Data Analysis

가설

2019년의 월별 지하철역(1~9호선) 이용객 수 상위 5개역 분석

정보보안암호수학과 20182209 김수빈

데이터 준비, 결합

목차

- 데이터 준비
  - 데이터 불러오기
  - 데이터형태 파악하기
    - 구조 및 구성변수의 형태
    - 데이터 값의 특성 등을 중심으로 파악

- 데이터 결합
  데이터 파악

데이터 준비, 결합

데이터 불러오기

- 방법
  1. ‘서울 열린데이터 광장’에서 ‘서울시 지하철호선별 역별 승하차 인원 정보’ 중 2019년 1월~12월
  (‘CARD_SUBWAY_MONTH_201901.csv’ ~ 'CARD_SUBWAY_MONTH_201912.csv’)에 해당되는 파일을 다운받는다.

  2. R script에서 다운받은 12개의 파일을 불러온다.

- 개념설명
  2019년의 월별 지하철역 이용객 수를 분석할 것이기 때문에
  ‘CARD_SUBWAY_MONTH_201901.csv’ ~ 'CARD_SUBWAY_MONTH_201912.csv’ 12개 파일을 모두 다운받는다.

데이터 준비, 결합

데이터 불러오기

- 명령코드

```
subway1 <- read.csv('c:/Rstudy/subway/CARD_SUBWAY_MONTH_201901.csv')
subway2 <- read.csv('c:/Rstudy/subway/CARD_SUBWAY_MONTH_201902.csv')
subway3 <- read.csv('c:/Rstudy/subway/CARD_SUBWAY_MONTH_201903.csv')
subway4 <- read.csv('c:/Rstudy/subway/CARD_SUBWAY_MONTH_201904.csv')
subway5 <- read.csv('c:/Rstudy/subway/CARD_SUBWAY_MONTH_201905.csv')
subway6 <- read.csv('c:/Rstudy/subway/CARD_SUBWAY_MONTH_201906.csv')
subway7 <- read.csv('c:/Rstudy/subway/CARD_SUBWAY_MONTH_201907.csv')
subway8 <- read.csv('c:/Rstudy/subway/CARD_SUBWAY_MONTH_201908.csv')
subway9 <- read.csv('c:/Rstudy/subway/CARD_SUBWAY_MONTH_201909.csv')
subway10 <- read.csv('c:/Rstudy/subway/CARD_SUBWAY_MONTH_201910.csv')
subway11 <- read.csv('c:/Rstudy/subway/CARD_SUBWAY_MONTH_201911.csv')
subway12 <- read.csv('c:/Rstudy/subway/CARD_SUBWAY_MONTH_201912.csv')
```

- 명령코드 설명
  "CARD_SUBWAY_MONTH_201901.csv“ ~ “CARD_SUBWAY_MONTH_201901.csv” 파일을
  subway1 ~ subway12 변수에 각각 저장한다.

데이터 준비, 결합

데이터 형태 파악하기

-명령 코드 결과 (View(subway1))

	사용일자	노선명	역명	승차총승객수	하차총승객수	등록일자
1	20190101	2호선	을지로4가	3862	3728	20190104
2	20190101	2호선	을지로3가	8104	7554	20190104
3	20190101	2호선	을지로입구	22478	21330	20190104
4	20190101	2호선	시청	8381	6049	20190104
5	20190101	1호선	동묘앞	8045	8504	20190104
6	20190101	1호선	청량리(서울시립대입구)	15007	15397	20190104
7	20190101	1호선	제기동	10187	10178	20190104
8	20190101	1호선	신설동	6832	6930	20190104
9	20190101	1호선	동대문	9337	10457	20190104
10	20190101	1호선	종로5가	13578	13282	20190104

사용일자, 노선명, 역명, 승차총승객수, 하차총승객수, 등록일자 변수로 이루어져 있다.
사용일자 데이터는 20190101 ~ 20190131까지 일별로 저장되어 있다.

데이터 준비, 결합

데이터 형태 파악하기

-명령 코드 결과 (table(subway1$노선명))

1호선	2호선	3호선	4호선	5호선	6호선
3650	18250	12340	9490	18615	13700
7호선	8호선	9호선	9호선2~3단계	경강선	경부선
18615	6205	9125	4745	4015	14235
경원선	경의선	경인선	경춘선	공항철도 1호선	과천선
10734	9427	7300	6935	5110	2920
분당선	수인선	안산선	우이신설선	일산선	장항선
12596	4745	4745	4745	3672	2190
중앙선					
7665					

노선명은 1호선~9호선 이외에도 경의선, 경춘선, 과천선, 분당선 등 많은 데이터들이 저장되어 있다.

데이터 준비, 결합

데이터 결합

- 개념 설명
  subway1 ~ subway12 데이터들을 결합한다.

- 명령 코드

```
subway <- bind_rows(subway1, subway2, subway3, subway4, subway5, subway6, subway7,
subway8, subway9, subway10, subway11, subway12)
```

- 명령 코드 설명
  bind_rows 함수를 이용해서 subway1 ~ subway12 데이터들을 세로결합해서 subway 변수에 저장한다.

데이터 가공 및 통합

목차

- 1차 데이터 추출 : 1~9호선
- 2차 데이터 추출 : 상위 5개 역
- 최종 데이터셋(월별 이용객수) 구하기
  - 사용일자 데이터 이름 변경
  - 데이터 요약하기

데이터 가공 및 통합

1차 데이터 추출 : 1~9호선

- 개념설명
  subway에서 노선명이 1호선 ~ 9호선인 데이터들만 추출한다.

- 명령 코드

```
subway <- subset(subway, 노선명 %in% c('1호선','2호선','3호선','4호선','5호선','6호선','7호선','8호선','9호선'))
```

- 명령 코드 결과 (table(subway$노선명))

```
1호선 2호선 3호선 4호선 5호선 6호선 7호선 8호선 9호선
 3650 18250 12340  9490 18615 13700 18615  6205  9125
```

데이터 가공 및 통합

2차 데이터 추출 : 상위 5개 역

- 개념설명
  이용객 수 = 승차총승객수 + 하차총승객수
  상위 5개 역명을 구하기 위해 역명을 그룹화하여 데이터를 요약한다.

- 명령 코드

```
subway$tot <- subway$승차총승객수 + subway$하차총승객수
st5 <- subway %>% group_by(역명) %>% summarise(tot=sum(tot)) %>% arrange(desc(tot)) %>% head(5)
st5_name <- st5$역명
st5_name
```

- 명령 코드 설명
  1. subway의 승차총승객수와 하차총승객수를 더해 tot 파생변수를 만든다.
  2. 역명을 기준으로 그룹화 해서 각 tot들을 더해서 내림차순으로 정렬하고, 상위 5개 역을 st5 변수에 저장한다.
  3. 상위 5개 역명을 st5_name 변수에 저장한다.

- 명령 코드 결과 (st5_name)

```
[1] "잠실(송파구청)" "강남"          "고속터미널"    "홍대입구"      "서울역"
```

데이터 가공 및 통합

최종 데이터 셋(월별 이용객 수) 구하기 – 사용일자 데이터 이름 변경

- 개념설명
  필요한 사용일자 데이터는 일별 데이터가 아니라 월별 데이터가 필요하기 때문에 월별로 데이터 이름을 변경해준다.

- 방법
  20190101 ~ 20190131 → 01 → Jan
  20190201 ~ 20190230 → 02 → Feb

  ,,,

  20191201 ~ 20191231 → 12 → Dec

데이터 가공 및 통합

최종 데이터 셋(월별 이용객 수) 구하기 – 사용일자 데이터 이름 변경

- 명령 코드

```
subway$사용일자 <- substr(subway$사용일자, 5, 6)
subway$사용일자 <- ifelse(subway$사용일자 == '01', 'Jan',
                          ifelse(subway$사용일자 == '02', 'Feb',
                          ifelse(subway$사용일자 == '03', 'Mar',
                          ifelse(subway$사용일자 == '04', 'Apr',
                          ifelse(subway$사용일자 == '05', 'May',
                          ifelse(subway$사용일자 == '06', 'Jun',
                          ifelse(subway$사용일자 == '07', 'Jul',
                          ifelse(subway$사용일자 == '08', 'Aug',
                          ifelse(subway$사용일자 == '09', 'Sep',
                          ifelse(subway$사용일자 == '10', 'Oct',
                          ifelse(subway$사용일자 == '11', 'Nov', 'Dec'))))))))))
table(subway$사용일자)
```

데이터 가공 및 통합

최종 데이터 셋(월별 이용객 수) 구하기 – 사용일자 데이터 이름 변경

- 명령 코드 설명
  - 먼저 사용일자 데이터들이 20190101 ~ 20190131처럼 일별로 되어있는데, 나는 월별 데이터를 원하기 때문에 substr(subway$사용일자, 5, 6)을 이용해 5번째부터 6번째 문자열만 잘라 01 ~ 12 데이터를 얻는다.

  - 그 후 ifelse 함수를 이용해서 사용일자 데이터가 01이면 Jan, 02면 Feb, 03이면 Mar, 04면 Apr, 05면 May, 06이면 Jun, 07이면 Jul, 08이면 Aug, 09면 Sep, 10이면 Oct, 11이면 Nov, 12면 Dec으로 이름을 변경해준다.

- 명령 코드 결과 (table(subway$사용일자))

Apr	Aug	Dec	Feb	Jan	Jul	Jun	Mar	May	Nov	Oct	Sep
17749	18319	18333	16543	18334	18305	17724	18342	18336	17741	18320	17723

데이터 가공 및 통합

최종 데이터 셋(월별 이용객 수) 구하기 – 데이터 요약하기

- 개념설명
  월별 이용객 수를 구하기 위해 사용일자, 역명별로 그룹화한 후 데이터를 요약한다.

- 명령 코드

```
month_st5 <- subway %>% filter(역명 %in% st5_name) %>% group_by(사용일자, 역명) %>% summarise(tot=sum(tot))
View(month_st5)
```

- 명령 코드 설명
  subway에서 역명이 st5_name인 데이터 중에 사용일자와 역명을 기준으로 그룹화 하고, 그 그룹을 기준으로 각각의 tot들을 더해준다.

데이터 가공 및 통합

최종 데이터 셋(월별 이용객 수) 구하기 – 데이터 요약하기

- 명령 코드 결과 (View(month_st5))

	사용일자	역명	tot
1	Apr	강남	6055208
2	Apr	고속터미널	5802498
3	Apr	서울역	4616377
4	Apr	잠실(송파구청)	6672071
5	Apr	홍대입구	4965649
6	Aug	강남	6350121
7	Aug	고속터미널	6141724
8	Aug	서울역	4484502
9	Aug	잠실(송파구청)	6263663
10	Aug	홍대입구	5215759

사용일자와 역명을 기준으로 tot가 계산되었다.

데이터 분석 및 결론

목차

- 최적 시각화 레벨 설정
- 그래프를 통해 분석 결과 시각화

데이터 분석 및 결론

최적 시각화 레벨 설정

- 개념설명
  역명은 이용객수가 많은 순으로, 사용일자는 날짜 순으로 정렬한다.

- 명령 코드

```
month_st5$역명 <- factor(month_st5$역명, levels = st5_name)
month_st5$사용일자 <- factor(month_st5$사용일자, levels = c('Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul',
'Aug', 'Sep', 'Oct', 'Nov', 'Dec'))
```

- 명령 코드 설명
  month_st5의 역명을 st5_name 순으로 정렬한다.
  month_st5의 사용일자를 'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec’ 순으로 정렬한다.

데이터 분석 및 결론

그래프를 통해 분석 결과 시각화

- 개념설명
  ggplot을 이용해서 2019년 월별 지하철역(1호선~9호선) 이용객 수 상위 5개 역을 시각화 해보기

- 명령 코드

```
ggplot(month_st5, aes(x=사용일자, y=tot, fill=역명)) + geom_col() + xlab('month') +
scale_y_continuous(labels = scales::comma)
```

- 명령 코드 설명
  x좌표에 사용일자, y좌표에 tot 값, fill에 역명을 놓고, 막대그래프를 그려 시각화 한다.
  x좌표 제목을 month라고 놓고, y좌표 값들에 천단위로 쉼표를 찍어준다.

데이터 분석 및 결론

그래프를 통해 분석 결과 시각화

- 명령 코드 결과