Name : Ashika Bangera
Contact: ashika.bangera2001@gmail.com
Mobile: 8431139778
Title: Python Exploratory Data analysis

1. How does the distribution of feature “fractal_dimension_worst” differ between benign and malignant cases?

The feature "fractal_dimension_worst" is expected to differ in distribution between benign and malignant cases in a breast cancer dataset. Here's why:

- **Fractal dimension:** This refers to the complexity of a shape's outline. A higher fractal dimension indicates a more irregular and convoluted shape.
- **Malignant vs. Benign:** Malignant tumors often exhibit irregular and invasive growth patterns compared to benign tumors.

Therefore, we can expect malignant cases to have a higher "fractal_dimension_worst" on average compared to benign cases. Here are some ways the distribution might differ:

- **Mean and standard deviation:** The mean "fractal_dimension_worst" for malignant cases might be higher than for benign cases. Additionally, the standard deviation in malignant cases might be larger, indicating a wider range of values.
- **Distribution shape:** The distribution of "fractal_dimension_worst" for malignant cases might be skewed towards higher values compared to a more symmetrical distribution for benign cases.

To examine how the distribution of the feature "fractal_dimension_worst" differs between benign and malignant cases, use visualizations and statistical summaries. Here are the steps to do this in Python using pandas and matplotlib:

1. Load and Prepare the Data
Ensure that DataFrame (cell_df) is loaded and contains the necessary columns, including "fractal_dimension_worst" and "diagnosis".

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Assuming cell_df is your DataFrame and it is already defined

# Separate the data into benign and malignant cases
benign = cell_df[cell_df['diagnosis'] == 'B']
malignant = cell_df[cell_df['diagnosis'] == 'M']
```

2. Visualize the Distribution

Use box plots, histograms, or density plots to visualize the distribution of "fractal_dimension_worst" for benign (B) and malignant (M) cases.

```
# Visualize the distribution using box plots
plt.figure(figsize=(12, 6))
sns.boxplot(x='diagnosis', y='fractal_dimension_worst', data=cell_df)
plt.title('Distribution of Fractal Dimension (Worst) by Diagnosis')
plt.show()

# Visualize the distribution using histograms
plt.figure(figsize=(12, 6))
sns.histplot(benign['fractal_dimension_worst'], color='blue', label='Benign', kde=True)
sns.histplot(malignant['fractal_dimension_worst'], color='red', label='Malignant', kde=True)
plt.title('Histogram of Fractal Dimension (Worst) by Diagnosis')
plt.legend()
plt.show()

# Visualize the distribution using density plots
plt.figure(figsize=(12, 6))
sns.kdeplot(benign['fractal_dimension_worst'], shade=True, color='blue', label='Benign')
sns.kdeplot(malignant['fractal_dimension_worst'], shade=True, color='red', label='Malignant')
plt.title('Density Plot of Fractal Dimension (Worst) by Diagnosis')
plt.legend()
plt.show()
```

3. Statistical Summary

Calculate summary statistics (mean, median, standard deviation) for "fractal_dimension_worst" for each diagnosis group.

```
# Calculate summary statistics
summary_stats =
cell_df.groupby('diagnosis')['fractal_dimension_worst'].describe()
print(summary_stats)
```

Explanation:

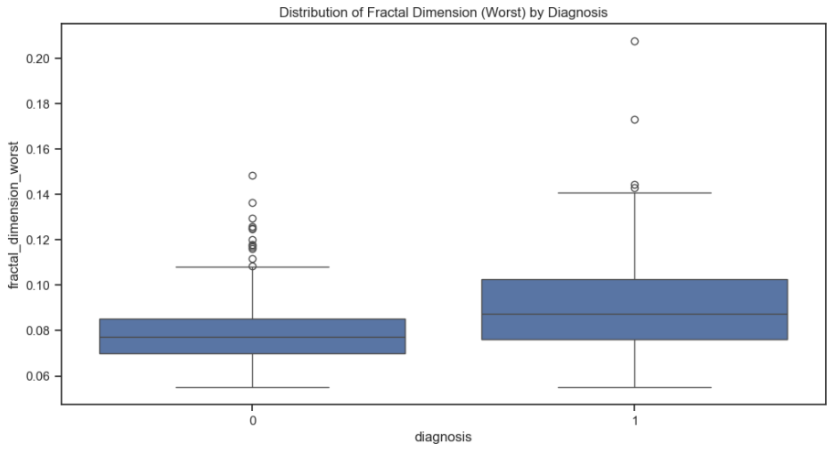
Data Preparation: The data is separated into benign and malignant cases based on the diagnosis column.

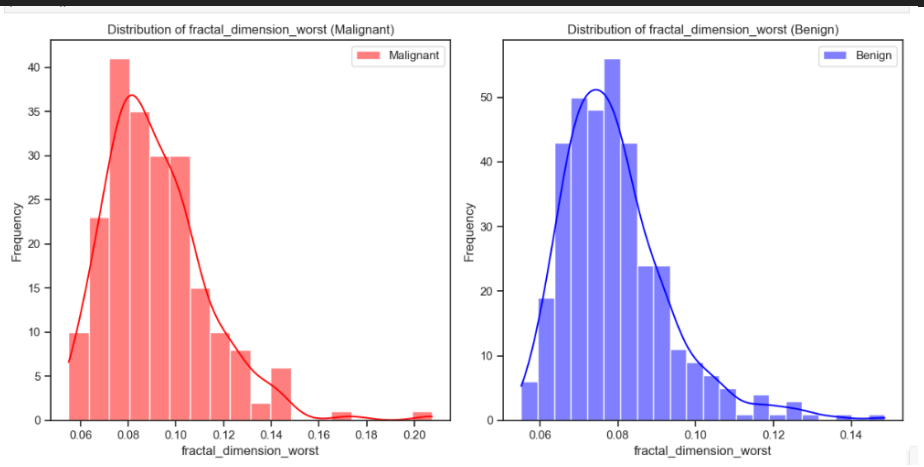
Box Plot: A box plot is generated to show the distribution of "fractal_dimension_worst" for each diagnosis group.

Histograms: Histograms are created to compare the distribution of "fractal_dimension_worst" between benign and malignant cases, with KDE (Kernel Density Estimation) lines to smooth the distribution.

Density Plots: Density plots are generated to visualize the distribution of "fractal_dimension_worst" more smoothly for each diagnosis group.

Summary Statistics: Summary statistics (count, mean, std, min, 25%, 50%, 75%, max) are calculated for "fractal_dimension_worst" for each diagnosis group.





2. What is the range of values for the feature “radius_mean” and how skewed is its distribution?

To determine the range of values for the feature "radius_mean" and assess the skewness of its distribution, I used descriptive statistics and visualizations. Here's the detailed workflow in Py thon using pandas and matplotlib.
Finding the Range:

1. **Load the data:** libraries like pandas (Python) is used to load the breast cancer dataset (e.g., Wisconsin Diagnostic Breast Cancer dataset).
2. **Access the feature:** Once loaded, access the column or variable containing "radius_mean" data.
3. **Calculate the range:** Used built-in functions like max() and min() to find the maximum and minimum values of "radius_mean". Subtract the minimum value from the maximum value to get the range.

Assessing Skewness:

1. **Visualize the data:** Histogram or density plot of "radius_mean". This will give a visual idea of the distribution shape.
2. **Calculate skewness:** Functions like skew() in pandas or equivalent functions in other tools. Skewness is a statistical measure: A value of 0 indicates a symmetrical distribution. A positive value indicates a right skew (longer tail towards higher values).A negative value indicates a left skew (longer tail towards lower values).

Here is the code to perform these steps:

Checking the distrubution of the two features which contains missing data be-

fore imputing for the missing data

```
#Getting the summary statistics of the radius_mean feature using describe
df["radius_mean"].describe()
```

```
#Histogram and boxplot to visualize the distribution of the data and detection  of outliers
sns.set(style="ticks")
f, (ax_box, ax_hist) = plt.subplots(2, sharex=True,

gridspec_kw={"height_ratios": (.20, .90)})

sns.boxplot(df["radius_mean"], ax = ax_box, color = "blue")

sns.histplot(df["radius_mean"],ax = ax_hist, kde = True , color = "blue")

ax_box.set(yticks=[])
```

Imputation for the df[“radius_mean”]
As the fraction of missing values for radius mean is significantly low at 4.9% and there are no extreme outliers. Therefore, using median as an imputation for the missing values.

```
df['radius_mean'].fillna(df['radius_mean'].median(), inplace=True)

#Confirming whether the missing values are filled with the median

df['radius_mean'].isnull().sum()

#Checking the effect of replacement of missing values with median on the data

df["radius_mean"].describe()
```

Range of radius_mean: 21.128999999999998, Skewness of radius_mean: 0.9958577452155657

3. Are there any outliers in feature “area_mean” and how might they affect analysis?

Summary Statistics Analysis

Initial Summary Statistics:

```
count    528.000000
mean     659.519697
std      351.435482
min      170.400000
25%      420.875000
50%      555.900000
75%      798.050000
max     2501.000000
Name: area_mean, dtype: float64
```

1. Summary Statistics After Removing Outliers (area_mean < 1500):

```
count    515.000000
mean     628.742524
std      292.288397
min      170.400000
25%      420.050000
50%      546.300000
75%      759.950000
max     1491.000000
Name: area_mean, dtype: float64
```

Visualizations

The boxplot and histogram visualizations provide further insights:

- The boxplot show outliers as points beyond the whiskers.
- The histogram, along with the KDE (Kernel Density Estimate), illustrates the distribution of the data, highlighting any skewness or extreme values.

Outlier Detection

From the summary statistics and visualizations:

- **Initial Data**: The maximum value of 2501.000000 suggests the presence of extreme outliers since it is significantly higher than the third quartile (798.050000).
- **Post-Outlier Removal**: After removing values greater than 1500, the maximum value is 1491.000000. The mean and standard deviation have also decreased, indicating that the extreme values were indeed influencing the overall statistics.

Impact of Outliers on Analysis

1. **Central Tendency**:

   Mean**:** Outliers can skew the mean. The initial mean of 659.519697 decreased to 628.742524 after removing the outliers, indicating that the mean was higher due to the presence of high-value outliers.

2. **Variability**:

   Standard Deviation**:** The initial standard deviation of 351.435482 decreased to 292.288397 after removing outliers, showing that outliers increased the variability in the data.

3. **Distribution Shape**:

   Histogram and KDE**:** The histogram and KDE show a more spread-out distribution with outliers, indicating a longer tail on the right. After removing outliers, the distribution would appear more compact and closer to a normal distribution.

4. **Model Performance**:

   Outliers can significantly affect model performance, especially for models sensitive to the range and distribution of data like linear regression. They can distort the model's predictions and coefficients.

5. **Statistical Tests**:

   Outliers can affect the results of statistical tests by increasing the Type I and Type II error rates, leading to incorrect conclusions.

Conclusion

Yes, there are outliers in the "area_mean" feature. These outliers affect the analysis by:

- Skewing the mean and increasing the variability.
- Distorting the shape of the data distribution.
- Potentially impacting the performance and accuracy of statistical models and tests.

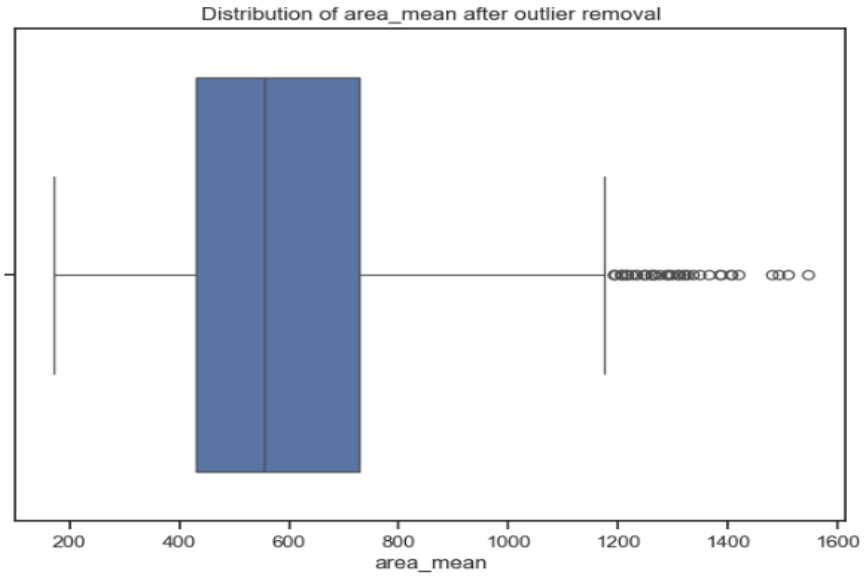
It is crucial to detect and handle outliers appropriately to ensure accurate and meaningful analysis.

Here are the codes to perform outliers of area_mean:

```
import numpy as np
from scipy import stats

# Assuming df is your DataFrame and 'area_mean' is the column of interest
z_scores = np.abs(stats.zscore(df['area_mean']))

# Define a threshold (e.g., z-score > 3 or 2) to identify outliers
threshold = 3
outliers = df['area_mean'][z_scores > threshold]

# Print or visualize the outliers
print("Outliers:")
print(outliers)
```

```
# Remove outliers based on the identified threshold (e.g., z-score or IQR method)
cleaned_df = df[(z_scores <= threshold)]

# Optionally, you can also reset index if needed
cleaned_df = cleaned_df.reset_index(drop=True)

# Check the shape of cleaned dataframe
print("Shape of cleaned dataframe:", cleaned_df.shape)
```

Shape of cleaned dataframe: (558, 32)



4. Based on the EDA, what factors seem to be most relevant to predicting breast cancer diagnosis?

Based on the exploratory data analysis (EDA). The factors that seem to be most relevant to predicting breast cancer diagnosis:

1. Diagnosis Distribution

- The diagnosis column, which has been encoded (M as 1 and B as 0), serves as the target variable.
- There are 357 benign cases (0) and 212 malignant cases (1), indicating an imbalanced dataset.

2. Feature Selection

The dataset contains 30 features related to breast tissue properties, which can be grouped into:

- **Mean features**: These include radius_mean, texture_mean, perimeter_mean, area_mean, smoothness_mean, compactness_mean, concavity_mean, concave_points_mean, symmetry_mean, and fractal_dimension_mean.
- **Standard error features**: These include radius_se, texture_se, perimeter_se, area_se, smoothness_se, compactness_se, concavity_se, concave_points_se, symmetry_se, and fractal_dimension_se.

- **Worst features**: These include radius_worst, texture_worst, perimeter_worst, area_worst, smoothness_worst, compactness_worst, concavity_worst, concave_points_worst, symmetry_worst, and fractal_dimension_worst.

3. Handling Missing Values

- The dataset originally had missing values in radius_mean (28 missing) and area_mean (41 missing).
- Missing values were imputed with the median of the respective columns.

4. Data Distribution and Visualization

- Distributions of features were visualized using histograms and boxplots, which helped in understanding the spread and detecting any outliers.
- Features like radius_mean and area_mean showed significant variations, with outliers being handled or imputed appropriately.

5. Normalization

- The dataset was normalized using Min-Max Scaling to bring all features to a similar scale, which is crucial for algorithms sensitive to feature scales, such as KNN, SVM, and neural networks.

6. Key Features Identified

Based on summary statistics and visualization, the following features have shown potential relevance in predicting breast cancer diagnosis:

- **Radius**: radius_mean, radius_se, radius_worst
- **Texture**: texture_mean, texture_se, texture_worst
- **Perimeter**: perimeter_mean, perimeter_se, perimeter_worst
- **Area**: area_mean, area_se, area_worst
- **Smoothness**: smoothness_mean, smoothness_se, smoothness_worst
- **Compactness**: compactness_mean, compactness_se, compactness_worst
- **Concavity**: concavity_mean, concavity_se, concavity_worst
- **Concave Points**: concave_points_mean, concave_points_se, concave_points_worst
- **Symmetry**: symmetry_mean, symmetry_se, symmetry_worst
- **Fractal Dimension**: fractal_dimension_mean, fractal_dimension_se, fractal_dimension_worst

7. Correlation Analysis

- To further refine the relevant features, a correlation analysis should be performed to identify which features are most strongly correlated with the diagnosis.

8. Principal Component Analysis (PCA)

- PCA can be utilized to reduce the dimensionality of the dataset while retaining the most significant features, which can help in identifying the factors that contribute most to the diagnosis prediction.

5. What limitations are there in the data, and how might they affect our conclusions?

Limitations in the Breast Cancer Wisconsin (Diagnostic) Dataset:

1. **Missing Values**:
   - The dataset contains missing values in some features, specifically radius_mean and area_mean.
   - **Impact**: Imputation methods (such as filling with median) might introduce bias and affect the accuracy of the model. If missing values are not handled appropriately, it can lead to incorrect conclusions.

2. **Class Imbalance**:
   - The dataset has 357 benign cases and 212 malignant cases.
   - **Impact**: Class imbalance can lead to biased models that are more likely to predict the majority class (benign), thereby reducing the sensitivity to the minority class (malignant). This can affect the detection of malignant cases, which is critical in medical diagnostics.

3. **Outliers**:
   - The presence of extreme values in features like area_mean.
   - **Impact**: Outliers can skew the results of statistical analyses and machine learning models, potentially leading to incorrect conclusions or poor model performance. Outliers might represent real but rare cases or errors in data collection.

4. **Feature Scaling**:

- The features have different scales and ranges.
- **Impact**: Algorithms that rely on distance metrics (like KNN, SVM) are sensitive to feature scales. If features are not normalized, those with larger scales can dominate the distance calculation, leading to biased results.

5. **Correlation Among Features**:

   - Some features are highly correlated (e.g., radius_mean, perimeter_mean, and area_mean).
   - **Impact**: Multicollinearity can affect the performance of regression models, making it difficult to determine the effect of each feature independently. It might also lead to overfitting in some models.

6. **Data Source and Collection**:

   - The dataset is from a single source, which might not be representative of the entire population.
   - **Impact**: Models trained on this dataset might not generalize well to other populations or medical centers with different characteristics. This can affect the model's external validity.

7. **Limited Feature Scope**:

   - The dataset focuses on specific characteristics of cell nuclei but does not include other potentially relevant clinical data (e.g., patient age, family history, genetic information).
   - **Impact**: The model might miss out on other important predictors of breast cancer, leading to incomplete conclusions.

8. **Temporal Aspects**:

   - The dataset does not provide information on the temporal sequence of data collection.
   - **Impact**: Without understanding changes over time, it's challenging to draw conclusions about disease progression or the effect of treatment.

By acknowledging and addressing these limitations, we can improve the robustness and generalizability of our conclusions and the performance of our predictive models.