

Lab 4 Submission - Srijan Kumar

Data Wrangler Script and Screenshot: CMSC

```
from wrangler import dw
import sys

if(len(sys.argv) < 3):
    sys.exit('Error: Please include an input and output file. Example python script.py
input.csv output.csv')

w = dw.DataWrangler()

# Split data repeatedly on newline into rows
w.add(dw.Split(column=["data"],
    table=0,
    status="active",
    drop=True,
    result="row",
    update=False,
    insert_position="right",
    row=None,
    on="\n",
    before=None,
    after=None,
    ignore_between=None,
    which=1,
    max=0,
    positions=None,
    quote_character=None))

# Split data repeatedly on '|'
w.add(dw.Split(column=["data"],
    table=0,
    status="active",
    drop=True,
    result="column",
    update=False,
    insert_position="right",
    row=None,
    on="\\|",
    before=None,
    after=None,
    ignore_between=None,
    which=1,
    max=0,
    positions=None,
    quote_character="\""))

# Cut on '"'
w.add(dw.Cut(column=[],
    table=0,
    status="active",
```

```
drop=False,  
result="column",  
update=True,  
insert_position="right",  
row=None,  
on="",  
before=None,  
after=None,  
ignore_between=None,  
which=1,  
max=0,  
positions=None))
```

```
# Drop split
```

```
w.add(dw.Drop(column=["split"],  
table=0,  
status="active",  
drop=True))
```

```
# Extract from split1 between '[' and ' FIFA'
```

```
w.add(dw.Extract(column=["split1"],  
table=0,  
status="active",  
drop=False,  
result="column",  
update=False,  
insert_position="right",  
row=None,  
on=".*",  
before=" FIFA",  
after="\\[\\[",  
ignore_between=None,  
which=1,  
max=1,  
positions=None))
```

```
# Drop split1
```

```
w.add(dw.Drop(column=["split1"],  
table=0,  
status="active",  
drop=True))
```

```
# Extract from split2 between '[' and ' FIFA'
```

```
w.add(dw.Extract(column=["split2"],  
table=0,  
status="active",  
drop=False,  
result="column",  
update=False,  
insert_position="right",  
row=None,  
on=".*",  
before=" FIFA",  
after="\\[\\[",
```

```
ignore_between=None,  
which=1,  
max=1,  
positions=None))
```

```
# Drop split2
```

```
w.add(dw.Drop(column=["split2"],  
table=0,  
status="active",  
drop=True))
```

```
# Extract from split3 before '}}'
```

```
w.add(dw.Extract(column=["split3"],  
table=0,  
status="active",  
drop=False,  
result="column",  
update=False,  
insert_position="right",  
row=None,  
on=".*",  
before="}}",  
after=None,  
ignore_between=None,  
which=1,  
max=1,  
positions=None))
```

```
# Drop split3
```

```
w.add(dw.Drop(column=["split3"],  
table=0,  
status="active",  
drop=True))
```

```
# Extract from split4 between '[' and ' FIFA'
```

```
w.add(dw.Extract(column=["split4"],  
table=0,  
status="active",  
drop=False,  
result="column",  
update=False,  
insert_position="right",  
row=None,  
on=".*",  
before=" FIFA",  
after="\\[\\[",  
ignore_between=None,  
which=1,  
max=1,  
positions=None))
```

```
# Drop split4
```

```
w.add(dw.Drop(column=["split4"],  
table=0,
```

```
status="active",
drop=True))
```

```
# Extract from split5 between '[' and ' FIFA'
```

```
w.add(dw.Extract(column="split5",
    table=0,
    status="active",
    drop=False,
    result="column",
    update=False,
    insert_position="right",
    row=None,
    on=".*",
    before=" FIFA",
    after="\\[\\[",
    ignore_between=None,
    which=1,
    max=1,
    positions=None))
```

```
# Drop split5
```

```
w.add(dw.Drop(column="split5",
    table=0,
    status="active",
    drop=True))
```

```
# Extract from split6 between '[' and ' FIFA'
```

```
w.add(dw.Extract(column="split6",
    table=0,
    status="active",
    drop=False,
    result="column",
    update=False,
    insert_position="right",
    row=None,
    on=".*",
    before=" FIFA",
    after="\\[\\[",
    ignore_between=None,
    which=1,
    max=1,
    positions=None))
```

```
# Drop split6
```

```
w.add(dw.Drop(column="split6",
    table=0,
    status="active",
    drop=True))
```

```
# Drop split7
```

```
w.add(dw.Drop(column="split7",
    table=0,
    status="active",
    drop=True))
```

```

# Delete empty rows
w.add(dw.Filter(column=[],
    table=0,
    status="active",
    drop=False,
    row=dw.Row(column=[],
        table=0,
        status="active",
        drop=False,
        conditions=[dw.Empty(column=[],
            table=0,
            status="active",
            drop=False,
            percent_valid=0,
            num_valid=0)])))

# Fill extract2 with values from above
w.add(dw.Fill(column=["extract2"],
    table=0,
    status="active",
    drop=False,
    direction="down",
    method="copy",
    row=None))

# Merge extract2, extract, extract1, extract3... with glue ,
w.add(dw.Merge(column=["extract2","extract","extract1","extract3","extract4","extract5"],
    table=0,
    status="active",
    drop=False,
    result="column",
    update=False,
    insert_position="right",
    row=None,
    glue=","))

# Drop extract, extract1, extract2, extract3...
w.add(dw.Drop(column=["extract","extract1","extract2","extract3","extract4","extract5"],
    table=0,
    status="active",
    drop=True))

# Split merge repeatedly on ','
w.add(dw.Split(column=["merge"],
    table=0,
    status="active",
    drop=True,
    result="column",
    update=False,
    insert_position="right",
    row=None,
    on=",",
    before=None,

```

```

after=None,
ignore_between=None,
which=1,
max="0",
positions=None,
quote_character=None))

```

```

# Delete rows where split8 is null
w.add(dw.Filter(column=[],
    table=0,
    status="active",
    drop=False,
    row=dw.Row(column=[],
        table=0,
        status="active",
        drop=False,
        conditions=[dw.IsNull(column=[],
            table=0,
            status="active",
            drop=False,
            lcol="split8",
            value=None,
            op_str="is null")]))))

```

```

w.apply_to_file(sys.argv[1]).print_csv(sys.argv[2])

```

DataWrangler - Mozilla Firefox

Open Source Software I... datascience-fall14/L... Mera Joota Hai Jap... DataWrangler

vis.stanford.edu/wrangler/app/ data wrangler switch columns

Split Out Extract Edit Fill Translate Drop Merge Wrap Delete Promote Fold Unfold Transpose

Suggestions

rows: 116 prev next

	Course_No	Section_No	Instructor	Waitlist_seats	Open_seats	Total_seats	Day
1	CMSC100	0101	Charles Kassir	0	4	45	M
2	CMSC106	0101	Jianwu Wang	5	0	45	TuTh
3	CMSC131	0101	Evan Golub	0	0	31	MWF
4	CMSC131	0102	Evan Golub	0	0	31	MWF
5	CMSC131	0103	Evan Golub	0	1	31	MWF
6	CMSC131	0104	Evan Golub	0	1	31	MWF
7	CMSC131	0201	Evan Golub	0	1	31	MWF
8	CMSC131	0202	Evan Golub	0	1	31	MWF
9	CMSC131	0203	Evan Golub	0	0	31	MWF
10	CMSC131	0204	Evan Golub	1	0	31	MWF
11	CMSC131	0301	Thomas Reinhardt	0	0	31	MWF
12	CMSC131	0302	Thomas Reinhardt	0	0	31	MWF
13	CMSC131	0303	Thomas Reinhardt	0	0	31	MWF
14	CMSC131	0304	Thomas Reinhardt	0	0	31	MWF
15	CMSC131	0401	Thomas Reinhardt	0	0	31	MWF
16	CMSC131	0402	Thomas Reinhardt	0	7	31	MWF
17	CMSC131	0403	Thomas Reinhardt	0	0	31	MWF
18	CMSC131	0404	Thomas Reinhardt	0	7	31	MWF
19	CMSC132	0101	Laurence Herman	2	0	34	MWF
20	CMSC132	0102	Laurence Herman	0	0	34	MWF
21	CMSC132	0103	Laurence Herman	0	0	34	MWF
22	CMSC132	0104	Laurence Herman	2	0	34	MWF
23	CMSC132	0201	Laurence Herman	0	6	34	MWF
24	CMSC132	0202	Laurence Herman	0	1	34	MWF
25	CMSC132	0203	Laurence Herman	0	0	34	MWF

Script

- Set wrap2 name to Instructor
- Set extract2 name to Waitlist seats
- Set extract1 name to Open seats
- Set extract name to Total seats
- Set split name to Day
- Set split1 name to Time

Data Wrangler Script and Screenshot: World Cup 1

```
from wrangler import dw
import sys

if(len(sys.argv) < 3):
    sys.exit('Error: Please include an input and output file. Example python script.py
input.csv output.csv')

w = dw.DataWrangler()

# Split data repeatedly on newline into rows
w.add(dw.Split(column=["data"],
    table=0,
    status="active",
    drop=True,
    result="row",
    update=False,
    insert_position="right",
    row=None,
    on="\n",
    before=None,
    after=None,
    ignore_between=None,
    which=1,
    max=0,
    positions=None,
    quote_character=None))

# Split data repeatedly on '|'
w.add(dw.Split(column=["data"],
    table=0,
    status="active",
    drop=True,
    result="column",
    update=False,
    insert_position="right",
    row=None,
    on="\\|",
    before=None,
    after=None,
    ignore_between=None,
    which=1,
    max=0,
    positions=None,
    quote_character="\""))

# Cut on '"'
w.add(dw.Cut(column=[],
    table=0,
    status="active",
    drop=False,
    result="column",
```

```
update=True,  
insert_position="right",  
row=None,  
on="",  
before=None,  
after=None,  
ignore_between=None,  
which=1,  
max=0,  
positions=None))
```

```
# Drop split
```

```
w.add(dw.Drop(column=["split"],  
table=0,  
status="active",  
drop=True))
```

```
# Extract from split1 between '[' and ' FIFA'
```

```
w.add(dw.Extract(column=["split1"],  
table=0,  
status="active",  
drop=False,  
result="column",  
update=False,  
insert_position="right",  
row=None,  
on=".*",  
before=" FIFA",  
after="\\[\\[",  
ignore_between=None,  
which=1,  
max=1,  
positions=None))
```

```
# Drop split1
```

```
w.add(dw.Drop(column=["split1"],  
table=0,  
status="active",  
drop=True))
```

```
# Extract from split2 between '[' and ' FIFA'
```

```
w.add(dw.Extract(column=["split2"],  
table=0,  
status="active",  
drop=False,  
result="column",  
update=False,  
insert_position="right",  
row=None,  
on=".*",  
before=" FIFA",  
after="\\[\\[",  
ignore_between=None,  
which=1,
```



```
max=1,  
positions=None))
```

```
# Drop split2
```

```
w.add(dw.Drop(column=["split2"],  
table=0,  
status="active",  
drop=True))
```

```
# Extract from split3 before '}}'
```

```
w.add(dw.Extract(column=["split3"],  
table=0,  
status="active",  
drop=False,  
result="column",  
update=False,  
insert_position="right",  
row=None,  
on=".*",  
before="}}",  
after=None,  
ignore_between=None,  
which=1,  
max=1,  
positions=None))
```

```
# Drop split3
```

```
w.add(dw.Drop(column=["split3"],  
table=0,  
status="active",  
drop=True))
```

```
# Extract from split4 between '[' and ' FIFA'
```

```
w.add(dw.Extract(column=["split4"],  
table=0,  
status="active",  
drop=False,  
result="column",  
update=False,  
insert_position="right",  
row=None,  
on=".*",  
before=" FIFA",  
after="\\[\\[",  
ignore_between=None,  
which=1,  
max=1,  
positions=None))
```

```
# Drop split4
```

```
w.add(dw.Drop(column=["split4"],  
table=0,  
status="active",  
drop=True))
```

```
# Extract from split5 between '[' and ' FIFA'
w.add(dw.Extract(column=["split5"],
    table=0,
    status="active",
    drop=False,
    result="column",
    update=False,
    insert_position="right",
    row=None,
    on=".*",
    before=" FIFA",
    after="\\[\\[",
    ignore_between=None,
    which=1,
    max=1,
    positions=None))
```

```
# Drop split5
w.add(dw.Drop(column=["split5"],
    table=0,
    status="active",
    drop=True))
```

```
# Extract from split6 between '[' and ' FIFA'
w.add(dw.Extract(column=["split6"],
    table=0,
    status="active",
    drop=False,
    result="column",
    update=False,
    insert_position="right",
    row=None,
    on=".*",
    before=" FIFA",
    after="\\[\\[",
    ignore_between=None,
    which=1,
    max=1,
    positions=None))
```

```
# Drop split6
w.add(dw.Drop(column=["split6"],
    table=0,
    status="active",
    drop=True))
```

```
# Drop split7
w.add(dw.Drop(column=["split7"],
    table=0,
    status="active",
    drop=True))
```

```
# Delete empty rows
```

```
w.add(dw.Filter(column=[],
    table=0,
    status="active",
    drop=False,
    row=dw.Row(column=[],
table=0,
status="active",
drop=False,
conditions=[dw.Empty(column=[],
    table=0,
    status="active",
    drop=False,
    percent_valid=0,
    num_valid=0)])))
```

Fill extract2 with values from above

```
w.add(dw.Fill(column=["extract2"],
    table=0,
    status="active",
    drop=False,
    direction="down",
    method="copy",
    row=None))
```

Merge extract2, extract, extract1, extract3... with glue ,

```
w.add(dw.Merge(column=["extract2","extract","extract1","extract3","extract4","extract5"],
    table=0,
    status="active",
    drop=False,
    result="column",
    update=False,
    insert_position="right",
    row=None,
    glue=","))
```

Drop extract, extract1, extract2, extract3...

```
w.add(dw.Drop(column=["extract","extract1","extract2","extract3","extract4","extract5"],
    table=0,
    status="active",
    drop=True))
```

Split merge repeatedly on ','

```
w.add(dw.Split(column=["merge"],
    table=0,
    status="active",
    drop=True,
    result="column",
    update=False,
    insert_position="right",
    row=None,
    on="," ,
    before=None,
    after=None,
    ignore_between=None,
```

```

which=1,
max="0",
positions=None,
quote_character=None))

```

```

# Delete rows where split8 is null

```

```

w.add(dw.Filter(column=[],
    table=0,
    status="active",
    drop=False,
    row=dw.Row(column=[],
        table=0,
        status="active",
        drop=False,
        conditions=[dw.IsNull(column=[],
            table=0,
            status="active",
            drop=False,
            lcol="split8",
            value=None,
            op_str="is null")]))))

```

```

w.apply_to_file(sys.argv[1]).print_csv(sys.argv[2])

```

DataWrangler - Mozilla Firefox

vis.stanford.edu/wrangler/app/

Split Cut Extract Edit Fill Translate Drop Merge Wrap Delete Promote Fold Unfold Transpose

rows: 46 prev next

	split	split8	split9	split10	split11
1	BRA	1958	1970	1994	2002
2	BRA	1950			
3	BRA	1938	1978		
4	BRA	1974	2014		
5	GER	1954	1990	2014	
6	GER	1966	1982	2002	
7	GER	1934	1970	2010	
8	GER	1958			
9	ITA	1934	1938	1982	2006
10	ITA	1970	1994		
11	ITA	1990			
12	ITA	1978			
13	ARG	1978	1986		
14	ARG	1930	1990		
15	URU	1930	1950		
16	URU	1954	1970		
17	FRA	1998			
18	FRA	2006			
19	FRA	1958	1986		
20	FRA	1982			
21	ENG	1966			
22	ENG	1990			
23	ESP	2010			
24	ESP	1950			
25	ESP	1974	1978		

Script

- Fill extract2 with values from above
- Merge extract2, extract, extract1, extract3... with glue ,
- Drop extract, extract1, extract2, extract3...
- Split merge repeatedly on ,
- Delete rows where split8 is null

Data Wrangler Script and Screenshot: World Cup 2

Too difficult to do

UNIX Tools Command: CMSC

```
cat cmsc.txt | grep -v '^$' | awk -v OFS=' ' '/^(CSI|AVW|JMP|ITV|MTH)/ {print $1, $2} !/^(CSI|AVW|JMP|ITV|MTH) / {print $0}' | awk -F' ' '/^Seats/ {print $1} !/^Seats/ {print $0}' | awk -F' ' '/^Seats/ {print $1, $2, $3} !/^Seats/ {print $0}' | awk -v OFS=' ' '/^CMSC/ {course = $1} /^0/ {print course, $0} !/^CMSC|^0|^Seats/ {print $0} /^Seats/ {print $3,$5,$7} ' | awk '/^CMSC/ {print combined; combined = $0} !/^CMSC/ {combined = combined, "$0;}" END {print combined}'
```

UNIX Tools Command: World Cup 1

```
cat worldcup.txt | tail +3 | sed 's/|style="#fff68f"|//g' | sed 's/{ { //g' | sed 's/} } \ /g' | sed 's/} } \ /g' | sed 's/(fb\|\\fb\|)//g' | awk '/^(fb\|\\fb)/ {num = 0; print $0} !^(fb\|\\fb)/ {num = num + 1; if(num < 5) print num, " ", $0}' | sed 's/|fb\|//g' | sed 's/fb\|//g' | awk '/^[A-Z]/ {country = $1} !^[A-Z]/ {print country, $0}' | grep -v 'sort dash' | grep -v '-' | sed 's/|[0-9][0-9][0-9][0-9]/g' | sed 's/FIFA\ World\ Cup//g' | sed 's/(\\|//g' | sed 's/|)]//g' | sed 's/\\|//g' | sed 's/|)]//g' | sed 's/#*//g' | sed 's/\\*//g' | sed 's/\\,//g' | sed 's/[0-9]\\|//g' | awk -v OFS=' ' '// { country = $1; medal = $2; for(i = 4; i <= NF; i++) {print country, $i, medal;}}
```

Python Script: CMSC

```
import unicodcsv
```

```
fw = open("cmssc-python-edited.csv","w")  
writer = unicodcsv.writer(fw, encoding="utf8", lineterminator="\n")  
writer.writerow(("Course No.,""Section  
No.,""Instructor","Seats","Open","Waitlist","Days","Time","Bldg.,""Room No."))
```

```
f = open("cmssc.txt","r")
```

```
l = f.readline().strip()  
while l!="":  
    course = l  
    l = f.readline().strip()  
    while l!="":  
        section = l  
        instr = f.readline().strip()  
        l = f.readline().strip().split(": ")  
        print l  
        totalseats = l[1].split(",")[0]  
        openseats = l[2].split(",")[0]  
        waitlist = l[3].split(" ")[0]  
        l = f.readline().strip().split()  
        day = l[0]  
        time = ' '.join(l[1:])  
        l = f.readline().strip().split()  
        bldg = l[0]  
        room = l[1]  
        writer.writerow((course, section, instr, totalseats, openseats, waitlist, day, time, bldg, room))  
        l = f.readline().strip()  
    l = f.readline().strip()  
f.close()  
fw.close()
```

Python Script: World Cup 1

```
import unicodcsv
import re

fw = open("wc1-python-edited.csv","w")
writer = unicodcsv.writer(fw, encoding="utf8", lineterminator="\n")
writer.writerow(("Team","Year","Position"))

f = open("worldcup.txt","r")
f.readline()

l = f.readline().strip()
while l!="|}":
    l = f.readline().strip()
    country = l.split("{fb|")[1].split("}")[0]
    for i in range(4):
        pos = re.findall("\\d{4}]", f.readline().strip())
        for pos in pos:
            writer.writerow((country, pos[1:-2], i+1))
    f.readline()
    l = f.readline().strip()
fw.close()
f.close()
```

Python Script: World Cup 2

```

import unicodcsv
import re

fw = open("wc2-python-edited.csv","w")
writer = unicodcsv.writer(fw, encoding="utf8", lineterminator="\n")

awards = {}
f = open("worldcup.txt","r")
f.readline()

l = f.readline().strip()
while l!="|}":
    l = f.readline().strip()
    country = l.split("{{fb|"}[1].split("}}")[0]
    countryawards = awards.get(country, {})
    for i in range(4):
        pos = re.findall("\|d{4}]]", f.readline().strip())
        for pos in pos:
            countryawards[pos[1:-2]] = i+1
    awards[country] = countryawards
    f.readline()
    l = f.readline().strip()

f.close()

writer.writerow([i for i in range(1930,2015,4)])
for k in awards:
    x = [k]
    print k, awards[k].keys()
    for i in range(1930, 2015, 4):
        x.append('-' if str(i) not in awards[k].keys() else awards[k][str(i)])
    writer.writerow(x)
fw.close()

```