Chapter14 Behavior Sequences

Status 비어 있음

Assign 비어 있음

댓글 추가

▼ Check-in (5′)

Q. Ch1-Ch14 중 다시 자세히 보고 싶은 Chapter는? 그 이유는?

- 이승희 : Ch12+모두 다
- 이진재 : Ch12 Association Rule. 업무에서 활용도가 높을거 같아서
- Eunice Seo : 전체 다 + ch12
- 손경희 : Ch12 Association Rule. 생소했던 내용이라 다시 보고 싶네용
- 강동오 : Ch12를 다시 보고 싶습니다.
- 정시앙 : 오늘 챕터, 업무 활용에 필요한 거 위주로 전반적으로 다시 보고 싶네요
- 박규서 : 전 9장 이후 ~
- 채충일 : Chapter 12,14 오늘 챕터. 마코브 체인, 트렌지션 메트릭스에 대한 적용에 대한 아이디어 그리고 함께 쓸 방법들에 대한 생각들이 많이 생겼습니다.
- 윤승원 : data simulation; because there are lot of times when real data are not avaialable

Chapter 14 Behavior Sequence (25′)

▼ Key Points

- 마케터로서 고객이 하는 일련의 행동에 대한 궁금증이 있음, 본 과정에서는 public web server log를 가지고 분석해 봄 : In this chapter, we examined sequential patterns of movement across discrete states. This kind of analysis may be applied to web site behavior, yet it is also applicable to many kinds of sequence data, such as historical purchasing patterns, life events, biomedical data, and others.
- 전처리 작업:
  - Actual Web log를 분석하는 기능에 대한 검토 : This chapter demonstrates the most complex data cleaning process in this text.
    Real-world data needs careful, stepwise processing with careful consideration and inspection along the way. Throughout the chapter, we limited analyses to relevant data such as "GET" events and HTML page requests (Sects. 14.1.5 and 14.4.2).
  - 시간,날짜 데이터 형식 맞추기: When text data represents dates or times, it requires careful translation to a nativeR
    data format. The relatively simple but rigid strptime() command (Sect. 14.1.4) can do this for user-defined patterns. Other packages provide more comprehensive and flexible options (see Sect. 14.7).
  - Regular expressions (regexes) provide very powerful processing for text data with identifiable structure. We decoded a regex that parses HTTP requests and identifies the specific page names (Sect. 14.1.5) and used another one for text substitutions (Sect. 14.4.2).
  - Session 길이에 대한 정의 : Log data often needs to be divided into sessions, where events represent actions taken by one user in a specific block of time. It can be difficult and somewhat arbitrary to determine the boundaries of sessions—how much inactivity is long
    enough? The rle() function is especially useful to find repeating values (e.g., recurrent identifiers) that help to identify sessions. We found session boundaries with a combination of users' identifiers along with time gaps between actions (Sect. 14.3.1).
- 마르코프 체인 :
  - Markov chains are a good initial choice for a method to analyze transitions among different behavior states (Sect.14.4). A Markov chain proposes a transition matrix that defines the odds of moving from one state to another. We used the click steam package to estimate a transition matrix for the Top 20 pages in the EPA data. Because a web site may have hundreds, thousands, or millions of pages, one may need to reduce the state space before modeling (Sect. 14.4.2).
  - One decision is whether to model end states, such as leaving the site or purchasing an item. These are often of high interest because they reveal where a site may be working poorly (or well) relative to our goals (Sect. 14.4.2). However, end states may also be dominant and lead to uninteresting predictive modeling (Sect. 14.4.5).
  - Heatmaps are useful to visualize transition matrices. They are even more useful when the rows and columns are clustered to reveal groupings of items whose

transition patterns are similar. The clustering options in superheat can reveal common patterns of page transitions (Sect. 14.4.4).

- **Higher order Markov chains may be used to predict states from a series of multiple previous states** (Sect. 14.4.5). To fit them, you may need to filter the data to a subset
  with sequences of sufficient length.
- In the click stream package, predict (object,startPattern)may
  be used to predict the next state, or multiple successive states, for a given pattern (Sect. 14.4.5).

▼ 14.1 WebLog Data

▼ 14.1.1 EPA (Environmental Protection Agency) Web Data : 데이터 다운로드 후 검토

```
> summary(epa.df)
      host          timestamp            request              status
 host1986:  294   Length:47748       Length:47748       http200:36712  ...
 host1032:  292   Class :character   Class :character   http304: 5300  ...
```

```
     bytes            rawhost             datetime
 Min.   :      0   Length:47748       Min.   :1995-08-29 23:53:25  ...
 1st Qu.:    231   Class :character   1st Qu.:1995-08-30 10:58:37  ...
 reqtype        pagetype          page
 GET :46014   gif  :22418   Length:47748       ...
 HEAD:  106   html : 8687   Class :character   ...
```

▼ 14.1.2 Processing the Raw Data : Column에 이름 부여

- names ( ) : column에 이름 부여
- str ( ) : 데이터셋 구조 확인

```
> epa.df.raw <- read.table("https://goo.gl/LPqmGb", sep=" ",
+                          header=FALSE, stringsAsFactors = FALSE)
> # name the columns
> names(epa.df.raw) <- c("host", "timestamp", "request", "status", "bytes")
> str(epa.df.raw)
'data.frame':   47748 obs. of  5 variables:
 $ host     : chr  "141.243.1.172" "query2.lycos.cs.cmu.edu" ...
 $ timestamp: chr  "[29:23:53:25]" "[29:23:53:36]" ...
 $ request  : chr  "GET /Software.html HTTP/1.0" ...
 $ status   : int  200 200 200 200 200 200 200 200 200 200 ...
 $ bytes    : chr  "1497" "1325" "1014" "4889" ...
```

▼ 14.1.3 Cleansing the Data

빈 토글입니다. 클릭하거나 내부로 블록을 드롭하세요.

▼ 14.1.4 Handling Dates and Times

```
> epa.df.raw <- read.table("https://goo.gl/LPqmGb", sep=" ",
+                          header=FALSE, stringsAsFactors = FALSE)
> # name the columns
> names(epa.df.raw) <- c("host", "timestamp", "request", "status", "bytes")
> str(epa.df.raw)
'data.frame':   47748 obs. of  5 variables:
 $ host     : chr  "141.243.1.172" "query2.lycos.cs.cmu.edu" ...
 $ timestamp: chr  "[29:23:53:25]" "[29:23:53:36]" ...
 $ request  : chr  "GET /Software.html HTTP/1.0" ...
 $ status   : int  200 200 200 200 200 200 200 200 200 200 ...
 $ bytes    : chr  "1497" "1325" "1014" "4889" ...
```

[29:23:53:25] ⇒ 1995-08-29 23:53:25 EDT 로 표기 형식 전환 위해 다음을 활용;

- strptime(Data, format, time zone)
- substr(data,start,stop)
- as. POSIXct : 문자 변수를 날짜 타입 변수로 변환

```
> epa.df$datetime <- strptime(paste0("1995:08:",
+                                    substr(epa.df.raw$timestamp, 2, 12)),
+                             "%Y:%m:%d:%H:%M:%S", tz="America/New_York")
```

```
> epa.df$datetime <- as.POSIXct(epa.df$datetime)
```

▼ 14.1.5 Request and Page Types : 어떤 리퀘스트가 많은지, 리퀘스트를 통해 가져오는 내용이 무엇인지 파악

- 사용자 브라우저 리퀘스트 상의 구조 확인:

```
> head(epa.df$request)
[1] "GET /Software.html HTTP/1.0"
[2] "GET /Consumer.html HTTP/1.0" ...
```

- Get + 파일 or 페이지 + Communication Protocol (Http)

- sub(pattern, replacement) : 필요 없는 Http를 공란으로 교체

```
> epa.df$request <- sub(" HTTP/1.0", "", epa.df.raw$request)
> head(epa.df$request)
[1] "GET /Software.html"                "GET /Consumer.html"   ...
```

- Browser action 타입 (e.g. Get :서버에서 컨텐트 가져오기, Post :데이터 전송, HEAD: 페이지 로딩 없이 정보 가져오기) 결정 ⇒ 예상대로 Get request가 가장 많음을 알 수 있음

```
> epa.df$reqtype <- NA
> epa.df$reqtype[grepl("POST ", epa.df.raw$request)] <- "POST"
> epa.df$reqtype[grepl("GET ", epa.df.raw$request)]  <- "GET"
> epa.df$reqtype[grepl("HEAD ", epa.df.raw$request)] <- "HEAD"
> epa.df$reqtype <- factor(epa.df$reqtype)
> table(epa.df$reqtype)
  GET  HEAD  POST
46014   106  1622
```

  - reqtype( ) : to code the request type for each line
  - grepl() : to search for the presence of a particular text string

- 사용자가 요청한 컨텐트 타입 (e.g. HTML, GIF, JPG, PDF 파일 등) 점검:

```
> epa.df$pagetype <- "other"
> epa.df$pagetype[grepl("\\.gif",  epa.df.raw$request,
+                       ignore.case=TRUE)] <- "gif"
> epa.df$pagetype[grepl("\\.html", epa.df.raw$request,
+                       ignore.case=TRUE)] <- "html"
> epa.df$pagetype[grepl("\\.pdf",  epa.df.raw$request,
+                       ignore.case=TRUE)] <- "pdf"
> epa.df$pagetype <- factor(epa.df$pagetype)
>
> table(epa.df$pagetype)
  gif  html other   pdf
22418  8687 16536   107
```

  - grepl(pattern, x, ignore.case=T) : to search for the presence of a particular text string, x라는 문자열에서 해당 패턴이 존재하는지 알아봄

14.2 Basic Event Statistics : 이벤트 발생 빈도수는?, 무슨 일이 일어났는가?, 어떤 사용자가 가장 엑티브한가

14.2.1 Events

- 전체 이벤트 중 발생빈도 Top10 보기:

```
> head(sort(table(epa.df$page), decreasing = TRUE), 10)
/icons/circle_logo_small.gif  3203
/                             2381
/logos/small_gopher.gif       1851
/logos/us-flag.gif            1817  ...
```

- 사용자의 관심사 파악위해 HTML request 필터링해서 page interaction 이해 :

```
> head(sort(table(epa.df$page[epa.df$pagetype=="html"]),
+           decreasing = TRUE), 10)

        /Rules.html             /Software.html        /docs/WhatsNew.html
                312                       169                        159
         /Info.html              /Offices.html        /docs/Internet.html   ...
                151                       139                        137   ...
```

14.2.2 Events by Time : 사용자는 언제 가장 엑티브 한가?

- geom_density() : density curve그려서 파악

```
> library(scales)         # install if needed
> p <- ggplot(epa.df, aes(x=datetime, fill=I("lightblue"))) + # color
+   geom_density(alpha=0.5, bw = "SJ-dpi", adjust=2.0) +       # granular
+   scale_x_datetime(breaks = date_breaks("2 hours"),
+                    date_labels = "%b %d %H:%M") +            # axis labels
+   theme(axis.text.x =                                       # rotate
+           element_text(angle = 45, vjust = 1, hjust = 1)) +
+   ylab("HTTP Requests (proportion)") +                      # label axes
+   xlab("Date / Time")
> p
```

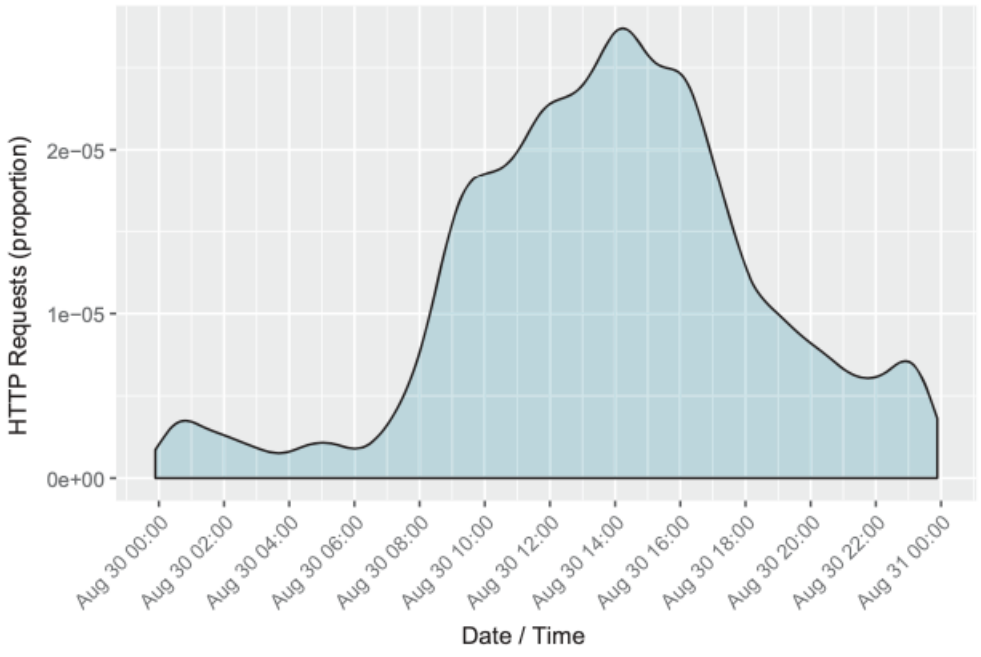

Fig. 14.1 Web requests by time in the EPA data

⇒ Not surprisingly, we see that the server is busiest during daytime hours, and there is an additional peak around midnight. We might use such data to inform decisions about when to staff a call center, update files, or set shipping times. Also, it suggests patterns that might be explored in qualitative research such as interviews or observation.

▼ 14.2.3 Errors

빈 토글입니다. 클릭하거나 내부로 블록을 드롭하세요.

▼ 14.2.4 Active Users

- IP address를 통해서 사용자 인지 (정확도 이슈는 있음 : e.g. 한 명이 다수의 네트워크 사용, 하나의 네트워크를 다수의 사용자가 공유 등)

```
> length(unique(epa.df$host))
[1] 2333
```

- plot() 통해서 Active 사용자 파악 : table()로 active user count⇒ plot()으로 배치⇒ top10user 인지

```
> host.tab <- sort(table(epa.df$host), decreasing = TRUE)
> plot(host.tab)
> head(host.tab, 10)
host1986 host1032 host2247 host1439 host1028 host1656 host1511  host924 ...
     294      292      266      263      248      176      174      172 ...
```

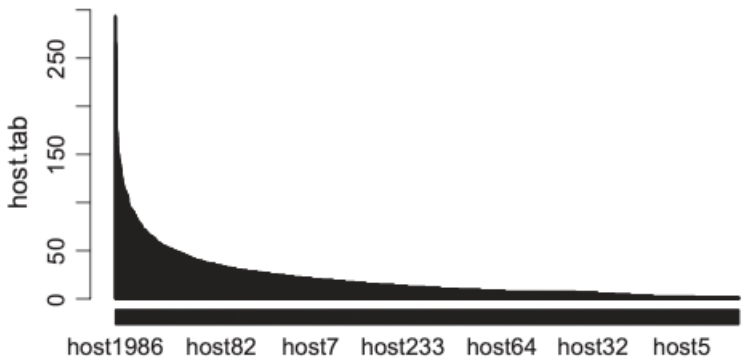

Fig. 14.2 Total page requests by user. A small number of users request many pages

⇒ The resulting plot is shown in Fig.14.2. As is typical in such data, the pattern is roughly anti-logarithmic; a small proportion of users account for most transactions. (Try plotting the log() of the table to see a more linear relationship.)

14.3 Identifying Sequences (Sessions)

- Sequence를 구성하는 시간 기준 결정 필요:
- 예) 놀이동산: 하루, 은행: 연간, 개인 건강:생애주기

14.3.1 Extracting Sessions

- Order () : to sort data

14.3.2 Session Statistics

- cumsum(): cumulative(running) totals
- rule() : to count repetitions of values

14.4 Markov Chains (MC) for Behavior Transitions

: 고객 행동의 시퀀스를 모델링하는 방법 중 하나

14.4.1 Key Concept and Demonstration

- MC의 구성 요소 :
  - 1) a set of states (현 상태)
    - 예) state : unmarried / married / divorced / widowed
  - 2) a matrix of transition probabilities (각 상태로 전환되는 가능성)
    - 예) unmarried⇒married : p=0.2

      married⇒divorced in one yr: p=0.02

      unmarried⇒divorced : p=0
- State :

```
> p.start <- c(0.7, 0.2, 0.1)
```

  - page1에 70%의 사용자, page2에 20%의 사용자, page3에 10%의 사용자가 위치
- Transition probabilities:

```
> p.trans <- matrix(c(0.1, 0.5, 0.2,
+                     0.6, 0.4, 0.8,
+                     0.3, 0.1, 0.0), nrow=3, byrow=TRUE)
```

  - matrix position[1,1] p=0.1 : page1에 있던 사용자가 클릭 후 page1 에 그대로 있는 경우
  - matrix position[2,1] p=0.6 : page1에 있던 사용자가 클릭 후 page2에 있는 경우
  - matrix position [3,1] p=0.3 : page1에 있던 사용자가 클릭 후 page3에 있는 경우
- 한 번 클릭 후 사용자의 위치 확률:

```
> p.trans %*% p.start                  # one click from start
      [,1]
[1,] 0.19
[2,] 0.58
[3,] 0.23
>
```

  - 한 번 클릭 후 page1에 있는 사용자 비율= 0.19 = 0.7*0.1+0.2*0.5+0.1*0.2
  - 한 번 클릭 후 page2에 있는 사용자 비율= 0.58 = 0.7*0.6+0.2*0.4+0.1*0.8
  - 한 번 클릭 후 page3에 있는 사용자 비율= 0.23 = 0.7*0.3+0.2*0.1+0.1*0.0
- 두 번 클릭 후 사용자의 위치 확률:

```
> p.trans %*% p.trans %*% p.start          # two clicks from start
       [,1]
[1,] 0.355
[2,] 0.530
[3,] 0.115
```

  - 두 번 클릭 후 page1에 있는 사용자 비율= 0.355 = 0.19*0.1+0.58*0.5+0.23*0.2
  - 두 번 클릭 후 page2에 있는 사용자 비율= 0.530 = 0.19*0.6+0.58*0.4+0.23*0.8
  - 두 번 클릭 후 page3에 있는 사용자 비율= 0.115 = 0.19*0.3+0.58*0.2+0.23*0.0
- n 번 클릭 후에 각 페이지에 있을 사용자 비율은 n승 계산하는 것과 동일한 과정
- 100번 클릭의 경우

```
> library(expm)                       # matrix exponentiate, install if needed
> p.trans %^% 100                     # 100 steps
      [,1]  [,2]  [,3]
[1,] 0.325 0.325 0.325
[2,] 0.525 0.525 0.525
[3,] 0.150 0.150 0.150
```

- 시작 페이지의 위치와 상관없이, 32.5% 의 사용자는 page1, 52.5%의 사용자는 page2, 15%의 사용자는 page3에 위치

▼ 14.4.2 Formatting the EPA Data for clickstream Analysis : Cleansing 작업

- EPA data 에서 transition probabilities 찾기 위해서 clickstream package 활용

```
> library(clickstream)        # install first, if needed
```

- EPA data가 가진 6565 unique pages를 분량상 20 most common pages로 추려냄

```
> # .. for simplicity here, restrict to top 20 most frequent pages
> top.pages <- names(head(sort(table(epa.df$page[epa.df$pagetype=="html"]),
+                              decreasing = TRUE), 20))
> epa.html    <- subset(epa.ordered, pagetype=="html" & page %in% top.pages)
```

- Session 별로 data 분리하고, length1 이상인 세션은 제거

```
> # split the sessions
> epa.session <- split(epa.html, epa.html$session)
> # .. optional, remove any of length 1
> epa.stream.len <- lapply(epa.session, nrow)
> epa.session <- epa.session[epa.stream.len > 1]
```

- 세션이 정확하게 분리되었는지 확인: 521개의 세션이 있음을 확인

```
> str(head(epa.session))
List of 6
 $ 13:'data.frame': 3 obs. of  13 variables:
  ..$ host      : Factor w/ 2333 levels "host1","host10",..: 10 10 10
  ..$ timestamp : chr [1:3] "[30:15:55:42]" "[30:16:10:08]" "[30:16:10:16]"
  ..$ request   : chr [1:3] "GET /docs/WhatsNew.html" "GET /Offices.html" "GET
    /Offices.html"
...
> length(epa.session)
[1] 521
```

- end state을 어떻게 어떻게 인지 할 것인지 결정 필요:

  Now we are able to compile the events onto individual lines for each session. To do this, we iterate over the sessions, and for each one we aggregate a line comprising the user identifier (host), the sequence of events (multiple page entries), and a final end state: 여기선 각 사용자의 sequence 마다 END 문구를 넣음

```
> head(epa.stream)
13   "host1006,/docs/WhatsNew.html,/Offices.html,/Offices.html,END"
17   "host101,/Info.html,/Research.html,END" ...
```

  위 결과를 만들기 위해서 아래의 과정을 거침;

```
> epa.stream <- unlist(lapply(epa.session,
+                             function(x)
+                               paste0(unique(x$host), ",",
+                                      paste0(unlist(x$page), collapse=","),
+                                      ",END")))
```

- /⇒ii로, .⇒ 로 replace 작업 : clickstream 이 "/" , "," 등의 character를 제거해버리기 때문에 미리 다른 character로 교체 작업 진행

```
> head(epa.stream)
13   "host1006,/docs/WhatsNew.html,/Offices.html,/Offices.html,END"
17   "host101,/Info.html,/Research.html,END" ...
```

- 임시 파일을 만들어서 clickstream object로 데이터 가져오기:

```
13  "host1006,iidocsiiWhatsNew,iiOffices,iiOffices,END"
```

```
> click.tempfile <- tempfile()
> writeLines(epa.stream, click.tempfile)
> epa.trans <- readClickstreams(click.tempfile, header = TRUE)
```

- 시작과 끝 부분을 확인해서 이슈 사항 유무 점검:

```
> head(epa.stream)
13  "host1006,iidocsiiWhatsNew,iiOffices,iiOffices,END"
17  "host101,iiInfo,iiResearch,END"
...
> head(epa.trans)
$host1006
[1] "iidocsiiWhatsNew" "iiOffices"        "iiOffices"        "END"
$host101
[1] "iiInfo"     "iiResearch" "END"
...
> tail(epa.stream)
...
3298 "host987,iidocsiiWelcomeiiEPA,iiPIC,END"
3310 "host996,iiRules,iiInitiatives,END"
> tail(epa.trans)
```

```
...
$host987
[1] "iidocsiiWelcomeiiEPA" "iiPIC"                "END"
$host996
[1] "iiRules"       "iiInitiatives" "END"
```

⇒ epa.trans object 는 이제 clickstream 분석을 위한 준비가 되었음

▼ 14.4.3 Estimating the Markov Chain

- State의 수를 나타내는 order 결정 필요
- 일반적으로, order=1 : 다음 상태는 오직 현재 상태에만 의존, 즉, 현재 상태에 의해서만 결정된다.

```
> epa.mc <- fitMarkovChain(epa.trans, order=1)
```

- epa.mc object의 transition matrix:

```
> epa.mc@transitions
$`1`
                        END     iiInfo iiInitiatives     iiNews  iiOffices
END                       0 0.24358974     0.19696970 0.36923077 0.39743590
iiInfo                    0 0.11538462     0.03030303 0.07692308 0.01282051
iiInitiatives             0 0.02564103     0.07575758 0.04615385 0.03846154
iiNews                    0 0.10256410     0.01515152 0.03076923 0.02564103
iiOffices                 0 0.07692308     0.03030303 0.04615385 0.10256410
...
```

column-to-row 방식으로 읽는다면;

{info⇒infor} p=0.115

{info⇒News} p=0.102

{New⇒info} p=0.0769

{office⇒End} p= 0.397

Q: End page로 갈 확률이 가장 높은 케이스는?

⇒ Office page, 해당 페이지에서 유저 retention 을 높일 수 있는 액션 필요

(데이터 클리징만 잘 된다면-95%의 노력 투입 필요, 이런 인사이트는 쉽게 발견할 수 있음)

▼ 14.4.4 Visualizing the MC Results : Transition matrix를 heat map 형태로 전환해서 검토하는 방법

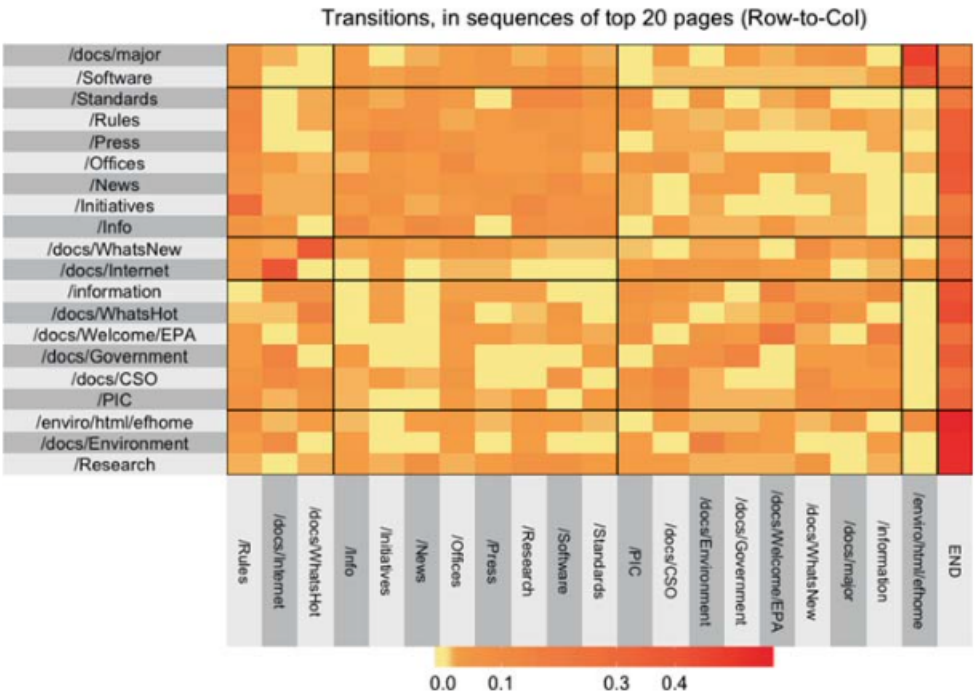

Fig. 14.5 Heat map of the EPA Top 20 page Markov chain transitions using `superheat`, clustering the rows and columns. Note that the transition matrix here is read as transitioning from the *row* to the *column*, as is traditional for Markov chains

- transition matrix를 별도 object로 추출해서, "/" 등의 character 교체 작업, t()를 활용해서 읽기 쉬운 row-to-column 형태로 전환

```
> epa.mc.mat <- t(epa.mc@transitions[[1]])      # t() because easier to read
> dimnames(epa.mc.mat)[[1]] <- gsub("ii", "/", dimnames(epa.mc.mat)[[1]])
> dimnames(epa.mc.mat)[[2]] <- gsub("ii", "/", dimnames(epa.mc.mat)[[2]])
```

- superheat() 과 디자인 수정 옵션 활용

```
> library(superheat)                          # install if needed
> set.seed(70510)
> superheat(epa.mc.mat[-1, ],                 # remove transitions from *END*
+           bottom.label.size = 0.4,
+           bottom.label.text.size = 3.5,
+           bottom.label.text.angle = 270,
+           left.label.size = 0.3,
+           left.label.text.size = 4,
+           heat.col.scheme = "red",
+           n.clusters.rows = 5, n.clusters.cols = 5,
+           left.label = "variable", bottom.label = "variable",
+           title="Transitions, in sequences of top 20 pages (Row-to-Col)")
```

- Pattern:
  - the close relationship between the "WhatsHot" and "Internet" pages
  - a strong association of "efhome" (the unitary page in the next- to-last column cluster) as a destination from "docs/major"and "Software"
- Plot() 활용한 그래프 레이아웃 변경:

```
> set.seed(59911)
> plot(epa.mc, minProbability=0.25)      # layout varies; partially random
```

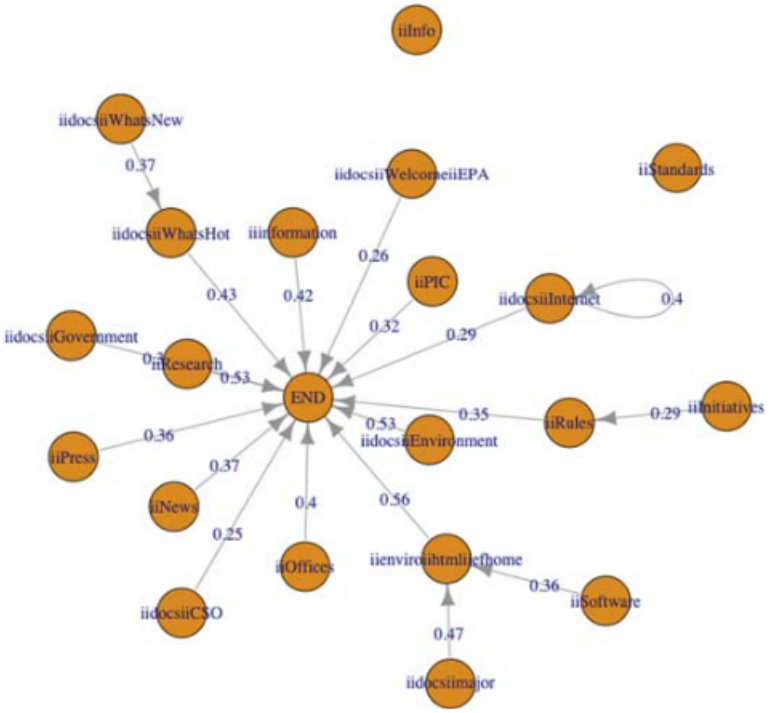

Fig. 14.6 Graph of the EPA Top 20 page transitions in the Markov chain model, for transition probabilities ≥ 0.25, using the default plot function method Markov chains with a

▼ 14.4.5 Higher Order Chains and Prediction

- 다수의 페이지로 부터 얻어나는 전환 모델이 필요한 경우? Markov chain with a higher order parameter
- 이를 위해서는 적어도 3개 이상의 세션이 있는 경우가 필요, End state을 고려한다면 length>4 :

```
> epa.trans.ge3 <- epa.trans[lapply(epa.trans, length) >= 4]
> epa.mc2       <- fitMarkovChain(epa.trans.ge3, order=2)
```

- This yields two @transition matrices : 이전에 일어난 두 개의 확률을 합한 것

  Consider sequence 160 in our data, which happens to have length of 10 page views.

```
> epa.trans[160]
$host1632
 [1] "iiRules"              "iidocsiimajor"        "iidocsiiInternet"
 [4] "iidocsiiGovernment"   "iiinformation"        "iidocsiiWelcomeiiEPA"
 [7] "iiPIC"                "iiRules"              "iiRules"
[10] "iiInitiatives"        "END"
> epa.ex <- new("Pattern", sequence=head(unlist(epa.trans[160]), -1))
```

- We use predict() to predict the next page (“ dist=1” means a distance of one page):

```
> predict(epa.mc2, epa.ex, dist=1)
Sequence: iiRules
Probability: 0.2858429 ...
```

- The next page is most likely to be “Rules”, with probability 0.286. We could predict additional pages after that, if we wished, such as the next 4 likely pages:

```
> predict(epa.mc2, epa.ex, dist=4)     # most just end up with "END"
Sequence: iiRules END
Probability: 0.0730788 ...
```

After one or two pages, sequences are likely to arrive at the end state. This is consistent with the paths shown for the first order model in Fig. 14.6. End states may dominate the model for a simple reason: there may be only one or a few end states, but many other

states lead to them. Thus, transitions to the relatively small number of end states will have higher probabilities. If you are interested in transitions among the other states (pages), it may be helpful to do a parallel analysis without the end states.

▼ **Discussion and Questions (30′)**

Q. 업무에서 Behavior Sequences 를 분석한 사례는, 분석 후 발견한 이슈, 이에 대응 방안을 공유해 주실까요?

- 이승희 :
- 이진재 :
- Eunice Seo :
- 손경희 :
- 강동오 :
- 정시앙 :
- 박규서 :
- 채충일 :
- 윤승원 :

▼ 전체 과정 회고 (30′)

빈 토글입니다. 클릭하거나 내부로 블록을 드롭하세요.