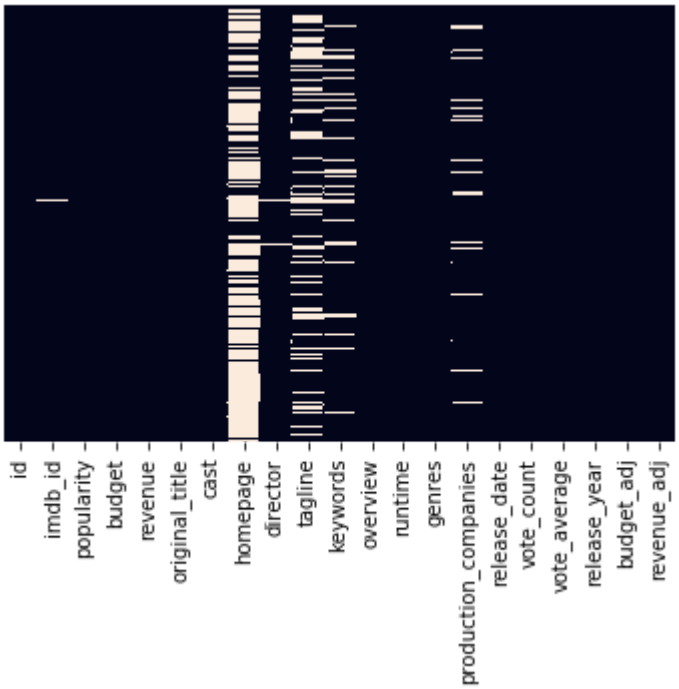
ML-MINOR-JUNE

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
df=pd.read_csv("tmdb-movies (1).csv")
```

In [3]:
```
import seaborn as sns
sns.heatmap(df.isnull(),yticklabels=False,cbar=False)#find the null values
```

Out[3]: <matplotlib.axes._subplots.AxesSubplot at 0x20725eafb88>



In [4]:
```
df.isnull().sum()#check null values
```

Out[4]:
```
id                        0
imdb_id                  10
popularity                0
budget                    0
revenue                   0
original_title            0
cast                     76
homepage               7930
director                 44
tagline                2824
keywords               1493
overview                  4
runtime                   0
genres                   23
production_companies    1030
release_date              0
vote_count                0
vote_average              0
release_year              0
budget_adj                0
revenue_adj               0
dtype: int64
```

In [5]:
```
df.dropna(inplace = True)
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
id                      0
imdb_id                 0
popularity              0
budget                  0
revenue                 0
original_title          0
cast                    0
homepage                0
director                0
tagline                 0
keywords                0
overview                0
runtime                 0
genres                  0
production_companies    0
release_date            0
vote_count              0
vote_average            0
release_year            0
budget_adj              0
revenue_adj             0
dtype: int64
```

In [7]:
```
df
```

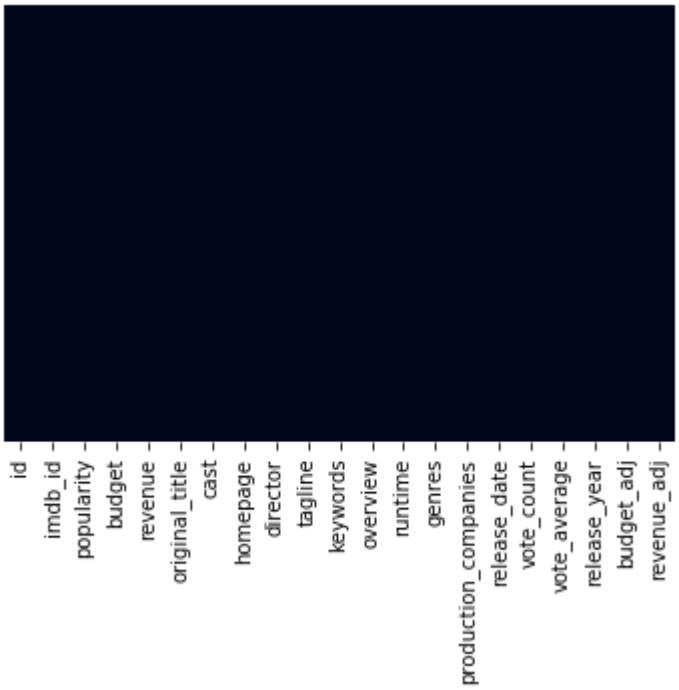
Out[7]:

	id	imdb_id	popularity	budget	revenue	original_title	cast	
0	135397	tt0369610	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	
1	76341	tt1392190	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	
2	262500	tt2908446	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	http
3	140607	tt2488496	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	
4	168259	tt2820852	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	
...	...	...	...	...	...	...	...	
10724	668	tt0064757	1.778746	7000000	81974493	On Her Majesty's Secret Service	George Lazenby|Diana Rigg|Telly Savalas|Gabrie...	http
10759	948	tt0077651	1.198849	300000	70000000	Halloween	Donald Pleasence|Jamie Lee Curtis|P.J. Soles|N...	http:
10760	8469	tt0077975	1.157930	2700000	141000000	Animal House	John Belushi|Tim Matheson|John Vernon|Verna Bl...	
10817	13963	tt0077838	0.064029	0	321952	The Last Waltz	Robbie Robertson|Rick Danko|Levon Helm|Richard...	
10819	16214	tt0077696	0.044675	0	78000000	Hooper	Burt Reynolds|Robert Klein|Adam West|Jan-Micha...	

1992 rows × 21 columns

In [8]:
```
import seaborn as sns
sns.heatmap(df.isnull(),yticklabels=False,cbar=False)
```

Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x207245e7c88>



QUESTION 5

In [9]:
```
df[df['release_year']==2006]['runtime'].mean() #average runtime of movies in t
he year 2006
```

Out[9]: 109.96739130434783

In [ ]:

In [ ]:

In [10]:
```
df
```

Out[10]:

	id	imdb_id	popularity	budget	revenue	original_title	cast	
0	135397	tt0369610	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	
1	76341	tt1392190	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	
2	262500	tt2908446	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	http
3	140607	tt2488496	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	
4	168259	tt2820852	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	
...	...	...	...	...	...	...	...	
10724	668	tt0064757	1.778746	7000000	81974493	On Her Majesty's Secret Service	George Lazenby|Diana Rigg|Telly Savalas|Gabrie...	http
10759	948	tt0077651	1.198849	300000	70000000	Halloween	Donald Pleasence|Jamie Lee Curtis|P.J. Soles|N...	http:
10760	8469	tt0077975	1.157930	2700000	141000000	Animal House	John Belushi|Tim Matheson|John Vernon|Verna Bl...	
10817	13963	tt0077838	0.064029	0	321952	The Last Waltz	Robbie Robertson|Rick Danko|Levon Helm|Richard...	
10819	16214	tt0077696	0.044675	0	78000000	Hooper	Burt Reynolds|Robert Klein|Adam West|Jan-Micha...	

1992 rows × 21 columns

In [ ]:
```

```

Question 1

In [13]:
```
df[['original_title','budget']].sort_values(by=['budget'] ,ascending=False )[:3] #Third highest budget is titanic
```

Out[13]:

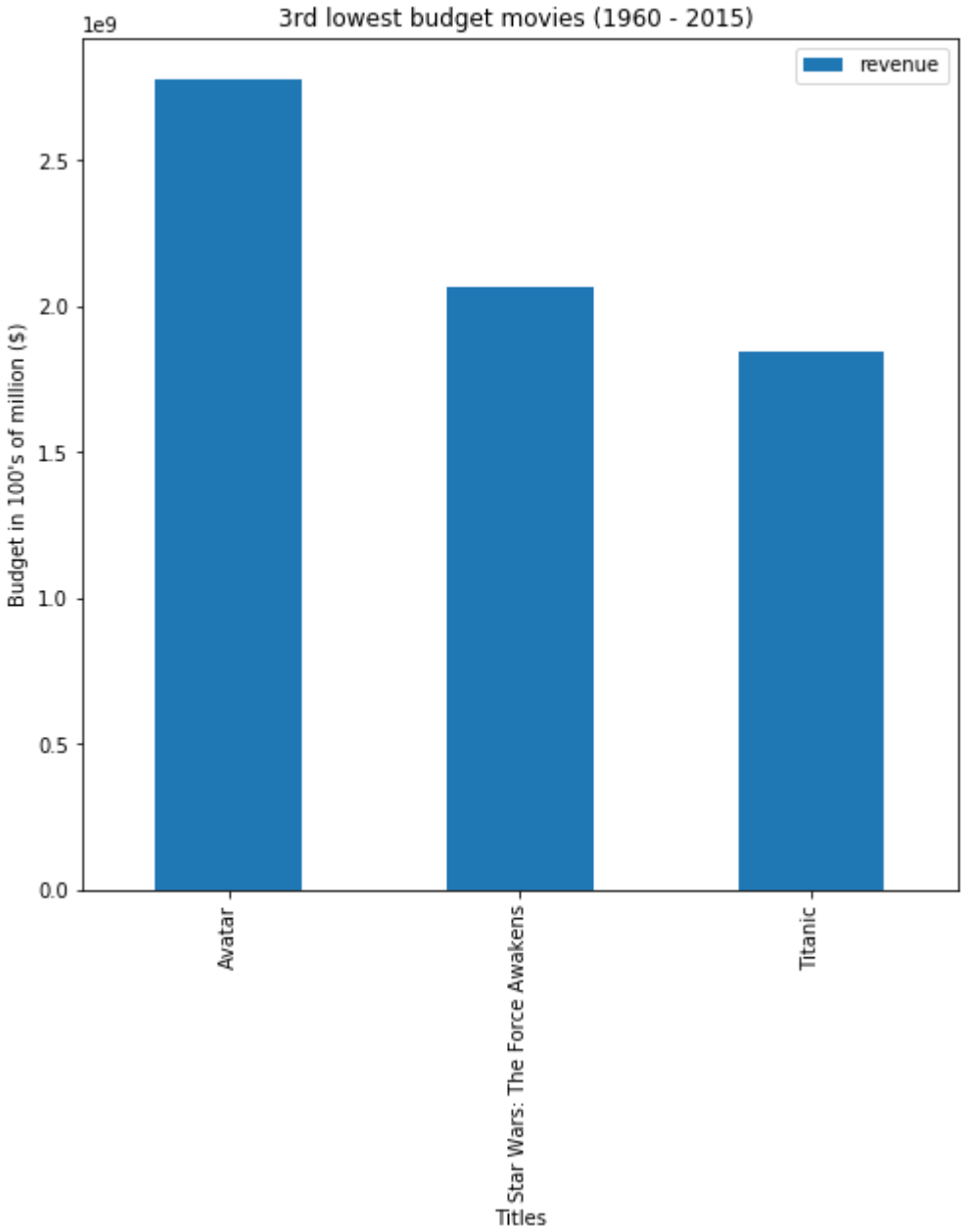
	original_title	budget
2244	The Warrior's Way	425000000
3375	Pirates of the Caribbean: On Stranger Tides	380000000
7387	Pirates of the Caribbean: At World's End	300000000

In [14]:
```
df[['original_title','budget']].sort_values(by=['budget'] ,ascending=False )[2:3] #Third Highest budget
```

Out[14]:

	original_title	budget
7387	Pirates of the Caribbean: At World's End	300000000

In [16]:
```
#using matplotlib
sorted_budget = df['revenue'].sort_values(ascending=False)[:3]
high_budget=pd.DataFrame()
titles_exp=[]
budgets=[]
for i in sorted_budget.index:
    titles_exp.append(df.loc[i,'original_title'])
    budgets.append(sorted_budget.loc[i])
high_budget['Titles']=titles_exp
high_budget['revenue']=budgets
high_budget.set_index('Titles',inplace=True)
high_budget.plot(kind ='bar',figsize=(8,8))
plt.title('3rd lowest budget movies (1960 - 2015) ');
plt.ylabel('Budget in 100\'s of million ($)');
```

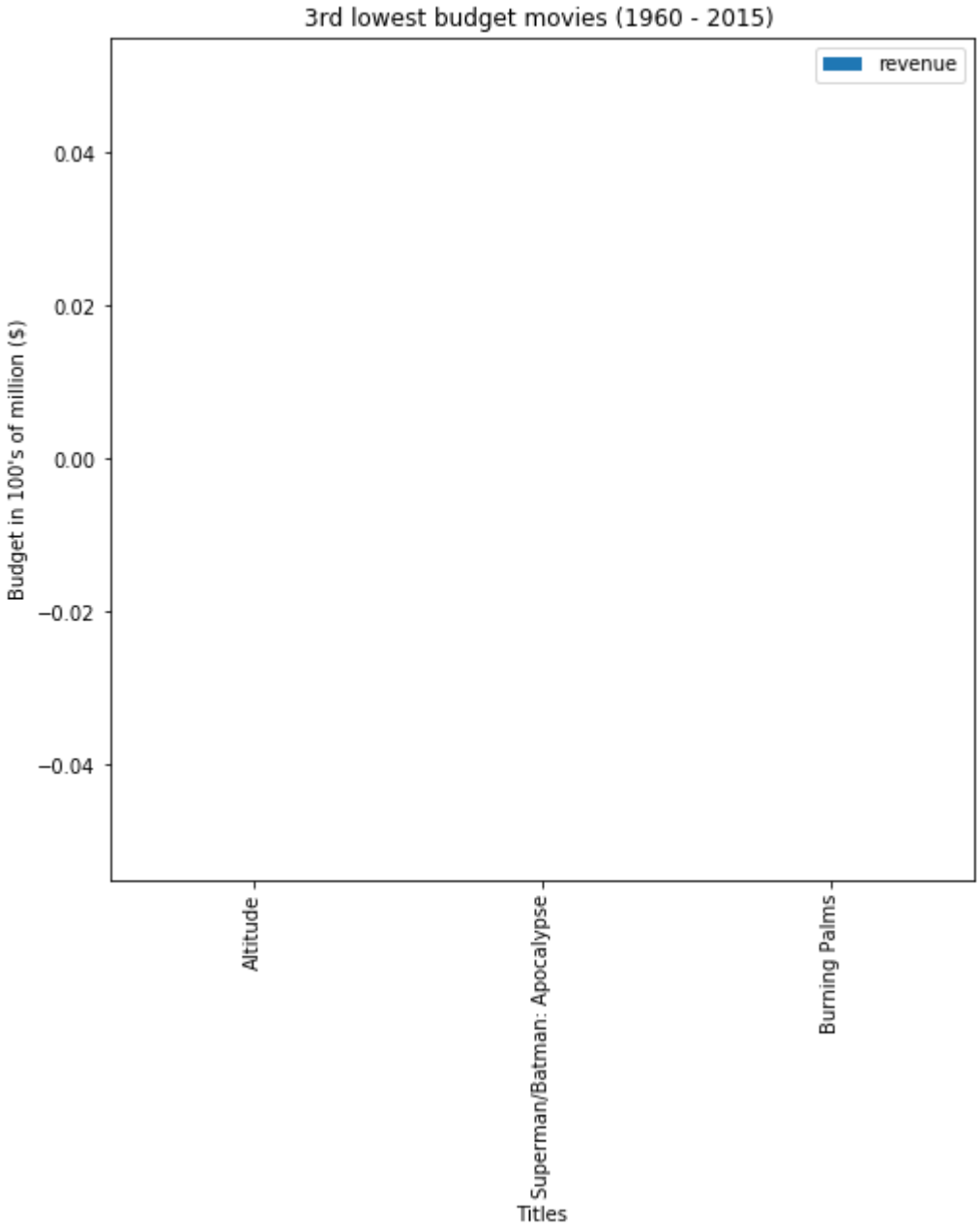


In [70]:
```
df[['original_title','budget']].sort_values(by=['budget'] ,ascending=True )[0:3] #lowest budget
```

Out[70]:

	original_title	budget
10819	Hooper	0
2130	Dog Pound	0
2160	Justice League: Crisis on Two Earths	0

In [17]:
```
#using matplotlib to show lowest budget
sorted_budget = df['revenue'].sort_values(ascending=True)[:3]
low_budget=pd.DataFrame()
titles_exp=[]
budgets=[]
for i in sorted_budget.index:
    titles_exp.append(df.loc[i,'original_title'])
    budgets.append(sorted_budget.loc[i])
low_budget['Titles']=titles_exp
low_budget['revenue']=budgets
low_budget.set_index('Titles',inplace=True)
low_budget.plot(kind ='bar',figsize=(8,8))
plt.title('3rd lowest budget movies  ');
plt.ylabel('Budget in 100\'s of million ');
```



In [19]:
```
##budgetadj and revenueadj show the budget and revenue of the associated movie
in terms of 2010 dollars, accounting for inflation over time.
```

if use budget_adj

In [20]:
```
df[['original_title','budget_adj']].sort_values(by=['budget_adj'] ,ascending=False )[2:3] #Third Highest budget
```

Out[20]:

	original_title	budget_adj
7387	Pirates of the Caribbean: At World's End	3.155006e+08

In [23]:
```
df[['original_title','budget_adj']].sort_values(by=['budget_adj'] ,ascending=True )[2:3] #3rd lowest budget
```

Out[23]:

	original_title	budget_adj
2075	Senna	0.0

QUESTION 4

In [ ]:
```

```

In [80]:
```
#Most earned Revenue
df[['original_title','revenue']].sort_values(by=['revenue'] ,ascending=False )[0:1]
```

Out[80]:

	original_title	revenue
1386	Avatar	2781505847

In [81]:
```
#Least earned Revenue
df[['original_title','revenue']].sort_values(by=['revenue'] ,ascending=True )[0:1]
```

Out[81]:

	original_title	revenue
2197	Altitude	0

In [82]:
```
df[['original_title','revenue']].min()
```

Out[82]:
```
original_title    (500) Days of Summer
revenue                              0
dtype: object
```

In [ ]:
```

```

In [ ]:
```
##QUESTION 5
```

In [85]:
```
df[df['release_year']==2006]['runtime'].mean() #average runtime of movies in t
he year 2006
```

Out[85]:
```
109.96739130434783
```

In [ ]:
```

```

QUESTION 6

In [86]:
```
df["production_companies"]=df["production_companies"].fillna(df["production_co
mpanies"].mode()[0])
```

In [88]:
```
df
```

Out[88]:

	id	imdb_id	popularity	budget	revenue	original_title	cast	
0	135397	tt0369610	32.985763	150000000	1513528810	Jurassic World	Chris Pratt|Bryce Dallas Howard|Irrfan Khan|Vi...	
1	76341	tt1392190	28.419936	150000000	378436354	Mad Max: Fury Road	Tom Hardy|Charlize Theron|Hugh Keays-Byrne|Nic...	
2	262500	tt2908446	13.112507	110000000	295238201	Insurgent	Shailene Woodley|Theo James|Kate Winslet|Ansel...	http
3	140607	tt2488496	11.173104	200000000	2068178225	Star Wars: The Force Awakens	Harrison Ford|Mark Hamill|Carrie Fisher|Adam D...	
4	168259	tt2820852	9.335014	190000000	1506249360	Furious 7	Vin Diesel|Paul Walker|Jason Statham|Michelle ...	
...	...	...	...	...	...	...	...	
10724	668	tt0064757	1.778746	7000000	81974493	On Her Majesty's Secret Service	George Lazenby|Diana Rigg|Telly Savalas|Gabrie...	http
10759	948	tt0077651	1.198849	300000	70000000	Halloween	Donald Pleasence|Jamie Lee Curtis|P.J. Soles|N...	http:
10760	8469	tt0077975	1.157930	2700000	141000000	Animal House	John Belushi|Tim Matheson|John Vernon|Verna Bl...	
10817	13963	tt0077838	0.064029	0	321952	The Last Waltz	Robbie Robertson|Rick Danko|Levon Helm|Richard...	
10819	16214	tt0077696	0.044675	0	78000000	Hooper	Burt Reynolds|Robert Klein|Adam West|Jan-Micha...	

1992 rows × 21 columns

In [99]:
```
df[['production_companies','revenue']].sort_values(by=['revenue'] ,ascending=True )[0:3]
```

Out[99]:

	production_companies	revenue
2197	Darclight Films	0
2141	DC Comics|Warner Premiere	0
2145	Films In Motion	0

In [3]:
```
!jupyter nbconvert --to html ML_MINOR_JUNE.ipynb
```

```
This application is used to convert notebook files (*.ipynb) to various other
formats.

WARNING: THE COMMANDLINE INTERFACE MAY CHANGE IN FUTURE RELEASES.

Options

-------



Arguments that take values are actually convenience aliases to full
Configurables, whose aliases are listed on the help line. For more informatio
n
on full configurables, see '--help-all'.


--debug

    set log level to logging.DEBUG (maximize logging output)

--generate-config

    generate default config file

-y

    Answer yes to any questions instead of prompting.

--execute

    Execute the notebook prior to export.

--allow-errors

    Continue notebook execution even if one of the cells throws an error and
include the error message in the cell output (the default behaviour is to abo
rt conversion). This flag is only relevant if '--execute' was specified, too.

--stdin

    read a single notebook file from stdin. Write the resulting notebook with
default basename 'notebook.*'

--stdout

    Write notebook output to stdout instead of files.

--inplace

    Run nbconvert in place, overwriting the existing notebook (only
    relevant when converting to notebook format)

--clear-output

    Clear output of current file and save in place,
    overwriting the existing notebook.
```

```
--no-prompt

    Exclude input and output prompts from converted document.

--no-input

    Exclude input cells and output prompts from converted document.
    This mode is ideal for generating code-free reports.
--log-level=<Enum> (Application.log_level)

    Default: 30

    Choices: (0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR', 'CRITI
CAL')

    Set the log level by value or name.

--config=<Unicode> (JupyterApp.config_file)

    Default: ''

    Full path of a config file.

--to=<Unicode> (NbConvertApp.export_format)

    Default: 'html'

    The export format to be used, either one of the built-in formats

    ['asciidoc', 'custom', 'html', 'latex', 'markdown', 'notebook', 'pdf',

    'python', 'rst', 'script', 'slides'] or a dotted object name that represe
nts

    the import path for an `Exporter` class

--template=<Unicode> (TemplateExporter.template_file)

    Default: ''

    Name of the template file to use

--writer=<DottedObjectName> (NbConvertApp.writer_class)

    Default: 'FilesWriter'

    Writer class used to write the  results of the conversion

--post=<DottedOrNone> (NbConvertApp.postprocessor_class)

    Default: ''

    PostProcessor class used to write the results of the conversion

--output=<Unicode> (NbConvertApp.output_base)
```

```
    Default: ''

    overwrite base name use for output files. can only be used when convertin
g

    one notebook at a time.

--output-dir=<Unicode> (FilesWriter.build_directory)

    Default: ''

    Directory to write output(s) to. Defaults to output to the directory of e
ach

    notebook. To recover previous default behaviour (outputting to the curren
t

    working directory) use . as the flag value.

--reveal-prefix=<Unicode> (SlidesExporter.reveal_url_prefix)

    Default: ''

    The URL prefix for reveal.js (version 3.x). This defaults to the reveal C
DN,

    but can be any url pointing to a copy  of reveal.js.

    For speaker notes to work, this must be a relative path to a local  copy
of

    reveal.js: e.g., "reveal.js".

    If a relative path is given, it must be a subdirectory of the current

    directory (from which the server is run).

    See the usage documentation

    (https://nbconvert.readthedocs.io/en/latest/usage.html#reveal-js-html-

    slideshow) for more details.

--nbformat=<Enum> (NotebookExporter.nbformat_version)

    Default: 4

    Choices: [1, 2, 3, 4]

    The nbformat version to write. Use this to downgrade notebooks.

To see all available configurables, use `--help-all`

Examples
--------

    The simplest way to use nbconvert is
```

```
    > jupyter nbconvert mynotebook.ipynb

    which will convert mynotebook.ipynb to the default format (probably HTM
L).

    You can specify the export format with `--to`.
    Options include ['asciidoc', 'custom', 'html', 'latex', 'markdown', 'note
book', 'pdf', 'python', 'rst', 'script', 'slides'].

    > jupyter nbconvert --to latex mynotebook.ipynb

    Both HTML and LaTeX support multiple output templates. LaTeX includes
    'base', 'article' and 'report'.  HTML includes 'basic' and 'full'. You
    can specify the flavor of the format used.

    > jupyter nbconvert --to html --template basic mynotebook.ipynb

    You can also pipe the output to stdout, rather than a file

    > jupyter nbconvert mynotebook.ipynb --stdout

    PDF is generated via latex

    > jupyter nbconvert mynotebook.ipynb --to pdf

    You can get (and serve) a Reveal.js-powered slideshow

    > jupyter nbconvert myslides.ipynb --to slides --post serve

    Multiple notebooks can be given at the command line in a couple of
    different ways:

    > jupyter nbconvert notebook*.ipynb
    > jupyter nbconvert notebook1.ipynb notebook2.ipynb

    or you can specify the notebooks list in a config file, containing::

        c.NbConvertApp.notebooks = ["my_notebook.ipynb"]

    > jupyter nbconvert --config mycfg.py


[NbConvertApp] WARNING | pattern 'ML_MINOR_JUNE.ipynb' matched no files
```

In [ ]: