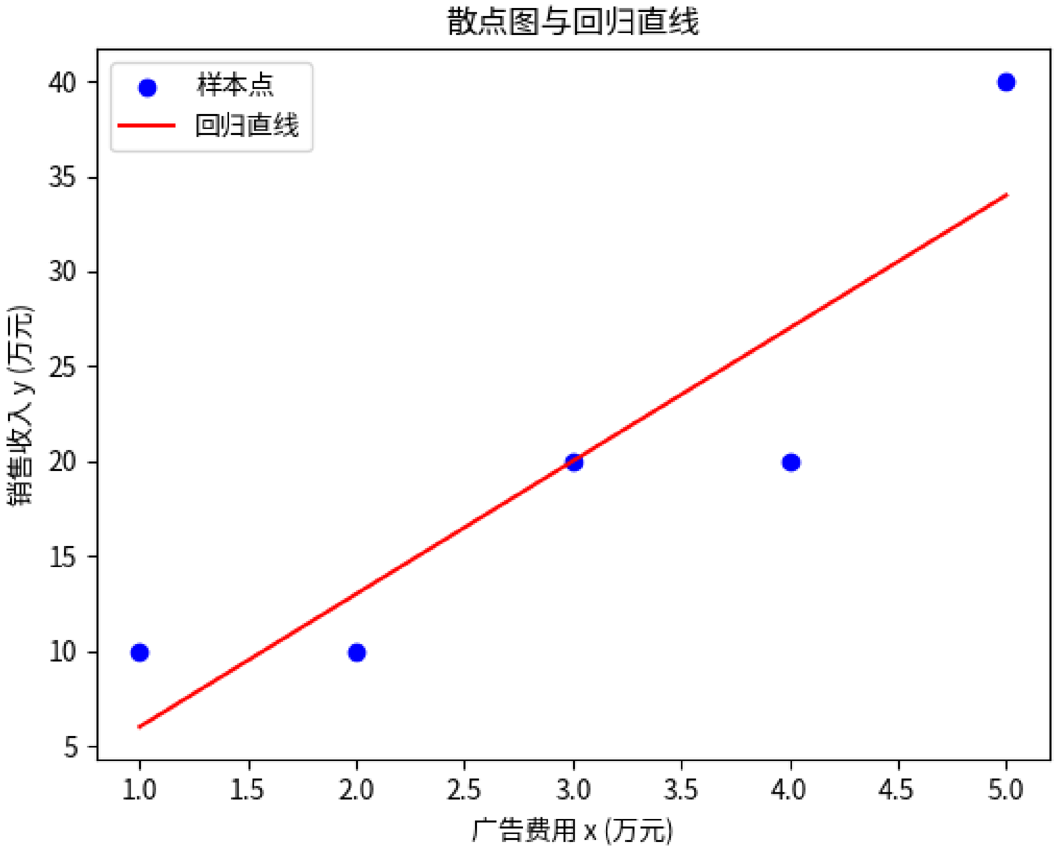
MS2_hw1

September 27, 2025

[6]:

```
# 2-14
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib import font_manager as fm
import statsmodels.api as sm
import statsmodels.formula.api as smf
from scipy import stats

#
x = np.array([1, 2, 3, 4, 5])
y = np.array([10, 10, 20, 20, 40])

#
X = sm.add_constant(x)

# #
model = sm.OLS(y, X).fit()

#
my_font = fm.FontProperties(fname="/usr/share/fonts/opentype/noto/
 ↪NotoSansCJK-Regular.ttc")
plt.rcParams['font.family'] = my_font.get_name()
plt.rcParams['axes.unicode_minus'] = False

plt.scatter(x, y, color="blue", label="  ")
plt.plot(x, model.predict(X), color="red", label="  ")
plt.xlabel("   x ( )")
plt.ylabel("   y ( )")
plt.legend()
plt.title("    ")
plt.show()

#
r, p_value = stats.pearsonr(x, y)
print("(2)    r = %.4f, p = %.4f" % (r, p_value))
```

```
#
print("(3)    : y = %.4f + %.4f x" % (model.params[0], model.params[1]))

#
sigma = np.sqrt(model.mse_resid)
print("(4)      %.4f" % sigma)

#    95%
conf_int = model.conf_int(alpha=0.05)
print("(5)    \beta_0  95%% :", conf_int[0])
print("(5)    \beta_1  95%% :", conf_int[1])

#    R^2
print("(6)     R^2 = %.4f" % model.rsquared)
```



```
(2)     r = 0.9037, p = 0.0354
(3)    : y = -1.0000 + 7.0000 x
(4)      6.0553
(5)   eta_0  95%% : [-21.21124854  19.21124854]
(5)   eta_1  95%% : [ 0.90607928 13.09392072]
```

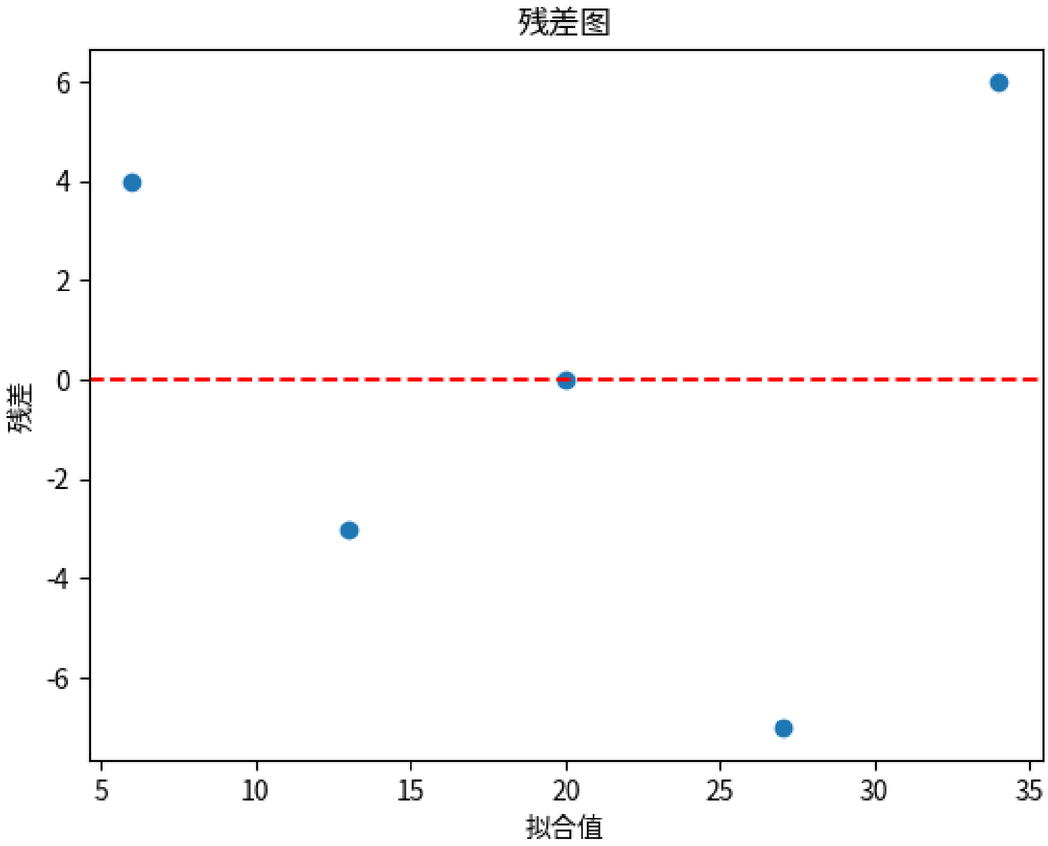
```
(6)    R^2 = 0.8167
```

[7]:
```python
import statsmodels.api as sm

#   DataFrame
df = pd.DataFrame({"x": [1,2,3,4,5],
                   "y": [10,10,20,20,40]})

#
model_fml = smf.ols("y ~ x", data=df).fit()

#
anova_table = sm.stats.anova_lm(model_fml, typ=2)
print("(7)    :")
print(anova_table)
```

```
(7)    :
          sum_sq   df          F    PR(>F)
x          490.0  1.0  13.363636  0.035353
Residual   110.0  3.0        NaN       NaN
```

[8]:
```python
#     1
print("(8)     : t=%.4f, p=%.4f" % (model.tvalues[1], model.pvalues[1]))

#
t_r = r * np.sqrt(len(x) - 2) / np.sqrt(1 - r**2)
p_r = 2 * (1 - stats.t.cdf(abs(t_r), df=len(x) - 2))
print("(9)      : t=%.4f, p=%.4f" % (t_r, p_r))

#
residuals = model.resid
plt.scatter(model.fittedvalues, residuals)
plt.axhline(0, color="red", linestyle="--")
plt.xlabel("  ")
plt.ylabel("  ")
plt.title("  ")
plt.show()

#   95%
x_new = np.array([[1, 4.2]])   #     1
pred = model.get_prediction(x_new)
pred_summary = pred.summary_frame(alpha=0.05)

print("(11)    =4.2   :")
print("    = %.4f" % pred_summary["mean"][0])
print(
    "95%%   = (%.4f, %.4f)"
    % (pred_summary["mean_ci_lower"][0], pred_summary["mean_ci_upper"][0])
)
```

```
(8)      : t=3.6556, p=0.0354
(9)       : t=3.6556, p=0.0354
```



```
(11)     =4.2   :
    = 28.4000
95%    = (17.0975, 39.7025)
```