

06__empirical_homogeneity_test

December 18, 2025

1 Empirical Homogeneity Test

1.1 Notebook Setup

```
[1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm
import torch
import sys
sys.path.append('../')
from utils import utils
sys.executable
```

```
[1]: '/Users/fanghema/Desktop/aaSTAT_5200/STAT_5200_final_project/env/bin/python'
```

```
[2]: data = pd.read_csv(
    '../data/processed/data_extended.csv',
    index_col=0,
    parse_dates=True
)

factors = ['Mkt-RF', 'SMB', 'HML', 'RMW', 'CMA']
assets = [col for col in data.columns if col != 'RF' and col not in factors]
data['Quarter'] = data.index.to_period("Q")
```

1.2 Set up empirical testing parameters

```
[3]: factor_options = [
    ['Mkt-RF'],
    ['Mkt-RF', 'SMB', 'HML'],
    ['Mkt-RF', 'SMB', 'HML', 'RMW', 'CMA'],
]
R_options = [1, 2, 5]
sample_period_options = [
    ('1963-01-01', '2025-12-31'),
    ('1963-01-01', '1983-01-01'),
    ('1973-01-01', '1993-01-01'),
]
```

```

('1983-01-01', '2003-01-01'),
('1993-01-01', '2013-01-01'),
('2003-01-01', '2023-01-01'),
]

results = pd.DataFrame(
    index=pd.MultiIndex.from_product([
        list(map(tuple, factor_options)), # convert lists to tuples
        R_options,
        sample_period_options
    ]),
    columns=['gamma_a_lam', 'gamma_a', 'gamma_lam']
)

print(f"Total combinations: {results.shape[0]}")
counter = 0

for factors in factor_options:
    K = len(factors)
    for R in R_options:
        for sample_period in sample_period_options:
            print(f"Processing {counter}/{results.shape[0]}: {factors} - {R} - {sample_period}")
            data_slice = data.loc[
                (data.index > sample_period[0]) &
                (data.index < sample_period[1])
            ]
            beta_loading, returns_df, realized_covariance, residuals = utils.
            calculate_factor_loading(
                data_slice,
                factors=factors,
                assets=assets
            )

            excess_returns = returns_df.groupby("Quarter").sum()[assets].T.
            values
            industries = beta_loading.index.get_level_values(0).unique().
            tolist()
            factors_names = beta_loading.index.get_level_values(1).unique().
            tolist()

            N = len(industries)
            K = len(factors)
            T = beta_loading.shape[1]

            beta_hat_np = np.zeros((N, K, T))

```

```

for i, asset in enumerate(industries):
    for j, factor in enumerate(factors):
        beta_hat_np[i, j, :] = beta_loading.loc[(asset, factor)].

↪ values

eta, G, beta_star, objective = utils.iterative_convergence(
    beta_hat_np,
    excess_returns,
    N = N,
    K = K,
    R = R,
    T = T,
    n_iter=2000
)

avar = utils.estimate_avar(
    beta_hat=beta_hat_np,
    excess_returns=excess_returns,
    eta=eta,
    G=G,
    beta_star=beta_star,
    realized_covariance=realized_covariance,
    residuals=residuals,
    N = N,
    K = K,
    R = R,
    T = T,
)

gamma_a_lambda = utils.full_homogeneity_test(
    eta = eta,
    avar = avar,
    N = N,
    K = K,
    T = T
)

gamma_a = utils.intercept_homogeneity_test(
    eta = eta,
    avar = avar,
    N = N,
    K = K,
    T = T
)

```

```

gamma_lambda = utils.slope_homogeneity_test(
    eta = eta,
    avar = avar,
    N = N,
    K = K,
    T = T
)
print(f"Test statistics")
print(f"gamma_a_lam: {gamma_a_lambda}")
print(f"gamma_a: {gamma_a}")
print(f"gamma_lam: {gamma_lambda}")

results.loc[(
    tuple(factors), R, sample_period
)] = np.asarray([
    gamma_a_lambda,
    gamma_a,
    gamma_lambda
])
counter += 1
print(f"=====")

```

Total combinations: 54

Processing 0/54: ['Mkt-RF'] - 1 - ('1963-01-01', '2025-12-31')

Test statistics

gamma_a_lam: 44.021375176126504

gamma_a: -4.700491211791159

gamma_lam: -4.793114003302496

=====

Processing 1/54: ['Mkt-RF'] - 1 - ('1963-01-01', '1983-01-01')

Test statistics

gamma_a_lam: -19.45507836193608

gamma_a: -5.290005147715778

gamma_lam: -4.82127703591286

=====

Processing 2/54: ['Mkt-RF'] - 1 - ('1973-01-01', '1993-01-01')

Test statistics

gamma_a_lam: 38.51023758818772

gamma_a: -4.780716850584511

gamma_lam: -4.7891599721430245

=====

Processing 3/54: ['Mkt-RF'] - 1 - ('1983-01-01', '2003-01-01')

Test statistics

gamma_a_lam: -29.992553862296607

gamma_a: -4.8616347938768305

gamma_lam: -4.796865405543951

=====

```

Processing 4/54: ['Mkt-RF'] - 1 - ('1993-01-01', '2013-01-01')
Test statistics
gamma_a_lam: -11.007995129966412
gamma_a: -5.291507502318423
gamma_lam: -4.738953249229063
=====
Processing 5/54: ['Mkt-RF'] - 1 - ('2003-01-01', '2023-01-01')
Test statistics
gamma_a_lam: 24.688723708717475
gamma_a: -3.749937797792885
gamma_lam: -4.3028255952482075
=====
Processing 6/54: ['Mkt-RF'] - 2 - ('1963-01-01', '2025-12-31')
Test statistics
gamma_a_lam: -2.6732672426974027
gamma_a: -5.339212629468527
gamma_lam: -4.798137830129114
=====
Processing 7/54: ['Mkt-RF'] - 2 - ('1963-01-01', '1983-01-01')
Test statistics
gamma_a_lam: 88.12448718838729
gamma_a: 0.48587747852576674
gamma_lam: -2.3129955443075003
=====
Processing 8/54: ['Mkt-RF'] - 2 - ('1973-01-01', '1993-01-01')
Test statistics
gamma_a_lam: -42.373442941233144
gamma_a: -4.8611064892949125
gamma_lam: -4.798696771105146
=====
Processing 9/54: ['Mkt-RF'] - 2 - ('1983-01-01', '2003-01-01')
Test statistics
gamma_a_lam: -66.67978493809103
gamma_a: -4.304672737445764
gamma_lam: -5.388708326488772
=====
Processing 10/54: ['Mkt-RF'] - 2 - ('1993-01-01', '2013-01-01')
Test statistics
gamma_a_lam: -52.79965093886
gamma_a: -6.252624253692823
gamma_lam: -5.062956245535808
=====
Processing 11/54: ['Mkt-RF'] - 2 - ('2003-01-01', '2023-01-01')
Test statistics
gamma_a_lam: -4.784644320688956
gamma_a: -5.196610265700589
gamma_lam: -4.767763418798012
=====

```

```

Processing 12/54: ['Mkt-RF'] - 5 - ('1963-01-01', '2025-12-31')
Test statistics
gamma_a_lam: 2438.235754570473
gamma_a: -4.408409913623353
gamma_lam: 10.27941437482029
=====
Processing 13/54: ['Mkt-RF'] - 5 - ('1963-01-01', '1983-01-01')
Test statistics
gamma_a_lam: -54.07376914753523
gamma_a: -2.3270995295242027
gamma_lam: -2.9256391278549447
=====
Processing 14/54: ['Mkt-RF'] - 5 - ('1973-01-01', '1993-01-01')
Test statistics
gamma_a_lam: 54.92341803552409
gamma_a: -4.614662623076495
gamma_lam: -4.375630327908701
=====
Processing 15/54: ['Mkt-RF'] - 5 - ('1983-01-01', '2003-01-01')
Test statistics
gamma_a_lam: -110.7462205971981
gamma_a: -0.6415817446507754
gamma_lam: -3.2707208769144454
=====
Processing 16/54: ['Mkt-RF'] - 5 - ('1993-01-01', '2013-01-01')
Test statistics
gamma_a_lam: 23007.892323800672
gamma_a: -4.420165690507534
gamma_lam: 405.0376607175024
=====
Processing 17/54: ['Mkt-RF'] - 5 - ('2003-01-01', '2023-01-01')
Test statistics
gamma_a_lam: 2327.3686235461323
gamma_a: -4.628706308110402
gamma_lam: 52.707782016738044
=====
Processing 18/54: ['Mkt-RF', 'SMB', 'HML'] - 1 - ('1963-01-01', '2025-12-31')
Test statistics
gamma_a_lam: -8.923047093232306
gamma_a: -8.342481795516516
gamma_lam: -8.316274429301354
=====
Processing 19/54: ['Mkt-RF', 'SMB', 'HML'] - 1 - ('1963-01-01', '1983-01-01')
Test statistics
gamma_a_lam: -5.314649047383824
gamma_a: -8.280847252572762
gamma_lam: -8.281782396994634
=====

```

```

Processing 20/54: ['Mkt-RF', 'SMB', 'HML'] - 1 - ('1973-01-01', '1993-01-01')
Test statistics
gamma_a_lam: -10.300182757377934
gamma_a: -8.308301059276616
gamma_lam: -8.302502751359713
=====
Processing 21/54: ['Mkt-RF', 'SMB', 'HML'] - 1 - ('1983-01-01', '2003-01-01')
Test statistics
gamma_a_lam: -2.406316386752981
gamma_a: -8.300168588980867
gamma_lam: -8.319993557702267
=====
Processing 22/54: ['Mkt-RF', 'SMB', 'HML'] - 1 - ('1993-01-01', '2013-01-01')
Test statistics
gamma_a_lam: -0.41990634528970106
gamma_a: -8.319842344640795
gamma_lam: -8.194226581906426
=====
Processing 23/54: ['Mkt-RF', 'SMB', 'HML'] - 1 - ('2003-01-01', '2023-01-01')
Test statistics
gamma_a_lam: -3.4937121635169968
gamma_a: -8.412641044776443
gamma_lam: -8.185507994135017
=====
Processing 24/54: ['Mkt-RF', 'SMB', 'HML'] - 2 - ('1963-01-01', '2025-12-31')
Test statistics
gamma_a_lam: -6.179766852180811
gamma_a: -8.332050591414788
gamma_lam: -8.304815827058258
=====
Processing 25/54: ['Mkt-RF', 'SMB', 'HML'] - 2 - ('1963-01-01', '1983-01-01')
Test statistics
gamma_a_lam: -5.485200270298585
gamma_a: -8.320074970812707
gamma_lam: -8.234234510709241
=====
Processing 26/54: ['Mkt-RF', 'SMB', 'HML'] - 2 - ('1973-01-01', '1993-01-01')
Test statistics
gamma_a_lam: -8.930289623322828
gamma_a: -8.307522529574804
gamma_lam: -8.296730946931552
=====
Processing 27/54: ['Mkt-RF', 'SMB', 'HML'] - 2 - ('1983-01-01', '2003-01-01')
Test statistics
gamma_a_lam: 28.525921920863002
gamma_a: -7.92538694894371
gamma_lam: -7.476400000084076
=====

```

```

Processing 28/54: ['Mkt-RF', 'SMB', 'HML'] - 2 - ('1993-01-01', '2013-01-01')
Test statistics
gamma_a_lam: -5.057213948893936
gamma_a: -8.306454288597754
gamma_lam: -8.205388413949166
=====
Processing 29/54: ['Mkt-RF', 'SMB', 'HML'] - 2 - ('2003-01-01', '2023-01-01')
Test statistics
gamma_a_lam: 6.245207755020384
gamma_a: -8.295843603336781
gamma_lam: -8.119037880321539
=====
Processing 30/54: ['Mkt-RF', 'SMB', 'HML'] - 5 - ('1963-01-01', '2025-12-31')
Test statistics
gamma_a_lam: 25.315847021418865
gamma_a: -8.303484886785856
gamma_lam: -8.153324480349433
=====
Processing 31/54: ['Mkt-RF', 'SMB', 'HML'] - 5 - ('1963-01-01', '1983-01-01')
Test statistics
gamma_a_lam: 80.82782726012826
gamma_a: -8.145131563688048
gamma_lam: -5.364330857647703
=====
Processing 32/54: ['Mkt-RF', 'SMB', 'HML'] - 5 - ('1973-01-01', '1993-01-01')
Test statistics
gamma_a_lam: -8.791553803265451
gamma_a: -8.306774051476598
gamma_lam: -8.293809428971358
=====
Processing 33/54: ['Mkt-RF', 'SMB', 'HML'] - 5 - ('1983-01-01', '2003-01-01')
Test statistics
gamma_a_lam: 160.97248409693034
gamma_a: -5.5733069604654455
gamma_lam: -8.02360459901131
=====
Processing 34/54: ['Mkt-RF', 'SMB', 'HML'] - 5 - ('1993-01-01', '2013-01-01')
Test statistics
gamma_a_lam: 20.58239283143639
gamma_a: -8.309451357234803
gamma_lam: -7.91183313085424
=====
Processing 35/54: ['Mkt-RF', 'SMB', 'HML'] - 5 - ('2003-01-01', '2023-01-01')
Test statistics
gamma_a_lam: 190.7316567042458
gamma_a: -8.28268522981492
gamma_lam: -5.35425478148479
=====

```



```

Processing 36/54: ['Mkt-RF', 'SMB', 'HML', 'RMW', 'CMA'] - 1 - ('1963-01-01',
'2025-12-31')
Test statistics
gamma_a_lam: -6.989102222335923
gamma_a: -10.729210072723074
gamma_lam: -10.71198401688972
=====
Processing 37/54: ['Mkt-RF', 'SMB', 'HML', 'RMW', 'CMA'] - 1 - ('1963-01-01',
'1983-01-01')
Test statistics
gamma_a_lam: -318.5416871338544
gamma_a: -10.740611204993483
gamma_lam: -14.80512922722086
=====
Processing 38/54: ['Mkt-RF', 'SMB', 'HML', 'RMW', 'CMA'] - 1 - ('1973-01-01',
'1993-01-01')
Test statistics
gamma_a_lam: -10.775980300456077
gamma_a: -10.723980282508062
gamma_lam: -10.711166523369874
=====
Processing 39/54: ['Mkt-RF', 'SMB', 'HML', 'RMW', 'CMA'] - 1 - ('1983-01-01',
'2003-01-01')
Test statistics
gamma_a_lam: -11.416336712717651
gamma_a: -10.721325132066054
gamma_lam: -10.704642047423233
=====
Processing 40/54: ['Mkt-RF', 'SMB', 'HML', 'RMW', 'CMA'] - 1 - ('1993-01-01',
'2013-01-01')
Test statistics
gamma_a_lam: -5.13661953183519
gamma_a: -10.729204576609552
gamma_lam: -10.63414572420469
=====
Processing 41/54: ['Mkt-RF', 'SMB', 'HML', 'RMW', 'CMA'] - 1 - ('2003-01-01',
'2023-01-01')
Test statistics
gamma_a_lam: 8.601292836020288
gamma_a: -10.724989530371124
gamma_lam: -10.439807841455991
=====
Processing 42/54: ['Mkt-RF', 'SMB', 'HML', 'RMW', 'CMA'] - 2 - ('1963-01-01',
'2025-12-31')
Test statistics
gamma_a_lam: -8.436648789139339
gamma_a: -10.723460705114645
gamma_lam: -10.694462133694998

```

```

=====
Processing 43/54: ['Mkt-RF', 'SMB', 'HML', 'RMW', 'CMA'] - 2 - ('1963-01-01',
'1983-01-01')
Test statistics
gamma_a_lam: -7.207820064974214
gamma_a: -10.73009196773403
gamma_lam: -10.679977870666
=====
Processing 44/54: ['Mkt-RF', 'SMB', 'HML', 'RMW', 'CMA'] - 2 - ('1973-01-01',
'1993-01-01')
Test statistics
gamma_a_lam: -11.08305022537954
gamma_a: -10.723819109374606
gamma_lam: -10.714400535178392
=====
Processing 45/54: ['Mkt-RF', 'SMB', 'HML', 'RMW', 'CMA'] - 2 - ('1983-01-01',
'2003-01-01')
Test statistics
gamma_a_lam: -9.90601732155809
gamma_a: -10.723600199018447
gamma_lam: -10.700533788162488
=====
Processing 46/54: ['Mkt-RF', 'SMB', 'HML', 'RMW', 'CMA'] - 2 - ('1993-01-01',
'2013-01-01')
Test statistics
gamma_a_lam: -1.924426667705066
gamma_a: -10.71666063535228
gamma_lam: -10.458018252513854
=====
Processing 47/54: ['Mkt-RF', 'SMB', 'HML', 'RMW', 'CMA'] - 2 - ('2003-01-01',
'2023-01-01')
Test statistics
gamma_a_lam: 0.33669647749347487
gamma_a: -10.717627169786844
gamma_lam: -10.581157489857768
=====
Processing 48/54: ['Mkt-RF', 'SMB', 'HML', 'RMW', 'CMA'] - 5 - ('1963-01-01',
'2025-12-31')
Test statistics
gamma_a_lam: -9.916105003169564
gamma_a: -10.713971677782427
gamma_lam: -10.705596837215989
=====
Processing 49/54: ['Mkt-RF', 'SMB', 'HML', 'RMW', 'CMA'] - 5 - ('1963-01-01',
'1983-01-01')
Test statistics
gamma_a_lam: 1.3329101008414095
gamma_a: -10.592708095027165

```

```

gamma_lam: -10.59276717465792
=====
Processing 50/54: ['Mkt-RF', 'SMB', 'HML', 'RMW', 'CMA'] - 5 - ('1973-01-01',
'1993-01-01')
Test statistics
gamma_a_lam: -7.920070704109096
gamma_a: -10.723821766053717
gamma_lam: -10.670876394277588
=====
Processing 51/54: ['Mkt-RF', 'SMB', 'HML', 'RMW', 'CMA'] - 5 - ('1983-01-01',
'2003-01-01')
Test statistics
gamma_a_lam: 3.275619012570612
gamma_a: -10.556758491551385
gamma_lam: -10.589651287052614
=====
Processing 52/54: ['Mkt-RF', 'SMB', 'HML', 'RMW', 'CMA'] - 5 - ('1993-01-01',
'2013-01-01')
Test statistics
gamma_a_lam: 39.095190938146686
gamma_a: -10.723974414915215
gamma_lam: -10.027807855086415
=====
Processing 53/54: ['Mkt-RF', 'SMB', 'HML', 'RMW', 'CMA'] - 5 - ('2003-01-01',
'2023-01-01')
Test statistics
gamma_a_lam: 1.2406245003043173
gamma_a: -10.724461939758651
gamma_lam: -10.542239363092206
=====

```

[4]: results

```

[4]:
(Mkt-RF,)
1 (1963-01-01, 2025-12-31)    gamma_a_lam \
    (1963-01-01, 1983-01-01)    44.021375
    (1973-01-01, 1993-01-01)   -19.455078
    (1983-01-01, 2003-01-01)   38.510238
    (1993-01-01, 2013-01-01)  -29.992554
    (2003-01-01, 2023-01-01)  -11.007995
    (2023-01-01, 2025-12-31)   24.688724
2 (1963-01-01, 2025-12-31)    -2.673267
    (1963-01-01, 1983-01-01)   88.124487
    (1973-01-01, 1993-01-01)  -42.373443
    (1983-01-01, 2003-01-01)  -66.679785
    (1993-01-01, 2013-01-01)  -52.799651
    (2003-01-01, 2023-01-01)   -4.784644
5 (1963-01-01, 2025-12-31)   2438.235755

```

		(1963-01-01, 1983-01-01)	-54.073769	
		(1973-01-01, 1993-01-01)	54.923418	
		(1983-01-01, 2003-01-01)	-110.746221	
		(1993-01-01, 2013-01-01)	23007.892324	
		(2003-01-01, 2023-01-01)	2327.368624	
(Mkt-RF, SMB, HML)	1	(1963-01-01, 2025-12-31)	-8.923047	
		(1963-01-01, 1983-01-01)	-5.314649	
		(1973-01-01, 1993-01-01)	-10.300183	
		(1983-01-01, 2003-01-01)	-2.406316	
		(1993-01-01, 2013-01-01)	-0.419906	
		(2003-01-01, 2023-01-01)	-3.493712	
	2	(1963-01-01, 2025-12-31)	-6.179767	
		(1963-01-01, 1983-01-01)	-5.4852	
		(1973-01-01, 1993-01-01)	-8.93029	
		(1983-01-01, 2003-01-01)	28.525922	
		(1993-01-01, 2013-01-01)	-5.057214	
		(2003-01-01, 2023-01-01)	6.245208	
	5	(1963-01-01, 2025-12-31)	25.315847	
		(1963-01-01, 1983-01-01)	80.827827	
		(1973-01-01, 1993-01-01)	-8.791554	
		(1983-01-01, 2003-01-01)	160.972484	
		(1993-01-01, 2013-01-01)	20.582393	
		(2003-01-01, 2023-01-01)	190.731657	
(Mkt-RF, SMB, HML, RMW, CMA)	1	(1963-01-01, 2025-12-31)	-6.989102	
		(1963-01-01, 1983-01-01)	-318.541687	
		(1973-01-01, 1993-01-01)	-10.77598	
		(1983-01-01, 2003-01-01)	-11.416337	
		(1993-01-01, 2013-01-01)	-5.13662	
		(2003-01-01, 2023-01-01)	8.601293	
	2	(1963-01-01, 2025-12-31)	-8.436649	
		(1963-01-01, 1983-01-01)	-7.20782	
		(1973-01-01, 1993-01-01)	-11.08305	
		(1983-01-01, 2003-01-01)	-9.906017	
		(1993-01-01, 2013-01-01)	-1.924427	
		(2003-01-01, 2023-01-01)	0.336696	
	5	(1963-01-01, 2025-12-31)	-9.916105	
		(1963-01-01, 1983-01-01)	1.33291	
		(1973-01-01, 1993-01-01)	-7.920071	
		(1983-01-01, 2003-01-01)	3.275619	
		(1993-01-01, 2013-01-01)	39.095191	
		(2003-01-01, 2023-01-01)	1.240625	
			gamma_a	gamma_lam
(Mkt-RF,)	1	(1963-01-01, 2025-12-31)	-4.700491	-4.793114
		(1963-01-01, 1983-01-01)	-5.290005	-4.821277
		(1973-01-01, 1993-01-01)	-4.780717	-4.78916
		(1983-01-01, 2003-01-01)	-4.861635	-4.796865

		(1993-01-01, 2013-01-01)	-5.291508	-4.738953
		(2003-01-01, 2023-01-01)	-3.749938	-4.302826
2		(1963-01-01, 2025-12-31)	-5.339213	-4.798138
		(1963-01-01, 1983-01-01)	0.485877	-2.312996
		(1973-01-01, 1993-01-01)	-4.861106	-4.798697
		(1983-01-01, 2003-01-01)	-4.304673	-5.388708
		(1993-01-01, 2013-01-01)	-6.252624	-5.062956
		(2003-01-01, 2023-01-01)	-5.19661	-4.767763
5		(1963-01-01, 2025-12-31)	-4.40841	10.279414
		(1963-01-01, 1983-01-01)	-2.3271	-2.925639
		(1973-01-01, 1993-01-01)	-4.614663	-4.37563
		(1983-01-01, 2003-01-01)	-0.641582	-3.270721
		(1993-01-01, 2013-01-01)	-4.420166	405.037661
		(2003-01-01, 2023-01-01)	-4.628706	52.707782
(Mkt-RF, SMB, HML)	1	(1963-01-01, 2025-12-31)	-8.342482	-8.316274
		(1963-01-01, 1983-01-01)	-8.280847	-8.281782
		(1973-01-01, 1993-01-01)	-8.308301	-8.302503
		(1983-01-01, 2003-01-01)	-8.300169	-8.319994
		(1993-01-01, 2013-01-01)	-8.319842	-8.194227
		(2003-01-01, 2023-01-01)	-8.412641	-8.185508
	2	(1963-01-01, 2025-12-31)	-8.332051	-8.304816
		(1963-01-01, 1983-01-01)	-8.320075	-8.234235
		(1973-01-01, 1993-01-01)	-8.307523	-8.296731
		(1983-01-01, 2003-01-01)	-7.925387	-7.4764
		(1993-01-01, 2013-01-01)	-8.306454	-8.205388
		(2003-01-01, 2023-01-01)	-8.295844	-8.119038
	5	(1963-01-01, 2025-12-31)	-8.303485	-8.153324
		(1963-01-01, 1983-01-01)	-8.145132	-5.364331
		(1973-01-01, 1993-01-01)	-8.306774	-8.293809
		(1983-01-01, 2003-01-01)	-5.573307	-8.023605
		(1993-01-01, 2013-01-01)	-8.309451	-7.911833
		(2003-01-01, 2023-01-01)	-8.282685	-5.354255
(Mkt-RF, SMB, HML, RMW, CMA)	1	(1963-01-01, 2025-12-31)	-10.72921	-10.711984
		(1963-01-01, 1983-01-01)	-10.740611	-14.805129
		(1973-01-01, 1993-01-01)	-10.72398	-10.711167
		(1983-01-01, 2003-01-01)	-10.721325	-10.704642
		(1993-01-01, 2013-01-01)	-10.729205	-10.634146
		(2003-01-01, 2023-01-01)	-10.72499	-10.439808
	2	(1963-01-01, 2025-12-31)	-10.723461	-10.694462
		(1963-01-01, 1983-01-01)	-10.730092	-10.679978
		(1973-01-01, 1993-01-01)	-10.723819	-10.714401
		(1983-01-01, 2003-01-01)	-10.7236	-10.700534
		(1993-01-01, 2013-01-01)	-10.716661	-10.458018
		(2003-01-01, 2023-01-01)	-10.717627	-10.581157
	5	(1963-01-01, 2025-12-31)	-10.713972	-10.705597
		(1963-01-01, 1983-01-01)	-10.592708	-10.592767
		(1973-01-01, 1993-01-01)	-10.723822	-10.670876

(1983-01-01, 2003-01-01)	-10.556758	-10.589651
(1993-01-01, 2013-01-01)	-10.723974	-10.027808
(2003-01-01, 2023-01-01)	-10.724462	-10.542239

```
[5]: def clean_results_index(results):

    fac_idx = results.index.get_level_values(0)
    R_idx    = results.index.get_level_values(1)
    t_idx    = results.index.get_level_values(2)

    fac_new = [len(x) for x in fac_idx]

    t_new = [
        f"{str(start)[:4]}--{str(end)[:4]}"
        for start, end in t_idx
    ]

    new_index = pd.MultiIndex.from_arrays(
        [fac_new, R_idx, t_new],
        names=["K", "R", "Period"]
    )

    cleaned = results.copy()
    cleaned.index = new_index
    return cleaned

cleaned_results = clean_results_index(results)
cleaned_results
```

```
[5]:
```

			gamma_a_lam	gamma_a	gamma_lam
	K	R	Period		
1	1	1963-2025	44.021375	-4.700491	-4.793114
		1963-1983	-19.455078	-5.290005	-4.821277
		1973-1993	38.510238	-4.780717	-4.78916
		1983-2003	-29.992554	-4.861635	-4.796865
		1993-2013	-11.007995	-5.291508	-4.738953
		2003-2023	24.688724	-3.749938	-4.302826
2	1963-2025	-2.673267	-5.339213	-4.798138	
	1963-1983	88.124487	0.485877	-2.312996	
	1973-1993	-42.373443	-4.861106	-4.798697	
	1983-2003	-66.679785	-4.304673	-5.388708	
	1993-2013	-52.799651	-6.252624	-5.062956	
	2003-2023	-4.784644	-5.19661	-4.767763	
5	1963-2025	2438.235755	-4.40841	10.279414	
	1963-1983	-54.073769	-2.3271	-2.925639	
	1973-1993	54.923418	-4.614663	-4.37563	
	1983-2003	-110.746221	-0.641582	-3.270721	

	1993-2013	23007.892324	-4.420166	405.037661
	2003-2023	2327.368624	-4.628706	52.707782
3	1 1963-2025	-8.923047	-8.342482	-8.316274
	1963-1983	-5.314649	-8.280847	-8.281782
	1973-1993	-10.300183	-8.308301	-8.302503
	1983-2003	-2.406316	-8.300169	-8.319994
	1993-2013	-0.419906	-8.319842	-8.194227
	2003-2023	-3.493712	-8.412641	-8.185508
2	1963-2025	-6.179767	-8.332051	-8.304816
	1963-1983	-5.4852	-8.320075	-8.234235
	1973-1993	-8.93029	-8.307523	-8.296731
	1983-2003	28.525922	-7.925387	-7.4764
	1993-2013	-5.057214	-8.306454	-8.205388
	2003-2023	6.245208	-8.295844	-8.119038
5	1963-2025	25.315847	-8.303485	-8.153324
	1963-1983	80.827827	-8.145132	-5.364331
	1973-1993	-8.791554	-8.306774	-8.293809
	1983-2003	160.972484	-5.573307	-8.023605
	1993-2013	20.582393	-8.309451	-7.911833
	2003-2023	190.731657	-8.282685	-5.354255
5	1 1963-2025	-6.989102	-10.72921	-10.711984
	1963-1983	-318.541687	-10.740611	-14.805129
	1973-1993	-10.77598	-10.72398	-10.711167
	1983-2003	-11.416337	-10.721325	-10.704642
	1993-2013	-5.13662	-10.729205	-10.634146
	2003-2023	8.601293	-10.72499	-10.439808
2	1963-2025	-8.436649	-10.723461	-10.694462
	1963-1983	-7.20782	-10.730092	-10.679978
	1973-1993	-11.08305	-10.723819	-10.714401
	1983-2003	-9.906017	-10.7236	-10.700534
	1993-2013	-1.924427	-10.716661	-10.458018
	2003-2023	0.336696	-10.717627	-10.581157
5	1963-2025	-9.916105	-10.713972	-10.705597
	1963-1983	1.33291	-10.592708	-10.592767
	1973-1993	-7.920071	-10.723822	-10.670876
	1983-2003	3.275619	-10.556758	-10.589651
	1993-2013	39.095191	-10.723974	-10.027808
	2003-2023	1.240625	-10.724462	-10.542239

```
[6]: table_gamma_a_lam = cleaned_results["gamma_a_lam"].unstack(level=[0,1])
table_gamma_a = cleaned_results["gamma_a"].unstack(level=[0,1])
table_gamma_lam = cleaned_results["gamma_lam"].unstack(level=[0,1])
```

```
[7]: def add_p_values(table):
    table_numeric = table.apply(pd.to_numeric, errors="coerce")

    periods = table_numeric.index
```

```

columns = table_numeric.columns

new_rows = []
new_index = []

for period in periods:
    stats = table_numeric.loc[period].values.astype(float)

    new_rows.append(stats)
    new_index.append((period, "$\gamma$"))

    pvals = 2 * (1 - norm.cdf(np.abs(stats)))
    new_rows.append(pvals)
    new_index.append((period, "$p$"))

multi_index = pd.MultiIndex.from_tuples(new_index, names=["Period", "Type"])
new_table = pd.DataFrame(new_rows, index=multi_index, columns=columns)

return new_table

```

```
[8]: table_gamma_a_lam
```

```

[8]: K          1          3          \
R          1          2          5          1          2
Period
1963-1983 -19.455078  88.124487  -54.073769  -5.314649  -5.4852
1963-2025  44.021375  -2.673267  2438.235755  -8.923047  -6.179767
1973-1993  38.510238 -42.373443   54.923418 -10.300183  -8.93029
1983-2003 -29.992554 -66.679785 -110.746221  -2.406316  28.525922
1993-2013 -11.007995 -52.799651  23007.892324  -0.419906  -5.057214
2003-2023  24.688724  -4.784644  2327.368624  -3.493712   6.245208

K          5
R          5          1          2          5
Period
1963-1983  80.827827 -318.541687  -7.20782   1.33291
1963-2025  25.315847  -6.989102 -8.436649  -9.916105
1973-1993  -8.791554 -10.77598 -11.08305  -7.920071
1983-2003 160.972484 -11.416337 -9.906017   3.275619
1993-2013  20.582393  -5.13662 -1.924427  39.095191
2003-2023 190.731657   8.601293  0.336696   1.240625

```

```

[9]: table_gamma_a_lam = add_p_values(table_gamma_a_lam)
table_gamma_a = add_p_values(table_gamma_a)
table_gamma_lam = add_p_values(table_gamma_lam)

```



```

[ ]: latex_a_lam = table_gamma_a_lam.round(3).to_latex(
    multirow=True,
    multicolumn=True,
    index=True,
    escape=False,
    caption="Joint Homogeneity Test ( $\Gamma_{\alpha, \lambda}$ ) with
    ↪p-values",
    label="tab:gamma_a_lam_with_p",
    float_format="%.2f",
)

latex_a = table_gamma_a.round(3).to_latex(
    multirow=True,
    multicolumn=True,
    index=True,
    escape=False,
    caption="Intercept Homogeneity Test ( $\Gamma_{\alpha}$ ) with p-values",
    label="tab:gamma_a",
    float_format="%.2f",
)

latex_lam = table_gamma_lam.round(3).to_latex(
    multirow=True,
    multicolumn=True,
    index=True,
    escape=False,
    caption="Slope Homogeneity Test ( $\Gamma_{\lambda}$ ) with p-values",
    label="tab:gamma_lam",
    float_format="%.2f",
)

print(latex_a_lam)
print(latex_a)
print(latex_lam)

```