GPH-GU 2182 · Sample Quiz 1

Practice for Quiz 1 · Weeks 1–2 Topics · 20 Minutes · Multiple Choice

How to use this

This is a practice quiz. It is **not** the quiz you will sit, and none of these questions appear on it. It matches the real quiz in every other way: ten multiple-choice questions, one point each, twenty minutes, and the same topics in the same proportions. If you can do this comfortably without help, you are ready.

The real Quiz 1 is on **Wednesday September 16**, during the last 30 minutes of class. It is **closed book, closed notes, and closed AI**, with no laptops, phones, or watches. Nothing to install, nothing to submit: it is on paper.

Practise it the way you will sit it. Put the slides away, close your browser, and do not run the code. Working out what R returns in your head is the skill being tested, so checking an answer in the Console first defeats the point. Check afterwards, then read the answer key.

Everything covered comes from Weeks 1 and 2, which is R02Pro chapters 1 to 3.

Q1. Packages (1 point)

You have already installed the package. Which command must you run **in every new R session** before using its functions?

A. `library(r02pro)`
B. `install.packages("r02pro")`
C. `update.packages("r02pro")`
D. Nothing, an installed package is always available

Answer: _____

Q2. Quarto code chunks (1 point)

You add the option `#| eval: false` to a code chunk. When you render the document, what happens?

A. The code is run but hidden, and the output appears
B. The chunk is removed from the document entirely
C. The code is shown but not run, so no output appears
D. The document fails to render

Answer: _____

Q3. Type coercion (1 point)

What does the following return?

```
typeof(c("a", 1, TRUE))
```

A. `"logical"`
B. `"character"`
C. `"double"`
D. `"list"`

Answer: _____

Q4. Vector recycling (1 point)

What is the exact output of the following?

```
a <- c(1, 2, 3, 4, 5, 6)
b <- c(10, 20, 30)
a + b
```

A. `11 22 33`
B. `11 22 33 4 5 6`
C. An error, because the vectors have different lengths
D. `11 22 33 14 25 36`

Answer: _____

Q5. Repetition (1 point)

What is the exact output of the following?

```
rep(c(1, 2), times = 3)
```

A. `1 2 1 2 1 2`
B. `1 1 1 2 2 2`
C. `1 2 3`
D. `3 3`

Answer: _____

Q6. Sequences (1 point)

What is the exact output of the following?

```
seq(2, 10, by = 2)
```

A. `2 6 10`
B. `2.0 4.0 6.0 8.0 10.0`
C. `2 4 6 8 10`
D. `1 2 3 4 5`

Answer: _____

Q7. Negative indexing (1 point)

Let `doses <- c(4, 8, 12, 16)`. What does `doses[-1]` return?

A. `4`
B. `8 12 16`
C. `16`
D. `4 8 12`

Answer: _____

Q8. Logical vectors and means (1 point)

Four patients are screened, and `TRUE` means the test was positive:

```
positive <- c(TRUE, FALSE, TRUE, TRUE)
mean(positive)
```

What is the result, and what does it represent?

A. `3`, the number of positive tests
B. `NA`, because you cannot average logical values
C. `TRUE`, because most values are `TRUE`
D. `0.75`, the proportion of positive tests

Answer: _____

Q9. Sorting (1 point)

What is the exact output of the following?

```
sort(c(7, 3, 9, 1), decreasing = TRUE)
```

A. `9 7 3 1`
B. `1 3 7 9`
C. `4 2 1 3`
D. `3 2 4 1`

Answer: _____

Q10. Modifying a vector in place (1 point)

A survey codes "no answer" as `-99`. After running the following, what does `bp` contain?

```
bp <- c(88, -99, 95, -99, 72)
bp[bp == -99] <- NA
bp
```

A. `88 95 72`
B. `88 -99 95 -99 72`
C. `88 NA 95 NA 72`
D. `NA NA NA NA NA`

Answer: _____

Answers: on the next page, or in the separate answer key. Mark your own answers first.

Answers

Q	1	2	3	4	5	6	7	8	9	10
Answer	A	C	B	D	A	C	B	D	A	C

Q1 A. `install.packages()` is once per computer; `library()` is once per session.

Q2 C. `eval` controls whether the code *runs*. With it off, the code is still printed, but nothing is executed, so there is no output.

Q3 B. A vector holds a single type, and character is the most general, so the presence of `"a"` forces everything to character. Note that `TRUE` becomes the string `"TRUE"`.

Q4 D. `b` is recycled twice over `a` because 6 is a multiple of 3, giving `10 20 30 10 20 30`. No warning appears, which is exactly why recycling is dangerous.

Q5 A. `times` repeats the whole vector; `each` (option B) repeats every element in place.

Q6 C. `by = 2` sets the step size. Compare `length.out`, which instead fixes how many values you get and computes the step.

Q7 B. A negative index drops that position, so `doses[-1]` returns everything except the first element. It does not count from the end, which is what students coming from Python often expect.

Q8 D. Logical values coerce to 1 and 0, so the mean is the proportion that are `TRUE`. This is the idiom for computing a prevalence, and it is worth remembering.

Q9 A. `sort()` returns the sorted values. Options C and D are `order()` and `rank()` output, which return positions instead.

Q10 C. The logical vector `bp == -99` picks the two sentinel values, and assigning `NA` replaces them in place. Length is unchanged, which is the point: the missing readings stay in the data as missing rather than disappearing.

One more thing worth knowing

This did not fit as a question, but it catches people constantly, so it is worth thirty seconds now.

```
sbp <- c(118, NA, 152, 141, NA)
sbp[sbp > 140]
```

You might expect `152 141`. What you actually get is:

```
NA 152 141 NA
```

The reason is that `NA > 140` is not `FALSE`, it is `NA`: R cannot say whether an unknown value exceeds 140. Subsetting by `NA` yields `NA`, so each missing reading comes back as a missing element.

When you want only the real readings above the threshold, say so explicitly:

```
sbp[!is.na(sbp) & sbp > 140]
# 152 141
```

The same idea applies to summaries, which is why `mean(sbp, na.rm = TRUE)` needs that argument to give you an answer rather than `NA`.