In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
% matplotlib inline

np.random.seed(42)
```

In [2]:
```
df = pd.read_csv('classroom_actions.csv')
df.head()
```

Out[2]:

	timestamp	id	group	total_days	completed
0	2015-08-10 17:06:01.032740	610019	experiment	97	True
1	2015-08-10 17:15:28.950975	690224	control	75	False
2	2015-08-10 17:34:40.920384	564994	experiment	128	True
3	2015-08-10 17:50:39.847374	849588	experiment	66	False
4	2015-08-10 19:10:40.650599	849826	experiment	34	False

In [5]:
```
# Create dataframe with all control records
control_df = df.query('group == "control"')

# Compute completion rate
control_ctr = control_df['completed'].mean()

# Display completion rate
control_ctr
```

Out[5]:
```
0.37199519230769229
```

In [6]:
```
# Create dataframe with all experiment records
experiment_df = df.query('group == "experiment"')

# Compute completion rate
experiment_ctr = experiment_df['completed'].mean()

# Display completion rate
experiment_ctr
```

Out[6]:
```
0.39353348729792148
```

In [7]:
```
# Compute observed difference in completion rates
obs_diff = experiment_ctr - control_ctr

# Display observed difference in completion rates
obs_diff
```

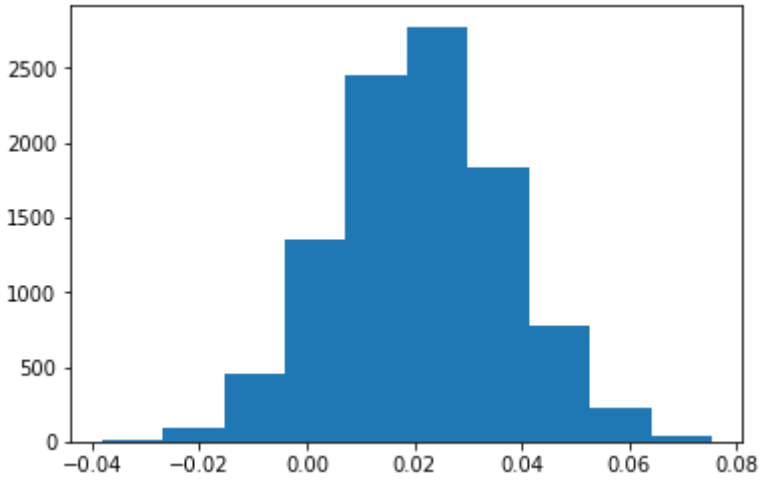
Out[7]:
```
0.02153829499022919
```

In [8]:
```
# Create sampling distribution for difference in completion rates
# with boostrapping
diffs = []
size = df.shape[0]
for _ in range(10000):

    boot = df.sample(size, replace=True)
    control_df = boot.query('group == "control"')
    experiment_df = boot.query('group == "experiment"')
    control_ctr = control_df['completed'].mean()
    experiment_ctr = experiment_df['completed'].mean()
    diffs.append(experiment_ctr - control_ctr)
```

In [9]:
```
# convert to numpy array
diffs = np.array(diffs)
```

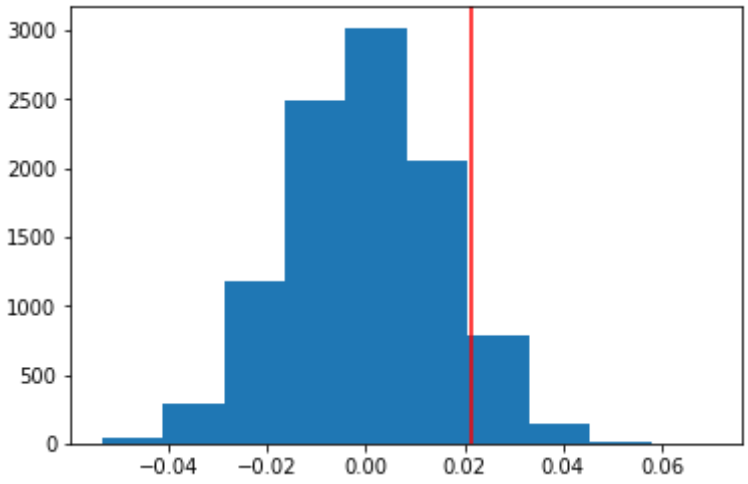
In [10]:
```
# plot distribution
plt.hist(diffs);
```



In [11]:
```
# create distribution under the null hypothesis
null_vals = np.random.normal(0, diffs.std(), diffs.size)
```

In [12]:
```
# plot null distribution
plt.hist(null_vals);

# plot line for observed statistic
plt.axvline(obs_diff, c='r');
```



In [13]:
```
# compute p value
(null_vals > obs_diff).mean()
```

Out[13]: 0.084599999999999995

In [ ]:
```

```