import all required packages

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
path=r"C:\Users\Sruth\Downloads\achou-nba-draft-combine-measurements\achou-nba-d
df=pd.read_csv(path)
df
```

Out[2]:

	column_a	player	year	draft_pick	height_no_shoes	height_with_shoes	wing
0	0	Blake Griffin	2009	1.0	80.50	82.00	
1	1	Terrence Williams	2009	11.0	77.00	78.25	
2	2	Gerald Henderson	2009	12.0	76.00	77.00	
3	3	Tyler Hansbrough	2009	13.0	80.25	81.50	
4	4	Earl Clark	2009	14.0	80.50	82.25	
...	...	...	...	...	...	...	
512	512	Peter Jok	2017	NaN	76.25	77.75	
513	513	Rawle Alkins	2017	NaN	74.50	75.75	
514	514	Sviatoslav Mykhailiuk	2017	NaN	78.50	79.50	
515	515	Thomas Welsh	2017	NaN	83.50	84.50	
516	516	V.J. Beachem	2017	NaN	78.25	80.00	

517 rows × 19 columns

Data quick checks

1.shape

In [3]:
```
# there is 517rows and 19 columns
df.shape
```

Out[3]:
```
(517, 19)
```

2.size

In [4]:
```
# which returns overall size of dataframe
```

```
df.size
```

Out[4]: 9823

3.dtypes

In [5]:
```
# gives type of column
df.dtypes
```

Out[5]:
```
column_a                   int64
player                    object
year                       int64
draft_pick               float64
height_no_shoes          float64
height_with_shoes        float64
wingspan                 float64
standing_reach           float64
vertical_max             float64
vertical_max_reach       float64
vertical_no_step         float64
vertical_no_step_reach   float64
weight                   float64
body_fat                 float64
hand_length              float64
hand_width               float64
bench                    float64
agility                  float64
sprint                   float64
dtype: object
```

4.columns

In [6]:
```
# gives total columns present in the dataset
df.columns
```

Out[6]:
```
Index(['column_a', 'player', 'year', 'draft_pick', 'height_no_shoes',
       'height_with_shoes', 'wingspan', 'standing_reach', 'vertical_max',
       'vertical_max_reach', 'vertical_no_step', 'vertical_no_step_reach',
       'weight', 'body_fat', 'hand_length', 'hand_width', 'bench', 'agility',
       'sprint'],
      dtype='object')
```

5.info

In [7]:
```
# gives information about the colums that is either categorical column(object),n
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 517 entries, 0 to 516
Data columns (total 19 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   column_a                517 non-null    int64
 1   player                  517 non-null    object
 2   year                    517 non-null    int64
 3   draft_pick              384 non-null    float64
 4   height_no_shoes         517 non-null    float64
 5   height_with_shoes       516 non-null    float64
 6   wingspan                517 non-null    float64
 7   standing_reach          517 non-null    float64
 8   vertical_max            450 non-null    float64
 9   vertical_max_reach      450 non-null    float64
 10  vertical_no_step        450 non-null    float64
 11  vertical_no_step_reach  450 non-null    float64
 12  weight                  516 non-null    float64
 13  body_fat                514 non-null    float64
 14  hand_length             470 non-null    float64
 15  hand_width              468 non-null    float64
 16  bench                   284 non-null    float64
 17  agility                 444 non-null    float64
 18  sprint                  446 non-null    float64
dtypes: float64(16), int64(2), object(1)
memory usage: 76.9+ KB
```

Convert categorical columns

In [8]:
```
cat_cols=df.select_dtypes(include="object").columns
cat_cols
```

Out[8]:
```
Index(['player'], dtype='object')
```

conversion of numerical columns

In [9]:
```
num_cols=df.select_dtypes(exclude="object").columns
num_cols
```

Out[9]:
```
Index(['column_a', 'year', 'draft_pick', 'height_no_shoes',
       'height_with_shoes', 'wingspan', 'standing_reach', 'vertical_max',
       'vertical_max_reach', 'vertical_no_step', 'vertical_no_step_reach',
       'weight', 'body_fat', 'hand_length', 'hand_width', 'bench', 'agility',
       'sprint'],
      dtype='object')
```

checking the missing values

In [10]:
```
# 0 represents no missing values: except 0 any numerical values represents missi
df.isnull().sum()
```

Out[10]:
```
column_a                    0
player                      0
year                        0
draft_pick                133
height_no_shoes             0
height_with_shoes           1
wingspan                    0
standing_reach              0
vertical_max               67
vertical_max_reach         67
vertical_no_step           67
vertical_no_step_reach     67
weight                      1
body_fat                    3
hand_length                47
hand_width                 49
bench                     233
agility                    73
sprint                     71
dtype: int64
```

From above there are missing values

Filling the missing values

In [11]:
```
df=pd.read_csv(path)
draft_pick=df['draft_pick'].mode()
df['draft_pick'].fillna(draft_pick[0],inplace=True)

height_with_shoes=df['height_with_shoes'].mode()
df['height_with_shoes'].fillna(height_with_shoes[0],inplace=True)

vertical_max=df['vertical_max'].mode()
df['vertical_max'].fillna(vertical_max[0],inplace=True)

vertical_max_reach=df['vertical_max_reach'].mode()
df['vertical_max_reach'].fillna(vertical_max_reach[0],inplace=True)

vertical_no_step=df['vertical_no_step'].mode()
df['vertical_no_step'].fillna(vertical_no_step[0],inplace=True)

vertical_no_step_reach=df['vertical_no_step_reach'].mode()
df['vertical_no_step_reach'].fillna(vertical_no_step_reach[0],inplace=True)

weight=df['weight'].mode()
df['weight'].fillna(weight[0],inplace=True)

body_fat=df['body_fat'].mode()
df['body_fat'].fillna(body_fat[0],inplace=True)

hand_length=df['hand_length'].mode()
df['hand_length'].fillna(hand_length[0],inplace=True)

hand_width=df['hand_width'].mode()
df['hand_width'].fillna(hand_width[0],inplace=True)

bench=df['bench'].mode()
df['bench'].fillna(bench[0],inplace=True)
```

```
agility=df['agility'].mode()
df['agility'].fillna(agility[0],inplace=True)

sprint=df['sprint'].mode()
df['sprint'].fillna(sprint[0],inplace=True)
```

In [12]:
```
# dataframe without any missing values
df
```

Out[12]:

	column_a	player	year	draft_pick	height_no_shoes	height_with_shoes	winc
0	0	Blake Griffin	2009	1.0	80.50	82.00	
1	1	Terrence Williams	2009	11.0	77.00	78.25	
2	2	Gerald Henderson	2009	12.0	76.00	77.00	
3	3	Tyler Hansbrough	2009	13.0	80.25	81.50	
4	4	Earl Clark	2009	14.0	80.50	82.25	
...	...	...	...	...	...	...	
512	512	Peter Jok	2017	14.0	76.25	77.75	
513	513	Rawle Alkins	2017	14.0	74.50	75.75	
514	514	Sviatoslav Mykhailiuk	2017	14.0	78.50	79.50	
515	515	Thomas Welsh	2017	14.0	83.50	84.50	
516	516	V.J. Beachem	2017	14.0	78.25	80.00	

517 rows × 19 columns

No missing values , all the missing values are filled above

In [13]:
```
# there is no missing values in dataset
df.isnull().sum()
```

Out[13]:
```
column_a                 0
player                   0
year                     0
draft_pick               0
height_no_shoes          0
height_with_shoes        0
wingspan                 0
standing_reach           0
vertical_max             0
vertical_max_reach       0
vertical_no_step         0
vertical_no_step_reach   0
weight                   0
body_fat                 0
hand_length              0
hand_width               0
bench                    0
agility                  0
sprint                   0
dtype: int64
```

Categorical Data Analysis

In [14]:
```
cat_cols
```

Out[14]:
```
Index(['player'], dtype='object')
```

In [15]:
```
# There are 516 unique items in dataframe
df['player'].nunique()
```

Out[15]:
```
516
```

In [16]:
```
df['player'].value_counts()
```

Out[16]:
```
player
Marcus Thornton     2
Jordan Mickey       1
Kevon Looney        1
Chris McCullough    1
R.J. Hunter         1
                   ..
Tony Wroten         1
Jared Cunningham    1
John Jenkins        1
Fab Melo            1
V.J. Beachem        1
Name: count, Length: 516, dtype: int64
```

Numerical Column Analysis

In [17]:
```
num_cols
```

Out[17]:
```
Index(['column_a', 'year', 'draft_pick', 'height_no_shoes',
       'height_with_shoes', 'wingspan', 'standing_reach', 'vertical_max',
       'vertical_max_reach', 'vertical_no_step', 'vertical_no_step_reach',
       'weight', 'body_fat', 'hand_length', 'hand_width', 'bench', 'agility',
       'sprint'],
      dtype='object')
```

Method-Correlation

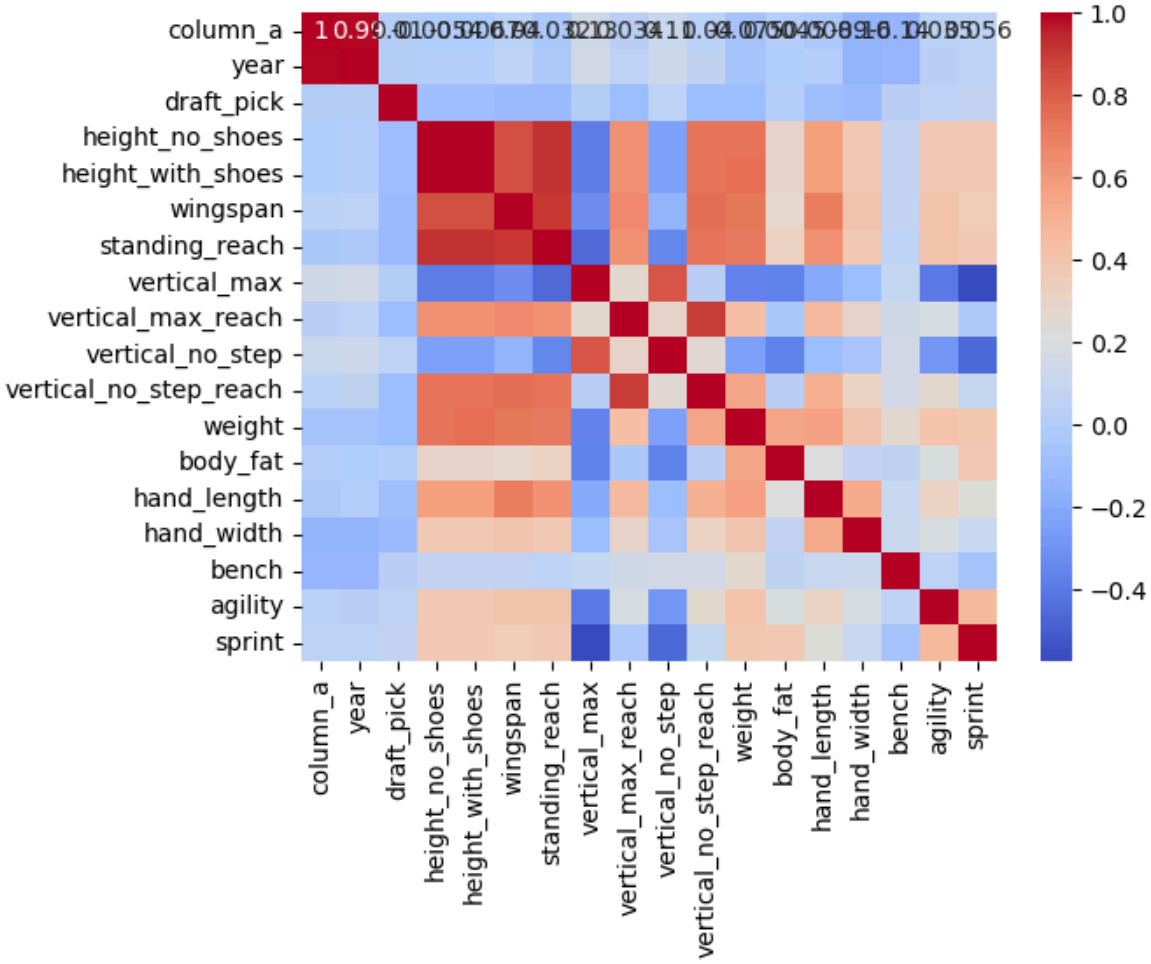
In [18]:
```
df.corr(numeric_only=True)
```

Out[18]:

	column_a	year	draft_pick	height_no_shoes	height_with_sh
column_a	1.000000	0.992944	0.009979	-0.005443	-0.006
year	0.992944	1.000000	0.012478	0.005332	0.004
draft_pick	0.009979	0.012478	1.000000	-0.093051	-0.091
height_no_shoes	-0.005443	0.005332	-0.093051	1.000000	0.995
height_with_shoes	-0.006703	0.004975	-0.091260	0.995220	1.000
wingspan	0.040212	0.051316	-0.114980	0.846048	0.843
standing_reach	-0.032146	-0.024578	-0.109112	0.921464	0.920
vertical_max	0.133720	0.147984	0.012902	-0.378723	-0.385
vertical_max_reach	0.033574	0.051841	-0.100302	0.629287	0.622
vertical_no_step	0.106966	0.123862	0.043796	-0.235172	-0.243
vertical_no_step_reach	0.040380	0.060176	-0.101921	0.735780	0.730
weight	-0.075281	-0.074153	-0.095277	0.741385	0.743
body_fat	0.004468	-0.007667	0.006432	0.280954	0.280
hand_length	-0.008950	0.001826	-0.081943	0.578244	0.581
hand_width	-0.158802	-0.159639	-0.111546	0.367684	0.369
bench	-0.135496	-0.140121	0.020223	0.089620	0.087
agility	0.035009	0.021478	0.045494	0.382487	0.379
sprint	0.056455	0.044251	0.075731	0.382458	0.375

Heat_map

In [19]:
```
corr_data=df.corr(numeric_only=True)
sns.heatmap(corr_data,annot=True,cmap='coolwarm')
plt.show()
```



pre defined function

In [20]:
```
df.describe()
```

Out[20]:

	column_a	year	draft_pick	height_no_shoes	height_with_shoes	wings
count	517.000000	517.000000	517.000000	517.000000	517.000000	517.00
mean	258.000000	2013.187621	24.764023	77.609284	78.901838	82.49
std	149.389312	2.531507	15.086264	3.287633	3.273419	3.94
min	0.000000	2009.000000	1.000000	68.250000	69.500000	70.00
25%	129.000000	2011.000000	14.000000	75.250000	76.750000	79.75
50%	258.000000	2013.000000	19.000000	77.750000	79.000000	82.50
75%	387.000000	2015.000000	37.000000	80.000000	81.250000	85.50
max	516.000000	2017.000000	60.000000	85.250000	86.500000	92.50

Histogram analysis for numerical columns

In [21]:
```
num_cols
```

Out[21]:
```
Index(['column_a', 'year', 'draft_pick', 'height_no_shoes',
       'height_with_shoes', 'wingspan', 'standing_reach', 'vertical_max',
       'vertical_max_reach', 'vertical_no_step', 'vertical_no_step_reach',
       'weight', 'body_fat', 'hand_length', 'hand_width', 'bench', 'agility',
       'sprint'],
      dtype='object')
```

In [22]:
```
for i in num_cols:
    skew=round(df[i].skew(),2)
    print(f"skew of  column {i} is {skew}")
```

```
skew of  column column_a is 0.0
skew of  column year is -0.1
skew of  column draft_pick is 0.63
skew of  column height_no_shoes is -0.24
skew of  column height_with_shoes is -0.24
skew of  column wingspan is -0.24
skew of  column standing_reach is -0.24
skew of  column vertical_max is 0.01
skew of  column vertical_max_reach is -0.52
skew of  column vertical_no_step is 0.24
skew of  column vertical_no_step_reach is -0.45
skew of  column weight is 0.32
skew of  column body_fat is 1.32
skew of  column hand_length is 0.11
skew of  column hand_width is 0.13
skew of  column bench is 0.3
skew of  column agility is 0.91
skew of  column sprint is 0.77
```

In [23]:
```
for i in num_cols:
    kurt=round(df[i].kurt(),2)
    print(f"kurtosis of  column {i} is {skew}")
```

```
kurtosis of  column column_a is 0.77
kurtosis of  column year is 0.77
kurtosis of  column draft_pick is 0.77
kurtosis of  column height_no_shoes is 0.77
kurtosis of  column height_with_shoes is 0.77
kurtosis of  column wingspan is 0.77
kurtosis of  column standing_reach is 0.77
kurtosis of  column vertical_max is 0.77
kurtosis of  column vertical_max_reach is 0.77
kurtosis of  column vertical_no_step is 0.77
kurtosis of  column vertical_no_step_reach is 0.77
kurtosis of  column weight is 0.77
kurtosis of  column body_fat is 0.77
kurtosis of  column hand_length is 0.77
kurtosis of  column hand_width is 0.77
kurtosis of  column bench is 0.77
kurtosis of  column agility is 0.77
kurtosis of  column sprint is 0.77
```

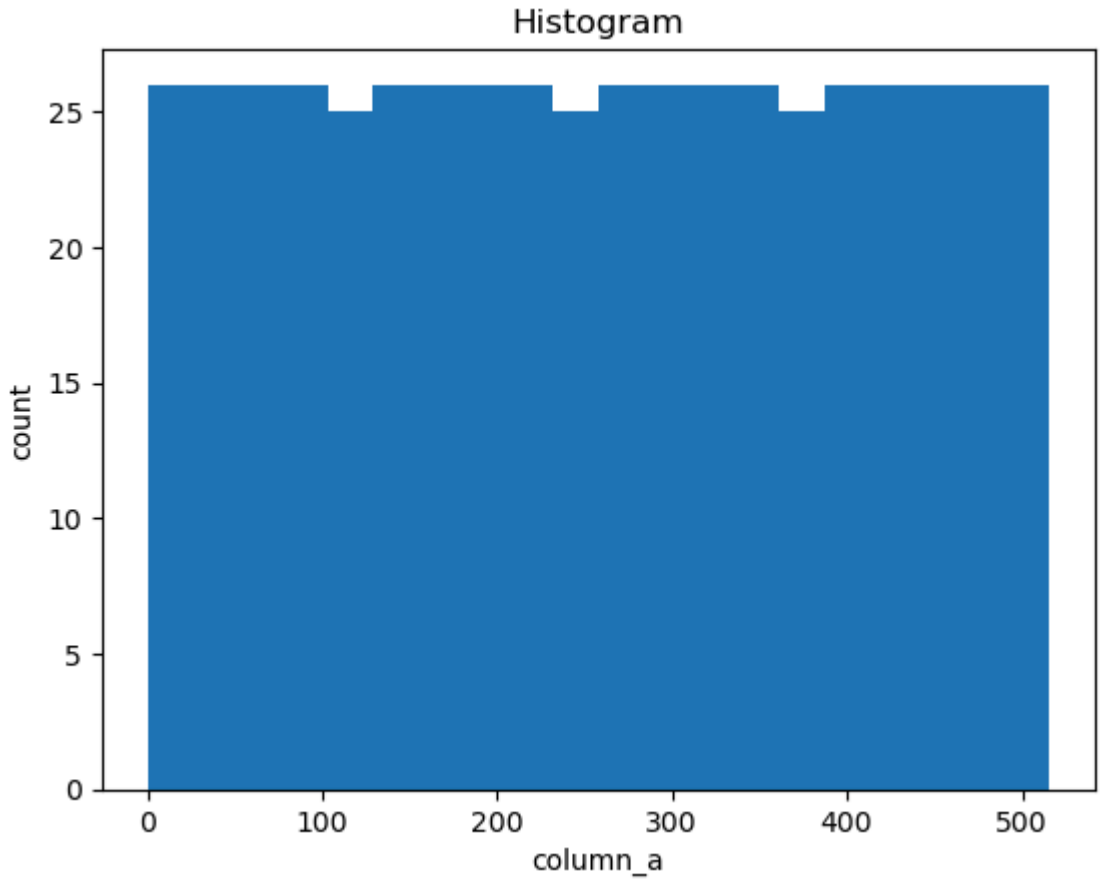
In [24]:
```
len(num_cols)
```

Out[24]: 18

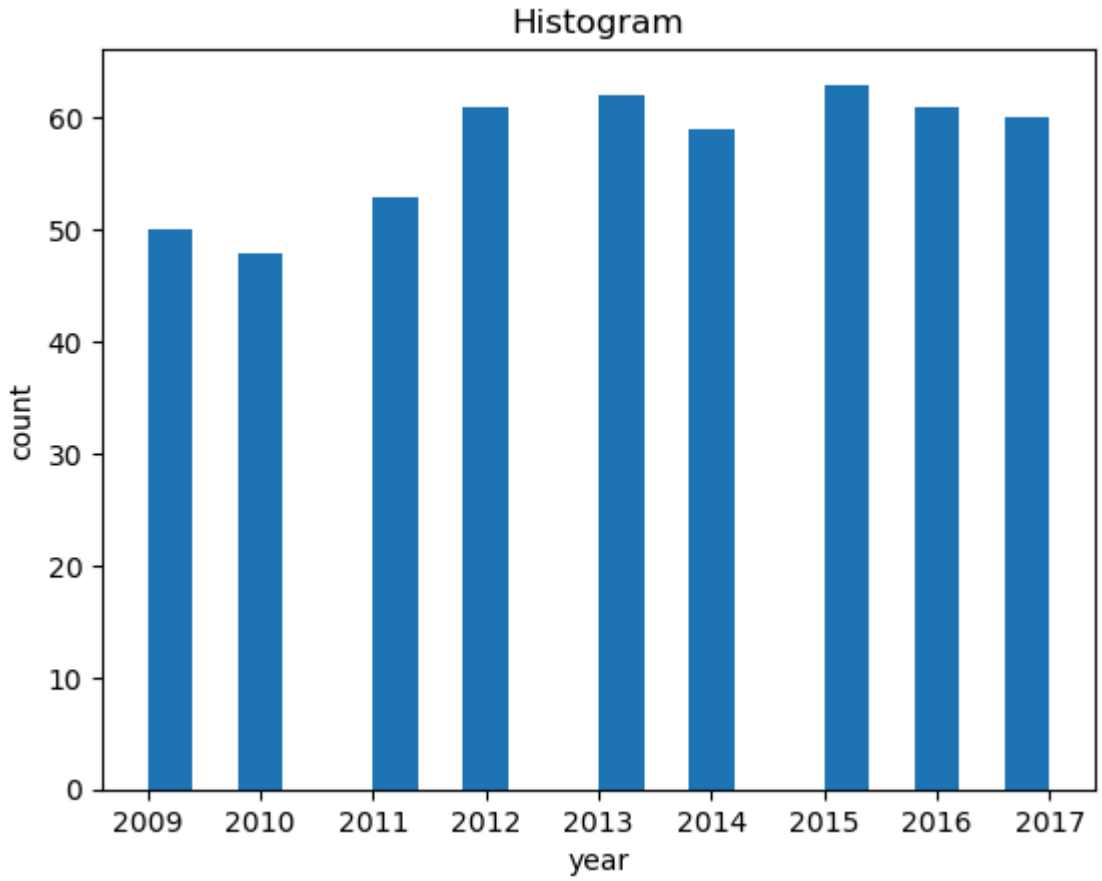
In [25]:
```
for i in num_cols:
    year_data=df[i]
    count,bins,x=plt.hist(year_data,bins=20)
```

```
    plt.xlabel(i)
    plt.ylabel('count')
    plt.title('Histogram')
    print(len(count))
    print(len(bins))
    print(x)
    plt.show()
```

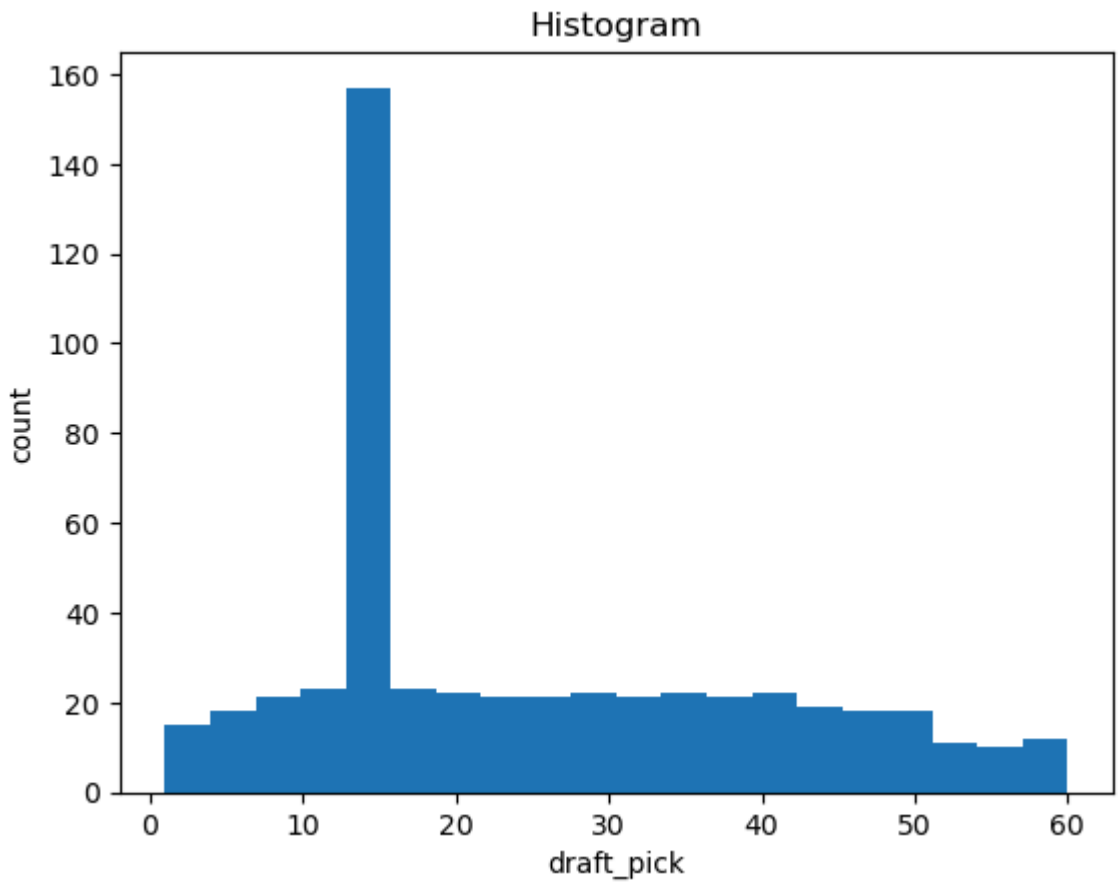
```
20
21
<BarContainer object of 20 artists>
```



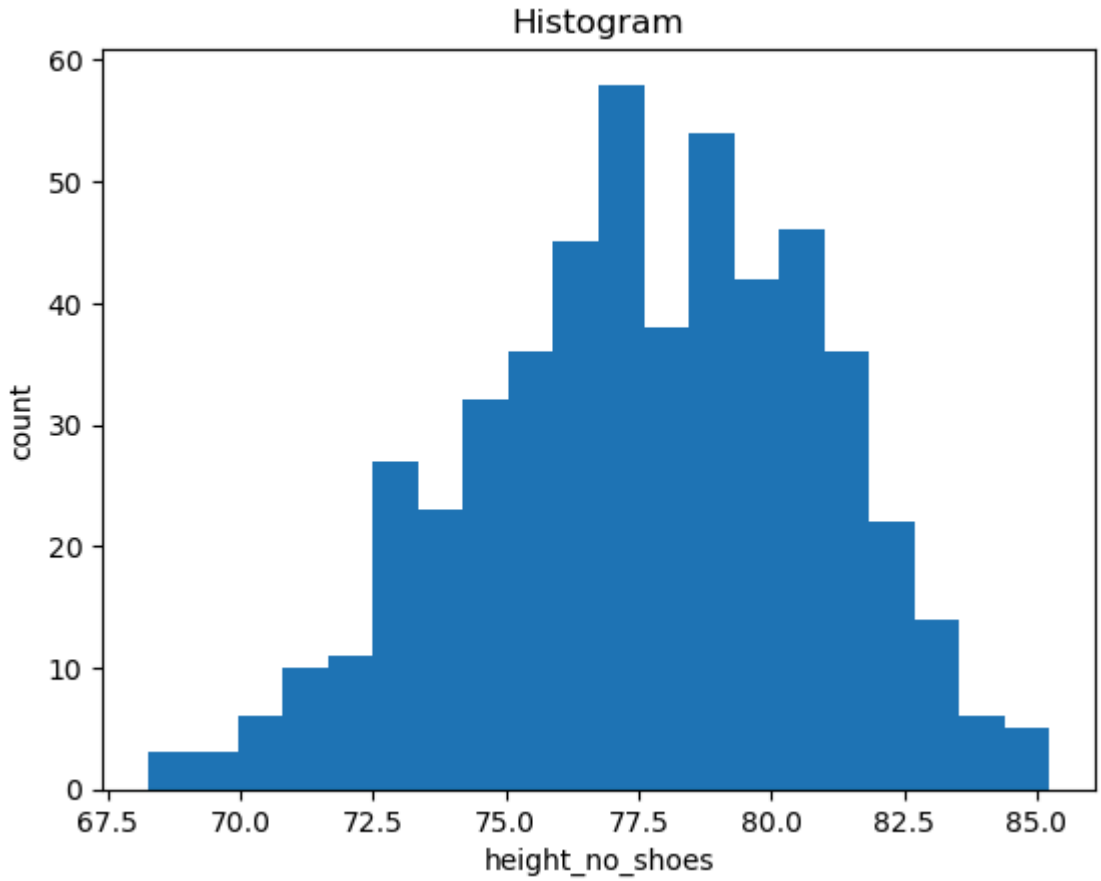

```
20
21
<BarContainer object of 20 artists>
```



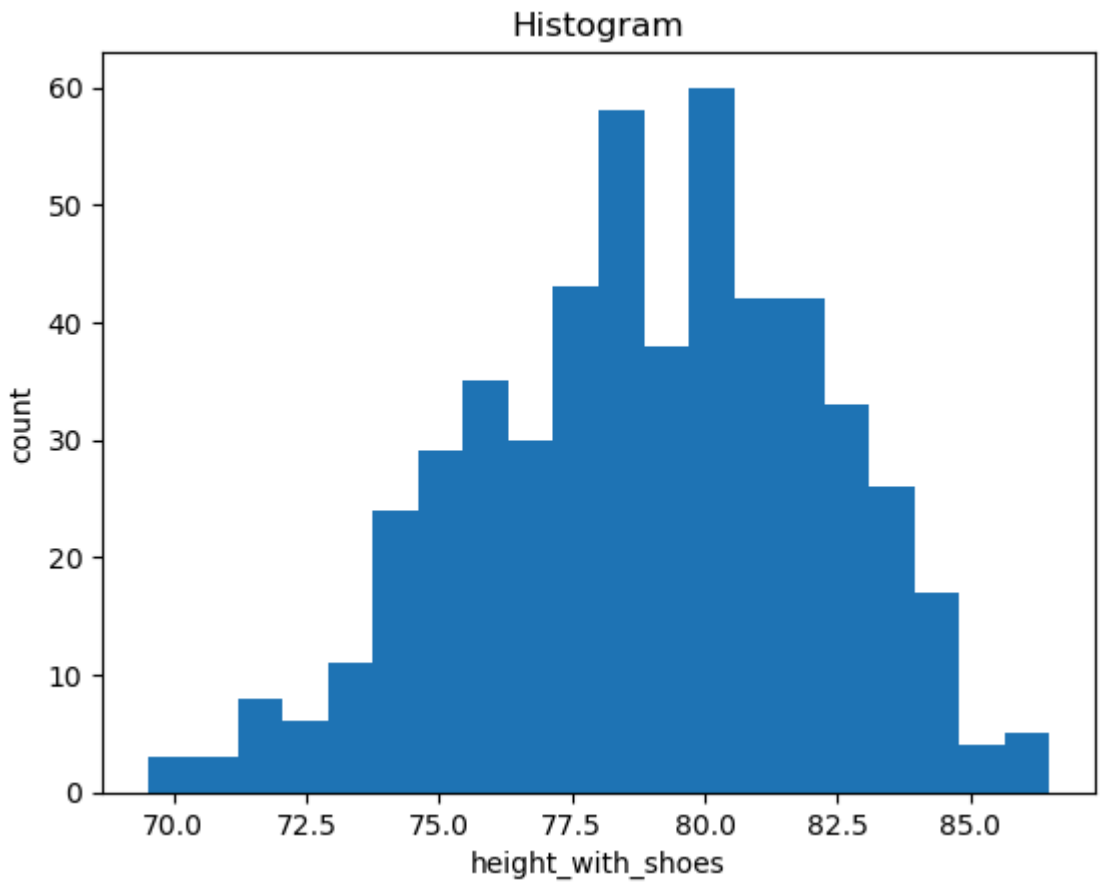

```
20
21
<BarContainer object of 20 artists>
```



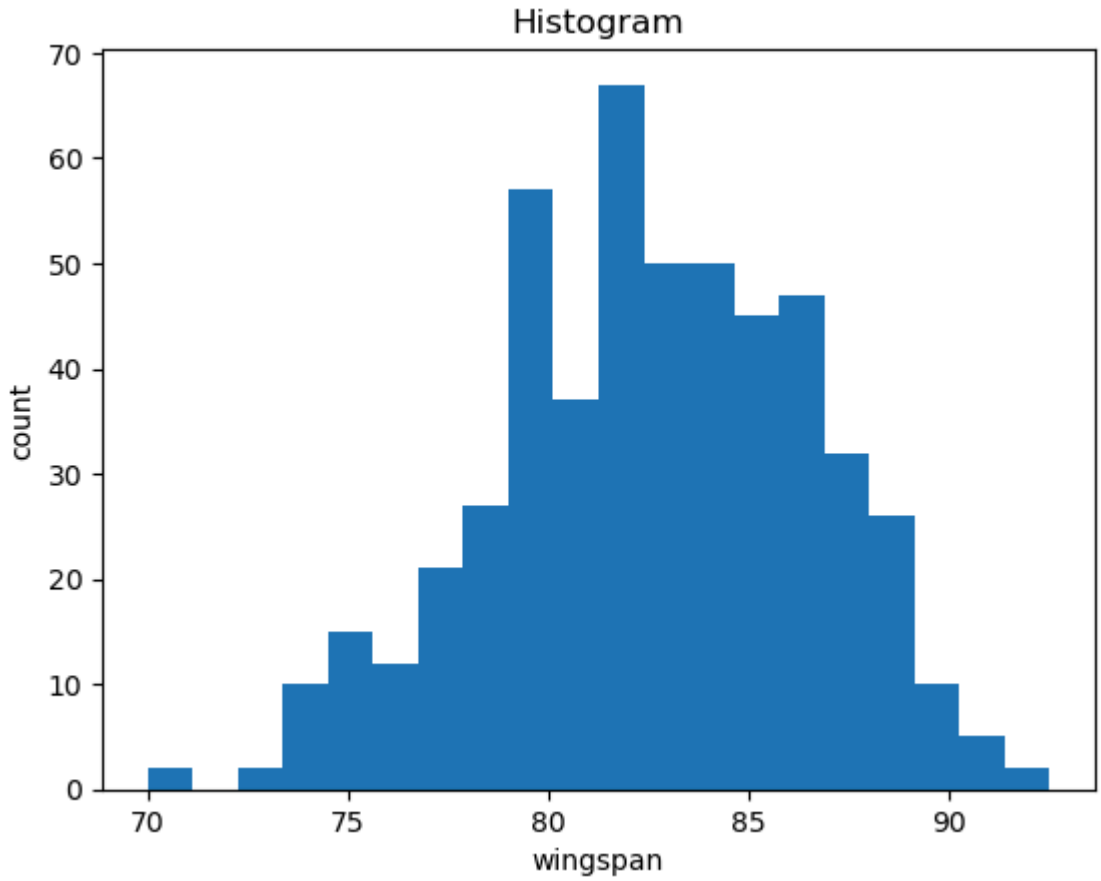

```
20
21
<BarContainer object of 20 artists>
```



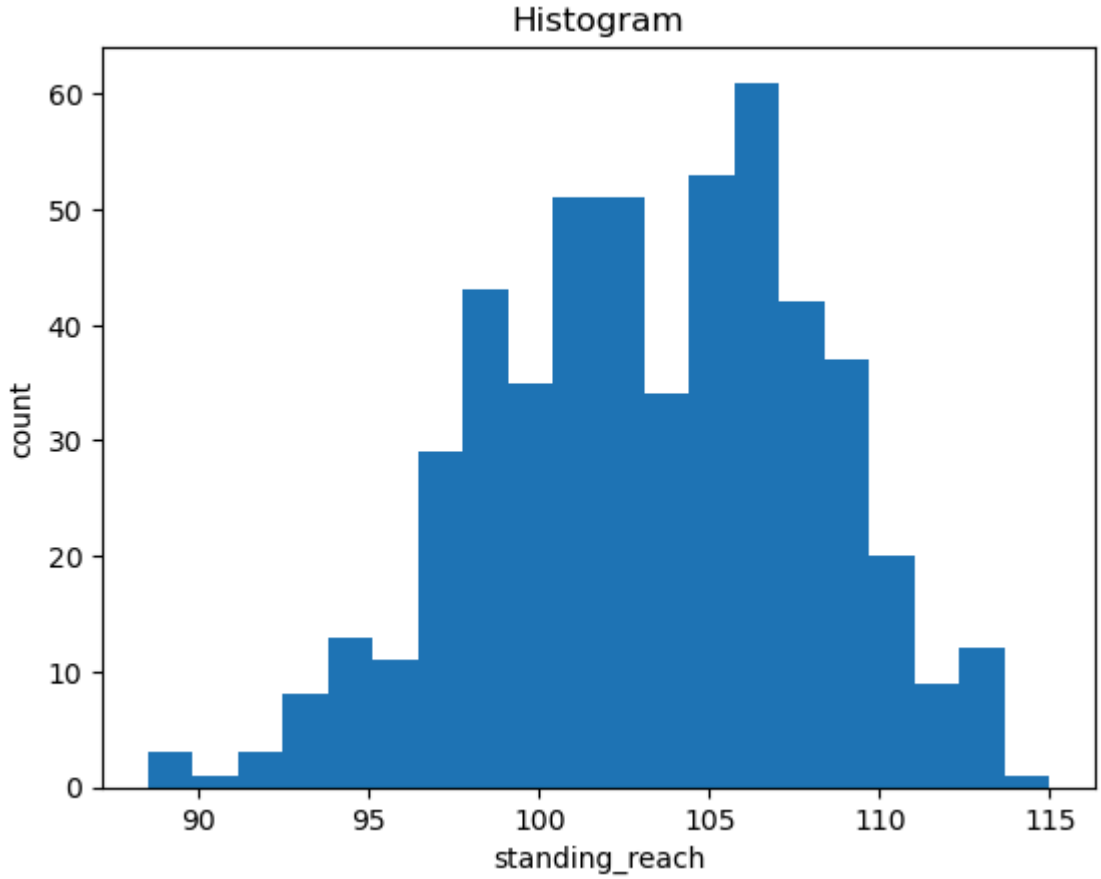

```
20
21
<BarContainer object of 20 artists>
```



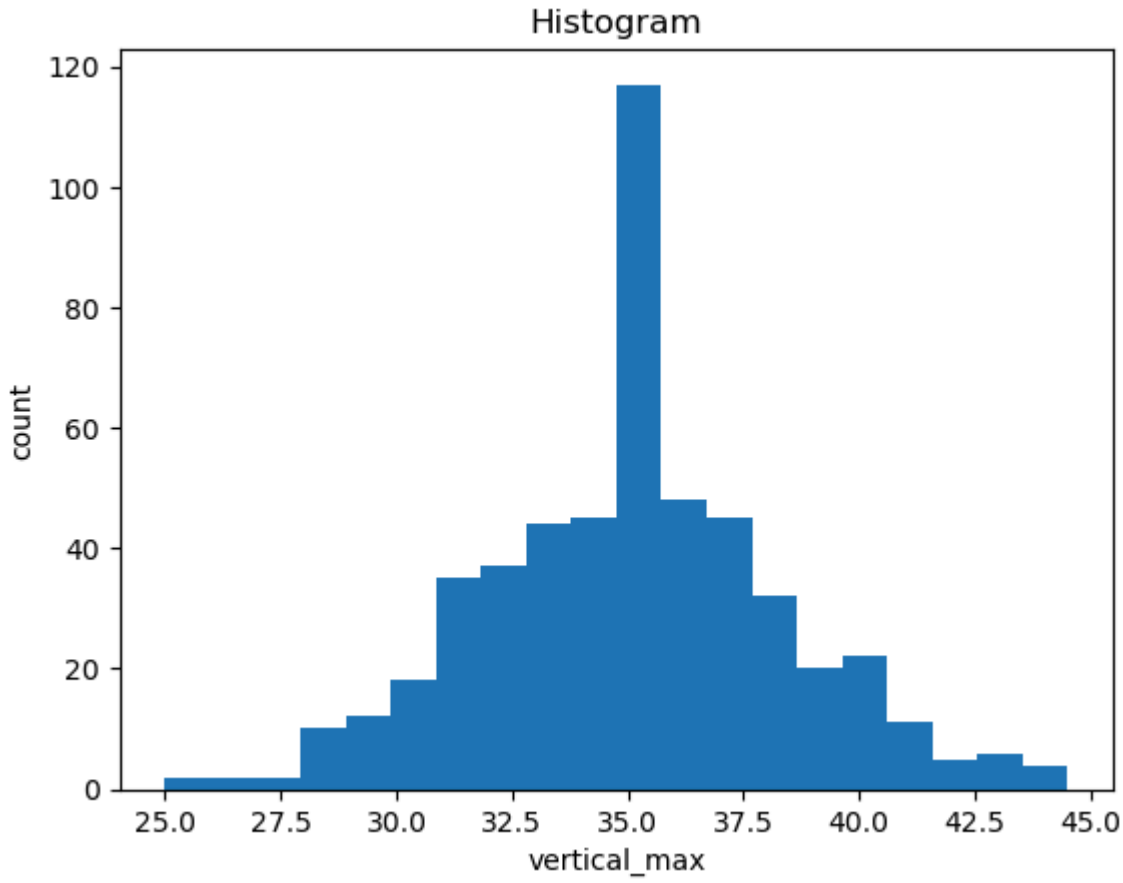

```
20
21
<BarContainer object of 20 artists>
```



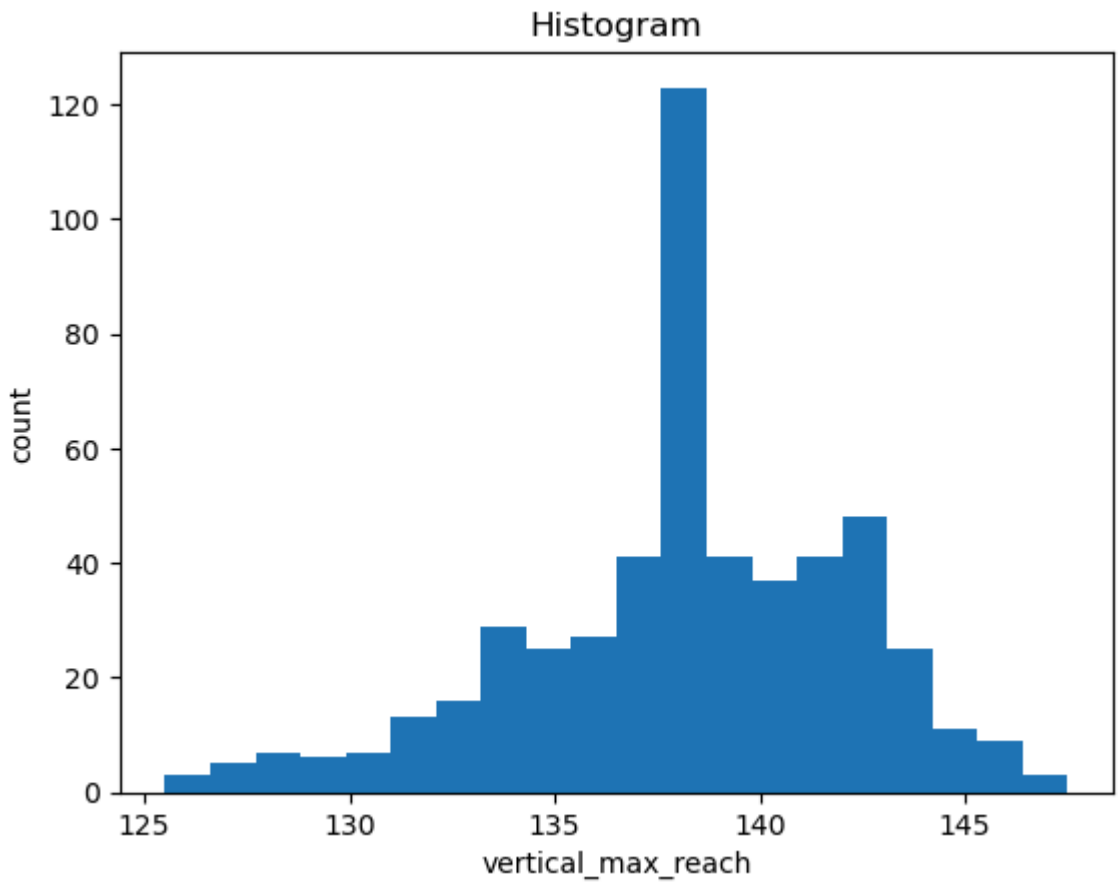

```
20
21
<BarContainer object of 20 artists>
```



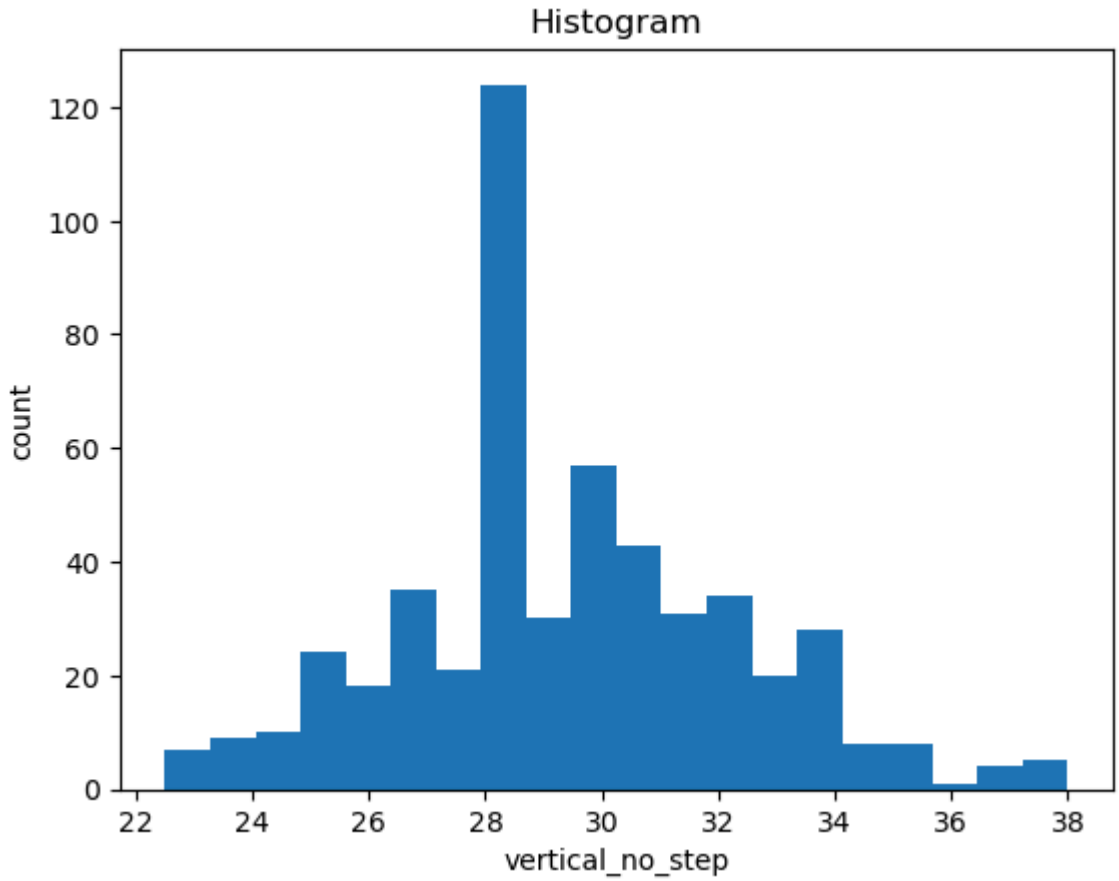

```
20
21
<BarContainer object of 20 artists>
```



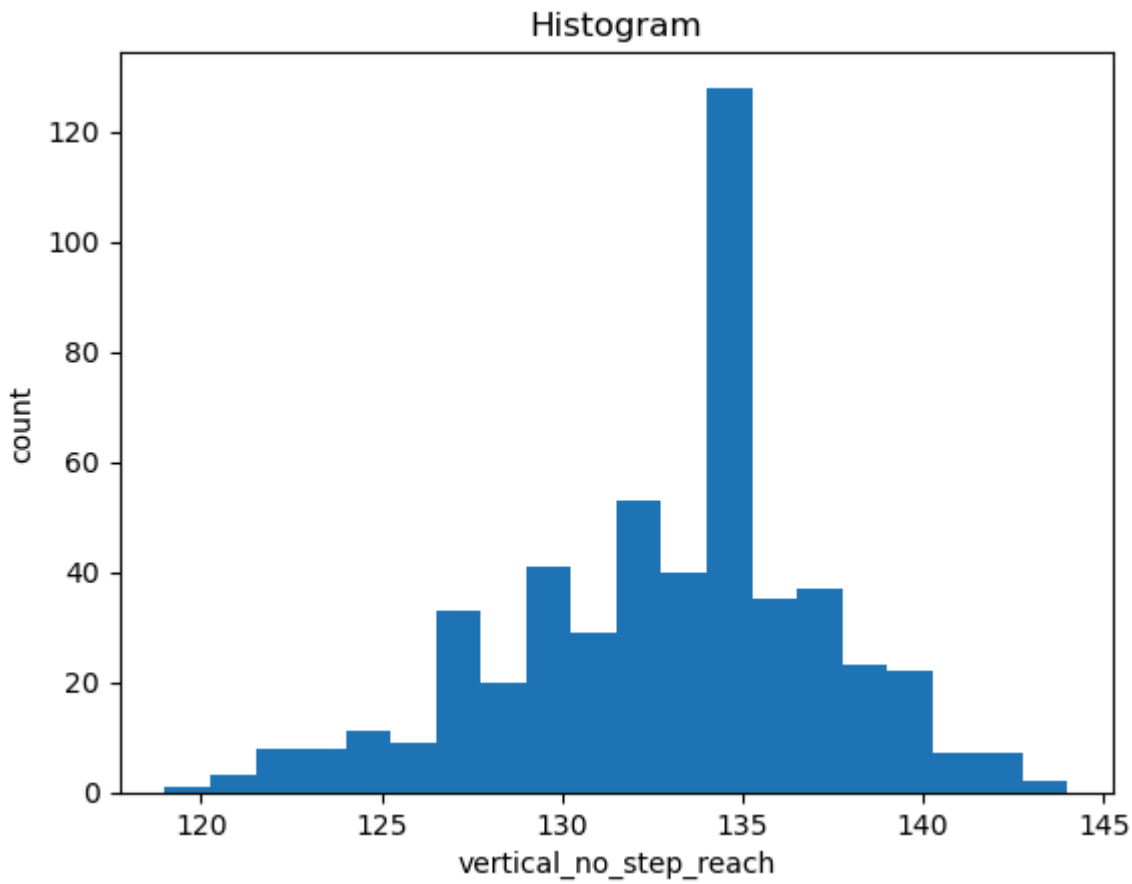

```
20
21
<BarContainer object of 20 artists>
```



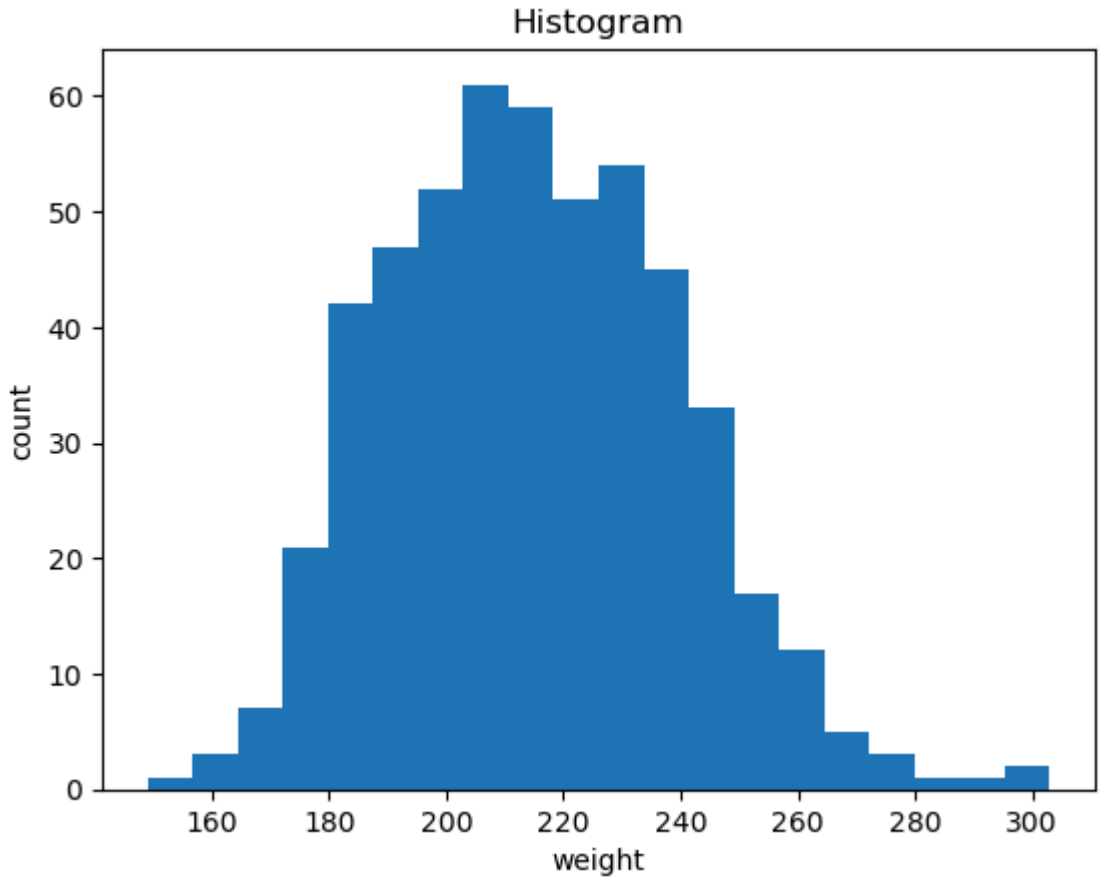

```
20
21
<BarContainer object of 20 artists>
```



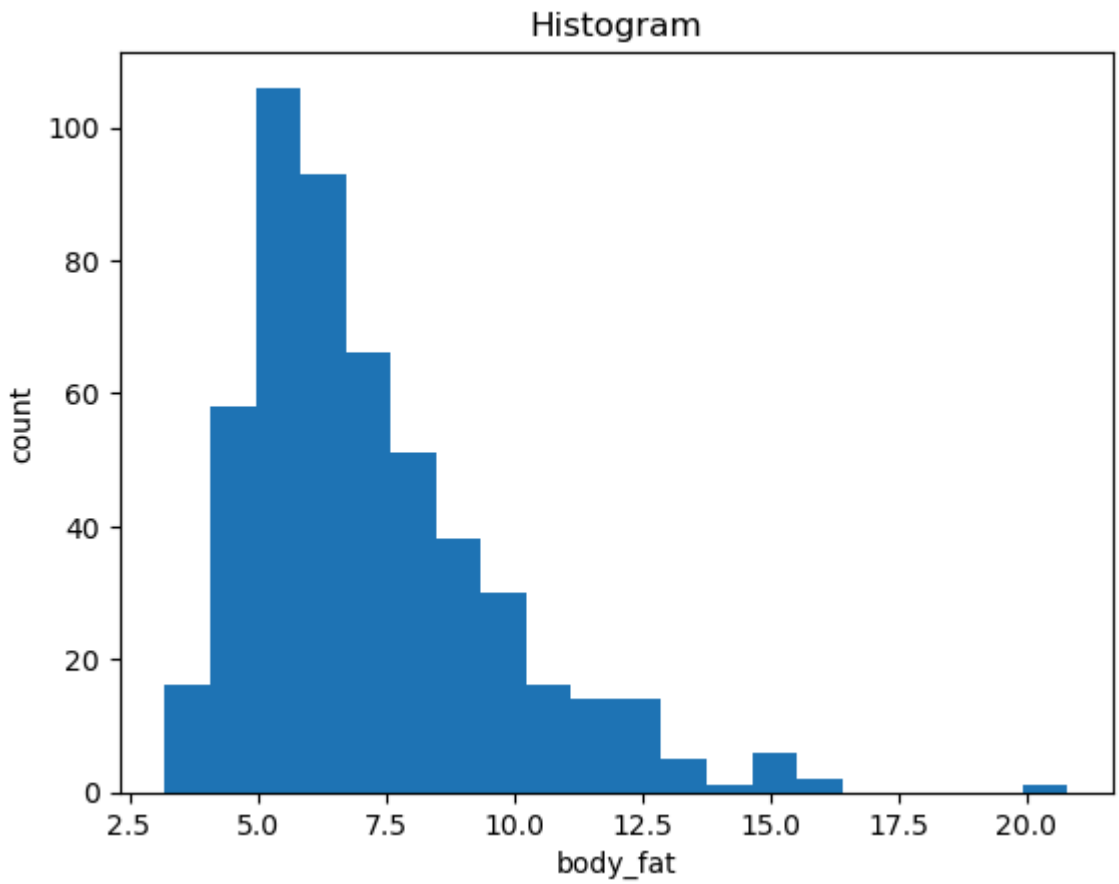

```
20
21
<BarContainer object of 20 artists>
```



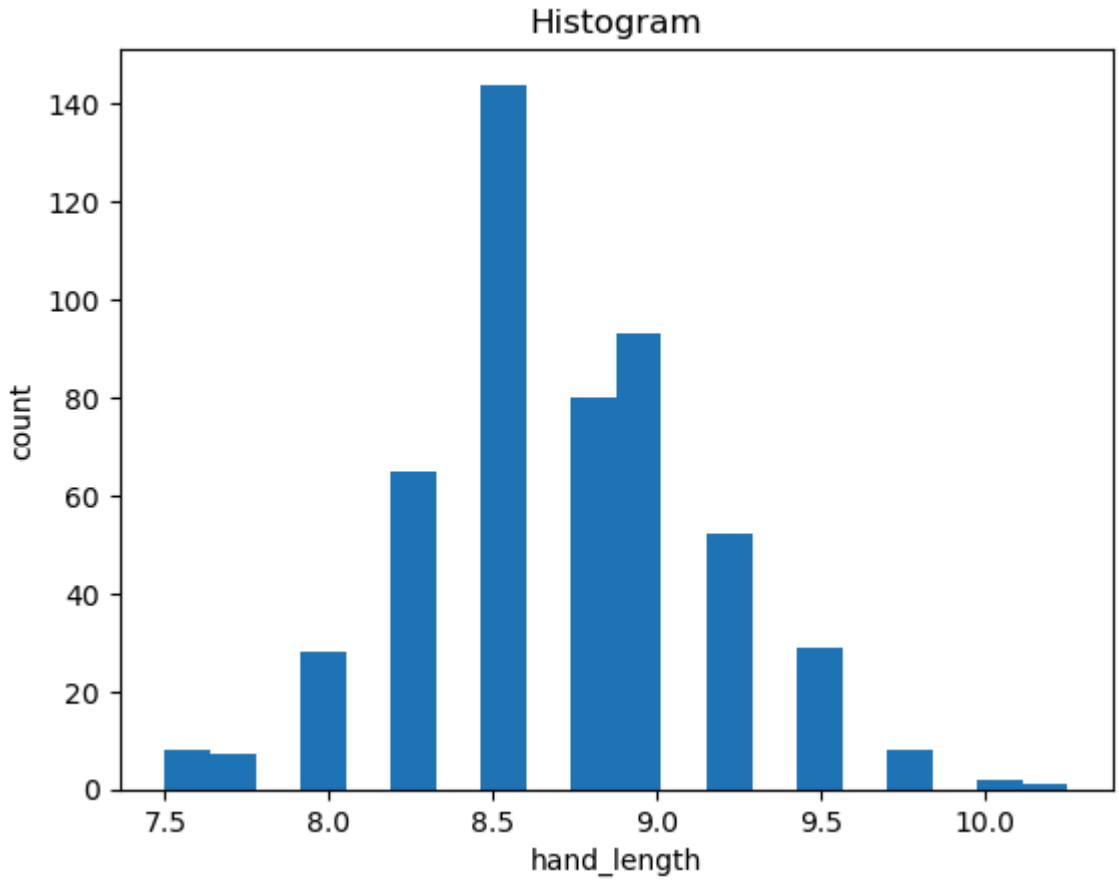

```
20
21
<BarContainer object of 20 artists>
```



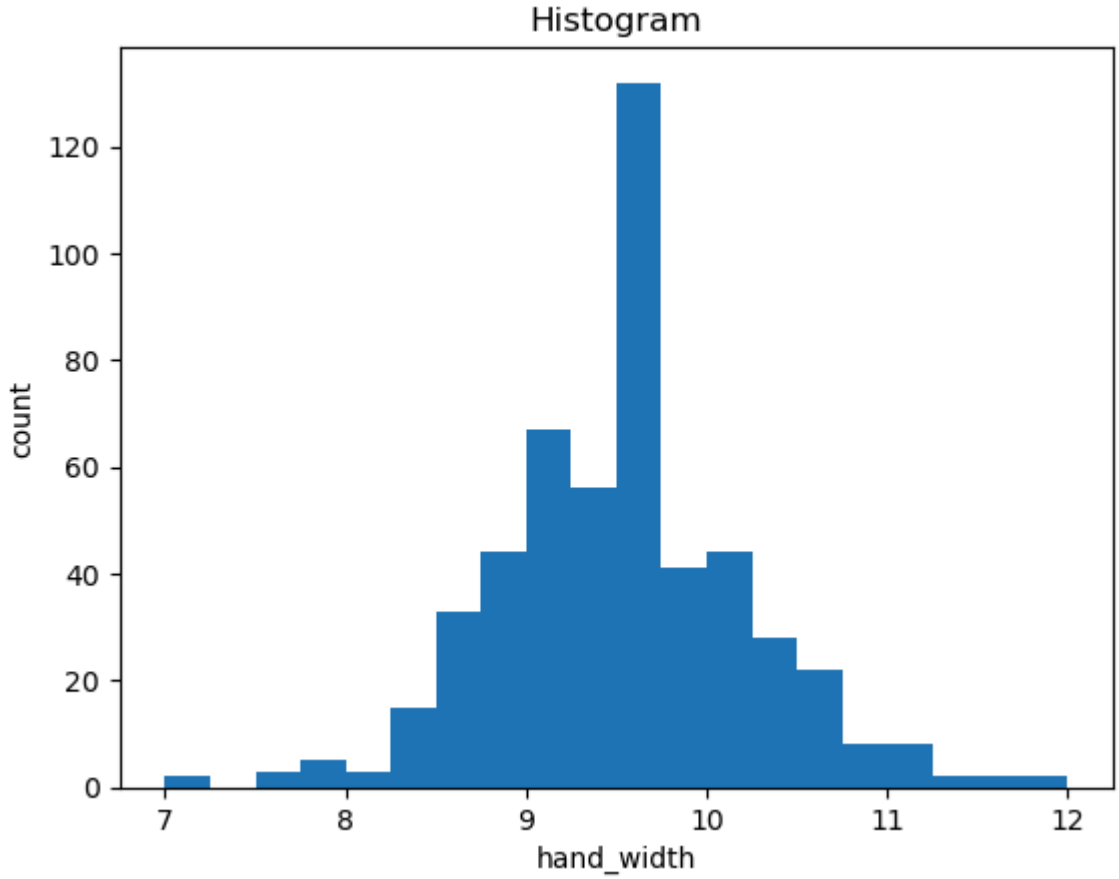

```
20
21
<BarContainer object of 20 artists>
```



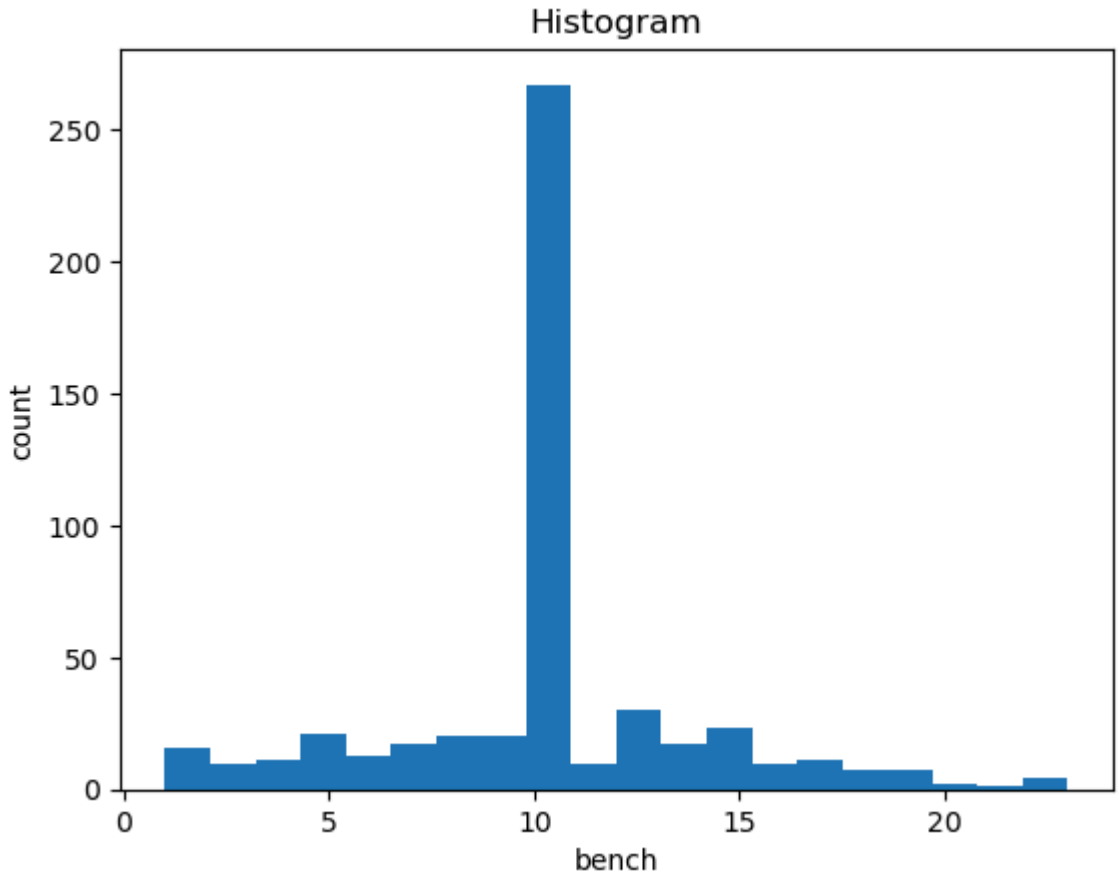

```
20
21
<BarContainer object of 20 artists>
```



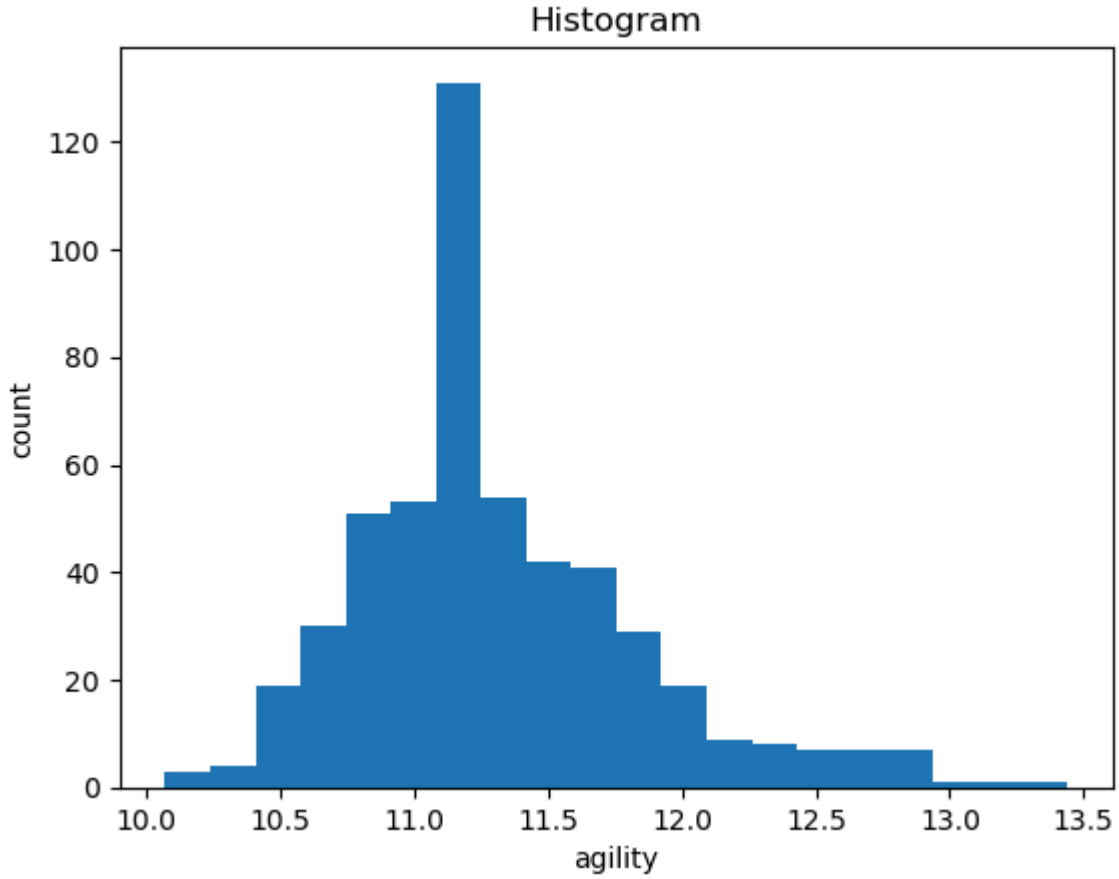

```
20
21
<BarContainer object of 20 artists>
```



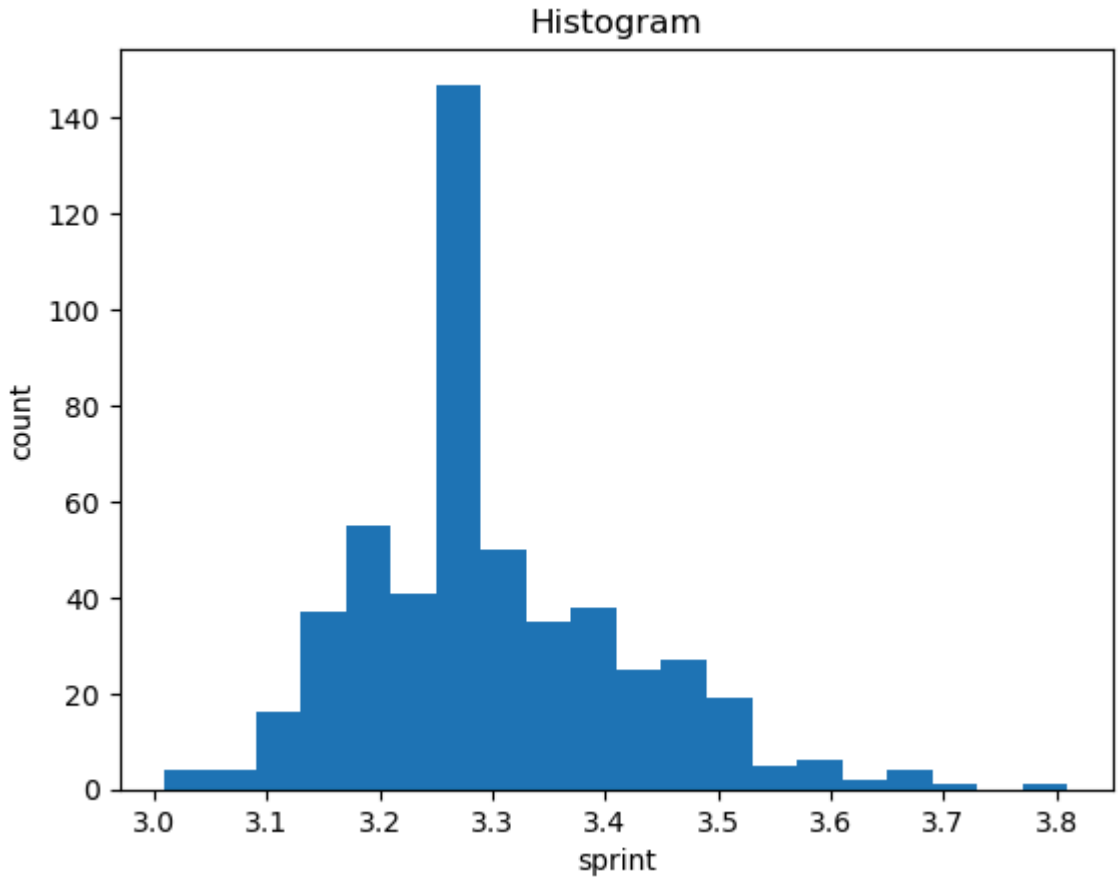

```
20
21
<BarContainer object of 20 artists>
```




```
20
21
<BarContainer object of 20 artists>
```

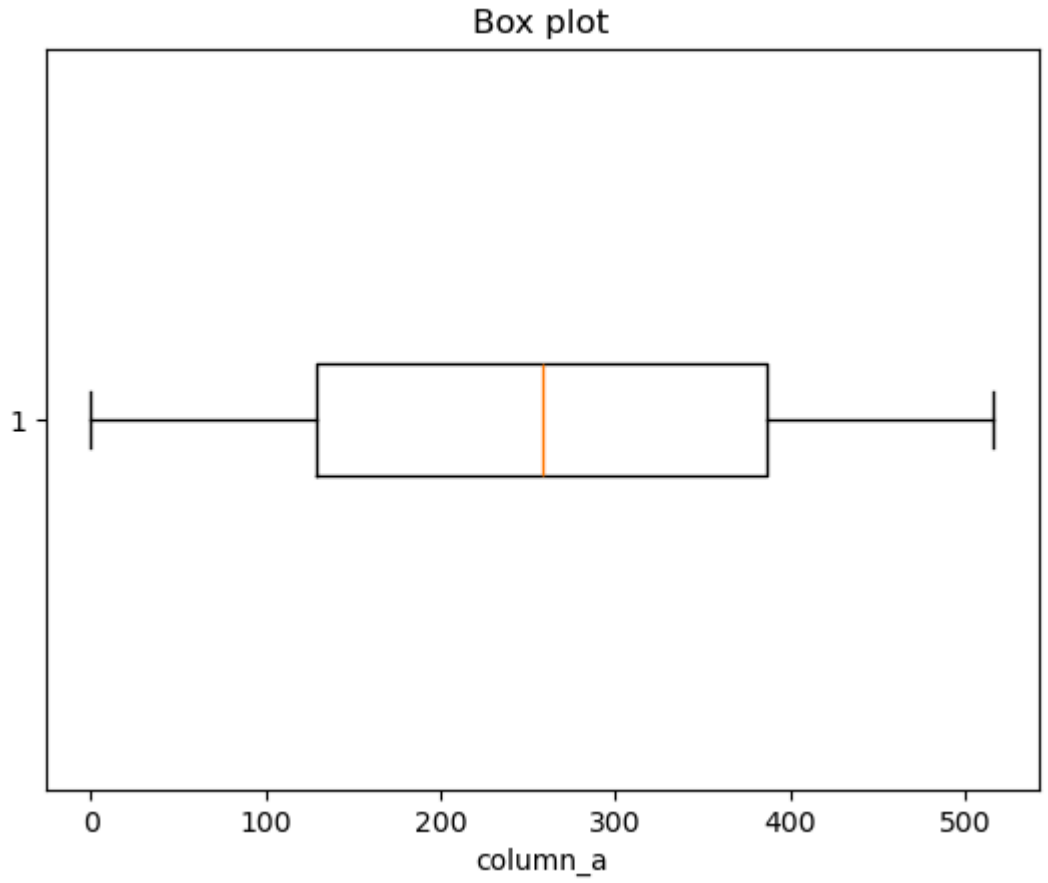


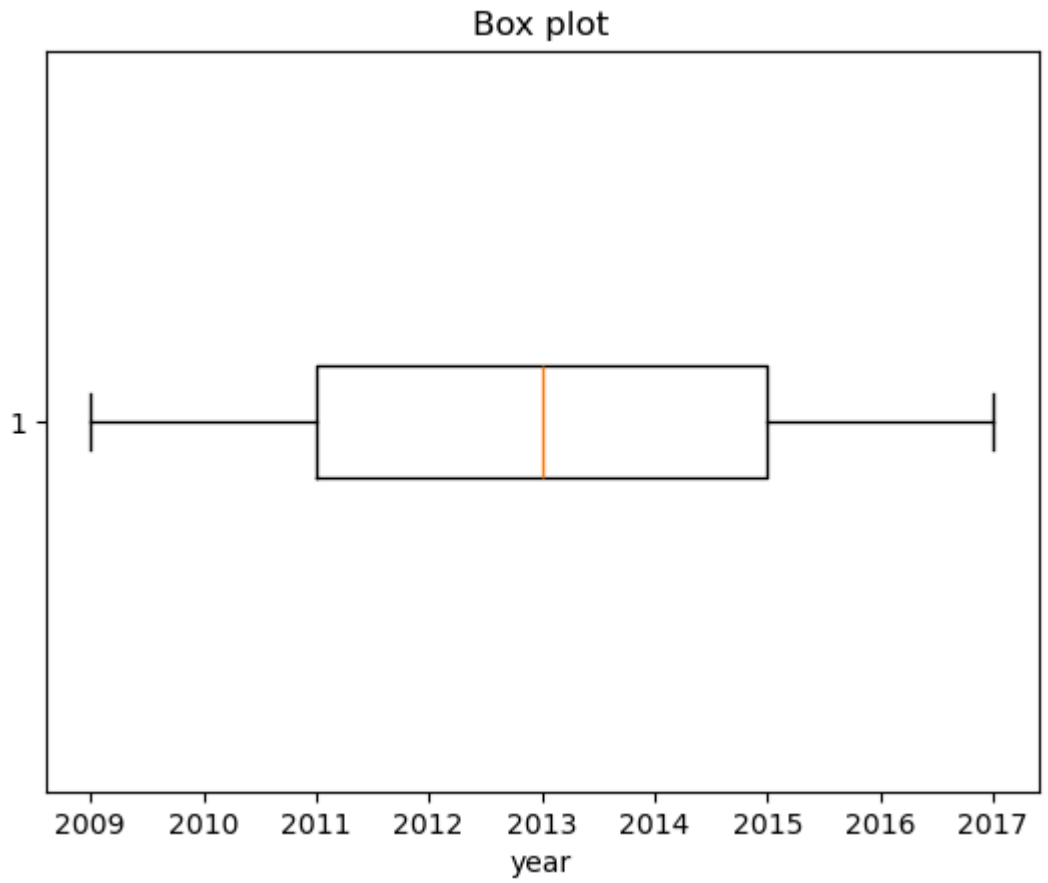

```
20
21
<BarContainer object of 20 artists>
```

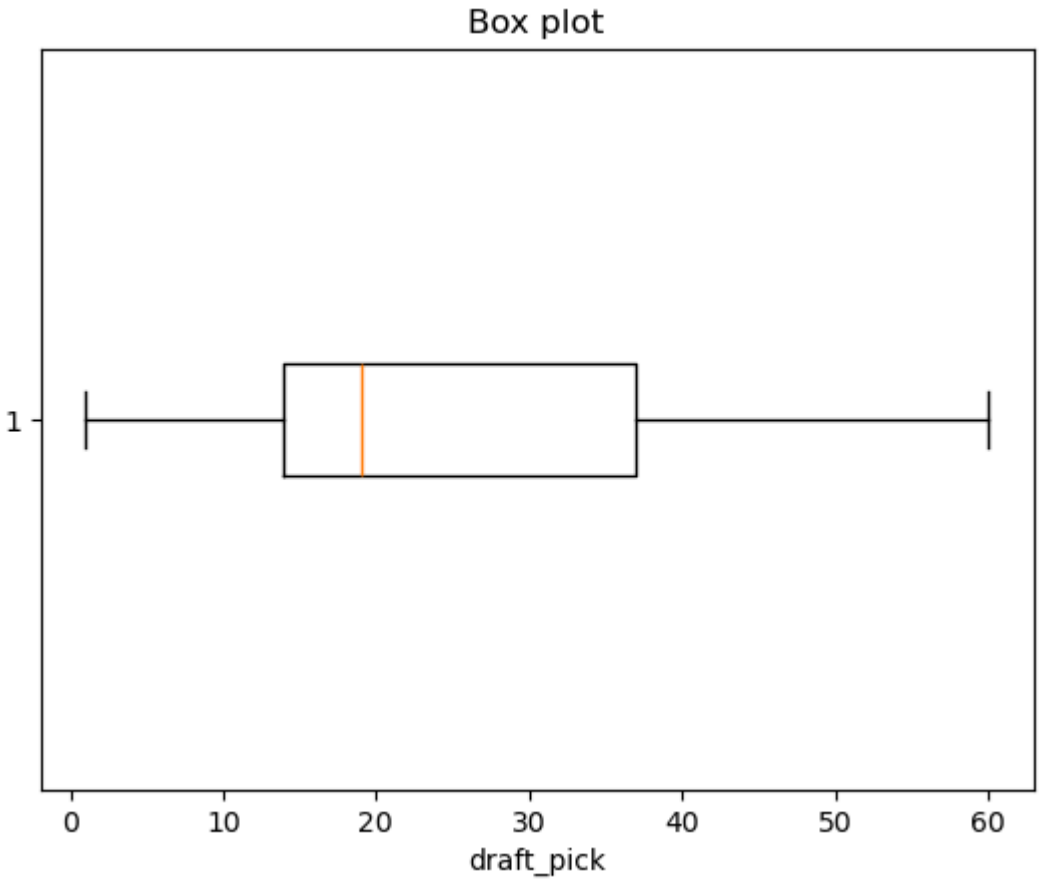


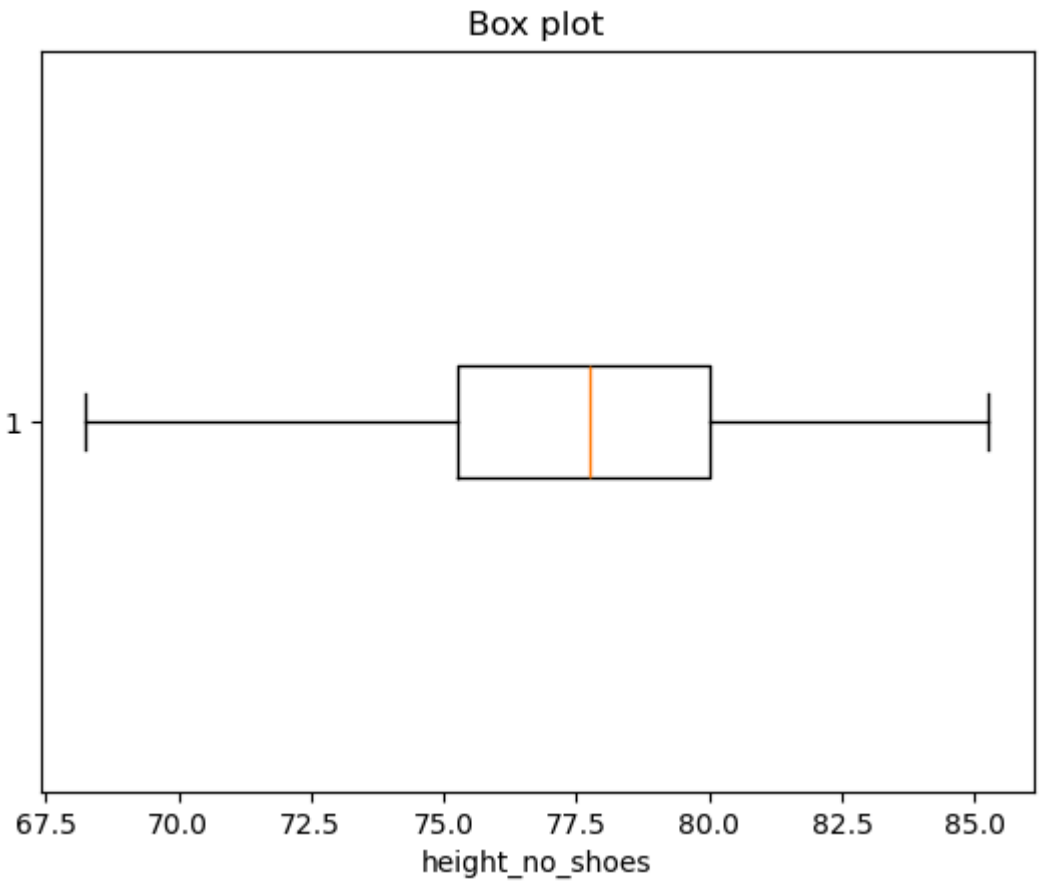
Box plot creation Outlier analysis for numerical columns

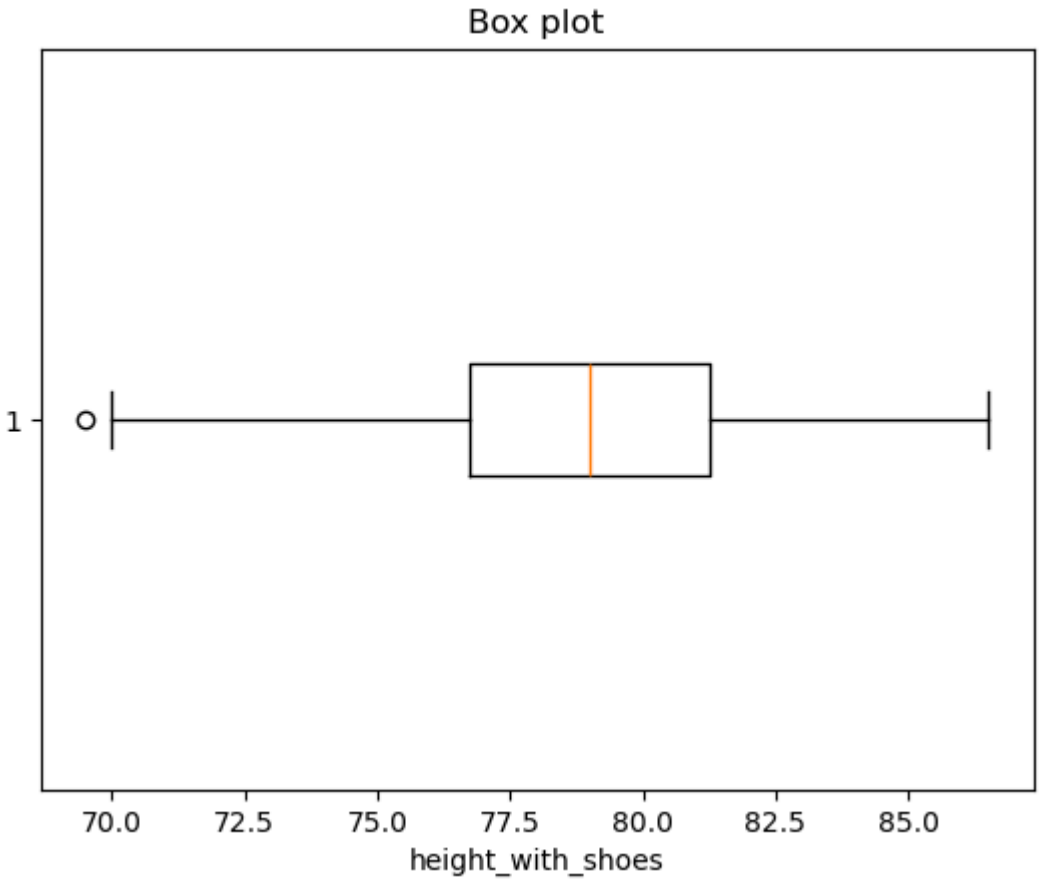
In [26]:
```
for i in num_cols:
    year_data=df[i]
    plt.boxplot(year_data,vert=False)
    plt.title("Box plot")
    plt.xlabel(i)
    plt.show()
```

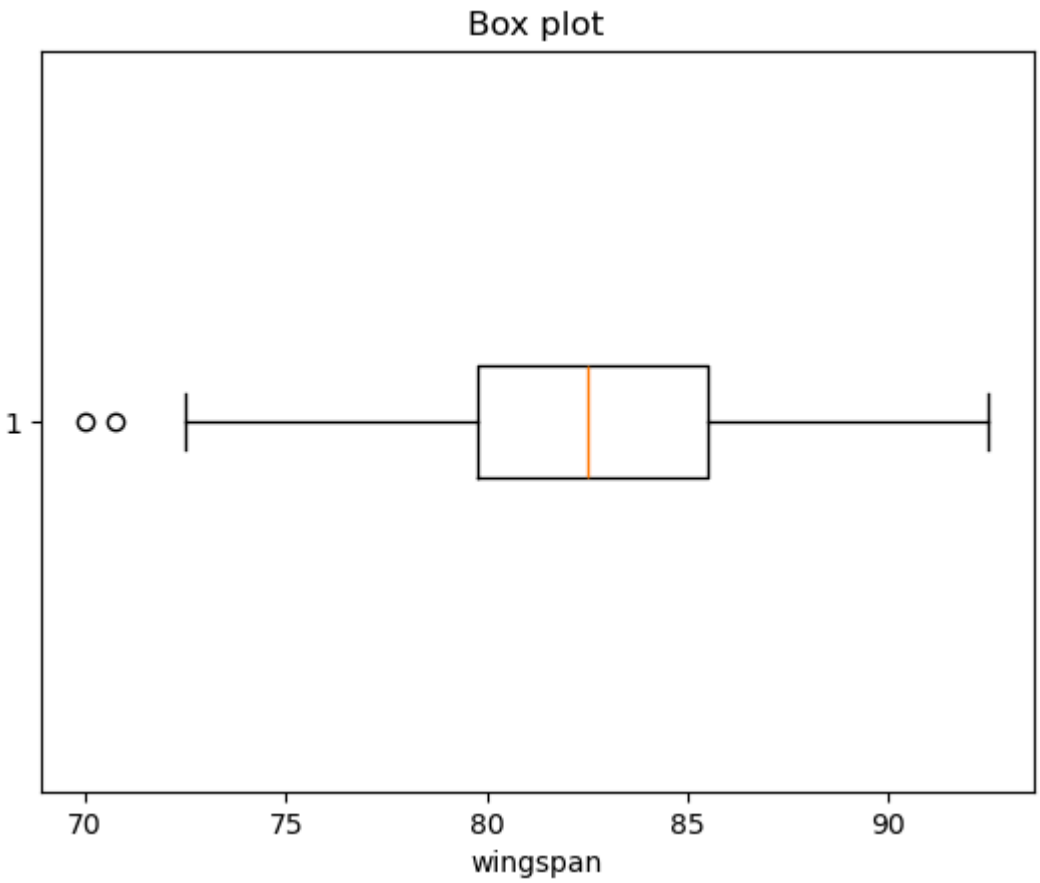

Box plot
1
0
100
200
300
400
500
column_a

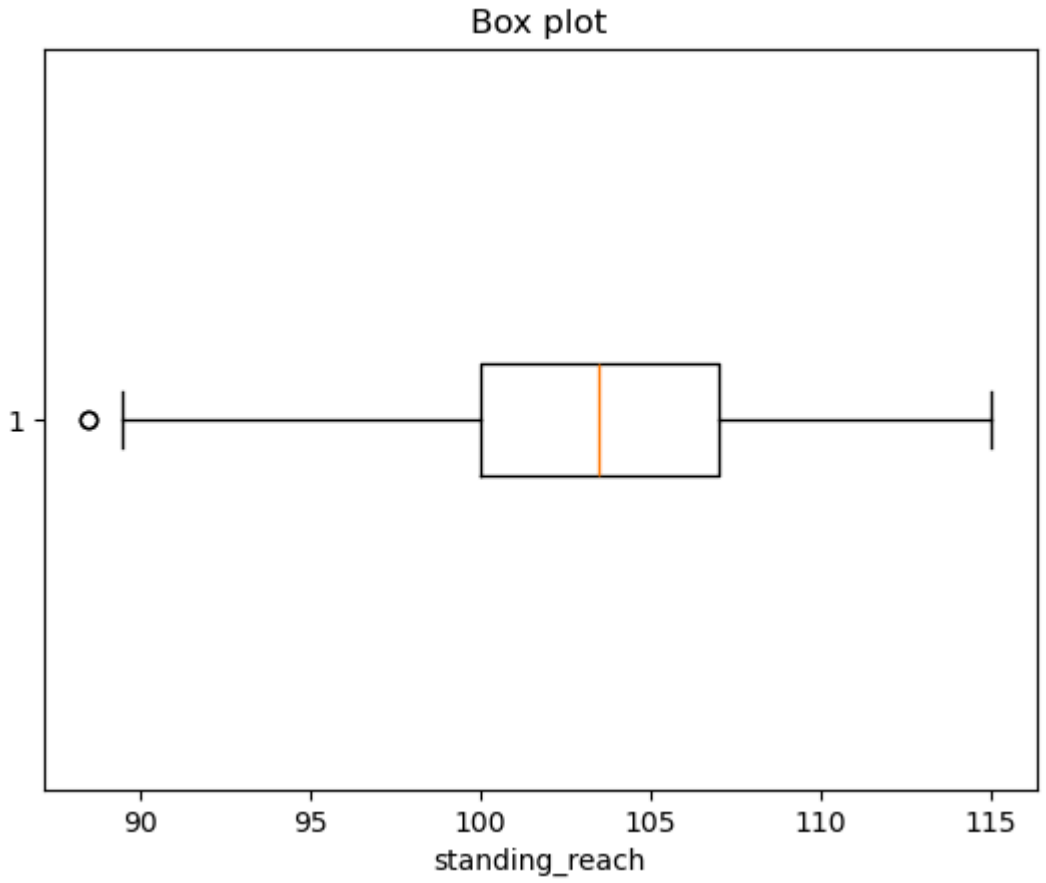

Box plot
1
2009
2010
2011
2012
2013
2014
2015
2016
2017
year

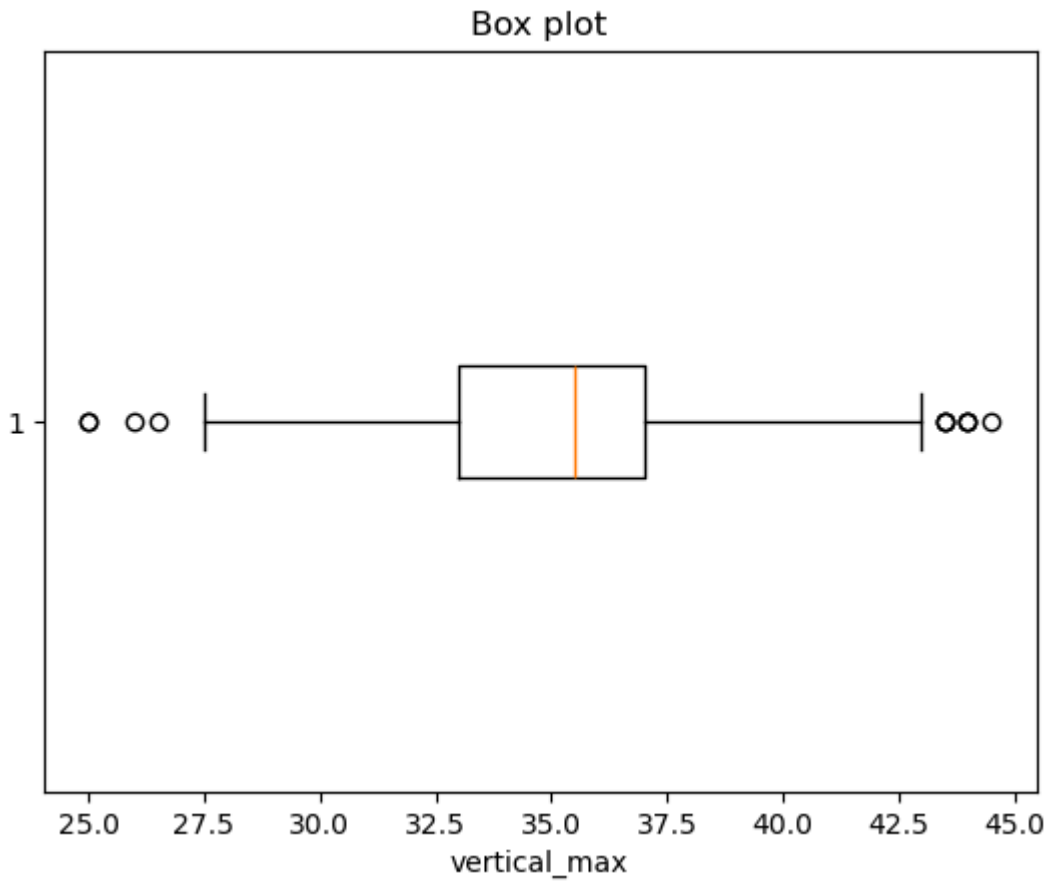

Box plot
1
0
10
20
30
40
50
60
draft_pick

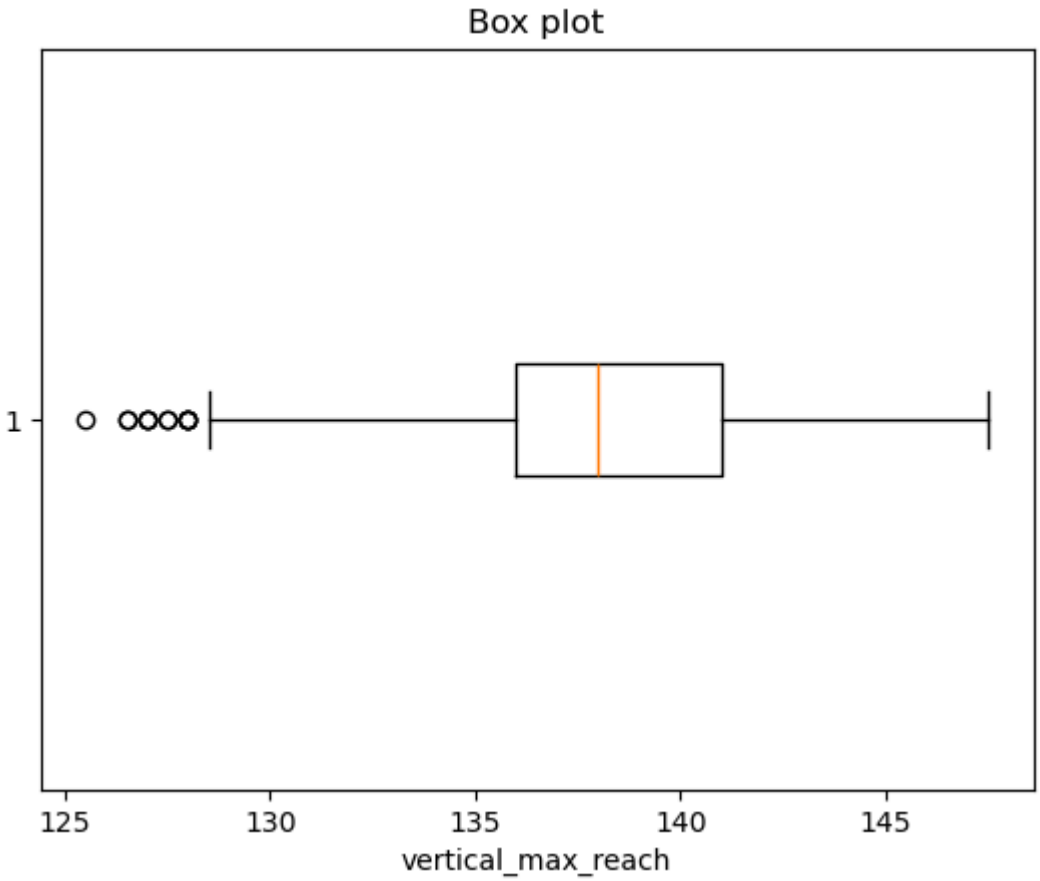

Box plot
1
67.5
70.0
72.5
75.0
77.5
80.0
82.5
85.0
height_no_shoes

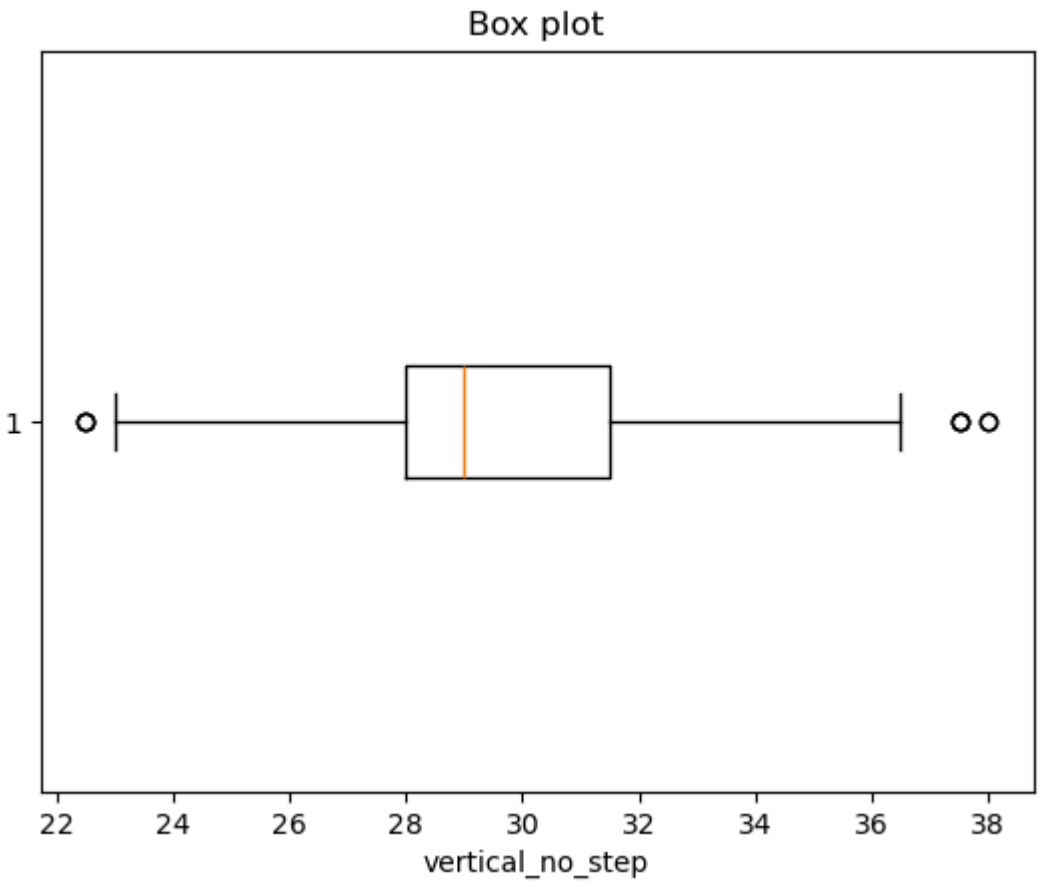

Box plot
1
70.0
72.5
75.0
77.5
80.0
82.5
85.0
height_with_shoes

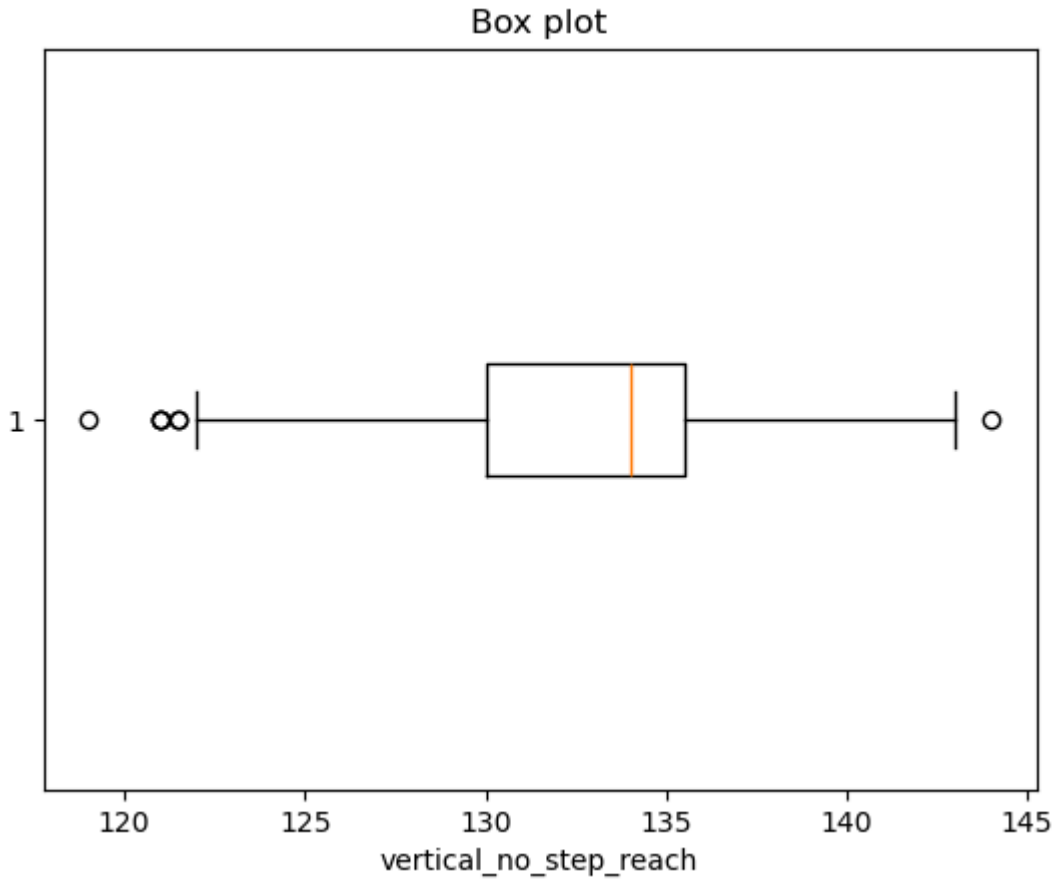

Box plot
1
70
75
80
85
90
wingspan

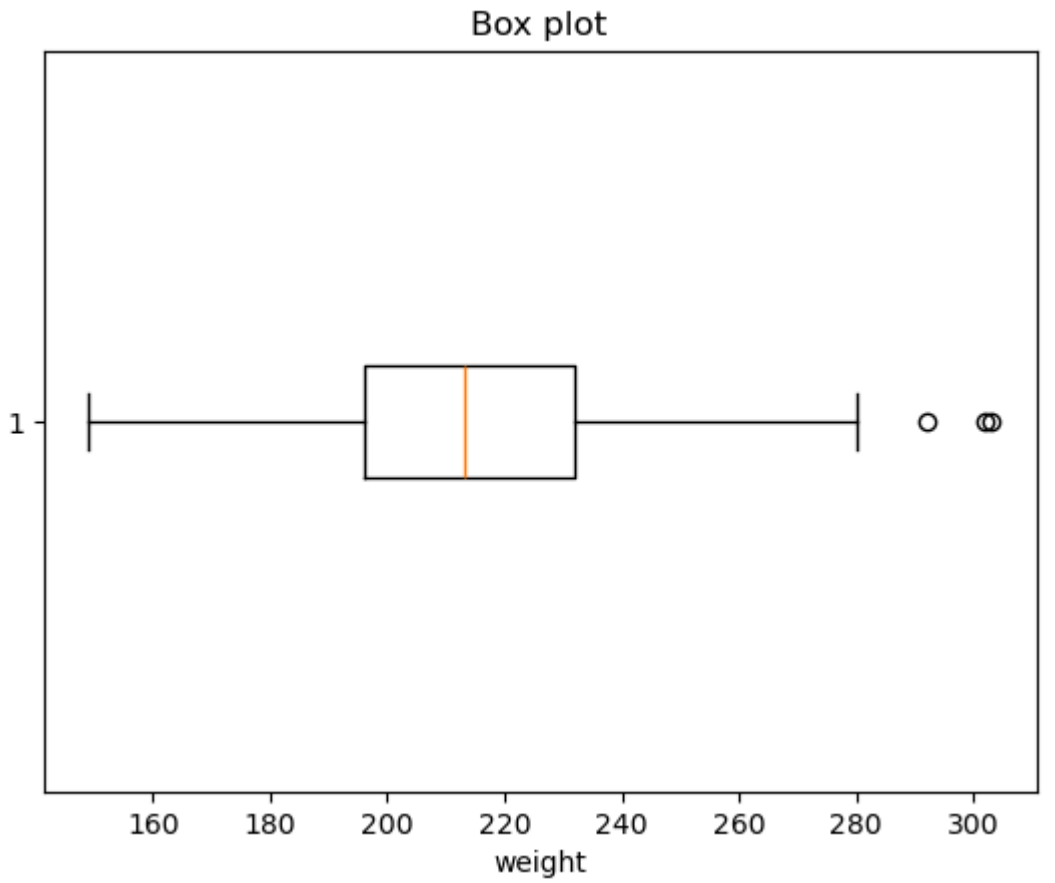

Box plot
1
90
95
100
105
110
115
standing_reach

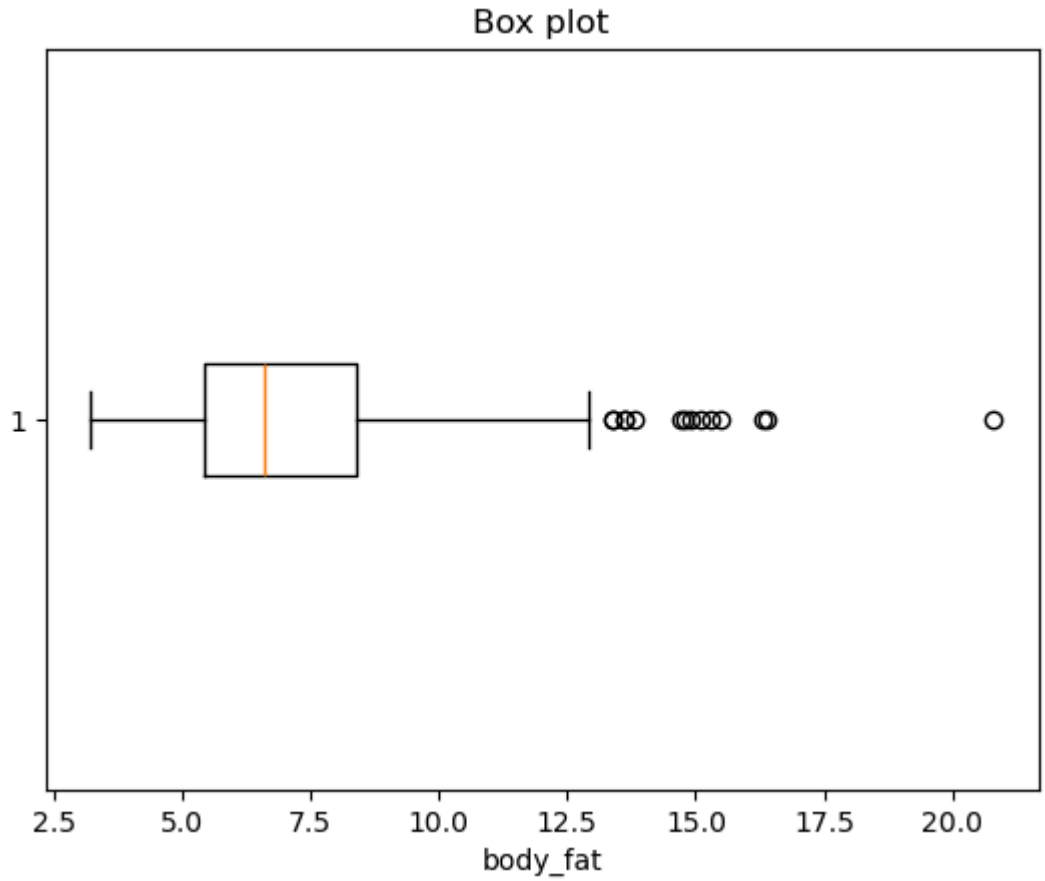

Box plot
1
25.0
27.5
30.0
32.5
35.0
37.5
40.0
42.5
45.0
vertical_max

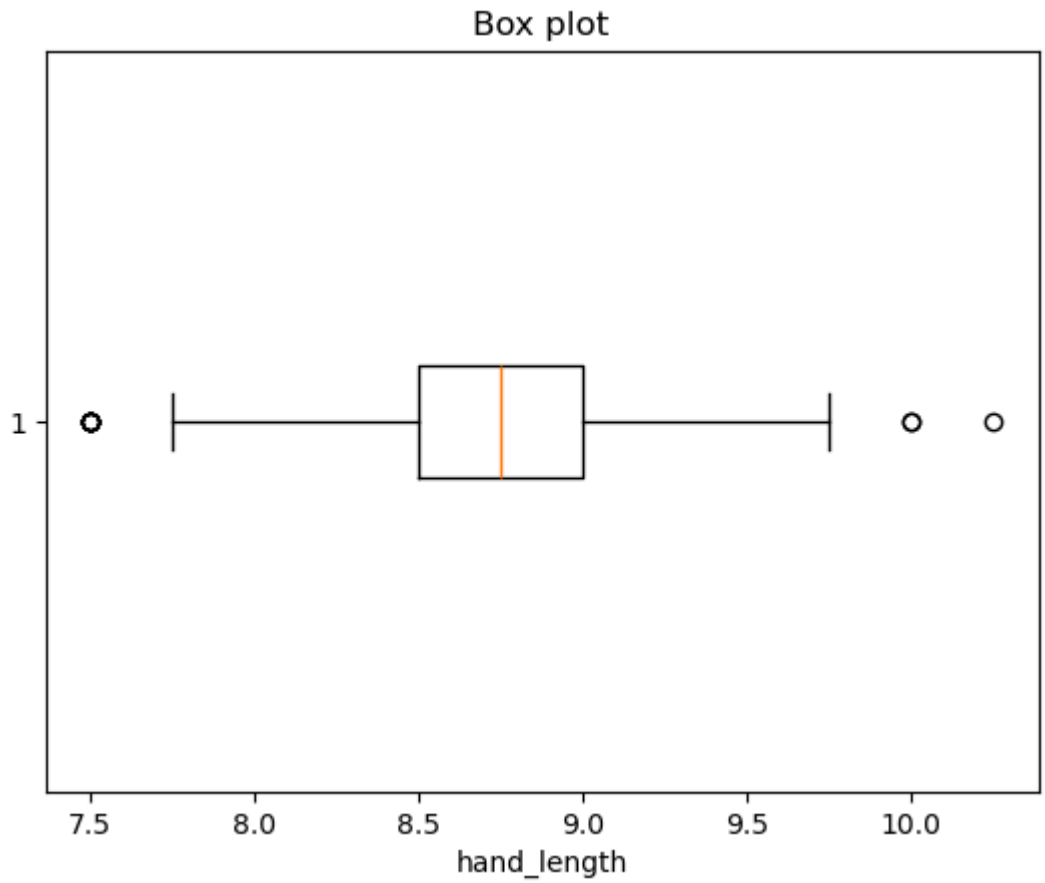

Box plot
1
125
130
135
140
145
vertical_max_reach

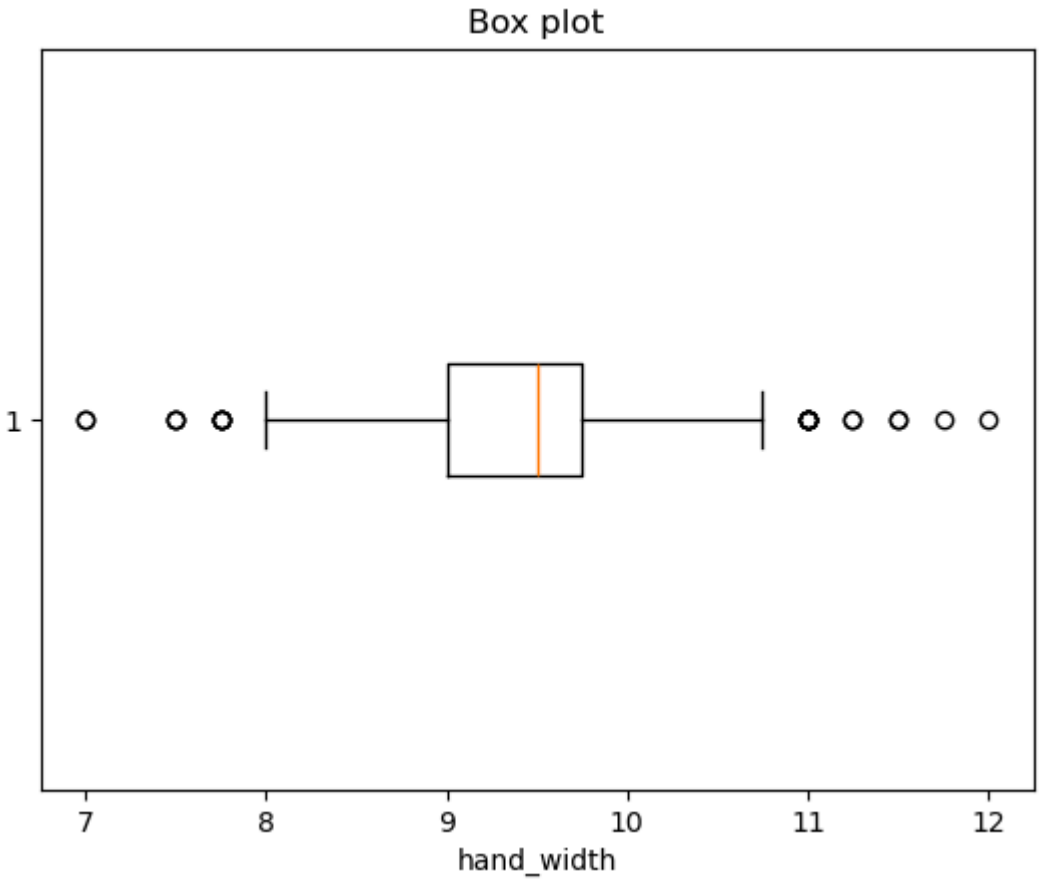

Box plot
1
22
24
26
28
30
32
34
36
38
vertical_no_step

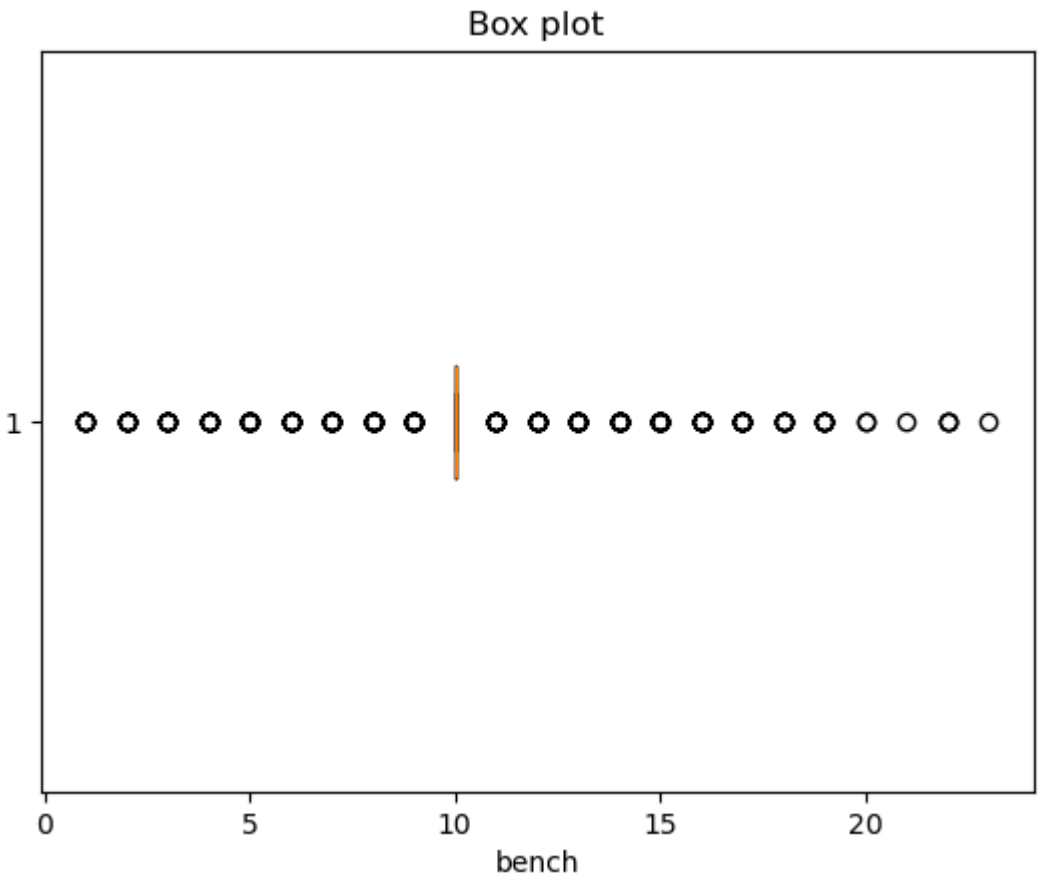

Box plot
1
120
125
130
135
140
145
vertical_no_step_reach

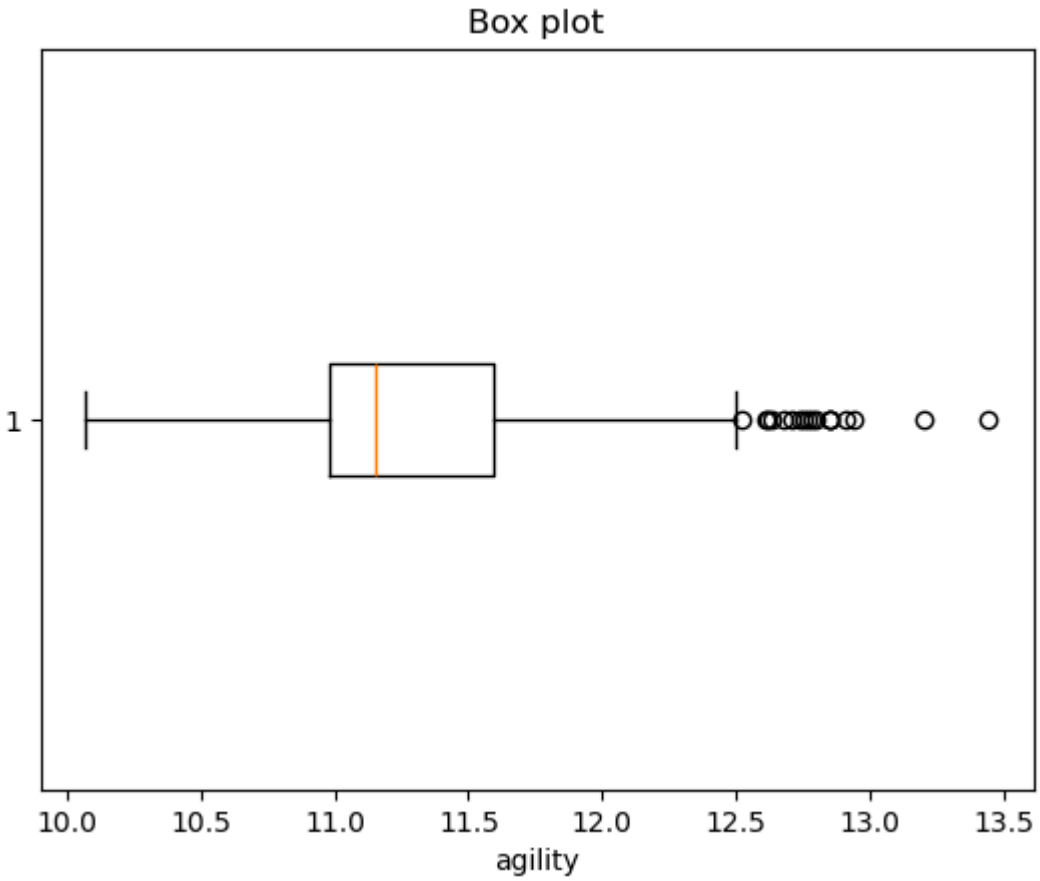

Box plot
1
160
180
200
220
240
260
280
300
weight

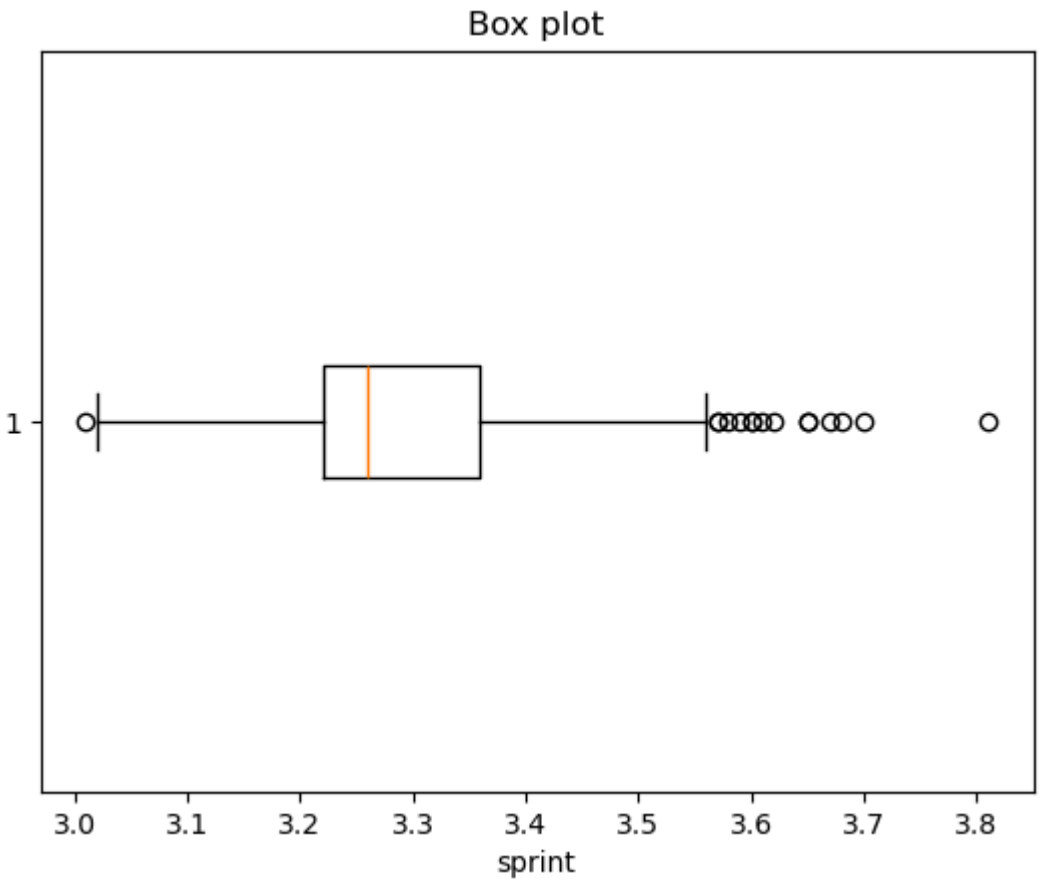

Box plot
1
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
body_fat


Box plot
1
7.5
8.0
8.5
9.0
9.5
10.0
hand_length


Box plot
1
7
8
9
10
11
12
hand_width


Box plot
1
0
5
10
15
20
bench





Removing outliers from the data

removing outliers refres that non-outliers data

In [27]:
```
# by using IQR we are able to remove outliers in data
height_data=df['height_no_shoes']
q1=np.percentile(height_data,25)
q2=np.percentile(height_data,50)
q3=np.percentile(height_data,75)

IQR=q3-q1

lb=(q1-(1.5*IQR))
ub=(q3+(1.5*IQR))

con1=height_data>lb
con2=height_data<ub
con3=con1&con2
non_outliers_data=height_data[con3]
non_outliers_data
```

Out[27]:
```
0      80.50
1      77.00
2      76.00
3      80.25
4      80.50
       ...
512    76.25
513    74.50
514    78.50
515    83.50
516    78.25
Name: height_no_shoes, Length: 517, dtype: float64
```

In [28]:
```
# data frame without any outliers
non_outliers_df=df[con3]
non_outliers_df.dropna(inplace=True)
non_outliers_df
```

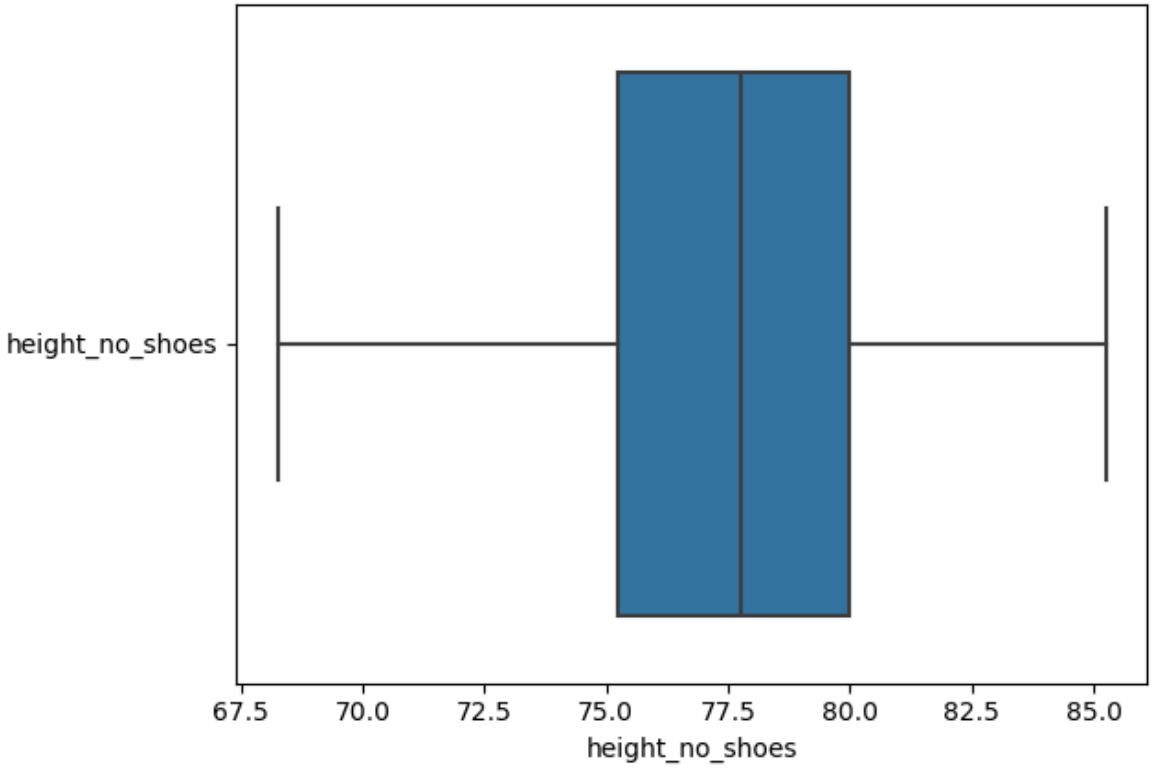
Out[28]:

	column_a	player	year	draft_pick	height_no_shoes	height_with_shoes	wing
0	0	Blake Griffin	2009	1.0	80.50	82.00	
1	1	Terrence Williams	2009	11.0	77.00	78.25	
2	2	Gerald Henderson	2009	12.0	76.00	77.00	
3	3	Tyler Hansbrough	2009	13.0	80.25	81.50	
4	4	Earl Clark	2009	14.0	80.50	82.25	
...	...	...	...	...	...	...	
512	512	Peter Jok	2017	14.0	76.25	77.75	
513	513	Rawle Alkins	2017	14.0	74.50	75.75	
514	514	Sviatoslav Mykhailiuk	2017	14.0	78.50	79.50	
515	515	Thomas Welsh	2017	14.0	83.50	84.50	
516	516	V.J. Beachem	2017	14.0	78.25	80.00	

517 rows × 19 columns

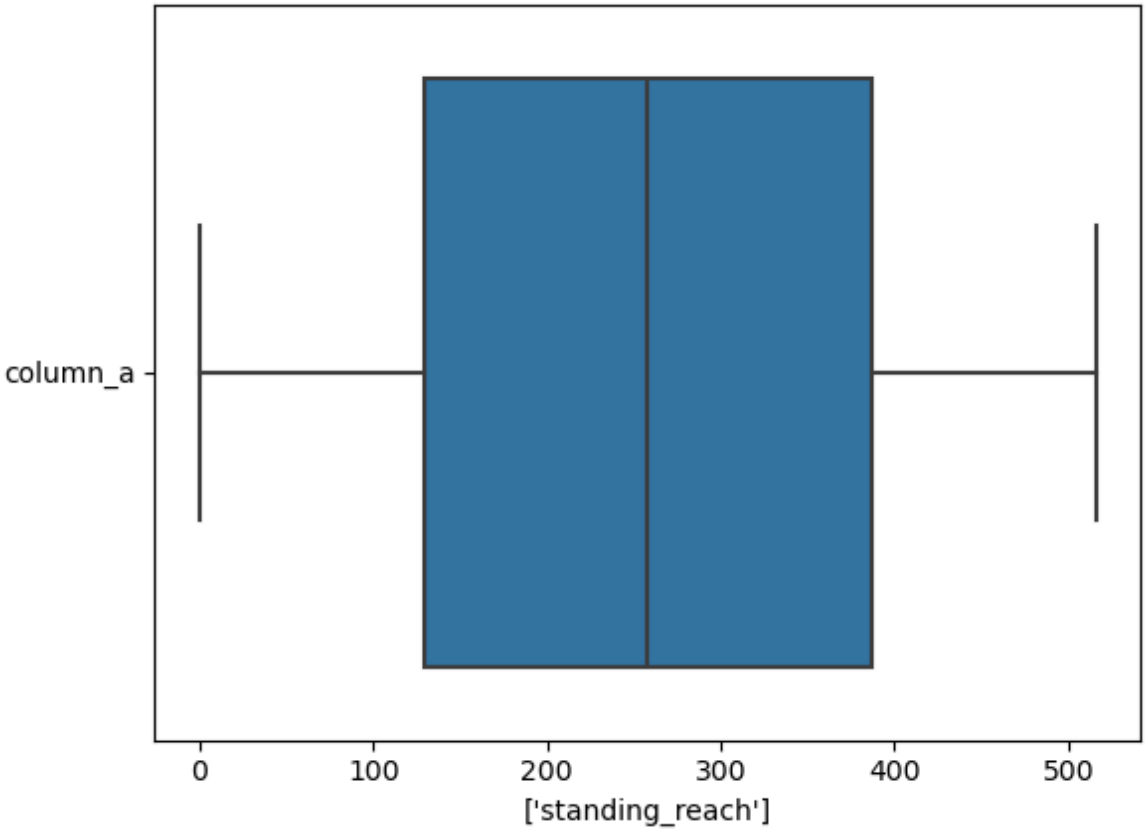
In [29]:

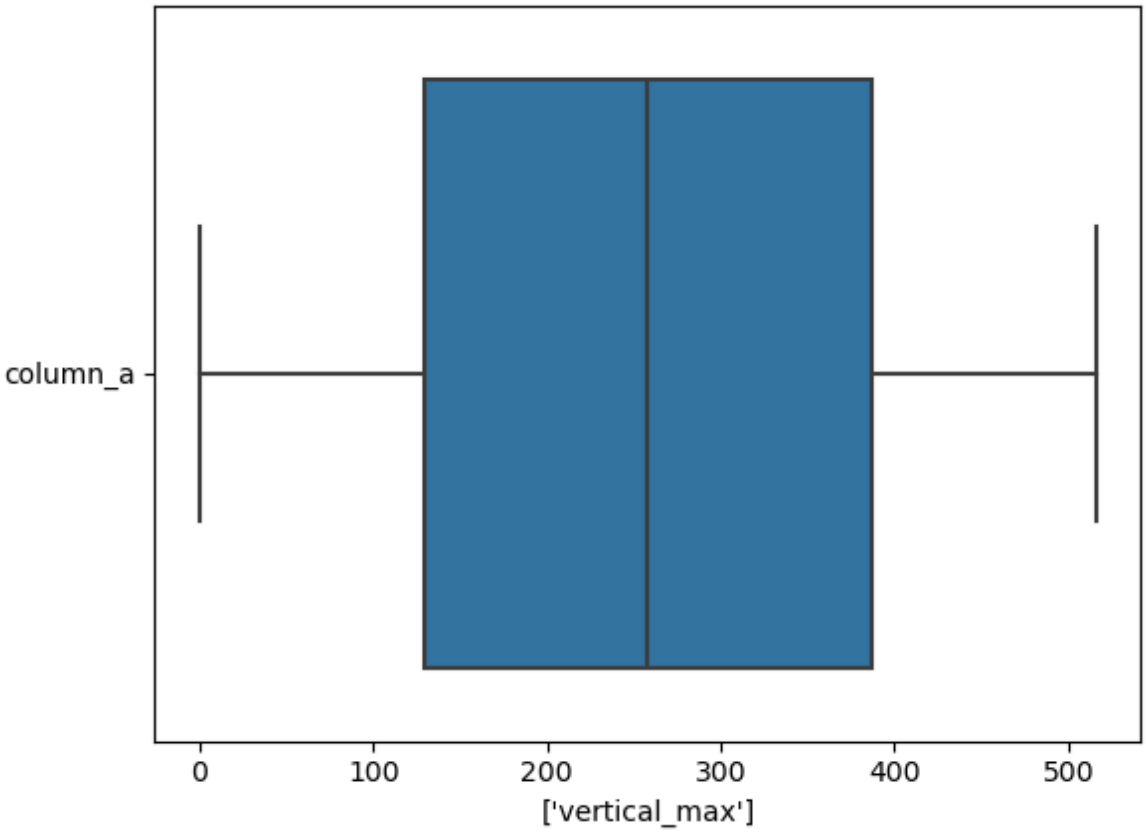
```
# Without any outliers
height=non_outliers_df[['height_no_shoes']]
sns.boxplot(height,orient='h')
plt.xlabel('height_no_shoes')
plt.show()
```

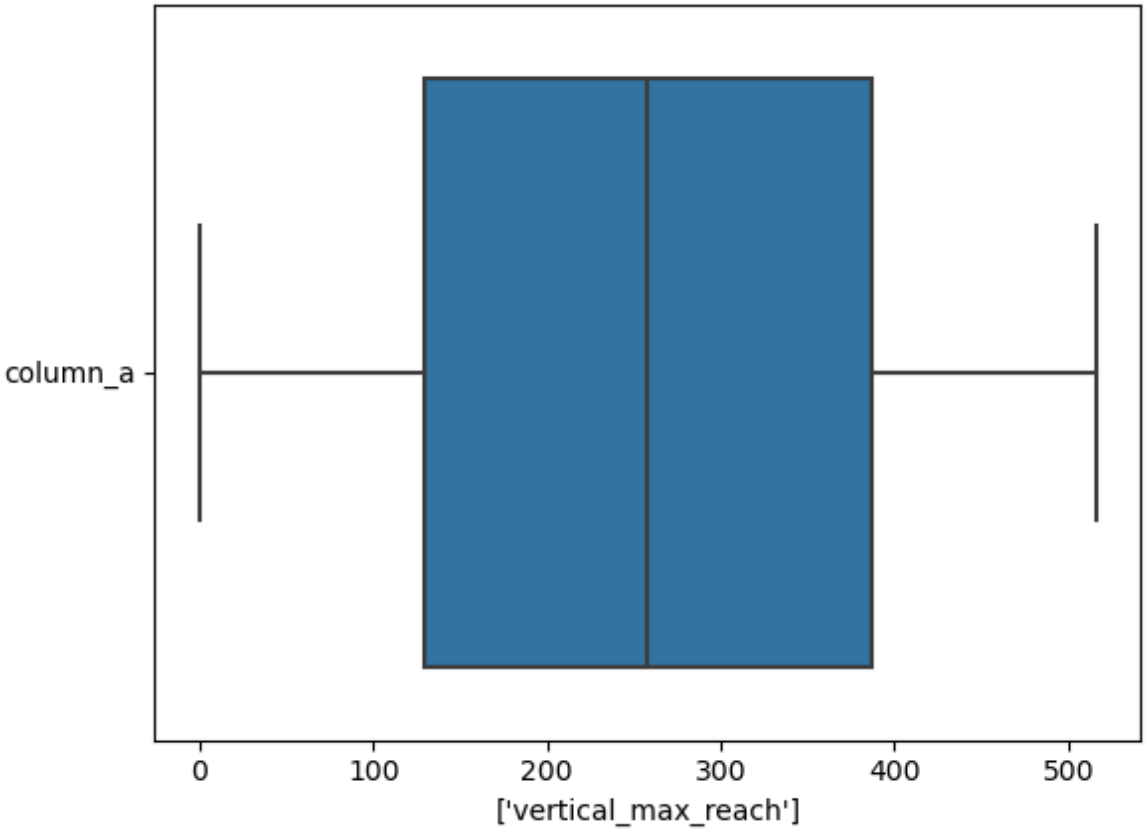


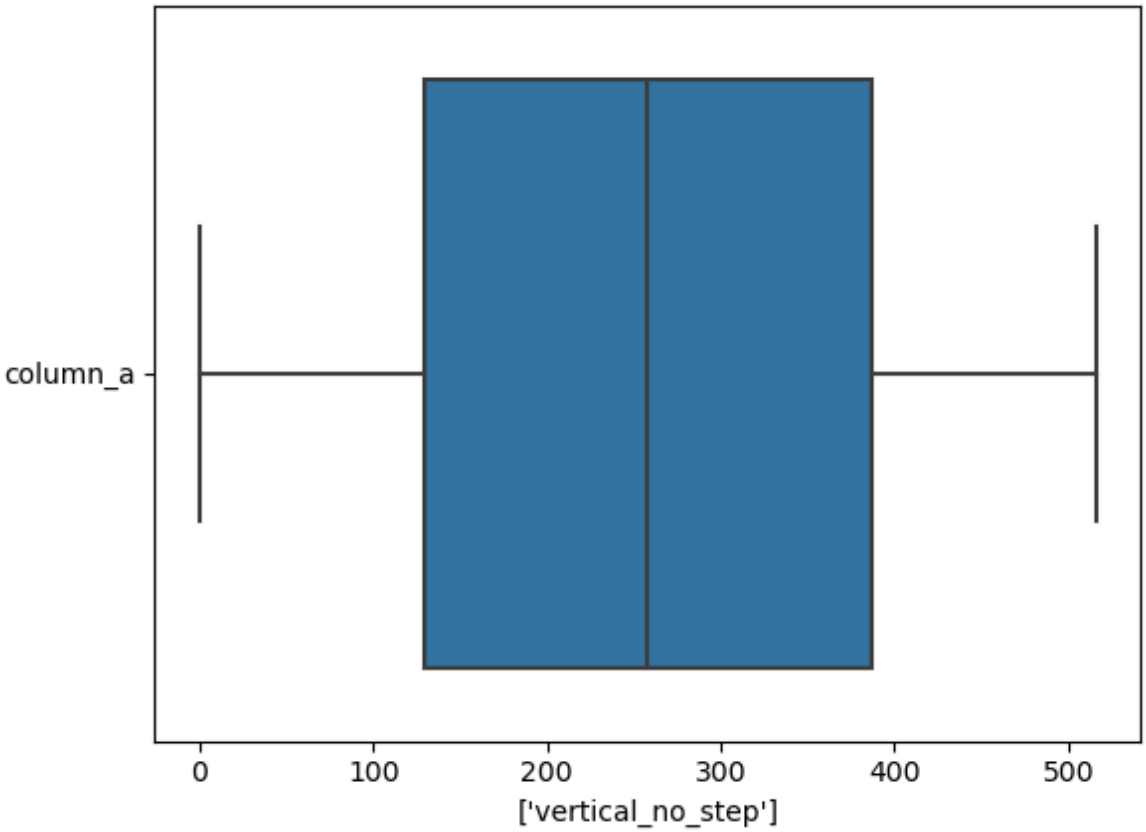
box plot without any outliers

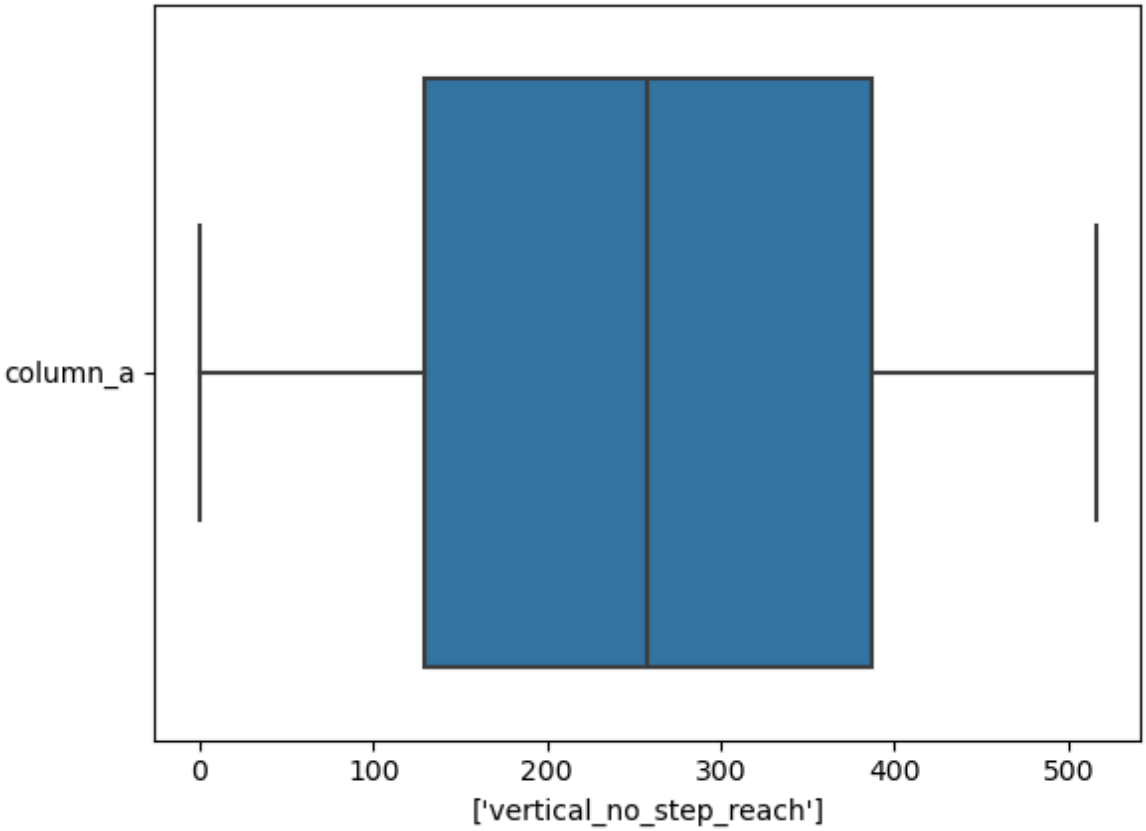
In [30]:
```
for i in num_cols[6:]:
    height=non_outliers_df[['column_a']]
    sns.boxplot(height,orient='h')
    plt.xlabel([i])
    plt.show()
```

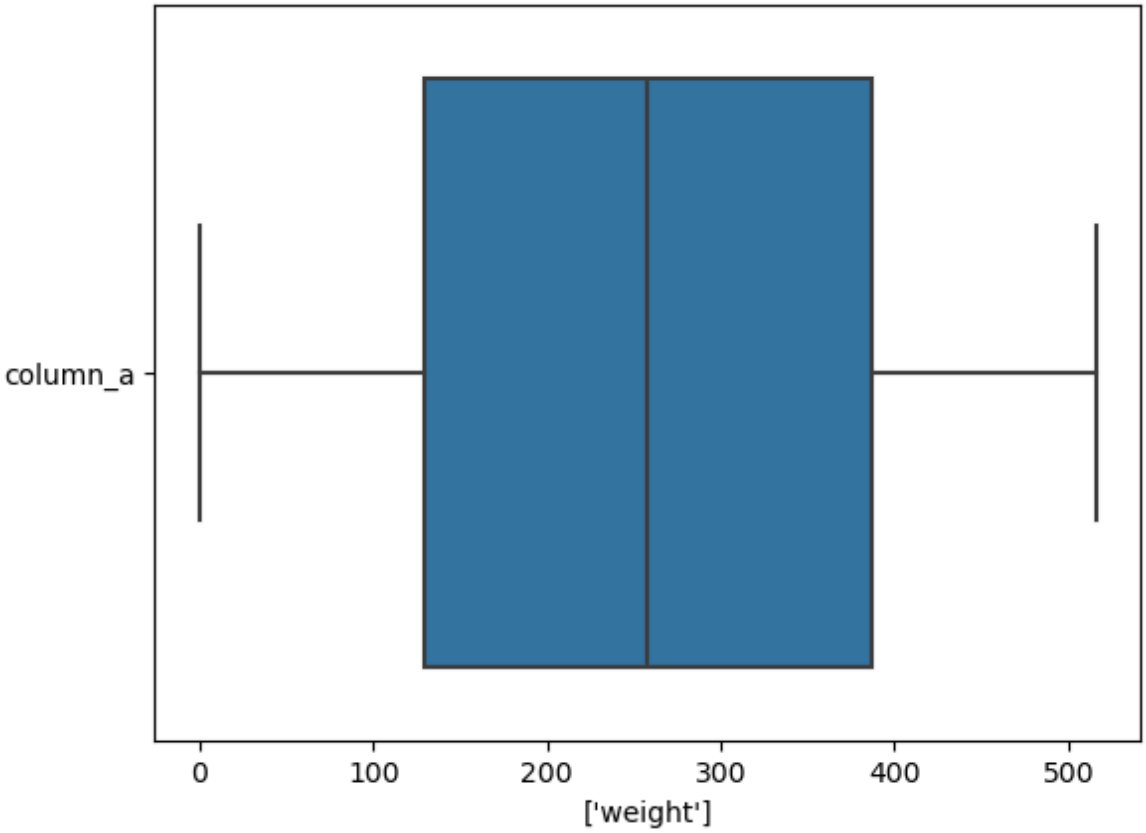


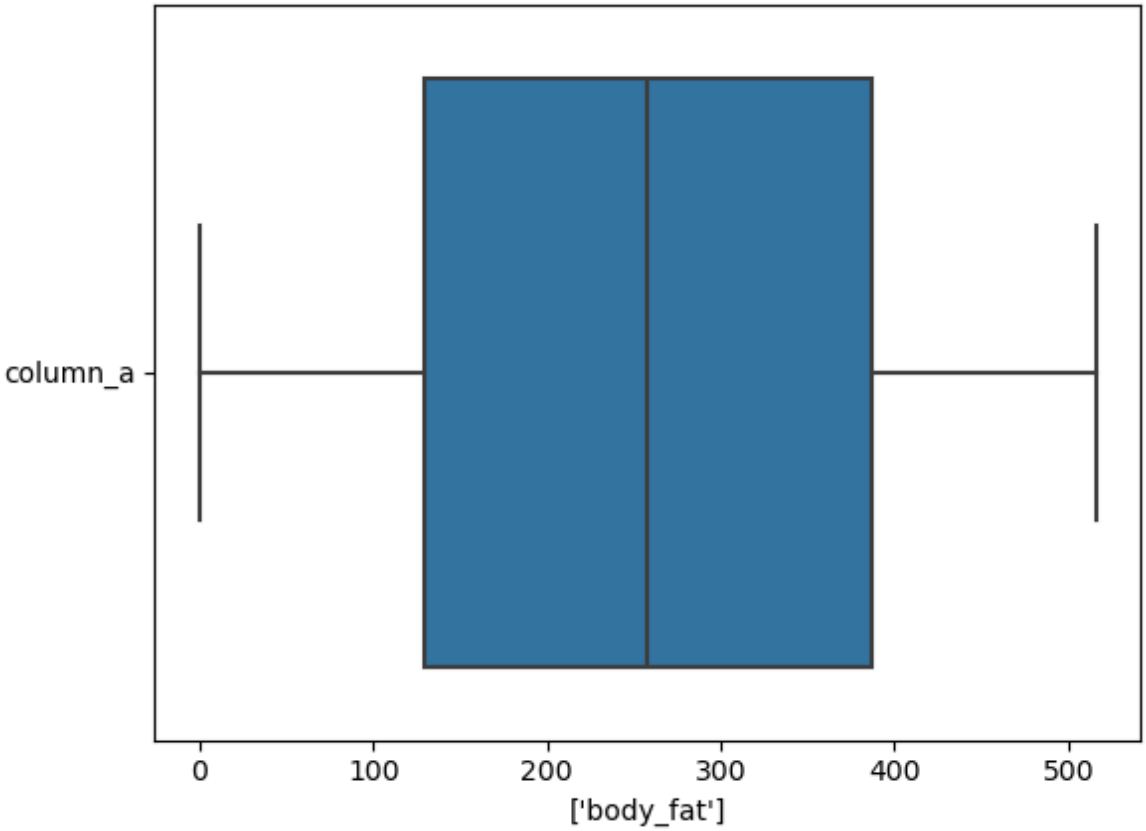

column_a
0
100
200
300
400
500
['vertical_max']

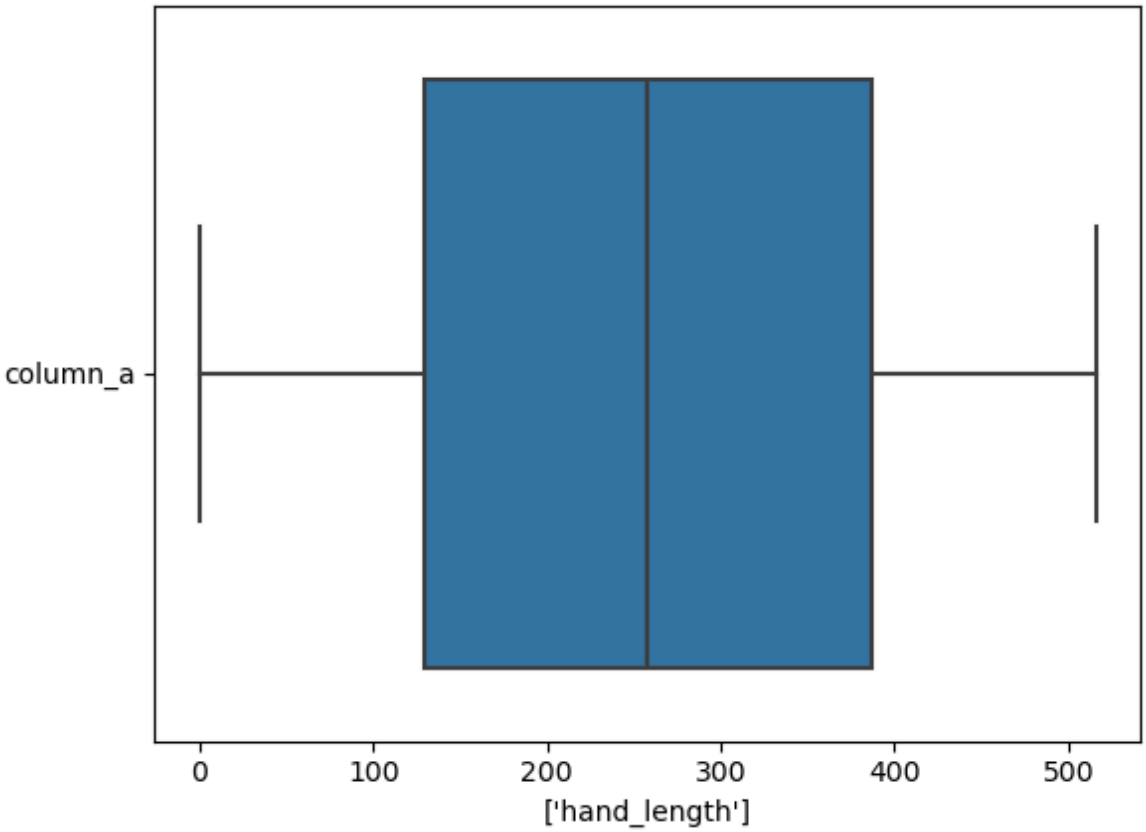

column_a
0
100
200
300
400
500
['vertical_max_reach']

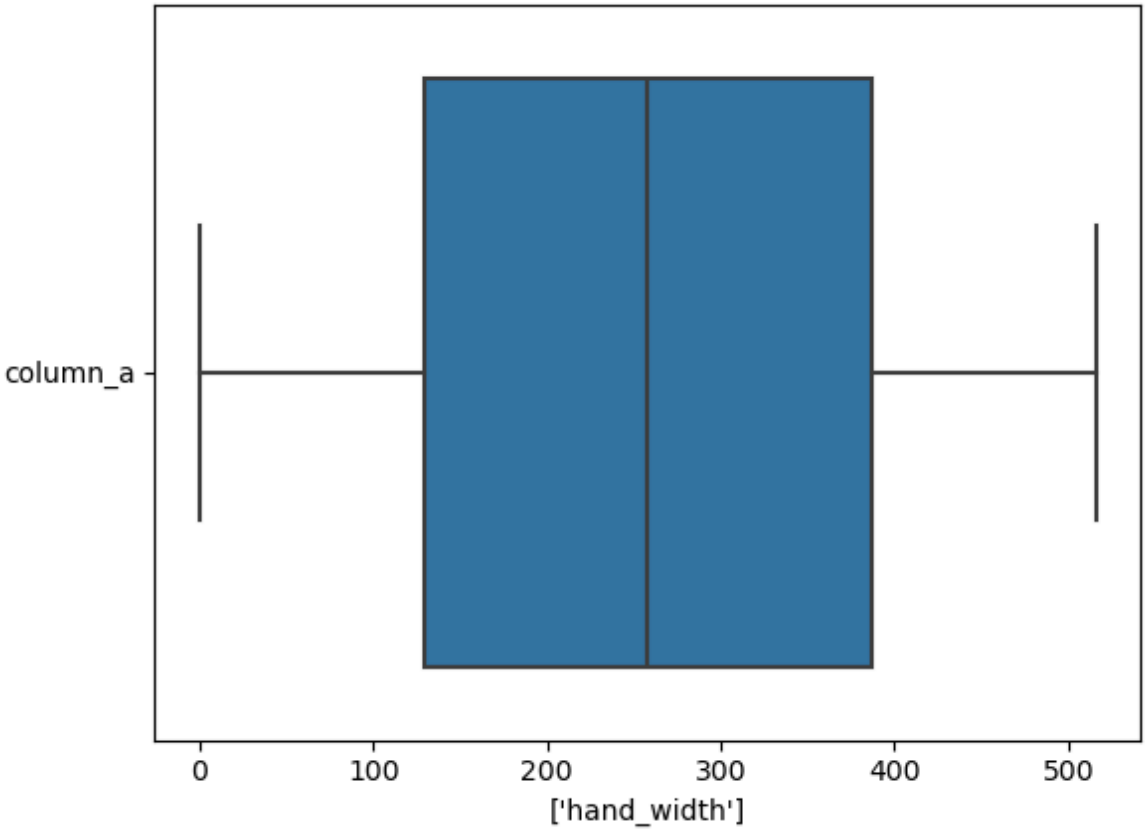

column_a
0
100
200
300
400
500
['vertical_no_step']

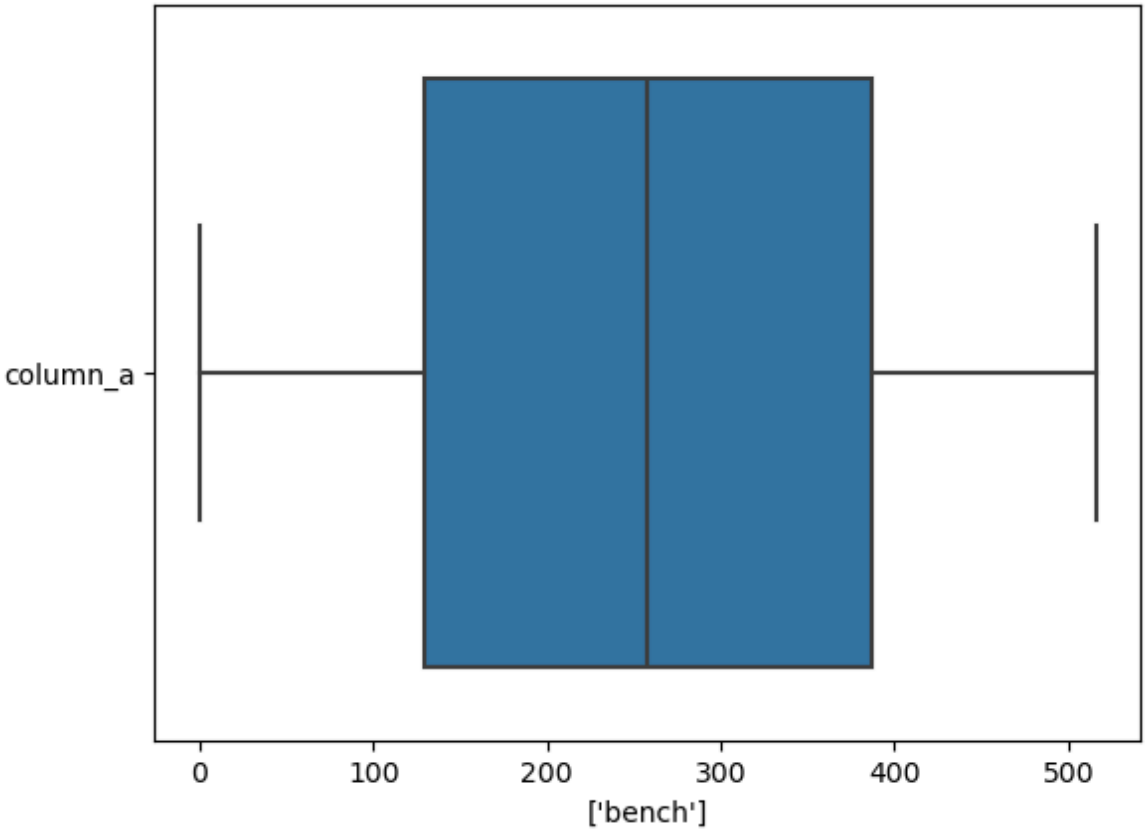

column_a
0
100
200
300
400
500
['vertical_no_step_reach']

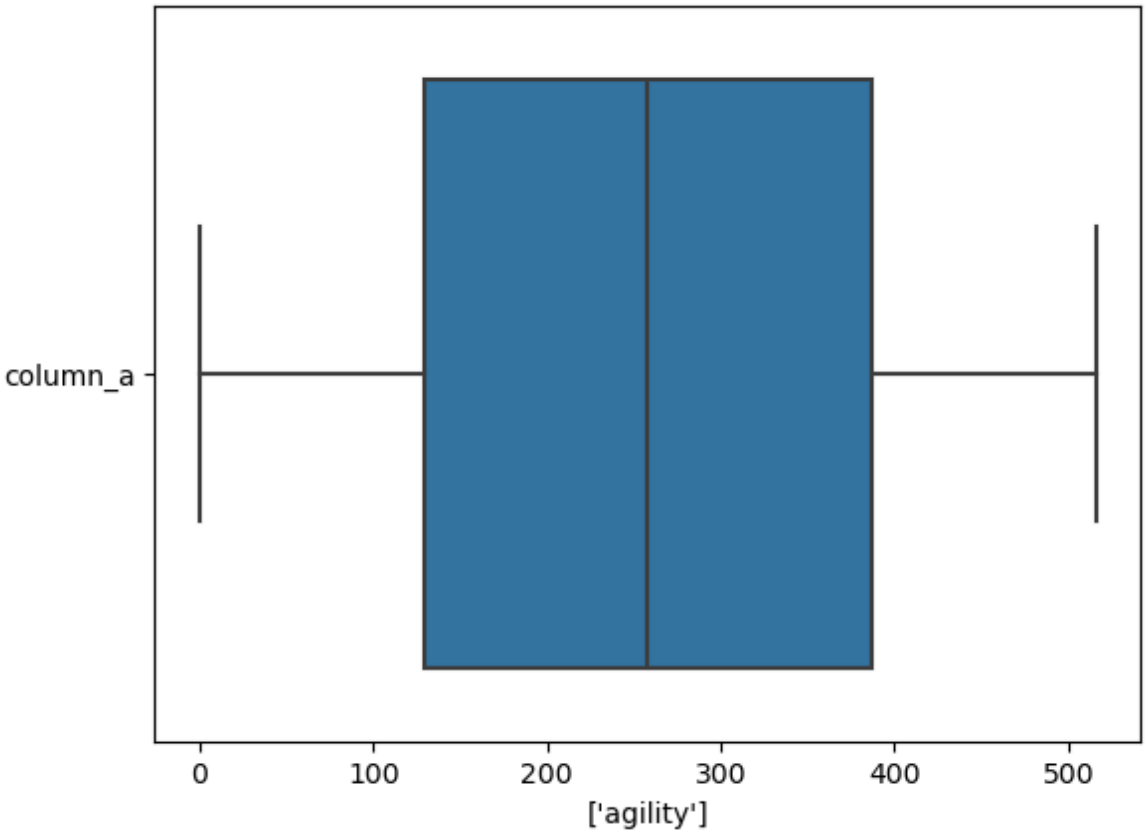

column_a
0
100
200
300
400
500
['weight']

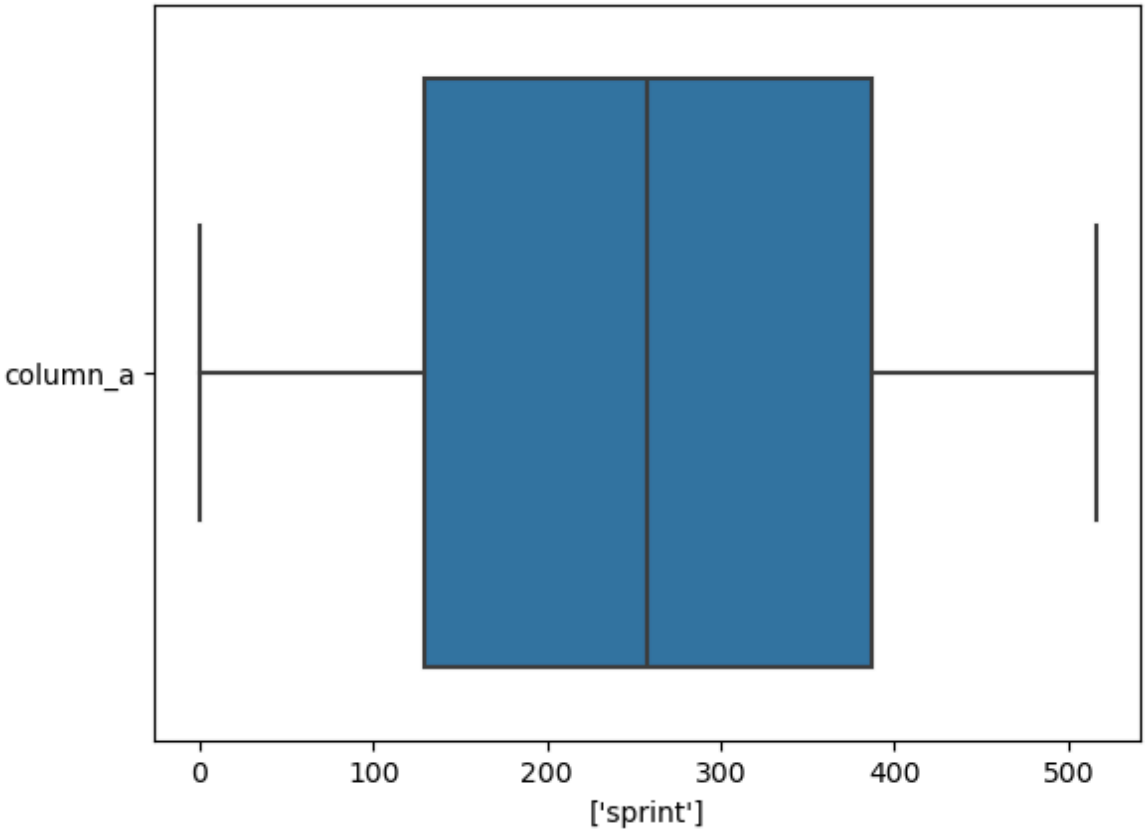

column_a
0
100
200
300
400
500
['body_fat']


column_a
0
100
200
300
400
500
['hand_length']


column_a
0
100
200
300
400
500
['hand_width']


column_a
0
100
200
300
400
500
['bench']


column_a
0
100
200
300
400
500
['agility']



In [ ]: