Data Engineering Day 14:

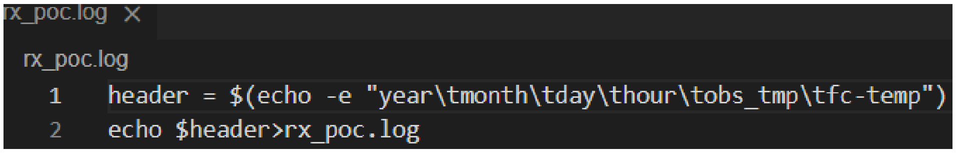
The credit for this course goes to Coursera. Click More

Another link : Azure data Engineer

Historical weather forecast Comparison to actuals.

In this mini-project, I am going to apply extract the raw weather forecast data implementing ETL and make it used for other purposes using bash scripting.

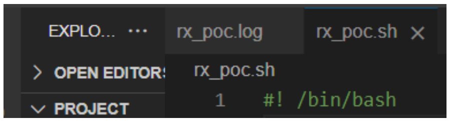
Step 1: make a file named ***rx_poc.log*** using ***touch*** command and write header initialize weather variable with year, month, day, hour, and the temperature as shown in the figure below.



```
header = $(echo -e "year\tmonth\tday\thour\tobs_tmp\tfc-temp")
echo $header>rx_poc.log
```

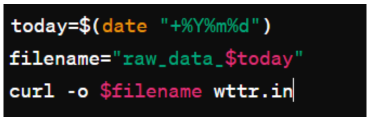
Step 2: Write a bash script that downloads the raw weather data, and extracts and loads the required data.

2.1 Create a text file called ***rx_poc.sh*** and make it a bash script.

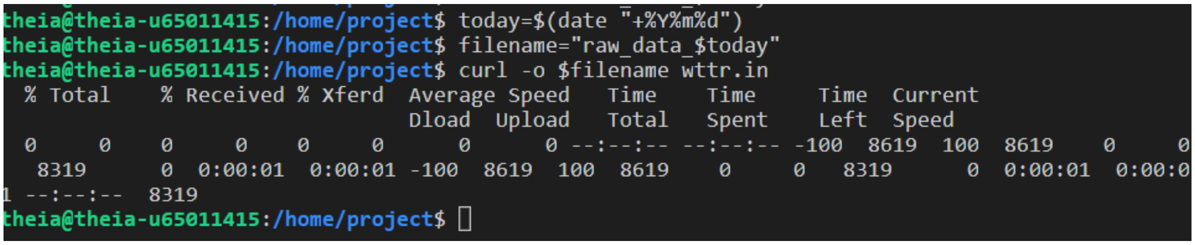


```
#! /bin/bash
```

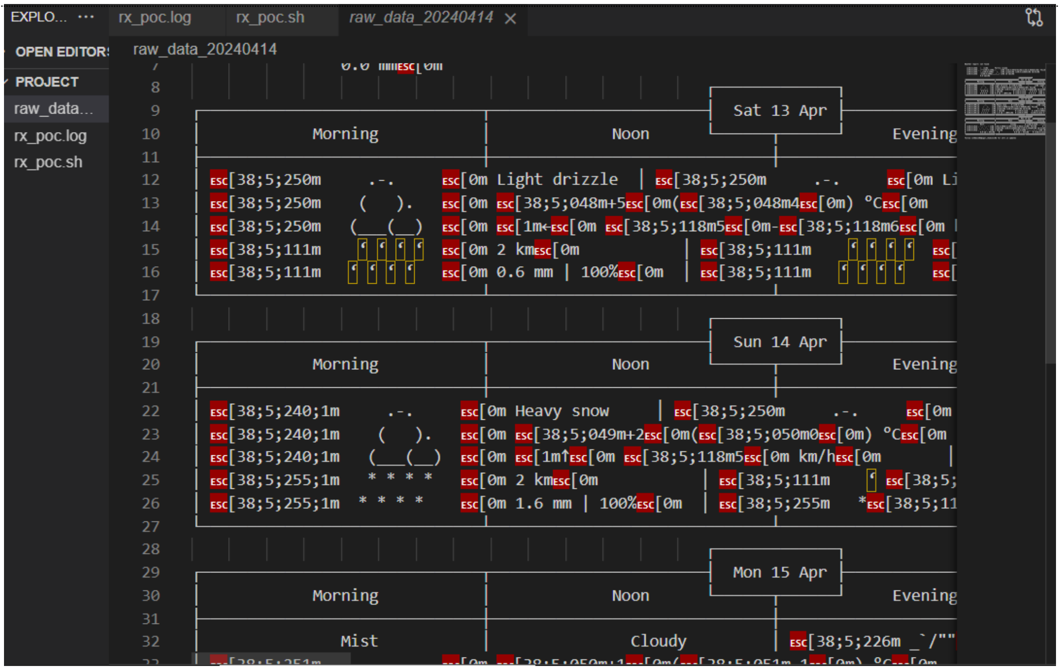
2.2 Download today's weather report from wttr.in.



```
today=$(date "+%Y%m%d")
filename="raw_data_$today"
curl -o $filename wttr.in
```

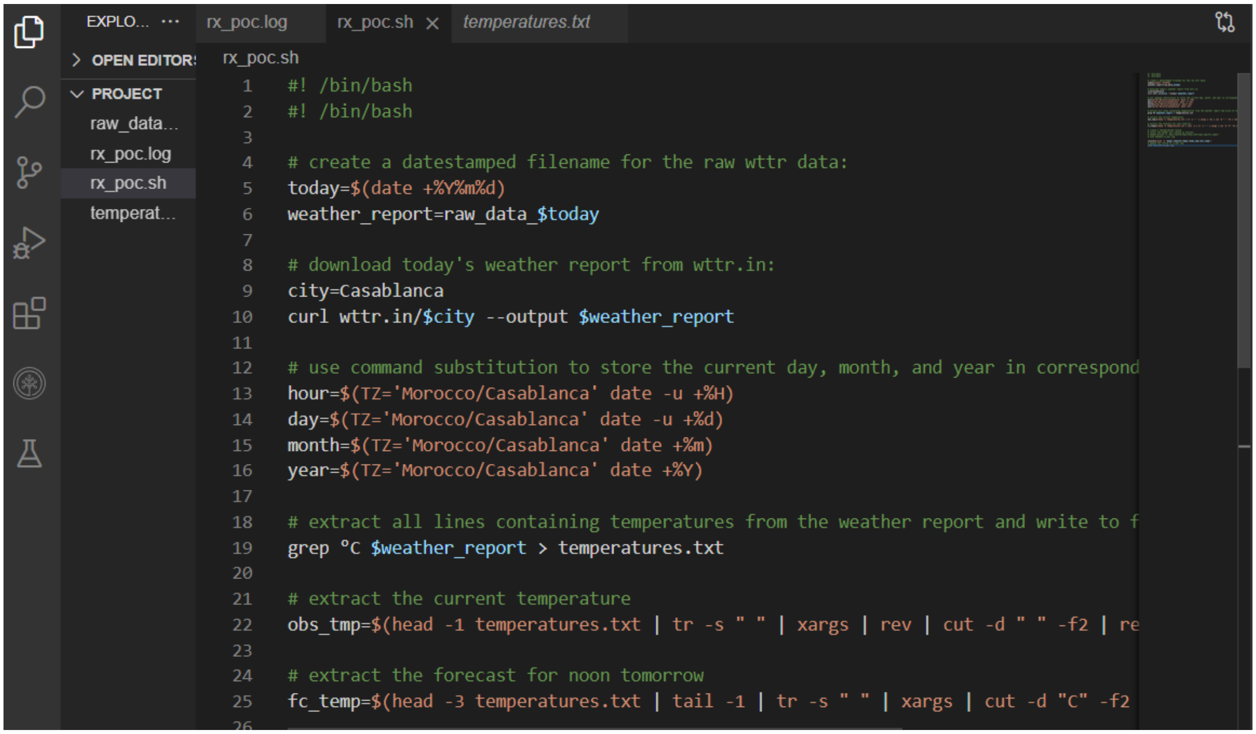


```
theia@theia-u65011415:/home/project$ today=$(date "+%Y%m%d")
theia@theia-u65011415:/home/project$ filename="raw_data_$today"
theia@theia-u65011415:/home/project$ curl -o $filename wttr.in
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- -100  8619  100  8619    0     0
   8319      0  0:00:01  0:00:01 -100  8619  100  8619    0     0   8319      0  0:00:01  0:00:0
1 --:--:--  8319
theia@theia-u65011415:/home/project$ 
```

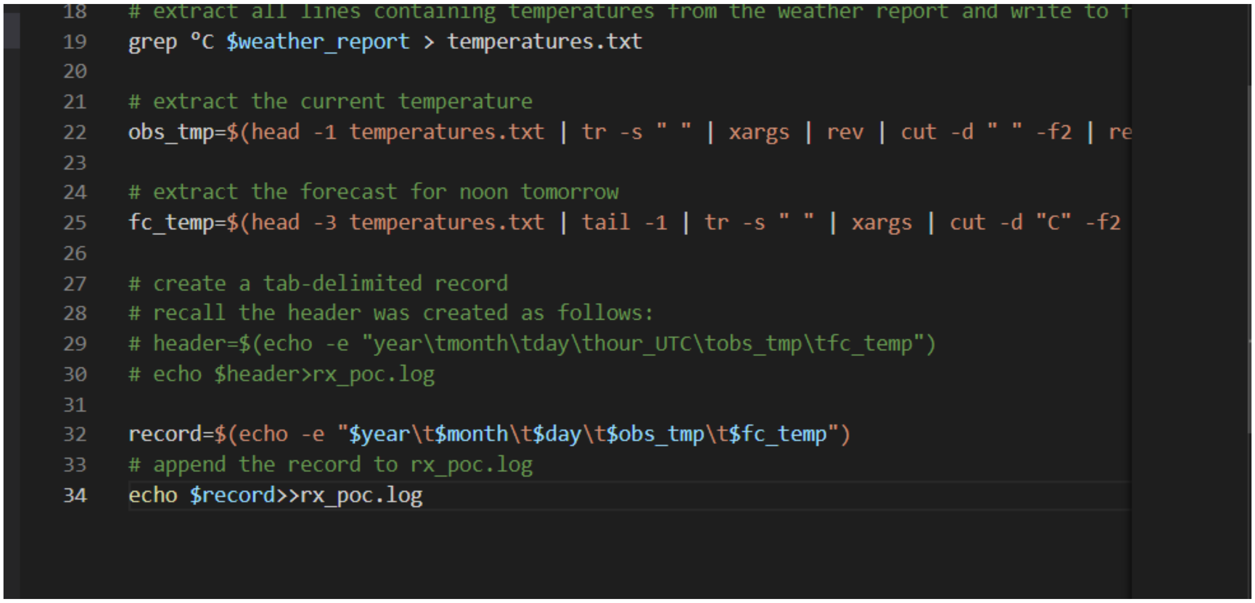


```
                          0.0 mmESC[0m
                                                       ┌─────────────┐
┌──────────────────────────────┬───────────────────────┤  Sat 13 Apr ├───────────────────────┬
│            Morning           │             Noon      └──────┬──────┘     Evening
├──────────────────────────────┼──────────────────────────────┼─────────────────────────────
│ESC[38;5;250m     .-.     ESC[0m Light drizzle  │ESC[38;5;250m     .-.     ESC[0m Li
│ESC[38;5;250m    (   ).   ESC[0m ESC[38;5;048m+5ESC[0m(ESC[38;5;048m4ESC[0m) °CESC[0m
│ESC[38;5;250m   (___(__)  ESC[0m ESC[1m←ESC[0m ESC[38;5;118m5ESC[0m-ESC[38;5;118m6ESC[0m
│ESC[38;5;111m    ʻ ʻ ʻ ʻ  ESC[0m 2 kmESC[0m           │ESC[38;5;111m    ʻ ʻ ʻ ʻ  ESC[
│ESC[38;5;111m   ʻ ʻ ʻ ʻ   ESC[0m 0.6 mm | 100%ESC[0m  │ESC[38;5;111m   ʻ ʻ ʻ ʻ   ESC[
└──────────────────────────────┴──────────────────────────────┴─────────────────────────────
                                                       ┌─────────────┐
┌──────────────────────────────┬───────────────────────┤  Sun 14 Apr ├───────────────────────┬
│            Morning           │             Noon      └──────┬──────┘     Evening
├──────────────────────────────┼──────────────────────────────┼─────────────────────────────
│ESC[38;5;240;1m     .-.     ESC[0m Heavy snow     │ESC[38;5;250m     .-.     ESC[0m
│ESC[38;5;240;1m    (   ).   ESC[0m ESC[38;5;049m+2ESC[0m(ESC[38;5;050m0ESC[0m) °CESC[0m
│ESC[38;5;240;1m   (___(__)  ESC[0m ESC[1m↑ESC[0m ESC[38;5;118m5ESC[0m km/hESC[0m        │
│ESC[38;5;255;1m    * * * *  ESC[0m 2 kmESC[0m            │ESC[38;5;111m    ʻ ESC[38;5;
│ESC[38;5;255;1m   * * * *   ESC[0m 1.6 mm | 100%ESC[0m   │ESC[38;5;255m   *ESC[38;5;11
└──────────────────────────────┴──────────────────────────────┴─────────────────────────────
                                                       ┌─────────────┐
┌──────────────────────────────┬───────────────────────┤  Mon 15 Apr ├───────────────────────┬
│            Morning           │             Noon      └──────┬──────┘     Evening
├──────────────────────────────┼──────────────────────────────┼─────────────────────────────
│              Mist            │              Cloudy          │ESC[38;5;226m _`/""
```

# The final bash file should look like below.



```bash
#! /bin/bash
#! /bin/bash

# create a datestamped filename for the raw wttr data:
today=$(date +%Y%m%d)
weather_report=raw_data_$today

# download today's weather report from wttr.in:
city=Casablanca
curl wttr.in/$city --output $weather_report

# use command substitution to store the current day, month, and year in correspond
hour=$(TZ='Morocco/Casablanca' date -u +%H)
day=$(TZ='Morocco/Casablanca' date -u +%d)
month=$(TZ='Morocco/Casablanca' date +%m)
year=$(TZ='Morocco/Casablanca' date +%Y)

# extract all lines containing temperatures from the weather report and write to f
grep °C $weather_report > temperatures.txt

# extract the current temperature
obs_tmp=$(head -1 temperatures.txt | tr -s " " | xargs | rev | cut -d " " -f2 | re

# extract the forecast for noon tomorrow
fc_temp=$(head -3 temperatures.txt | tail -1 | tr -s " " | xargs | cut -d "C" -f2
```



```
# extract all lines containing temperatures from the weather report and write to f
grep °C $weather_report > temperatures.txt

# extract the current temperature
obs_tmp=$(head -1 temperatures.txt | tr -s " " | xargs | rev | cut -d " " -f2 | re

# extract the forecast for noon tomorrow
fc_temp=$(head -3 temperatures.txt | tail -1 | tr -s " " | xargs | cut -d "C" -f2

# create a tab-delimited record
# recall the header was created as follows:
# header=$(echo -e "year\tmonth\tday\thour_UTC\tobs_tmp\tfc_temp")
# echo $header>rx_poc.log

record=$(echo -e "$year\t$month\t$day\t$obs_tmp\t$fc_temp")
# append the record to rx_poc.log
echo $record>>rx_poc.log
```

Skills Network

IBM Course: Hands-on Introduction to Linux Commands and Shell Scripting

Introduction to Linux

Command	Syntax	Description	Example
List	`ls [OPTIONS] [FILE/DIRECTORY]`	List files and directories at path	`ls /home/user/documents`
Print Working Directory	`pwd`	Print present working directory	`pwd`
Change Directory	`cd [DIRECTORY]`	Change current directory	`cd /home/user/documents`
Super user do	`sudo [COMMAND]`	Run command with superuser privileges	`sudo apt update`
Text Editor	`nano [FILE]`	Open file with Nano text editor	`nano myfile.txt`

Introduction to Linux Commands

Informational, Navigational, & Management Commands

Command	Syntax	Description	Example
Who Am I	`whoami`	Return username	`whoami`
User ID	`id`	Return current user or group ID	`id`
System Information	`uname [OPTIONS]`	Display system information	`uname -a`
Manual Pages	`man [COMMAND]`	Display manual page for a command	`man ls`
Curl	`curl [OPTIONS] [URL]`	Transfer data from or to server	`curl https://some_website.com`
Date	`date [OPTIONS]`	Display current date and time	`date`
Find	`find [DIRECTORY] [OPTIONS]`	Find files and directories at specified path	`find /home/user -name '*.txt'`
Make Directory	`mkdir [DIRECTORY]`	Create new directory	`mkdir myfolder`

Command	Syntax	Description	Example
Remove Directory	`rmdir [DIRECTORY]`	Remove empty directory	`rmdir myfolder`
Process Status	`ps [OPTIONS]`	Display process status information	`ps -ef`
Table of Processes	`top`	Display live system resource usage	`top`
Disk Usage	`df [OPTIONS] [FILESYSTEM]`	Display disk space usage	`df -h`
Create Empty File	`touch [FILE]`	Create new file or update timestamp	`touch myfile.txt`
Copy	`cp [OPTIONS] [SOURCE] [DESTINATION]`	Copy files or directories from source to destination	`cp myfile.txt /home/user/documents`
Move	`mv [OPTIONS] [SOURCE] [DESTINATION]`	Move or rename files and directories	`mv myfile.txt /home/user/documents`
Remove	`rm [OPTIONS] [FILE/DIRECTORY]`	Remove files	`rm my_scratch_file.txt`

Command	Syntax	Description	Example
		Remove nonempty directory	`rm -r path_to_temp_directory`
	`rmdir [OPTIONS] [DIRECTORY]`	Remove empty directory	`rmdir path_to_my_directory`
Change Mode	`chmod [OPTIONS] [MODE] [FILE]`	Change file or directory permissions	`chmod u+x myfile.txt`

Working with Text Files, Networking & Archiving Commands

Command	Syntax	Description	Example
Concatenate	`cat [FILE]`	Display the contents of a file	`cat myfile.txt`
		Concatentate and display contents of multiple files	`cat file1 file2`
More	`more [FILE]`	Display file one screen at a time	`more myfile.txt`
Head	`head [OPTIONS] [FILE]`	Display first N lines of file	`head -5 myfile.txt`
Tail	`tail [OPTIONS] [FILE]`	Display last N lines of file	`tail -5 myfile.txt`

Command	Syntax	Description	Example
Echo	`echo [ARGUMENTS]`	Display arguments in console	`echo Hello, World!`
Sort	`sort [OPTIONS] [FILE]`	Alphanumerically sort file contents	`sort file.txt`
Unique	`uniq [OPTIONS] [FILE]`	Report or remove consecutively repeated lines in file	`uniq file.txt`
Word Count	`wc [OPTIONS] [FILE]`	Print the number of lines, words, and characters in a file	`wc file.txt`
Grep	`grep [OPTIONS] PATTERN [FILE]`	Search for a specified pattern in a file	`grep "hello" file.txt`
Paste	`paste [OPTIONS] [FILE1] [FILE2]`	Merge lines of files side by side	`paste file1.txt file2.txt`
Cut	`cut [OPTIONS] [FILE]`	Remove sections from each line of a file	`cut -d":" -f1 /etc/passwd`

Command	Syntax	Description	Example
Tar	`tar [OPTIONS] [FILE]`	Archive files together into a single file	`tar -czvf archive.tar.gz /directory`
Zip	`zip [OPTIONS] [FILE]`	Compress files into a zip archive	`zip archive.zip file1.txt file2.txt`
Unzip	`unzip [OPTIONS] [FILE]`	Uncompress files from a zip archive	`unzip archive.zip`
Hostname	`hostname`	Print the name of the current host system	`hostname`
Ping	`ping [OPTIONS] HOSTNAME/IP`	Send ICMP ECHO_REQUEST packets to a network host	`ping google.com`
Ifconfig	`ifconfig [INTERFACE]`	Display or configure network interface parameters	`ifconfig`
IP	`ip [OPTIONS]`	Show or manipulate routing, devices, policy routing, and tunnels	`ip addr`

Command	Syntax	Description	Example
Curl	`curl [OPTIONS] URL`	Transfer data from or to a server	`curl https://some_website.com`
Wget	`wget [OPTIONS] URL`	Download files from the web	`wget https://some_website.com/some_file.txt`

Introduction to Shell Scripting

Command	Syntax	Description	Example
Shebang	#!/bin/[shell]	First line of shell script	#!/bin/bash
Pipe	filter1 | filter2	Chain any number of filters	ls | sort -r
Locate executable	`which [EXECUTABLE]`	Display location of `bash` executable	`which bash`
Bash	`bash [SCRIPT]`	Interpret and run script using Bash shell	`bash script.txt`
Set	`set [OPTION]`	List all shell variables	`set`
Define variable	`[VARIABLE_NAME]=[VALUE]`	Define shell variable by name and assign value	`name="John"`
Read	`read [VARIABLE]`	Read from standard input and	`read name`

Command	Syntax	Description	Example
		store result in variable	
Env	`env`	Print all environment variables and their values	`env`
Export	`export [VARIABLE]`	Extend scope of local variable to all child processes	`export name`
Crontab	`crontab [OPTIONS]`	Open crontab default editor	`crontab -e`
		List all cron jobs	`crontab -l`
Schedule tasks to run at specified times using cron daemon	`m h dom mon dow command`	Append date/time to file every Sunday at 6:15 pm	`15 18 * * 0 date >> sundays.txt`
		Back up home directory every Monday at 3:00 am	`0 3 * * 1 tar -cvf my_backup_path\my_archive.tar.gz $HOME\`
		Run shell script first minute of first day of each month	`1 0 1 * * ./My_Script.sh`

Authors

Jeff Grossman
Sam Propupchuk

Other Contributors

Rav Ahuja

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2023-06-07	1.2	Jeff Grossman	Add missing content
2023-05-10	2.0	Nick Yi	Created table version
2023-04-26	1.3	Nick Yi	Updated to reflect cheat sheet changes
2023-03-20	1.2	Jeff Grossman	Add text wrangling content
2023-03-08	1.1	Jeff Grossman	Add content for M4 Projects
2023-03-08	1.0	Jeff Grossman	Merge from module cheat sheets