

# BLOOMIMPORT

Stata package to import Bloomberg exported data

---

*Ultimo aggiornamento: 20 settembre 2024*

Nicola Tommasi  
C.I.D.E.  
[nicola.tommasi@univr.it](mailto:nicola.tommasi@univr.it)

# 1. Il comando bloomimport

bloomimport è un nuovo comando Stata che permette

- di importare in Stata i dati esportati tramite una particolare procedura che usa le API Bloomberg di Excel
- di trasformarli in serie storica
- di convertire in numeriche le variabili create

I dati devono essere stati esportati tramite la procedura "Creare spreadsheet" -> "Tabella dati storici" e devono rispettare la disposizione-layout mostrata in figura 1 e d'ora in avanti definita "formato wide"

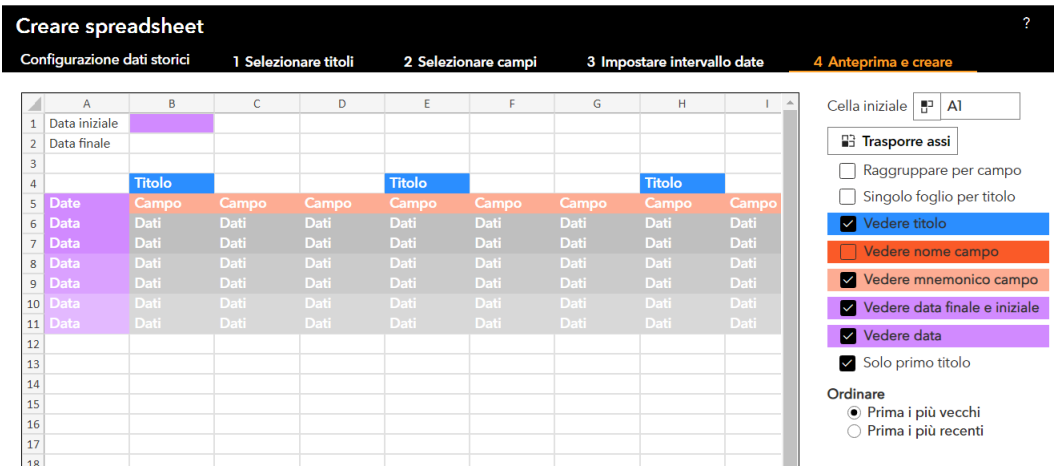


Figura 1 – Layout wide dei dati

oppure rispettare la disposizione-layout mostrata in figura 2, e d'ora in avanti definita "formato long", che si ottiene cliccando sul 'Transpose Axes', in alto a sinistra<sup>1</sup>.

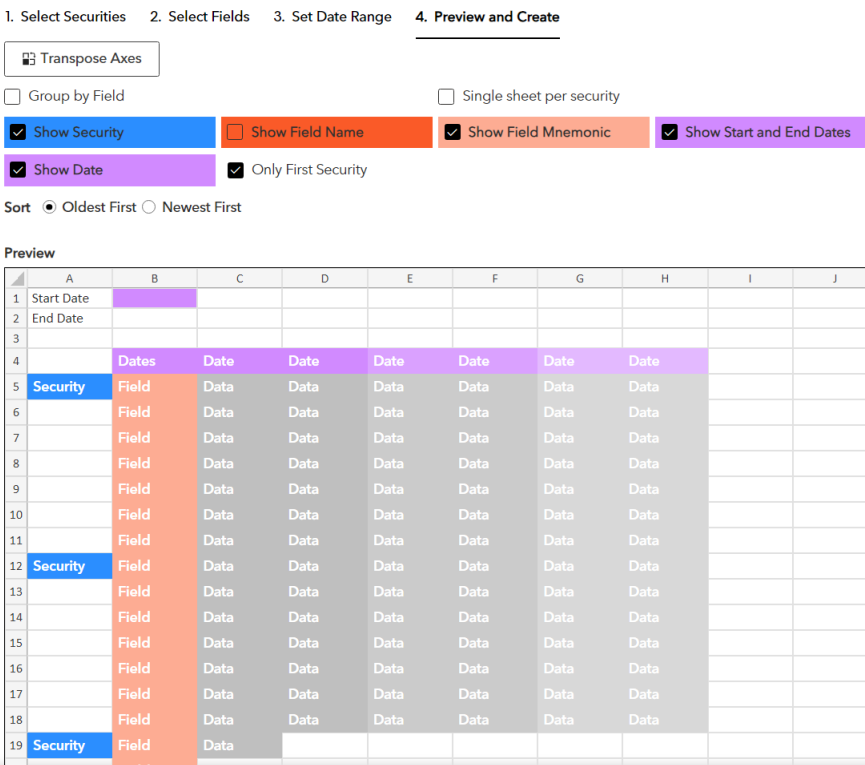


Figura 2 – Layout long dei dati

1. La figura figura 1 fa riferimento alla precedente versione di esportazione di Bloomberg, la figura 2 a quella più recente.

Notare che per entrambi i layout non deve essere selezionato "Vedere nome campo" (Show Field Name). Se i dati esportati presentano questa riga bisogna selezionarla, eliminarla e ris salvare il file. Sia per il formato long che per il formato wide è possibile avere una sola colonna con la data valida per tutti i tickers oppure una colonna data per ogni singolo ticker. Questa possibilità dipende dalla spunta vicino alla voce 'Solo primo titolo' (Only First Security) accanto alla spunta 'Vedere data' (Show Date), come si può vedere sia in Figura 1 che in Figura 2.

Quindi le possibili combinazioni tra layout e date compatibili con il comando sono:

- disposizione-layout wide con data unica per tutti i ticker (Figura 3)
- disposizione-layout wide con data singola per ciascun ticker (Figura 4)
- disposizione-layout long con data unica per tutti i ticker (Figura 5)
- disposizione-layout long con data singola per ciascun ticker (Figura 6)

Nelle figure seguenti viene mostrato un esempio per ciascuna delle 4 possibili combinazioni.

Si noti che in Figura 3 il secondo ticker inizia subito dopo il primo (colonna E), mentre in Figura 4 una colonna vuota separa i diversi ticker e che i dati di ciascun ticker iniziano con una colonna data (colonne A e G).

	A	B	C	D	E	F	G
1	Data iniziale	31/12/2013					
2	Data finale	31/12/2023					
3							
4		BACB BU Equity			BMED IM Equity		
5	Dates	CLIMATE_CHG_POLICY	CLIMATE_CHG_PRODS	CLIMATE_CHG_OPPORTUNITIES	CLIMATE_CHG_POLICY	CLIMATE_CHG_PRODS	CLIMATE_CHG_OPPORTUNITIES
6	31/12/2013	#N/A N/A	#N/A N/A	#N/A N/A	0	0	0
7	31/12/2014	#N/A N/A	#N/A N/A	#N/A N/A	0	0	0
8	31/12/2015	#N/A N/A	#N/A N/A	#N/A N/A	1	0	0
9	31/12/2016	#N/A N/A	#N/A N/A	#N/A N/A	1	0	0
10	31/12/2017	#N/A N/A	#N/A N/A	#N/A N/A	1	0	0
11	31/12/2018	#N/A N/A	#N/A N/A	#N/A N/A	1	0	0
12	31/12/2019	#N/A N/A	#N/A N/A	#N/A N/A	1	0	0
13	31/12/2020	0	0	0	1	0	0
14	31/12/2021	0	0	0	1	0	0
15	31/12/2022	1	0	0	1	0	0
16	31/12/2023	1	0	0	1	0	0

Figura 3 — Esempio di dati esportati in formato wide e con data unica per tutti i ticker

	A	B	C	D	E	F	G	H	I	J	K
1	Data iniziale	01/01/2016									
2	Data finale	30/06/2024									
3											
4	AAPL UW Equity						A UN Equity				
5	Dates	ENTERPRISE_VALUE	PX_TO_BOOK_RATIO	TOT_COMMON_EQY	BS_TOT_ASSET		Dates	ENTERPRISE_VALUE	PX_TO_BOOK_RATIO	TOT_COMMON_EQY	BS_TOT_ASSET
6	29/01/2016	#N/A N/A	4,2076	#N/A N/A	#N/A N/A		29/01/2016	#N/A N/A	3,053	#N/A N/A	#N/A N/A
7	29/02/2016	#N/A N/A	4,1795	#N/A N/A	#N/A N/A		31/01/2016	12154,2	3,053	4045	7302
8	26/03/2016	425851,3888	4,1795	130457	305277		29/02/2016	12154,2	3,0286	4045	7302
9	31/03/2016	425851,3888	4,577	130457	305277		31/03/2016	12154,2	3,2313	4045	7302
10	29/04/2016	425851,3888	3,9365	130457	305277		29/04/2016	12154,2	3,1953	4045	7302
11	31/05/2016	425851,3888	4,1935	130457	305277		30/04/2016	13052	3,1953	4162	7640
12	25/06/2016	357136,611	4,1935	126541	305602		31/05/2016	13052	3,5834	4162	7640
13	30/06/2016	357136,611	4,0745	126541	305602		30/06/2016	13052	3,464	4162	7640
14	29/07/2016	357136,611	4,4414	126541	305602		29/07/2016	13052	3,5891	4162	7640
15	31/08/2016	357136,611	4,522	126541	305602		31/07/2016	15278,64	3,5891	4343	7734
16	24/09/2016	450886,2699	4,522	128249	321686		31/08/2016	15278,64	3,5048	4343	7734
17	30/09/2016	450886,2699	4,7038	128249	321686		30/09/2016	15278,64	3,513	4343	7734
18	31/10/2016	450886,2699	4,7242	128249	321686		31/10/2016	13734,68	3,3271	4243	7794
19	30/11/2016	450886,2699	4,5985	128249	321686		30/11/2016	13734,68	3,3584	4243	7794
20	30/12/2016	450886,2699	4,5977	128249	321686		30/12/2016	13734,68	3,479	4243	7794
21	31/12/2016	450147,0919	4,5977	132390	331141		31/01/2017	15522,34	3,6696	4297	7872

Figura 4 — Esempio di dati esportati in formato wide e con data singola per ciascun ticker

Come si vede in Figura 5 il secondo ticker inizia subito dopo il primo (riga 11), mentre in Figura 6 una riga vuota separa i diversi ticker e i dati di ciascun ticker iniziano con una riga per data (righe 4 e 17).

	A	B	C	D	E	F
1	Start Date	CY 2005				
2	End Date	CY 2023				
3						
4		Dates	30/12/2005	29/12/2006	31/12/2007	31/12/2008
5	FBMS US Equity	BS_NON_PERFORM_ASSET	0,5937	2,3461	4,188	4,9694
6		TANGIBLE_COMMON_EQUITY	18,4776	30,9064	34,9724	35,3291
7		BS_RSRV_LOAN_LOSS	2,3668	3,7929	4,2212	4,7849
8		EPS_GROWTH	51,4019	66,6667	-5,1852	-51,5625
9		GROWTH_IN_TOT_LOAN	22,1255	44,1342	30,7444	-12,9675
10		GROWTH_IN_TOT_DPST	54,2751	45,3701	9,7937	-2,0948
11	HBNC US Equity	BS_NON_PERFORM_ASSET	1,845	2,625	3,1	7,863
12		TANGIBLE_COMMON_EQUITY	44,963	53,678	62,79	71,658
13		BS_RSRV_LOAN_LOSS	8,368	8,738	9,791	11,41
14		EPS_GROWTH	-0,431	2,1645	7,6271	9,4488
15		GROWTH_IN_TOT_LOAN	29,9077	16,9506	3,7243	-0,1046
16		GROWTH_IN_TOT_DPST	39,7488	6,8267	-2,2221	-5,8741

Figura 5 – Esempio di dati esportati in formato long e con data unica per tutti i ticker

	A	B	C	D	E
1	Data iniziale	01/01/2016			
2	Data finale	30/06/2024			
3					
4	AAPL UW Equity	Dates	29/01/2016	29/02/2016	26/03/2016
5		HISTORICAL_MARKET_CAP	#N/A N/A	#N/A N/A	578907,3888
6		ENTERPRISE_VALUE	#N/A N/A	#N/A N/A	425851,3888
7		PX_TO_BOOK_RATIO	4,2076	4,1795	4,1795
8		TOT_COMMON_EQY	#N/A N/A	#N/A N/A	130457
9		BS_TOT_ASSET	#N/A N/A	#N/A N/A	305277
10		RETURN_COM_EQY	#N/A N/A	#N/A N/A	39,0638
11		ESG_SCORE	#N/A N/A	#N/A N/A	#N/A N/A
12		SA_ESG_RISK_SCR	#N/A N/A	#N/A N/A	#N/A N/A
13		CUMULATIVE_TOT_RETURN_GROSS_DVDS	-7,5242	-7,6473	-7,6473
14		BETA_RAW_OVERRIDABLE	1,4092	1,3362	1,3362
15		DAY_TO_DAY_TOT_RETURN_GROSS_DVDS	-7,5242	-0,1331	-0,1331
16					
17	A UN Equity	Dates	29/01/2016	31/01/2016	29/02/2016
18		HISTORICAL_MARKET_CAP	#N/A N/A	12349,2	12349,2
19		ENTERPRISE_VALUE	#N/A N/A	12154,2	12154,2
20		PX_TO_BOOK_RATIO	3,053	3,053	3,0286
21		TOT_COMMON_EQY	#N/A N/A	4045	4045
22		BS_TOT_ASSET	#N/A N/A	7302	7302
23		RETURN_COM_EQY	#N/A N/A	11,0843	11,0843
24		ESG_SCORE	#N/A N/A	#N/A N/A	#N/A N/A
25		SA_ESG_RISK_SCR	#N/A N/A	#N/A N/A	#N/A N/A
26		CUMULATIVE_TOT_RETURN_GROSS_DVDS	-9,9498	-9,9498	-10,6673
27		BETA_RAW_OVERRIDABLE	1,3419	1,3419	1,5814
28		DAY_TO_DAY_TOT_RETURN_GROSS_DVDS	-9,9498	-9,9498	-0,7968
29					

Figura 6 – Esempio di dati esportati in formato long e con data singola per ciascun ticker

## 2. Sintassi

Per installare il comando, dalla command bar di Stata digitate:

```
net install bloomimport, from(https://raw.githubusercontent.com/NicolaTommasi8/bloomimport/main/)
```

Poi usate il comando `ado update`, `update` permetterà di aggiornare il comando ad eventuali nuove versioni.

La sintassi del comando è la seguente:

```
bloomimport using filename, sheet("sheetname") cellrange([start][:end])
[export(string) dates(string) nvar(#) lasttick(string)
from(string) to(string)]
```

dove:

**filename** percorso e nome del file Excel dei dati esportati da Bloomberg

**sheet(sheetname)** foglio in cui sono presenti i dati

**cellrange([start][:end])** range di celle da importare dal foglio excel. Il range viene specificato usando la notazione di Excel, per esempio `cellrange(A1)` o `cellrange(A1:BC2000)`. Non serve specificare l'intero range, basta specificare la cella di partenza da cui leggere i dati; nell'esempio riportato in

Figura 3 sarebbe `cellrange(A4)`. Quando si usa l'opzione `dates(multi)` potrebbe succedere che il comando `import excel` non riesca ad individuare correttamente tutti i dati da caricare; in questo caso usare la forma completa di `cellrange([start][:end])` specificando sia `[start]` che `[:end]`

**export(string)** specifica la disposizione con cui sono stati esportati i dati. La disposizione può essere `export(wide)` (opzione di default), oppure `export(long)`

**dates(string)** specifica se la data è unica per tutti i ticker o se ogni ticker ha una proprio colonna o riga per la data. uò essere `dates(single)` (opzione di default), oppure `dates(multi)`

**nvar(#)** specifica il numero di campi esportati per ciascun titolo (opzione obbligatoria se `export(wide)`)

**lasttick(string)** specifica la colonna dell'ultimo ticker dei dati da importare, da indicare secondo le regole di Excel (opzione obbligatoria se `export(wide)`).

**from(varlist)** lista dei campi che devono essere rinominati, tipicamente perché il nome è troppo lungo o perché il nome sarebbe incompatibile con le regole di Stata sui nomi delle variabili (è ancora da vedere bene se farlo e come farlo, per ora non usare questa opzione!)

**to(varlist)** lista dei nuovi nomi (nello stesso ordine) da assegnare ai campi specificati in `from(varlist)` (per ora non usare questa opzione, vedi `from(varlist)`)

`bloomimport` cancella gli eventuali dati in memoria di Stata e li sostituisce con quelli caricati dal comando.

## 2.1. Esempi

Nel primo esempio si importano i dati esportati in formato wide dal foglio "Foglio1" dove ci sono 32 campi per ciascun ticker e l'ultimo di questi ticker è nella colonna DEP del foglio medesimo. L'opzione `dates()` non è specificata e quindi si assume che sia `dates(single)`

```
. bloomimport using "data/Vantaggio competitivo e WACC.xlsx", cellrange(A4) ///
> sheet("Foglio1") nvar(32) lasttick(DEP)

. summ
```

Variable	Obs	Mean	Std. dev.	Min	Max
-----+-----					
ticker	0				
date	0				
ENTERPRISE~E	1,242	11074.45	22131.1	5.125	275673.4
CF_FREE_CA~M	1,150	464.8148	1048.991	-3928.55	8677.674
FREE_CASH_~Y	1,220	535.3761	1699.12	-5986	21368
-----+-----					
CF_FREE_CA~2	1,252	322.9462	957.2912	-5003	8431
WACC	1,250	7.00434	2.775853	-25.4714	21.747
WACC_COST_~Y	1,250	9.316001	3.114569	-25.6074	23.8439
WACC_WEIGH~Y	1,250	71.21741	22.55256	1.8608	100
WACC_COST_~T	1,250	1.568361	1.5902	-.9692	8.7885
-----+-----					
WACC_WEIGH~T	1,250	28.67527	22.45683	0	98.1392
SHORT_AND_~T	1,252	3320.436	7371.259	0	66354
CASH_AND_M~S	1,254	1399.036	3407.764	.083	36060
FCF~R_GROWTH	712	11.17473	23.38947	-72.5478	226.1581
FREE_CASH_~H	850	9.434474	28.7091	-54.3235	310.6351
-----+-----					
FCF~L_GROWTH	1,088	88.41307	1820.618	-19456.96	32075.89
HISTORICAL~P	93	227.6864	316.9819	14.3072	1669.341
EBIT	1,254	658.7461	1528.055	-16169	13266
IS_INC_TAX~P	1,254	150.5178	341.3357	-1689	2932

CAPITAL_EX~D	1,252	-541.5659	1569.208	-16797	0
-----					
CF_CASH_FR~T	1,252	-729.68	2334.02	-34399	14286
IS_IMPAIR~S	745	96.29379	481.9777	-103	9158
IS_IMPAIR~L	834	101.086	469.6049	-11	4786
IS_IMPAIR~S	475	36.19235	100.9268	-50	1126
CF_CHNG_NO~P	1,238	-8.101133	397.3753	-2390	8759
-----					
RETURN_COM~Y	1,246	8.061317	28.50765	-527.0218	239.998
RETURN_ON~T	1,248	3.600192	15.08787	-136.1412	429.4941
RETURN_ON~P	1,221	5.655745	22.31824	-231.2528	139.3538
RETURN_ON~L	1,221	5.172286	25.98726	-734.1741	131.1622
OPER_MARGIN	1,254	43.76568	2343.848	-11153.5	81623.19
-----					
PE_RATIO	93	46.10994	106.7487	.1578	1025
PX_TO_BOOK~O	93	1.471926	1.489012	.2286	6.6377
EV_TO_T12M~A	1,226	53.0463	851.3633	.4938	21008.69
NET_DEBT_T~A	1,236	9.017561	169.2792	-38.8095	4180

Nel secondo esempio il numero di campi è 13 e l'ultimo ticker è posizionato nella colonna GQO

```
. set maxvar 8000
```

```
. bloomimport using "data/us_banks.xlsx", cellrange(A4) sheet("US1") ///
> nvar(13) lasttick(GQO)
```

```
. summ
```

Variable	Obs	Mean	Std. dev.	Min	Max
-----					
ticker	0				
date	0				
NET_INT_INC	26,861	94.85882	565.9889	-449.202	12644
IS_ACT_LOA~T	25,901	12.36292	151.3245	-576	16609
BS_TOT_LOAN	26,745	8090.337	46071.2	0	1033521
-----					
BS_LOAN_MTG	26,751	7966.754	45301.14	0	1022258
BS_TOTAL_D~S	3,903	457.9059	2468.646	-2.345	37205
BS_DEMAND~T	26,336	2634.726	16761.92	0	529051
INTEREST_B~O	26,605	6670.138	40938.49	0	996854
BS_CUSTOME~S	26,736	9263.505	57186.44	0	1482479
-----					
TOTAL_EQUITY	26,783	1384.682	8689.521	-.385	208079
BS_TIER1_C~O	26,077	13.74289	80.34161	0	12960
BS_NON_PER~T	26,140	104.7876	875.102	0	34573
BS_TOT_ASSET	26,751	12827.44	82475.08	0	1981349
BS_CASH_NE~M	26,738	283.0214	1641.788	0	63904

Nel terzo esempio, si importano i dati in formato long

```
. bloomimport using "data/File270524bis.xlsx", sheet("FoglioA_download") cellrange(A4) ///
> export(long)
```

```
. summ
```

Variable	Obs	Mean	Std. dev.	Min	Max
----------	-----	------	-----------	-----	-----

ticker	0				
date	0				
ANN_NET_IN~N	1,545	3.650048	2.52467	.9355	92.0942
ARDR_LIQUI~O	38	118.4816	10.27879	106	165
ARDR_NET_S~O	14	3.43e+10	2.25e+10	124	4.80e+10
BS_CUSTOME~S	1,557	35835.06	131601.6	219.2884	1482479
BS_DEMAND_~T	1,525	10774.64	38850.63	10.323	527748
BS_LEV_RAT~P	1,531	9.64031	1.861595	4	19.4
BS_LIQUIDI~O	38	118.4816	10.27879	106	165
BS_NON_PER~T	1,528	480.0016	4973.84	0	181872
BS_NON_PER~S	1,522	416.6706	4513.457	0	166212
BS_RE_LOAN	1,554	14605.54	49766.79	66.217	495842
BS_RISK_WE~S	1,524	61941.18	957363	0	3.70e+07
BS_RSRV_LO~S	1,557	458.3737	1773.278	1.453	24516
BS_T~P_RATIO	1,544	12.80234	2.890617	6	32.57
BS_T~Y_RATIO	895	11.65907	2.740535	-.1206	20.88
BS_TOT_ASSET	1,557	49138.65	186836.3	258.9534	1952911
BS_TOT_CAP~P	1,542	14.59836	2.864258	8	32.57
BS_TOT_LOAN	1,557	29945.87	102357.4	182.2295	993993
COM_EQY_TO~T	1,557	9.805642	2.3783	-1.6355	19.8588
EFF_RATIO	1,556	60.47438	13.45661	11.1733	165.3167
EPS_GROWTH	1,530	16.11401	177.2297	-2575	3400
FDIC~250000K	18	12881.62	13000.88	1273.907	36408.25
GROWTH_IN_~T	1,540	10.90226	18.44787	-33.7049	362.9414
GROWTH_IN_~N	1,540	10.80239	18.11365	-63.1226	319.6432
HISTORICAL~P	1,480	6492.774	24236.84	9.7506	296774.4
IS_ACT_LOA~T	1,538	162.7568	880.0535	-335	18168
NET_INC_GR~H	1,538	16.53337	220.1798	-2618.819	5497.222
NON_PERFOR~N	1,528	1.818147	15.20118	0	574.5461
NON_PERF_A~T	1,528	1.245741	11.03249	0	419.922
NUM_OF_EMP~S	1,529	7255.333	28973.21	28	281000
RETURN_COM~Y	1,538	9.143434	14.5989	-264.7305	203.696
RETURN_ON_~T	1,540	.9632708	.784239	-6.0911	7.0985
RETURN_TOT~Y	1,538	9.511867	9.280677	-124.8476	74.5401
TANGIBLE_C~Y	1,523	3584.578	13595.56	-109.045	149373
TEXAS_RATIO	1,492	16.06594	148.3282	0	5481.544
TOT_LOAN_T~T	1,557	87.59001	17.23081	15.3967	219.9147

In questo caso si importano i dati in formato wide con date multiple per ciascun ticker

```
. bloomimport using "data/File per Nicola 260824.xlsx", cellrange(A4) sheet("companies") ///
> nvar(11) lasttick(IQA) dates(multi)
```

```
. summ
```

Variable	Obs	Mean	Std. dev.	Min	Max
ticker	0				
date	0				
HISTORICAL~P	55,908	60604.69	153900.4	62.8917	3322626

ENTERPRISE~E	54,396	71091.17	161826.6	-13806.28	3344935
PX_TO_BOOK~O	54,820	16.93354	124.657	.1528	7090.25
-----					
TOT_COMMON~Y	56,350	16462.51	35766.19	-18316	601697
BS_TOT_ASSET	56,350	77181.8	252194.6	152.509	4143003
RETURN_COM~Y	54,399	31.30487	100.6898	-435.7846	2065.267
ESG_SCORE	51,213	4.0367	1.365718	.65	8.63
SA_ESG_RIS~R	20,115	21.70186	7.148309	5.28	47.93
-----					
CUMULATIVE~S	56,665	120.2247	304.017	-87.7504	15257.94
BETA_RAW_O~E	56,549	1.042058	4.593328	-845.781	453.4317
DAY_TO_DAY~S	56,609	1.365365	9.225383	-78.6728	494.5685

Infine, in questo caso si importano i dati in formato long con date multiple per ciascun ticker

```
. bloomimport using "data/File per Nicola 260824.xlsx", cellrange(A4) sheet("multi_long") ///
> dates(multi) export(long)
```

```
. summ
```

Variable	Obs	Mean	Std. dev.	Min	Max
-----					
ticker	0				
date	0				
DBETA_RAW~E	1,091	1.065243	.6020008	-5.8442	2.3763
DBS_TOT_AS~T	986	86266.49	107075.7	6613.089	406794
DENTERPRIS~E	978	301271.1	592509.5	10558.49	3154375
-----					
DESG_SCORE	933	4.21433	1.398658	1.4	7.33
DHISTORICA~P	978	305557.5	615701.9	5154.987	3206112
DPX_TO_B00~O	992	16.32091	19.4739	.875	82.2959
DRETURN_CO~Y	966	47.19229	43.88256	-9.1584	197.4311
DSA_ESG_RI~R	399	19.90053	6.138061	8.65	29.93
-----					
DTOT_COMMO~Y	986	21838.51	31409.93	-8940	140199
CUMULATIVE~S	1,093	130.9515	161.4119	-74.054	781.5646
DAY_TO_DAY~S	1,092	1.323135	8.210708	-36.0105	30.1131

```
. qui log close
```

Il comando non converte la variabile date in variabile numerica ma la lascia in formato stringa come presente nel file excel. Dato che le possibili formattazioni di questa variabile sono molte numerose, questa è una operazione da fare dopo l'esecuzione di blomimport in base alla specifica formattazione di date. Quello che segue è la costruzione delle variabili data e monthly a partire dalla variabile stringa date dell'ultimo esempio

```
. desc date
```

Variable name	Storage type	Display format	Value label	Variable label
-----				
date	str9	%9s		

```
. gen data=date(date,"DMY")
```

```
. order data, after(date)
```



```
. format data %td
. gen monthly= mofd(data)
. format monthly %tm
. fre data
```

data

		Freq.	Percent	Valid	Cum.
Valid	29jan2016	9	0.81	0.81	0.81
	31jan2016	1	0.09	0.09	0.90
	29feb2016	9	0.81	0.81	1.71
	04mar2016	1	0.09	0.09	1.80
	26mar2016	1	0.09	0.09	1.89
	31mar2016	10	0.90	0.90	2.79
	29apr2016	9	0.81	0.81	3.60
	30apr2016	1	0.09	0.09	3.69
	31may2016	9	0.81	0.81	4.50
	03jun2016	1	0.09	0.09	4.59
	25jun2016	1	0.09	0.09	4.68
	30jun2016	10	0.90	0.90	5.58
	29jul2016	9	0.81	0.81	6.38
	31jul2016	1	0.09	0.09	6.47
	31aug2016	9	0.81	0.81	7.28
	02sep2016	1	0.09	0.09	7.37
	24sep2016	1	0.09	0.09	7.46
	30sep2016	10	0.90	0.90	8.36
	31oct2016	9	0.81	0.81	9.17
	30nov2016	9	0.81	0.81	9.98
	:	:	:	:	:
	01sep2023	1	0.09	0.09	88.76
	29sep2023	10	0.90	0.90	89.66
	30sep2023	6	0.54	0.54	90.20
	31oct2023	10	0.90	0.90	91.10
	30nov2023	10	0.90	0.90	92.00
	01dec2023	1	0.09	0.09	92.09
	29dec2023	10	0.90	0.90	92.99
	30dec2023	1	0.09	0.09	93.08
	31dec2023	5	0.45	0.45	93.53
	31jan2024	10	0.90	0.90	94.42
	29feb2024	10	0.90	0.90	95.32
	01mar2024	1	0.09	0.09	95.41
	29mar2024	10	0.90	0.90	96.31
	30mar2024	1	0.09	0.09	96.40
	31mar2024	5	0.45	0.45	96.85
	30apr2024	10	0.90	0.90	97.75
	31may2024	10	0.90	0.90	98.65
	28jun2024	9	0.81	0.81	99.46
	29jun2024	1	0.09	0.09	99.55
	30jun2024	5	0.45	0.45	100.00
	Total	1112	100.00	100.00	

```
. fre monthly
```

monthly

		Freq.	Percent	Valid	Cum.
Valid	2016m1	10	0.90	0.90	0.90
	2016m2	9	0.81	0.81	1.71
	2016m3	12	1.08	1.08	2.79
	2016m4	10	0.90	0.90	3.69
	2016m5	9	0.81	0.81	4.50
	2016m6	12	1.08	1.08	5.58
	2016m7	10	0.90	0.90	6.47
	2016m8	9	0.81	0.81	7.28
	2016m9	12	1.08	1.08	8.36
	2016m10	9	0.81	0.81	9.17
	2016m11	9	0.81	0.81	9.98
	2016m12	16	1.44	1.44	11.42
	2017m1	9	0.81	0.81	12.23
	2017m2	9	0.81	0.81	13.04
	2017m3	11	0.99	0.99	14.03
	2017m4	11	0.99	0.99	15.02
	2017m5	9	0.81	0.81	15.83
	2017m6	11	0.99	0.99	16.82
	2017m7	10	0.90	0.90	17.72
	2017m8	9	0.81	0.81	18.53
	:	:	:	:	:
	2022m11	10	0.90	0.90	79.50
	2022m12	17	1.53	1.53	81.03
	2023m1	10	0.90	0.90	81.92
	2023m2	10	0.90	0.90	82.82
	2023m3	11	0.99	0.99	83.81
	2023m4	12	1.08	1.08	84.89
	2023m5	10	0.90	0.90	85.79
	2023m6	11	0.99	0.99	86.78
	2023m7	11	0.99	0.99	87.77
	2023m8	10	0.90	0.90	88.67
	2023m9	17	1.53	1.53	90.20
	2023m10	10	0.90	0.90	91.10
	2023m11	10	0.90	0.90	92.00
	2023m12	17	1.53	1.53	93.53
	2024m1	10	0.90	0.90	94.42
	2024m2	10	0.90	0.90	95.32
	2024m3	17	1.53	1.53	96.85
	2024m4	10	0.90	0.90	97.75
	2024m5	10	0.90	0.90	98.65
	2024m6	15	1.35	1.35	100.00
	Total	1112	100.00	100.00	

. qui log close

Nel file "imp\_bloom.do", oltre al codice per replicare i due precedenti esempi, sono riportati ulteriori esempi relativi a dati presenti nel file testXbloomimport.xlsx.

### 3. Note Finali

bloomimport usa il comando aggiuntivo xframeappend di Roger Newson (Imperial College London, United Kingdom) e include un pezzo di codice scritto da William Matsuoka per convertire l'identificativo delle

colonne di Excel, nel corrispondente numero di colonna. Trovate l'articolo e il codice qui: [Putexcel Part II: numofbase26\(\)](#).

Dati esportati con layout differente da quello previsto per il comando `bloomimport` non dovrebbero funzionare, in questo caso provate a sottoporvi i vostri dati e vediamo cosa si può fare.

Se un campo viene, per errore, esportato due volte, per la seconda occorrenza viene aggiunto 2 alla fine del nome della variabile risultante in Stata. Per adesso non è contemplato il caso in cui lo stesso campo venga esportato tre o più volte.

Il comando funziona solo dalla versione 17 di Stata perché usa i frame per costruire il database finale. Il codice che consente di far girare il comando in una versione precedente è in versione sperimentale, ma già inserito nel file `.ado`. In teoria funziona, in pratica ... ai posteri l'ardua sentenza

Siamo ancora in fase di testing con diverse fonti di dati esportate da Bloomberg, quindi il comando è da ritenersi ancora in fase beta.

## Riferimenti bibliografici

- [1] Roger Newson, 2021. "XFRAMEAPPEND: Stata module to append data frames to the end of the current data frame", Statistical Software Components S458923, Boston College Department of Economics. <https://ideas.repec.org/c/boc/bocode/s458923.html>.
- [2] Nicola Tommasi, 2022. [Pagina del comando su GitHub](#).
- [3] William Matsuoka, 2015. [Putexcel Part II: numofbase26\(\)](#).