# DSC530_Final_KristaKnuckey

June 1, 2024

[72]:

```
#Krista Knuckey
#Week 12- Final Project
```

[73]:

```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
from empiricaldist import Pmf
from empiricaldist import Cdf
from scipy.stats import pearsonr
from scipy.stats import ttest_ind
from scipy.stats import norm
import statsmodels.api as sm
```

[74]:

```
#Upload Breast Cancer Wisconsin Data Set

df = pd.read_csv('breast_cancer_data.csv')
df.head()
```

[74]:

```
         id diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
0    842302         M        17.99         10.38          122.80     1001.0
1    842517         M        20.57         17.77          132.90     1326.0
2  84300903         M        19.69         21.25          130.00     1203.0
3  84348301         M        11.42         20.38           77.58      386.1
4  84358402         M        20.29         14.34          135.10     1297.0

   smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
0          0.11840           0.27760          0.3001              0.14710
1          0.08474           0.07864          0.0869              0.07017
2          0.10960           0.15990          0.1974              0.12790
3          0.14250           0.28390          0.2414              0.10520
4          0.10030           0.13280          0.1980              0.10430

   …  texture_worst  perimeter_worst  area_worst  smoothness_worst  \
0  …          17.33           184.60      2019.0            0.1622
1  …          23.41           158.80      1956.0            0.1238
2  …          25.53           152.50      1709.0            0.1444
3  …          26.50            98.87       567.7            0.2098
```

```
4  …         16.67          152.20      1575.0            0.1374

   compactness_worst  concavity_worst  concave points_worst  symmetry_worst  \
0             0.6656           0.7119                0.2654          0.4601
1             0.1866           0.2416                0.1860          0.2750
2             0.4245           0.4504                0.2430          0.3613
3             0.8663           0.6869                0.2575          0.6638
4             0.2050           0.4000                0.1625          0.2364

   fractal_dimension_worst  Unnamed: 32
0                  0.11890          NaN
1                  0.08902          NaN
2                  0.08758          NaN
3                  0.17300          NaN
4                  0.07678          NaN

[5 rows x 33 columns]
```

[75]:

```
#Explain 5 variables in dataset

#radius_mean - the mean distance from center to the perimeter
#texture_mean - standard deviation of the gray-scale values
#perimeter_mean - mean of tumor
#area_mean - mean area of the tumor
#smoothness_mean - local variation in radius lengths
```

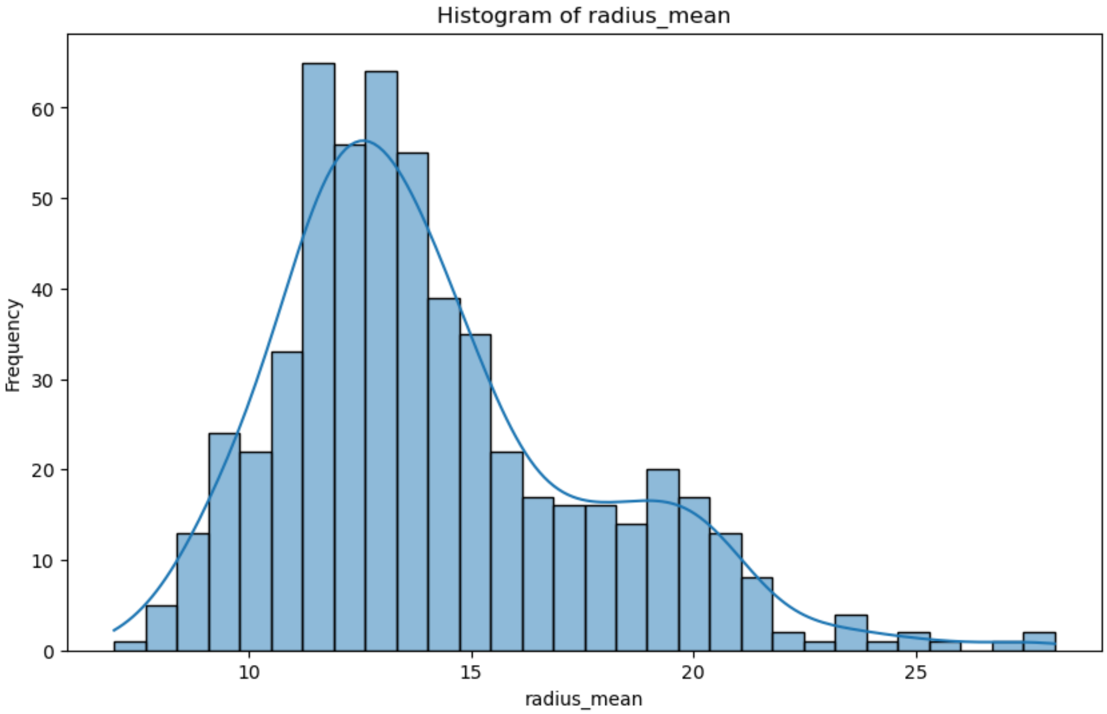
[76]:

```
#Plot histograms for 5 variables
```

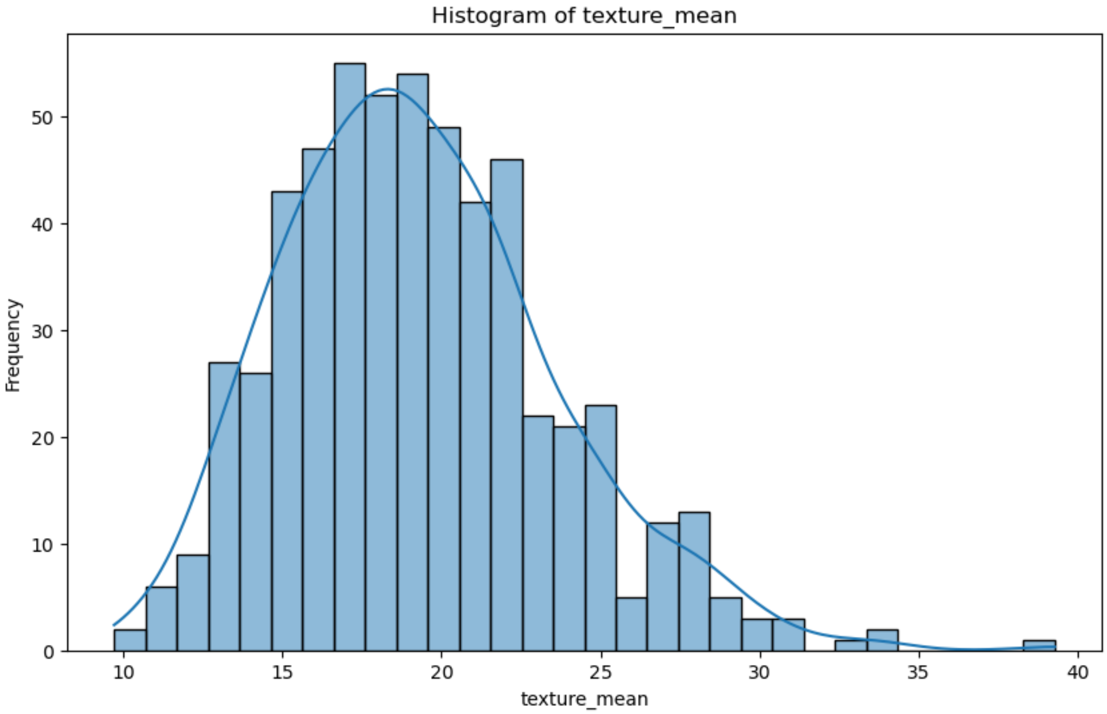
[77]:

```
variables = ['radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean',␣
 ↪'smoothness_mean']

for var in variables:
    plt.figure(figsize=(10, 6))
    sns.histplot(df[var], kde=True, bins=30)
    plt.title(f'Histogram of {var}')
    plt.xlabel(var)
    plt.ylabel('Frequency')
    plt.show()
```

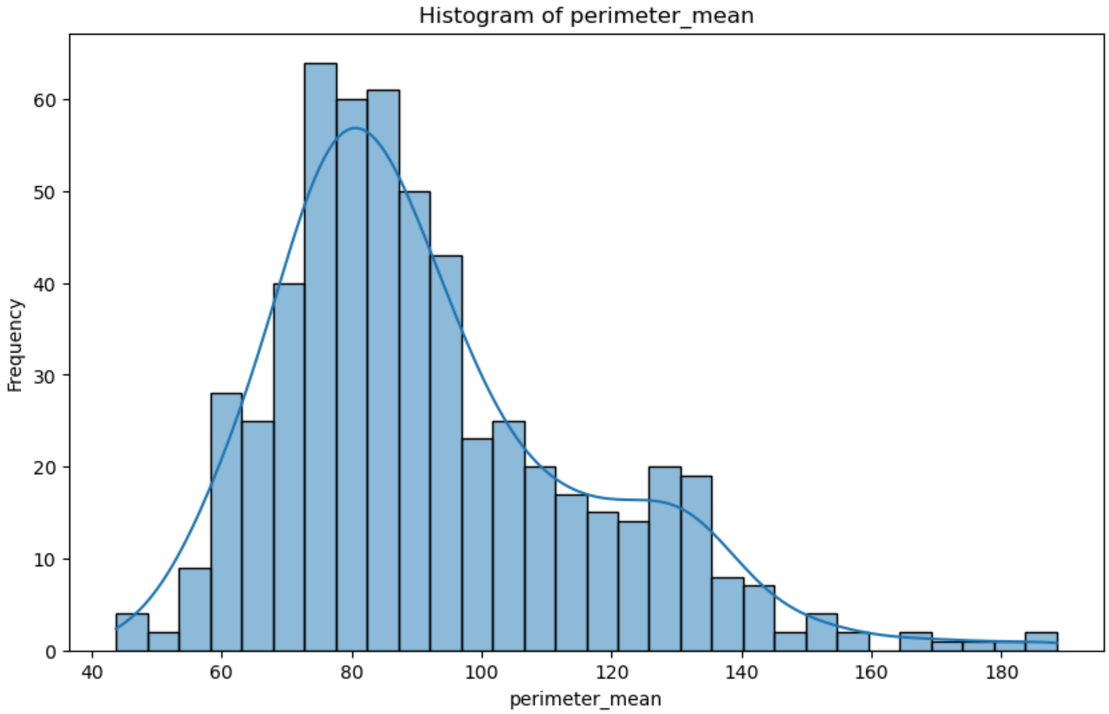
```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



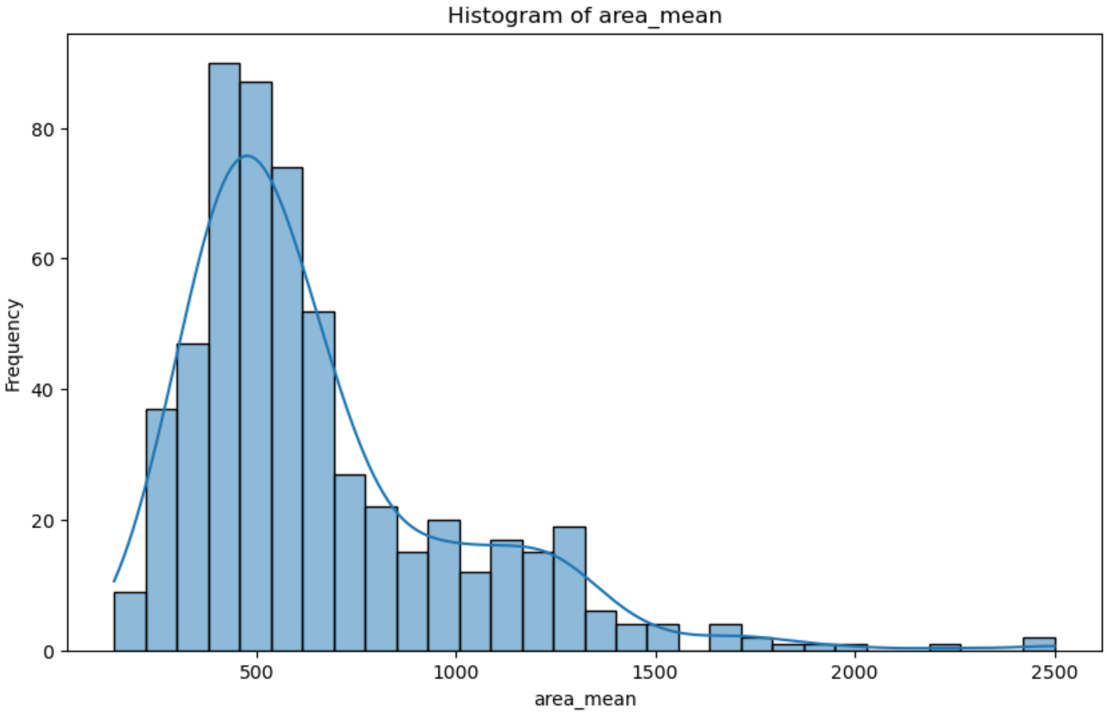
```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



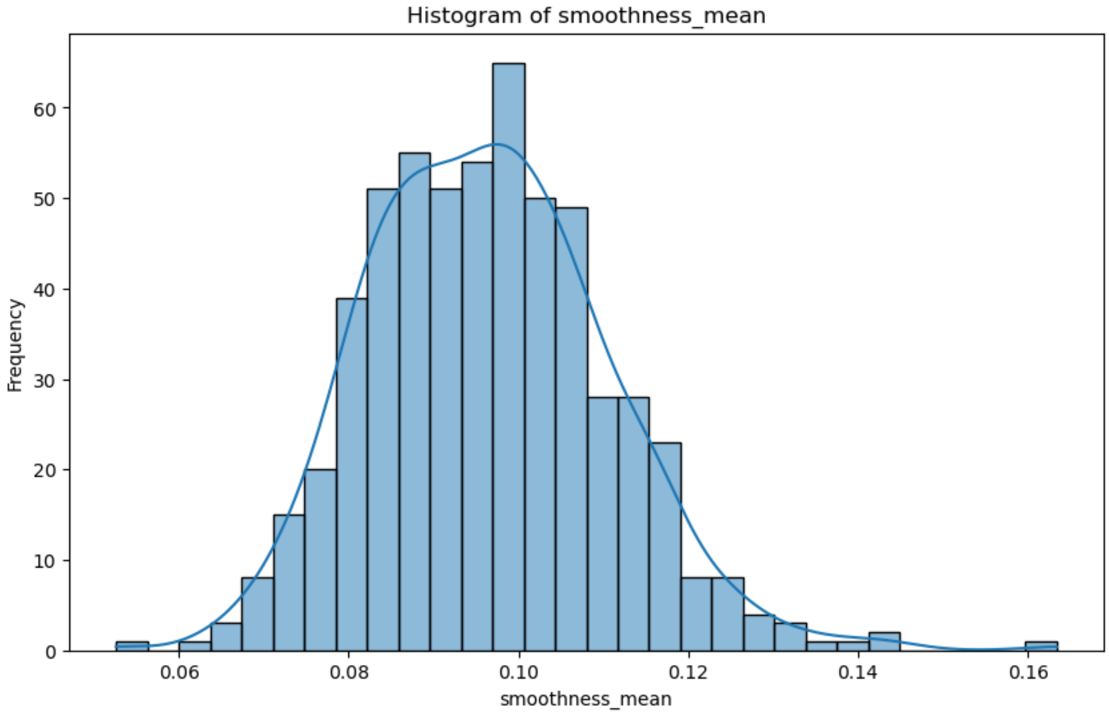
```
/opt/anaconda3/lib/python3.11/site-packages/seaborn/_oldcore.py:1119:
FutureWarning: use_inf_as_na option is deprecated and will be removed in a
future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



```
/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 operating 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 operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



[78]:

```
#Results-
#The radius mean histogram is positively skewed, meaning that many tumors are␣
 ↪smaller in size. The outliers in this histogram are that there are
#some tumors that are larger than normal.

#The texture mean histogram is slightly positvely skewed, but is close to a␣
 ↪normal distribution.
#This means that most tumors have similar texture, but the outliers would have␣
 ↪different textures than the mean

#The perimeter mean historgram is very similarly shaped to the radius mean␣
 ↪histogram, which makes sense as it the perimeter is the entire tumor size.
#The outliers here would also mean that the there are some tumors that are much␣
 ↪larger than normal.

#The area mean histogram is also positively skewed, like the radius and␣
 ↪perimeter as they all have different, but similar measurements when it
#involves a tumor.

#The smoothness mean histogram is very close to a normal distribution meaning␣
 ↪that the smoothness is evenly distributed amoungst all tumors.
```

[79]:

```
#descriptive characteristics about variables
```

[80]:

```
for var in variables:
    mean = df[var].mean()
    mode = df[var].mode()[0]
    std_dev = df[var].std()
    min_val = df[var].min()
    max_val = df[var].max()

    print(f'\nDescriptive statistics for {var}:')
    print(f'Mean: {mean}')
    print(f'Mode: {mode}')
    print(f'Standard Deviation: {std_dev}')
    print(f'Min: {min_val}')
    print(f'Max: {max_val}')
```

```
Descriptive statistics for radius_mean:
Mean: 14.127291739894552
Mode: 12.34
Standard Deviation: 3.524048826212078
Min: 6.981
Max: 28.11

Descriptive statistics for texture_mean:
Mean: 19.289648506151142
Mode: 14.93
Standard Deviation: 4.301035768166949
Min: 9.71
Max: 39.28

Descriptive statistics for perimeter_mean:
Mean: 91.96903339191564
Mode: 82.61
Standard Deviation: 24.2989810387549
Min: 43.79
Max: 188.5

Descriptive statistics for area_mean:
Mean: 654.8891036906855
Mode: 512.2
Standard Deviation: 351.9141291816527
Min: 143.5
Max: 2501.0

Descriptive statistics for smoothness_mean:
Mean: 0.0963602811950791
Mode: 0.1007
Standard Deviation: 0.014064128137673616
Min: 0.05263
```

```
Max: 0.1634
```

[81]:

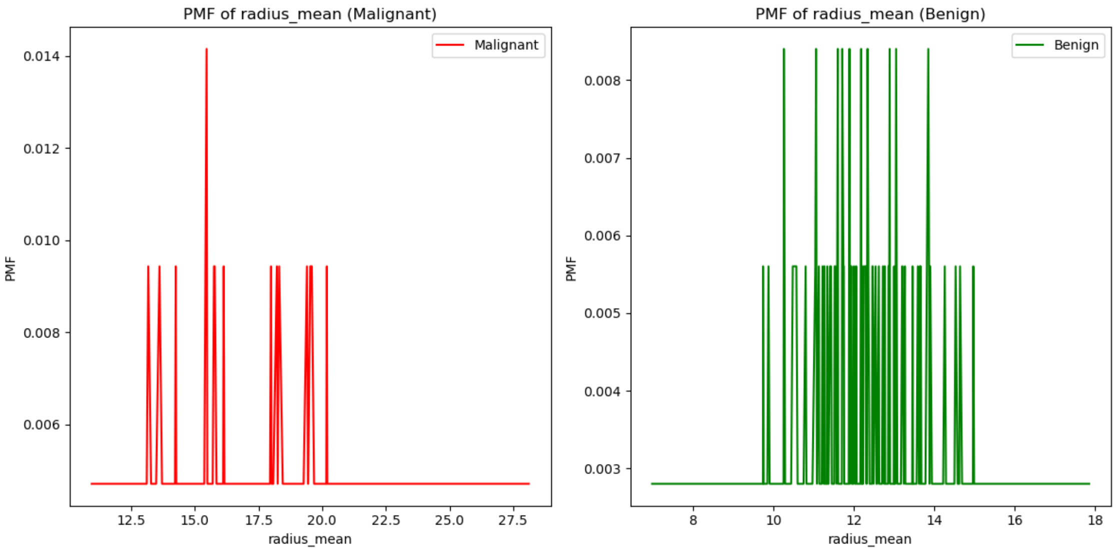
```
#Results-
#The descriptive characteristics about the variables give us great information␣
 ↪about the data for the tumor. We see that the average radius mean
#is 14.13, but the outliers can be smaller or larger. Also, the perimeter of␣
 ↪the tumor has a mean of 91.97, meaning that there is a high variability.
#The smoothness of tumors has the lowest variability meaning that that the␣
 ↪tumors are all very similar in value for smoothness.
```

[82]:

```
#PMF
```

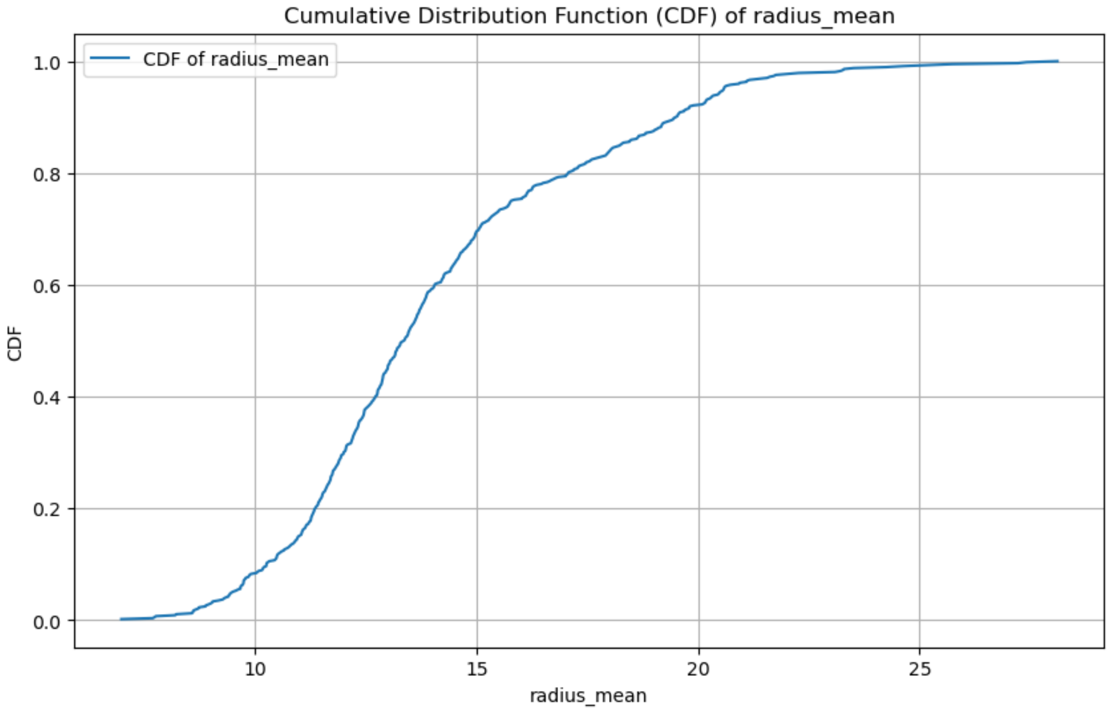
[83]:

```
malignant = df[df['diagnosis'] == 'M']
benign = df[df['diagnosis'] == 'B']

pmf_malignant = Pmf.from_seq(malignant['radius_mean'])
pmf_benign = Pmf.from_seq(benign['radius_mean'])

plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
pmf_malignant.plot(label='Malignant', color='red')
plt.xlabel('radius_mean')
plt.ylabel('PMF')
plt.title('PMF of radius_mean (Malignant)')
plt.legend()

plt.subplot(1, 2, 2)
pmf_benign.plot(label='Benign', color='green')
plt.xlabel('radius_mean')
plt.ylabel('PMF')
plt.title('PMF of radius_mean (Benign)')
plt.legend()

plt.tight_layout()
plt.show()
```



[84]:
```
#Results-
#For the Probability Mass Function (PMF), I created visualizations of the␣
 ↪radius mean for both malignant and benign tumors. With this visualization
#we can see that malignant tumors typically have a larger radius and there is a␣
 ↪wider gap in the PMF, which means that there is more variability in the size.
#The benign tumors have less variability with the PMF and tend to have a␣
 ↪smaller radius. The malignant visualization displays a longer right tail,␣
 ↪whcih could mean
#that they can experience larger tumors (like our previous outliers suggested).
```

[85]:
```
#CDF

cdf_radius_mean = Cdf.from_seq(df['radius_mean'])

plt.figure(figsize=(10, 6))
cdf_radius_mean.plot(label='CDF of radius_mean')
plt.xlabel('radius_mean')
plt.ylabel('CDF')
plt.title('Cumulative Distribution Function (CDF) of radius_mean')
plt.legend()
plt.grid(True)
plt.show()
```

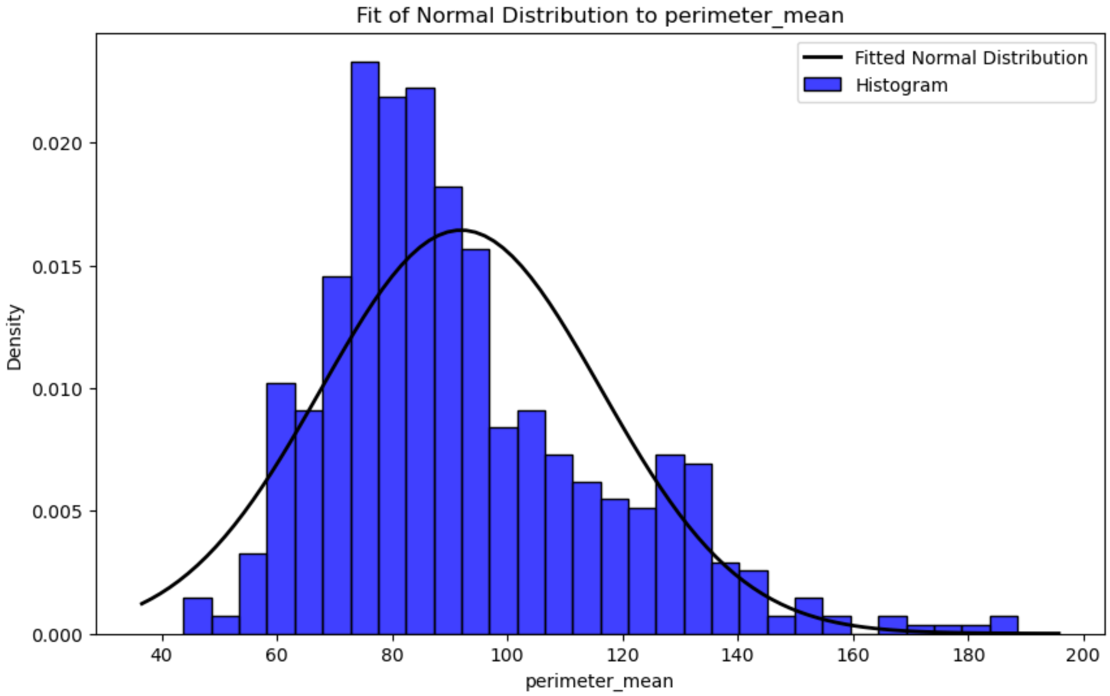


[86]:
```
#Results-
#The Cumulative Distribution Function (CDF) of the radius mean shows a steep incline of the data- especially between 10-15 on x- axis (radius mean).
#This also follows our previous point that the average radius mean is around ~14. The curve flattens out, which means there are fewer tumors
#larger than our mean.
```

[87]:
```
#Analytical Distribution
```

[88]:
```python
data = df['perimeter_mean']

mu, std = norm.fit(data)

plt.figure(figsize=(10, 6))
sns.histplot(data, kde=False, bins=30, color='blue', stat='density', label='Histogram')

xmin, xmax = plt.xlim()
x = np.linspace(xmin, xmax, 100)
p = norm.pdf(x, mu, std)
plt.plot(x, p, 'k', linewidth=2, label='Fitted Normal Distribution')

plt.title('Fit of Normal Distribution to perimeter_mean')
plt.xlabel('perimeter_mean')
plt.ylabel('Density')
plt.legend()
plt.show()

print(f"Fitted normal distribution parameters: mu = {mu:.2f}, std = {std:.2f}")
```

```
/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 operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```



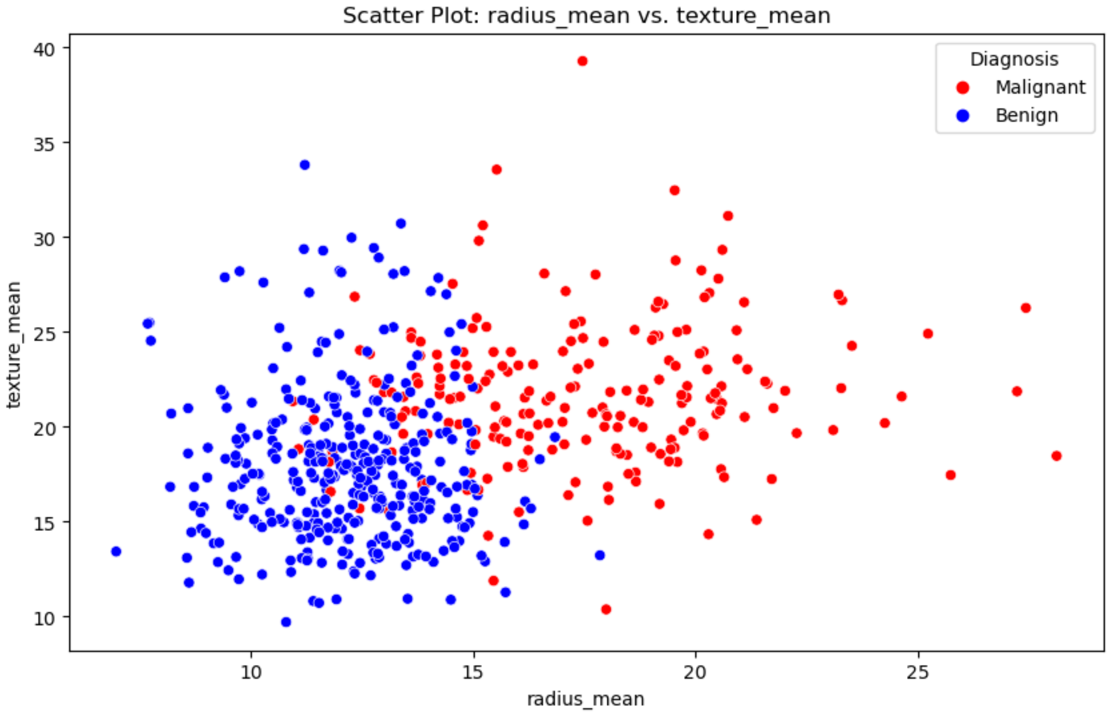
```
Fitted normal distribution parameters: mu = 91.97, std = 24.28
```

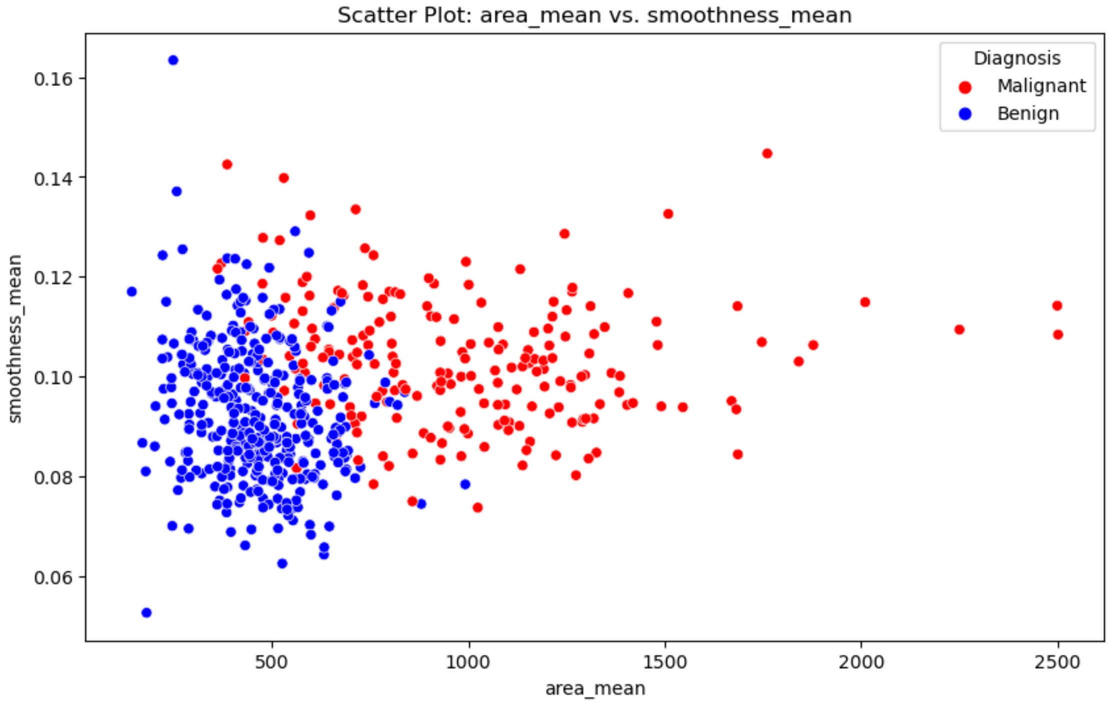
[89]:
```
#Results-
#I plotted a normal distribution of the perimeter_mean data as the perimeter of␣
↪tumor is important to investigate in terms of being malignant
#or benign. Here we can see there is a list skewness in the data, but it does␣
↪lean into the normal distribution. The peak of the normal distribution
#is slightly off from the histogram, but this could also be due to the␣
↪variability in the perimeter data.
```

[90]:
```
#Two Scatter Plots
```

[91]:
```
plt.figure(figsize=(10, 6))
```

```python
scatter1 = sns.scatterplot(x='radius_mean', y='texture_mean', data=df,␣
 ↪hue='diagnosis', palette={'M': 'red', 'B': 'blue'})
plt.title('Scatter Plot: radius_mean vs. texture_mean')
plt.xlabel('radius_mean')
plt.ylabel('texture_mean')

handles, labels = scatter1.get_legend_handles_labels()
scatter1.legend(handles=handles, labels=['Malignant', 'Benign'],␣
 ↪title='Diagnosis')

plt.show()

plt.figure(figsize=(10, 6))
scatter2 = sns.scatterplot(x='area_mean', y='smoothness_mean', data=df,␣
 ↪hue='diagnosis', palette={'M': 'red', 'B': 'blue'})
plt.title('Scatter Plot: area_mean vs. smoothness_mean')
plt.xlabel('area_mean')
plt.ylabel('smoothness_mean')

handles, labels = scatter2.get_legend_handles_labels()
scatter2.legend(handles=handles, labels=['Malignant', 'Benign'],␣
 ↪title='Diagnosis')

plt.show()

def calculate_statistics(x, y):
    covariance = np.cov(x, y)[0, 1]
    pearson_corr, _ = pearsonr(x, y)
    return covariance, pearson_corr

cov1, corr1 = calculate_statistics(df['radius_mean'], df['texture_mean'])
print(f"radius_mean vs. texture_mean:\nCovariance: {cov1:.2f}\nPearson's␣
 ↪Correlation: {corr1:.2f}\n")

cov2, corr2 = calculate_statistics(df['area_mean'], df['smoothness_mean'])
print(f"area_mean vs. smoothness_mean:\nCovariance: {cov2:.2f}\nPearson's␣
 ↪Correlation: {corr2:.2f}\n")
```





```
radius_mean vs. texture_mean:
```

```
Covariance: 4.91
Pearson's Correlation: 0.32

area_mean vs. smoothness_mean:
Covariance: 0.88
Pearson's Correlation: 0.18
```

[92]:

```
#Results-
#The first scatter plot of radius vs texture has a covariance of 4.91 and␣
 ↪Pearson's Correlation of .32. The positive covariance score demonstrates
#that the two variables tend to increase together, but the .32 Pearson's␣
 ↪Correlation demonstrates a weak linear relationship between the two points.

#The second scatter plot of area vs smoothness has a positive covariance as␣
 ↪well, meaning that both points will increase together, but have a very low
#Pearson's Correlation of .18, meaning there is a very weak linear relationship.
 ↪

#It is very interesting to visualy see the benign and malignant plots and how␣
 ↪they tend to not overlap at all in both scatterplots.
#I did run into an issue with the legends of these scatterplots, which is why␣
 ↪the code is separated. This is to make sure the data points were
#accuratley recorded by color red and blue.
```

[93]:

```
#Hypothesis test
```

[94]:

```
malignant = df[df['diagnosis'] == 'M']['radius_mean']
benign = df[df['diagnosis'] == 'B']['radius_mean']

t_stat, p_value = ttest_ind(malignant, benign)

print(f"t-statistic: {t_stat:.2f}")
print(f"p-value: {p_value:.4f}")
```

```
t-statistic: 25.44
p-value: 0.0000
```

[95]:

```
#Results-
#We can see that we have a high t-statistic of 25.44 and low p-value of 0.00␣
 ↪when running a hypothesis test of the radius mean of malignant and benign␣
 ↪tumors
#This result suggests that there is a very significant difference between the␣
 ↪radius mean of a malignant and benign tumor.
```

[96]:

```
#Regression Analysis
```

[97]:

```python
X = df[['texture_mean', 'area_mean', 'smoothness_mean']]
y = df['radius_mean']

X = sm.add_constant(X)

model = sm.OLS(y, X).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:            radius_mean   R-squared:                       0.975
Model:                            OLS   Adj. R-squared:                  0.975
Method:                 Least Squares   F-statistic:                     7327.
Date:                Sat, 01 Jun 2024   Prob (F-statistic):               0.00
Time:                        19:37:40   Log-Likelihood:                -474.81
No. Observations:                 569   AIC:                             957.6
Df Residuals:                     565   BIC:                             975.0
Df Model:                           3
Covariance Type:            nonrobust
================================================================================
===
                      coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
---
const               7.6398      0.197     38.796      0.000       7.253
8.027
texture_mean        0.0059      0.006      1.020      0.308      -0.005
0.017
area_mean           0.0099   7.18e-05    137.478      0.000       0.010
0.010
smoothness_mean    -0.9394      1.702     -0.552      0.581      -4.282
2.403
==============================================================================
Omnibus:                      406.010   Durbin-Watson:                   1.905
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             7631.703
Skew:                          -2.912   Prob(JB):                         0.00
Kurtosis:                      19.970   Cond. No.                     5.42e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
[2] The condition number is large, 5.42e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

```
[ ]: #Results-

     #I ran the regression analysis with radius mean as the depdent variable and␣
      ↪texture, area, and smoothness as the explantory variables.
     #The R-squared of .975 demonstrates a good fit with the model. Also, the t␣
      ↪value of area mean, 137.478, demonstrates a strong relationship
     #with area and radius. The smoothness and texture do not have a significant␣
      ↪relationship with radius.
```