a1_task1

October 2, 2021

1 Arttu Häkkinen

2 596077

[1]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[2]:
```python
# Download the data as pandas dataframe object
data = pd.read_csv("corrtestdata.csv")
data.head()
```

[2]:
```
     id  day  weight  gender  liverind  heartind    appind  femstate  \
0  rat1   67    22.0       1  0.036364  0.005091  0.013636         4
1  rat2  251   182.0       1  0.010440  0.004396  0.008791         4
2  rat3  230    37.5       1  0.051200  0.006133  0.020800         4
3  rat4  261    45.0       2  0.047333  0.005556  0.105111        -1
4  rat5  262    43.0       1  0.052558  0.005349  0.021628         4

  gonfatind    batind  sulcer  kmethod   tailind  blength  place  year  \
0  0.000000  0.001864       1        2  0.714286     10.5      1     5
1  0.023077  0.000742       3        2  0.666667     19.5      3     3
2  0.000000  0.001467       2        1  0.904762     10.5      2     8
3  0.293333  0.002178       1        1  0.869565     11.5      4     2
4  0.000000  0.001884       1        1  0.720000     12.5      2     8

    ADWBind    gonind       BMI
0  0.454545  0.000000  0.199546
1  0.302198  1.648659  0.478632
2  0.653333  0.000000  0.340136
3  0.260000  2.653242  0.340265
4  0.500000  0.000000  0.275200
```

[3]:
```python
data.describe()
```

[3]:
```
              day      weight      gender    liverind    heartind      appind  \
count  563.000000  563.000000  563.000000  563.000000  563.000000  563.000000
```

mean	185.609236	266.009769	1.373002	0.053103	0.011241	0.014330
std	94.541938	99.563358	0.484033	0.017590	0.167074	0.006936
min	12.000000	22.000000	1.000000	0.010440	0.002215	0.002949
25%	127.000000	202.500000	1.000000	0.042737	0.003615	0.010406
50%	177.000000	275.000000	1.000000	0.051402	0.004106	0.013456
75%	250.500000	339.000000	2.000000	0.061439	0.004637	0.016789
max	400.000000	500.000000	2.000000	0.309273	3.968379	0.105111

	femstate	gonfatind	batind	sulcer	kmethod	tailind	\
count	563.000000	563.000000	563.000000	563.000000	563.000000	563.000000	
mean	1.415631	0.009233	0.001392	1.436945	1.547069	0.783158	
std	2.095896	0.017207	0.000620	0.852578	1.207555	0.060422	
min	-1.000000	0.000000	0.000031	1.000000	1.000000	0.500000	
25%	-1.000000	0.001846	0.000929	1.000000	1.000000	0.750000	
50%	1.000000	0.005686	0.001287	1.000000	1.000000	0.778846	
75%	4.000000	0.010382	0.001723	2.000000	2.000000	0.824316	
max	4.000000	0.293333	0.004866	5.000000	6.000000	0.986667	

	blength	place	year	ADWBind	gonind	BMI
count	563.000000	563.000000	563.000000	563.000000	563.000000	563.000000
mean	20.462877	2.490231	6.662522	0.386439	0.953519	0.603400
std	2.897211	1.571938	2.985695	0.131427	0.769399	0.130781
min	10.500000	1.000000	2.000000	0.052299	0.000000	0.130874
25%	19.500000	2.000000	5.000000	0.292102	0.262364	0.514815
50%	21.000000	2.000000	7.000000	0.379808	0.970779	0.607743
75%	22.500000	2.000000	8.000000	0.466617	1.410987	0.694640
max	26.500000	9.000000	15.000000	0.974359	3.945264	1.005949

2.1 a)

Calculate all pairwise correlations between features, excluding only the rat id. Report strongest correlations involving i) categorical features, ii) temporal features day and year, ii) other numerical features.

[4]:
```
pairwise_correlations = data.corr(method='pearson').round(decimals=2)
pairwise_correlations
```

[4]:

	day	weight	gender	liverind	heartind	appind	femstate	\
day	1.00	0.12	-0.25	-0.12	-0.01	0.22	0.26	
weight	0.12	1.00	-0.09	-0.24	-0.01	-0.19	-0.01	
gender	-0.25	-0.09	1.00	-0.21	0.06	-0.16	-0.89	
liverind	-0.12	-0.24	-0.21	1.00	-0.04	0.30	0.16	
heartind	-0.01	-0.01	0.06	-0.04	1.00	-0.06	-0.05	
appind	0.22	-0.19	-0.16	0.30	-0.06	1.00	0.21	
femstate	0.26	-0.01	-0.89	0.16	-0.05	0.21	1.00	
gonfatind	0.39	0.15	-0.06	-0.26	0.01	0.43	0.10	
batind	-0.14	-0.20	-0.00	0.09	-0.04	0.15	0.09	

sulcer	-0.24	-0.11	0.05	-0.04	0.13	-0.11	-0.06
kmethod	0.45	0.19	-0.15	-0.32	0.01	0.12	0.26
tailind	0.28	-0.22	-0.07	0.09	-0.04	0.17	0.12
blength	0.00	0.88	-0.10	-0.19	0.02	-0.22	-0.03
place	0.70	0.24	-0.25	-0.31	0.01	0.22	0.30
year	0.40	0.24	-0.11	-0.07	-0.05	0.17	0.23
ADWBind	-0.20	-0.41	-0.24	0.33	-0.06	-0.01	0.19
gonind	0.44	0.53	-0.04	-0.40	0.02	-0.01	0.04
BMI	0.20	0.88	-0.11	-0.24	-0.03	-0.15	0.04

	gonfatind	batind	sulcer	kmethod	tailind	blength	place	year \
day	0.39	-0.14	-0.24	0.45	0.28	0.00	0.70	0.40
weight	0.15	-0.20	-0.11	0.19	-0.22	0.88	0.24	0.24
gender	-0.06	-0.00	0.05	-0.15	-0.07	-0.10	-0.25	-0.11
liverind	-0.26	0.09	-0.04	-0.32	0.09	-0.19	-0.31	-0.07
heartind	0.01	-0.04	0.13	0.01	-0.04	0.02	0.01	-0.05
appind	0.43	0.15	-0.11	0.12	0.17	-0.22	0.22	0.17
femstate	0.10	0.09	-0.06	0.26	0.12	-0.03	0.30	0.23
gonfatind	1.00	0.15	-0.11	0.49	0.08	0.06	0.57	0.27
batind	0.15	1.00	0.04	0.20	-0.04	-0.21	0.14	0.01
sulcer	-0.11	0.04	1.00	-0.00	-0.21	-0.02	-0.19	-0.21
kmethod	0.49	0.20	-0.00	1.00	0.06	0.10	0.77	0.54
tailind	0.08	-0.04	-0.21	0.06	1.00	-0.42	0.18	0.30
blength	0.06	-0.21	-0.02	0.10	-0.42	1.00	0.14	-0.01
place	0.57	0.14	-0.19	0.77	0.18	0.14	1.00	0.46
year	0.27	0.01	-0.21	0.54	0.30	-0.01	0.46	1.00
ADWBind	-0.32	0.03	0.28	-0.25	0.05	-0.37	-0.38	-0.16
gonind	0.68	0.07	-0.18	0.52	-0.03	0.44	0.66	0.34
BMI	0.19	-0.20	-0.17	0.24	0.08	0.60	0.28	0.43

	ADWBind	gonind	BMI
day	-0.20	0.44	0.20
weight	-0.41	0.53	0.88
gender	-0.24	-0.04	-0.11
liverind	0.33	-0.40	-0.24
heartind	-0.06	0.02	-0.03
appind	-0.01	-0.01	-0.15
femstate	0.19	0.04	0.04
gonfatind	-0.32	0.68	0.19
batind	0.03	0.07	-0.20
sulcer	0.28	-0.18	-0.17
kmethod	-0.25	0.52	0.24
tailind	0.05	-0.03	0.08
blength	-0.37	0.44	0.60
place	-0.38	0.66	0.28
year	-0.16	0.34	0.43
ADWBind	1.00	-0.53	-0.38

```
gonind        -0.53    1.00  0.50
BMI           -0.38    0.50  1.00
```

2.1.1 i)

Categorical variables are the following:

- gender binary (nominal)
- femstate nominal
- sulcer ordinal
- kmethod nominal
- place nominal

[5]:
```python
pairwise_correlations[['gender', 'femstate', 'sulcer', 'kmethod', 'place']].
↪where(lambda x: (np.abs(x) >= 0.4))
```

[5]:

	gender	femstate	sulcer	kmethod	place
day	NaN	NaN	NaN	0.45	0.70
weight	NaN	NaN	NaN	NaN	NaN
gender	1.00	-0.89	NaN	NaN	NaN
liverind	NaN	NaN	NaN	NaN	NaN
heartind	NaN	NaN	NaN	NaN	NaN
appind	NaN	NaN	NaN	NaN	NaN
femstate	-0.89	1.00	NaN	NaN	NaN
gonfatind	NaN	NaN	NaN	0.49	0.57
batind	NaN	NaN	NaN	NaN	NaN
sulcer	NaN	NaN	1.0	NaN	NaN
kmethod	NaN	NaN	NaN	1.00	0.77
tailind	NaN	NaN	NaN	NaN	NaN
blength	NaN	NaN	NaN	NaN	NaN
place	NaN	NaN	NaN	0.77	1.00
year	NaN	NaN	NaN	0.54	0.46
ADWBind	NaN	NaN	NaN	NaN	NaN
gonind	NaN	NaN	NaN	0.52	0.66
BMI	NaN	NaN	NaN	NaN	NaN

The strongest pairwise correlations involving categorical features are between - gender and femstate (-0.89) - killing method and place/group (0.77) - place/group and date of death (0.70)

2.1.2 ii)

[6]:
```python
pairwise_correlations[['day', 'year']].where(lambda x: (np.abs(x) >= 0.4))
```

[6]:

	day	year
day	1.00	0.40
weight	NaN	NaN
gender	NaN	NaN
liverind	NaN	NaN

```
heartind     NaN   NaN
append       NaN   NaN
femstate     NaN   NaN
gonfatind    NaN   NaN
batind       NaN   NaN
sulcer       NaN   NaN
kmethod     0.45  0.54
tailind      NaN   NaN
blength      NaN   NaN
place       0.70  0.46
year        0.40  1.00
ADWBind      NaN   NaN
gonind      0.44   NaN
BMI          NaN  0.43
```

The strongest pairwise correlations involving temporal features day and year are between - date of death and place/group (0.70) - year of death and killing method (0.54)

2.1.3 iii)

[7]:
```
pairwise_correlations[['weight', 'liverind', 'heartind', 'append', 'gonfatind',␣
↪'batind',
                        'tailind', 'blength', 'ADWBind', 'gonind', 'BMI']].
↪where(lambda x: (np.abs(x) >= 0.4))
```

[7]:
```
           weight  liverind  heartind  append  gonfatind  batind  tailind  \
day           NaN       NaN       NaN     NaN        NaN     NaN      NaN
weight       1.00       NaN       NaN     NaN        NaN     NaN      NaN
gender        NaN       NaN       NaN     NaN        NaN     NaN      NaN
liverind      NaN       1.0       NaN     NaN        NaN     NaN      NaN
heartind      NaN       NaN       1.0     NaN        NaN     NaN      NaN
append        NaN       NaN       NaN    1.00       0.43     NaN      NaN
femstate      NaN       NaN       NaN     NaN        NaN     NaN      NaN
gonfatind     NaN       NaN       NaN    0.43       1.00     NaN      NaN
batind        NaN       NaN       NaN     NaN        NaN     1.0      NaN
sulcer        NaN       NaN       NaN     NaN        NaN     NaN      NaN
kmethod       NaN       NaN       NaN     NaN       0.49     NaN      NaN
tailind       NaN       NaN       NaN     NaN        NaN     NaN     1.00
blength      0.88       NaN       NaN     NaN        NaN     NaN    -0.42
place         NaN       NaN       NaN     NaN       0.57     NaN      NaN
year          NaN       NaN       NaN     NaN        NaN     NaN      NaN
ADWBind     -0.41       NaN       NaN     NaN        NaN     NaN      NaN
gonind       0.53      -0.4       NaN     NaN       0.68     NaN      NaN
BMI          0.88       NaN       NaN     NaN        NaN     NaN      NaN

           blength  ADWBind  gonind   BMI
day            NaN      NaN    0.44   NaN
```

```
weight        0.88   -0.41    0.53  0.88
gender         NaN     NaN     NaN   NaN
liverind       NaN     NaN   -0.40   NaN
heartind       NaN     NaN     NaN   NaN
appind         NaN     NaN     NaN   NaN
femstate       NaN     NaN     NaN   NaN
gonfatind      NaN     NaN    0.68   NaN
batind         NaN     NaN     NaN   NaN
sulcer         NaN     NaN     NaN   NaN
kmethod        NaN     NaN    0.52   NaN
tailind      -0.42     NaN     NaN   NaN
blength       1.00     NaN    0.44  0.60
place          NaN     NaN    0.66   NaN
year           NaN     NaN     NaN  0.43
ADWBind        NaN    1.00   -0.53   NaN
gonind        0.44   -0.53    1.00  0.50
BMI           0.60     NaN    0.50  1.00
```

The strongest pairwise correlations involving other numerical features than day and year are between - weight and body mass index (0.88) - normalized gonadal fat weight and gonad fat index (0.68) - body length and body mass index (0.60) - place/group and gonad fat index (0.66)

2.2 b)

[8]:
```python
# Remove outliers
dataB = data.set_index('id').drop(['rat2', 'rat53', 'rat120','rat434'])
```

[9]:
```python
# Calculate pairwise correlations
pairwise_correlationsB = dataB.corr(method='pearson').round(decimals=2)
```

[10]:
```python
# Compare the correlations involving heartind before and after removing the␣
↪outliers
change = pd.concat([pairwise_correlations['heartind'].rename('heartind,␣
↪before'),
                    pairwise_correlationsB['heartind'].rename('heartind,␣
↪after')],
                   axis=1)
change
```

[10]:
```
           heartind, before  heartind, after
day                   -0.01            -0.45
weight                -0.01            -0.49
gender                 0.06             0.26
liverind              -0.04             0.23
heartind               1.00             1.00
appind                -0.06            -0.03
femstate              -0.05            -0.23
```

```
gonfatind                0.01             -0.30
batind                  -0.04              0.17
sulcer                   0.13              0.20
kmethod                  0.01             -0.35
tailind                 -0.04             -0.12
blength                  0.02             -0.42
place                    0.01             -0.47
year                    -0.05             -0.41
ADWBind                 -0.06              0.31
gonind                   0.02             -0.55
BMI                     -0.03             -0.55
```

The correlation between heartind and

- day
- weight
- blength
- place
- year
- gonind
- BMI

changed a lot from weak to strong.

There were some other big changes too but they are not mentioned because the absolute after values were not greater than 0.4. All the correlations involving heartind had weak correlation before.

[11]:
```
# Compare the correlations involving liverind before and after removing the␣
↪outliers
change = pd.concat([pairwise_correlations['liverind'].rename('liverind,␣
↪before'),
                    pairwise_correlationsB['liverind'].rename('liverind,␣
↪after')],
                   axis=1)
change
```

[11]:
```
           liverind, before  liverind, after
day                   -0.12            -0.18
weight                -0.24            -0.24
gender                -0.21            -0.25
liverind               1.00             1.00
heartind              -0.04             0.23
appind                 0.30             0.09
femstate               0.16             0.16
gonfatind             -0.26            -0.32
batind                 0.09             0.01
sulcer                -0.04            -0.02
kmethod               -0.32            -0.40
tailind                0.09             0.08
```

```
blength                 -0.19            -0.24
place                   -0.31            -0.38
year                    -0.07            -0.12
ADWBind                  0.33             0.46
gonind                  -0.40            -0.48
BMI                     -0.24            -0.19
```

The correlation between liverind and

- kmethod

changed a little from weak to strong.

The correlation between liverind and

- ADWBind

changed a lot from weak to strong.

There were some other big changes too but they are not mentioned because the absolute after values were not greater or equal than 0.4. Most of the correlation changes were less than 0.1 though.

[12]:
```
# Calculate the mean absolute difference after removing the outliers
ma_heartind = np.mean(np.abs(pairwise_correlations['heartind'] -␣
 ↪pairwise_correlationsB['heartind']), axis=0)
ma_liverind = np.mean(np.abs(pairwise_correlations['liverind'] -␣
 ↪pairwise_correlationsB['liverind']), axis=0)
ma_heartind, ma_liverind
```

[12]:
```
(0.2983333333333333, 0.07)
```

The mean absolute differences tells us that the correlations involving heartind changed by 0.30 on average and the correlations involving liverind changed by 0.07 on average.

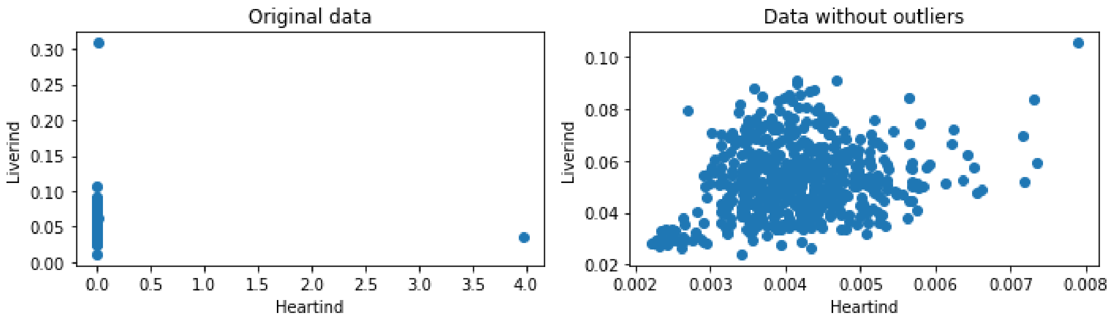
By removing the outliers we decreased the variability of the data and this increased the statistical power. The pearson correlation coefficient measures the strength and direction of a linear relationship between two variables. The removed outliers shifted the data mean too much and by removing those the mean now better described the sample. This is why the changes were quite big on some of the correlations. (See the figure below as an additional illustration of this.)

[13]:
```
fig, ax = plt.subplots(1, 2, figsize=(10,3));

ax[0].scatter(data.heartind, data.liverind)
ax[0].set_title('Original data')
ax[0].set_xlabel('Heartind')
ax[0].set_ylabel('Liverind')

ax[1].scatter(dataB.heartind, dataB.liverind)
ax[1].set_title('Data without outliers')
ax[1].set_xlabel('Heartind')
ax[1].set_ylabel('Liverind')
```

```
fig.tight_layout()
```



2.3 c)

Continue with the data from b), where the listed outliers are removed. Change the special codes for the freezer rats: day=0 and year=-1

[14]:
```
dataC = dataB.copy()
dataC['day'].replace({400: 0}, inplace=True)
dataC['year'].replace({15: -1}, inplace=True)
# print the original correlations
dataC.head()
```

[14]:
```
      day  weight  gender  liverind  heartind    appind  femstate  gonfatind  \
id
rat1   67    22.0       1  0.036364  0.005091  0.013636         4   0.000000
rat3  230    37.5       1  0.051200  0.006133  0.020800         4   0.000000
rat4  261    45.0       2  0.047333  0.005556  0.105111        -1   0.293333
rat5  262    43.0       1  0.052558  0.005349  0.021628         4   0.000000
rat6  169    40.0       1  0.057250  0.006500  0.014000         4   0.000000

        batind  sulcer  kmethod   tailind  blength  place  year   ADWBind  \
id
rat1  0.001864       1        2  0.714286     10.5      1     5  0.454545
rat3  0.001467       2        1  0.904762     10.5      2     8  0.653333
rat4  0.002178       1        1  0.869565     11.5      4     2  0.260000
rat5  0.001884       1        1  0.720000     12.5      2     8  0.500000
rat6  0.002925       2        1  0.826087     11.5      2     7  0.575000

        gonind       BMI
id
rat1  0.000000  0.199546
rat3  0.000000  0.340136
rat4  2.653242  0.340265
```

```
rat5   0.000000   0.275200
rat6   0.000000   0.302457
```

How did the correlations involving either day or year change?

[15]:
```
pairwise_correlationsC = dataC.corr(method='pearson').round(decimals=2)
change = pairwise_correlationsB[['day', 'year']] -␣
 ↪pairwise_correlationsC[['day', 'year']]
change
```

[15]:
```
            day  year
day        0.00  0.11
weight     0.26  0.33
gender    -0.19 -0.26
liverind  -0.39 -0.51
heartind  -0.47 -0.60
appind     0.22  0.29
femstate   0.32  0.42
gonfatind  0.55  0.72
batind     0.23  0.28
sulcer    -0.11 -0.15
kmethod    0.98  1.27
tailind    0.16  0.20
blength    0.11  0.13
place      0.90  1.19
year       0.11  0.00
ADWBind   -0.31 -0.41
gonind     0.61  0.81
BMI        0.36  0.43
```

[16]:
```
np.abs(change).mean(axis=0)
```

[16]:
```
day     0.348889
year    0.450556
dtype: float64
```

All correlations involving day and year had a really strong change after the special code modifications. On average the changes were 0.35 on correlations involving day and 0.45 increase on correlations involving year.

What happens if you remove all freezer rats?

[17]:
```
dataC_first_part = dataC.copy()
pairwise_correlationsC_first_part = pairwise_correlationsC.copy()
dataC['day'].replace({0: np.nan}, inplace=True)
dataC['year'].replace({-1: np.nan}, inplace=True)
dataC = dataC.dropna(axis=0, how='any')
dataC
```

[17]:

```
          day  weight  gender  liverind  heartind    appind  femstate  \
id
rat1     67.0    22.0       1  0.036364  0.005091  0.013636         4
rat3    230.0    37.5       1  0.051200  0.006133  0.020800         4
rat4    261.0    45.0       2  0.047333  0.005556  0.105111        -1
rat5    262.0    43.0       1  0.052558  0.005349  0.021628         4
rat6    169.0    40.0       1  0.057250  0.006500  0.014000         4
…         …     …     …      …       …        …         …
rat574  127.0   442.0       1  0.067738  0.003507  0.013552         3
rat575  269.0   381.8       1  0.078575  0.003667  0.014667         4
rat576  169.0   416.0       1  0.079303  0.003918  0.014760         1
rat577  244.0   476.0       1  0.080693  0.004286  0.008803         2
rat578  230.0   459.0       1  0.085251  0.003682  0.016144         2

        gonfatind    batind  sulcer  kmethod   tailind  blength  place  year  \
id
rat1     0.000000  0.001864       1        2  0.714286     10.5      1   5.0
rat3     0.000000  0.001467       2        1  0.904762     10.5      2   8.0
rat4     0.293333  0.002178       1        1  0.869565     11.5      4   2.0
rat5     0.000000  0.001884       1        1  0.720000     12.5      2   8.0
rat6     0.000000  0.002925       2        1  0.826087     11.5      2   7.0
…             …        …     …       …        …        …     …    …
rat574   0.010656  0.002602       1        1  0.808163     24.5      2   7.0
rat575   0.004976  0.001035       1        1  0.777273     22.0      2   6.0
rat576   0.004087  0.001337       1        1  0.760870     23.0      2   7.0
rat577   0.007437  0.001592       1        1  0.714286     24.5      2   8.0
rat578   0.001089  0.001218       1        1  0.816327     24.5      2   8.0

         ADWBind    gonind       BMI
id
rat1    0.454545  0.000000  0.199546
rat3    0.653333  0.000000  0.340136
rat4    0.260000  2.653242  0.340265
rat5    0.500000  0.000000  0.275200
rat6    0.575000  0.000000  0.302457
…            …         …         …
rat574  0.372172  1.742219  0.736360
rat575  0.381090  1.064711  0.788843
rat576  0.469952  0.993252  0.786389
rat577  0.380252  1.512927  0.793003
rat578  0.517429  0.405465  0.764681

[525 rows x 18 columns]
```

[18]:

```python
pairwise_correlationsC = dataC.corr(method='pearson').round(decimals=2)
change = pairwise_correlationsC_first_part[['day', 'year']] -␣
 ↪pairwise_correlationsC[['day', 'year']]
```

```
change
```

[18]:

```
             day  year
day         0.00  0.30
weight     -0.11 -0.18
gender      0.11  0.11
liverind    0.17  0.17
heartind    0.27  0.33
appind     -0.12 -0.14
femstate   -0.16 -0.20
gonfatind  -0.28 -0.26
batind     -0.04 -0.07
sulcer      0.08  0.12
kmethod    -0.23 -0.24
tailind    -0.11 -0.16
blength    -0.04 -0.03
place      -0.72 -0.25
year        0.30  0.00
ADWBind     0.15  0.17
gonind     -0.32 -0.33
BMI        -0.16 -0.29
```

[19]:

```
np.abs(change).mean(axis=0)
```

[19]:

```
day     0.187222
year    0.186111
dtype: float64
```

After removing the freezer rats the both correlations involving day or year change by 0.19 on average. So the change is again quite strong on average. Most of the pairwise correlations involving either of these two have a strong change after removing the freezer rats.

[20]:

```
change = pairwise_correlationsB[['day', 'year']] -
 ↪pairwise_correlationsC[['day', 'year']]
change
```

[20]:

```
             day  year
day         0.00  0.41
weight      0.15  0.15
gender     -0.08 -0.15
liverind   -0.22 -0.34
heartind   -0.20 -0.27
appind      0.10  0.15
femstate    0.16  0.22
gonfatind   0.27  0.46
batind      0.19  0.21
sulcer     -0.03 -0.03
```

```
kmethod    0.75  1.03
tailind    0.05  0.04
blength    0.07  0.10
place      0.18  0.94
year       0.41  0.00
ADWBind   -0.16 -0.24
gonind     0.29  0.48
BMI        0.20  0.14
```

[21]:
```
np.abs(change).mean(axis=0)
```

[21]:
```
day     0.195000
year    0.297778
dtype: float64
```

Same thing happens if we compare the data after frerexer rat removal to data afteer b part. On average 0.20 and 0.30 changes in correlations and most of the correlations having big change (>0.10).

What is your conclusion on the changing correlations involving day or year? When changing the correlations we can have a great impact on what the data set tells us. One should be careful by making any quick interpretations based on correlations solely, as they can be manipulated using predetermined values for the values that don't fit the type of the data. For example the original freezer rat special codes here didn't fit the otherwise temporally structured feature spaces columns day and year and by manipulating its special codes did have an impact on the correlations involving these two features.

The changes in year caused a bigger change on average in correlations involving year. This is because year is a interval scale numerical temporal feature. THe circular numerical day didn't seem to have such as big change even though the change was also strong.

2.4 e)

Continue with the data where all freezer rats are removed. Test changing codes of categorical features femstate, kmethod and place. Can you generate any big changes in correlations involving these features?

[22]:
```
dataE = dataC.copy()
# Change the spacial code for male rats from -1 to 1000
dataE['femstate'].replace({-1: 1000}, inplace=True)
# Change the special codes for kmethods from (1, 2, 6) to (10, 122, 123)
dataE['kmethod'].replace({1: 10, 2: 122, 6: 123}, inplace=True)
# Change the special codes for place from (1, 2, 3, 4, 5, 6, 7, 8, 9) to
# (1, 2, 3, 101, 102, 103, 104, 105, 106)
dataE['place'].replace({4: 101, 5: 102, 6: 103, 7: 104, 8: 105, 9: 106},␣
↪inplace=True)

pairwise_correlationsE = dataE.corr(method='pearson').round(decimals=2)
```

[23]:
```
change = pairwise_correlationsC[['femstate', 'kmethod', 'place']] -␣
 ↪pairwise_correlationsE[['femstate', 'kmethod', 'place']]
change
```

[23]:

	femstate	kmethod	place
day	0.27	0.00	0.17
weight	-0.02	0.00	0.07
gender	-1.89	0.00	-0.02
liverind	0.67	0.00	-0.02
heartind	-0.31	0.00	-0.02
appind	0.27	0.00	-0.11
femstate	0.00	-0.20	0.23
gonfatind	-0.15	0.00	0.06
batind	-0.03	0.00	-0.22
sulcer	-0.05	0.00	-0.08
kmethod	-0.20	0.00	-0.09
tailind	0.13	0.00	0.06
blength	0.04	0.00	0.05
place	0.23	-0.09	0.00
year	-0.03	0.00	0.11
ADWBind	0.65	0.00	-0.14
gonind	-0.29	0.00	0.19
BMI	0.00	0.00	0.09

[24]:
```
np.abs(change.mean(axis=0))
```

[24]:
```
femstate    0.039444
kmethod     0.016111
place       0.018333
dtype: float64
```

By changing the codes of the categorical features by completely random and ridiculously large values, the correlations involving these features didn't change much on average. This is beacuse of the categorical structure of the variables.

2.5 f)

What is your final conclusion, which correlations were reliable? The correlations reliable are the ones that don't change much even if we were to manipulate the feature spaces manually. These seem to include the categorical variables especially. And also the correlatoions where the outliers were removed were reliable.

Are there any strong correlations showing a linear trend? Note:

I think the exercise description is a bit unclear here. Should I use the original data or the data of the previous part here? I think that because of this: "After each analysis step, you are asked to analyze changes to the previous step." I should use the data of the previous step.

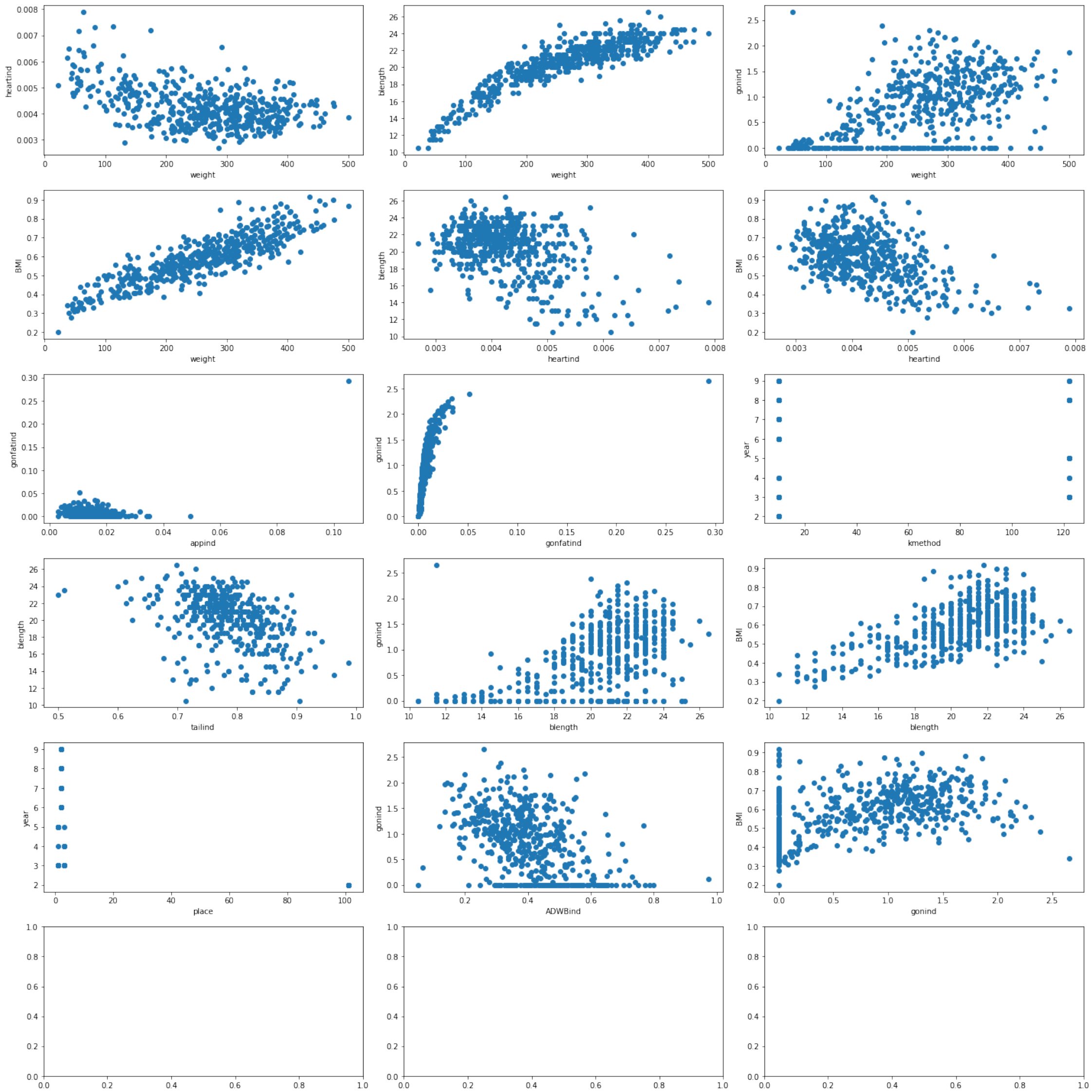
[25]:
```
# Find the strong pairwise correlations
pairs = []
for row in pairwise_correlationsE.index.values:
    for column in pairwise_correlationsE.columns.values:
        x = pairwise_correlationsE.loc[row, column]
        if np.abs(x) >= 0.4 and x != 1:
            if (column, row) not in pairs:
                pairs.append((row, column))
pairs
```

[25]:
```
[('weight', 'heartind'),
 ('weight', 'blength'),
 ('weight', 'gonind'),
 ('weight', 'BMI'),
 ('heartind', 'blength'),
 ('heartind', 'BMI'),
 ('append', 'gonfatind'),
 ('gonfatind', 'gonind'),
 ('kmethod', 'year'),
 ('tailind', 'blength'),
 ('blength', 'gonind'),
 ('blength', 'BMI'),
 ('place', 'year'),
 ('ADWBind', 'gonind'),
 ('gonind', 'BMI')]
```

[26]:
```
a = (len(pairs) // 3 + 1)
fig, axs = plt.subplots(a, 3, figsize=(20, 20))

for i in range(len(pairs)):
    r = i//3
    c = i%3
    xdata = dataE[pairs[i][0]]
    ydata = dataE[pairs[i][1]]
    axs[r,c].scatter(xdata, ydata)
    axs[r,c].set_xlabel(pairs[i][0])
    axs[r,c].set_ylabel(pairs[i][1])

fig.tight_layout()
```



So some of the strong correlations show a linear trend more clearly and some don't. The clearest linear trend is between BMI and weight.

```
[ ]:
```