Linux Laboratory-ENCS313

Shell Scripting Project

Faculty of Engineering and Technology

Department of Electrical and Computer Systems Engineering

Made By:

Entimaa Rummaneh **ID** :1180841 Fanan Tamim **ID** :1180070

Instructor: Dr. Mohammad Jubran

Assistant : Eng. Shahd

Section :3

Abstract:

The project aims to create a background for the student on how to write a Shell script program, and perform a range of different functions for data by building a script shell program that does basic dataset preprocessing and manipulations.

❖ Procedure & Discussion:

1) Main Menu:

you have three choices the first one if you enter **D** , you will get dimension, while enter **C** the program will compute Basic statistics the Min, Max, Mean and the standard Deviation of each column. and if you enter **S** and if a sample row contains a missed value as below, the program will substitute the missed value by the mean of the column.

```
#display menu
while true
do
 printf "\nD: for dimension
C: for statistics
S: for substitutes missing values
E: exit\n"
 read operation
 case $operation in
   "D") dimension;;
   "C") stat;;
   "S") substitute;;
   "E") exit 0;;
   *) echo "Enter a valid chiose"
 esac
done
```

2) Enter File:

we enter the file name and search if it exist by (read file name) , then we checked if it is exist and the format is csv file the rest of the program will appear , Other display error message .

```
fanan@fanan-VirtualBox:~$ ./project
Enter file name:
test.txt
file extension error
fanan@fanan-VirtualBox:~$
```

```
fanan@fanan-VirtualBox:~$ ./project
Enter file name:
tst.csv
No such file
fanan@fanan-VirtualBox:~$
```

3) Test Case:

We will test code in file name (test.csv) that contain 5 row and four Column

```
fanan@fanan-VirtualBox:~$ pico test.csv
fanan@fanan-VirtualBox:~$ ./project
Enter file name:
test.csv
sepal.length,sepal.length,petal.length,petal.width
5.1,3.5,1.4,0.2
4.9,3,1.4,0.2
4.7,3.2,1.3,0.2
4.6,3.1,1.5,0.2
5,3.6,1.4,0.2

D: for dimension
C: for statistics
S: for substitutes missing values
E: exit
D
The dimension is 5 x 4

D: for dimension
C: for statistics
S: for substitutes missing values
E: exit
```

```
fanan@fanan-VirtualBox:~$ pico project
fanan@fanan-VirtualBox:~$ ./project
Enter file name:
test.csv
sepal.length,sepal.length,petal.length,petal.width
5.1,3.5,1.4,0.2
4.9,3,1.4,0.2
4.7,3.2,1.3,0.2
4.6,3.1,1.5,0.2
5,3.6,1.4,0.2

D: for dimension
C: for statistics
S: for substitutes missing values
E: exit
C
Min      4.6      3        1.3      0.2
Max      5.1      3.6      1.5      0.2
Mean     4.86     3.28     1.40     .20
STDEV    .18547236         .23151673         .06324555         0

D: for dimension
C: for statistics
S: for substitutes missing values
E: exit
E
fanan@fanan-VirtualBox:~$
```

When enter C then calculate min ,max , mean , the standard Deviation of each column

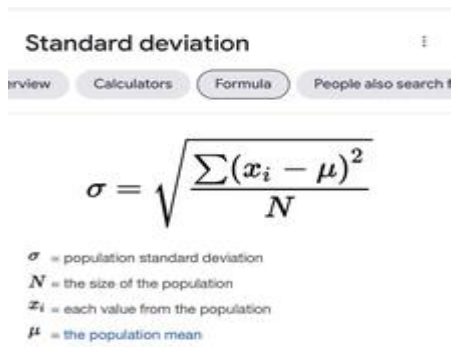
for example, first column that has value **5.1, 4.9, 4.7, 4.6, 5**

min value was 4.6

max value was 5.1

mean value was: **5.1 + 4.9 + 4.7 +4.6 +5=24.3/5=4.86**

the standard Deviation :0.18547236 calculate in formula show in figure:



:

The step for it in code is:

```
#calculate STDEV
m=1
sumDEV=0
while [ "$m" -le "$r" ]
do
  numDEV=$(cat tmp2.txt | cut -d',' -f$m)
  diff="$( bc <<<"$numDEV - $mean" )"
  diff2="$( bc <<<"scale=10; $diff^2" )"
  sumDEV="$( bc <<<"$sumDEV + $diff2" )"
  m=$((m + 1))
done
sd="$( bc <<<"scale=8; $sumDEV / $r" )"
STDEV=$(echo "sqrt($sd)" | bc)
echo $STDEV >> column$i.txt
```

When enter S , the second row contains a missed value as below, the program will substitute the missed value by the mean of the column 3.35

```
fanan@fanan-VirtualBox:~$ pico project
fanan@fanan-VirtualBox:~$ pico test.csv
fanan@fanan-VirtualBox:~$ ./project
Enter file name:
test.csv
sepal.length,sepal.length,petal.length,petal.width
5.1,3.5,1.4,0.2
4.9, ,1.4,0.2
4.7,3.2,1.3,0.2
4.6,3.1,1.5,0.2
5,3.6,1.4,0.2

D: for dimension
C: for statistics
S: for substitutes missing values
E: exit
S
sepal.length,sepal.length,petal.length,petal.width
5.1,3.5,1.4,0.2
4.9,3.35,1.4,0.2
4.7,3.2,1.3,0.2
4.6,3.1,1.5,0.2
5,3.6,1.4,0.2

D: for dimension
C: for statistics
S: for substitutes missing values
E: exit
E
```

another tested file (test2.csv) that contain 7 row and three column :

```
fanan@fanan-VirtualBox:~$ pico project
fanan@fanan-VirtualBox:~$ ./project
Enter file name:
test2.csv
sepal.length,sepal.width,petal.length
2.1,5,3.4
2.6,4.5,3
2.9,5.2,3.5
1.9,4.7,3.3
2.2,4.2,4.1
2.5,5.1,3.8
2,4.4,3.6

D: for dimension
C: for statistics
S: for substitutes missing values
E: exit
D
The dimension is 7 x 3
```

```
fanan@fanan-VirtualBox:~$ ./project
Enter file name:
test2.csv
sepal.length,sepal.width,petal.length
2.1,5,3.4
2.6,4.5,3
2.9,5.2,3.5
1.9,4.7,3.3
2.2,4.2,4.1
2.5,5.1,3.8
2,4.4,3.6

D: for dimension
C: for statistics
S: for substitutes missing values
E: exit
C
Min     1.9     4.2     3
Max     2.9     5.2     4.1
Mean    2.31    4.72    3.52
STDEV   .33566563       .35351297       .32837260
```

```
test2.csv
sepal.length,sepal.width,petal.length
2.1,5,3.4
2.6,4.5,3
 ,5.2,3.5
1.9,4.7,3.3
2.2,4.2,4.1
2.5,5.1,3.8
2,4.4,3.6

D: for dimension
C: for statistics
S: for substitutes missing values
E: exit
S
sepal.length,sepal.width,petal.length
2.1,5,3.4
2.6,4.5,3
2.21,5.2,3.5
1.9,4.7,3.3
2.2,4.2,4.1
2.5,5.1,3.8
2,4.4,3.6
```

the third row contains a missed value as below, the program will substitute the missed value by the mean of the column 2,21

test case for calculations in file contains a missed value in second row:

```
fanan@fanan-VirtualBox:~$ ./project
Enter file name:
test.csv
sepal.length,sepal.length,petal.length,petal.width
5.1,3.5,1.4,0.2
4.9, ,1.4,0.2
4.7,3.2,1.3,0.2
4.6,3.1,1.5,0.2
5,3.6,1.4,0.2

D: for dimension
C: for statistics
S: for substitutes missing values
E: exit
C
Min      4.6      3.1      1.3      0.2
Max      5.1      3.6      1.5      0.2
Mean     4.86     3.35     1.40     .20
STDEV    .18547236         .20615528         .06324555         0

D: for dimension
C: for statistics
S: for substitutes missing values
E: exit
E
fanan@fanan-VirtualBox:~$
```

CONCLUSION:

In this project, we learned how to program with a Shell script, we are trained in shell programming and tried to include as many cases as possible.

Appendix :

The code :

```
echo "Min" > stats.txt
echo "Max" >> stats.txt
echo "Mean" >> stats.txt
echo "STDEV" >> stats.txt

dimension()
{
  #number of rows = number of lines - 1
  allrows=$(cat $filename | wc -l)
  rowno=$((allrows-1))
  columno=$(awk -F',' '{print NF}' $filename | uniq)
  echo "The dimension is $rowno x $columno"
}

stat()
{
 allrows=$(cat $filename | wc -l)
 r=$((allrows-1))
 columno=$(awk -F',' '{print NF}' $filename | uniq)
 i=1
 while [ "$i" -le "$columno" ] #loop on columns
 do
   #take one column without the first line in it and sort it
   cut -d',' -f$i $filename | tail -$r | sort -n > temp.txt
   #ignore empty value and decrease rows number
   firstline=$(head -1 temp.txt)
   if [ "$firstline" = " " ]
   then
     r=$((r - 1))
     cat temp.txt | tail -$r > newtemp.txt
     cp newtemp.txt temp.txt
```

```
fi
#first line is the minimum value
min=$(head -1 temp.txt)
echo $min > column$i.txt
#last line is the maximum value
max=$(tail -1 temp.txt)
echo $max >> column$i.txt
#calculate mean
k=1
sum=0
cat temp.txt | tr '\12' ',' > tmp2.txt
while [ "$k" -le "$r" ]
do
  num=$(cat tmp2.txt | cut -d',' -f$k)
  sum="$( bc <<<"$sum + $num" )"
  k=$((k + 1))
done
mean="$( bc <<<"scale=2; $sum / $r" )"
echo $mean >> column$i.txt

#calculate STDEV
m=1
sumDEV=0
while [ "$m" -le "$r" ]
do
  numDEV=$(cat tmp2.txt | cut -d',' -f$m)
  diff="$( bc <<<"$numDEV - $mean" )"
  diff2="$( bc <<<"scale=10; $diff^2" )"
  sumDEV="$( bc <<<"$sumDEV + $diff2" )"
  m=$((m + 1))
done
sd="$( bc <<<"scale=8; $sumDEV / $r" )"
STDEV=$(echo "sqrt($sd)" | bc)
```

```
  echo $STDEV >> column$i.txt
  #recalculate number of rows
  allrows=$(cat $filename | wc -l)
  r=$((allrows-1))
  i=$((i + 1))
 done

#disdlay result
#results of each column saved in separete file
#combine all results in one file
 cp stats.txt result.txt
 j=1
 while [ "$j" -le "$columno" ]
 do
  paste result.txt column$j.txt > finalresult.txt
  cp finalresult.txt result.txt
  j=$((j + 1))
 done
 cat result.txt
}

substitute()
{
 allrows=$(cat $filename | wc -l)
 r=$((allrows-1))
 columno=$(awk -F',' '{print NF}' $filename | uniq)
 i=1
 while [ "$i" -le "$columno" ]
 do
  cut -d',' -f$i $filename | tail -$r | sort -n > temp.txt
  #calculate mean of the column to subsitute it
  k=1
  sum=0
```

```
  flag=0
  cat temp.txt | tr '\12' ',' > tmp2.txt
  while [ "$k" -le "$r" ]
  do
    num=$(cat tmp2.txt | cut -d',' -f$k)
    if [ "$num" = " " ]
    then
      flag=$((flag + 1))
    else
      sum="$( bc <<<"$sum + $num" )"
    fi
    k=$((k + 1))
  done
  r=$((r - flag))
  meanSUB="$( bc <<<"scale=2; $sum / $r" )"
  #check if any value is empty and subsitute mean
  j=1
  while [ "$j" -le "$r" ]
  do
    num2=$(cat tmp2.txt | cut -d',' -f$j)
    if [ "$num2" = " " ]
    then
       sed 's/ /'$meanSUB'/' $filename > temptest.txt
       cp temptest.txt $filename
       cat $filename
    fi
  j=$((j + 1))
  done
  i=$((i + 1))
 done
}

echo "Enter file name: "
```

```
read filename
if ! [[ -f "$filename" ]];then #detecting file is avaliable or not
 echo "No such file"
 exit 2
fi

if ! [[ "${filename: -4}" == ".csv" ]];then #cheaking format of the file
 echo "file extension error"
 exit 2
fi

#display file contant
cat $filename
#display menu
while true
do
 printf "\nD: for dimension
C: for statistics
S: for substitutes missing values
E: exit\n"
 read operation
 case $operation in
   "D") dimension;;
   "C") stat;;
   "S") substitute;;
   "E") exit 0;;
   *) echo "Enter a valid chiose"
 esac
done
```

Reference :

https://linuxconfig.org/how-to-count-number-of-columns-in-csv-file-using-bash-shell#:~:text=Probably%20the%20easiest%20way%20to%20count%20number%20of,number%20of%20characters.%20The%20file%20has%205%20columns.

https://www.unix.com/shell-programming-and-scripting/241323-how-get-column-number-awk.html