

# Robinson et al. 2020 Supplemental Information: Comparing the Radiocarbon and Tree-ring Records for the Upland US Southwest

Kyle Bocinsky<sup>1,\*</sup>

Erick Robinson<sup>2</sup>

19 January, 2020

## Contents

<b>Methods</b>	<b>1</b>
Study Area . . . . .	1
Datasets . . . . .	3
Preparing the SPDs . . . . .	10
Comparing radiocarbon and tree-ring SPDs . . . . .	13
References . . . . .	16
Colophon . . . . .	17

<sup>1</sup> Crow Canyon Archaeological Center

<sup>2</sup> Utah State University

\* Correspondence: Kyle Bocinsky <bocinsky@gmail.com>

## Methods

Here, we directly compare two large collections of radiocarbon and tree-ring dates drawn from the Upland US Southwest (UUSW). After harmonizing those datasets to with one another and calibrating the radiocarbon dates, we calculate binned summed probability distributions across the UUSW and two regions intensely studied during the Village Ecodynamics Project (VEP) — the Central Mesa Verde and the Northern Rio Grande [1]. We compare the proxy records to each other, and to population estimates primarily derived from architecture and ceramic data [2]. The radiocarbon dates we analyze here are drawn from the [DARC: DATABASE AND REFERENCE HERE], and are available [HERE]; The tree-ring data are those presented in [3] and archived on the Digital Archaeological Record (<https://doi.org/10.6067/XCV82J6D7B>).

## Study Area

The UUSW was geographically defined by [3] as the region between 105–113°W and 32–38°N. Here, we refine that definition in two ways. First, because location data for our radiocarbon database are only resolved to the county level for most sites, we define our study region to include all counties whose geographic centroids fall within the region from [3]. Second, because of the generally poor tree-ring date record in the Sonoran Desert, we roughly exclude the Phoenix and Tucson basins by dropping Maricopa, Pinal, and Pima counties in Arizona. We also analyze subsets of both datasets for the Village Ecodynamic Project (VEP) study areas in the Central Mesa Verde (CMV) and Northern Rio Grande (NRG) regions [1]. Figure shows the original UUSW from [3] and revised study area used in this analysis, as well as the two VEP study areas.

```
# A nice projection of the Four Corners states to use for mapping
four_corners_lcc_proj <-
  robinson2020::four_corners_lcc_proj
```

```

# The Four Corners states in four_corners_lcc_proj
four_corners_states <-
  robinson2020::four_corners_states

# The UUSW as defined in Bocinsky et al. 2016
uusw_boundary <- robinson2020::uusw_boundary

# The UUSW study area for this analysis
uusw_counties <-
  robinson2020::uusw_counties

# The VEP study areas
vep_study_areas <-
  robinson2020::vep_study_areas

theme_map <- function(base_size = 6.5,
                      base_family = ""){
  theme_bw(base_size = base_size,
            base_family = base_family) %+replace%
  theme(axis.line = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        axis.title = element_blank(),
        panel.background = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank(),
        panel.grid.major = element_line(colour = 'transparent'),
        panel.grid.minor = element_line(colour = 'transparent'),
        legend.key = element_blank(),
        legend.position = "top",
        plot.background = element_blank(),
        plot.margin = margin(t = 0, r = 0, b = 0, l = 0, unit = "npc")
  )
}

ggplot() +
  geom_sf(data = four_corners_states) +
  geom_raster(data = robinson2020::swus_hillshade_500m,
             mapping = aes(x = x,
                           y = y,
                           alpha = ID),
             na.rm = TRUE) +
  scale_alpha(range = c(0.8, 0),
             na.value = 0,
             limits = c(0,255),
             guide = "none") +
  geom_sf(data = four_corners_states,
         fill = NA) +
  geom_sf(data = uusw_counties %>%
          sf::st_union() %>%
          rmapshaper::ms_filter_islands(),
         fill = NA,
         color = "black") +

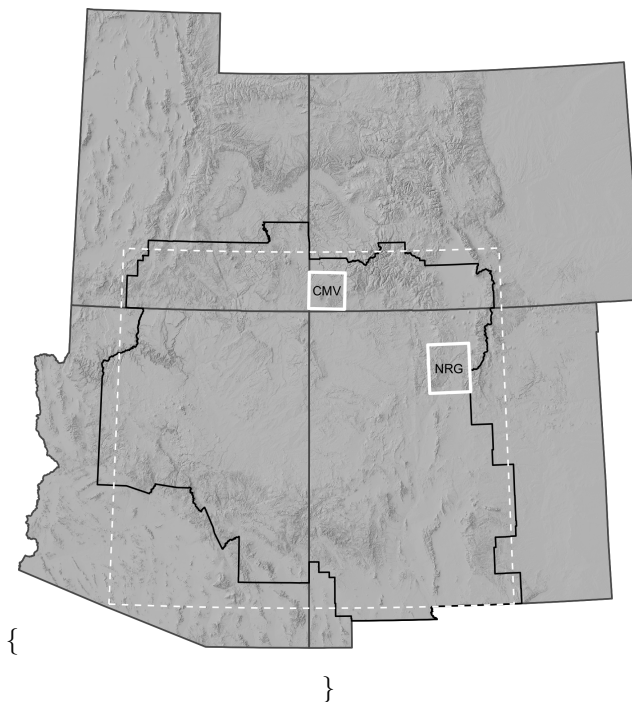
```

```

geom_sf(data = uusw_boundary,
        fill = NA,
        linetype = 2,
        color = "white") +
geom_sf(data = vep_study_areas,
        fill = NA,
        color = "white",
        size = 1) +
geom_sf_text(data = vep_study_areas,
            aes(label = `Study Area`),
            size = 3) +
theme_map(16)

```

\begin{figure}



\caption{The UUSW as defined in this study (region with black border). The white dashed area represents the UUSW as defined in [3]. VEP study areas are white boxes. CMV: Central Mesa Verde; NRG: Northern Rio Grande.} \end{figure}

## Datasets

The radiocarbon data presented here are derived from the [RADIOCARBON DATA DETAILS HERE]. Data from Arizona, Colorado, and New Mexico in the [DATABASE] are spatially resolved to only the county level; we therefor translated the Utah dates to their counties as well, and then spatially cropped the data to our study area.

```

## Read in the radiocarbon database
## radiocarbon_data <-
# readr::read_csv(here::here("analysis/data/raw_data/0403_sbox_clean.csv")) %>%
# # Convert to spatial object
# sf::st_as_sf(coords = c("Long", "Lat"),
#               crs = 4326) %>%

```

```

# sf::st_transform(four_corners_lcc_proj) %>%
# # Select salient columns
# dplyr::select(labnumber,
#               date,
#               sd,
#               Site,
#               SiteID) %>%
# dplyr::rename(Lab_Number = labnumber,
#               Date_BP = date,
#               SD = sd,
#               Site_Number = Site) %>%
# sf::st_intersection(usuw_counties) %>%
# sf::st_drop_geometry() %>%
# dplyr::mutate(Type = "Radiocarbon",
#               SiteID = SiteID %>% stringr::str_c("Radiocarbon - ", .)) %>%
# dplyr::select(Type,
#               SiteID,
#               Site_Number,
#               County,
#               Lab_Number,
#               Date_BP,
#               SD) %T>%
# # Write the output to a new file (csv)
# readr::write_csv(here::here("analysis/data/derived_data/radiocarbon_data.csv"))

```

```

radiocarbon_data <-
  readr::read_csv(here::here("analysis/data/derived_data/radiocarbon_data.csv"))

```

We draw on the tree-ring date database presented in [3] and archived in the Digital Archaeological Record (<https://doi.org/10.6067/XCV82J6D7B>). The tree-ring data we use here were compiled by Tim Kohler and Rebecca Higgins from many contributors, and the vast majority of dates were determined by researchers in the Laboratory of Tree Ring Research at the University of Arizona. The original dataset contained 32,863 cutting, near-cutting, and non-cutting dates from across the US Southwest. Here, we only use the dates that are most likely to be accurate — cutting dates ('B', 'G', 'L', 'c', or 'r' dates, with or without a '+') for which we know the year the tree died, and near-cutting dates ('v' or 'v+' dates) which are likely within 0–3 years of the true date of the outermost ring. The [3] data contain site locations; for this analysis, we first spatially cropped the database to our study area, and then translated those locations to the county level. We converted the original date determinations to BP to match radiocarbon conventions.

```

# # Read in the Bocinsky et al. 2016 tree ring data
# # https://doi.org/10.6067/XCV82J6D7B
# dendro_data <-
#   readr::read_csv(here::here("analysis/data/raw_data/BOCINSKY_ET_AL_2015_TREE_RINGS.csv")) %>%
#   # Convert to spatial object
#   sf::st_as_sf(coords = c("LON", "LAT"),
#                 crs = 4326) %>%
#   sf::st_transform(four_corners_lcc_proj) %>%
#   # Convert dates to BP
#   dplyr::mutate(Date_BP = 1950 - Outer_Date_AD,
#                 Type = ifelse(C_Level %in% c(2,3),
#                               "Cutting or Near-cutting",
#                               "Non-cutting")) %>%
#   # drop C_Level
#   dplyr::select(Seq_Number,

```

```

#           Site_Number,
#           Lab_Number,
#           Type,
#           Date_BP) %>%
#   dplyr::rename(SiteID = Seq_Number) %>%
#   sf::st_intersection(uusw_counties) %>%
#   sf::st_drop_geometry() %>%
#   dplyr::mutate(Type = Type %>% stringr::str_c("Dendro - ", .),
#                 SiteID = SiteID %>% stringr::str_c("Dendro - ", .),
#                 SD = 0) %>%
#   dplyr::filter(Type == "Dendro - Cutting or Near-cutting") %>%
#   dplyr::mutate(Type = "Tree-ring") %>%
#   dplyr::select(Type,
#                 SiteID,
#                 Site_Number,
#                 County,
#                 Lab_Number,
#                 Date_BP,
#                 SD) %T>%
#   # Write the output to a new file (csv)
#   readr::write_csv(here::here("analysis/data/derived_data/dendro_data.csv"))

dendro_data <-
  readr::read_csv(here::here("analysis/data/derived_data/dendro_data.csv"))

```

Archaeological site locations are protected by US federal and state statutes. In order to protect these sensitive data, we pre-processed the both the radiocarbon and tree-ring data; data included in this compendium do not include site locations, and researchers will need to gain permission from the State Historic Preservation Office (SHPO) of each state in order to access the site location data.

```

all_data <- list(Radiocarbon = radiocarbon_data,
                 `Tree-ring` = dendro_data) %>%
  purrr::map(~dplyr::mutate(.x,
                             `Study Area` = ifelse(County == "Montezuma, CO",
                                                    "CMV",
                                                    NA),
                             `Study Area` = ifelse(County %in% c("Los Alamos, NM",
                                                                "Rio Arriba, NM",
                                                                "Sandoval, NM",
                                                                "Santa Fe, NM",
                                                                "Taos, NM"),
                                                    "NRG",
                                                    `Study Area`)))

```

Finally, due to much of the radiocarbon data only being resolved to the county level, we had to approximate inclusion in the VEP CMV and NRG study areas. The CMV study area conforms well to Montezuma county, Colorado; the NRG overlaps with Los Alamos, Rio Arriba, Sandoval, Santa Fe, and Taos counties in New Mexico. Our final dataset includes 2,633 radiocarbon dates from 7980–90 BP (uncalibrated; 6029 BC–1860 AD) and 15,492 tree-ring dates from 1943 – -45 BP (7–1995 AD). Table 1 presents the counts of each date type in the CMV and NRG study areas.

```

all_data %>%
  dplyr::bind_rows() %>%
  dplyr::bind_rows(all_data %>%
                    dplyr::bind_rows() %>%

```

Table 1: Counts of radiocarbon and tree-ring dates in this study.

Study Area	Radiocarbon	Tree-ring
UUSW Total	2,633	15,492
CMV	55	4,480
NRG	265	2,365

```

      dplyr::mutate(`Study Area` = "UUSW Total")) %>%
dplyr::filter(!is.na(`Study Area`)) %>%
dplyr::mutate(`Study Area` = factor(`Study Area`, levels = c("UUSW Total", "CMV", "NRG"))) %>%
dplyr::group_by(Type, `Study Area`) %>%
dplyr::summarise(Count = dplyr::n()) %>%
tidyr::spread(Type, Count) %>%
knitr::kable(caption = "Counts of radiocarbon and tree-ring dates in this study.",
              booktabs = TRUE,
              format.args = list(big.mark = ','))

```

Date densities across the study area were very uneven, with high counts of radiocarbon dates deriving from Navajo County, Arizona, and almost a third of the tree-ring dates in our study being from Montezuma County, Colorado. Below, we attempt to overcome some of these sampling issues by binning dates by site and occupation phase. Still, these disparities likely demonstrate more fundamental differences in research design across archaeological projects. Figure 1 presents the number of radiocarbon and tree-ring dates by county within our study area. Figure 2 presents the number of unique sites with radiocarbon and tree-ring dates by county within our study area.

```

county_counts <-
  all_data %>%
  dplyr::bind_rows() %>%
  dplyr::group_by(Type, County) %>%
  dplyr::summarise(Dates = dplyr::n()) %>%
  dplyr::left_join(uusw_counties,
                  by = "County") %>%
  sf::st_as_sf()

theme_map <- function(base_size = 6.5,
                      base_family = ""){
  theme_bw(base_size = base_size,
            base_family = base_family) %+replace%
  theme(axis.line = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        axis.title = element_blank(),
        panel.background = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank(),
        panel.grid.major = element_line(colour = 'transparent'),
        panel.grid.minor = element_line(colour = 'transparent'),
        legend.key = element_blank(),
        legend.position = "bottom",
        legend.background = element_blank(),
        legend.key.width = unit(0.15, "npc"),
        legend.text = element_text(size = rel(1)),
        plot.background = element_blank(),

```

```

        plot.margin = margin(t = 0, r = 0, b = 0, l = 0, unit = "npc")
    )
}

```

```

county_counts %>%
  dplyr::filter(Type == "Radiocarbon") %>%
  ggplot() +
  geom_sf(data = four_corners_states,
    fill = NA) +
  geom_sf(aes(fill = Dates),
    color = "black") +
  scale_fill_distiller(name = "Number of Radiocarbon Dates",
    palette = "Greys",
    direction = 1,
    limits = c(0,500),
    minor_breaks = seq(50,450,100),
    breaks = seq(0,500,100),
    expand = expand_scale(mult = 0, add = 0),
    guide = guide_colorbar(title.position = "top")) +
  geom_sf(data = uusw_counties,
    fill = NA,
    color = "black") +
  theme_map(16)

```

```

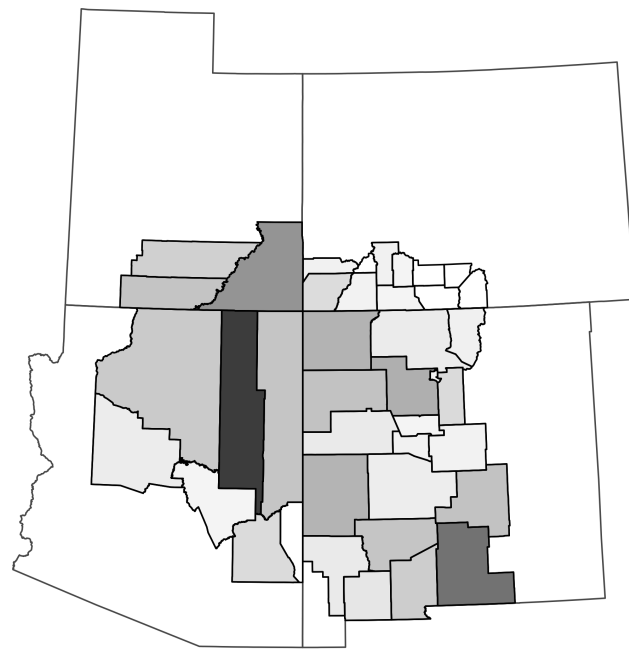
county_counts %>%
  dplyr::filter(Type == "Tree-ring") %>%
  ggplot() +
  geom_sf(data = four_corners_states,
    fill = NA) +
  geom_sf(aes(fill = Dates),
    color = "black") +
  scale_fill_distiller(name = "Number of Tree-ring Dates",
    palette = "Greys",
    direction = 1,
    limits = c(0,5000),
    minor_breaks = seq(500,4500,1000),
    breaks = seq(0,5000,1000),
    expand = expand_scale(mult = 0, add = 0),
    guide = guide_colorbar(title.position = "top")) +
  geom_sf(data = uusw_counties,
    fill = NA,
    color = "black") +
  theme_map(16)

```

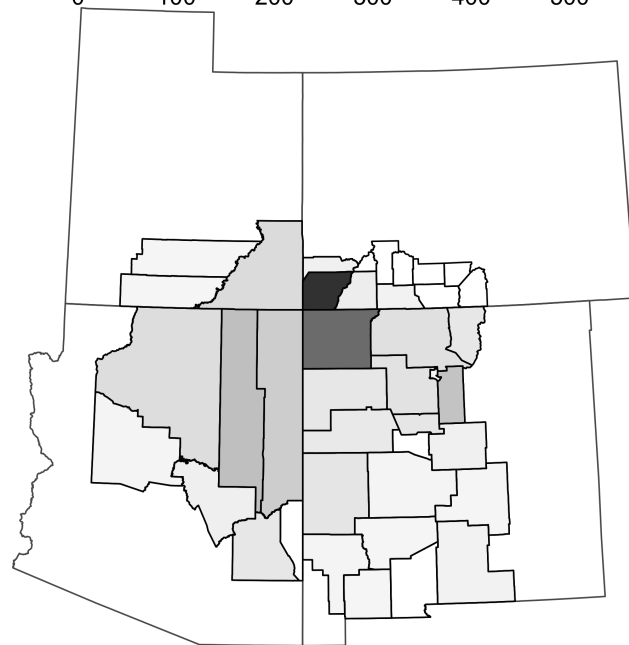
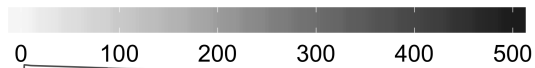
```

county_counts <-
  all_data %>%
  dplyr::bind_rows() %>%
  dplyr::select(Type, SiteID, County) %>%
  dplyr::distinct() %>%
  dplyr::group_by(Type, County) %>%
  dplyr::summarise(Dates = dplyr::n()) %>%
  dplyr::left_join(uusw_counties,
    by = "County") %>%
  sf::st_as_sf()

```



Number of Radiocarbon Dates



Number of Tree-ring Dates

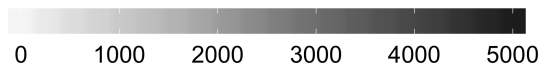


Figure 1: The number of radiocarbon and tree-ring dates by county in this study.



```

theme_map <- function(base_size = 6.5,
                      base_family = ""){
  theme_bw(base_size = base_size,
            base_family = base_family) %+replace%
  theme(axis.line = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        axis.title = element_blank(),
        panel.background = element_blank(),
        panel.border = element_blank(),
        panel.grid = element_blank(),
        panel.grid.major = element_line(colour = 'transparent'),
        panel.grid.minor = element_line(colour = 'transparent'),
        legend.key = element_blank(),
        # legend.position = "top",
        # legend.justification = c(0,0),
        legend.position = "bottom",
        # legend.position = c(0.5,1),
        legend.background = element_blank(),
        legend.key.width = unit(0.15,"npc"),
        # legend.key.height = unit(0.15,"in"),
        legend.text = element_text(size = rel(1)),
        plot.background = element_blank(),
        plot.margin = margin(t = 0, r = 0, b = 0, l = 0, unit = "npc")
  )
}

```

```

county_counts %>%
  dplyr::filter(Type == "Radiocarbon") %>%
  ggplot() +
  geom_sf(data = four_corners_states,
          fill = NA) +
  geom_sf(aes(fill = Dates),
          color = "black") +
  scale_fill_distiller(name = "Number of Sites with Radiocarbon Dates",
                      palette = "Greys",
                      direction = 1,
                      limits = c(0,80),
                      minor_breaks = seq(10,70,20),
                      breaks = seq(0,80,20),
                      expand = expand_scale(mult = 0, add = 0),
                      guide = guide_colorbar(title.position = "top")) +
  geom_sf(data = uusw_counties,
          fill = NA,
          color = "black") +
  theme_map(16)

```

```

county_counts %>%
  dplyr::filter(Type == "Tree-ring") %>%
  ggplot() +
  geom_sf(data = four_corners_states,
          fill = NA) +
  geom_sf(aes(fill = Dates),

```

```

    color = "black") +
scale_fill_distiller(name = "Number of Sites with Tree-ring Dates",
  palette = "Greys",
  direction = 1,
  limits = c(0,250),
  minor_breaks = seq(25,225,50),
  breaks = seq(0,250,50),
  expand = expand_scale(mult = 0, add = 0),
  guide = guide_colorbar(title.position = "top")) +
geom_sf(data = uusw_counties,
  fill = NA,
  color = "black") +
theme_map(16)

```

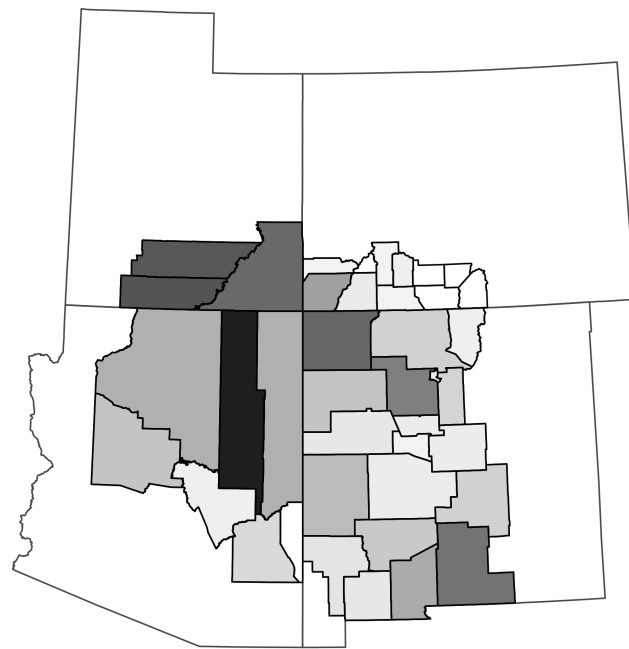
## Preparing the SPDs

To prepare the radiocarbon dataset, we used functions provided by the `rcarbon` package for *R* [4]. We calibrated the radiocarbon dates using the methods of [5], using the ‘`intcal13`’ calibration curve. To control for inconsistent sample size across the archaeological sites represented in the database, we aggregated calibrated radiocarbon dates known to be from the same phase of same archaeological site using the `rcarbon::binprep` function with an `h`-value of 100. This effectively clusters dates from the same site that are within 100 years of each other, and splits site occupations into distinct phases separated by 100 years or more. We constructed summed probability distributions (SPDs) from the calibrated dates using the `rcarbon::spd` function. We averaged probability distributions of binned dates, and then constructed SPDs from the site-phase distributions between 1949–0 BP (1–1950 AD). We smoothed the SPD using a 21-year moving average, and normalised the total probability mass of the smoothed SPD to sum to unity.

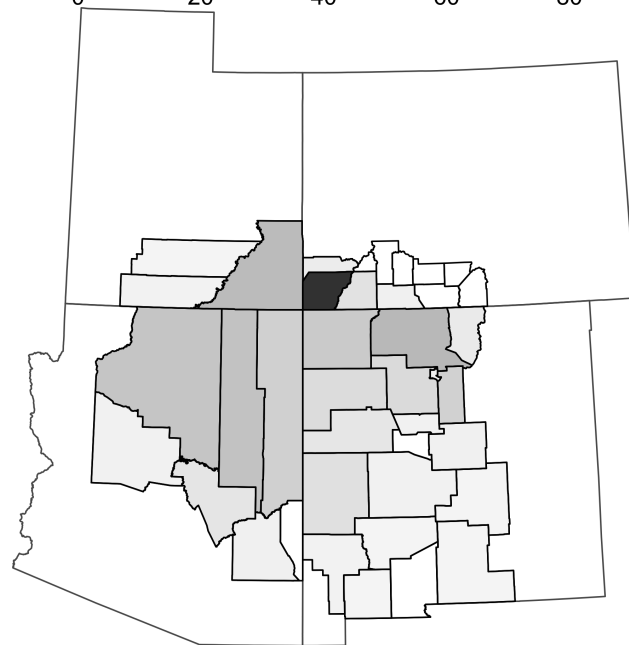
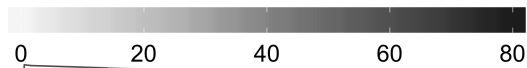
```

if(!file.exists(here::here("analysis/data/derived_data/radiocarbon_spd.rds"))){
  all_data$Radiocarbon %>%
  split(.,.$`Study Area`) %>%
  tibble::tibble(`Study Area` = names(.),
    Dates = .) %>%
  dplyr::bind_rows(tibble::tibble(`Study Area` = "UUSW",
    Dates = list(all_data$Radiocarbon))) %>%
  dplyr::rowwise() %>%
  dplyr::mutate(Calibrated = list(Dates %$%
    rcarbon::calibrate(x = Date_BP,
      errors = SD,
      calMatrix = TRUE,
      verbose = FALSE)),
    Bins = list(rcarbon::binPrep(sites = Dates$SiteID,
      ages = Calibrated,
      h = 100)),
    SPD = list(rcarbon::spd(x = Calibrated,
      bins = Bins,
      datenormalised = TRUE,
      spdnormalised = TRUE,
      timeRange = c(1949,0),
      runm = 21,
      verbose = FALSE))) %>%
  dplyr::ungroup() %>%
  dplyr::mutate(Density = SPD %>%
    purrr::map(function(x){
      x$grid %>%

```



Number of Sites with Radiocarbon Dates



Number of Sites with Tree-ring Dates

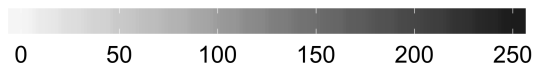


Figure 2: The number of sites with radiocarbon and tree-ring dates by county in this study.

```

        tibble::as_tibble() %$%
        tibble::tibble(Type = "Radiocarbon Density",
                        Date_BP = calBP,
                        Density = PrDens / sum(PrDens, na.rm = T))
      })
    ) %>%
    readr::write_rds(here::here("analysis/data/derived_data/radiocarbon_spd.rds"),
                     compress = "gz")

radiocarbon_spd <- readr::read_rds(here::here("analysis/data/derived_data/radiocarbon_spd.rds"))

```

We treated the tree-ring dates much in the same way as the radiocarbon dates. We modeled faux-calibrated tree-ring date probabilities as discrete degenerate distributions ( $\Pr(X = k_0) = 1$ ) with  $k_0$  equal to the tree-ring date — in other words, a probability of 1 at the date value, and 0 elsewhere. Thinking of tree ring dates this way allows us to use the same binning and SPD algorithms as for radiocarbon dates.<sup>1</sup> Here, we did just that, binning the dates into site-phases using the `rcarbon::binprep` function, and constructing SPDs using `rcarbon::spd`, both with the same parameter selections as used for the radiocarbon samples. As above, we smoothed normalised the SPD.

```

calibrate_tree_ring_dates <- function(x){
  out <- list(
    metadata = tibble::tibble(
      DateID = 1:length(x),
      CRA = x,
      Error = 0,
      Details = NA,
      CalCurve = "intcal13",
      ResOffsets = 0,
      ResErrors = 0,
      StartBP = 50000,
      EndBP = 0,
      Normalised = TRUE,
      F14C = FALSE,
      CalePS = 1e-05
    ),
    grids = NA,
    calmatrix =
      tibble::tibble(index = 1:length(x),
                     year_bp = x,
                     value = TRUE) %>%
      tidyr::pivot_wider(names_from = index,
                         values_from = value) %>%
      dplyr::bind_rows(tibble::tibble(year_bp = 50000:0)) %>%
      dplyr::filter(!duplicated(year_bp)) %>%
      dplyr::arrange(-year_bp) %>%
      dplyr::filter(year_bp >= 0) %>%
      tibble::column_to_rownames("year_bp") %>%
      as.matrix() %>%
      tidyr::replace_na(0) %>%
      magrittr::set_colnames(NULL)
  )
}

```

<sup>1</sup>One might consider alternative probability distributions for non-cutting or near-cutting dates. The probability mass function in such a case would be zero for years prior to the date of the outer ring of the sample, and non-zero for subsequent years, presumably decaying quickly. For instance, we could imagine a study that derives empirical probability distributions for non-cutting dates based on cutting-dates from the same contexts.

```

)

attr(out,"class") <- c("CalDates", "list")

return(out)
}

if(!file.exists(here::here("analysis/data/derived_data/dendro_spd.rds")))
  all_data$`Tree-ring` %>%
  split(.,.$`Study Area`) %>%
  tibble::tibble(`Study Area` = names(.),
                 Dates = .) %>%
  dplyr::bind_rows(tibble::tibble(`Study Area` = "UUSW",
                                   Dates = list(all_data$`Tree-ring`))) %>%

  dplyr::rowwise() %>%
  dplyr::mutate(Calibrated = list(Dates %$%
                                   calibrate_tree_ring_dates(x = Date_BP)),
               Bins = list(rcarbon::binPrep(sites = Dates$SiteID,
                                             ages = Dates$Date_BP,
                                             h = 100)),
               SPD = list(rcarbon::spd(x = Calibrated,
                                       bins = Bins,
                                       datenormalised = TRUE,
                                       spdnormalised = TRUE,
                                       timeRange = c(1949,0),
                                       runm = 21,
                                       verbose = FALSE))) %>%

  dplyr::ungroup() %>%
  dplyr::mutate(Density = SPD %>%
                purrr::map(function(x){
                  x$grid %>%
                    tibble::as_tibble() %$%
                    tibble::tibble(Type = "Tree-ring Density",
                                    Date_BP = calBP,
                                    Density = PrDens / sum(PrDens, na.rm = T))
                })
  ) %>%
  readr::write_rds(here::here("analysis/data/derived_data/dendro_spd.rds"),
                  compress = "gz")

dendro_spd <- readr::read_rds(here::here("analysis/data/derived_data/dendro_spd.rds"))

```

## Comparing radiocarbon and tree-ring SPDs

```

all_density <-
  dendro_spd %>%
  dplyr::select(`Study Area`,Density) %>%
  tidyr::unnest(cols = c(Density)) %>%
  dplyr::bind_rows(radiocarbon_spd %>%
                    dplyr::select(`Study Area`,Density) %>%
                    tidyr::unnest(cols = c(Density))) %>%
  dplyr::mutate(`Year AD` = 1950 - Date_BP) %>%

```

```

dplyr::select(`Year AD`,
              Date_BP,
              `Study Area`,
              Type,
              Density) %>%
dplyr::rename(Value = Density)

vep_demography <-
  robinson2020::vep_demography %>%
  dplyr::rowwise() %>%
  dplyr::mutate(`Year AD` = list((Start+1):End)) %>%
  dplyr::select(`Study Area`,
                `Year AD`,
                Population) %>%
  tidyr::unnest(cols = c("Year AD")) %>%
  dplyr::mutate(Type = "Population",
                Date_BP = 1950 - `Year AD`) %>%
  dplyr::rename(Value = Population)

all_density %>%
  dplyr::bind_rows(vep_demography) %>%
  dplyr::mutate(Type = factor(Type,
                              levels = c("Radiocarbon Density",
                                          "Tree-ring Density",
                                          "Population"),
                              ordered = TRUE),
                `Study Area` = factor(`Study Area`,
                                       levels = c(`Upland US Southwest` = "UUSW",
                                                  `Central Mesa Verde` = "CMV",
                                                  `Northern Rio Grande` = "NRG"))
  ) %>%
  ggplot(aes(x = `Year AD`)) +
  geom_line(aes(y = Value,
                color = `Study Area`,
                size = `Study Area`)) +
  scale_color_manual(values = c("UUSW" = "black",
                                "CMV" = "gray50",
                                "NRG" = "gray80")) +
  scale_size_manual(values = c("UUSW" = 1.25,
                                "CMV" = 0.75,
                                "NRG" = 0.75)) +
  # coord_cartesian(xlim = c(600,1600)) +
  xlab("Year AD") +
  scale_x_continuous(breaks = c(1,seq(200,2000,200))) +
  theme_minimal(18) +
  theme(legend.title = element_blank(),
        legend.justification = c(0, 1),
        legend.position = "bottom",
        axis.title.x = element_text(size = 12),
        axis.text.x = element_text(size = 10),
        axis.title.y = element_blank(),
        axis.text.y = element_blank(),
        panel.grid.major.y = element_blank(),

```

```

panel.grid.minor.y = element_blank(),
plot.margin = unit(c(0.25,0.5,0.25,0.25),"in")
) +
facet_wrap("Type",
  nrow = 3,
  scales = "free_y")

```

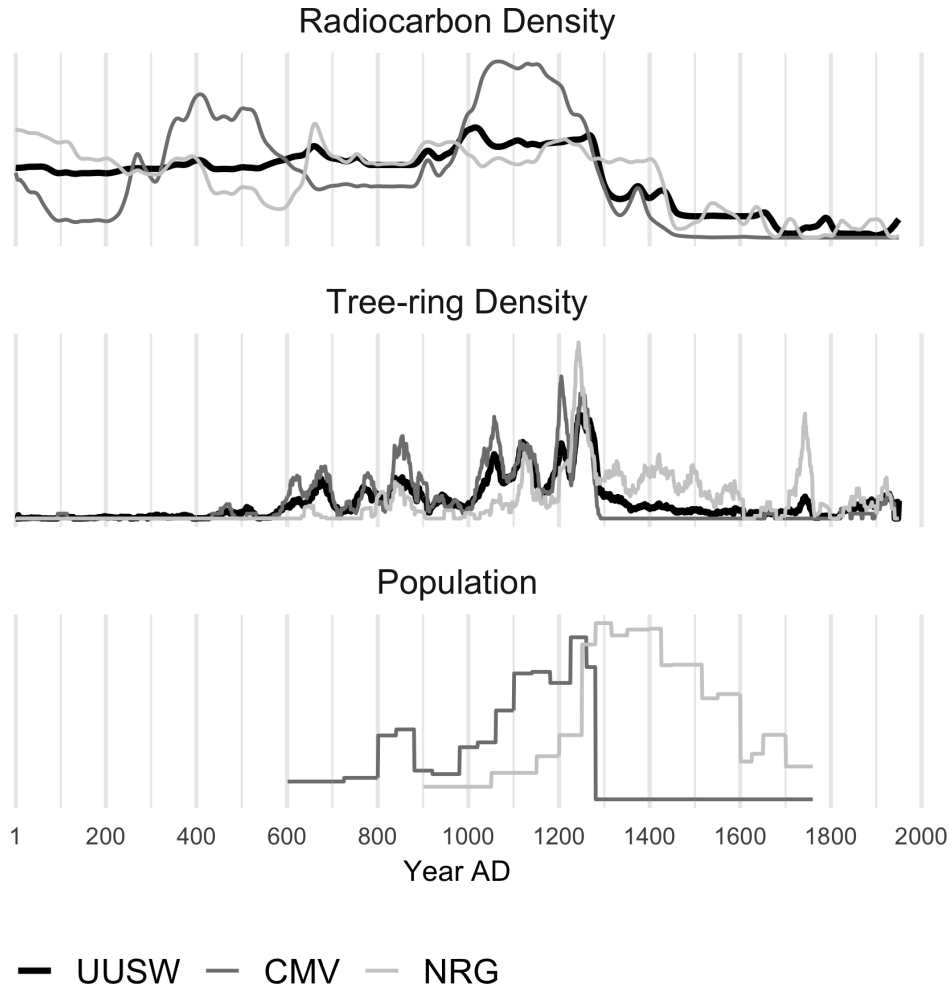


Figure 3: Radiocarbon and tree-ring density distributions in the UUSW, CMV, and NRG regions, and VEP population estimates for the CMV and NRG. Y-axis labels are omitted for simplicity; the maximum population in the lower panel is 30,000 people, and the minimum is 0. Dates are given in AD.

## References

1. Schwindt DM, Bocinsky RK, Ortman SG, Glowacki DM, Varien MD, Kohler TA. 2016 The social consequences of climate change in the Central Mesa Verde region. *American Antiquity* **81**, 74–96.
2. Ortman SG. 2014 Uniform probability density analysis and population history in the Northern Rio Grande. *Journal of Archaeological Method and Theory*
3. Bocinsky RK, Rush J, Kintigh KW, Kohler TA. 2016 Exploration and exploitation in the macrohistory of the pre-Hispanic Pueblo Southwest. *Science Advances* **2**, e1501532.
4. Bevan A, Crema ER. 2018 *rcarbon: Methods for calibrating and analysing radiocarbon dates*. See <https://github.com/ahb108/rcarbon>.
5. Bronk Ramsey C. 2008 Radiocarbon dating: Revolutions in understanding. *Archaeometry* **50**, 249–275.
6. Marwick B. 2017 Computational reproducibility in archaeological research: Basic principles and a case study of their implementation. *Journal of Archaeological Method and Theory* **24**, 424–450.
7. Marwick B. 2019 *rrtools: Creates a Reproducible Research Compendium*. See <https://github.com/benmarwick/rrtools>.



## Colophon

This report is part of a research compendium developed using the `rrtools` package for reproducible research [6]. This report was generated on 2020-01-19 09:26:11 using the following computational environment and dependencies:

```
# which R packages and versions?
if ("devtools" %in% installed.packages()) devtools::session_info()

#> - Session info -----
#> setting      value
#> version      R version 3.6.2 (2019-12-12)
#> os           macOS Catalina 10.15.2
#> system       x86_64, darwin15.6.0
#> ui           RStudio
#> language      (EN)
#> collate     en_US.UTF-8
#> ctype        en_US.UTF-8
#> tz           America/Denver
#> date         2020-01-19
#>

#> - Packages -----
#> package      * version      date      lib source
#> abind         1.4-5        2016-07-21 [1] CRAN (R 3.6.0)
#> assertthat    0.2.1        2019-03-21 [1] CRAN (R 3.6.0)
#> backports     1.1.5        2019-10-02 [1] CRAN (R 3.6.0)
#> bookdown      0.17         2020-01-11 [1] CRAN (R 3.6.2)
#> callr         3.4.0        2019-12-09 [1] CRAN (R 3.6.0)
#> class         7.3-15       2019-01-01 [1] CRAN (R 3.6.2)
#> classInt      0.4-2        2019-10-17 [1] CRAN (R 3.6.0)
#> cli           2.0.1        2020-01-08 [1] CRAN (R 3.6.0)
#> clisymbols    1.2.0        2017-05-21 [1] CRAN (R 3.6.0)
#> codetools     0.2-16       2018-12-24 [1] CRAN (R 3.6.2)
#> colorspace    1.4-1        2019-03-18 [1] CRAN (R 3.6.0)
#> crayon        1.3.4        2017-09-16 [1] CRAN (R 3.6.0)
#> crul          0.9.0        2019-11-06 [1] CRAN (R 3.6.0)
#> curl          4.3          2019-12-02 [1] CRAN (R 3.6.0)
#> DBI           1.1.0        2019-12-15 [1] CRAN (R 3.6.0)
#> deldir        0.1-23       2019-07-31 [1] CRAN (R 3.6.0)
#> desc          1.2.0        2018-05-01 [1] CRAN (R 3.6.0)
#> devtools      2.2.1        2019-09-24 [1] CRAN (R 3.6.0)
#> digest        0.6.23       2019-11-23 [1] CRAN (R 3.6.0)
#> doParallel    1.0.15       2019-08-02 [1] CRAN (R 3.6.0)
#> dplyr         0.8.3        2019-07-04 [1] CRAN (R 3.6.0)
#> e1071         1.7-3        2019-11-26 [1] CRAN (R 3.6.0)
#> ellipsis      0.3.0        2019-09-20 [1] CRAN (R 3.6.0)
#> evaluate      0.14         2019-05-28 [1] CRAN (R 3.6.0)
#> fansi         0.4.1        2020-01-08 [1] CRAN (R 3.6.0)
#> farver        2.0.1        2019-11-13 [1] CRAN (R 3.6.0)
#> foreach       1.4.7        2019-07-27 [1] CRAN (R 3.6.0)
#> foreign       0.8-74       2019-12-26 [1] CRAN (R 3.6.0)
#> fs            1.3.1        2019-05-06 [1] CRAN (R 3.6.0)
#> geojson       0.3.2        2019-01-31 [1] CRAN (R 3.6.0)
#> geojsonio     0.8.0        2019-10-29 [1] CRAN (R 3.6.0)
#> geojsonlint   0.3.0        2019-02-08 [1] CRAN (R 3.6.0)
#> ggplot2       * 3.2.1      2019-08-10 [1] CRAN (R 3.6.0)
```

```

#> git2r          0.26.1    2019-06-29 [1] CRAN (R 3.6.0)
#> glue           1.3.1.9000 2020-01-17 [1] Github (tidyverse/glue@8094d3b)
#> goftest        1.2-2      2019-12-02 [1] CRAN (R 3.6.0)
#> gtable         0.3.0      2019-03-25 [1] CRAN (R 3.6.0)
#> here           0.1        2017-05-28 [1] CRAN (R 3.6.0)
#> highr          0.8        2019-03-20 [1] CRAN (R 3.6.0)
#> hms            0.5.2      2019-10-30 [1] CRAN (R 3.6.0)
#> htmltools      0.4.0      2019-10-04 [1] CRAN (R 3.6.0)
#> httpcode       0.2.0      2016-11-14 [1] CRAN (R 3.6.0)
#> httr           1.4.1      2019-08-05 [1] CRAN (R 3.6.0)
#> iterators      1.0.12     2019-07-26 [1] CRAN (R 3.6.0)
#> jqr            1.1.0      2018-10-22 [1] CRAN (R 3.6.0)
#> jsonlite       1.6        2018-12-07 [1] CRAN (R 3.6.0)
#> jsonvalidate   1.1.0      2019-06-25 [1] CRAN (R 3.6.0)
#> KernSmooth     2.23-16    2019-10-15 [1] CRAN (R 3.6.2)
#> knitr          1.27       2020-01-16 [1] CRAN (R 3.6.2)
#> labeling       0.3        2014-08-23 [1] CRAN (R 3.6.0)
#> lattice        0.20-38    2018-11-04 [1] CRAN (R 3.6.2)
#> lazyeval       0.2.2      2019-03-15 [1] CRAN (R 3.6.0)
#> lifecycle      0.1.0      2019-08-01 [1] CRAN (R 3.6.0)
#> magick         2.2        2019-08-26 [1] CRAN (R 3.6.0)
#> magrittr       * 1.5      2014-11-22 [1] CRAN (R 3.6.0)
#> maptools       0.9-9      2019-12-01 [1] CRAN (R 3.6.0)
#> Matrix         1.2-18     2019-11-27 [1] CRAN (R 3.6.2)
#> memoise        1.1.0      2017-04-21 [1] CRAN (R 3.6.0)
#> mgcv           1.8-31     2019-11-09 [1] CRAN (R 3.6.2)
#> munsell        0.5.0      2018-06-12 [1] CRAN (R 3.6.0)
#> nlme           3.1-143    2019-12-10 [1] CRAN (R 3.6.0)
#> pillar         1.4.3      2019-12-20 [1] CRAN (R 3.6.0)
#> pkgbuild       1.0.6      2019-10-09 [1] CRAN (R 3.6.0)
#> pkgconfig      2.0.3      2019-09-22 [1] CRAN (R 3.6.0)
#> pkgload        1.0.2      2018-10-29 [1] CRAN (R 3.6.0)
#> polyclip       1.10-0     2019-03-14 [1] CRAN (R 3.6.0)
#> prettyunits    1.1.0      2020-01-09 [1] CRAN (R 3.6.0)
#> processx       3.4.1      2019-07-18 [1] CRAN (R 3.6.0)
#> ps             1.3.0      2018-12-21 [1] CRAN (R 3.6.0)
#> purrr          0.3.3      2019-10-18 [1] CRAN (R 3.6.0)
#> R6             2.4.1      2019-11-12 [1] CRAN (R 3.6.0)
#> rcarbon        1.3.0      2019-12-16 [1] CRAN (R 3.6.0)
#> RColorBrewer   1.1-2      2014-12-07 [1] CRAN (R 3.6.0)
#> Rcpp           1.0.3      2019-11-08 [1] CRAN (R 3.6.0)
#> readr          1.3.1      2018-12-21 [1] CRAN (R 3.6.0)
#> remotes        2.1.0      2019-06-24 [1] CRAN (R 3.6.0)
#> rgdal          1.4-8      2019-11-27 [1] CRAN (R 3.6.0)
#> rgeos          0.5-2      2019-10-03 [1] CRAN (R 3.6.0)
#> rlang          0.4.2      2019-11-23 [1] CRAN (R 3.6.0)
#> rmapshaper     0.4.1      2018-10-16 [1] CRAN (R 3.6.0)
#> rmarkdown      2.0        2019-12-12 [1] CRAN (R 3.6.0)
#> robinson2020   * 1.0.0.9000 2020-01-18 [1] local
#> rpart          4.1-15     2019-04-12 [1] CRAN (R 3.6.2)
#> rprojroot      1.3-2      2018-01-03 [1] CRAN (R 3.6.0)
#> rstudioapi     0.10       2019-03-19 [1] CRAN (R 3.6.0)
#> scales         1.1.0      2019-11-18 [1] CRAN (R 3.6.0)

```

```
#> sessioninfo      1.1.1      2018-11-05 [1] CRAN (R 3.6.0)
#> sf                0.8-0      2019-09-17 [1] CRAN (R 3.6.0)
#> sp                1.3-2      2019-11-07 [1] CRAN (R 3.6.0)
#> spatstat          1.62-2     2019-12-10 [1] CRAN (R 3.6.0)
#> spatstat.data     1.4-0      2018-10-04 [1] CRAN (R 3.6.0)
#> spatstat.utils    1.15-0     2019-12-02 [1] CRAN (R 3.6.0)
#> stringi           1.4.5      2020-01-11 [1] CRAN (R 3.6.0)
#> stringr           1.4.0      2019-02-10 [1] CRAN (R 3.6.0)
#> tensor            1.5        2012-05-05 [1] CRAN (R 3.6.0)
#> testthat          2.3.1      2019-12-01 [1] CRAN (R 3.6.0)
#> tibble            2.1.3      2019-06-06 [1] CRAN (R 3.6.0)
#> tidyr             1.0.0      2019-09-11 [1] CRAN (R 3.6.0)
#> tidyselect        0.2.5      2018-10-11 [1] CRAN (R 3.6.0)
#> units             0.6-5      2019-10-08 [1] CRAN (R 3.6.0)
#> usethis           1.5.1      2019-07-04 [1] CRAN (R 3.6.0)
#> utf8              1.1.4      2018-05-24 [1] CRAN (R 3.6.0)
#> V8                2.3        2019-07-02 [1] CRAN (R 3.6.0)
#> vctrs             0.2.1      2019-12-17 [1] CRAN (R 3.6.0)
#> withr             2.1.2      2018-03-15 [1] CRAN (R 3.6.0)
#> xfun              0.12       2020-01-13 [1] CRAN (R 3.6.0)
#> yaml              2.2.0      2018-07-25 [1] CRAN (R 3.6.0)
#> zeallot           0.1.0      2018-01-28 [1] CRAN (R 3.6.0)
#>
#> [1] /Library/Frameworks/R.framework/Versions/3.6/Resources/library
```

The current Git commit details are:

```
# what commit is this file at?
if ("git2r" %in% installed.packages() & git2r::in_repository(path = ".")) git2r::repository(here::here())
#> Local:      master /Users/bocinsky/git/robinson2020
#> Remote:     master @ origin (https://github.com/bocinsky/robinson2020.git)
#> Head:       [4f0c918] 2020-01-19: .DS_Store banished!
```