

Getting data for all equities for a particular country in Datastream using R

Aaron Kaijser

29-10-2021

This post is made for those who do not have access to Refinitiv's (formerly Thomson Reuters) Datastream API and are required to use the Refinitiv Eikon Excel add-on instead (most likely university students). However, if you do have API access, you can still follow this guide as the code below can still be useful. If you are not familiar with R you can still copy and paste the code below. Install R and Rstudio and you should be good to go.

I will show you how to acquire a list of equities for a particular country for (international) asset pricing research purposes based on the most recent recommendations in financial research.

As an example, I will demonstrate how to obtain a list of equities for The Netherlands that is comprised of dead and active equities. The same principles can be applied to other countries, albeit that you might have to make some small adjustments (more on that later).

The datasets and script can be found here: <https://github.com/aaron-kaijser/Datastream-Equities>

Constituent Lists

The first step in obtaining your sample of international stocks is to use the right constituent lists. Researchers mostly use Datastream, Worldscope and so-called "dead" lists (you want to include dead equities to eliminate survivorship bias). Asset pricing literature provides a good indication of which constituent lists you should use but may vary per country. For example, for the United Kingdom, I have often seen constituent lists *WSCOPEUK*, *FBRIT* and *DEADUK* being used. However, some researchers may use additional (user-created) constituent lists besides these 3. However, when I was writing my thesis I found that adding additional lists barely made a difference. After applying static and dynamic filters (more on those later) I usually ended up with approximately the same sample.

As a starting point, I would suggest using the constituent lists mentioned in Schmidt et al. (2015) and other recent (working) papers such as the one by Hanauer and Windmüller (2020). Tip: if you directly search for "*WSCOPEUK DEADUK FBRIT DATASTREAM*", Google will automatically show you the titles of (recent) papers that have used these lists as well as any additional lists.

Since I am looking to acquire all equities for The Netherlands I will use the following lists: *WSCOPENL*, *FHOL* and *DEADNL*.

Static Screens

As a second step you most likely want to exclude non-common equities from your sample. Ince and Porter (2006) among others identify multiple screens to filter out non-common equities. These are called static screens. The following screens are commonly used in financial research:

Datastream Item	Description
MAJOR = “Y”	Only keeps major listings (i.e. firms that have more listed securities only the largest and most liquid one is used)
GEOGN = [country]	The company itself must be located in the country of interest (e.g. for the Netherlands this would be GEOGN = “NETHERLANDS”)
GEOLN = [country]	The securities must be listed in the country of interest
TYPE = “EQ”	The security must be an equity security (removes derivatives and debt securities etc.)
ISINID = “P”	Only keeps primary quotations of a security
PCUR = [currency]	Securities with a quoted currency that is different than the currency of the particular country of interest are removed (e.g. for Dutch equities PCUR should be equal to “E” and “F”. Note that “E” refers to the euro and “F” refers to The Netherlands’ former currency, the guilder)
GGISN = [country]	Securities with an international ISIN code that is different than the ISIN code of the country of interest are removed (e.g. GGISN = “NL”)

As you may have noticed, some of these filters are country dependent. For example, countries located in the eurozone had different currencies prior to the introduction of the euro. Moreover, there are some exceptions to the static screens mentioned above. For example, for Russian equities, it is recommended that PCUR is also set to “USD”, besides “RUB”. See Ince and Porter (2006) for more details.

Next to that, you should also delete any securities whose Datastream company names contain any of the generic Datastream keywords and/or country-specific keywords listed below. If Datastream variables *NAME*, *ENAME* and *ECNAME* contain any of these keywords, the respective equities are non-common equities and should therefore be removed from the sample.

Non-common Equity Types	Generic Keywords
Duplicates	1000DUPL, DULP, DUP, DUPE, DUPL, DUPLI, DUPLICATE, XSQ, XETa
Depository Receipts	ADR, GDR
Preferred Stock	PF, ‘PF’, PFD, PREF, PREFERRED, PRF
Warrants	WARR, WARRANT, WARRANTS, WARRT, WTS, WTS2
Debt	%, DB, DCB, DEB, DEBENTURE, DEBENTURES, DEBT
Unit Trusts	.IT, .ITb, TST, INVESTMENT TRUST, RLST IT, TRUST, TRUST UNIT, TRUST UNITS, TST, TST UNIT, TST UNITS, UNIT, UNIT TRUST, UNITS, UNT, UNT TST, UT
ETFs	AMUNDI, ETF, INAV, ISHARES, JUNGE, LYXOR, X-TR
Expired Securities	EXPD, EXPIRED, EXPIRY, EXPY

Non-common Equity Types	Generic Keywords
Other	ADS, BOND, CAP.SHS, CONV, DEFER, DEP, DEPY, ELKS, FD, FUND, GW.FD, HI.YIELD, HIGH INCOME, IDX, INC.&GROWTH, INC.&GW, INDEX, LP, MIPS, MITS, MITT, MPS, NIKKEL, NOTE, OPCVM, ORTF, PARTNER, PERQS, PFC, PFCL, PINES, PRTF, PTNS, PTSHP, QUIBS, QUIDS, RATE, RCPTS, REAL EST, RECEIPTS, REIT, RESPT, RETUR, RIGHTS, RST, RTN.INC, RTS, SBVTG, SCORE, SPDR, STRYPES, TOPRS, UTS, VCT, VTG.SAS, XXXXX, YIELD, YLD

The country-specific keywords are listed below.

Country	Keywords
Australia	PART RAID, RTS DEF, DEF SETT, CDI
Austria	PC, PARTICIPATION CERTIFICATE, GENUSSSCHEINE, GENUSSSCHEINE
Belgium	VVPR, CONVERSION, STRIP
Brazil	PN, PNA, PNB, PNC, PND, PNE, PNF, PNG, RCSA, RCTB
Canada	EXCHANGEABLE, SPLIT, SPLITSHARE, VTG\\., SBVTG\\., VOTING, SUB VTG, SERIES \\)CSE\\)
Denmark	USE
Finland	ADP, CI, SICAV, \\)SICAV\\), SICAV-
France	GENUSSSCHEINE
Germany	PR
Greece	FB DEAD, FOREIGN BOARD
India	P1, 1, 5
Israel	RNC, RP, PRIVILEGIES
Italy	1P
Korea	'L', 'C'
Mexico	'A'
Malaysia	CERTIFICATE, CERTIFICATES, CERTIFICATES\\), CERT, CERTS, STK\\.
Netherlands	RTS, RIGHTS
New Zealand	INVERSION, INVN, INV
Peru	PDR
Philippines	N', OPTS\\., CPF\\., CUMULATIVE
South Africa	PREFERENCE
Sweden	CONVERTED INTO, USE, CONVERTED-, CONVERTED - SEE
Switzerland	CONVERTED INTO, CONVERSION, CONVERSION SEE
United Kingdom	PAID, CONVERSION TO, NON VOTING, CONVERSION 'A'

Obtaining Initial List of Equities

Open up Refinitiv Eikon's Excel application, create a new workbook and sign in. Next, go to the Refinitiv Eikon Datastream tab and select **Static Request**. The figure below shows the window that should appear.

The screenshot shows the 'Static Request' dialog box. It has a title bar with a close button. The main area is divided into sections: 'Request Details' and 'Options'. In 'Request Details', there are two dropdown menus: 'Series/List' and 'Datatypes/Expressions'. The 'Series/List' dropdown has a 'Find Series' button next to it. Below it is a checkbox for 'RIC'. The 'Datatypes/Expressions' dropdown has a 'Datatypes' button next to it. There is also a 'Date' dropdown set to 'Latest Value' and a 'History' button. The 'Options' section contains several checkboxes: 'Display Custom Header' (disabled), 'Display Row Titles' (checked), 'Display Column Titles' (checked), 'Display Headings' (unchecked), 'Transpose Data' (unchecked), 'Display Code' (unchecked), 'Display Currency' (unchecked), 'Display Latest Value First' (unchecked), 'Hyperlink to Series Metadata' (unchecked), 'Hyperlink to Datatype Definition' (unchecked), 'Display Datatype' (radio buttons for 'Description' and 'Mnemonic', with 'Description' selected), 'Display Expression' (radio buttons for 'Description' and 'Number', with 'Description' selected), 'Embed Formula' (checked), and 'Auto Refresh' (checked). There is an 'Edit' button next to 'Display Custom Header'. At the bottom, there are buttons for 'Help', 'Default Option', 'Submit', and 'Cancel'.

Figure 1: Static request in Datastream

I will start by obtaining the constituent list *WSCOPENL*. In the *Series/List* field you should type **WSCOPENL**. Note that you can also use the *Find Series* button to find constituent lists. However, from my personal experience, this did not always work. I do not include the other constituent lists yet, since I will have to repeat the process outlined below for the other 2 constituent lists.

In the *Datatypes/Expressions* field select the following Datastream items:

- MAJOR
- GEOGN
- GEOLN
- TYPE
- ISINID
- PCUR
- GGISN
- NAME
- ENAME
- EENAME

- **ISIN** (although I did not mention this above, obtaining the ISIN codes is hugely beneficial as it allows you to look up information for a particular firm on the internet, something that is not possible with Datastream's default identifier code. Moreover, it allows you to create a .txt file which can serve as input in Compustat Global for example.)

Leave the default boxes checked as they are and you should obtain the output below. Datastream adds the first column 'Type' by default, which lists Datastream's unique identifier codes.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Date	23-9-2021											
2	Type	MAJOR FL	STOCK TY	QUOTE IN	GEOGRAP	GEOG DES	CURRENC	ISIN	ISSUE	NAME	FULL NAM	COMPANY	ISIN CODE
3	41294N	Y	GDR	P	NETHERLA	UNITED KI	E	US		A&D PHAF	A&D PHAF	A&D PHAF	US0024812085
4	51134P	Y	EQ	S	HUNGARY	CZECH REP	CK	NL		AAA AUTC	AAA AUTC	AAA AUTC	NL0006033375
5	749352	Y	EQ	P	NETHERLA	NETHERLA	E	NL		AALBERTS	AALBERTS	AALBERTS	NL0000852564
6	8721AT	Y	EQ	P	NETHERLA	NETHERLA	E	NL		ABN AMR	ABN AMR	ABN AMR	NL0011540547
7	505972	Y	EQ	P	NETHERLA	NETHERLA	E	NL		ABN AMR	ABN AMR	ABN AMR	NL0000301109
8	688324	Y	EQ	P	NETHERLA	NETHERLA	E	NL		ACCELL GF	ACCELL GF	ACCELL GF	NL0009767532
9	27187D	Y	EQ	P	NETHERLA	NETHERLA	E	N/A		ACHMEA F	ACHMEA F	ACHMEA F	NA
10	265628	Y	EQ	P	GERMANY	GERMANY	E	NL		AD PEPPE	AD PEPPE	AD PEPPE	NL0000238145
11	9307KF	Y	EQ	P	NETHERLA	NETHERLA	E	NL		ADYEN	ADYEN	ADYEN NV	NL0012969182
12	922956	Y	EQ	P	NETHERLA	NETHERLA	E	NL		AEGON	AEGON	AEGON	NL0000303709
13	539762	Y	UT	NA	NETHERLA	NA	E	NL		AEGON AA	AEGON AA	AEGON AA	NL0000284743

Figure 2: Datastream static request output for constituent list WSCOPENL

Save this file as a .CSV file (not .XLS or .XLSX), create a new workbook and repeat the process for your other constituent lists. In this example, I would have to repeat this for the *FHOL* and *DEADNL* lists.

Applying Static Screens in R

If you don't have the `data.table` and `stringr` packages installed yet, you should install them by using the following code:

```
install.packages(c("data.table", "stringr"))
```

Next, you should import the different .csv files you downloaded earlier. Since I use 3 constituent lists, I have 3 different datasets. You can use `data.table`'s `fread()` function to read .csv files.

```
library(data.table) # imports data.table library
library(stringr)   # imports stringr library

# Imports constituent lists (change the path to your
# directory)
WSCOPENL <- fread("C:/WSCOPENL.csv")
FHOL <- fread("C:/FHOL.csv")
DEADNL <- fread("C:/DEADNL.csv")
```

The raw output of the WSCOPENL file in R should look something like this (please note that I display the dataframe in tibble format for aesthetic purposes only - there is no need to convert the dataframe to tibble format):

```
## # A tibble: 10 x 12
```

```
##      V1      V2      V3      V4      V5      V6      V7      V8      V9      V10     V11     V12
##      <chr> <chr>      <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr> <chr>
## 1 Date   23-9-2021    ""      ""      ""      ""      ""      ""      ""      ""      ""      ""
## 2 Type   MAJOR FLAG  "STO~  "QUO~  "GEO~  "GEO~  "CUR~  "ISI~  "NAM~  "FUL~  "COM~  "ISI~
## 3 41294N Y          "GDR"  "P"    "NET~  "UNI~  "E"    "US"   "A&D~  "A&D~  "A&D~  "USO~
## 4 51134P Y          "EQ"   "S"    "HUN~  "CZE~  "CK"   "NL"   "AAA~  "AAA~  "AAA~  "NLO~
## 5 749352 Y          "EQ"   "P"    "NET~  "NET~  "E"    "NL"   "AAL~  "AAL~  "AAL~  "NLO~
## 6 8721AT Y          "EQ"   "P"    "NET~  "NET~  "E"    "NL"   "ABN~  "ABN~  "ABN~  "NLO~
## 7 505972 Y          "EQ"   "P"    "NET~  "NET~  "E"    "NL"   "ABN~  "ABN~  "ABN~  "NLO~
## 8 688324 Y          "EQ"   "P"    "NET~  "NET~  "E"    "NL"   "ACC~  "ACC~  "ACC~  "NLO~
## 9 27187D Y          "EQ"   "P"    "NET~  "NET~  "E"    "N/A"  "ACH~  "ACH~  "ACH~  <NA>
## 10 265628 Y         "EQ"   "P"    "GER~  "GER~  "E"    "NL"   "AD ~  "AD ~  "AD ~  "NLO~
```

Since you don't have any column names you should store the names of the second row in a variable `col_names`. You can first transpose the second row and then convert it to a vector class (the transpose function `t()` returns an array or matrix class by default):

```
col_names <- as.vector(t(WSCOPENL[2, ]))
```

```
## [1] "Type"           "MAJOR FLAG"      "STOCK TYPE"
## [4] "QUOTE INDICATOR" "GEOGRAPHIC DESCR." "GEOG DESC OF LSTNG"
## [7] "CURRENCY"       "ISIN ISSUER CTRY" "NAME"
## [10] "FULL NAME"      "COMPANY NAME"    "ISIN CODE"
```

You should then remove the first 2 rows as you do not need them and replace the current column names (V1 to V12) by the column names we just saved using `data.table`'s `setnames()` function:

```
WSCOPENL <- WSCOPENL[-c(1:2), ] # removes first two rows
setnames(WSCOPENL, old = colnames(WSCOPENL), new = col_names) # changes column names
```

```
## # A tibble: 10 x 12
##   Type 'MAJOR FLAG' 'STOCK TYPE' 'QUOTE INDICATOR' 'GEOGRAPHIC DESCR.'
##   <chr> <chr>      <chr>      <chr>      <chr>
## 1 41294N Y      GDR        P          NETHERLANDS
## 2 51134P Y      EQ         S          HUNGARY
## 3 749352 Y      EQ         P          NETHERLANDS
## 4 8721AT Y      EQ         P          NETHERLANDS
## 5 505972 Y      EQ         P          NETHERLANDS
## 6 688324 Y      EQ         P          NETHERLANDS
## 7 27187D Y      EQ         P          NETHERLANDS
## 8 265628 Y      EQ         P          GERMANY
## 9 9307KF Y      EQ         P          NETHERLANDS
## 10 922956 Y     EQ         P          NETHERLANDS
## # ... with 7 more variables: GEOG DESC OF LSTNG <chr>, CURRENCY <chr>,
## #   ISIN ISSUER CTRY <chr>, NAME <chr>, FULL NAME <chr>, COMPANY NAME <chr>,
## #   ISIN CODE <chr>
```

At this point, the `WSCOPENL` dataframe contains 435 unique stocks:

```
length(unique(WSCOPENL$`ISIN CODE`))
```

```
## [1] 435
```

You can then apply the rest of the static filters.

```
# 1. Major = Y
WSCOPENL <- WSCOPENL[`MAJOR FLAG` == "Y"]
# 2. Type = EQ
WSCOPENL <- WSCOPENL[`STOCK TYPE` == "EQ"]
# 3. ISINID = P
WSCOPENL <- WSCOPENL[`QUOTE INDICATOR` == "P"]
# 4. GEOGN = Country
WSCOPENL <- WSCOPENL[`GEOGRAPHIC DESCR.` == "NETHERLANDS"]
# 5. GEOLN = Country
WSCOPENL <- WSCOPENL[`GEOG DESC OF LSTNG` == "NETHERLANDS"]
# 6. PCUR = E or FL
WSCOPENL <- WSCOPENL[CURRENCY == "E" | CURRENCY == "FL"]
# FL was The Netherlands' currency prior to the euro (E) 7.
# GGISN = Country
WSCOPENL <- WSCOPENL[`ISIN ISSUER CTRY` == "NL"]
```

These static filters already filter out a large number of stocks. The total number of unique stocks is 287 at this point.

```
length(unique(WSCOPENL$`ISIN CODE`))
```

```
## [1] 287
```

Next, you should create a vector that contains the strings of all the generic keywords and the country-specific keywords mentioned above. I already went ahead and added the country-specific keywords for The Netherlands to the generic keywords. Note that in R a double backslash (`\\`) has to be duplicated since a double backslash will return an error. If you intend on obtaining equities for a country other than The Netherlands please make sure to include the country-specific keywords in this vector.

```
# 8. NAME/ENAME/ECNAME that are often linked to non-common
# stock must be removed
keyword_filter = c("1000DUPL", "DULP", "DUP", "DUPE", "DUPL",
  "DUPLI", "DUPLICATE", "XSQ", "XETa", "ADR", "GDR", "PF",
  "'PF'", "PFD", "PREF", "PREFERRED", "PRF", "WARR", "WARRANT",
  "WARRANTS", "WARRT", "WTS", "WTS2", "%", "DB", "DCB", "DEB",
  "DEBENTURE", "DEBENTURES", "DEBT", ".IT", ".ITb", "TST",
  "INVESTMENT TRUST", "RLST IT", "TRUST", "TRUST UNIT", "TRUST UNITS",
  "TST", "TST UNIT", "TST UNITS", "UNIT", "UNIT TRUST", "UNITS",
  "UNT", "UNT TST", "UT", "AMUNDI", "ETF", "INAV", "ISHARES",
  "JUNGE", "LYXOR", "X-TR", "EXPD", "EXPIRED", "EXPIRY", "EXPY",
  "ADS", "BOND", "CAP.SHS", "CONV", "DEFER", "DEP", "DEPY",
  "ELKS", "FD", "FUND", "GW.FD", "HI.YIELD", "HIGH INCOME",
  "IDX", "INC.&GROWTH", "INC.&GW", "INDEX", "LP", "MIPS", "MITS",
  "MITT", "MPS", "NIKKEI", "NOTE", "OPCVM", "ORTF", "PARTNER",
  "PERQS", "PFC", "PFCL", "PINES", "PRTF", "PTNS", "PTSHP",
  "QUIBS", "QUIDS", "RATE", "RCPTS", "REAL EST", "RECEIPTS",
  "REIT", "RESPT", "RETUR", "RIGHTS", "RST", "RTN.INC", "RTS",
  "SBVTG", "SCORE", "SPDR", "STRYPES", "TOPRS", "UTS", "VCT",
  "VTG.SAS", "XXXXX", "YIELD", "YLD", "CERTIFICATE", "CERTIFICATES",
  "CERT", "CERTS", "CERTIFICATES\\\\\\\\", "STK\\\\\\\\")
# very long keyword filter vector
```

You can then run a for loop which checks whether the Datastream variables NAME (Name), ENAME (Full name) and ECNAME (Company name) contain any of these forbidden keywords and excludes them from the WSCOPENL dataframe (notice the '!').

```
# Checks whether these variables contain any of the strings
# in the keyword filter and removes these observations
for (i in 1:length(keyword_filter)) {
  WSCOPENL <- WSCOPENL[!NAME %like% keyword_filter[i]]
  WSCOPENL <- WSCOPENL[!`FULL NAME` %like% keyword_filter[i]]
  WSCOPENL <- WSCOPENL[!`COMPANY NAME` %like% keyword_filter[i]]
}
```

The total number of unique stocks has now dropped to 251 stocks.

```
length(unique(WSCOPENL$`ISIN CODE`))
```

```
## [1] 251
```

The first 20 stocks in this case are:

```
## [1] "NL0011540547" "NL0000301109" "NL0009767532" "NL0012969182" "NL0000303709"
## [6] "NL0000018034" "NL0000335669" "NL0013267909" "NL0012817175" "NL0010696704"
## [11] "NL0011333752" "NL0012194724" "NL0000389799" "NL0000888691" "NL0000886968"
## [16] "NL0006237562" "NL0000334118" "NL0010273215" "NL0011872643" "NL0000335651"
```

Rather than repeating the process above for each of your constituent lists, you can simply run 1 function that I created. The function below does the exact same as the code I showed above and outputs the unique ISIN codes of each constituent list:

```
transform_static <- function(data) {
  # Data wrangling
  col_names <- as.vector(t(data[2, ])) # saves second row as vector
  data <- data[-c(1:2), ] # removes first two rows
  setnames(data, old = colnames(data), new = col_names) # changes column names

  # Applying filters 1. Major = Y
  data <- data[`MAJOR FLAG` == "Y"]
  # 2. Type = EQ
  data <- data[`STOCK TYPE` == "EQ"]
  # 3. ISINID = P
  data <- data[`QUOTE INDICATOR` == "P"]
  # 4. GEOGN = Country
  data <- data[`GEOGRAPHIC DESCR.` == "NETHERLANDS"]
  # 5. GEOLN = Country
  data <- data[`GEOG DESC OF LSTNG` == "NETHERLANDS"]
  # 6. PCUR = E or FL
  data <- data[CURRENCY == "E" | CURRENCY == "FL"] # FL is old currency
  # 7. GGISN = Country
  data <- data[`ISIN ISSUER CTRY` == "NL"]
  # 8. NAME/ENAME/ECNAME that are often linked to
  # non-common stock must be removed
  keyword_filter = c("1000DUPL", "DULP", "DUP", "DUPE", "DUPL",
```



```

"DUPLI", "DUPLICATE", "XSQ", "XETa", "ADR", "GDR", "PF",
"'PF'", "PFD", "PREF", "PREFERRED", "PRF", "WARR", "WARRANT",
"WARRANTS", "WARRT", "WTS", "WTS2", "%", "DB", "DCB",
"DEB", "DEBENTURE", "DEBENTURES", "DEBT", ".IT", ".ITb",
"TST", "INVESTMENT TRUST", "RLST IT", "TRUST", "TRUST UNIT",
"TRUST UNITS", "TST", "TST UNIT", "TST UNITS", "UNIT",
"UNIT TRUST", "UNITS", "UNT", "UNT TST", "UT", "AMUNDI",
"ETF", "INAV", "ISHARES", "JUNGE", "LYXOR", "X-TR", "EXPD",
"EXPIRED", "EXPIRY", "EXPY", "ADS", "BOND", "CAP.SH",
"CONV", "DEFER", "DEP", "DEPY", "ELKS", "FD", "FUND",
"GW.FD", "HI.YIELD", "HIGH INCOME", "IDX", "INC.&GROWTH",
"INC.&GW", "INDEX", "LP", "MIPS", "MITS", "MITT", "MPS",
"NIKKEI", "NOTE", "OPCVM", "ORTF", "PARTNER", "PERQS",
"PFC", "PFCL", "PINES", "PRTF", "PTNS", "PTSH", "QUIBS",
"QUIDS", "RATE", "RCPTS", "REAL EST", "RECEIPTS", "REIT",
"RESPT", "RETUR", "RIGHTS", "RST", "RTN.INC", "RTS",
"SBVTG", "SCORE", "SPDR", "STYPES", "TOPRS", "UTS",
"VCT", "VTG.SAS", "XXXXX", "YIELD", "YLD", "CERTIFICATE",
"CERTIFICATES", "CERT", "CERTS", "CERTIFICATES\\\\",
"STK\\\\")
# very long keyword filter vector (make sure to include
# the country-specific keywords for your country of
# interest!)

# Remove firms with keywords in name columns
for (i in 1:length(keyword_filter)) {
  data <- data[!NAME %like% keyword_filter[i]]
  data <- data[!`FULL NAME` %like% keyword_filter[i]]
  data <- data[!`COMPANY NAME` %like% keyword_filter[i]]
}

# replace `ISIN CODE` by `TYPE` if you would like to
# obtain a vector of Datastream identifier codes
# instead of ISIN identifiers. I would suggest ISIN
# codes so you can look up the firm later.
return(data$`ISIN CODE`)
}

```

You can therefore obtain 1 large vector which contains all the unique stocks of each constituent list using the custom `transform_static()` function. Passing multiple vectors as arguments to the `c()` function essentially combines all of them:

```

stock_vector <- c(transform_static(WSCOPENL), transform_static(FHOL),
  transform_static(DEADNL))

```

However, as all 3 constituent lists may partially contain the same equities, this vector contains is bound to contain duplicates. We only want to keep unique ISIN codes.

```

stock_vector <- unique(stock_vector) # removes duplicate values
length(stock_vector) # total number of unique stocks

```

```
## [1] 288
```

We end up with a total of 288 unique stocks for The Netherlands after applying the static filters.

```
head(stock_vector, 50) # prints first 50
```

```
## [1] "NL0011540547" "NL0000301109" "NL0009767532" "NL0012969182" "NL0000303709"
## [6] "NL0000018034" "NL0000335669" "NL0013267909" "NL0012817175" "NL0010696704"
## [11] "NL0011333752" "NL0012194724" "NL0000389799" "NL0000888691" "NL0000886968"
## [16] "NL0006237562" "NL0000334118" "NL0010273215" "NL0011872643" "NL0000335651"
## [21] "NL0000335743" "NL0000380210" "NL0000336006" "NL0012047823" "NL0000335974"
## [26] "NL0000336444" "NL0000336352" "NL0000336543" "NL0000337319" "NL0006292906"
## [31] "NL0012866412" "NL0000338465" "NL0000339612" "NL0000339703" "NL0000285278"
## [36] "NL0000343655" "NL0000335578" "NL0000340347" "NL0000341394" "NL0000852580"
## [41] "NL0000342632" "NL0000331650" "NL0010776944" "NL0000343119" "NL0000362432"
## [46] "NL0000343846" "NL0000344000" "NL0000344059" "NL0000344117" "NL0000344315"
```

I would then suggest saving this vector as a .txt file.

```
write.table(final_codes, "C:/NL_final_codes.txt", quote = F,
  row.names = F, col.names = F)
# the additional arguments quote, row.names and col.names
# are all set to FALSE such that the .txt file only
# contains the ISIN codes (without quotes)
```

This allows you to copy and paste the ISIN codes to Excel. Using the Refinitiv Eikon Excel add-on you can then create a custom list using these ISIN codes, which in turn allows you to download time-series data for each different stock.

Simply paste the entire ISIN code selection to your Excel sheet and go to the tab *Refinitiv Eikon Datastream* and click on *Lists (Create from Range)* and complete the subsequent steps. More information on creating custom constituent lists can be found in this Datastream user guide (page 23) on Refinitiv's website: https://fmc.refinitiv.com/clientFacing/pdf/DFO_User_Guide.pdf

There you go, you have now completed the first step in acquiring (panel) data for equity research. The next step is to download time-series data for each of the equities in your custom constituent list and apply dynamic screens which will shrink your sample even more. A guide on this is coming soon.

Extra: Saving Company Names

If you would like to save the company names along with the ISIN codes, replace the following line in the function:

```
return(data$`ISIN CODE`)
```

by this line of code, which returns a data.table with 2 columns:

```
return(data[, list(`ISIN CODE`, `COMPANY NAME`)])
# returns ISIN code and the corresponding company names
```

You can then create a unique stock vector with the corresponding company names by using the `rbind()` function:

```
stock_vector <- unique(rbind(transform_static(WSCOPENL), transform_static(FHOL),
  transform_static(DEADNL)), by = "ISIN CODE")
```

Which will yield the following result:

```
## # A tibble: 20 x 2
##   'ISIN CODE' 'COMPANY NAME'
##   <chr>      <chr>
## 1 NL0011540547 ABN AMRO BANK NV
## 2 NL0000301109 ABN AMRO HOLDING NV
## 3 NL0009767532 ACCELL GROUP NV
## 4 NL0012969182 ADYEN NV
## 5 NL0000303709 AEGON
## 6 NL0000018034 AFC AJAX NV
## 7 NL0000335669 AINO
## 8 NL0013267909 AKZO NOBEL NV
## 9 NL0012817175 ALFEN NV
## 10 NL0010696704 ALMUNDA PROFESSIONALS NV
## 11 NL0011333752 ALTICE EUROPE NV
## 12 NL0012194724 ALUMEXX NV
## 13 NL0000389799 AM NV
## 14 NL0000888691 AMG ADVANCED METALLURGICAL GROUP
## 15 NL0000886968 AMT HOLDING
## 16 NL0006237562 ARCADIS NV
## 17 NL0000334118 ASM INTERNATIONAL
## 18 NL0010273215 ASML HOLDING NV
## 19 NL0011872643 ASR NEDERLAND
## 20 NL0000335651 ASTRA INFORMATICA GROEP
```

References

- Hanauer, M. X., & Windmüller, S. (2020). Enhanced momentum strategies. Available at SSRN 3437919.
- Ince, O. S., & Porter, R. B. (2006). Individual equity return data from Thomson Datastream: Handle with care!. *Journal of Financial Research*, 29(4), 463-479.
- Schmidt, P. S., Von Arx, U., Schrimpf, A., Wagner, A. F., & Ziegler, A. (2015). On the construction of common size, value and momentum factors in international stock markets: A guide with applications. CCRS Working Paper Series, (01/11).