2 Task 2 (Extra)

First,we will get the clean dataset from Task 1 after removing the listed outliers but keeping only wild rats (place 1-3).Then we convert day feature into circular form using $\theta = day/365 * 2\pi$.Then we can calculate the Pearson correlation coefficient as the basis of the R^2 correlation shown in Fig.3. With the function written in Appendix B in Task document, we can calculate the R^2 of (day,gonfatind) and (day,batind),which are $R_g^2 = 0.017, R_g = 0.132$ and $R_b^2 = 0.232, R_b = 0.486$.

```

# convert and get related data features
data_task2["day_c"]= data_task2["day"]/365*2*np.pi
data_task2["day_sin"]= np.sin(data_task2["day_c"])
data_task2["day_cos"]= np.cos(data_task2["day_c"])
data_task2=data_task2[["day","day_c","day_sin","day_cos","gonfatind","batind"]]

# Compute correlations
corr_task2=data_task2.corr(method='pearson').round(3)


# numbers are from correlation matrix

# day--gonfatind
R2_g=((-0.087)**2+(-0.116)**2-2*(-0.087*-0.116*0.221))/(1-0.221**2)
print(R2_g)
print(np.sqrt(R2_g))

#day --batind
R2_b=((0.341)**2+(0.407)**2-2*(0.341*0.407*0.221))/(1-0.221**2)
print(R2_b)
print(np.sqrt(R2_b))
```

	day	day_c	day_sin	day_cos	gonfatind	batind
day	1.000	1.000	-0.826	-0.468	0.124	-0.441
day_c	1.000	1.000	-0.826	-0.468	0.124	-0.441
day_sin	-0.826	-0.826	1.000	0.221	-0.116	0.407
day_cos	-0.468	-0.468	0.221	1.000	-0.087	0.341
gonfatind	0.124	0.124	-0.116	-0.087	1.000	-0.058
batind	-0.441	-0.441	0.407	0.341	-0.058	1.000

Figure 3: Day Related Matrix

From the indices I computed above, when turning day feature into circular form,we know there is no strong correlation between the time of a year and gofatind, but it shows strong correlation between the time of a year and batind.