Assignment 5

Due at 11:59pm on November 26.

Molly Fisch-Friedman & Brittany Owen

You may work in pairs or individually for this assignment. Make sure you join a group in Canvas if you are working in pairs. Turn in this assignment as an HTML or PDF file to ELMS. Make sure to include the R Markdown or Quarto file that was used to generate it. Include the GitHub link for the repository containing these files.

GitHub link: https://github.com/mollyfischfriedman/FischFriedman-Owen-a5

```
library(censusapi)
```

```
Attaching package: 'censusapi'

The following object is masked from 'package:methods':

    getFunction
```

```
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.4.2     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
```

```
-- Conflicts ---------------------------------------- tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becor
```

```
library(magrittr)
```

```
Attaching package: 'magrittr'

The following object is masked from 'package:purrr':

    set_names

The following object is masked from 'package:tidyr':

    extract
```

```
library(factoextra)
```

```
Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(cluster)
library(factoextra)
```

Exploring ACS Data

In this notebook, we use the Census API to gather data from the American Community Survey (ACS). This requires an access key, which can be obtained here:

https://api.census.gov/data/key_signup.html

```
cs_key <- read_file("~/Downloads/census-key.txt")
acs_il_c <- getCensus(name = "acs/acs5",
                      vintage = 2016,
                      vars = c("NAME", "B01003_001E", "B19013_001E", "B19301_001E"),
                      region = "county:*",
                      regionin = "state:17",
                      key = cs_key) %>%
            rename(pop = B01003_001E,
```

```
                    hh_income = B19013_001E,
                    income = B19301_001E)
head(acs_il_c)
```

```
  state county                        NAME    pop hh_income income
1    17    067   Hancock County, Illinois  18633     50077  25647
2    17    063    Grundy County, Illinois  50338     67162  30232
3    17    091  Kankakee County, Illinois 111493     54697  25111
4    17    043    DuPage County, Illinois 930514     81521  40547
5    17    003 Alexander County, Illinois   7051     29071  16067
6    17    129    Menard County, Illinois  12576     60420  31323
```

Pull map data for Illinois into a data frame.

```
il_map <- map_data("county", region = "illinois")
head(il_map)
```

```
       long      lat group order   region subregion
1 -91.49563 40.21018     1     1 illinois     adams
2 -90.91121 40.19299     1     2 illinois     adams
3 -90.91121 40.19299     1     3 illinois     adams
4 -90.91121 40.10704     1     4 illinois     adams
5 -90.91121 39.83775     1     5 illinois     adams
6 -90.91694 39.75754     1     6 illinois     adams
```

Join the ACS data with the map data. Not that `il_map` has a column `subregion` which includes county names. We need a corresponding variable in the ACS data to join both data sets. This needs some transformations, among which the function `tolower()` might be useful. Call the joined data `acs_map`.

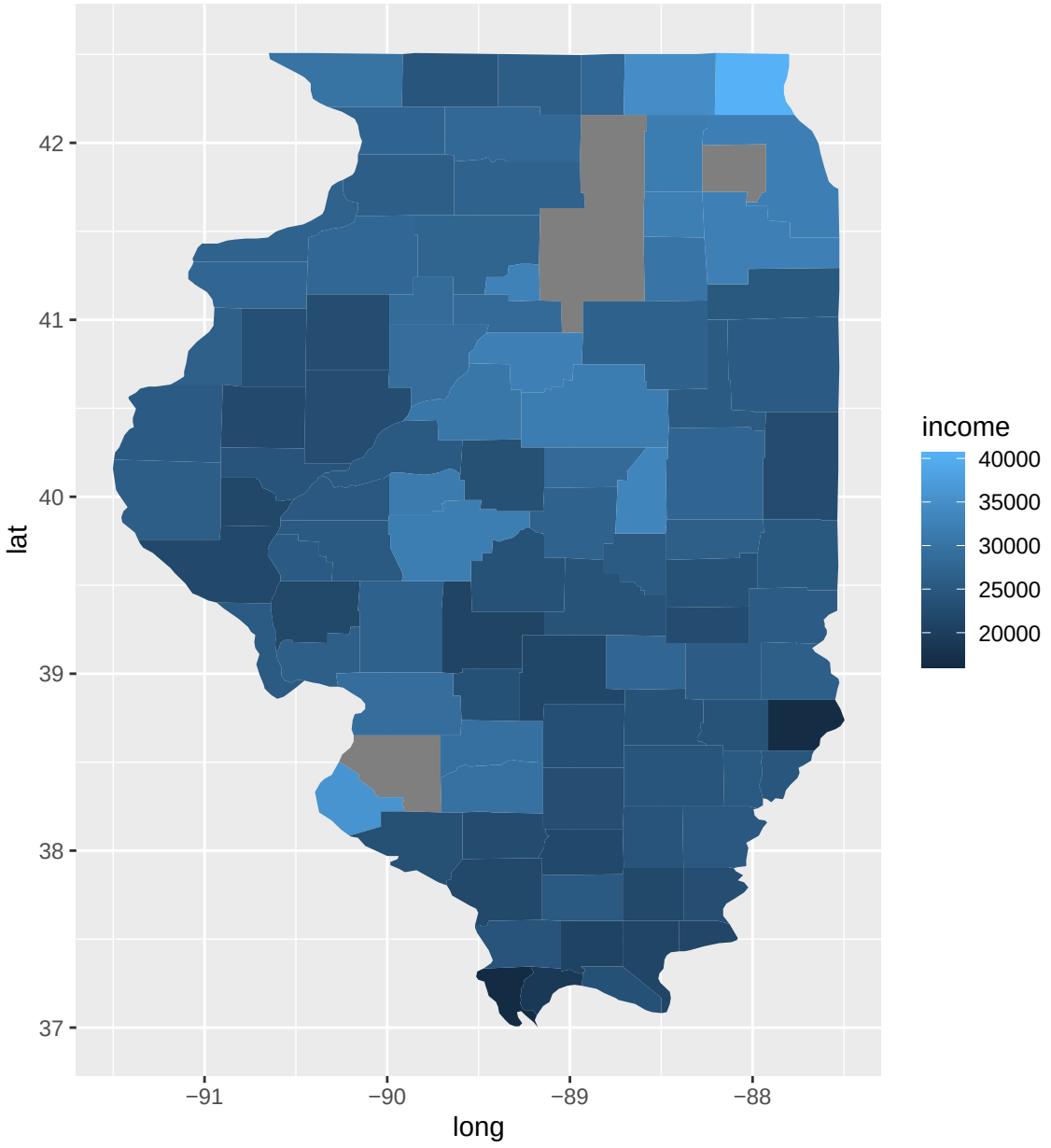
```
acs_il_c$subregion <- tolower(gsub("\\sCounty.+","",acs_il_c$NAME))
acs_map <- left_join(il_map, acs_il_c, by="subregion")
head(acs_map)
```

```
       long      lat group order   region subregion state county
1 -91.49563 40.21018     1     1 illinois     adams    17    001
2 -90.91121 40.19299     1     2 illinois     adams    17    001
3 -90.91121 40.19299     1     3 illinois     adams    17    001
4 -90.91121 40.10704     1     4 illinois     adams    17    001
5 -90.91121 39.83775     1     5 illinois     adams    17    001
```

```
6 -90.91694 39.75754     1     6 illinois     adams    17    001
                    NAME   pop hh_income income
1 Adams County, Illinois 66949     48065  26053
2 Adams County, Illinois 66949     48065  26053
3 Adams County, Illinois 66949     48065  26053
4 Adams County, Illinois 66949     48065  26053
5 Adams County, Illinois 66949     48065  26053
6 Adams County, Illinois 66949     48065  26053
```

After you do this, plot a map of Illinois with Counties colored by per capita income.

```
ggplot(acs_map) + geom_polygon(aes(x = long, y = lat, group = group, fill = income))
```



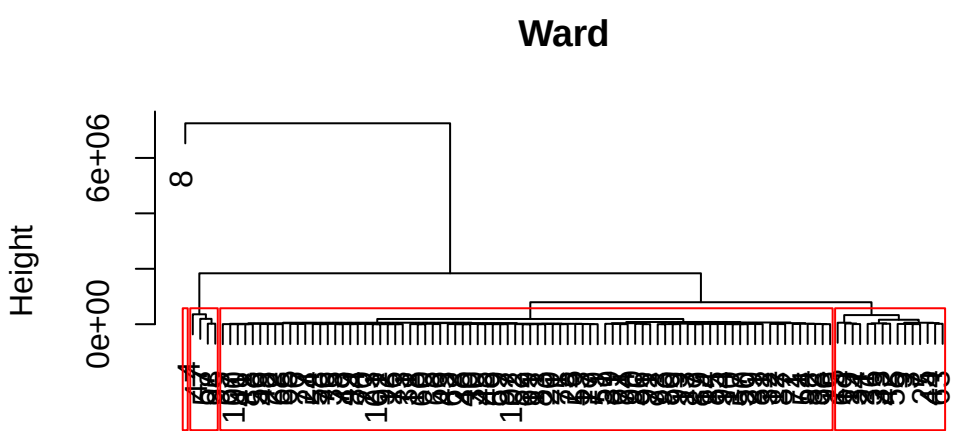
Hierarchical Clustering

We want to find clusters of counties that are similar in their population, average household income and per capita income. First, clean the data so that you have the appropriate variables to use for clustering. Next, create the distance matrix of the cleaned data. This distance matrix can be used to cluster counties, e.g. using the ward method.

```
hclust <- acs_il_c %>% select(pop, hh_income, income)
hclust_dist <- dist(hclust)
hc_ward <- hclust(hclust_dist, method = "ward.D2")
```

Plot the dendrogram to find a reasonable number of clusters. Draw boxes around the clusters of your cluster solution.

```
plot(hc_ward, main = "Ward", xlab = "", sub = "")
rect.hclust(hc_ward,
            k = 4,
            border = "red")
```



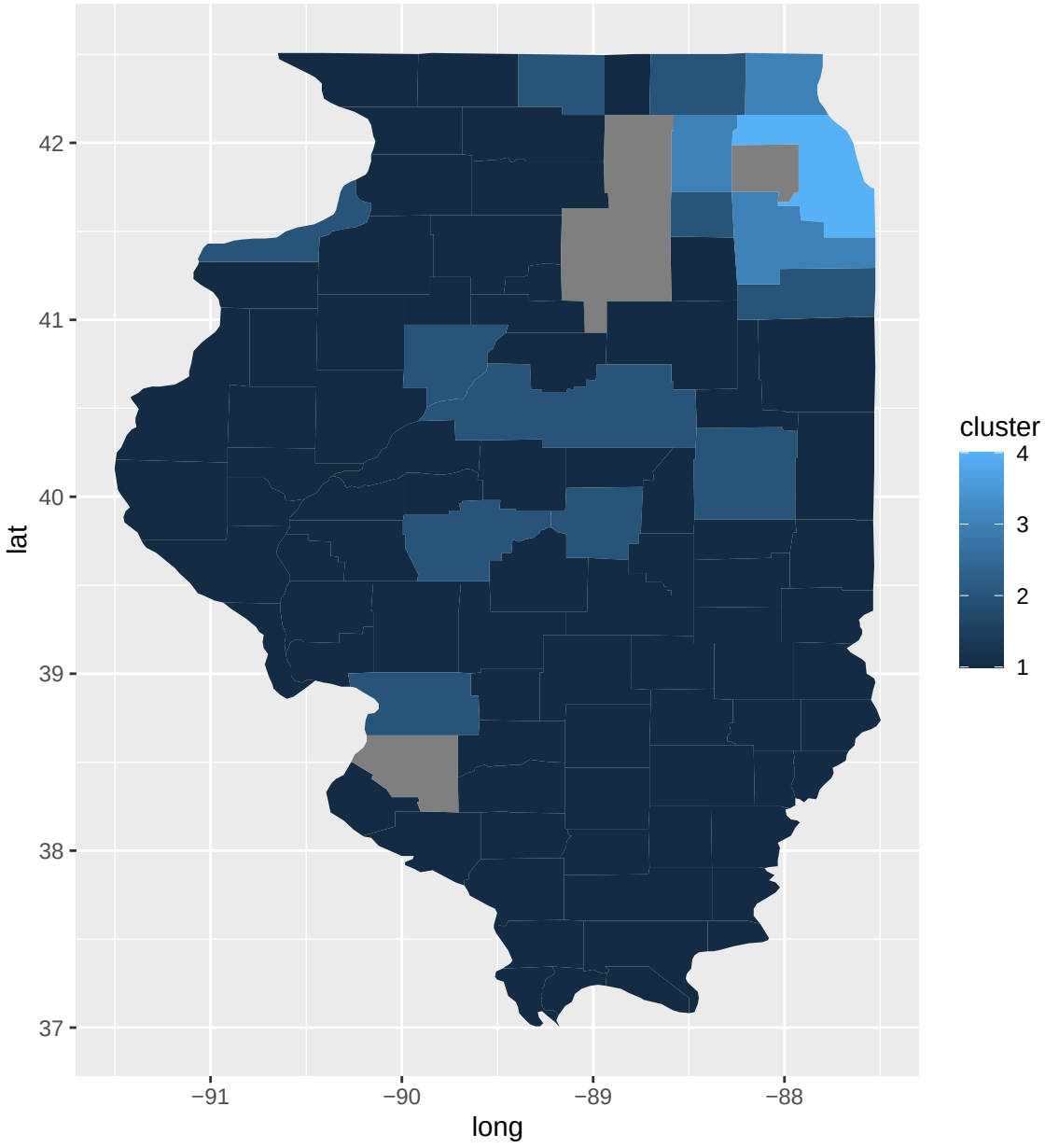
Visualize the county clusters on a map. For this task, create a new `acs_map` object that now also includes cluster membership as a new column. This column should be called `cluster`.

```
cluster <- cutree(hc_ward, 4)
acs_il_c$cluster <- cluster
acs_map <- left_join(il_map, acs_il_c, by="subregion")
head(acs_map)
```

```
       long      lat group order   region subregion state county
1 -91.49563 40.21018     1     1 illinois     adams    17    001
```

```
2 -90.91121 40.19299     1     2 illinois     adams    17    001
3 -90.91121 40.19299     1     3 illinois     adams    17    001
4 -90.91121 40.10704     1     4 illinois     adams    17    001
5 -90.91121 39.83775     1     5 illinois     adams    17    001
6 -90.91694 39.75754     1     6 illinois     adams    17    001
                    NAME   pop hh_income income cluster
1 Adams County, Illinois 66949     48065  26053       1
2 Adams County, Illinois 66949     48065  26053       1
3 Adams County, Illinois 66949     48065  26053       1
4 Adams County, Illinois 66949     48065  26053       1
5 Adams County, Illinois 66949     48065  26053       1
6 Adams County, Illinois 66949     48065  26053       1
```

```
ggplot(acs_map) + geom_polygon(aes(x = long, y = lat, group = group, fill = cluster))
```



Census Tracts

For the next section we need ACS data on a census tract level. We use the same variables as before.

```
acs_il_t <- getCensus(name = "acs/acs5",
                      vintage = 2016,
                      vars = c("NAME", "B01003_001E", "B19013_001E", "B19301_001E"),
                      region = "tract:*",
                      regionin = "state:17",
                      key = cs_key) %>%
             rename(pop = B01003_001E,
                    hh_income = B19013_001E,
                    income = B19301_001E)

head(acs_il_t)
```

```
  state county  tract                                          NAME  pop
1    17    031 806002 Census Tract 8060.02, Cook County, Illinois 7304
2    17    031 806003 Census Tract 8060.03, Cook County, Illinois 7577
3    17    031 806400    Census Tract 8064, Cook County, Illinois 2684
4    17    031 806501 Census Tract 8065.01, Cook County, Illinois 2590
5    17    031 750600    Census Tract 7506, Cook County, Illinois 3594
6    17    031 310200    Census Tract 3102, Cook County, Illinois 1521
  hh_income income
1     56975  23750
2     53769  25016
3     62750  30154
4     53583  20282
5     40125  18347
6     63250  31403
```

```
#saving Rda files
save(acs_il_c, file="~/Downloads/acs_il_c.Rda")
save(acs_il_t, file="~/Downloads/acs_il_t.Rda")
save(acs_map, file="~/Downloads/acs_map.Rda")
```

k-Means

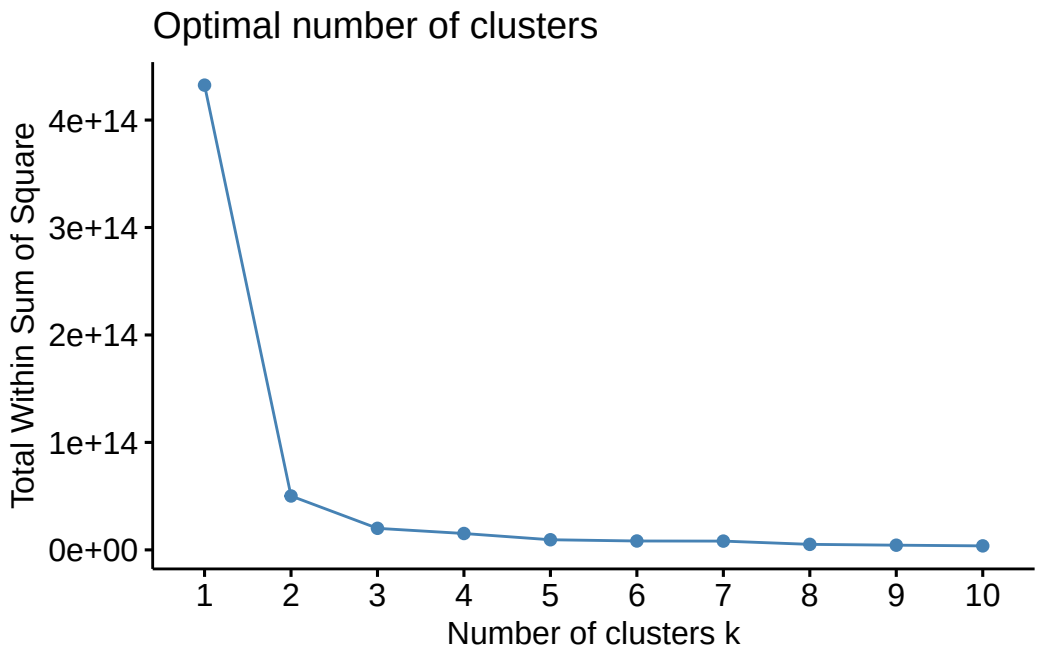
As before, clean our data for clustering census tracts based on population, average household income and per capita income.

```
tclust <- acs_il_t %>% select(pop, hh_income, income, tract, county, state)
tclust_dist <- dist(tclust)

tclust[tclust == -666666666] <- NA
```

Since we want to use K Means in this section, we start by determining the optimal number of K that results in Clusters with low within but high between variation. Plot within cluster sums of squares for a range of K (e.g. up to 20).

```
fviz_nbclust(na.omit(tclust), kmeans, method = "wss")
```



Run `kmeans()` for the optimal number of clusters based on the plot above.

```
km_1 <- kmeans(na.omit(tclust), 2, nstart = 20)
km_1
```

```
K-means clustering with 2 clusters of sizes 1918, 1191

Cluster means:
       pop hh_income   income    tract   county state
1 4341.583  66127.53 31116.94 826445.3 72.18457    17
2 3789.817  53196.33 29850.75 105177.8 87.00336    17

Clustering vector:
   1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
   1    1    1    1    1    2    1    1    1    1    1    1    1    1    1    1
  17   18   19   20   21   22   23   24   25   26   27   28   29   30   31   32
```

1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
1	1	1	1	1	2	2	2	2	1	1	1	1	1	1	1
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112
2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1
113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128
1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2
129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144
2	2	1	1	1	2	2	1	1	1	1	1	1	1	1	1
145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
2	2	2	2	2	2	2	2	1	1	2	2	2	2	2	2
161	162	163	164	165	166	167	168	169	170	171	172	173	175	176	177
2	2	2	2	2	2	2	2	2	1	2	2	2	2	2	2
178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193
2	2	2	2	2	2	2	2	1	2	2	2	1	1	1	2
194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209
1	1	1	1	1	2	1	1	1	1	1	2	2	2	1	1
210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225
1	1	1	1	2	2	2	2	2	2	2	1	1	1	1	1
226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241
1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2
242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273
2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1
274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305
1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2
306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337
1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2
338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369
1	1	1	1	1	1	1	1	2	2	2	2	1	1	1	1

370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385
1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	1
386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401
1	1	2	1	1	1	2	2	2	2	2	2	2	2	2	2
402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417
2	2	2	1	1	2	2	2	2	2	1	1	1	2	2	2
418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433
2	2	1	1	1	1	1	2	2	2	2	2	1	1	1	2
434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449
2	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2
450	451	452	453	454	455	456	457	458	459	460	461	462	464	465	466
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482
2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1
483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530
1	1	1	1	1	1	1	1	1	1	2	2	1	1	2	2
531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546
2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1
547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
595	596	597	598	599	600	601	602	603	604	605	606	607	608	609	610
1	1	1	1	1	1	2	2	2	2	2	1	1	1	1	2
611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642
2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1
643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658
1	1	1	1	1	1	1	1	1	1	1	2	2	1	1	2
659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690
1	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1
691	692	693	694	695	696	697	698	699	700	701	702	703	704	705	706
2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1
707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738
1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
739	741	742	743	744	746	747	748	749	750	751	752	753	754	755	756
1	1	2	2	2	1	1	1	1	1	1	1	1	1	1	1
757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
773	774	775	776	777	778	779	780	782	783	784	785	786	787	788	789
1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2
790	791	792	793	794	795	796	797	798	799	800	801	802	803	804	805
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
806	807	808	809	810	811	812	813	815	816	817	818	819	820	821	822
2	2	2	2	2	2	2	1	2	1	1	1	2	1	1	1
823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838
2	2	2	1	1	1	1	1	1	2	2	1	1	1	1	1
839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886
1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2
887	888	889	890	891	892	893	894	895	896	897	898	899	900	901	902
2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1
903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918
1	1	1	1	1	1	1	1	1	1	1	1	2	2	1	1
919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934
1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	1
935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966
1	1	1	1	1	1	1	1	1	1	2	2	2	1	1	1
967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
999	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012	1013	1014
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1031	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1047	1048	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062
1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2

1063	1064	1065	1066	1067	1068	1069	1070	1071	1072	1073	1074	1075	1076	1077	1078
2	2	2	2	2	2	2	2	1	1	1	1	1	1	1	1
1079	1080	1081	1082	1083	1084	1085	1086	1087	1088	1089	1090	1091	1092	1093	1094
1	2	2	2	2	2	2	2	2	1	1	1	1	1	1	1
1095	1096	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110
1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2
1111	1112	1113	1114	1115	1116	1117	1118	1119	1120	1121	1122	1123	1124	1125	1126
2	2	2	1	1	1	1	2	2	2	2	2	2	2	2	2
1127	1128	1129	1130	1131	1132	1133	1134	1135	1136	1137	1138	1139	1140	1141	1142
2	2	2	2	2	2	1	1	1	1	1	1	1	1	1	1
1143	1144	1145	1146	1147	1148	1149	1150	1151	1152	1153	1154	1155	1156	1157	1158
1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2
1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170	1171	1172	1173	1174
2	2	1	1	1	1	1	2	2	2	2	2	1	2	2	2
1175	1176	1177	1178	1179	1180	1181	1182	1183	1184	1185	1186	1187	1188	1189	1190
2	2	2	1	1	1	1	1	1	1	1	1	1	2	2	2
1191	1192	1193	1194	1195	1196	1197	1198	1199	1200	1201	1202	1203	1204	1205	1206
2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1
1207	1208	1209	1210	1211	1212	1213	1214	1215	1216	1217	1218	1219	1220	1221	1222
1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1223	1224	1225	1226	1227	1228	1229	1230	1231	1232	1233	1234	1235	1236	1237	1238
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1239	1240	1241	1242	1243	1244	1245	1246	1247	1248	1249	1250	1251	1252	1253	1254
1	1	1	1	1	1	1	2	2	2	2	1	1	1	1	1
1255	1256	1257	1258	1259	1260	1261	1262	1263	1264	1265	1266	1267	1268	1269	1270
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1271	1272	1273	1274	1275	1276	1277	1278	1279	1280	1281	1282	1283	1284	1285	1286
1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2
1287	1288	1289	1290	1291	1292	1293	1294	1295	1296	1297	1298	1299	1300	1301	1302
2	2	1	1	1	2	2	2	2	2	2	2	2	2	2	2
1303	1304	1305	1306	1307	1308	1309	1310	1311	1312	1313	1314	1315	1316	1317	1318
2	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2
1319	1320	1321	1322	1323	1324	1325	1326	1327	1328	1329	1330	1331	1332	1333	1334
2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1335	1336	1337	1338	1339	1340	1341	1342	1343	1344	1345	1346	1347	1348	1349	1350
1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2
1351	1352	1353	1354	1355	1356	1357	1358	1359	1360	1361	1362	1363	1364	1365	1366
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1367	1368	1369	1370	1371	1372	1373	1374	1375	1376	1377	1378	1379	1380	1381	1382
1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1
1383	1384	1385	1386	1387	1388	1389	1390	1391	1392	1393	1394	1395	1396	1397	1398
1	1	2	2	2	1	1	1	1	1	1	1	1	1	1	1
1399	1400	1401	1402	1403	1404	1405	1406	1407	1408	1409	1410	1411	1412	1413	1414

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1415	1416	1417	1418	1419	1420	1421	1422	1423	1424	1425	1426	1427	1428	1429	1430
1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1
1431	1432	1433	1434	1435	1436	1437	1438	1439	1440	1441	1442	1443	1444	1445	1446
1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2
1447	1448	1449	1450	1451	1452	1453	1454	1455	1456	1457	1458	1459	1460	1461	1462
2	2	2	2	2	2	2	2	2	2	1	1	1	1	1	1
1463	1464	1465	1466	1467	1468	1469	1470	1471	1472	1473	1474	1475	1476	1477	1478
1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
1479	1480	1481	1482	1483	1484	1485	1486	1487	1488	1489	1490	1491	1492	1493	1494
2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1
1495	1496	1497	1498	1499	1500	1501	1502	1503	1504	1505	1506	1507	1508	1509	1510
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1511	1512	1513	1514	1515	1516	1517	1518	1519	1520	1521	1522	1523	1524	1525	1526
1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
1527	1528	1529	1530	1531	1532	1533	1534	1535	1536	1537	1538	1539	1540	1541	1542
2	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2
1543	1544	1545	1546	1547	1548	1549	1550	1551	1552	1553	1554	1555	1556	1557	1558
2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1
1559	1560	1561	1562	1563	1564	1565	1566	1569	1570	1571	1573	1574	1575	1576	1577
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1578	1579	1580	1581	1582	1583	1584	1585	1586	1587	1588	1589	1590	1591	1592	1593
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1594	1595	1596	1597	1598	1599	1600	1601	1602	1603	1604	1605	1606	1607	1608	1609
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1610	1611	1612	1613	1614	1615	1616	1617	1618	1619	1620	1621	1622	1623	1624	1625
1	1	1	1	1	2	1	1	1	1	1	1	2	2	2	2
1626	1627	1628	1629	1630	1631	1632	1633	1634	1635	1636	1637	1638	1639	1640	1641
2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1
1642	1643	1644	1645	1646	1647	1648	1649	1650	1651	1652	1653	1654	1655	1656	1657
1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2
1658	1659	1660	1661	1662	1663	1664	1665	1666	1667	1668	1669	1670	1671	1672	1673
2	2	2	2	2	2	2	2	2	2	1	1	2	2	2	1
1674	1675	1676	1677	1678	1679	1680	1681	1682	1683	1684	1685	1686	1687	1688	1689
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1690	1691	1692	1693	1694	1695	1696	1697	1698	1699	1700	1701	1702	1703	1704	1705
1	1	1	1	1	1	2	2	2	2	2	2	1	1	2	2
1706	1707	1708	1709	1710	1711	1712	1713	1714	1715	1716	1717	1718	1719	1720	1721
1	1	1	1	1	1	2	2	2	1	1	1	1	1	1	1
1722	1723	1724	1725	1726	1727	1728	1729	1730	1731	1732	1736	1737	1738	1739	1740
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1741	1742	1743	1744	1745	1746	1747	1748	1749	1750	1751	1752	1753	1754	1755	1756
1	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1

1757	1758	1759	1760	1761	1762	1763	1764	1765	1766	1767	1768	1769	1770	1771	1772
1	2	2	2	2	2	1	1	1	1	1	1	1	2	1	2
1773	1774	1775	1776	1777	1778	1779	1780	1781	1782	1783	1784	1785	1786	1787	1788
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1789	1790	1791	1792	1793	1794	1795	1796	1797	1798	1799	1800	1801	1802	1803	1804
1	1	1	2	2	2	2	2	2	2	2	2	2	1	1	1
1805	1806	1807	1808	1809	1810	1811	1812	1813	1814	1815	1816	1817	1818	1819	1820
1	1	1	1	1	1	1	2	2	1	1	2	2	2	2	2
1821	1822	1823	1824	1825	1826	1827	1828	1829	1830	1831	1832	1833	1834	1835	1836
2	2	2	2	2	2	2	2	2	1	1	1	1	1	2	2
1837	1838	1839	1840	1841	1842	1843	1844	1845	1846	1847	1848	1849	1850	1851	1852
2	2	2	2	2	2	2	1	1	1	1	2	2	2	2	1
1853	1854	1855	1856	1857	1858	1859	1860	1861	1862	1863	1864	1865	1866	1867	1868
1	1	1	1	1	1	1	2	2	2	2	1	1	1	1	1
1869	1870	1871	1872	1873	1874	1875	1876	1877	1878	1879	1880	1881	1882	1883	1884
2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1
1885	1886	1887	1888	1889	1890	1891	1892	1893	1894	1895	1896	1897	1898	1899	1900
1	1	2	2	2	2	2	2	2	2	2	2	1	1	1	1
1901	1902	1903	1904	1905	1906	1907	1908	1909	1910	1911	1912	1913	1914	1915	1916
1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2
1917	1918	1919	1920	1921	1922	1923	1924	1925	1926	1927	1928	1929	1930	1931	1932
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
1933	1934	1935	1936	1937	1938	1939	1940	1941	1942	1943	1944	1945	1946	1947	1948
2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1
1949	1950	1951	1952	1953	1954	1955	1956	1957	1958	1959	1960	1961	1962	1963	1964
1	1	1	1	1	1	2	1	1	1	1	1	1	1	1	1
1965	1966	1967	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980
1	1	1	1	1	2	2	1	1	1	1	1	1	1	1	1
1981	1982	1983	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1997	1998	1999	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
2029	2030	2031	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2045	2046	2047	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1
2061	2062	2063	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076
1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2
2077	2078	2079	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092
2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
2093	2094	2095	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108

1	2	2	1	2	2	2	2	1	1	1	1	1	1	1	1
2109	2110	2111	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2125	2126	2127	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2141	2142	2143	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156
1	1	1	1	1	1	1	2	2	1	2	2	1	2	2	2
2157	2158	2159	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2173	2174	2175	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188
1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2
2189	2190	2191	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2205	2206	2207	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220
2	2	2	2	2	2	2	2	2	2	1	1	1	2	2	2
2221	2222	2223	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2237	2238	2239	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252
2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1
2253	2254	2255	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268
1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2
2269	2270	2271	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284
2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1
2285	2286	2287	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2301	2302	2303	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332
2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1
2333	2334	2335	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348
1	1	1	1	1	1	1	1	1	2	2	1	1	1	2	2
2349	2350	2351	2352	2353	2354	2355	2357	2358	2359	2360	2361	2362	2363	2364	2365
2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1
2366	2367	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381
1	1	1	1	1	1	1	1	2	2	1	2	1	1	1	1
2382	2383	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2398	2399	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413
1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	1
2414	2415	2416	2417	2418	2419	2420	2421	2422	2423	2424	2425	2426	2427	2428	2429
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2430	2431	2432	2433	2434	2435	2436	2437	2438	2439	2440	2441	2442	2443	2444	2445
2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	2

2446	2447	2448	2449	2450	2451	2452	2453	2454	2455	2456	2457	2458	2459	2460	2461
2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2462	2463	2464	2465	2466	2467	2468	2469	2470	2471	2472	2473	2474	2475	2476	2477
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2478	2479	2480	2481	2482	2483	2484	2485	2486	2487	2488	2489	2490	2491	2492	2493
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
2494	2495	2496	2497	2498	2499	2500	2501	2502	2503	2504	2505	2506	2507	2508	2509
2	2	2	2	2	2	2	2	2	2	1	2	2	2	2	2
2510	2511	2512	2513	2514	2515	2516	2517	2518	2519	2520	2521	2522	2523	2524	2525
2	2	1	1	2	2	2	2	2	2	2	2	2	2	2	2
2526	2527	2528	2529	2530	2531	2532	2533	2534	2535	2536	2537	2538	2539	2540	2541
2	2	2	1	1	1	1	1	1	1	1	2	2	2	2	2
2542	2543	2544	2545	2546	2547	2548	2549	2550	2551	2552	2553	2554	2555	2556	2557
2	2	2	1	2	2	2	2	2	1	1	1	1	1	1	1
2558	2559	2560	2561	2562	2563	2564	2565	2566	2567	2568	2569	2570	2571	2572	2573
1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2
2574	2575	2576	2577	2578	2579	2580	2581	2582	2583	2584	2585	2586	2587	2588	2589
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2590	2591	2592	2593	2594	2595	2596	2597	2598	2599	2600	2601	2602	2603	2604	2605
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2606	2607	2608	2609	2610	2611	2612	2613	2614	2615	2616	2617	2618	2619	2620	2621
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2622	2623	2624	2625	2626	2627	2628	2629	2630	2631	2632	2633	2634	2635	2636	2637
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2638	2639	2640	2641	2642	2643	2644	2645	2646	2647	2648	2649	2650	2651	2652	2653
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2654	2655	2656	2657	2658	2659	2660	2661	2662	2663	2664	2665	2666	2667	2668	2669
1	1	1	1	2	2	2	2	2	2	2	1	1	2	2	2
2670	2671	2672	2673	2674	2675	2676	2677	2678	2679	2680	2681	2682	2683	2684	2685
2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1
2686	2687	2688	2689	2690	2691	2692	2693	2694	2695	2696	2697	2698	2699	2700	2701
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2702	2703	2704	2705	2706	2707	2708	2709	2710	2711	2712	2713	2714	2715	2716	2717
1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2
2718	2719	2720	2721	2722	2723	2724	2725	2726	2727	2728	2729	2730	2731	2732	2733
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2734	2735	2736	2737	2738	2739	2740	2741	2742	2743	2744	2745	2746	2747	2748	2749
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1
2750	2751	2752	2753	2754	2755	2756	2757	2758	2759	2760	2761	2762	2763	2764	2765
1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2
2766	2767	2768	2769	2770	2771	2772	2773	2774	2775	2776	2777	2778	2779	2780	2781
2	2	2	1	1	1	1	1	2	1	1	1	1	2	2	1
2782	2783	2784	2785	2786	2787	2788	2789	2790	2791	2792	2793	2794	2795	2796	2797

1	1	1	2	2	2	2	2	2	1	1	1	1	1	1	1
2798	2799	2800	2801	2802	2803	2804	2805	2806	2807	2808	2809	2810	2811	2812	2813
1	1	1	1	1	2	2	2	2	2	2	1	1	1	2	1
2814	2815	2816	2817	2818	2819	2820	2821	2822	2823	2824	2825	2826	2827	2828	2829
2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2
2830	2831	2832	2833	2834	2835	2836	2837	2838	2839	2840	2841	2842	2843	2844	2845
1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2
2846	2847	2848	2849	2850	2851	2852	2853	2854	2855	2856	2857	2858	2859	2860	2861
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
2862	2863	2864	2865	2866	2867	2868	2869	2870	2871	2872	2873	2874	2875	2876	2877
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2878	2879	2880	2881	2882	2883	2884	2885	2886	2887	2888	2889	2890	2891	2892	2893
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2894	2895	2896	2897	2898	2899	2900	2901	2902	2903	2904	2905	2906	2907	2908	2909
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2910	2911	2912	2913	2914	2915	2916	2917	2918	2919	2920	2921	2922	2923	2924	2925
2	2	2	1	1	1	2	2	2	2	2	2	1	1	1	2
2926	2927	2928	2929	2930	2931	2932	2933	2934	2935	2936	2937	2938	2939	2940	2941
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2942	2943	2944	2945	2946	2947	2948	2949	2950	2951	2952	2953	2954	2955	2956	2957
2	2	2	2	2	1	1	1	1	1	1	1	1	1	1	2
2958	2959	2960	2961	2962	2963	2964	2965	2966	2967	2968	2969	2970	2971	2972	2973
2	2	2	2	2	2	2	1	1	1	1	1	1	1	1	1
2974	2975	2976	2977	2978	2979	2980	2981	2982	2983	2984	2985	2986	2987	2988	2989
1	1	2	2	2	2	2	1	1	1	1	1	1	1	1	1
2990	2991	2992	2993	2994	2995	2996	2997	2998	2999	3000	3001	3002	3003	3004	3005
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3006	3008	3009	3010	3011	3012	3013	3014	3015	3016	3017	3018	3019	3020	3021	3022
1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2
3023	3024	3025	3026	3027	3028	3029	3030	3031	3032	3033	3034	3035	3036	3037	3038
2	2	2	2	2	2	2	2	2	2	2	2	2	2	1	1
3039	3040	3041	3042	3043	3044	3045	3046	3047	3048	3049	3050	3051	3052	3053	3054
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3055	3056	3057	3058	3059	3060	3061	3062	3063	3064	3065	3066	3067	3068	3069	3070
2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1
3071	3072	3073	3074	3075	3076	3077	3078	3079	3080	3081	3082	3083	3084	3085	3086
1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2
3087	3088	3089	3090	3091	3092	3093	3094	3095	3096	3097	3098	3099	3100	3101	3102
2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1
3103	3104	3105	3106	3107	3108	3109	3110	3111	3112	3113	3114	3115	3116	3117	3118
1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2
3119	3120	3121	3122	3123											
2	2	2	2	2											

```
Within cluster sum of squares by cluster:
[1] 2.541980e+13 2.471886e+13
 (between_SS / total_SS =  88.4 %)

Available components:

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

Find the mean population, household income and per capita income grouped by clusters. In addition, display the most frequent county that can be observed within each cluster.

```
tclust$cluster[which(!is.na(tclust$hh_income))] <- km_1$cluster

tclust %>%
  filter(cluster==1) %>%
  summarise(mean_pop=mean(pop),
            mean_hh_income=mean(hh_income),
            mean_income=mean(income),
            most_freq_county=names(sort(-table(tclust$county)))[1])
```

```
  mean_pop mean_hh_income mean_income most_freq_county
1 4341.583       66127.53    31116.94              031
```

```
tclust %>%
  filter(cluster==2) %>%
  summarise(mean_pop=mean(pop),
            mean_hh_income=mean(hh_income),
            mean_income=mean(income),
            most_freq_county=names(sort(-table(tclust$county)))[1])
```

```
  mean_pop mean_hh_income mean_income most_freq_county
1 3789.817       53196.33    29850.75              031
```

As you might have seen earlier, it's not always clear which number of clusters is the optimal choice. To automate K Means clustering, program a function based on `kmeans()` that takes K as an argument. You can fix the other arguments, e.g. such that a specific dataset is always used when calling the function.

```r
kfunction <- function(k){
  kmeans(na.omit(tclust), k)
}
```

We want to utilize this function to iterate over multiple Ks (e.g., K = 2, ..., 10) and – each time – add the resulting cluster membership as a new variable to our (cleaned) original data frame (`acs_il_t`). There are multiple solutions for this task, e.g. think about the `apply` family or `for` loops.

```r
kfunction2 <- function(k){
  km <- kfunction(k)
  new_column <- paste0("cluster_", k)
  acs_il_t[[new_column]] <- NA
  acs_il_t[[new_column]][which(!is.na(tclust$hh_income))] <- km$cluster
  return(acs_il_t)
}

for (i in seq(2, 10, by=2)) {
  acs_il_t <- kfunction2(i)
}
```

Finally, display the first rows of the updated data set (with multiple cluster columns).

```r
head(acs_il_t)
```

```
  state county  tract                                         NAME  pop
1    17    031 806002 Census Tract 8060.02, Cook County, Illinois 7304
2    17    031 806003 Census Tract 8060.03, Cook County, Illinois 7577
3    17    031 806400    Census Tract 8064, Cook County, Illinois 2684
4    17    031 806501 Census Tract 8065.01, Cook County, Illinois 2590
5    17    031 750600    Census Tract 7506, Cook County, Illinois 3594
6    17    031 310200    Census Tract 3102, Cook County, Illinois 1521
  hh_income income cluster_2 cluster_4 cluster_6 cluster_8 cluster_10
1     56975  23750         1         2         4         4          5
2     53769  25016         1         2         4         4          5
3     62750  30154         1         2         4         4          5
4     53583  20282         1         2         4         4          5
5     40125  18347         1         2         4         4          5
6     63250  31403         2         3         2         6          4
```