In [62]:
```
import pandas as pd
import seaborn as sns
sns.set()
import matplotlib.pyplot as plt
%matplotlib inline
```

1) In what way does customer waiting time influences complaints ?

A) the goodwill pay is higher when the complaint is treated over the deadline.

OVERALL AVERAGE

The average of goodwill pay when the complaint is treated under the deadline is 124.9 pounds

The average of goodwill pay when the complaint is treated under the deadline is 132.1 pounds

AVERAGE SPLIT BY COMPLAINT TYPE

In [63]:
```
df0 = pd.read_csv('goodwill_spend_over_under_deadline.csv')
df0['under_deadline_payment'] = df0['under_deadline_payment'].round(1)
df0['over_deadline_payment'] = df0['over_deadline_payment'].round(1)
df0.head()
```

Out[63]:

	under_deadline_payment	over_deadline_payment	area_of_dissatisfaction
0	127.2	131.0	Disputes over sums/charges
1	85.6	89.5	Other
2	127.5	141.1	Other General Admin / Customer Service
3	76.8	17.0	Product Disclosure Information
4	125.5	134.3	Product Performance/Features

The tab above shows us that **for every area of dissatisfaction (except for the area of dissatisfaction "Product Disclosure Information", the goodwill payment over the deadline is superior to the one treated in time.** Which could mean that complaints not treated in time cost more to Monzo.

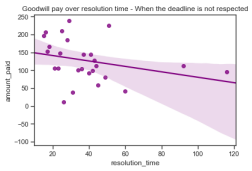
B) When the complaint is not treated in time, the amount of goodwill pay decreases with the resolution time taken

In [64]:
```
df2 = pd.read_csv('monzo_resolution_time_amount_paid_overdeadline.csv')
df2.columns =  ['amount_paid','resolution_time']
```

In [65]:
```
sns.set_style("ticks")
reg = sns.regplot(x="resolution_time", y="amount_paid", data=df2, color = 'purple')
reg.set_title('Goodwill pay over resolution time - When the deadline is not respected')
;
```

```
C:\Users\Marianne\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which w
ill result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[65]: ''



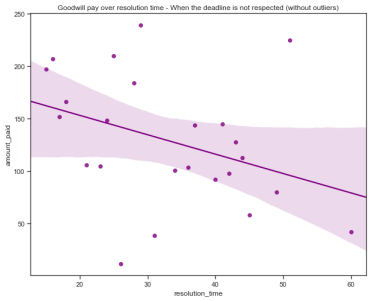
Lets remove the two outliers to have a clear view of the tendency

In [66]:
```
without_outliers = df2.loc[df2['resolution_time']<80,:]
```

In [67]:
```
plt.figure(figsize = (10, 8))
reg2 = sns.regplot(x="resolution_time", y="amount_paid", data=without_outliers, color = 'purple')
reg2.set_title('Goodwill pay over resolution time - When the deadline is not respected (without outliers)')
;
```

```
C:\Users\Marianne\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which w
ill result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[67]: ''



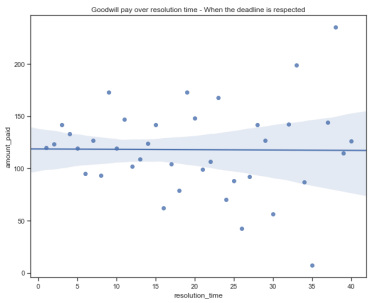
C) When the complaint is treated in time, the resolution time has no relation with the goodwill amount paid (see graph below)

In [68]:
```
df1 = pd.read_csv('monzo_resolution_time_amount_paid_underdeadline.csv')
df1.columns =  df2.columns
```

In [69]:
```
plt.figure(figsize = (10, 8))
reg3 = sns.regplot(x="resolution_time", y="amount_paid", data=df1)
reg3.set_title('Goodwill pay over resolution time - When the deadline is respected')
;
```

```
C:\Users\Marianne\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which w
ill result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[69]: ''

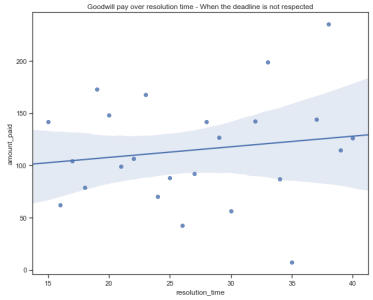


We see no correlation between the amount_paid and the resolution time but then again, let's have a look at resolution_time > 15 to compare better with complaints treated over the deadlines (the colmplaints treated over the deadline all have a resolution time > 15)

In [70]:
```
plt.figure(figsize = (10, 8))
same_resolution_time =df1.loc[df1["resolution_time"]>14,:]
reg3 = sns.regplot(x="resolution_time", y="amount_paid", data=same_resolution_time)
reg3.set_title('Goodwill pay over resolution time - When the deadline is not respected')
;
```

```
C:\Users\Marianne\Anaconda3\lib\site-packages\scipy\stats\stats.py:1713: FutureWarning: Using a non-tuple sequence for multidimensional indexing is deprecated; use `arr[tuple(seq)]` instead of `arr[seq]`. In the future this will be interpreted as an array index, `arr[np.array(seq)]`, which w
ill result either in an error or a different result.
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[70]: ''



That there is almost no correlation between the amount of goodwill paid and the resolution time **when the demand is treated under the deadline.**

2) What's the correlation between incident types and complaint types?

We have 3 columns in the incidents table. Incident_dates, Incidents_Customer_Impact, Incidents_count. *I assume that incident type is the same as Incidents_Customer_Impact*

There are 3 Incident types : Internal, Mobile external, Possible external. Complaint types (area of dissatisfaction) are the following : Disputes over sums/charges, Other General Admin / Customer Service, Product Disclosure Information, Product Performance/Features.]

Given the data provided in the tables monzo_risk.incidents and monzo_risk.raw_complaints, There is no apparent correlation between incident types and complaint types. let's try to find if the number of complaints and the number of incidents have the same evolution over time.

The first reflex was to assume that as complaints grow ncidents would grow as well but it seems that over the time incidents remain stable, at the contrary of complaints.

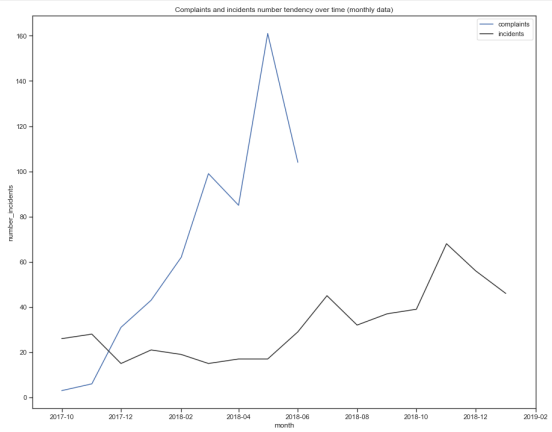
A) incidents number and complaints number do not follow a similar trend

In [71]:
```
complaints = pd.read_csv('complaints_per_month.csv')
incidents = pd.read_csv('incidents_per_month.csv')
```

In [72]:
```
merge=pd.merge(complaints,incidents, how='outer', on='months')
merge['months'] = pd.to_datetime(merge['months'])
merge.columns = ['number_complaints','month','number_incidents']
```

In [73]:
```
merge = merge[['month', 'number_complaints', 'number_incidents']]
merge = merge.sort_values('month')
```

In [74]:
```
plt.figure(figsize = (15, 12))
lm = sns.lineplot(x=merge['month'],y= merge['number_complaints'],label="complaints" ) # plotting t, a separately
ls = sns.lineplot(x=merge['month'],y= merge['number_incidents'],label="incidents",color=".2") # plotting t, a separately
ls.set(xlim=('2017-09-01', '2019-02-01'))
ls.set_title('Complaints and incidents number tendency over time (monthly data)')
ls.legend()
plt.show()
```



In [75]:
```
# source for complaints
# https://bigquery.cloud.google.com/results/analytics-take-home-test:US.bquijob_397511be_168b5777fab
# source for incidents
#https://bigquery.cloud.google.com/results/analytics-take-home-test:US.bquijob_71040735_168b5757295
#source for conversations
#https://bigquery.cloud.google.com/results/analytics-take-home-test:US.bquijob_52f69bc5_168b7db0f98
```

3) Can conversations and incidents data predict number of complaints? If not - why?

We have seen before that number of complaints and number of indicents do not follow the same tendency for the observed period.

When comparing the tendencies of the number of conversations and the number of complaints, it seems that the number of conversations grows over time, and the number of complaints also grows over time.

My answer is : although we can take the number of incidents and the number of conversations as risk indicators, we should not use them to predict the number of complaints.

The best approach is to use a timeseries model and to fit it on the number of complaints over time dataset. Using a train an a test dataset, we would be able to see the accuracy of our model.

A) The number of conversations and the number of complaints grow over time

In [76]:
```
conversations = pd.read_csv('conversations_per_month.csv')
```

In [77]:
```
merge2=pd.merge(complaints,conversations, how='outer', on='months')
merge2['months'] = pd.to_datetime(merge2['months'])
merge2.columns = ['number_complaints','month','number_conversations']
```

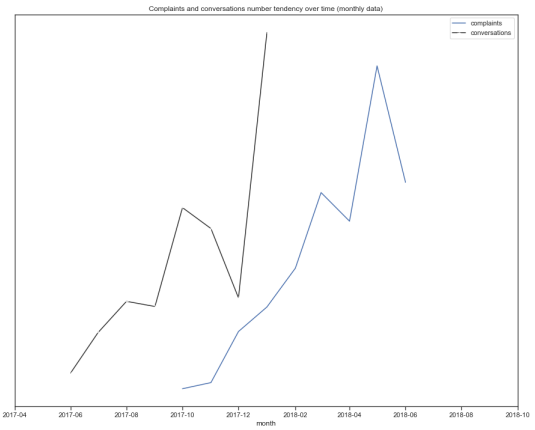
In [78]:
```
merge2 = merge2[['month', 'number_complaints', 'number_conversations']]
merge2 = merge2.sort_values('month')
```

It is clear that the number of complaints and of conversations are on a different scale. Let's put these variables on a same scale, substracting the mean and dividing by the standard deviation.

This means that the y axis value are not important since the number of complaints and the number of conversation are scaled, we only compare trends

In [79]:
```
from sklearn.preprocessing import scale
number_complaints_std = scale(merge2['number_complaints'],axis=0, with_mean=True, with_std=True, copy=True)
number_conversations_std = scale(merge2['number_conversations'],axis=0, with_mean=True, with_std=True, copy=True)
```

In [80]:
```
plt.figure(figsize = (15, 12))
lm = sns.lineplot(x=merge2['month'],y= number_complaints_std,label="complaints") # plotting t, a separately
ls = sns.lineplot(x=merge2['month'],y= number_conversations_std,label="conversations",marker="+",color=".2") # plotting t, a separately
ls.set(xlim=('2017-04-01', '2018-10-01'))
ls.set(yticks=[])
ls.set_title('Complaints and conversations number tendency over time (monthly data)')
ls.legend()
plt.show()
```



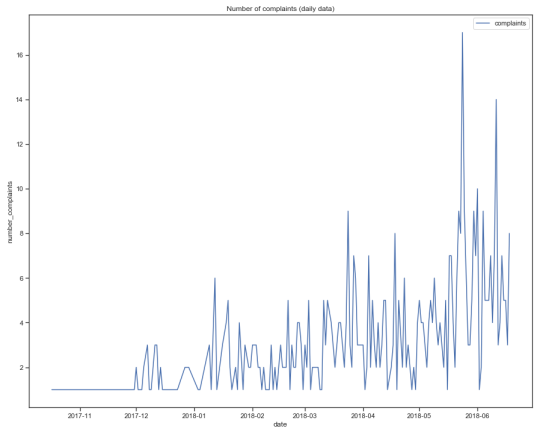
In [81]:
```
daily_complaints = pd.read_csv('day_to_day_datacomplaints.csv')
daily_complaints['months'] = pd.to_datetime(daily_complaints['months'])
daily_complaints = daily_complaints.sort_values('months')
daily_complaints.columns = ['number_complaints','date']
daily_complaints.head()
```

Out[81]:

	number_complaints	date
0	1	2017-10-17
1	1	2017-10-24
2	1	2017-10-30
3	1	2017-11-15
4	1	2017-11-16

B) The best predictor would be a time-series model (although the data should be made stationary first, i.e without trend)

In [82]:
```
plt.figure(figsize = (15, 12))
plot = sns.lineplot(x=daily_complaints['date'],y= daily_complaints['number_complaints'],label="complaints" )
plot.set_title('Number of complaints (daily data)')
plt.show()
```



3) To what type of complaints do different types of incidents lead to?

There are 3 Incident types : Internal, Mobile external, Possible external.

- Internal incidents may lead to Product Performance/Features, Disputes over sums/charges area of dissatisfaction
- Mobile external may lead to Product Disclosure Information area of dissatisfaction
- Possible external may lead to Other General Admin / Customer Service area of dissatisfaction

-Thanks for reading me!

Marianne Benkamoun