In [493]:

```
#Copy and Paste Code From Postman


import requests

url = "https://api.peloton.com/v1/pceusintern/wellview/data/wvJobReportCost

payload  = {}
headers = {
  'Content-Type': 'application/json',
  'wellview': 'dwB2AHwAMQAwAC4AMAAwAHwAMQAwAC4ANAAwAHwAaQBkAHcAZQBsAGwAfAB3
  'Ocp-Apim-Subscription-Key': 'd7abf37b71b14254b47b539f8631d905',
  'Authorization': 'Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6Ijc4NTFGNDMwQjcxQjZE
  'Cookie': '_ga=GA1.2.774963620.1594670862; hostregion=usw01; login_pref=s
}

response = requests.request("GET", url, headers=headers, data = payload)

print(response.text.encode('utf8'))
```

```
user":"amna.yasin@peloton.com"},{"idwell":"2E9D4FCDE9B24839B0FF8FC4C32BF3
B5","idrecparent":"1F2E24D33A1544B8971C34B653423B72","syscarryfwdp":0,"id
rec":"00CE5891C0F34B6AAEBD1A4DC3648F6A","code1":"4400","code2":"6301","co
de3":"COMP","cost":415.0,"des":"Miscellaneous well services","userboolean
1":0,"vendor":"United","sysseq":20,"syslockdate":"2020-06-02T17:52:39.14
3","sysmoddate":"2020-07-21T16:35:20.333","sysmoduser":"amna.yasin@peloto
n.com","syscreatedate":"2020-06-02T17:52:39.143","syscreateuser":"amna.ya
sin@peloton.com"},{"idwell":"2E9D4FCDE9B24839B0FF8FC4C32BF3B5","idrecpare
nt":"5BB3C0CEEFCC42E4B9866981510E6A73","syscarryfwdp":0,"idrec":"0135CADD
9ABD485FBAD804EEFF547214","code1":"2400","code2":"2410","code3":"DRLG","c
ost":569.0,"des":"Miscellaneous well services","userboolean1":0,"vendo
r":"EXPRO","sysseq":5,"syslockdate":"2020-06-02T17:52:39.143","sysmoddat
e":"2020-06-02T17:52:39.143","sysmoduser":"amna.yasin@peloton.com","syscr
eatedate":"2020-06-02T17:52:39.143","syscreateuser":"amna.yasin@peloton.c
om"},{"idwell":"2E9D4FCDE9B24839B0FF8FC4C32BF3B5","idrecparent":"2D96E687
AD0D44C2B4BF71581194BEE7","syscarryfwdp":0,"idrec":"014F2805F8614C2289F7F
F9778C6CA52","code1":"1500","code2":"1580","code3":"DRLG","cost":40329.
0,"des":"Work/Supply Boats","userboolean1":0,"vendor":"Swire","sysseq":1
2,"syslockdate":"2020-06-02T17:52:39.143","sysmoddate":"2020-06-02T17:52:
39.143","sysmoduser":"amna.yasin@peloton.com","syscreatedate":"2020-06-02
```

In [494]:

```
Daily_costs = response.text
```

In [495]:

```
Daily_costs
```

```
2:39.143","syscreateuser":"amna.yasin@peloton.com"},{"idwell":"2E9D4FCDE9
B24839B0FF8FC4C32BF3B5","idrecparent":"002FC2B44398403586A3CB550DD16F3
4","syscarryfwdp":0,"idrec":"11322FAF63ED4B81828718FEAB0D2733","code1":"4
400","code2":"6301","code3":"COMP","cost":200.0,"des":"Miscellaneous well
services","userboolean1":0,"vendor":"EXPRO","sysseq":29,"syslockdate":"20
20-06-02T17:52:39.143","sysmoddate":"2020-07-29T20:43:56.833","sysmoduse
r":"amna.yasin@peloton.com","syscreatedate":"2020-06-02T17:52:39.143","sy
screateuser":"amna.yasin@peloton.com"},{"idwell":"2E9D4FCDE9B24839B0FF8FC

4C32BF3B5","idrecparent":"F5F19ED53E0749FCAA33263D16373716","syscarryfwd
p":0,"idrec":"11633D56A91642D192C5BDC9D6982BEE","code1":"2000","code2":"2
010","code3":"DRLG","cost":-419557.3125,"des":"Drilling & completion flui
ds","userboolean1":0,"vendor":"Baroid","sysseq":1,"syslockdate":"2020-06-
02T17:52:39.143","sysmoddate":"2020-06-02T17:52:39.143","sysmoduser":"amn
a.yasin@peloton.com","syscreatedate":"2020-06-02T17:52:39.143","syscreate
user":"amna.yasin@peloton.com"},{"idwell":"2E9D4FCDE9B24839B0FF8FC4C32BF3
B5","idrecparent":"F88ECA69DAB644389A348286320FE545","syscarryfwdp":0,"id
rec":"11B90234008A4B708B559E81E19B0383","code1":"4600","code2":"3620","co
de3":"COMP","cost":354.0,"des":"Directional Drilling Services","userboole
an1":0,"vendor":"Anadril","sysseq":24,"syslockdate":"2020-06-02T17:52:39.
```

In [496]:

```
parsed = json.loads(Daily_costs)
parsed
```

Out[496]:

```
[{'idwell': '2E9D4FCDE9B24839B0FF8FC4C32BF3B5',
  'idrecparent': '0B4B000DF56944C6B410D389E467AF2E',
  'syscarryfwdp': 0,
  'idrec': '004672EB0BE14781B04E9549CC18A0F7',
  'code1': '5400',
  'code2': '3095',
  'code3': 'COMP',
  'cost': 650.0,
  'des': 'Casing/Tubing Crew and Tools',
  'userboolean1': 0,
  'vendor': 'Weatherford',
  'sysseq': 17,
  'syslockdate': '2020-06-02T17:52:39.143',
  'sysmoddate': '2020-07-21T17:55:54.917',
  'sysmoduser': 'amna.yasin@peloton.com',
  'syscreatedate': '2020-06-02T17:52:39.143',
  'syscreateuser': 'amna.yasin@peloton.com'},
 {'idwell': '2E9D4FCDE9B24839B0FF8FC4C32BF3B5',
  'idrecparent': 'EF32394CFBD24408A70C51FC0A762172',
```

In [497]:
```
#Turn Data into a Data Frame
df_costs = pd.read_json(Daily_costs)
df_costs
```

Out[497]:

	code1	code2	code3	cost	des	idrec
0	5400	3095	COMP	650.000000	Casing/Tubing Crew and Tools	004672EB0BE14781B04E9549CC18A0F7
1	4600	3620	COMP	5945.000000	Directional Drilling Services	006146AB965D46C0BF3986C8E6FD4E3B
2	2700	2780	DRLG	500.000000	Subsea wellhead equipment	007397B33AE64EFCB75BA2D4174F19A8
3	1500	1580	DRLG	40329.000000	Work/Supply Boats	00C3C64F877542A7985653FF95074455
4	4400	6301	COMP	415.000000	Miscellaneous well services	00CE5891C0F34B6AAEBD1A4DC3648F6A
5	2400	2410	DRLG	569.000000	Miscellaneous well services	0135CADD9ABD485FBAD804EEFF547214

In [498]:
```
df_costs.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1248 entries, 0 to 1247
Data columns (total 20 columns):
code1                     1248 non-null int64
code2                     1248 non-null int64
code3                     1248 non-null object
cost                      1248 non-null float64
des                       1248 non-null object
idrec                     1248 non-null object
idrecparent               1248 non-null object
idrecphasecustom          1 non-null object
idrecphasecustomtk        1 non-null object
idwell                    1248 non-null object
note                      1 non-null object
syscarryfwdp              1248 non-null int64
syscreatedate             1248 non-null object
syscreateuser             1248 non-null object
syslockdate               1248 non-null object
sysmoddate                1248 non-null object
sysmoduser                1248 non-null object
sysseq                    1248 non-null int64
userboolean1              1248 non-null int64
vendor                    1247 non-null object
dtypes: float64(1), int64(5), object(14)
memory usage: 195.1+ KB
```

In [499]:

```
df_costs.max()

#The Cost For This Specific Day that was the most Expensive was Work/Supply
```

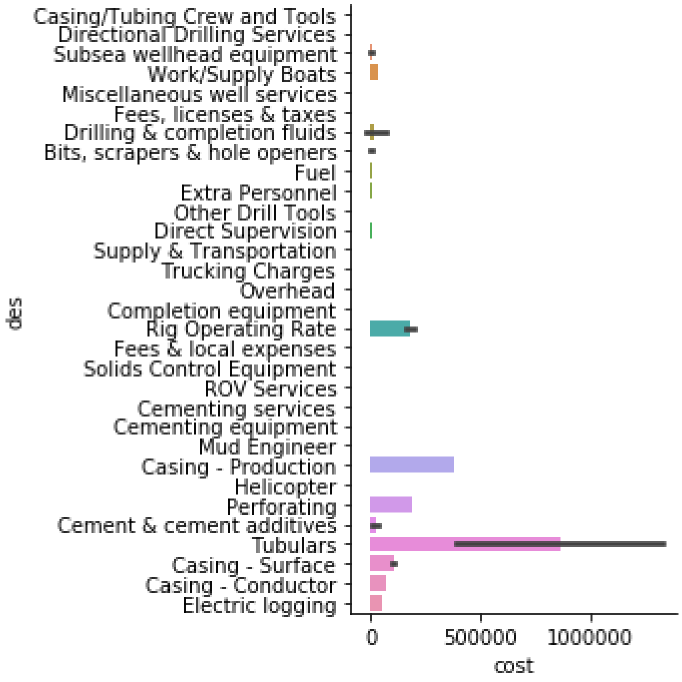
Out[499]:

```
code1                                              8200
code2                                              7602
code3                                              DRLG
cost                                        1.32530e+06
des                                  Work/Supply Boats
idrec                  FF73073396724D078483CB16C881FA02
idrecparent            F88ECA69DAB644389A348286320FE545
idwell                 2E9D4FCDE9B24839B0FF8FC4C32BF3B5
syscarryfwdp                                          0
syscreatedate                   2020-06-02T17:52:39.143
syscreateuser                     amna.yasin@peloton.com
syslockdate                     2020-06-02T17:52:39.143
sysmoddate                      2020-07-29T20:44:03.863
sysmoduser                        amna.yasin@peloton.com
sysseq                                            32767
userboolean1                                          0
dtype: object
```

In [ ]:

In [500]:

```
import seaborn as sns
sns.catplot(x="cost", y="des",kind='bar',data=df_costs);
#The most expensive equipment overall is Tubulars
```



```
In [501]: df_costs['vendor']
```

```
Out[501]:
0                    Weatherford
1                        Anadril
2                          Sedco
3                          Swire
4                         United
5                          EXPRO
6                          Swire
7                        Peloton
8                         Baroid
9                    GeoServices
10                        United
11                        Tasman
12                          Esso
13                   Weatherford
14                         Sedco
15                       Peloton
16                        Tasman
17                   Weatherford
18                       Peloton
19                  Schlumberger
20                         EXPRO
21                        Tasman
22                   Weatherford
23                        Tasman
24                   GeoServices
25                          EUDC
26                    Scientific
27              Triple H Rentals
28                        Wellco
29                        Tasman
                  ...
1218                 Weatherford
1219                       Vetco
1220                 Halliburton
1221                      United
1222                      United
1223                 GeoServices
1224    Mac's Oilfield Equipment
1225                     Peloton
1226                      Tasman
1227                   Oil Tools
1228                 Halliburton
1229                     Peloton
1230                 GeoServices
1231                       Swire
1232                      Tasman
1233                     Peloton
1234                      Tasman
1235                 Weatherford
1236                      Baroid
1237            Triple H Rentals
1238                      Baroid
1239                      Tasman
1240                     Peloton
1241                 Halliburton
```

```
1242                      Baroid
1243                        Esso
1244                   Oil Tools
1245                      Tasman
1246                 Weatherford
1247                        EUDC
Name: vendor, Length: 1248, dtype: object
```

```
In [502]: df_costs['cost']
```

```
Out[502]: 0          650.000000
          1         5945.000000
          2          500.000000
          3        40329.000000
          4          415.000000
          5          569.000000
          6        40329.000000
          7         1250.000000
          8        16836.128906
          9          120.000000
          10         758.000000
          11         205.000000
          12       30150.000000
          13         650.000000
          14         500.000000
          15        2188.000000
          16         410.000000
          17         650.000000
          18        5250.000000
          19        9600.000000
          20         200.000000
          21          33.000000
          22        1460.000000
          23          33.000000
          24         120.000000
          25       12545.000000
          26         886.000000
          27        2200.000000
          28        2200.000000
          29         410.000000
                       ...
          1218      1460.000000
          1219      1818.000000
          1220       778.000000
          1221       758.000000
          1222       415.000000
          1223       120.000000
          1224      2200.000000
          1225      5250.000000
          1226      1004.000000
          1227       640.000000
          1228       778.000000
          1229       250.000000
          1230       120.000000
          1231     40329.000000
          1232       205.000000
          1233      1250.000000
          1234     18000.000000
          1235       650.000000
          1236      1300.000000
          1237      2200.000000
          1238      1036.800049
          1239      1004.000000
          1240      5250.000000
          1241        15.000000
```

```
1242     1300.000000
1243     4200.000000
1244      640.000000
1245     1004.000000
1246     1460.000000
1247    12545.000000
Name: cost, Length: 1248, dtype: float64
```

In [504]:

```python
import requests

url = "https://api.peloton.com/v1/pceusintern/wellview/data/wvjob/entityid/

payload  = {}
headers = {
  'Content-Type': 'application/json',
  'wellview': 'dwB2AHwAMQAwAC4AMAAwAHwAMQAwAC4ANAAwAHwAaQBkAHcAZQBsAGwAfAB3
  'Ocp-Apim-Subscription-Key': 'd7abf37b71b14254b47b539f8631d905',
  'Authorization': 'Bearer eyJhbGciOiJSUzI1NiIsImtpZCI6Ijc4NTFGNDMwQjcxQjZE
  'Cookie': '_ga=GA1.2.774963620.1594670862; hostregion=usw01; login_pref=s
}

response = requests.request("GET", url, headers=headers, data = payload)

print(response.text.encode('utf8'))
```

```
b'[{"idwell":"2E9D4FCDE9B24839B0FF8FC4C32BF3B5","idrec":"8C9F000821C44E32
92F4FB01F2C939F3","afeamtcalc":17945777.0,"afeamtnormcalc":17945777.0,"af
enumbercalc":"9876543","afecosttypcalc":"9876543-Capital","afeperdurmlcal
c":612568.71186440683,"afeperdurmlnormcalc":612568.71186440683,"afepertar
getdepthcalc":5155.4197340321462,"afepertargetdepthnormcalc":5155.4197340
321462,"afesupamtcalc":125000.0,"afesupamtnormcalc":125000.0,"afetotalcal
c":18070777.0,"afetotalnormcalc":18070777.0,"bhadrillruncalc":10.0,"bhato
talruncalc":10.0,"bitrevscalc":2191725.0119492412,"costafeforecastvarcal
c":18070777.0,"costfinalactual":10255618.0,"costforecastfieldvarcalc":-10
127291.467214966,"costmaxtotalcalc":28050000.0,"costmintotalcalc":2490090
0.0,"costmltotalcalc":26675000.0,"costmltotalnoplanchangecalc":26675000.
0,"costmlnoexcludecalc":26675000.0,"costnormafeforecastvarcalc":18070777.
0,"costnormforecastfieldvarcalc":-10127291.467214966,"costnormperdepthcal
c":2987.4016127477776,"costnormtotalcalc":10127291.467214966,"costperdept
hcalc":2987.4016127477776,"costperdepthplanmlcalc":6864.0317881626788,"co
stnormperdepthplanmlcalc":6864.0317881626788,"costtechlimittotalcalc":0.
0,"costpertldurcalc":383367.49852060893,"costpertldurnormcalc":383367.498
52060893,"costtotalcalc":10127291.467214966,"depthdrilledcalc":3390.0,"de
pthdrilledperbhacalc":339.0,"depthdrilledspudtorrcalc":129.9161676646706
7,"depthdrilledperreportnocalc":121.07142857142857,"depthperdurplanmlcal
c":131.7355915651483,"depthperratiodurationcalc":3849.8422530607654,"dept
hplanmaxcalc":3886.199951171875,"depthrotatingcalc":3178.712158203125,"de
pthslidingcalc":211.287841796875,"dttmend":"2001-03-18T02:15:00","dttmend
planmaxcalc":"2000-03-25T12:00:00","dttmendplanmincalc":"2000-03-13T00:0
0:00","dttmendplanmlcalc":"2000-03-20T12:00:00","dttmspud":"2000-02-21T0
0:00:00","dttmstart":"2000-02-21T00:00:00","dttmstartplan":"2000-02-20T0
0:00:00","dttmtotaldepthcalc":"2000-03-15T00:00:00","durationmaxtotalcal
c":34.5,"durationmintotalcalc":22.0,"durationmltotalcalc":29.5,"durationn
oproblemtimecalc":25.937500119023024,"durationproblemtimecalc":0.47916667
275130748,"durationspudtoplanmlcalc":28.5,"durationspudtoplanmaxcalc":33.
5,"durationspudtoplanmincalc":21.0,"durationspudtimelogcalc":22.229166666
666664,"durationspudtotdcalc":23.0,"durationspudtorrcalc":26.09375,"durat
iontechlimittotalcalc":0.0,"durationtimelogtotalcalc":26.41666679177433
3,"durmltotalnoplanchangecalc":29.5,"durmlnoexcludecalc":29.5,"estcostnor
msavecalc":625.0,"estcostsavecalc":625.0,"estproblemcostcalc":38500.0,"es
tproblemcostnormcalc":38500.0,"estproblemtimecalc":0.47916667722165585,"e
sttimesavecalc":0.3333333432674408,"finalinvoicetotalcalc":10217000.0,"fi
nalinvoicetotalnormcalc":10217000.0,"idreclastrigcalc":"DAEF596E35524191B
B9FFBE84A66C49E","idreclastrigcalctk":"wvjobrig","idrecwellbore":"EE9B537
```

```
CAAC643F7A8BBB6DE31C5F341","idrecwellboretk":"wvwellbore","idrecwellborec
alc":"E9FF61F2537A4DE393E9F37C8C5BAE0E","idrecwellborecalctk":"wvwellbor
e","jobsupplycostcalc":0.0,"jobsupplycostnormcalc":0.0,"jobtyp":"Drilling
- original","mudcostcalc":178846.14833641052,"mudcostnormcalc":178846.148
33641052,"mudcostperdepthcalc":52.756975910445583,"mudcostperdepthnormcal
c":52.756975910445583,"muddensitymaxcalc":1342.0560302734375,"muddensitym
incalc":1018.524658203125,"mudtypcalc":"PETROFREE","objective":"To drill,
case, and cement the well into the Lower Zone and hand well over to compl
etion.","pctproblemtimecalc":0.018138801406258842,"percenttmrotatingcal
c":0.92535402237699194,"percenttmslidingcalc":0.074645977623008092,"perce
ntdepthrotatingcalc":0.93767320300977142,"percentdepthslidingcalc":0.0623
2679699022861 1,"programmuddensitymaxcalc":1318.0906982421875,"programmudd
ensitymincalc":1078.4378662109375,"projectrefnumbercalc":"abc123","ratiod
epthactualplancalc":0.9806494899601863,"ratiodepthactualtargetcalc":1.087
2418273102762,"ratiodurtimelogrefhourscalc":0.88055555972581112,"reportno
calc":28.0,"ropavgfromspudcalc":147.39130434782609,"ropcalc":367.70111342
093014,"roprotatingcalc":372.59629550861456,"ropslidingcalc":307.01765023
442397,"ropspudtimelogcalc":152.50234301780697,"roptimelogcalc":128.32807
510202551,"safetyincnocalc":5.0,"safetyincreportnocalc":0.0,"status1":"Jo
b Complete","summary":"No major problems were encountered while drilling
this well.  Note that the well was completed under budget and within the
allocated number of days.","targetdepth":3505.199951171875,"targetfor
m":"Blue Heron Shale","tmcirccalc":0.0,"tmdrillcalc":9.219444478861987
6,"tmothercalc":0.0,"tmrotatingcalc":8.53125003259629,"tmslidingcalc":0.6
8819444626569748,"tmtripcalc":0.0,"totaldepthcalc":3811.0,"totaldepthtvdc
alc":2909.7178717625857,"tdtomudcalc":3658.5999908447266,"userboolean1":
0,"userboolean2":0,"varianceafefinalcalc":7853777.0,"variancefieldcalc":7
943485.5327850338,"variancefieldfinalcalc":-89708.5327850338,"variancefin
alcalc":7815159.0,"variancenormafefinalcalc":7853777.0,"variancenormfield
calc":7943485.5327850338,"variancenormfieldfinalcalc":-89708.532785033
8,"variancenormfinalcalc":7815159.0,"wvtyp":"Drilling","syslockdate":"202
0-06-02T17:52:39.143","sysmoddate":"2020-06-02T17:52:39.143","sysmoduse
r":"amna.yasin@peloton.com","syscreatedate":"2020-06-02T17:52:39.143","sy
screateuser":"amna.yasin@peloton.com"},{"idwell":"2E9D4FCDE9B24839B0FF8FC
4C32BF3B5","idrec":"FAF36292F63A4BD0BA67BAAF015F77A1","afeamtcalc":775300
0.0,"afeamtnormcalc":7753000.0,"afenumbercalc":"1234567C","afecosttypcal
c":"1234567C-Capital","afepertargetdepthcalc":2211.85670090175,"afepertar
getdepthnormcalc":2211.85670090175,"afetotalcalc":7753000.0,"afetotalnorm
calc":7753000.0,"bhadrillruncalc":0.0,"bhatotalruncalc":0.0,"costafeforec
astvarcalc":7753000.0,"costfinalactual":7816255.0,"costforecastfieldvarca
lc":-7672928.4710083008,"costmaxtotalcalc":0.0,"costmintotalcalc":0.0,"co
stmltotalcalc":0.0,"costmlnoexcludecalc":0.0,"costnormafeforecastvarcal
c":7753000.0,"costnormforecastfieldvarcalc":-7672928.4710083008,"costnorm
totalcalc":7672928.4710083008,"costtechlimittotalcalc":0.0,"costpertldurc
alc":443201.64393872686,"costpertldurnormcalc":443201.64393872686,"costto
talcalc":7672928.4710083008,"depthperratiodurationcalc":0.0,"dttmend":"20
00-06-03T00:00:00","dttmstart":"2000-05-16T00:00:00","durationmaxtotalcal
c":0.0,"durationmintotalcalc":0.0,"durationmltotalcalc":0.0,"durationnopr
oblemtimecalc":17.312500023283064,"durationspudtorrcalc":102.0,"durationt
echlimittotalcalc":0.0,"durationtimelogtotalcalc":17.312500023283064,"dur
mlnoexcludecalc":0.0,"idreclastrigcalc":"389FB7BC4D53441E9533C4845733918
E","idreclastrigcalctk":"wvjobrig","idrecwellbore":"E9FF61F2537A4DE393E9F
37C8C5BAE0E","idrecwellboretk":"wvwellbore","idrecwellborecalc":"E9FF61F2
537A4DE393E9F37C8C5BAE0E","idrecwellborecalctk":"wvwellbore","jobsubty
p":"Initial Completion","jobsupplycostcalc":0.0,"jobsupplycostnormcalc":
0.0,"jobtyp":"Completion","mudcostcalc":0.0,"mudcostnormcalc":0.0,"mudden
sitymaxcalc":1066.4552001953125,"muddensitymincalc":1018.524658203125,"mu
```

```
dtypcalc":"BRINE","projectrefnumbercalc":"abc123","ratiodurtimelogrefhour
scalc":0.57708333410943546,"reportnocalc":18.0,"safetyincnocalc":1.0,"saf
etyincreportnocalc":0.0,"targetdepth":3505.199951171875,"userboolean1":
0,"userboolean2":0,"varianceafefinalcalc":7753000.0,"variancefieldcalc":8
0071.528991699219,"variancefieldfinalcalc":7672928.4710083008,"variancefi
nalcalc":-63255.0,"variancenormafefinalcalc":7753000.0,"variancenormfield
calc":80071.528991699219,"variancenormfieldfinalcalc":7672928.471008300
8,"variancenormfinalcalc":-63255.0,"wvtyp":"Completion/Workover","syslock
meui":0,"syslockdate":"2020-06-02T17:52:39.143","sysmoddate":"2020-07-21T
16:27:51.683","sysmoduser":"amna.yasin@peloton.com","syscreatedate":"2020
-06-02T17:52:39.143","syscreateuser":"amna.yasin@peloton.com"}]'
```

In [505]:

```
import json
import pandas as pd
Drilling_Completions = response.text

Drilling_Completions
```

Out[505]:

```
'[{"idwell":"2E9D4FCDE9B24839B0FF8FC4C32BF3B5","idrec":"8C9F000821C44E329
2F4FB01F2C939F3","afeamtcalc":17945777.0,"afeamtnormcalc":17945777.0,"afe
numbercalc":"9876543","afecosttypcalc":"9876543-Capital","afeperdurmlcal
c":612568.71186440683,"afeperdurmlnormcalc":612568.71186440683,"afepertar
getdepthcalc":5155.4197340321462,"afepertargetdepthnormcalc":5155.4197340
321462,"afesupamtcalc":125000.0,"afesupamtnormcalc":125000.0,"afetotalcal
c":18070777.0,"afetotalnormcalc":18070777.0,"bhadrillruncalc":10.0,"bhato
talruncalc":10.0,"bitrevscalc":2191725.0119492412,"costafeforecastvarcal
c":18070777.0,"costfinalactual":10255618.0,"costforecastfieldvarcalc":-10
127291.467214966,"costmaxtotalcalc":28050000.0,"costmintotalcalc":2490090
0.0,"costmltotalcalc":26675000.0,"costmltotalnoplanchangecalc":26675000.
0,"costmlnoexcludecalc":26675000.0,"costnormafeforecastvarcalc":18070777.
0,"costnormforecastfieldvarcalc":-10127291.467214966,"costnormperdepthcal
c":2987.4016127477776,"costnormtotalcalc":10127291.467214966,"costperdept
hcalc":2987.4016127477776,"costperdepthplanmlcalc":6864.0317881626788,"co
stnormperdepthplanmlcalc":6864.0317881626788,"costtechlimittotalcalc":0.
0,"costpertldurcalc":383367.49852060893,"costpertldurnormcalc":383367.498
52060893,"costtotalcalc":10127291.467214966,"depthdrilledcalc":3390.0,"de
pthdrilledperbhacalc":339.0,"depthdrilledspudtorrcalc":129.9161676646706
7,"depthdrilledperreportnocalc":121.07142857142857,"depthperdurplanmlcal
c":131.7355915651483,"depthperratiodurationcalc":3849.8422530607654,"dept
hplanmaxcalc":3886.199951171875,"depthrotatingcalc":3178.712158203125,"de
pthslidingcalc":211.287841796875,"dttmend":"2001-03-18T02:15:00","dttmend
planmaxcalc":"2000-03-25T12:00:00","dttmendplanmincalc":"2000-03-13T00:0
0:00","dttmendplanmlcalc":"2000-03-20T12:00:00","dttmspud":"2000-02-21T0
0:00:00","dttmstart":"2000-02-21T00:00:00","dttmstartplan":"2000-02-20T0
0:00:00","dttmtotaldepthcalc":"2000-03-15T00:00:00","durationmaxtotalcal
c":34.5,"durationmintotalcalc":22.0,"durationmltotalcalc":29.5,"durationn
oproblemtimecalc":25.937500119023024,"durationproblemtimecalc":0.47916667
275130748,"durationspudtoplanmlcalc":28.5,"durationspudtoplanmaxcalc":33.
5,"durationspudtoplanmincalc":21.0,"durationspudtimelogcalc":22.229166666
666664,"durationspudtotdcalc":23.0,"durationspudtorrcalc":26.09375,"durat
iontechlimittotalcalc":0.0,"durationtimelogtotalcalc":26.41666679177433
3,"durmltotalnoplanchangecalc":29.5,"durmlnoexcludecalc":29.5,"estcostnor
msavecalc":625.0,"estcostsavecalc":625.0,"estproblemcostcalc":38500.0,"es
tproblemcostnormcalc":38500.0,"estproblemtimecalc":0.47916667722165585,"e
sttimesavecalc":0.333333432674408,"finalinvoicetotalcalc":10217000.0,"fi
nalinvoicetotalnormcalc":10217000.0,"idreclastrigcalc":"DAEF596E35524191B
B9FFBE84A66C49E","idreclastrigcalctk":"wvjobrig","idrecwellbore":"EE9B537
CAAC643F7A8BBB6DE31C5F341","idrecwellboretk":"wvwellbore","idrecwellborec
alc":"E9FF61F2537A4DE393E9F37C8C5BAE0E","idrecwellborecalctk":"wvwellbor
e","jobsupplycostcalc":0.0,"jobsupplycostnormcalc":0.0,"jobtyp":"Drilling
- original","mudcostcalc":178846.14833641052,"mudcostnormcalc":178846.148
33641052,"mudcostperdepthcalc":52.756975910445583,"mudcostperdepthnormcal
c":52.756975910445583,"muddensitymaxcalc":1342.0560302734375,"muddensitym
incalc":1018.524658203125,"mudtypcalc":"PETROFREE","objective":"To drill,
case, and cement the well into the Lower Zone and hand well over to compl
etion.","pctproblemtimecalc":0.018138801406258842,"percenttmrotatingcal
c":0.92535402237699194,"percenttmslidingcalc":0.074645977623008092,"perce
ntdepthrotatingcalc":0.93767320300977142,"percentdepthslidingcalc":0.0623
2679699022861l,"programmuddensitymaxcalc":1318.0906982421875,"programmudd
ensitymincalc":1078.4378662109375,"projectrefnumbercalc":"abc123","ratiod
epthactualplancalc":0.9806494899601863,"ratiodepthactualtargetcalc":1.087
2418273102762,"ratiodurtimelogrefhourscalc":0.8805555972581112,"reportno
calc":28.0,"ropavgfromspudcalc":147.39130434782609,"ropcalc":367.70111342
093014,"roprotatingcalc":372.59629550861456,"ropslidingcalc":307.01765023
442397,"ropspudtimelogcalc":152.50234301780697,"roptimelogcalc":128.32807
510202551,"safetyincnocalc":5.0,"safetyincreportnocalc":0.0,"status1":"Jo
b Complete","summary":"No major problems were encountered while drilling
this well.  Note that the well was completed under budget and within the
allocated number of days.","targetdepth":3505.199951171875,"targetfor
m":"Blue Heron Shale","tmcirccalc":0.0,"tmdrillcalc":9.219444478861987
6,"tmothercalc":0.0,"tmrotatingcalc":8.53125003259629,"tmslidingcalc":0.6
881944462656974 8,"tmtripcalc":0.0,"totaldepthcalc":3811.0,"totaldepthtvdc
alc":2909.7178717625857,"tdtomudcalc":3658.5999908447266,"userboolean1":
0,"userboolean2":0,"varianceafefinalcalc":7853777.0,"variancefieldcalc":7
943485.5327850338,"variancefieldfinalcalc":-89708.5327850338,"variancefin
alcalc":7815159.0,"variancenormafefinalcalc":7853777.0,"variancenormfield
calc":7943485.5327850338,"variancenormfieldfinalcalc":-89708.532785033
8,"variancenormfinalcalc":7815159.0,"wvtyp":"Drilling","syslockdate":"202
0-06-02T17:52:39.143","sysmoddate":"2020-06-02T17:52:39.143","sysmoduse
r":"amna.yasin@peloton.com","syscreatedate":"2020-06-02T17:52:39.143","sy
screateuser":"amna.yasin@peloton.com"},{"idwell":"2E9D4FCDE9B24839B0FF8FC
4C32BF3B5","idrec":"FAF36292F63A4BD0BA67BAAF015F77A1","afeamtcalc":775300
0.0,"afeamtnormcalc":7753000.0,"afenumbercalc":"1234567C","afecosttypcal
c":"1234567C-Capital","afepertargetdepthcalc":2211.85670090175,"afepertar
getdepthnormcalc":2211.85670090175,"afetotalcalc":7753000.0,"afetotalnorm
calc":7753000.0,"bhadrillruncalc":0.0,"bhatotalruncalc":0.0,"costafeforec
astvarcalc":7753000.0,"costfinalactual":7816255.0,"costforecastfieldvarca
lc":-7672928.4710083008,"costmaxtotalcalc":0.0,"costmintotalcalc":0.0,"co
stmltotalcalc":0.0,"costmlnoexcludecalc":0.0,"costnormafeforecastvarcal
c":7753000.0,"costnormforecastfieldvarcalc":-7672928.4710083008,"costnorm
totalcalc":7672928.4710083008,"costtechlimittotalcalc":0.0,"costpertldurc
alc":443201.64393872686,"costpertldurnormcalc":443201.64393872686,"costto
talcalc":7672928.4710083008,"depthperratiodurationcalc":0.0,"dttmend":"20
00-06-03T00:00:00","dttmstart":"2000-05-16T00:00:00","durationmaxtotalcal
c":0.0,"durationmintotalcalc":0.0,"durationmltotalcalc":0.0,"durationnopr
oblemtimecalc":17.312500023283064,"durationspudtorrcalc":102.0,"durationt
echlimittotalcalc":0.0,"durationtimelogtotalcalc":17.312500023283064,"dur
mlnoexcludecalc":0.0,"idreclastrigcalc":"389FB7BC4D53441E9533C4845733918
E","idreclastrigcalctk":"wvjobrig","idrecwellbore":"E9FF61F2537A4DE393E9F
37C8C5BAE0E","idrecwellboretk":"wvwellbore","idrecwellborecalc":"E9FF61F2
537A4DE393E9F37C8C5BAE0E","idrecwellborecalctk":"wvwellbore","jobsubty
p":"Initial Completion","jobsupplycostcalc":0.0,"jobsupplycostnormcalc":
0.0,"jobtyp":"Completion","mudcostcalc":0.0,"mudcostnormcalc":0.0,"mudden
sitymaxcalc":1066.4552001953125,"muddensitymincalc":1018.524658203125,"mu
dtypcalc":"BRINE","projectrefnumbercalc":"abc123","ratiodurtimelogrefhour
scalc":0.57708333410943546,"reportnocalc":18.0,"safetyincnocalc":1.0,"saf
etyincreportnocalc":0.0,"targetdepth":3505.199951171875,"userboolean1":
0,"userboolean2":0,"varianceafefinalcalc":7753000.0,"variancefieldcalc":8
0071.528991699219,"variancefieldfinalcalc":7672928.4710083008,"variancefi
nalcalc":-63255.0,"variancenormafefinalcalc":7753000.0,"variancenormfield
calc":80071.528991699219,"variancenormfieldfinalcalc":7672928.471008300
8,"variancenormfinalcalc":-63255.0,"wvtyp":"Completion/Workover","syslock
meui":0,"syslockdate":"2020-06-02T17:52:39.143","sysmoddate":"2020-07-21T
16:27:51.683","sysmoduser":"amna.yasin@peloton.com","syscreatedate":"2020
-06-02T17:52:39.143","syscreateuser":"amna.yasin@peloton.com"}]'
```

In [506]:
```
parsed = json.loads(Drilling_Completions)
parsed
```

```
  'idrecwellborecalctk': 'wvwellbore',
  'jobsubtyp': 'Initial Completion',
  'jobsupplycostcalc': 0.0,
  'jobsupplycostnormcalc': 0.0,
  'jobtyp': 'Completion',
  'mudcostcalc': 0.0,
  'mudcostnormcalc': 0.0,
  'muddensitymaxcalc': 1066.4552001953125,
  'muddensitymincalc': 1018.524658203125,
  'mudtypcalc': 'BRINE',
  'projectrefnumbercalc': 'abc123',
  'ratiodurtimelogrefhourscalc': 0.5770833341094355,
  'reportnocalc': 18.0,
  'safetyincnocalc': 1.0,
  'safetyincreportnocalc': 0.0,
  'targetdepth': 3505.199951171875,
  'userboolean1': 0,
  'userboolean2': 0,
  'varianceafefinalcalc': 7753000.0,
  'variancefieldcalc': 80071.52899169922,
```

In [507]:
```
df_Drill = pd.read_json(Drilling_Completions)
df_Drill
```

Out[507]:

	afeamtcalc	afeamtnormcalc	afecosttypcalc	afenumbercalc	afeperdurmlcalc	afeperdurmlnormca
0	17945777	17945777	9876543-Capital	9876543	612568.711864	612568.71186
1	7753000	7753000	1234567C-Capital	1234567C	NaN	Na

2 rows × 144 columns

In [508]:
```
df_Drill['jobtyp']
```

Out[508]:
```
0    Drilling - original
1             Completion
Name: jobtyp, dtype: object
```

In [509]:
```
df_Drill['costfinalactual']
```

Out[509]:
```
0    10255618
1     7816255
Name: costfinalactual, dtype: int64
```

In [510]:
```
df_Drill['costfinalactual'][0]+df_Drill['costfinalactual'][1] #Cost for Bot
```

Out[510]:
```
18071873
```

In [511]:
```
import seaborn as sns
```

```
#Mapped Names From Peloton Platform to rename the columns


          "afeamtcalc" : "Total AFE Amount"
         "afeamtnormcalc" : "Normalized Total AFE Amount"
         "afenumbercalc": "AFE Number"
         "afecosttypcalc": "AFE Number - Cost Type"
         "afeperdurmlcalc": "AFE/Duration ML"
         "afeperdurmlnormcalc": "Normalized AFE/Duration ML"
         "afepertargetdepthcalc": "AFE/Target Depth"
         "afepertargetdepthnormcalc": "Normalized AFE/Target Depth"
         "afesupamtcalc": "Total AFE Supplemental Amoun"
         "afesupamtnormcalc": "Normalized Total AFE Supp Amount"
         "afetotalcalc": "Total AFE + Supp Amount"
         "afetotalnormcalc": "Normalized Total AFE + Supp Amount"
         "bhadrillruncalc": "BHA Drill Runs"
         "bhatotalruncalc": "BHA Runs"
         "bitrevscalc": "Bit Revs"
         "costafeforecastvarcalc": "AFE - Forecast"
         "costfinalactual": "Actual Final Job Cost"
         "costforecastfieldvarcalc": "Forecast - Field Est"
         "costmaxtotalcalc": "Max Cost"
         "costmintotalcalc": "Min Cost"
         "costmltotalcalc": "Planned Most Likely Cost"
         "costmltotalnoplanchangecalc": "Planned Most Likely Cos"
         "costmlnoexcludecalc": "Cum ML Phase Cost"
         "costnormafeforecastvarcalc": "Normalized AFE - Forecast"
         "costnormforecastfieldvarcalc": "Normalized Forecast - Field"
         "costnormperdepthcalc": "Normalized Cost Per Depth Drilled"
         "costnormtotalcalc": "Normalized Total Field Est"
         "costperdepthcalc": "Cost Per Depth Drilled"
         "costperdepthplanmlcalc": "Planned ML Cost / Depth"
         "costnormperdepthplanmlcalc": "Normalized Planned ML Cost / Depth"
         "costtechlimittotalcalc": "Plan Tech Limit Cost"
         "costpertldurcalc": "Cost/Hour"
         "costpertldurnormcalc": "Normalized Cost Per Hour"
         "costtotalcalc": "Total Field Estimate"
         "depthdrilledcalc": "Total Depth Drilled"
         "depthdrilledperbhacalc": "Depth Drilled/BHA Drill Runs"
         "depthdrilledspudtorrcalc": "Depth Drilled/(Spud to RR)"
         "depthdrilledperreportnocalc": "Depth Drilled/Report #"
         "depthperdurplanmlcalc": "Planned Depth / ML Duration"
         "depthperratiodurationcalc": "DepthDrilled/(TL Hours/Ref Hours)"
         "depthplanmaxcalc": "Planned Max Depth"
         "depthrotatingcalc": "Depth Rotating"
         "depthslidingcalc": "Depth Sliding"
         "dttmend": "End Date"
         "dttmendplanmaxcalc": "Max Planned End Date"
         "dttmendplanmincalc": "Min Planned End Date"
         "dttmendplanmlcalc": "Planned Most Likely End Date"
         "dttmspud": "Spud Date"
         "dttmstartplan": "Planned Start Date"
         "dttmtotaldepthcalc": "Date TD Reached (wellbore)"
         "durationmaxtotalcalc": "Max Duration"
         "durationmintotalcalc": "Min Duration"
         "durationmltotalcalc": "Most Likely Duration"
         "durationnoproblemtimecalc": "Time Log - Problem Hours"
```

```
        "durationproblemtimecalc": "Total Problem Hours"
        "durationspudtoplanmlcalc": "Spud to Plan ML Duration"
        "durationspudtoplanmaxcalc": "Spud to Plan Max Duration"
        "durationspudtoplanmincalc": "Spud to Plan Min Duration"
        "durationspudtimelogcalc": "Total Time Log Hours (Spud to TD)"
        "durationspudtotdcalc": "Spud to TD Duration"
        "durationspudtorrcalc": "Spud to RR Duration"
        "durationtechlimittotalcalc": "Tech Limit Duration"
        "durationtimelogtotalcalc": "Total Time Log Hours"
        "durmltotalnoplanchangecalc": "Most Likely Duration (no plan
change)"
        "durmlnoexcludecalc": "Cum ML Duration (no excludes)"
        "estcostnormsavecalc": "Normalized Possible Cost Savings"
        "estcostsavecalc": "Possible Cost Savings"
        "estproblemcostcalc": "Estimated Problem Cost"
        "estproblemcostnormcalc": "Normalized Estimated Problem Cost"
        "estproblemtimecalc": "Estimated Problem Time"
        "esttimesavecalc": "Possible Time Savings"
        "finalinvoicetotalcalc": "Total Final Invoice"
        "finalinvoicetotalnormcalc": "Normalized Total Final Invoice"
        "idreclastrigcalc": "Rig"
        "idreclastrigcalctk": "Rig Table"
        "idrecwellbore": "Primary Wellbore Affected"
        "idrecwellboretk": "Primary Wellbore Affected Table"
        "idrecwellborecalc": "Wellbore"
        "idrecwellborecalctk": "Wellbore Table"
        "jobsupplycostcalc": "Total Job Supply Field Est"
        "jobsupplycostnormcalc": "Normalized Total Job Supply Field Est"
        "jobtyp": "Primary Job Type"
        "mudcostcalc": "Total Mud Additive Field Est"
        "mudcostnormcalc": "Normalized Total Mud Additive Field Est"
        "mudcostperdepthcalc": "Total Mud Cost Per Depth Drilled"
        "mudcostperdepthnormcalc": "Normalized Total Mud Cost Per Depth
Drilled"
        "muddensitymaxcalc": "Max Program Mud Density"
        "muddensitymincalc": "Min Mud Density"
        "mudtypcalc": "Mud Type"
         "pctproblemtimecalc": "Problem Time"
        "percenttmrotatingcalc": "Rotating Time"
        "percenttmslidingcalc": "Sliding Time"
        "percentdepthrotatingcalc": "% Rotating Depth"
        "percentdepthslidingcalc": "% Sliding Depth"
        "programmuddensitymaxcalc": "Max Program Mud Density"
        "programmuddensitymincalc": "Min Program Mud Density"
        "projectrefnumbercalc": "Project Ref #"
        "ratiodepthactualplancalc": "Actual Depth/Planned Depth"
        "ratiodepthactualtargetcalc": "Actual Depth/Target Depth"
        "ratiodurtimelogrefhourscalc": "Time Log Hours/Ref Hours"
        "reportnocalc": "Total Number of Reports"
        "ropavgfromspudcalc": "Avg ROP from Spud"
        "ropcalc": "Avg ROP"
        "roprotatingcalc": "Rotating ROP"
        "ropslidingcalc": "Sliding ROP"
        "ropspudtimelogcalc": "Avg ROP from Spud Time"
        "roptimelogcalc": "Avg ROP from Time Log"
        "safetyincnocalc": "# of Safety Incidents"
        "safetyincreportnocalc": "# of Reportable Safety Incidents"
```

```
"targetform": "Target Formation"
"tmcirccalc": "Circulating Time"
"tmdrillcalc": "Drilling Time"
"tmothercalc": "Other Time"
"tmrotatingcalc": "Rotating Time"
"tmslidingcalc": "Sliding Time"
"tmtripcalc": "Tripping Time"
"totaldepthcalc": "Total Depth Reached (wellbore)"
"totaldepthtvdcalc": "Total Depth (TVD) Reached"
"tdtomudcalc": "TD (wellbore) - Mud Line"
"varianceafefinalcalc": "AFE - Final Invoice"
"variancefieldcalc": "AFE - Field Est"
"variancefieldfinalcalc": "Field - Final Invoice"
"variancefinalcalc": "AFE - Final Job Cost"
"variancenormafefinalcalc": "Normalized AFE - Final Invoice"
"variancenormfieldcalc": "Normalized AFE - Field Est"
"variancenormfieldfinalcalc": "Normalized Field - Final Invoice"
"variancenormfinalcalc": "Normalized AFE - Final Job Cost"
"wvtyp": "Job Category"
"syslockdate": "Lock Date"
"sysmoddate": "Last Mod Date"
"sysmoduser": "Last Mod By"
"syscreatedate": "Create Date"
"syscreateuser": "Created By"
```

In [512]:

```
df_Drill_renamed = df_Drill.rename(columns={ "afeamtcalc": "Total AFE Amoun
        "afenumbercalc": "AFE Number",
        "afecosttypcalc": "AFE Number - Cost Type",
        "afeperdurmlcalc": "AFE/Duration ML",
        "afeperdurmlnormcalc": "Normalized AFE/Duration ML",
        "afepertargetdepthcalc": "AFE/Target Depth",
        "afepertargetdepthnormcalc": "Normalized AFE/Target Depth",
        "afesupamtcalc": "Total AFE Supplemental Amount",
        "afesupamtnormcalc": "Normalized Total AFE Supp Amount",
        "afetotalcalc": "Total AFE + Supp Amount",
        "afetotalnormcalc": "Normalized Total AFE + Supp Amount",
        "bhadrillruncalc": "BHA Drill Runs",
        "bhatotalruncalc": "BHA Runs",
        "bitrevscalc": "Bit Revs",
        "costafeforecastvarcalc": "AFE - Forecast",
        "costfinalactual": "Actual Final Job Cost",
        "costforecastfieldvarcalc": "Forecast - Field Est",
        "costmaxtotalcalc": "Max Cost",
        "costmintotalcalc": "Min Cost",
        "costmltotalcalc": "Planned Most Likely Cost",
        "costmltotalnoplanchangecalc": "Planned Most Likely Cost",
        "costmlnoexcludecalc": "Cum ML Phase Cost",
        "costnormafeforecastvarcalc": "Normalized AFE - Forecast",
        "costnormforecastfieldvarcalc": "Normalized Forecast - Field",
        "costnormperdepthcalc": "Normalized Cost Per Depth Drilled",
        "costnormtotalcalc": "Normalized Total Field Est",
        "costperdepthcalc": "Cost Per Depth Drilled",
        "costperdepthplanmlcalc": "Planned ML Cost / Depth",
        "costnormperdepthplanmlcalc": "Normalized Planned ML Cost / Depth",
        "costtechlimittotalcalc": "Plan Tech Limit Cost",
        "costpertldurcalc": "Cost/Hour",
        "costpertldurnormcalc": "Normalized Cost Per Hour",
        "costtotalcalc": "Total Field Estimate",
        "depthdrilledcalc": "Total Depth Drilled",
        "depthdrilledperbhacalc": "Depth Drilled/BHA Drill Runs",
        "depthdrilledspudtorrcalc": "Depth Drilled/(Spud to RR)",
        "depthdrilledperreportnocalc": "Depth Drilled/Report #",
        "depthperdurplanmlcalc": "Planned Depth / ML Duration",
        "depthperratiodurationcalc": "DepthDrilled/(TL Hours/Ref Hours)",
        "depthplanmaxcalc": "Planned Max Depth",
        "depthrotatingcalc": "Depth Rotating",
        "depthslidingcalc": "Depth Sliding",
        "dttmend": "End Date",
        "dttmendplanmaxcalc": "Max Planned End Date",
        "dttmendplanmincalc": "Min Planned End Date",
        "dttmendplanmlcalc": "Planned Most Likely End Date",
        "dttmspud": "Spud Date",
        "dttmstartplan": "Planned Start Date",
        "dttmtotaldepthcalc": "Date TD Reached (wellbore)",
        "durationmaxtotalcalc": "Max Duration",
        "durationmintotalcalc": "Min Duration",
        "durationmltotalcalc": "Most Likely Duration",
        "durationnoproblemtimecalc": "Time Log - Problem Hours",
        "durationproblemtimecalc": "Total Problem Hours",
        "durationspudtoplanmlcalc": "Spud to Plan ML Duration",
        "durationspudtoplanmaxcalc": "Spud to Plan Max Duration",
        "durationspudtoplanmincalc": "Spud to Plan Min Duration",
        "durationspudtimelogcalc": "Total Time Log Hours (Spud to TD)",
        "durationspudtotdcalc": "Spud to TD Duration",
        "durationspudtorrcalc": "Spud to RR Duration",
        "durationtechlimittotalcalc": "Tech Limit Duration",
        "durationtimelogtotalcalc": "Total Time Log Hours",
        "durmltotalnoplanchangecalc": "Most Likely Duration (no plan change
        "durmlnoexcludecalc": "Cum ML Duration (no excludes)",
        "estcostnormsavecalc": "Normalized Possible Cost Savings",
        "estcostsavecalc": "Possible Cost Savings",
        "estproblemcostcalc": "Estimated Problem Cost",
        "estproblemcostnormcalc": "Normalized Estimated Problem Cost",
        "estproblemtimecalc": "Estimated Problem Time",
        "esttimesavecalc": "Possible Time Savings",
        "finalinvoicetotalcalc": "Total Final Invoice",
        "finalinvoicetotalnormcalc": "Normalized Total Final Invoice",
        "idreclastrigcalc": "Rig",
        "idreclastrigcalctk": "Rig Table",
        "idrecwellbore": "Primary Wellbore Affected",
        "idrecwellboretk": "Primary Wellbore Affected Table",
        "idrecwellborecalc": "Wellbore",
        "idrecwellborecalctk": "Wellbore Table",
        "jobsupplycostcalc": "Total Job Supply Field Est",
        "jobsupplycostnormcalc": "Normalized Total Job Supply Field Est",
        "jobtyp": "Primary Job Type",
        "mudcostcalc": "Total Mud Additive Field Est",
        "mudcostnormcalc": "Normalized Total Mud Additive Field Est",
        "mudcostperdepthcalc": "Total Mud Cost Per Depth Drilled",
        "mudcostperdepthnormcalc": "Normalized Total Mud Cost Per Depth Dri
        "muddensitymaxcalc": "Max Program Mud Density",
        "muddensitymincalc": "Min Mud Density",
        "mudtypcalc": "Mud Type",
         "pctproblemtimecalc": "Problem Time",
        "percenttmrotatingcalc": "Rotating Time",
        "percenttmslidingcalc": "Sliding Time",
        "percentdepthrotatingcalc": "% Rotating Depth",
        "percentdepthslidingcalc": "% Sliding Depth",
        "programmuddensitymaxcalc": "Max Program Mud Density",
        "programmuddensitymincalc": "Min Program Mud Density",
        "projectrefnumbercalc": "Project Ref #",
        "ratiodepthactualplancalc": "Actual Depth/Planned Depth",
        "ratiodepthactualtargetcalc": "Actual Depth/Target Depth",
        "ratiodurtimelogrefhourscalc": "Time Log Hours/Ref Hours",
        "reportnocalc": "Total Number of Reports",
        "ropavgfromspudcalc": "Avg ROP from Spud",
        "ropcalc": "Avg ROP",
        "roprotatingcalc": "Rotating ROP",
        "ropslidingcalc": "Sliding ROP",
        "ropspudtimelogcalc": "Avg ROP from Spud Time",
        "roptimelogcalc": "Avg ROP from Time Log",
        "safetyincnocalc": "# of Safety Incidents",
        "safetyincreportnocalc": "# of Reportable Safety Incidents",
        "targetform": "Target Formation",
        "tmcirccalc": "Circulating Time",
        "tmdrillcalc": "Drilling Time",
        "tmothercalc": "Other Time",
        "tmrotatingcalc": "Rotating Time",
        "tmslidingcalc": "Sliding Time",
        "tmtripcalc": "Tripping Time",
        "totaldepthcalc": "Total Depth Reached (wellbore)",
        "totaldepthtvdcalc": "Total Depth (TVD) Reached",
        "tdtomudcalc": "TD (wellbore) - Mud Line",
        "varianceafefinalcalc": "AFE - Final Invoice",
        "variancefieldcalc": "AFE - Field Est",
        "variancefieldfinalcalc": "Field - Final Invoice",
        "variancefinalcalc": "AFE - Final Job Cost",
        "variancenormafefinalcalc": "Normalized AFE - Final Invoice",
        "variancenormfieldcalc": "Normalized AFE - Field Est",
        "variancenormfieldfinalcalc": "Normalized Field - Final Invoice",
        "variancenormfinalcalc": "Normalized AFE - Final Job Cost",
        "wvtyp": "Job Category",
        "syslockdate": "Lock Date",
        "sysmoddate": "Last Mod Date",
        "sysmoduser": "Last Mod By",
        "syscreatedate": "Create Date",
        "syscreateuser": "Created By",})
```

```
In [513]: df_Drill_renamed
```

Out[513]:

	Total AFE Amount	Normalized Total AFE Amount	AFE Number - Cost Type	AFE Number	AFE/Duration ML	Normalized AFE/Duration ML	AFE/Target Depth	Norm AFE/
0	17945777	17945777	9876543-Capital	9876543	612568.711864	612568.711864	5155.419734	5155.4
1	7753000	7753000	1234567C-Capital	1234567C	NaN	NaN	2211.856701	2211.8

2 rows × 144 columns

```
In [514]: df_Drill_renamed['Job Category']
```

```
Out[514]: 0                Drilling
          1    Completion/Workover
          Name: Job Category, dtype: object
```

```
In [515]: df_Drill_renamed['summary'][0] #Drilling Summary=Successful Drilling Job
```

```
Out[515]: 'No major problems were encountered while drilling this well.  Note that
          the well was completed under budget and within the allocated number of da
          ys.'
```

```
In [516]: df_Drill_renamed['Avg ROP'][0] #ft/hr
```

```
Out[516]: 367.70111342093014
```

```
In [517]: df_Drill_renamed['Total Depth Drilled'][0] #ft
```

```
Out[517]: 3390.0
```

In [518]:

```
import pandas as pd
import numpy as np
%matplotlib inline
```

In [519]:

```
from plotly import __version__
```

In [520]:

```
import cufflinks as cf
```

In [521]:

```
from plotly.offline import download_plotlyjs,init_notebook_mode,plot,iplot
```

In [522]:

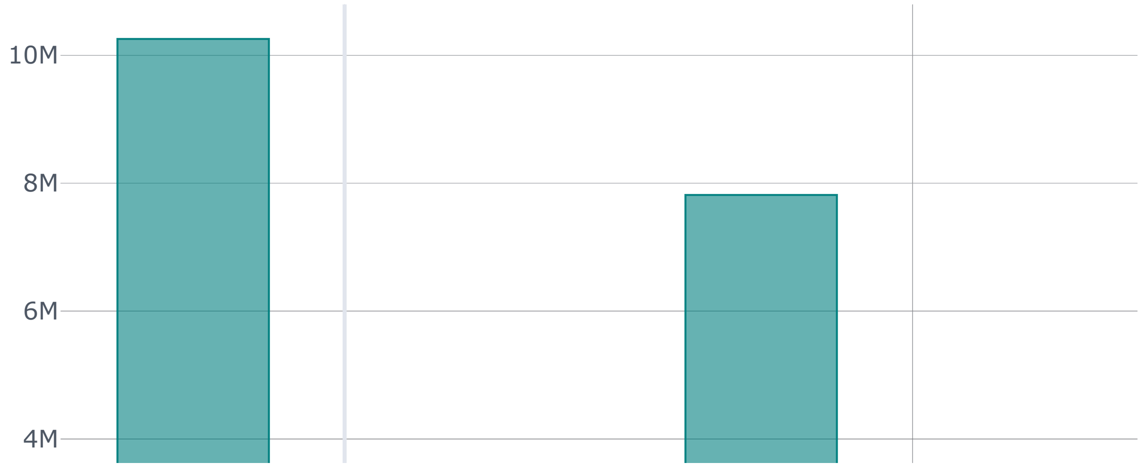
```
init_notebook_mode(connected=True)
```

In [523]:

```
cf.go_offline()
```

In [445]:

```
df_Drill_renamed.iplot(kind='bar')
```



In [524]:

```
#Machine Learning Part
#Logistic Regression, 1=Successful Well 0=Unsuccessful Well
#Add New Column To Data Frame

import pandas as pd
df_Drill_Final = df_Drill_renamed.assign(Successful_Well = ['1', '0'])
df_Drill_Final
```

Out[524]:

	Total AFE Amount	Normalized Total AFE Amount	AFE Number - Cost Type	AFE Number	AFE/Duration ML	Normalized AFE/Duration ML	AFE/Target Depth	Norm AFE/
0	17945777	17945777	9876543-Capital	9876543	612568.711864	612568.711864	5155.419734	5155.4
1	7753000	7753000	1234567C-Capital	1234567C	NaN	NaN	2211.856701	2211.8

2 rows × 145 columns

In [525]:

```
df_Drill_Final.iloc[0][30] #Cost/Hour
```

Out[525]:

```
383367.49852060893
```

In [526]:

```
df_Drill_Final.iloc[0][16] #Actual Final Job Cost
```

Out[526]:

```
10255618
```

In [527]:

```
df_Drill_Final.iloc[0][34] #Total Depth Drilled
```

Out[527]:

```
3390.0
```

In [528]:

```
Drill_Final = df_Drill_Final[['Cost/Hour','Actual Final Job Cost','Total De
#Make Separate DataFrame To Include only the Variables needed
```

In [529]:

```
#Combined Columns and Added a New Column
Drill_Final
```

Out[529]:

	Cost/Hour	Actual Final Job Cost	Total Depth Drilled	Successful_Well
0	383367.498521	10255618	3390.0	1
1	443201.643939	7816255	NaN	0

In [452]:

```
Drilling_Final = Drill_Final.fillna(0) #Used to Fill any unknown value
Drilling_Final
```

Out[452]:

	Cost/Hour	Actual Final Job Cost	Total Depth Drilled	Successful_Well
0	383367.498521	10255618	3390.0	1
1	443201.643939	7816255	0.0	0

In [453]:

```
#Machine Learning

train_X = Drilling_Final.drop(columns=['Successful_Well'])
train_X
```

Out[453]:

	Cost/Hour	Actual Final Job Cost	Total Depth Drilled
0	383367.498521	10255618	3390.0
1	443201.643939	7816255	0.0

In [454]:

```
train_y = Drilling_Final[['Successful_Well']]
```

In [455]:

```
import matplotlib.pyplot as plt
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix
```

In [456]:

```
model = LogisticRegression(solver='liblinear', random_state=0)
```

In [457]:

```
model.fit(train_X, train_y)
```

```
/Applications/anaconda3/lib/python3.7/site-packages/sklearn/utils/validat
ion.py:761: DataConversionWarning:

A column-vector y was passed when a 1d array was expected. Please change
the shape of y to (n_samples, ), for example using ravel().
```

Out[457]:

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=Tr
ue,
          intercept_scaling=1, max_iter=100, multi_class='warn',
          n_jobs=None, penalty='l2', random_state=0, solver='liblinear',
          tol=0.0001, verbose=0, warm_start=False)
```

In [458]:

```
model = LogisticRegression(solver='liblinear', random_state=0).fit(train_X,
```

In [459]:

```
model.classes_
```

Out[459]:

```
array(['0', '1'], dtype=object)
```

In [460]:

```
model.intercept_
```

Out[460]:

```
array([-1.91676192e-10])
```

In [461]:

```
model.coef_
```

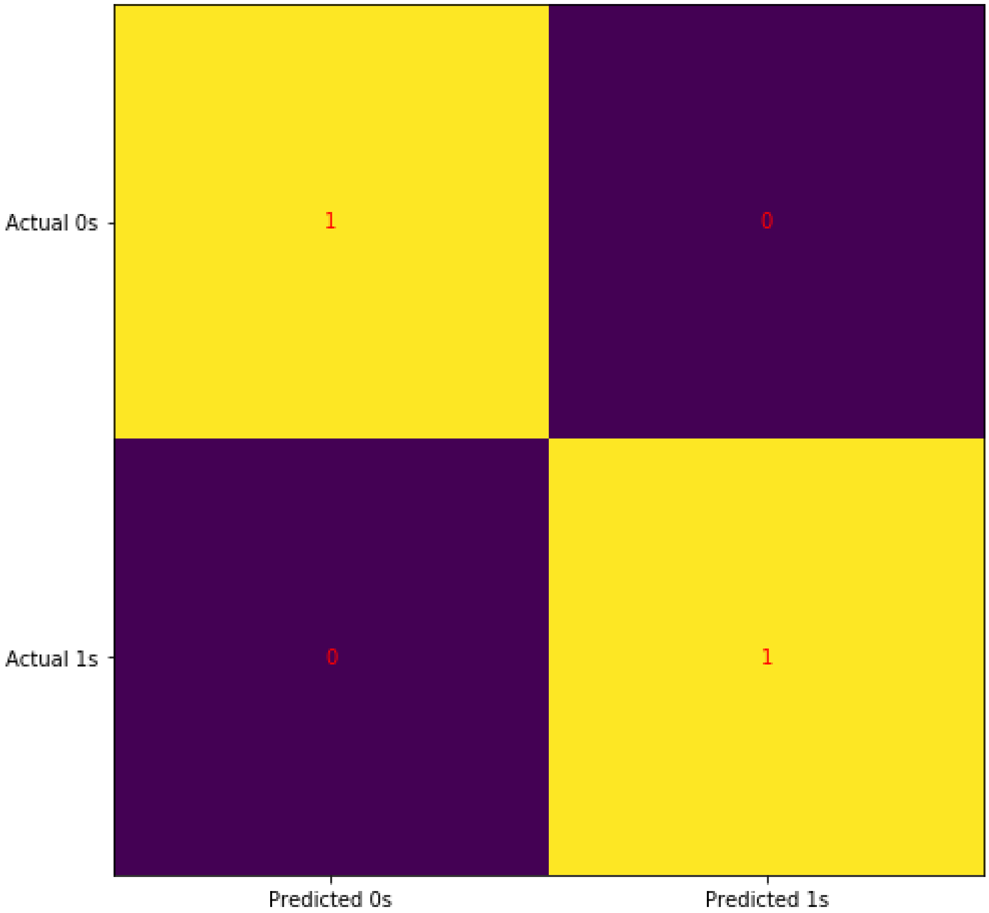
Out[461]:

```
array([[-1.21836608e-04,  5.58165413e-06,  2.08980209e-06]])
```

```
In [462]: model.predict_proba(train_X)

Out[462]: array([[2.63978596e-05, 9.99973602e-01],
                 [9.99968659e-01, 3.13409679e-05]])

In [463]: model.predict(train_X)

Out[463]: array(['1', '0'], dtype=object)

In [464]: model.score(train_X, train_y)

Out[464]: 1.0

In [465]: confusion_matrix(train_y, model.predict(train_X))

Out[465]: array([[1, 0],
                 [0, 1]])
```

In [530]:

```
cm = confusion_matrix(y, model.predict(train_X))

fig, ax = plt.subplots(figsize=(8, 8))
ax.imshow(cm)
ax.grid(False)
ax.xaxis.set(ticks=(0, 1), ticklabels=('Predicted 0s', 'Predicted 1s'))
ax.yaxis.set(ticks=(0, 1), ticklabels=('Actual 0s', 'Actual 1s'))
ax.set_ylim(1.5, -0.5)
for i in range(2):
    for j in range(2):
        ax.text(j, i, cm[i, j], ha='center', va='center', color='red')
plt.show()
```



In [532]:

```
df_Drill_Final
```

Out[532]:

	Total AFE Amount	Normalized Total AFE Amount	AFE Number - Cost Type	AFE Number	AFE/Duration ML	Normalized AFE/Duration ML	AFE/Target Depth	Norm AFE/
0	17945777	17945777	9876543-Capital	9876543	612568.711864	612568.711864	5155.419734	5155.4
1	7753000	7753000	1234567C-Capital	1234567C	NaN	NaN	2211.856701	2211.8

2 rows × 145 columns

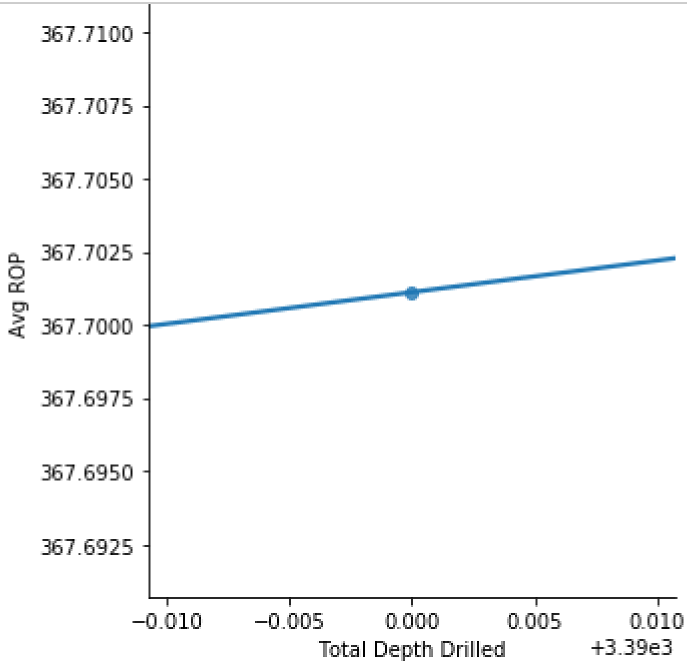
```
In [537]: df_Drill_Final['Avg ROP'][0]
```

```
Out[537]: 367.70111342093014
```

```
In [536]: df_Drill_Final['Total Depth Drilled'][0]
```

```
Out[536]: 3390.0
```

```
In [540]: sns.lmplot('Total Depth Drilled','Avg ROP',df_Drill_Final)
```



```
In [541]: #New Data Set, not Sample 11
```

In [543]:

```
dict = {'ROP(m/hr)':[69.53, 46.67, 50.48, 39.27,62.07,91.31,30.96,40.27,92.
        'WOB(klbs)': [26.9, 16.1, 25.8, 33.3,30.8,36.3,15.8,49.2,23.7,32.9,
        'RPM (rpm)':[60, 60, 135, 135,135,204,205,128,204,205,204,205,133,8

df = pd.DataFrame(dict)


df
```

Out[543]:

```
     ROP(m/hr)  WOB(klbs)  RPM (rpm)
 0       69.53       26.9         60
 1       46.67       16.1         60
 2       50.48       25.8        135
 3       39.27       33.3        135
 4       62.07       30.8        135
 5       91.31       36.3        204
 6       30.96       15.8        205
 7       40.27       49.2        128
 8       92.38       23.7        204
 9       37.70       32.9        205
10       60.79       31.8        204
11       71.56       34.2        205
12       11.28       42.4        133
13       33.44       31.4         81
14       36.43       33.1         90
```

In [544]:

```
from keras.models import Sequential
from keras.layers import Dense
import pandas as pd
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
```

In [545]:

```
dataset = df.values
```

```
In [546]: dataset
```

```
Out[546]: array([[ 69.53,  26.9 ,  60.  ],
                 [ 46.67,  16.1 ,  60.  ],
                 [ 50.48,  25.8 , 135.  ],
                 [ 39.27,  33.3 , 135.  ],
                 [ 62.07,  30.8 , 135.  ],
                 [ 91.31,  36.3 , 204.  ],
                 [ 30.96,  15.8 , 205.  ],
                 [ 40.27,  49.2 , 128.  ],
                 [ 92.38,  23.7 , 204.  ],
                 [ 37.7 ,  32.9 , 205.  ],
                 [ 60.79,  31.8 , 204.  ],
                 [ 71.56,  34.2 , 205.  ],
                 [ 11.28,  42.4 , 133.  ],
                 [ 33.44,  31.4 ,  81.  ],
                 [ 36.43,  33.1 ,  90.  ]])
```

```
In [550]: X = dataset[:,1:3]
          y = dataset[:,0]
```

```
In [551]: from sklearn import preprocessing
          min_max_scaler = preprocessing.MinMaxScaler()
          X_scale = min_max_scaler.fit_transform(X)
          X_scale
```

```
Out[551]: array([[0.33233533, 0.        ],
                 [0.00898204, 0.        ],
                 [0.2994012 , 0.51724138],
                 [0.5239521 , 0.51724138],
                 [0.4491018 , 0.51724138],
                 [0.61377246, 0.99310345],
                 [0.        , 1.        ],
                 [1.        , 0.46896552],
                 [0.23652695, 0.99310345],
                 [0.51197605, 1.        ],
                 [0.47904192, 0.99310345],
                 [0.5508982 , 1.        ],
                 [0.79640719, 0.50344828],
                 [0.46706587, 0.14482759],
                 [0.51796407, 0.20689655]])
```

```
In [552]: X_train, X_test, y_train, y_test = train_test_split(X_scale, y, test_size=0
```

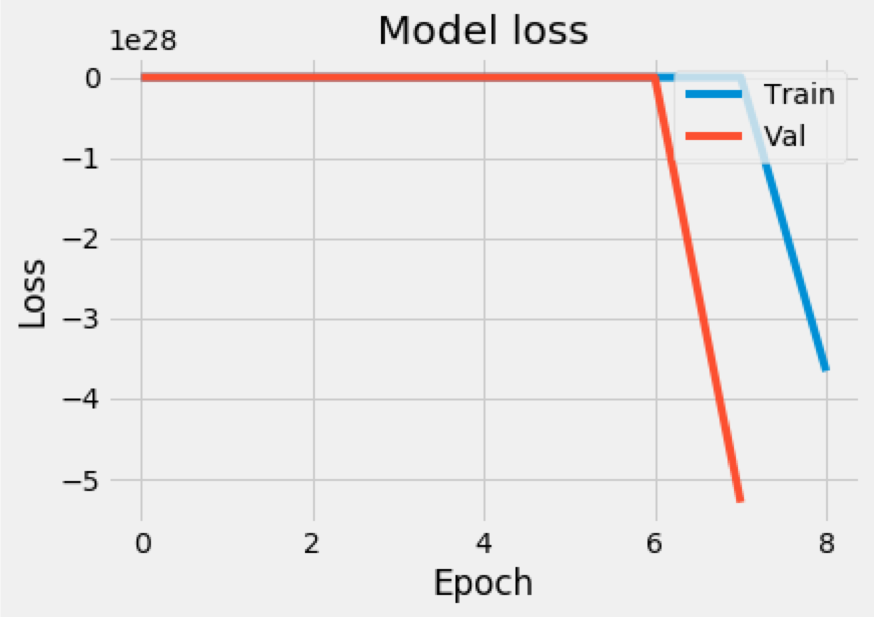
```
In [559]: model = Sequential([
              Dense(12, activation='relu', input_shape=( 2 ,)),
              Dense(15, activation='relu'),
              Dense(1, activation='sigmoid')
          ])
```

```
In [560]: model.compile(optimizer='sgd',
                        loss='binary_crossentropy',
                        metrics=['accuracy'])
```

In [561]:
```
hist = model.fit(X_train, y_train,
          batch_size=57, epochs=1000, validation_split=0.2)
```

```
Epoch 1/1000
1/1 [==============================] - 0s 375ms/step - loss: -4.4936 - ac
curacy: 0.0000e+00 - val_loss: -88.8579 - val_accuracy: 0.0000e+00
Epoch 2/1000
1/1 [==============================] - 0s 27ms/step - loss: -66.0701 - ac
curacy: 0.0000e+00 - val_loss: -257.3883 - val_accuracy: 0.0000e+00
Epoch 3/1000
1/1 [==============================] - 0s 32ms/step - loss: -190.2391 - a
ccuracy: 0.0000e+00 - val_loss: -1018.2347 - val_accuracy: 0.0000e+00
Epoch 4/1000
1/1 [==============================] - 0s 50ms/step - loss: -727.5964 - a
ccuracy: 0.0000e+00 - val_loss: -10112.3545 - val_accuracy: 0.0000e+00
Epoch 5/1000
1/1 [==============================] - 0s 46ms/step - loss: -7024.1084 -
accuracy: 0.0000e+00 - val_loss: -596277.4375 - val_accuracy: 0.0000e+00
Epoch 6/1000
1/1 [==============================] - 0s 41ms/step - loss: -411619.3438
- accuracy: 0.0000e+00 - val_loss: -1241003264.0000 - val_accuracy: 0.000
0e+00
```

In [562]:
```
#visualize the training loss and the validation loss to see if the model is
plt.plot(hist.history['loss'])
plt.plot(hist.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Val'], loc='upper right')
plt.show()
```



In [572]:

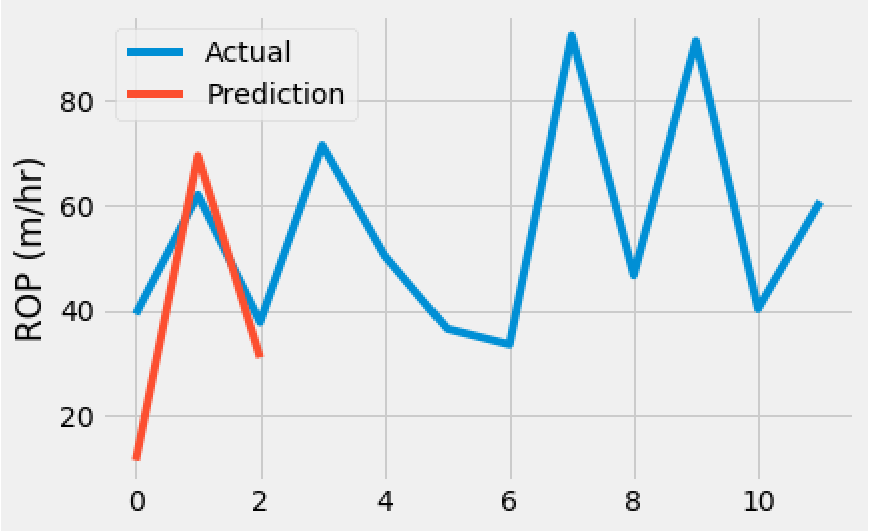
```
#Make a prediction & print the actual values
prediction = model.predict(X_test)
prediction  = [1 if y>=0.5 else 0 for y in prediction] #Threshold
print(prediction)
print(y_test)
```

```
[0, 0, 0]
[11.28 69.53 30.96]
```

In [585]:

```
plt.plot(y_train,label='Actual')
plt.plot(y_test,label='Prediction')
#plt.xlabel('index')
plt.ylabel('ROP (m/hr)')
plt.legend()
print(y_train)
print(y_test)
```

```
[39.27 62.07 37.7  71.56 50.48 36.43 33.44 92.38 46.67 91.31 40.27 60.79]
[11.28 69.53 30.96]
```



In [ ]: