
Sam Taylor

# Data Pipeline & Analysis

August 2024

# TABLE OF CONTENTS

# PROJECT OVERVIEW

# PROJECT AIMS

The aim of this project is to create an end-to-end ETL pipeline to extract data from a website and present the findings.

**01.** To scrape tabular Dead by Daylight data from the following website: https://dennisreep.nl/dbd/

**02.** To model the data into staging, dimension & fact tables

**03.** To clean the data to aid in data analysis

**04.** To feature engineer metrics to aid in data analysis

**05.** To analyse the final data tables in order to understand the game, its design and its player base.

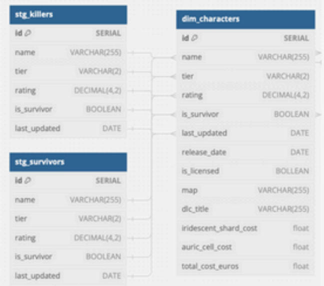
# PROJECT PROCESS

In order to build our end-to-end ETL pipeline, the project will be split into 5 steps.

## 01.


**Scrape**

## 02.




**Model**

## 03.

```
# Drop the 'unix' column
fact_matches.drop(columns=['unix'], inplace=True)
```

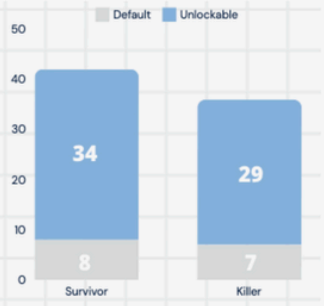
**Clean**

## 04.

```
# Convert DLC data to a more accessible format
dlc_info = {}
for dlc in dlc_data:
    for role in ['Killer', 'Survivor(s)']:
        characters = dlc[role].split(', ') if dlc[role] else []
        for character in characters:
            dlc_info[character] = {
                'Release Date': dlc['Release Date'],
                'is_licensed': 1 if '†' in dlc['Title'] else 0,
                'Map': dlc['Map (Realm)'] if 'Map (Realm)' in dlc else '',
                'DLC Title': dlc['Title']
            }
```

**Feature engineer**

## 05.




**Analyse**

# PROJECT QUESTIONS

The aim of collecting the data is to provide an understanding of the game, its design and its player base.

**01.** What is the **least amount of money** you could pay to play the game with all its content?

**02.** What **unwritten player norms** are there in the online player base?

**03.** Is the game **biased** towards one category of player?

**04.** Is the game **biased by design** or **by player skill**?

**05.** Which **perks** tend to be used together together and why?

# GAME OVERVIEW

# GAME OVERVIEW

- Dead by Daylight (DBD) is a popular online multiplayer game.
- Each game has 4 survivors & 1 killer on a randomly selected map.
- Survivors have to repair 5 generators to power an exit gate, so that they can escape.
- Killers have to stop the reparation of the generators and hook the survivors to slow their progress & prevent them from escaping.

# DATA HIERARCHY

# DATA HIERARCHY

- **The data tables are organised into 5 categories:**
  - Character | Map | Perk | Add-on | Match
- **The following table types are used:**
  - **Temporary Tables (temp):** Used to hold data temporarily during ETL processes for intermediate processing.
  - **Staging Tables (stg):** Used to store raw data from source system before it is processed.
  - **Dimension Tables (dim):** Used to store descriptive attributes that provide context and details for data analysis.
  - **Fact Tables (fact):** Used to store quantitative data for analysis and reporting.

# Data hierarchy: scrape

In order to acquire data to analyse, data was scraped, with permission, from the following website:

- https://dennisreep.nl/dbd/

This code has two main functions: **url_get_contents()** and **scrape_tables()**.

The **url_get_contents** function fetches the content of a given URL using a specific user-agent to mimic a browser.

The **scrape_tables** function extracts tables from this HTML content within a specified range (start & end date) and combines these tables into a single dataframe. Then, the dataframe is assigned to a variable name.

The following slides show the dataframes returned.

```python
# Function to get the contents of the URL
def url_get_contents(url):
    req = urllib.request.Request(url=url, headers={'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64;
x64; rv:90.0) Gecko/20100101 Firefox/90.0'})
    f = urllib.request.urlopen(req)
    return f.read()

# Function to scrape tables and format column headers
def scrape_tables(url, start, end, dataframe_name):
    # Defining the HTML contents of the URL
    xhtml = url_get_contents(url).decode('utf-8')

    # Defining the HTMLTableParser object
    p = HTMLTableParser()

    # Feeding the HTML contents to the HTMLTableParser object
    p.feed(xhtml)

    # Initialize a list to store DataFrames
    dataframes = []

    # Loop through the specified range of tables
    for i in range(start, end + 1):
        # Create an empty list to store the split strings
        split_result = []

        # Iterate through each list in the table header (first row)
        for column in p.tables[i][0]:
            # Join all elements in the inner list into a string
            ini_string = ''.join(column)

            # Apply getVals operation to ini_string
            getVals = [val.lower() for val in ini_string if val.isalnum() or val.isspace() or val ==
'-']

            # Add underscores to separate words
            result = '_'.join(''.join(getVals).split()).replace('-', '_')

            # Append the split result to the list
            split_result.append(result)

        # Create a DataFrame for the current table
        table_df = pd.DataFrame(p.tables[i][1:], columns=split_result)

        # Append the DataFrame to the list
        dataframes.append(table_df)

    # Concatenate all DataFrames into one
    concatenated_df = pd.concat(dataframes, ignore_index=True)

    # Assign variable name
    globals()[dataframe_name] = concatenated_df

    # Return dataframe
    return concatenated_df
```

# DATA HIERARCHY: CHARACTERS

## STG_KILLERS

| | |
|---|---|
| name | STRING(PK) |
| tier | CATEGORY |
| rating | FLOAT |
| is_survivor | BOOLEAN |
| last_updated | DATETIME |

## STG_SURVIVORS

| | |
|---|---|
| name | STRING(PK) |
| tier | CATEGORY |
| rating | FLOAT |
| is_survivor | BOOLEAN |
| last_updated | DATETIME |

## DIM_CHARACTERS

| | |
|---|---|
| name | STRING(PK) |
| tier | CATEGORY |
| rating | FLOAT |
| is_survivor | BOOLEAN |
| last_updated | DATETIME |
| release_date | DATETIME |
| is_licensed | BOOLEAN |
| map | STRING |
| dlc_title | STRING |
| iridescent_shard_cost | FLOAT |
| auric_cell_cost | FLOAT |
| total_cost_euros | FLOAT |

**INFO**

Playerable characters are split into 2 categories: survivors & killers

# DATA HIERARCHY: MAPS

## STG_MAPS_KILLER

map_name ....... STRING(PK)
rating ........................... FLOAT
last_updated ........ DATETIME
killer_name ............... STRING

## DIM_MAPS

map_name ....... STRING(PK)
avg_score .................. FLOAT
bias .................................. INT
last_updated ...... DATETIME

**INFO**

Each game is played on one of several maps

# DATA HIERARCHY: PERKS

### STG_PERKS_SURVIVOR

| | |
|---|---|
| perk_name | STRING(PK) |
| description | STRING |
| acquired_from | STRING |
| tier | CATEGORY |
| rating | FLOAT |
| last_updated | DATETIME |
| is_survivor | BOOLEAN |

### STG_PERKS_KILLER

| | |
|---|---|
| perk_name | STRING(PK) |
| description | STRING |
| acquired_from | STRING |
| tier | CATEGORY |
| rating | FLOAT |
| last_updated | DATETIME |
| is_survivor | BOOLEAN |

### DIM_PERKS

| | |
|---|---|
| perk_name | STRING(PK) |
| description | STRING |
| acquired_from | STRING |
| tier | CATEGORY |
| rating | FLOAT |
| last_updated | DATETIME |
| is_survivor | BOOLEAN |
| category | STRING |

**INFO**

Perks provide characters with special powers.

# DATA HIERARCHY: KILLER PERK RATINGS

**STG_PERKS_KILLER_SPECIFIC**

| | |
|---|---|
| perk_name | STRING(PK) |
| description | STRING |
| acquired_from | STRING |
| tier | CATEGORY |
| rating | FLOAT |
| last_updated | DATETIME |
| killer_name | STRING |
| category | STRING |

**FACT_PERKS_KILLER_SPECIFIC**

| | |
|---|---|
| perk_name | STRING(PK) |
| description | STRING |
| acquired_from | STRING |
| tier | CATEGORY |
| rating | FLOAT |
| last_updated | DATETIME |
| killer_name | STRING |
| category | STRING |

**FACT_PERKS_KILLER_SPECIFIC_WIDE**

| | |
|---|---|
| perk_name | STRING(PK) |
| description | STRING |
| acquired_from | STRING |
| rating | FLOAT |
| last_updated | DATETIME |
| killer_name | STRING |
| category | STRING |
| tier_unknown | FLOAT |
| tier_a | FLOAT |
| tier_b | FLOAT |
| tier_c | FLOAT |
| tier_d | FLOAT |
| tier_f | FLOAT |
| tier_s | FLOAT |

**INFO**

Each character has 4 unique perks that can be taught to others

# DATA HIERARCHY: PERKS

**FACT_PERKS**

| | |
|---|---|
| perk_name | STRING(PK) |
| description | STRING |
| acquired_from | STRING |
| overall_tier | CATEGORY |
| overall_rating | FLOAT |
| is_survivor | BOOLEAN |
| last_updated | DATETIME |
| category | STRING |
| tier_unknown | FLOAT |
| tier_a | FLOAT |
| tier_b | FLOAT |
| tier_c | FLOAT |
| tier_d | FLOAT |
| tier_f | FLOAT |
| tier_s | FLOAT |

**INFO**

Up to 4 perks can be used at the same time.

# DATA HIERARCHY: ADDONS

### STG_ADDONS_KILLER

| | |
|---|---|
| name | STRING |
| description | STRING |
| tier | CATEGORY |
| rating | FLOAT |
| last_updated | DATETIME |
| killer_name | STRING |

### DIM_ADDONS_KILLER

| | |
|---|---|
| killer_addon_id | STRING(PK) |
| name | STRING |
| description | STRING |
| tier | CATEGORY |
| rating | FLOAT |
| last_updated | DATETIME |
| killer_name | STRING |

**INFO**

Add-ons are used to enhance characters’ powers

# DATA HIERARCHY: MATCHES

## FACT_MATCHES

| | |
|---|---|
| fact_match_id | STRING(PK) |
| date | DATETIME |
| match | INTEGER |
| character | STRING |
| is_survivor | BOOLEAN |
| is_data_recorder | BOOLEAN |
| perk1 | STRING |
| perk2 | STRING |
| perk3 | STRING |
| perk4 | STRING |
| perks_equipped_count | INTEGER |
| map | STRING |
| generators_complete | FLOAT |
| bloodpoints | INTEGER |
| notes | STRING |
| surv_went_to_2nd_phase | BOOLEAN |
| killer_alllowed_surv_escape | BOOLEAN |
| attempted_adept | BOOLEAN |
| killer_didn't_stay | BOOLEAN |
| surv_died_on_2nd_hook | BOOLEAN |
| surv_was_tunneled | BOOLEAN |
| player_disconnected | BOOLEAN |
| surv_hatch_escape | BOOLEAN |
| surv_moried | BOOLEAN |
| surv_threw_first_hook | BOOLEAN |
| surv_threw_second_hook | BOOLEAN |
| killer_friendly | BOOLEAN |
| match_cancelled | BOOLEAN |
| player_afk | BOOLEAN |
| killer_4k | BOOLEAN |
| killer_3k | BOOLEAN |
| killer_win | BOOLEAN |
| draw | BOOLEAN |
| survivor_won | BOOLEAN |

**INFO**

Fact_matches consists of ~900 individual games recorded by a player

```python
def format_dict(d):
    return "\n".join(f"{key}: {value}" for key, value in d.items())

def dataframe_report(df):
    report = {}

    # .info() information
    buffer = io.StringIO()
    df.info(buf=buffer)
    report['info'] = buffer.getvalue()

    # Summary statistics for numerical columns rounded to 2 decimal places
    report['summary_statistics'] = df.describe().round(2).to_dict()

    # Number of duplicates in the dataframe
    num_duplicates = df.duplicated().sum()
    report['num_duplicates'] = num_duplicates

    # Duplicate rows in the dataframe
    duplicate_rows = df[df.duplicated()]
    report['duplicate_rows'] = duplicate_rows

    # Number of blanks in each column (assuming blanks are empty strings)
    blank_counts = df.apply(lambda x: (x == '').sum())
    report['num_blanks'] = blank_counts[blank_counts > 0].to_dict()

    # Number of nulls in each column
    null_counts = df.isnull().sum()
    report['num_nulls'] = null_counts[null_counts > 0].to_dict()

    # Data types of each column
    report['data_types'] = df.dtypes.to_dict()

    # Number of unique values in each column
    report['num_unique_values'] = df.nunique().to_dict()

    # Constant columns (columns with a single unique value)
    report['constant_columns'] = [col for col in df.columns if df[col].nunique() == 1]
```

# Data hierarchy: clean

Now that we have data, the last step is to ensure that the dataframes are clean & fix any issues there are.

The **dataframe_report()** function generates a report containing:

- Basic dataframe information
- Summary statistics
- Number of duplicates
- Counts of blanks and nulls
- A list of affected rows for duplicates, blanks & nulls
- Number of unique values
- Constant columns

The results are stored in a dictionary called report, which is later printed (not pictured) for the user, in an easy-to-digest summary.

# ANALYSIS

# Characters

Characters are categorised as '**killer**' or '**survivor**'. They either come with the base game (default) or can be purchased additionally (unlockable).

**78**

**Total Characters**

**42** (54%)

**Survivors**

**36** (46%)

**Killers**

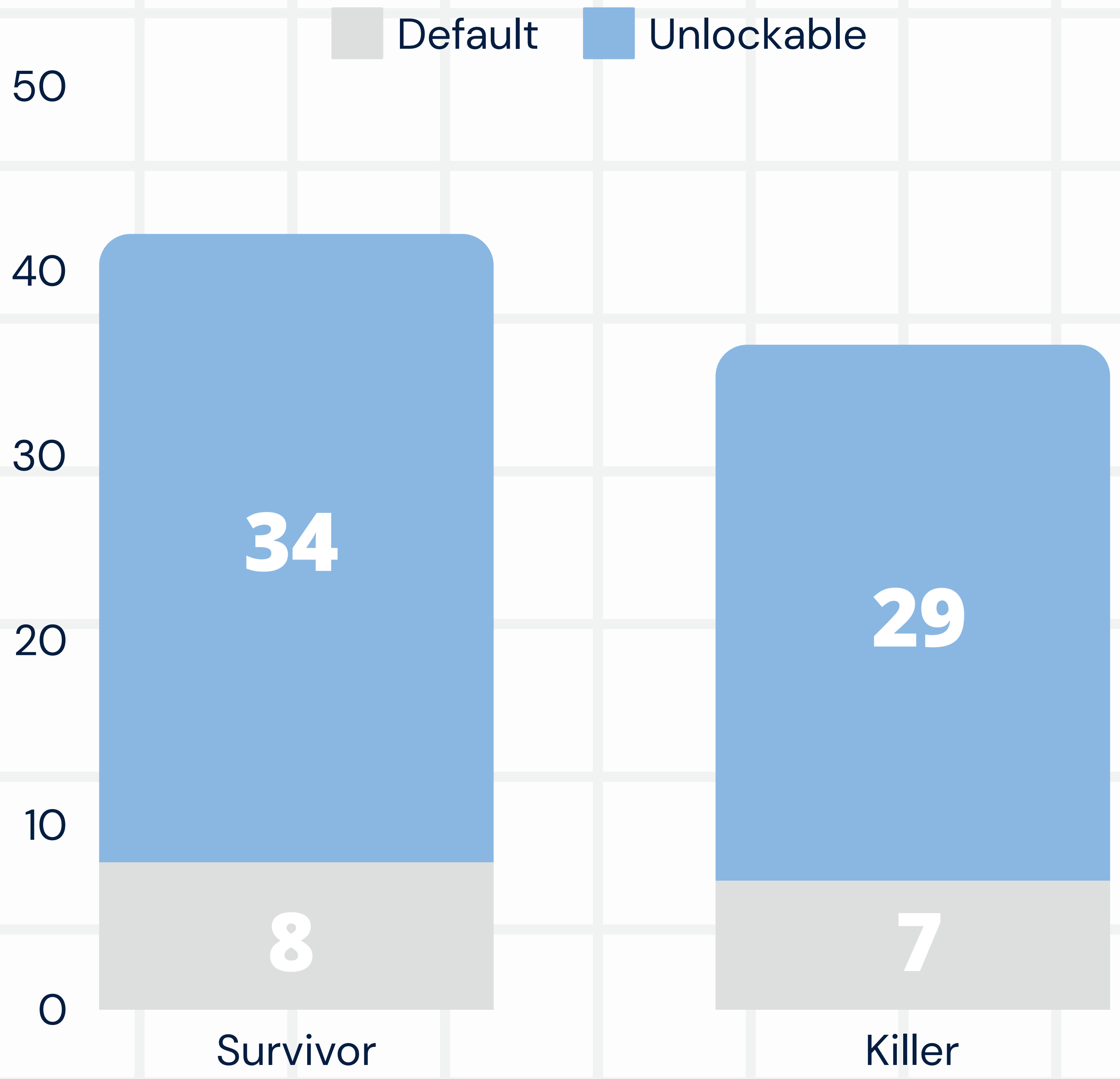
**63** (81%)

**Unlockable characters**

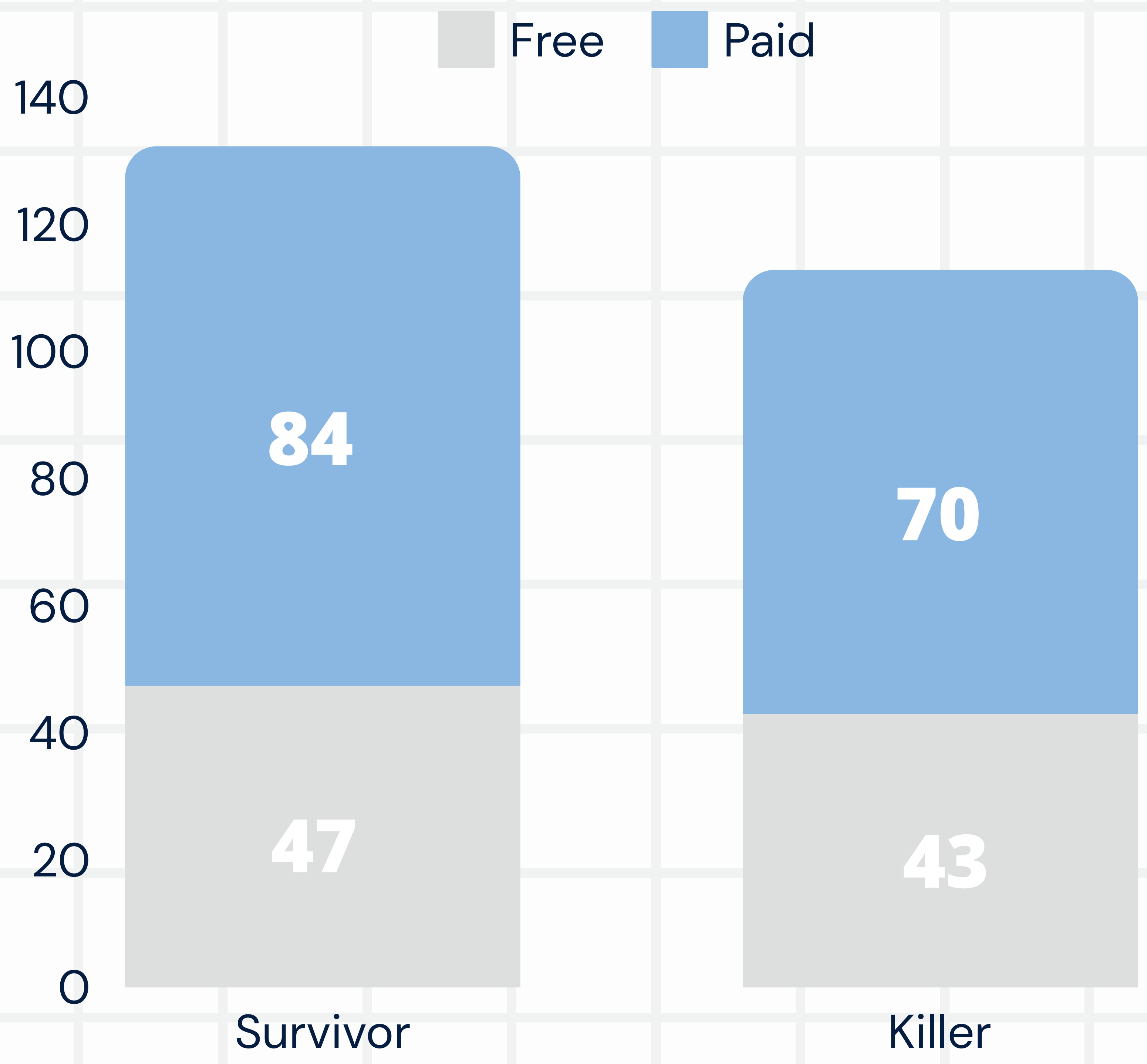
**15** (19%)

**Default characters**

## TOTAL CHARACTERS vs CATEGORY



## TOTAL COST OF CHARACTERS (€)




# Character cost

Characters can either be purchased with in-game currency (iridescent shards) or with real money. Some characters can only be purchased with real money.

**63**

**Unlockable characters**

**32** (51%)

**Free characters**

**31** (49%)

**Paid characters**

**90€**

**Free characters**

**155€**

**Paid characters**

# Game cost

**[Total]**

Dead by Daylight involves several costs to access its full content. How much could be spent per player, if a player bought everything?

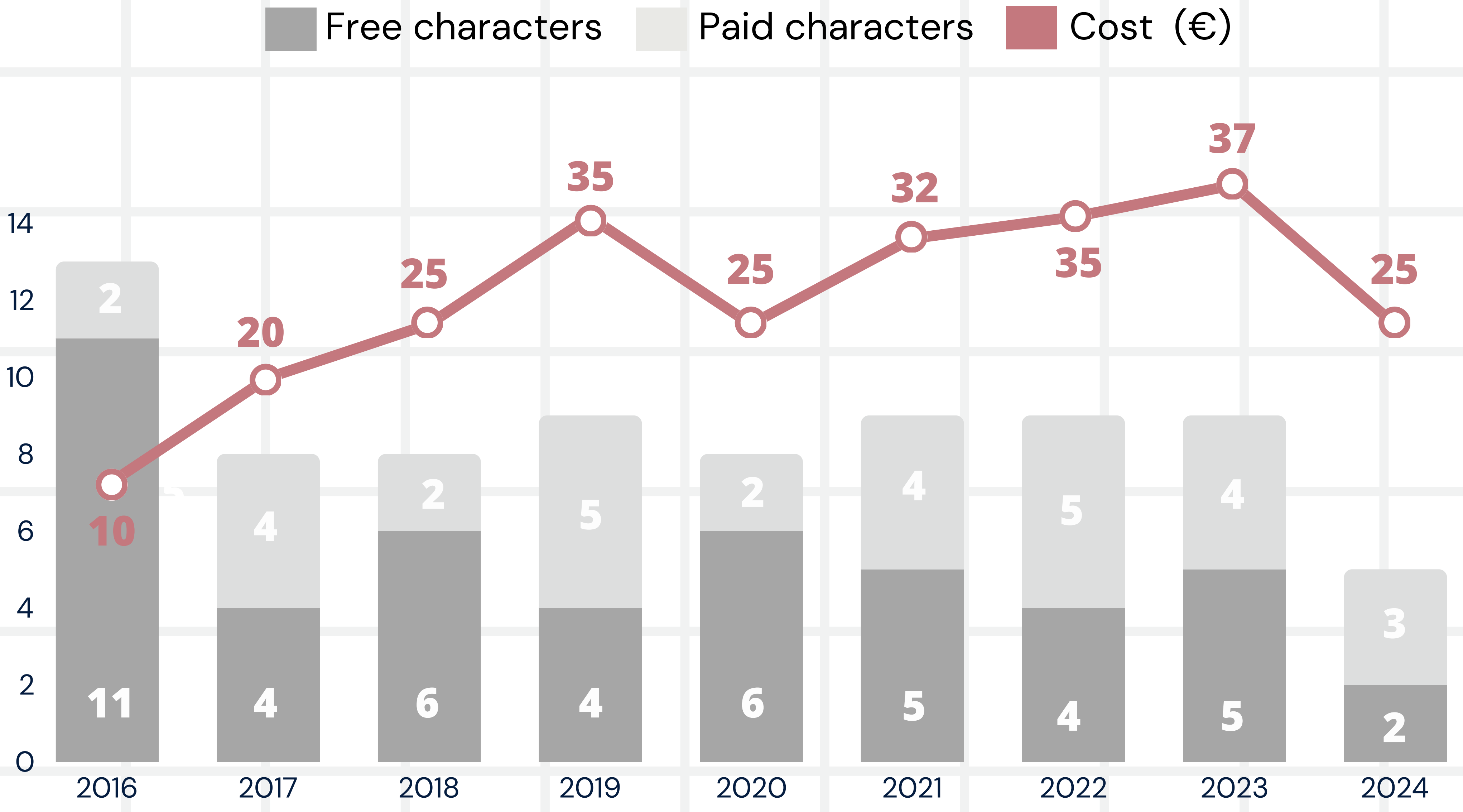
**275€**
**Total cost**

**30€**
**Base game cost**

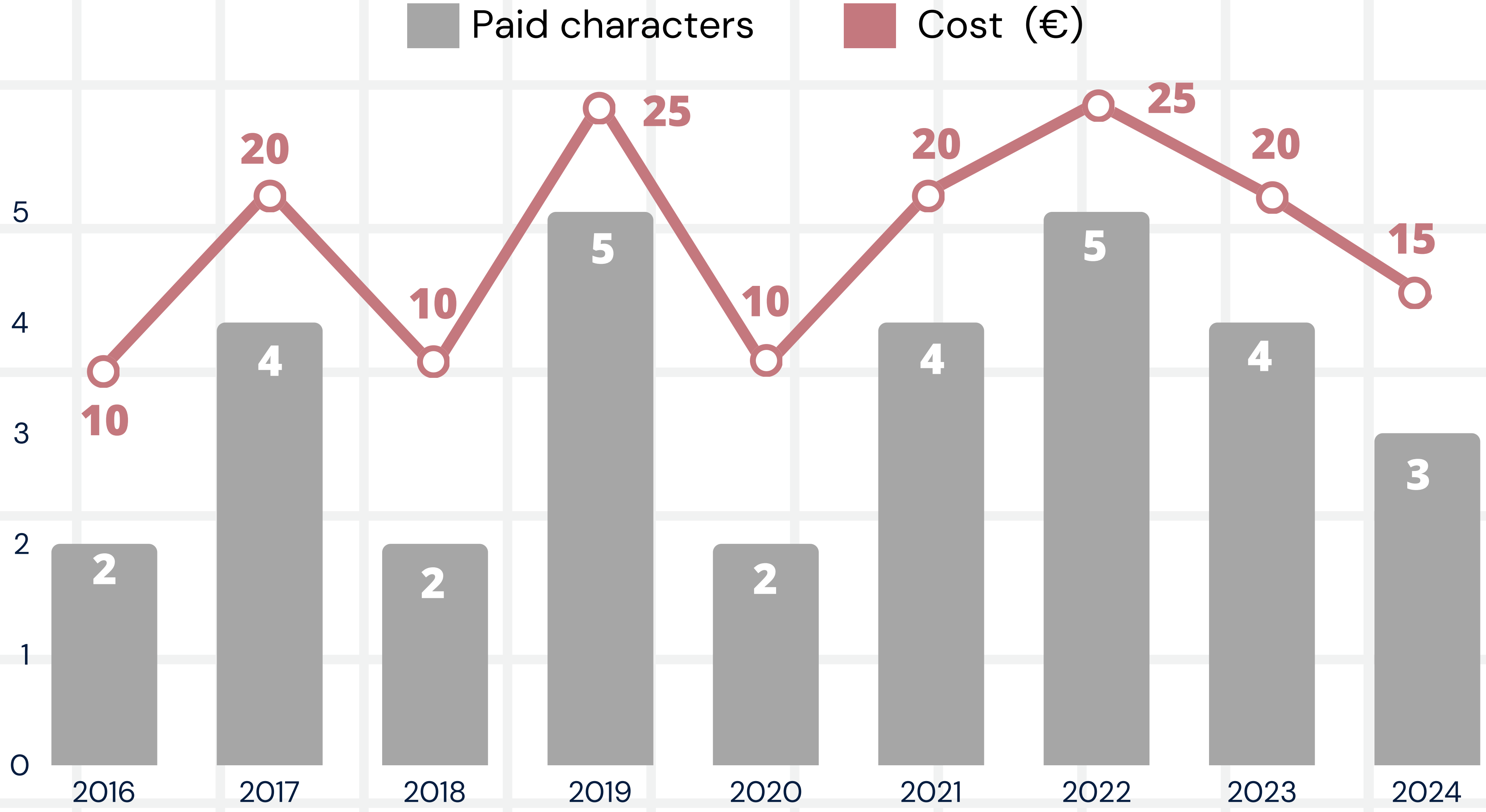
**245€**
**Total character cost**

**26.50€**
**Yearly average**

## TOTAL CHARACTER RELEASES vs PRICE



# PAID CHARACTER RELEASES vs PRICE




# Game cost

**[Not-free]**

How much money do you absolutely need to spend to get all available content?

**185€**
**Total cost**

**30€**
**Base game cost**

**155€**
**Total character cost**

**17.22€**
**Yearly average**

# ANALYSIS

# Player behaviour

Depending on how the game is progressing, players can choose to exploit the game in certain ways.

The term **'throwing'** refers to the act of a survivor deliberately getting themselves killed, so that they can leave the game quicker.

The majority of behaviours recorded over ~900 matches were seen as negative. Only a small number of behaviours would be considered as positive.

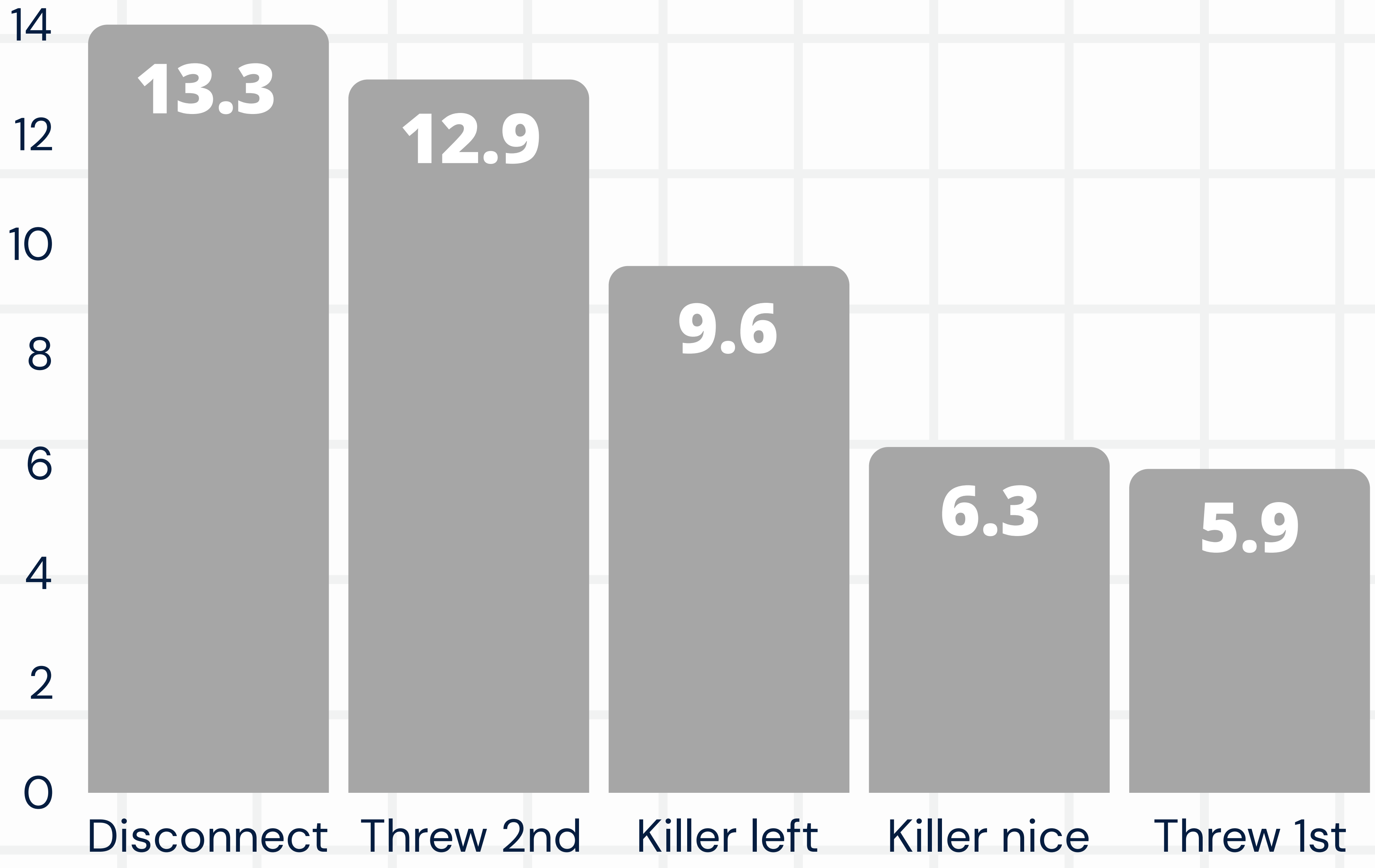
**43%**

of events witnessed were negative

**14%**

of events witnessed were positive

## PLAYER BEHAVIOUR EVENTS (%)




Events are not mutually exclusive*

# [CORRELATION ANALYSIS]
# KILLER BEHAVIOUR

| | killer_win | perks_equipped_count | bloodpoints | killer_allowed_survivor_escape | player_attempted_adept | killer_didn't_stay | player_disconnected | killer_was_friendly |
|---|---|---|---|---|---|---|---|---|
| # perks | 0.22 | | | | | | | |
| bloodpoints | 0.24 | 0.23 | | | | | | |
| killer_allowed_escape | -0.038 | 0.027 | -0.0042 | | | | | |
| adept | 0.085 | -0.011 | 0.00059 | -0.0032 | | | | |
| killer_left | -0.24 | -0.49 | -0.24 | -0.016 | -0.022 | | | |
| diconnected | -0.11 | 0.023 | -0.11 | -0.0069 | -0.0097 | -0.047 | | |
| killer_friendly | -0.21 | 0.063 | -0.0091 | -0.012 | -0.018 | -0.054 | -0.038 | |
| match_cancelled | -0.093 | 0.016 | -0.088 | -0.0056 | -0.0079 | -0.038 | -0.017 | -0.031 |

# Killer behaviour

A correlation analysis of the effect of various killer-related variables on each other.

**Key findings**

- Positive behaviours are punished by the game design **[killer_friendly]**
- Killers with fewer perks equipped are (i) less likely to stay and (ii) less likely to win, as a result **[perks_equipped, killer_left]**

# [CORRELATION ANALYSIS] SURVIVOR BEHAVIOUR

| | killer_win | # perks | bloodpoints | killer_allowed_escape | died_on_2nd_hook | tunneled | hatch_escape | threw_1st_hook |
|---|---|---|---|---|---|---|---|---|
| # perks | -0.036 | | | | | | | |
| bloodpoints | -0.079 | 0.056 | | | | | | |
| killer_allowed_escape | -0.015 | -0.00039 | 0.054 | | | | | |
| died_on_2nd_hook | 0.18 | 0.0085 | -0.028 | -0.027 | | | | |
| tunneled | 0.028 | -0.0011 | -0.052 | -0.013 | -0.025 | | | |
| hatch_escape | 0.22 | 0.016 | 0.081 | -0.022 | -0.042 | -0.02 | | |
| threw_1st_hook | 0.11 | 0.016 | -0.035 | -0.015 | -0.028 | -0.013 | 0.067 | |
| threw_2nd_hook | 0.11 | 0.0036 | -0.06 | -0.022 | -0.042 | 0.26 | 0.066 | -0.022 |

# Survivor behaviour

A correlation analysis of the effect of various survivor-related variables on each other.

**Key findings**

- Killers are more likely to be friendly if they have won the match **[hatch_escape]**
- Survivors are more likely end the match early if the killer pursues only them **[tunneled]**

# ANALYSIS

# Game bias: design

How biased or balanced the game is towards survivors & killers is hotly discussed in the community.

The developers claim a 60% bias towards killers [source]. Does this hold true?

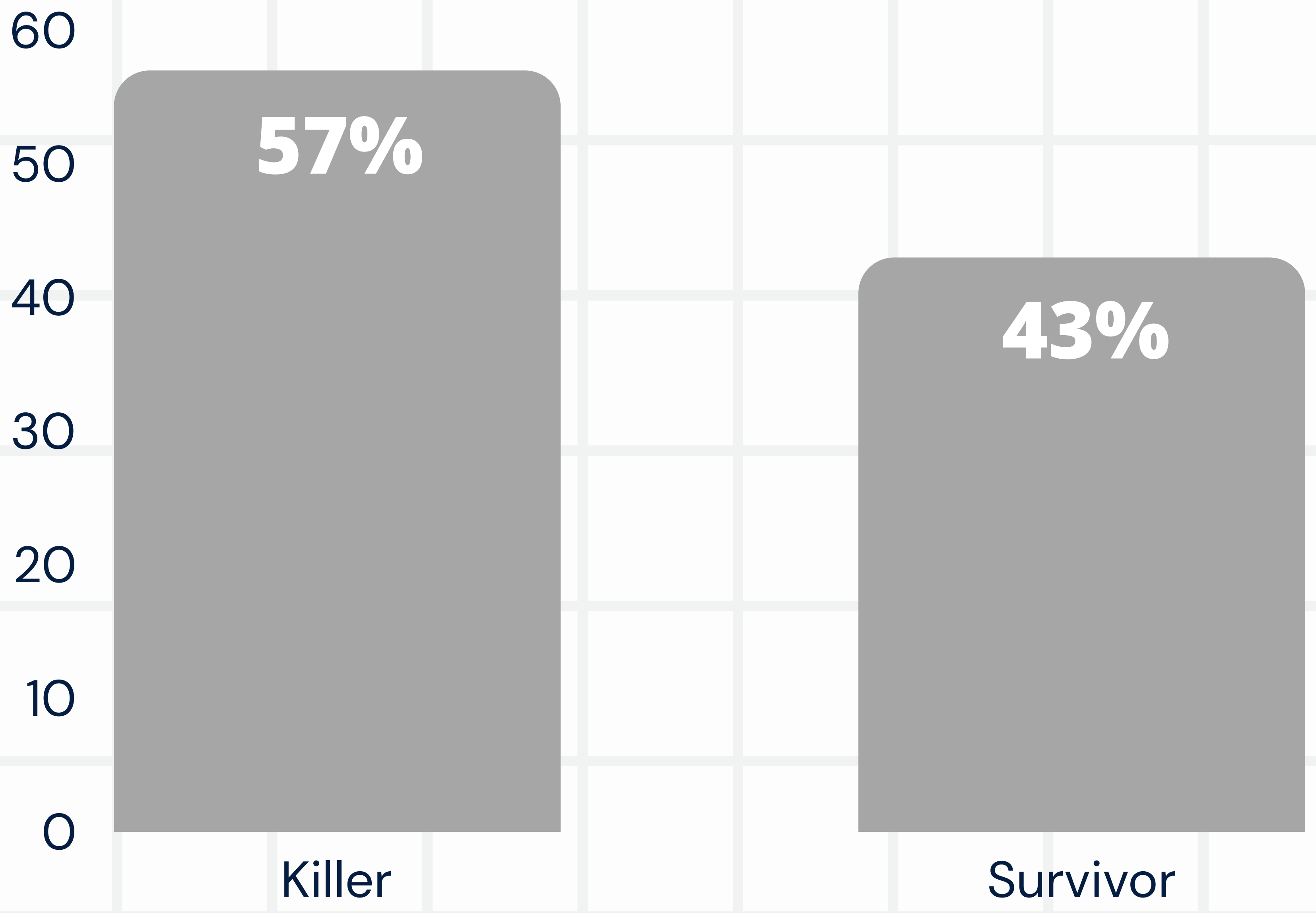
According to the game design itself, this holds true.

**57%**
**Killer bias: map design**

**25**
**Killer-friendly maps**

**19**
**Survivor-friendly maps**

## MAP BIAS vs PLAYER CATEGORY



# Game bias: design

How biased or balanced the game is towards survivors & killers is hotly discussed in the community.

The developers claim a 60% bias towards killers [source]. Does this hold true?

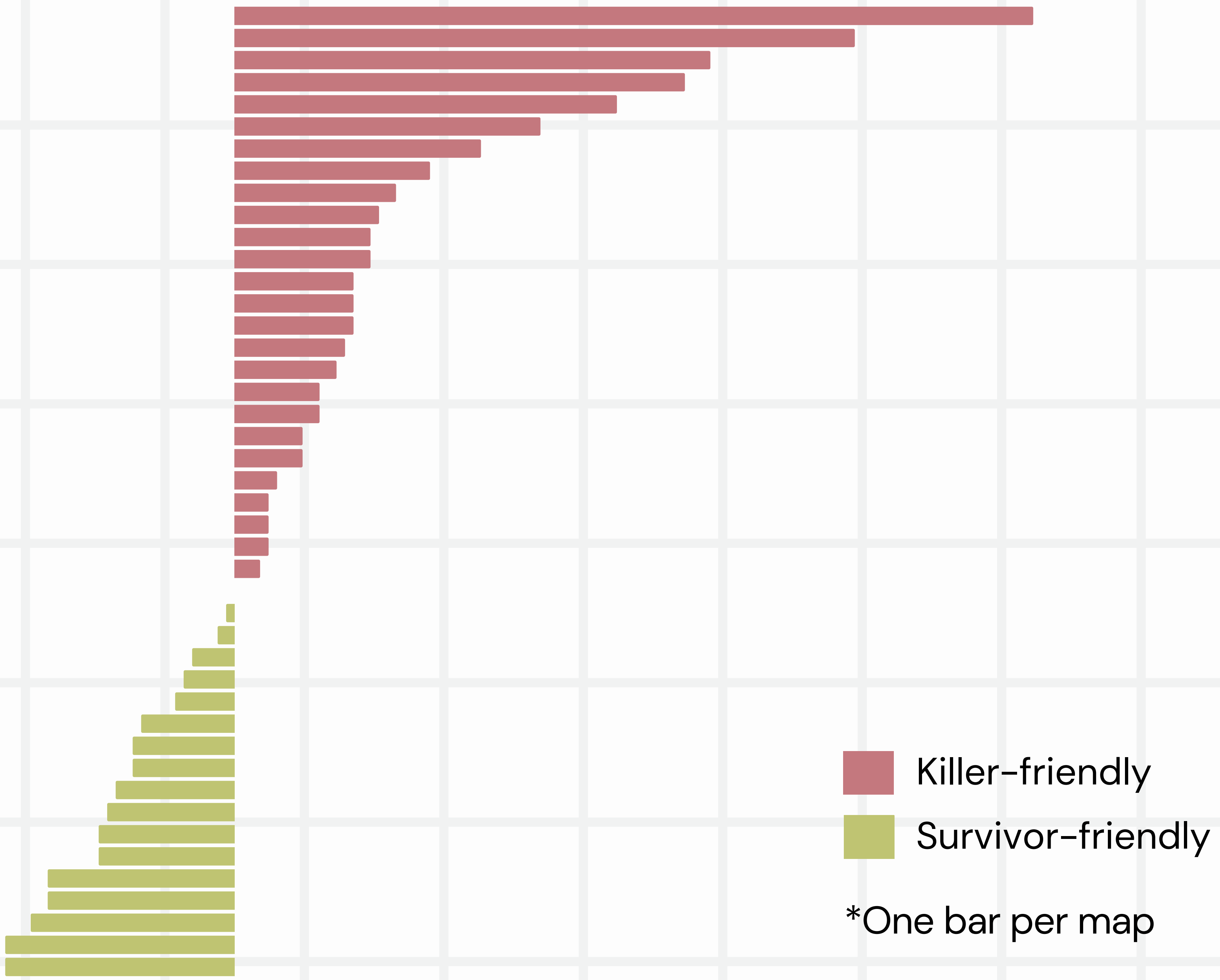
According to the game design itself, this holds true.

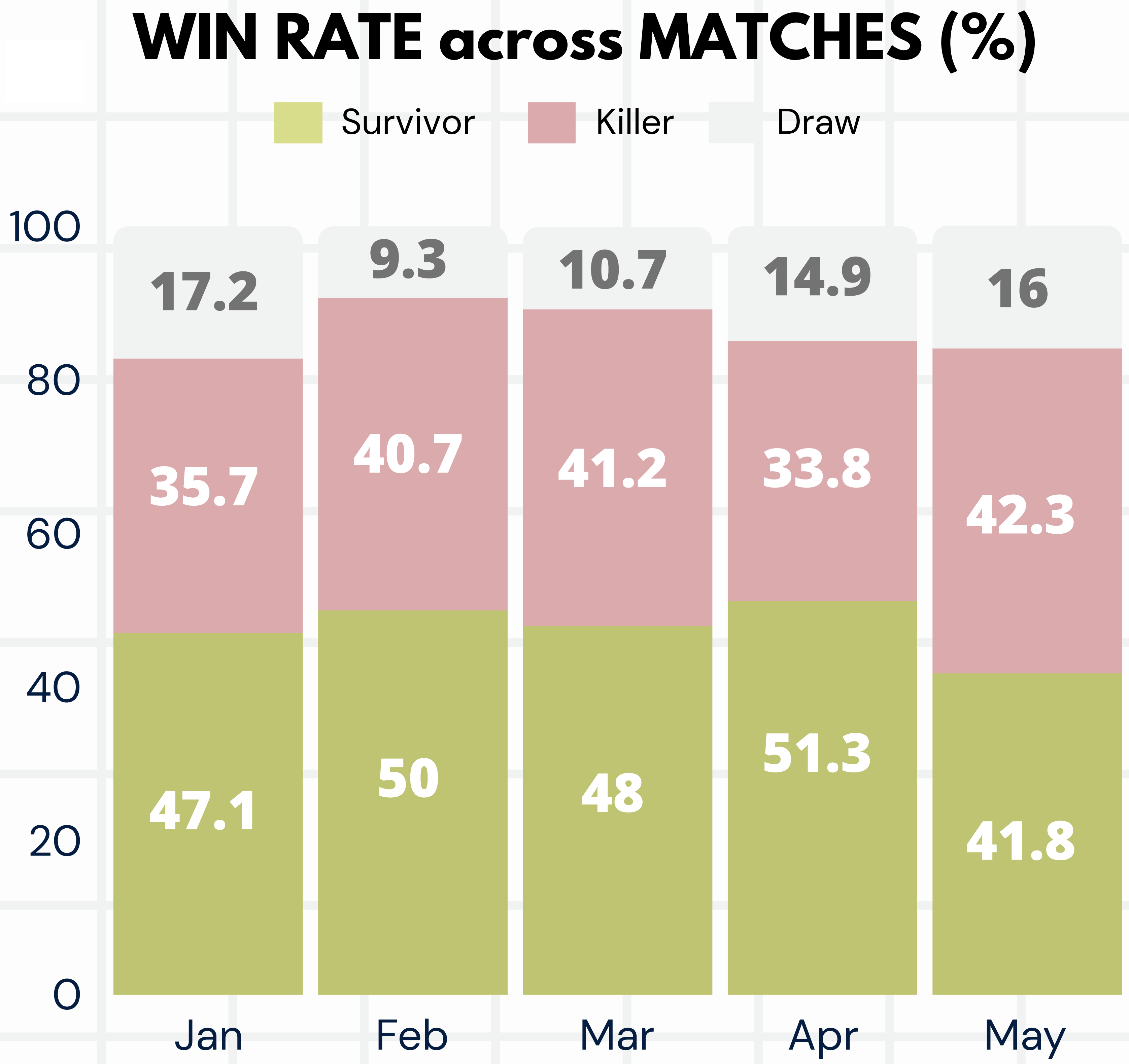
**57%**
**Killer bias: map design**

**25**
**Killer-friendly maps**

**19**
**Survivor-friendly maps**

## MAP BIAS vs PLAYER CATEGORY






# Game bias: matches

Although the game design agrees with a 60% killer bias, as the developers suggest, does this match the reality that players experience?

When we consider the match data [**fact_matches**], it seems not. The game appears to be in the survivors favour (47%), if we use 'win' rate as the metric for bias.

**882**

Matches

**47.4%**
Survivor win rate

**38.8%**
Killer win rate

**13.8%**
Draw rate

**Killer win rate** = (3 | 4) kills out of 4
**Survivor win rate** = 1 kill out of 4
**Draw** = 2 kills out of 4

# Game bias: matches

The developers aim for a 60% kill rate, which they define as number of kills over a given match, with 50% being 2 kills out of 4 per match.

When calculating this metric from over ~500 games, however, the data still suggests that the game is in the survivors' favour, with an average kill rate of 46%

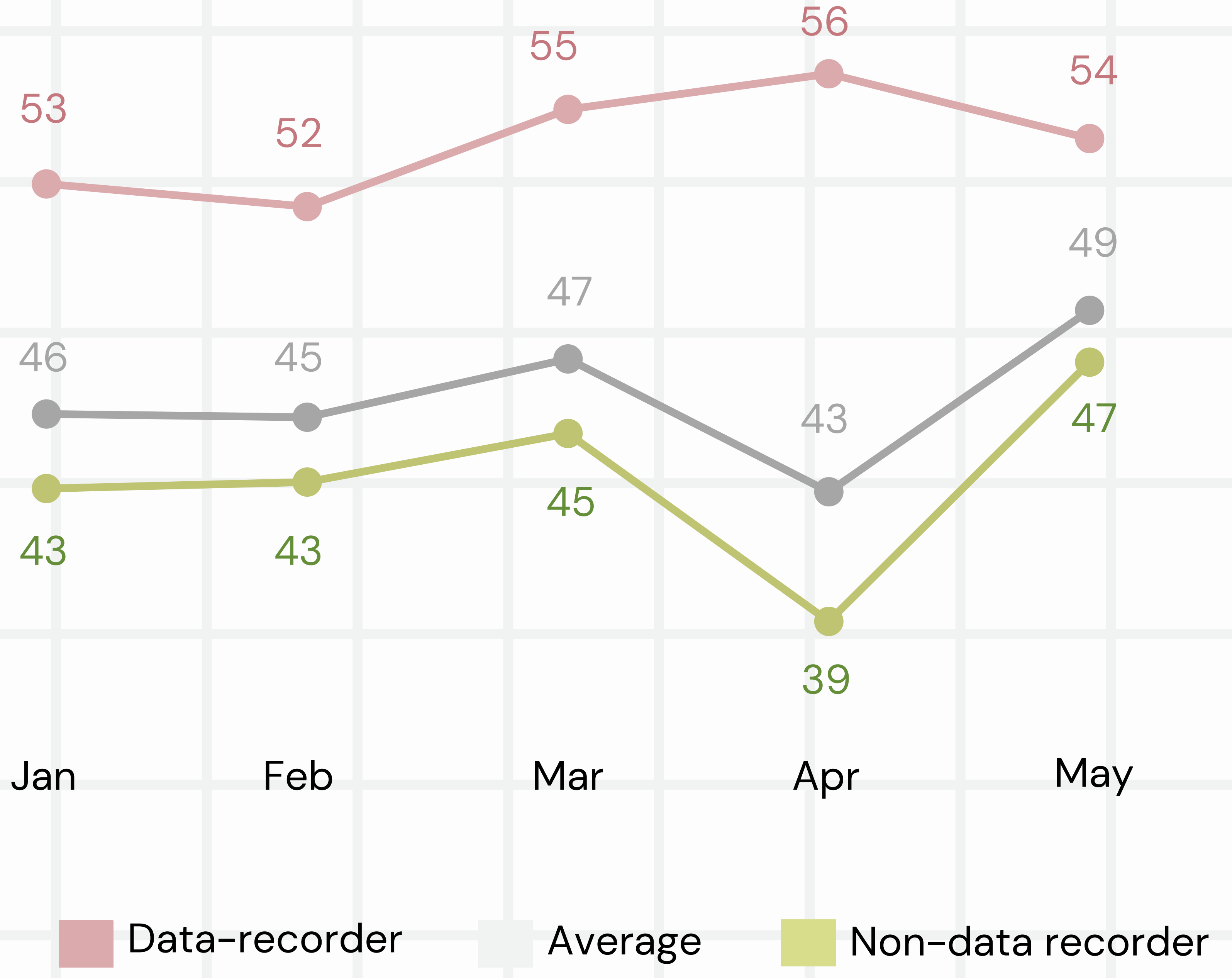
**882**
Matches

**46%**
Average kill rate

**44%**
Average kill rate: non-data recorder

**54%**
Average kill rate: data recorder

## KILL RATE across MATCHES (%)




**Data recorder:** The person who recorded the match data
**Non-data recorder**: The other people who did not record the match data

# ANALYSIS

# Perks

Players can equip up to 4 unique perks which offer them unique bonuses during a match.

Killers and survivors have their own pool of perks to choose from.

**261**
**Total perks**

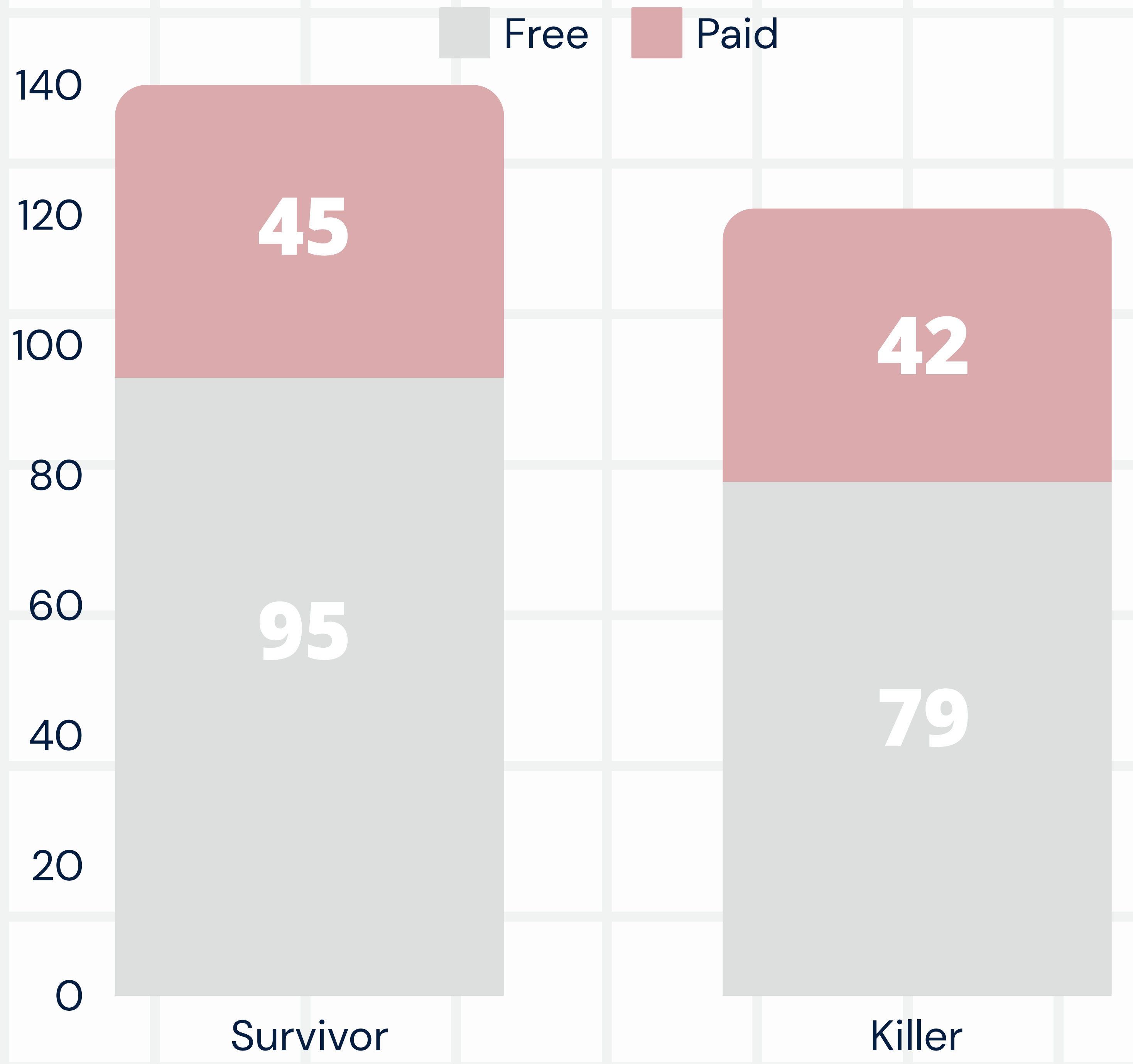
**140**
**Survivor perks**

**121**
**Killer perks**

**174**
**Free perks**

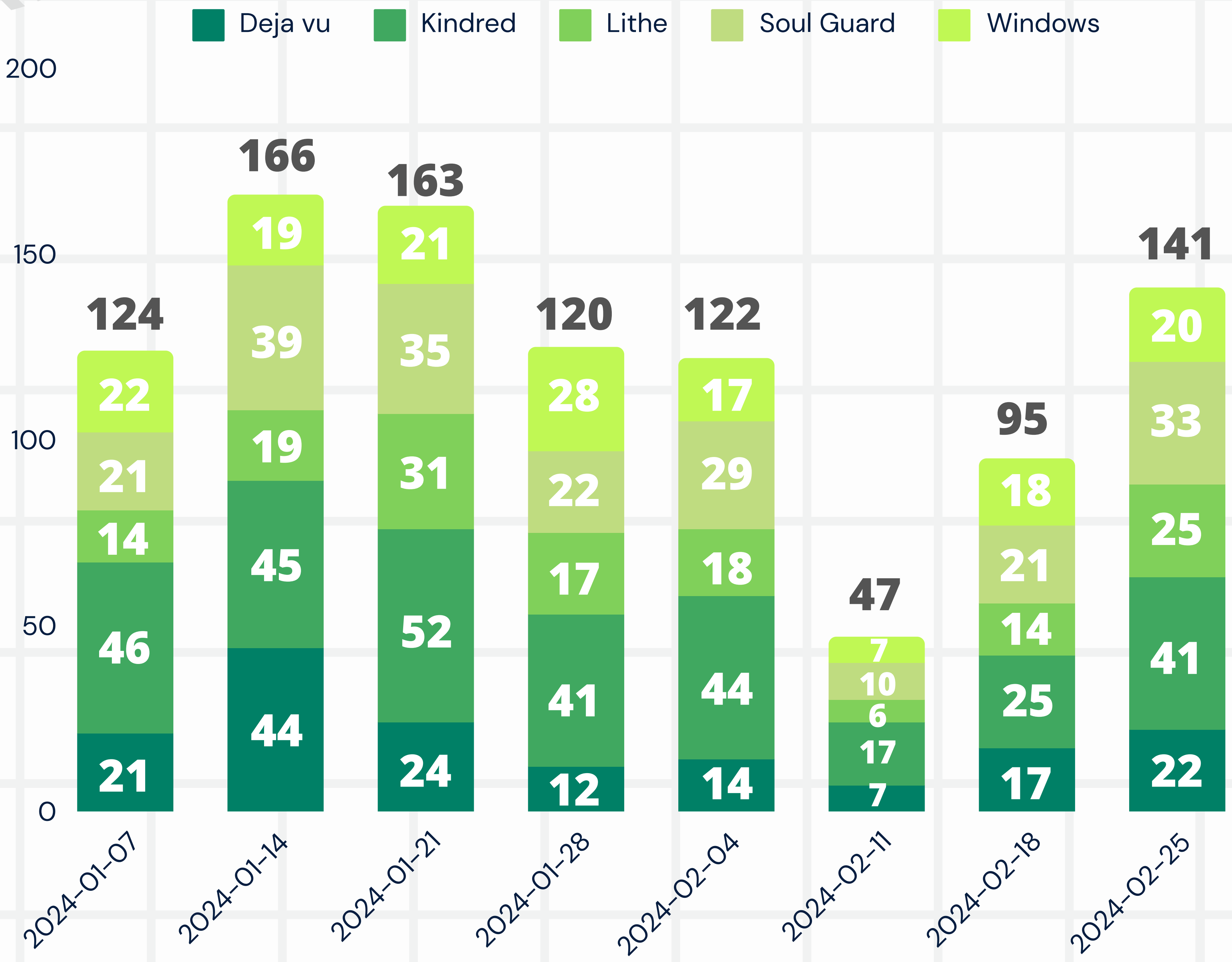
**87**
**Paid perks**

## PERK COUNT vs CHARACTER TYPE & COST




**Free:** No real currency is needed to access
**Non-free:** Real currency is needed to access

# [WEEKLY] TOP 5 SURVIVOR PERKS




# Perks: survivor

Most players tend to gravitate towards the perks they and the community consider the strongest (‘meta perks’).

For survivors, we see perks that are used for:
(i) escaping killers, (ii) locating generators & (iii) finding injured survivors

**261**
**Total perks**

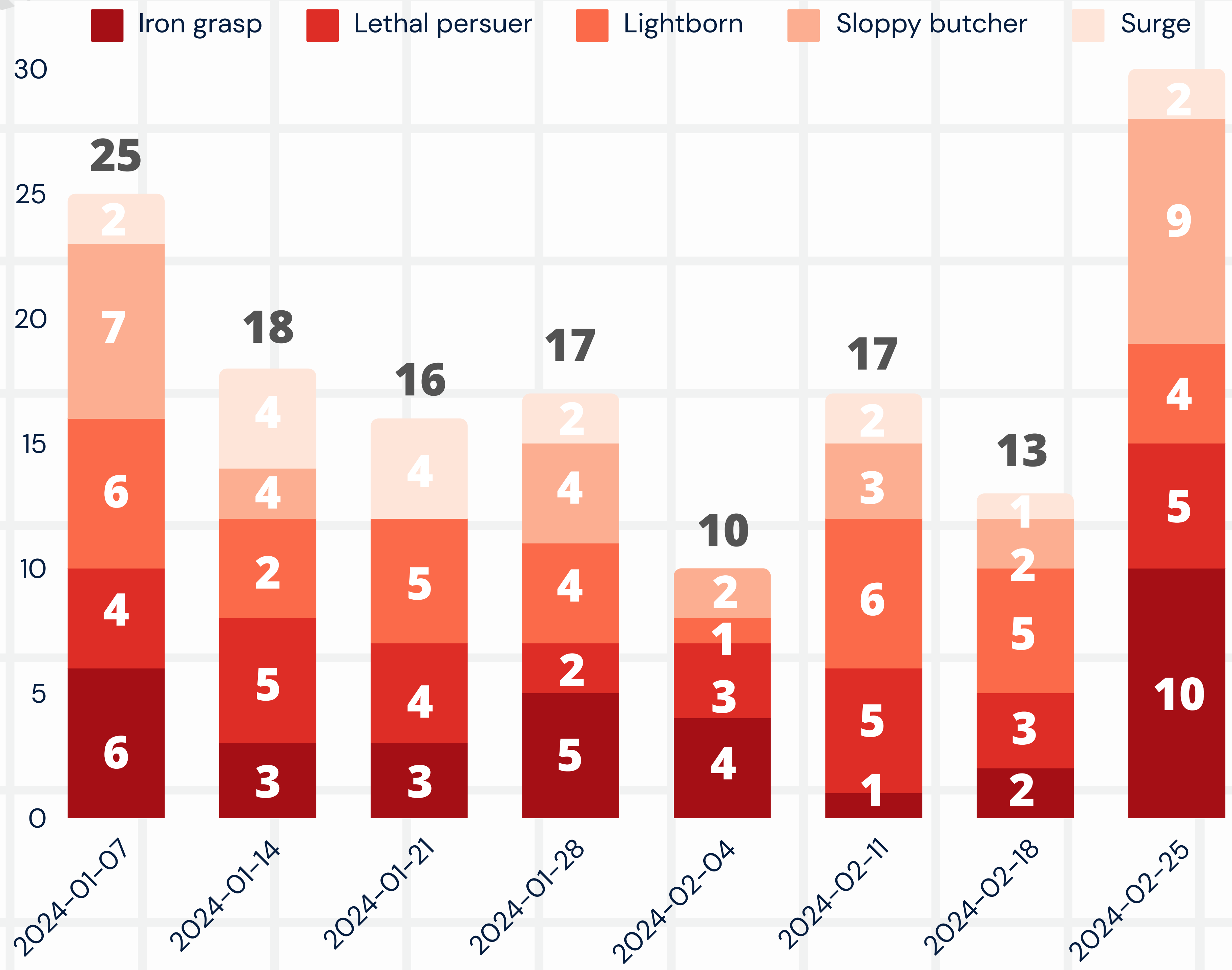
**140**
**Survivor perks**

**121**
**Killer perks**

**15,787,066**
**Possible survivor perk combinations**

**8,790,722**
**Possible killer perk combinations**

# [WEEKLY] TOP 5 KILLER PERKS




# Perks: killer

Most players tend to gravitate towards the perks they and the community consider the strongest ('meta perks').

For killers, the top perks help: (i) find survivors, (ii) reduce healing effectiveness and (iii) stop generator repair progress

**261**
**Total perks**

**140**
**Survivor perks**

**121**
**Killer perks**

**15,787,066**
**Possible survivor perk combinations**

**8,790,722**
**Possible killer perk combinations**

# SURVIVOR PERKS: ASSOCIATION RULES

| Antecedents | Consequents | Antecedent support | Consequent support | Confidence | Lift | Leverage |
|---|---|---|---|---|---|---|
| **None** | **No perk data** | 0.098 | 0.057 | 0.589 | 10.255 | 0.519 |
| **No perk data** | **None** | 0.057 | 0.057 | 1 | 10.255 | 0.519 |
| **Lithe** | **Windows of opportunity** | 0.164 | 0.051 | 0.313 | 1.608 | 0.019 |
| **Windows of opportunity** | **Lithe** | 0.195 | 0.051 | 0.263 | 1.608 | 0.019 |

**TERMINOLOGY**

**Support:** How often both items appear together in the dataset

**Lift:** How much more likely two items occur together compared to if they were unrelated.

**Confidence:** How likely one item appears if the other item appears

**Leverage:** How much more frequently two items occur together than expected by chance

# Perks: survivor

**140**

**Survivor perks**

**15,787,066**

**Possible survivor perk combinations**

Although there are over 15 million possible combinations of perks that survivors can choose from, there are some perks that occur statistically more often together (support >0.05).

Here we can see that two complimentary perks (i) **Lithe** and (ii) **Windows** occur frequently together.

- If **Lithe** is equipped first, there is a 31% chance that **Lithe** will follow (**confidence**).
- If **Windows** is equipped first, **Lithe** follows 26% of the time (**confidence**)

# KILLER PERKS: ASSOCIATION RULES

| Antecedents | Consequents | Antecedent support | Consequent support | Confidence | Lift | Leverage |
|---|---|---|---|---|---|---|
| None | No perk data | 0.272 | 0.085 | 0.314 | 3.677 | 0.062 |
| No perk data | None | 0.085 | 0.272 | 1 | 3.677 | 0.062 |

**TERMINOLOGY**

**Support:** How often both items appear together in the dataset

**Lift:** How much more likely two items occur together compared to if they were unrelated.

**Confidence:** How likely one item appears if the other item appears

**Leverage:** How much more frequently two items occur together than expected by chance

## Perks: killer

**121**

**Killer perks**

**8,790,722**

**Possible killer perk combinations**

In terms of the killer association rules, we can see that there aren't any perks that are statistically likely to occur with one another (support > 0.05).

This is likely because each killer has a different play style:

- 1 unique power per killer
- **Ranged** attack vs. **melee** attack
- **Stealth** focus vs. **non-stealth** focus

This, in turn, leads to killer perks being more differential in nature, with certain perks benefiting some killers more than others.

# DATA LIMITATIONS

# DATA LIMITATIONS

Here are some limitations of the data collected, with potential solutions provided, where relevant.

**01.** Data is currently scraped. Any changes to the website would make the code unusable

**Solution:** Use API

**02.** The website data resets monthly, leading to data loss with current pipeline

**Solution:** Append monthly data

**03.** The ratings of maps & killer effectiveness are user-generated & are subject to bias and rating manipulation

**04.** Fact_matches data is manually inputted and uploaded via spreadsheet.

# FURTHER IMPROVEMENT

# FURTHER IMPROVEMENT

Due to the limited scope of the current analysis, here would be some areas of improvement, that could be worked on, if more time & resources were available

**01.** Perk categories are currently manually assigned. It would be worthwhile to use machine learning to auto-generate categories from their descriptions.

**02.** Character downloadable content data is manually inputted. It could be automated.

**03.** Cleaning of fact_matches was done in a spreadsheet. Could automate as part of the pipeline.

**04.** Swap over from scraped data to use the website API, for a more stable & scaleable data ingression

**05.** Analyse the dim_addons_killer table. It was scraped from the website but wasn't used in the current analysis.

# REFERENCES

# REFERENCES


- **CrypticGirl (2024).** '*DBD Match Data Spreadsheet.*' Behaviour Interactive Inc. Forum. https://forums.bhvr.com/dead-by-daylight/discussion/401797/dbd-match-data-spreadsheet. Accessed: 02 August, 2024.
- **Dzzplayz. (2024).** '*I've categorized all of the perks, because I felt like it.*' Reddit. https://www.reddit.com/r/deadbydaylight/comments/1awkgnt/ive_categorized_all_of_the_perks_because_i_felt/. Accessed: 02 August, 2024.
- **GeeksforGeeks. (2021).** '*Scrape Tables From any website using Python.*' https://www.geeksforgeeks.org/scrape-tables-from-any-website-using-python/. Accessed: 02 August, 2024.

# REFERENCES


- **Peanits. (2024).** '*Developer Update | Stats!*' Behaviour Interactive Inc. https://forums.bhvr.com/dead-by-daylight/kb/articles/433-developer-update-stats. Accessed: 02 August, 2024.
- **Reep, D. (2024).** '*Dead by Daylight Statistics Website.*' https://dennisreep.nl/dbd/. Accessed: 02 August, 2024.
- **Wikipedia. (2024).** '*Dead by Daylight.*' https://en.wikipedia.org/wiki/Dead_by_Daylight. Accessed: 02 August, 2024.