

Analysis of UCMR5 data to identify populations potentially impacted by rollbacks of PFAS regulations

Southern Environmental Law Center – July 2026

Libbie Weimer, GISP – lweimer@selc.org

About this Analysis

The statistics in this analysis were calculated using monitoring results from the January 2026 UCMR5 data release.¹ These data compile nearly 65,000 sampling events reporting levels of 29 different PFAS. They do not cover all public water systems nationwide, with priority given to utilities serving over 3,300 people. For more detailed information about UCMR5, please refer to [EPA's UCMR5 documentation](#).

To estimate the population potentially impacted by rollbacks to EPA's current PFAS Maximum Contaminant Limits (MCLs), we took the UCMR5 data and identified which public utilities had *any single* sampling result above *any* of the enforceable levels for the six PFAS chemicals with standards. For example, if a utility tested five groundwater wells and found GenX above 10 ppt in any of those wells, that utility counts towards the estimate of potentially impacted population.

Utilities identified in this analysis are *not* proven to be in violation of the current PFAS MCLs, since compliance is determined by running annual averages, including data that has not yet been released.

Current PFAS Standards

Chemical	MCL (enforceable levels) ²
PFOA	4 parts per trillion (ppt)
PFOS	4 ppt
GenX (HPFO-DA)	10 ppt
PFHxS	10 ppt
PFNA	10 ppt

¹ U.S. Environmental Protection Agency, Fifth Unregulated Contaminant Monitoring Rule, January 2026 release, downloaded 17 February 2026, <https://www.epa.gov/dwucmr/occurrence-data-unregulated-contaminant-monitoring-rule#5>.

² U.S. Environmental Protection Agency, Safe Drinking Water Act, Per- and Polyfluoroalkyl Substances (PFAS) Final PFAS National Primary Drinking Water Regulation, 2024, accessed 18 May 2026, <https://www.epa.gov/sdwa/and-polyfluoroalkyl-substances-pfas>.

Mixtures containing two or more of GenX, PFHxS, PFNA, and PFBS	1 (unitless) Hazard Index
---	------------------------------

Population totals were calculated by linking the public water systems in UCMR5 to data collected by EPA’s Safe Drinking Water Information System (SDWIS) via EPA's ECHO Safe Drinking Water API.³ The API was queried on 20 March 2026, yielding an estimated population served by each utility. In this analysis, all population percents are calculated out of the total population of all drinking water utilities tested under UCMR5.

Data Sources

PFAS Monitoring – UCMR5 January 2026 release: U.S. Environmental Protection Agency, Fifth Unregulated Contaminant Monitoring Rule, January 2026 release, downloaded 17 February 2026, <https://www.epa.gov/dwucmr/occurrence-data-unregulated-contaminant-monitoring-rule#5>.

Utility Characteristics – SDWA via ECHO’s API: U.S. Environmental Protection Agency, Safe Drinking Water Information System, accessed via the Enforcement and Compliance History Online Safe Drinking Water Act API, queried 20 March 2026, <https://echo.epa.gov/tools/web-services/facility-search-drinking-water>.

Results

The following tables present summary results. Population and number of utilities are rounded to 3 significant digits. Percentages are rounded to 2 significant digits. Please see the attached dataset for the full results of the analysis.

Summary of Impacted Public Water Systems by Regulatory Standard

Across the monitoring results available in UCMR5, over 80 million people are served by a public water system where one or more PFAS were found above the current enforceable limits. These systems are present in urban, rural and suburban communities across the United States, including 48 states and four U.S. territories.

To date, UCMR5 has released monitoring results for public water systems serving around 300 million people. For those whose water has been tested, 1 out of 4 are served by a system where PFAS above the current enforceable limits was reported.

³ U.S. Environmental Protection Agency, Safe Drinking Water Information System, accessed via the Enforcement and Compliance History Online Safe Drinking Water Act API, queried 20 March 2026, <https://echo.epa.gov/tools/web-services/facility-search-drinking-water>.

	Public Water Systems		Population Served	
	# PWS	% PWS	Population	% Population
Total Tested	10,300	-	302 million	-
Above any PFAS enforceable limit	1,720	17%	80.1 million	26%
Above PFOA or PFOS limit	1,690	16%	79.8 million	26%
Above any Hazard Index limit	225	2.2%	14.9 million	4.9%

Detections of Legacy PFAS

Legacy PFAS (PFOA and PFOS) are commonly present above current regulatory standards, impacting nearly 80 million people. These contaminants were found in some public water sources at levels in the hundreds of ppt.

Chemical	PFOA	PFOS
95th %ile*	17.1	25.8
Max Value	235	490

*Note: these are the 95th %ile of samples *where the analyte was detected*. If you include non-detects the values drop to 4.4 for PFOS and 4.5 for PFOA.

Crossover Detections of Legacy and Hazard Index PFAS

Legacy PFAS (PFOA and PFOS) were detected at 1,710 public water systems. Of those systems, 1,270, or 74% *also* had detectible hazard index PFAS (HFPO-DA, PFHxS, PFNA or PFBS). Hazard index PFAS were detected at 2,020 public water systems. Of those systems, again, 1,271 or 63% *also* had detections of legacy PFAS.

Data Dictionary

Column Name	Description	Source
PWSID	Unique identifier for Public Water System	SDWA
PWSName	Name of Public Water System	SDWA
PFOS	Maximum value of PFOS for all water sources and all monitoring dates w/in UCMR5	UCMR5
PFOA	Maximum value of PFOA for all water sources and all monitoring dates w/in UCMR5	UCMR5
PFBS	Maximum value of PFBS for all water sources and all monitoring dates w/in UCMR5	UCMR5
PFNA	Maximum value of PFNA for all water sources and all monitoring dates w/in UCMR5	UCMR5
HFPO-DA	Maximum value of HFPO-DA for all water sources and all monitoring dates w/in UCMR5	UCMR5
PFHxS	Maximum value of PFHxS for all water sources and all monitoring dates w/in UCMR5	UCMR5

Hazard_index	Maximum value of the hazard index for all water sources and all monitoring dates w/in UCMR5	UCMR5
Total_PFAS	Maximum value of total pfas for all water sources and all monitoring dates w/in UCMR5	UCMR5
CitiesServed	Cities served by utility	SDWA
StateCode	State where utility located	SDWA
PWSTypeDesc	Description of utility type	SDWA
PrimarySource	Primary source of drinking water for utility	SDWA
Population	Population served by utility	SDWA

Analysis Script

The data analysis was completed using R version 4.2.1.

```
#### MCL Rulemaking Nationwide Analysis

# monitoring data -----
library(tidyverse)

# ucmr 5 results JAN 2026 RELEASE ####
ucmr5_raw <- read_tsv("source_data/UCMR/ucmr5-occurrence-data/UCMR5_All.txt")
ucmr5_sampling_events <- ucmr5_raw |>
  #change from ppb to ppt, change non-detects to 0
  mutate(AnalyticalResultValue = replace_na(AnalyticalResultValue, 0),
         AnalyticalResultValue = AnalyticalResultValue*1000) |>
  #remove lithium
  filter(Contaminant != 'lithium') |>
  #select cols
  select(PWSID, PWSName, FacilityID, FacilityName, SamplePointName,
         CollectionDate, SamplePointID, SampleID, Contaminant,
         AnalyticalResultValue) |>
  #pivot so each row is a sampling event at one location
  pivot_wider(names_from = Contaminant,
              values_from = AnalyticalResultValue,
              values_fill = 0)

ucmr5_sampling_events <- ucmr5_sampling_events |>
  #create hazard index
  mutate(hazard_index = `HFPO-DA`/10 + PFBS/2000 + PFNA/10 + PFHxS/10) |>
  rowwise() |>
  mutate(total_pfas = sum(c_across(`8:2 FTS`:PFUnA))) |>
  ungroup()

ucmr5_pws_summary <- ucmr5_sampling_events |>
  group_by(PWSID, PWSName) |>
  #summarize across public water systems to report maximum result
  summarise(across(c("PFOS", "PFOA", "PFBS", "PFNA", "HFPO-DA", "PFHxS",
```

```

      "hazard_index", "total_pfas"), max)) |>
ungroup()

# PWS (ECHO API) ####
library(httr)
library(jsonlite)

#grab metadata
response <- GET(url =
'https://echodata.epa.gov/echo/sdw_rest_services.metadata?output=JSON')
echo_sdwa_metatdata <- fromJSON(response$url)
echo_sdwa_metatdata <- echo_sdwa_metatdata$Results$ResultColumns

# we want cols 1:6, 10:16
echo_api_cols <- echo_sdwa_metatdata |>
  filter(ColumnID %in% c(1:6, 10:16)) |>
  #assign col types to help with api function
  mutate(readr_type = case_when(
    DataType == 'VARCHAR2' ~ 'c',
    DataType == 'CHAR' ~ 'c',
    DataType == 'NUMBER' ~ 'd',
    DataType == 'DATE' ~ 'D',
    TRUE ~ '?'))

#function to use echo api by pwsid
FUN_get_echoAPI_bypwsid <- function(pwsids, echo_cols, data_type) {
  #create the api url and its parameters
  url <- parse_url("https://echodata.epa.gov/")
  url$path <- "echo/sdw_rest_services.get_systems"
  url$query <-list(
    #output is JSON
    output = 'JSON',
    #pwsid code
    p_pid = pwsids)
  #call get.facilities to get info on query
  request <- GET(build_url(url))
  #print status
  print(paste0("status:", request$status_code))
  #parse
  try(content <- fromJSON(request$url))

  #switch to get_download endpoint
  url$path <- "echo/sdw_rest_services.get_download"
  url$query <- list(
    #add query ID
    qid = content$Results$QueryID,
    #specify cols
    qcolumns = paste(echo_cols, collapse = ","))
  #call url

```

```

request <- GET(build_url(url))
#print status
print(paste0("get.download status:", request$status_code))

#read data
try(read_csv(request$url, col_types = paste(data_type, collapse = "")))
}

#create vector of lists of no more than 100 pwsids to iterate over
pwsids_chunks_vector <- ucmr5_raw |>
  pull(PWSID) |>
  unique() %>%
  split(ceiling(seq_along(.) / 100)) %>%
  map_chr(~ str_c(.x, collapse = ","))

#run function politely
echo_pws_sdwa <- map_dfr(pwsids_chunks_vector,
  #use slowly to keep it from throwing 500 errors
  slowly(FUN_get_echoAPI_bypwsid,
    rate = rate_delay(pause = 5)),
  echo_api_cols$ColumnID, echo_api_cols$readr_type)

## join with ucmr summary
ucmr5_pws_summary <- ucmr5_pws_summary |>
  left_join(echo_pws_sdwa |>
    select(PWSID, CitiesServed, StateCode, CountiesServed,
      PWSTypeDesc, PrimarySourceDesc, PopulationServedCount)
  |>
    unique(),
    by = join_by(PWSID == PWSID))
# analysis -----

## total tested ####
ucmr5_pws_summary |>
  summarise(sum(PopulationServedCount, na.rm =T),
    n_distinct(PWSID),
    n_distinct(StateCode)) |>
  ungroup()

## any MCL ####
ucmr5_pws_summary |>
  filter(PFOA>4|PFOS>4|hazard_index>1) |>
  summarise(sum(PopulationServedCount, na.rm =T),
    n_distinct(PWSID),
    n_distinct(StateCode)) |>
  ungroup()

## PFOA/PFOS MCLs ####
ucmr5_pws_summary |>

```

```

filter(PFOA>4|PFOS>4) |>
summarise(sum(PopulationServedCount, na.rm =T),
           n_distinct(PWSID),
           n_distinct(StateCode)) |>
ungroup()

## haz index MCLs ####
ucmr5_pws_summary |>
  filter(hazard_index>1) |>
  summarise(sum(PopulationServedCount, na.rm =T),
            n_distinct(PWSID),
            n_distinct(StateCode)) |>
ungroup()

## not pfoa/pfos haz index ####
ucmr5_pws_summary |>
  filter(hazard_index>1, PFOA<4, PFOS<4) |>
  summarise(sum(PopulationServedCount, na.rm =T),
            n_distinct(PWSID),
            n_distinct(StateCode)) |>
ungroup()

ucmr5_pws_summary |>
  filter(hazard_index>1, PFOS<4, PFOA<4) |>
  pull(StateCode) |> unique()

## population by state impacted by any MCL ####
mcl_summary_by_state <- ucmr5_pws_summary |>
  mutate(mcl=if_else(PFOA>4|PFOS>4|hazard_index>1, "Y", "N")) |>
  group_by(StateCode, mcl) |>
  summarize(pop = sum(PopulationServedCount)) |>
  ungroup() |>
  pivot_wider(names_from = mcl, names_prefix = "mcl_",
              values_from = pop,
              values_fill = 0) |>
  mutate(total_pop = mcl_Y + mcl_N)

# other stats ####

##%iles ####

#PFOS with zeroes
ucmr5_sampling_events |>
  select(PFOS) |>
  mutate(pctile = percent_rank(PFOS) *100,
         find95 = abs(95-pctile)) |>
  slice_min(find95) |>
  summarise(mean(PFOS))
#w/o

```

```

ucmr5_sampling_events |>
  filter(PFOS>0) |>
  select(PFOS) |>
  mutate(pctile = percent_rank(PFOS) *100,
         find95 = abs(95-pctile)) |>
  slice_min(find95) |>
  summarise(mean(PFOS))

#PFOA with zeroes
ucmr5_sampling_events |>
  select(PFOA) |>
  mutate(pctile = percent_rank(PFOA) *100,
         find95 = abs(95-pctile)) |>
  slice_min(find95) |>
  summarise(mean(PFOA))
#without zeroes
ucmr5_sampling_events |>
  filter(PFOA>0) |>
  select(PFOA) |>
  mutate(pctile = percent_rank(PFOA) *100,
         find95 = abs(95-pctile)) |>
  slice_min(find95) |>
  summarise(mean(PFOA))

## % overlap ####

# 1712 with a detection of PFOA or PFOS
ucmr5_sampling_events |>
  filter(PFOA>0|PFOS>0) |>
  summarise(n_distinct(PWSID))
# 2023 w/ detection of haz index
ucmr5_sampling_events |>
  filter(`HFPO-DA`>0|PFBS>0|PFHxS>0|PFNA>0) |>
  summarise(n_distinct(PWSID))
#1271 w/ detections of both
ucmr5_sampling_events |>
  filter((`HFPO-DA`>0|PFBS>0|PFHxS>0|PFNA>0),
         (PFOA>0|PFOS>0)) |>
  summarise(n_distinct(PWSID))

```