Explore_Data_EH

May 12, 2024

[165]:
```
import pandas as pd
import altair as alt
from pathlib import Path

TABLE_ID = '80072ned'
TABLE_PATH = Path(f'./data/{TABLE_ID}')

# required for export to pdf with images?
alt.renderers.enable('png')
```

[165]:
```
RendererRegistry.enable('png')
```

1 UWV Exploratory Analysis

Read classification information

According to the metadata, there are types of data on sick leave percentages:

- Accumulated over all economic activities
- Split into different sectors using SBI 2008
- Split on company size (1-10, 10-100 and 100+)

[173]:
```
sbi: pd.DataFrame = pd.read_csv(TABLE_PATH /␣
 ↪f'{TABLE_ID}_BedrijfskenmerkenSBI2008.csv', sep=',')
sbi['Key'] = sbi['Key'].astype('category')
sbi.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 39 entries, 0 to 38
Data columns (total 4 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Key              39 non-null     category
 1   Title            39 non-null     object
 2   Description      39 non-null     object
 3   CategoryGroupID  39 non-null     int64
dtypes: category(1), int64(1), object(2)
memory usage: 2.4+ KB
```

[169]:
```
sbi
```

[169]:

	Key	Title	Description	CategoryGroupID
0	T001081	A-U Alle economische activiteiten	Alle economische activiteiten \r\nDeze categor…	1
1	301000	A Landbouw, bosbouw en visserij	Landbouw, bosbouw en visserij \r\nDeze sectie …	2
2	305700	B Delfstoffenwinning	Winning van delfstoffen \r\nDeze sectie omvat:…	3
3	300003	B-F Nijverheid en energie		
4	307500	C Industrie		
5	307610	10-12 Voedings-, genotmiddelenindustrie		
6	317105	17-18 Papier- en grafische industrie		
7	320005	19-22 Raffinaderijen en chemie		
8	328110	24-30, 33 Metaal-elektro industrie		
9	346600	D Energievoorziening		
10	348000	E Waterbedrijven en afvalbeheer		
11	350000	F Bouwnijverheid		
12	300007	G-N Commerciële dienstverlening		
13	354200	G Handel		
14	354300	45 Autohandel en -reparatie		
15	356900	46 Groothandel en handelsbemiddeling		
16	371600	47 Detailhandel (niet in auto's)		
17	383100	H Vervoer en opslag		
18	383200	49 Vervoer over land		
19	389100	I Horeca		
20	391600	J Informatie en communicatie		
21	396300	K Financiële dienstverlening		
22	402000	L Verhuur en handel van onroerend goed		
23	403300	M Specialistische zakelijke diensten		
24	410200	N Verhuur en overige zakelijke diensten		
25	415300	812 Schoonmaakbedrijven		
26	300013	O-U Niet-commerciële dienstverlening		
27	417400	O Openbaar bestuur en overheidsdiensten		
28	419000	P Onderwijs		
29	422400	Q Gezondheids- en welzijnszorg		
30	422500	86 Gezondheidszorg		
31	422600	861 Ziekenhuizen		
32	425300	87 Verpleging en zorg met overnachting		
33	426600	88 Welzijnszorg zonder overnachting		
34	428100	R Cultuur, sport en recreatie		
35	435500	S Overige dienstverlening		
36	WP19078	1 tot 10 werkzame personen		
37	WP19091	10 tot 100 werkzame personen		
38	WP19098	100 of meer werkzame personen		

```
3   Nijverheid en energie \r\nDeze categorie is ee…              2
4   Industrie \r\nDeze sectie omvat: \r\n- de mech…              3
5   Vervaardiging van voedingsmiddelen, dranken en…              4
6   Vervaardiging van papier, karton en papier- en…              4
7   Raffinaderijen, chemie, vervaardiging van farm…              4
8   Metaal-elektro industrie\r\nDeze categorie is …              4
9   Productie en distributie van en handel in elek…              3
10  Winning en distributie van water; afval- en af…              3
11  Bouwnijverheid \r\nDeze sectie omvat: \r\n- al…              3
12  Commerciële dienstverlening \r\nDeze categorie…              2
13  Groot- en detailhandel; reparatie van auto's \…              3
14  Handel in en reparatie van auto's, motorfietse…              4
15  Groothandel en handelsbemiddeling (niet in aut…              4
16  Detailhandel (niet in auto's) \r\nTot de detai…              4
17  Vervoer en opslag \r\nDeze sectie omvat: \r\n-…              3
18  Vervoer over land \r\nDeze afdeling omvat: \r\…              4
19  Logies-, maaltijd- en drankverstrekking \r\nDe…              3
20  Informatie en communicatie \r\nDeze sectie omv…              3
21  Financiële instellingen \r\nDeze sectie omvat:…              3
22  Verhuur van en handel in onroerend goed \r\nDe…              3
23  Advisering, onderzoek en overige specialistisc…              3
24  Verhuur van roerende goederen en overige zakel…              3
25  Reiniging \r\nDeze klasse omvat: \r\n- glazenw…              4
26  Niet-commerciële dienstverlening \r\nDeze cate…              2
27  Openbaar bestuur, overheidsdiensten en verplic…              3
28  Onderwijs \r\nDeze sectie omvat: \r\n- alle vo…              3
29  Gezondheids- en welzijnszorg \r\nDeze sectie o…              3
30  Gezondheidszorg \r\nDeze afdeling omvat de gro…              4
31  Ziekenhuizen en geestelijke gezondheids- en ve…              4
32  Verpleging, verzorging en begeleiding met over…              4
33  Maatschappelijke dienstverlening zonder overna…              4
34  Cultuur, sport en recreatie \r\nDeze sectie om…              3
35  Overige dienstverlening \r\nDeze sectie omvat …              3
36  Het aantal "werkzame personen" bestaat uit: \r…              5
37  Het aantal "werkzame personen" bestaat uit: \r…              5
38  Het aantal "werkzame personen" bestaat uit: \r…              5
```

[174]:
```
sbi.CategoryGroupID.value_counts()
```

[174]:
```
CategoryGroupID
3    18
4    13
2     4
5     3
1     1
Name: count, dtype: int64
```

[175]:
```
groups = pd.read_csv(TABLE_PATH / f'{TABLE_ID}_CategoryGroups.csv', sep=',')
groups
```

[175]:

	ID	DimensionKey	Title	Description	ParentID
0	0	BedrijfskenmerkenSBI2008	Bedrijfstakken (SBI 2008) en -grootte	De Nederlandse hiërarchische indeling van econ…	NaN
1	1	BedrijfskenmerkenSBI2008	Totaal	NaN	0.0
2	2	BedrijfskenmerkenSBI2008	Bedrijfssector	NaN	0.0
3	3	BedrijfskenmerkenSBI2008	Bedrijfstak	NaN	0.0
4	4	BedrijfskenmerkenSBI2008	Bedrijfsklasse	NaN	0.0
5	5	BedrijfskenmerkenSBI2008	Bedrijfsgrootte	NaN	0.0

1.1 Read data

The CSV file contains four columns (ID, BedrijfskenmerkenSBI2008, Perioden and Ziekteverzuim-percentage_1)

There are rows with only a spaces and a period which seem to indicate missing data in the Untyped dataset The typed dataset would list this as empty.

See: https://opendata.cbs.nl/statline/portal.html?_la=nl&_catalog=CBS&tableId=80072ned&_theme=178#, pages 23 and 24.

[177]:
```
uwv = pd.read_csv(TABLE_PATH / f'{TABLE_ID}_UntypedDataSet.csv', sep=',',␣
 ↪na_values='       .')

uwv.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5460 entries, 0 to 5459
Data columns (total 4 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   ID                         5460 non-null   int64
 1   BedrijfskenmerkenSBI2008   5460 non-null   object
 2   Perioden                   5460 non-null   object
 3   Ziekteverzuimpercentage_1  5150 non-null   float64
dtypes: float64(1), int64(1), object(2)
memory usage: 170.8+ KB
```

1.2 Data fixing

Split the "Perioden" into three columns: Jaar (Year), RijType (is this row a quarterly row), Volgnummer (for quarterly rows, this indicates which quarter).

[178]:
```
uwv[['Jaar', 'VerzuimType', 'Kwartaalnummer']] = uwv['Perioden'].str.
 ↪extract(r'(\d+)(KW|JJ)(\d+)', expand=True)

uwv['Jaar'] = uwv['Jaar'].astype(int)
uwv['Kwartaalnummer'] = uwv['Kwartaalnummer'].astype(int)
uwv['VerzuimType'] = uwv['VerzuimType'].astype('category')

uwv['Kwartaal'] = uwv.apply(lambda row: row['Jaar'] * 10 +␣
 ↪row['Kwartaalnummer'], axis=1)

uwv['BedrijfskenmerkenSBI2008'] = uwv['BedrijfskenmerkenSBI2008'].
 ↪astype('category')

uwv
```

[178]:

	ID	BedrijfskenmerkenSBI2008	Perioden	Ziekteverzuimpercentage_1	\
0	0	T001081	1996KW01	5.5	
1	1	T001081	1996KW02	4.6	
2	2	T001081	1996KW03	4.0	
3	3	T001081	1996KW04	4.7	
4	4	T001081	1996JJ00	4.7	
…	…	…	…	…	
5455	5455	WP19098	2023KW01	6.5	
5456	5456	WP19098	2023KW02	5.7	
5457	5457	WP19098	2023KW03	5.5	
5458	5458	WP19098	2023KW04	6.4	
5459	5459	WP19098	2023JJ00	6.0	

	Jaar	VerzuimType	Kwartaalnummer	Kwartaal
0	1996	KW	1	19961
1	1996	KW	2	19962
2	1996	KW	3	19963
3	1996	KW	4	19964
4	1996	JJ	0	19960
…	…	…	…	…
5455	2023	KW	1	20231
5456	2023	KW	2	20232
5457	2023	KW	3	20233
5458	2023	KW	4	20234
5459	2023	JJ	0	20230

[5460 rows x 8 columns]

1.3 Get train and test

We will use 2022 and up as the final test data. All prior tot 2022 will be training data. To test the trained model, we wil use 2021. So we get three splits:

- All data prior to 2021 is the real train data. This is the data to perform exploratory data analysis on.
- All data from 2021 wil be the test set to test our trained models on.
- When we are really done, 2022 and onwards will be the final test set.

Additionally, we will only use the quarterly numbers (Verzuimtype = 'KW')

[179]:
```
uwv_test = uwv[(uwv['Jaar'] >= 2022) & (uwv['VerzuimType'] == 'KW')]
uwv_train = uwv[(uwv['Jaar'] < 2021) & (uwv['VerzuimType'] == 'KW')]
uwv_train_test = uwv[(uwv['Jaar'] == 2021) & (uwv['VerzuimType'] == 'KW')]
```

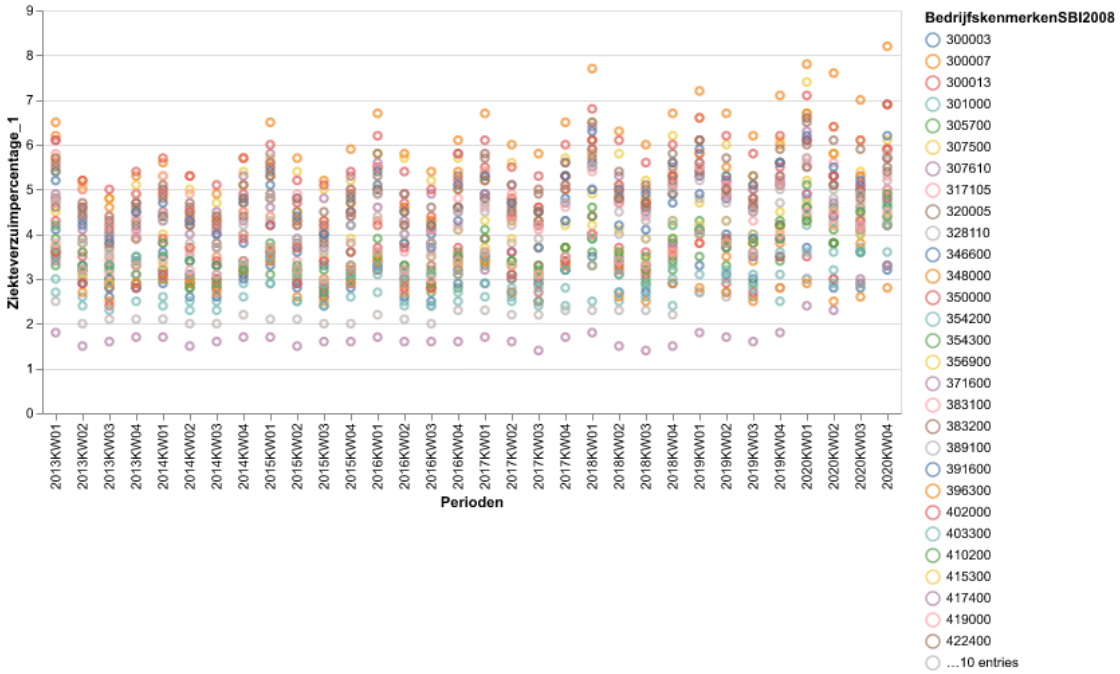
[180]:
```
uwv_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 3900 entries, 0 to 5443
Data columns (total 8 columns):
 #   Column                      Non-Null Count  Dtype
---  ------                      --------------  -----
 0   ID                          3900 non-null   int64
 1   BedrijfskenmerkenSBI2008    3900 non-null   category
 2   Perioden                    3900 non-null   object
 3   Ziekteverzuimpercentage_1   3652 non-null   float64
 4   Jaar                        3900 non-null   int64
 5   VerzuimType                 3900 non-null   category
 6   Kwartaalnummer              3900 non-null   int64
 7   Kwartaal                    3900 non-null   int64
dtypes: category(2), float64(1), int64(4), object(1)
memory usage: 222.4+ KB
```

[181]:
```
uwv_train_2012plus = uwv_train[uwv_train['Jaar'] > 2012]
```

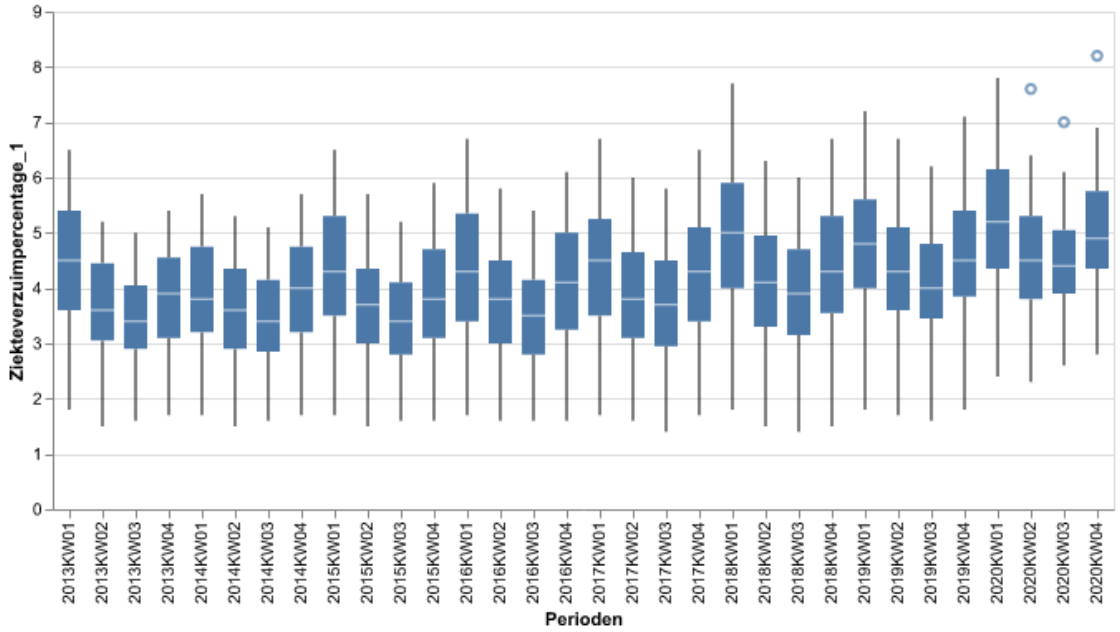
[182]:
```
alt.Chart(uwv_train_2012plus).mark_point().encode(
    x='Perioden',
    y='Ziekteverzuimpercentage_1',
    color='BedrijfskenmerkenSBI2008'
)
```

[182]:



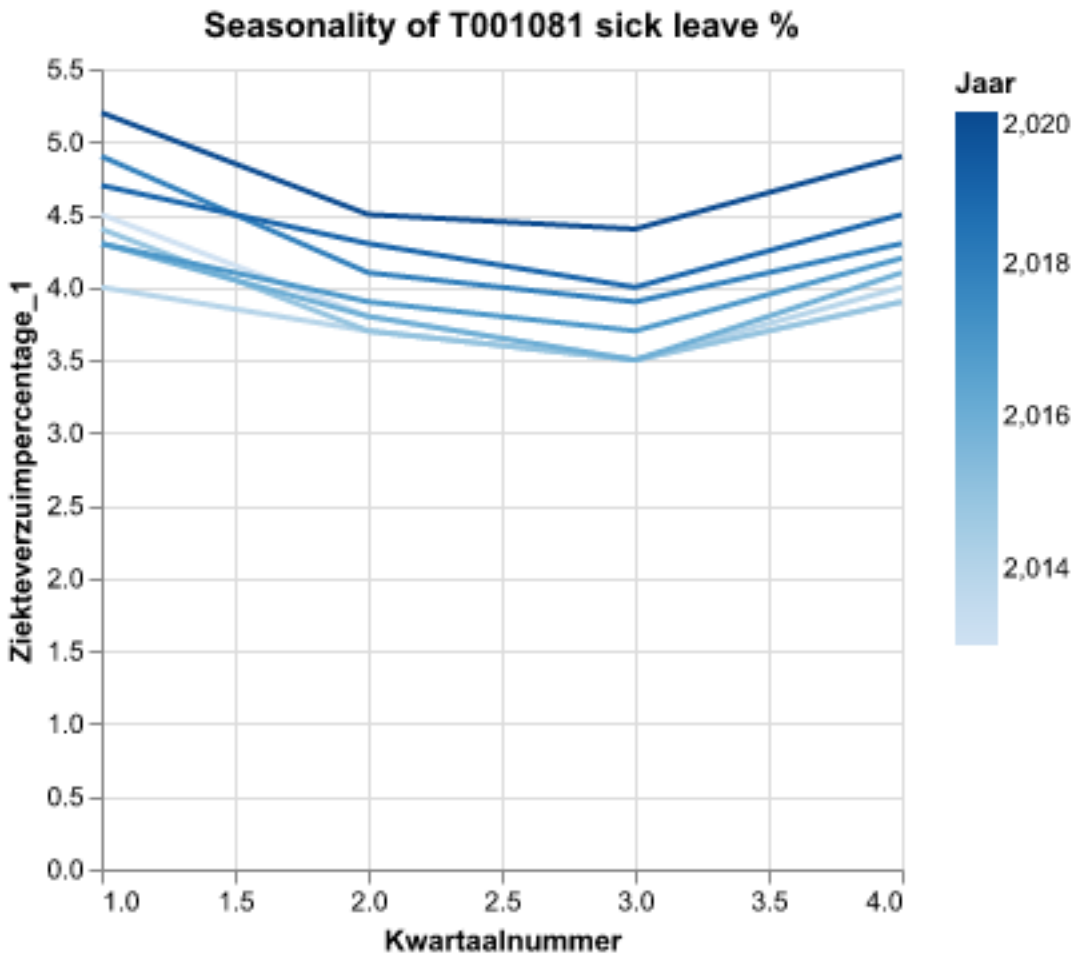
[183]:
```
alt.Chart(uwv_train_2012plus).mark_boxplot().encode(
    x='Perioden',
    y='Ziekteverzuimpercentage_1',
)
```

[183]:



[184]:
```
alt.
  ↪Chart(uwv_train_2012plus[uwv_train_2012plus['BedrijfskenmerkenSBI2008']=='T001081']).
  ↪mark_line().encode(
     x='Kwartaalnummer',
     y='Ziekteverzuimpercentage_1',
     color='Jaar'
).properties(title='Seasonality of T001081 sick leave %')
```

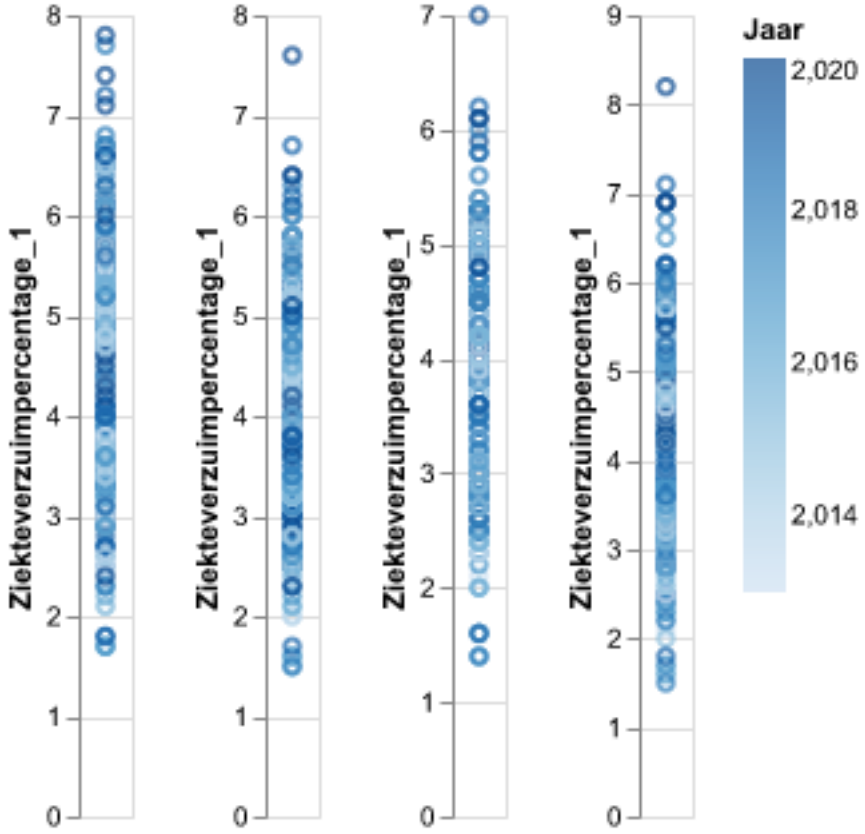
[184]:



[185]:
```
alt.Chart(uwv_train_2012plus[uwv_train_2012plus['Kwartaalnummer'] == 1]).
  ↪mark_point().encode(
     y='Ziekteverzuimpercentage_1',
     color='Jaar'
) | alt.Chart(uwv_train_2012plus[uwv_train_2012plus['Kwartaalnummer'] == 2]).
  ↪mark_point().encode(
     y='Ziekteverzuimpercentage_1',
     color='Jaar'
```

```
) | alt.Chart(uwv_train_2012plus[uwv_train_2012plus['Kwartaalnummer'] == 3]).
  ↪mark_point().encode(
    y='Ziekteverzuimpercentage_1',
    color='Jaar'
) | alt.Chart(uwv_train_2012plus[uwv_train_2012plus['Kwartaalnummer'] == 4]).
  ↪mark_point().encode(
    y='Ziekteverzuimpercentage_1',
    color='Jaar'
)
```

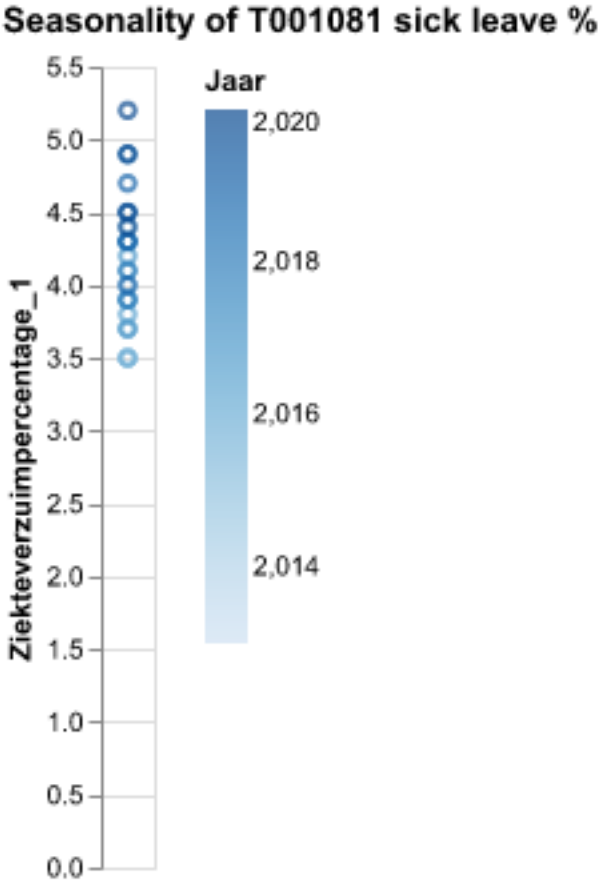
[185]:



[186]:
```
alt.
  ↪Chart(uwv_train_2012plus[uwv_train_2012plus['BedrijfskenmerkenSBI2008']=='T001081']).
  ↪mark_point().encode(
    x=alt.X(alt.repeat("column"), type='ordinal'),
    y='Ziekteverzuimpercentage_1',
    color='Jaar'
).repeat(
    row=['Kwartaalnummer']
).properties(
    title='Seasonality of T001081 sick leave %'
```

```
)
```

[186]:

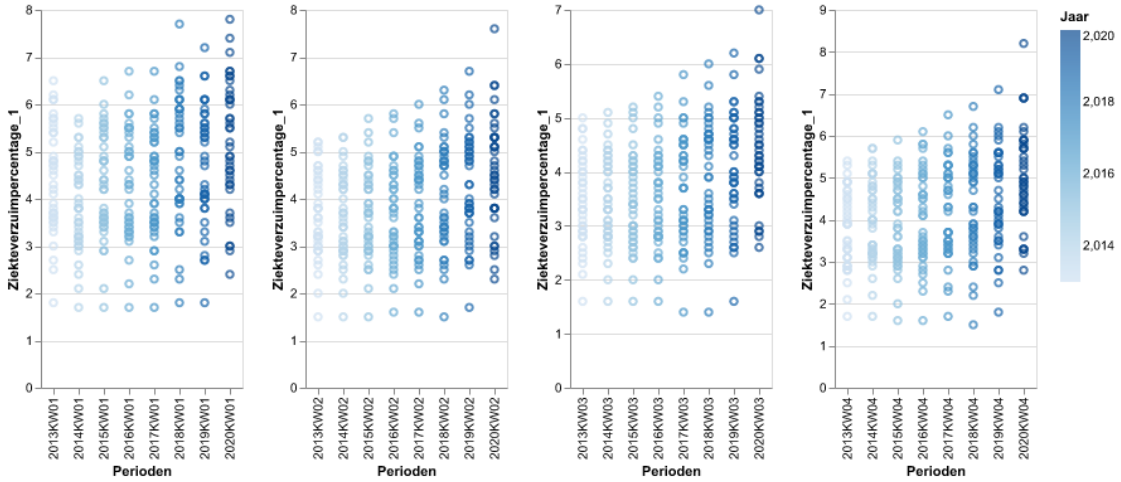


[187]:

```
alt.Chart(uwv_train_2012plus[uwv_train_2012plus['Kwartaalnummer'] == 1]).
 ↪mark_point().encode(
    x='Perioden',
    y='Ziekteverzuimpercentage_1',
    color='Jaar'
) | alt.Chart(uwv_train_2012plus[uwv_train_2012plus['Kwartaalnummer'] == 2]).
 ↪mark_point().encode(
    x='Perioden',
    y='Ziekteverzuimpercentage_1',
    color='Jaar'
) | alt.Chart(uwv_train_2012plus[uwv_train_2012plus['Kwartaalnummer'] == 3]).
 ↪mark_point().encode(
    x='Perioden',
    y='Ziekteverzuimpercentage_1',
    color='Jaar'
) | alt.Chart(uwv_train_2012plus[uwv_train_2012plus['Kwartaalnummer'] == 4]).
 ↪mark_point().encode(
```

```
    x='Perioden',
    y='Ziekteverzuimpercentage_1',
    color='Jaar'
)
```

[187]:



[188]:
```
uwv_train_2012plus_T =␣
↪uwv_train_2012plus[uwv_train_2012plus['BedrijfskenmerkenSBI2008'] ==␣
↪'T001081']
uwv_train_2012plus_T
```

[188]:

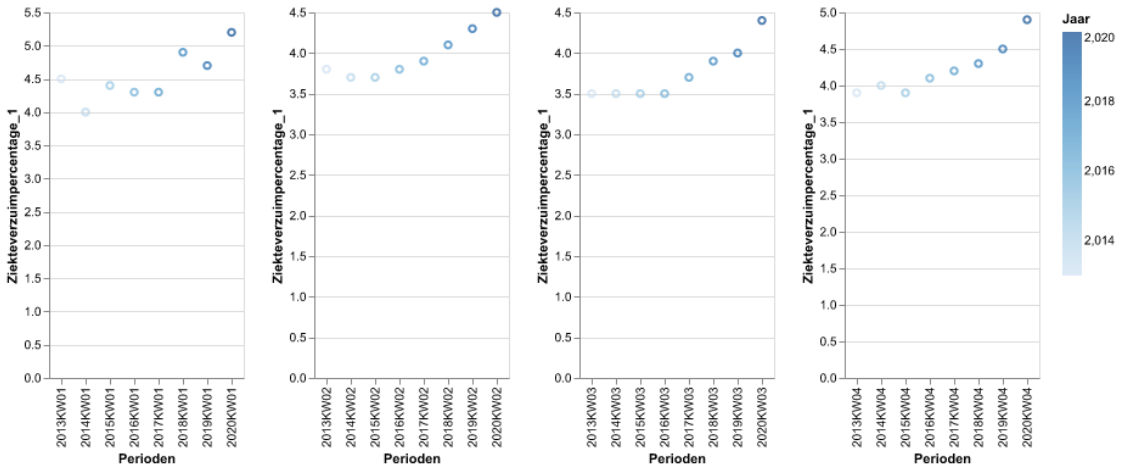
	ID	BedrijfskenmerkenSBI2008	Perioden	Ziekteverzuimpercentage_1	Jaar	\
85	85	T001081	2013KW01	4.5	2013	
86	86	T001081	2013KW02	3.8	2013	
87	87	T001081	2013KW03	3.5	2013	
88	88	T001081	2013KW04	3.9	2013	
90	90	T001081	2014KW01	4.0	2014	
91	91	T001081	2014KW02	3.7	2014	
92	92	T001081	2014KW03	3.5	2014	
93	93	T001081	2014KW04	4.0	2014	
95	95	T001081	2015KW01	4.4	2015	
96	96	T001081	2015KW02	3.7	2015	
97	97	T001081	2015KW03	3.5	2015	
98	98	T001081	2015KW04	3.9	2015	
100	100	T001081	2016KW01	4.3	2016	
101	101	T001081	2016KW02	3.8	2016	
102	102	T001081	2016KW03	3.5	2016	
103	103	T001081	2016KW04	4.1	2016	
105	105	T001081	2017KW01	4.3	2017	
106	106	T001081	2017KW02	3.9	2017	
107	107	T001081	2017KW03	3.7	2017	

108	108	T001081	2017KW04	4.2	2017
110	110	T001081	2018KW01	4.9	2018
111	111	T001081	2018KW02	4.1	2018
112	112	T001081	2018KW03	3.9	2018
113	113	T001081	2018KW04	4.3	2018
115	115	T001081	2019KW01	4.7	2019
116	116	T001081	2019KW02	4.3	2019
117	117	T001081	2019KW03	4.0	2019
118	118	T001081	2019KW04	4.5	2019
120	120	T001081	2020KW01	5.2	2020
121	121	T001081	2020KW02	4.5	2020
122	122	T001081	2020KW03	4.4	2020
123	123	T001081	2020KW04	4.9	2020

	VerzuimType	Kwartaalnummer	Kwartaal
85	KW	1	20131
86	KW	2	20132
87	KW	3	20133
88	KW	4	20134
90	KW	1	20141
91	KW	2	20142
92	KW	3	20143
93	KW	4	20144
95	KW	1	20151
96	KW	2	20152
97	KW	3	20153
98	KW	4	20154
100	KW	1	20161
101	KW	2	20162
102	KW	3	20163
103	KW	4	20164
105	KW	1	20171
106	KW	2	20172
107	KW	3	20173
108	KW	4	20174
110	KW	1	20181
111	KW	2	20182
112	KW	3	20183
113	KW	4	20184
115	KW	1	20191
116	KW	2	20192
117	KW	3	20193
118	KW	4	20194
120	KW	1	20201
121	KW	2	20202
122	KW	3	20203
123	KW	4	20204

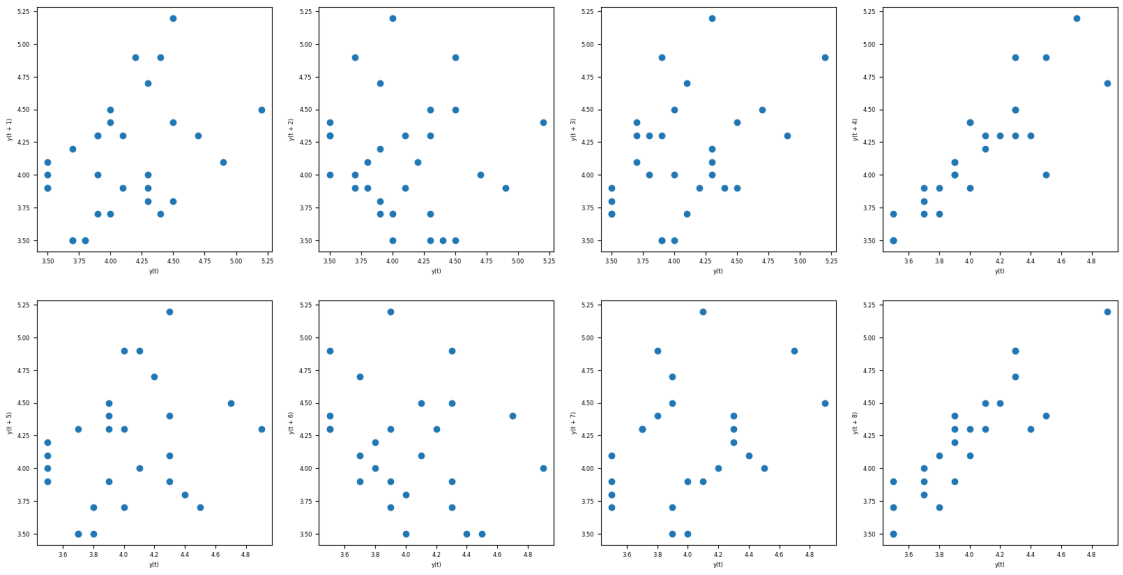
[189]:
```
alt.Chart(uwv_train_2012plus_T[(uwv_train_2012plus_T['Kwartaalnummer'] == 1) ]).
 ↪mark_point().encode(
    x='Perioden',
    y='Ziekteverzuimpercentage_1',
    color='Jaar'
) | alt.Chart(uwv_train_2012plus_T[uwv_train_2012plus_T['Kwartaalnummer'] ==␣
 ↪2]).mark_point().encode(
    x='Perioden',
    y='Ziekteverzuimpercentage_1',
    color='Jaar'
) | alt.Chart(uwv_train_2012plus_T[uwv_train_2012plus_T['Kwartaalnummer'] ==␣
 ↪3]).mark_point().encode(
    x='Perioden',
    y='Ziekteverzuimpercentage_1',
    color='Jaar'
) | alt.Chart(uwv_train_2012plus_T[uwv_train_2012plus_T['Kwartaalnummer'] ==␣
 ↪4]).mark_point().encode(
    x='Perioden',
    y='Ziekteverzuimpercentage_1',
    color='Jaar'
)
```

[189]:



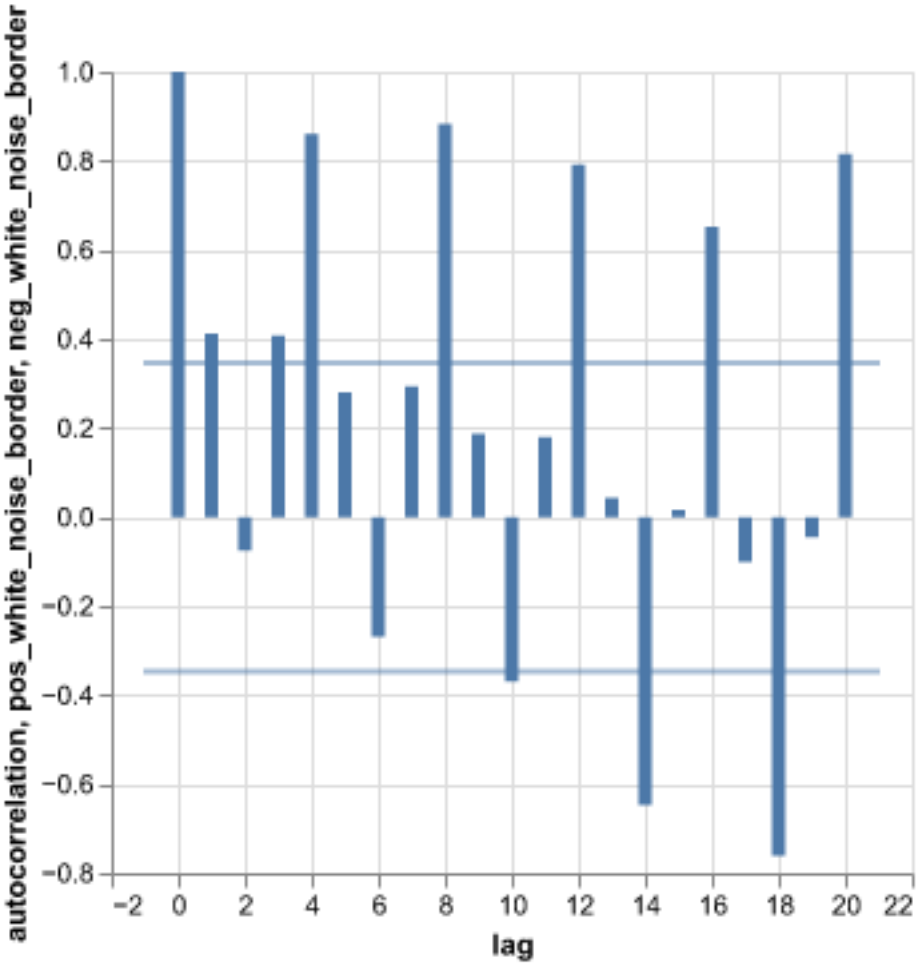
[190]:
```
from pandas.plotting import lag_plot
import matplotlib.pyplot as plt

fig, axes = plt.subplots(nrows=2, ncols=4, figsize=(20,10))

lag_plot(uwv_train_2012plus_T['Ziekteverzuimpercentage_1'], lag=1, ax=axes[0,␣
 ↪0])
```

```
lag_plot(uwv_train_2012plus_T['Ziekteverzuimpercentage_1'], lag=2, ax=axes[0,␣
 ↪1])
lag_plot(uwv_train_2012plus_T['Ziekteverzuimpercentage_1'], lag=3, ax=axes[0,␣
 ↪2])
lag_plot(uwv_train_2012plus_T['Ziekteverzuimpercentage_1'], lag=4, ax=axes[0,␣
 ↪3])
lag_plot(uwv_train_2012plus_T['Ziekteverzuimpercentage_1'], lag=5, ax=axes[1,␣
 ↪0])
lag_plot(uwv_train_2012plus_T['Ziekteverzuimpercentage_1'], lag=6, ax=axes[1,␣
 ↪1])
lag_plot(uwv_train_2012plus_T['Ziekteverzuimpercentage_1'], lag=7, ax=axes[1,␣
 ↪2])
lag_plot(uwv_train_2012plus_T['Ziekteverzuimpercentage_1'], lag=8, ax=axes[1,␣
 ↪3])

plt.show()
```



[191]:
```
import math

start_lag = 0
lag_length = 21

lagged_autocorrelation = pd.DataFrame()
lagged_autocorrelation['lag'] = range(start_lag, lag_length)

white_noise_border = 1.96 / math.
 ↪sqrt(len(uwv_train_2012plus_T['Ziekteverzuimpercentage_1']))
```

[191]:
```
wn_border = pd.DataFrame()
wn_border['lag'] = range(start_lag - 1, lag_length + 1)
wn_border['pos_white_noise_border'] = [white_noise_border for _ in␣
↪range(start_lag - 1, lag_length + 1)]
wn_border['neg_white_noise_border'] = [-white_noise_border for _ in␣
↪range(start_lag - 1, lag_length + 1)]

lagged_autocorrelation['autocorrelation'] =␣
↪[uwv_train_2012plus_T['Ziekteverzuimpercentage_1'].autocorr(lag=lag) for lag␣
↪in lagged_autocorrelation['lag']]

alt.Chart(lagged_autocorrelation).mark_bar().encode(
    x='lag',
    y='autocorrelation',
) + alt.Chart(wn_border).mark_line(strokeDash=[1,1]).encode(
    x='lag',
    y='pos_white_noise_border',
) + alt.Chart(wn_border).mark_line(strokeDash=[1,1]).encode(
    x='lag',
    y='neg_white_noise_border'
)
```



[192]:
```
moving_average = pd.DataFrame()

moving_average['quarter'] = uwv_train_2012plus_T['Perioden']
moving_average['sick'] = uwv_train_2012plus_T['Ziekteverzuimpercentage_1']

for window in range(3, 16, 2):
    moving_average[f'{window}-MA'] =␣
 ↪uwv_train_2012plus_T['Ziekteverzuimpercentage_1'].rolling(window,␣
 ↪center=True).mean()

moving_average
```

[192]:

	quarter	sick	3-MA	5-MA	7-MA	9-MA	11-MA	13-MA	\
85	2013KW01	4.5	NaN	NaN	NaN	NaN	NaN	NaN	
86	2013KW02	3.8	3.933333	NaN	NaN	NaN	NaN	NaN	
87	2013KW03	3.5	3.733333	3.94	NaN	NaN	NaN	NaN	
88	2013KW04	3.9	3.800000	3.78	3.842857	NaN	NaN	NaN	

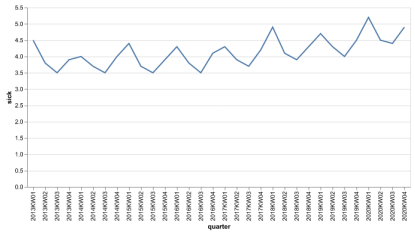
90	2014KW01	4.0	3.866667	3.72	3.771429	3.922222	NaN	NaN
91	2014KW02	3.7	3.733333	3.82	3.857143	3.833333	3.863636	NaN
92	2014KW03	3.5	3.733333	3.92	3.885714	3.800000	3.809091	3.900000
93	2014KW04	4.0	3.966667	3.86	3.828571	3.844444	3.854545	3.846154
95	2015KW01	4.4	4.033333	3.82	3.814286	3.888889	3.881818	3.823077
96	2015KW02	3.7	3.866667	3.90	3.900000	3.866667	3.845455	3.869231
97	2015KW03	3.5	3.700000	3.96	3.942857	3.844444	3.854545	3.900000
98	2015KW04	3.9	3.900000	3.84	3.871429	3.911111	3.909091	3.892308
100	2016KW01	4.3	4.000000	3.80	3.828571	3.944444	3.945455	3.892308
101	2016KW02	3.8	3.866667	3.92	3.914286	3.888889	3.918182	3.946154
102	2016KW03	3.5	3.800000	4.00	3.971429	3.888889	3.900000	4.015385
103	2016KW04	4.1	3.966667	3.92	3.942857	3.966667	4.009091	3.992308
105	2017KW01	4.3	4.100000	3.90	3.928571	4.077778	4.063636	4.007692
106	2017KW02	3.9	3.966667	4.04	4.085714	4.055556	4.063636	4.069231
107	2017KW03	3.7	3.933333	4.20	4.171429	4.066667	4.063636	4.130769
108	2017KW04	4.2	4.266667	4.16	4.142857	4.155556	4.145455	4.130769
110	2018KW01	4.9	4.400000	4.16	4.142857	4.222222	4.218182	4.146154
111	2018KW02	4.1	4.300000	4.28	4.257143	4.222222	4.209091	4.223077
112	2018KW03	3.9	4.100000	4.38	4.342857	4.233333	4.227273	4.307692
113	2018KW04	4.3	4.300000	4.26	4.314286	4.322222	4.345455	4.323077
115	2019KW01	4.7	4.433333	4.24	4.257143	4.433333	4.418182	4.361538
116	2019KW02	4.3	4.333333	4.36	4.414286	4.388889	4.436364	4.453846
117	2019KW03	4.0	4.266667	4.54	4.500000	4.422222	4.436364	NaN
118	2019KW04	4.5	4.566667	4.50	4.514286	4.533333	NaN	NaN
120	2020KW01	5.2	4.733333	4.52	4.542857	NaN	NaN	NaN
121	2020KW02	4.5	4.700000	4.70	NaN	NaN	NaN	NaN
122	2020KW03	4.4	4.600000	NaN	NaN	NaN	NaN	NaN
123	2020KW04	4.9	NaN	NaN	NaN	NaN	NaN	NaN

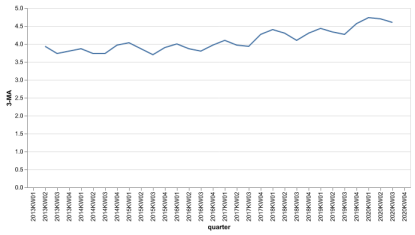
	15-MA
85	NaN
86	NaN
87	NaN
88	NaN
90	NaN
91	NaN
92	NaN
93	3.866667
95	3.840000
96	3.873333
97	3.900000
98	3.886667
100	3.900000
101	3.980000
102	4.020000
103	4.013333
105	4.006667

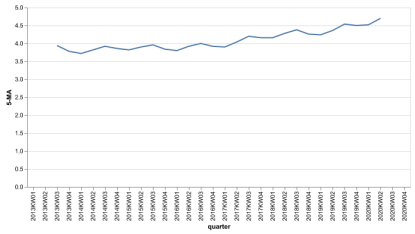
```
106  4.073333
107  4.126667
108  4.133333
110  4.146667
111  4.240000
112  4.306667
113  4.326667
115  4.366667
116       NaN
117       NaN
118       NaN
120       NaN
121       NaN
122       NaN
123       NaN
```

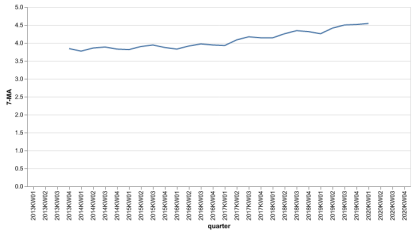
[193]:
```
charts = [alt.Chart(moving_average).mark_line().encode(x='quarter', y='sick')]

for window in range(3, 16, 2):
    charts.append(alt.Chart(moving_average).mark_line().encode(x='quarter',␣
 ↪y=f'{window}-MA'))

alt.vconcat(*charts)
```

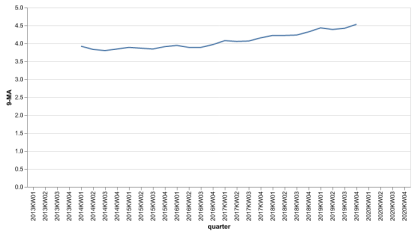
[193]:

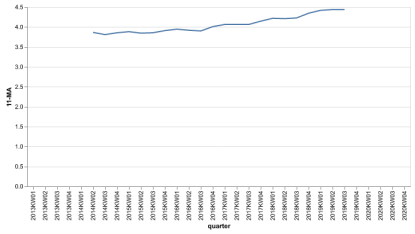

kick
quarter

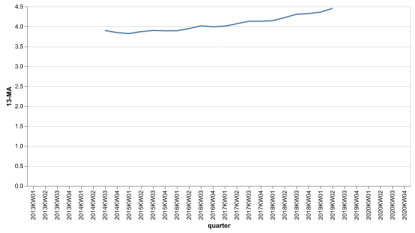

3-MA
quarter

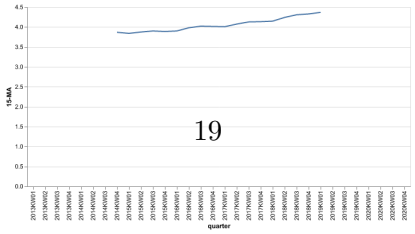

5-MA
quarter


7-MA
quarter


9-MA
quarter


11-MA
quarter


13-MA
quarter

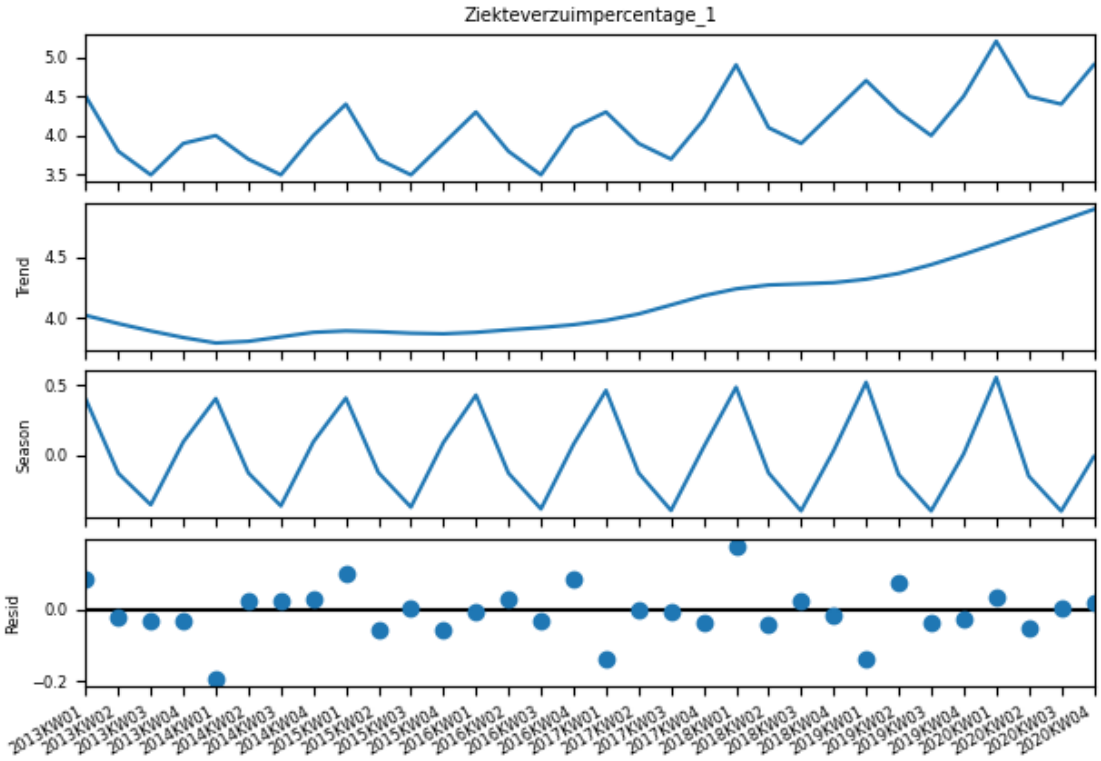

15-MA
quarter

[194]:
```
from statsmodels.tsa.seasonal import STL

slp = uwv_train_2012plus_T['Ziekteverzuimpercentage_1']
slp.index = uwv_train_2012plus_T['Perioden']
slp
```

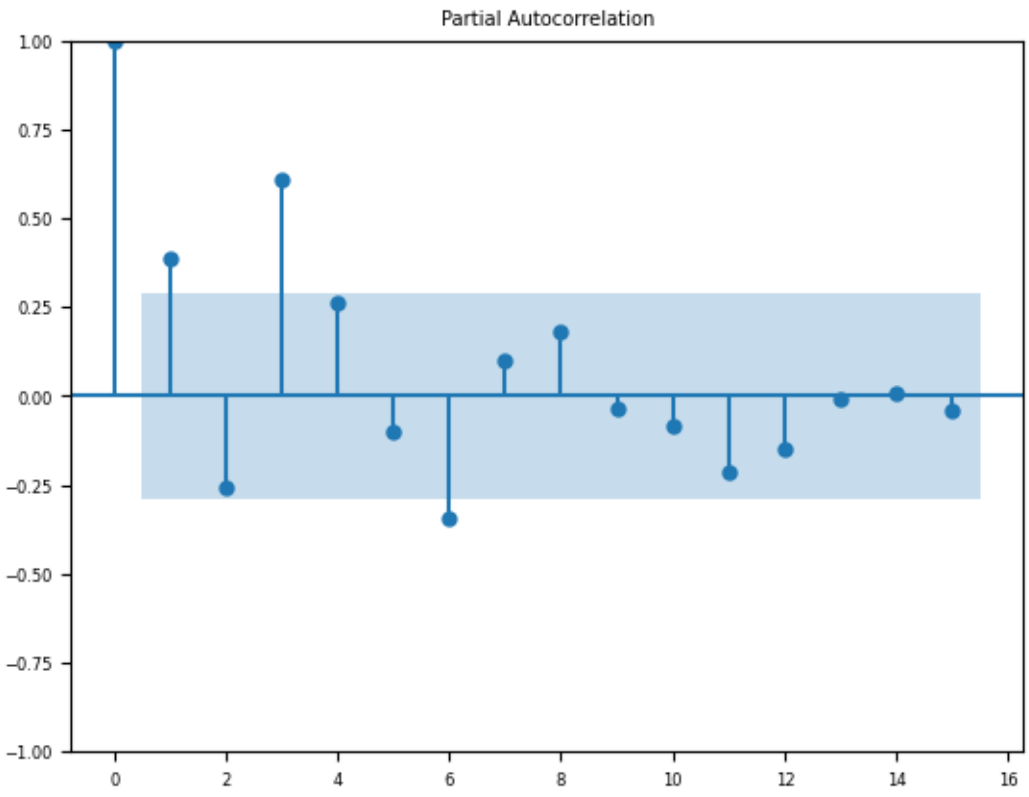
[194]:
```
Perioden
2013KW01    4.5
2013KW02    3.8
2013KW03    3.5
2013KW04    3.9
2014KW01    4.0
2014KW02    3.7
2014KW03    3.5
2014KW04    4.0
2015KW01    4.4
2015KW02    3.7
2015KW03    3.5
2015KW04    3.9
2016KW01    4.3
2016KW02    3.8
2016KW03    3.5
2016KW04    4.1
2017KW01    4.3
2017KW02    3.9
2017KW03    3.7
2017KW04    4.2
2018KW01    4.9
2018KW02    4.1
2018KW03    3.9
2018KW04    4.3
2019KW01    4.7
2019KW02    4.3
2019KW03    4.0
2019KW04    4.5
2020KW01    5.2
2020KW02    4.5
2020KW03    4.4
2020KW04    4.9
Name: Ziekteverzuimpercentage_1, dtype: float64
```

[195]:
```
plt.rc("font", size=6)
stl = STL(slp, period=4)
res = stl.fit()
```

```
fig = res.plot()
fig.autofmt_xdate()
```



[196]:
```
from statsmodels.graphics.tsaplots import plot_pacf

plot_pacf(slp, lags=15, alpha=0.1)
plt.show()
```


Partial Autocorrelation
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
2
4
6
8
10
12
14
16