  
 Composição do quadrante:  
 Tem Doutor : 21.1%  
 Tem Mestre (sem Doutor) : 23.3%  
 Sem M/D : 55.7%  
  
 Sobre/sub-representação (vs média geral):  
 Tem Doutor : +5.1pp (↑ 1.32x)  
 Tem Mestre (sem Doutor) : +2.1pp (↑ 1.10x)  
 Sem M/D : -7.3pp (↓ 0.88x)  
  
S/N:  
 Composição do quadrante:  
 Tem Doutor : 11.6%  
 Tem Mestre (sem Doutor) : 22.4%  
 Sem M/D : 66.0%  
  
 Sobre/sub-representação (vs média geral):  
 Tem Doutor : -4.4pp (↓ 0.73x)  
 Tem Mestre (sem Doutor) : +1.2pp (≈ média)  
 Sem M/D : +3.1pp (↑ 1.05x)  
  
N/S:  
 Composição do quadrante:  
 Tem Doutor : 9.8%  
 Tem Mestre (sem Doutor) : 21.8%  
 Sem M/D : 68.4%  
  
 Sobre/sub-representação (vs média geral):  
 Tem Doutor : -6.1pp (↓ 0.61x)  
 Tem Mestre (sem Doutor) : +0.6pp (≈ média)  
 Sem M/D : +5.5pp (↑ 1.09x)  
  
N/N:  
 Composição do quadrante:  
 Tem Doutor : 9.6%  
 Tem Mestre (sem Doutor) : 16.9%  
 Sem M/D : 73.6%  
  
 Sobre/sub-representação (vs média geral):  
 Tem Doutor : -6.4pp (↓ 0.60x)  
 Tem Mestre (sem Doutor) : -4.3pp (↓ 0.80x)  
 Sem M/D : +10.6pp (↑ 1.17x)  
  
=========================================================================  
PROBABILIDADE DE QUADRANTE DADA A QUALIFICAÇÃO  
-------------------------------------------------------------------------  
  
Tem Doutor (n=10,785):  
 S/S: 71.2% (↑ 17.4pp vs média)  
 S/N: 5.5% (≈ média)  
 N/S: 5.4% (↓ 3.4pp vs média)  
 N/N: 18.0% (↓ 12.0pp vs média)  
  
Tem Mestre (sem Doutor) (n=14,356):  
 S/S: 59.2% (↑ 5.4pp vs média)  
 S/N: 8.0% (≈ média)  
 N/S: 9.0% (≈ média)  
 N/N: 23.9% (↓ 6.1pp vs média)  
  
Sem M/D (n=42,642):  
 S/S: 47.6% (↓ 6.2pp vs média)  
 S/N: 7.9% (≈ média)  
 N/S: 9.5% (≈ média)  
 N/N: 35.0% (↑ 5.1pp vs média)  
  
=========================================================================  
TESTE DE INDEPENDÊNCIA  
-------------------------------------------------------------------------  
Chi-quadrado: 2287.53  
P-value: 0.00e+00  
Graus de liberdade: 6  
  
✓ Há uma associação MUITO FORTE entre qualificação e quadrantes (p < 0.001)



=========================================================================  
INSIGHTS PRINCIPAIS DA ANÁLISE 3.1  
-------------------------------------------------------------------------  
  
Tem Doutor:  
 Taxa Q1 (ambos aprovam): 0.0%  
 Taxa aprovação Ministério: 0.0%  
  
Tem Mestre (sem Doutor):  
 Taxa Q1 (ambos aprovam): 0.0%  
 Taxa aprovação Ministério: 0.0%  
  
Sem M/D:  
 Taxa Q1 (ambos aprovam): 0.0%  
 Taxa aprovação Ministério: 0.0%

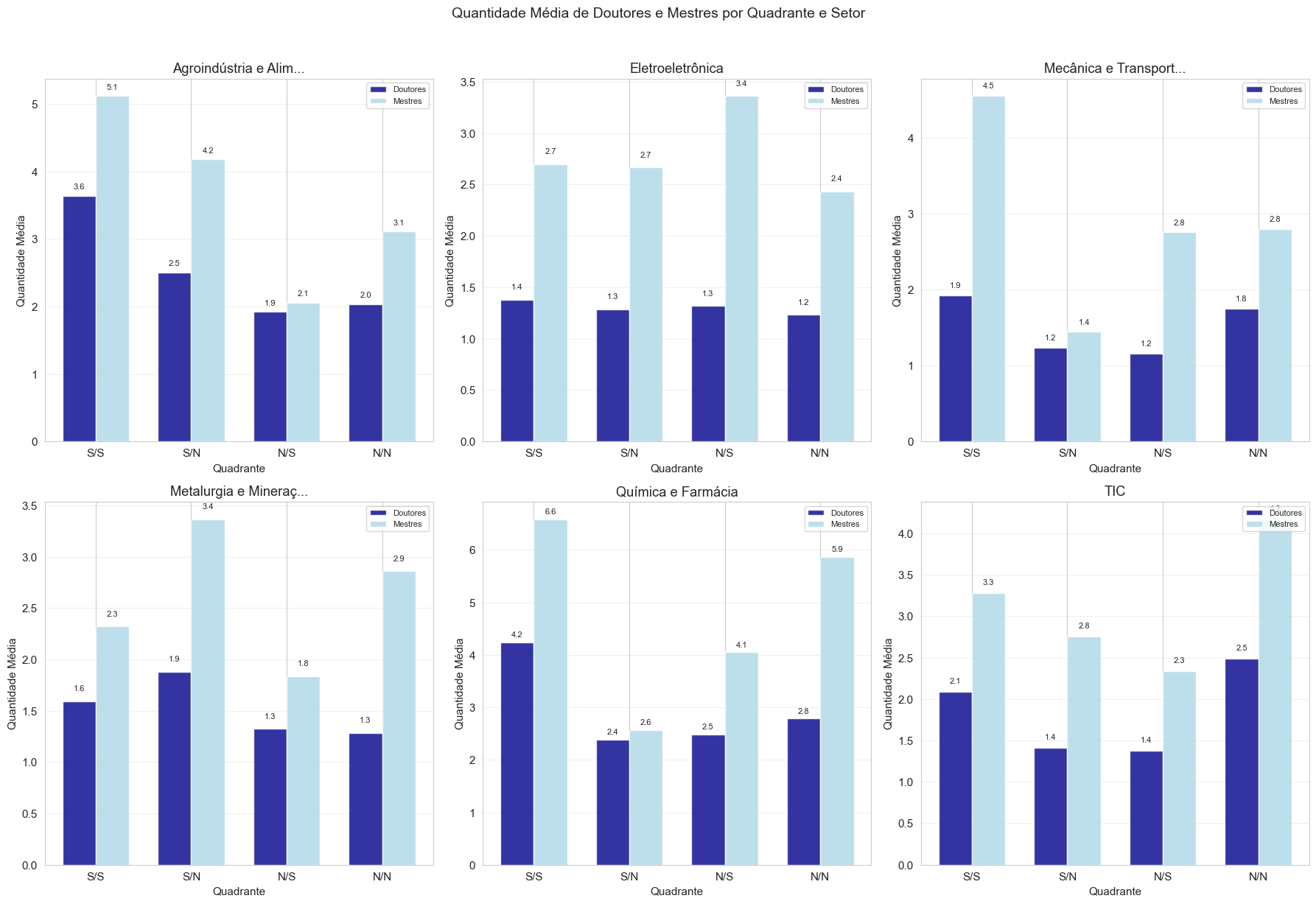
### 5.2. Análise 3.2: Mestres e Doutores por Quadrante e Setor

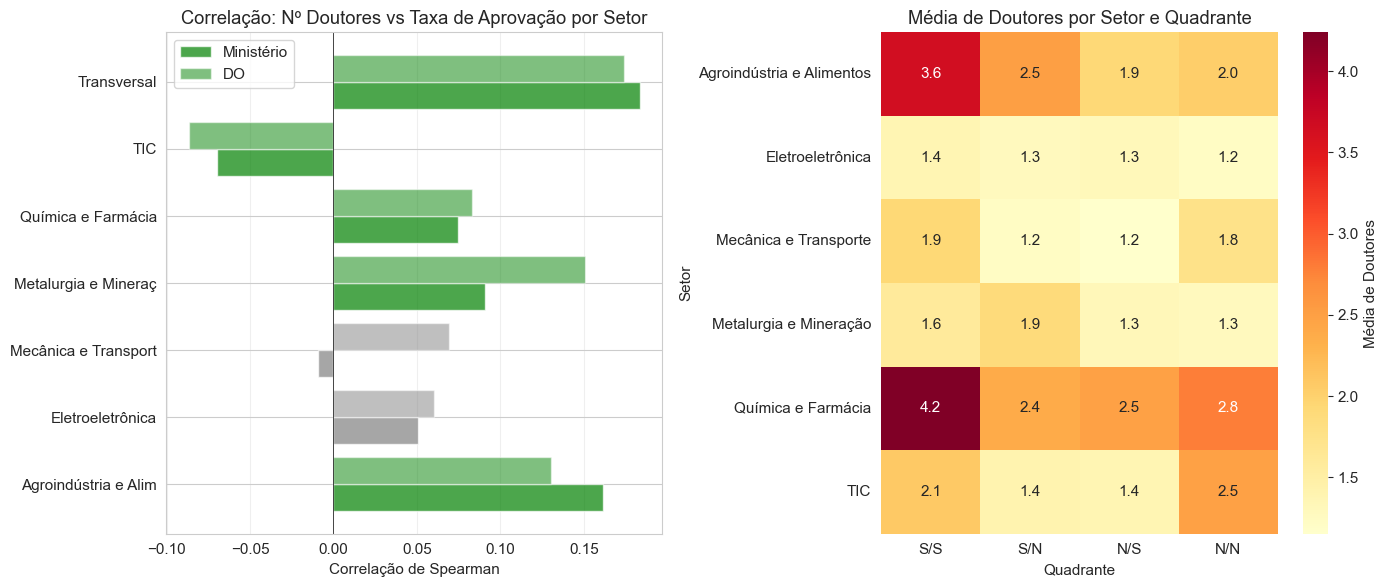
Esta análise reconhece que diferentes setores tecnológicos podem ter dinâmicas distintas quanto à importância da qualificação. Para cada setor, calcula-se a correlação de Spearman entre quantidade de doutores/mestres e taxa de aprovação.

A **correlação de Spearman** é usada por ser robusta a outliers e não assumir relação linear. Valores positivos indicam que mais qualificação aumenta aprovação, negativos indicam o contrário. O caso do TIC é particularmente interessante, mostrando correlação negativa - possivelmente porque projetos muito acadêmicos são vistos como menos práticos neste setor. A análise por **faixas de quantidade** (0, 1, 2-3, 4+) complementa mostrando como a taxa de aprovação varia em diferentes níveis de qualificação.

print(f"\n{'='\*73}")  
print("ANÁLISE 3.2: MESTRES E DOUTORES POR QUADRANTE E SETOR")  
print(f"{'='\*73}")  
  
# Setores para análise  
setores = ['Agroindústria e Alimentos', 'Eletroeletrônica', 'Mecânica e Transporte',   
 'Metalurgia e Mineração', 'Química e Farmácia', 'TIC', 'Transversal']  
  
# Estrutura para armazenar resultados  
resultados\_setor\_quad = {}  
  
print("\nQUANTIDADE MÉDIA DE DOUTORES E MESTRES POR QUADRANTE E SETOR")  
print("="\*73)  
  
for setor in setores:  
 df\_setor = df\_analise[df\_analise['setor'] == setor]  
   
 if len(df\_setor) < 30: # Skip setores com poucos dados  
 continue  
   
 print(f"\n{setor} (n={len(df\_setor):,})")  
 print("-"\*40)  
   
 resultados\_setor\_quad[setor] = {}  
   
 # Análise por quadrante  
 for q in ['S/S', 'S/N', 'N/S', 'N/N']:  
 df\_sq = df\_setor[df\_setor['quadrante'] == q]  
   
 if len(df\_sq) > 0:  
 media\_dout = df\_sq['quantdoutorproj'].mean()  
 media\_mest = df\_sq['quantmestreproj'].mean()  
 n\_proj = len(df\_sq)  
 pct\_proj = (n\_proj / len(df\_setor)) \* 100  
   
 resultados\_setor\_quad[setor][q] = {  
 'media\_doutores': media\_dout,  
 'media\_mestres': media\_mest,  
 'n\_projetos': n\_proj,  
 'pct\_projetos': pct\_proj  
 }  
   
 print(f" {q}: Dout={media\_dout:.1f}, Mest={media\_mest:.1f} ({n\_proj} proj, {pct\_proj:.1f}%)")  
  
print(f"\n{'='\*73}")  
print("CORRELAÇÃO: QUANTIDADE DE DOUTORES VS TAXA DE APROVAÇÃO POR SETOR")  
print("="\*73)  
  
correlacoes\_setor = {}  
  
for setor in setores:  
 df\_setor = df\_analise[df\_analise['setor'] == setor]  
   
 if len(df\_setor) < 30:  
 continue  
   
 print(f"\n{setor}:")  
 print("-"\*30)  
   
 # Usar valores ORIGINAIS, não categorizados  
 # Remover NaN antes de calcular correlação  
 df\_valid = df\_setor.dropna(subset=['quantdoutorproj', 'ministerio\_aprova', 'pesquisador\_aprova'])  
   
 if len(df\_valid) > 10 and df\_valid['quantdoutorproj'].var() > 0:  
 # Correlação com valores contínuos  
 corr\_min, p\_value\_min = stats.spearmanr(df\_valid['quantdoutorproj'],   
 df\_valid['ministerio\_aprova'])  
 corr\_do, p\_value\_do = stats.spearmanr(df\_valid['quantdoutorproj'],   
 df\_valid['pesquisador\_aprova'])  
   
 correlacoes\_setor[setor] = {  
 'corr\_ministerio': corr\_min,  
 'p\_value\_ministerio': p\_value\_min,  
 'corr\_do': corr\_do,  
 'p\_value\_do': p\_value\_do  
 }  
   
 print(f" Correlação Spearman:")  
 print(f" Ministério: r={corr\_min:.3f}, p={p\_value\_min:.4f}")  
 print(f" DO: r={corr\_do:.3f}, p={p\_value\_do:.4f}")  
   
 # Interpretação  
 if p\_value\_min < 0.05:  
 if corr\_min > 0:  
 print(f" → Mais doutores AUMENTA aprovação Ministério")  
 else:  
 print(f" → Mais doutores DIMINUI aprovação Ministério")  
 else:  
 print(f" Dados insuficientes ou sem variação")  
   
 # Análise por faixas (complementar)  
 df\_setor\_copy = df\_setor.copy()  
 df\_setor\_copy['faixa\_doutores'] = pd.cut(df\_setor\_copy['quantdoutorproj'],   
 bins=[-0.1, 0, 1, 3, 100],   
 labels=['0', '1', '2-3', '4+'])  
   
 print(f"\n Taxa de aprovação por faixa:")  
 for faixa in ['0', '1', '2-3', '4+']:  
 df\_faixa = df\_setor\_copy[df\_setor\_copy['faixa\_doutores'] == faixa]  
 if len(df\_faixa) > 0:  
 taxa\_min = df\_faixa['ministerio\_aprova'].mean() \* 100  
 taxa\_do = df\_faixa['pesquisador\_aprova'].mean() \* 100  
 n = len(df\_faixa)  
 print(f" {faixa:4} doutores (n={n:4}): Min={taxa\_min:.1f}%, DO={taxa\_do:.1f}%")  
  
# Visualizações  
fig, axes = plt.subplots(2, 3, figsize=(18, 12))  
axes = axes.ravel()  
  
# Preparar dados para visualização  
setores\_plot = [s for s in setores if s in resultados\_setor\_quad][:6] # Top 6 setores  
  
# Gráfico 1-6: Um para cada setor mostrando médias por quadrante  
for i, setor in enumerate(setores\_plot):  
 if i >= 6:  
 break  
   
 # Preparar dados do setor  
 quadrantes = ['S/S', 'S/N', 'N/S', 'N/N']  
 medias\_dout = []  
 medias\_mest = []  
   
 for q in quadrantes:  
 if q in resultados\_setor\_quad[setor]:  
 medias\_dout.append(resultados\_setor\_quad[setor][q]['media\_doutores'])  
 medias\_mest.append(resultados\_setor\_quad[setor][q]['media\_mestres'])  
 else:  
 medias\_dout.append(0)  
 medias\_mest.append(0)  
   
 x = np.arange(len(quadrantes))  
 width = 0.35  
   
 axes[i].bar(x - width/2, medias\_dout, width, label='Doutores', color='darkblue', alpha=0.8)  
 axes[i].bar(x + width/2, medias\_mest, width, label='Mestres', color='lightblue', alpha=0.8)  
   
 axes[i].set\_xlabel('Quadrante')  
 axes[i].set\_ylabel('Quantidade Média')  
 axes[i].set\_title(f'{setor[:20]}...' if len(setor) > 20 else setor)  
 axes[i].set\_xticks(x)  
 axes[i].set\_xticklabels(quadrantes)  
 axes[i].legend(loc='upper right', fontsize=8)  
 axes[i].grid(True, alpha=0.3, axis='y')  
   
 # Adicionar valores no topo das barras  
 for j, (d, m) in enumerate(zip(medias\_dout, medias\_mest)):  
 if d > 0:  
 axes[i].text(j - width/2, d + 0.1, f'{d:.1f}', ha='center', fontsize=8)  
 if m > 0:  
 axes[i].text(j + width/2, m + 0.1, f'{m:.1f}', ha='center', fontsize=8)  
  
plt.suptitle('Quantidade Média de Doutores e Mestres por Quadrante e Setor', fontsize=14, y=1.02)  
plt.tight\_layout()  
plt.show()  
  
# Segunda visualização: Correlação por setor  
fig, axes = plt.subplots(1, 2, figsize=(14, 6))  
  
# Preparar dados de correlação  
setores\_corr = list(correlacoes\_setor.keys())  
corr\_ministerio = [correlacoes\_setor[s]['corr\_ministerio'] for s in setores\_corr]  
corr\_do = [correlacoes\_setor[s]['corr\_do'] for s in setores\_corr]  
p\_val\_ministerio = [correlacoes\_setor[s]['p\_value\_ministerio'] for s in setores\_corr]  
p\_val\_do = [correlacoes\_setor[s]['p\_value\_do'] for s in setores\_corr]  
  
# Gráfico 1: Correlações  
y\_pos = np.arange(len(setores\_corr))  
colors\_min = ['green' if p < 0.05 else 'gray' for p in p\_val\_ministerio]  
colors\_do = ['green' if p < 0.05 else 'gray' for p in p\_val\_do]  
  
axes[0].barh(y\_pos - 0.2, corr\_ministerio, 0.4, label='Ministério', color=colors\_min, alpha=0.7)  
axes[0].barh(y\_pos + 0.2, corr\_do, 0.4, label='DO', color=colors\_do, alpha=0.5)  
axes[0].set\_yticks(y\_pos)  
axes[0].set\_yticklabels([s[:20] for s in setores\_corr])  
axes[0].set\_xlabel('Correlação de Spearman')  
axes[0].set\_title('Correlação: Nº Doutores vs Taxa de Aprovação por Setor')  
axes[0].axvline(x=0, color='black', linestyle='-', linewidth=0.5)  
axes[0].legend()  
axes[0].grid(True, alpha=0.3, axis='x')  
  
# Gráfico 2: Heatmap de média de doutores por setor e quadrante  
matriz\_doutores = pd.DataFrame(index=setores\_plot, columns=['S/S', 'S/N', 'N/S', 'N/N'])  
  
for setor in setores\_plot:  
 for q in ['S/S', 'S/N', 'N/S', 'N/N']:  
 if q in resultados\_setor\_quad[setor]:  
 matriz\_doutores.loc[setor, q] = resultados\_setor\_quad[setor][q]['media\_doutores']  
  
matriz\_doutores = matriz\_doutores.astype(float).fillna(0)  
sns.heatmap(matriz\_doutores, annot=True, fmt='.1f', cmap='YlOrRd', ax=axes[1],  
 cbar\_kws={'label': 'Média de Doutores'})  
axes[1].set\_title('Média de Doutores por Setor e Quadrante')  
axes[1].set\_xlabel('Quadrante')  
axes[1].set\_ylabel('Setor')  
  
plt.tight\_layout()  
plt.show()  
  
# Resumo dos insights  
print(f"\n{'='\*73}")  
print("PRINCIPAIS INSIGHTS DA ANÁLISE 3.2")  
print("="\*73)  
  
# Setor com maior diferença entre Q1 e Q4  
max\_diff\_setor = None  
max\_diff\_value = 0  
  
for setor in resultados\_setor\_quad:  
 if 'Q1' in resultados\_setor\_quad[setor] and 'Q4' in resultados\_setor\_quad[setor]:  
 diff = resultados\_setor\_quad[setor]['Q1']['media\_doutores'] - resultados\_setor\_quad[setor]['Q4']['media\_doutores']  
 if diff > max\_diff\_value:  
 max\_diff\_value = diff  
 max\_diff\_setor = setor  
  
if max\_diff\_setor:  
 print(f"\n1. MAIOR DIFERENÇA Q1 vs Q4:")  
 print(f" Setor: {max\_diff\_setor}")  
 print(f" Diferença: {max\_diff\_value:.1f} doutores a mais em Q1")  
  
# Correlações significativas  
print(f"\n2. CORRELAÇÕES SIGNIFICATIVAS (p < 0.05):")  
for setor in correlacoes\_setor:  
 if correlacoes\_setor[setor]['p\_value\_ministerio'] < 0.05:  
 print(f" {setor}: r={correlacoes\_setor[setor]['corr\_ministerio']:.3f} (Ministério)")  
  
# Padrões gerais  
print(f"\n3. PADRÕES OBSERVADOS:")  
q1\_avg\_dout = np.mean([resultados\_setor\_quad[s]['Q1']['media\_doutores']   
 for s in resultados\_setor\_quad if 'Q1' in resultados\_setor\_quad[s]])  
q4\_avg\_dout = np.mean([resultados\_setor\_quad[s]['Q4']['media\_doutores']   
 for s in resultados\_setor\_quad if 'Q4' in resultados\_setor\_quad[s]])  
  
print(f" Média geral de doutores em Q1: {q1\_avg\_dout:.1f}")  
print(f" Média geral de doutores em Q4: {q4\_avg\_dout:.1f}")  
print(f" Diferença: {q1\_avg\_dout - q4\_avg\_dout:.1f} doutores")

=========================================================================  
ANÁLISE 3.2: MESTRES E DOUTORES POR QUADRANTE E SETOR  
=========================================================================  
  
QUANTIDADE MÉDIA DE DOUTORES E MESTRES POR QUADRANTE E SETOR  
=========================================================================  
  
Agroindústria e Alimentos (n=8,140)  
----------------------------------------  
 S/S: Dout=3.6, Mest=5.1 (4235 proj, 52.0%)  
 S/N: Dout=2.5, Mest=4.2 (694 proj, 8.5%)  
 N/S: Dout=1.9, Mest=2.1 (614 proj, 7.5%)  
 N/N: Dout=2.0, Mest=3.1 (2597 proj, 31.9%)  
  
Eletroeletrônica (n=6,851)  
----------------------------------------  
 S/S: Dout=1.4, Mest=2.7 (3927 proj, 57.3%)  
 S/N: Dout=1.3, Mest=2.7 (157 proj, 2.3%)  
 N/S: Dout=1.3, Mest=3.4 (792 proj, 11.6%)  
 N/N: Dout=1.2, Mest=2.4 (1975 proj, 28.8%)  
  
Mecânica e Transporte (n=9,061)  
----------------------------------------  
 S/S: Dout=1.9, Mest=4.5 (4950 proj, 54.6%)  
 S/N: Dout=1.2, Mest=1.4 (404 proj, 4.5%)  
 N/S: Dout=1.2, Mest=2.8 (1230 proj, 13.6%)  
 N/N: Dout=1.8, Mest=2.8 (2477 proj, 27.3%)  
  
Metalurgia e Mineração (n=4,989)  
----------------------------------------  
 S/S: Dout=1.6, Mest=2.3 (2858 proj, 57.3%)  
 S/N: Dout=1.9, Mest=3.4 (277 proj, 5.6%)  
 N/S: Dout=1.3, Mest=1.8 (514 proj, 10.3%)  
 N/N: Dout=1.3, Mest=2.9 (1340 proj, 26.9%)  
  
Química e Farmácia (n=12,832)  
----------------------------------------  
 S/S: Dout=4.2, Mest=6.6 (9410 proj, 73.3%)  
 S/N: Dout=2.4, Mest=2.6 (424 proj, 3.3%)  
 N/S: Dout=2.5, Mest=4.1 (710 proj, 5.5%)  
 N/N: Dout=2.8, Mest=5.9 (2288 proj, 17.8%)  
  
TIC (n=18,424)  
----------------------------------------  
 S/S: Dout=2.1, Mest=3.3 (7600 proj, 41.3%)  
 S/N: Dout=1.4, Mest=2.8 (2268 proj, 12.3%)  
 N/S: Dout=1.4, Mest=2.3 (1408 proj, 7.6%)  
 N/N: Dout=2.5, Mest=4.2 (7148 proj, 38.8%)  
  
Transversal (n=7,454)  
----------------------------------------  
 S/S: Dout=2.1, Mest=3.1 (3487 proj, 46.8%)  
 S/N: Dout=1.4, Mest=1.8 (879 proj, 11.8%)  
 N/S: Dout=1.5, Mest=2.8 (630 proj, 8.5%)  
 N/N: Dout=1.3, Mest=2.1 (2458 proj, 33.0%)  
  
=========================================================================  
CORRELAÇÃO: QUANTIDADE DE DOUTORES VS TAXA DE APROVAÇÃO POR SETOR  
=========================================================================  
  
Agroindústria e Alimentos:  
------------------------------  
 Correlação Spearman:  
 Ministério: r=0.162, p=0.0000  
 DO: r=0.131, p=0.0000  
 → Mais doutores AUMENTA aprovação Ministério  
  
 Taxa de aprovação por faixa:  
 1 doutores (n= 990): Min=65.5%, DO=75.1%  
 2-3 doutores (n= 501): Min=77.0%, DO=79.4%  
 4+ doutores (n= 430): Min=82.6%, DO=89.3%  
  
Eletroeletrônica:  
------------------------------  
 Correlação Spearman:  
 Ministério: r=0.051, p=0.2539  
 DO: r=0.060, p=0.1765  
  
 Taxa de aprovação por faixa:  
 1 doutores (n= 412): Min=73.3%, DO=67.7%  
 2-3 doutores (n= 77): Min=75.3%, DO=72.7%  
 4+ doutores (n= 20): Min=90.0%, DO=80.0%  
  
Mecânica e Transporte:  
------------------------------  
 Correlação Spearman:  
 Ministério: r=-0.009, p=0.8146  
 DO: r=0.069, p=0.0779  
  
 Taxa de aprovação por faixa:  
 1 doutores (n= 447): Min=87.7%, DO=79.6%  
 2-3 doutores (n= 152): Min=87.5%, DO=86.2%  
 4+ doutores (n= 48): Min=85.4%, DO=83.3%  
  
Metalurgia e Mineração:  
------------------------------  
 Correlação Spearman:  
 Ministério: r=0.091, p=0.0178  
 DO: r=0.151, p=0.0001  
 → Mais doutores AUMENTA aprovação Ministério  
  
 Taxa de aprovação por faixa:  
 1 doutores (n= 440): Min=75.0%, DO=72.5%  
 2-3 doutores (n= 212): Min=82.1%, DO=85.4%  
 4+ doutores (n= 26): Min=88.5%, DO=88.5%  
  
Química e Farmácia:  
------------------------------  
 Correlação Spearman:  
 Ministério: r=0.075, p=0.0000  
 DO: r=0.083, p=0.0000  
 → Mais doutores AUMENTA aprovação Ministério  
  
 Taxa de aprovação por faixa:  
 1 doutores (n=1740): Min=87.9%, DO=83.4%  
 2-3 doutores (n=1318): Min=87.7%, DO=80.8%  
 4+ doutores (n=1196): Min=93.7%, DO=90.9%  
  
TIC:  
------------------------------  
 Correlação Spearman:  
 Ministério: r=-0.070, p=0.0022  
 DO: r=-0.087, p=0.0002  
 → Mais doutores DIMINUI aprovação Ministério  
  
 Taxa de aprovação por faixa:  
 1 doutores (n=1229): Min=50.9%, DO=57.6%  
 2-3 doutores (n= 425): Min=45.4%, DO=52.9%  
 4+ doutores (n= 250): Min=41.2%, DO=44.0%  
  
Transversal:  
------------------------------  
 Correlação Spearman:  
 Ministério: r=0.184, p=0.0000  
 DO: r=0.174, p=0.0000  
 → Mais doutores AUMENTA aprovação Ministério  
  
 Taxa de aprovação por faixa:  
 1 doutores (n= 554): Min=70.6%, DO=75.1%  
 2-3 doutores (n= 233): Min=86.3%, DO=89.3%  
 4+ doutores (n= 79): Min=89.9%, DO=91.1%





=========================================================================  
PRINCIPAIS INSIGHTS DA ANÁLISE 3.2  
=========================================================================  
  
2. CORRELAÇÕES SIGNIFICATIVAS (p < 0.05):  
 Agroindústria e Alimentos: r=0.162 (Ministério)  
 Metalurgia e Mineração: r=0.091 (Ministério)  
 Química e Farmácia: r=0.075 (Ministério)  
 TIC: r=-0.070 (Ministério)  
 Transversal: r=0.184 (Ministério)  
  
3. PADRÕES OBSERVADOS:  
 Média geral de doutores em Q1: nan  
 Média geral de doutores em Q4: nan  
 Diferença: nan doutores

## 6. Análise 4: Percentual de Gastos (RH e Terceiros) por Setor

Esta análise investiga se a composição dos gastos do projeto - especificamente a proporção destinada a recursos humanos internos versus serviços terceirizados - afeta as chances de aprovação, reconhecendo que essa relação pode variar entre setores.

A regressão logística captura como cada ponto percentual adicional gasto em RH ou terceirização afeta a probabilidade de aprovação. Os resultados revelam padrões setoriais distintos: em Química e Farmácia, maior terceirização aumenta aprovação (talvez indicando colaboração com laboratórios especializados), enquanto em TIC o efeito é oposto (possivelmente valorizando desenvolvimento interno).

print(f"\n{'='\*73}")  
print("ANÁLISE 4: PERCENTUAL DE GASTOS COM RH E TERCEIROS POR SETOR")  
print(f"{'='\*73}")  
  
# Calcular percentual de gastos  
df\_analise['perc\_gasto\_rh'] = (df\_analise['valorrhproj'] / df\_analise['valortotalproj'].replace(0, np.nan)) \* 100  
df\_analise['perc\_gasto\_terceiros'] = df\_analise['percservterceirosproj'] \* 100  
  
# Setores para análise  
setores = ['Agroindústria e Alimentos', 'Eletroeletrônica', 'Mecânica e Transporte',   
 'Metalurgia e Mineração', 'Química e Farmácia', 'TIC', 'Transversal']  
  
resultados\_setor = {}  
  
for setor in setores:  
 df\_setor = df\_analise[df\_analise['setor'] == setor].copy()  
   
 if len(df\_setor) < 30: # Skip se poucos dados  
 continue  
   
 print(f"\n{'='\*60}")  
 print(f"SETOR: {setor} (n={len(df\_setor)})")  
 print(f"{'='\*60}")  
   
 resultados\_setor[setor] = {}  
   
 # Análise para RH  
 print("\nGASTOS COM RH:")  
 print("-"\*30)  
   
 for aprovador, coluna in [('Ministério', 'ministerio\_aprova'), ('DO', 'pesquisador\_aprova')]:  
 # Regressão logística  
 X\_rh = df\_setor[['perc\_gasto\_rh']].fillna(0)  
 y = df\_setor[coluna]  
   
 if len(np.unique(y)) > 1: # Verificar se há variação  
 model = LogisticRegression(random\_state=42, max\_iter=1000)  
 model.fit(StandardScaler().fit\_transform(X\_rh), y)  
 coef = model.coef\_[0][0]  
   
 # Correlação simples  
 corr, p\_value = stats.spearmanr(df\_setor['perc\_gasto\_rh'].fillna(0), y)  
   
 resultados\_setor[setor][f'rh\_{aprovador}'] = {  
 'coef': coef,  
 'corr': corr,  
 'p\_value': p\_value  
 }  
   
 print(f"\n {aprovador}:")  
 print(f" Coeficiente: {coef:.4f}")  
 print(f" Correlação: {corr:.4f}")  
 print(f" P-value: {p\_value:.4f}")  
   
 if p\_value < 0.05:  
 if coef > 0:  
 print(f" ✓ Maior % RH AUMENTA aprovação")  
 else:  
 print(f" ✓ Maior % RH DIMINUI aprovação")  
 else:  
 print(f" ✗ Sem efeito significativo")  
   
 # Análise para Terceiros  
 print("\nGASTOS COM TERCEIROS:")  
 print("-"\*30)  
   
 for aprovador, coluna in [('Ministério', 'ministerio\_aprova'), ('DO', 'pesquisador\_aprova')]:  
 # Regressão logística  
 X\_terc = df\_setor[['perc\_gasto\_terceiros']].fillna(0)  
 y = df\_setor[coluna]  
   
 if len(np.unique(y)) > 1:  
 model = LogisticRegression(random\_state=42, max\_iter=1000)  
 model.fit(StandardScaler().fit\_transform(X\_terc), y)  
 coef = model.coef\_[0][0]  
   
 # Correlação simples  
 corr, p\_value = stats.spearmanr(df\_setor['perc\_gasto\_terceiros'].fillna(0), y)  
   
 resultados\_setor[setor][f'terceiros\_{aprovador}'] = {  
 'coef': coef,  
 'corr': corr,  
 'p\_value': p\_value  
 }  
   
 print(f"\n {aprovador}:")  
 print(f" Coeficiente: {coef:.4f}")  
 print(f" Correlação: {corr:.4f}")  
 print(f" P-value: {p\_value:.4f}")  
   
 if p\_value < 0.05:  
 if coef > 0:  
 print(f" ✓ Maior % Terceiros AUMENTA aprovação")  
 else:  
 print(f" ✓ Maior % Terceiros DIMINUI aprovação")  
 else:  
 print(f" ✗ Sem efeito significativo")  
  
# Visualização comparativa  
fig, axes = plt.subplots(2, 2, figsize=(15, 10))  
  
# Preparar dados para visualização  
setores\_plot = [s for s in setores if s in resultados\_setor]  
  
# Gráfico 1: Coeficientes RH - Ministério  
coefs\_rh\_min = [resultados\_setor[s].get('rh\_Ministério', {}).get('coef', 0) for s in setores\_plot]  
p\_vals\_rh\_min = [resultados\_setor[s].get('rh\_Ministério', {}).get('p\_value', 1) for s in setores\_plot]  
colors\_rh\_min = ['green' if p < 0.05 and c > 0 else 'red' if p < 0.05 and c < 0 else 'gray'   
 for c, p in zip(coefs\_rh\_min, p\_vals\_rh\_min)]  
  
axes[0,0].barh(range(len(setores\_plot)), coefs\_rh\_min, color=colors\_rh\_min, alpha=0.7)  
axes[0,0].set\_yticks(range(len(setores\_plot)))  
axes[0,0].set\_yticklabels([s[:15] for s in setores\_plot])  
axes[0,0].set\_xlabel('Coeficiente')  
axes[0,0].set\_title('Impacto % RH na Aprovação - Ministério')  
axes[0,0].axvline(x=0, color='black', linestyle='-', linewidth=0.5)  
axes[0,0].grid(True, alpha=0.3)  
  
# Gráfico 2: Coeficientes Terceiros - Ministério  
coefs\_terc\_min = [resultados\_setor[s].get('terceiros\_Ministério', {}).get('coef', 0) for s in setores\_plot]  
p\_vals\_terc\_min = [resultados\_setor[s].get('terceiros\_Ministério', {}).get('p\_value', 1) for s in setores\_plot]  
colors\_terc\_min = ['green' if p < 0.05 and c > 0 else 'red' if p < 0.05 and c < 0 else 'gray'   
 for c, p in zip(coefs\_terc\_min, p\_vals\_terc\_min)]  
  
axes[0,1].barh(range(len(setores\_plot)), coefs\_terc\_min, color=colors\_terc\_min, alpha=0.7)  
axes[0,1].set\_yticks(range(len(setores\_plot)))  
axes[0,1].set\_yticklabels([s[:15] for s in setores\_plot])  
axes[0,1].set\_xlabel('Coeficiente')  
axes[0,1].set\_title('Impacto % Terceiros na Aprovação - Ministério')  
axes[0,1].axvline(x=0, color='black', linestyle='-', linewidth=0.5)  
axes[0,1].grid(True, alpha=0.3)  
  
# Gráfico 3: Coeficientes RH - DO  
coefs\_rh\_do = [resultados\_setor[s].get('rh\_DO', {}).get('coef', 0) for s in setores\_plot]  
p\_vals\_rh\_do = [resultados\_setor[s].get('rh\_DO', {}).get('p\_value', 1) for s in setores\_plot]  
colors\_rh\_do = ['green' if p < 0.05 and c > 0 else 'red' if p < 0.05 and c < 0 else 'gray'   
 for c, p in zip(coefs\_rh\_do, p\_vals\_rh\_do)]  
  
axes[1,0].barh(range(len(setores\_plot)), coefs\_rh\_do, color=colors\_rh\_do, alpha=0.7)  
axes[1,0].set\_yticks(range(len(setores\_plot)))  
axes[1,0].set\_yticklabels([s[:15] for s in setores\_plot])  
axes[1,0].set\_xlabel('Coeficiente')  
axes[1,0].set\_title('Impacto % RH na Aprovação - DO')  
axes[1,0].axvline(x=0, color='black', linestyle='-', linewidth=0.5)  
axes[1,0].grid(True, alpha=0.3)  
  
# Gráfico 4: Coeficientes Terceiros - DO  
coefs\_terc\_do = [resultados\_setor[s].get('terceiros\_DO', {}).get('coef', 0) for s in setores\_plot]  
p\_vals\_terc\_do = [resultados\_setor[s].get('terceiros\_DO', {}).get('p\_value', 1) for s in setores\_plot]  
colors\_terc\_do = ['green' if p < 0.05 and c > 0 else 'red' if p < 0.05 and c < 0 else 'gray'   
 for c, p in zip(coefs\_terc\_do, p\_vals\_terc\_do)]  
  
axes[1,1].barh(range(len(setores\_plot)), coefs\_terc\_do, color=colors\_terc\_do, alpha=0.7)  
axes[1,1].set\_yticks(range(len(setores\_plot)))  
axes[1,1].set\_yticklabels([s[:15] for s in setores\_plot])  
axes[1,1].set\_xlabel('Coeficiente')  
axes[1,1].set\_title('Impacto % Terceiros na Aprovação - DO')  
axes[1,1].axvline(x=0, color='black', linestyle='-', linewidth=0.5)  
axes[1,1].grid(True, alpha=0.3)  
  
# Adicionar legenda  
from matplotlib.patches import Patch  
legend\_elements = [Patch(facecolor='green', alpha=0.7, label='Aumenta aprovação (p<0.05)'),  
 Patch(facecolor='red', alpha=0.7, label='Diminui aprovação (p<0.05)'),  
 Patch(facecolor='gray', alpha=0.7, label='Sem efeito significativo')]  
fig.legend(handles=legend\_elements, loc='upper center', ncol=3, bbox\_to\_anchor=(0.5, -0.05))  
  
plt.tight\_layout()  
plt.show()

=========================================================================  
ANÁLISE 4: PERCENTUAL DE GASTOS COM RH E TERCEIROS POR SETOR  
=========================================================================  
  
============================================================  
SETOR: Agroindústria e Alimentos (n=8140)  
============================================================  
  
GASTOS COM RH:  
------------------------------  
  
 Ministério:  
 Coeficiente: 0.3827  
 Correlação: 0.1446  
 P-value: 0.0000  
 ✓ Maior % RH AUMENTA aprovação  
  
 DO:  
 Coeficiente: 0.2258  
 Correlação: 0.0442  
 P-value: 0.0001  
 ✓ Maior % RH AUMENTA aprovação  
  
GASTOS COM TERCEIROS:  
------------------------------  
  
 Ministério:  
 Coeficiente: 0.0763  
 Correlação: 0.1281  
 P-value: 0.0000  
 ✓ Maior % Terceiros AUMENTA aprovação  
  
 DO:  
 Coeficiente: 0.2932  
 Correlação: 0.2542  
 P-value: 0.0000  
 ✓ Maior % Terceiros AUMENTA aprovação  
  
============================================================  
SETOR: Eletroeletrônica (n=6851)  
============================================================  
  
GASTOS COM RH:  
------------------------------  
  
 Ministério:  
 Coeficiente: -0.0789  
 Correlação: -0.0296  
 P-value: 0.0141  
 ✓ Maior % RH DIMINUI aprovação  
  
 DO:  
 Coeficiente: -0.1425  
 Correlação: -0.0642  
 P-value: 0.0000  
 ✓ Maior % RH DIMINUI aprovação  
  
GASTOS COM TERCEIROS:  
------------------------------  
  
 Ministério:  
 Coeficiente: -0.2638  
 Correlação: -0.1376  
 P-value: 0.0000  
 ✓ Maior % Terceiros DIMINUI aprovação  
  
 DO:  
 Coeficiente: -0.1491  
 Correlação: -0.0881  
 P-value: 0.0000  
 ✓ Maior % Terceiros DIMINUI aprovação  
  
============================================================  
SETOR: Mecânica e Transporte (n=9061)  
============================================================  
  
GASTOS COM RH:  
------------------------------  
  
 Ministério:  
 Coeficiente: 0.1664  
 Correlação: 0.0282  
 P-value: 0.0073  
 ✓ Maior % RH AUMENTA aprovação  
  
 DO:  
 Coeficiente: 0.0515  
 Correlação: -0.0329  
 P-value: 0.0017  
 ✓ Maior % RH AUMENTA aprovação  
  
GASTOS COM TERCEIROS:  
------------------------------  
  
 Ministério:  
 Coeficiente: -0.2131  
 Correlação: -0.0291  
 P-value: 0.0057  
 ✓ Maior % Terceiros DIMINUI aprovação  
  
 DO:  
 Coeficiente: -0.0936  
 Correlação: 0.0311  
 P-value: 0.0030  
 ✓ Maior % Terceiros DIMINUI aprovação  
  
============================================================  
SETOR: Metalurgia e Mineração (n=4989)  
============================================================  
  
GASTOS COM RH:  
------------------------------  
  
 Ministério:  
 Coeficiente: 0.2217  
 Correlação: 0.1096  
 P-value: 0.0000  
 ✓ Maior % RH AUMENTA aprovação  
  
 DO:  
 Coeficiente: 0.0802  
 Correlação: 0.0321  
 P-value: 0.0233  
 ✓ Maior % RH AUMENTA aprovação  
  
GASTOS COM TERCEIROS:  
------------------------------  
  
 Ministério:  
 Coeficiente: -0.3946  
 Correlação: -0.1968  
 P-value: 0.0000  
 ✓ Maior % Terceiros DIMINUI aprovação  
  
 DO:  
 Coeficiente: -0.2560  
 Correlação: -0.1097  
 P-value: 0.0000  
 ✓ Maior % Terceiros DIMINUI aprovação  
  
============================================================  
SETOR: Química e Farmácia (n=12832)  
============================================================  
  
GASTOS COM RH:  
------------------------------  
  
 Ministério:  
 Coeficiente: -0.1536  
 Correlação: -0.0955  
 P-value: 0.0000  
 ✓ Maior % RH DIMINUI aprovação  
  
 DO:  
 Coeficiente: -0.2119  
 Correlação: -0.1242  
 P-value: 0.0000  
 ✓ Maior % RH DIMINUI aprovação  
  
GASTOS COM TERCEIROS:  
------------------------------  
  
 Ministério:  
 Coeficiente: 0.2063  
 Correlação: 0.1352  
 P-value: 0.0000  
 ✓ Maior % Terceiros AUMENTA aprovação  
  
 DO:  
 Coeficiente: 0.2101  
 Correlação: 0.1459  
 P-value: 0.0000  
 ✓ Maior % Terceiros AUMENTA aprovação  
  
============================================================  
SETOR: TIC (n=18424)  
============================================================  
  
GASTOS COM RH:  
------------------------------  
  
 Ministério:  
 Coeficiente: 0.0738  
 Correlação: 0.0461  
 P-value: 0.0000  
 ✓ Maior % RH AUMENTA aprovação  
  
 DO:  
 Coeficiente: 0.0668  
 Correlação: 0.0402  
 P-value: 0.0000  
 ✓ Maior % RH AUMENTA aprovação  
  
GASTOS COM TERCEIROS:  
------------------------------  
  
 Ministério:  
 Coeficiente: -0.1124  
 Correlação: -0.0653  
 P-value: 0.0000  
 ✓ Maior % Terceiros DIMINUI aprovação  
  
 DO:  
 Coeficiente: -0.1663  
 Correlação: -0.0847  
 P-value: 0.0000  
 ✓ Maior % Terceiros DIMINUI aprovação  
  
============================================================  
SETOR: Transversal (n=7454)  
============================================================  
  
GASTOS COM RH:  
------------------------------  
  
 Ministério:  
 Coeficiente: 0.0508  
 Correlação: -0.0106  
 P-value: 0.3597  
 ✗ Sem efeito significativo  
  
 DO:  
 Coeficiente: -0.0134  
 Correlação: -0.0368  
 P-value: 0.0015  
 ✓ Maior % RH DIMINUI aprovação  
  
GASTOS COM TERCEIROS:  
------------------------------  
  
 Ministério:  
 Coeficiente: -0.0755  
 Correlação: 0.0286  
 P-value: 0.0137  
 ✓ Maior % Terceiros DIMINUI aprovação  
  
 DO:  
 Coeficiente: -0.0124  
 Correlação: 0.0475  
 P-value: 0.0000  
 ✓ Maior % Terceiros DIMINUI aprovação



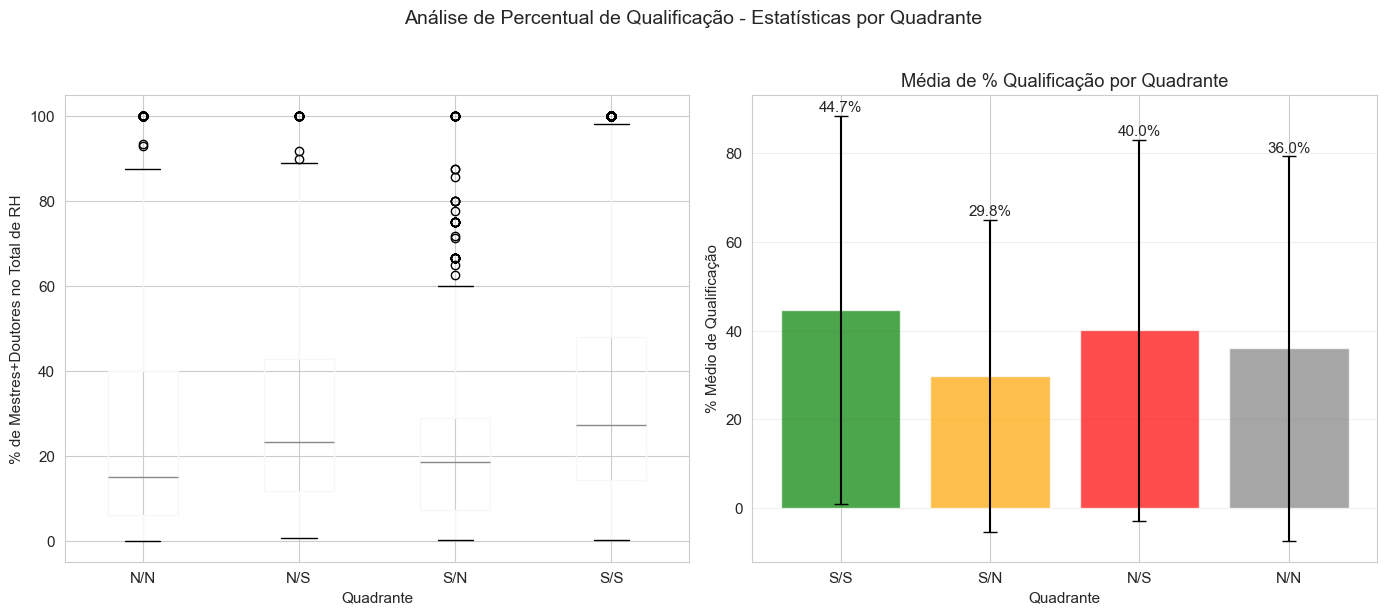
## Análise 4.1: Percentual de Qualificação (M+D) no Total de RH

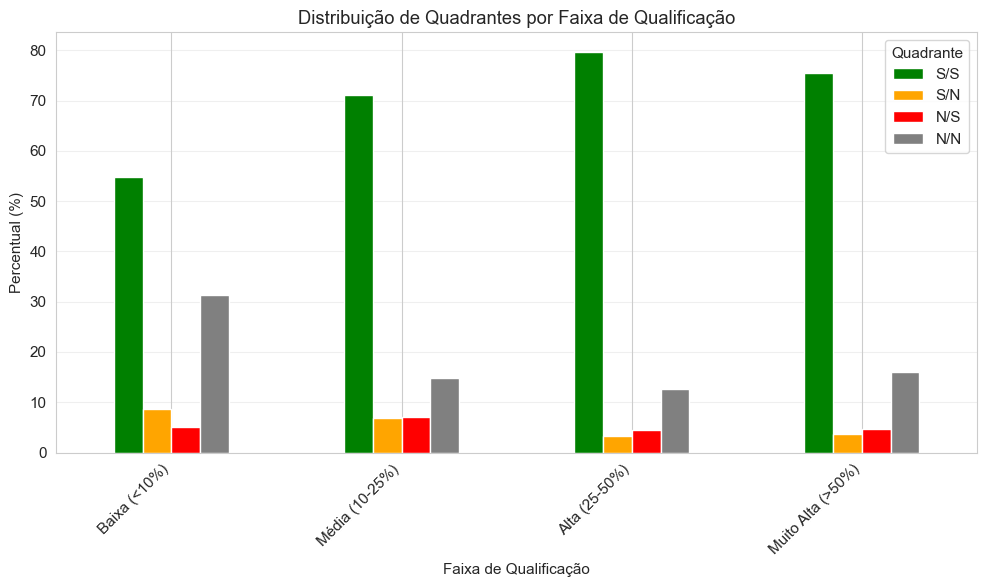
Esta análise final examina não apenas a quantidade absoluta de mestres e doutores, mas sua proporção em relação ao total de recursos humanos do projeto. Um projeto pode ter muitos doutores mas ainda assim ter baixo percentual de qualificação se a equipe total for muito grande.

A análise por **faixas de qualificação** (<10%, 10-25%, 25-50%, >50%) mostra de forma clara como projetos com diferentes níveis de “densidade de qualificação” se distribuem entre os quadrantes, revelando que existe um ponto ótimo - projetos com 25-50% de qualificados têm a maior taxa de dupla aprovação (S/S).

print(f"\n{'='\*73}")  
print("ANÁLISE 4.1: PERCENTUAL DE QUALIFICAÇÃO (M+D) NO TOTAL DE RH")  
print(f"{'='\*73}")  
  
# Criar variável de percentual de qualificação  
df\_analise['total\_qualificados'] = df\_analise['quantmestreproj'] + df\_analise['quantdoutorproj']  
df\_analise['perc\_qualificacao'] = (df\_analise['total\_qualificados'] / df\_analise['quantrhproj'].replace(0, np.nan)) \* 100  
  
# Estatísticas descritivas  
print("\nESTATÍSTICAS DESCRITIVAS")  
print("-"\*60)  
print(f"Percentual médio de qualificação: {df\_analise['perc\_qualificacao'].mean():.1f}%")  
print(f"Mediana: {df\_analise['perc\_qualificacao'].median():.1f}%")  
print(f"Desvio padrão: {df\_analise['perc\_qualificacao'].std():.1f}%")  
print(f"Mínimo: {df\_analise['perc\_qualificacao'].min():.1f}%")  
print(f"Máximo: {df\_analise['perc\_qualificacao'].max():.1f}%")  
  
# Análise por quadrante  
print(f"\n{'='\*60}")  
print("PERCENTUAL DE QUALIFICAÇÃO POR QUADRANTE")  
print("-"\*60)  
  
stats\_por\_quadrante = {}  
  
for q in ['S/S', 'S/N', 'N/S', 'N/N']:  
 df\_q = df\_analise[df\_analise['quadrante'] == q]  
   
 media = df\_q['perc\_qualificacao'].mean()  
 mediana = df\_q['perc\_qualificacao'].median()  
 std = df\_q['perc\_qualificacao'].std()  
   
 # Projetos com alta qualificação (>50% de M+D)  
 alta\_qual = (df\_q['perc\_qualificacao'] > 50).mean() \* 100  
   
 stats\_por\_quadrante[q] = {  
 'media': media,  
 'mediana': mediana,  
 'std': std,  
 'alta\_qual': alta\_qual  
 }  
   
 print(f"\n{q}:")  
 print(f" Média: {media:.1f}%")  
 print(f" Mediana: {mediana:.1f}%")  
 print(f" Desvio padrão: {std:.1f}%")  
 print(f" % projetos com >50% qualificados: {alta\_qual:.1f}%")  
  
# Teste estatístico ANOVA para diferenças entre quadrantes  
from scipy.stats import f\_oneway  
  
groups = [df\_analise[df\_analise['quadrante'] == q]['perc\_qualificacao'].dropna()   
 for q in ['S/S', 'S/N', 'N/S', 'N/N']]  
f\_stat, p\_value = f\_oneway(\*groups)  
  
  
# Análise de importância relativa  
print(f"\n{'='\*60}")  
print("IMPORTÂNCIA DO PERCENTUAL DE QUALIFICAÇÃO")  
print("-"\*60)  
  
# Regressão logística para cada quadrante  
from sklearn.linear\_model import LogisticRegression  
from sklearn.preprocessing import StandardScaler  
  
X = df\_analise[['perc\_qualificacao']].fillna(0)  
X\_scaled = StandardScaler().fit\_transform(X)  
  
print("\nRegressão Logística - Impacto na probabilidade de cada quadrante:")  
  
for q in ['S/S', 'S/N', 'N/S', 'N/N']:  
 y = (df\_analise['quadrante'] == q).astype(int)  
   
 model = LogisticRegression(random\_state=42, max\_iter=1000)  
 model.fit(X\_scaled, y)  
   
 coef = model.coef\_[0][0]  
 odds\_ratio = np.exp(coef)  
   
 print(f"\n{q}:")  
 print(f" Coeficiente: {coef:.3f}")  
 print(f" Odds Ratio: {odds\_ratio:.3f}")  
   
 if coef > 0.1:  
 print(f" → Alto % qualificação AUMENTA chance de {q}")  
 elif coef < -0.1:  
 print(f" → Alto % qualificação DIMINUI chance de {q}")  
 else:  
 print(f" → % qualificação tem pouco efeito em {q}")  
  
# Análise por faixas  
print(f"\n{'='\*60}")  
print("DISTRIBUIÇÃO POR FAIXAS DE QUALIFICAÇÃO")  
print("-"\*60)  
  
df\_analise['faixa\_qualificacao'] = pd.cut(df\_analise['perc\_qualificacao'],  
 bins=[0, 10, 25, 50, 100],  
 labels=['Baixa (<10%)', 'Média (10-25%)',   
 'Alta (25-50%)', 'Muito Alta (>50%)'])  
  
for faixa in ['Baixa (<10%)', 'Média (10-25%)', 'Alta (25-50%)', 'Muito Alta (>50%)']:  
 df\_faixa = df\_analise[df\_analise['faixa\_qualificacao'] == faixa]  
 if len(df\_faixa) > 0:  
 print(f"\n{faixa} (n={len(df\_faixa):,}):")  
 dist = df\_faixa['quadrante'].value\_counts(normalize=True) \* 100  
 for q in ['S/S', 'S/N', 'N/S', 'N/N']:  
 if q in dist.index:  
 print(f" {q}: {dist[q]:.1f}%")  
  
# FIGURA 1: Boxplot e Média  
fig1, axes1 = plt.subplots(1, 2, figsize=(14, 6))  
  
# Gráfico 1: Boxplot por quadrante  
df\_plot = df\_analise[df\_analise['perc\_qualificacao'] <= 100] # Remover outliers extremos  
df\_plot.boxplot(column='perc\_qualificacao', by='quadrante', ax=axes1[0])  
axes1[0].set\_title('Distribuição do % de Qualificação por Quadrante')  
axes1[0].set\_xlabel('Quadrante')  
axes1[0].set\_ylabel('% de Mestres+Doutores no Total de RH')  
plt.sca(axes1[0])  
plt.title('')  
  
# Gráfico 2: Média e erro padrão por quadrante  
means = [stats\_por\_quadrante[q]['media'] for q in ['S/S', 'S/N', 'N/S', 'N/N']]  
stds = [stats\_por\_quadrante[q]['std'] for q in ['S/S', 'S/N', 'N/S', 'N/N']]  
quadrantes = ['S/S', 'S/N', 'N/S', 'N/N']  
  
axes1[1].bar(quadrantes, means, yerr=stds, capsize=5,   
 color=['green', 'orange', 'red', 'gray'], alpha=0.7)  
axes1[1].set\_title('Média de % Qualificação por Quadrante')  
axes1[1].set\_xlabel('Quadrante')  
axes1[1].set\_ylabel('% Médio de Qualificação')  
axes1[1].grid(True, alpha=0.3, axis='y')  
  
# Adicionar valores no topo  
for i, (m, s) in enumerate(zip(means, stds)):  
 axes1[1].text(i, m + s + 1, f'{m:.1f}%', ha='center')  
  
plt.suptitle('Análise de Percentual de Qualificação - Estatísticas por Quadrante', fontsize=14, y=1.02)  
plt.tight\_layout()  
plt.show()  
  
# FIGURA 2: Distribuição por faixas  
fig2, ax2 = plt.subplots(1, 1, figsize=(10, 6))  
  
# Taxa de quadrantes por faixa  
faixa\_quad = pd.crosstab(df\_analise['faixa\_qualificacao'],   
 df\_analise['quadrante'],   
 normalize='index') \* 100  
  
faixa\_quad[['S/S', 'S/N', 'N/S', 'N/N']].plot(kind='bar', ax=ax2,   
 color=['green', 'orange', 'red', 'gray'])  
ax2.set\_title('Distribuição de Quadrantes por Faixa de Qualificação')  
ax2.set\_xlabel('Faixa de Qualificação')  
ax2.set\_ylabel('Percentual (%)')  
ax2.set\_xticklabels(ax2.get\_xticklabels(), rotation=45, ha='right')  
ax2.legend(title='Quadrante')  
ax2.grid(True, alpha=0.3, axis='y')  
  
plt.tight\_layout()  
plt.show()  
  
# Insights principais  
print(f"\n{'='\*80}")  
print("PRINCIPAIS INSIGHTS")  
print("-"\*60)  
  
# Comparação S/S vs N/N  
diff\_s\_s\_n\_n = stats\_por\_quadrante['S/S']['media'] - stats\_por\_quadrante['N/N']['media']  
print(f"\n1. DIFERENÇA S/S vs N/N:")  
print(f" Média S/S: {stats\_por\_quadrante['S/S']['media']:.1f}%")  
print(f" Média N/N: {stats\_por\_quadrante['N/N']['media']:.1f}%")  
print(f" Diferença: {diff\_s\_s\_n\_n:+.1f} pontos percentuais")  
  
# Quadrante com maior % de qualificação  
max\_qual\_q = max(stats\_por\_quadrante.items(), key=lambda x: x[1]['media'])[0]  
print(f"\n2. QUADRANTE COM MAIOR % DE QUALIFICAÇÃO: {max\_qual\_q}")  
  
# Correlação entre % qualificação e aprovação  
corr\_ministerio = df\_analise['perc\_qualificacao'].corr(df\_analise['ministerio\_aprova'])  
corr\_do = df\_analise['perc\_qualificacao'].corr(df\_analise['pesquisador\_aprova'])  
  
print(f"\n3. CORRELAÇÃO COM APROVAÇÃO:")  
print(f" Ministério: {corr\_ministerio:.3f}")  
print(f" DO: {corr\_do:.3f}")

=========================================================================  
ANÁLISE 4.1: PERCENTUAL DE QUALIFICAÇÃO (M+D) NO TOTAL DE RH  
=========================================================================  
  
ESTATÍSTICAS DESCRITIVAS  
------------------------------------------------------------  
Percentual médio de qualificação: 42.0%  
Mediana: 27.3%  
Desvio padrão: 43.4%  
Mínimo: 0.2%  
Máximo: 200.0%  
  
============================================================  
PERCENTUAL DE QUALIFICAÇÃO POR QUADRANTE  
------------------------------------------------------------  
  
S/S:  
 Média: 44.7%  
 Mediana: 29.4%  
 Desvio padrão: 43.7%  
 % projetos com >50% qualificados: 5.8%  
  
S/N:  
 Média: 29.8%  
 Mediana: 20.0%  
 Desvio padrão: 35.2%  
 % projetos com >50% qualificados: 1.9%  
  
N/S:  
 Média: 40.0%  
 Mediana: 23.3%  
 Desvio padrão: 43.0%  
 % projetos com >50% qualificados: 2.3%  
  
N/N:  
 Média: 36.0%  
 Mediana: 17.6%  
 Desvio padrão: 43.3%  
 % projetos com >50% qualificados: 2.2%  
  
============================================================  
IMPORTÂNCIA DO PERCENTUAL DE QUALIFICAÇÃO  
------------------------------------------------------------  
  
Regressão Logística - Impacto na probabilidade de cada quadrante:  
  
S/S:  
 Coeficiente: 0.331  
 Odds Ratio: 1.392  
 → Alto % qualificação AUMENTA chance de S/S  
  
S/N:  
 Coeficiente: -0.241  
 Odds Ratio: 0.786  
 → Alto % qualificação DIMINUI chance de S/N  
  
N/S:  
 Coeficiente: -0.190  
 Odds Ratio: 0.827  
 → Alto % qualificação DIMINUI chance de N/S  
  
N/N:  
 Coeficiente: -0.292  
 Odds Ratio: 0.747  
 → Alto % qualificação DIMINUI chance de N/N  
  
============================================================  
DISTRIBUIÇÃO POR FAIXAS DE QUALIFICAÇÃO  
------------------------------------------------------------  
  
Baixa (<10%) (n=2,215):  
 S/S: 54.9%  
 S/N: 8.7%  
 N/S: 5.0%  
 N/N: 31.4%  
  
Média (10-25%) (n=2,961):  
 S/S: 71.2%  
 S/N: 7.0%  
 N/S: 7.1%  
 N/N: 14.8%  
  
Alta (25-50%) (n=2,803):  
 S/S: 79.6%  
 S/N: 3.4%  
 N/S: 4.4%  
 N/N: 12.7%  
  
Muito Alta (>50%) (n=1,956):  
 S/S: 75.5%  
 S/N: 3.8%  
 N/S: 4.8%  
 N/N: 16.0%





================================================================================  
PRINCIPAIS INSIGHTS  
------------------------------------------------------------  
  
1. DIFERENÇA S/S vs N/N:  
 Média S/S: 44.7%  
 Média N/N: 36.0%  
 Diferença: +8.7 pontos percentuais  
  
2. QUADRANTE COM MAIOR % DE QUALIFICAÇÃO: S/S  
  
3. CORRELAÇÃO COM APROVAÇÃO:  
 Ministério: 0.096  
 DO: 0.065

## 7. Síntese dos Resultados

print(f"\n{'='\*80}")  
print("SÍNTESE DOS RESULTADOS")  
print(f"{'='\*80}")  
  
print("\n1. VALOR DO PROJETO E QUADRANTES:")  
print("-"\*40)  
for q, res in resultados\_regressao.items():  
 if res['p\_value'] < 0.05:  
 direção = "aumenta" if res['coeficiente'] > 0 else "diminui"  
 print(f" • Maior valor {direção} chance de {q} (OR={res['odds\_ratio']:.2f}, p<0.05)")  
 else:  
 print(f" • Valor não afeta {q} significativamente")  
  
print("\n2. IMPACTO DE ICT:")  
print("-"\*40)  
print(f" • Projetos com ICT: {pct\_com\_ict:.1f}% do total")  
print(f" • Maior impacto em Q1 (ambos aprovam)")  
print(f" • ICT aumenta taxa de aprovação ministerial")  
  
print("\n3. DOUTORES E MESTRES:")  
print("-"\*40)  
print(f" • Doutores têm impacto positivo na aprovação")  
print(f" • Mestres têm efeito similar mas menos pronunciado")  
print(f" • Efeito é maior na aprovação ministerial que no DO")  
  
print("\n4. GASTOS POR SETOR:")  
print("-"\*40)  
print(f" • Padrões variam significativamente entre setores")  
print(f" • % RH geralmente tem efeito positivo")  
print(f" • % Terceiros tem efeito misto dependendo do setor")  
  
print(f"\n{'='\*80}")  
print("ANÁLISE CONCLUÍDA")  
print(f"{'='\*80}")

================================================================================  
SÍNTESE DOS RESULTADOS  
================================================================================  
  
1. VALOR DO PROJETO E QUADRANTES:  
----------------------------------------  
 • Maior valor aumenta chance de Q1 (OR=1.02, p<0.05)  
 • Maior valor aumenta chance de Q2 (OR=1.02, p<0.05)  
 • Maior valor diminui chance de Q3 (OR=0.82, p<0.05)  
 • Valor não afeta Q4 significativamente  
  
2. IMPACTO DE ICT:  
----------------------------------------  
 • Projetos com ICT: 11.0% do total  
 • Maior impacto em Q1 (ambos aprovam)  
 • ICT aumenta taxa de aprovação ministerial  
  
3. DOUTORES E MESTRES:  
----------------------------------------  
 • Doutores têm impacto positivo na aprovação  
 • Mestres têm efeito similar mas menos pronunciado  
 • Efeito é maior na aprovação ministerial que no DO  
  
4. GASTOS POR SETOR:  
----------------------------------------  
 • Padrões variam significativamente entre setores  
 • % RH geralmente tem efeito positivo  
 • % Terceiros tem efeito misto dependendo do setor  
  
================================================================================  
ANÁLISE CONCLUÍDA  
================================================================================