In [87]:
```
import numpy as np
import pandas as pd
import sklearn
import sklearn.model_selection as ms
#import sklearn.grid_search as gs
from sklearn.model_selection import GridSearchCV as gs
import sklearn.feature_extraction.text as text
import sklearn.naive_bayes as nb
import matplotlib.pyplot as plt
from scipy import stats
import sklearn.linear_model as sklm

%matplotlib inline
```

In [88]:
```
import os
files = []
for i in os.listdir():
    if i.endswith('.csv'):
        files.append(i)
```

In [89]:
```
tweets = []
total= 0
for i in os.listdir():
    if i.endswith('.xlsx'):
        tweets.append(i)
for i in range(len(tweets)):
    current = tweets[i]
    app = pd.read_excel(current)
    total += len(app)
total
```

Out[89]: 3281

In [90]:
```
dict = {}

for i in range(len(files)):
    current = files[i]
    df = pd.read_csv(current)
    misplaced = list(df)
    misplaced1 = misplaced[0]
    df1 = df[misplaced1]
    current = np.array(df1)
    files[i] = files[i].replace("Num.csv", "")
    dict[files[i]] = current

df = pd.DataFrame({ key:pd.Series(value) for key, value in dict.items() })

copy = df
```

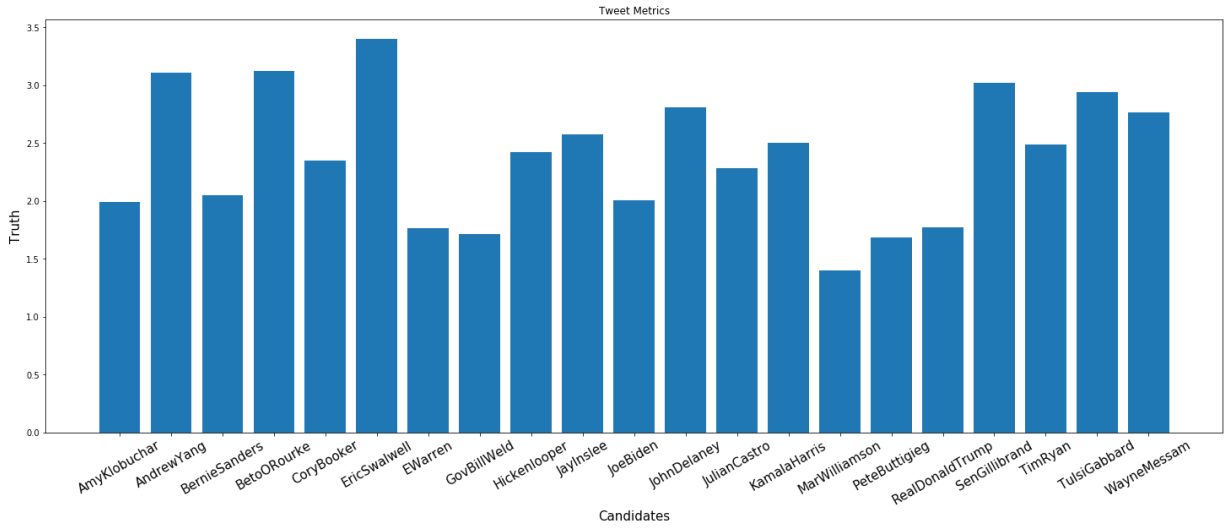
In [91]:
```
average = list()
for i in files:
    average.append(np.average(dict[i]))
```

In [92]:
```
averageDict = {}
for i in range(len(files)):
    averageDict[files[i]] = average[i]
```

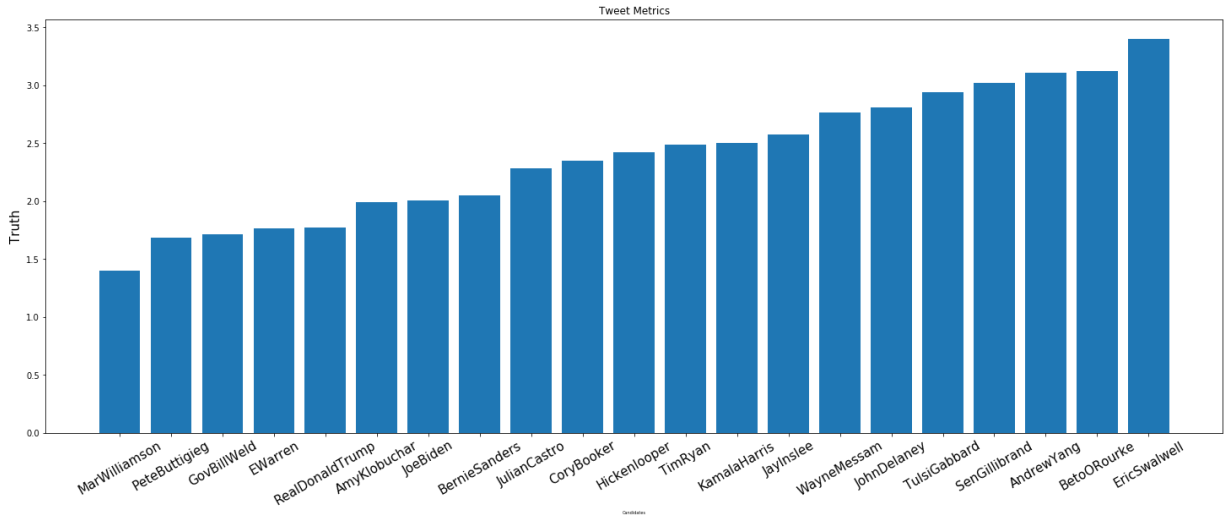
In [93]:
```
averageDict
```

Out[93]:
```
{'AmyKlobuchar': 1.9947643979057592,
 'AndrewYang': 3.1052631578947367,
 'BernieSanders': 2.05,
 'BetoORourke': 3.1237113402061856,
 'CoryBooker': 2.3518518518518516,
 'EricSwalwell': 3.4,
 'EWarren': 1.7630057803468209,
 'GovBillWeld': 1.7123287671232876,
 'Hickenlooper': 2.423841059602649,
 'JayInslee': 2.574803149606299,
 'JoeBiden': 2.006172839506173,
 'JohnDelaney': 2.8120805369127515,
 'JulianCastro': 2.281045751633987,
 'KamalaHarris': 2.5,
 'MarWilliamson': 1.402116402116402,
 'PeteButtigieg': 1.6868686868686869,
 'RealDonaldTrump': 1.7692307692307692,
 'SenGillibrand': 3.0228571428571427,
 'TimRyan': 2.4842767295597485,
 'TulsiGabbard': 2.9427083333333335,
 'WayneMessam': 2.7615384615384615}
```

In [95]:
```
index = np.arange(len(files))
plt.bar(index, average)
plt.xlabel('Candidates', fontsize=15)
plt.ylabel('Truth', fontsize=15)
plt.xticks(index, files, fontsize=15, rotation=30)
plt.title('Tweet Metrics')
plt.rcParams["figure.figsize"] = [25,9]
plt.show()
```



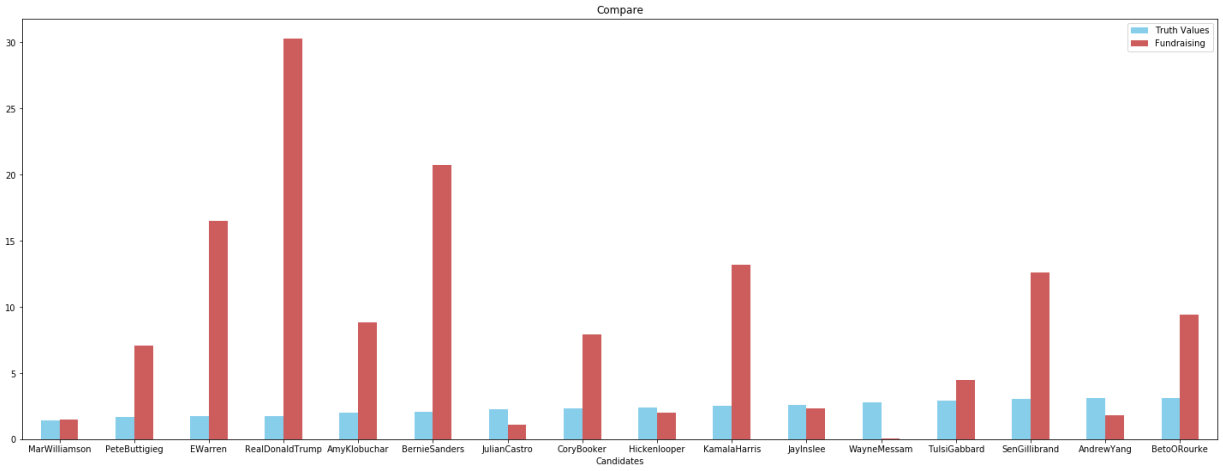
In [96]:
```
from operator import itemgetter
sortDict = sorted(averageDict.items(), key=itemgetter(1))

sortPeople, sortScores = [], []

for i in range(len(files)):
    sortPeople.append(sortDict[i][0])
    sortScores.append(sortDict[i][1])


index = np.arange(len(files))
plt.bar(index, sortScores)
plt.xlabel('Candidates', fontsize=5)
plt.ylabel('Truth', fontsize=15)
plt.xticks(index, sortPeople, fontsize=15, rotation=30)
plt.title('Tweet Metrics')
plt.rcParams["figure.figsize"] = [25,9]
plt.show()
```




In [97]:
```
fund = np.array([1.5,7.1,np.NaN,16.5,30.3, 8.8, np.NaN, 20.7,1.1,7.9,2.0,np.NaN,
```

In [98]:
```
scatterScores = sortScores
scatterFund = fund
scatterPeople = sortPeople
record = []
for i in range(len(scatterFund)):
    if np.isnan(scatterFund[i]):
        record.append(i)
scatterFund = np.delete(scatterFund, record)
scatterScores = np.delete(scatterScores, record)
scatterPeople = np.delete(scatterPeople, record)
```

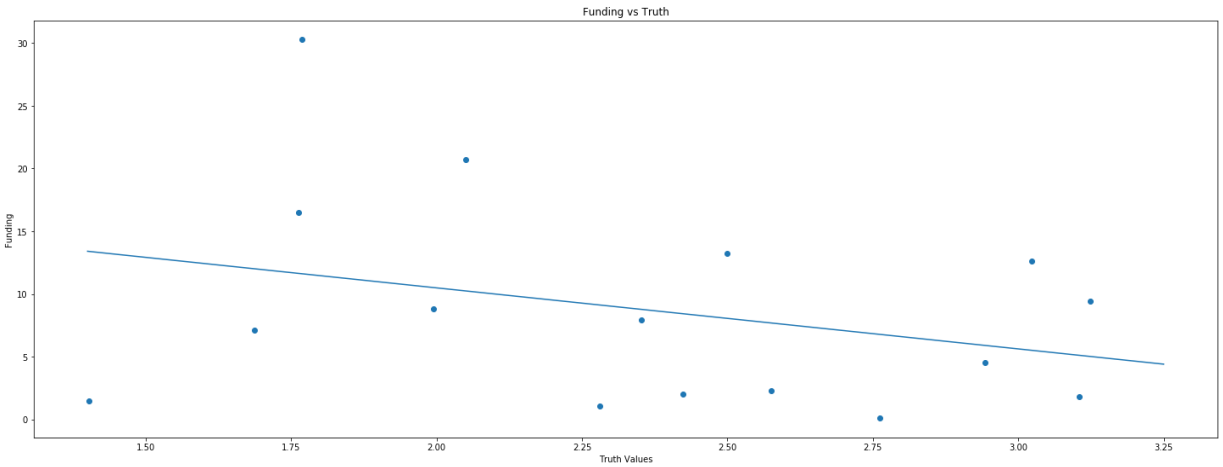
In [99]:
```
df = pd.DataFrame({"Truth Values":scatterScores,"Fundraising":scatterFund})
ax = df.plot.bar(color=["SkyBlue","IndianRed"], rot=0, title="Compare")
ax.set_xlabel("Candidates")
ax.xaxis.set_major_formatter(plt.FixedFormatter(scatterPeople))
```



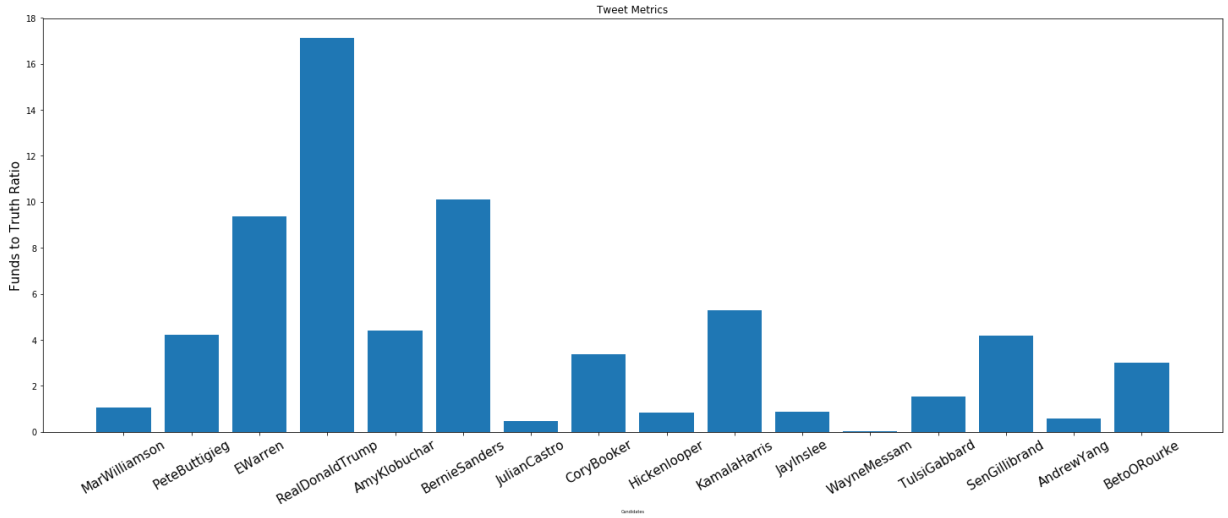

In [100]:
```
X = np.reshape(scatterScores, (-1,1))
y = scatterFund
model = sklm.LinearRegression().fit(X, y)
```

In [101]:
```
plt.scatter(scatterScores, scatterFund)
plt.xlabel('Truth Values', fontsize=10)
plt.ylabel('Funding', fontsize=10)
plt.title('Funding vs Truth')
line = np.linspace(1.4,3.25, 500)
C = (line * model.coef_) + model.intercept_
plt.plot(line, C)
plt.show
print(model.score(X,y))
```

```
0.10093438464069338
```



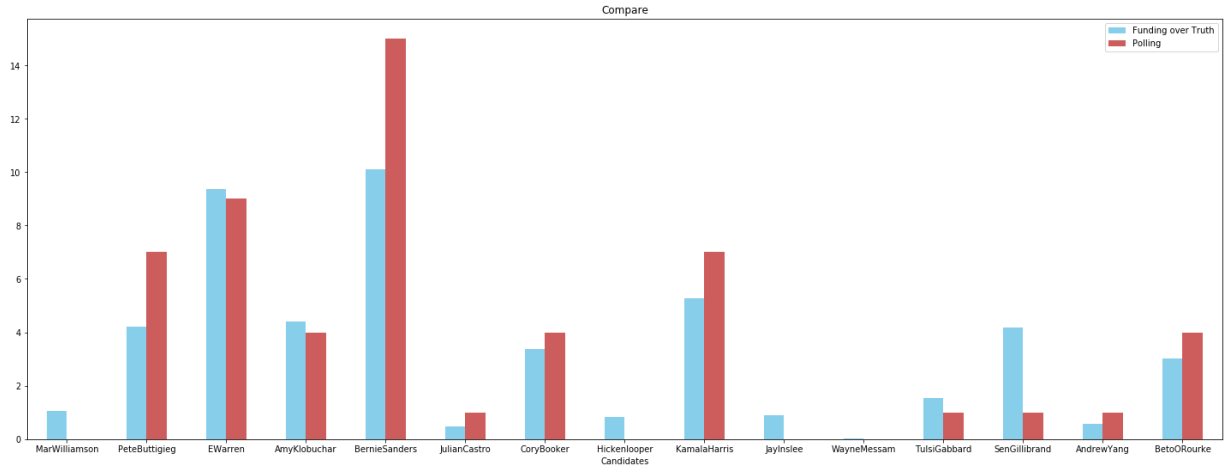
In [103]:
```
ratioList = []
for i in range(len(scatterFund)):
    ratioList.append(scatterFund[i]/scatterScores[i])
index = np.arange(len(scatterFund))
plt.bar(index, ratioList)
plt.xlabel('Candidates', fontsize=5)
plt.ylabel('Funds to Truth Ratio', fontsize=15)
plt.xticks(index, scatterPeople, fontsize=15, rotation=30)
plt.title('Tweet Metrics')
plt.rcParams["figure.figsize"] = [25,9]
plt.show()
```



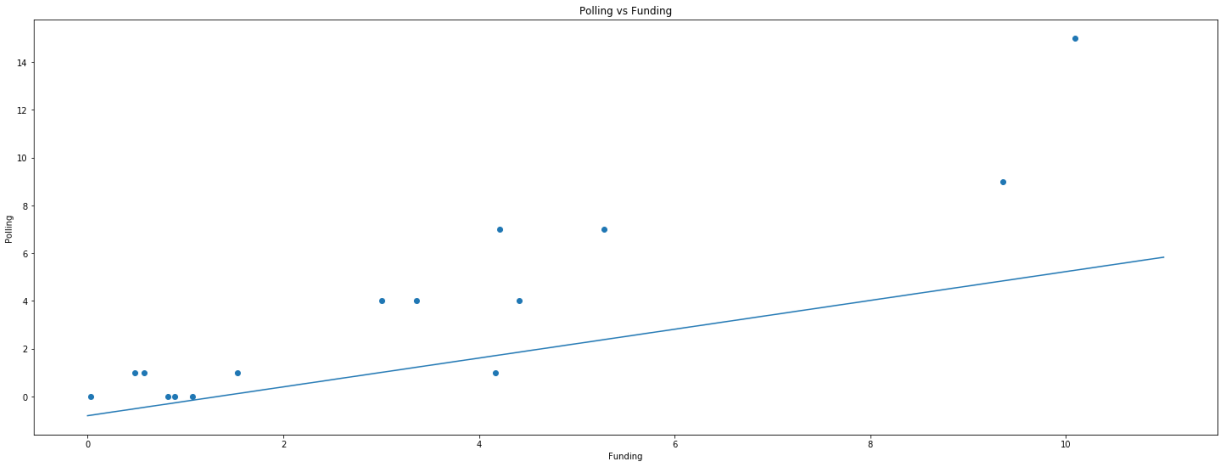

In [104]:
```
iowa = np.array([0,7,9,np.NaN, 4, 15,1,4,0,7,0,0,1,1,1,4])
```

In [105]:
```
newRec = []
for i in range(len(iowa)):
    if np.isnan(iowa[i]):
        newRec.append(i)

iowa = np.delete(iowa, newRec)
ratioList = np.delete(ratioList, newRec)
scatterPeople = np.delete(scatterPeople, newRec)
scatterFund = np.delete(scatterFund, newRec)
```

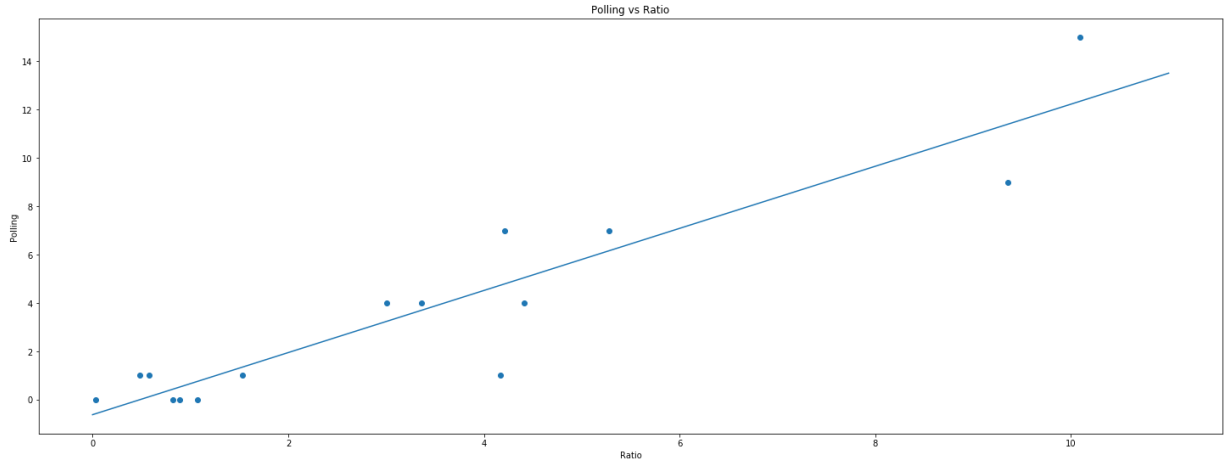
In [106]:
```
df = pd.DataFrame({"Funding over Truth":ratioList,"Polling":iowa})
ax = df.plot.bar(color=["SkyBlue","IndianRed"], rot=0, title="Compare")
ax.set_xlabel("Candidates")
ax.xaxis.set_major_formatter(plt.FixedFormatter(scatterPeople))
```



In [107]:
```
X = np.reshape(scatterFun, (-1,1))
y = iowa
model = sklm.LinearRegression().fit(X, y)
r_squared_noRatio = model.score(X,y)
plt.scatter(ratioList, iowa)
plt.xlabel('Funding', fontsize=10)
plt.ylabel('Polling', fontsize=10)
plt.title('Polling vs Funding')
line = np.linspace(0,11, 1000)
C = (line * model.coef_) + model.intercept_
plt.plot(line, C)
plt.show()
print('R Squared of ', r_squared_noRatio)

X = np.reshape(ratioList, (-1,1))
y = iowa
model = sklm.LinearRegression().fit(X, y)
r_squared_ratio = model.score(X,y)
plt.scatter(ratioList, iowa)
plt.xlabel('Ratio', fontsize=10)
plt.ylabel('Polling', fontsize=10)
plt.title('Polling vs Ratio')
line = np.linspace(0,11, 1000)
C = (line * model.coef_) + model.intercept_
plt.plot(line, C)
plt.show()
print('R Squared of ', r_squared_ratio)
```




```
R Squared of  0.7638249941458373
```



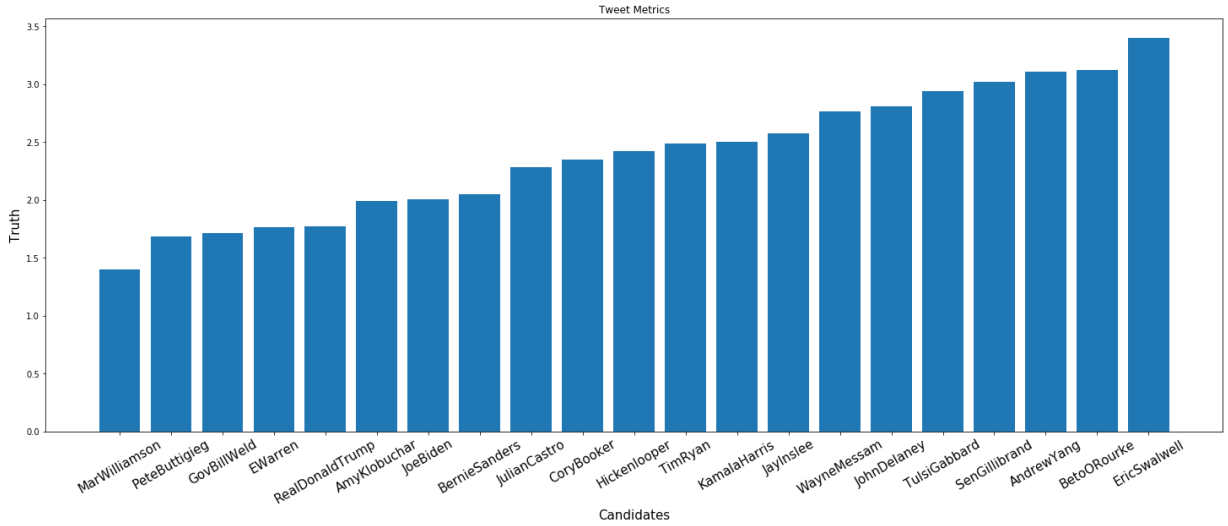

```
R Squared of  0.8571993106393145
```

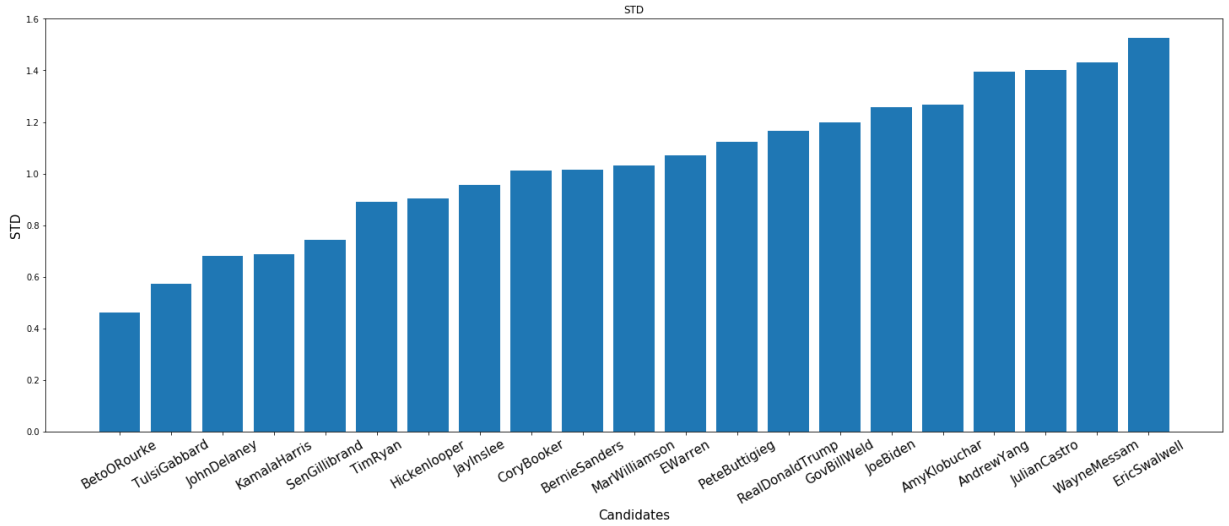
In [108]:
```
lazo = copy
lazo.hist(figsize = (20,20), layout = (7,3))
```

Out[108]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000001D78EAEE358>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001D78E9C6F28>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001D78EA354E0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000001D78EA5CA58>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001D78EA83FD0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001D78EF33588>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000001D78EF5BB00>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001D78F9BA0F0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001D78F9BA128>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000001D78EEDECC0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001D78EAEEAC8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001D790E1BE80>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000001D78EC0FB00>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001D78EB896A0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001D78EE425F8>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000001D78EC8D710>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001D78EDB83C8>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001D78F07B6A0>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000001D78E9586A0>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001D790DB5208>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001D7909CE860>]],
      dtype=object)
```


AmyKlobuchar
AndrewYang
BernieSanders
BetoORourke
CoryBooker
EWarren
EricSwalwell
GovBillWeld
Hickenlooper
JayInslee
JoeBiden
JohnDelaney
JulianCastro
KamalaHarris
MarWilliamson
PeteButtigieg
RealDonaldTrump
SenGillibrand
TimRyan
TulsiGabbard
WayneMessam

In [109]:
```
data = lazo.describe()
std = data.loc['std',:]

from operator import itemgetter
candidateScore = sorted(std.items(), key=itemgetter(1))

candidates = []
candidateScores = []
for i in range(len(candidateScore)):
    candidates.append(candidateScore[i][0])
    candidateScores.append(candidateScore[i][1])


index = np.arange(len(files))
plt.bar(index, sortScores)
plt.xlabel('Candidates', fontsize=15)
plt.ylabel('Truth', fontsize=15)
plt.xticks(index, sortPeople, fontsize=15, rotation=30)
plt.title('Tweet Metrics')
plt.rcParams["figure.figsize"] = [25,9]
plt.show()


index = np.arange(len(candidates))
plt.bar(index, candidateScores)
plt.xlabel('Candidates', fontsize=15)
plt.ylabel('STD', fontsize=15)
plt.xticks(index, candidates, fontsize=15, rotation=30)
plt.title('STD')
plt.rcParams["figure.figsize"] = [25,9]
plt.show()
```






In [110]:
```
pd.set_option('display.max_columns', 22)
data.T
```

Out[110]:

| | count | mean | std | min | 25% | 50% | 75% | max |
|---|---|---|---|---|---|---|---|---|
| **AmyKlobuchar** | 191.0 | 1.994764 | 1.266979 | 1.0 | 1.00 | 1.0 | 3.0 | 4.0 |
| **AndrewYang** | 152.0 | 3.105263 | 1.396106 | 0.0 | 1.75 | 4.0 | 4.0 | 5.0 |
| **BernieSanders** | 160.0 | 2.050000 | 1.014362 | 1.0 | 1.00 | 2.0 | 2.0 | 5.0 |
| **BetoORourke** | 97.0 | 3.123711 | 0.462280 | 2.0 | 3.00 | 3.0 | 3.0 | 5.0 |
| **CoryBooker** | 162.0 | 2.351852 | 1.012176 | 1.0 | 2.00 | 2.0 | 3.0 | 5.0 |
| **EricSwalwell** | 160.0 | 3.400000 | 1.526290 | 1.0 | 2.00 | 4.0 | 5.0 | 5.0 |
| **EWarren** | 173.0 | 1.763006 | 1.070979 | 1.0 | 1.00 | 1.0 | 3.0 | 5.0 |
| **GovBillWeld** | 146.0 | 1.712329 | 1.197462 | 0.0 | 1.00 | 1.0 | 2.0 | 5.0 |
| **Hickenlooper** | 151.0 | 2.423841 | 0.905075 | 1.0 | 2.00 | 3.0 | 3.0 | 5.0 |
| **JayInslee** | 127.0 | 2.574803 | 0.955516 | 1.0 | 2.00 | 3.0 | 3.0 | 4.0 |
| **JoeBiden** | 162.0 | 2.006173 | 1.258496 | 1.0 | 1.00 | 1.0 | 3.0 | 5.0 |
| **JohnDelaney** | 149.0 | 2.812081 | 0.681504 | 1.0 | 2.00 | 3.0 | 3.0 | 4.0 |
| **JulianCastro** | 153.0 | 2.281046 | 1.402336 | 1.0 | 1.00 | 2.0 | 4.0 | 5.0 |
| **KamalaHarris** | 190.0 | 2.500000 | 0.688146 | 1.0 | 2.00 | 3.0 | 3.0 | 3.0 |
| **MarWilliamson** | 189.0 | 1.402116 | 1.029974 | 0.0 | 1.00 | 1.0 | 1.0 | 5.0 |
| **PeteButtigieg** | 99.0 | 1.686869 | 1.121647 | 1.0 | 1.00 | 1.0 | 3.0 | 4.0 |
| **RealDonaldTrump** | 143.0 | 1.769231 | 1.166989 | 0.0 | 1.00 | 2.0 | 2.0 | 5.0 |
| **SenGillibrand** | 175.0 | 3.022857 | 0.742428 | 1.0 | 3.00 | 3.0 | 3.0 | 5.0 |
| **TimRyan** | 159.0 | 2.484277 | 0.891985 | 1.0 | 2.00 | 3.0 | 3.0 | 4.0 |
| **TulsiGabbard** | 192.0 | 2.942708 | 0.571440 | 1.0 | 3.00 | 3.0 | 3.0 | 4.0 |
| **WayneMessam** | 130.0 | 2.761538 | 1.429500 | 1.0 | 1.00 | 3.0 | 4.0 | 5.0 |