03-R Markdown-demo

Karl Benedict

3/3/2022

R Markdown and R

Abstract

Workshop tutorial with hands-on demonstration of Markdown, RStudio interface, and R Markdown structure and rendering.

keywords: markdown, RStudio, R, literate computing

R Markdown Tips

- R Markdown Reference Guide

R Markdown files are written and compiled into the target export format in a two step process. While in Jupyter notebooks you can render individual cells of Markdown text by "executing" them, with R Markdown you develop your R Markdown document and then render the whole document into one or more output files.

R Markdown files are broken into chunks of Markdown and executable code, with the code code starting with `` ```{r} `` and ending with `` ``` ``. The language that should be used to interpret the code chunk (R in our case) is provided in the `{}`. In RStudio you can manually type in the start and end elements of a code chunk, or use the apprpriate keyboard shortcut for your operating system (option-command-I on the Mac).

You can execute individual lines of R code or individual code blocks without rendering the whole document.

- *command-enter* (Mac) to execute the current line of code or the selected lines of code
- *command-shift-enter* (Mac) to execute the current chunk/block of code

Here is a sample code block in which three mathematical calculations are performed (including a comment that highlights that commenting within code blocks is a good practice)

```r
# Use comments to add text or descriptive info to a code cell.

196 * 786527
```

```
## [1] 154159292
```

```r
261876 / 19871987
```

```
## [1] 0.01317815
```

```r
3**88
```

```
## [1] 9.697737e+41
```

```r
# using print functions instead of default output for calculation results

print(196 * 786527)
```

```
## [1] 154159292
```

```
print(261876 / 19871987)
```

```
## [1] 0.01317815
```

```
print(3**88)
```

```
## [1] 9.697737e+41
```

How is the output of these two variations on the calculation different from the output from the comparable calculations in the Jupyter Notebook?

> **R Markdown rendering of code chunks is line by line, with the default behavior of displaying first the R command and then the output of that command as output blocks in the editor, and output cells in the generated output file**

You can modify this default behavior by using options in the start of the code chunk:

the `echo=FALSE` option:

```
## [1] 154159292
```

```
## [1] 0.01317815
```

```
## [1] 9.697737e+41
```

the `collapse=TRUE` option:

```
# Collapse the input and output into a single output cell with the "collapse=TRUE" option

196 * 786527
## [1] 154159292
261876 / 19871987
## [1] 0.01317815
3**88
## [1] 9.697737e+41
```

the `results=hide` option:

```
# Just show the source code without the output using the "results='hide'" option

196 * 786527
261876 / 19871987
3**88
```

combining the `results=hide, echo=FALSE` options:

Variables and Sequence of Execution

Like Jupyter notebooks the sequence of execution controls the values of R objects at any time in your work with your R Markdown document. **But**, when you `knit` your R Markdown document into a rendered file, the source document is always executed anew from beginning to end, ignoring any changes you have made by executing individual lines or chunks of code.

Inserting R Variables and Code in Markdown Blocks

One powerful capability that R Markdown provides is the insertion of values of R variables or the output of R commands into Markdown chunks.

```
# Assigning output to variables

a <- 196 * 786527
b <- 261876 / 19871987
c <- 3**88
```

Insertion of values associated with variables or executing R code within Markdown chunks is done by placing the variable or R code into an embedded execution block within your Markdown. The execution block is wrapped in back-tick characters (`` ` ``) and start with the language interpreter that should be used. For example `` ` r 2+3 ` `` (without the space before the `r`) which is rendered as: 5.

Embedding R variables follows this model as well as with:

- Variable `a` = 1.5415929×10^8
- Variable `b` = 0.0131781
- Variable `c` = 9.6977373×10^{41}

Working with Real Data

For this demo we will be using data from Albuquerque's open data portal. The dataset is the *City Checkbook*, which includes a list of invoices paid to vendors:

> City of Albuquerque, Accounts Payable Section, Accounting Division of the Department of Financial and Administrative > Services (2021). *City Checkbook* http://data.cabq.gov/government/vendorcheckbook/VendorCheckBookCABQ-en-us.csv

What follows is a rough outline or sketch of an example workflow for developing and reporting an analysis using R and R Markdown. This document is a demonstration and should not be taken as a robust analysis of spending trends by the city of Albuquerque during the COVID pandemic.

Generally, we will address the following questions:

1. Did city spending increase of decrease during the pandemic, compared to the year before?
2. Were specific vendors impacted more than others?
3. Did the pandemic affect the interval between when an invoice was billed and when it was paid?

Methods

In this section we would describe methods relative to:

- sampling
- data collection
- data cleaning or quality assurance
- analysis

The notebook environment allows us to demonstrate some of these processes interactively (and openly!).

Importing R Packages

Most R analyses depent upon functionality provided by packages outside of the base R environment. We can import the `tidyverse` library that contains the additional functionality required by this sample analysis by executing the `library(tidyverse)` command.

```
# the inclusion of the "message=FALSE" option suppresses the display/generation of the output
# of the library function
library(tidyverse)
```

Import the data

We can then import the dataset from Albuquerque's data site. In this case we retain the messages generated by the execution of the `read_csv` command so we can see the variable names and types for the imported CSV file.

```
# use the read_csv function to read the locally stored CSV file
ckbk <- read_csv("./data/abq_vendor_data_2019-2021.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   NAME1 = col_character(),
##   `PAYMENT REFERENCE NUMBER` = col_double(),
##   `INVOICE NUMBER` = col_character(),
##   `INVOICE DATE` = col_date(format = ""),
##   `PAYMENT DATE` = col_date(format = ""),
##   `INVOICE AMOUNT` = col_double(),
##   invoice_year = col_double(),
##   invoice_month = col_double(),
##   payment_year = col_double(),
##   payment_month = col_double(),
##   billed_duration = col_double()
## )
```

Looking at the data in table form

R provides a useful default display of the resulting table within the editor.

```
ckbk
```

```
## # A tibble: 456,066 x 11
##    NAME1       `PAYMENT REFERENC~ `INVOICE NUMBER` `INVOICE DATE` `PAYMENT DATE`
##    <chr>                    <dbl> <chr>            <date>         <date>
##  1 1 ST HEALT~            9411323 NMSM4976-110320~ 2020-11-03     2020-12-08
##  2 10 TANKER ~            9414239 TCA122320        2020-12-23     2021-01-06
##  3 101 PROPER~            2668912 SMLL_BUS_GRNT_2~ 2020-12-02     2020-12-04
##  4 110 SUNPOR~            2668724 SMLL_BUS_GRNT_1~ 2020-11-18     2020-11-19
##  5 13TH JUDIC~            2670551 05212021         2021-05-21     2021-05-21
##  6 1ST AND 10~            2668387 FA0124601_858452 2020-10-08     2020-10-14
##  7 1ST PREMIE~            9368532 FCS2901019       2019-10-10     2019-10-31
##  8 1ST PREMIE~            9368532 FCS2901019       2019-10-10     2019-10-31
##  9 1ST PREMIE~            9370275 fcs2901108       2019-11-08     2019-11-15
## 10 1ST PREMIE~            9370275 fcs2901108       2019-11-08     2019-11-15
## # ... with 456,056 more rows, and 6 more variables: INVOICE AMOUNT <dbl>,
## #   invoice_year <dbl>, invoice_month <dbl>, payment_year <dbl>,
## #   payment_month <dbl>, billed_duration <dbl>
```

But we will want to use one of the available additional packages for rendering the imported *dataframe* as a nice printed table. As highlighted in the tablessection of the R Markdown tutorial from RStudio there are a number of packages you might use. We will be using `knitr`'s `kable` package to generate a table for our HTML document.

First we need to import the `kable` package

```
library(knitr)
```

... and then we can use it to render the imported data as a nicely formatted HTML table

Table 1: A sample of the rows and columns from the ABQ dataset

NAME1	PAYMENT REFERENCE NUMBER	INVOICE NUMBER
1 ST HEALTH INC	9411323	NMSM4976-110320FMV
10 TANKER AIR CARRIER	9414239	TCA122320
101 PROPERTY, LLC	2668912	SMLL_BUS_GRNT_236
110 SUNPORT LLC DBA HOLIDAY	2668724	SMLL_BUS_GRNT_146
13TH JUDICIAL DISTRICT COURT	2670551	05212021

Generating some statistics

We often want to calculate descriptive statistics for some or all of our variables. We can use variations of the `summarise` function to generate selected statistics for numeric data columns.

```
summaryStats <- ckbk %>%
  select("INVOICE AMOUNT", "billed_duration") %>%
  summarise_all(list(mean = mean, sd = sd, min = min, max = max))
kable(t(summaryStats))
```

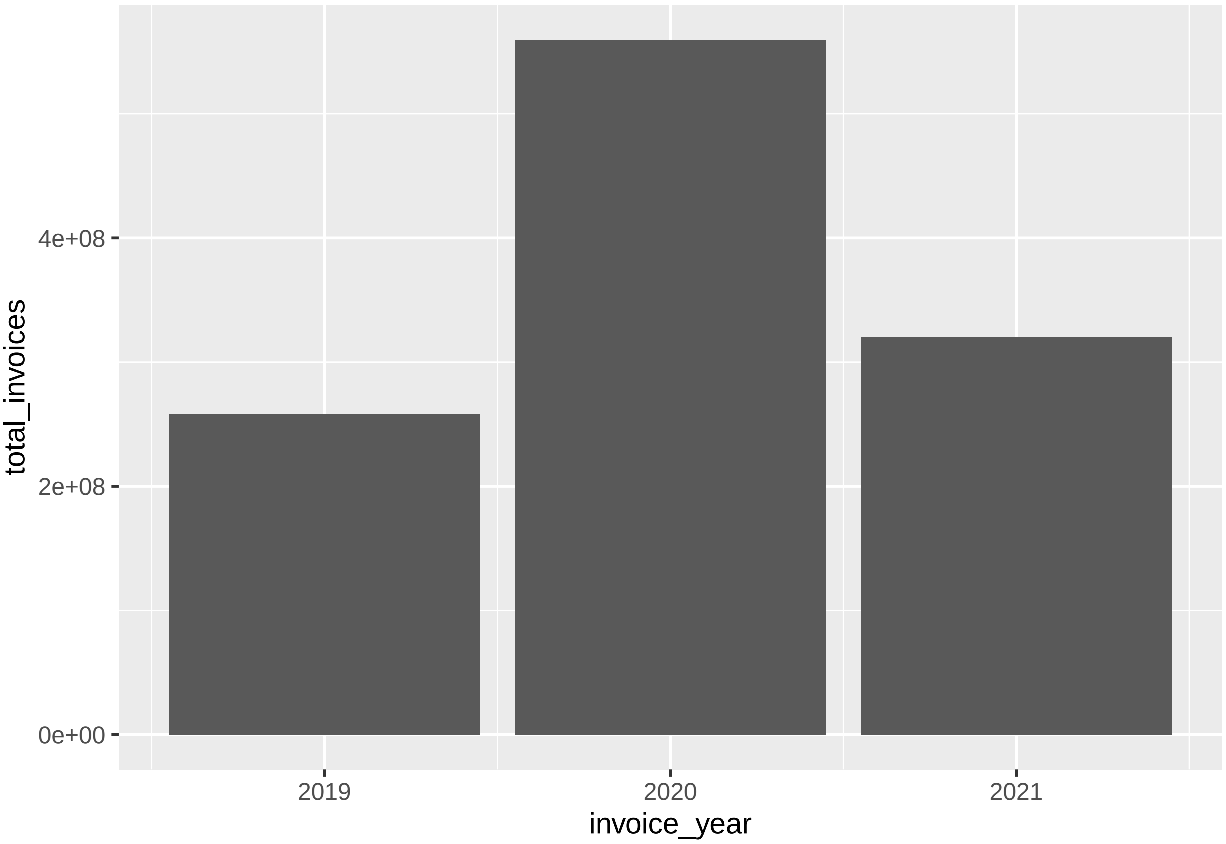
INVOICE AMOUNT_mean	2494.4168
billed_duration_mean	26.2630
INVOICE AMOUNT_sd	38739.9936
billed_duration_sd	39.9638
INVOICE AMOUNT_min	-204612.1000
billed_duration_min	-259.0000
INVOICE AMOUNT_max	9081197.2100
billed_duration_max	795.0000

We can also calculate descriptive statistics for groups of data.

Summary by Invoice Year in a table

```
# summarize by invoice year
summaryStatsByYear <- ckbk %>%
  select("invoice_year","INVOICE AMOUNT", "billed_duration") %>%
  group_by(invoice_year) %>%
  summarise(ct_invoices = n(),
            total_invoices = sum(`INVOICE AMOUNT`))
kable(summaryStatsByYear)
```

invoice_year	ct_invoices	total_invoices
2019	122267	258238434
2020	221375	559475468
2021	112424	319904772
Summary by Invo	ice Year in a	graph



Summary by Vendor

Table 4: Sample of vendor statistics

Vendor	ct_inv	sum_inv	sum_ttp	avg_inv	avg_ttp
1 ST HEALTH INC	1	34.25	35	34	35
10 TANKER AIR CARRIER	1	7635.13	14	7635	14
101 PROPERTY, LLC	1	10000.00	2	10000	2
110 SUNPORT LLC DBA HOLIDAY	1	10000.00	1	10000	1
13TH JUDICIAL DISTRICT COURT	1	6200.00	0	6200	0
1ST AND 10 CONCESSIONS	1	120.00	6	120	6
1ST PREMIER HOME CARE INC	53	225173.75	790	4249	15
2 BEAR PAINTING DBA WEEMS	1	10000.00	2	10000	2

Vendor	ct_inv	sum_inv	sum_ttp	avg_inv	avg_ttp
2000 VIETNAM RESTAURANT	1	700.00	6	700	6
208 A/B/C & 2012 A/B/C SAN PABLO ST NE	1	580.00	7	580	7
2424 BRODWAY LLC	1	510.00	8	510	8
2540 GROUP FORMERLY	1	10000.00	1	10000	1
328 CHINESE CUISINE	1	455.20	6	455	6
370 BOSQUE	1	500.00	26	500	26
3B ELECTRICAL LLC	1	10000.00	1	10000	1
3B YOGA	1	10000.00	1	10000	1
3D GLASS SOLUTIONS INC	1	58912.00	21	58912	21
3DKUTZ BARBERSHOP	1	10000.00	2	10000	2
3M COMPANY	48	173692.12	1979	3619	41
4 DAUGHTERS LAND & CATTLE CO INC	4	50124.60	133	12531	33

From this summary by vendor we can extract and present subsets of the vendors that meet specific criteria. For example, selecting and printing the top 20 vendors by total amount invoiced.

```
top20Vendors <- summaryStatsByVendor %>%
  arrange(desc(sum_inv)) %>%
  top_n(20)
kable(top20Vendors)
```

Vendor	ct_inv	sum_inv	sum_ttp	avg_inv	avg_ttp
WS ACQUISITION LLC	1	4304.92	384	4305	384
PEABODY LLC	10	3228.42	4346	323	435
SOLIS JAVIER & ROSELINDA	1	2651.58	428	2652	428
YUKIYO KAWANO	1	2000.00	407	2000	407
MOODYS INVESTORS SERVICE INC	1	700.00	623	700	623
CENTRAL STATION LLC (ALARM MONITOR)	11	582.89	4847	53	441
SPORTS MEDICINE RESEARCH & TESTING LAB	1	450.00	482	450	482
MATTHEW MEADOW NEIGHBORHOOD ASSOCIATION	1	428.68	522	429	522
HENRY DOORLY ZOO	1	425.21	527	425	527
HOSPITALIST MEDICINE PHYSICIANS	2	352.02	820	176	410
KATYA GOMEZ	1	150.00	526	150	526
JOHN M NARANJO	8	132.00	3049	16	381
SANTA FE IMAGING LLC	1	127.07	541	127	541
ASHELY GIBSON	1	75.00	391	75	391
ASHLEY GIBSON	1	75.00	387	75	387
CAMERON MARTINEZ	1	75.00	404	75	404
CINDY SISNEROS	1	66.00	490	66	490

Vendor	ct_inv	sum_inv	sum_ttp	avg_inv	avg_ttp
ADVANCE FRESH CONCEPTS FRANCHISE	1	59.64	380	60	380
CHRISTOPHER ROVETO MD	1	25.00	410	25	410
PRIME DIGITAL RADIOLOGY PC	1	23.20	599	23	599