

```

1 > # Assignment: ASSIGNMENT 2
2 > # Name: Lastname, Firstname
3 > # Date: 2010-02-14
4 >
5 > ## Check your current working directory using `getwd()`
6 > getwd()
7 [1] "C:/Users/Daisy/Documents/Xin/Data science/dsc520"
8 >
9 > ## List the contents of the working directory with the `dir()` function
10 > dir(getwd())
11 [1] "assignments" "completed" "data" "example-code.JPG"
12     "helpful_resource"
13 [6] "LICENSE" "R-notes.txt" "README.md" "RMarkdown.md"
14     "scratch"
15 >
16 > ## If the current directory does not contain the `data` directory, set the
17 > ## working directory to project root folder (the folder should contain the `data`
18 > ## directory
19 > ## Use `setwd()` if needed
20 > setwd("/Users/Daisy/Documents/Xin/Data science/dsc520")
21 >
22 > ## Load the file `data/tidynomicon/person.csv` to `person_df1` using `read.csv`
23 > ## Examine the structure of `person_df1` using `str()`
24 > person_df1 <- read.csv('data/tidynomicon/person.csv')
25 > str(person_df1)
26 'data.frame':   5 obs. of  3 variables:
27 $ person_id : chr  "dyer" "pb" "lake" "roe" ...
28 $ personal_name: chr  "William" "Frank" "Anderson" "Valentina" ...
29 $ family_name : chr  "Dyer" "Pabodie" "Lake" "Roerich" ...
30 >
31 > ## R interpreted names as factors, which is not the behavior we want
32 > ## Load the same file to person_df2 using `read.csv` and setting `stringsAsFactors` to
33 > ## `FALSE`
34 > ## Examine the structure of `person_df2` using `str()`
35 > person_df2 <- read.csv('data/tidynomicon/person.csv', stringsAsFactors=FALSE)
36 > str(person_df2)
37 'data.frame':   5 obs. of  3 variables:
38 $ person_id : chr  "dyer" "pb" "lake" "roe" ...
39 $ personal_name: chr  "William" "Frank" "Anderson" "Valentina" ...
40 $ family_name : chr  "Dyer" "Pabodie" "Lake" "Roerich" ...
41 >
42 > ## Read the file `data/scores.csv` to `scores_df`
43 > ## Display summary statistics using the `summary()` function
44 > scores_df <- read.csv('data/scores.csv', stringsAsFactors=FALSE)
45 > ##str(scores_df)
46 > summary(scores_df)
47      Count      Score      Section
48 Min.   :10.00   Min.   :200.0   Length:38
49 1st Qu.:10.00   1st Qu.:300.0   Class :character
50 Median :10.00   Median :322.5   Mode  :character
51 Mean   :14.47   Mean   :317.5
52 3rd Qu.:20.00   3rd Qu.:357.5
53 Max.   :30.00   Max.   :395.0
54 >
55 > ## Load the `readxl` library
56 > #install.packages("readxl")
57 > library(readxl)
58 >
59 > ## Using the excel_sheets() function from the `readxl` package,
60 > ## list the worksheets from the file `data/G04ResultsDetail2004-11-02.xls`
61 > setwd("/Users/Daisy/Documents/Xin/Data science/dsc520")
62 > library(readxl)
63 > excel_sheets('data/G04ResultsDetail2004-11-02.xls')
64 [1] "Instructions" "Voter Turnout" "President" "House of
65     Rep"
66 [5] "Co Clerk" "Co Reg Deeds" "Co Public Defender" "Co Comm
67     1"
68 [9] "Co Comm 3" "Co Comm 5" "Co Comm 7" "St Bd of
69     Ed 2"

```

```

63 [13] "St Bd of Ed 4" ..... "Legislature 5" ..... "Legislature 7" .....
    "Legislature 9" .....
64 [17] "Legislature 11" ..... "Legislature 13" ..... "Legislature 23" .....
    "Legislature 31" .....
65 [21] "Legislature 39" ..... "MCC 1" ..... "MCC 2" ..... "MCC
    3" .....
66 [25] "MCC 4" ..... "OPPD" ..... "MUD" ..... "NRD
    3" .....
67 [29] "NRD 5" ..... "NRD 7" ..... "NRD 9" ..... "OPS
    2" .....
68 [33] "OPS 4" ..... "OPS 6" ..... "OPS 8" ..... "OPS
    10" .....
69 [37] "OPS 11" ..... "OPS 12" ..... "ESU 2" ..... "ESU
    3" .....
70 [41] "Arlington Sch 24" ..... "Bennington Sch 59" ..... "Elkhorn Sch 10" ..... "Fremont
    Sch 1" .....
71 [45] "Ft Calhoun Sch 3" ..... "Gretna Sch 37" ..... "Millard Sch 17" ..... "Ralston
    Sch 54" .....
72 [49] "Valley Sch 33" ..... "Waterloo Sch 11" ..... "Bennington Mayor" ..... "Elkhorn
    Mayor" .....
73 [53] "Valley Mayor" ..... "Ralston Mayor" ..... "Ralston Library Bd" ..... "Bennington
    City Cnc 1" .....
74 [57] "Bennington City Cnc 2" "Elkhorn City Cnc A" ..... "Elkhorn City Cnc B" ..... "Elkhorn
    City Cnc C" .....
75 [61] "Ralston City Cnc 1" ..... "Ralston City Cnc 2" ..... "Ralston City Cnc 6" ..... "Waterloo
    Bd Trustees" .....
76 [65] "Valley City Cnc" ..... "Amendment 1" ..... "Amendment 2" ..... "Amendment
    3" .....
77 [69] "Amendment 4" ..... "Initiative 417" ..... "Initiative 418" ..... "Initiative
    419" .....
78 [73] "Initiative 420" .....
79 >
80 > ## Using the `read_excel` function, read the Voter Turnout sheet
81 > ## from the `data/G04ResultsDetail2004-11-02.xls`
82 > ## Assign the data to the `voter_turnout_df1`
83 > ## The header is in the second row, so make sure to skip the first row
84 > ## Examine the structure of `voter_turnout_df1` using `str()`
85 >
86 > voter_turnout_df1 <- read_excel('./data/G04ResultsDetail2004-11-02.xls', sheet =
    'Voter Turnout', skip = 1)
87 > str(voter_turnout_df1)
88 tibble [342 × 4] (S3: tbl_df/tbl/data.frame)
89 $ Ward Precinct : chr [1:342] "01-01" "01-02" "01-03" "01-04" ...
90 $ Ballots Cast : num [1:342] 421 443 705 827 527 323 358 410 440 500 ...
91 $ Registered Voters: num [1:342] 678 691 1148 1308 978 ...
92 $ Voter Turnout : num [1:342] 0.621 0.641 0.614 0.632 0.539 ...
93 >
94 > ## Using the `read_excel()` function, read the Voter Turnout sheet
95 > ## from `data/G04ResultsDetail2004-11-02.xls`
96 > ## Skip the first two rows and manually assign the columns using `col_names`
97 > ## Use the names "ward_precint", "ballots_cast", "registered_voters", "voter_turnout"
98 > ## Assign the data to the `voter_turnout_df2`
99 > ## Examine the structure of `voter_turnout_df2` using `str()`
100 > my_col =c("ward_precint", "ballots_cast", "registered_voters", "voter_turnout")
101 > voter_turnout_df2 <- read_excel('./data/G04ResultsDetail2004-11-02.xls', sheet =
    'Voter Turnout', skip = 2, col_names = my_col)
102 > str(voter_turnout_df2)
103 tibble [342 × 4] (S3: tbl_df/tbl/data.frame)
104 $ ward_precint : chr [1:342] "01-01" "01-02" "01-03" "01-04" ...
105 $ ballots_cast : num [1:342] 421 443 705 827 527 323 358 410 440 500 ...
106 $ registered_voters: num [1:342] 678 691 1148 1308 978 ...
107 $ voter_turnout : num [1:342] 0.621 0.641 0.614 0.632 0.539 ...
108 >
109 > ## Load the `DBI` library
110 > ##install.packages("DBI")
111 > library(DBI)
112 > #install.packages("RSQLite")
113 > library(RSQLite)
114 >

```

```

115 >
116 > ## Create a database connection to `data/tidynomicon/example.db` using the dbConnect()
function
117 > ## The first argument is the database driver which in this case is `RSQLite::SQLite()`
118 > ## The second argument is the path to the database file
119 > ## Assign the connection to `db` variable
120 > db <- dbConnect(RSQLite::SQLite(), 'data/tidynomicon/example.db')
121 >
122 > ## Query the Person table using the `dbGetQuery` function and the
123 > ## `SELECT * FROM PERSON;` SQL statement
124 > ## Assign the result to the `person_df` variable
125 > ## Use `head()` to look at the first few rows of the `person_df` dataframe
126 > person_df <- dbGetQuery(db, "SELECT * FROM PERSON")
127 > head(person_df, N = 3)
128   person_id personal_name family_name
129 1 ..... dyer ..... William ..... Dyer
130 2 ..... pb ..... Frank ..... Pabodie
131 3 ..... lake ..... Anderson ..... Lake
132 4 ..... roe ..... Valentina ..... Roerich
133 5 ..... danforth ..... Frank ..... Danforth
134 >
135 > ## List the tables using the `dbListTables()` function
136 > ## Assign the result to the `table_names` variable
137 > table_names <- dbListTables(db)
138 > table_names
139 [1] "Measurements" "Person" ..... "Site" ..... "Visited"
140 > tables
141 [[1]]
142   visit_id person_id quantity reading
143 1 ..... 619 ..... dyer ..... rad ..... 9.82
144 2 ..... 619 ..... dyer ..... sal ..... 0.13
145 3 ..... 622 ..... dyer ..... rad ..... 7.80
146 4 ..... 622 ..... dyer ..... sal ..... 0.09
147 5 ..... 734 ..... pb ..... rad ..... 8.41
148 6 ..... 734 ..... lake ..... sal ..... 0.05
149 7 ..... 734 ..... pb ..... temp ..... -21.50
150 8 ..... 735 ..... pb ..... rad ..... 7.22
151 9 ..... 735 ..... <NA> ..... sal ..... 0.06
152 10 ..... 735 ..... <NA> ..... temp ..... -26.00
153 11 ..... 751 ..... pb ..... rad ..... 4.35
154 12 ..... 751 ..... pb ..... temp ..... -18.50
155 13 ..... 751 ..... lake ..... sal ..... 0.00
156 14 ..... 752 ..... lake ..... rad ..... 2.19
157 15 ..... 752 ..... lake ..... sal ..... 0.09
158 16 ..... 752 ..... lake ..... temp ..... -16.00
159 17 ..... 752 ..... roe ..... sal ..... 41.60
160 18 ..... 837 ..... lake ..... rad ..... 1.46
161 19 ..... 837 ..... lake ..... sal ..... 0.21
162 20 ..... 837 ..... roe ..... sal ..... 22.50
163 21 ..... 844 ..... roe ..... rad ..... 11.25
164
165 [[2]]
166   person_id personal_name family_name
167 1 ..... dyer ..... William ..... Dyer
168 2 ..... pb ..... Frank ..... Pabodie
169 3 ..... lake ..... Anderson ..... Lake
170 4 ..... roe ..... Valentina ..... Roerich
171 5 ..... danforth ..... Frank ..... Danforth
172
173 [[3]]
174   site_id latitude longitude
175 1 ..... DR-1 ..... -49.85 ..... -128.57
176 2 ..... DR-3 ..... -47.15 ..... -126.72
177 3 ..... MSK-4 ..... -48.87 ..... -123.40
178
179 [[4]]
180   visit_id site_id visit_date
181 1 ..... 619 ..... DR-1 1927-02-08
182 2 ..... 622 ..... DR-1 1927-02-10

```

```

183 3 ..... 734 ..... DR-3 1930-01-07
184 4 ..... 735 ..... DR-3 1930-01-12
185 5 ..... 751 ..... DR-3 1930-02-26
186 6 ..... 752 ..... DR-3 ..... <NA>
187 7 ..... 837 ..... MSK-4 1932-01-14
188 8 ..... 844 ..... DR-1 1932-03-22
189
190 > dbDisconnect(db)
191
192 >
193 > ## Import the `jsonlite` library
194 > #install.packages("jsonlite")
195 > library(jsonlite)
196 >
197 > ## Convert the scores_df dataframe to JSON using the `toJSON()` function
198 > toJSON(x = scores_df, dataframe = 'values')
199 [[10,200,"Sports"],[10,205,"Sports"],[20,235,"Sports"],[10,240,"Sports"],[10,250,"Sports"
], [10,265,"Regular"], [10,275,"Regular"], [30,285,"Sports"], [10,295,"Regular"], [10,300,"Reg
ular"], [20,300,"Sports"], [10,305,"Sports"], [10,305,"Regular"], [10,310,"Regular"], [10,310,
"Sports"], [20,320,"Regular"], [10,305,"Regular"], [10,315,"Sports"], [20,320,"Regular"], [10,
325,"Regular"], [10,325,"Sports"], [20,330,"Regular"], [10,330,"Sports"], [30,335,"Sports"], [
10,335,"Regular"], [20,340,"Regular"], [10,340,"Sports"], [30,350,"Regular"], [20,360,"Regula
r"], [10,360,"Sports"], [20,365,"Regular"], [20,365,"Sports"], [10,370,"Sports"], [10,370,"Reg
ular"], [20,375,"Regular"], [10,375,"Sports"], [20,380,"Regular"], [10,395,"Sports"]]
200 >
201 > ## Convert the scores dataframe to JSON using the `toJSON()` function with the
`pretty=TRUE` option
202 > toJSON(x = scores_df, dataframe = 'values', pretty = TRUE)
203 [
204   [10, 200, "Sports"],
205   [10, 205, "Sports"],
206   [20, 235, "Sports"],
207   [10, 240, "Sports"],
208   [10, 250, "Sports"],
209   [10, 265, "Regular"],
210   [10, 275, "Regular"],
211   [30, 285, "Sports"],
212   [10, 295, "Regular"],
213   [10, 300, "Regular"],
214   [20, 300, "Sports"],
215   [10, 305, "Sports"],
216   [10, 305, "Regular"],
217   [10, 310, "Regular"],
218   [10, 310, "Sports"],
219   [20, 320, "Regular"],
220   [10, 305, "Regular"],
221   [10, 315, "Sports"],
222   [20, 320, "Regular"],
223   [10, 325, "Regular"],
224   [10, 325, "Sports"],
225   [20, 330, "Regular"],
226   [10, 330, "Sports"],
227   [30, 335, "Sports"],
228   [10, 335, "Regular"],
229   [20, 340, "Regular"],
230   [10, 340, "Sports"],
231   [30, 350, "Regular"],
232   [20, 360, "Regular"],
233   [10, 360, "Sports"],
234   [20, 365, "Regular"],
235   [20, 365, "Sports"],
236   [10, 370, "Sports"],
237   [10, 370, "Regular"],
238   [20, 375, "Regular"],
239   [10, 375, "Sports"],
240   [20, 380, "Regular"],
241   [10, 395, "Sports"]
242 ]
243 >

```