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 [3]:
```
# The total_days represents the total amount of time
# each student has spent in classroom.
# get the average classroom time for control group
control_mean = df.query('group == "control"').total_days.mean()

# get the average classroom time for experiment group
experiment_mean = df.query('group == "experiment"').total_days.mean()

# display average classroom time for each group
control_mean, experiment_mean
```

Out[3]:
```
(73.368990384615387, 74.671593533487297)
```

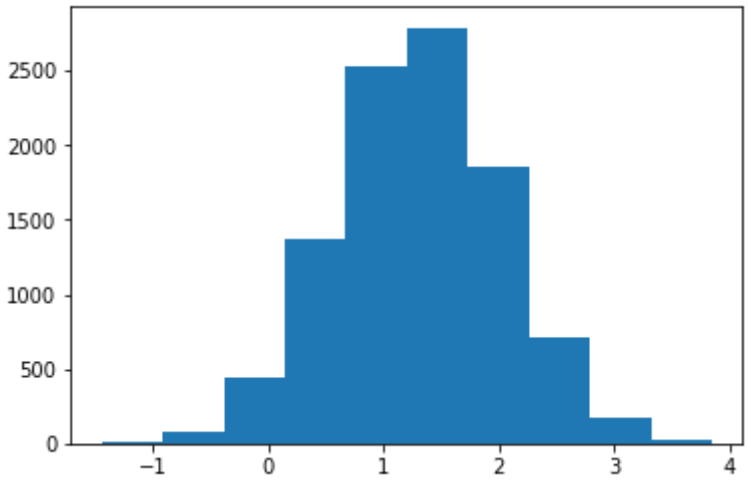
In [4]:
```
# compute observed difference in classroom time
obs_diff = experiment_mean - control_mean

# display observed difference
obs_diff
```

Out[4]:
```
1.3026031488719099
```

In [9]:
```
# create sampling distribution of difference in average classroom times
# with boostrapping
diffs = []
size = df.shape[0]
for _ in range(10000):
    bootsamp = df.sample(size, replace=True)
    control_mean = bootsamp.query('group == "control"').total_days.mean
()
    experiment_mean = bootsamp.query('group == "experiment"').total_days
.mean()
    diffs.append(experiment_mean - control_mean)
```

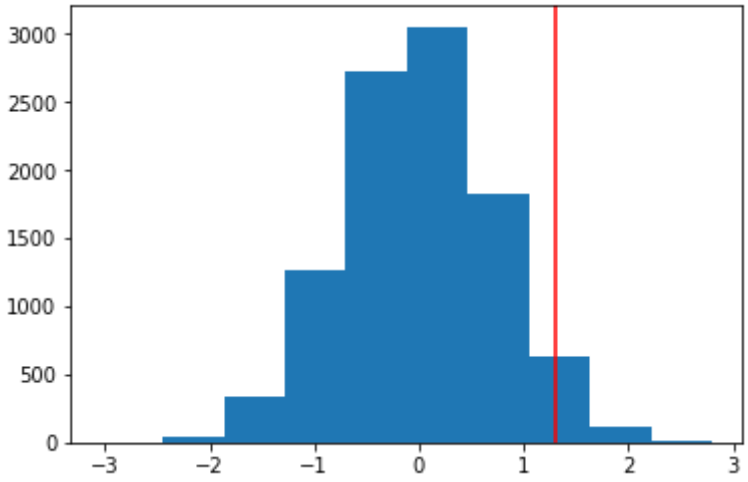
In [10]:
```
# convert to numpy array

diffs = np.array(diffs)
```

In [11]:
```
# plot sampling distribution

plt.hist(diffs);
```



In [12]:
```
# simulate distribution under the null hypothesis
null_vals = np.random.normal(0, diffs.std(), diffs.size)
```

In [13]:
```
# plot null distribution
plt.hist(null_vals);

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



In [14]:
```
# compute p value
(null_vals > obs_diff).mean()
```

Out[14]: 0.035099999999999999

In [ ]: