

DDWU\_Bigdata 스터디 2번째 :

## 카카오톡 대화분석

## Contents

1

### 환경 설정하기

- 카카오톡 대화 내보내기
- colab 에 업로드

2

### 카카오톡 대화 정제

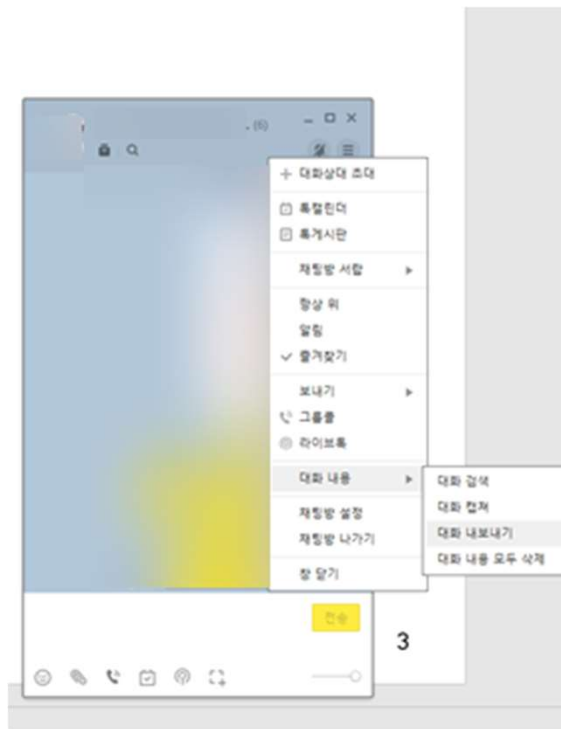
- txt -> csv
- Csv 정제해서 pk 형태로 저장

3

### 카카오톡 사용자별/연도별 빈도분석

- 워드클라우드
- 가장 빈도가 높은 단어 라인그래프

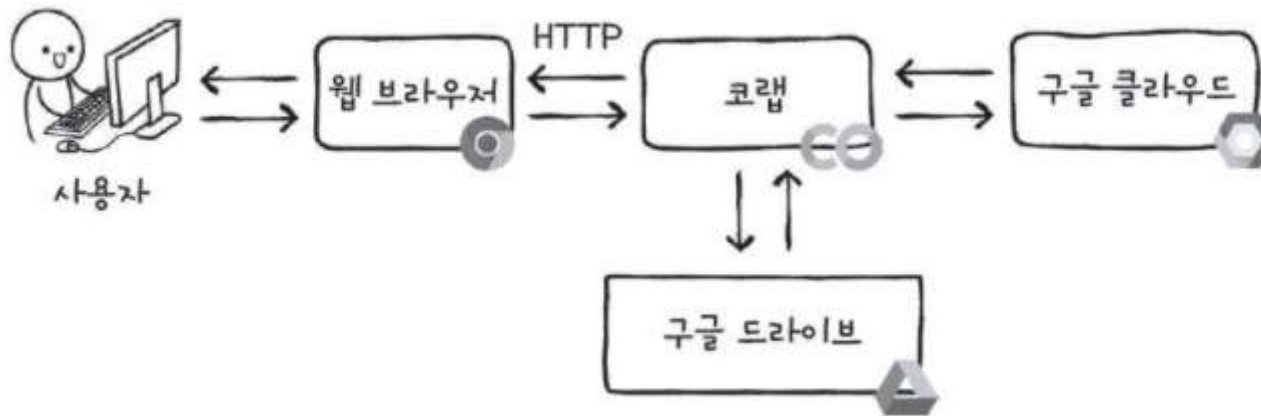
## 카카오톡 대화 텍스트 다운로드



- 카카오톡 단톡방에서  
오른쪽 위 메뉴 -[대화 내용]-[대화 내보내기]-[저장]

## Colab 소개

- 구글에서 교육과 과학 연구를 목적으로 개발한 도구로, 2017년에 무료로 공개
- 파이썬 코드를 실행하거나 텍스트를 작성할 수도 있고 그래프를 그릴 수 있다.
- 웹 브라우저를 통해 제어하고 실제 파이썬 코드 실행은 구글 클라우드의 가상 서버에서 이루어집니다.
- 코랩에서 만든 파일인 노트북은 구글 드라이브에 저장하고 불러올 수 있습니다.

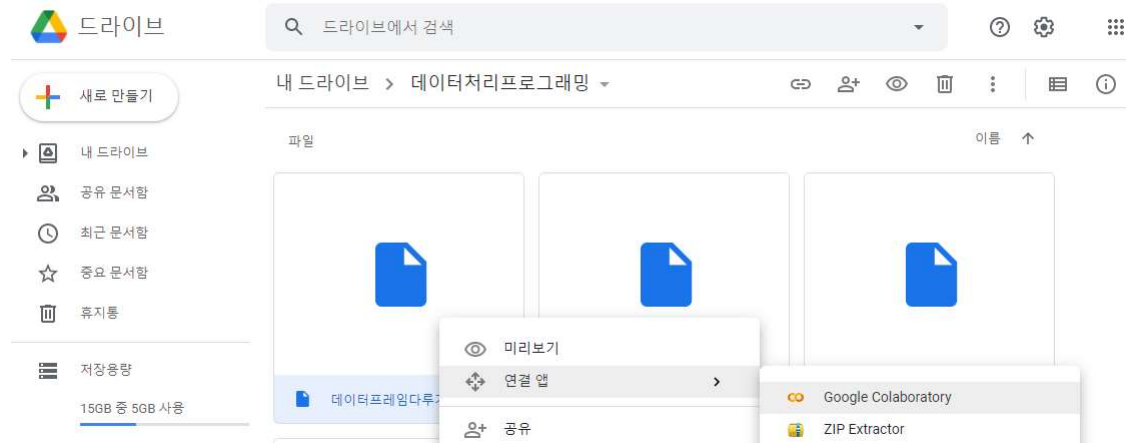


# 구글 Colab 환경

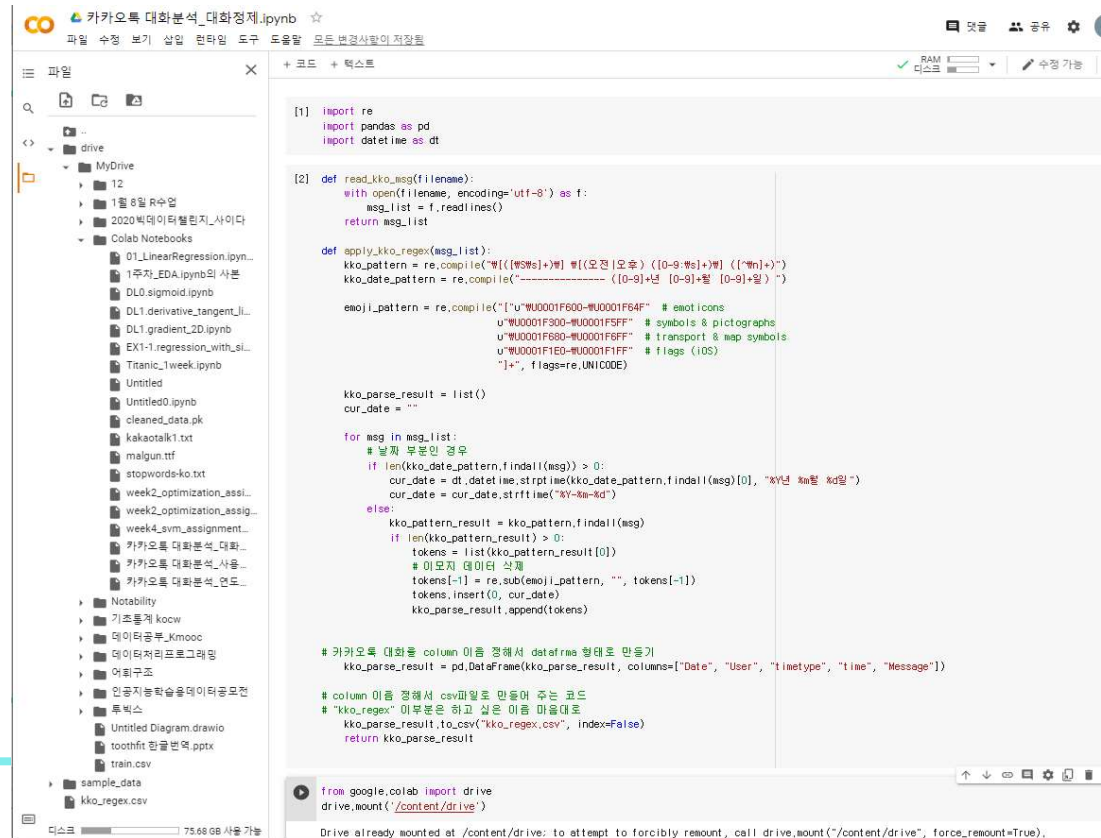
## 1. 구글 드라이브: 보낸 파일 모두 업로드

malgun	2021-01-31 오후 4:27	트루타입 글꼴 파일	9,371KB
stopwords-ko	2021-01-30 오후 2:15	텍스트 문서	7KB
PC 카카오톡 대화분석_대화정제	2021-01-31 오후 4:28	IPYNB 파일	53KB
PC 카카오톡 대화분석_사용자별빈도분석	2021-01-31 오후 4:28	IPYNB 파일	678KB
PC 카카오톡 대화분석_연도별빈도분석	2021-01-31 오후 4:28	IPYNB 파일	310KB

## 2. ipynb 파일 : 구글 코랩으로 연결하기



# 구글 Colab 환경



```
[1] import re
import pandas as pd
import datetime as dt

[2] def read_kko_msg(filename):
    with open(filename, encoding='utf-8') as f:
        msg_list = f.readlines()
    return msg_list

def apply_kko_regex(msg_list):
    kko_pattern = re.compile("([WSws]+)#[([오전|오후]) ([0-9:WSws]+)#[([~\n]+)~]
    kko_date_pattern = re.compile("----- ([0-9]+년 [0-9]+월 [0-9]+일) ~)

    emoji_pattern = re.compile("([u"u0000F600-u0000F64F" # emoticons
    u"u0000F300-u0000F3FF" # symbols & pictographs
    u"u0000F680-u0000F6FF" # transport & map symbols
    u"u0000F1E0-u0000F1FF" # flags (iOS)
    ])+", flags=re.UNICODE)

    kko_parse_result = list()
    cur_date = ""

    for msg in msg_list:
        # 날짜 부분만 경우
        if len(kko_date_pattern.findall(msg)) > 0:
            cur_date = dt.datetime.strptime(kko_date_pattern.findall(msg)[0], "%Y년 %m월 %d일 ")
            cur_date = cur_date.strftime("%Y-%m-%d")
        else:
            kko_pattern_result = kko_pattern.findall(msg)
            if len(kko_pattern_result) > 0:
                tokens = list(kko_pattern_result[0])
                # 이모지 데이터 삭제
                tokens[-1] = re.sub(emoji_pattern, "", tokens[-1])
                tokens.insert(0, cur_date)
                kko_parse_result.append(tokens)

# 카카오톡 대화록 column 이름 정해서 dataframe 형태로 만들기
kko_parse_result = pd.DataFrame(kko_parse_result, columns=["Date", "User", "timetype", "time", "Message"])

# column 이름 정해서 csv파일로 만들어 주는 코드
# "kko_regex" 이부분은 하고 싶은 이름 마음대로
kko_parse_result.to_csv("kko_regex.csv", index=False)
return kko_parse_result

from google.colab import drive
drive.mount('/content/drive')

Drive already mounted at /content/drive: to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```