

Sesion 4

Juan José Merino Zarco

30/6/2021

for

Verificar si un “chapingero” pasaba el año, promedio > 6.6.

```
verificar_p <- function(calificaciones){  
  promedio <- mean(calificaciones)  
  if(promedio >= 6.6){"Pasa el año"}else{"No pasa"}  
}
```

```
alumno1 <- verificar_p(c(9,10,9,8))  
alumno1
```

```
## [1] "Pasa el año"
```

```
alumno2 <- verificar_p(c(6.6,9,7,4))  
alumno2
```

```
## [1] "Pasa el año"
```

for

for(elemento *in* objeto){operacion con cada elemento}

```
dado <- c(1:6)  
dado
```

```
## [1] 1 2 3 4 5 6
```

```
dup <- NULL
```

```
for(elemento in dado){  
  dup[elemento] <- elemento + 3  
}
```

```
dup
```

```
## [1] 4 5 6 7 8 9
```

Librerias

Manual

```
install.packages("readr")
```

```
library("readr")
```

Automatico

```
pacman::p_load(readr)
```

Cargar una base de datos de inegi, Censo economico del 2019, archivo .csv

```
getwd()
```

```
## [1] "C:/Users/Omen/Desktop/Doctorado/Sesion 4"
```

Cargar la base de datos

```
pacman::p_load(readr)
```

El archivo que se uso esta muy pesado, por lo que dejo el enlace directo para que descarguen el archivo por si necesitan replicar el ejercicio.

https://www.inegi.org.mx/contenidos/programas/ce/2019/Datosabiertos/ce2019_mich_csv.zip

```
CE2019Mich <- read_csv(file = "ce2019_mich.csv")
```

```
##
## -- Column specification -----
## cols(
##   .default = col_double(),
##   MUNICIPIO = col_logical(),
##   CODIGO = col_character()
## )
## i Use 'spec()' for the full column specifications.

## Warning: 198145 parsing failures.
## row col   expected   actual      file
##   1 -- 186 columns 187 columns 'ce2019_mich.csv'
##   2 -- 186 columns 187 columns 'ce2019_mich.csv'
##   3 -- 186 columns 187 columns 'ce2019_mich.csv'
##   4 -- 186 columns 187 columns 'ce2019_mich.csv'
##   5 -- 186 columns 187 columns 'ce2019_mich.csv'
## ... ..
## See problems(...) for more details.
```

```
head(CE2019Mich)
```

```
## # A tibble: 6 x 186
##   ENTIDAD MUNICIPIO CODIGO ID_ESTRATO   UE   A111A   A121A   A131A A211A A221A
##   <dbl> <lg1>      <chr>      <dbl> <dbl>   <dbl>   <dbl>   <dbl> <dbl> <dbl>
## 1    16 NA        <NA>        NA 230966 325855. 173867. 151988. 4357. 4534.
## 2    16 NA        <NA>         1 223776 73074. 30682. 42392. 1379. 1595.
## 3    16 NA        <NA>         2  6059 56443. 21826. 34617.  806.  983.
## 4    16 NA        <NA>         3   988 70290. 35607. 34683. 1071. 1133.
## 5    16 NA        <NA>         4   143 126048. 85753. 40296. 1102.  822.
## 6    16 NA        11        NA   842 1557.  874.  683. 151. 154.
## # ... with 176 more variables: A511A <dbl>, A700A <dbl>, A800A <dbl>,
## #   H000A <dbl>, H000B <dbl>, H000C <dbl>, H000D <dbl>, H001A <dbl>,
## #   H001B <dbl>, H001C <dbl>, H001D <dbl>, H010A <dbl>, H010B <dbl>,
## #   H010C <dbl>, H010D <dbl>, H020A <dbl>, H020B <dbl>, H020C <dbl>,
## #   H020D <dbl>, H101A <dbl>, H101B <dbl>, H101C <dbl>, H101D <dbl>,
## #   H203A <dbl>, H203B <dbl>, H203C <dbl>, H203D <dbl>, I000A <dbl>,
## #   I000B <dbl>, I000C <dbl>, I000D <dbl>, I100A <dbl>, I100B <dbl>,
## #   I100C <dbl>, I100D <dbl>, I200A <dbl>, I200B <dbl>, I200C <dbl>,
## #   I200D <dbl>, J000A <dbl>, J010A <dbl>, J203A <dbl>, J300A <dbl>,
## #   J400A <dbl>, J500A <dbl>, J600A <dbl>, K000A <dbl>, K010A <dbl>,
## #   K020A <dbl>, K030A <dbl>, K042A <dbl>, K412A <dbl>, K050A <dbl>,
## #   K060A <dbl>, K070A <dbl>, K090A <dbl>, K096A <dbl>, K311A <dbl>,
## #   K610A <dbl>, K620A <dbl>, K810A <dbl>, K820A <dbl>, K910A <dbl>,
## #   K950A <dbl>, K976A <dbl>, M000A <dbl>, M010A <dbl>, M020A <dbl>,
## #   M030A <dbl>, M050A <dbl>, M090A <dbl>, M091A <dbl>, M700A <dbl>,
## #   O010A <dbl>, O020A <dbl>, P000A <dbl>, P000B <dbl>, P000C <dbl>,
## #   P030A <dbl>, P030B <dbl>, P030C <dbl>, P100A <dbl>, P100B <dbl>,
## #   Q000A <dbl>, Q000B <dbl>, Q000C <dbl>, Q000D <dbl>, Q010A <dbl>,
## #   Q020A <dbl>, Q030A <dbl>, Q400A <dbl>, Q900A <dbl>, A112A <dbl>,
## #   A113A <dbl>, A114A <dbl>, A115A <dbl>, A116A <dbl>, A117A <dbl>,
## #   A118A <dbl>, A119A <dbl>, ...
```

```
class(CE2019Mich)
```

```
## [1] "spec_tbl_df" "tbl_df"      "tbl"        "data.frame"
```

Crear una variable que nos permita identificar cada renglon de nuestro dataframe

```
ID <- seq.int(nrow(CE2019Mich))
```

```
CE2019Mich_ID <- cbind(CE2019Mich, ID)
```

Filtrar por las variables de interes

```
CE2019Mich_ID_filt <- CE2019Mich_ID[c("ENTIDAD", "CODIGO", "ID_ESTRATO", "UE", "A111A", "A121A", "A800A", "H001D")]
```

```
head(CE2019Mich_ID_filt)
```

```
##   ENTIDAD CODIGO ID_ESTRATO   UE   A111A   A121A   A800A   H001D
```

```
## 1      16  <NA>      NA 230966 325855.326 173867.374 536969.499 1821937.672
## 2      16  <NA>      1 223776  73073.998  30681.831 135879.604 1052094.544
## 3      16  <NA>      2   6059  56443.361  21825.972 133376.542  291887.188
## 4      16  <NA>      3    988  70289.599  35606.960 137701.429  237918.788
## 5      16  <NA>      4    143 126048.367  85752.611 130011.924  240037.152
## 6      16    11      NA   842   1557.166    874.271   1602.665    3248.424
##      H010A ID
## 1 385018  1
## 2 152175  2
## 3  90686  3
## 4  70500  4
## 5  71657  5
## 6   2474  6
```

Ahora, vamos a quedarnos unicamente con el sector y subsectores de interes

```
CE2019Mich_ID_filt_filt <- CE2019Mich_ID_filt[CE2019Mich_ID_filt$ID > 295 & CE2019Mich_ID_filt$ID < 132]
```

Crear una variable CLAVE, que no tenga guiones

```
pacman::p_load(taRifx)
```

```
CLAVE <- destring(CE2019Mich_ID_filt_filt$CODIGO, keep = "0-9")
```

```
CE2019Mich_toedit <- cbind(CE2019Mich_ID_filt_filt, CLAVE)
```

Solo me interesan sectores y subsectores, procedemos a filtrar nuevamente

```
CE_MICH_2019 <- CE2019Mich_toedit[CE2019Mich_toedit$CLAVE < 1000 | CE2019Mich_toedit$CODIGO == "31-33",]
```

Sustituir el 3133 por un 300(Industria manufacturera)

```
CE_MICH_2019$CLAVE[CE_MICH_2019$CLAVE == 3133] <- 300
```

Vamos a crear una CLAVE_UNICA

comando paste0(), sirve para combinar los valores de varias columnas.

```
CE_MICH_2019$CLAVE_UNICA <- paste0(CE_MICH_2019$ENTIDAD, CE_MICH_2019$CLAVE, CE_MICH_2019$ID)
```

Esta lista nuestra base de datos, esta limpia

```
CE_MICH_2019
```

```
##      ENTIDAD CODIGO ID_ESTRATO    UE      A111A      A121A      A800A
## 296      16  31-33          NA 37608 144845.463 106396.564 142078.797
## 297      16  31-33           1 36655  12913.915   7480.214  12927.468
## 298      16  31-33           2   776   6804.066   4290.875   6903.137
## 299      16  31-33           3   138  22705.843  16982.296  23517.345
## 300      16  31-33           4    39 102421.639  77643.179  98730.847
## 301      16    311          NA 11089  40625.673  29732.544  41085.166
```

## 302	16	311	1	10810	5443.710	3148.499	5453.327
## 303	16	311	2	204	2212.020	1490.969	2259.278
## 304	16	311	3	57	8428.085	6287.545	8876.762
## 305	16	311	4	18	24541.858	18805.531	24495.799
## 462	16	312	NA	607	12117.904	6064.175	12126.500
## 463	16	312	99	607	12117.904	6064.175	12126.500
## 492	16	313	NA	276	645.785	589.060	656.293
## 493	16	313	99	276	645.785	589.060	656.293
## 528	16	314	NA	2695	2300.905	1754.179	2351.636
## 529	16	314	1	2668	166.837	84.517	167.512
## 530	16	314	2	16	84.924	53.266	86.050
## 531	16	314	3	7	809.746	674.887	862.423
## 532	16	314	4	4	1239.398	941.509	1235.651
## 565	16	315	NA	1786	810.559	473.605	843.581
## 566	16	315	99	1786	810.559	473.605	843.581
## 604	16	316	NA	1075	916.785	598.821	916.164
## 605	16	316	99	1075	916.785	598.821	916.164
## 637	16	321	NA	3840	2820.308	2160.643	2836.647
## 638	16	321	99	3840	2820.308	2160.643	2836.647
## 683	16	322	NA	126	4336.029	3024.680	4604.037
## 684	16	322	99	126	4336.029	3024.680	4604.037
## 709	16	323	NA	608	906.492	595.672	904.629
## 710	16	323	99	608	906.492	595.672	904.629
## 729	16	324	NA	1	NA	NA	NA
## 730	16	324	99	1	NA	NA	NA
## 737	16	325	NA	171	14889.371	11914.447	15436.870
## 738	16	325	1	124	129.844	87.300	129.859
## 739	16	325	2	31	749.431	468.956	751.215
## 740	16	325	3	12	3050.580	2123.351	3127.965
## 741	16	325	4	4	10959.516	9234.840	11427.831
## 816	16	326	NA	126	9102.791	6259.051	9417.442
## 817	16	326	1	76	117.521	84.641	116.736
## 818	16	326	2	21	251.208	171.160	253.743
## 819	16	326	3	24	3048.370	2301.161	3183.985
## 820	16	326	4	5	5685.692	3702.089	5862.978
## 878	16	327	NA	5537	2168.393	1390.953	2265.653
## 879	16	327	1	5496	904.017	525.470	908.100
## 880	16	327	2	38	564.901	440.150	580.035
## 881	16	327	3	3	699.475	425.333	777.518
## 954	16	331	NA	32	46219.038	37811.865	41629.568
## 955	16	331	99	32	46219.038	37811.865	41629.568
## 998	16	332	NA	3674	2953.207	1636.929	2957.449
## 999	16	332	99	3674	2953.207	1636.929	2957.449
## 1075	16	333	NA	65	970.297	577.405	944.621
## 1076	16	333	1	44	52.428	29.029	50.373
## 1077	16	333	2	17	274.565	155.163	285.866
## 1078	16	333	3	4	643.304	393.213	608.382
## 1141	16	334	NA	5	NA	NA	NA
## 1142	16	334	99	5	NA	NA	NA
## 1157	16	335	NA	18	NA	NA	NA
## 1158	16	335	99	18	NA	NA	NA
## 1189	16	336	NA	51	108.013	56.422	100.525
## 1190	16	336	99	51	108.013	56.422	100.525
## 1228	16	337	NA	3366	1945.833	1124.206	1960.014

##	1229	16	337	1	3290	1358.016	786.821	1361.354
##	1230	16	337	2	71	372.337	201.866	372.828
##	1231	16	337	3	5	215.480	135.519	225.832
##	1266	16	339	NA	2460	730.006	392.711	734.858
##	1267	16	339	99	2460	730.006	392.711	734.858
##		H001D	H010A	ID	CLAVE	CLAVE_UNICA		
##	296	322443.130	65263	296	300	16300296		
##	297	171717.750	25085	297	300	16300297		
##	298	36533.548	12232	298	300	16300298		
##	299	39130.832	12826	299	300	16300299		
##	300	75061.000	15120	300	300	16300300		
##	301	127771.058	29540	301	311	16311301		
##	302	64161.746	10964	302	311	16311302		
##	303	10786.480	3364	303	311	16311303		
##	304	15449.832	5576	304	311	16311304		
##	305	37373.000	9636	305	311	16311305		
##	462	11638.708	2647	462	312	16312462		
##	463	11638.708	2647	463	312	16312463		
##	492	2454.680	591	492	313	16313492		
##	493	2454.680	591	493	313	16313493		
##	528	13971.318	2730	528	314	16314528		
##	529	6555.282	235	529	314	16314529		
##	530	730.036	202	530	314	16314530		
##	531	2966.000	743	531	314	16314531		
##	532	3720.000	1550	532	314	16314532		
##	565	9771.252	1899	565	315	16315565		
##	566	9771.252	1899	566	315	16315566		
##	604	8485.640	2193	604	316	16316604		
##	605	8485.640	2193	605	316	16316605		
##	637	18572.796	3887	637	321	16321637		
##	638	18572.796	3887	638	321	16321638		
##	683	3960.524	584	683	322	16322683		
##	684	3960.524	584	684	322	16322684		
##	709	5290.562	1502	709	323	16323709		
##	710	5290.562	1502	710	323	16323710		
##	729	NA	NA	729	324	16324729		
##	730	NA	NA	730	324	16324730		
##	737	11456.176	2573	737	325	16325737		
##	738	849.184	202	738	325	16325738		
##	739	1663.992	585	739	325	16325739		
##	740	2519.000	809	740	325	16325740		
##	741	6424.000	977	741	325	16325741		
##	816	12714.368	3736	816	326	16326816		
##	817	660.528	141	817	326	16326817		
##	818	1110.840	334	818	326	16326818		
##	819	6314.000	2127	819	326	16326819		
##	820	4629.000	1134	820	326	16326820		
##	878	26178.214	2358	878	327	16327878		
##	879	24191.398	1740	879	327	16327879		
##	880	1380.816	351	880	327	16327880		
##	881	606.000	267	881	327	16327881		
##	954	16906.024	448	954	331	16331954		
##	955	16906.024	448	955	331	16331955		
##	998	18062.542	3849	998	332	16332998		

```
## 999 18062.542 3849 999 332 16332999
## 1075 2698.736 1048 1075 333 163331075
## 1076 325.224 96 1076 333 163331076
## 1077 977.512 386 1077 333 163331077
## 1078 1396.000 566 1078 333 163331078
## 1141 NA NA 1141 334 163341141
## 1142 NA NA 1142 334 163341142
## 1157 NA NA 1157 335 163351157
## 1158 NA NA 1158 335 163351158
## 1189 580.776 165 1189 336 163361189
## 1190 580.776 165 1190 336 163361190
## 1228 19718.106 3457 1228 337 163371228
## 1229 15846.114 2083 1229 337 163371229
## 1230 2661.992 881 1230 337 163371230
## 1231 1210.000 493 1231 337 163371231
## 1266 11305.570 1698 1266 339 163391266
## 1267 11305.570 1698 1267 339 163391267
```

Vamos a guardar la base de datos. `write.csv()`

```
write.csv(x = CE_MICH_2019, file = "CE_MICH_2019_CS.csv", row.names = FALSE)
```

para juntarla con otros estados, usariamos el comando `merge()`.
