In [32]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [33]:
```
data1 = pd.read_csv('russia_losses_equipment.csv')
data2 = pd.read_csv('russia_losses_troop.csv')
```

In [34]:
```
data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4 entries, 0 to 3
Data columns (total 14 columns):
 #   Column                     Non-Null Count  Dtype 
---  ------                     --------------  ----- 
 0   date                       4 non-null      object
 1   day                        4 non-null      int64 
 2   plain                      4 non-null      int64 
 3   helicopter                 4 non-null      int64 
 4   tank                       4 non-null      int64 
 5   armored personnel carrier  4 non-null      int64 
 6   field artillery            4 non-null      int64 
 7   BUK                        4 non-null      int64 
 8   MRL Grad                   4 non-null      int64 
 9   military auto              4 non-null      int64 
 10  cistern                    4 non-null      int64 
 11  UAV, RPA, drone            4 non-null      int64 
 12  naval ship                 4 non-null      int64 
 13  anti-aircraft warfare      4 non-null      int64 
dtypes: int64(13), object(1)
memory usage: 576.0+ bytes
```

In [35]:
```
data1.describe()
```

Out[35]:

	day	plain	helicopter	tank	armored personnel carrier	field artillery	BUK	MRL Grad	military auto	cistern	UAV, RPA, drone	naval ship	anti-aircraft warfare
count	4.000000	4.000000	4.000000	4.000000	4.000000	4.000000	4.0	4.000000	4.000000	4.0	4.00	4.0	4.00000
mean	2.500000	28.750000	28.750000	176.250000	807.500000	71.250000	1.0	22.250000	245.250000	60.0	2.75	2.0	5.25000
std	1.290994	1.258306	2.061553	33.089525	70.301731	15.542951	0.0	14.750706	146.105841	0.0	0.50	0.0	3.86221
min	1.000000	27.000000	26.000000	146.000000	706.000000	49.000000	1.0	4.000000	30.000000	60.0	2.00	2.0	0.00000
25%	1.750000	28.500000	28.250000	149.000000	788.500000	67.750000	1.0	16.750000	225.750000	60.0	2.75	2.0	3.75000
50%	2.500000	29.000000	29.000000	174.000000	831.000000	75.500000	1.0	22.500000	298.000000	60.0	3.00	2.0	6.00000
75%	3.250000	29.250000	29.500000	201.250000	850.000000	79.000000	1.0	28.000000	317.500000	60.0	3.00	2.0	7.50000
max	4.000000	30.000000	31.000000	211.000000	862.000000	85.000000	1.0	40.000000	355.000000	60.0	3.00	2.0	9.00000

In [36]:
```
data1.describe().columns
```

Out[36]:
```
Index(['day', 'plain', 'helicopter', 'tank', 'armored personnel carrier',
       'field artillery', 'BUK', 'MRL Grad', 'military auto', 'cistern',
       'UAV, RPA, drone', 'naval ship', 'anti-aircraft warfare'],
      dtype='object')
```

In [37]:
```
data2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4 entries, 0 to 3
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   date    4 non-null      object 
 1   day     4 non-null      int64  
 2   troops  4 non-null      int64  
 3   POW     1 non-null      float64
dtypes: float64(1), int64(2), object(1)
memory usage: 256.0+ bytes
```

In [38]:
```
data2.describe()
```

Out[38]:

	day	troops	POW
count	4.000000	4.000000	1.0
mean	2.500000	5287.500000	200.0
std	1.290994	697.393958	NaN
min	1.000000	4300.000000	200.0
25%	1.750000	5050.000000	200.0
50%	2.500000	5505.000000	200.0
75%	3.250000	5742.500000	200.0
max	4.000000	5840.000000	200.0

In [39]:
```
data2.describe().columns
```

Out[39]:
```
Index(['day', 'troops', 'POW'], dtype='object')
```

In [40]:
```
data1.isnull().sum()
```

Out[40]:
```
date                          0
day                           0
plain                         0
helicopter                    0
tank                          0
armored personnel carrier     0
field artillery               0
BUK                           0
MRL Grad                      0
military auto                 0
cistern                       0
UAV, RPA, drone               0
naval ship                    0
anti-aircraft warfare         0
dtype: int64
```

In [41]:
```
data2.isnull().sum()
```

Out[41]:
```
date      0
day       0
troops    0
POW       3
dtype: int64
```

In [42]:
```
data2['POW'].fillna(data2['POW'].mode()[0], inplace = True)
data2.isnull().any().any()
```

Out[42]:
```
False
```

In [43]:
```
data1
```

Out[43]:

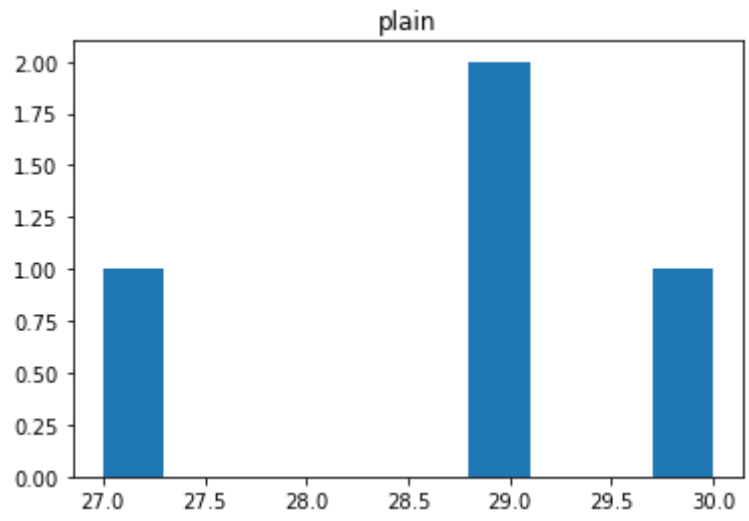
	date	day	plain	helicopter	tank	armored personnel carrier	field artillery	BUK	MRL Grad	military auto	cistern	UAV, RPA, drone	naval ship	anti-aircraft warfare
0	26-02-2022	1	27	26	146	706	49	1	4	30	60	2	2	0
1	28-02-2022	2	29	29	150	816	74	1	21	291	60	3	2	5
2	01-03-2022	3	29	29	198	846	77	1	24	305	60	3	2	7
3	02-03-2022	4	30	31	211	862	85	1	40	355	60	3	2	9

In [44]:
```
data2
```

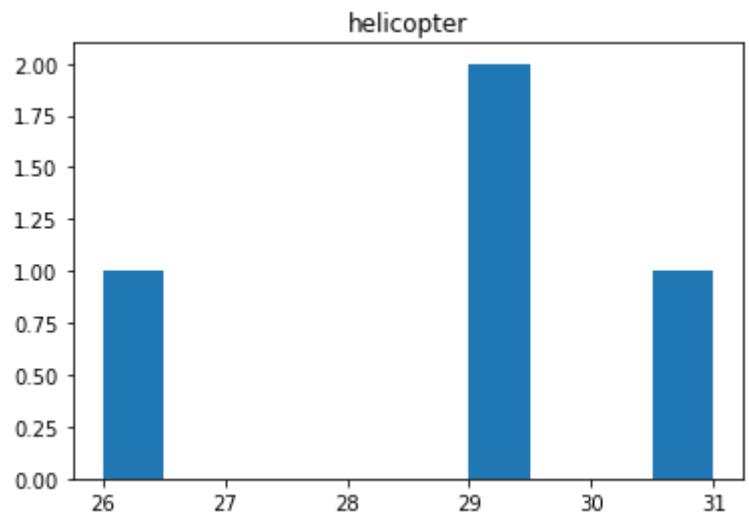
Out[44]:

	date	day	troops	POW
0	26-02-2022	1	4300	200.0
1	26-02-2022	2	5300	200.0
2	26-02-2022	3	5710	200.0
3	26-02-2022	4	5840	200.0

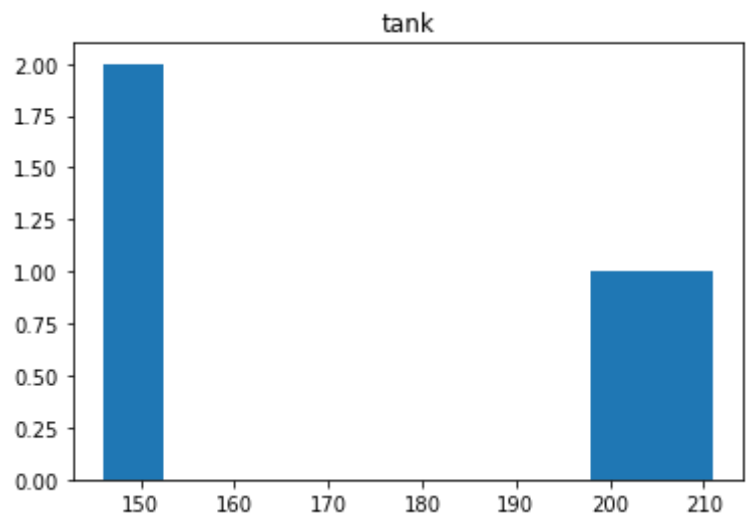
In [45]:
```
plt.hist(data1['plain'], bins=10)
plt.title("plain")
plt.show()
```



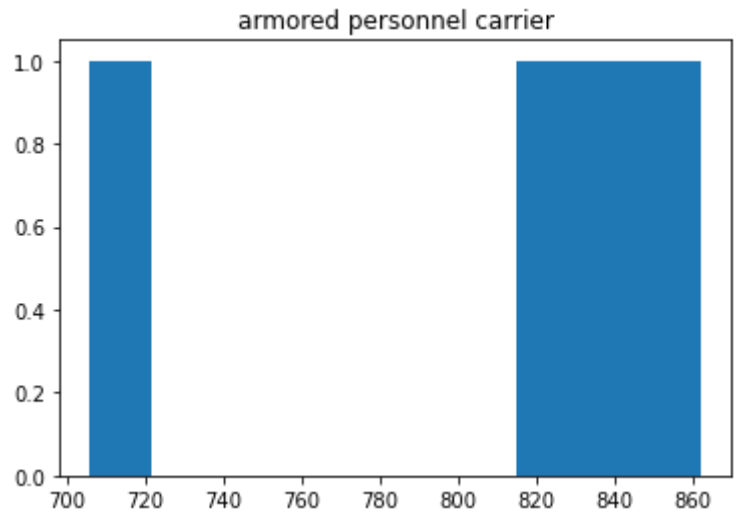
In [46]:
```
plt.hist(data1['helicopter'], bins=10)
plt.title("helicopter")
plt.show()
```



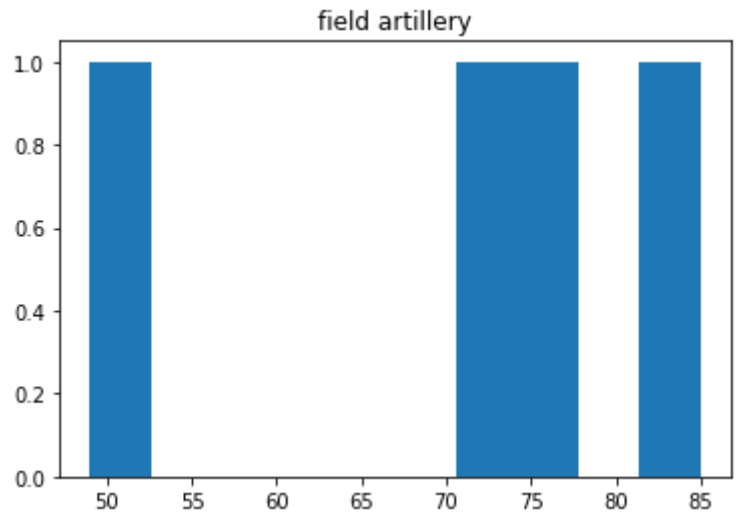
In [47]:
```
plt.hist(data1['tank'], bins=10)
plt.title("tank")
plt.show()
```



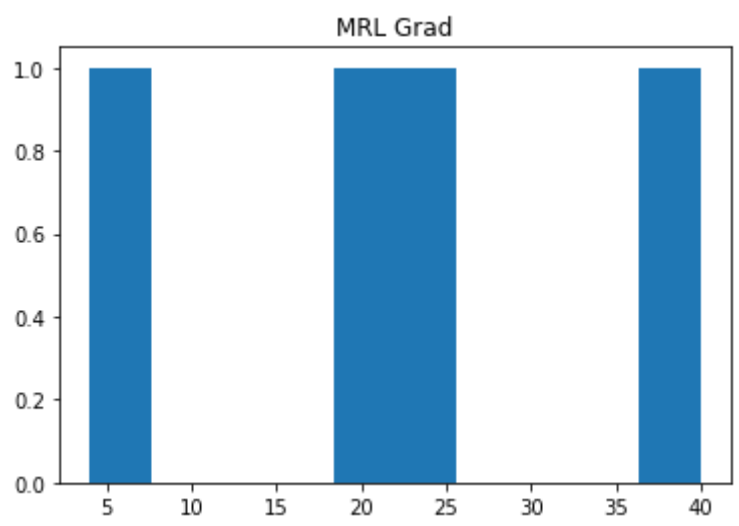
In [48]:
```
plt.hist(data1['armored personnel carrier'], bins=10)
plt.title("armored personnel carrier")
plt.show()
```



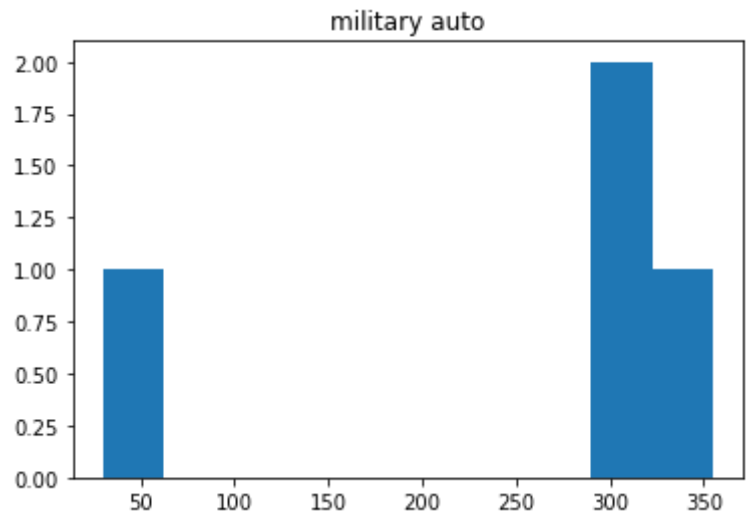
In [49]:
```
plt.hist(data1['field artillery'], bins=10)
plt.title("field artillery")
plt.show()
```



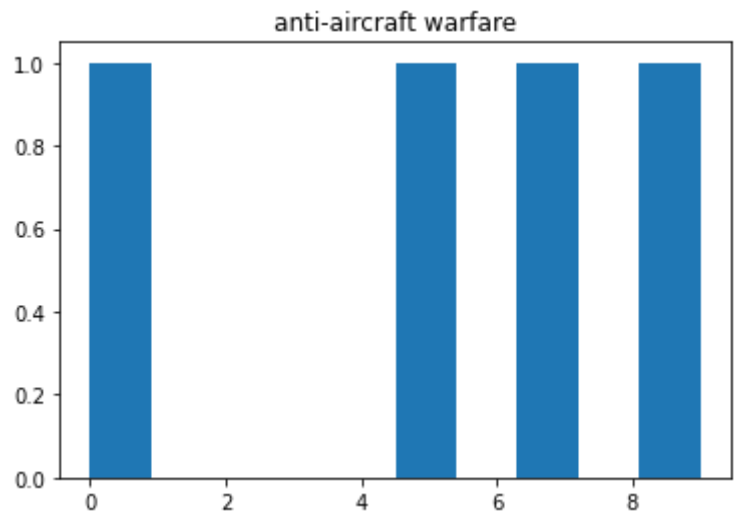
In [50]:
```
plt.hist(data1['MRL Grad'], bins=10)
plt.title("MRL Grad")
plt.show()
```



In [22]:
```
plt.hist(data1['military auto'], bins=10)
plt.title("military auto")
plt.show()
```



In [51]:
```
plt.hist(data1['anti-aircraft warfare'], bins=10)
plt.title("anti-aircraft warfare")
plt.show()
```



In [52]:
```
x = data1.drop(['date', 'day'], axis = 1)
```

In [53]:
```
y = data1.day
```

In [54]:
```
x.shape
```

Out[54]: (4, 12)

In [55]:
```
y.shape
```

Out[55]: (4,)

In [64]:
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.2)
```

In [65]:
```
model1 = RandomForestRegressor(n_estimators = 100, random_state = 0)
model1.fit(X_train, y_train)
```

Out[65]: RandomForestRegressor(random_state=0)

In [66]:
```
y_pred = model1.predict(X_test)
```

In [67]:
```
print("Training Accuracy :", model1.score(X_train, y_train))
print("Testing Accuracy :", model1.score(X_test, y_test))
```

```
Training Accuracy : 0.83335
Testing Accuracy : nan

c:\python\lib\site-packages\sklearn\metrics\_regression.py:796: UndefinedMetricWarning: R^2 score is not well-d
efined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
```

In [68]:
```
x1 = data2.drop(['date', 'day'], axis = 1)
```

In [70]:
```
y1 = data2.day
```

In [71]:
```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(x1, y1, test_size = 0.2)
```

In [72]:
```
model2 = RandomForestRegressor(n_estimators = 100, random_state = 0)
model2.fit(X_train, y_train)
```

Out[72]: RandomForestRegressor(random_state=0)

In [73]:
```
y_pred1 = model2.predict(X_test)
```

In [74]:
```
print("Training Accuracy :", model2.score(X_train, y_train))
print("Testing Accuracy :", model2.score(X_test, y_test))
```

```
Training Accuracy : 0.86695
Testing Accuracy : nan

c:\python\lib\site-packages\sklearn\metrics\_regression.py:796: UndefinedMetricWarning: R^2 score is not well-d
efined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
```