R 치트시트

1. 데이터 구조

- 벡터 (Vector): 동일 타입의 1차원 배열. `c()`, `1:5`, `seq()`, `rep()`.
- 매트릭스 (Matrix): 동일 타입의 2차원 배열. `matrix(data, nrow, ncol)`.
- 배열 (Array): 2차원 이상의 배열. `array(data, dim)`.
- 리스트 (List): 서로 다른 타입의 객체를 포함. `list(name="A", value=1)`.
- 데이터 프레임 (Data Frame): 테이블 형태. 각 열은 벡터이며 서로 다른 타입을 가질 수 있음. `data.frame(...)`.
- 팩터 (Factor): 범주형 데이터를 저장. `factor(vector, levels, labels)`. 통계 모델링에 중요.
- tibble: `tidyverse`의 현대적인 데이터 프레임. 더 나은 출력 형식과 엄격한 규칙을 가짐. `tibble(...)`.

2. 데이터 접근 (인덱싱)

- 위치 기반:
  - `x[5]`: 5번째 요소.
  - `df[1, 2]`: 1행 2열.
  - `df[1, ]`: 1행 전체.
- 이름 기반:
  - `x["name"]`
  - `df[, "col_name"]`
  - `df$col_name`
- 논리 벡터 기반:
  - `x[x > 3]`
  - `subset(df, subset = condition)`
- 리스트 접근:
  - `my_list[[1]]` 또는 `my_list[["name"]]`: 요소 자체를 반환.
  - `my_list[1]`: 요소가 포함된 리스트를 반환.

3. 제어 구조 및 함수

- if-else: `if (condition) { ... } else { ... }`. `ifelse(test, yes, no)` 벡터화된 버전.
- for, while, repeat: 루프. R에서는 벡터화 연산이 더 효율적이므로 가능한 한 루프를 피하는 것이 좋음.
- apply 계열 함수:
  - `apply(X, MARGIN, FUN)`: MARGIN (1: 행, 2: 열)에 함수 적용.
  - `lapply(X, FUN)`: 리스트/벡터 각 요소에 함수 적용, 결과를 리스트로 반환.
  - `sapply(X, FUN)`: `lapply`와 유사하지만 결과를 벡터/매트릭스로 단순화 시도.
  - `tapply(X, INDEX, FUN)`: 그룹(INDEX 팩터)별로 함수 적용.
  - `mapply(FUN, ...)`: 여러 리스트/벡터에 함수를 요소별로 적용.
- 함수 정의:

```
my_function ← function(arg1, arg2 = default) {
  # 함수 본문
  return(result) # return()은 명시적으로 사용하거나 마지막 표현식의 값이 반환됨
}
```

4. 패키지 관리

- `install.packages("package_name")`: 패키지 설치.
- `library(package_name)`: 패키지를 현재 세션에 로드.
- `require(package_name)`: `library`와 유사하나, 실패 시 `FALSE`를 반환하여 스크립트 내에서 조건부 로드에 사용 가능.
- `update.packages()`: 설치된 모든 패키지 업데이트.
- `remove.packages("package_name")`: 패키지 제거.
- `help(package = "package_name")`: 패키지 도움말 보기.

5. tidyverse 생태계

`tidyverse`는 데이터 과학을 위한 R 패키지 모음입니다. 일관된 철학, 문법, 데이터 구조를 공유합니다.

dplyr: 데이터 조작의 문법

- `%>%` (파이프 연산자): 왼쪽의 결과를 오른쪽 함수의 첫 번째 인수로 전달. 코드 가독성 향상.
- 핵심 동사:
  - `filter()`: 조건에 맞는 행 필터링.
  - `select()`: 특정 열 선택/제외. (도우미 함수: `starts_with()`, `contains()`)
  - `mutate()`: 새 열 추가 또는 기존 열 수정.
  - `arrange()`: 행 정렬. (`desc()`로 내림차순).
  - `summarise()` 또는 `summarize()`: 데이터를 요약. (예: `mean()`, `sd()`, `n()`).
  - `group_by()`: 데이터를 그룹별로 묶어 이후의 작업을 그룹 단위로 수행.
  - `count()`: 그룹별로 개수를 셈.
- Join (결합):
  - `inner_join()`, `left_join()`, `right_join()`, `full_join()`

ggplot2: 데이터 시각화의 문법

그래프를 구성 요소(데이터, 심미성 매핑, 기하학적 객체 등)의 레이어로 만듦.

- 기본 구조:

```
ggplot(data = <DATA>, mapping = aes(<MAPPINGS>)) +
  <GEOM_FUNCTION>()
```

- `aes(x=, y=, color=, shape=, size=, fill=)`: 데이터 변수를 시각적 속성에 매핑.
- Geoms:
  - `geom_point()`: 산점도
  - `geom_line()`: 선 그래프
  - `geom_bar(stat="identity")`: 막대 그래프 (y 값 직접 지정)
  - `geom_histogram()`: 히스토그램
  - `geom_boxplot()`: 박스 플롯
  - `geom_smooth(method="lm")`: 회귀선 추가
- Facets: 하위 그룹별로 그래프를 나눠서 그림.
  - `facet_wrap(~ variable)`
  - `facet_grid(rows ~ cols)`
- 기타: `labs()`, `scale_*()`, `theme()` 등으로 그래프의 제목, 축, 색상, 전체적인 모양을 조정.

tidyr: 데이터 정돈 (Tidy Data)

- `pivot_longer()`: 넓은 형식(wide format)의 데이터를 긴 형식(long format)으로 변환.
- `pivot_wider()`: 긴 형식의 데이터를 넓은 형식으로 변환.
- `separate()`: 한 열을 여러 열로 분리.
- `unite()`: 여러 열을 한 열로 결합.

readr: 데이터 읽기/쓰기

- `read_csv()`, `read_tsv()`: 빠르고 일관된 방식으로 CSV/TSV 파일 읽기.
- `write_csv()`, `write_tsv()`: 데이터 프레임을 파일로 저장.

6. R Markdown

코드, 결과(텍스트, 그래프), 서술을 하나의 문서로 통합.

- 문서 구조: YAML 헤더, R 코드 청크(```{r} ... ```), 마크다운 텍스트.
- 출력 형식: HTML, PDF, Word 등 다양한 형식으로 렌더링(Knit) 가능.
- 코드 청크 옵션: `echo=FALSE` (코드 숨김), `eval=FALSE` (코드 실행 안 함), `results='hide'` (결과 숨김), `fig.width`, `fig.height` (그림 크기).