Analyzing Human Behaviour Complexity Data

In [11]:

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
%matplotlib inline
```

In [478]:

```
df = pd.read_csv('lifespan.csv', sep=' ')
df.dropna(how='any', inplace=True)
task = 'toss.RT'
```

1. Random sequences by Age

In [479]:

```
fix, ax = plt.subplots(1, 1, figsize=(4, 3))
ax.scatter(df['age'],df['toss.K'],alpha=0.1,c='g')
ax.set_xlabel('Age')
ax.set_ylabel('Mean Complexity')
ax.set_title('Ability to generate random sequencies by age')
ax.grid(True)
ax.set_frame_on(False)
```

2. Design a Binary Classifier

Add a column to dataset to define category for Binary Classification

1 for Category 1 (Age 15 to 65) 0 for Category 2 (All other ages)

In [480]:

```
category = []
for age in df['age']:
    if(age >= 15 and age <= 65):
        category.append(1)
    else:
        category.append(0)

df['category'] = category
df[[task,'age','category']]
```

Out[480]:

	toss.RT	age	category

	toss_RT	age	category
0	45.1	41.000000	1
1	9.1	42.000000	1
2	6.2	53.000000	1
3	16.3	53.000000	1
4	9.8	49.000000	1
...	...	...	...
3444	12.9	22.326027	1
3445	7.3	15.928767	1
3446	7.2	32.893151	1
3447	11.8	26.764384	1
3448	9.7	18.997260	1

3382 rows × 3 columns

Scatter plot to view data distribution

Cat 1 The person's ablility to generate sequence between age 15 and 65

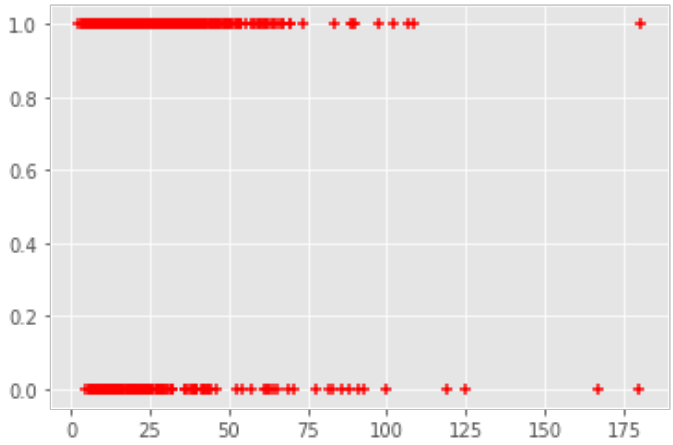
Cat 2 The person's ablility to generate sequence at all other ages

In [481]:

```
plt.scatter(df[task],df.category,marker='+',color='red')
```

Out[481]:

```
<matplotlib.collections.PathCollection at 0x179b8c02408>
```



Splitting the dataset into Testing and Training datasets

In [482]:

```
from sklearn.model_selection import train_test_split
```

In [483]:

```
X_train, X_test, y_train, y_test = train_test_split(df[[task]],df.category,test_size=0.1
)
```

In [484]:

```
X_train
```

Out[484]:

	toss.RT
2277	12.0
2229	13.1
2463	28.8
2987	17.5
183	10.0
...	...
1781	5.8
824	22.9
1462	8.1
2712	32.0
2158	9.0

3043 rows × 1 columns

In [485]:

```
X_test
```

Out[485]:

	toss.RT
1176	10.1
264	6.3
912	31.6
1026	29.0
3115	13.6
...	...
1278	13.4
3445	7.3
345	13.0
590	5.0
1437	24.5

339 rows × 1 columns

Train the model using logistic regression

In [486]:

```
from sklearn.linear_model import LogisticRegression
```

In [487]:

```
model = LogisticRegression()
```

In [488]:

```
model.fit(X_train,y_train)
```

```
C:\Users\cinni\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:432: FutureWa
rning: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence thi
s warning.
  FutureWarning)
```

Out[488]:

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='warn', n_jobs=None, penalty='l2',
                   random_state=None, solver='warn', tol=0.0001, verbose=0,
                   warm_start=False)
```

In [489]:

```
model.predict(X_test)
```

Out[489]:

```
array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 1, 1, 1, 1, 1], dtype=int64)
```

Score

In [490]:

```
model.score(X_test,y_test)
```

Out[490]:

```
0.9498525073746312
```

Check Probability of person's age being in Category 1 (1) and 2 (0)

In [491]:

```
lr_probs = model.predict_proba(X_test)
lr_probs
```

Out[491]:

```
array([[0.03914743, 0.96085257],
       [0.03395748, 0.96604252],
       [0.08585652, 0.91414348],
       [0.07825412, 0.92174588],
       [0.04459442, 0.95540558],
       [0.04186396, 0.95813604],
       [0.05772486, 0.94227514],
       [0.09346175, 0.90653825],
       [0.10860737, 0.89139263],
       [0.04627904, 0.95372096],
       [0.03552031, 0.96447969],
       [0.0672664 , 0.9327336 ],
       [0.07468831, 0.92531169],
       [0.04361187, 0.95638813],
       [0.04124512, 0.95875488],
       [0.04170842, 0.95829158],
       [0.06232539, 0.93767461],
       [0.04442921, 0.95557079],
       [0.0592218 , 0.9407782 ],
       [0.06232539, 0.93767461],
       [0.03842335, 0.96157665],
```

```
[0.03958814, 0.96041186],
[0.04202005, 0.95797995],
[0.08110236, 0.91889764],
[0.04679613, 0.95320387],
[0.03209708, 0.96790292],
[0.04410053, 0.95589947],
[0.05965614, 0.94034386],
[0.05965614, 0.94034386],
[0.04459442, 0.95540558],
[0.04328897, 0.95671103],
[0.06775549, 0.93224451],
[0.05094676, 0.94905324],
[0.05094676, 0.94905324],
[0.05364141, 0.94635859],
[0.03605648, 0.96394352],
[0.06323965, 0.93676035],
[0.05113491, 0.94886509],
[0.03771214, 0.96228786],
[0.08616189, 0.91383811],
[0.0592218 , 0.9407782 ],
[0.03757142, 0.96242858],
[0.08616189, 0.91383811],
[0.05463597, 0.94536403],
[0.06899267, 0.93100733],
[0.05857585, 0.94142415],
[0.03828009, 0.96171991],
[0.05227768, 0.94772232],
[0.0594386 , 0.9405614 ],
[0.03473042, 0.96526958],
[0.07714067, 0.92285933],
[0.96768416, 0.03231584],
[0.05423608, 0.94576392],
[0.05688548, 0.94311452],
[0.03525511, 0.96474489],
[0.04476021, 0.95523979],
[0.03345143, 0.96654857],
[0.05266394, 0.94733606],
[0.0324611 , 0.9675389 ],
[0.03592171, 0.96407829],
[0.03828009, 0.96171991],
[0.06974498, 0.93025502],
[0.04280889, 0.95719111],
[0.0518941 , 0.9481059 ],
[0.05247047, 0.94752953],
[0.04593734, 0.95406266],
[0.04891991, 0.95108009],
[0.15432771, 0.84567229],
[0.03592171, 0.96407829],
[0.35750942, 0.64249058],
[0.05483696, 0.94516304],
[0.23079416, 0.76920584],
[0.03383028, 0.96616972],
[0.031737  , 0.968263  ],
[0.04610789, 0.95389211],
[0.03799509, 0.96200491],
[0.03447091, 0.96552909],
[0.06899267, 0.93100733],
[0.04542929, 0.95457071],
[0.04928265, 0.95071735],
[0.05364141, 0.94635859],
[0.04377418, 0.95622582],
[0.03757142, 0.96242858],
[0.03408514, 0.96591486],
[0.03988459, 0.96011541],
[0.04377418, 0.95622582],
[0.04233391, 0.95766609],
[0.07549769, 0.92450231],
[0.03673766, 0.96326234],
[0.04731871, 0.95268129],
[0.45843116, 0.54156884],
[0.04767017, 0.95232983],
[0.03715229, 0.96284771],
```

```
[0.03473042, 0.96526958],
[0.06949337, 0.93050663],
[0.06899267, 0.93100733],
[0.03701358, 0.96298642],
[0.04526114, 0.95473886],
[0.04186396, 0.95813604],
[0.04109178, 0.95890822],
[0.04361187, 0.95638813],
[0.05247047, 0.94752953],
[0.0471439 , 0.9528561 ],
[0.04312837, 0.95687163],
[0.04377418, 0.95622582],
[0.05266394, 0.94733606],
[0.09215354, 0.90784646],
[0.04377418, 0.95622582],
[0.03771214, 0.96228786],
[0.06558029, 0.93441971],
[0.0518941 , 0.9481059 ],
[0.03885624, 0.96114376],
[0.05038623, 0.94961377],
[0.10860737, 0.89139263],
[0.03578743, 0.96421257],
[0.04018316, 0.95981684],
[0.12845338, 0.87154662],
[0.04233391, 0.95766609],
[0.04410053, 0.95589947],
[0.05564789, 0.94435211],
[0.09680537, 0.90319463],
[0.04063504, 0.95936496],
[0.05001579, 0.94998421],
[0.03660044, 0.96339956],
[0.04983155, 0.95016845],
[0.04280889, 0.95719111],
[0.06393354, 0.93606646],
[0.05208556, 0.94791444],
[0.04003361, 0.95996639],
[0.04983155, 0.95016845],
[0.03929381, 0.96070619],
[0.04249167, 0.95750833],
[0.04361187, 0.95638813],
[0.04361187, 0.95638813],
[0.05001579, 0.94998421],
[0.03871143, 0.96128857],
[0.04361187, 0.95638813],
[0.04233391, 0.95766609],
[0.03221798, 0.96778202],
[0.08927008, 0.91072992],
[0.03408514, 0.96591486],
[0.0372915 , 0.9627085 ],
[0.04328897, 0.95671103],
[0.0421767 , 0.9578233 ],
[0.04696971, 0.95303029],
[0.03197662, 0.96802338],
[0.04838055, 0.95161945],
[0.09612821, 0.90387179],
[0.06278097, 0.93721903],
[0.05094676, 0.94905324],
[0.03813733, 0.96186267],
[0.05324838, 0.94675162],
[0.03958814, 0.96041186],
[0.05038623, 0.94961377],
[0.07127241, 0.92872759],
[0.11088445, 0.88911555],
[0.04873949, 0.95126051],
[0.09782909, 0.90217091],
[0.05285808, 0.94714192],
[0.05879043, 0.94120957],
[0.04345013, 0.95654987],
[0.03701358, 0.96298642],
[0.07631512, 0.92368488],
[0.03605648, 0.96394352],
[0.03460043, 0.96539957],
```

```
[0.04155344, 0.95844656],
[0.03383028, 0.96616972],
[0.04361187, 0.95638813],
[0.05751393, 0.94248607],
[0.04361187, 0.95638813],
[0.04542929, 0.95457071],
[0.04078675, 0.95921325],
[0.0457674 , 0.9542326 ],
[0.04542929, 0.95457071],
[0.10600276, 0.89399724],
[0.03525511, 0.96474489],
[0.03578743, 0.96421257],
[0.03813733, 0.96186267],
[0.0421767 , 0.9578233 ],
[0.04186396, 0.95813604],
[0.041399  , 0.958601  ],
[0.07050487, 0.92949513],
[0.12416692, 0.87583308],
[0.07362154, 0.92637846],
[0.03799509, 0.96200491],
[0.05585238, 0.94414762],
[0.06278097, 0.93721903],
[0.04063504, 0.95936496],
[0.04249167, 0.95750833],
[0.0372915 , 0.9627085 ],
[0.06031324, 0.93968676],
[0.03619174, 0.96380826],
[0.09714554, 0.90285446],
[0.06463452, 0.93536548],
[0.04109178, 0.95890822],
[0.0471439 , 0.9528561 ],
[0.05266394, 0.94733606],
[0.06581873, 0.93418127],
[0.06629804, 0.93370196],
[0.03701358, 0.96298642],
[0.04063504, 0.95936496],
[0.04509357, 0.95490643],
[0.03282911, 0.96717089],
[0.0594386 , 0.9405614 ],
[0.03552031, 0.96447969],
[0.04063504, 0.95936496],
[0.03842335, 0.96157665],
[0.0397361 , 0.9602639 ],
[0.03743121, 0.96256879],
[0.04093899, 0.95906101],
[0.03701358, 0.96298642],
[0.04542929, 0.95457071],
[0.03988459, 0.96011541],
[0.03605648, 0.96394352],
[0.04393707, 0.95606293],
[0.04249167, 0.95750833],
[0.05344455, 0.94655545],
[0.07825412, 0.92174588],
[0.05132371, 0.94867629],
[0.05020069, 0.94979931],
[0.09851692, 0.90148308],
[0.03885624, 0.96114376],
[0.14019091, 0.85980909],
[0.0471439 , 0.9528561 ],
[0.05364141, 0.94635859],
[0.03988459, 0.96011541],
[0.04202005, 0.95797995],
[0.04946498, 0.95053502],
[0.04345013, 0.95654987],
[0.04662316, 0.95337684],
[0.0324611 , 0.9675389 ],
[0.04018316, 0.95981684],
[0.03282911, 0.96717089],
[0.04328897, 0.95671103],
[0.05227768, 0.94772232],
[0.03757142, 0.96242858],
[0.03332606, 0.96667394],
```

```
[0.03221798, 0.96778202],
[0.03900157, 0.96099843],
[0.10860737, 0.89139263],
[0.03673766, 0.96326234],
[0.04170842, 0.95829158],
[0.04509357, 0.95490643],
[0.06053377, 0.93946623],
[0.05038623, 0.94961377],
[0.03370354, 0.96629646],
[0.03828009, 0.96171991],
[0.0372915 , 0.9627085 ],
[0.03944071, 0.96055929],
[0.06097711, 0.93902289],
[0.03842335, 0.96157665],
[0.06800127, 0.93199873],
[0.05364141, 0.94635859],
[0.05605758, 0.94394242],
[0.03512322, 0.96487678],
[0.05266394, 0.94733606],
[0.05626349, 0.94373651],
[0.17968543, 0.82031457],
[0.04063504, 0.95936496],
[0.05132371, 0.94867629],
[0.05667743, 0.94332257],
[0.07362154, 0.92637846],
[0.08555213, 0.91444787],
[0.0464508 , 0.9535492 ],
[0.04233391, 0.95766609],
[0.06053377, 0.93946623],
[0.05170331, 0.94829669],
[0.12000381, 0.87999619],
[0.05038623, 0.94961377],
[0.0464508 , 0.9535492 ],
[0.07549769, 0.92450231],
[0.03421326, 0.96578674],
[0.03383028, 0.96616972],
[0.05151318, 0.94848682],
[0.06278097, 0.93721903],
[0.04109178, 0.95890822],
[0.03512322, 0.96487678],
[0.07204766, 0.92795234],
[0.03578743, 0.96421257],
[0.03632748, 0.96367252],
[0.06209876, 0.93790124],
[0.03320115, 0.96679885],
[0.05324838, 0.94675162],
[0.06800127, 0.93199873],
[0.05730372, 0.94269628],
[0.03383028, 0.96616972],
[0.03885624, 0.96114376],
[0.10860737, 0.89139263],
[0.05423608, 0.94576392],
[0.06097711, 0.93902289],
[0.03944071, 0.96055929],
[0.04233391, 0.95766609],
[0.03646372, 0.96353628],
[0.0421767 , 0.9578233 ],
[0.0324611 , 0.9675389 ],
[0.04393707, 0.95606293],
[0.0723078 , 0.9276922 ],
[0.04018316, 0.95981684],
[0.04983155, 0.95016845],
[0.04855971, 0.95144029],
[0.04426458, 0.95573542],
[0.07797439, 0.92202561],
[0.04063504, 0.95936496],
[0.03871143, 0.96128857],
[0.0518941 , 0.9481059 ],
[0.04233391, 0.95766609],
[0.03632748, 0.96367252],
[0.05305289, 0.94694711],
[0.08374626, 0.91625374],
```

```
       [0.05544411, 0.94455589],
       [0.04155344, 0.95844656],
       [0.11556417, 0.88443583],
       [0.03499181, 0.96500819],
       [0.03757142, 0.96242858],
       [0.03944071, 0.96055929],
       [0.03771214, 0.96228786],
       [0.10674126, 0.89325874],
       [0.03885624, 0.96114376],
       [0.04393707, 0.95606293],
       [0.04983155, 0.95016845],
       [0.04361187, 0.95638813],
       [0.09714554, 0.90285446],
       [0.03332606, 0.96667394],
       [0.05564789, 0.94435211],
       [0.03701358, 0.96298642],
       [0.04593734, 0.95406266],
       [0.04983155, 0.95016845],
       [0.03565363, 0.96434637],
       [0.03885624, 0.96114376],
       [0.06370146, 0.93629854],
       [0.05443568, 0.94556432],
       [0.03771214, 0.96228786],
       [0.06119991, 0.93880009],
       [0.07604175, 0.92395825],
       [0.04426458, 0.95573542],
       [0.03525511, 0.96474489],
       [0.04361187, 0.95638813],
       [0.03233932, 0.96766068],
       [0.06653891, 0.93346109]])
```

3. ROC Curve

In [492]:

```
# roc curve and auc
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score
```

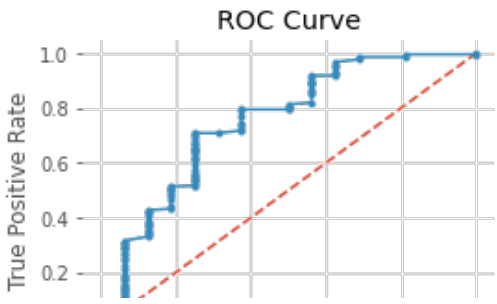
In [493]:

```
# generate a Cat 1 prediction (majority class)
ns_probs = [0 for _ in range(len(y_test))]

# keep probabilities for the positive outcome only
lr_probs = lr_probs[:, 1]

# calculate roc curves
ns_fpr, ns_tpr, _ = roc_curve(y_test, ns_probs)
lr_fpr, lr_tpr, _ = roc_curve(y_test, lr_probs)

# plot the roc curve for the model
fix, ax = plt.subplots(1, 1, figsize=(4, 3))
ax.plot(ns_fpr, ns_tpr, linestyle='--')
ax.plot(lr_fpr, lr_tpr, marker='.')
ax.set_xlabel('False Positive Rate')
ax.set_ylabel('True Positive Rate')
ax.set_title('ROC Curve')
ax.grid(True)
ax.set_frame_on(False)
```



In [ ]: