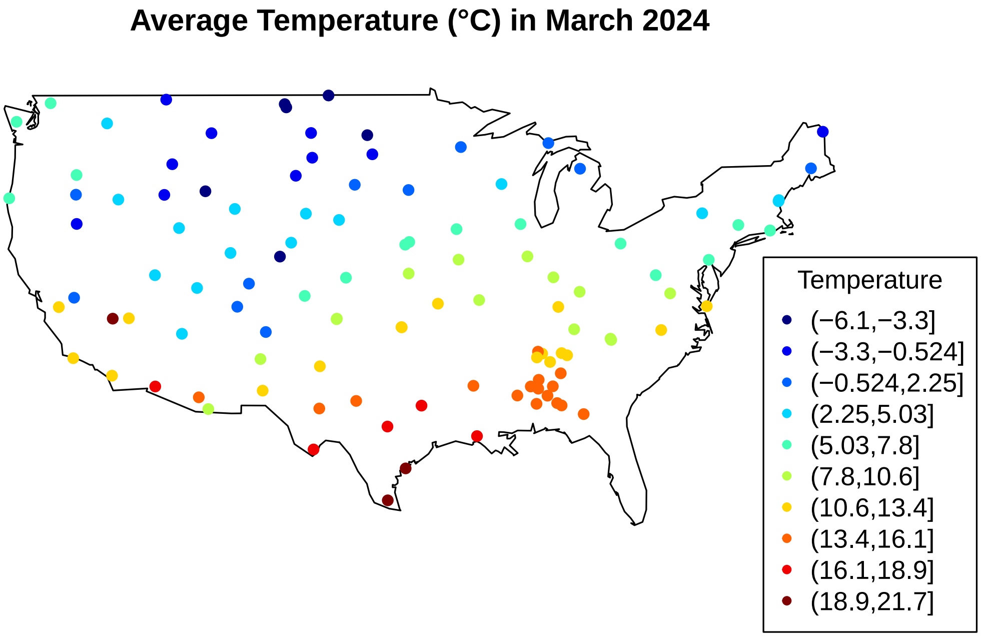
vignette

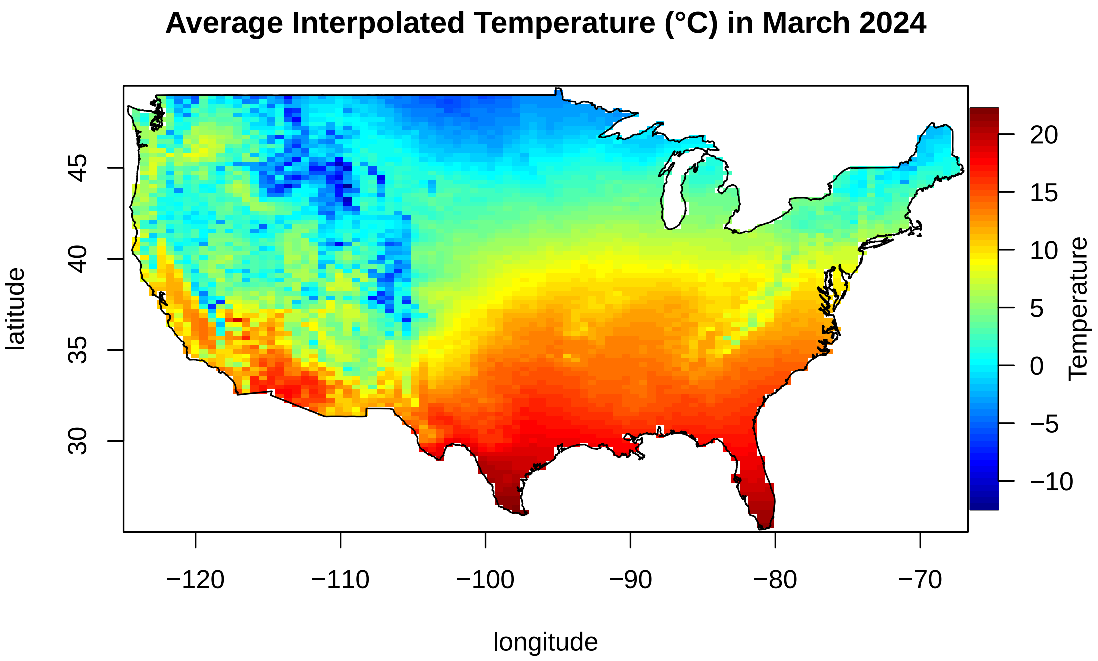
```r
library(USweather)
```

```r
# map of the average temperature of stations in March 2024 vs the interpolated data map

# create list of all possible non contiguous US states/provinces
non_contig <- c("AK",
                "HI",
                "NL",
                "PE",
                "NS",
                "NB",
                "QC",
                "ON",
                "MB",
                "SK",
                "AB",
                "BC",
                "YT",
                "NT",
                "NU")

# station info filtered to only contiguous US stations
contig_stations <- station_info[!station_info$state %in% non_contig, ]

# get march 2024 average temperatures for contiguous stations
march_2024_data <- daily_weather |> dplyr:::filter(LST_DATE >= "2024-03-01",
                                                   LST_DATE < "2024-04-01",
                                                   !state %in% non_contig) |>
  dplyr:::group_by(`station name`, LONGITUDE, LATITUDE) |>
  dplyr:::summarise(avg_temp = mean(T_DAILY_AVG))

# define color map to match the interpolated graph
colmap <- colorRampPalette(
  c(
    "#00007F",
    "blue",
    "#007FFF",
    "cyan",
    "#7FFF7F",
    "yellow",
    "#FF7F00",
    "red",
    "#7F0000"
  )
)
```

```r
# plot map of contiguous US with station March 2024 averages
maps::map("usa")
points(
  march_2024_data$LONGITUDE,
  march_2024_data$LATITUDE,
  pch = 16,
  col =  colmap(10)[cut(march_2024_data$avg_temp, 10)]
)
legend(
  x = -72,
  y = 40,
  title = "Temperature",
  legend = levels(cut(march_2024_data$avg_temp, 10)),
  col = colmap(10),
  pch = 20,
  xpd = TRUE
)
title("Average Temperature (°C) in March 2024")

# create and plot interpolated March 2024 average temperatures
grid <- create_grid(resolution_X = 100, resolution_Y = 100)
```

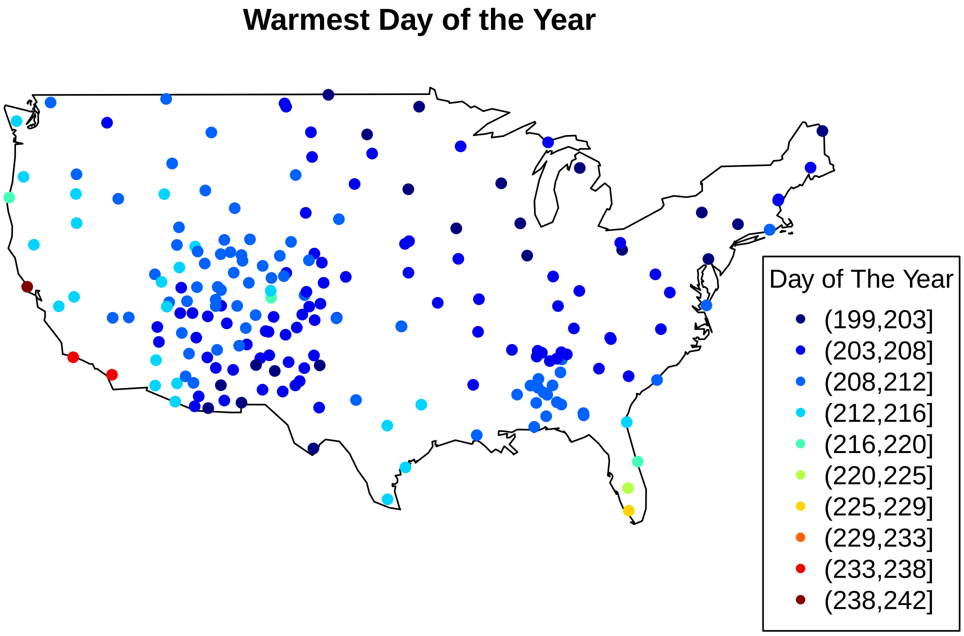


```r
interp_data = interpolate_data(
  march_2024_data$avg_temp,
  march_2024_data$LONGITUDE,
  march_2024_data$LATITUDE,
  gridpoints = grid,
  use_elev = T
)
#> 17 observations removed due to missingness or Inf
#> Assuming columns 1 and 2 of locs are (longitude,latidue) in degrees
plot_interpolations(interp_data)
title("Average Interpolated Temperature (°C) in March 2024")
mtext("Temperature", side = 4, line = 3)
```

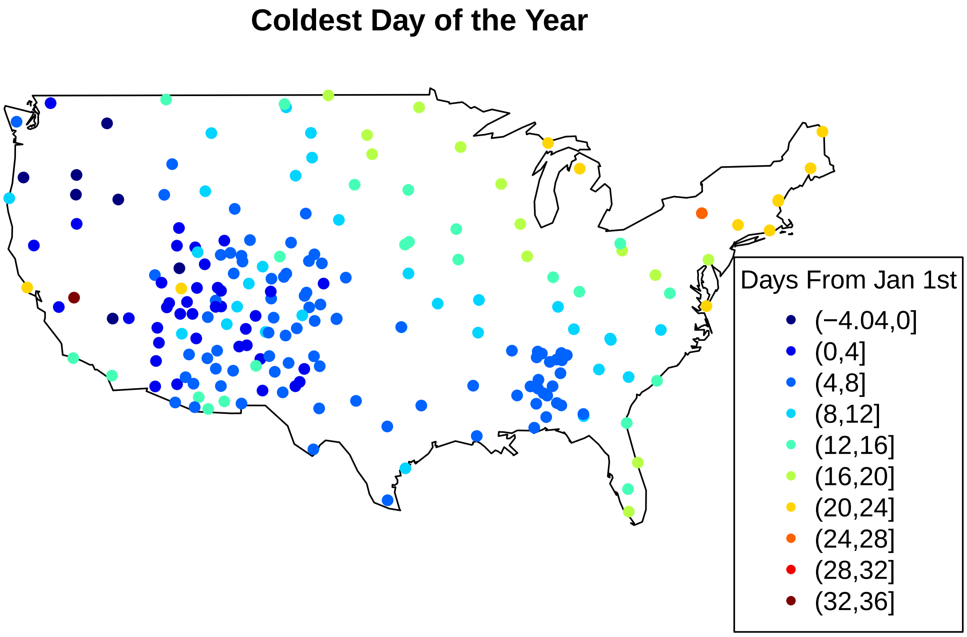


```
# estimating the warmest and coldest days of the year

# create empty vectors for low and high days
num_stations <- nrow(contig_stations)
stations_high_day <- rep(NA, times = num_stations)
stations_low_day <- rep(NA, times = num_stations)

# iterate through the contiguous stations and find and assign low and highs days
i <- 1
for (station_id in contig_stations$WBANNO) {
  yearly_estimates = get_yearly_cycle(station_id)
  min_row <- which.min(yearly_estimates$avg_temp)
  max_row <- which.max(yearly_estimates$avg_temp)
  stations_high_day[i] <- yearly_estimates[max_row, "day_of_year"]
  stations_low_day[i] <- yearly_estimates[min_row, "day_of_year"]
  i <- i + 1
}

# calculate days from Jan 1 to prevent coldest days in December from being large
days_from_jan1 <- ifelse(stations_low_day > 100, stations_low_day - 365, stations_low_day)

# plot station warmest day of the year
maps::map("usa")
points(
  contig_stations$LONGITUDE,
```

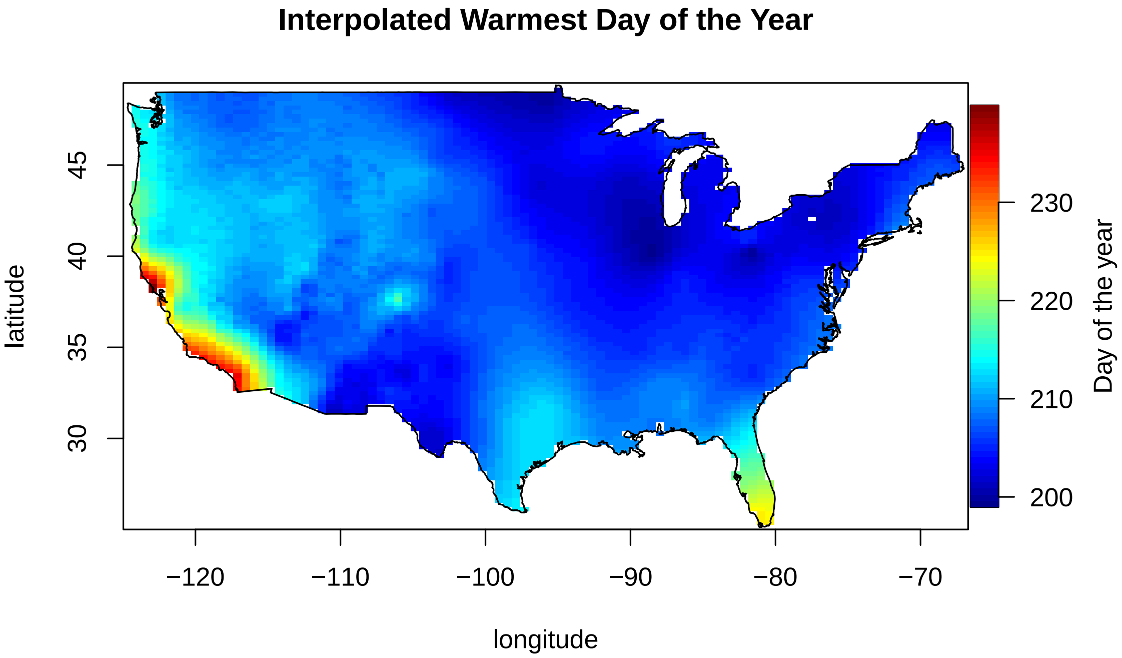
```r
  contig_stations$LATITUDE,
  pch = 16,
  col =  colmap(10)[cut(stations_high_day, 10)]
)
legend(
  x = -72,
  y = 40,
  title = "Day of The Year",
  legend = levels(cut(stations_high_day, 10)),
  col = colmap(10),
  pch = 20,
  xpd = TRUE
)
title("Warmest Day of the Year")
```



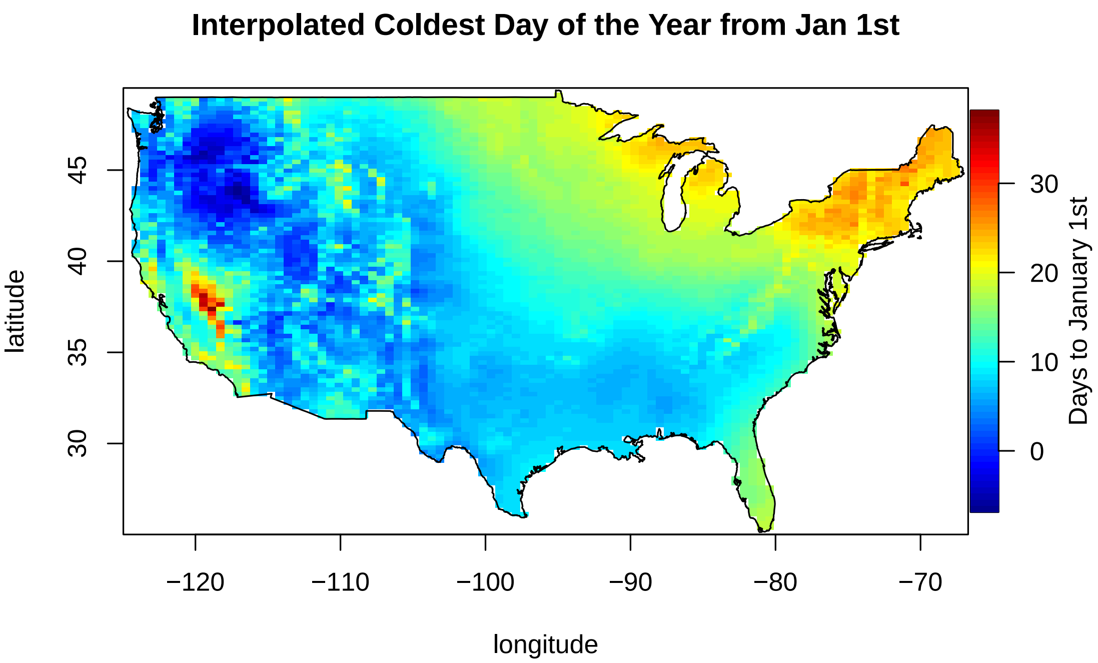
```r
# plot station coldest day of the year (from Jan 1)
maps::map("usa")
points(
  contig_stations$LONGITUDE,
  contig_stations$LATITUDE,
  pch = 16,
  col =  colmap(10)[cut(days_from_jan1, 10)]
)
legend(
  x = -74,
  y = 40,
  title = "Days From Jan 1st",
  legend = levels(cut(days_from_jan1, 10)),
  col = colmap(10),
  pch = 20,
  xpd = TRUE
)
title("Coldest Day of the Year")
```



```
# interpolate and plot warmest day of the year
high_day_interp_data <- interpolate_data(
  stations_high_day,
  contig_stations$LONGITUDE,
  contig_stations$LATITUDE,
  gridpoints = grid,
  use_elev=T
)
#> 1 observations removed due to missingness or Inf
#> Assuming columns 1 and 2 of locs are (longitude,latidue) in degrees
plot_interpolations(high_day_interp_data)
title("Interpolated Warmest Day of the Year")
mtext("Day of the year", side = 4, line = 3.8)
```



```
# interpolate and plot coldest day of the year
low_day_interp_data <- interpolate_data(days_from_jan1,
                                        contig_stations$LONGITUDE,
                                        contig_stations$LATITUDE,
                                        gridpoints = grid,
                                        use_elev=T)
#> 1 observations removed due to missingness or Inf
#> Assuming columns 1 and 2 of locs are (longitude,latidue) in degrees
plot_interpolations(low_day_interp_data)
title("Interpolated Coldest Day of the Year from Jan 1st")
mtext("Days to January 1st", side = 4, line = 3)
```

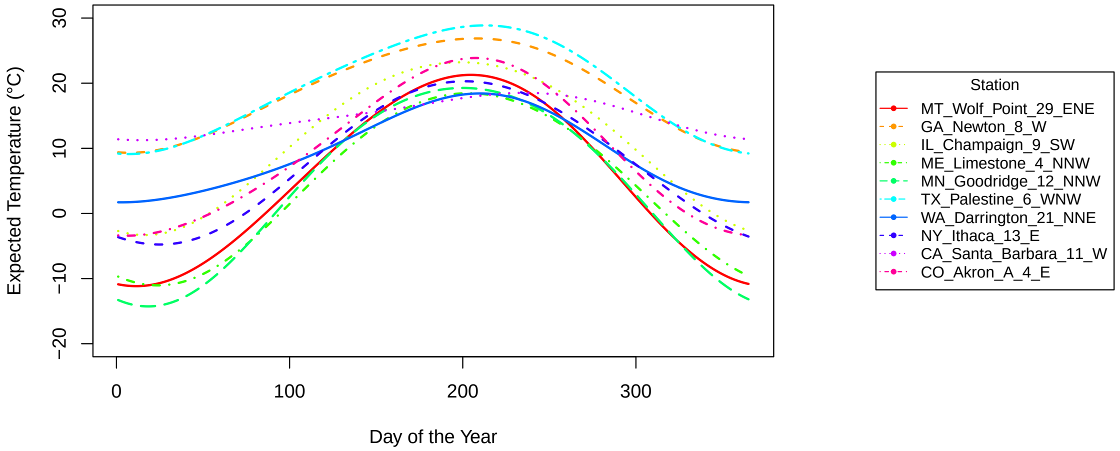


To estimate the warmest and coldest day of the year for each station, we first estimate the yearly temperature cycle for the stations. For each station, we create a ordinary least squares regression based on historical station data which optimizes to predict the daily average temperature for each day of they year using a model with the covariates: $sin(\frac{2\pi*dayofY}{yearlength})$, $cos(\frac{2\pi*dayofY}{yearlength})$, $sin(\frac{4\pi*dayofY}{yearlength})$, and $cos(\frac{4\pi*dayofY}{yearlength})$. These parameters are used to use represent days of the year as a cycle, representing January 1st and December 31st as similar days for weather as an assumption. The linear regression also assumes that each datapoint follows the formula $y_i = x_i * b + \epsilon_i$, where y_i is the average temperature on a day of the year for a station, x_i is vector with 1 (for the intercept term) and $sin(\frac{2\pi*dayofY}{yearlength})$, $cos(\frac{2\pi*dayofY}{yearlength})$, $sin(\frac{4\pi*dayofY}{yearlength})$, and $cos(\frac{4\pi*dayofY}{yearlength})$ as entries, b is a vector of coefficients for each entry in x_i, and ϵ_i is an error distributed normally with mean 0 and variance σ^2, independent from other errors. The optimization finds the b vector that results in the lowest sum of squares: $\sum((y - \hat{y})^2)$. This can be solved both numerically and with a closed-form solution.

Then, to find the coldest and warmest days of the year for each station, we take the days with the minimum and maximum temperature respectively. However, to plot the coldest day of the year, we find the days from January 1st to ensure that days near the end and start of the year are probably encoded as similar dates due to the cyclic nature of the year.

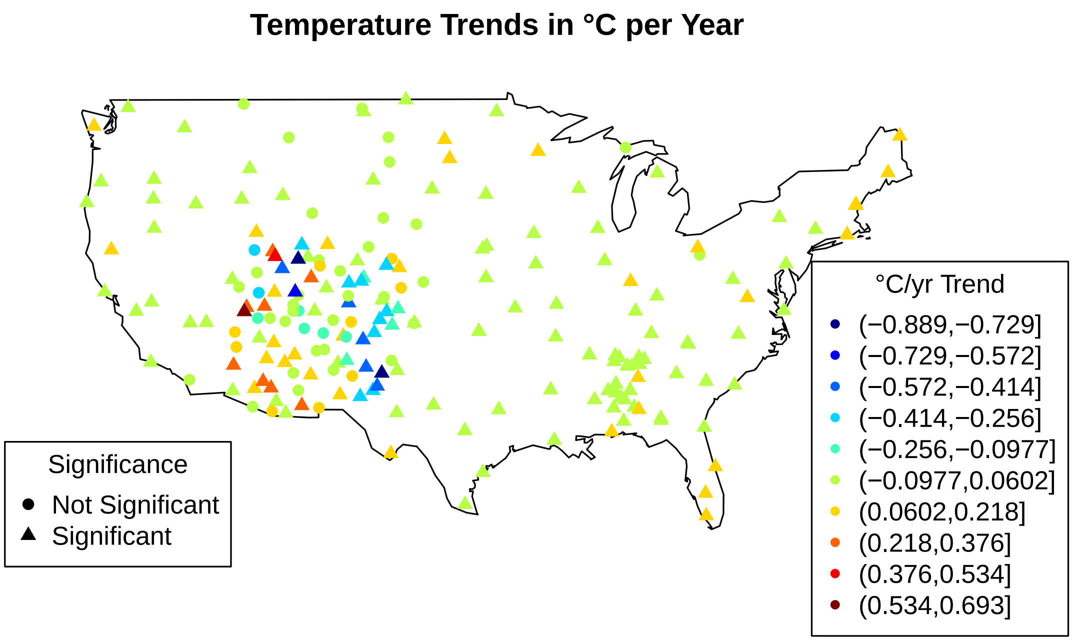
The warmest day of the year is much later in places that tend to be warmer such as CA or FL, while the coldest day of the year is earlier the further you get inland.

```
# yearly weather cycles for 10 diverse US stations

# pick station ID's, done by inspection of the data set
diverse_stations_ids <- sort(c(63828, 94060, 4994, 64758,
                               53152, 94092, 53968, 94645,
                               54808, 4223))
diverse_stations <- station_info[station_info$WBANNO %in% diverse_stations_ids, ]

# initialize plot for the cycles
par(mar = c(4, 5, 3, 20), xpd = TRUE)
plot(
  1,
  xlab = "Day of the Year",
  ylab = "Expected Temperature (°C)",
  xlim = c(1, 365),
  ylim = c(-20, 30),
  type = 'n'
)

# iterate through the diverse station IDs to estimate and plot cycles
i <- 1
for (station_id in diverse_stations$WBANNO) {
  yearly_cycle <- get_yearly_cycle(station_id)
  points(
    x = yearly_cycle$day_of_year,
    y = yearly_cycle$avg_temp,
    type = "l",
    col = rainbow(10)[i],
    lwd = 2,
    lty = i
  )
  i = i + 1
}
par(cex = 0.8)
legend(
  x = "right",
  inset = c(-0.5, 0),
  title = "Station",
  legend = diverse_stations$`station name`,
  col = rainbow(10),
  lty = 1:10,
  pch = 20
)
```

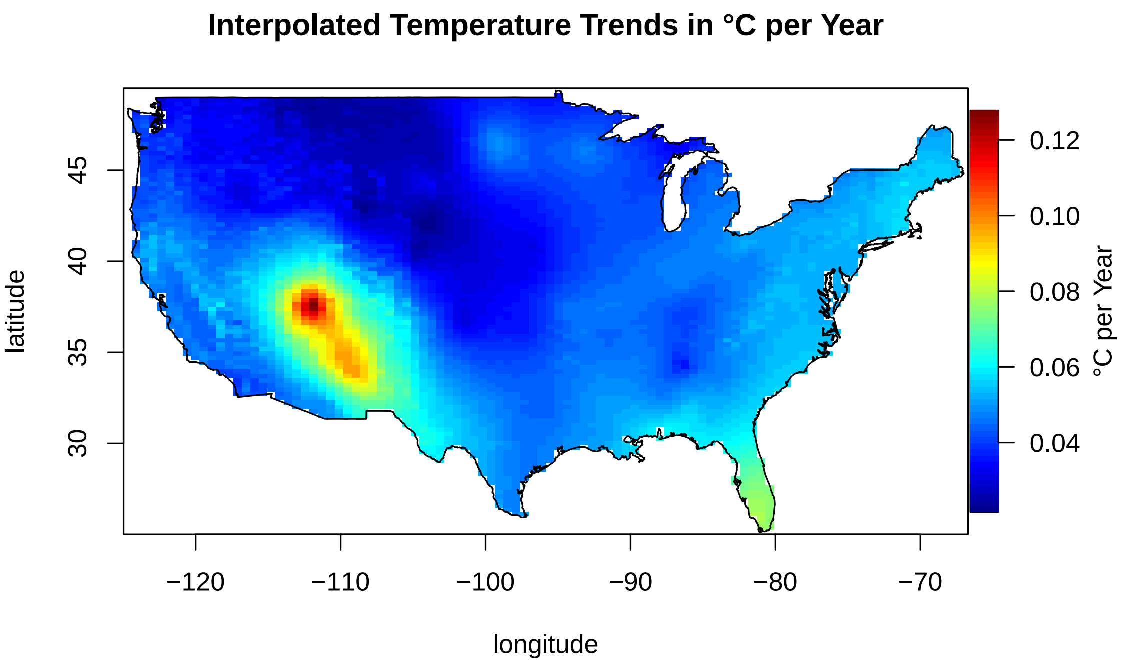


```
# estimating the yearly trend of US weather

# initialize empty vectors for coefficients, SEs, and significance
trend_coefs <- rep(NA, length = nrow(contig_stations))
```

```r
standard_errors <- rep(NA, length = nrow(contig_stations))
is_sig <- rep(F, length = nrow(contig_stations))

# iterate through all contiguous US stations
i <- 1
for (station_id in contig_stations$WBANNO) {
  station_trend = temperature_trend(station_id)
  trend_coefs[i] = station_trend$coefficients["years_elapsed", "Estimate"]
  standard_errors[i] = station_trend$coefficients["years_elapsed", "Std. Error"]
  is_sig[i] = station_trend$coefficients["years_elapsed", "Pr(>|t|)"] <= 0.05
  i = i + 1
}

# plot the stations with their trend coefficient and significance
maps::map("usa")
points(
  contig_stations$LONGITUDE,
  contig_stations$LATITUDE,
  pch = 16 + is_sig,
  col =  colmap(10)[cut(trend_coefs, 10)]
)
legend(
  x = -74,
  y = 40,
  title = "°C/yr Trend",
  legend = levels(cut(trend_coefs, 10)),
  col = colmap(10),
  pch = 20,
  xpd = TRUE
)
legend(
  x = -130,
  y = 30,
  title = "Significance",
  legend = c("Not Significant", "Significant"),
  col = "black",
  pch = c(16, 17),
  xpd = TRUE
)
title("Temperature Trends in °C per Year")
```



```
# determine the indices in the bottom 2/3 of standard errors
low_se_indices = order(standard_errors)[1:as.integer(length(standard_errors) * 2 /
                                                                          3)]

# interpolate and plot the bottom 2/3 of standard errors of trend coefs
trend_interp_data <- interpolate_data(trend_coefs[low_se_indices],
                                      contig_stations$LONGITUDE[low_se_indices],
                                      contig_stations$LATITUDE[low_se_indices],
                                      gridpoints = grid,
                                      use_elev=T)
#> 1 observations removed due to missingness or Inf
#> Assuming columns 1 and 2 of locs are (longitude,latidue) in degrees
plot_interpolations(trend_interp_data)
title("Interpolated Temperature Trends in °C per Year")
mtext("°C per Year", side = 4, line = 3.8)
```



For each station, a least squares regression was fit on the daily average temperature with the covariate years_elapsed, which represents the years elapsed since the start of 2000. $sin(2\pi * years\_elapsed)$, $cos(2\pi * years\_elapsed)$, $sin(4\pi * years\_elapsed)$, and $cos(4\pi * years\_elapsed)$ were added as well to the model's covariates to ignore influence of cyclic temperature patterns such as seasons. We did this to analyze only the year to year trends in weather as opposed to seasonal trend. Then, we used the coeffecient for the years_elapsed term to estimate the linear temperature trend over the years for each station.

Weather stations experiencing statistically significant changes in temperature per year are indicated with a triangle, as opposed to a circle, on the map. The method to determine statistical significance was a t-test. For each weather station in the contiguous US, this test was conducted with the null hypothesis being that there is no linear yearly temperature trend and the alternative hypothesis being that there is a linear yearly temperature trend. Each weather station then had a T statistic computed, this statistic is defined as $\frac{b_{years\_elapsed}}{SE_{b_{years\_elapsed}}}$, where $b_{years\_elapsed}$ is the station's average temperature trend per year coefficient in the linear model, $SE_{b_{years\_elapsed}}$) is the standard error of the coefficient. This T statistic is proven to be t distributed with (sample_size - na_samples - parameters) degrees of freedom under the null hypothesis. If this test statistic is beyond the middle $(1 - \alpha)$% of this t distribution, we reject the null hypothesis and conclude that the temperature trend is significant over time since the null hypothesis is very unlikely to see given the data. In this case, we use a 95% confidence level and see that most of the station have a statistically significant weather trend over time.

```
# compare our calculated weather trends to those from an reputable source

# according to https://www.weather.gov/media/slc/
# ClimateBook/Annual%20Average%20Temperature%20By%20Year.pdf
# the National Weather Service

# could not find a good source that had already estimated yearly trend in C

# manual input
years = 2004:2023
last_20_avg_F <-
  c(
```

```r
    52.1,
    53.4,
    53.8,
    53.8,
    51.9,
    52.1,
    52.7,
    51.8,
    56.6,
    53.3,
    55.6,
    56.3,
    56.2,
    56.1,
    56.2,
    53.6,
    55.7,
    56.3,
    55.9,
    55.6
  )

# convert from F to C to match our data
last_20_avg_C <- (last_20_avg_F - 32) / 1.8

# estimating the 20 year temperature trend
gov_cycle_lm <- lm(last_20_avg_C ~
                 years)

# the average temperature trend per year from our low SE data
data_avg_change <- round(mean(trend_coefs[low_se_indices]), 2)

# the predicted temperature trend per year from the government data
gov_avg_change <- round(summary(gov_cycle_lm)$coefficients["years", "Estimate"], 2)

print(paste("Weather data average trend:", data_avg_change))
#> [1] "Weather data average trend: 0.05"
print(paste("Government data average trend:", gov_avg_change))
#> [1] "Government data average trend: 0.11"
```

According to the National Weather Service data for the average US temperature over the last 20 years, the average change in temperature has been 0.11 degrees Celsius per year. Our data collection and analysis shows that the average change in temperature has been 0.05 degrees Celsius per year. The two estimates are close and may differ due to different locations for the collection of data. However, they both show a positive trend in the temperature in recent years, potentially due to human global warming,