Teknisk dokumentation af Udbyttedata

Der er produceret et kvalitetssikret datasæt for vinterhvede og et kvalitetssikret datasæt for majshelsæd. Der er indhentet udbyttedata fra 2011 til 2017, hvilket gav i alt 1083 udbyttekort. Udbyttedata har gennemgået oprensning og kvalitetssikring inden anvendelse i analyser. Efter kvalitetssikring var der 205 udbyttekort for vinterhvede og 133 udbyttekort for majshelsæd.
Yderligere er der for udbyttedata fra 2016 og 2017 (rasterdata) samlet følgende sammen med de interpolerede udbytter, NDVI, NDRE, MSAVI2, og 13 satellitbånd fra Sentinel 2, højde over hav-overfalde (DTM_mean), koordinater (UTM32) og markpolygoner. Alle opløsningerne ligger med en grid-størrelse på 5 x 5 meter.

Kvalitetssikring og oprensning af udbyttedata

Kvalitetssikring:

- Data udenfor markpolygon er fjernet.
- Udbyttet er standardiseret (klar definition af overskrifter i tabeller, enhed og kolonner som er interessante)
- Tjek af afgrødetype på markerne

Oprensning:

- Sort yield measurements by timestamp.
- Min/Max threshold: Remove yield measurements lower and higher than a user-defined dry_yield min_value and max_value in kg/ha per crop type. (We use the values VINTERHVEDE_MIN = 1000, VINTERHVEDE_MAX = 25000, MAJSHELSAED_MIN = 1000, and MAJSHELSAED_MAX = 30000)
- Remove statistical outliers: Filter out distance based errors by outlier analysis using the distance-to-yield ratio. This is performed because, sensor errors give unrealistic yields when distance travelled is minimal. Plotting the histogram of -log(distance/yield) shows the clear peaks in yield measurements, which we want to filter away. Thus, we filter away yields where the distance-to-yield is higher than mean of distance-to-yield plus one standard deviation.
- Remove incomplete maps: Remove yield map from dataset if convex hull area of merged, trimmed yield maps covers at least 75% of total field area AND point-circle area approximation covers at least 75 % of the total field area.

Udbytte data Majshelsæd

Majsudbyttedata			
	Udbytte_ma	**Dry_matter**	**Fugtindhold**
Enheder	Kg/ha	%	%
Beskrivelse	Grønmasse	Tørstofprocenten	Vandprocent
Typisk udbytteniveau	40-50 t/ha	ligger på omkring 30 %	ligger på omkring 70 %
Kommentar	Hvis det havde været udbyttet i tørstof, havde det lagt på 10-15 t/ha	30-52 % i dataeksempel. Gennemsnitlig for marken skal det nok ligge på omkring 30 %.	47,8-70 % i dataeksempel. Denne parameter er mindre vigtig når TS % haves.

Vinterhvede

Normudbytter			
	Vinterhvede	Vårbyg	vårhvede
Enheder	t/ha	t/ha	t/ha
Typisk udbytteniveau	9-11 t/ha med 15 % vand	4-7 t/ha med	5-7 t/ha

Report of yield map data set pipeline¶

In [1]:

```
from datetime import datetime
import os
import glob

from IPython.display import display
import dask
import dask.distributed
import dask.bag as db
import numpy as np
import pandas as pd
import geopandas as gpd
import pyarrow as pa
import pyarrow.parquet as pq
import matplotlib.pyplot as plt
plt.ion()
plt.style.use('ggplot')

# Global variables: Input
PROJECT_DATA_PATH = '/projects/BDICG/'
YIELD_MAP_DATA_PATH = PROJECT_DATA_PATH + 'pipeline_output_backup/yield_map_2011_2017_pip
eline_output_2018_09_24_13_49_18/'
```

In [2]:

```
client = dask.distributed.Client('localhost:8786')
client
```

```
distributed.comm.tcp - WARNING - Could not set timeout on TCP stream: [Errno 92] Protocol
not available
distributed.comm.tcp - WARNING - Could not set timeout on TCP stream: [Errno 92] Protocol
not available
```

Out[2]:

Client

Cluster

- **Scheduler:** tcp://localhost:8786
- **Dashboard:** http://localhost:8787/status
- **Workers:** 20
- **Cores:** 20
- **Memory:** 8006.38 GB

BDICG-88: Final cleaned yield map output¶

- Determine the number of yield maps per year for vinterhvede and majshelsæd
- Determine the number of yield maps from the same field for 2011 and 2017.

Loading files¶

In [3]:

```
# Note that we have shorten vinterhvede to VH and majshelsaed to MH

gdf_cleaned_VH_fields_overview = dask.delayed(gpd.read_file)(YIELD_MAP_DATA_PATH + 'yield
_maps/vinterhvede/cleaned_vinterhvede_fields_overview.geojson').compute()

gdf_cleaned_MH_fields_overview = dask.delayed(gpd.read_file)(YIELD_MAP_DATA_PATH + 'yield
_maps/majshelsaed/cleaned_majshelsaed_fields_overview.geojson').compute()
```

In [5]:

```
gdf_cleaned_VH_fields_overview['geometry_WKB'] = gdf_cleaned_VH_fields_overview['geometry
'].apply(lambda x: x.wkb)

gdf_cleaned_MH_fields_overview['geometry_WKB'] = gdf_cleaned_MH_fields_overview['geometry
'].apply(lambda x: x.wkb)
```

Determine the number of yield maps per year for vinterhvede and majshelsæd¶

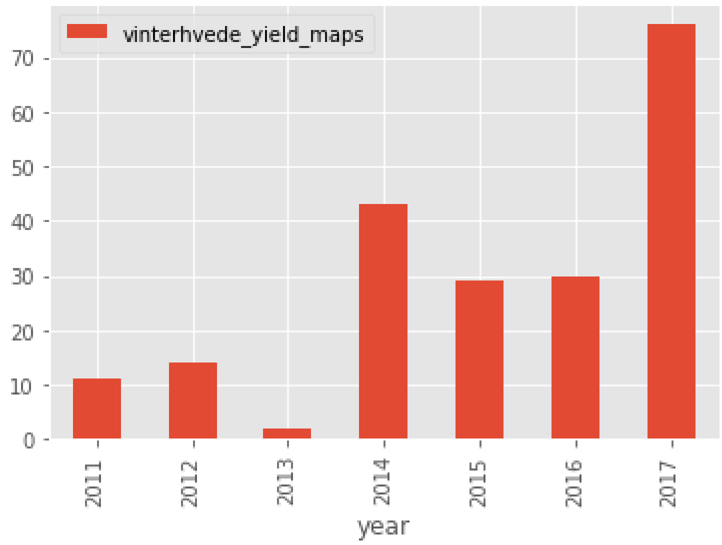
In [6]:

```
# Vinterhvede VH

ymap_VH_count = gdf_cleaned_VH_fields_overview[['id_DDS', 'year']].groupby('year').count(
)

ymap_VH_count = ymap_VH_count.rename(columns={'id_DDS':'vinterhvede_yield_maps'})

ymap_VH_count = ymap_VH_count.append(ymap_VH_count.sum().rename('Total'))

display(ymap_VH_count)

ymap_VH_count.iloc[0:-1].plot(kind='bar')
```

	vinterhvede_yield_maps
year	
2011	11
2012	14
2013	2
2014	43
2015	29
2016	30
2017	76
Total	205

Out[6]:

<matplotlib.axes._subplots.AxesSubplot at 0x7fca67fd63c8>



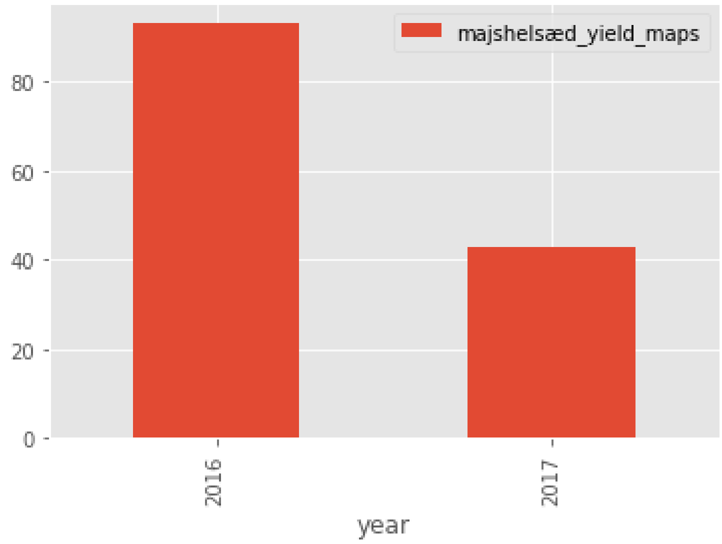
In [7]:

```
# Majshelsæd MH
ymap_MH_count = gdf_cleaned_MH_fields_overview[['id_DDS', 'year']].groupby('year').count(
)
ymap_MH_count = ymap_MH_count.rename(columns={'id_DDS':'majshelsæd_yield_maps'})
ymap_MH_count = ymap_MH_count.append(ymap_MH_count.sum().rename('Total'))
display(ymap_MH_count)
ymap_MH_count.iloc[0:-1].plot(kind='bar')
```

	majshelsæd_yield_maps
year	
2016	93
2017	43
Total	136

Out[7]:

<matplotlib.axes._subplots.AxesSubplot at 0x7fca64328a58>



BDICG-88: Determine the number of yield maps from the same field over all years¶

We start with a narive aproche with not considuration that IMK polygon can change at all between years!

Conclusion:

- Vinterhvede: We only have one field with yield maps from 3 years, and we have four fields with yield maps from 2 years.
- Majshelsæd: We only have seven fields with yield maps from 2 years.

In [8]:

```
ymap_VH_year_count = gdf_cleaned_VH_fields_overview[['id_DDS', 'geometry_WKB']].groupby('
geometry_WKB').count()

ymap_VH_year_count = ymap_VH_year_count.reset_index(drop=True).rename(columns={'id_DDS':'
vinterhvede_year_maps'})

ymap_VH_year_count = ymap_VH_year_count.reset_index().groupby('vinterhvede_year_maps').co
unt().rename(columns={'index': 'number_of_fields'}).T

ymap_VH_year_count
```

Out[8]:

vinterhvede_year_maps	1	2	3
number_of_fields	194	4	1

In [9]:

```
ymap_MH_year_count = gdf_cleaned_MH_fields_overview[['id_DDS', 'geometry_WKB']].groupby('
geometry_WKB').count()

ymap_MH_year_count = ymap_MH_year_count.reset_index(drop=True).rename(columns={'id_DDS':'
majshelsæd_yield_maps'})

ymap_MH_year_count = ymap_MH_year_count.reset_index().groupby('majshelsæd_yield_maps').co
unt().rename(columns={'index': 'number_of_fields'}).T

ymap_MH_year_count
```

Out[9]:

majshelsæd_yield_maps	1	2
number_of_fields	122	7

Improvement: Find yield maps where the IMK polygon, for the fields with vinterhvede or majshelsæd, overlaps between years. Such overlap will show that a IMK polygon have changes over years for the same field.

Conclusion: There are no vinterhvede or majshelsæd IMK polygons which overlaps between years. Thus, the year count presented above is the final amount.

In [10]:

```
def display_intersections_between_years(gdf):
    for year in gdf.year.unique():
        gdf_year = gdf[gdf.year == year]
        for delta_year in range(year + 1, gdf.year.max() + 1):
            gdf_delta_year = gdf[gdf.year == delta_year]


            # Display field intersections between years
            gdf_display = gdf_year[gdf_year.intersects(gdf_delta_year)]
            print('Field intersections between years x: {} and y {} contains: {} rows.'.f
ormat(year, delta_year, gdf_display.shape[0]))
```

In [11]:

```
display_intersections_between_years(gdf_cleaned_VH_fields_overview)
```

```
Field intersections between years x: 2011 and y 2012 contains: 0 rows.
Field intersections between years x: 2011 and y 2013 contains: 0 rows.
Field intersections between years x: 2011 and y 2014 contains: 0 rows.
Field intersections between years x: 2011 and y 2015 contains: 0 rows.
Field intersections between years x: 2011 and y 2016 contains: 0 rows.
Field intersections between years x: 2011 and y 2017 contains: 0 rows.
Field intersections between years x: 2012 and y 2013 contains: 0 rows.
Field intersections between years x: 2012 and y 2014 contains: 0 rows.
Field intersections between years x: 2012 and y 2015 contains: 0 rows.
Field intersections between years x: 2012 and y 2016 contains: 0 rows.
Field intersections between years x: 2012 and y 2017 contains: 0 rows.
Field intersections between years x: 2013 and y 2014 contains: 0 rows.
Field intersections between years x: 2013 and y 2015 contains: 0 rows.
Field intersections between years x: 2013 and y 2016 contains: 0 rows.
Field intersections between years x: 2013 and y 2017 contains: 0 rows.
Field intersections between years x: 2014 and y 2015 contains: 0 rows.
Field intersections between years x: 2014 and y 2016 contains: 0 rows.
Field intersections between years x: 2014 and y 2017 contains: 0 rows.
Field intersections between years x: 2015 and y 2016 contains: 0 rows.
Field intersections between years x: 2015 and y 2017 contains: 0 rows.
Field intersections between years x: 2016 and y 2017 contains: 0 rows.
/home/ubuntu_FOGH/anaconda3/envs/py36_otrta/lib/python3.6/site-packages/geopandas/geodata
frame.py:455: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  result = super(GeoDataFrame, self).__getitem__(key)
```

In [12]:

```
display_intersections_between_years(gdf_cleaned_MH_fields_overview)
```

```
Field intersections between years x: 2016 and y 2017 contains: 0 rows.
/home/ubuntu_FOGH/anaconda3/envs/py36_otrta/lib/python3.6/site-packages/geopandas/geodata
frame.py:455: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
  result = super(GeoDataFrame, self).__getitem__(key)
```

Markpolygoner

Majshelsæd bliver i markpolygonerne fra 2016 og 2017 skrevet som ”Silomajs” mens vinterhvedemarker kan hedde ”Vinterhvede” og ”Vinterhvede, brødhvede” (se tabel 1). Udbyttedataene skal klippes med markpolygonerne for det enkelte år - hvor mange marker er der med Silomajs og hvor mange der er enten med Vinterhvede eller vinterhvede, brødhvede.

Tabel 1. vigtige afgrøder i markpolygonerne fra 2016 og 2017.

Markpolygoner 2017	Afgrødedefinition
Silomajs	Majshelsæd
Vinterhvede	Vinterhvede
Vinterhvede, brødhvede	Vinterhvede