통계모델링 프로젝트

- 광주광역시 시내버스 승하차 교통량 분석 -

2024년 11월 12일

청주대학교 데이터사이언스학과

2021012800 베네딕터스 에스라 헤르노오

목차

1. 소개

1.1. 프로젝트 개요

이 프로젝트의 목적은 한국 정부가 제공한 공공 데이터의 실제 데이터를 사용하여 광주의 버스 교통 패턴을 분석하는 것입니다. 도시 전체의 승객 흐름을 조사함으로써 이용 동향과 피크 교통 시간을 더 잘 이해할 수 있으며 광주의 여러 지역에서 사람들이 어떻게 이동하는지에 대한 정보를 제공할 수 있습니다. 이 프로젝트는 도시 계획, 자원 할당 및 도시 버스 시스템의 전반적인 효율성 향상에 도움이 될 수 있는 교통 요구 사항에 대한 통찰력을 제공합니다.

1.2. 프로젝트의 중요성

대중교통은 광주의 많은 주민들에게 필수적이며, 안정적이고 저렴한 가격으로 도시를 이동할 수 있는 방법을 제공합니다. 시간, 요일, 역에 따라 교통량이 어떻게 달라지는지 이해하면 시에서 데이터를 기반으로 버스 교통 시스템을 개선하기 위한 결정을 내리는 데 도움이 됩니다. 예를 들어, 특정 역의 가장 바쁜 시간을 알고 있다면 버스 일정을 최적화하여 대기 시간을 줄이고 서비스 품질을 향상시킬 수 있습니다. 또한 이 분석은 탑승 및 하차 패턴의 추세를 강조할 수 있으며 이는 피크 시간 동안 군중을 관리하는 데 중요할 수 있습니다.

1.3. 프로젝트 범위 및 제한 사항

이 프로젝트는 다양한 시간 간격과 정류장에서의 승객 수에 주목하면서 일정 기간 동안의 광주 버스 교통 데이터에 초점을 맞췄습니다. 데이터세트에는 날짜(날짜), 역번호(역번호), 역명(역명), 승하차 상태(구분), 시간대(시간대), 교통량(traffic_count)과 같은 데이터 필드가 포함됩니다. 그러나 몇 가지 제한 사항이 있습니다. 이 분석에서는 버스 이용에 영향을 미칠 수 있는 휴일, 기상 조건 또는 기타 외부 요인을 고려하지 않습니다. 또한 데이터 세트는 특정 역과 기간으로 제한되어 있으므로 여기에서 도출된 결론은 전체 버스 네트워크에 완전히 일반화되지 않을 수 있습니다.

2. 배경

2.1. 광주버스시스템 소개

대중교통 서비스, 특히 버스 시스템은 광주 시민들에게 도시와 인근 지역으로의 쉬운 이동을 제공하기 때문에 매우 중요합니다. 버스 시스템은 지역 당국에 속해 있으므로 주거 지역과 워크스테이션 구역, 교육 및 레크리에이션 시설을 연결함으로써 주민과 관광객 모두에게 수많은 상호 연관된 기능을 제공합니다. 광주에서는 버스가 올바른 시간표에 따라 운행되며, 대부분의 지역을 상당히 높은 정기 및 빈번 운행으로 운행합니다.

네트워크는 많은 사람과 역에 서비스를 제공하는 여러 경로로 구성되어 있으며 하루에 최소 수천 명의 사람이 운영됩니다. 광주 버스 시스템에는 급행, 쾌속, 일반 등 여러 버스 노선이 있으며 각각 통근 시스템 내에서 특정 대중의 요구를 충족합니다. 도심의 거리를 단축하고 교외로 진출하더라도 지리적 위치에 관계없이 모든 시스템 사용자에게 효율적인 서비스를 제공하는 것이 목적입니다.

이러한 맥락에서 우리는 여러 역에 탑승하는 승객 수(승차) 및 하차하는 승객 수(하차)와 같은 특정 매개변수에 대한 연구를 강조합니다. 이 데이터 조각은 대부분의 승객이 운행 시간, 요일 및 계절을 사용하는 가장 많은 탑승 경로에 대한 정보를 제공하고 전반적인 승객 프로필을 제공합니다. 이를 통해 우리는 도시의 버스 교통 시스템 기능을 수요의 함수로 모델링할 수 있으며, 이에 대한 지식은 기존 시스템을 수정하고 광주 대중 교통 시스템에 새로운 자원을 할당하는 데 중추적인 역할을 할 것입니다.

3. 데이터 수집

3.1. 데이터 소스

본 분석을 위한 데이터는 다양한 공공 데이터 세트에 대한 공개 접근을 제공하는 정부 플랫폼인 한국의 공공 데이터 포털(data.go.kr)에서 얻은 것입니다. 포털에서 제공한 데이터는 2024년 1월 1일부터 2024년 10월 31일까지의 기간을 다루고 있었습니다.

이 데이터는 광주 버스 시스템 전체의 최신 승객 교통 데이터 입력을 위해 수집되었으며, 이는 장기간에 걸친 추세를 조사하는 데 사용할 수 있습니다.

3.2. 데이터 개요

A data.frame: 6 × 28

	날짜	역번호	역명	구분	X03.04시	X04.05시	X05.06시	X06.07시	X07.08시	X08.09시	...	X17.18시	X18.19시	X19.20시	X20.21시	X21.22시	X22.23시	X23.00시	X00.01시	X01.02시	X02.03시
	<chr>	<int>	<chr>	<chr>	<int>	<int>	<int>	<int>	<int>	<int>	...	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>
1	2024-01-01	1101	관암	승차	0	66	61	55	50	87	...	120	129	54	47	33	13	13	2	0	0
2	2024-01-01	1101	관암	하차	0	53	50	74	48	51	...	145	130	111	85	73	94	62	17	0	0
3	2024-01-01	1102	신흥	승차	0	0	18	106	21	40	...	41	37	30	15	16	8	3	0	0	0
4	2024-01-01	1102	신흥	하차	0	0	9	104	17	15	...	45	69	39	37	27	39	20	6	0	0
5	2024-01-01	1103	대동	승차	0	27	94	31	51	77	...	116	84	102	52	49	37	28	0	0	0
6	2024-01-01	1103	대동	하차	0	25	85	58	70	32	...	103	112	85	98	85	106	99	14	0	0

데이터 세트에는 관측소 및 시간 수준의 일일 교통량을 자세히 설명하는 여러 변수가 포함되어 있습니다. 중요한 변수로는 다음과 같습니다.

- 날짜: 데이터 수집일을 나타내기 위해 yyyy-mm-dd 형식으로 기록된 날짜입니다.
- 역번호: 각 버스 정류장의 식별 번호 역할을 하는 버스 정류장 코드입니다.
- 역명: 지정된 버스 정류장의 이름을 의미하는 버스 정류장 제목.
- 구분: 탑승(승차) 또는 하차(하차) 관련 데이터를 나타내며, 탑승객 통행량 분석과 출국객 통행량 분석이 모두 가능합니다.
- 03-04시 ~ 02-03시: 하루 종일, 즉 이른 아침부터 아주 늦은 밤까지 시간별 교통량을 계산하여 하루 중 시간대에 따라 버스 사용량이 어떻게 달라지는지 보여줍니다.

전체적으로 데이터 세트에는 12,716개의 변수가 포함되어 있으며 각 열은 하루 중 시간 또는 승객 수와 관련된 기타 필드를 나타냅니다. 이를 통해 시간과 공간에 걸쳐 심층적인 트래픽 분석을 보장하고 광범위한 연구의 기반을 마련합니다.

4. 데이터 전처리

4.1. 데이터 형식화

첫 번째 단계에서는 데이터를 분석에 더 적합하게 만들기 위해 데이터를 정리해야 했습니다. '시' 접미사로 구성된 열은 형식 표준화를 위해 기간에서 삭제되었습니다. 이렇게 하면 시간 데이터가 문자열 기반이 아닌 숫자로 생성되어 조작 및 분석이 더 쉬워집니다. 이후 컬럼명 전반에 걸쳐 데이터 일관성을 확보하고, 분석에 방해가 될 수 있는 불규칙성을 점검했습니다.

4.2. 데이터 재구성

데이터가 정리된 후 R의 Pivot_longer 함수를 사용하여 긴 형식으로 재구성하여 데이터를 다시 한 번 재처리했습니다. 이 변환에는 시간별 열을 두 개의 새로운 열인 time_ period 및 Traffic_count로 통합하는 작업이 포함되었습니다.. time_기간 열은 이제 하루 중 각 시간 변수로 구성되고, Traffic_count는 관련 승객 수를 보유합니다. 이러한 방식으로 데이터를 재구성하면 여러 시간별 열의 제약 없이 시간이 지남에 따라 패턴을 더 효과적으로 분석할 수 있습니다.

4.3. 만들기 새 데이터세트

날짜	역번호	역명	구분	time_period	traffic_count
<chr>	<int>	<chr>	<chr>	<chr>	<dbl>
2024-01-01	1101	판암	승차	03	0
2024-01-01	1101	판암	승차	04	66
2024-01-01	1101	판암	승차	05	61
2024-01-01	1101	판암	승차	06	55
2024-01-01	1101	판암	승차	07	50

형태를 변경한 후 새 데이터세트(data_clean)은 다음 열로 구성되었습니다.

- 날짜: 데이터 기록의 날짜입니다.
- 역번호: Unique station code.
- 역명: 역 이름.

- 구분: Boarding (승차) or alighting (하차) indicator.
- time_기간: 시간 간격으로 표시되는 시간입니다.
- Traffic_count: 해당 특정 시간의 승객 수입니다.

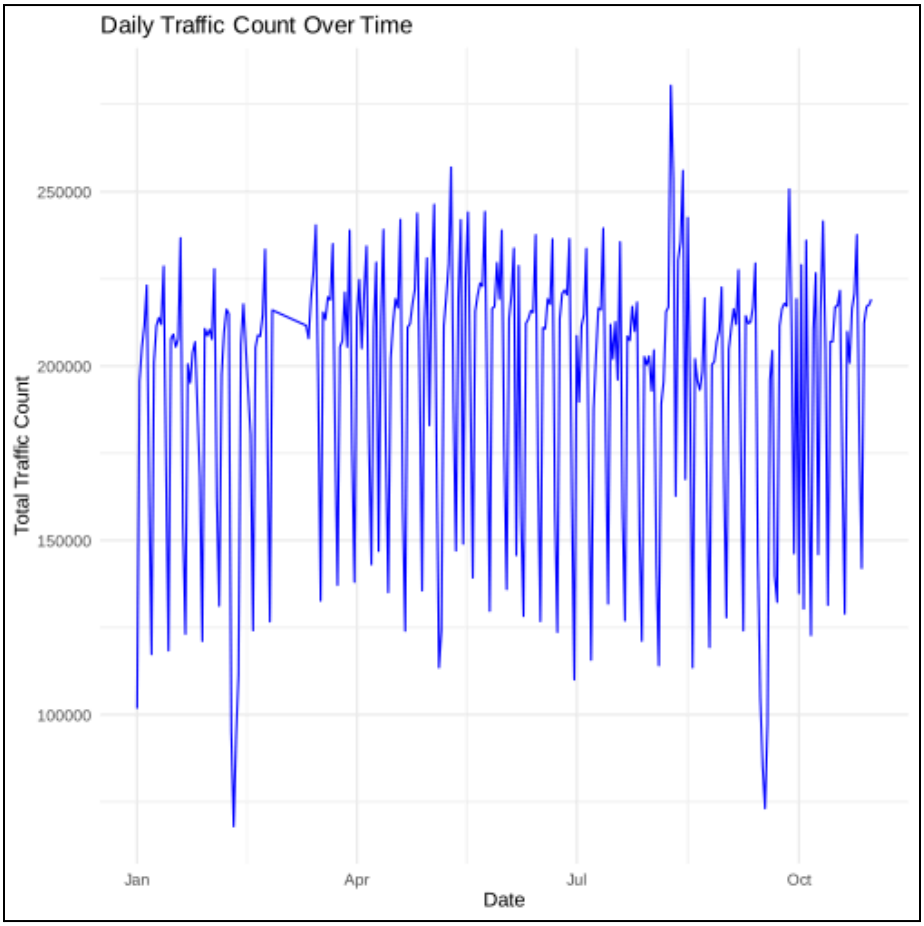
이 구조는 시간 기반 및 역 기반 분석을 위해 데이터에 더 쉽게 접근할 수 있도록 하여 시간과 위치에 따른 탑승 및 하차 패턴을 집중적으로 탐색할 수 있는 기반을 마련합니다.

4.4. 누락된 값 처리

검사 결과 Traffic_count 열에 누락된 값(NA)이 없었으므로 누락된 데이터를 처리하기 위해 추가 조치가 필요하지 않았습니다. 이 검사를 통해 데이터 세트를 분석할 준비가 되었습니다.

5. 탐색적 데이터 분석

5.1. 시간 경과에 따른 일일 트래픽 수



분석을 시작하기 위해 데이터 기간인 2024년 1월부터 10월까지의 전체 트래픽 수를 조사하였다. 이는 일년 내내 버스 이용 추세의 변동을 관찰하는 데 도움이 되었습니다.

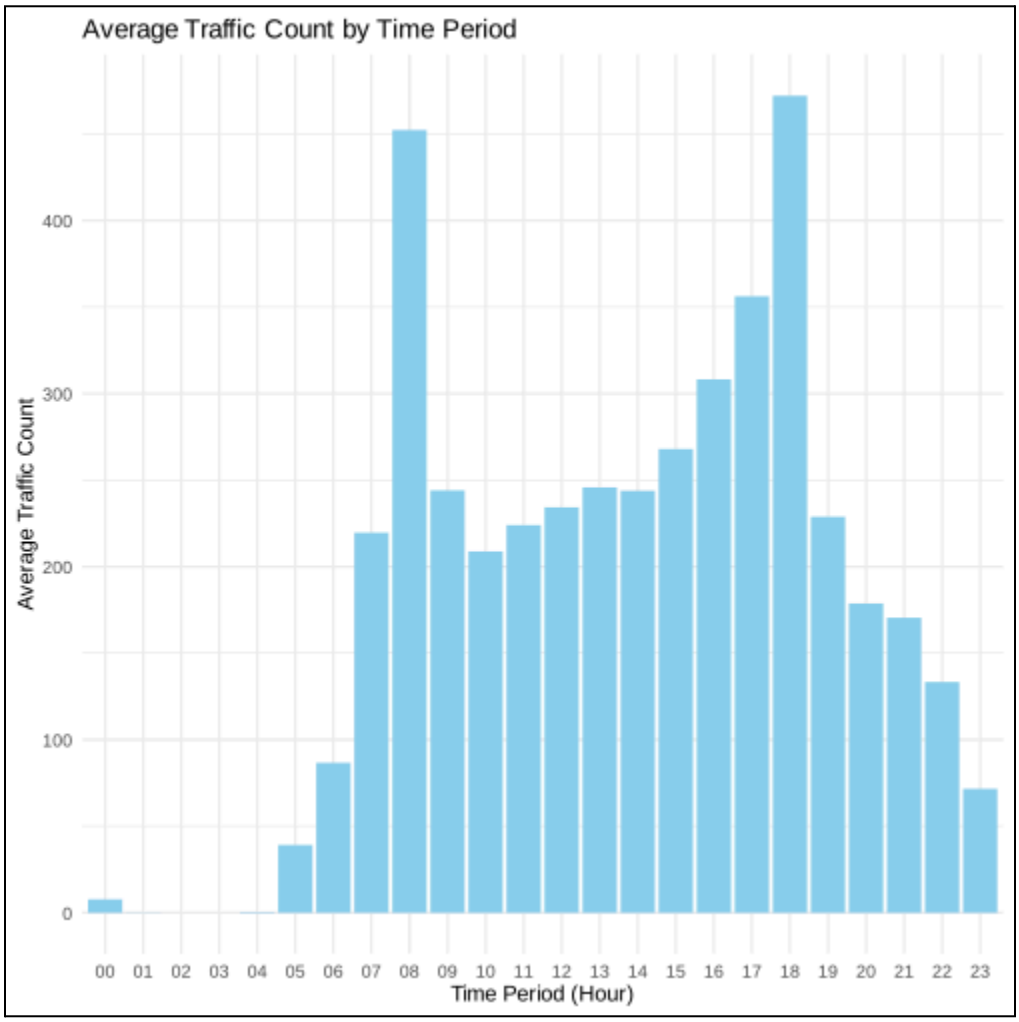
이 기간 동안 몇 가지 중요한 급증과 하락이 있음을 확인할 수 있습니다.

- **2**월 그리고 구월: 이번 달에는 평소 월별 평균 추세를 벗어나 트래픽 수가 크게 감소했습니다.
- 팔월**:** 반면, 이번 달은 급증세를 보이며 다른 달에 비해 트래픽이 많아 월간 추세의 평균 추세선을 넘어섰습니다.

다른 달’ 교통량은 일년 내내 상대적으로 안정적으로 유지되었으며, 일반적으로 월간 승객 수는 100,000명에서 250,000명 사이로 변동했습니다.

이러한 극심한 변동은 계절적 사건, 공휴일 또는 통근 패턴의 변화와 같은 여러 요인으로 인해 발생할 가능성이 높습니다. 스파이크 및 하락의 원인을 결론짓기 위해서는 추가 분석이 필요했습니다.

5.2. 기간별 트래픽 분포

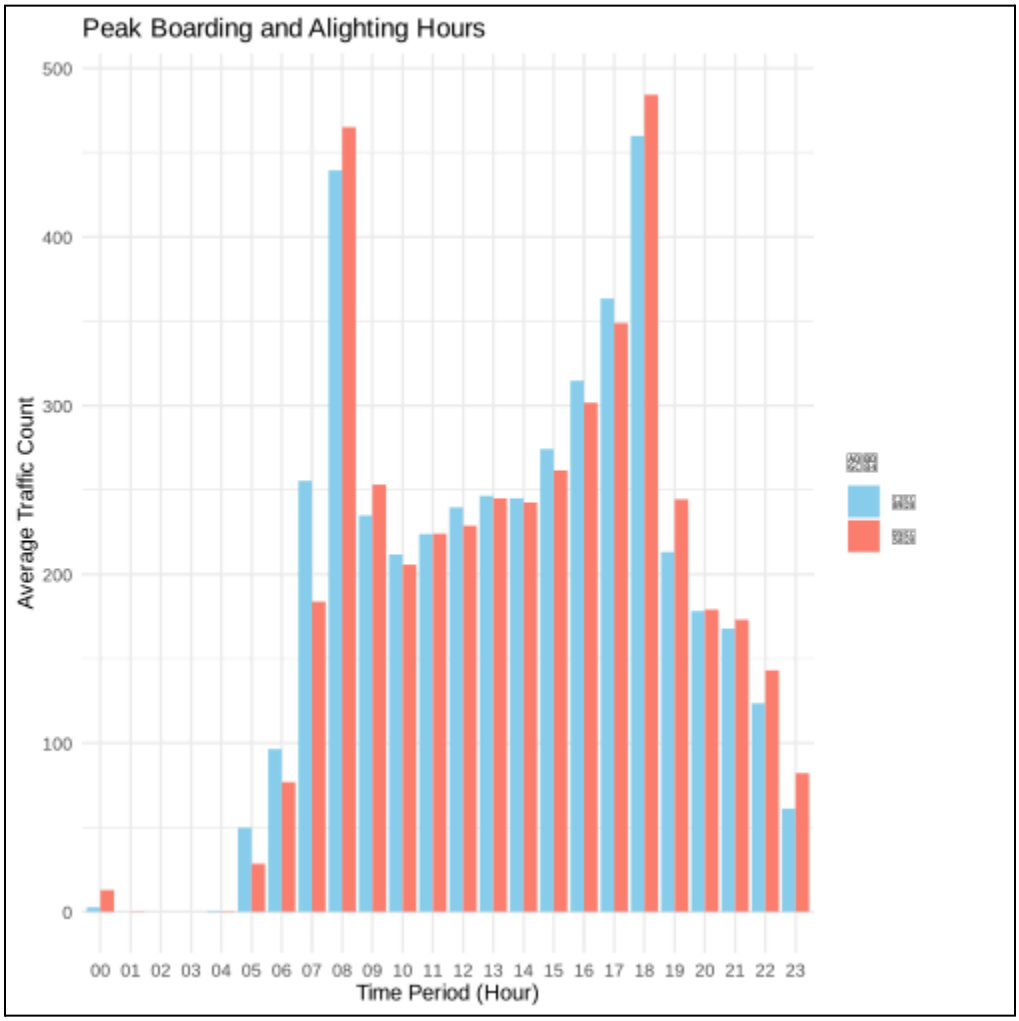


다양한 시간 간격에 대한 교통 분포 분석은 출퇴근 피크 시간과 근무 시간 동안의 일관된 사용량을 반영하여 서로 다릅니다. 쉽게 알아볼 수 있는 최고점 중 하나는 오전 8시인데, 교통량이 450명을 넘는 것은 아마도 사람들이 직장과 학교에 가는 아침 통근 때문일 것입니다. 저녁에 또 다른 눈에 띄는 봉우리에 도달하는 것은 오후 6시경이며, 이 저녁 봉우리에는 460개를 약간 넘는 봉우리가 도달합니다. 이 저녁 피크는 오후 3시부터 오후 5시까지 점진적인 경사를 지나 오후 6시에 피크에 도달하는 승객에 의해 발생합니다.

오전 9시부터 오후 5시까지 하루 종일 교통량이 200명 이하로 떨어지지 않아 승객의 꾸준한 이동을 보여준다. 이 출퇴근 경로에서는 근로자와 학생, 대중교통

수단을 이용하는 다른 사람들이 공유할 수도 있습니다. 이는 버스 서비스 수요의 일일 변동에 대한 정보를 추가로 제공할 수 있는 시간별 추세입니다.

5.3. 피크 탑승 및 하차 시간

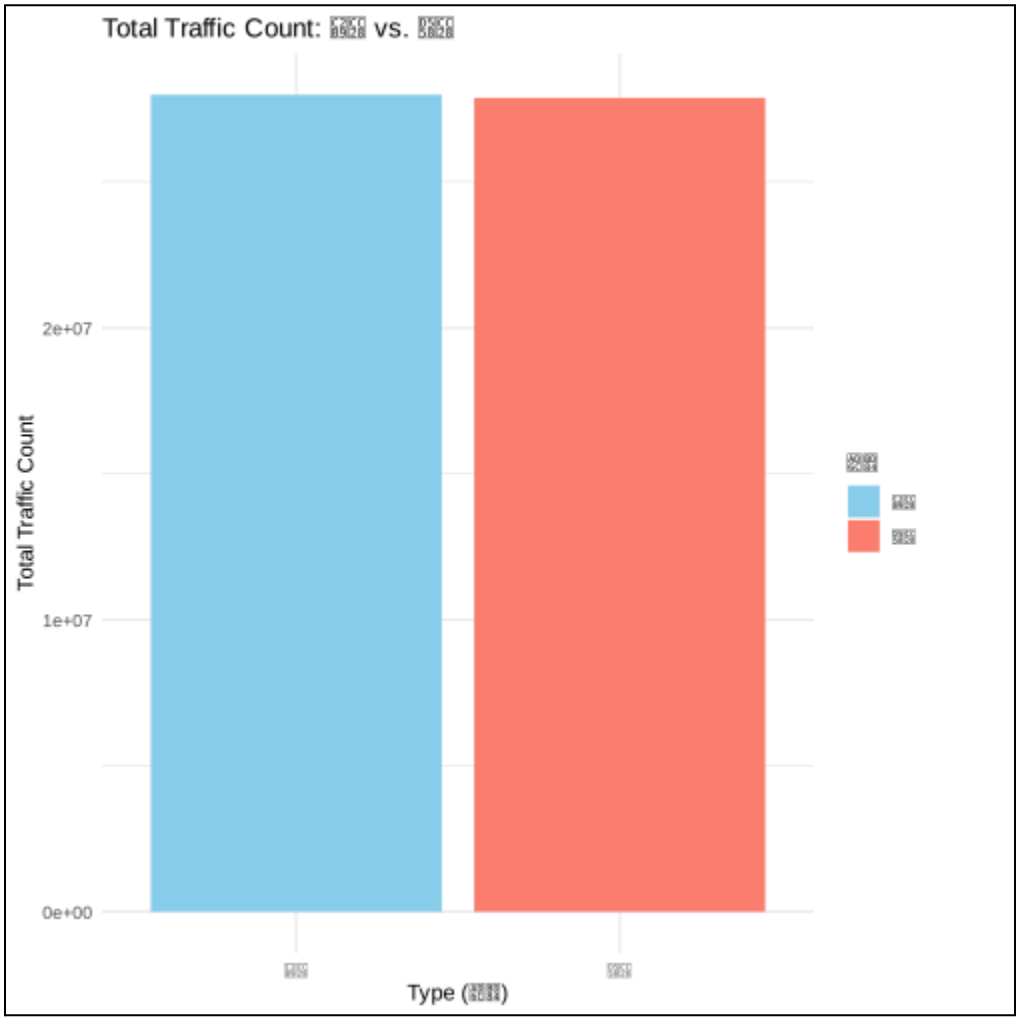


그 결과 가장 활동적인 시간대는 오전 8시와 오후 6시로 주근무시간과 일치하는 것으로 나타났다. 오전 8시 탑승인원은 약 439명, 하차인원은 약 465명이다. 이 시간에 꽤 많은 사람들이 버스에서 내려 사무실, 학교 또는 기타 아침 약속으로 향할 것으로 예상되기 때문에 이러한 차이는 분명합니다.

동일한 사례가 저녁 6시에도 발생하여 버스에 탑승하는 승객은 약 460명으로 많은 반면, 하차하는 승객의 수는 약 484명으로 상당히 높습니다. 이러한 추세는 교통 정체가 있는 저녁 피크 시간대의 특징입니다. 승객들은 퇴근 후 집이나 다른 곳으로 가기 위해 서둘러 버스에 탑승합니다.

이러한 탑승 및 하차 피크 시간에는 아침 및 저녁 피크 시간에 여행 수요 관리 및 용량 활용에 많은 중점이 필요했습니다. 또한, 피크 시간 동안 승객의 탑승 및 하차 수치의 작은 차이는 예를 들어 아침에 하차하는 승객이 많고 저녁에 탑승하는 승객이 늘어나는 등 일반적인 흐름 추세를 나타냅니다.

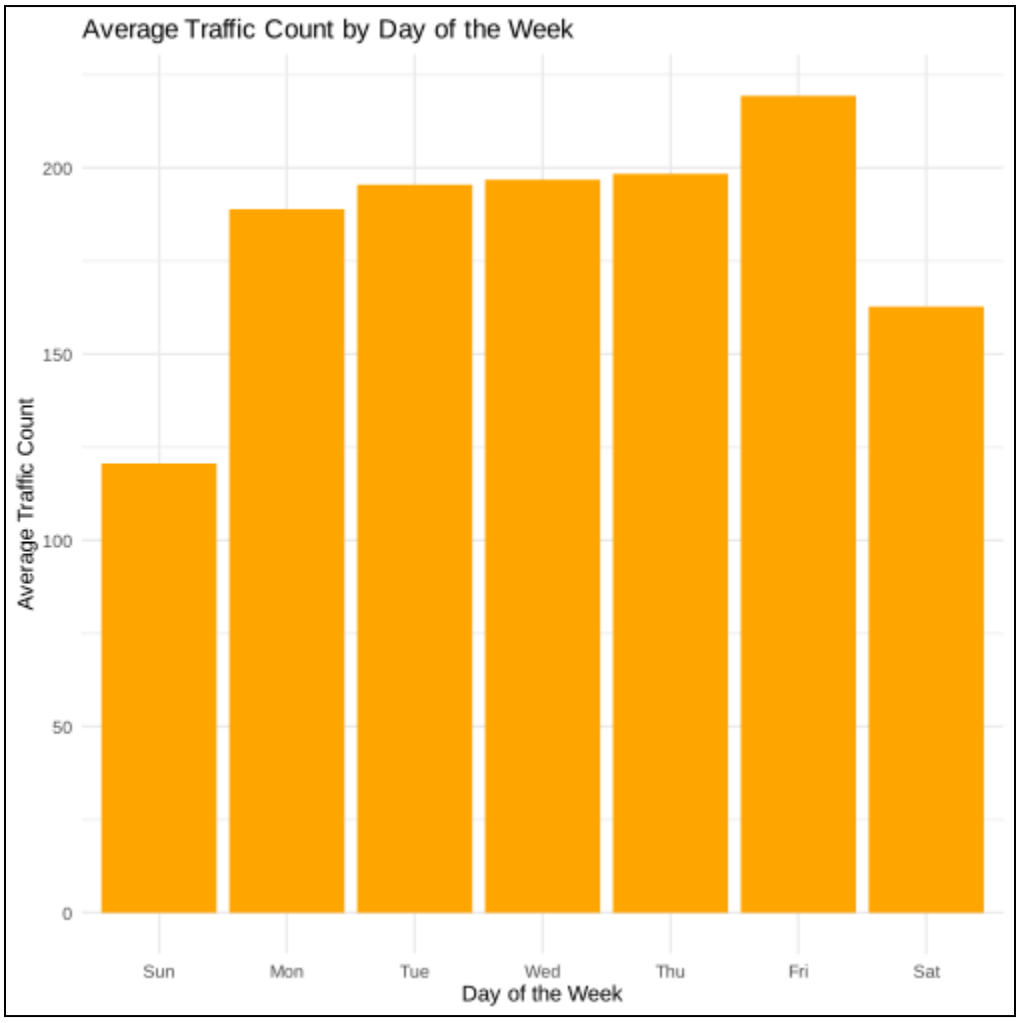
5.4. **Total Traffic:** 승차 **vs** 하차



광주버스 시스템 내에서 관찰된 기간 동안의 승차 및 하차의 총 횟수입니다. 전체 탑승객수는 27,985,335명, 하차객수는 27,878,543명으로 바짝 뒤따랐다.

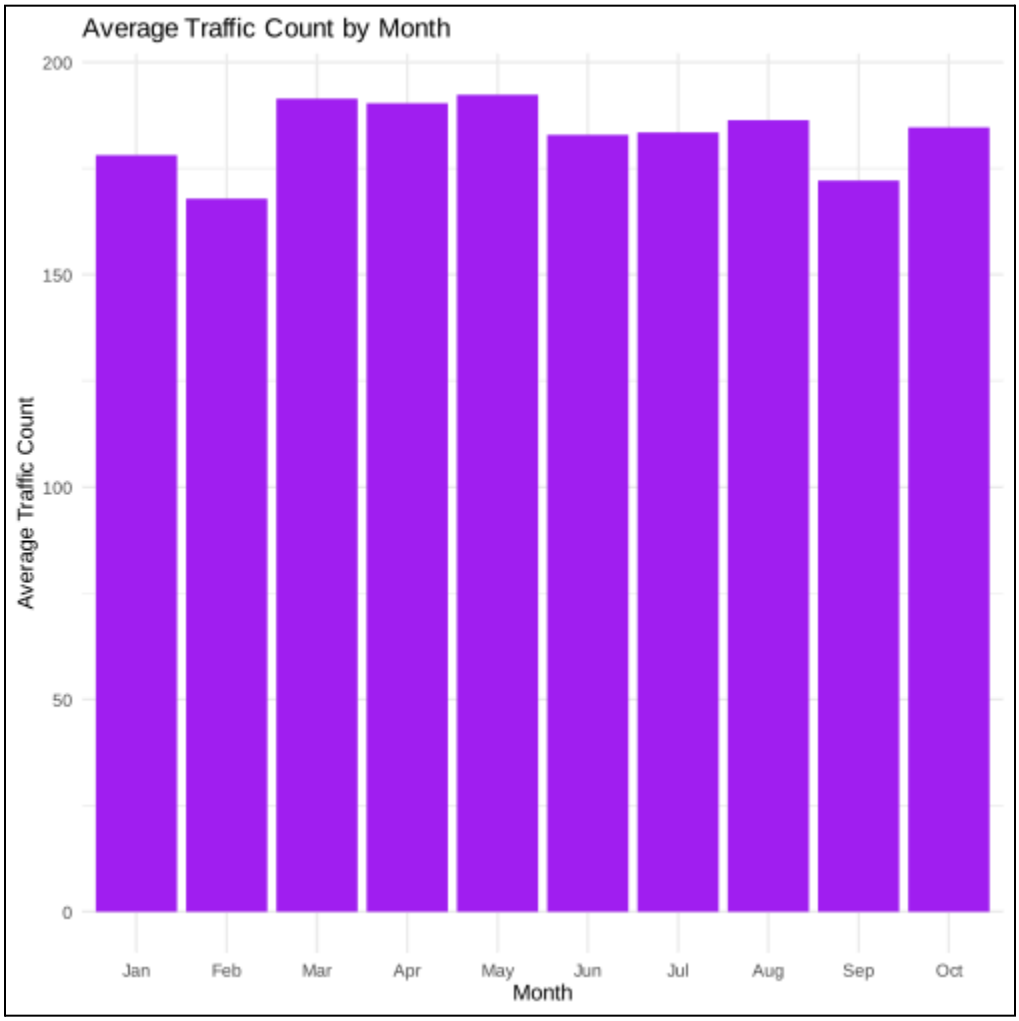
탑승 횟수와 하차 횟수가 거의 동일하다는 것은 시스템에 들어오고 나가는 승객의 균형이 잘 잡힌 분포를 의미합니다. 이러한 균형은 효과적인 경로 적용 범위와 시스템 활용도를 나타내는 긍정적인 지표입니다. 이는 대부분의 승객이 특정 경로에서 상당한 오버플로나 사용량 부족 없이 네트워크 내에서 여행을 완료하고 있음을 의미하기 때문입니다.

5.5. 요일별 교통량



데이터를 통해 월요일부터 목요일까지의 평균 승객 수는 약 175명에서 200명 사이로 비교적 안정적으로 유지되고 있음을 알 수 있습니다. 이는 통근이나 학교 등 정규적인 주중 활동에 영향을 받을 가능성이 있는 일관된 수요를 시사합니다. 특히 금요일은 평균 승객 수가 213명을 넘어 최대 승객 수를 기록했습니다. 이러한 증가는 사교 모임 및 일부 서비스의 운영 시간 연장을 포함하여 근무일이 끝날 때 발생하는 추가 활동에 기인할 수 있습니다.

5.6. 월별 평균 트래픽 수

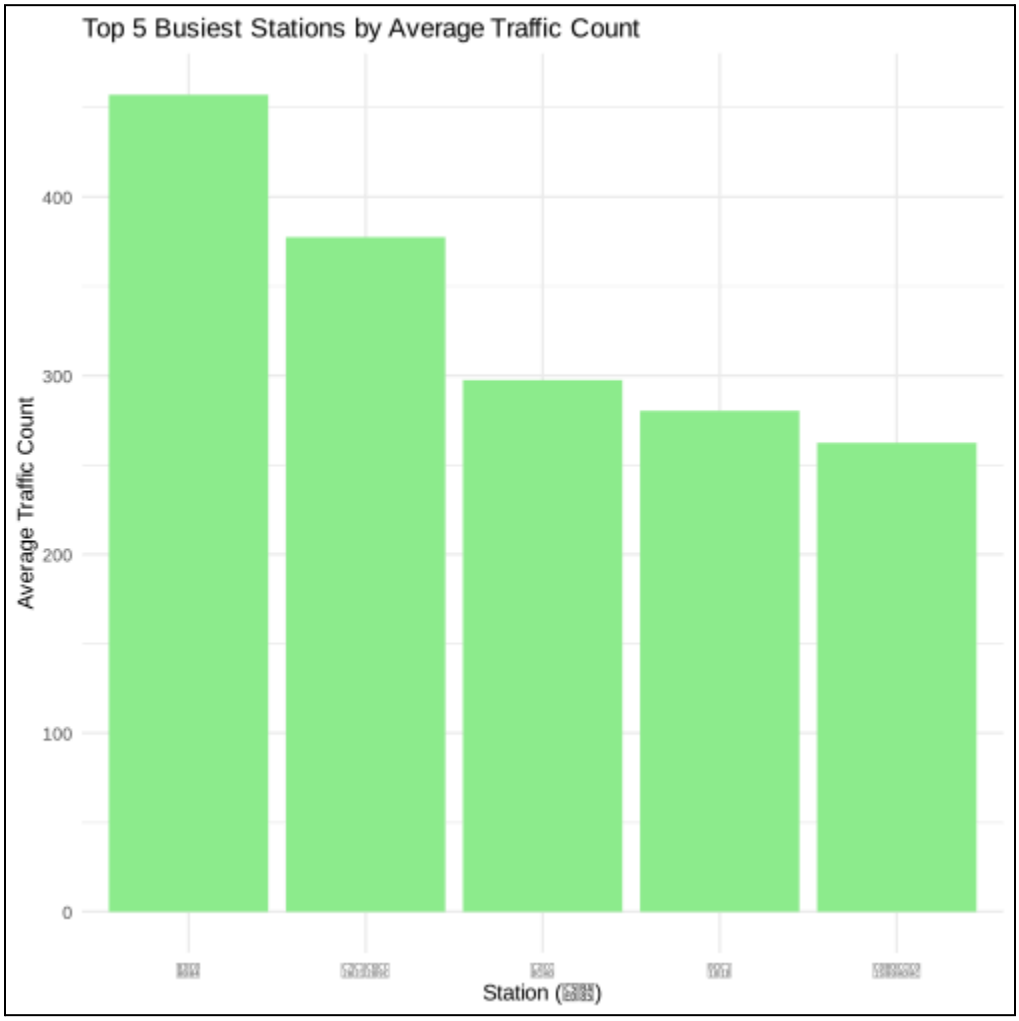


월평균 교통량에 대한 데이터에 따르면 광주버스 시스템에 대한 수요는 연중 지속적으로 존재하며, 시스템을 이용하는 승객은 월 175~185명에 이릅니다. 버스 시스템의 이용 안정성은 전체적으로 버스 시스템이 주요 계절 변화에 관계없이 일년 내내 상대적으로 동일한 수준의 이용을 갖는다는 것을 의미합니다.

그럼에도 불구하고 2월과 9월에는 트래픽이 다소 감소하는 모습을 보이고 있습니다. 이 달의 월별 교통량은 175보다 약간 낮습니다. 이러한 부진은 해당 달의 특정 특성으로 설명될 수 있습니다. 예를 들어 2월은 아마도 여행에 적합하지 않은 겨울 달이거나 9월은 학교 또는 대중의 휴일 시즌의 달일 수 있습니다. 부문.

승객과 관련된 일반적인 월별 교통량에 관한 한, 자원 계획 및 관리에 대한 월별 교통 이메일의 일관성과 안정성은 약간의 불황을 제외하면 일년 내내 거의 동일한 패턴을 따라야 합니다. 2월과 9월에는 주로 문제를 조사하고 이해하기 위한 목적으로 관심을 기울일 필요가 있습니다.

5.7. 평균 교통량 기준 상위 5개 역

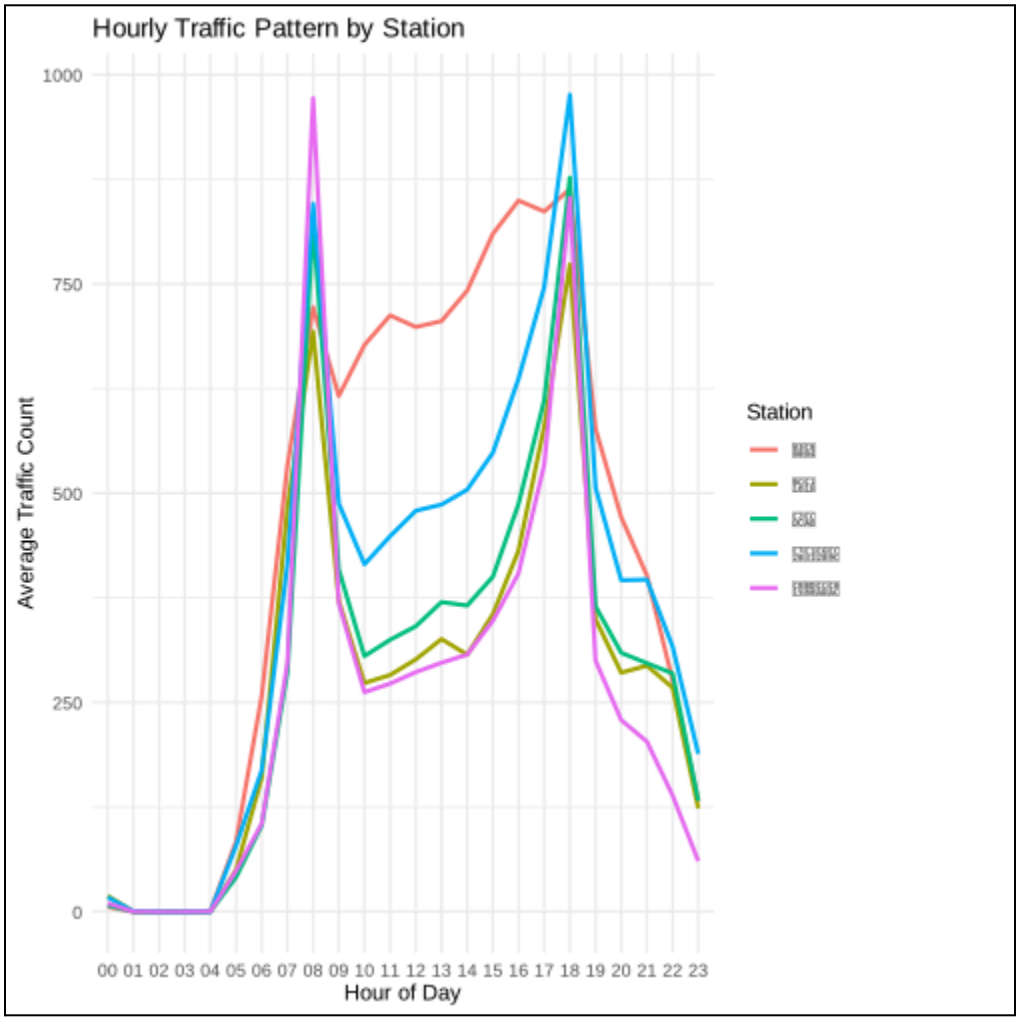


역별 평균 교통량 분석을 통해 가장 붐비는 상위 5개 위치를 식별하고 승객 활동이 가장 높은 주요 지역을 강조합니다. 이들 정거장은 운송 네트워크 내에서 중요한 노드이며 대중교통 수요의 중요한 지점을 나타냅니다. 평균 교통량 기준 상위 5개 역은 다음과 같습니다.

- 대전: 이 역은 457.15명의 승객이 탑승하여 가장 높은 평균 교통량을 보유하고 있습니다. 중앙 허브로서 이 역은 매일 많은 통근자를 수용할 가능성이 높습니다.
- 유성온천: 평균 통행량이 377.48회로 두 번째로 붐비는 역으로 주요 환승역으로서의 중요성을 시사합니다.
- 시청: 시청역(시청역)의 평균 통행량은 297.49건으로 행정 및 중심상업지역과의 연결 역할을 반영하고 있습니다.

- 반석: 이 역은 평균 280.30명의 승객을 기록하여 데이터 세트에서 네 번째로 붐비는 역으로 지정되었습니다.
- 정부청사: 마지막으로, 정부청사역은 평균 승객 262.39명으로 공무원과 시민 모두가 자주 이용하는 또 다른 교통량이 많은 역입니다.

5.8. 역별 시간별 교통 패턴



가장 붐비는 상위 5개 역(대전, 유성온천, 시청, 반석, 정부청사)의 시간별 교통 패턴은 특히 출퇴근 시간 동안 뚜렷한 피크와 흐름을 보여줍니다. 가장 붐비는 역인 대전역은 하루 종일 지속적으로 높은 교통량을 유지하며 일반적인 출퇴근 시간에 맞춰 오전 7~8시, 오후 6~7시쯤에 정체 현상이 뚜렷하게 나타납니다. 유성온천도 비슷한 패턴을 따르지만 피크 수가 약간 낮아 또 다른 주요 통근 허브로서의 역할을 나타냅니다. 시청은 업무 시간 중 중심 위치라는 점을 반영하듯 오전과 늦은 오후에 트래픽이 크게 증가합니다. 반석은 하루 종일

변동이 적고 보다 안정적인 흐름을 보이는 반면, 정부청사는 표준 근무 시간과 일치하여 오전과 늦은 오후에 정기적으로 피크를 경험합니다. 이러한 패턴은 통근 수요를 효과적으로 충족하기 위해 중요한 시간과 위치에서 교통 자원을 조정하는 것의 중요성을 강조합니다.

6. 결과

6.1. 결과 및 얻은 통찰력의 해석

광주광역시 버스교통행태 분석을 통해 몇 가지 통찰을 얻을 수 있었다. 우선, 2024년 1월부터 10월까지의 교통량을 보면 승객 수가 산발적으로 예외적으로 증가하는 등 대체로 일정하게 유지된 것으로 나타났습니다. 8월, 10월, 2월, 9월에는 계절별 여행 행태나 특정 활동으로 인해 교통량이 크게 감소했습니다. 교통의 시간 분포도 피크 시간대에 높은 버스 이용률을 강화했으며, 이는 오전 7시와 오후 6시를 중심으로 급격한 상승 추세를 보였습니다. 이러한 피크 버스 이용 기간은 주로 직장이나 학교에 가기 위해 낮 중 특정 시간 동안 집을 떠났다가 다른 정해진 시간에 집으로 돌아가는 대부분의 개인 활동의 순환적 특성에 기인할 수 있습니다. 그러므로 이 시기에는 대중교통 시스템의 이용률이 매우 높다는 분명한 증거가 있습니다.

탑승(승차) 대 하차(하차)를 분석한 결과, 두 활동 모두 피크 시간이 오전 8시와 오후 6시에 발생하며, 오전에 하차 횟수가, 저녁에 탑승 횟수가 약간 더 높은 것으로 나타났습니다. 이는 특정 지역이 아침에 도착 허브 역할을 하고 저녁에 출발 지점 역할을 한다는 것을 의미합니다. 가장 붐비는 상위 5개 역(대전, 유성온천, 시청, 반석, 정부청사)은 승객 수가 많은 중요한 대중교통 허브를 더욱 강조합니다. 평균 교통량이 가장 많은 역인 대전역은 하루 종일, 특히 출퇴근 시간에도 일관되게 이용되어 중앙 허브로서의 역할을 강조합니다. 이들 역의 시간별 교통 패턴은 일부 역은 꾸준한 이용을 유지하는 반면, 다른 역은 출퇴근 피크 시간에 맞추고 광주 대중교통망에서 역의 역할을 반영하여 변동하는 것으로 나타났습니다.

6.2. 제한 사항 및 가정

이 분석 중에 몇 가지 제한 사항과 가정이 발생했습니다. 첫째, 데이터는 2024년 1월부터 10월까지만 다루므로 누락된 달의 계절적 패턴이나 특이한 사건은 설명되지 않을 수 있습니다. 또한 실제 null 값이 없기 때문에 모든 누락된 값이 유효한 0이라고 가정합니다. 또 다른 한계는 교통 패턴에 큰 영향을 미칠 수 있는 날씨, 휴일, 특별 이벤트와 같은 외부 요인을 분석에서 고려하지 않았다는 것입니다. 마지막으로, 분석에서는 경로 변경이나 일정 조정과 같은 다른 요인도

이러한 수치에 영향을 미칠 수 있지만 교통량과 일반 대중 수요 사이의 직접적인 상관관계를 가정합니다.

7. 결론

이번 연구의 목적은 2024년 1월부터 10월까지 얻은 광주 버스 교통 데이터를 평가하여 다양한 시간, 위치, 모드에 따른 승객 수요 추세를 파악하는 것입니다. 특히 이른 아침과 저녁 피크시간대, 특히 오전 7시와 오후 6시에는 교통량이 증가한 것이 확연히 드러났습니다. 대전, 유성온천 등 주요 교통 허브는 서비스가 운영될 때마다 의도적으로 교통량이 증가해 광주의 대중교통망에서 이들 지역의 중요성을 입증했습니다. 또한, 정착지 탑승 패턴과 정착지 하차 패턴에서 눈에 띄는 차이는 임금 이주가 존재함을 시사하며 특정 역은 아침에 출입하고 저녁에 눈에 띄게 빈번하게 나타납니다.

요약하면, 이 연구는 통근을 위한 버스 서비스에 대한 광주 인구의 상당한 의존도를 강조하고 이러한 서비스의 피크 시간 동안 일정에 대한 잠재적인 조정을 제안합니다. 본 연구는 중요한 결과를 제시하지만 향후 연구는 대중 교통 이용에 영향을 미치는 외부 요인에 대한 이해를 높이기 위해 날씨 정보 또는 이벤트 관리 시스템과 같은 추가 데이터 소스를 통합하는 데 집중할 수 있습니다.

참고

- 공공데이터포털(data.go.kr). "사고 유형별 교통사고 통계." 검색 위치: https://data.go.kr
- 해들리 위컴, 가렛 그롤먼드. 데이터 과학을 위한 *R*. O'Reilly Media, 2016. 이 책은 R을 사용한 데이터 랭글링, 데이터 시각화 및 탐색적 데이터 분석에 대한 기본 개념을 제공합니다.
- 김현경 외. (2018). “날씨가 대한민국 서울의 대중교통 이용률에 미치는 영향.” 교통 지리 저널, 69, 95-108페이지. 이 기사에서는 대중 교통 패턴에 영향을 미치는 요인에 대해 논의하고 결과를 맥락화하는 데 유용할 수 있습니다.
- 광주시 교통부. (2023). 광주 대중교통 이용신고. Gwangju.go.kr에서 가져왔습니다. 이 보고서는 상황별 배경을 논의하는 데 관련된 교통 동향에 대한 자세한 통찰력을 제공합니다.
- 제임스, G. 등. (2013). *R* 애플리케이션을 이용한 통계 학습 소개. 통계의 Springer 텍스트. 교통 데이터를 해석하는 데 유용한 통찰력을 제공하는 통계 분석 및 모델링에 대한 포괄적인 가이드입니다.

충수

A. 데이터 소스

A.1. 데이터

https://www.data.go.kr/en/data/15060591/fileData.do#/

A.2. 데이터 테이블

A data.frame: 6 × 28

	날짜	역번호	역명	구분	X03.04시	X04.05시	X05.06시	X06.07시	X07.08시	X08.09시	...	X17.18시	X18.19시	X19.20시	X20.21시	X21.22시	X22.23시	X23.00시	X00.01시	X01.02시	X02.03시
	<chr>	<int>	<chr>	<chr>	<int>	<int>	<int>	<int>	<int>	<int>	...	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>	<int>
1	2024-01-01	1101	판암	승차	0	66	61	55	50	87	...	120	129	54	47	33	13	13	2	0	0
2	2024-01-01	1101	판암	하차	0	53	50	74	48	51	...	145	130	111	85	73	94	62	17	0	0
3	2024-01-01	1102	신흥	승차	0	0	18	106	21	40	...	41	37	30	15	16	8	3	0	0	0
4	2024-01-01	1102	신흥	하차	0	0	9	104	17	15	...	45	69	39	37	27	39	20	6	0	0
5	2024-01-01	1103	대동	승차	0	27	94	31	51	77	...	116	84	102	52	49	37	28	0	0	0
6	2024-01-01	1103	대동	하차	0	25	85	58	70	32	...	103	112	85	98	85	106	99	14	0	0

B. 소스 코드

```
# Data Preparation
## Packages load
# Load Packages
library(tidyverse)
library(ggplot2)
library(dplyr)
library(readr)
## Dataset Load
#Load Dataset
data <- read.csv("data.csv", fileEncoding= "EUC-KR")
head(data)
str(data)
colnames(data)
## Data Cleaning
# Standardize column names
colnames(data) <- gsub("시", "", colnames(data))
colnames(data) <- gsub("\\..*", "", colnames(data))
# Reshape data to long format
data_long <- data %>%
```

```
  pivot_longer(cols = starts_with("X"),
               names_to = "time_period",
               values_to = "traffic_count")
# Clean data suffix
data_long$time_period <- gsub("X", "", data_long$time_period)
data_long$time_period <- gsub("\\.", "-", data_long$time_period)
data_long$traffic_count <- as.numeric(data_long$traffic_count)
# Clean Data with NA Values
data_clean <- data_long %>%
  filter(!is.na(traffic_count))
colnames(data_clean)
str(data_clean)
## Handling Missing Values
# Confirm for NA Values
sum(is.na(data_clean))
summary(data_clean)
# Explarotary Data Analysis
## Traffic Count Over Time
data <- data_clean
# Making new group by feature
daily_traffic <- data %>%
  group_by(날짜) %>%
  summarise(total_traffic = sum(traffic_count, na.rm = TRUE))

# Convert 날짜 to Date format
daily_traffic$날짜 <- as.Date(daily_traffic$날짜)

# Data Visualization
ggplot(daily_traffic, aes(x = 날짜, y = total_traffic)) +
  geom_line(color = "blue") +
  labs(title = "Daily Traffic Count Over Time",
       x = "Date",
       y = "Total Traffic Count") +
  theme_minimal()
## Traffic Distribution by Time Period
```

```
# Group by time_period and average traffic count
time_period_traffic <- data %>%
  group_by(time_period) %>%
  summarise(avg_traffic = mean(traffic_count, na.rm = TRUE))

# Data Visualization
ggplot(time_period_traffic, aes(x = time_period, y = avg_traffic)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Average Traffic Count by Time Period",
       x = "Time Period (Hour)",
       y = "Average Traffic Count") +
  theme_minimal()
## 승차 vs. 하차 Comparasion
# Group by 구분 and sum the total traffic
boarding_alighting <- data %>%
  group_by(구분) %>%
  summarise(total_traffic = sum(traffic_count, na.rm = TRUE))
boarding_alighting

# Data Visualization
ggplot(boarding_alighting, aes(x = 구분, y = total_traffic, fill = 구분)) +
  geom_bar(stat = "identity") +
  labs(title = "Total Traffic Count: 승차 vs. 하차",
       x = "Type (구분)",
       y = "Total Traffic Count") +
  theme_minimal() +
  scale_fill_manual(values = c("skyblue", "salmon"))

## Top 5 Busiest Stations
# Group by 역명 and calculate the average traffic_count
busiest_stations <- data %>%
  group_by(역명) %>%
  summarise(avg_traffic = mean(traffic_count, na.rm = TRUE)) %>%
  arrange(desc(avg_traffic)) %>%
  slice(1:5)  # Select top 5
```

```
busiest_stations

# Plot the top 5 busiest stations
ggplot(busiest_stations, aes(x = reorder(역명, -avg_traffic), y =
avg_traffic)) +
  geom_bar(stat = "identity", fill = "lightgreen") +
  labs(title = "Top 5 Busiest Stations by Average Traffic Count",
       x = "Station (역명)",
       y = "Average Traffic Count") +
  theme_minimal()
## Average Traffic Count by Day of the Week
# Load necessary library for date manipulation
library(lubridate)

# Add a new column for the day of the week
data <- data %>%
  mutate(day_of_week = wday(날짜, label = TRUE, abbr = TRUE))

# Group by day_of_week and calculate the average traffic_count
weekly_traffic <- data %>%
  group_by(day_of_week) %>%
  summarise(avg_traffic = mean(traffic_count, na.rm = TRUE))

# Plot the average traffic by day of the week
ggplot(weekly_traffic, aes(x = day_of_week, y = avg_traffic)) +
  geom_bar(stat = "identity", fill = "orange") +
  labs(title = "Average Traffic Count by Day of the Week",
       x = "Day of the Week",
       y = "Average Traffic Count") +
  theme_minimal()
## Hourly Traffic Pattern by Station
selected_stations <- busiest_stations$역명[1:5]

# Filter data for selected stations and group by 역명 and time_period
hourly_station_traffic <- data %>%
```

```
  filter(역명 %in% selected_stations) %>%
  group_by(역명, time_period) %>%
  summarise(avg_traffic = mean(traffic_count, na.rm = TRUE))

# Plot the hourly traffic pattern for selected stations
ggplot(hourly_station_traffic, aes(x = time_period, y = avg_traffic,
color = 역명, group = 역명)) +
  geom_line(size = 1) +
  labs(title = "Hourly Traffic Pattern by Station",
       x = "Hour of Day",
       y = "Average Traffic Count",
       color = "Station") +
  theme_minimal()
## Monthly Traffic Trend
# Extract the month from 날짜 and add as a new column
data <- data %>%
  mutate(month = month(날짜, label = TRUE, abbr = TRUE))

# Group by month and calculate the average traffic count
monthly_traffic <- data %>%
  group_by(month) %>%
  summarise(avg_traffic = mean(traffic_count, na.rm = TRUE))

# Plot the average traffic count by month
ggplot(monthly_traffic, aes(x = month, y = avg_traffic)) +
  geom_bar(stat = "identity", fill = "purple") +
  labs(title = "Average Traffic Count by Month",
       x = "Month",
       y = "Average Traffic Count") +
  theme_minimal()
## Peak boarding vs Alighting
# Group by 구분 and time_period to calculate average traffic_count
peak_hours <- data %>%
  group_by(구분, time_period) %>%
  summarise(avg_traffic = mean(traffic_count, na.rm = TRUE))
```

```
peak_hours


# Plot boarding and alighting traffic count by time period
ggplot(peak_hours, aes(x = time_period, y = avg_traffic, fill = 구분)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Peak Boarding and Alighting Hours",
       x = "Time Period (Hour)",
       y = "Average Traffic Count") +
  theme_minimal() +
  scale_fill_manual(values = c("승차" = "skyblue", "하차" = "salmon"))
## Traffic Pattern on Weekdays vs Weekends
# Add a new column for weekday vs weekend
data <- data %>%
  mutate(day_type = ifelse(wday(날짜) %in% c(1, 7), "Weekend",
"Weekday"))

# Group by day_type and calculate the average traffic count
day_type_traffic <- data %>%
  group_by(day_type) %>%
  summarise(avg_traffic = mean(traffic_count, na.rm = TRUE))

# Plot the average traffic for weekdays vs weekends
ggplot(day_type_traffic, aes(x = day_type, y = avg_traffic, fill =
day_type)) +
  geom_bar(stat = "identity") +
  labs(title = "Average Traffic Count: Weekdays vs Weekends",
       x = "Day Type",
       y = "Average Traffic Count") +
  theme_minimal() +
  scale_fill_manual(values = c("Weekday" = "lightblue", "Weekend" =
"orange"))
```