```python
import week6
import sgd
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import sys

def run_constant(alpha=1):
    T = pd.read_csv("data/T.csv").values
    fg = week6.generate_optimisation_functions(
        T, minibatch_size=5, seed=None)
    o = sgd.StochasticGradientDescent()
    o.alg("constant")
    o.step_size(alpha)
    o.function_generator(fg)
    xs = []
    fs = []
    start = np.array([3, 3])
    o.start(start)
    xs.append(o._x_value)
    for i in range(200):
        o.step()
        print(f"alpha={alpha}:", o._x_value)
        xs.append(o._x_value)
        fs.append(week6.f(o._x_value, T))
    return {
        "x1": [x[0] for x in xs],
        "x2": [x[1] for x in xs],
        "f": fs,
    }

np.random.seed(int(sys.argv[1]))

T = pd.read_csv("data/T.csv").values

x_min, x_max, y_min, y_max = [-5, 5, -5, 5]
# Generate data for wireframe plot
resolution = 100
x_range = np.linspace(x_min, x_max, resolution)
y_range = np.linspace(y_min, y_max, resolution)
X, Y = np.meshgrid(x_range, y_range)

# Plot wireframe
fig = plt.figure(figsize=(12, 6))
resolution = 100
Z_contour = np.zeros_like(X)
for i in range(resolution):
    for j in range(resolution):
        Z_contour[i, j] = week6.f([X[i, j], Y[i, j]], T)

# Plot contour
ax_contour = fig.add_subplot(122)
contour = ax_contour.contourf(X, Y, Z_contour, levels=20, cmap='viridis')
plt.colorbar(contour, ax=ax_contour, label='$f_T(x)$')
ax_contour.set_xlabel('$x_1$')
ax_contour.set_ylabel('$x_2$')
ax_contour.set_xlim([-5,5])
ax_contour.set_ylim([-5,5])
plt.suptitle('Stochastic Gradient Descent on $f_T(x)$ with mini-batches of size 5')

ax_f = fig.add_subplot(121)

for i in range(4):
    alpha = 0.5
    run = run_constant(alpha)
    label = f"$\\alpha={alpha}$, run {i}, best $f_T(x)={min(run['f']):.3f}$"
    ax_contour.plot(run["x1"], run["x2"], label=label)
    ax_f.plot(run['f'], label=label)

ax_f.set_yscale('log')
ax_f.set_xlabel("iteration $t$")
ax_f.set_ylabel("$f_T(x_t)$")
ax_f.legend(loc="upper right")
plt.savefig("fig/bii.pdf")
plt.show()
```