DOD-M982

```r
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
# contracts from usaspending.gov
usaspending_m982 <- read_csv("Contracts_PrimeTransactions_2025-02-22_H19M41S38_1.csv")
```

```
Rows: 306 Columns: 297
-- Column specification --------------------------------------------------------
Delimiter: ","
chr  (142): contract_transaction_unique_key, contract_award_unique_key, awar...
dbl   (27): transaction_number, parent_award_modification_number, federal_ac...
lgl  (121): parent_award_agency_id, parent_award_agency_name, parent_award_i...
dttm   (1): period_of_performance_potential_end_date
date   (6): action_date, period_of_performance_start_date, period_of_perform...

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
# contracts from fpds.gov
fpds_m982 <- read_csv("M982_fpds.csv")
```

```
New names:
Rows: 228 Columns: 27
-- Column specification
-------------------------------------------------------- Delimiter: "," chr
(21): Contract ID, Reference IDV, Modification Number, Award/IDV Type, A... dbl
(5): Transaction Number, Contracting Agency ID, NAICS, Entity ZIP Code,... lgl
(1): ...27
i Use `spec()` to retrieve the full column specification for this data. i
Specify the column types or set `show_col_types = FALSE` to quiet this message.
* `` -> `...27`
```

```r
fpds_m982 <- fpds_m982 |>
  mutate(`Date Signed` = as_date(format(parse_date(`Date Signed`, format = "%b %d, %Y"), "%Y
         `Action Obligation ($)` = str_replace_all(`Action Obligation ($)`, "\\$|,", "") |>
           as.numeric())
```

```r
# for use with fpds.gov data, which does not come with fiscal year column
calculate_fiscal_year <- function(date) {
  fiscal_year <- if_else(
    month(date) >= 10,
    year(date) + 1,
    year(date)
  )
  return(fiscal_year)
}
```

```r
# manually entered data from FY2013 Selected Acquisition Reports (SAR)
# uses "End Item Recurring Flyaway TY $M" since this covers M982 production and excludes sta
sar_m982 <- data.frame(
  fiscal_year = seq(2005, 2016),
  quantity_procured = c(127, 321, 793, 400, 435, 900, 100, 744, 840, 929, 416, 472),
  end_item_recurring_flyaway_TY_M = c(35.1, 48.3, 84.5, 47.5, 57.9, 103.2, 30.5, 56.1, 65.9,
) |>
  mutate(unit_cost_TY_M = end_item_recurring_flyaway_TY_M / quantity_procured,
         CPI_inflator_2024 = c(1.61, 1.56, 1.52, 1.46, 1.47, 1.44, 1.4, 1.37, 1.35, 1.33, 1.
)
```

```r
# manually entered data from DACIS budget report
dacis_m982 <- data.frame(
  fiscal_year = seq(2017, 2024),
  quantity_procured = c(9641 - sum(sar_m982$quantity_procured), 2841, 2208, 804, 737, 226, 2
  end_item_recurring_flyaway_TY_M = c(927.578 - sum(sar_m982$end_item_recurring_flyaway_TY_M
) |>
  mutate(unit_cost_TY_M = end_item_recurring_flyaway_TY_M / quantity_procured,
         CPI_inflator_2024 = c(1.28, 1.25, 1.23, 1.21, 1.16, 1.07, 1.03, 1)
)
# data for FY2017 is not listed in either report; SAR has FY2005 (production start) through
# thus data for FY2017 is estimated by subtracting summed values in the SAR data from the pre
```

```r
# merging this data
pre_2019_m982 <- rbind(sar_m982, dacis_m982) |>
  mutate(year = fiscal_year,
         total_expenditure = end_item_recurring_flyaway_TY_M * 1000000,
         unit_price = unit_cost_TY_M * 1000000) |>
  select(year, total_expenditure, unit_price, quantity_procured, CPI_inflator_2024)
pre_2019_m982
```

```
   year total_expenditure unit_price quantity_procured CPI_inflator_2024
1  2005          35100000  276377.95               127              1.61
2  2006          48300000  150467.29               321              1.56
3  2007          84500000  106557.38               793              1.52
4  2008          47500000  118750.00               400              1.46
5  2009          57900000  133103.45               435              1.47
6  2010         103200000  114666.67               900              1.44
7  2011          30500000  305000.00               100              1.40
8  2012          56100000   75403.23               744              1.37
9  2013          65900000   78452.38               840              1.35
10 2014          75800000   81593.11               929              1.33
11 2015          34600000   83173.08               416              1.33
12 2016          44700000   94703.39               472              1.31
13 2017         243478000   76952.59              3164              1.28
14 2018         220234000   77519.89              2841              1.25
15 2019         173891000   78754.98              2208              1.23
16 2020          93486000  116276.12               804              1.21
17 2021          86282000  117071.91               737              1.16
18 2022          41068000  181716.81               226              1.07
19 2023          43000000  182203.39               236              1.03
20 2024          43000000  185344.83               232              1.00
```

```r
# DACIS source gives $96,211.80 for pre-2018 unit cost
# based on below calculation, SAR source and my 2017 estimation align with DACIS source; con
sum(pre_2019_m982$total_expenditure[1:14]) / sum(pre_2019_m982$quantity_procured[1:14])
```

```
[1] 91957.38
```

```
# DACIS source is dated March 2019, additional procurement contracts have been issued since
# https://thedefensepost.com/2022/02/03/us-army-raytheon-excalibur-munition/ : early 2022, $6
# https://thedefensepost.com/2022/12/20/us-army-raytheon-excalibur/ : late 2022, $84 million

# data from post March 2019 contracts from usaspending.gov and fpds.gov

# usaspending.gov data for post-March 2019 contracts since DACIS source is dated March 2019
post_2018_usaspending_m982 <- usaspending_m982 |>
  select(3, 10, 22:25, 51, 96:97, 46, 87, 95, 99, 105, 111, 117, 125, 295:297) |>
  filter(federal_action_obligation != 0,
         foreign_funding_description != "FOREIGN FUNDS FMS",
         place_of_manufacture != "NOT A MANUFACTURED END PRODUCT") |>
  arrange(desc(federal_action_obligation)) |>
  # calculations below
  group_by("year" = action_date_fiscal_year) |>
  summarize("total_expenditure" = sum(federal_action_obligation)) |>
  mutate("unit_price" = pre_2019_m982$unit_price[4:19],
         "quantity_procured" = floor(total_expenditure / unit_price)) |>
  filter(year >= 2019)

# fpds.gov data for post-March 2019 contracts since DACIS source is dated March 2019
post_2018_fpds_m982 <- fpds_m982 |>
  filter(`Action Obligation ($)` != 0) |>
  mutate(`Fiscal Year` = calculate_fiscal_year(`Date Signed`)) |>
  arrange(`Date Signed`) |>
  # calculations below
  group_by("year" = `Fiscal Year`) |>
  summarize("total_expenditure" = sum(`Action Obligation ($)`)) |>
  mutate("unit_price" = pre_2019_m982$unit_price,
         "quantity_procured" = floor(total_expenditure / unit_price)) |>
  filter(year >= 2019)

# fpds.gov data potentially includes non-related contracts (thus higher values) because fpds
post_2018_usaspending_m982
```

```
# A tibble: 5 x 4
   year total_expenditure unit_price quantity_procured
  <dbl>             <dbl>      <dbl>             <dbl>
1  2019        200234186.     78755.              2542
2  2020         95511150.    116276.               821
```

```
3  2021         84996062.    117072.               726
4  2022         66221560.    181717.               364
5  2023        568170438.    182203.              3118
```

```
post_2018_fpds_m982
```

```
# A tibble: 6 x 4
   year total_expenditure unit_price quantity_procured
  <dbl>             <dbl>      <dbl>             <dbl>
1  2019        200234192.     78755.              2542
2  2020        100011284.    116276.               860
3  2021        117876055.    117072.              1006
4  2022         75778750.    181717.               417
5  2023        568170438.    182203.              3118
6  2024         10648417     185345.                57
```

```
# numbers roughly align with SAR/DACIS merged data, except for FY2023 which is when the conti
# usaspending.gov and fpds.gov data perfectly align for FY2023 likely representing said new
m982_clean <- pre_2019_m982
m982_clean$quantity_procured[19] = 3118
m982_clean$total_expenditure[19] = 568170438
m982_clean
```

```
   year total_expenditure unit_price quantity_procured CPI_inflator_2024
1  2005          35100000  276377.95               127              1.61
2  2006          48300000  150467.29               321              1.56
3  2007          84500000  106557.38               793              1.52
4  2008          47500000  118750.00               400              1.46
5  2009          57900000  133103.45               435              1.47
6  2010         103200000  114666.67               900              1.44
7  2011          30500000  305000.00               100              1.40
8  2012          56100000   75403.23               744              1.37
9  2013          65900000   78452.38               840              1.35
10 2014          75800000   81593.11               929              1.33
11 2015          34600000   83173.08               416              1.33
12 2016          44700000   94703.39               472              1.31
13 2017         243478000   76952.59              3164              1.28
14 2018         220234000   77519.89              2841              1.25
15 2019         173891000   78754.98              2208              1.23
16 2020          93486000  116276.12               804              1.21
17 2021          86282000  117071.91               737              1.16
```

```
18 2022          41068000  181716.81               226               1.07
19 2023         568170438  182203.39              3118               1.03
20 2024          43000000  185344.83               232               1.00
```

```
# depreciation calculations
m982_production <- m982_clean |>
  rename(num_m982_produced = quantity_procured) |>
  select(1, 5, 2:4) |>
  mutate(num_delivered_from_year = c(num_m982_produced[1:12], 7000 - sum(num_m982_produced[1
         unit_price_inflation_adjusted = CPI_inflator_2024 * unit_price,
         unit_depreciated_value = pmin(unit_price_inflation_adjusted - (2022 - year) * (unit_
         non_inflation_adjusted_depreciated_value = pmin(unit_price - (2022 - year) * (unit_p

m982_production
```

```
   year CPI_inflator_2024 total_expenditure unit_price num_m982_produced
1  2005              1.61          35100000  276377.95               127
2  2006              1.56          48300000  150467.29               321
3  2007              1.52          84500000  106557.38               793
4  2008              1.46          47500000  118750.00               400
5  2009              1.47          57900000  133103.45               435
6  2010              1.44         103200000  114666.67               900
7  2011              1.40          30500000  305000.00               100
8  2012              1.37          56100000   75403.23               744
9  2013              1.35          65900000   78452.38               840
10 2014              1.33          75800000   81593.11               929
11 2015              1.33          34600000   83173.08               416
12 2016              1.31          44700000   94703.39               472
13 2017              1.28         243478000   76952.59              3164
14 2018              1.25         220234000   77519.89              2841
15 2019              1.23         173891000   78754.98              2208
16 2020              1.21          93486000  116276.12               804
17 2021              1.16          86282000  117071.91               737
18 2022              1.07          41068000  181716.81               226
19 2023              1.03         568170438  182203.39              3118
20 2024              1.00          43000000  185344.83               232
   num_delivered_from_year unit_price_inflation_adjusted unit_depreciated_value
1                      127                     444968.50               66745.28
2                      321                     234728.97               46945.79
3                      793                     161967.21               40491.80
4                      400                     173375.00               52012.50
5                      435                     195662.07               68481.72
```

```
6                      900                  165120.00              66048.00
7                      100                  427000.00             192150.00
8                      744                  103302.42              51651.21
9                      840                  105910.71              58250.89
10                     929                  108518.84              65111.30
11                     416                  110620.19              71903.13
12                     472                  124061.44              86843.01
13                     523                   98499.32              73874.49
14                       0                   96899.86              77519.89
15                       0                   96868.63              82338.33
16                       0                  140694.10             126624.69
17                       0                  135803.42             129013.25
18                       0                  194436.99             194436.99
19                       0                  187669.49             187669.49
20                       0                  185344.83             185344.83
   non_inflation_adjusted_depreciated_value
1                                  41456.69
2                                  30093.46
3                                  26639.34
4                                  35625.00
5                                  46586.21
6                                  45866.67
7                                 137250.00
8                                  37701.61
9                                  43148.81
10                                 48955.87
11                                 54062.50
12                                 66292.37
13                                 57714.44
14                                 62015.91
15                                 66941.73
16                                104648.51
17                                111218.32
18                                181716.81
19                                182203.39
20                                185344.83
```

```r
sum(m982_production$num_m982_produced[1:12])
```

```
[1] 6477
```

```r
sum(m982_production$num_delivered_from_year)
```

```
[1] 7000
```

```r
sum(m982_production$num_delivered_from_year * m982_production$unit_depreciated_value)
```

```
[1] 442294607
```

```r
sum(m982_production$num_delivered_from_year * m982_production$unit_price_inflation_adjusted)
```

```
[1] 1028796143
```

```r
cat("Number of units produced:",
    formatC(
         sum(m982_production$num_m982_produced),
         format = "f", big.mark = ",", digits = 0
       ),
    "\n")
```

```
Number of units produced: 19,807
```

```r
cat("Number of units delivered to Ukraine:",
    formatC(
         sum(m982_production$num_delivered_from_year),
         format = "f", big.mark = ",", digits = 0
       ),
    "\n")
```

```
Number of units delivered to Ukraine: 7,000
```

```r
cat(paste0("Total depreciated value delivered to Ukraine (inflation adjusted): $",
       formatC(
         round(sum(m982_production$num_delivered_from_year * m982_production$unit_depreciated
         format = "f", big.mark=",", digits = 2
       )),
    "\n")
```

```
Total depreciated value delivered to Ukraine (inflation adjusted): $442,294,607.00
```

```r
cat(paste0("Total original value delivered to Ukraine (inflation adjusted): $",
       formatC(
         round(sum(m982_production$num_delivered_from_year * m982_production$unit_price_infla
         format = "f", big.mark = ",", digits = 2
       )),
    "\n")
```

```
Total original value delivered to Ukraine (inflation adjusted): $1,028,796,143.00
```

```r
cat(paste0("Total depreciated value delivered to Ukraine (non-adjusted): $",
       formatC(
         round(sum(m982_production$num_delivered_from_year * m982_production$non_inflation_ac
         format = "f", big.mark=",", digits = 2
       )),
    "\n")
```

```
Total depreciated value delivered to Ukraine (non-adjusted): $319,309,654.00
```