

# Лабораторная работа №2

**Задача:** Визуализация многомерных данных с matplotlib и seaborn

## Описание датасета "Predict Online Gaming Behavior Dataset"

Этот набор данных фиксирует комплексные метрики и демографию, связанные с поведением игроков в онлайн-игровых средах. Он включает такие переменные, как демография игроков, детали, характерные для игры, метрики вовлеченности и целевую переменную, отражающую удержание игроков.

Переменная	Описание
PlayerID	Уникальный идентификатор для каждого игрока.
Age	Возраст игрока.
Gender	Пол игрока.
Location	Географическое местоположение игрока.
GameGenre	Жанр игры, в которой участвует игрок.
PlayTimeHours	Среднее количество часов, проведенных за игрой за одну сессию.
InGamePurchases	Признак того, делает ли игрок внутриигровые покупки (0 — Нет, 1 — Да).
GameDifficulty	Уровень сложности игры.
SessionsPerWeek	Количество игровых сессий в неделю.
AvgSessionDurationMinutes	Средняя продолжительность каждой игровой сессии в минутах.
PlayerLevel	Текущий уровень игрока в игре.
AchievementsUnlocked	Количество достижений, разблокированных игроком.
EngagementLevel	Категоризированный уровень вовлеченности, отражающий удержание игроков ('Высокий', 'Средний', 'Низкий').

Целевая переменная — EngagementLevel — указывает на уровень вовлеченности игрока и категоризируется как 'Высокий', 'Средний' или 'Низкий'.

```
In [33]: import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import math
```

```
In [34]: df = pd.read_csv("../online_gaming_behavior_dataset.csv", index_col='PlayerID')
```

```
In [35]: df.info()

<class 'pandas.core.frame.DataFrame'>
Index: 40034 entries, 9000 to 49033
Data columns (total 12 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Age                                    40034 non-null  int64
1   Gender                                40034 non-null  object
2   Location                              40034 non-null  object
3   GameGenre                             40034 non-null  object
4   PlayTimeHours                         40034 non-null  float64
5   InGamePurchases                       40034 non-null  int64
6   GameDifficulty                         40034 non-null  object
7   SessionsPerWeek                       40034 non-null  int64
8   AvgSessionDurationMinutes             40034 non-null  int64
9   PlayerLevel                           40034 non-null  int64
10  AchievementsUnlocked                  40034 non-null  int64
11  EngagementLevel                       40034 non-null  object
dtypes: float64(1), int64(6), object(5)
memory usage: 4.0+ MB
```

## Визуализация данных

```
In [36]: # Настройка стиля графиков
sns.set(style="whitegrid");
plt.figure(figsize=(12, 6));
```

<Figure size 1200x600 with 0 Axes>

Целевая переменная

```
In [37]: column_name = 'EngagementLevel'
plt.figure(figsize=(10, 4))

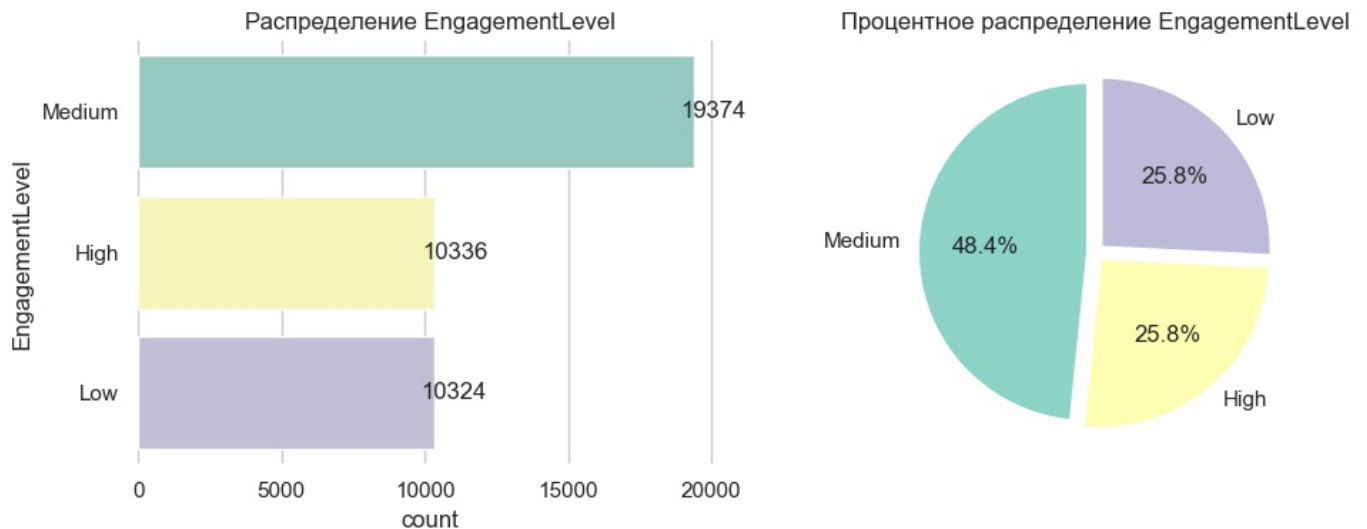
plt.subplot(1, 2, 1)
sns.countplot(y=column_name, data=df, palette='Set3', hue=column_name, legend=False)
plt.title(f'Распределение {column_name}')

ax = plt.gca()
for p in ax.patches:
    ax.annotate(f'{int(p.get_width())}', (p.get_width(), p.get_y() + p.get_height() / 2),
                ha='center', va='center', xytext=(10, 0), textcoords='offset points')

sns.despine(left=True, bottom=True)

plt.subplot(1, 2, 2)
df[column_name].value_counts().plot.pie(autopct='%1.1f%%', colors=sns.color_palette('Set3'), startangle=90, expand=True)
plt.title(f'Процентное распределение {column_name}')
plt.ylabel('')

plt.tight_layout()
plt.show()
```



### Числовые характеристики

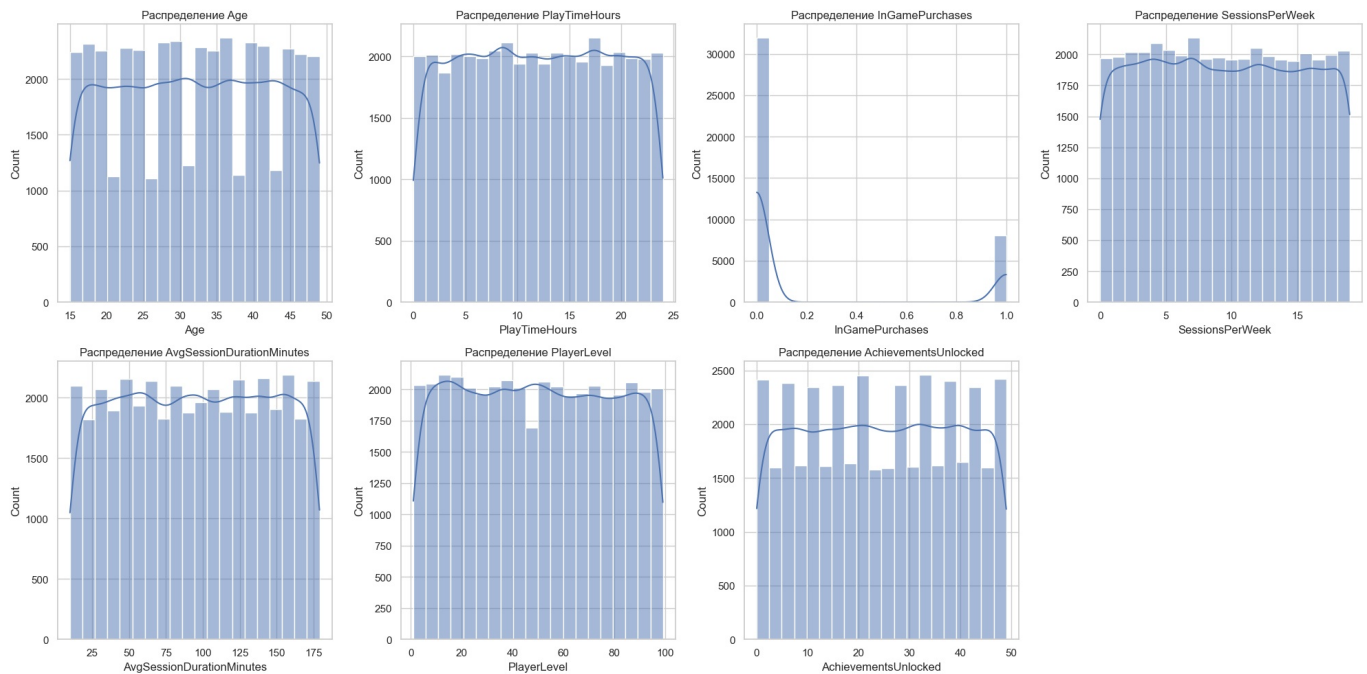
```
In [38]: numerical_columns = ['Age', 'PlayTimeHours', 'InGamePurchases', 'SessionsPerWeek',
                             'AvgSessionDurationMinutes', 'PlayerLevel', 'AchievementsUnlocked']

plt.figure(figsize=(20, 10))
bins = round(1 + 3.3 * math.log10(df.shape[0]))

for i, col in enumerate(numerical_columns[:4]):
    plt.subplot(2, 4, i+1)
    sns.histplot(df[col], bins=20, kde=True)
    plt.title(f'Распределение {col}')

for i, col in enumerate(numerical_columns[4:]):
    plt.subplot(2, 4, i+5)
    sns.histplot(df[col], bins=20, kde=True)
    plt.title(f'Распределение {col}')

plt.tight_layout()
plt.show()
```

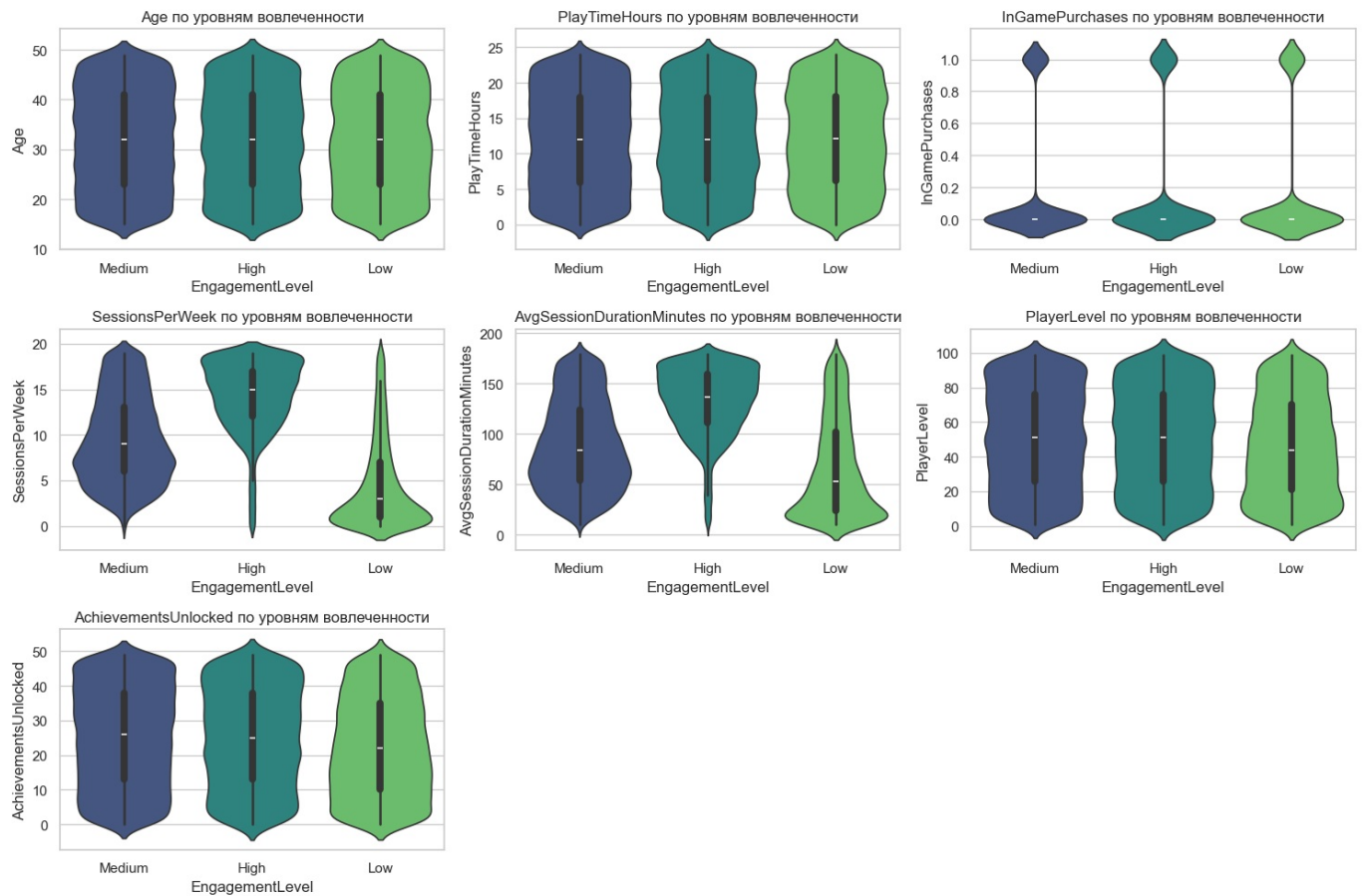


### Вывод по числовым характеристикам:

- Age:** Средний возраст игроков составляет около 32 лет, с относительно большим разбросом (стандартное отклонение ~10 лет). Самому молодому игроку 15 лет, а самому старшему — 49 лет. Большинство игроков находятся в возрасте от 23 лет (25-й процентиль) до 41 года (75-й процентиль), что указывает на то, что игра привлекает разнообразную возрастную группу, но имеет небольшой перекося в сторону молодых людей.
- PlayTimeHours:** Среднее время игры составляет 12 часов, с широким диапазоном от почти 0 часов (минимум) до 24 часов (максимум). Медианное время игры составляет 12 часов, что говорит о том, что распределение времени игры примерно симметрично. 25-й и 75-й процентиля (6 и ~18 часов соответственно) указывают на то, что значительная часть базы игроков играет от 6 до 18 часов. Некоторые игроки показывают большое количество игровых часов, что, возможно, отражает высокую вовлеченность или хардкорных геймеров.
- InGamePurchases:** Данные являются бинарными, большинство игроков не совершают внутриигровых покупок (0), в то время как меньшая часть совершает (1). Это соответствует типичным тенденциям, когда только меньшинство игроков вносит вклад в внутриигровую монетизацию.
- SessionsPerWeek:** Игроки заходят в среднем ~9,5 раз в неделю, со стандартным отклонением ~5,76 сеансов. 25-й процентиль составляет 4 сеанса, что означает, что менее частые игроки заходят через день, а 75-й процентиль составляет 14 сеансов, что означает, что некоторые игроки заходят дважды в день.
- AvgSessionDurationMinutes:** Продолжительность сеанса показывает нормальное распределение, сосредоточенное вокруг 90–100 минут на сеанс. Большинство игроков укладываются в 50–150 минут, экстремальные длительности редки.
- PlayerLevel:** Распределение равномерное, игроки распределены по всем уровням от 1 до 99. Значительных скачков нет, что говорит о том, что прогресс уровней распределен равномерно.
- AchievementsUnlocked:** Данные достигают пика около 15–30 разблокированных достижений, при этом меньшее количество игроков разблокирует много достижений. Распределение подчеркивает умеренные уровни вовлеченности, при этом меньшее количество игроков полностью завершают списки достижений.

### Числовые признаки и целевая переменная

```
In [39]: numeric_cols = ['Age', 'PlayTimeHours', 'InGamePurchases', 'SessionsPerWeek',
                        'AvgSessionDurationMinutes', 'PlayerLevel', 'AchievementsUnlocked']
plt.figure(figsize=(15, 10))
for i, col in enumerate(numeric_cols, 1):
    plt.subplot(3, 3, i)
    sns.violinplot(data=df, x='EngagementLevel', y=col, hue='EngagementLevel', palette='viridis', legend=False)
    plt.title(f'{col} по уровням вовлеченности')
plt.tight_layout()
plt.show()
```



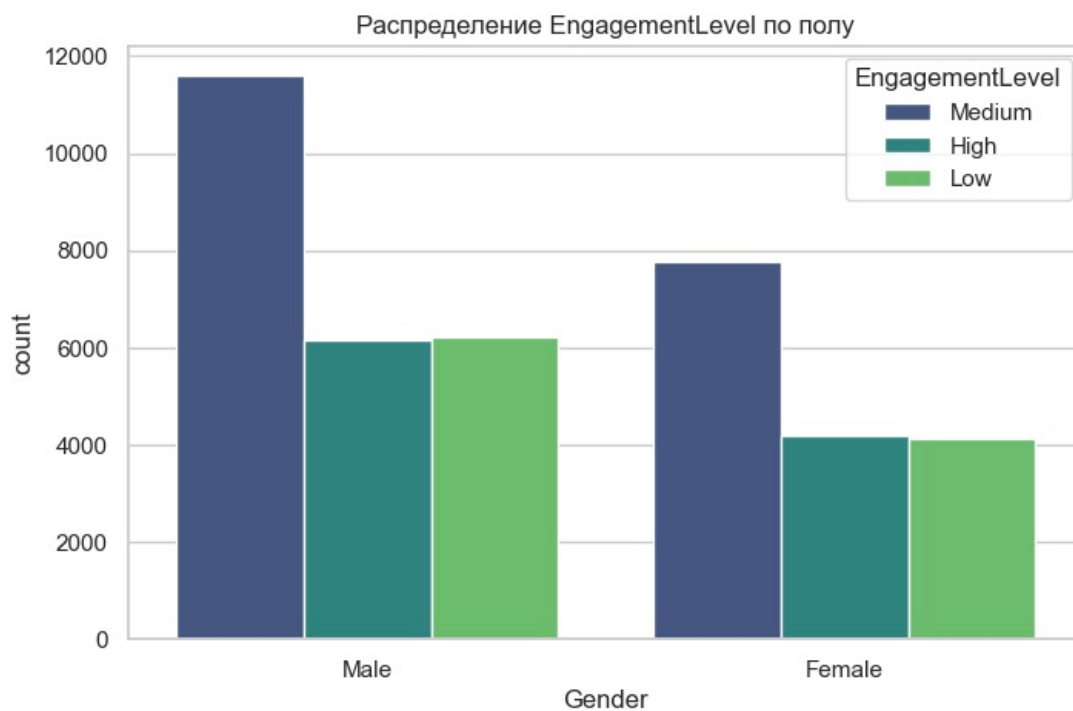
#### Вывод:

1. SessionPerWeek, AvgSessionDurationMinutes - из графиков видно, что те кто больше увлечены играми, те больше проводят времени за играми
2. AchievementsUnlocked - игроки с высоким EngagementLevel имеют больше достижений.

#### Категориальные признаки

##### Гендер

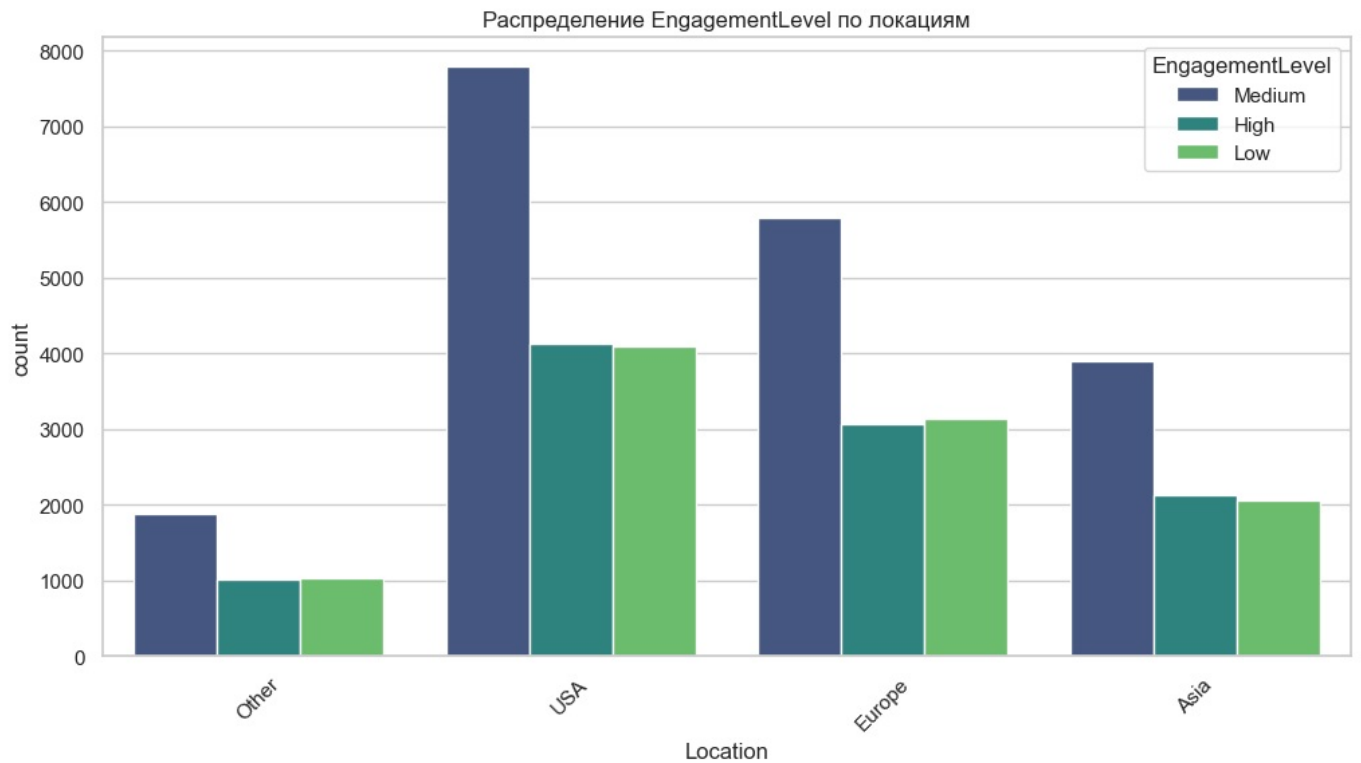
```
In [40]: plt.figure(figsize=(8, 5))
sns.countplot(data=df, x='Gender', hue='EngagementLevel', palette='viridis')
plt.title('Распределение EngagementLevel по полу')
plt.show()
```



Мужчины демонстрируют более высокий уровень вовлеченности по сравнению с женщинами.

Локация

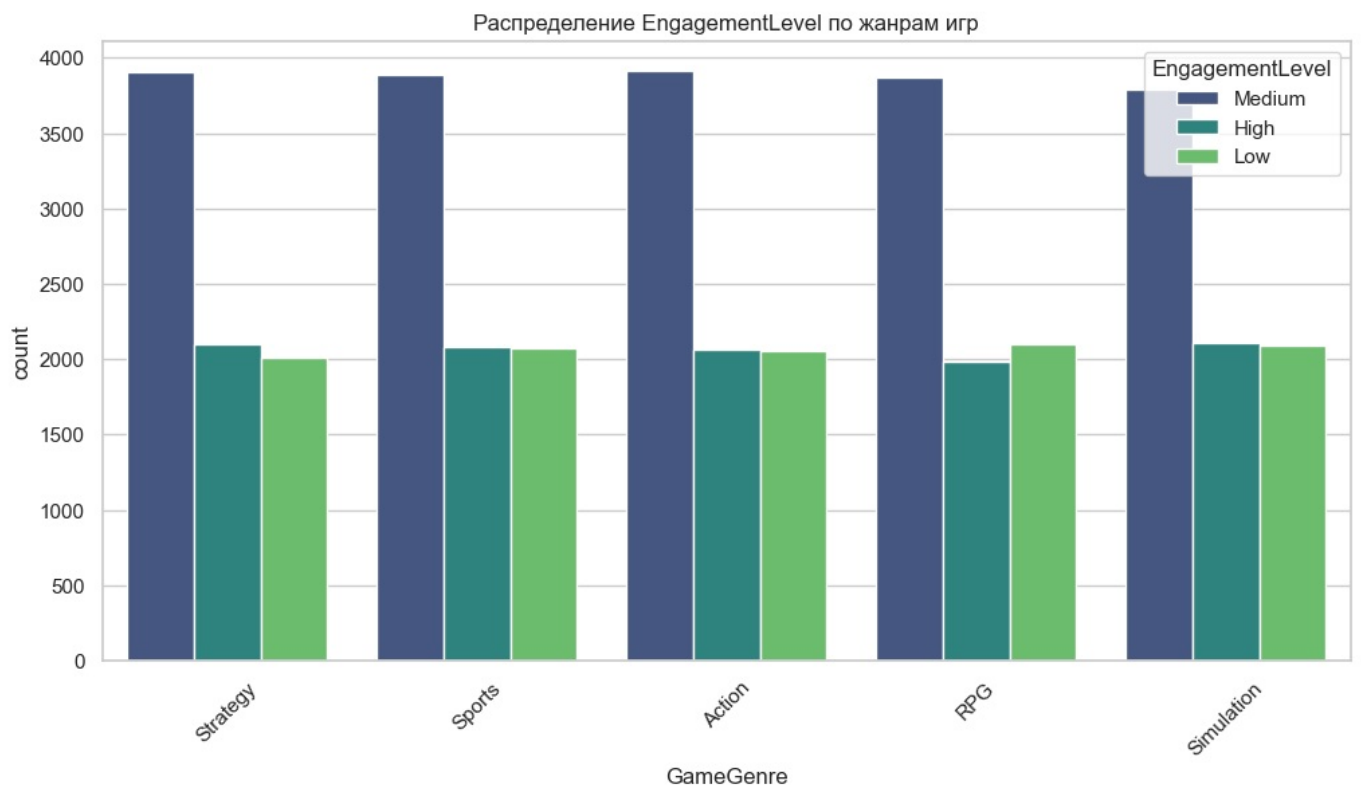
```
In [41]: plt.figure(figsize=(12, 6))
sns.countplot(data=df, x='Location', hue='EngagementLevel', palette='viridis')
plt.title('Распределение EngagementLevel по локациям')
plt.xticks(rotation=45)
plt.show()
```



Игроки из USA и Europe чаще имеют высокий уровень вовлеченности.

Жанр игры

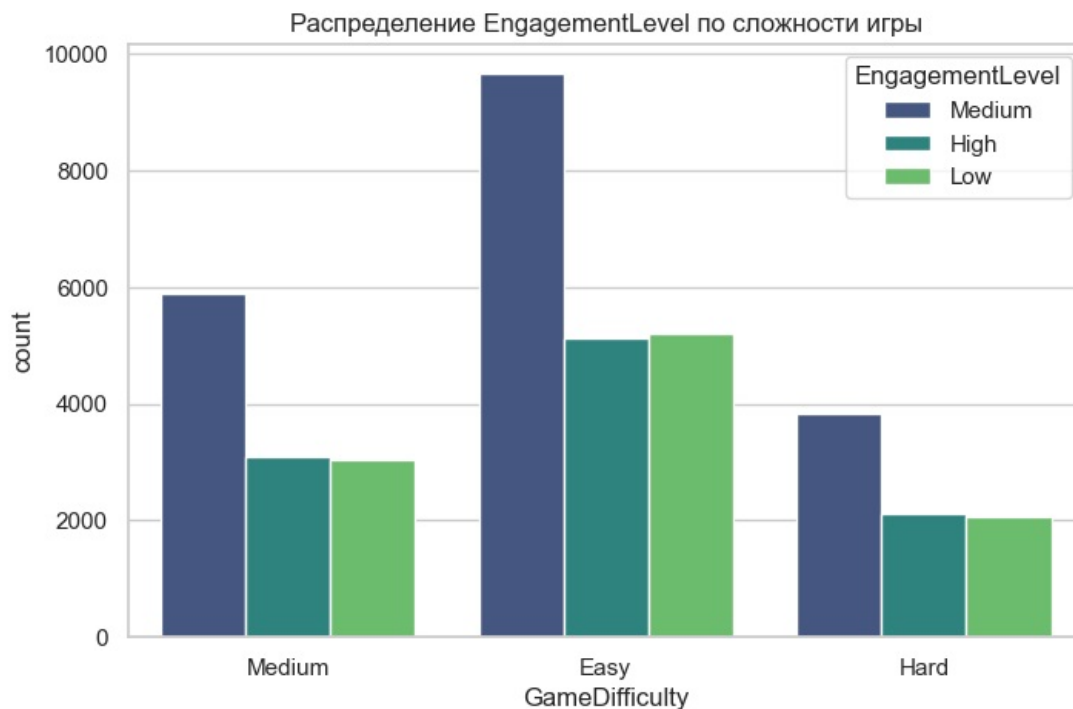
```
In [42]: plt.figure(figsize=(12, 6))
sns.countplot(data=df, x='GameGenre', hue='EngagementLevel', palette='viridis')
plt.title('Распределение EngagementLevel по жанрам игр')
plt.xticks(rotation=45)
plt.show()
```



Графики похожие и никаких зависимостей нет

Сложность игры

```
In [43]: plt.figure(figsize=(8, 5))
sns.countplot(data=df, x='GameDifficulty', hue='EngagementLevel', palette='viridis')
plt.title('Распределение EngagementLevel по сложности игры')
plt.show()
```



Низкая сложность привлекает больше игроков с высокой вовлеченностью

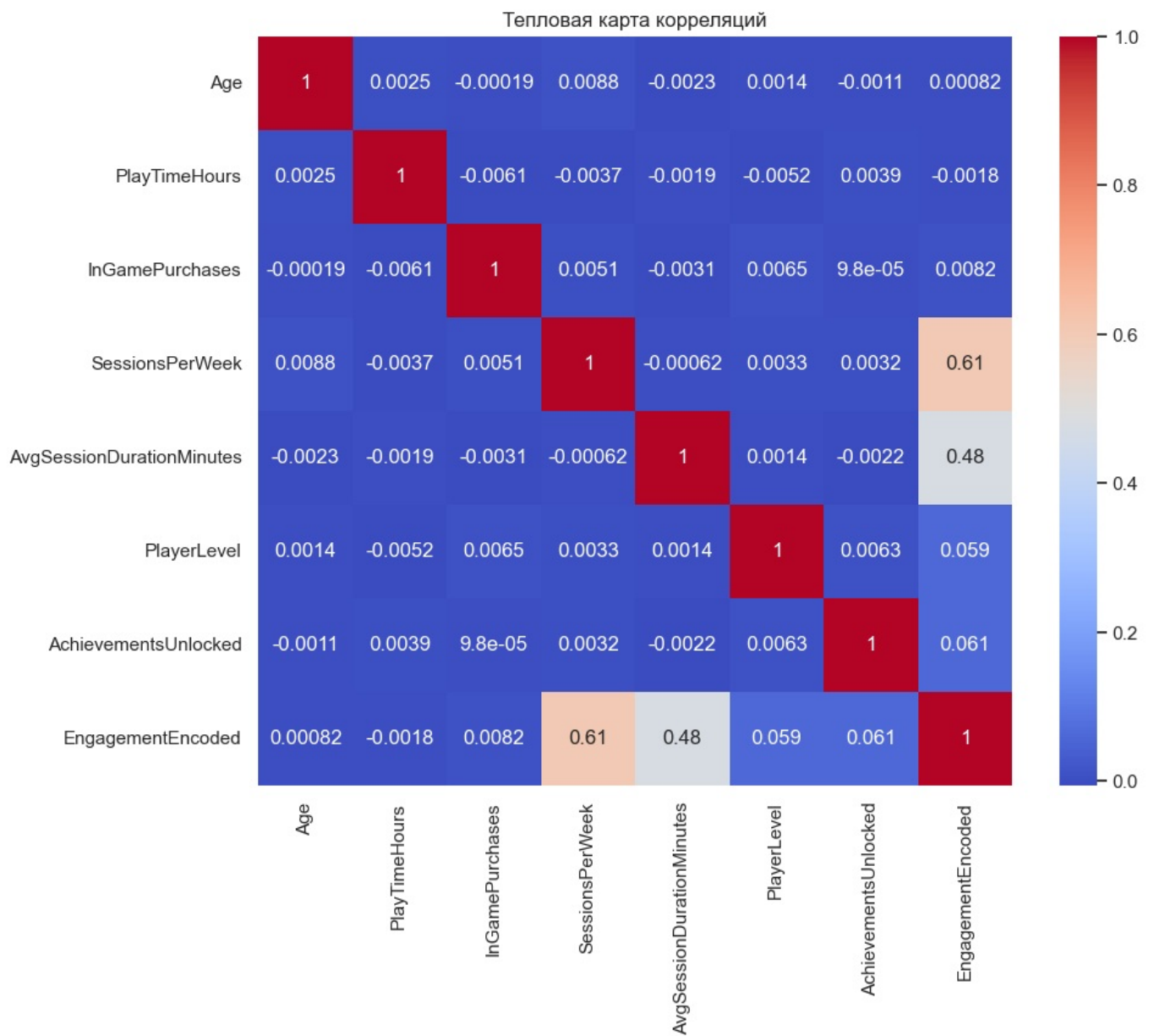
#### Вывод:

1. Гендер: Мужчины более вовлечены в игры, чем женщины.
2. Локация: Игроки из USA и Europe чаще имеют высокий уровень вовлеченности.
3. Сложность игры: Низкая сложность привлекает больше игроков с высокой вовлеченностью

#### Корреляционный анализ

```
In [44]: engagement_order = {'Low':1, 'Medium':2, 'High':3}
df['EngagementEncoded'] = df['EngagementLevel'].map(engagement_order)

plt.figure(figsize=(10, 8))
corr = df[numeric_cols + ['EngagementEncoded']].corr()
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title('Тепловая карта корреляций')
plt.show()
```



**Вывод:** EngagementEncoded коррелирует с SessionPerWeek И AvgSessionDurationMinutes

Анализ сессий

```
In [45]: plt.figure(figsize=(10, 6))
sns.scatterplot(
    data=df,
    x='SessionsPerWeek',
    y='AvgSessionDurationMinutes',
    hue='EngagementLevel',
    palette='viridis',
    alpha=0.7
)
plt.title('Зависимость EngagementLevel от параметров сессий')
plt.show()
```



1. Игроки с высоким уровнем вовлеченности образуют отдельный кластер в правом верхнем углу графика, что указывает на их активность и продолжительность игровых сессий.
2. Игроки с низким уровнем вовлеченности сгруппированы в левом нижнем углу, что свидетельствует о их минимальной активности.

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js