ESC: Software Testing Mini Campaign

**Equivalence Class Parititoning & Boundary Value Analysis**

**Scenario 1: Valid/Invalid arguments passed to system**

* Input Space: Arguments passed into system
* **Equivalence Class Paritions**:
  + **Category 1: Number of arguements**
    - **Invalid Partition 1: No argument passed to system**
      * Boundary Value Analysis
        + Minimum**:** No argument passed
    - **Invalid Partition 2: 1 argument passed to system**
      * Boundary Value Analysis
        + Minimum**:** 1 argument passed
    - **Valid Partition 1: 2 arguments passed to system**
      * Boundary Value Analysis
        + Minimum**:** 2 arguments passed
    - **Invalid Partition 3: >2 arguments passed to system**
      * Boundary Value Analysis
        + Just Above Minimum:3 arguments passed; 4 arguments passed, 5 arguments passed..
        + Middle: 10 Arguments
        + Just below maximum: 1000 arguments passed
  + **Category 2: Validity of argument**
    - **Invalid Partition 1: Argument contains illegal characters for a file path**
      * Boundary Value Analysis
        + Just Above Minimum:#%&{}/$!`”:@
        + Middle: my%other#secret{file
        + Just Below Maximum: my#file
    - **Invalid Partition 2: Argument does not contain illegal character but is not a file path (i.e. incorrect file path syntax, has no extension)**
      * Boundary Value Analysis
        + Just Above Minimum:file\c:;
        + Middle: C:\\\\; C:\myfile, myfile;
        + Just Below Maximum: C:\app\myfile
    - **Invalid Partition 3: Argument is a file path, but the file extension is not “.csv”**
      * Boundary Value Analysis
        + Just Above Minimum:C:\app\myfile.txt
        + Middle: C:\app\myfile.c
        + Just Below Maximum: C:\app\myfile.cs
    - **Invalid Partition 4: Argument is a file path with “.csv” file extension, but file path length exceeds 256 characters**
      * Boundary Value Analysis
        + Just Above Minimum:app0123456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990919293949596979899100101102103104105106107108109110111112113114115116117118119.csv
    - **Invalid Partition 5: Argument is a file path with “.csv” file extension, with <= 256** character but file doesn’t exist
      * Boundary Value Analysis
        + Just Above Minimum:myfile.csv
        + Middle: app123456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990919293949596979899100101102103104105106107108109110111112113114115116117118119.csv
        + Just Below Maximum: C:\app\app\23456789\0\2\3\4\5\6\7\8\9202\2223242526272829303\3233343536373839404\4243444546474849505\5253545556575859606\6263646566676869707\7273747576777879808\8283848586878889909\9293949596979899\00\0\02\03\04\05\06\07\08\09\0\2\3\4\5\6\7\8\9\101112.csv
    - **Invalid Partition 6: Argument is a file path to an existing csv file but system is not able to access/read file**
      * **Subclass 1: system has no read permission to file**
        + Boundary Value Analysis

Just Above Minimum:myfile.csv

* + - * **Subclass 2: other processes are writing to or locking file**
        + Boundary Value Analysis

Just Above Minimum:myfile.csv

* + - * **Subclass 3: file is an online-only cloud file**
        + Boundary Value Analysis

Just Above Minimum:myfile.csv

* + - **Valid Partition 1: Argument is a file path to an existing csv file and system is able to read file**
      * Boundary Value Analysis
        + Just Above Minimum:myfile.csv
        + Middle: app123456789101112131415161718192021222324252627282930313233343536373839404142434445464748495051525354555657585960616263646566676869707172737475767778798081828384858687888990919293949596979899100101102103104105106107108109110111112113114115116117118119.csv
        + Just Below Maximum: C:\app\app\23456789\0\2\3\4\5\6\7\8\9202\2223242526272829303\3233343536373839404\4243444546474849505\5253545556575859606\6263646566676869707\7273747576777879808\8283848586878889909\9293949596979899\00\0\02\03\04\05\06\07\08\09\0\2\3\4\5\6\7\8\9\101112.csv

**Scenario 2: Valid/Invalid file format passed**

* Input Space: File’s data content
* **Equivalence Class Paritions:**
  + **Category 1: Comma seperated file content format**
    - **Invalid Partition 1: File is empty and has no content**
      * Boundary Value Analysis
        + Minimum:File is empty
    - **Invalid Partition 2: File is not empty but file is not encoded in UTF-8 format**
      * Boundary Value Analysis
        + Just Above Minimum:File is UTF-32, UTF-16, ASCII, ansi encoding
    - **Invalid Partition 3: File contents are in UTF-8 format, but file only contains whitespaces and/or only newlines**
      * Boundary Value Analysis
        + Just Above Minimum:File contains: “ “; “\n”
        + Middle: File contains: “ \n \n”
        + Just Below Maximum: File CONTAINS: “ \n\n\n \t\t\t “
    - **Valid Parition 1: File contents are in UTF-8 encoding and contains 1 line of non-whitespace/newlines content**
      * Boundary Value Analysis
        + Just Above Minimum: “mycol1”
        + Middle: “mycol1,mycol2”
        + Just Below Maximum: “mycol1,mycol2,mycol3,mycol4”
    - **Invalid Parition 4**: File contents are in UTF-8 encoding and contains more than 1 line of non-whitespace/newlines content but the number of commas (which are not wrapped within double quotes) between lines are inconsistent
      * Boundary Value Analysis
        + Just Above Minimum:

Example 1 File Contains:

Line1: col1,col2,col3

Line2: col1,col2,col3,

* + - * + Middle:

Example 2 File Contains:

Line1: col1,col2,col3

Line2: col1,col2,col3, col4,

* + - * + Just Below Maximum:

Example 3 File Contains:

Line1: col1,col2,col3

Line2: col1,col2,,,col3,col4,col5,col6,,,,

Line3: col1,,

* + - **Valid Partition 2: File contents are in UTF-8 encoding and contains more than 1 line of non-whitespace/newlines content and the number of commas (which are not wrapped within double quotes) between lines are consistent for all lines**
      * Boundary Value Analysis
        + Just Above Minimum:

Example 1 File Contains:

Line1: col1,col2,col3

Line2: col1,col2,col3

* + - * + Middle:

Example 2 File Contains:

Line1: col1,col2,col3,col4,col5

Line2: col1,col2,col3,,

* + - * + Just Below Maximum:

Example 3 File Contains:

Line1: col1,col2,col3

Line2: col1,”col2,this,  
is,all,1,string”,col3

Example 4 File Contains:

Line1: col1,col2,col3

Line2: col1,”col2””,this,  
is,all,1,string”””,col3

**Scenario 3: File comparasions**

* Input Space: 2 Valid CSV Files (that have passed Scenario 1 & 2)
* **Equivalence Class Paritions:**
  + **Category 1: Commas inconsistencies between files**
    - **Invalid Partition 1: Commas inconsistencies between files**
      * Boundary Value Analysis
        + Just Above Minimum:

Example 1 Contains:

File 1 Line1: col1,col2,col3

File 2 Line1: myothercol1, myothercol2, myothercol3,

* + - * + Middle:

Example 2 Contains:

File 1 Line1: col1,col2,col3

File 2 Line1: myothercol1, myothercol2, myothercol3, myothercol4,

* + - * + Just Below Maximum:

Example 3 Contains:

File 1 Line1: col1,col2,col3

File 2 Line1: myothercol1, myothercol2,,,col5,col6,col7,col8,,,,

* + - **Valid Partition 1: Commas consistencies between files**
      * Boundary Value Analysis
        + Just Above Minimum:

Example 1 Contains:

File 1 Line1: col1

File 2 Line1: myothercol1

* + - * + Middle:

Example 2 Contains:

File 1 Line1: col1, col2

File 2 Line1: myothercol1, myothercol2

Example 3 Contains:

Line1: col1,col2,col3,col4,col5

File 2 Line1: col1,col2,col3,,

* + - * + Just Below Maximum:

Example 4 Contains:

Line1: col1,col2,col3

Line2: col1,”col2,this,  
is,all,1,string”,col3

Example 5 Contains:

Line1: col1,col2,col3

Line2: col1,”col2””,this,  
is,all,1,string”””,col3

* **Scenario 4: Cell to Cell comparasions**
  + Input Space: 1 cell of from 1 CSV File and 1 corresponding cell from the other CSV File
  + **Equivalence Class Paritions:**
    - **Category 1: Finding matching unique identifier cell**
    - **Valid Partition 1: No matching ID columns in other file - Missing**
      * Boundary Value Analysis
        + Just Above Minimum:

Example 1 Contains:

File 1:

**IDcol1A**,**IDcol2A**,col3A

IDcol1B,IDcol2B,col3B

File 2:

IDcol1E,IDcol2E,col3E

IDcol1C,IDcol2C,col3C

* + - * + Middle

Example 1 Contains:

File 1:

**IDcol1A**,**IDcol2A**,col3A

IDcol1B,IDcol2B,col3B

File 2:

**IDcol1A**,IDcol2E,col3E

IDcol1C,IDcol2C,col3C

* + - * + Just Below Maximum

Example 1 Contains:

File 1:

**IDcol1A**,**IDcol2A**,col3A

**IDcol1B**,**IDcol2B**,col3B

File 2:

**IDcol1A**, **IDcol2B**,col3E

**IDcol1B**, **IDcol2A**,col3F

* + - **Valid Partition 2: Matching ID columns in other file**
      * Boundary Value Analysis
        + Just Above Minimum:

Example 1 Contains:

File 1:

**IDcol1A,IDcol2A**,col3A

IDcol1B,IDcol2B,col3B

File 2:

**IDcol1A,IDcol2A**,col3E

IDcol1C,IDcol2C,col3C

* + - **Category 2: Find mismatch, once unique column IDs identified**
      * **Invalid Partition 1: False Positive - mismatch when identical data**
        + Boundary Value Analysis

Just Above Minimum:

Example 1 Contains:

File 1:

IDcol1A,IDcol2A,**100**

File 2:

IDcol1A,IDcol2A,**100**

Middle:

Example 2 Contains:

File 1:

IDcol1A,IDcol2A,**100**

File 2:

IDcol1A,IDcol2A,**100.0**

Just Below Maximum:

Example 3 Contains:

File 1:

IDcol1A,IDcol2A,**”100.00”**

File 2:

IDcol1A,IDcol2A,**100**

* + - * **Invalid Partition 2: False Negative - no mismatch when different data**
        + Boundary Value Analysis

Just Above Minimum:

Example 1 Contains:

File 1:

IDcol1A,IDcol2A,**100**

File 2:

IDcol1A,IDcol2A,**300**

Just Below Maximum:

Example 2 Contains:

File 1:

IDcol1A,IDcol2A,**”30.00”**

File 2:

IDcol1A,IDcol2A,”**300.0”**

* + - * **Valid Partition 1: True Negative – No mismatch when no difference**
        + Boundary Value Analysis

Just Above Minimum:

Example 1 Contains:

File 1:

IDcol1A,IDcol2A,**100**

File 2:

IDcol1A,IDcol2A,**100**

Middle:

Example 2 Contains:

File 1:

IDcol1A,IDcol2A,**100**

File 2:

IDcol1A,IDcol2A,**100.0**

Just Below Maximum:

Example 3 Contains:

File 1:

IDcol1A,IDcol2A,**”100.00”**

File 2:

IDcol1A,IDcol2A,**100**

* + - * **Valid Partition 2: True Positive - Mismatch when different data**
        + Boundary Value Analysis

Just Above Minimum:

Example 1 Contains:

File 1:

IDcol1A,IDcol2A,**100**

File 2:

IDcol1A,IDcol2A,**300**

Just Below Maximum:

Example 2 Contains:

File 1:

IDcol1A,IDcol2A,**”30.00”**

File 2:

IDcol1A,IDcol2A,”**300.0”**