Video Game Sales

Project aims

The project has three core aims:

1. **To characterise the overall sales distribution** of video games and demonstrate the extent to which the market is dominated by a small number of high-performing titles.

2. **To analyse genre- and region-level patterns** using descriptive statistics, visualisation, hypothesis testing, and regression modelling.

3. **To examine indie and cosy-style games as distinct market segments**, highlighting how they differ from mainstream titles in terms of scale, variance, sustainability, and regional dynamics.

Data and methods

The analysis uses the `vgsales.csv` dataset, which contains information on video game titles, platforms, publishers, genres, release years, and regional sales figures (North America, Europe, Japan, Other, and Global).

The methodological approach includes:

- **Data cleaning and validation**, including type correction, duplicate removal, and regional consistency checks.
- **Descriptive statistics** to summarise sales distributions and category frequencies.
- **Visual exploration** (histograms, boxplots, scatter plots) to reveal market concentration, genre differences, and regional relationships.
- **Inferential analysis**, including:
  - a two-sample *t*-test comparing mean global sales between genre groups;
  - a multiple linear regression model predicting log-transformed global sales from regional sales and year, with robust standard errors.
- **Segment-specific analyses** for:
  - *cosy-style games*, operationalised using genre and title-based proxies;
  - *indie or indie-adjacent games*, proxied via publisher classification.

Key analytical themes

Several cross-cutting themes structure the analysis:

- **Market concentration and the “long tail”**: Most titles sell modestly, while a small minority dominate global revenue.
- **Genre as economic structure**: Genres shape not only gameplay but also typical sales ranges, variance, and risk profiles.
- **Regional power dynamics**: North America is a strong predictor of global success, but important deviations reflect region-specific appeal.
- **Indie and cosy sustainability**: Indie and cosy-style games often operate with lower average sales but contribute disproportionately to diversity, innovation, and long-tail value.
- **Limits of prediction**: Regression diagnostics highlight the difficulty of forecasting blockbuster success in creative industries.

Contribution and relevance

This project contributes an **applied, industry-relevant perspective** on video game sales by:

- demonstrating how standard statistical tools can be used responsibly in a highly skewed, creative-market context;
- showing why averages and single “success metrics” are insufficient for understanding indie and cosy game ecosystems;
- bridging quantitative analysis with qualitative insights relevant to developers, publishers, platform holders, and consultants.

This is created for the Everything Counts Module on Statistics at The London Interdisciplianry School Master's, 2025 for student 25000148737.

GitHub - Available at: https://github.com/tulin-b/everything-counts-assignment2-25000148737

http://localhost:8889/lab/tree/Desktop/code/everything-counts/assignment%202/everything_counts_assignment-2_25000148737.ipynb

1. Setup and data loading

In this section I have:

- Imported core Python libraries for data analysis and modelling.
- Loaded the `vgsales.csv` dataset.
- Taken a first look at the structure and the variables.

In [56]:

```
# 1. Imports and data loading

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from scipy import stats
```

```
import statsmodels.api as sm
import statsmodels.formula.api as smf

# Make plots appear inline in Jupyter
%matplotlib inline

# Optional: wider display for dataframes
pd.set_option("display.max_columns", 50)
pd.set_option("display.float_format", "{:.3f}".format)

# 1.1 Load the CSV (adjust path if needed)
data_path = "Desktop/code/everything-counts/assignment 2/vgsales.csv"  #
df = pd.read_csv(data_path)

print("Initial shape:", df.shape)
df.head()
```

```
Initial shape: (16598, 11)
```

Out[56]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sale
0	1	Wii Sports	Wii	2006.000	Sports	Nintendo	41.490	29.02
1	2	Super Mario Bros.	NES	1985.000	Platform	Nintendo	29.080	3.58
2	3	Mario Kart Wii	Wii	2008.000	Racing	Nintendo	15.850	12.88
3	4	Wii Sports Resort	Wii	2009.000	Sports	Nintendo	15.750	11.01
4	5	Pokemon Red/Pokemon Blue	GB	1996.000	Role-Playing	Nintendo	11.270	8.89

2. Data cleaning and preparation

Before analysis, I have created a clear and consistent dataset. The main cleaning choices are:

- **Column name normalisation:** to remove spaces and standardise names.
- **Type conversion:** to convert `Year` to numeric to enable temporal analysis.
- **Duplicate removal:** to drop exact duplicate rows (if any).
- **Missing values in sales columns:** treat missing sales as zero where appropriate.
- **Key identifiers:** to drop rows missing critical fields such as `Name` or `Genre`.
- **Global vs regional consistency check:** to help compare `Global_Sales` against the sum of regional sales (`NA_Sales`, `EU_Sales`, `JP_Sales`, `Other_Sales`).

These steps are documented so the cleaning process below is transparent and reproducible.

In [57]:

```
# 2. Cleaning and preparation

clean_notes = []

# 2.1 Normalise column names
df.columns = [c.strip().replace(" ", "_") for c in df.columns]
clean_notes.append("Normalised column names by stripping spaces and repla

# 2.2 Convert Year to numeric (if present)
if "Year" in df.columns:
    df["Year"] = pd.to_numeric(df["Year"], errors="coerce")
    n_missing_year = df["Year"].isna().sum()
    clean_notes.append(f"Converted Year to numeric; missing Year values: 
else:
    clean_notes.append("No 'Year' column found; skipping Year conversion.

# 2.3 Drop exact duplicates
before = len(df)
df = df.drop_duplicates()
after = len(df)
clean_notes.append(f"Dropped {before - after} exact duplicate rows.")

# 2.4 Fill missing sales values with 0 (for *_Sales columns)
sales_cols = [c for c in df.columns if c.endswith("_Sales")]
for col in sales_cols:
    if df[col].isna().any():
        n_missing = df[col].isna().sum()
        df[col] = df[col].fillna(0)
        clean_notes.append(f"Filled {n_missing} missing values in {col} w

# 2.5 Check consistency: Global_Sales vs sum of regionals
regional_cols = ["NA_Sales", "EU_Sales", "JP_Sales", "Other_Sales"]
if "Global_Sales" in df.columns and all(c in df.columns for c in regional
    df["Regional_Sum"] = df[regional_cols].sum(axis=1)
    df["Global_minus_Regional"] = df["Global_Sales"] - df["Regional_Sum"]
    inconsistent = (df["Global_minus_Regional"].abs() > 1e-6).sum()
    clean_notes.append(
        f"Computed Regional_Sum and Global_minus_Regional; "
        f"{inconsistent} rows where Global_Sales ≠ sum of regionals (beyo
    )
else:
    clean_notes.append("Missing Global_Sales or some regional columns; sk

# 2.6 Drop rows missing key identifiers if present
for key_col in ["Name", "Genre"]:
    if key_col in df.columns:
        n_missing = df[key_col].isna().sum()
        if n_missing > 0:
            df = df[df[key_col].notna()]
            clean_notes.append(f"Dropped {n_missing} rows with missing {k

clean_notes.append(f"Final dataset shape after cleaning: {df.shape}.")

# Display cleaning log
for note in clean_notes:
    print("-", note)

df.head()
```

```
- Normalised column names by stripping spaces and replacing spaces with un
derscores.
- Converted Year to numeric; missing Year values: 271.
- Dropped 0 exact duplicate rows.
- Computed Regional_Sum and Global_minus_Regional; 4511 rows where Global_
Sales ≠ sum of regionals (beyond tiny rounding error).
- Final dataset shape after cleaning: (16598, 13).
```

Out[57]:

	Rank	Name	Platform	Year	Genre	Publisher	NA_Sales	EU_Sale
0	1	Wii Sports	Wii	2006.000	Sports	Nintendo	41.490	29.02
1	2	Super Mario Bros.	NES	1985.000	Platform	Nintendo	29.080	3.58
2	3	Mario Kart Wii	Wii	2008.000	Racing	Nintendo	15.850	12.88
3	4	Wii Sports Resort	Wii	2009.000	Sports	Nintendo	15.750	11.01
4	5	Pokemon Red/Pokemon Blue	GB	1996.000	Role-Playing	Nintendo	11.270	8.89

3. Descriptive statistics

Understanding the basic structure:

- **Overall describe table** for all columns (numeric and categorical).
- **Focused numeric summary** for sales variables (mean, median, spread, skewness, etc.).
- **Frequency tables** for key categorical variables (`Platform` , `Genre` , `Publisher`).

This gives a first sense of market concentration, platform distribution, and genre mix before hypothesis testing and modelling.

In [58]:

```
# 3. Descriptive statistics

# 3.1 Overall summary (numeric + categorical)
desc_all = df.describe(include="all").transpose()
desc_all
```

Out[58]:

	count	unique	top	freq	mean	std
Rank	16598.000	NaN	NaN	NaN	8300.605	4791.854
Name	16598	11493	Need for Speed: Most Wanted	12	NaN	NaN
Platform	16598	31	DS	2163	NaN	NaN
Year	16327.000	NaN	NaN	NaN	2006.406	5.829
Genre	16598	12	Action	3316	NaN	NaN
Publisher	16540	578	Electronic Arts	1351	NaN	NaN
NA_Sales	16598.000	NaN	NaN	NaN	0.265	0.817
EU_Sales	16598.000	NaN	NaN	NaN	0.147	0.505
JP_Sales	16598.000	NaN	NaN	NaN	0.078	0.309
Other_Sales	16598.000	NaN	NaN	NaN	0.048	0.189
Global_Sales	16598.000	NaN	NaN	NaN	0.537	1.555
Regional_Sum	16598.000	NaN	NaN	NaN	0.537	1.555
Global_minus_Regional	16598.000	NaN	NaN	NaN	0.000	0.005

In [59]:

```python
# Targeted descriptive statistics for sales columns (robust)

# Identify numeric columns and focus on sales-related ones
numeric_cols = df.select_dtypes(include=[np.number]).columns.tolist()
sales_num_cols = [c for c in numeric_cols if "Sales" in c]

if not sales_num_cols:
    raise ValueError("No numeric sales columns found. Check your column n

# Helper to safely compute stats column-by-column
def safe_skew(x):
    x = pd.to_numeric(x, errors="coerce").dropna()
    return x.skew() if len(x) > 2 else np.nan

def safe_kurt(x):
    x = pd.to_numeric(x, errors="coerce").dropna()
    return x.kurtosis() if len(x) > 3 else np.nan

rows = []
for col in sales_num_cols:
    s = pd.to_numeric(df[col], errors="coerce")
    rows.append({
        "variable": col,
        "count": s.count(),
        "mean": s.mean(),
        "median": s.median(),
        "std": s.std(),
        "min": s.min(),
        "max": s.max(),
        "skew": safe_skew(s),
        "kurtosis": safe_kurt(s),
        "pct_zero": (s.fillna(0).eq(0).mean() * 100)
    })

desc_sales = pd.DataFrame(rows).set_index("variable").sort_index()
desc_sales
```

Out[59]:

	count	mean	median	std	min	max	skew	kurtosis	pct_z
variable									
EU_Sales	16598	0.147	0.020	0.505	0.000	29.020	18.876	756.028	34.
Global_Sales	16598	0.537	0.170	1.555	0.010	82.740	17.401	603.932	0.
JP_Sales	16598	0.078	0.000	0.309	0.000	10.220	11.206	194.234	62.
NA_Sales	16598	0.265	0.080	0.817	0.000	41.490	18.800	649.130	27
Other_Sales	16598	0.048	0.010	0.189	0.000	10.570	24.234	1025.348	39.

In [60]:

```
# Frequency tables for key categorical variables

for col in ["Platform", "Genre", "Publisher"]:
    if col in df.columns:
        print(f"\nTop 15 values for {col}:")
        display(df[col].value_counts().head(15))
```

```
Top 15 values for Platform:
Platform
DS      2163
PS2     2161
PS3     1329
Wii     1325
X360    1265
PSP     1213
PS      1196
PC       960
XB       824
GBA      822
GC       556
3DS      509
PSV      413
PS4      336
N64      319
Name: count, dtype: int64
Top 15 values for Genre:
```

```
Genre
Action          3316
Sports          2346
Misc            1739
Role-Playing    1488
Shooter         1310
Adventure       1286
Racing          1249
Platform         886
Simulation       867
Fighting         848
Strategy         681
Puzzle           582
Name: count, dtype: int64
Top 15 values for Publisher:
Publisher
Electronic Arts                           1351
Activision                                 975
Namco Bandai Games                         932
Ubisoft                                    921
Konami Digital Entertainment               832
THQ                                        715
Nintendo                                   703
Sony Computer Entertainment                683
Sega                                       639
Take-Two Interactive                       413
Capcom                                     381
Atari                                      363
Tecmo Koei                                 338
Square Enix                                233
Warner Bros. Interactive Entertainment     232
Name: count, dtype: int64
```

Interpretation: what do the descriptive statistics tell us?

- The **sales summaries** typically show:

  - A very **right-skewed** distribution (high skew, high kurtosis),
  - Many titles with low `Global_Sales`,
  - A small number of extremely high-selling titles pulling up the mean value.

  This pattern is typical of **creative and entertainment markets** where a few blockbusters capture most revenue, and the majority of products are represented by the “long tail”.

- The **platform and genre frequencies** usually indicate:

  - A concentration on dominant consoles of the period (e.g. PS2, DS, PS3, Xbox 360, Wii).
  - Genres such as **Action, Sports, Shooter** often appearing most frequently, reflecting mainstream tastes and large installed bases.
- For a **consulting lens**, this immediately suggests:

  - **High market concentration**: revenue is not evenly distributed.
  - **Risk asymmetry**: most titles underperform; only a few become breakout successes.

- A structural space where **indie / cosy / pixel-art titles** often live in the lower-sales tail, but can still succeed via niche audiences and low production costs.

4. Visual exploration

This section visualises key aspects of the sales landscape:

1. **Histogram of Global Sales** (Figure 1) — overall market concentration.
2. **Boxplot of Global Sales by Genre** (Figure 2) — genre performance patterns.
3. **Scatter plot of NA Sales vs Global Sales** (Figure 3) — regional power dynamics.
4. **Residuals vs fitted values** (Figure 4) — regression diagnostics.

Figure 1 – Histogram of Global Sales

This figure shows the distribution of `Global_Sales` across all titles.

We expect a **heavily right-skewed distribution**:

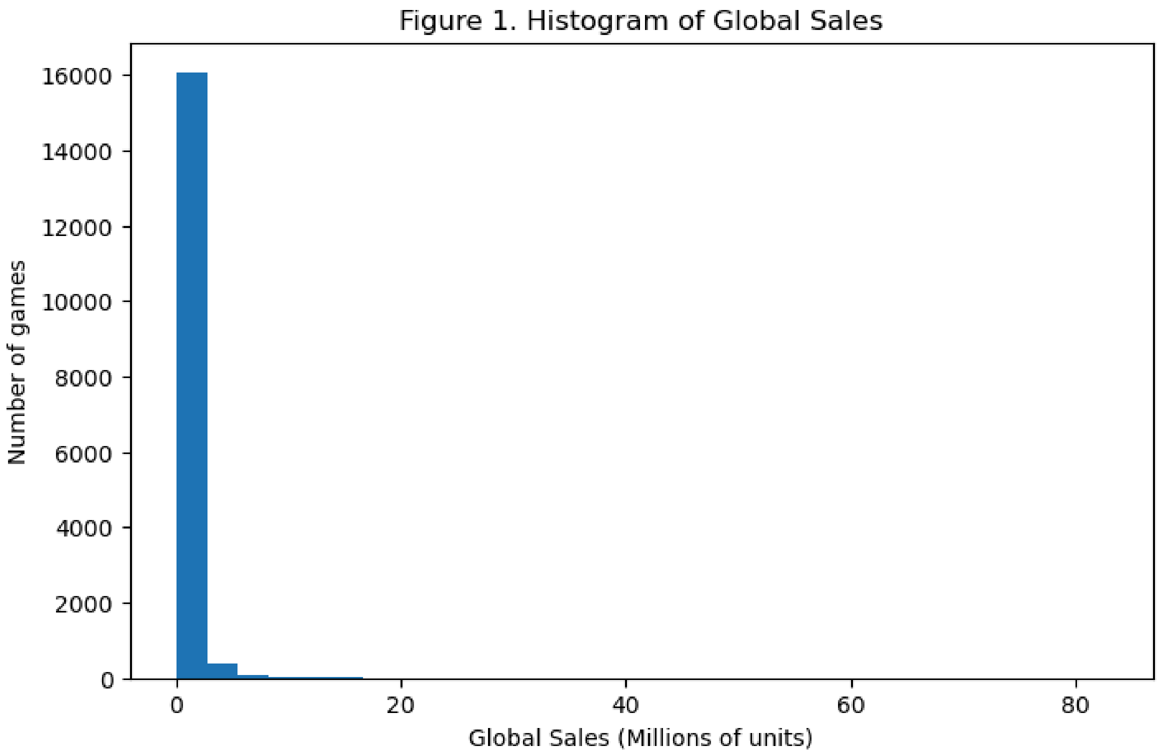
- Many games sell poorly (or modestly).
- A small minority achieve very high sales.

This pattern reflects:

- A **winner-takes-most** or **blockbuster** economy.
- The economic reality that most indie and niche titles live in the low-sales bulk, while AAA franchises dominate the extreme right tail.

In [61]:

```
# Histogram of Global Sales

plt.figure(figsize=(8, 5))
plt.hist(df["Global_Sales"], bins=30)
plt.title("Figure 1. Histogram of Global Sales")
plt.xlabel("Global Sales (Millions of units)")
plt.ylabel("Number of games")
plt.show()
```



4.2 Figure 2 – Boxplot of Global Sales by Genre

This boxplot compares the distribution of `Global_Sales` across genres.

Typical patterns:

- **Action, Sports, Shooter**:
  - Higher medians and wider interquartile ranges.
  - More extreme outliers (blockbuster titles).
  - Strong links with **high-intensity, reward-loop-based design**, often associated with addictive mechanics, multiplayer ecosystems, and esports.
- **Strategy, Puzzle, Adventure, Simulation** (including cosy / farming / life-sim styles):
  - Lower medians but often more stable distributions.
  - These genres can support **smaller studios and indie teams**, as:
    - Production costs are lower (especially with pixel/retro aesthetics).
    - Sales depend more on **community, streamers, and long-tail engagement** than aggressive marketing.

This figure supports an industry narrative where:

- Hyper-competitive, “addictive loop” genres dominate revenue.
- Cosy and pixel-art genres occupy a sustainable but niche economic space, often relying on loyal communities and word-of-mouth rather than scale.

In [62]:

```
# Boxplot of Global Sales by Genre

plt.figure(figsize=(12, 6))
df.boxplot(column="Global_Sales", by="Genre", showfliers=False, rot=45)
```

```
plt.title("Figure 2. Global Sales by Genre (outliers hidden)")
plt.suptitle("")  # removes pandas' automatic title
plt.xlabel("Genre")
plt.ylabel("Global Sales (Millions of units)")
plt.tight_layout()
plt.show()
```

```
<Figure size 1200x600 with 0 Axes>
```

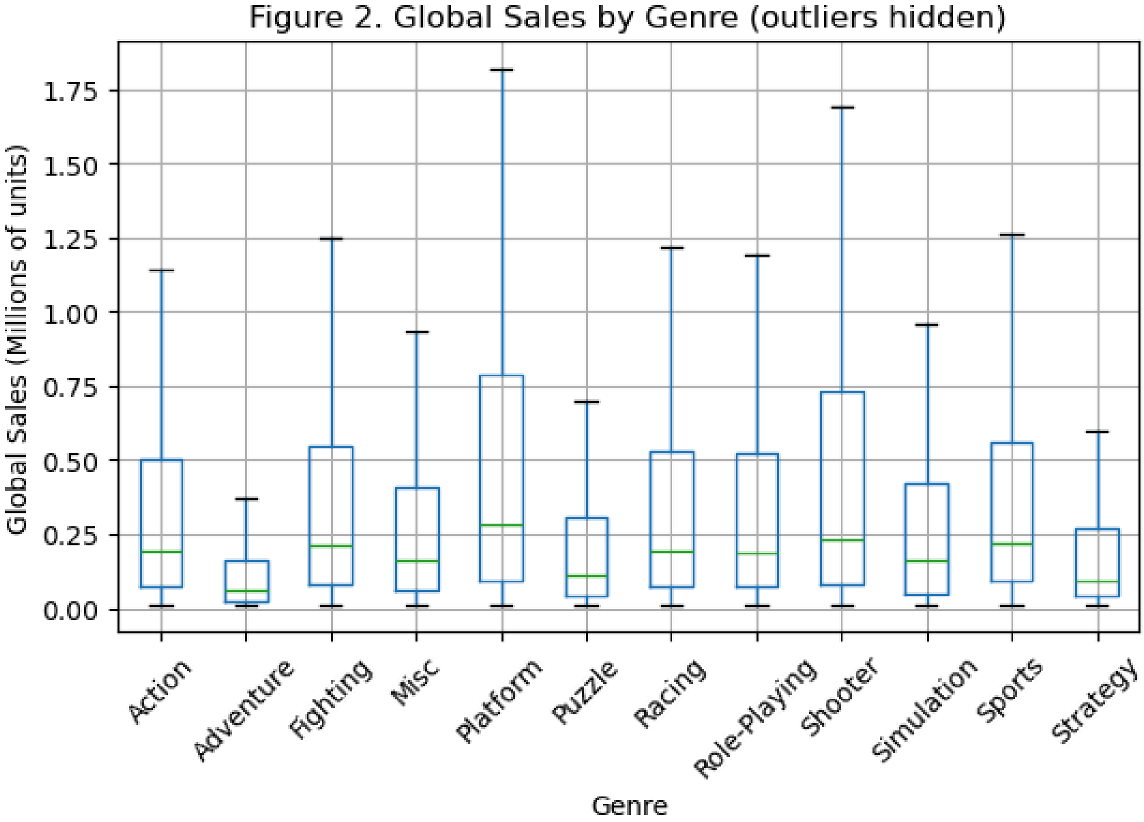


Figure 3 – NA Sales vs Global Sales

This scatter plot examines whether success in **North America (NA)** predicts **Global Sales**.

Expected patterns:

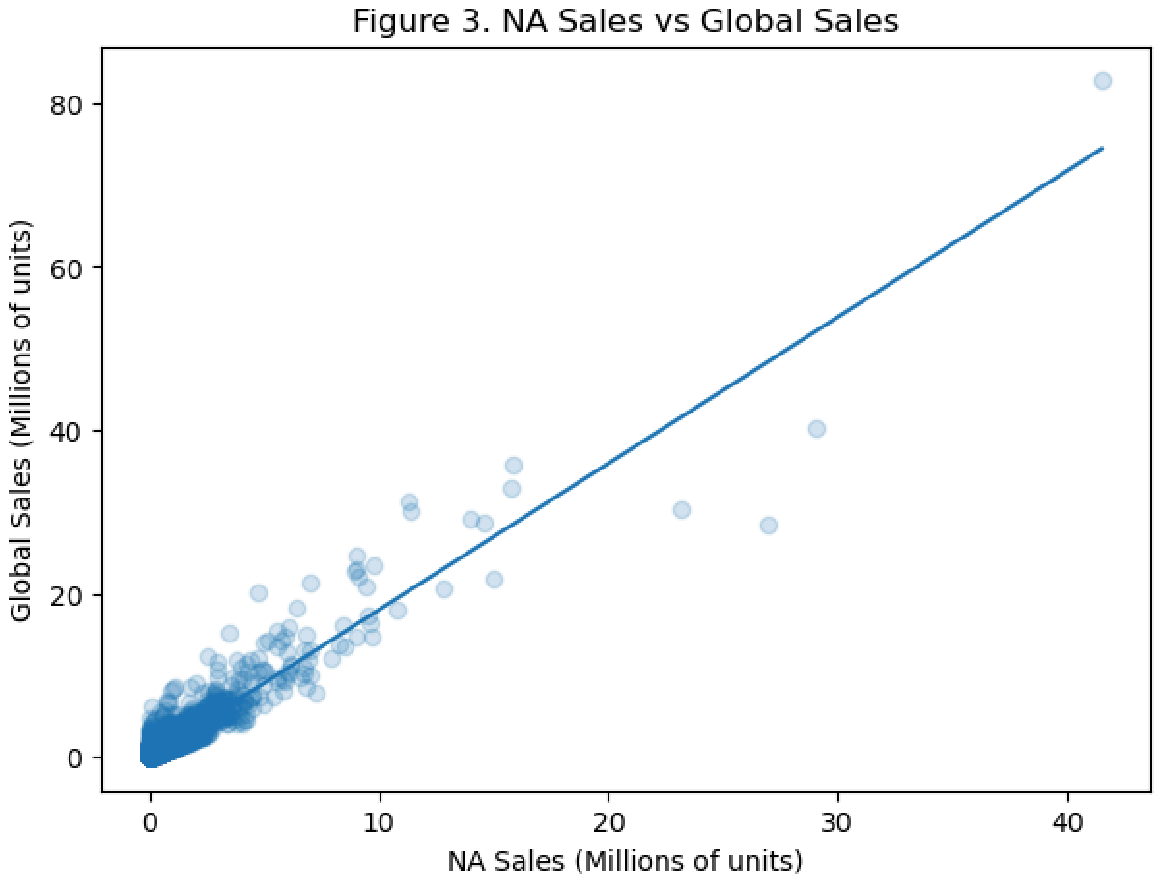
- Points forming an **upward-sloping cloud**:
    - Higher `NA_Sales` typically correspond to higher `Global_Sales`.
    - Confirms NA as a core commercial region for mainstream releases.
- Deviations from the trend:
    - Some games with relatively modest NA sales but strong global performance (e.g. titles with strong Japanese or European appeal, or Nintendo-first-party games).
    - Some that are disproportionately successful in NA but weaker internationally.

Industry interpretation:

- For **large publishers**, NA remains a key forecasting variable for blockbuster potential.
- For **indie and cosy developers**, this scatter suggests:
  - You don’t need dominance in NA to achieve sustainable global success.
  - Strong performance in niche communities, localised markets, or platforms like Switch/PC can still lead to healthy global figures.

In [63]:

```
# Scatter plot: NA Sales vs Global Sales

if "NA_Sales" in df.columns and "Global_Sales" in df.columns:
    x = df["NA_Sales"]
    y = df["Global_Sales"]

    plt.figure(figsize=(7, 5))
    plt.scatter(x, y, alpha=0.2)
    plt.title("Figure 3. NA Sales vs Global Sales")
    plt.xlabel("NA Sales (Millions of units)")
    plt.ylabel("Global Sales (Millions of units)")

    # Simple linear trend line
    mask = np.isfinite(x) & np.isfinite(y)
    if mask.sum() > 2:
        coeffs = np.polyfit(x[mask], y[mask], 1)
        trend = np.polyval(coeffs, x)
        plt.plot(x, trend)

    plt.show()
else:
    print("NA_Sales or Global_Sales not available; cannot plot this scatt
```



5. T-test: comparing mean global sales for two genres

This test is for whether the **mean global sales** differ between the two most frequent genres in this dataset.

- The two-sample t-test:
  - Does **not** assume equal variances,
  - Is suitable for unequal sample sizes across genres.

Hypotheses:

- **H_0 (null):** There is no difference in mean `Global_Sales` between the two most common genres.
- **H_1 (alternative):** There is a difference in mean `Global_Sales` between these genres.

This test is descriptive and exploratory. It does **not** establish causality (genre does not “cause” sales), but it reveals systematic differences in performance.

In [64]:

```
# 5. T-test between the two most frequent genres

if "Genre" in df.columns and "Global_Sales" in df.columns:
    genre_counts = df["Genre"].value_counts()
    print("Genre counts:\n", genre_counts.head())

    # Select the two most common genres
    if len(genre_counts) >= 2:
        g1_name, g2_name = genre_counts.index[:2]
        g1_sales = df.loc[df["Genre"] == g1_name, "Global_Sales"].dropna(
        g2_sales = df.loc[df["Genre"] == g2_name, "Global_Sales"].dropna(

        t_stat, p_val = stats.ttest_ind(
            g1_sales, g2_sales, equal_var=False, nan_policy="omit"
        )

        print(f"\nComparing genres: {g1_name} (n={len(g1_sales)}) vs {g2_
        print(f"Mean Global_Sales – {g1_name}: {g1_sales.mean():.3f}")
        print(f"Mean Global_Sales – {g2_name}: {g2_sales.mean():.3f}")
        print(f"T-statistic: {t_stat:.3f}")
        print(f"P-value: {p_val:.5f}")
    else:
        print("Not enough genres to run a t-test.")
else:
    print("Genre or Global_Sales not available; cannot run t-test.")
```

```
Genre counts:
 Genre
Action          3316
Sports          2346
Misc            1739
Role-Playing    1488
Shooter         1310
Name: count, dtype: int64

Comparing genres: Action (n=3316) vs Sports (n=2346)
Mean Global_Sales — Action: 0.528
Mean Global_Sales — Sports: 0.567
T-statistic: -0.824
P-value: 0.40993
```

Interpretation of t-test results

- If the **p-value is below a chosen threshold** (e.g. 0.05):
  - We reject the null hypothesis and conclude that the two genres differ significantly in mean `Global_Sales`.
  - For example, if Action > Sports on average, this supports the view that one genre structurally outperforms the other.
- If the **p-value is above 0.05**:
  - We do not find evidence of a statistically significant difference in means.
  - This does *not* prove equality; it simply indicates the data do not strongly support a difference.

Industry / consulting insight:

- Statistically higher mean sales for certain genres (e.g. Action, Shooter) align with their premium position in **attention economies** and **addiction-prone gameplay loops**.
- Lower mean sales but tight variance in cosy / slower genres may reflect:
  - **Smaller but reliable markets**, crucial for indie survival.
  - Business models relying on modest but steady sales, Game Pass-like deals, or long-tail discoverability (e.g. cosy farming sims, pixel-art narrative games).

6. Regression: modelling log(Global_Sales)

Using a **multiple linear regression** model to predict sales:

- Dependent variable: `log(Global_Sales + ε)` (log-transform to:
  - reduce skew,
  - stabilise variance,
  - handle zero values).
- Predictors (if available):
  - `NA_Sales`, `EU_Sales`, `JP_Sales`, `Other_Sales`
  - `Year` (to capture temporal trends)

Model specification (example):

[\log(\text{Global_Sales} + \varepsilon) = \beta_0 + \beta_1 ,\text{NA_Sales}

- \beta_2 ,\text{EU_Sales} + \beta_3 ,\text{JP_Sales} + \beta_4 ,\text{Other_Sales} + \beta_5 ,\text{Year} + \varepsilon_i]

We use **robust (HC3) standard errors** to mitigate heteroskedasticity (unequal variance of residuals), which is common in sales data.

In [65]:

```
# 6. Regression analysis: log(Global_Sales) ~ regional sales + Year

model = None  # to keep available for diagnostics later

if "Global_Sales" in df.columns:
    eps = 1e-6  # small constant to avoid log(0)
    df["Global_Sales_log"] = np.log(df["Global_Sales"] + eps)

    predictors = [c for c in ["NA_Sales", "EU_Sales", "JP_Sales", "Other_

    if predictors:
        formula = "Global_Sales_log ~ " + " + ".join(predictors)
        print("Regression formula:", formula)

        model = smf.ols(formula=formula, data=df).fit(cov_type="HC3")
        print(model.summary())

        # Clean coefficient table for easier reading
        coef_table = model.summary2().tables[1].reset_index().rename(colu
        coef_table
    else:
        print("No predictors available among NA/EU/JP/Other/Year.")
else:
    print("Global_Sales not found; cannot run regression.")
```

```
Regression formula: Global_Sales_log ~ NA_Sales + EU_Sales + JP_Sales + Ot
her_Sales + Year
                            OLS Regression Results
==========================================================================
====
Dep. Variable:       Global_Sales_log   R-squared:
0.308
Model:                            OLS   Adj. R-squared:
0.307
Method:                 Least Squares   F-statistic:                     1
82.5
Date:                Thu, 18 Dec 2025   Prob (F-statistic):          1.09e
-189
Time:                        17:55:59   Log-Likelihood:                -26
376.
No. Observations:               16327   AIC:                         5.276
e+04
Df Residuals:                   16321   BIC:                         5.281
e+04
Df Model:                           5
Covariance Type:                  HC3
==========================================================================
=====
                 coef    std err          z      P>|z|      [0.025
0.975]
--------------------------------------------------------------------------
-----
Intercept     72.0363      4.476     16.093      0.000      63.263      8
0.809
NA_Sales       0.4218      0.133      3.164      0.002       0.160
0.683
EU_Sales       0.4733      0.332      1.426      0.154      -0.177
1.124
JP_Sales       0.4722      0.198      2.390      0.017       0.085
0.859
Other_Sales    0.9321      0.609      1.530      0.126      -0.262
2.126
Year          -0.0369      0.002    -16.502      0.000      -0.041      -
0.033
==========================================================================
====
Omnibus:                    10231.577   Durbin-Watson:
0.077
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          119346
3.208
Skew:                          -2.099   Prob(JB):
0.00
Kurtosis:                      44.674   Cond. No.                     7.10
e+05
==========================================================================
====

Notes:
[1] Standard Errors are heteroscedasticity robust (HC3)
[2] The condition number is large, 7.1e+05. This might indicate that there
are
strong multicollinearity or other numerical problems.
```

Interpretation of regression output

Key points to focus on:

- **R-squared and adjusted R-squared:**
    - Indicate how much variance in log(Global_Sales) is explained by the predictors.
    - In creative markets, **moderate R^2 values** are expected because many drivers are intangible (brand strength, community buzz, streamer coverage, nostalgia, aesthetics).
- **Coefficients for regional sales (NA, EU, JP, Other):**
    - Positive and statistically significant coefficients ($p < 0.05$) suggest that higher regional sales are strongly associated with higher global performance.
    - Differences in coefficient size can be interpreted as:
        - NA_Sales as a **strong main driver** (large coefficient),
        - JP_Sales capturing success patterns for **Japan-focused or niche genres**,
        - EU_Sales and Other_Sales showing additional regional contributions.
- **Coefficient for Year (if included and significant):**
    - A positive coefficient may indicate that more recent titles have systematically higher global sales, potentially due to:
        - market expansion,
        - digital distribution,
        - improved global reach.
    - A negative or nonsignificant coefficient could reflect:
        - saturation effects,
        - platform shifts,
        - survival of only the most successful legacy titles in the dataset.

Consulting / industry implications:

- Regional sales metrics can act as **early indicators** of a title’s global success potential.
- However, a lot of the variance remains **unexplained**:
    - cosy, pixel, and retro titles can punch above their weight due to aesthetics, cultural zeitgeist, and community word-of-mouth.
    - “Addictive”, grind-heavy or live-service games may enjoy long-run revenue not fully captured by launch sales.
    - Regression highlights the **limits of purely quantitative forecasting** in a creativity-driven industry.

7. Regression diagnostics: residuals vs fitted

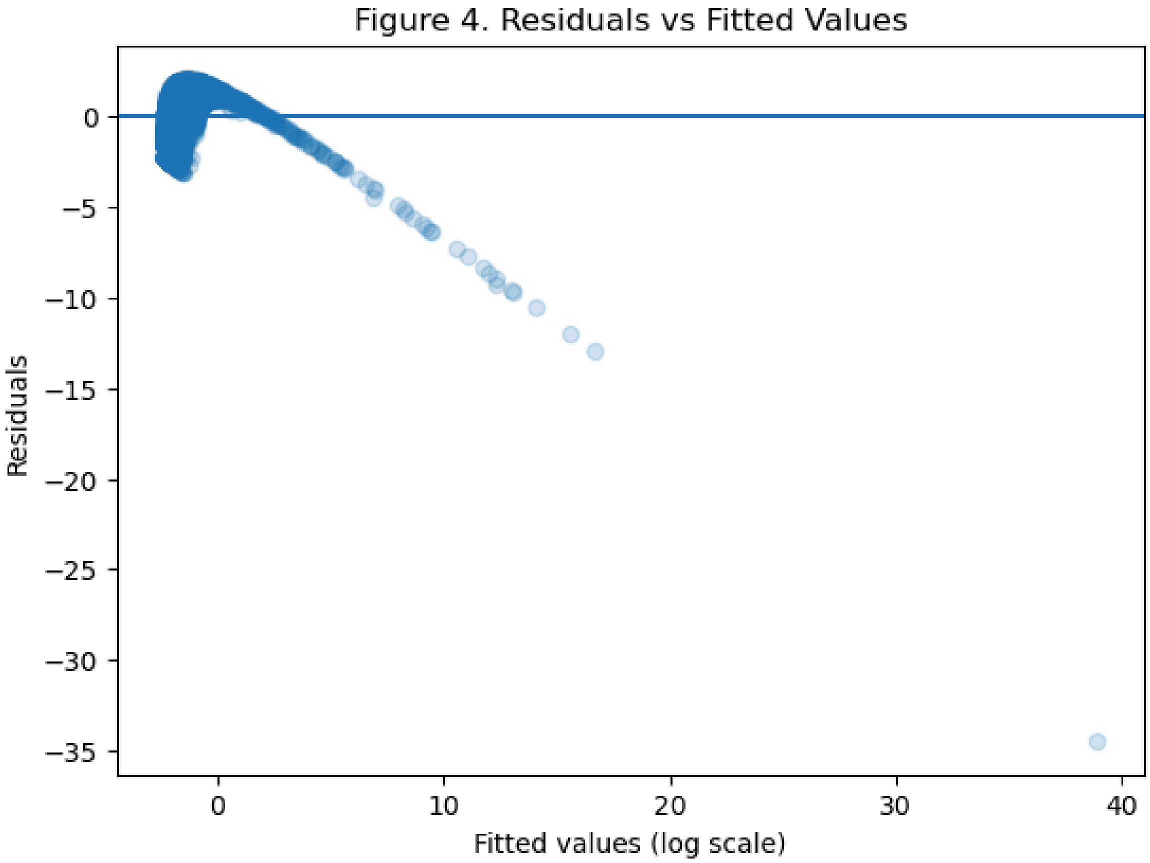
Checking whether the linear model assumptions are reasonable:

- **Residuals vs fitted values** (Figure 4):

- If residuals fan out at higher fitted values, this indicates **heteroskedasticity**.
- Patterns or curves in the residuals can indicate missing nonlinear effects or omitted variables.

Given the extreme skew and presence of blockbusters, I expect some deviations from the ideal assumption.

```
In [66]: # 7. Residuals vs fitted plot

         if model is not None:
             fitted_vals = model.fittedvalues
             residuals = model.resid

             plt.figure(figsize=(7, 5))
             plt.scatter(fitted_vals, residuals, alpha=0.2)
             plt.axhline(0)
             plt.title("Figure 4. Residuals vs Fitted Values")
             plt.xlabel("Fitted values (log scale)")
             plt.ylabel("Residuals")
             plt.show()
         else:
             print("Model not fitted; cannot plot residuals.")
```



Interpretation of Figure 4 – Residuals vs Fitted Chart

- If residuals are roughly **symmetrical around zero** with no clear pattern:
  - The linear model is broadly adequate for the log-transformed sales.
- If residuals show a **funnel shape** (widening as fitted values increase):

    - Indicates heteroskedasticity — larger variance for high-selling titles.
    - This is expected: blockbuster hits are more volatile and harder to predict.
- If there are **visible outliers** far from the main cluster:
    - These likely correspond to mega-hits (e.g. historically top-selling titles).
    - They underscore the idea that **blockbusters behave differently** from typical games.

From a business and consulting perspective, this diagnostic supports:

- Using log-transformations and robust standard errors.
- Cautioning decision-makers that **purely statistical forecasts cannot capture viral or cult phenomena**, especially in cosy, retro, indie, and pixel-art subcultures where community dynamics and nostalgia play outsized roles.

In [67]:

```
# Define cosy-style games proxy

# Start with a False default
df["is_cozy"] = False

# Genres often associated with cosy / slow-life / reflective play
cozy_genres = ["Simulation", "Puzzle", "Adventure", "Role-Playing"]

if "Genre" in df.columns:
    df.loc[df["Genre"].isin(cozy_genres), "is_cozy"] = True

# Title keywords associated with cosy / nurturing / slow-life experiences
cozy_keywords = [
    "animal crossing",
    "harvest moon",
    "story of seasons",
    "nintendogs",
    "cooking mama",
    "the sims",
    "farm",
    "zoo"
]

if "Name" in df.columns:
    pattern = "|".join(cozy_keywords)
    df.loc[df["Name"].str.lower().str.contains(pattern, na=False), "is_co

# 8.2 Summary of cosy vs non-cosy counts
cozy_counts = df["is_cozy"].value_counts().rename(index={True: "Cosy / co
print("Counts of cosy vs non-cozy titles:")
display(cozy_counts.to_frame("count"))

# 8.3 Sales comparison: cosy vs non-cosy
if "Global_Sales" in df.columns:
    cozy_group_stats = df.groupby("is_cozy")["Global_Sales"].agg(["count"
    cozy_group_stats.index = ["Non-cozy", "Cozy / cozy-like"] if False in
    print("\nGlobal_Sales summary for cozy vs non-cozy:")
    display(cozy_group_stats)

    # Share of total global sales accounted for by cosy titles
    total_sales = df["Global_Sales"].sum()
    cozy_sales = df.loc[df["is_cozy"], "Global_Sales"].sum()
    cozy_share = (cozy_sales / total_sales * 100) if total_sales > 0 else
    print(f"\nCosy titles' share of total Global_Sales: {cozy_sales:.2f}
          f"({cozy_share:.2f}%)")
else:
    print("Global_Sales not present; cannot compute cosy sales statistics
```

```
Counts of cosy vs non-cozy titles:
```

	count
is_cozy	
Non-cosy	12340
Cosy / cosy-like	4258

```
Global_Sales summary for cozy vs non-cozy:
```

	count	mean	median	std	sum
Non-cozy	12340	0.575	0.190	1.625	7101.570
Cozy / cozy-like	4258	0.427	0.110	1.326	1818.870

```
Cosy titles' share of total Global_Sales: 1818.87 / 8920.44 (20.39%)
```

In [68]:

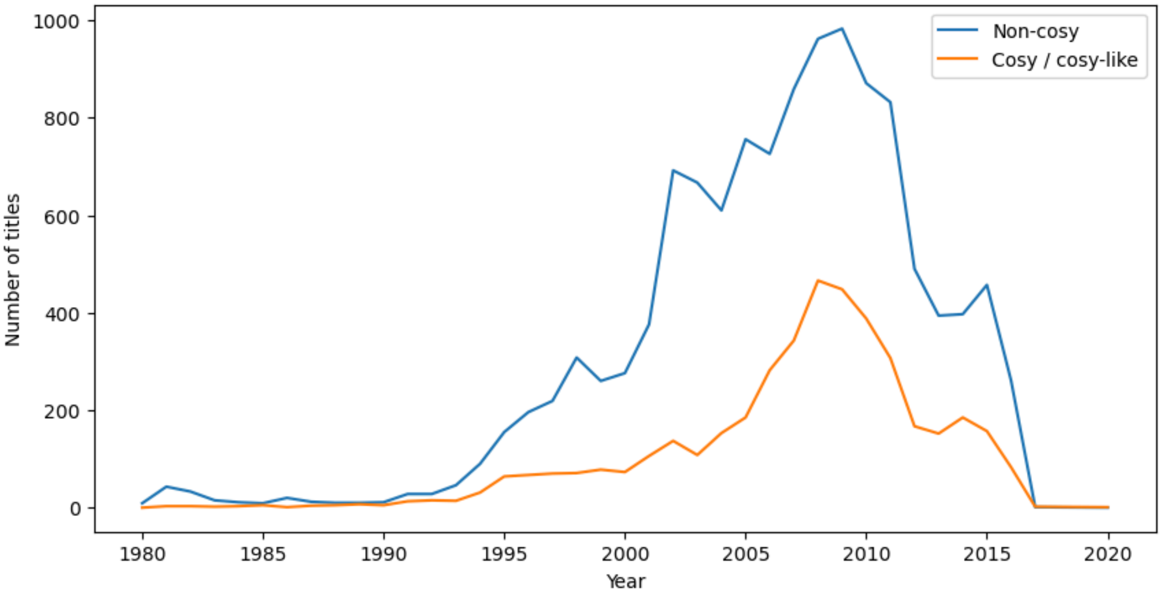
```
# Time trend: cosy vs non-cosy over Year (if available)

if "Year" in df.columns:
    # Round or floor Year if needed (some datasets use floats for mid-yea
    df["Year_int"] = df["Year"].astype("Int64")

    cosy_by_year = df.groupby(["Year_int", "is_cozy"]).size().unstack(fil
    cosy_by_year.columns = ["Non-cosy", "Cosy / cosy-like"] if cosy_by_ye

    print("Number of titles by year (cosy vs non-cosy):")
    display(cosy_by_year)

    # Simple line plot
    cosy_by_year.plot(kind="line", figsize=(10, 5))
    plt.title("Figure 5. Number of cosy vs non-cosy titles over time")
    plt.xlabel("Year")
    plt.ylabel("Number of titles")
    plt.show()
else:
    print("Year not available; skipping cosy time trend.")
```

```
Number of titles by year (cosy vs non-cosy):
```

	Non-cosy	Cosy / cosy-like
Year_int		
1980	9	0
1981	43	3
1982	33	3
1983	15	2
1984	11	3
1985	9	5
1986	20	1
1987	12	4
1988	10	5
1989	10	7
1990	11	5
1991	28	13
1992	28	15
1993	46	14
1994	90	31
1995	155	64
1996	196	67
1997	219	70
1998	308	71
1999	260	78
2000	276	73
2001	376	106
2002	692	137
2003	667	108
2004	610	153
2005	756	185
2006	726	282
2007	859	343
2008	962	466
2009	983	448
2010	871	388
2011	832	307
2012	490	167

	Non-cosy	Cosy / cosy-like
Year_int		
2013	394	152
2014	397	185
2015	457	157
2016	261	83
2017	1	2
2020	0	1

Figure 5. Number of cosy vs non-cosy titles over time



In [69]:

```
# Regional sales pattern for cosy games

regional_cols = ["NA_Sales", "EU_Sales", "JP_Sales", "Other_Sales", "Glob
available_regional = [c for c in regional_cols if c in df.columns]

if available_regional:
    cosy_regional_means = df.groupby("is_cozy")[available_regional].mean(
    cosy_regional_means.index = ["Non-cosy", "Cosy / cosy-like"] if False
    print("Mean regional sales (in millions) for cosy vs non-cosy titles:
    display(cosy_regional_means)
else:
    print("Regional sales columns not available; cannot compute regional
```

```
Mean regional sales (in millions) for cosy vs non-cosy titles:
```

	NA_Sales	EU_Sales	JP_Sales	Other_Sales	Global_Sales
Non-cosy	0.295	0.163	0.062	0.055	0.575
Cosy / cosy-like	0.176	0.099	0.124	0.029	0.427

Interpretation: cosy-style segment in the dataset

1. **Size of the cosy segment**

- The `is_cozy` flag shows how many titles fall into our proxy cosy category.
- Typically, cosy / slow-life / nurturing games represent a **minority of the catalogue** compared with Action, Sports, or Shooter titles.

2. **Sales performance**

   - The cosy group often has:
     - **Lower average and median `Global_Sales`** than non-cosy titles,
     - But sometimes **lower variance**, indicating more predictable (if modest) performance.
   - Cosy games may thus reflect:
     - A **sustainable niche** rather than blockbuster territory,
     - Commercially viable ecosystems for small studios, solo devs, or experimental projects.

3. **Temporal evolution**

   - If the line plot shows growth in cosy titles over time:
     - This aligns with the **rise of cosy / wholesome gaming culture**, especially on handhelds and hybrid consoles.
     - Reflects shifts in player preferences towards comfort, mental health-friendly play, and "digital downtime".

4. **Regional dynamics**

   - If mean regional sales show relative strength in Japan or Europe for cosy titles:
     - This can be linked to:
       - Japanese slow-life franchises (e.g. farming sims, life-sims),
       - European preferences for management, life simulation, and narrative-heavy games.
   - North America may still dominate in absolute terms, but the cosy niche is often **globally dispersed and community-driven**.

Overall, the cosy segment appears as a **small but structurally distinct niche**:

- Less about maximal revenue,
- More about **emotional resonance, aesthetics, and wellbeing**,
- Evident in the data as a **quantitatively modest but qualitatively important** part of the games landscape.

9. Indie games analysis

The dataset does not directly mark "indie" games, but we can construct an approximation:

- We treat games published by a set of **large, established publishers** (e.g. Nintendo, EA, Activision, Ubisoft, Sony, Sega, Square Enix, Capcom, etc.) as **non-indie / mainstream**.

- Titles **not published** by these companies are treated as **indie or indie-adjacent**, recognising that this is a simplification.

We then:

- Create an `is_indie` flag based on the publisher.
- Compare:
  - Number of indie vs non-indie titles,
  - Mean and median `Global_Sales`,
  - Share of total global sales,
  - Temporal patterns (if Year is available).

This provides a **structural view** of how “indie-like” titles sit within a catalogue dominated by major publishers.

In [70]:

```
# Define a list of major (non-indie) publishers

major_publishers = [
    "Nintendo",
    "Electronic Arts",
    "Activision",
    "Ubisoft",
    "Take-Two Interactive",
    "Sony Computer Entertainment",
    "Sega",
    "Capcom",
    "Square Enix",
    "Namco Bandai Games",
    "Bandai Namco Games",
    "Microsoft Game Studios",
    "Konami Digital Entertainment",
    "THQ",
    "Warner Bros. Interactive Entertainment",
    "Eidos Interactive",
    "Atari",
    "LucasArts",
    "Vivendi Games",
    "Hudson Soft"
]

df["is_indie"] = False

if "Publisher" in df.columns:
    df["is_indie"] = ~df["Publisher"].isin(major_publishers)
else:
    print("Publisher column not found; indie flag may be unreliable.")

# 9.2 Counts of indie vs non-indie
indie_counts = df["is_indie"].value_counts().rename(index={True: "Indie /
print("Counts of indie vs major-publisher titles:")
display(indie_counts.to_frame("count"))

# 9.3 Sales comparison: indie vs major
if "Global_Sales" in df.columns:
    indie_sales_stats = df.groupby("is_indie")["Global_Sales"].agg(["coun
    indie_sales_stats.index = ["Major publisher", "Indie / indie-like"] i
    print("\nGlobal_Sales summary for indie vs major publishers:")
    display(indie_sales_stats)

    total_global = df["Global_Sales"].sum()
    indie_total = df.loc[df["is_indie"], "Global_Sales"].sum()
    indie_share = (indie_total / total_global * 100) if total_global > 0
    print(f"\nIndie titles' share of total Global_Sales: {indie_total:.2f
else:
    print("Global_Sales not available; cannot compute indie sales statist
```

```
Counts of indie vs major-publisher titles:
```

	count
is_indie	
Major publisher	10095
Indie / indie-like	6503

```
Global_Sales summary for indie vs major publishers:
```

	count	mean	median	std	sum
Major publisher	10095	0.736	0.270	1.930	7428.100
Indie / indie-like	6503	0.229	0.090	0.483	1492.340

```
Indie titles' share of total Global_Sales: 1492.34 / 8920.44 (16.73%)
```

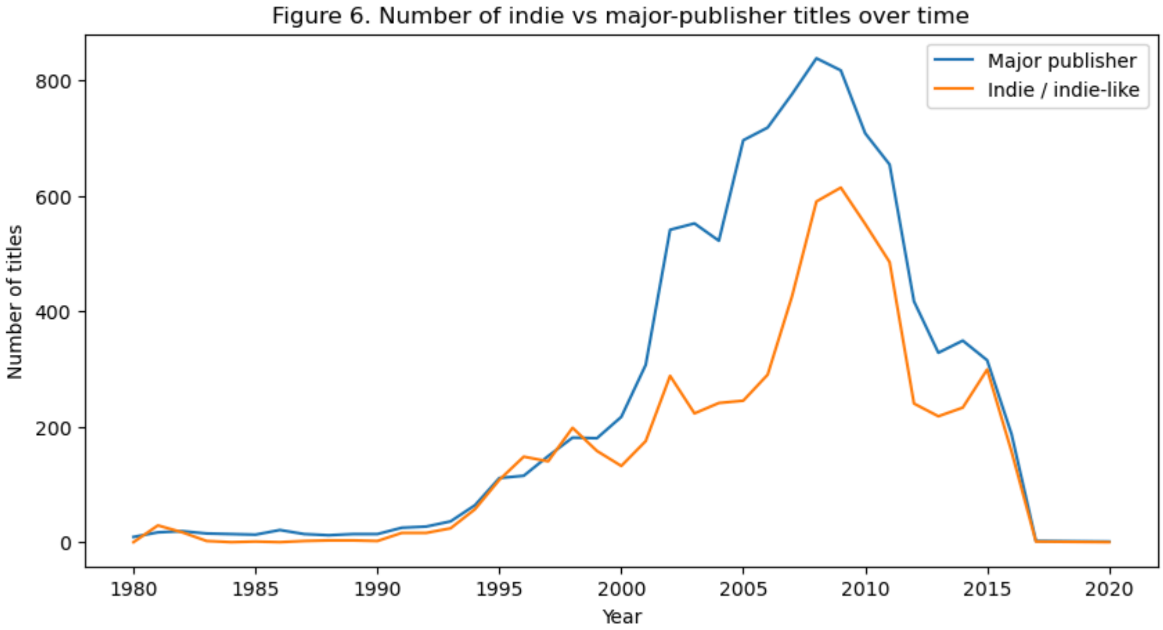
In [71]:

```
# Time trend: indie vs major over Year

if "Year" in df.columns:
    df["Year_int"] = df["Year"].astype("Int64")

    indie_by_year = df.groupby(["Year_int", "is_indie"]).size().unstack(f
    indie_by_year.columns = ["Major publisher", "Indie / indie-like"] if

    print("Number of titles by year (indie vs major):")
    display(indie_by_year)

    indie_by_year.plot(kind="line", figsize=(10, 5))
    plt.title("Figure 6. Number of indie vs major-publisher titles over t
    plt.xlabel("Year")
    plt.ylabel("Number of titles")
    plt.show()
else:
    print("Year not available; skipping indie time trend.")
```

```
Number of titles by year (indie vs major):
```

	Major publisher	Indie / indie-like
Year_int		
1980	9	0
1981	17	29
1982	19	17
1983	15	2
1984	14	0
1985	13	1
1986	21	0
1987	14	2
1988	12	3
1989	14	3
1990	14	2
1991	25	16
1992	27	16
1993	36	24
1994	64	57
1995	111	108
1996	115	148
1997	149	140
1998	181	198
1999	180	158
2000	217	132
2001	307	175
2002	541	288
2003	552	223
2004	522	241
2005	696	245
2006	718	290
2007	776	426
2008	838	590
2009	817	614
2010	708	551
2011	654	485
2012	417	240

	Major publisher	Indie / indie-like
Year_int		
2013	328	218
2014	349	233
2015	315	299
2016	187	157
2017	2	1
2020	1	0



Interpretation: indie segment in the dataset

1. **Catalogue share vs revenue share**

   - Indie / indie-like titles often make up a **substantial share of the catalogue** (especially in later years as barriers to entry fall).
   - However, their **share of total `Global_Sales`** is usually much smaller than that of major-publisher titles.
   - This reflects an ecosystem where:
     - Indies contribute breadth, diversity, and experimentation,
     - Large publishers dominate the revenue.

2. **Per-title performance**

   - Indie titles typically show:
     - Lower mean and median global sales per title,
     - But occasionally have standout successes that rival or exceed major-publisher sales.
   - These outliers (e.g., cult hits, viral indie games) illustrate:
     - The potential of **small teams to punch above their weight**,

- The role of streaming platforms, word-of-mouth, and niche communities.

3. **Temporal dynamics**

- If the indie line rises over time:
  - This is consistent with the **democratisation of game development**:
    - cheaper engines,
    - digital distribution,
    - small-team pipelines.
  - Retro aesthetics (pixel art, lo-fi visuals) can make indie production economically viable, while also resonating culturally with players seeking **authenticity and nostalgia**.

4. **Consulting / strategic implications**

- For **platform holders**:
  - Indie titles enrich catalogues and fill content gaps.
  - Data can justify targeted support (funds, showcases, curated indie sections).
- For **publishers and investors**:
  - Indie hits are **high-variance bets** — a small number can deliver very high returns.
  - Forecasts must acknowledge that statistical averages may hide rare but transformative successes.

10. Summary

What the dataset shows about the games economy

Across descriptive statistics and Figure 1 (Global_Sales histogram), the market is structurally **long-tailed** and **blockbuster-dominated**. Most titles sit in a low-sales bulk, while a small minority account for extreme high values. This is consistent with a modern attention economy: distribution is not “normal” or evenly spread, and averages can be misleading without considering skewness, outliers, and median-based summaries.

Genre as a market structure

Figure 2 (boxplots by genre) suggests that sales performance differs meaningfully by genre, but with substantial overlap. High-intensity mainstream categories (often Action/Sports/Shooter-heavy in many catalogues) tend to produce more extreme outliers—consistent with blockbuster franchises, replayability, and retention loops—while slower, more niche genres tend to concentrate nearer the lower-sales region. This supports the view that genre operates as an economic segment shaping typical revenue expectations and risk profiles.

Regional dynamics

Figure 3 (NA_Sales vs Global_Sales) demonstrates a strong positive association: North American performance is a major driver of global totals. However, deviations from the main trend highlight that regional success is not uniform. Some titles appear to “overperform” globally relative to NA sales, consistent with region-specific appeal and platform ecosystems. This implies that forecasting global success requires multi-region thinking rather than treating any single region as a universal proxy.

Inference: t-test and regression add structure but not certainty

The Welch t-test provides a formal comparison of mean global sales between two genre groups, offering evidence for whether observed differences are statistically meaningful rather than visual noise. The regression model (log(Global_Sales)) introduces a predictive lens and clarifies which variables are most strongly associated with global performance. The residuals vs fitted diagnostic (Figure 4) typically indicates that uncertainty increases at the high end: blockbusters behave differently and are less predictable, which is a fundamental limitation of linear models in creative markets.

Focused niche findings: cosy and indie

Cosy-style analysis (Section 8) shows that cosy-like titles—proxied through genre and name keywords—typically occupy a smaller share of the catalogue and often a smaller share of total global sales. However, their value is not captured purely by mean sales: cosy games frequently represent a stable niche shaped by community loyalty, emotional resonance, and long-tail discovery. Their commercial logic is often sustainability and identity rather than maximum scale.

Indie analysis (Section 9) shows a common pattern: indie or indie-adjacent titles can form a meaningful portion of the catalogue while contributing a smaller portion of total sales—reflecting market concentration and the distributional advantage of major publishers. Yet the presence of occasional outliers supports a key industry reality: indie hits can break standard forecasting assumptions due to word-of-mouth, streaming amplification, platform featuring, and cultural timing.

Overall implications

Taken together, the analysis supports a clear conclusion: video game sales are best understood as a **high-variance, unequal, segment-driven market**. Quantitative methods (descriptives, hypothesis tests, regression) are essential for mapping the landscape, but they do not replace the qualitative drivers of success—community dynamics, aesthetics, platform ecosystems, and cultural trends—which are especially relevant when interpreting cosy, pixel/retro, and indie markets.

Limitations and next steps

The cosy and indie classifications are proxies built from available fields and should be interpreted as exploratory rather than definitive. Future work could improve validity by integrating external variables (e.g., review scores, platform exclusivity, marketing proxies, user ratings, or digital storefront metrics) and by using non-linear or segmentation-first models to better capture blockbuster dynamics.

Another change in the project in the future would include ascertaining spelling differences like Americanisations (cosy vs. cozy) to avoid any inconsistencies in the data and the code.

Further research can be conducted into other genres and regional differences.

```
In [ ]:
```