TO8 comparison of tumour burden at cull with hematological analysis

Section 1 Data Preparation

Most of this first part is data preparation, reading in the data, making it in a format that the algorithms can process and making items numerical so they can be used in analysis (e.g DKO will correspond to 1 and Control to 0, Zol also will be 1 and Control 0 etc)

So you can skip until section 2 to be able to go through the analysis.

The analysis was done on tumour-bearing animals and the flux [p/s] of their ex-vivo imaging of the hind limbs only was used. There are some mice which had significant tumours in their spine but this has not been accounted for in this analysis as the previous analysis focused on hind-limb tumours also.

In [1]:

```python
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
%matplotlib inline
```

In [2]:

```python
data = pd.read_csv('TO8-Hematological-Analysis.csv')
```

In [3]:

```python
df = pd.DataFrame(data)
```

In [4]:

```python
df.head()
```

Out[4]:

	Mouse	Cells	Treatment	Hind-Limb Tumour	Signal	Tumour	Total Flux [p/s]	WBC 10^3/mm^3	LYM %	MON %
0	1	DKO	Zol	No	N	0	23500000.0	2.2	62.5	10.2
1	2	DKO	Zol	Yes	W	1	698000000.0	2.1	77.1	4.9
2	3	DKO	PBS	No	N	0	150000000.0	2.3	71.2	8.3
3	4	DKO	PBS	Yes	S	1	105000000.0	3.4	80.6	6.0
4	5	DKO	Zol	No	N	0	3110000.0	3.8	71.6	4.9

5 rows × 25 columns

In [8]:

```
df
```

Out[8]:

	Mouse	Cells	Treatment	Hind-Limb Tumour	Tumour	Total Flux [p/s]	WBC 10^3/mm^3	LYM %	MON %	GRA %
0	1	1	1	0	0	23500000.0	2.2	62.5	10.2	27.3
1	2	1	1	1	1	698000000.0	2.1	77.1	4.9	18.0
2	3	1	0	0	0	150000000.0	2.3	71.2	8.3	0.5
3	4	1	0	1	1	105000000.0	3.4	80.6	6.0	13.4
4	5	1	1	0	0	3110000.0	3.8	71.6	4.9	23.5
5	6	1	1	1	1	133000.0	3.0	73.1	6.5	20.4
6	7	1	0	0	0	863000.0	3.5	83.5	7.3	9.2
7	8	1	0	1	1	58000.0	2.1	72.4	8.6	19.0
8	9	1	0	0	0	38100000.0	2.0	76.3	8.3	15.4
9	10	1	0	1	1	26800000.0	2.1	68.2	8.4	23.4
10	11	1	1	1	1	171000000.0	2.1	54.8	13.8	31.4
11	12	1	1	1	1	262000000.0	2.8	74.8	9.4	15.8
12	13	1	1	1	1	151000000.0	2.4	69.1	6.5	24.4
13	14	1	1	1	1	141000000.0	1.5	76.4	6.8	16.8
14	15	1	0	0	0	193000.0	1.2	76.1	5.6	18.3
15	16	1	0	1	1	146000.0	2.0	71.8	10.2	18.0
16	17	0	1	1	1	10900000.0	1.5	91.5	3.0	5.5
17	18	0	1	1	1	368000000.0	1.8	80.7	3.1	16.2
18	19	0	0	1	1	16100000.0	3.1	70.9	6.3	22.8
19	20	0	0	1	1	471000.0	2.2	70.2	10.4	19.4
20	21	0	1	0	0	471000000.0	1.3	83.2	6.8	10.0
21	22	0	0	1	1	455000000.0	3.1	79.4	2.1	18.5
22	23	0	1	0	0	552000.0	1.9	72.2	6.2	21.6
23	24	0	0	1	1	1170000.0	0.3	93.2	2.2	4.6
24	25	0	0	1	1	1070000.0	0.7	92.2	2.5	5.3
25	26	0	1	1	1	3420000.0	1.7	67.9	6.9	25.2
26	27	0	1	1	1	605000000.0	0.8	86.6	4.3	9.1
27	29	0	1	1	1	703000000.0	1.8	68.9	12.3	18.8
28	30	0	1	0	0	118000.0	1.0	82.3	7.8	9.9
29	32	0	0	0	1	156000.0	1.6	70.0	3.5	26.5

30 rows × 24 columns

In [11]:

```
df.drop('Hind-Limb Tumour', inplace=True,axis=1)
```
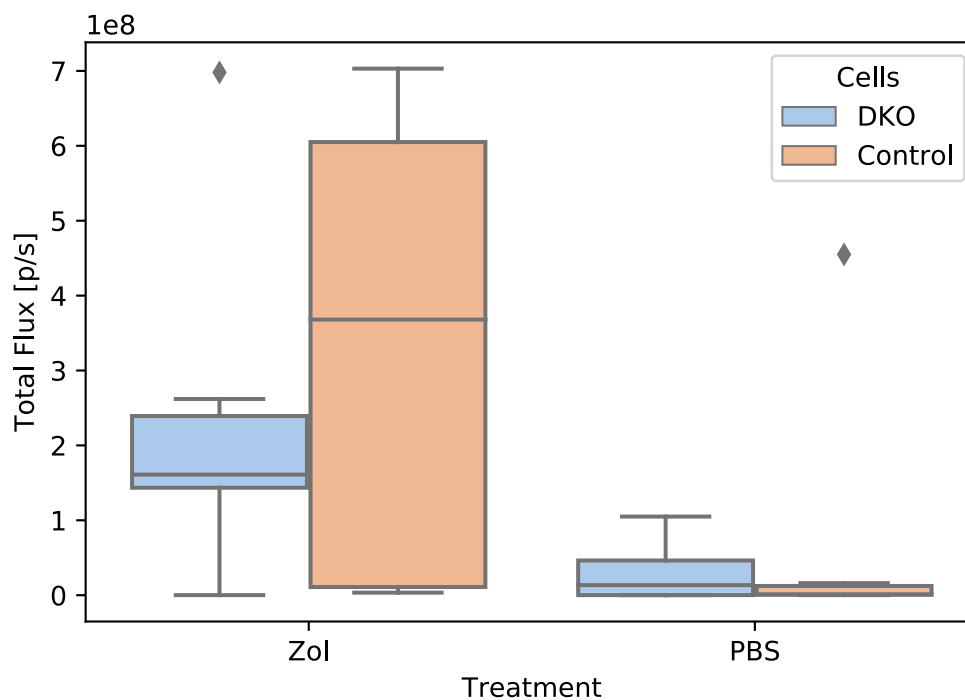
Section 2 - Analysis

I started off by charting the distribution of the total flux compared to cells and treatment

In [51]:

```
sns.boxplot(data=df_tumour,y='Total Flux [p/s]',x='Treatment',hue='Cells',palette='pastel')
```

Out[51]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x21120785308>
```


In [ ]:

```
It seems that the PBS treated has a generally lower tumour burden of those that developed tumours.
```

In [ ]:

```
sns.pairplot(df_num, hue='Total Flux [p/s]')
```

In [14]:

```
df_tumour = df[df['Tumour']==1]
```

In [24]:

```
df_tumour['Cells'].replace({0:'Control',1:'DKO'}, inplace=True)
```

The next thing I wanted to see was the spread between DKO and Control cells using a violinplot overlaid with points. This is shown in the figure below. This shows that the distribution of the two cell types is similar. The DKO cells have a higher average but lower spread of tumour burden, but the control cells have a lower average but much larger spread.

In [49]:

```
sns.boxplot(data=df_tumour,x='Cells',y='Total Flux [p/s]',palette='pastel')
plt.show()
```
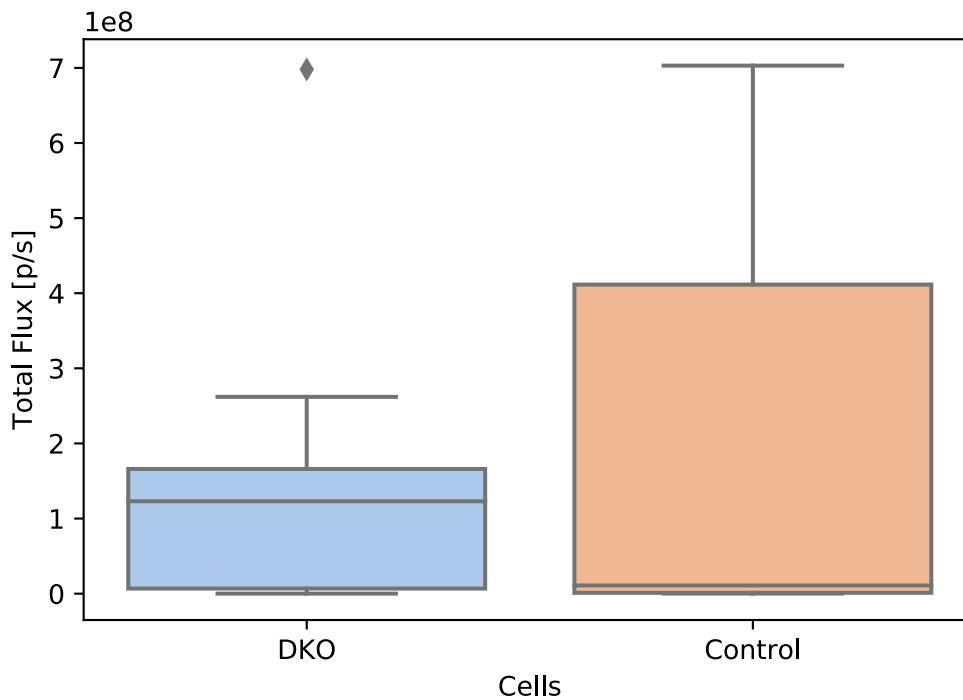

So I decided to plot the total flux against all other columns as shown below. This would be much clearer. I coloured them by treatment, and also set a large size for DKO cells and small size for control cells as can be seen in the legends of the plots below.

There seems to be a general non-pattern spread between the cell types and treatments against tumour burden.

All the axes on the left are the same to the power (1-e8)

In [62]:

```python
for column in df_tumour.columns[5:]:
    plt.figure(figsize=(15,15))
    sns.relplot(x=column,y='Total Flux [p/s]',data=df_tumour,hue='Treatment',size='Cell
s',alpha=0.8,palette='pastel')
    filename = 'TO8-Tumour_Analyses/' + column.replace(' ', '-').replace('/','-') + '-F
inal_Scan.png'
    plt.title('Tumour burden (flux) as compared to '+ column,loc='right',fontdict={'hor
izontalalignment':'right'})
    plt.savefig(fname=filename)
    plt.show()
```

```
<Figure size 1080x1080 with 0 Axes>
```


Tumour burden (flux) as compared to WBC 10^3/mm^3

```
<Figure size 1080x1080 with 0 Axes>
```

Tumour burden (flux) as compared to LYM %

<Figure size 1080x1080 with 0 Axes>

Tumour burden (flux) as compared to MON %

<Figure size 1080x1080 with 0 Axes>

Tumour burden (flux) as compared to GRA %


```
<Figure size 1080x1080 with 0 Axes>
```

Tumour burden (flux) as compared to EOS %


```
<Figure size 1080x1080 with 0 Axes>
```
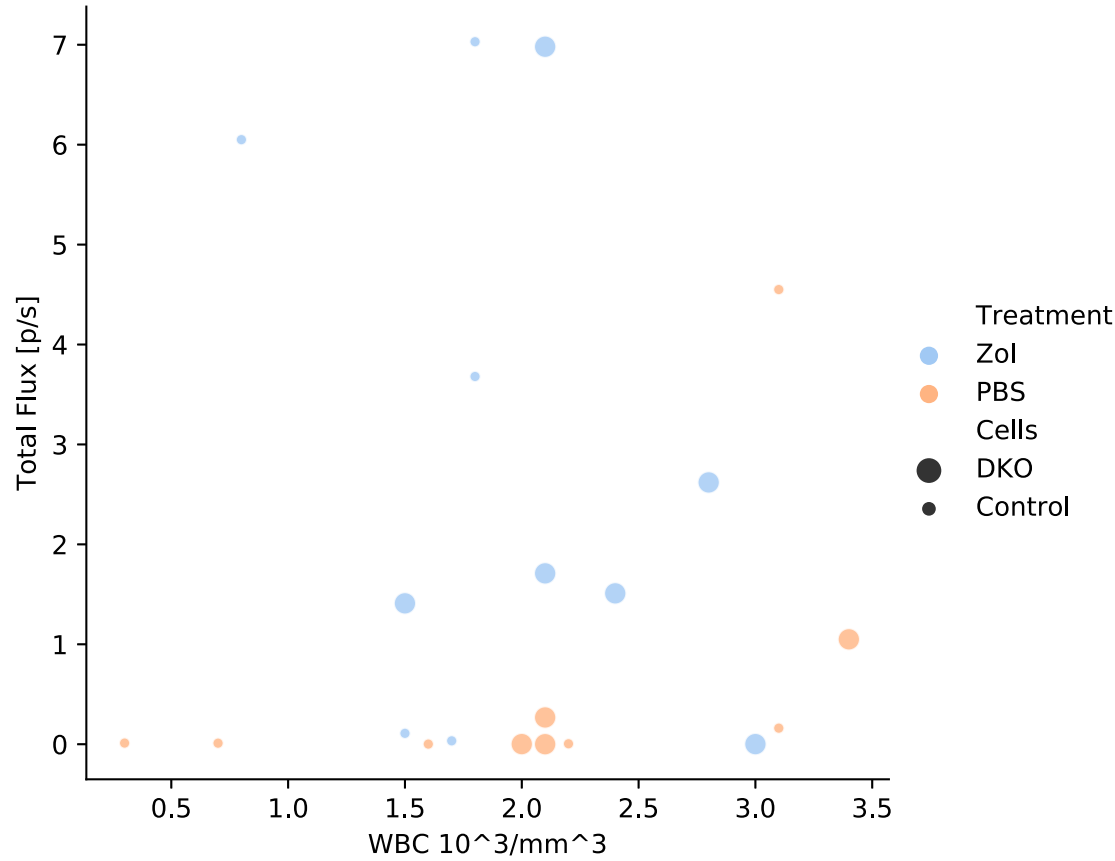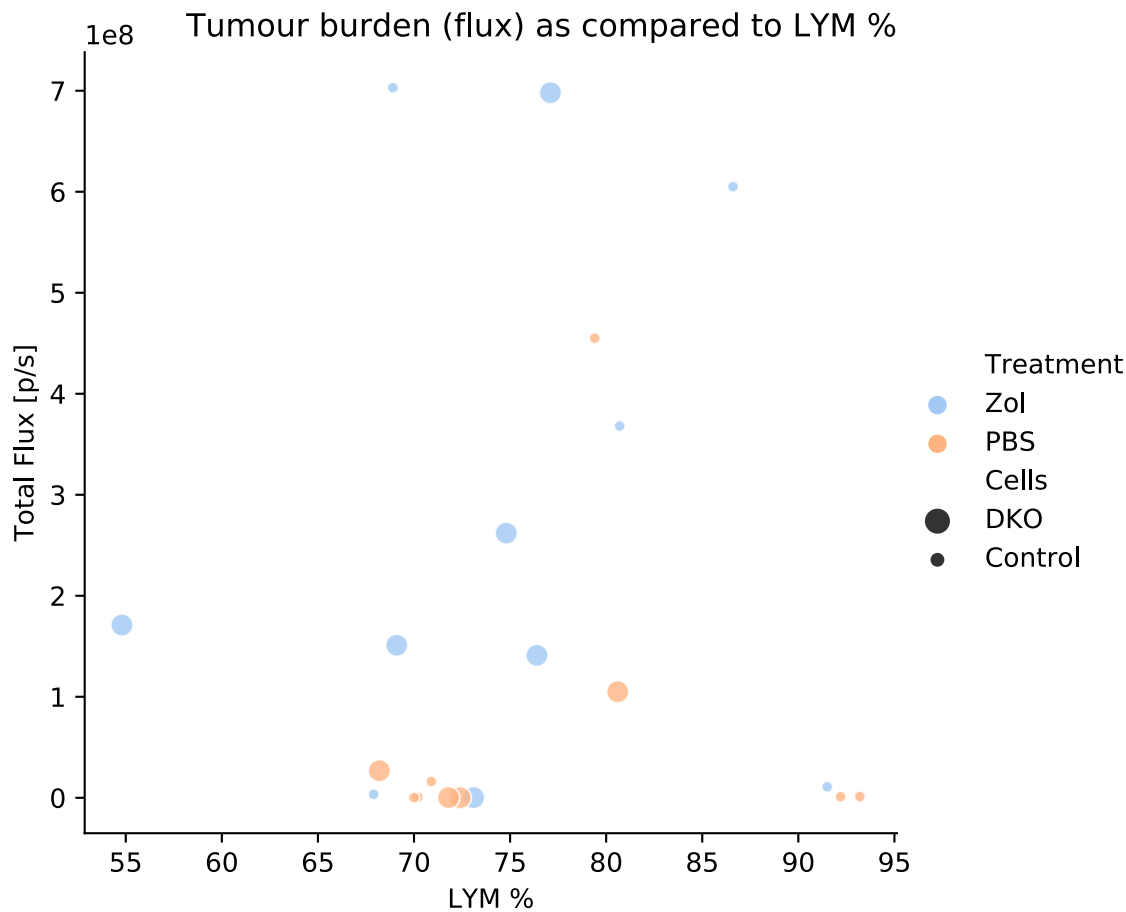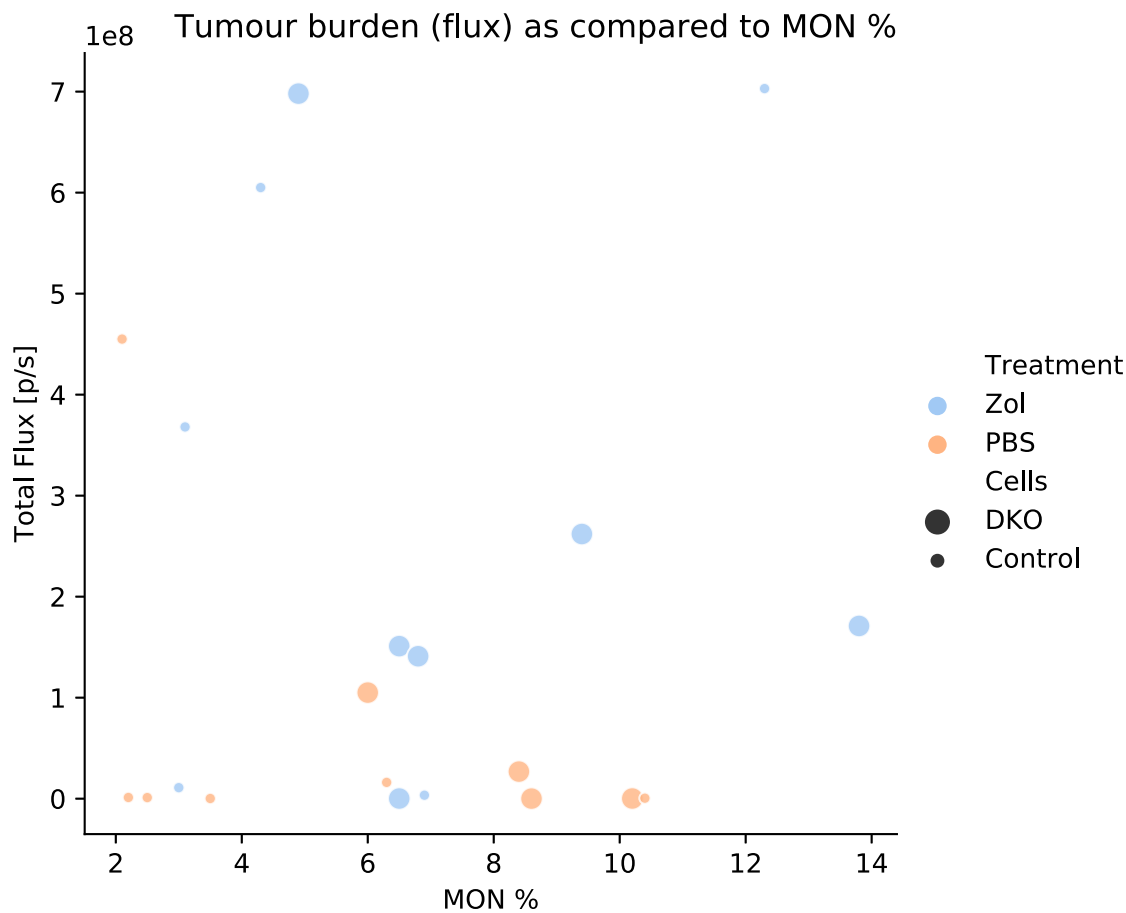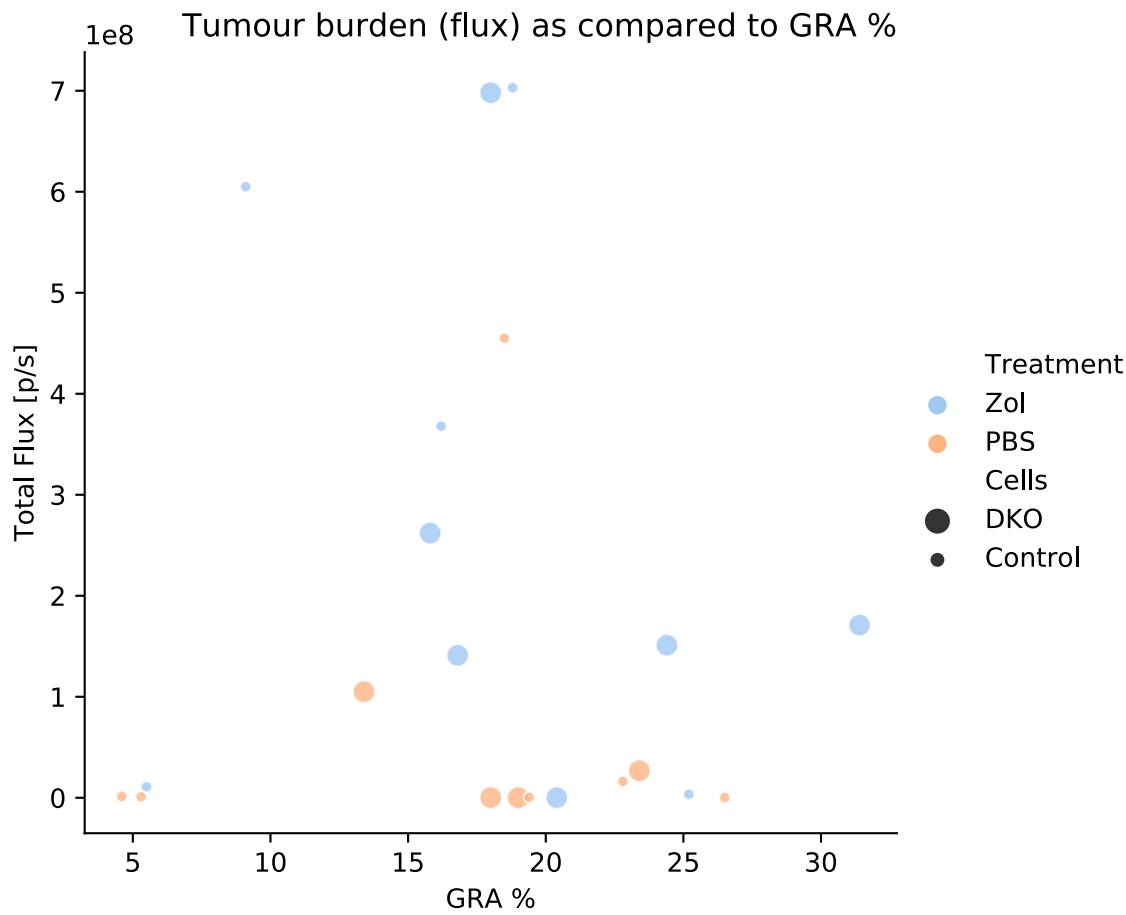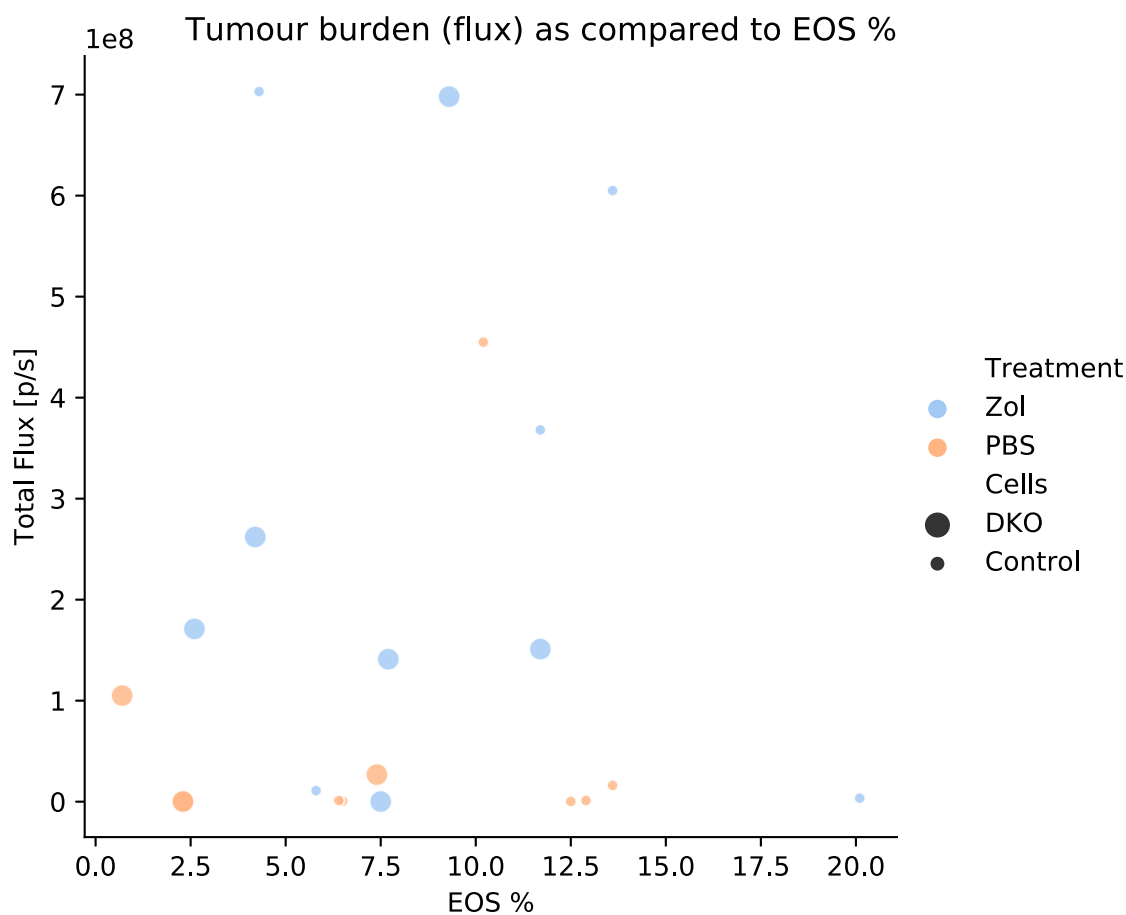
Tumour burden (flux) as compared to LYM # 10^3/mm^3


```
<Figure size 1080x1080 with 0 Axes>
```
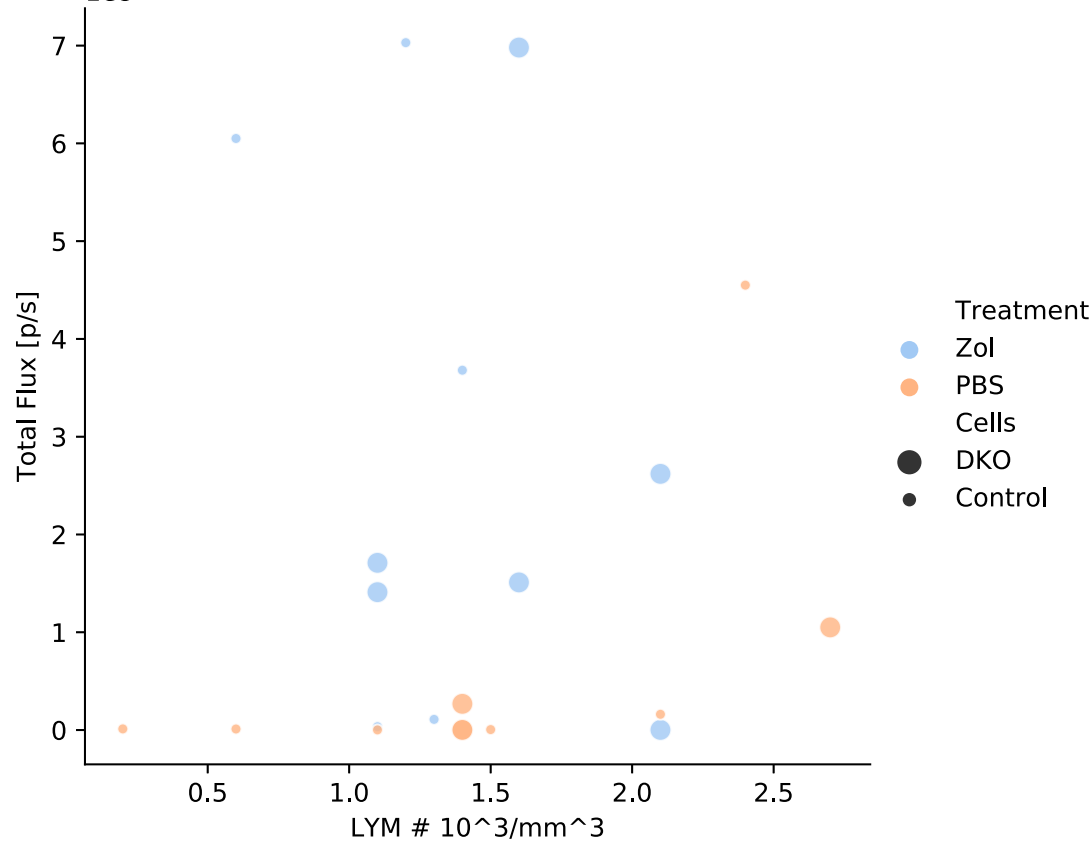
Tumour burden (flux) as compared to MON # 10^3/mm^3


```
<Figure size 1080x1080 with 0 Axes>
```
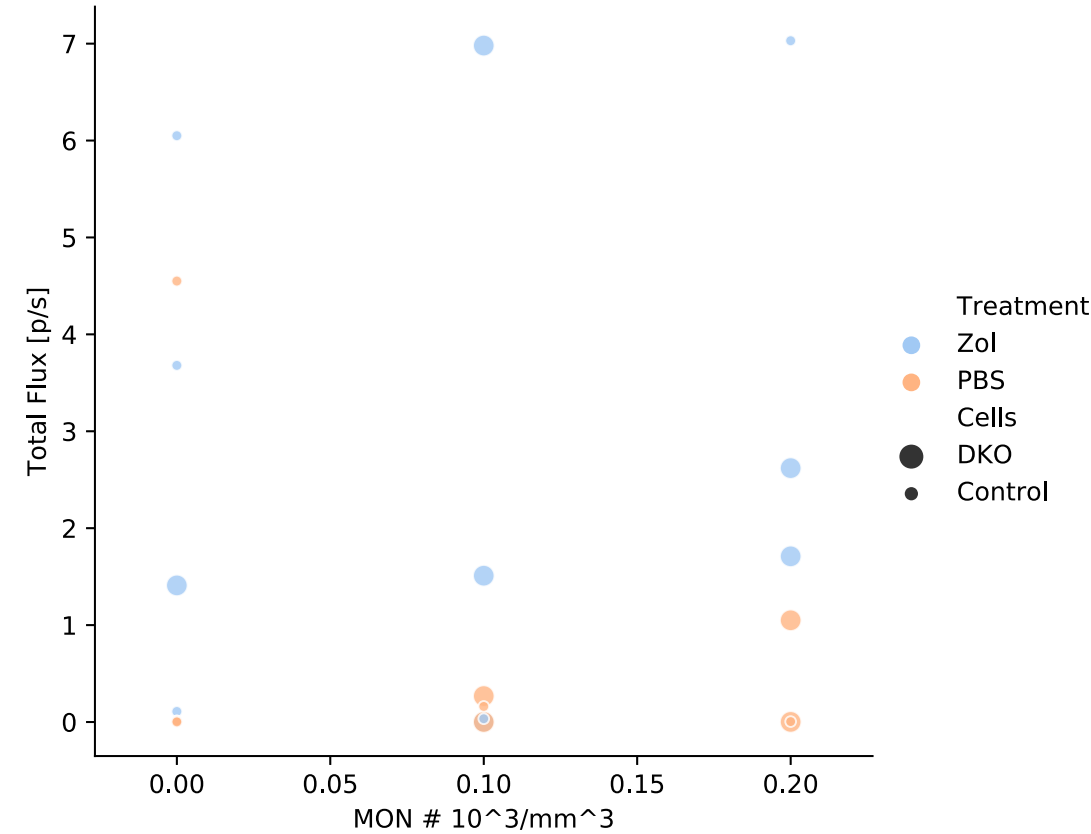
Tumour burden (flux) as compared to GRA # 10^3/mm^3


```
<Figure size 1080x1080 with 0 Axes>
```
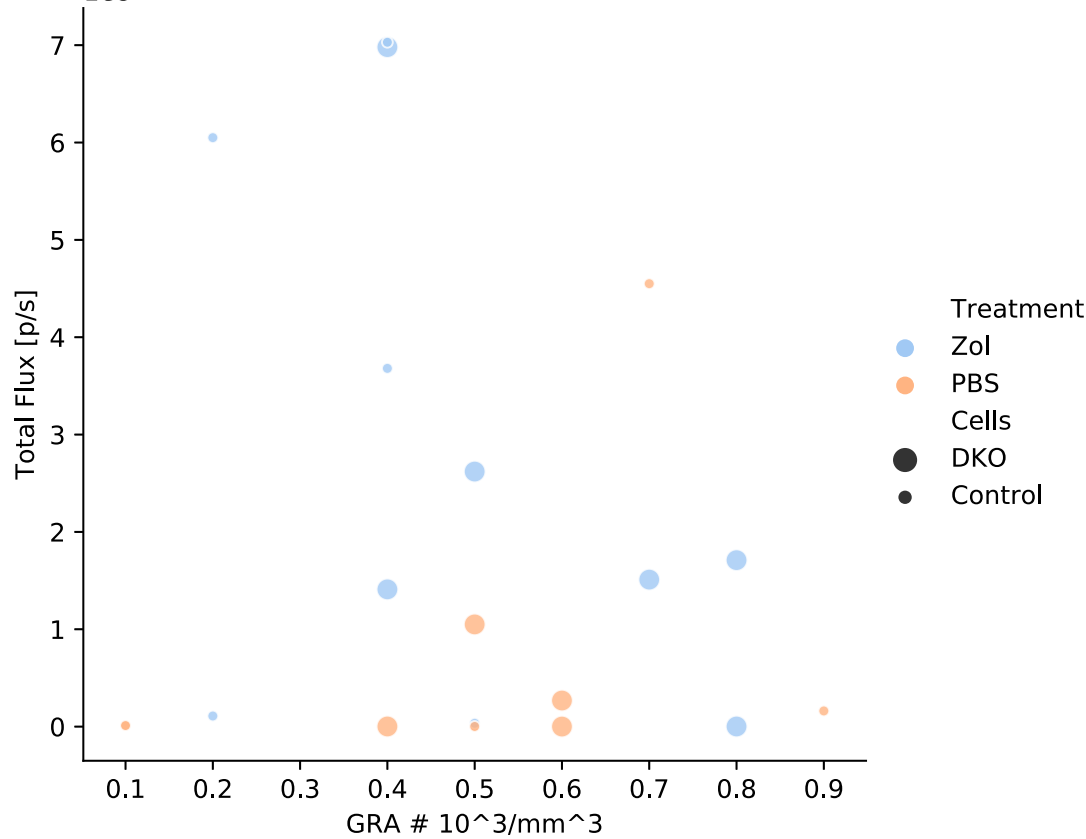
Tumour burden (flux) as compared to EOS # 10^3/mm^3


```
<Figure size 1080x1080 with 0 Axes>
```
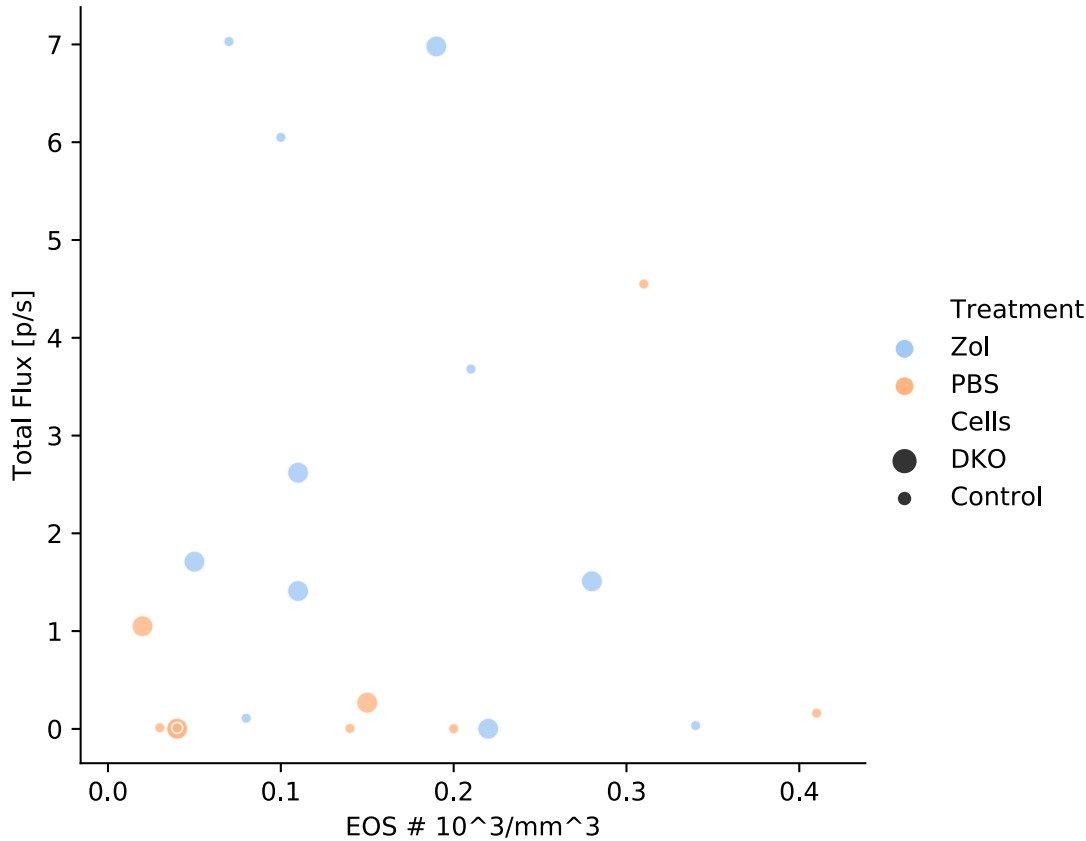
Tumour burden (flux) as compared to RBC 10^6/mm^3


```
<Figure size 1080x1080 with 0 Axes>
```

Tumour burden (flux) as compared to HGB g/dL
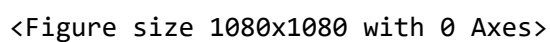


```
<Figure size 1080x1080 with 0 Axes>
```
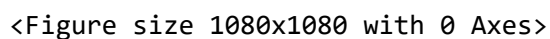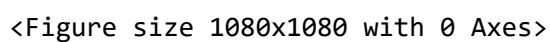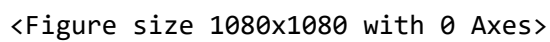
Tumour burden (flux) as compared to HCT %

<Figure size 1080x1080 with 0 Axes>

Tumour burden (flux) as compared to MCV um^3

<Figure size 1080x1080 with 0 Axes>

Tumour burden (flux) as compared to MCH pg

<Figure size 1080x1080 with 0 Axes>

Tumour burden (flux) as compared to MCHC g/dL

<Figure size 1080x1080 with 0 Axes>

Tumour burden (flux) as compared to RDW %

<Figure size 1080x1080 with 0 Axes>

Tumour burden (flux) as compared to PLT 10^3/mm^3

<Figure size 1080x1080 with 0 Axes>
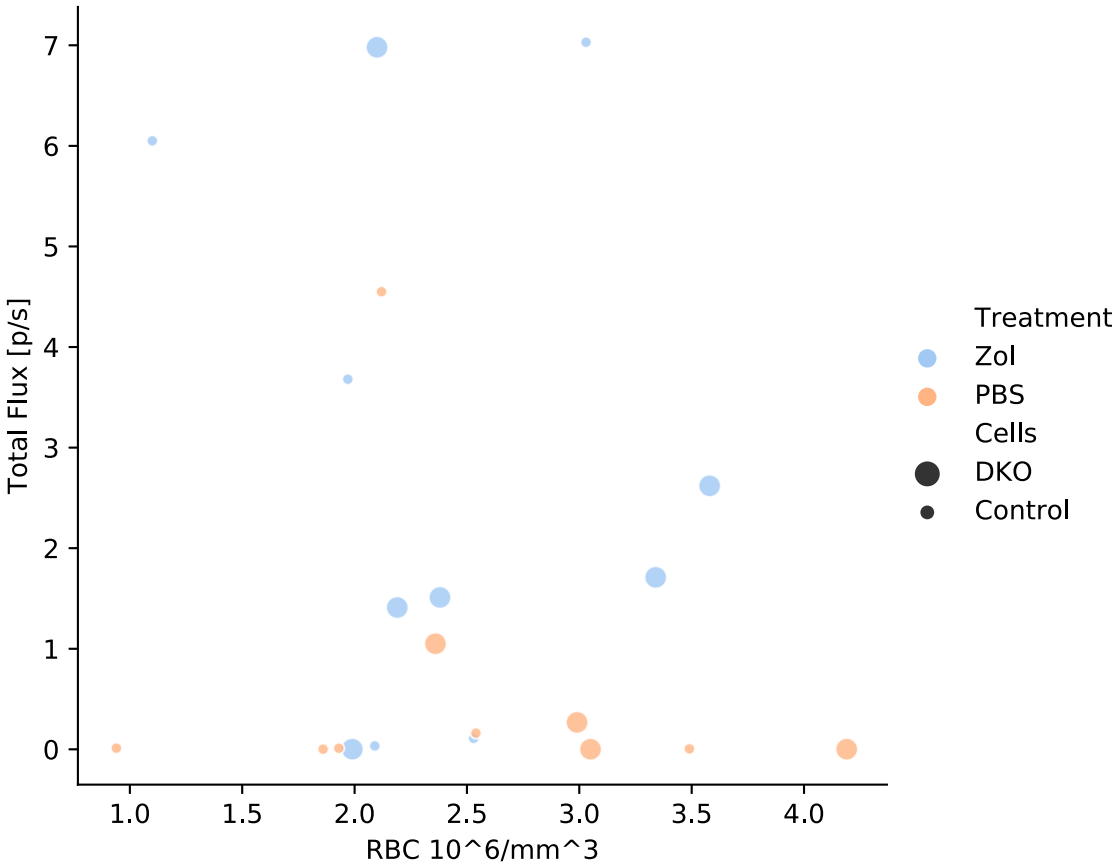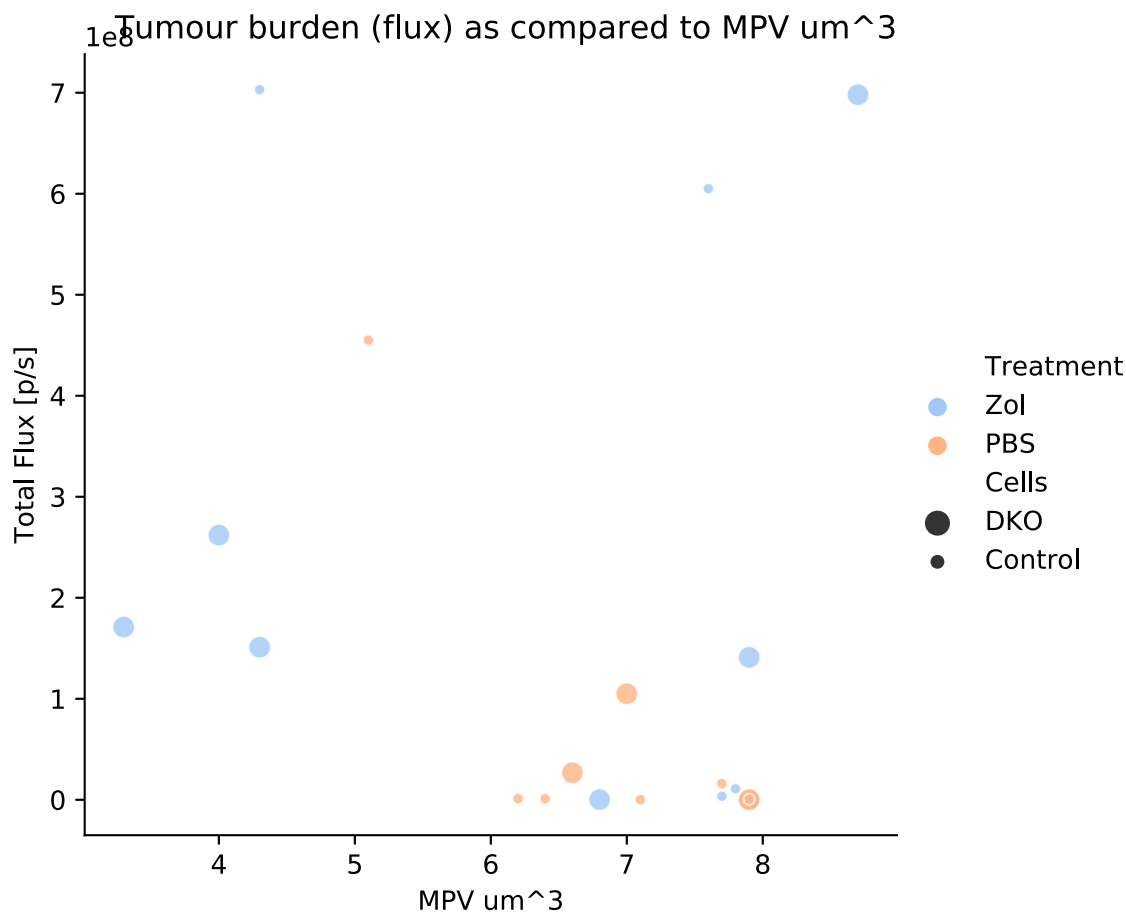
Tumour burden (flux) as compared to MPV um^3
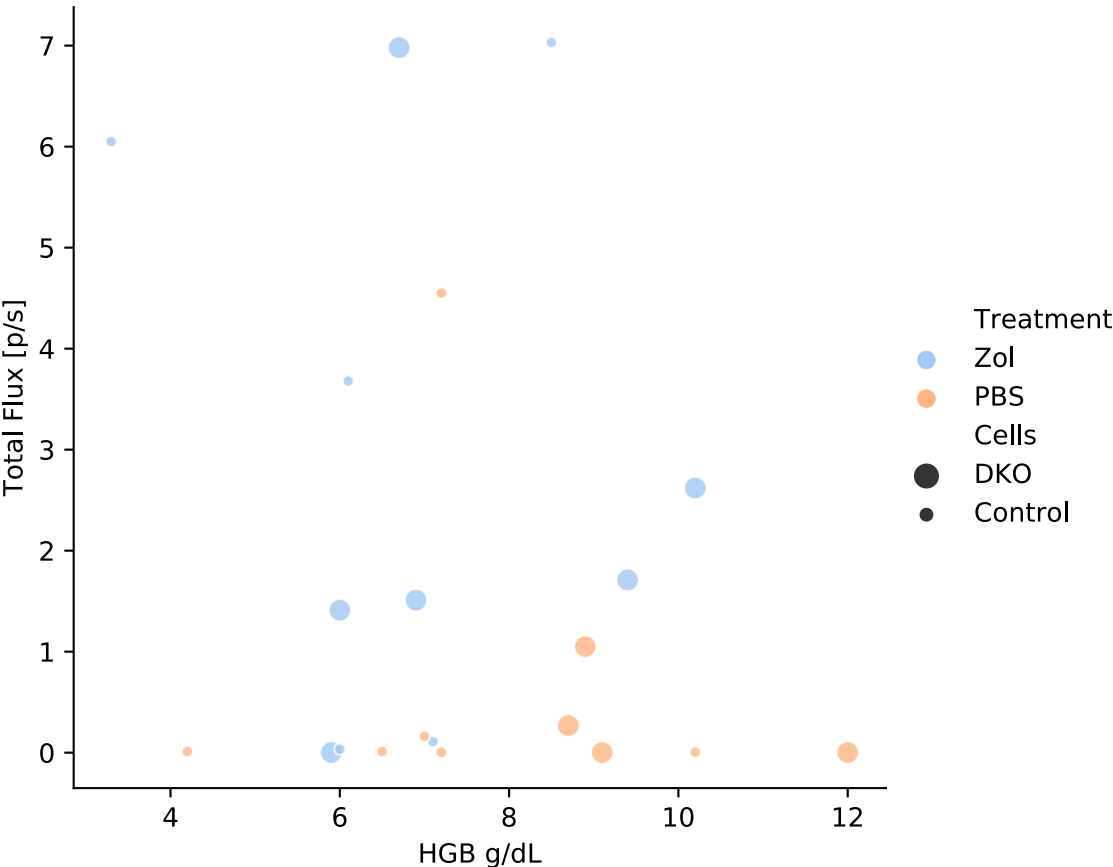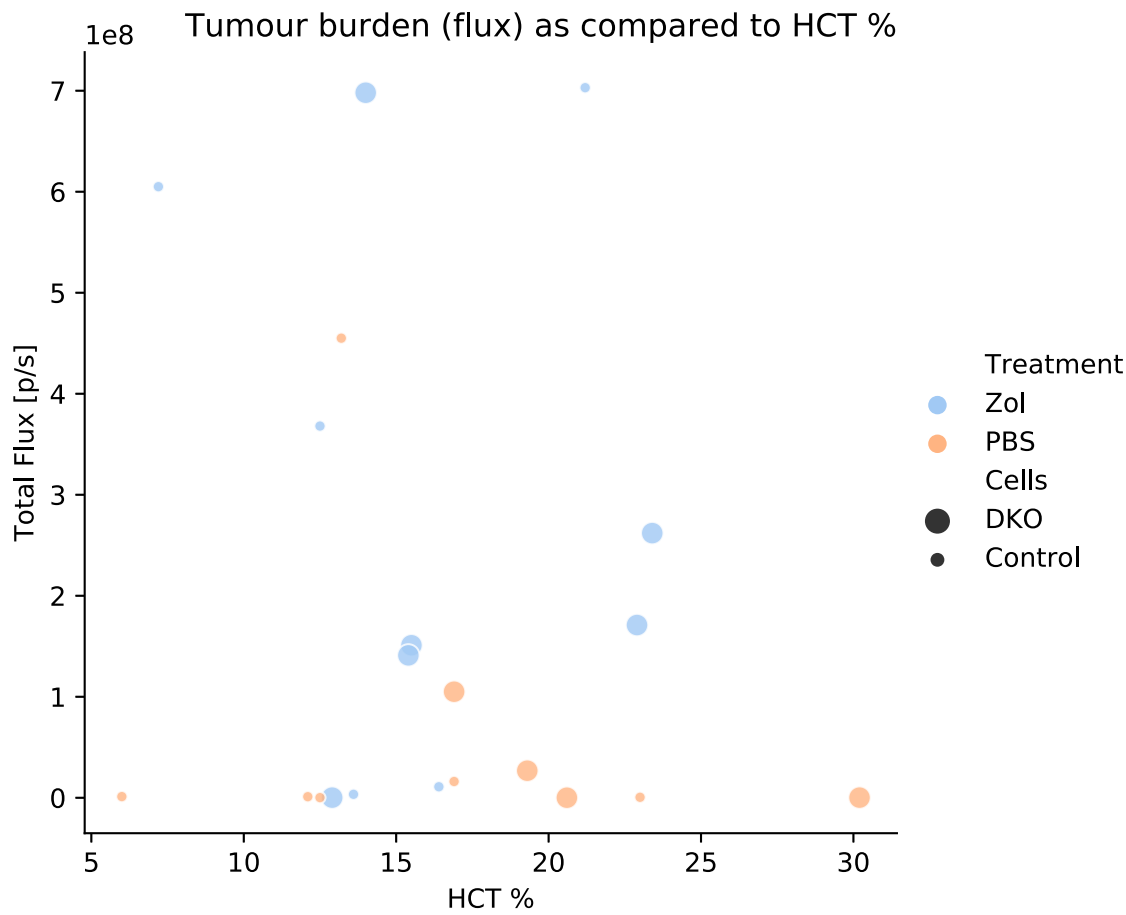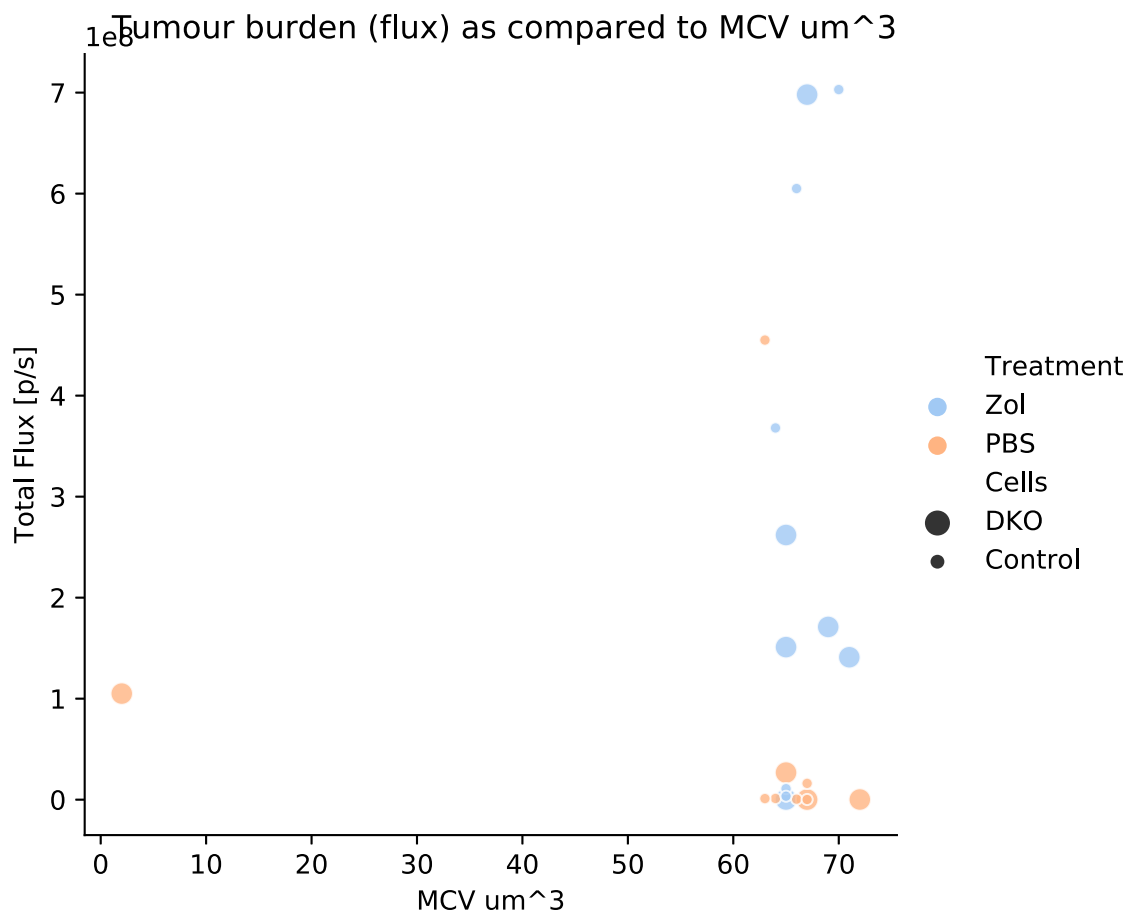
In [27]:

```
df_num = df_tumour.replace({'DKO':1,'Control':0,'Zol':1,'PBS':0})
```

In [29]:

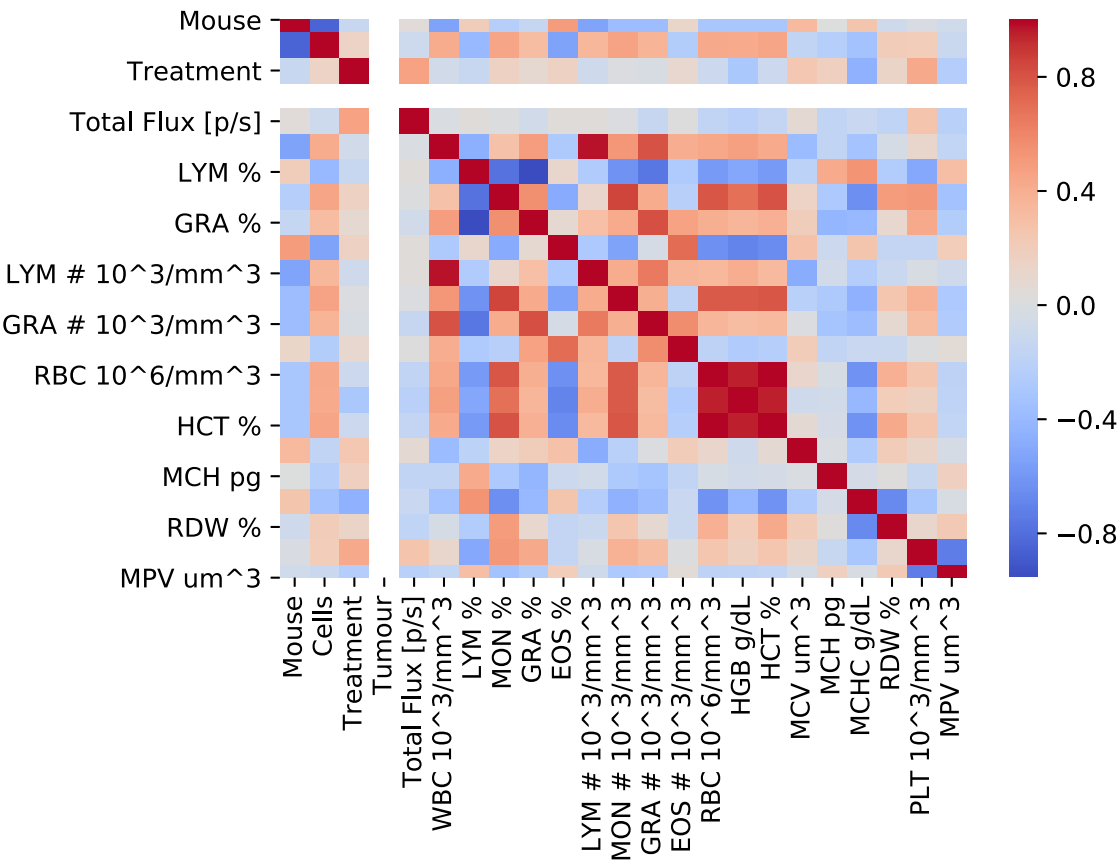```
df_corr = df_num.corr()
```

I then decided to plot a heatmap from the correlations above in order to see if there was any correlation between the tumour burden and any other columns. It seems the only thing it is a bit correlated with is treatment and platelets (PLT 10^3/mm^3)

In [31]:

```
sns.heatmap(df_corr, cmap='coolwarm')
```

Out[31]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2117d109e08>
```


So I decided to just look at the graph of the treatment and flux

In [53]:

```
sns.boxplot(data=df_tumour, y='Total Flux [p/s]', x='Treatment', hue='Cells', palette=
'pastel')
```
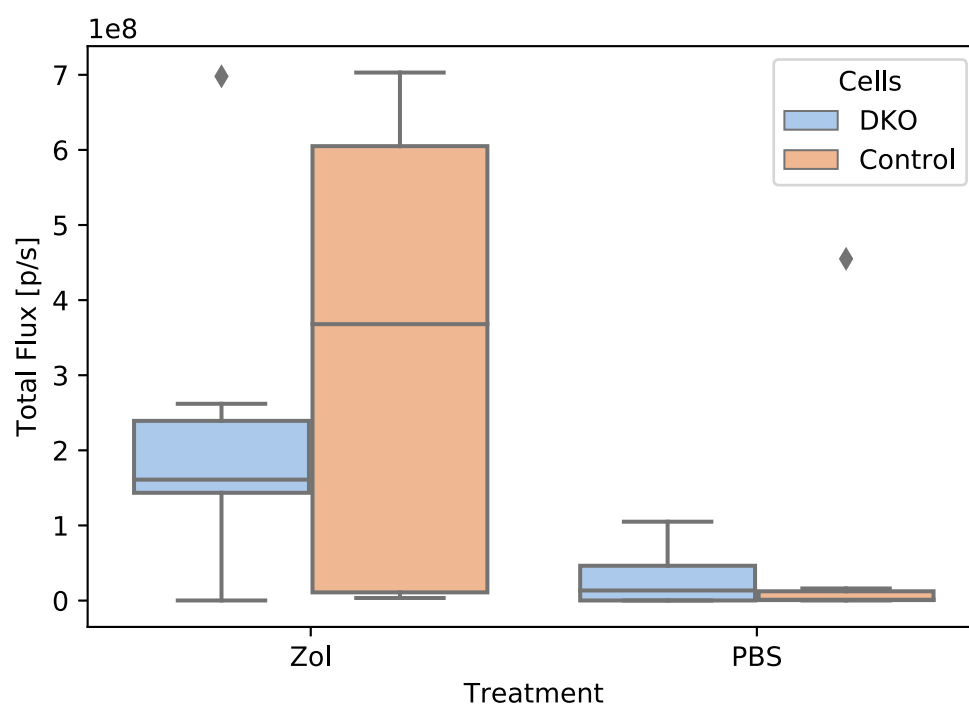
Out[53]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x2111a711748>
```


It seems that the zol treated animals had a higher tumour burden than the PBS-treated.