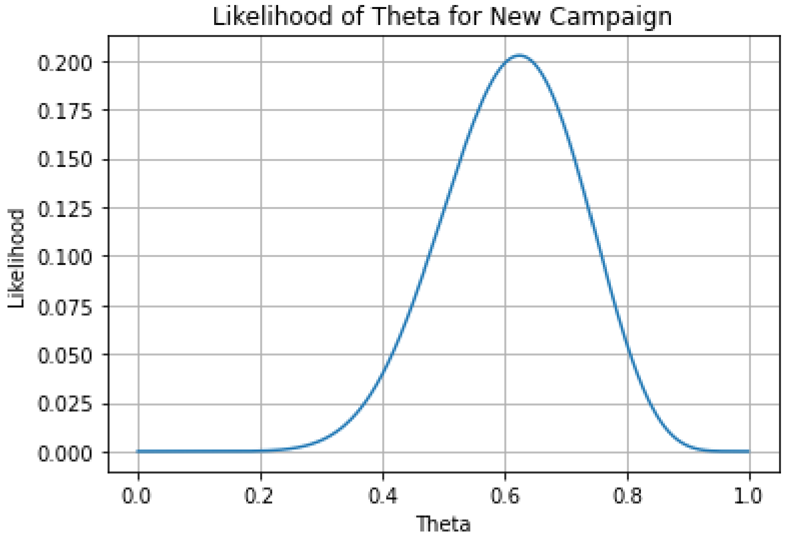
In [3]:

```
import math
from scipy import stats
from scipy.special import factorial
from matplotlib import pyplot as plt
import pandas as pd
import numpy as np

def likelihood(theta, n, x):
    return (factorial(n) / (factorial(x) * factorial(n - x))) * (theta ** x) * ((1 - theta)

def pprob(prior, posterior, n_occured, n_events):
    return pd.Series(map(lambda theta: likelihood(theta, n_events, n_occured), prior))

def generative_model(n_events, p):
    return np.random.binomial(n_events, p)

def ABC(n_occured, n_events, n_draws=1000):
    prior = pd.Series(sorted(np.random.uniform(0, 1, size=n_draws)))
    sim_data = [generative_model(n_events ,p) for p in prior]
    posterior = prior[list(map(lambda x: x == n_occured, sim_data))]
    posterior_probability = pprob(prior, posterior, n_occured, n_events)

    # Let's see what we got
    f, ax = plt.subplots(1)
    ax.plot(prior, posterior_probability)
    ax.set_xlabel("Theta")
    ax.set_ylabel("Likelihood")
    ax.grid()
    ax.set_title("Likelihood of Theta for New Campaign")
    plt.show()

ABC(10, 16)
```



In [4]:

```
# example of generating a small classification dataset
from sklearn.datasets import make_blobs

# generate 2d classification dataset

X, y = make_blobs(n_samples=100, centers=2, n_features=2, random_state=1)

# summarize
print(X.shape, y.shape)

print(X[:5])

print(y[:5])
```

```
(100, 2) (100,)
[[-0.79415228  2.10495117]
 [-9.15155186 -4.81286449]
 [-3.10367371  3.90202401]
 [-1.42946517  5.16850105]
 [-7.4693868  -4.20198333]]
[0 1 0 0 1]
```

In [5]:

```
# fit a probability distribution to a univariate data sample
def fit_distribution(data):
    # estimate parameters
    mu = mean(data)
    sigma = std(data)
    print(mu, sigma)
    # fit distribution
    dist = norm(mu, sigma)
    return dist
```

In [6]:

```
# sort data into classes
Xy0 = X[y == 0]
Xy1 = X[y == 1]
print(Xy0.shape, Xy1.shape)
```

```
(50, 2) (50, 2)
```

In [7]:

```
# calculate priors
prioryθ = len(Xy0) / len(X)
priory1 = len(Xy1) / len(X)
print(priory0, priory1)
```

```
0.5 0.5
```

In [9]:

```python
# summarize probability distributions of the dataset
from sklearn.datasets import make_blobs
from scipy.stats import norm
from numpy import mean
from numpy import std

# fit a probability distribution to a univariate data sample
def fit_distribution(data):
    # estimate parameters
    mu = mean(data)
    sigma = std(data)
    print(mu, sigma)
    # fit distribution
    dist = norm(mu, sigma)
    return dist

# generate 2d classification dataset
X, y = make_blobs(n_samples=100, centers=2, n_features=2, random_state=1)
# sort data into classes
Xy0 = X[y == 0]
Xy1 = X[y == 1]
print(Xy0.shape, Xy1.shape)

# calculate priors
priory0 = len(Xy0) / len(X)
priory1 = len(Xy1) / len(X)
print(priory0, priory1)

# create PDFs for y==0
X1y0 = fit_distribution(Xy0[:, 0])
X2y0 = fit_distribution(Xy0[:, 1])

# create PDFs for y==1
X1y1 = fit_distribution(Xy1[:, 0])
X2y1 = fit_distribution(Xy1[:, 1])
```

```
(50, 2) (50, 2)
0.5 0.5
-1.5632888906409914 0.787444265443213
4.426680361487157 0.958296071258367
-9.681177100524485 0.8943078901048118
-3.9713794295185845 0.9308177595208521
```