SDS 439 - Final Project

Due May 7, 5:00 pm

Weather Data

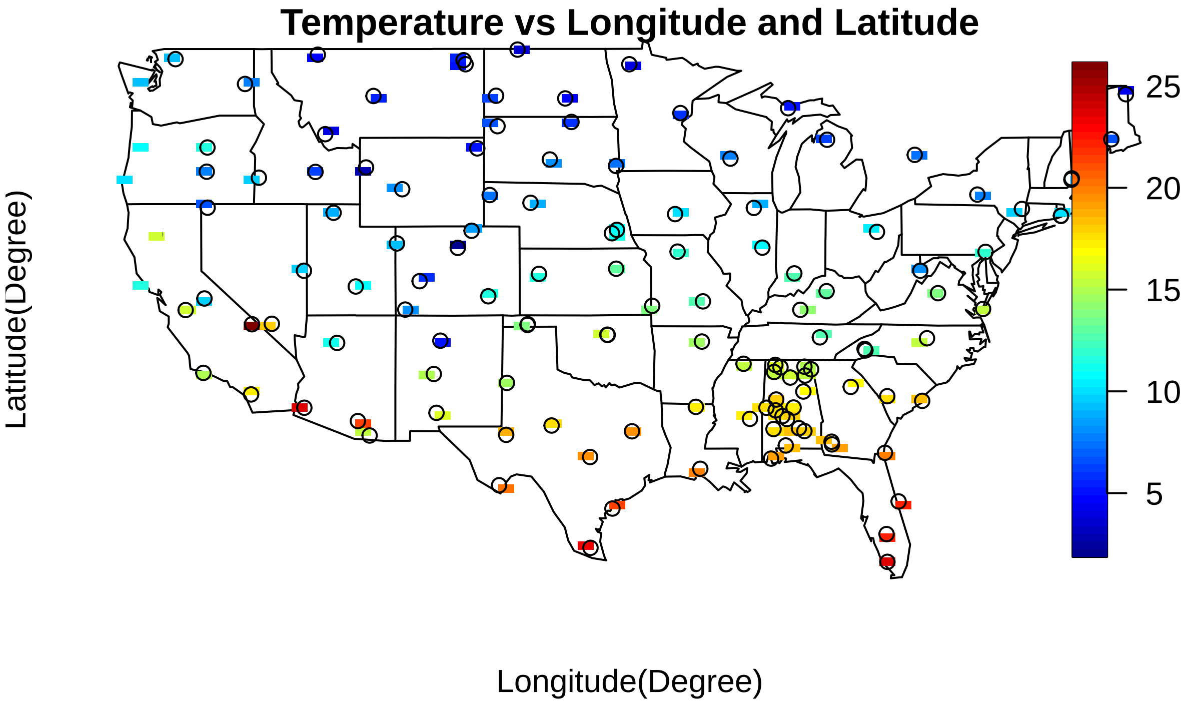
We have analyzed data from the US Climate Reference Network (USCRN) (https://www.ncei.noaa.gov/access/crn/) a few times this semester, looking at data from a handful of stations. Now, we'll do a more complete analysis of data from all of the lower-48 stations.

There is a file in the course github: `datasets/uscrn_daily.RData` which contains daily observations from 132 different sites.

1. As usual, start by plotting the data. I'll help by making one plot that uses the maps package to create a map. Make three more plots showing various features of the data, focusing on the `T_DAILY_AVG` variable, which will be our primary response of interest. One of the plots should explore how temperatures vary over the course of the year.

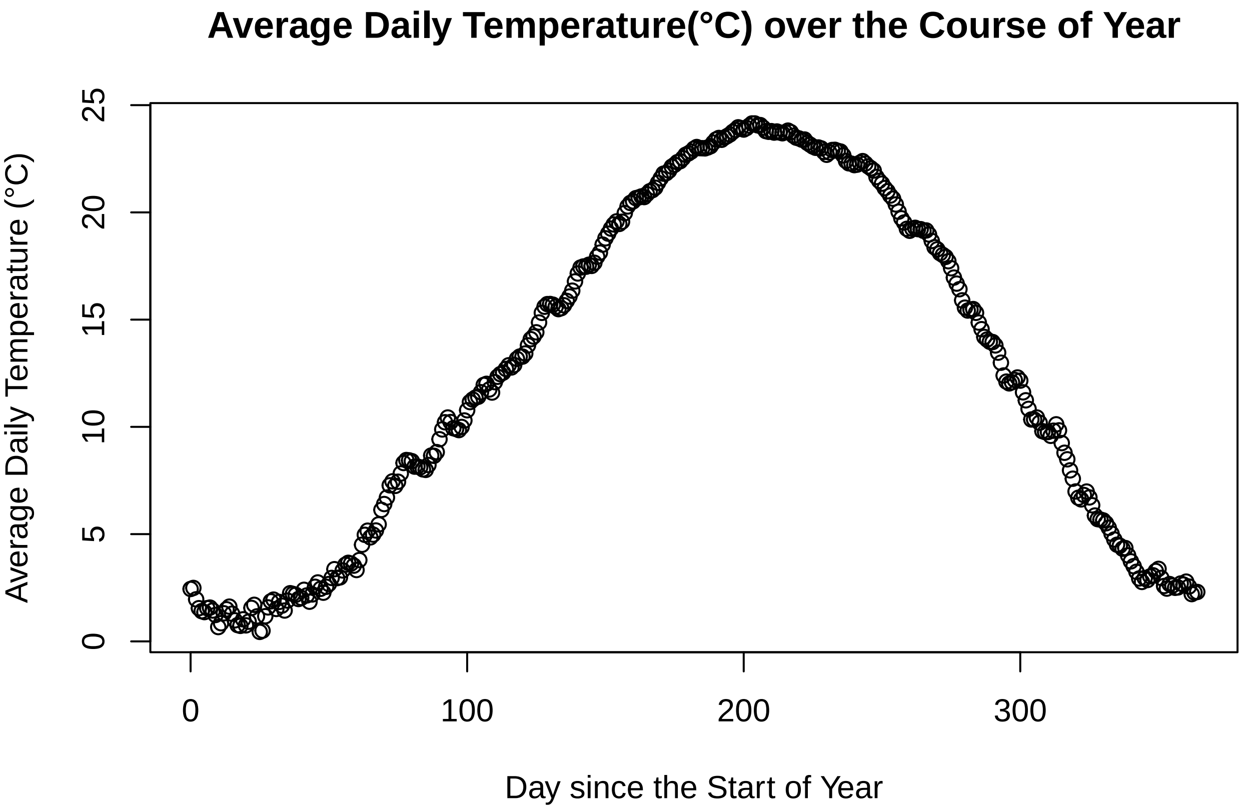
```
load("/Users/jackyang/Documents/Homework/2025 Spring/SDS 439/linear_models_sp25/datasets/uscrn_daily.RDa
library("maps")

site_locs <- unique( dat[c("LONGITUDE","LATITUDE")] )
site_levels <- unique(dat$site)

ii <- dat$T_DAILY_AVG == -9999.0
dat$T_DAILY_AVG[ii] <- NA

# First graph: Average Temperature vs Longitude and Latitude
temp_loc <- data.frame(
  site = character(),
  logitude = numeric(),
  latitude = numeric(),
  temp = numeric()
)
for (level in site_levels) {
  temp <- mean(dat$T_DAILY_AVG[dat$site == level], na.rm = TRUE)

  new_row <- data.frame(
    site = level,
    logitude = dat$LONGITUDE[dat$site == level][1],
    latitude = dat$LATITUDE[dat$site == level][1],
    temp = temp
  )

  temp_loc <- rbind(temp_loc, new_row)
}

map("state")
mtext("Longitude(Degree)", side = 1, line = 2)
mtext("Latitude(Degree)", side = 2, line = 2)
```

```
title("Temperature vs Longitude and Latitude", line = 0)

xlim <- c(min(temp_loc$logitude, na.rm = TRUE)-5,
          max(temp_loc$logitude, na.rm = TRUE)+5)
ylim <- c(min(temp_loc$latitude, na.rm = TRUE)-10,
          max(temp_loc$latitude, na.rm = TRUE)+10)
zlim <- c(min(temp_loc$temp, na.rm = TRUE),
          max(temp_loc$temp, na.rm = TRUE))

fields::quilt.plot(
    temp_loc$logitude, temp_loc$latitude, temp_loc$temp,
    xlim = xlim,
    ylim = ylim,
    zlim = zlim,
    add = TRUE
)
points( site_locs$LONGITUDE, site_locs$LATITUDE )
```



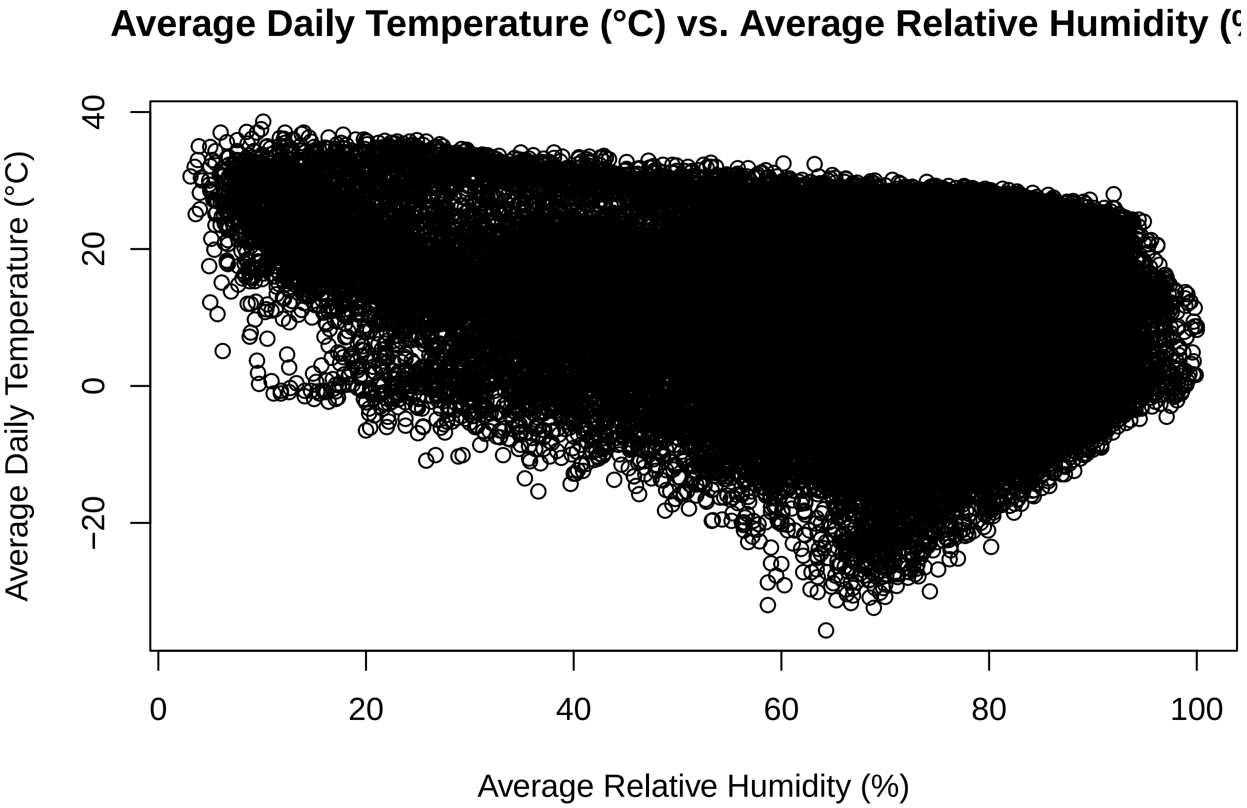
```
# Finding: The northern states will have a lower average daily temperature than
# southern states.Eastern and western states are generally warmer than the
# central states.

# Second graph: Average Daily Temperature over the Year
dat$num_date <- as.Date(as.character(dat$LST_DATE), format = "%Y%m%d")
min_date <- as.Date("2000-01-01")
dat$doy <- as.numeric(dat$num_date - min_date) %% 365
dat_avg <- data.frame(
  doy = numeric(),
  avg_temp = numeric()
)
```

```r
for (i in 0:364) {
  avg_temp <- mean(dat$T_DAILY_AVG[dat$doy==i], na.rm = TRUE)
  new_row <- data.frame(
    doy = i,
    avg_temp = avg_temp
  )

  dat_avg <- rbind(dat_avg, new_row)
}

plot(dat_avg$doy, dat_avg$avg_temp,
     xlab = "Day since the Start of Year",
     ylab = "Average Daily Temperature (°C)",
     main = "Average Daily Temperature(°C) over the Course of Year"
)
```



```r
# Finding: The general trend of average daily temperature across America follows
# the general trend of the weather cycle.

# Third graph: Average Daily Temperature vs. Average Relative Humidity
ii <- dat$RH_DAILY_AVG == -9999.0
dat$RH_DAILY_AVG[ii] <- NA

# I ask ChatGPT to help me choose 10 best representative sites in US
selected_sites <- c("NY_Ithaca_13_E", "FL_Sebring_23_SSE", "IL_Champaign_9_SW",
                    "CO_Boulder_14_W", "AZ_Tucson_11_W", "WA_Spokane_17_SSW",
                    "CA_Santa_Barbara_11_W", "KS_Manhattan_6_SSW",
                    "MN_Goodridge_12_NNW", "AK_Fairbanks_10_N")
```

```r
ii <- (!is.na(dat$RH_DAILY_AVG) & !is.na(dat$T_DAILY_AVG)
       & dat$site %in% selected_sites)
plot(x = dat$RH_DAILY_AVG[ii],
     y = dat$T_DAILY_AVG[ii],
     xlab = "Average Relative Humidity (%)",
     ylab = "Average Daily Temperature (°C)",
     main = "Average Daily Temperature (°C) vs. Average Relative Humidity (%)")
```



```r
# Although the data cloud is so dense, we can still find the general trend such
# that hgiher relative humidity will tend to have lower average daily
# temperature. This makes sense because we would expect raining to lower the
# temperature of that day and the following days.
```

2. Fix all of the missing values, and create a `date` column in the dataframe, using the `as.Date` function and the `LST_DATE` column. Also create a `days` column that is equal to the number of days since `2020-01-01`. For reference, see the documentation at https://www.ncei.noaa.gov/pub/data/uscrn/products/daily01/readme.txt

```r
ii <- dat$T_DAILY_AVG == -9999
dat$T_DAILY_AVG[ii] <- NA

dat$LST_DATE <- as.character(dat$LST_DATE)
dat$date <- as.Date(dat$LST_DATE, format = "%Y%m%d")
dat$days <- as.numeric((dat$date - as.Date('2020-01-01')))
```

3. Notate the data as follows: Let i refer to the row of the dataset, y_i the daily average, d_i the number of days since `2020-01-01`, and $j(i)$ be the site.

   We will be fitting the following model to the responses:

$$Y_i = b_{0,j(i)} + b_{1,j(i)} d_i + b_{2,j(i)} \sin\left(\frac{2\pi d_i}{365.25}\right) + b_{3,j(i)} \cos\left(\frac{2\pi d_i}{365.25}\right) + b_{4,j(i)} \sin\left(\frac{4\pi d_i}{365.25}\right) + b_{5,j(i)} \cos\left(\frac{4\pi d_i}{365.25}\right) + \varepsilon_i$$

$$\varepsilon_i \overset{ind}{\sim} N(0, \sigma^2_{j(i)})$$

where $b_{3,j(i)}$ just means that each site has a separate regression coefficient.

How do you interpret the intercept $b_{0,j(i)}$? Think carefully because it's actually not possible to set all of the covariates equal to zero at the same time.

```
# Interpretation:
# The mean "average daily temperature" around a fixed interval like half an
# year before 2020-01-01 and half an year after 2020-01-01. This is because
# the odd property of b2sinx, b4sin2x, and b1x and also the symmetric property
# of b3cosx and b5cos2x. When integrate those terms from -a*pi to a*pi, the
# result will be 0. In this specific example, when we take the integral from
# 2019-01-01 to 2021-01-01, all terms except for b0 will be 0. And b0 will be
# integrated to 730b0. If we divide this term by number of days between the
# integral, we can extract the b0 term. So, b0 will represent the mean mean
# "average daily temperature" between this selected time interval at the
# location at level j(i).
```

How do you interpret $b_{1,j(i)}$?

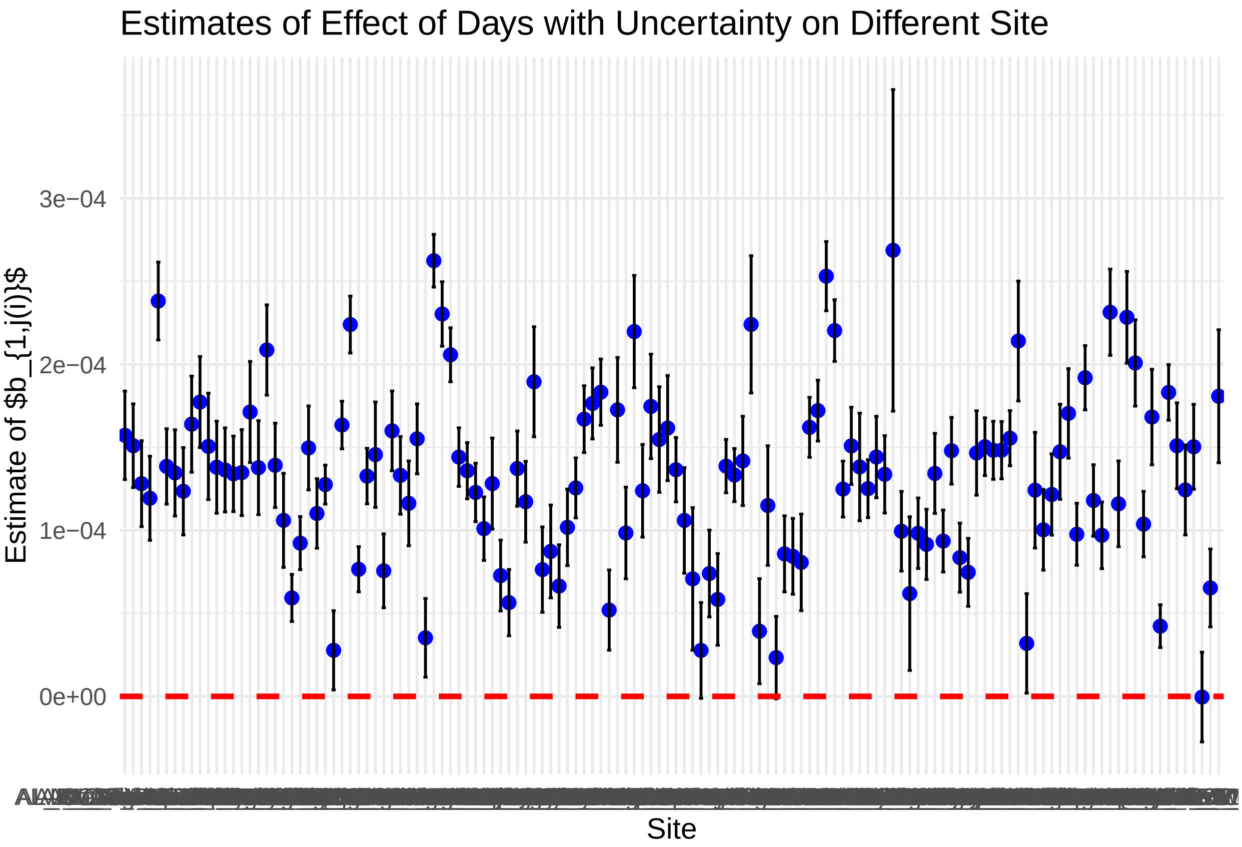
```
# Interpretation: $b_{1,j(i)}$ is the effect of the day count since 2020-01-01
# on the Mean air temperature of the site at level j(i). The sign of this will
# control the overall trend of the model. If b1 is positive, it means the
# daily average temperature slightly increases as the year goes. If negative,
# it means the daily average temperature slightly decreases across different
# years. If zero, it means the temperature is always kept at the stable level
# and have no change over the long time period.
```

4. Fit this model separately to each site, and store the results. (this is a short prompt, but it will require some coding).

```
dat$sin <- sin(2*pi*dat$days/365.25)
dat$cos <- cos(2*pi*dat$days/365.25)
dat$sin2 <- sin(4*pi*dat$days/365.25)
dat$cos2 <- cos(4*pi*dat$days/365.25)

models <- list()
model_summaries <- list()

for (level in site_levels) {

  site_data <- subset(dat, dat$site == level)

  m1 <- lm(T_DAILY_AVG ~ days + sin + cos + sin2 + cos2, data = site_data)

  models[[level]] <- m1

  model_summary <- summary(m1)

  model_summaries[[level]] <- model_summary
}
```

5. Make a plot summarizing the results of estimating $b_{1,j}$. Your plot should include information about the estimates and their uncertainties, and it should be visually informative. Make sure that outliers are not dominating the plot.

```r
b1_estimates <- list()
b1_se <- list()
for (site in site_levels) {
  b1_estimates[site] <- model_summaries[[site]]$coefficients["days", "Estimate"]
  b1_se[site] <- model_summaries[[site]]$coefficients["days", "Std. Error"]
}

b1_data <- data.frame(
  site = names(b1_estimates),
  estimate = unlist(b1_estimates),
  se = unlist(b1_se)
)

b1_data <- na.omit(b1_data)

library(ggplot2)

# Create the plot with error bars (standard errors)
ggplot(b1_data, aes(x = site,
  y = estimate, ymin = estimate - se, ymax = estimate + se)) +
  geom_point(color = "blue", size = 2) +
  geom_errorbar(width = 0.5) +
  geom_hline(yintercept = 0, linetype = "dashed", color = "red", linewidth = 1)+
  labs(x = "Site", y = "Estimate of $b_{1,j(i)}$",
  title = "Estimates of Effect of Days with Uncertainty on Different Site") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  theme_minimal()
```



6. From your analysis of $b_{1,j(i)}$, what can you conclude about long term trends in temperature in the continental US? Be specific about the size of the effects and their uncertainties.

```
mean(unlist(b1_estimates), na.rm = TRUE)
```

```
## [1] 0.0001321737
```

```
# From the plot above, we can observe that the range of effect of the days on
# average daily temperature is between the 0 and 0.0003. Almost  all the
# estimates with the error bar are above 0, which indicates that there is a
# slightly increasing trend on the average daily temperature across
# the years on most sites in America. This means, overall, the temperature is
# increasing as the year goes in America.
# There are one site that have nearly zero estimate of b1. This indicates that
# the average daily temperature is quite stable on those places. There is no
# trend of change.
# So, overall, the temperature has an increasing trend in continental America.
# The effect of days increasing on temperature is on average 0.000132 for the
# continental America.
```

7. Comment on any problems with the model, how they might be influencing the results, and how you might address the problems.

```
# Periodicity:
# This model assumes the periodic effect on the weather to be fixed every year.
# However, there may be some unexpected influence on the weather cycle, such as
# lagged wind and forest fire. Assuming the fixed periodic weather cycle may not
# capture all the trends on the daily average temperatures.
# Solution: We can use a even higher order function to capture the non-fixed
# periodic effects. Also we can add more covariate into account rather than
# the days only.


# Site effect:
# This model assumes there is no interaction across different sites on the
# weather as we fit the model for each site independently. However, there is
# potentially strong influence on the average temperature by different sites.
# For example, if one state becomes colder on a certain year, it is more likely
# the nearby states will also be somewhat cooler than before. So, we shouldn't
# ignore the interaction across different sites.

# Solution: We can treat the site as a factor and make a factor-numeric model
# with interaction between site and days. This may take into account that the
# site will have some effects on the average daily temperature. But in this way,
# the model is single and too complicated. We need to find a better way or do
# different analysis of the site effects on the temperature.


# Complexity of the Model:
# We not only fit the general sinx+cosx periodic trend into the model but also
# the higher order periodic function sin2x+cos2x. This makes our model quite
# complicated. This may result in the overfitting problem, which the model is
# so powerful that it takes the outliers into account as well.

# Solution: We can try to compare the difference between sinx+cosx model,
# sin2x+cos2x model, and the full model to see which one has better performance
# such as residual standard error.
```

8. For each site, calculate the following quantites based on your fitted models

- Amplitude: difference in temperature between warmest and coldest day
- Day number with coldest temperature: (1-365)
- Day number with warmest temperature: (1-365)
- Daily standard deviation: how much actual temperatures tend to differ from the expected temperature.

  You can accomplish this in various ways, but a relatively straightforward way is to create a data frame with a row for each day of the year, and use your fitted model for each site and the `predict` function to make a prediction of temperature on each day in the data frame.

```
site_summaries <- data.frame(
  site = character(),
  amplitude = numeric(),
  coldest_day = numeric(),
  warmest_day = numeric(),
  daily_sd = numeric(),
  daily_sd_summary = numeric(),
  stringsAsFactors = FALSE
)
site_levels <- names(model_summaries)

days_of_year <- data.frame(
  days = 1:365,
  sin = sin(2 * pi * (1:365) / 365.25),
  cos = cos(2 * pi * (1:365) / 365.25),
  sin2 = sin(4 * pi * (1:365) / 365.25),
  cos2 = cos(4 * pi * (1:365) / 365.25)
)

predicted_values <- list()

for (level in site_levels) {

  fitted_values <- predict(models[[level]], newdata = days_of_year)

  predicted_values[[level]] <- fitted_values

  coldest_day <- which.min(fitted_values)
  warmest_day <- which.max(fitted_values)

  amplitude <- max(fitted_values) - min(fitted_values)

  site_data <- subset(dat, dat$site == level)

  residuals <- site_data$T_DAILY_AVG -
    predict(models[[level]], newdata = site_data)

  daily_sd <- sd(residuals, na.rm = TRUE)

  daily_sd_summary <- model_summaries[[level]]$sigma

  new_row <- data.frame(
    site = level,
    amplitude = amplitude,
    coldest_day = coldest_day,
    warmest_day = warmest_day,
    daily_sd = daily_sd,
    daily_sd_summary = daily_sd_summary
  )

  site_summaries <- rbind(site_summaries, new_row)
}
```

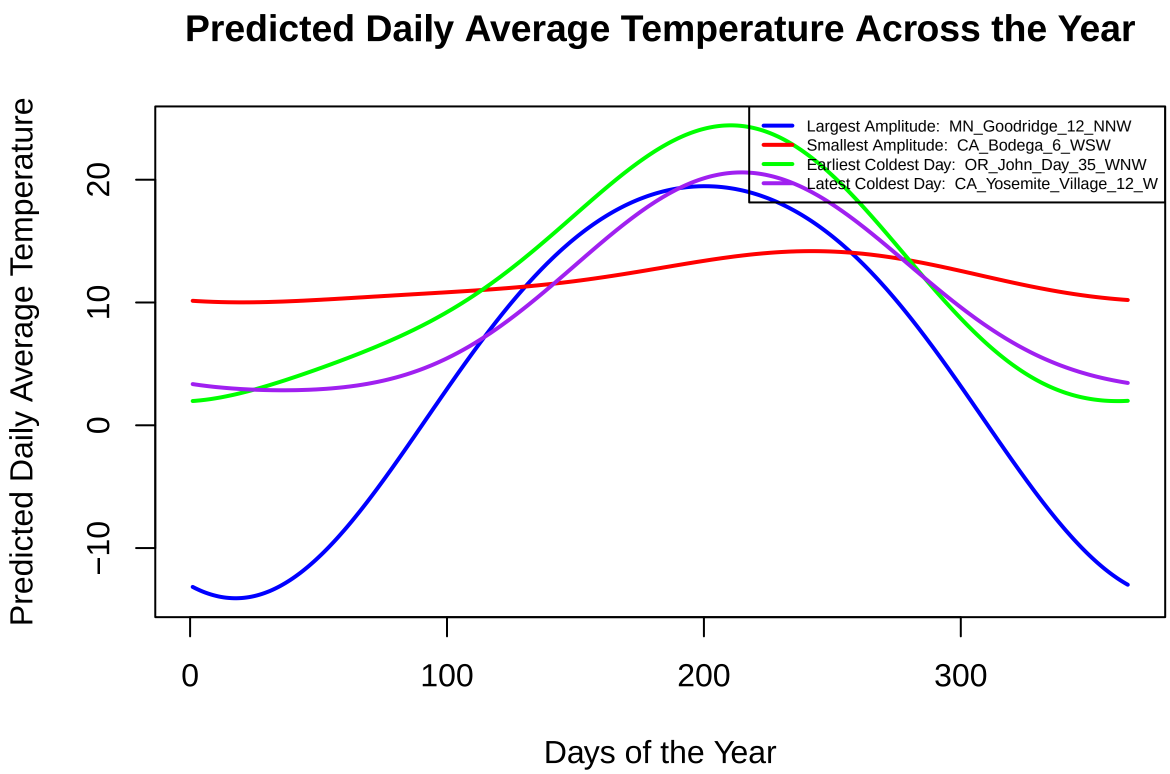
9. Plot the yearly cycles for the following sites, making sure to label your plots:

- the site with the largest amplitude
- the site with the smallest amplitude
- the site with the earliest coldest day in the winter season
- the site with the latest coldest day in the winter season

  For the coldest day, you'll need to deal with the fact that December 31 is earlier in the winter season than January 1. Try to get all of the yearly cycles on the same plot, if it looks ok.

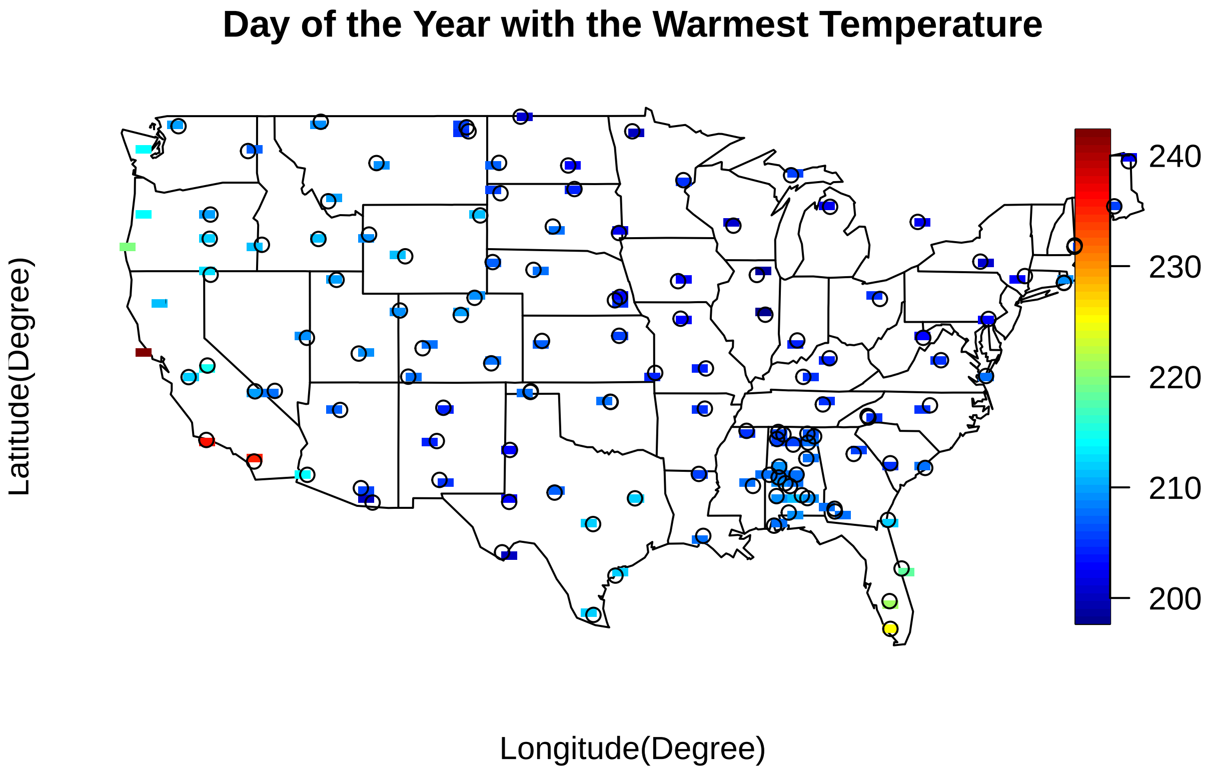
```
site_largest_amplitude <-
  site_summaries$site[which.max(site_summaries$amplitude)]

site_smallest_amplitude <-
```

```r
  site_summaries$site[which.min(site_summaries$amplitude)]

early_winter_session <- site_summaries[site_summaries$coldest_day > 182,]
late_winter_session <- site_summaries[site_summaries$coldest_day < 182,]

site_earliest_coldest_day <-
  early_winter_session$site[which.min(early_winter_session$coldest_day)]

site_latest_coldest_day <-
  late_winter_session$site[which.max(late_winter_session$coldest_day)]

plot(x = days_of_year$days,
     y = predict(models[[site_largest_amplitude]], newdata = days_of_year),
     xlab = "Days of the Year",
     ylab = "Predicted Daily Average Temperature",
     main = "Predicted Daily Average Temperature Across the Year",
     type = "l",
     col = "blue",
     ylim = range(c(predicted_values[[site_largest_amplitude]],
                    predicted_values[[site_smallest_amplitude]],
                    predicted_values[[site_earliest_coldest_day]],
                    predicted_values[[site_latest_coldest_day]])),
     lwd = 2)

lines(x = days_of_year$days,
      y = predicted_values[[site_smallest_amplitude]],
      col = "red",
      lwd = 2)

lines(x = days_of_year$days,
      y = predicted_values[[site_earliest_coldest_day]],
      col = "green",
      lwd = 2)

lines(x = days_of_year$days,
      y = predicted_values[[site_latest_coldest_day]],
      col = "purple",
      lwd = 2)

legend("topright",
       legend = c(paste("Largest Amplitude: ", site_largest_amplitude),
                  paste("Smallest Amplitude: ", site_smallest_amplitude),
                  paste("Earliest Coldest Day: ", site_earliest_coldest_day),
                  paste("Latest Coldest Day: ", site_latest_coldest_day)),
       col = c("blue", "red", "green", "purple"),
       lwd = 2, cex = 0.5)
```



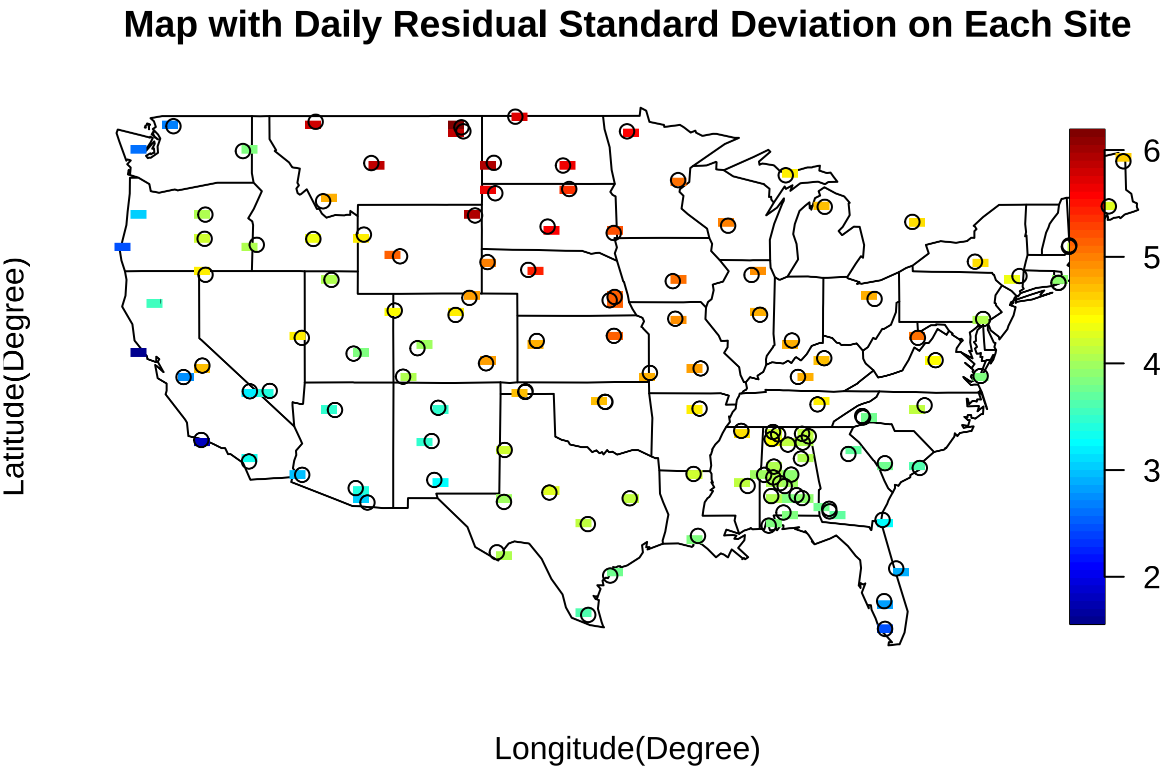
10. Make a map showing the day of year with the warmest temperature. You might try combining the `fields::quilt.plot` function with the `maps::map` function.

```r
map("state")
mtext("Longitude(Degree)", side = 1, line = 2)
mtext("Latitude(Degree)", side = 2, line = 2)
title("Day of the Year with the Warmest Temperature")

for (i in 1:nrow(site_summaries)) {
  site <- site_summaries$site[i]
  warmest_day <- site_summaries$warmest_day[i]
  site_summaries$longitude[i] <- dat$LONGITUDE[dat$site == site][1]
  site_summaries$latitude[i] <- dat$LATITUDE[dat$site == site][1]
}

zlim <- c(min(site_summaries$warmest_day, na.rm = TRUE),
          max(site_summaries$warmest_day, na.rm = TRUE))

fields::quilt.plot(
  site_summaries$longitude, site_summaries$latitude, site_summaries$warmest_day,
  xlim = xlim,
  ylim = ylim,
  zlim = zlim,
  add = TRUE
)

points( site_summaries$longitude, site_summaries$latitude )
```



11. Make a map showing the daily residual standard deviation. Where in the U.S. do the temperatures deviate most from expected?

```
map("state")
mtext("Longitude(Degree)", side = 1, line = 2)
mtext("Latitude(Degree)", side = 2, line = 2)
title("Map with Daily Residual Standard Deviation on Each Site")

zlim <- c(min(site_summaries$daily_sd, na.rm = TRUE),
          max(site_summaries$daily_sd, na.rm = TRUE))

fields::quilt.plot(
  site_summaries$longitude, site_summaries$latitude, site_summaries$daily_sd,
  xlim = xlim,
  ylim = ylim,
  zlim = zlim,
  add = TRUE
)

points( site_summaries$longitude, site_summaries$latitude )
```



```
# In the northern central area of America, the weather temperature varies most
# from the expected value of the temeprature. This is probably because the
# wind from the northern country will significantly influence the temperature.
# In the eastern and western area, the current will make the temperature quite
# stable. In the southern area, most of the states are in tropical area. The
# temperature is stable and high across the whole year.
```