Korean LoL Pro Players Tier Change Analysis Report

Executive Summary:

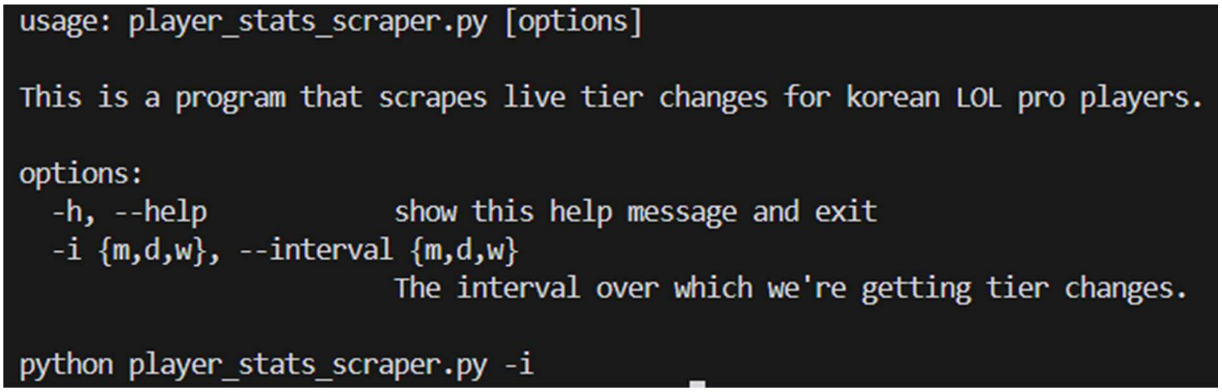
This report analyzes tier changes and performance metrics for Korean League of Legends professional players across daily, weekly, and monthly timeframes. The project scrapes data from OP.GG's live tier changes tracker, processes the information, and provides statistical insights on player and team performance.

Project Structure:

1. **Data Collection:**

   The project uses Playwright to scrape data from OP.GG with the following key components:

   - Command-line Interface: Allows users to specify data collection intervals (daily, weekly, monthly)



```
usage: player_stats_scraper.py [options]

This is a program that scrapes live tier changes for korean LOL pro players.

options:
  -h, --help            show this help message and exit
  -i {m,d,w}, --interval {m,d,w}
                        The interval over which we're getting tier changes.

python player_stats_scraper.py -i
```

   - Web Scraping Logic: Extracts player IDs, team affiliations, LP changes, and win rates
   - Data Export: Saves collected data to CSV files for further analysis

2. **Data Analysis**:

   The analysis system processes the collected data to provide insights on:

   - Player Performance: Tracks individual LP gains/losses and win rates
   - Team Performance: Aggregates metrics to identify top-performing teams
   - Correlation Analysis: Examines relationships between win rates and LP changes
   - Visual Reporting: Creates visualizations for easier interpretation of results

Technical Implementation:

1. **Scraping Module:**

   The scraping module uses Playwright for browser automation with the following workflow:

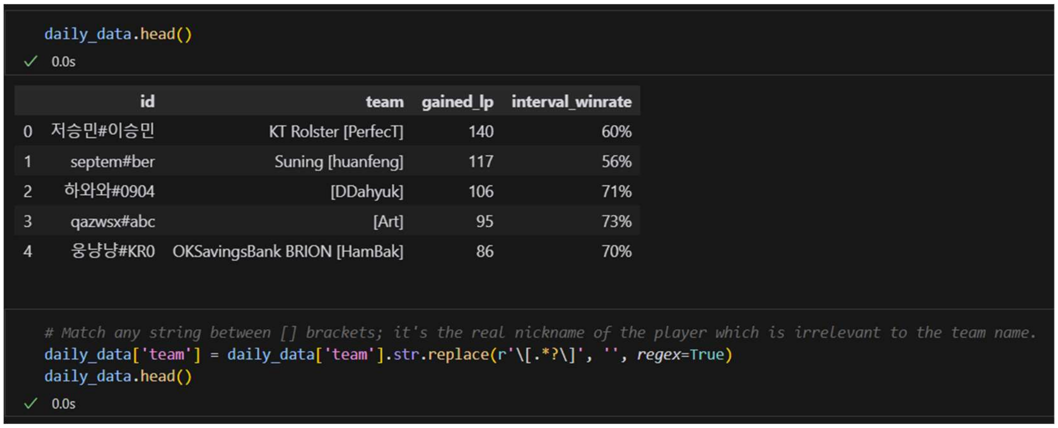
   - Parses command-line arguments to determine data collection interval
   - Navigates to the appropriate OP.GG URL based on the selected interval

- Extracts player data including:
    - Player ID
    - Team
    - LP Changes (positive or negative)
    - Interval-specific win rate
- Exports data to CSV files (daily_improvement.csv, weekly_improvement.csv, monthly_improvement.csv)

2. Analysis Module:

The analysis module processes the scraped data through several steps:

- Data Cleaning:
    - Extracts team names by removing extraneous text in brackets



```
daily_data.head()
```

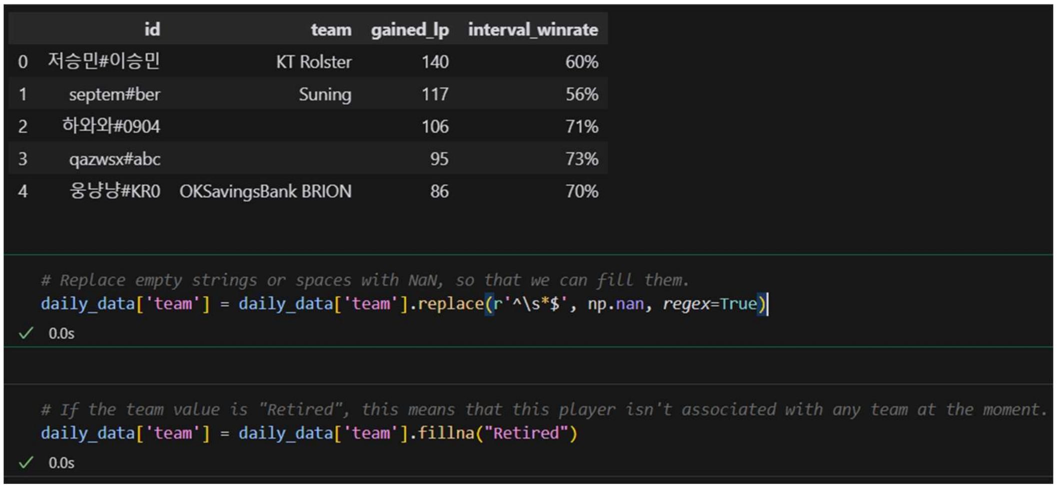
✓ 0.0s

	id	team	gained_lp	interval_winrate
0	저승민#이승민	KT Rolster [PerfecT]	140	60%
1	septem#ber	Suning [huanfeng]	117	56%
2	하와와#0904	[DDahyuk]	106	71%
3	qazwsx#abc	[Art]	95	73%
4	웅냥냥#KR0	OKSavingsBank BRION [HamBak]	86	70%

```
# Match any string between [] brackets; it's the real nickname of the player which is irrelevant to the team name.
daily_data['team'] = daily_data['team'].str.replace(r'\[.*?\]', '', regex=True)
daily_data.head()
```

✓ 0.0s

- Standardizes team information, marking unaffiliated players as "Retired"



	id	team	gained_lp	interval_winrate
0	저승민#이승민	KT Rolster	140	60%
1	septem#ber	Suning	117	56%
2	하와와#0904		106	71%
3	qazwsx#abc		95	73%
4	웅냥냥#KR0	OKSavingsBank BRION	86	70%

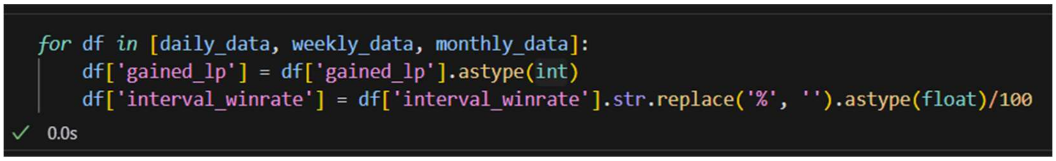
```
# Replace empty strings or spaces with NaN, so that we can fill them.
daily_data['team'] = daily_data['team'].replace(r'^\s*$', np.nan, regex=True)
```

✓ 0.0s

```
# If the team value is "Retired", this means that this player isn't associated with any team at the moment.
daily_data['team'] = daily_data['team'].fillna("Retired")
```

✓ 0.0s

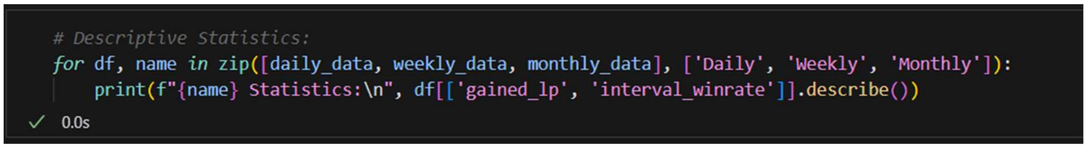
- Converts win rates to decimal format and LP changes to integers



```
for df in [daily_data, weekly_data, monthly_data]:
    df['gained_lp'] = df['gained_lp'].astype(int)
    df['interval_winrate'] = df['interval_winrate'].str.replace('%', '').astype(float)/100
```

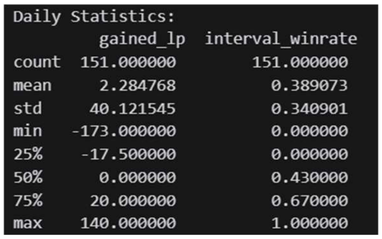
✓ 0.0s

- Statistical Analysis:
    - Calculates descriptive statistics for LP changes and win rates

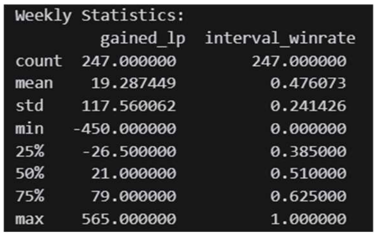


```
# Descriptive Statistics:
for df, name in zip([daily_data, weekly_data, monthly_data], ['Daily', 'Weekly', 'Monthly']):
    print(f"{name} Statistics:\n", df[['gained_lp', 'interval_winrate']].describe())
```

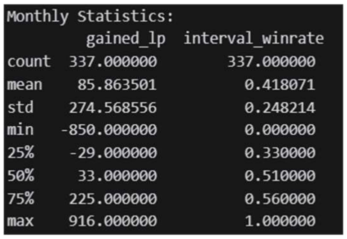
✓ 0.0s



```
Daily Statistics:
           gained_lp  interval_winrate
count  151.000000        151.000000
mean     2.284768          0.389073
std     40.121545          0.340901
min   -173.000000          0.000000
25%    -17.500000          0.000000
50%      0.000000          0.430000
75%     20.000000          0.670000
max    140.000000          1.000000
```

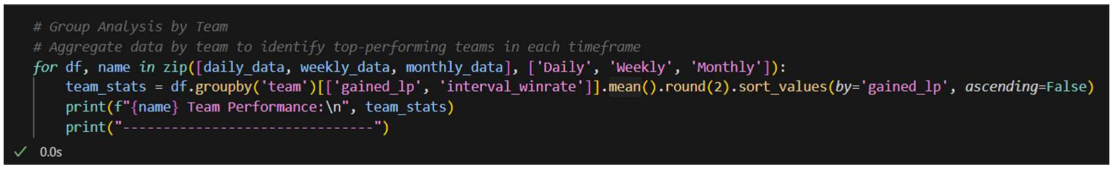


```
Weekly Statistics:
           gained_lp  interval_winrate
count  247.000000        247.000000
mean    19.287449          0.476073
std    117.560062          0.241426
min   -450.000000          0.000000
25%    -26.500000          0.385000
50%     21.000000          0.510000
75%     79.000000          0.625000
max    565.000000          1.000000
```



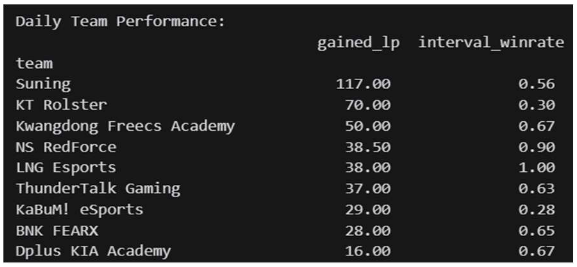
```
Monthly Statistics:
           gained_lp  interval_winrate
count  337.000000        337.000000
mean    85.863501          0.418071
std    274.568556          0.248214
min   -850.000000          0.000000
25%    -29.000000          0.330000
50%     33.000000          0.510000
75%    225.000000          0.560000
max    916.000000          1.000000
```

- Groups data by team to identify high-performing organizations

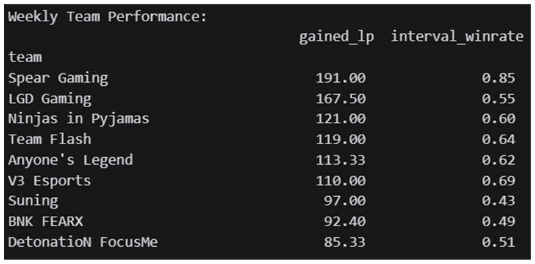


```
# Group Analysis by Team
# Aggregate data by team to identify top-performing teams in each timeframe
for df, name in zip([daily_data, weekly_data, monthly_data], ['Daily', 'Weekly', 'Monthly']):
    team_stats = df.groupby('team')[['gained_lp', 'interval_winrate']].mean().round(2).sort_values(by='gained_lp', ascending=False)
    print(f"{name} Team Performance:\n", team_stats)
    print("------------------------------")
```

✓ 0.0s

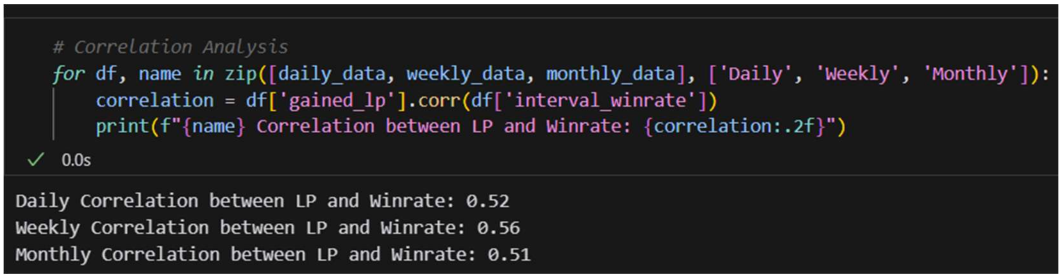


```
Daily Team Performance:
                            gained_lp  interval_winrate
team
Suning                        117.00              0.56
KT Rolster                     70.00              0.30
Kwangdong Freecs Academy       50.00              0.67
NS RedForce                    38.50              0.90
LNG Esports                    38.00              1.00
ThunderTalk Gaming             37.00              0.63
KaBuM! eSports                 29.00              0.28
BNK FEARX                      28.00              0.65
Dplus KIA Academy              16.00              0.67
```



```
Weekly Team Performance:
                                 gained_lp  interval_winrate
team
Spear Gaming                        191.00              0.85
LGD Gaming                          167.50              0.55
Ninjas in Pyjamas                   121.00              0.60
Team Flash                          119.00              0.64
Anyone's Legend                     113.33              0.62
V3 Esports                          110.00              0.69
Suning                               97.00              0.43
BNK FEARX                            92.40              0.49
DetonatioN FocusMe                   85.33              0.51
```

- Analyzes correlation between win rates and LP changes

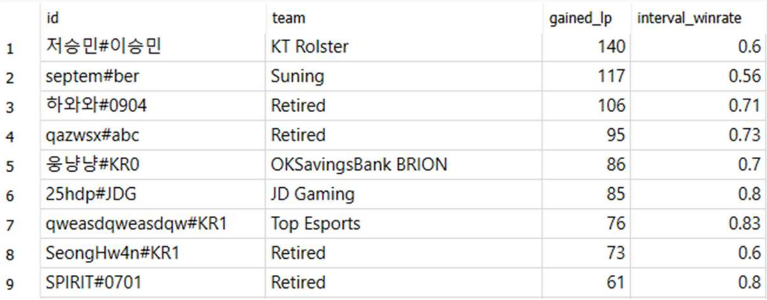


```
# Correlation Analysis
for df, name in zip([daily_data, weekly_data, monthly_data], ['Daily', 'Weekly', 'Monthly']):
    correlation = df['gained_lp'].corr(df['interval_winrate'])
    print(f"{name} Correlation between LP and Winrate: {correlation:.2f}")
```

✓ 0.0s

```
Daily Correlation between LP and Winrate: 0.52
Weekly Correlation between LP and Winrate: 0.56
Monthly Correlation between LP and Winrate: 0.51
```

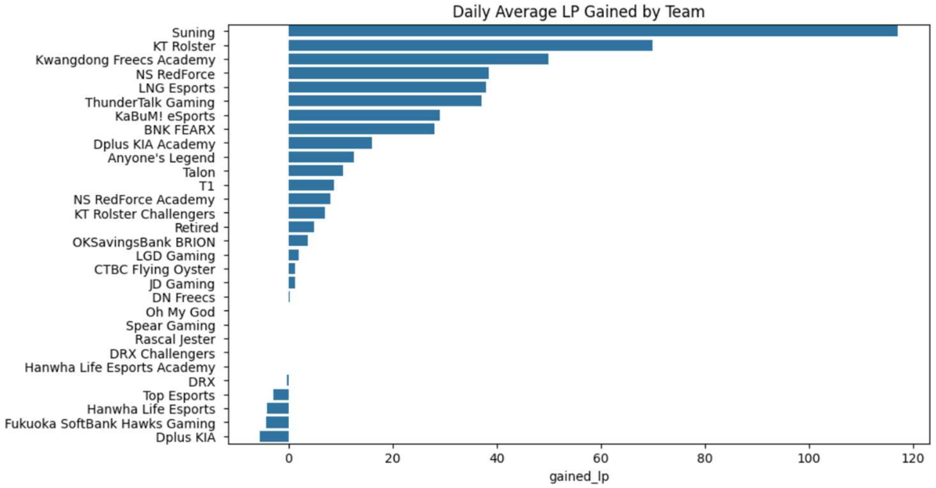
- Data Storage:
    - Stores processed data in SQLite database for persistence
    - Creates separate tables for daily, weekly, and monthly data

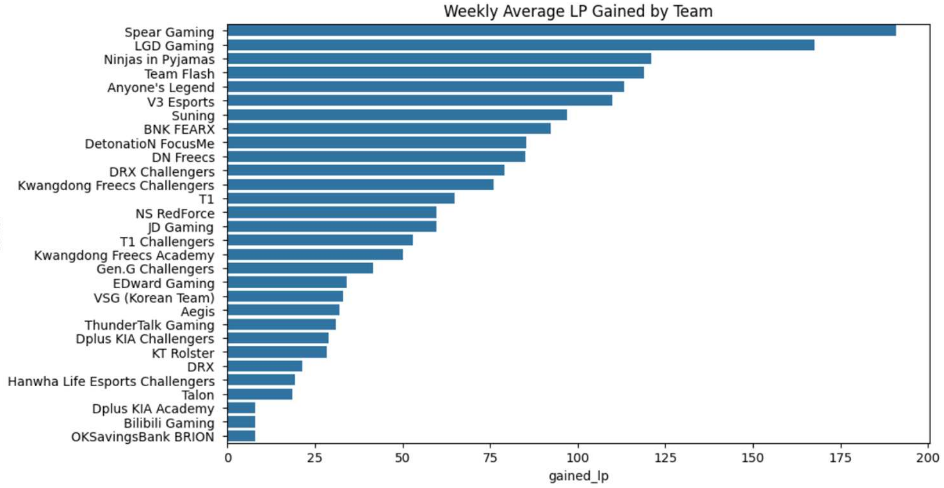


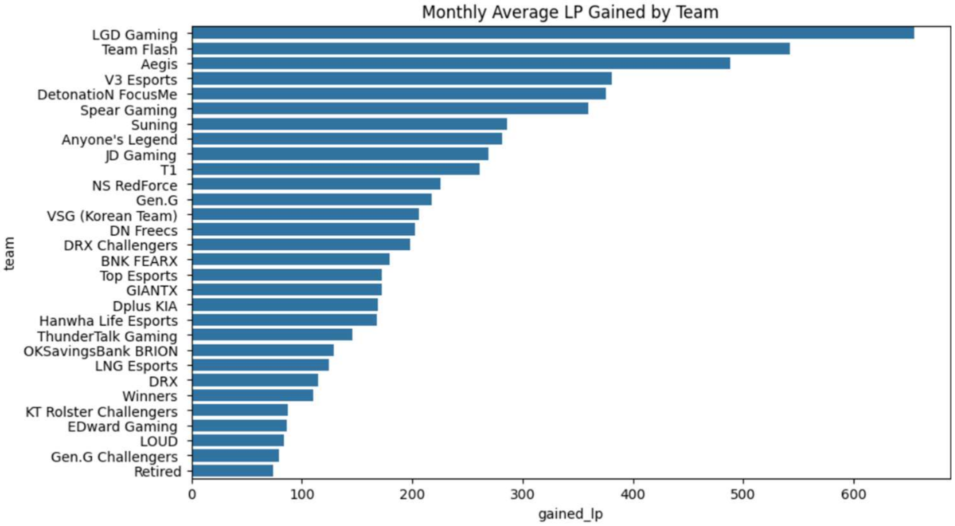
	id	team	gained_lp	interval_winrate
1	저승민#이승민	KT Rolster	140	0.6
2	septem#ber	Suning	117	0.56
3	하와와#0904	Retired	106	0.71
4	qazwsx#abc	Retired	95	0.73
5	웅냥냥#KR0	OKSavingsBank BRION	86	0.7
6	25hdp#JDG	JD Gaming	85	0.8
7	qweasdqweasdqw#KR1	Top Esports	76	0.83
8	SeongHw4n#KR1	Retired	73	0.6
9	SPIRIT#0701	Retired	61	0.8

table 1 (daily stats)

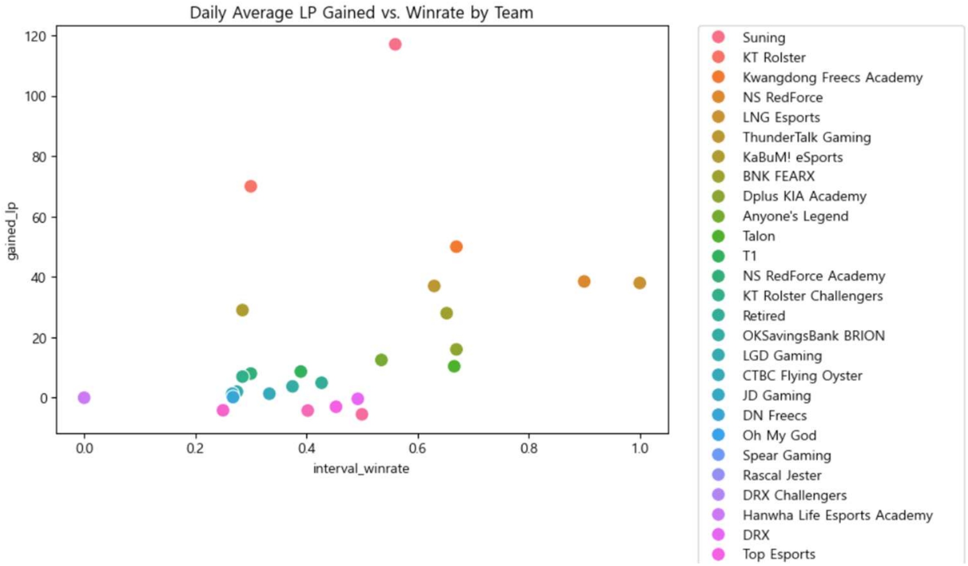
- Visualization:
    - Generates bar plots showing LP gains by team

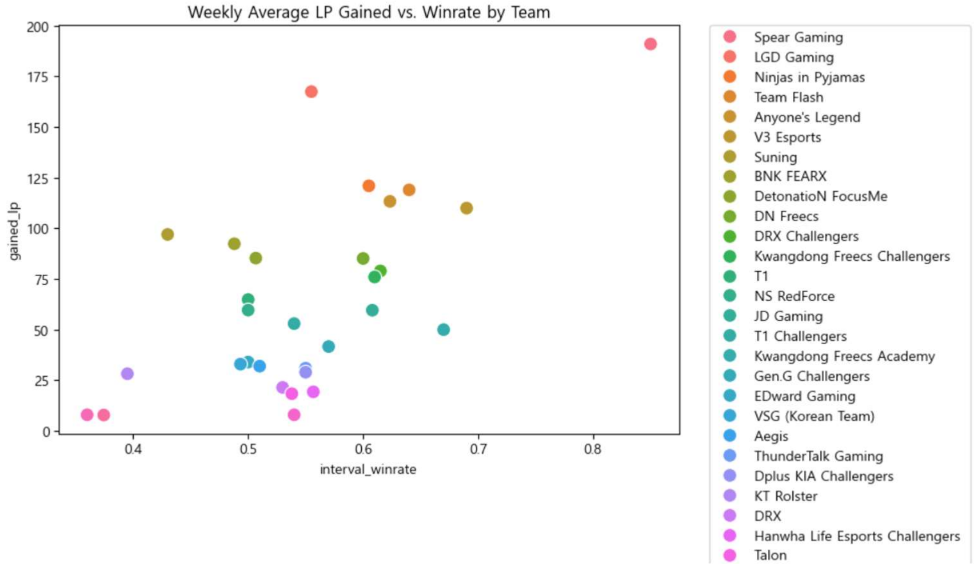


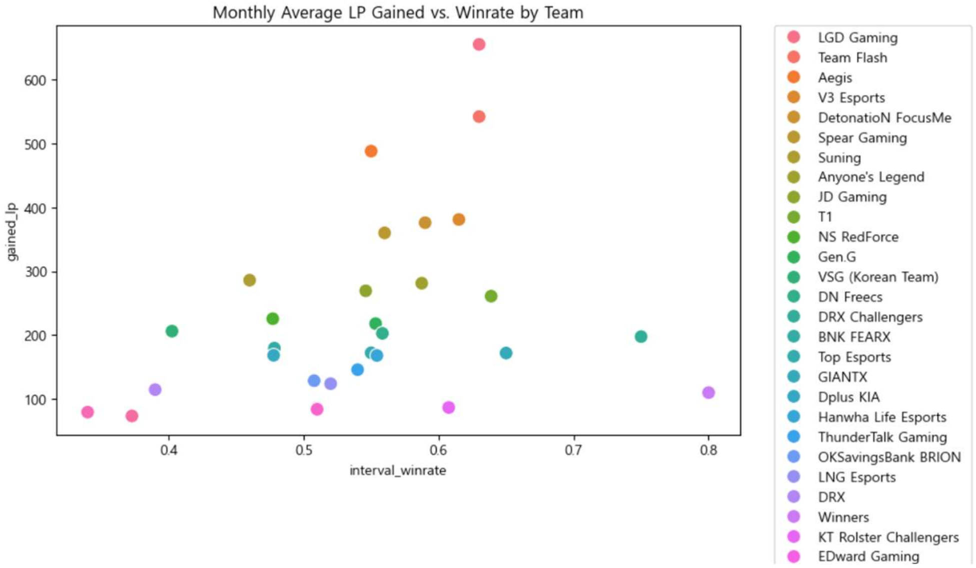




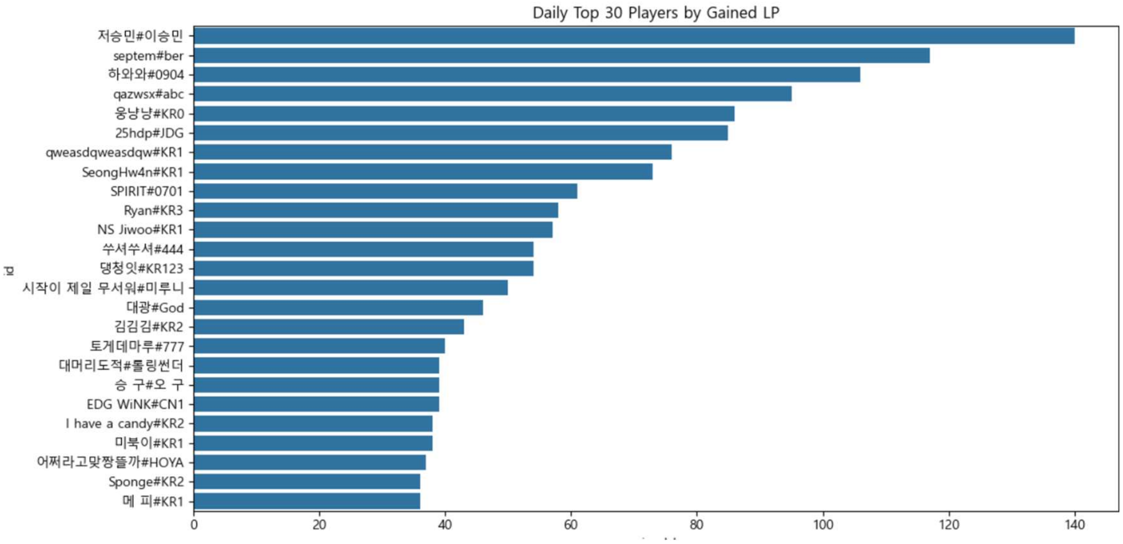
- Creates scatter plots comparing team win rates and LP changes

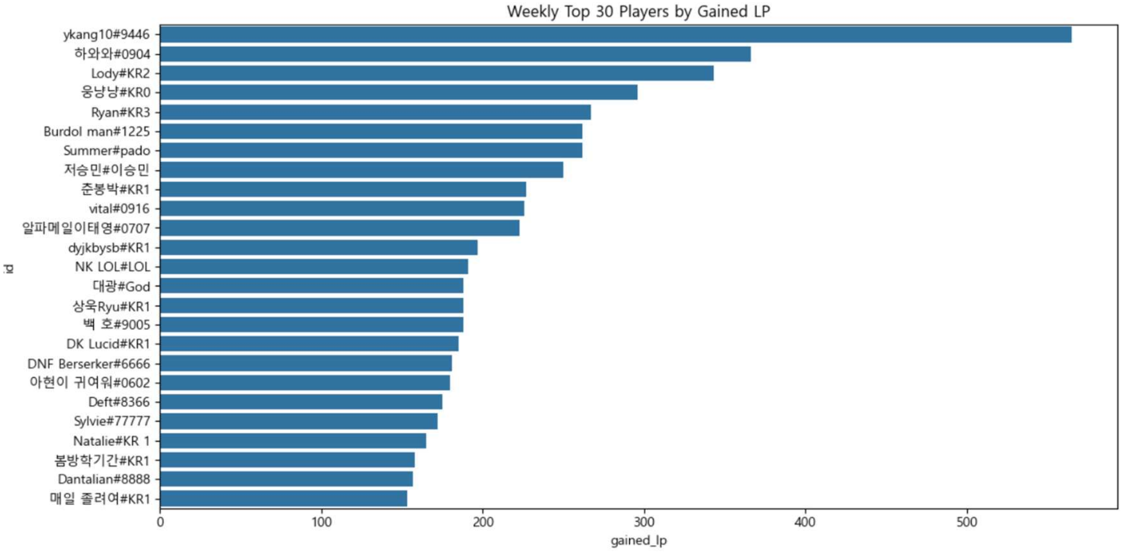


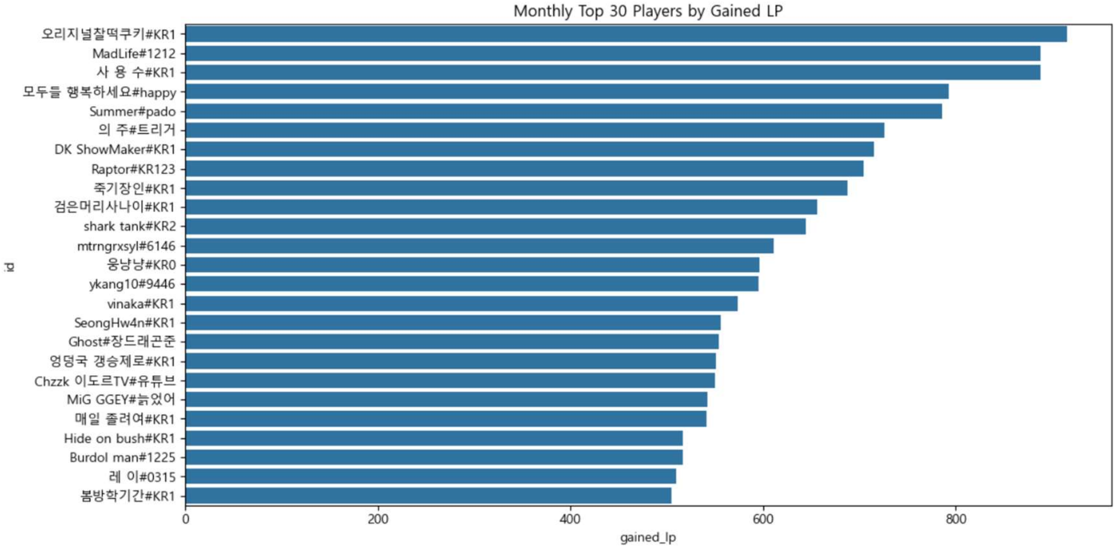




- Produces player performance charts with special handling for Korean character display







Key Findings:

- Top-performing teams based on average LP gains
- Correlation strength between win rates and LP progression
- Performance trends across different time intervals
- Teams with the most consistent player improvements
- Individual player performance rankings across different timeframes

Conclusion:

This Korean LoL Pro Players Tier Change Analysis project provides valuable insights into player and team performance trends. By tracking LP changes and win rates across different time intervals, the system helps identify which teams are most effectively supporting their players' ranked progression.

The project demonstrates effective solutions to technical challenges, including Korean language support in data visualization, creating a comprehensive analysis framework for the Korean professional League of Legends ecosystem.

Participants:

1.	Ahmed Ehab Ahmed Aboulmagd	23012057
2.	Nour Kamal Ahmed Mohamed Salem	23012061
3.	Abdallah Saber Desouki	23012064