Reporting using Quarto

Taro Mieno

# Basic illustrations

## Quarto

This is a Quarto document.

summary(cars)

speed dist   
 Min. : 4.0 Min. : 2.00   
 1st Qu.:12.0 1st Qu.: 26.00   
 Median :15.0 Median : 36.00   
 Mean :15.4 Mean : 42.98   
 3rd Qu.:19.0 3rd Qu.: 56.00   
 Max. :25.0 Max. :120.00

## Including Plots

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.

## Including a table

knitr::kable(iris[1:5, ], caption = "A caption")

A caption

| Sepal.Length | Sepal.Width | Petal.Length | Petal.Width | Species |
| --- | --- | --- | --- | --- |
| 5.1 | 3.5 | 1.4 | 0.2 | setosa |
| 4.9 | 3.0 | 1.4 | 0.2 | setosa |
| 4.7 | 3.2 | 1.3 | 0.2 | setosa |
| 4.6 | 3.1 | 1.5 | 0.2 | setosa |
| 5.0 | 3.6 | 1.4 | 0.2 | setosa |

## Inline r code

set.seed(2334)  
yield <- 200 \* runif(100)  
mean\_yield <- mean(yield)

The average yield is 103.4483146.

# Markdown basics (level 1 header)

## level 2 header

### level 3 header

See the source rmd file:

* list item 1
  + another list
* list item 2
* list item 3
* *italics*
* **bold**
* code (do not put r right after the first tick mark)
* inline math: I hate math. What does this even mean?
* math:
* link: [Markdown basics](https://rmarkdown.rstudio.com/lesson-8.html)
* citationL: The best resource to learn how to use Rmarkdown is [@xie2018r].

# Chunk options

## echo and eval

**R code and results**

summary(cars)

speed dist   
 Min. : 4.0 Min. : 2.00   
 1st Qu.:12.0 1st Qu.: 26.00   
 Median :15.0 Median : 36.00   
 Mean :15.4 Mean : 42.98   
 3rd Qu.:19.0 3rd Qu.: 56.00   
 Max. :25.0 Max. :120.00

**R code suppressed**

speed dist   
 Min. : 4.0 Min. : 2.00   
 1st Qu.:12.0 1st Qu.: 26.00   
 Median :15.0 Median : 36.00   
 Mean :15.4 Mean : 42.98   
 3rd Qu.:19.0 3rd Qu.: 56.00   
 Max. :25.0 Max. :120.00

**Only R code (R code was not evaluated)**

summary(cars)

**Both R codes and results not shown**

## messages and warnings

### both messages and warnings

library(tidyverse)

── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
✔ dplyr 1.1.3 ✔ readr 2.1.4  
✔ forcats 1.0.0 ✔ stringr 1.5.0  
✔ ggplot2 3.4.3 ✔ tibble 3.2.1  
✔ lubridate 1.9.2 ✔ tidyr 1.3.0  
✔ purrr 1.0.2   
── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
✖ dplyr::filter() masks stats::filter()  
✖ dplyr::lag() masks stats::lag()  
ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

detach("package:tidyverse", unload = TRUE)

### messages suppressed

library(tidyverse)  
detach("package:tidyverse", unload = TRUE)

### both messages and warnings suppressed

library(tidyverse)  
detach("package:tidyverse", unload = TRUE)

## results

No results are shown.

a <- runif(100)  
a

But, this shows the above code was evaluated and object a was created.

a

[1] 0.442721578 0.274517669 0.832489557 0.496599494 0.222737525 0.713532158  
 [7] 0.087589334 0.561854588 0.840218469 0.712486817 0.276695837 0.342732609  
 [13] 0.040782033 0.206631539 0.661236504 0.560264982 0.132385486 0.275213450  
 [19] 0.315067072 0.423239615 0.488549697 0.523500251 0.144676441 0.402919444  
 [25] 0.078005459 0.513939508 0.243389158 0.118420537 0.558043503 0.880511739  
 [31] 0.010506572 0.339365883 0.018768277 0.210161775 0.924343563 0.901447605  
 [37] 0.447185481 0.247151799 0.748219973 0.538129915 0.030727680 0.410451712  
 [43] 0.356162286 0.169928680 0.176799986 0.490199240 0.889828242 0.802454498  
 [49] 0.129069953 0.694477625 0.701561396 0.082388175 0.079017293 0.114650592  
 [55] 0.778099407 0.038132727 0.487081168 0.278149180 0.025802752 0.249283593  
 [61] 0.846890552 0.223062190 0.210762547 0.376761291 0.082064051 0.233958469  
 [67] 0.342962488 0.730572376 0.894250010 0.988800461 0.497710637 0.946655608  
 [73] 0.144414078 0.251685329 0.346703513 0.335317247 0.892013595 0.130301862  
 [79] 0.081560876 0.045353443 0.344891126 0.410071261 0.621566948 0.818836779  
 [85] 0.329713708 0.293244870 0.089540284 0.132215122 0.496356317 0.359872945  
 [91] 0.571551344 0.304231321 0.803927564 0.839980256 0.665425682 0.002669636  
 [97] 0.537816064 0.728669597 0.252346275 0.724311874

## include = FALSE

No results are shown.

But, this shows the above code was evaluated and object a was created.

b

[1] 0.6512069055 0.6651519733 0.6681661860 0.4336436889 0.4142855881  
 [6] 0.6184352725 0.4629165505 0.4269558790 0.8218558307 0.1010162444  
 [11] 0.0123708961 0.0313172399 0.9946714859 0.2823574010 0.8826807365  
 [16] 0.9619706806 0.6970768331 0.3793270397 0.4593482118 0.9015936744  
 [21] 0.7884225431 0.8172776175 0.0025936761 0.0419812859 0.1375661893  
 [26] 0.0807574349 0.0001389624 0.0964157293 0.1365731203 0.2667404707  
 [31] 0.1942046226 0.7312312324 0.2722529070 0.9049718524 0.0646821384  
 [36] 0.7556912473 0.9640818359 0.1830108573 0.4955519645 0.3714570620  
 [41] 0.1313385291 0.4950157166 0.4644586414 0.4623787857 0.1028932761  
 [46] 0.7315323201 0.9178925341 0.0023527150 0.2764258326 0.4405686152  
 [51] 0.3944474871 0.9386764970 0.6078627864 0.7109344315 0.7814479736  
 [56] 0.1052157704 0.0022772625 0.4846693375 0.0733176514 0.1580989382  
 [61] 0.8278034444 0.1411051280 0.9072100250 0.0400990990 0.1020173531  
 [66] 0.7970790255 0.0129714953 0.8611674064 0.9640595755 0.1485804692  
 [71] 0.4508498593 0.0106433611 0.2158272723 0.0898387097 0.7344084904  
 [76] 0.3068988514 0.9020985381 0.9966172257 0.2738207695 0.4925583762  
 [81] 0.1939275623 0.0793949533 0.0870394167 0.1351574806 0.1819930521  
 [86] 0.7270940391 0.2675342008 0.2667735845 0.2446220699 0.8374004695  
 [91] 0.5338095035 0.6106576971 0.3987668650 0.3012215076 0.0457166785  
 [96] 0.2340719013 0.1729808906 0.7637378986 0.3256172710 0.8985493686

# Caching

#--- repeat steps 1-3 B times ---#  
MC\_sim <- function(i) {  
 N <- 50000 # sample size  
  
 #--- steps 1 and 2: ---#  
 mu <- rnorm(N) # the common term shared by both x and u  
 x <- rnorm(N) + mu # independent variable  
 v <- rnorm(N) + mu # error  
 y <- 1 + 2 \* x + v # dependent variable  
 data <- data.frame(y = y, x = x)  
  
 #--- OLS ---#  
 reg <- lm(y ~ x, data = data) # OLS  
  
 #--- return the coef ---#  
 return(reg$coef["x"])  
}  
  
MC\_results <- lapply(1:1000, MC\_sim) %>%  
 unlist() %>%  
 mean()  
  
MC\_results

MC\_results