In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Perform the data wrangling/Data analysis for the given dataset housing.csv

In [3]:
```
df = pd.read_csv('housing.csv')
df.drop(['Unnamed: 0'],axis=1,inplace=True)
df.head()
```

Out[3]:

	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

In [20]:
```
df.isnull().any()
```

Out[20]:
```
crim       False
zn         False
indus      False
chas       False
nox        False
rm         False
age        False
dis        False
rad        False
tax        False
ptratio    False
black      False
lstat      False
medv       False
dtype: bool
```

This dataframe does not contains any null vlue

Apply basic statistics such as central value, variability and distribution. Visualize them using Box plots

In [21]:
```
df.describe()
```

Out[21]:

	crim	zn	indus	chas	nox	rm	age	dis
count	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000	506.000000
mean	3.613524	11.363636	11.136779	0.069170	0.554695	6.284634	68.574901	3.795043

	crim	zn	indus	chas	nox	rm	age	dis
std	8.601545	23.322453	6.860353	0.253994	0.115878	0.702617	28.148861	2.105710
min	0.006320	0.000000	0.460000	0.000000	0.385000	3.561000	2.900000	1.129600
25%	0.082045	0.000000	5.190000	0.000000	0.449000	5.885500	45.025000	2.100175
50%	0.256510	0.000000	9.690000	0.000000	0.538000	6.208500	77.500000	3.207450
75%	3.677083	12.500000	18.100000	0.000000	0.624000	6.623500	94.075000	5.188425
max	88.976200	100.000000	27.740000	1.000000	0.871000	8.780000	100.000000	12.126500

Mean of each columns

In [4]:
```
df.mean()
```

Out[4]:
```
crim          3.613524
zn           11.363636
indus        11.136779
chas          0.069170
nox           0.554695
rm            6.284634
age          68.574901
dis           3.795043
rad           9.549407
tax         408.237154
ptratio      18.455534
black       356.674032
lstat        12.653063
medv         22.532806
dtype: float64
```

Median for each columns

In [5]:
```
df.median()
```

Out[5]:
```
crim          0.25651
zn            0.00000
indus         9.69000
chas          0.00000
nox           0.53800
rm            6.20850
age          77.50000
dis           3.20745
rad           5.00000
tax         330.00000
ptratio      19.05000
black       391.44000
lstat        11.36000
medv         21.20000
dtype: float64
```

Standard Deviation for each column

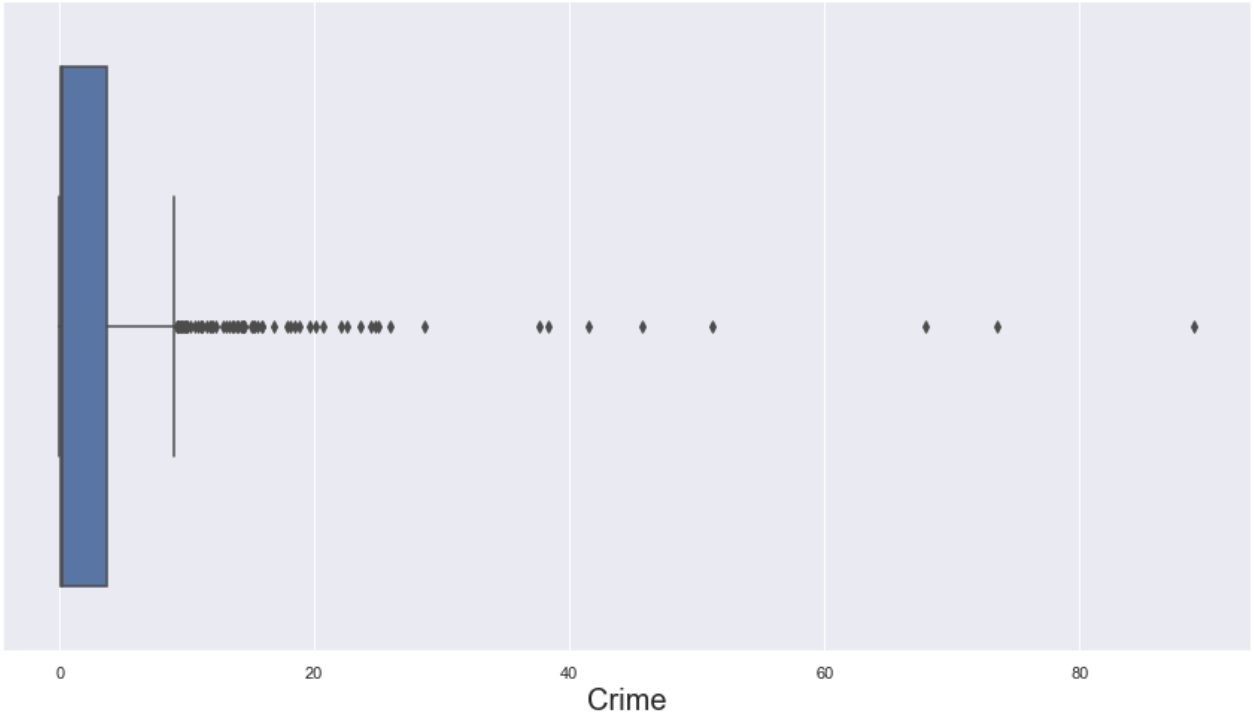
In [7]:
```
df.std()
```

Out[7]:
```
crim          8.601545
```

```
zn          23.322453
indus        6.860353
chas         0.253994
nox          0.115878
rm           0.702617
age         28.148861
dis          2.105710
rad          8.707259
tax        168.537116
ptratio      2.164946
black       91.294864
lstat        7.141062
medv         9.197104
dtype: float64
```

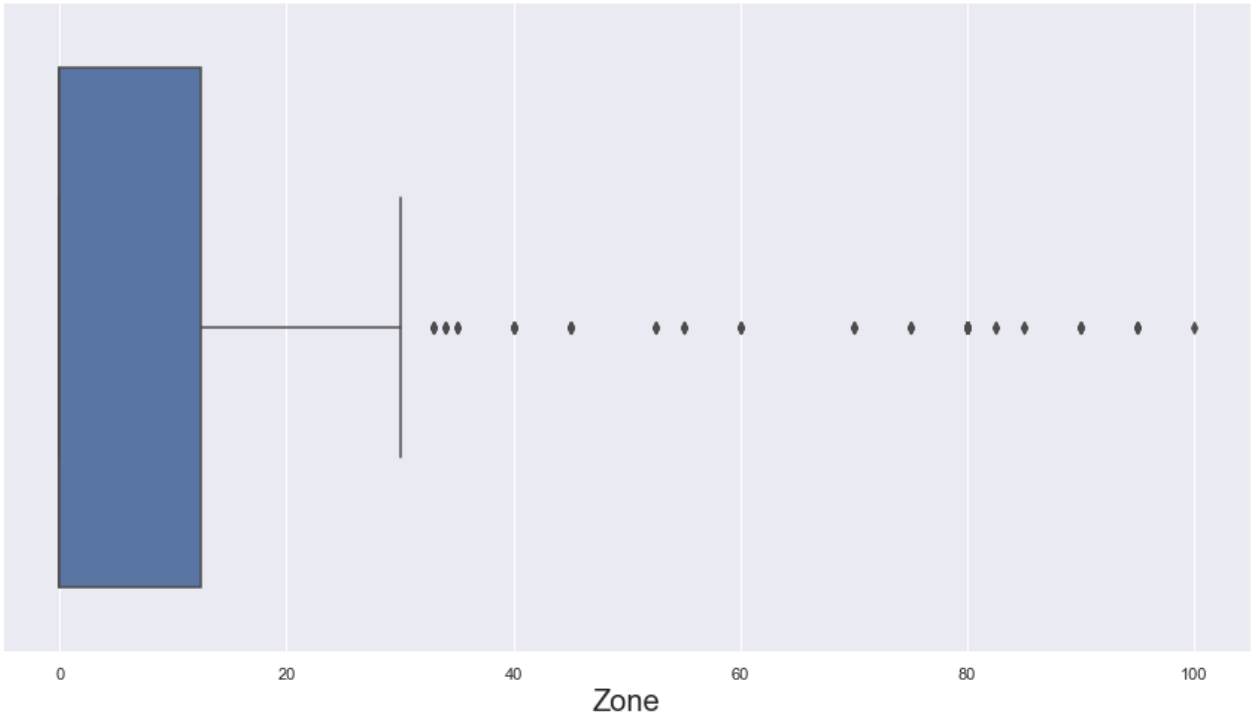
In [9]:
```
sns.boxplot(x=df['crim'])
sns.set(rc = {'figure.figsize':(15,8)})
plt.xlabel('Crime',fontsize=20)
```

Out[9]: Text(0.5, 0, 'Crime')



In [10]:
```
sns.boxplot(x=df['zn'])
sns.set(rc = {'figure.figsize':(15,8)})
plt.xlabel('Zone',fontsize=20)
```

Out[10]: Text(0.5, 0, 'Zone')



In [11]:
```
sns.boxplot(x=df['indus'])
sns.set(rc = {'figure.figsize':(15,8)})
plt.xlabel('indus(Proportion of non-retail business acres per town)',fontsize=20)
```

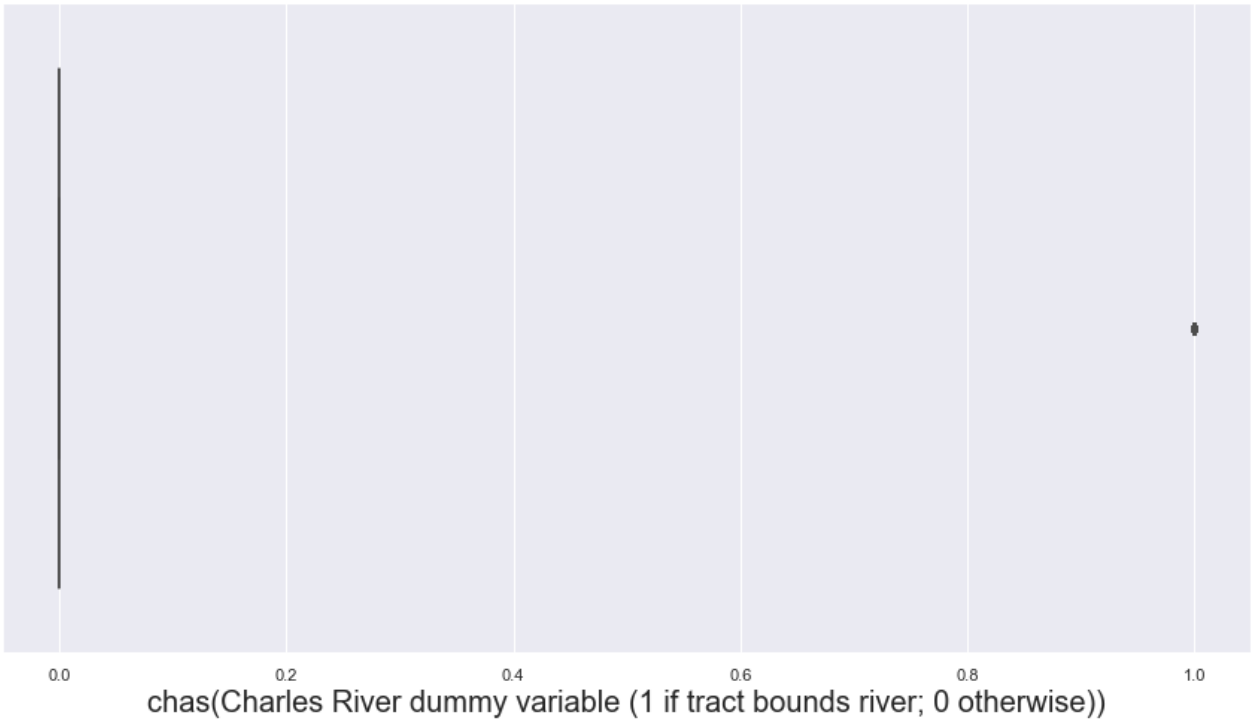
Out[11]: Text(0.5, 0, 'indus(Proportion of non-retail business acres per town)')

In [12]:
```
sns.boxplot(x=df['chas'])
sns.set(rc = {'figure.figsize':(15,8)})
plt.xlabel('ZN(proportion of residential land zoned for lots over 25,000 sq.ft)',fontsi
```

Out[12]: Text(0.5, 0, 'ZN(proportion of residential land zoned for lots over 25,000 sq.ft)')

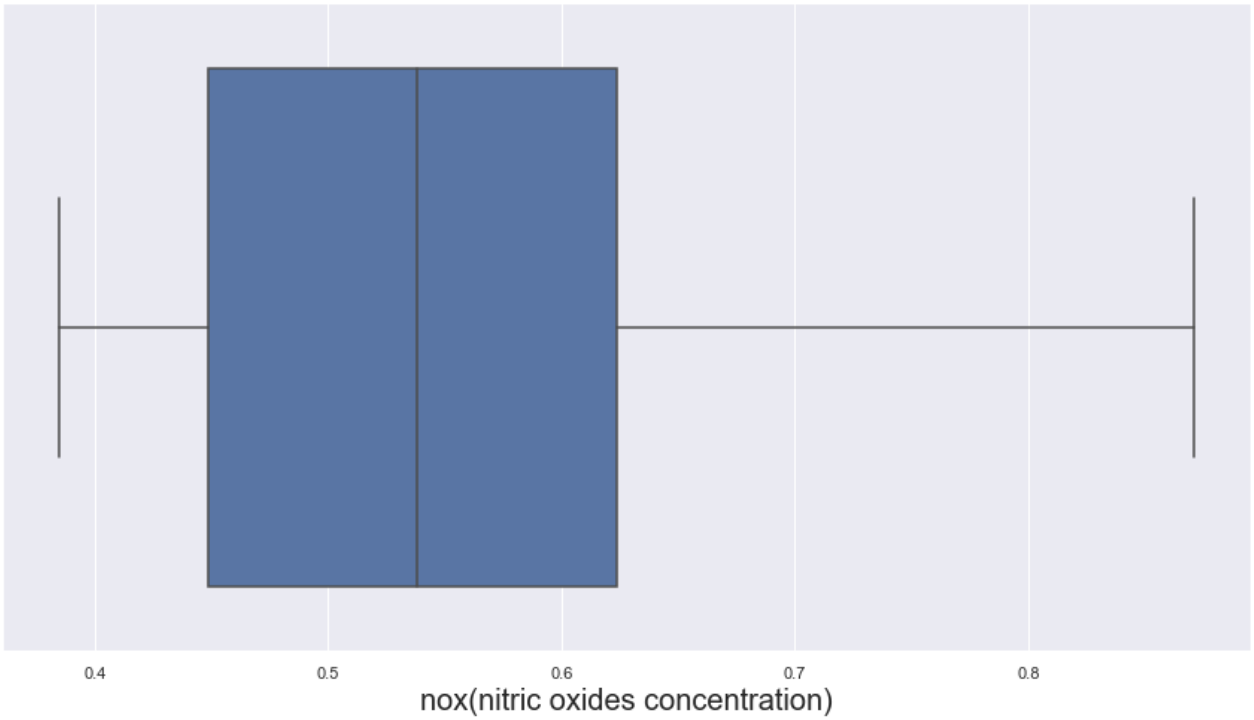
In [14]:
```
sns.boxplot(x=df['chas'])
sns.set(rc = {'figure.figsize':(15,8)})
plt.xlabel('chas(Charles River dummy variable (1 if tract bounds river; 0 otherwise))',
```

Out[14]:
```
Text(0.5, 0, 'chas(Charles River dummy variable (1 if tract bounds river; 0 otherwis
e))')
```



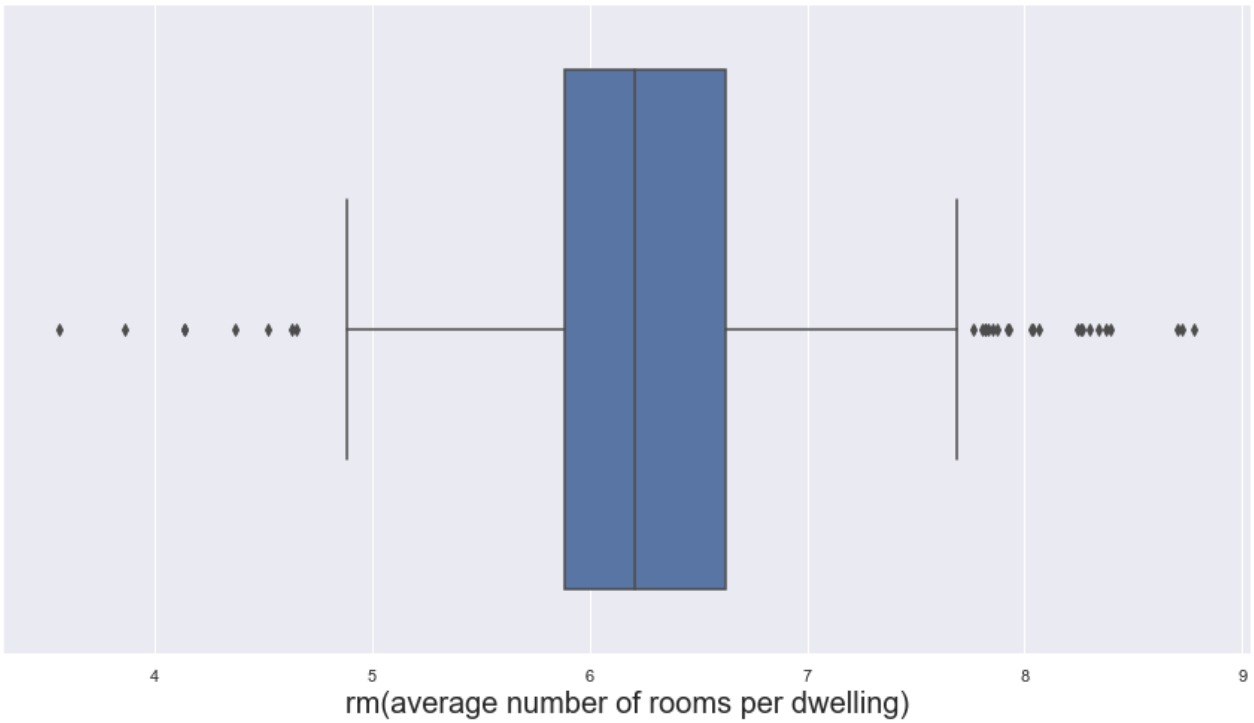
In [15]:
```
sns.boxplot(x=df['nox'])
sns.set(rc = {'figure.figsize':(15,8)})
plt.xlabel('nox(nitric oxides concentration)',fontsize=20)
```

Out[15]:
```
Text(0.5, 0, 'nox(nitric oxides concentration)')
```



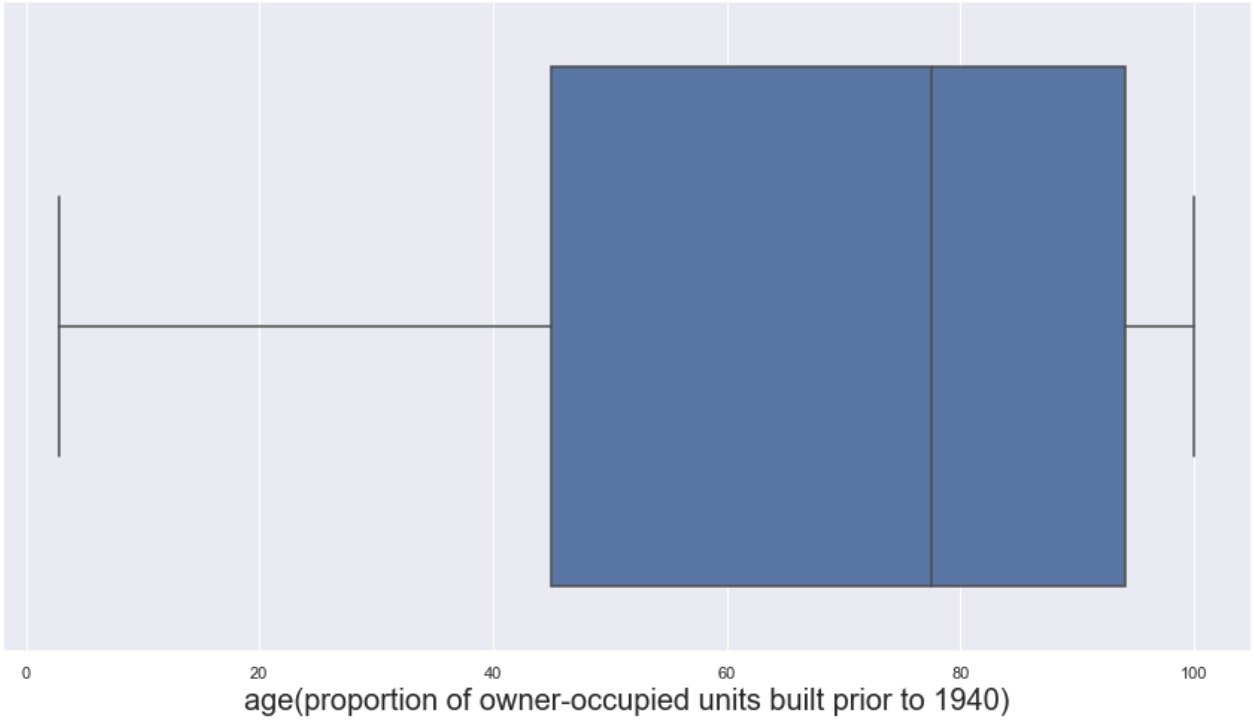
In [49]:
```
sns.boxplot(x=df['rm'])
sns.set(rc = {'figure.figsize':(15,8)})
plt.xlabel('rm(average number of rooms per dwelling)',fontsize=20)
```

Out[49]: Text(0.5, 0, 'rm(average number of rooms per dwelling)')



In [50]:
```
sns.boxplot(x=df['age'])
sns.set(rc = {'figure.figsize':(15,8)})
plt.xlabel('age(proportion of owner-occupied units built prior to 1940)',fontsize=20)
```

Out[50]: Text(0.5, 0, 'age(proportion of owner-occupied units built prior to 1940)')

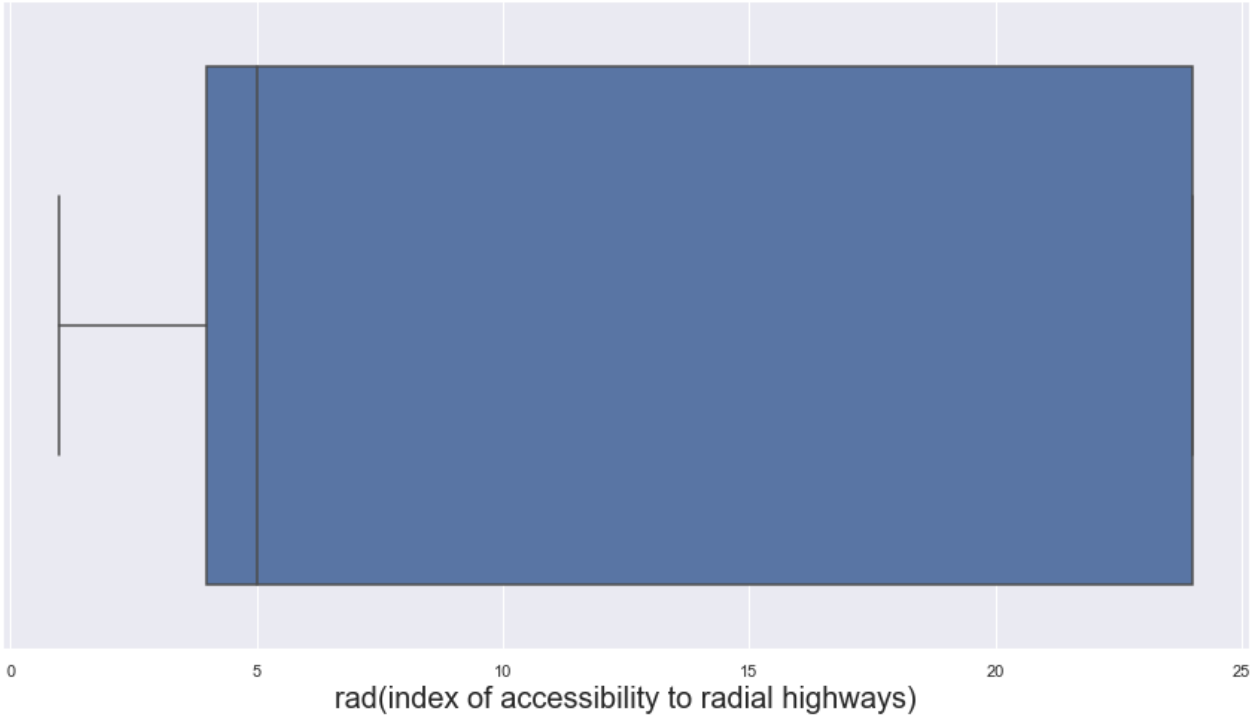


In [51]:
```
sns.boxplot(x=df['dis'])
sns.set(rc = {'figure.figsize':(15,8)})
plt.xlabel('dis(weighted distances to five Boston employment centres)',fontsize=20)
```

Out[51]: Text(0.5, 0, 'dis(weighted distances to five Boston employment centres)')

In [52]:
```
sns.boxplot(x=df['rad'])
sns.set(rc = {'figure.figsize':(15,8)})
plt.xlabel('rad(index of accessibility to radial highways)',fontsize=20)
```

Out[52]: Text(0.5, 0, 'rad(index of accessibility to radial highways)')

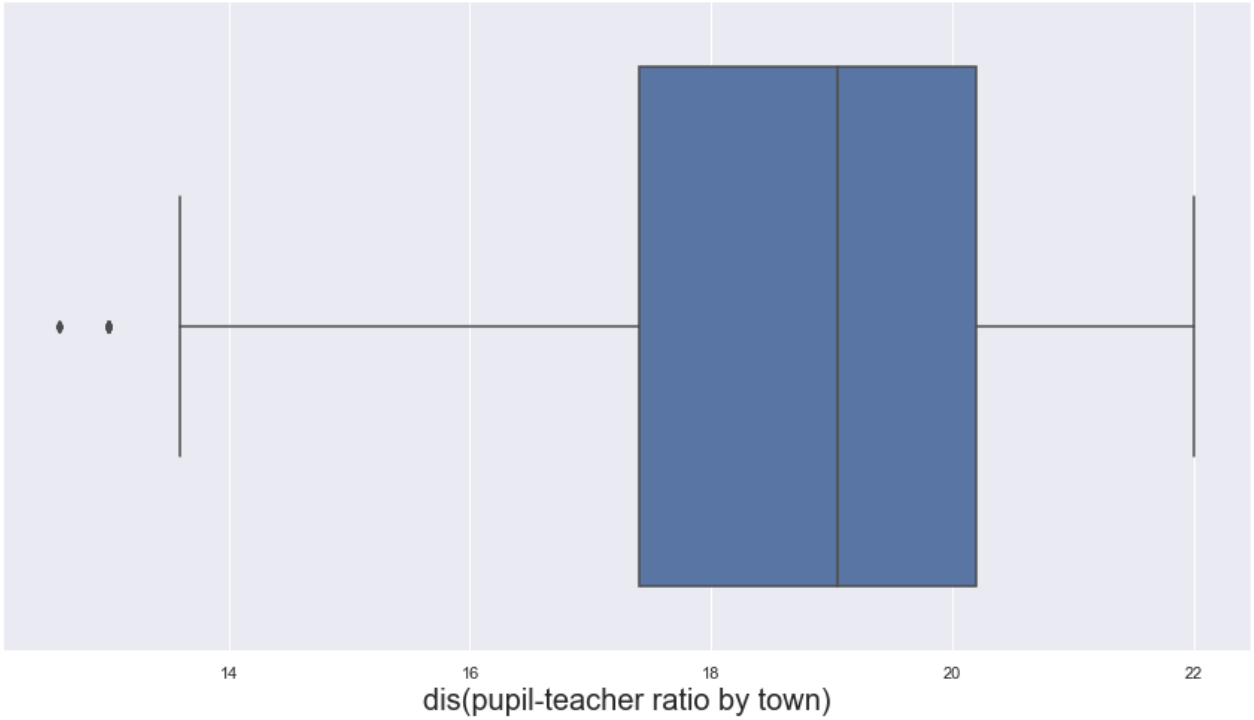


In [53]:
```
sns.boxplot(x=df['tax'])
sns.set(rc = {'figure.figsize':(15,8)})
plt.xlabel('tax(full-value property-tax rate per $10,000)',fontsize=20)
```

Out[53]: Text(0.5, 0, 'tax(full-value property-tax rate per $10,000)')

In [54]:
```
sns.boxplot(x=df['ptratio'])
sns.set(rc = {'figure.figsize':(15,8)})
plt.xlabel('dis(pupil-teacher ratio by town)',fontsize=20)
```

Out[54]: Text(0.5, 0, 'dis(pupil-teacher ratio by town)')

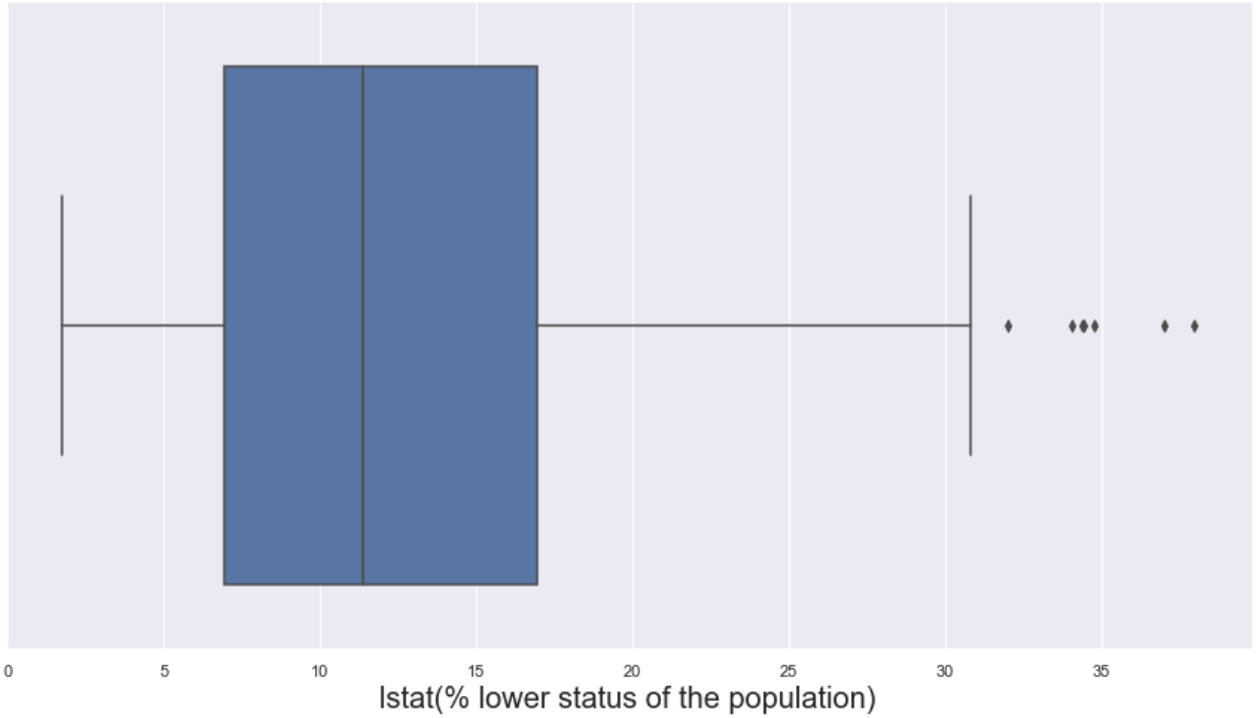


In [56]:
```
sns.boxplot(x=df['dis'])
sns.set(rc = {'figure.figsize':(15,8)})
plt.xlabel('blck(B - 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town',fo
```

Out[56]: Text(0.5, 0, 'blck(B - 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town')

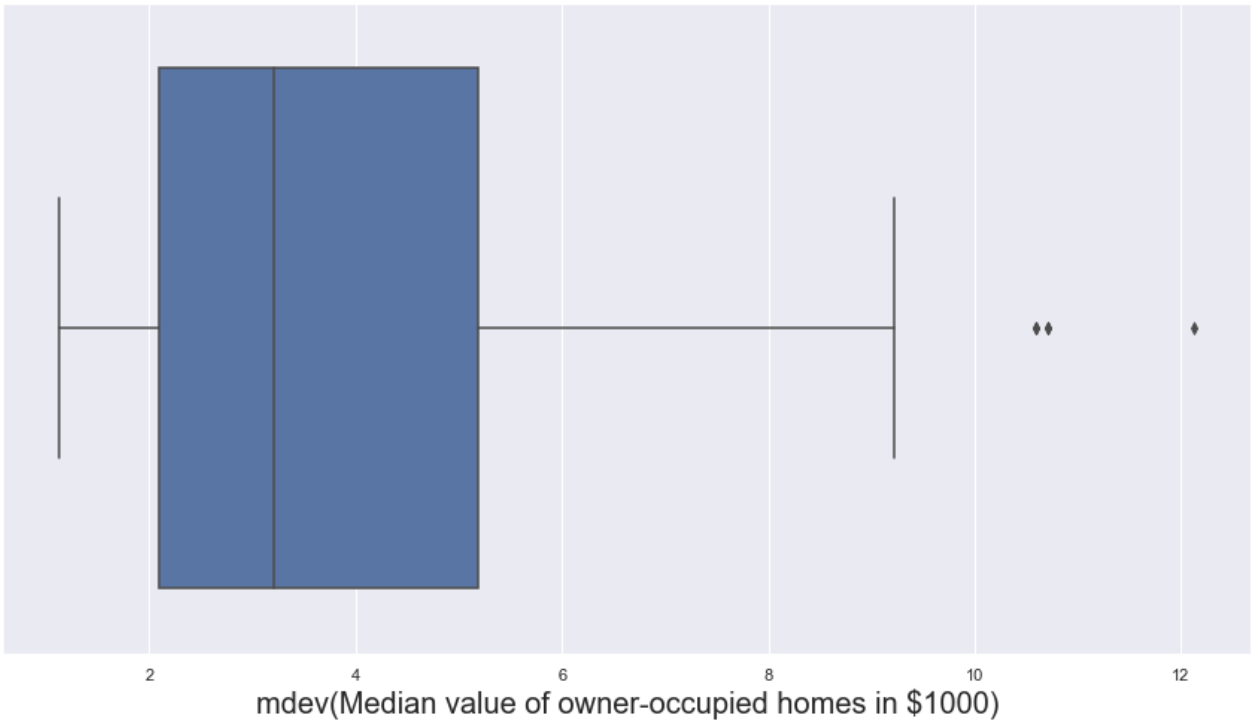
In [57]:
```
sns.boxplot(x=df['lstat'])
sns.set(rc = {'figure.figsize':(15,8)})
plt.xlabel('lstat(% lower status of the population)',fontsize=20)
```

Out[57]: Text(0.5, 0, 'lstat(% lower status of the population)')



In [60]:
```
sns.boxplot(x=df['dis'])
sns.set(rc = {'figure.figsize':(15,8)})
plt.xlabel('mdev(Median value of owner-occupied homes in $1000)',fontsize=20)
```

Out[60]: Text(0.5, 0, 'mdev(Median value of owner-occupied homes in $1000)')



Check for relationships between variables and visualize them using Heatmap

In [61]:
```
df.corr()
```

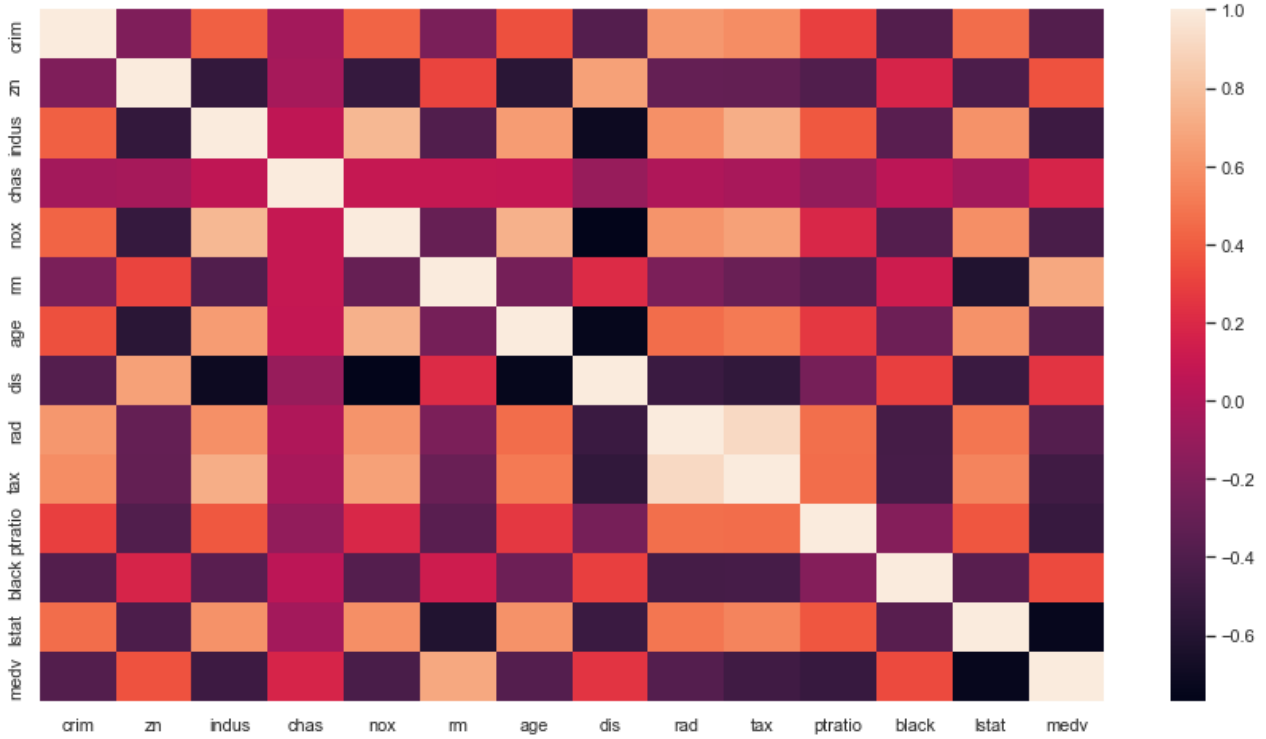
Out[61]:

	crim	zn	indus	chas	nox	rm	age	dis	rad
crim	1.000000	-0.200469	0.406583	-0.055892	0.420972	-0.219247	0.352734	-0.379670	0.625505
zn	-0.200469	1.000000	-0.533828	-0.042697	-0.516604	0.311991	-0.569537	0.664408	-0.311948
indus	0.406583	-0.533828	1.000000	0.062938	0.763651	-0.391676	0.644779	-0.708027	0.595129
chas	-0.055892	-0.042697	0.062938	1.000000	0.091203	0.091251	0.086518	-0.099176	-0.007368
nox	0.420972	-0.516604	0.763651	0.091203	1.000000	-0.302188	0.731470	-0.769230	0.611441
rm	-0.219247	0.311991	-0.391676	0.091251	-0.302188	1.000000	-0.240265	0.205246	-0.209847
age	0.352734	-0.569537	0.644779	0.086518	0.731470	-0.240265	1.000000	-0.747881	0.456022
dis	-0.379670	0.664408	-0.708027	-0.099176	-0.769230	0.205246	-0.747881	1.000000	-0.494588
rad	0.625505	-0.311948	0.595129	-0.007368	0.611441	-0.209847	0.456022	-0.494588	1.000000
tax	0.582764	-0.314563	0.720760	-0.035587	0.668023	-0.292048	0.506456	-0.534432	0.910228
ptratio	0.289946	-0.391679	0.383248	-0.121515	0.188933	-0.355501	0.261515	-0.232471	0.464741
black	-0.385064	0.175520	-0.356977	0.048788	-0.380051	0.128069	-0.273534	0.291512	-0.444413
lstat	0.455621	-0.412995	0.603800	-0.053929	0.590879	-0.613808	0.602339	-0.496996	0.488676
medv	-0.388305	0.360445	-0.483725	0.175260	-0.427321	0.695360	-0.376955	0.249929	-0.381626

In [62]:

```
sns.heatmap(df.corr())
```

Out[62]: <AxesSubplot:>



In [ ]: