**우리 “시각화” 팀은!?**

어떤 데이터들을 보기 좋게 표현하여 사람에게 정보를 쉽고, 정확하게 전달하는 것이 목표예요.

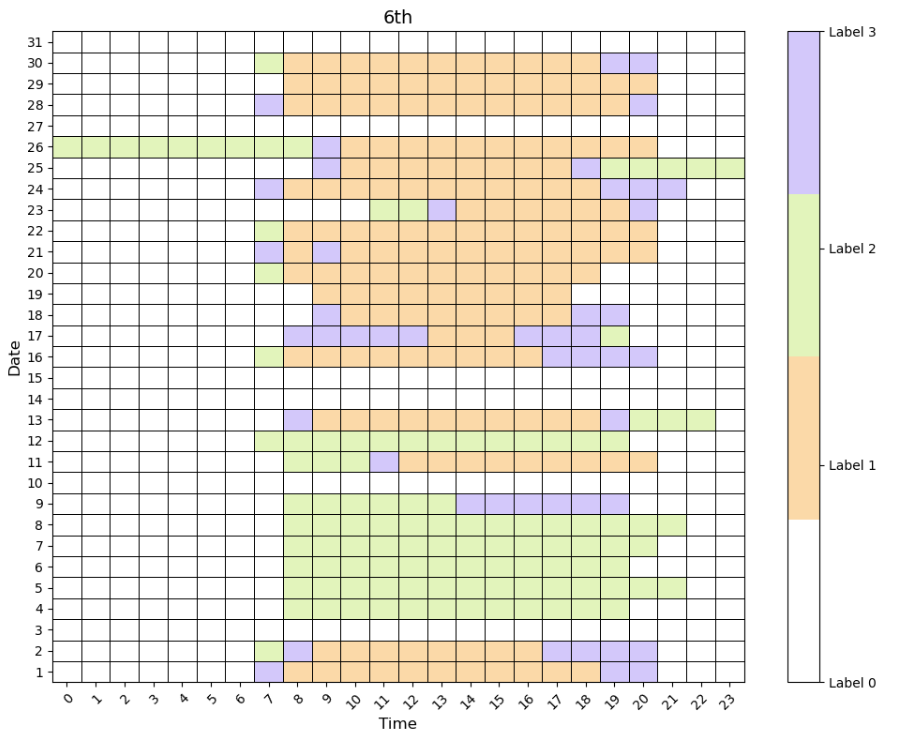
우리의 데이터 표현 방법, 협업 방식에 대해 알아보아요.

우리 팀은 “파스텔 색”을 사용하는 것이 원칙이에요.

“파스텔 색”은 톤이 연하고, 채도가 낮기 때문에 밝고, 부드러운 효과를 낼 수 있어요.

“원색”은 톤이 진하고, 채도가 높기 때문에 뚜렷하고, 강렬한 효과를 낼 수 있어서 정보 전달을 할 때 “원색”을 이용하는 것이 뚜렷하다고 생각할 수 있어요. 하지만 아래 그림을 보면 알 수 있는데, 뚜렷하고 강렬하다 해서 정보 전달이 잘 되는 것은 절대 아닙니다.

오히려 데이터를 여러 가지로 라벨링해야 한다면 부드럽게 이어지는 “파스텔 색”을 이용하는 것이 시각적으로 더 좋은 효과를 가져올 수 있습니다.

도표, 스크린샷, 라인, 다채로움이(가) 표시된 사진

자동 생성된 설명다만 “파스텔 색”은 비슷한 색을 이용할 경우 분류의 효과가 너무 떨어질 수도 있으니 주의하세요.

**코드 스타일 가이드**

1. 들여쓰기와 공백:

- 들여쓰기: 4개의 스페이스 사용

- 연산자 주변에 공백 추가: x = y + z

2. 변수와 함수명:

- 의미 전달: user\_name, calculate\_total\_amount()

- 카멜 케이스 또는 밑줄 사용

3. 주석:

- 코드 의도와 기능 설명

- 복잡한 부분이나 중요한 정보 주석으로 기록

4. 함수와 클래스:

- 한 가지 기능만 수행하는 함수

- 클래스는 한 가지 책임 가지기

5. 파일 구조와 모듈:

- 관련 기능 묶어 파일/모듈 구성

- 파일명은 소문자, 필요 시 밑줄 사용

6. 문자열 리터럴:

- 작은 따옴표(') 또는 큰 따옴표(") 사용

- 같은 프로젝트 내에서 동일한 스타일 유지

7. 주요 프로그래밍 언어 관례:

- Python: PEP 8 스타일 가이드 준수

- JavaScript: ESLint 규칙 따르기

8. 예외 처리:

- 각 예외 상황에 맞는 에러 메시지 제공

- 필요한 곳에서 예외 처리 사용

팀 내에서 이러한 가이드라인을 공유하고 따르면서 코드 품질을 유지하고 협업을 원활하게 진행하세요. 주기적으로 가이드라인을 검토하며 프로젝트의 요구사항과 팀의 선호도에 맞게 조정할 수 있습니다.