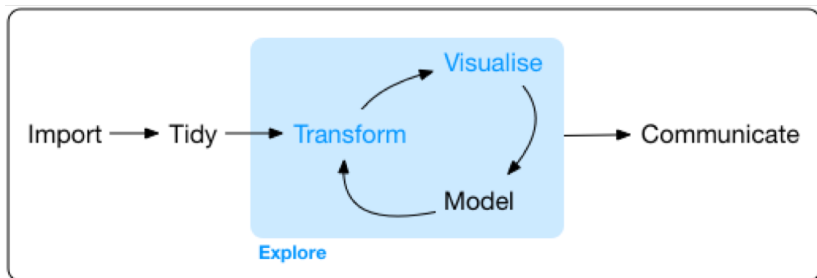


10. R Markdown

Communicating Results

R Markdown

- ▶ R Markdown provides a unified authoring framework for data science, combining your code, results and prose commentary
- ▶ R Markdown documents are fully reproducible and support many output formats (pdfs, slideshows, and more).
- ▶ Course slides developed with RMarkdown, see <https://github.com/JimDuggan/DSORR>



Program

Why use R Markdown?

- ▶ For communicating to decision makers, who want to focus on conclusions, not the code behind the analysis.
- ▶ For collaborating with other data scientists, who are interested in your conclusions, and how you reached them
- ▶ As an environment in which to do data science, where you capture not only what you did, but what you were thinking

R Markdown Elements

- ▶ An (optional) YAML header surrounded by —
- ▶ Chunks of R Code, surrounded by “`”`
- ▶ Text mixed with simple text formatting

R Markdown Example

```
---  
title: "Diamond Sizes"  
date: 2017-08-25  
output:  
  html_document: default  
---
```

Here is an example of using **R Markdown**.

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

```
```{r, echo=FALSE}  
smaller <- diamonds %>%
 filter(carat <= 2.5)
```
```

We have data about `nrow(diamonds)` diamonds in our data set. Only `nrow(diamonds) - nrow(smaller)` are larger than 2.5 carats.

The distribution of the remainder is show below:

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

“knit” to HTML

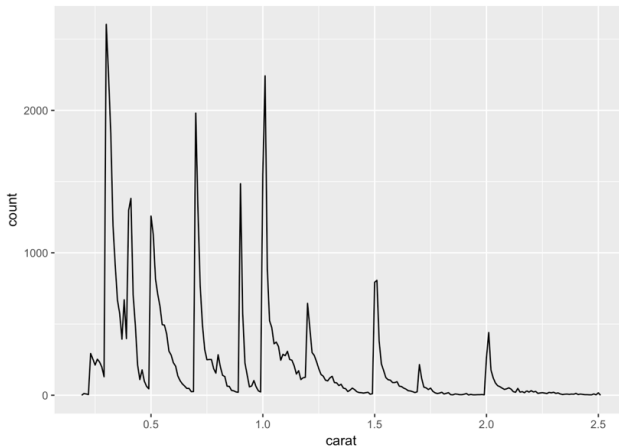
Diamond Sizes

2017-08-25

Here is an example of using **R Markdown**.

We have data about 53940 diamonds in our data set. Only **126** are larger than 2.5 carats.

The distribution of the remainder is show below:



“knit” to PDF

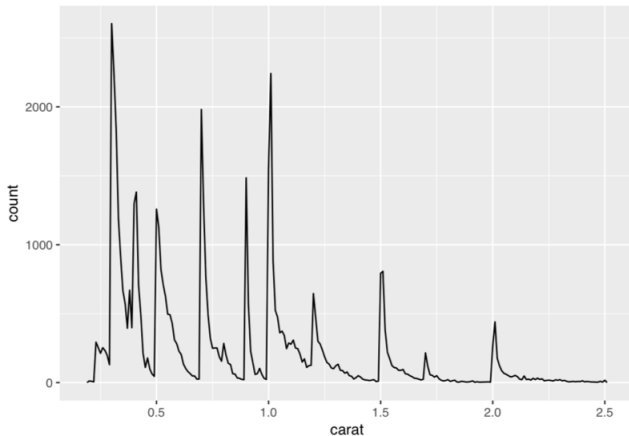
Diamond Sizes

2017-08-25

Here is an example of using **R Markdown**.

We have data about 53940 diamonds in our data set. Only **126** are larger than 2.5 carats.

The distribution of the remainder is show below:



Text formatting with Markdown

- ▶ Text Formatting
 - ▶ **italic**
 - ▶ ****bold****
 - ▶ `'code'`
- ▶ Headings
 - ▶ # First Level header
 - ▶ ## Second Level header
 - ▶ ### Third Level header
- ▶ Lists
 - ▶ * Bulleted list item 1
 - ▶ 1 Numbered list item 1

Inserting Chunks

This table summarizes what types of output each option suppresses. . .

| Option | Run code | Show Code | Output | Plots | Messages | Warnings |
|-----------------|----------|-----------|--------|-------|----------|----------|
| eval = FALSE | X | | X | X | X | X |
| include = FALSE | | X | X | X | X | X |
| echo = FALSE | | X | | | | |
| results= "hide" | | | X | | | |
| fig.show="hide" | | | | X | | |
| message=FALSE | | | | | X | |
| Warning=FALSE | | | | | | X |

Creating a Table

```
---  
title: "Table Test"  
output: html_document  
---
```

```
```{r}  
mtcars[1:5,1:10]
```
```

```
```{r}  
knitr::kable(
 mtcars[1:5,1:10],
 caption="A knitr kable"
)
```
```

Table Test

```
mtcars[1:5,1:10]
```

```
##           mpg  cyl  disp  hp  drat    wt  qsec vs am gear  
## Mazda RX4      21.0   6  160  110  3.90  2.620 16.46  0  1   4  
## Mazda RX4 Wag  21.0   6  160  110  3.90  2.875 17.02  0  1   4  
## Datsun 710     22.8   4  108   93  3.85  2.320 18.61  1  1   4  
## Hornet 4 Drive  21.4   6  258  110  3.08  3.215 19.44  1  0   3  
## Hornet Sportabout 18.7   8  360  175  3.15  3.440 17.02  0  0   3
```

```
knitr::kable(  
  mtcars[1:5,1:10],  
  caption="A knitr kable"  
)
```

A knitr kable

| | mpg | cyl | disp | hp | drat | wt | qsec | vs | am | gear |
|-------------------|------|-----|------|-----|------|-------|-------|----|----|------|
| Mazda RX4 | 21.0 | 6 | 160 | 110 | 3.90 | 2.620 | 16.46 | 0 | 1 | 4 |
| Mazda RX4 Wag | 21.0 | 6 | 160 | 110 | 3.90 | 2.875 | 17.02 | 0 | 1 | 4 |
| Datsun 710 | 22.8 | 4 | 108 | 93 | 3.85 | 2.320 | 18.61 | 1 | 1 | 4 |
| Hornet 4 Drive | 21.4 | 6 | 258 | 110 | 3.08 | 3.215 | 19.44 | 1 | 0 | 3 |
| Hornet Sportabout | 18.7 | 8 | 360 | 175 | 3.15 | 3.440 | 17.02 | 0 | 0 | 3 |

YAML Header

- ▶ “Yet Another Markup Language”
- ▶ Useful features
 - ▶ Parameters
 - ▶ Bibliographies

YAML Example

```
---
title: "Parameter Test"
bibliography: ref.bib
params:
  my_class: suv
  my_time: !r lubridate::now()

output:
  html_document: default
  pdf_document: default
---
|
The time is now `r params$my_time`

The reference is [paper1]
```

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

class <- mpg %>% filter(class == params$my_class)

```

```{r, message=FALSE}
ggplot(class,aes(x=displ,y=hwy))+
 geom_point()+
 geom_smooth(se=F)

```
```

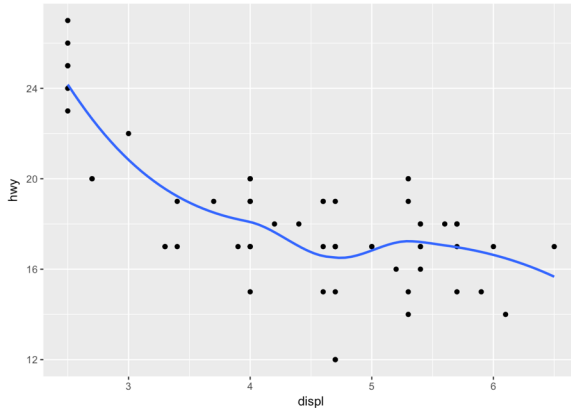
Sample Output

Parameter Test

The time is now 2017-11-15 19:41:58

The reference is (Koppeschaar et al. 2017)

```
ggplot(class,aes(x=displ,y=hwy))+  
  geom_point()+  
  geom_smooth(se=F)
```



Koppeschaar, E. Carl, Vittoria Colizza, Caroline Guerrisi, Clément Turbelin, Jim Duggan, John W. Edmunds, Charlotte Kjelsø, et al. 2017. "Influenzanet: Citizens Among 10 Countries Collaborating to Monitor Influenza in Europe." *JMIR Public Health Surveill* 3 (3): e66. <http://publichealth.jmir.org/2017/3/e66/>.

Presentations

Challenge 1.3

Write an R function (evens) that filters a vector to return all the even numbers. Use the modulus operator `%%`, and also logical filtering of vectors.

```
```{r,echo=F}  
evens <- function(x)x[x %% 2 == 0]
```
```



```
```{r,echo=T}  
x <- 1:6
x
y <- evens(x)
y
```
```



Output

Challenge 1.3

Write an R function (`evens`) that filters a vector to return all the even numbers. Use the modulus operator `%%`, and also logical filtering of vectors.

```
x <- 1:6
```

```
x
```

```
## [1] 1 2 3 4 5 6
```

```
y <- evens(x)
```

```
y
```

```
## [1] 2 4 6
```

Test Slide with Plot

