In [2]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [4]:
```
mdff = pd.read_csv("datasets/008ExerciseFile.csv")
display(mdff)
```

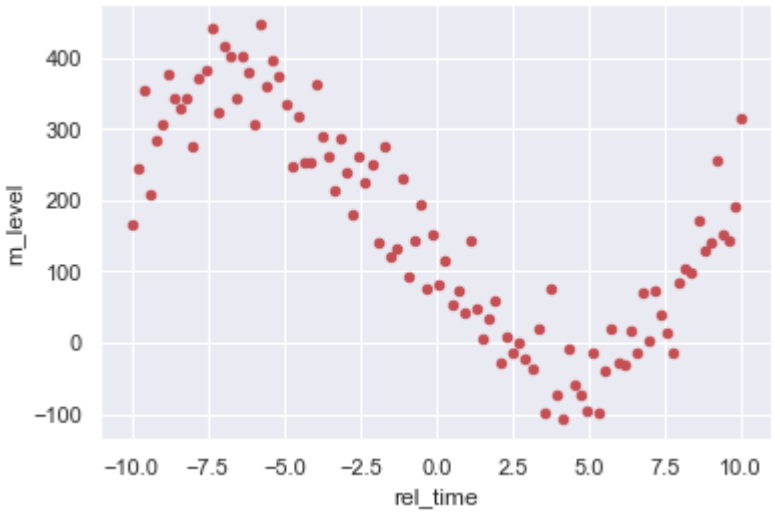
	rel_time	m_level
0	-10.000000	166.846602
1	-9.797980	243.656949
2	-9.595960	354.591642
3	-9.393939	209.023218
4	-9.191919	283.431508
...	...	...
95	9.191919	255.174991
96	9.393939	152.987761
97	9.595960	142.685158
98	9.797980	189.684193
99	10.000000	314.140057

100 rows × 2 columns

Try a straight line through the data

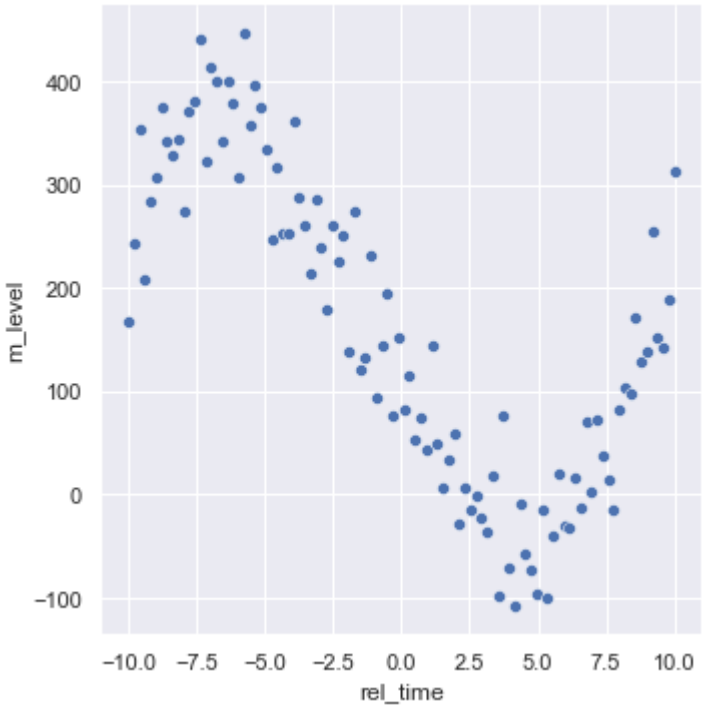
In [8]:
```
mdff.plot(x='rel_time', y='m_level', kind='scatter', color='r')
```

Out[8]: <AxesSubplot:xlabel='rel_time', ylabel='m_level'>



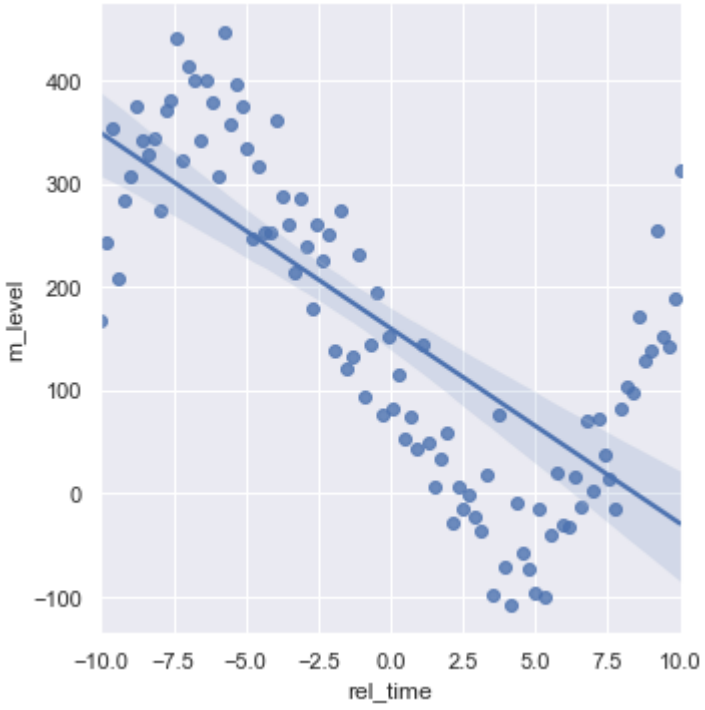
In [9]:
```
sns.set_theme()
sns.relplot(data=mdff, x='rel_time', y='m_level')
```

Out[9]: <seaborn.axisgrid.FacetGrid at 0x1634d68a8e0>



In [10]:
```
sns.lmplot(data=mdff, x='rel_time', y='m_level')
```

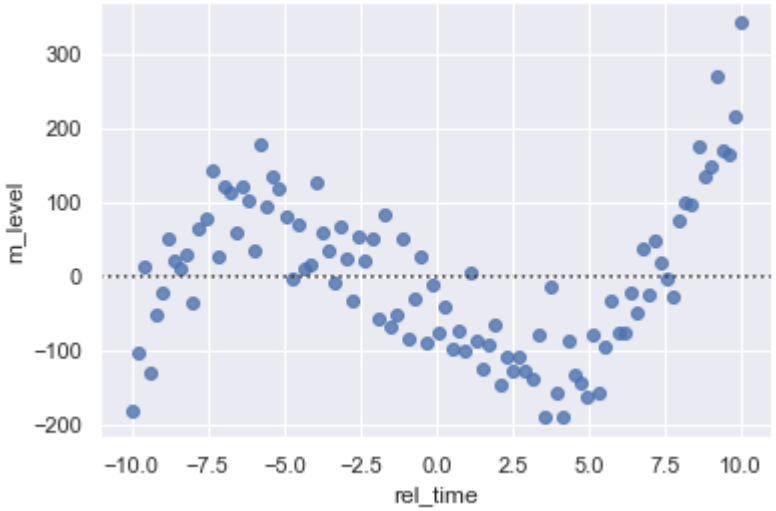
Out[10]: <seaborn.axisgrid.FacetGrid at 0x1634d68ac70>



The straight line does not depict a good line of fit for the data. Lets check the residual graph

In [11]:
```
sns.residplot(data=mdff, x='rel_time', y='m_level')
```

Out[11]: <AxesSubplot:xlabel='rel_time', ylabel='m_level'>

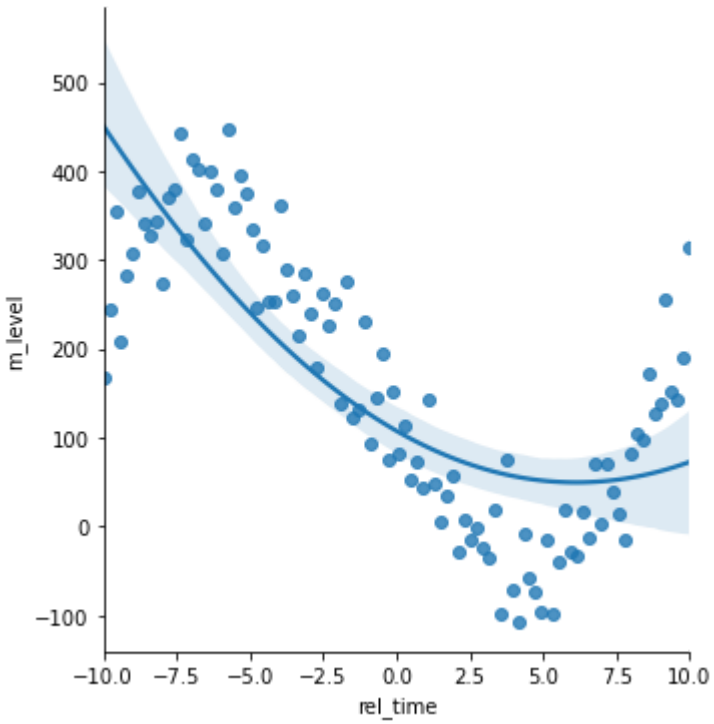


The graph shows a pattern to the residuals. Because the residuals are not random, we can reject that the straight line is a fit for the data

Now try to find a fit that works

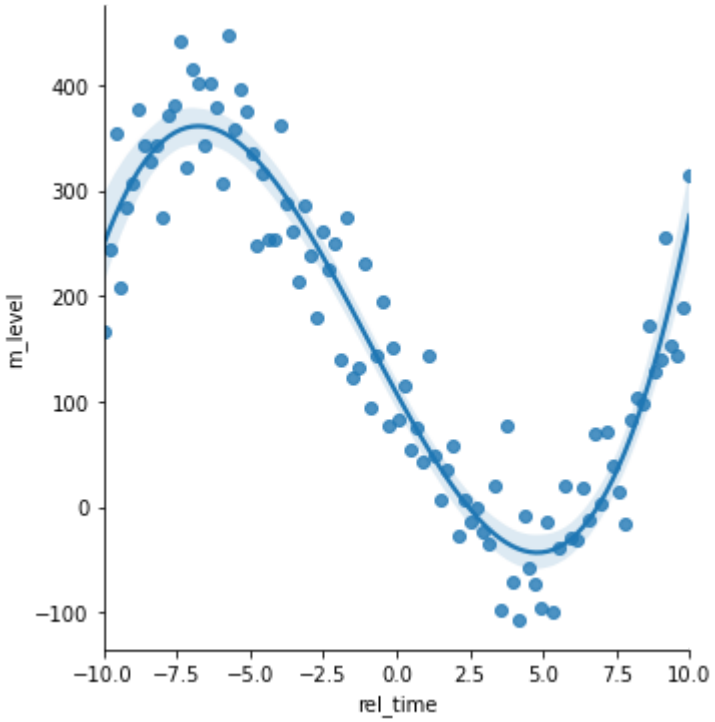
In [12]:
```
sns.lmplot(data=mdff, x='rel_time', y='m_level', order=2)
```

Out[12]: <seaborn.axisgrid.FacetGrid at 0x1fd583d4d30>



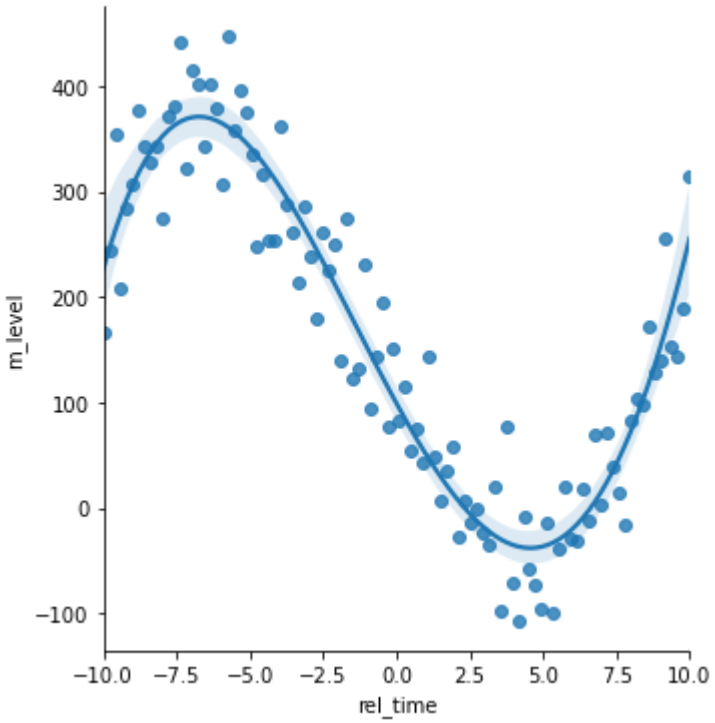
In [16]:
```
sns.lmplot(data=mdff, x='rel_time', y='m_level', order=3)
```

Out[16]: <seaborn.axisgrid.FacetGrid at 0x1fd583cc100>



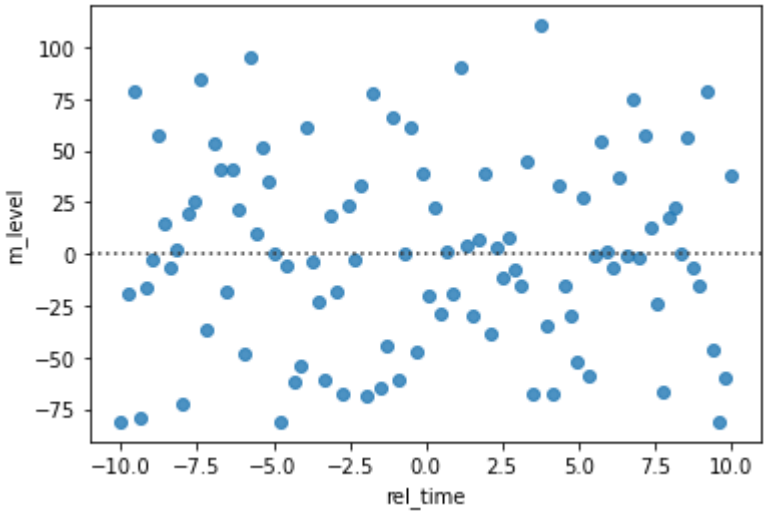
In [17]:
```
sns.lmplot(data=mdff, x='rel_time', y='m_level', order=4)
```

Out[17]: <seaborn.axisgrid.FacetGrid at 0x1fd59621e80>

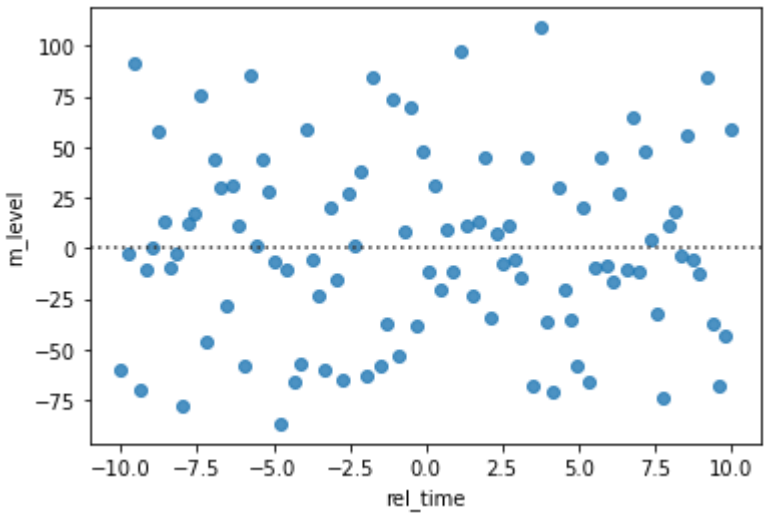


The line at order 3 and 4 appear to fit the data better. Let's check the residual graphs for these orders

In [18]:
```
g = sns.residplot(data=mdff, x='rel_time', y='m_level', order=3)
```



In [19]:
```
h = sns.residplot(data=mdff, x='rel_time', y='m_level', order=4)
```



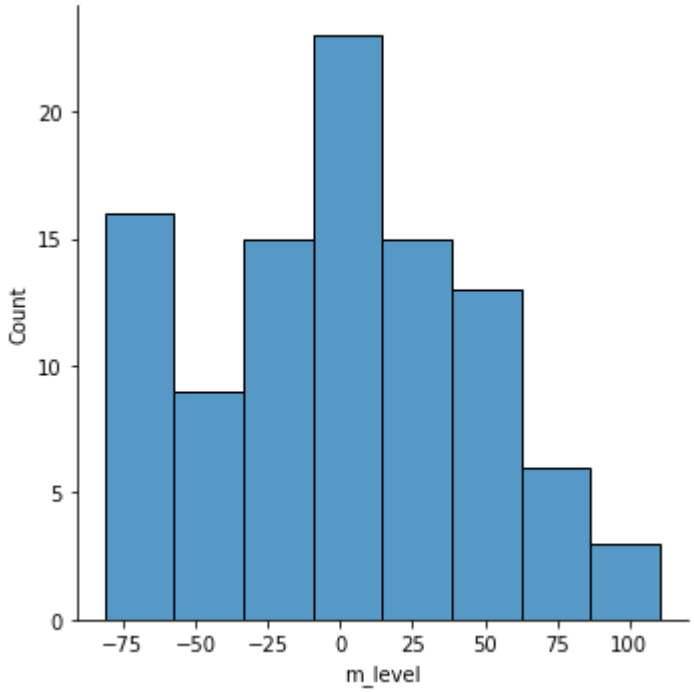
Both graphs show a good line of fit, but order=3 is a simpler model so it is the one we will go with

In [20]:
```
myfit = np.polyfit(x=mdff['rel_time'], y=mdff['m_level'], deg=3)
print(myfit)
```

```
[  0.52406776   1.5417871  -50.9901931  107.82613075]
```

In [21]:
```
fitvals = np.polyval(myfit, mdff['rel_time'])
myres = mdff['m_level'] - fitvals
sns.displot(myres)
```

Out[21]: <seaborn.axisgrid.FacetGrid at 0x1fd597384f0>



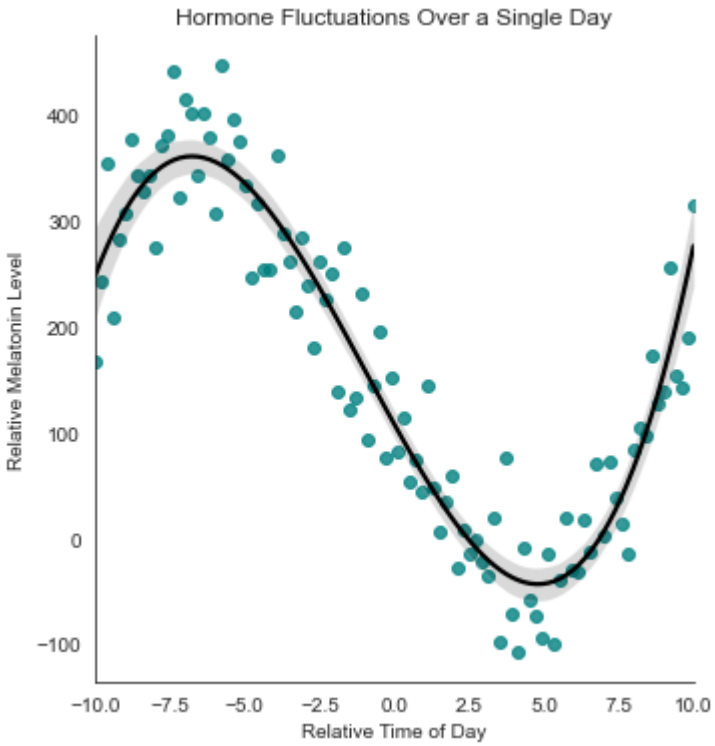
Our residuals shows a relatively normally distributed graph

In [22]:
```
np.std(myres)
```

Out[22]: 46.02814372678917

Let's make our graph pretty and combine the information we know

In [47]:
```
fig = sns.lmplot(data=mdff, x='rel_time', y='m_level', scatter_kws={'color':
plt.title("Hormone Fluctuations Over a Single Day")
plt.xlabel("Relative Time of Day")
plt.ylabel("Relative Melatonin Level")
sns.set_style("white")
```



The line of fit in our data is best captured by a third order polynomial. Our findings show the equation to be: y = 107.8 + -50.99x + 1.54x^2 + 0.52x^3 + N(0,46)

In [ ]:
```
```