

# Data Wrangling Lab Report

## Lab #28 - Data Wrangling Techniques



**Ahmad Mukhtar**

**National University of Sciences and Technology (NUST)  
Chip Design Centre (NCDC), Islamabad, Pakistan**

October 13, 2024

# Contents

0.1	Task 1: Interactive Regex Tutorial . . . . .	2
0.1.1	Objective . . . . .	2
0.1.2	Screenshots of Completed Exercises . . . . .	2
0.2	Task 2: Finding Words . . . . .	6
0.2.1	Part A: Extracting Assembly Instructions . . . . .	6
0.2.2	Part B: Processing Words in article.txt . . . . .	7

## 0.1 Task 1: Interactive Regex Tutorial

### 0.1.1 Objective

In this task, the aim was to complete a series of 15 exercises to explore the fundamentals of regular expressions (regex), including character matching, quantifiers, groups, and anchors.

### 0.1.2 Screenshots of Completed Exercises

Exercise 1: Matching Characters

Task	Text	
Match	abcdefg	✓
Match	abcde	✓
Match	abc	✓

Continue >

Solve the above task to continue on to the next problem, or read the [Solution](#).

Figure 1: Exercise 1: Matching characters

Exercise 1½: Matching Digits

Task	Text	
Match	abc123xyz	✓
Match	define "123"	✓
Match	var g = 123;	✓

Continue >

Solve the above task to continue on to the next problem, or read the [Solution](#).

Figure 2: Exercise 2: Character Ranges and Classes

Exercise 2: Matching With Wildcards

Task	Text	
Match	cat.	✓
Match	896.	✓
Match	?=+.	✓
Skip	abc1	

Continue >

Solve the above task to continue on to the next problem, or read the [Solution](#).

Figure 3: Exercise 3: Using Quantifiers

Exercise 3: Matching Characters

Task	Text	
Match	can	✓
Match	man	✓
Match	fan	✓
Skip	dan	
Skip	ran	
Skip	pan	

Solve the above task to continue on to the next problem, or read the [Solution](#).

Figure 4: Exercise 4: Escaping Special Characters

Exercise 4: Excluding Characters

Task	Text	
Match	hog	✓
Match	dlog	✓
Skip	bog	

Solve the above task to continue on to the next problem, or read the [Solution](#).

Figure 5: Exercise 5: Anchors

Exercise 5: Matching Character Ranges

Task	Text	
Match	Ana	✓
Match	Bob	✓
Match	Cpc	✓
Skip	aax	
Skip	bby	
Skip	ccz	

Solve the above task to continue on to the next problem, or read the [Solution](#).

Figure 6: Exercise 6: Word Boundaries

Exercise 6: Matching Repeated Characters

Task	Text	
Match	wazzzzup	✓
Match	wazzzup	✓
Skip	wazup	

Solve the above task to continue on to the next problem, or read the [Solution](#).

Figure 7: Exercise 6 Part 2: Using Word Boundaries

Exercise 7: Matching Repeated Characters

Task	Text	
Match	aaabcc	✓
Match	aabbbbc	✓
Match	aacc	✓
Skip	a	

[Continue >](#)

Solve the above task to continue on to the next problem, or read the [Solution](#).

Figure 8: Exercise 7: Matching Digits and Non-Digits

Exercise 8: Matching Optional Characters

Task	Text	
Match	1 file found?	✓
Match	2 files found?	✓
Match	24 files found?	✓
Skip	No files found.	

[Continue >](#)

Solve the above task to continue on to the next problem, or read the [Solution](#).

Figure 9: Exercise 8: Matching Whitespace

Exercise 9: Matching Whitespace

Task	Text	
Match	1. abc	✓
Match	2. abc	✓
Match	3. abc	✓
Skip	4.abc	

[Continue >](#)

Figure 10: Exercise 9: Capturing Groups

Exercise 10: Matching Lines

Task	Text	
Match	Mission: successful	✓
Skip	Last Mission: unsuccessful	
Skip	Next Mission: successful upon capture of target	

[Continue >](#)

Solve the above task to continue on to the next problem, or read the [Solution](#).

Figure 11: Exercise 10: Repetitions and Quantifiers

Exercise 11: Matching Groups

Task	Text	Capture Groups
Capture	file_record_transcript.pdf	file_record_transcript ✓
Capture	file_07241999.pdf	file_07241999 ✓
Skip	testfile_fake.pdf.tmp	

[Continue >](#)

Figure 12: Exercise 11: Alternation (OR)

Exercise 12: Matching Nested Groups

Task	Text	Capture Groups
Capture	Jan 1987	Jan 1987 1987 ✓
Capture	May 1969	May 1969 1969 ✓
Capture	Aug 2011	Aug 2011 2011 ✓

[Continue >](#)

*Solve the above task to continue on to the next problem, or read the [Solution](#).*

Figure 13: Exercise 12: Matching Patterns with Subgroups

Exercise 13: Matching Nested Groups

Task	Text	Capture Groups
Capture	1280x720	1280 720 ✓
Capture	1920x1600	1920 1600 ✓
Capture	1024x768	1024 768 ✓

[Continue >](#)

*Solve the above task to continue on to the next problem, or read the [Solution](#).*

Figure 14: Exercise 13: Non-Capturing Groups

Exercise 14: Matching Conditional Text

Task	Text	
Match	I love cats	✓
Match	I love dogs	✓
Skip	I love logs	
Skip	I love cogs	

[Continue >](#)

*Solve the above task to continue on to the next problem, or read the [Solution](#).*

Figure 15: Exercise 14: Greedy vs Lazy Matching

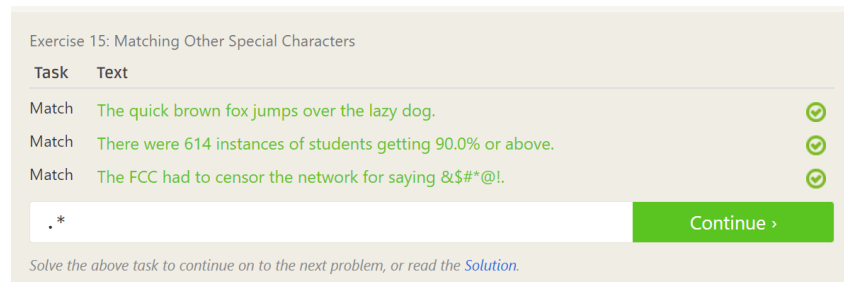


Figure 16: Exercise 15: Final Completion

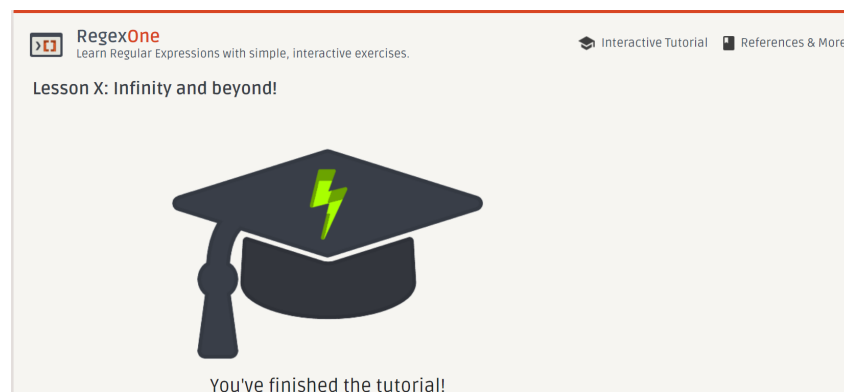


Figure 17: Completion Screenshot: Regex Tutorial Completion

## 0.2 Task 2: Finding Words

### 0.2.1 Part A: Extracting Assembly Instructions

#### Objective

The goal of this task was to write a shell script to search for and extract assembly instructions associated with a given PC value from the `core.txt` file.

#### Bash Script

Listing 1: Task 2: Extracting Assembly Instructions

```
#!/bin/bash

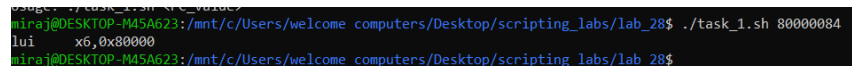
# Check if the user provided a PC value
if [ $# -ne 1 ]; then
    echo "Usage: _$0_<PC_value>"
    exit 1
fi

# Store the input PC value
PC_VALUE=$1
```

```
# Search for the instruction in the file
result=$(grep "^${PC_VALUE}," core.txt)

# Check if the result is empty (no matching PC value)
if [ -z "$result" ]; then
    echo "Instruction_not_present_in_file"
else
    # Extract the full assembly instruction if the PC value is found
    echo "$result" | awk -F',' '{match($0,/"([^\"]+)/,arr);print arr[1]}'
fi
```

## Output



```
miraj@DESKTOP-M45A623:/mnt/c/Users/welcome_computers/Desktop/scripting_labs/lab_28$ ./task_1.sh 80000084
lui      x6,0x80000
miraj@DESKTOP-M45A623:/mnt/c/Users/welcome_computers/Desktop/scripting_labs/lab_28$
```

Figure 18: Output for Task 2, Part A - Extracting Assembly Instructions

## 0.2.2 Part B: Processing Words in article.txt

### Objective

The goal of this task is to process the `article.txt` file to:

- Find words containing at least one 'a' but not ending in 'i'.
- Identify the three most common two-letter suffixes.
- Count the unique two-letter combinations.
- List the two-letter combinations that do not occur.

### Bash Script

Listing 2: Task 2: Processing Words in article.txt

```
#!/bin/bash

# Step 1: Convert the file to lowercase for case insensitivity
cat article.txt | tr '[:upper:]' '[:lower:]' > clean_article.txt

# Step 2: Extract words containing at least one 'a' and not ending with 'i'
grep -o '\b\w*a\w*\b' clean_article.txt | grep -v 'i$' > filtered_words.txt

# Step 3: Find the three most common last two-letter combinations
echo "Three_most_common_last_two-letter_combinations:"
```



```
awk '{print substr($0, length($0)-1)}' filtered_words.txt | sort |
  uniq -c | sort -nr | head -3

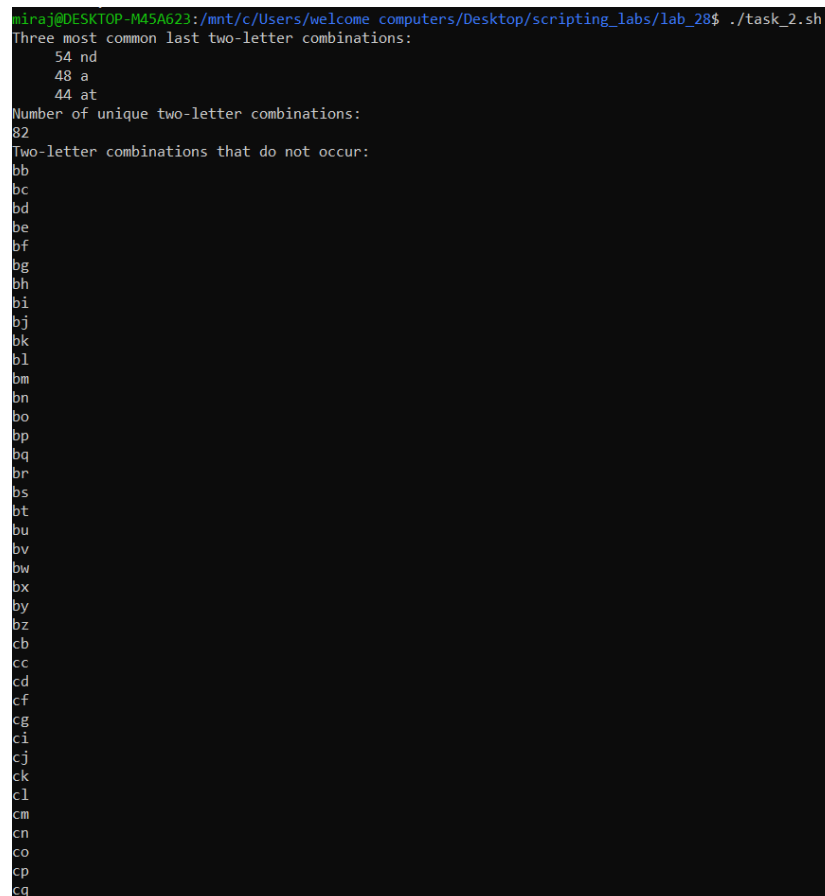
# Step 4: Count how many unique two-letter combinations there are
echo "Number_of_unique_two-letter_combinations:"
awk '{print substr($0, length($0)-1)}' filtered_words.txt | sort |
  uniq | wc -l

# Step 5: Find which two-letter combinations do not occur (
  additional challenge)
echo {a..z}{a..z} | tr '_' '\n' > all_combinations.txt

# Find the combinations that are present in the filtered words
awk '{print substr($0, length($0)-1)}' filtered_words.txt | sort |
  uniq > present_combinations.txt

# Find which two-letter combinations do not occur
echo "Two-letter_combinations_that_do_not_occur:"
grep -v -f present_combinations.txt all_combinations.txt
```

## Output



```
mira@DESKTOP-M45A623:/mnt/c/Users/welcome_computers/Desktop/scripting_labs/lab_28$ ./task_2.sh
Three most common last two-letter combinations:
  54 nd
  48 a
  44 at
Number of unique two-letter combinations:
82
Two-letter combinations that do not occur:
bb
bc
bd
be
bf
bg
bh
bi
bj
bk
bl
bm
bn
bo
bp
bq
br
bs
bt
bu
bv
bw
bx
by
bz
cb
cc
cd
ce
cf
cg
ch
ci
cj
ck
cl
cm
cn
co
cp
cq
```

Figure 19: Output for Task 2, Part B - Processing Words in article.txt