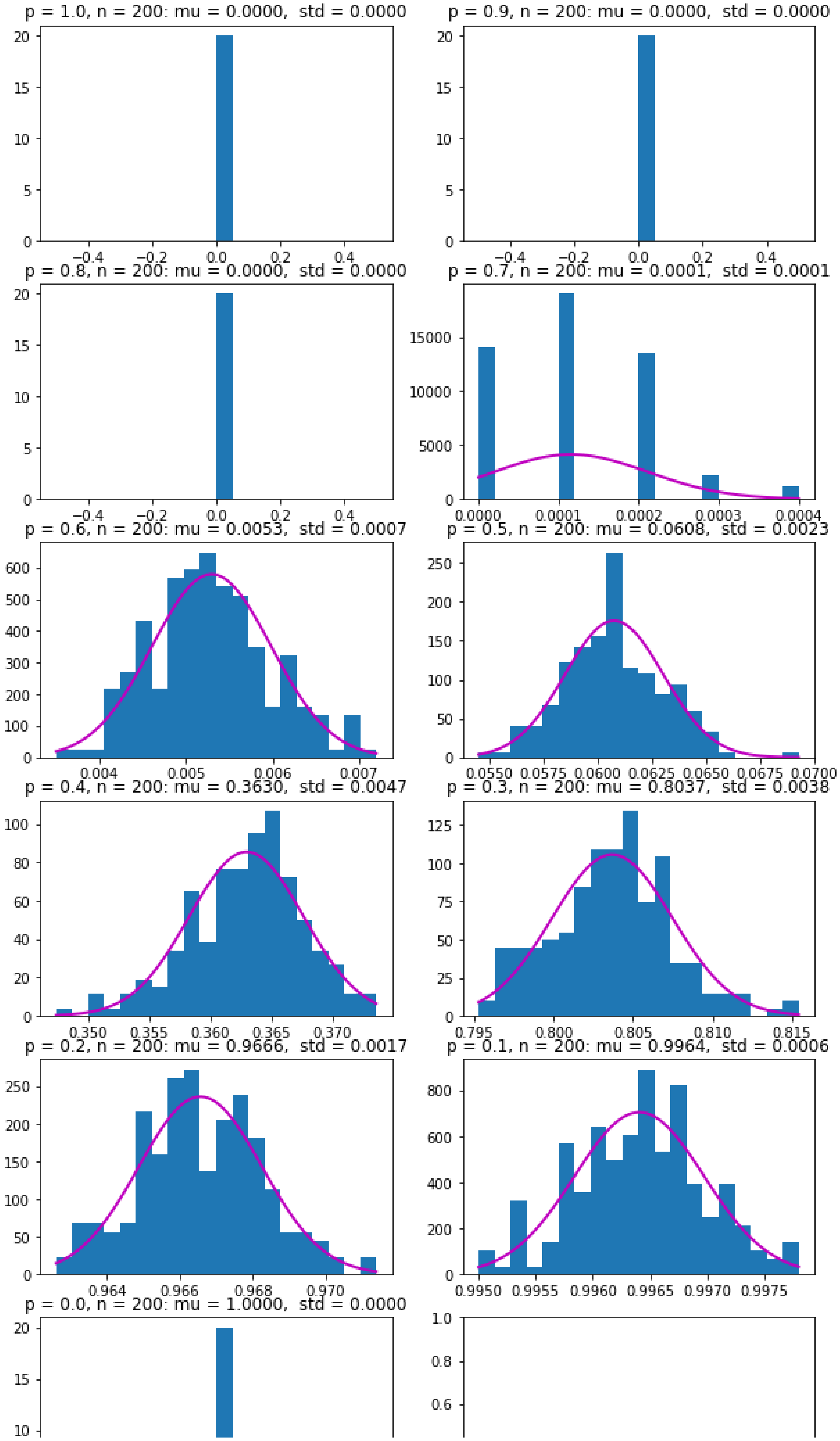
```
import pandas as pd

df = pd.read_csv("runs.csv")
df.head()
```

	Unnamed: 0	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
0	0	0.0	0.0	0.0	0.0003	0.0053	0.0604	0.3622	0.8093	0.9663	0.9952
1	1	0.0	0.0	0.0	0.0001	0.0068	0.0606	0.3649	0.8058	0.9680	0.9963
2	2	0.0	0.0	0.0	0.0000	0.0053	0.0590	0.3694	0.7968	0.9626	0.9966
3	3	0.0	0.0	0.0	0.0002	0.0048	0.0596	0.3691	0.8070	0.9666	0.9967
4	4	0.0	0.0	0.0	0.0001	0.0057	0.0598	0.3569	0.7978	0.9659	0.9966

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import norm

distns = []

plt_rows = 6
plt_cols = 2
fig, a = plt.subplots(plt_rows, plt_cols)
fig.set_figwidth(10)
fig.set_figheight(20)

for i in range(1, 12):
  c = df.columns[i]
  runs = df[c]
  mu, std = norm.fit(runs)
  distns.append((mu, std))

  row = (i - 1) // plt_cols
  col = (i - 1) % plt_cols

  r = (min(runs), max(runs))

  a[row][col].hist(runs, bins=20, density=True, range=r)
  if (r[1] > r[0]):
    x = np.linspace(r[0], r[1], 100)
    pn = norm.pdf(x, mu, std)
    a[row][col].plot(x, pn, 'm', linewidth=2)
  title = "p = {:s}, n = {:d}: mu = {:.4f},  std = {:.4f}".format(c, len(df.inde
  a[row][col].set_title(title)

plt.show()
```



p = 1.0, n = 200: mu = 0.0000, std = 0.0000

p = 0.9, n = 200: mu = 0.0000, std = 0.0000

p = 0.8, n = 200: mu = 0.0000, std = 0.0000

p = 0.7, n = 200: mu = 0.0001, std = 0.0001

p = 0.6, n = 200: mu = 0.0053, std = 0.0007

p = 0.5, n = 200: mu = 0.0608, std = 0.0023

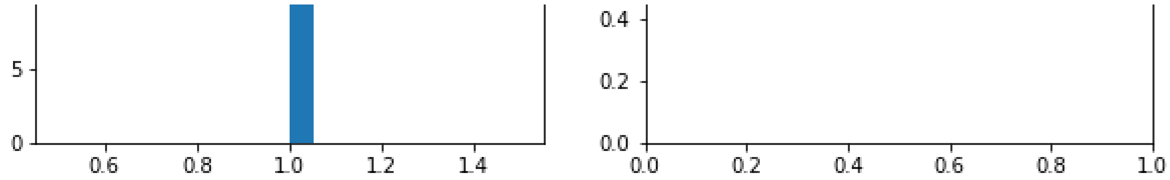
p = 0.4, n = 200: mu = 0.3630, std = 0.0047

p = 0.3, n = 200: mu = 0.8037, std = 0.0038

p = 0.2, n = 200: mu = 0.9666, std = 0.0017

p = 0.1, n = 200: mu = 0.9964, std = 0.0006

p = 0.0, n = 200: mu = 1.0000, std = 0.0000



```
# test null hypothesis that a result came from this distribution
def p_val_result_from_distn(result, distn):
  p_val = 1.0
  if distn[1] == 0.0:
    if result != distn[0]:
      p_val = 0.0
  else:
    z = result - distn[0]
    p_val = 2 * norm.sf(abs(z), scale=distn[1])
  return p_val

def accept_result_from_distn(result, distn, sig=0.05):
  return p_val_result_from_distn(result, distn) > sig


# check for runs that fail null hypothesis
count = 0
for i in df.index:
  rs = df.loc[i].to_list()[1:]
  accepts = []
  for j, r in enumerate(rs):
    accepts.append(accept_result_from_distn(r, distns[j], sig=0.0015))
  if sum(accepts) < 11:
    count = count + 1
    print('reject {}'.format(i))
    print(accepts)

print('reject count {}'.format(count))
```

```
reject 6
[True, True, True, True, True, False, True, True, True, True, True]
reject 126
[True, True, True, True, True, True, False, True, True, True, True]
reject count 2
```

```
# compare to example results in problem statement

prob_res = [0.000, 0.000, 0.000, 0.000, 0.011, 0.074, 0.625, 0.940, 1.000, 1.000,
accepts = [accept_result_from_distn(r, distns[i], sig=0.01) for i, r in enumerate
print("Accept problem results came from same distribution as my results\n(unless
for i, a in enumerate(accepts):
  print("{} {}".format(df.columns[i + 1], a))
```

```
Accept problem results came from same distribution as my results
(unless 99% chance that's wrong):
1.0 True
0.9 True
0.8 True
0.7 True
0.6 False
0.5 False
0.4 False
0.3 False
0.2 False
0.1 False
0.0 True
```