Advanced Statistical Analysis

Assignment 1

과제는 pdf 파일로 제출하며 Rmarkdown 혹은 Quarto를 이용하여 만든 파일을 권장. 마크다운에 익숙하지 않은 경우 word등 다른 문서 편집기로 편집 후 pdf로 변환하여 제출.

Problem 1

이 과제에서는 프로젝트 세팅과 상대경로 입출력 GitHub 업로드를 실습한다. R에서 프로젝트를 만들고 아래와 같은 구조를 잡는다:

- 루트(프로젝트 폴더)
  - `data/` : 데이터 파일(CSV 등) 저장
  - `plots/` : 그림 파일(PNG 등) 저장
  - `R/` : R 스크립트(R, Rmd, qmd 등) 저장

회귀함수 추정을 위한 간단한 예제 데이터를 생성한다.

```r
set.seed(1)
n = 200
x = seq(0, 1, length.out = n)
y = sin(2*pi*x) + rnorm(n, sd = 0.15)
```

아래 모든 문제에서 프로젝트 루트폴더를 기준으로 상대경로를 사용한다.

(a) 위 데이터를 `data/` 경로에 csv로 저장하는 코드를 작성하시오.

(b) `R/` 폴더 아래 R 스크립트를 작성하여, 위에서 저장한 데이터를 불러오고, `ggplot2`를 바탕으로 산점도와 회귀곡선 적합을 시각화하시오 (`ggplot2`의 `goem_smooth()`에서 적절한 method 선택).

(c) `ggsave()` 함수를 이용하여 작성한 플랏을 `plots/` 폴더에 저장하는 코드를 작성하시오.

(d) R에서 Git 버전 컨트롤을 사용하여 `commit`하고 본인의 GitHub에 올린 후 **GitHub 저장소 링크** 를 같이 제출하시오.

Problem 2

(a) 버블 정렬은 기본적으로 배열의 인접한 두 수를 선택한 뒤, 만약 그 두 수가 정렬되었다면 놔두고 아니라면 두 수를 바꾸는 방식으로 진행된다. 오름차순으로 정렬한다면 배열의 첫번째 원소에서 반복을 시작하여 2개의 항을 읽어 앞의 항이 뒤의 항보다 큰 경우 두 항을 교환한다. 이 작업을 반복하여 마지막 2개 항까지 수행하고 나면 가장 큰 항이 배열의 제일 뒤에 위치하게 된다. 다시 처음 원소에서 시작하여 마지막 항을 제외하고 반복을 수행하면 두번째로 큰 항이 배열의 끝에서 두번째 위치에 배치된다. 이렇게 모든 원소가 정렬될 때까지 반복을 수행하면 되며, 알고리즘은 조기에 종료될 수 있다.

버블 정렬을 R 함수로 구현하고, 아래 테스트 케이스를 사용하여 결과를 확인하시오. 오름차순과 내림차순을 옵션으로 받을 수 있게 하여 두 경우 모두 결과를 제시하시오.

```r
set.seed(1)
x = runif(10)
```

(b) 다른 기본적인 정렬 알고리즘에 비해 평균적으로 빠르게 작업을 수행하는 퀵 정렬을 구현하고자 한다. 퀵 정렬은 분할 정복 방법을 통해 배열의 원소를 정렬한다. 배열에서 하나의 원소를 고른 후 (이를 pivot이라고 한다) pivot의 왼쪽과 오른쪽으로 배열을 둘로 분할한다. 이때 오름차순 정렬의 경우, 분할된 배열에서 pivot보다 작은 원소들은 왼쪽에, pivot보다 큰 원소들은 오른쪽에 위치한다. 이후 재귀적으로 나누어진 분할에 대해 이 과정을 반복한다. 재귀적 반복은 대상 배열의 크기가 0이나 1이 될 때까지 반복한다. 재귀 호출이 한번 수행될 때 최소 하나의 원소는 위치가 정해지게되므로, 이 알고리즘의 수렴이 보장된다.

퀵 정렬을 R 함수로 구현하고, 아래 테스트 케이스를 사용하여 결과를 확인하시오. 오름차순과 내림차순을 옵션으로 받을 수 있게 하여 두 경우 모두 결과를 제시하시오.

```r
set.seed(1)
x = runif(10)
```

Problem 3

수치 미분은 극한을 이용한 도함수의 정의에서 h를 작은 값을 설정하여 근사시켜 계산한다. 구체적으로 한 점 x에서 함수 f의 미분을 근사하는 방법은 다음과 같다.

- **전진차분**:
  $f'(x) \approx \frac{f(x+h)-f(x)}{h}$ (오차: $O(h)$)
- **후진차분**:
  $f'(x) \approx \frac{f(x)-f(x-h)}{h}$ (오차: $O(h)$)
- **중심차분**:
  $f'(x) \approx \frac{f(x+h)-f(x-h)}{2h}$ (오차: $O(h^2)$)

(a) 수치 미분함수를 구현하시오. 이때 함수의 인자로는 대상 함수 `f`와 미분값이 계산되는 점인 `x`, `h` 값과, `method`를 설정한다. 이때 `h=1e-6`을 디폴트 옵션으로 사용하고 `method`는 "forward", "backward", "central" 중에서 선택할 수 있도록 구현한다. 구현한 함수를 이용해 $f(x) = cos(x) - x$ 함수를 $[0, 2\pi]$에서 미분하고, 해석적인 도함수 계산과 비교하여 시각화 하시오. (이때 구간을 100개의 점으로 균등 분할하여 x 값을 생성한다.)

(b) Newton–Rapshon 방법은 비선형 식 $f(x) = 0$의 해를 찾기 위한 반복 알고리즘이다. 초깃값을 x_0라고 할 때, 이 알고리즘은

$$x_t = x_{t-1} - \frac{f(x_{t-1})}{f'(x_{t-1})}$$

공식을 이용하여 수렴할 때까지 반복을 수행한다. 수렴성을 체크하는 기준 중 하나로 $|x_t - x_{t-1}| < \varepsilon$을 작은 ε 값에 대해 체크하여 차이가 작은 경우 알고리즘을 종료시키는 것이다.

위 알고리즘을 R 함수로 구현하시오. 함수의 인자는 `f` 함수와, `fprime` 함수 (defualt=NULL), 초깃값 `x0`, `maxiter (defualt = 100)`, `h(defualt = 1e-6)`, `epsilon (defualt = 1e-10)`으로 설정하시오. `fprime`은 도함수로 사용자가 입력하는 형식으로 작성하고, 만약 도함수가 제공되지 않으면 `h` 인자와 `central` method를 사용하여 수치미분으로 해를 찾는 방식으로 구현한다.

(c) $f(x) = cos(x) - x = 0$을 만족하는 해를 찾으려고 한다. 위에서 구현한 함수를 바탕으로 수치미분 버전과, 도함수 제공 버전으로 코드를 실행하고 결과를 제시하시오. 초깃값 $x_0 = 0.5$를 사용한다.

Problem 4

수치적분의 가장 간단한 형태는 구분구적법의 유한차원 구현으로 이해할 수 있으며, 구체적인 형태는 다음과 같다. 구간 $[a, b]$, 등간격 $h = \frac{b-a}{n}$, 분할점 $x_i = a + ih$, $f_i = f(x_i)$.

- Left Rectangle: $\int_a^b f(x)\,dx \approx h\sum_{i=0}^{n-1} f_i$ (오차 $O(h)$)
- Trapezoid: $\displaystyle\int_a^b f(x)\,dx \approx \frac{h}{2}(f_0 + 2\sum_{i=1}^{n-1} f_i + f_n)$ (오차 $O(h^2)$)
- Simpson(n 짝수): $\displaystyle\int_a^b f(x)\,dx \approx \frac{h}{3}\Big(f_0 + 4\sum_{i\ \text{odd}} f_i + 2\sum_{i\ \text{even},\, i\neq n} f_i + f_n\Big)$ (오차 $O(h^4)$)

(a) Left Rectangle 방식을 R코드로 구현하시오. 함수의 인자로 적분대상 함수 `f`와 적분 구간 `a`, `b`, 그리고 `n`을 입력 받는다.

(b) Trapezoid 방식을 R코드로 구현하시오. 함수의 인자는 위와 같다.

(c) Simpson 방식을 R코드로 구현하시오. 함수의 인자는 위와 같다.

(d) `sin(x)` 함수를 $[0, \pi]$ 구간에서 적분한 값을 세 개의 알고리즘으로 계산하시오. (n = 100으로 설정한다.)

(e) 해석적으로 구한 값과 위 알고리즘의 차이를 $n = 10, 30, 60, 100, 150, 200$에 대해 계산하고 알고리즘 비교를 위한 시각화를 수행하시오.

Problem 5

선형방정식 $Ax = b$를 푸는 알고리즘을 구현하고자 한다.

(a) A가 양정치 행렬인 경우 Cholesky 분해를 통해 $A = LL^\top$를 만족하는 하삼각행렬 L을 구할 수 있다. R에서 `chol()` 함수를 이용하면 $A = U^\top U$를 만족하는 상삼각행렬을 반환하며, 이를 transpose하여 L을 구할 수 있다. 아래 행렬 A에 해당하는 L을 구하고, $LL^\top$을 출력하여 A와 같음을 확인하시오. (rounding error 허용)

```r
A = matrix(c(4,2,2,2,5,1,2,1,3), 3)
```

(b) 하삼각행렬 L에 대해 $Lz = b$를 푸는 알고리즘은 다음과 같다. i-th 식은 아래와 같다 $(i = 1, \dots, n)$:

$$\sum_{j=1}^{i} l_{ij} z_j = y_i,$$

따라서 z의 계산은 다음과 같이 수행할 수 있다 $(i = 1, \dots, n)$:

$$z_i = \frac{1}{l_{ii}} \left(y_i - \sum_{j=1}^{i-1} l_{ij} z_j \right).$$

이 식을 풀기위한 `forward` 함수를 작성하시오. (힌트: R의 `forwardsovle` 함수와 결과를 비교할 수 있다.)

(c) 한편 $L^\top \alpha = z$을 푸는 방식은 다음과 같다. i-th 식은 아래와 같다 $(i = 1, \dots, n)$:

$$\sum_{j=i}^{n} l_{ji} x_j = z_i,$$

따라서 z의 계산은 다음과 같이 수행할 수 있다 $(i = 1, \dots, n)$:

$$x_i = \frac{1}{l_{ii}} \left(z_i - \sum_{j=i+1}^{n} l_{ji} x_j \right).$$

이 식을 풀기위한 `backward` 함수를 작성하시오. (힌트: R의 `backwardsolve` 함수와 결과를 비교할 수 있다.)

(d) (b)에서 작성한 `forward()` 함수와 (c)에서 작성한 `backward()` 함수를 이용하여, 다음 벡터

```
b = c(1, -2, 3)
```

에 대해 선형방정식 ($Ax = b$)를 푸시오. 구한 해를 R의 내장 함수 `solve(A, b)`로 구한 해와 비교하여, 두 결과가 일치함을 확인하시오. (rounding error 허용)

Problem 6

Kernel Ridge Regression (KRR)은 주어진 데이터에 기반하여 회귀함수를 추정하는 비모수적 방법이다. 훈련 데이터 $(x_i, y_i)_{i=1}^n$가 주어졌을 때, Gaussian kernel 함수

$$\mathcal{K}(x, x') = \exp(-\rho|x - x'|^2)$$

을 이용하여, 커널 행렬 $K \in \mathbb{R}^{n \times n}$를 계산할 수 있다. 이때 커널 행렬의 (i,j)-th 원소는 $K_{ij} = \mathcal{K}(x_i, x_j)$로 계산된다. 훈련 데이터 $(x_i, y_i)_{i=1}^n$, 커널행렬 K ($K_{ij} = \mathcal{K}(x_i, x_j)$), 규제 파라미터 $\lambda > 0$가 있을 때 계수 α는

$$\alpha = (K + \lambda I_n)^{-1} y$$

로 계산되며, 임의의 x에서의 예측값은

$$\hat{f}(x) = k(x, X)^\top \alpha$$

로 계산된다. 여기서 $k(x, X) = (\mathcal{K}(x, x_1), \ldots, \mathcal{K}(x, x_n))$이다.

(a) Gaussian kernel 함수를 계산하는 R 함수를 작성하시오. 이때 $\rho = 1$을 default로 사용한다.

(b) KRR을 적합하는 함수를 작성하시오. 이때 함수의 return에서 class를 `krr`로 정의하여 아래 문제에 활용하기 위한 class를 정의하시오.이때 $\lambda = 0.0001$을 default로 사용한다. (return 값은 새로운 데이터에 대한 예측과 시각화가 가능하도록 적절한 값들을 선택하여 작성한다.)

(c) `predict` 함수를 `krr` 클래스에 대해 확장하여 KRR 예측값을 계산해주는 함수를 작성하시오.

(d) `plot` 함수를 `krr` 클래스에 대해 확장하여 데이터의 산점도와 예측함수 $\hat{f}(x)$를 시각화하는 함수를 작성하시오.

(e) 아래의 데이터를 시뮬레이션하고, KRR을 적합한 이후 `predict`와 `plot` 함수를 통해 결과를 시각화하시오.

```r
set.seed(1)
n = 150
X = matrix(runif(n, -1, 1), ncol = 1)
ftrue = function(x) sin(2*pi*x) + 0.5*cos(4*pi*x)
y = ftrue(X[,1]) + rnorm(n, sd = 0.1)
```