

Lecture 08 - 리팩토링 #1

개요

- 일자별 온도 데이터셋
- 첫번째 시도 - 그냥 만들기
- 개선 #1 - 함수로 나누기
- 개선 #2 - 다른 방법으로 모듈화
- 연습: Fibonacci sequence

일자별 온도 데이터셋 (10m)

세 개의 데이터셋

date	temp	region	date	lat	lon	temperature	month and date	fahrenheit
20150301	9	Seoul	20150301	37.132	126.031	9	March 1	55
20150302	8	Jeju	20150302	37.103	125.954	8	March 2	53
20150303	9	Seoul	20150303	37.132	126.031	9	March 3	56
20150304	10	Jeju	20150304	37.103	125.954	10	March 4	57
20150305	11	Daegu	20150305	37.132	126.031	11	March 5	57
20150306	12	Seoul	20150306	37.103	125.954	12	March 6	53
20150307	13	Busan	20150307	37.132	126.031	13	March 7	59
20150308	10	Seoul	20150308	37.103	125.954	10	March 8	60
20150309	9	Daegu	20150309	37.132	126.031	9	March 9	61

<http://s.g15e.com/temp01.csv>

<http://s.g15e.com/temp02.csv>

<http://s.g15e.com/temp03.csv>

목표: 일자와 온도(섭씨)만 뽑아서 하나의 data frame 만들기

첫번째 시도 - 그냥 만들기 (20m)

```
# Read CSV
temp01 <- read.csv('http://s.g15e.com/temp01.csv')
temp02 <- read.csv('http://s.g15e.com/temp02.csv')
temp03 <- read.csv('http://s.g15e.com/temp03.csv')
```

```

# Convert date column
temp01 <- transform(temp01, date=as.integer(substr(date, 7, 9)))
temp02 <- transform(temp02, dt=as.integer(substr(dt, 7, 9)))
temp03 <- transform(temp03, month.and.date=as.integer(substr(month.and.date, 5, 7)))

# Convert fahrenheit to celcius
temp03 <- transform(temp03, fahrenheit=(fahrenheit-32) / 1.8)

# Select relevant columns
temp01 <- subset(temp01, select=c("date", "temp"))
temp02 <- subset(temp02, select=c("dt", "temperature"))
temp03 <- subset(temp03, select=c("month.and.date", "fahrenheit"))

# Change column names
colnames(temp02) <- c("date", "temp")
colnames(temp03) <- c("date", "temp")

# Bind
df <- rbind(temp01, temp02, temp03)

```

개선 #1 - 함수로 나누기 (20m)

문제점들:

- 전체 변수와 전체 코드가 잠재적으로 상호 의존: “이 변수는 어디에서 어떻게 쓰이지?”
- 실행 흐름에 따라 변수의 내용이 변경됨: “이 변수에 지금 무슨 값이 담겨 있더라?”

개선책:

- 관련된 코드와 변수를 함수로 묶어주기
- 되도록 변수를 변경하지 않기

첫번째 파일 처리하기

```

cleanse01 <- function() {
  temp01 <- read.csv('http://s.g15e.com/temp01.csv')
  temp01 <- transform(temp01, date=as.integer(substr(date, 7, 9)))
  temp01 <- subset(temp01, select=c("date", "temp"))
  return(temp01)
}

```

두번째 파일 처리하기

```

cleanse02 <- function() {
  temp02 <- read.csv('http://s.g15e.com/temp02.csv')

```

```

temp02 <- transform(temp02, dt=as.integer(substr(dt, 7, 9)))
temp02 <- subset(temp02, select=c("dt", "temperature"))
colnames(temp02) <- c("date", "temp")
return(temp02)
}

```

세번째 파일 처리하기

```

cleanse03 <- function() {
  temp03 <- read.csv('http://s.g15e.com/temp03.csv')
  temp03 <- transform(temp03, month.and.date=as.integer(substr(month.and.date, 5, 7)))
  temp03 <- transform(temp03, fahrenheit=(fahrenheit-32) / 1.8)
  temp03 <- subset(temp03, select=c("month.and.date", "fahrenheit"))
  colnames(temp03) <- c("date", "temp")
  return(temp03)
}

```

합치기:

```
df <- rbind(cleanse01(), cleanse02(), cleanse03())
```

변수 이름을 더이상 01, 02, 03 등으로 구분할 필요 없음

```

cleanse01 <- function() {
  temp <- read.csv('http://s.g15e.com/temp01.csv')
  temp <- transform(temp, date=as.integer(substr(date, 7, 9)))
  temp <- subset(temp, select=c("date", "temp"))
  return(temp)
}

```

변수가 함수 안에 갇혀 있으므로(local variable), 같은 변수를 덮어쓰지 말고 새 변수를 마음 편하게 남발하기
+ 좀 더 설명적인 이름으로 바꾸기:

```

cleanse01 <- function() {
  raw <- read.csv('http://s.g15e.com/temp01.csv')
  converted <- transform(raw, date=as.integer(substr(date, 7, 9)))
  reduced <- subset(converted, select=c("date", "temp"))
  return(reduced)
}

```

현재 상황:

```

cleanse01 <- function() {
  raw <- read.csv('http://s.g15e.com/temp01.csv')
  converted <- transform(raw, date=as.integer(substr(date, 7, 9)))
  reduced <- subset(converted, select=c("date", "temp"))
  return(reduced)
}

```

```

cleanse02 <- function() {
  raw <- read.csv('http://s.g15e.com/temp02.csv')
  converted <- transform(raw, dt=as.integer(substr(dt, 7, 9)))
  reduced <- subset(converted, select=c("dt", "temperature"))
  colnames(reduced) <- c("date", "temp")
  return(reduced)
}

cleanse03 <- function() {
  raw <- read.csv('http://s.g15e.com/temp03.csv')
  converted <- transform(raw, month.and.date=as.integer(substr(month.and.date, 5, 7)))
  converted2 <- transform(converted, fahrenheit=(fahrenheit-32) / 1.8)
  reduced <- subset(converted2, select=c("month.and.date", "fahrenheit"))
  colnames(reduced) <- c("date", "temp")
  return(reduced)
}

df <- rbind(cleanse01(), cleanse02(), cleanse03())

```

개선 #2 - 다른 방식으로 모듈화 (30m)

지향점: 중복을 제거하고 의도를 드러내기

중복을 들어내기(remove) 전에 중복을 드러내기(reveal).

read.csv() 라인을 완전히 동일한 형태로:

```

cleanse01 <- function() {
  src <- 'http://s.g15e.com/temp01.csv'
  raw <- read.csv(src)
  converted <- transform(raw, date=as.integer(substr(date, 7, 9)))
  reduced <- subset(converted, select=c("date", "temp"))
  return(reduced)
}

```

칼럼명 변환을 추가하여 더 유사해지게:

```

cleanse01 <- function() {
  src <- 'http://s.g15e.com/temp01.csv'
  raw <- read.csv(src)
  converted <- transform(raw, date=as.integer(substr(date, 7, 9)))
  reduced <- subset(converted, select=c("date", "temp"))
  colnames(reduced) <- c("date", "temp")
  return(reduced)
}

```

reduce와 rename을 먼저하면 그 이후 단계의 코드를 통일시킬 수 있음

```
cleanse01 <- function() {  
  src <- 'http://s.g15e.com/temp01.csv'  
  raw <- read.csv(src)  
  reduced <- subset(raw, select=c("date", "temp"))  
  colnames(reduced) <- c("date", "temp")  
  converted <- transform(reduced, date=as.integer(substr(date, 7, 9)))  
  return(converted)  
}
```

reduce에서 선택할 칼럼명을 변수로 빼내기:

```
cleanse01 <- function() {  
  src <- 'http://s.g15e.com/temp01.csv'  
  columns <- c("date", "temp")  
  
  raw <- read.csv(src)  
  reduced <- subset(raw, select=columns)  
  colnames(reduced) <- c("date", "temp")  
  converted <- transform(reduced, date=as.integer(substr(date, 7, 9)))  
  return(converted)  
}
```

각 변환을 함수로 빼내기.

```
sameform <- function(df, columns) {  
  reduced <- subset(df, select=columns)  
  colnames(reduced) <- c("date", "temp")  
  return(reduced)  
}  
  
cleanse01 <- function() {  
  src <- 'http://s.g15e.com/temp01.csv'  
  columns <- c("date", "temp")  
  
  raw <- read.csv(src)  
  reduced <- sameform(raw, columns)  
  converted <- transform(reduced, date=as.integer(substr(date, 7, 9)))  
  return(converted)  
}
```

변환 목록을 변수로 빼내기.

(코드 생략)

변수를 파라미터로 바꾸기.

(코드 생략)

Baby-steps 연습하기: Fibonacci sequence 유도해내기 (30m)

(코드 생략)