Statens vegvesen

Jan Kristian Jensen

Python geopandas

Gis + tabulære data =

GitHub - LtGlahn/kostrarapportering2021

pandas

GeoPandas

GeoPandas

Timeline

- **2013**: Beginning of the development
- **2014**: GeoPandas 0.1.0 released
- **2020**: GeoPandas became NumFOCUS Affiliated Project

pandas

Timeline

- **2008**: Development of *pandas* started
- **2009**: *pandas* becomes open source
- **2012**: First edition of *Python for Data Analysis* is published
- **2015**: *pandas* becomes a NumFOCUS sponsored project

Detta ække nytt
– men det er nyttig!

python™

DataFrame is a 2-dimensional labeled data structure with columns of potentially different types.

You can think of it like a **spreadsheet or SQL table**

https://pandas.pydata.org/pandas-docs/stable/getting_started/dsintro.html

```
1 bomstasjoner[['Navn bomstasjon', 'Takst liten bil', 'Takst stor bil']].head()
```

	Navn bomstasjon	Takst liten bil	Takst stor bil
0	Øvstabø	64.0	84.0
1	Sandviken	32.0	72.0
2	Gyldenpris	32.0	72.0
3	Straume	32.0	72.0
4	Gravdal	32.0	72.0

«A GeoDataFrame is a DataFrame that has a column with geometry»

```
1 bomstasjoner[['Navn bomstasjon', 'Takst liten bil', 'Takst stor bil', 'geometry']].head()
```

	Navn bomstasjon	Takst liten bil	Takst stor bil	geometry
0	Øvstabø	64.0	84.0	POINT Z (8105.874 6552587.819 380.748)
1	Sandviken	32.0	72.0	POINT Z (-32178.727 6737228.442 31.701)
2	Gyldenpris	32.0	72.0	POINT Z (-32637.918 6733197.696 17.242)
3	Straume	32.0	72.0	POINT Z (-36045.964 6727021.841 5.105)
4	Gravdal	32.0	72.0	POINT Z (-35230.758 6733476.926 14.993)

Python shapely - objekter

Shapely

https://shapely.readthedocs.io/

2D GIS-operasjoner

Joda, geometri kan være 3D

- men alle operasjoner er i kartplan

```
1 bomstasjoner[['Navn bomstasjon', 'Takst liten bil', 'Takst stor bil', 'geometry']].head()
```

	Navn bomstasjon	Takst liten bil	Takst stor bil	geometry
0	Øvstabø	64.0	84.0	POINT Z (8105.874 6552587.819 380.748)
1	Sandviken	32.0	72.0	POINT Z (-32178.727 6737228.442 31.701)
2	Gyldenpris	32.0	72.0	POINT Z (-32637.918 6733197.696 17.242)
3	Straume	32.0	72.0	POINT Z (-36045.964 6727021.841 5.105)
4	Gravdal	32.0	72.0	POINT Z (-35230.758 6733476.926 14.993)

Python shapely - objekter

Min arbeidsflyt

1. Importer

Mitt eget bibliotek for NVDB
https://github.com/LtGlahn/nvdbapi-V3

https://pypi.org/project/nvdbapi-v3/

```
import nvdbapiv3
import pandas as pd
```

Pandas er med i nyere Python-installasjoner

Søkeobjekt – sjekk
https://github.com/LtGlahn/nvdbapi-V3

NVDB Objekttype 45 = Bomstasjon
https://datakatalogen.atlas.vegvesen.no/#/45-Bomstasjon

2. Søk (og filtrer)

```
mittSok = nvdbapiv3.nvdbFagdata(45)
mittSok.filter( {'fylke' : [11, 46] } )
mittSok.filter( { 'vegsystemreferanse' : 'Ev,Rv'})
```

Vil ha bomstasjoner innafor
Rogaland og Vestland fylke
På Europa- og riksveg

3. Gjør om til DataFrame

Lager liste med dictionaries

```
bomst = pd.DataFrame( mittSok.to_records() )
```

4. Utforsk data

```
bomst.head()
```

	objekttype	nvdbId	versjon	startdato	Rushtid morgen, til	Tidsdifferensiert takst	Rushtidstakst liten bil	Rushtidstaks stor bi
0	45	82443541	12	2022-01-01	08:59	Ja	59.0	130.(
1	45	82559833	11	2022-01-01	08:59	Ja	59.0	130.(
2	45	82559836	12	2022-01-01	08:59	Ja	59.0	130.(
3	45	141140381	11	2022-01-01	08:59	Ja	59.0	130.(
4	45	264392510	7	2021-11-01	NaN	Nei	NaN	NaN

5 rows × 47 columns

`bomst.dtypes`

```
objekttype                int64
nvdbId                    int64
Takst liten bil         float64
Takst stor bil          float64
Navn bomstasjon          object
kommune                   int64
fylke                     int64
vref                     object
vegkategori              object
fase                     object
vegnummer                 int64
geometri                 object

...
```

```
bomst['Takst liten bil'].describe()
```

```
count     38
mean      37.710526
std       32.707473
min       12.000000
25%       25.000000
50%       30.000000
75%       32.000000
max      149.000000
Name: Takst liten bil, dtype: float64
```

```
bomst.groupby( ['fylke', 'vegkategori'] ).agg( {

                                        'nvdbId' : 'count',
                                        'Takst liten bil' : 'mean’

} )
```

		nvdbId	Takst liten bil
fylke	vegkategori		
11	E	8	18.500000
	R	10	52.300000
46	E	17	33.666667
	R	5	51.400000

```
bomst.groupby( ['fylke', 'vegkategori'] ).agg( {

                                'nvdbId' : 'count',
                                'Takst liten bil' : 'mean'

                                                    } )
```

fylke	vegkategori	nvdbId	Takst liten bil
11	E	8	18.500000
	R	10	52.300000
46	E	17	33.666667
	R	5	51.400000

Teller opp per fylke og vegkategori

agg = aggregering
Hva teller vi?
Hvordan teller vi det?

```
resultat = bomst.groupby( ['fylke', 'vegkategori'] ).agg( {
                                                'nvdbId' : 'count',
                                                'Takst liten bil' : 'mean'
                                                    } ).reset_index()
```

	fylke	vegkategori	nvdbId	Takst liten bil
0	11	E	8	18.500000
1	11	R	10	52.300000
2	46	E	17	33.666667
3	46	R	5	51.400000

Ny DataFrame

5. Lag nye kolonner

```
Bomst['antall']  = 1
```

Lag ny kolonne med shapely geometri

Shapely

https://shapely.readthedocs.io/

Shapely

```
In [45]: bomst['geometri']
Out[45]:
0    POINT Z(-32178.727 6737228.442 31.701)
1    POINT Z(-35230.758 6733476.926 14.993)
2     POINT Z(-31572.63 6732758.286 10.152)
3    POINT Z(-31875.942 6729771.812 14.607)
4   POINT Z(-38302.253 6630101.496 58.359)
5    ....
```

Kolonne «geometri» inneholder tekst

Well Known Text

WKT Wikipedia

```
from shapely import wkt
Bomst['geometry']  = Bomst['geometri'].apply(wkt.loads  )
```

Bomst['vref'] =

Ev39 S79D50 m7106
Ev39 S79D1 m18734
Ev39 S79D1 m12371
Rv555 S1D1 m4051

Ønsker:

Ev39 S79
Fv5158 S1

```
Bomst['ny'] = Bomst['vref'].apply( lambda x : x.split('D')[0] )
```

lambda ???

Anonym funksjon med ett – 1 – enkelt uttrykk
kort levetid

```
Bomst['trafikantgruppe'] = K eller G
```

K EV39 S79
K RV555 S1
…

Ønsker:

Trafikantgruppe + *Ev39* S79
Trafikantgruppe + *Rv555* S1

```
Bomst['ny kolonne']  = Bomst['trafikantgruppe'] + " " +        \
                Bomst['vref'].apply( lambda x : x.split('D')[0] )
```

Bruk funksjon med hele raden som argument

For bilister EV39 S79
For bilister RV555 S1
…

```
def lagnykolonne( row):
    trafikantgruppe = 'For bilister'
    if row['trafikantgruppe'] == 'G':
        trafikantgruppe = 'For gående'
    return trafikantgruppe + " " + row['vref'].split( 'D' )[0]
```

```
Bomst['ny kolonne']  = Bomst.apply( lagnykolonne, axis=1 )
```

```
Bomst['ny kolonne']  = Bomst.apply( lambda x : lagnykolonne( x ), axis=1 )
```

Nytt karteksempel
– gøy med shapely geometri

Shapely

https://shapely.readthedocs.io/

Bergensere liker ikke bompenger!

Hypotese: Bergenseres bompengehat er omvendt proporsjonalt med avstanden til den blå steinen på Torgallmenningen

SRID=5973;Point(-32064, 6734257)

Dataprepp

for hypotese:

Bergenseres bompengehat er omvendt proporsjonalt med avstanden til den blå steinen på Torgallmenningen

blåstein = wkt.loads('Point(-32064 6734257)')

Shapely

https://shapely.readthedocs.io/

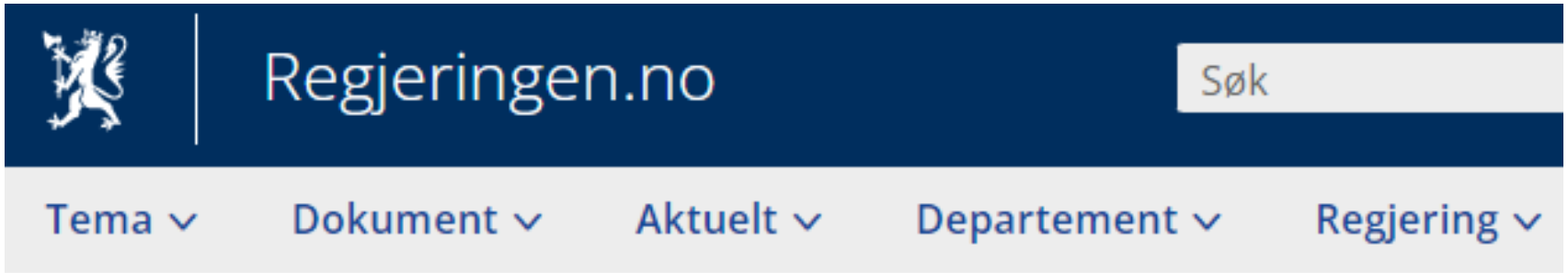
Bomst['Avstand blå stein'] = Bomst['geometry'].apply(lambda x : blåstein.distance(x))

6. Lagre

```
Bomst.to_excel('minExcelFil.xlsx', index=False)
```


```
import geopandas as gpd
from shapely import wkt
Bomst['geometry']  = Bomst['geometri'].apply( wkt.loads  )

Geobom = gpd.GeoDataFrame( Bomst, geometry='geometry', crs=5973)

Geobom.to_file('minfil.gpkg', layer='mittlag', driver='GPKG')
```

eller



Forslag til ny modell for beregning av kriteriet for fylkesveg i inntektssystemet for fylkeskommunene

Rapport | Dato: 02.07.2021

https://www.regjeringen.no/no/dokumenter/id2864850/

Faktor: Antall ferjekaibruer og tilleggskaier

1. Uttak av data fra NVDB (samme rapport som for bruer)

Uthenting av mengdegrunnlag – kilde:

NVDB rapporter(https://www.vegdata.no/produkter-og-tjenester/nvdb-rapporter/

Velg: Vegnett og objektdata for driftskontrakter

https://nvdb-vegnett-og-objektdata.atlas.vegvesen.no/generisk/

Velg: *Egendefinerte rapporter*

Velg: *Detaljert mengdeoversikt (V4)*

Velg objekttype: *«Bru»*

Velg fylke: *(alle fylker må hentes hver for seg eks Oslo)*

Velg Vegfilter: F (fylkesveg)

Lagre fil som regneark for hvert fylke

Legg inn to kolonner til venstre og legg inn fylkesnummer og fylkesnavn på hver linje.

Faktor: Antall ferjekaibruer og tilleggskaier

1. Uttak av data fra NVDB (samme rapport som for bruer)

Brukategori 1263

Vegbru 7304

Bru i fylling 7305

G/S-bru 7306

Ferjeleie 7307

Tunnel/Vegoverbygg 7309

Støttekonstruksjon 7310

Jernbanebru 7311

Annen byggv.kategori 7312

Objekttype 60 Bru i NVDB = mye mere enn vegbru!
https://datakatalogen.atlas.vegvesen.no/#/60-Bru

Faktor: Antall ferjekaibruer og tilleggskaier

1. Uttak av data fra NVDB (samme rapport som for bruer)

!!

Velg fylke: *(alle fylker må hentes hver for seg eks Oslo)*

Velg Vegfilter: F (fylkesveg)

Lagre fil som regneark for hvert fylke

Legg inn to kolonner til venstre og legg inn fylkesnummer og fylkesnavn på hver linje.

Filtrere samlet fil med følgende filtre for hvert fylke (med f.eks. bruk av pivottabell i Excel)

Trafikantgruppe:	K (kjørende)
Vegkategori:	F (fylkesveg)
Fase:	V (eksisterende)
Brukategori	Ferjeleie
Status:	Trafikkert, (blank)
Filtreringshjelp	1
Byggverkstype	810, 811, 812, 820, 822, 823, 824

Med bruk av pivot-tabell i Excel kan summering av brulengder for ulike materialtyper gjøres slik:

Trafikantgruppe	K
Vegkategori	F
Fase	V
Brukategori	Ferjeleie
Status	(Multiple Items)
Filteringshjelp	1

Count of NummerOgNavn	Byggverkstype								
Fylke	Ferjekaibru (810)	Ferjekaibru (811)	Ferjekaibru (812)	Ferjekaibru (819)	Kai (820)	Tilleggskai (822)	Tilleggskai (823)	Tilleggskai (824)	Grand Total
Agder		4			1				5
Innlandet	2								2
Møre og Romsdal	47		4						51
Nordland	14	17	23	1		6	5	4	70
Rogaland	16	3	1	1		1			22
Troms og Finnmark	12	3	27			1			43
Trøndelag	20	2	1						23
Vestland	1	17	21	19					58
Viken	1	1	2			1			5
Grand Total	**113**	**47**	**79**	**21**	**1**	**9**	**5**	**4**	**279**

Regjeringen.no

Tema Dokument Aktuelt Departemen

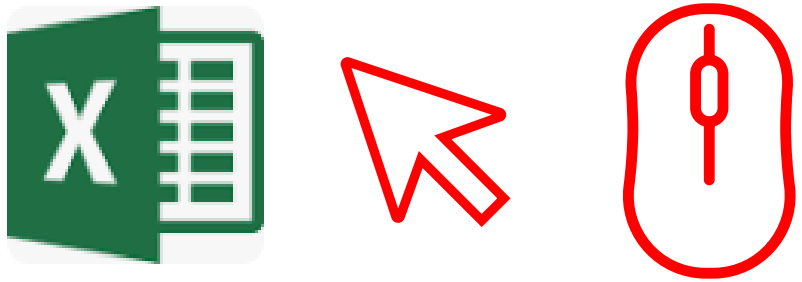
Du er her: Forsiden • Dokument • Rapporter og planer •

Forslag til ny modell for beregning av kriteriet for fylkesveg i inntektssysteme

Forslag til ny modell for beregning av kriteriet for fylkesveg i inntektssystemet for fylkeskommunene

Rapport | Dato: 02.07.2021

Last ned per fylke
=> *11 x last ned*
Sammenstill manuelt
Filtrer
Lag Pivot-tabell



Faktor: Antal

1. Uttak av da

Uthenting av

NVDB rapport

Velg: Vegnett og objektdata for driftskontrak

https://nvdb-vegnett-og-objektdata.atlas.vegvese

Velg: *Egendefinerte rapporter*

Brukategori = Ferjeleie

https://datakatalogen.atlas.vegvesen.no/#/60-Bru

```
sok = nvdbapiv3.nvdbFagdata(60)
sok.filter( {'vegsystemreferanse' : 'Fv' } )
sok.filter( { 'egenskap' : '(1263=7307)'} )
mydf = pd.DataFrame( sok.to_records())
mydf.drop_duplicates( subset='nvdbId')
```

Filtrere samlet fil med følgende filtre for hvert fylke (med f.eks. bruk av pivottabell i Excel)

Trafikantgruppe:	K (kjørende)
Vegkategori:	F (fylkesveg)
Fase:	V (eksisterende)
Brukategori	Ferjeleie
Status:	Trafikkert, (blank)
Filtreringshjelp	1
Byggverkstype	810, 811, 812, 820, 822, 823, 824

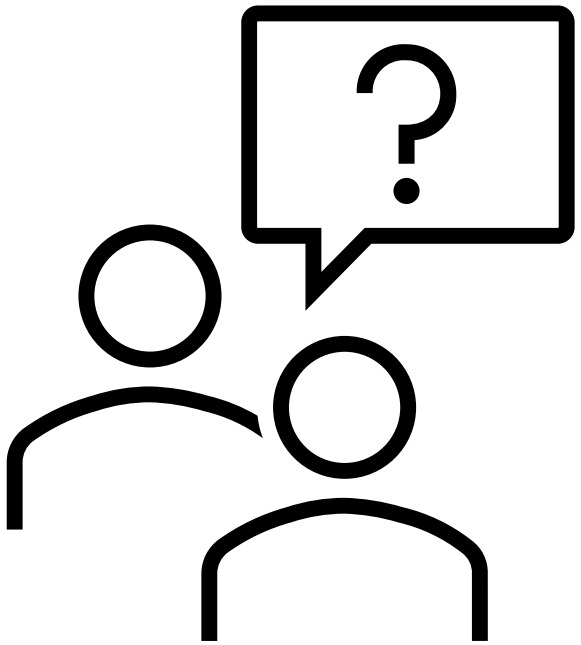
```
sok = nvdbapiv3.nvdbFagdata(60)
sok.filter( {'vegsystemreferanse' : 'Fv' } )
sok.filter( { 'egenskap' : '(1263=7307)'} )
```

Filtrere samlet fil med følgende filtre for hvert fylke (med f.eks. bruk av pivottabell i Excel)

Byggverkstype	810, 811, 812, 820, 822, 823, 824

```
bvType = ['Ferjekaibru (810)',
          'Ferjekaibru (811)', 'Ferjekaibru (812)',
          'Kai (820)', 'Tilleggskai (822)',
          'Tilleggskai (823)', 'Tilleggskai (824)' ]

 mydf = mydf[ mydf['Byggverkstype'].isin(bvType)].copy()
```

```
mydf[ mydf['Byggverkstype'].isin(bvType) ]
```



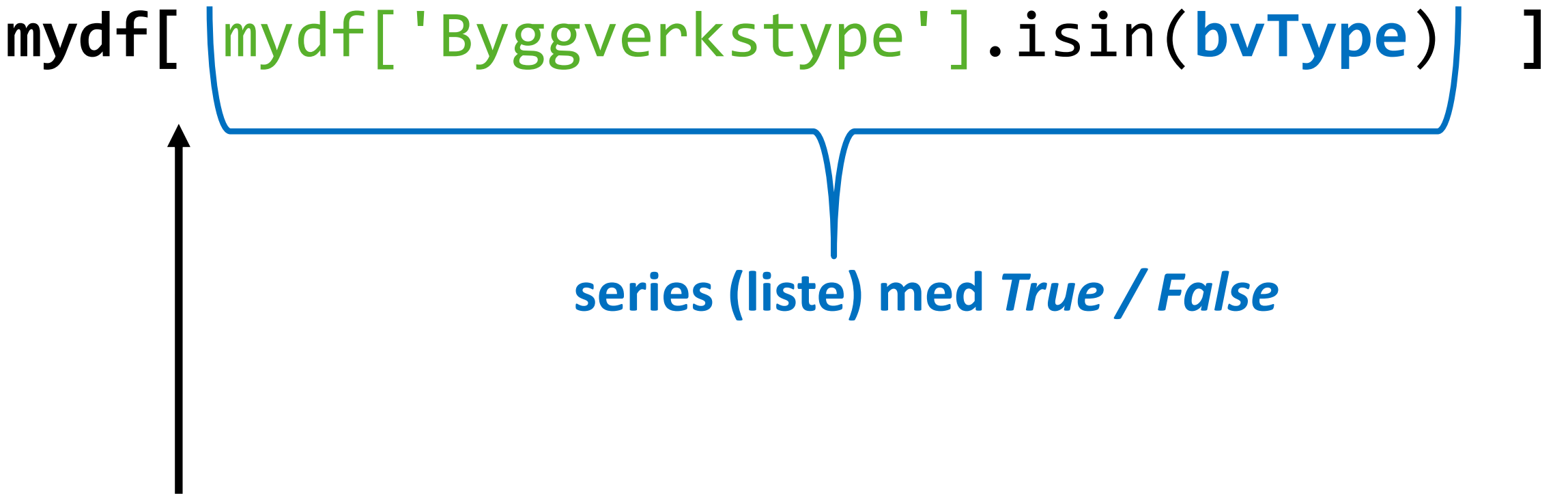
mydf[*en eller annen betingelse* **]**

DataFrame med hakeparantes =
gi meg subsett av DataFrame

en eller annen betingelse

Betingelse = *Generer **«liste»** med True / False*

Liste-aktig…
Pandas series



Returnerer ny Dataframe med de elementene der vi har*True* inni «*listene*» i hakeparantes

Filtrere samlet fil med følgende filtre for hvert fylke (med f.eks. bruk av pivottabell i Excel)

Byggverkstype	810, 811, 812, 820, 822, 823, 824

```
bvType = ['Ferjekaibru (810)',
          'Ferjekaibru (811)', 'Ferjekaibru (812)',
          'Kai (820)', 'Tilleggskai (822)',
          'Tilleggskai (823)', 'Tilleggskai (824)' ]
mydf = mydf[ mydf['Byggverkstype'].isin(bvType)].copy()
```

```
dfB = mydfA[ betingelse ]    .copy()
```

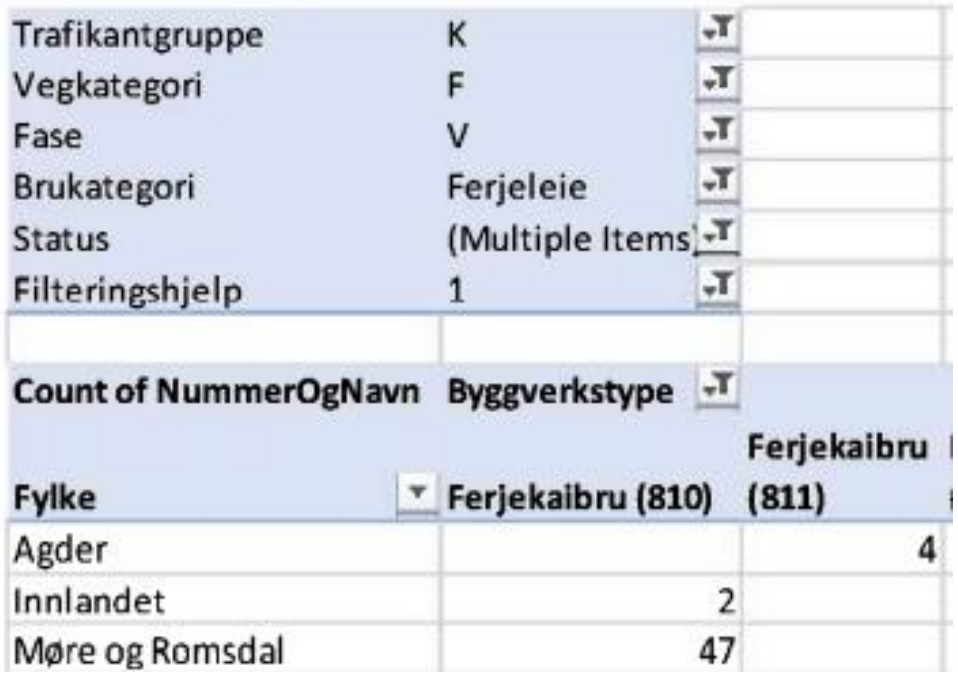

`dfB` ikke egen kopi, men referer til mydfA
Kan ikke endre `dfB`

Overstyres med .copy() => endrbar `dfB`

Med bruk av pivot-tabell i Excel kan summering av brulengder for ulike materialtyper gjøres slik:

Trafikantgruppe	K
Vegkategori	F
Fase	V
Brukategori	Ferjeleie
Status	(Multiple Items)
Filteringshjelp	1

Count of NummerOgNavn	Byggverkstype								
Fylke	Ferjekaibru (810)	Ferjekaibru (811)	Ferjekaibru (812)	Ferjekaibru (819)	Kai (820)	Tilleggskai (822)	Tilleggskai (823)	Tilleggskai (824)	Grand Total
Agder		4			1				5
Innlandet	2								2
Møre og Romsdal	47		4						51
Nordland	14	17	23	1		6	5	4	70
Rogaland	16	3	1	1		1			22
Troms og Finnmark	12	3	27			1			43
Trøndelag	20	2	1						23
Vestland	1	17	21	19					58
Viken	1	1	2			1			5
Grand Total	**113**	**47**	**79**	**21**	**1**	**9**	**5**	**4**	**279**



Trafikantgruppe	K
Vegkategori	F
Fase	V
Brukategori	Ferjeleie
Status	(Multiple Items)
Filteringshjelp	1

Count of NummerOgNavn	Byggverkstype	
Fylke	Ferjekaibru (810)	Ferjekaibru (811)
Agder		4
Innlandet	2	
Møre og Romsdal	47	

```
mydf['antall'] = 1
```

```
byggverkstype = mydf[['fylke', 'Byggverkstype', 'antall']].pivot_table(

                    index='fylke',
                    columns='Byggverkstype',
                    aggfunc='count',
                    fill_value=0  )
```

```
In [19]: byggverkstype
Out[19]:
                        antall
Byggverkstype Ferjekaibru (810) Ferjekaibru (811) Ferjekaibru (812) Kai (820) Tilleggskai (822) Tilleggskai (823) Tilleggskai (824)
fylke
11                           17                3                 2        0                 1                 0                 0
15                           52                0                 4        0                 0                 0                 0
18                           15               17                25        2                 6                 5                 4
30                            1                1                 2        0                 1                 0                 0
34                            5                0                 0        0                 0                 0                 0
42                            0                4                 0        1                 0                 0                 0
46                            3               18                22        0                 0                 0                 0
50                           21                2                 1        0                 0                 0                 0
54                           22                3                27        0                 1                 0                 0
```

```
In [19]: byggverkstype
Out[19]:
                        antall
Byggverkstype Ferjekaibru (810) Ferjekaibru (811) Ferjekaibru (812) Kai (820) Tilleggskai (822) Tilleggskai (823) Tilleggskai (824)
fylke
11                           17                3                 2         0                 1                 0                 0
15                           52                0                 4         0                 0                 0                 0
18                           15               17                25         2                 6                 5                 4
30                            1                1                 2         0                 1                 0                 0
34                            5                0                 0         0                 0                 0                 0
42                            0                4                 0         1                 0                 0                 0
46                            3               18                22         0                 0                 0                 0
50                           21                2                 1         0                 0                 0                 0
54                           22                3                27         0                 1                 0                 0
```

```
# Litt dataframe-wodoo for å få vekk multilevel-kolonnenavn
#  https://stackoverflow.com/a/44023799
byggverkstype.columns = byggverkstype.columns.droplevel(0)
```

```
Byggverkstype.columns =
                    MultiIndex([('antall', 'Ferjekaibru (810)'),
                                ('antall', 'Ferjekaibru (811)'),
                                ('antall', 'Ferjekaibru (812)'),
                                ('antall',       'Kai (820)'),
                                ('antall', 'Tilleggskai (822)'),
                                ('antall', 'Tilleggskai (823)'),
                               ('antall', 'Tilleggskai (824)')],
                              names=[None, 'Byggverkstype'])
```

Genialt, men forvirrende!

```
                          antall
Byggverkstype Ferjekaibru (810) Ferjekaibru (811) Ferjekaibru (812) Ka
fylke
11                            17                 3                  2
```

```
# Litt dataframe-wodoo for å få vekk multilevel-kolonnenavn
#  https://stackoverflow.com/a/44023799
byggverkstype.columns = byggverkstype.columns.droplevel(0)
```

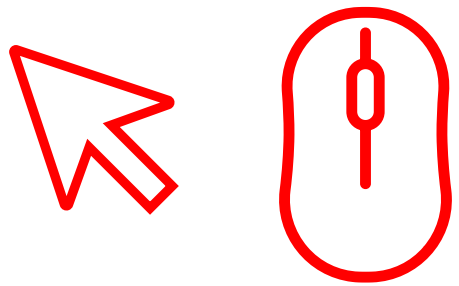
Byggverkstype fylke	Ferjekaibru (810)	Ferjekaibru (811)	Ferjekaibru (812)	Kai (820)	T
11	17	3	2	0	
15	52	0	4	0	
18	15	17	25	2	
30	1	1	2	0	
34	5	0	0	0	
42	0	4	0	1	
46	3	18	22	0	
50	21	2	1	0	
54	22		27	0	

Flat tabell

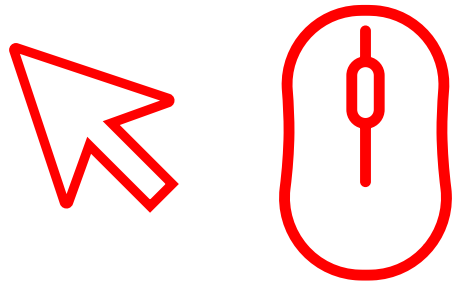
```
import pandas as pd
import nvdbapiv3
if __name__ == '__main__':

    sok = nvdbapiv3.nvdbFagdata(60)
    sok.filter( {'vegsystemreferanse' : 'Fv' } )
    sok.filter( { 'egenskap' : '(1263=7307)'} )      # Brukategori=Ferjeleie
    mydf = pd.DataFrame( sok.to_records( ))
    mydf.drop_duplicates( subset='nvdbId', inplace=True)

    # Disse brutypene er definert i oppskriften: 810, 811, 812, 820, 822, 823, 824.
    # Merk at vi hopper over 821.
    disseByggverkTypene = ['Ferjekaibru (810)', 'Ferjekaibru (811)', 'Ferjekaibru (812)',
                'Kai (820)', 'Tilleggskai (822)',  'Tilleggskai (823)', 'Tilleggskai (824)' ]
    mydf = mydf[ mydf['Byggverkstype'].isin( disseByggverkTypene )].copy()
    mydf['antall'] = 1
    byggverkstype = mydf[['fylke', 'Byggverkstype', 'antall']].pivot_table(
                index='fylke', columns='Byggverkstype', aggfunc='count', fill_value=0 )

    # Litt dataframe-wodoo for å få vekk multilevel-kolonnenavn
    byggverkstype.columns = byggverkstype.columns.droplevel(0)
```

https://github.com/LtGlahn/kostrarapportering2021/blob/adhoc-statsbudsjettH2022/kode/script_ferjekaibruer.py

40 linjer kode



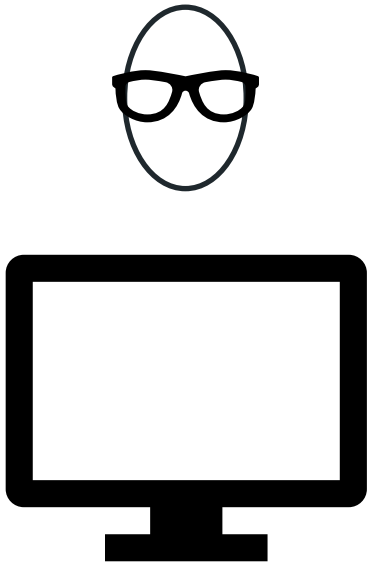
pandas



40 linjer kode

Repeterbart
Dokumenterbart
Etterprøvbart

Lær av data science folka!

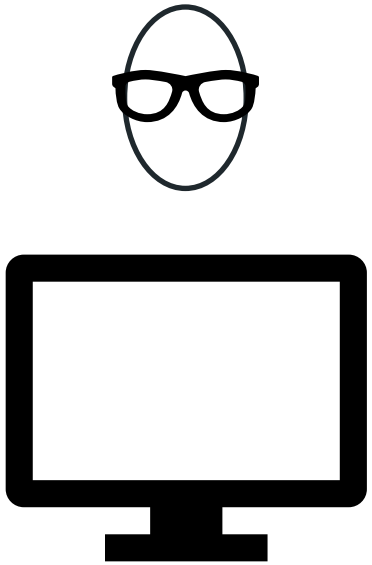


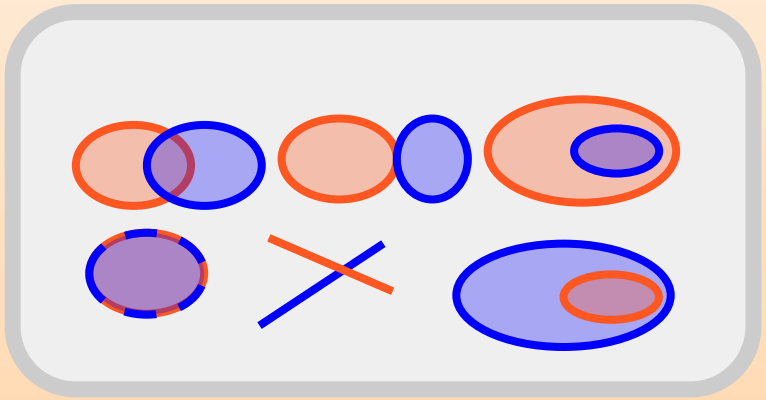
pandas

GeoPandas

Ny teknologi
Nye analyseteknikker
STORDATA teknologi
Metodikk & **beste praksis**

Spatial is no longer special





GIS metodikk
GIS analyser
GIS verktøy

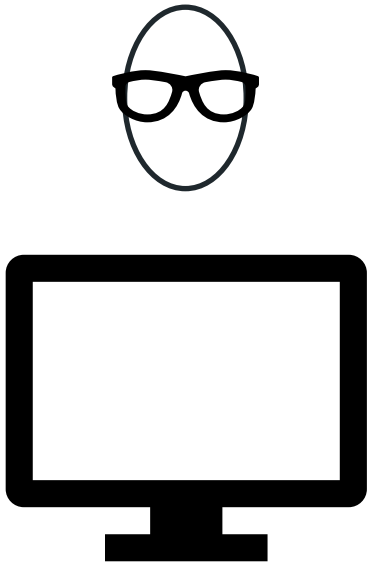
Spatial is **no longer** *special*

For a data scientists:

A map … is just another visualization

A spatial analysis … is just another analysis

Lær av data science folka!



pandas

GeoPandas

Ny teknologi
Nye analyseteknikker
STORDATA teknologi
Metodikk & **beste praksis**