# Základní informace

## Původní myšlenka

Nápad na tento projekt vznikl když jsem dostal chuť zkusit vytvořit něco s databázemi. V té chvíli jsem ještě nevěděl co přesně, ale první otázka ovšem byla, kde nějakou databázi získat. Přirozeně se moje myšlenky okamžitě přesunuly směrem k technologickým gigantům dnešního světa, kterým jsou databáze všeho druhu jistě důvěrně známé a odtud už nebylo daleko k Facebooku samotnému a využití možnosti, kterou nám pro změnu poskytlo nedávno zavedené GDPR – stažení svých dat.

Během víkendu byla první funkční verze programu hotová. Ta byla za nějakou dobu nahrazena druhou, a delší pauze konečně třetí, ke které se váže tato dokumentace.

## Popis

Aplikace se skládá ze dvou částí: 2

* Knihovna *fb\_disassemble.py –* Zajišťuje načtení dat Facebook schránky do interního datového formátu.
  + Jak už jméno napovídá – rozebere strukturu složek Facebooku a načte je, zároveň provede základní analýzu dat (najde časy nejstarších a nejnovějších zpráv, celkový počet zpráv, rozřadí konverzace podle typu a zaznamená účastníky konverzací)
  + Implementuje základní funkce pro práci s načtenými daty – výběr určitých konverzací, požadavek na vybrané konverzace a získání statistik (nebo debug informací) v čitelné formě.
* *analyze.py*
  + Funkce zprostředkující základní textové rozhraní (zadaní parametrů, jmen souborů, ...)
  + Ukládání výstupových souborů
  + Samotné algoritmy, které pracují s načtenými daty
  + Možnost anonymního režimu

# Uživatelské rozhraní

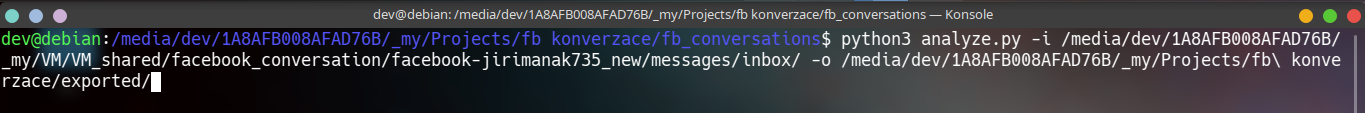
Aplikace nemá grafické uživatelské rozhraní, ovládá se přes konzoli. Jak jsem již zmínil v úvodu, aplikace nebyla zamýšlena jako nástroj pro analýzu samotnou, nýbrž pouze jako jakýsi prostředník mezi nezpracovanými daty a například aplikací Excel, která je uzpůsobena pro zobrazování dat toho formátu.

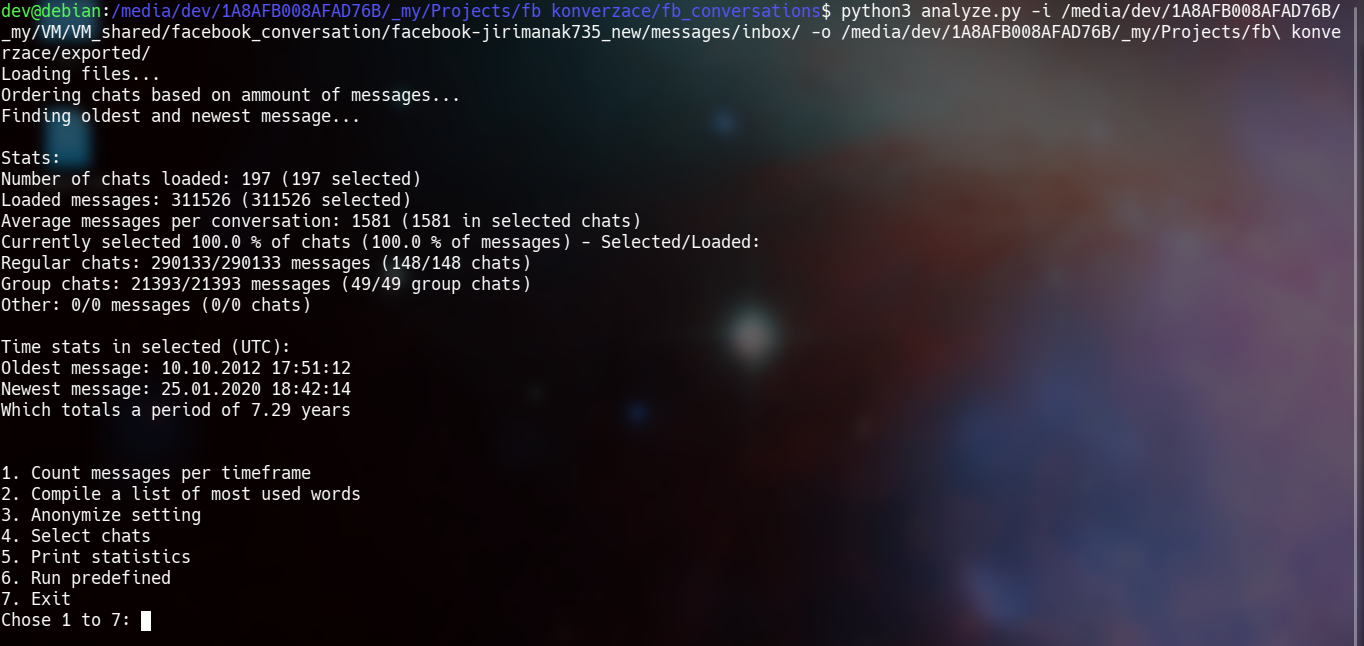
I tak je potřeba alespoň základní interakce s programem. Jediný parametr, který se definuje předem, je cesta ke složce *inbox*, která obsahuje Facebook schránku zpráv všech kontaktů uživatele (popřípadě cesta k výstupní složce). Za chodu umožňuje program změnit jména výstupních souborů, volbu kontaktů a režimů (více v ukázkách).

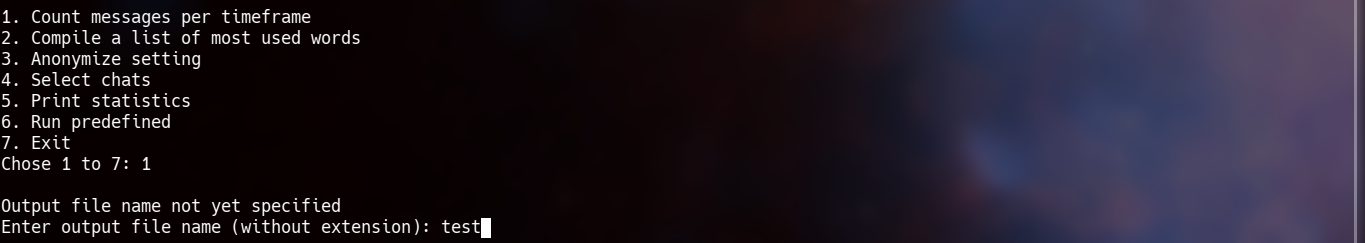
Přidat grafické prostředí je samozřejmě možné, celá architektura programu je docela modulární bez vzájemně provázaných proměnných – také důvod pro vytvoření samostatné knihovny. Při vývoji bylo dbáno na to, aby byla zachována hierarchie jakéhosi stromového uspořádání, takže nadstavby jsou možné a objevují se i ve stávající aplikaci (například v předdefinovaném režimu a dialozích).

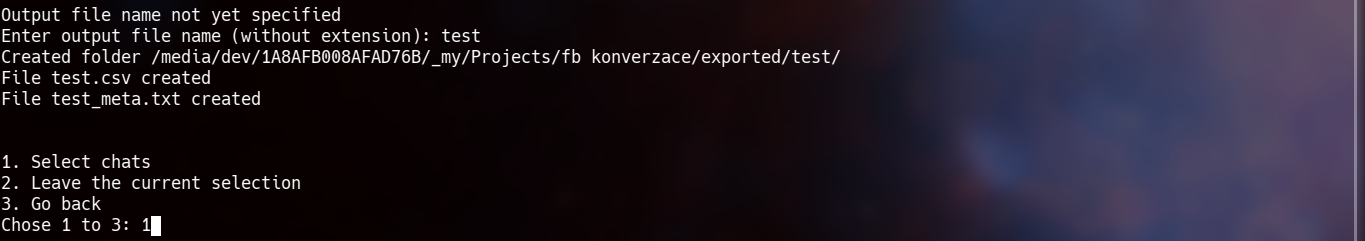
# Ukázky

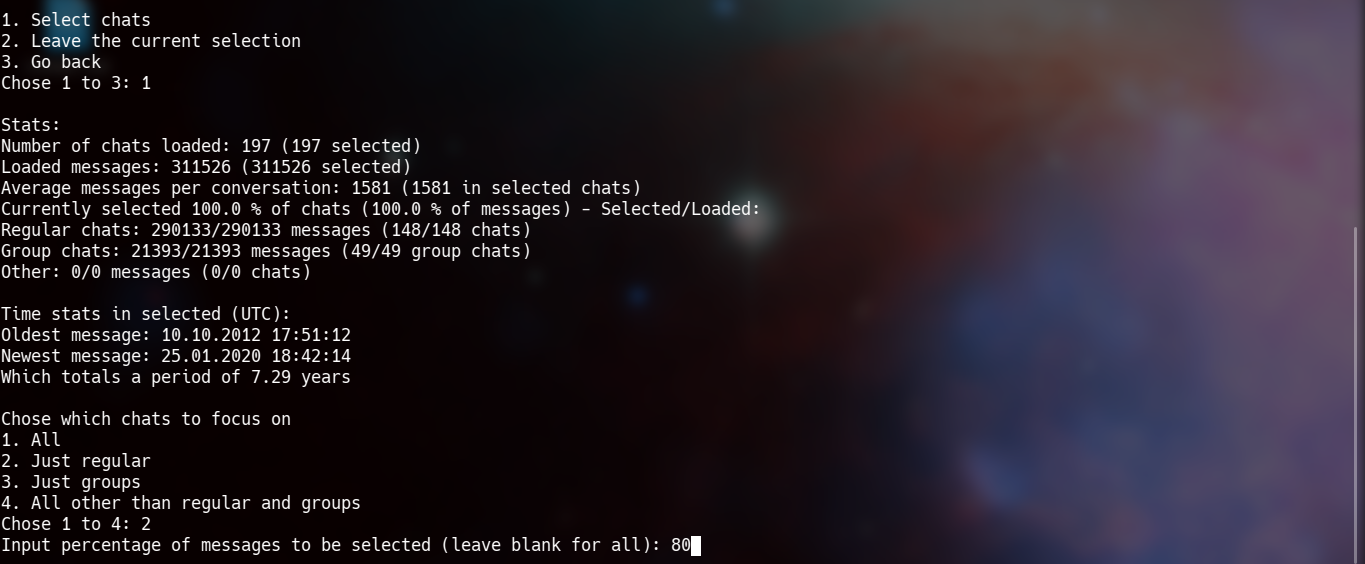
## Export frekvence zpráv pro tabulkový procesor

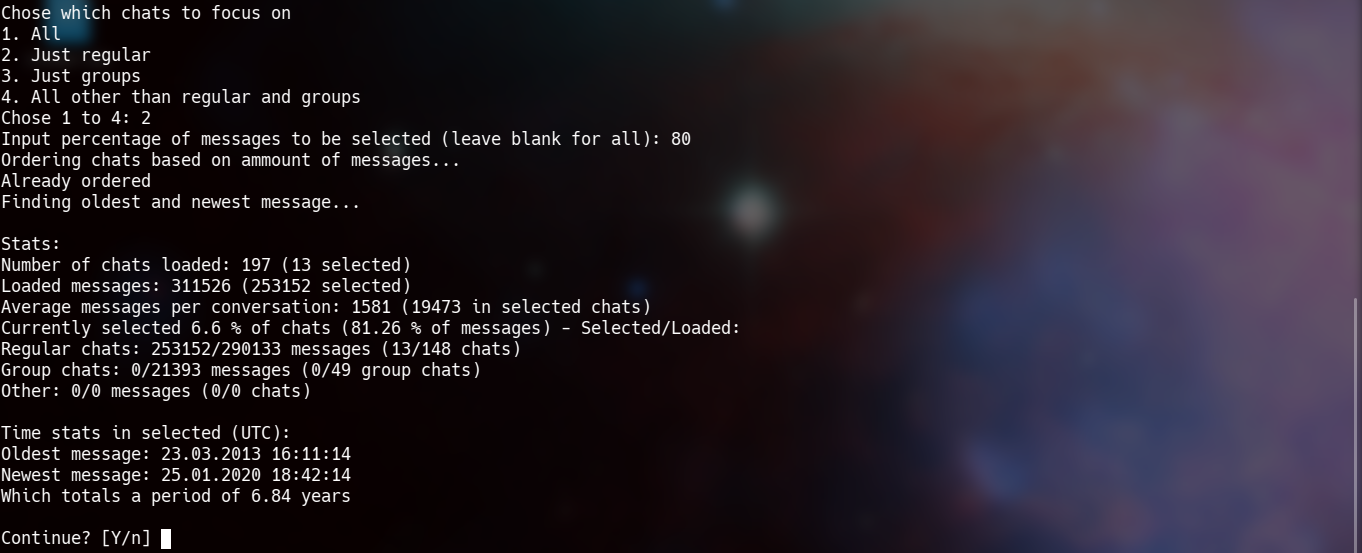
Spuštění *analyze.py* (volba -o není vyžadována, cesta by byla automaticky nastavena na složku *facebook-jirimanak735\_new).*

Po načtení jsou zobrazeny základní údaje (celkový počet načtených konverzací, počet zpráv, kolik z nich je ve skupinách a podobně). Ze začátku jsou vybrány všechny konverzace.

Po vybrání požadovaného režimu je potřeba zadat název exportovaného souboru. (pokud byl již někdy zadán, aplikace se zeptá jestli ho ponechat nebo změnit)

Další krok je nastavení filtrování konverzací.

Aplikace znovu zobrazí statistiky pro kontrolu výběru.

Velké množství statisticky sledovaných jevů následuje takzvaný Paretův princip a konverzace na Facebooku nejsou vyjímkou. Konverzace jsou vnitřně seřazeny podle počtu zpráv, takže volba 80 % zpráv znamená výběr tolika uživatelů z nejvyšších příček tak, aby se dosáhlo žádaného poměru.

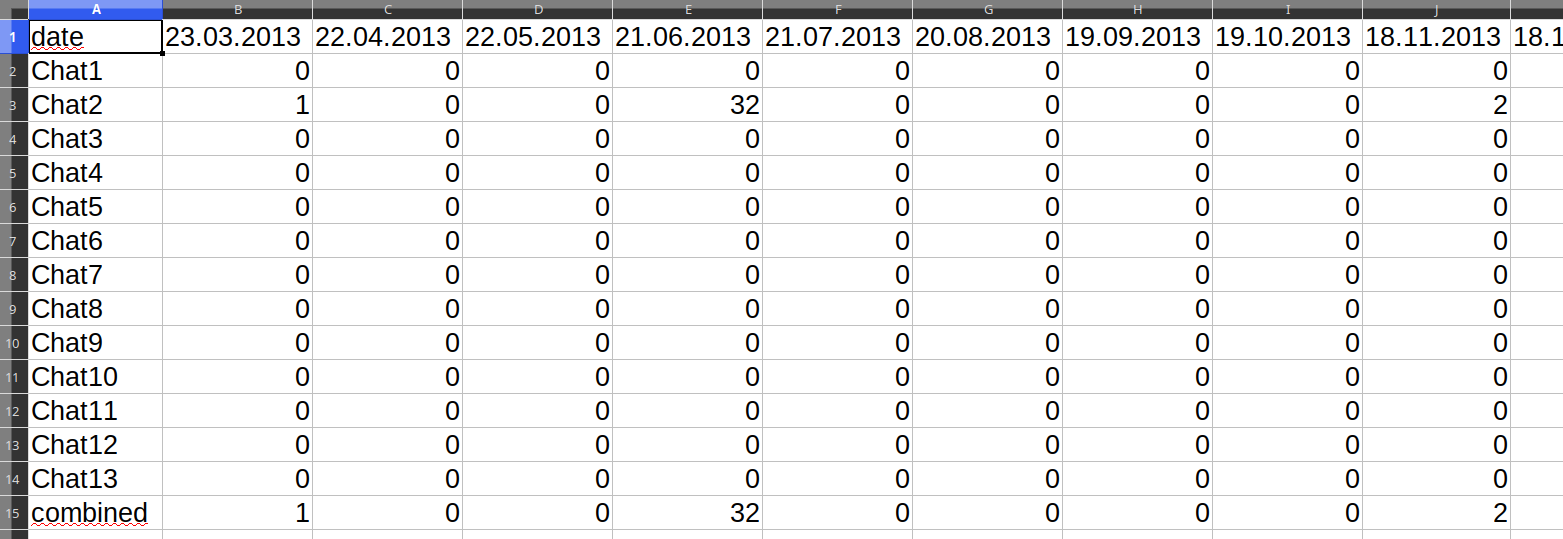
Tento krok je možné opakovat dokola dokud nepotvrdíte, že chcete pokračovat. V případě tohoto vzorku stačí zprávy prvních 13 konverzací s nejvíce zprávami k naplnění požadavku alespoň 80 % z celkového počtu zpráv. (Důležitý je řádek *Currently selected* a další tři pod ním)

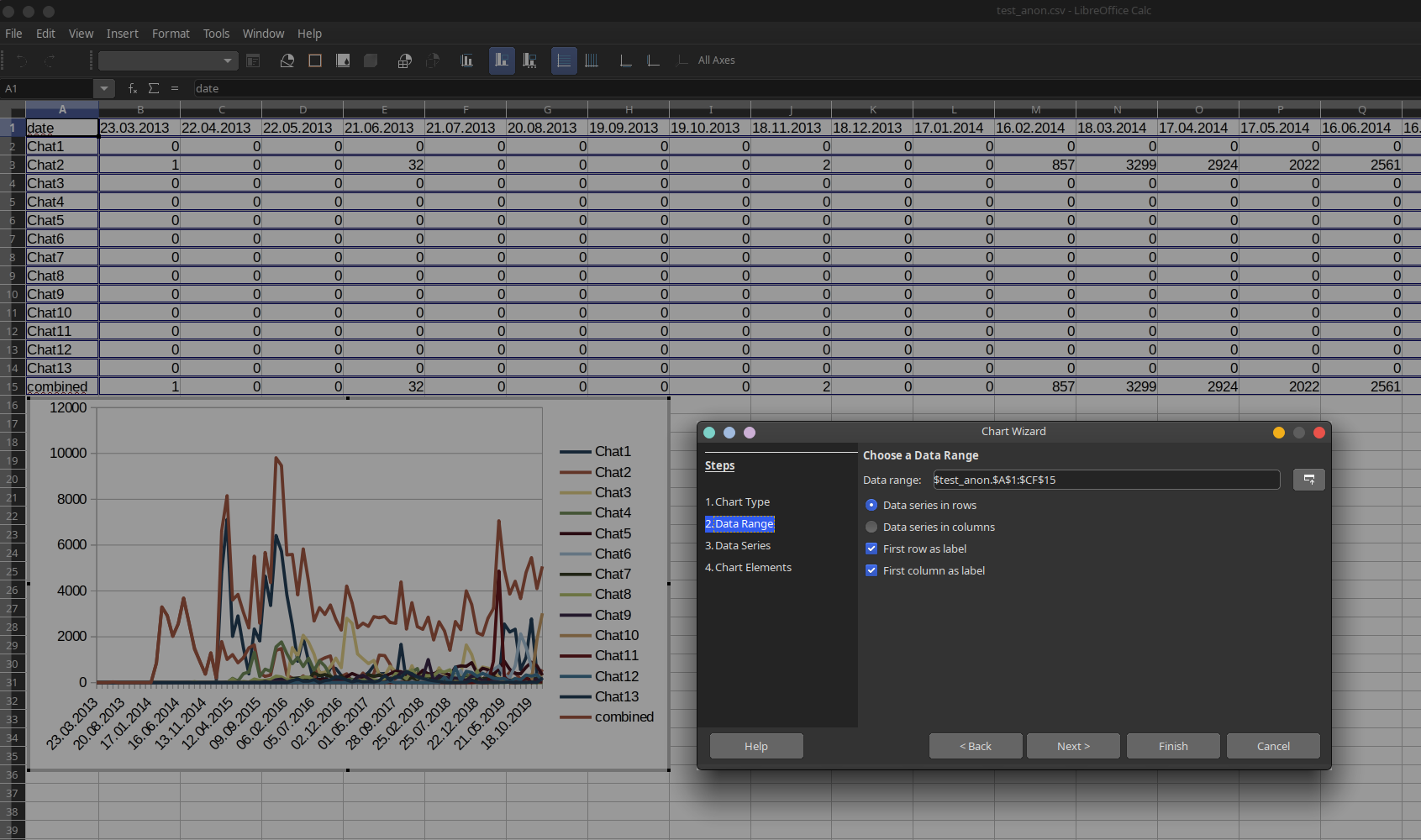
Po potvrzení výběru je potřeba zadat počet dní pro jeden časový úsek, který se objeví jako jeden bod na grafu pro daného uživatele. (optimální je 7 dní pro podrobnější data a měsíc pro přehled, u nižších hodnot se grafy stávají nepřehlednými, jelikož je počet zpráv za období velice proměnlivý)

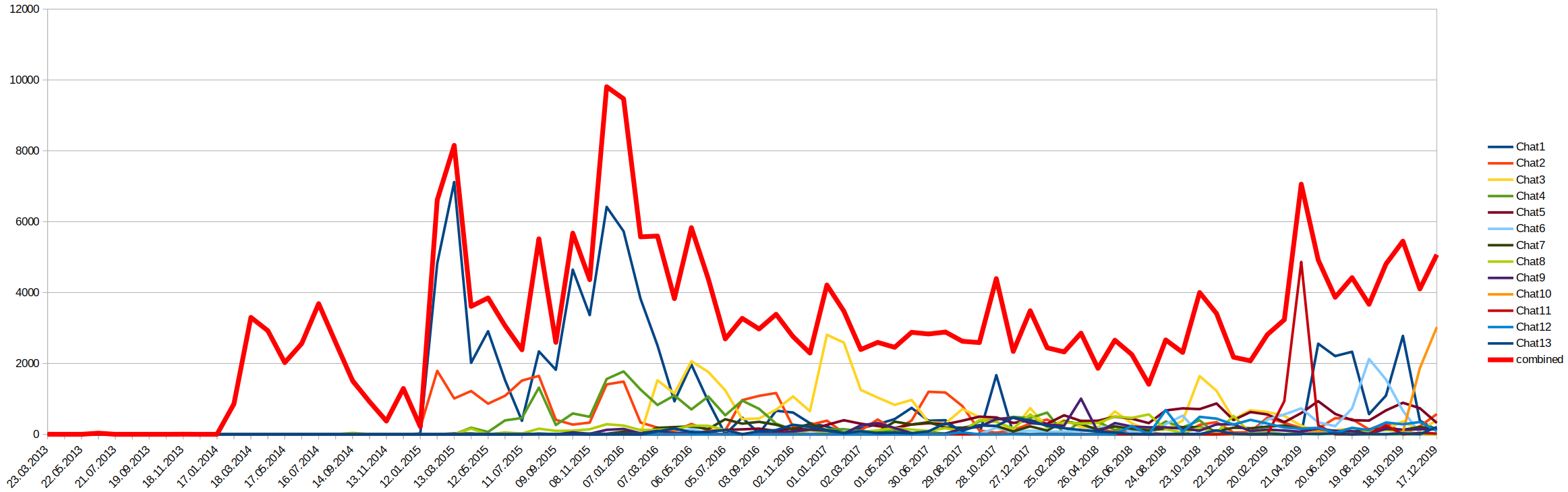
Po potvrzení se spustí samotné sčítání a ukládání výsledků. Operace obvykle trvá několik sekund až minut u velkých vzorků.

Zároveň je také vygenerován soubor se souhrnnými informacemi ke každé konverzaci a celému vzorku.

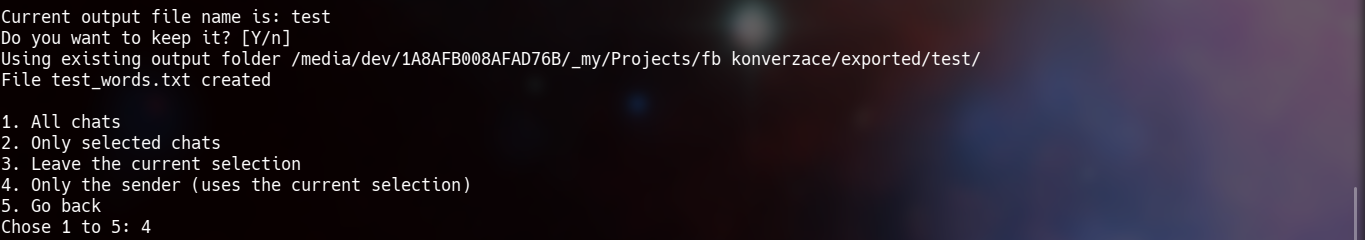
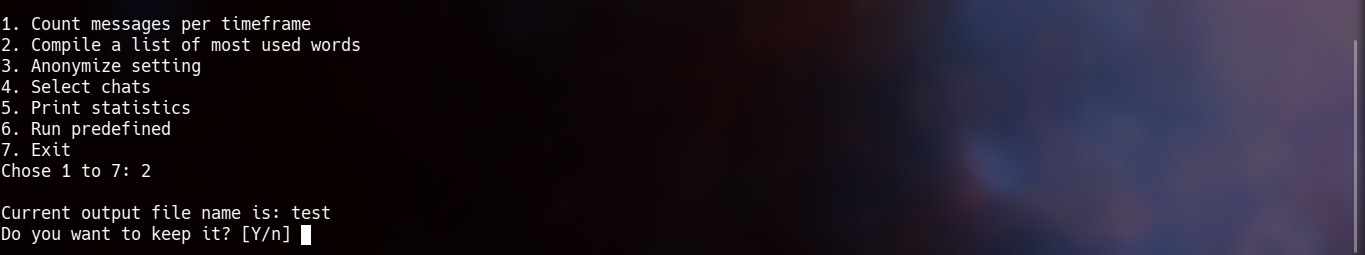
Program se poté vrátí na výchozí menu.

Začátek otevřeného .csv výstupu.

Nastavení pro generování grafu.

Výsledný graf

## Export seznamu nejpoužívanějších slov

Po spuštění a načtení vybereme režim pro export nejpoužívanějších slov. Jelikož je program stále spuštěný z předchozí ukázky, nabízí stejné jméno pro výstupní složku a soubor.

Upozornění, že je používána stejná složka. Opět stejný výběr jako u předchozí ukázky, zde nás obvykle zajímají nejpoužívanější slova odesílatele.

Po načtení všech slov a vytvoření slovníku je možné vybrat kolik slov se má uložit.

Samotné třídění slov je nejzdlouhavější operace.

# Závěr

Jak jsem již zmínil, aplikace je docela rozšiřitelná. Bylo by například možné přidat možnost načtení několika vzorků najednou a porovnávat mezi nimi nebo přidat více režimů – export zpráv, zaměření na jiné aspekty (nejpoužívanější emoji, kdo z konverzace je aktivnější, atd.), ale zároveň osobně neprahnu po podrobné analýze svých konverzací, nebo konverzací jiných. Primární cíl tohoto projektu byl seznámit se s dalším aspektem práce s daty a zkusit co Python dovede.

Jsem si jistý, že s drobnými úpravami by bylo možné načíst data z dalších platforem. Podezírám Facebook z úmyslného ztěžování práce s jejich databázemi. Potýkal jsem se například s problémem špatného kódování prakticky všech znaků, které nejsou v základní ASCI. Nakonec vyšlo najevo, že jejich interní formát je odlišný od formátu, do kterého jsou kódovány znaky ve stažených databázích – interně UTF-8, ale ve staženém jsonu latin-1 – což velice komplikuje načítání a vyžaduje náročné dekódování a následné kódování do jiného formátu na úrovni binárních dat.