TABLE OF CONTENTS

I. Identify the attribute type of each attribute 1

II. Identify the values of the summarizing properties for each attribute 3

1. RA_ICRS, DE_ICRS 3
2. Plx, PM 4
3. pmRA, pmDE 5
4. Gmag, BPmag, RPmag, GRVSmag 5
5. e_Gmag, e_BPmag, e_RPmag, e_GRVSmag 6
6. BP-RP, BP-G, G-RP 7
7. pscol 8
8. Teff 8
9. Dist 9
10. Rad, Lum-Flame, Mass-Flame, Age-Flame 10
11. z-Flame 12
12. SpType-ELS 12

III. Explore multiple attributes relationship 13

1. Teff – Interesting Attribute 14
2. Dist Related Relationships 15
3. Physical Characteristics Relationships 16
4. Brightness' Relationships 16

IV. Smoothing for RA_ICRS and DE_ICRS 17

1. Equi-width Binning 17
2. Equi-depth Binning 19

V. Summary 22

1. Attribute Findings 22
2. Relationship Findings 22

```python
utils.py > detect_outliers
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec
import seaborn as sns


def detect_outliers(feature):
    if not np.issubdtype(feature.dtype, np.number): return None, None, None
    Q1, Q3 = np.nanpercentile(feature, [25, 75]) # Calculate the 1st and 3rd quartiles without
    IQR = Q3 - Q1

    # Define outliers as those beyond 1.5 * IQR from the Q1 and Q3
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    is_outliers = (feature < lower_bound) | (feature > upper_bound)
    return is_outliers, lower_bound, upper_bound


def stats_summary(df):
    summary = df.describe(include=[np.number]).T # Calculate descriptive statistics for all nu
    summary['variance'] = df.var(numeric_only=True) # Variance = E[(X - E[X])^2]
    summary['iqr_size'] = df.quantile(0.75, numeric_only=True) - df.quantile(0.25, numeric_onl
    summary['skewness'] = df.skew(numeric_only=True) # Skew > 0 ⇒ Right-skewed distribution
    summary['kurtosis'] = df.kurtosis(numeric_only=True) # Kurtosis > 0 ⇒ This distribution i
    summary['nulls_count'] = df.isnull().sum() # Checking for null values to identify attribut
    summary['outliers_count'] = df.select_dtypes(include=[np.number]).apply(lambda col: detect_
    summary['nulls_percent'] = summary['nulls_count'] * 100 / df.shape[0]
    summary['outliers_percent'] = summary['outliers_count'] * 100 / summary['count']
    return summary


def hist_box_plot(figsize, X, list_of_cols, ncols=2, xlabel='Data Values', ylabel='Frequency',
    fig = plt.figure(figsize=figsize)
```

```python
utils.py > hist_box_plot
def hist_box_plot(figsize, X, list_of_cols, ncols=2, xlabel='Data Values', ylabel='Frequency',
    fig = plt.figure(figsize=figsize)
    outer_grid = gridspec.GridSpec(len(list_of_cols) // ncols + 1, ncols)

    for i, col_name in enumerate(list_of_cols):
        # For each cell in the 3x2 grid, create a nested grid for the histogram and boxplot
        inner_grid = outer_grid[i].subgridspec(2, 1, height_ratios=[5, 1], hspace=0)
        ax_hist = fig.add_subplot(inner_grid[0])
        ax_box = fig.add_subplot(inner_grid[1])

        sns.histplot(X[col_name], ax=ax_hist, kde=True, color="skyblue", edgecolor='red')
        ax_hist.set_xlabel('')  # Remove the x-axis label
        ax_hist.set_xticks([]) # Remove x-ticks for histogram
        ax_hist.set_ylabel(ylabel)
        ax_hist.yaxis.grid(True, linestyle='--', which='major', color='lightgrey', alpha=0.7)
        ax_hist.set_title(f'Distribution with Histogram and Boxplot for {col_name}', fontweight

        _, lower_bound, upper_bound = detect_outliers(X[col_name])
        sns.boxplot(x=X[col_name], ax=ax_box, width=0.5, color="lightgreen", fliersize=5, line

        # Show the range considered as outliers
        plt.axvspan(xmin=lower_bound, xmax=upper_bound, color='green', alpha=0.2)
        plt.axvspan(xmin=X[col_name].min(), xmax=lower_bound, color='red', alpha=0.3)
        plt.axvspan(xmin=upper_bound, xmax=X[col_name].max(), color='red', alpha=0.3)

        ax_box.set_xlabel(xlabel)
        ax_box.set_ylabel('')
        ax_box.yaxis.grid(True, linestyle='--', which='major', color='lightgrey', alpha=0.7)

    if plt_show:
        plt.tight_layout()
        plt.show()
    return fig, outer_grid
```

*This **utils.py** file contains utility functions for below data visualization and statistical analysis.*

I. Identify the attribute type of each attribute

```python
missing_values = ['?', '.', '', ' ', '_', 'Na', 'NULL', 'null', 'not', 'Not', 'NaN', 'NA', '??', 'nan', 'inf']
raw_data = pd.read_csv('32130_AT2_25076833.csv', na_values=missing_values)
raw_data = raw_data.loc[:, ~raw_data.columns.str.contains('^Unnamed|Source')] # Drop 3 unnecessary columns
print('Number of entries:', raw_data.shape[0])
print('Number of attributes:', raw_data.shape[1])
```

✓ 0.0s

```
Number of entries: 3000
Number of attributes: 26
```

The dataset contains 26 attributes, excluding the **Source & 2 Unnamed** columns as they have no specific astronomical relevance.

RA_ICRS	*Quantitative, Interval*	Position on celestial sphere, implying direction but no true zero point (0^o isn't absence of angle - the zero is arbitrarily defined), making comparisons of difference meaningful but not of ratio.
DE_ICRS	*Quantitative, Interval*	Angular distance north/south of celestial equator. Lack a true zero point, making it also an interval type for same reasons as **RA_ICRS**.
Source	*Qualitative, Nominal*	Unique code for each star, making it categorical variable without any intrinsic order or numerical value, implying no quantitative relationship => Set as **Index**.
Plx	*Quantitative, Ratio*	Apparent shift of object's position due to Earth's movement around the Sun, so it can have a true zero (infinite distance). Can be meaningfully compared using ratios.

PM	*Quantitative, Ratio*	Total movement with a true zero demonstrating no movement, making it ratio data.
pmRA **pmDE**	*Quantitative, Ratio*	Movement in **RA_ICRS** and **DE_ICRS** direction. True zero indicates no motion.
Gmag **BPmag** **RPmag** **GRVSmag**	*Quantitative, Interval*	Follow an interval scale due to the logarithmic nature of magnitude scales (there's no true zero; an object with 0 magnitude is arbitrarily bright).
e_Gmag **e_BPmag** **e_RPmag** **e_GRVSmag**	*Quantitative, Ratio*	Positive values represent the uncertainty in brightness measurements, which can be 0 (no error), hence ratio.
BP-RP **BP-G** **G-RP**	*Quantitative, Interval*	Follow interval scale as they are derived from ratios but do not have true zero point. 0 isn't absence of color but rather a balance between magnitudes used in the index
pscol	*Quantitative, Ratio*	Physical measure related to spectral characteristics, fitting ratio type due to its continuous nature and zero being meaningful, despite its scale might be complex
Teff	*Quantitative, Ratio*	Can be 0 (theoretically, at absolute 0 temperature). Differences and ratios are meaningful.
Dist	*Quantitative, Ratio*	Inverse of **Plx**. Ratio-scale since it can be compared meaningfully with a true 0 (object is infinitely far away).
Rad **Lum-Flame** **Mass-Flame** **Age-Flame**	*Quantitative, Ratio*	Each has meaningful zero point and can be compared as ratios (e.g., twice as massive/luminous)
z-Flame	*Quantitative, Ratio*	Can be negative (blueshift) - Measure with true 0 (no movement away), indicating ratio characteristics with meaningful comparisons.
SpType-ELS	*Qualitative, Nominal*	Categorize stars based on characteristics without any numerical value or inherent order, hence Nominal.

II. Identify the values of the summarizing properties for each attribute

```
X = raw_data.drop(['SpType-ELS'], axis=1) # Set X to all columns except the target
Y = raw_data['SpType-ELS'].str.strip().str.upper() # Set Y to the target column
stats_summary(raw_data).round(2)
```

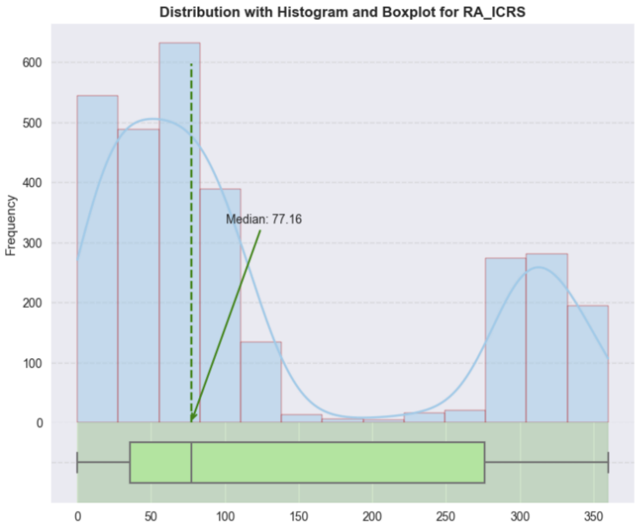
✓ 0.1s

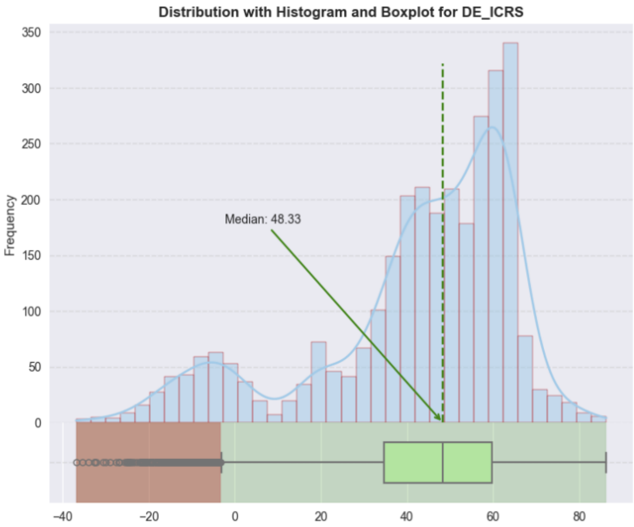
	count	mean	std	min	25%	50%	75%	max	variance	iqr_size	skewness	kurtosis	nulls_count	outliers_count	nulls_percent	outliers_percent
RA_ICRS	3000.0	125.90	117.06	0.06	35.59	77.16	275.96	359.96	13702.79	240.38	0.89	-0.86	0	0	0.00	0.00
DE_ICRS	3000.0	42.26	23.40	-36.84	34.58	48.33	59.76	86.32	547.75	25.18	-1.16	0.62	0	259	0.00	8.63
Plx	3000.0	0.91	1.10	-1.18	0.29	0.58	1.20	20.32	1.21	0.90	5.37	56.70	0	158	0.00	5.27
PM	3000.0	4.34	6.28	0.04	1.35	2.61	5.25	129.52	39.46	3.91	8.03	110.98	0	201	0.00	6.70
pmRA	3000.0	-0.57	5.26	-85.96	-2.08	-0.63	0.54	87.00	27.62	2.62	-0.72	75.39	0	337	0.00	11.23
pmDE	3000.0	-2.00	5.13	-98.09	-3.20	-1.10	-0.20	40.45	26.35	3.00	-5.76	97.94	0	270	0.00	9.00
Gmag	3000.0	13.16	2.17	4.02	11.64	13.21	14.84	17.65	4.72	3.20	-0.26	-0.33	0	8	0.00	0.27
e_Gmag	3000.0	0.00	0.00	0.00	0.00	0.00	0.00	0.01	0.00	0.00	16.57	357.34	0	367	0.00	12.23
BPmag	3000.0	13.46	2.31	3.95	11.81	13.48	15.30	18.53	5.32	3.49	-0.24	-0.42	0	8	0.00	0.27
e_BPmag	3000.0	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.00	0.00	9.29	134.83	0	413	0.00	13.77
RPmag	3000.0	12.71	2.03	4.09	11.35	12.75	14.20	17.79	4.11	2.85	-0.25	-0.20	0	19	0.00	0.63
e_RPmag	3000.0	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	8.12	87.33	0	441	0.00	14.70
GRVSmag	1706.0	11.41	1.49	4.88	10.62	11.45	12.44	14.04	2.23	1.82	-0.50	0.60	1294	38	43.13	2.23
e_GRVSmag	1706.0	0.04	0.05	0.00	0.01	0.02	0.04	0.56	0.00	0.03	3.77	23.67	1294	166	43.13	9.73
BP-RP	3000.0	0.74	0.43	-0.37	0.43	0.67	1.00	2.36	0.19	0.57	0.61	-0.01	0	22	0.00	0.73
BP-G	3000.0	0.30	0.20	-0.30	0.15	0.25	0.40	1.26	0.04	0.25	1.00	0.87	0	85	0.00	2.83
G-RP	3000.0	0.45	0.23	-0.25	0.28	0.42	0.60	1.25	0.05	0.33	0.28	-0.39	0	5	0.00	0.17
pscol	96.0	1.61	0.09	1.38	1.58	1.62	1.67	1.83	0.01	0.09	-0.73	0.57	2904	8	96.80	8.33
Teff	3000.0	9546.62	2307.55	5341.50	7714.28	9381.05	10386.80	32348.00	5324798.70	2672.52	1.84	6.67	0	153	0.00	5.10
Dist	3000.0	2320.36	2088.68	50.11	832.86	1726.13	3396.06	24511.88	4362601.49	2563.20	3.11	19.03	0	46	0.00	1.53
Rad	3000.0	2.81	1.58	0.95	1.90	2.39	3.26	39.60	2.50	1.37	6.34	107.20	0	147	0.00	4.90
Lum-Flame	2960.0	95.69	255.46	1.40	15.00	33.28	77.04	3384.95	65261.67	62.04	7.47	67.92	40	308	1.33	10.41
Mass-Flame	2747.0	2.42	0.80	1.36	1.83	2.28	2.73	7.11	0.64	0.90	1.97	5.70	253	122	8.43	4.44
Age-Flame	2237.0	0.63	0.35	0.20	0.34	0.52	0.90	1.96	0.12	0.56	0.76	-0.16	763	10	25.43	0.45
z-Flame	2960.0	0.54	0.16	0.17	0.44	0.51	0.61	1.58	0.03	0.17	1.49	4.63	40	116	1.33	3.92

1. RA_ICRS, DE_ICRS

RA_ICRS exhibits a mean of 125.9º with a wide range (0.05º-359.95º), covering nearly the full possible range of right ascension values. The std (117.05) is wide and the variance is particularly large (13,702.79), suggesting a uniform distribution. The positive skewness (0.89) indicates a longer tail towards higher values, emphasizing a concentration of stars in certain **RA_ICRS**.

DE_ICRS has a spread range of -36.83º to 86.32º, missing only the most extreme southern celestial pole regions. The mean approximately 42.26º indicates a slight northward bias in the data. The IQR for **DE_ICRS** (34.6-59.7) is narrower than **RA_ICRS** (35.6-275.96), suggesting Declination is more centrally clustered. Its negative skewness (-1.16) indicates a tail extending towards lower declinations, meaning more stars are observed at higher northern declinations in this dataset.

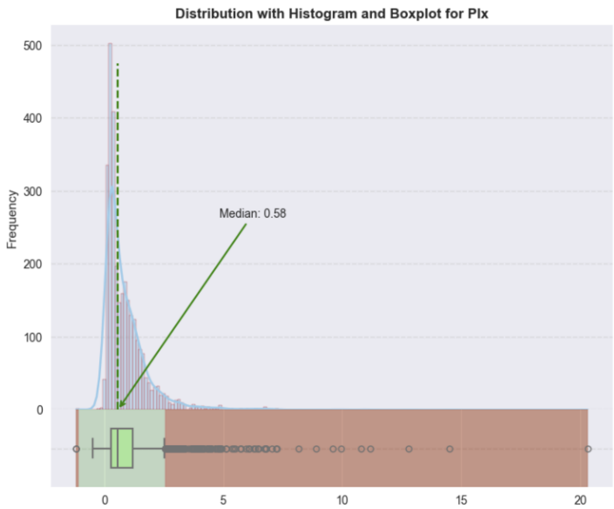


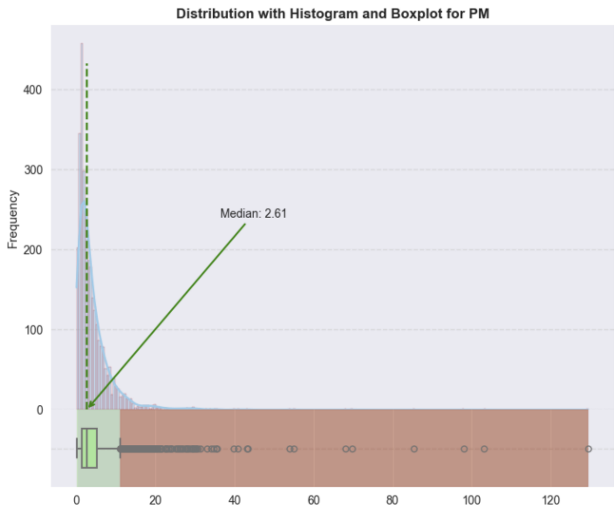


2. Plx, PM

The range from -1.18 to over 20.32 milliarcseconds with a high variance of 1.21 reflects the diversity in **Plx**. Its Histogram exhibits significant right skew (5.37) and very high kurtosis (56.7) with a mean of 0.91 and median slightly lower at 0.58, showing a significant concentration of relatively far away stars. The long tail towards higher values represents closer stars, but these are fewer in number (outliers).

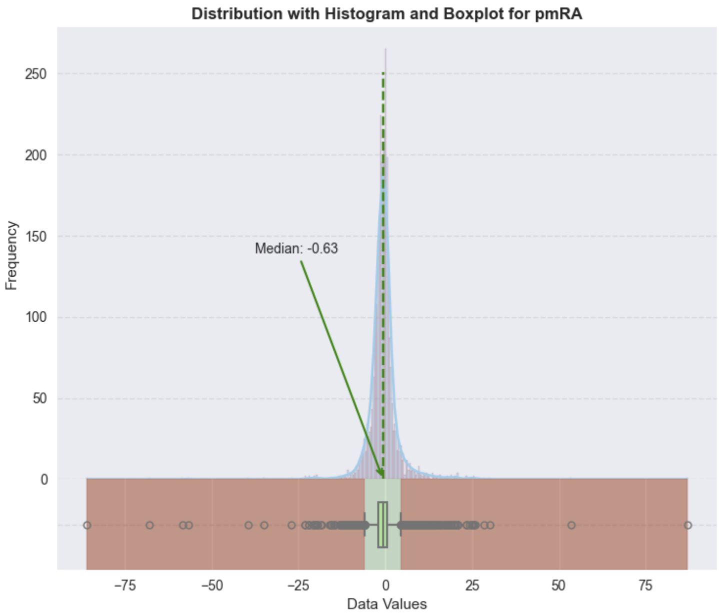
PM also exhibits a broad range (0.042-129.52) and high variance (39.46), suggesting motions from nearly stationary to exceptionally fast-moving stars. Its distribution is characterized by very high skewness (8.02) and kurtosis (110.98), indicating it is heavily skewed towards lower values but includes rare outliers with significantly high **PM** in a long tail. This skewness is a mix of relatively nearby stars (move faster) and distant stars (smaller apparent motions). The mean (4.34) being higher than median (2.61), along with above skewness and a compact IQR, confirms that while most stars exhibit modest motion, there are outliers moving much faster.

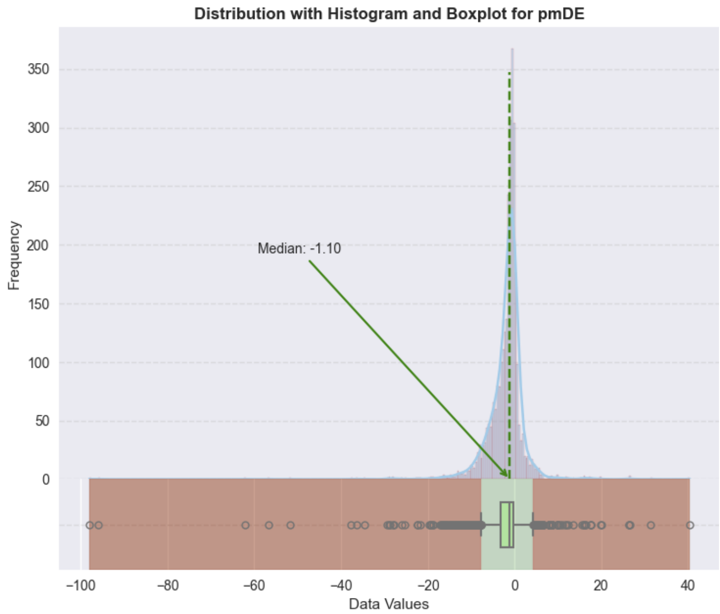




3. pmRA, pmDE

Their Histograms and Boxplots show a mean close to 0 for **pmRA** (-0.57) and slightly negative for **pmDE** (-2.00), suggesting no prevalent direction in movement. However, these components exhibit significant variances (**pmRA**:27.62; **pmDE**:26.35) and extreme range (**pmRA**: -85.956 to 87.005; **pmDE**: -98.089 to 40.447), suggesting outliers with unusually high proper motion present. Especially **pmDE**, its distribution shows substantial negative skewness (-5.7) with high kurtosis (97.94), indicating an asymmetry towards lower values and more pronounced outliers in declination motion.



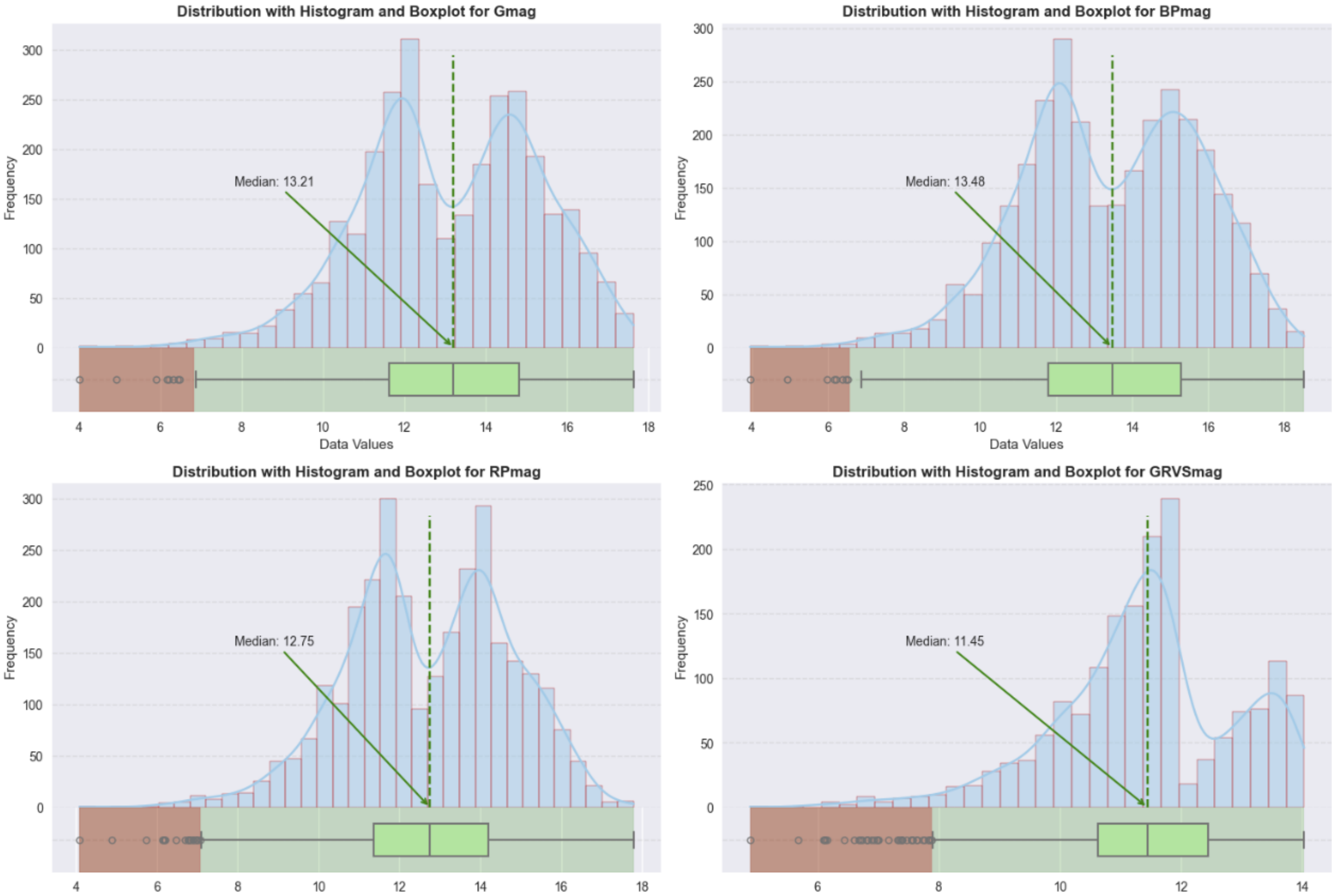


4. Gmag, BPmag, RPmag, GRVSmag

These magnitudes are quite similarly distributed and show a broad spectrum of brightness. Their means and medians are within expected ranges, suggesting a balanced dataset not biased towards either very bright or very dim stars. Their stds and variance in magnitudes reflect a significant luminosity diversity, with a slight negative skewness hinting at a tail towards lower values (as seen from Earth, lower numbers indicating brighter stars).

Gmag, for example, showcases a mean magnitude of 13.16 within a range of 4.02-17.65. Its slight negative skewness (-0.26) and a variance of 4.72 suggest a greater concentration of distant/fainter stars around an IQR of 11.6-14.8, though its distribution remains relatively close to Normal due to its low kurtosis (-0.33). The Histogram peaks for brighter stars and gradually decreases, which is expected as fainter objects are harder to detect. It also slightly tails towards both brighter and dimmer magnitudes.

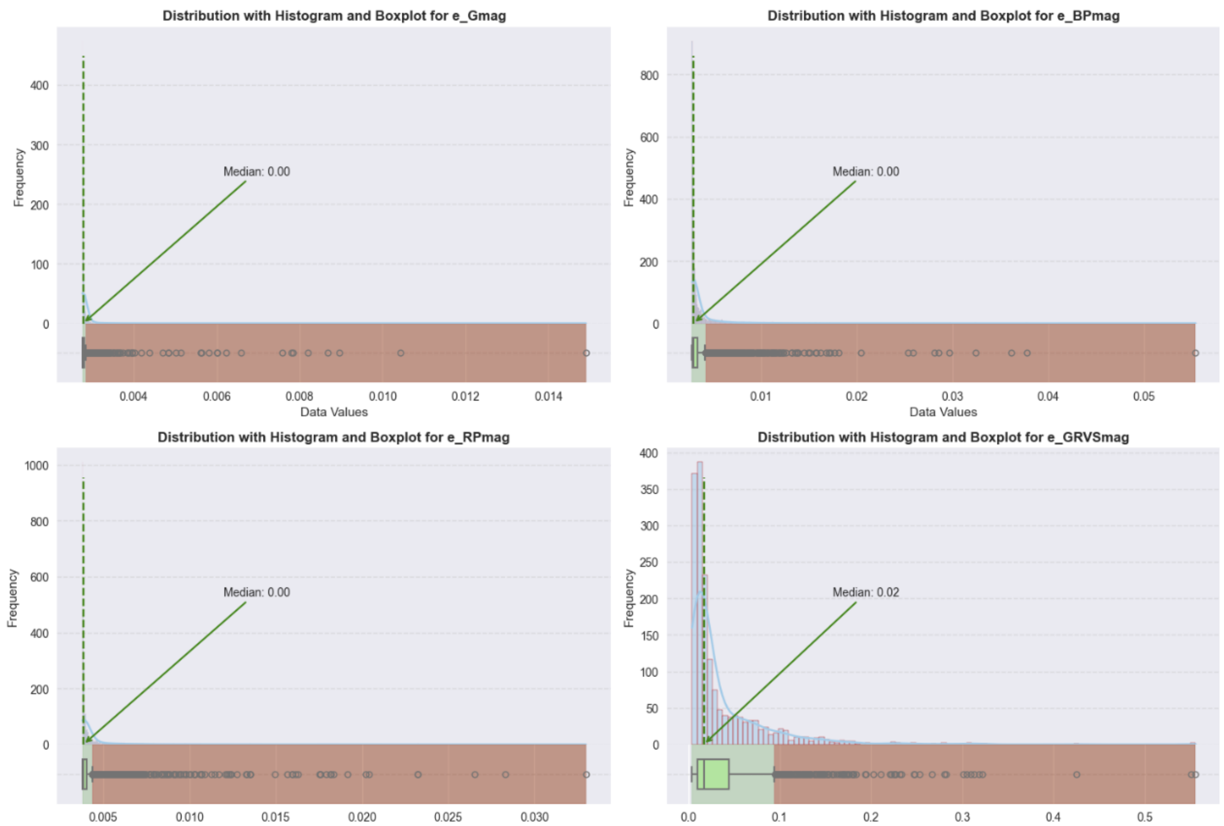
Specifically**, GRVSmag** reveals 1294 null values (~40% of the dataset), indicating that GRVS band might not be uniformly available and reliable for analysis, likely due to Gaia's detection limits or the inherent characteristics of some celestial bodies.



5. e_Gmag, e_BPmag, e_RPmag, e_GRVSmag

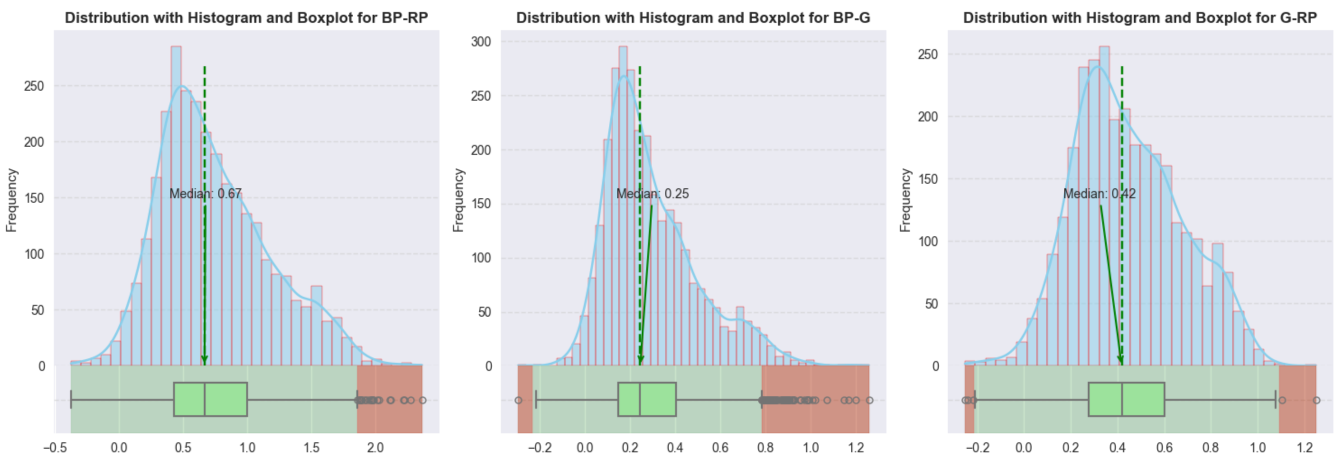
These errors are notably low on average with means ranging from 0.0028-0.0043 and high compact IQRs in boxplots, indicating high precision across magnitudes. However, **e_GRVSmag** shows a broader spread with a higher mean error (0.0365) and a maximum up to 0.56, suggesting a subset of measurements is less reliable, possibly due to the faintness of objects or the high missing values in **RVS** band.

Errors' distributions are characterized by significant positive skewness, especially **e_Gmag** (16.56) and **e_RPmag** (8.11), which also have very high kurtosis, indicating several observations with significantly higher uncertainty. This is further evidenced by their maximum values and variance far exceeding the 75th percentile, alongside very small IQR (**e_Gmag**:0.000029; **e_BPmag**:0.000532; **e_RPmag**:0.000221), indicating while most photometric observations are reliably precise, their error distributions emphasize critical need for careful consideration of certain data in sensitive analyses.



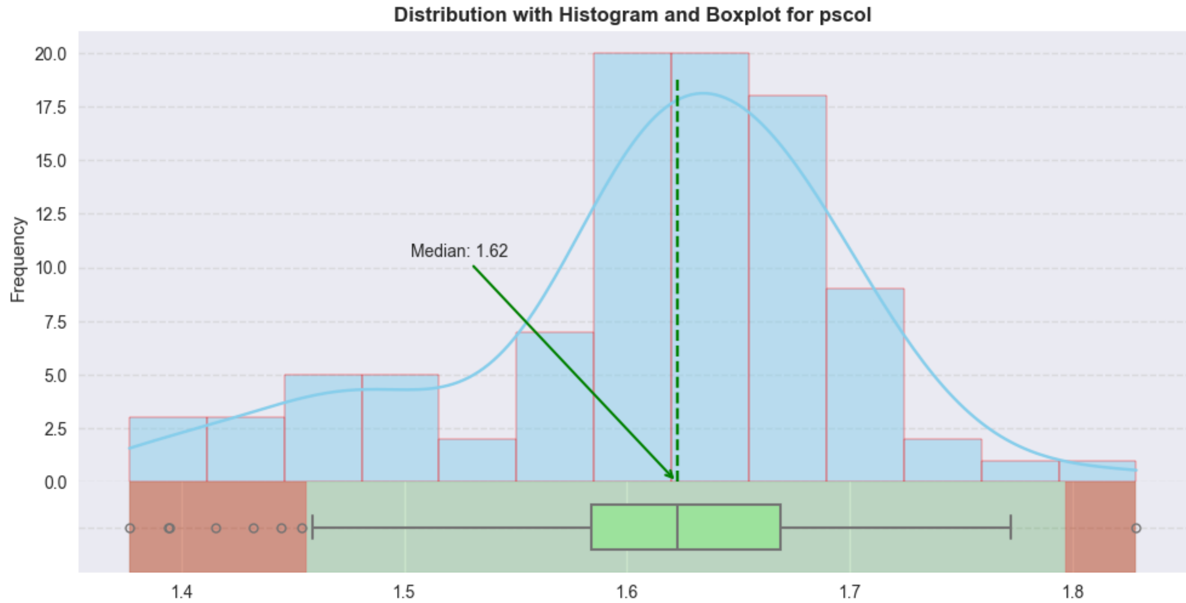
6. BP-RP, BP-G, G-RP

The means (**BP-RP**:0.742; **BP-G**:0.296; **G-RP**:0.446) and indices from negative to positive values suggest most stars align with typical main-sequence (O,B,A,F,G,K,M) star characteristics. Also, the presence of extreme values (**BP-RP**:2.36; **BP-G**:1.26; **G-RP**:1.25) further supports the wide spectrum of stellar **Temperature** and intrinsic brightnesses. These indices show moderate spread in IQRs (**BP-RP**:0.57; **BP-G**:0.25; **G-RP**:0.33), denoting relatively consistent color characteristics among most stars. Their low variance with moderate skewness towards lower values and relatively low kurtosis also reveals a fairly regular distribution of colors, though some outliers towards higher end in the long tail, as shown in boxplots, indicating unusually high-indices stars.



7. pscol

With a very high number of null values (2904), this attribute can be dropped as its statistical analysis is less reliable. However, its mean (1.61) and narrow IQR (1.58-1.67) suggest a small subset of stars for which **pseudocolor** is reported.



8. Teff

The broad range of 5341.5-32348K, alongside a significant variance and a std of 2307.55K, underscores the dataset's comprehensive coverage across different stages of stellar lifecycles, from the cooler to hotter ends of the spectrum. **Teff**'s Histogram shows a bimodal distribution with significant temperature diversity, reflecting **2 primary clusters** of stars, possibly distinguishing cooler F from hotter A/B stars.

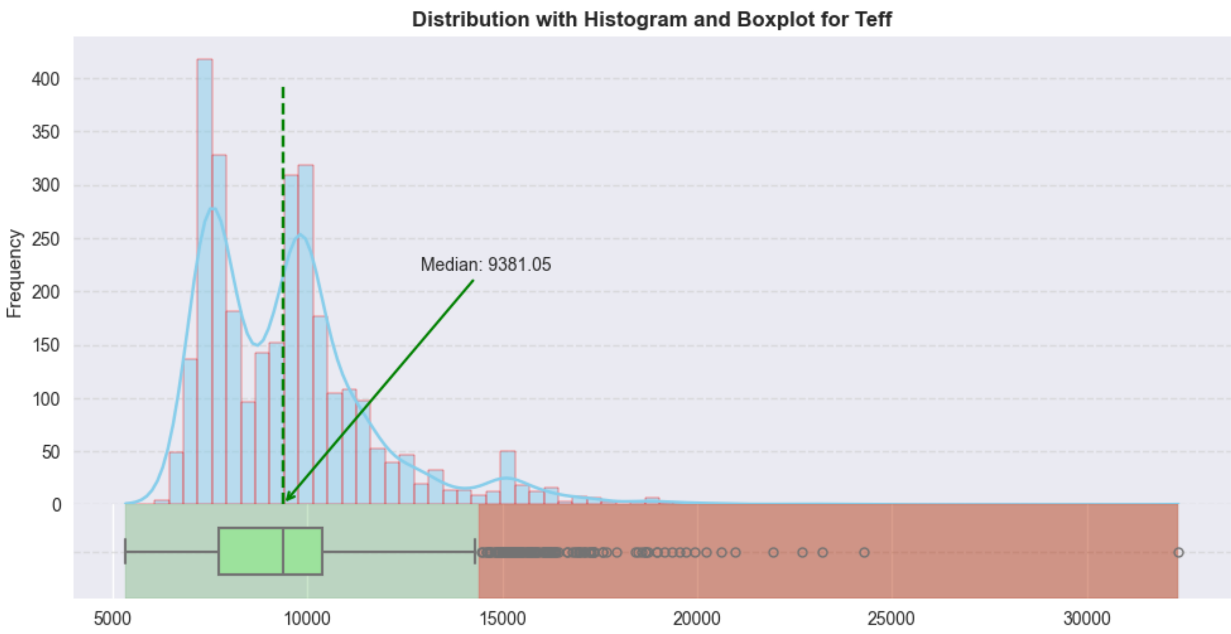
```
# https://jila.colorado.edu/~ajsh/courses/astr1200_18/star.html
main_seq_labels = [
    'M: ≤3700K', 'K: 3700-5200', 'G: 5200-6000K', 'F: 6000-7500K',
    'A: 7500-10000K', 'B: 10000-30000K', 'O: ≥30000K'
]
main_seq_bins = [-np.inf, 3700, 5200, 6000, 7500, 10000, 30000, np.inf]
pd.cut(
    X['Teff'], include_lowest=True, right=False,
    bins=main_seq_bins, labels=main_seq_labels
).value_counts()
```

✓ 0.0s

```
Teff
A: 7500-10000K     1492
B: 10000-30000K     955
F: 6000-7500K       551
G: 5200-6000K         1
O: ≥30000K            1
M: ≤3700K             0
K: 3700-5200          0
```

Star type	Occurrence	Interpretation
A (7500-10000K)	1492	**Significant** presence of hot, white to blue stars that are larger, more luminous than the Sun
B (10000-30000K)	955	**Considerable** number of even hotter and much more luminous stars
F (6000-7500K)	551	Slightly hotter and luminous than the Sun
Others	Very few stars at the cooler (G, K, M) and the hottest (O) ends of the spectrum, with only 1 occurrence each in 'G' (5000-6000K) and 'O' (>=30000K). This distribution suggests higher detection efficiency of Gaia's mission toward hotter stars	

While the dataset includes many hot stars, the distribution is right skewed towards cooler stars with a high mean (9546K) and a considerable IQR (2672.5K), which are more prevalent in our galaxy. This is further evidenced through a positive skewness (1.84) and high kurtosis (6.66), demonstrating a long tail towards higher temperatures, where some extremely hot objects, less frequent stars are indicated as outliers.



9. Dist

The significant range of 50.11pc-24,511.88pc demonstrates nearby to far-away stars. The mean of 2320.4pc, along with a high std (2088.68) and significant IQR of 2563.2pc, showcases Gaia's observational capability across vast expanse of space.

- Most stars (1821) fall within the 1-5kpc range, indicating that Gaia's observations are effectively probing deep into the Milky Way.
- The 500-1000pc and 100-500pc categories have 692 and 259 stars, respectively, showing substantial local galactic observation.
- There are relatively few stars within the very close (<100pc) and very far (>10kpc) categories, with only 6 stars closer than 100pc and 29 farther than 10kpc.

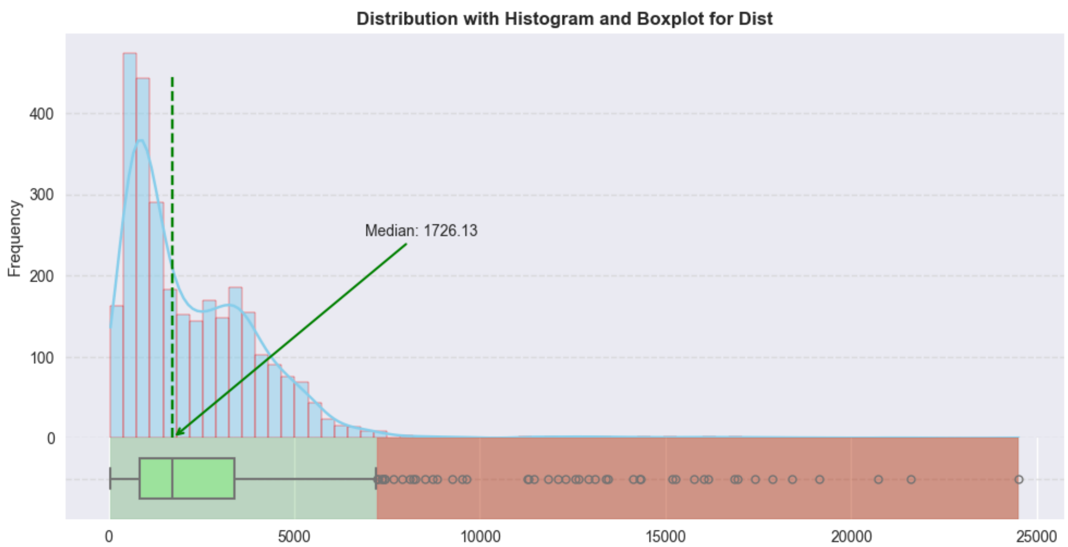
```python
pd.cut(
    X['Dist'], include_lowest=True, right=False,
    bins=[-np.inf, 100, 500, 1000, 5000, 10000, np.inf] # Distance in parsecs
).value_counts()
```

```
0.0s
```

```
Dist
[1000.0, 5000.0)     1821
[500.0, 1000.0)       692
[100.0, 500.0)        259
[5000.0, 10000.0)     193
[10000.0, inf)         29
[-inf, 100.0)           6
```

This distribution is further characterized by high skewness (3.11) and kurtosis (19.02), emphasizing:

- Concentration of nearby stars: Crucial for mapping local stellar neighborhoods.
- A long tail of much more distant objects considered as significant number of outliers in the Boxplot, reflecting challenges of measuring these stars along with their **Plx**.
- The non-uniform distribution of celestial objects and the vastness of space.



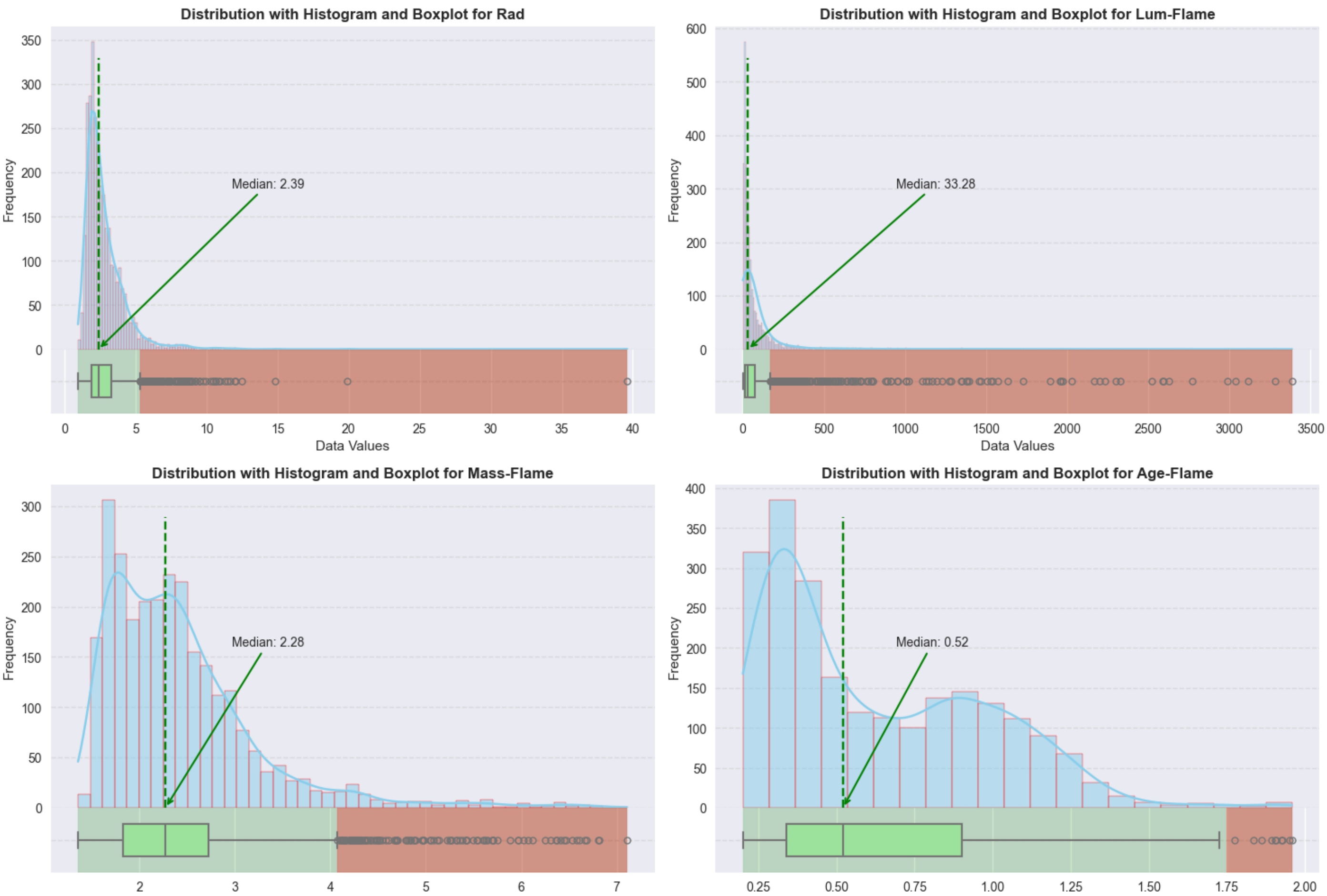
10. Rad, Lum-Flame, Mass-Flame, Age-Flame

The moderate null values in them (**Lum-Flame**:40; **Mass-Flame**:253; **Age-Flame**:763) requires attention during preprocessing.

Rad has right-skewed distribution across Histogram and Boxplot, reflecting the domination of F-A stars with a noticeable long outlier tail highlighting significant number of larger radii in unusually large stars. The range (0.95-39.6) from those smaller than the Sun to several times its size underscores the vast diversity of star sizes.

Lum-Flame is also extremely right-skewed and has a relatively small IQR with high variance (65261), indicating most stars exhibit lower luminosity, while a long tail extends towards higher luminosity in O-A stars. These findings are further evidenced by the Boxplot, where numerous outliers on high-luminosity end reflect stars with extraordinary energy output. Such differences are crucial as luminosity is closely linked to properties like **Mass** and **Age**.

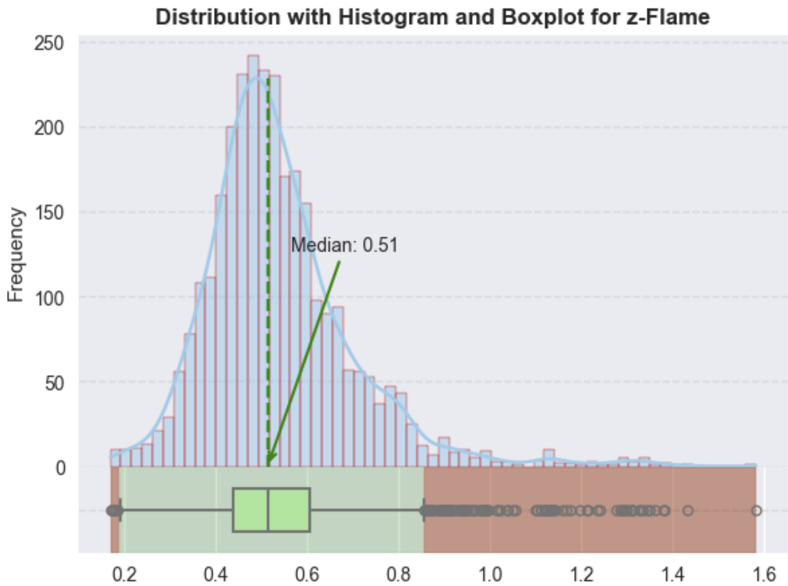
Mass distribution is positively skewed, like **Luminosity**, with most stars having lower masses but including a tail of higher-mass stars. Outliers in **Mass-Flame** indicate massive stars that are less common but crucial for understanding stellar evolution and galaxy dynamics.**Age** distribution is slightly skewed towards younger stars and includes outliers at both ends of the spectrum, representing very young to ancient stars, providing insights into different stages of stellar evolution.



11. z-Flame

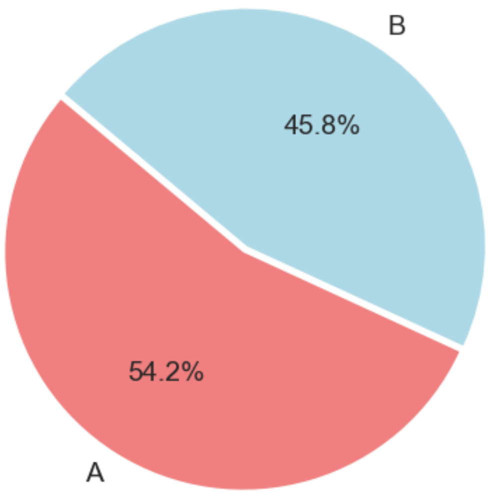
Although the range of 0.17-1.58km/s with a mean close to 0 indicates objects moving away at moderate velocities, the moderate skewness (1.49) and kurtosis (4.63) suggest a tail towards higher values. This pattern demonstrates most stars moving at modest speeds relative to Earth, with a tight clustering around the median (0.51).

However, the outliers with relatively high redshift present exceptionally fast-moving objects, possibly due to peculiar motions or distant galaxies. The low variance (0.026) and IQR (0.44-0.6) further underscore these high-velocity outliers as significant despite the overall uniform distribution of **z-Flame**.



12. SpType-ELS

The attribute classifies stars into 2 spectral classes, **A** and **B**, with occurrences of **1627** and **1373** respectively, showing a balanced yet slightly A-dominant distribution.

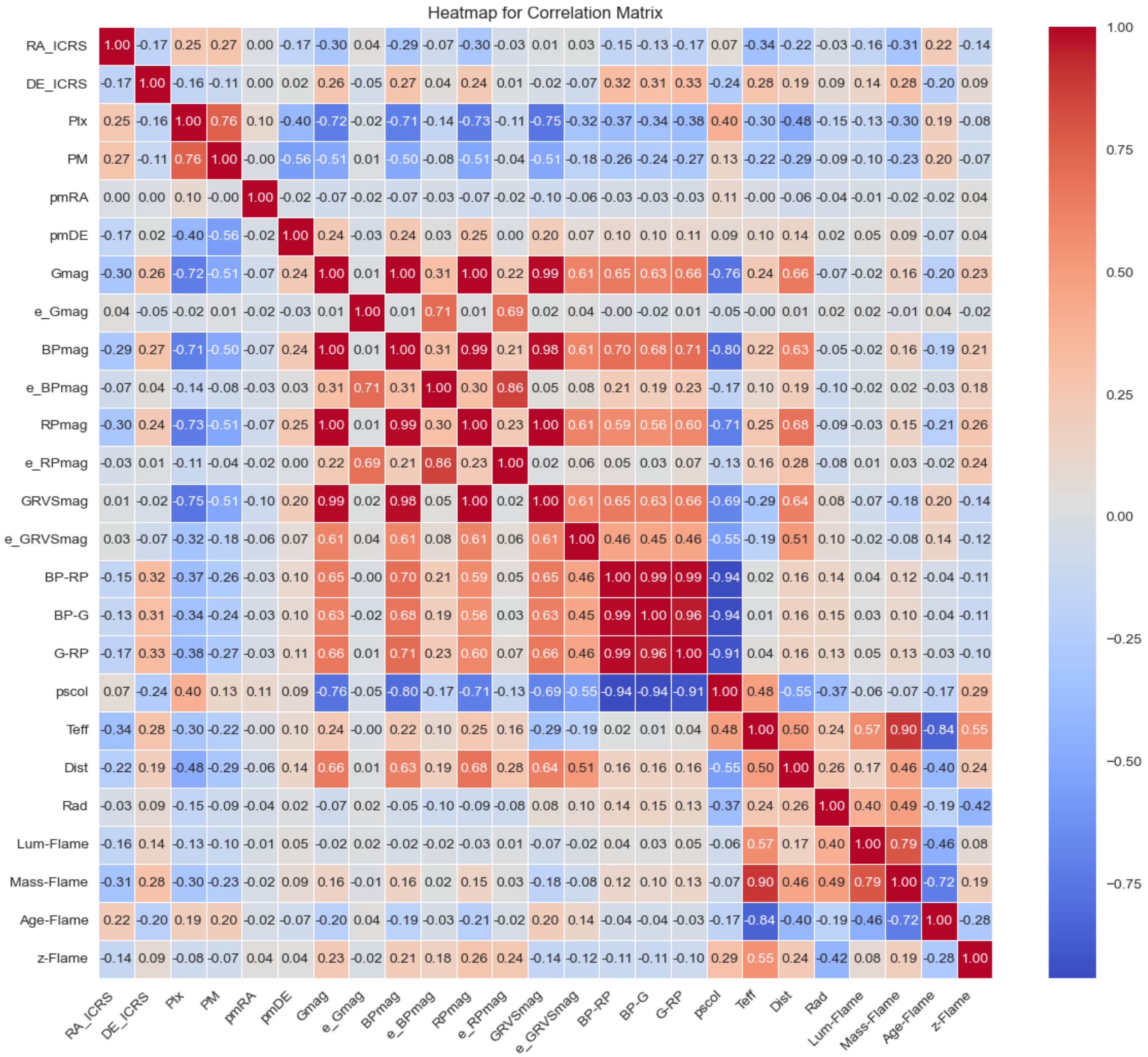


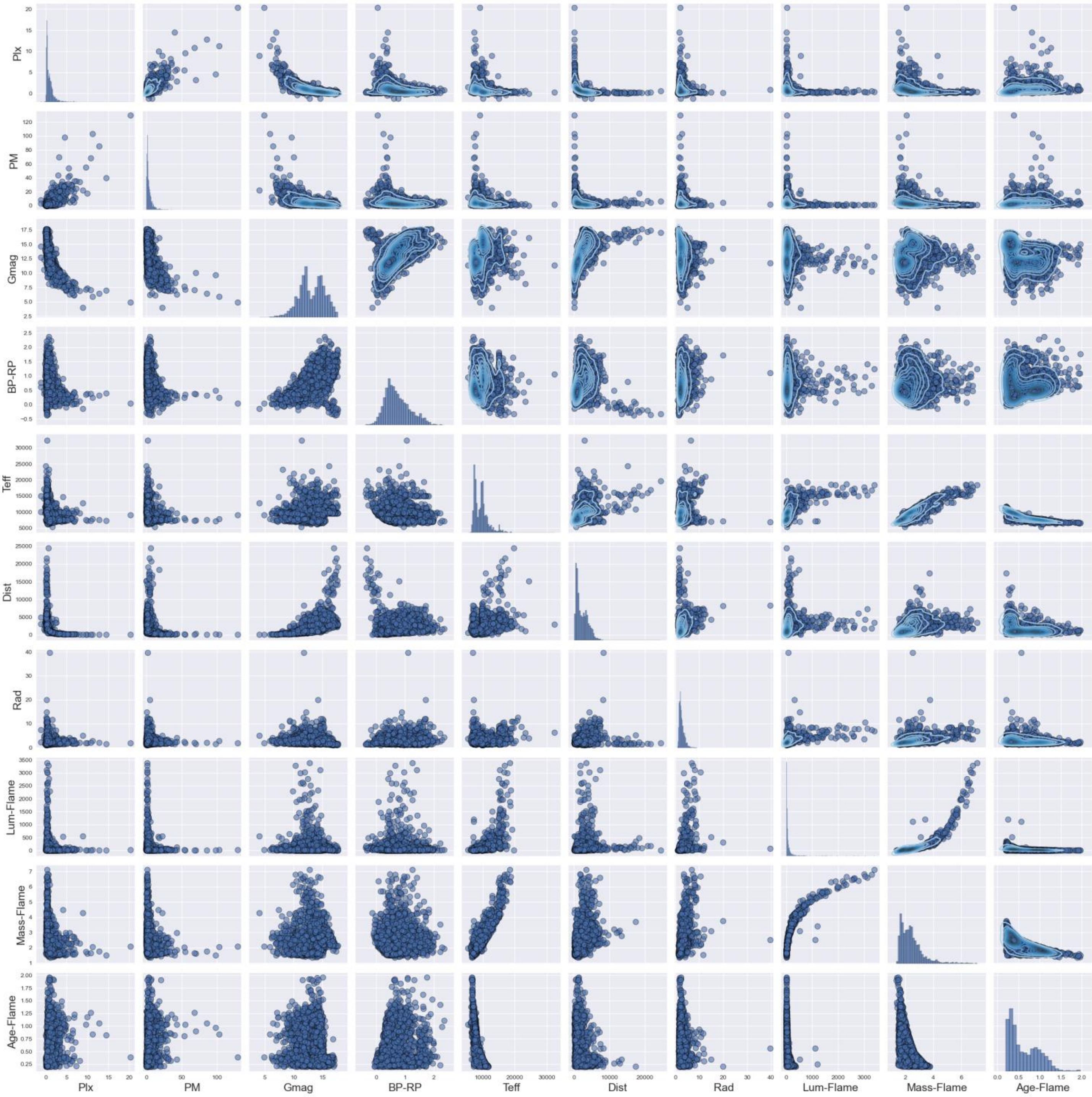

```python
sp_type_counts = Y.value_counts()
plt.pie(
    sp_type_counts,
    labels = sp_type_counts.index,
    colors = ['lightcoral', 'lightblue'],
    explode = (0, 0.03),
    autopct = '%1.1f%%',
    startangle = 140,
    textprops = {'fontsize': 14}
)
```

III. Explore multiple attributes relationship

```
# Calculate the correlation matrix and plot its heatmap
corr_matrix = X.select_dtypes(include=[np.number]).corr()
# mask = np.triu(np.ones_like(corr_matrix, dtype=bool), k=1) # Generate a mask for the upper triangle
plt.figure(figsize=(14, 12))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=.5)
plt.xticks(rotation=45, ha="right")
plt.title('Heatmap for Correlation Matrix')
plt.show()

# Pairplot for the numerical features to visualize the relationships
g = sns.pairplot(
    X[['Plx', 'PM', 'Gmag', 'BP-RP', 'Teff', 'Dist', 'Rad', 'Lum-Flame', 'Mass-Flame', 'Age-Flame']],
    palette = sns.cubehelix_palette(8, start=.5, rot=-.75, as_cmap=True), # Create a custom color pal
    plot_kws = {'alpha':0.6, 's':80, 'edgecolor':'k'},
)
g.map_upper(sns.kdeplot, cmap="Blues_d") # Adjust the top right plot to have a different kind of plot
plt.tight_layout()
plt.show()
```





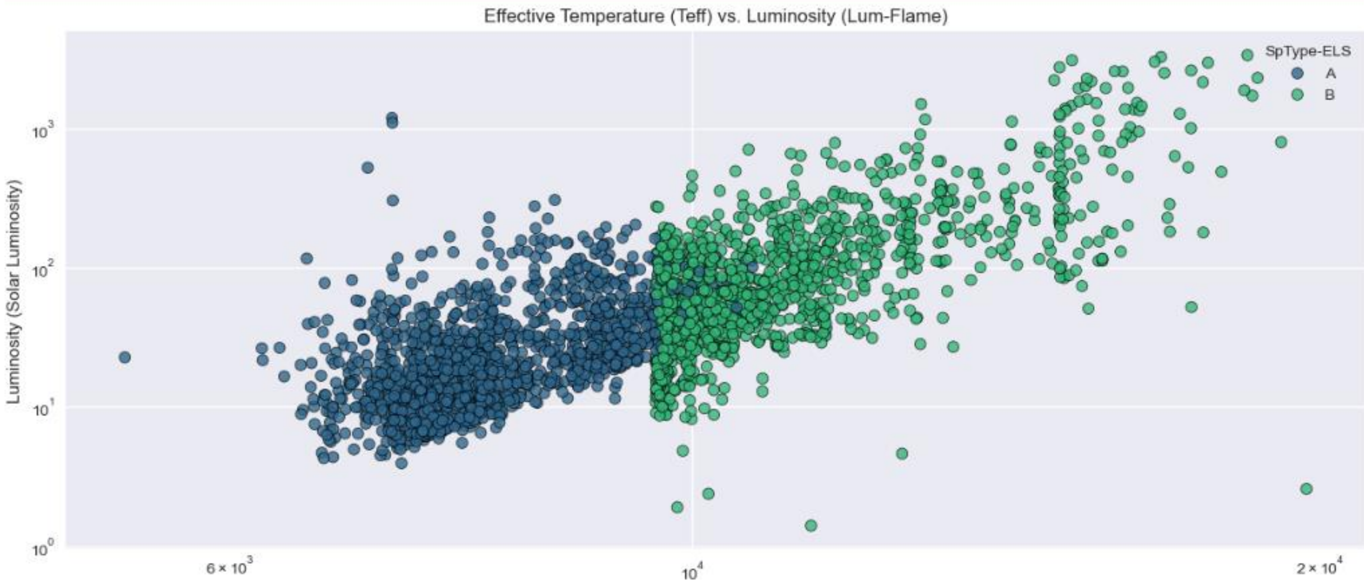
1. **Teff – Interesting Attribute**

The number of high correlations with other attributes (not same type) of **Teff** is the **highest** compared to other attributes. Many relationships involving **Teff** can reveal how observed properties of objects vary with their temperature.

- **Teff vs. Dist** (0.5, *Moderate positive relationship*): Hotter, more luminous stars tend to be observed at greater distances, potentially reflecting the survey's capability to detect luminous, hot stars far away.

- **Teff vs. Age-Flame** (-0.84, *Robust negative correlation*): Older stars tend to be cooler, reflecting stars' cooling as they age or the evolution of massive, hotter stars into end-of-life stages more quickly.
- **Teff vs. Mass-Flame** (0.90, *Robust positive correlation*): Higher mass stars tend to be hotter, aligning with theoretical expectations about stellar physics.
- **Teff vs. Lum-Flame** (0.57, *Moderate positive correlation*):
  - Hotter stars are generally more luminous: The scatter suggests **2 groups** occupying specific regions, with **SpType-ELS A** stars forming a diagonal band from lower left (cooler, less luminous) to upper right (hotter, more luminous) of **SpType-ELS B** class.
  - Outliers, especially in **Luminosity**, can represent rare stellar phenomena, like hypergiants or stars at critical evolutionary stages.

```
sns.scatterplot(x='Teff', y='Lum-Flame', data=X, hue=Y, palette='viridis', edgecolor='k', s=50, alpha=0.8)
plt.title('Effective Temperature (Teff) vs. Luminosity (Lum-Flame)')
plt.ylabel('Luminosity (Solar Luminosity)')
plt.xscale('log') # Logarithmic scales on both axes helps visualize the wide range of values and the multiplicative relationship.
plt.yscale('log') # This scaling is crucial for handling the significant variance in stellar properties across the dataset.
plt.show()
```



2. Dist Related Relationships

- **Dist vs. Lum-Flame** (0.17, *Weak correlation*): More luminous stars can be observed at greater distances, but the wide dispersion suggests that **Luminosity** alone doesn't determine visibility; factors like interstellar dust can affect it.
- **Dist vs. Gmag/BPmag/RPmag/GRVSmag** (~0.6, *Moderate positive relationship*): Stars further away are fainter, an expected result of the inverse square law of light.

- **Dist vs. Plx** (-0.48, *Moderate negative*): As expected, distance is the inverse of parallax, with larger parallax value indicating closer star.
- **Plx vs. PM** (0.76, *Strong positive*): **Closer** stars exhibit more significant apparent motions. This relationship is expected as **closer** stars can move more quickly, given their actual motion through space and their proximity to the Solar System.

3. Physical Characteristics Relationships

- **Lum-Flame vs. Mass-Flame** (0.79, *Strong positive correlation*): Consistent with the mass-luminosity relationship for main-sequence stars, where more massive stars are generally more luminous.
- **Mass-Flame vs. Age-Flame** (-0.72, *Strong negative correlation*): More massive stars have shorter lifespans, a well-known aspect of stellar evolution.
- **Triple Relationships**: **Radius** and **Luminosity** are related through the Stefan-Boltzmann law, where a larger radius at a given **Temperature** leads to higher **Luminosity**, which is expected due to the luminosity's dependence on both the surface area (related to **Radius**) and the fourth power of the surface **Temperature**.

4. Brightness' Relationships

As expected, there's an *absolute positive correlation* (~1) among **magnitudes (Gmag/BPmag/RPmag/GRVSmag)**, indicating 1 magnitude increases, others tend to increase as well. This is logical, given that these **magnitudes** measure brightness in different bands but of the same objects.

Magnitudes and **Color Indices** (**BP-RP/BP-G/G-RP**) don't show a strong/direct relationship with **Teff,** suggesting that they alone might not be a reliable indicator of **Temperature** without considering other factors:

- **Gmag/BPmag/RPmag/GRVSmag vs. Teff** (~0.22 each, *Weak correlations*): Although there's some relationship between star **temperature** and its apparent **magnitude**, it's not as strong/direct as other relationships, likely due to the wide range of **Distances** and **Luminosities** of stars in the dataset.
- **BP-RP/BP-G/G-RP vs. Teff** (~0.02 each, *Weakest correlations*): Almost no direct linear relationship, suggesting **Color Indices**, while informative, cannot solely predict a star's **Temperature**.

However, there's a visible positive trend in pairplot correlating **Magnitudes** with **Color Indices**, reflecting the underlying relationship between a star's brightness and observed color, where brighter stars (lower **Magnitude**) tend to have lower **Indices**.

IV. Smoothing for RA_ICRS and DE_ICRS

The choice of bins should balance spatial detail with generalization. Too few bins might oversimplify the data, while too many could complicate the analysis. A selection of **24** bins for **RA_ICRS** and **18** for **DE_ICRS**, given the dataset's size (3000) and their full range of possible values (0-360^{o} for **RA** and -90^{o} to +90^{o} for **DE**), in **both** binning strategies will align with above goal.

24 bins for **RA_ICRS**, in particular, correlate with the 24-hour celestial clock. This choice aligns with how celestial maps are often segmented and allows easy interpretation of data in a familiar context.

1. Equi-width Binning

This technique divides the range of a variable into smaller intervals of equal size to simplify the data. It provides a straightforward way to understand the overall spread and central tendencies, revealing the uniformity/variance in distribution.

Steps for Equi-width Binning:

1) Define the number of bins: **24** for **RA_ICRS** and **18** for **DE_ICRS**.
2) Apply **pandas.cut()** function on each attribute with the choice of bins in the ***bins*** parameter to assign each observation to its corresponding bin.
3) **Rename** bins to include bin number and the interval.
4) **Group** the DataFrame by bins and calculate their **statistics**.

This method ensures that each bin covers an equal range of values:

```
bins_ra_equiwidth = 24  # 24 bins for RA_ICRS
bins_de_equiwidth = 18  # 18 bins for DE_ICRS
```

- **RA_ICRS**: Evenly distributed into **24** intervals with ~15-degree segments, ranging from -0.304^{o} to ~360^{o} (359.956^{o}). This reflects the full possible range of RA values. The means within these bins range from 7.63^{o}-351.8^{o}, illustrating how objects are distributed across these equal intervals.

```python
X['RA_ICRS_EquiWidth_Bin'] = pd.cut(
    X['RA_ICRS'],
    bins=bins_ra_equiwidth, # Number of bins
    include_lowest=True # Include the lowest value in the bin
)

X['RA_ICRS_EquiWidth_Bin'] = X['RA_ICRS_EquiWidth_Bin']\
    .cat.rename_categories([
        f'Bin {i + 1}: {itv}' # Rename categories to include bin number
        for i, itv in enumerate(X['RA_ICRS_EquiWidth_Bin'].cat.categories)
    ])

ra_icrs_equiwidth_stats = X.groupby('RA_ICRS_EquiWidth_Bin')['RA_ICRS']\
    .agg(['min', 'max', 'mean', 'count'])\
    .rename(columns={
        'min': 'bin_min',
        'max': 'bin_max',
        'mean': 'bin_mean',
        'count': 'bin_size'
    })
ra_icrs_equiwidth_stats
```

RA_ICRS_EquiWidth_Bin	bin_min	bin_max	bin_mean	bin_size
Bin 1: (-0.304, 15.052]	0.056661	15.026099	7.630146	280
Bin 2: (15.052, 30.048]	15.067836	30.033446	23.459322	331
Bin 3: (30.048, 45.044]	30.077667	45.031178	37.069460	313
Bin 4: (45.044, 60.04]	45.180042	59.984370	53.188644	178
Bin 5: (60.04, 75.036]	60.068962	75.033579	68.286941	328
Bin 6: (75.036, 90.031]	75.074498	89.723854	80.442088	306
Bin 7: (90.031, 105.027]	90.034653	104.863268	99.387881	184
Bin 8: (105.027, 120.023]	105.036853	119.870161	110.841191	231
Bin 9: (120.023, 135.019]	120.064384	134.594113	124.995206	35
Bin 10: (135.019, 150.015]	136.229195	149.874372	142.123304	11
Bin 11: (150.015, 165.01]	150.516126	154.634432	152.329009	5
Bin 12: (165.01, 180.006]	171.967711	171.967711	171.967711	1
Bin 13: (180.006, 195.002]	182.322442	194.062293	186.631208	6
Bin 14: (195.002, 209.998]	197.066600	197.066600	197.066600	1
Bin 15: (209.998, 224.994]	211.567141	223.147460	218.559869	5
Bin 16: (224.994, 239.989]	227.288142	238.325075	231.178218	11
Bin 17: (239.989, 254.985]	240.660401	253.928829	247.343292	9
Bin 18: (254.985, 269.981]	259.098175	269.596745	262.041656	11
Bin 19: (269.981, 284.977]	271.134144	284.583553	279.982496	24
Bin 20: (284.977, 299.973]	285.835636	299.882868	295.056120	171
Bin 21: (299.973, 314.968]	300.137941	314.927927	306.607676	201
Bin 22: (314.968, 329.964]	314.976559	329.871320	322.218169	142
Bin 23: (329.964, 344.96]	329.972418	344.846893	336.963167	111
Bin 24: (344.96, 359.956]	344.976732	359.955738	351.800714	105

```python
X[['RA_ICRS', 'RA_ICRS_EquiWidth_Bin']]\
    .merge(ra_icrs_equiwidth_stats, on='RA_ICRS_EquiWidth_Bin')\
    .sort_values('RA_ICRS')
```

	RA_ICRS	RA_ICRS_EquiWidth_Bin	bin_min	bin_max	bin_mean	bin_size
2071	0.056661	Bin 1: (-0.304, 15.052]	0.056661	15.026099	7.630146	280
2075	0.120534	Bin 1: (-0.304, 15.052]	0.056661	15.026099	7.630146	280
1997	0.126239	Bin 1: (-0.304, 15.052]	0.056661	15.026099	7.630146	280
2047	0.202935	Bin 1: (-0.304, 15.052]	0.056661	15.026099	7.630146	280
1952	0.210756	Bin 1: (-0.304, 15.052]	0.056661	15.026099	7.630146	280
...	...	...	...	...	...	...
2835	358.630830	Bin 24: (344.96, 359.956]	344.976732	359.955738	351.800714	105
2848	359.005901	Bin 24: (344.96, 359.956]	344.976732	359.955738	351.800714	105
2838	359.366908	Bin 24: (344.96, 359.956]	344.976732	359.955738	351.800714	105
2837	359.474779	Bin 24: (344.96, 359.956]	344.976732	359.955738	351.800714	105
2895	359.955738	Bin 24: (344.96, 359.956]	344.976732	359.955738	351.800714	105

3000 rows × 6 columns

- **DE_ICRS**: **18** intervals with approximately 10-degree segments, covering the range from -90° to +90°, although due to the dataset's specific range, it effectively spanned from approximately -36.96° to +86.322°. The mean values in each bin gradually increase, from an average of -33.01° to 82.38°.

```
X['DE_ICRS_EquiWidth_Bin'] = pd.cut(
    X['DE_ICRS'],
    bins=bins_de_equiwidth, # Number of bins
    include_lowest=True # Include the lower bound in the interval
)

X['DE_ICRS_EquiWidth_Bin'] = X['DE_ICRS_EquiWidth_Bin']\
    .cat.rename_categories([
        f'Bin {i + 1}: {itv}' # Rename categories to include bin number
        for i, itv in enumerate(X['DE_ICRS_EquiWidth_Bin'].cat.categories)
    ])

de_icrs_equiwidth_stats = X.groupby('DE_ICRS_EquiWidth_Bin')['DE_ICRS']\
    .agg(['min', 'max', 'mean', 'count'])\
    .rename(columns={
        'min': 'bin_min',
        'max': 'bin_max',
        'mean': 'bin_mean',
        'count': 'bin_size'
    })
de_icrs_equiwidth_stats
```

DE_ICRS_EquiWidth_Bin	bin_min	bin_max	bin_mean	bin_size
Bin 1: (-36.961, -29.995]	-36.836939	-30.024048	-33.012212	8
Bin 2: (-29.995, -23.153]	-29.025192	-23.398555	-25.441197	13
Bin 3: (-23.153, -16.31]	-22.800680	-16.314553	-18.909556	43
Bin 4: (-16.31, -9.468]	-16.221155	-9.492822	-12.892364	84
Bin 5: (-9.468, -2.626]	-9.455728	-2.644397	-5.993829	122
Bin 6: (-2.626, 4.216]	-2.598615	4.181907	0.217255	90
Bin 7: (4.216, 11.058]	4.229728	10.820982	6.676293	27
Bin 8: (11.058, 17.9]	11.303179	17.897458	15.103663	54
Bin 9: (17.9, 24.743]	17.902774	24.582772	20.950715	118
Bin 10: (24.743, 31.585]	24.878072	31.565620	28.777791	108
Bin 11: (31.585, 38.427]	31.617043	38.417112	35.476411	250
Bin 12: (38.427, 45.269]	38.449678	45.259455	41.832473	414
Bin 13: (45.269, 52.111]	45.282817	52.038592	48.749401	397
Bin 14: (52.111, 58.953]	52.112991	58.952519	55.913396	452
Bin 15: (58.953, 65.795]	58.983512	65.744911	62.310084	655
Bin 16: (65.795, 72.638]	65.816492	72.336955	68.296244	108
Bin 17: (72.638, 79.48]	72.776297	79.147266	75.608416	42
Bin 18: (79.48, 86.322]	79.954977	86.321954	82.386341	15

```
X[['DE_ICRS', 'DE_ICRS_EquiWidth_Bin']]\
    .merge(de_icrs_equiwidth_stats, on='DE_ICRS_EquiWidth_Bin')\
    .sort_values('DE_ICRS')
```

	DE_ICRS	DE_ICRS_EquiWidth_Bin	bin_min	bin_max	bin_mean	bin_size
2968	-36.836939	Bin 1: (-36.961, -29.995]	-36.836939	-30.024048	-33.012212	8
2971	-35.472313	Bin 1: (-36.961, -29.995]	-36.836939	-30.024048	-33.012212	8
2967	-34.019772	Bin 1: (-36.961, -29.995]	-36.836939	-30.024048	-33.012212	8
2970	-32.571064	Bin 1: (-36.961, -29.995]	-36.836939	-30.024048	-33.012212	8
2966	-32.505914	Bin 1: (-36.961, -29.995]	-36.836939	-30.024048	-33.012212	8
...	...	...	...	...	...	...
2982	84.847554	Bin 18: (79.48, 86.322]	79.954977	86.321954	82.386341	15
2972	84.852630	Bin 18: (79.48, 86.322]	79.954977	86.321954	82.386341	15
2973	85.285817	Bin 18: (79.48, 86.322]	79.954977	86.321954	82.386341	15
2977	85.704616	Bin 18: (79.48, 86.322]	79.954977	86.321954	82.386341	15
2980	86.321954	Bin 18: (79.48, 86.322]	79.954977	86.321954	82.386341	15

2. Equi-depth Binning

This technique segments the data into bins so that each bin has approximately the same number of data points. It is useful for handling skewed data by ensuring that each bin is equally represented.

Steps for Equi-depth Binning:

1) Define the number of bins: **24** for **RA_ICRS** and **18** for **DE_ICRS**.
2) Apply **pandas.qcut()** function on each attribute with the choice of bins in the ***q*** parameter to divide observations so that each bin has the same size.
3) **Rename** bins to include bin number and the boundary.
4) **Group** the DataFrame by bins and calculate their **statistics**.

This technique adapts to the data's density, providing insights into where stars are more/less concentrated, which may not be as apparent with **Equi-width** binning:

```
bins_ra_equidepth = 24  # Matching the equi-wi
bins_de_equidepth = 18  # Matching the equi-wi
```

- **RA_ICRS**: Bins don't have equal widths and vary from narrower to wider segments where data are sparser. However, they are designed to contain approximately same number of objects. Each bin mean highlights how objects are not uniformly distributed; for example, there's a high jump of means in bins covering higher **RA_ICRS** ranges (from Bin 18th), indicating clusters of objects in specific regions.

```
X['RA_ICRS_EquiDepth_Bin'] = pd.qcut(
    X['RA_ICRS'],
    q=bins_ra_equidepth # Number of quantiles
)

X['RA_ICRS_EquiDepth_Bin'] = X['RA_ICRS_EquiDepth_Bin']\
    .cat.rename_categories([
        f'Bin {i + 1}: {itv}' # Renaming categories to include bin number
        for i, itv in enumerate(X['RA_ICRS_EquiDepth_Bin'].cat.categories)
    ])

ra_icrs_equidepth_stats = X.groupby('RA_ICRS_EquiDepth_Bin')['RA_ICRS']\
    .agg(['min', 'max', 'mean', 'count'])\
    .rename(columns={
        'min': 'bin_min',
        'max': 'bin_max',
        'mean': 'bin_mean',
        'count': 'bin_size'
    })
ra_icrs_equidepth_stats
```

RA_ICRS_EquiDepth_Bin	bin_min	bin_max	bin_mean	bin_size
Bin 1: (0.0557, 6.622]	0.056661	6.480101	3.270302	125
Bin 2: (6.622, 13.831]	6.628199	13.760977	10.371937	125
Bin 3: (13.831, 20.621]	13.837377	20.588782	17.298099	125
Bin 4: (20.621, 26.0]	20.626016	25.982600	23.415338	125
Bin 5: (26.0, 30.561]	26.003655	30.542909	28.250420	125
Bin 6: (30.561, 35.586]	30.565580	35.531030	33.326660	125
Bin 7: (35.586, 42.099]	35.603699	42.082331	39.088483	125
Bin 8: (42.099, 51.916]	42.105922	51.910594	46.579429	125
Bin 9: (51.916, 61.082]	51.918936	61.054705	57.311828	125
Bin 10: (61.082, 68.186]	61.098298	68.174496	64.615772	125
Bin 11: (68.186, 72.973]	68.193959	72.957977	70.894654	125
Bin 12: (72.973, 77.161]	72.986508	77.095388	75.124538	125
Bin 13: (77.161, 81.193]	77.226938	81.176071	79.180417	125
Bin 14: (81.193, 92.473]	81.212834	92.469999	85.405737	125
Bin 15: (92.473, 103.024]	92.477042	102.992217	98.626198	125
Bin 16: (103.024, 108.335]	103.076236	108.307627	105.778280	125
Bin 17: (108.335, 116.166]	108.389417	116.160821	112.012218	125
Bin 18: (116.166, 275.965]	116.178019	275.619782	168.663714	125
Bin 19: (275.965, 296.023]	277.000016	296.009398	291.359281	125
Bin 20: (296.023, 303.244]	296.073953	303.230186	299.630992	125
Bin 21: (303.244, 313.616]	303.312394	313.605562	307.959367	125
Bin 22: (313.616, 325.809]	313.686982	325.807549	319.548268	125
Bin 23: (325.809, 341.539]	325.820968	341.537680	333.548892	125
Bin 24: (341.539, 359.956]	341.572848	359.955738	350.412180	125

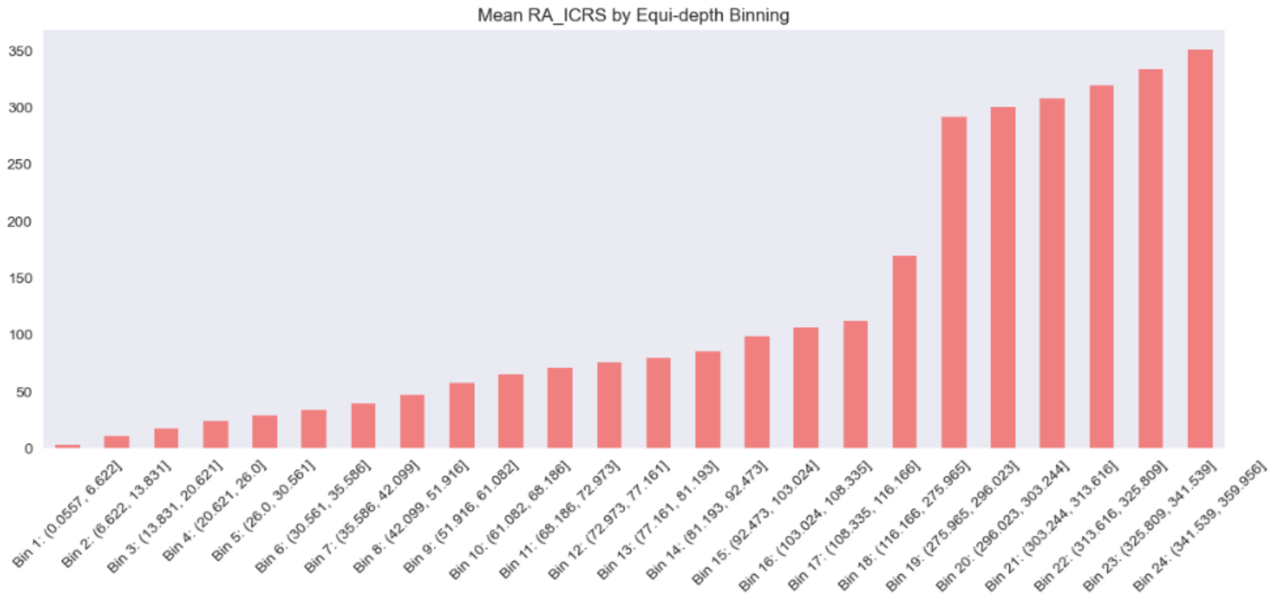
```
X[['RA_ICRS', 'RA_ICRS_EquiDepth_Bin']]\
    .merge(ra_icrs_equidepth_stats, on='RA_ICRS_EquiDepth_Bin')\
    .sort_values('RA_ICRS')
```

	RA_ICRS	RA_ICRS_EquiDepth_Bin	bin_min	bin_max	bin_mean	bin_size
2359	0.056661	Bin 1: (0.0557, 6.622]	0.056661	6.480101	3.270302	125
2361	0.120534	Bin 1: (0.0557, 6.622]	0.056661	6.480101	3.270302	125
2326	0.126239	Bin 1: (0.0557, 6.622]	0.056661	6.480101	3.270302	125
2349	0.202935	Bin 1: (0.0557, 6.622]	0.056661	6.480101	3.270302	125
2305	0.210756	Bin 1: (0.0557, 6.622]	0.056661	6.480101	3.270302	125
...	...	...	...	...	...	...
1513	358.630830	Bin 24: (341.539, 359.956]	341.572848	359.955738	350.412180	125
1527	359.005901	Bin 24: (341.539, 359.956]	341.572848	359.955738	350.412180	125
1516	359.366908	Bin 24: (341.539, 359.956]	341.572848	359.955738	350.412180	125
1515	359.474779	Bin 24: (341.539, 359.956]	341.572848	359.955738	350.412180	125
1580	359.955738	Bin 24: (341.539, 359.956]	341.572848	359.955738	350.412180	125

```
ra_icrs_equidepth_stats['bin_mean'].plot(
    kind='bar', figsize=(14, 5), fontsize=10, rot=45,
    color='lightcoral', legend=None, grid=False,
    title='Mean RA_ICRS by Equi-depth Binning'
)
```

2.4s

```
xes: title={'center': 'Mean RA_ICRS by Equi-depth Binning'}, xlabel='RA_ICRS_EquiDepth_Bin'>
```



- **DE_ICRS**: Similar to **RA_ICRS**, **DE_ICRS**’ bins are tailored to encompass equal numbers of objects, resulting in non-uniform bin widths. This reveals the density of stars across different **Declination** ranges. The bins’ means show variability in the distribution of objects, with denser areas reflected by narrower bins.

```
X['DE_ICRS_EquiDepth_Bin'] = pd.qcut(
    X['DE_ICRS'],
    q=bins_de_equidepth # Number of quantiles
)

X['DE_ICRS_EquiDepth_Bin'] = X['DE_ICRS_EquiDepth_Bin']\
    .cat.rename_categories([
        f'Bin {i + 1}: {itv}' # Renaming categories to include bin number
        for i, itv in enumerate(X['DE_ICRS_EquiDepth_Bin'].cat.categories)
    ])

de_icrs_equidepth_stats = X.groupby('DE_ICRS_EquiDepth_Bin')['DE_ICRS']\
    .agg(['min', 'max', 'mean', 'count'])\
    .rename(columns={
        'min': 'bin_min',
        'max': 'bin_max',
        'mean': 'bin_mean',
        'count': 'bin_size'
    })
de_icrs_equidepth_stats
```

DE_ICRS_EquiDepth_Bin	bin_min	bin_max	bin_mean	bin_size
Bin 1: (-36.838, -8.242]	-36.836939	-8.247184	-15.917778	167
Bin 2: (-8.242, 1.425]	-8.238922	1.403917	-3.643614	167
Bin 3: (1.425, 20.62]	1.497284	20.571353	13.281094	166
Bin 4: (20.62, 31.588]	20.629553	31.565620	26.560105	167
Bin 5: (31.588, 36.932]	31.617043	36.930892	34.402979	167
Bin 6: (36.932, 39.891]	36.952916	39.887685	38.470356	166
Bin 7: (39.891, 42.446]	39.892593	42.436257	41.213454	167
Bin 8: (42.446, 45.286]	42.469997	45.285045	43.760790	166
Bin 9: (45.286, 48.327]	45.286433	48.314320	46.767169	167
Bin 10: (48.327, 51.174]	48.339553	51.170209	49.732106	167
Bin 11: (51.174, 53.909]	51.208755	53.902529	52.489188	166
Bin 12: (53.909, 56.887]	53.911837	56.875921	55.600325	167
Bin 13: (56.887, 58.809]	56.909859	58.806497	57.818204	166
Bin 14: (58.809, 60.717]	58.809229	60.715600	59.751927	167
Bin 15: (60.717, 62.42]	60.717624	62.417258	61.569910	167
Bin 16: (62.42, 63.701]	62.435643	63.700143	63.071472	166
Bin 17: (63.701, 65.682]	63.701541	65.659199	64.521984	167
Bin 18: (65.682, 86.322]	65.716883	86.321954	71.370088	167

```
X[['DE_ICRS', 'DE_ICRS_EquiDepth_Bin']]\
    .merge(de_icrs_equidepth_stats, on='DE_ICRS_EquiDepth_Bin')\
    .sort_values('DE_ICRS')
```

	DE_ICRS	DE_ICRS_EquiDepth_Bin	bin_min	bin_max	bin_mean	bin_size
411	-36.836939	Bin 1: (-36.838, -8.242]	-36.836939	-8.247184	-15.917778	167
472	-35.472313	Bin 1: (-36.838, -8.242]	-36.836939	-8.247184	-15.917778	167
404	-34.019772	Bin 1: (-36.838, -8.242]	-36.836939	-8.247184	-15.917778	167
455	-32.571064	Bin 1: (-36.838, -8.242]	-36.836939	-8.247184	-15.917778	167
382	-32.505914	Bin 1: (-36.838, -8.242]	-36.836939	-8.247184	-15.917778	167
...	...	...	...	...	...	...
111	84.847554	Bin 18: (65.682, 86.322]	65.716883	86.321954	71.370088	167
016	84.852630	Bin 18: (65.682, 86.322]	65.716883	86.321954	71.370088	167
025	85.285817	Bin 18: (65.682, 86.322]	65.716883	86.321954	71.370088	167
075	85.704616	Bin 18: (65.682, 86.322]	65.716883	86.321954	71.370088	167
094	86.321954	Bin 18: (65.682, 86.322]	65.716883	86.321954	71.370088	167

V. Summary

1. Attribute Findings

Most attributes are continuous ratio-scaled quantitative variables suitable for many statistical analyses and visualization techniques to explore the dataset.

Temperature and **Distance** are of interest. They display a considerable number of moderate to high relationships across attributes, demonstrating the dataset's comprehensive coverage of different types of celestial objects and their varying distances. However, they exhibit significant outliers far beyond the upper quartiles, along with **Plx, PM, Lum-Flame, and Rad**, which, in particular, exhibit significant positive skewness and kurtosis as well, suggesting non-normal distributions and long tails towards the higher values with outliers significantly larger than most of the data. Moreover, **Lum-Flame** and **Rad** show extreme max values, potentially indicative of rare or unusual celestial phenomena.

The dataset also contains null values in columns (**pscol**: 2904; **GRVSmag**: 1294; **e_GRVSmag**: 1294; **Mass-Flame**: 253; **Age-Flame**: 763), which could significantly impact analysis, particularly in understanding full spectral and physical characteristics of the stars. Therefore, further strategies need to be developed to handle them, particularly for **pscol** and **GRVS**-related attributes.

These findings underscore the complexity of the Gaia dataset and indicate that any analysis/modeling should consider the potential impact of these values.

2. Relationship Findings

The correlation matrix does not show strong/clear linear relationships among most attributes. However, there are notable exceptions with the strongest correlations among magnitude attributes (**Gmag/BPmag/RPmag**). This is expected as they are all measures of stellar brightness, albeit in different spectral bands. Their slight skewness towards brighter values and relatively lower kurtosis compared to other attributes suggest a balanced distribution of star brightness, with fewer extreme outliers than in other properties.

Other correlations are relatively low to negligible correlations. For example, most attributes show weak correlations with positional data (**RA_ICRS**/**DE_ICRS**) and motion (**PM/pmRA/pmRE**), indicating these properties do not directly relate to the star's position or motion in the Galaxy. This underscores stellar characteristics are influenced by a multitude of factors, making simple linear models insufficient for describing most star properties.

Temperature's distribution suggests a bimodal grouping. These potential clusters, particularly when considering the relationship of **Temperature** and **Luminosity**, represent distinct stellar populations (such as main-sequence stars and red giants) or evolutionary stages in 2 **SpType-ELS** types. The correlation between **Teff** and **Lum-Flame** suggests a relationship aligned with the Stefan-Boltzmann law, reflecting fundamental principles of stellar physics. This association deserves rigorous statistical analysis to quantify the relationship and its implications for stellar classification.

The Equi-depth binning of **RA_ICRS** and **DE_ICRS** highlighted non-uniform distributions across the sky, with denser regions potentially mapping to the galactic plane or known star clusters. This non-uniformity is critical for mapping the Milky Way's structure and understanding the stars' distribution for further astronomical exploration.

Further statistical or ML analysis is recommended to understand the complex relationships between stellar attributes, such as the triple relationship of **Temperature**, **Luminosity**, and **Rad**, or how intrinsic and apparent **Magnitudes** relate to **Distance**. Moreover, employing clustering techniques to identify inherent groupings based on multi-attribute relationships could unveil patterns related to stellar formation, evolution, and the influence of galactic environment on stellar characteristics. Developing predictive models that incorporate all the above findings can help the Head of the Analytics Unit enhance the ability to classify stars and predict their evolution accurately.