PyPI repository

The actual package can be found at https://pypi.org/project/ocpexplore/ (https://pypi.org/project/ocpexplore/)

Github

The codes of the functions and a readme file can be found on github at
https://github.com/martonberta/ocpexplore (https://github.com/martonberta/ocpexplore)

Possum dataset

The possum data frame consists of nine morphometric measurements on each of 104 **mountain brushtail possums**, trapped at seven sites from Southern Victoria to central Queensland.

Load data and packages

In [1]:

```
import pandas as pd
import numpy as np
```

In [2]:

```
possums = pd.read_csv("https://vincentarelbundock.github.io/Rdatasets/csv/DAAG/possum.csv")
```

In [3]:

```
# create a random date feature for the sake of the presentation
import random
possums['temp'] = [random.uniform(0,1) for i in range(possums.shape[0])]

possums.loc[possums.temp <= 0.2, 'date'] = pd.to_datetime("2019-06-01")
possums.loc[(possums.temp > 0.2) & (possums.temp <= 0.4) , 'date'] = pd.to_datetime("2019-07-01")
possums.loc[possums.temp >= 0.8, 'date'] = pd.to_datetime("2019-10-01")
possums.loc[(possums.temp >= 0.6) & (possums.temp < 0.8) , 'date'] = pd.to_datetime("2019-09-01")
possums.loc[possums.date.isnull(), 'date'] = pd.to_datetime("2019-08-01")

# rename cols
possums = possums.rename(columns={"case": "possum_id", "site":"site_id", "Pop":"region"})


# drop some features
todrop = ['temp', 'Unnamed: 0', 'hdlngth', 'skullw', 'taill', 'footlgth','earconch', 'eye']
possums = possums.drop(todrop, axis = 1)
```

In [4]:

```
possums.head(10)
```

Out[4]:

	possum_id	site_id	region	sex	age	totlngth	chest	belly	date
0	1	1	Vic	m	8.0	89.0	28.0	36.0	2019-06-01
1	2	1	Vic	f	6.0	91.5	28.5	33.0	2019-09-01
2	3	1	Vic	f	6.0	95.5	30.0	34.0	2019-06-01
3	4	1	Vic	f	6.0	92.0	28.0	34.0	2019-06-01
4	5	1	Vic	f	2.0	85.5	28.5	33.0	2019-09-01
5	6	1	Vic	f	1.0	90.5	30.0	32.0	2019-09-01
6	7	1	Vic	m	2.0	89.5	30.0	34.5	2019-08-01
7	8	1	Vic	f	6.0	91.0	29.0	34.0	2019-07-01
8	9	1	Vic	f	9.0	91.5	28.0	33.0	2019-07-01
9	10	1	Vic	f	6.0	89.5	27.5	32.0	2019-08-01

Functions of the ocpexplore package

The library can be loaded with the following code

In [5]:

```
!pip install ocpexplore
import ocpexplore.ocpexplore as expl
```

```
Requirement already satisfied: ocpexplore in /home/ec2-user/anaconda3/env
s/python3/lib/python3.6/site-packages (0.1.0)
Requirement already satisfied: Click>=7.0 in /home/ec2-user/anaconda3/env
s/python3/lib/python3.6/site-packages (from ocpexplore) (7.1.1)
You are using pip version 10.0.1, however version 20.0.2 is available.
You should consider upgrading via the 'pip install --upgrade pip' command.
```

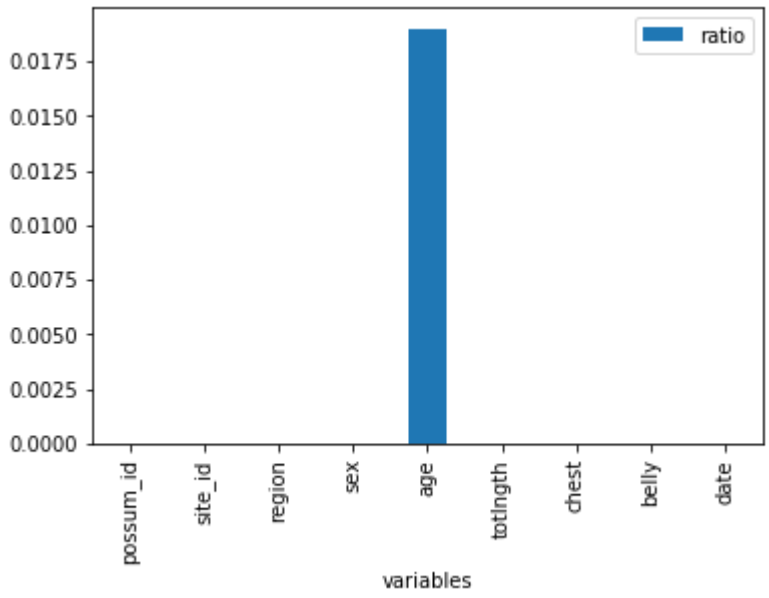
Missing values

From the chart it can be seen that only the age variable has missing values. This feature has 2 missing values, which is nearly 2 % of the total number of observations.

In [6]:

```
expl.check_NA(possums)
```

```
   variables  NANs  ratio
0  possum_id     0  0.000
1    site_id     0  0.000
2     region     0  0.000
3        sex     0  0.000
4        age     2  0.019
5   totlngth     0  0.000
6      chest     0  0.000
7      belly     0  0.000
8       date     0  0.000
```



Keys in the data

The possum_id variable is unique in the data, while site_id is not (as expected), with 7 unique observations

In [7]:

```
expl.check_ID(possums[['possum_id', 'site_id']])
```

```
possum_id has 104 values, and 104 of which are unique (100.0%).
site_id has 104 values, and 7 of which are unique (6.73%).
```

Object variables

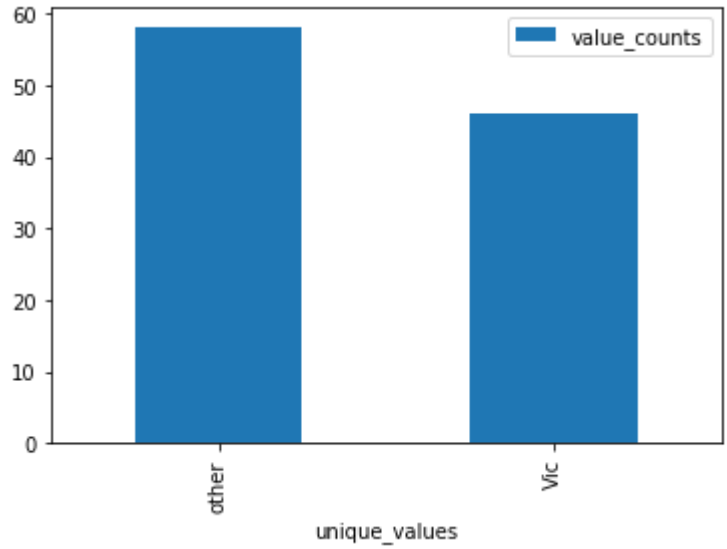
With the `value_counter` function we can check the distribution of object type variables if the number of unique observations is less than 25.

In [8]:

```
expl.value_counter(possums)
```

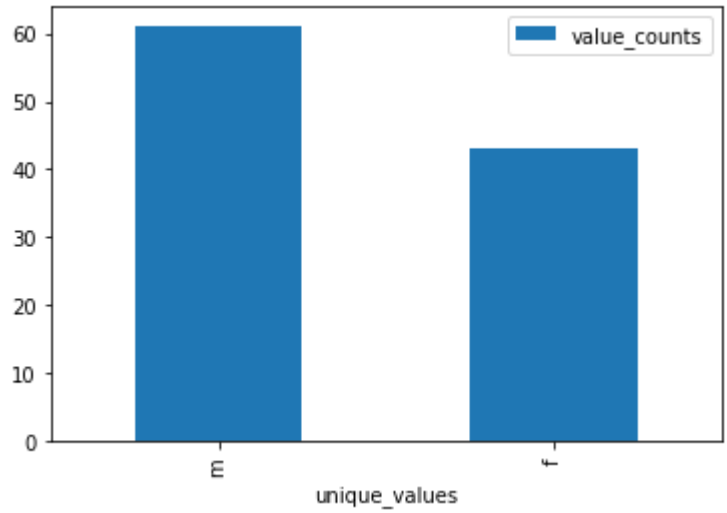
region

	unique_values	value_counts	ratio
0	other	58	0.558
1	Vic	46	0.442



sex

	unique_values	value_counts	ratio
0	m	61	0.587
1	f	43	0.413



Unprocessed variables with object dtype: NA

Explore numeric variables

We can check the distribution of numeric variables visually and also with a descriptives table

In [9]:

```
expl.describe_continuous(possums[['age','totlngth','chest','belly']])
```

Out[9]:

	variable	Min	0.01	0.05	0.1	0.25	0.5	0.75	0.9	0.95	0.99	Max	Mean	NaN
0	age	1.0	1.0	1.0	2.0	2.0	3.0	5.0	6.0	7.0	9.0	9.0	3.833333	2
1	totlngth	75.0	76.0	80.5	81.5	84.0	88.0	90.0	92.0	93.5	96.0	96.5	87.088462	0
2	chest	22.0	23.0	23.5	24.5	25.5	27.0	28.0	30.0	30.5	31.0	32.0	27.000000	0
3	belly	25.0	27.0	28.0	29.0	31.0	32.5	34.0	36.0	36.5	39.0	40.0	32.586538	0

We drop (or impute) the rows with missing values, because the plot function does not handle missing values yet.

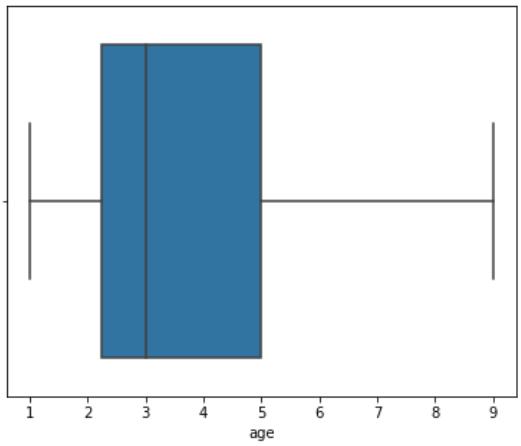
In [10]:

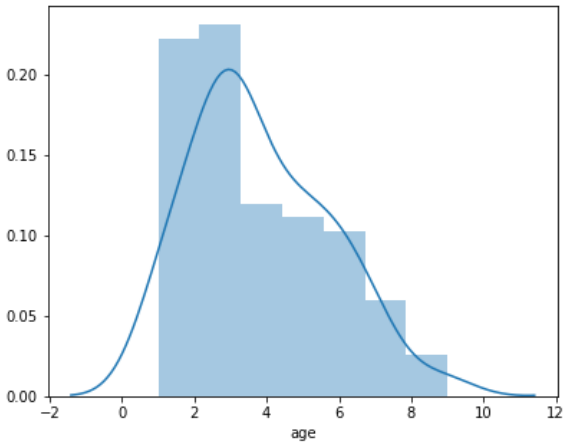
```
possums = possums.dropna()
```

In [11]:

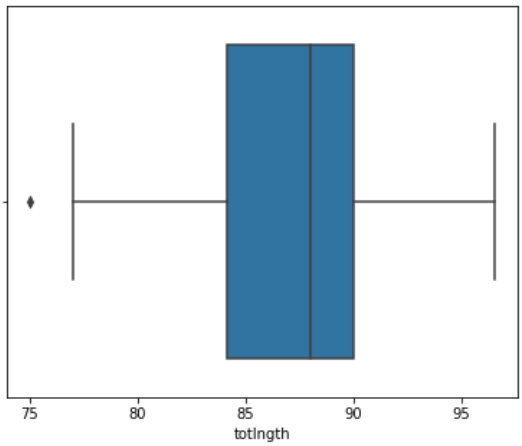
```
expl.plot_continuous(possums[['age','totlngth','chest','belly']])
```

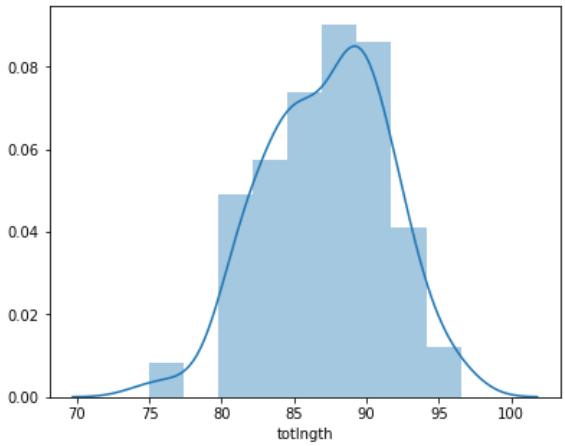
age



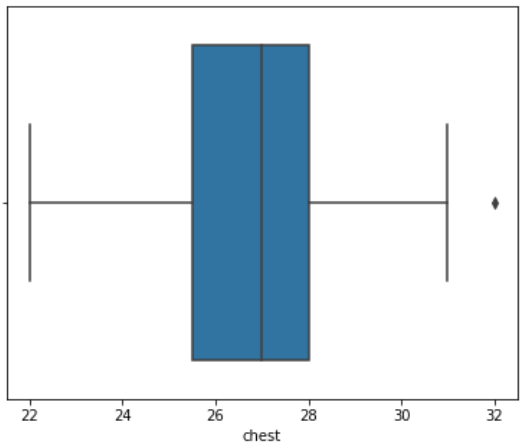


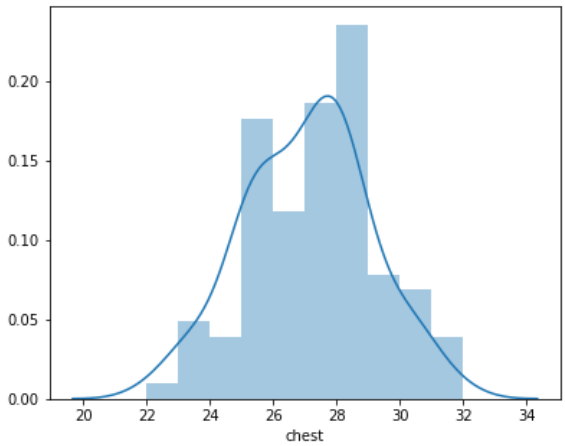
totlngth

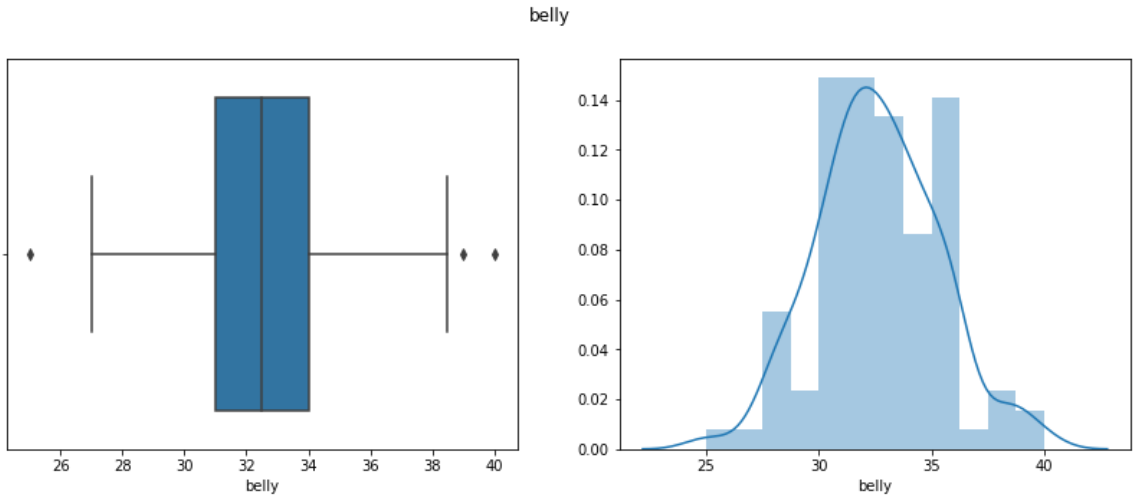




chest







The boxplots showed some outliers, which we can check separately with the `tail_density_table`

In [12]:

```
expl.tail_density_table(possums[['age','totlngth','chest','belly']])
```

Out[12]:

bucket	low_extreme	low_outlier	non_outlier	high_outlier	high_extreme
age	0	0	102	0	0
totlngth	0	1	101	0	0
chest	0	0	101	1	0
belly	0	1	99	2	0

Explore date variables

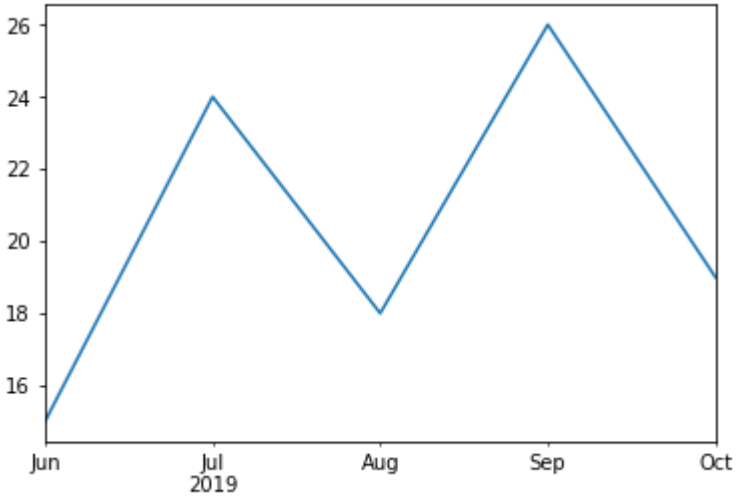
We can check if a day/month/year is missing from the data with the `obs_by_date` function. Contrary to the rest of the functions, this takes a single column as input.

In [13]:

```
expl.obs_by_date(possums.date, date_aggregation = 'M')
```

Out[13]:

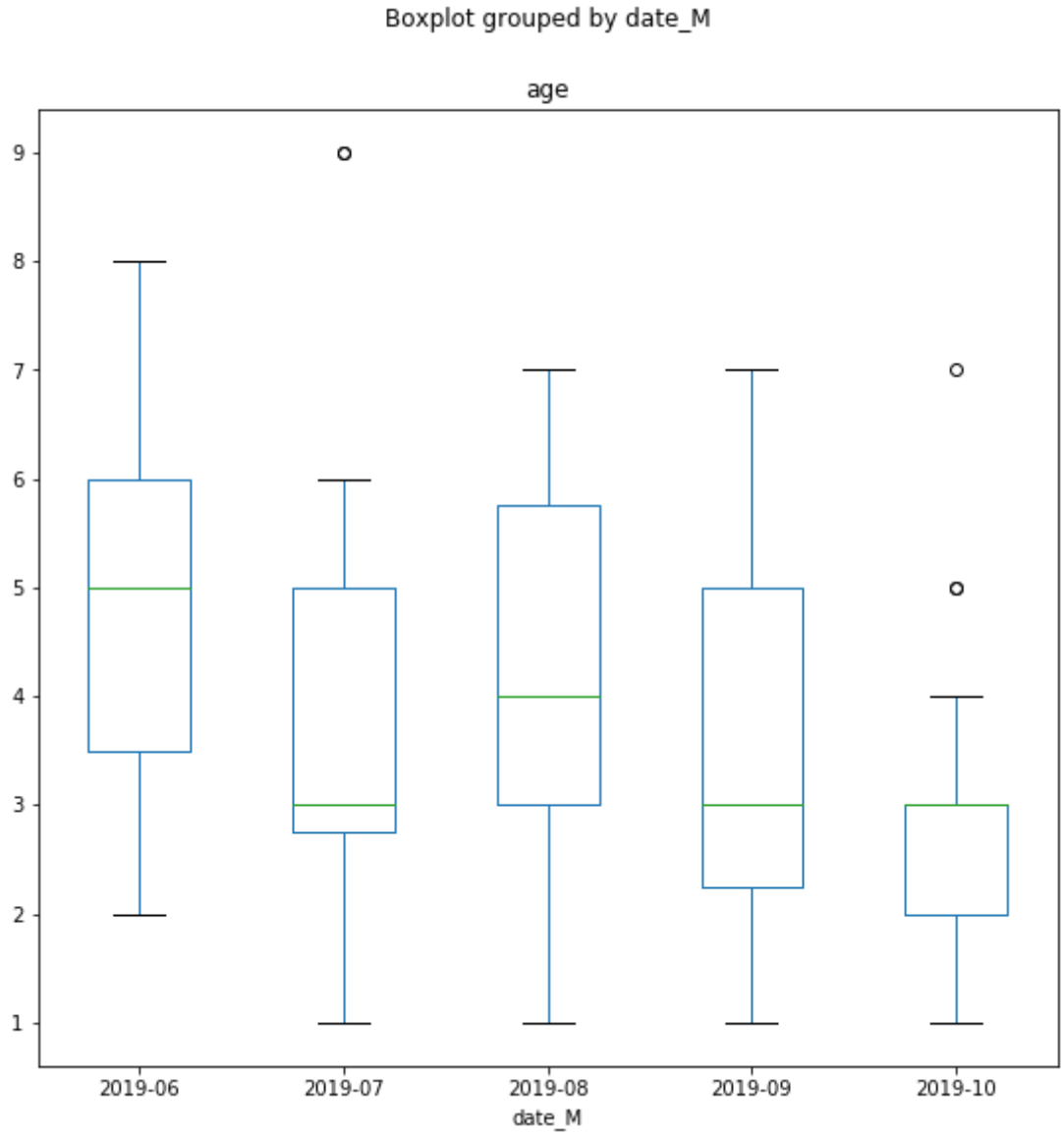
```
2019-06    15
2019-07    24
2019-08    18
2019-09    26
2019-10    19
Freq: M, Name: date, dtype: int64
```

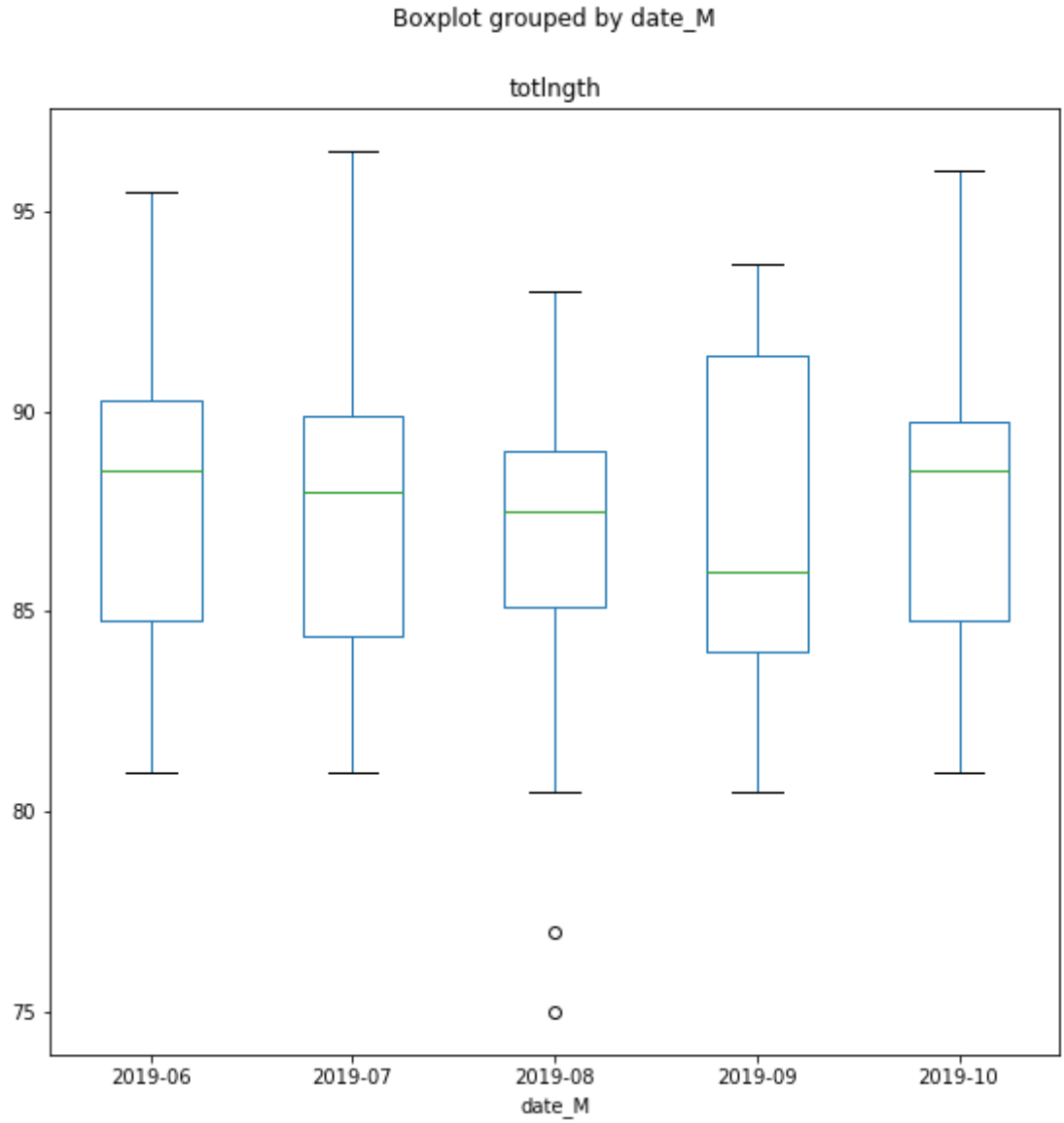


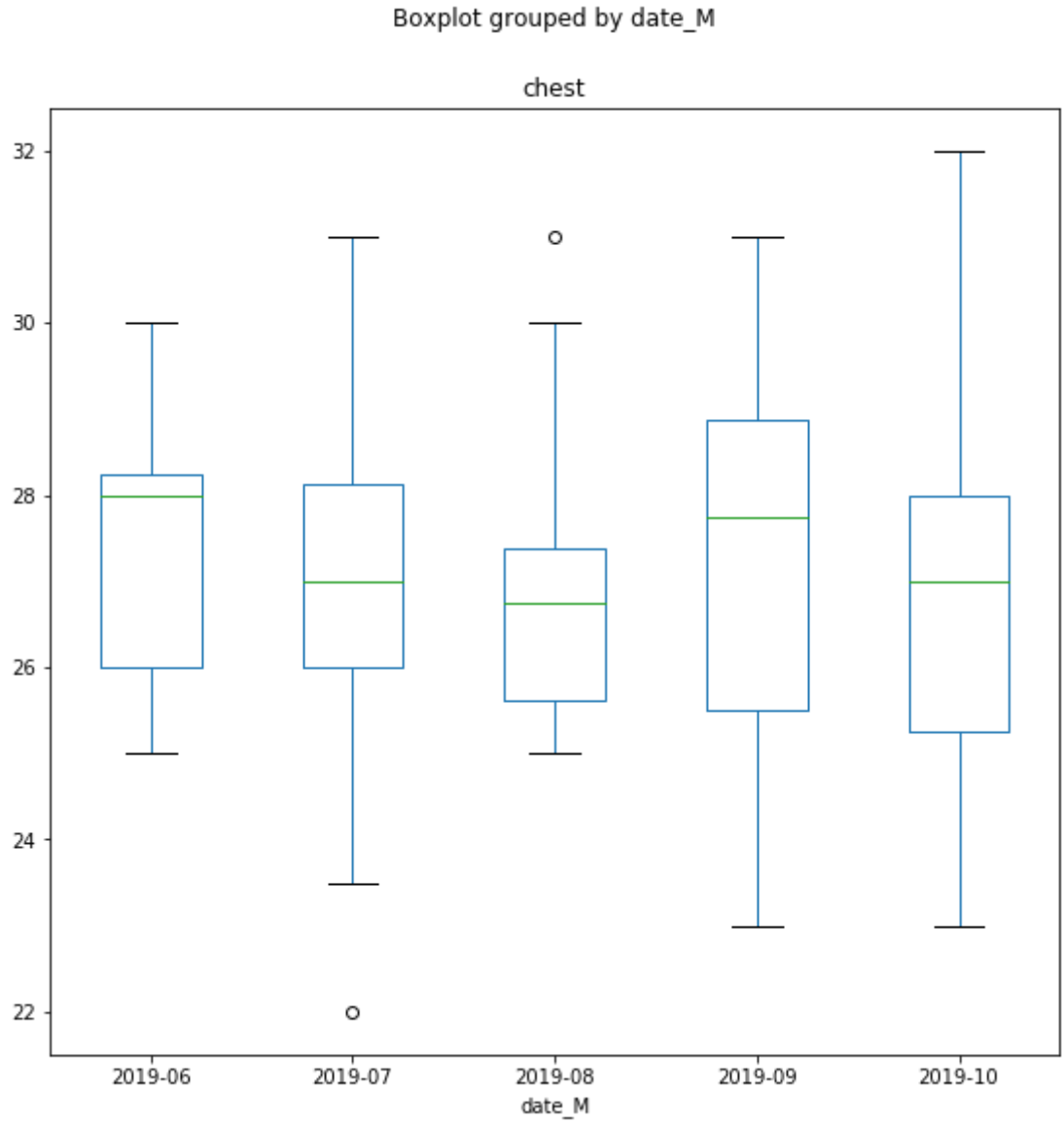
Also, we can check the distribution of numeric variables over time with the `values_by_date` function

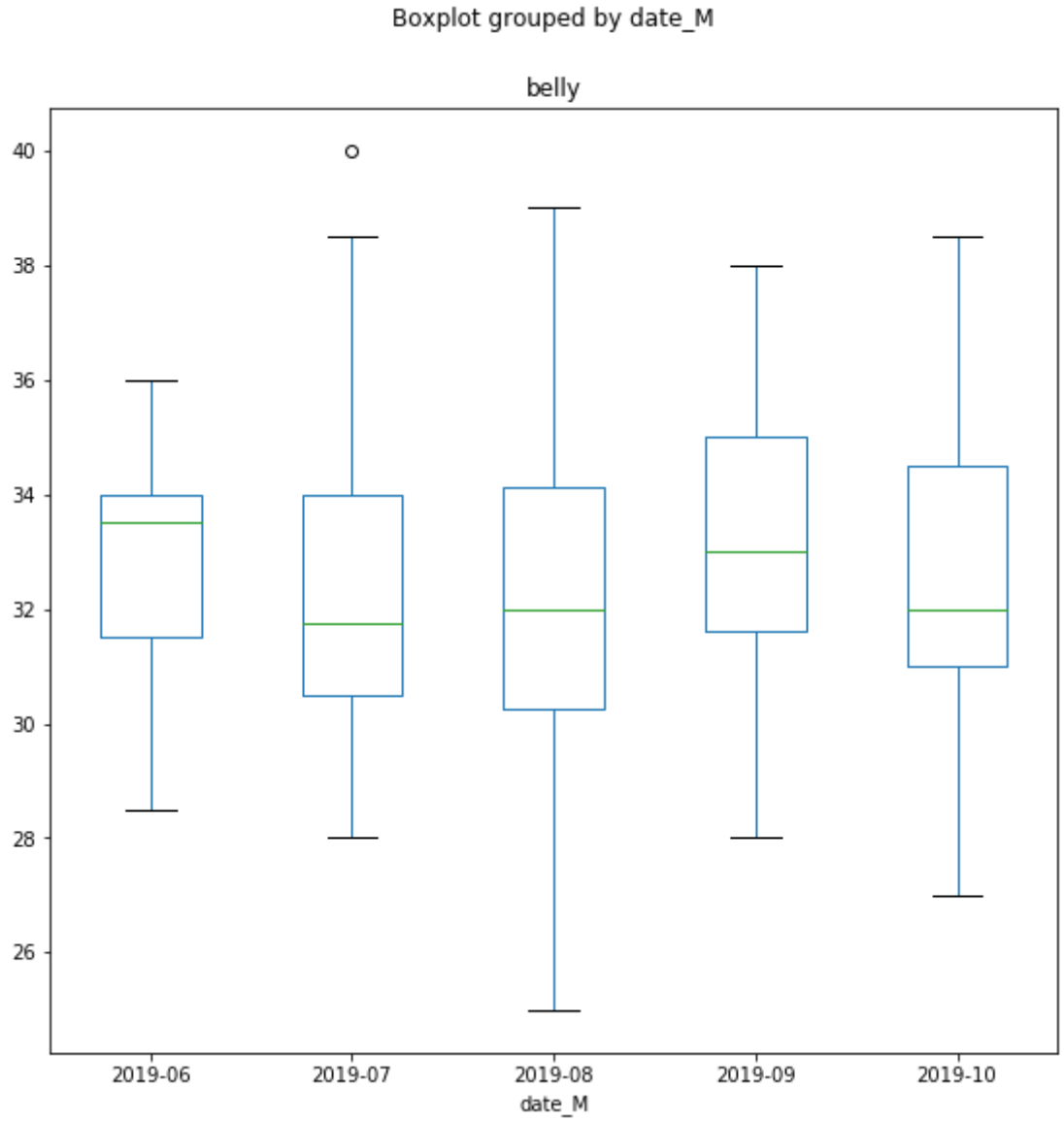
In [14]:

```
expl.values_by_date(possums[['date','age','totlngth','chest','belly']], 'date')
```


Boxplot grouped by date_M
age
9
8
7
6
5
4
3
2
1
2019-06
2019-07
2019-08
2019-09
2019-10
date_M


Boxplot grouped by date_M
totlngth
95
90
85
80
75
2019-06
2019-07
2019-08
2019-09
2019-10
date_M


Boxplot grouped by date_M
chest
32
30
28
26
24
22
2019-06
2019-07
2019-08
2019-09
2019-10
date_M


Boxplot grouped by date_M
belly
40
38
36
34
32
30
28
26
2019-06
2019-07
2019-08
2019-09
2019-10
date_M

Lifecycle configuration

Doug created a process description, which makes it possible to have any package preinstalled upon opening a notebook instance in AWS. That can also be used to install this package.