

Formulate 20 problem statements for a given dataset using Numpy and Pandas and Apply Numpy and pandas methods to find the solution for the formulated problem statements.(PAPER REVIEW)

1. **Total Number of Reviews**

```
total_reviews = len(df)
```

2. **Average Review Score**

```
average_score = df['score'].mean()
```

3. **Standard Deviation of Review**

```
score_std_dev = df['score'].std()
```

4. **Count of Unique Reviewers**

```
unique_reviewers = df['reviewer_id'].nunique()
```

5. **Count of Unique Papers Reviewed**

```
unique_papers = df['paper_id'].nunique()
```

6. **Average Number of Reviews per Paper**

```
avg_reviews_per_paper = df.groupby('paper_id')['reviewer_id'].count().mean()
```

7. **Average Number of Reviews per Reviewer**

```
avg_reviews_per_reviewer = df.groupby('reviewer_id')['paper_id'].count().mean()
```

8. **Average Length of Review Text**

```
avg_review_length = df['review_text'].str.len().mean()
```

9. **Number of Reviews with Score Above 8**

```
high_score_reviews = df[df['score'] > 8].shape[0]
```

10. **Number of Reviews with Score Below 4**

```
low_score_reviews = df[df['score'] < 4].shape[0]
```

11. **Distribution of Review Scores**

```
score_distribution = df['score'].value_counts().sort_index()
```

12. **Average Review Score per Paper**

```
avg_score_per_paper = df.groupby('paper_id')['score'].mean()
```

13. **Average Review Score per Reviewer**

```
avg_score_per_reviewer = df.groupby('reviewer_id')['score'].mean()
```

14. **Top 5 Reviewers by Number of Reviews**

```
top_reviewers = df['reviewer_id'].value_counts().head(5)
```

15. **Top 5 Papers by Number of Reviews**

```
top_papers = df['paper_id'].value_counts().head(5)
```

16. **Average Time Taken to Review (in Days)**

```
df['submission_date'] = pd.to_datetime(df['submission_date'])
```

```
df['review_date'] = pd.to_datetime(df['review_date'])
```

Formulate 20 problem statements for a given dataset using Numpy and Pandas and Apply Numpy and pandas methods to find the solution for the formulated problem statements.(PAPER REVIEW)

```
df['review_time'] = (df['review_date'] - df['submission_date']).dt.days  
avg_review_time = df['review_time'].mean()
```

#### **17. Number of Accepted Papers**

```
accepted_papers = df[df['decision'] == 'accept']['paper_id'].nunique()
```

#### **18. Number of Rejected Papers**

```
rejected_papers = df[df['decision'] == 'reject']['paper_id'].nunique()
```

#### **19. Correlation Between Review Score and Review Length**

```
df['review_length'] = df['review_text'].str.len()  
correlation = df['score'].corr(df['review_length'])
```

#### **20. Monthly Distribution of Reviews**

```
df['review_month'] = df['review_date'].dt.to_period('M')  
monthly_reviews = df['review_month'].value_counts().sort_index()
```