Práctica 0

Álvaro Francisco Ruiz Cornejo

14/9/2021

Ejercicio 1

```
(dado_honesto = sample (1:6, size = 100, replace = TRUE))
```

```
##   [1] 4 4 2 2 2 1 5 6 4 6 1 1 2 2 2 2 2 5 5 1 5 2 5 2 4 3 4 6 5 4 3 2 2 6 3 2 1
##  [38] 4 2 4 2 4 2 6 4 6 4 3 6 4 5 2 6 2 2 4 4 6 2 5 2 2 2 1 6 1 2 3 6 6 2 1 2 4
##  [75] 2 6 2 1 5 3 3 2 3 5 4 6 6 3 4 4 2 6 3 6 3 1 3 2 1 2
```

```
table(dado_honesto) #frecuencia absoluta R básico
```

```
## dado_honesto
##  1  2  3  4  5  6 
## 11 32 12 18 10 17
```

```
signif(prop.table(table(dado_honesto)),2) # frecuencia relativa R básico
```

```
## dado_honesto
##    1    2    3    4    5    6 
## 0.11 0.32 0.12 0.18 0.10 0.17
```

```
t = tibble(col_tirada = dado_honesto)
t %>%
  count(col_tirada) #frecuencia absoluta con dplyr
```

```
## # A tibble: 6 x 2
##   col_tirada     n
##        <int> <int>
## 1          1    11
## 2          2    32
## 3          3    12
## 4          4    18
## 5          5    10
## 6          6    17
```

```
t %>%
  count(col_tirada) %>%
  mutate(col_tirada, relFreq = prop.table(n), n = NULL) #frecuencia relativa con dplyr
```

```
## # A tibble: 6 x 2
##   col_tirada relFreq
##        <int>   <dbl>
## 1          1    0.11
## 2          2    0.32
## 3          3    0.12
## 4          4    0.18
## 5          5    0.1
## 6          6    0.17
```

Ejercicio 2

Trucamos un dado para duplicar la posibilidad de sacar un seis

```
(dado_cargado = sample(1:6, size = 100, replace = TRUE,
                      prob = c((1/7),(1/7), (1/7), (1/7), (1/7),(2/7))))
```

```
##   [1] 2 6 6 3 3 4 4 3 6 1 1 4 6 2 6 6 2 6 6 5 3 1 1 5 2 6 4 5 6 5 3 5 5 6 5 3 4
##  [38] 4 2 4 3 6 1 6 4 6 5 6 3 3 1 6 1 6 6 6 6 5 6 6 4 5 3 6 2 5 1 5 4 4 3 6 6 3
##  [75] 6 5 4 4 1 3 6 6 6 5 6 1 1 4 1 6 3 6 3 3 1 2 5 2 6 5
```

```
table(dado_cargado) #frecuencia absoluta
```

```
## dado_cargado
##  1  2  3  4  5  6
## 13  8 16 14 16 33
```

```
signif(prop.table(table(dado_cargado)),2) # frecuencia relativa
```

```
## dado_cargado
##    1    2    3    4    5    6
## 0.13 0.08 0.16 0.14 0.16 0.33
```

Ejercicio 3

```
(v1 = rep(seq(from = 1, to = 4, by = 1), each = 4))
```

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

```
(v2 = rep(seq(from = 1, to = 5, by = 1), times = c(1,2,3,4,5)))
```

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

```
(v3 = rep(seq(from = 1, to = 4, by = 1), times = 4))
```

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

Ejercicio 4

```
(mpg2 <- mpg %>%
          filter(class == 'pickup') %>%
            select(starts_with("c")))
```

```
## # A tibble: 33 x 3
##      cyl   cty class
##    <int> <int> <chr>
##  1     6    15 pickup
##  2     6    14 pickup
##  3     6    13 pickup
##  4     6    14 pickup
##  5     8    14 pickup
##  6     8    14 pickup
##  7     8     9 pickup
##  8     8    11 pickup
##  9     8    11 pickup
## 10     8    12 pickup
## # ... with 23 more rows
```

Ejercicio 5

5.1

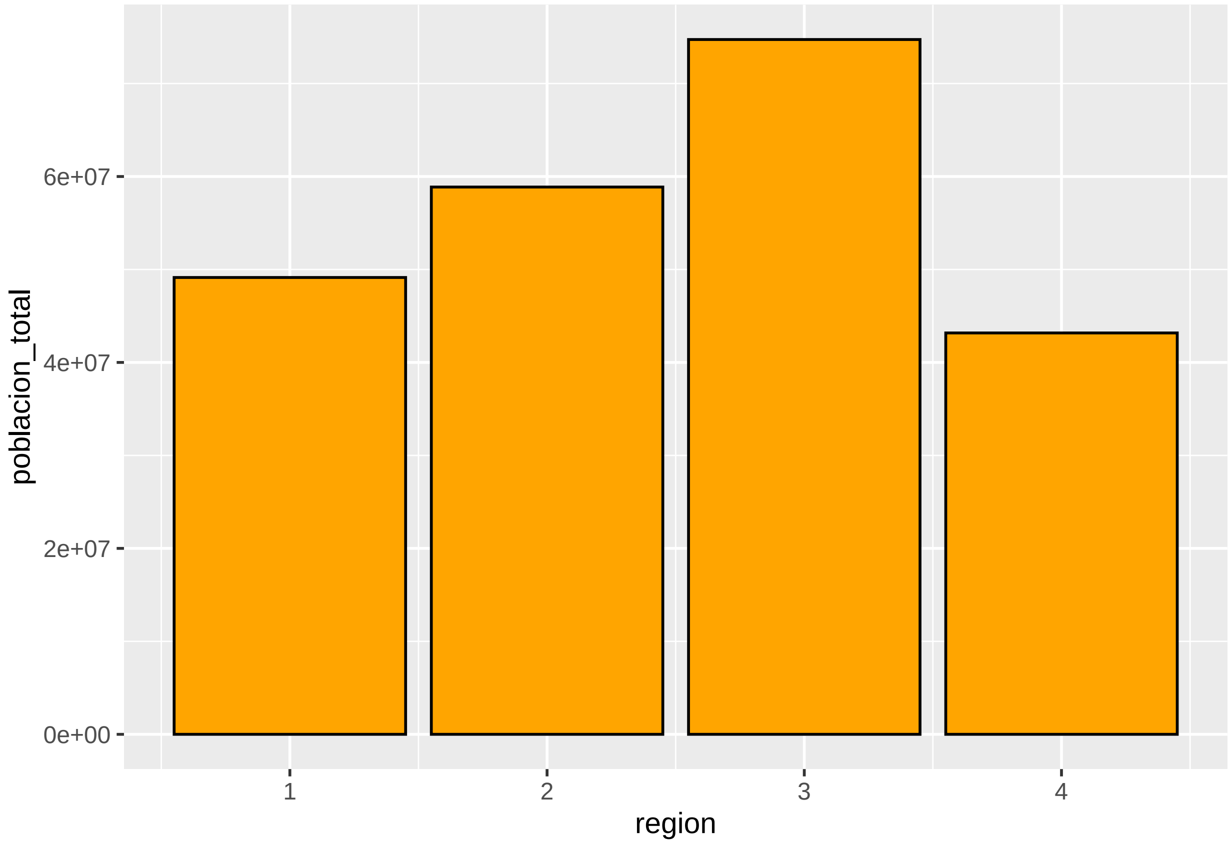
```
(pob_region = census %>%
              group_by(region) %>%
                summarise(poblacion_total = sum(pop), n = n()))
```

```
## # A tibble: 4 x 3
##         region poblacion_total     n
##      <dbl+lbl>           <dbl> <int>
## 1 1 [NE]              49135283     9
## 2 2 [N Cntrl]         58865670    12
## 3 3 [South]           74734029    16
## 4 4 [West]            43172490    13
```

5.2

```
ggplot(pob_region) +
  geom_col(aes(region, poblacion_total), fill = 'orange',
           color = 'black')
```

```
## Don't know how to automatically pick scale for object of type haven_labelled/vctrs_vctr/double. Defau
```



5.3

```
(orden <- census %>%
          arrange(desc(pop)))
```

```
## # A tibble: 50 x 12
##    state       region    pop poplt5 pop5_17 pop18p pop65p popurban medage  death
##    <chr>    <dbl+lbl>  <dbl>  <dbl>   <dbl>  <dbl>  <dbl>    <dbl>  <dbl>  <dbl>
##  1 Califor~ 4 [West]  2.37e7 1.71e6 4680558 1.73e7 2.41e6 21607606   29.9 186428
##  2 New York 1 [NE]    1.76e7 1.14e6 3551938 1.29e7 2.16e6 14858068   31.9 171769
##  3 Texas    3 [South] 1.42e7 1.17e6 3137045 9.92e6 1.37e6 11333017   28.2 108019
##  4 Pennsyl~ 1 [NE]    1.19e7 7.47e5 2375838 8.74e6 1.53e6  8220851   32.1 123261
##  5 Illinois 2 [N Cnt~ 1.14e7 8.42e5 2400796 8.18e6 1.26e6  9518039   29.9 102230
##  6 Ohio     2 [N Cnt~ 1.08e7 7.87e5 2307170 7.70e6 1.17e6  7918259   29.9  98268
##  7 Florida  3 [South] 9.75e6 5.70e5 1789412 7.39e6 1.69e6  8212385   34.7 104190
##  8 Michigan 2 [N Cnt~ 9.26e6 6.85e5 2066873 6.51e6 9.12e5  6551551   28.8  75102
##  9 New Jer~ 1 [NE]    7.36e6 4.63e5 1527572 5.37e6 8.60e5  6557377   32.2  68762
## 10 N. Caro~ 3 [South] 5.88e6 4.04e5 1253659 4.22e6 6.03e5  2822852   29.6  48426
## # ... with 40 more rows, and 2 more variables: marriage <dbl>, divorce <dbl>
```

5.4

```
(tasa_matdiv <- census %>%
  select(state, marriage, divorce) %>%
  mutate(tasa = divorce/marriage)) %>%
  arrange(desc(tasa))
```

```
## # A tibble: 50 x 4
##    state       marriage divorce  tasa
##    <chr>          <dbl>   <dbl> <dbl>
##  1 Oregon         23004   17762 0.772
##  2 Indiana        57853   40006 0.692
##  3 Florida       108344   71579 0.661
##  4 Arizona        30223   19908 0.659
##  5 Alaska          5361    3517 0.656
##  6 California    210864  133541 0.633
##  7 New Mexico     16641   10426 0.627
##  8 N. Carolina    46718   28050 0.600
##  9 Washington     47728   28642 0.600
## 10 Arkansas       26513   15882 0.599
## # ... with 40 more rows
```

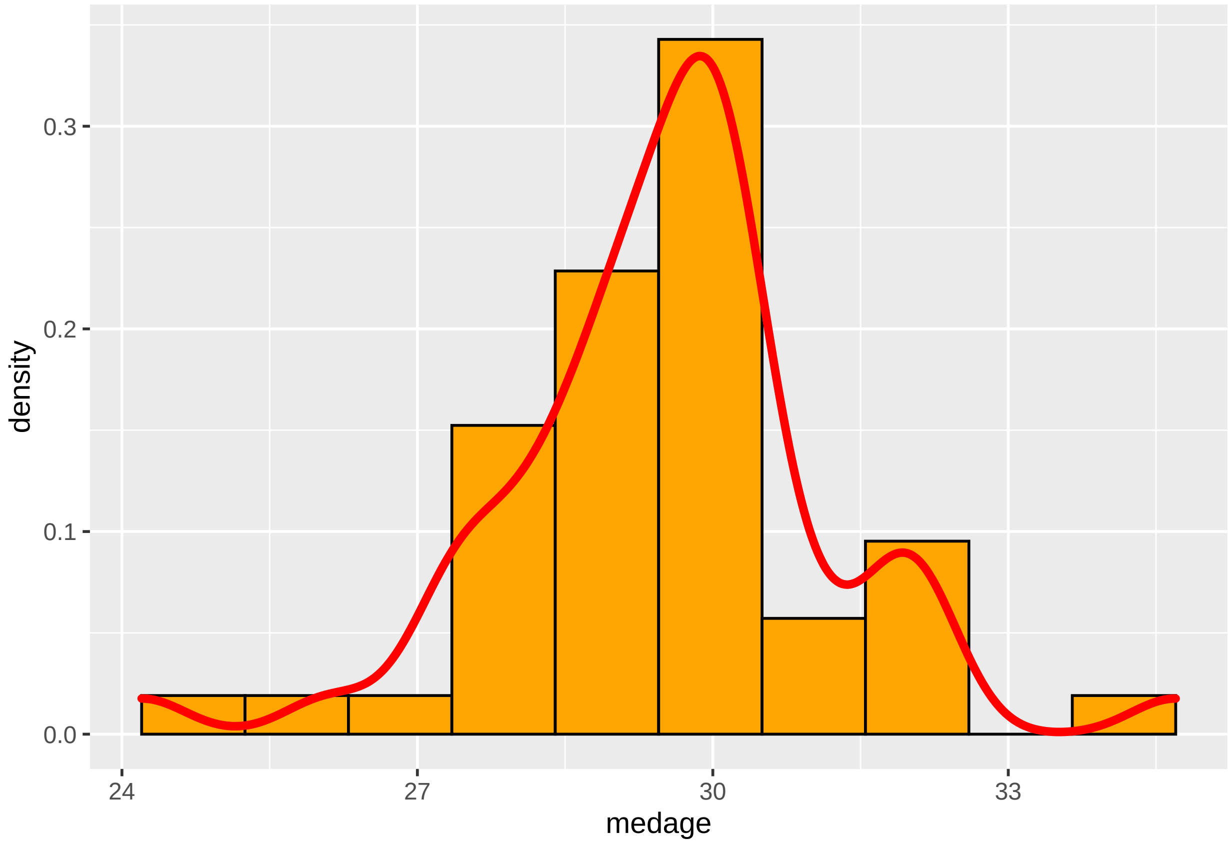
5.5

```
census %>%
  summarise(state, medage,
            mediana = median(medage), desvMediana = (medage-mediana),
            propMayorEdad = pop65p/pop) %>%
    arrange(desc(propMayorEdad)) %>%
      head(10)
```

```
## # A tibble: 10 x 5
##    state         medage mediana desvMediana propMayorEdad
##    <chr>          <dbl>   <dbl>       <dbl>         <dbl>
##  1 Florida         34.7    29.8      4.95           0.173
##  2 Arkansas        30.6    29.8      0.850          0.137
##  3 Rhode Island    31.8    29.8      2.05           0.134
##  4 Iowa            30      29.8      0.25           0.133
##  5 Missouri        30.9    29.8      1.15           0.132
##  6 S. Dakota       28.9    29.8     -0.850          0.132
##  7 Nebraska        29.7    29.8     -0.0500         0.131
##  8 Kansas          30.1    29.8      0.350          0.130
##  9 Pennsylvania    32.1    29.8      2.35           0.129
## 10 Massachusetts   31.2    29.8      1.45           0.127
```

5.6

```r
cortes = seq(min(census$medage), max(census$medage), length.out = 11)
ggplot(data = census, aes(x=medage)) +
  geom_histogram(mapping = aes(y=stat(density)), breaks = cortes,
                 fill = "orange", color = "black") +
  geom_density(color = "red", size = 1.5)
```


density
medage