Exercise Set 1

- Submit your answer to Moodle by **Wednesday, 12 November 2025**, no later than 23:59.
- You may answer anonymously or include your name, as you prefer.
- Each assignment must be written independently. Collaborative discussion is encouraged, and using external resources—including web searches—is permitted, but all answers must be your own work and not copied.
- All work will be peer-reviewed (by yourself and randomly assigned peers). For expectations, grading rubric, and further details, see the Exercise and Peer Review Instructions. A good answer should:
  - **Clearly explain the reasoning and process**, not just present final results.
  - **Be well-organised and concise**, with logical structure, precise technical language, and properly labelled code snippets/figures/tables as needed.
  - **Demonstrate factual correctness** with relevant, sufficient evidence and well-justified claims.
  - **Show critical analysis**, including justification of methods, discussion of alternatives or limitations, and integration of course concepts to the extend applicable.
  - **Reflect master's-level understanding**: not just procedural work, but explanation, synthesis, and professional communication.
  - **Not look like a code dump**: unless explicitly requested, you do not need to include program code in your answer.
- The language of submission and review is English.
- Upload a single, well-formatted PDF. Double-check before submitting that all figures and sections are present; Jupyter Notebooks can produce incomplete PDFs. We recommend R Markdown (or Quarto). For help, visit https://rmarkdown.rstudio.com/lesson-1.html and https://bookdown.org/yihui/rmarkdown/. It is a good idea to learn a system for writing proper reports, also for future studies and life after graduation. Incomplete or unreadable submissions may result in no points awarded.
- Answer the problems in the order given.
- Consult Moodle's general instructions and grading criteria before starting.
- Main textbook: ISLR_v2 ("An Introduction to Statistical Learning with Applications in R", 2nd edition, James et al.) or ISLP (Python version)—either is acceptable. You may use other materials as well.
- Submit as early as possible: you can revise and resubmit until the deadline. Due to peer review, we cannot accept late submissions. If you miss the deadline (and don't even submit empty PDF) you will loose points from the exercise set, including peer review points. Check in advance, well before the deadline date, that you can produce legible PDF files (especially with Jupyter notebooks students have often had last-minute surprises as PDF generation has now worked as expected). You have been warned.
- Work at your own pace, but refer to the study schedule on Moodle to avoid last-day rush.
- If you require special arrangements (e.g., sudden illness), inform staff promptly. By default, we assume you have followed the study schedule up to the time of notification; if you contact us at the last moment, we assume you have had time to do all exercises before the force majeure reason.
- If you find the problems difficult:
  - Follow the study schedule. Solving tasks after corresponding lectures reduces difficulty and stress.
  - Attend lectures and read the recommended textbook chapters.
  - Participate in exercise workshops and ask for help in workshops or the Team.
  - Talk with fellow students.

Full details on peer review, grading expectations, and evaluation criteria for master's-level work are found in the separate Exercise and Peer Review Instructions file.

Problem 1

[6 points]

Objective: familiarity with tools, basic description of the data set, familiarisation with the term project data

In this problem, you will preprocess and explore a data set about new particle formation. The data set is, incidentally, the same one you use to train your model in the term project.

Lab section 2.3 of ISLR_v2 (or ISLP) contains useful information for solving this problem.

The instructions below are in R and Python. Before reading the data into R or Python, you can view it in Excel or a text editor. For a good book about the topic, we recommend R for Data Science.

To use Python in R Markdown, you *might* need to specify the path to your Python executable using `reticulate::use_python(path_to_python)`, unless your default Python environment has everything installed.

Task a

Visit the term project Kaggle page following the instructions on the term project Moodle page and download the `train.csv` data set.

Next, read `train.csv` into a data frame and familiarize yourself with the data. You may want to read the data description available in the SMEAR research group's Xwiki.

Drop the columns `["id", "partlybad"]`.

Task b

Select the columns `["T84.mean","UV_A.mean","CS.mean"]` from the data frame and print their summary statistics.

Tip: In Python you can use `pd.describe()`. In R you can use `summary()`.

Task c

Extract the data in the column `T84.mean` of the data frame to an array. Calculate the mean and standard deviation of this array.

Tip: In Python you can use `.values`. In R, you can extract a column from a data frame `df` into an array using `df$T84.mean` or `df[,"T84.mean"]`.

Note: In Python, did you notice a difference between the mean and standard deviation calculated using pandas in Task b and numpy? The latter uses `float.64` as the default `dtype` whereas pandas uses `float.32` for computation. As such, numpy often returns more precise results for computations.

Task d

Produce side-by-side plots of:

- a bar plot of `class4`.
- a histogram of `CO242.mean`.

Tip: In Python, you can use `bars()` and `hist()` in the `matplotlib` package, and `df["class4"].value_counts` to calculate unique item counts. In R, you can use `barplot` and `hist`. The command `par(mfrow=c(1,2))` divides the plot window into two regions so that you can visualize the 2 plots simultaneously. To get class frequencies (number of unique items), you can use `table(df$class4)`.

Task e

Produce a scatterplot matrix of the variables `["UV_A.mean","T84.mean","H2O84.mean"]`.

Tip: In Python you can use `seaborn.pairplot()`. In R, you can use `pairs()`.

Task f

In this task you will make an initial *dummy* submission to the term project Kaggle competition (the link to the competition is in the course `Moodle page>Materials>Term project`). Construct a dummy predictor and submit the predictions to the Kaggle competition (see discussion below for details on the dummy model). In the competition, we ask you to predict both the class of NPF event ("class4") and the probability ("p") that *any* kind of event occurs. For the dummy predictor, the predicted class ("class4") should be the most commonly observed class in the training data. To estimate the probability of an event ("p"), use the relative frequency of "class4" != "nonevent" observations in the training data.

For this task, please use the full dataset which you get from Kaggle and read the instructions on the competition page carefully. Report the score your "model" gets in the public leaderboard.

About the dummy model

A *dummy model* is a supervised learning model that gives the same constant output regardless of the values of the covariates. The outcome might be such that it minimises the loss of the training data. For example, if we have logistic regression, this would be the most frequent class in the training data. Including a dummy model in model comparison is often helpful because it costs you almost nothing and gives you a good baseline for the performance of your "real" supervised learning models.

In addition to the dummy model, it is always helpful to include a simple *baseline*, such as OLS (ordinary least squares) linear regression for regression problems and logistic regression for classification problems. It happens surprisingly often that your complex machine-learning model does not substantially outperform the simple baseline model.

Problem 2

[8 points]

Learning objective: learning linear regression, concrete use of validation set, k-fold cross-validation, using regression models from ML libraries, and generalisation

In this problem, you will fit regression models and study their losses. One of the purposes of this problem - in addition to theory - is to make you more comfortable with various machine learning workflows.

Sections 5.1 and 5.3.2 (lab section) of ISLR_v2 contain helpful information for solving this problem.

Tasks a-b use a synthetic data set, and Task c uses a real data set:

- The synthetic data are given in the CSV files `train_syn`, `valid_syn`, and `test_syn` (the *training set*, *validation set*, and *test set* respectively).
- The real data are meteorological forecasts and geographic data from Cho et al. (2020)[1]. They are given in the CSV files `train_real` and `test_real` (the *training set* and *test set* respectively).

Task a

In this task, you will fit polynomials $\hat{y} = \sum_{k=0}^{p} w_k x^k$ to the synthetic data for several polynomial degrees p by using ordinary least squares (OLS) regression.

Produce the following table:

Degree	Train	Validation	Test	TestTRVA	CV
0	?	?	?	?	?
1	?	?	?	?	?
2	?	?	?	?	?
3	?	?	?	?	?
4	?	?	?	?	?
5	?	?	?	?	?
6	?	?	?	?	?
7	?	?	?	?	?
8	?	?	?	?	?

where:

- `Train` is the training loss (train model on training set, report error on *training* set)
- `Validation` is the validation loss (train model on training set, report error on *validation* set)
- `Test` is the testing loss (train model on training set, report error on *test* set)
- `TestTRVA` is another testing loss (train model on the *combined training and validation data*, report error on test set)
- `CV` is the MSE from 10-fold cross-validation on the combined training and validation data

Use MSE as the loss function.

Explain how you would choose the polynomial order if given a combined training and validation set when the losses on the test set would be unknown.

Task b

For each value of $p \in \{0, 1, 2, 3, 4, 8\}$, produce a plot showing the points (x_i, y_i) in the training set and the fitted polynomial in the interval $[-3, 3]$.

[1] Cho, D., Yoo, C., Im, J., Cha, D., 2020. Comparative Assessment of Various Machine Learning-Based Bias Correction Methods for Numerical Weather Prediction Model Forecasts of Extreme Air Temperatures in Urban Areas. Earth and Space Science 7. https://doi.org/10.1029/2019EA000740

Make sure to plot continuous curves for the polynomials (using, e.g., `x <- seq(from=-3, to=3, length.out=256)` in R or `x = np.linspace(start=-3, stop=3, num=256)` in Python) and not only lines connecting the values of x appearing in your data!

Task c

In this task, you will fit the following regressors to the real data to predict the next day's maximum temperature (variable `Next_Tmax`):

- dummy model (see the discussion below)
- OLS linear regression (simple baseline)
- random forest (RF)
- support vector regression (SVR)
- one more regression model implemented in your machine learning library not mentioned above.

Produce the following table:

Regressor	Train	Test	CV
Dummy	?	?	?
OLS	?	?	?
RF	?	?	?
SVR	?	?	?
?	?	?	?

where `Train` is the training (RMSE) loss, `Test` is the testing loss, and `CV` is the loss for 10-fold cross-validation.

Using the table, answer the following:

1. Which regressor is the best? Why?
2. How does `Train` compare to `Test`? How does `CV` compare to `Test`?
3. How can you improve the performance of these regressors (on this training set)?

About the dummy model

A *dummy model* is a supervised learning model that gives the same constant output regardless of the values of the covariates. The outcome might be such that it minimises the loss of the training data. For example, if we have OLS regression, this would be the mean of the dependent variable. Including the dummy model in your list is often helpful because it costs you almost nothing and gives you a good baseline for the performance of your "real" supervised learning models.

In addition to the dummy model, it is always helpful to include a simple *baseline*, such as OLS linear regression for regression problems and logistic regression for classification problems. It happens surprisingly often that your complex machine-learning model does not substantially outperform the simple baseline model.

Problem 3

[6 points]

Learning objectives: bias and variance and model flexibility

In this problem, you will study the bias-variance decomposition in the context of model selection.

Section 2.2 of ISLR_v2 will be helpful in solving this problem.

Task a

Describe the typical behaviour of the following terms, as we go from less flexible to more flexible statistical learning methods:

- training error and testing error
- (squared) bias
- variance
- irreducible (or Bayes) error.

Explain why each term has the described behaviour.

You can describe the behaviours in words or you can sketch them as curves. In the sketches the x-axis should represent the flexibility of the method, and the y-axis should represent the values for each term. There should be five curves in total so make sure to label each one.

Task b

In this task, you will test the bias-variance trade-off in practice using polynomial functions.

Assume a data point (x, y) is generated as $y = f(x) + \epsilon$ where $f(x) = -2 - x + 0.5x^2$, $\epsilon \sim \text{Normal}(0, 0.4^2)$, and $x \sim \text{Uniform}(-3, 3)$. Assume a polynomial regression function $\hat{f}$ of degree p is trained using a data set D of n data points (x, y).

According to Eq. (2.7) of ISLR_v2, you can decompose the expected squared loss at $x = 0$ as a sum of irreducible error, the bias term, and the variance term as

$$\text{squared loss} = \text{irreducible} + \text{bias}^2 + \text{var},$$

or:

$$E_D\left[(y_0 - \hat{f}_0)^2\right] = E_D\left[(y_0 - f_0)^2\right] + \left(E_D[\hat{f}_0] - f_0\right)^2 + E_D\left[(\hat{f}_0 - E_D[\hat{f}_0])^2\right],$$

where $f_0 = f(0)$ is the true function value at $x = 0$ and $\hat{f}_0 = \hat{f}(0)$ is the regression model prediction at $x = 0$. The expectation E_D is over the training data set D.

(i) Produce the following table:

Degree	Irreducible	BiasSq	Variance	Total	MSE
0	?	?	?	?	?
1	?	?	?	?	?
2	?	?	?	?	?
3	?	?	?	?	?
4	?	?	?	?	?
5	?	?	?	?	?
6	?	?	?	?	?

where

- `Degree` is the polynomial degree of the regression function.

- `Irreducible`, `BiasSq`, `Variance` are as described above. `MSE` is the mean squared error.
- `Total` is the sum of `Irreducible`, `BiasSq`, `Variance`.

(ii) Plot these four terms (squared loss, irreducible error, bias term, variance term) as a function of polynomial degree (i.e. make four curves).

(iii) Do the terms behave as you would expect from the discussion in Task a? Does `Total` approximately equal `MSE`?

How to compute the terms

To compute the expectations for degree p (one row in the table):

- Generate 1000 *training* sets each of which contains $n = 10$ data items, using the data generating process described above.
- For each training set:
  - train a polynomial regression function $\hat{f}(x)$ with degree p.
  - generate one *test* data point at $x = 0$, i.e., $(0, y_0)$, where $y_0 = f(0) + \epsilon$ where f and ϵ are as described above.
  - save the following numbers: $f(0)$, y_0, and $\hat{f}(0)$.

If you construct a table with 1000 rows and three columns $f(0)$, y_0 and $\hat{f}(0)$, then you can easily estimate the required expectations.

Problem 4 (Hard)

[6 points]

Topic: theoretical properties of generalisation loss and OLS linear regression [Ch. 2-3]

This is a more complex problem. While the proofs needed are quite short, they can feel a bit technical if you are not used to these kinds of problems. It is recommended that you start solving this problem only after you have solved the other problems.

Consider a linear regression model $\hat{f}(\mathbf{x}) = \hat{\beta}^T \mathbf{x}$, where $\hat{\beta} \in \mathbb{R}^p$ is fit by ordinary least squares (OLS) to a set of training data $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, where the pairs $(\mathbf{x}_i, y_i)$ have been drawn at random with replacement from a finite population, where $\mathbf{x}_i \in \mathbb{R}^p$, $y_i \in \mathbb{R}$, and $i \in \{1, \ldots, n\}$. Suppose we have a testing data $(\overline{\mathbf{x}}_1, \overline{y}_1), \ldots, (\overline{\mathbf{x}}_m, \overline{y}_m)$ drawn in the same way from the same population. Denote

$$L_{train} = \frac{1}{n} \sum\nolimits_{i=1}^{n} \left(y_i - \hat{\beta}^T \mathbf{x}_i \right)^2$$

and

$$L_{test} = \frac{1}{m} \sum\nolimits_{i=1}^{m} \left(\overline{y}_i - \hat{\beta}^T \overline{\mathbf{x}}_i \right)^2 .$$

The expectations below are defined over resampling the training and testing data.

Task a

Prove that

$$E\left[L_{test}\right] = E\left[\left(\overline{y}_1 - \hat{\beta}^T \overline{\mathbf{x}}_1 \right)^2 \right] .$$

Task b

Prove that L_{test} is an unbiased estimate of the generalisation error for the OLS regression.

Task c

Prove that $E\left[L_{train}\right] \leq E\left[L_{test}\right]$.

Task d

Explain how the task result above relates to the machine learning generalisation problem.

About expectations

There are (at least) two ways to define the expectations $E\left[L_{train}\right]$ and $E\left[L_{test}\right]$:

1. sampling the testing data while keeping the training data fixed, in which case $\hat{\beta}$ is constant, resulting in $E\left[L_{test}\right]$ being the generalisation error for this particular training data and regression solution.
2. sampling both the training and testing data, in which $\hat{\beta}$ is a random variable and a function of the training data, resulting in $E\left[L_{test}\right]$ being the generalisation error averaged over all possible training data sets.

It is sometimes tricky to keep track of which is a random variable and over what expectations to take, and often, this is not explicitly mentioned. Both of the definitions above make sense but have slightly different interpretations. **Please use the second definition in this task.** You can read, e.g., Bengio et al. (2004)[2] for a more in-depth discussion of the expectations if interested, where "PE" of Eq. (1) corresponds to the first and "EPE" of Eq. (2) corresponds to the second definition.

[2] Bengio, Y., Grandvalet, Y., 2004. No unbiased estimator of the variance of K-fold cross-validation. J. Mach. Learn. Res. 5, 1089-1105. https://www.jmlr.org/papers/v5/grandvalet04a.html

Problem 5

[6 points]

Objective: properties of estimators [Ch 3]

So far, we have tried only to estimate the loss (to choose the best model). It is also essential to understand the model and diagnose potential problems.

In this problem, we study 1-variable linear regression $\hat{y} = w_0 + w_1 x$ using the four data sets named `d1.csv`, `d2.csv`, `d3.csv`, and `d4.csv`.

Task a

For each data set, fit an OLS linear regression and report:

- the intercept term estimate, standard error, and p-value,
- the slope term estimate, standard error, and p-value,
- the R-squared value of the model.

The slope term may be positive (or negative) with high confidence. Can you safely conclude that when `x` increases (or decreases), `y` tends to increase (and vice versa)?

Tip: See Section 3.6.2 of ISLR_v2.

Task b

Make a plot of each data set showing a scatterplot of `x` vs `y` along with the fitted regression line. What commonalities do you notice between the data sets and their fitted models?

Task c

Sect. 3.3.3 of ISLR_v2 lists six potential problems with linear regression models. Which of the six problems would (potentially) apply to each dataset? What tricks and plots did you use to detect and diagnose the problems? Produce at least one diagnostic plot that shows these problems (other than just plotting `x` vs `y`, which you can do here because the data is 1-dimensional and which would not work for higher-dimensional data sets).

Problem 6

[6 points]

Objective: properties of estimators and Bootstrap [Ch 3 & 5.2]

Task a

Compute the standard errors for the regression coefficient estimates for the data set `d2.csv` of the previous problem using bootstrap. Compare the bootstrap standard errors to the ones you got in Task A of the previous problem; which of the estimates is more trustworthy and why?

Task b

Describe briefly in your own words how the bootstrap algorithm computes the standard errors for the intercept and slope parameters in the task above.

Task c

In bootstrap, you sample n data points from a population of n points with replacement. Argue that the probability that the jth observation is *not* in the bootstrap sample is about 0.368 when n is very large.

Hints

Please see lab Section 5.3.4 of ISLR_v2 or 5.3.3 of ISLP, subsection "Estimating the Accuracy of a Linear Regression Model", for guidance for Task a. For Task c, you can find lots of hints in Problem 2 in Section 5 of ISLR_v2, and by observing that $\lim_{n\to\infty}(1-1/n)^n = 1/e \approx 0.368$.

Problem 7

[2 points]

Objectives: self-reflection, giving feedback on the course

Task a

Write a learning diary of the topics of lectures 1-4 and this exercise set.

Guiding questions: What did I learn? What did I not understand? Was there something relevant for other studies or (future) work? The length of your reply should be 1-3 paragraphs of text. You can also give feedback on the course.

Task b

Give an estimate of the hours used in solving the problems in this exercise set.