

Data en MATLAB-code bijlage bachelor opdracht.

D.DOPPENBERG
Universiteit Twente
d.doppenberg@student.utwente.nl,

26 juni 2016

1. LIJST MET VOLGORDEN

Tabel 1: Lijsten met CBS-codes gesorteerd op de volgorde van binnenkomst voor de verschillende volgorden.

Volgorde 1	Volgorde 2	Volgorde 3	Volgorde 4	Volgorde 5	Echte volgorde
512	546	585	716	51	88
1651	1721	1734	589	678	277
1884	893	1842	437	623	96
715	860	614	938	588	339
351	1900	17	1674	353	523
1674	1987	733	983	158	56
1699	17	772	479	1908	504
703	1667	1598	241	1525	58
1724	1926	420	81	537	981
984	1709	184	772	381	1903
638	377	15	262	432	317
209	356	7	5	584	755
744	15	1740	718	1651	147
826	737	852	209	424	376
844	861	1926	1672	860	938
166	944	441	610	416	297
373	668	1916	180	1621	643
399	85	301	22	344	9
737	1904	50	1705	518	163
303	1742	860	553	512	448
606	882	1702	285	502	1663
321	796	1892	870	3	158
532	626	748	512	376	846
56	1700	1507	355	951	756
962	828	163	244	1507	273
1719	388	687	340	225	893
397	203	736	358	1690	848
784	1708	502	395	301	313
22	1701	175	758	209	5

342	184	189	584	362	797
653	1773	1908	547	193	424
786	168	81	1955	197	928
267	707	361	513	417	888
478	56	375	175	406	957
1911	293	473	351	703	907
1676	451	689	858	342	53
748	537	654	15	632	620
482	361	840	263	889	638
432	312	957	1731	370	244
1525	93	197	556	119	664
664	907	796	415	785	7
644	608	34	820	1955	72
352	716	236	457	93	304
1709	1714	643	153	420	779
797	627	603	85	79	736
946	197	1901	327	282	275
758	175	743	504	946	294
629	230	484	1581	758	225
74	994	788	632	392	451
358	114	479	203	644	168
861	1696	294	785	899	556
385	928	532	1926	437	576
118	352	1903	202	171	321
847	599	753	173	173	48
394	1719	584	590	553	370
1509	310	1681	25	981	1742
611	119	209	1729	273	1696
363	1740	147	47	858	476
523	575	1714	1663	243	511
393	289	96	53	213	1700
1684	189	944	140	358	262
1702	765	623	1904	168	503
1771	180	499	321	246	617
626	1598	928	693	321	222
484	1771	738	424	230	707
297	881	351	196	1774	197
1903	946	365	765	177	285
757	1685	150	93	755	994
388	786	1774	90	267	1842
448	1681	396	98	385	765
277	34	664	847	395	425
678	1507	1684	1525	221	828
938	1586	756	1734	17	971
983	402	1891	399	614	340
7	845	511	614	756	299
848	55	416	345	40	63
860	603	597	797	815	363

473	590	88	503	928	1987
788	1676	547	756	861	860
603	848	243	82	1714	226
415	213	425	458	1916	844
398	304	1669	644	852	342
1842	542	588	703	856	716
377	499	214	637	995	47
1685	236	308	606	984	417
96	425	457	545	34	1699
867	715	599	707	779	55
25	396	1896	642	938	396
503	321	757	1699	285	274
177	643	962	1708	441	484
281	879	381	971	1667	202
1671	269	907	86	608	98
643	638	1876	777	545	415
312	579	828	611	737	453
420	216	424	453	1696	703
988	846	545	1586	1734	1900
1908	1731	1884	163	59	542
779	532	703	1773	753	24
556	794	556	275	394	263
1895	772	399	1894	277	303
406	888	994	307	762	585
302	694	352	1842	7	1621
246	642	340	352	638	858
1669	874	482	851	523	478
917	1891	777	222	643	579
402	106	983	301	55	889
1663	971	576	620	451	874
10	158	263	58	668	148
668	757	213	1701	748	310
1891	331	523	150	1672	479
214	5	917	482	513	173
216	504	310	491	400	715
1955	14	632	737	1742	1680
14	505	417	826	405	588
632	141	489	951	373	845
86	501	873	370	988	622
424	244	274	786	203	119
268	951	889	3	757	1641
232	484	1671	281	363	512
392	571	981	1676	274	183
1783	899	622	848	590	737
856	279	230	365	1669	377
654	96	60	233	90	171
893	744	59	228	72	851
613	345	58	243	1641	1709

222	177	820	845	310	738
1701	183	225	342	629	951
1900	1730	879	189	983	826
542	37	1509	1722	252	406
304	273	606	18	37	265
263	448	611	994	1695	352
981	498	1859	678	617	293
1916	166	501	448	327	153
119	585	534	269	450	880
230	358	758	743	1659	109
479	63	114	1695	874	965
568	294	9	88	717	335
356	957	393	376	971	79
1892	395	282	755	303	777
439	855	1525	852	1684	623
828	703	880	873	183	1714
770	1702	303	70	1731	289
164	865	171	339	1655	757
183	47	844	928	399	375
213	147	355	109	402	1876
296	7	37	416	1706	606
1690	365	85	1719	279	279
140	196	82	1895	1926	1684
72	400	874	1696	50	687
150	393	861	96	24	962
677	518	222	1509	147	1955
881	335	627	246	236	744
416	233	356	236	453	1695
718	478	826	294	53	1669
252	637	677	1771	1894	1671
716	50	344	511	1701	358
866	530	385	579	1722	1672
733	748	281	608	60	164
505	301	293	824	397	627
585	398	626	612	765	397
457	785	1659	501	398	441
798	1655	80	576	687	1581
612	986	439	221	828	81
370	1658	384	214	1883	654
175	1894	809	580	10	610
851	153	327	353	716	388
241	406	537	798	1699	986
396	606	90	881	58	457
1735	150	252	523	63	40
1896	1895	965	588	355	420
335	613	370	603	479	1721
971	72	1663	860	14	345
1658	753	1904	484	228	416

344	502	1655	880	809	637
200	246	478	809	415	362
450	797	400	396	882	331
707	1723	503	597	96	307
569	617	53	388	265	106
874	523	141	114	1681	717
627	788	569	1730	109	74
559	171	579	643	5	355
1729	1672	86	1883	1598	1722
9	232	183	302	458	626
1714	736	160	375	1773	216
1987	1525	63	478	694	383
546	307	79	865	677	60
375	1690	1711	1690	794	881
265	209	1728	356	335	393
511	568	1651	534	396	603
1901	88	770	537	352	1774
547	274	1724	1706	312	946
70	610	846	317	393	1911
79	376	363	417	653	866
40	755	202	1723	547	580
951	226	398	106	823	398
623	1640	405	274	114	392
317	559	289	312	498	1883
1641	664	453	17	1896	25
355	852	1641	431	47	852
279	424	893	299	200	1651
762	576	226	303	388	18
1667	1659	432	995	642	395
289	588	233	282	603	677
307	733	1652	531	491	268
81	1695	1667	363	1676	213
576	282	1680	981	743	85
269	1734	610	559	365	351
531	118	5	879	216	870
855	363	866	815	511	175
55	693	1674	1652	82	246
599	965	1700	160	606	1730
794	770	845	944	18	1586
1734	689	865	405	241	1598
501	984	1911	935	610	302
339	60	608	168	15	547
458	82	1773	1728	118	1896
381	784	513	1667	612	196
571	738	392	252	1904	365
995	981	951	984	1842	613
82	678	168	40	214	530
244	1699	505	571	532	590

846	267	613	14	106	140
824	840	55	1700	599	439
608	766	265	613	1652	531
1708	317	938	653	772	584
114	385	241	37	1876	1892
225	381	637	1721	297	193
879	629	394	441	476	748
553	1896	448	532	627	513
907	815	269	1655	484	1735
47	623	715	267	262	189
823	340	313	1507	693	233
361	416	458	1891	580	230
588	847	1706	420	85	402
1706	823	228	888	457	502
451	580	415	60	377	505
504	553	24	779	585	282
243	313	74	406	503	1702
331	1908	1895	50	1892	281
620	420	173	393	226	611
498	222	668	1640	1581	228
228	22	986	297	22	840
1507	880	345	385	797	356
1742	809	1719	451	294	944
820	241	823	753	738	301
160	405	307	171	965	770
870	612	678	988	589	14
1696	299	373	738	1987	1773
1774	373	946	866	439	794
899	983	331	265	986	458
168	1663	612	1740	569	1690
852	654	716	1621	1783	559
880	866	1783	1702	1658	788
944	59	762	1681	175	203
50	1916	216	794	1680	766
153	221	275	293	56	385
425	1955	47	373	1900	269
340	826	335	310	263	984
400	1671	694	439	576	632
293	1722	518	733	1671	1667
597	265	406	518	568	1658
395	450	755	1711	733	15
24	569	118	164	166	1729
173	611	546	74	1663	644
1621	3	164	874	160	935
617	1711	1708	24	501	344
59	824	1696	1735	296	653
637	53	166	183	499	70
694	193	388	158	777	1731

865	383	72	476	840	899
755	225	106	840	622	796
1598	228	376	344	304	241
51	109	717	585	425	93
537	339	1640	200	340	1685
589	873	575	289	361	879
233	263	1581	757	473	820
301	503	785	304	798	753
756	762	870	907	1771	180
889	1669	794	335	893	786
743	24	339	1774	148	983
1904	370	491	1876	1729	34
845	453	1701	499	140	450
858	252	268	489	141	1711
717	512	14	1916	202	597
48	308	302	694	542	353
203	1724	1771	623	505	546
1773	51	1731	717	281	865
499	743	1690	384	1735	1884
584	79	766	230	820	823
88	1706	638	599	275	1701
882	214	589	268	1895	1891
777	160	1730	867	707	1659
1721	614	200	542	1700	815
299	355	568	861	715	141
345	392	693	677	1509	589
273	718	312	225	847	1740
502	489	273	1911	848	758
476	268	109	855	317	491
994	1652	858	882	888	394
491	417	848	277	268	1728
1681	243	984	432	824	1652
1740	441	531	1651	844	473
441	622	881	141	907	361
1894	148	642	1714	796	17
193	1651	193	766	478	762
437	1901	377	627	150	405
965	296	995	118	1640	614
189	432	542	1903	866	856
575	620	437	784	1903	532
928	1641	177	629	339	1706
282	870	851	715	1685	678
888	479	48	568	504	1719
274	342	882	166	855	498
1659	394	299	79	530	847
1722	86	450	1896	489	166
313	756	653	184	611	824
58	262	317	362	289	1655

147	351	779	313	9	431
766	867	140	856	873	1908
5	513	158	450	788	733
1728	90	617	957	626	232
513	717	824	400	766	82
80	677	1658	1709	556	400
590	632	855	1908	383	569
226	1729	3	279	302	882
417	163	504	1598	1719	534
534	856	70	308	1674	221
202	777	1722	1685	1723	432
772	844	1699	1684	531	312
327	1876	1672	828	575	575
376	798	530	744	880	861
431	25	888	473	448	545
15	917	765	59	597	873
362	375	279	1724	867	381
610	10	1721	899	654	784
1695	1892	358	232	189	51
180	70	644	398	1901	3
1672	344	815	9	196	499
1705	482	1723	63	870	1676
310	285	402	498	744	399
1680	491	246	397	269	553
693	48	476	55	957	1723
796	1728	203	664	180	855
1640	589	553	331	1586	809
873	476	1987	823	375	1895
518	1884	935	80	994	1509
642	531	285	226	351	785
353	1783	1729	1884	80	1771
1859	457	98	638	917	798
98	327	51	986	637	86
90	399	262	34	1859	642
1652	545	798	569	232	772
384	962	797	1892	431	1894
37	397	784	1987	944	482
148	353	196	1901	1702	1859
935	858	498	846	81	160
308	1774	971	962	482	437
17	534	1900	788	308	718
285	384	718	546	293	629
196	275	856	1671	1891	1904
365	889	1685	383	163	917
1655	1842	590	575	244	200
809	140	1735	147	559	373
294	415	620	1680	331	1681
622	820	321	762	86	114

580	597	221	216	1730	689
957	644	867	748	1708	267
383	584	297	770	826	1507
1926	511	1709	917	164	90
63	1735	431	72	571	743
163	687	899	844	786	501
765	547	56	893	534	50
275	473	512	505	313	608
1731	58	40	530	356	59
738	302	277	10	1711	1734
53	362	1955	736	613	867
158	80	1676	273	1721	150
85	18	1705	51	1884	214
236	74	383	48	1705	571
197	988	847	1641	184	1640
3	556	397	965	846	1708
221	851	1894	796	881	693
1730	9	395	148	865	1674
687	1674	22	119	70	694
171	1705	342	197	620	995
736	1581	707	1658	718	236
545	1883	244	296	770	177
489	40	744	654	1728	1901
1711	1859	1742	213	579	568
262	758	232	1900	1911	327
18	938	180	377	1709	612
184	1621	559	425	299	1525
530	431	304	177	851	384
753	653	267	889	48	80
1700	1509	18	946	233	537
405	439	786	1859	784	668
60	303	451	1659	879	243
141	98	153	622	546	988
109	200	1621	689	664	37
106	1684	629	1742	74	296
1876	173	362	617	1724	1705
785	164	580	193	736	1916
579	202	737	1669	1740	252
34	1680	988	1783	962	209
1723	297	296	402	25	1724
1883	779	93	56	845	10
840	935	10	626	307	1926
815	437	571	381	98	118
986	277	25	392	88	22
614	81	148	668	222	184
1581	1911	1695	7	153	308
93	995	119	361	384	518
1586	1903	1586	687	689	1783

689	281	353	502	345	489
453	458	1883	394	935	599

2. LIJST GROOTTE GEMEENTEN

Tabel 2: In deze lijst staan de CBS-codes van gemeenten gesorteerd op het aantal stemmen van groot naar klein. In de eerste kolom staan plek 1 tot 150, in kolom 2 staan plek 151 tot 300 en in kolom 3 staan plek 301 tot 416.

Gemeente	aantal stemmen	Gemeente	aantal stemmen	Gemeente	aantal stemmen
363	385008	512	18751	473	10003
599	280509	18	18602	951	9813
518	267584	556	18264	946	9770
344	176999	762	18114	180	9762
14	115287	37	18104	1651	9742
772	114477	327	18026	743	9689
855	105393	585	17915	907	9647
268	96476	858	17872	1729	9447
758	96299	756	17831	576	9434
34	94134	385	17780	1658	9415
200	90224	1586	17715	395	9397
392	86219	86	17596	880	9336
307	82969	542	17491	184	9296
153	82150	358	17446	501	9277
796	80798	757	16851	689	9237
394	80108	294	16781	48	9232
202	79092	302	16689	848	9228
479	78480	1721	16637	312	9213
193	69686	148	16596	40	9131
637	68697	899	16517	798	9121
546	68687	753	16480	72	9101
935	64608	845	16429	499	9023
228	64251	397	16391	1685	9002
505	62371	451	16374	874	8898
114	60035	51	16365	614	8872
1783	59756	1652	16332	643	8864
80	54692	547	16144	293	8803
503	54578	236	16113	53	8780
150	54491	197	16101	962	8456
1883	54254	233	16095	707	8416
361	53844	1655	16078	331	8365
983	53272	416	16076	1908	8249
402	48891	569	16060	335	8152
1900	48734	1680	15973	986	8043
917	47515	1774	15967	437	8016
828	46573	59	15867	82	7986
164	46247	216	15860	611	7820
362	45237	1884	15839	889	7725
794	44657	1681	15754	788	7696
439	43211	47	15731	450	7690
1916	42614	531	15717	738	7648

484	41880	214	15605	678	7630
513	41115	870	15562	568	7608
1674	41072	971	15498	893	7511
405	39024	297	15407	244	7440
453	38658	627	15342	511	7431
995	38497	85	15337	1702	7371
612	38453	575	15302	786	7369
622	38391	824	15274	79	7302
106	38173	98	15209	476	7247
1904	37736	1719	15198	351	7228
606	37451	717	15155	1667	7196
141	37167	1903	15073	608	7172
345	36204	1690	14877	1987	7110
502	35445	226	14865	304	7090
537	35361	209	14831	417	6985
355	35174	308	14782	765	6964
748	35035	865	14748	15	6854
356	34894	716	14720	3	6780
160	34762	1696	14683	588	6752
90	32419	189	14658	733	6697
222	32175	158	14636	432	6653
203	31840	273	14601	140	6634
118	31406	677	14546	815	6634
1621	31105	285	14486	221	6590
400	30817	10	14465	653	6486
1581	30397	784	14422	5	6431
826	30132	629	14412	81	6425
632	29890	399	14322	415	6416
957	29629	626	14263	965	6396
166	29389	613	14202	1671	6388
715	28991	263	14193	1663	6379
321	28673	766	14163	24	6357
687	28025	58	14152	580	6267
398	27852	1641	14138	623	6220
1911	27832	584	14129	491	6114
988	27771	820	14016	196	6112
342	27401	352	13995	589	6043
1926	27302	377	13922	63	6018
603	27263	1684	13906	25	5979
1859	27196	610	13861	693	5976
301	27074	579	13849	694	5948
310	26870	553	13689	1723	5928
1705	26814	785	13537	282	5872
489	26654	1714	13532	376	5833
597	26636	230	13379	279	5649
275	26525	654	13351	571	5614
736	26345	384	13332	431	5558
1708	26224	269	13218	7	5519

1734	26196	1598	13193	755	5507
867	25916	809	13190	370	5425
171	25876	851	13062	617	5421
243	25514	168	13017	317	5354
642	25415	1740	12988	504	5092
406	25352	1896	12765	644	5061
74	25278	844	12763	944	5026
861	25265	1701	12716	1722	5022
1894	24916	147	12683	638	4954
420	24859	534	12456	881	4827
797	24748	1659	12453	1695	4683
718	24292	1669	12435	9	4488
928	24213	703	12431	981	4436
1892	24183	213	12387	93	4417
296	24052	183	12385	365	4111
1507	24040	532	12240	424	4061
984	23562	17	12177	478	3928
267	23560	840	12021	744	3762
856	23217	938	11991	265	3457
1876	23184	70	11850	393	3351
1509	22975	313	11848	458	3300
1742	22944	50	11816	60	2933
777	22743	779	11808	339	2886
303	22591	1706	11712	277	1060
1895	22355	873	11666	96	993
383	22252	55	11662	88	912
1676	22208	620	11571		
396	22199	1672	11525		
177	22089	545	11492		
1735	21916	498	11482		
1771	21912	56	11452		
163	21872	340	11449		
530	21841	22	11371		
664	21713	241	11334		
281	21544	1842	11273		
882	21462	246	11259		
1525	21462	668	11216		
109	21394	1891	11191		
1640	21349	441	11183		
262	21313	879	11167		
860	20989	852	11026		
375	20948	175	11025		
1709	20612	1728	10952		
1731	20562	482	10919		
289	20516	994	10862		
1730	20417	770	10843		
1955	20204	559	10790		
373	20064	523	10709		

1901	19925	1724	10640
232	19863	847	10580
274	19750	846	10571
353	19460	823	10564
1711	19408	388	10555
119	19317	425	10475
1699	19202	1773	10472
737	19108	457	10424
1700	19087	225	10362
381	19084	888	10280
299	19056	252	10245
173	18866	448	10137
590	18818	866	10011

3. LIJST MET MODELGEMEENTEN

Tabel 3: In deze lijst staan de CBS-codes van gemeenten gesorteerd op welke gemeentelijke uitslagen het dichtst bij de echte uitslag ligt. In de eerste kolom staan plek 1 tot 155, in kolom 2 staan plek 156 tot 310 en in kolom 3 staan plek 311 tot 416.

Gemeente	het kwadraat van de norm	Gemeente	het kwadraat van de norm	Gemeente	het kwadraat van de norm
200	0,001	935	0,011	228	0,0274
388	0,0012	733	0,0111	907	0,0275
396	0,0015	1904	0,0113	345	0,0276
216	0,0016	51	0,0114	1525	0,0278
512	0,0017	1690	0,0116	1651	0,0278
356	0,0021	40	0,0116	1598	0,028
1734	0,0021	786	0,0117	478	0,0282
226	0,0024	1676	0,0117	870	0,0285
299	0,0024	118	0,0117	37	0,0286
880	0,0026	168	0,0118	1723	0,0286
453	0,0028	632	0,0118	90	0,0288
405	0,0029	873	0,0119	874	0,0289
556	0,003	221	0,0119	233	0,0291
232	0,0031	373	0,0119	881	0,0291
766	0,0031	420	0,012	340	0,0295
307	0,0032	984	0,012	1884	0,0295
1705	0,0033	431	0,012	15	0,0296
402	0,0034	599	0,0121	53	0,03
784	0,0034	476	0,0121	363	0,0301
375	0,0035	406	0,0122	79	0,0304
637	0,0035	147	0,0122	5	0,0306
505	0,0035	951	0,0122	56	0,0306
748	0,0035	687	0,0123	1901	0,0308
796	0,0035	846	0,0124	175	0,0308
620	0,0035	1659	0,0124	50	0,0312
119	0,0037	1706	0,0125	482	0,0313
785	0,004	856	0,0125	7	0,0316
274	0,004	866	0,0127	473	0,0319
281	0,0041	399	0,0128	815	0,0326
296	0,0041	1735	0,0128	14	0,0328
209	0,0042	1911	0,0129	576	0,0332
398	0,0042	150	0,0129	81	0,0337
457	0,0042	798	0,013	1621	0,034
664	0,0043	362	0,0131	9	0,0342
826	0,0043	308	0,0131	575	0,0344
758	0,0043	530	0,0132	331	0,0345
82	0,0044	1681	0,0133	882	0,0348
502	0,0044	171	0,0135	59	0,0348
513	0,0044	1641	0,0136	1722	0,0351
677	0,0045	214	0,0137	537	0,0356

275	0,0045	202	0,0137	644	0,0356
1674	0,0045	415	0,0138	569	0,0359
995	0,0045	88	0,0138	166	0,036
225	0,0045	501	0,0138	559	0,036
777	0,0046	828	0,0139	610	0,0361
252	0,0046	1581	0,0141	1842	0,0366
213	0,0046	568	0,0141	370	0,0367
603	0,0047	788	0,0142	917	0,0367
1903	0,0047	193	0,0143	397	0,0374
93	0,0048	1900	0,0143	1696	0,0375
642	0,0048	243	0,0144	589	0,0376
1708	0,0049	824	0,0145	18	0,0381
173	0,0049	85	0,0146	840	0,0382
1714	0,005	437	0,0147	189	0,0388
855	0,005	1586	0,0147	80	0,0389
622	0,005	1640	0,0148	1700	0,039
715	0,005	983	0,015	588	0,0392
1955	0,005	1728	0,0152	58	0,0393
757	0,005	546	0,0153	3	0,0397
518	0,0051	762	0,0153	899	0,0402
532	0,0051	424	0,0154	614	0,0402
590	0,0053	327	0,0155	1987	0,0413
1695	0,0053	627	0,0155	47	0,0418
938	0,0053	48	0,0159	1895	0,042
1859	0,0054	946	0,0159	1891	0,0445
547	0,0054	1667	0,0161	1774	0,0453
643	0,0054	197	0,0163	736	0,0466
718	0,0055	889	0,0164	707	0,0471
861	0,0055	823	0,0164	928	0,0481
441	0,0055	70	0,0164	1702	0,049
867	0,0055	22	0,0165	694	0,0495
484	0,0055	770	0,0165	358	0,0497
1771	0,0057	612	0,0169	425	0,0507
222	0,0059	743	0,0169	1783	0,052
321	0,006	1892	0,017	273	0,0532
448	0,006	534	0,0172	351	0,0554
753	0,006	301	0,0174	376	0,058
772	0,0061	1685	0,0175	765	0,0582
164	0,0061	1652	0,0175	377	0,0582
141	0,0061	981	0,0175	504	0,0596
355	0,0061	1721	0,0176	385	0,0603
1709	0,0061	1507	0,0177	277	0,0603
608	0,0061	1883	0,0177	756	0,0629
400	0,0061	1663	0,0178	230	0,0642
1509	0,0062	1672	0,0178	716	0,0645
944	0,0062	263	0,018	183	0,0685
439	0,0062	1658	0,0182	279	0,0692
779	0,0063	613	0,0183	511	0,0751

384	0,0063	303	0,0183	269	0,0755
293	0,0064	879	0,0187	523	0,0776
845	0,0064	395	0,0188	1896	0,0815
361	0,0065	1894	0,0189	203	0,0821
282	0,0065	177	0,0189	738	0,0839
1876	0,0066	432	0,0191	339	0,0842
109	0,0067	265	0,0192	1742	0,089
1701	0,0067	1680	0,0193	302	0,0915
479	0,0069	626	0,0193	629	0,0916
986	0,007	848	0,0197	1740	0,0917
1773	0,0071	236	0,0197	693	0,0969
858	0,0071	1684	0,0197	417	0,0993
285	0,0072	499	0,0197	571	0,1118
294	0,0074	689	0,0198	313	0,1159
34	0,0074	542	0,0199	703	0,1236
353	0,0075	653	0,0199	180	0,1588
852	0,0075	342	0,02	184	0,3793
617	0,0077	847	0,0203		
25	0,0077	365	0,0203		
1719	0,0079	393	0,0203		
994	0,0079	344	0,0203		
809	0,008	744	0,0204		
844	0,0081	458	0,0205		
585	0,0082	86	0,0207		
241	0,0083	1724	0,0207		
352	0,0083	106	0,0211		
820	0,0083	678	0,0212		
797	0,0084	498	0,0215		
60	0,0084	304	0,0216		
98	0,0084	1671	0,0217		
1655	0,0085	579	0,022		
1731	0,0085	394	0,022		
865	0,0085	1926	0,0221		
196	0,0086	63	0,0222		
638	0,0087	317	0,0226		
962	0,0088	717	0,0229		
96	0,0088	1669	0,023		
988	0,0088	74	0,0232		
957	0,009	553	0,0233		
416	0,0091	148	0,0234		
545	0,0093	246	0,0235		
17	0,0093	971	0,0235		
851	0,0094	297	0,0236		
392	0,0094	140	0,0236		
153	0,0096	335	0,024		
262	0,0096	531	0,0241		
1730	0,0096	55	0,0246		
611	0,0097	755	0,0246		

1699	0,0097	965	0,0247
383	0,0097	491	0,0248
503	0,0098	244	0,025
794	0,0101	160	0,0251
888	0,0101	1908	0,0251
312	0,0102	267	0,0251
158	0,0103	451	0,0256
1729	0,0104	737	0,0257
310	0,0105	654	0,0259
584	0,0105	268	0,026
1916	0,0107	10	0,026
597	0,0108	381	0,0263
450	0,0109	623	0,0265
489	0,0109	163	0,0266
860	0,0109	24	0,0267
580	0,0109	114	0,0268
668	0,0109	893	0,027
606	0,011	289	0,0272
72	0,011	1711	0,0272

4. MATLAB CODE

4.1. Start

```
1 %Start is een file dat alle benodigde functies aanroept in de juiste
2 %volgorde om het gewenste resultaat te behalen (namelijk een voorspelling
3 %geven).
4
5 %Verwijder alle variabelen
6 clear all
7 %Sluit alle vensters
8 close all
9
10 %Input voor Programma
11 [grafieken, jaartal, binnenkomst, wegen]=InputProgramma();
12
13 %Laad data verkiezingen
14 [DataNieuw, DataOud, Echte_Oude_Uitslag, Echte_Nieuwe_Uitslag, Volgorde,
15     Echte_Oude_Uitslag_Stemmen, Echte_Nieuwe_Uitslag_Stemmen]
16     =LadenData(jaartal, binnenkomst);
17
18 %werk gemeentelijke veranderingen bij
19 if jaartal==2010
20     [DataOud, DataNieuw]=DataCorrector20062010(DataOud, DataNieuw);
21 end
22 if jaartal==2012
23     [DataOud, DataNieuw]=DataCorrector20102012(DataOud, DataNieuw);
24 end
25
26 %Doe taken aan de hand van de gekozen input
27 [Volgorde, WegingFactor]=ConsequentieInput(binnenkomst, wegen, DataOud,
28     DataNieuw, Volgorde);
29
30 %Sla de input data op vanwege de noodzaak om later nog met het aantal
31 %stemmen door te rekenen voor Methode C
32 DataNieuwStemmen=DataNieuw;
33 DataOudStemmen=DataOud;
34
35 %Percentages Maken
36 [DataOud]=PercentagesMaken(DataOud);
37 [DataNieuw]=PercentagesMaken(DataNieuw);
38
39 %construeer T
40 [U, V, T, Topslag]=DataSort(DataOud, DataNieuw, Volgorde, WegingFactor, wegen);
41
42 %bekijk convergentie T
43 [KwadratischeFout]=ConvergentieT(Topslag, grafieken);
44
45 %export uitslag naar excel spreadsheet
46 Uitslag=T*Echte_Oude_Uitslag;
47 OutputExcelSheet(Uitslag, jaartal);
48
49 %Bekijk hoe de voorspelling zich gedraagt in de tijd
50 [Verloop]=VerloopVoorspelling(Topslag, jaartal, Echte_Oude_Uitslag, grafieken);
51
52 %Methode C
53 [LambdaOpslag, NormOpslagMethC] = MethodeC(DataOudStemmen, DataNieuwStemmen,
54     Volgorde, Echte_Oude_Uitslag_Stemmen, Echte_Nieuwe_Uitslag_Stemmen);
```

```

55
56 %Bekijk hoe de voorspelling van methode B zich gedraagt tov de methode A
57 [NormOpslagMethAenB]=VergelijkMethodes(TOpslag,Echte_Oude_Uitslag,
58 Echte_Nieuwe_Uitslag,DataNieuwStemmen,Volgorde,grafieken,NormOpslagMethC);
59
60 %Bekijk wanneer de voospelde uitslag in de norm epsilon dicht bij de
61 %einduitslag ligt
62 [ConvergentiePuntA]=ConvergentieMoment(NormOpslagMethAenB(:,1));
63 [ConvergentiePuntB]=ConvergentieMoment(NormOpslagMethAenB(:,2));
64 if jaartal==2012
65     [ConvergentiePuntC]=ConvergentieMoment(NormOpslagMethC(13:415,1));
66 else
67     [ConvergentiePuntC]=ConvergentieMoment(NormOpslagMethC(11:430,1));
68 end
69
70 %Model Gemeente zoeken
71 [FoutOpslag,LijstPotentieleModelGemeenten]=ModelGemeenten(V,
72     Echte_Nieuwe_Uitslag,Volgorde);

```

4.2. InputProgramma

```
1 function [grafieken, jaartal, binnenkomst, wegen]=InputProgramma()
2 % Functie: InputProgramma()
3 %
4 % Output: [grafieken, jaartal, binnenkomst, wegen]
5 % *grafieken is 0 of 1. 0 betekent geen grafieken maken, 1 betekent wel
6 %   grafieken maken.
7 % *jaartal is 2012 of 2010. jaartal representeert het verkiezingsjaar.
8 % *binnenkomst is 0,1,2,3 of 4. binnenkomst geeft aan welke volgorde er
9 %   wordt gebruikt bij het voorspellen.
10 % *wegen is 0 of 1. 0 betekent niet wegen, 1 wel wegen.
11 %
12 % Uitleg: De functie vraagt om input van de gebruiker. Deze input bepaalt
13 %   welk verkiezingsjaar en welke volgorde er wordt gebruikt.
14 %   Verder moet de gebruiker aangeven of deze een weegfactor wil
15 %   toekennen aan gemeenten.
16 %
17 % Input: geen.
18
19 grafieken=input('Grafieken? (0=nee, 1=ja):');
20 jaartal=input('Verkiezingsjaar (2010,2012):');
21
22 if jaartal==2010
23     binnenkomst=input('Volgorde: 0=alfabetisch, 2=random, 3=laag naar hoog,
24                       4=hoog naar laag:');
25 else
26     binnenkomst=input('Volgorde: 0=alfabetisch, 1=echte volgorde, 2=random,
27                       3=laag naar hoog, 4=hoog naar laag:');
28 end
29
30 wegen=input('Weging? (0=nee, 1=ja):');
```

4.3. LadenData

```
1 function [DataNieuw,DataOud,Echte_Oude_Uitslag,Echte_Nieuwe_Uitslag,
2         Volgorde,Echte_Oude_Uitslag_Stemmen,Echte_Nieuwe_Uitslag_Stemmen]
3         =LadenData(jaartal,binnenkomst)
4 % Functie: LadenData(jaartal,binnenkomst)
5 %
6 % Output:[DataNieuw,DataOud,Echte_Oude_Uitslag,Echte_Nieuwe_Uitslag,
7 %         Volgorde,Echte_Oude_Uitslag_Stemmen,Echte_Nieuwe_Uitslag_Stemmen]
8 % *DataNieuw/DataOud is een mxn matrix met gemeentelijke
9 %   verkiezingsuitslagen.
10 % *Echte_Oude/Nieuwe_Uitslag: is een pxl matrix met de landelijke
11 %   verkiezingsuitslag in percentages.
12 % *Echte_Oude/Nieuwe_Uitslag_Stemmen: is een pxl matrix met de landelijke
13 %   verkiezingsuitslag in stemmen.
14 % *Volgorde is een lxm matrix waarin CBS-codes staan gesorteerd op
15 %   binnenkomst. Volgorde is leeg als binnenkomst ongelijk is aan 1. In
16 %   dit geval wordt Volgorde buiten deze functie bepaald.
17 %
18 % Uitleg: Deze code importeert de benodigde data voor het voorspellen uit
19 %   geprepareerde Excelfiles.
20 %
21 % Input:
22 % *jaartal is een getal uit [2010,2012]
23 % *binnenkomst is een getal uit [0,1,2]
24
25 %Laden van de uitslagen van de data van 2010 en 2012
26 if jaartal==2010
27     A=importdata('Data_2006_2010.xlsx');
28     B=importdata('Data_2010_2010.xlsx');
29     C=importdata('Echte_Uitslag_2006_2010.xlsx');
30     D=importdata('Echte_Uitslag_2010_2010.xlsx');
31     C1=C.data;
32     D1=D.data;
33
34     DataOud=A.data;
35     DataNieuw=B.data;
36
37     %Deze zijn nodig voor de Alternatieve Methode
38     Echte_Oude_Uitslag_Stemmen=C1(1:11,1);
39     Echte_Nieuwe_Uitslag_Stemmen=D1(1:11,1);
40
41     %Deze zijn nodig voor de Kl.Kw.M.
42     %in kolom 2 staan de percentages van de uitslag
43     Echte_Oude_Uitslag=C1(:,2);
44     Echte_Nieuwe_Uitslag=D1(:,2);
45
46     [rows,columns]=size(Echte_Oude_Uitslag);
47     Echte_Oude_Uitslag(rows,:)=[];
48     Echte_Nieuwe_Uitslag(rows,:)=[];
49
50     if binnenkomst==0 || binnenkomst==2 || binnenkomst==3 || binnenkomst==4
51         % in dit geval wordt de volgorde buiten deze functie bepaald.
52         Volgorde=[];
53     end
54 end
55
56 if jaartal==2012
57     A=importdata('Data_2010_2012.xlsx');
```

```

58 B=importdata('Data_2012_2012.xlsx');
59 C=importdata('Echte_Uitslag_2010_2012.xlsx');
60 D=importdata('Echte_Uitslag_2012_2012.xlsx');
61 C1=C.data;
62 D1=D.data;
63
64 DataOud=A.data;
65 DataNieuw=B.data;
66
67 %Deze zijn nodig voor de Alternatieve Methode
68 Echte_Oude_Uitslag_Stemmen=C1(1:13,1);
69 Echte_Nieuwe_Uitslag_Stemmen=D1(1:13,1);
70
71 %Deze zijn nodig voor de Kl.Kw.M.
72 Echte_Oude_Uitslag=C1(:,2);
73 Echte_Nieuwe_Uitslag=D1(:,2);
74
75 [rows,columns]=size(Echte_Oude_Uitslag);
76 Echte_Oude_Uitslag(rows,:)=[];
77 Echte_Nieuwe_Uitslag(rows,:)=[];
78
79 if binnenkomst==1
80     E=importdata('volgorde_gemeentes_2012.xlsx');
81     Volgorde=E.data(:,3);
82     Volgorde=transpose(Volgorde);
83 end
84 if binnenkomst==0 || binnenkomst==2 || binnenkomst==3 || binnenkomst==4
85     % in dit geval wordt de volgorde buiten deze functie bepaald.
86     Volgorde=[];
87 end
88 end

```

4.4. DataCorrector20062010

```
1 function [DataOud,DataNieuw]=DataCorrector20062010(DataOud,DataNieuw)
2 %Functie: DataCorrector20062010(DataOud,DataNieuw)
3 %
4 %Output:=DataCorrector20062010(DataOud,DataNieuw)
5 % *DataOud is een qxn matrix met de oude, aangepaste, gemeentelijke
6 %   verkiezingsuitslagen.
7 % *DataNieuw is een qxn matrix met de nieuwe, aangepaste, gemeentelijke
8 %   verkiezingsuitslagen.
9 %
10 %Uitleg: De functie roept DataCompressor aan voor gemeentes die
11 %   samengevoegd moeten worden. Wanneer nodig verwijdert de functie
12 %   data. De functie zorgt ervoor dat voor het voorspellen van de
13 %   uitslagen van 2010 DataOud en DataNieuw dezelfde CBS-codes bevatten.
14 %
15 %Input:
16 % *DataOud is een mxn matrix met de oude niet aangepaste, gemeentelijke
17 %   verkiezingsuitslagen.
18 % *DataNieuw is een pxn matrix met de nieuwe, niet aangepaste,
19 %   gemeentelijke verkiezingsuitslagen.
20
21
22
23
24 %Verwijder 9999 uit DataOud
25 [row,column]=find(DataOud(:,1)==9999);
26 DataOud(row,:)=[];
27
28 %Neem 240 als 1586
29 B=[240];
30 C=1586;
31 DataOud=DataCompressor(DataOud,B,C);
32
33 %Neem 429, 558 samen als 1598
34 B=[429,558];
35 C=1598;
36 DataOud=DataCompressor(DataOud,B,C);
37
38 %Neem 492,493,495 samen als 1621
39 B=[492,493,495];
40 C=1621;
41 DataOud=DataCompressor(DataOud,B,C);
42
43 %Neem 914,920,925,1670 samen als 1640
44 B=[914,920,925,1670];
45 C=1640;
46 DataOud=DataCompressor(DataOud,B,C);
47
48 %Neem 977,1937,933 samen als 1641
49 B=[977,1937,933];
50 C=1641;
51 DataOud=DataCompressor(DataOud,B,C);
52
53 %Neem 483,645 samen als 1884
54 B=[483,645];
55 C=1884;
56 DataOud=DataCompressor(DataOud,B,C);
57
```



```

58 %Neem 65 samen als 1891
59 B=[65];
60 C=1891;
61 DataOud=DataCompressor(DataOud,B,C);
62
63 %Neem 563,567,1666 samen als 1892
64 B=[563,567,1666];
65 C=1892;
66 DataOud=DataCompressor(DataOud,B,C);
67
68 %Neem 918,929,934,941 samen als 1894
69 B=[918,929,934,941];
70 C=1894;
71 DataOud=DataCompressor(DataOud,B,C);
72
73 %Neem 39,52, 1661 samen als 1895
74 B=[39,52,1661];
75 C=1895;
76 DataOud=DataCompressor(DataOud,B,C);
77
78 %Tel 517 op bij 585
79 B=[517,585];
80 C=585;
81 DataOud=DataCompressor(DataOud,B,C);
82
83 %Tel 372 op bij 377
84 B=[372,377];
85 C=377;
86 DataOud=DataCompressor(DataOud,B,C);
87
88 %Tel 466,529 op bij 420
89 B=[466,529,420];
90 C=420;
91 DataOud=DataCompressor(DataOud,B,C);
92
93 %Tel 480,1673 op bij 569
94 B=[480,1673,569];
95 C=569;
96 DataOud=DataCompressor(DataOud,B,C);
97
98 %Tel 600 op bij 599
99 B=[600,599];
100 C=599;
101 DataOud=DataCompressor(DataOud,B,C);
102
103 %Tel 885 op bij 983
104 B=[885,983];
105 C=983;
106 DataOud=DataCompressor(DataOud,B,C);
107
108 %Tel 964 op bij 1894
109 B=[964,1894];
110 C=1894;
111 DataOud=DataCompressor(DataOud,B,C);
112
113 %Tel 975 op 957
114 B=[975,957];
115 C=957;
116 DataOud=DataCompressor(DataOud,B,C);
117
118 %Tel 993 op bij 984

```

```
119 B=[993,984];
120 C=984;
121 DataOud=DataCompressor(DataOud,B,C);
122
123 %Tel 1679 op bij 1669
124 B=[1679,1669];
125 C=1669;
126 DataOud=DataCompressor(DataOud,B,C);
```

4.5. DataCorrector20102012

```
1 function [DataOud,DataNieuw]=DataCorrector20102012(DataOud,DataNieuw)
2 %Functie: DataCorrector20102012(DataOud,DataNieuw)
3 %
4 %Output:=DataCorrector20102012(DataOud,DataNieuw)
5 % *DataOud is een qxn matrix met de oude, aangepaste, gemeentelijke
6 %   verkiezingsuitslagen.
7 % *DataNieuw is een qxn matrix met de nieuwe, aangepaste, gemeentelijke
8 %   verkiezingsuitslagen.
9 %
10 %Uitleg: De functie roept DataCompressor aan voor gemeentes die
11 %   samengevoegd moeten worden. Wanneer nodig verwijdert de functie
12 %   data. De functie zorgt ervoor dat voor het voorspellen van de
13 %   uitslagen van 2012 DataOud en DataNieuw dezelfde CBS-codes bevatten.
14 %
15 %Input:
16 % *DataOud is een mxn matrix met de oude niet aangepaste, gemeentelijke
17 %   verkiezingsuitslagen.
18 % *DataNieuw is een pxn matrix met de nieuwe, niet aangepaste,
19 %   gemeentelijke verkiezingsuitslagen.
20
21 %Verwijder Bonaire, Sint Eustatius en Saba (9001-9003) uit DataNieuw
22 [row,column]=find(DataNieuw(:,1)==9001);
23 DataNieuw(row,:)=[];
24
25 [row,column]=find(DataNieuw(:,1)==9002);
26 DataNieuw(row,:)=[];
27
28 [row,column]=find(DataNieuw(:,1)==9003);
29 DataNieuw(row,:)=[];
30
31 %Voeg Lith (808) toe aan Oss (828)
32 B=[808,828];
33 C=828;
34 DataOud=DataCompressor(DataOud,B,C);
35
36 %Neem 91, 683, 64, 710 en 104 samen als 1900
37 B=[91, 683, 64, 710, 104];
38 C=1900;
39 DataOud=DataCompressor(DataOud,B,C);
40
41 %Neem 497 en 595 samen als 1901
42 B=[497,595];
43 C=1901;
44 DataOud=DataCompressor(DataOud,B,C);
45
46 %Nweem 905 en 936 samen als 1903
47 B=[905,936];
48 C=1903;
49 DataOud=DataCompressor(DataOud,B,C);
50
51 %Neem 311, 329 en 333 samen als 1904
52 B=[311,329,333];
53 C=1904;
54 DataOud=DataCompressor(DataOud,B,C);
55
56 %Neem 83 als 1908
57 [row,column]=find(DataOud(:,1)==83);
```

```
58 DataOud(row,1)=1908;
59
60 %Neem 366, 412, 462, 463 samen als 1911
61 B=[366, 412, 462, 463];
62 C=1911;
63 DataOud=DataCompressor(DataOud,B,C);
64
65 %Tel 364 en 459 op bij 420
66 B=[364, 459, 420];
67 C=420;
68 DataOud=DataCompressor(DataOud,B,C);
69
70 %Tel 305 op bij 736
71 B=[305, 736];
72 C=736;
73 DataOud=DataCompressor(DataOud,B,C);
```

4.6. ConsequentieInput

```
1 function [Volgorde,WegingFactor]=ConsequentieInput (binnenkomst,wegen,  
2                                     DataOud,DataNieuw,Volgorde)  
3 % Functie: ConsequentieInput (binnenkomst,wegen,DataOud,DataNieuw,Volgorde)  
4 %  
5 % Output: [Volgorde,WegingFactor]  
6 % *Volgorde is een lxn matrix met CBS-codes gesorteerd op binnenkomst  
7 % *WegingFactor is een nx3 met in de eerste kolom  
8 %   CBS-codes, tweede kolom het aantal stemmen van de gemeente en kolom 3  
9 %   het daarbij horende weegfactor. WegingFactor is een lege matrix als er  
10 %   geen weging wordt meegenomen.  
11 %  
12 % Uitleg: Deze functie bewerkt een aantal variabelen aan de hand van of er  
13 %   wel of niet gewogen wordt en aan de hand van welke volgorde er  
14 %   gebruikt wordt.  
15 %  
16 % Input:  
17 % *binnenkomst is een getal uit [0,1,2,3,4]. Het getal geeft aan welke  
18 %   volgorde er gebruikt wordt om te voorspellen.  
19 % *wegen is een getal uit [0,1]. 0 is niet wegen, 1 wel wegen.  
20 % *DataOud is een mxn matrix met de oude verkiezingsuitslagen.  
21 % *DataNieuw is een mxn matrix met de nieuwe verkiezingsuitslagen.  
22 % *Volgorde is leeg als binnenkomst==0,2,3,4. Volgorde is een lxn matrix  
23 %   met gesorteerde CBS-codes op binnenkomst als binnenkomst==1.  
24  
25 %Roep functie aan als Volgorde nog een lege matrix is.  
26 if binnenkomst==0 || binnenkomst==2 || binnenkomst==3 || binnenkomst==4  
27     [Volgorde]=MaakVolgorde(DataNieuw,binnenkomst);  
28 end  
29  
30 %Wegingfactoren bepalen en functie aanroepen indien nodig.  
31 if wegen==1  
32     [WegingFactor]=WegingEerlijk(DataOud,Volgorde);  
33 else  
34     WegingFactor=[];  
35 end
```

4.7. PercentagesMaken

```
1 function [A]=PercentagesMaken(A)
2 % Functie: PercentagesMaken(A)
3 %
4 % Output: [A]
5 % *A is een nxm matrix die de gemeentelijke verkiezingsuitslagen bevat in
6 % percentages.
7 %
8 % Uitleg: Deze functie zet het aantal stemmen per partij per gemeente om in
9 % percentages per partij per gemeente.
10 %
11 % Input:
12 % *A is een nxm matrix die de gemeentelijke verkiezingsuitslagen bevat in
13 % percentages.
14
15 [rijen,kolommen]=size(A);
16
17 for i=1:rijen
18     for j=5:kolommen
19         %vanaf kolom 5 want daar staat het eerste aantal stemmen
20         A(i,j)=A(i,j)/A(i,2); %kolom 2 bevat het aantal stemmen
21     end
22 end
```

4.8. DataSort

```
1 function [U,V,T,TOpslag]=DataSort(C,D,Volgorde,WegingFactor,wegen)
2 % Functie: DataSort(C,D,Volgorde,WegingFactor,wegen)
3 %
4 % Output: [U,V,T,TOpslag]
5 % *U bevat de gemeentelijke uitslagen van de vorige verkiezing
6 %   gesorteerd op binnenkomst
7 % *V bevat de gemeentelijke uitslagen van de huidige verkiezing
8 %   gesorteerd op binnenkomst
9 % *T is een nxn matrix. De T die output is, is de transformatiematrix die
10 % wordt gevonden als alle gemeenten binnengekomen zijn.
11 % *TOpslag is een pxq matrix waarin alle gevonden T zijn opgeslagen.
12 %
13 % Uitleg: Deze functie zet de nieuwe en oude data in de goede volgorde.
14 %   Vervolgens constueert deze functie een transformatiematrix. De
15 %   geconstueerde T's worden opgeslagen in de matrix TOpslag.
16 %
17 % Input:
18 % *C is een mxn matrix met de oude verkiezingsdata
19 % *D is een mxn matrix met de nieuwe verkiezingsdata
20 % *Volgorde is een lijst met CBS-codes gesorteerd op binnenkomst
21 % *WegingFactor is een lijst met de weegfactor per gemeente gesorteerd op
22 %   binnenkomst.
23 % *Wegen is een getal (0 of 1), 0 betekent niet wegen, 1 wel wegen.
24
25 U=[]; %oude data
26 V=[]; %nieuwe data
27 j=1; %j houdt bij in welke iteratie je zit
28 counter=0;
29 counter2=0;
30
31 [rowsC,columnsC]=size(C);
32 A=zeros((columnsC-4),(columnsC-4));
33 B=zeros((columnsC-4),(columnsC-4));
34
35 list=(columnsC-4):1:1000;
36
37 %Zorg dat de gemeente op plek i uit V dezelfde gemeente is als de gemeente
38 %op plek i uit U.
39 for i=Volgorde %i is dus een CBS-code
40   counter=counter+1;
41   [row_old,column_old]=find(C(:,1)==i); %zoek de juiste gemeente in C
42   [row_new,column_new]=find(D(:,1)==i); %zoek dezelfde gemeente als bij C
43
44   if wegen==1 %als wegen is 1 dan kennen we een weegfactor toe
45     weging=WegingFactor(j,3);
46   else
47     weging=1;
48   end
49
50   %sommeren vanaf 5 want in kolom 5 staat het eerste percentage
51   for k=5:columnsC
52     U((k-4),j)=C(row_old,k);
53     V((k-4),j)=D(row_new,k);
54   end
55
56   %bereken de getransformeerde van U
57   U_T=transpose(U);
```

```

58
59     if j==1 %stap 1 van het update algoritme
60         %ken hier gelijk de weging toe
61         A=weging*U*U_T;
62         B=weging*V*U_T;
63         [rowA,columnA]=size(A);
64     else
65         for a=1:rowA
66             for b=1:columnA
67                 %ken hier de weging toe
68                 A(a,b)=A(a,b)+weging*U(a,j)*U_T(j,b);
69                 B(a,b)=B(a,b)+weging*V(a,j)*U_T(j,b);
70             end
71         end
72     end
73
74     %Bereken hier T en sla deze op in Save
75     %value neemt de wordt een niet lege matrix als de waarde van counter in
76     %de lijst staat. Als dat zo is, is de numel 1 en kan je voorspellen. Er
77     %kan pas voorspeld worden als er evenveel gemeentes binnen zijn gekomen
78     %als er partijen zijn.
79     value=find(list==counter);
80     if numel(value)==1
81         T=B/A;
82         %Sla de T op. Elke rij van T wordt een rij in Save. Rij 1 van T
83         %nummer 1 staat dus in rij 1 van Save. Rij 2 van T nummer 1 staat
84         %dus op rij nummer 2 van Save, enz. Als T nxn is, dan staat
85         %rij 1 van T nummer 2 op rij nummer n+1 van Save. Dan staat rij 2
86         %van T nummer 2 op rij nummer n+2 van Save, enz.
87         for a=1:(columnsC-4)
88             Topslag((counter2*(columnsC-4))+a,:)=T(a,:);
89         end
90         counter2=counter2+1;
91     end
92
93     j=j+1;
94 end
95
96 %Bereken de uiteindelijke T die gevonden wordt als alle gemeenten zijn
97 %binnengekomen.
98 T=B/A;

```


4.9. ConvergentieT

```
1 function [KwadratischeFout] = ConvergentieT(TOpslag,grafieken)
2 % Functie: ConvergentieT(SAVE,grafieken)
3 %
4 % Output: [KwadratischeFout]
5 % *KwadratischeFout is een mxn matrix. Per rij staan de waarden van de
6 % norm in het kwadraat van de elementen van T tov van de uiteindelijk
7 % gevonden waarden van de laatste T.
8 %
9 % Uitleg: Deze functie bekijkt hoe de elementen van T zich gedragen in de
10 % loop van de tijd.
11 % Input:
12 % *TOpslag is een mxn-matrix waar alle T's in zijn opgeslagen
13 % *grafieken is een getal uit [0,1]. 0 betekent geen grafieken maken, 1
14 % betekent wel grafieken maken.
15
16 [rowsTOpslag,columnsTOpslag]=size(TOpslag);
17 aantalT=rowsTOpslag/columnsTOpslag;
18
19 for i=1:aantalT
20     for a=1:columnsTOpslag %haal uit TOpslag T terug
21         T(a,:)=TOpslag(a+((i-1)*columnsTOpslag),:);
22     end
23
24     b=0;
25     for c=1:columnsTOpslag
26         %Sla de elementen van T opnieuw op, waarbij elke kolom een element
27         %van T representeert (als T nxn dan in n^2 kolommen)
28         for d=1:columnsTOpslag
29             b=b+1;
30             GrafiekGegevens(i,b)=T(c,d);
31         end
32     end
33 end
34
35 [rowsG,columnsG]=size(GrafiekGegevens);
36
37 %bekijk de kwadratische fout van elke element van T tov de laatste T
38 for a=1:(rowsG-1)
39     for b=1:columnsG
40         KwadratischeFout(a,b)=(GrafiekGegevens(a,b)-GrafiekGegevens(rowsG,b))^2;
41     end
42 end
43
44 if grafieken==1
45     figure
46     plot(1:(rowsG-1),KwadratischeFout(:,1))
47     title('Kwadratische fout van T in elke iteratie tov laatste T')
48 end
```

4.10. OutputExcelSheet

```
1 function []=OutputExcelSheet(Uitslag,jaartal)
2 % Functie: OutputExcelSheet(Uitslag,jaartal)
3 %
4 % Output: []
5 %
6 % Uitleg: Deze functie zet de voorspelde uitslag in een Excelfile.
7 %
8 % Input:
9 % *Uitslag is een mx1-matrix die de voorspelde uitslagen per partij
10 %   bevat.
11 % *jaartal is een getal (2012 of 2010). jaartal bepaalt welke verkiezing
12 %   wordt voorspeld.
13
14 if jaartal==2010
15     filename = 'Voorspelde_Uitslag_2010.xlsx';
16     xlswrite(filename,Uitslag,1,'D2'); %filename, Data, Sheet, Range
17 end
18 if jaartal==2012
19     filename = 'Voorspelde_Uitslag_2012.xlsx';
20     xlswrite(filename,Uitslag,1,'D2'); %filename, Data, Sheet, Range
21 end
```

4.11. MethodeC

```
1 function [LambdaOpslag,MethCNormOpslag] = MethodeC(DataOudStemmen,
2           DataNieuwStemmen,Volgorde,Echte_Oude_Uitslag_Stemmen,
3           Echte_Nieuwe_Uitslag_Stemmen)
4 % Functie: MethodeC(DataOudStemmen,DataNieuwStemmen,Volgorde,
5 %           Echte_Oude_Uitslag_Stemmen,Echte_Nieuwe_Uitslag_Stemmen)
6 %
7 % Output: [LambdaOpslag,MethCNormOpslag]
8 % *LambdaOpslag is een mxn matrix waar de gevonden diagonaal elementen
9 %   voor de transformatiematrix in op worden geslagen.
10 % *MethCNormOpslag is een mx1 matrix waar de norm in het kwadraat van
11 %   de verschilvector van de voorspelling tov van de einduitslag in
12 %   opgeslagen zijn.
13 %
14 % Uitleg: Deze functie doet een voorspelling van de einduitslag op basis
15 %   van de zogenaamde "Alternatieve Methode".
16 %
17 % Input:
18 % *DataOudStemmen is een mxn matrix met de gemeentelijke
19 %   verkiezingsuitslagen van de vorige verkiezing in stemmen
20 % *DataNieuwStemmen is een mxn matrix met de gemeentelijke
21 %   verkiezingsuitslagen van de huidige verkiezing in stemmen
22 % *Volgorde is een lxp matrix waarin de CBS codes van de gemeentes zijn
23 %   gesorteerd op binnenkomst
24 % *Echte_Oude_Uitslag_Stemmen is een mx1 matrix met de oude landelijke
25 %   uitslag in stemmen
26 % *Echte_Nieuwe_Uitslag is een mx1 matrix met de nieuwe landelijke
27 %   uitslag in stemmen
28
29 %DataSort zorgt ervoor dat de data in de juiste volgorde komt te staan in
30 %de matrix U en V. Deze matrices bevatten dan ook alleen maar de uitslagen
31 %van de gemeenten, dus geen CBS codes of opkomstpercentages meer.
32 [U,V,T,Topslag]=DataSort(DataOudStemmen,DataNieuwStemmen,Volgorde,[],0);
33
34 %Bereken de grootte van U
35 [rowsU,columnsU]=size(U);
36
37 %Bereken het totaal aantal uitgebrachte stemmen van de oude verkiezing
38 totaalstemmen=sum(Echte_Nieuwe_Uitslag_Stemmen);
39
40 %maak van de van de huidige uitslag in stemmen een huidige uitslag in
41 %percentages.
42 for j=1:rowsU
43     Echte_Nieuwe_Uitslag_Perc(j,1)=Echte_Nieuwe_Uitslag_Stemmen(j,1)/totaalstemmen;
44 end
45
46 %Vind de juiste lambda's
47 for a=1:columnsU
48
49     %Ulopend en Vlopend bevatten de verkiezingsdata die tot nu toe zijn
50     %binnengekomen. 'a' duidt het aantal gemeentes aan dat tot nu toe zijn
51     %binnengekomen.
52     Ulopend=U(:,1:a);
53     Vlopend=V(:,1:a);
54
55     %Houd bij hoeveel stemmen er tot nu toe zijn binnengekomen.
56     for b=1:rowsU
57         StemmenOudLopend(b,1)=sum(Ulopend(b,:));
```

```

58     StemmenNieuwLopend(b,1)=sum(Vlopend(b,:));
59 end
60 %Sla de Lambda's per keer op in een kolomvector
61 for b=1:rowsU
62     LambdaOpslag(b,a)=StemmenNieuwLopend(b,1)/StemmenOudLopend(b,1);
63 end
64 end
65
66 %Voorspellen
67
68 for c = 1:columnsU
69
70     %Haal de juiste Lambda's uit de opslag. LAMBDA is een diagonaal matrix.
71     for i=1:rowsU
72         LAMBDA(i,i)=LambdaOpslag(i,c);
73     end
74
75     %De daadwerkelijke voorspelling in stemmen
76     Voorspelling=LAMBDA*Echte_Oude_Uitslag_Stemmen;
77     VoorspellingTotaalStemmen=sum(Voorspelling);
78
79     %Zet de daadwerkelijke voorspelling in stemmen om in een daadwerkelijke
80     %voorspelling in percentages.
81     for j=1:rowsU
82         VoorspellingPercentages(j,1)=Voorspelling(j,1)/VoorspellingTotaalStemmen;
83     end
84
85     %Sla de norm in het kwadraat van de daadwerkelijke voorspelling in
86     %percentages tov de huidige einduitslag in percentages op in een
87     %kolomvector
88     MethCNormOpslag(c,1)=norm(VoorspellingPercentages-Echte_Nieuwe_Uitslag_Perc)^2;
89 end

```

4.12. ConvergentieMoment

```
1 function [ConvergentiePunt]=ConvergentieMoment (NormOpslag)
2 % Functie: ConvergentieMoment (NormOpslag)
3 %
4 % Output: [ConvergentiePunt]
5 % *Convergentiepunt is een getal dat aangeeft na hoeveel gemeenten de
6 % voorspelling een gekozen epsilon in de buurt zit van de echte uitslag,
7 % onder de voorwaarde dat de fout nooit meer groter wordt dan epsilon
8 % wanneer er meer gemeenten binnenkomen.
9 %
10 % Uitleg: Deze functie bepaalt ConvergentiePunt.
11 %
12 % Input:
13 % *NormOpslag is een nxl matrix die de waarden van de norm in het
14 % kwadraat van alle voorspellingen van een methode tov de daadwerkelijke
15 % uitslag bevat.
16
17 value=0;
18 epsilon=1e-5; %de gekozen nauwkeurigheid van de norm voor convergentie
19 i=1;
20 [rows,columns]=size(NormOpslag);
21
22 while value==0 && i<rows
23
24     %vind waarde van de fout bij i'de voorspelling
25     fout=NormOpslag(i,1);
26     nalopen=0;
27
28     if fout>=epsilon %Als fout groter dan epsilon, ga verder naar i+1
29         i=i+1;
30     else
31         %Als fout kleiner is dan epsilon controleer of er geen waarden meer
32         %hoger gaan dan epsilon na i T's
33         j=i+1;
34         while j<rows && nalopen==0
35
36             ControleFout=NormOpslag(j,1);
37
38             if ControleFout<epsilon %Fout is nog steeds kleiner dan epsilon
39                 nalopen=0;
40                 j=j+1; %ga naar het volgende element
41             else
42                 %In dit geval is er een waarde die weer boven epsilon gaat
43                 nalopen=1;
44                 %Kijk dan of er na j T's nog waardes boven epsilon
45                 %uitkomen
46                 i=j;
47             end
48         end
49
50         if nalopen==0 %als na de while-loop nalopen==0, dan is i de eerste
51             %waarde die kleiner is dan epsilon
52             value=1;
53         end
54     end
55 end
56
57 ConvergentiePunt=i;
```

4.13. ModelGemeenten

```
1 function [FoutOpslag,LijstPotentieleModelGemeenten]=ModelGemeenten(V,  
2                                     Echte_Uitslag,Volgorde)  
3 % Functie: ModelGemeenten(V,Echte_Uitslag,Volgorde)  
4 %  
5 % Output: [FoutOpslag,LijstPotentieleModelGemeenten]  
6 % *FoutOpslag is een nx2 matrix. In de eerste kolom staan de CBS-codes,  
7 %   in de tweede kolom staan de waarden van de norm in het kwadraat van de  
8 %   verschilvector van de gemeentelijke uitslag ten opzichte van de  
9 %   landelijke uitslag.  
10 % *LijstPotentieleModelGemeenten is een 5x1 matrix met 10 CBS-codes die  
11 %   kleinste waarden hebben voor de norm in het kwadraat in FoutOpslag.  
12 %  
13 % Uitleg: Deze functie onderzoekt wat de fout van gemeentelijke uitslagen  
14 %   is ten opzichte van de echte uitslag.  
15 %  
16 % Input:  
17 % *V is een mxn matrix met de gemeentelijke uitslagen van de huidige  
18 %   verkiezing.  
19 % *Echte_Uitslag is een px1 matrix waar de landelijke uitslag in  
20 %   percentages in staat.  
21 % *Volgorde is een lxm matrix waar de CBS-codes gesorteerd op binnenkomst  
22 %   in staan.  
23  
24 j=0;  
25 [rowsV,columnsV]=size(V);  
26 PotentieleModelGemeente=[];  
27  
28 for i=1:columnsV  
29  
30     % pak de data voor de gemeente (V bevat al percentages en is al  
31     % gesorteerd, dus het eerste gemeente van V komt overeen met de eerste  
32     % gemeente van Volgorde)  
33     for a=1:rowsV  
34         Gemeente(a,1)=V(a,i);  
35     end  
36  
37     % Bereken de fout van de gemeentelijke uitslag ten opzichte van de  
38     % landelijke uitslag en sla deze op met de CBS code in FoutOpslag  
39     Fout=norm(Gemeente-Echte_Uitslag)^2;  
40     FoutOpslag(i,1)=Volgorde(i);  
41     FoutOpslag(i,2)=Fout;  
42 end  
43  
44 %Sorteer FoutOpslag op de norm in het kwadraat van klein naar groot  
45 [Y,Index]=sort(FoutOpslag(:,2),'ascend');  
46 X=FoutOpslag(Index,1); %zet de CBS-codes ook in de juiste volgorde  
47 FoutOpslag=[X,Y]; %FoutOpslag is nu gesorteerd  
48  
49 %Bekijk de top 5 en geef deze in het command window weer  
50 LijstPotentieleModelGemeenten=FoutOpslag(1:10,1);
```