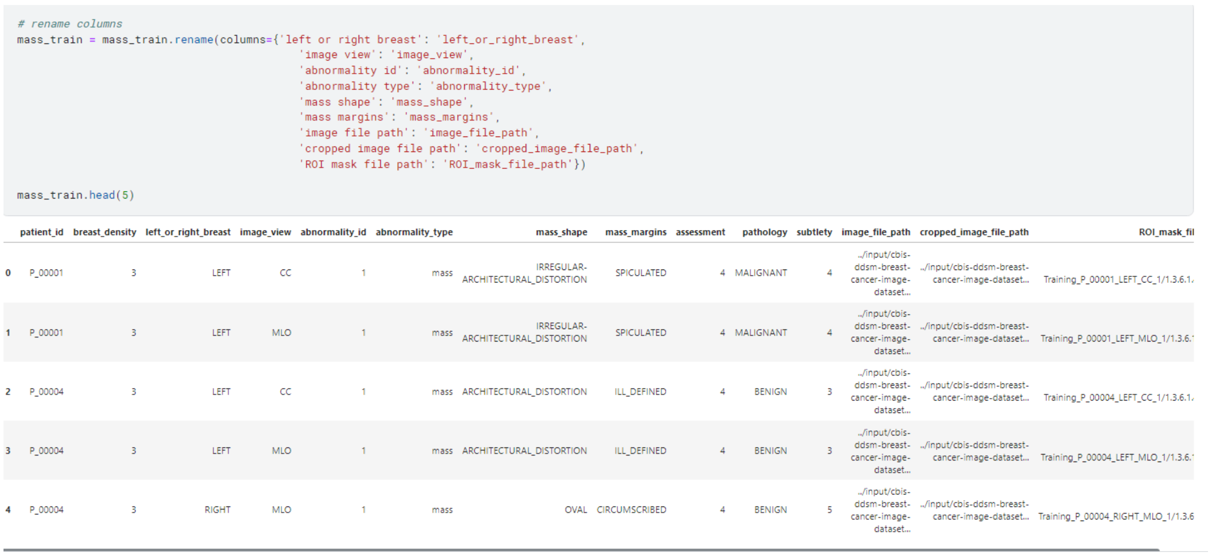
Output results: -

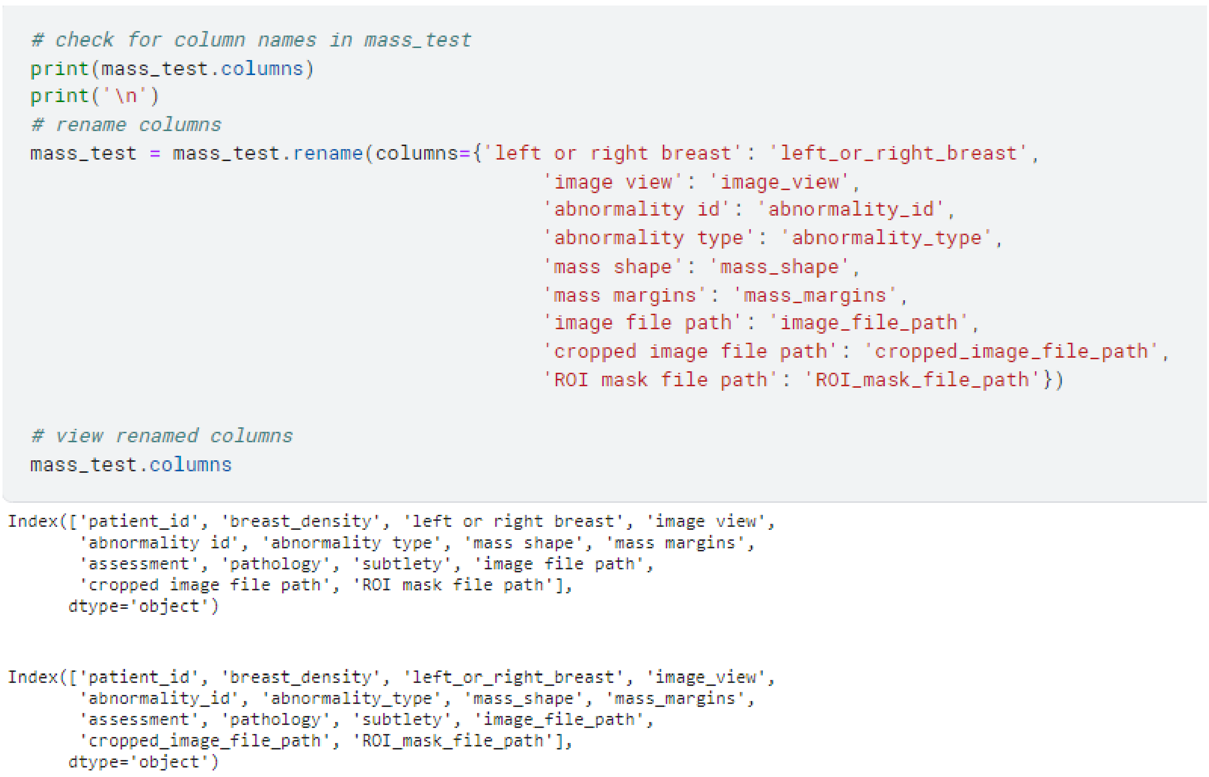
1. Training:-



```
# rename columns
mass_train = mass_train.rename(columns={'left or right breast': 'left_or_right_breast',
                                        'image view': 'image_view',
                                        'abnormality id': 'abnormality_id',
                                        'abnormality type': 'abnormality_type',
                                        'mass shape': 'mass_shape',
                                        'mass margins': 'mass_margins',
                                        'image file path': 'image_file_path',
                                        'cropped image file path': 'cropped_image_file_path',
                                        'ROI mask file path': 'ROI_mask_file_path'})


mass_train.head(5)
```

	patient_id	breast_density	left_or_right_breast	image_view	abnormality_id	abnormality_type	mass_shape	mass_margins	assessment	pathology	subtlety	image_file_path	cropped_image_file_path	ROI_mask_fil
0	P_00001	3	LEFT	CC	1	mass	IRREGULAR-ARCHITECTURAL_DISTORTION	SPICULATED	4	MALIGNANT	4	../input/cbis-ddsm-breast-cancer-image-dataset...	../input/cbis-ddsm-breast-cancer-image-dataset...	Training_P_00001_LEFT_CC_1/1.3.6.1.
1	P_00001	3	LEFT	MLO	1	mass	IRREGULAR-ARCHITECTURAL_DISTORTION	SPICULATED	4	MALIGNANT	4	../input/cbis-ddsm-breast-cancer-image-dataset...	../input/cbis-ddsm-breast-cancer-image-dataset...	Training_P_00001_LEFT_MLO_1/1.3.6.
2	P_00004	3	LEFT	CC	1	mass	ARCHITECTURAL_DISTORTION	ILL_DEFINED	4	BENIGN	3	../input/cbis-ddsm-breast-cancer-image-dataset...	../input/cbis-ddsm-breast-cancer-image-dataset...	Training_P_00004_LEFT_CC_1/1.3.6.1.
3	P_00004	3	LEFT	MLO	1	mass	ARCHITECTURAL_DISTORTION	ILL_DEFINED	4	BENIGN	3	../input/cbis-ddsm-breast-cancer-image-dataset...	../input/cbis-ddsm-breast-cancer-image-dataset...	Training_P_00004_LEFT_MLO_1/1.3.6.
4	P_00004	3	RIGHT	MLO	1	mass	OVAL	CIRCUMSCRIBED	4	BENIGN	5	../input/cbis-ddsm-breast-cancer-image-dataset...	../input/cbis-ddsm-breast-cancer-image-dataset...	Training_P_00004_RIGHT_MLO_1/1.3.6

2.

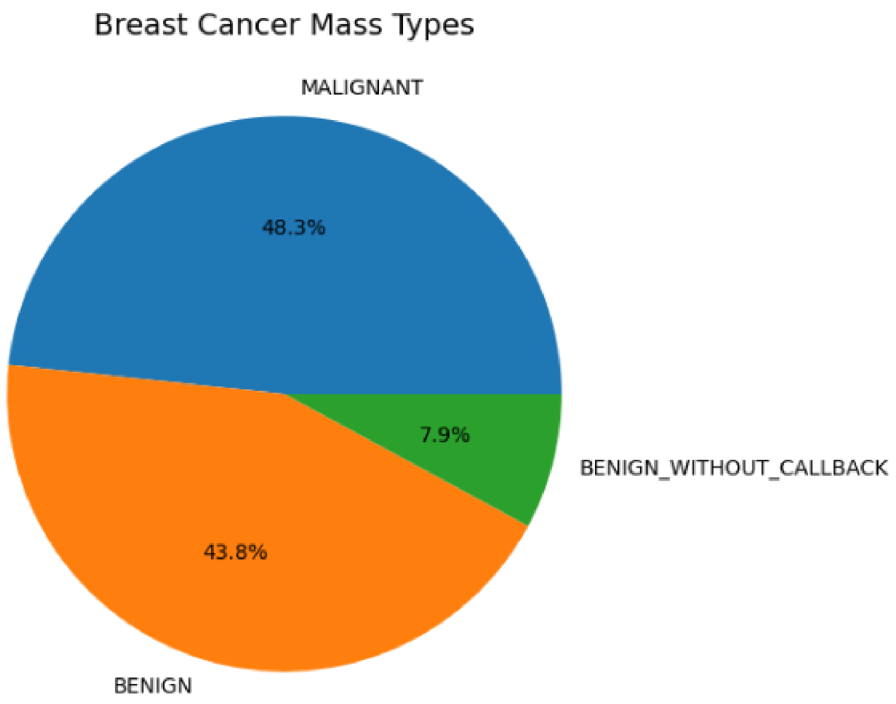


```
# check for column names in mass_test
print(mass_test.columns)
print('\n')
# rename columns
mass_test = mass_test.rename(columns={'left or right breast': 'left_or_right_breast',
                                      'image view': 'image_view',
                                      'abnormality id': 'abnormality_id',
                                      'abnormality type': 'abnormality_type',
                                      'mass shape': 'mass_shape',
                                      'mass margins': 'mass_margins',
                                      'image file path': 'image_file_path',
                                      'cropped image file path': 'cropped_image_file_path',
                                      'ROI mask file path': 'ROI_mask_file_path'})


# view renamed columns
mass_test.columns
```

```
Index(['patient_id', 'breast_density', 'left or right breast', 'image view',
       'abnormality id', 'abnormality type', 'mass shape', 'mass margins',
       'assessment', 'pathology', 'subtlety', 'image file path',
       'cropped image file path', 'ROI mask file path'],
      dtype='object')


Index(['patient_id', 'breast_density', 'left_or_right_breast', 'image_view',
       'abnormality_id', 'abnormality_type', 'mass_shape', 'mass_margins',
       'assessment', 'pathology', 'subtlety', 'image_file_path',
       'cropped_image_file_path', 'ROI_mask_file_path'],
      dtype='object')
```

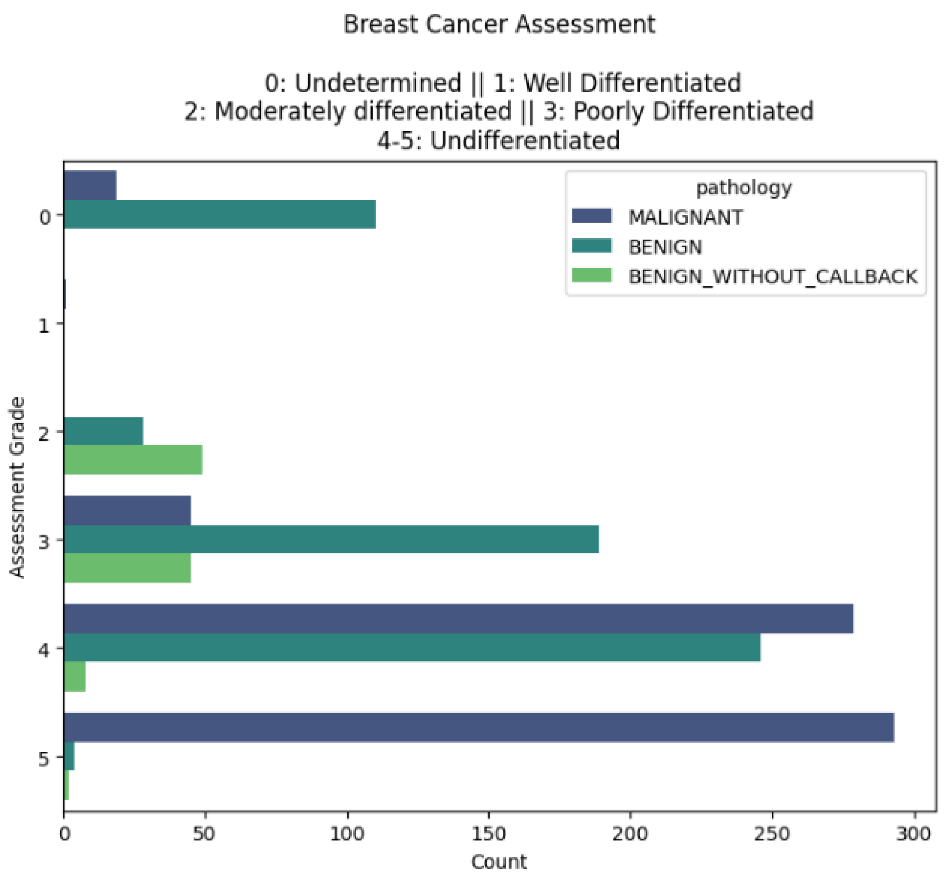
3.

```python
# pathology distributions
value = mass_train['pathology'].value_counts()
plt.figure(figsize=(8,6))

plt.pie(value, labels=value.index, autopct='%1.1f%%')
plt.title('Breast Cancer Mass Types', fontsize=14)
plt.savefig('/kaggle/working/pathology_distributions_red.png')
plt.show()
```



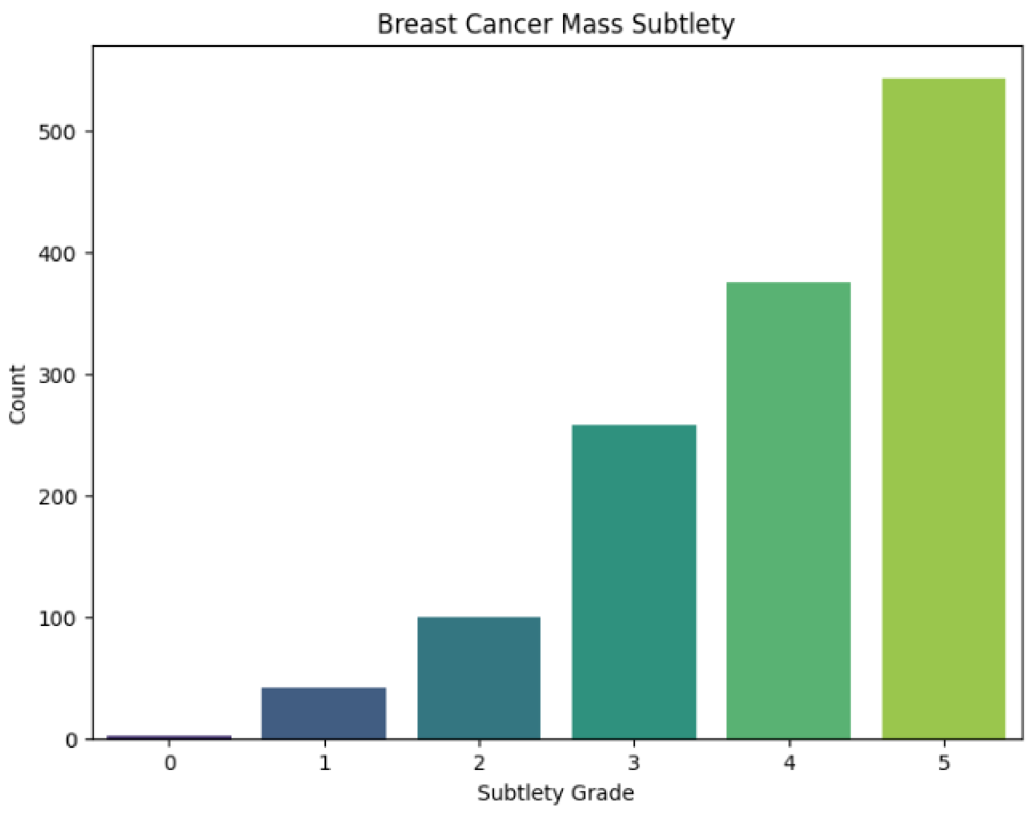
4.

```python
# examine breast assessment types
plt.figure(figsize=(8, 6))

# Utilisez countplot pour afficher la distribution des types d'évaluation en fonction de la pathologie
sns.countplot(data=mass_train, y='assessment', hue='pathology', palette='viridis')

# Ajoutez un titre au graphique
plt.title('Breast Cancer Assessment\n\n 0: Undetermined || 1: Well Differentiated\n2: Moderately differentiated || 3: Poorly Differentiated\n4-5: Undifferentiated', fontsize=12)

# Ajoutez des étiquettes aux axes
plt.ylabel('Assessment Grade')
plt.xlabel('Count')

# Sauvegardez le graphique en tant qu'image (optionnel)
plt.savefig('/kaggle/working/breast_assessment_red.png')

# Affichez le graphique
plt.show()
```


Breast Cancer Assessment
0: Undetermined || 1: Well Differentiated
2: Moderately differentiated || 3: Poorly Differentiated
4-5: Undifferentiated
pathology
MALIGNANT
BENIGN
BENIGN_WITHOUT_CALLBACK
Assessment Grade
0
1
2
3
4
5
0
50
100
150
200
250
300
Count

5.

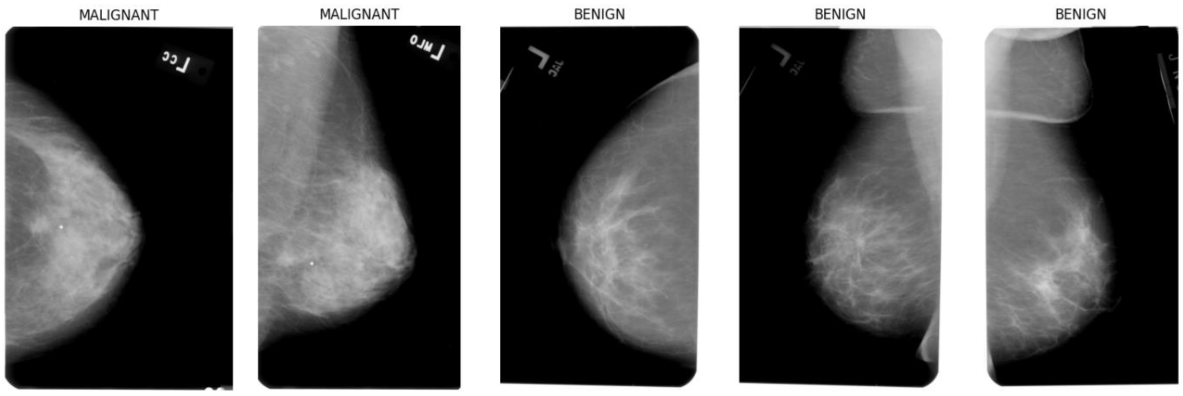
```
# examine cancer subtlety
plt.figure(figsize=(8,6))
sns.countplot(mass_train, x='subtlety', palette='viridis')
plt.title('Breast Cancer Mass Subtlety', fontsize=12)
plt.xlabel('Subtlety Grade')
plt.ylabel('Count')
plt.savefig('/kaggle/working/cancer_subtlety_red.png')
plt.show()
```



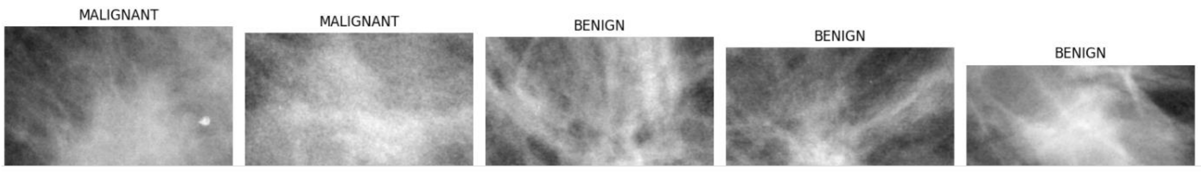
6.

```python
# Display some images
import matplotlib.image as mpimg

# create function to display images
def display_images(column, number):
    """displays images in dataset"""
    # create figure and axes
    number_to_visualize = number
    rows = 1
    cols = number_to_visualize
    fig, axes = plt.subplots(rows, cols, figsize=(15, 5))

    # Loop through rows and display images
    for index, row in mass_train.head(number_to_visualize).iterrows():
        image_path = row[column]
        image = mpimg.imread(image_path)
        ax = axes[index]
        ax.imshow(image, cmap='gray')
        ax.set_title(f"{row['pathology']}")
        ax.axis('off')
    plt.tight_layout()
    plt.show()

print('Full Mammograms:\n')
display_images('image_file_path', 5)
print('Cropped Mammograms:\n')
display_images('cropped_image_file_path', 5)
```

```
Full Mammograms:
```



```
Cropped Mammograms:
```



```
from keras.preprocessing import image

plt.figure(figsize = (15, 15))

some_non = np.random.randint(0, len(non_can_img), 18)
some_can = np.random.randint(0, len(can_img), 18)

s = 0
for num in some_non:

        img = image.load_img((non_can_img[num]), target_size=(100, 100))
        img = image.img_to_array(img)

        plt.subplot(6, 6, 2*s+1)
        plt.axis('off')
        plt.title('no cancer')
        plt.imshow(img.astype('uint8'))
        s += 1

s = 1
for num in some_can:

        img = image.load_img((can_img[num]), target_size=(100, 100))
        img = image.img_to_array(img)
        plt.subplot(6, 6, 2*s)
        plt.axis('off')
        plt.title('cancer')
        plt.imshow(img.astype('uint8'))
        s += 1
```