**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))

# 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 '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))

# Wrap rows where data starts with '0'

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

table=0,

status="active",

drop=False,

row=dw.Row(column=[],

table=0,

status="active",

drop=False,

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

table=0,

status="active",

drop=False,

lcol="data",

value="0",

op\_str="starts with")])))

# 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])])))

# Drop wrap5

w.add(dw.Drop(column=["wrap5"],

table=0,

status="active",

drop=True))

# Drop wrap3

w.add(dw.Drop(column=["wrap3"],

table=0,

status="active",

drop=True))

# Drop wrap7

w.add(dw.Drop(column=["wrap7"],

table=0,

status="active",

drop=True))

# Drop wrap9

w.add(dw.Drop(column=["wrap9"],

table=0,

status="active",

drop=True))

# Drop wrap10

w.add(dw.Drop(column=["wrap10"],

table=0,

status="active",

drop=True))

# Drop wrap11

w.add(dw.Drop(column=["wrap11"],

table=0,

status="active",

drop=True))

# Set wrap1 name to Course number

w.add(dw.SetName(column=["wrap1"],

table=0,

status="active",

drop=True,

names=["Course number"],

header\_row=None))

# Split wrap8 on ' '

w.add(dw.Split(column=["wrap8"],

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=1,

positions=None,

quote\_character=None))

# Extract from wrap4 between ' Waitlist: ' and ')'

w.add(dw.Extract(column=["wrap4"],

table=0,

status="active",

drop=False,

result="column",

update=False,

insert\_position="right",

row=None,

on=".\*",

before="\\)",

after=" Waitlist: ",

ignore\_between=None,

which=1,

max=1,

positions=None))

# Extract from wrap4 between 'Open: ' and ','

w.add(dw.Extract(column=["wrap4"],

table=0,

status="active",

drop=False,

result="column",

update=False,

insert\_position="right",

row=None,

on=".\*",

before=",",

after="Open: ",

ignore\_between=None,

which=1,

max=1,

positions=None))

# Extract from wrap4 between 'Total: ' and ','

w.add(dw.Extract(column=["wrap4"],

table=0,

status="active",

drop=False,

result="column",

update=False,

insert\_position="right",

row=None,

on=".\*",

before=",",

after="Total: ",

ignore\_between=None,

which=1,

max=1,

positions=None))

# Drop wrap4

w.add(dw.Drop(column=["wrap4"],

table=0,

status="active",

drop=True))

# Split wrap6 between ' any word ' and ' any number '

w.add(dw.Split(column=["wrap6"],

table=0,

status="active",

drop=True,

result="column",

update=False,

insert\_position="right",

row=None,

on=".\*",

before="\\d+",

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

ignore\_between=None,

which=1,

max=1,

positions=None,

quote\_character=None))

# Set wrap name to Section

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

table=0,

status="active",

drop=True,

names=["Section"],

header\_row=None))

# Set Section name to Section No.

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

table=0,

status="active",

drop=True,

names=["Section No."],

header\_row=None))

# Set Course\_number name to Course No.

w.add(dw.SetName(column=["Course\_number"],

table=0,

status="active",

drop=True,

names=["Course No."],

header\_row=None))

# Set wrap2 name to Instructor

w.add(dw.SetName(column=["wrap2"],

table=0,

status="active",

drop=True,

names=["Instructor"],

header\_row=None))

# Set extract2 name to Seats

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

table=0,

status="active",

drop=True,

names=["Seats"],

header\_row=None))

# Set extract1 name to Open

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

table=0,

status="active",

drop=True,

names=["Open"],

header\_row=None))

# Set extract name to Waitlist

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

table=0,

status="active",

drop=True,

names=["Waitlist"],

header\_row=None))

# Set split2 name to Days

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

table=0,

status="active",

drop=True,

names=["Days"],

header\_row=None))

# Set split3 name to Time

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

table=0,

status="active",

drop=True,

names=["Time"],

header\_row=None))

# Set split name to Bldg.

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

table=0,

status="active",

drop=True,

names=[" Bldg."],

header\_row=None))

# Set split1 name to Room No.

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

table=0,

status="active",

drop=True,

names=["Room No."],

header\_row=None))

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

# Drop split

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

table=0,

status="active",

drop=True))

# Delete rows where split1 = '-'

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="split1",

value="-",

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

# Delete rows where split2 = '{{sort dash}}'

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="split2",

value="{{sort dash}}",

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

# Delete rows where split2 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="split2",

value=None,

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

# Extract from split2 before '}}'

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

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))

# Extract from split1 before ' '

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

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))

# Fill extract with values from above

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

table=0,

status="active",

drop=False,

direction="down",

method="copy",

row=None))

# Delete rows where split1 = '{{fb'

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="split1",

value="{{fb",

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

# 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 ' '

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

table=0,

status="active",

drop=False,

result="column",

update=False,

insert\_position="right",

row=None,

on=".\*",

before=" ",

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 between '[\[' and ' '

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

table=0,

status="active",

drop=False,

result="column",

update=False,

insert\_position="right",

row=None,

on=".\*",

before=" ",

after="\\[\\[",

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))

# Drop split6

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

table=0,

status="active",

drop=True))

# Fold extract2, extract3, extract4, extract5... using header as a key

w.add(dw.Fold(column=["extract2","extract3","extract4","extract5","extract6"],

table=0,

status="active",

drop=False,

keys=[-1]))

# Drop fold

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

table=0,

status="active",

drop=True))

# Set value name to Year

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

table=0,

status="active",

drop=True,

names=["Year"],

header\_row=None))

# Set extract1 name to Position

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

table=0,

status="active",

drop=True,

names=["Position"],

header\_row=None))

# Set extract name to County

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

table=0,

status="active",

drop=True,

names=["County"],

header\_row=None))

# Delete rows where Year 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="Year",

value=None,

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

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

# Promote row 1 to header

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

table=0,

status="active",

drop=True,

names=[],

header\_row=0))

# Delete rows where County 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="County",

value=None,

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

# Sort by Year

w.add(dw.Sort(column=["Year"],

table=0,

status="active",

drop=False,

direction=[],

as\_type=[dw.Int(column=[],

table=0,

status="active",

drop=False)]))

# Unfold Year on Position

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

table=0,

status="active",

drop=False,

measure="Position"))

# Sort by County

w.add(dw.Sort(column=["County"],

table=0,

status="active",

drop=False,

direction=[],

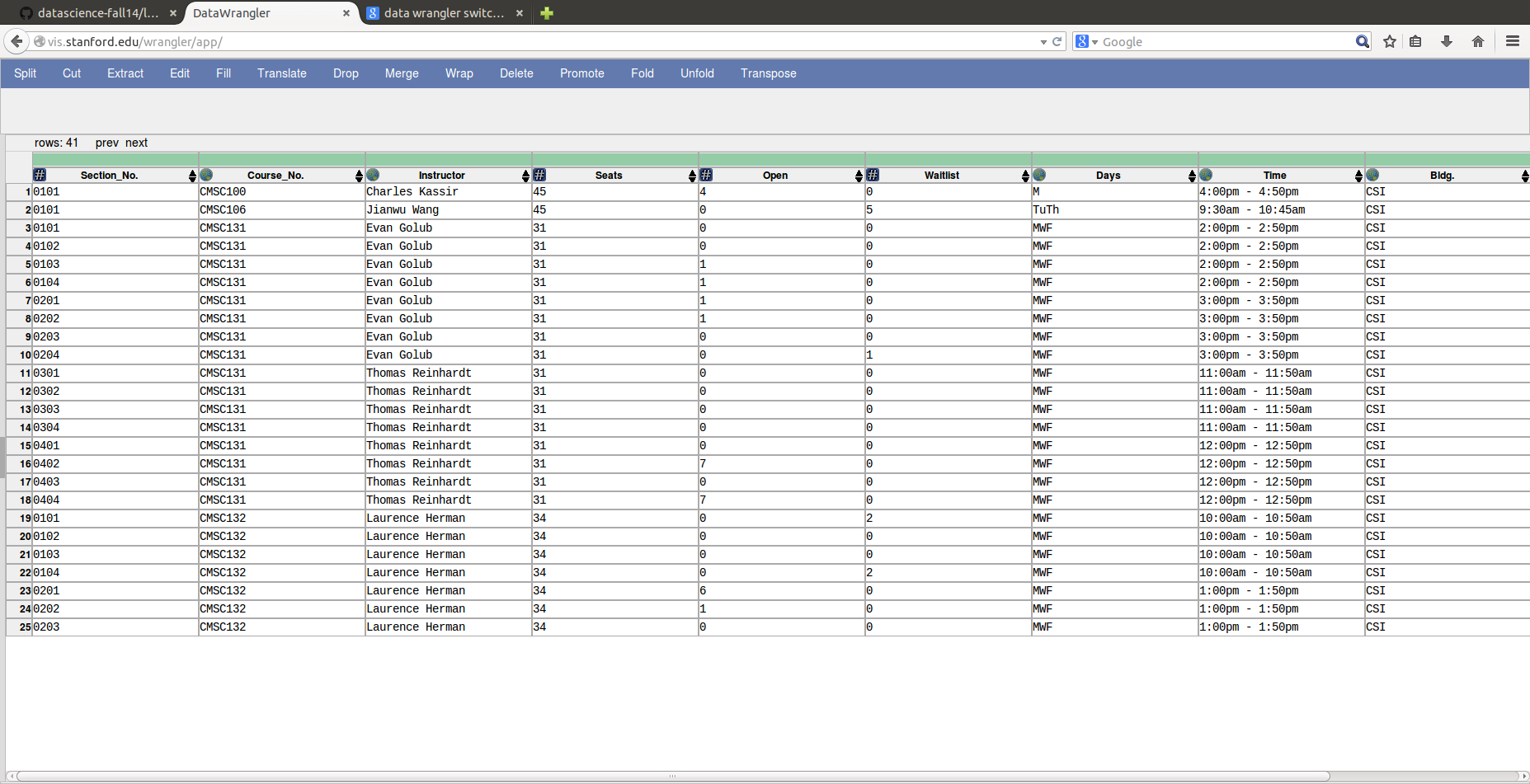
as\_type=[dw.String(column=[],

table=0,

status="active",

drop=False)]))

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])])))**

**# Drop split**

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

**table=0,**

**status="active",**

**drop=True))**

**# Delete rows where split1 = '-'**

**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="split1",**

**value="-",**

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

**# Delete rows where split2 = '{{sort dash}}'**

**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="split2",**

**value="{{sort dash}}",**

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

**# Delete rows where split2 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="split2",**

**value=None,**

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

**# Extract from split2 before '}}'**

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

**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))**

**# Extract from split1 before ' '**

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

**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))**

**# Fill extract with values from above**

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

**table=0,**

**status="active",**

**drop=False,**

**direction="down",**

**method="copy",**

**row=None))**

**# Delete rows where split1 = '{{fb'**

**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="split1",**

**value="{{fb",**

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

**# 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 ' '**

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

**table=0,**

**status="active",**

**drop=False,**

**result="column",**

**update=False,**

**insert\_position="right",**

**row=None,**

**on=".\*",**

**before=" ",**

**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 between '[\[' and ' '**

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

**table=0,**

**status="active",**

**drop=False,**

**result="column",**

**update=False,**

**insert\_position="right",**

**row=None,**

**on=".\*",**

**before=" ",**

**after="\\[\\[",**

**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))**

**# Drop split6**

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

**table=0,**

**status="active",**

**drop=True))**

**# Fold extract2, extract3, extract4, extract5... using header as a key**

**w.add(dw.Fold(column=["extract2","extract3","extract4","extract5","extract6"],**

**table=0,**

**status="active",**

**drop=False,**

**keys=[-1]))**

**# Drop fold**

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

**table=0,**

**status="active",**

**drop=True))**

**# Set value name to Year**

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

**table=0,**

**status="active",**

**drop=True,**

**names=["Year"],**

**header\_row=None))**

**# Set extract1 name to Position**

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

**table=0,**

**status="active",**

**drop=True,**

**names=["Position"],**

**header\_row=None))**

**# Set extract name to County**

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

**table=0,**

**status="active",**

**drop=True,**

**names=["County"],**

**header\_row=None))**

**# Delete rows where Year 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",**

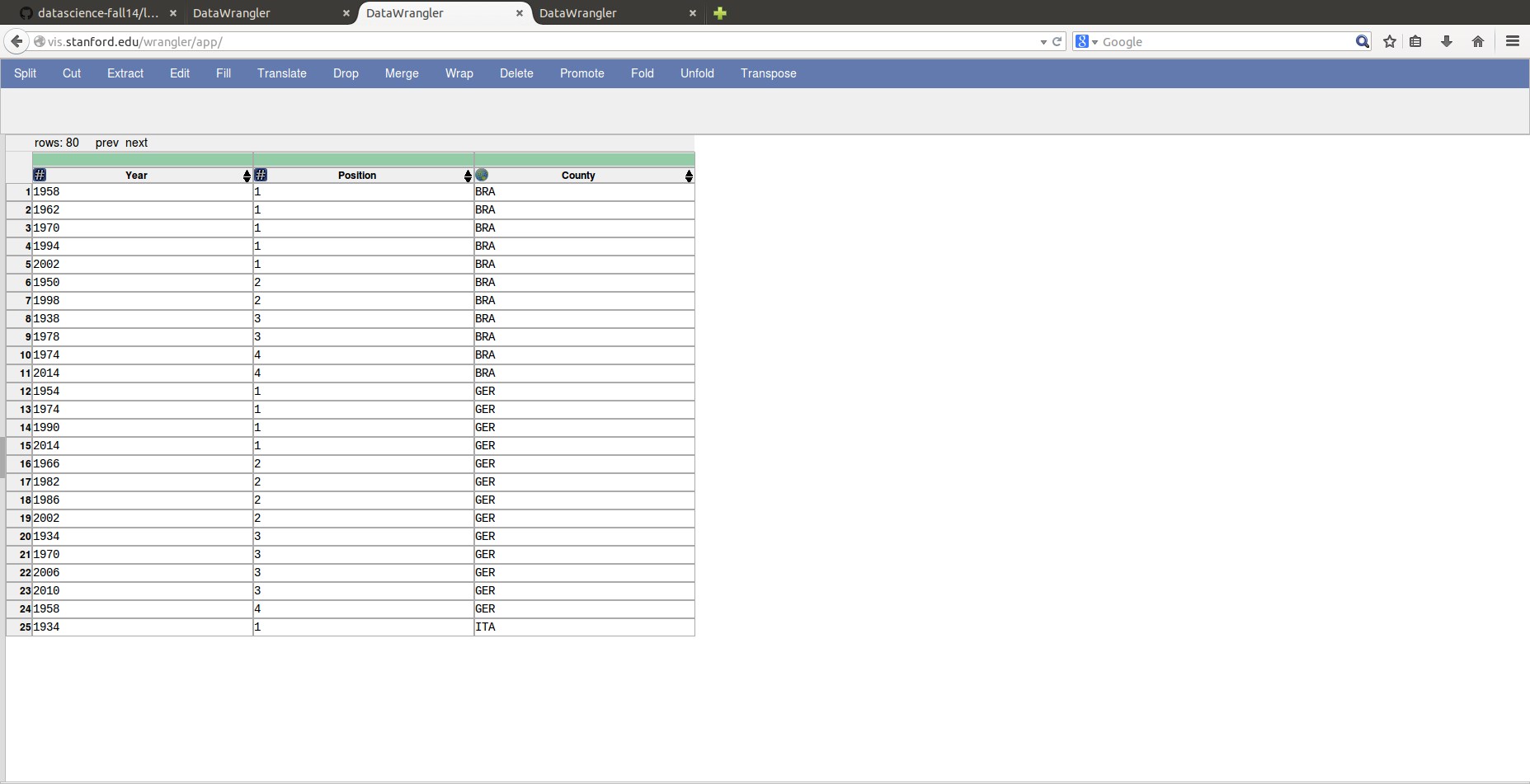
**drop=False,**

**lcol="Year",**

**value=None,**

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

**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))**

**# Promote row 1 to header**

**w.add(dw.SetName(column=[],**

**table=0,**

**status="active",**

**drop=True,**

**names=[],**

**header\_row=0))**

**# Delete rows where County 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="County",**

**value=None,**

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

**# Sort by Year**

**w.add(dw.Sort(column=["Year"],**

**table=0,**

**status="active",**

**drop=False,**

**direction=[],**

**as\_type=[dw.Int(column=[],**

**table=0,**

**status="active",**

**drop=False)]))**

**# Unfold Year on Position**

**w.add(dw.Unfold(column=["Year"],**

**table=0,**

**status="active",**

**drop=False,**

**measure="Position"))**

**# Sort by County**

**w.add(dw.Sort(column=["County"],**

**table=0,**

**status="active",**

**drop=False,**

**direction=[],**

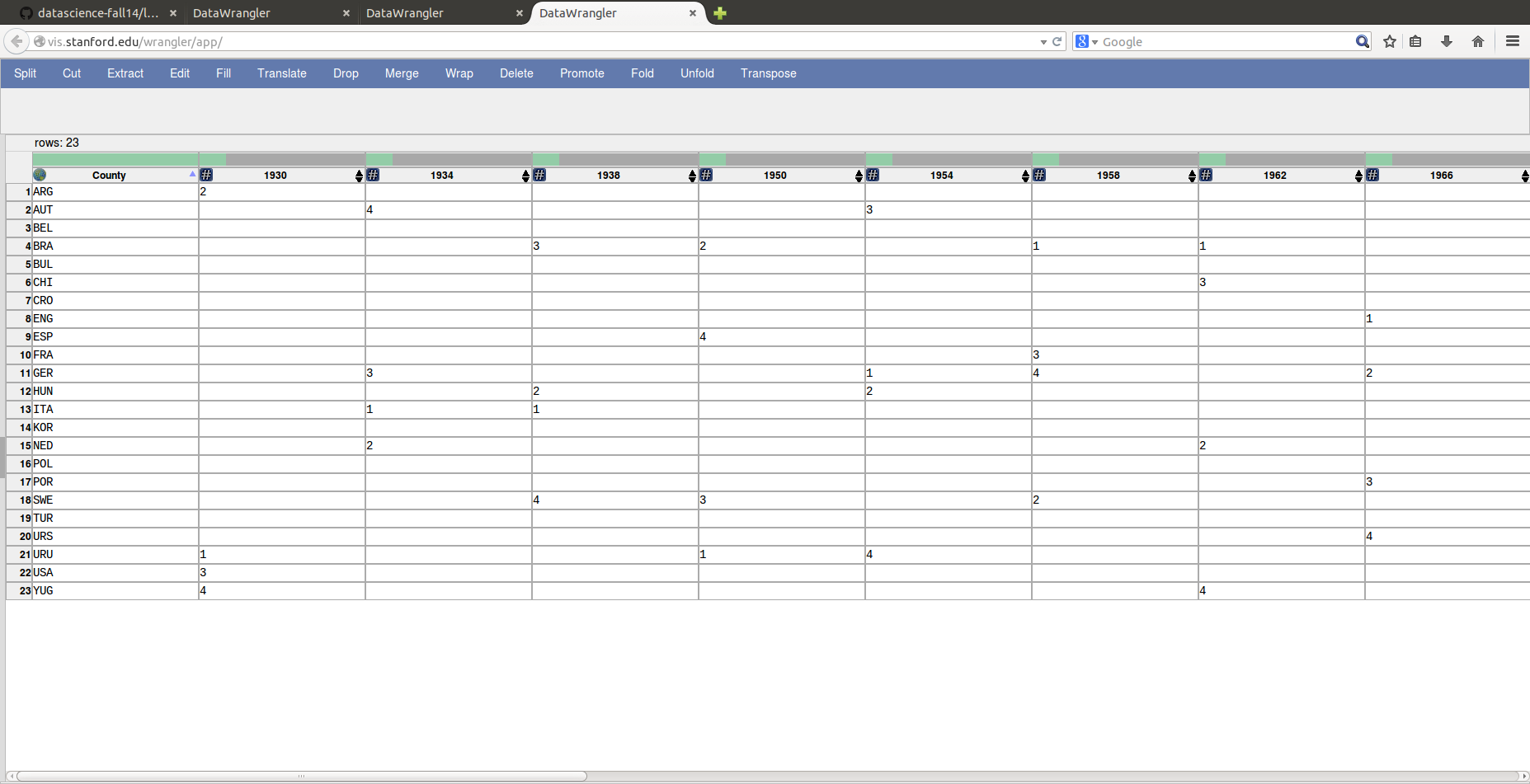
**as\_type=[dw.String(column=[],**

**table=0,**

**status="active",**

**drop=False)]))**

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

****

**UNIX Tools Command: CMSC**

**cat cmsc.txt |sed 's/\([A-Z]\+\) \([[:digit:]]\+\)/\1,\2/'| sed 's/\([A-Za-z]\+\) \([[:digit:]]\+:[[:digit:]]\+[ap]m - [[:digit:]]\+:[[:digit:]]\+[ap]m\)/\1, \2/'|grep -e '^$' -v |awk -F',' 'BEGIN {printf "Course No., Section No., Instructor, Seats, Open, Waitlist, Days, Time, Bldg., Room No."} /^CMSC/ {print d; c=$0} /^[0-9][0-9][0-9][0-9]/ {print d;d=c", "$0;} !/^([0-9])|(CMSC)/ {d=d", "$0} END {print d}'| sed 's/Seats (Total: \([[:digit:]]\+\), Open: \([[:digit:]]\+\), Waitlist: \([[:digit:]]\+\))/\1, \2, \3/'^**

**UNIX Tools Command: World Cup 1**

**cat worldcup.txt |awk '/\|([0-9])|(align)/ {print c $0 ;c=c+1} /^.\*\|{{fb\|[A-Z]\*}}/ {print $1;c=1}' | grep -v 'sort dash'|grep -v '^[0-9]|[0-9]\+|'|sed 's/.\*|{{fb|\([A-Z]\+\)}}.\*/\1/' |sed 's/\([0-9]\+\) FIFA World Cup|[0-9]\+/\1/g'|sed 's/\[\[#1|\\*\]\]//'| awk '/^[A-Z]/ {state = $0} !/^[A-Z]/ {print state, $0}'|sed 's/ \([0-9]\)\(|style="background:#fff68f"\)\*|[0-9]/, \1/'|sed 's/\]\]//g'|sed 's/\[\[//g'|sed 's/(/,/'|sed 's/)//'|awk -F',' '{print $1", " $3"," $2;if($4){print $1"," $4"," $2;if($5){print $1"," $5"," $2;if($6){print $1"," $6"," $2;if($7){print $1"," $7"," $2;if($8){print $1"," $8"," $2;if($9){print $1"," $9"," $2;}}}}}}}'**

**Python Script: CMSC**

**f=open('cmsc.txt','r')**

**content = f.readlines()**

**line=""**

**c=""**

**for x in range(len(content)):**

**if( re.match(r"CMSC",content[x])):**

**c=content[x].strip()+", "**

**m=re.match(r"Seats \(Total: (\d\*), Open: (\d\*), Waitlist: (\d\*)\)",content[x])**

**if(m):**

**line+=m.group(1)+", "+m.group(2)+", "+m.group(3)+", "**

**if(re.match(r"^\w+ \w+\s\*$",content[x])):**

**line+=content[x].strip()+", "**

**m=re.match(r"(\w+) (\d+:\d+[pa]m - \d+:\d+[pa]m)",content[x])**

**if(m):**

**line+=m.group(1)+", "+m.group(2)+", "**

**m=re.match(r"^(\w+)\s+(\d+)$",content[x])**

**if(m):**

**line+=m.group(1)+", "+m.group(2)**

**print line**

**m=re.match(r"^(\d+)$",content[x])**

**if(m):**

**line=c+content[x].strip()+", "**

**Python Script: World Cup 1**

**f=open('worldcup.txt')**

**out=open('wcout.txt','w')**

**content = f.readlines()**

**content.pop(0)**

**c=""**

**p=0**

**out.write('country,year,position \n')**

**for x in range(len(content)):**

**m=re.search(r"{{fb\|([A-Z]+)}}",content[x])**

**if(m):**

**c=m.group(1)**

**p=0**

**years=re.findall(r"(\d\*) FIFA World Cup\|\d\*",content[x])**

**if(f):**

**for y in years:**

**out.write( c+", "+y+", "+str(p)+"\n")**

**p+=1**

**out.close()**

**Python Script: World Cup 2**

db=pd.read\_csv('wcout.txt')

db=db.pivot(index='country',columns='year')

db.replace( 'NaN', '-')