Assignment 2 – Antonin Beranger

Income distribution (ECON 473)

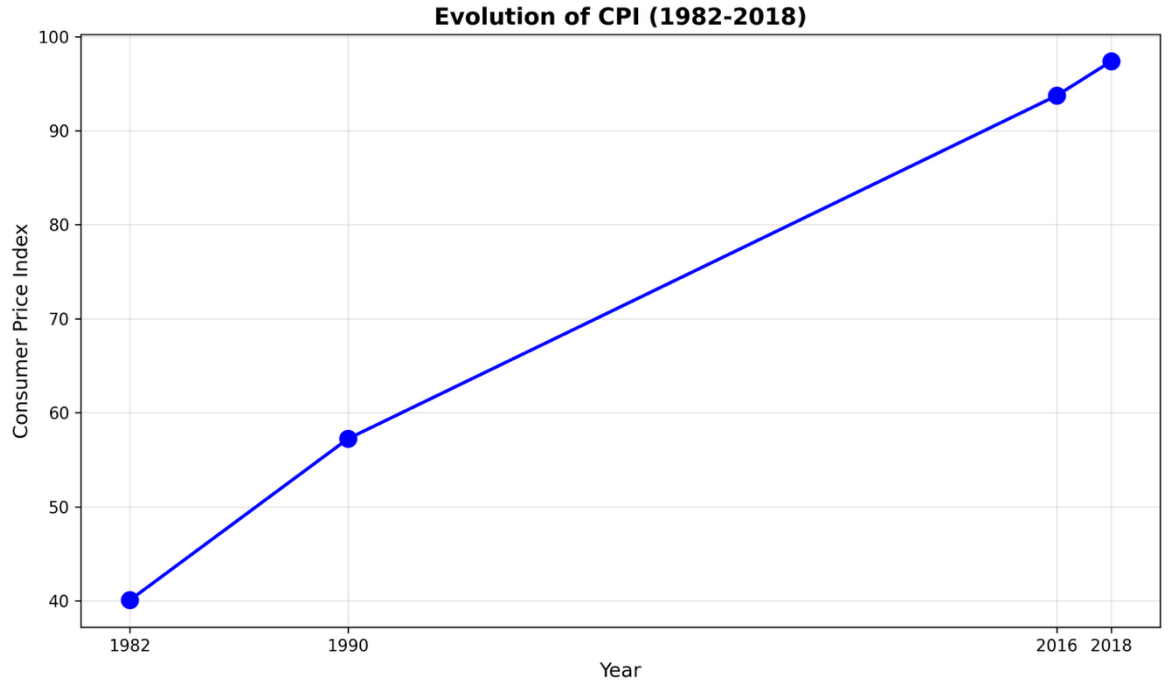
Question 1:

1982 – 1990:	2016-2018:
*Total Income: TOTINC	*Total Income: TTINC
*After-tax income: INCAFTTX	*After-tax income: ATINC
*Wages and Salaries: WAGSAL	*Wages and Salaries: WGSAL
*Hours worked: HRSWRK	*Hours worked: USHRWK
*Weeks worked: WKSWRK	*Weeks worked: WKSEM
*Education: EDUC	*Education: HLEV2G
*Age: AGE	*Age: AGEGP
*Gender: SEX	*Gender: SEX
*WEIGHTS: WEIGHT	*WEIGHTS: FWEIGHT

Hourly wage:

➔ df['hourly_wage'] = df[wage_var] / (df[hours_var] * df[weeks_var])

Question 2:



Real hourly wage in 2020 $:

- cpi = {1982: 40.07, 1990: 57.23, 2016: 93.72, 2018: 97.37}
- for y, df in datasets.items():
- df['hourly_real_wage'] = (df['hourly_wage'] / cpi[y]) * 100

Question 3:

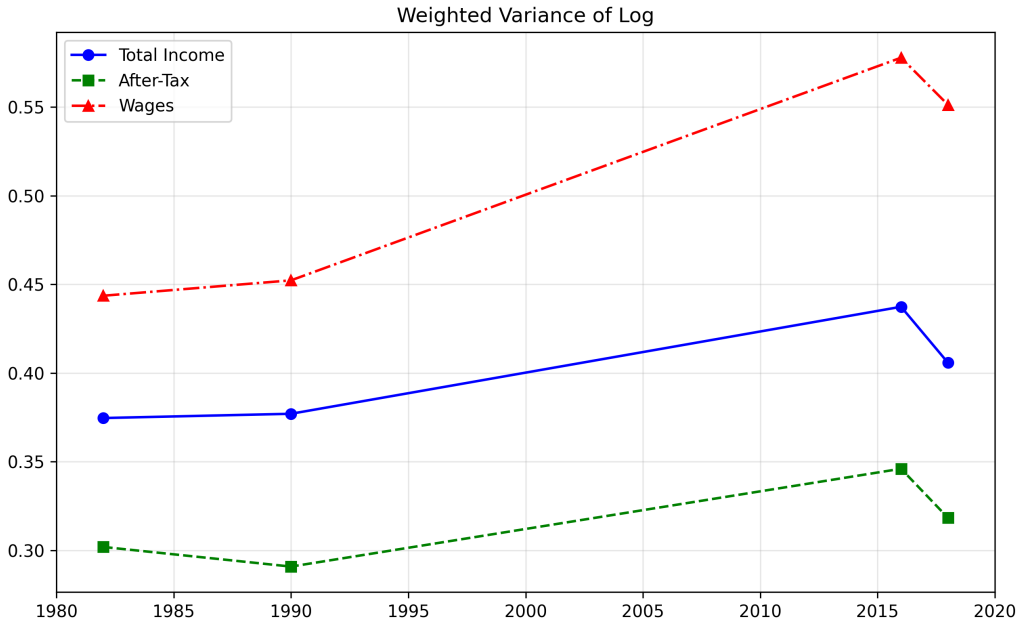
```
#question 3 - sample selection and replication of table 1
unfiltered_datasets = {year: df.copy() for year, df in datasets.items()}

min_wage_thresholds = [3.7375*0.5, 5.0500001*0.5, 11.3*0.5, 13.4125*0.5]
years = [1982, 1990, 2016, 2018]

for year, wage in zip(years, min_wage_thresholds):
    df = datasets[year]

    print("\n" + "="*65)
    print(f"Table 1: \n Sample Selection in the {year} Dataset")
    print("="*65)

    if year in [1982, 1990]:
        hours_var, weeks_var, age_var = 'HRSWRK', 'WKSWRK', 'AGE'
    else:
        hours_var, weeks_var, age_var = 'USHRWK', 'WKSEM', 'AGEGP'

    n_total = len(df)

    # Wage & Work Filter
    df_clean = df[
        (df['hourly_wage'].notna()) &
        (df['hourly_wage'] >= wage) &
        ((df[hours_var] * df[weeks_var]) > 260)
    ]
    n_after_wage = len(df_clean)
    drop1 = n_total - n_after_wage

    # Age Filter
    if year in [1982, 1990]:
        df_clean = df_clean[(df_clean[age_var] >= 25) & (df_clean[age_var] <= 60)]
    else:
        df_clean = df_clean[(df_clean[age_var] >= 7) & (df_clean[age_var] <= 14)]

    n_after_age = len(df_clean)
    drop2 = n_after_wage - n_after_age

    # table 1 replication
    summary = pd.DataFrame({
        'Step': ['Initial data', 'After wage filtering', 'After age filtering'],
        'Observations': [n_total, n_after_wage, n_after_age],
        'Dropped': [0, drop1, drop2],
        'Cumulative Dropped': [0, drop1, drop1 + drop2]
    })
    print(summary.to_string(index=False))
    print("="*65)

    datasets[year] = df_clean
```

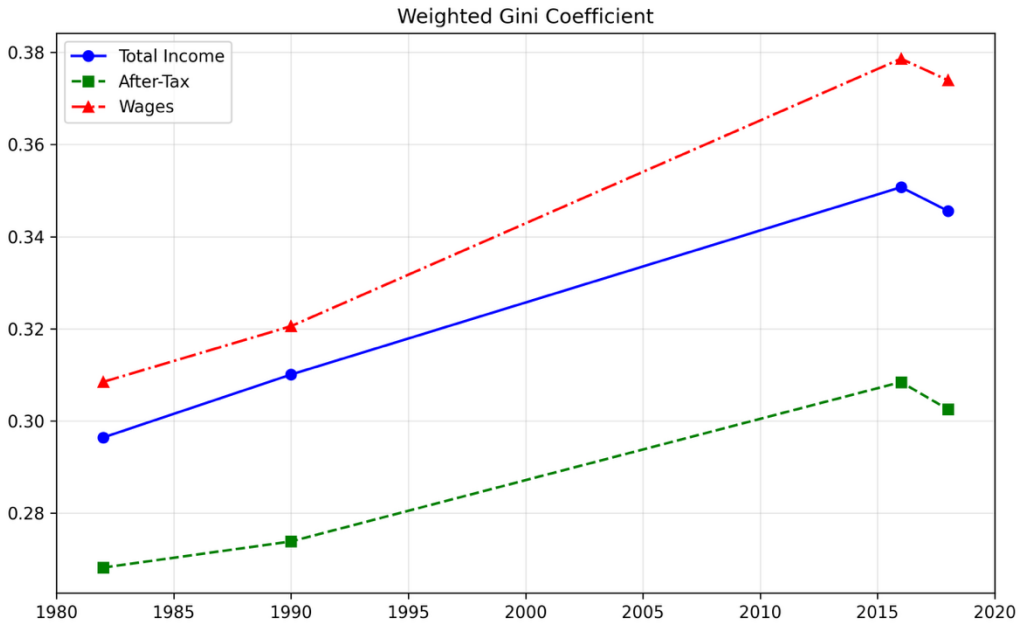
Table 1 reproduction from HPV

```
==========================================================================
Table 1:
 Sample Selection in the 1982 Dataset
==========================================================================
                Step  Observations  Dropped  Cumulative Dropped
        Initial data         42570        0                   0
After wage filtering         34256     8314                8314
 After age filtering         25940     8316               16630
==========================================================================


==========================================================================
Table 1:
 Sample Selection in the 1990 Dataset
==========================================================================
                Step  Observations  Dropped  Cumulative Dropped
        Initial data         46510        0                   0
After wage filtering         38145     8365                8365
 After age filtering         31350     6795               15160
==========================================================================


==========================================================================
Table 1:
 Sample Selection in the 2016 Dataset
==========================================================================
                Step  Observations  Dropped  Cumulative Dropped
        Initial data         34287        0                   0
After wage filtering         28327     5960                5960
 After age filtering         23517     4810               10770
==========================================================================


==========================================================================
Table 1:
 Sample Selection in the 2018 Dataset
==========================================================================
                Step  Observations  Dropped  Cumulative Dropped
        Initial data         50588        0                   0
After wage filtering         41807     8781                8781
 After age filtering         32469     9338               18119
==========================================================================
```

Question 4:

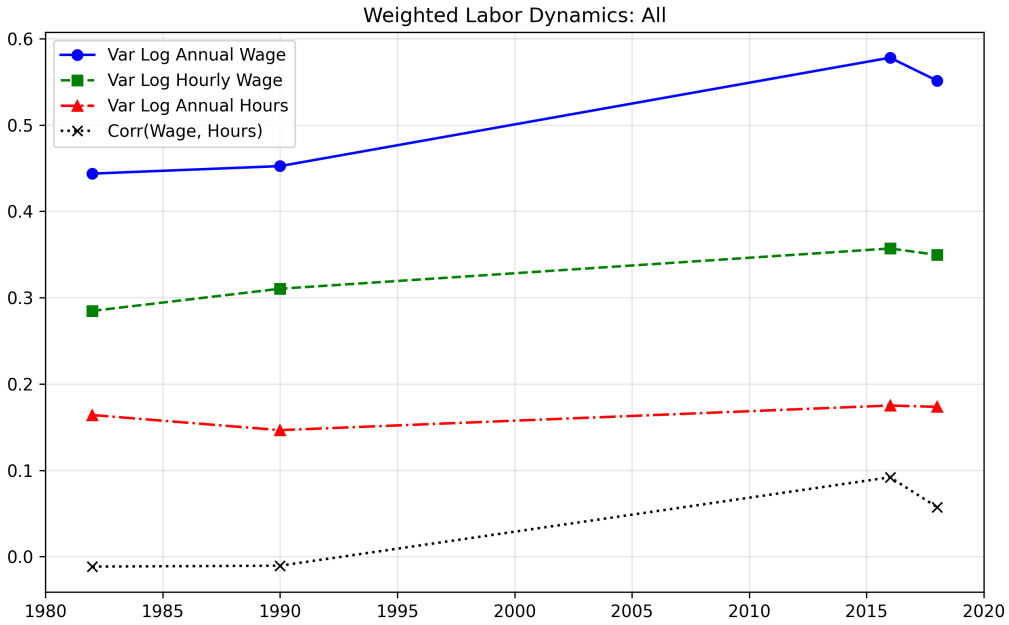
Part A)



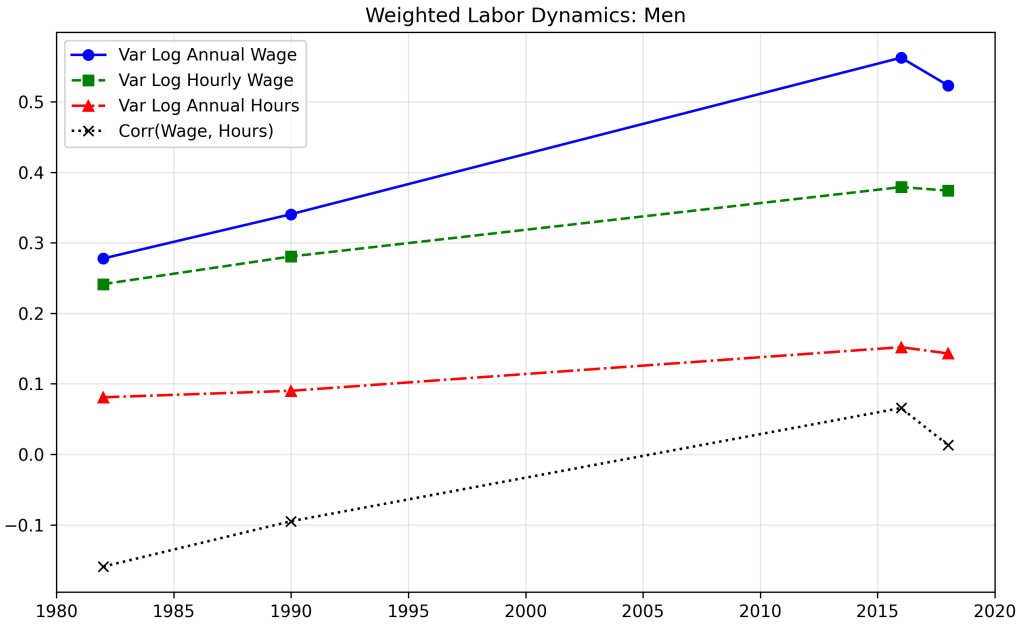


Part B)

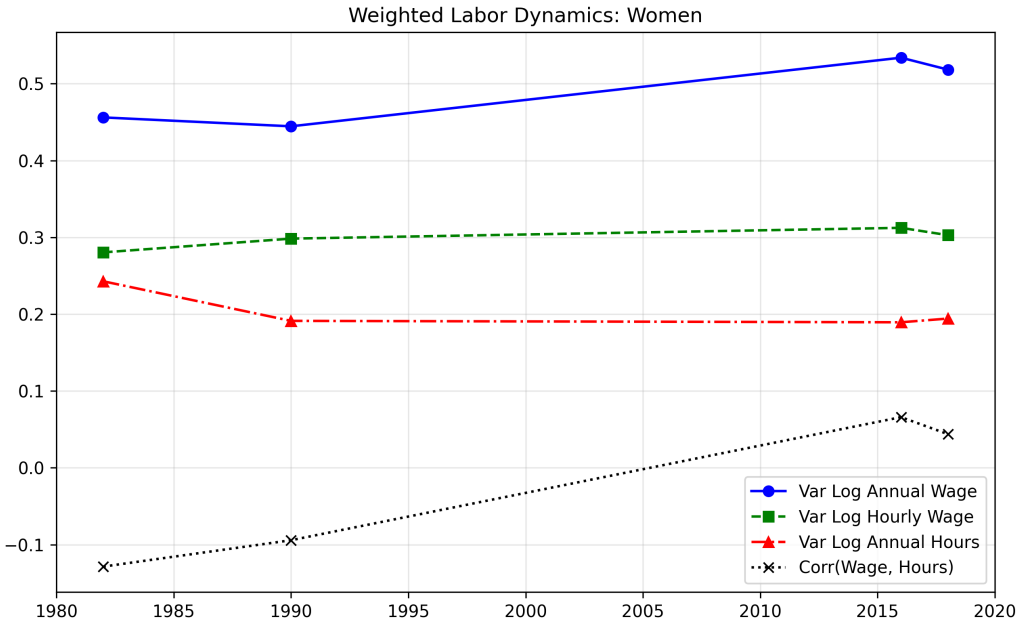
Total Population:



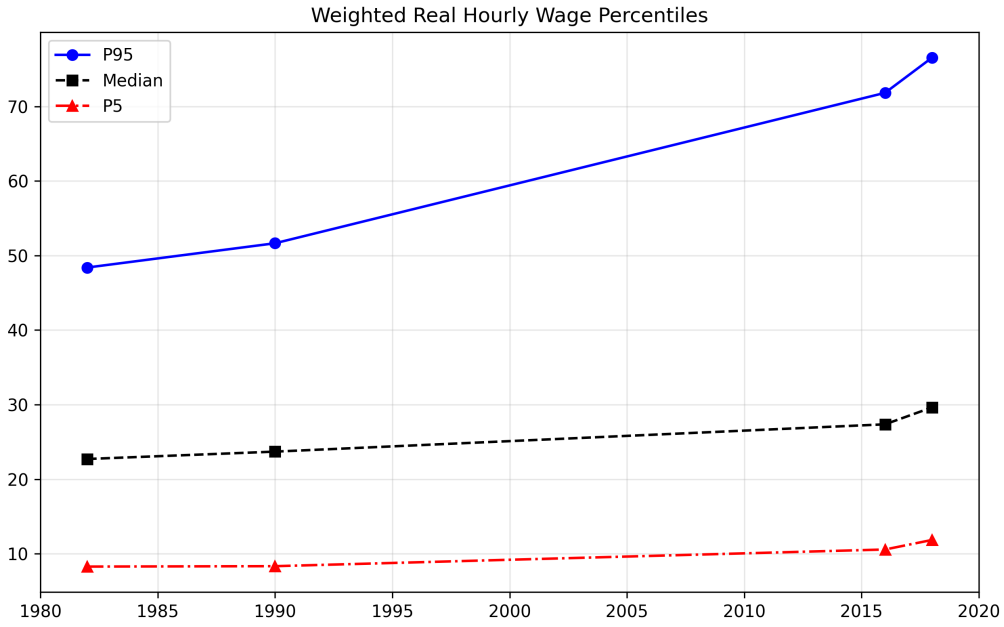
Men:



Women:


Weighted Labor Dynamics: Women
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
1980
1985
1990
1995
2000
2005
2010
2015
2020
Var Log Annual Wage
Var Log Hourly Wage
Var Log Annual Hours
Corr(Wage, Hours)

Part C)


Weighted Real Hourly Wage Percentiles
P95
Median
P5
70
60
50
40
30
20
10
1980
1985
1990
1995
2000
2005
2010
2015
2020

Question 5:

A) df['potential_experience'] = df['age_approx'] - df['years_of_schooling'] – 6

*all variables were defined in the function 'prep_composition_adjusted_vars(df, year)' for all datasets

B)

```
#5-B
# Verify Group Consistency
print("\nVerifying Gender x Education x Experience Groups:")
all_unique_cells = set()

for year, df in akk_datasets.items():
    # Get unique cells for this year
    unique_cells_year = set(df['akk_cell'].unique())
    all_unique_cells.update(unique_cells_year)
    df_valid_groups = df.dropna(subset=['akk_cell'])
    num_groups = df_valid_groups['akk_cell'].nunique()

    print(f"Year {year}: {num_groups} populated groups out of 40 possible.")

print(f"\nTotal unique group identifiers found across ALL years:
{len(all_unique_cells)}")
print(f"Sample Group Names: {list(all_unique_cells)[:3]}")
```

Total number of groups per year:

```
Year 1982: 24 populated groups out of 40 possible.
Year 1990: 23 populated groups out of 40 possible.
Year 2016: 30 populated groups out of 40 possible.
Year 2018: 30 populated groups out of 40 possible.
```

C & D)

```
group_stats = []

for year, df in akk_datasets.items():
    # Only keep valid wages and positive weights
    df_valid = df[(df['hourly_real_wage'] > 0) & (df['weight'] > 0)].copy()

    # Calculate log real hourly wage
    df_valid['log_wage'] = np.log(df_valid['hourly_real_wage'])
    df_valid['total_hours_weight'] = df_valid['weight'] * df_valid['annual_hours']

    total_hours_year = df_valid['total_hours_weight'].sum()

    # Group by the AKK cell
    grouped = df_valid.groupby('akk_cell')

    for cell_name, group in grouped:
        # C: Share of sample (using hours-weighted metric as per AKK)
```

```
        group_hours = group['total_hours_weight'].sum()
        share = group_hours / total_hours_year

        # D: Compute mean log wage in group (Weighted)
        mean_log_wage = np.average(group['log_wage'],
weights=group['total_hours_weight'])

        # Split cell name to get gender and education back
        gender, edu, exp = cell_name.split('_')

        group_stats.append({
            'Year': year,
            'akk_cell': cell_name,
            'Gender': gender,
            'Education': edu,
            'Share': share,
            'Mean_Log_Wage': mean_log_wage
        })

df_stats = pd.DataFrame(group_stats)
```

E)

```
--- Composition-Corrected vs Naive Mean Log Wages ---

TOTAL POPULATION:
 Year  Naive Mean Log Wage  Corrected Mean Log Wage
 1982             3.073104                 3.141894
 1990             3.118680                 3.126706
 2016             3.342218                 3.243272
 2018             3.408005                 3.311638

WOMEN:
 Year  Naive Mean Log Wage  Corrected Mean Log Wage
 1982             2.843850                 2.945552
 1990             2.938551                 2.968465
 2016             3.236435                 3.115962
 2018             3.307347                 3.191370

MEN:
 Year  Naive Mean Log Wage  Corrected Mean Log Wage
 1982             3.187091                 3.272802
 1990             3.236770                 3.238209
 2016             3.424692                 3.326145
 2018             3.485395                 3.389928

UNIVERSITY GRADUATES:
 Year  Naive Mean Log Wage  Corrected Mean Log Wage
 1982             3.213555                 3.275485
 1990             3.131780                 3.131849
 2016             3.427771                 3.424481
 2018             3.487554                 3.492898

HIGH-SCHOOL GRADUATES:
 Year  Naive Mean Log Wage  Corrected Mean Log Wage
 1982             3.070456                 3.056962
 1990             3.118312                 3.123708
 2016             3.158992                 3.148675
 2018             3.212790                 3.217292
```

F)

Over the sample period, naive wage growth consistently overstates true wage growth because they capture a workforce that became significantly older and more educated. By holding these demographics constant, composition-adjusted wages reveal much smaller “true” wage gains. This adjustment heavily dampens the apparent wage growth for women, who experienced massive educational gains and exposes severe real wage stagnation for high-school graduates, while confirming a rising premium for university graduates.