In [19]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
import statsmodels.api as sm
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_squared_error

np.random.seed(42)

n = 36
revenues = np.random.uniform(10, 500, n)

beta_0 = 8000
beta_1 = 180
noise = np.random.normal(0, 8000, n)
gaviscon_income = beta_0 + beta_1 * revenues + noise
gaviscon_income = np.clip(gaviscon_income, 0, None)

df = pd.DataFrame({
    'Month': range(1, n + 1),
    'Revenue (USD Millions)': np.round(revenues, 2),
    'Gaviscon Supplier Income (USD)': np.round(gaviscon_income, 2)
})

print('Dataset preview (first 10 months):')
print(df.head(10).to_string(index=False))
print(f'\nDataset shape: {df.shape}')
print(f'Revenue range: ${df["Revenue (USD Millions)"].min():.1f}M - ${df["Revenue (USD Millio
print(f'Gaviscon income range: ${df["Gaviscon Supplier Income (USD)"].min():,.0f} - ${df["Gav
```

```
Dataset preview (first 10 months):
 Month  Revenue (USD Millions)  Gaviscon Supplier Income (USD)
     1                  193.52                        38029.33
     2                  475.85                        91319.45
     3                  368.68                        69548.21
     4                  303.34                        77419.90
     5                   86.45                        23452.87
     6                   86.44                        15097.03
     7                   38.46                        21503.33
     8                  434.43                        76429.99
     9                  304.55                        64489.25
    10                  356.96                        56574.64

Dataset shape: (36, 3)
Revenue range: $20.1M - $485.3M
Gaviscon income range: $990 - $96,921
```

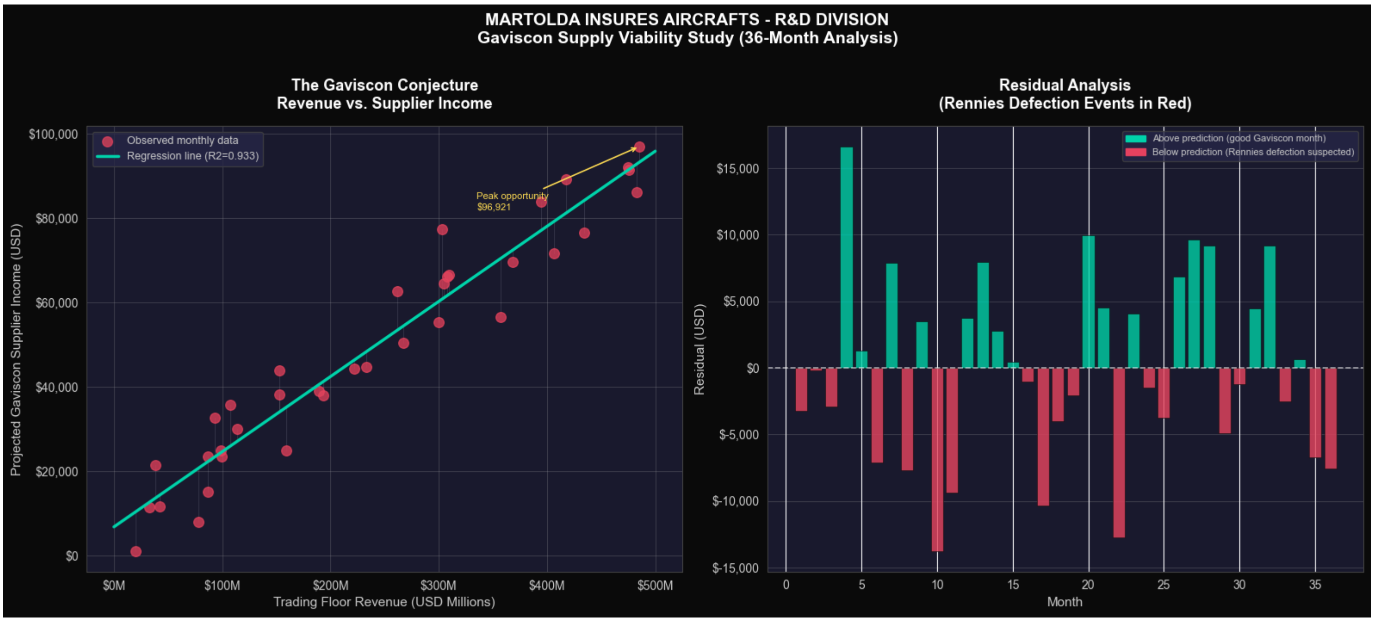
In [20]:

```
X = df[['Revenue (USD Millions)']].values
y = df['Gaviscon Supplier Income (USD)'].values

model = LinearRegression()
model.fit(X, y)
y_pred = model.predict(X)
residuals = y - y_pred

print(f'Model trained successfully.')
print(f'Intercept: ${model.intercept_:,.2f}')
print(f'Coefficient: ${model.coef_[0]:,.2f} per $1M revenue')
```

```
Model trained successfully.
Intercept: $6,799.10
Coefficient: $178.04 per $1M revenue
```

In [21]:
```python
fig, axes = plt.subplots(1, 2, figsize=(16, 7))
fig.patch.set_facecolor('#0d0d0d')

for ax in axes:
    ax.set_facecolor('#1a1a2e')
    ax.tick_params(colors='#cccccc')
    ax.xaxis.label.set_color('#cccccc')
    ax.yaxis.label.set_color('#cccccc')
    ax.title.set_color('#ffffff')
    for spine in ax.spines.values():
        spine.set_edgecolor('#444444')

ax1 = axes[0]
ax1.scatter(X, y, color='#e94560', alpha=0.75, s=70, zorder=3, label='Observed monthly data')

x_line = np.linspace(0, 500, 300).reshape(-1, 1)
y_line = model.predict(x_line)
ax1.plot(x_line, y_line, color='#00d4aa', linewidth=2.5,
         label=f'Regression line (R2={r2_score(y, y_pred):.3f})', zorder=4)

for xi, yi, yp in zip(X.flatten(), y, y_pred):
    ax1.plot([xi, xi], [yi, yp], color='#ffffff', alpha=0.12, linewidth=0.8, zorder=2)

ax1.set_xlabel('Trading Floor Revenue (USD Millions)', fontsize=11)
ax1.set_ylabel('Projected Gaviscon Supplier Income (USD)', fontsize=11)
ax1.set_title('The Gaviscon Conjecture\nRevenue vs. Supplier Income', fontsize=13, fontweight
ax1.legend(facecolor='#2a2a4a', edgecolor='#444444', labelcolor='#cccccc', fontsize=9)
ax1.yaxis.set_major_formatter(plt.FuncFormatter(lambda x, _: f'${x:,.0f}'))
ax1.xaxis.set_major_formatter(plt.FuncFormatter(lambda x, _: f'${x:.0f}M'))
ax1.grid(True, alpha=0.15, color='#ffffff')

max_idx = np.argmax(y)
ax1.annotate(
    f'Peak opportunity\n${y[max_idx]:,.0f}',
    xy=(X[max_idx][0], y[max_idx]),
    xytext=(X[max_idx][0] - 150, y[max_idx] - 15000),
    color='#ffdd57', fontsize=8,
    arrowprops=dict(arrowstyle='->', color='#ffdd57', lw=1.2)
)

ax2 = axes[1]
colors = ['#e94560' if r < 0 else '#00d4aa' for r in residuals]
ax2.bar(range(1, n + 1), residuals, color=colors, alpha=0.8, edgecolor='#0d0d0d', linewidth=0
ax2.axhline(0, color='#ffffff', linewidth=1.2, linestyle='--', alpha=0.6)
ax2.set_xlabel('Month', fontsize=11)
ax2.set_ylabel('Residual (USD)', fontsize=11)
ax2.set_title('Residual Analysis\n(Rennies Defection Events in Red)', fontsize=13, fontweight
ax2.yaxis.set_major_formatter(plt.FuncFormatter(lambda x, _: f'${x:,.0f}'))
ax2.grid(True, alpha=0.15, color='#ffffff', axis='y')

green_patch = mpatches.Patch(color='#00d4aa', label='Above prediction (good Gaviscon month)')
red_patch = mpatches.Patch(color='#e94560', label='Below prediction (Rennies defection suspec
ax2.legend(handles=[green_patch, red_patch], facecolor='#2a2a4a', edgecolor='#444444',
           labelcolor='#cccccc', fontsize=8, loc='upper right')

fig.suptitle(
    'MARTOLDA INSURES AIRCRAFTS - R&D DIVISION\nGaviscon Supply Viability Study (36-Month Ana
    fontsize=14, fontweight='bold', color='#ffffff', y=1.01
)
```

```python
plt.tight_layout()
plt.savefig('gaviscon_regression.png', dpi=150, bbox_inches='tight', facecolor='#0d0d0d')
plt.show()
```



In [22]:
```python
milestones = [50, 100, 200, 300, 400, 500]

print('=' * 60)
print('   PROJECTED GAVISCON INCOME AT KEY REVENUE MILESTONES')
print('=' * 60)
print(f'  {"Revenue":>15}  {"Projected Monthly":>20}  {"Projected Annual":>17}')
print('-' * 60)
for rev in milestones:
    monthly = model.predict([[rev]])[0]
    annual = monthly * 12
    print(f'  ${rev:>12}M  ${monthly:>19,.2f}  ${annual:>16,.2f}')
print('=' * 60)
print()
print('Note: Annual figures assume sustained revenue levels.')
print('Results do not account for bulk discount negotiations,')
print('competitor antacid brands, or sudden onset of mindfulness')
print('practices among trading floor personnel.')
```

```
============================================================
   PROJECTED GAVISCON INCOME AT KEY REVENUE MILESTONES
============================================================
          Revenue     Projected Monthly   Projected Annual
------------------------------------------------------------
  $          50M  $          15,701.34  $       188,416.04
  $         100M  $          24,603.58  $       295,242.91
  $         200M  $          42,408.05  $       508,896.63
  $         300M  $          60,212.53  $       722,550.36
  $         400M  $          78,017.01  $       936,204.09
  $         500M  $          95,821.48  $     1,149,857.82
============================================================

Note: Annual figures assume sustained revenue levels.
Results do not account for bulk discount negotiations,
competitor antacid brands, or sudden onset of mindfulness
practices among trading floor personnel.
```

In [23]:
```python
X_sm = sm.add_constant(df['Revenue (USD Millions)'].values)
ols_model = sm.OLS(y, X_sm).fit()
print(ols_model.summary(
    title='OLS Regression Results - The Gaviscon Conjecture',
```

```
    yname='Gaviscon Supplier Income (USD)',
    xname=['Intercept (Structural Heartburn)', 'Revenue (USD Millions)']
))
```

```
                             OLS Regression Results - The Gaviscon Conjecture
==========================================================================================
Dep. Variable:     Gaviscon Supplier Income (USD)   R-squared:                       0.933
Model:                                        OLS   Adj. R-squared:                  0.931
Method:                             Least Squares   F-statistic:                     473.8
Date:                            Sun, 22 Feb 2026   Prob (F-statistic):           1.54e-21
Time:                                    11:51:59   Log-Likelihood:                -369.67
No. Observations:                              36   AIC:                             743.3
Df Residuals:                                  34   BIC:                             746.5
Df Model:                                       1
Covariance Type:                        nonrobust
================================================================================================
======
                                       coef    std err          t      P>|t|      [0.025
0.975]
------------------------------------------------------------------------------------------------
------
Intercept (Structural Heartburn)  6799.0984   2282.109      2.979      0.005    2161.296    1.
14e+04
Revenue (USD Millions)             178.0448      8.180     21.766      0.000     161.421     1
94.668
==============================================================================
Omnibus:                        0.201   Durbin-Watson:                   1.913
Prob(Omnibus):                  0.905   Jarque-Bera (JB):                0.399
Skew:                           0.098   Prob(JB):                        0.819
Kurtosis:                       2.523   Cond. No.                         533.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```