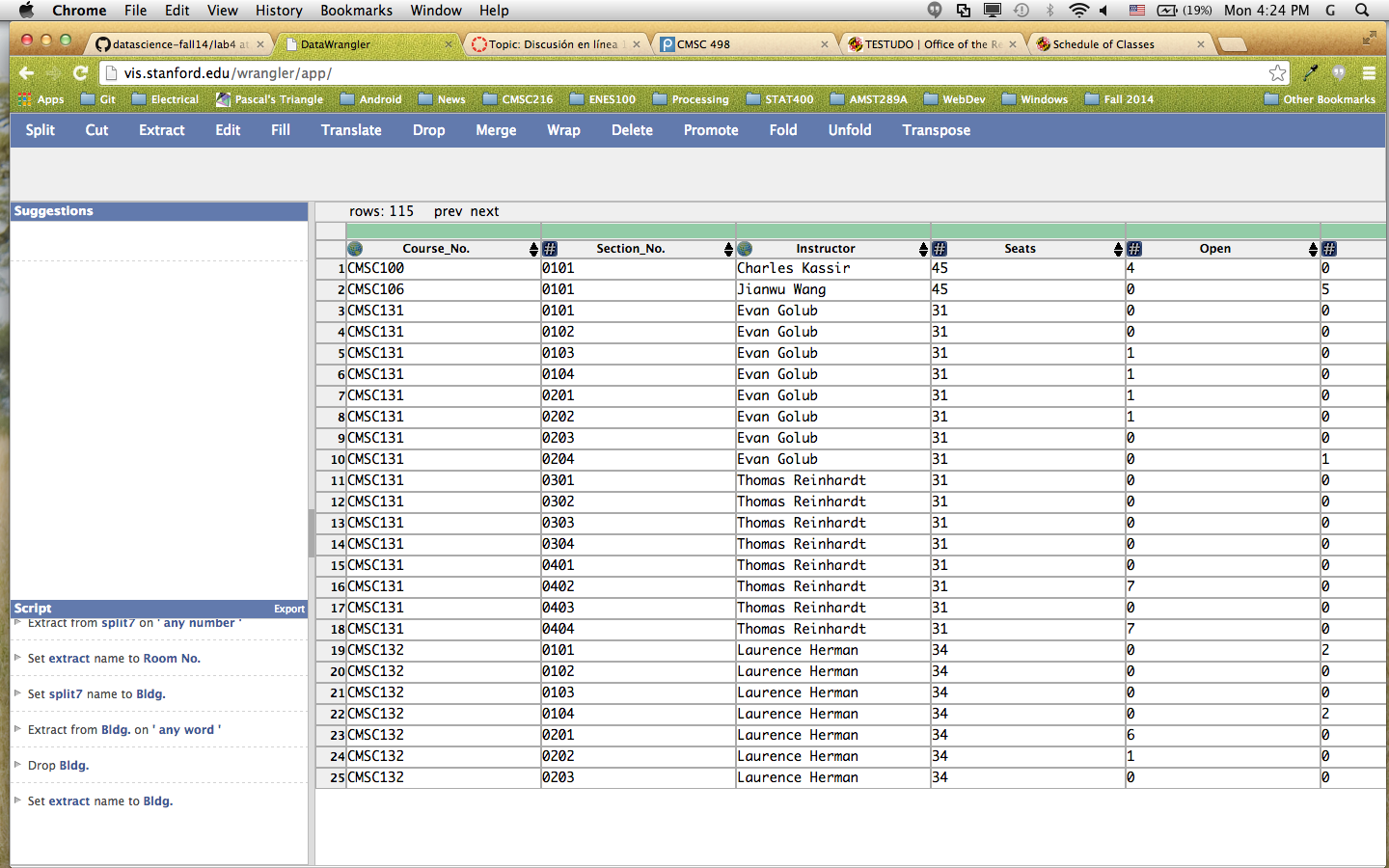
**Lab 4 Submission**

**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=None))

# Set split name to Course\_No.

w.add(dw.SetName(column=["split"],

table=0,

status="active",

drop=True,

names=["Course\_No."],

header\_row=None))

# Set split1 name to Section No.

w.add(dw.SetName(column=["split1"],

table=0,

status="active",

drop=True,

names=["Section No."],

header\_row=None))

# Set split2 name to Instructor

w.add(dw.SetName(column=["split2"],

table=0,

status="active",

drop=True,

names=["Instructor"],

header\_row=None))

# Set split3 name to Seats

w.add(dw.SetName(column=["split3"],

table=0,

status="active",

drop=True,

names=["Seats"],

header\_row=None))

# Set split4 name to Open

w.add(dw.SetName(column=["split4"],

table=0,

status="active",

drop=True,

names=["Open"],

header\_row=None))

# Set split5 name to Waitlist

w.add(dw.SetName(column=["split5"],

table=0,

status="active",

drop=True,

names=["Waitlist"],

header\_row=None))

# Extract from split6 on ' any number : any number any lowercase word - any number : any number any lowercase word '

w.add(dw.Extract(column=["split6"],

table=0,

status="active",

drop=False,

result="column",

update=False,

insert\_position="right",

row=None,

on="\\d+:\\d+[a-z]+ - \\d+:\\d+[a-z]+",

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None))

# Set extract name to Time

w.add(dw.SetName(column=["extract"],

table=0,

status="active",

drop=True,

names=["Time"],

header\_row=None))

# Set split6 name to Days

w.add(dw.SetName(column=["split6"],

table=0,

status="active",

drop=True,

names=["Days"],

header\_row=None))

# Cut from Days on ' any number : any number any lowercase word - any number : any number any lowercase word '

w.add(dw.Cut(column=["Days"],

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="\\d+:\\d+[a-z]+ - \\d+:\\d+[a-z]+",

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None))

# Extract from split7 on ' any number '

w.add(dw.Extract(column=["split7"],

table=0,

status="active",

drop=False,

result="column",

update=False,

insert\_position="right",

row=None,

on="\\d+",

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None))

# Set extract name to Room No.

w.add(dw.SetName(column=["extract"],

table=0,

status="active",

drop=True,

names=["Room No."],

header\_row=None))

# Set split7 name to Bldg.

w.add(dw.SetName(column=["split7"],

table=0,

status="active",

drop=True,

names=["Bldg."],

header\_row=None))

# Extract from Bldg. on ' any word '

w.add(dw.Extract(column=["Bldg."],

table=0,

status="active",

drop=False,

result="column",

update=False,

insert\_position="right",

row=None,

on="[a-zA-Z]+",

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None))

# Drop Bldg.

w.add(dw.Drop(column=["Bldg."],

table=0,

status="active",

drop=True))

# Set extract name to Bldg.

w.add(dw.SetName(column=["extract"],

table=0,

status="active",

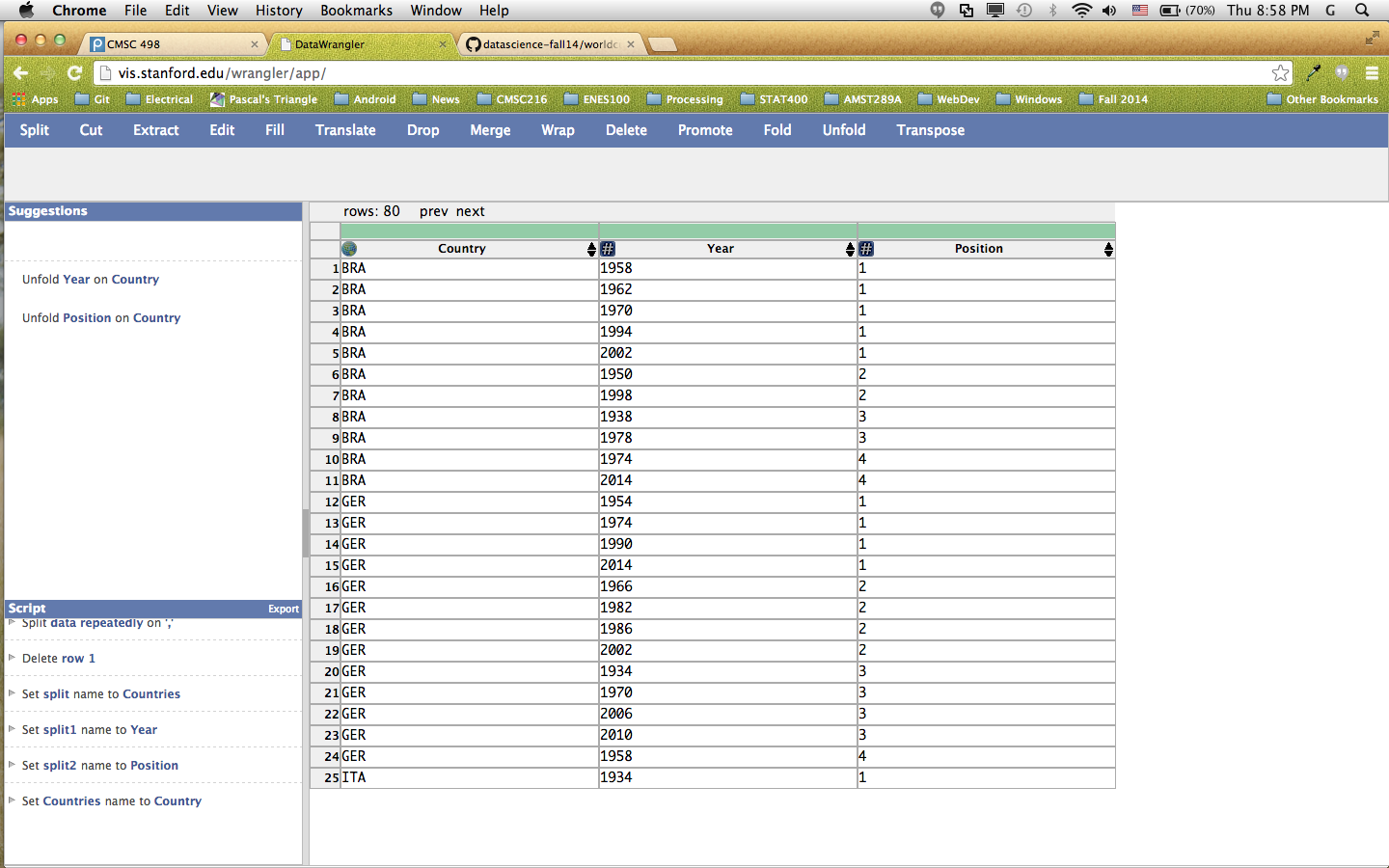
drop=True,

names=["Bldg."],

header\_row=None))

w.apply\_to\_file(sys.argv[1]).print\_csv(sys.argv[2])

**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=None))

# Delete row 1

w.add(dw.Filter(column=[],

table=0,

status="active",

drop=False,

row=dw.Row(column=[],

table=0,

status="active",

drop=False,

conditions=[dw.RowIndex(column=[],

table=0,

status="active",

drop=False,

indices=[0])])))

# Set split name to Countries

w.add(dw.SetName(column=["split"],

table=0,

status="active",

drop=True,

names=["Countries"],

header\_row=None))

# Set split1 name to Year

w.add(dw.SetName(column=["split1"],

table=0,

status="active",

drop=True,

names=["Year"],

header\_row=None))

# Set split2 name to Position

w.add(dw.SetName(column=["split2"],

table=0,

status="active",

drop=True,

names=["Position"],

header\_row=None))

# Set Countries name to Country

w.add(dw.SetName(column=["Countries"],

table=0,

status="active",

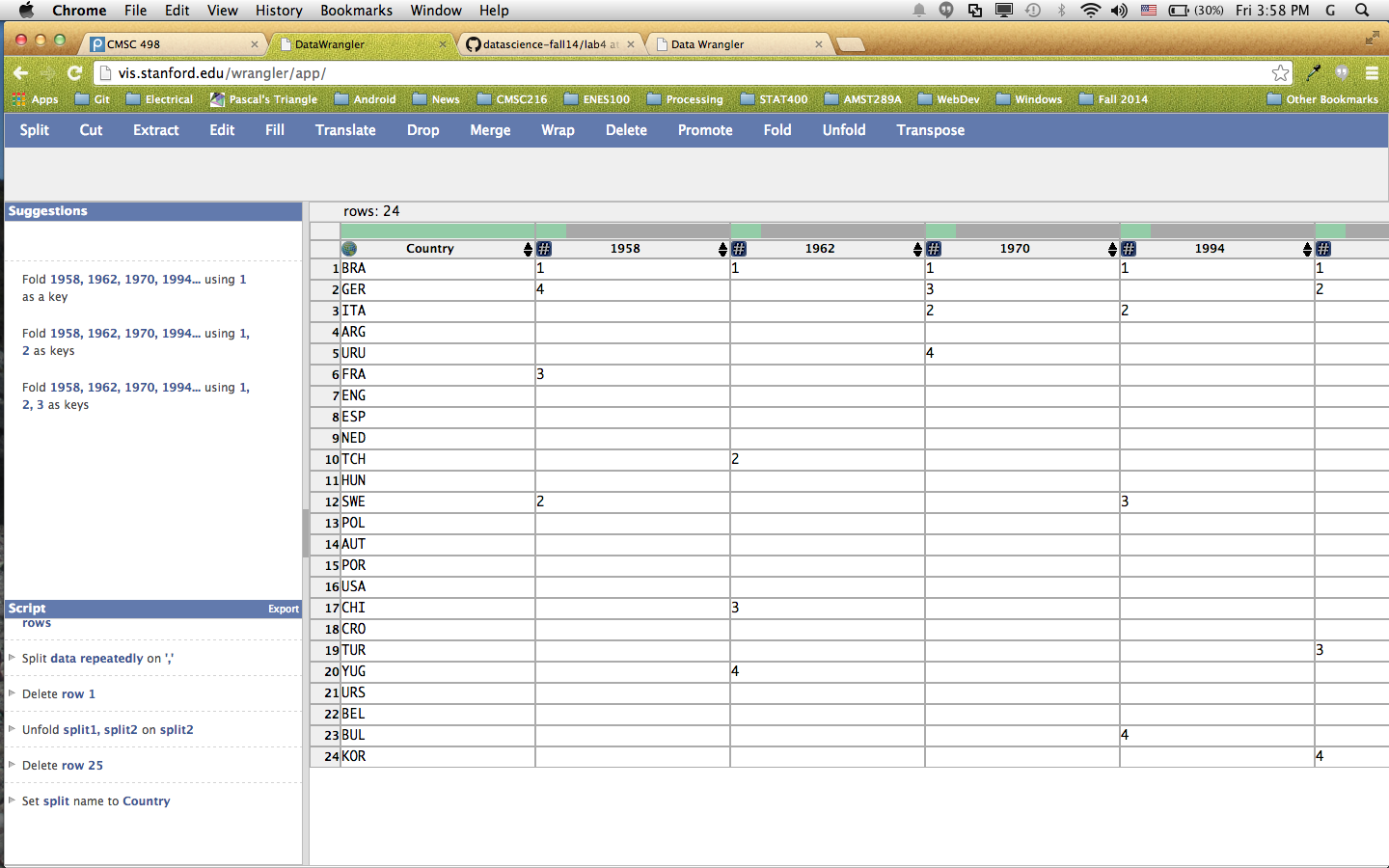
drop=True,

names=["Country"],

header\_row=None))

w.apply\_to\_file(sys.argv[1]).print\_csv(sys.argv[2])

**Data Wrangler Script and Screenshot: World Cup 2**

****

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=None))

# Delete row 1

w.add(dw.Filter(column=[],

table=0,

status="active",

drop=False,

row=dw.Row(column=[],

table=0,

status="active",

drop=False,

conditions=[dw.RowIndex(column=[],

table=0,

status="active",

drop=False,

indices=[0])])))

# Unfold split1, split2 on split2

w.add(dw.Unfold(column=["split1","split2"],

table=0,

status="active",

drop=False,

measure="split2"))

# Delete row 25

w.add(dw.Filter(column=[],

table=0,

status="active",

drop=False,

row=dw.Row(column=[],

table=0,

status="active",

drop=False,

conditions=[dw.RowIndex(column=[],

table=0,

status="active",

drop=False,

indices=[24])])))

# Set split name to Country

w.add(dw.SetName(column=["split"],

table=0,

status="active",

drop=True,

names=["Country"],

header\_row=None))

w.apply\_to\_file(sys.argv[1]).print\_csv(sys.argv[2])

**UNIX Tools Command: CMSC**

awk '/^CMSC[0-9][0-9][0-9]/ { a=$0; } /^[0-9][0-9][0-9][0-9]/ { b=a", "$0; } /^[a-zA-Z:\.]+ [a-zA-Z\-]+/{ c=b", "$0; } /^Seats/ { d=c", "$3" "$5" "$7; } /^[MTuWThF]+ [0-9]+:/ { e=d", "$1", "$2" "$3" "$4; } /^[A-Z][A-Z][A-Z] / { f=e", "$1", "$2; print f}' cmsc.txt | sed 's/)//g'

**UNIX Tools Command: World Cup 1**

tail +2 worldcup.txt | sed 's/|align=center|{{sort dash}}/|0/g; s/style="background:#fff68f"|//g; s/FIFA World Cup|[0-9]\{4\}//g s/|style=white-space:nowrap|//g; s/|.\*||.\*||.\*//g; s/(//g; s/)//g; s/|//g; s/\[\[//g; s/\]\]//g; s/#1\\*//g; s/,//g; s/#2\^//g; s/<sup>#3#<\/sup>//g; s/{{fb//g; s/}//g; s/-//g; /^$/d' | awk '{if ($0 ~ /[A-Z]{3}/) country=$0; } {if ($0 ~ /[A-Z]{3}/) pos=1; } {if ($0 !~ /[A-Z]{3}/) print country", "$2", "pos"\n"country", "$3", "pos"\n"country", "$4", "pos"\n"country", "$5", "pos"\n"country", "$6", "pos"\n"country", "$7","pos++}' | sed '/^[A-Z]\{3\}, ,/d'

**Python Script: CMSC**

import re

with open('cmsc.txt', 'r') as f:

lines = f.readlines()

all\_records = []

sing\_record = []

course = ""

# patterns

total\_p = re.compile('Total: ([0-9]+)')

open\_p = re.compile('Open: ([0-9]+)')

wait\_p = re.compile('Waitlist: ([0-9]+)')

days\_p = re.compile('^([MTuWThF]+)\s[0-9]')

time\_p = re.compile('([0-9]+:[0-9]+[pam]+ - [0-9]+:[0-9]+[pam]+)')

bldg\_p = re.compile('([A-Z]{3})\s+([0-9]{4})')

for x in lines:

line = x.strip()

if re.match('^\s\*$', line):

continue

elif re.match('^CMSC[0-9]{3}', line):

course = line

elif re.match('^[0-9]{4}', line):

sing\_record = []

sing\_record.append(course)

sing\_record.append(line)

elif re.match('^[A-Z]{3}\s+[0-9]{4}', line):

sing\_record.append(bldg\_p.search(line).groups()[0])

sing\_record.append(bldg\_p.search(line).groups()[1])

all\_records.append(sing\_record)

else:

if re.match('^Seats', line):

sing\_record.append(total\_p.search(line).group(1))

sing\_record.append(open\_p.search(line).group(1))

sing\_record.append(wait\_p.search(line).group(1))

elif re.match('^([MTuWThF]+)\s[0-9]+:', line):

sing\_record.append(days\_p.search(line).group(1))

sing\_record.append(time\_p.search(line).group(1))

else:

sing\_record.append(line)

for row in all\_records:

s = ''

for field in row:

s += field + ", "

print s

**Python Script: World Cup 1**

import re

with open('worldcup.txt', 'r') as f:

lines = f.readlines()

country\_p = re.compile('([A-Z]{3})')

n\_titles\_p = re.compile('^\|.\*([0-9])\s\(')

year\_p = re.compile('\[\[([0-9]{4})')

all\_countries = []

row = []

country = ""

num\_titles = 0

years = []

count = 0

for l in lines:

line = l.strip()

# country

if re.match('^\|.\*\{\{fb\|[A-Z]{3}', line):

country = country\_p.search(line).group(1)

row.append(country)

# position and year

elif re.match('^\|.\*[0-9]\s\(', line):

# print line

num\_titles = int(n\_titles\_p.search(line).group(1))

years = year\_p.findall(line)

row.append(num\_titles)

row.append(years)

count+=1

# no position

elif re.match('^\|align', line):

num\_titles = 0

row.append(num\_titles)

row.append([])

count+=1

if (count == 4):

all\_countries.append(row)

row = []

count = 0

output = ''

output += 'country,year,position'

for row in all\_countries:

s = ''

country = ''

pos = 1

for field in row:

if isinstance(field, int):

continue

if isinstance(field, str):

country += field

elif isinstance(field, list):

for year in field:

s += country + "," + str(year) + "," + str(pos)

output += "\n" + s

s = ''

pos += 1

print output

**Python Script: World Cup 2**

import re

import pandas as pd

import numpy as np

with open('worldcup.txt', 'r') as f:

lines = f.readlines()

country\_p = re.compile('([A-Z]{3})')

n\_titles\_p = re.compile('^\|.\*([0-9])\s\(')

year\_p = re.compile('\[\[([0-9]{4})')

all\_countries = []

row = []

country = ""

num\_titles = 0

years = []

count = 0

for l in lines:

line = l.strip()

# country

if re.match('^\|.\*\{\{fb\|[A-Z]{3}', line):

country = country\_p.search(line).group(1)

row.append(country)

# position and year

elif re.match('^\|.\*[0-9]\s\(', line):

# print line

num\_titles = int(n\_titles\_p.search(line).group(1))

years = year\_p.findall(line)

row.append(num\_titles)

row.append(years)

count+=1

# no position

elif re.match('^\|align', line):

num\_titles = 0

row.append(num\_titles)

row.append([])

count+=1

if (count == 4):

all\_countries.append(row)

row = []

count = 0

output = ''

for row in all\_countries:

s = ''

country = ''

pos = 1

for field in row:

if isinstance(field, int):

continue

if isinstance(field, str):

country += field

elif isinstance(field, list):

for year in field:

s += country + "," + str(year) + "," + str(pos)

output += s + "\n"

s = ''

pos += 1

# done part2 here

with open("wc\_output.csv", "w") as csv\_file:

csv\_file.write(output)

df = pd.read\_csv('wc\_output.csv')

print df