Statistical Computing with R: Masters in Data Sciences 503, S30 First Batch, SMS, TU, 2021

Shital Bhandary

Associate Professor

Statistics/Bio-statistics, Demography and Public Health Informatics

Patan Academy of Health Sciences, Lalitpur, Nepal

Faculty, Data Analysis and Decision Modeling, MBA, Pokhara University, Nepal

Faculty, FAIMER Fellowship in Health Professions Education, India/USA.

Review Preview:

- Communication
- R markdown
- R notebook

R for data science: Chapter 26

- So far, you've learned the tools to get your data into R, tidy it into a form convenient for analysis, and then understand your data through transformation, visualization and modelling.
- However, it doesn't matter how great your analysis is unless you can explain it to others: you need to **communicate** your results.

Communicating with R markdown:

- R Markdown, a tool for integrating prose, code, and results.
- You can use R Markdown in notebook mode for analyst-to-analyst communication, and in report mode for analyst-to-decision-maker communication.
- Thanks to the power of R Markdown formats, you can even use the same document for both purposes.
- R markdown formats: many other varieties of outputs you can produce using R Markdown, including dashboards, websites, and books.

R markdown:

- R Markdown provides an unified authoring framework for data science, combining your code, its results, and your prose commentary.
- R Markdown documents are fully reproducible and support dozens of output formats, like PDFs, Word files, slideshows, and more.
- R Markdown files are designed to be used in three ways:

1. For communicating to decision makers, who want to focus on the conclusions, not the code behind the analysis.
2. For collaborating with other data scientists (including future you!), who are interested in both your conclusions, and how you reached them (i.e. the code).
3. As an environment in which to *do* data science, as a modern day lab notebook where you can capture not only what you did, but also what you were thinking.

Pre-requisites:

- You need the **rmarkdown** package, but you don’t need to explicitly install it or load it, as RStudio automatically does both when needed.

R markdown basics: it is a plain text file with .Rmd extension

The YAML header, surrounded by three --- in the beginning and end:

---

title: "Diamond sizes"

date: 2016-08-25

output: html_document

The R code chunk: surrounded by three ``` in the beginning and end:

```{r setup, include = FALSE}
library(ggplot2)

library(dplyr)

smaller <- diamonds %>%
filter(carat <= 2.5) ```

**Why to use “included = FALSE”?**
```

R markdown basics: it is a plain text file with .Rmd extension

We have data about `r nrow(diamonds)` diamonds.

Only `r nrow(diamonds) - nrow(smaller)` are larger than 2.5 carats.

The distribution of the remainder is shown below:

```
```{r, echo = FALSE} smaller %>%
ggplot(aes(carat)) +
geom_freqpoly(binwidth = 0.01)
```
```

What does echo=FALSE mean?

Chunk options:

- Chunk output can be customised with **options**, arguments supplied to chunk header.
- Knitr provides almost 60 options that you can use to customize your code chunks.
- eval = FALSE prevents code from being evaluated. (And obviously if the code is not run, no results will be generated).
- This is useful for displaying example code, or for disabling a large block of code without commenting each line.

Chunk options:

- include = FALSE runs the code, but doesn’t show the code or results in the final document. Use this for setup code that you don’t want cluttering your report.
- message = FALSE or warning = FALSE prevents messages or warnings from appearing in the finished file.
- echo = FALSE prevents code, but not the results from appearing in the finished file.
- Use this when writing reports aimed at people who don’t want to see the underlying R code.
- results = 'hide' hides printed output; fig.show = 'hide' hides plots.

Chunk options:

- error = TRUE causes the render to continue even if code returns an error.
- This is rarely something you'll want to include in the final version of your report, but can be very useful if you need to debug exactly what is going on inside your .Rmd.
- It's also useful if you want to deliberately include an error.
- The default, error = FALSE causes knitting to fail if there is a single error in the document.

Text formatting with R markdown:

Text formatting

---

italic or _italic_

bold __bold__

`code`

superscript^2^ and subscript~2~

Headings

1st Level Header

2nd Level Header

3rd Level Header

Text formatting with R markdown:

Lists

---

* Bulleted list item 1
* Item 2
    * Item 2a
    * Item 2b

1. Numbered list item 1
2. Item 2.

The numbers are incremented automatically in the output.

Links and images

<http://example.com>

[linked phrase](http://example.com)

![optional caption text](path/to/img.png)

Text formatting with R markdown:

Tables

```
-----------------------------------------
First Header | Second Header
-------------- | ------------------
Content Cell | Content Cell
Content Cell | Content Cell
```

- The best way to learn these is simply to try them out.
- It will take a few days, but soon they will become second nature, and you won't need to think about them.
- f you forget, you can get to a handy reference sheet with *Help > Markdown Quick Reference*.

Table with “knitr::kable” function (and xtable, stargazer, pander, tables, and ascii packages!)

- We have used kable already
- We have also used xtable already

Global options: More here - https://bookdown.org/yihui/rmarkdown-cookbook/

```
# When writing books and tutorials you can set:

knitr::opts_chunk$set(
        comment = "#>",
        collapse = TRUE
)
```

- This uses preferred comment formatting, and ensures that the code and output are kept closely entwined.

```
# When writing a report you can set:

knitr::opts_chunk$set(
        echo = FALSE
)
```

- That will hide the code by default, so only showing the chunks you deliberately choose to show (with echo = TRUE).

More to learn in the chapter 26:

- Inline code
- Troubleshooting
- YAML header
  - Parameters
- Bibliographies and citations
- More can be found at the Rmarkdown Rstudio website:

http://rmarkdown.rstudio.com/

Example: My Project → File → New file → R markdown → MDS_503_Session30.Rmd

````
---
title: "MDS 503 Session 30"
author: "Shital Bhandary"
date: "1/10/2022"
output: html_document
---

#Global option

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = FALSE)
```
````

Example: My Project → MDS_503_Session30.Rmd

Summary of diamonds data of tidyverse package

```{r tidyverse, echo=FALSE}
library(tidyverse)
summary(diamonds)
```

Why “tidyverse” package message is appearing?

#Scatterplot of carat and price with hex plots

```{r}
ggplot(diamonds, aes(carat, price)) +
geom_hex()
```

What is the interpretation?

Example: My Project → MDS_503_Session30.Rmd

```{r}
df <- diamonds %>%
mutate(make_model = row.names(diamonds)) %>%
filter(cut == "Fair") %>%
select(make_model, price, carat) %>%
arrange(make_model)
```

#What did you find with:

```{r}
summary(df)
```

How many variables? All needed to fit a linear model?

Example: My Project → MDS_503_Session30.Rmd

```{r, echo=TRUE}
df1 <- df[,3]
head(df1)
df2 <- df[,2]
head(df2)
df <- cbind(df1,df2)
head(df)
```

#Fitting a linear model now:

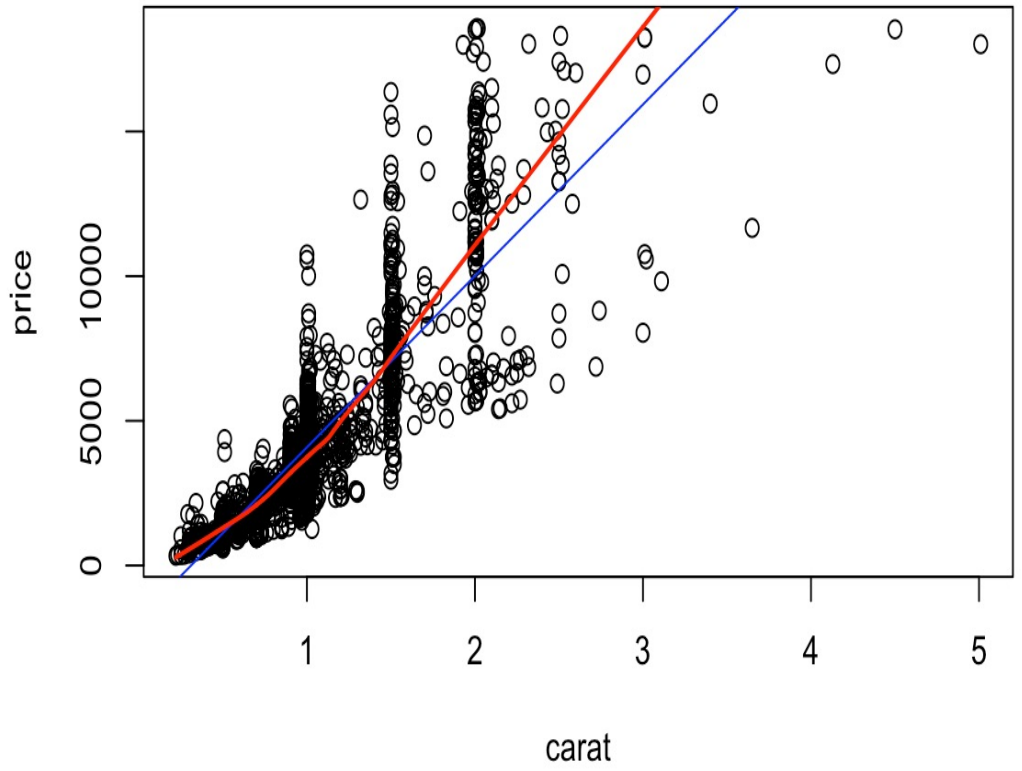
```{r, echo=TRUE}
fit.lm <- lm(price ~carat, data = df)
summary(fit.lm)
```

What is the R-square and RSE?

Example: My Project → MDS_503_Session30.Rmd

#Plotting the model and lowess:

```{r}
plot(df)
abline(fit.lm, col="blue")
lines(lowess(df$price~df$carat), lwd=2, col="red")
```



Let’s fit the linear model with a diamond of size up to 1 carat:

```{r}
df <- df %>%
filter(carat<=1)
summary(df)
```

#What have I done here?

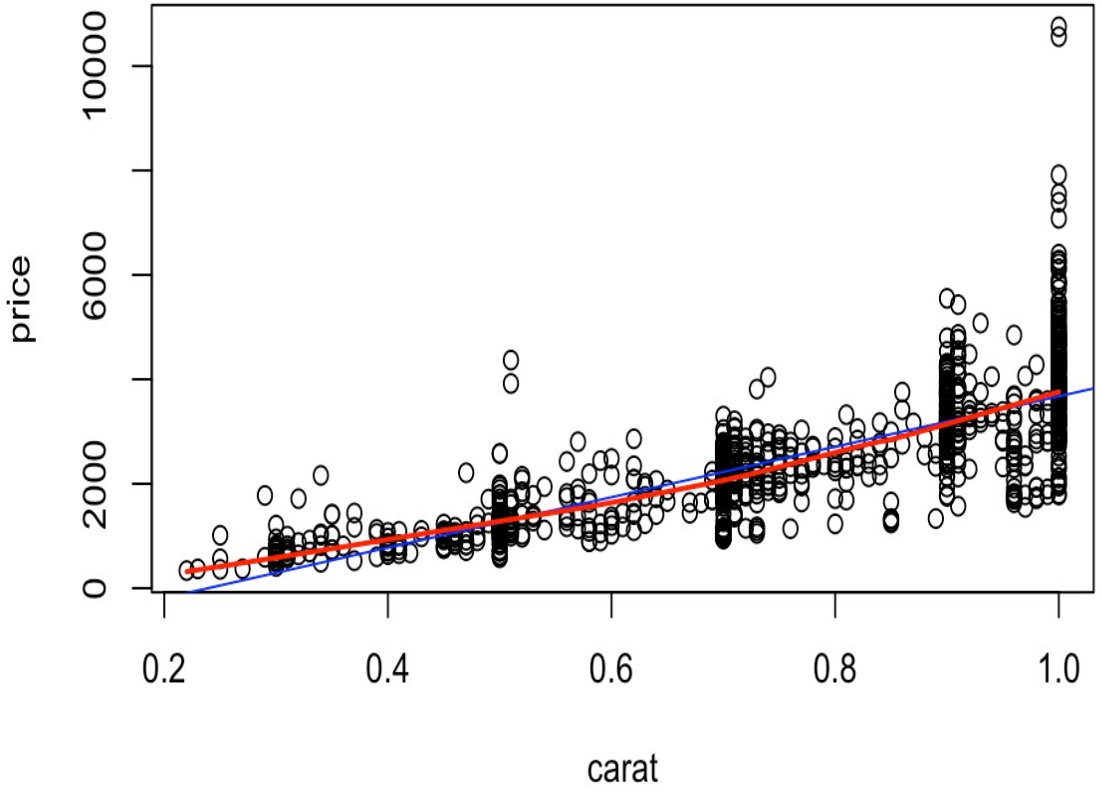
#Summary

```{r}
lm.fit <- lm(price~carat, data=df)
summary(lm.fit)
```

What is the R-square and RSE?

Did it improve the fit?

```{r}
plot(df)
abline(lm.fit, col="blue")
lines(lowess(df$price~df$carat), lwd=2, col="red")
```



What do you suggest now?

Assignment 10: Do as follows with rmarkdown, publish it in Rpubs and paste its link in MS Teams!

- Create two quantitative variables with random samples of size 1000 each:
- Age: 0 to 99 (random samples)
- BMI: 10 to 40 (random samples)
- Create a binary variable sex (1=Male and 0=Female) of 1000 random samples
- Split the data into “train” and “test” data using 80-20 partition
- For replication of the results, use your **class roll number as random.seed** during analysis
- Fit a linear regression model with BMI as dependent variable and age and sex and predictors in the train data samples

Assignment 10: Do as follows with rmarkdown, publish it in Rpubs and paste its link in MS Teams!

- Conduct residual analysis of the fitted model
- Use the fitted model to predict the test data samples
- Get R-square, MSE and RMSE for training as well as test data
- Take decision and write conclusion
- Use headings, comments and bullets in the markdown document
- Knit it and publish it in Rpubs!

You must do this in your Rpubs assignment:

- YAML Title must contain the following:
- Your name and roll number
- Couse name and code
- Course instructor’s name
- School/University
- Date
- Time

- R chunks:
- Make sure to show the codes as well as results
- But, do not show the messages of the package loadings!
- Past the link of your Rpubs in MS Teams assignment 10 for review.

Question/queries?

- Next classes
- Oral exam
- Revision

Thank you!

@shitalbhandary