Chapter 1

완전 초보에요

지은이가 이 문서에서 독자를 "초보"라고 생각할 때는 아래에 나열된 사항들 중 두가지 이상에 해당하시는 분이라고 가정하였습니다.

- R이라는 프로그래밍 언어 이전에 다른 종류의 프로그래밍 언어를 경험한 적이 없으신 분,
- 유닉스와 같은 사용환경에 익숙하지 않은 분, 그리고
- 기초 통계 분석을 수행하는데 어려움을 많이 느끼시는 분.

1.1 사용전 알아야 할 숙지사항들

독자 스스로가 초보라고 생각하신다면 본 섹션의 내용을 꼭 읽어보시길 부탁드립니다. 그리고, 아래의 설명들에 대해서 "왜 그렇게 하는 것이지?" 라고 의구심을 가지기 보다는 "하늘을 하늘이라고 부른다"라는 식으로 받아들여주시길 부탁드립니다. 그 이유는 현재 이러한 내용들을 설명하고 이해하기에는 보다 많은 배경지식들이 필요하기 때문입니다. 따라서, 지금은 아래의 내용의 내용들에 대해서 "그냥 이런 전제조건을 알고 가야하는구나" 라고 생각해주시면 좋을 것 같습니다.

데이터를 읽고 조작하는 것에 대해서: R에서 입력되고 사용되며 조작이 되는 모든 데이터들은 열방향으로 나열된 후 이루어지게 됩니다. 여기에서 "열방향"이라는 말의 뜻을 이해하기 위해 아래의 예를 보세요. 수학적으로 1부터 12까지 12개의 정수로 이루어진 수열 (즉, $\{1, 2, \ldots, 11, 12\}$)이 있다고 가정합니다. 이것은 1차원 데이터라고 하고, 벡터라고 합니다. 이제 이 벡터를 아래와 같이 4행 3열로 이루어진 2차원상 공간의 수학적 개념으로 머릿속에 생각해봅니다. 어떤 독자는 이 수열을 행방향으로 나열해 보려고 하실 수 있고, 어떤 독자는 열방향으로 나열해 보려고 할 수 도 있습니다. R은 아래와 같이 열방향으로 이 벡터를 나열합니다.

```
1  5   9
2  6  10
3  7  11
4  8  12
```

따라서, 이 행렬에서 5 라는 행렬의 구성요소는 1행 2열에 위치하고 있다고 할 수 있으며, 벡터의 5번째의 값이라고도 할 수 있습니다. (만약 어떤 프로그램이 행으로 데이터를 나열하는 방식을 택하여 디자인되었다면 이 행렬의 5번째의 값은 6이 될 것입니다).

R에서 이렇게 데이터를 열방향으로 나열하는데에는 일반적인 통계적 요약이 변수별로 이루어지는 통계적 실무가 설계의 바탕이 되기 때문입니다. 그리고, 이러한 처리는 벡터를 기반으로 이루어지게 됩니다. 이 벡터를 얼마나 효과적으로 잘 다루는가는 R을 이용한 데이터 조작과 처리를 얼마나 R의 특징을 살려 프로그래밍할 수 있는가와 직결됩니다. 따라서, R을 시작하기 위해서는 벡터와 관련된 챕터를 제일 먼저 읽어주시길 부탁드립니다.

사용환경에 익숙해진다는 것에 대해서: 어떤 주어진 작업을 수행함에 있어서, 자신이 상상한 대로 원하는 작업결과를 만들어 내고자 하는 것은 도구를 이용하는 사용자로서 기본적인 욕구입니다. 그리고, 도구를 얼마나 잘 사용하는가에 따라서 욕구는 충족이 될 수 있고 그렇지 않을 수도 있습니다.

R은 유닉스와 같은 환경에서 영어사용자를 그 토대로 만들어졌기 때문에 많은 명령어들이 영어의 약자이거나, 유닉스 명령어들과 매우 유사하게 되어 있습니다. 따라서, 윈도우즈 사용자에게는 예상치 못한 작동이 발생할 수도 있으며 한글로 되지 않은 사용자 환경에 부담감을 줄 것입니다. 대표적인 예로 한국어 사용자환경, 한글데이터, 그리고 시간과 날짜 데이터는 로케일(locale)이라고 불리는 운영체제가 사용되는 지역설정과 관계가 있습니다. 이러한 것들과 관련된 문제들은 R과는 무관하며 무관한 운영체제의 작동방식을 따르는 것이므로, 이를 스스로 해결해 나가면서 R를 사용하는 것은 아마도 큰 시간적 낭비가 될 수도 있습니다. 그럼에도 불구하고 R을 써야 하는 독자가 있다면 **시간과 날짜 데이터 다루기** 챕터의 로케일에 대한 부분과 **환경설정과 유틸리티** 챕터에서 인코딩과 한글이라는 부분을 읽어보면 도움이 될 것이라는 것을 기억하시길 부탁드립니다.

알고리즘, 객체, 그리고 분석결과의 확인에 대해서: R은 사용자가 주어지는 업무를 시키는 대로만 수행하는 수많은 컴퓨터 응용프로그램들 중 하나입니다. 어떤 주어진 문제를 어떻게 풀어나가야 한다는 알고리즘의 정립은 사용자의 머릿속에서 일하는 일이므로, 사용자에게 어떤 주어진 분석이 있다면 사용자는 반드시 분석을 수행하는 프로세스에 대해서 잘 알고 있어야 합니다. 그리고, 이러한 프로세스를 자동화 하거나 혹은 결과의 재생산을 위하여 R은 분석 프로세스의 과정을 컴퓨터가 이해할 수 있도록 해주는 프로그래밍의 언어로서 쓰이는 것입니다. 이러한 알고리즘의 작성에 꼭 필요한 논리적인 설계는 R이 하는 것이 아니라 사용자의 머릿속에서 이루어지기 때문에 사용자에게 꼭 필요한 기능은 프로세스 각 단계의 중간결과를 확인해 보는 것입니다. 이러한 중간결과의 단계로부터 생성되어지는 모든 것들을 "객체"라고 합니다. (객체에 대한 이해는 추후에 프로그래밍의 관점에서 객체지향프로그래밍이라고 불리는 전산분야의 배경지식이 요구되지만, 여기에서는 단순히 R에서 생성되고 다루어지는 그 모든 것을 객체라고 합니다. 가장 단순한 예로 위에서 설명한 입력된 벡터와 행렬 모두 객체의 일종입니다. 그 이유는 입력되어 R이 인식한 어떠한 종류의 결과물이라고 말할 수있기 때문입니다). 올바른 중간단계의 결과를 바탕으로 다음 단계의 프로세스를 올바르게 진행할 수 있기 때문에 R은 분석되어 생성된 모든 결과를 한 번에 미리 다 보여주기 보다는 이렇게 중간에 작업된 결과를 확인해 볼 수 있도록 합니다.

데이터 분석을 위한 R의 사용에 대해서: 데이터 분석을 일반적으로 얘기하면 데이터가 가진 특징을 수학적 표현으로서 설명함으로서 동일한 상황속에서 결과를 재생산함을 의미하기도 합니다. 이는 아래와 같은 일반적인 수행절차를 밟게 됩니다.

1. 데이터 입출력과 클리닝, 그리고 전처리

2. 분석전 탐색적 시각화 작업

3. 통계모형의 결정 및 적용

4. 모형 적용 후의 보고서 생성 및 시각화 작업

5. 통계 모형 자체의 개발 또는 자동화 시스템 구축.

이 문서는 위에서 설명한 과정에 해당하는 순서대로 챕터들을 구성하고자 합니다.

안전하게 사용하는 방법에 대해서: R을 안전하게 사용한다는 의미는 사용을 하면서 접하게 되는 에러를 최소화한다는 의미입니다. 에러는 일반적으로 잘못된 문법적 사용과 논리적 오류로 인하여 발생되게 됩니다. 전자는 주로 R에서 주어진 방식대로 사용하지 않음을 의미하지만, 후자는 보통 알고리즘을 논리적 작성에 기인하게 됩니다. 논리적 오류는 해당 분야의 전문적인 지식과 경험을 통해 좌지우지되기 때문에 여기에서는 다루기 곤란한 합니다. 그러나, 최소한 문법적 오류를 방지하기 위해서는 아래와 같은 점을 미리 아시면 도움이 될 것이라고 생각합니다.

첫번째 예로, R은 대소문자를 구분하여 사용합니다. 이는 대문자 A라고 입력한 것은 소문자 a 라고 입력한 것과 서로 다른 것으로 인식하게 된다는 의미입니다.

두번째 예로, 어떤 함수를 사용할 때 입력인자의 순서에 민감합니다. 함수란 어떤 입력인자를 가지고 특정한 프로세스를 수행하여 결과를 얻는 블랙박스로 생각할 수 있습니다. 만약, `fn`이라는 함수가 존재하는데 이 함수는 `a`와 `b`라는 입력인자를 가지며 `fn(a,b)`로 사용되어져야 한다고 미리 정의가 되어있다고 가정합니다. 이때 `fn(b,a)` 또는 `fn(a)` 라고 함수를 사용하게 되면 R은 에러를 보여주거나 예상하지 못한 결과를 가져오게 됩니다. 이러한 실수를 최소화 하기 위해서 지시된 인자(named argument)를 함께 활용해야 합니다. 여기에서 지시된 인자란 함수 `fn`은 `a`와 `b`라는 입력인자를 사용하여 정의되었으니, 사용자가 실제적으로 `a`와 `b`에 이용하는 값 `val1`과 `val2` 을 `fn(a=val1, b=val2)`라는 방식으로 미리 정의된 인자명을 함께 사용하라는 의미입니다. 따라서, 어떤 함수를 사용하기 전에는 반드시 `args()`라는 함수를 사용하여 사용하고자 하는 함수의 인자명을 반드시 확인하는 것이 좋습니다.

세번째 예로, R은 애드온(add-on) 패키지 시스템이라는 것을 사용합니다. 이것은 R에서 기본적으로 제공하는 표준 배포판 외의 기능을 사용하고자 한다면 추가적인 패키지를 붙여 쓴다는 것을 의미합니다. 따라서, 사용자가 필요한 함수를 인터넷 검색을 통해 알게 되었을지라도 만약 그 함수가 표준배포판에 포함되어 있지 않다면 해당 함수를 가지고 있는 패키지를 설치전까지는 쓸 수 없다는 의미입니다. 따라서, 이를 모르는 상태에서 초보자가 가장 많이 겪는 실수는 어떤 함수를 사용하고자 할 때 "xxx 함수가 없습니다" 또는 "xxx 함수를 찾을 수 없습니다" 라는 것일 것입니다.

그러나, 꼭 아셔야 할 점은 패키지 설치가 아닌 사용자가 수행하고자 하는 분석에 적합한 패키지를 선택하는 방법입니다. 이는 해당분야의 지식이 요구되며, 제공되는 패키지는 주로 개발자 자신만의 연구에 특화되어 있다는 점입니다. 따라서, 이 문서는 패키지의 설치와 관리라는 부분을 통계 모형의 선택과 적용 그리고 장단점이라는 부분에 위치시켰습니다. 또한, 대다수의 패키지는 R의 표준 배포판을 이용하여 작성되는 것이므로, 본 문서는 가장 처음에 언급한 바와 같이 다양한 종류의 패키지를 어떻게 사용하는가 보다는 표준배포판만을 이용하여 관련문제를 해결하는 알고리즘 위주의 설명을 전개하도록 노력할 것입니다.

1.2 왜 R을 사용하나요?

R을 사용하는 이유는 아마도 아래와 같은 이유이기 때문입니다. (R Documentation에는 없는 이 문서의 지은이 개인의 생각임을 명심하시길 바랍니다)

1. R은 매우 다양한 분야에서 개발되고 적용되는 최신 통계기법을 적용할 수 있는 자유소프트웨어이기 때문입니다.
2. 행렬기반의 객체지향적 프로그래밍 언어이기 때문입니다.
3. 다른 소프트웨어들에 비교하여 문법적 사용의 자유롭기 때문일 것입니다.

1.3 통계소프트웨어의 종류

R이라는 통계소프트웨어를 대체할 수 있는 다른 소프트웨어들은 다음과 같습니다.

- S-PLUS (상업용 버전의 S 언어 소프트웨어)
- MATLAB (R과 같은 행렬기반의 언어)
- SPSS
- Octave (MATLAB의 GNU 버전)
- Python (프로그래밍 언어)
- SAS
- STATA