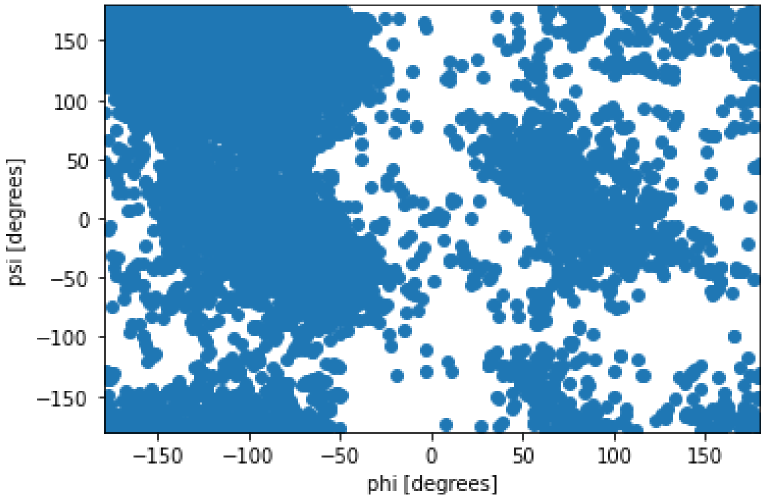
In [ ]:

```python
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

from sklearn import cluster
```

In [ ]:

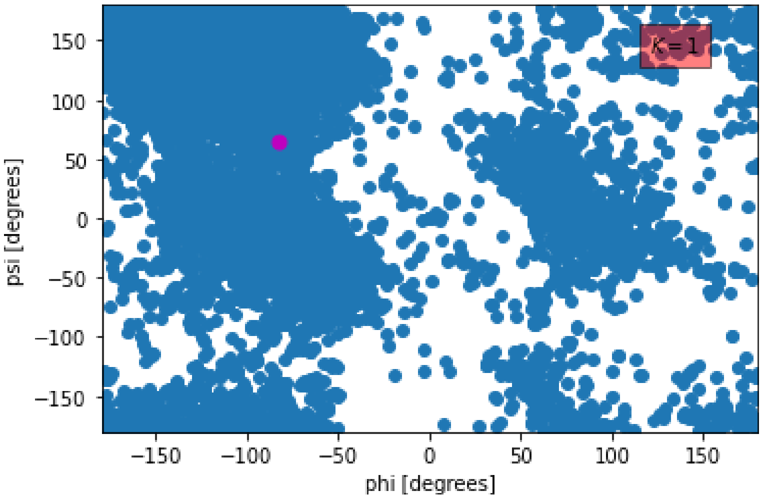
```python
df = pd.read_csv("data_all.csv")

# df = df[df["residue name"] == "GLY"] #comment in this line to get only one residue
print("Dataset size: ", len(df["residue name"]))

plt.xlabel("phi [degrees]")
plt.ylabel("psi [degrees]")
plt.xlim([-180, 180])
plt.ylim([-180, 180])
plt.scatter(df.phi, df.psi)
plt.show()
```

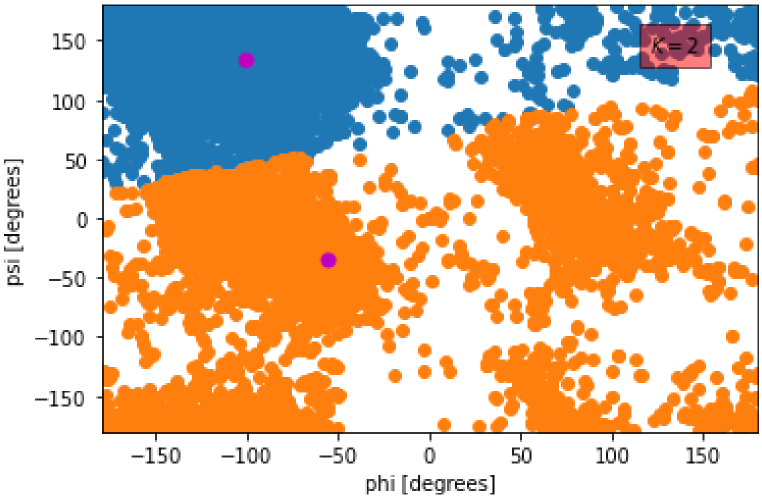
```
Dataset size:  29369
```

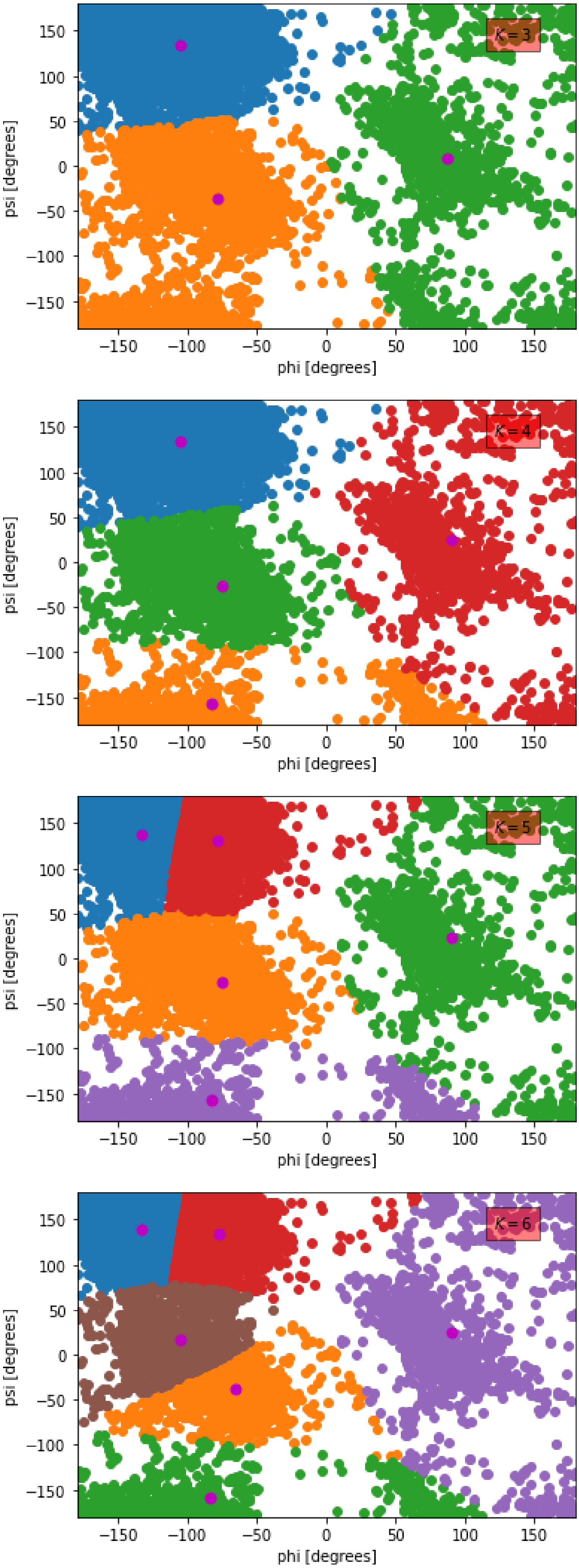


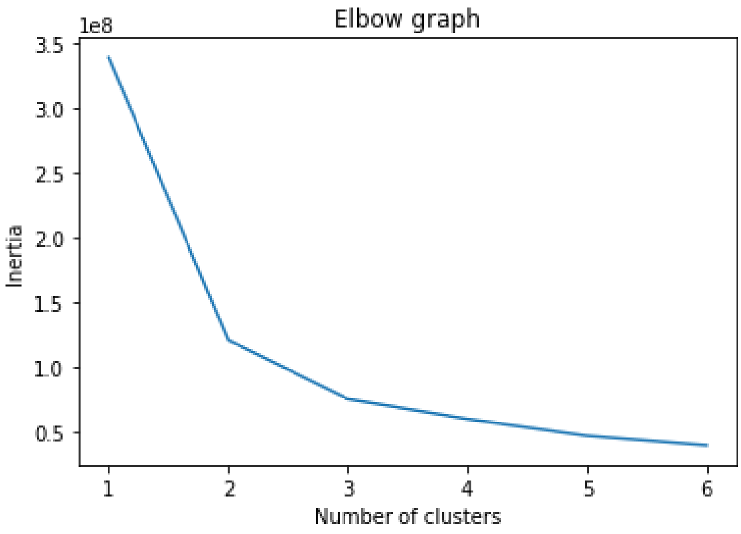
In [ ]:

```python
inertias = []

phi = list(df.phi)
psi = list(df.psi)

angles = [list(a) for a in zip(phi, psi)]

for i in range(1, 7):
    n_clusters = i

    kmean = cluster.KMeans(n_clusters = n_clusters, random_state = 0).fit(angles)

    inertias.append(kmean.inertia_)

    centers = kmean.cluster_centers_
    labels = kmean.labels_

    l, all = [list(a) for a in zip(*sorted(zip(labels, angles)))]
    all = np.array(all)


    plt.xlabel("phi [degrees]")
    plt.ylabel("psi [degrees]")
    start = 0
    stop = -1
    for i in range(n_clusters):
        try:
            stop = l.index(i + 1)
        except ValueError:
            stop = -1

        plt.scatter(all[start:stop, 0], all[start:stop, 1])
        start = stop


    for i in range(n_clusters):
        plt.plot(centers[i][0], centers[i][1], 'mo', ms = 7)

    plt.text(120, 140, r"$K = $" + str(n_clusters), bbox = {'facecolor': 'red', 'alp
    plt.xlim([-180, 180])
    plt.ylim([-180, 180])
    plt.show()

plt.title("Elbow graph")
plt.ylabel("Inertia")
plt.xlabel("Number of clusters")
plt.plot(list(range(1, 7)), inertias)
plt.show()
```



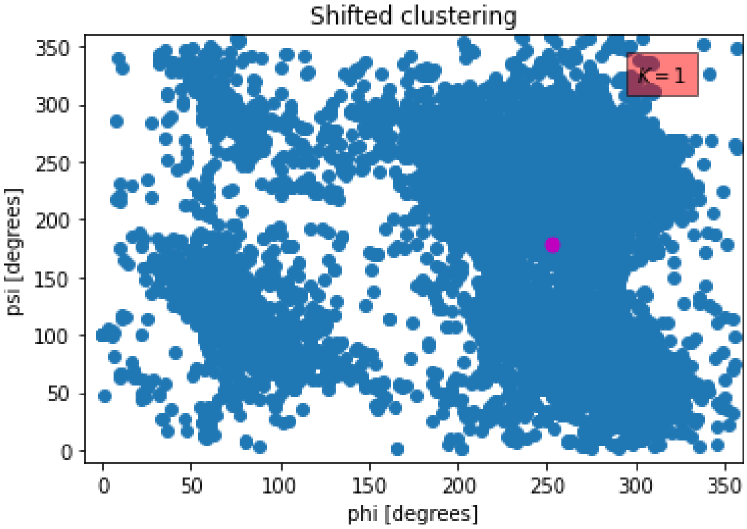


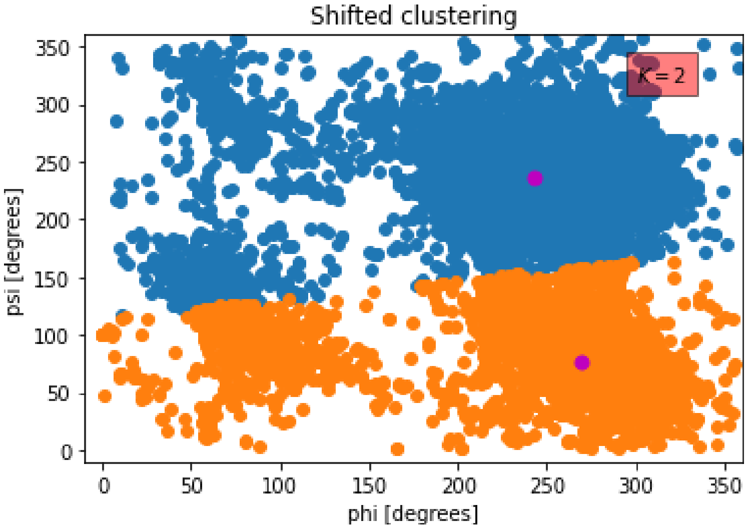

K = 3
psi [degrees]
phi [degrees]
150
100
50
0
−50
−100
−150
−150
−100
−50
0
50
100
150
K = 4
psi [degrees]
phi [degrees]
K = 5
psi [degrees]
phi [degrees]
K = 6
psi [degrees]
phi [degrees]

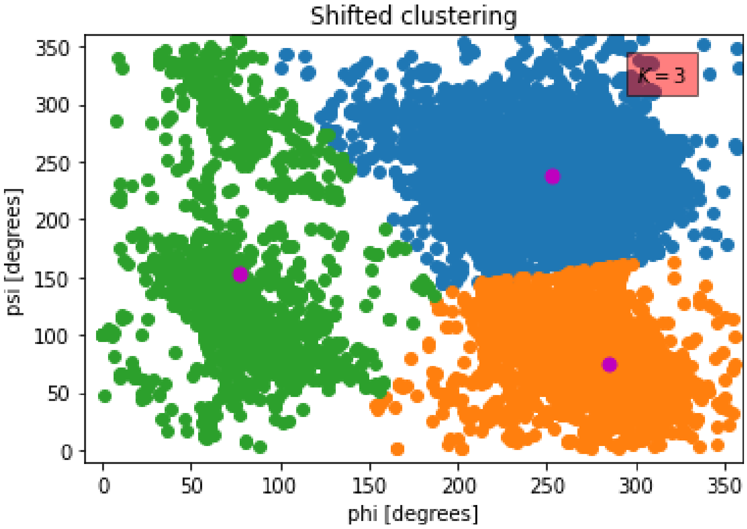


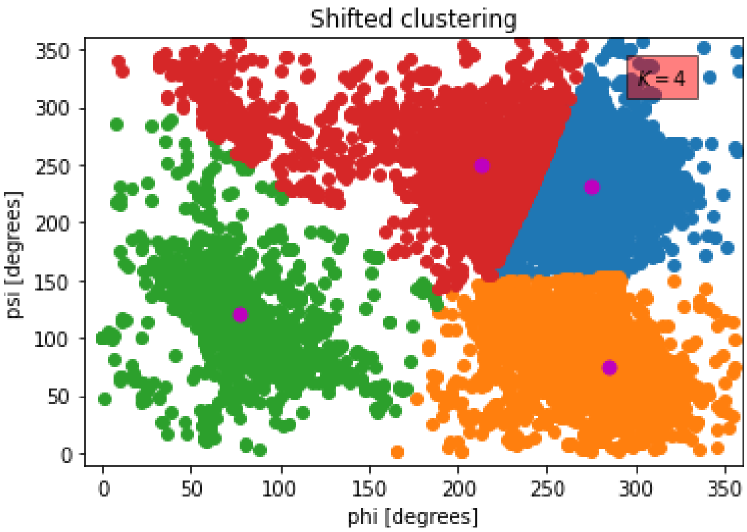
In [ ]:

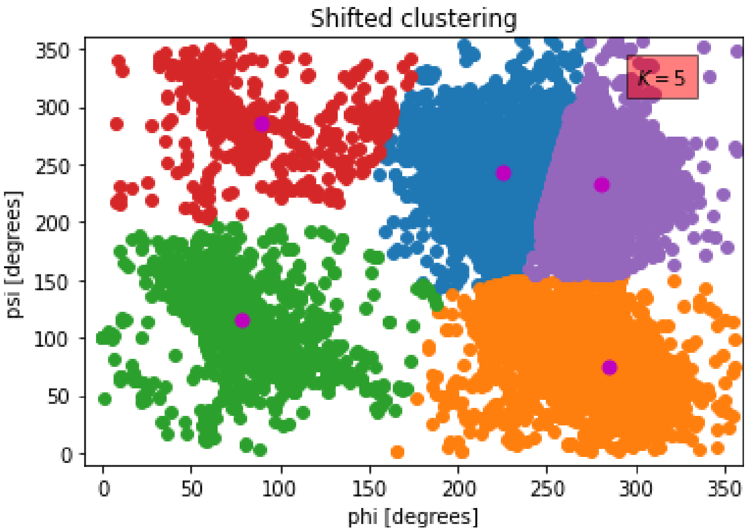
```
#shift all angles so periodic wrap happens in area with few points

psi_shifted = df.psi.apply(lambda x: (x + 100) % 360)
phi_shifted = df.phi.apply(lambda x: x % 360)

shifted_angles = [list(a) for a in zip(phi_shifted, psi_shifted)]

inertias = []

for i in range(1, 7):
    n_clusters = i

    kmean = cluster.KMeans(n_clusters = n_clusters, random_state = 0).fit(shifted_ar

    inertias.append(kmean.inertia_)
    centers = kmean.cluster_centers_
    labels = kmean.labels_

    l, all = [list(a) for a in zip(*sorted(zip(labels, shifted_angles)))]
    all = np.array(all)


    start = 0
    stop = -1
    for i in range(n_clusters):
        try:
            stop = l.index(i + 1)
        except ValueError:
            stop = -1

        plt.scatter(all[start:stop, 0], all[start:stop, 1])
        start = stop

    for j in range(n_clusters):
        plt.plot(centers[j][0], centers[j][1], 'mo', ms = 7)

    plt.text(300, 320, r"$K = $" + str(n_clusters), bbox = {'facecolor': 'red', 'alp
    plt.xlabel("phi [degrees]")
    plt.ylabel("psi [degrees]")
    plt.xlim([-10, 360])
    plt.ylim([-10, 360])
    plt.title("Shifted clustering")
    plt.show()

plt.title("Elbow graph")
plt.ylabel("Inertia")
plt.xlabel("Number of clusters")
plt.plot(list(range(1, 7)), inertias)
plt.show()
```

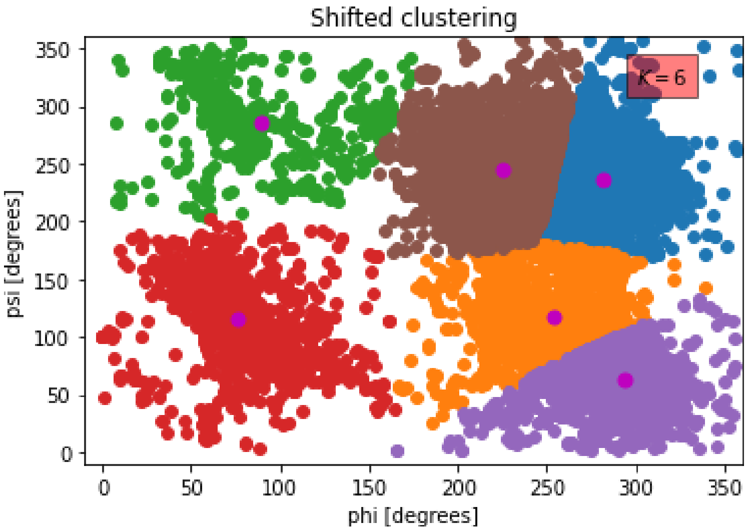


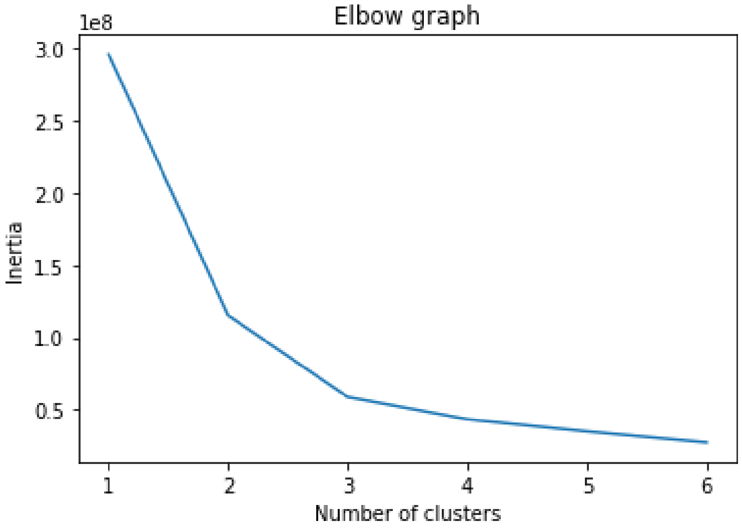





Shifted clustering
$K = 4$
psi [degrees]
0
50
100
150
200
250
300
350
phi [degrees]
0
50
100
150
200
250
300
350

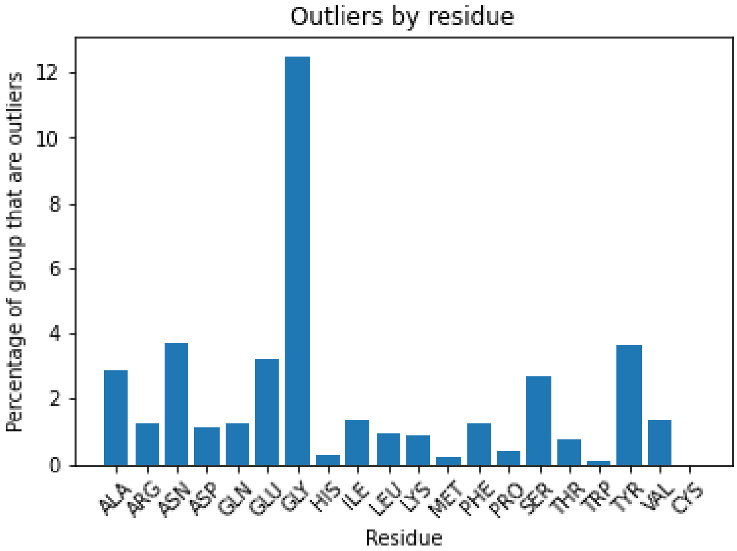

Shifted clustering
$K = 5$
psi [degrees]
0
50
100
150
200
250
300
350
phi [degrees]
0
50
100
150
200
250
300
350

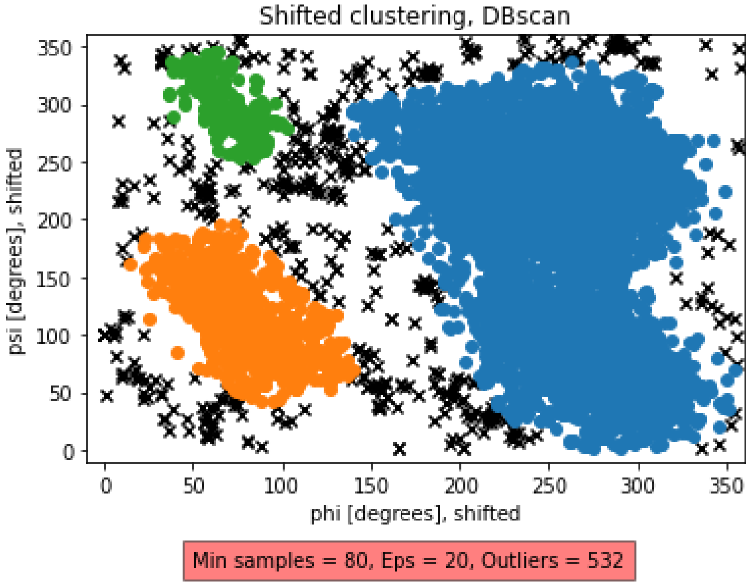

Shifted clustering
$K = 6$
psi [degrees]
0
50
100
150
200
250
300
350
phi [degrees]
0
50
100
150
200
250
300
350



In [ ]:

```
#DB SCAN
for i in range(4):
    min_samples = 40 + 30*i
    eps = 10 + 5*i

    dbscan = cluster.DBSCAN(min_samples=min_samples, eps=eps).fit(shifted_angles)

    labels = dbscan.labels_

    df["label"] = labels

    outliers_grouped = df[df.label == - 1].groupby(["residue name"]).size()
    print()
    for a in df["residue name"]:
        if a not in outliers_grouped:
            outliers_grouped.loc[a] = 0

    plt.bar(outliers_grouped.index , height=(outliers_grouped.values/df.groupby(["re
    plt.xticks(rotation = 45)
    plt.title("Outliers by residue")
    plt.xlabel("Residue")
    plt.ylabel("Percentage of group that are outliers")
    plt.show()

    l, all = [list(a) for a in zip(*sorted(zip(labels, shifted_angles)))]
    n_clusters = l[-1] + 1

    all = np.array(all)
    outliers = l.index(0)

    start = 0
    stop = -1

    marker = None
    color = None
    if l[0] == -1:
        marker = 'x'
        color = 'k'


    for i in range(n_clusters + 1):
        try:
            stop = l.index(i)
        except ValueError:
            stop = -1
        plt.scatter(all[start:stop, 0], all[start:stop, 1], marker = marker, c = col
        marker = None
        color = None
        start = stop

    plt.text(50, -100, f"Min samples = {min_samples}, Eps = {eps}, Outliers = {outli
    plt.xlim([-10, 360])
    plt.ylim([-10, 360])
    plt.title("Shifted clustering, DBscan")
    plt.xlabel("phi [degrees], shifted")
    plt.ylabel("psi [degrees], shifted")
    plt.show()
```


Outliers by residue
Percentage of group that are outliers
ALA
ARG
ASN
ASP
GLN
GLU
GLY
HIS
ILE
LEU
LYS
MET
PHE
PRO
SER
THR
TRP
TYR
VAL
CYS
Residue


Shifted clustering, DBscan
psi [degrees], shifted
phi [degrees], shifted
Min samples = 80, Eps = 20, Outliers = 532