Module 3 – lesson 06: Building a Document Template – part 2

Script

This lesson picks up where we left off in creating the steak survey article. Go ahead and log back into your Github account and open your Module3 repository. If you need to, start RStudio and open the Module3 project and open your “steakArticle.Rmd” R markdown document.

At this point, we have configured the YAML header the way we want it and we have customized the first R code chunk to set up the environment necessary for the document.

Let’s organize the content for the rest of the document. For this document template we will create 4 main sections:

A Background section that describes the steak survey dataset

A section that describes the purpose of article to describe steak preferences by gender within a specific region of the United States

A third sections that provides summary tables for the demographics of the respondents in the steak survey dataset

And a final section that provides a plot of steak preferences by gender for the region specified in the document.

For the first background section, refer to your read ahead materials to type in the following level 2 header and paragraph of text into your document. You’ll notice that there are 2 web links provided to refer the reader to the original journal article published by 538.com on the steak survey.

[THIS SECTION WILL BE CUT AND PASTED IN – LEAVE THIS SECTION OFF OF TELE-PROMPTER]

## Background

In May 2014, Walt Hickey at [538.com](http://fivethirtyeight.com/) published an article entitled ["How Americans Like Their Steak"](https://fivethirtyeight.com/features/how-americans-like-their-steak/). This article utilized survey data collected from 550 people which asked questions related to various risky activities, such as whether they speed in traffic, if they smoke or go skydiving, and if they prefer a riskier lottery. The survey also asked if they eat steak and if so, how they prefer their steak.

Go ahead and click Save and KNIT to HTML to view the results. HERE IS A TIP – in general as you develop a document template, it is a good idea to make smaller incremental changes in terms of R code added to a document or sections of text. So, after you add an R code chunk or add more code to an R code chunk or add major section of text, be sure to save and KNIT your document. That way it will be easier to find errors and correct them before you end up with a really long document trying to go back and figure out where the errors are occurring.

Go ahead and put in the second section for the Purpose of the article – cut and paste in from the read ahead materials.

[THIS SECTION WILL BE CUT AND PASTED IN – LEAVE THIS SECTION OFF OF TELE-PROMPTER]

## Purpose

The "steak survey" dataset that was used for the ["How Americans Like Their Steak"](https://fivethirtyeight.com/features/how-americans-like-their-steak/) article, is available through the R package [`fivethirtyeight`](https://cran.r-project.org/web/packages/fivethirtyeight/index.html). Using this dataset, this article summarizes the steak preferences for the \_\*\*Mountain\*\*\_ Region of the United States by gender.

For our summary presented here, a subset of the original steak survey was extracted to only include responses with no missing data and only included respondents from the \_\*\*Mountain\*\*\_ region of the US. The data subset used for our summary had `r nrow(sdat)` respondents.

This second section has R markdown syntax we’ve seen before – backtick marks around the fivethirtyeight package name and syntax for inserting website links. I have also highlighted the word Mountain in both bold and italics formatting. In the final sentence, there is a short inline section of R code which computes the number of rows of the sdat dataset using the nrow function. When this runs the number of rows will be inserted – you will not see the R code in the final document.

Now that we’ve added this next section of text, go ahead and save and KNIT your document to check that everything is working as expected.

Go ahead and add the third section of text and the second R code chunk from the read ahead materials.

[THIS SECTION WILL BE CUT AND PASTED IN – LEAVE THIS SECTION OFF OF TELE-PROMPTER]

## Demographic Summary Tables

The age categories of the `r nrow(sdat)` survey respondents were:

```{r}

# create table of the age categories

tb <- sdat %>%

select(age) %>%

table() %>%

prop.table()\*100

# convert table to data frame

dt <- as.data.frame(tb)

# use kable from knitr package to make table

# and use kable styling from kableExtra package

knitr::kable(dt, format="html",

col.names=c("Ages","%"),

digits=2,

caption="Ages of Survey Respondents") %>%

kableExtra::kable\_styling(bootstrap\_options = "striped",

full\_width=FALSE)

```

Let’s walk through this next section of R code. Using the dplyr workflow that I’ve mentioned before which is part of the tidyverse package, we will use the pipes syntax of the %>% commands to add functions on top of each other to follow the data processing workflow easier. In the first set of sequential R commands we are creating a basic table object called tb which is made from the sdat data subset we created earlier in the first code chunk

To make the table object “tb” the age variable (column) is selected from the sdat data subset and piped into the table function which summarizes the number of respondents who selected each age category and that resulting table is piped into the prop.table function which computes the relative proportions of the responses. Finally each of those proportion are multiplied by 100 to computer percents instead.

The next line of code converts the tb table object to a data frame object which works better with the knitr kable function.

Then the dt data frame is used in the knitr kable function to create a table. There are several options added to the kable function statement – the format is set to HTML – the 2 column names are provided – the number of digits to be displayed after the decimal point are set to 2 for the table and a caption was added for the table. After the knitr kable statement, some kable styling is added using the kableExtra package adding striped rows and setting the option full\_width to false so the table will NOT span the whole width of the resulting HTML webpage.

Save the document and KNIT to HTML to see the results for this newly added section and R code chunk.

Let’s add the next section of text and the next R code chunk for the second table summarizing the education levels for the respondents in the dataset. You’ll notice that this code is pretty much identical to what we did for summarizing the age categories above – the only differences are the variable has changed to educ and the column name and caption have been updated.

[THIS SECTION WILL BE CUT AND PASTED IN – LEAVE THIS SECTION OFF OF TELE-PROMPTER]

The education levels of the `r nrow(sdat)` survey respondents were:

```{r}

# repeat for education

tb <- sdat %>%

select(educ) %>%

table() %>%

prop.table()\*100

dt <- as.data.frame(tb)

knitr::kable(dt, format="html",

col.names=c("Education","%"),

digits=2,

caption="Education of Survey Respondents") %>%

kableExtra:: kable\_styling(bootstrap\_options = "striped",

full\_width=FALSE)

```

Save the document and KNIT to HTML to make sure everything worked with this added section and next R code chunk.

Add the last section of text and R code for the third summary table for income levels of the respondents. Again this section and code is the same as the above sections except the code is run for the hhold\_income variable with updated column labels and caption.

[THIS SECTION WILL BE CUT AND PASTED IN – LEAVE THIS SECTION OFF OF TELE-PROMPTER]

The income levels of the `r nrow(sdat)` survey respondents were:

```{r}

# repeat for household income

tb <- sdat %>%

select(hhold\_income) %>%

table() %>%

prop.table()\*100

dt <- as.data.frame(tb)

knitr::kable(dt, format="html",

col.names=c("Income","%"),

digits=2,

caption="Income of Survey Respondents") %>%

kableExtra:: kable\_styling(bootstrap\_options = "striped",

full\_width=FALSE)

```

Save your document and KNIT to HTML to view results and check for any errors.

The next section of text and R code is provided for your future reference. It is optional and you can skip this step now if you wish to do so. Making tables using R code with R markdown and knitr functions is challenging. So, this next R code chunk provides an example of how to merge the 3 tables we created above into 1 single data frame object which is then used by the knitr kable function to make a larger table. This larger table now has the relative percents of the respondents for each of the categories for age, education and income. The kableExtra styling adds the row striping and adds borders to the table. And one additional row of code is shown that uses the kableExtra add\_header\_above to add another row of column names with the labels for age, education and income.

To learn more about these functions, refer to the read ahead supporting materials for merging data frames and the kableExtra functions.

[THIS SECTION WILL BE CUT AND PASTED IN – LEAVE THIS SECTION OFF OF TELE-PROMPTER]

## \_OPTIONAL\_ Demographics of Survey Respondents in a Merged Table

This section is provided as an example of how to merge the three demographic summary tables for the `r nrow(sdat)` survey respondents above into a single merged table.

```{r}

# create table summary for ages

tb1 <- sdat %>%

select(age) %>%

table() %>%

prop.table()\*100

# create table summary for education

tb2 <- sdat %>%

select(educ) %>%

table() %>%

prop.table()\*100

# create table summary for income

tb3 <- sdat %>%

select(hhold\_income) %>%

table() %>%

prop.table()\*100

# convert all tables to data frames

tb1df <- as.data.frame(tb1)

tb2df <- as.data.frame(tb2)

tb3df <- as.data.frame(tb3)

# merge 1st 2 data frames together

mtb <- merge(data.frame(tb1df, row.names = NULL),

data.frame(tb2df, row.names = NULL),

by=0, all=TRUE)[-1]

# merge result with 3rd data frame

mtb2 <- merge(data.frame(mtb, row.names = NULL),

data.frame(tb3df, row.names = NULL),

by=0, all=TRUE)[-1]

# use the final data frame

# make into a table with kable

# add styling with kableExtra

# add header with labels spanning 2 columns each

mtb2 %>%

knitr::kable(format="html",

col.names=c("Category","%",

"Category","%",

"Category","%"),

digits=2,

caption="Demographics of Survey Respondents") %>%

kableExtra::kable\_styling(c("striped","bordered"),

full\_width=FALSE) %>%

add\_header\_above(c("Ages"=2,"Education"=2,"Income"=2))

```

If you added this optional section with the merged table, go ahead and save the document and KNIT to HTML to view the result.

The last section of the document creates a clustered barchart figure showing the relative numbers of respondents who prefer their steak prepared via the different choices recorded, clustered by gender so the reader can compare the steak preparation choices between men and women. The last code chunk includes the ggplot function similar to what we did in the R script created in the last lesson. However, the code shown here adds additional options and layers for including labels for the X horizontal axis and the Y vertical axis as well as a title for the plot. The second to last option added uses the scale\_fill\_manual function from ggplot2 to customize the colors of the bar chart and provide nicer labels for the legend of the plot. The last line adds some styling by adding theme\_fivethirtyeight to the plot. theme\_fivethirtyeight is a function from the ggthemes package.

[THIS SECTION WILL BE CUT AND PASTED IN – LEAVE THIS SECTION OFF OF TELE-PROMPTER]

## Steak Preparation Preference by Gender

Finally, here is a breakdown of the `r nrow(sdat)` survey respondents for the \_\*\*Mountain\*\*\_ region of the US on how they prefer their steak to be prepared by gender.

```{r}

ggplot(sdat,

aes(x = steak\_prep, fill = female)) +

geom\_bar(position="dodge",colour="black") +

ggtitle(paste0("Steak Preference by Gender: ",

"Mountain"," Region")) +

xlab("Steak Preparation Preference") +

ylab("Number of Respondents") +

scale\_fill\_manual(values=c("skyblue","palevioletred"),

name="Gender",

breaks=c(FALSE,TRUE),

labels=c("Male", "Female")) +

theme\_fivethirtyeight()

```

Save the document one more time and KNIT to HTML to see everything completed.

Now let’s go ahead and back everything up to your Github account.

Open Git Bash and make sure you are in the correct directory:

C:\RepTemplates\Module3

Once in that directory, type in the following 4 Git commands to check the status of your local files compared to your Github cloud repository; add or stage the modified files; commit your changes; and then push the changes to your Github cloud repository.

git status

git add .

git commit –m “update steakArticle document”

git push

Now go to your Github repository, refresh to see your newly committed files.