In [104…
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import math
```

In [26]:
```
dataset = pd.read_csv("Boston.csv")
```

In [27]:
```
dataset.head()
```

Out[27]:

	Unnamed: 0	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
0	1	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
1	2	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
2	3	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
3	4	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
4	5	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2

In [28]:
```
type(dataset)
dataset.pop("Unnamed: 0")
dataset
```

Out[28]:

	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
0	0.00632	18.0	2.31	0	0.538	6.575	65.2	4.0900	1	296	15.3	396.90	4.98	24.0
1	0.02731	0.0	7.07	0	0.469	6.421	78.9	4.9671	2	242	17.8	396.90	9.14	21.6
2	0.02729	0.0	7.07	0	0.469	7.185	61.1	4.9671	2	242	17.8	392.83	4.03	34.7
3	0.03237	0.0	2.18	0	0.458	6.998	45.8	6.0622	3	222	18.7	394.63	2.94	33.4
4	0.06905	0.0	2.18	0	0.458	7.147	54.2	6.0622	3	222	18.7	396.90	5.33	36.2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
501	0.06263	0.0	11.93	0	0.573	6.593	69.1	2.4786	1	273	21.0	391.99	9.67	22.4
502	0.04527	0.0	11.93	0	0.573	6.120	76.7	2.2875	1	273	21.0	396.90	9.08	20.6
503	0.06076	0.0	11.93	0	0.573	6.976	91.0	2.1675	1	273	21.0	396.90	5.64	23.9
504	0.10959	0.0	11.93	0	0.573	6.794	89.3	2.3889	1	273	21.0	393.45	6.48	22.0
505	0.04741	0.0	11.93	0	0.573	6.030	80.8	2.5050	1	273	21.0	396.90	7.88	11.9

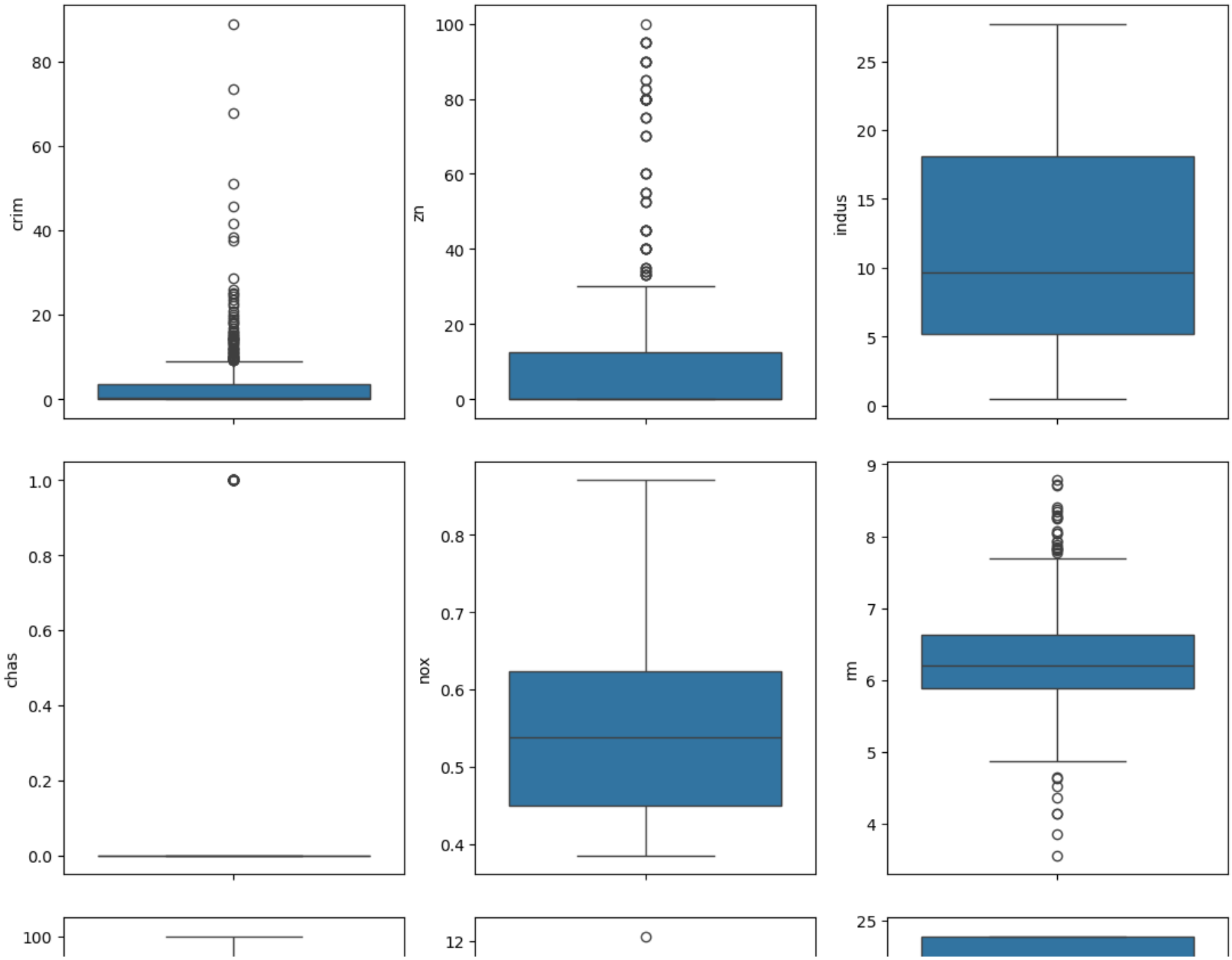
506 rows × 14 columns

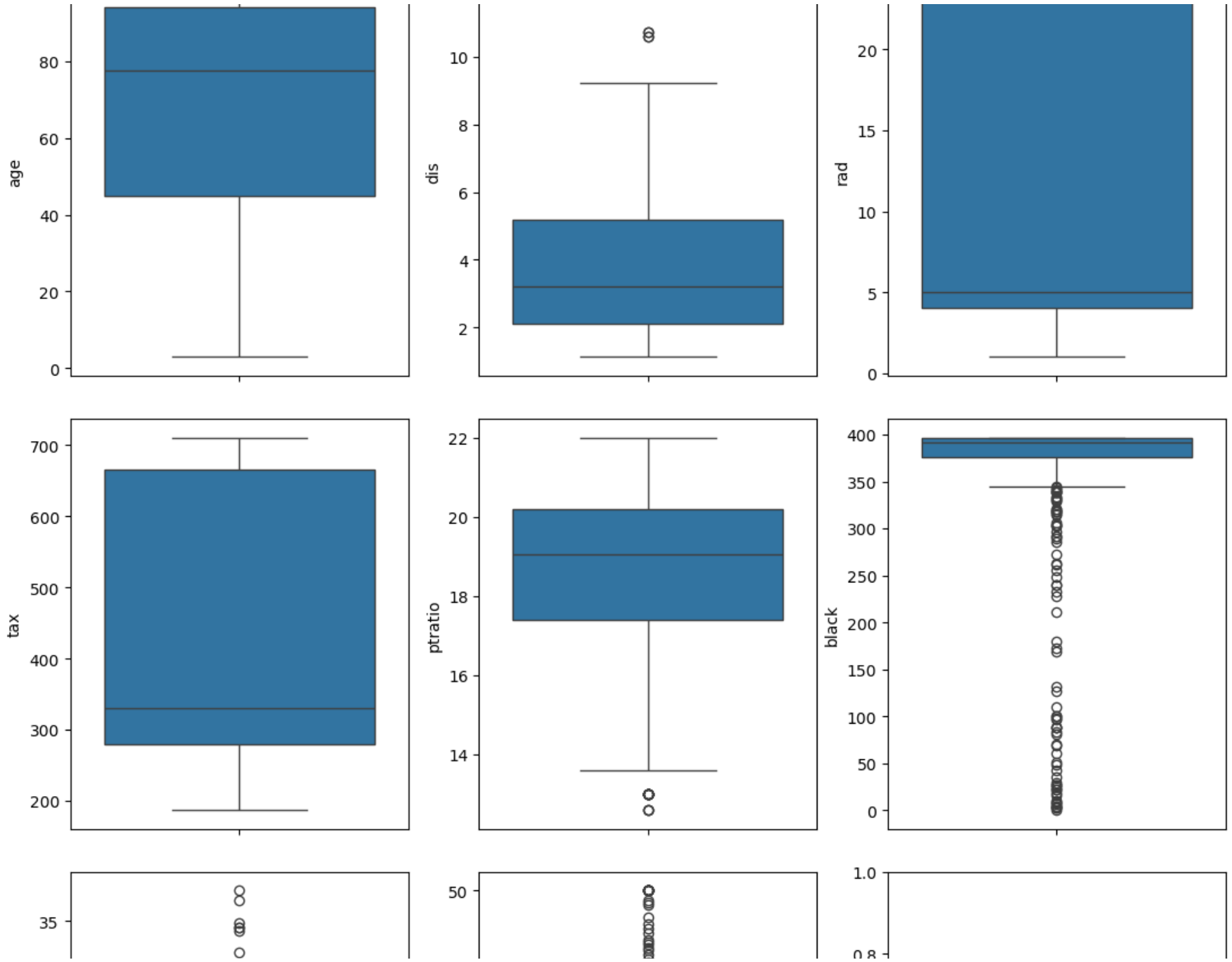
In [29]:
```
# Descriptive statistics for each column
dataset.describe()
```

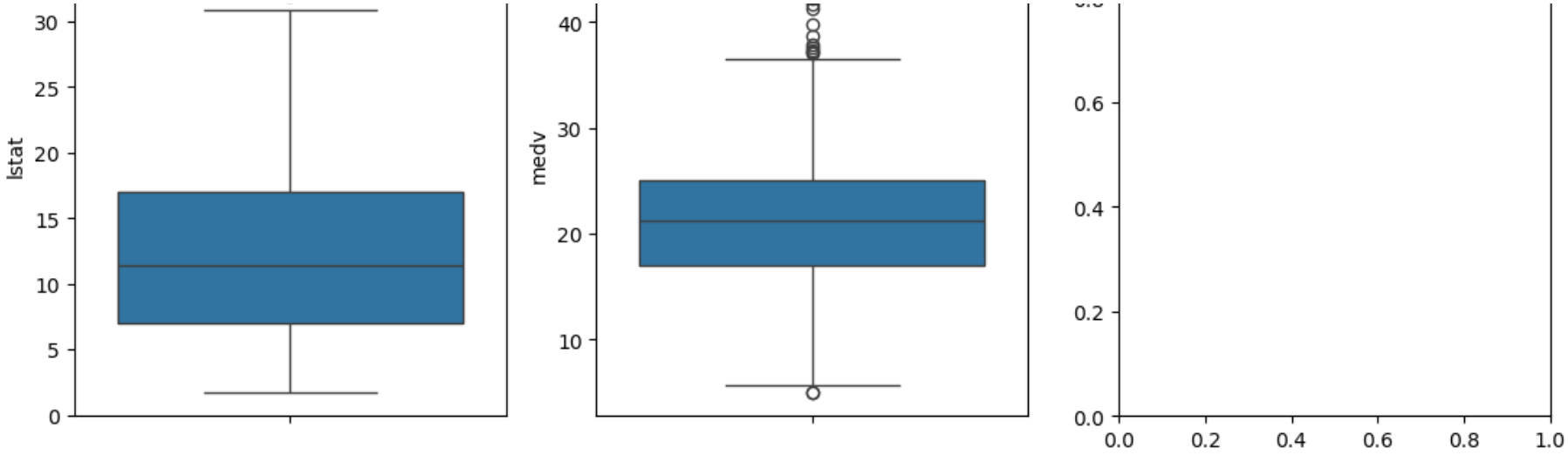
Out[29]:

	crim	zn	indus	chas	nox	rm	age	dis	rad	
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506
mean	3.613524	11.363636	11.136779	0.069170	0.554695	6.284634	68.574901	3.795043	9.549407	408
std	8.601545	23.322453	6.860353	0.253994	0.115878	0.702617	28.148861	2.105710	8.707259	168
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	2.900000	1.129600	1.000000	187
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500	45.025000	2.100175	4.000000	279
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500	77.500000	3.207450	5.000000	330
75%	3.677083	12.500000	18.100000	0.000000	0.624000	6.623500	94.075000	5.188425	24.000000	666
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	100.000000	12.126500	24.000000	711

In [63]:
```
# outliers for each column
fig, axs = plt.subplots(ncols=3, nrows=5, figsize=(11,20))
index = 0
axs = axs.flatten()
for k,v in dataset.items():
    sns.boxplot(y=k, data=dataset, ax=axs[index])
    index += 1
plt.tight_layout(pad=0.4, w_pad=0.5, h_pad=2.0)
```


crim
zn
indus
chas
nox
rm


age
dis
rad
tax
ptratio
black



In [102…

```
# outlier analysis
#Percent of autliers in each column

outlier_df_col =[]
outlier_df_val =[]
for k, v in dataset.items():
    outlier_df_col.append(k)

    col_summary = v.describe()
    iqr = col_summary["75%"] - col_summary["25%"]
    lb = col_summary["25%"]-(1.5*iqr)
    if lb < col_summary["min"]:
        lb=col_summary["min"]
    ub = col_summary["75%"]+(1.5*iqr)
    if ub> col_summary["max"]:
        ub=col_summary["max"]
    outliers  = len(v[(v<lb)|(v>ub)])
    outlier_df_val.append(round((outliers/col_summary["count"])*100,2))
    '''
    print(f"Outliers analysis for {k}:")
    print(f"Total rows: {col_summary["count"]}")
    print(f"Total outliers:{outliers}")
    print("Outlier Percent: {percent:.2f}%".format(percent=(outliers/col_summary["count"])*100))
    print()
```

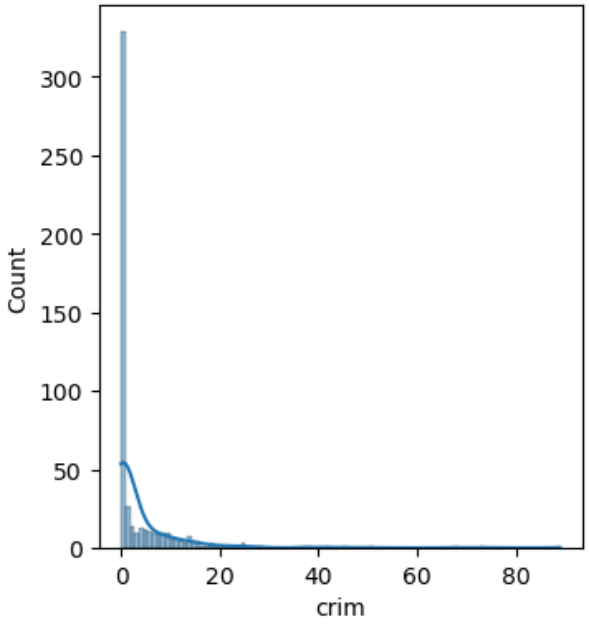
```
    '''
outlier_df = pd.DataFrame(data=np.array([outlier_df_val]), columns = outlier_df_col)
outlier_df
```

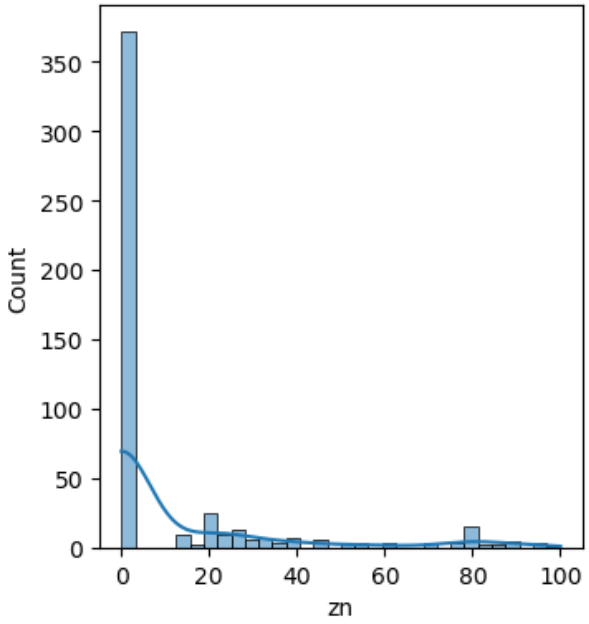
Out[102…

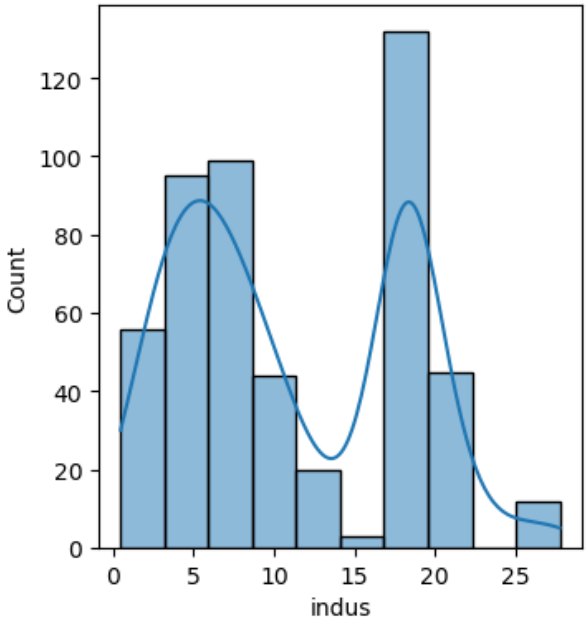
	crim	zn	indus	chas	nox	rm	age	dis	rad	tax	ptratio	black	lstat	medv
0	13.04	13.44	0.0	6.92	0.0	5.93	0.0	0.99	0.0	0.0	2.96	15.22	1.38	7.91

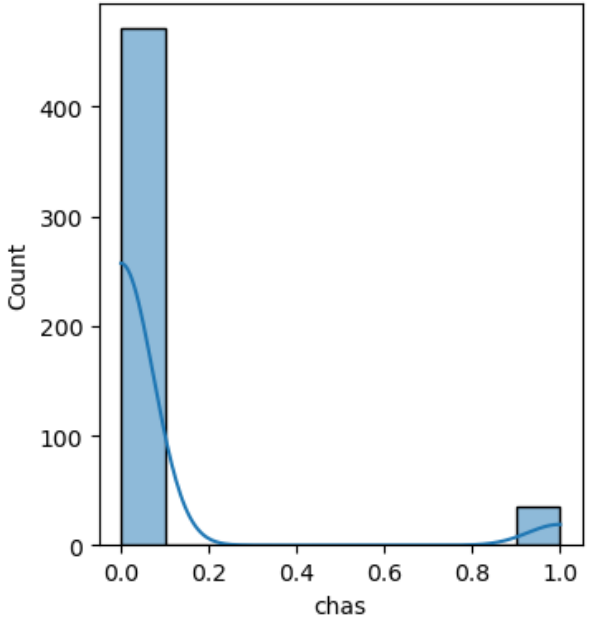
In [78]:

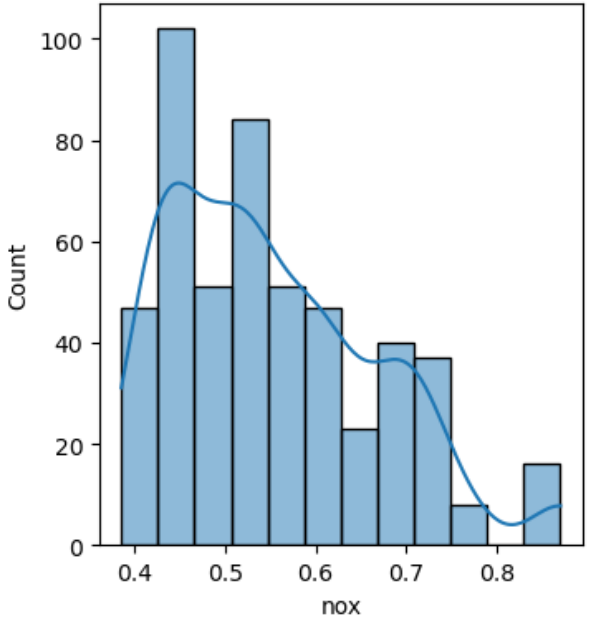
```
# skewness plots for each column
fig, axs= plt.subplots(ncols=3, nrows=5, figsize=(11,20))
index =0
axs=axs.flatten()
for k, v in dataset.items():
    sns.histplot(data=v, ax=axs[index],kde=True)
    index+=1
plt.tight_layout(pad=0.4, w_pad=0.4, h_pad=2.0)
```

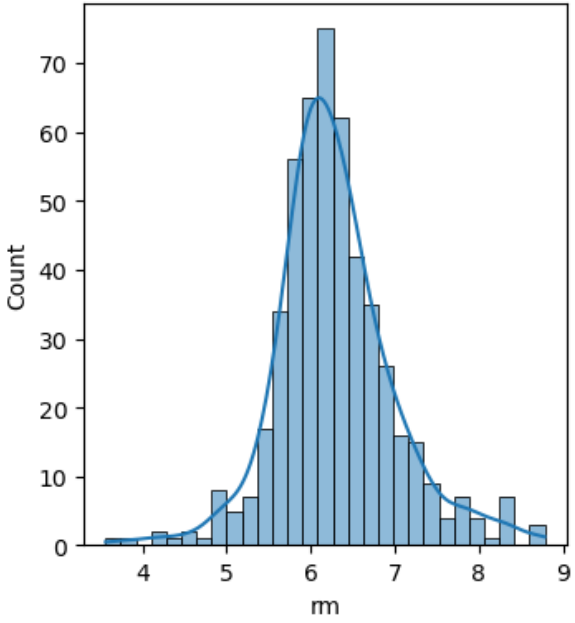

Count
crim

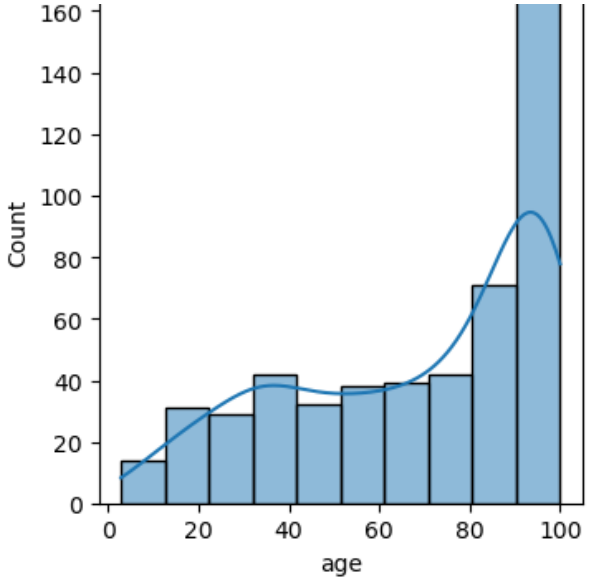

Count
zn

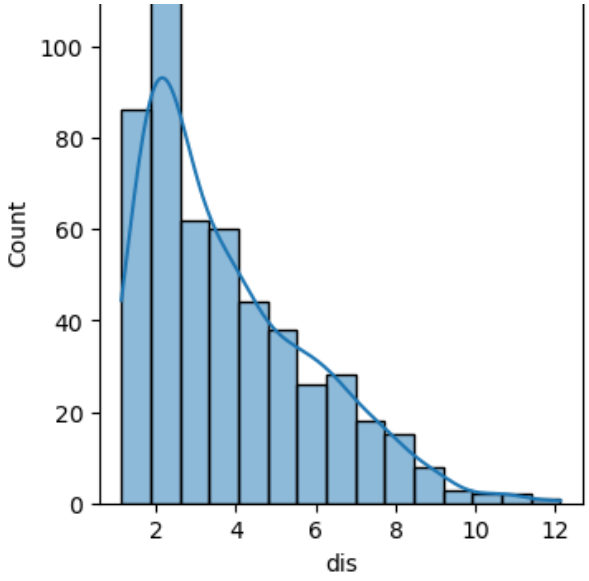

Count
indus

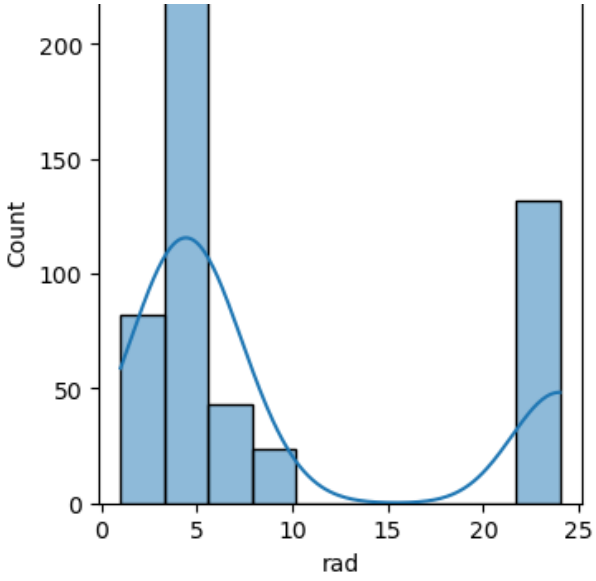

Count
chas

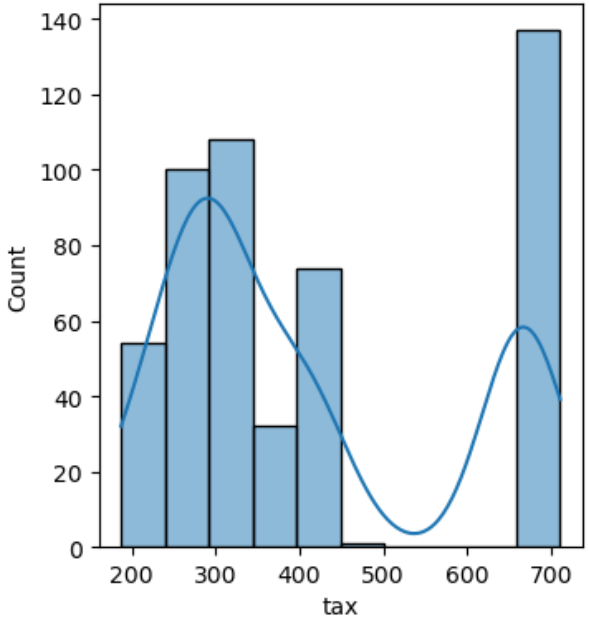

Count
nox

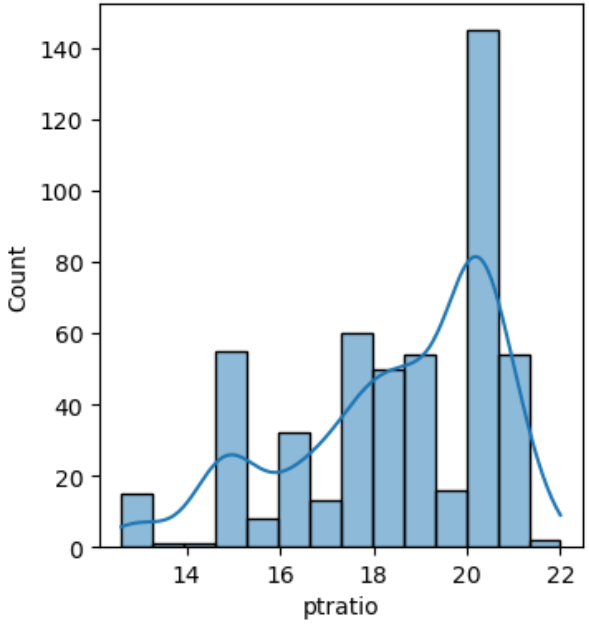

Count
rm

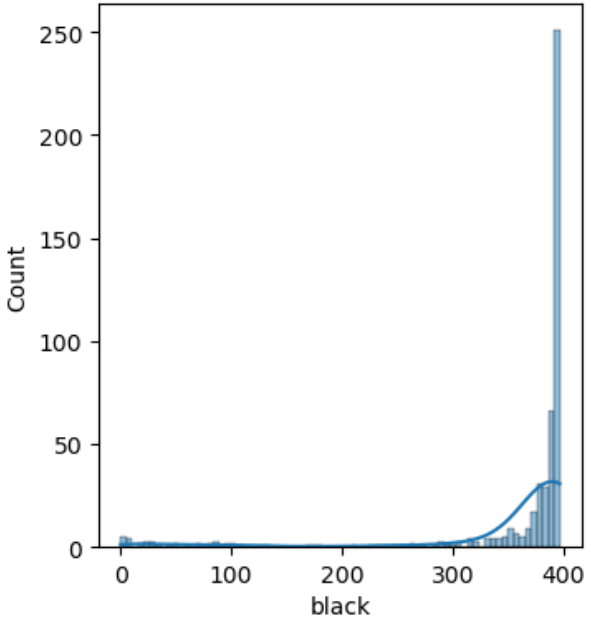

Count
age

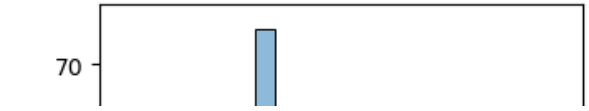

Count
dis

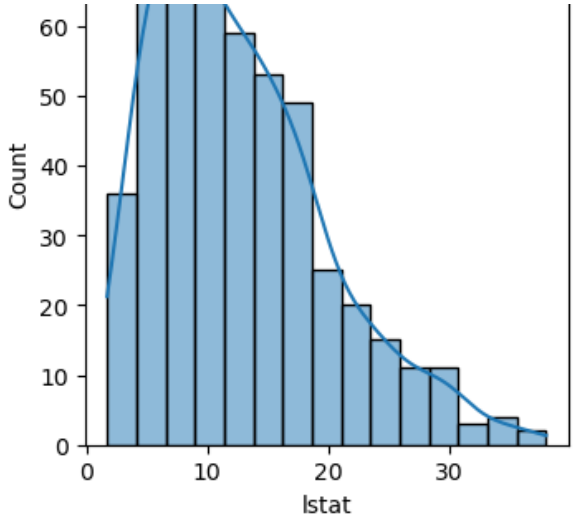

Count
rad

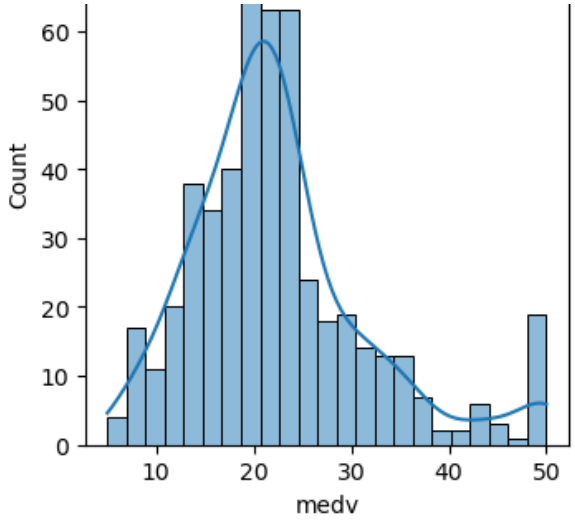

Count
tax

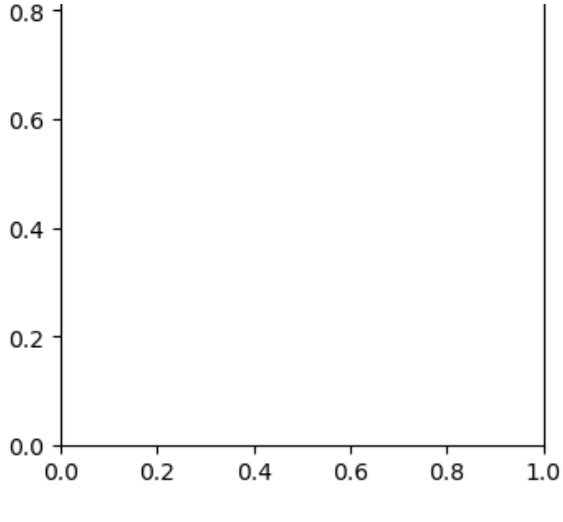

Count
ptratio


Count
black









In [112…

```
# skewness and kurtosis analysis

df_val = []
df_col= ["Column Name", "Skewness", "Kurtosis"]
for k, v in dataset.items():
    skew = v.skew()
    kurt = v.kurtosis()

    f_skew  = int(v.skew())
    f_kurt = math.floor(v.kurtosis())

    if (f_skew>=1):
        skew_val = "Right Skewed"
    elif( f_skew <=-1):
        skew_val= "Left Skewed"
    else:
        skew_val="Normal"

    if(f_kurt>3):
        kurt_val = "Leptokurtic"
    elif(f_kurt<3 and f_kurt !=0):
        kurt_val = "Platykurtic"
    elif(f_kurt==3):
        kurt_val = "Mesokurtic"
```

```
    else:
        kurt_val="Normal"

    df_val.append([k, str(skew) + " : "+ skew_val , str(kurt)+ " : "+kurt_val ])

df_skew_kurt = pd.DataFrame(data=np.array(df_val), columns=df_col)
df_skew_kurt
```

Out[112…

	Column Name	Skewness	Kurtosis
0	crim	5.223148798243851 : Right Skewed	37.13050912952203 : Leptokurtic
1	zn	2.2256663227354307 : Right Skewed	4.031510083739155 : Leptokurtic
2	indus	0.29502156787351164 : Normal	-1.2335396011495188 : Platykurtic
3	chas	3.405904172058746 : Right Skewed	9.638263777819526 : Leptokurtic
4	nox	0.7293079225348787 : Normal	-0.06466713336542629 : Platykurtic
5	rm	0.40361213328874385 : Normal	1.8915003664993404 : Platykurtic
6	age	-0.5989626398812962 : Normal	-0.9677155941626912 : Platykurtic
7	dis	1.0117805793009007 : Right Skewed	0.4879411222443908 : Normal
8	rad	1.0048146482182057 : Right Skewed	-0.8672319936034931 : Platykurtic
9	tax	0.669955941795016 : Normal	-1.1424079924768082 : Platykurtic
10	ptratio	-0.8023249268537809 : Normal	-0.28509138330538875 : Platykurtic
11	black	-2.8903737121414492 : Left Skewed	7.226817549260753 : Leptokurtic
12	lstat	0.9064600935915367 : Normal	0.49323951739272776 : Normal
13	medv	1.1080984082549072 : Right Skewed	1.495196944165818 : Platykurtic

Inference

1. The average crime rate per capita by town is 3.61.
2. On a average 11.36% of the land is residential land for every 25,000km of plot. Although around 75% of the land have almost 0% residential land.
3. Similar to the proportion of land available for residential purposes, there is 11.13% of land available for setting up factories. Although the lowest is 0.5% land which is still higher compared to the land available for residential purposes.
4. A mean of 0.06 ie 0.1 indicates that there are more housing areas that do not tract the Charles river.
5. The average Nitric oxide concentration is acceptable.
6. The average number of room per housing structure is 6 rooms, withe the maximum being 9 rooms.
7. About 65.6% of the housing structures are built prior to 1940 and still the owners reside in them.
8. The average travelling distance to 5 Boston workplaces is rought 4km.
9. The average tax rate is 4% with 408.2 dollars for every 10,000 dollars.
10. There is on average 1 teacher for every 19 pupils. with the lowest being 0.32 meaning that there are more teachers than pupil themselves and with a max of 396 pupils to be handled by one teacher indicating less teachers.
11. 12.65 % of the population is of lower social economic status, although 75% of the values are below 11.5%.
12. The median price is $22,500 dollars.
13. The skewness, kurtosis and outliers can be seen in the above tables.