```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import minimize

#
==========================================
==========================================
===
# 1. REAL OBSERVATIONAL DATA: Cosmic
Chronometers (CC) H(z) Data
#
==========================================
==========================================
===
cc_observational_data = np.array([
,[12.0 ,69.0 ,0.090]  ,[19.6 ,69.0 ,0.070]
,[26.2 ,68.6 ,0.120]
```

,[4.0 ,75.0 ,0.179] ,[8.0 ,83.0 ,0.170]
,[5.0 ,75.0 ,0.199]
,[2.6 ,79.6 ,0.240] ,[29.6 ,72.9 ,0.200]
,[14.0 ,77.0 ,0.270]
,[14.0 ,83.0 ,0.352] ,[36.6 ,88.8 ,0.280]
,[13.5 ,83.0 ,0.3802]
,[10.2 ,77.0 ,0.4004] ,[17.0 ,95.0 ,0.400]
,[11.2 ,87.1 ,0.4247]
,[50.0 ,89.0 ,0.470] ,[12.9 ,92.8 ,0.4497]
,[9.0 ,80.9 ,0.4783]
,[13.0 ,104.0 ,0.593] ,[62.0 ,97.0 ,0.480]
,[8.0 ,92.0 ,0.680]
,[17.0 ,125.0 ,0.875] ,[12.0 ,105.0 ,0.781]
,[40.0 ,90.0 ,0.880]
,[20.0 ,154.0 ,1.037] ,[23.0 ,117.0 ,0.900]
,[17.0 ,168.0 ,1.300]
,[18.0 ,177.0 ,1.430] ,[33.6 ,160.0 ,1.363]
,[14.0 ,140.0 ,1.530]
[50.4 ,186.5 ,1.965] ,[40.0 ,202.0 ,1.750]
([

z_obs = cc_observational_data[:, 0]
H_obs = cc_observational_data[:, 1]

```
sigma_H = cc_observational_data[:, 2]

# ================================================================
# 2. THEORETICAL MODELS AND FIXED PARAMETERS
# ================================================================
t0 = 4.352e17
km_s_Mpc_to_SI = 3.24077929e-20

# H0 is derived directly from the age of the universe and β=1.0 (zero free parameters)
beta_RPT_fixed = 1.0
H0_RPT_SI = beta_RPT_fixed / t0
H0_RPT = H0_RPT_SI / km_s_Mpc_to_SI

def H_RPT_fixed(z):
    """RPT with fully locked β=1.0 — no
```

```
        freedom whatsoever"""
        denominator = 1.0 - (1.0 + z)**(-1.0 /
            beta_RPT_fixed)
        denominator = np.where(denominator
            == 0, 1e-10, denominator)
        return H0_RPT * (z / denominator)

    def H_LCDM(z, H0, Om0):
        """Standard ΛCDM with two free
            parameters"""
        return H0 * np.sqrt(Om0 * (1.0 + z)**3 +
            (1.0 - Om0))

    #
================================================
================================================
    ===
    # 3. CHI-SQUARED CALCULATION FOR
        MODEL COMPARISON
    #
================================================
================================================
    ===
```

```
def chi2_LCDM(params):
    H0, Om0 = params
    H_pred = H_LCDM(z_obs, H0, Om0)
    return np.sum(((H_obs - H_pred) /
sigma_H)**2)

# Fitting ΛCDM to the data
res_LCDM = minimize(chi2_LCDM,
x0=[70.0, 0.3], bounds=[(50, 90), (0.1, 0.5)])
best_H0_lcdm, best_Om0_lcdm =
res_LCDM.x

# Computing χ² for RPT (without any
fitting)
H_pred_RPT = H_RPT_fixed(z_obs)
chi2_RPT = np.sum(((H_obs - H_pred_RPT)
/ sigma_H)**2)

# Degrees of freedom and reduced χ²
dof_RPT = len(z_obs)
dof_LCDM = len(z_obs) - 2
red_chi2_RPT = chi2_RPT / dof_RPT
red_chi2_LCDM = res_LCDM.fun /
```

```
                                                dof_LCDM

    # BIC (Bayesian Information Criterion) —
      fewer parameters means lower penalty
                                    n_data = len(z_obs)
                          k_RPT = 0     # zero free parameters
                        k_LCDM = 2    # two free parameters
                              BIC_RPT = chi2_RPT + k_RPT *
                                              np.log(n_data)
                   BIC_LCDM = res_LCDM.fun + k_LCDM *
                                              np.log(n_data)

                                                        #
==========================================
==========================================
                                                      ===
                    # 4. PLOTTING THE COMPARISON
                                                        #
==========================================
==========================================
                                                      ===
                       fig, (ax1, ax2) = plt.subplots(2, 1,
                                             figsize=(12, 10),
```

```
gridspec_kw={'height_ratios': [3, 1]})

z_grid = np.linspace(0.01, 2.1, 300)

# Main plot
ax1.errorbar(z_obs, H_obs, yerr=sigma_H,
fmt='ko', capsize=3, alpha=0.7,
label='Observational Data (32 CC points)')
ax1.plot(z_grid, H_RPT_fixed(z_grid), 'b-',
linewidth=2.5, label=f'RPT ($\\beta$ = 1.0,
NO Free Parameters)')
ax1.plot(z_grid, H_LCDM(z_grid,
best_H0_lcdm, best_Om0_lcdm), 'r--',
linewidth=2, label=f'ΛCDM
(H₀={best_H0_lcdm:.1f},
Ωm={best_Om0_lcdm:.2f})')

ax1.set_ylabel('H(z) [km/s/Mpc]',
fontsize=12)
ax1.set_title('RPT vs ΛCDM: Model with
Zero Free Parameters against Real Data',
fontsize=14, fontweight='bold')
ax1.legend(loc='upper left', fontsize=10)
```

```
ax1.grid(True, alpha=0.3)

# Residuals plot
res_RPT_vals = H_obs - H_pred_RPT
res_LCDM_vals = H_obs - H_LCDM(z_obs,
best_H0_lcdm, best_Om0_lcdm)

ax2.axhline(y=0, color='gray', linestyle='--')
ax2.errorbar(z_obs, res_RPT_vals,
yerr=sigma_H, fmt='bo', alpha=0.6,
label='RPT Residuals')
ax2.errorbar(z_obs, res_LCDM_vals,
yerr=sigma_H, fmt='rx', alpha=0.4,
label='ΛCDM Residuals')
ax2.set_xlabel('Redshift ($z$)', fontsize=12)
ax2.set_ylabel('Δ H(z)', fontsize=12)
ax2.set_title('Residuals: (Data - Model)',
fontsize=11)
ax2.legend(loc='upper right', fontsize=9)
ax2.grid(True, alpha=0.3)

plt.tight_layout()
plt.show()
```

```
#
==========
==========
===
# 5. FINAL COMPARISON REPORT
#
==========
==========
===
print("=" * 70)
print("    RPT vs ΛCDM — Zero Free
Parameters Challenge")
print("=" * 70)
print(f"RPT (fixed β = 1.0):")
print(f"  • H₀ (derived from cosmic age):
{H0_RPT:.1f} km/s/Mpc")
print(f"  • Free parameters: 0")
print(f"  • χ² = {chi2_RPT:.2f}  |  χ²/dof =
{red_chi2_RPT:.3f}")
print(f"  • BIC = {BIC_RPT:.2f}")
print()
print(f"ΛCDM (fitted):")
```

```
print(f"  • H₀ = {best_H0_lcdm:.1f} km/s/Mpc")
print(f"  • Ωm = {best_Om0_lcdm:.3f}")
print(f"  • Free parameters: 2")
print(f"  • χ² = {res_LCDM.fun:.2f}  |  χ²/dof = {red_chi2_LCDM:.3f}")
print(f"  • BIC = {BIC_LCDM:.2f}")
print()
if BIC_RPT < BIC_LCDM:
print("✅ RPT has a lower BIC, making it the statistically superior model!")
print("   (even with zero free parameters)")
else:
print("⚠️ ΛCDM still has a lower BIC, but RPT with 0 parameters is a close competitor.")
print("=" * 70)
```