

Import libraries

```
In [1]: import pandas as pd  
import numpy as np  
import matplotlib as plt
```

Get forest fire data

```
In [2]: amazon_df = pd.read_csv("amazon.csv")
```

```
In [3]: print(len(amazon_df))
```

6454

```
In [4]: amazon_df.head(50)
```

```
Out[4]:
```

	year	state	month	number	date
0	1998	Acre	Janeiro	0.0	1998-01-01
1	1999	Acre	Janeiro	0.0	1999-01-01
2	2000	Acre	Janeiro	0.0	2000-01-01
3	2001	Acre	Janeiro	0.0	2001-01-01
4	2002	Acre	Janeiro	0.0	2002-01-01
5	2003	Acre	Janeiro	10.0	2003-01-01
6	2004	Acre	Janeiro	0.0	2004-01-01
7	2005	Acre	Janeiro	12.0	2005-01-01
8	2006	Acre	Janeiro	4.0	2006-01-01
9	2007	Acre	Janeiro	0.0	2007-01-01
10	2008	Acre	Janeiro	0.0	2008-01-01
11	2009	Acre	Janeiro	0.0	2009-01-01
12	2010	Acre	Janeiro	1.0	2010-01-01
13	2011	Acre	Janeiro	0.0	2011-01-01
14	2012	Acre	Janeiro	0.0	2012-01-01
15	2013	Acre	Janeiro	0.0	2013-01-01
16	2014	Acre	Janeiro	0.0	2014-01-01
17	2015	Acre	Janeiro	1.0	2015-01-01
18	2016	Acre	Janeiro	12.0	2016-01-01
19	2017	Acre	Janeiro	0.0	2017-01-01
20	1998	Acre	Fevereiro	0.0	1998-01-01
21	1999	Acre	Fevereiro	0.0	1999-01-01
22	2000	Acre	Fevereiro	0.0	2000-01-01
23	2001	Acre	Fevereiro	0.0	2001-01-01
24	2002	Acre	Fevereiro	1.0	2002-01-01
25	2003	Acre	Fevereiro	0.0	2003-01-01
26	2004	Acre	Fevereiro	3.0	2004-01-01
27	2005	Acre	Fevereiro	0.0	2005-01-01
28	2006	Acre	Fevereiro	0.0	2006-01-01
29	2007	Acre	Fevereiro	5.0	2007-01-01
30	2008	Acre	Fevereiro	0.0	2008-01-01
31	2009	Acre	Fevereiro	2.0	2009-01-01
32	2010	Acre	Fevereiro	0.0	2010-01-01
33	2011	Acre	Fevereiro	0.0	2011-01-01
34	2012	Acre	Fevereiro	0.0	2012-01-01
35	2013	Acre	Fevereiro	0.0	2013-01-01
36	2014	Acre	Fevereiro	0.0	2014-01-01
37	2015	Acre	Fevereiro	2.0	2015-01-01
38	2016	Acre	Fevereiro	5.0	2016-01-01
39	2017	Acre	Fevereiro	1.0	2017-01-01
40	1998	Acre	Março	0.0	1998-01-01
41	1999	Acre	Março	0.0	1999-01-01
42	2000	Acre	Março	11.0	2000-01-01
43	2001	Acre	Março	0.0	2001-01-01
44	2002	Acre	Março	0.0	2002-01-01
45	2003	Acre	Março	0.0	2003-01-01

	year	state	month	number	date
46	2004	Acre	Mar	0.0	2004-01-01
47	2005	Acre	Mar	3.0	2005-01-01
48	2006	Acre	Mar	0.0	2006-01-01
49	2007	Acre	Mar	2.0	2007-01-01

```
In [5]: month_map={'Janeiro': 1, 'Fevereiro': 2, 'Março': 3, 'Abril': 4, 'Maio': 5,
                'Junho': 6, 'Julho': 7, 'Agosto': 8, 'Setembro': 9, 'Outubro': 10,
                'Novembro': 11, 'Dezembro': 12}
amazon_df['month_nbr']=amazon_df['month'].map(month_map)
amazon_df['yearmonth'] = amazon_df['year']*100 + amazon_df['month_nbr']
amazon_df.head(50)
```

Out[5]:

	year	state	month	number	date	month_nbr	yearmonth
0	1998	Acre	Janeiro	0.0	1998-01-01	1	199801
1	1999	Acre	Janeiro	0.0	1999-01-01	1	199901
2	2000	Acre	Janeiro	0.0	2000-01-01	1	200001
3	2001	Acre	Janeiro	0.0	2001-01-01	1	200101
4	2002	Acre	Janeiro	0.0	2002-01-01	1	200201
5	2003	Acre	Janeiro	10.0	2003-01-01	1	200301
6	2004	Acre	Janeiro	0.0	2004-01-01	1	200401
7	2005	Acre	Janeiro	12.0	2005-01-01	1	200501
8	2006	Acre	Janeiro	4.0	2006-01-01	1	200601
9	2007	Acre	Janeiro	0.0	2007-01-01	1	200701
10	2008	Acre	Janeiro	0.0	2008-01-01	1	200801
11	2009	Acre	Janeiro	0.0	2009-01-01	1	200901
12	2010	Acre	Janeiro	1.0	2010-01-01	1	201001
13	2011	Acre	Janeiro	0.0	2011-01-01	1	201101
14	2012	Acre	Janeiro	0.0	2012-01-01	1	201201
15	2013	Acre	Janeiro	0.0	2013-01-01	1	201301
16	2014	Acre	Janeiro	0.0	2014-01-01	1	201401
17	2015	Acre	Janeiro	1.0	2015-01-01	1	201501
18	2016	Acre	Janeiro	12.0	2016-01-01	1	201601
19	2017	Acre	Janeiro	0.0	2017-01-01	1	201701
20	1998	Acre	Fevereiro	0.0	1998-01-01	2	199802
21	1999	Acre	Fevereiro	0.0	1999-01-01	2	199902
22	2000	Acre	Fevereiro	0.0	2000-01-01	2	200002
23	2001	Acre	Fevereiro	0.0	2001-01-01	2	200102
24	2002	Acre	Fevereiro	1.0	2002-01-01	2	200202
25	2003	Acre	Fevereiro	0.0	2003-01-01	2	200302
26	2004	Acre	Fevereiro	3.0	2004-01-01	2	200402
27	2005	Acre	Fevereiro	0.0	2005-01-01	2	200502
28	2006	Acre	Fevereiro	0.0	2006-01-01	2	200602
29	2007	Acre	Fevereiro	5.0	2007-01-01	2	200702
30	2008	Acre	Fevereiro	0.0	2008-01-01	2	200802
31	2009	Acre	Fevereiro	2.0	2009-01-01	2	200902
32	2010	Acre	Fevereiro	0.0	2010-01-01	2	201002
33	2011	Acre	Fevereiro	0.0	2011-01-01	2	201102
34	2012	Acre	Fevereiro	0.0	2012-01-01	2	201202
35	2013	Acre	Fevereiro	0.0	2013-01-01	2	201302
36	2014	Acre	Fevereiro	0.0	2014-01-01	2	201402
37	2015	Acre	Fevereiro	2.0	2015-01-01	2	201502
38	2016	Acre	Fevereiro	5.0	2016-01-01	2	201602
39	2017	Acre	Fevereiro	1.0	2017-01-01	2	201702
40	1998	Acre	Março	0.0	1998-01-01	3	199803
41	1999	Acre	Março	0.0	1999-01-01	3	199903

	year	state	month	number	date	month_nbr	yearmonth
42	2000	Acre	Mar	11.0	2000-01-01	3	200003
43	2001	Acre	Mar	0.0	2001-01-01	3	200103
44	2002	Acre	Mar	0.0	2002-01-01	3	200203
45	2003	Acre	Mar	0.0	2003-01-01	3	200303
46	2004	Acre	Mar	0.0	2004-01-01	3	200403
47	2005	Acre	Mar	3.0	2005-01-01	3	200503
48	2006	Acre	Mar	0.0	2006-01-01	3	200603
49	2007	Acre	Mar	2.0	2007-01-01	3	200703

```
In [6]: # Check for NaN values
amazon_df['month_nbr'].isnull().values.any()
```

Out[6]: False

```
In [7]: k = amazon_df['state'].unique()
k.sort()
print(k)

['Acre' 'Alagoas' 'Amapa' 'Amazonas' 'Bahia' 'Ceara' 'Distrito Federal'
'Espirito Santo' 'Goias' 'Maranhao' 'Mato Grosso' 'Minas Gerais'
'Paraiba' 'Parana' 'Pernambuco' 'Piau' 'Rio' 'Rondonia' 'Roraima'
'Santa Catarina' 'Sao Paulo' 'Sergipe' 'Tocantins']
```

```
In [8]: sc_df = pd.read_csv("automatic_stations_codes_2000_2021.csv",delimiter=";")
sc_df.head(20)
```

Out[8]:

	REGIAO	UF	ESTACAO	CODIGO	LATITUDE	LONGITUDE	ALTITUDE
0	N	PA	SANTA MARIA DAS BARREIRAS	A256	-8.729722	-49.856389	165.00
1	SE	SP	CRIOSEFERA	C891	-84.000000	-79.494167	1285.00
2	CO	DF	BRASILIA	A001	-15.789444	-47.925833	1159.54
3	NE	BA	SALVADOR	A401	-13.016667	-38.516667	51.41
4	N	AM	MANAUS	A101	-3.103333	-60.016389	61.25
5	SE	RJ	ECOLOGIA AGRICOLA	A601	-22.800000	-43.683333	33.00
6	S	RS	PORTO ALEGRE	A801	-30.050000	-51.166667	46.97
7	CO	GO	GOIANIA	A002	-16.642778	-49.220000	770.00
8	CO	GO	MORRINHOS	A003	-17.716667	-49.100000	771.42
9	CO	MS	CAMPO GRANDE	A702	-20.450000	-54.600000	530.00
10	CO	MS	PONTA PORA	A703	-22.552500	-55.716389	675.00

```
In [9]: k = sc_df['UF'].unique()
k.sort()
print(k)
#MS
#PR
# RN pohjoisessa

['AC' 'AL' 'AM' 'AP' 'BA' 'CE' 'DF' 'ES' 'GO' 'MA' 'MG' 'MS' 'MT' 'PA'
'PB' 'PE' 'PI' 'PR' 'RJ' 'RN' 'RO' 'RR' 'RS' 'SC' 'SE' 'SP' 'TO']
```

```
In [10]: codigo = sc_df['CODIGO']
uf = sc_df['UF']
state_map2 = dict(zip(codigo,uf))
print(state_map2)
print(len(codigo))
```

```
{'A256': 'PA', 'C891': 'SP', 'A001': 'DF', 'A401': 'BA', 'A101': 'AM', 'A601': 'RJ', 'A801': 'RS',
'A002': 'GO', 'A003': 'GO', 'A702': 'MS', 'A703': 'MS', 'A704': 'MS', 'A402': 'BA', 'A705': 'SP', 'A
802': 'RS', 'A803': 'RS', 'A804': 'RS', 'A805': 'RS', 'A901': 'MT', 'A902': 'MT', 'A904': 'MT', 'A90
5': 'MT', 'A906': 'MT', 'A404': 'BA', 'A405': 'BA', 'A502': 'MG', 'A505': 'MG', 'A506': 'MG', 'A50
7': 'MG', 'A508': 'MG', 'A602': 'RJ', 'A603': 'RJ', 'A604': 'RJ', 'A706': 'SP', 'A708': 'SP', 'A70
9': 'MS', 'A903': 'MT', 'A907': 'MT', 'A303': 'AL', 'A406': 'BA', 'A407': 'BA', 'A408': 'BA', 'A41
0': 'BA', 'A412': 'BA', 'A305': 'CE', 'A306': 'CE', 'A203': 'MA', 'A307': 'PE', 'A308': 'PI', 'A30
2': 'RN', 'A304': 'RN', 'A409': 'SE', 'A201': 'PA', 'A202': 'PA', 'A707': 'SP', 'A807': 'PR', 'A80
6': 'SC', 'A310': 'PB', 'A301': 'PE', 'A309': 'PE', 'A311': 'PI', 'A312': 'PI', 'A009': 'TO', 'A50
9': 'MG', 'A010': 'TO', 'A510': 'MG', 'A011': 'GO', 'A012': 'GO', 'A710': 'MS', 'A717': 'MS', 'A71
9': 'MS', 'A720': 'MS', 'A721': 'MS', 'A722': 'MS', 'A723': 'MS', 'A724': 'MS', 'A730': 'MS', 'A73
1': 'MS', 'A732': 'MS', 'A908': 'MT', 'A910': 'MT', 'A912': 'MT', 'A913': 'MT', 'A914': 'MT', 'A91
5': 'MT', 'A917': 'MT', 'A920': 'MT', 'A922': 'MT', 'A313': 'PB', 'A018': 'TO', 'A019': 'TO', 'A61
2': 'ES', 'A614': 'ES', 'A615': 'ES', 'A616': 'ES', 'A617': 'ES', 'A511': 'MG', 'A512': 'MG', 'A51
3': 'MG', 'A514': 'MG', 'A515': 'MG', 'A516': 'MG', 'A517': 'MG', 'A519': 'MG', 'A520': 'MG', 'A52
1': 'MG', 'A522': 'MG', 'A523': 'MG', 'A524': 'MG', 'A525': 'MG', 'A527': 'MG', 'A528': 'MG', 'A53
0': 'MG', 'A531': 'MG', 'A538': 'MG', 'A606': 'RJ', 'A607': 'RJ', 'A608': 'RJ', 'A609': 'RJ', 'A61
0': 'RJ', 'A611': 'RJ', 'A618': 'RJ', 'A619': 'RJ', 'A701': 'SP', 'A711': 'SP', 'A712': 'SP', 'A71
3': 'SP', 'A714': 'SP', 'A715': 'SP', 'A716': 'SP', 'A718': 'SP', 'A725': 'SP', 'A726': 'SP', 'A72
7': 'SP', 'A728': 'SP', 'A729': 'SP', 'A818': 'PR', 'A819': 'PR', 'A820': 'PR', 'A821': 'PR', 'A82
2': 'PR', 'A823': 'PR', 'A824': 'PR', 'A835': 'PR', 'A808': 'RS', 'A809': 'RS', 'A810': 'RS', 'A81
2': 'RS', 'A813': 'RS', 'A826': 'RS', 'A828': 'RS', 'A829': 'RS', 'A838': 'RS', 'A839': 'RS', 'A84
0': 'RS', 'A817': 'SC', 'A013': 'GO', 'A014': 'GO', 'A015': 'GO', 'A016': 'GO', 'A017': 'GO', 'A02
2': 'GO', 'A023': 'GO', 'A024': 'GO', 'A025': 'GO', 'A026': 'GO', 'A032': 'GO', 'A033': 'GO', 'A03
5': 'GO', 'A036': 'GO', 'A916': 'MT', 'A919': 'MT', 'A921': 'MT', 'A924': 'MT', 'A323': 'AL', 'A32
7': 'AL', 'A413': 'BA', 'A414': 'BA', 'A415': 'BA', 'A416': 'BA', 'A418': 'BA', 'A314': 'CE', 'A31
5': 'CE', 'A319': 'CE', 'A324': 'CE', 'A325': 'CE', 'A332': 'CE', 'A339': 'CE', 'A204': 'MA', 'A20
5': 'MA', 'A320': 'PB', 'A321': 'PB', 'A333': 'PB', 'A334': 'PB', 'A322': 'PE', 'A329': 'PE', 'A34
1': 'PE', 'A326': 'PI', 'A330': 'PI', 'A331': 'PI', 'A335': 'PI', 'A336': 'PI', 'A337': 'PI', 'A33
8': 'PI', 'A316': 'RN', 'A317': 'RN', 'A318': 'RN', 'A340': 'RN', 'A417': 'SE', 'A212': 'PA', 'A21
3': 'PA', 'A925': 'RO', 'A020': 'TO', 'A021': 'TO', 'A613': 'ES', 'A518': 'MG', 'A526': 'MG', 'A52
9': 'MG', 'A532': 'MG', 'A533': 'MG', 'A534': 'MG', 'A536': 'MG', 'A537': 'MG', 'A539': 'MG', 'A54
0': 'MG', 'A541': 'MG', 'A542': 'MG', 'A543': 'MG', 'A544': 'MG', 'A545': 'MG', 'A546': 'MG', 'A54
7': 'MG', 'A548': 'MG', 'A549': 'MG', 'A550': 'MG', 'A551': 'MG', 'A552': 'MG', 'A553': 'MG', 'A55
4': 'MG', 'A621': 'RJ', 'A652': 'RJ', 'A733': 'SP', 'A734': 'SP', 'A735': 'SP', 'A736': 'SP', 'A73
7': 'SP', 'A738': 'SP', 'A739': 'SP', 'A740': 'SP', 'A842': 'PR', 'A843': 'PR', 'A847': 'PR', 'A85
5': 'PR', 'A811': 'RS', 'A827': 'RS', 'A830': 'RS', 'A831': 'RS', 'A832': 'RS', 'A836': 'RS', 'A84
4': 'RS', 'A852': 'RS', 'A853': 'RS', 'A854': 'RS', 'A841': 'SC', 'A845': 'SC', 'A851': 'SC', 'A804
5': 'DF', 'A027': 'GO', 'A034': 'GO', 'A742': 'MS', 'A743': 'MS', 'A749': 'MS', 'A750': 'MS', 'A75
1': 'MS', 'A752': 'MS', 'A754': 'MS', 'A926': 'MT', 'A927': 'MT', 'A928': 'MT', 'A929': 'MT', 'A93
0': 'MT', 'A931': 'MT', 'A932': 'MT', 'A933': 'MT', 'A934': 'MT', 'A935': 'MT', 'A936': 'MT', 'A93
7': 'MT', 'A353': 'AL', 'A355': 'AL', 'A356': 'AL', 'A422': 'BA', 'A423': 'BA', 'A424': 'BA', 'A42
5': 'BA', 'A426': 'BA', 'A427': 'BA', 'A428': 'BA', 'A429': 'BA', 'A430': 'BA', 'A431': 'BA', 'A43
2': 'BA', 'A433': 'BA', 'A434': 'BA', 'A435': 'BA', 'A436': 'BA', 'A437': 'BA', 'A438': 'BA', 'A43
9': 'BA', 'A440': 'BA', 'A441': 'BA', 'A442': 'BA', 'A443': 'BA', 'A342': 'CE', 'A347': 'CE', 'A35
8': 'CE', 'A359': 'CE', 'A206': 'MA', 'A207': 'MA', 'A217': 'MA', 'A218': 'MA', 'A219': 'MA', 'A22
0': 'MA', 'A221': 'MA', 'A222': 'MA', 'A223': 'MA', 'A224': 'MA', 'A225': 'MA', 'A237': 'MA', 'A23
8': 'MA', 'A348': 'PB', 'A352': 'PB', 'A328': 'PE', 'A349': 'PE', 'A350': 'PE', 'A351': 'PE', 'A35
7': 'PE', 'A343': 'PI', 'A345': 'PI', 'A346': 'PI', 'A354': 'PI', 'A344': 'RN', 'A419': 'SE', 'A42
0': 'SE', 'A421': 'SE', 'A102': 'AC', 'A104': 'AC', 'A110': 'AM', 'A111': 'AM', 'A112': 'AM', 'A11
3': 'AM', 'A117': 'AM', 'A119': 'AM', 'A120': 'AM', 'A121': 'AM', 'A122': 'AM', 'A123': 'AM', 'A12
4': 'AM', 'A125': 'AM', 'A126': 'AM', 'A128': 'AM', 'A242': 'AP', 'A209': 'PA', 'A210': 'PA', 'A21
1': 'PA', 'A214': 'PA', 'A215': 'PA', 'A226': 'PA', 'A227': 'PA', 'A229': 'PA', 'A230': 'PA', 'A23
1': 'PA', 'A233': 'PA', 'A234': 'PA', 'A235': 'PA', 'A236': 'PA', 'A241': 'PA', 'A938': 'RO', 'A93
9': 'RO', 'A940': 'RO', 'A038': 'TO', 'A039': 'TO', 'A622': 'ES', 'A623': 'ES', 'A535': 'MG', 'A55
5': 'MG', 'A620': 'RJ', 'A741': 'SP', 'A746': 'SP', 'A747': 'SP', 'A753': 'SP', 'A846': 'PR', 'A84
9': 'PR', 'A850': 'PR', 'A869': 'PR', 'A871': 'PR', 'A873': 'PR', 'A875': 'PR', 'A876': 'PR', 'A83
4': 'RS', 'A837': 'RS', 'A856': 'RS', 'A878': 'RS', 'A879': 'RS', 'A880': 'RS', 'A899': 'RS', 'A81
4': 'SC', 'A815': 'SC', 'A816': 'SC', 'A848': 'SC', 'A857': 'SC', 'A858': 'SC', 'A859': 'SC', 'A86
0': 'SC', 'A861': 'SC', 'A862': 'SC', 'A863': 'SC', 'A866': 'SC', 'A867': 'SC', 'A444': 'BA', 'A44
5': 'BA', 'A446': 'BA', 'A447': 'BA', 'A360': 'CE', 'A361': 'PI', 'A362': 'PI', 'A363': 'PI', 'A36
4': 'PI', 'A136': 'AC', 'A137': 'AC', 'A138': 'AC', 'A140': 'AC', 'A240': 'PA', 'A044': 'TO', 'A83
3': 'RS', 'A864': 'SC', 'A037': 'GO', 'A756': 'MS', 'A366': 'PE', 'A365': 'PI', 'A367': 'RN', 'A24
6': 'PA', 'A135': 'RR', 'A556': 'MG', 'A624': 'RJ', 'A748': 'SP', 'A881': 'RS', 'A868': 'SC', 'A75
7': 'MS', 'A758': 'MS', 'A909': 'MT', 'A133': 'AM', 'A134': 'AM', 'A248': 'PA', 'A657': 'ES', 'A75
5': 'SP', 'A872': 'PR', 'A874': 'PR', 'A760': 'MS', 'A761': 'MS', 'A941': 'MT', 'A109': 'AM', 'A23
2': 'PA', 'A239': 'PA', 'A040': 'TO', 'A041': 'TO', 'A043': 'TO', 'A557': 'MG', 'A882': 'RS', 'A88
3': 'RS', 'A028': 'GO', 'A759': 'MS', 'A249': 'AP', 'F501': 'MG', 'A884': 'RS', 'A046': 'DF', 'A86
5': 'SC', 'A448': 'BA', 'A450': 'BA', 'A108': 'AC', 'A250': 'PA', 'A052': 'TO', 'A054': 'TO', 'A56
```

```

1': 'MG', 'A659': 'RJ', 'A667': 'RJ', 'A449': 'BA', 'A452': 'BA', 'A454': 'BA', 'A455': 'BA', 'S10
8': 'MA', 'S102': 'AM', 'S103': 'AM', 'S105': 'AM', 'S115': 'AM', 'S116': 'AM', 'S117': 'AM', 'S12
1': 'AM', 'A216': 'PA', 'A247': 'PA', 'S107': 'PA', 'S111': 'PA', 'S120': 'PA', 'S101': 'RO', 'S10
4': 'RO', 'S110': 'RO', 'S113': 'RR', 'S114': 'RR', 'A048': 'TO', 'A049': 'TO', 'A050': 'TO', 'A05
1': 'TO', 'A053': 'TO', 'A055': 'TO', 'A559': 'MG', 'A563': 'MG', 'A565': 'MG', 'A569': 'MG', 'S12
2': 'MG', 'A625': 'RJ', 'A626': 'RJ', 'A762': 'SP', 'A764': 'SP', 'B803': 'PR', 'B804': 'PR', 'B80
6': 'PR', 'A886': 'RS', 'A889': 'RS', 'A894': 'RS', 'A897': 'RS', 'A870': 'SC', 'A042': 'DF', 'A04
7': 'DF', 'A029': 'GO', 'A031': 'GO', 'A911': 'MT', 'A923': 'MT', 'A371': 'AL', 'A373': 'PB', 'A37
0': 'PE', 'A372': 'RN', 'A451': 'SE', 'A453': 'SE', 'A251': 'AP', 'A228': 'PA', 'A631': 'ES', 'A63
2': 'ES', 'A633': 'ES', 'A634': 'ES', 'A562': 'MG', 'A564': 'MG', 'A566': 'MG', 'A567': 'MG', 'A56
8': 'MG', 'A570': 'MG', 'A628': 'RJ', 'A635': 'RJ', 'A636': 'RJ', 'A744': 'SP', 'A763': 'SP', 'A76
5': 'SP', 'A766': 'SP', 'A767': 'SP', 'A768': 'SP', 'A769': 'SP', 'A005': 'GO', 'A056': 'GO', 'S70
1': 'MS', 'S702': 'MS', 'S703': 'MS', 'S704': 'MS', 'S705': 'MS', 'S706': 'MS', 'S707': 'MS', 'S70
8': 'MS', 'S709': 'MS', 'S710': 'MS', 'S711': 'MS', 'S712': 'MS', 'S713': 'MS', 'S714': 'MS', 'S71
5': 'MS', 'S716': 'MS', 'S717': 'MS', 'A456': 'BA', 'A458': 'BA', 'A368': 'CE', 'A369': 'CE', 'A37
4': 'PI', 'A375': 'PI', 'A376': 'PI', 'A252': 'PA', 'A253': 'PA', 'A571': 'MG', 'A627': 'RJ', 'A62
9': 'RJ', 'A630': 'RJ', 'A771': 'SP', 'A893': 'RS', 'A942': 'MT', 'A943': 'MT', 'A944': 'MT', 'A25
5': 'MA', 'A377': 'PI', 'A144': 'AM', 'A244': 'AP', 'A254': 'PA', 'A770': 'SP', 'A825': 'PR', 'A88
7': 'RS', 'A895': 'SC', 'A898': 'SC'}
612

```

```

In [11]: k = sc_df['REGIAO'].unique()
k.sort()
print(k)

['CO' 'N' 'NE' 'S' 'SE']

```

```

In [12]: state_map={'Acre':'AC', 'Alagoas':'AL', 'Amapa':'AP', 'Amazonas':'AM', 'Bahia':'BA', 'Ceara':'CE',
'Distrito Federal':'DF', 'Espirito Santo':'ES', 'Goias':'GO', 'Maranhao':'MA',
'Mato Grosso':'MT', 'Minas Gerais':'MG', 'Pará':'PA', 'Paraiba':'PB', 'Pernambuco':'PE',
'Piau':'PI', 'Rio':'RJ', 'Rondonia':'RO', 'Roraima':'RR', 'Santa Catarina':'SC',
'Sao Paulo':'SP', 'Sergipe':'SE', 'Tocantins':'TO'}

```

```

In [13]: amazon_df['code']=amazon_df['state'].map(state_map)
print(amazon_df)

```

	year	state	month	number	date	month_nbr	yearmonth	code
0	1998	Acre	Janeiro	0.0	1998-01-01	1	199801	AC
1	1999	Acre	Janeiro	0.0	1999-01-01	1	199901	AC
2	2000	Acre	Janeiro	0.0	2000-01-01	1	200001	AC
3	2001	Acre	Janeiro	0.0	2001-01-01	1	200101	AC
4	2002	Acre	Janeiro	0.0	2002-01-01	1	200201	AC
...
6449	2012	Tocantins	Dezembro	128.0	2012-01-01	12	201212	TO
6450	2013	Tocantins	Dezembro	85.0	2013-01-01	12	201312	TO
6451	2014	Tocantins	Dezembro	223.0	2014-01-01	12	201412	TO
6452	2015	Tocantins	Dezembro	373.0	2015-01-01	12	201512	TO
6453	2016	Tocantins	Dezembro	119.0	2016-01-01	12	201612	TO

[6454 rows x 8 columns]

```

In [14]: amazon_df.to_csv("amazon2.csv", sep=';')

```

```

In [15]: amazon_df = None

```

Get weather data

```

In [16]: weather_df = pd.read_csv("automatic_weather_stations_inmet_brazil_2000_2021.csv", delimiter=";")

```

In [17]: weather_df.head(40)

Out[17]:

	ESTACAO	DATA (YYYY-MM-DD)	HORA (UTC)	PRECIPITACAO TOTAL HORARIO (mm)	PRESSAO ATMOSFERICA AO NIVEL DA ESTACAO, HORARIA (mB)	PRESSAO ATMOSFERICA MAX.NA HORA ANT. (AUT) (mB)	PRESSAO ATMOSFERICA MIN. NA HORA ANT. (AUT) (mB)	RADIACAO GLOBAL (W/m2)	TEMPERATURA DO AR - BULBO SECO, HORARIA (C)
0	A001	2021-01-01	0	0.0	887.1	887.1	886.3	NaN	20.9
1	A001	2021-01-01	100	0.0	887.3	887.5	887.1	NaN	20.5
2	A001	2021-01-01	200	0.0	887.9	888.0	887.2	NaN	19.6
3	A001	2021-01-01	300	0.0	887.6	888.0	887.6	NaN	19.4
4	A001	2021-01-01	400	0.0	887.0	887.6	887.0	NaN	18.7
5	A001	2021-01-01	500	0.0	887.1	887.2	887.0	NaN	21.0
6	A001	2021-01-01	600	0.0	886.5	887.1	886.4	NaN	19.4
7	A001	2021-01-01	700	0.0	886.5	886.5	886.3	NaN	18.5
8	A001	2021-01-01	800	0.0	886.8	886.8	886.4	NaN	18.9
9	A001	2021-01-01	900	0.0	887.3	887.3	886.8	NaN	18.3
10	A001	2021-01-01	1000	0.0	887.9	887.9	887.3	298.2	20.2
11	A001	2021-01-01	1100	0.0	888.2	888.2	887.8	1203.7	23.2
12	A001	2021-01-01	1200	0.0	888.6	888.6	888.2	1664.9	24.9
13	A001	2021-01-01	1300	0.0	888.7	888.7	888.6	2851.7	25.9
14	A001	2021-01-01	1400	0.0	888.5	888.7	888.4	3115.7	27.0
15	A001	2021-01-01	1500	0.0	888.2	888.5	888.2	2970.1	28.1
16	A001	2021-01-01	1600	0.0	887.5	888.2	887.5	3528.2	28.8
17	A001	2021-01-01	1700	0.0	886.6	887.5	886.6	3372.1	28.5
18	A001	2021-01-01	1800	1.2	886.0	886.6	886.0	777.3	21.3
19	A001	2021-01-01	1900	0.2	885.8	886.0	885.8	700.5	23.0
20	A001	2021-01-01	2000	0.0	886.0	886.2	885.8	238.3	21.8
21	A001	2021-01-01	2100	0.0	885.9	886.0	885.7	125.0	21.8
22	A001	2021-01-01	2200	0.0	886.7	886.7	885.7	19.1	20.8
23	A001	2021-01-01	2300	0.0	887.7	887.8	886.6	NaN	20.7
24	A001	2021-01-02	0	0.0	887.8	887.9	887.5	NaN	18.9
25	A001	2021-01-02	100	0.0	887.8	888.0	887.7	NaN	18.5
26	A001	2021-01-02	200	0.0	888.0	888.0	887.8	NaN	18.8
27	A001	2021-01-02	300	0.0	887.2	888.0	887.2	NaN	18.5

	ESTACAO	DATA (YYYY-MM-DD)	HORA (UTC)	PRECIPITACAO TOTAL HORARIO (mm)	PRESSAO ATMOSFERICA AO NIVEL DA ESTACAO, HORARIA (mB)	PRESSAO MAX.NA HORA ANT. (AUT) (mB)	PRESSAO MIN. NA HORA ANT. (AUT) (mB)	RADIACAO GLOBAL (W/m2)	TEMPERATURA DO AR - BULBO SECO, HORARIA (C)
28	A001	2021-01-02	400	0.0	886.5	887.2	886.4	NaN	19.0
29	A001	2021-01-02	500	0.0	886.1	886.5	886.1	NaN	18.4
30	A001	2021-01-02	600	0.0	885.8	886.1	885.7	NaN	18.4
31	A001	2021-01-02	700	0.0	885.3	885.7	885.3	NaN	18.8
32	A001	2021-01-02	800	0.0	885.8	885.8	885.3	NaN	17.9
33	A001	2021-01-02	900	0.0	886.4	886.4	885.8	NaN	18.6
34	A001	2021-01-02	1000	0.0	886.6	886.6	886.3	147.3	19.7
35	A001	2021-01-02	1100	0.0	886.9	886.9	886.6	944.1	20.0
36	A001	2021-01-02	1200	0.0	887.2	887.2	886.8	1748.5	21.8
37	A001	2021-01-02	1300	0.0	887.0	887.3	887.0	3042.3	23.7
38	A001	2021-01-02	1400	0.0	887.0	887.1	886.9	2484.4	24.6
39	A001	2021-01-02	1500	0.0	886.7	887.0	886.7	2585.8	25.9

```
In [18]: h = weather_df.columns
# print(h)
new_header=['station', 'date', 'hour', 'precipitation', 'pressure', 'max_preasure', 'min_preasure',
            'radiation', 'temperature', 'dew_point', 'max_temperature', 'min_temperature',
            'max_dew_point', 'min_dew_point', 'max_humidity', 'min_humidity', 'humidity', 'wind_direct',
            'wind_gust', 'wind_speed']
z = dict(zip(h,new_header))
weather_df= weather_df.rename(columns=z)
#weather_df.head(20)
```

```
In [19]: cols = ['precipitation', 'pressure', 'max_preasure', 'min_preasure',
                'radiation', 'temperature', 'dew_point', 'max_temperature', 'min_temperature',
                'max_dew_point', 'min_dew_point', 'max_humidity', 'min_humidity', 'humidity',
                'wind_gust', 'wind_speed']
for c in cols:
    weather_df[c]= weather_df[c].apply(lambda x: np.nan if x<-1000 else x)
```

```
In [20]: weather_df['year'] = [int(x[0:4]) for x in weather_df['date']]
weather_df['month'] = [int(x[5:7]) for x in weather_df['date']]
weather_df['yearmonth'] = [(100*int(x[0:4]) + int(x[5:7])) for x in weather_df['date']]
```

```
In [21]: weather_df['code']=weather_df['station'].map(state_map2)
weather_df.head(20)
```

Out[21]:

	station	date	hour	precipitation	pressure	max_preasure	min_preasure	radiation	temperature	dew_point	...	max_h
0	A001	2021-01-01	0	0.0	887.1	887.1	886.3	NaN	20.9	15.9	...	
1	A001	2021-01-01	100	0.0	887.3	887.5	887.1	NaN	20.5	16.4	...	
2	A001	2021-01-01	200	0.0	887.9	888.0	887.2	NaN	19.6	16.1	...	
3	A001	2021-01-01	300	0.0	887.6	888.0	887.6	NaN	19.4	16.7	...	
4	A001	2021-01-01	400	0.0	887.0	887.6	887.0	NaN	18.7	16.3	...	
5	A001	2021-01-01	500	0.0	887.1	887.2	887.0	NaN	21.0	16.0	...	
6	A001	2021-01-01	600	0.0	886.5	887.1	886.4	NaN	19.4	16.2	...	
7	A001	2021-01-01	700	0.0	886.5	886.5	886.3	NaN	18.5	16.3	...	
8	A001	2021-01-01	800	0.0	886.8	886.8	886.4	NaN	18.9	16.4	...	
9	A001	2021-01-01	900	0.0	887.3	887.3	886.8	NaN	18.3	16.2	...	
10	A001	2021-01-01	1000	0.0	887.9	887.9	887.3	298.2	20.2	17.1	...	
11	A001	2021-01-01	1100	0.0	888.2	888.2	887.8	1203.7	23.2	17.2	...	
12	A001	2021-01-01	1200	0.0	888.6	888.6	888.2	1664.9	24.9	18.0	...	
13	A001	2021-01-01	1300	0.0	888.7	888.7	888.6	2851.7	25.9	15.3	...	
14	A001	2021-01-01	1400	0.0	888.5	888.7	888.4	3115.7	27.0	15.9	...	
15	A001	2021-01-01	1500	0.0	888.2	888.5	888.2	2970.1	28.1	15.9	...	
16	A001	2021-01-01	1600	0.0	887.5	888.2	887.5	3528.2	28.8	14.3	...	
17	A001	2021-01-01	1700	0.0	886.6	887.5	886.6	3372.1	28.5	13.9	...	
18	A001	2021-01-01	1800	1.2	886.0	886.6	886.0	777.3	21.3	17.2	...	
19	A001	2021-01-01	1900	0.2	885.8	886.0	885.8	700.5	23.0	15.6	...	

20 rows × 24 columns

```
In [22]: sc_df = None
```

```
In [23]: dataset_mean = weather_df.groupby(['yearmonth', 'code']).mean()
dataset_mean.to_csv("weather_mean.csv", sep=';')
dataset_mean = None
```

```
In [24]: dataset_max = weather_df.groupby(['yearmonth', 'code']).max()
dataset_max.to_csv("weather_max.csv", sep=';')
dataset_max = None
```

```
In [25]: dataset_min = weather_df.groupby(['yearmonth', 'code']).min()
dataset_min.to_csv("weather_min.csv", sep=';')
dataset_min = None
```

```
In [26]: dataset_count = weather_df.groupby(['yearmonth', 'code']).count()
dataset_count.to_csv("weather_count.csv", sep=';')
dataset_count = None
```

```
In [27]: code_data      = weather_df['code']
yearmonth_data = weather_df['yearmonth']
dataset_nan = weather_df.isna().astype(int)
dataset_nan['code'] = code_data
dataset_nan['yearmonth'] = yearmonth_data

code_data = None
yearmonth_data = None
```

```
In [28]: weather_df = None
```

```
In [29]: dataset_nan_count = dataset_nan.groupby(['yearmonth', 'code']).sum()
```

```
In [30]: dataset_nan_count.to_csv("weather_nan_count.csv", sep=';')
dataset_nan_count = None
dataset_nan = None
```

Combine weather data to one file

```
In [32]: dataset_mean = pd.read_csv("weather_mean.csv", delimiter=";")
```

```
In [33]: dataset_min = pd.read_csv("weather_min.csv", delimiter=";")
```

```
In [34]: dataset_max = pd.read_csv("weather_max.csv", delimiter=";")
```

```
In [35]: dataset_count = pd.read_csv("weather_count.csv", delimiter=";")
```

```
In [36]: dataset_nan_count = pd.read_csv("weather_nan_count.csv", delimiter=";")
```

```
In [37]: print(dataset_mean)
```

	yearmonth	code	hour	precipitation	pressure	max_preature	\
0	200005	AM	1150	0.108621	1004.492026	1004.611944	
1	200005	BA	1150	NaN	NaN	NaN	
2	200005	DF	1150	0.000000	888.052408	888.264933	
3	200005	RJ	1150	0.097595	1013.941924	1014.193127	
4	200006	AM	1150	0.066667	1004.887222	1005.166806	
...	
5690	202101	RS	1150	0.270264	970.451039	970.744835	
5691	202101	SC	1150	0.387190	937.037016	937.239039	
5692	202101	SE	1150	0.054829	986.552973	986.786306	
5693	202101	SP	1150	0.194781	943.513852	943.731830	
5694	202101	TO	1150	0.212583	978.623905	978.909113	

	min_preature	radiation	temperature	dew_point	...	max_dew_point	\
0	1003.992974	1362.209302	26.237716	23.545690	...	23.935831	
1	NaN	NaN	NaN	NaN	...	NaN	
2	887.849711	1590.901887	20.258767	11.701927	...	12.252601	
3	1013.714777	1128.380952	20.402062	15.879038	...	16.307216	
4	1004.598047	1541.997199	26.419861	23.185417	...	23.603068	
...	
5690	970.165243	1483.619544	23.093614	18.060220	...	18.631668	
5691	936.752187	1089.517891	21.328842	18.421613	...	18.890421	
5692	986.317057	1550.354050	26.211219	19.465738	...	20.047545	
5693	943.204353	1440.642286	23.425897	18.375079	...	18.997354	
5694	978.279720	1361.118030	26.152189	21.711326	...	22.327716	

	min_dew_point	max_humidity	min_humidity	humidity	wind_direction	\
0	23.119204	88.765808	82.957845	85.922414	-2126.811594	
1	NaN	NaN	NaN	NaN	-9999.000000	
2	11.174374	63.485549	57.059730	60.231214	-1215.966667	
3	15.421649	80.542955	74.474227	77.676976	-5068.998333	
4	22.774477	86.107392	80.301255	83.377778	115.451389	
...	
5690	17.510556	78.888372	73.354783	76.256422	153.528814	
5691	17.957523	87.377891	82.595190	85.177888	169.205421	
5692	18.891817	71.804740	65.663939	68.765269	97.635049	
5693	17.790187	78.677669	72.363707	75.497656	183.709829	
5694	21.115645	81.759041	74.809441	78.476122	170.169940	

	wind_gust	wind_speed	year	month
0	5.058216	0.936853	2000	5
1	NaN	NaN	2000	5
2	4.277649	2.073603	2000	5
3	4.377273	1.953952	2000	5
4	4.777514	1.028889	2000	6
...
5690	5.861439	2.424934	2021	1
5691	4.991402	2.081546	2021	1
5692	3.253888	1.388496	2021	1
5693	4.624024	1.726921	2021	1
5694	3.459843	1.184579	2021	1

[5695 rows x 22 columns]

```
In [38]: dataset_weather = dataset_mean.copy()
```

```
In [39]: dataset_weather.drop(columns=['hour'])
```

Out[39]:

	yearmonth	code	precipitation	pressure	max_preasure	min_preasure	radiation	temperature	dew_point	max_
0	200005	AM	0.108621	1004.492026	1004.611944	1003.992974	1362.209302	26.237716	23.545690	
1	200005	BA	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2	200005	DF	0.000000	888.052408	888.264933	887.849711	1590.901887	20.258767	11.701927	
3	200005	RJ	0.097595	1013.941924	1014.193127	1013.714777	1128.380952	20.402062	15.879038	
4	200006	AM	0.066667	1004.887222	1005.166806	1004.598047	1541.997199	26.419861	23.185417	
...	
5690	202101	RS	0.270264	970.451039	970.744835	970.165243	1483.619544	23.093614	18.060220	
5691	202101	SC	0.387190	937.037016	937.239039	936.752187	1089.517891	21.328842	18.421613	
5692	202101	SE	0.054829	986.552973	986.786306	986.317057	1550.354050	26.211219	19.465738	
5693	202101	SP	0.194781	943.513852	943.731830	943.204353	1440.642286	23.425897	18.375079	
5694	202101	TO	0.212583	978.623905	978.909113	978.279720	1361.118030	26.152189	21.711326	

5695 rows × 21 columns

```
In [40]: dataset_weather['max_preasure'] = dataset_max['max_preasure']
dataset_weather['min_preasure'] = dataset_min['min_preasure']
dataset_weather['max_temperature'] = dataset_max['max_temperature']
dataset_weather['min_temperature'] = dataset_min['min_temperature']
dataset_weather['max_dew_point'] = dataset_max['max_dew_point']
dataset_weather['min_dew_point'] = dataset_min['min_dew_point']
```

```
In [41]: dataset_weather.head(20)
```

Out[41]:

	yearmonth	code	hour	precipitation	pressure	max_preasure	min_preasure	radiation	temperature	dew_point
0	200005	AM	1150	0.108621	1004.492026	1009.4	1000.2	1362.209302	26.237716	23.545690
1	200005	BA	1150	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	200005	DF	1150	0.000000	888.052408	893.2	884.4	1590.901887	20.258767	11.701927
3	200005	RJ	1150	0.097595	1013.941924	1019.6	1006.6	1128.380952	20.402062	15.879038
4	200006	AM	1150	0.066667	1004.887222	1009.1	1000.4	1541.997199	26.419861	23.185417
5	200006	BA	1150	NaN	NaN	NaN	NaN	NaN	NaN	NaN
6	200006	DF	1150	0.000000	889.278653	893.5	886.0	1556.682857	19.364183	9.880372
7	200006	RJ	1150	0.003611	1014.639722	1022.6	1006.6	1097.097222	21.710000	15.218333
8	200007	AM	1150	0.144280	1005.176043	1011.2	1000.2	1404.256065	25.800000	22.401615
9	200007	BA	1150	NaN	NaN	NaN	NaN	NaN	NaN	NaN
10	200007	DF	1150	0.000269	888.613978	893.2	884.7	1491.048387	19.102285	9.539919
11	200007	RJ	1150	0.103495	1015.330645	1023.4	1005.7	948.951482	19.630914	14.334946
12	200008	AM	1150	0.079032	1004.761559	1010.0	999.6	1689.431635	26.854973	22.629973
13	200008	BA	1150	NaN	NaN	NaN	NaN	NaN	NaN	NaN
14	200008	DF	1150	0.070699	888.634409	892.6	881.2	1540.715426	21.476478	8.532392
15	200008	RJ	1150	0.063477	1015.681499	1023.8	1001.7	1213.551282	20.754386	14.555821
16	200009	AM	1150	0.283728	1003.505621	1008.8	998.5	1599.811429	27.208580	22.730325
17	200009	BA	1150	NaN	NaN	NaN	NaN	NaN	NaN	NaN
18	200009	DF	1150	0.112061	887.563511	892.6	883.2	1697.347692	21.478015	13.077099
19	200009	RJ	1150	0.076821	1014.412086	1025.1	1002.4	1239.479751	22.133775	17.206457

20 rows × 22 columns

```
In [42]: dataset_weather.to_csv("amazon_weather.csv", sep=";")
```

```
In [43]: dataset_count.drop(columns=['station', 'date', 'hour'])
```

Out[43]:

	yearmonth	code	precipitation	pressure	max_preasure	min_preasure	radiation	temperature	dew_point	max_tempe
0	200005	AM	464	464	427	427	215	464	464	
1	200005	BA	0	0	0	0	0	0	0	
2	200005	DF	519	519	519	519	265	519	519	
3	200005	RJ	291	291	291	291	147	291	291	
4	200006	AM	720	720	717	717	357	720	720	
...	
5690	202101	RS	25259	25259	25255	25255	14925	25259	23959	
5691	202101	SC	15488	15488	15477	15477	9994	15488	14362	
5692	202101	SE	3717	3717	3717	3717	2074	3717	3552	
5693	202101	SP	19275	19932	19893	19893	12704	19933	19409	
5694	202101	TO	7057	7057	7012	7012	4620	7057	6931	

5695 rows × 21 columns

```
In [44]: dataset_count['month'] = dataset_count['yearmonth']%100
dataset_count['year'] = dataset_count['yearmonth']//100
dataset_count['type'] = 'count'
dataset_count.head(20)
```

Out[44]:

	yearmonth	code	station	date	hour	precipitation	pressure	max_preasure	min_preasure	radiation	...	min_dew_poin
0	200005	AM	552	552	552	464	464	427	427	215	...	42
1	200005	BA	456	456	456	0	0	0	0	0	...	0
2	200005	DF	600	600	600	519	519	519	519	265	...	519
3	200005	RJ	600	600	600	291	291	291	291	147	...	29
4	200006	AM	720	720	720	720	720	717	717	357	...	717
5	200006	BA	720	720	720	0	0	0	0	0	...	0
6	200006	DF	720	720	720	698	698	650	650	350	...	650
7	200006	RJ	720	720	720	720	720	720	720	360	...	720
8	200007	AM	744	744	744	743	743	743	743	371	...	743
9	200007	BA	744	744	744	0	0	0	0	0	...	0
10	200007	DF	744	744	744	744	744	744	744	372	...	744
11	200007	RJ	744	744	744	744	744	744	744	371	...	744
12	200008	AM	744	744	744	744	744	744	744	373	...	744
13	200008	BA	744	744	744	0	0	0	0	0	...	0
14	200008	DF	744	744	744	744	744	743	743	376	...	744
15	200008	RJ	744	744	744	627	627	626	626	312	...	627
16	200009	AM	720	720	720	676	676	675	675	350	...	676
17	200009	BA	720	720	720	0	0	0	0	0	...	0
18	200009	DF	720	720	720	655	655	654	654	325	...	655
19	200009	RJ	720	720	720	604	604	604	604	321	...	604

20 rows × 25 columns

In [45]: dataset_nan_count.head(20)

Out[45]:

	yearmonth	code	station	date	hour	precipitation	pressure	max_preasure	min_preasure	radiation	...	max_dew_poir
0	200005	AM	0	0	0	88	88	125	125	337	...	12
1	200005	BA	0	0	0	456	456	456	456	456	...	45
2	200005	DF	0	0	0	81	81	81	81	335	...	8
3	200005	RJ	0	0	0	309	309	309	309	453	...	30
4	200006	AM	0	0	0	0	0	3	3	363	...	
5	200006	BA	0	0	0	720	720	720	720	720	...	72
6	200006	DF	0	0	0	22	22	70	70	370	...	7
7	200006	RJ	0	0	0	0	0	0	0	360	...	
8	200007	AM	0	0	0	1	1	1	1	373	...	
9	200007	BA	0	0	0	744	744	744	744	744	...	74
10	200007	DF	0	0	0	0	0	0	0	372	...	
11	200007	RJ	0	0	0	0	0	0	0	373	...	
12	200008	AM	0	0	0	0	0	0	0	371	...	
13	200008	BA	0	0	0	744	744	744	744	744	...	74
14	200008	DF	0	0	0	0	0	1	1	368	...	
15	200008	RJ	0	0	0	117	117	118	118	432	...	11
16	200009	AM	0	0	0	44	44	45	45	370	...	4
17	200009	BA	0	0	0	720	720	720	720	720	...	72
18	200009	DF	0	0	0	65	65	66	66	395	...	6
19	200009	RJ	0	0	0	116	116	116	116	399	...	11

20 rows × 24 columns

```
In [46]: dataset_nan_count.drop(columns=['station', 'date', 'hour'])
dataset_nan_count['month'] = dataset_nan_count['yearmonth']%100
dataset_nan_count['year'] = dataset_nan_count['yearmonth']//100
dataset_nan_count['type'] = 'nan'
dataset_nan_count.head(20)
```

Out[46]:

	yearmonth	code	station	date	hour	precipitation	pressure	max_preasure	min_preasure	radiation	...	min_dew_poin
0	200005	AM	0	0	0	88	88	125	125	337	...	125
1	200005	BA	0	0	0	456	456	456	456	456	...	456
2	200005	DF	0	0	0	81	81	81	81	335	...	81
3	200005	RJ	0	0	0	309	309	309	309	453	...	309
4	200006	AM	0	0	0	0	0	3	3	363	...	0
5	200006	BA	0	0	0	720	720	720	720	720	...	720
6	200006	DF	0	0	0	22	22	70	70	370	...	22
7	200006	RJ	0	0	0	0	0	0	0	360	...	0
8	200007	AM	0	0	0	1	1	1	1	373	...	1
9	200007	BA	0	0	0	744	744	744	744	744	...	744
10	200007	DF	0	0	0	0	0	0	0	372	...	0
11	200007	RJ	0	0	0	0	0	0	0	373	...	0
12	200008	AM	0	0	0	0	0	0	0	371	...	0
13	200008	BA	0	0	0	744	744	744	744	744	...	744
14	200008	DF	0	0	0	0	0	1	1	368	...	0
15	200008	RJ	0	0	0	117	117	118	118	432	...	117
16	200009	AM	0	0	0	44	44	45	45	370	...	44
17	200009	BA	0	0	0	720	720	720	720	720	...	720
18	200009	DF	0	0	0	65	65	66	66	395	...	65
19	200009	RJ	0	0	0	116	116	116	116	399	...	116

20 rows × 25 columns

```
In [47]: amazon_count = dataset_count.copy()
```

```
In [48]: amazon_count = amazon_count.append(dataset_nan_count)
```



```
In [49]: print(amazon_count)
```

	yearmonth	code	station	date	hour	precipitation	pressure	\
0	200005	AM	552	552	552	464	464	
1	200005	BA	456	456	456	0	0	
2	200005	DF	600	600	600	519	519	
3	200005	RJ	600	600	600	291	291	
4	200006	AM	720	720	720	720	720	
...	
5690	202101	RS	0	0	0	7477	7477	
5691	202101	SC	0	0	0	2368	2368	
5692	202101	SE	0	0	0	1491	1491	
5693	202101	SP	0	0	0	12717	12060	
5694	202101	TO	0	0	0	7823	7823	

	max_preasure	min_preasure	radiation	...	min_dew_point	max_humidity	\
0	427	427	215	...	427	427	
1	0	0	0	...	0	0	
2	519	519	265	...	519	519	
3	291	291	147	...	291	291	
4	717	717	357	...	717	717	
...	
5690	7481	7481	17811	...	8815	8880	
5691	2379	2379	7862	...	3599	3545	
5692	1491	1491	3134	...	1664	1664	
5693	12099	12099	19288	...	12641	12633	
5694	7868	7868	10260	...	7996	7995	

	min_humidity	humidity	wind_direction	wind_gust	wind_speed	year	\
0	427	464	552	426	464	2000	
1	0	0	456	0	0	2000	
2	519	519	600	519	519	2000	
3	291	291	600	286	291	2000	
4	717	720	720	716	720	2000	
...	
5690	8882	8873	9119	8491	8464	2021	
5691	3553	3493	3393	3410	3393	2021	
5692	1664	1655	2235	2237	2235	2021	
5693	12633	12583	13465	13527	13464	2021	
5694	7995	7949	8719	8749	8713	2021	

	month	type
0	5	count
1	5	count
2	5	count
3	5	count
4	6	count
...
5690	1	nan
5691	1	nan
5692	1	nan
5693	1	nan
5694	1	nan

[11390 rows x 25 columns]

```
In [50]: amazon_count.to_csv("amazon_count.csv", sep=";")
```

Group data

```
In [51]: amazon_weather = pd.read_csv("amazon_weather.csv", delimiter=";")
amazon_weather.drop(amazon_weather.columns[amazon_weather.columns.str.contains('unnamed',case = False)
amazon_weather.head(10)
```

Out[51]:

	yearmonth	code	hour	precipitation	pressure	max_preasure	min_preasure	radiation	temperature	dew_point	..
0	200005	AM	1150	0.108621	1004.492026	1009.4	1000.2	1362.209302	26.237716	23.545690	..
1	200005	BA	1150	NaN	NaN	NaN	NaN	NaN	NaN	NaN	..
2	200005	DF	1150	0.000000	888.052408	893.2	884.4	1590.901887	20.258767	11.701927	..
3	200005	RJ	1150	0.097595	1013.941924	1019.6	1006.6	1128.380952	20.402062	15.879038	..
4	200006	AM	1150	0.066667	1004.887222	1009.1	1000.4	1541.997199	26.419861	23.185417	..
5	200006	BA	1150	NaN	NaN	NaN	NaN	NaN	NaN	NaN	..
6	200006	DF	1150	0.000000	889.278653	893.5	886.0	1556.682857	19.364183	9.880372	..
7	200006	RJ	1150	0.003611	1014.639722	1022.6	1006.6	1097.097222	21.710000	15.218333	..
8	200007	AM	1150	0.144280	1005.176043	1011.2	1000.2	1404.256065	25.800000	22.401615	..
9	200007	BA	1150	NaN	NaN	NaN	NaN	NaN	NaN	NaN	..

10 rows × 22 columns

```
In [52]: a = amazon_weather['code']
b = amazon_weather['yearmonth']
amazon_weather['id'] =amazon_weather[['code','yearmonth']].astype(str).apply(''.join,1)
amazon_weather= amazon_weather.drop(columns=['hour','yearmonth'])
amazon_weather.head(10)
```

Out[52]:

	code	precipitation	pressure	max_preasure	min_preasure	radiation	temperature	dew_point	max_temperature	n
0	AM	0.108621	1004.492026	1009.4	1000.2	1362.209302	26.237716	23.545690	32.4	
1	BA	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
2	DF	0.000000	888.052408	893.2	884.4	1590.901887	20.258767	11.701927	28.9	
3	RJ	0.097595	1013.941924	1019.6	1006.6	1128.380952	20.402062	15.879038	29.8	
4	AM	0.066667	1004.887222	1009.1	1000.4	1541.997199	26.419861	23.185417	32.9	
5	BA	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
6	DF	0.000000	889.278653	893.5	886.0	1556.682857	19.364183	9.880372	27.7	
7	RJ	0.003611	1014.639722	1022.6	1006.6	1097.097222	21.710000	15.218333	33.8	
8	AM	0.144280	1005.176043	1011.2	1000.2	1404.256065	25.800000	22.401615	33.5	
9	BA	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

10 rows × 21 columns

```
In [53]: amazon_fire = pd.read_csv("amazon2.csv", delimiter=";")
amazon_fire.drop(amazon_fire.columns[amazon_fire.columns.str.contains('unnamed',case = False)],axis =
amazon_fire.head(10)
```

Out[53]:

	year	state	month	number	date	month_nbr	yearmonth	code
0	1998	Acre	Janeiro	0.0	1998-01-01	1	199801	AC
1	1999	Acre	Janeiro	0.0	1999-01-01	1	199901	AC
2	2000	Acre	Janeiro	0.0	2000-01-01	1	200001	AC
3	2001	Acre	Janeiro	0.0	2001-01-01	1	200101	AC
4	2002	Acre	Janeiro	0.0	2002-01-01	1	200201	AC
5	2003	Acre	Janeiro	10.0	2003-01-01	1	200301	AC
6	2004	Acre	Janeiro	0.0	2004-01-01	1	200401	AC
7	2005	Acre	Janeiro	12.0	2005-01-01	1	200501	AC
8	2006	Acre	Janeiro	4.0	2006-01-01	1	200601	AC
9	2007	Acre	Janeiro	0.0	2007-01-01	1	200701	AC

```
In [54]: amazon_fire['id'] =amazon_fire[['code','yearmonth']].astype(str).apply('.',join,1)
amazon_fire = amazon_fire.drop(columns=['year', 'month_nbr','yearmonth','date','code'])
amazon_fire.head(10)
```

Out[54]:

	state	month	number	id
0	Acre	Janeiro	0.0	AC199801
1	Acre	Janeiro	0.0	AC199901
2	Acre	Janeiro	0.0	AC200001
3	Acre	Janeiro	0.0	AC200101
4	Acre	Janeiro	0.0	AC200201
5	Acre	Janeiro	10.0	AC200301
6	Acre	Janeiro	0.0	AC200401
7	Acre	Janeiro	12.0	AC200501
8	Acre	Janeiro	4.0	AC200601
9	Acre	Janeiro	0.0	AC200701

```
In [55]: amazon3 = pd.merge(amazon_fire, amazon_weather, how='inner', left_on='id', right_on ='id')
```

```
In [56]: amazon3.head(20)
```

```
Out[56]:
```

	state	month_x	number	id	code	precipitation	pressure	max_preasure	min_preasure	radiation	...	max_
0	Acre	Janeiro	0.0	AC200901	AC	0.306864	987.438694	995.5	978.6	1086.153071	...	
1	Acre	Janeiro	1.0	AC201001	AC	0.324505	986.416071	996.8	976.6	1259.078120	...	
2	Acre	Janeiro	0.0	AC201101	AC	0.237873	982.884571	990.2	977.1	1272.938571	...	
3	Acre	Janeiro	0.0	AC201201	AC	0.351075	985.741962	997.6	977.3	965.069782	...	
4	Acre	Janeiro	0.0	AC201301	AC	0.577901	992.505543	1022.9	977.1	1204.605007	...	
5	Acre	Janeiro	0.0	AC201401	AC	0.411559	987.714561	997.1	978.6	1169.141654	...	
6	Acre	Janeiro	1.0	AC201501	AC	0.396447	988.278551	998.0	979.1	1765.080380	...	
7	Acre	Janeiro	12.0	AC201601	AC	0.303943	986.228245	996.3	976.9	1200.502658	...	
8	Acre	Janeiro	0.0	AC201701	AC	0.484808	987.571083	997.2	978.6	1128.071878	...	
9	Acre	Fevereiro	2.0	AC200902	AC	0.323639	986.738628	994.4	978.2	940.684153	...	
10	Acre	Fevereiro	0.0	AC201002	AC	0.294883	985.342277	994.4	978.7	1235.067427	...	
11	Acre	Fevereiro	0.0	AC201102	AC	0.201241	984.224643	989.8	978.5	1180.156852	...	
12	Acre	Fevereiro	0.0	AC201202	AC	0.501905	985.665073	996.1	977.2	957.682123	...	
13	Acre	Fevereiro	0.0	AC201302	AC	0.329167	990.043296	1049.4	977.6	1170.457826	...	
14	Acre	Fevereiro	0.0	AC201402	AC	0.366386	986.986030	995.9	977.5	1177.370475	...	
15	Acre	Fevereiro	2.0	AC201502	AC	0.393095	987.253869	997.9	977.2	1527.185725	...	
16	Acre	Fevereiro	5.0	AC201602	AC	0.407759	986.993370	997.4	976.6	1238.982471	...	
17	Acre	Fevereiro	1.0	AC201702	AC	0.315607	986.365435	994.2	978.4	1293.083170	...	
18	Acre	Março	1.0	AC200903	AC	0.231317	986.825202	994.9	978.4	984.922993	...	
19	Acre	Março	0.0	AC201003	AC	0.230078	984.847907	991.4	977.6	1256.335148	...	

20 rows × 24 columns

```
In [57]: amazon3 = amazon3.dropna(axis=0)
amazon3.head(10)
```

```
Out[57]:
```

	state	month_x	number	id	code	precipitation	pressure	max_preasure	min_preasure	radiation	...	max_
0	Acre	Janeiro	0.0	AC200901	AC	0.306864	987.438694	995.5	978.6	1086.153071	...	
1	Acre	Janeiro	1.0	AC201001	AC	0.324505	986.416071	996.8	976.6	1259.078120	...	
2	Acre	Janeiro	0.0	AC201101	AC	0.237873	982.884571	990.2	977.1	1272.938571	...	
3	Acre	Janeiro	0.0	AC201201	AC	0.351075	985.741962	997.6	977.3	965.069782	...	
4	Acre	Janeiro	0.0	AC201301	AC	0.577901	992.505543	1022.9	977.1	1204.605007	...	
5	Acre	Janeiro	0.0	AC201401	AC	0.411559	987.714561	997.1	978.6	1169.141654	...	
6	Acre	Janeiro	1.0	AC201501	AC	0.396447	988.278551	998.0	979.1	1765.080380	...	
7	Acre	Janeiro	12.0	AC201601	AC	0.303943	986.228245	996.3	976.9	1200.502658	...	
8	Acre	Janeiro	0.0	AC201701	AC	0.484808	987.571083	997.2	978.6	1128.071878	...	
9	Acre	Fevereiro	2.0	AC200902	AC	0.323639	986.738628	994.4	978.2	940.684153	...	

10 rows × 24 columns

```
In [58]: amazon3.to_csv("amazon3_fire_weather.csv",sep=";",index=False)
```

```
In [59]: amazon3.columns
```

```
Out[59]: Index(['state', 'month_x', 'number', 'id', 'code', 'precipitation', 'pressure',  
               'max_preasure', 'min_preasure', 'radiation', 'temperature', 'dew_point',  
               'max_temperature', 'min_temperature', 'max_dew_point', 'min_dew_point',  
               'max_humidity', 'min_humidity', 'humidity', 'wind_direction',  
               'wind_gust', 'wind_speed', 'year', 'month_y'],  
              dtype='object')
```

```
In [ ]:
```