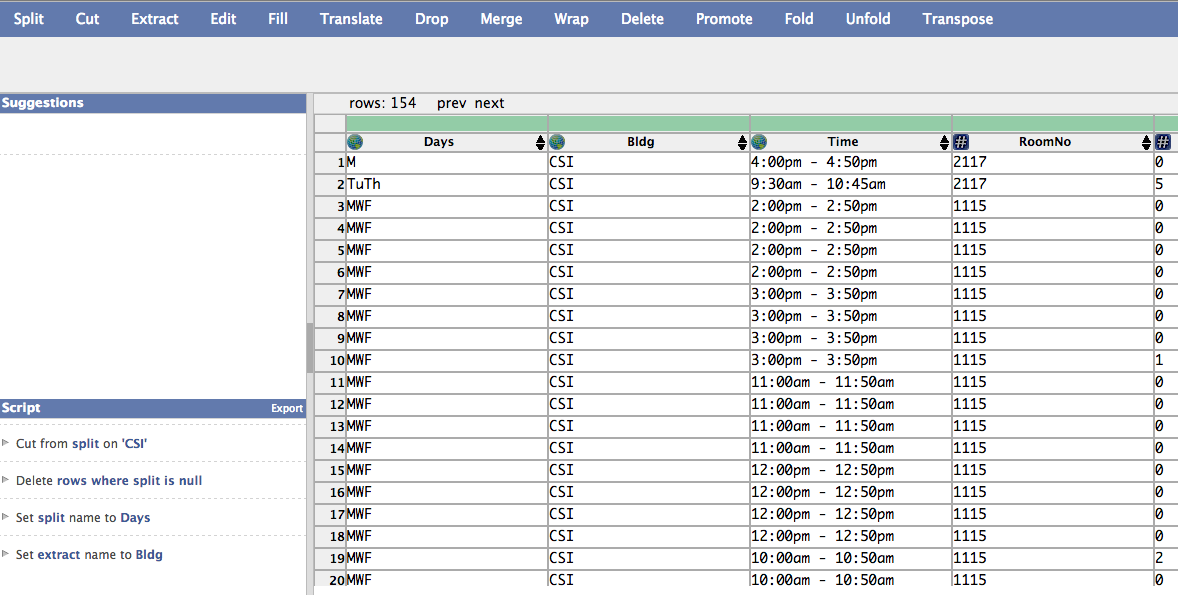
**Lab 4 Submission**

**NOTE: While working with the world cup dataset, data wrangler would always crash after I tried to work much further than some preprocessing. I have included what I was able to complete for parts 1 and 2 of part 1.**

**Data Wrangler Script and Screenshot: CMSC**

Screenshot



CSV output, copied

Days,Bldg,Time,RoomNo,Waitlist,Open,Total,Instructor,SectionNo,CourseNo

M ,CSI,4:00pm - 4:50pm, 2117,0,4,45,Charles Kassir,0101,CMSC100

TuTh ,CSI,9:30am - 10:45am, 2117,5,0,45,Jianwu Wang,0101,CMSC106

MWF ,CSI,2:00pm - 2:50pm, 1115,0,0,31,Evan Golub,0101,CMSC131

MWF ,CSI,2:00pm - 2:50pm, 1115,0,0,31,Evan Golub,0102,CMSC131

MWF ,CSI,2:00pm - 2:50pm, 1115,0,1,31,Evan Golub,0103,CMSC131

MWF ,CSI,2:00pm - 2:50pm, 1115,0,1,31,Evan Golub,0104,CMSC131

MWF ,CSI,3:00pm - 3:50pm, 1115,0,1,31,Evan Golub,0201,CMSC131

MWF ,CSI,3:00pm - 3:50pm, 1115,0,1,31,Evan Golub,0202,CMSC131

MWF ,CSI,3:00pm - 3:50pm, 1115,0,0,31,Evan Golub,0203,CMSC131

MWF ,CSI,3:00pm - 3:50pm, 1115,1,0,31,Evan Golub,0204,CMSC131

MWF ,CSI,11:00am - 11:50am, 1115,0,0,31,Thomas Reinhardt,0301,CMSC131

MWF ,CSI,11:00am - 11:50am, 1115,0,0,31,Thomas Reinhardt,0302,CMSC131

MWF ,CSI,11:00am - 11:50am, 1115,0,0,31,Thomas Reinhardt,0303,CMSC131

MWF ,CSI,11:00am - 11:50am, 1115,0,0,31,Thomas Reinhardt,0304,CMSC131

MWF ,CSI,12:00pm - 12:50pm, 1115,0,0,31,Thomas Reinhardt,0401,CMSC131

MWF ,CSI,12:00pm - 12:50pm, 1115,0,7,31,Thomas Reinhardt,0402,CMSC131

MWF ,CSI,12:00pm - 12:50pm, 1115,0,0,31,Thomas Reinhardt,0403,CMSC131

MWF ,CSI,12:00pm - 12:50pm, 1115,0,7,31,Thomas Reinhardt,0404,CMSC131

MWF ,CSI,10:00am - 10:50am, 1115,2,0,34,Laurence Herman,0101,CMSC132

MWF ,CSI,10:00am - 10:50am, 1115,0,0,34,Laurence Herman,0102,CMSC132

MWF ,CSI,10:00am - 10:50am, 1115,0,0,34,Laurence Herman,0103,CMSC132

MWF ,CSI,10:00am - 10:50am, 1115,2,0,34,Laurence Herman,0104,CMSC132

MWF ,CSI,1:00pm - 1:50pm, 1115,0,6,34,Laurence Herman,0201,CMSC132

MWF ,CSI,1:00pm - 1:50pm, 1115,0,1,34,Laurence Herman,0202,CMSC132

MWF ,CSI,1:00pm - 1:50pm, 1115,0,0,34,Laurence Herman,0203,CMSC132

MWF ,CSI,1:00pm - 1:50pm, 1115,0,0,34,Laurence Herman,0204,CMSC132

MWF ,CSI,2:00pm - 2:50pm, 2117,0,3,30,Laurence Herman,0301,CMSC132

MWF ,CSI,2:00pm - 2:50pm, 2117,1,0,29,Laurence Herman,0302,CMSC132

MWF ,CSI,2:00pm - 2:50pm, 2117,0,8,29,Laurence Herman,0303,CMSC132

-Perez ,CSI,2:00pm - 2:50pm, 1115,0,8,29,Nelson Padua,0101,CMSC216

TuTh ,CSI,9:30am - 10:45am, 1115,0,2,28,Nelson Padua,0101,CMSC216

-Perez ,CSI,9:30am - 10:45am, 1115,0,2,28,Nelson Padua,0102,CMSC216

TuTh ,CSI,9:30am - 10:45am, 1115,0,0,28,Nelson Padua,0102,CMSC216

-Perez ,CSI,9:30am - 10:45am, 1115,0,0,28,Nelson Padua,0103,CMSC216

TuTh ,CSI,9:30am - 10:45am, 1115,1,0,28,Nelson Padua,0103,CMSC216

-Perez ,CSI,9:30am - 10:45am, 1115,1,0,28,Nelson Padua,0104,CMSC216

TuTh ,CSI,9:30am - 10:45am, 1115,0,0,28,Nelson Padua,0104,CMSC216

-Perez ,CSI,9:30am - 10:45am, 1115,0,0,28,Nelson Padua,0201,CMSC216

TuTh ,CSI,11:00am - 12:15pm, 1115,0,0,28,Nelson Padua,0201,CMSC216

-Perez ,CSI,11:00am - 12:15pm, 1115,0,0,28,Nelson Padua,0202,CMSC216

TuTh ,CSI,11:00am - 12:15pm, 1115,1,0,28,Nelson Padua,0202,CMSC216

-Perez ,CSI,11:00am - 12:15pm, 1115,1,0,28,Nelson Padua,0203,CMSC216

TuTh ,CSI,11:00am - 12:15pm, 1115,1,0,28,Nelson Padua,0203,CMSC216

-Perez ,CSI,11:00am - 12:15pm, 1115,1,0,28,Nelson Padua,0204,CMSC216

TuTh ,CSI,11:00am - 12:15pm, 1115,0,0,28,Nelson Padua,0204,CMSC216

-Perez ,CSI,11:00am - 12:15pm, 1115,0,0,28,Nelson Padua,0301,CMSC216

TuTh ,CSI,2:00pm - 3:15pm, 1115,0,0,28,Nelson Padua,0301,CMSC216

-Perez ,CSI,2:00pm - 3:15pm, 1115,0,0,28,Nelson Padua,0302,CMSC216

TuTh ,CSI,2:00pm - 3:15pm, 1115,0,2,28,Nelson Padua,0302,CMSC216

-Perez ,CSI,2:00pm - 3:15pm, 1115,0,2,28,Nelson Padua,0303,CMSC216

TuTh ,CSI,2:00pm - 3:15pm, 1115,0,0,28,Nelson Padua,0303,CMSC216

-Perez ,CSI,2:00pm - 3:15pm, 1115,0,0,28,Nelson Padua,0304,CMSC216

TuTh ,CSI,2:00pm - 3:15pm, 1115,0,0,28,Nelson Padua,0304,CMSC216

TuTh ,CSI,2:00pm - 3:15pm, 2117,4,0,29,Clyde Kruskal,0101,CMSC250

TuTh ,CSI,2:00pm - 3:15pm, 2117,5,0,29,Clyde Kruskal,0102,CMSC250

TuTh ,CSI,2:00pm - 3:15pm, 2117,3,0,29,Clyde Kruskal,0103,CMSC250

TuTh ,CSI,3:30pm - 4:45pm, 1115,5,0,29,Clyde Kruskal,0201,CMSC250

TuTh ,CSI,3:30pm - 4:45pm, 1115,3,0,29,Clyde Kruskal,0202,CMSC250

TuTh ,CSI,3:30pm - 4:45pm, 1115,3,0,29,Clyde Kruskal,0203,CMSC250

TuTh ,CSI,3:30pm - 4:45pm, 1115,0,14,29,Clyde Kruskal,0204,CMSC250

TuTh ,CSI,11:00am - 12:15pm, 3117,1,0,29,Thomas Reinhardt,0301,CMSC250

TuTh ,CSI,11:00am - 12:15pm, 3117,0,1,29,Thomas Reinhardt,0302,CMSC250

TuTh ,CSI,11:00am - 12:15pm, 3117,5,0,29,Thomas Reinhardt,0303,CMSC250

TuTh ,CSI,12:30pm - 1:45pm, 3118,0,0,25,Thomas Goldstein,0101,CMSC250

TuTh ,CSI,11:00am - 12:15pm, 2117,36,0,60,James Reggia,0101,CMSC289

Chau-Wen Tseng ,CSI,11:00am - 12:15pm, 3117,36,0,60,Wen Tseng,0101,CMSC330

TuTh ,CSI,3:30pm - 4:45pm, 3117,0,0,27,Wen Tseng,0101,CMSC330

Chau-Wen Tseng ,CSI,3:30pm - 4:45pm, 3117,0,0,27,Wen Tseng,0102,CMSC330

TuTh ,CSI,3:30pm - 4:45pm, 3117,0,0,27,Wen Tseng,0102,CMSC330

Chau-Wen Tseng ,CSI,3:30pm - 4:45pm, 3117,0,0,27,Wen Tseng,0103,CMSC330

TuTh ,CSI,3:30pm - 4:45pm, 3117,0,1,27,Wen Tseng,0103,CMSC330

Chau-Wen Tseng ,CSI,3:30pm - 4:45pm, 3117,0,1,27,Wen Tseng,0201,CMSC330

TuTh ,CSI,12:30pm - 1:45pm, 3117,0,0,27,Wen Tseng,0201,CMSC330

Chau-Wen Tseng ,CSI,12:30pm - 1:45pm, 3117,0,0,27,Wen Tseng,0202,CMSC330

TuTh ,CSI,12:30pm - 1:45pm, 3117,1,0,27,Wen Tseng,0202,CMSC330

Chau-Wen Tseng ,CSI,12:30pm - 1:45pm, 3117,1,0,27,Wen Tseng,0203,CMSC330

TuTh ,CSI,12:30pm - 1:45pm, 3117,0,13,27,Wen Tseng,0203,CMSC330

Chau-Wen Tseng ,CSI,12:30pm - 1:45pm, 3117,0,13,27,Wen Tseng,0301,CMSC330

TuTh ,CSI,2:00pm - 3:15pm, 3117,0,0,27,Wen Tseng,0301,CMSC330

Chau-Wen Tseng ,CSI,2:00pm - 3:15pm, 3117,0,0,27,Wen Tseng,0302,CMSC330

TuTh ,CSI,2:00pm - 3:15pm, 3117,0,0,27,Wen Tseng,0302,CMSC330

Chau-Wen Tseng ,CSI,2:00pm - 3:15pm, 3117,0,0,27,Wen Tseng,0303,CMSC330

TuTh ,CSI,2:00pm - 3:15pm, 3117,0,1,27,Wen Tseng,0303,CMSC330

MWF ,CSI,10:00am - 10:50am, 3117,2,0,88,Hamid Mahini,0101,CMSC351

MWF ,CSI,11:00am - 11:50am, 3117,4,0,88,Hamid Mahini,0201,CMSC351

MWF ,CSI,3:00pm - 3:50pm, 2117,0,0,88,Hamid Mahini,0301,CMSC351

, Neil Spring,CSI,3:00pm - 3:50pm, 1122,0,0,88,Atif Memon,0101,CMSC396

W ,CSI,1:00pm - 1:50pm, 1122,0,0,25,Atif Memon,0101,CMSC396

AVW ,CSI,4172, 1122,0,0,25,Atif Memon,0101,CMSC396

TuTh ,CSI,3:30pm - 4:45pm, 1122,3,0,45,Michelle Hugue,0101,CMSC411

TuTh ,CSI,2:00pm - 3:15pm, 1122,3,0,45,Michelle Hugue,0201,CMSC411

TuTh ,CSI,11:00am - 12:15pm, 1122,4,0,25,Neil Spring,0101,CMSC412

TuTh ,CSI,11:00am - 12:15pm, 1122,3,0,25,Neil Spring,0102,CMSC412

A.U. Shankar ,CSI,11:00am - 12:15pm, 1122,3,0,25,Neil Spring,0101,CMSC414

TuTh ,CSI,12:30pm - 1:45pm, 1122,6,0,40,Neil Spring,0101,CMSC414

MW ,CSI,3:30pm - 4:45pm, 1122,0,1,50,Elaine Shi,0201,CMSC414

TuTh ,CSI,11:00am - 12:15pm, 3120,0,5,40,Ashok Agrawala,0101,CMSC417

Instructor: TBA,CSI,11:00am - 12:15pm, 2107,0,5,40,Ashok Agrawala,0201,CMSC417

TuTh ,CSI,3:30pm - 4:45pm, 2107,0,27,40,Ashok Agrawala,0201,CMSC417

TuTh ,CSI,12:30pm - 1:45pm, 2120,0,0,40,Hanan Samet,0101,CMSC420

MW ,CSI,3:30pm - 4:45pm, 3120,10,0,40,Michelle Hugue,0201,CMSC420

TuTh ,CSI,2:00pm - 3:15pm, 1121,2,0,50,Donald Perlis,0101,CMSC421

TuTh ,CSI,9:30am - 10:45am, 3117,3,0,46,Donald Perlis,0201,CMSC421

Bravo,CSI,9:30am - 10:45am, 1121,3,0,46,Hector Corrada,0101,CMSC423

MW ,CSI,3:30pm - 4:45pm, 1121,0,20,48,Hector Corrada,0101,CMSC423

TuTh ,CSI,2:00pm - 3:15pm, 3120,2,0,40,Nicholas Roussopoulos,0101,CMSC424

TuTh ,CSI,11:00am - 12:15pm, 2120,0,7,40,John Aloimonos,0101,CMSC426

MW ,CSI,2:00pm - 3:15pm, 1121,0,2,40,Zia Khan,0101,CMSC427

MW ,CSI,3:30pm - 4:45pm, 2107,0,1,40,Jeffrey Foster,0101,CMSC430

TuTh ,CSI,3:30pm - 4:45pm, 1121,15,0,50,Michael Hicks,0101,CMSC433

TuTh ,CSI,9:30am - 10:45am, 1122,8,0,44,Jon Froehlich,0101,CMSC434

TuTh ,CSI,11:00am - 12:15pm, 1121,6,0,44,Vibha Sazawal,0201,CMSC434

TuTh ,CSI,12:30pm - 1:45pm, 1121,4,0,50,James Purtilo,0101,CMSC435

TuTh ,CSI,9:30am - 10:45am, 1121,14,0,50,Adam Porter,0101,CMSC436

TuTh ,CSI,12:30pm - 1:45pm, 2117,11,0,60,Atif Memon,0201,CMSC436

TuTh ,CSI,12:30pm - 1:45pm, 3120,1,0,40,Aravind Srinivasan,0101,CMSC451

TuTh ,CSI,2:00pm - 3:15pm, 3118,0,11,32,William Gasarch,0101,CMSC452

MWF ,CSI,12:00pm - 12:50pm, 1121,3,0,40,Jonathan Katz,0101,CMSC456

TuTh ,CSI,12:30pm - 1:45pm, 2120,2,0,33,Harland Glaz,0101,CMSC460

MTH ,CSI,0304, 2120,2,0,33,Harland Glaz,0101,CMSC460

TuTh ,CSI,2:00pm - 3:15pm, 2120,0,2,25,Changhui Tan,0201,CMSC460

MTH ,CSI,0409, 2120,0,2,25,Changhui Tan,0201,CMSC460

MWF ,CSI,11:00am - 11:50am, 2120,0,11,25,Maria Cameron,0101,CMSC466

MTH ,CSI,1311, 2120,0,11,25,Maria Cameron,0101,CMSC466

TuTh ,CSI,3:30pm - 4:45pm, 2120,0,9,40,Dana Nau,0101,CMSC474

MW ,CSI,11:00am - 12:15pm, 2107,0,7,10,Uzi Vishkin,0101,CMSC498

ITV ,CSI,1100, 2107,0,7,10,Uzi Vishkin,0101,CMSC498

TuTh ,CSI,11:00am - 12:15pm, 2107,0,8,40,Amol Deshpande,0101,CMSC498

Th ,CSI,6:00pm - 8:45pm, 3118,4,0,10,Marshini Chetty,0101,CMSC498

HBK ,CSI,0123, 3118,4,0,10,Marshini Chetty,0101,CMSC498

Horn,CSI,0123, 3118,4,0,10,David Van,0101,CMSC631

TuTh ,CSI,3:30pm - 4:45pm, 3118,0,3,20,David Van,0101,CMSC631

TuTh ,CSI,3:30pm - 4:45pm, 3120,3,0,25,Ramani Duraiswami,0101,CMSC660

TuTh ,CSI,5:00pm - 6:15pm, 4122,0,15,20,Howard Elman,0101,CMSC663

Tobias von Petersdorff ,CSI,5:00pm - 6:15pm, 4122,0,15,20,Howard Elman,0101,CMSC666

TuTh ,CSI,2:00pm - 3:15pm, 4122,0,8,25,Howard Elman,0101,CMSC666

TuTh ,CSI,9:30am - 10:45am, 3120,0,9,40,Mihai Pop,0101,CMSC701

A.U. Shankar ,CSI,9:30am - 10:45am, 2120,0,9,40,Mihai Pop,0101,CMSC712

TuTh ,CSI,2:00pm - 3:30pm, 2120,0,0,20,Mihai Pop,0101,CMSC712

AVW ,CSI,3258, 2120,0,0,20,Mihai Pop,0101,CMSC712

TuTh ,CSI,2:00pm - 3:15pm, 2120,10,0,40,Hal Daume,0101,CMSC723

TuTh ,CSI,11:00am - 12:15pm, 3118,17,0,32,David Jacobs,0101,CMSC733

TuTh ,CSI,9:30am - 10:45am, 2118,0,0,20,Atif Memon,0101,CMSC737

TuTh ,CSI,12:30pm - 1:45pm, 2118,15,0,32,David Mount,0101,CMSC754

F ,CSI,11:00am - 12:50pm, 2117,0,10,40,Jeffrey Foster,0101,CMSC798

TuTh ,CSI,11:00am - 12:15pm, 1115,0,1,20,Peter Keleher,0101,CMSC818

AVW ,CSI,3258, 1115,0,1,20,Peter Keleher,0101,CMSC818

Instructor: TBA,CSI,3258, 1115,0,1,20,Peter Keleher,0101,CMSC818

MW ,CSI,4:00pm - 5:00pm, 1115,0,145,145,Peter Keleher,0101,CMSC818

MW ,CSI,12:30pm - 1:45pm, 2118,0,13,15,Tudor Dumitras,0101,CMSC818

JMP ,CSI,1202, 2118,0,13,15,Tudor Dumitras,0101,CMSC818

TuTh ,CSI,2:00pm - 3:15pm, 2118,0,2,25,Larry Davis,0101,CMSC828

TuTh ,CSI,9:30am - 10:45am, 3118,0,12,25,Venkatramanan Subrahmanian,0101,CMSC828

TuTh ,CSI,3:30pm - 4:45pm, 2118,0,3,20,Mohammad Hajiaghayi,0101,CMSC858

Python Script

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))

# 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)])))

# Extract from data on 'CMSC any number '

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

table=0,

status="active",

drop=False,

result="column",

update=False,

insert\_position="right",

row=None,

on="CMSC\\d+",

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None))

# Fill extract with values from above

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

table=0,

status="active",

drop=False,

direction="down",

method="copy",

row=None))

# Cut from data on 'CMSC any number '

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="CMSC\\d+",

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None))

# Delete rows where data 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="data",

value=None,

op\_str="is null")])))

# Extract from data on '{begin}[0-9]{4}${end}'

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

table=0,

status="active",

drop=False,

result="column",

update=False,

insert\_position="right",

row=None,

on="^[0-9]{4}$$",

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None))

# Cut from data on '{begin}[0-9]{4}{end}'

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="^[0-9]{4}$",

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None))

# Fill extract1 with values from above

w.add(dw.Fill(column=["extract1"],

table=0,

status="active",

drop=False,

direction="down",

method="copy",

row=None))

# Delete rows where data 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="data",

value=None,

op\_str="is null")])))

# Set extract1 name to SectionNo

w.add(dw.SetName(column=["extract1"],

table=0,

status="active",

drop=True,

names=["SectionNo"],

header\_row=None))

# Set extract name to CourseNo

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

table=0,

status="active",

drop=True,

names=["CourseNo"],

header\_row=None))

# Extract from data on '[A-Z] any lowercase word [A-Z] any lowercase word '

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

table=0,

status="active",

drop=False,

result="column",

update=False,

insert\_position="right",

row=None,

on="[A-Z][a-z]+ [A-Z][a-z]+",

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None))

# Fill extract with values from above

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

table=0,

status="active",

drop=False,

direction="down",

method="copy",

row=None))

# Set extract name to Instructor

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

table=0,

status="active",

drop=True,

names=["Instructor"],

header\_row=None))

# Cut from data on '{begin}[A-Z] any lowercase word [A-Z] any lowercase word '

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="^[A-Z][a-z]+ [A-Z][a-z]+",

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None))

# Delete rows where data 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="data",

value=None,

op\_str="is null")])))

# Extract from data on ' any number ' after ': '

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

table=0,

status="active",

drop=False,

result="column",

update=False,

insert\_position="right",

row=None,

on="\\d+",

before=None,

after=": ",

ignore\_between=None,

which=1,

max="0",

positions=None))

# Set extract name to Total

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

table=0,

status="active",

drop=True,

names=["Total"],

header\_row=None))

# Extract from data on ' any number ' after 'Open: '

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

table=0,

status="active",

drop=False,

result="column",

update=False,

insert\_position="right",

row=None,

on="\\d+",

before=None,

after="Open: ",

ignore\_between=None,

which=1,

max=1,

positions=None))

# Extract from data on ' any number ' after 'Waitlist: '

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

table=0,

status="active",

drop=False,

result="column",

update=False,

insert\_position="right",

row=None,

on="\\d+",

before=None,

after="Waitlist: ",

ignore\_between=None,

which=1,

max=1,

positions=None))

# Fill extract2 with values from above

w.add(dw.Fill(column=["extract2"],

table=0,

status="active",

drop=False,

direction="down",

method="copy",

row=None))

# Set extract2 name to Waitlist

w.add(dw.SetName(column=["extract2"],

table=0,

status="active",

drop=True,

names=["Waitlist"],

header\_row=None))

# Fill extract with values from above

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

table=0,

status="active",

drop=False,

direction="down",

method="copy",

row=None))

# Set extract name to Open

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

table=0,

status="active",

drop=True,

names=["Open"],

header\_row=None))

# Fill Total with values from above

w.add(dw.Fill(column=["Total"],

table=0,

status="active",

drop=False,

direction="down",

method="copy",

row=None))

# Split data between 'Seats' and ' '

w.add(dw.Split(column=["data"],

table=0,

status="active",

drop=True,

result="column",

update=False,

insert\_position="right",

row=None,

on=".\*",

before=" ",

after="Seats",

ignore\_between=None,

which=1,

max=1,

positions=None,

quote\_character=None))

# Delete rows where split = 'Seats'

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

table=0,

status="active",

drop=False,

row=dw.Row(column=[],

table=0,

status="active",

drop=False,

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

table=0,

status="active",

drop=False,

lcol="split",

value="Seats",

op\_str="=")])))

# Drop split1

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

table=0,

status="active",

drop=True))

# Split split between 'CSI' and ' any number '

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

table=0,

status="active",

drop=True,

result="column",

update=False,

insert\_position="right",

row=None,

on=".\*",

before=" \\d+",

after="CSI",

ignore\_between=None,

which=1,

max=1,

positions=None,

quote\_character=None))

# Split split2 between ' ' and ' any number '

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

table=0,

status="active",

drop=True,

result="column",

update=False,

insert\_position="right",

row=None,

on=".\*",

before="\\d+",

after=" ",

ignore\_between=None,

which=1,

max=1,

positions=None,

quote\_character=None))

# Fill split3 with values from below

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

table=0,

status="active",

drop=False,

direction="up",

method="copy",

row=None))

# Set split3 name to RoomNo

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

table=0,

status="active",

drop=True,

names=["RoomNo"],

header\_row=None))

# Set split4 name to Time

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

table=0,

status="active",

drop=True,

names=["Time"],

header\_row=None))

# Fill Time with values from above

w.add(dw.Fill(column=["Time"],

table=0,

status="active",

drop=False,

direction="down",

method="copy",

row=None))

# Extract from split on 'CSI'

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

table=0,

status="active",

drop=False,

result="column",

update=False,

insert\_position="right",

row=None,

on="CSI",

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None))

# Fill extract with values from below

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

table=0,

status="active",

drop=False,

direction="up",

method="copy",

row=None))

# Cut from split on 'CSI'

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="CSI",

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None))

# Delete rows where split 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="split",

value=None,

op\_str="is null")])))

# Set split name to Days

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

table=0,

status="active",

drop=True,

names=["Days"],

header\_row=None))

# Set extract name to Bldg

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

table=0,

status="active",

drop=True,

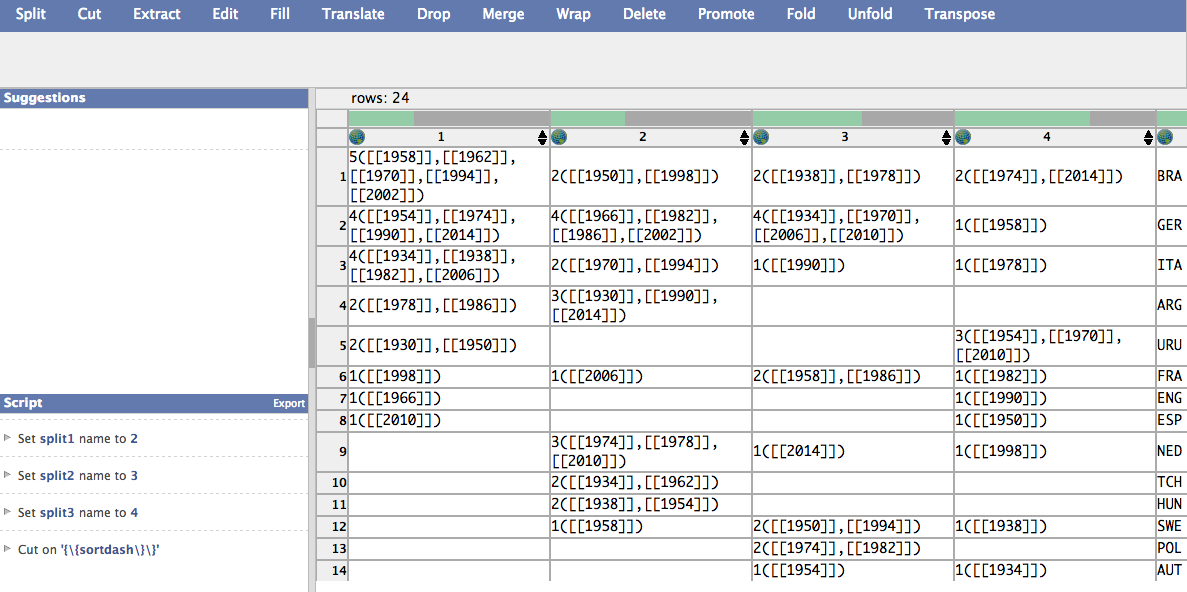
names=["Bldg"],

header\_row=None))

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

**Data Wrangler Script and Screenshot: World Cup 1**

**For the olympics dataset, I was only able to proceed a couple steps after I reached the following point: This was mostly cleaning the dataset, and was common to both**

****

**Copy of CSV output:**

1,2,3,4,Country

5([[1958]],[[1962]],[[1970]],[[1994]],[[2002]])

,2([[1950]],[[1998]]),2([[1938]],[[1978]]),2([[1974]],[[2014]])

,BRA

4([[1954]],[[1974]],[[1990]],[[2014]])

,4([[1966]],[[1982]],[[1986]],[[2002]]),4([[1934]],[[1970]],[[2006]],[[2010]]),1([[1958]])

,GER

4([[1934]],[[1938]],[[1982]],[[2006]])

,2([[1970]],[[1994]]),1([[1990]]),1([[1978]])

,ITA

2([[1978]],[[1986]])

,3([[1930]],[[1990]],[[2014]]),,

,ARG

2([[1930]],[[1950]])

,,,3([[1954]],[[1970]],[[2010]])

,URU

1([[1998]])

,1([[2006]]),2([[1958]],[[1986]]),1([[1982]])

,FRA

1([[1966]])

,,,1([[1990]])

,ENG

1([[2010]])

,,,1([[1950]])

,ESP

,3([[1974]],[[1978]],[[2010]]),1([[2014]]),1([[1998]])

,NED

,2([[1934]],[[1962]]),,,TCH

,2([[1938]],[[1954]]),,

,HUN

,1([[1958]]),2([[1950]],[[1994]]),1([[1938]])

,SWE

,,2([[1974]],[[1982]]),,POL

,,1([[1954]]),1([[1934]]),AUT

,,1([[1966]]),1([[2006]]),POR

,,1([[1930]]),,USA

,,1([[1962]]),,CHI

,,1([[1998]]),,CRO

,,1([[2002]]),,TUR

,,,2([[1930]],[[1962]]),YUG

,,,1([[1966]]),URS

,,,1([[1986]]),BEL

,,,1([[1994]]),BUL

,,,1([[2002]]),KOR

**Python Script:**

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 '|-' into rows

w.add(dw.Split(column=["data"],

table=0,

status="active",

drop=True,

result="row",

update=False,

insert\_position="right",

row=None,

on="\\|-",

before=None,

after=None,

ignore\_between=None,

which=1,

max=0,

positions=None,

quote\_character=None))

# 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))

# 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])])))

# Cut from data on ' |style=background:#fff68f'

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on=" |style=background:#fff68f",

before=None,

after=None,

ignore\_between=None,

which=1,

max="0",

positions=None))

# Extract from on '[A-Z]{3}'

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

table=0,

status="active",

drop=False,

result="column",

update=False,

insert\_position="right",

row=None,

on="[A-Z]{3}",

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None))

# Set extract name to Country

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

table=0,

status="active",

drop=True,

names=["Country"],

header\_row=None))

# Cut from data on '[0-9]{4}[A-Z]{5}[a-z]{4}[A-Z][a-z]{2}|'

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="[0-9]{4}[A-Z]{5}[a-z]{4}[A-Z][a-z]{2}\\|",

before=None,

after=None,

ignore\_between=None,

which=1,

max="0",

positions=None))

# Cut from data on '|\|\{\{fb\|[A-Z]{3}\}\}'

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="\\|\\|\\{\\{fb\\|[A-Z]{3}\\}\\}",

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None))

# Cut from data on '| any number \|\| any number \|\| any number '

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="\\|\\d+\\|\\|\\d+\\|\\|\\d+",

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None))

# Cut from data on '[\[\# any number \|\\*\]\]'

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="\\[\\[\\#\\d+\\|\\\*\\]\\]",

before=None,

after=None,

ignore\_between=None,

which=1,

max="0",

positions=None))

# Cut from data on '|\|'

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

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=1,

positions=None))

# Cut from data on '|align=center'

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="\\|align=center",

before=None,

after=None,

ignore\_between=None,

which=1,

max="0",

positions=None))

# Cut from data on '{begin} '

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

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="1",

positions=None))

# Cut from data on '[\[\# any number \|\^\]\]'

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="\\[\\[\\#\\d+\\|\\^\\]\\]",

before=None,

after=None,

ignore\_between=None,

which=1,

max="0",

positions=None))

# Cut from data on '|style=white-space\:nowrap\|'

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="\\|style=white-space\\:nowrap\\|",

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None))

# Cut from data on '(|)\{\{fb\|[A-Z]{3}\}\}'

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="(\\|)\\{\\{fb\\|[A-Z]{3}\\}\\}",

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None))

# Edit data row 10 to ' {{sortdash}}|2([[1934]],[[1962]])|{{sortdash}}|{{sortdash}} '

w.add(dw.Edit(column=["data"],

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=dw.Row(column=[],

table=0,

status="active",

drop=False,

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

table=0,

status="active",

drop=False,

indices=[9])]),

on=None,

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None,

to="{{sortdash}}|2([[1934]],[[1962]])|{{sortdash}}|{{sortdash}}",

update\_method=None))

# Edit data row 20 to ' {{sortdash}}|{{sortdash}}|{{sortdash}}|2([[1930]],[[1962]])|2{{sortdash}}||{{sortdash}} '

w.add(dw.Edit(column=["data"],

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=dw.Row(column=[],

table=0,

status="active",

drop=False,

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

table=0,

status="active",

drop=False,

indices=[19])]),

on=None,

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None,

to="{{sortdash}}|{{sortdash}}|{{sortdash}}|2([[1930]],[[1962]])|2{{sortdash}}||{{sortdash}}",

update\_method=None))

# Edit data row 21 to ' {{sortdash}}|{{sortdash}}|{{sortdash}}|1([[1966]])|1{{sortdash}}||{{sortdash}} '

w.add(dw.Edit(column=["data"],

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=dw.Row(column=[],

table=0,

status="active",

drop=False,

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

table=0,

status="active",

drop=False,

indices=[20])]),

on=None,

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None,

to="{{sortdash}}|{{sortdash}}|{{sortdash}}|1([[1966]])|1{{sortdash}}||{{sortdash}}",

update\_method=None))

# Cut from data on '|' between '{begin}' and '[|\{'

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="\\|",

before="\\[|\\{",

after="^",

ignore\_between=None,

which=1,

max=1,

positions=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))

# Drop split4

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

table=0,

status="active",

drop=True))

# Drop split5

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

table=0,

status="active",

drop=True))

# Drop split6

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

table=0,

status="active",

drop=True))

# Set split name to 1

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

table=0,

status="active",

drop=True,

names=["1"],

header\_row=None))

# Set split1 name to 2

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

table=0,

status="active",

drop=True,

names=["2"],

header\_row=None))

# Set split2 name to 3

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

table=0,

status="active",

drop=True,

names=["3"],

header\_row=None))

# Set split3 name to 4

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

table=0,

status="active",

drop=True,

names=["4"],

header\_row=None))

# Cut on '{\{sortdash\}\}'

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="\\{\\{sortdash\\}\\}",

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None))

# Cut on newline

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="\n",

before=None,

after=None,

ignore\_between=None,

which=1,

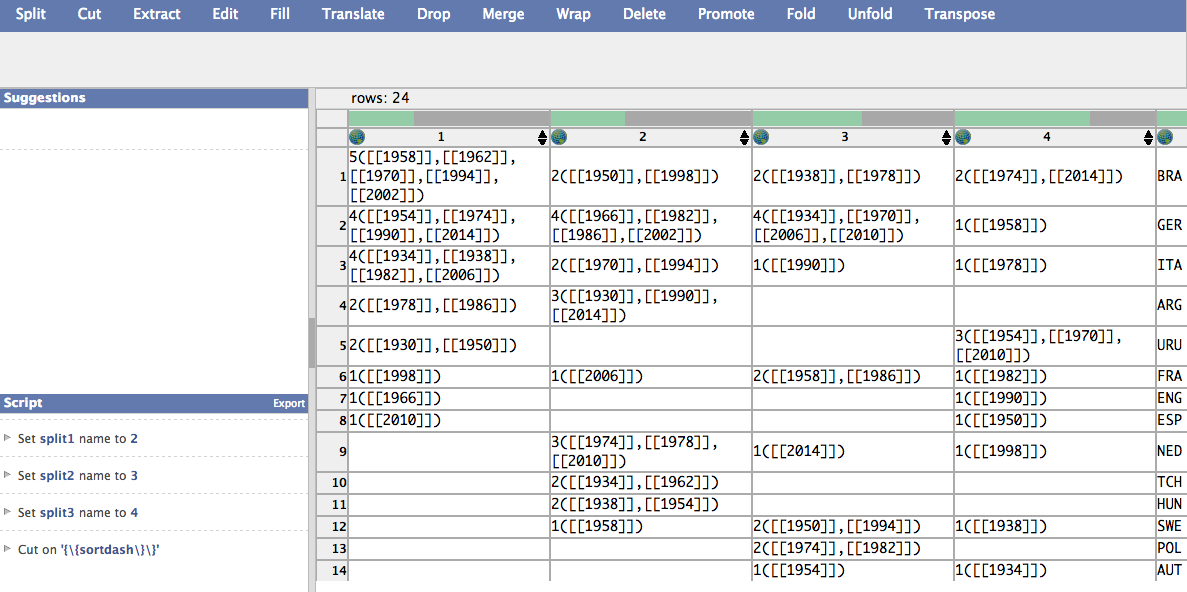
max=1,

positions=None))

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

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

**For the olympics dataset, I was only able to proceed a couple steps after I reached the following point: This was mostly cleaning the dataset, and was common to both**

****

**Copy of CSV output:**

1,2,3,4,Country

5([[1958]],[[1962]],[[1970]],[[1994]],[[2002]])

,2([[1950]],[[1998]]),2([[1938]],[[1978]]),2([[1974]],[[2014]])

,BRA

4([[1954]],[[1974]],[[1990]],[[2014]])

,4([[1966]],[[1982]],[[1986]],[[2002]]),4([[1934]],[[1970]],[[2006]],[[2010]]),1([[1958]])

,GER

4([[1934]],[[1938]],[[1982]],[[2006]])

,2([[1970]],[[1994]]),1([[1990]]),1([[1978]])

,ITA

2([[1978]],[[1986]])

,3([[1930]],[[1990]],[[2014]]),,

,ARG

2([[1930]],[[1950]])

,,,3([[1954]],[[1970]],[[2010]])

,URU

1([[1998]])

,1([[2006]]),2([[1958]],[[1986]]),1([[1982]])

,FRA

1([[1966]])

,,,1([[1990]])

,ENG

1([[2010]])

,,,1([[1950]])

,ESP

,3([[1974]],[[1978]],[[2010]]),1([[2014]]),1([[1998]])

,NED

,2([[1934]],[[1962]]),,,TCH

,2([[1938]],[[1954]]),,

,HUN

,1([[1958]]),2([[1950]],[[1994]]),1([[1938]])

,SWE

,,2([[1974]],[[1982]]),,POL

,,1([[1954]]),1([[1934]]),AUT

,,1([[1966]]),1([[2006]]),POR

,,1([[1930]]),,USA

,,1([[1962]]),,CHI

,,1([[1998]]),,CRO

,,1([[2002]]),,TUR

,,,2([[1930]],[[1962]]),YUG

,,,1([[1966]]),URS

,,,1([[1986]]),BEL

,,,1([[1994]]),BUL

,,,1([[2002]]),KOR

**Python Script:**

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 '|-' into rows

w.add(dw.Split(column=["data"],

table=0,

status="active",

drop=True,

result="row",

update=False,

insert\_position="right",

row=None,

on="\\|-",

before=None,

after=None,

ignore\_between=None,

which=1,

max=0,

positions=None,

quote\_character=None))

# 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))

# 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])])))

# Cut from data on ' |style=background:#fff68f'

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on=" |style=background:#fff68f",

before=None,

after=None,

ignore\_between=None,

which=1,

max="0",

positions=None))

# Extract from on '[A-Z]{3}'

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

table=0,

status="active",

drop=False,

result="column",

update=False,

insert\_position="right",

row=None,

on="[A-Z]{3}",

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None))

# Set extract name to Country

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

table=0,

status="active",

drop=True,

names=["Country"],

header\_row=None))

# Cut from data on '[0-9]{4}[A-Z]{5}[a-z]{4}[A-Z][a-z]{2}|'

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="[0-9]{4}[A-Z]{5}[a-z]{4}[A-Z][a-z]{2}\\|",

before=None,

after=None,

ignore\_between=None,

which=1,

max="0",

positions=None))

# Cut from data on '|\|\{\{fb\|[A-Z]{3}\}\}'

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="\\|\\|\\{\\{fb\\|[A-Z]{3}\\}\\}",

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None))

# Cut from data on '| any number \|\| any number \|\| any number '

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="\\|\\d+\\|\\|\\d+\\|\\|\\d+",

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None))

# Cut from data on '[\[\# any number \|\\*\]\]'

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="\\[\\[\\#\\d+\\|\\\*\\]\\]",

before=None,

after=None,

ignore\_between=None,

which=1,

max="0",

positions=None))

# Cut from data on '|\|'

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

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=1,

positions=None))

# Cut from data on '|align=center'

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="\\|align=center",

before=None,

after=None,

ignore\_between=None,

which=1,

max="0",

positions=None))

# Cut from data on '{begin} '

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

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="1",

positions=None))

# Cut from data on '[\[\# any number \|\^\]\]'

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="\\[\\[\\#\\d+\\|\\^\\]\\]",

before=None,

after=None,

ignore\_between=None,

which=1,

max="0",

positions=None))

# Cut from data on '|style=white-space\:nowrap\|'

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="\\|style=white-space\\:nowrap\\|",

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None))

# Cut from data on '(|)\{\{fb\|[A-Z]{3}\}\}'

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="(\\|)\\{\\{fb\\|[A-Z]{3}\\}\\}",

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None))

# Edit data row 10 to ' {{sortdash}}|2([[1934]],[[1962]])|{{sortdash}}|{{sortdash}} '

w.add(dw.Edit(column=["data"],

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=dw.Row(column=[],

table=0,

status="active",

drop=False,

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

table=0,

status="active",

drop=False,

indices=[9])]),

on=None,

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None,

to="{{sortdash}}|2([[1934]],[[1962]])|{{sortdash}}|{{sortdash}}",

update\_method=None))

# Edit data row 20 to ' {{sortdash}}|{{sortdash}}|{{sortdash}}|2([[1930]],[[1962]])|2{{sortdash}}||{{sortdash}} '

w.add(dw.Edit(column=["data"],

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=dw.Row(column=[],

table=0,

status="active",

drop=False,

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

table=0,

status="active",

drop=False,

indices=[19])]),

on=None,

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None,

to="{{sortdash}}|{{sortdash}}|{{sortdash}}|2([[1930]],[[1962]])|2{{sortdash}}||{{sortdash}}",

update\_method=None))

# Edit data row 21 to ' {{sortdash}}|{{sortdash}}|{{sortdash}}|1([[1966]])|1{{sortdash}}||{{sortdash}} '

w.add(dw.Edit(column=["data"],

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=dw.Row(column=[],

table=0,

status="active",

drop=False,

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

table=0,

status="active",

drop=False,

indices=[20])]),

on=None,

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None,

to="{{sortdash}}|{{sortdash}}|{{sortdash}}|1([[1966]])|1{{sortdash}}||{{sortdash}}",

update\_method=None))

# Cut from data on '|' between '{begin}' and '[|\{'

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="\\|",

before="\\[|\\{",

after="^",

ignore\_between=None,

which=1,

max=1,

positions=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))

# Drop split4

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

table=0,

status="active",

drop=True))

# Drop split5

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

table=0,

status="active",

drop=True))

# Drop split6

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

table=0,

status="active",

drop=True))

# Set split name to 1

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

table=0,

status="active",

drop=True,

names=["1"],

header\_row=None))

# Set split1 name to 2

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

table=0,

status="active",

drop=True,

names=["2"],

header\_row=None))

# Set split2 name to 3

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

table=0,

status="active",

drop=True,

names=["3"],

header\_row=None))

# Set split3 name to 4

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

table=0,

status="active",

drop=True,

names=["4"],

header\_row=None))

# Cut on '{\{sortdash\}\}'

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="\\{\\{sortdash\\}\\}",

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None))

# Cut on newline

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

table=0,

status="active",

drop=False,

result="column",

update=True,

insert\_position="right",

row=None,

on="\n",

before=None,

after=None,

ignore\_between=None,

which=1,

max=1,

positions=None))

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

**UNIX Tools Command: CMSC**

(cat cmsc.txt | awk 'BEGIN {print "Course No., Section No., Instructor, Seats, Open, Waitlist, Days, Time, Bldg., Room No."}

/^CMSC[0-9]{3}$/ {class = $0}

/^[0-9]{4}$/ {section = $0}

/^[A-Z][a-z]+ [A-Z][a-z]+$/ {instructor = $0}

/^Seats \(Total: [0-9]+, Open: [0-9]+, Waitlist: [0-9]+\)$/ {total = $3; open = $5; waitlist = $7}

/^[A-Z]{3} [0-9]{4}$/ {building = $1; room = $2}

/^[A-Z]+ [0-9]:[0-5]{2}[a-z]{2} - [0-9]:[0-5]{2}[a-z]{2}$/ {days = $1; time = $2$3$4}

/^$/{printf class; printf ","; printf section; printf ","; printf instructor; printf ","; printf total; printf open; printf waitlist; printf days; printf ","; printf time; printf ","; printf building; printf ","; printf room; printf "\n"}'

) | tr \) ,

**UNIX Tools Command: World Cup 1**

(cat worldcup.txt | sed '

s/|style="background:#fff68f"//g;

s/|align=center|{{sort dash}}//g;

s/<sup>.\*<\/sup>//g;

s/|style=white-space:nowrap//g;

s/\[\[\#1\|\\*\]\]//g;

s/\[\[\#2\|\^\]\]//g;

s/]])//g;

s/\([0-9][0-9][0-9][0-9]\)/&/g

') | awk -F'\|' 'BEGIN {print "country,year,position"}

/\|\{\{fb\|[A-Z]{3}\}\}/ {country = substr($0,7,3); place = 0}

!/\|\{\{fb\|[A-Z]{3}\}\}/ {place = place + 1}

/[0-9]{4}/ {

for(i=1; i<=NF; i++) {

tmp=match($i, /[0-9]{4}/)

if(tmp) {

printf country;

printf ",";

printf $i;

printf ",";

printf place;

printf "\n"

}

}

}

' | sed '

s/ FIFA World Cup//g;

s/[0-9][0-9][0-9][0-9]\]\], //g;

s/[0-9] //g

' | tr -d '([[' | uniq

**Python Script: CMSC**

import re

cmsc\_input = open("cmsc.txt", 'rb').read().split('\n')

c\_no\_reg = r'^CMSC[0-9][0-9][0-9]$'

s\_no\_reg = r'^[0-9][0-9][0-9][0-9]$'

instructor\_reg = r'^[A-Z][a-z]+ [A-Z][a-z]+$'

course = ""

section = ""

instructor = ""

seats = ""

open\_ = ""

waitlist = ""

days = ""

time = ""

bldg = ""

room = ""

print "Course No., Section No., Instructor, Seats, Open, Waitlist, Days, Time, Bldg., Room No."

for line in cmsc\_input:

if(re.match(c\_no\_reg, line)):

course = line

elif(re.match(s\_no\_reg, line)):

section = line

elif(re.match(instructor\_reg, line)):

instructor = line

elif("Seats" in line):

s = re.findall(r'\d+', line)

seats = s[0]

open\_ = s[1]

waitlist = s[2]

elif("-" in line):

ind = line.find(" ")

days = line[:ind]

time = line[(ind + 1) : ]

elif(" " in line):

result = line.split(" ")

bldg = result[0]

room = result[1]

else:

print course + "," + section + "," + instructor + "," + seats + "," + open\_ + "," + waitlist + "," + days + "," + time + "," + bldg + "," + room

**Python Script: World Cup 1**

import re

wcup\_input = open("worldcup.txt", 'rb').read()

wcup\_input = wcup\_input.replace('|style="background:#fff68f"', '')

wcup\_input = wcup\_input.replace('<sup>[[#3|#]]</sup>', '')

wcup\_input = wcup\_input.replace('|align=center|{{sort dash}}', '')

wcup\_input = wcup\_input.replace('[[#1|\*]]', '')

wcup\_input = wcup\_input.replace('[[#2|^]]', '')

wcup\_input = wcup\_input.replace('|style=white-space:nowrap', '')

place = 0

mapping = {}

curr\_nation = ""

lines = wcup\_input.split('\n')

print "country,year,place"

for line in lines:

if(re.search(r'fb\|[A-Z][A-Z][A-Z]', line)):

place = 0

curr\_nation = line[6:-2]

mapping[curr\_nation] = {}

else:

place += 1

s = re.findall(r'[0-9][0-9][0-9][0-9]', line)

for year in s:

if (year not in mapping[curr\_nation]):

mapping[curr\_nation][year] = place

for nation in mapping:

for year in mapping[nation]:

print nation + "," + year + "," + str(mapping[nation][year])

**Python Script: World Cup 2**

import re

wcup\_input = open("worldcup.txt", 'rb').read()

wcup\_input = wcup\_input.replace('|style="background:#fff68f"', '')

wcup\_input = wcup\_input.replace('<sup>[[#3|#]]</sup>', '')

wcup\_input = wcup\_input.replace('|align=center|{{sort dash}}', '')

wcup\_input = wcup\_input.replace('[[#1|\*]]', '')

wcup\_input = wcup\_input.replace('[[#2|^]]', '')

wcup\_input = wcup\_input.replace('|style=white-space:nowrap', '')

place = 0

mapping = {}

curr\_nation = ""

years = []

lines = wcup\_input.split('\n')

for line in lines:

if(re.search(r'fb\|[A-Z][A-Z][A-Z]', line)):

place = 0

curr\_nation = line[6:-2]

mapping[curr\_nation] = {}

else:

place += 1

s = re.findall(r'[0-9][0-9][0-9][0-9]', line)

for year in s:

if year not in years:

years.append(year)

if (year not in mapping[curr\_nation]):

mapping[curr\_nation][year] = place

list.sort(years)

years\_out = "\t"

for year in years:

years\_out += year + '\t'

print years\_out

for nation in mapping:

curr\_nation = nation + '\t'

for year in years:

if (year in mapping[nation]):

curr\_nation += str(mapping[nation][year])

else:

curr\_nation += "-"

curr\_nation += '\t'

print curr\_nation