INSIGHT 2 — The SKU Complexity ROI

Non-Standard Color Demand Analysis

```
# ■■ BLOCK 2: SKU COMPLEXITY ROI ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

core_colors = ['Silver', 'Space Gray', 'Black']
nonstandard_colors = ['Red', 'Gold', 'Blue', 'White']

df['color_tier'] = df['color'].apply(lambda c: 'Core' if c in core_colors else 'Non-Standard')

# Category × Color Tier aggregation
cat_color = df.groupby(['category', 'color_tier']).agg(
    avg_reviews=('review_count', 'mean'),
    median_reviews=('review_count', 'median'),
    total_reviews=('review_count', 'sum'),
    sku_count=('product_id', 'count'),
    avg_price=('price', 'mean')
).reset_index()

# Mann-Whitney U test per category
test_results = []
for cat in sorted(df['category'].unique()):
    core = df[(df['category'] == cat) & (df['color_tier'] == 'Core')]['review_count'].values
    nonst = df[(df['category'] == cat) & (df['color_tier'] == 'Non-Standard')]['review_count'].values
    u_stat, p_val = mannwhitneyu(core, nonst)
    test_results.append({
        'category': cat,
        'core_median': np.median(core),
        'nonstandard_median': np.median(nonst),
        'demand_lift': ((np.median(nonst) - np.median(core)) / np.median(core) * 100) if np.median(core)
            > 0 else 0,
        'U_statistic': u_stat,
        'p_value': p_val,
        'significant': p_val < 0.05
    })
test_df = pd.DataFrame(test_results)

# ■■ VISUALIZATION ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
fig, ax = plt.subplots(figsize=(12, 6))

pivot = cat_color.pivot(index='category', columns='color_tier', values='avg_reviews')
categories = pivot.index.tolist()
x = np.arange(len(categories))
width = 0.35

bars_core = ax.bar(x - width/2, pivot['Core'], width,
                   label='Core Colors (Silver, Space Gray, Black)',
                   color=PALETTE[0], edgecolor='white', linewidth=0.8)
bars_ns = ax.bar(x + width/2, pivot['Non-Standard'], width,
                 label='Non-Standard (Red, Gold, Blue, White)',
                 color=PALETTE[1], edgecolor='white', linewidth=0.8)

# Add significance stars
for i, row in test_df.iterrows():
    idx = categories.index(row['category'])
    max_val = max(pivot.loc[row['category']])
    if row['significant']:
        ax.text(idx, max_val + 30, '★', ha='center', fontsize=14, color=PALETTE[3])
    else:
        ax.text(idx, max_val + 30, 'n.s.', ha='center', fontsize=8, color='gray')

ax.set_xticks(x)
ax.set_xticklabels(categories, rotation=25, ha='right')
ax.set_ylabel('Average Review Count (Market Demand Proxy)')
ax.set_title('Does Color Complexity Drive Demand?\nAvg Review Count by Color Tier × Category',
             fontweight='bold')
ax.legend(loc='upper left', framealpha=0.9)

# Add value labels
for bar in list(bars_core) + list(bars_ns):
    height = bar.get_height()
    ax.text(bar.get_x() + bar.get_width()/2., height + 5,
```

INSIGHT 2 — The SKU Complexity ROI

Non-Standard Color Demand Analysis

```
            f'{height:,.0f}', ha='center', va='bottom', fontsize=8)

plt.tight_layout()
plt.savefig('block2_color_complexity.png', dpi=150, bbox_inches='tight')
plt.show()

# ■■ BLOCK 2: SUMMARY TABLE FOR EXPORT ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

print("■ TABLE 2: Color Complexity ROI by Category")
print("="*100)
export_color = test_df.copy()
export_color['demand_lift'] = export_color['demand_lift'].map('{:+.1f}%'.format)
export_color['p_value'] = export_color['p_value'].map('{:.6f}'.format)
export_color['verdict'] = export_color['significant'].map(
    {True: '✓ Color drives demand', False: '✗ No significant lift'}
)
export_color = export_color[['category', 'core_median', 'nonstandard_median',
                             'demand_lift', 'p_value', 'verdict']]
export_color.columns = ['Category', 'Core Median Reviews', 'Non-Std Median Reviews',
                        'Demand Lift', 'p-value', 'Verdict']
print(export_color.to_string(index=False))

# Global test
core_all = df[df['color_tier'] == 'Core']['review_count'].values
nonst_all = df[df['color_tier'] == 'Non-Standard']['review_count'].values
u_global, p_global = mannwhitneyu(core_all, nonst_all)
print(f"\nGLOBAL TEST: Mann-Whitney U = {u_global:,.0f}, p = {p_global:.6f}")
print(f"Conclusion: {'Color complexity HAS a measurable demand effect overall' if p_global < 0.05 else 'N
    o measurable global effect — category-specific analysis is what matters'}")
```

Block 2 Complete | 98 lines of executable Python | Statistical tests + Visualization + Export table | 2 page(s)