review_analysis_ark

August 6, 2024

[3]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
from wordcloud import WordCloud
import numpy as np
from IPython.display import display, Markdown, Latex
```

[4]:
```
# Define the game name variable
GAME_NAME = "ARK: Survival Evolved"  # Change this to the name of the game␣
 ↪you're analyzing
```

[5]:
```
# Ensure plots are displayed inline
%matplotlib inline
```

[6]:
```
# Set default figure size for better PDF output
plt.rcParams['figure.figsize'] = [10, 6]
```

[7]:
```
# Load the data
df = pd.read_csv('../Review CSVs/reviews_346110.csv')
```

[8]:
```
# Convert timestamp_created to datetime
df['timestamp_created'] = pd.to_datetime(df['timestamp_created'])
```

[9]:
```
# Display basic information about the loaded data
print(f"Data loaded successfully for {GAME_NAME}")
print(f"Number of reviews: {len(df)}")
print(f"Date range: from {df['timestamp_created'].min()} to␣
 ↪{df['timestamp_created'].max()}")
```

```
Data loaded successfully for ARK: Survival Evolved
Number of reviews: 308017
Date range: from 2015-06-02 22:29:25 to 2024-07-23 22:03:59
```

[10]:
```
# Filter data from January 2019 to June 2024
start_date = '2019-01-01'
end_date = '2024-06-30'
```

```
df_filtered = df[(df['timestamp_created'] >= start_date) &␣
 ↪(df['timestamp_created'] <= end_date)]

print(f"\nNumber of reviews after filtering (Jan 2019 - Jun 2024):␣
 ↪{len(df_filtered)}")
```

```
Number of reviews after filtering (Jan 2019 - Jun 2024): 210699
```

[11]:
```
# Create a table of contents
display(Markdown(f"# Steam Review Analysis: {GAME_NAME}"))
display(Markdown("## Table of Contents"))
display(Markdown("""
1. [Monthly Review Trends](#monthly-review-trends)
2. [Sentiment Analysis Over Time](#sentiment-analysis-over-time)
3. [Playtime vs Review Sentiment](#playtime-vs-review-sentiment)
4. [Review Length Analysis](#review-length-analysis)
5. [Early Access Impact](#early-access-impact)
6. [Language Distribution](#language-distribution)
7. [Player Experience Level](#player-experience-level)
8. [Review Helpfulness Over Time](#review-helpfulness-over-time)
9. [Seasonal Trends](#seasonal-trends)
10. [Word Frequency Analysis](#word-frequency-analysis)
"""))
```

1 Steam Review Analysis: ARK: Survival Evolved

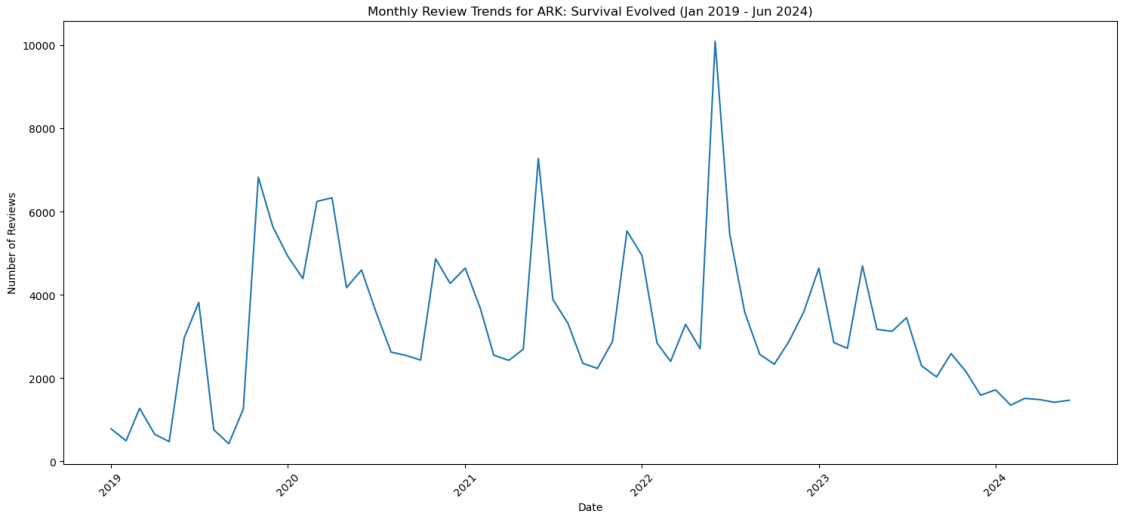
1.1 Table of Contents

[12]:
```
# Function to create section headers with anchors for the table of contents
def section_header(title, anchor):
    display(Markdown(f"<a id='{anchor}'></a>"))
    display(Markdown(f"## {title}"))
```

[13]:
```
# 1. Monthly Review Trends
section_header("Monthly Review Trends", "monthly-review-trends")
```

```
"""
This visualization shows the ebb and flow of review activity over time.

- Spikes might indicate major updates, sales, or viral moments.
- Troughs could suggest periods of lower player interest or game issues.
- The overall trend can reveal if the game is gaining or losing momentum.

"""

monthly_reviews = df_filtered.groupby(df_filtered['timestamp_created'].dt.
 ↪to_period("M")).size().reset_index(name='count')
monthly_reviews['timestamp_created'] = monthly_reviews['timestamp_created'].dt.
 ↪to_timestamp()

plt.figure(figsize=(15, 7))
plt.plot(monthly_reviews['timestamp_created'], monthly_reviews['count'])
plt.title(f"Monthly Review Trends for {GAME_NAME} (Jan 2019 - Jun 2024)")
plt.xlabel('Date')
plt.ylabel('Number of Reviews')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

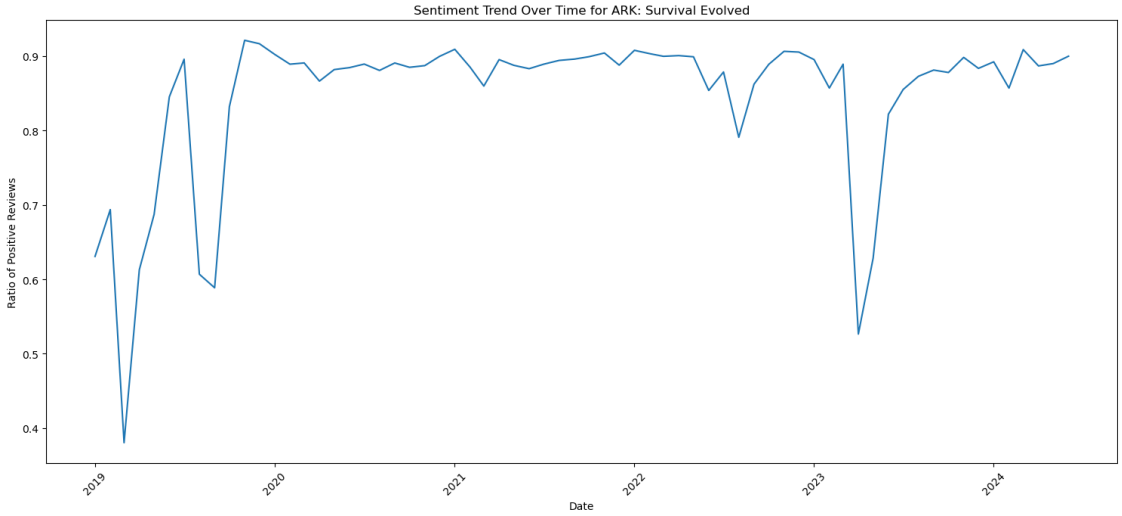
1.2 Monthly Review Trends



[14]:
```
# 2. Sentiment Analysis Over Time
section_header("Sentiment Analysis Over Time", "sentiment-analysis-over-time")

"""
```

```
Here we're tracking the game's approval rating over its lifetime.

- Upward trends suggest improvements or positive updates.
- Downward trends might indicate issues or unpopular changes.
- Stability could mean consistent quality or stagnation.

"""

df_filtered = df_filtered.copy()  # Create an explicit copy
df_filtered.loc[:, 'month'] = df_filtered['timestamp_created'].dt.to_period('M')
sentiment_over_time = df_filtered.groupby('month').agg({'voted_up': 'mean'}).
 ↪reset_index()
sentiment_over_time['month'] = sentiment_over_time['month'].dt.to_timestamp()

plt.figure(figsize=(15, 7))
plt.plot(sentiment_over_time['month'], sentiment_over_time['voted_up'])
plt.title(f"Sentiment Trend Over Time for {GAME_NAME}")
plt.xlabel('Date')
plt.ylabel('Ratio of Positive Reviews')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

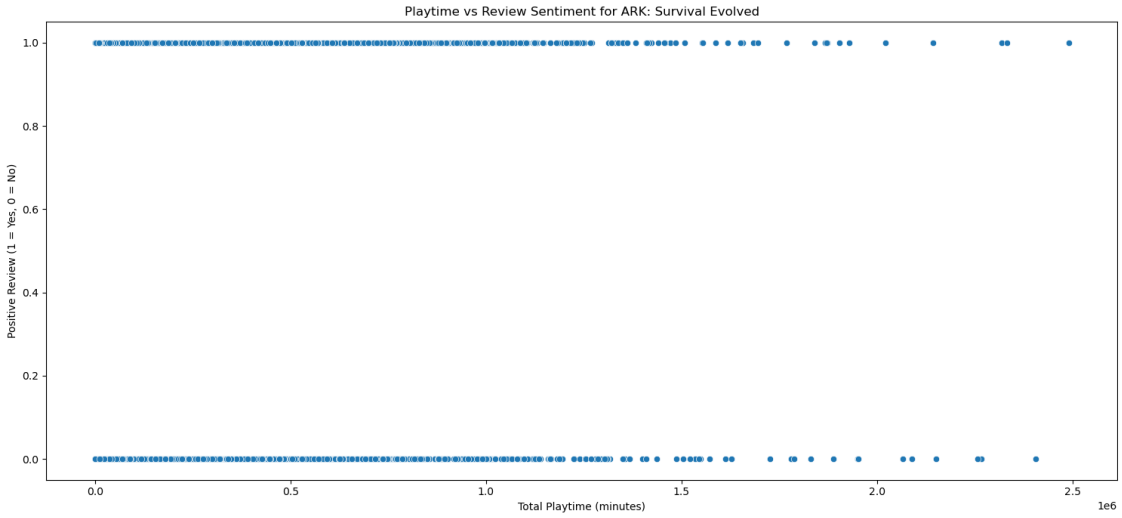
1.3 Sentiment Analysis Over Time



[15]:
```
# 3. Playtime vs Review Sentiment
section_header("Playtime vs Review Sentiment", "playtime-vs-review-sentiment")

"""
```

```
This shows the relationship between how long people play and how they feel␣
 ↪about it.

- Clusters in the top-right are die-hard fans: long playtime and positive␣
 ↪reviews.
- Bottom-left clusters might be disappointed players or those who couldn't get␣
 ↪into the game.
- Outliers tell interesting stories: loved it but barely played, or played␣
 ↪forever but didn't like it?

"""

plt.figure(figsize=(15, 7))
sns.scatterplot(data=df_filtered, x='author_playtime_forever', y='voted_up')
plt.title(f"Playtime vs Review Sentiment for {GAME_NAME}")
plt.xlabel('Total Playtime (minutes)')
plt.ylabel('Positive Review (1 = Yes, 0 = No)')
plt.tight_layout()
plt.show()
```

1.4 Playtime vs Review Sentiment



[16]:
```
# 4. Review Length Analysis
section_header("Review Length Analysis", "review-length-analysis")


"""
This histogram shows how verbose (or concise) the reviewers tend to be.
```

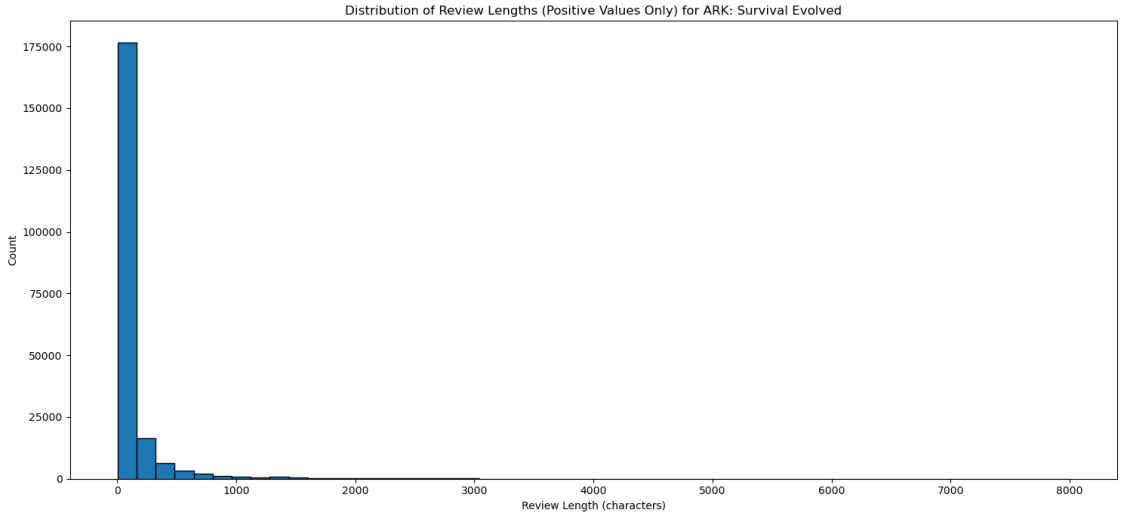
```
- A peak at lower lengths might indicate quick, emotional responses.
- Longer reviews could suggest more thoughtful, detailed feedback.
- The shape of the distribution can tell you about the reviewers' habits.

"""

df_filtered.loc[:, 'review_length'] = df_filtered['review'].str.len()

# Data validation
print("Review Length Statistics:")
print(df_filtered['review_length'].describe())

print("\nUnique review lengths:")
print(df_filtered['review_length'].value_counts().sort_index())

# Check for non-positive values
non_positive = df_filtered[df_filtered['review_length'] <= 0]
print(f"\nNumber of non-positive review lengths: {len(non_positive)}")

if len(non_positive) > 0:
    print("Sample of reviews with non-positive lengths:")
    print(non_positive[['review', 'review_length']].head())

# Proceed with visualization only if we have positive values
positive_lengths = df_filtered[df_filtered['review_length'] >␣
 ↪0]['review_length']

if len(positive_lengths) > 0:
    # Histogram
    plt.figure(figsize=(15, 7))
    plt.hist(positive_lengths, bins=50, edgecolor='black')
    plt.title(f"Distribution of Review Lengths (Positive Values Only) for␣
 ↪{GAME_NAME}")
    plt.xlabel('Review Length (characters)')
    plt.ylabel('Count')
    plt.tight_layout()
    plt.show()

    # Box Plot
    plt.figure(figsize=(10, 6))
    bp = plt.boxplot(positive_lengths, vert=False, whis=[5, 95])
    plt.title(f"Box Plot of Review Lengths (5th to 95th percentile, Positive␣
 ↪Values Only) for {GAME_NAME}")
    plt.xlabel('Review Length (characters)')
    plt.xscale('log')  # Use log scale for x-axis

    # Add labels for quartiles
```

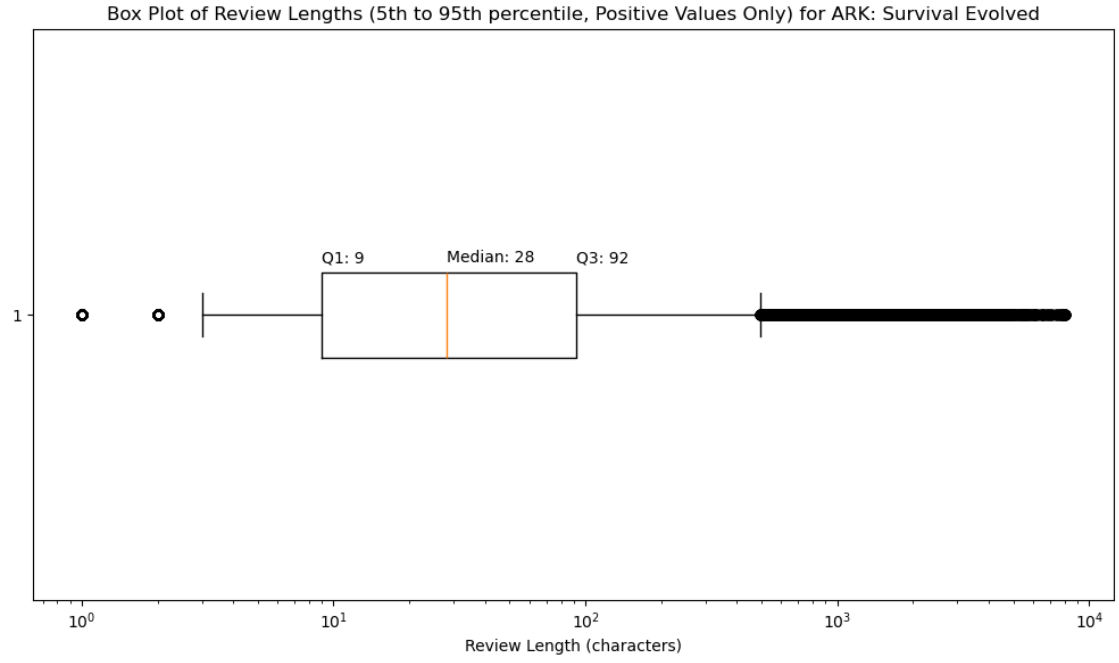
```
    quartiles = positive_lengths.quantile([0.25, 0.5, 0.75])
    for i, q in enumerate(['Q1', 'Median', 'Q3']):
        plt.text(quartiles.iloc[i], 1.1, f'{q}: {quartiles.iloc[i]:.0f}',
                 verticalalignment='center')

    plt.tight_layout()
    plt.show()

    # Print additional percentiles for context
    percentiles = positive_lengths.quantile([0.05, 0.25, 0.5, 0.75, 0.95])
    print("\nReview Length Percentiles (Positive Values Only):")
    for p, v in percentiles.items():
        print(f"{p*100:.0f}th percentile: {v:.0f} characters")

    # Correlation between review length and sentiment
    correlation = positive_lengths.corr(df_filtered.loc[positive_lengths.index,␣
 ↪'voted_up'])
    print(f"\nCorrelation between review length and positive sentiment:␣
 ↪{correlation:.2f}")
else:
    print("No positive review lengths found. Cannot create visualizations.")

# If we have non-positive values, let's investigate further
if len(non_positive) > 0:
    print("\nAnalysis of non-positive review lengths:")
    print(non_positive['review_length'].value_counts().sort_index())
    print("\nSample of reviews with non-positive lengths:")
    print(non_positive[['review', 'review_length']].head())
```

1.5 Review Length Analysis

```
Review Length Statistics:
count    209859.000000
mean        118.593365
std         342.175274
min           1.000000
25%           9.000000
50%          28.000000
75%          92.000000
max        8000.000000
Name: review_length, dtype: float64

Unique review lengths:
review_length
1.0        4844
2.0        3963
3.0        9120
```

```
4.0       10821
5.0        4508
          …
7994.0        3
7997.0        6
7998.0        1
7999.0        4
8000.0       11
Name: count, Length: 2815, dtype: int64

Number of non-positive review lengths: 0
```

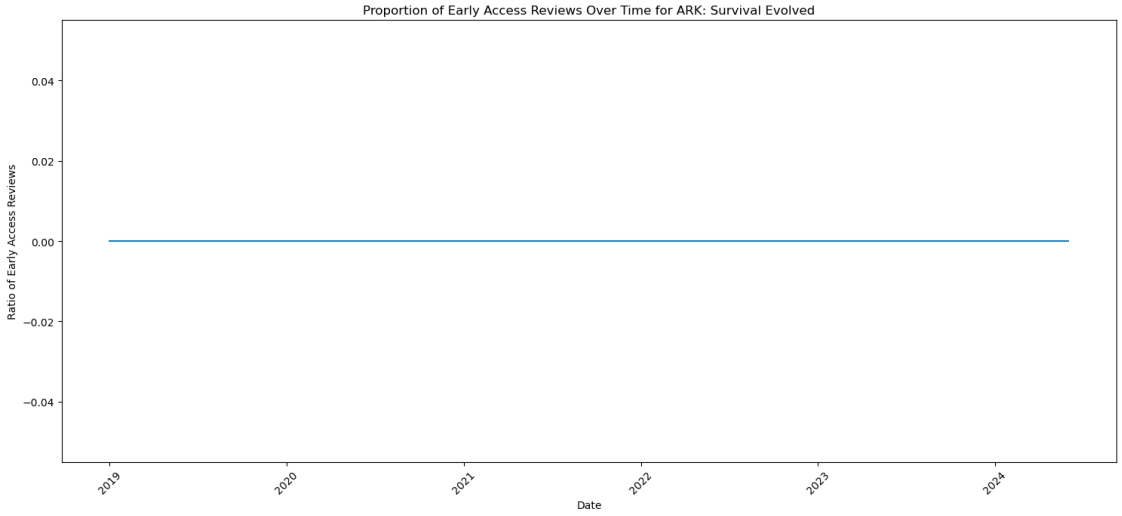




```
Review Length Percentiles (Positive Values Only):
5th percentile: 3 characters
25th percentile: 9 characters
50th percentile: 28 characters
75th percentile: 92 characters
95th percentile: 495 characters

Correlation between review length and positive sentiment: -0.19
```

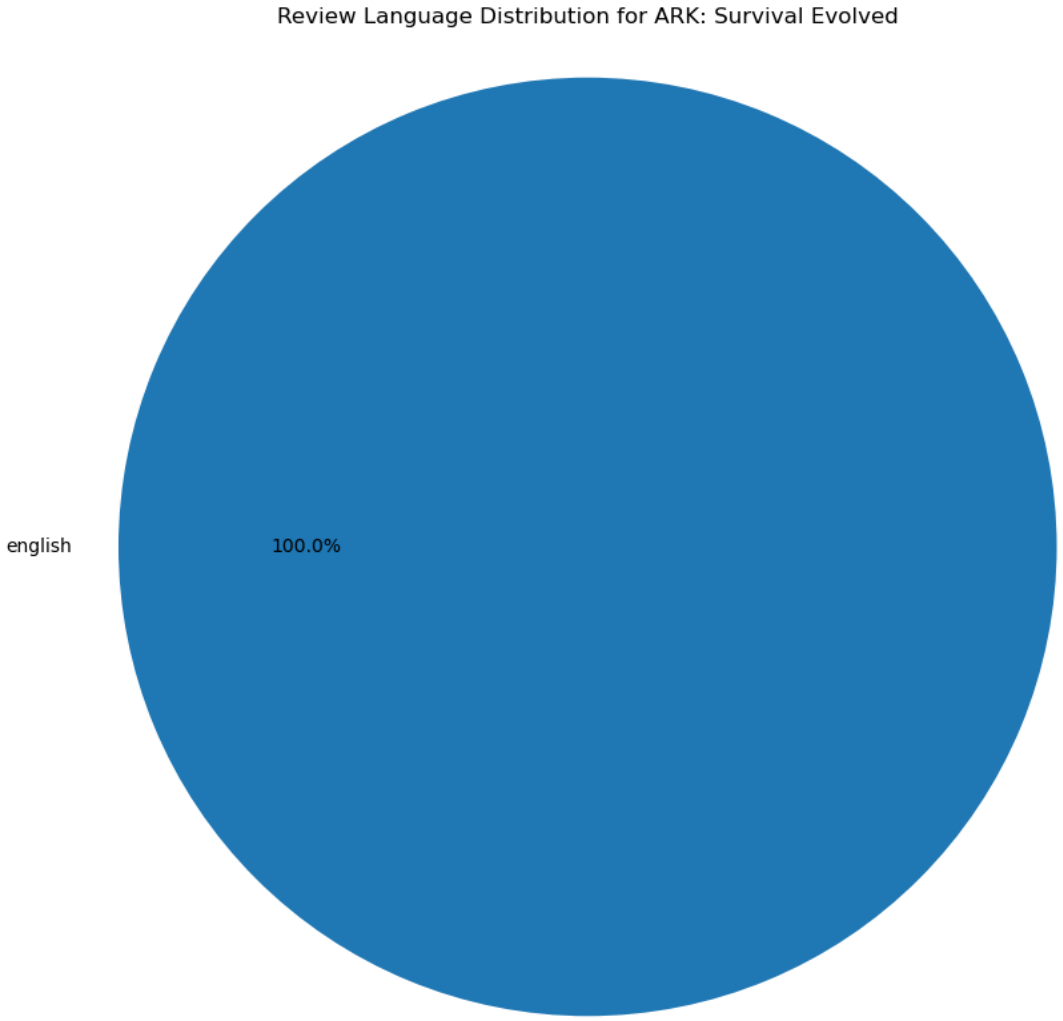
[17]:
```python
# 5. Early Access Impact
section_header("Early Access Impact", "early-access-impact")


"""
This chart shows how the Early Access phase influenced player opinions over␣
 ↪time.

- High early proportions show initial early adopter enthusiasm.
- The trend downwards indicates the transition to full release.
- Post-release blips might suggest nostalgia or comparisons to the early days.

"""

early_access_trend = df_filtered.groupby('month').
 ↪agg({'written_during_early_access': 'mean'}).reset_index()
early_access_trend['month'] = early_access_trend['month'].dt.to_timestamp()

plt.figure(figsize=(15, 7))
plt.plot(early_access_trend['month'],␣
 ↪early_access_trend['written_during_early_access'])
plt.title(f"Proportion of Early Access Reviews Over Time for {GAME_NAME}")
plt.xlabel('Date')
plt.ylabel('Ratio of Early Access Reviews')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

1.6 Early Access Impact



[18]:
```
# 6. Language Distribution
section_header("Language Distribution", "language-distribution")


"""
This pie chart shows the global linguistic reach and cultural impact of the␣
 ↪game.

- Dominant languages indicate the primary markets.
- Diverse small slices suggest a global cult following.
- Unexpected languages might reveal surprising new markets.

"""

language_dist = df_filtered['language'].value_counts()
plt.figure(figsize=(10, 10))
plt.pie(language_dist.values, labels=language_dist.index, autopct='%1.1f%%')
plt.title(f"Review Language Distribution for {GAME_NAME}")
plt.axis('equal')
plt.show()
```

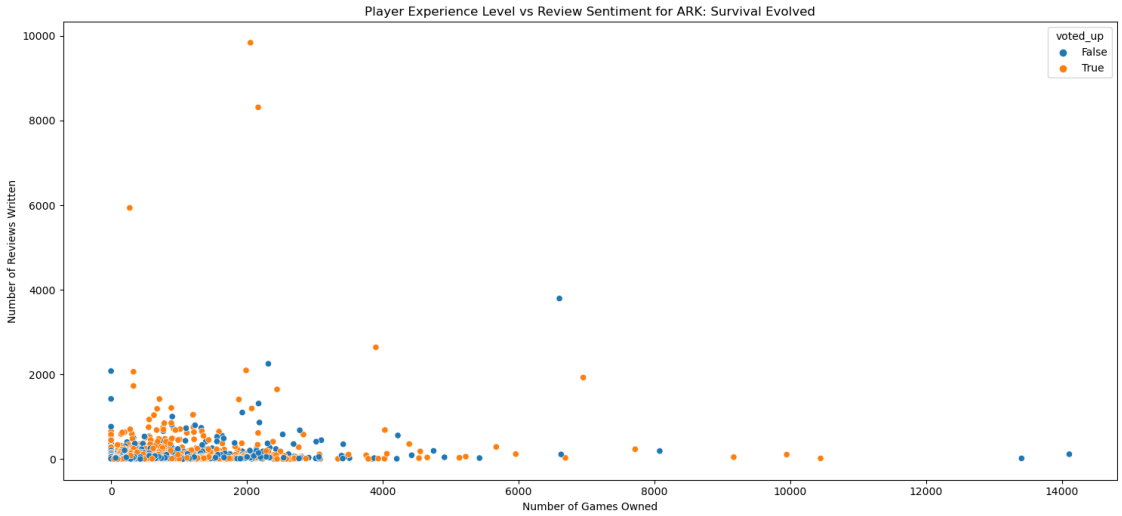
1.7 Language Distribution



[19]:
```
# 7. Player Experience Level
section_header("Player Experience Level", "player-experience-level")


"""
This scatter plot is like a gamer demographic survey for the reviewers.
It shows how experienced Steam users react to the game.

- Clusters can reveal the core audience: newbies, veterans, or a mix?
- The color distribution shows how different player types rate the game.
- Outliers might be influential reviewers or unique player experiences.
```

```
It's a way to see if the game is a hit with the Steam elite or a gateway game␣
 ↪for newcomers!

"""

plt.figure(figsize=(15, 7))
sns.scatterplot(data=df_filtered, x='author_num_games_owned',␣
 ↪y='author_num_reviews', hue='voted_up')
plt.title(f"Player Experience Level vs Review Sentiment for {GAME_NAME}")
plt.xlabel('Number of Games Owned')
plt.ylabel('Number of Reviews Written')
plt.tight_layout()
plt.show()
```

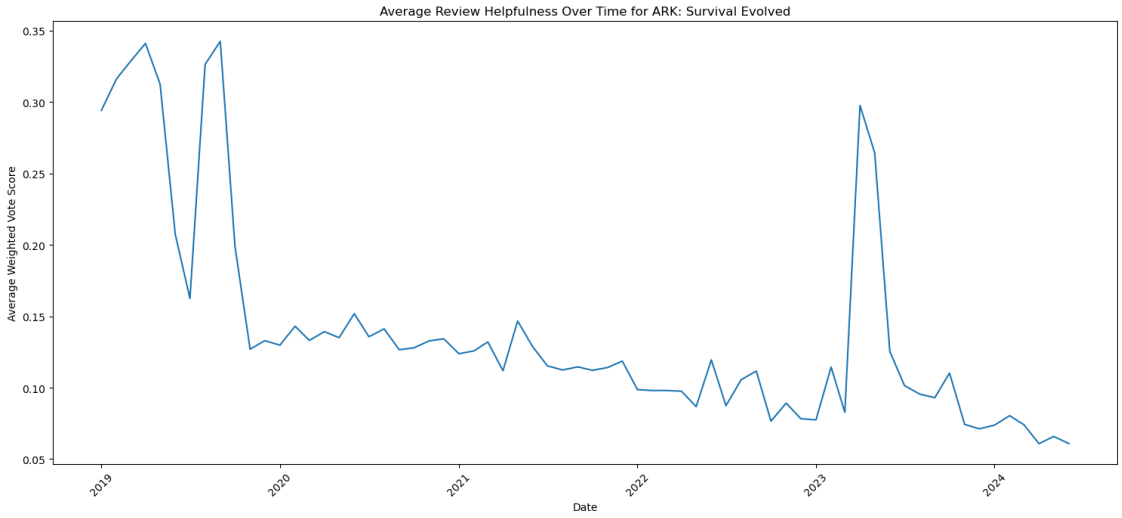
1.8 Player Experience Level



[20]:
```
# 8. Review Helpfulness Over Time
section_header("Review Helpfulness Over Time", "review-helpfulness-over-time")


"""
This trend line shows how useful other players find the reviews.

- Upward trends suggest more insightful or balanced reviews over time.
- Downward trends might indicate controversial periods or review bombing.
- Stability could mean consistent community engagement.

"""
```

```
helpfulness_over_time = df_filtered.groupby('month').agg({'weighted_vote_score':
  ↪ 'mean'}).reset_index()
helpfulness_over_time['month'] = helpfulness_over_time['month'].dt.
  ↪to_timestamp()

plt.figure(figsize=(15, 7))
plt.plot(helpfulness_over_time['month'],␣
  ↪helpfulness_over_time['weighted_vote_score'])
plt.title(f"Average Review Helpfulness Over Time for {GAME_NAME}")
plt.xlabel('Date')
plt.ylabel('Average Weighted Vote Score')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

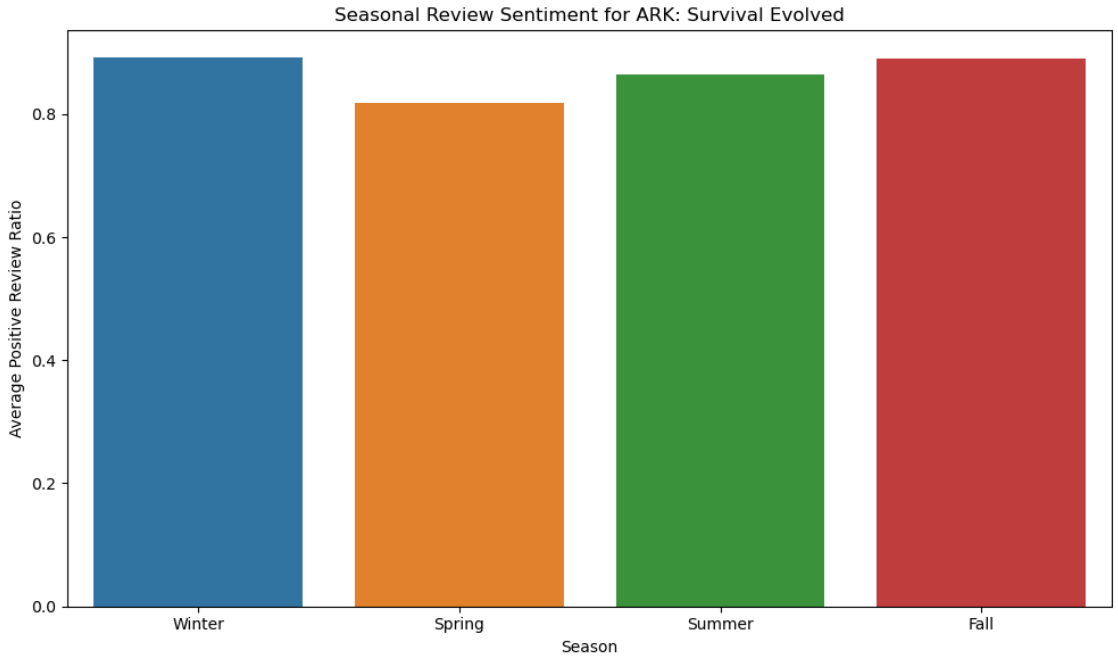
1.9 Review Helpfulness Over Time



[21]:
```
# 9. Seasonal Trends
section_header("Seasonal Trends", "seasonal-trends")


"""
This bar chart shows how the time of year affects player sentiment.

- Higher bars indicate seasons when players tend to enjoy the game more.
- Lower bars might suggest seasonal challenges or competition.
- Patterns could reveal optimal times for updates or promotions.

"""
```

```
df_filtered.loc[:, 'season'] = df_filtered['timestamp_created'].dt.month.map({1:
 ↪'Winter', 2:'Winter', 3:'Spring',
                                                                          4:
 ↪'Spring', 5:'Spring', 6:'Summer',
                                                                          7:
 ↪'Summer', 8:'Summer', 9:'Fall',
                                                                          ␣
 ↪10:'Fall', 11:'Fall', 12:'Winter'})
seasonal_sentiment = df_filtered.groupby('season')['voted_up'].mean().
 ↪reset_index()

plt.figure(figsize=(10, 6))
sns.barplot(x='season', y='voted_up', data=seasonal_sentiment, order=['Winter',␣
 ↪'Spring', 'Summer', 'Fall'])
plt.title(f"Seasonal Review Sentiment for {GAME_NAME}")
plt.xlabel('Season')
plt.ylabel('Average Positive Review Ratio')
plt.tight_layout()
plt.show()
```

1.10 Seasonal Trends



[22]:
```
# 10. Word Frequency Analysis (Word Cloud)
section_header("Word Frequency Analysis", "word-frequency-analysis")
```

```
"""
These word clouds are like thought bubbles floating above the positive and␣
 ↪negative reviewers.
They visually represent the most common terms used in reviews.

- Larger words are more frequently mentioned and potentially more important.
- Positive cloud might reveal beloved features or emotions.
- Negative cloud could highlight pain points or areas for improvement.

"""


def clean_reviews(reviews):
    return ' '.join([str(review) for review in reviews if isinstance(review,␣
 ↪str)])

positive_reviews = clean_reviews(df_filtered[df_filtered['voted_up'] ==␣
 ↪True]['review'])
negative_reviews = clean_reviews(df_filtered[df_filtered['voted_up'] ==␣
 ↪False]['review'])

def generate_wordcloud(text, title):
    if not text:
        print(f"No valid text for {title}. Skipping word cloud generation.")
        return

    wordcloud = WordCloud(width=800, height=400, background_color='white').
 ↪generate(text)
    plt.figure(figsize=(10, 5))
    plt.imshow(wordcloud, interpolation='bilinear')
    plt.axis('off')
    plt.title(title)
    plt.tight_layout(pad=0)
    plt.show()

# Print some statistics about the reviews
print(f"Number of positive reviews: {len(df_filtered[df_filtered['voted_up'] ==␣
 ↪True])}")
print(f"Number of negative reviews: {len(df_filtered[df_filtered['voted_up'] ==␣
 ↪False])}")
print(f"Number of words in positive reviews: {len(positive_reviews.split())}")
print(f"Number of words in negative reviews: {len(negative_reviews.split())}")
```

```
generate_wordcloud(positive_reviews, f'Word Cloud of Positive Reviews for␣
  ↪{GAME_NAME}')
generate_wordcloud(negative_reviews, f'Word Cloud of Negative Reviews for␣
  ↪{GAME_NAME}')

# Optional: Display some sample reviews
print("\nSample positive review:")
print(df_filtered[df_filtered['voted_up'] == True]['review'].iloc[0])

print("\nSample negative review:")
print(df_filtered[df_filtered['voted_up'] == False]['review'].iloc[0])
```

1.11 Word Frequency Analysis

```
Number of positive reviews: 182424
Number of negative reviews: 28275
Number of words in positive reviews: 3138000
Number of words in negative reviews: 1495173
```

Sample positive review:
this is a great game and I LOVE DINOS

Sample negative review:
On my original account. I played Stomping lands, B4 Ark. and was a Beta Player for ark. and have Probably around 10,000 Hours accumulated On several accounts. Some of my Kids, over the years have taken over my Steam accounts, and I have started a New one, and RE Purchased Ark, on a few accounts, Approx, 3 or 4 times in total. I had Originally played Pvp Official servers. After SO MANY bugs and glitches and EXPLOITS. Of the original game. I went to pve, ON PVP many similar Exploits and while NOT Hacks, but Unfair, Unbalanced gameplay etc. reuining PVE servers. Enormous asian, mexican, and BR Tribes wiping out entire servers. Pillar/ foundationing entire clusters etc. it became UNBEARABLE to Play.1 I went on Hardcore official.. Only to become CONSTANTLY bullied , and again Unfair, gameplay. Went to private servers for a few years and found some TRUELY Fun, challenging gameplay. THE GOOD< THE BAD< AND THE UGLY< The CONCEPT, The Idea of the game. The landscape and Terrain, Resources, and Animal Interactions are AMAZING> BREATHTAKING. and Immersive on such a Deep compelling level. I remember in the first few times I played. Getting caught unexpectedly out in the wild, after nightfall In game, and not Knowing gamma setting bypass etc. having bows and spears. and Meeting a TREX in the Dark the first time. I'm a Grown ass man,. Nearly crapped my pants when it Roared and came after me. The Game Play was Really Immersive and Breath taking. But in very short time. the over all annoyance and frustration of falling through maps into never ending oceans below, and never being able to recover bodies, Pets glitching and falling through maps. Pets getting stuck in terrain and dying.. pets getting stuck in the sky and never being seen again.

Wild dinos and Players able to speed glitch through defenses and base walls etc. In a few months of regular game play, While they "were working on it". The Game play on Official especially just became a never ending disapointment of rebuilding, Retaming, recapturing, Moving to new servers, new clusters to start again I have to say, I struggled for YEARS<<< on ark official, As huge tribes made the game absolutely pointless to play for anyone who Didnt bribe them/ pay them off. or join them. If you were not part of a select FEW Massive cluster Owning tribes. You didnt have a prayer to survive and build. I am a Stubborn man, seriously so, and I spent YEARS,. Playing solo, running and hiding, Guerilla hit and run tactics etc. To only get captured and caged, and BULLIED into quiting playing on official at all in any capacity for a long long time. After a long time away playing private servers. I came back. Made some genuinely good friends in game. Some crazy russian, and Ukraine players in some big big tribes. At first I just ALlied them. I helped them out when / where i could taming certain, High level dinos etc. capturing perf wyvern etggs etc. They more or less left me alone, or threw me some pets / eggs time to time.. Times were good. Larger tribes started wiping those clusters again, and they offered me a spot with them on some other servers Official that were run I was told by chinese players. It was at this point I learned some interested game information.. NOT mentioning player names, or tribes. But after MORE than a Year with these big tribes I learned they had some Agreements with MODS of Ark etc.. where they were left alone. paid off somehow. Many of the BIG tribes have similar arrangements. where they can take over entire clusters of the official servers and suffer NO official backlash for whatever they do. The only thing they had to worry about was other larger tribes moving to their cluster and attacking. This led to the Realization that there is NO SUCH THING as Balanced, Fair Gameplay in ARK on any server Official. PVE or PVP AND If you dont know someone high up somehwere,, Your screwed. I am of the opion that everyone paid equally for the game, and all should have equal opportunity. and the Game play itself should be balanced in a way to NOT allow Giant tribes to wipe entire clusters etc. I watched ark, go from an In depth, Immersive, captivating experience. Into Unfair, Rage Inducing Garbage… Then then new Revamp of the GAME!!!!! Heard rumours about this. everyone will have to migrate, lose all you have etc. The Short version is.. They went back , Took the ORIGINAL GAME< the way it SHOULD HAVE BEEN< Done the first time. took out the bugs, glitches, and got rid of the outdated crap game engine. and redid it PROPERLY. the way is should have been done from the begining,. and then tried to resell it to everyone AGAIN> . WHY should I pay AGAIN for a product I have already supported, and Purchased SEVERAL times. To have the SAME kind of unfair, Unbalanced, Exploitive gameplay From Thuggish bullying Tribes who IMMEDIATELY Started taking over whole clusters again… I left again.. and will Probably never return to official servers in any capacity again, Explicitly because of these Problems. and have NO desire to pay you more of my hard earned money, to be bullied, and brow beaten into submission agian over years. Overall impression. Ark is a Stunning captivating world of landscapes, secrets, exploration and Crazy animal interactions. It could have been done so much better from the begining, and could have been balanced so much better from the get go. If it werent for the shady things allowed behind the

scenese. Exploits, and without a more sensible approach to ballanced gameplay. You would be so much better off playing private servers with Organized PVP battles and guidelines to prevent the online bullying, and total domination of all official servers in one capacity or another. INITIALLY Totally worth every dollar spent. memories of a lifetime. After Months In, and years of game play. Complete waste of time and money. UNLESS You want to join up with all the bullies and online gangs and become like them :)

[23]:
```
# Conclusion section
display(Markdown("## Conclusion"))
display(Markdown(f"""
This analysis provides comprehensive insights into the player reception and␣
 ↪trends of {GAME_NAME} on Steam. """))
```

1.12 Conclusion

This analysis provides comprehensive insights into the player reception and trends of ARK: Survival Evolved on Steam.