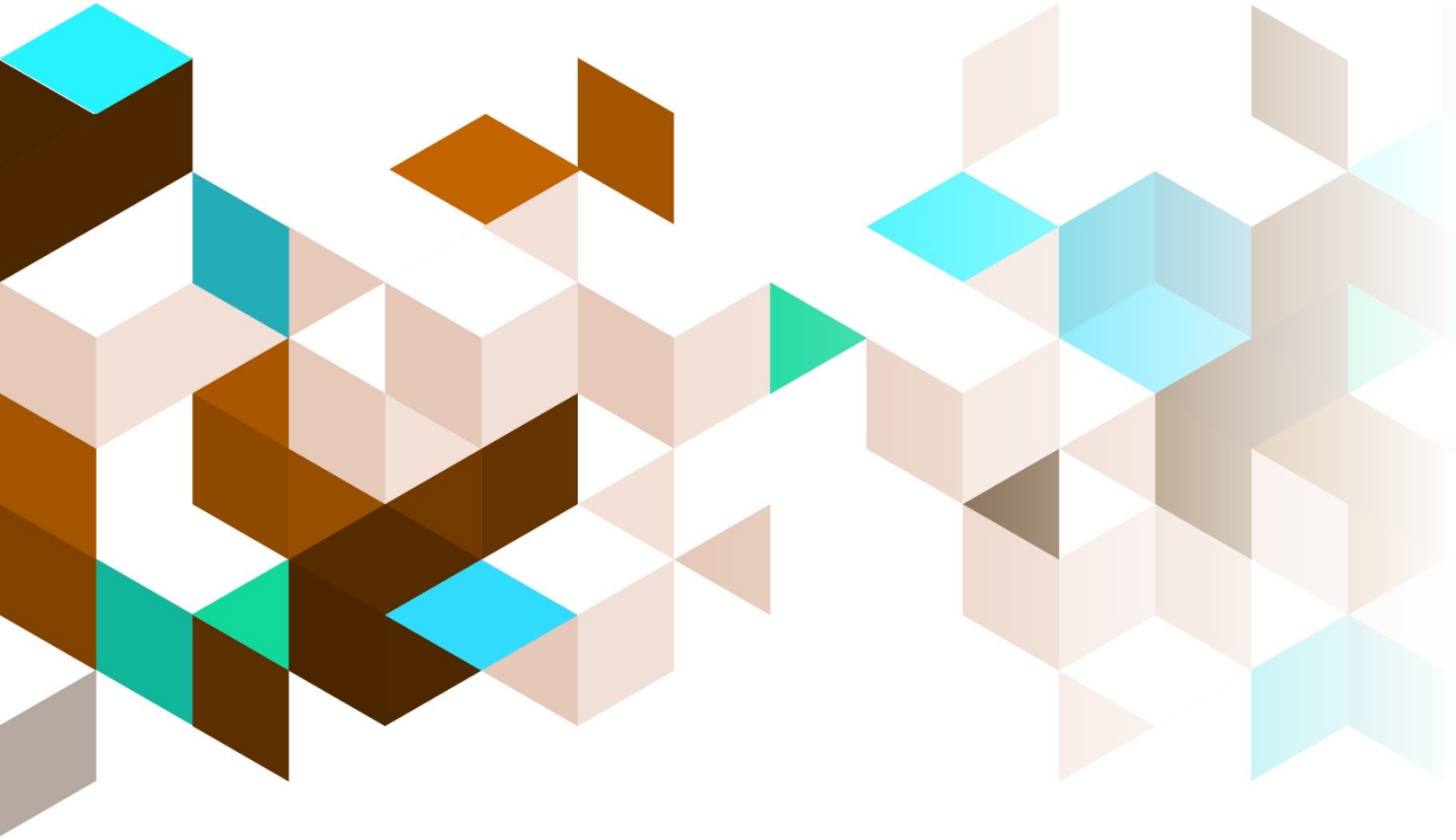


Healthcare EDA Analysis

Kaushik Palepu

Background and Objective

Background:

- You are a data scientist working for a wearable technology company that produces smartwatches with vital signs sensors. These sensors monitor heart rate and PPG (Photoplethysmography) signals, which include variations in green, red, and infrared light. One of the key features of your company's smartwatch is its ability to detect and alert users to potential drowsiness based on their physiological data.

Objective:

- Your task is to perform an Exploratory Data Analysis (EDA) on a dataset collected from these smartwatches. The dataset includes various physiological parameters along with a 'drowsiness' label, which indicates the level of sleepiness based on an adapted Karolinska Sleepiness Scale (KSS).

Dataset Details- Kaggle –Drowsiness dataset

•**heartRate:** Heart rate readings from the smartwatch sensors.

•**ppgGreen**, **ppgRed**, **ppgIR**: PPG (Photoplethysmography) sensor readings in green, red, and infrared wavelengths respectively.

•**drowsiness:** Label indicating the level of drowsiness based on an adapted Karolinska Sleepiness Scale (KSS). Values range from 0.0 to 2.0, where 0.0 represents alertness and 2.0 represents significant drowsiness.

Data Loading and Familiarising

- All the columns are numerical type.
- There is a total of 4890260 entries.

```
[254]: #Data Loading and Familiarising
print(df.head())
print('\n')
print(df.info())
print('\n')
print(df.describe())
```

```
   heartRate   ppgGreen     ppgRed      ppgIR  drowsiness  heart_rate  \
0       54.0  1584091.0  5970731.0  6388383.0         0.0        54.0
1       54.0  1584091.0  5971202.0  6392174.0         0.0        54.0
2       54.0  1581111.0  5971295.0  6391469.0         0.0        54.0
3       54.0  1579343.0  5972599.0  6396137.0         0.0        54.0
4       54.0  1579321.0  5971906.0  6392898.0         0.0        54.0

   ppgGreen_cleaned  ppgRed_cleaned  ppgIR_cleaned  heartRate_cleaned
0         1584091.0       5970731.0      6388383.0               54.0
1         1584091.0       5971202.0      6392174.0               54.0
2         1581111.0       5971295.0      6391469.0               54.0
3         1579343.0       5972599.0      6396137.0               54.0
4         1579321.0       5971906.0      6392898.0               54.0


<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4890260 entries, 0 to 4890259
Data columns (total 10 columns):
 #   Column             Dtype
---  ------             -----
 0   heartRate          float64
 1   ppgGreen           float64
 2   ppgRed             float64
 3   ppgIR              float64
 4   drowsiness         float64
 5   heart_rate         float64
 6   ppgGreen_cleaned   float64
 7   ppgRed_cleaned     float64
 8   ppgIR_cleaned      float64
 9   heartRate_cleaned  float64
dtypes: float64(10)
memory usage: 373.1 MB
None


          heartRate      ppgGreen        ppgRed         ppgIR    drowsiness  \
count  4.890260e+06  4.890260e+06  4.890260e+06  4.890260e+06  4.890260e+06
mean   7.814245e+01  2.073589e+06  5.643653e+06  5.728191e+06  8.593592e-01
std    1.296635e+01  4.418773e+05  3.909626e+05  4.313052e+05  8.370285e-01
min    5.000000e+01  5.897580e+05  4.441989e+06  4.409976e+06  0.000000e+00
25%    6.800000e+01  1.780621e+06  5.368700e+06  5.402542e+06  0.000000e+00
50%    7.800000e+01  2.044658e+06  5.646039e+06  5.818748e+06  1.000000e+00
75%    8.700000e+01  2.333117e+06  5.927128e+06  6.016016e+06  2.000000e+00
max    1.190000e+02  3.530798e+06  6.842637e+06  7.061799e+06  2.000000e+00

         heart_rate  ppgGreen_cleaned  ppgRed_cleaned  ppgIR_cleaned  \
count  4.890260e+06      4.890260e+06    4.890260e+06   4.890260e+06
mean   7.813246e+01      2.073666e+06    5.643649e+06   5.728154e+06
std    1.293650e+01      4.292590e+05    3.908602e+05   4.311964e+05
min    5.000000e+01      9.518770e+05    4.531058e+06   4.482331e+06
25%    6.800000e+01      1.780621e+06    5.368700e+06   5.402542e+06
50%    7.800000e+01      2.044658e+06    5.646039e+06   5.818748e+06
75%    8.700000e+01      2.333117e+06    5.927128e+06   6.016016e+06
max    1.155000e+02      3.161861e+06    6.764770e+06   6.936227e+06

       heartRate_cleaned
count       4.890260e+06
mean        7.813246e+01
std         1.293650e+01
min         5.000000e+01
25%         6.800000e+01
50%         7.800000e+01
75%         8.700000e+01
max         1.155000e+02
```

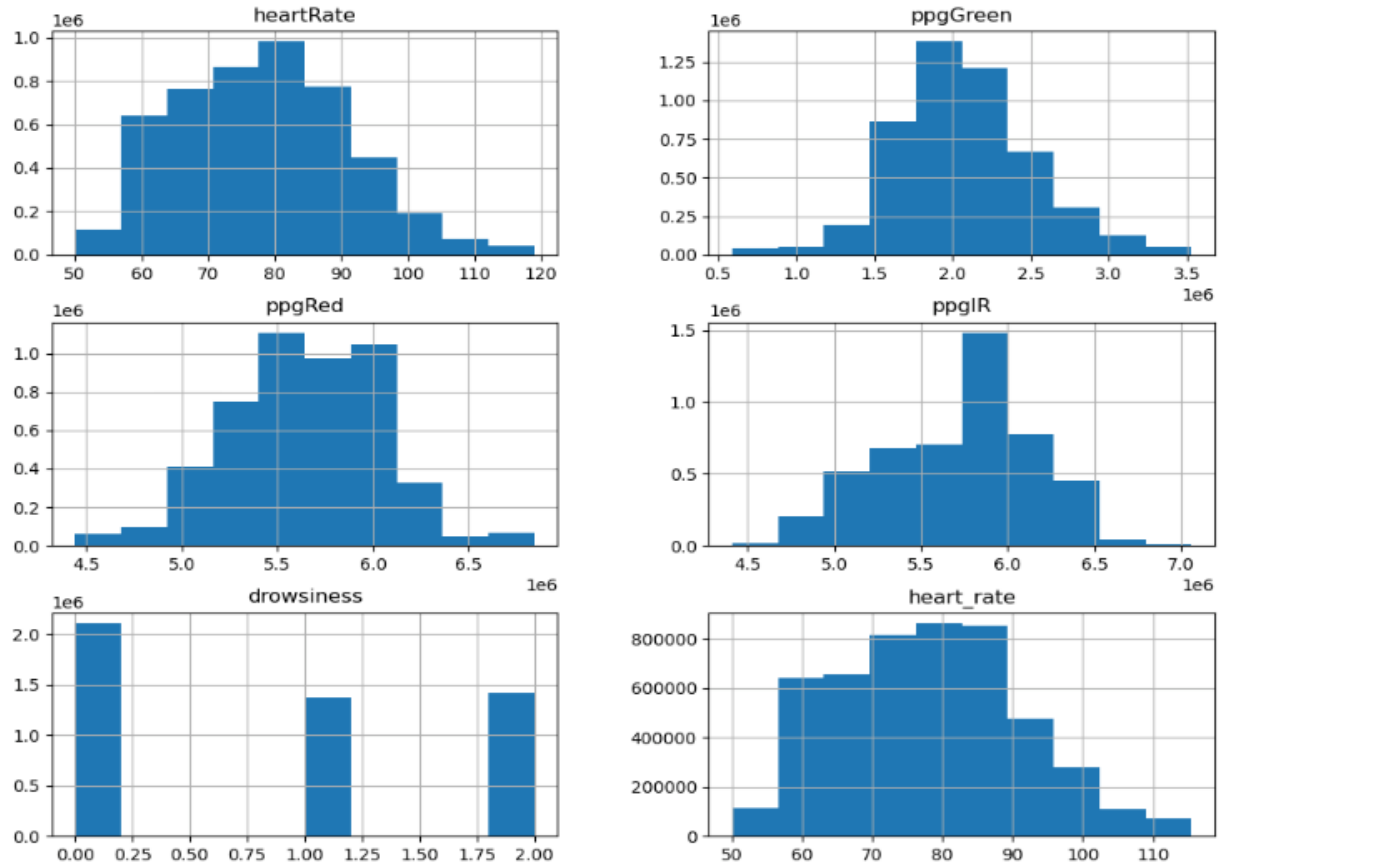
Data Distribution

- The variable distribution is fairly shown in the graphs
- While heartRate variable looks right skewed, the rest of the variables look to be non-skewed and all are normally distributed.
- Drowsiness variable is distributed only across 3 points(0,1,2)

```
[31]: #Check for any missing values
      print(df.isnull().sum())
```

```
heartRate     0
ppgGreen      0
ppgRed        0
ppgIR         0
drowsiness    0
dtype: int64
```

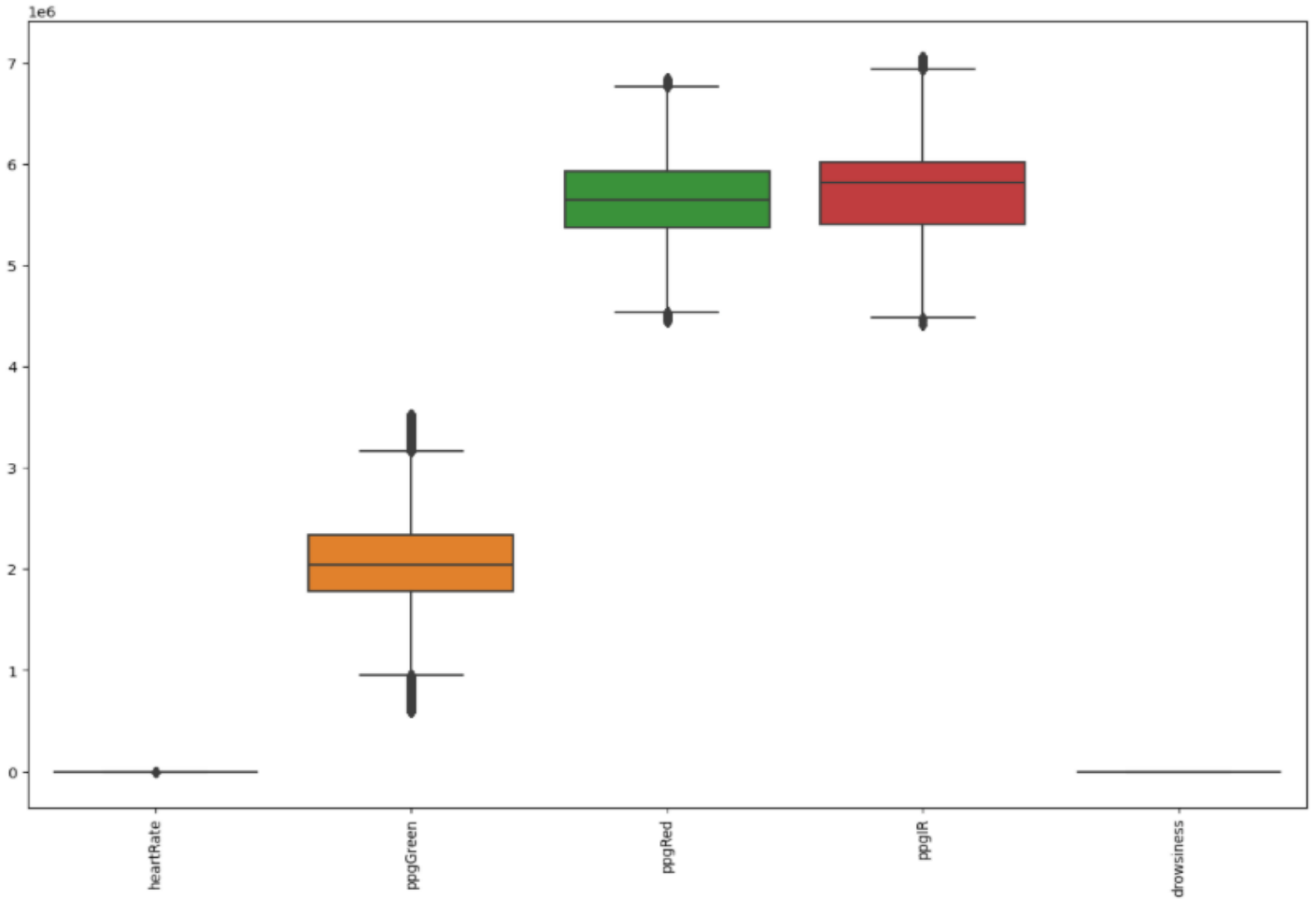
```
[218]: #Histogram
       df.hist(bins=10, figsize=(12,9))
       plt.show()
```



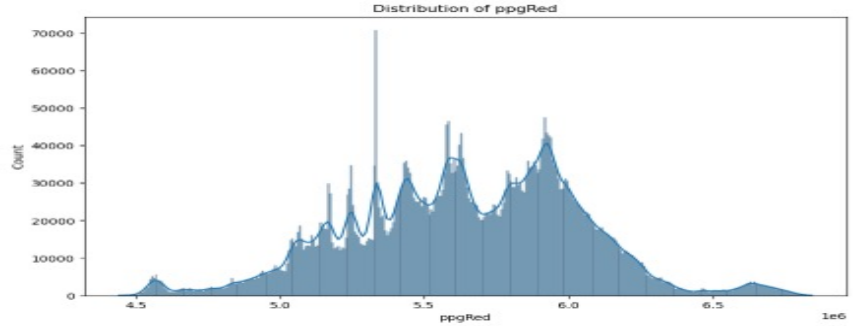
Boxplot visualisation

- The box plot shows the outliers and the distribution of the variable values
- ppgGreen is having high outliers.
- heartRate and drowsiness is having least outliers.

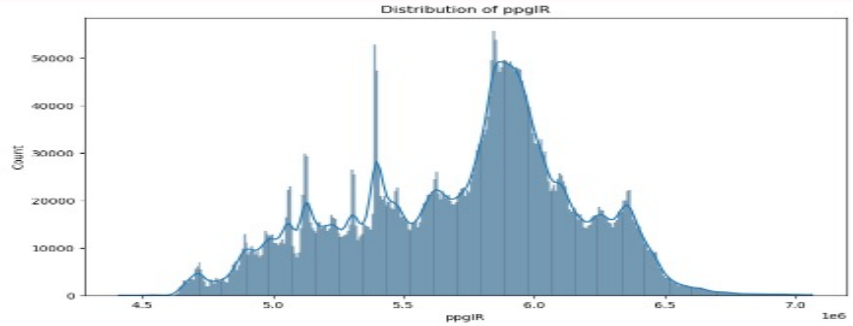
```
[39]: #Box plot
plt.figure(figsize=(15,10))
sns.boxplot(data=df)
plt.xticks(rotation=90)
plt.show()
```



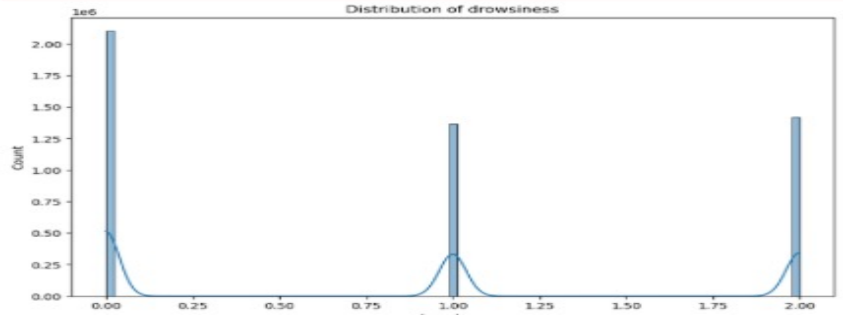
Distribution Visualization-Histograms




```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na
option is deprecated and will be removed in a future version. Convert inf values to NaN before oper
ating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

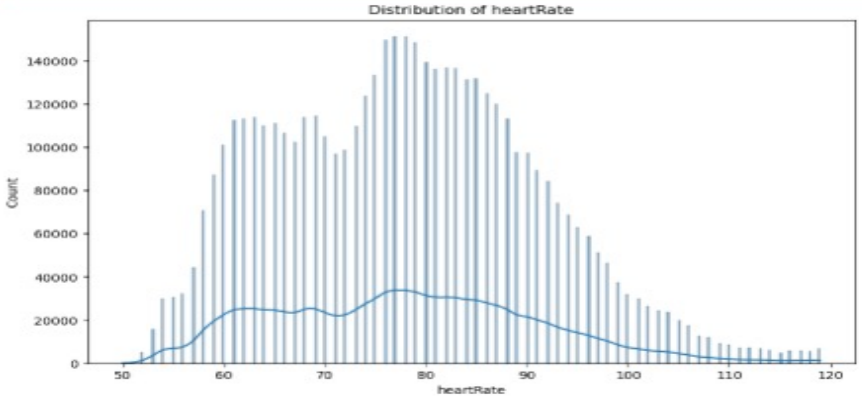



```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na
option is deprecated and will be removed in a future version. Convert inf values to NaN before oper
ating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

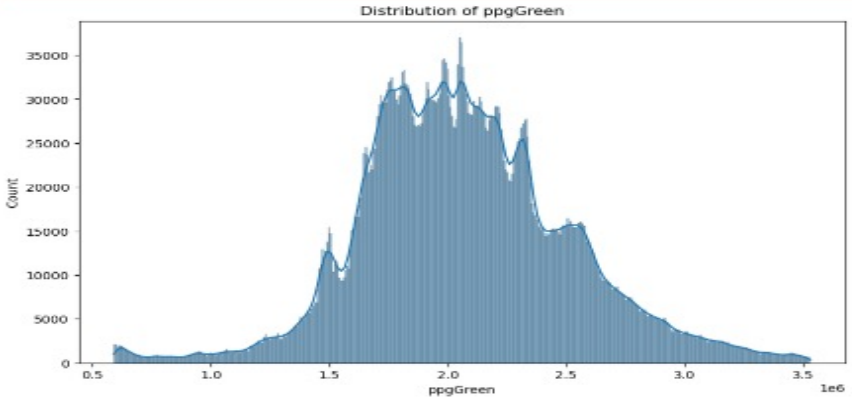



```
[51…
for i in df:
    plt.figure(figsize=(10, 6))
    sns.histplot(df[i], kde=True)
    plt.title(f'Distribution of {i}')
    plt.show()
```

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na
option is deprecated and will be removed in a future version. Convert inf values to NaN before oper
ating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```




```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na
option is deprecated and will be removed in a future version. Convert inf values to NaN before oper
ating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

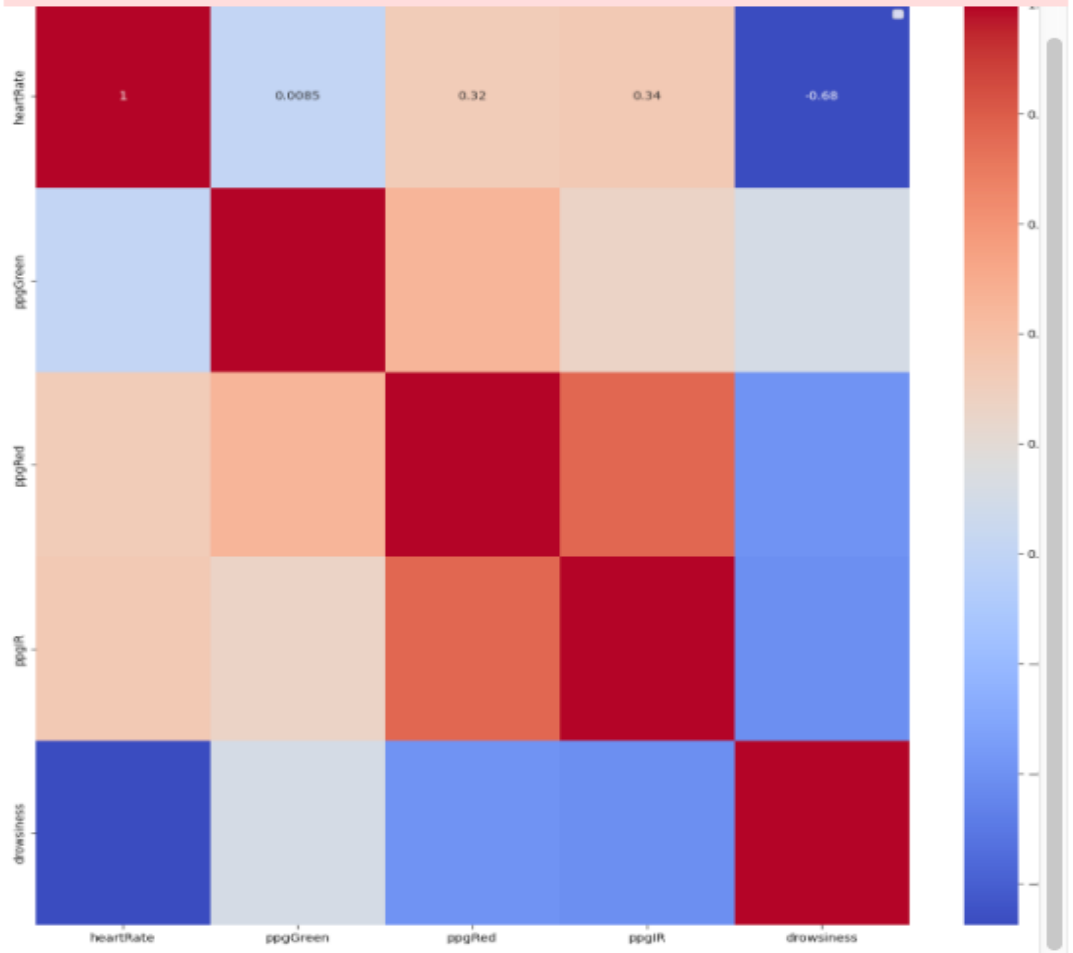


Correlation Matrix

- Through the heatmap, it is understood that there is a positive correlation between drowsiness and ppgGreen.
- A moderate correlation exists between drowsiness and both ppgRed and ppgIR.
- A negative correlation is existing between drowsiness and heartRate.

```
[19…
corr_matrix= df.corr()
plt.figure(figsize=(15, 15))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Heatmap of correlation between variables')
plt.legend()
plt.show()
```

```
No artists with labels found to put in legend.  Note that artists whose label start with an undersc
ore are ignored when legend() is called with no argument.
```



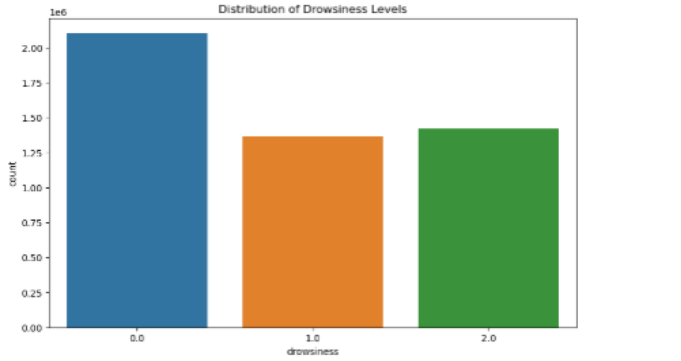
ANOVA test findings

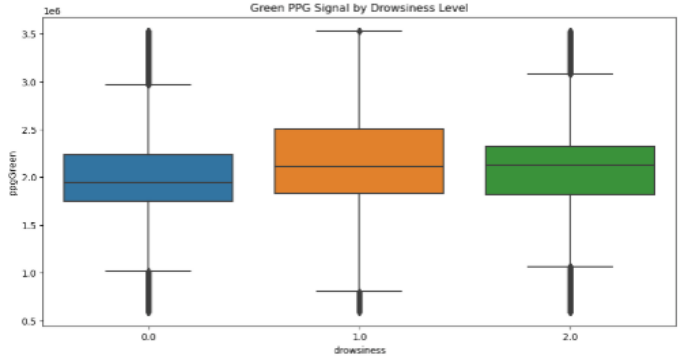
- The ANOVA test results explain that there is no significant difference between drowsiness and each other variable.

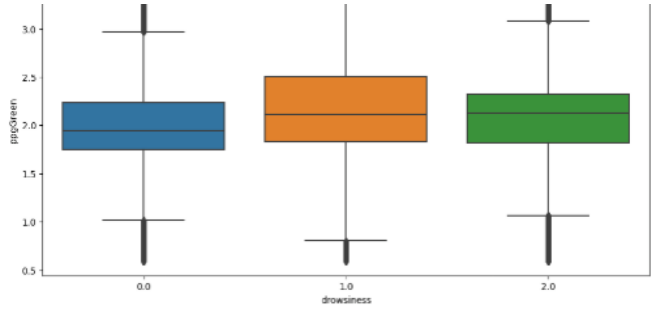
```
[25…
# ANOVA test to compare heart rates across drowsiness levels

drowsiness_levels = df['drowsiness'].unique()
heart_rates = [df[df['drowsiness'] == i]['heartRate'] for i in drowsine
f_statistic, p_value = stats.f_oneway(*heart_rates)
print(f"ANOVA results heartrate: F-statistic = {f_statistic}, p-value =


drowsiness_levels = df['drowsiness'].unique()
ppgGreen_levels = [df[df['drowsiness'] == i]['ppgGreen'] for i in drows
f_statistic, p_value = stats.f_oneway(*ppgGreen_levels)
print(f"ANOVA results ppgGreen: F-statistic = {f_statistic}, p-value =


drowsiness_levels = df['drowsiness'].unique()
ppgRed_levels = [df[df['drowsiness'] == i]['ppgRed'] for i in drowsines
f_statistic, p_value = stats.f_oneway(*ppgRed_levels)
print(f"ANOVA results ppgRed: F-statistic = {f_statistic}, p-value = {p


drowsiness_levels = df['drowsiness'].unique()
ppgIR_levels = [df[df['drowsiness'] == i]['ppgIR'] for i in drowsiness_
f_statistic, p_value = stats.f_oneway(*ppgIR_levels)
print(f"ANOVA results ppgIR: F-statistic = {f_statistic}, p-value = {p_

ANOVA results heartrate: F-statistic = 2256845.7001167205, p-value = 0.
ANOVA results ppgGreen: F-statistic = 60758.16828651622, p-value = 0.0
ANOVA results ppgRed: F-statistic = 436362.17633283796, p-value = 0.0
ANOVA results ppgIR: F-statistic = 549350.4240909232, p-value = 0.0
```

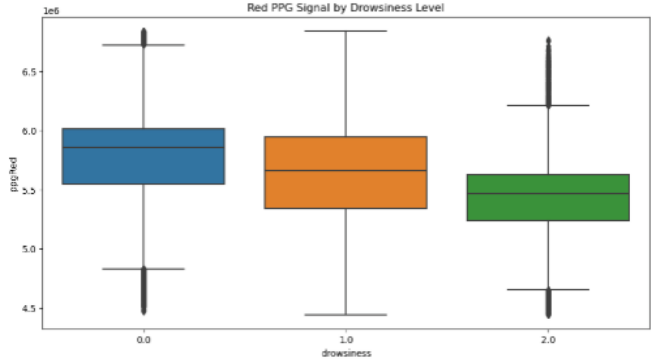
Boxplot visualization and distribution

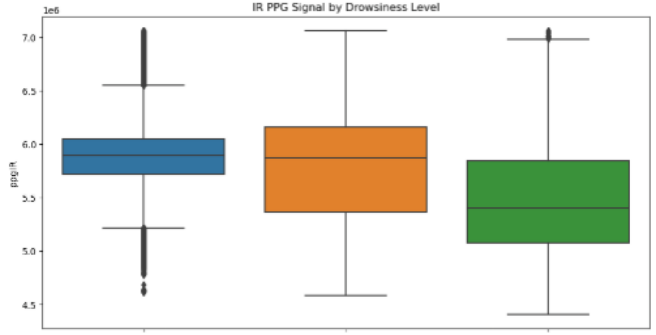
```
plt.title('Distribution of Drowsiness Levels')
plt.show()

# Box plots for PPG signals across drowsiness levels
plt.figure(figsize=(12, 6))
sns.boxplot(x='drowsiness', y='ppgGreen', data=df)
plt.title('Green PPG Signal by Drowsiness Level')
plt.show()

plt.figure(figsize=(12, 6))
sns.boxplot(x='drowsiness', y='ppgRed', data=df)
plt.title('Red PPG Signal by Drowsiness Level')
plt.show()

plt.figure(figsize=(12, 6))
sns.boxplot(x='drowsiness', y='ppgIR', data=df)
plt.title('IR PPG Signal by Drowsiness Level')
plt.show()
```











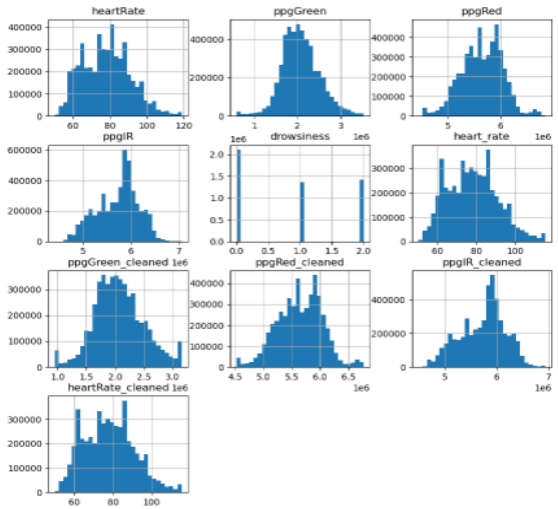
- Each variable is box-plotted against drowsiness to understand the outlier points and the distribution mapping of drowsiness with respect to other variables.

Outlier calculation and data cleaning for further analysis

```
# Outliers (IQR method)

Q1 = df['heartRate'].quantile(0.25)
Q3 = df['heartRate'].quantile(0.75)
IQR = Q3 - Q1
lower_bound_range = Q1 - 1.5 * IQR
upper_bound_range = Q3 + 1.5 * IQR
df['heartRate_cleaned'] = df['heartRate'].clip(lower_bound_range, upper_bound_range)

Q1 = df['ppgGreen'].quantile(0.25)
Q3 = df['ppgGreen'].quantile(0.75)
IQR = Q3 - Q1
lower_bound_range = Q1 - 1.5 * IQR
upper_bound_range = Q3 + 1.5 * IQR
df['ppgGreen_cleaned'] = df['ppgGreen'].clip(lower_bound_range, upper_bound_range)


Q1 = df['ppgRed'].quantile(0.25)
Q3 = df['ppgRed'].quantile(0.75)
IQR = Q3 - Q1
lower_bound_range = Q1 - 1.5 * IQR
upper_bound_range = Q3 + 1.5 * IQR
df['ppgRed_cleaned'] = df['ppgRed'].clip(lower_bound_range, upper_bound_range)

Q1 = df['ppgIR'].quantile(0.25)
Q3 = df['ppgIR'].quantile(0.75)
IQR = Q3 - Q1
lower_bound_range = Q1 - 1.5 * IQR
upper_bound_range = Q3 + 1.5 * IQR
df['ppgIR_cleaned'] = df['ppgIR'].clip(lower_bound_range, upper_bound_range)

df
```

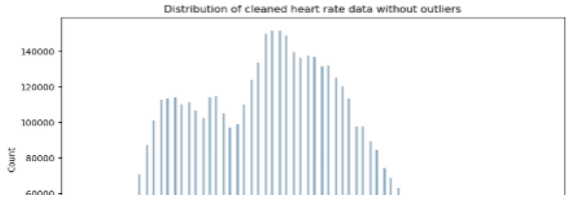
ppgIR	drowsiness	heart_rate	ppgGreen_cleaned	ppgRed_cleaned	ppgIR_cleaned	heartRat
6388383.0	0.0	54.0	1584091.0	5970731.0	6388383.0	
6392174.0	0.0	54.0	1584091.0	5971202.0	6392174.0	
6391469.0	0.0	54.0	1581111.0	5971295.0	6391469.0	
6396137.0	0.0	54.0	1579343.0	5972599.0	6396137.0	
6392898.0	0.0	54.0	1579321.0	5971906.0	6392898.0	
...	...	...	...	...	...	
6356797.0	2.0	63.0	2286384.0	5783226.0	6356797.0	
6357004.0	2.0	63.0	2289887.0	5783786.0	6357004.0	
6358348.0	2.0	63.0	2291928.0	5784221.0	6358348.0	
6358565.0	2.0	63.0	2295386.0	5785012.0	6358565.0	
6357466.0	2.0	63.0	2296992.0	5783386.0	6357466.0	

```
#Histogram
df.hist(bins=30, figsize=(10,10))
plt.show()
```




```
plt.figure(figsize=(10, 6))
sns.histplot(df['heart_rate'], kde=True)
plt.title(f'Distribution of cleaned heart rate data without outliers')
plt.show()
```

```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119: FutureWarning: use_inf_as_na
option is deprecated and will be removed in a future version. Convert inf values to NaN before oper
ating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



- There is a significant difference in the data point distribution of the heartRate, when its cleaned.
- The other variables remain fairly unaffected.

Conclusions of EDA Analysis

- ppgGreen sensor readings are the best indicator of drowsiness levels
- Heart rate has a negative correlation with drowsiness, meaning that drowsy people tend to have a lower heart rate.
- Additional EDA analysis and other tests might reveal further insights on the dataset.

Thank you