Replication Noteboook

J. Eduardo Vera-Valdés
Aalborg University
eduardo@math.aau.dk

Table of contents

> Note
>
> The following is a replication notebook of the paper by M. Bennedsen, E. Hillebrand, and S. J. Koopman [1]. The replication is done in Julia using Quarto as part of the paper *"On measurements errors in CO_2 airborne fraction estimates"* by *J. Eduardo Vera-Valdés* and *Charisios Grivas*. This notebook contains no new results.

Notebook setup

This notebook is written in Julia and uses the following packages:

- `DataFrames` for data manipulation
- `XLSX` for reading data from an Excel file
- `Plots`
- `LinearAlgebra`
- `Statsistics`
- `HypothesisTests`

All packages are available in the Julia registry and can be installed using the Julia package manager with the following command:

```julia
using Pkg
Pkg.add("DataFrames", "XLSX", "Plots", "LinearAlgebra", "Statistics",
"HypothesisTests")
```

In the following, we load a project environment that contains the necessary packages. This step is not required if the packages are already installed in the current environment.

Airborne fraction

The airborne fraction is the fraction of CO_2 emissions that remain in the atmosphere. It is a key parameter in the carbon cycle and is used to estimate the impact of human activities on the climate system. The airborne fraction is defined as the ratio of the increase in atmospheric CO_2 concentration to the total CO_2 emissions. It is usually expressed as a percentage.

Data

We load the data from an Excel file and plot the CO_2 emissions and the atmospheric CO_2 concentration over time.

The data is neatly collected in an Excel file in the author's GitHub repository at the following link.

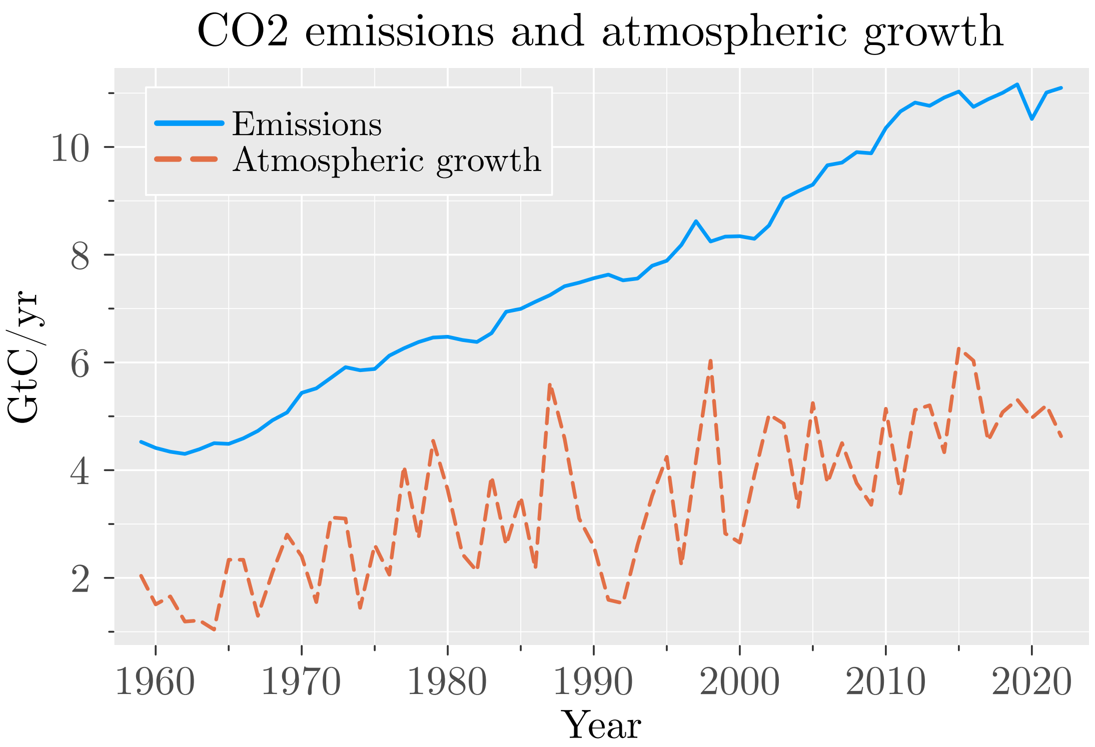
To ease things up, we have downloaded the data directly from the repository and saved it in the file `AF_data.xlsx` in the local folder.

We can read the data using the `XLSX.jl` package, convert it to a data frame using `DataFrames.jl`. We then recover the year, emissions, and coverage variables. Note that emissions are defined as the sum of fossil fuels (FF) and land-use and land-coverage changes (LULCC).

```julia
using DataFrames, XLSX, LinearAlgebra

path = "AF_data.xlsx"

data = DataFrame(XLSX.readtable(path, "Data"))

year = data[!, 1];
fossilfuels = Vector{Float64}(data[!, 4]);
lulcc = Vector{Float64}(data[!, 6]);
emissions = fossilfuels .+ lulcc;
coverage = Vector{Float64}(data[!, 5]);

VAI = Vector{Float64}(data[!,9]);
ENSO = Vector{Float64}(data[!, 10]);
E = emissions;
G = coverage;
```

Note that atmospheric concentration growth, denoted `G` from hereinafter, is transformed into a vector of `Float64` at definition to avoid type issues. Emissions, the sum of fossil fuels (`fossilfuels`) and land-use and land-coverage changes (`lulcc`), are denoted by `E`.

Once loaded, we can plot the data using the `Plots.jl` package.



Classic estimation

Commonly, the airborne fraction is estimated using the following formula:

$$\frac{G_t}{E_t} = \alpha + \epsilon_t$$

where G_t is the atmospheric CO_2 concentration at time t, E_t is the total CO_2 emissions at time t, and ϵ_t is the error term that captures the natural variability in the carbon cycle. The parameter α is the airborne fraction.

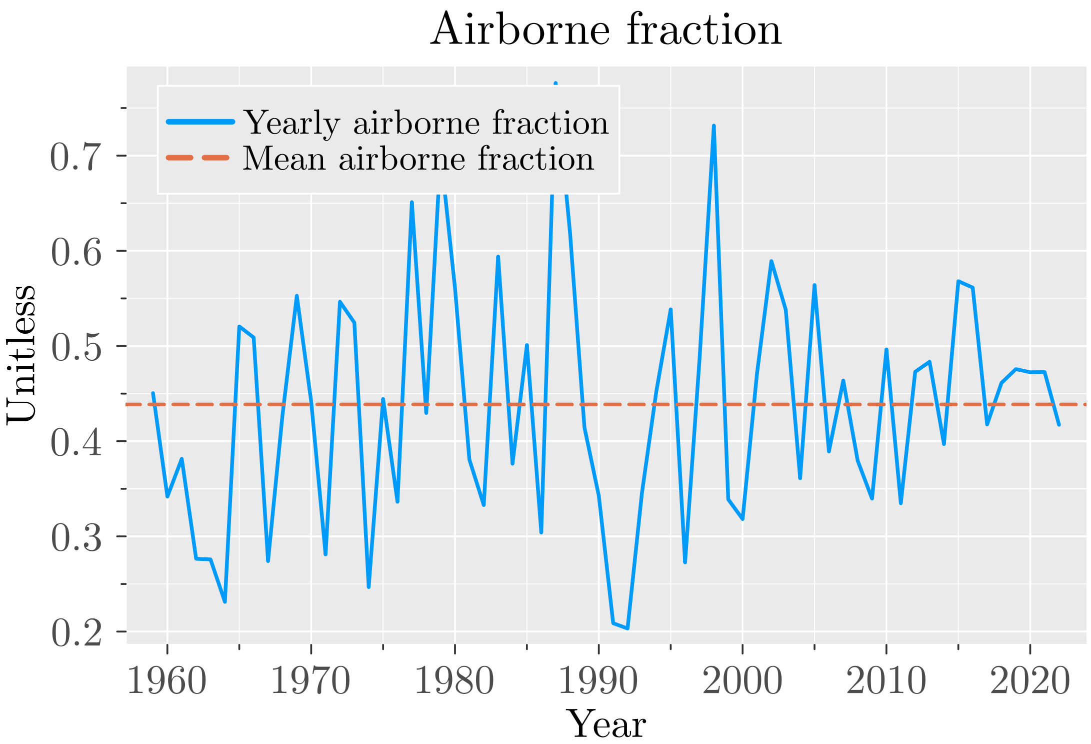
In practice, we estimate α by taking the mean of the ratio of the coverage to the emissions.

```
using Statistics

α₁ = mean(G ./ E)
```

```
0.43856861803874964
```

This value is the estimated airborne fraction, which is the consensus value in the literature.

We plot the yearly airborne fraction and the estimated mean airborne fraction.



A new approach

Recently, M. Bennedsen, E. Hillebrand, and S. J. Koopman [1] has suggested a new approach to estimate the airborne fraction. They propose to use the following formula:

$$G_t = \alpha E_t + \epsilon_t,$$

and estimate α, the airborne fraction, using ordinary least squares (OLS). They argue that this approach provides better statistical properties. Among them, the OLS estimator is super-consistent, meaning that it converges to the true value at a faster rate than the standard OLS estimator. They also show that the estimator has lower variance and is asymptotically normal.

The new approach relies on estimating the **cointegration relationship** between the emissions and the coverage using OLS. As for all cointegration analyses, as a first step, we need to check if the variables are integrated of the same order. We can do this by testing for the presence of a unit root in the series.

Unit root test

We use the Augmented Dickey-Fuller (ADF) [2] test to test for the presence of a unit root in the series. The null hypothesis is that the series has a unit root, while the alternative hypothesis is that the series is stationary.

In Julia, we can use the `ADFTest` function from the `HypothesisTests.jl` package to perform the test.

As a demonstration, we test the emissions series for the presence of a unit root in a model with a trend and two lags.

```julia
using HypothesisTests
```

```
τᵉₜ = ADFTest(E, :trend, 2)
```

```
Augmented Dickey-Fuller unit root test
--------------------------------------
Population details:
    parameter of interest:   coefficient on lagged non-differenced variable
    value under h_0:         0
    point estimate:          -0.262114

Test summary:
    outcome with 95% confidence: fail to reject h_0
    p-value:                     0.2907

Details:
    sample size in regression:          61
    number of lags:                     2
    ADF statistic:                      -2.57687
    Critical values at 1%, 5%, and 10%: [-4.10768 -3.48147 -3.16849]
```

In this case, the null hypothesis of a unit root in the emissions series is not rejected.

We can perform the same test for the coverage series while considering different combinations of models and lags.

```
# Dataframe to store the results
resultsdf = DataFrame("Variable" => String[], "Model" => String[], "L = 0" =>
Float64[], "L = 1" => Float64[], "L = 2" => Float64[], "L = 3" => Float64[],
"L = 4" => Float64[], "L = 5" => Float64[])

for variable in [:E, :G]
    for model in [:none, :constant, :trend]
        fila = zeros(6)
        for lags in 0:5
            τ = ADFTest(eval(variable), model, lags)
            fila[lags+1] = pvalue(τ)
        end
                           push!(resultsdf,   [titlecase(string(variable)),
titlecase(string(model)), fila...])
    end
end

resultsdf
```

Based on these results, we cannot reject the null hypothesis of a unit root in the `emissions` series for all models and lags. For the `coverage` series, the tests reject except for the model with a trend. This suggests that both series seem stationary.

Cointegration test

We can test for cointegration between the emissions and the coverage using the R. F. Engle and C. W. J. Granger [3] test. The null hypothesis is that there is no cointegration relationship between the series, while the alternative hypothesis is that there is a cointegration relationship.

To test for cointegration, we first estimate the OLS regression of the coverage on the emissions. We then test the residuals for a unit root using the ADF test. Note that the residuals should be stationary if there is a cointegration relationship between the series.

```
α₂ = (E'E) \ (E'G)
```

```
0.4477918844144535
```

The estimated airborne fraction is slightly larger than the classical estimate.

To test if there is a cointegration relationship, and thus that we have a valid estimate, we must recover the residuals and test them for a unit root.

```
res₂ = G - α₂ * E

τᵣ = ADFTest(res₂, :none, 0)
```

```
Augmented Dickey-Fuller unit root test
--------------------------------------
Population details:
    parameter of interest:   coefficient on lagged non-differenced variable
    value under h_0:         0
    point estimate:          -0.963507

Test summary:
    outcome with 95% confidence: reject h_0
    p-value:                     <1e-11

Details:
    sample size in regression:          63
    number of lags:                     0
    ADF statistic:                      -7.58332
    Critical values at 1%, 5%, and 10%: [-2.60156 -1.9459 -1.61324]
```

The test statistic has to be compared against the critical values generated by J. G. MacKinnon [4]. We can reject the null hypothesis of a unit root in the residuals, which suggests that there is a cointegration relationship between the emissions and the coverage and the estimate is valid.

Standard errors

We compute the standard errors of the estimates of the airborne fraction using the formula:

$$SD(\hat{\alpha}) = \sqrt{\hat{\sigma}_{\epsilon}^{2} (E'E)^{-1}},$$

where $\hat{\sigma}_{\epsilon}^{2} = \sum \hat{\epsilon}^{2} / N$ is the estimate of the variance of the error term and $\hat{\epsilon}_{t}$ are the residuals from the regression.

```
rss₂ = sum((G - α₂ * E).^2)
σ²₂ = rss₂ / (length(G) - 1)

sd₍α₂₎ = sqrt( σ²₂ / (E'E) )
```

```
0.014241317441433234
```

Additional covariates

We consider adding additional covariates to the model. In particular, we consider adding $ENSO$ (El Niño Southern Oscillation) and VAI (volcanic activity index) as covariates. These variables are known to affect the carbon cycle and can potentially influence the airborne fraction.

We estimate the following model:

$$G_t = \alpha E_t + \gamma_1 ENSO_t + \gamma_2 VAI_t + \epsilon_t,$$

where $ENSO_t$ and VAI_t are the El Niño Southern Oscillation and volcanic activity index at time t, respectively.

Note that the authors first detrend the ENSO series before estimating the model. We can do this by regressing the series on a time trend and taking the residuals.

```
T = length(ENSO)
Xₜ = [ones(T) collect(1:T)]
ρ = (Xₜ'Xₜ) \ (Xₜ'ENSO)
ENSOᵨ = ENSO - Xₜ * ρ;
```

We estimate the extended model using OLS.

```
Xₑ = [E ENSOᵨ VAI]

αₑ = (Xₑ'Xₑ) \ (Xₑ'G)
```

```
3-element Vector{Float64}:
   0.4696645797106208
   1.0024516231210219
 -15.1482109617327
```

Standard errors are computed with the multivariate version of the variance formula:

$$Var(\hat{\alpha}) = \hat{\sigma}_{\epsilon}^2 (X'X)^{-1},$$

where X is the matrix of all regressors.

```
rssₑ = sum((G - Xₑ*αₑ).^2)
σ²ₑ = rssₑ / (length(G) - 3)
var₍αₑ₎ = σ²ₑ *  inv(Xₑ'Xₑ)
sqrt.(var₍αₑ₎[1,1])
```

```
0.01058726080675776
```

Recent subsample

Given the variability of the LULCC measurements at the beginning of the series, we consider a recent subsample of the data. We consider the data from 1992 and estimate the airborne fraction using the new approach.

Getting subsample data.

```
E92 = E[year .>= 1992];
G92 = G[year .>= 1992];
VAI92 = VAI[year .>= 1992];
ENSOₚ92 = ENSOₚ[year .>= 1992];
```

Estimating in the subsample.

Simple specification.

```
α92₂ = (E92'E92) \ (E92'G92)
rss92₂ = sum((G92 - α92₂ * E92).^2)
σ92²₂ = rss92₂ / (length(G92) - 1)
sd₍α92₂₎ = sqrt( σ92²₂ / (E92'E92) )

[α92₂ sd₍α92₂₎]
```

```
1×2 Matrix{Float64}:
 0.44967  0.017309
```

```
X92ₑ = [E92 ENSOₚ92 VAI92]
α92ₑ = (X92ₑ'X92ₑ) \ (X92ₑ'G92)

rss92ₑ = sum((G92 - X92ₑ*α92ₑ).^2)
σ92²ₑ = rss92ₑ / (length(G92) - 3)
var₍α92ₑ₎ = σ92²ₑ * inv(X92ₑ'X92ₑ)

[α92ₑ var₍α92ₑ₎]
```

```
3×4 Matrix{Float64}:
   0.461253   0.000126359   9.12279e-5  -0.0092892
   1.02214    9.12279e-5    0.0296269   -0.157186
 -17.5878    -0.0092892    -0.157186    13.5323
```

Other datasets

We can also test the new approach on other datasets. We can use the same methodology to estimate the airborne fraction. The preferred data for the LULCC emissions are from the Global Carbon Project [5]. However, we can also use data from R. A. Houghton and A. Castanho [6] and M. J. van Marle, D. van Wees, R. A. Houghton, R. D. Field, J. Verbesselt, and G. R. van der Werf [7].

```
lulcc₂ = Vector{Float64}(data[!, 7]);
lulcc₃ = Vector{Float64}(data[!, 8]);

E₂ = fossilfuels .+ lulcc₂;
E₃ = fossilfuels .+ lulcc₃;

α₆ = (E₂'E₂) \ (E₂'G)
rss₆ = sum((G - α₆ * E₂).^2)
σ²₆ = rss₆ / (length(G) - 1)
sd₍α₆₎ = sqrt( σ²₆ / (E₂'E₂) )

α₇ = (E₃'E₃) \ (E₃'G)
rss₇ = sum((G - α₇ * E₃).^2)
σ²₇ = rss₇ / (length(G) - 1)
sd₍α₇₎ = sqrt( σ²₇ / (E₃'E₃) )

[α₆ sd₍α₆₎; α₇ sd₍α₇₎]
```

```
2×2 Matrix{Float64}:
 0.475539  0.0152315
 0.490211  0.0157142
```

Deming regression estimator

We can also estimate the airborne fraction using Deming regression [8]. Deming regression is a method for estimating the parameters of a linear regression model when both the dependent and independent variables are subject to measurement error. The Deming regression assumes that the measurement errors in the dependent and independent variables are normally distributed with known variances.

The Deming regression estimator is given by:

$$\hat{\alpha}_{Deming} = \frac{M_{GG} - \delta M_{EE} + \sqrt{\left(M_{GG} - \delta M_{EE}\right)^2 + 4\delta M_{EG}^2}}{2M_{EG}},$$

where

$$M_{GG} = \frac{1}{T}\sum_{t=1}^{T} G_t^2,$$

$$M_{EE} = \frac{1}{T}\sum_{t=1}^{T} E_t^2,$$

$$M_{EG} = \frac{1}{T}\sum_{t=1}^{T} E_t G_t,$$

and $\delta = \frac{Variance(\omega_{G,t})}{Variance(\eta_{E,t})}$ is the ratio of the variance of the measurement error in the coverage to the variance of the measurement error in emissions.

Several values for δ are tried, given that the true value is unknown.

```
M₍ₑₑ₎ = E'E
M₍ₑₐ₎ = E'G
M₍ₐₐ₎ = G'G

δ = zeros(2, 5)
δ[ 1, :] = [0.2 0.5 1 2 5]

for ii = 1:5
    δ[ 2, ii] = ( M₍ₐₐ₎ - δ[1, ii] * M₍ₑₑ₎ + sqrt( (M₍ₐₐ₎ - δ[1, ii] * M₍ₑₑ₎)^2 + 4 * δ[1, ii] * M₍ₑₐ₎^2 ) ) / (2 * M₍ₑₐ₎)
end

display(δ)
```

```
2×5 Matrix{Float64}:
 0.2       0.5       1.0     2.0       5.0
 0.462305  0.456067  0.4526  0.450406  0.448895
```

Other drawbacks of the Deming regression is that there is no closed form expression to obtain the standard errors. It also cannot easily handle additional regressors like in the preferred specification. We solve these issues in the paper: *On measurements errors in CO$_2$ airborne fraction estimates* by *J. Eduardo Vera-Valdés* and *Charisios Grivas*.

Summary of results

```
results_replication = DataFrame("Model" => String[], "Estimate" => Float64[], "Std. Error" => Float64[], "Confidence Int." => Vector{Float64}[] )
nd = 4;

push!(results_replication, ["Regression", α₂, sd₍α₂₎, round.([α₂ - 1.96 * sd₍α₂₎, α₂ + 1.96 * sd₍α₂₎], digits=nd) ] )
push!(results_replication, ["Regression with ENSO and VAI", αₑ[1], sqrt(var₍αₑ₎[1,1]), round.([αₑ[1] - 1.96 * sqrt(var₍αₑ₎[1,1]), αₑ[1] + 1.96 * sqrt(var₍αₑ₎[1,1])], digits=nd) ] )
push!(results_replication, ["Regression from 1992", α92₂, sd₍α92₂₎, round.([α92₂ - 1.96 * sd₍α92₂₎, α92₂ + 1.96 * sd₍α92₂₎], digits=nd) ] )
push!(results_replication, ["Regression from 1992 with ENSO and VAI", α92ₑ[1], sqrt(var₍α92ₑ₎[1,1]), round.([α92ₑ[1] - 1.96 * sqrt(var₍α92ₑ₎[1,1]), α92ₑ[1] + 1.96 * sqrt(var₍α92ₑ₎[1,1])], digits=nd) ] )
push!(results_replication, ["Regression with LULCC (H&N)", α₆, sd₍α₆₎, round.([α₆ - 1.96 * sd₍α₆₎, α₆ + 1.96 * sd₍α₆₎], digits=nd) ] )
push!(results_replication, ["Regression with LULCC (vMa)", α₇, sd₍α₇₎, round.([α₇ - 1.96 * sd₍α₇₎, α₇ + 1.96 * sd₍α₇₎], digits=nd) ] )
push!(results_replication, ["Deming regression (δ=0.2)", δ[2, 1], NaN, [NaN] ] )
push!(results_replication, ["Deming regression (δ=0.5)", δ[2, 2], NaN, [NaN] ] )
push!(results_replication, ["Deming regression (δ=1)", δ[2, 3], NaN, [NaN] ] )
push!(results_replication, ["Deming regression (δ=2)", δ[2, 4], NaN, [NaN] ] )
push!(results_replication, ["Deming regression (δ=5)", δ[2, 5], NaN, [NaN] ] )
results_replication.Estimate = round.(results_replication.Estimate,
digits=nd)
results_replication."Std. Error" = round.(results_replication."Std. Error",
digits=nd)
display(results_replication)
```

Note that M. Bennedsen, E. Hillebrand, and S. J. Koopman [1] considered heteroscedasticity and autocorrelation robust standard errors. Nonetheless, their selected bandwidth is quite small, so that they are almost identical to the OLS standard errors. We report here the latter for simplicity.

References

Bibliography

[1] M. Bennedsen, E. Hillebrand, and S. J. Koopman, “A regression-based approach to the CO2 airborne fraction,” *Nature Communications*, vol. 15, no. 1, p. 8507–8508, Oct. 2024, doi: 10.1038/s41467-024-52728-1.

[2] D. A. Dickey and W. A. Fuller, “Distribution of the estimators for autoregressive time series with a unit root,” *Journal of the American statistical association*, vol. 74, no. 366a, pp. 427–431, 1979.

[3] R. F. Engle and C. W. J. Granger, “Co-Integration and Error Correction: Representation, Estimation, and Testing,” *Econometrica*, vol. 55, no. 2, pp. 251–276, Mar. 1987, [Online]. Available: https://www.jstor.org/stable/1913236

[4] J. G. MacKinnon, “Critical values for cointegration tests,” 2010.

[5] P. Friedlingstein *et al.*, “Global carbon budget 2023,” *Earth System Science Data*, vol. 15, no. 12, pp. 5301–5369, 2023.

[6] R. A. Houghton and A. Castanho, “Annual emissions of carbon from land use, land-use change, and forestry 1850–2020,” *Earth System Science Data Discussions*, vol. 2022, pp. 1–36, 2022.

[7] M. J. van Marle, D. van Wees, R. A. Houghton, R. D. Field, J. Verbesselt, and G. R. van der Werf, “RETRACTED ARTICLE: New land-use-change emissions indicate a declining CO2 airborne fraction,” *Nature*, vol. 603, no. 7901, pp. 450–454, 2022.

[8] W. E. Deming, *Statistical adjustment of data.* wiley, 1943.