INSIGHT 5 — The "Premium Remorse" Quadrant

High Price, Low Satisfaction Segmentation

```python
# ■■ BLOCK 5: PREMIUM REMORSE QUADRANT ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

price_75 = df['price'].quantile(0.75)
rating_25 = df['rating'].quantile(0.25)

# Assign quadrants
def assign_quadrant(row):
    if row['price'] >= price_75 and row['rating'] <= rating_25:
        return 'Premium Remorse'
    elif row['price'] >= price_75 and row['rating'] > rating_25:
        return 'Premium Justified'
    elif row['price'] < price_75 and row['rating'] <= rating_25:
        return 'Budget Disappointment'
    else:
        return 'Value Sweet Spot'

df['quadrant'] = df.apply(assign_quadrant, axis=1)

# Premium Remorse deep dive
remorse = df[df['quadrant'] == 'Premium Remorse']
total_lost_volume = remorse['review_count'].sum()
total_market_volume = df['review_count'].sum()

# Biggest offenders by category
remorse_by_cat = remorse.groupby('category').agg(
    sku_count=('product_id', 'count'),
    total_reviews=('review_count', 'sum'),
    avg_price=('price', 'mean'),
    avg_rating=('rating', 'mean')
).reset_index().sort_values('total_reviews', ascending=False)

# Biggest offenders by model
remorse_by_model = remorse.groupby('model_name').agg(
    sku_count=('product_id', 'count'),
    total_reviews=('review_count', 'sum'),
    avg_price=('price', 'mean'),
    avg_rating=('rating', 'mean'),
    category=('category', 'first')
).reset_index().sort_values('total_reviews', ascending=False).head(10)

# ■■ VISUALIZATION: 4-Quadrant Scatter ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
fig, ax = plt.subplots(figsize=(13, 8))

quad_colors = {
    'Premium Remorse': PALETTE[3],
    'Premium Justified': PALETTE[0],
    'Budget Disappointment': PALETTE[2],
    'Value Sweet Spot': PALETTE[5]
}

for quad, color in quad_colors.items():
    mask = df['quadrant'] == quad
    subset = df[mask]
    ax.scatter(subset['rating'], subset['price'],
               s=subset['review_count'] / 50, alpha=0.25,
               c=color, label=f"{quad} ({mask.sum():,} SKUs)", edgecolors='none')

# Quadrant lines
ax.axhline(price_75, color='black', linestyle='--', alpha=0.4, linewidth=1)
ax.axvline(rating_25, color='black', linestyle='--', alpha=0.4, linewidth=1)

# Quadrant labels
ax.text(1.1, df['price'].max() * 0.98, 'PREMIUM\nREMORSE', fontsize=11,
        fontweight='bold', color=PALETTE[3], ha='left', va='top',
        bbox=dict(boxstyle='round,pad=0.3', facecolor='#FFEEEE', alpha=0.8))
ax.text(4.9, df['price'].max() * 0.98, 'PREMIUM\nJUSTIFIED', fontsize=11,
        fontweight='bold', color=PALETTE[0], ha='right', va='top',
        bbox=dict(boxstyle='round,pad=0.3', facecolor='#EEF5FF', alpha=0.8))
ax.text(1.1, df['price'].min() * 1.05, 'BUDGET\nDISAPPOINTMENT', fontsize=9,
```

INSIGHT 5 — The "Premium Remorse" Quadrant

High Price, Low Satisfaction Segmentation

```
        color=PALETTE[2], ha='left', va='bottom',
        bbox=dict(boxstyle='round,pad=0.3', facecolor='#FFF5EE', alpha=0.8))
ax.text(4.9, df['price'].min() * 1.05, 'VALUE\nSWEET SPOT', fontsize=9,
        color=PALETTE[5], ha='right', va='bottom',
        bbox=dict(boxstyle='round,pad=0.3', facecolor='#EEFFEE', alpha=0.8))

ax.set_xlabel('Customer Rating', fontsize=12)
ax.set_ylabel('Price ($)', fontsize=12)
ax.set_title(f'The "Premium Remorse" Quadrant Map\n'
             f'Bubble size = Review Count  |  {len(remorse):,} SKUs in Premium Remorse = '
             f'{total_lost_volume:,} lost-confidence reviews ({total_lost_volume/total_market_volume:.1%}
                  of market)',
             fontweight='bold', fontsize=12)
ax.legend(loc='center left', fontsize=9, framealpha=0.9,
          bbox_to_anchor=(0.01, 0.5))

plt.tight_layout()
plt.savefig('block5_premium_remorse.png', dpi=150, bbox_inches='tight')
plt.show()

print(f"\n{'='*70}")
print(f"PREMIUM REMORSE SEGMENT SUMMARY")
print(f"  Total SKUs in Premium Remorse:         {len(remorse):,}")
print(f"  Total review volume (lost confidence):  {total_lost_volume:,}")
print(f"  Share of total market volume:           {total_lost_volume/total_market_volume:.1%}")
print(f"  Avg Price in segment:                  ${remorse['price'].mean():,.2f}")
print(f"  Avg Rating in segment:                 {remorse['rating'].mean():.2f}")
print(f"{'='*70}")

# ■■ BLOCK 5: SUMMARY TABLES FOR EXPORT ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

print("■ TABLE 5A: Premium Remorse by Category")
print("="*80)
export_cat = remorse_by_cat.copy()
export_cat['avg_price'] = export_cat['avg_price'].map('${:,.2f}'.format)
export_cat['avg_rating'] = export_cat['avg_rating'].round(2)
export_cat['pct_of_remorse_volume'] = (remorse_by_cat['total_reviews'] / total_lost_volume * 100).round(1
    )
export_cat.columns = ['Category', 'SKU Count', 'Total Reviews', 'Avg Price', 'Avg Rating', '% of Remorse
    Volume']
print(export_cat.to_string(index=False))

print("\n\n■ TABLE 5B: Top 10 Model Names in Premium Remorse (Biggest Offenders)")
print("="*90)
export_model = remorse_by_model.copy()
export_model['avg_price'] = export_model['avg_price'].map('${:,.2f}'.format)
export_model['avg_rating'] = export_model['avg_rating'].round(2)
export_model.columns = ['Model', 'SKU Count', 'Total Reviews', 'Avg Price', 'Avg Rating', 'Category']
print(export_model.to_string(index=False))

print("\n\n■ TABLE 5C: Quadrant Distribution Summary")
print("="*70)
quad_summary = df.groupby('quadrant').agg(
    sku_count=('product_id', 'count'),
    total_reviews=('review_count', 'sum'),
    avg_price=('price', 'mean'),
    avg_rating=('rating', 'mean')
).round(2).reset_index()
quad_summary['pct_volume'] = (quad_summary['total_reviews'] / total_market_volume * 100).round(1)
quad_summary.columns = ['Quadrant', 'SKUs', 'Total Reviews', 'Avg Price', 'Avg Rating', '% Market Volume'
    ]
print(quad_summary.to_string(index=False))
```

Block 5 Complete | 131 lines of executable Python | Statistical tests + Visualization + Export table | 2 page(s)