Problem Set 4

2024-02-14

1. This is my recreation David and Pickering's Table 1. I have included the code for doing so below.

```
load("ssa_water.Rdata")
New_set <- water %>%
  group_by(countryModule) %>%
  summarize(
    N = n(),
    Year = first(interviewYear),
    Walk_time_Mean = mean(waterTimeMins, na.rm = TRUE),
    SD_Walk_time = sd(waterTimeMins, na.rm = TRUE),
    Median_Walk_time = median(waterTimeMins, na.rm = TRUE),
    Percent_on_plot = (sum(waterTimeMins == 0, na.rm = TRUE) / N * 100)
  )

sum(New_set$N)
```

```
## [1] 623926
```

```
mean(New_set$Walk_time_Mean)
```

```
## [1] 22.92044
```

```
mean(New_set$SD_Walk_time)
```

```
## [1] 32.25327
```

```
mean(New_set$Median_Walk_time)
```

```
## [1] 12.82143
```

```
mean(New_set$Percent_on_plot)
```

```
## [1] 23.78786
```

```
new_row = data.frame(
  countryModule = c("Total"),
  N = 623926,
  Year = NA,
  Walk_time_Mean = 22.92044,
  SD_Walk_time = 32.25327,
```

```r
  Median_Walk_time = 12.82143,
  Percent_on_plot = 23.78786
)

New_new = rbind(New_set, new_row) %>%
  rename( country = countryModule,
          year = Year,
          mean = Walk_time_Mean,
          SD = SD_Walk_time,
          med = Median_Walk_time,
          `% on plot`= Percent_on_plot)

 kable(New_new)
```

country	N	year	mean	SD	med	% on plot
BF6	28849	2010	20.30765	18.96601	15.00000	9.258553
BJ6	32486	2012	14.31397	30.77288	5.00000	31.896817
CD6	33656	2014	33.29898	30.86483	30.00000	5.939506
CI6	16298	2012	14.91906	29.28821	5.00000	42.723033
CM6	24163	2011	20.30567	27.45360	10.00000	16.951538
ET6	27715	2003	52.50238	71.83227	30.00000	13.292441
GA6	12854	2012	21.40812	36.04820	5.00000	44.639801
GH5	15697	2008	18.22342	26.30430	10.00000	16.385297
GN6	16380	2012	23.26777	24.87105	20.00000	23.394383
KE5	13105	2009	27.99992	40.75140	15.00000	27.508585
KM6	8141	2012	10.33154	24.79671	0.00000	66.060681
LB6	16229	2013	17.05797	19.16343	10.00000	7.412656
LS5	14300	2009	20.14084	24.04501	10.00000	12.552448
MD5	31744	2008	16.69193	41.38327	10.00000	14.475176
ML6	20598	2012	10.17318	27.79201	3.00000	35.993786
MW5	43732	2010	29.90817	31.87291	20.00000	9.087167
MZ6	22075	2011	29.18957	49.97817	15.00000	22.989808
NG6	58812	2013	20.33144	29.20488	10.00000	20.470312
NM6	12171	2013	13.99334	26.47023	0.00000	49.428970
RW6	19246	2011	36.39345	33.44114	30.00000	6.006443
SL6	25749	2013	21.04759	22.97983	15.00000	10.046992
SN6	26564	2011	12.25503	31.27550	0.00000	53.613914
SZ5	8025	2006	21.12428	30.88023	10.00000	32.186916
TG6	16280	2013	23.70795	29.45507	15.00000	12.610565
TZ5	17251	2010	29.99512	39.11909	16.00000	13.262999
UG6	16740	2011	45.74853	49.90601	30.00000	9.510155
ZM6	30871	2013	17.68385	23.84441	10.00000	23.306663
ZW6	14195	2010	19.45173	30.33077	10.00000	35.054597
Total	623926	NA	22.92044	32.25327	12.82143	23.787860

```r
 #%>% add_header_above(header = c(" " = 3, "walk time (min)" = 3, " " = 1))
```

2. I have included the code for the 3 merged datasets below.

```r
Unique_file_1 = read_csv("exp10.csv") %>%
  select(start, CompCode, Bonus)

Unique_file_2 = read_csv("exp11.csv") %>%
  select(start, CompCode, Pay) %>%
  rename(Bonus = Pay)


Unique_file_3 = read_csv("Crid.csv") %>%
  select(AssignmentID, rID,CompCode)

Merge_u_12 =
  full_join(
    Unique_file_2,
    Unique_file_1,
    by = join_by("CompCode", "Bonus", "start")
  ) %>%
  select(CompCode, Bonus)

second_try =
  inner_join(
    Unique_file_3,
    Merge_u_12,
    by = join_by("CompCode")
  )

str(second_try)
```

```
## tibble [1,301 x 4] (S3: tbl_df/tbl/data.frame)
##  $ AssignmentID: chr [1:1301] "3M0BCWMB8Z1DXVB1HGCN25MD5ETBWY" "3D4CH1LGEEYYCG644RU9PW5ZNQ09GN" "3HL
##  $ rID         : chr [1:1301] "A1P6LFEAY9MWAY" "A22OPXKTGLCX7B" "A2N4Q60TCBWBL4" "A22HIX1M4QXZBB" ..
##  $ CompCode    : chr [1:1301] "R_3oHeE7Vq9RAfwrA" "R_2cqk5ElS6J1eXWu" "R_3G95bZjmyBO2OFV" "R_bpwQBDM
##  $ Bonus       : num [1:1301] 71 65 64 63 65 53 65 70 62 61 ...
```

3. Here is the code I used for identifying the most common letters

```r
Wordle = load("WordleDictionary.Rdata")

wider =
  wordle %>%
  select("let1", "let2", "let3", "let4", "let5")

new_long =
  wider %>%
  pivot_longer(
    cols = starts_with('let'),
    names_to = 'let1',
    values_to = 'letters'
  )

  By_letter =
new_long %>%
  group_by(letters) %>%
```

```
  summarise(frequency =n()) %>%
  arrange(desc(frequency))

  kable(By_letter, digits =1)
```

letters	frequency
s	6665
e	6662
a	5990
o	4438
r	4158
i	3759
l	3371
t	3295
n	2952
u	2511
d	2453
y	2074
c	2028
p	2019
m	1976
h	1760
g	1644
b	1627
k	1505
f	1115
w	1039
v	694
z	434
j	291
x	288
q	112

4. Here is the code for how I identified the frequency of appearance in the dictionary of each letter. The top 5 highest scoring words are esses with a score of 33319, asses with a score of 32647, sasse with a score of 32647, sessa with a score of 32647, and eases with a score of 32644. The lowest scoring words were muzzy with a score of 7429, whizz with a score of 7426, huzzy with a score of 7213, buzzy with a score of 7080, fuzzy with a score of 6568.

```
letter_to_number = c("s" = 6665, "e" = 6662, "a" = 5990, "o" = 4438,
                     "r" = 4158, "i" = "3759", "l" = "3371", "t" = 3295,
                     "n" = 2952, "u" = 2511, "d" = "2453", "y" = 2074,
                     "c" = 2028, "p" = 2019, "m" = 1976, "h" = 1760, "g" = 1644,
                     "b" = 1627, "k" = 1505, "f" = 1115, "w" = 1039, "v" = 694,
                     "z" = 434, "j" = 291, "x" = 288, "q" = 112)

Wordle_score =
  wordle %>%
  mutate(across(starts_with("let"), ~ifelse(. %in% names(letter_to_number),
                                            letter_to_number[.], .))) %>%
  mutate(across(starts_with("let"), as.numeric)) %>%
```

```
  mutate(Wordle_score = rowSums(select(., starts_with("let")), na.rm = TRUE)) %>%
  arrange(desc(Wordle_score))
```

5. Among words with no duplicate letters, the best words to use are areos with a score of 27913, arose with a score of 27913, soare with a scoare of 27913, aesir with a score of 27234, and arise with a score of 27234. The worst words to use are whump with a score of 9305, judgy with a score of 8973, jumpy with a score 8871, vughy with a score of 8683, and jumby with a score of 8479.

```
No_duplicates = Wordle_score %>%
  mutate(duplicates = apply(select(., starts_with("let")), 1, function(row)
    any(duplicated(row)))) %>%
   filter(duplicates == "FALSE")
```