MATEMATISKA INSTITUTIONEN
STOCKHOLMS UNIVERSITET
Avd. Matematik
Examinator: Per Alexandersson

ANSWER KEY
Tentamensskrivning i
Statistisk databehandling
MT4007
2026-01-12

— ANSWER KEY —

Part I

1. According to the principles of Functional Programming described in the course, what is the defining characteristic of a *pure function*?

   (a) It modifies global variables to ensure state consistency across the program.

   (b) It always returns `void` or `None`.

   (c) **Its output depends solely on its input arguments, and it produces no side effects.**

   (d) It is a function that must be called from within a Jupyter Notebook class.

   (e) It automatically handles version control commits when executed.

   (f) It only accepts one parameter as input.

2. What does immutability mean in functional programming?

   (a) Functions cannot be passed as arguments.

   (b) **Values cannot be changed after creation; new values are created instead.**

   (c) Variables must always be global.

   (d) Code cannot be modified after deployment.

   (e) Only numeric types can be used.

   (f) You can not update the code after source is made public.

3. What is the recommended reason to use a branch when using Git?

   (a) To permanently replace `main` with experimental work.

   (b) To avoid writing commit messages.

   (c) **To try changes or updates safely without affecting the stable `main` branch.**

   (d) To prevent merge conflicts by never merging.

   (e) To store large data files in a tree structure.

   (f) To create a new directory in the repository.

4. Why is it recommended to include a `.gitignore` file in your repository root?

   (a) To prevent the repository from accepting code that contains syntax errors.

   (b) To automatically format Markdown files before they are pushed to GitHub.

   (c) To list the Python or R packages required to run the code.

   (d) **To prevent local files (like `.ipynb_checkpoints` or large data files) from being tracked by Git.**

   (e) To specify which warning messages should be ignored.

   (f) To encrypt sensitive passwords and API tokens automatically.

5. In a few sentences, explain the difference between Git and GitHub. Include what each one is used for in a typical workflow.

   Sample answer: Git is a *local* version control system that tracks changes to files on your computer. It lets you create commits (snapshots of your work), branch, merge, and revert changes. GitHub is a *remote* hosting platform for Git repositories. It stores your repository online so you can back up your work, share it with collaborators, and receive feedback (e.g., via pull requests). In a typical workflow, you use Git locally to commit changes, and then `git push` to upload them to GitHub.

Part II

6. In a Tidy Data Frame, how is data organized?

   (a) Variables are in rows, observations are in columns, and values are in colors.

   (b) **Variables are in columns, observations are in rows, and each cell holds a single value.**

   (c) The data is stored in SQL using the TDF package.

   (d) Observations are grouped by date, with multiple values per cell.

   (e) Column data descriptions is stored in the first row, and data starts from row 2.

   (f) The structure does not matter as long as the file is a .csv.

7. Which core operation (verb) would you use to create a *new* column (e.g., calculating BMI) based on existing columns (Weight and Height)?

   (a) Filter (b) Select (c) Group By (d) **Mutate** (e) Sort/Arrange (f) Summarize

8. You want to visualize the distribution of a single numeric variable (e.g., the age of participants). Which plot type is most appropriate?

   (a) Scatter plot
   (b) Stacked bar chart
   (c) **Histogram or density plot**
   (d) Line chart
   (e) Heatmap
   (f) Pie chart

9. What is a common best practice emphasized in the material?

   (a) Avoid axis labels to reduce clutter.
   (b) Use the same plot type for all questions for consistency.
   (c) **Always label axes (including units if needed) and provide context.**
   (d) Use 3D plots to illustrate categorical data.
   (e) Hide legends so viewers infer categories.
   (f) Use the standard green/red colors for true/false.

10. Give two examples of data formats that are organized as 'tree' structures, rather than tables.

    Sample answer: JSON (JavaScript Object Notation) and **XML** (eXtensible Markup Language). Both store data as hierarchical, nested structures (objects containing objects, arrays within arrays) rather than flat rows and columns. HTML is also an acceptable answer.

Part III

11. In a few sentences, describe the *exploratory data analysis* loop.

    Sample answer: The EDA loop is an iterative process: (1) **Pose a question** about the data (e.g., "Are older customers buying more?"). (2) **Transform, filter, or select** the relevant variables. (3) **Visualise** or summarise the result. (4) **Interpret** what you see—did it answer the question? Did it reveal something unexpected? (5) **Formulate the next question** based on what you learned. You repeat this cycle until you have built sufficient understanding of the data's structure, quality, and patterns.

12. When using the Interquartile Range (IQR) rule to flag outliers, how is the lower bound typically calculated?

    (a) $Mean - 2 \times StandardDeviation$

    (b) $Median - 1.5 \times IQR$

    (c) $Q1 - 1.5 \times IQR$

    (d) $Q3 + 1.5 \times IQR$

    (e) $Minimum\ value/2$

    (f) $Mode - 1 \times IQR$

13. Why might you choose to keep outliers in your dataset rather than removing them?

    (a) Because outliers help to identify bugs in the analysis.

    (b) **Because they might represent rare events that are crucial to the analysis.**

    (c) Because you cannot calculate the mean if outliers are removed.

    (d) Because plotting libraries expect to have some outliers to remove.

    (e) Because removing them may interfere with the Git commit history.

    (f) Because the IQR rule cannot be applied if these are removed.

14. When joining two tables, what should you verify to avoid unexpected results?

    (a) That both tables have the same number of columns.

    (b) **That join keys are clean, unique (or appropriately handled), and row counts make sense.**

    (c) That all column names are identical.

    (d) That both tables are sorted the same way.

    (e) That both tables use the same database format.

    (f) That both tables have no missing values.

15. Which of the following is NOT a common strategy for handling missing data?

    (a) Leave as missing and use methods robust to NA.

    (b) Drop rows with missing values.

    (c) Impute using mean, median, or mode.

    (d) **Replace with -1 to signal data is missing.**

    (e) Use grouped imputation based on categories.

    (f) Drop the specific column if it is mostly empty.

Part IV

16. Which SQL join type keeps all rows from both tables, filling with NULLs where there is no match?

    (a) `INNER JOIN`

    (b) `LEFT JOIN`

    (c) `RIGHT JOIN`

    (d) **`FULL OUTER JOIN` (or `FULL JOIN`)**

    (e) `CROSS JOIN`

    (f) `SELF JOIN`

17. What does the `PRIMARY KEY` constraint ensure in a SQL table?

    (a) The column can contain NULL values.

    (b) The column must reference another table.

    (c) The column must be numeric.

    (d) The column is automatically sorted.

    (e) The column is the first among all columns.

    (f) **The column values must be unique and NOT NULL.**

18. List four advantages of storing data in a database, compared to storing data in plain files on a disk.

    Sample answer: (1) **Data integrity via constraints**: databases enforce rules like PRIMARY KEY, NOT NULL, UNIQUE, and FOREIGN KEY, preventing invalid data from being inserted. (2) **Efficient querying**: SQL allows you to filter, join, and aggregate large datasets without loading everything into memory. (3) **Concurrent access**: multiple users or processes can read/write simultaneously without corrupting the data. (4) **Structured relationships**: tables can be linked via foreign keys, avoiding data duplication and keeping the data normalised. Other valid answers include: ACID transactions, indexing for performance, access control/permissions, and built-in backup/recovery mechanisms.

19. In the regex pattern `\d{3}-\d{2}-\d{4}`, what does `\d{3}` match?

    (a) Exactly 3 characters equal to `d`.

    (b) **Exactly 3 digits.**

    (c) 3 or more digits.

    (d) Up to 3 characters equal to `d`.

    (e) Any 3 data characters.

(f) The third day in a week.

20. Briefly, how you would **extract the date** from the following HTML element appearing on a web page you are scraping:

`<span id="date" class="c-dark" title="Today">2016-01-12</span>`.

Sample answer: Use an HTML parser (e.g., BeautifulSoup in Python or rvest in R) to locate the element, then extract its text content. For example in Python: parse the page with `BeautifulSoup(html, "html.parser")`, then call `soup.find("span", id="date")` (or use the CSS selector `#date`) to locate the `<span>` element, and finally call `.get_text()` (or `.text`) on it to retrieve the string `"2016-01-12"`. You can locate it by tag name plus `id="date"`, or by the CSS selector `span#date`, or by `span.c-dark`—any approach that uniquely identifies the element is acceptable.

Part V

21. Which data format is most commonly returned by modern REST APIs?

(a) PDF
(b) Excel (.xlsx)
(c) **JSON (JavaScript Object Notation)**
(d) Word (.docx)
(e) SQL dump
(f) Zip archive

22. What is the purpose of pagination in API responses?

(a) To encrypt data during transfer.
(b) **To split large datasets into smaller chunks.**
(c) To authenticate users securely.
(d) To sort results alphabetically.
(e) To compress JSON responses.
(f) To make scraping more difficult for hackers.

23. What is the main difference between using a REST API and Web Scraping?

    (a) Scraping HTML is faster than using an API.

    (b) **APIs are intended for returning data, while scraping extracts data manually from HTML.**

    (c) API access can cost money, while scraping is always free.

    (d) Scraping requires a web browser, while APIs do not.

    (e) APIs can only return text, while scraping can return anything.

    (f) APIs do not require handling character encodings.

24. If a webpage is *dynamic* (e.g. loaded via JavaScript), why might a simple `GET` request (like `requests.get()` in Python or `read_html()` in R) fail to find the data you see in the browser?

    (a) Because the data is encrypted.

    (b) Because `GET` request does not send the correct cookie data.

    (c) **Because only initial HTML is loaded, and the data is loaded by Javascript separately.**

    (d) Because Javascript is unsafe.

    (e) Because Javascript must be included as a separate package.

    (f) Because the data is returned as a Javascript file.

25. You encounter an HTTP `429 Too Many Requests` error. What is the appropriate programmatic response? Describe in a few sentences.

    Sample answer: A 429 status code means you have exceeded the server's rate limit. The appropriate response is to **pause** your script—check the `Retry-After` header in the response to see how long the server asks you to wait, then sleep for that duration before retrying. If no `Retry-After` header is provided, implement **exponential backoff**: wait a short time (e.g., 1 second), and double the wait on each successive 429 response. Wrap the request in a retry loop with a maximum number of attempts so your script does not run forever. The key principle is to respect the server's limits rather than hammering it with rapid retries.