Introduction to R Markdown

etibhar

2016-07-25

Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see `http://rmarkdown.rstudio.com`.

Using the **rmarkdown** package (`pandoc` version 1.12.3 or higher is required), the rendering command is

```
rmarkdown::render("your rmd file")
```

The default is HTML output (because `output` has been specified as such in the heading of the current document). If you would like a markdown (.md) or github (to include graphics, with preview in html) document, add an argument:

```
rmarkdown::render("your rmd file", md_document())
rmarkdown::render("your rmd file", github_document())
```

The same way, PDF output (`pdflatex` is needed, that is `texlive-base`, `texlive-latex-base`, `texlive-latex-extra`, and `texlive-fonts-recommended`):

```
rmarkdown::render("your rmd file", pdf_document())
```

or OpenOffice or Word document:

```
rmarkdown::render("your rmd file", odt_document())
rmarkdown::render("your rmd file", word_document())
```

or `beamer` (PDF) or `slidy` (HTML) presentation:

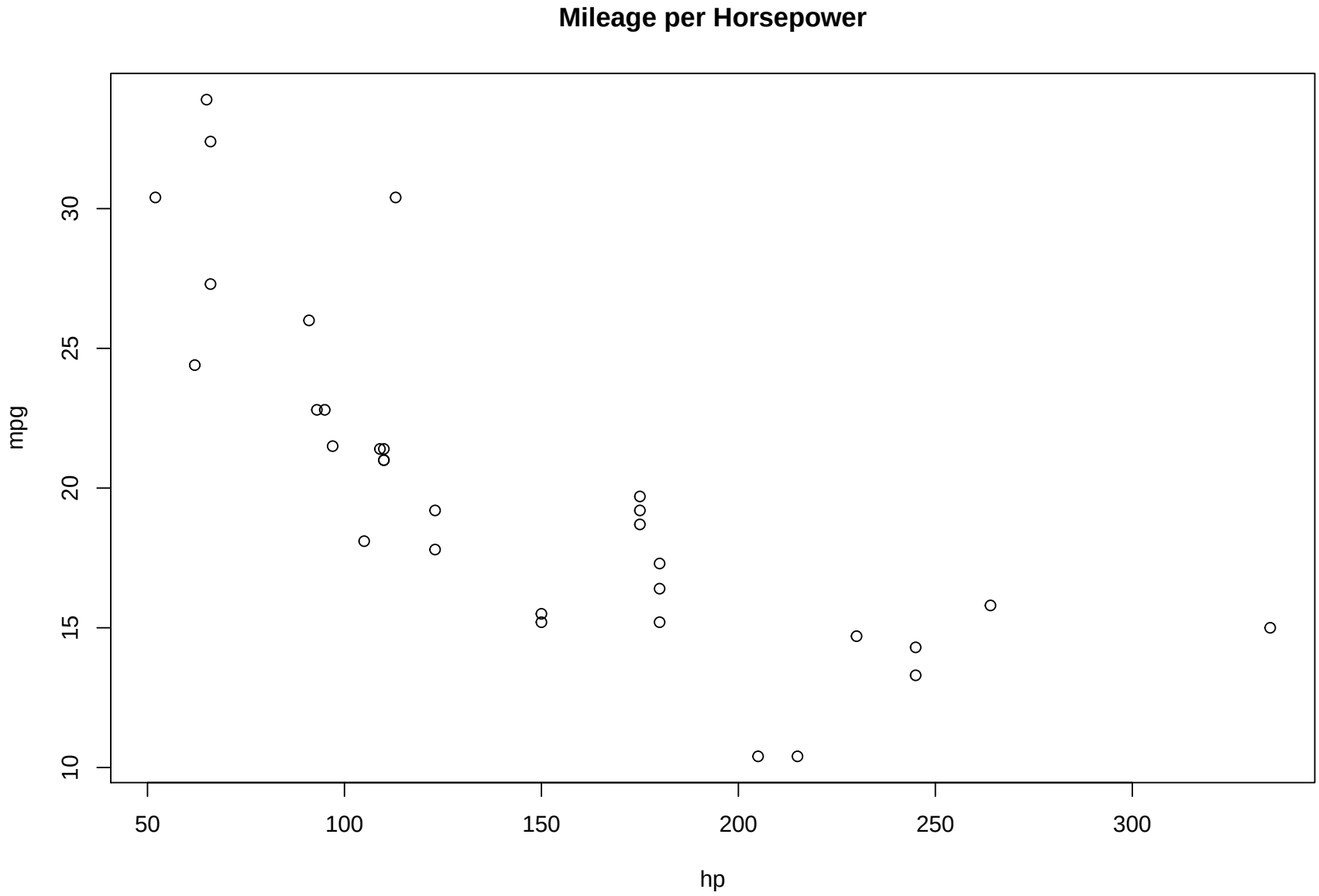
```
rmarkdown::render("your rmd file", beamer_presentation())
rmarkdown::render("your rmd file", slidy_presentation())
```

When the document is generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(mtcars)
```

```
##       mpg             cyl             disp             h
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.
##  Median :19.20   Median :6.000   Median :196.3   Median
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.
##       drat             wt             qsec             v
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.
##  Median :3.695   Median :3.325   Median :17.71   Median
##  Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.
##  Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.
```

You can also embed plots, for example:



Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Session Information

```
sessionInfo()
```

```
## R version 3.3.1 (2016-06-21)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 14.04.4 LTS
##
## locale:
##  [1] LC_CTYPE=en_GB.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_GB.UTF-8        LC_COLLATE=en_GB.UTF-8
##  [5] LC_MONETARY=en_GB.UTF-8    LC_MESSAGES=en_GB.UTF-8
##  [7] LC_PAPER=en_GB.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_GB.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  me
##
## other attached packages:
```

Supported Languages

```
library(knitr)
names(knitr::knit_engines$get())
```

```
##  [1] "awk"       "bash"      "coffee"    "gawk"      "gr
##  [6] "haskell"   "lein"      "mysql"     "node"      "pe
## [11] "psql"      "python"    "Rscript"   "ruby"      "sa
## [16] "scala"     "sed"       "sh"        "stata"     "zs
## [21] "highlight" "Rcpp"      "tikz"      "dot"       "c'
## [26] "fortran"   "asy"       "cat"       "asis"      "st
## [31] "block"     "js"        "css"
```

Some Perl

```
package TS v0.0.1 {
    sub ts {
        my @art = localtime(time);
        return sprintf("%4d%02d%02d_%02d%02d%02d",
                       $art[5]+1900, # year
                       ++$art[4],    # month
                       $art[3],      # day
                       $art[2],      # hour
                       $art[1],      # minute
                       $art[0]);     # second
    }
}
1;
```

Some C

```c
#include <stdio.h>
#include <stdlib.h>
#include <signal.h>
#include <time.h>
#include <unistd.h>
#include <stdbool.h>

time_t start, end;

void signal_callback_handler(int signum)
{
   printf("Caught signal %d\n",signum);
   printf("end: %ld\n",(long)end);
   double time_difference = difftime(end,start);
   printf("%lfs has passed.\n",time_difference);
   exit(signum);
}
```