Final Term Project

Introduction to Data Science

Zerin Tasnim

ID: 20-43032-1

Section: D

Description of the Dataset:

The data set refers to clients of a wholesale distributor. It includes the annual spending in monetary units (m.u.) on diverse product categories.

Data Set Information:

Provide all relevant information about data set.

Attribute Information:

1) FRESH: annual spending (m.u.) on fresh products (Continuous);

2) MILK: annual spending (m.u.) on milk products (Continuous);

3) GROCERY: annual spending (m.u.) on grocery products (Continuous);

4) FROZEN: annual spending (m.u.) on frozen products (Continuous)

5) DETERGENTS_PAPER: annual spending (m.u.) on detergents and paper products (Continuous)

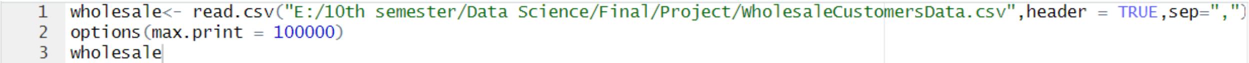
6) DELICATESSEN: annual spending (m.u.) on and delicatessen products (Continuous);

7) CHANNEL: customersâ€™ Channel - Horeca (Hotel/Restaurant/CafÃ©) or Retail channel (Nominal)

8) REGION: customersâ€™ Region â€“ Lisbon, Oporto or Other (Nominal)

Dataset Link: https://archive-beta.ics.uci.edu/dataset/292/wholesale+customers

Import the data set as csv and print the data set:



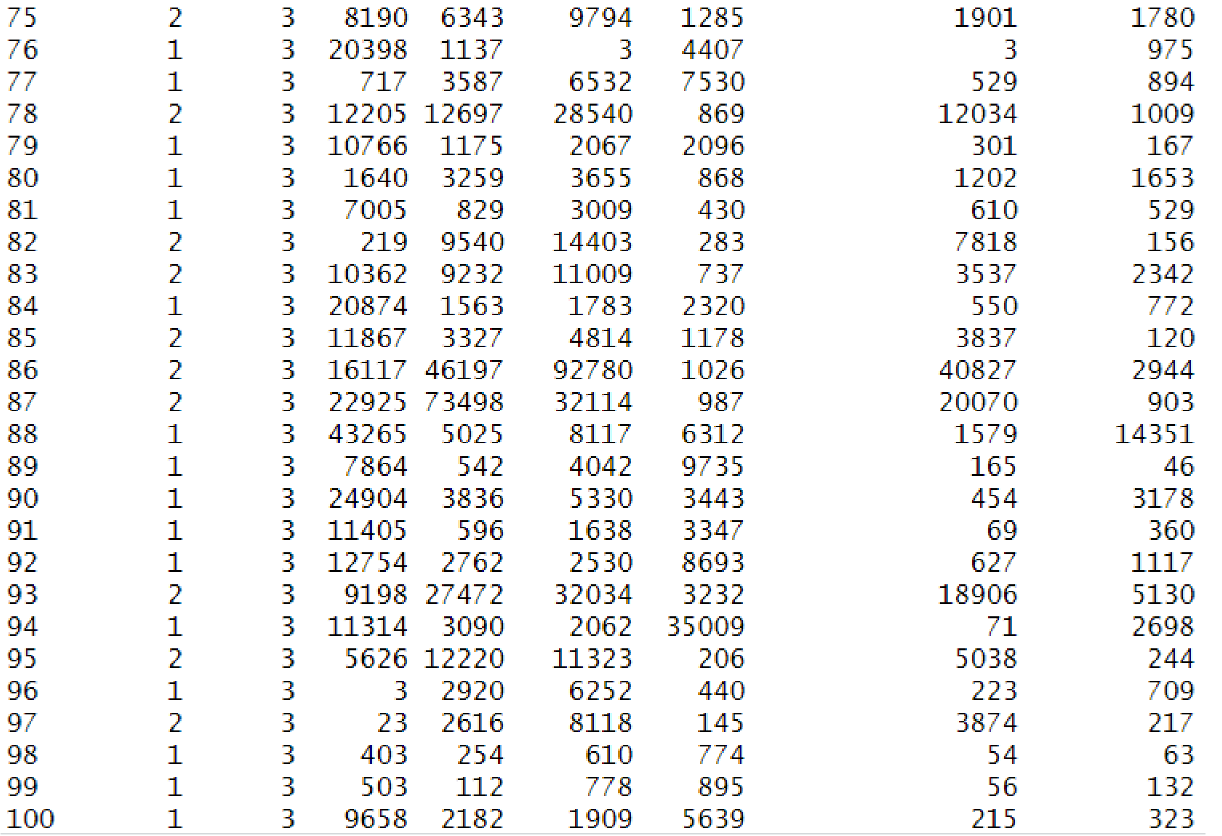
```
wholesale<- read.csv("E:/10th semester/Data Science/Final/Project/WholesaleCustomersData.csv",header = TRUE,sep=",")
options(max.print = 100000)
wholesale
```

Here is the code of import the dataset as csv file. In this code also has the location of the csv dataset file.

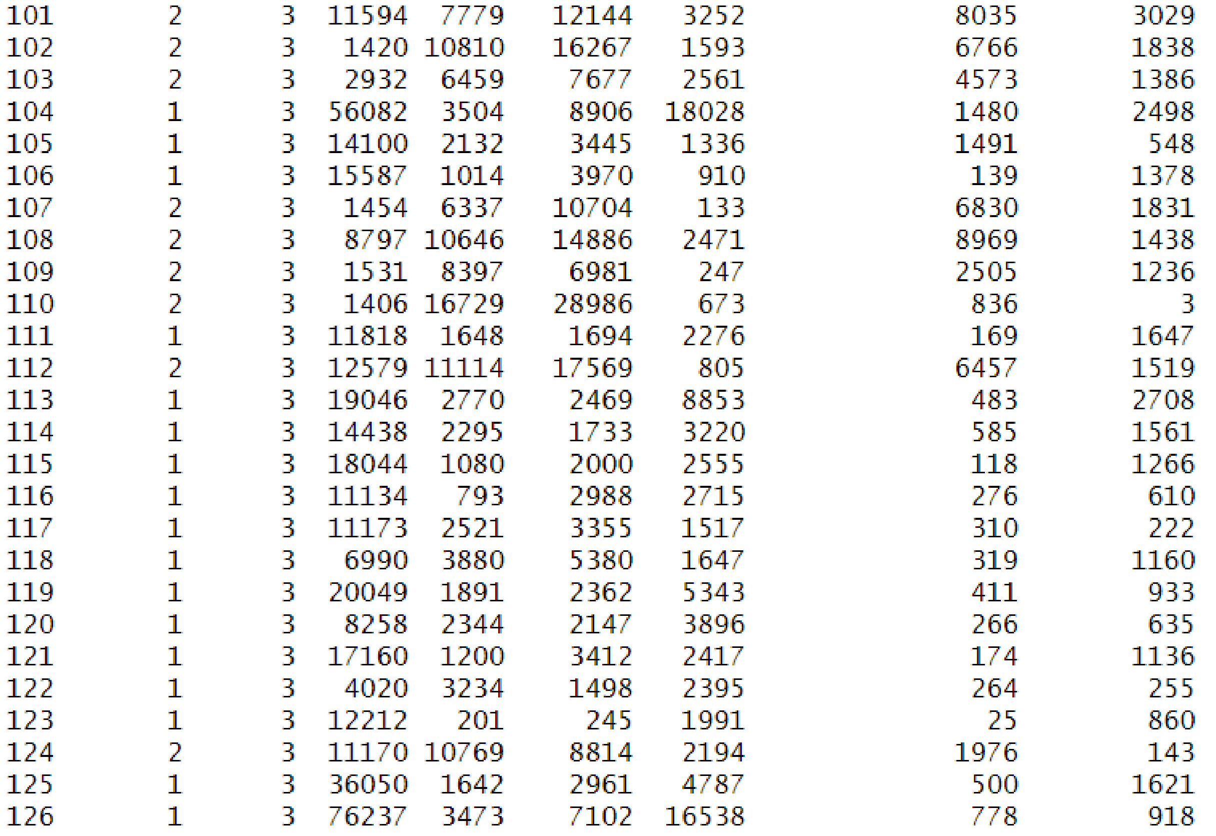
Output:

```
   Channel Region  Fresh  Milk Grocery Frozen Detergents_Paper Delicassen
1        2      3  12669  9656    7561    214             2674       1338
2        2      3   7057  9810    9568   1762             3293       1776
3        2      3   6353  8808    7684   2405             3516       7844
4        1      3  13265  1196    4221   6404              507       1788
5        2      3  22615  5410    7198   3915             1777       5185
6        2      3   9413  8259    5126    666             1795       1451
7        2      3  12126  3199    6975    480             3140        545
8        2      3   7579  4956    9426   1669             3321       2566
9        1      3   5963  3648    6192    425             1716        750
10       2      3   6006 11093   18881   1159             7425       2098
11       2      3   3366  5403   12974   4400             5977       1744
12       2      3  13146  1124    4523   1420              549        497
13       2      3  31714 12319   11757    287             3881       2931
14       2      3  21217  6208   14982   3095             6707        602
15       2      3  24653  9465   12091    294             5058       2168
16       1      3  10253  1114    3821    397              964        412
17       2      3   1020  8816   12121    134             4508       1080
18       1      3   5876  6157    2933    839              370       4478
19       2      3  18601  6327   10099   2205             2767       3181
20       1      3   7780  2495    9464    669             2518        501
21       2      3  17546  4519    4602   1066             2259       2124
22       1      3   5567   871    2010   3383              375        569
23       1      3  31276  1917    4469   9408             2381       4334
24       2      3  26373 36423   22019   5154             4337      16523
25       2      3  22647  9776   13792   2915             4482       5778
26       2      3  16165  4230    7595    201             4003         57
27       1      3   9898   961    2861   3151              242        833
28       1      3  14276   803    3045    485              100        518
29       2      3   4113 20484   25957   1158             8604       5206
30       1      3  43088  2100    2609   1200             1107        823
31       1      3  18815  3610   11107   1148             2134       2963
32       1      3   2612  4339    3133   2088              820        985
33       1      3  21632  1318    2886    266              918        405
34       1      3  29729  4786    7326   6130              361       1083
35       1      3   1502  1979    2262    425              483        395
36       2      3    688  5491   11091    833             4239        436
37       1      3  29955  4362    5428   1729              862       4626
38       2      3  15168 10556   12477   1920             6506        714
39       2      3   4591 15729   16709     33             6956        433
40       1      3  56159   555     902  10002              212       2916
41       1      3  24025  4332    4757   9510             1145       5864
42       1      3  19176  3065    5956   2033             2575       2802
43       2      3  10850  7555   14961    188             6899         46
44       2      3    630 11095   23998    787             9529         72
45       2      3   9670  7027   10471    541             4618         65
46       2      3   5181 22044   21531   1740             7353       4985
47       2      3   3103 14069   21955   1668             6792       1452
48       2      3  44466 54259   55571   7782            24171       6465
```

49	2	3	11519	6152	10868	584	5121	1476
50	2	3	4967	21412	28921	1798	13583	1163
51	1	3	6269	1095	1980	3860	609	2162
52	1	3	3347	4051	6996	239	1538	301
53	2	3	40721	3916	5876	532	2587	1278
54	2	3	491	10473	11532	744	5611	224
55	1	3	27329	1449	1947	2436	204	1333
56	1	3	5264	3683	5005	1057	2024	1130
57	2	3	4098	29892	26866	2616	17740	1340
58	2	3	5417	9933	10487	38	7572	1282
59	1	3	13779	1970	1648	596	227	436
60	1	3	6137	5360	8040	129	3084	1603
61	2	3	8590	3045	7854	96	4095	225
62	2	3	35942	38369	59598	3254	26701	2017
63	2	3	7823	6245	6544	4154	4074	964
64	2	3	9396	11601	15775	2896	7677	1295
65	1	3	4760	1227	3250	3724	1247	1145
66	2	3	85	20959	45828	36	24231	1423
67	1	3	9	1534	7417	175	3468	27
68	2	3	19913	6759	13462	1256	5141	834
69	1	3	2446	7260	3993	5870	788	3095
70	1	3	8352	2820	1293	779	656	144
71	1	3	16705	2037	3202	10643	116	1365
72	1	3	18291	1266	21042	5373	4173	14472
73	1	3	4420	5139	2661	8872	1321	181
74	2	3	19899	5332	8713	8132	764	648



75	2	3	8190	6343	9794	1285	1901	1780
76	1	3	20398	1137	3	4407	3	975
77	1	3	717	3587	6532	7530	529	894
78	2	3	12205	12697	28540	869	12034	1009
79	1	3	10766	1175	2067	2096	301	167
80	1	3	1640	3259	3655	868	1202	1653
81	1	3	7005	829	3009	430	610	529
82	2	3	219	9540	14403	283	7818	156
83	2	3	10362	9232	11009	737	3537	2342
84	1	3	20874	1563	1783	2320	550	772
85	2	3	11867	3327	4814	1178	3837	120
86	2	3	16117	46197	92780	1026	40827	2944
87	2	3	22925	73498	32114	987	20070	903
88	1	3	43265	5025	8117	6312	1579	14351
89	1	3	7864	542	4042	9735	165	46
90	1	3	24904	3836	5330	3443	454	3178
91	1	3	11405	596	1638	3347	69	360
92	1	3	12754	2762	2530	8693	627	1117
93	2	3	9198	27472	32034	3232	18906	5130
94	1	3	11314	3090	2062	35009	71	2698
95	2	3	5626	12220	11323	206	5038	244
96	1	3	3	2920	6252	440	223	709
97	2	3	23	2616	8118	145	3874	217
98	1	3	403	254	610	774	54	63
99	1	3	503	112	778	895	56	132
100	1	3	9658	2182	1909	5639	215	323



101	2	3	11594	7779	12144	3252	8035	3029
102	2	3	1420	10810	16267	1593	6766	1838
103	2	3	2932	6459	7677	2561	4573	1386
104	1	3	56082	3504	8906	18028	1480	2498
105	1	3	14100	2132	3445	1336	1491	548
106	1	3	15587	1014	3970	910	139	1378
107	2	3	1454	6337	10704	133	6830	1831
108	2	3	8797	10646	14886	2471	8969	1438
109	2	3	1531	8397	6981	247	2505	1236
110	2	3	1406	16729	28986	673	836	3
111	1	3	11818	1648	1694	2276	169	1647
112	2	3	12579	11114	17569	805	6457	1519
113	1	3	19046	2770	2469	8853	483	2708
114	1	3	14438	2295	1733	3220	585	1561
115	1	3	18044	1080	2000	2555	118	1266
116	1	3	11134	793	2988	2715	276	610
117	1	3	11173	2521	3355	1517	310	222
118	1	3	6990	3880	5380	1647	319	1160
119	1	3	20049	1891	2362	5343	411	933
120	1	3	8258	2344	2147	3896	266	635
121	1	3	17160	1200	3412	2417	174	1136
122	1	3	4020	3234	1498	2395	264	255
123	1	3	12212	201	245	1991	25	860
124	2	3	11170	10769	8814	2194	1976	143
125	1	3	36050	1642	2961	4787	500	1621
126	1	3	76237	3473	7102	16538	778	918

127	1	3	19219	1840	1658	8195	349	483
128	2	3	21465	7243	10685	880	2386	2749
129	1	3	140	8847	3823	142	1062	3
130	1	3	42312	926	1510	1718	410	1819
131	1	3	7149	2428	699	6316	395	911
132	1	3	2101	589	314	346	70	310
133	1	3	14903	2032	2479	576	955	328
134	1	3	9434	1042	1235	436	256	396
135	1	3	7388	1882	2174	720	47	537
136	1	3	6300	1289	2591	1170	199	326
137	1	3	4625	8579	7030	4575	2447	1542
138	1	3	3087	8080	8282	661	721	36
139	1	3	13537	4257	5034	155	249	3271
140	1	3	5387	4979	3343	825	637	929
141	1	3	17623	4280	7305	2279	960	2616
142	1	3	30379	13252	5189	321	51	1450
143	1	3	37036	7152	8253	2995	20	3
144	1	3	10405	1596	1096	8425	399	318
145	1	3	18827	3677	1988	118	516	201
146	2	3	22039	8384	34792	42	12591	4430
147	1	3	7769	1936	2177	926	73	520
148	1	3	9203	3373	2707	1286	1082	526
149	1	3	5924	584	542	4052	283	434
150	1	3	31812	1433	1651	800	113	1440
151	1	3	16225	1825	1765	853	170	1067
152	1	3	1289	3328	2022	531	255	1774
153	1	3	18840	1371	3135	3001	352	184
154	1	3	3463	9250	2368	779	302	1627
155	1	3	622	55	137	75	7	8
156	2	3	1989	10690	19460	233	11577	2153
157	2	3	3830	5291	14855	317	6694	3182
158	1	3	17773	1366	2474	3378	811	418
159	2	3	2861	6570	9618	930	4004	1682
160	2	3	355	7704	14682	398	8077	303
161	2	3	1725	3651	12822	824	4424	2157
162	1	3	12434	540	283	1092	3	2233
163	1	3	15177	2024	3810	2665	232	610
164	2	3	5531	15726	26870	2367	13726	446
165	2	3	5224	7603	8584	2540	3674	238
166	2	3	15615	12653	19858	4425	7108	2379
167	2	3	4822	6721	9170	993	4973	3637
168	1	3	2926	3195	3268	405	1680	693
169	1	3	5809	735	803	1393	79	429
170	1	3	5414	717	2155	2399	69	750
171	2	3	260	8675	13430	1116	7015	323
172	2	3	200	25862	19816	651	8773	6250
173	1	3	955	5479	6536	333	2840	707
174	2	3	514	7677	19805	937	9836	716
175	1	3	286	1208	5241	2515	153	1442
176	2	3	2343	7845	11874	52	4196	1697

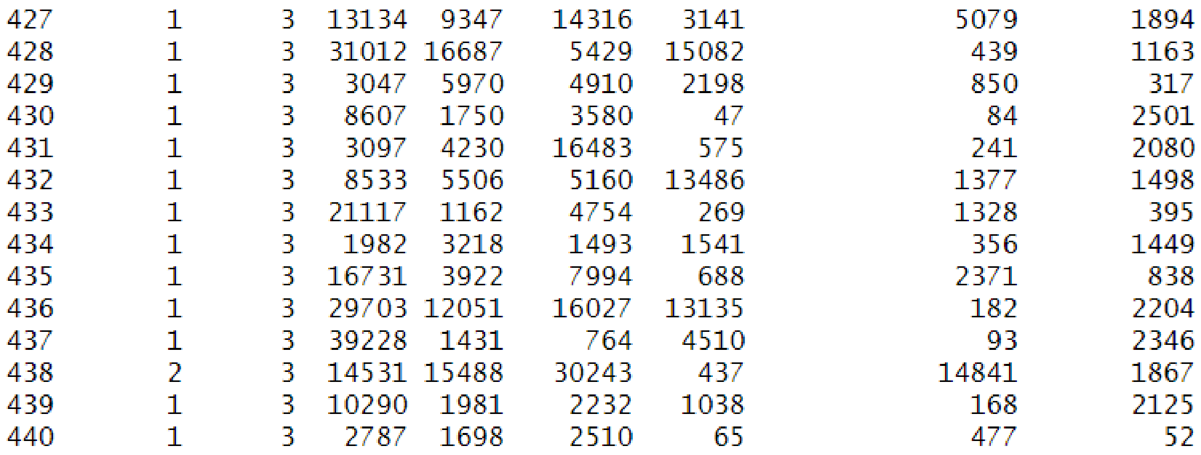
177	1	3	45640	6958	6536	7368	1532	230
178	1	3	12759	7330	4533	1752	20	2631
179	1	3	11002	7075	4945	1152	120	395
180	1	3	3157	4888	2500	4477	273	2165
181	1	3	12356	6036	8887	402	1382	2794
182	1	3	112151	29627	18148	16745	4948	8550
183	1	3	694	8533	10518	443	6907	156
184	1	3	36847	43950	20170	36534	239	47943
185	1	3	327	918	4710	74	334	11
186	1	3	8170	6448	1139	2181	58	247
187	1	3	3009	521	854	3470	949	727
188	1	3	2438	8002	9819	6269	3459	3
189	2	3	8040	7639	11687	2758	6839	404
190	2	3	834	11577	11522	275	4027	1856
191	1	3	16936	6250	1981	7332	118	64
192	1	3	13624	295	1381	890	43	84
193	1	3	5509	1461	2251	547	187	409
194	2	3	180	3485	20292	959	5618	666
195	1	3	7107	1012	2974	806	355	1142
196	1	3	17023	5139	5230	7888	330	1755
197	1	1	30624	7209	4897	18711	763	2876
198	2	1	2427	7097	10391	1127	4314	1468
199	1	1	11686	2154	6824	3527	592	697
200	1	1	9670	2280	2112	520	402	347
201	2	1	3067	13240	23127	3941	9959	731
202	2	1	4484	14399	24708	3549	14235	1681
203	1	1	25203	11487	9490	5065	284	6854
204	1	1	583	685	2216	469	954	18
205	1	1	1956	891	5226	1383	5	1328
206	2	1	1107	11711	23596	955	9265	710
207	1	1	6373	780	950	878	288	285
208	2	1	2541	4737	6089	2946	5316	120
209	1	1	1537	3748	5838	1859	3381	806
210	2	1	5550	12729	16767	864	12420	797
211	1	1	18567	1895	1393	1801	244	2100
212	2	1	12119	28326	39694	4736	19410	2870
213	1	1	7291	1012	2062	1291	240	1775
214	1	1	3317	6602	6861	1329	3961	1215
215	2	1	2362	6551	11364	913	5957	791
216	1	1	2806	10765	15538	1374	5828	2388
217	2	1	2532	16599	36486	179	13308	674
218	1	1	18044	1475	2046	2532	130	1158
219	2	1	18	7504	15205	1285	4797	6372
220	1	1	4155	367	1390	2306	86	130
221	1	1	14755	899	1382	1765	56	749
222	1	1	5396	7503	10646	91	4167	239
223	1	1	5041	1115	2856	7496	256	375
224	2	1	2790	2527	5265	5612	788	1360
225	1	1	7274	659	1499	784	70	659
226	1	1	12680	3243	4157	660	761	786

227	2	1	20782	5921	9212	1759	2568	1553
228	1	1	4042	2204	1563	2286	263	689
229	1	1	1869	577	572	950	4762	203
230	1	1	8656	2746	2501	6845	694	980
231	2	1	11072	5989	5615	8321	955	2137
232	1	1	2344	10678	3828	1439	1566	490
233	1	1	25962	1780	3838	638	284	834
234	1	1	964	4984	3316	937	409	7
235	1	1	15603	2703	3833	4260	325	2563
236	1	1	1838	6380	2824	1218	1216	295
237	1	1	8635	820	3047	2312	415	225
238	1	1	18692	3838	593	4634	28	1215
239	1	1	7363	475	585	1112	72	216
240	1	1	47493	2567	3779	5243	828	2253
241	1	1	22096	3575	7041	11422	343	2564
242	1	1	24929	1801	2475	2216	412	1047
243	1	1	18226	659	2914	3752	586	578
244	1	1	11210	3576	5119	561	1682	2398
245	1	1	6202	7775	10817	1183	3143	1970
246	2	1	3062	6154	13916	230	8933	2784
247	1	1	8885	2428	1777	1777	430	610
248	1	1	13569	346	489	2077	44	659
249	1	1	15671	5279	2406	559	562	572
250	1	1	8040	3795	2070	6340	918	291
251	1	1	3191	1993	1799	1730	234	710
252	2	1	6134	23133	33586	6746	18594	5121
253	1	1	6623	1860	4740	7683	205	1693
254	1	1	29526	7961	16966	432	363	1391
255	1	1	10379	17972	4748	4686	1547	3265
256	1	1	31614	489	1495	3242	111	615
257	1	1	11092	5008	5249	453	392	373
258	1	1	8475	1931	1883	5004	3593	987
259	1	1	56083	4563	2124	6422	730	3321
260	1	1	53205	4959	7336	3012	967	818
261	1	1	9193	4885	2157	327	780	548
262	1	1	7858	1110	1094	6818	49	287
263	1	1	23257	1372	1677	982	429	655
264	1	1	2153	1115	6684	4324	2894	411
265	2	1	1073	9679	15445	61	5980	1265
266	1	1	5909	23527	13699	10155	830	3636
267	2	1	572	9763	22182	2221	4882	2563
268	1	1	20893	1222	2576	3975	737	3628
269	2	1	11908	8053	19847	1069	6374	698
270	1	1	15218	258	1138	2516	333	204
271	1	1	4720	1032	975	5500	197	56
272	1	1	2083	5007	1563	1120	147	1550
273	1	1	514	8323	6869	529	93	1040
274	1	3	36817	3045	1493	4802	210	1824
275	1	3	894	1703	1841	744	759	1153
276	1	3	680	1610	223	862	96	379

277	1	3	27901	3749	6964	4479	603	2503
278	1	3	9061	829	683	16919	621	139
279	1	3	11693	2317	2543	5845	274	1409
280	2	3	17360	6200	9694	1293	3620	1721
281	1	3	3366	2884	2431	977	167	1104
282	2	3	12238	7108	6235	1093	2328	2079
283	1	3	49063	3965	4252	5970	1041	1404
284	1	3	25767	3613	2013	10303	314	1384
285	1	3	68951	4411	12609	8692	751	2406
286	1	3	40254	640	3600	1042	436	18
287	1	3	7149	2247	1242	1619	1226	128
288	1	3	15354	2102	2828	8366	386	1027
289	1	3	16260	594	1296	848	445	258
290	1	3	42786	286	471	1388	32	22
291	1	3	2708	2160	2642	502	965	1522
292	1	3	6022	3354	3261	2507	212	686
293	1	3	2838	3086	4329	3838	825	1060
294	2	2	3996	11103	12469	902	5952	741
295	1	2	21273	2013	6550	909	811	1854
296	2	2	7588	1897	5234	417	2208	254
297	1	2	19087	1304	3643	3045	710	898
298	2	2	8090	3199	6986	1455	3712	531
299	2	2	6758	4560	9965	934	4538	1037
300	1	2	444	879	2060	264	290	259
301	2	2	16448	6243	6360	824	2662	2005
302	2	2	5283	13316	20399	1809	8752	172
303	2	2	2886	5302	9785	364	6236	555
304	2	2	2599	3688	13829	492	10069	59
305	2	2	161	7460	24773	617	11783	2410
306	2	2	243	12939	8852	799	3909	211
307	2	2	6468	12867	21570	1840	7558	1543
308	1	2	17327	2374	2842	1149	351	925
309	1	2	6987	1020	3007	416	257	656
310	2	2	918	20655	13567	1465	6846	806
311	1	2	7034	1492	2405	12569	299	1117
312	1	2	29635	2335	8280	3046	371	117
313	2	2	2137	3737	19172	1274	17120	142
314	1	2	9784	925	2405	4447	183	297
315	1	2	10617	1795	7647	1483	857	1233
316	2	2	1479	14982	11924	662	3891	3508
317	1	2	7127	1375	2201	2679	83	1059
318	1	2	1182	3088	6114	978	821	1637
319	1	2	11800	2713	3558	2121	706	51
320	2	2	9759	25071	17645	1128	12408	1625
321	1	2	1774	3696	2280	514	275	834
322	1	2	9155	1897	5167	2714	228	1113
323	1	2	15881	713	3315	3703	1470	229
324	1	2	13360	944	11593	915	1679	573
325	1	2	25977	3587	2464	2369	140	1092
326	1	2	32717	16784	13626	60869	1272	5609

327	1	2	4414	1610	1431	3498	387	834
328	1	2	542	899	1664	414	88	522
329	1	2	16933	2209	3389	7849	210	1534
330	1	2	5113	1486	4583	5127	492	739
331	1	2	9790	1786	5109	3570	182	1043
332	2	2	11223	14881	26839	1234	9606	1102
333	1	2	22321	3216	1447	2208	178	2602
334	2	2	8565	4980	67298	131	38102	1215
335	2	2	16823	928	2743	11559	332	3486
336	2	2	27082	6817	10790	1365	4111	2139
337	1	2	13970	1511	1330	650	146	778
338	1	2	9351	1347	2611	8170	442	868
339	1	2	3	333	7021	15601	15	550
340	1	2	2617	1188	5332	9584	573	1942
341	2	3	381	4025	9670	388	7271	1371
342	2	3	2320	5763	11238	767	5162	2158
343	1	3	255	5758	5923	349	4595	1328
344	2	3	1689	6964	26316	1456	15469	37
345	1	3	3043	1172	1763	2234	217	379
346	1	3	1198	2602	8335	402	3843	303
347	2	3	2771	6939	15541	2693	6600	1115
348	2	3	27380	7184	12311	2809	4621	1022
349	1	3	3428	2380	2028	1341	1184	665
350	2	3	5981	14641	20521	2005	12218	445
351	1	3	3521	1099	1997	1796	173	995
352	2	3	1210	10044	22294	1741	12638	3137
353	1	3	608	1106	1533	830	90	195
354	2	3	117	6264	21203	228	8682	1111
355	1	3	14039	7393	2548	6386	1333	2341
356	1	3	190	727	2012	245	184	127
357	1	3	22686	134	218	3157	9	548
358	2	3	37	1275	22272	137	6747	110
359	1	3	759	18664	1660	6114	536	4100
360	1	3	796	5878	2109	340	232	776
361	1	3	19746	2872	2006	2601	468	503
362	1	3	4734	607	864	1206	159	405
363	1	3	2121	1601	2453	560	179	712
364	1	3	4627	997	4438	191	1335	314
365	1	3	2615	873	1524	1103	514	468
366	2	3	4692	6128	8025	1619	4515	3105
367	1	3	9561	2217	1664	1173	222	447
368	1	3	3477	894	534	1457	252	342
369	1	3	22335	1196	2406	2046	101	558
370	1	3	6211	337	683	1089	41	296
371	2	3	39679	3944	4955	1364	523	2235
372	1	3	20105	1887	1939	8164	716	790
373	1	3	3884	3801	1641	876	397	4829
374	2	3	15076	6257	7398	1504	1916	3113
375	1	3	6338	2256	1668	1492	311	686
376	1	3	5841	1450	1162	597	476	70

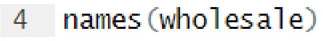
377	2	3	3136	8630	13586	5641		4666	1426
378	1	3	38793	3154	2648	1034		96	1242
379	1	3	3225	3294	1902	282		68	1114
380	2	3	4048	5164	10391	130		813	179
381	1	3	28257	944	2146	3881		600	270
382	1	3	17770	4591	1617	9927		246	532
383	1	3	34454	7435	8469	2540		1711	2893
384	1	3	1821	1364	3450	4006		397	361
385	1	3	10683	21858	15400	3635		282	5120
386	1	3	11635	922	1614	2583		192	1068
387	1	3	1206	3620	2857	1945		353	967
388	1	3	20918	1916	1573	1960		231	961
389	1	3	9785	848	1172	1677		200	406
390	1	3	9385	1530	1422	3019		227	684
391	1	3	3352	1181	1328	5502		311	1000
392	1	3	2647	2761	2313	907		95	1827
393	1	3	518	4180	3600	659		122	654
394	1	3	23632	6730	3842	8620		385	819
395	1	3	12377	865	3204	1398		149	452
396	1	3	9602	1316	1263	2921		841	290
397	2	3	4515	11991	9345	2644		3378	2213
398	1	3	11535	1666	1428	6838		64	743
399	1	3	11442	1032	582	5390		74	247
400	1	3	9612	577	935	1601		469	375
401	1	3	4446	906	1238	3576		153	1014
402	1	3	27167	2801	2128	13223		92	1902
403	1	3	26539	4753	5091	220		10	340
404	1	3	25606	11006	4604	127		632	288
405	1	3	18073	4613	3444	4324		914	715
406	1	3	6884	1046	1167	2069		593	378
407	1	3	25066	5010	5026	9806		1092	960
408	2	3	7362	12844	18683	2854		7883	553
409	2	3	8257	3880	6407	1646		2730	344
410	1	3	8708	3634	6100	2349		2123	5137
411	1	3	6633	2096	4563	1389		1860	1892
412	1	3	2126	3289	3281	1535		235	4365
413	1	3	97	3605	12400	98		2970	62
414	1	3	4983	4859	6633	17866		912	2435
415	1	3	5969	1990	3417	5679		1135	290
416	2	3	7842	6046	8552	1691		3540	1874
417	2	3	4389	10940	10908	848		6728	993
418	1	3	5065	5499	11055	364		3485	1063
419	2	3	660	8494	18622	133		6740	776
420	1	3	8861	3783	2223	633		1580	1521
421	1	3	4456	5266	13227	25		6818	1393
422	2	3	17063	4847	9053	1031		3415	1784
423	1	3	26400	1377	4172	830		948	1218
424	2	3	17565	3686	4657	1059		1803	668
425	2	3	16980	2884	12232	874		3213	249
426	1	3	11243	2408	2593	15348		108	1886



```
427        1       3  13134  9347    14316   3141             5079        1894
428        1       3  31012 16687     5429  15082              439        1163
429        1       3   3047  5970     4910   2198              850         317
430        1       3   8607  1750     3580     47               84        2501
431        1       3   3097  4230    16483    575              241        2080
432        1       3   8533  5506     5160  13486             1377        1498
433        1       3  21117  1162     4754    269             1328         395
434        1       3   1982  3218     1493   1541              356        1449
435        1       3  16731  3922     7994    688             2371         838
436        1       3  29703 12051    16027  13135              182        2204
437        1       3  39228  1431      764   4510               93        2346
438        2       3  14531 15488    30243    437            14841        1867
439        1       3  10290  1981     2232   1038              168        2125
440        1       3   2787  1698     2510     65              477          52
```

It is the output of the dataset which is import in Rstudio.

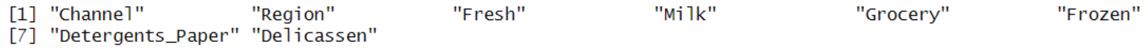
To see the column name of the data set:



```
names(wholesale)
```

This code is to see the column name of the dataset. Here with this code can see the attributes names.

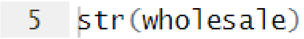
Output:



```
[1] "Channel"          "Region"           "Fresh"            "Milk"             "Grocery"          "Frozen"          
[7] "Detergents_Paper" "Delicassen"
```

The output of the name() function where we can see the attributes of the dataset.

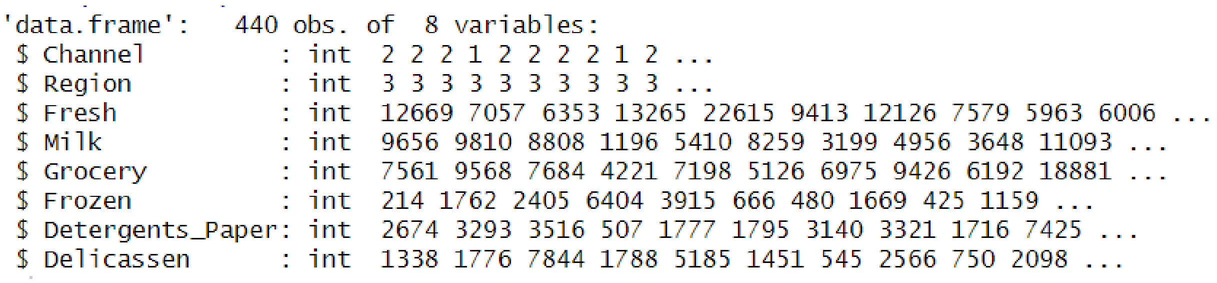
Summary of the structure of data set:



```
str(wholesale)
```

Here is the code to see the summary of the structure of dataset.

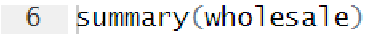
Output:



```
'data.frame':   440 obs. of  8 variables:
 $ Channel         : int  2 2 2 1 2 2 2 2 1 2 ...
 $ Region          : int  3 3 3 3 3 3 3 3 3 3 ...
 $ Fresh           : int  12669 7057 6353 13265 22615 9413 12126 7579 5963 6006 ...
 $ Milk            : int  9656 9810 8808 1196 5410 8259 3199 4956 3648 11093 ...
 $ Grocery         : int  7561 9568 7684 4221 7198 5126 6975 9426 6192 18881 ...
 $ Frozen          : int  214 1762 2405 6404 3915 666 480 1669 425 1159 ...
 $ Detergents_Paper: int  2674 3293 3516 507 1777 1795 3140 3321 1716 7425 ...
 $ Delicassen      : int  1338 1776 7844 1788 5185 1451 545 2566 750 2098 ...
```

In the output we can see the summary of the structure of the dataset. The dataset has 440 observations (rows) and 8 variables (columns). Also here is showed the data types of the dataset.

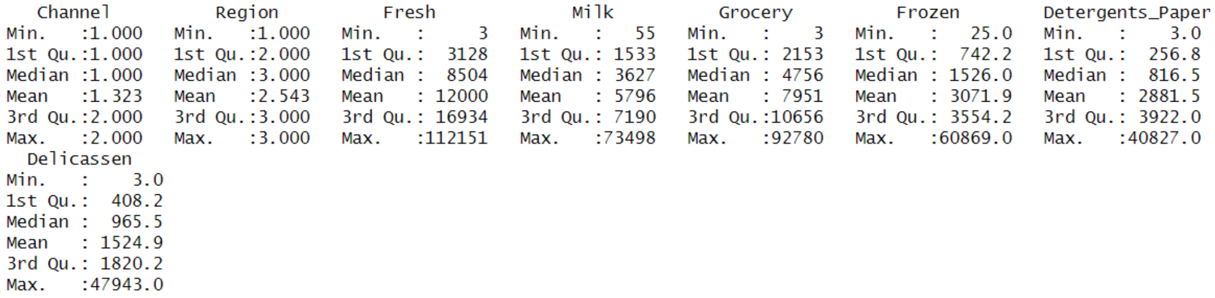
Descriptive Statistics Using summary() Function:



```
summary(wholesale)
```

Here is the code to see the descriptive Statistics. To see descriptive statistic we use the summary() function.

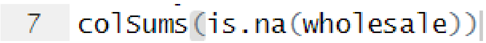
Output:



```
    Channel          Region          Fresh             Milk          Grocery          Frozen        Detergents_Paper
 Min.   :1.000   Min.   :1.000   Min.   :     3   Min.   :   55   Min.   :    3   Min.   :   25.0   Min.   :    3.0  
 1st Qu.:1.000   1st Qu.:2.000   1st Qu.:  3128   1st Qu.: 1533   1st Qu.: 2153   1st Qu.:  742.2   1st Qu.:  256.8  
 Median :1.000   Median :3.000   Median :  8504   Median : 3627   Median : 4756   Median : 1526.0   Median :  816.5  
 Mean   :1.323   Mean   :2.543   Mean   : 12000   Mean   : 5796   Mean   : 7951   Mean   : 3071.9   Mean   : 2881.5  
 3rd Qu.:2.000   3rd Qu.:3.000   3rd Qu.: 16934   3rd Qu.: 7190   3rd Qu.:10656   3rd Qu.: 3554.2   3rd Qu.: 3922.0  
 Max.   :2.000   Max.   :3.000   Max.   :112151   Max.   :73498   Max.   :92780   Max.   :60869.0   Max.   :40827.0  
   Delicassen     
 Min.   :    3.0  
 1st Qu.:  408.2  
 Median :  965.5  
 Mean   : 1524.9  
 3rd Qu.: 1820.2  
 Max.   :47943.0  
```

In the output here min, max, median, and mean are shown.

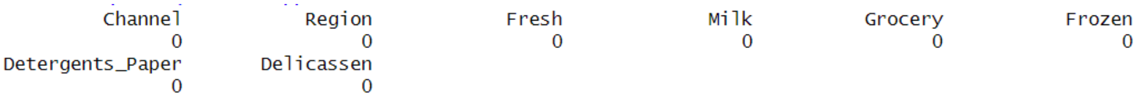
Counting number of Missing values in each column:



```
colSums(is.na(wholesale))
```

Here the code for counting number of missing values in each column.

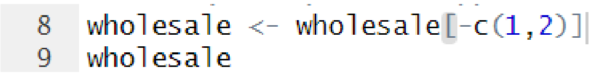
Output:



```
         Channel           Region            Fresh             Milk          Grocery           Frozen 
               0                0                0                0                0                0 
Detergents_Paper       Delicassen 
               0                0 
```

In this output we can see there is no missing value.

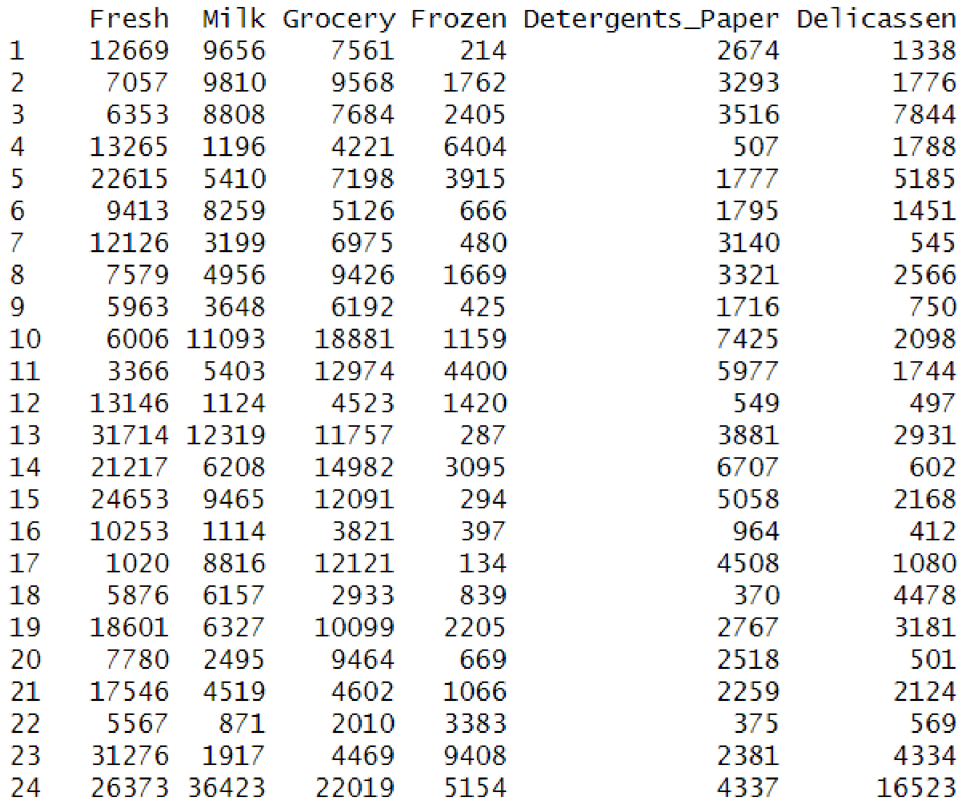
Data Preparation:



```
wholesale <- wholesale[-c(1,2)]
wholesale
```

All the attributes are of same scale except “channel” and “region”. We can ignore those attributes for clustering and normalization is not required.

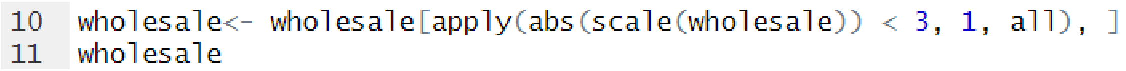
Output:



```
     Fresh  Milk Grocery Frozen Detergents_Paper Delicassen
1    12669  9656    7561    214             2674       1338
2     7057  9810    9568   1762             3293       1776
3     6353  8808    7684   2405             3516       7844
4    13265  1196    4221   6404              507       1788
5    22615  5410    7198   3915             1777       5185
6     9413  8259    5126    666             1795       1451
7    12126  3199    6975    480             3140        545
8     7579  4956    9426   1669             3321       2566
9     5963  3648    6192    425             1716        750
10    6006 11093   18881   1159             7425       2098
11    3366  5403   12974   4400             5977       1744
12   13146  1124    4523   1420              549        497
13   31714 12319   11757    287             3881       2931
14   21217  6208   14982   3095             6707        602
15   24653  9465   12091    294             5058       2168
16   10253  1114    3821    397              964        412
17    1020  8816   12121    134             4508       1080
18    5876  6157    2933    839              370       4478
19   18601  6327   10099   2205             2767       3181
20    7780  2495    9464    669             2518        501
21   17546  4519    4602   1066             2259       2124
22    5567   871    2010   3383              375        569
23   31276  1917    4469   9408             2381       4334
24   26373 36423   22019   5154             4337      16523
```

In this output we can see channel and region attribute are removed.

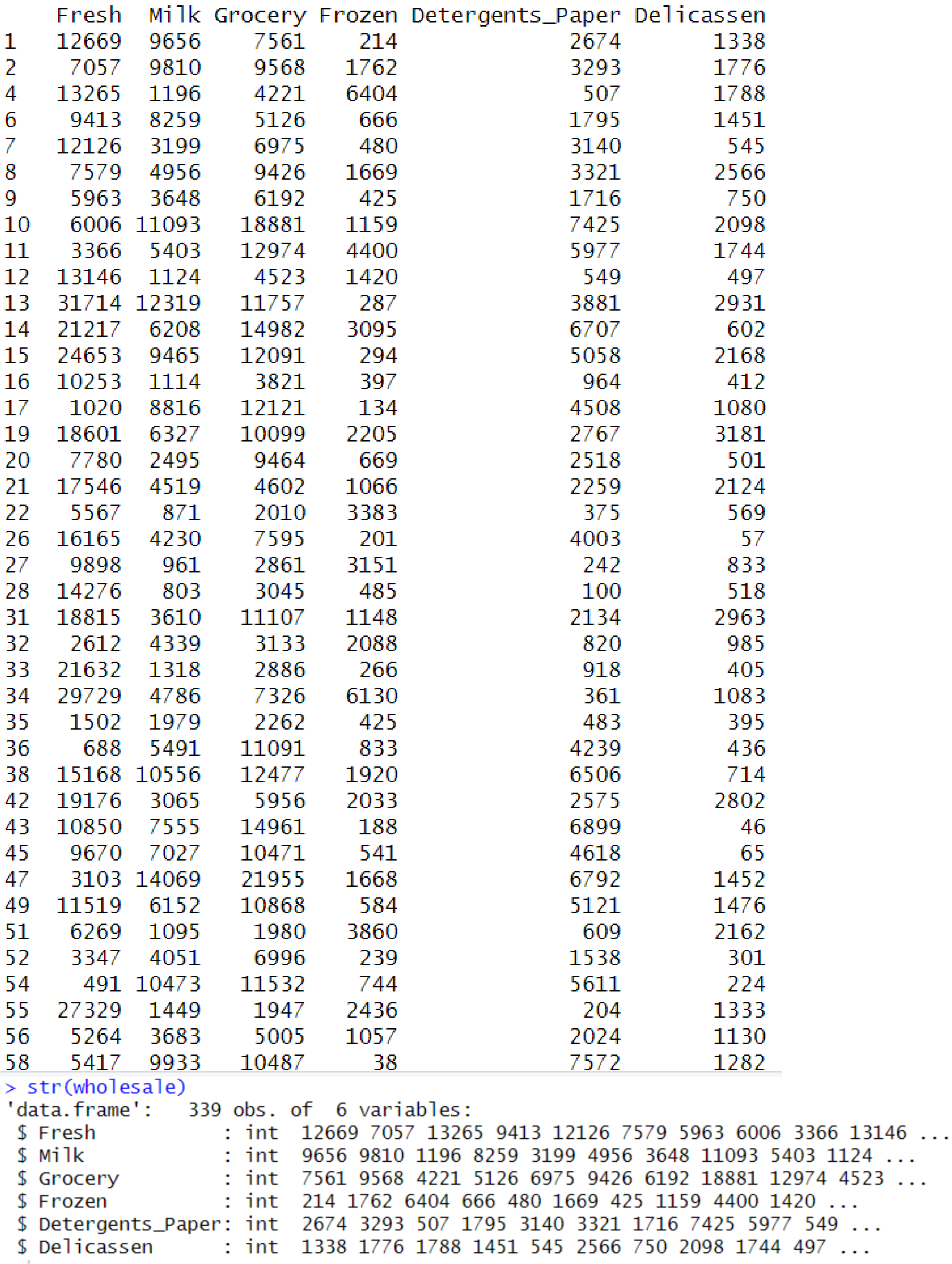
Remove Outlier:



```
wholesale<- wholesale[apply(abs(scale(wholesale)) < 3, 1, all), ]
wholesale
```

This code is use to remove outliers using z score method. Here scale normalize the data then abs is used to calculate absolute value. Rows whose z score is greater than 3 and less than -3 is removed.

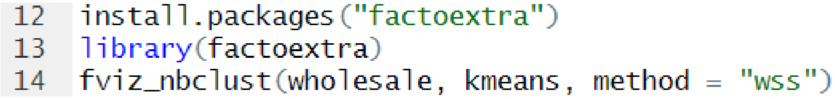
Output:



```
    Fresh   Milk Grocery Frozen Detergents_Paper Delicassen
1   12669   9656    7561    214             2674       1338
2    7057   9810    9568   1762             3293       1776
4   13265   1196    4221   6404              507       1788
6    9413   8259    5126    666             1795       1451
7   12126   3199    6975    480             3140        545
8    7579   4956    9426   1669             3321       2566
9    5963   3648    6192    425             1716        750
10   6006  11093   18881   1159             7425       2098
11   3366   5403   12974   4400             5977       1744
12  13146   1124    4523   1420              549        497
13  31714  12319   11757    287             3881       2931
14  21217   6208   14982   3095             6707        602
15  24653   9465   12091    294             5058       2168
16  10253   1114    3821    397              964        412
17   1020   8816   12121    134             4508       1080
19  18601   6327   10099   2205             2767       3181
20   7780   2495    9464    669             2518        501
21  17546   4519    4602   1066             2259       2124
22   5567    871    2010   3383              375        569
26  16165   4230    7595    201             4003         57
27   9898    961    2861   3151              242        833
28  14276    803    3045    485              100        518
31  18815   3610   11107   1148             2134       2963
32   2612   4339    3133   2088              820        985
33  21632   1318    2886    266              918        405
34  29729   4786    7326   6130              361       1083
35   1502   1979    2262    425              483        395
36    688   5491   11091    833             4239        436
38  15168  10556   12477   1920             6506        714
42  19176   3065    5956   2033             2575       2802
43  10850   7555   14961    188             6899         46
45   9670   7027   10471    541             4618         65
47   3103  14069   21955   1668             6792       1452
49  11519   6152   10868    584             5121       1476
51   6269   1095    1980   3860              609       2162
52   3347   4051    6996    239             1538        301
54    491  10473   11532    744             5611        224
55  27329   1449    1947   2436              204       1333
56   5264   3683    5005   1057             2024       1130
58   5417   9933   10487     38             7572       1282
> str(wholesale)
'data.frame':	339 obs. of  6 variables:
 $ Fresh           : int  12669 7057 13265 9413 12126 7579 5963 6006 3366 13146 ...
 $ Milk            : int  9656 9810 1196 8259 3199 4956 3648 11093 5403 1124 ...
 $ Grocery         : int  7561 9568 4221 5126 6975 9426 6192 18881 12974 4523 ...
 $ Frozen          : int  214 1762 6404 666 480 1669 425 1159 4400 1420 ...
 $ Detergents_Paper: int  2674 3293 507 1795 3140 3321 1716 7425 5977 549 ...
 $ Delicassen      : int  1338 1776 1788 1451 545 2566 750 2098 1744 497 ...
```

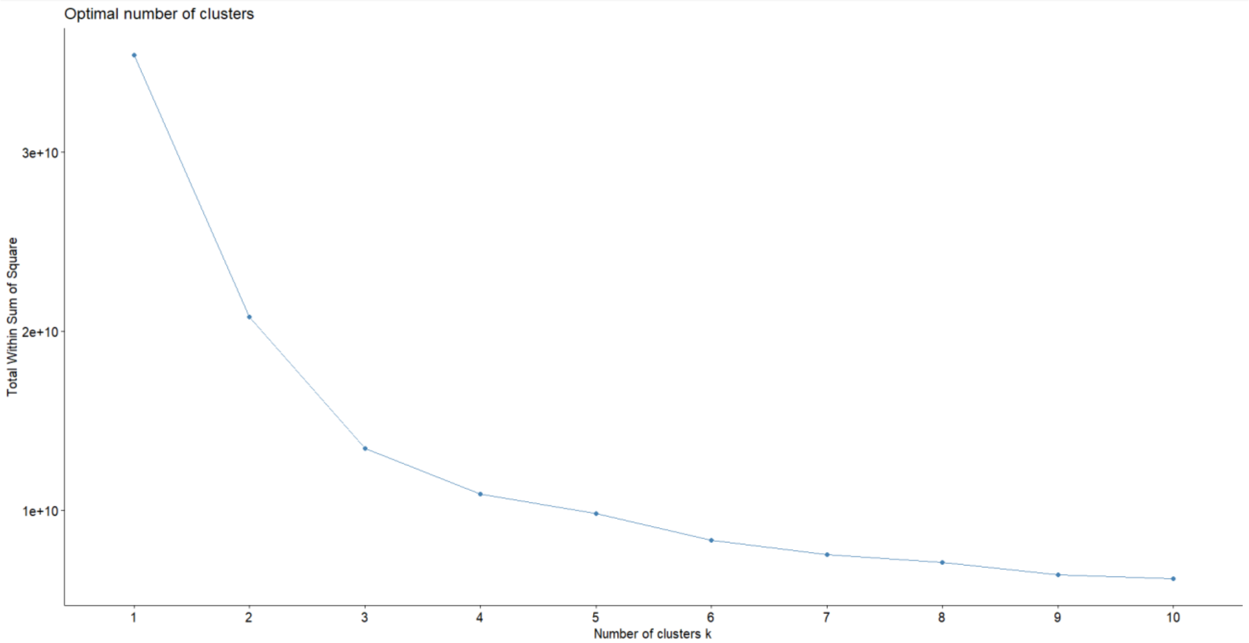
In the output after remove the outlier we found 339 instances from 440 instances.

Estimating the Optimal number of Cluster:

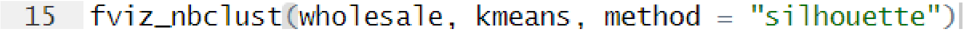


```
install.packages("factoextra")
library(factoextra)
fviz_nbclust(wholesale, kmeans, method = "wss")
```

Output:

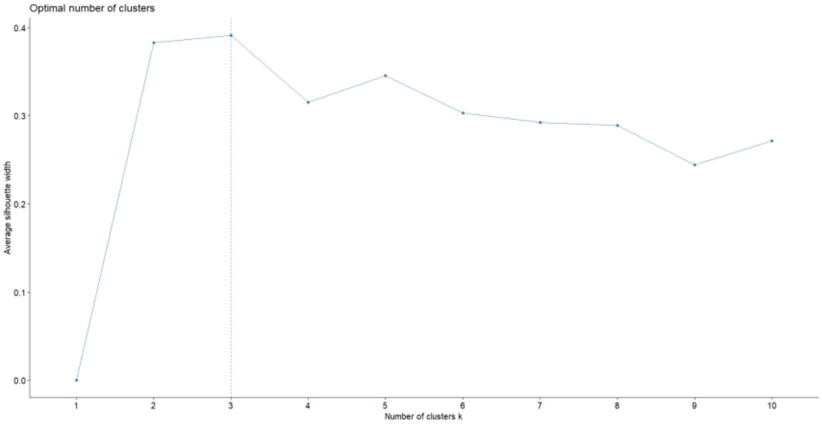


Here WSS method is used to determine the optimal number of clusters for the data. Here the plot shows the relation between the total WSS and the number of clusters. Here the bend is 5 to 4.

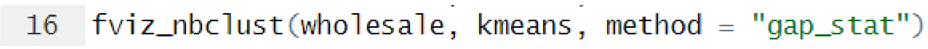


```
fviz_nbclust(wholesale, kmeans, method = "silhouette")
```

Output:

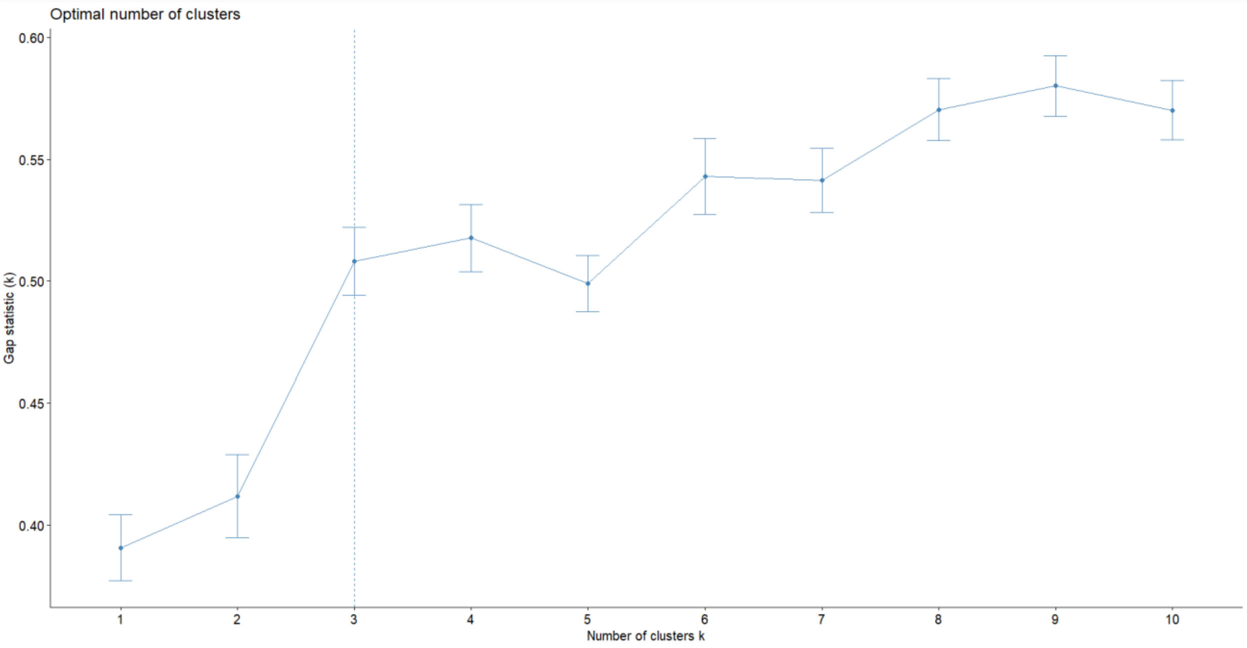


Here Silhouette method is a measure of the quality of a clustering solution that takes into account both how well each data point is assigned to its own cluster (cohesion) and how distinct the cluster is from other clusters (separation). Here the highest score is 3.



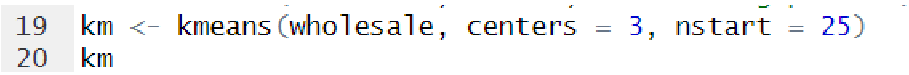
```
fviz_nbclust(wholesale, kmeans, method = "gap_stat")
```

Output:



Here the highest point is 9 and the 3 is the baseline of this plot.

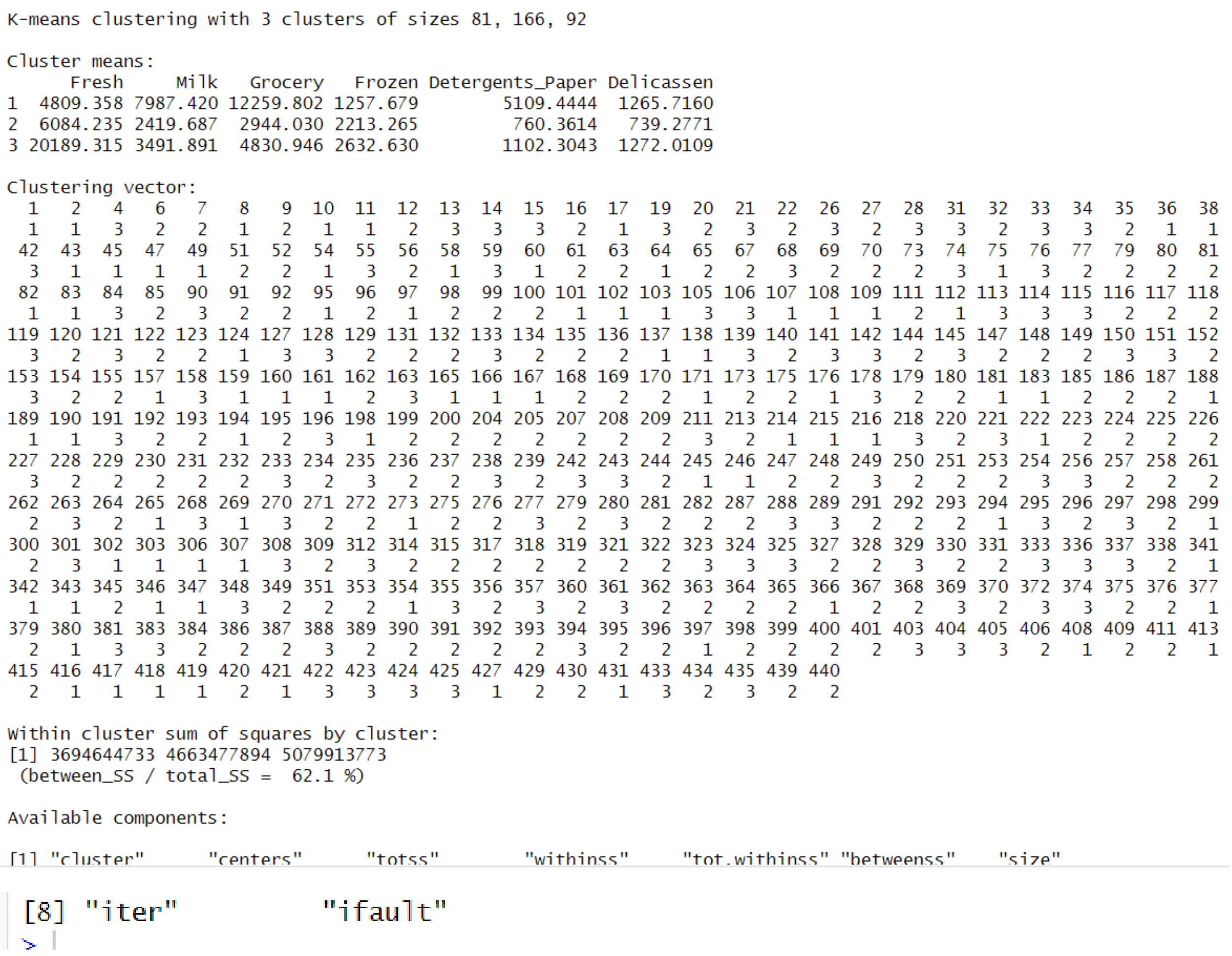
Perform K-means Clustering Algorithm on Dataset:



```
km <- kmeans(wholesale, centers = 3, nstart = 25)
km
```

Here the number of clusters set we consider is 3. The “nstart” parameter controls the number of times the clustering algorithm is run with different initial starting points.

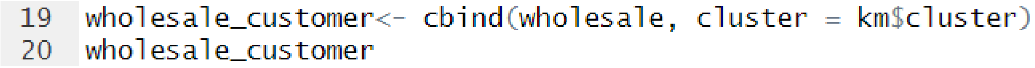
Output:



```
K-means clustering with 3 clusters of sizes 81, 166, 92

Cluster means:
      Fresh     Milk   Grocery   Frozen Detergents_Paper Delicassen
1  4809.358 7987.420 12259.802 1257.679        5109.4444  1265.7160
2  6084.235 2419.687  2944.030 2213.265         760.3614   739.2771
3 20189.315 3491.891  4830.946 2632.630        1102.3043  1272.0109

Clustering vector:
  1   2   4   6   7   8   9  10  11  12  13  14  15  16  17  19  20  21  22  26  27  28  31  32  33  34  35  36  38 
  1   1   3   2   2   1   2   1   1   2   3   3   3   2   1   3   2   3   2   3   2   3   3   2   3   3   2   1   1 
 42  43  45  47  49  51  52  54  55  56  58  59  60  61  63  64  65  67  68  69  70  73  74  75  76  77  79  80  81 
  3   1   1   1   1   2   2   1   3   2   1   3   1   2   2   1   2   2   3   2   2   2   3   1   3   2   2   2   2 
 82  83  84  85  90  91  92  95  96  97  98  99 100 101 102 103 105 106 107 108 109 111 112 113 114 115 116 117 118 
  1   1   3   2   3   2   2   1   2   1   2   2   2   1   1   1   3   3   1   1   1   2   1   3   3   3   2   2   2 
119 120 121 122 123 124 127 128 129 131 132 133 134 135 136 137 138 139 140 141 142 144 145 147 148 149 150 151 152 
  3   2   3   2   2   1   3   3   2   2   2   3   2   2   2   1   1   3   2   3   3   2   3   2   2   2   3   3   2 
153 154 155 157 158 159 160 161 162 163 165 166 167 168 169 170 171 173 175 176 178 179 180 181 183 185 186 187 188 
  3   2   2   1   3   1   1   1   2   3   1   1   1   2   2   2   1   2   2   1   3   2   2   1   1   2   2   2   1 
189 190 191 192 193 194 195 196 198 199 200 204 205 207 208 209 211 213 214 215 216 218 220 221 222 223 224 225 226 
  1   1   3   2   2   1   2   3   1   2   2   2   2   2   2   2   3   2   1   1   1   3   2   3   1   2   2   2   2 
227 228 229 230 231 232 233 234 235 236 237 238 239 242 243 244 245 246 247 248 249 250 251 253 254 256 257 258 261 
  3   2   2   2   2   2   3   2   3   2   2   3   2   3   3   2   1   1   2   2   3   2   2   2   3   3   2   2   2 
262 263 264 265 268 269 270 271 272 273 275 276 277 279 280 281 282 287 288 289 291 292 293 294 295 296 297 298 299 
  2   3   2   1   3   1   3   2   2   1   2   2   3   2   3   2   2   2   3   3   2   2   2   1   3   2   3   2   1 
300 301 302 303 306 307 308 309 312 314 315 317 318 319 321 322 323 324 325 327 328 329 330 331 333 336 337 338 341 
  2   3   1   1   1   1   3   2   3   2   2   2   2   2   2   2   3   3   3   2   2   3   2   2   3   3   3   2   1 
342 343 345 346 347 348 349 351 353 354 355 356 357 360 361 362 363 364 365 366 367 368 369 370 372 374 375 376 377 
  1   1   2   1   1   3   2   2   2   1   3   2   3   2   3   2   2   2   2   1   2   2   3   2   3   3   2   2   1 
379 380 381 383 384 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 403 404 405 406 408 409 411 413 
  2   1   3   3   2   2   2   3   2   2   2   2   2   3   2   2   1   2   2   2   2   3   3   3   2   1   2   2   1 
415 416 417 418 419 420 421 422 423 424 425 427 429 430 431 433 434 435 439 440 
  2   1   1   1   1   2   1   3   3   3   3   1   2   2   1   3   2   3   2   2 

Within cluster sum of squares by cluster:
[1] 3694644733 4663477894 5079913773
 (between_SS / total_SS =  62.1 %)

Available components:

[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"
[8] "iter"         "ifault"
> 
```

In the output we found the 3 clusters size which are 81,166,92. Here we also found the clusters mean values and sum of squares

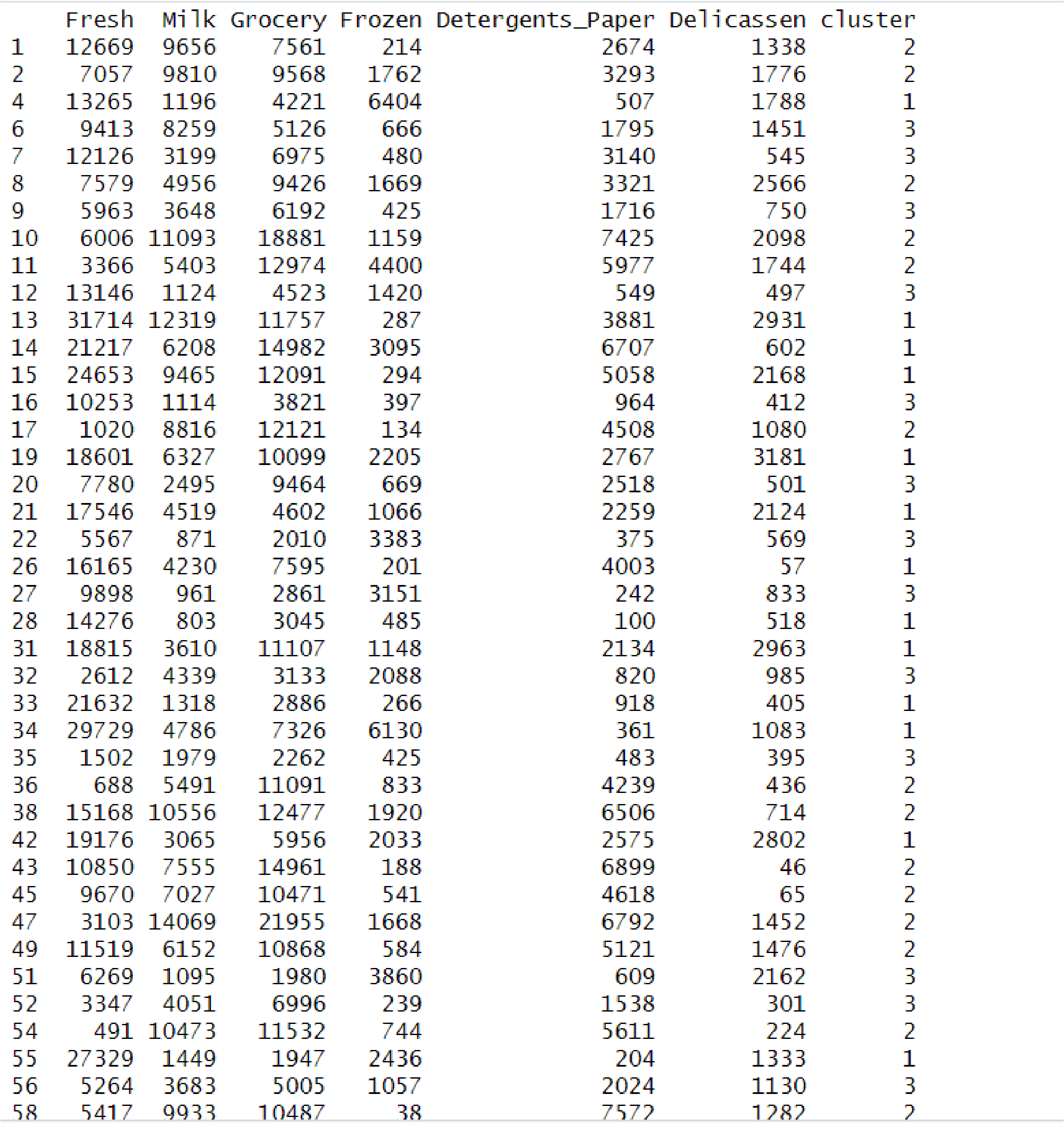
Visualize the output of K-means Clustering Algorithm:



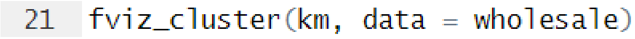
```
wholesale_customer<- cbind(wholesale, cluster = km$cluster)
wholesale_customer
```

This line of code is adding a new column to the "wholesale" dataset called "cluster". The values in this new column are determined by the cluster assignments generated by the k-means clustering algorithm stored in the km object. The cbind() function is used to combine the "wholesale" dataset with a new column containing the cluster assignments.

Output:

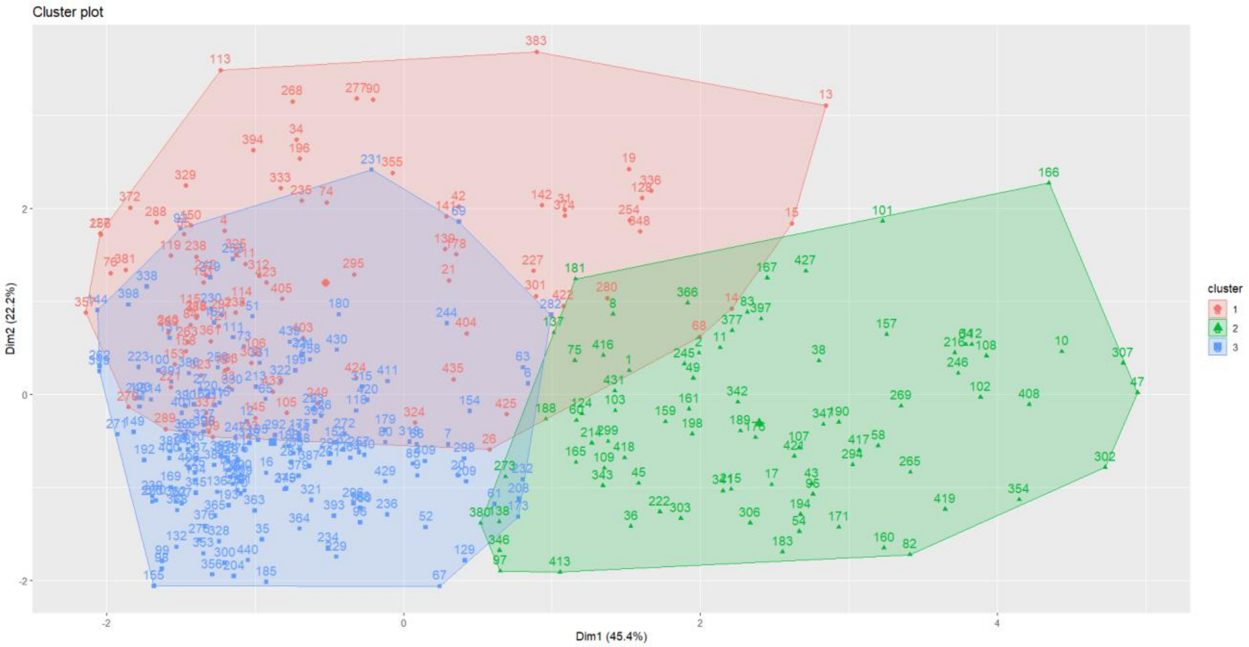


```
   Fresh  Milk Grocery Frozen Detergents_Paper Delicassen cluster
1  12669  9656    7561    214             2674       1338       2
2   7057  9810    9568   1762             3293       1776       2
4  13265  1196    4221   6404              507       1788       1
6   9413  8259    5126    666             1795       1451       3
7  12126  3199    6975    480             3140        545       3
8   7579  4956    9426   1669             3321       2566       2
9   5963  3648    6192    425             1716        750       3
10  6006 11093   18881   1159             7425       2098       2
11  3366  5403   12974   4400             5977       1744       2
12 13146  1124    4523   1420              549        497       3
13 31714 12319   11757    287             3881       2931       1
14 21217  6208   14982   3095             6707        602       1
15 24653  9465   12091    294             5058       2168       1
16 10253  1114    3821    397              964        412       3
17  1020  8816   12121    134             4508       1080       2
19 18601  6327   10099   2205             2767       3181       1
20  7780  2495    9464    669             2518        501       3
21 17546  4519    4602   1066             2259       2124       1
22  5567   871    2010   3383              375        569       3
26 16165  4230    7595    201             4003         57       1
27  9898   961    2861   3151              242        833       3
28 14276   803    3045    485              100        518       1
31 18815  3610   11107   1148             2134       2963       1
32  2612  4339    3133   2088              820        985       3
33 21632  1318    2886    266              918        405       1
34 29729  4786    7326   6130              361       1083       1
35  1502  1979    2262    425              483        395       3
36   688  5491   11091    833             4239        436       2
38 15168 10556   12477   1920             6506        714       2
42 19176  3065    5956   2033             2575       2802       1
43 10850  7555   14961    188             6899         46       2
45  9670  7027   10471    541             4618         65       2
47  3103 14069   21955   1668             6792       1452       2
49 11519  6152   10868    584             5121       1476       2
51  6269  1095    1980   3860              609       2162       3
52  3347  4051    6996    239             1538        301       3
54   491 10473   11532    744             5611        224       2
55 27329  1449    1947   2436              204       1333       1
56  5264  3683    5005   1057             2024       1130       3
58  5417  9933   10487     38             7572       1282       2
```



```
fviz_cluster(km, data = wholesale)
```

Output:



The fviz_cluster() function is used to create a visualization of the clusters generated by a clustering algorithm. Here this plots I consider 3 clusters. Here 3 clusters are overlapping with each other because here may be different clusters in terms of the types and quantities of products purchased by the customers. Here "Dim1 (45.4%)" and "Dim2 (22.2%)" in the plot means that Dim1 explains 45.4% of the total variance in the data, and Dim2 explains 22.2% of the total variance.