问题1

In [1]:

```
import math
import plotly
import numpy as np
import pandas as pd
import cufflinks as cf
from mitosheet import sheet
import plotly.express as px
import plotly.graph_objects as go

from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = 'all'
```

In [ ]:

```
```

In [2]:

```
# 附件参数
copper_ball_material = 'copper'  # 小球材质
copper_ball_R = 10e-3     # m 小球半径
copper_ball_M = 35e-3     # kg 小球质量
mag_sen_rubber_L = 10e-2  # m 磁敏橡胶长

# 查找参数
mu1 = copper_Poisson_ratio = 0.3          # 铜的泊松比
mu2 = mag_sen_rubber_Poisson_ratio = 0.5  # 橡胶的泊松比
E1 = copper_elastic_modulus = 110000      # MPa 铜的弹性模量
E2 = mag_sen_rubber_elastic_modulus = 6   # MPa 橡胶的弹性模量
```

In [ ]:

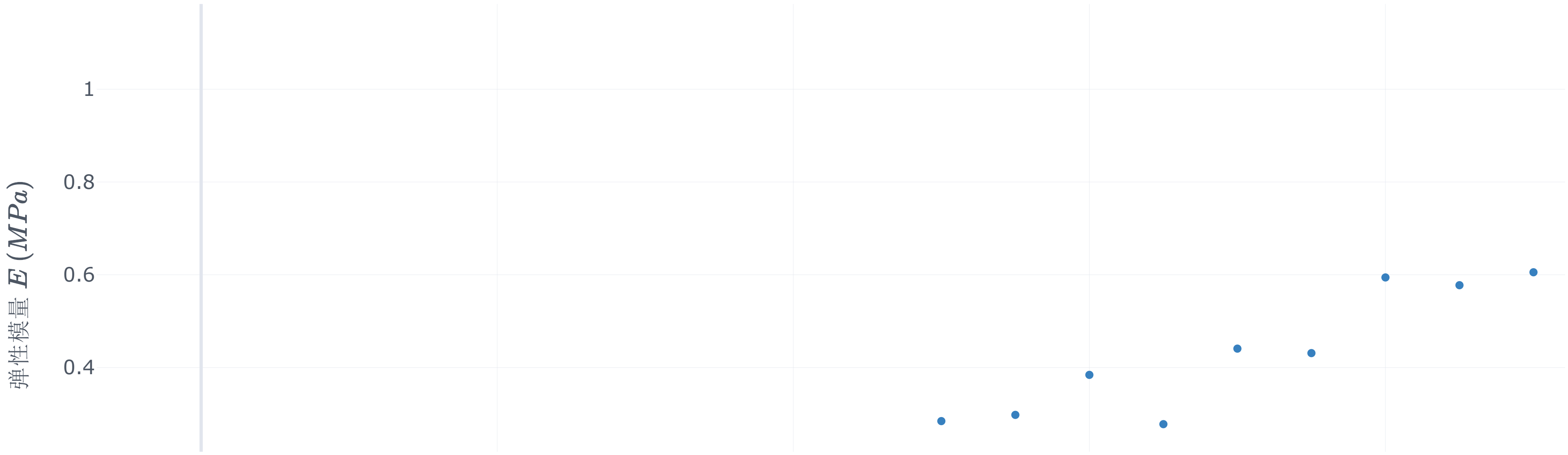
```
```

In [3]:

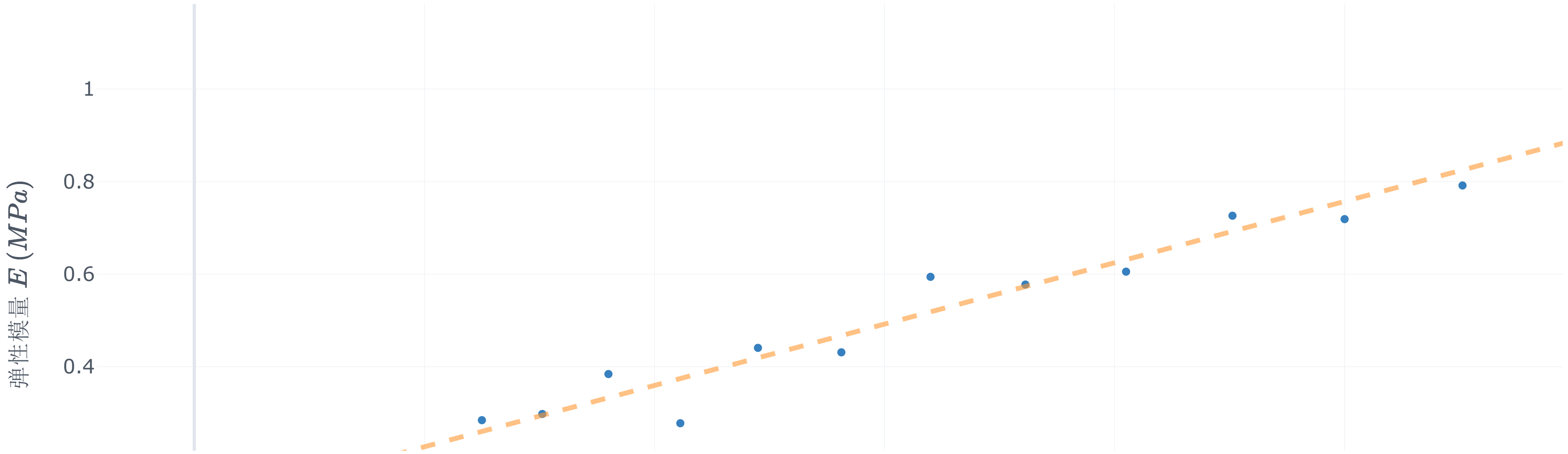
```
n = 1 + 12 * 2
mean, sigma = 0, 2
B = np.array([i / 1000 for i in range(0, 601, 25)])
B_2 = B**2
E = 2.6 * B_2 + np.random.uniform(0, 0.1, n) * sigma
data_B_E = pd.concat([pd.DataFrame(B), pd.DataFrame(E)], axis=1, ignore_index=True)
data_B_E.rename(columns={0: "磁感应强度", 1: "弹性模量"}, inplace=True)
data_B2_E = pd.concat([pd.DataFrame(B_2), pd.DataFrame(E)], axis=1, ignore_index=True)
data_B2_E.rename(columns={0: "磁感应强度的平方", 1: "弹性模量"}, inplace=True)
# sheet(data_B_E, data_B2_E)
```

```python
fig = data_B_E.iplot(
    kind='scatter',
    mode='markers',
    x='磁感应强度',
    y='弹性模量',
    xTitle=r"$磁感应强度~B~(T)$",
    yTitle=r"$弹性模量~E~(MPa)$",
#      bestfit=True,
    colors=['blue'],
    size=5,
)

data_B2_E.iplot(
    kind='scatter',
    mode='markers',
    x='磁感应强度的平方',
    y='弹性模量',
    xTitle=r"$磁感应强度的平方~B^2~(T^2)$",
    yTitle=r"$弹性模量~E~(MPa)$",
    bestfit=True,
    colors=['blue'],
    size=5,
)
```



1
0.8
0.6
0.4
弹性模量 $E\,(MPa)$



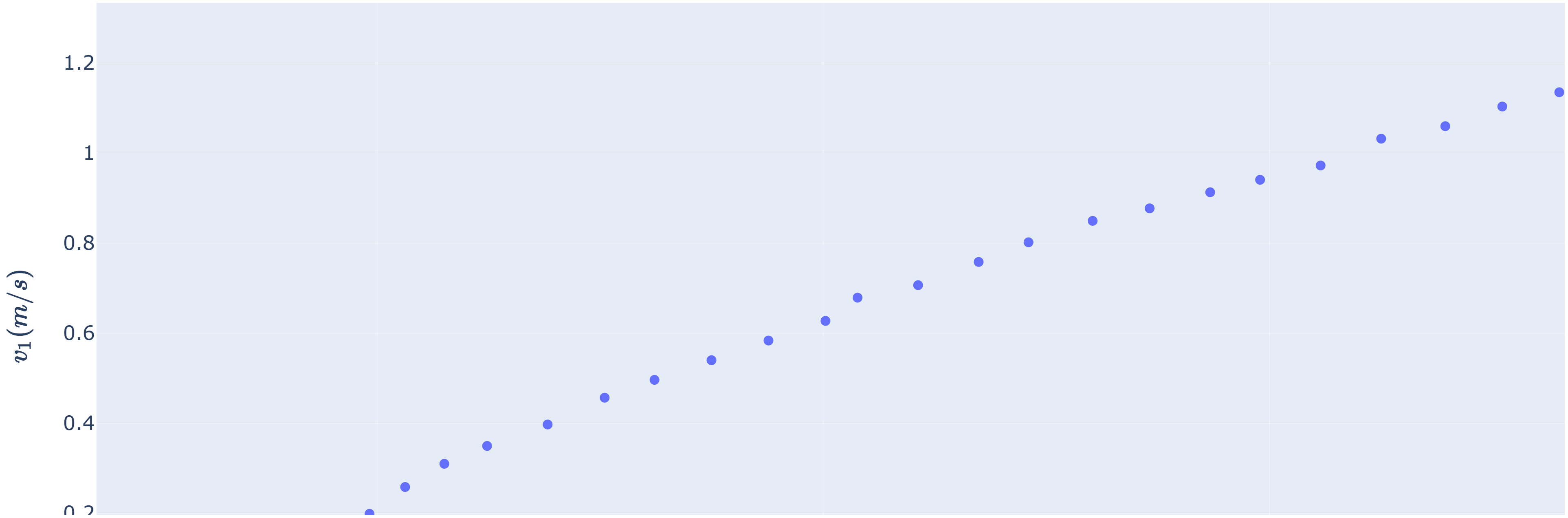
In [ ]:

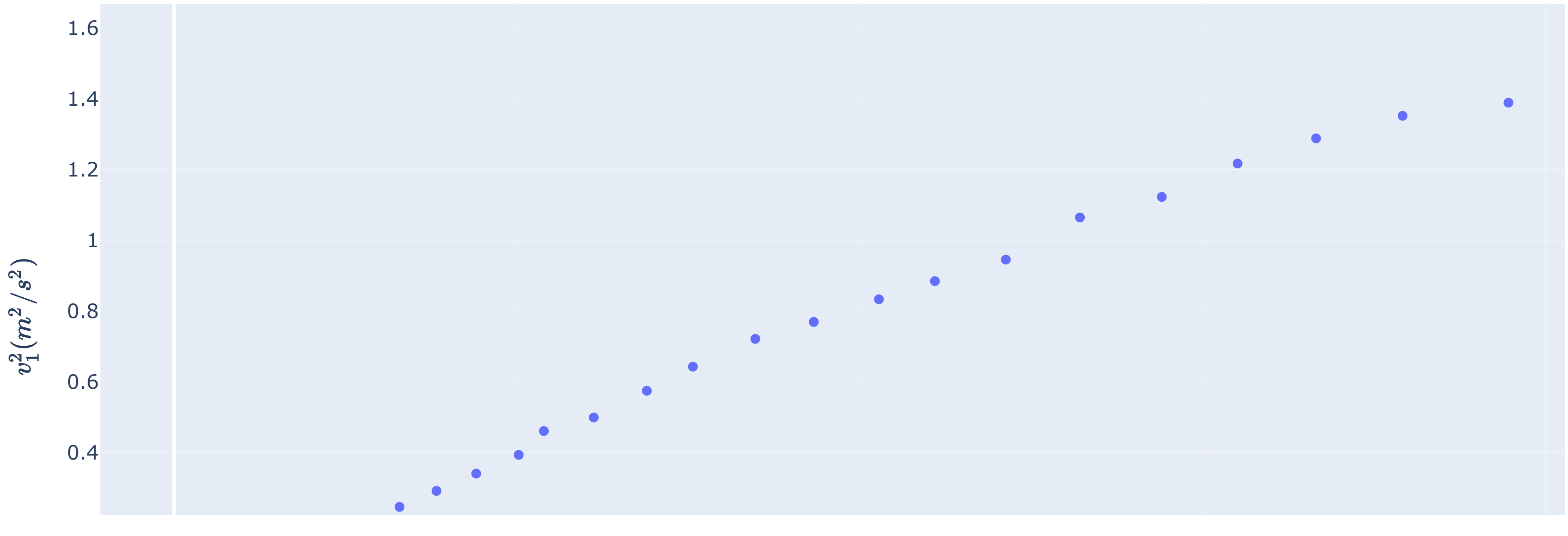
In [4]:

```
def plot_data(data, xtitle=r'$v_0 (m/s)$', ytitle=r'$v_1 (m/s)$'):
    fig = px.scatter(
        data_frame=data,
        x=data.columns[0],
        y=data.columns[1],
    )
    fig.update_layout(
        xaxis_title=xtitle,
        yaxis_title=ytitle,
    )
    fig.show()
```

```python
sheet_names = ["0 mT", "50 mT", "100 mT", "150 mT", "200mT（测试集）"]
data = pd.read_excel("A题附件1.xlsx", sheet_name=sheet_names[4]).iloc[:, :2]
# data
plot_data(data)

data2 = data**2
data2.rename(columns={"初速度（m/s）":"初速度的平方（m^2/s^2）", "末速度（m/s）":"末速度的平方（m^2/s^2）"}, inplace=True)
# data2
plot_data(data2, r'$v_0^2 (m^2/s^2)$', r'$v_1^2 (m^2/s^2)$')
```





In [ ]:

In [5]:

```
import math
import latexify

@latexify.with_latex
def K(v_0, v_1):  # latex
    return 33.2 * R - 23.2 * R**2 * math.sqrt((v_0**2 - v_1**2) / L)
K
```

Out[5]:

$$\mathrm{K}(v_0, v_1) \triangleq 33.2R - 23.2R^2\sqrt{\frac{v_0^2 - v_1^2}{L}}$$

In [6]:

```
def get_K(v0_2, v1_2, R=copper_ball_R, L=mag_sen_rubber_L):
    return 33.2 * R - 23.2 * R**2 * ((v0_2 - v1_2) / L)**0.5
```

In [ ]:

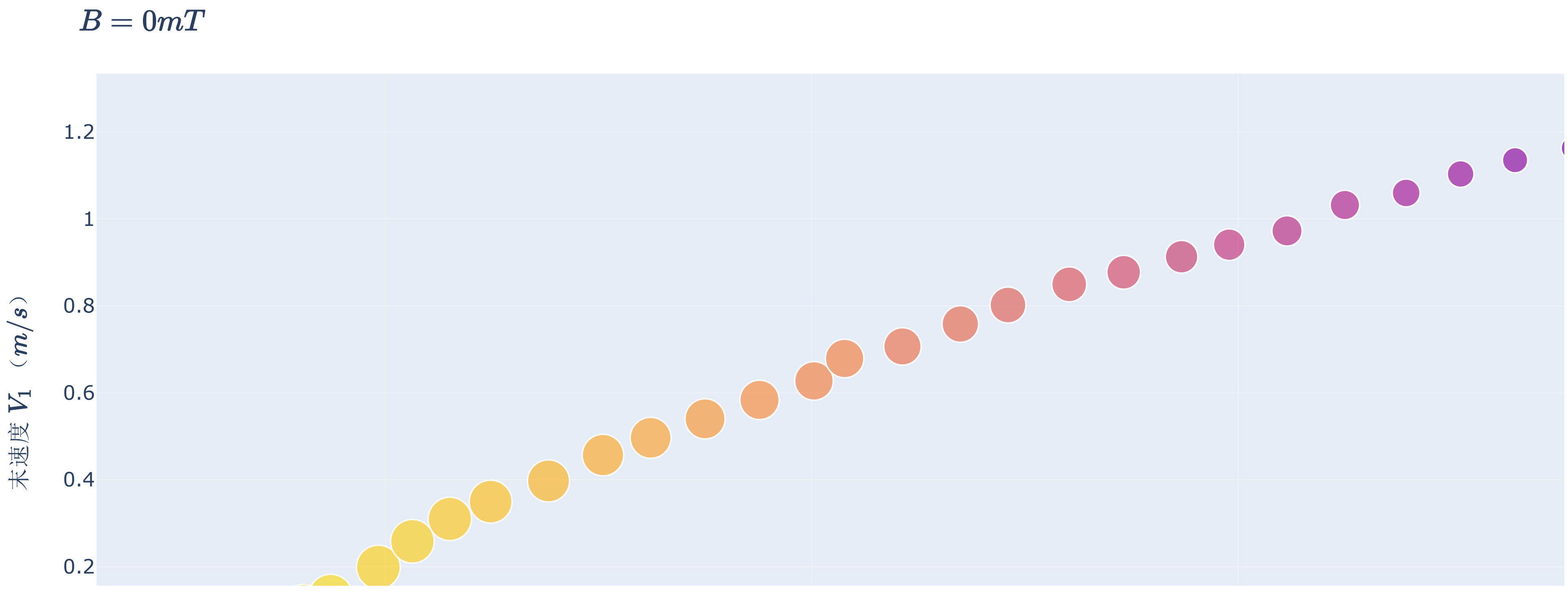
```
```

In [7]:

```
K = pd.DataFrame(get_K(data2.iloc[:, 0].values, data2.iloc[:, 1].values))
data_K = pd.concat([data, K], axis=1, ignore_index=True)
data_K = pd.concat(
    [data_K, (data_K.iloc[:, 2] - min(data_K.iloc[:, 2])) / (max(data_K.iloc[:, 2]) - min(data_K.iloc[:, 2]))],
    axis=1,
    ignore_index=True,
)

titles = [r"$B = 0 mT$", r"$B = 50 mT$", r"$B = 100 mT$", r"$B = 150 mT$", r"$B = 200 mT$", ]
cols = {0: r"$初速度~V_0~（m/s）$", 1: r"$末速度~V_1~（m/s）$", 2: r"滚动摩擦<br> 系数 K", 3: "K_norm"}
data_K.rename(columns=cols, inplace=True)
# data_K

fig = px.scatter(
    data_K,
    x=cols[0],
    y=cols[1],
    labels=cols[2],
    size=cols[3],
    color=cols[2],
    hover_name=cols[2],
)
fig.update_layout(
    title=titles[0],
)
# fig.show()
fig.write_image("v0-v1-K,B=0mT.png")
```



```
In [ ]:
```