INSIGHT 1 — The Stock-Out Penalty

Supply Chain Failures & Customer Sentiment

```python
# ■■ BLOCK 1: STOCK-OUT PENALTY ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

# 1A. Worst sellers by stock-out rate (min 20 products to avoid noise)
seller_stats = df.groupby('seller_id').agg(
    total_products=('in_stock', 'count'),
    out_of_stock=('in_stock', lambda x: (~x).sum()),
    avg_rating=('rating', 'mean')
).reset_index()
seller_stats['stockout_rate'] = seller_stats['out_of_stock'] / seller_stats['total_products']
seller_stats = seller_stats[seller_stats['total_products'] >= 20]
top_sellers = seller_stats.nlargest(10, 'stockout_rate')

# 1B. Country-level stock-out rates and rating splits
country_list = []
for country in df['country_origin'].unique():
    cdf = df[df['country_origin'] == country]
    country_list.append({
        'country_origin': country,
        'total': len(cdf),
        'out_of_stock': (~cdf['in_stock']).sum(),
        'avg_rating_in': cdf[cdf['in_stock'] == True]['rating'].mean(),
        'avg_rating_oos': cdf[cdf['in_stock'] == False]['rating'].mean()
    })
country_stats = pd.DataFrame(country_list)
country_stats['stockout_rate'] = country_stats['out_of_stock'] / country_stats['total']

# 1C. T-test: rating of in-stock vs out-of-stock products
in_stock_ratings = df[df['in_stock'] == True]['rating'].values
oos_ratings = df[df['in_stock'] == False]['rating'].values
t_stat, p_value = ttest_ind(in_stock_ratings, oos_ratings)

# ■■ VISUALIZATION ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
fig, axes = plt.subplots(1, 2, figsize=(14, 5.5), gridspec_kw={'width_ratios': [1.3, 1]})

# Left: Top 10 worst sellers
ax1 = axes[0]
bars = ax1.barh(
    top_sellers['seller_id'], top_sellers['stockout_rate'] * 100,
    color=PALETTE[3], edgecolor='white', linewidth=0.5
)
ax1.set_xlabel('Stock-Out Rate (%)')
ax1.set_title('Top 10 Sellers by Stock-Out Rate\n(min 20 products)', fontweight='bold')
ax1.invert_yaxis()
for bar, rate in zip(bars, top_sellers['stockout_rate']):
    ax1.text(bar.get_width() + 0.5, bar.get_y() + bar.get_height()/2,
             f'{rate:.1%}', va='center', fontsize=8, color=PALETTE[3], fontweight='bold')
ax1.axvline(50, color='gray', linestyle='--', alpha=0.5, label='50% threshold')
ax1.legend(loc='lower right', fontsize=8)

# Right: Country stock-out rate + rating gap
ax2 = axes[1]
countries_sorted = country_stats.sort_values('stockout_rate', ascending=False)
x = np.arange(len(countries_sorted))
width = 0.35
ax2.bar(x - width/2, countries_sorted['avg_rating_in'], width,
        label='In-Stock Avg Rating', color=PALETTE[0], edgecolor='white')
ax2.bar(x + width/2, countries_sorted['avg_rating_oos'], width,
        label='Out-of-Stock Avg Rating', color=PALETTE[3], edgecolor='white')
ax2.set_xticks(x)
ax2.set_xticklabels(countries_sorted['country_origin'], rotation=30, ha='right')
ax2.set_ylabel('Average Rating')
ax2.set_title('Rating Gap: In-Stock vs Out-of-Stock\nby Manufacturing Country', fontweight='bold')
ax2.legend(fontsize=8)

# Add p-value annotation
sig_text = f"Welch's T-test: t={t_stat:.3f}, p={p_value:.4f}"
if p_value < 0.05:
    sig_text += " ★ SIGNIFICANT"
fig.text(0.5, -0.02, sig_text, ha='center', fontsize=10,
```

INSIGHT 1 — The Stock-Out Penalty

Supply Chain Failures & Customer Sentiment

```
         style='italic', color=PALETTE[3] if p_value < 0.05 else 'gray')

plt.tight_layout()
plt.savefig('block1_stockout_penalty.png', dpi=150, bbox_inches='tight')
plt.show()
print(f"\n{'='*70}")
print(f"STATISTICAL TEST: Welch's T-test (In-Stock vs Out-of-Stock Ratings)")
print(f"  In-Stock mean rating:     {np.mean(in_stock_ratings):.4f} (n={len(in_stock_ratings):,})")
print(f"  Out-of-Stock mean rating: {np.mean(oos_ratings):.4f} (n={len(oos_ratings):,})")
print(f"  t-statistic: {t_stat:.4f}")
print(f"  p-value:     {p_value:.6f}")
print(f"  Conclusion:  {'Rating penalty is STATISTICALLY SIGNIFICANT' if p_value < 0.05 else 'No statisti
    cally significant difference detected'}")
print(f"{'='*70}")

# ■■ BLOCK 1: SUMMARY TABLE FOR EXPORT ■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

# Seller-level summary
print("\n■ TABLE 1A: Top 10 Sellers by Stock-Out Rate")
print("="*80)
export_sellers = top_sellers[['seller_id', 'total_products', 'out_of_stock', 'stockout_rate', 'avg_rating
    ']].copy()
export_sellers['stockout_rate'] = export_sellers['stockout_rate'].map('{:.1%}'.format)
export_sellers['avg_rating'] = export_sellers['avg_rating'].round(2)
export_sellers.columns = ['Seller ID', 'Total Products', 'Out of Stock', 'Stock-Out Rate', 'Avg Rating']
print(export_sellers.to_string(index=False))

# Country-level summary
print("\n\n■ TABLE 1B: Stock-Out Rate by Manufacturing Country")
print("="*80)
export_country = country_stats[['country_origin', 'total', 'stockout_rate', 'avg_rating_in', 'avg_rating_
    oos']].copy()
export_country['stockout_rate'] = export_country['stockout_rate'].map('{:.1%}'.format)
export_country['avg_rating_in'] = export_country['avg_rating_in'].round(3)
export_country['avg_rating_oos'] = export_country['avg_rating_oos'].round(3)
export_country['rating_gap'] = (country_stats['avg_rating_in'] - country_stats['avg_rating_oos']).round(3
    )
export_country.columns = ['Country', 'Total SKUs', 'Stock-Out Rate', 'Avg Rating (In)', 'Avg Rating (OOS)
    ', 'Rating Gap']
print(export_country.to_string(index=False))
```

Block 1 Complete | 109 lines of executable Python | Statistical tests + Visualization + Export table | 2 page(s)