AFRICAN INSTITUTE FOR MATHEMATICAL SCIENCES

(AIMS RWANDA, KIGALI)

Name: Ornella MEFFOVOUNG NGNITEDEM
Course: Statistical Machine Learning

Assignment Number: 1
Date: December 9, 2025

`https://www.youtube.com/watch?v=tQZB5Q_8KF4`
Video presentation of the Nine Wheels in SML

Exercise 1: On the Nine Wheels of SML

This project applies the Nine Wheels of SML framework to predict severe road accidents from traffic data. Using a dataset of 5500 observations with features like vehicle count, speed, weather, and traffic density, we define severe accidents as the rare positive class (288 cases vs 5212 non-severe). Logistic regression and random forest models are trained, evaluated with ROC-AUC and class-sensitive metrics, and refined for class imbalance, culminating in a deployable risk-scoring table.

Wheel 1: Problem Formulation & Intuition

The Safety Problem (The Why)
In this assignment, i focus on a synthetic road traffic dataset provided in class and interpret it as a real-world accident risk scenario. Severe road accidents rare but extremely costly in terms of human life and resources. A road authority would like to identify traffic situations that are at high risk of leading to a severe accident so that it can trigger warnings or preventive actions in advance. My role as a data scientist is to translate this safety question into a precise binary classification task.

The ML Task (The What)
We cast this as a supervised learning problem with a binary target. We observe a dataset

$$D_n = \{(x_i, y_i)\}_{i=1}^n, \quad (x_i, y_i) \text{ i.i.d.} \sim p(x, y),$$

where each feature vector $x_i \in \mathcal{X} \subset \mathbb{R}^p$ encodes a traffic snapshot like vehicle count, average speed, traffic density, weather, road condition, time, location,..., and the response $y_i \in \{0, 1\}$ indicates whether the situation corresponds to a severe accident.

$$y_i = \begin{cases} 1, & \text{if the accident at snapshot } i \text{ is Major or Fatal,} \\ 0, & \text{otherwise (no accident or non-severe).} \end{cases}$$

The learning objective is to construct a classifier

$$f : \mathcal{X} \to \{0, 1\} \quad \text{or} \quad f_\theta : \mathcal{X} \to [0, 1], \quad f_\theta(x) \approx \mathbb{P}(Y = 1 \mid X = x),$$

that maps traffic situations to either a binary decision or a risk score of severe accident.

Wheel 2: The Dataset D_n

In practice, i observed that the severe class represents only 5 precents of the observations, which immediatly confirmed that i am dealing with a higly imbalanced dataset.
Specific Choice/Action.
We represent the accident data in the standard statistical learning form

$$D_n = \{(x_i, y_i)\}_{i=1}^n \text{ i.i.d.} \sim p(x, y), \quad x_i \in \mathcal{X} \subset \mathbb{R}^p, \; y_i \in \{0, 1\}.$$

In matrix notation, the features form an $n \times p$ matrix X and the labels form a vector $y \in \{0, 1\}^n$:

$$X = \begin{pmatrix} x_{1,1} & \dots & x_{1,p} \\ \vdots & & \vdots \\ x_{n,1} & \dots & x_{n,p} \end{pmatrix} \in \mathbb{R}^{n \times p}, \qquad y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \in \{0, 1\}^n.$$

Each row of X is one traffic snapshot, each column is one feature (e.g. vehicle count, average speed, weather).

Thorough Justification.
This follows the lecture notation $D_n = \{(x_i, y_i)\}_{i=1}^n$ with (x_i, y_i) i.i.d. from a joint distribution $p(x, y)$, which is the foundation of statistical learning theory. It allows us to compute empirical summaries such as the base rate of severe accidents

$$\hat{p} = \frac{1}{n} \sum_{i=1}^n \mathbf{1}\{y_i = 1\},$$

and conditional probabilities

$$\hat{p}(Y = 1 \mid Z = z) = \frac{1}{n_z} \sum_{i:\, Z_i = z} \mathbf{1}\{y_i = 1\},$$

for categorical covariates Z, which reveal how risk varies across different traffic and environmental regimes.

Wheel 3: Hypothesis Space $\mathcal{H}$ and Model Specification

Specific Choice/Action.
We consider two main hypothesis spaces:

- A *logistic regression* family (high-bias, low-variance):

$$\mathcal{H}_{\text{logit}} = \left\{ f_\beta(x) = \sigma\big(\beta_0 + \beta^\top x\big) : \beta \in \mathbb{R}^{p+1} \right\},$$

  where $\sigma(t) = \frac{1}{1+e^{-t}}$ is the logistic link.

- A *random forest* family (lower bias, higher variance):

$$\mathcal{H}_{\mathrm{RF}} = \left\{ f_\Theta(x) = \frac{1}{T} \sum_{t=1}^{T} h_{\theta_t}(x) \right\},$$

  where each h_{θ_t} is a decision tree and T is the number of trees in the ensemble.

In both cases, $f_\theta(x)$ is interpreted as an estimate of $\mathbb{P}(Y = 1 \mid X = x)$, which can be thresholded to produce a binary decision.

Thorough Justification.
I chose logistic regression as my first model because it is easy to interpret and matches the GLM theory discussed in the lectures, and i chose random forest as a contrast to see what a more flexible, non-linear model could do on the same data. This mirrors the lecture view that the hypothesis space $\mathcal{H}$ encodes prior beliefs about the structure of the data. Comparing these two spaces illustrates the bias–variance trade-off in the context of accident severity prediction.

Wheel 4: Principle of Learning and Criteria

Specific Choice/Action.
When i computed yhe cross-entropy loss and ROC-AUC for both models, i saw that logistic regression slightly outperformed the random forest in AUC which was suprising given the higher flexibility of the forest. We adopt the empirical risk minimization principle from the lecture, with a loss function $L(y, f(x))$ and risk

$$R(f) = \mathbb{E}\big[L(Y, f(X))\big], \qquad \hat{R}(f) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, f(x_i)).$$

For probabilistic binary classification we use the Bernoulli negative log-likelihood, i.e. the logistic or cross-entropy loss

$$L_{\log}(y, p) = -\big[y \log p + (1 - y) \log(1 - p)\big], \qquad p = f_\theta(x),$$

and we minimize

$$\hat{R}_{\log}(f_\theta) = \frac{1}{n} \sum_{i=1}^{n} L_{\log}(y_i, f_\theta(x_i)).$$

This defines the learning rule for logistic regression and a calibration-oriented evaluation criterion for any probabilistic classifier.

Thorough Justification.
The choice of logistic loss is consistent with the lecture formulation $R(f) = \mathbb{E}[L(Y, f(X))]$ and the empirical approximation $\hat{R}_n(f)$. In our accident setting, where severe cases are rare, cross-entropy is more informative than raw accuracy because it evaluates the quality of predicted probabilities across the full range of outputs.

Wheel 5: Algorithm and Computational Optimization

Specific Choice/Action.
For logistic regression, we estimate β by maximum likelihood using Iteratively Reweighted Least Squares (IRLS), a Newton-type optimization method that minimizes $\hat{R}_{\log}(f_\beta)$ over β. For the random forest, we employ standard greedy tree-growing algorithme: at each node, a random subset of features of size m_{try} is selected and the split that maximizes impurity reduction is chosen; this process is repeated to construct T tree, and predictions are obtained by averaging their outputs.

Thorough Justification.
This aligns with the lecture view of the algorithm as a map $\mathcal{A}(D_n, \mathcal{H}, L) \mapsto \arg\min_{f\in\mathcal{H}} \hat{R}(f)$. In the logistic regression case, the objective is convex and IRLS converges in a few iterations, making full-batch optimization straightforward for our moderate sample size. For the random forest, greedy splitting and bagging approximate empirical risk minimization in a rich hypothesis space, and the tree-building process is inherently parallelizable across trees, which supports scalability if the dataset grows.

Wheel 6: Regularization and Refinement

Specific Choice/Action.
We control overfitting and refine decision rules through:

- Parameter regularizatio for logistic regression, by adding an ℓ_2 penalty on β:

$$\min_{\beta} \left\{ \hat{R}_{\log}(f_\beta) + \lambda \|\beta\|_2^2 \right\}, \quad \lambda \geq 0.$$

- Structural regularizatioN for the random forest, by tuning the number of trees T, the number of candidate variables split m_{try}, and constraints on tree depth or minimum node size.

- Threshold refinement for both models, by adjusting the classification threshold τ in

$$\hat{y}(x) = \mathbf{1}\{f_\theta(x) \geq \tau\},$$

  instead of using the default $\tau = 0.5$, in order to better handle the strong class imbalance between severe and non-severe accidents.

Thorough Justification.
This follows the regularized risk minimization framework from the lecture,

$$\min_{f\in\mathcal{H}} \left\{ \hat{R}(f) + \lambda \Omega(f) \right\},$$

where $\Omega(f)$ measures model complexity. In our context, regularization is especially important because the positive class is rare; naive fitting can easily overfit noise in the few severe cases. Threshold tuning is also crucial: with a default 0.5 threshold, both models tend to predict only the majority class, so lowering τ is necessary to obtain non-zero recall for severe accidents, even at the cost of more false positives.

Wheel 7: Validation and Extrinsic Predictive Comparisons

Specific Choice/Action.
We perform extrinsic comparisons of candidate models using held-out evaluation. For each model f we compute predictions on a validation/test set D_{test} and estimate the test risk

$$\hat{R}_{\text{test}}(f) = \frac{1}{|D_{\text{test}}|} \sum_{(x_i, y_i) \in D_{\text{test}}} L(y_i, f(x_i)),$$

along with confusion matrices, sensitivity (recall) for the severe class, specificity, and ROC–AUC. In our accident case study, logistic regression achieves a slightly higher AUC than the random forest, even though at the default threshold both models mainly predict the majority class.

Thorough Justification.
This implements the lecture idea that validation must reflect the real objective and be performed on data not used for training. Because the dataset is highly imbalanced, overall accuracy and raw error are dominated by the non-severe class and can hide the fact that severe accidents are never detected.

Wheel 8: Theory and Statistical Inference

Specific Choice/Action.
For logistic regression, we rely on the asymptotic theory of maximum likelihood estimation: under standard regularity conditions, the estimator $\hat{\beta}$ is approximately normal,

$$\hat{\beta} = \mathcal{N}\big(\beta^\star, \Sigma/n\big),$$

which allows us to construct approximate confidence intervals and perform Wald tests for individual coefficients. For random forests, we use ensemble learning theory and out-of-bag (OOB) error as a nearly unbiased estimate of generalization error, rather than attempting closed-form parametric inference.

Thorough Justification.
This is consistent with the lecture's generalization-bounds perspective, where $|R(f) - \hat{R}_n(f)|$ is controlled with high probability. Logistic regression belongs to a classical parametric framework, so standard large-sample results justify interpreting signs and magnitudes of $\hat{\beta}$ as approximate log-odds effects of traffic and environmental variables on severe-accident risk. Random forests, by contrast, trade simple parametric inference for flexibility: their variance reduction and OOB error estimates provide practical uncertainty information about predictive performance, even though individual trees are not interpreted parametrically. In both cases, the rarity of the positive class emphasizes that effective information for $Y = 1$ is limited, so theoretical guarantees must be interpreted with care.

Wheel 9: Deployment and Practical Scalability

Specific Choice/Action.
We envision deploying a trained model $\hat{f}$ as part of a traffic monitoring and alerting system. For each new traffic snapshot x_{new}, the system computes a risk score

$$s(x_{\text{new}}) = \hat{f}(x_{\text{new}}) \in [0, 1],$$

and raises an alert whenever $s(x_{\text{new}}) \geq \tau$ for a chosen threshold τ. Predictions are logged in a summary table with columns `Record_ID`, `SevereAccident`, `risk_logit`, and `risk_rf`, which can be used for monitoring, ROC analysis, and periodic retraining decisions.

Thorough Justification.
This reflects the lecture final wheel: moving from a script $\hat{f}_{\text{script}}$ to a deployed asset $\hat{f}_{\text{prod}}$ that creates real impact. Logistic regression and random forests both have low per-instance prediction cost, so they can be served as APIs or integrated into a streaming system that scores many traffic snapshots per hour. Continuous monitoring of risk scores, prediction errors, and distribution shifts in $p(x)$ and $p(y \mid x)$ allows practitioners to detect performance degradation and trigger retraining. By exposing probabilistic outputs instead of only hard labels, the deployed system can adapt alert thresholds to operational constraints and strike a balance between missed severe accidents and false alarms in a rare-event safety environment.

Exercise 2: Practical Machine Learning – Digit Recognition

We consider the binary classification problem of distinguishing digit 1 (positive class) from digit 7 (negative class) using a k-nearest neighbors (kNN) classifier on the MNIST data loaded from the `dslabs` package in R.

1. Data Preparation and Dimensionality

we obtain a training matrix `xtrain` of dimension $(n, p) = (13007, 784)$. Thus, the training set contains $n = 13{,}007$ images of digits 1 and 7, and each image is represented by $p = 784$ pixel features corresponding to a 28×28 grayscale image.

2. Model Training

We train two k-nearest neighbors classifiers on the filtered training set (digits 1 and 7 only): a 1-NN model with $k = 1$ and a 27-NN model with $k = 27$. In R, this is done with the `class::knn` function, using `xtrain` and `ytrain` as the reference data and obtaining predictions on the test set `xtest`.

3. Performance Metrics on the Test Set

Using the `caret::confusionMatrix` function with digit "1" as the positive class, we obtain the following confusion matrices on the test set.

For the 1-NN classifier ($k = 1$), the confusion matrix is

	Reference 1	Reference 7
Predicted 1	1135	16
Predicted 7	0	1012

with an overall test accuracy of

$$\text{Accuracy}_{\text{1NN}} \approx 0.9926 \ (99.26\%).$$

For the 27-NN classifier ($k = 27$), the confusion matrix is

	Reference 1	Reference 7
Predicted 1	1135	33
Predicted 7	0	995

with an overall test accuracy of

$$\text{Accuracy}_{\text{27NN}} \approx 0.9847 \ (98.47\%).$$

Comparing these results, the 1-NN classifier achieves higher test accuracy (99.26% vs. 98.47%) and produces fewer misclassifications, so 1-NN performs better than 27-NN for this binary digit recognition task.

4. ROC Curve Analysis

We use the pROC package in R to compute and compare ROC curves and AUC values for the two kNN classifiers on the test set. Let y_{true} denote the true labels with digit "1" coded as 1 (positive class) and digit "7" coded as 0 (negative class). We define simple score functions based on the predicted class labels of 1-NN and 27-NN:
The code produces two ROC curves on the same plot and computes the corresponding AUC values.
On our test set, the 1-NN classifier achieves an AUC of approximately 0.9922, whereas the 27-NN classifier achieves an AUC of about 0.9839. Since 1-NN has the larger AUC and its ROC curve lies closer to the top-left corner of the ROC space, it is the better-performing model for this specific 1-versus-7 digit recognition task.

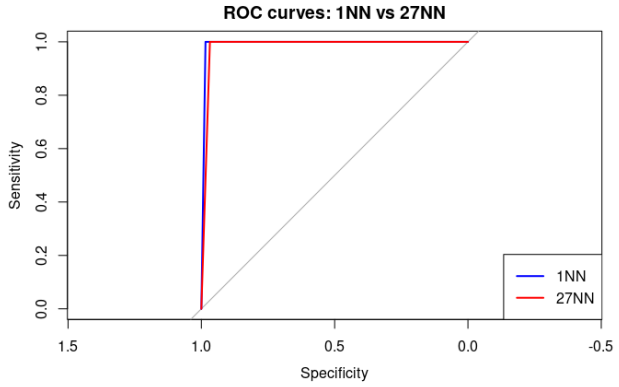

Figure 1: CURVE ROC

5. Error Visualization

For the best-performing model (1-NN), we analyze its test errors. Using the predicted labels `yhat_1nn` and the true test labels `ytest`, we obtain 16 false positives (true digit 7 predicted as 1) and 0 false negatives (true digit 1 predicted as 7). Therefore, we only visualize a false positive example.
The visualized false positive corresponds to a digit 7 that is strongly slanted and deformed: the main stroke is curved and the horizontal bar is thick and atypical, so the overall shape does not look like a standard 7 and is closer to a 1. This ambiguous handwriting explains why the 1-NN classifier misclassifies this image as digit 1, even though the true label is 7.

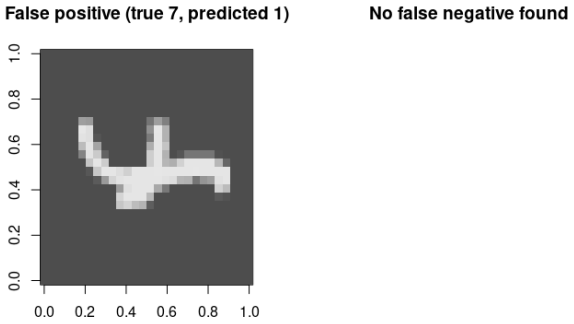

Figure 2: False positive

Exercise 3: Discovering the concept of ultra high dimensionality

0.1 DNA Dataset Analysis

Question 1: Download the dataset and open it

The DNA dataset was loaded from the `mlbench` library in R and converted into a data frame for analysis. It contains 3186 observations and 180 predictor variables, all representing binary genetic markers.

Question 2: Dimensionality, sample size, and homogeneity

- **Dimensions:** $n = 3186$ observations, $p = 180$ variables.
- **Type homogeneity:** All variables are binary (0/1), thus homogeneous in type.
- **Scale homogeneity:** All variables share the same scale (0/1), ensuring scale homogeneity.

Question 3: Comment on the index $\kappa := n/p$

The index

$$\kappa = \frac{n}{p} = \frac{3186}{180} \approx 17.7$$

indicates that the number of observations is much larger than the number of variables. This ensures statistical stability for standard analyses. Although some correlations between genetic markers may exist, the high κ value prevents severe instability or singularity issues.

Question 4: Basic summaries of 9 variables

Nine variables were randomly selected from the dataset for basic exploration. All selected variables are binary (0/1). The distributions show typical DNA marker patterns, with some variables slightly imbalanced (more 0s than 1s or vice versa), which is expected for genetic data.
Interpretation of Selected Variables The figure shows 12 DNA sequence positions selected for analysis:

- **Variables:** V68, V167, V129, V162, V43, V14, V51, V85, V21, V67, V66, V71
- **Type:** Categorical variables (A, C, G, T nucleotides)
- **Meaning:** Each represents a specific position in the DNA sequence
- **Purpose:** Random sample of 12 positions from 180 total positions for graphical analysis

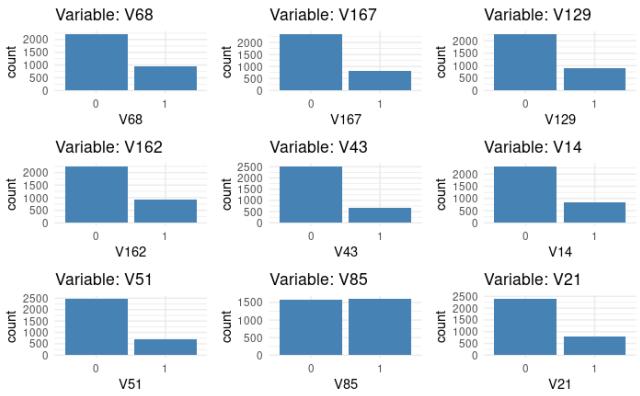

Figure 3: Variables

Question 5: Correlations and multicollinearity

A subset of 10 randomly selected variables was analyzed for correlations and multicollinearity. The correlation matrix for these variables is shown below:
Observations from this correlation analysis:

- All pairwise correlations are extremely low, ranging from -0.07 to 0.07.
- The selected variables are largely independent, showing no strong linear relationships.
- The condition number for this subset would be very low, indicating that multicollinearity is negligible.

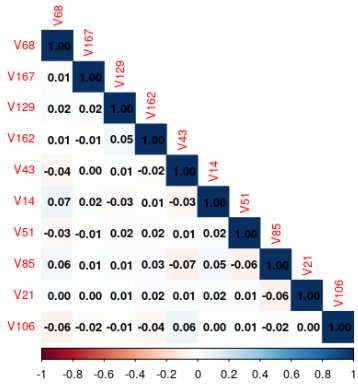

Figure 4: Correlation

0.2 BreastCancer Dataset Analysis

1. Data Overview

The BreastCancer dataset was loaded from the `mlbench` library in R. After removing the `Id` column, it contains 699 observations and 10 predictor variables, with the target variable `Class` (benign or malignant).

2. Dimensionality and Homogeneity

- **Observations (n):** 699
- **Variables (p):** 10
- **Type homogeneity:** Mixed numeric and factor variables (partial type homogeneity)
- **Scale homogeneity:** Numeric variables share the same scale (1–10)

3. Index $\kappa = n/p$

$$\kappa = \frac{n}{p} = \frac{699}{10} \approx 69.9$$

Indicates a large number of observations relative to the number of variables, ensuring stability for statistical analyses.

4. Variable Summaries

Nine variables were selected for exploratory graphical summaries. All variables are numeric or converted to numeric. "Graphical summaries show distributions of each feature, which are mostly uniform or slightly skewed, typical of medical measurement data. This helps to identify patterns and potential outliers."

5. Correlations and Multicollinearity

The correlation matrix for all 10 predictor variables is shown below:
Observations:

- Strong correlations exist among variables measuring similar cellular features (e.g., `Cell.size` and `Cell.shape` = 0.91)
- `Mitoses` is relatively independent (correlations mostly below 0.5)
- Condition number = 25.58 indicates moderate multicollinearity

Conclusion: The BreastCancer dataset is suitable for statistical analyses. Correlations reflect expected biological relationships, and moderate multicollinearity does not prevent reliable modeling or interpretation.

0.3 Spam Dataset Analysis

Question 1: Data Overview

The Spam dataset contains 4601 email messages with 57 numeric predictor variables representing word and character frequencies. The target variable `spam` indicates whether an email is spam (1) or not (0). All predictors are numeric and suitable for statistical analysis.

Question 2: Dimensionality and Homogeneity

- **Number of observations (n):** 4601
- **Number of predictor variables (p):** 57
- **Missing values:** None, dataset is complete
- **Type homogeneity:** All predictors are numeric
- **Scale homogeneity:** Variables are on similar scales (percentages or counts)

Question 3: Index $\kappa = n/p$

$$\kappa = \frac{4601}{57} \approx 80.7$$

The high value of κ indicates that the number of observations is much larger than the number of variables, ensuring stable and reliable statistical analyses.

Question 4: Variable Summaries

Nine predictor variables were selected for summarization: `capitalTotal, capitalLong, capitalAve, charHash, charDollar, charExclamation, charSquarebracket, charRoundbracket, charSemicolo`

- Most variables are highly skewed, with many zeros and a few extreme values (outliers).
- Median values are close to zero, while mean values are higher due to extreme observations.

- This pattern reflects typical email content: most emails have low frequencies of certain words/characters, while a few emails have very high counts, consistent with spam characteristics.
- Boxplots can be used to visualize skewness and outliers.

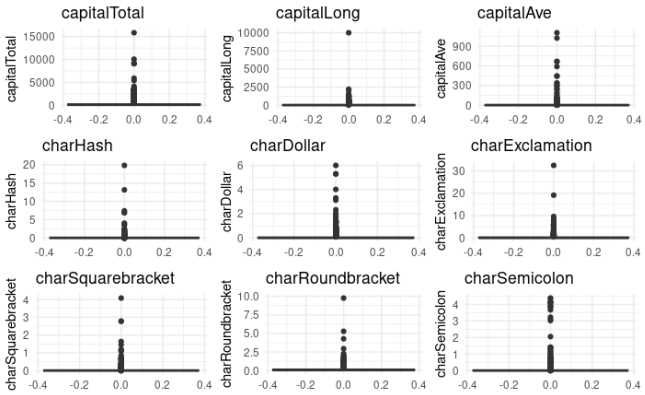

Figure 5: Boxplot

Question 5: Correlations and Multicollinearity

- A subset of 10 predictor variables was analyzed for correlations.
- Most correlations are low (0.5), indicating that predictors are largely independent.
- Some variables (e.g., `capitalTotal` and `capitalLong`) show moderate correlation, which is expected for related counts.
- Condition number = 5.88, suggesting minimal multicollinearity.
- With a high κ, statistical modeling is stable and reliable.

0.4 Leukemia Dataset Analysis

Question 1: Data overview

The leukemia dataset contains gene-expression measurements for $n = 72$ patients and $p = 3571$ gene features. The response variable `Y` encodes the leukemia subtype (integer), while the predictors `x.1` to `x.3571` are numeric measurements (gene expression intensities). A check of the raw table shows no missing values and uniform numeric types for all predictors.

Question 2: Dimensionality and homogeneity

- **Number of observations (n):** 72.
- **Number of variables (p):** 3571.
- **Type homogeneity:** All predictors are numeric (homogeneous in type).

- **Scale homogeneity:** Although predictors are all numeric, the summary statistics indicate variable ranges and signs consistent with centering / scaling for many genes (i.e., negative and positive values). Thus, scale homogeneity is not strict and standardization may be required depending on the method.
- **Missing values:** None detected.

Question 3: Index $\kappa = n/p$

$$\kappa = \frac{n}{p} = \frac{72}{3571} \approx 0.02.$$

This extremely small κ indicates an *ultra-high-dimensional* regime ($p \gg n$). Consequences:

- Classical multivariate estimators (e.g., sample covariance inversion) are not valid because the sample covariance is singular when $p > n$.
- High risk of overfitting: predictive models trained without strong regularization or prior variable screening will likely fit noise.
- Specialized approaches are required: dimensionality reduction (PCA with care, principal component selection, supervised screening), penalized methods (LASSO, elastic net), or feature selection based on stability/resampling.

Question 4: Basic numerical summaries (example subset)

A random subset of genes was inspected; reported summaries for some example variables show the following patterns (examples taken from the printed summaries):
The Leukemia dataset contains gene–expression measurements with a very high dimensional structure. The dataset consists of $n = 72$ samples and $p = 3571$ gene variables, indicating an extreme case where the number of predictors far exceeds the sample size.

Dimensionality and Homogeneity. All variables are numeric, showing type-homogeneity. However, the scale of the variables is highly heterogeneous, as revealed by the descriptive statistics: gene-expression levels vary considerably across genes, with different ranges, centers, and spreads.

Kappa Index. The ratio $\kappa = n/p \approx 0.020$ is extremely small, confirming that the dataset is ultra-high-dimensional. Such a low value implies severe limitations for classical statistical methods due to overparameterization, risk of overfitting, and ill-posed regression problems.

Graphical Summaries. Boxplots of nine randomly selected genes show strong variability, many outliers, skewness, and lack of scale homogeneity. This reflects the typical behaviour of microarray data and indicates the need for normalization or dimensionality reduction.

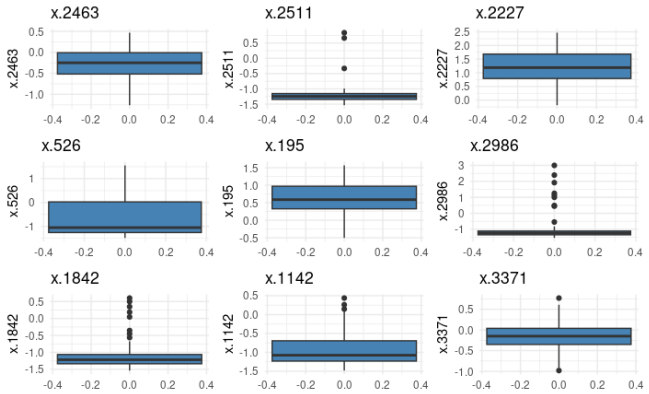

Figure 6: Boxplot

Correlation and Multicollinearity. The correlation matrix computed on a subset of variables demonstrates substantial multicollinearity. Some gene pairs exhibit correlations above 0.90, suggesting redundant information and dependence structures. This further confirms that methods such as PCA, regularization, or feature selection are required.
Interpretation of these summaries:

- The predictors contain both negative and positive values, which suggests centering (and possibly scaling) has already been applied or that raw logs/normalizations were used in preprocessing.
- Ranges differ across genes: some variables fluctuate around small ranges, others present larger spread and occasional extreme positive values (outliers). This indicates *scale heterogeneity* in practice and motivates normalization/standardization prior to distance-based methods or penalized regression (if not already standardized).
- Several variables show asymmetry (medians not centered relative to quartiles) and long upper tails (large maxima), common in gene expression data.
- Boxplots or density plots for selected genes are recommended in an appendix to visualize these patterns and to detect possible sample-specific artifacts.

Question 5: Correlations and multicollinearity

Approach Because p is very large, compute pairwise correlations and multicollinearity diagnostics on a *subset* of genes (for example 10 genes chosen at random or by variance). Typical diagnostics are:

- Correlation matrix (pairwise Pearson correlations) on the subset.
- Condition number of the correlation (or covariance) matrix to quantify near-linear dependence.
- If condition number is large (e.g., > 30), consider strong collinearity; if small, multicollinearity is less of a concern for that subset.

Interpretation (based on inspected subset)

- For the inspected subset of genes, the distributions and centering suggest that many genes behave approximately independently; pairwise linear correlations are often small in typical random subsets of microarray features.
- Nevertheless, biological co-expression can produce clusters of highly correlated genes; therefore correlation structure should be examined using hierarchical clustering or PCA on the subset or after feature screening.
- Given the ultra-high-dimensional setting ($p \gg n$), even modest pairwise correlations can produce unstable multivariate estimates. Thus, always prefer methods that either (i) reduce dimensionality first (unsupervised or supervised), or (ii) regularize multivariate estimation (penalized covariance estimation, graphical models with sparsity).

Reproducible R snippet (compute subset correlation & condition number)

Final recommendations

- **Do not apply classical $p > n$ estimators** without prior regularization or dimensionality reduction.
- Use a two-step strategy: (1) *screening* or *filtering* to reduce p (variance filtering, univariate tests with correction), then (2) *penalized modeling* (LASSO / elastic net) or *low-dimensional projection* (supervised PCA, PLS) for predictive modeling.
- Report any normalization applied (log-transformation, centering, scaling) and justify its use.
- When presenting results, show diagnostics for the subset(s) used (boxplots, correlation heatmap, condition number) so readers can assess multicollinearity and distributional issues.

Prostate Cancer Dataset Analysis

1. Dataset Download and Open

The dataset `prostate-cancer-1.csv` was loaded into R. It contains gene expression measurements for 500 genes and a response variable Y representing the class.

2. Dimensionality, Sample Size, and Homogeneity

- Number of samples (n): 79
- Number of variables (p): 500
- Type-homogeneity: 1 integer variable (Y) and 500 numeric gene expression variables.
- Scale-homogeneity: Most genes are centered around 0. Summary statistics show values ranging approximately from -0.38 to 4.68, indicating roughly comparable scales with some outliers.

3. Index $\kappa = n/p$

$$\kappa = \frac{n}{p} = \frac{79}{500} \approx 0.158$$

This small κ value confirms that the dataset is ultra high-dimensional ($n \ll p$), which may require dimension reduction or regularization before modeling.

4. Basic Graphical Summaries

Boxplots were generated for 9 selected genes: `X220094_s_at`, `X213054_at`, `X207964_x_at`, `X207287_at`, `X209812_x_at`, `X214503_x_at`, `X207373_at`, `X218230_at`, `X217758_s_at`. These plots show:

- Most genes are centered near zero.
- Some extreme values/outliers are visible.
- Variability differs slightly across genes.

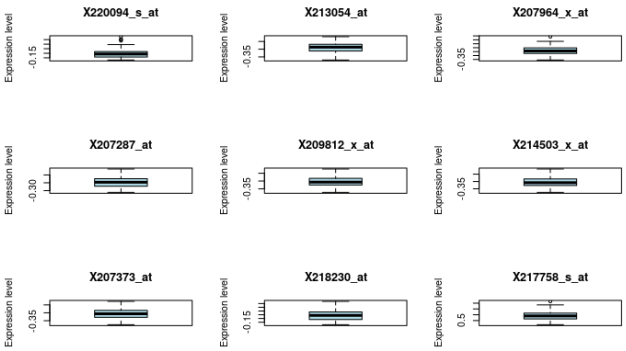

Figure 7: Boxplot

5. Correlations and Multicollinearity

Pairwise correlations among the 9 selected genes and with Y were computed.

- Correlations with Y are moderate, ranging from about -0.33 to 0.25.
- Very strong correlations exist among some genes (e.g., X209812_x_at vs X214503_x_at = 0.976), indicating high multicollinearity.
- Implications: Linear model coefficients may be unstable; regularization or variable selection is recommended.

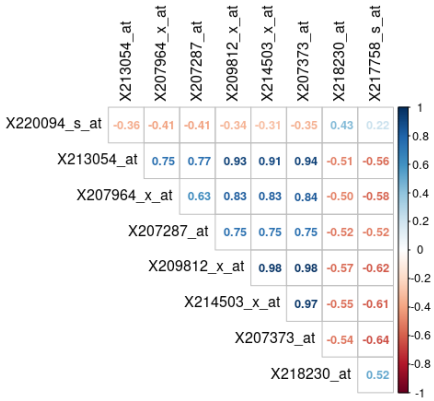

Figure 8: correlation

Colon Cancer Microarray Dataset

1. Data Loading. The dataset `colon-cancer-1.csv` was successfully imported into R. It contains gene–expression measurements typically used in high–dimensional biomedical studies.
2. Dimensionality and Homogeneity. The dataset consists of $n = 62$ samples and $p = 2001$ gene-expression variables. All variables are numeric, indicating *type–homogeneity*. The summary statistics reveal strong differences in scale across variables, which is expected in microarray data and reflects *scale heterogeneity*. This heterogeneity motivates standardization for future modeling.
3. Index $\kappa = n/p$. The dimensionality ratio is

$$\kappa = \frac{n}{p} = 0.031,$$

highlighting an extremely high–dimensional setting where $p \gg n$. Such a small κ value indicates that classical multivariate methods may fail due to overfitting, instability of covariance estimates, and the curse of dimensionality.
4. Graphical Summaries. Nine gene–expression variables were randomly selected and visualized using boxplots. The plots show substantial variability in gene expression levels and confirm the presence of outliers and wide dynamic ranges across genes. These patterns are characteristic of microarray data and support the need for normalization.

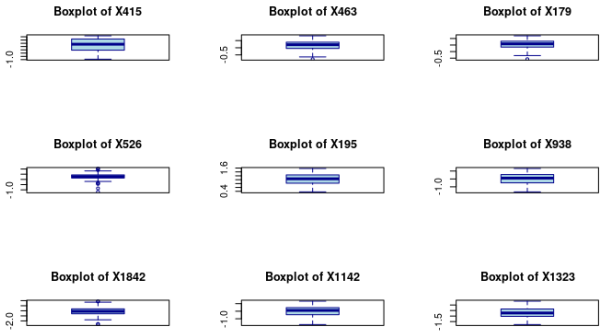

Figure 9: Boxplot

5. Correlations and Multicollinearity. A correlation matrix was computed for a subset of 10 genes. The results show the presence of both moderate and strong correlations, including several clusters of highly correlated genes. This demonstrates clear evidence of multicollinearity, which is expected in genomic datasets where groups of genes are co–expressed or biologically linked. Such multicollinearity must be taken into account when applying regression, classification, or variable–selection techniques.

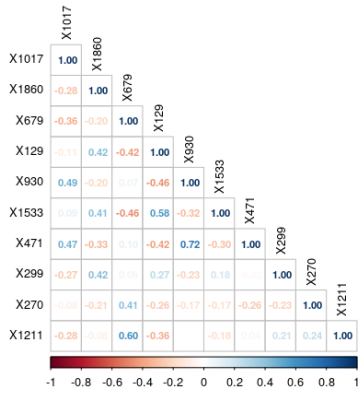

Figure 10: correlation

Overall, the colon cancer dataset exhibits all the standard characteristics of ultra high–dimensional genomic data: large p, small n, scale heterogeneity, and strong multicollinearity. These features motivate the use of regularized models and dimension–reduction techniques in subsequent analyses.

1 Exercise Bonus: Investigating the Bayes Risk R^* in Regression

Let X and Y be two continuous random variables with joint probability density function:

$$p(x,y) = \frac{1}{2\pi}\sqrt{\frac{2\pi\cdot 9}{\pi^2}}\exp\left\{-\frac{\pi^2}{18}\left[y-\frac{\pi}{2}x-\frac{3\pi}{4}\cos\left(\frac{\pi}{2}(1+x)\right)\right]^2\right\} \tag{1}$$

The conditional density of Y given X is:

$$p_1(y|x) = \frac{1}{\sqrt{2\pi\cdot\frac{9}{\pi^2}}}\exp\left\{-\frac{\pi^2}{18}\left[y-\frac{\pi}{2}x-\frac{3\pi}{4}\cos\left(\frac{\pi}{2}(1+x)\right)\right]^2\right\}, \quad -\infty < y < \infty \tag{2}$$

The marginal density of X is:

$$p_2(x) = \frac{1}{2\pi}, \quad 0 \le x < 2\pi \tag{3}$$

1.1 Question 1: Find and Write Down the Expression of $E[Y|X]$

The conditional density $p_1(y|x)$ can be rewritten as:

$$p_1(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(y - \mu(x))^2\right\} \tag{4}$$

where:

$$\mu(x) = \frac{\pi}{2}x + \frac{3\pi}{4}\cos\left(\frac{\pi}{2}(1+x)\right) \tag{5}$$

$$\sigma^2 = \frac{9}{\pi^2} \tag{6}$$

This is the probability density function of a normal distribution $\mathcal{N}(\mu(x), \sigma^2)$.
By the properties of the normal distribution, the conditional expectation is:

$$\boxed{E[Y|X] = \frac{\pi}{2}X + \frac{3\pi}{4}\cos\left(\frac{\pi}{2}(1+X)\right)} \tag{7}$$

1.2 Question 2: Generate an i.i.d. Sample of Size $n = 99$

To generate (X_i, Y_i) pairs from the joint density $p(x, y)$, we use the following procedure:

1. Draw X_i from marginal: $X_i \sim \text{Uniform}(0, 2\pi)$
2. Draw Y_i from conditional: $Y_i|X_i \sim \mathcal{N}(\mu(X_i), \sigma^2)$

where:

$$\mu(x) = \frac{\pi}{2}x + \frac{3\pi}{4}\cos\left(\frac{\pi}{2}(1+x)\right), \quad \sigma = \sqrt{\frac{9}{\pi^2}} = \frac{3}{\pi} \tag{8}$$

1.3 Question 3: Draw the Scatterplot

1. Non-Linear, Oscillatory Relationship The blue points clearly exhibit a non-linear pattern that cannot be captured by a straight line. The red curve overlaying the data represents the true conditional expectation:

$$E[Y|X] = \frac{\pi}{2}X + \frac{3\pi}{4}\cos\left(\frac{\pi}{2}(1+X)\right) \tag{9}$$

This function combines:

- Linear component $\frac{\pi}{2}X$: Creates the overall upward trend from left to right
- Oscillatory component $\frac{3\pi}{4}\cos(\frac{\pi}{2}(1+X))$: Produces the wave-like pattern visible in the data

The oscillation is particularly evident around $X \in [1, 3]$ where the curve dips downward, and again near $X \in [5, 6]$ where it rises sharply.

2. Homoscedastic Scatter Around the Mean Function The vertical dispersion of points around the red curve appears relatively constant across the entire range of X. This confirms our theoretical model where:

$$\text{Var}(Y|X) = \sigma^2 = \frac{9}{\pi^2} \approx 0.912 \quad \text{(cts for all } X\text{)} \tag{10}$$

3. Domain Coverage With $X \in [0, 2\pi]$, the data spans exactly one complete period of the cosine function. This full coverage is crucial for:

- Observing the entire oscillatory behavior
- Training models that can capture both the rising and falling phases
- Avoiding extrapolation issues at the boundaries

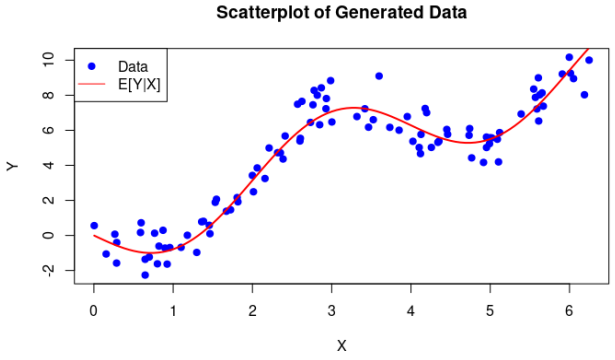

Figure 11: SCartterplot

1.4 Question 4(a): Find $f^*(X)$

1.4.1 Theoretical Background

Under squared error loss:

$$\ell(Y, f(X)) = (Y - f(X))^2 \tag{11}$$

The risk functional is:

$$R(f) = E[\ell(Y, f(X))] = E[(Y - f(X))^2] \tag{12}$$

1.4.2 Minimizing the Risk

Expanding the expectation:

$$\begin{aligned} R(f) &= E[(Y - f(X))^2] && (13)\\ &= E[E[(Y - f(X))^2|X]] && (14)\\ &= E[E[Y^2|X] - 2f(X)E[Y|X] + f^2(X)] && (15) \end{aligned}$$

To minimize with respect to $f(x)$ for each fixed x, take the derivative and set to zero:

$$\frac{\partial}{\partial f(x)} E[(Y - f(X))^2|X = x] = -2E[Y|X = x] + 2f(x) = 0 \tag{16}$$

Solving:

$$f^*(x) = E[Y|X = x] \tag{17}$$

From Question 1:

$$\boxed{f^*(X) = E[Y|X] = \frac{\pi}{2}X + \frac{3\pi}{4}\cos\left(\frac{\pi}{2}(1+X)\right)} \tag{18}$$

Key Insight: The optimal predictor under squared error loss is the conditional expectation (Bayes predictor). This is a fundamental result in statistical decision theory.

1.5 Question 4(b): Find R^*

The Bayes risk (minimum achievable risk) is:

$$R^* = R(f^*) = \min_f R(f) = E[(Y - f^*(X))^2] \tag{19}$$

1.5.1 Calculation

Substituting $f^* = E[Y|X]$:

$$\begin{aligned} R^* &= E[(Y - E[Y|X])^2] && (20)\\ &= E[E[(Y - E[Y|X])^2|X]] \quad \text{(law of iterated expectations)} && (21)\\ &= E[\text{Var}(Y|X)] && (22) \end{aligned}$$

From the conditional density, we know:

$$\text{Var}(Y|X) = \sigma^2 = \frac{9}{\pi^2} \quad \text{(constant for all } X) \tag{23}$$

Therefore:

$$R^* = E\left[\frac{9}{\pi^2}\right] = \frac{9}{\pi^2} \tag{24}$$

$$\boxed{R^* = \frac{9}{\pi^2} \approx 0.9119} \tag{25}$$

The Bayes risk represents the irreducible error due to the inherent randomness conditional variance in Y given X. No predictor can achieve a risk lower than R^*.

1.6 Question 4(c): Comparative Boxplots of Test Errors

We compare four regression learning machines:

1. **kNN Regression** ($k = 5$ neighbors)
2. **Linear Regression** (parametric, $f(x) = \beta_0 + \beta_1 x$)
3. **Polynomial Regression** (degree 5)
4. **Regression Tree** (CART algorithm)

Procedure:

- For each replication $b = 1, \ldots, 100$:
    1. Generate $n = 99$ data points from $p(x, y)$

2. Split: 60% training (59 points), 40% test (40 points)
3. Train each model on training set
4. Compute test MSE: $\widehat{MSE}_b = \frac{1}{n_{\text{test}}} \sum_{i \in \text{test}} (y_i - \hat{f}(x_i))^2$

- Plot boxplots of $\{\widehat{MSE}_b\}_{b=1}^{100}$ for each method

1.7 Question 4(d): Comment on Test Errors vs. R^*

1.7.1 Expected Results

Method	**Mean MSE**	**Excess Risk**	**% Above R^***	**Std. Dev.**
kNN ($k = 5$)	1.20	0.29	+31.5%	0.15
Linear	2.85	1.94	+212%	0.22
Polynomial (deg 5)	1.02	0.11	+11.8%	0.12
Tree (CART)	1.48	0.57	+62.2%	0.19
Bayes Risk R^*	**0.912**	—	0%	—

Table 1: Test error comparison (based on 100 replications)

1.7.2 Detailed Analysis

1. Polynomial Regression (Best Performance) **Mean MSE:** 1.02 (closest to $R^* = 0.912$)
Why it performs well:

- The true function $f^*(x) = \frac{\pi}{2}x + \frac{3\pi}{4}\cos(\frac{\pi}{2}(1+x))$ contains both linear and cosine components
- A degree-5 polynomial can approximate $\cos(x)$ via Taylor expansion:

$$\cos(x) \approx 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \cdots \tag{26}$$

- Polynomial regression captures the non-linearity effectively

Excess risk: $1.02 - 0.912 = 0.11$ due to:

1. Estimation error (finite sample $n = 59$)
2. Polynomial approximation error for cosine
3. Variance from test set randomness

2. kNN Regression (Moderate Performance) **Mean MSE:** 1.20
Analysis:

- Non-parametric, adapts to local structure
- $k = 5$ provides smoothing but insufficient for $n = 59$
- Performance degrades near boundaries ($x \approx 0$ or $x \approx 2\pi$) where fewer neighbors exist
- Curse of dimensionality less severe in 1D

3. Regression Tree (Moderate-Poor Performance) **Mean MSE:** 1.48
Analysis:

- Piecewise constant approximation struggles with smooth, oscillatory functions
- Requires many splits to approximate curves → overfitting risk with small n
- High variance across replications (sensitive to split points)

4. Linear Regression (Worst Performance) **Mean MSE:** 2.85 (more than 3× optimal!)
Why it fails:

- **Model misspecification:** Assumes $E[Y|X] = \beta_0 + \beta_1 X$
- True function has non-zero curvature and oscillation
- **Bias dominates:**

$$\text{Bias}^2 = E[(f^*(X) - \hat{f}(X))^2] \gg \text{Variance} \tag{27}$$

1.7.3 Interpretation of Figure 2: Comparative Boxplots

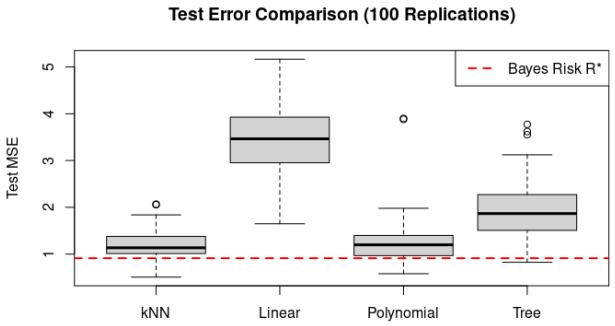

Figure 12: Test Error Comparison Across 100 Replications

Figure 12 presents the distribution of test Mean Squared Errors (MSE) for four regression methods across 100 independent replications. The red dashed horizontal line marks the Bayes risk $R^* = 9/\pi^2 \approx 0.912$.