The University of Texas at Austin
School of Information

Stats: More R & Quarto

Mick McQuaid

2024-05-11

Week THREE

Week 2 Exercises

No pre-grading

You shouldn't ask me to look over your homework and make sure that everything is okay. That is because it amounts to pre-grading. If I say it looks okay and then you turn it in and then I take points off for something I didn't previously notice, you will object, saying "But you said it was okay." Therefore I don't want to get into this kind of situation.

On the other hand, it's okay to come to me with vague questions, like "I don't get question two." That opens up the possibility of explaining it better.

Removing instructions

You should remove the instructions from the file you turn in. That means remove the first two paragraphs and the last sentence. Leave the questions in and interleave the questions with your answers.

Texas has the most mortgages

Several people said that California did, even though California only leads in rentals.

Leave a blank line before headings

One person formatted the document incorrectly, leading to a heading not appearing as a heading. You should always review the work before you turn it in.

Name the files as I ask

I will take off points in future for files not correctly named.

Turn in the assignment on time!

I will go over the homework on the Monday after it's due, so I can't accept late submissions after that. If I accept something between Friday and Monday, it will be with a substantial penalty.

Scores

```
1 score ← c(2, 1.9, 1.9, 2, 1.9, 1.9, 2, 2, 1.9, 1.9, 1.9)
2 stem(score)
```

```
The decimal point is 1 digit(s) to the left of the |

19 | 0000000
19 |
20 | 0000
```

More on scores

- I went easy on you if you turned something in
- Lesson: always turn something in
- It will be harsher in future, though, as I expect more and more of you

Solutions

Look at the solution file! There are a lot of tips there!

More on R

Loading the project data:

```r
pacman::p_load(tidyverse)
df ← read_csv(paste0(Sys.getenv("STATS_DATA_DIR"),"/amesHousing2011.csv"))
# df ← read_csv("amesHousing2011.csv")
str(df)
```

```
spc_tbl_ [2,925 × 82] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ Order         : num [1:2925] 1498 2738 2446 2667 2451 ...
 $ PID           : chr [1:2925] "0908154080" "0905427030" "0528320060"
"0902400110" ...
 $ MSSubClass    : chr [1:2925] "020" "075" "060" "075" ...
 $ MSZoning      : chr [1:2925] "RL" "RL" "RL" "RM" ...
 $ LotFrontage   : num [1:2925] 123 60 118 90 114 87 NA 60 60 47 ...
 $ LotArea       : num [1:2925] 47007 19800 35760 22950 17242 ...
 $ Street        : chr [1:2925] "Pave" "Pave" "Pave" "Pave" ...
 $ Alley         : chr [1:2925] NA NA NA NA ...
 $ LotShape      : chr [1:2925] "IR1" "Reg" "IR1" "IR2" ...
 $ LandContour   : chr [1:2925] "Lvl" "Lvl" "Lvl" "Lvl" ...
 $ Utilities     : chr [1:2925] "AllPub" "AllPub" "AllPub" "AllPub" ...
 $ LotConfig     : chr [1:2925] "Inside" "Inside" "CulDSac" "Inside" ...
 $ LandSlope     : chr [1:2925] "Gtl" "Gtl" "Gtl" "Gtl" ...
```

Note on loading data

- Two main ways, depending on input file
- The `load()` function *creates* a data frame
- The `read_csv()` function *returns* a data frame
- To create a data frame with `read_csv`, you must read it into a variable, e.g., `df←read_csv("filename")`! If you just say `read_csv("filename")`, it will display the data frame, but not save it
- It is a mistake to say `df ← load("filename")` or to say `read_csv("filename")`

Convert many columns to factors

Some columns should simply be removed, such as Order and PID. Others are useful as factors. How to tell?

```
1  with(df,table(MSSubClass))
```

```
MSSubClass
 020  030  040  045  050  060  070  075  080  085  090  120  150  160  180  190
1078  139    6   18  287  571  128   23  118   48  109  192    1  129   17   61
```

Use it in conjunction with `amesHousing2011doc.txt`.

Another way, using Tidyverse

```
1 df ▷ count(MSSubClass,sort=TRUE)
```

```
# A tibble: 16 × 2
   MSSubClass     n
   <chr>      <int>
 1 020         1078
 2 060          571
 3 050          287
 4 120          192
 5 030          139
 6 160          129
 7 070          128
 8 080          118
 9 090          109
10 190           61
11 085           48
12 075           23
```

R is changing and the Tidyverse is changing

- But they are changing at different rates
- Tidyverse is changing faster than base R
- Implies that many StackOverflow answers for Tidyverse are outdated
- You must learn to read cryptic error messages about deprecation

Tidyverse has an updated website

https://www.tidyverse.org

Tidyverse consists of packages, listed at https://www.tidyverse.org/packages/

Tidyverse package website lists several sections for learning: Installation and use, Core tidyverse, Import, Wrangle, and others.

Visit dplyr

The dplr package includes the following functions for data manipulation

- `mutate()` adds new columns that are functions of existing columns
- `select()` picks columns based on their names.
- `filter()` picks rows based on their values.
- `summarise()` reduces multiple values down to a single summary.
- `arrange()` changes the ordering of the rows.

Example of selecting all but a few columns

```
1 (dfReduced ← df ▷ select(!c(PID,Order)))
```

```
# A tibble: 2,925 × 80
   MSSubClass MSZoning LotFrontage LotArea Street Alley LotShape LandContour
   <chr>      <chr>           <dbl>   <dbl> <chr>  <chr> <chr>    <chr>
 1 020        RL                123   47007 Pave   <NA>  IR1      Lvl
 2 075        RL                 60   19800 Pave   <NA>  Reg      Lvl
 3 060        RL                118   35760 Pave   <NA>  IR1      Lvl
 4 075        RM                 90   22950 Pave   <NA>  IR2      Lvl
 5 060        RL                114   17242 Pave   <NA>  IR1      Lvl
 6 075        RM                 87   18386 Pave   <NA>  Reg      Lvl
 7 050        RL                 NA   14100 Pave   <NA>  IR1      Lvl
 8 190        RH                 60   10896 Pave   Pave  Reg      Bnk
 9 060        RL                 60   18062 Pave   <NA>  IR1      HLS
10 060        RL                 47   53504 Pave   <NA>  IR2      HLS
# ℹ 2,915 more rows
# ℹ 72 more variables: Utilities <chr>, LotConfig <chr>, LandSlope <chr>,
#   Neighborhood <chr>, Condition1 <chr>, Condition2 <chr>, BldgType <chr>
```

Example of several dplyr functions

```
 1 dfClasses ← read_tsv("classes.tsv")
 2 (dfPriceByClass ← df ▷
 3   select(c(MSSubClass,SalePrice)) ▷
 4   group_by(MSSubClass) ▷
 5   summarize(avPriceByClass = mean(SalePrice),n=n()) ▷
 6   arrange(desc(avPriceByClass)) ▷
 7   inner_join(dfClasses))
```

```
# A tibble: 16 × 4
   MSSubClass avPriceByClass     n subClassDescr
   <chr>                <dbl> <int> <chr>
 1 060                237810.   571 2-STORY 1946 & NEWER
 2 120                208019.   192 1-STORY PUD (Planned Unit Development) -
194…
 3 075                199978.    23 2-1/2 STORY ALL AGES
 4 020                187359.  1078 1-STORY 1946 & NEWER ALL STYLES
 5 080                168009.   118 SPLIT OR MULTI-LEVEL
 6 070                156526.   128 2-STORY 1945 & OLDER
 7 085                149842.    48 SPLIT FOYER
 8 150                148400      1 1-1/2 STORY PUD - ALL AGES
 9 040                144917.     6 1-STORY W/FINISHED ATTIC ALL AGES
10 090                139809.   109 DUPLEX - ALL STYLES AND AGES
11 050                137433.   287 1-1/2 STORY FINISHED ALL AGES
```

More on Quarto

What is Quarto?

- A document production system
- A way to conduct *reproducible research*
- A way to practice *literate programming*

Naming convention for Quarto files

By default, Quarto files end in `.qmd`, although other extensions will work. When you feed a `.qmd` file to RStudio, it assumes that it's a Quarto file and opens it accordingly.

Contents of a Quarto file

A quarto file just contains plain text, no binary information. It can be read by any text editor, although what they do with it depends on how Quarto-aware the editor is.

For example, an R chunk (prefaced by a blank line followed by three backticks and `r` in curly braces) is assumed to be R code. It is syntax-highlighted as R and, in some editors such as RStudio, can be independently executed. In RStudio this is done by clicking a green triangle to the right of the chunk.

Everything not in a code chunk is assumed to be Pandoc-flavored Markdown.

Pandoc-flavored Markdown

Since Markdown was invented around 2004, many flavors of it have developed. The one we're using is the one interpreted by the program `pandoc`, documented at https://pandoc.org/.

Why so many flavors?

Markdown was originally devised as a shorthand for HTML by a person (Jon Gruber) who was tired of having to write out lengthy HTML constructs for his blog. He wanted something simpler but also readable on its own. By the way, the original Markdown description is still on the web after all these years at https://daringfireball.net/projects/markdown/, although there are many more descriptive sites. What happened in the years since was that (A) people wanted their own shorthand sets, and (B) it turned out to be really easy to write a converter from Markdown to HTML.

Some of Markdown's features

You have experienced some of Markdown's features, such as a blank line followed by two hashtags followed by a space for a level two heading. You might have surrounded a word by asterisks to italicize it, or double asterisks to bold-face it. You might have used straight quotation marks and found them converted to typographical quotation marks (a different opening and closing mark).

An important extension: inline code

You can write inline code in Markdown chunks! Use a single backtick, followed by an `r` in curly braces, then the code, then a single backtick. Hence, you can let the data speak instead of laboriously running the file and extracting results from R chunks and manually inserting them into a Markdown chunk.

Other extensions

URLs can be included in Markdown chunks by saying `[displayname](actualURL)`. I usually make the `displayname` be the actual URL, but you can put anything you want in the `displayname` brackets.

Pictures can be included by saying

`![caption](filename)`

on a line by itself.

END

Colophon

This slideshow was produced using `quarto`

Fonts are *Roboto Condensed Bold*, *JetBrains Mono Nerd Font*, and *STIX2*