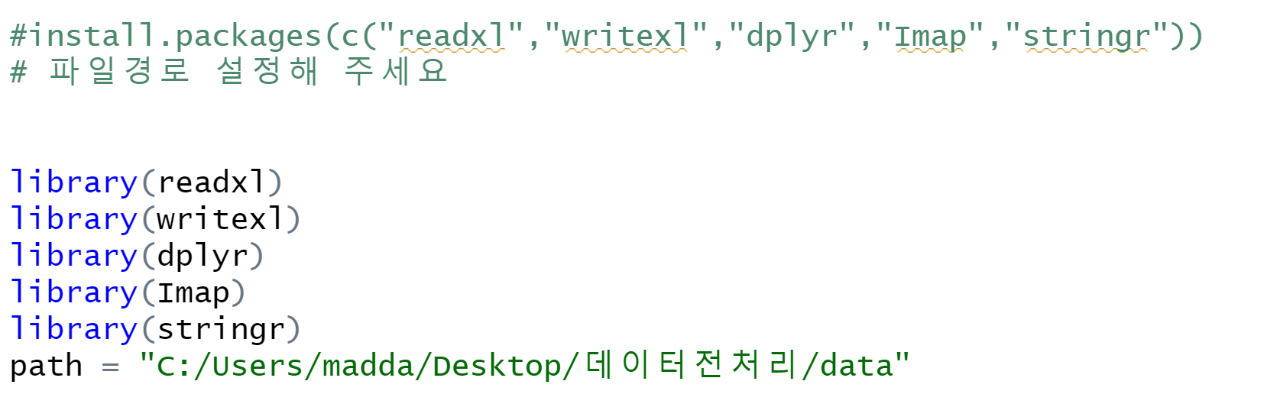
데이터전처리 설명 파일입니다.

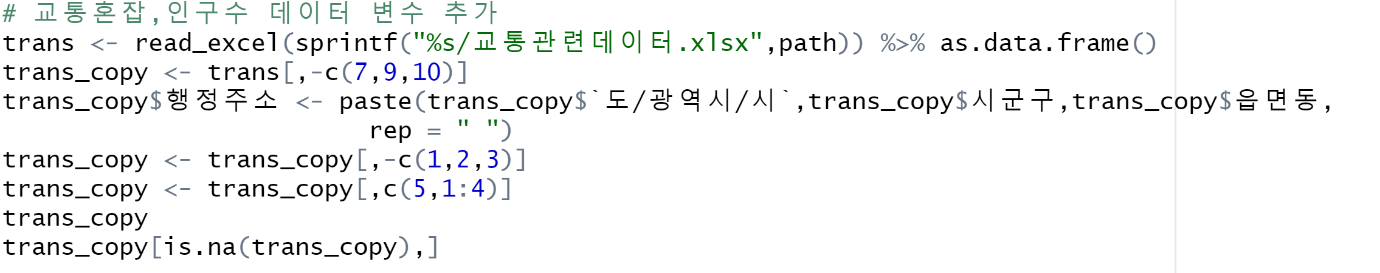
**Working Directory는 path에 입력합니다.**

**Path는 다음처럼 입력해줍니다.(반드시 데이터전처리/data로 끝나야합니다.**



맨 첫줄에서 주석표시를 없애고 패키지를 다운로드합니다. 다운로드 완료시 library()를 합니다.

**1. 교통관련데이터 불러오기**



“trans” 명으로 “교통관련데이터” xlsx 파일을 읽고 데이터프레임 형식으로 바꿉니다.

“trans” 데이터셋 일부

테이블이(가) 표시된 사진

자동 생성된 설명

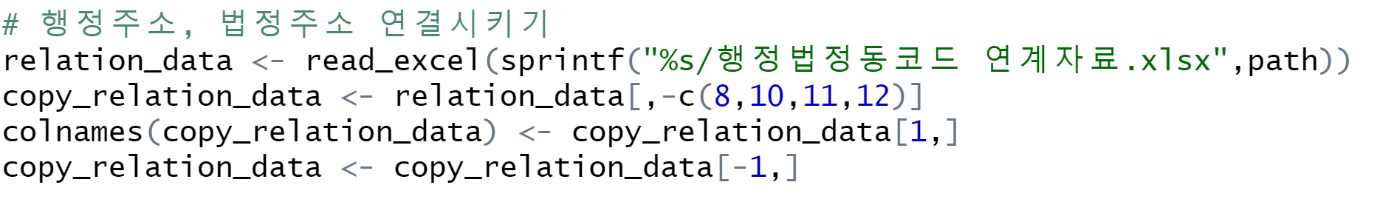
그 후 불필요한 데이터는 제거하여 “trans\_copy”에 저장합니다. Paste 함수를 써서 “도/광역시/시” + “시군구” + “읍면동” 데이터를 합쳐서 “행정주소”를 만듭니다. 그 뒤 “도/광역시/시” + “시군구” + “읍면동” 데이터를 지우고, 변수 순서를 재배치하여 “trans\_copy”에 저장하고 결측값을 확인합니다.

“trans\_copy” 데이터셋 일부

테이블이(가) 표시된 사진

자동 생성된 설명

**2. “행정법정동코드 연계자료” 데이터 읽기**



“relation\_data” 명으로 “행정법정동코드 연계자료”를 읽어 저장합니다.

“relation\_data” 일부

테이블이(가) 표시된 사진

자동 생성된 설명

데이터셋을 합치기 위해서 데이터를 가공해야 합니다.

8,10,11,12행은 사용하지 않을 데이터이므로 제거하고, 1열은 열 제목으로 설정합니다. 2열은 제거합니다.

**3. 읍면동 단위 남기고 데이터 제거 및 결측값 수정**

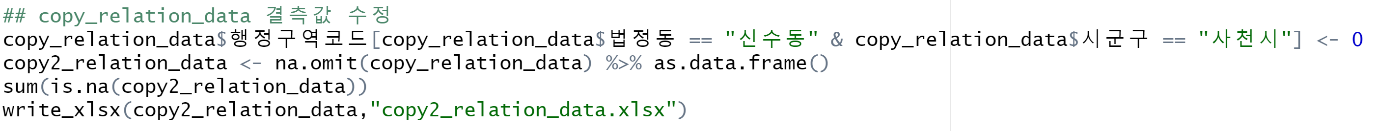
텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

위에서 했던 작업을 다시 해줍니다. “relation\_data” 데이터셋은 “리” 단위까지 나와있어서 “읍면동” 단위 이외의 것들을 추출한 뒤 제거하여 줍니다(“도”,”시”,”군”,”리”,”구” 해당). “grep”함수를 이용하여 끝자리가 “읍면동”으로 끝나지 않는 것들을 부분을 추출한 뒤 value = 1로 설정하여 문자열을 추출하여 %in%를 사용하여 제거하여 줍니다.



그 후 결측값을 찾아서 값을 보고 검색하여 수정해줍니다.

결측값이 있는지 확인한 뒤 제거하고 “copy2\_relation\_data”에 저장합니다.

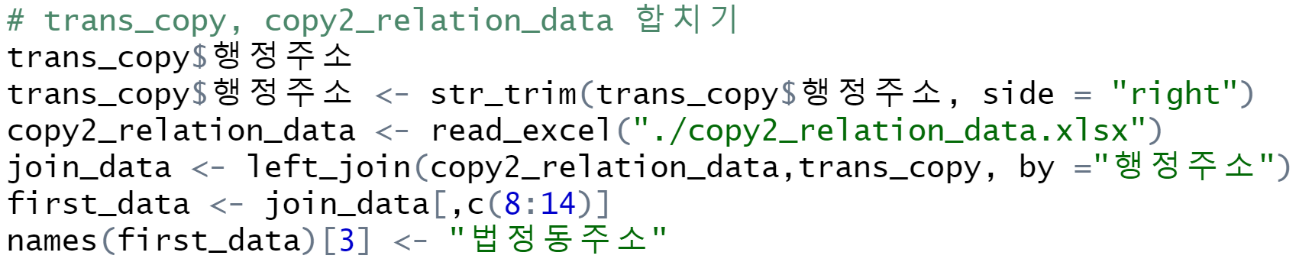
그 뒤 “copy2\_relation\_data.xlsx”로 파일을 저장합니다.

“copy2\_relation\_data” 일부

테이블이(가) 표시된 사진

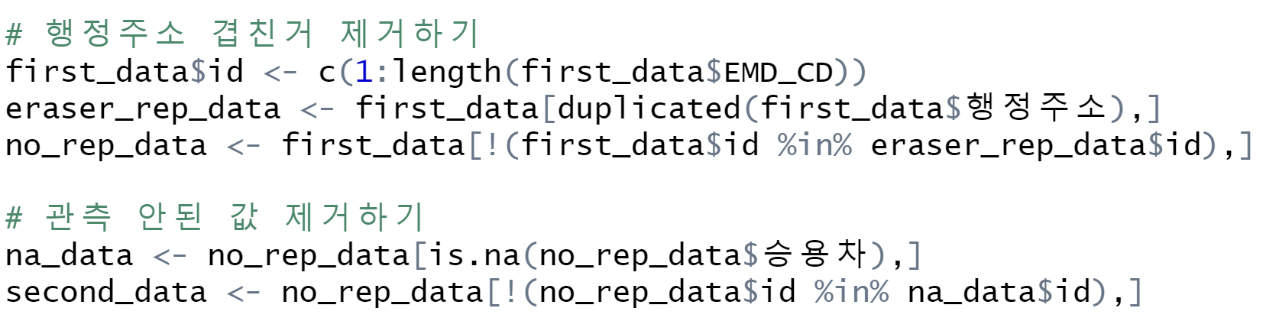
자동 생성된 설명

**4. “trans\_copy”, “copy2\_relation\_data” 합치기**



“trans\_copy$행정주소”를 할 시 오른쪽에 공백이 생긴다. 따라서 str\_trim함수를 이용해서 오른쪽 공백을 없애준다. 또한 제대로 left\_join이 되지 않아서 “copy2\_relation\_data”를 다시 한번 불러와서 “trans\_copy”와 “행정주소”를 기준으로 left\_join을 해주고 “join\_data”에 저장합니다. 이 후 필요한 칼럼들만 뽑아서 “first\_data”에 저장하고 3번째 칼럼의 이름을 “법정동주소”로 바꿉니다.

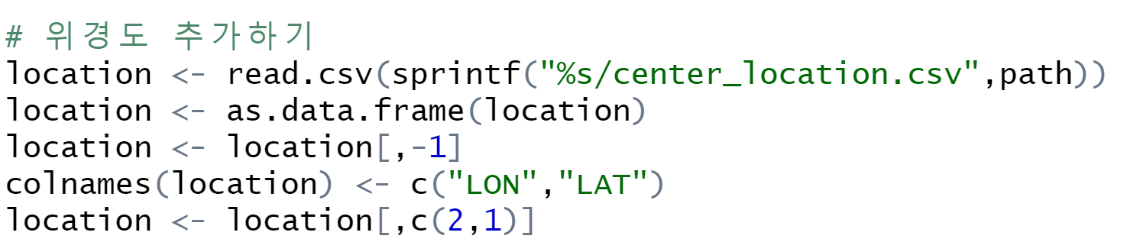
**5. 중복된 값, 결측값 처리하기**



“first\_data”를 보면 “행정주소”가 중복되어있다. 따라서 1개의 값만 남기고 중복된 값은 제거한다. “eraser\_rep\_data”에 “행정주소”가 중복인 데이터를 duplicated()를 이용해서 저장한다. “no\_rep\_data”에 %in%를 사용하여 first\_data$id에 포함되지 않는 row값들을 저장합니다.

“na\_data”에 관측이 안된 row값들을 저장합니다. “second\_data”에 %in%을 이용하여 na\_data$id가 no\_rep\_data$id에 포함이 되어 있지 않은 row값들을 저장합니다. 그 뒤 “2차가공데이터” 이름으로 엑셀파일로 저장합니다.

**6. 각 읍면동 위도/경도 데이터 추가하기**

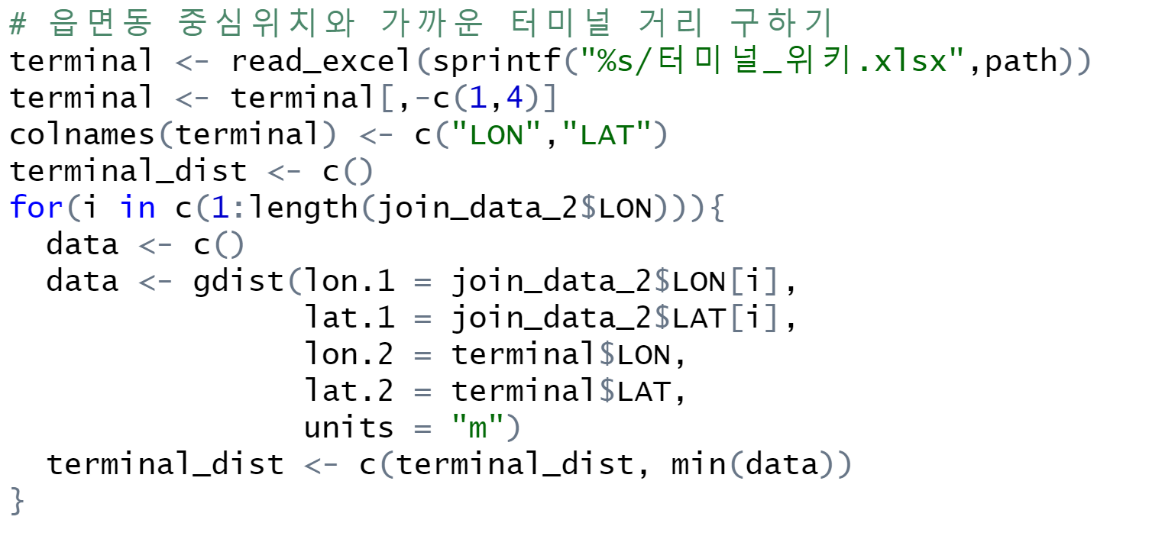


“location” 이름으로 “center\_location.csv”파일을 저장합니다. Location의 데이터들을 각 읍면동 위/경도 데이터로 경도, 위도에 해당되는 칼럼들의 이름을 각각 “LON”,”LAT”로 바꾸고 필요없는 칼럼또한 제거한 뒤, 위/경도 순서로 칼럼위치를 바꿉니다. 그 뒤 “join\_data\_2” 명으로 second\_data와 location 데이터셋을 1개의 데이터프레임으로 만들어줍니다. 그 뒤 결측값 개수를 확인합니다.

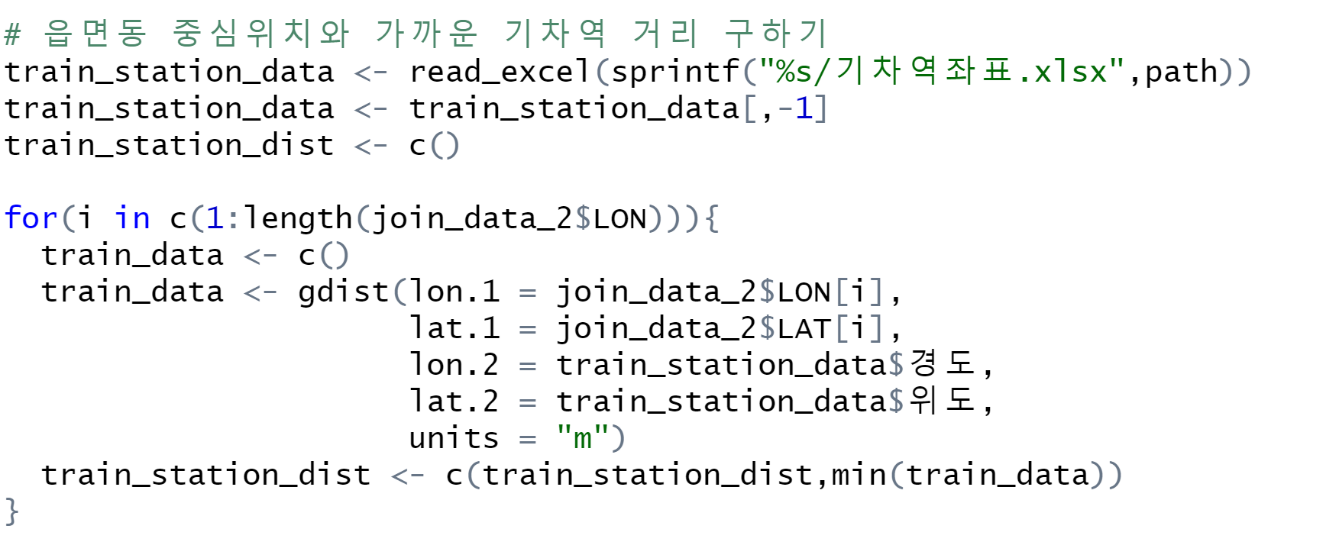
“join\_data\_2” 일부



**7. 각 읍면동 위/경도를 통해서 터미널/기차역/공항과의 거리 구하기**



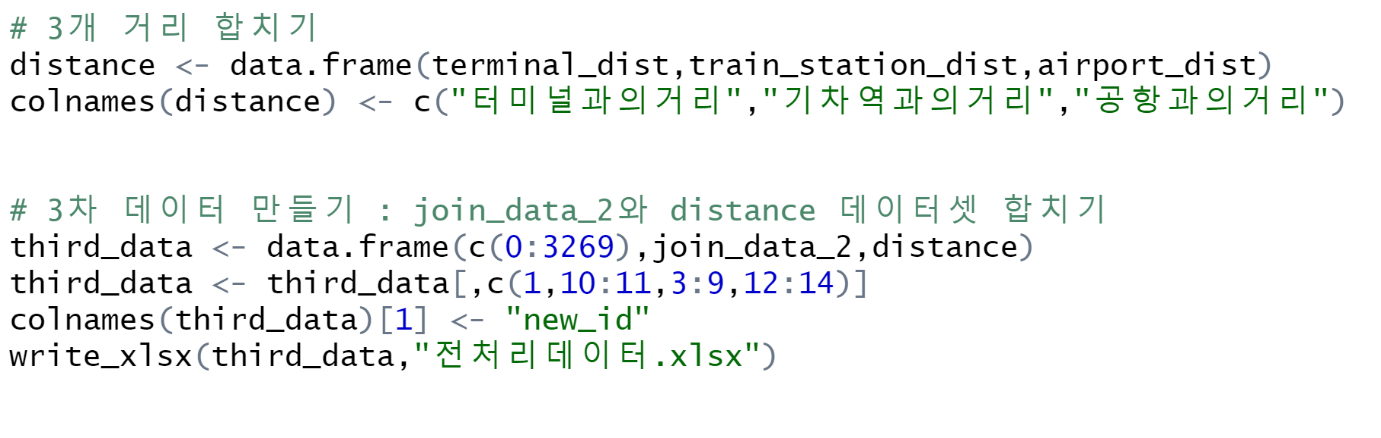
“terminal”에 터미널 위/경도가 적혀있는 데이터를 불러와 저장합니다. 필요없는 칼럼들은 제거해주고, 이름을 위/경도 데이터에 맞춰 “LAT”,”LON”으로 변경합니다. 그 후 for문을 이용하여 각 읍면동별 위/경도를 뽑아서 터미널 위/경도와의 거리를 계산합니다. 이때 사용하는 거리공식은 Imap 패키지의 gdist를 이용하였습니다. 그 뒤 terminal\_dist 이름으로 각 터미널까지와의 거리중 최소값을 저장해서 모든 읍면동의 거리 데이터를 저장합니다.





터미널과 같은 방법으로 기차역/공항과의 최소거리를 구하여 각각 train\_station\_dist, airport\_dist에 저장합니다. airport에서는 칼럼의 이름만을 달리하여 같은 방법으로 거리를 계산합니다.

8. join\_data\_2와 distance 데이터셋 합치기



“distance” 명으로 위에서 구한 terminal\_dist, train\_station\_dist, airport\_dist를 데이터프레임 형태로 저장합니다. 그 뒤 이름을 코드와 같이 바꿉니다.

“third\_data” 명으로 순서를 의미하는 데이터와 join\_data\_2, distance데이터를 데이터프레임 형태로 “third\_data”에 저장합니다. 그 뒤 필요한 칼럼들만 다시 저장하고 첫번째 칼럼 이름을 “new\_id”로 바꿔준 뒤, “가공데이터” 이름으로 third\_data를 엑셀파일로 저장합니다.