

December 11, 2022

The results below are generated from an R script.

```
# Assignment: ASSIGNMENT 2
# Name: Rodriguez, Felipe
# Date: 2022-12-11

## Check your current working directory using 'getwd()'
getwd()

## [1] "/Users/feliperodriguez/Library/CloudStorage/OneDrive-BellevueUniversity/DSC 520 Statistics/Week

## List the contents of the working directory with the 'dir()' function
dir()

## [1] "assignment_00_RodriguezFelipe_DSC520.pdf" "assignment_01_LastnameFirstname.log"
## [3] "assignment_01_LastnameFirstname.pdf"      "assignment_01_LastnameFirstname.tex"
## [5] "assignment_02_Rodriguez_Felipe.R"

## If the current directory does not contain the 'data' directory, set the
## working directory to project root folder (the folder should contain the 'data' directory
## Use 'setwd()' if needed
setwd("/Users/feliperodriguez/OneDrive - Bellevue University/Github/dsc520/")

## Load the file 'data/tidynomicon/person.csv' to 'person_df1' using 'read.csv'
## Examine the structure of 'person_df1' using 'str()'
person_df1 <- read.csv(file="data/tidynomicon/person.csv")
str(person_df1)

## 'data.frame': 5 obs. of 3 variables:
## $ person_id : chr "dyer" "pb" "lake" "roe" ...
## $ personal_name: chr "William" "Frank" "Anderson" "Valentina" ...
## $ family_name : chr "Dyer" "Pabodie" "Lake" "Roerich" ...

## R interpreted names as factors, which is not the behavior we want
## Load the same file to person_df2 using 'read.csv' and setting 'stringsAsFactors' to 'FALSE'
## Examine the structure of 'person_df2' using 'str()'
person_df2 <- read.csv(file='data/tidynomicon/person.csv', stringsAsFactors = FALSE)
str(person_df2)

## 'data.frame': 5 obs. of 3 variables:
## $ person_id : chr "dyer" "pb" "lake" "roe" ...
## $ personal_name: chr "William" "Frank" "Anderson" "Valentina" ...
## $ family_name : chr "Dyer" "Pabodie" "Lake" "Roerich" ...

## Read the file 'data/scores.csv' to 'scores_df'
## Display summary statistics using the 'summary()' function
scores_df <- read.csv(file='data/scores.csv')
summary(scores_df)
```

```

##      Count      Score      Section
## Min.   :10.00  Min.   :200.0  Length:38
## 1st Qu.:10.00  1st Qu.:300.0  Class :character
## Median :10.00  Median :322.5  Mode  :character
## Mean   :14.47  Mean   :317.5
## 3rd Qu.:20.00  3rd Qu.:357.5
## Max.   :30.00  Max.   :395.0

## Load the 'readxl' library
library('readxl')

## Using the excel_sheets() function from the 'readxl' package,
## list the worksheets from the file 'data/G04ResultsDetail2004-11-02.xls'
excel_sheets('data/G04ResultsDetail2004-11-02.xls')

## [1] "Instructions"      "Voter Turnout"      "President"
## [4] "House of Rep"      "Co Clerk"           "Co Reg Deeds"
## [7] "Co Public Defender" "Co Comm 1"          "Co Comm 3"
## [10] "Co Comm 5"         "Co Comm 7"          "St Bd of Ed 2"
## [13] "St Bd of Ed 4"     "Legislature 5"      "Legislature 7"
## [16] "Legislature 9"     "Legislature 11"     "Legislature 13"
## [19] "Legislature 23"    "Legislature 31"     "Legislature 39"
## [22] "MCC 1"             "MCC 2"              "MCC 3"
## [25] "MCC 4"             "OPPD"               "MUD"
## [28] "NRD 3"             "NRD 5"              "NRD 7"
## [31] "NRD 9"             "OPS 2"              "OPS 4"
## [34] "OPS 6"             "OPS 8"              "OPS 10"
## [37] "OPS 11"            "OPS 12"             "ESU 2"
## [40] "ESU 3"             "Arlington Sch 24"   "Bennington Sch 59"
## [43] "Elkhorn Sch 10"    "Fremont Sch 1"      "Ft Calhoun Sch 3"
## [46] "Gretna Sch 37"     "Millard Sch 17"     "Ralston Sch 54"
## [49] "Valley Sch 33"     "Waterloo Sch 11"    "Bennington Mayor"
## [52] "Elkhorn Mayor"     "Valley Mayor"       "Ralston Mayor"
## [55] "Ralston Library Bd" "Bennington City Cnc 1" "Bennington City Cnc 2"
## [58] "Elkhorn City Cnc A" "Elkhorn City Cnc B"  "Elkhorn City Cnc C"
## [61] "Ralston City Cnc 1" "Ralston City Cnc 2"  "Ralston City Cnc 6"
## [64] "Waterloo Bd Trustees" "Valley City Cnc"    "Amendment 1"
## [67] "Amendment 2"       "Amendment 3"        "Amendment 4"
## [70] "Initiative 417"    "Initiative 418"     "Initiative 419"
## [73] "Initiative 420"

## Using the 'read_excel' function, read the Voter Turnout sheet
## from the 'data/G04ResultsDetail2004-11-02.xls'
## Assign the data to the 'voter_turnout_df1'
## The header is in the second row, so make sure to skip the first row
## Examine the structure of 'voter_turnout_df1' using 'str()'

voter_turnout_df1 <- read_excel('data/G04ResultsDetail2004-11-02.xls',
                                sheet = 'Voter Turnout', skip = 1)

str(voter_turnout_df1)

## tibble [342 x 4] (S3: tbl_df/tbl/data.frame)
## $ Ward Precinct : chr [1:342] "01-01" "01-02" "01-03" "01-04" ...
## $ Ballots Cast : num [1:342] 421 443 705 827 527 323 358 410 440 500 ...
## $ Registered Voters: num [1:342] 678 691 1148 1308 978 ...
## $ Voter Turnout : num [1:342] 0.621 0.641 0.614 0.632 0.539 ...

```

```

## Using the 'read_excel()' function, read the Voter Turnout sheet
## from 'data/G04ResultsDetail2004-11-02.xls'
## Skip the first two rows and manually assign the columns using 'col_names'
## Use the names "ward_precint", "ballots_cast", "registered_voters", "voter_turnout"
## Assign the data to the 'voter_turnout_df2'
## Examine the structure of 'voter_turnout_df2' using 'str()'
voter_turnout_df2 <- read_excel('data/G04ResultsDetail2004-11-02.xls', sheet = 'Voter Turnout', col_names = col_names)
str(voter_turnout_df2)

## tibble [342 x 4] (S3: tbl_df/tbl/data.frame)
## $ ward_precint      : chr [1:342] "01-01" "01-02" "01-03" "01-04" ...
## $ ballots_cast      : num [1:342] 421 443 705 827 527 323 358 410 440 500 ...
## $ registered_voters : num [1:342] 678 691 1148 1308 978 ...
## $ voter_turnout      : num [1:342] 0.621 0.641 0.614 0.632 0.539 ...

## Load the 'DBI' library
library('DBI')

## Create a database connection to 'data/tidynomicon/example.db' using the dbConnect() function
## The first argument is the database driver which in this case is 'RSQLite::SQLite()'
## The second argument is the path to the database file
## Assign the connection to 'db' variable
db <- dbConnect(RSQLite::SQLite(), 'data/tidynomicon/example.db')

## Query the Person table using the 'dbGetQuery' function and the
## 'SELECT * FROM PERSON;' SQL statement
## Assign the result to the 'person_df' variable
## Use 'head()' to look at the first few rows of the 'person_df' dataframe
person_df <- dbGetQuery(db, 'Select * FROM Person')
head(person_df)

##   person_id personal_name family_name
## 1      dyer      William      Dyer
## 2       pb       Frank    Pabodie
## 3      lake      Anderson      Lake
## 4       roe     Valentina    Roerich
## 5 danforth      Frank    Danforth

## List the tables using the 'dbListTables()' function
## Assign the result to the 'table_names' variable
table_names <- dbListTables(db)

## Read all of the tables at once using the 'lapply' function and assign the result to the 'tables' variable
## Use 'table_names', 'dbReadTable', and 'conn = db' as arguments
## Print out the tables
tables <- lapply(table_names, dbReadTable, conn = db)

## Warning in result_fetch(res@ptr, n = n): Column 'reading': mixed type, first seen values
## of type real, coercing other values of type string

tables

## [[1]]
##   visit_id person_id quantity reading
## 1      619      dyer      rad    9.82

```

```

## 2      619      dyer      sal      0.13
## 3      622      dyer      rad      7.80
## 4      622      dyer      sal      0.09
## 5      734      pb       rad      8.41
## 6      734      lake     sal      0.05
## 7      734      pb       temp    -21.50
## 8      735      pb       rad      7.22
## 9      735      <NA>     sal      0.06
## 10     735      <NA>     temp    -26.00
## 11     751      pb       rad      4.35
## 12     751      pb       temp    -18.50
## 13     751      lake     sal      0.00
## 14     752      lake     rad      2.19
## 15     752      lake     sal      0.09
## 16     752      lake     temp    -16.00
## 17     752      roe      sal      41.60
## 18     837      lake     rad      1.46
## 19     837      lake     sal      0.21
## 20     837      roe      sal      22.50
## 21     844      roe      rad      11.25
##
## [[2]]
##   person_id personal_name family_name
## 1      dyer      William      Dyer
## 2      pb       Frank      Pabodie
## 3      lake      Anderson     Lake
## 4      roe      Valentina    Roerich
## 5  danforth      Frank      Danforth
##
## [[3]]
##   site_id latitude longitude
## 1    DR-1   -49.85   -128.57
## 2    DR-3   -47.15   -126.72
## 3   MSK-4   -48.87   -123.40
##
## [[4]]
##   visit_id site_id visit_date
## 1      619    DR-1 1927-02-08
## 2      622    DR-1 1927-02-10
## 3      734    DR-3 1930-01-07
## 4      735    DR-3 1930-01-12
## 5      751    DR-3 1930-02-26
## 6      752    DR-3      <NA>
## 7      837   MSK-4 1932-01-14
## 8      844    DR-1 1932-03-22

## Use the 'dbDisconnect' function to disconnect from the database
dbDisconnect(db)

## Import the 'jsonlite' library
library('jsonlite')

## Convert the scores_df dataframe to JSON using the 'toJSON()' function
toJSON(scores_df)

```

```

## [{"Count":10,"Score":200,"Section":"Sports"}, {"Count":10,"Score":205,"Section":"Sports"}, {"Count":20,
##
## Convert the scores dataframe to JSON using the 'toJSON()' function with the 'pretty=TRUE' option
toJSON(scores_df, pretty = TRUE)
## [
##   {
##     "Count": 10,
##     "Score": 200,
##     "Section": "Sports"
##   },
##   {
##     "Count": 10,
##     "Score": 205,
##     "Section": "Sports"
##   },
##   {
##     "Count": 20,
##     "Score": 235,
##     "Section": "Sports"
##   },
##   {
##     "Count": 10,
##     "Score": 240,
##     "Section": "Sports"
##   },
##   {
##     "Count": 10,
##     "Score": 250,
##     "Section": "Sports"
##   },
##   {
##     "Count": 10,
##     "Score": 265,
##     "Section": "Regular"
##   },
##   {
##     "Count": 10,
##     "Score": 275,
##     "Section": "Regular"
##   },
##   {
##     "Count": 30,
##     "Score": 285,
##     "Section": "Sports"
##   },
##   {
##     "Count": 10,
##     "Score": 295,
##     "Section": "Regular"
##   },
##   {
##     "Count": 10,
##     "Score": 300,

```

```

##      "Section": "Regular"
##    },
##    {
##      "Count": 20,
##      "Score": 300,
##      "Section": "Sports"
##    },
##    {
##      "Count": 10,
##      "Score": 305,
##      "Section": "Sports"
##    },
##    {
##      "Count": 10,
##      "Score": 305,
##      "Section": "Regular"
##    },
##    {
##      "Count": 10,
##      "Score": 310,
##      "Section": "Regular"
##    },
##    {
##      "Count": 10,
##      "Score": 310,
##      "Section": "Sports"
##    },
##    {
##      "Count": 20,
##      "Score": 320,
##      "Section": "Regular"
##    },
##    {
##      "Count": 10,
##      "Score": 305,
##      "Section": "Regular"
##    },
##    {
##      "Count": 10,
##      "Score": 315,
##      "Section": "Sports"
##    },
##    {
##      "Count": 20,
##      "Score": 320,
##      "Section": "Regular"
##    },
##    {
##      "Count": 10,
##      "Score": 325,
##      "Section": "Regular"
##    },
##    {
##      "Count": 10,

```

```

##      "Score": 325,
##      "Section": "Sports"
##    },
##    {
##      "Count": 20,
##      "Score": 330,
##      "Section": "Regular"
##    },
##    {
##      "Count": 10,
##      "Score": 330,
##      "Section": "Sports"
##    },
##    {
##      "Count": 30,
##      "Score": 335,
##      "Section": "Sports"
##    },
##    {
##      "Count": 10,
##      "Score": 335,
##      "Section": "Regular"
##    },
##    {
##      "Count": 20,
##      "Score": 340,
##      "Section": "Regular"
##    },
##    {
##      "Count": 10,
##      "Score": 340,
##      "Section": "Sports"
##    },
##    {
##      "Count": 30,
##      "Score": 350,
##      "Section": "Regular"
##    },
##    {
##      "Count": 20,
##      "Score": 360,
##      "Section": "Regular"
##    },
##    {
##      "Count": 10,
##      "Score": 360,
##      "Section": "Sports"
##    },
##    {
##      "Count": 20,
##      "Score": 365,
##      "Section": "Regular"
##    },
##    {

```

```
##      "Count": 20,
##      "Score": 365,
##      "Section": "Sports"
##    },
##    {
##      "Count": 10,
##      "Score": 370,
##      "Section": "Sports"
##    },
##    {
##      "Count": 10,
##      "Score": 370,
##      "Section": "Regular"
##    },
##    {
##      "Count": 20,
##      "Score": 375,
##      "Section": "Regular"
##    },
##    {
##      "Count": 10,
##      "Score": 375,
##      "Section": "Sports"
##    },
##    {
##      "Count": 20,
##      "Score": 380,
##      "Section": "Regular"
##    },
##    {
##      "Count": 10,
##      "Score": 395,
##      "Section": "Sports"
##    }
## ]
```

The R session information (including the OS info, R version and all packages used):

```
sessionInfo()

## R version 4.2.2 (2022-10-31)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Monterey 12.5.1
##
## Matrix products: default
## LAPACK: /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods    base
##
## other attached packages:
```



```
## [1] jsonlite_1.8.4 DBI_1.1.3      readxl_1.4.1   knitr_1.41
##
## loaded via a namespace (and not attached):
## [1] Rcpp_1.0.9      magrittr_2.0.3  bit_4.0.5      rlang_1.0.6    fastmap_1.1.0
## [6] fansi_1.0.3     stringr_1.5.0   blob_1.2.3     highr_0.9      tools_4.2.2
## [11] xfun_0.35       utf8_1.2.2      tinytex_0.42   cli_3.4.1      htmltools_0.5.4
## [16] yaml_2.3.6      bit64_4.0.5     digest_0.6.30  tibble_3.1.8   lifecycle_1.0.3
## [21] vctrs_0.5.1     memoise_2.0.1   glue_1.6.2     cachem_1.0.6   RSQLite_2.2.19
## [26] evaluate_0.18   rmarkdown_2.18  stringi_1.7.8  pillar_1.8.1   cellranger_1.1.0
## [31] compiler_4.2.2  pkgconfig_2.0.3

Sys.time()

## [1] "2022-12-11 09:21:31 MST"
```