welly_test

December 14, 2020

1 Using Welly Test

Created by Thomas Martin, PhD canidate at CoRE at Colorado School of Mines. Personal website is here, and email is thomasmartin@mines.edu. One page research profile of CoRE is here. Feel free to reach out with any questions.

[1]:
```
!pip install lasio
!pip install welly==0.4.7
```

```
Collecting lasio
  Downloading https://files.pythonhosted.org/packages/5e/8e/ce58a22ec8454a12f923
33a50f2add5f6131218c4815952d6ca7cbd578f0/lasio-0.28-py3-none-any.whl
Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-packages
(from lasio) (1.18.5)
Installing collected packages: lasio
Successfully installed lasio-0.28
Collecting welly==0.4.7
  Downloading https://files.pythonhosted.org/packages/a3/a0/6e8a5e58fb0c53
eb34706e755055eebfbf30867169136c84fa3d869be051/welly-0.4.7.tar.gz (47kB)
     || 51kB 2.6MB/s
Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-
packages (from welly==0.4.7) (1.18.5)
Requirement already satisfied: scipy in /usr/local/lib/python3.6/dist-packages
(from welly==0.4.7) (1.4.1)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.6/dist-
packages (from welly==0.4.7) (3.2.2)
Requirement already satisfied: lasio in /usr/local/lib/python3.6/dist-packages
(from welly==0.4.7) (0.28)
Collecting striplog
  Downloading https://files.pythonhosted.org/packages/4a/8c/73e1ff0eb8a94b
01e10b893ce909afbdf4d837f33a177987c3b5d88c79b1/striplog-0.8.7.tar.gz (78kB)
     || 81kB 4.2MB/s
Requirement already satisfied: tqdm in /usr/local/lib/python3.6/dist-
packages (from welly==0.4.7) (4.41.1)
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.6
/dist-packages (from matplotlib->welly==0.4.7) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.6/dist-
```

```
packages (from matplotlib->welly==0.4.7) (0.10.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in
/usr/local/lib/python3.6/dist-packages (from matplotlib->welly==0.4.7) (2.4.7)
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.6
/dist-packages (from matplotlib->welly==0.4.7) (2.8.1)
Requirement already satisfied: six in /usr/local/lib/python3.6/dist-packages
(from cycler>=0.10->matplotlib->welly==0.4.7) (1.15.0)
Building wheels for collected packages: welly, striplog
  Building wheel for welly (setup.py) ... done
  Created wheel for welly: filename=welly-0.4.7-cp36-none-any.whl size=51556
sha256=1093c139b8519e50561f0c1d87670dd23312f60741bc51bd12a848b694fa78c7
  Stored in directory: /root/.cache/pip/wheels/1f/a6/e3/8543907612eab77a66ec2193
df115c2580fd863bea3c092a44
  Building wheel for striplog (setup.py) ... done
  Created wheel for striplog: filename=striplog-0.8.7-cp36-none-any.whl
size=83396
sha256=149dfd4752ade02c26dd65ddb8c947db60925b4a804bc8a2564752459d195ebb
  Stored in directory: /root/.cache/pip/wheels/20/b6/dc/ee41c95d309d463308768f96
7efeb67884a7b9f6560a127728
Successfully built welly striplog
Installing collected packages: striplog, welly
Successfully installed striplog-0.8.7 welly-0.4.7
```

[2]:
```
import lasio
print('lasio ver:',lasio.__version__)
import welly
print('welly ver:', welly.__version__)

import pandas as pd
import numpy as np

from glob import glob
```

```
lasio ver: 0.28
welly ver: 0.4.7
```

[19]:
```
cd drive/My Drive/OVV_Learn_welly_lasio
```

```
/content/drive/My Drive/OVV_Learn_welly_lasio
```

[20]:
```
ls
```

```
1044753454.las                  awesome-plot.pdf  well_data.csv
1_UsingColab_andLasio.ipynb     kgs_2014/
welly_test.ipynb
2_well_loading_with_welly.ipynb  petro-stats.csv
3_sliderbar_decline.ipynb       RAPI3723253.csv
```

1.1 Let's load these in using welly

[6]:
```
from welly import Project
from welly import Well

from welly.defaults import ALIAS
```

[7]:
```
p = Project.from_las("1044753454.las")
```

```
0it [00:00, ?it/s]/usr/local/lib/python3.6/dist-packages/welly/well.py:173:
FutureWarning: From v0.5 the default will be 'original', keeping whatever is
used in the LAS file. If you want to force conversion to metres, change your
code to use `index='m'`.
  warnings.warn(m, FutureWarning)
1it [00:03,  3.03s/it]
```

Deleted some wells just for the sake of speed, but ~500 wells took <10 minutes on a weird Colab enviroment. I would budget 1-2s per well.

[8]:
```
p
```

[8]:
```
Project(1 wells: 15175221820000)
```

[9]:
```
df_orig = p.df()
df_orig
```

[9]:

UWI	Depth	CALIPER	DT	GAMMARAY	ITTT	...	RT90	RXO	RXRT	SP
15175221820000	495.1476	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
	495.2238	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
	495.3000	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
	495.3762	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
	495.4524	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
...		...	..	...	...	...	...	...	...	..
	1793.3670	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
	1793.4432	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
	1793.5194	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
	1793.5956	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
	1793.6718	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN

[17042 rows x 36 columns]

[10]:
```
df_alias = p.df(keys=['CAL', 'GR', 'DT', 'SP', 'DENS', 'PE', 'RESD', 'PHIN',␣
 ↪'PHID'], alias=ALIAS)
df_alias = df_alias.dropna(thresh=7) # dropping all rows with 7 NaN's, just for␣
 ↪this demo
df_alias.head(10)
```

[10]:

UWI	Depth	CAL	GR	DT	...	RESD	PHIN	PHID
15175221820000	515.1882	4.5	41.8421	61.0528	...	94.9999	0.0000	-0.0000

```
515.2644  4.5  39.9914  61.8054  ...  189.9998  0.0000 -0.0001
515.3406  4.5  38.8726  62.5141  ...  284.9996  0.0000 -0.0001
515.4168  4.5  37.7538  63.3363  ...  379.9995  0.0000 -0.0002
515.4930  4.5  39.1040  64.0372  ...  474.9994  0.0000 -0.0002
515.5692  4.5  40.4541  64.5227  ...  569.9993  0.0000 -0.0003
515.6454  4.5  41.4804  64.6648  ...  664.9991  0.0000 -0.0003
515.7216  4.5  42.5067  64.3887  ...  759.9990  0.0001 -0.0003
515.7978  4.5  41.6865  63.7198  ...  854.9989  0.0001 -0.0004
515.8740  4.5  40.8663  62.8735  ...  949.9988  0.0001 -0.0004

[10 rows x 9 columns]
```

1.1.1 Welly 0.4.8

[1]:
```
!pip install welly --upgrade
```

```
Requirement already up-to-date: welly in /usr/local/lib/python3.6/dist-packages
(0.4.8)
Requirement already satisfied, skipping upgrade: lasio in
/usr/local/lib/python3.6/dist-packages (from welly) (0.28)
Requirement already satisfied, skipping upgrade: scipy in
/usr/local/lib/python3.6/dist-packages (from welly) (1.4.1)
Requirement already satisfied, skipping upgrade: numpy in
/usr/local/lib/python3.6/dist-packages (from welly) (1.18.5)
Requirement already satisfied, skipping upgrade: striplog in
/usr/local/lib/python3.6/dist-packages (from welly) (0.8.7)
Requirement already satisfied, skipping upgrade: matplotlib in
/usr/local/lib/python3.6/dist-packages (from welly) (3.2.2)
Requirement already satisfied, skipping upgrade: tqdm in
/usr/local/lib/python3.6/dist-packages (from welly) (4.41.1)
Requirement already satisfied, skipping upgrade: cycler>=0.10 in
/usr/local/lib/python3.6/dist-packages (from matplotlib->welly) (0.10.0)
Requirement already satisfied, skipping upgrade:
pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.6/dist-
packages (from matplotlib->welly) (2.4.7)
Requirement already satisfied, skipping upgrade: python-dateutil>=2.1 in
/usr/local/lib/python3.6/dist-packages (from matplotlib->welly) (2.8.1)
Requirement already satisfied, skipping upgrade: kiwisolver>=1.0.1 in
/usr/local/lib/python3.6/dist-packages (from matplotlib->welly) (1.3.1)
Requirement already satisfied, skipping upgrade: six in /usr/local/lib/python3.6
/dist-packages (from cycler>=0.10->matplotlib->welly) (1.15.0)
```

[6]:
```
import lasio
print('lasio ver:',lasio.__version__)
import welly
print('welly ver:', welly.__version__)
```

```
from welly import Project
from welly import Well
```

```
lasio ver: 0.28
welly ver: 0.4.8
```

[12]:
```
foo = welly.defaults.ALIAS
```

[13]:
```
foo
```

[13]:
```
{'CAL': ['HHCA',
  'SA',
  'HCALX',
  'DCAL',
  'CLXC',
  'CAX',
  'CAPD',
  'CANC',
  'CALZ',
  'CALX',
  'CALI',
  'CAL1',
  'CAL',
  'CADF',
  'CADE',
  'CA2',
  'CA1',
  'C13Z',
  'C13P',
  'C13M',
  'C13L',
  'C13I',
  'C13H',
  'C13-A',
  'C13A',
  'C13',
  'ACAL',
  'C1',
  'C2',
  'C24',
  'C24A',
  'C24-A',
  'C24H',
  'C24I',
  'C24L',
  'C24M',
  'C24P',
  'C24Z',
```

```
'CA',
'CAL2',
'CAL3',
'CALA',
'CAL-A',
'CALD',
'CALE',
'CALH',
'CALI_SPCS',
'CALL',
'CALM',
'CALN',
'CALP',
'CALS',
'CALT',
'CALX-A',
'CALXH',
'CALXM',
'CALX-ML',
'CALXQ8',
'CALXQH',
'CALX-R',
'CALY',
'CALY-A',
'CALYH',
'CALYM',
'CALY-ML',
'CALYQ8',
'CALYQH',
'CAY',
'CLCM',
'CLDC',
'CLDM',
'CLLO',
'CLTC',
'CLYC',
'CQLI',
'HCAL',
'HCAL2',
'HCALI',
'HCALY',
'HD',
'HD_1',
'HD1',
'HD2',
'HD3',
'HDAR',
```

```
 'HDIA',
 'HDMI',
 'HDMN',
 'HDMX',
 'HLCA',
 'LCAL',
 'MCAL',
 'MLTC',
 'TAC2',
 'C3',
 'CLS2',
 'MBTC',
 'TACC',
 'DCAV',
 'CALCFM',
 'CALI_WL',
 'CALI_LWD',
 'APPC'],
'DCOR': ['Z-COR',
 'HHDR',
 'ZCORQH',
 'ZCOR2QH',
 'ZCOR2',
 'ZCOR',
 'RPCL_DCOR',
 'HHDRA',
 'HDRH',
 'HDRA',
 'HBDC',
 'DRHO',
 'DRH',
 'DECR',
 'DCOR',
 'DC',
 'DENC',
 '2DRH'],
'DENS': ['RHOB', 'RHO8', 'DEN', 'DENS', 'ZDEN', 'ZDNC', 'RHOM', 'RHOZ'],
'DT': ['AC1',
 'TTC',
 'DTP',
 'DTMX',
 'DTMN',
 'DTLF',
 'DTL',
 'DTCO',
 'DTCM',
 'DTC2',
```

```
 'DTC',
 'DT4P_C',
 'DT4P',
 'DT41',
 'DT',
 'AC',
 'ACCO',
 'ACL',
 'ACN',
 'DT24',
 'DT24-A',
 'DT24AQI',
 'DT24QI',
 'DT24SQA',
 'DT34',
 'DT35',
 'DT5',
 'DTC1',
 'DTCA',
 'DTCR',
 'DTCX',
 'DTSC',
 'MSTT'],
'DTS': ['TTS', 'SHSL', 'DTSF', 'DTSD', 'DTS', 'DT4S', 'D4S', 'DTSS'],
'GR': ['HEHG',
 'GRS',
 'HGR-5A',
 'HGR-1A',
 'HGR_STGC',
 'HGR',
 'HCGR',
 'GRQH',
 'GRNP',
 'GRNC',
 'GRN',
 'GRGM',
 'GRGC',
 'GRDI',
 'GRDE',
 'GRCO',
 'GRC',
 'GR-A',
 'GR_ARC_FILT',
 'GGCE',
 'EHGR',
 'ECGR',
 'CGR',
```

```
 'G',
 'GAB',
 'GAM',
 'GAM1',
 'GAPI',
 'GR',
 'GR_AR',
 'GR_STGC',
 'GR1',
 'GR-1A',
 'GR2',
 'GR-5A',
 'GRA',
 'GRDA',
 'GRG',
 'GRGS',
 'GRH',
 'GRHD',
 'GRI',
 'GR-I',
 'GRLL',
 'GRML',
 'GR-ML',
 'GRP',
 'GRPD',
 'GRR',
 'GR-R',
 'GRSD',
 'GRSL',
 'GRZ',
 'GRZD',
 'HHGR',
 'HSGR',
 'MGR',
 'P01LGR',
 'P02LGR',
 'P03LGR',
 'RGR',
 'SGR',
 'SGRD',
 'GCPS',
 'GR_WL',
 'GR_LWD',
 'GRM1'],
'PE': ['LPE',
 'PEFL',
 'PEFL',
```

```
 'PEFI',
 'PEFI',
 'HPEF',
 'HPEF',
 'SPEF',
 'PEQH',
 'PEFZ',
 'PEFSA',
 'PEFA',
 'PEF8',
 'PEF_SLDT_HR',
 'PEF_SLDT',
 'PEF',
 'PEDN',
 'PEDF',
 'PE2QH',
 'PE2',
 'PE',
 'PDPE',
 'HPEF8',
 'HPEDN',
 '2PEF',
 'PEFS',
 'PEF_',
 'DPEFM',
 'SNPE'],
'PHID': ['PHID',
 '2DPS',
 'APSC',
 'APSU',
 'DDPP',
 'DEPO',
 'DNPH',
 'DPH8',
 'DPHI',
 'DPHI_',
 'DPHI_SAN',
 'DPHI_SLDT',
 'DPHI_SLDT_HR',
 'DPHIS',
 'DPHS',
 'DPHZ',
 'DPHZSS',
 'DPO',
 'DPOR',
 'DPRS',
 'DPS8',
```

```
 'DPSS',
 'DPSZ',
 'HDPH',
 'HDPH_SAN',
 'HDPH8',
 'HDPHI',
 'HDPO',
 'HDPSS',
 'HFSC',
 'NDPH',
 'PDS',
 'POD',
 'POR',
 'PORS',
 'PORZ',
 'PORZ2QH',
 'PORZQH',
 'PORZS2',
 'PORZS2QH',
 'PORZSS',
 'PORZSS2',
 'PORZSSQH',
 'PXND',
 'PZCSS',
 'PZCSSQH',
 'PZCSSS',
 'PZSS',
 'PZSS2',
 'PZSS2QH',
 'PZSSQH',
 'PZSSS',
 'RPSS',
 'SSD',
 'DPH',
 'DPS1'],
'PHIN': ['PHIN',
 'NEUT',
 'NPHI_S',
 'HNPH',
 'TNPH_SAN1',
 'TNPH_SAN',
 'POS',
 'NPSS_C',
 'NPSS',
 'NPRS',
 'NPORSS',
 'NPOR_SAN',
```

```
 'NPHS',
 'NPHIS',
 'NPHI_SAN',
 'HTNP_SAN',
 'HNPSS',
 'HNPS',
 'HNPO_SAN',
 'HASC',
 'CNSS',
 'CNCSSQH',
 'CNCSS',
 'CNCSGS',
 'CNCS',
 'CN',
 'CNC',
 'FNPS',
 'FPSC',
 'HHNPO',
 'HNPHI',
 'HNPO',
 'HTNP',
 'NCN',
 'NCNL',
 'NPHI',
 'NPOR',
 'NPOR_',
 'NPOR_1',
 'PRON',
 'QNP-1A',
 'QNP-5A',
 'RPOR',
 'SNP',
 'TNPH',
 'ENPH',
 'SPLE'],
'PHIS': ['ACPD',
 'ACPL',
 'ACPS',
 'PORA',
 'RPHI',
 'SPHI',
 'PHIS',
 'SPSS',
 'TPHI'],
'POTA': ['KCPS', 'HPOT', 'POTA', 'K'],
'RESD': ['RESD',
 'RDEP',
```

```
'RT_H',
'RLLD',
'HATR',
'HAT9',
'ASTR',
'AST9',
'AS90',
'AIT9',
'AHTR',
'AHT9',
'ID40',
'VRSD',
'VP65',
'RT_HRLT',
'RT',
'RLA3',
'M4RX',
'M1RX',
'MORX',
'ILD2',
'IDVR',
'IDPH',
'HRLD',
'FE2',
'DVR2',
'DIPH',
'ATRT',
'AT90',
'AST90',
'AIT90',
'AIT120',
'AHTRT',
'A22H',
'A22H_',
'A22H_UNC',
'A28H',
'A28H_',
'A28H_UNC',
'A34H',
'A34H_',
'A34H_UNC',
'AF60',
'AF90',
'AFRT',
'AHF60',
'AHF90',
'AHO60',
```

```
'AHO90',
'AHT60',
'AHT90',
'AILD',
'AIT',
'AIT60',
'AO12',
'AO60',
'AO90',
'AORT',
'ASF60',
'ASF90',
'ASO60',
'ASO90',
'AST60',
'AT60',
'DVR1',
'DVR4',
'GRD',
'HILD',
'HLLD',
'HRL4',
'HRL5',
'IDER',
'IL',
'ILD',
'ILD4',
'LL',
'LL3',
'LL7',
'LLD',
'LN',
'M0R6',
'M0R9',
'M1R6',
'M1R9',
'M2R6',
'M2R9',
'M2RX',
'M4R6',
'M4R9',
'P10H',
'P10H_',
'P10H_UNC',
'P16H',
'P16H_',
'P16H_UNC',
```

```
 'P22H',
 'P22H_',
 'P22H_UNC',
 'P28H',
 'P28H_',
 'P28H_UNC',
 'P34H',
 'P34H_',
 'P34H_UNC',
 'P40H',
 'RFOC',
 'RILD',
 'RIPD',
 'RLA1',
 'RLA2',
 'RLA4',
 'RLA5',
 'RLL3',
 'RLL4',
 'RLL5',
 'VILD',
 'AIT6',
 'DDLL',
 'EIRD',
 'ID10',
 'ID20',
 'ID25',
 'ILD1',
 'LLG',
 'REID',
 'RLL',
 'AHF6',
 'AHF9',
 'AHO6',
 'AHO9',
 'AHOR',
 'AHT6',
 'AS60',
 'ASF6',
 'ASF9',
 'ASFR',
 'ASRT',
 'AST6',
 'RLA5',
 'HAT6'],
'RESM': ['RESM',
 'RMED',
```

```
'SEWP',
'RLLS',
'HAT3',
'AST3',
'AS30',
'AHT3',
'VP25',
'RIPM',
'RILM',
'MVR2',
'M4R3',
'M2R3',
'M1R3',
'M0R3',
'IMVR',
'IMPH',
'IMER',
'IM',
'ILM2',
'ILM',
'HRL3',
'HLLS',
'HILM',
'AT30',
'AST30',
'AIT30',
'AHT30',
'AF30',
'AHF30',
'AHO30',
'AO30',
'ASF30',
'ASO30',
'ILM4',
'LLS',
'MVR1',
'MVR4',
'VILM',
'AE60',
'ILD25',
'ILM1',
'ILM25',
'IM10',
'IM20',
'IM25',
'IM40',
'PIRM',
```

```
 'REIM',
 'DSLL',
 'AIT3',
 'AHF3',
 'AHO3',
 'ASF3',
 'RLA4',
 'RLA3',
 'RLA2',
 'RMLL'],
'RESS': ['RESS',
 'RSHAL',
 'RXOH',
 'MSFF',
 'HSFL',
 'HAT1',
 'AST1',
 'AS10',
 'AHT1',
 'RXOZ',
 'RXOI',
 'RXO8',
 'RXO1D',
 'RXO_HRLT',
 'RXO',
 'RLL1',
 'RESS',
 'HRL1',
 'ATRX',
 'AHTRX',
 'AF10',
 'AF20',
 'AFRX',
 'AHF10',
 'AHF20',
 'AHO10',
 'AHO20',
 'AHSFI',
 'AHT10',
 'AHT20',
 'AIT10',
 'AIT20',
 'AO10',
 'AO20',
 'AORX',
 'ASF10',
 'ASF20',
```

```
'ASFI',
'ASFL',
'ASN',
'ASO10',
'ASO20',
'AST10',
'AST20',
'AT10',
'AT12',
'AT20',
'FE1',
'FEFE',
'FR',
'FRA',
'HRL2',
'HRLS',
'HSFLU',
'ILS',
'LL8',
'LLA',
'M0R1',
'M0R2',
'M1R1',
'M1R2',
'M2R1',
'M2R2',
'M4R1',
'M4R2',
'MFR',
'MLL',
'MSFL',
'PSHG',
'RES',
'RILS',
'RLL2',
'RS',
'RSFE',
'RSFL',
'RSG',
'SFBC',
'SFL',
'SFL4',
'SFLA',
'SFLR',
'SFLU',
'SN',
'AE20',
```

```
 'AE30',
 'AHF1',
 'AHF2',
 'AHO1',
 'AHO2',
 'AHT2',
 'AS20',
 'ASF1',
 'ASF2',
 'ASRX',
 'AST2',
 'HAT2'],
'SP': ['SP_S',
 'AHSF',
 'SPSB',
 'SPH',
 'SPDH',
 'SP',
 'HSP',
 'PCH1',
 'SPBR',
 'SPC',
 'SPCG',
 'SPD',
 'SPDF',
 'SPP',
 'SSPK',
 'SPS',
 'AHSC',
 'SPA_'],
'THOR': ['TCPS', 'THOR', 'HTHO', 'TH'],
'URAN': ['UZ', 'UCPS', 'URAN', 'U', 'HURA']}
```

[14]:
```
j = Project.from_las("1044753454.las")
```

```
0it [00:00, ?it/s]
```

[16]:
```
df_alias2 = j.df(keys=['PE', 'RESD', 'PHIN', 'PHID'], alias=foo)
df_alias2 = df_alias2.dropna(thresh=7) # dropping all rows with 7 NaN's, just␣
 ↪for this demo
df_alias2.head(10)
```

```
      ␣
↪---------------------------------------------------------------------------

       ValueError                                Traceback (most recent call␣
↪last)
```

```
      <ipython-input-16-d4081a17f860> in <module>()
   ----> 1 df_alias2 = j.df(keys=['PE', 'RESD', 'PHIN', 'PHID'], alias=foo)
         2 df_alias2 = df_alias2.dropna(thresh=7) # dropping all rows with 7␣
↪NaN's, just for this demo
         3 df_alias2.head(10)


      /usr/local/lib/python3.6/dist-packages/welly/project.py in df(self,␣
↪keys, basis, alias, rename_aliased)
      609             dfs.append(df)
      610
   --> 611         return pd.concat(dfs)
      612
      613     def data_as_matrix(self, X_keys,


      /usr/local/lib/python3.6/dist-packages/pandas/core/reshape/concat.py in␣
↪concat(objs, axis, join, ignore_index, keys, levels, names, verify_integrity,␣
↪sort, copy)
      282         verify_integrity=verify_integrity,
      283         copy=copy,
   --> 284         sort=sort,
      285     )
      286


      /usr/local/lib/python3.6/dist-packages/pandas/core/reshape/concat.py in␣
↪__init__(self, objs, axis, join, keys, levels, names, ignore_index,␣
↪verify_integrity, copy, sort)
      329
      330         if len(objs) == 0:
   --> 331             raise ValueError("No objects to concatenate")
      332
      333         if keys is None:


      ValueError: No objects to concatenate
```

[22]:
```
!apt-get install texlive texlive-xetex texlive-latex-extra pandoc
!pip install pypandoc
!jupyter nbconvert --to pdf /content/drive/My Drive/OVV_Learn_welly_lasio/
 ↪welly_test.ipynb
```

```
Reading package lists... Done
Building dependency tree
```

```
Reading state information... Done
pandoc is already the newest version (1.19.2.4~dfsg-1build4).
texlive is already the newest version (2017.20180305-1).
texlive-latex-extra is already the newest version (2017.20180305-2).
texlive-xetex is already the newest version (2017.20180305-1).
0 upgraded, 0 newly installed, 0 to remove and 14 not upgraded.
Requirement already satisfied: pypandoc in /usr/local/lib/python3.6/dist-
packages (1.5)
Requirement already satisfied: pip>=8.1.0 in /usr/local/lib/python3.6/dist-
packages (from pypandoc) (19.3.1)
Requirement already satisfied: wheel>=0.25.0 in /usr/local/lib/python3.6/dist-
packages (from pypandoc) (0.36.1)
Requirement already satisfied: setuptools in /usr/local/lib/python3.6/dist-
packages (from pypandoc) (50.3.2)
[NbConvertApp] WARNING | pattern u'/content/drive/My' matched no files
[NbConvertApp] WARNING | pattern u'Drive/OVV_Learn_welly_lasio/welly_test.ipynb'
matched no files
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 information
on full configurables, see '--help-all'.

--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 abort
conversion). This flag is only relevant if '--execute' was specified, too.
--no-input
    Exclude input cells and output prompts from converted document.
    This mode is ideal for generating code-free reports.
--stdout
    Write notebook output to stdout instead of files.
--stdin
    read a single notebook file from stdin. Write the resulting notebook with
default basename 'notebook.*'
--inplace
    Run nbconvert in place, overwriting the existing notebook (only
    relevant when converting to notebook format)
-y
    Answer yes to any questions instead of prompting.
```

```
--clear-output
    Clear output of current file and save in place,
    overwriting the existing notebook.
--debug
    set log level to logging.DEBUG (maximize logging output)
--no-prompt
    Exclude input and output prompts from converted document.
--generate-config
    generate default config file
--nbformat=<Enum> (NotebookExporter.nbformat_version)
    Default: 4
    Choices: [1, 2, 3, 4]
    The nbformat version to write. Use this to downgrade notebooks.
--output-dir=<Unicode> (FilesWriter.build_directory)
    Default: ''
    Directory to write output(s) to. Defaults to output to the directory of each
    notebook. To recover previous default behaviour (outputting to the current
    working directory) use . as the flag value.
--writer=<DottedObjectName> (NbConvertApp.writer_class)
    Default: 'FilesWriter'
    Writer class used to write the  results of the conversion
--log-level=<Enum> (Application.log_level)
    Default: 30
    Choices: (0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR',
'CRITICAL')
    Set the log level by value or name.
--reveal-prefix=<Unicode> (SlidesExporter.reveal_url_prefix)
    Default: u''
    The URL prefix for reveal.js (version 3.x). This defaults to the reveal CDN,
    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.
--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 represents
    the import path for an `Exporter` class
--template=<Unicode> (TemplateExporter.template_file)
    Default: u''
    Name of the template file to use
--output=<Unicode> (NbConvertApp.output_base)
    Default: ''
```

```
    overwrite base name use for output files. can only be used when converting
    one notebook at a time.
--post=<DottedOrNone> (NbConvertApp.postprocessor_class)
    Default: u''
    PostProcessor class used to write the results of the conversion
--config=<Unicode> (JupyterApp.config_file)
    Default: u''
    Full path of a config file.

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 HTML).

    You can specify the export format with `--to`.
    Options include ['asciidoc', 'custom', 'html', 'latex', 'markdown',
'notebook', '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
```

[21]:
```
pwd
```

[21]:
```
'/content/drive/My Drive/OVV_Learn_welly_lasio'
```

[ ]:
```

```