

```
In [1]: import pandas as pd
pd.options.mode.chained_assignment = None # default='warn'
from datetime import datetime, date, timedelta
import pickle
```

```
In [2]: now = datetime.now().date()
sdate= datetime.now().date()-timedelta(days=365*2)
datediff = (now-sdate).days
```

```
In [3]: allwind = pd.read_pickle('brommer_raw_2020-10-04_2022-10-03.pkl')
```

```
In [4]: allwind[::len(allwind)//10]
```

```
Out[4]:
```

	Date	Time	Season	Month	Daypart	Temp	Wind	Azimuth	Speed	Gust
0	2020-10-04	12:04 AM	Fall	Oct	Overnight	55.6	East	90.0	0.0	0.0
20405	2020-12-14	12:34 AM	Winter	Dec	Overnight	45.4	WSW	247.5	0.0	0.9
40810	2021-02-23	12:24 AM	Winter	Feb	Overnight	49.4	WSW	247.5	0.0	0.0
61215	2021-05-20	7:24 AM	Spring	May	Morning	47.6	NE	45.0	1.2	4.0
81620	2021-07-30	1:29 PM	Summer	Jul	Midday	63.9	SW	225.0	4.3	8.0
102025	2021-10-09	4:29 PM	Fall	Oct	Afternoon	67.8	WSW	247.5	3.0	8.0
122430	2021-12-19	9:52 PM	Winter	Dec	Evening	51.1	SSW	202.5	7.0	8.0
142835	2022-03-01	8:14 PM	Spring	Mar	Evening	54.9	East	90.0	0.9	6.0
163240	2022-05-12	12:24 AM	Spring	May	Overnight	47.7	NNE	22.5	2.4	4.0
183645	2022-07-23	12:19 AM	Summer	Jul	Overnight	56.2	West	270.0	0.9	3.0
204050	2022-10-03	11:39 PM	Fall	Oct	Overnight	57.8	WNW	292.5	0.2	2.0

```
In [37]: wind_wsw = allwind.query('Wind == "WSW"')
wind_ene = allwind.query('Wind == "ENE"')
```

```
In [82]: print(f'wsw = {round(100*len(wind_wsw)/len(allwind),1)}% | ene = {round(100*len(wind_ene)/len(allwind),1)}%')
wsw = 3.2% | ene = 4.2%
```

```
In [51]: def dftostrFF(df):
print(df.reset_index(drop=False).to_string(index=False))
def dftostrTT(df):
print(df.reset_index(drop=True).to_string(index=True))
def dftostrTF(df):
print(df.reset_index(drop=True).to_string(index=False))
def dftostrFT(df):
print(df.reset_index(drop=False).to_string(index=True))
```

```
In [48]: ordidir={"North":0.0,
               "NNE":22.5,
               "NE": 45.0,
               "ENE":67.5,
               "East":90.0,
               "ESE":112.5,
               "SE":135.0,
               "SSE":157.5,
               "South":180,
               "SSW":202.5,
               "SW":225.0,
               "WSW":247.5,
               "West":270.0,
               "WNW":292.5,
               "NW":315.0,
               "NNW":337.5}
revdir = [(value, key) for key, value in ordidir.items()]
print('mapping of 16 angular wind directions to 16 compass points:\n')
print(pd.DataFrame(revdir,columns=['Azimuth','Compass']).to_string(index=False))
revord = dict(revdir)
```

mapping of 16 angular wind directions to 16 compass points:

Azimuth	Compass
0.0	North
22.5	NNE
45.0	NE
67.5	ENE
90.0	East
112.5	ESE
135.0	SE
157.5	SSE
180.0	South
202.5	SSW
225.0	SW
247.5	WSW
270.0	West
292.5	WNW
315.0	NW
337.5	NNW

```
In [49]: def ordinalify(df,groupcol,grouporder):
          avgdir = df.groupby(groupcol)['Azimuth'].mean()[grouporder]
          avgint = avgdir.astype(int)
          clsdir = round(avgdir/22.5,0)*22.5
          dford = pd.concat([pd.concat([pd.DataFrame(avgint),pd.DataFrame(clsdir)
          dford.columns=['Azimuth','Closest','Compass']
          dford = dford.reset_index(drop=False)
          return dford
```

```
In [52]: print("average wind Azimuth by season and daypart:\n")
dfs = allwind.groupby(['Season', 'Daypart'])['Azimuth'].mean().round(1)
dfs = dfs.reset_index(drop=False)
dfs['Season'] = pd.Categorical(dfs['Season'], ['Winter', 'Spring', 'Summer', 'Fall'])
dfs['Daypart'] = pd.Categorical(dfs['Daypart'], ['Overnight', 'Morning', 'Midday', 'Afternoon', 'Evening'])
dftostrTF(dfs.sort_values(['Season', 'Daypart']))
```

average wind Azimuth by season and daypart:

Season	Daypart	Azimuth
Winter	Overnight	171.5
Winter	Morning	165.7
Winter	Midday	175.7
Winter	Afternoon	194.0
Winter	Evening	175.4
Spring	Overnight	133.5
Spring	Morning	134.0
Spring	Midday	158.1
Spring	Afternoon	151.4
Spring	Evening	152.7
Summer	Overnight	157.5
Summer	Morning	147.4
Summer	Midday	180.9
Summer	Afternoon	163.8
Summer	Evening	146.7
Fall	Overnight	140.6
Fall	Morning	146.9
Fall	Midday	179.0
Fall	Afternoon	181.2
Fall	Evening	152.6

```
In [53]: print("average wind speed by season and daypart:\n")
dfs = allwind.groupby(['Season', 'Daypart'])['Speed'].mean().round(1)
dfs = dfs.reset_index(drop=False)
dfs['Season'] = pd.Categorical(dfs['Season'], ['Winter', 'Spring', 'Summer', 'Fall'])
dfs['Daypart'] = pd.Categorical(dfs['Daypart'], ['Overnight', 'Morning', 'Midday', 'Afternoon', 'Evening'])
dftostrTF(dfs.sort_values(['Season', 'Daypart']))
```

average wind speed by season and daypart:

Season	Daypart	Speed
Winter	Overnight	1.3
Winter	Morning	1.3
Winter	Midday	3.8
Winter	Afternoon	2.8
Winter	Evening	1.5
Spring	Overnight	1.2
Spring	Morning	1.6
Spring	Midday	4.6
Spring	Afternoon	5.1
Spring	Evening	3.0
Summer	Overnight	1.2
Summer	Morning	1.7
Summer	Midday	4.1
Summer	Afternoon	4.9
Summer	Evening	3.2
Fall	Overnight	0.9
Fall	Morning	1.2
Fall	Midday	3.1
Fall	Afternoon	2.9
Fall	Evening	1.5

```
In [16]: print("ascending avg wind speed by season and daypart:\n")
dftostrTF(dfs.sort_values('Speed'))
```

ascending avg wind speed by season and daypart:

Season	Daypart	Speed
Fall	Overnight	0.9
Spring	Overnight	1.2
Summer	Overnight	1.2
Fall	Morning	1.2
Winter	Morning	1.3
Winter	Overnight	1.3
Fall	Evening	1.5
Winter	Evening	1.5
Spring	Morning	1.6
Summer	Morning	1.7
Winter	Afternoon	2.8
Fall	Afternoon	2.9
Spring	Evening	3.0
Fall	Midday	3.1
Summer	Evening	3.2
Winter	Midday	3.8
Summer	Midday	4.1

```
In [17]: print("unweighted average wind azimuth direction by season:\n")
orderszn = ['Winter', 'Spring', 'Summer', 'Fall']
dftostrTF(ordinalify(allwind, 'Season', orderszn))
print("\n\nunweighted average wind azimuth direction by month:\n")
ordermth=['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov']
dftostrTF(ordinalify(allwind, 'Month', ordermth))
```

unweighted average wind azimuth direction by season:

Season	Azimuth	Closest	Compass
Winter	175	180.0	South
Spring	144	135.0	SE
Summer	159	157.5	SSE
Fall	157	157.5	SSE

unweighted average wind azimuth direction by month:

Month	Azimuth	Closest	Compass
Jan	164	157.5	SSE
Feb	190	180.0	South
Mar	167	157.5	SSE
Apr	107	112.5	ESE
May	158	157.5	SSE
Jun	161	157.5	SSE
Jul	158	157.5	SSE
Aug	157	157.5	SSE
Sep	156	157.5	SSE
Oct	155	157.5	SSE
Nov	160	157.5	SSE
Dec	173	180.0	South

```
In [61]: def groupwavg(groupcol, grouporder):
    wgtldir = (nnwind['Speed']).multiply(nnwind['Azimuth'])
    dfwgt = pd.concat([nnwind, pd.DataFrame(wgtldir, columns=['Wgtldir'])], axis=1)
    dfsum = dfwgt.groupby(groupcol).sum()
    wavgd = (dfsum.Wgtldir/dfsum.Speed)
    wintd = wavgd.astype(int)
    wclsd = round(wavgd/22.5, 0)*22.5
    wcomp = wclsd.map(revord)
    dfwavg = pd.concat([pd.concat([wintd, wclsd], axis=1), wcomp], axis=1)
    dfwavg.columns = ['Azimuth', 'Closest', 'Compass']
    dfwavg = dfwavg.reset_index(drop=False)
    dfwavg[groupcol] = pd.Categorical(dfwavg[groupcol], grouporder)
    return dfwavg.sort_values(groupcol)
```

```
In [20]: nnwind = allwind[allwind['Azimuth'].notnull()]
```

```
In [25]: print("speed weighted average wind azimuth direction by Season:\n")
dftostrTF(groupwavg('Season', ['Winter', 'Spring', 'Summer', 'Fall']))
print("\n\nspeed weighted average wind azimuth direction by Month:\n")
dftostrTF(groupwavg('Month', ordermth))
```

speed weighted average wind azimuth direction by Season:

Season	Azimuth	Closest	Compass
Winter	185	180.0	South
Spring	165	157.5	SSE
Summer	161	157.5	SSE
Fall	169	180.0	South

speed weighted average wind azimuth direction by Month:

Month	Azimuth	Closest	Compass
Jan	174	180.0	South
Feb	204	202.5	SSW
Mar	185	180.0	South
Apr	132	135.0	SE
May	177	180.0	South
Jun	166	157.5	SSE
Jul	159	157.5	SSE
Aug	158	157.5	SSE
Sep	161	157.5	SSE
Oct	170	180.0	South
Nov	180	180.0	South
Dec	175	180.0	South

```
In [26]: print("difference in monthly average azimuth without vs with wind speed as
dfdiff=pd.concat([pd.Series(ordermth),(ordinalify(allwind,'Month',ordermth)
dfdiff.columns=['Month','Azimuth']
dftostrTF(dfdiff)
print("\n\ndifference in seasonal average azimuth without vs with wind spee
dfdiff=pd.concat([pd.Series(orderszn),(ordinalify(allwind,'Season',orderszn)
dfdiff.columns=['Season','Azimuth']
dftostrTF(dfdiff)
```

difference in monthly average azimuth without vs with wind speed as weights:

Month	Azimuth
Jan	32
Feb	32
Mar	-8
Apr	-97
May	-16
Jun	2
Jul	-8
Aug	-28
Sep	-21
Oct	-25
Nov	-10
Dec	12

difference in seasonal average azimuth without vs with wind speed as weights:

Season	Azimuth
Winter	6
Spring	-21
Summer	-2
Fall	-28

```
In [59]: list(ordir.keys())
```

```
Out[59]: ['North',
          'NNE',
          'NE',
          'ENE',
          'East',
          'ESE',
          'SE',
          'SSE',
          'South',
          'SSW',
          'SW',
          'WSW',
          'West',
          'WNW',
          'NW',
          'NNW']
```

```
In [80]: allwind.query("Azimuth.isnull()")
```

```
Out[80]:
```

	Date	Time	Season	Month	Daypart	Temp	Wind	Azimuth	Speed	Gust
2480	2020-10-12	2:44 PM	Fall	Oct	Afternoon	82.7	NaN	NaN	0.0	1.0
2481	2020-10-12	2:49 PM	Fall	Oct	Afternoon	82.8	NaN	NaN	0.0	1.0
2482	2020-10-12	2:54 PM	Fall	Oct	Afternoon	82.9	NaN	NaN	0.0	1.0
2483	2020-10-12	2:59 PM	Fall	Oct	Afternoon	83.1	NaN	NaN	0.0	1.9
2484	2020-10-12	3:04 PM	Fall	Oct	Afternoon	83.1	NaN	NaN	0.0	2.0
...
203848	2022-10-03	6:49 AM	Fall	Oct	Morning	53.4	NaN	NaN	1.1	3.0
203851	2022-10-03	7:04 AM	Fall	Oct	Morning	53.4	NaN	NaN	1.7	3.0
203855	2022-10-03	7:24 AM	Fall	Oct	Morning	53.4	NaN	NaN	1.2	4.0
203856	2022-10-03	7:29 AM	Fall	Oct	Morning	53.4	NaN	NaN	1.4	3.0
203861	2022-10-03	7:54 AM	Fall	Oct	Morning	54.4	NaN	NaN	1.2	3.0

7115 rows × 10 columns

```
In [79]: [print(x) for x in allwind.query("Azimuth.isnull()").Date.unique()]
```

```
2020-10-12
2020-11-06
2020-11-08
2020-11-09
2020-11-17
2020-11-25
2020-11-26
2020-11-27
2020-12-07
2020-12-08
2020-12-17
2020-12-20
2020-12-21
2020-12-22
2020-12-23
2020-12-26
2020-12-28
2020-12-31
2021-01-05
2021-01-06
```

```
In [81]: allwind.groupby('Wind')['Speed'].mean().round(1).mean()
```

```
Out[81]: 2.1374999999999997
```

```
In [ ]:
```



```
In [64]: print("average wind speed by direction:\n")
dfs = allwind.groupby('Wind')['Speed'].mean().round(1)
dfs = dfs.reset_index(drop=False)
dfs['Wind'] = pd.Categorical(dfs['Wind'], list(ordir.keys()))
dftostrTF(dfs.sort_values('Wind'))

print("ascending avg wind speed by direction:\n")
dftostrTF(dfs.sort_values('Speed'))
```

average wind speed by direction:

Wind	Speed
North	1.9
NNE	1.5
NE	1.2
ENE	1.9
East	2.2
ESE	3.0
SE	2.0
SSE	1.5
South	2.3
SSW	3.4
SW	2.8
WSW	2.0
West	1.4
WNW	2.4
NW	2.6
NNW	2.1

ascending avg wind speed by season and daypart:

Wind	Speed
NE	1.2
West	1.4
NNE	1.5
SSE	1.5
ENE	1.9
North	1.9
SE	2.0
WSW	2.0
NNW	2.1
East	2.2
South	2.3
WNW	2.4
NW	2.6
SW	2.8
ESE	3.0
SSW	3.4