# Wyników głosowań w kongresie USA w 1986 r.

**Tematem naszego projektu jest przewidywanie przynależności partyjnej członka Izby Reprezentantów amerykańskiego kongresu w 1986 roku na podstawie dokonanych przez niego wyborów podczas głosowań. Naszym zbiorem danych jest ramka zawierająca dane o przynależności partyjnej poszczególnych reprezentantów i ich głosach podczas 16 kluczowych w tym roku głosowań.**

In [1]:

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
import sklearn.metrics
import random
from sklearn import manifold
random.seed(42)
import warnings
warnings.filterwarnings('ignore')
```

In [2]:

```
df=pd.read_csv("congressional_voting_dataset.csv")
```

In [3]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 435 entries, 0 to 434
Data columns (total 17 columns):
 #   Column                                  Non-Null Count  Dtype 
---  ------                                  --------------  ----- 
 0   handicapped_infants                     435 non-null    object
 1   water_project_cost_sharing              435 non-null    object
 2   adoption_of_the_budget_resolution       435 non-null    object
 3   physician_fee_freeze                    435 non-null    object
 4   el_salvador_aid                         435 non-null    object
 5   religious_groups_in_schools             435 non-null    object
 6   anti_satellite_test_ban                 435 non-null    object
 7   aid_to_nicaraguan_contras               435 non-null    object
 8   mx_missile                              435 non-null    object
 9   immigration                             435 non-null    object
 10  synfuels_corporation_cutback            435 non-null    object
 11  education_spending                      435 non-null    object
 12  superfund_right_to_sue                  435 non-null    object
 13  crime                                   435 non-null    object
 14  duty_free_exports                       435 non-null    object
 15  export_administration_act_south_africa  435 non-null    object
 16  political_party                         435 non-null    object
dtypes: object(17)
memory usage: 57.9+ KB
```

In [4]:

```
df.head()
```

Out[4]:

| | handicapped_infants | water_project_cost_sharing | adoption_of_the_budget_resolution | physician_fee_freeze | el_salvador_aid | r |
|---|---|---|---|---|---|---|
| 0 | n | y | n | y | y | |
| 1 | n | y | n | y | y | |
| 2 | ? | y | y | ? | y | |

| | 3 | handicapped_infants | water_project_cost_sharing | adoption_of_the_budget_resolution | physician_fee_freeze | el_salvador_aid | r |
|---|---|---|---|---|---|---|---|
| | 4 | y | y | y | n | y | |

# Objaśnienie zmiennych

**Kolumny 0-15 zawierają wyniki głosowań na tematy skrótowo opisane w nazwach kolumn. Każdy rząd odpowiada jednemu reprezentantowi. Możliwe wartości:**
**y - głos na tak**
**n - głos na nie**
**? - brak głosu - niewzięcie udziału w głosowaniu lub wstrzymanie się od głosu**
**Ostatnia kolumna zawiera informacje o przynależności partyjnej reprezentanta - republican albo democrat. W naszej ramce danych nie występuje bezpośrednio problem braku danych, ale zapewne będzie trzeba jakoś rozwiązać kwestię wartości ?.**

In [5]:

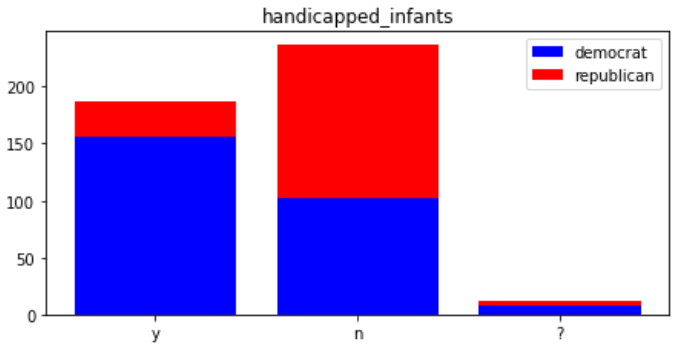
```
df.describe()
```

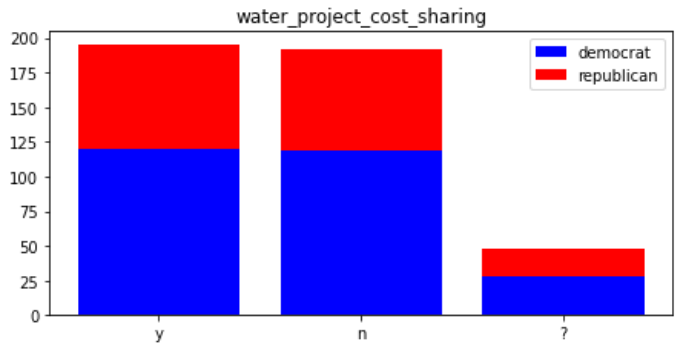
Out[5]:

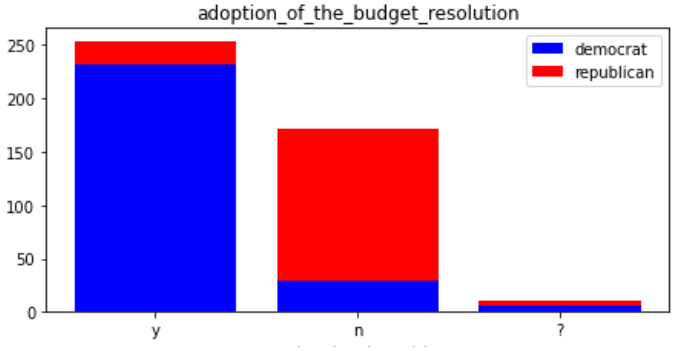
| | handicapped_infants | water_project_cost_sharing | adoption_of_the_budget_resolution | physician_fee_freeze | el_salvador_ |
|---|---|---|---|---|---|
| count | 435 | 435 | 435 | 435 | |
| unique | 3 | 3 | 3 | 3 | |
| top | n | y | y | n | |
| freq | 236 | 195 | 253 | 247 | |

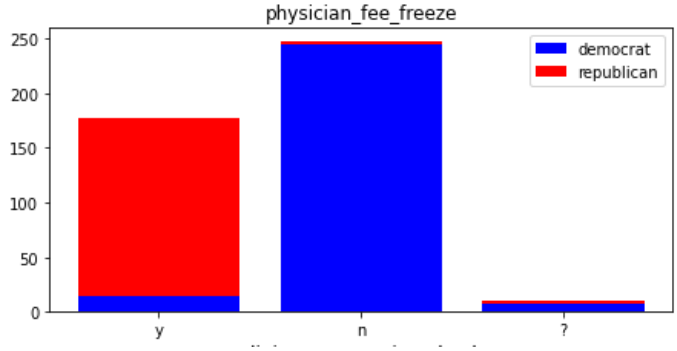
In [6]:

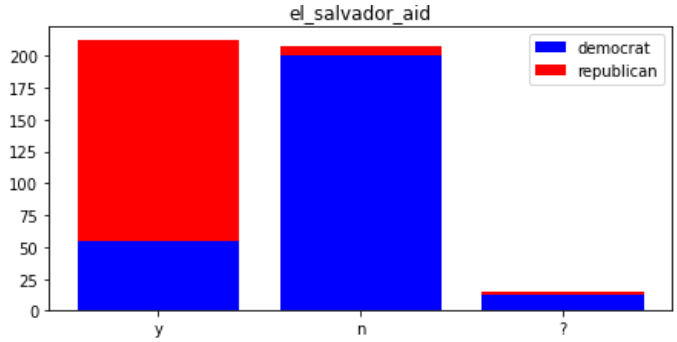
```
labels=["y", "n", "?"]
fig, axs = plt.subplots(ncols=2, nrows=8, figsize=(16, 32))
for i in range(len(df.columns)-1):
    col=df.columns[i]
    tmp=df[[col, "political_party"]].groupby(["political_party", col]).size().tolist()
    r, c= i//2, i%2
    axs[r,c].bar(labels, list(reversed(tmp[0:3])), label='democrat', color="blue")
    axs[r,c].bar(labels, list(reversed(tmp[3:6])), bottom=list(reversed(tmp[0:3])),
       label='republican', color="red")
    axs[r,c].legend()
    axs[r,c].set_title(col)
```

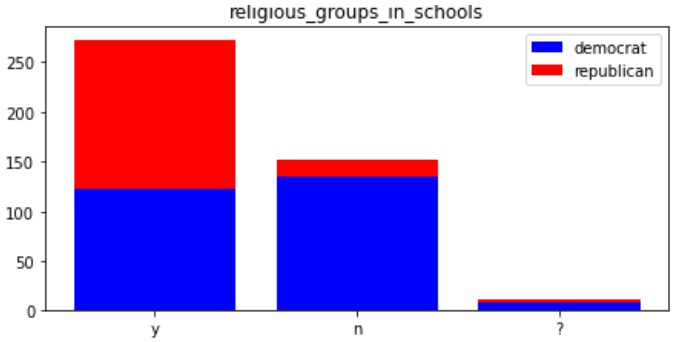


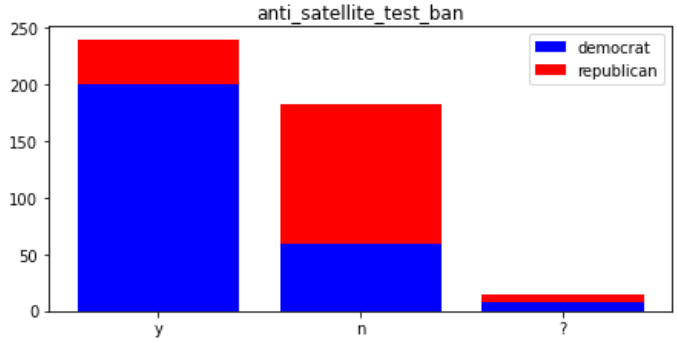



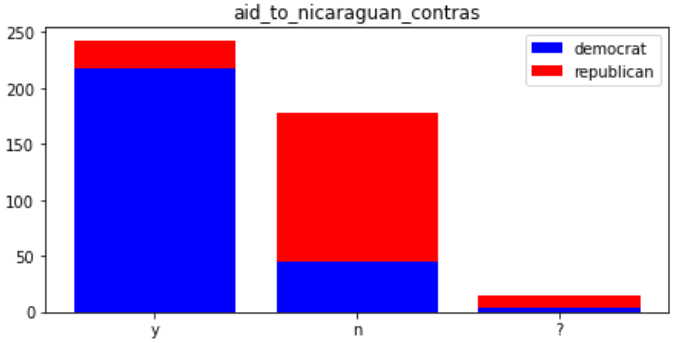



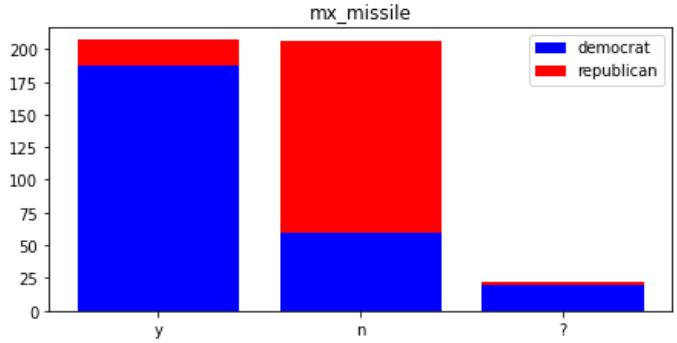



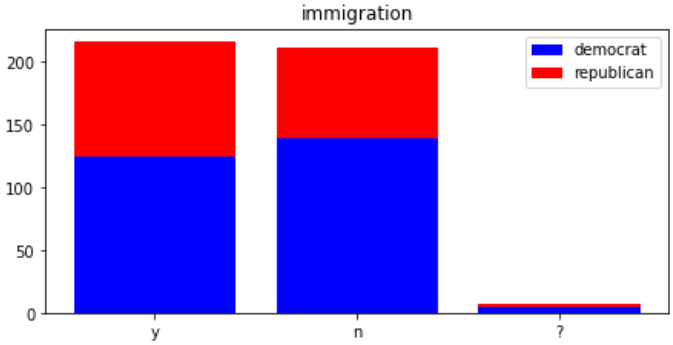


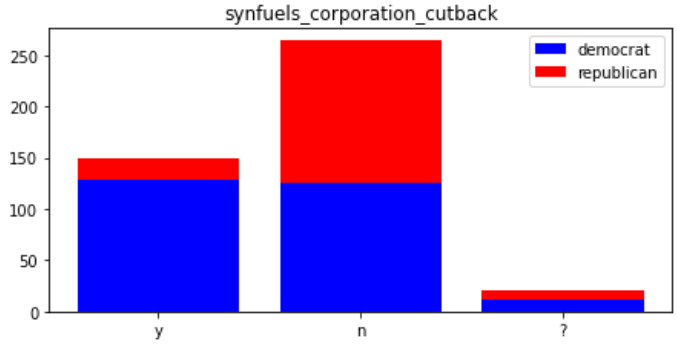



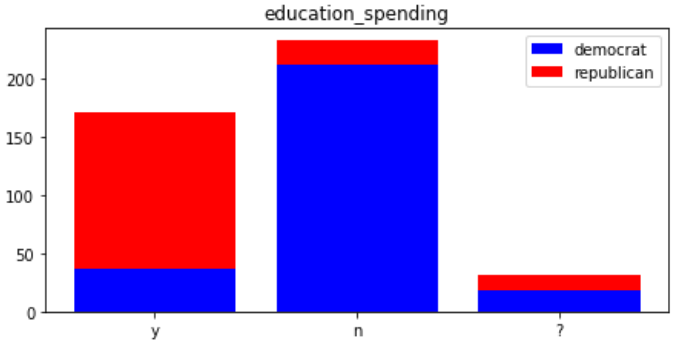



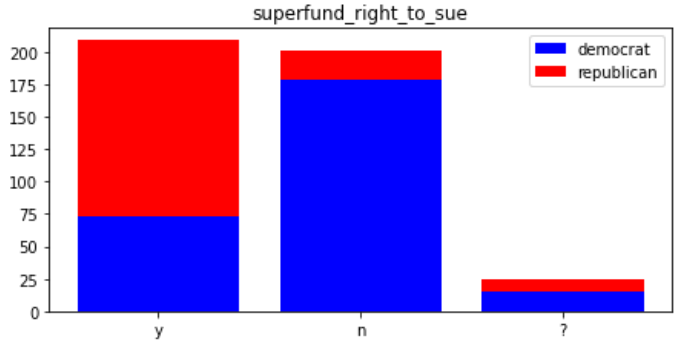



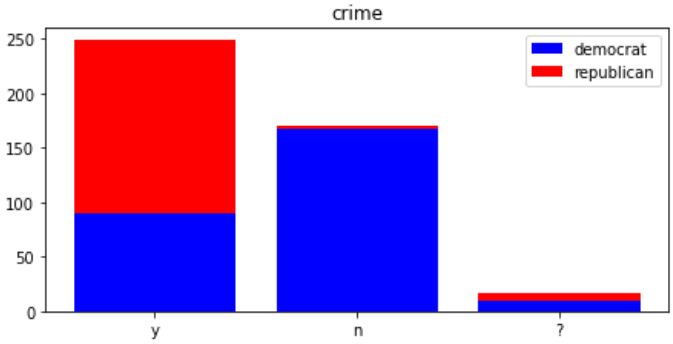



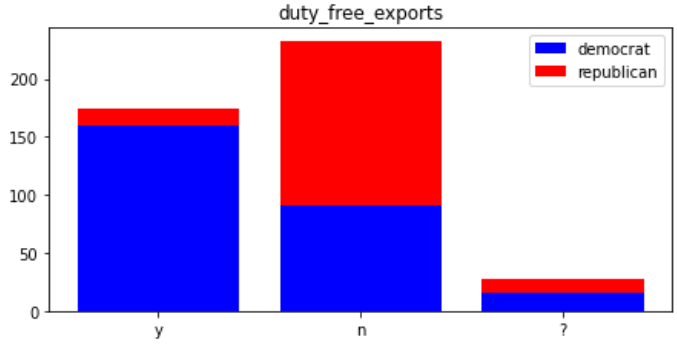



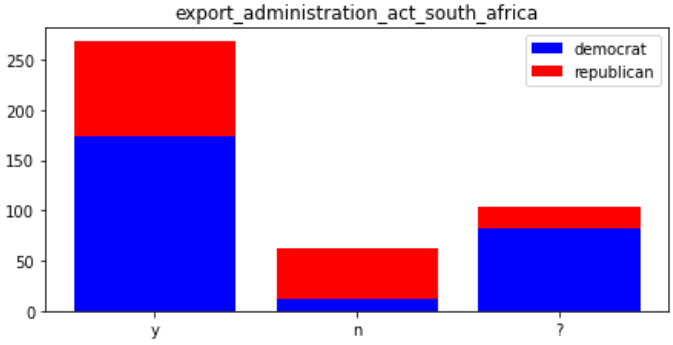



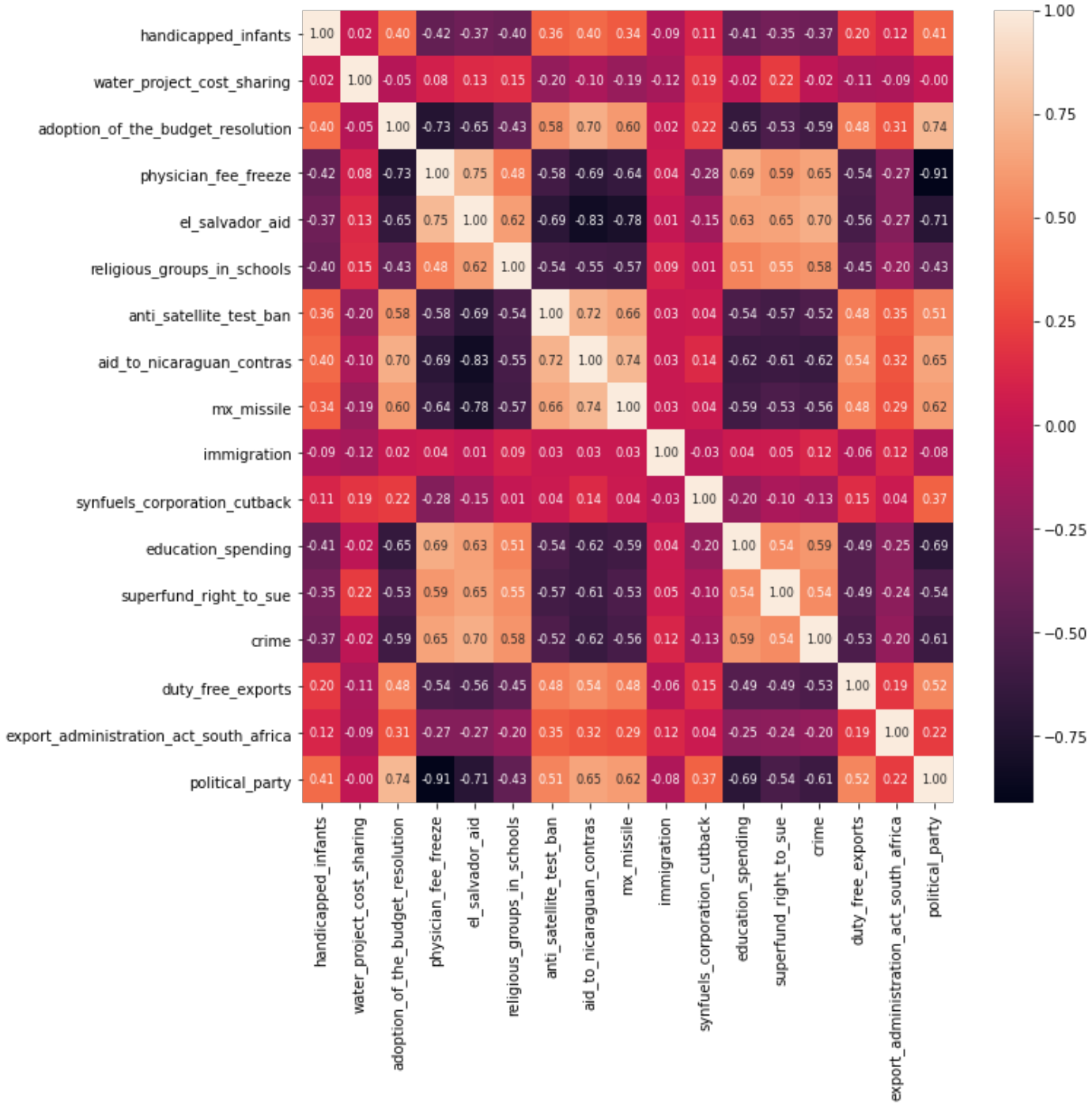

**Obie partie głosowały podobnie na `water_project_cost_sharing` oraz `imigration` (lecz u demokratów przeważa `no`, a u republikan `yes`) Widoczna róznica głosów dla:**

- `adoption_of_the_budget_resolution` **(r,no,d,yes)**

- `adoption_of_the_budget_resolution` **(r-no, d-yes)**
- `physician_fee_freeze` **(r-yes, d-no)**
- `el_salvador_aid` **(r-yes, d-no)**
- `education_spending` **(r-yes, d-no)**

In [7]:

```
df=df.replace("n", 0)
df=df.replace("y", 1)
df=df.replace("?",  0.5)    #rozwiązanie tymczasowe
df=df.replace("republican", 0)
df=df.replace("democrat",  1)
plt.figure(figsize=(10,10))
sns.heatmap(df.corr(), annot=True, annot_kws={'size': 8}, fmt='.2f')
plt.show()
```




**Jak widzimy, poziom korelacjii pomiędzy głosem a partią bardzo się różni w zależności od tematu głosowania - dla głosowania water_project_cost_sharing związek praktycznie nie istnieje, a dla physician_fee_freeze jest bardzo duży.**

**Spróbujemy teraz zobaczyć, na ile głosy poszczególnych reprezentantów przypominają głosy innych członków tej samej partii - w tym celu przekształcimy zapisy głosowań poszczególnych członków na wektory i policzymy**

**odległości pomiędzy każdą parą.**

In [8]:

```
adist=sklearn.metrics.pairwise_distances(df.drop(["political_party"], axis=1))
adist
```

Out[8]:

```
array([[0.        , 1.22474487, 2.17944947, ..., 1.22474487, 1.5       ,
        1.22474487],
       [1.22474487, 0.        , 1.93649167, ..., 1.22474487, 1.5       ,
        1.22474487],
       [2.17944947, 1.93649167, 0.        , ..., 1.93649167, 2.54950976,
        2.17944947],
       ...,
       [1.22474487, 1.22474487, 1.93649167, ..., 0.        , 1.5       ,
        1.87082869],
       [1.5       , 1.5       , 2.54950976, ..., 1.5       , 0.        ,
        1.80277564],
       [1.22474487, 1.22474487, 2.17944947, ..., 1.87082869, 1.80277564,
        0.        ]])
```

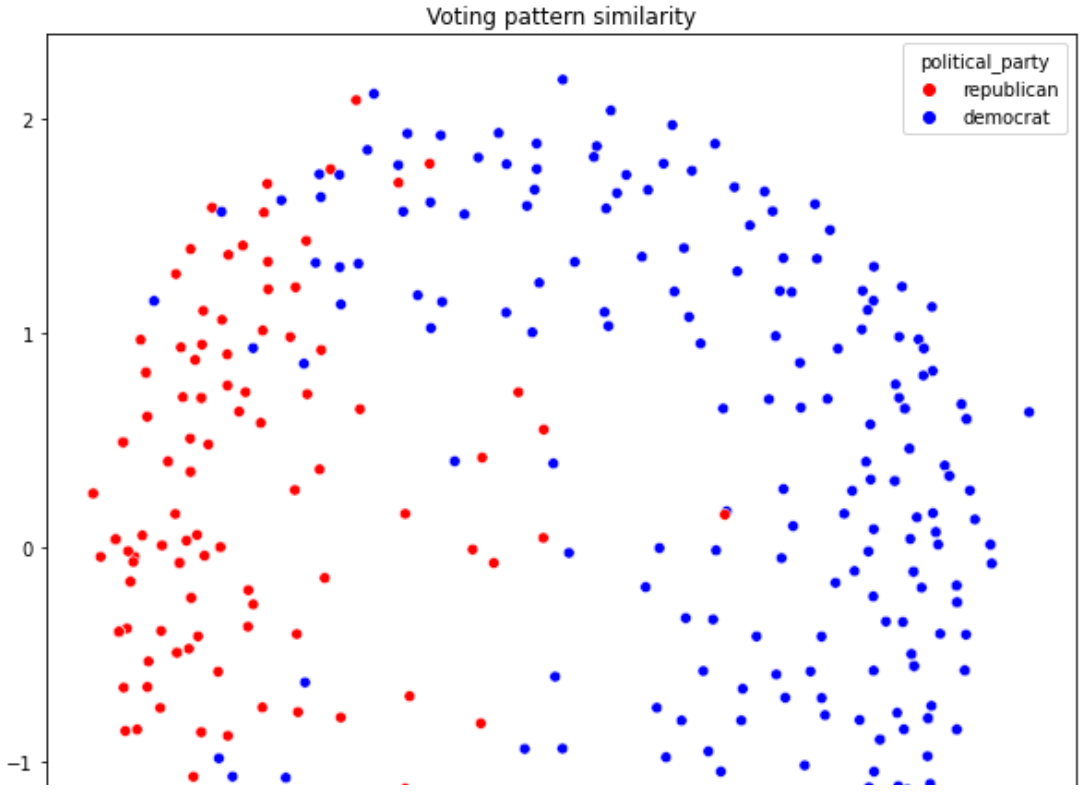
**Użyjemy teraz funkcji z pakietu manifold żeby przekształcić ramkę zawierającą wzajemne odległości na zbiór współrzędnych na dwuwymiarowej płaszczyźnie. Jest to rzut, który próbuje przekształcić wielowymiarowe zależności na płaszczyznę 2D.**

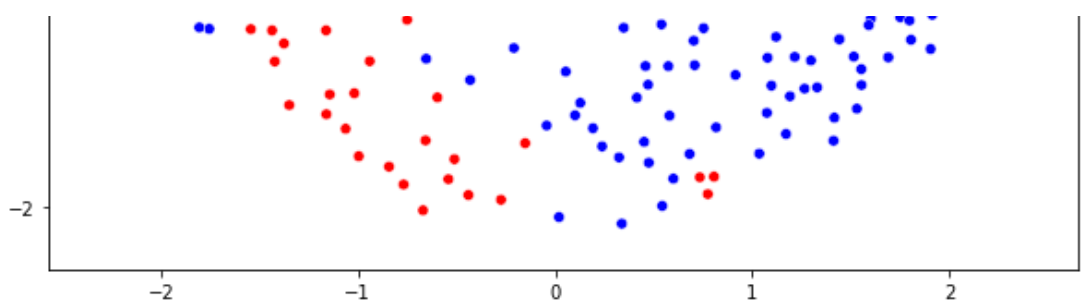
In [9]:

```
df["political_party"]=df["political_party"].replace(0, "republican")
df["political_party"]=df["political_party"].replace(1, "democrat")
adist=np.array(adist)
mds = manifold.MDS(n_components=2, dissimilarity="precomputed", random_state=6)
results = mds.fit(adist)
coords = results.embedding_
fig, ax = plt.subplots(figsize=(10,10))
sns.scatterplot(
    coords[:, 0], coords[:, 1], marker = 'o', hue=df["political_party"], palette=["red",
"blue"]
    )
ax.set_title("Voting pattern similarity")
```

Out[9]:

```
Text(0.5, 1.0, 'Voting pattern similarity')
```





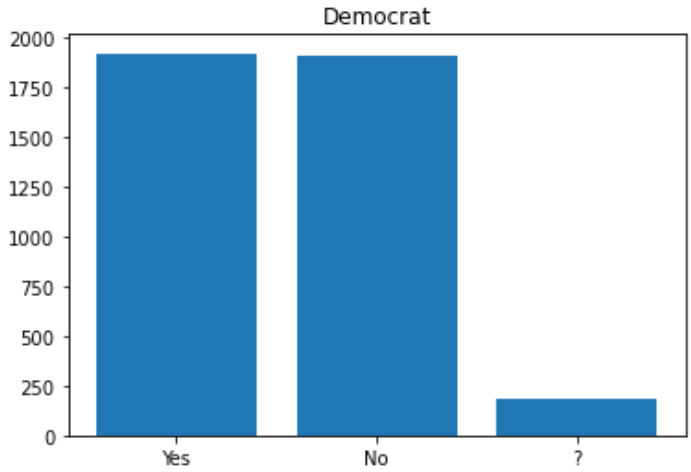

**Dodatkowo sprawdźmy czy któraś z parti ma skołonność do głosowania na tak lub nie.**

In [14]:

```
df=pd.read_csv("congressional_voting_dataset.csv")
democrat_df = df[df['political_party'] == 'democrat']
republican_df = df[df['political_party'] == 'republican']
```

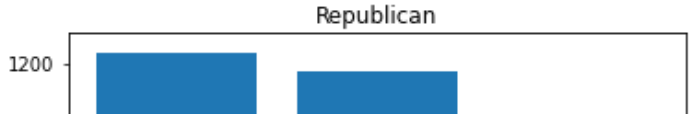
In [17]:

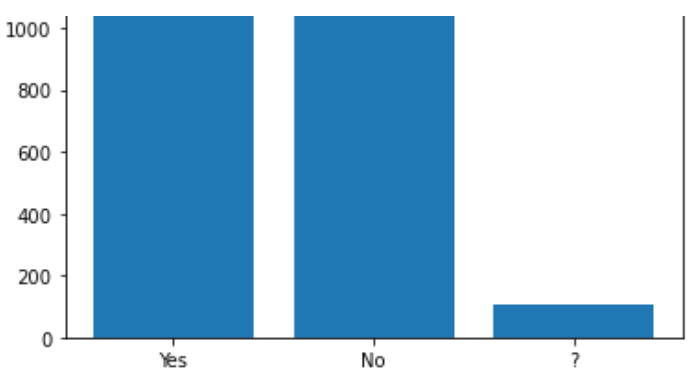
```
tak = 0
nie = 0
brak = 0
for i in range(0,15):
    tak += (democrat_df[democrat_df.columns[i]] == "y").sum()
    nie += (democrat_df[democrat_df.columns[i]] == "n").sum()
    brak += (democrat_df[democrat_df.columns[i]] == "?").sum()

labels = ['Yes', 'No', '?']
sizes = [tak, nie,brak]
plt.title("Democrat")
plt.bar(labels, sizes)
plt.show()
```




In [16]:

```
tak = 0
nie = 0
brak = 0
for i in range(0,15):
    tak += (republican_df[republican_df.columns[i]] == "y").sum()
    nie += (republican_df[republican_df.columns[i]] == "n").sum()
    brak += (republican_df[republican_df.columns[i]] == "?").sum()

labels = ['Yes', 'No', '?']
sizes = [tak, nie,brak]
plt.title("Republican")
plt.bar(labels, sizes)
plt.show()
```






**W obu przypadkach licza głosów jest dość wyrównana.**

In [ ]: