Panel Data Analysis of Microeconomic Decisions
Assignment 1

Mike Weltevrede (1257560)

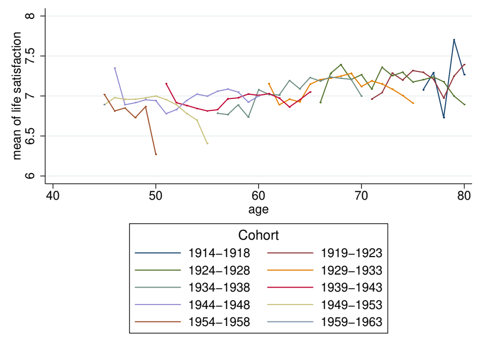
October 25, 2019

Tables and code are added to the appendix at the end of the document.

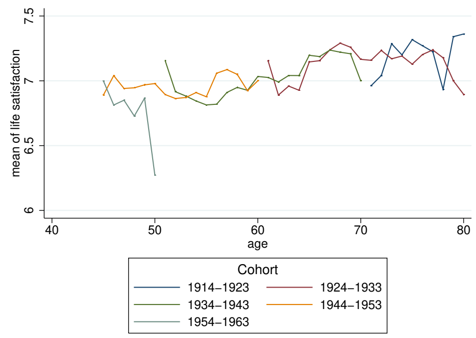
1 Profiles for different cohorts

1.a

In this section, we plot the mean life satisfaction against age while grouping several cohorts together. We define a cohort as people who are born in the same year, so this is computed as the difference between `year` and `age`. In Figure 1a, we see the spaghetti plot for grouping 5 cohorts together, while Figure 1b shows the spaghetti plot for combining cohorts in groups of 10.



(a) Grouping 5 cohorts together



(b) Grouping 10 cohorts together

Figure 1: Spaghetti plots

1.b

We want to see if there are specific cohort effects for our data, by which we mean whether there are substantial enough differences between various cohorts. When we receive similar results for different cohorts, then using these different cohorts in our analysis (such as a regression) does not give us extra explanatory power, so it is not useful to use cohorts. Vice versa, if results for different cohorts differ, then they provide explanatory power about differences between cohorts, so it will be useful to add them to our analysis.

Consider the plots in Figure 1. For intermediate ages (for example 60-70), the differences are not that large between different cohorts. However, there are some substantial differences, such as at age 50, 55, and around age 75 (where we see that the cohort from 1929-1933 has a bit of a drop compared to the others). For the former two, we note that the satisfaction of life drops around 2007-2008 (adding the respective age of 50 and 55 to the cohort interval). Notice that this is the year that the financial crisis started to have larger effects. It could be possible that this is a reason why the life satisfaction is dropping, independent of the cohort. In addition, notice two specific cohorts that behave differently from the rest, namely the first (1914-1918) and the last (1959-1963). There are larger drops and spikes than other cohorts, which are more similar to one another.

1.c

We perform three regressions to see if we need to control for a time trend:

```
reg s_life age year $cohort_5_dummies $cohort_10_dummies // all dummies
reg s_life age year $cohort_5_dummies
reg s_life age year $cohort_10_dummies
```

where *cohort_5_dummies* and *cohort_10_dummies* are macros representing the dummies for cohort groupings of size 5 and cohort groupings of size 10, respectively. The results of these are found in Tables 8, 9, and 10, respectively. Results for omitted variables due to avoiding multicollinearity are left out of the regression tables. It makes most sense to us to regard the latter two regressions, seeing as the former introduces a lot of multicollinearity due to overlapping groups.

In the regression including only the cohort groupings of size 5, we see that there is no statistical evidence for an effect of cohorts on the (average) satisfaction of life, not even at a significance level of 10%. However, when considering the cohort groupings of size 10, we see that 3 out of 4 cohorts have a statistically significant (negative) effect, namely the cohorts 1914-1923, 1924-1933, and 1934-1943, with the cohorts 1944-1953 having no statistically significant effect and the cohort 1954-1963 not being included due to multicollinearity. This would mean that including cohorts of size 10 to account for the cohort effects would likely be a good idea, as significant differences can be found (compared to the control group, being the cohort of 1954-1963). As such, we can conclude that there is a time trend to be taken into account.

2 One draw of simulated data

2.a

We generate artificial data and want to explain the DGP line by line.

```
set seed 345398
```

This line sets a random starting seed. This means that the random number generator will generate the same random numbers the next time when running the code. In essence, the numbers that we generate are a function of this seed, so that the same seed generates the same set of random numbers.

```
drawnorm alpha_i , n(200)
```

The `drawnorm` function in combination with `n(200)` draws 200 random numbers, in this case storing them in the variable `alpha_i`, according to the Gaussian distribution. Since no parameters on the mean and standard deviation is stated, this is the standard Normal distribution.

```
expand 5
```

`expand 5` is used to create copies of the existing data 5 times. Seeing as `alpha_i` represents the individual unobserved effect, this would be used to specify that our dataset will be a panel data set with 5 time periods (5 observations per individual).

```
drawnorm nu_it e_it , n(1000)
```

Again, we create random standard Gaussian variables, in this case 1000 of them. They are stored in `nu_it` and `e_it`.

```
gen x_it = nu_it + alpha_i
```

The variable `x_it` is generated with the `gen` function and is defined as the sum of `nu_it` and `alpha_it`.

```
drop nu_it
```

After creating the `x_it` variable, `nu_it` is not needed anymore. The `drop` command removes `nu_it` from the dataset.

```
gen y_it = 3 + alpha_i + 2*x_it + e_it
```

Lastly, we use the `gen` command again to create the dependent variable `y_it`, defined as the linear combination specified. This is created according to the linear model:

$$
\begin{aligned}
y_{it} &= \beta_0 + \beta_1 x_{it} + \alpha_i + e_{it} \\
&=: \beta_0 + \beta_1 x_{it} + u_{it}
\end{aligned}
$$

2.b

Here, we find the all pairwise correlation coefficients in a matrix using the `pwcorr` command. The option `sig` gives the significance level of each correlation coefficient. In Table 1 we give the results of running this command on our data.

Table 1: Correlation matrix (significance level in parentheses)

	alpha_i	e_it	x_it	y_it
alpha_i	1			
e_it	0.0203	1		
	(0.5213)			
x_it	0.7232	-0.0069	1	
	(0.0000)	(0.8268)		
y_it	0.8169	0.2639	0.9451	1
	(0.0000)	(0.0000)	(0.0000)	

We can see that, as expected, there is a high (positive) correlation between `alpha_i` and `x_it`, as well as between `y_it` and both `alpha_it` and `x_it`. This is because there is a clear positive linear relationship between the variables, as that is how they were simulated in the DGP. Since there is a correlation of 0.9451 between `x_it` and `y_it`, being positive, along with a positive impact of `alpha_i` on both `x_it` and `y_it` we expect the OLS results from regressing `y_it` on `x_it` to be upwards biased.

2.c

We know that OLS will not be unbiased for panel data models, including an individual specific variable in α_i. Consider the covariance between `x_it` and `y_it`:

$$\begin{aligned}
\mathrm{Cov}(x_{it}, y_{it}) &= \mathrm{Cov}(\nu_{it} + \alpha_i, 3 + \alpha_i + 2x_{it} + e_{it}) && (1)\\
&= \mathrm{Cov}(\nu_{it} + \alpha_i, 3 + 3\alpha_i + 2\nu_{it} + e_{it}) && (2)\\
&= 2\mathrm{Var}(\nu_{it}) + 3\mathrm{Var}(\alpha_i) && (3)
\end{aligned}$$

In (1), we simply fill in $x_{it} = \nu_{it}$ and $y_{it} = 3 + \alpha_i + 2x_{it} + e_{it}$. Subsequently, in (2), we fill in x_{it} again, but this time in the formula for y_{it}. Finally, in (3), we compute this covariance. To compute this, notice that, by assumption:

$$\mathrm{Cov}(\alpha_i, e_{it}) = \mathrm{Cov}(\nu_{it}, e_{it}) = \mathrm{Cov}(\nu_{it}, \alpha_i) = 0 \qquad \forall i, t$$

i.e. there is no correlation between these terms.

Notice that the data for `y_it` and `x_it` is generated dependent on `alpha_i`. As such, in the regression from 2b, there is bias in the OLS estimate. If we regress `y_it` on `x_it` and `alpha_i`, this dependency is explained now by `alpha_i`. Taking the assumption that `alpha_i` and `e_it` are mutually independent and that they are independent of `x_it` into account, we get that OLS is indeed unbiased and consistent. However, note that the individual specific `alpha_i` does not change over time, so there is autocorrelation present in the model. Therefore, the standard errors computed by OLS are not correct and can be estimated well by a GLS model. As such, OLS is not a good method for this data.

3 Many draws of simulated data

We want to generate data many times and run a regression for each new data set that we create. The following code is used for this:

```
set seed 345398
capture program drop mcprog
program mcprog
        clear
        drawnorm alpha_i, n(200)
        expand 5
        drawnorm nu_it e_it, n(1000)
        gen x_it = nu_it + alpha_i
        drop nu_it
        gen y_it = 3 + alpha_i + 2*x_it + e_it
        regress y_it x_it
end
simulate _b _se, reps(100): mcprog
sum
```

3.a

Running the simulation study above, we get the results as in Table 2.

Table 2: Summary results from the simulation study

Variable	Obs	Mean	Std. Dev.	Min	Max
_b_x_it	100	2.503027	.0336145	2.413495	2.581013
_b_cons	100	3.003006	.0520344	2.90234	3.125339
_se_x_it	100	.0272081	.0009898	.0246237	.0307716
_se_cons	100	.0387386	.0008656	.0366121	.0414571

Note that the standard deviation of `_b_x_it` across simulated samples (0.0336145) is relatively a lot higher than the mean of the standard error `_se_x_it`, namely 0.0272081 (23% higher). This is because the variance-covariance matrix is not defined correctly. More explicitly, since there is autocorrelation, the covariance matrix is nondiagonal. Recall that the variance of $\hat{\beta}$ is:

$$\begin{aligned}\mathrm{Var}(\hat{\beta}) &= \mathrm{Var}((X'X)^{-1}X'\epsilon \mid X) \\ &= (X'X)^{-1}X'\mathrm{Var}(\epsilon \mid X)X(X'X)^{-1} \\ &= \sigma^2(X'X)^{-1}X'\Omega X(X'X)^{-1}.\end{aligned}$$

As such, since under autocorrelation $\Omega \neq I$, this does not reduce to $\sigma^2(X'X)^{-1}$ and there are invalid standard errors and similarly related results.

An improvement could be made with clustered standard errors. In this case, the autocorrelation is taken into account when constructing the covariance matrix. When trying this, we get the results from Table 3.

Table 3: Summary results from the simulation study, with clustering

Variable	Obs	Mean	Std. Dev.	Min	Max
_b_x_it	100	2.503027	.0336145	2.413495	2.581013
_b_cons	100	3.003006	.0520344	2.90234	3.125339
_se_x_it	100	.0348017	.0028776	.0278422	.0427458
_se_cons	100	.0497832	.0020862	.0446874	.056195

In these results, we see that the standard deviation of _b_x_it is 0.0336145, where the mean standard deviation _se_x_it is 0.0348017. This is almost identical to one another. This is due to taking personal clusters into account. Now, this means that some sort of shift in some individual's status (i.e. an external effect, such as a health issue) only has an effect on that individual. Before clustering, this was perpetuated to the other time periods that this individual was observed and it was regarded as an effect on multiple observations (people) in the dataset.

3.b

Note that the regression that we do it only of `y_it` on `x_it`,i.e. we do not include the individual specific effect `alpha_i`. As deduced in question 2c, since both `y_it` and `x_it` depend on `alpha_i`, i.e. there is a causal effect on the regressor as well as the regressand, there is bias in the OLS estimator. The lack of `alpha_i` means that now our regressor `x_it` has to account for the effect that `alpha_i` has on `y_it`. As such, the value for _b_x_it is not equal to 2 (which is the true value according to the DGP we used). Therefore, when this regression model is used to predict new data, it is not precise and will not yield a proper value.

4 Fixed effects and first differences estimation

We are interested in applying the fixed effects (FE) and the first-difference (FD) estimator to the data generated as in question 2. We first discuss what the models look like and under which conditions they provide consistent estimation results.

For the FE estimator, the model uses a within transformation to eliminate. It looks like:

$$y_{it} - \bar{y}_i = (x_{it} - \bar{x}_i)'\beta + (u_{it} - \bar{u}_i), \tag{4}$$

where the variables with a bar are the time averages of the respective variable. Indeed, since α_i is time-constant, it is eliminated from the model. Note that also other time-constant properties are eliminated, however, such as personal properties like age at entry or sex. Equation (4) is then used to perform the regression. This model yields consistent results under some assumptions on the error term u_{it}. Most importantly, we need strict exogeneity:

$$\mathrm{E}[u_{it} \mid x_i, \alpha_i] = 0,\ t = 1, \ldots, T,\ x_i \coloneqq (x_{i1}, \ldots, x_{iT}). \tag{5}$$

This means that the x_{it} in each time period are uncorrelated with u_{it} in every time period. The other assumptions are similar to the simple model, such as independence of the data as well as the individual effect with the error term. The results from this regression are found in Table 4.

Table 4: Summary results for the FE estimator simulation

Variable	Obs	Mean	Std. Dev.	Min	Max
_b_x_it	100	2.002906	.0362488	1.906549	2.0905
_b_cons	100	3.0065	.0777624	2.803353	3.213826
_se_x_it	100	.0352607	.0010911	.0317797	.0382965
_se_cons	100	.0317179	.0007524	.0300808	.0335437

We see that, again, the standard deviation of _b_x_it is quite close to the mean of _se_x_it.

The FD estimator relies on subtracting the one-period lag instead of the mean data. It also relies on the strict exogeneity assumption in (5). Note that this gives consistency due to the following:

$$\mathrm{E}[(x_{it} - x_{i,t-1})(u_{it} - u_{i,t-1})] = 0,\ t = 2, \ldots, T.$$

Working out the parentheses indeed shows that this is the case, under strict exogeneity. The model then becomes:

$$y_{it} - y_{i,t-1} = (x_{it} - x_{i,t-1})'\beta + (u_{it} - u_{i,t-1}),\ t = 2, \ldots, T. \tag{6}$$

The results from running the regression on (6) are found in Table 5.

Table 5: Summary results from the FD estimator simulation

Variable	Obs	Mean	Std. Dev.	Min	Max
_b_x_it	100	1.999952	.0415777	1.886875	2.110715
_b_cons	100	-.0011509	.0234278	-.0538805	.0588437
_se_x_it	100	.0352389	.0012898	.0324049	.0393236
_se_cons	100	.0499387	.0014099	.0463558	.054004

Also here we see that the standard deviation of _b_x_it is quite close to the mean of _se_x_it. As such, both estimator are good in capturing this due to dealing with the nature of the dataset, excluding the individual α_i. With regards to which estimator would be best for this DGP, we notice that the FE estimator overestimates the coefficient for x_{it} by 0.002906 and that the FD estimator underestimates it by 0.000048. As such, it seems that the FD estimator gives a better estimate. However, also note that the FD estimator gives a higher standard deviation, indicating some sense of less precision. Even though the difference is small, the FE estimator is to be preferred. Also note that, since different results are produced by the two estimators, it is likely that strict exogeneity does not hold.

Now, we would like to know when correct standard errors are obtained for the FE and FD estimators. Recall that the FD estimator represents the opposite extreme of the FE estimator in the sense that the first differences of the idiosyncratic error terms e_{it} are serially uncorrelated (i.e. they constitute a random walk for u_{it}) and that they have constant variance. That is:

$$\mathrm{E}[e_{it}e_{it}' \mid x_{i1}, \ldots, x_{iT}, \alpha_i] = \sigma_e^2,\ t = 2, \ldots, T \tag{7}$$

where $e_{it} := \Delta u_{it} - u_{i,t-1},\ t = 2, \ldots, T$. Then, a consistent estimate for σ_e^2 is:

$$\hat{\sigma}_e^2 = \frac{1}{N(T-1)-K} \sum_{i=1}^{N} \sum_{t=1}^{T} \hat{e}_{it}^2, \tag{8}$$

where $\hat{e}_{it}^2 := \Delta y_{it} - \Delta x_{it} \hat{\beta}_{FD}$ are the OLS residuals from the regression of (6).

For the FE estimator, we have that the consistent estimator for σ_u^2 is given by:

$$\hat{\sigma}_u^2 = \frac{1}{N(T-1)-K} \sum_{i=1}^{N} \sum_{t=1}^{T} \Big((y_{it} - \bar{y}_i) - (x_{it} - \bar{x}_i)\hat{\beta}_{FE} \Big)^2. \tag{9}$$

This is provided under the following additional assumptions:

- Constant variances across time: $\mathrm{E}[u_{it}^2] = \sigma_u^2,\ t = 1, \ldots, T$
- Serially uncorrelated: $\mathrm{E}[u_{it} u_{is}] = 0$ for all $t, s = 1, \ldots, T$ and $t \neq s$.

5 Dynamic model

To create a dynamic model, we include the one-period lagged $y_{i,t-1}$ in the model. In the DGP, we define no effect of $y_{i,t-1}$ on y_{it}, i.e. the coefficient for $y_{i,t-1}$ is equal to zero. We then apply the FE estimator in a Monte Carlo simulation as before. We do this for 5, 10, 20, and 50 time periods and for true values of the coefficient of 0.5. We get the following results as in Table 6.

Table 6: Summary results from the bias of the FE estimate

t	Obs	Mean	Std. Dev.	Min	Max
5	100	-.2833054	0.121162	-.308199	-.2503623
10	100	-.1834849	.0109706	-.2112846	-.1562011
20	100	-.1148717	.0094153	-.1420249	-.0963236
50	100	-.0569837	.0055699	-.0711013	-.0450554

As expected, the bias decreases as T increases, however it does not disappear. Even at $T = 50$, there is still a bias of 3%. This is not substantial, nonetheless. The FE estimator is biased is because of the correlation between y_{it} and its lag. For simplification purposes, consider the model that regresses y_{it} on its lagged component. That is:

$$y_{it} = \delta_{FE} y_{i,t-1} + \alpha_i + e_{it}.$$

Using the FE estimator for this model gives the following estimate for δ:

$$\hat{\delta}_{FE} = \delta + \frac{\sum_{i=1}^{N} \sum_{t=1}^{T} (e_{it} - \bar{e}_i)(y_{i,t-1} - \bar{y}_{i,-1})}{\sum_{i=1}^{N} \sum_{t=1}^{T} (y_{i,t-1} - \bar{y}_{i,-1})^2}, \tag{10}$$

where $\bar{y}_{i,-1}$ refers to the average dependent variable for individual i over all time periods except for $T = 1$. In equation (10), note that the second part does not converge to zero. As such, $\mathrm{E}[\hat{\delta}_{FE}] \neq \delta_{FE}$ and the FE estimator would be biased. Note that the expression above does not change when other independent variables get added.

6 Instrumental variables estimation

In the last part, we want to use the Arellano-Bond (AB) estimator to estimate the effect of the lagged dependent variable for $t = 5$ time periods. Consider the following model equation:

$$y_{it} - y_{i,t-1} = \gamma(y_{i,t-1} - y_{i,t-2}) + u_{it} - u_{i,t-1}, \ t = 2, \ldots, T.$$

The AB estimator exploits moment conditions varying with t. It takes an instrumental variable approach where we use as (internal) instruments for $(y_{i,t-1} - y_{i,t-2})$ the following:

All $y_{i,t-2-j}, \ j = 0, \ldots, J$ which satisfy:

- $\mathrm{E}[(u_{it} - u_{i,t-1})y_{i,t-2-j}] = 0$
- $\mathrm{E}[(y_{i,t-1} - y_{i,t-2})y_{i,t-2-j}] \neq 0$

The results from this are found in Table 7.

Table 7: Summary results from the Arellano-Bond estimator simulation

Variable	Obs	Mean	Std. Dev.	Min	Max
_sim_1	100	.3526171	.0215123	.3038269	.4037142
_b_x_it	100	1.858366	.0518794	1.732158	1.967276
_b_cons	100	3.671628	.139697	3.311198	4.118244
_sim_4	100	.0173606	.0007921	.0154896	.0193629
_se_x_it	100	.0476555	.0016563	.0433666	.0517161
_se_cons	100	.0865253	.0066716	.0725158	.1086359

Recall that an instrument zit should satisfy two requirements:

- Relevance: $\mathrm{Cov}(x_{it}, zit) \neq 0, \ \forall i, t$
- Exogeneity: $\mathrm{Cov}(u_{it}, zit) = 0, \ \forall i, t$

The AB estimator is consistent for the coefficient of the lagged dependent variable because of the following reasoning. Consider the IV estimator with instruments as described above in the matrix Z. The IV estimator then is given by:

$$\begin{aligned}\hat{\beta}_{IV} &= (Z'X)^{-1}Z'(X\beta + u) \\ &= \beta + (Z'X)^{-1}Z'u \\ &= \beta + \left(\frac{Z'X}{n}\right)^{-1}\frac{Z'u}{n}\end{aligned}$$

For $\hat{\beta}_{IV}$ to be consistent, we need that $plim\,\hat{\beta}_{IV} = \beta$. By exogeneity of the instruments, we have that $plim\,\frac{Z'u}{n} = 0$. By relevance, we have that $plim\,\left(\frac{Z'X}{n}\right) = E(Z'X)$, which is finite. Therefore,

$$\begin{aligned}plim\,\hat{\beta}_{IV} &= plim\,\beta + \left(\frac{Z'X}{n}\right)^{-1}\frac{Z'u}{n} \\ &= \beta + plim\left(\frac{Z'X}{n}\right)^{-1} \times plim\,\frac{Z'u}{n} \\ &= \beta.\end{aligned}$$

A Tables

Table 8: Regression table using all cohort dummies (*cohort_1924_c5, cohort_1934_c5, cohort_1959_c5, cohort_1914_c10, cohort_1944_c10*, and *cohort_1954_c10* omitted due to collinearity)

Source	SS	df	MS		
Model	426.881855	11	38.8074414	Number of obs	= 29,841
Residual	97870.2281	29,829	3.28104288	F(11, 29836)	= 11.83
Total	98297.1099	29,840	3.29413907	Prob > F	= 0.0000
				R-squared	= 0.0043
				Adj R-squared	= 0.0040
				Root MSE	= 1.8114

s_life	Coef.	Std. Err.	t	$P > \|t\|$	[95% Conf.	Interval]
age	.0233146	.0075627	3.08	0.002	.0084915	.0381378
year	-.0195396	.0082081	-2.38	0.017	-.0356278	-.0034514
cohort_1914_c5	-.5808521	.3872992	-1.50	0.134	-1.339975	.1782711
cohort_1919_c5	-.5224744	.3296013	-1.59	0.113	-1.168507	.1235585
cohort_1929_c5	.0302908	.0604532	0.50	0.616	-.0882001	.1487816
cohort_1939_c5	-.0295462	.0492908	-0.60	0.549	-.1261583	.0670659
cohort_1944_c5	-.2063494	.1879713	-1.10	0.272	-.5747813	.1620825
cohort_1949_c5	-.1617778	.1727053	-0.94	0.349	-.5002878	.1767321
cohort_1954_c5	-.122746	.1642332	-0.75	0.455	-.4446502	.1991583
cohort_1924_c10	-.4216132	.2965595	-1.42	0.155	-1.002883	.1596564
cohort_1934_c10	-.3209727	.2350033	-1.37	0.172	-.7815894	.1396441
_cons	45.00124	16.14547	2.79	0.005	13.35541	76.64708

Table 9: Regression table using cohort dummies grouping 5 cohorts (*cohort_1914_c5* omitted due to collinearity)

Source	SS	df	MS
Model	426.881855	11	38.8074414
Residual	97870.2281	29,829	3.28104288
Total	98297.1099	29,840	3.29413907

Number of obs	= 29,841
F(11, 29836)	= 11.83
Prob > F	= 0.0000
R-squared	= 0.0043
Adj R-squared	= 0.0040
Root MSE	= 1.8114

s_life	Coef.	Std. Err.	t	$P > \|t\|$	[95% Conf.	Interval]
age	.0233146	.0075627	3.08	0.002	.0084915	.0381378
year	-.0195396	.0082081	-2.38	0.017	-.0356278	-.0034514
cohort_1919_c5	.0583777	.1640287	0.36	0.722	-.2631256	.379881
cohort_1924_c5	.1592389	.1695722	0.94	0.348	-.17313	.4916078
cohort_1929_c5	.1895297	.1868957	1.01	0.311	-.1767941	.5558535
cohort_1934_c5	.2598794	.2119456	1.23	0.220	-.1555431	.6753019
cohort_1939_c5	.2303332	.2370165	0.97	0.331	-.2342295	.6948959
cohort_1944_c5	.3745027	.2693082	1.39	0.164	-.1533532	.9023586
cohort_1949_c5	.4190743	.298223	1.41	0.160	-.1654558	1.003604
cohort_1954_c5	.4581061	.3315473	1.38	0.167	-.1917411	1.107953
cohort_1959_c5	.5808521	.3872992	1.50	0.134	-.1782711	1.339975
_cons	44.42039	15.84646	2.80	0.005	13.36064	75.48014

Table 10: Regression table using cohort dummies grouping 10 cohorts (*cohort_1954_c10* omitted due to collinearity)

Source	SS	df	MS
Model	417.179186	6	69.5298643
Residual	97879.9308	29,834	3.28081822
Total	98297.1099	29,840	3.29413907

Number of obs	= 29,841
F(11, 29836)	= 21.19
Prob > F	= 0.0000
R-squared	= 0.0042
Adj R-squared	= 0.0040
Root MSE	= 1.8113

s_life	Coef.	Std. Err.	t	$P > \|t\|$	[95% Conf.	Interval]
age	.0211974	.0040859	5.19	0.000	.0131889	.0292059
year	-.0170695	.0052466	-3.25	0.001	-.0273531	-.006786
cohort_1914_c10	-.3378195	.1558264	-2.17	0.030	-.6432459	-.0323931
cohort_1924_c10	-.2293381	.1166138	-1.97	0.049	-.4579062	-.0007699
cohort_1934_c10	-.1830169	.0791145	-2.31	0.021	-.3380848	-.0279489
cohort_1944_c10	-.0505025	.0492687	-1.03	0.305	-.1470713	.0460663
_cons	40.03708	10.36835	3.86	0.000	19.71467	60.3595

B Code

```
clear
est clear
set more off
set mem 10m

cd "C:\Users\mikew\Desktop\University\TilburgUniversity\Master\panel_data_analysis\
    linear_models\assignment_1"
global datapath "data"
global output "output"

// Load data
use "$datapath\soep.dta", clear

// Declare this data to be panel data
xtset persnr year

// Exercise 1a
gen cohort = year - age
label variable cohort "cohort (birth year)"

// Grouping 5 cohorts together
forvalues i = 1914(5)1963 {
        by age, sort: egen mean_s_life_`i'_c5 = ///
                mean(cond(inrange(cohort, `i', `i'+4), s_life, .))
}

graph twoway line mean_s_life_1914_c5 age|| line mean_s_life_1919_c5 age ///
        || line mean_s_life_1924_c5 age || line mean_s_life_1929_c5 age ///
        || line mean_s_life_1934_c5 age || line mean_s_life_1939_c5 age ///
        || line mean_s_life_1944_c5 age || line mean_s_life_1949_c5 age ///
        || line mean_s_life_1954_c5 age || line mean_s_life_1959_c5 age, ///
        legend(order(1 "1914-1918" 2 "1919-1923" ///
        3 "1924-1928" 4 "1929-1933" 5 "1934-1938" ///
        6 "1939-1943" 7 "1944-1948" 8 "1949-1953" ///
        9 "1954-1958"  10 "1959-1963") subtitle("Cohort (birth year)")) ///
        title("Life satisfaction versus age per cohort") ///
        ytitle("Mean life satisfaction") graphregion(color(white))

gr export "$output/spaghetti_plot_5_cohorts.eps", as(eps) preview(off) replace

// Grouping 10 cohorts together
forvalues i = 1914(10)1963 {
        by age, sort: egen mean_s_life_`i'_c10 = ///
                mean(cond(inrange(cohort, `i', `i'+9), s_life, .))
}

graph twoway line mean_s_life_1914_c10 age|| line mean_s_life_1924_c10 age ///
        || line mean_s_life_1934_c10 age || line mean_s_life_1944_c10 age ///
        || line mean_s_life_1954_c10 age, ///
        legend(order(1 "1914-1923"  2 "1924-1933" 3 "1934-1943" ///
        4 "1944-1953" 5 "1954-1963") subtitle("Cohort")) ///
        title("Life satisfaction versus age per cohort") ///
        ytitle("Mean life satisfaction") graphregion(color(white))

gr export "$output/spaghetti_plot_10_cohorts.eps", as(eps) preview(off) replace

// Exercise 1c
// Create dummies for cohorts
forvalues i = 1914(5)1963 {
        gen cohort_`i'_c5 = cohort>=`i' & cohort<=`i'+4
}

forvalues i = 1914(10)1963 {
```

```
        gen cohort_`i'_c10 = cohort>=`i' & cohort<=`i'+9
}

// Create macro
global cohort_5_dummies "cohort_1914_c5 cohort_1919_c5 cohort_1924_c5 cohort_1929_c5
    cohort_1934_c5 cohort_1939_c5 cohort_1944_c5 cohort_1949_c5 cohort_1954_c5 cohort_1959_c5"
global cohort_10_dummies "cohort_1914_c10 cohort_1924_c10 cohort_1934_c10 cohort_1944_c10
    cohort_1954_c10"

// Run the regression
reg s_life age year $cohort_5_dummies $cohort_10_dummies // all dummies
reg s_life age year $cohort_5_dummies
reg s_life age year $cohort_10_dummies

// Exercise 2
clear
set seed 345398
drawnorm alpha_i, n(200)
expand 5

drawnorm nu_it e_it, n(1000)
gen x_it = nu_it + alpha_i
drop nu_it
gen y_it = 3 + alpha_i + 2*x_it + e_it

// 2b
pwcorr, sig

// Exercise 3
clear
set seed 345398
capture program drop mcprog
program mcprog
        clear
        drawnorm alpha_i, n(200)
        expand 5

        drawnorm nu_it e_it, n(1000)
        gen x_it = nu_it + alpha_i
        drop nu_it
        gen y_it = 3 + alpha_i + 2*x_it + e_it

        regress y_it x_it
end
simulate _b _se, reps(100): mcprog
sum

// a
clear
set seed 345398
capture program drop mcprog_cluster
program mcprog_cluster
        clear
        drawnorm alpha_i, n(200)
        egen persnr = seq(), f(1) t(200)
        expand 5

        drawnorm nu_it e_it, n(1000)
        gen x_it = nu_it + alpha_i
        drop nu_it
        gen y_it = 3 + alpha_i + 2*x_it + e_it

        regress y_it x_it, cluster(persnr)
end
simulate _b _se, reps(100): mcprog_cluster
```

```
sum

// Exercise 4
// Fixed Effects
clear
set seed 345398
capture program drop mc_fixed_effects
program mc_fixed_effects
        clear
        drawnorm alpha_i, n(200)
        egen persnr = seq(), f(1) t(200)
        expand 5
        bysort persnr: gen t = _n-1

        drawnorm nu_it e_it, n(1000)
        gen x_it = nu_it + alpha_i
        drop nu_it
        gen y_it = 3 + alpha_i + 2*x_it + e_it

        xtset persnr t

        xtreg y_it x_it, i(persnr) fe
end
simulate _b _se, reps(100): mc_fixed_effects
sum

// First Difference
clear
set seed 345398
capture program drop mc_first_difference
program mc_first_difference
        clear
        drawnorm alpha_i, n(200)
        egen persnr = seq(), f(1) t(200)
        expand 5
        bysort persnr: gen t = _n-1

        drawnorm nu_it e_it, n(1000)
        gen x_it = nu_it + alpha_i
        drop nu_it
        gen y_it = 3 + alpha_i + 2*x_it + e_it

        xtset persnr t
        xtreg d.y_it d.x_it
end
simulate _b _se, reps(100): mc_first_difference
sum

// Exercise 5
// For t=5
clear
set seed 345398
capture program drop mc_lag_5
program mc_lag_5
        clear
        drawnorm alpha_i, n(200)
        egen persnr = seq(), f(1) t(200)
        expand 5
        bysort persnr: gen t = _n-1

        drawnorm nu_it e_it, n(1000)
        gen x_it = nu_it + alpha_i
        drop nu_it
        gen y_it = 3 + alpha_i + 2*x_it + e_it
        replace y_it = 3 + alpha_i + 2*x_it + e_it + 0.5*y_it[_n-1] if t!=0
```

```
        gen y_it_lag_1 = y_it[_n-1]

        xtset persnr t
        xtreg y_it x_it y_it_lag_1, fe
end

simulate _b _se, reps(100): mc_lag_5
gen bias_lag = _b_y_it_lag_1 - 0.5
sum bias_lag

// For t=10
clear
set seed 345398
capture program drop mc_lag_10
program mc_lag_10
        clear
        drawnorm alpha_i, n(200)
        egen persnr = seq(), f(1) t(200)
        expand 10
        bysort persnr: gen t = _n-1

        drawnorm nu_it e_it, n(2000)
        gen x_it = nu_it + alpha_i
        drop nu_it
        gen y_it = 3 + alpha_i + 2*x_it + e_it
        replace y_it = 3 + alpha_i + 2*x_it + e_it + 0.5*y_it[_n-1] if t!=0
        gen y_it_lag_1 = y_it[_n-1]

        xtset persnr t
        xtreg y_it x_it y_it_lag_1, fe
end

simulate _b _se, reps(100): mc_lag_10
gen bias_lag = _b_y_it_lag_1 - 0.5
sum bias_lag

// For t=20
clear
set seed 345398
capture program drop mc_lag_20
program mc_lag_20
        clear
        drawnorm alpha_i, n(200)
        egen persnr = seq(), f(1) t(200)
        expand 20
        bysort persnr: gen t = _n-1

        drawnorm nu_it e_it, n(4000)
        gen x_it = nu_it + alpha_i
        drop nu_it
        gen y_it = 3 + alpha_i + 2*x_it + e_it
        replace y_it = 3 + alpha_i + 2*x_it + e_it + 0.5*y_it[_n-1] if t!=0
        gen y_it_lag_1 = y_it[_n-1]

        xtset persnr t
        xtreg y_it x_it y_it_lag_1, fe
end

simulate _b _se, reps(100): mc_lag_20
gen bias_lag = _b_y_it_lag_1 - 0.5
sum bias_lag

// For t=50
clear
set seed 345398
```

```
capture program drop mc_lag_50
program mc_lag_50
        clear
        drawnorm alpha_i, n(200)
        egen persnr = seq(), f(1) t(200)
        expand 50
        bysort persnr: gen t = _n-1

        drawnorm nu_it e_it, n(10000)
        gen x_it = nu_it + alpha_i
        drop nu_it
        gen y_it = 3 + alpha_i + 2*x_it + e_it
        replace y_it = 3 + alpha_i + 2*x_it + e_it + 0.5*y_it[_n-1] if t!=0
        gen y_it_lag_1 = y_it[_n-1]

        xtset persnr t
        xtreg y_it x_it y_it_lag_1, fe
end

simulate _b _se, reps(100): mc_lag_50
gen bias_lag = _b_y_it_lag_1 - 0.5
sum bias_lag

// Exercise 6
clear
set seed 345398
capture program drop mc_arellano_bond
program mc_arellano_bond
        clear
        drawnorm alpha_i, n(200)
        egen persnr = seq(), f(1) t(200)
        expand 5
        bysort persnr: gen t = _n-1

        drawnorm nu_it e_it, n(1000)
        gen x_it = nu_it + alpha_i
        drop nu_it
        gen y_it = 3 + alpha_i + 2*x_it + e_it
        replace y_it = 3 + alpha_i + 2*x_it + e_it + 0.5*y_it[_n-1] if t!=0
        gen y_it_lag_1 = y_it[_n-1]

        xtset persnr t
        xtabond y_it x_it y_it_lag_1
end

simulate _b _se, reps(100): mc_arellano_bond
sum
```