Кольца Илья Вячеславович

Лабораторная работа № 5

Вариант 4

In [10]:

```
import pandas as pd
import warnings
import numpy as np
import scipy.stats as sts
import matplotlib.pyplot as plt
from matplotlib.patches import Ellipse
import matplotlib.transforms as transform
import math
warnings.filterwarnings('ignore')
%matplotlib inline
```

In [6]:

```
df = pd.read_table('spambase.dat', sep=',', index_col=None,header=None)
columns = ['Word_freq_make', 'Word_freq_address', 'Word_freq_all', 'Word_freq_3d', 'Word
_freq_our', 'Word_freq_over', 'Word_freq_remove', 'Word_freq_internet', 'Word_freq_order'
, 'Word_freq_mail', 'Word_freq_receive', 'Word_freq_will', 'Word_freq_people', 'Word_freq
_report', 'Word_freq_addresses', 'Word_freq_free', 'Word_freq_business', 'Word_freq_email
', 'Word_freq_you', 'Word_freq_credit', 'Word_freq_your', 'Word_freq_font', 'Word_freq_00
0', 'Word_freq_money', 'Word_freq_hp', 'Word_freq_hpl', 'Word_freq_george', 'Word_freq_65
0', 'Word_freq_lab', 'Word_freq_labs', 'Word_freq_telnet', 'Word_freq_857', 'Word_freq_da
ta', 'Word_freq_415', 'Word_freq_85', 'Word_freq_technology', 'Word_freq_1999', 'Word_fre
q_parts', 'Word_freq_pm', 'Word_freq_direct', 'Word_freq_cs', 'Word_freq_meeting', 'Word_
freq_original', 'Word_freq_project', 'Word_freq_re', 'Word_freq_edu', 'Word_freq_table',
'Word_freq_conference', 'Char_freq1', 'Char_freq2', 'Char_freq3', 'Char_freq4', 'Char_fr
eq5', 'Char_freq6', 'Capital_run_length_average', 'Capital_run_length_longest', 'Capital_
run_length_total', 'Spam']
df.columns = columns
df.head()
```

Out[6]:

	Word_freq_make	Word_freq_address	Word_freq_all	Word_freq_3d	Word_freq_our	Word_freq_over	Word_freq_remove	Wo
0	0.00	0.64	0.64	0.0	0.32	0.00	0.00	
1	0.21	0.28	0.50	0.0	0.14	0.28	0.21	
2	0.00	0.00	0.00	0.0	0.63	0.00	0.31	
3	0.00	0.00	0.00	0.0	1.85	0.00	0.00	
4	0.00	0.00	0.00	0.0	1.92	0.00	0.00	

5 rows × 58 columns

In [ ]:

```

```

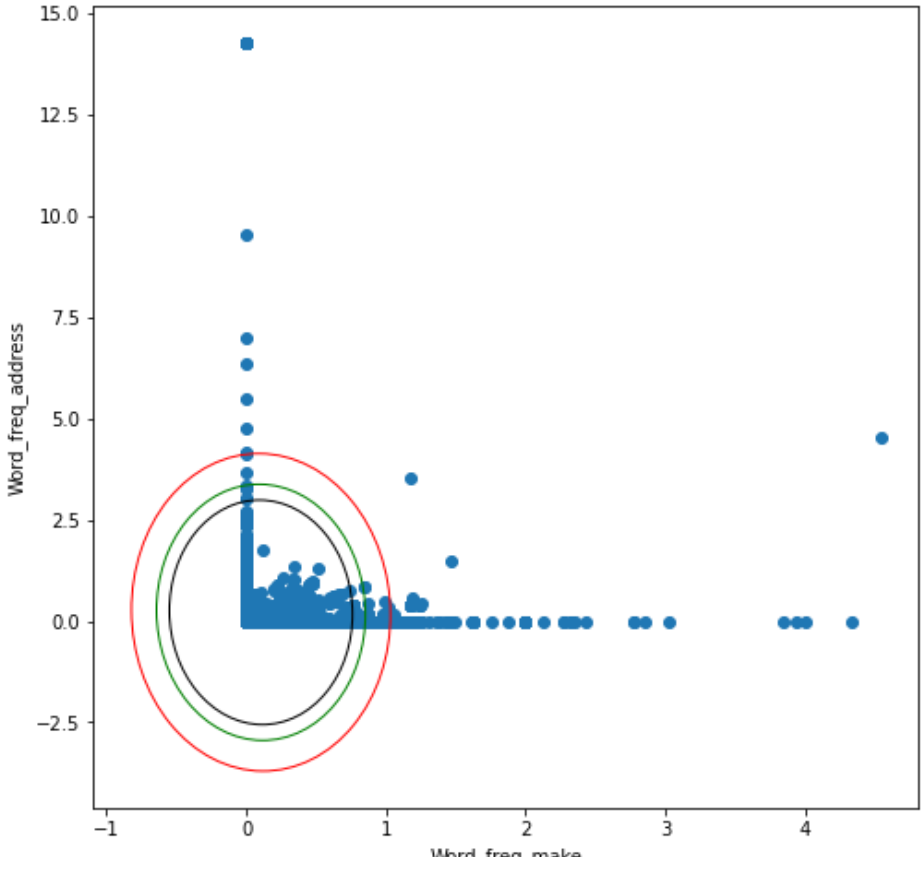
In [11]:

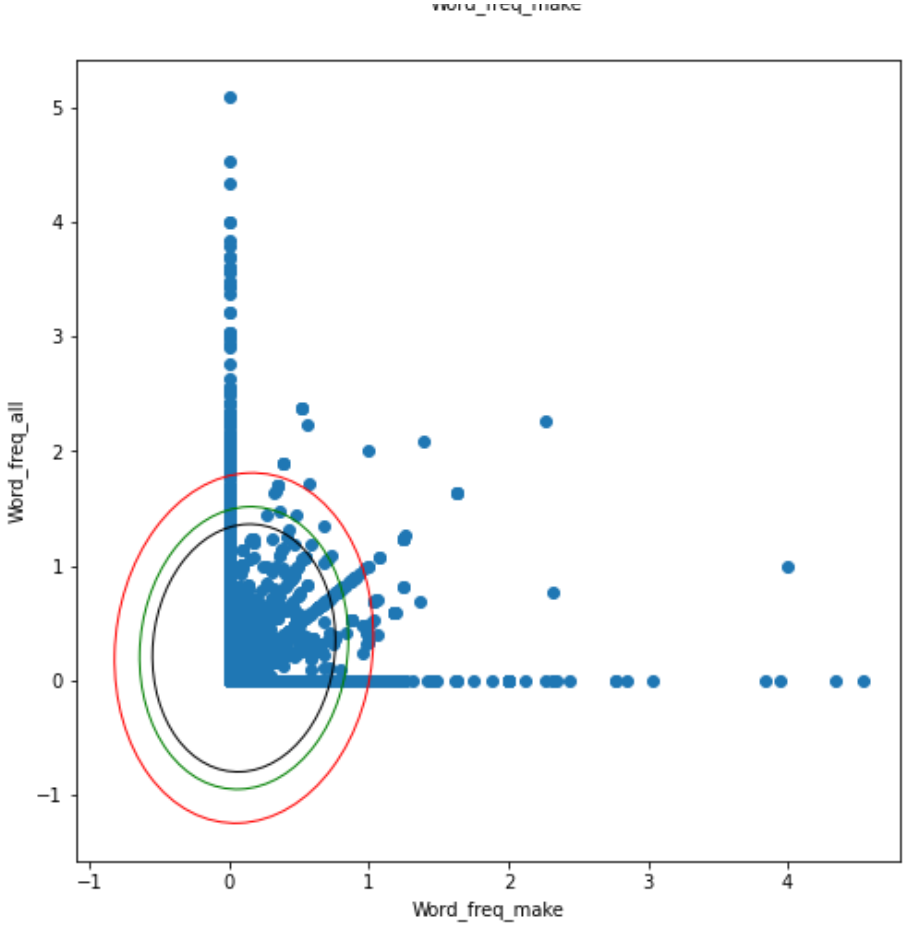
```
def confidence_ellipse(x, y, ax, p_value, facecolor='none', **kwargs):
    if x.size != y.size:
        raise ValueError("x и y должны иметь одинкаковый size")

    cov = np.cov(x,y)
```

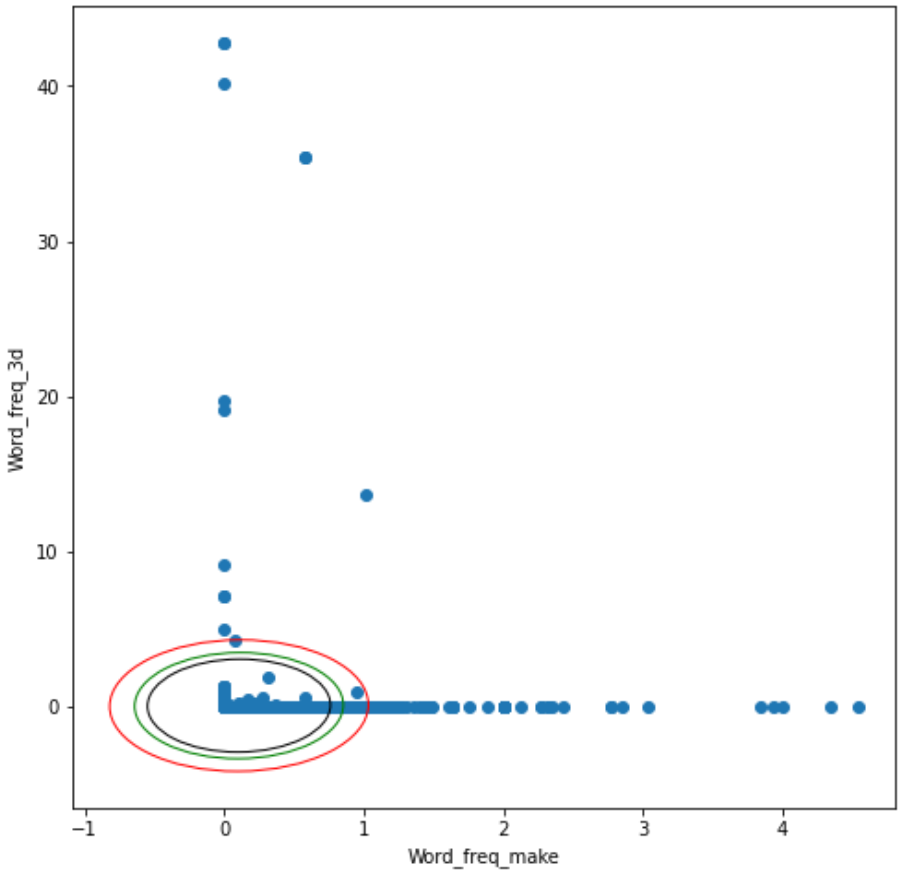
```
    pearson = cov[0, 1]/np.sqrt(cov[0, 0]*cov[1, 1])

    ell_radius_x = np.sqrt(1 + pearson)
    ell_radius_y = np.sqrt(1 - pearson)
    ellipse = Ellipse(
        (0, 0),
        width=ell_radius_x*2,
        height=ell_radius_y*2,
        facecolor=facecolor,
        **kwargs
    )

    if p_value < 0 or p_value > 1:
        raise ValueError

    n_std = math.sqrt(-2 * math.log(p_value))
    scale_x = np.sqrt(cov[0, 0]) * n_std
    mean_x = np.mean(x)

    scale_y = np.sqrt(cov[1, 1]) * n_std
    mean_y = np.mean(y)

    transf = transform.Affine2D().rotate_deg(45).scale(scale_x, scale_y).translate(mean_
x, mean_y)

    ellipse.set_transform(transf + ax.transData)
    return ax.add_patch(ellipse)
```

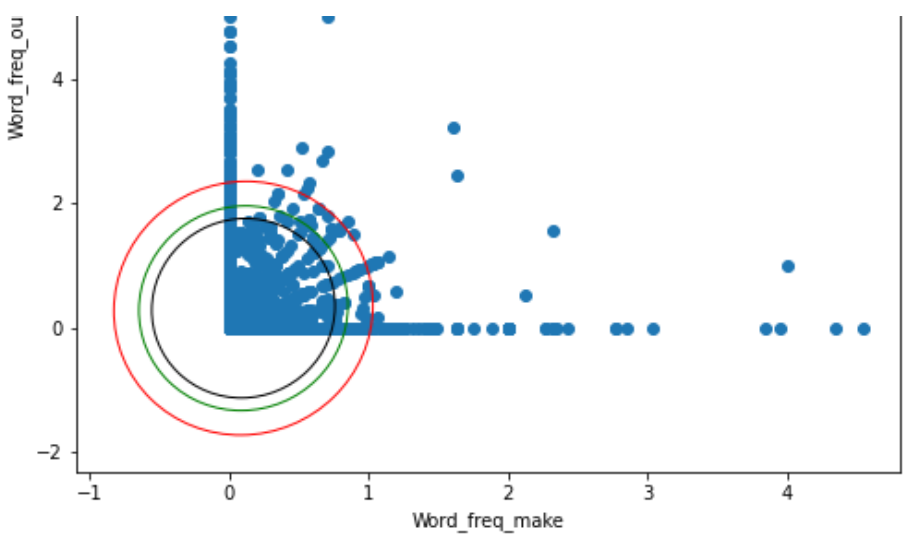
In [12]:

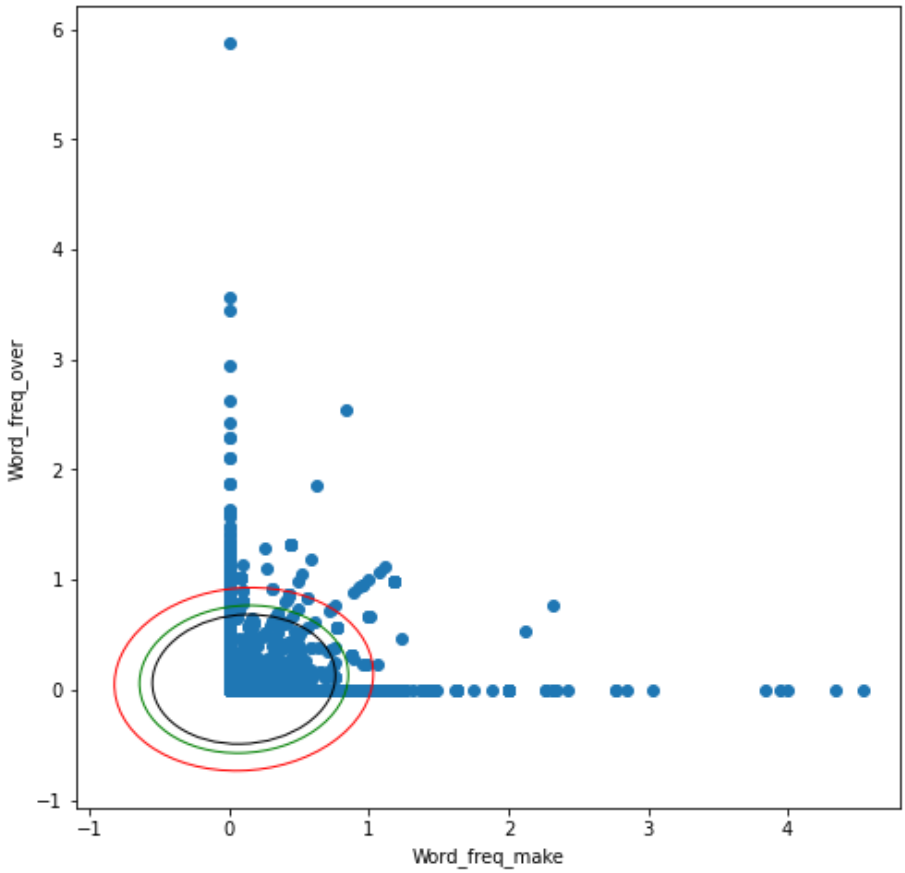
```
def print_ellipse(n, m):
    for i in df.columns[df.columns != 'Spam'][:n]:
        for j in df.columns[(df.columns != 'Spam') & (df.columns != i)][:m]:
            fig, ax = plt.subplots(figsize=(8, 8))
            ax.set_xlabel(i)
            ax.set_ylabel(j)
            confidence_ellipse(df[i], df[j], ax, 0.01, edgecolor='red')
            confidence_ellipse(df[i], df[j], ax, 0.05, edgecolor='green')
            confidence_ellipse(df[i], df[j], ax, 0.10, edgecolor='black')
            ax.scatter(df[i], df[j])
    plt.show()

print_ellipse(2, 10)
```

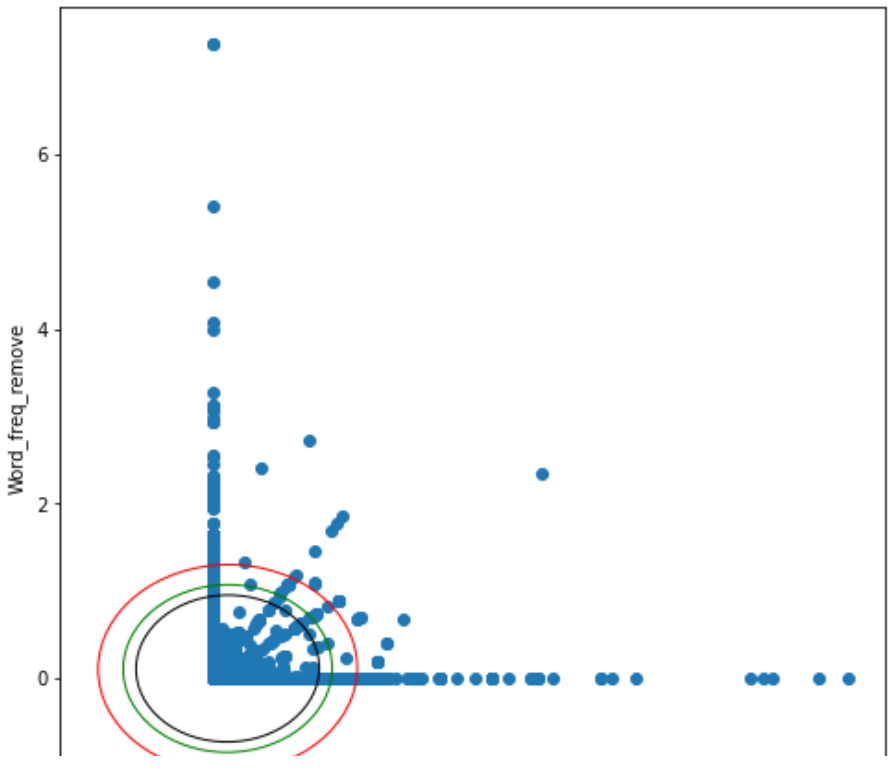


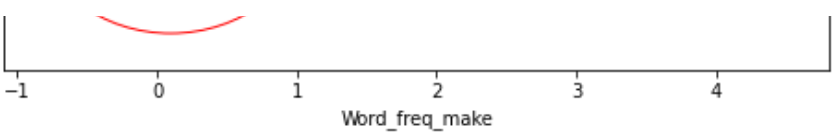

Word_freq_all
Word_freq_make

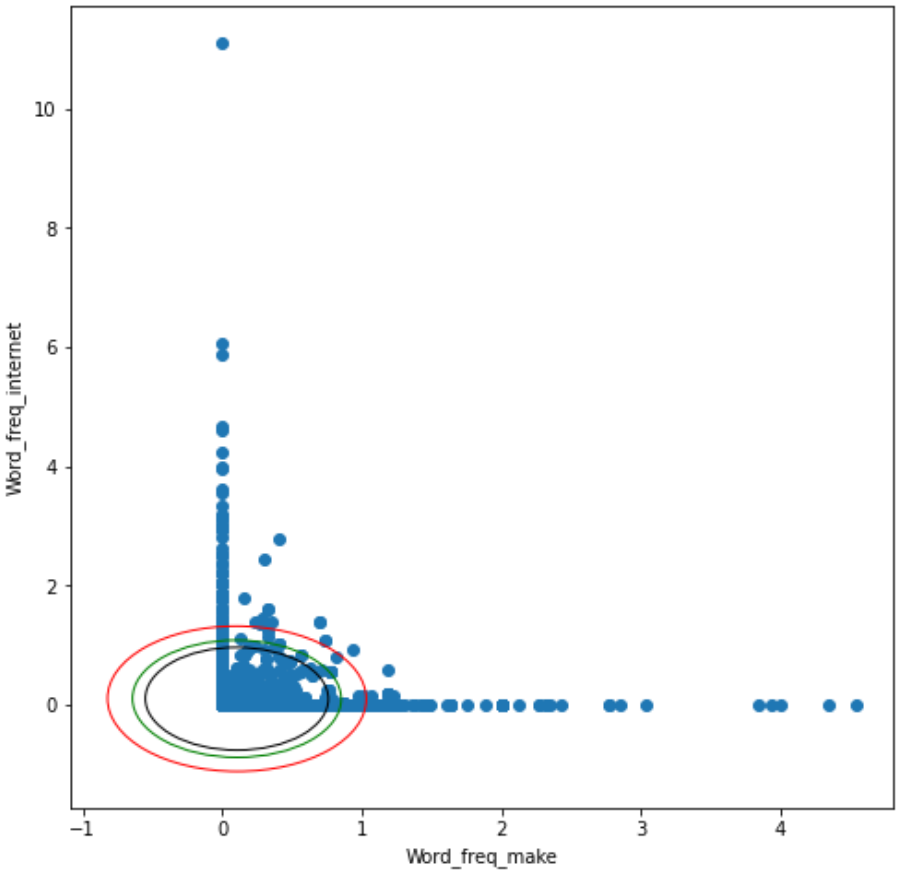

Word_freq_3d
Word_freq_make

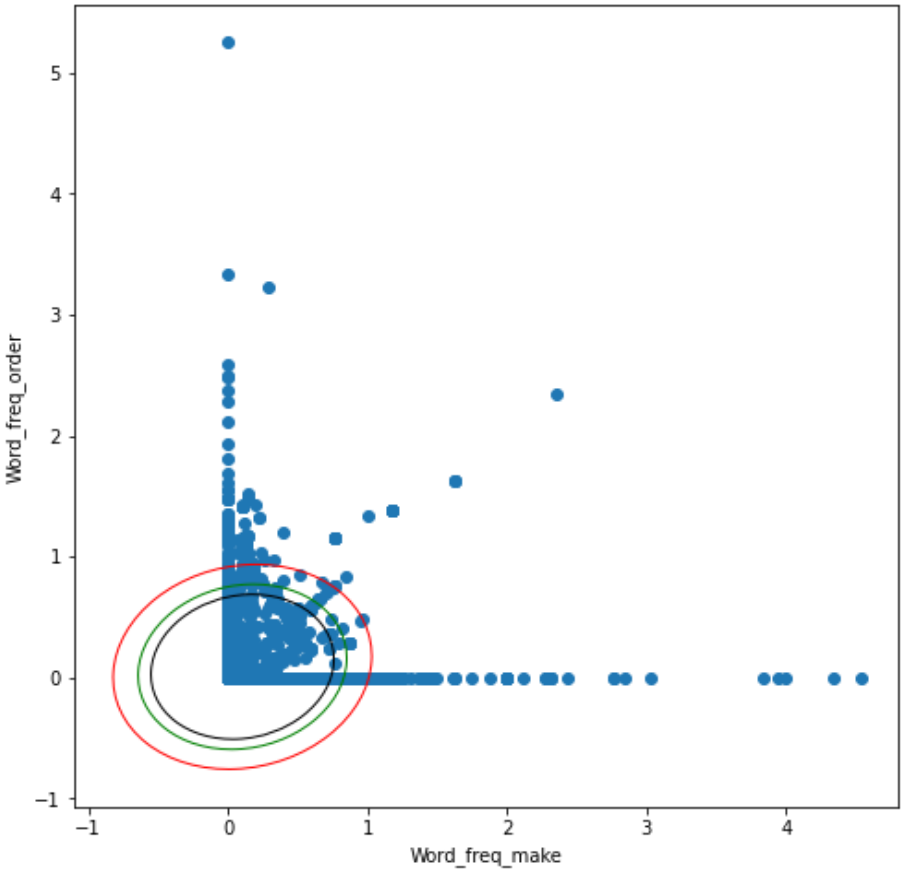

Word_freq_ou
4
2
0
−2
−1
0
1
2
3
4
Word_freq_make


6
5
4
3
2
1
0
−1
Word_freq_over
−1
0
1
2
3
4
Word_freq_make

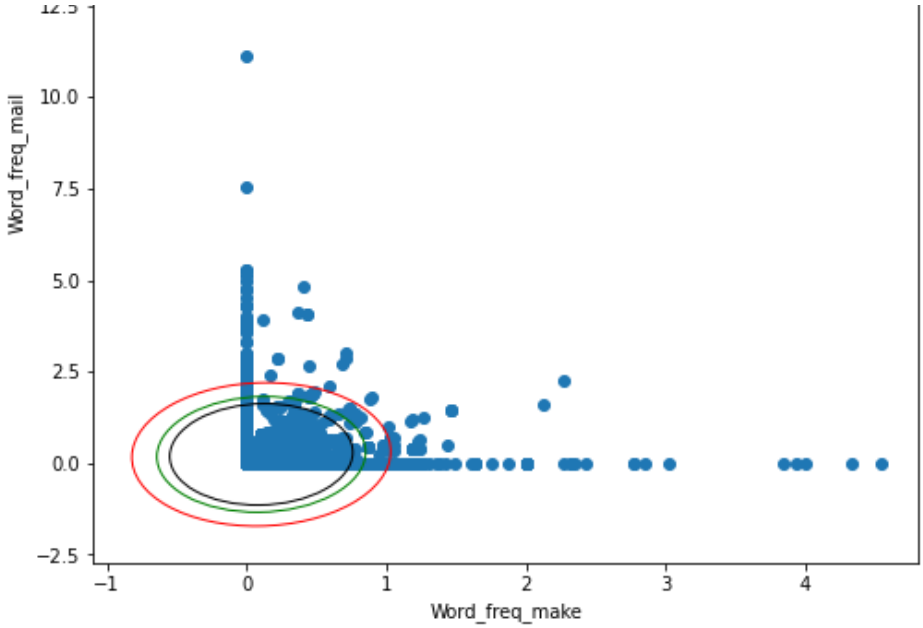

6
4
2
0
Word_freq_remove

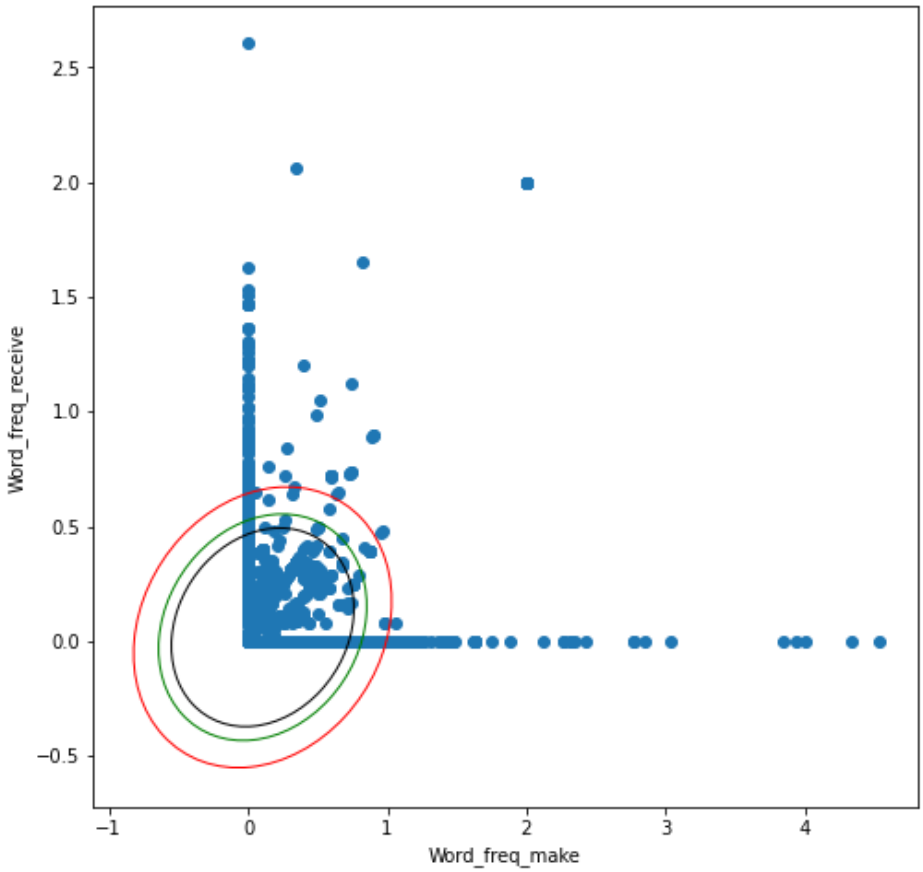

−1
0
1
2
3
4
Word_freq_make

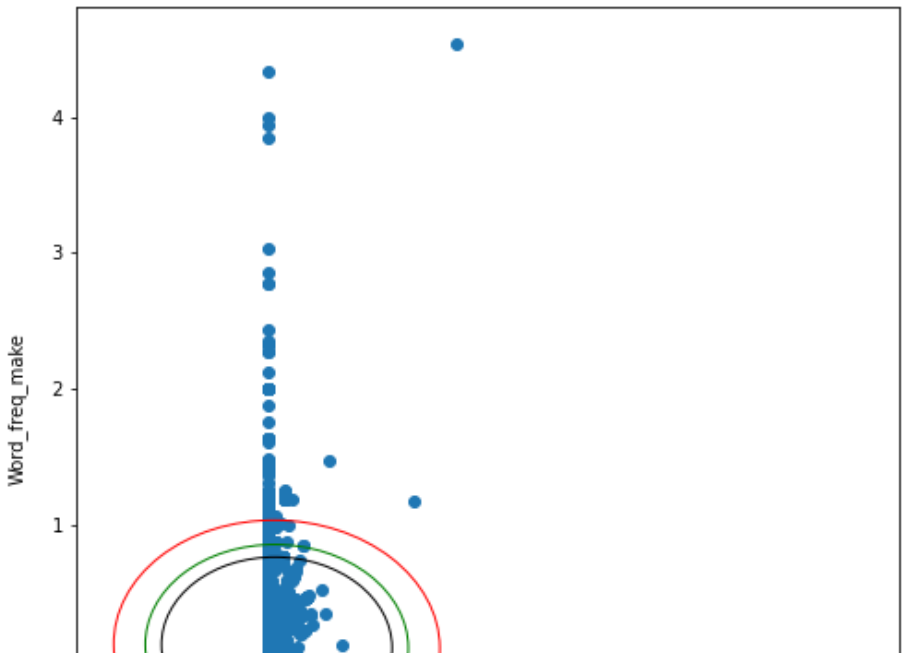

10
8
6
4
2
0
Word_freq_internet
−1
0
1
2
3
4
Word_freq_make

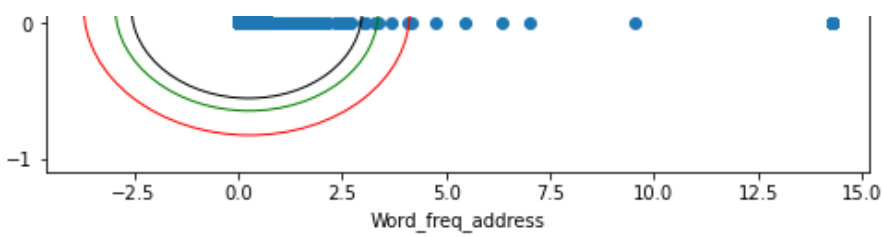

5
4
3
2
1
0
−1
Word_freq_order
−1
0
1
2
3
4
Word_freq_make

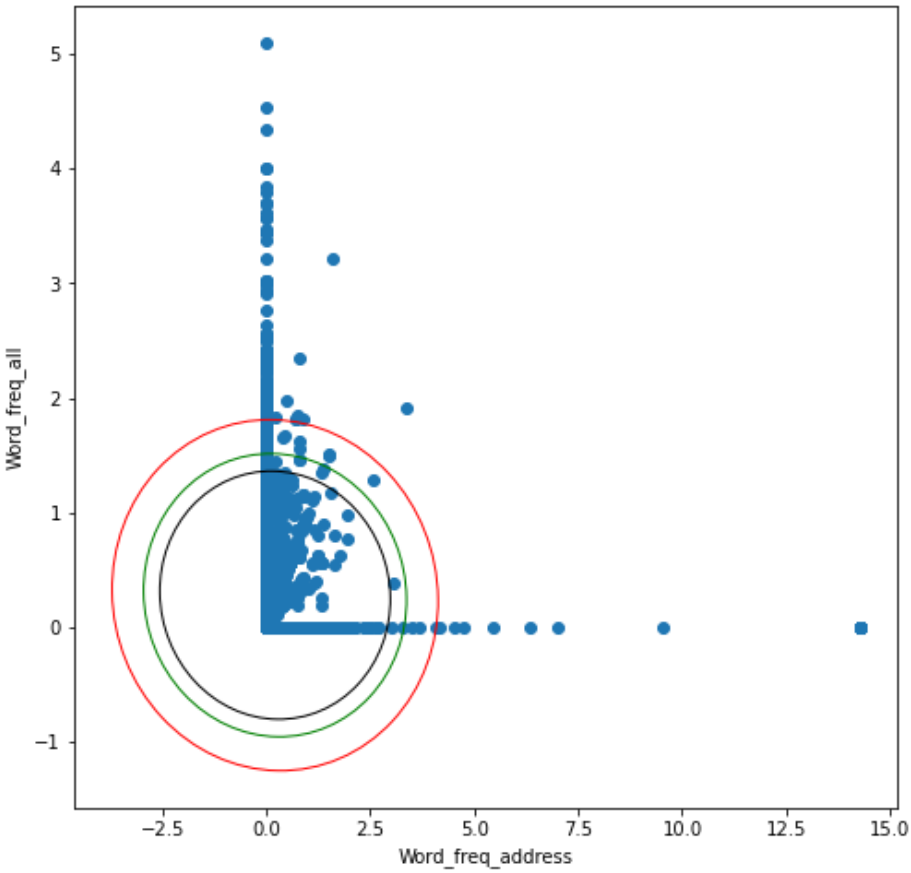

17.5
15.0

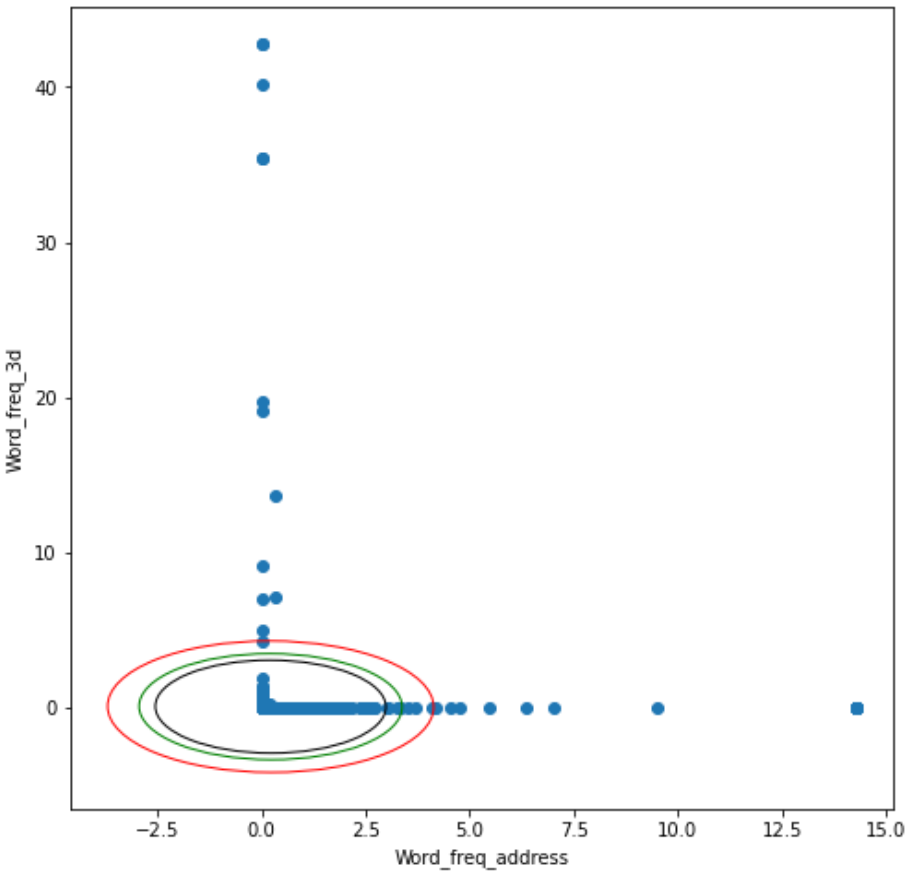

Word_freq_mail
Word_freq_make
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−1
0
1
2
3
4


Word_freq_receive
Word_freq_make
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
−1
0
1
2
3
4

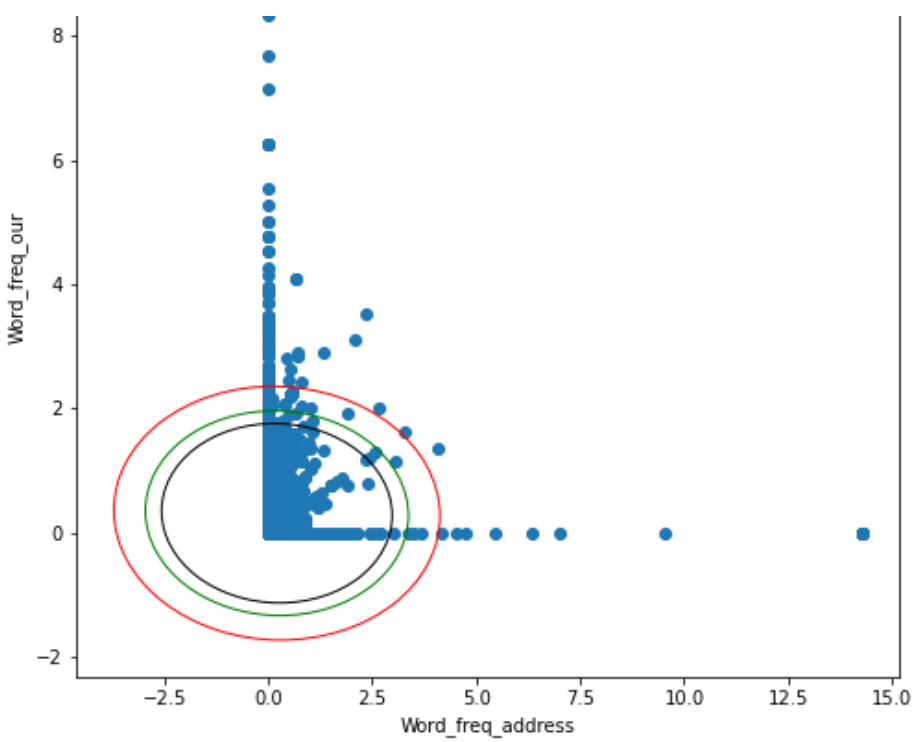

Word_freq_make
4
3
2
1

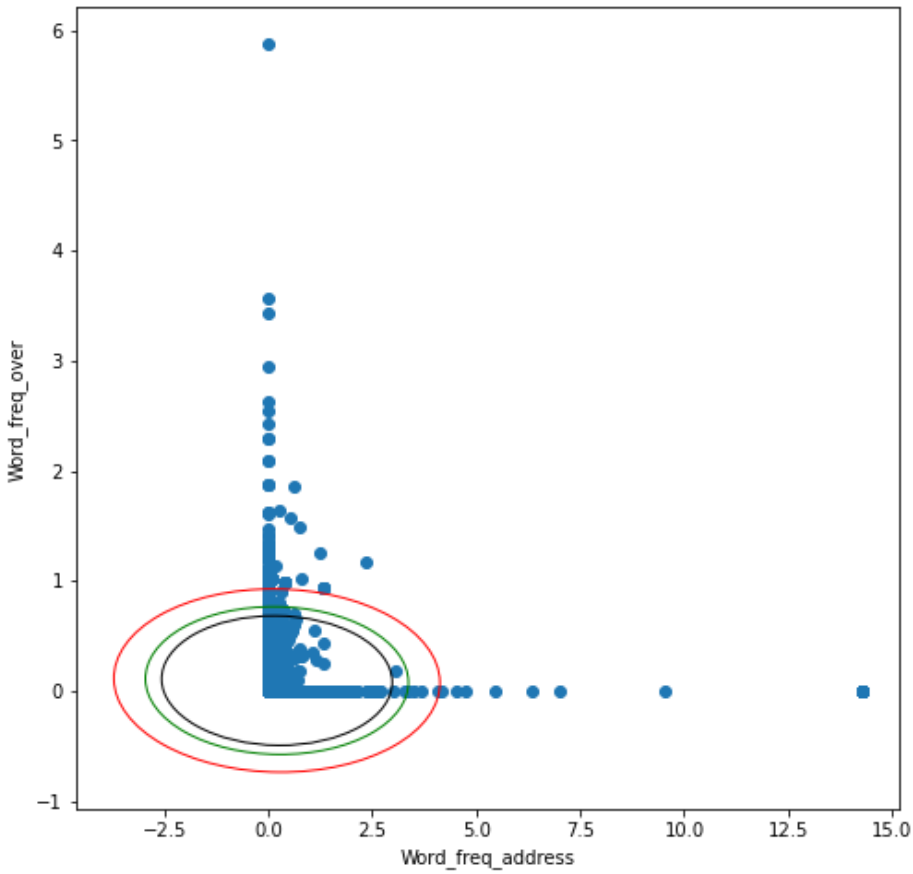

0
−1
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0
Word_freq_address

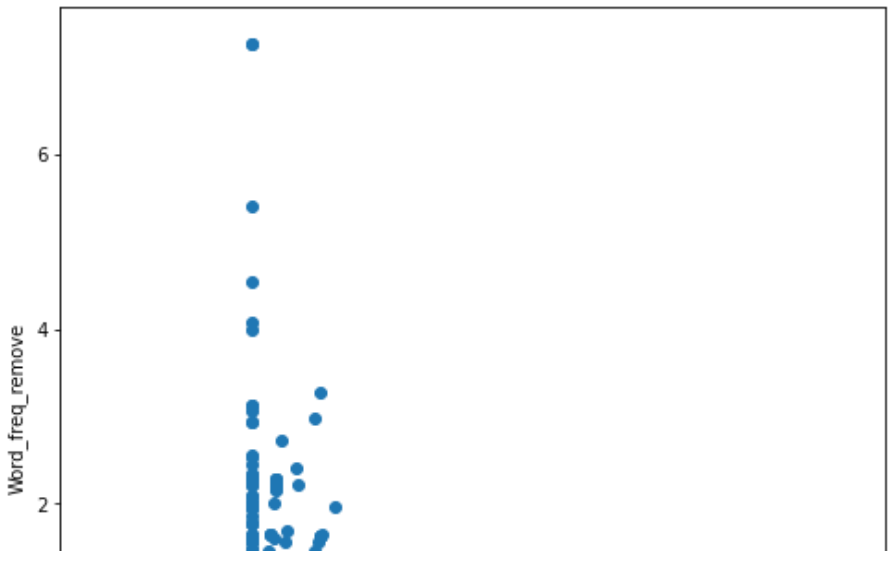

5
4
3
2
1
0
−1
Word_freq_all
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0
Word_freq_address

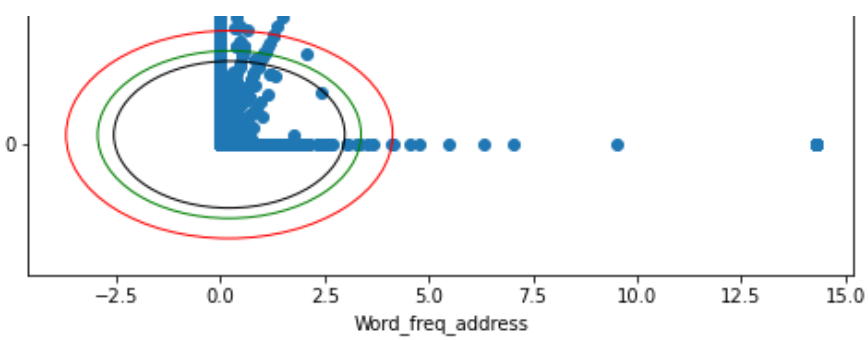

40
30
20
10
0
Word_freq_3d
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0
Word_freq_address

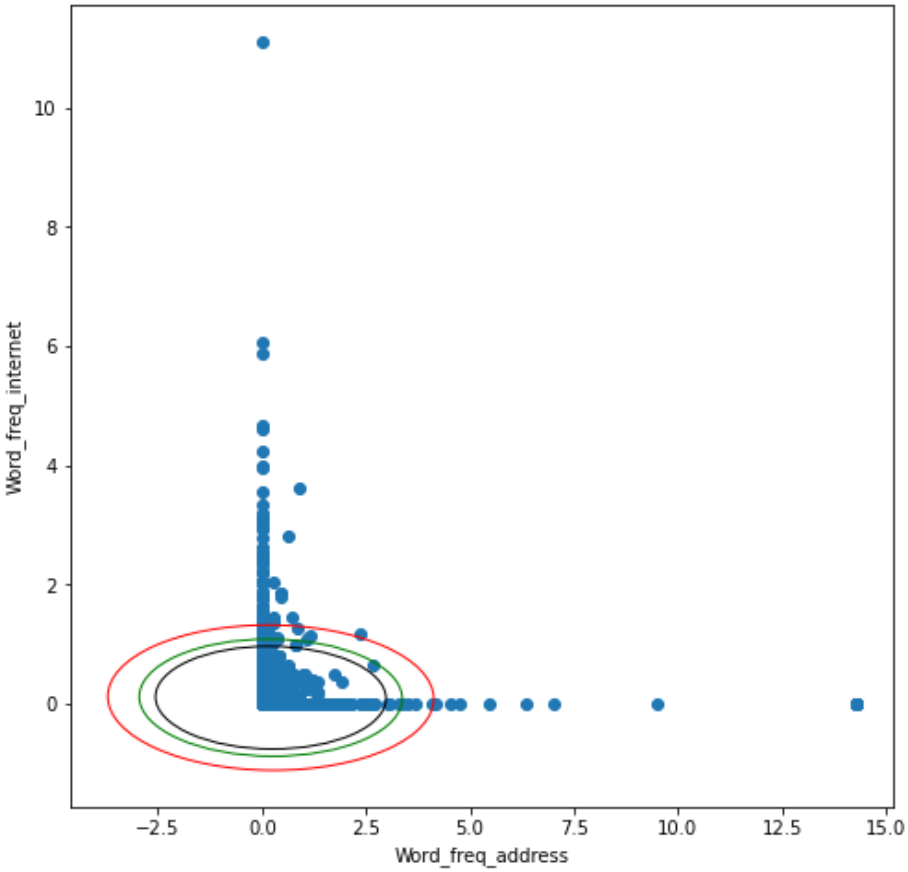

10

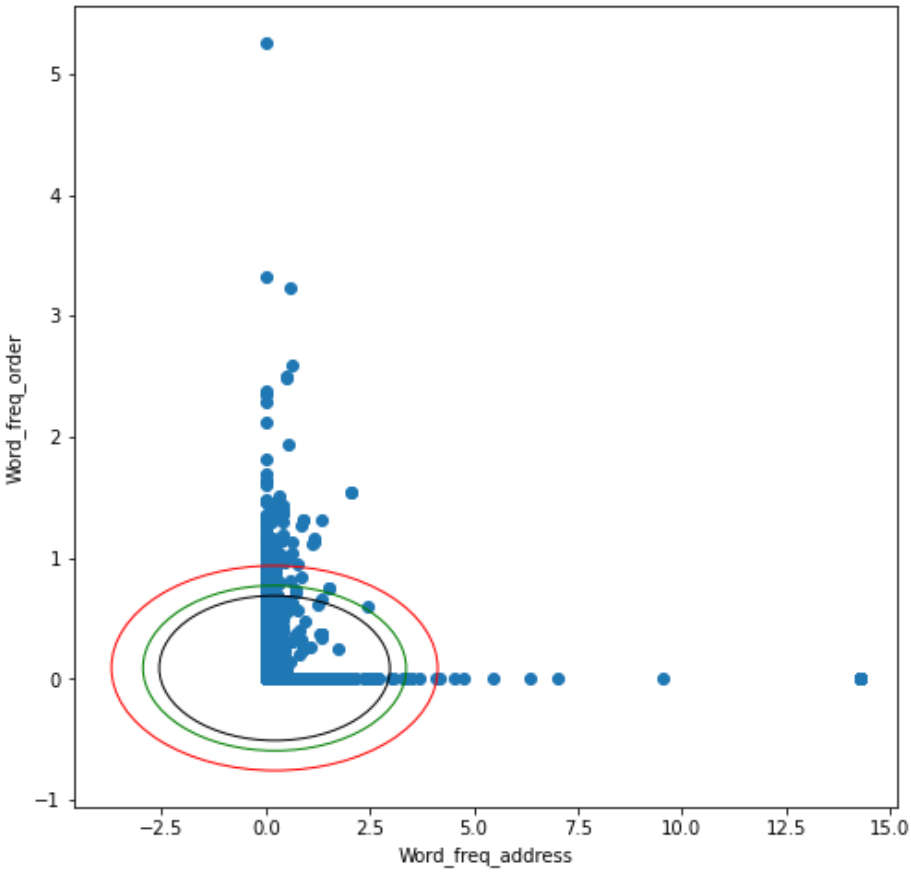

Word_freq_our
Word_freq_address
8
6
4
2
0
−2
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0

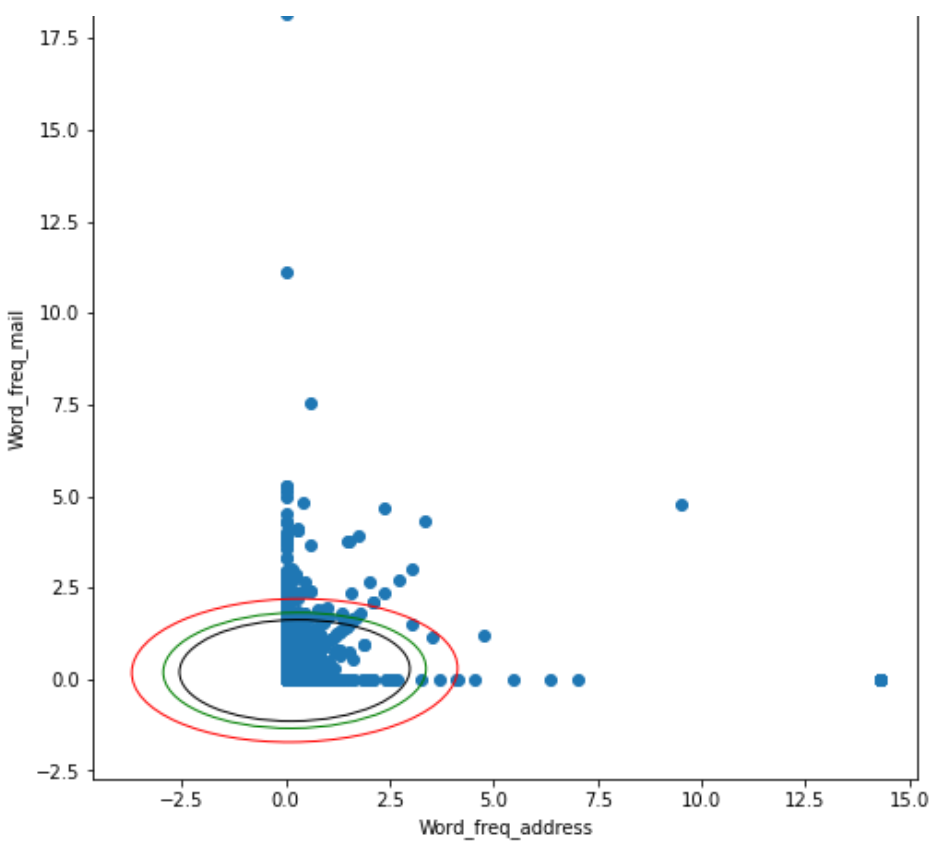

Word_freq_over
Word_freq_address
6
5
4
3
2
1
0
−1
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0

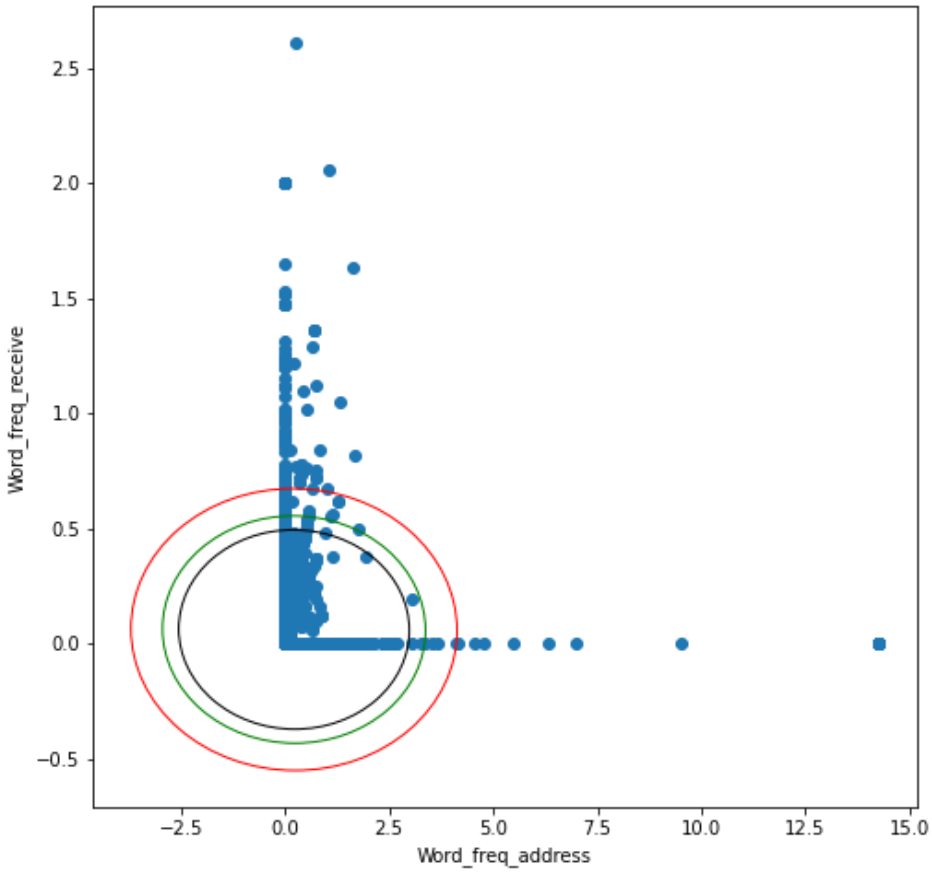

Word_freq_remove
6
4
2


0
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0
Word_freq_address


Word_freq_internet
10
8
6
4
2
0
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0
Word_freq_address


Word_freq_order
5
4
3
2
1
0
−1
−2.5
0.0
2.5
5.0
7.5
10.0
12.5
15.0
Word_freq_address





In [13]:

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
df[df.columns[df.columns != 'Spam']] = scaler.fit_transform(df[df.columns[df.columns != 'Spam']])
df
```

Out[13]:

	Word_freq_make	Word_freq_address	Word_freq_all	Word_freq_3d	Word_freq_our	Word_freq_over	Word_freq_remove
0	-0.342532	0.330601	0.713333	-0.04692	0.011273	-0.350224	-0.291932

1	**Word_freq_address**	**Word_freq_addresses**	**Word_freq_all**	**Word_freq_3d**	**Word_freq_our**	**Word_freq_over**	**Word_freq_remove**
2	-0.342532	-0.165146	-0.556341	-0.04692	0.472113	-0.350224	0.499784
3	-0.342532	-0.165146	-0.556341	-0.04692	2.285742	-0.350224	-0.291932
4	-0.342532	-0.165146	-0.556341	-0.04692	2.389803	-0.350224	-0.291932
...	...	...	...	...	...	...	...
4592	-0.342532	-0.165146	0.713333	-0.04692	-0.464433	-0.350224	-0.291932
4593	-0.342532	-0.165146	-0.556341	-0.04692	-0.464433	-0.350224	-0.291932
4594	-0.342532	-0.165146	-0.556341	-0.04692	-0.464433	-0.350224	-0.291932
4595	0.214032	-0.165146	0.792687	-0.04692	0.041005	0.891087	-0.291932
4596	-0.342532	-0.165146	1.923491	-0.04692	3.252020	-0.350224	-0.291932

4597 rows × 58 columns

In [18]:

```
# Компоненты, объясняющие 95% изменчивости
from sklearn.decomposition import PCA

pca = PCA(n_components=57)
table = pca.fit_transform(df.drop('Spam', axis=1))
comp = pca.components_
explained_var = pca.explained_variance_
explained_var2 = pca.explained_variance_ratio_
n = 0
while sum(explained_var2[:n]) < 0.95:
    n += 1
print(f'{n} главных компонент объясняют {sum(explained_var2[:n])} дисперсии.')
```

```
48 главных компонент объясняют 0.95271864041545 дисперсии.
```

In [15]:

```
cov = pca.get_covariance()
cov
```

Out[15]:

```
array([[ 1.00021758e+00, -1.68025676e-02,  6.58282052e-02, ...,
         4.44762257e-02,  6.13498106e-02,  8.91304741e-02],
       [-1.68025676e-02,  1.00021758e+00, -3.34761629e-02, ...,
         2.06929837e-03,  2.50131833e-04, -2.27032964e-02],
       [ 6.58282052e-02, -3.34761629e-02,  1.00021758e+00, ...,
         9.75180033e-02,  1.07632661e-01,  7.03600257e-02],
       ...,
       [ 4.44762257e-02,  2.06929837e-03,  9.75180033e-02, ...,
         1.00021758e+00,  4.92740185e-01,  1.62339309e-01],
       [ 6.13498106e-02,  2.50131833e-04,  1.07632661e-01, ...,
         4.92740185e-01,  1.00021758e+00,  4.75571049e-01],
       [ 8.91304741e-02, -2.27032964e-02,  7.03600257e-02, ...,
         1.62339309e-01,  4.75571049e-01,  1.00021758e+00]])
```

In [16]:

```
#Кайзер
print("Число компонет по методу Кайзера:")
len(explained_var[explained_var >= 1])
```

```
Число компонет по методу Кайзера:
```

Out[16]:

```
20
```

In [20]:

```
# Компоненты по Кайзеру
```

```
pca = PCA(n_components=20)
table = pca.fit_transform(df.drop("Spam", axis=1))
pd.DataFrame(pca.components_.T)
```

Out[20]:

	0	1	2	3	4	5	6	7	8	9	10	11	
0	-0.043685	0.169399	-0.064332	-0.097097	0.021577	0.028225	-0.015486	0.210651	-0.272708	-0.136579	0.091783	0.247722	0
1	-0.011123	-0.017070	-0.009835	0.043066	-0.070820	-0.030315	0.039577	-0.094976	0.052753	0.055055	0.073017	0.276427	0
2	-0.047123	0.165656	-0.021305	-0.042509	0.101406	-0.113031	-0.048483	0.111796	0.033034	0.153340	0.285720	-0.139712	0
3	-0.006229	0.010777	0.011809	0.040338	-0.045703	0.029310	-0.005422	0.012176	-0.004153	0.042886	-0.036756	0.211645	0
4	-0.036760	0.121642	-0.137243	0.089186	0.274592	0.074134	-0.097343	-0.006337	0.262081	0.151965	-0.044635	0.000861	0
5	-0.045844	0.167805	0.005642	-0.060584	-0.083192	-0.137270	-0.145652	-0.062084	0.115254	0.017558	-0.121073	-0.150907	0
6	-0.046228	0.144570	-0.129924	0.014431	-0.025199	0.188365	0.026846	-0.129826	0.224934	0.234025	0.067582	0.066315	0
7	-0.033924	0.132267	-0.046079	-0.112627	-0.003327	0.195901	0.049585	-0.253998	0.121026	-0.115116	-0.228838	-0.191433	0
8	-0.045562	0.234904	0.130366	-0.055797	0.052958	-0.010601	0.042656	-0.169257	-0.008097	-0.182080	-0.005385	0.048444	0
9	-0.020227	0.153603	0.059762	-0.097742	0.010881	0.081936	-0.051933	-0.109133	-0.022381	-0.046966	0.103533	0.037513	0
10	-0.050376	0.218744	-0.101713	-0.094259	0.065855	0.269712	0.044847	-0.090934	-0.074152	-0.176526	0.002260	0.161779	0
11	-0.022207	0.076269	-0.093654	-0.035913	0.323718	0.067484	-0.113518	0.115201	-0.204521	-0.174668	0.225367	-0.134806	0
12	-0.037566	0.125129	-0.010087	-0.081967	-0.059469	-0.240888	-0.103124	0.125666	-0.106827	-0.104827	-0.188047	-0.028242	0
13	-0.017955	0.068913	0.061149	0.045565	0.037316	-0.018815	0.071299	0.068248	-0.242690	-0.219906	-0.255267	-0.037720	0
14	-0.030902	0.217254	0.149059	-0.159214	-0.002689	-0.311039	-0.230191	-0.215909	0.206259	0.005078	0.071168	0.055747	0
15	-0.042789	0.101652	-0.126778	0.047975	-0.046753	0.098241	0.053468	0.031144	0.107757	0.202634	-0.032070	0.311466	0
16	-0.045230	0.199130	-0.112957	-0.051907	0.042489	0.285095	0.106819	-0.166094	0.097413	-0.056915	-0.155349	-0.193903	0
17	-0.019461	0.168678	-0.049814	-0.109622	-0.049988	-0.141488	-0.110156	-0.153229	0.123356	0.127637	0.332066	0.175463	0
18	-0.083449	0.182582	-0.247601	-0.124588	-0.101236	0.033905	-0.063345	0.300669	-0.108547	0.138015	0.046254	-0.091665	0
19	-0.030774	0.133779	-0.007950	0.020263	0.015171	0.267764	0.133135	-0.121550	0.031375	-0.116354	-0.149371	-0.118497	0
20	-0.075635	0.272648	-0.203539	-0.098563	0.015466	0.200109	0.036471	0.049825	-0.074547	-0.009905	0.042482	0.050927	0
21	-0.014648	-0.005885	0.153144	0.297421	-0.246341	0.285268	-0.441945	0.006128	-0.048428	0.005071	-0.031503	0.018374	0
22	-0.053331	0.247819	0.062840	-0.104019	-0.063698	-0.276894	-0.255739	-0.090238	0.085098	-0.035301	-0.145465	-0.070645	0
23	-0.039925	0.148519	-0.081458	-0.057362	-0.078152	-0.004509	-0.001813	0.160135	-0.239007	-0.069843	0.059596	0.348124	0
24	0.211572	-0.054277	0.074889	-0.052862	0.055237	0.048726	-0.055830	-0.161929	-0.121199	-0.010903	0.207057	-0.194180	0

	0	1	2	3	4	5	6	7	8	9	10	11	
25	0.207940	-0.050061	0.067842	-0.087264	0.104964	0.036315	-0.064749	-0.150429	-0.114530	-0.029993	0.229385	-0.156429	
26	0.038724	-0.116912	-0.018458	0.116507	-0.112027	-0.153656	0.161050	-0.133679	0.024046	0.028639	-0.226339	0.272691	
27	0.278579	0.042019	0.000008	-0.004092	-0.026858	-0.015389	0.040054	-0.011033	-0.008853	-0.020646	0.092974	-0.033219	
28	0.219285	0.032534	-0.080139	0.111332	0.306665	-0.015464	-0.170893	0.163837	0.106204	-0.072623	-0.137686	0.089069	
29	0.303487	0.062695	-0.026432	-0.004705	-0.050543	0.016496	-0.010343	-0.028617	0.006271	-0.029268	0.112966	-0.048035	
30	0.311910	0.090778	-0.050746	0.013629	-0.057655	-0.006718	0.012431	0.010551	0.002281	0.000197	-0.026096	0.029055	
31	0.348439	0.111856	-0.056025	0.013054	-0.065675	0.009055	0.010732	0.063825	0.023619	0.024005	-0.109456	0.058618	
32	0.005369	-0.084844	0.088082	-0.020794	0.091223	-0.044140	0.019977	-0.144046	-0.071095	0.019032	-0.038064	0.106268	
33	0.347536	0.112808	-0.053674	0.014010	-0.065095	0.009494	0.013069	0.066225	0.022902	0.025564	-0.108987	0.058841	
34	0.268528	0.053390	-0.025201	0.009775	-0.032822	-0.017541	0.012535	-0.021401	0.010497	-0.047237	0.066024	-0.013878	
35	0.316781	0.074820	-0.034185	0.009235	-0.043144	-0.002062	0.014937	-0.016421	-0.025956	-0.004454	0.001816	-0.014119	
36	0.046208	-0.128720	0.221607	-0.398651	0.092497	0.187797	-0.080101	-0.040235	-0.024526	0.076488	-0.040534	0.087726	
37	-0.002016	-0.018218	-0.025173	0.083023	0.260998	0.000517	-0.095536	0.001661	0.155536	0.001676	-0.063948	0.091747	
38	0.036849	-0.084066	0.123504	-0.302619	0.128908	0.097345	-0.074260	0.048102	-0.152513	0.228182	-0.153324	0.068463	
39	0.320152	0.138705	-0.056496	0.003622	-0.064703	-0.006026	0.002144	0.026907	0.048671	0.014166	-0.097145	0.046568	
40	0.008612	-0.086351	0.109343	-0.245505	-0.124351	0.109973	-0.009526	0.251364	0.379334	-0.326304	0.119158	0.049696	
41	0.024242	-0.073861	-0.050323	0.135550	0.535838	-0.030608	-0.254950	0.160838	0.156090	-0.070863	-0.128858	0.094227	
42	0.065097	-0.070779	0.149151	-0.405143	0.081795	0.133313	-0.129851	-0.029572	-0.069210	0.198644	-0.151786	0.153668	
43	0.011608	-0.057088	-0.002269	0.042124	0.070005	-0.038174	0.040047	-0.056150	-0.036304	-0.069846	-0.077701	-0.048975	
44	0.008918	-0.092634	-0.024918	-0.180921	-0.049939	-0.002634	0.031389	0.192442	-0.030816	0.311373	-0.171093	-0.226956	
45	-0.002973	-0.108327	0.083283	-0.215188	-0.172259	0.055630	0.054001	0.335019	0.403648	-0.320247	0.067417	-0.002571	
46	0.002971	-0.008487	0.000983	-0.011269	0.014670	-0.018219	-0.025566	0.010016	-0.098036	0.011882	0.185108	-0.054984	
47	0.003533	-0.053294	0.021210	0.000925	0.060184	-0.029578	0.016731	-0.045007	-0.094410	-0.055854	0.112027	-0.066288	
48	-0.002023	-0.040639	0.201849	0.195695	-0.199973	0.287074	-0.414275	0.080540	-0.051665	-0.000418	-0.003829	-0.000030	
49	0.140713	0.061925	0.291230	0.016972	0.054913	0.008756	0.167074	0.164826	0.017538	0.121563	-0.040755	-0.028831	
50	0.018434	-0.050406	0.108884	-0.142009	0.051028	0.032957	-0.068838	-0.153490	-0.034216	0.045995	-0.042942	0.135921	
51	-0.044644	0.120946	-0.043238	-0.001625	-0.087224	0.038528	-0.023032	0.232958	0.019023	0.297380	-0.040357	-0.215697	
52	-0.051356	0.236069	0.100782	-0.018747	-0.058305	-0.187676	-0.082714	0.100594	-0.068074	0.009788	-0.198231	-0.099614	
53	0.001007	0.028149	0.081540	0.166437	-0.145031				0.018001	0.074324	0.093018	0.012822	

	0	1	2	3	4	5	6	7	8	9	10	11
54	-0.016394	0.135497	0.285428	0.154469	0.130779	0.106727	0.285737	0.112215	0.088684	0.168467	0.138157	0.018674
55	-0.029501	0.252195	0.444817	0.139024	0.115558	0.013886	0.225926	0.127002	0.070235	0.129994	0.058549	0.020071
56	-0.043676	0.225370	0.373126	0.099192	0.032686	-0.022730	0.035739	0.026027	-0.127512	-0.143250	-0.053998	0.021699

In [21]:

```
#новые признаки
df.drop("Spam", axis=1) @ pca.components_.T
```

Out[21]:

	0	1	2	3	4	5	6	7	8	9	10	11
0	-0.731694	-0.042418	-0.579372	0.275785	-0.158930	-0.550722	0.015283	0.031319	0.288896	0.734581	1.261732	0.186586
1	-1.184979	2.067532	0.040801	-0.417820	-0.311988	-0.847389	-0.481976	0.640062	-0.845505	-0.474849	-0.122487	0.146151
2	-0.806288	0.426563	-0.579036	0.021666	-0.045184	0.540751	0.387035	-0.480560	0.200077	-0.351933	-0.429434	-0.030725
3	-0.492776	-0.490047	-0.368708	0.422867	0.439445	0.711374	0.521656	-1.469893	1.387905	-0.238546	-1.331810	-0.872859
4	-1.025857	1.020138	-1.727360	-0.000551	0.919238	1.444387	0.005674	-0.090521	0.060503	-0.274234	0.412262	0.866362
...	...	...	...	...	...	...	...	...	...	...	...	...
4592	-0.240446	-1.437519	0.598757	-1.476768	-0.651428	0.179334	0.373923	1.957817	0.774774	0.725139	-0.022194	-0.791989
4593	-0.306226	-2.282756	0.840331	-1.528081	-1.136189	0.371810	0.496464	1.515212	2.207795	-1.743083	0.181777	0.174680
4594	0.606132	-1.355527	0.203776	0.603586	-0.759628	-0.202020	0.404838	-0.005050	0.540894	-0.515036	0.029352	-0.014310
4595	-0.852773	0.065496	-1.161792	-0.162885	-0.513167	-0.352502	-0.048911	0.748825	-0.016829	0.558378	0.453633	-0.399030
4596	-0.607480	-0.700749	-0.402167	0.088988	0.555383	-0.745835	-0.134695	0.918131	1.730560	-0.136167	0.015583	-0.571040

4597 rows × 20 columns