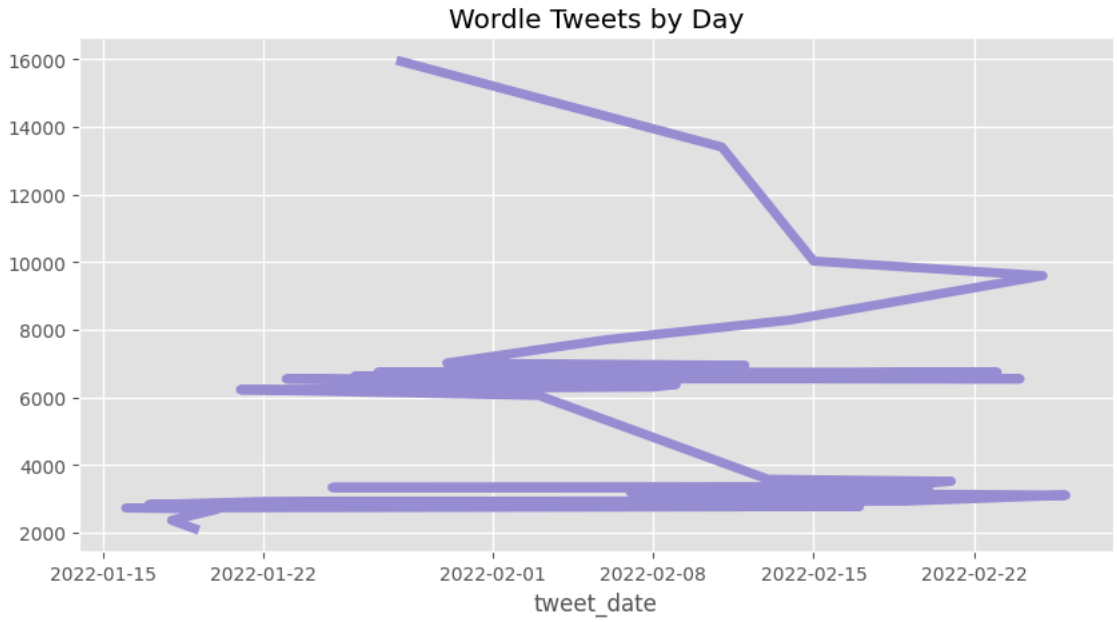
```
import pandas as pd
import numpy as np
import matplotlib.pylab as plt
import seaborn as sns
import re

pd.set_option('display.max_columns', 500)

from itertools import cycle

plt.style.use("ggplot")
color_pal = plt.rcParams["axes.prop_cycle"].by_key()["color"]
color_cycle = cycle(plt.rcParams["axes.prop_cycle"].by_key()["color"])
```

```
tweets = pd.read_csv("/content/tweets.csv")
```

```
def process_tweets(tweets):
    tweets['tweet_datetime'] = pd.to_datetime(tweets['tweet_date'])
    tweets['tweet_date'] = tweets['tweet_datetime'].dt.date
    tweets['wordle_id'] = tweets['tweet_text'].str[:10]
    tweets['n_attempts'] = tweets['tweet_text'].str[11].astype('int')
    tweets['id'] = tweets['tweet_text'].str[7:10].astype('int')
    return tweets

tweets = process_tweets(tweets)
```

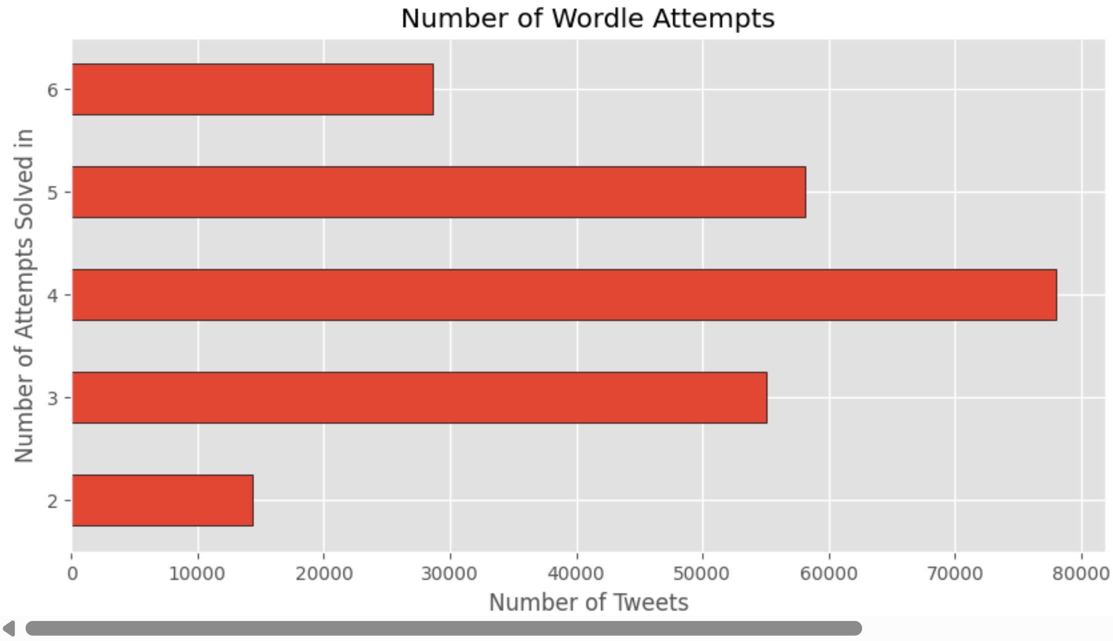
```
tweets['tweet_date'].value_counts() \
    .plot(figsize=(10, 5), title='Wordle Tweets by Day',
          color=color_pal[2], lw=5)
plt.show()
```



```
tweets.groupby('wordle_id')['n_attempts'].value_counts() \
    .unstack() \
    .style.background_gradient(axis=1)
```

```
n_attempts    2     3     4     5     6
wordle_id
Wordle 210  222   897  1064   489   165
Wordle 211  196   903  1016   564   215
Wordle 212  191   787   820   413   188
Wordle 213   51   260   630   697   481
Wordle 214  383  1012   823   363   122
Wordle 215  246   877  1017   579   224
Wordle 216  461  1952  2078  1199   415
Wordle 217  102   566  1031   843   418
Wordle 218  170  1009  1384   730   187
Wordle 219   38   343   927  1039   583
Wordle 220  392  1629  2389  1609   637
Wordle 221  274  1530  2630  1608   603
Wordle 222  567  2010  2397  1322   468
Wordle 223  597  2885  5086  4594  2614
Wordle 224  443  2100  2528  1418   457
Wordle 225  123  1305  2914  1856   699
Wordle 226  687  1801  1796  1189   628
Wordle 227  911  2436  2122   995   355
Wordle 228  774  1951  1925   968   388
Wordle 229  446  1614  2032  1605   807
Wordle 230  691  1993  2087  1323   571
Wordle 231  360  1836  2832  1974   811
Wordle 232   87   631  1195   937   474
Wordle 233  195   943  1762  2061  1492
Wordle 234  668  1423  1633  1574   927
Wordle 235  290  1425  2332  1688   751
Wordle 236  792  2519  3417  2260  1066
Wordle 237  383  1835  3293  3131  1518
Wordle 238  400  1587  2553  1793   732
Wordle 239  312  1382  1754  1169   423
Wordle 240  110   726  2258  2303  1051
Wordle 241  353  1791  2386  1572   733
Wordle 242  158   740  1063   775   486
Wordle 243  207   556   729   713   567
Wordle 244   97   540  1085  1062   573
Wordle 245   43   302   684  1026   878
Wordle 246  138   756  1157   878   399
Wordle 247  329   964  1130   765   332
Wordle 248  480  1408  1163   416   130
Wordle 249  307  1233  1805  1612  1253
Wordle 250  476  1603  2419  1782   826
Wordle 251  243   987  2661  3196  2002
```

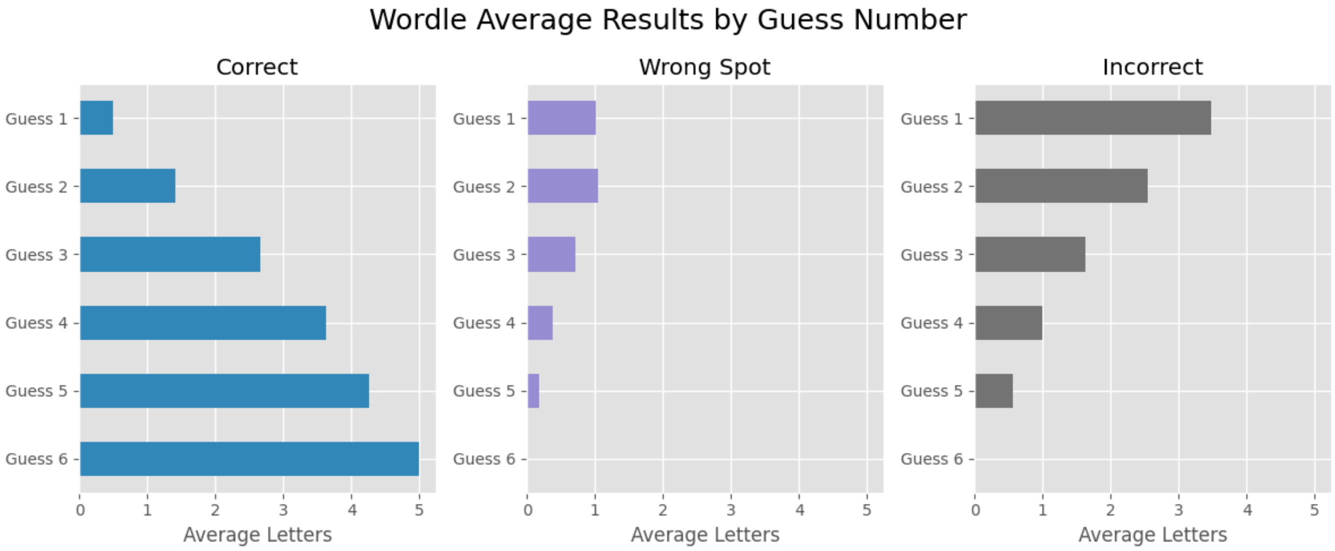
```
ax = tweets['n_attempts'].value_counts() \
    .sort_index() \
    .plot(figsize=(10, 5),
          kind='barh',
          title='Number of Wordle Attempts',
          edgecolor='black')
ax.set_xlabel('Number of Tweets')
ax.set_ylabel('Number of Attempts Solved in')
plt.show()
```



```
tweets['tweet_text'] = tweets['tweet_text'].str.replace('⬜','⬛')


def parse_tweet_text(tweets):
    for n in range(6):
        n += 1
        tweets[f'has_guess{n}'] = tweets['tweet_text'].str.split('\n').str[n+1] \
            .str.contains('|'.join(['🟨','⬛','🟩'])) \
            .fillna(False)

        # Parse out the tweet text
        tweets.loc[tweets[f'has_guess{n}'], f'guess{n}'] = \
                tweets['tweet_text'].str.split('\n').str[n + 1].str[:5]
        # Count number correct, misplaced, and incorrect
        tweets.loc[tweets[f'has_guess{n}'],
                       f'guess{n}_incorrect'] = tweets[f'guess{n}'].str.count('⬛')
        tweets.loc[tweets[f'has_guess{n}'],
                   f'guess{n}_wrong_spot'] = tweets[f'guess{n}'].str.count('🟨')
        tweets.loc[tweets[f'has_guess{n}'],
                   f'guess{n}_correct'] = tweets[f'guess{n}'].str.count('🟩')

        tweets.loc[tweets[f'guess{n}_correct'] == 6, 'final_guess'] = n
    return tweets

tweets = parse_tweet_text(tweets)
```

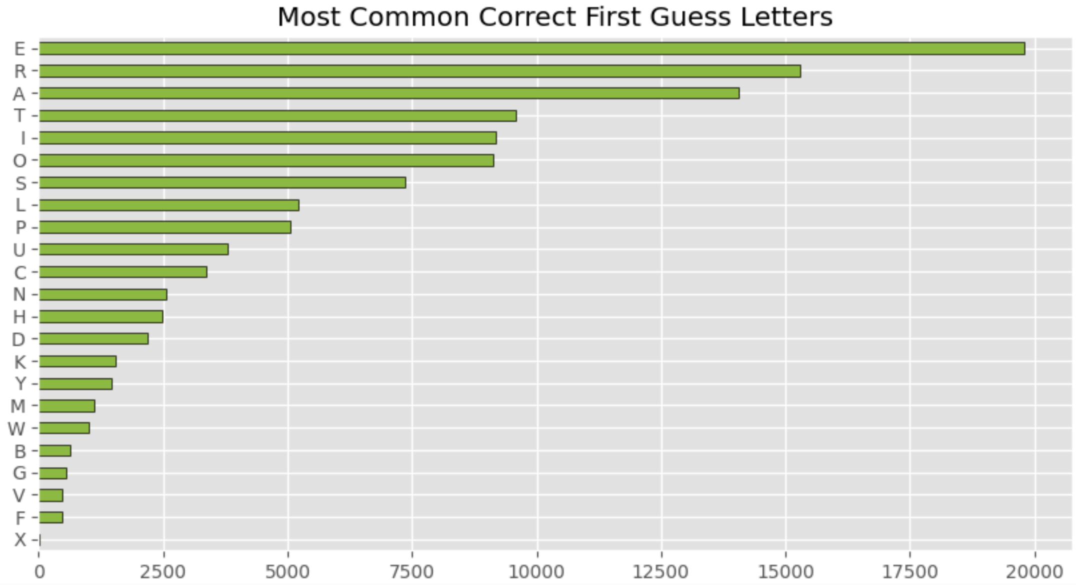
```
<ipython-input-10-eed98fc3e0d1>:6: FutureWarning: Downcasting object dtype arrays on .fillna, .ffill, .bfill is deprecated and will
  .fillna(False)
<ipython-input-10-eed98fc3e0d1>:6: FutureWarning: Downcasting object dtype arrays on .fillna, .ffill, .bfill is deprecated and will
  .fillna(False)
<ipython-input-10-eed98fc3e0d1>:6: FutureWarning: Downcasting object dtype arrays on .fillna, .ffill, .bfill is deprecated and will
  .fillna(False)
<ipython-input-10-eed98fc3e0d1>:6: FutureWarning: Downcasting object dtype arrays on .fillna, .ffill, .bfill is deprecated and will
  .fillna(False)
```

```
tweets['additional_text'] = \
    tweets.loc[~tweets['tweet_text'] \
    .str.split('\n').str[-1].str.contains('🟩')] \
    ['tweet_text'].str.split('\n').str[-1]


fig, axs = plt.subplots(1, 3, figsize=(12, 5), sharex=True)

for i, x in enumerate(['_correct','_wrong_spot','_incorrect']):
    col_subset = [c for c in tweets.columns if x in c]
    guess_avg = tweets[col_subset].mean()
    guess_avg.index = [f'Guess {i+1}' for i in range(6)]
    guess_avg.sort_index(ascending=False) \
        .plot(kind='barh',
              title=f'{x.strip("_").replace("_"," ").title()}',
              ax=axs[i],
                color=color_pal[i+1])
    axs[i].set_xlabel('Average Letters')
fig.suptitle('Wordle Average Results by Guess Number', fontsize=18)
plt.tight_layout()
plt.show()
```



```
answer_dict = {252 : 'SPILL',
     251 : 'VIVID',
     250 : 'BLOKE',
     249 : 'TROVE',
     248 : 'THORN',
     247 : 'OTHER',
     246 : 'TACIT',
     245 : 'SWILL',
     244 : 'DODGE',
     243 : 'SHAKE',
     242 : 'CAULK',
     241 : 'AROMA',
     240 : 'CYNIC',
     239 : 'ROBIN',
     238 : 'ULTRA',
     237 : 'ULCER',
     236 : 'PAUSE',
     235 : 'HUMOR',
     234 : 'FRAME',
     233 : 'ELDER',
     232 : 'SKILL',
     231 : 'ALOFT',
     230 : 'PLEAT',
     229 : 'SHARD',
     228 : 'MOIST',
     227 : 'THOSE',
     204 : 'GORGE',
     205 : 'QUERY',
     206 : 'DRINK',
     207 : 'FAVOR',
     208 : 'ABBEY',
     209 : 'TANGY',
     210 : 'PANIC',
     211 : 'SOLAR',
     212 : 'SHIRE',
     213 : 'PROXY',
     214 : 'POINT',
     215 : 'ROBOT',
     216 : 'PRICK',
     217 : 'WINCE',
     218 : 'CRIMP',
     219 : 'KNOLL',
     220 : 'SUGAR',
     221 : 'WHACK',
     222 : 'MOUNT',
     223 : 'PERKY',
     224 : 'COULD',
     225 : 'WRUNG',
     226 : 'LIGHT',
}
```

```
tweets['answer'] = tweets['id'].map(answer_dict)


first_guess_correct = []
for i, d in tweets.dropna(subset=['answer']).iterrows():
    example_text = d['guess1']
    example_solution = d['answer']
    results =[x.span()[0] for x in re.finditer('🟩', example_text)]
    first_guess_letters = [example_solution[i] for i in results]
    first_guess_correct += first_guess_letters


pd.Series(first_guess_correct).value_counts(ascending=True) \
    .plot(kind='barh', figsize=(10, 5),
         title='Most Common Correct First Guess Letters',
         edgecolor='black', color=color_pal[5])
plt.show()
```


Most Common Correct First Guess Letters
E
R
A
T
I
O
S
L
P
U
C
N
H
D
K
Y
M
W
B
G
V
F
X