```
# import python packages
import pandas as pd
print("import package libraries")
```

```
import package libraries
```

```
# load dataset
tree_census = pd.read_csv('/trees.csv')
print("load dataset may take long to load")
```

```
load dataset may take long to load
```

```
# look at the first five rows
tree_census.head()
```

	created_at	tree_id	block_id	the_geom	tree_dbh	stump_dia
0	08/27/2015	180,683	348,711	POINT (-73.84421521958048 40.723091773924274)	3	
1	09/03/2015	200,540	315,986	POINT (-73.81867945834878 40.79411066708779)	21	
2	09/05/2015	204,026	218,365	POINT (-73.93660770459083 40.717580740099116)	3	
3	09/05/2015	204,337	217,969	POINT (-73.93445615919741 40.713537494833226)	10	
4	08/30/2015	189,565	223,043	POINT (-73.97597938483258 40.66677775537875)	21	

5 rows × 42 columns

```
# look at the last five rows
tree_census.tail()
```

	created_at	tree_id	block_id	the_geom	tree_dbh	stump
683783	08/18/2015	155,433	217,978	POINT (-73.95494401022562 40.7132107823145)	25	
683784	08/29/2015	183,795	348,185	POINT (-73.85665019989099 40.71519444267162)	7	
683785	08/22/2015	166,161	401,670	POINT (-74.13651724205825 40.62076152739799)	12	
683786	08/29/2015	184,028	504,204	POINT (-73.90311472453581 40.850828186655754)	9	
683787	09/03/2015	200,607	306,527	POINT (-73.78752645502483 40.73216525220126)	23	

5 rows × 42 columns

```
# list of column names
tree_census.columns
```

```
Index(['created_at', 'tree_id', 'block_id', 'the_geom', 'tree_dbh',
       'stump_diam', 'curb_loc', 'status', 'health', 'spc_latin',
'spc_common',
       'steward', 'guards', 'sidewalk', 'user_type', 'problems',
'root_stone',
       'root_grate', 'root_other', 'trnk_wire', 'trnk_light',
'trnk_other',
       'brnch_ligh', 'brnch_shoe', 'brnch_othe', 'address', 'zipcode',
       'zip_city', 'cb_num', 'borocode', 'boroname', 'cncldist',
'st_assem',
       'st_senate', 'nta', 'nta_name', 'boro_ct', 'state', 'Latitude',
       'longitude', 'x_sp', 'y_sp'],
      dtype='object')
```

```
# identify the size, number of rows and columns in the dataset
tree_census.shape
```

```
(683788, 42)
```

```
# summary of the dataset
tree_census.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 683788 entries, 0 to 683787
Data columns (total 42 columns):
 #   Column      Non-Null Count   Dtype 
---  ------      --------------   ----- 
 0   created_at  683788 non-null  object
 1   tree_id     683788 non-null  object
 2   block_id    683788 non-null  object
 3   the_geom    683788 non-null  object
 4   tree_dbh    683788 non-null  int64 
 5   stump_diam  683788 non-null  int64 
 6   curb_loc    683788 non-null  object
 7   status      683788 non-null  object
 8   health      652172 non-null  object
 9   spc_latin   652169 non-null  object
 10  spc_common  652169 non-null  object
 11  steward     164350 non-null  object
 12  guards      79866 non-null   object
 13  sidewalk    652172 non-null  object
 14  user_type   683788 non-null  object
 15  problems    225844 non-null  object
 16  root_stone  683788 non-null  object
 17  root_grate  683788 non-null  object
 18  root_other  683788 non-null  object
 19  trnk_wire   683788 non-null  object
 20  trnk_light  683788 non-null  object
 21  trnk_other  683788 non-null  object
 22  brnch_ligh  683788 non-null  object
 23  brnch_shoe  683788 non-null  object
 24  brnch_othe  683788 non-null  object
 25  address     683788 non-null  object
 26  zipcode     683788 non-null  int64 
 27  zip_city    683788 non-null  object
 28  cb_num      683788 non-null  int64 
 29  borocode    683788 non-null  int64 
 30  boroname    683788 non-null  object
 31  cncldist    683788 non-null  int64 
 32  st_assem    683788 non-null  int64 
 33  st_senate   683788 non-null  int64 
 34  nta         683788 non-null  object
 35  nta_name    683788 non-null  object
 36  boro_ct     683788 non-null  int64
 37  state       683788 non-null  object
 38  Latitude    683788 non-null  float64
 39  longitude   683788 non-null  float64
 40  x_sp        683788 non-null  object
 41  y_sp        683788 non-null  object
dtypes: float64(2), int64(9), object(31)
memory usage: 219.1+ MB
```

```
# health status of trees
tree_census.health.value_counts(dropna=False)
```

```
health
Good    528850
Fair     96504
NaN      31616
Poor     26818
Name: count, dtype: int64
```

```
# get status on the trees
tree_census.status.value_counts(dropna=False)
```

```
status
Alive    652173
Stump     17654
Dead      13961
Name: count, dtype: int64
```

```
# subset of the original, removed columns not interested in
trees_subset = tree_census[['tree_id', 'tree_dbh',
      'stump_diam', 'curb_loc', 'status', 'health', 'spc_latin', 'spc_commo
      'steward', 'guards', 'sidewalk', 'user_type', 'problems', 'root_stone
      'root_grate', 'root_other', 'trnk_wire', 'trnk_light', 'trnk_other',
      'brnch_ligh', 'brnch_shoe', 'brnch_othe']]
# list the first 5 rows of the new subset
trees_subset.head()
```

	tree_id	tree_dbh	stump_diam	curb_loc	status	health	spc_latin	sp
0	180,683	3	0	OnCurb	Alive	Fair	Acer rubrum	
1	200,540	21	0	OnCurb	Alive	Fair	Quercus palustris	
2	204,026	3	0	OnCurb	Alive	Good	Gleditsia triacanthos var. inermis	h
3	204,337	10	0	OnCurb	Alive	Good	Gleditsia triacanthos var. inermis	h
4	189,565	21	0	OnCurb	Alive	Good	Tilia americana	

5 rows × 22 columns

```
# check for any null values
trees_subset.isna().sum()
```

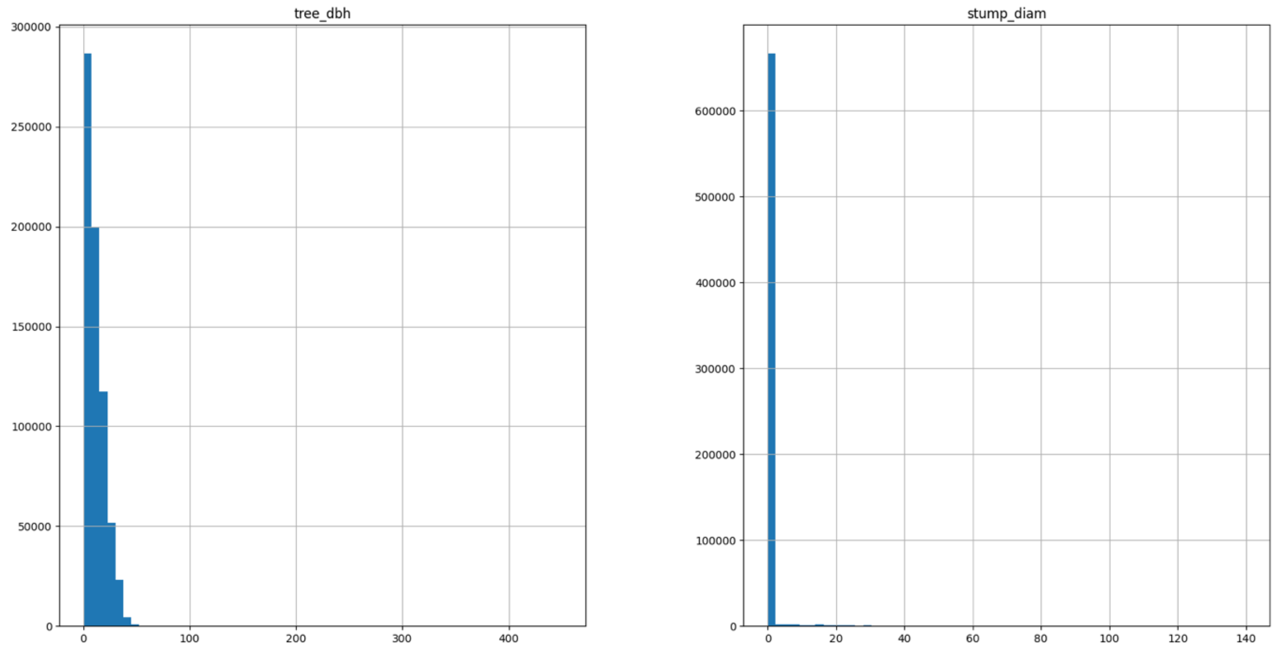
```
tree_id            0
tree_dbh           0
stump_diam         0
curb_loc           0
status             0
health         31616
spc_latin      31619
spc_common     31619
steward       519438
guards        603922
sidewalk       31616
user_type          0
problems      457944
root_stone         0
root_grate         0
root_other         0
trnk_wire          0
trnk_light         0
trnk_other         0
brnch_ligh         0
brnch_shoe         0
brnch_othe         0
dtype: int64
```

```
# show all that are none values in health, alot of missing values NaN
tree_census.describe()
```

	tree_dbh	stump_diam	zipcode	cb_num	borocode
count	683788.000000	683788.000000	683788.000000	683788.000000	683788.00000
mean	11.279787	0.432463	10916.246044	343.505404	3.35850
std	8.723042	3.290241	651.553364	115.740601	1.16674
min	0.000000	0.000000	83.000000	101.000000	1.00000
25%	4.000000	0.000000	10451.000000	302.000000	3.00000
50%	9.000000	0.000000	11214.000000	402.000000	4.00000
75%	16.000000	0.000000	11365.000000	412.000000	4.00000
max	450.000000	140.000000	11697.000000	503.000000	5.00000

```
# generate histogram of data distribution
trees_subset.hist(bins=60, figsize=(20,10))
```

```
array([[<Axes: title={'center': 'tree_dbh'}>,
        <Axes: title={'center': 'stump_diam'}>]], dtype=object)
```



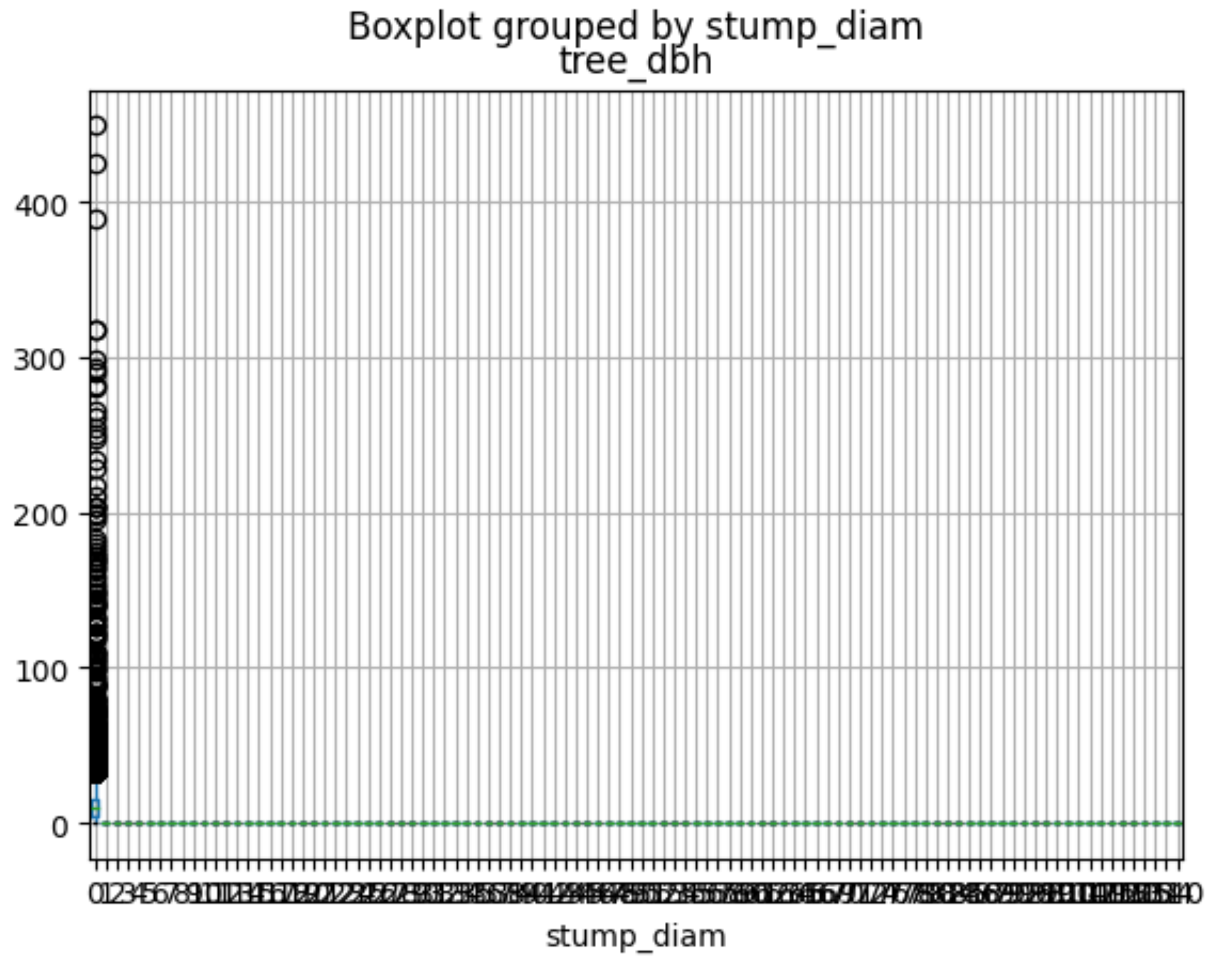
```
# trees larger than 50
big_trees = trees_subset[trees_subset['tree_dbh']> 50]
big_trees.head()
```

	tree_id	tree_dbh	stump_diam	curb_loc	status	health	spc_l
2385	168,583	425	0	OnCurb	Alive	Good	Qu b
3724	199,546	51	0	OnCurb	Alive	Good	saccha
4874	139,665	72	0	OffsetFromCurb	Alive	Good	saccha
6711	209,349	122	0	OnCurb	Alive	Good	Qu pa
10053	215,075	169	0	OnCurb	Alive	Good	Gle triaca var. ir

5 rows × 22 columns

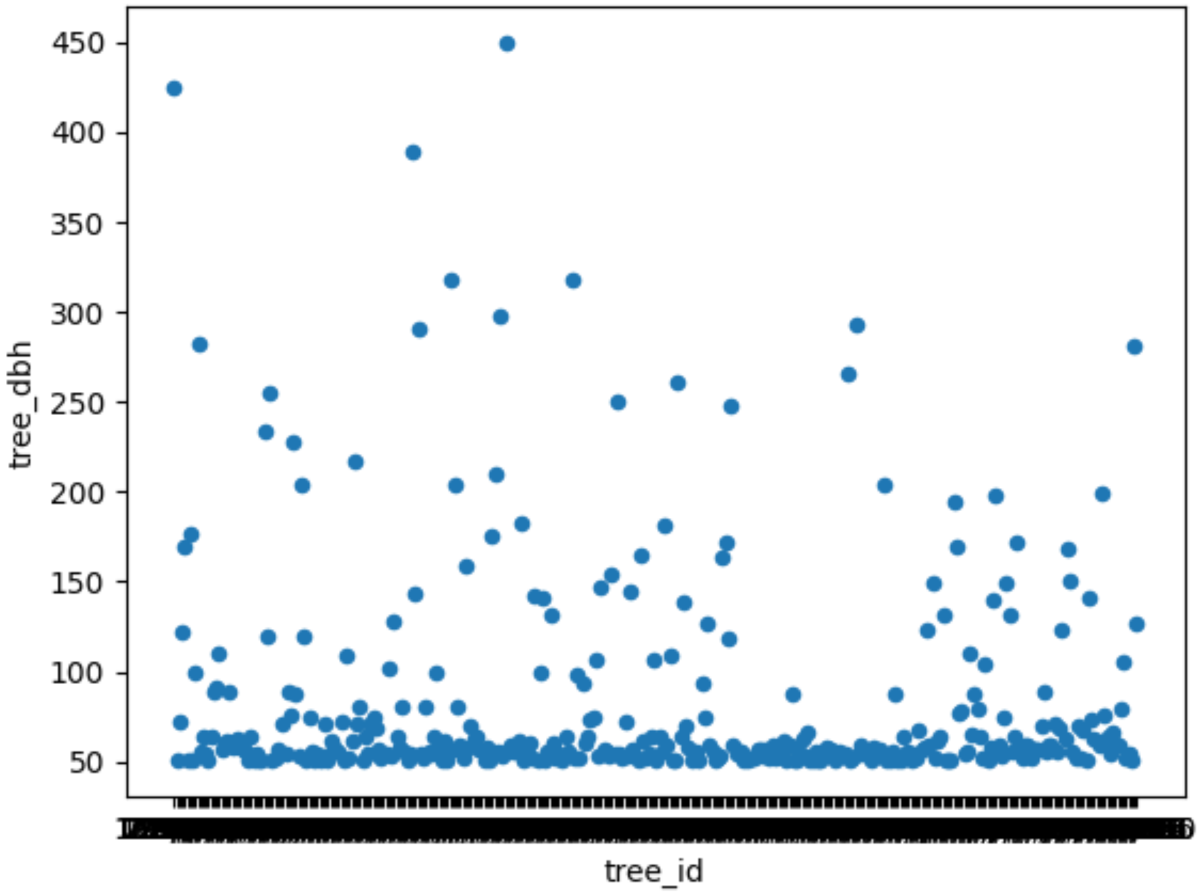
```
# box plot
tree_census.boxplot(column='tree_dbh', by='stump_diam')
```

```
<Axes: title={'center': 'tree_dbh'}, xlabel='stump_diam'>
```



```
atter plolt
trees[['tree_id', 'tree_dbh']].plot(kind='scatter',x='tree_id', y='tree_dbh'
```

```
<Axes: xlabel='tree_id', ylabel='tree_dbh'>
```


tree_dbh
tree_id