Ⅲ. 데이터 분석

제1장. 데이터 분석 이해

1. 데이터 처리 과정

1. **데이터 처리 과정**
    - 데이터 분석을 위해 _ _ _ _ 나 _ _ _ _ 를 통해 분석데이터를 구성
    - 신규데이터나 DW에 없는 데이터는 _ _ _ _ _ _ _ _ _ _ _ (_ _ _ _ _ _)에서 가져오기보다는 임시로 저장하는 _ _ _ _ _ _ _ _ 영역에서 데이터를 _ _ _ _ 해서 운영데이터저장소(_ _ _ _)에 저장된 데이터를 _ _ _ _ 와 _ _ _ _ 과 결합하여 데이터를 구성

2. **시각화 기법**
3. **공간분석**
4. **탐색적 자료 분석(EDA)**
    - 다양한 _ _ _ _ 과 _ _ 을 조합해 가며 특이점이나 의미있는 사실을 도출하고 분석의 최종목적을 달성해가는 과정

5. **데이터마이닝**
    - 자료에 존재하는 _ _ _ _, _ _ _ _, _ _ _ _ 등을 탐색하고 이를 _ _ _ _ _ _ 함으로써 이전에 알지 못한 유용한 지식을 추출하는 방법.
    - 기계학습(_ _ _ _ _ _, _ _ _ _ _ _ _ _, _ _ _ _ _ _, _ _ _ _ _ _), 패턴인식(_ _ _ _ _ _, _ _ _ _ _ _ _ _)

6. **시뮬레이션**
    - 복잡한 실제상황을 _ _ _ _ _ _ 해 컴퓨터상의 _ _ _ _ 을 만들어 재현하거나 변경함으로써 현상을 보다 잘 이해하고 미래의 변화에 따른 결과를 예측하는데 사용하는 고급 기법.

7. **최적화**
    - _ _ _ _ _ _ 값을 최대화 또는 최소화하는 것을 목표로 하는 방법
    - _ _ _ _ _ _ 하에서 _ _ _ _ 값을 개선하는 방식으로 _ _ _ _ _ _ 와 _ _ _ _ _ _ 을 정의해 문제를 해결

제2장. 빅데이터 분석도구 R

1. R 개

1. **R프로그래밍 언어**

- 통계 계산과 그래픽을 위한 프로그래맹 언어이자 소프트웨어 환경.
- 뉴질랜드 오클랜드 대학의 _ _ _ _ _ _ _ _ _ _ 와 _ _ _ _ _ _ _ _ _ _ 에 의해 시작되어 현재 R 코어 팀이 개발.
- _ _ _ _ (_ _ _ _ _ _ _ _ _ _ _ _)하에 배포되는 S프로그래밍 언어로 _ _ _ _ _ _ 라고 함.

2. **R의 특징**

- 표준 플렛폼(_ _ 언어 기반)
- 모든 운영체제에서 사용 가능(맥, 리눅스, 윈도우)
- _ _ _ _ _ _ 저장방식
- _ _ _ _ _ _ _ _ 언어이며 _ _ _ _ _ _ 언어
- _ _ _ _ _ _ 프로그램으로 무료

3. **통계분석도구의 비교**

2. R 들어가기

1. **편리한 기능**

- 작업환경 설정 : _ _ _ _ _ _ ("작업 디렉토리")
- 도움말 : _ _ _ _ (함수), _ _ _ _ (함수), _ _ _ _ _ _ _ _ ("함수명")
- 히스토리 : _ _ _ _ _ _ (), _ _ _ _ _ _ _ _ (file="파일명"), _ _ _ _ _ _ _ _ _ _ (file="파일명")

2. **스크립트 실행하기**

- 실행 : _ _ _ _ _ + _ _
- 주석처리 : _ _ _ _

3. **패키지**

- 자동설치 : _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ ("패키지명")
- 수동설치 : _ _ _ _ _ _ _ _ _ _ _ _ _ _ _ ("패키지명", "패키지위치")

4. **배치 파일**

3. R 기초

1. **변수 다루기**

- R에서는 변수명만 선언하고 값을 할당하면 자료형태를 _ _ _ _ _ _ 인식하고 선언
- 대입연산자 _ _ _ _ 를 추천
- 불필요한 변수 확인을 위해 _ _ _ _ 를 활용, 삭제는 _ _ _ _ 을 사용

2. **기본적인 통계량 계산**

- 평균 : _ _ _ _ _ _ / 중간값 : _ _ _ _ _ _ / 표준편차 : _ _ _ _ _ _
- 분산 : _ _ _ _ _ _ / 공분산 : _ _ _ _ _ _ / 상관계수 : _ _ _ _ _ _

3. **함수의 생성 및 활용**

4. 입력과 출력

1. **데이터 입력과 출력**

- 부동소수점 표현시 7자리 수를 기본으로 세팅되어 있으며 _ _ _ _ _ _ 함수, _ _ _ _ ="숫자"를 지정하여 자리수 변경.
- 문자출력시 _ _ _ _ _ _ ("출력할 내용", file="파일명")
- 역슬레쉬()를 인식하지 못하므로 _ _ _ _ _ _ (_ _) 또는 _ _ _ _ _ _ (_ _)으로 파일 경로 지정.

2. **외부 파일 입력과 출력**

- 고정자리 변수 파일 : _ _ _ _ _ _ ("파일명", width=c(w1, w2, ...))
- 구분자 변수 파일 : _ _ _ _ _ _ _ ("파일명", sep="구분자")
- csv 파일 읽기 : _ _ _ _ _ _ _ _ ("파일명", header=T)
- csv 파일 출력 : _ _ _ _ _ _ _ _ (데이터 프레임, "파일명")

3. **웹 페이지에서 데이터 읽어오기**

- 파일 다운로드 : _ _ _ _ _ _ (http 주소)
- ftp 다운로드 : _ _ _ _ _ _ (ftp 주소)
- html에서 테이블 : library(_ _ _ _); _ _ _ _ _ _ _ _ ("url")

5. 데이터 구조

1. **데이터 구조의 정의**

- 단일값(_ _ _ _ _ _) : 원소가 하나인 벡터
- 행렬(_ _ _ _ _ _) : 차원을 가진 벡터로 인식
- 배열(_ _ _ _ _ _) : 3차원 또는 n차원까지 확대된 행렬
- 요인(_ _ _ _ _ _) : 고유값이 _ _ _ _ _ _ 으로 구성된 벡터

2. **리스트 다루기**

- 리스트 원소 선택 : _ _ _ _ , _ _ _ _ _ _ _ _ , _ _ _ _ _ _ _ _

3. **행렬 다루기**

- 행렬 설정 : _ _ _ _ _ _ <- c(2,3)
- 행 이름 붙이기 : _ _ _ _ _ _ (matrix) <- c("rowname1", "rowname2", ...)
- 열 이름 붙이기 : _ _ _ _ _ _ (matrix) <- c("column1", "column2", ...)

4. **데이터 구조 변경 방법**

- 벡터 → 리스트 : _ _ _ _ _ _ (vector)
- 벡터 → 행렬 : 1열짜리 행렬 : _ _ _ _ (vector) 또는 _ _ _ _ (vector) / 1행짜리 행렬 : _ _ _ _ _ _ (vector) / nm행렬 : _ _ _ _ _ _ (vecotr, n, m)
- 벡터 → 데이터프레임 : 1열짜리 데이터프레임 : _ _ _ _ _ _ _ _ (vector) / 1행짜리 데이터프레임 : _ _ _ _ _ _ _ _ (_ _ _ _ _ _)
- 리스트 → 벡터 : _ _ _ _ (list)
- 리스트 → 행렬 : 1열짜리 행렬 : _ _ _ _ _ _ (list) / 1행짜리 행렬 : _ _ _ _ _ _ (_ _ _ _ _ _) / nm행렬 : _ _ _ _ _ _ (list,n,m)
- 리스트 → 데이터프레임 : 원소들이 열 : _ _ _ _ _ _ _ _ (list) / 원소들이 행 : _ _ _ _ (_ _ _ _ _ _)
- 행렬 → 벡터 : _ _ _ _ _ _ _ _ (matrix)
- 행렬 → 리스트 : _ _ _ _ _ _ _ _ (matrix)

- 행렬 -> 데이터프레임 : _ _ _ _ _ _ _ _ _ (matrix)
- 데이터프레임 → 벡터 : 1열짜리 데이터프레임 : _ _ _ _ _ [[_ _]], _ _ _ _ [_ _] / 1행짜리 데이터프레임 : _ _ _ _ _ _ [_ _]
- 데이터프레임 → 리스트 : _ _ _ _ _ _ (datafram)
- 데이터프레임 → 행렬 : _ _ _ _ _ _ (dataframe)

6. 데이터 프레임

1. **집단으로 분할하기**

- 벡터 : _ _ _ _ (vector, factor) : 벡터와 팩터의 길이가 같아야 함
- 데이터프레임 : _ _ _ _ _ _ (dfm, fac)

2. **함수 적용하기**

- 행렬 : _ _ _ _ _ _ (matrix, 1, function), _ _ _ _ _ _ (matrix, 2, function)
- 리스트 : _ _ _ _ _ _ _ _ (list, function), _ _ _ _ _ _ _ _
- 데이터프레임 : _ _ _ _ _ _ (dataframe, function), _ _ _ _ _ _ (datagrame, function), _ _ _ _ _ _ (dataframe, function)

3. **집단별로 함수 적용하기**

- _ _ _ _ (vector, factor, function)
- _ _ _ _ (dataframe, factor, function)

4. **병렬 벡터들과 리스트에 함수 적용하기**

- 벡터 : _ _ _ _ (function, vector1,vector2, vector3, ...)
- 리스트 : _ _ _ _ (function, list1, list2, list3, ...)

7. 데이터 변환

1. **문자열 다루기**

- 문자열 길이 : _ _ _ _ _ _ ("문자열") / 벡터의 길이 : _ _ _ _ _ _ (vector)
- 문자열 연결하기 : _ _ _ _ _ _ ("단어", "문장", scalar), 하위 문자열 추출하기 : _ _ _ _ _ _ ("문자열", 시작번호, 끝번호)
- 구분자로 문자열 추출하기 : _ _ _ _ _ _ ("문자열", 구분자)
- 문자열 대체하기 : _ _ _ _ ("대상 문자열", "변경문자열", s), _ _ _ _ ("대상 문자열", "변경문자열", s)

2. **날짜 다루기**

- 문자열 → 날짜 : _ _ _ _ _ _ ("2018-10-07") / _ _ _ _ ("10/7/2018", format=" _ _ _ _ _ _ _ _ ")
- 날짜 → 문자열 : _ _ _ _ _ _ (Sys.Data(), foramt=" _ _ _ _ _ _ ")
  - 축약된 월 이름 : _ _ _ _ "Jan"
  - 전체 월이름 : _ _ _ _ "January"
  - 두자리 숫자 일 : _ _ _ _ "31"
  - 두자리 숫자 월 : _ _ _ _ "12"
  - 두자리 숫자 년 : _ _ _ _ "18"
  - 네자리 숫자 년 : _ _ _ _ "2018"