Reference : https://sparkbyexamples.com/pandas/pandas-map-function-explained/

1. Drops columns

df.drop("U\_Keys",axis=1, inplace=True)

# Remove two columns name is 'C' and 'D'

df.drop(['C', 'D'], axis**=**1)

2.Drops Duplicate

cutoff.drop\_duplicates(keep=False,inplace=True)

\*Remove duplicate \*

dd = dfRO.drop\_duplicates(subset='BKH - Booking Ref',keep=False)

=>>False (Drop all duplicate ) >>First (Except first occurance) >>Last(Except Last occurance)

Note : duplicate impacted due to lower and upper case sensivity.

# Remove all duplicate rows

df2 = df.drop\_duplicates(keep=False)

# Delete duplicate rows based on specific columns

df2 = df.drop\_duplicates(subset=["Courses", "Fee"], keep=False)

# Drop duplicate rows in place

df.drop\_duplicates(inplace=True)

# Using DataFrame.apply() and lambda function

df2 = df.apply(lambda x: x.astype(str).str.lower()).drop\_duplicates(subset=['Courses', 'Fee'], keep='first')

3.Replace blank cell with 0

df['Voyage Reference'].fillna(0,inplace=True)

4.map/merge \*\*https://datagy.io/vlookup-in-python-and-pandas-using-map-or-merge/

5.Rename to existing columns

df.rename(columns={'Voyage Reference':'Voyage'},inplace=True)

6.Concate 2 data series & add 2 columns.

df['Uniq'] = df['Voyage'].astype(str)+'|'+df['Stop Location'].astype(str)

7.Sort the pandas series

cutoff['Uniq'] = cutoff['Uniq'].sort\_values(axis=0,ascending=True, inplace=False)

dfRO = dfRO.sort\_values(by='VR - Rollover Date',ascending=True)

8.Vlookup function in python

df = pd.merge(df,cutoff[['Uniq','Cut-Off']],on='Uniq',how='left')

9. Converting pandas.tslib.Timestamp to datetime python

d2 = rw['Cut-Off\_y'].to\_pydatetime() #with date & time

d2 = rw['Cut-Off\_y'].to\_pydatetime().date() # with date

10.Add multiple Blank columns

colmns1 = ['Fault','Split','Original Booking','Door']

df[colmns1] = ""

11. Right function pandas dataseries

df['Job Reference'].str[-1:]

12.Selecting multiple columns in a Pandas dataframe

nRoll = dfRO[['BKH - Booking Ref','VR - Fault',]]

13.Filter all rows that do not contain letters (alpha) in ´pandas´

df = df[df['Col A.'].str.contains('[A-Za-z]')]

14. Filter Alphabet in column excluding blank

df[(df['Split'].str.contains('[A-Za-z]',na=False))

**import** **re**

**>>>** s1.str.contains('PARROT', flags=re.IGNORECASE, regex=**True**)

15. Convert Data type of pandas series String to number

df['Job Reference'] = df['Job Reference'].astype(str)

final\_df['Deliverable Qty'] = pd.to\_numeric(final\_df['Deliverable Qty'])

16.CountIf function in pytho

df['COUNTIFS'] = df.groupby('user\_id').cumcount() + 1

17. Fill empty cells in column with value of other columns

hc['ID'].fillna(hc['First Name'] + hc['Last Name'], inplace=True)

OR

hc.loc[hc["ID"].isnull(),'ID'] = hc["First Name"] + hc["Last Name"]

18.reduce day from date

currDate = date.today()-timedelta(days=3)

19.Convert string to date

cnvtFormat = datetime.strptime(rw['arrDtlocAct'], '%Y-%m-%d %H:%M')

20. DataFrame: if value in a cell, copy value to cells below it

df.UNP[‘action\_amount’].fillna(method='backfill')

ptners1 = ptners1.fillna(method='ffill')

21. convert pandas series to string values,

strVal = ''.join(df)

22.Extract method with list of ..

<https://www.w3resource.com/pandas/series/series-str-extractall.php>

<https://www.shecancode.io/blog/filter-a-pandas-dataframe-by-a-partial-string-or-pattern-in-8-ways>

# 23. [Convert Pandas Series to DateTime in a DataFrame](https://stackoverflow.com/questions/28133018/convert-pandas-series-to-datetime-in-a-dataframe)

df["TimeReviewed"] = pd.to\_datetime(df["TimeReviewed"])

# 24. Python – Find the closest date from a List

<https://www.geeksforgeeks.org/python-find-the-closest-date-from-a-list/>

abs = date values.

res **=** min(test\_date\_list, key**=lambda** sub: abs(sub **-** test\_date))

for id,rw in DB\_report1.iterrows():

    try:

        outPt = ptners1.loc[ptners1['protName'].str.contains(rw['POINT\_NAME'], case=False,na=False) &  ptners1['vesselName'].str.contains(rw['VESSEL\_NAME'], case=False,na=False),'arrDtlocAct']

        outPt = pd.to\_datetime(outPt)

        result = outPt.to\_list()

        # lkVal  =datetime.strptime(rw['ETA\_DATE'], '%Y-%m-%d %H:%M:%S')

        lkVal = rw['ETA\_DATE'].to\_pydatetime()

        res = min(result, key=lambda sub: abs(sub - lkVal))

        DB\_report1.loc[id,'new'] = res

    except:

        continue

25.Conditional formatting in datafram

<https://www.youtube.com/watch?v=qUj7BVZmuiw>

<https://stackoverflow.com/questions/68056855/pandas-style-conditional-formatting-highlight-on-text>

https://queirozf.com/entries/pandas-dataframe-examples-styling-cells-and-conditional-formatting

def color\_negative\_red(value):

    # print(row)

    # value = row.loc['Date\_RangeGap']

    if value == '30':

        color = 'yellow'

    elif value == '15-29':

        color = 'orange'

    elif value == '1-14':

        color = 'green'

    else:

        color = 'white'

    # return ['background-color : %s' % color for r in row]

    return 'background-color : %s' % color

DB\_report1 = DB\_report1.style.applymap(color\_negative\_red,subset='Date\_RangeGap')

* Styler.applymap(func) for element-wise styles.
* Styler.apply(func, axis=0) for column-wise styles.
* Styler.apply(func, axis=1) for row-wise styles.
* Styler.apply(func, axis=None) for tablewise styles.

26. build the regex expression online

https://regex101.com/

# 27. How to Use Like Operator in Pandas DataFrame

df[df['class'].str.contains('sh|rd', regex=True, na=True)]

28. Convert string to integer in Python

# convert the num into string

converted\_num **=** int(num)

# convert the num into string

converted\_num **=** float(num)

29. Drop rows from the dataframe based on certain condition applied on a column

# First filter out those rows which

# does not contain any data

df **=** df.dropna(how **=** 'all')

# Filter all rows for which the player's

# age is greater than or equal to 25

df.drop(df[df['deleted\_Lanes']=='Del'].index,inplace=True)

30.Regex find sub-string from string.

# re.findall(r'(.\*)day','-1 day 5 hours 39 minutes 47 seconds')

31.Read from multiple sheets

import pandas as pd

df = pd.read\_excel('users.xlsx', sheet\_name = [0,1,2])

df = pd.read\_excel('users.xlsx', sheet\_name = ['User\_info','compound'])

df = pd.read\_excel('users.xlsx', sheet\_name = None) # read all sheets

## pd.ExcelFile()

With this approach, we create a **pd.ExcelFile** object to represent the Excel file. We do not need to specify which sheets to read when using this method. Note that the previous **read\_excel()** method returns a dataframe or a dictionary of dataframes; whereas **pd.ExcelFile(**) returns a reference object to the Excel file.

f = pd.ExcelFile('users.xlsx')

>>> f

<pandas.io.excel.\_base.ExcelFile object at 0x00000138DAE66670>

To get data from a sheet, we can use the **parse()** method, and provide the sheet name.

>>> df.parse(sheet\_name = 'User\_info')

## 32. LEFT, RIGHT and MID Functions

https://www.listendata.com/2019/06/python-string-functions.html

|  |  |  |
| --- | --- | --- |
| **Function** | **Description** | **MS EXCEL FUNCTION** |
| mystring[:N] | Extract N number of characters from start of string. | LEFT( ) |
| mystring[-N:] | Extract N number of characters from end of string | RIGHT( ) |
| mystring[X:Y] | Extract characters from middle of string, starting from X position and ends with Y | MID( ) |
| str.split(sep=' ') | Split Strings | - |
| str.replace(old\_substring, new\_substring) | Replace a part of text with different sub-string | REPLACE( ) |
| str.lower() | Convert characters to lowercase | LOWER( ) |
| str.upper() | Convert characters to uppercase | UPPER( ) |
| str.contains('pattern', case=False) | Check if pattern matches  (Pandas Function) | SQL LIKE Operator |
| str.extract(regular\_expression) | Return matched values (Pandas Function) | - |
| str.count('sub\_string') | Count occurence of pattern in string | - |
| str.find( ) | Return position of sub-string or pattern | FIND( ) |
| str.isalnum() | Check whether string consists of only alphanumeric characters | - |
| str.islower() | Check whether characters are all lower case | - |
| str.isupper() | Check whether characters are all upper case | - |
| str.isnumeric() | Check whether string consists of only numeric characters | - |
| str.isspace() | Check whether string consists of only whitespace characters | - |
| len( ) | Calculate length of string | LEN( ) |
| cat( ) | Concatenate Strings (Pandas Function) | CONCATENATE( ) |
| separator.join(str) | Concatenate Strings | CONCATENATE( ) |

df['StateInitial'] = df['state'].str[:2]+ df[‘city'].str[:3]

df\_data['Job\_Bkg\_POL'] = df\_data['Job Reference'].str[:3] + df\_data['Booking POL'].str[:2]

## 33. VLOOKUP in Python

total\_merge = df1.merge(df2, on='id', how='outer', indicator=True)

R1 = total\_merge[total\_merge['\_merge']=='both']

R2 = total\_merge[total\_merge['\_merge']=='left\_only']

R3 = total\_merge[total\_merge['\_merge']=='right\_only']

ValueError: Cannot use name of an existing column for indicator column python

Because you did one merge, you now have a column called \_merge in the your df3. And when you merge again, you cannot create yet another \_merge.

### 33. **Remove spaces Trim**

**Remove spaces in the BEGINNING and END of a string:**

sentence= sentence.strip()

df1['State'] **=** df1['State'].str.strip()

**Remove spaces in the BEGINNING of a string:**

sentence = sentence.lstrip()

df1['State'] **=** df1['State'].str.lstrip()

**Remove spaces in the END of a string:**

sentence= sentence.rstrip()  
df1['State'] **=** df1['State'].str.rstrip()

\*\*\*\*eliminate all the whitespace from a string, on both ends, and in between words.\*\*

>>> import re

>>> re.sub("\s+", # one or more repetition of whitespace

'', # replace with empty string (->remove)

''' hello

... apple

... ''')

### strip()

Python String strip() function will remove leading and trailing whitespaces

[34.](https://gist.github.com/eldala07/c2ef7176b3e43b6f45405d189b5242e1)**[Find, open and reply all to an outlook mail with python](https://gist.github.com/eldala07/c2ef7176b3e43b6f45405d189b5242e1)**

Find last mail containing a certain subject in the 'sent items'

# Open the mail with reply all function

for message in messages\_sent:

    # print('Message class: {}'.format(message.Class))

    # print('Message subject: {}'.format(message.Subject.encode("utf-8")))

    # Message class 43 -> mail

    if message.Class == 43:

        if message\_subject\_to\_find in message.Subject:

            subject\_found = message.Subject

            found = True

            message.ReplyAll().Display()

            break

35. Validate the email id :

import re

valid\_regex = r'\b[A-Za-z0-9.\_%+-]+@[A-Za-z0-9.-]+\.[A-Z|a-z]{2,}\b'

def check(email):

if(re.fullmatch(valid\_regex, email)):

print("This is a valid email")

else:

print("This is an invalid email")

email = "correctemail@gmail.com"

check(email)

36.TkInter design

from tkinter import \*

import tkinter.font as font

gui = Tk(className='Python Examples - Button')

gui.geometry("500x200")

myFont = font.Font(family='Helvetica', size=20, weight='bold')

button = Button(gui, text='My Button', bg='#0052cc', fg='#ffffff')

# apply font to the button label

button['font'] = myFont

# add button to gui window

button.pack()

gui.mainloop()

37. Send mail for another accounts outlook

    outlook = win32com.client.Dispatch("Outlook.Application")

    oacctuse = None

    for oac in outlook.Session.Accounts.\_dispobj\_:

        if oac.DisplayName == frm:

            oacctuse = oac

            break

38.Create folder and write notepad

    target\_fldr = output\_dir/str(Subj)

    target\_fldr.mkdir(parents=True,exist\_ok=True)

    Path(target\_fldr/"Email\_body.txt").write\_text(str(body))

39. Save & rename file the attachment in outlook

   attachment.SaveAsFile(os.path.join(outputDir,attachment.FileName))

                        f1 = os.path.join(outputDir,attachment.FileName)

                        f2 = os.path.join(outputDir,newName)

                        os.rename(f1,f2)

40. Multiple columns combine

df['FullName'] = df[['First\_Name', 'Last\_Name']].apply(lambda x: '\_'.join(x), axis=1)

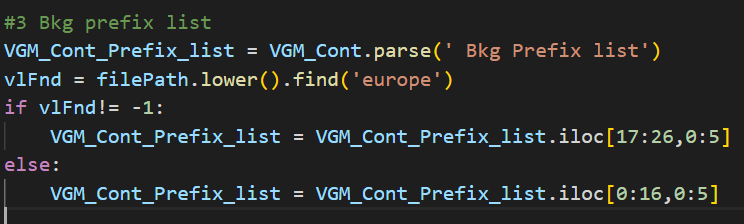
41. Replace NA with value

df\_4['VGM contact'] = df\_4['VGM contact'].fillna(";")

42.Drop duplicate the data based on multiple columns.

newdf = df1.drop\_duplicates(subset = ['order\_id', 'customer\_id'], keep = 'last').reset\_index(drop = True)

43. Find Funtion with row column range



44. askFile dialog

from tkinter import filedialog

def browseFoldr():

    fldr = filedialog.askdirectory()

    return fldr

45. Python Pandas Error tokenizing data – csv

df = pandas.read\_csv(filepath, sep='delimiter', header=None)

# 46. pandas.Series.str.split

**Series.str.split(*pat=None*, *n=- 1*, *expand=False*, *\**, *regex=None*)**

# 47. Concate dataframe with identical columns

dfCom = pd.concat([dfCom,dfN])

# 48. Pivot table : https://pandas.pydata.org/docs/user\_guide/reshaping.html

 = dfCom.pivot(index='Stock Name',columns='Date\_Based',values='RS')

# 49. [how to sort pandas dataframe from one column](https://stackoverflow.com/questions/37787698/how-to-sort-pandas-dataframe-from-one-column)

final\_df = df.sort\_values(by=['2'], ascending=False)

50. Convert string to date

pd.to\_datetime(final\_df.loc[id,'Date'])

51.Sort date in dataframe

        final\_df['Date'] = pd.to\_datetime(final\_df['Date'])

        final\_df.sort\_values(by='Date', ascending=False,inplace=True)

52. clean or drop dataframe data

final\_df.drop(final\_df.index[:],inplace=True)

**53.** ValueError: Index contains duplicate entries, cannot reshape

**df.pivot\_table(index='team', columns='position', values='points', aggfunc='sum')**

# 54. [Convert column of date objects in Pandas DataFrame to strings](https://stackoverflow.com/questions/19738169/convert-column-of-date-objects-in-pandas-dataframe-to-strings)

dfComNSE['Date'] = dfComNSE['Date'].dt.strftime('%d-%b-%Y')

55. Highlight the cell values

def color\_negative\_red(val):

    color = 'red' if val < 0 else 'green'

    return 'color: %s' % color

pvTbl = pvTbl.style.applymap(color\_negative\_red)

56. Cell color

<https://www.blog.pythonlibrary.org/2021/08/11/styling-excel-cells-with-openpyxl-and-python/>

57. Fille the sheet cells

def background\_colors(path):

    wb = load\_workbook('Result.xlsx')

    sheet = wb.active

    rw\_range = sheet.max\_row

    col\_range = sheet.max\_column

    for cl in range(2,(col\_range+1)):

        for rw in range(4,(rw\_range+1)):

            if sheet.cell(rw,cl).value <0:

                sheet.cell(rw,cl).fill = PatternFill(start\_color='FF0000', end\_color='FF0000', fill\_type="solid")

            else:

                sheet.cell(rw,cl).fill = PatternFill(start\_color='00B050', end\_color='00B050', fill\_type="solid")

    wb.save(path)

background\_colors("Result.xlsx")

58. google translate

https://medium.com/analytics-vidhya/exploring-google-cloud-translate-api-for-machine-translation-in-python-ef60c123fc37

59. read multiple sheets in excel

bkID = pd.ExcelFile('Bookler\_Emails\_List.xlsx')

bkID.parse('SheetName')

60. Rename with Reset index

nTbl.reset\_index().rename(columns={'Job Reference':'Booking','Container Number':'Container','Booking POL':'POL','Voyage':'Voyage Ref','Cut Off Local Date':'VGM Cut-off','Remaining time before cut off':'Day remaining to send VGM'},inplace=True)

61. Outlook Html body content replace with html table

mailT['Body11'].fillna("",inplace=True)

mail.HTMLBody = "{0}".format(mailT.to\_html(header=False,index=False,justify='left',border='0'))

mail.HTMLBody = mail.HTMLBody.replace("{0}",nTbl.to\_html())

mail.Display()

62. Remove file from folder

os.remove('c:\\Users\\ssc.achauhan\\Desktop\\Projects\\VGM\_\\Excel\\chaser2.csv')

63. String contains multiple substring

exlChanelN['POL\_N'].str.contains('JP|TW|SG')

64. Read XLSB/Binary File in Pandas Python

import pandas as pd

dfcluster = pd.read\_excel('c:/xml/baseline/distribucion.xlsb', sheet\_name='Cluster', index\_col=0, engine='pyxlsb')

65.Read PDF and merge

**from** PyPDF2 **import** PdfFileMerger

#Create and instance of PdfFileMerger() class

merger = PdfFileMerger()

#Create a list with file names

pdf\_files = ['pdf\_files/sample\_page1.pdf', 'pdf\_files/sample\_page2.pdf']

#Iterate over the list of file names

**for** pdf\_file **in** pdf\_files:

#Append PDF files

merger.append(pdf\_file)

#Write out the merged PDF

merger.write("merged\_2\_pages.pdf")

merger.close()

66. Pandas Series.str.startswith()

bool\_series **=** data["College"].str.startswith(search, na **=** False)

67. Extract email from string

import pandas as pd

import re as re

pd.set\_option('display.max\_columns', 10)

df = pd.DataFrame({

'name\_email': ['Alberto Franco af@gmail.com','Gino Mcneill gm@yahoo.com','Ryan Parkes rp@abc.io', 'Eesha Hinton', 'Gino Mcneill gm@github.com']

})

print("Original DataFrame:")

print(df)

def find\_email(text):

email = re.findall(r'[\w\.-]+@[\w\.-]+',str(text))

return ",".join(email)

df['email']=df['name\_email'].apply(lambda x: find\_email(x))

print("\Extracting email from dataframe columns:")

print(df)

68. Select columns basis on index

# Using iloc[] to select column by Index

df2 = df.iloc[:,[1,3,4]] # Select columns by Index

df2 = df.iloc[:,1:4] # Select between indexes 1 and 4 (2,3,4)

df2 = df.iloc[:,2:] # Select From 3rd to end

df2 = df.iloc[:,:2] # Select First Two Columns

Copy

69. select rows basis on index

result **=** df.iloc[2]

# to select multiple rows

result **=** df.iloc[[2, 3, 5]]

70. Get index of Dataframe

dd = df.index.to\_list()

71. Get 09 as month

"{:02d}".format(tday.month-1)

72. Get name of month

import calendar

for month\_idx in range(1, 13):

print (calendar.month\_name[month\_idx])

print (calendar.month\_abbr[month\_idx])

print ("")

73. Read excel file with specific columns

"usecols" should help, use range of columns (as per excel worksheet, A,B...etc.) below are the examples

**1. Selected Columns**

df = pd.read\_excel(file\_location,sheet\_name='Sheet1', usecols="A,C,F")

**2. Range of Columns and selected column**

df = pd.read\_excel(file\_location,sheet\_name='Sheet1', usecols="A:F,H")

**3. Multiple Ranges**

df = pd.read\_excel(file\_location,sheet\_name='Sheet1', usecols="A:F,H,J:N")

**4. Range of columns**

df = pd.read\_excel(file\_location,sheet\_name='Sheet1', usecols="A:N")

74. consider row as column header.

dk = df.drop(index=df.index[:skpRw], axis=0)

header\_row = dk.iloc[0]

df4 = pd.DataFrame(dk.values[1:], columns=header\_row)

75. Send outlook mail

import win32com.client as client

from pathlib import Path

id = 'ssc.achauhan@cma-cgm.com'

outlook = client.Dispatch("Outlook.Application")

ol\_msg = outlook.CreateItem(0)

ol\_msg.To = id

ol\_msg.Subject = 'New indicators : - EQM Tracking -' + shName

ol\_msg.Body = 'Hi,\n\nPlease find the attached file as requested.\n\nThank you'

attachment1 = shName + '.xlsx'

src\_file = Path.cwd() / attachment1

ol\_msg.Attachments.Add(str(src\_file))

# ol\_msg.display()

outlook.Send()

76. How to delete rows from a pandas DataFrame based on a conditional expression [duplicate].

lUnt = df\_11[df\_11['PTS Code']==df\_11['Final POD']].index

df\_11.drop(lUnt,inplace=True)

df.drop(df[df.score < 50].index, inplace=True)