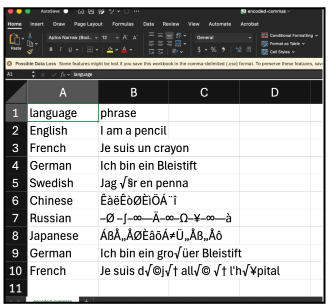
Homework #2: Part 3

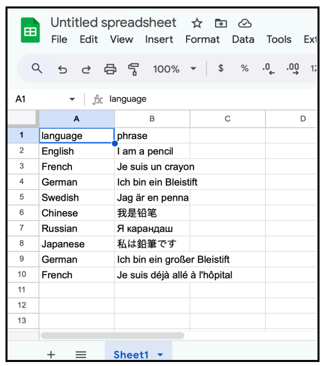
Foundations of Computing

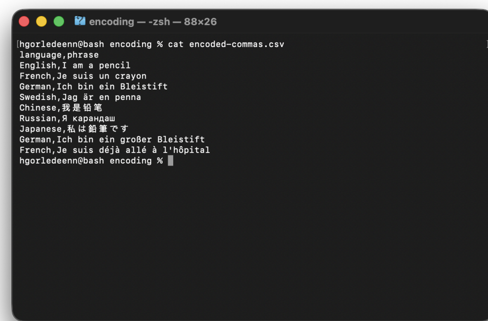
Holden Green
11-02-2025

I. Cat and character encodings

encoded-commas.csv:







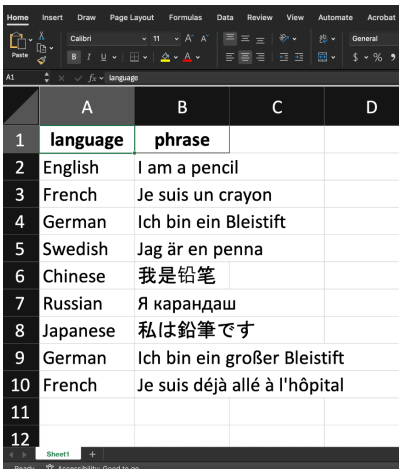
In Excel, the data looks … weird. Excel correctly formats the data (making a sheet with 2 columns and 10 rows), but, for any non-English character, it enters a weird string of symbols that are somewhat unintelligible.

In Google Sheets, this problem with non-English characters seems to not come up. Google sheets is able to both accurately parse the data into rows and columns and return non-English characters within those cells.

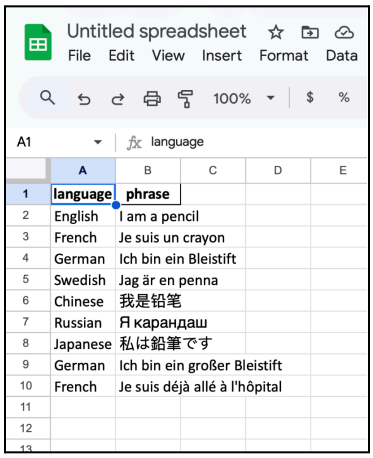
After setting the correct directory, I ran the code `cat encoded-commas.csv` to view the data. The cat function is also able to handle non-English characters, but it returns the csv file without the same column formatting as Google Sheets or Excel. Instead of returning a table, it has each row listed, showing the values in that row separated by commas. This might be helpful for a computer to read, but it is definitely less intuitive to look at than the output from Excel or Google Sheets.

One question I have after this exercise is: does having commas within a cell stop someone from being able to save or view data properly as a .csv? If the English sentence had been "I am a pencil, I think" would cat think that English had 3 columns – one "English" one "I am a pencil" and one "I think"?

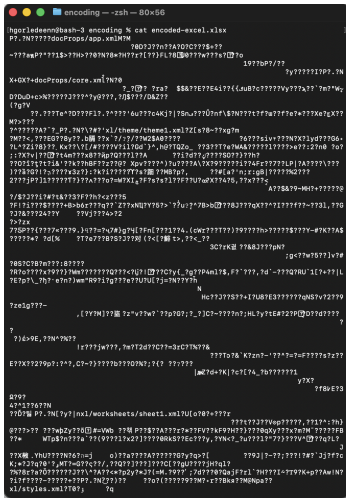
encoded-excel.xlsx:



	A	B	C	D
1	language	phrase		
2	English	I am a pencil		
3	French	Je suis un crayon		
4	German	Ich bin ein Bleistift		
5	Swedish	Jag är en penna		
6	Chinese	我是铅笔		
7	Russian	Я карандаш		
8	Japanese	私は鉛筆です		
9	German	Ich bin ein großer Bleistift		
10	French	Je suis déjà allé à l'hôpital		
11				
12				



	A	B	C	D	E
1	language	phrase			
2	English	I am a pencil			
3	French	Je suis un crayon			
4	German	Ich bin ein Bleistift			
5	Swedish	Jag är en penna			
6	Chinese	我是铅笔			
7	Russian	Я карандаш			
8	Japanese	私は鉛筆です			
9	German	Ich bin ein großer Bleistift			
10	French	Je suis déjà allé à l'hôpital			
11					
12					
13					



In this case, Excel did everything perfectly – correctly formatted the data and correctly displayed any non-English characters. It even included formatting for row 1 of the data.

Similarly to Excel, Google Sheets did everything right. It formatted the data, maintained non-English characters and formatted the first row of the data.

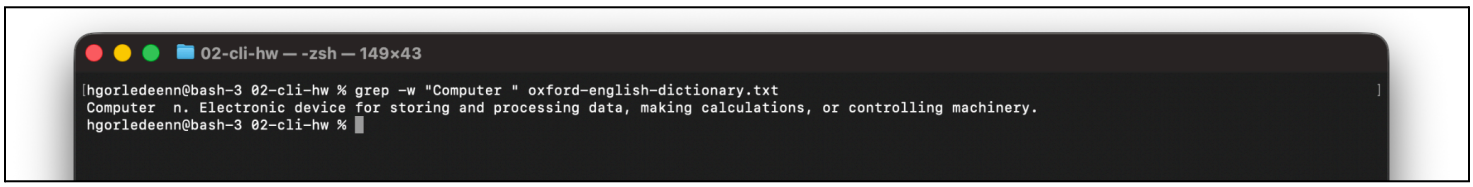
After setting the correct directory, I ran the code cat encoded-excel.xlsx to view the data. The cat function really did not like this file. Instead of anything human intelligible, running this function returned a string of characters that I really can't make sense of. I think the takeaway is that cats don't like Excel.

II. Grep

Search *Oxford English Dictionary* for "computer"

I ran the command line grep "computer" oxford-english-dictionary.txt and got a list of every line with the word computer. (except for, I think, the actual entry for the word computer). I think this happened because the first word of each line is capitalized, and my initial code was case-sensitive.

In fact, when I run the same line with computer capitalized (grep "Computer" oxford-english-dictionary.txt), I get 18 entries, all of which probably did not appear in my first search. I can correct this by entering the code grep -i "computer" oxford-english-dictionary.txt. The "-i" is the indicator for grep to search for my supplied string case-insensitive (would return instances of any variety of "computer", like "Computer" or "cOmPuTeR" or "compUTER").



I ran the line grep -w "Computer " oxford-english-dictionary.txt (note the space within the quotes after Computer) to return only the entry for the word "Computer". I noticed the syntax that each word being defined had two spaces following it at the start of the row. I first tried entering two spaces within the quotes after Computer, but that returned no results. For some reason that I don't fully understand, only including one quote after Computer returns the desired single dictionary entry.
My guess is that grep might automatically add a space after my entered string, so it is really searching every entry for "Computer_ _", even though I only told it to search for the string "Computer_".

III. Becoming independently wealthy by selling posters

I tried on my own to figure out how to use yt-dlp, but I was not doing it very well. I ended up using ChatGPT to help. My conversation is linked here.

I ran the code below (courtesy of ChatGPT) to download the video from YouTube.

```
Python
yt-dlp -f "bestvideo+bestaudio/best" -o
"/Users/hgorledeenn/Desktop/CJS/1025foundationsOfComputing/HW2/part
3/%(title)s.%(ext)s" --merge-output-format mp4
"https://www.youtube.com/watch?v=b53QJYP-lqY"
```

I wrote the ffmpeg code by myself but needed help from ChatGPT to debug it. I've included my conversation here, and you can see the code I wrote originally in my question.

After ChatGPT's help, I also added to the path to have the screenshots go into a new folder. I ran the code:

```
Python
## Download each frame's color
ffmpeg -i rush_troye_sivan.mp4 -vf scale=1:1
~/desktop/cjs/1025foundationsOfComputing/hw2/part3/rush_ss/%06d.png
```

I also used ChatGPT to help (only a tiny bit) when I couldn't figure out how to get mogrify and -scale to work. I included my conversation here (I just had to move the x from before the scale amount to after it, because I wanted to change the width and not the height).

```
Python
## Convert each individual frame to one vertical stack
convert -append ~/desktop/cjs/1025foundationsOfComputing/hw2/part3/rush_ss/*
rush_troye_sivan.png

## Use mogrify to make the out.png file taller and wider (to get aspect ratio
4x6)
mogrify -scale x9000\! rush_troye_sivan.png
mogrify -scale 6000x\! rush_troye_sivan.png
```

I used other online sources to help me figure out the code to add a border and a title.

```
Python
## Add a border to the image
mogrify -bordercolor white -border 500 rush_troye_sivan.png

## Added a title to the image
mogrify -gravity north -pointsize 300 -font "Butler" -fill black -annotate +0+125
"Rush, Troye Sivan" rush_troye_sivan.png

mogrify -gravity north -pointsize 300 -font "Butler" -fill black -annotate +0+125
"This Song, Conan Gray" this_song_conan_gray.png
```

You can view the posters I made here.

IV. AI-based transcription, summarization and translation

English Language Analysis

I ran the following lines in Terminal:

```python
## Download the video
yt-dlp -f "bestvideo[ext=mp4]+bestaudio[ext=mp4]/best[ext=mp4]" -o
"/Users/hgorledeenn/Desktop/CJS/1025foundationsOfComputing/HW2/part4/%(title)s.%(ext
)s" --merge-output-format mp4 "https://www.youtube.com/watch?v=g1BWEWyJyY8"

## Install Whisper
pip install -U openai-whisper

## Transcribe video using tiny.en model
whisper soma_video.mp4 --model tiny.en

## Transcribe video using base.en model
whisper soma_video.mp4 --model base.en
```

My first observation is that the base model was a lot slower than the tiny model. They were both super quick (under 2 mins), but if the only consideration for me were speed, I think I'd choose tiny.en.

You can see the conversation where I asked ChatGPT to summarize the transcript and provide ideas here.

Non-English Language Analysis

```python
## Download the video
yt-dlp -f "bestvideo[ext=mp4]+bestaudio[ext=mp4]/best[ext=mp4]" -o
"/Users/hgorledeenn/Desktop/CJS/1025foundationsOfComputing/HW2/part4/%(title)s.
%(ext)s" --merge-output-format mp4
"https://www.youtube.com/watch?v=qdSetYup3aA"

## Transcribe video using tiny model
whisper svenska_video.mp4 --model tiny --language Swedish

## Transcribe video using base model
whisper svenska_video.mp4 --model base --language Swedish
```

Similarly to before, the transcription from the tiny model was quicker than the base model. However, the tiny model made significantly more mistakes than the base model (the base model made mistakes too). Interestingly, the video I looked at was a short profile of a queer bar in Stockholm, and the people being interviewed talked about the need for spaces like that in Stockholm given the lack of queer-focused spaces. The model seemed to have a tough time transcribing the word "queer" in particular (maybe because it's a relatively new or little-discussed word in Swedish).

Both the models struggled anytime they used English words – like "safe space" – in the video. My understanding of normal Swedish speech patterns is that English words are sometimes sprinkled into everyday conversation because most Swedish people are bilingual. I'm curious if this is just a problem with these quicker models, or if this somewhat unique case can be handled in another way. Is it possible to run a bilingual transcription model?

Out of curiosity, I tried the turbo model, and it produced a pretty flawless transcription of the video. It did take the longest of the three models, but it did not have any of the errors that I mentioned above.

Your video link is here.
The Swedish video link is here.
You can view the transcripts here.

V. Exciting command line data analysis

I worked on the VisiData tutorial (though, admittedly, my brain was a little fried). I'll go through it again in a few days to make sure I have it down.