Задание 1.
Исходные данные:Прежде чем приступить к формальному описанию процесса проверки гипотез, давайте подумаем, как бы мы подошли к проверке такой гипотезы, если бы мы совсем ничего про это всё не знали. Вот у нас есть выборка из значений диаметра подшипника.
Решение:

```
Python 3.8.5 (default, May 27 2021, 13:30:53)

[GCC 9.3.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

>>> import numpy as np

>>> samples = np.array([0.6603, 0.9466, 0.5968, 1.3792,1.5481, 0.7515, 1.0681, 1.1134, 1.2088, 1.7010, 1.0282, 1.3579, 1.0191, 1.1784, 1.1168, 1.1372, 0.7273, 1.3958, 0.8665, 1.5112, 1.1610, 1.0232, 1.0865, 1.0200])

>>>

>>> samples.mean()

1.1084541666666665

>>>

>>> samples.std(ddof=1)

0.27936526343958135

>>>

>>> samples.shape

(24,)

>>>

>>> samples.std(ddof=1) / np.sqrt(samples.shape[0])

0.057025195607097914

>>>

>>> def statistic(samples: np.ndarray) -> float:

...     return (samples.mean() - 1) / (0.25 / np.sqrt(samples.shape[0]))

...

>>> alpha = 0.05

>>> import scipy

>>> from scipy import stats

>>> t1 = stats.norm.ppf(alpha / 2)

>>> t2 = stats.norm.ppf(1 - alpha / 2)

>>> t1

-1.9599639845400545
```

>>>

>>> t2

1.959963984540054

>>>

>>> s = statistic(samples)

>>> print(s)

2.1252589504967747

>>>

Задание 2.

Исходные данные:

В реальности у нас редко есть что-то кроме самой выборки. В частности, дисперсию случайной величины мы скорее всего не знаем. Как мы уже отмечали ранее, в таком случае мы тоже можем провести статистический тест, однако, нам нужно будет для этого взять другую статистику.

Решение:

Python 3.8.5 (default, May 27 2021, 13:30:53)

[GCC 9.3.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

>>> import numpy as np

>>> samples = np.array([0.6603, 0.9466, 0.5968, 1.3792,1.5481, 0.7515, 1.0681, 1.1134, 1.2088, 1.7010, 1.0282, 1.3579, 1.0191, 1.1784, 1.1168, 1.1372, 0.7273, 1.3958, 0.8665, 1.5112, 1.1610, 1.0232, 1.0865, 1.0200])

>>> samples.mean()

1.1084541666666665

>>>

>>> samples.std(ddof=1)

0.27936526343958135

>>>

>>> samples.shape

(24,)

>>> def statistic(samples: np.ndarray) -> float:

... return (samples.mean() - 1) / (samples.std(ddof=1) / np.sqrt(samples.shape[0]))

...

>>> n = samples.shape[0]

>>>

>>> import scipy

```
>>> from scipy import stats
>>> alpha = 0.05
>>> t1 = stats.t.ppf(alpha / 2, df=n - 1)
>>> t2 = stats.t.ppf(1 - alpha / 2, df=n - 1)
>>> t1
-2.068657610419041
>>>
>>> t2
2.0686576104190406
>>>
>>> s = statistic(samples)
>>> print(s)
1.9018640008517087
>>>
>>>
```

Задание 3.

Исходные данные:

В задании 2 мы проверяли гипотезу о математическом ожидании диаметра подшипника. Реализуем этот тест для различных α и посмотрим, как это влияет на результат.

Решение:

```
Python 3.8.5 (default, May 27 2021, 13:30:53)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import numpy as np
>>> samples = np.array([0.6603, 0.9466, 0.5968, 1.3792,1.5481, 0.7515, 1.0681, 1.1134, 1.2088, 1.7010, 1.0282, 1.3579, 1.0191, 1.1784, 1.1168, 1.1372, 0.7273, 1.3958, 0.8665, 1.5112, 1.1610, 1.0232, 1.0865, 1.0200])
>>> samples.mean()
1.1084541666666665
>>>
>>> samples.std(ddof=1)
0.27936526343958135
>>>
>>> samples.shape
(24,)
>>>
```

```
>>> def statistic(samples: np.ndarray) -> float:
...     return (samples.mean() - 1) / (samples.std(ddof=1) np.sqrt(samples.shape[0]))
  File "<stdin>", line 2
    return (samples.mean() - 1) / (samples.std(ddof=1) np.sqrt(samples.shape[0]))
                                                       ^
SyntaxError: invalid syntax
>>> def statistic(samples: np.ndarray) -> float:
...     return (samples.mean() - 1) / (samples.std(ddof=1) / np.sqrt(samples.shape[0]))
...
>>> n = samples.shape[0]
>>> import scipy
>>> from scipy import stats
>>> alpha = 0.05
>>> t1 = stats.t.ppf(alpha / 2, df=n - 1)
>>> t2 = stats.t.ppf(1 - alpha / 2, df=n - 1)
>>> s = statistic(samples)
>>> print(s)
1.9018640008517087
>>>
>>> print(n)
24
>>> print('alpha\tresult')
alpha    result
>>>
>>> print('-------------')
-------------
>>> for alpha in np.linspace(0, 0.15, 15):
...     t1 = stats.t.ppf(alpha / 2, df=n - 1)
...     t2 = stats.t.ppf(1 - alpha / 2, df=n - 1)
...
>>> print(round(alpha, 4), '\t', t1 <= S <= t2)
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
NameError: name 'S' is not defined
>>> print(round(alpha, 4), '\t', t1 <= s <= t2)
0.15       False
```

```
>>> print('alpha\tresult')
alpha	result
>>>
>>> alpha
0.15
>>> t1
-1.4892768970979848
>>> t2
1.489276897097985
>>> alpha = 0.0107
>>> print(round(alpha, 4), '\t', t1 <= s <= t2)
0.0107	False
>>> alpha
0.0107
>>> t1
-1.4892768970979848
>>>
>>> p_left = stats.t.cdf(s, df=n - 1)
>>> p_right = 1 - stats.t.cdf(s, df=n - 1)
>>> pvalue = 2 * (p_left, p_right)
>>> print(pvalue)
(0.9651066265685059, 0.034893373431494124, 0.9651066265685059, 0.034893373431494124)
>>>
```

Задание 4.

Исходные данные:

Построим доверительный интервал для мат. ожидания диаметра подшипника, используя выборку из задания 1. Будем считать, что дисперсия незивестна, и использовать t-статистику.

Решение:

```
Python 3.8.5 (default, May 27 2021, 13:30:53)

[GCC 9.3.0] on linux

Type "help", "copyright", "credits" or "license" for more information.

>>> import numpy as np
```

```
>>> samples = np.array([0.6603, 0.9466, 0.5968, 1.3792,1.5481, 0.7515, 1.0681, 1.1134, 1.2088, 1.7010, 1.0282, 1.3579, 1.0191, 1.1784, 1.1168, 1.1372, 0.7273, 1.3958, 0.8665, 1.5112, 1.1610, 1.0232, 1.0865, 1.0200])

>>> n = samples.shape[0]

>>> mean = samples.mean()

>>> std = samples.std(ddof=1)

>>> print(n, mean, std)

24 1.1084541666666665 0.27936526343958135

>>>

>>> import scipy

>>> from scipy import stats

>>> p = 0.95

>>> alpha = 1 - p

>>> t1 = stats.t.ppf(alpha / 2, df=n - 1)

>>> t2 = stats.t.ppf(1 - alpha / 2, df=n - 1)

>>> print(t1, t2)

-2.0686576104190406 2.0686576104190406

>>>

>>> (mean + t1 * std / np.sqrt(n), mean + t2 * std / np.sqrt(n))

(0.9904885617884089, 1.226419771544924)

>>>
```