



Certified R Programmer (CRPR)

Ten Sample Questions

These ten questions are written at the difficulty of the live exam. Six carry a figure, a data table, a configuration or a code listing; four are plain scenario questions, as on the live paper. Every one is a scenario a working R programmer would meet, and none of them appears in the live question bank.

The live exam is multistage adaptive: how you score on one stage selects the difficulty of the next, so no two candidates sit the same paper. It is a performance-based exam: each form can carry up to 5 to 12 performance-based questions alongside the multiple-choice questions, which means you build, sequence or complete something rather than only choosing from a list. Answers and full explanations follow each question here; the live exam does not disclose item-level answers.

Question 1

Domain 1: R Fundamentals and Syntax

[CONFIG](#)

Sarah Miller runs the following in the R console.

```
x <- c(1L, 2L, 3L)
typeof(x)
```

Listing 1.1 Checking a vector type.

What does typeof return?

- ☐ A double
- ☐ B numeric
- ☐ C integer
- ☐ D list

Correct answer: C

The suffix L makes each element an integer literal, so the vector has type integer and typeof returns integer. The class numeric groups doubles and integers, but typeof reports the exact storage type rather than the class.

Question 2

Domain 2: Data Types and Structures

[DATA](#)

James Carter compares R structures by whether each holds a single type, with one entry missing.

Structure	Single type only
Vector	Yes
List	(to choose)
Matrix	Yes
Data frame	No, by column

Table 2.1 Structures by element type.

What fits the List row?

- ☐ A Holds mixed types
- ☐ B Homogeneous only
- ☐ C Only whole numbers
- ☐ D Strings only

Correct answer: A

A list is the flexible container in R and can hold elements of different types and even other lists, so it holds mixed types. A vector and a matrix coerce their contents to one type, which is why they are marked single type.

Question 3

Domain 3: Control Flow and Functions

CONFIG

Emma Wilson applies an anonymous function across a vector.

```
nums <- c(2, 4, 6)
sapply(nums, function(v) v * v)
```

Listing 3.1 Squaring each element.

What does this return?

- ☐ A Error, not numeric
- ☐ B 2 4 6
- ☐ C 6 6 6
- ☐ D 4 16 36

Correct answer: D

sapply applies the function to each element and simplifies the result to a vector, so squaring 2, 4 and 6 yields 4, 16 and 36. The apply family replaces an explicit loop while returning one value per input element.

Question 4

Domain 4: Data Manipulation with the tidyverse

CONFIG

Olivia Davis writes a short dplyr pipeline over a data frame named df.

```
df %>%
  filter(score > 80) %>%
  summarise(mean(score))
```

Listing 4.1 A dplyr pipeline.

What does the pipeline compute?

- ☐ A Total row count
- ☐ B Mean high score
- ☐ C Median of scores
- ☐ D Every score sorted

Correct answer: B
The pipe passes df to filter, which keeps rows where score exceeds 80, then summarise computes the mean of those remaining scores. So the result is the mean high score, not a count, a median or a sorted list.

Question 5

Domain 5: Statistics and Modeling

DATA

Liam Anderson maps each task to its base R function, with one function missing.

Task	Function
Mean	mean()
Standard deviation	(to choose)
Correlation	cor()
Linear model	lm()

Table 5.1 Tasks and base functions.

Which function gives the standard deviation?

- ☐ A var()
- ☐ B summary()
- ☐ C sd()
- ☐ D sqrt()

Correct answer: C
The base function sd() returns the standard deviation of a numeric vector. The related var() returns the variance, and although the standard deviation is the square root of the variance, sqrt() alone does not compute it from the raw data.

Question 6

Domain 6: Visualization, Packages and Reproducibility

CONFIG

Ava Thompson builds a plot with ggplot2.

```
ggplot(df, aes(x, y)) +  
  geom_point()
```

Listing 6.1 A ggplot2 layer.

What kind of plot does this draw?

- ☐ A Scatter
- ☐ B Bar chart
- ☐ C Line chart
- ☐ D Histogram

Correct answer: A

The `geom_point` layer draws one point per row using the mapped `x` and `y`, which produces a scatter plot. A bar chart uses `geom_col` or `geom_bar`, a line uses `geom_line`, and a histogram uses `geom_histogram`.

Question 7

Domain 2: Data Types and Structures

A team builds a vector that mixes numbers and text in the same `c()` call, then finds arithmetic fails. What happened to the values?

- ☐ A Throws an error
- ☐ B Coerced to text
- ☐ C Numbers dropped
- ☐ D Mix is kept

Correct answer: B

A vector holds a single type, so R coerces every element up to the most general type present, which for numbers and text is character. That is why arithmetic fails: the numbers became text and are no longer numeric.

Question 8

Domain 3: Control Flow and Functions

A developer needs to add two numeric vectors of equal length element by element and wants the clearest, fastest R style. What is best?

- ☐ A Write a for loop
- ☐ B Use a while loop
- ☐ C Use `x + y`
- ☐ D Nest two loops

Correct answer: C

R operators are vectorized, so $x + y$ adds the vectors element by element in one clear, fast expression. An explicit loop is slower and more verbose here, and nesting loops is unnecessary for parallel elementwise work.

Question 9

Domain 4: Data Manipulation with the tidyverse

Two data frames share a customer id column and must be combined so matching rows sit side by side. What is the sound tidyverse approach?

- ☐ A Bind the rows
- ☐ B Copy by hand
- ☐ C Paste columns
- ☐ D Use a join

Correct answer: D

A join such as `inner_join` or `left_join` matches rows on the shared key and aligns the columns from both frames. Binding rows stacks frames vertically, and pasting columns blindly would misalign records that are not already in the same order.

Question 10

Domain 6: Visualization, Packages and Reproducibility

A colleague cannot reproduce an analysis that samples random numbers, because the results differ on every run. What one step fixes this?

- ☐ A Set a seed
- ☐ B Delete the data
- ☐ C Round the output
- ☐ D Rerun it often

Correct answer: A

Calling `set.seed` with a fixed value makes the random number generator produce the same sequence each run, so sampling becomes reproducible. Rounding or repeating runs does not pin the randomness, and deleting data loses the analysis.