getfame

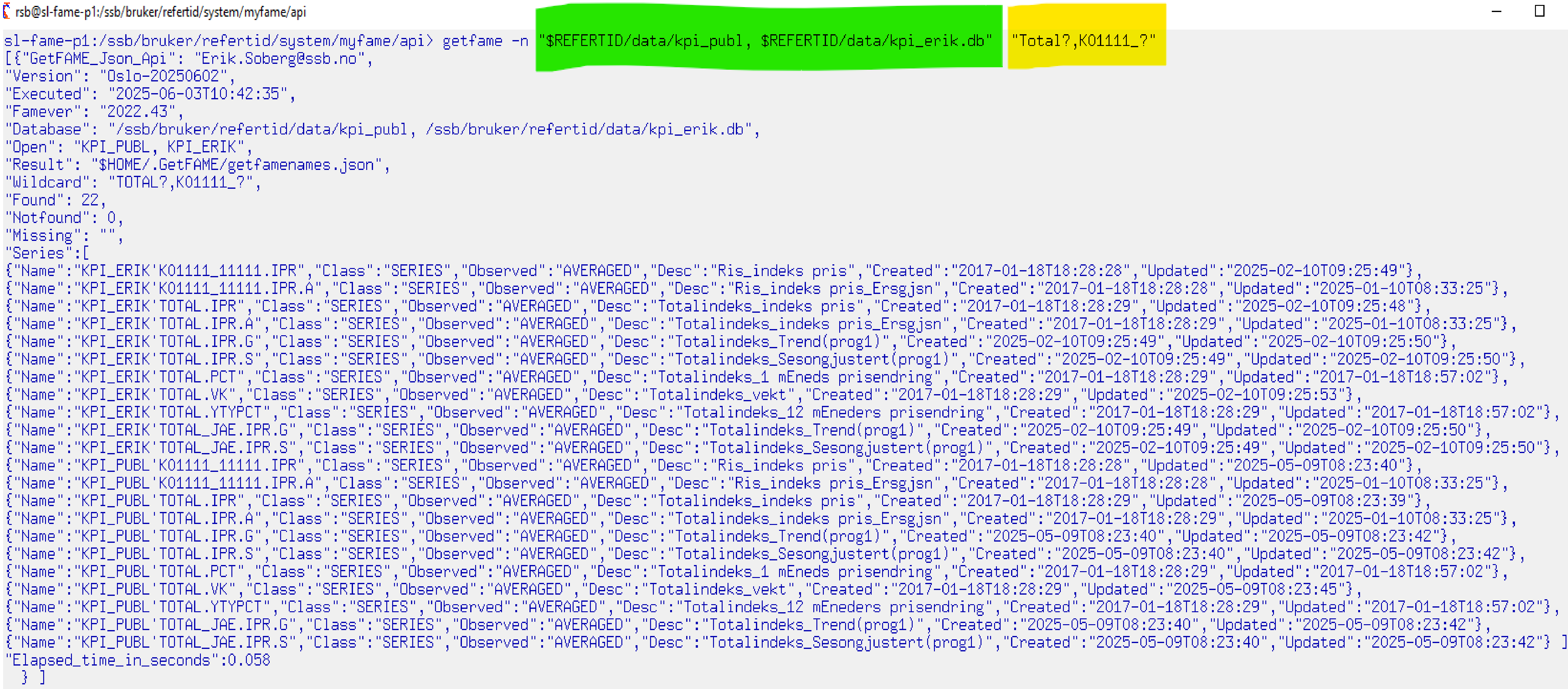
-n	names
-s	series
-e	**expression**

getfame Json api 2024 / 2025

Erik.Soberg@ssb.no

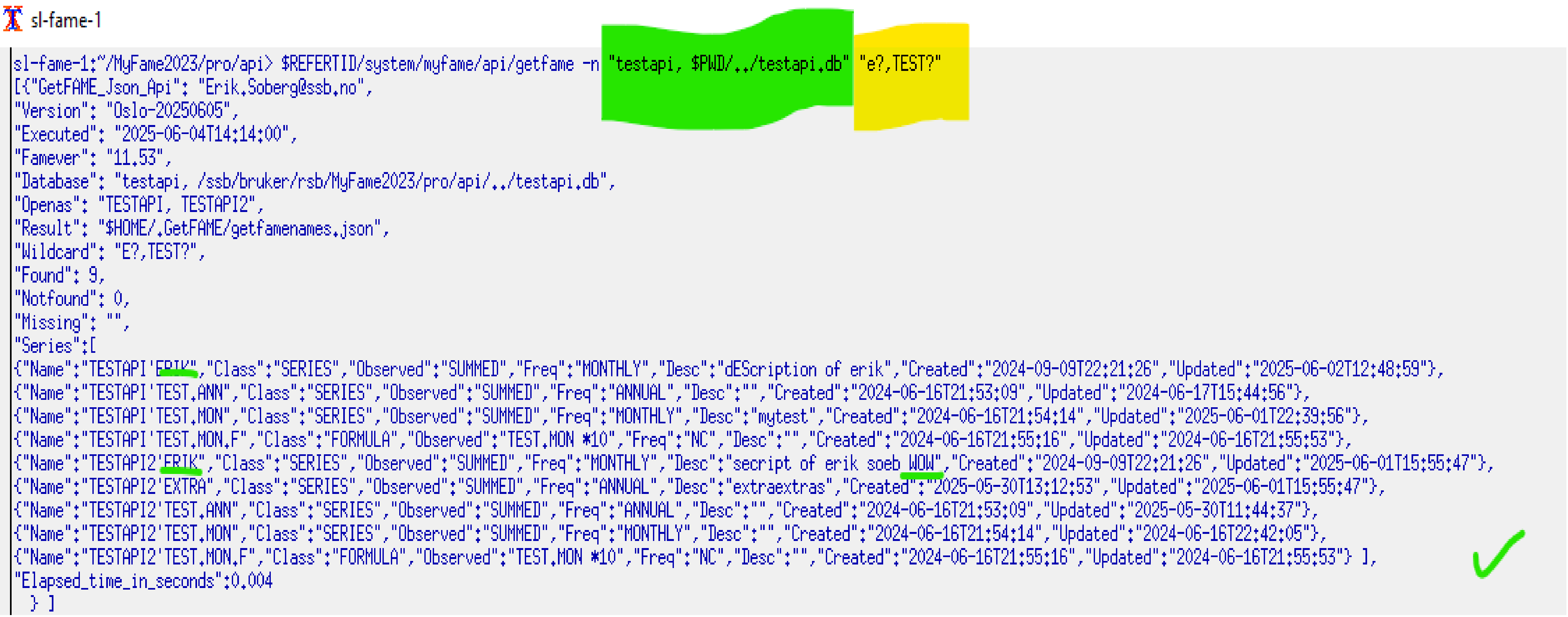
2025 Supports series w identical series names in different FAME databases, formulas can aggregate from several open databases

1. getfame -n = getfamenames gets FAME metadata



```
sl-fame-p1:/ssb/bruker/refertid/system/myfame/api> getfame -n "$REFERTID/data/kpi_publ, $REFERTID/data/kpi_erik.db" "Total?,K01111_?"
[{"GetFAME_Json_Api": "Erik.Soberg@ssb.no",
"Version": "Oslo-20250602",
"Executed": "2025-06-03T10:42:35",
"Famever": "2022.43",
"Database": "/ssb/bruker/refertid/data/kpi_publ, /ssb/bruker/refertid/data/kpi_erik.db",
"Open": "KPI_PUBL, KPI_ERIK",
"Result": "$HOME/.GetFAME/getfamenames.json",
"Wildcard": "TOTAL?,K01111_?",
"Found": 22,
"Notfound": 0,
"Missing": "",
"Series":[
{"Name":"KPI_ERIK'K01111_11111.IPR","Class":"SERIES","Observed":"AVERAGED","Desc":"Ris_indeks pris","Created":"2017-01-18T18:28:28","Updated":"2025-02-10T09:25:49"},
{"Name":"KPI_ERIK'K01111_11111.IPR.A","Class":"SERIES","Observed":"AVERAGED","Desc":"Ris_indeks pris_Ersgjsn","Created":"2017-01-18T18:28:28","Updated":"2025-01-10T08:33:25"},
{"Name":"KPI_ERIK'TOTAL.IPR","Class":"SERIES","Observed":"AVERAGED","Desc":"Totalindeks_indeks pris","Created":"2017-01-18T18:28:29","Updated":"2025-02-10T09:25:48"},
{"Name":"KPI_ERIK'TOTAL.IPR.A","Class":"SERIES","Observed":"AVERAGED","Desc":"Totalindeks_indeks pris_Ersgjsn","Created":"2017-01-18T18:28:29","Updated":"2025-01-10T08:33:25"},
{"Name":"KPI_ERIK'TOTAL.IPR.G","Class":"SERIES","Observed":"AVERAGED","Desc":"Totalindeks_Trend(prog1)","Created":"2025-02-10T09:25:49","Updated":"2025-02-10T09:25:50"},
{"Name":"KPI_ERIK'TOTAL.IPR.S","Class":"SERIES","Observed":"AVERAGED","Desc":"Totalindeks_Sesongjustert(prog1)","Created":"2025-02-10T09:25:49","Updated":"2025-02-10T09:25:50"},
{"Name":"KPI_ERIK'TOTAL.PCT","Class":"SERIES","Observed":"AVERAGED","Desc":"Totalindeks_1 mEneds prisendring","Created":"2017-01-18T18:28:29","Updated":"2017-01-18T18:57:02"},
{"Name":"KPI_ERIK'TOTAL.VK","Class":"SERIES","Observed":"AVERAGED","Desc":"Totalindeks_vekt","Created":"2017-01-18T18:28:29","Updated":"2025-02-10T09:25:53"},
{"Name":"KPI_ERIK'TOTAL.YTYPCT","Class":"SERIES","Observed":"AVERAGED","Desc":"Totalindeks_12 mEneders prisendring","Created":"2017-01-18T18:28:29","Updated":"2017-01-18T18:57:02"},
{"Name":"KPI_ERIK'TOTAL_JAE.IPR.G","Class":"SERIES","Observed":"AVERAGED","Desc":"Totalindeks_Trend(prog1)","Created":"2025-02-10T09:25:49","Updated":"2025-02-10T09:25:50"},
{"Name":"KPI_ERIK'TOTAL_JAE.IPR.S","Class":"SERIES","Observed":"AVERAGED","Desc":"Totalindeks_Sesongjustert(prog1)","Created":"2025-02-10T09:25:49","Updated":"2025-02-10T09:25:50"},
{"Name":"KPI_PUBL'K01111_11111.IPR","Class":"SERIES","Observed":"AVERAGED","Desc":"Ris_indeks pris","Created":"2017-01-18T18:28:28","Updated":"2025-05-09T08:23:40"},
{"Name":"KPI_PUBL'K01111_11111.IPR.A","Class":"SERIES","Observed":"AVERAGED","Desc":"Ris_indeks pris_Ersgjsn","Created":"2017-01-18T18:28:28","Updated":"2025-01-10T08:33:25"},
{"Name":"KPI_PUBL'TOTAL.IPR","Class":"SERIES","Observed":"AVERAGED","Desc":"Totalindeks_indeks pris","Created":"2017-01-18T18:28:29","Updated":"2025-05-09T08:23:39"},
{"Name":"KPI_PUBL'TOTAL.IPR.A","Class":"SERIES","Observed":"AVERAGED","Desc":"Totalindeks_indeks pris_Ersgjsn","Created":"2017-01-18T18:28:29","Updated":"2025-01-10T08:33:25"},
{"Name":"KPI_PUBL'TOTAL.IPR.G","Class":"SERIES","Observed":"AVERAGED","Desc":"Totalindeks_Trend(prog1)","Created":"2025-05-09T08:23:40","Updated":"2025-05-09T08:23:42"},
{"Name":"KPI_PUBL'TOTAL.IPR.S","Class":"SERIES","Observed":"AVERAGED","Desc":"Totalindeks_Sesongjustert(prog1)","Created":"2025-05-09T08:23:40","Updated":"2025-05-09T08:23:42"},
{"Name":"KPI_PUBL'TOTAL.PCT","Class":"SERIES","Observed":"AVERAGED","Desc":"Totalindeks_1 mEneds prisendring","Created":"2017-01-18T18:28:29","Updated":"2017-01-18T18:57:02"},
{"Name":"KPI_PUBL'TOTAL.VK","Class":"SERIES","Observed":"AVERAGED","Desc":"Totalindeks_vekt","Created":"2017-01-18T18:28:29","Updated":"2025-05-09T08:23:45"},
{"Name":"KPI_PUBL'TOTAL.YTYPCT","Class":"SERIES","Observed":"AVERAGED","Desc":"Totalindeks_12 mEneders prisendring","Created":"2017-01-18T18:28:29","Updated":"2017-01-18T18:57:02"},
{"Name":"KPI_PUBL'TOTAL_JAE.IPR.G","Class":"SERIES","Observed":"AVERAGED","Desc":"Totalindeks_Trend(prog1)","Created":"2025-05-09T08:23:40","Updated":"2025-05-09T08:23:42"},
{"Name":"KPI_PUBL'TOTAL_JAE.IPR.S","Class":"SERIES","Observed":"AVERAGED","Desc":"Totalindeks_Sesongjustert(prog1)","Created":"2025-05-09T08:23:40","Updated":"2025-05-09T08:23:42"} ]
"Elapsed_time_in_seconds":0.058
  } ]
```

getfame –n with identical databases identical seriesnames different places)



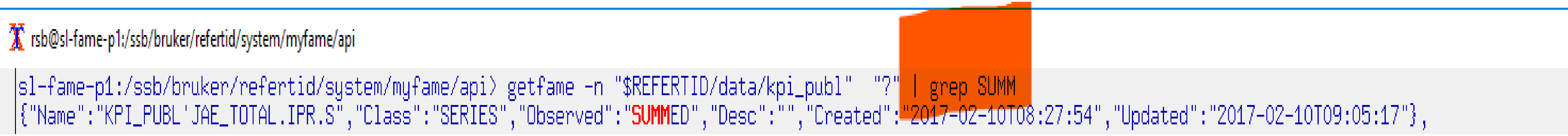
sl-fame-1

```
sl-fame-1:~/MyFame2023/pro/api> $REFERTID/system/myfame/api/getfame -n "testapi, $PWD/../testapi.db" "e?,TEST?"
[{"GetFAME_Json_Api": "Erik.Soberg@ssb.no",
"Version": "Oslo-20250605",
"Executed": "2025-06-04T14:14:00",
"Famever": "11.53",
"Database": "testapi, /ssb/bruker/rsb/MyFame2023/pro/api/../testapi.db",
"Openas": "TESTAPI, TESTAPI2",
"Result": "$HOME/.GetFAME/getfamenames.json",
"Wildcard": "E?,TEST?",
"Found": 9,
"Notfound": 0,
"Missing": "",
"Series":[
{"Name":"TESTAPI'ERIK","Class":"SERIES","Observed":"SUMMED","Freq":"MONTHLY","Desc":"dEScription of erik","Created":"2024-09-09T22:21:26","Updated":"2025-06-02T12:48:59"},
{"Name":"TESTAPI'TEST.ANN","Class":"SERIES","Observed":"SUMMED","Freq":"ANNUAL","Desc":"","Created":"2024-06-16T21:53:09","Updated":"2024-06-17T15:44:56"},
{"Name":"TESTAPI'TEST.MON","Class":"SERIES","Observed":"SUMMED","Freq":"MONTHLY","Desc":"mytest","Created":"2024-06-16T21:54:14","Updated":"2025-06-01T22:39:56"},
{"Name":"TESTAPI'TEST.MON.F","Class":"FORMULA","Observed":"TEST.MON *10","Freq":"NC","Desc":"","Created":"2024-06-16T21:55:16","Updated":"2024-06-16T21:55:53"},
{"Name":"TESTAPI2'ERIK","Class":"SERIES","Observed":"SUMMED","Freq":"MONTHLY","Desc":"secript of erik soeb WOW","Created":"2024-09-09T22:21:26","Updated":"2025-06-01T15:55:47"},
{"Name":"TESTAPI2'EXTRA","Class":"SERIES","Observed":"SUMMED","Freq":"ANNUAL","Desc":"extraextras","Created":"2025-05-30T13:12:53","Updated":"2025-06-01T15:55:47"},
{"Name":"TESTAPI2'TEST.ANN","Class":"SERIES","Observed":"SUMMED","Freq":"ANNUAL","Desc":"","Created":"2024-06-16T21:53:09","Updated":"2025-05-30T11:44:37"},
{"Name":"TESTAPI2'TEST.MON","Class":"SERIES","Observed":"SUMMED","Freq":"MONTHLY","Desc":"","Created":"2024-06-16T21:54:14","Updated":"2024-06-16T22:42:05"},
{"Name":"TESTAPI2'TEST.MON.F","Class":"FORMULA","Observed":"TEST.MON *10","Freq":"NC","Desc":"","Created":"2024-06-16T21:55:16","Updated":"2024-06-16T21:55:53"} ],
"Elapsed_time_in_seconds":0.004
  } ]
```

✓

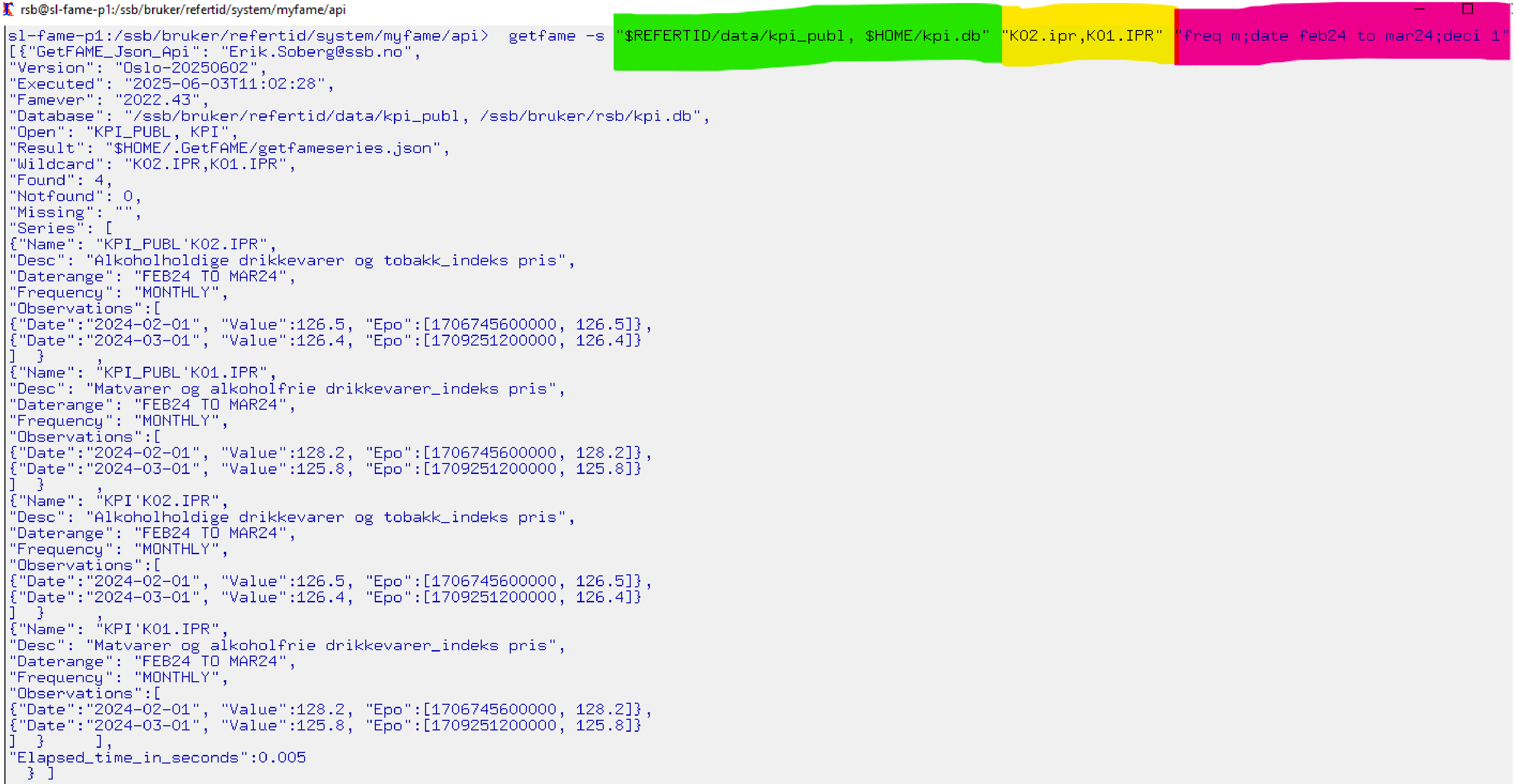
getfame -n $REFERTID/system/myfame/api/**getfamenames**

Combine with linux commands to find descriptions, or series with incorrect definitions
The command below lists all series in the database but only show the one with the text «SUMM»



```
sl-fame-p1:/ssb/bruker/refertid/system/myfame/api> getfame -n "$REFERTID/data/kpi_publ"  "?" | grep SUMM
{"Name":"KPI_PUBL'JAE_TOTAL.IPR.S","Class":"SERIES","Observed":"SUMMED","Desc":"","Created":"2017-02-10T08:27:54","Updated":"2017-02-10T09:05:17"},
```

2. getfame -s $REFERTID/system/myfame/api/**getfameseries**



rsb@sl-fame-p1:/ssb/bruker/refertid/system/myfame/api

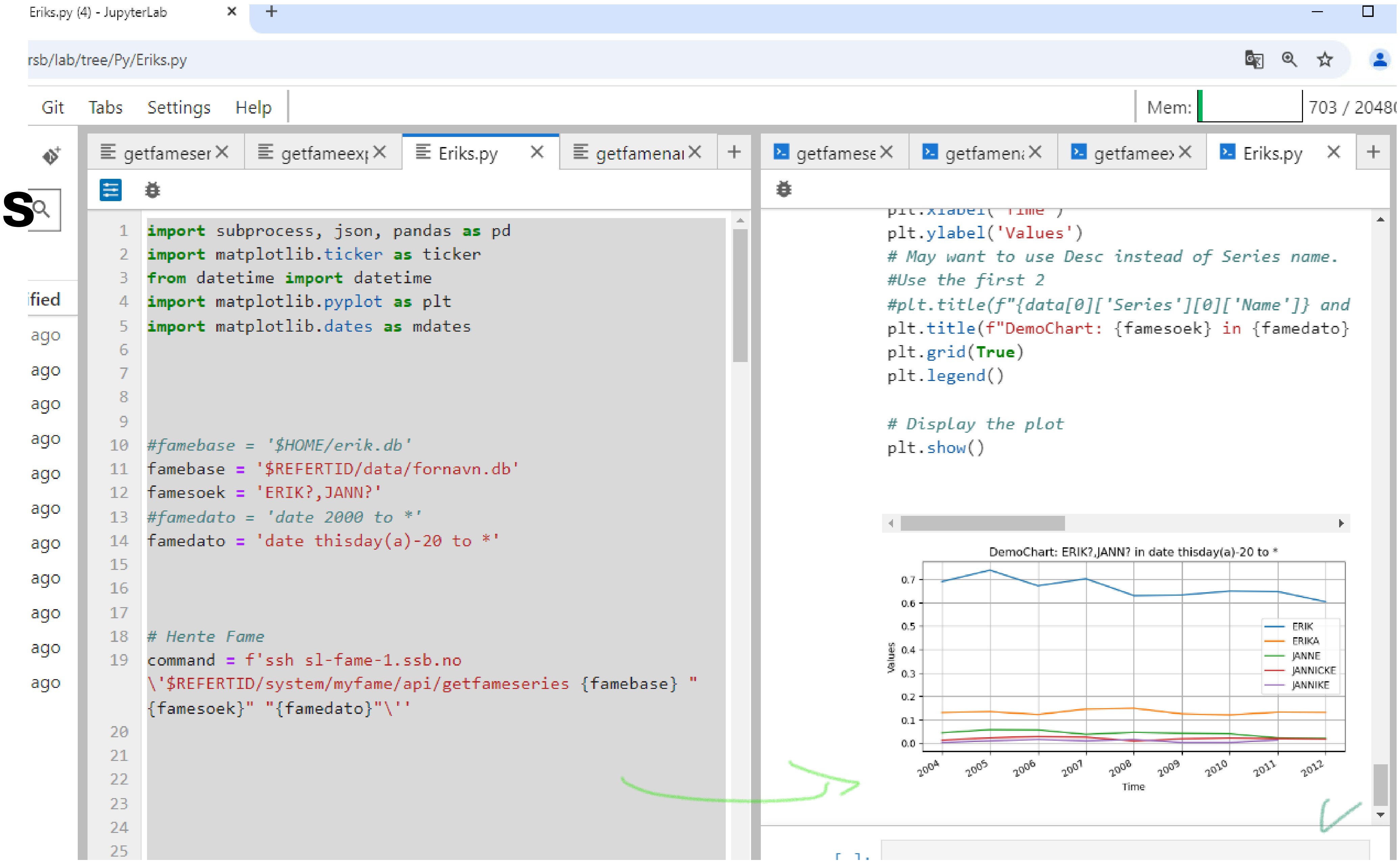
```
sl-fame-p1:/ssb/bruker/refertid/system/myfame/api>  getfame -s "$REFERTID/data/kpi_publ, $HOME/kpi.db" "K02.ipr,K01.IPR" "freq m;date feb24 to mar24;deci 1"
[{"GetFAME_Json_Api": "Erik.Soberg@ssb.no",
"Version": "Oslo-20250602",
"Executed": "2025-06-03T11:02:28",
"Famever": "2022.43",
"Database": "/ssb/bruker/refertid/data/kpi_publ, /ssb/bruker/rsb/kpi.db",
"Open": "KPI_PUBL, KPI",
"Result": "$HOME/.GetFAME/getfameseries.json",
"Wildcard": "K02.IPR,K01.IPR",
"Found": 4,
"Notfound": 0,
"Missing": "",
"Series": [
{"Name": "KPI_PUBL'K02.IPR",
"Desc": "Alkoholholdige drikkevarer og tobakk_indeks pris",
"Daterange": "FEB24 TO MAR24",
"Frequency": "MONTHLY",
"Observations":[
{"Date":"2024-02-01", "Value":126.5, "Epo":[1706745600000, 126.5]},
{"Date":"2024-03-01", "Value":126.4, "Epo":[1709251200000, 126.4]}
]  }         ,
{"Name": "KPI_PUBL'K01.IPR",
"Desc": "Matvarer og alkoholfrie drikkevarer_indeks pris",
"Daterange": "FEB24 TO MAR24",
"Frequency": "MONTHLY",
"Observations":[
{"Date":"2024-02-01", "Value":128.2, "Epo":[1706745600000, 128.2]},
{"Date":"2024-03-01", "Value":125.8, "Epo":[1709251200000, 125.8]}
]  }         ,
{"Name": "KPI'K02.IPR",
"Desc": "Alkoholholdige drikkevarer og tobakk_indeks pris",
"Daterange": "FEB24 TO MAR24",
"Frequency": "MONTHLY",
"Observations":[
{"Date":"2024-02-01", "Value":126.5, "Epo":[1706745600000, 126.5]},
{"Date":"2024-03-01", "Value":126.4, "Epo":[1709251200000, 126.4]}
]  }         ,
{"Name": "KPI'K01.IPR",
"Desc": "Matvarer og alkoholfrie drikkevarer_indeks pris",
"Daterange": "FEB24 TO MAR24",
"Frequency": "MONTHLY",
"Observations":[
{"Date":"2024-02-01", "Value":128.2, "Epo":[1706745600000, 128.2]},
{"Date":"2024-03-01", "Value":125.8, "Epo":[1709251200000, 125.8]}
]  }         ],
"Elapsed_time_in_seconds":0.005
  } ]
```

getfame -s **getfameseries** samples

```
$REFERTID/system/myfame/api/getfameseries  /ssb/bruker/refertid/data/kpi_publ.db  "total.ipr"

getfameseries  /ssb/bruker/refertid/data/kpi_publ.db  "total.ipr, K0?IPR " "date 2024 "

getfameseries  /ssb/bruker/refertid/data/kpi_publ.db "total.ipr"  "freq m; date thisday(m)-5 to *"

getfameseries   $REFERTID/data/fornavn.db  "?ERIK,KRISTIN,JIM?" "date 2010 to 2012 "

getfameseries " /ssb/bruker/refertid/data/fornavn.db"   "?JAN?" "date 2000 to 2005 "

getfameseries " /ssb/bruker/refertid/data/fornavn.db"   "JI? "  "date 2000 to *; deci 1 "

getfameseries  "fornavn.db, name.db"  "JI? ,MATT?"  "date 2000 to * ; deci 2 "

getfameseries  "pi1.db, cpi2.db,cpi_form.db"  "Total.ipr"  "date 2025 ; deci 2"
```

getfame -s

From jupyter with py:



```
import subprocess, json, pandas as pd
import matplotlib.ticker as ticker
from datetime import datetime
import matplotlib.pyplot as plt
import matplotlib.dates as mdates




#famebase = '$HOME/erik.db'
famebase = '$REFERTID/data/fornavn.db'
famesoek = 'ERIK?,JANN?'
#famedato = 'date 2000 to *'
famedato = 'date thisday(a)-20 to *'



# Hente Fame
command = f'ssh sl-fame-1.ssb.no \'$REFERTID/system/myfame/api/getfameseries {famebase} "{famesoek}" "{famedato}"\''
```

```
plt.xlabel('Time')
plt.ylabel('Values')
# May want to use Desc instead of Series name.
#Use the first 2
#plt.title(f"{data[0]['Series'][0]['Name']} and
plt.title(f"DemoChart: {famesoek} in {famedato}
plt.grid(True)
plt.legend()

# Display the plot
plt.show()
```

3. getfame -e $REFERTID/system/myfame/api/getfameexpr

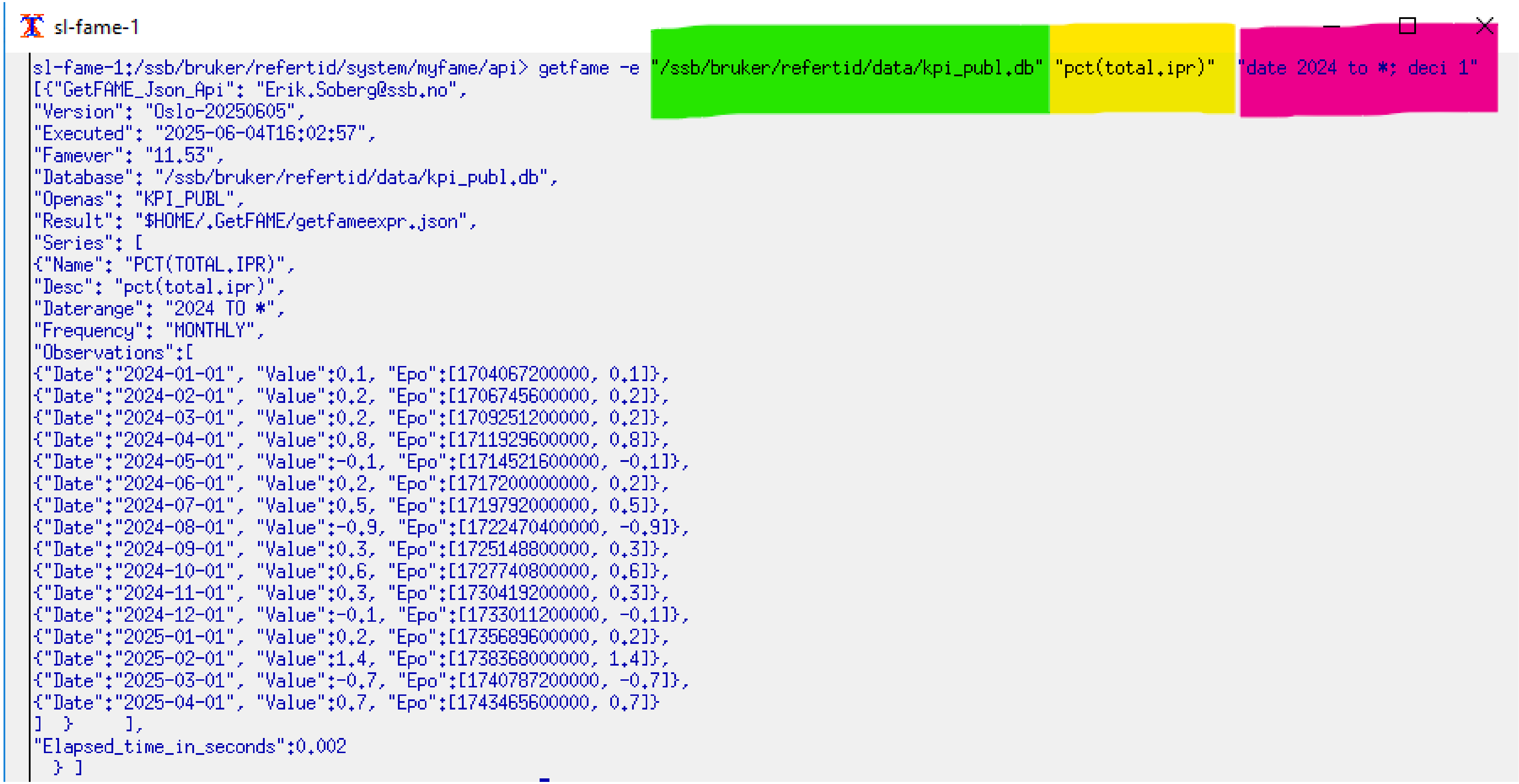
advanced mode

- Data-observations, from FAME database(s) given a fame-**expression:**

```
getfame -e  "$REFERTID/data/fornavn.db "  "mave(ERIK,2)" "date 2000 to 2010"
getfame -e  "$REFERTID/data/fornavn.db "  "Lsum(ERIK,EIRIK)"  "date 2000 to *"
getfame -e  "$REFERTID/data/fornavn.db "  "ERIK+EIRIK"  "date 2000 to *"
getfame –e "$REFERTID/data/kpi_publ.db, mycpi.db " "convert(total.ipr,annual,constant) "  "date *; deci 1 "
getfame -e  "$REFERTID/data/kpi_publ.db, mycpi.db "  "PCT(mycpi’K09.IPR)"  "date 2025; deci 1 "
getfame -e   "cpi1.db,cpi2.db,cpi_form.db" "cpi1’Total.ipr"  "date 2025 ; deci 2"
```

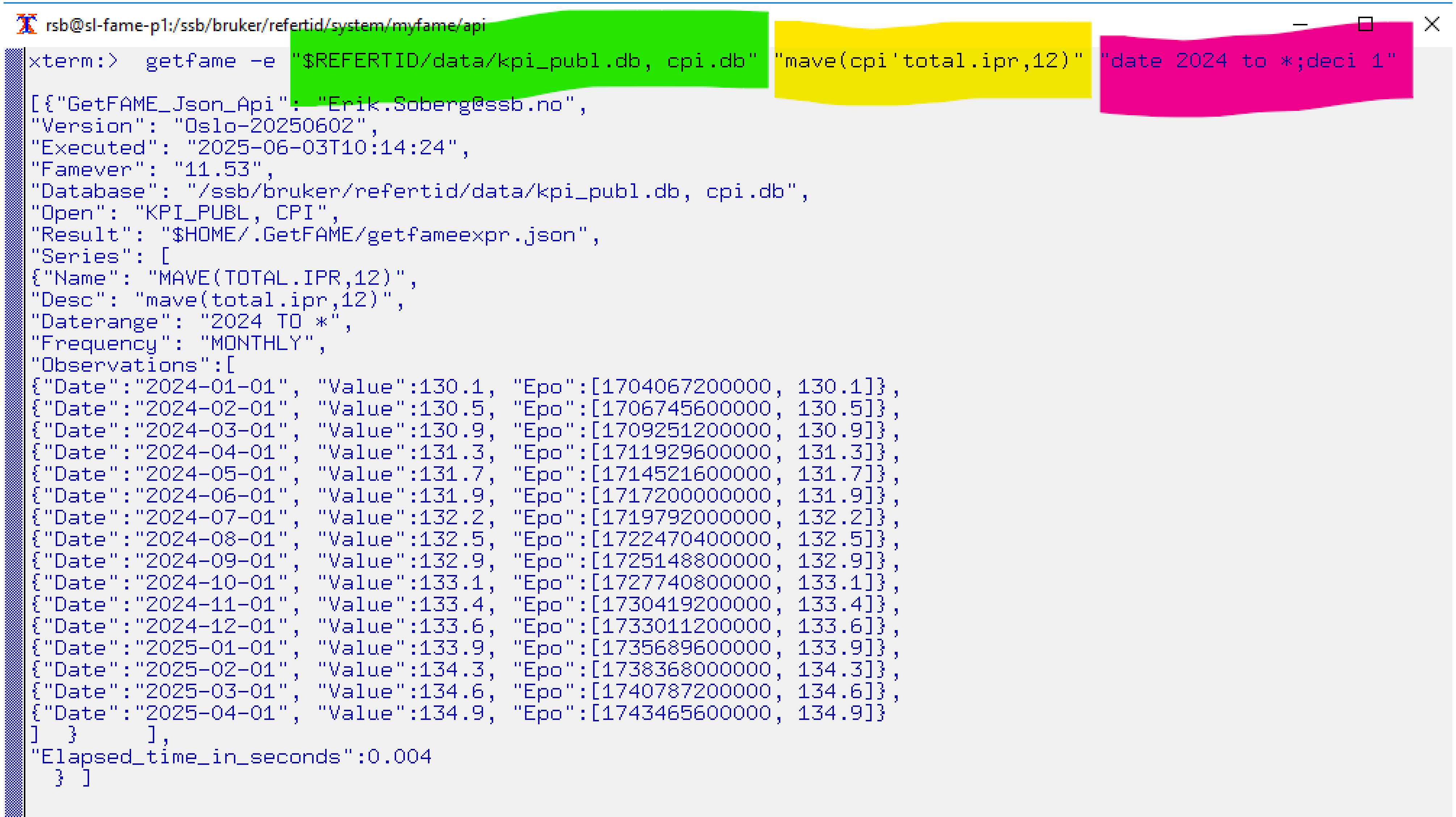
Be aware to **double quote arguments** when they contain special char like : , (‘ ;

getfame –e gets a fame-expression



```
sl-fame-1:/ssb/bruker/refertid/system/myfame/api> getfame -e "/ssb/bruker/refertid/data/kpi_publ.db" "pct(total.ipr)" "date 2024 to *; deci 1"
[{"GetFAME_Json_Api": "Erik.Soberg@ssb.no",
"Version": "Oslo-20250605",
"Executed": "2025-06-04T16:02:57",
"Famever": "11.53",
"Database": "/ssb/bruker/refertid/data/kpi_publ.db",
"Openas": "KPI_PUBL",
"Result": "$HOME/.GetFAME/getfameexpr.json",
"Series": [
{"Name": "PCT(TOTAL.IPR)",
"Desc": "pct(total.ipr)",
"Daterange": "2024 TO *",
"Frequency": "MONTHLY",
"Observations":[
{"Date":"2024-01-01", "Value":0.1, "Epo":[1704067200000, 0.1]},
{"Date":"2024-02-01", "Value":0.2, "Epo":[1706745600000, 0.2]},
{"Date":"2024-03-01", "Value":0.2, "Epo":[1709251200000, 0.2]},
{"Date":"2024-04-01", "Value":0.8, "Epo":[1711929600000, 0.8]},
{"Date":"2024-05-01", "Value":-0.1, "Epo":[1714521600000, -0.1]},
{"Date":"2024-06-01", "Value":0.2, "Epo":[1717200000000, 0.2]},
{"Date":"2024-07-01", "Value":0.5, "Epo":[1719792000000, 0.5]},
{"Date":"2024-08-01", "Value":-0.9, "Epo":[1722470400000, -0.9]},
{"Date":"2024-09-01", "Value":0.3, "Epo":[1725148800000, 0.3]},
{"Date":"2024-10-01", "Value":0.6, "Epo":[1727740800000, 0.6]},
{"Date":"2024-11-01", "Value":0.3, "Epo":[1730419200000, 0.3]},
{"Date":"2024-12-01", "Value":-0.1, "Epo":[1733011200000, -0.1]},
{"Date":"2025-01-01", "Value":0.2, "Epo":[1735689600000, 0.2]},
{"Date":"2025-02-01", "Value":1.4, "Epo":[1738368000000, 1.4]},
{"Date":"2025-03-01", "Value":-0.7, "Epo":[1740787200000, -0.7]},
{"Date":"2025-04-01", "Value":0.7, "Epo":[1743465600000, 0.7]}
]  }     ],
"Elapsed_time_in_seconds":0.002
  } ]
```

getfame –e with several databases in case u have formulas elsewhere

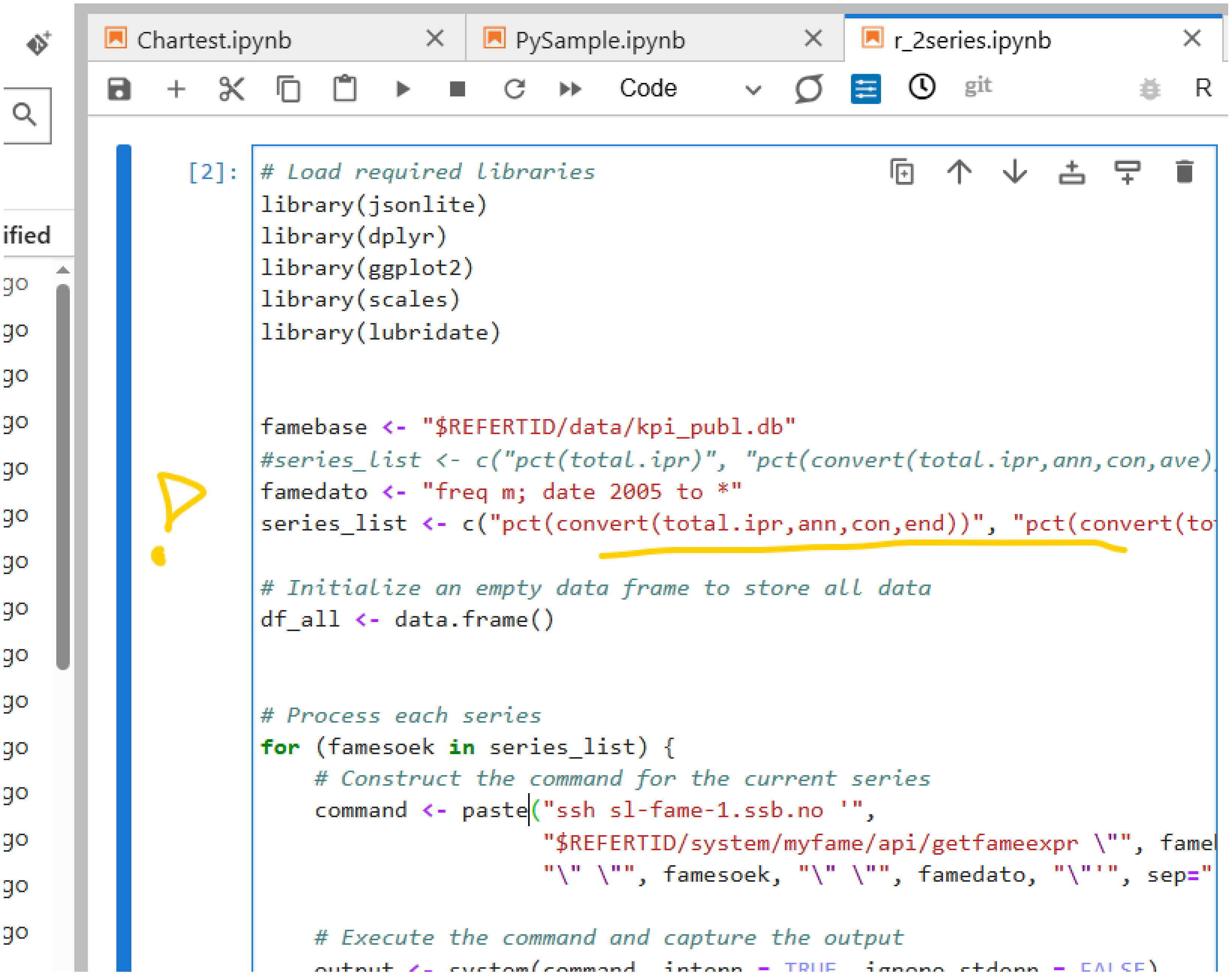


```
xterm:>  getfame -e "$REFERTID/data/kpi_publ.db, cpi.db" "mave(cpi'total.ipr,12)" "date 2024 to *;deci 1"

[{"GetFAME_Json_Api": "Erik.Soberg@ssb.no",
"Version": "Oslo-20250602",
"Executed": "2025-06-03T10:14:24",
"Famever": "11.53",
"Database": "/ssb/bruker/refertid/data/kpi_publ.db, cpi.db",
"Open": "KPI_PUBL, CPI",
"Result": "$HOME/.GetFAME/getfameexpr.json",
"Series": [
{"Name": "MAVE(TOTAL.IPR,12)",
"Desc": "mave(total.ipr,12)",
"Daterange": "2024 TO *",
"Frequency": "MONTHLY",
"Observations":[
{"Date":"2024-01-01", "Value":130.1, "Epo":[1704067200000, 130.1]},
{"Date":"2024-02-01", "Value":130.5, "Epo":[1706745600000, 130.5]},
{"Date":"2024-03-01", "Value":130.9, "Epo":[1709251200000, 130.9]},
{"Date":"2024-04-01", "Value":131.3, "Epo":[1711929600000, 131.3]},
{"Date":"2024-05-01", "Value":131.7, "Epo":[1714521600000, 131.7]},
{"Date":"2024-06-01", "Value":131.9, "Epo":[1717200000000, 131.9]},
{"Date":"2024-07-01", "Value":132.2, "Epo":[1719792000000, 132.2]},
{"Date":"2024-08-01", "Value":132.5, "Epo":[1722470400000, 132.5]},
{"Date":"2024-09-01", "Value":132.9, "Epo":[1725148800000, 132.9]},
{"Date":"2024-10-01", "Value":133.1, "Epo":[1727740800000, 133.1]},
{"Date":"2024-11-01", "Value":133.4, "Epo":[1730419200000, 133.4]},
{"Date":"2024-12-01", "Value":133.6, "Epo":[1733011200000, 133.6]},
{"Date":"2025-01-01", "Value":133.9, "Epo":[1735689600000, 133.9]},
{"Date":"2025-02-01", "Value":134.3, "Epo":[1738368000000, 134.3]},
{"Date":"2025-03-01", "Value":134.6, "Epo":[1740787200000, 134.6]},
{"Date":"2025-04-01", "Value":134.9, "Epo":[1743465600000, 134.9]}
]  }      ],
"Elapsed_time_in_seconds":0.004
  } ]
```

Using the power of FAME by

getfame –e

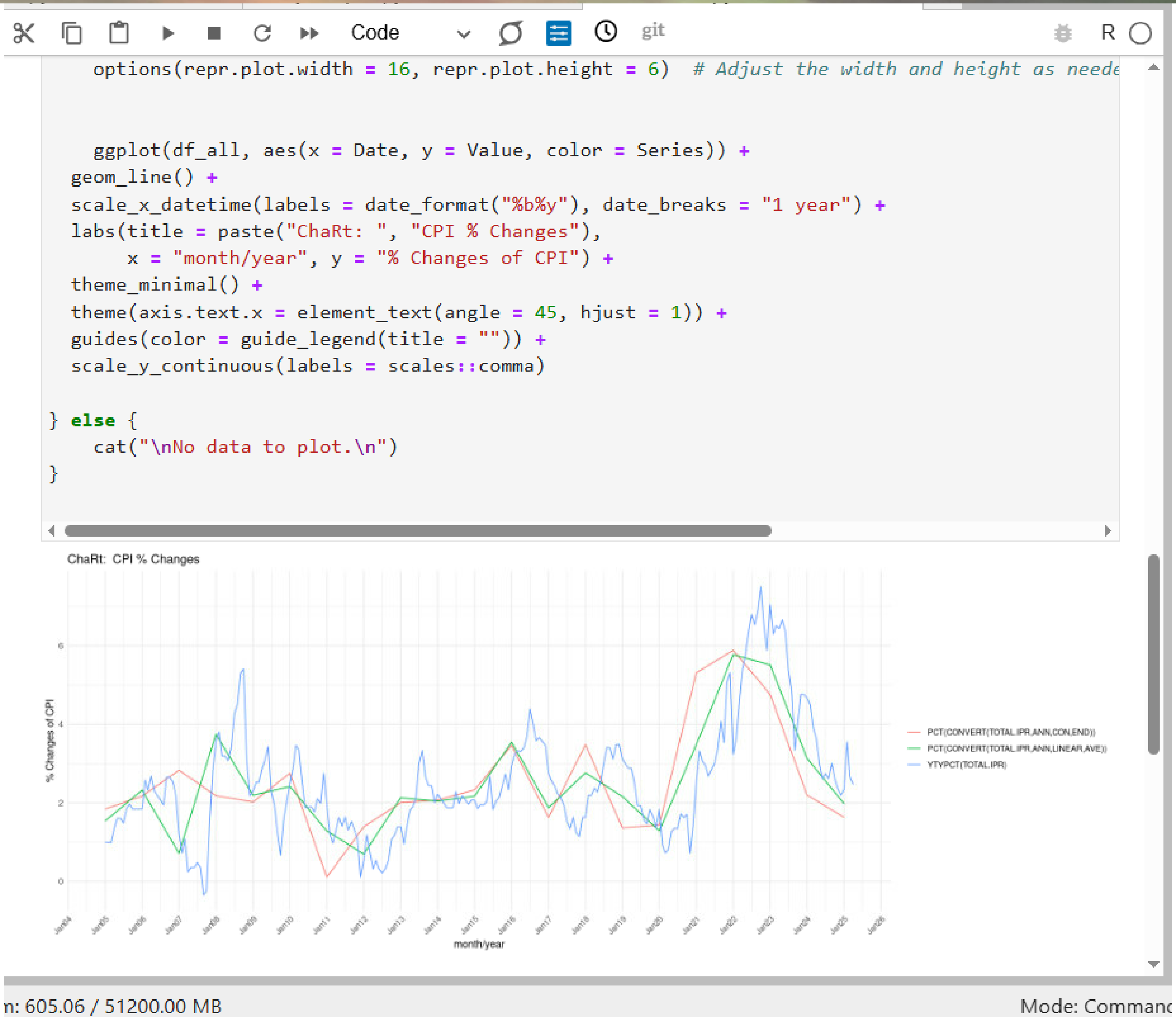
with R from Jupiterlab



```r
# Load required libraries
library(jsonlite)
library(dplyr)
library(ggplot2)
library(scales)
library(lubridate)


famebase <- "$REFERTID/data/kpi_publ.db"
#series_list <- c("pct(total.ipr)", "pct(convert(total.ipr,ann,con,ave)
famedato <- "freq m; date 2005 to *"
series_list <- c("pct(convert(total.ipr,ann,con,end))", "pct(convert(to

# Initialize an empty data frame to store all data
df_all <- data.frame()


# Process each series
for (famesoek in series_list) {
    # Construct the command for the current series
    command <- paste("ssh sl-fame-1.ssb.no '",
                     "$REFERTID/system/myfame/api/getfameexpr \"", fame
                     "\" \"", famesoek, "\" \"", famedato, "\"'", sep="

    # Execute the command and capture the output
    output <- system(command, intern = TRUE, ignore.stderr = FALSE)
```

getfame -e

with R
from
Jupiterlab



```
  options(repr.plot.width = 16, repr.plot.height = 6)  # Adjust the width and height as neede


  ggplot(df_all, aes(x = Date, y = Value, color = Series)) +
 geom_line() +
 scale_x_datetime(labels = date_format("%b%y"), date_breaks = "1 year") +
 labs(title = paste("ChaRt: ", "CPI % Changes"),
      x = "month/year", y = "% Changes of CPI") +
 theme_minimal() +
 theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
 guides(color = guide_legend(title = "")) +
 scale_y_continuous(labels = scales::comma)

} else {
  cat("\nNo data to plot.\n")
}
```

```
n: 605.06 / 51200.00 MB
```

Samples (shows help info, when no arguments passed)

getfamenames (gets series names & metadata from databases with a list of wildcards
getfame –n

getfameseries (gets observations from >= 1 series in database(s) given a list of wildcards)
getfame –s

getfameexpr (gets observations given 1 FAME expression)
getfame -e

For complete **jupyterlab** samples, see Github

Summary

- The **getfame –e** option use the full power of FAME and can evaluate formulas, functions, conversions among various series, formulas,frequiencies and databases

- To get more series with **getfame –e** simply loop by expression and add to same charts or dataset.

- **getfame –n** is powerful when combining ***grep | more |head*** to search for series/formulas names or metadata