Korean Baseball Pitching

By

Christian Huerta

Questions:

1. What is the average ERA (Earned Run Average) across teams?
2. Is there a correlation between the age of players and their ERA?
3. Which team has the highest average runs per game?
4. What is the average hits per 9 innings (H/9) for each team?
5. What is the distribution of wins and losses across teams?
6. What is the average strikeouts per 9 innings (K/9) for each team?
7. What are the top five teams with the most strikeouts?
8. Which team has the highest WHIP (Walks plus Hits per Inning Pitched) on average?
9. What team has the highest average of homeruns per 9 innings?
10. What is the distribution of wild pitches and hit batters across different teams?

Import Libraries:

In [490…

```
# Importing pandas for data manipulation
import pandas as pd
# Importing numpy for numerical operations
import numpy as np
# Importing matplotlib for data visualization
import matplotlib.pyplot as plt
# Importing scipy
from scipy import stats
# Importing seaborn for advanced data visualization
import seaborn as sns
# Setting the seaborn style for the plots
sns.set()
```

Import Data:

In [443…

```
# Reading the dataset into a pandas DataFrame
df = pd.read_csv('Korean Baseball.csv')
```

Inspect Data:

In [444…

```
# Reading the first 5 rows.
df.head()
```

Out[444…

	id	year	team	average_age	runs_per_game	wins	losses	win_loss_percentage	ER
0	1	2021	LG Twins	26.3	3.90	72	57	0.558	3.5
1	2	2021	KT Wiz	28.4	4.06	75	59	0.560	3.6
2	3	2021	Doosan Bears	27.5	4.57	70	65	0.519	4.2
3	4	2021	Samsung Lions	28.8	4.57	75	59	0.560	4.2
4	5	2021	NC Dinos	27.7	4.80	67	67	0.500	4.5

5 rows × 34 columns

In [445…

```
# Reading the last 5 rows.
df.tail()
```

Out[445…

	id	year	team	average_age	runs_per_game	wins	losses	win_loss_percentage
318	319	1982	OB Bears	26.5	3.98	56	24	0.700
319	320	1982	MBC Blue Dragons	25.9	4.38	46	33	0.582
320	321	1982	Lotte Giants	26.3	4.81	31	49	0.388
321	322	1982	Haitai Tigers	23.5	4.85	38	42	0.475
322	323	1982	Sammi Superstars	24.3	7.18	15	65	0.188

5 rows × 34 columns

In [446…

```
# Reading the information of the data.
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 323 entries, 0 to 322
Data columns (total 34 columns):
 #   Column              Non-Null Count  Dtype  
---  ------              --------------  -----  
 0   id                  323 non-null    int64  
 1   year                323 non-null    int64  
 2   team                323 non-null    object 
 3   average_age         323 non-null    float64
 4   runs_per_game       323 non-null    float64
 5   wins                323 non-null    int64  
 6   losses              323 non-null    int64  
 7   win_loss_percentage 323 non-null    float64
 8   ERA                 323 non-null    float64
 9   run_average_9       323 non-null    float64
 10  games               323 non-null    int64  
 11  games_started       184 non-null    float64
 12  games_finished      184 non-null    float64
 13  complete_game       323 non-null    int64  
 14  shutouts            323 non-null    int64  
 15  saves               323 non-null    int64  
 16  innings_pitched     323 non-null    float64
 17  hits                323 non-null    int64  
 18  runs                323 non-null    int64  
 19  earned_runs         323 non-null    int64  
 20  home_runs           323 non-null    int64  
 21  walks               323 non-null    int64  
 22  intentional_walks   184 non-null    float64
 23  strikeouts          323 non-null    int64  
 24  hit_batter          323 non-null    int64  
 25  balks               184 non-null    float64
 26  wild_pitches        184 non-null    float64
 27  batters_faced       323 non-null    int64  
 28  WHIP                323 non-null    float64
 29  hits_9              323 non-null    float64
 30  homeruns_9          323 non-null    float64
 31  walks_9             323 non-null    float64
 32  strikeouts_9        323 non-null    float64
 33  strikeout_walk      323 non-null    float64
dtypes: float64(17), int64(16), object(1)
memory usage: 85.9+ KB
```

In [447…

```
# Reading the description of the data.
df.describe()
```

Out[447…

	id	year	average_age	runs_per_game	wins	losses	win_
count	323.000000	323.000000	323.000000	323.000000	323.000000	323.000000	
mean	162.000000	2002.944272	26.886687	4.621858	62.507740	62.482972	
std	93.386294	11.501957	1.608472	0.734223	12.508225	12.446988	
min	1.000000	1982.000000	23.300000	2.980000	15.000000	24.000000	
25%	81.500000	1993.000000	25.700000	4.040000	54.000000	53.000000	
50%	162.000000	2003.000000	26.900000	4.620000	63.000000	62.000000	
75%	242.500000	2013.000000	28.000000	5.060000	71.000000	71.500000	
max	323.000000	2021.000000	32.400000	7.180000	93.000000	97.000000	

8 rows × 33 columns

Clean Data:

In [448…

```
# Reading if there is any Null
df.isnull().sum()
```

Out[448…
```
id                     0
year                   0
team                   0
average_age            0
runs_per_game          0
wins                   0
losses                 0
win_loss_percentage    0
ERA                    0
run_average_9          0
games                  0
games_started        139
games_finished       139
complete_game          0
shutouts               0
saves                  0
innings_pitched        0
hits                   0
runs                   0
earned_runs            0
home_runs              0
walks                  0
intentional_walks    139
strikeouts             0
hit_batter             0
balks                139
wild_pitches         139
batters_faced          0
WHIP                   0
hits_9                 0
homeruns_9             0
walks_9                0
strikeouts_9           0
strikeout_walk         0
dtype: int64
```

In [449…
```python
# Filling the Nulls
df.ffill(inplace=True)
```

In [450…
```python
# Showing that it filled the Nulls
df.isnull().sum()
```

Out[450…
```
id                     0
year                   0
team                   0
average_age            0
runs_per_game          0
wins                   0
losses                 0
win_loss_percentage    0
ERA                    0
run_average_9          0
games                  0
games_started          0
games_finished         0
complete_game          0
shutouts               0
saves                  0
innings_pitched        0
hits                   0
runs                   0
earned_runs            0
home_runs              0
walks                  0
intentional_walks      0
strikeouts             0
hit_batter             0
balks                  0
wild_pitches           0
batters_faced          0
WHIP                   0
hits_9                 0
homeruns_9             0
walks_9                0
strikeouts_9           0
strikeout_walk         0
dtype: int64
```

Renaming Columns:

In [451…
```
# Reading the Columns in the Data.
df.columns
```

Out[451…
```
Index(['id', 'year', 'team', 'average_age', 'runs_per_game', 'wins', 'losses',
       'win_loss_percentage', 'ERA', 'run_average_9', 'games', 'games_started',
       'games_finished', 'complete_game', 'shutouts', 'saves',
       'innings_pitched', 'hits', 'runs', 'earned_runs', 'home_runs', 'walks',
       'intentional_walks', 'strikeouts', 'hit_batter', 'balks',
       'wild_pitches', 'batters_faced', 'WHIP', 'hits_9', 'homeruns_9',
       'walks_9', 'strikeouts_9', 'strikeout_walk'],
      dtype='object')
```

In [452…
```
# Renaming the Columns.
df.rename(columns = {'id': 'ID', 'year': 'Year', 'team': 'Team', 'average_age': 'Av
       'win_loss_percentage': 'Win_Ratio', 'run_average_9': 'Run_Average_9', 'games
       'games_finished': 'Games_Finished', 'complete_game': 'Complete_Game', 'shuto
```

```
        'innings_pitched': 'Innings_Pitched', 'hits': 'Hits', 'runs': 'Runs', 'earne
        'intentional_walks': 'Intentional_Walks', 'strikeouts': 'Strikeouts', 'hit_b
        'wild_pitches': 'Wild_Pitches', 'batters_faced': 'Batters_Faced', 'hits_9':
        'walks_9': 'Walks_9', 'strikeouts_9': 'Strikeouts_9', 'strikeout_walk': 'Str
```

In [453…

```
# Showing to make sure the Columns got renamed.

df.head()
```

Out[453…

	ID	Year	Team	Average_Age	Runs_Per_Game	Wins	Losses	Win_Ratio	ERA	Run_
0	1	2021	LG Twins	26.3	3.90	72	57	0.558	3.57	
1	2	2021	KT Wiz	28.4	4.06	75	59	0.560	3.67	
2	3	2021	Doosan Bears	27.5	4.57	70	65	0.519	4.28	
3	4	2021	Samsung Lions	28.8	4.57	75	59	0.560	4.29	
4	5	2021	NC Dinos	27.7	4.80	67	67	0.500	4.50	

5 rows × 34 columns

Average ERA Across Teams:

In [454…

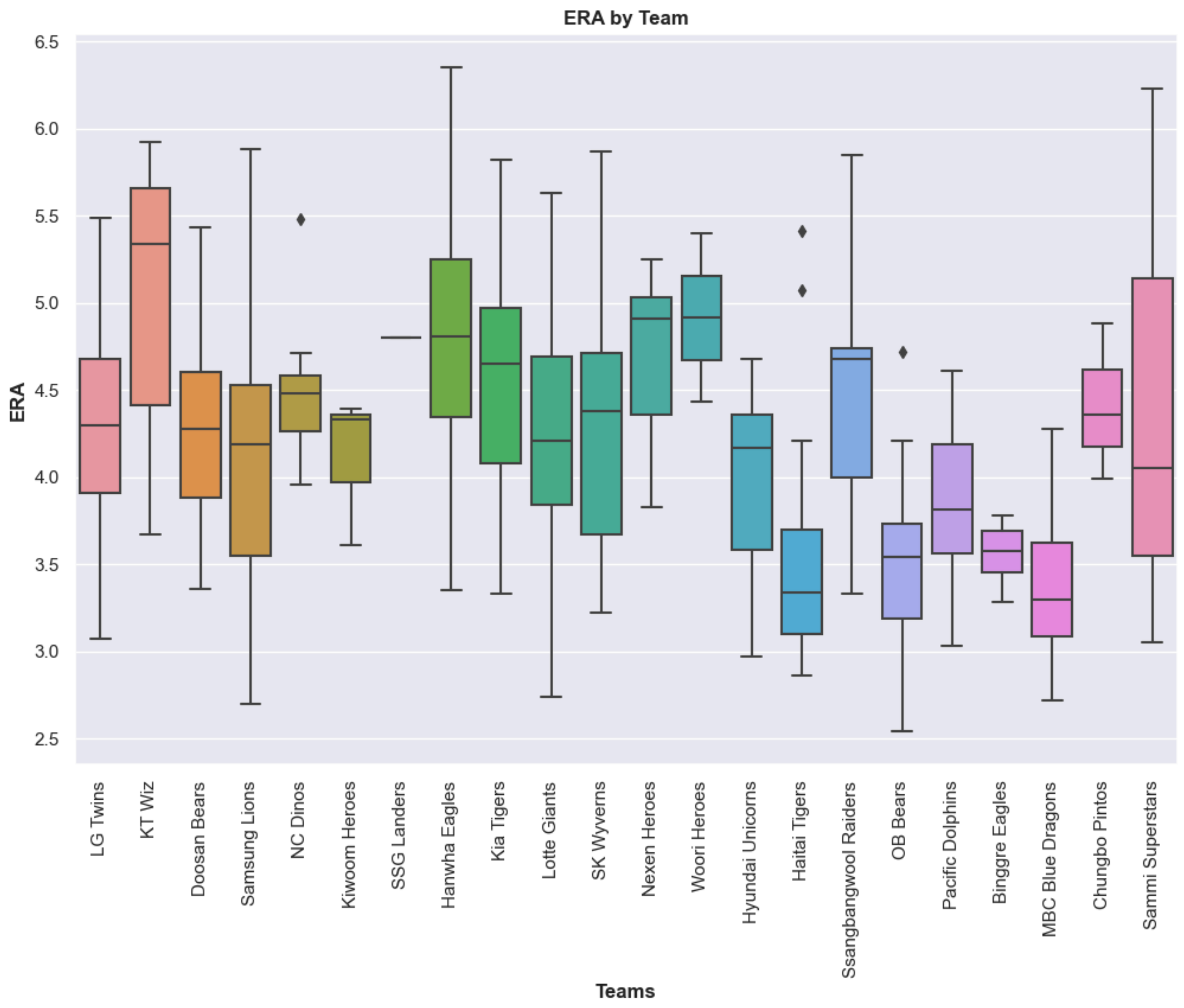
```
# This groupby operation calculates the average ERA for each team.
avg_era_team = df.groupby('Team')['ERA'].mean()

avg_era_team
```

Out[454…
```
Team
Binggre Eagles          3.561250
Chungbo Pintos          4.410000
Doosan Bears            4.252609
Haitai Tigers           3.533158
Hanwha Eagles           4.746429
Hyundai Unicorns        3.965833
KT Wiz                  5.010000
Kia Tigers              4.554286
Kiwoom Heroes           4.110000
LG Twins                4.289688
Lotte Giants            4.238500
MBC Blue Dragons        3.380000
NC Dinos                4.474444
Nexen Heroes            4.676667
OB Bears                3.512941
Pacific Dolphins        3.865000
SK Wyverns              4.286190
SSG Landers             4.800000
Sammi Superstars        4.443333
Samsung Lions           4.103000
Ssangbangwool Raiders   4.444444
Woori Heroes            4.915000
Name: ERA, dtype: float64
```

In [455…
```
# Calculate the average ERA across teams
average_era = df['ERA'].mean()

print(f"Average ERA across teams: {average_era:.2f}")
```

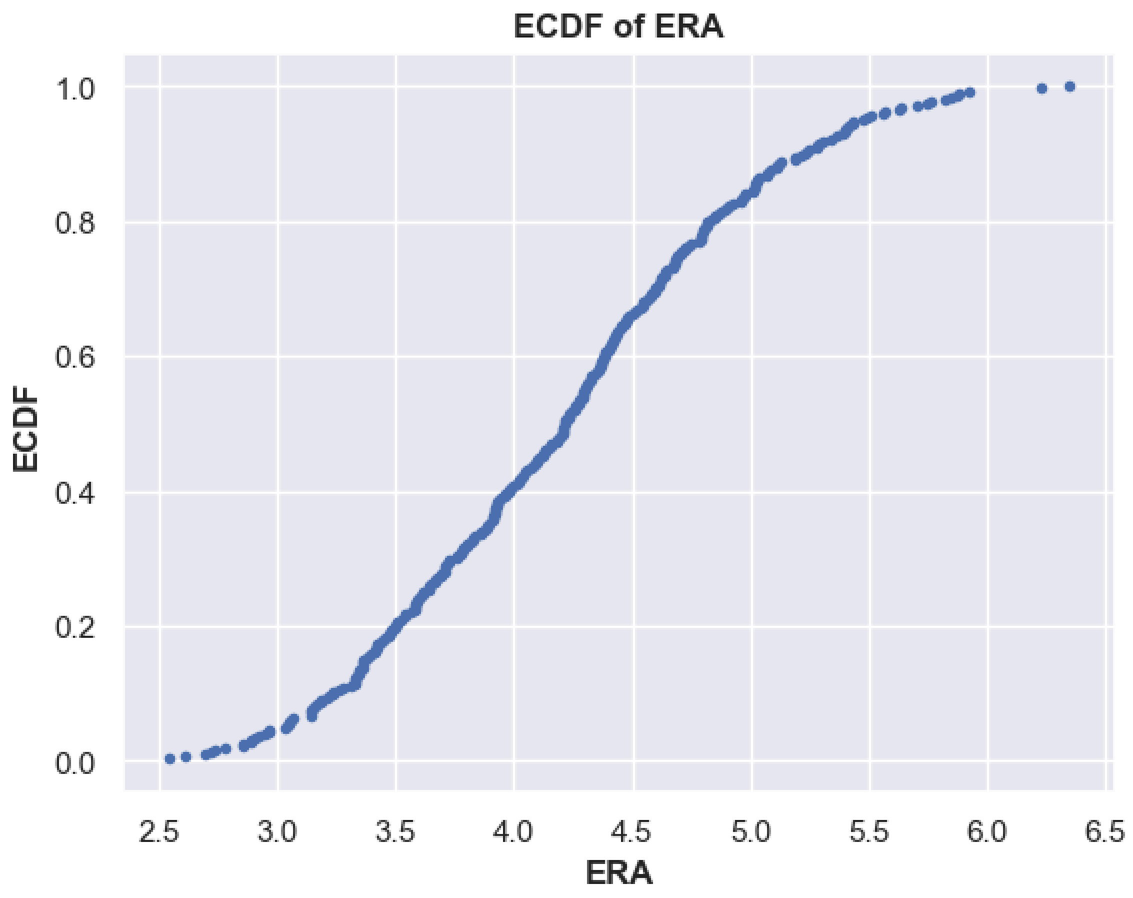
```
Average ERA across teams: 4.21
```

In [456…
```
# Boxplot of ERA.

plt.figure(figsize=(12, 8))
sns.boxplot(x ='Team', y ='ERA', data=df)
plt.title('ERA by Team', weight='bold')
plt.xlabel('Teams', weight='bold')
plt.ylabel('ERA', weight='bold')
plt.xticks(rotation = 90)
plt.show()
```



In [457…

```python
# Empirical Commutative Distribution plot.

# The ECDF plot shows the cumulative distribution of ERA values.
def ecdf(data):
    x = np.sort(data)
    y = np.arange(1, len(data) + 1) / len(data)
    return x, y

x, y = ecdf(df['ERA'])
plt.plot(x, y, marker='.', linestyle='none')
plt.title('ECDF of ERA', weight='bold')
plt.xlabel('ERA', weight='bold')
plt.ylabel('ECDF', weight='bold')
plt.show()
```

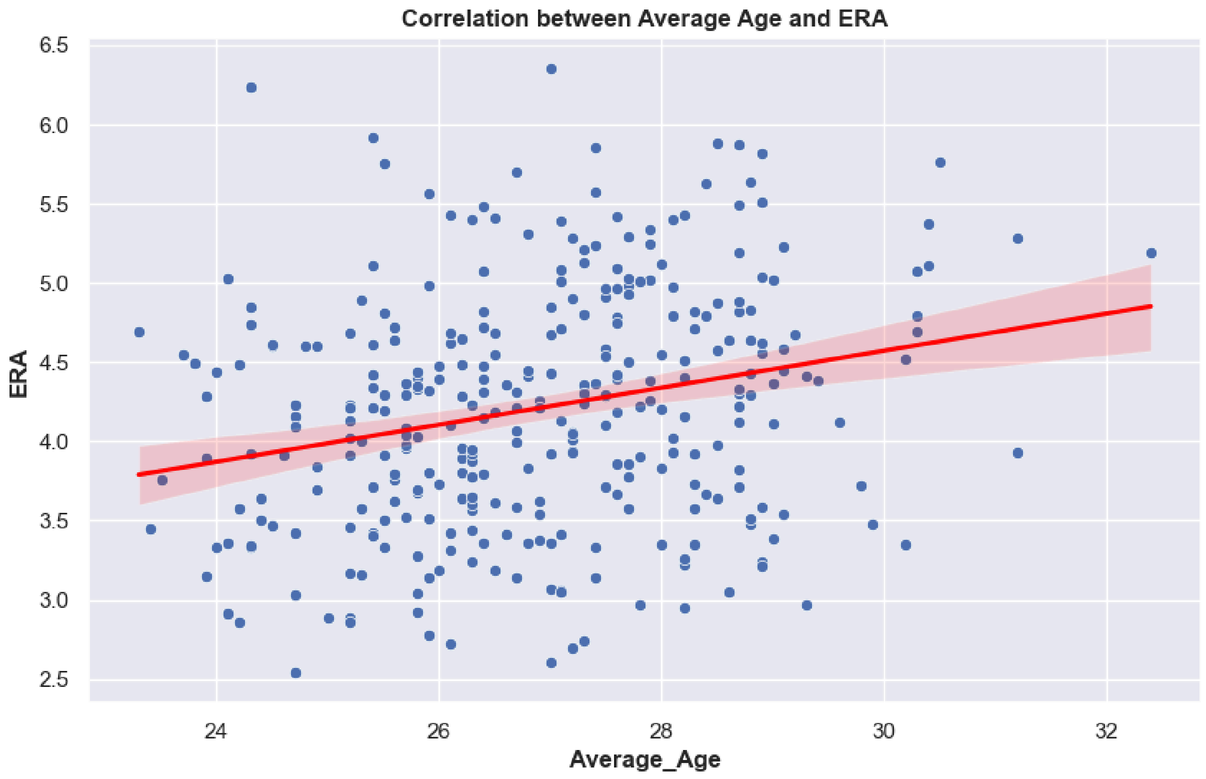


Correlation between the age of players and their ERA:

In [458…

```
# Calculate the correlation between the average age of players and their ERA
correlation_age_era = df['Average_Age'].corr(df['ERA'])
correlation_age_era_rounded = round(correlation_age_era, 2)
print(f"Correlation between Average Age and ERA: {correlation_age_era_rounded}")

# Scatter plot to visualize the correlation between Average Age and ERA
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Average_Age', y='ERA', data=df)
plt.title('Correlation between Average Age and ERA', weight='bold')
plt.xlabel('Average Age', weight='bold')
plt.ylabel('ERA', weight='bold')

# Plotting the regression line
sns.regplot(x='Average_Age', y='ERA', data=df, scatter=False, color='red')
plt.show()
```

```
Correlation between Average Age and ERA: 0.25
```



Highest Average Runs Per Game:

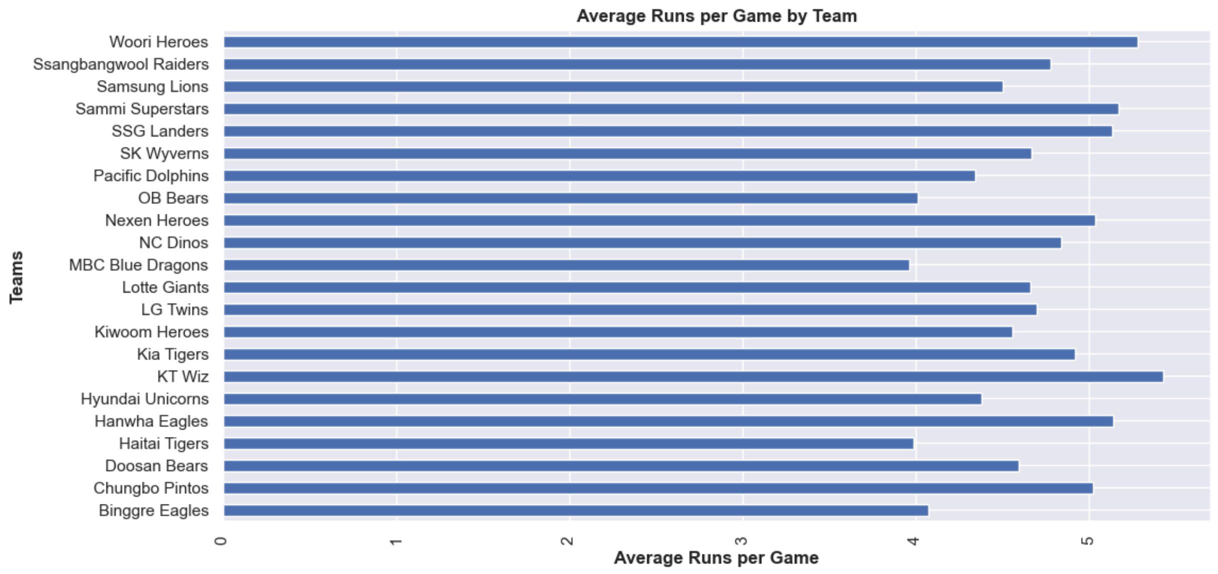
In [459…

```
# Calculate the team with the highest average runs per game
highest_runs_per_game_team = df.groupby('Team')['Runs_Per_Game'].mean().idxmax()
highest_runs_per_game_value = df.groupby('Team')['Runs_Per_Game'].mean().max()

print(f"Team with highest average runs per game: {highest_runs_per_game_team} with
```

```
Team with highest average runs per game: KT Wiz with an average of 5.43 runs per gam
e
```

In [460…

```
# Calculate the average runs per game for each team
average_runs_per_game = df.groupby('Team')['Runs_Per_Game'].mean()

# Creating a bar plot
plt.figure(figsize = (12, 6))
average_runs_per_game.plot(kind = 'barh', color = 'b' )
plt.title('Average Runs per Game by Team', weight='bold')
plt.xlabel('Average Runs per Game', weight='bold')
plt.ylabel('Teams', weight='bold')
plt.xticks(rotation = 90)
plt.show()
```



Hits Per 9 Innings:

In [461…

```
# Calculate the average strikeouts per 9 innings for each team
avg_hits_9 = df.groupby('Team')['Hits_9'].mean()

avg_hits_9
```

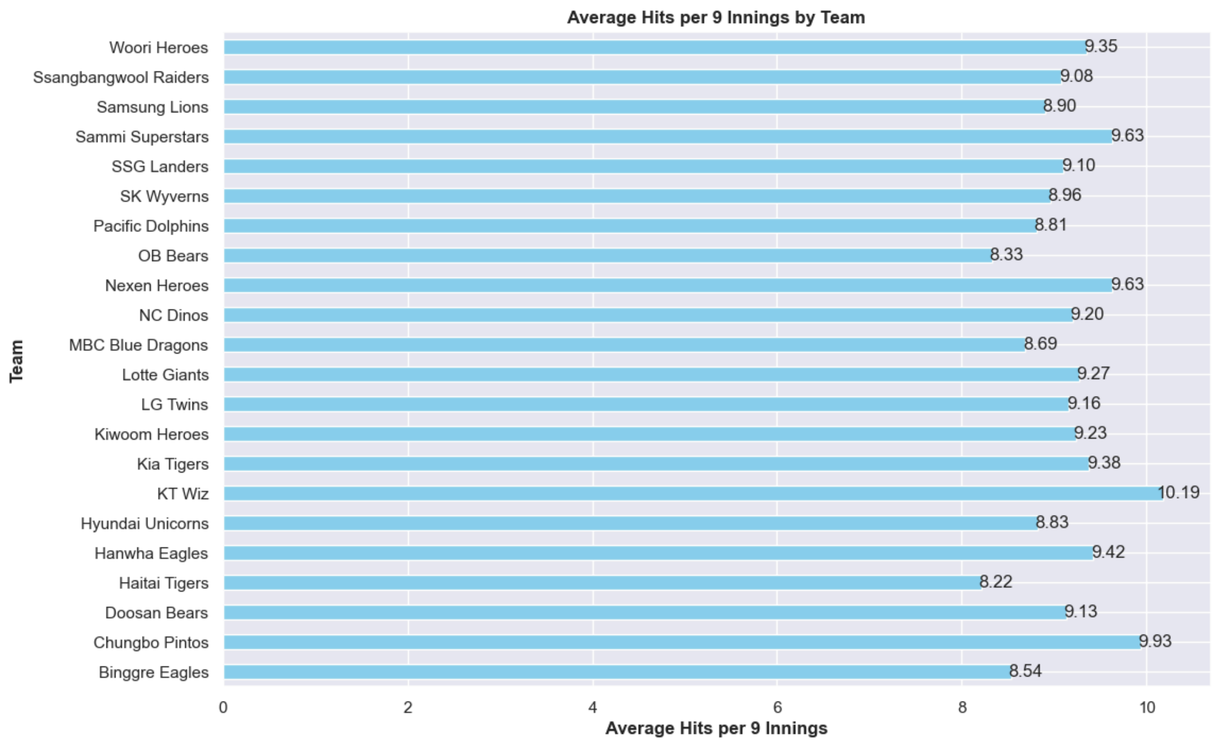
Out[461…

```
Team
Binggre Eagles            8.537500
Chungbo Pintos            9.933333
Doosan Bears              9.130435
Haitai Tigers             8.215789
Hanwha Eagles             9.425000
Hyundai Unicorns          8.825000
KT Wiz                   10.185714
Kia Tigers                9.376190
Kiwoom Heroes             9.233333
LG Twins                  9.159375
Lotte Giants              9.270000
MBC Blue Dragons          8.687500
NC Dinos                  9.200000
Nexen Heroes              9.633333
OB Bears                  8.329412
Pacific Dolphins          8.812500
SK Wyverns                8.961905
SSG Landers               9.100000
Sammi Superstars          9.633333
Samsung Lions             8.902500
Ssangbangwool Raiders     9.077778
Woori Heroes              9.350000
Name: Hits_9, dtype: float64
```

In [462…

```
# Calculate the average hits per 9 innings for each team
avg_hits_9 = df.groupby('Team')['Hits_9'].mean()

# Create a horizontal bar plot to visualize the average hits per 9 innings for each
```

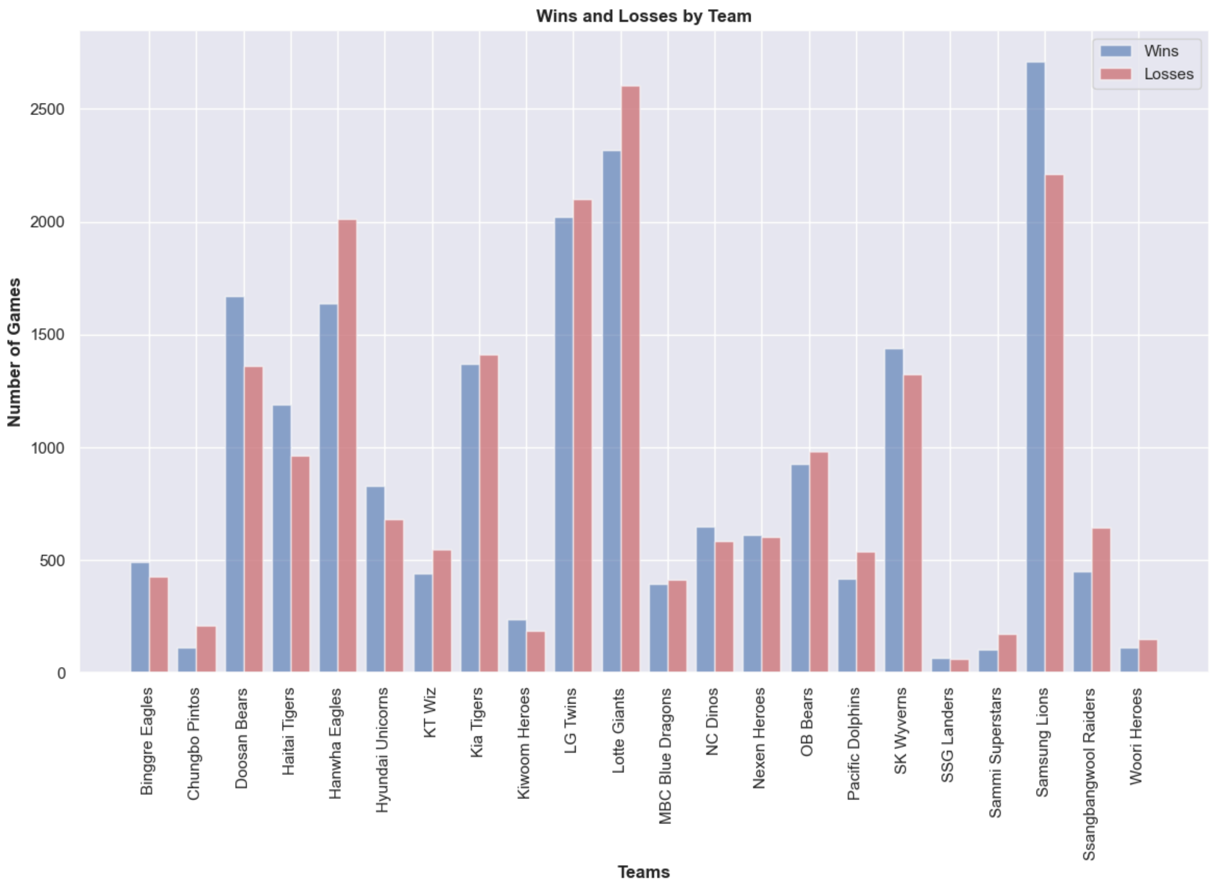
```
plt.figure(figsize=(12, 8))
ax = avg_hits_9.plot(kind='barh', color='skyblue')

# Add the hits per 9 innings values on top of the bars
for p in ax.patches:
    ax.annotate(f'{p.get_width():.2f}', (p.get_width(), p.get_y() + p.get_height()
                ha='center', va='center', xytext=(10, 0), textcoords='offset points

plt.title('Average Hits per 9 Innings by Team', weight='bold')
plt.xlabel('Average Hits per 9 Innings', weight='bold')
plt.ylabel('Team', weight='bold')
plt.show()
```



Win and Losses by Teams:

In [463…

```
# Prepare the data for plotting
wins_losses = df.groupby('Team')[['Wins', 'Losses']].sum().reset_index()

# Bar plot to show wins and losses grouped by each team
fig, ax = plt.subplots(figsize=(14, 8))

# Set the positions and width for the bars
positions = np.arange(len(wins_losses))
bar_width = 0.4

# Plot wins
ax.bar(positions - bar_width/2, wins_losses['Wins'], width=bar_width, label='Wins',

# Plot losses next to wins
ax.bar(positions + bar_width/2, wins_losses['Losses'], width=bar_width, label='Loss
ax.set_title('Wins and Losses by Team', weight='bold')
ax.set_xlabel('Teams', weight='bold')
ax.set_ylabel('Number of Games', weight='bold')
ax.set_xticks(positions)
ax.set_xticklabels(wins_losses['Team'])
ax.legend()

plt.xticks(rotation=90)
plt.show()
```



In [464…

```
# This pivot table summarizes the total wins and losses for each team.
pivot_wins_losses = df.pivot_table(index ='Team', values = ['Wins', 'Losses'], aggf

pivot_wins_losses
```

Out[464…

Team	Losses	Wins
Binggre Eagles	428	493
Chungbo Pintos	209	112
Doosan Bears	1359	1668
Haitai Tigers	964	1189
Hanwha Eagles	2010	1637
Hyundai Unicorns	680	830
KT Wiz	548	441
Kia Tigers	1412	1368
Kiwoom Heroes	187	235
LG Twins	2101	2023
Lotte Giants	2605	2317
MBC Blue Dragons	412	394
NC Dinos	582	649
Nexen Heroes	602	613
OB Bears	983	927
Pacific Dolphins	536	415
SK Wyverns	1325	1437
SSG Landers	63	66
Sammi Superstars	171	105
Samsung Lions	2211	2712
Ssangbangwool Raiders	646	449
Woori Heroes	148	110

Average Strikeouts Per 9 Innings:

In [477…

```
# Calculate the average strikeouts per 9 innings for each team
avg_strikeouts_9 = df.groupby('Team')['Strikeouts_9'].mean()

avg_strikeouts_9
```

Out[477…

```
Team
Binggre Eagles           4.437500
Chungbo Pintos           3.500000
Doosan Bears             6.539130
Haitai Tigers            5.631579
Hanwha Eagles            6.525000
Hyundai Unicorns         6.533333
KT Wiz                   6.771429
Kia Tigers               6.580952
Kiwoom Heroes            6.666667
LG Twins                 6.003125
Lotte Giants             5.695000
MBC Blue Dragons         3.700000
NC Dinos                 7.266667
Nexen Heroes             6.811111
OB Bears                 4.582353
Pacific Dolphins         4.875000
SK Wyverns               6.819048
SSG Landers              7.100000
Sammi Superstars         3.600000
Samsung Lions            5.915000
Ssangbangwool Raiders    5.188889
Woori Heroes             6.100000
Name: Strikeouts_9, dtype: float64
```

In [466…

```
# This pivot table provides the average ERA and strikeouts per 9 innings for each t
pivot_era_strikeouts = df.pivot_table(index='Team', values=['ERA', 'Strikeouts_9'],

pivot_era_strikeouts
```

Out[466…

Team	ERA	Strikeouts_9
Binggre Eagles	3.561250	4.437500
Chungbo Pintos	4.410000	3.500000
Doosan Bears	4.252609	6.539130
Haitai Tigers	3.533158	5.631579
Hanwha Eagles	4.746429	6.525000
Hyundai Unicorns	3.965833	6.533333
KT Wiz	5.010000	6.771429
Kia Tigers	4.554286	6.580952
Kiwoom Heroes	4.110000	6.666667
LG Twins	4.289688	6.003125
Lotte Giants	4.238500	5.695000
MBC Blue Dragons	3.380000	3.700000
NC Dinos	4.474444	7.266667
Nexen Heroes	4.676667	6.811111
OB Bears	3.512941	4.582353
Pacific Dolphins	3.865000	4.875000
SK Wyverns	4.286190	6.819048
SSG Landers	4.800000	7.100000
Sammi Superstars	4.443333	3.600000
Samsung Lions	4.103000	5.915000
Ssangbangwool Raiders	4.444444	5.188889
Woori Heroes	4.915000	6.100000

In [467…

```
# The correlation matrix reveals the relationships between different performance me

df_selected = df[['ERA', 'Average_Age', 'Runs_Per_Game', 'Wins', 'Losses', 'Strikeo

correlation_matrix = df_selected.corr(method='spearman').round(2)

correlation_matrix.style.background_gradient(cmap='Reds')
```

Out[467…

	ERA	Average_Age	Runs_Per_Game	Wins	Losses	Strikeouts_9
ERA	1.000000	0.260000	0.980000	-0.140000	0.710000	0.43000
Average_Age	0.260000	1.000000	0.230000	0.260000	0.110000	0.38000
Runs_Per_Game	0.980000	0.230000	1.000000	-0.200000	0.700000	0.37000
Wins	-0.140000	0.260000	-0.200000	1.000000	-0.490000	0.51000
Losses	0.710000	0.110000	0.700000	-0.490000	1.000000	0.23000
Strikeouts_9	0.430000	0.380000	0.370000	0.510000	0.230000	1.00000
WHIP	0.910000	0.250000	0.920000	-0.200000	0.670000	0.30000
Homeruns_9	0.790000	0.230000	0.770000	0.060000	0.440000	0.50000

Top 5 teams with the most strikeouts:

In [468…

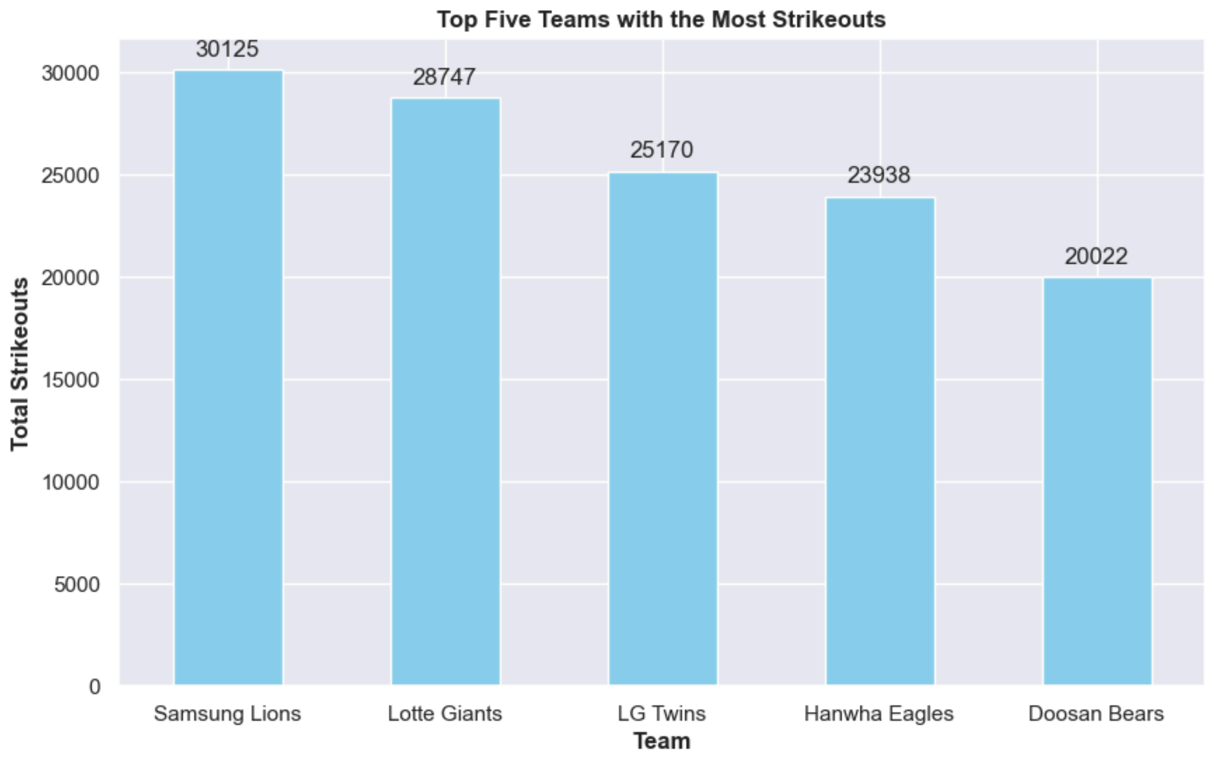
```python
# Calculate the total number of strikeouts for each team
total_strikeouts = df.groupby('Team')['Strikeouts'].sum()

# Get the top five teams with the most strikeouts
top_five_teams_strikeouts = total_strikeouts.nlargest(5)

top_five_teams_strikeouts
```

Out[468…

```
Team
Samsung Lions    30125
Lotte Giants     28747
LG Twins         25170
Hanwha Eagles    23938
Doosan Bears     20022
Name: Strikeouts, dtype: int64
```

In [469…

```python
# Calculate the total number of strikeouts for each team
total_strikeouts = df.groupby('Team')['Strikeouts'].sum()

# Get the top five teams with the most strikeouts
top_five_teams_strikeouts = total_strikeouts.nlargest(5)

# Create a bar plot to visualize the top five teams with the most strikeouts
plt.figure(figsize=(10, 6))
ax = top_five_teams_strikeouts.plot(kind='bar', color='skyblue')

# Add the strikeout numbers on top of the bars
for p in ax.patches:
    ax.annotate(f'{int(p.get_height())}', (p.get_x() + p.get_width() / 2., p.get_he
                ha='center', va='center', xytext=(0, 10), textcoords='offset points

plt.title('Top Five Teams with the Most Strikeouts', weight='bold')
plt.xlabel('Team', weight='bold')
plt.ylabel('Total Strikeouts', weight='bold')
plt.xticks(rotation = 0)
plt.show()
```



Team with the highest Walks plus Hits per inning Pitched(WHIP):

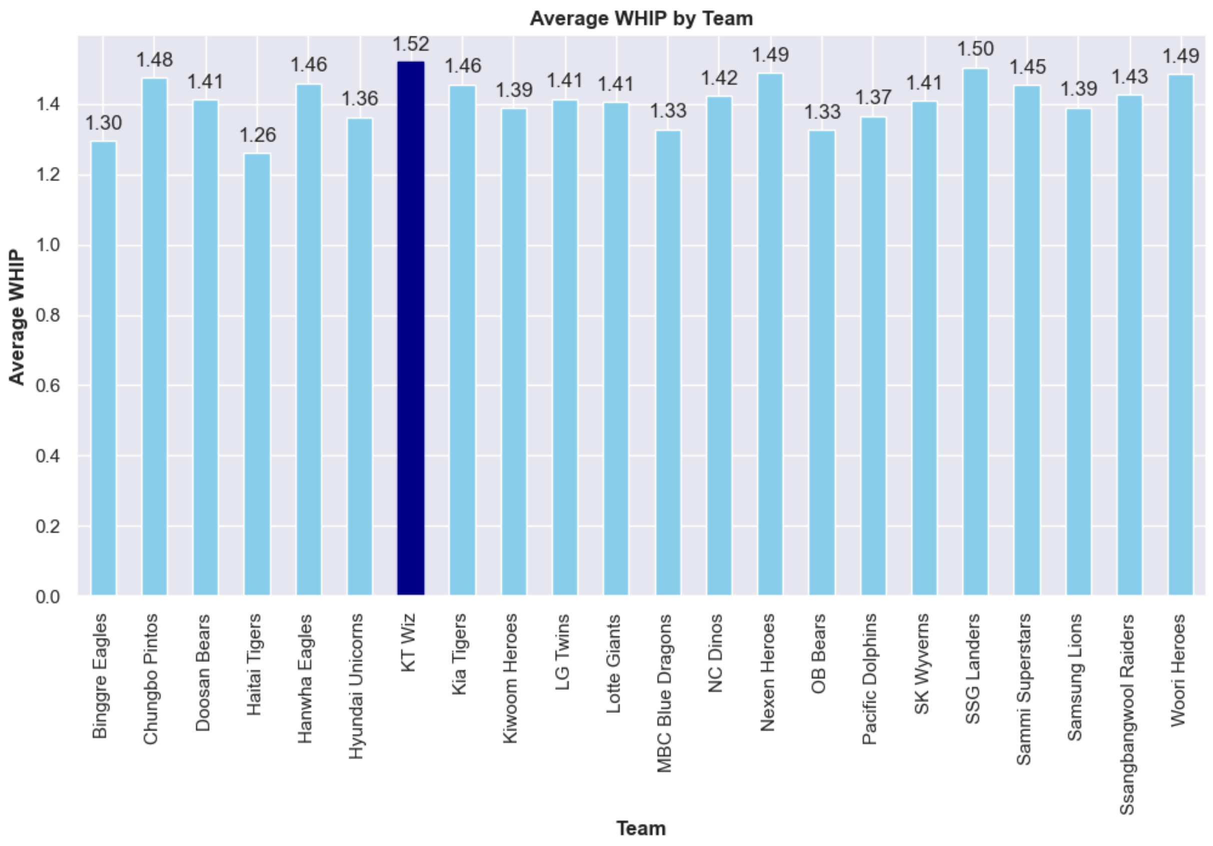
In [470…

```
# Calculate the average WHIP for each team
avg_whip = df.groupby('Team')['WHIP'].mean()
team_highest_whip = avg_whip.idxmax()
highest_whip_value = avg_whip.max()

print(f"Team with highest WHIP: {team_highest_whip} with a WHIP of {highest_whip_va
```

```
Team with highest WHIP: KT Wiz with a WHIP of 1.52
```

In [471…

```
# Create a bar plot to visualize the average WHIP for each team
plt.figure(figsize=(12, 6))
ax = avg_whip.plot(kind='bar', color='skyblue')

# Highlight the team with the highest WHIP
ax.patches[avg_whip.index.get_loc(team_highest_whip)].set_color('darkblue')

# Add the WHIP values on top of the bars
for p in ax.patches:
    ax.annotate(f'{p.get_height():.2f}', (p.get_x() + p.get_width() / 2., p.get_hei
                ha='center', va='center', xytext=(0, 10), textcoords='offset points

plt.title('Average WHIP by Team', weight='bold')
plt.xlabel('Team', weight='bold')
plt.ylabel('Average WHIP', weight='bold')
plt.xticks(rotation=90)
plt.show()
```



Highest average of Homeruns per 9 innings by a team:

In [482…

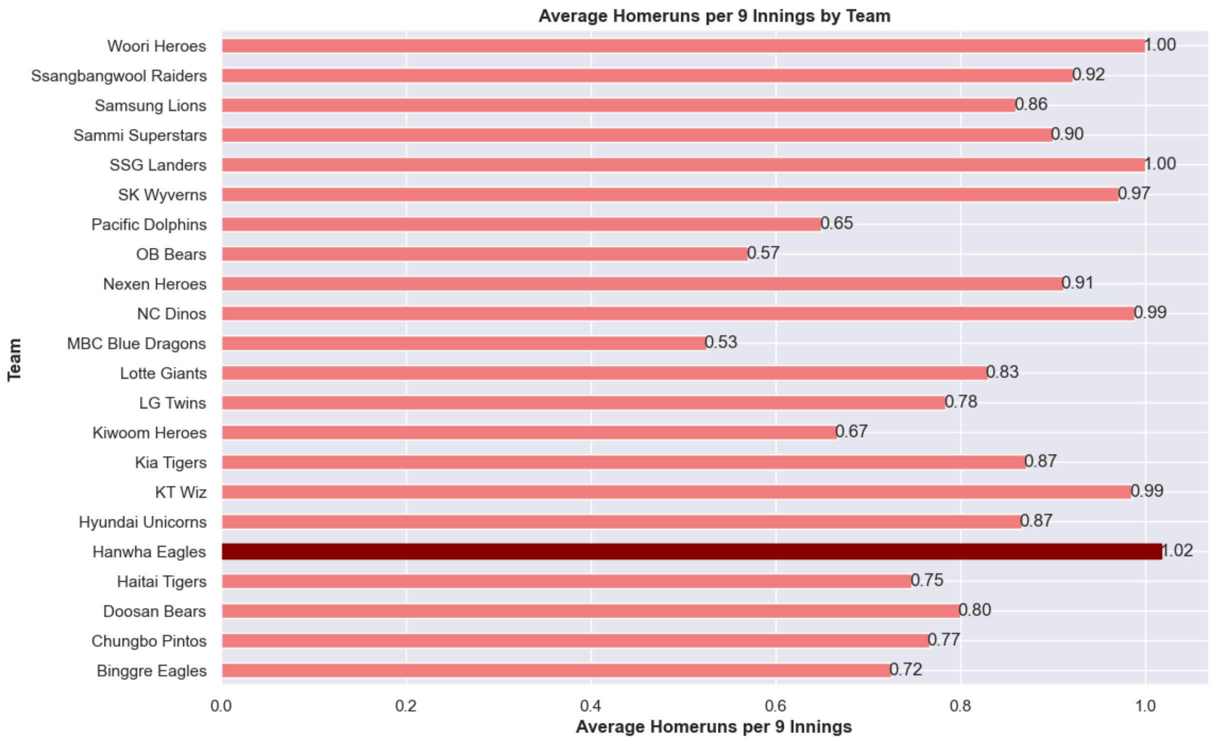
```
# Calculate the average homeruns per 9 innings for each team
avg_homeruns_9 = df.groupby('Team')['Homeruns_9'].mean()

# Identify the team that gave up the most homeruns per 9 innings
team_most_homeruns_9 = avg_homeruns_9.idxmax()
most_homeruns_9_value = avg_homeruns_9.max()

print(f"Team that gave up the most homeruns per 9 innings: {team_most_homeruns_9} w
```

```
Team that gave up the most homeruns per 9 innings: Hanwha Eagles with an average of
1.02 homeruns per 9 innings
```

In [481…

```
# Calculate the average homeruns per 9 innings for each team
avg_homeruns_9 = df.groupby('Team')['Homeruns_9'].mean()

# Identify the team that gave up the most homeruns per 9 innings
team_most_homeruns_9 = avg_homeruns_9.idxmax()
most_homeruns_9_value = avg_homeruns_9.max()

# Create a horizontal bar plot to visualize the average homeruns per 9 innings for
plt.figure(figsize=(12, 8))
ax = avg_homeruns_9.plot(kind='barh', color='lightcoral')
```

```
# Highlight the team with the most homeruns per 9 innings
ax.patches[avg_homeruns_9.index.get_loc(team_most_homeruns_9)].set_color('darkred')

# Add the homeruns per 9 innings values on top of the bars
for p in ax.patches:
    ax.annotate(f'{p.get_width():.2f}', (p.get_width(), p.get_y() + p.get_height()
                ha='center', va='center', xytext=(10, 0), textcoords='offset points

plt.title('Average Homeruns per 9 Innings by Team', weight='bold')
plt.xlabel('Average Homeruns per 9 Innings', weight='bold')
plt.ylabel('Team', weight='bold')
plt.show()
```



Distribution of wild pitches and hit batters:

In [474…

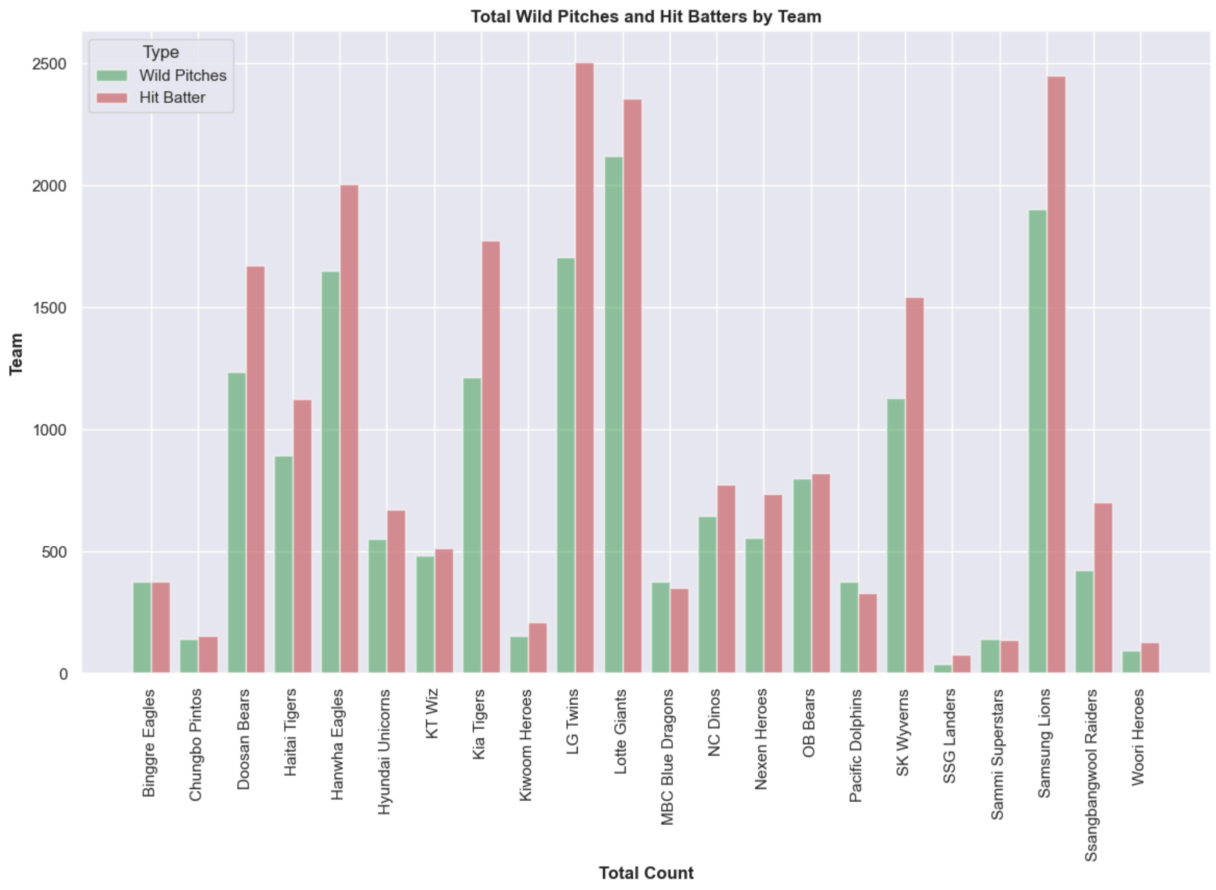
```
# Calculate the total number of wild pitches and hit batters for each team
wild_pitches_hit_batters = df.groupby('Team')[['Wild_Pitches', 'Hit_Batter']].sum()

wild_pitches_hit_batters
```

Out[474…

Team	Wild_Pitches	Hit_Batter
Binggre Eagles	376.0	378
Chungbo Pintos	141.0	153
Doosan Bears	1236.0	1670
Haitai Tigers	893.0	1126
Hanwha Eagles	1651.0	2006
Hyundai Unicorns	554.0	673
KT Wiz	484.0	512
Kia Tigers	1214.0	1772
Kiwoom Heroes	154.0	211
LG Twins	1707.0	2506
Lotte Giants	2121.0	2355
MBC Blue Dragons	376.0	352
NC Dinos	646.0	775
Nexen Heroes	558.0	734
OB Bears	799.0	821
Pacific Dolphins	376.0	328
SK Wyverns	1128.0	1544
SSG Landers	40.0	78
Sammi Superstars	141.0	136
Samsung Lions	1903.0	2450
Ssangbangwool Raiders	423.0	702
Woori Heroes	97.0	131

In [475…

```python
# Prepare the data for plotting
wild_pitches_hit_batters = df.groupby('Team')[['Wild_Pitches', 'Hit_Batter']].sum()

# Bar plot to show wild pitches and hit batter grouped by each team
fig, ax = plt.subplots(figsize=(14, 8))

# Set the positions and width for the bars
positions = np.arange(len(wild_pitches_hit_batters))
bar_width = 0.4

# Plot Wild_Pitches
ax.bar(positions - bar_width/2, wild_pitches_hit_batters['Wild_Pitches'], width=bar
# Plot Hit_Batter
ax.bar(positions + bar_width/2, wild_pitches_hit_batters['Hit_Batter'], width=bar_w

ax.set_title('Total Wild Pitches and Hit Batters by Team', weight='bold')
ax.set_xlabel('Total Count', weight='bold')
ax.set_ylabel('Team', weight='bold')
ax.set_xticks(positions)
ax.set_xticklabels(wild_pitches_hit_batters['Team'])
ax.legend(title='Type', labels=['Wild Pitches', 'Hit Batter'])

plt.xticks(rotation=90)
plt.show()
```



One-Sample Z-Test:

```
In [489…
# Parameters for hypothesis testing
population_mean = 4.21
sample_mean = df['ERA'].mean()
sample_std = df['ERA'].std()
sample_size = len(df['ERA'])

# Calculate the Z-score
z_score = (sample_mean - population_mean) / (sample_std / np.sqrt(sample_size))

# Calculate the p-value
p_value = stats.norm.sf(abs(z_score)) * 2
```

```
print(f"Z-score: {z_score:.2f}, P-value: {p_value:.2f}")
```

```
Z-score: -0.05, P-value: 0.96
```

Summary:

1. **What is the average ERA (Earned Run Average) across teams?**

The average ERA across teams is 4.21. 2. #### Is there a correlation between the age of players and their ERA? The correlation coefficient between the average age of players and their ERA is approximately 0.25, indicating a weak positive correlation. 3. #### Which team has the highest average runs per game? The team with the highest average runs per game is the NC Dinos with an average of approximately 5.43 runs per game. 4. #### What is the average hits per 9 innings (H/9) for each team? The average hits per 9 innings for each team but I only going to put the top five only are the KT Wiz at 10.19, Chungbo Pintos at 9.93, Nexen Heroes at 9.63, Sammi Superstar at 9.63, and the Kia Tigers at 9.38. 5. #### What is the distribution of wins and losses across teams? The distribution of wins and losses show that Samsung Lions are the number one team with the most wins and the last place team are Hanwa Eagles with the most losses through years of 1982 - 2021. 6. #### What is the average strikeouts per 9 innings (K/9) for each team? The average strikeouts per 9 innings for each team but I only going to put the top five only are the NC Dinos at 7.3, SSG Landers at 7.1, SK Wyverns at 6.8, at KT Wiz 6.8, and Kiwoom Heros at 6.7. 7. #### What are the top five teams with the most strikeouts? The top five are Samsung Lions at 30125, Lotte Giants at 28747, LG Twins at 25170, Hanwha Eagles at 23938, and Doosan Bears at 20022. 8. #### Which team has the highest WHIP (Walks plus Hits per Inning Pitched) on average? The team with highest WHIP is the KT Wiz with a WHIP of 1.52. 9. #### What team has the highest average of homeruns per 9 innings? The team that has the highest average homeruns per 9 innings is the Hanwha Eagles with an average of 1.02. 10. #### What is the distribution of wild pitches and hit batters across different teams? The LG Twins had the most hit batters and the Lottel Giants had the most wild pitches.

```
In [ ]:
```