Simulate of Gestational Periods (Binomial), Conception Dates (Mixture), and Temperature Exposures (Sine wave and AR(1)) during Pregnancy

Fan Wang

2021-09-29

Contents

1 Conception, Birth and Extreme Temperature **1**
1.1 Gestational Age at Birth Distribution (Binomial) 1
1.2 Conception Dates Distribution (Mixture Binomials + Random) 4
1.3 Temperature Process Sine Wave and AR(1) 6
1.4 Distributions of Conception, Birth, Temperature and Pre-Term 11
1.4.1 Conception and Birth Marginal and Joint Distributions 11
1.4.2 Extreme Temperature and Pre-Term Births 11
1.4.3 Days and Percent of Gestational Days Exposed to Cold 12
1.5 Simulating Datasets on Extreme Temperature Exposures Across Pregnancies 12
1.5.1 Compute Extreme Temperature Exposure 12
1.5.2 Compute Extreme Temperature Exposure 19
1.5.3 Visualize Pre-term, Full-term and Extreme Temperature Exposures 22

1 Conception, Birth and Extreme Temperature

The various simulations on this page are done in support of Same Environment, Stratified Impacts? Air Pollution, Extreme Temperatures, and Birth Weight in South China.

> Go to the **RMD**, **R**, **PDF**, or **HTML** version of this file. Go back to fan's REconTools Package, R Code Examples Repository (bookdown site), or Intro Stats with R Repository (bookdown site).

1.1 Gestational Age at Birth Distribution (Binomial)

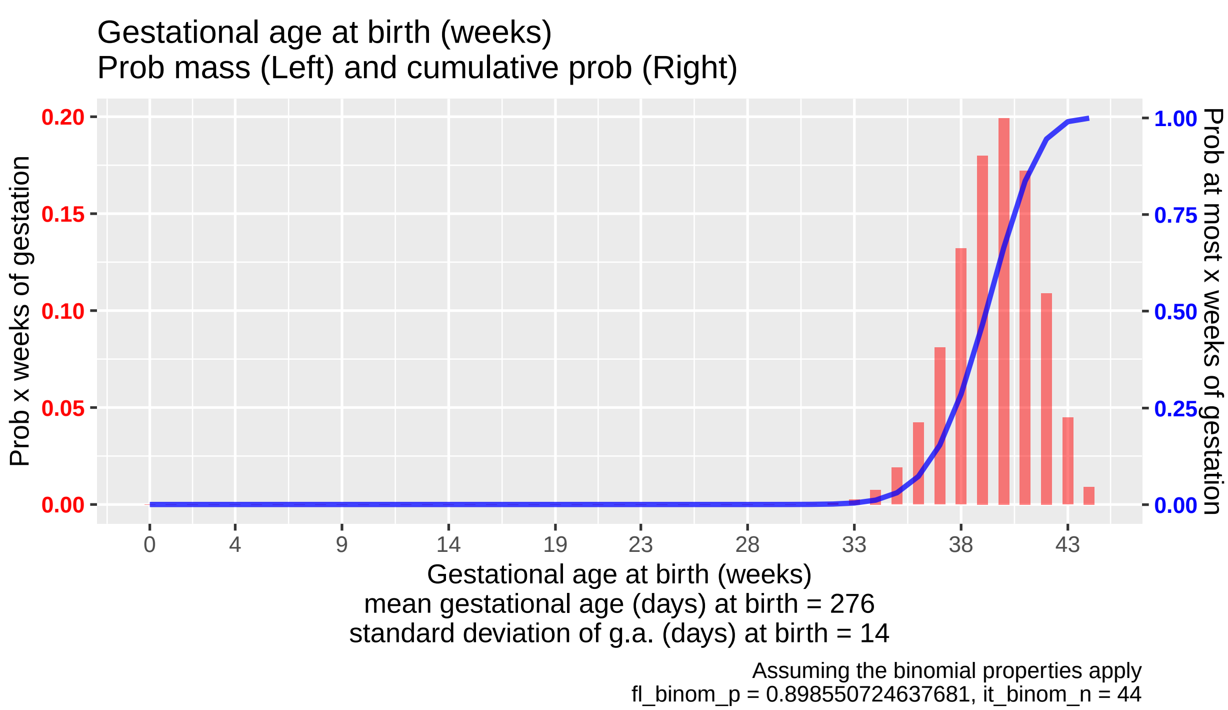
Suppose the number of weeks that an individual is pregnant follows the binomial distributions. According to data from the Right from the Start study from West Virginia, published in Hoffman et al. (2008), median gestational age at birth was 276/7 weeks and the standard deviation was around 14/7 weeks. We use binomial to approximate normal rules to generate the relevant binomial parameters.

On each date after conception, there is some chance to give birth, due to various purely random factors, denote this chance as $p_{\tau,i}$, where τ is discrete time since conception, $p_{\tau,i}$ follows the just defined binomial distribution. The distribution might or might not be individual specific. They would be individual-specific if individual experiences during the course of pregnancy, such as exposures to extreme temeprature, matter for gestational age at birth. The probability that an individual i gives birth during the week of conception is $p_{0,i}$. Conditional on proceeding to the day after conception, the probability of giving birth is $\frac{p_{1,i}}{1-p_{0,i}}$. Similarly for future dates. The conditional birth probability is the Hazard rate in this scenario.

Below, we formulate the gestational age at birth distribution function *ffi_gestation_age_at_birth_dist()*.

```r
# This function generates the distribution of gestational weeks at birth
ffi_gestation_age_at_birth_dist <- function(
  mu_gabirth_days = 276,
  sd_gabirth_days = 14
){
  # # Parameters
  # # gabirth = gestational age at birth
  # mu_gabirth_days <- 276
  # sd_gabirth_days <- 14
  # from https://fanwangecon.github.io/R4Econ/statistics/discrandvar/htmlpdfr/fs_disc_approx_cts.html
  it_binom_n <- round((mu_gabirth_days / 7)^2 / (mu_gabirth_days / 7 - (sd_gabirth_days / 7)^2))
  fl_binom_p <- 1 - (sd_gabirth_days / 7)^2 / (mu_gabirth_days / 7)

  # Same graphing code as from: https://fanwangecon.github.io/Stat4Econ/probability_discrete/htmlpdfr/b
  # Generate Data
  ar_grid_gabirth <- 0:it_binom_n
  ar_pdf_gabirth <- dbinom(ar_grid_gabirth, it_binom_n, fl_binom_p)
  ar_cdf_gabirth <- pbinom(ar_grid_gabirth, it_binom_n, fl_binom_p)
  df_dist_gabirth <- tibble(gabirth = (ar_grid_gabirth), prob = ar_pdf_gabirth, cum_prob = ar_cdf_gabir

  # Two axis colors
  axis_sec_ratio <- max(ar_cdf_gabirth) / max(ar_pdf_gabirth)
  right_axis_color <- "blue"
  left_axis_color <- "red"

  # Probabilities
  plt_dist_gabirth <- df_dist_gabirth %>%
    ggplot(aes(x = gabirth)) +
    geom_bar(aes(y = prob),
      stat = "identity", alpha = 0.5, width = 0.5, fill = left_axis_color
    )

  # Cumulative Probabilities
  plt_dist_gabirth <- plt_dist_gabirth +
    geom_line(aes(y = cum_prob / axis_sec_ratio),
      alpha = 0.75, size = 1, color = right_axis_color
    )

  # Titles Strings etc
  graph_title <- paste0("Gestational age at birth (weeks)\n",
      "Prob mass (Left) and cumulative prob (Right)")
  graph_caption <- paste0("Assuming the binomial properties apply\n",
      "fl_binom_p = ", fl_binom_p, ", it_binom_n = ", it_binom_n)
  graph_title_x <- paste0("Gestational age at birth (weeks)\n",
    "mean gestational age (days) at birth = ", mu_gabirth_days, "\n",
    "standard deviation of g.a. (days) at birth = ", sd_gabirth_days)
  graph_title_y_axisleft <- "Prob x weeks of gestation"
  graph_title_y_axisright <- "Prob at most x weeks of gestation"

  # Titles etc
  plt_dist_gabirth <- plt_dist_gabirth +
    labs(
      title = graph_title,
```

```
      x = graph_title_x,
      y = graph_title_y_axisleft,
      caption = graph_caption
    ) +
    scale_y_continuous(
      sec.axis =
        sec_axis(~ . * axis_sec_ratio, name = graph_title_y_axisright)
    ) +
    scale_x_continuous(
      labels = ar_grid_gabirth[floor(seq(1, it_binom_n, length.out = 10))],
      breaks = ar_grid_gabirth[floor(seq(1, it_binom_n, length.out = 10))]
    ) +
    theme(
      axis.text.y = element_text(face = "bold"),
      axis.text.y.right = element_text(color = right_axis_color),
      axis.text.y.left = element_text(color = left_axis_color)
    )

  # Print
  return(list(
    df_dist_gabirth=df_dist_gabirth,
    plt_dist_gabirth=plt_dist_gabirth
  ))
}
# Test the function
ls_gsbirth <- ffi_gestation_age_at_birth_dist(mu_gabirth_days = 276, sd_gabirth_days = 14)
# Figure
print(ls_gsbirth$plt_dist_gabirth)
```



```r
# Table
df_dist_gabirth <- ls_gsbirth$df_dist_gabirth
kable(df_dist_gabirth %>% filter(prob >= 0.01)) %>% kable_styling_fc()
```

gabirth	prob	cum_prob
35	0.0190913	0.0303855
36	0.0422736	0.0726591
37	0.0809563	0.1536154
38	0.1320866	0.2857020
39	0.1799862	0.4656882
40	0.1992704	0.6649586
41	0.1721918	0.8371504
42	0.1089377	0.9460881
43	0.0448780	0.9909661

Some research on the distribution of gestational ages at birth:

- Gage (2002)
- Nassar et al. (2013)
- Lei et al. (2016)
- Pereira et al. (2021)

1.2 Conception Dates Distribution (Mixture Binomials + Random)

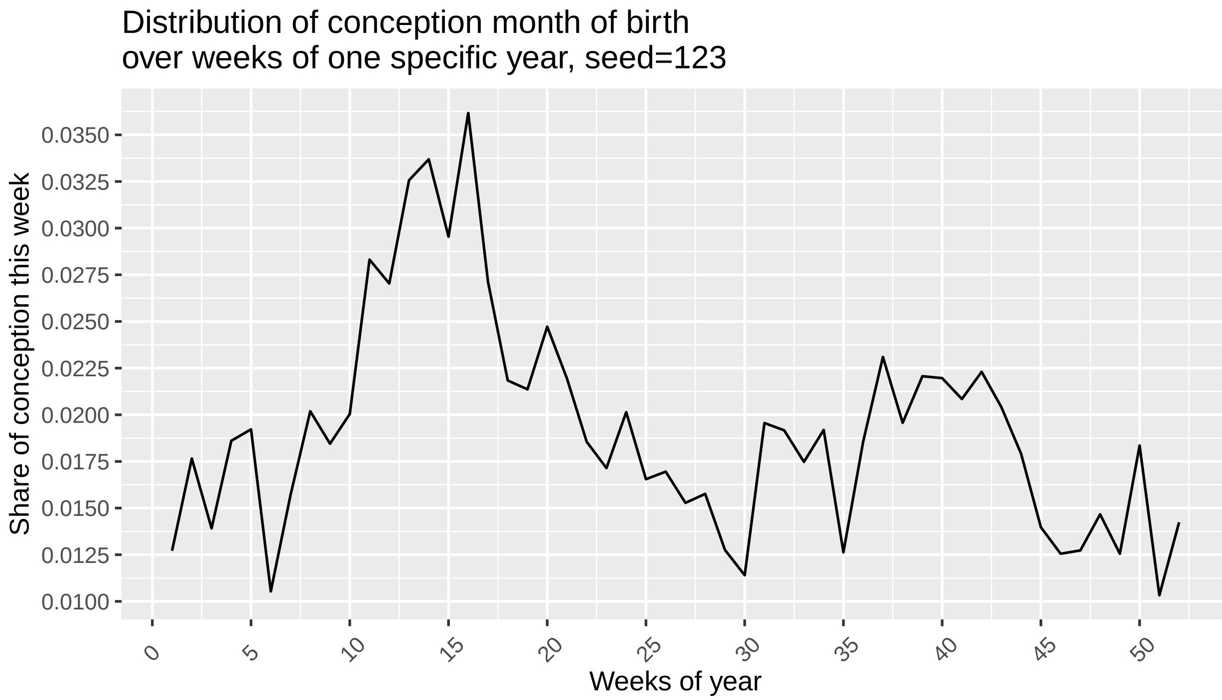
Denote week of the year with w. There are potentially 52 possible weeks of the year, starting at week 1, $w = 1$, and ending with $w = 52$. Conception can happen at any month.

Suppose the conception week of the year distribution is bimodal, we will pick two max conception count weeks, out of 52 weeks. Suppose week 15 and week 40 are two peak conception months, with week 15 more heavily weighted. What is the distribution of conception weeks? We will mix two binomial distribution with a largely-uniform distribution together. We will assume later that the day (of the week) of conception is uniformly distributed.

Below, we formulate the conception date distribution function *ffi_concept_distribution_year()*.

```r
# This function generates the distribution of conception weeks across
# the year, with two peak months, and some randomness
ffi_concept_distribution_year <- function(it_max_weeks = 52,
                                          it_peak_wk_1st = 15,
                                          it_peak_wk_2nd = 40,
                                          fl_binom_1st_wgt = 0.150,
                                          fl_binom_2nd_wgt = 0.075,
                                          it_runif_seed = 123) {
  # # Peak (local) months and weights
  # it_max_weeks <- 52
  # it_peak_wk_1st <- 15
  # it_peak_wk_2nd <- 40

  # # Weights for the two binomial and the remaining weight is for an uniform distribution
  # fl_binom_1st_wgt <- 0.25
  # fl_binom_2nd_wgt <- 0.10

  # Discrete random variables
  ar_fl_binom_1st <- dbinom(
    0:(it_max_weeks - 1), (it_max_weeks - 1),
```

```
    (it_peak_wk_1st - 1) / (it_max_weeks - 1)
  )
  ar_fl_binom_2nd <- dbinom(
    0:(it_max_weeks - 1), (it_max_weeks - 1),
    (it_peak_wk_2nd - 1) / (it_max_weeks - 1)
  )
  set.seed(it_runif_seed)
  ar_random_base <- runif(it_max_weeks, min = 0.5, max = 1)
  ar_random_base <- ar_random_base / sum(ar_random_base)

  # Mix two binomials and a uniform
  ar_fl_p_concept_week <- ar_fl_binom_1st * fl_binom_1st_wgt +
    ar_fl_binom_2nd * fl_binom_2nd_wgt +
    ar_random_base * (1 - fl_binom_1st_wgt - fl_binom_2nd_wgt)

  # Dataframe
  df_dist_conception <- tibble(conception_calendar_week = 1:it_max_weeks,
                               conception_prob = ar_fl_p_concept_week)

  # Line plot
  # Title
  st_title <- paste0(
      "Distribution of conception month of birth\n",
      "over weeks of one specific year, seed=", it_runif_seed
    )

  # Display
  plt_concept_week_of_year <- df_dist_conception %>%
    ggplot(aes(x = conception_calendar_week, y= conception_prob)) +
    geom_line() +
    labs(
      title = st_title,
      x = 'Weeks of year',
      y = 'Share of conception this week'
      ) +
    scale_x_continuous(n.breaks = 12) +
    scale_y_continuous(n.breaks = 10) +
    theme(
      axis.text.x = element_text(angle = 45, vjust = 0.1, hjust = 0.1)
    )

  # Return
  return(list(
    df_dist_conception = df_dist_conception,
    plt_concept_week_of_year = plt_concept_week_of_year
  ))
}
# Call function with defaults
ls_concept <- ffi_concept_distribution_year(it_max_weeks = 52,
                                            it_peak_wk_1st = 15,
                                            it_peak_wk_2nd = 40,
                                            it_runif_seed = 123)
ls_concept$plt_concept_week_of_year
```




```
df_dist_conception <- ls_concept$df_dist_conception
kable(df_dist_conception) %>% kable_styling_fc()
```

1.3 Temperature Process Sine Wave and AR(1)

Let d index day of survey. We model temperature in Fahrenheit as F_t using an AR(1) persistent process and a sine function. According to Wikipedia, which reports Climate data for Guangzhou, China, daily mean temperature varies between 57F (Jan) and 84F (June). Record high temperature was reported in July at 102.4F. Record low temperature was reported in December at 32.0F. We will mark out these temperature levels visually in graphs below.

Specifically, temperature in Fahrenheit is modeled as:

$$F_t = \exp\left(\mu + \epsilon_t + \gamma \sin\left(\frac{W_t}{52} \cdot 2 \cdot \pi + \frac{3 - \frac{1}{6}}{2} \cdot \pi\right)\right)$$

where γ scales the sine curve, and the shock process ϵ follows a AR(1) process:

$$\epsilon_t = \rho \cdot \epsilon_{t-1} + \nu_t$$

with $0 < \rho < 1$, and $\nu \sim N(0, \sigma_\nu)$. Using the function above, the coldest month will be somewhere around january.

Below, we formulate the temperature simulation function *ffi_daily_temp_simulation()*.

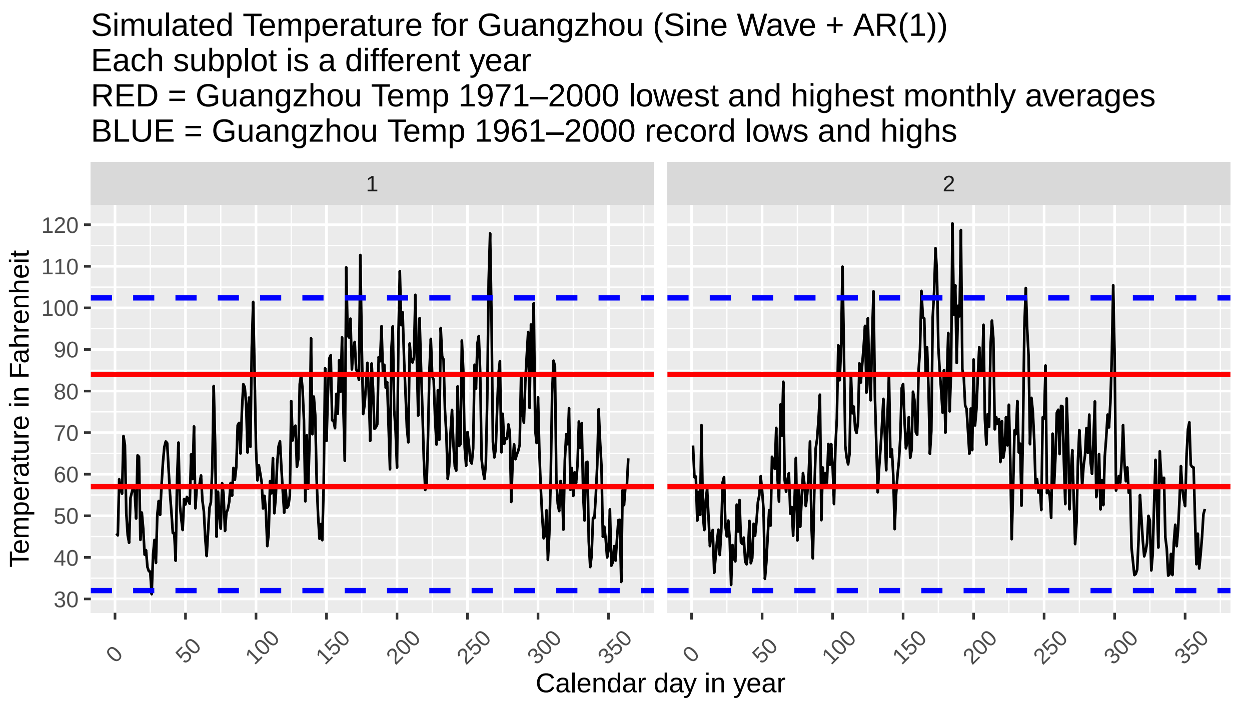
```
ffi_daily_temp_simulation <- function(
  fl_mthly_mean_lowest = 57,
  fl_mthly_mean_highest = 84,
  fl_record_lowest = 32,
  fl_record_highest = 102.4,
  it_weeks_in_year = 52,
  it_days_in_week = 7,
```

```r
  it_years = 6,
  fl_mu = 4.15,
  fl_sin_scaler = 0.20,
  fl_sigma_nv = 0.15,
  fl_rho_persist = 0.40,
  it_rand_seed = 123,
  st_extreme_cold_percentile = "p05",
  st_extreme_heat_percentile = "p95"
){
  # # Guangzhou temp info
  # fl_mthly_mean_lowest <- 57
  # fl_mthly_mean_highest <- 84
  # fl_record_lowest <- 32
  # fl_record_highest <- 102.4

  # # Total number of periods (over three years)
  # it_weeks_in_year <- 52
  # it_days_in_week <- 7
  # it_years <- 6
  it_max_days <- it_weeks_in_year*it_days_in_week
  T <- it_max_days * it_years

  # # Mean temp
  # fl_mu <- 4.15
  # fl_sin_scaler <- 0.20
  # # AR 1 parameter
  # fl_sigma_nv <- 0.15
  # fl_rho_persist <- 0.40

  # Generate a vector of shocks
  set.seed(it_rand_seed)
  ar_nv_draws <- rnorm(T, mean = 0, sd = fl_sigma_nv)

  # Generate a vector of epsilons
  ar_epsilon_ar1 <- vector("double", length=T)
  ar_epsilon_ar1[1] <- ar_nv_draws[1]
  for (it_t in 2:T) {
    ar_epsilon_ar1[it_t] <- fl_rho_persist*ar_epsilon_ar1[it_t-1] + ar_nv_draws[it_t]
  }

  # Generate week by week sin curve values
  ar_day_at_t <- rep(1:it_max_days, it_years)
  ar_year_at_t <- as.vector(t(matrix(data=rep(1:it_years, it_max_days),
                                     nrow=it_years, ncol=it_max_days)))
  ar_base_temp <- sin((ar_day_at_t/it_max_days)*2*pi + ((3-1/6)/2)*pi)

  # Generate overall temperature in Fahrenheit
  ar_fahrenheit_city_over_t <-
    exp(fl_mu + ar_epsilon_ar1 + fl_sin_scaler*ar_base_temp)

  # Dataframe with Temperatures
  mt_fahrenheit_info <- cbind(ar_day_at_t, ar_year_at_t, ceiling(ar_day_at_t/it_days_in_week),
    ar_fahrenheit_city_over_t,
```

```r
    exp(ar_base_temp), ar_epsilon_ar1, ar_nv_draws)
  ar_st_varnames <- c('survey_t','day_of_year', 'year', 'week_of_year',
    'Fahrenheit', 'FnoShock', 'AR1Shock', 'RandomDraws')

  # Combine to tibble, add name col1, col2, etc.
  df_fahrenheit <- as_tibble(mt_fahrenheit_info) %>%
    rowid_to_column(var = "t") %>%
    rename_all(~c(ar_st_varnames))

  # Generate extreme temperatures
  df_stats_fahrenheit <- REconTools::ff_summ_percentiles(df_fahrenheit, FALSE)
  # Add Extreme Thresholds
  fl_lowF_threshold <- df_stats_fahrenheit %>% filter(var == "Fahrenheit") %>% pull(st_extreme_cold_perc
  fl_highF_threshold <- df_stats_fahrenheit %>% filter(var == "Fahrenheit") %>% pull(st_extreme_heat_per
  df_fahrenheit <- df_fahrenheit %>%
    mutate(extreme_cold = case_when(Fahrenheit <= fl_lowF_threshold ~ 1, TRUE ~ 0)) %>%
    mutate(extreme_hot = case_when(Fahrenheit >= fl_highF_threshold ~ 1, TRUE ~ 0))
  # REconTools::ff_summ_percentiles(df_fahrenheit, FALSE)

  # Title
  st_title <- paste0('Simulated Temperature for Guangzhou (Sine Wave + AR(1))\n',
      'Each subplot is a different year\n',
      'RED = Guangzhou Temp 1971-2000 lowest and highest monthly averages\n',
      'BLUE = Guangzhou Temp 1961-2000 record lows and highs')

  # Display
  plt_fahrenheit <- df_fahrenheit %>%
    ggplot(aes(x = ar_day_at_t, y=Fahrenheit)) +
    geom_line() +
    geom_hline(yintercept = fl_mthly_mean_lowest, linetype = "solid", colour = "red", size = 1) +
    geom_hline(yintercept = fl_mthly_mean_highest, linetype = "solid", colour = "red", size = 1) +
    geom_hline(yintercept = fl_record_lowest, linetype = "dashed", colour = "blue", size = 1) +
    geom_hline(yintercept = fl_record_highest, linetype = "dashed", colour = "blue", size = 1) +
    facet_wrap(~ year) +
    labs(
      title = st_title,
      x = 'Calendar day in year',
      y = 'Temperature in Fahrenheit'
      ) +
    scale_x_continuous(n.breaks = 12) +
    scale_y_continuous(n.breaks = 10) +
    theme(
      axis.text.x = element_text(angle = 45, vjust = 0.1, hjust = 0.1)
    )

  # Return
  return(list(
    df_fahrenheit = df_fahrenheit,
    plt_fahrenheit = plt_fahrenheit
  ))
}
```

Having constructed the temperature simulation function, first we call it with AR1 shock + sin curve, persistence is high at $\rho = 0.7$.

```r
# Test 1: Call function with AR1 + Since Curve
ls_fahrenheit <- ffi_daily_temp_simulation(
  fl_mthly_mean_lowest = 57,
  fl_mthly_mean_highest = 84,
  fl_record_lowest = 32,
  fl_record_highest = 102.4,
  it_weeks_in_year = 52,
  it_days_in_week = 7,
  it_years = 2,
  fl_mu = 4.15,
  fl_sin_scaler = 0.25,
  fl_sigma_nv = 0.15,
  fl_rho_persist = 0.70,
  it_rand_seed = 123)
print(ls_fahrenheit$plt_fahrenheit)
```

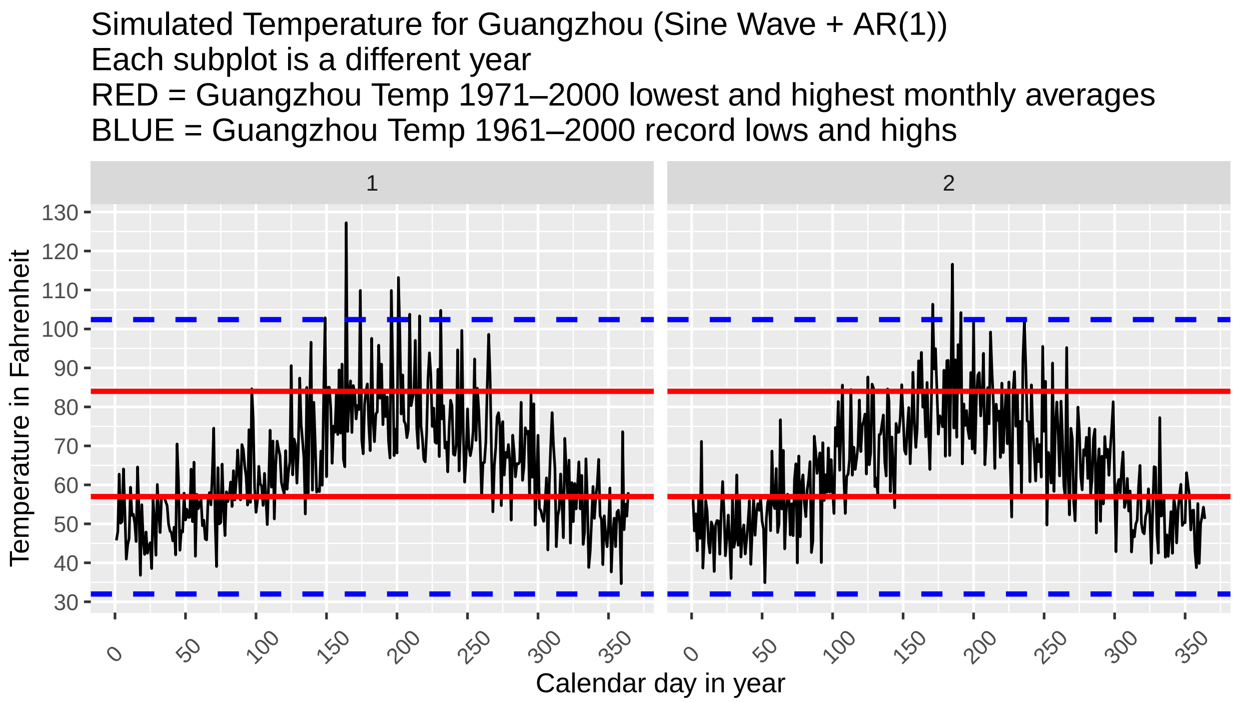



```r
# df_fahrenheit <- ls_fahrenheit$df_fahrenheit
# kable(df_fahrenheit) %>% kable_styling_fc()
```

Second, we call the same temperature function, but we reduce shock persistence to 0.

```r
# Test 2: Call function with defaults
ls_fahrenheit <- ffi_daily_temp_simulation(
  fl_mthly_mean_lowest = 57,
  fl_mthly_mean_highest = 84,
  fl_record_lowest = 32,
  fl_record_highest = 102.4,
  it_weeks_in_year = 52,
  it_days_in_week = 7,
  it_years = 2,
```

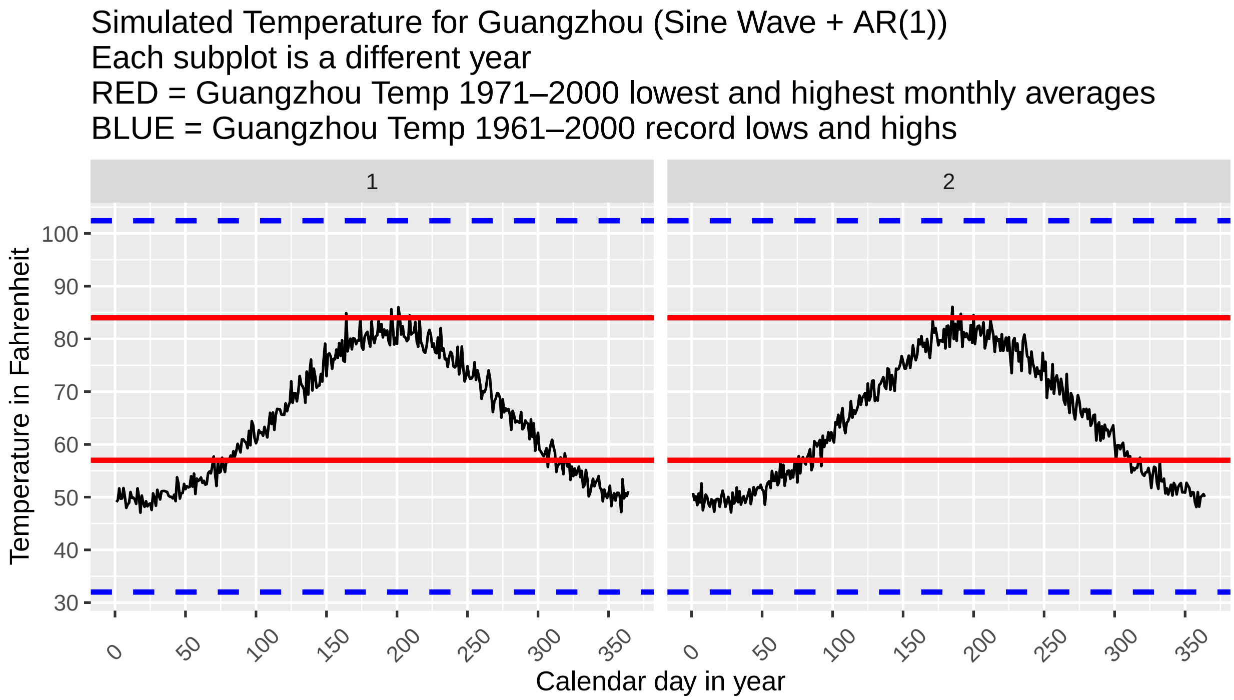
```
  fl_mu = 4.15,
  fl_sin_scaler = 0.25,
  fl_sigma_nv = 0.15,
  fl_rho_persist = 0.0,
  it_rand_seed = 123)
# Show
print(ls_fahrenheit$plt_fahrenheit)
```




```
# df_fahrenheit <- ls_fahrenheit$df_fahrenheit
# kable(df_fahrenheit) %>% kable_styling_fc()
```

Third, we call the same temperature function, but we reduce shock persistence to 0 and shock variance to almost zero.

```
# Test 2: Call function with defaults
ls_fahrenheit <- ffi_daily_temp_simulation(
  fl_mthly_mean_lowest = 57,
  fl_mthly_mean_highest = 84,
  fl_record_lowest = 32,
  fl_record_highest = 102.4,
  it_weeks_in_year = 52,
  it_days_in_week = 7,
  it_years = 2,
  fl_mu = 4.15,
  fl_sin_scaler = 0.25,
  fl_sigma_nv = 0.025,
  fl_rho_persist = 0.0,
  it_rand_seed = 123)
# Show
print(ls_fahrenheit$plt_fahrenheit)
```




```
# df_fahrenheit <- ls_fahrenheit$df_fahrenheit
# kable(df_fahrenheit) %>% kable_styling_fc()
```

1.4 Distributions of Conception, Birth, Temperature and Pre-Term

We will simulate several datasets on birth and relate to temperature distributions. Below we do that, we go through several key points that will be confirmed/demonstrated by the simulations.

1.4.1 Conception and Birth Marginal and Joint Distributions

There are three relevant distributional concepts here:

- $P(C_i)$: From *ffi_concept_distribution_year()* function generated earlier, this is the marginal discrete distribution of conception by calendar week of the year (perhaps some months are more popular for conception than others)
- $P(B_i)$: From *ffi_gestation_age_at_birth_dist()* function generated earlier, this is the marginal discrete distribution of gestational age by birth by day/week of gestation.
- $P(C_i, B_i)$: the joint distribution of C_i and B_i. If they are unrelated, than at each conception day, the conditional distribution of gestational age at birth is identical. To allow for arbitrary correlation between conception and gestational-age, we can draw from some mixture of joint normal distribution with some set of variance-covariance matrixes, invert to the draws to quantiles along each dimension, and then map the quantiles to discrete outcomes along C and B dimensions.

1.4.2 Extreme Temperature and Pre-Term Births

Pre-term birth or not is determined jointly by the gestational age at birth and conception date distributions, which determine the conception and birth dates. We establish a common threshold below which a birth is considered pre-term.

Extreme temperature and whether a child is pre-term or not can be related.

Under **scenario (A)**, conception dates are uniformly distribution across the year, and gestational age at birth is unrelated to conception time. Given this, extreme temperature or not has nothing to do with gestational age at birth.

Under **scenario (B)**, we can spuriously have pre-term experiencing less extreme-cold/heat when $P(C_i)$ is non-uniform/random, meaning $P(C_i) \neq \frac{1}{\text{CountTotalDatesOfConception}}$. Suppose all conception starts in Jan. and all extreme temp happen in Oct. Suppose pre-term means birth in Sep. rather than Nov.. If all extreme cold takes place in Oct. Then all pre-term births will not have experienced extreme cold. So the potential direction for spuriousness here is that pre-term will be correlated with less extreme temp. This is fixed by a conception month fixed effect.

Additionally, pre-term could experience more extreme-cold under **scenario (C.1)**, where extreme temperature has a causal impact on gestational-age, causally leading to more pre-term. This is the research hypothesis of interest.

But pre-term could also experience more extreme-cold under **scenario (C.2)** for another reason. Suppose the distribution of high-pregnancy-risk parents/mothers across calendar month in each year might be non-random: if all high risk mother get pregnant in August, and low risk mother get pregnant in March, we might end up with full-term birth not experiencing extreme temperature, but pre-term experiencing a lot of extreme-cold. The pre-term (or birthweight) and cold is hence due to the conception timing of high-risk pregnancies, and cold might not no effects on whether a birth is pre-term or not.

1.4.3 Days and Percent of Gestational Days Exposed to Cold

Under the various scenarios, the relationship between the *number* of days and *percentage* of conception days to pre-term birth of not will be different. They are:

- Under **Scenario (A)**:
  - the share of days exposed to extreme cold will be identical for pre-term and full-term births.
  - the number of days exposed to extreme cold is *less* for pre-term birth compared to full-term.
- Under **Scenario (B)**:
  - the share of days exposed to extreme cold will be *less* for pre-term and full-term births.
  - the number of days exposed to extreme cold is *less* for pre-term birth compared to full-term.
- Under **Scenario (C.1) and (C.2)**:
  - the share of days exposed to extreme cold is *more* for pre-term birth compared to full-term.
  - the number of days exposed to extreme cold might generally be *less* although potentially *more* for pre-term and full-term births.

In the exercises below, we only simulate Scenarios (A) and (B) for now.

1.5 Simulating Datasets on Extreme Temperature Exposures Across Pregnancies

1.5.1 Compute Extreme Temperature Exposure

We generate a dataframe with N individuals, each with conception day C_i, birth day B_i, and whether in the birth was full-term or not F_i. For simplicity, assume that there are 7 days per week, 4 weeks per month, and 12 months per year. This means there are 336 days of possible birth in a year.

Our gestational age distribution function can take any mean and standard deviation, we rescale the information mentioned prior from Right from the Start under the assumption of 336 days per year. (Plug in mean and sd as if there are 365 days into the function below, it will rescale that based on the actual number of days in the simulated year).

- Conception draws $P(C_i)$: For each of the N individuals, we first draw week of conception from the week of conception distribution function (adjust distribution depending on number of years and the starting month), then we draw the day of week of conception randomly, and the year of conception randomly as well.

- Gestation draws $P(B_i - C_i)$: For each of the N individuals, we draw from the gestational week at the time of birth distribution the number of weeks of gestation, multiply this by seven and add a random number between 1 to 7.
- Joint distribution $P(C_i, B_i)$: For the simulation below, the draws for day of birth is independent from the draws for gestational age at time of birth.

First, we create a function for generating birth conception and gestational distributions.

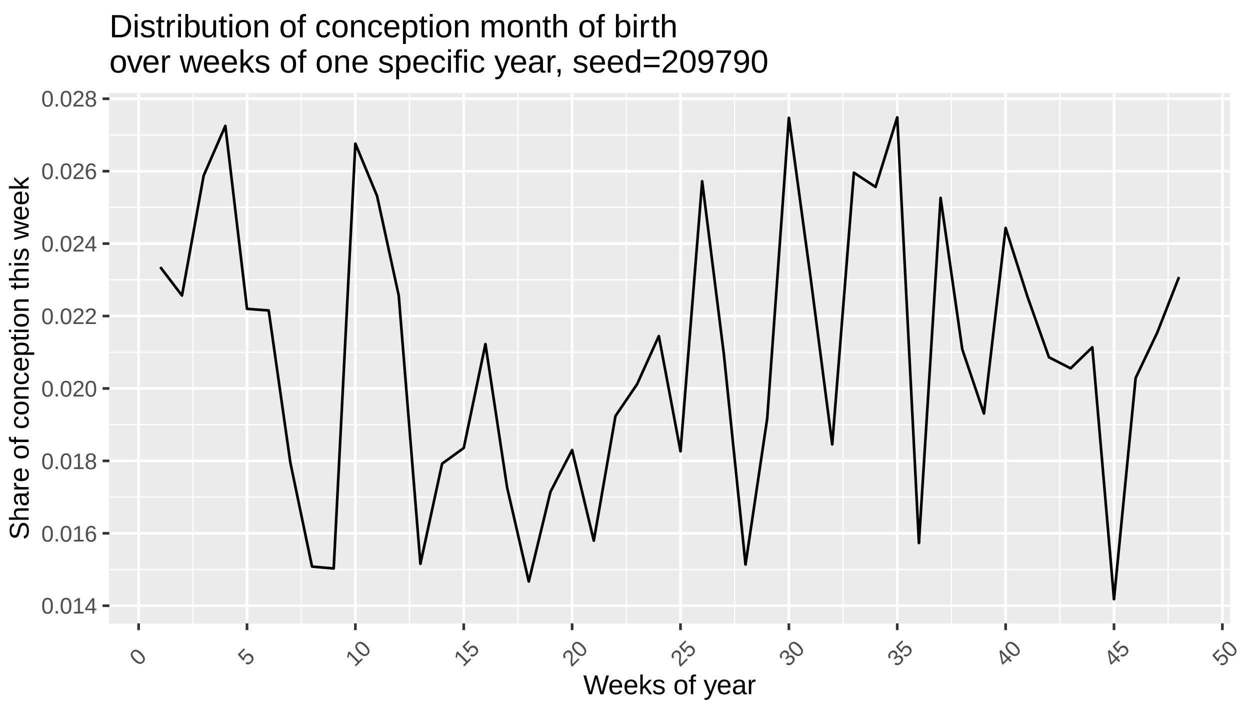
```
ffi_pop_concept_birth_simu <- function(
  it_pop_n = 50000,
  it_days_in_week = 7,
  it_weeks_in_month = 4,
  it_months_in_year = 12,
  it_years = 3,
  it_rng_seed = 123,
  fl_pre_term_ratio = 0.84,
  fl_peak_concept_frac_of_year_1st = 0.3,
  fl_peak_concept_frac_of_year_2nd = 0.9,
  fl_binom_1st_wgt = 0.15,
  fl_binom_2nd_wgt = 0.05,
  fl_mu_gabirth_days_365 = 276,
  fl_sd_gabirth_days_365 = 14
) {

  # # 1. Define parameters
  # # 1.a Number of individuals of interest
  # it_pop_n <- 50000
  # # 1.b Number of days per week, week per month
  # # for simplicity, 7 days per week, 4 weeks per month, 12 months
  # it_days_in_week <- 7
  # it_weeks_in_month <- 4
  # it_months_in_year <- 12
  # it_years <- 3
  # # 1.c random draw seed
  # it_rng_seed <- 123
  # # 1.d pre-term threshold
  # # what fraction of maximum pregnancy length is pre-term?
  # # Max length 44 weeks for example, 44*0.85 is about 37 months.
  # fl_pre_term_ratio <- 0.84

  # 1.e Other dates
  it_weeks_in_year <- it_weeks_in_month*it_months_in_year
  it_days_in_month <- it_days_in_week*it_weeks_in_month
  it_days_in_year <- it_days_in_week*it_weeks_in_month*it_months_in_year
  # 1.f Month of conception distribution
  it_peak_wk_1st <- round(it_weeks_in_year*fl_peak_concept_frac_of_year_1st)
  it_peak_wk_2nd <- round(it_weeks_in_year*fl_peak_concept_frac_of_year_2nd)
  # fl_binom_1st_wgt <- 0.15
  # fl_binom_2nd_wgt <- 0.05
  # # 1.g Gestational age at birth distribution parameters
  mu_gabirth_days <- round((fl_mu_gabirth_days_365/365)*it_days_in_year)
  sd_gabirth_days <- round((fl_sd_gabirth_days_365/365)*it_days_in_year)

  # 2. Date of conception random draws
  # 2.a Week of conception distribution
```

```r
ls_concept_fc <- ffi_concept_distribution_year(
  it_max_weeks = it_weeks_in_year,
  it_peak_wk_1st = it_peak_wk_1st, it_peak_wk_2nd = it_peak_wk_2nd,
  fl_binom_1st_wgt = fl_binom_1st_wgt, fl_binom_2nd_wgt = fl_binom_2nd_wgt,
  it_runif_seed = it_rng_seed*210)
df_dist_conception_week <- ls_concept_fc$df_dist_conception
# 2.b.1 Randomly (uniformly) drawing the day of birth
set.seed(it_rng_seed*221)
ar_draws_conception_day_of_week <- sample(
  it_days_in_week, it_pop_n, replace=TRUE)
# 2.b.2 Week of conception draws
set.seed(it_rng_seed*222)
ar_draws_conception_week <- sample(
  df_dist_conception_week$conception_calendar_week,
  it_pop_n,
  prob=df_dist_conception_week$conception_prob,
  replace=TRUE)
# 2.b.3 Randomly (uniformly) drawing the year of conception
set.seed(it_rng_seed*223)
ar_draws_conception_year <- sample(
  it_years, it_pop_n, replace=TRUE)
# 2.c Date of birth
ar_draws_concept_date <- (ar_draws_conception_year-1)*it_days_in_year +
  (ar_draws_conception_week-1)*it_days_in_week +
  ar_draws_conception_day_of_week
ar_draws_conception_day_of_year <- (ar_draws_conception_week-1)*it_days_in_week +
  ar_draws_conception_day_of_week

# 3. Gestational age at birth distribution simulation
# 3.a Gestational age distribution
ls_gsbirth_fc <- ffi_gestation_age_at_birth_dist(
  mu_gabirth_days = mu_gabirth_days, sd_gabirth_days = sd_gabirth_days)
df_dist_gabirth <- ls_gsbirth_fc$df_dist_gabirth
# 3.b.1 Gestational day of week draws (random)
set.seed(it_rng_seed*321)
ar_draws_gsbirth_day_of_week <- sample(
  it_days_in_week, it_pop_n, replace=TRUE)
# 3.b.2 Gestational week draws
set.seed(it_rng_seed*322)
ar_draws_gsbirth_week <- sample(
  df_dist_gabirth$gabirth,
  it_pop_n,
  prob=df_dist_gabirth$prob,
  replace=TRUE)
# 3.c Gestational days at birth
ar_draws_gsbirth_day <- ar_draws_gsbirth_week*it_days_in_week + ar_draws_gsbirth_day_of_week
ar_draws_birth_date <- ar_draws_concept_date + ar_draws_gsbirth_day

# 4. Create dataframe
# 4.a Variables and labels
mt_birth_data <- cbind(
  ar_draws_concept_date, ar_draws_birth_date, ar_draws_gsbirth_day,
  ar_draws_conception_year, ar_draws_conception_week,
```

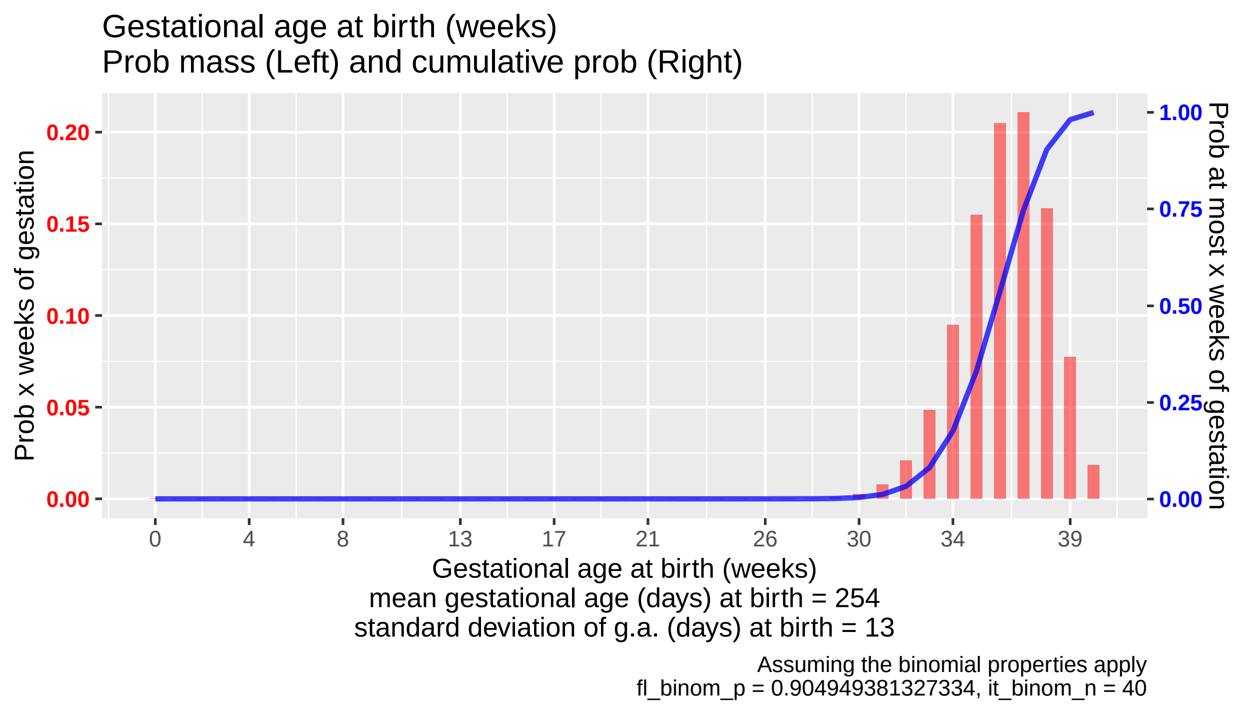
```
    ar_draws_conception_day_of_week, ar_draws_conception_day_of_year,
    ar_draws_gsbirth_week, ar_draws_gsbirth_day_of_week)
  ar_st_varnames <- c('id',
    'survey_date_conception', 'survey_date_birth', 'gestation_length_in_days',
    'conception_year', 'conception_week',
    'conception_day_of_week', 'concept_day_of_year',
    'gestational_week_at_birth', 'gestational_day_of_week_at_birth')
  # 4.b tibble with conception and birth data
  df_birth_data <- as_tibble(mt_birth_data) %>%
    rowid_to_column(var = "id") %>%
    rename_all(~c(ar_st_varnames)) %>%
    arrange(survey_date_conception, survey_date_birth)
  # 4.c generate cut-off for preterm
  it_pre_term_threshold <- round(fl_pre_term_ratio*length(df_dist_gabirth$gabirth)*it_days_in_week)
  df_birth_data <- df_birth_data %>% mutate(
    preterm = case_when(it_pre_term_threshold >= gestation_length_in_days ~ 1,
                        TRUE ~ 0) )

  # 5. Display data
  # Display
  st_title <- paste0('Day of year of conception and gestational age at birth\n',
    'pop=', it_pop_n, ', days-in-year=', it_days_in_year, ', seed=', it_rng_seed, '\n',
    'mean-ga-at-birth-in-month=', round(mu_gabirth_days/it_days_in_month, 3),
    ', sd-ga-at-birth-in-month=', round(sd_gabirth_days/it_days_in_month, 3))
  plt_concept_birth <- df_birth_data %>%
    mutate(preterm = factor(preterm)) %>%
    ggplot(aes(x = concept_day_of_year, y=gestation_length_in_days, color=preterm)) +
    facet_wrap(~ conception_year) +
    geom_point() +
    labs(
      title = st_title,
      x = 'Calendar day in year',
      y = 'Gestational age in days at birth'
      ) +
    scale_x_continuous(n.breaks = 12) +
    scale_y_continuous(n.breaks = 10) +
    theme(
      axis.text.x = element_text(angle = 45, vjust = 0.1, hjust = 0.1)
    )
  # print(plt_concept_birth)
  # kable(df_birth_data) %>% kable_styling_fc_wide()
  # plot(df_birth_data$survey_date_conception, df_birth_data$gestation_length_in_days)

  # Return
  return(list(
    df_birth_data = df_birth_data,
    plt_concept_birth = plt_concept_birth,
    ls_concept_fc = ls_concept_fc,
    ls_gsbirth_fc = ls_gsbirth_fc
  ))
}
```

Second, we generate a dataframe where the conception months are randomly/uniformly distributed, and there is no correlation between conception and gestation. This is **Scenario (A)**.

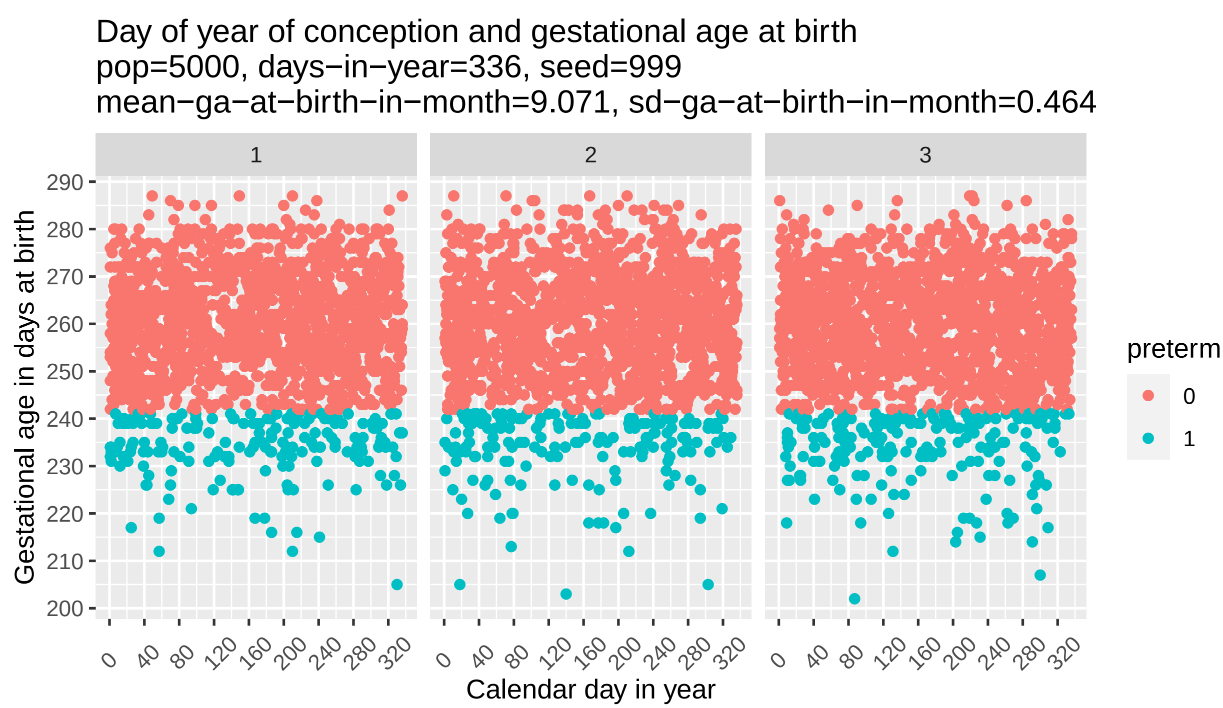
```
# Define some dates
it_days_in_week <- 7
it_weeks_in_month <- 4
it_months_in_year <- 12
it_years <- 3
# Simulate
ls_concept_birth <- ffi_pop_concept_birth_simu(
  it_pop_n = 5000,
  it_days_in_week = it_days_in_week,
  it_weeks_in_month = it_weeks_in_month,
  it_months_in_year = it_months_in_year,
  it_years = it_years,
  it_rng_seed = 999,
  fl_pre_term_ratio = 0.84,
  fl_peak_concept_frac_of_year_1st = 0.3,
  fl_peak_concept_frac_of_year_2nd = 0.9,
  fl_binom_1st_wgt = 0.00,
  fl_binom_2nd_wgt = 0.00,
  fl_mu_gabirth_days_365 = 276,
  fl_sd_gabirth_days_365 = 14
)
# Get dataframe and print distribution
df_birth_data_rand_cor0 <- ls_concept_birth$df_birth_data
print(ls_concept_birth$ls_concept_fc$plt_concept_week_of_year)
```




```
print(ls_concept_birth$ls_gsbirth_fc$plt_dist_gabirth)
```



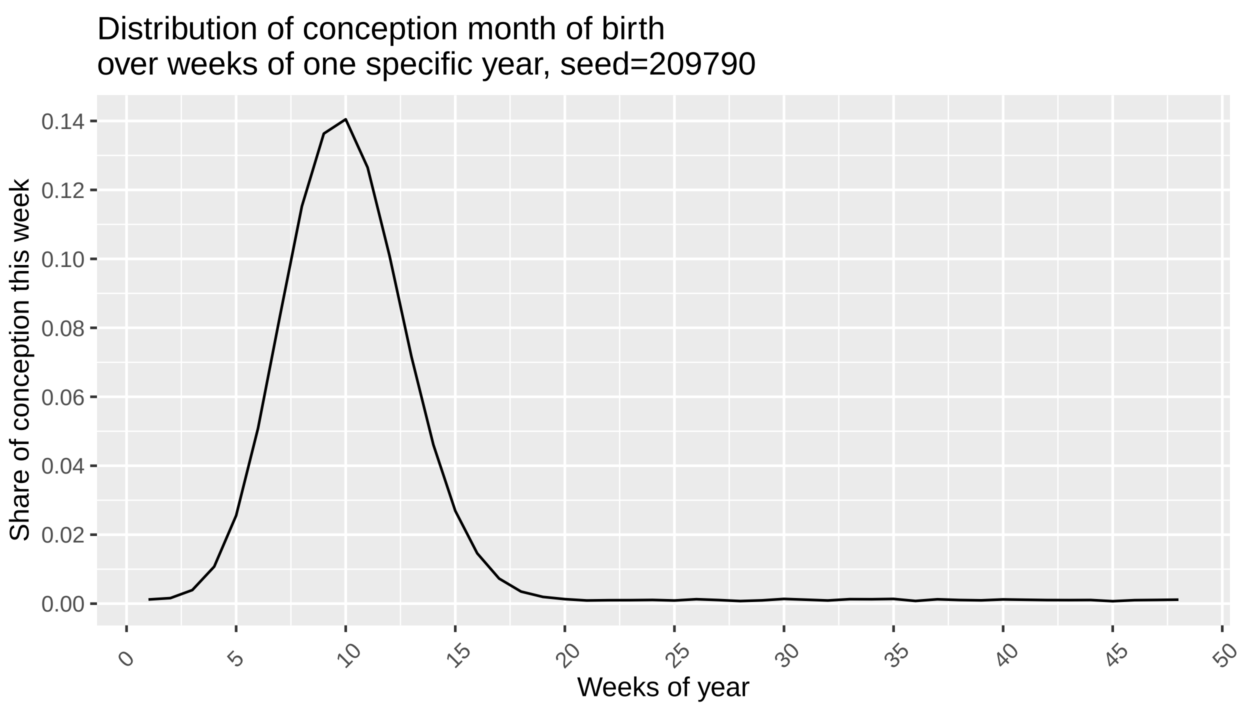

```
print(ls_concept_birth$plt_concept_birth)
```



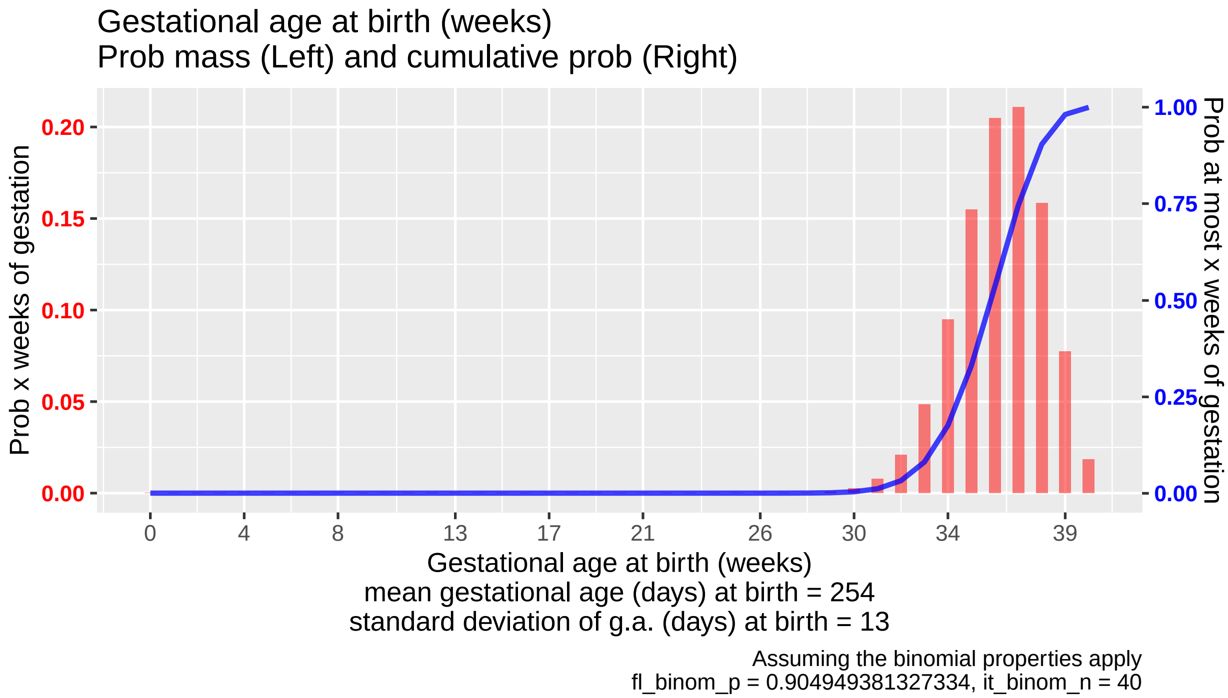
Third, we generate a dataframe where the conception distribution has a high peak around Feb, and there is no correlation between conception and gestation. This is **Scenario (B)**.

```
# Simulate
ls_concept_birth <- ffi_pop_concept_birth_simu(
  it_pop_n = 5000,
```

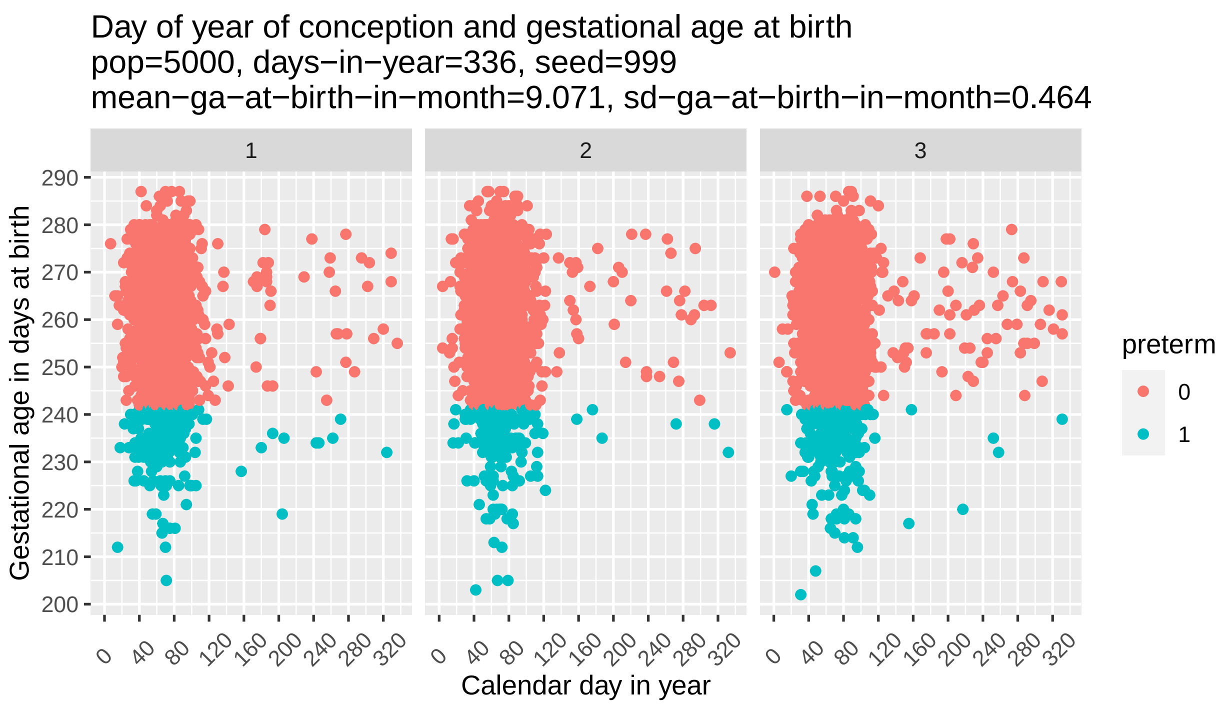
```
  it_days_in_week = it_days_in_week,
  it_weeks_in_month = it_weeks_in_month,
  it_months_in_year = it_months_in_year,
  it_years = it_years,
  it_rng_seed = 999,
  fl_pre_term_ratio = 0.84,
  fl_peak_concept_frac_of_year_1st = 0.2,
  fl_peak_concept_frac_of_year_2nd = 0.9,
  fl_binom_1st_wgt = 0.95,
  fl_binom_2nd_wgt = 0.00,
  fl_mu_gabirth_days_365 = 276,
  fl_sd_gabirth_days_365 = 14
)
# Get dataframe and print distribution
df_birth_data_CFeb_cor0 <- ls_concept_birth$df_birth_data
print(ls_concept_birth$ls_concept_fc$plt_concept_week_of_year)
```




```
print(ls_concept_birth$ls_gsbirth_fc$plt_dist_gabirth)
```




```
print(ls_concept_birth$plt_concept_birth)
```

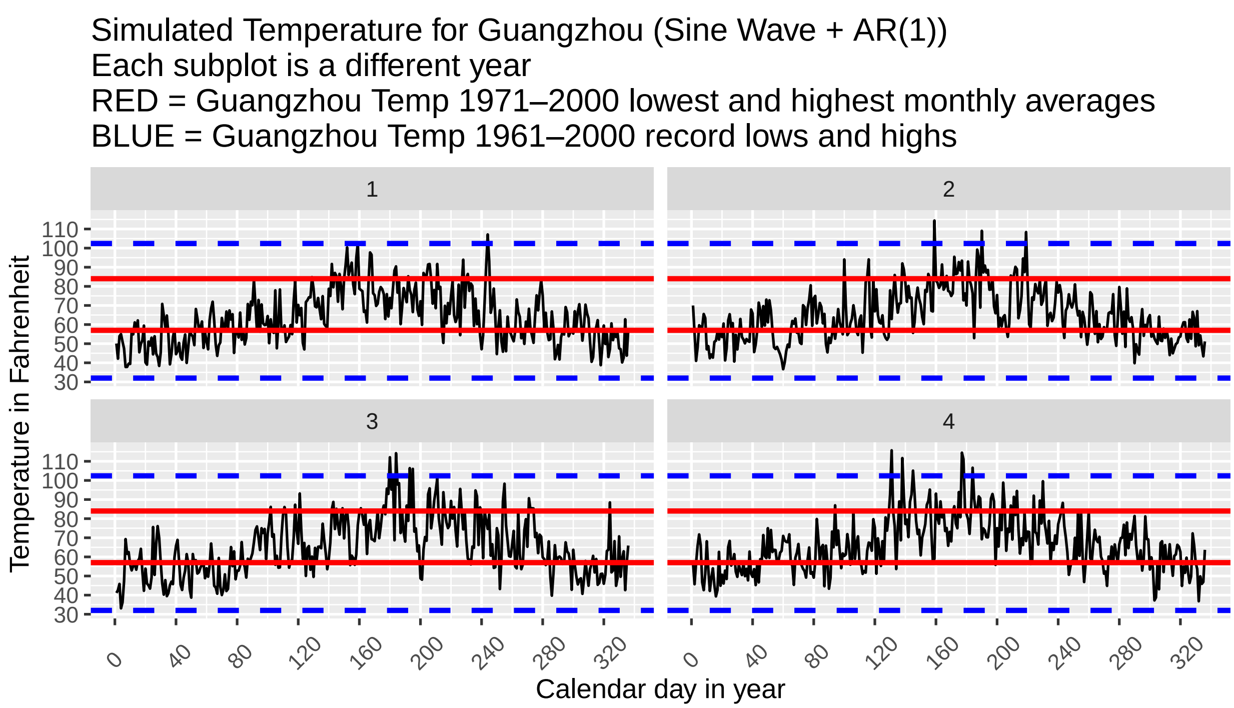


1.5.2 Compute Extreme Temperature Exposure

we simulate temperature corresponding to the span of data of interest that covers all individuals' potential course of pregnancy. We loop over each individual, and calculate the individual-specific extreme temperature exposure. Specifically, we store the percentage of days during pregnancy exposed to extreme cold or extreme hot. We use as threshold 5 percent temperature extreme tails. We also store the number of days as well as the percent of gestational days that are exposed to extreme cold or hot.

First, we generate the temperature distribution along with extreme cold and hot days.

```
# Simulate the temperature distribution using just define parameters
ls_fahrenheit <- ffi_daily_temp_simulation(
  fl_mthly_mean_lowest = 57,
  fl_mthly_mean_highest = 84,
  fl_record_lowest = 32,
  fl_record_highest = 102.4,
  it_weeks_in_year = it_months_in_year*it_weeks_in_month,
  it_days_in_week = it_days_in_week,
  it_years = it_years+1,
  it_rand_seed = 999,
  st_extreme_cold_percentile = "p05",
  st_extreme_heat_percentile = "p95")
print(ls_fahrenheit$plt_fahrenheit)
```




```
df_fahrenheit <- ls_fahrenheit$df_fahrenheit
summary(df_fahrenheit$Fahrenheit)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   33.08   54.47   63.38   65.06   74.59  115.74
```

```
df_stats_fahrenheit <- REconTools::ff_summ_percentiles(df_fahrenheit, FALSE)
```

Second, for one individual, we test computing the number of days exposed to extreme cold.

```
# Define the extreme cold function
ffi_extreme_cold_percent_gestation <- function(df_fahrenheit, it_date_conception, it_date_birth){

  # get extreme cold
```

```r
  ar_extreme_cold <- df_fahrenheit %>%
    filter(survey_t >= it_date_conception & survey_t <= it_date_birth) %>% pull(extreme_cold)

  # extreme cold days
  it_extreme_cold_days <- sum(ar_extreme_cold)

 return(it_extreme_cold_days)
}
# Test the function
it_extreme_cold_days <- ffi_extreme_cold_percent_gestation(df_fahrenheit, 11, 200)
print(it_extreme_cold_days)
```

```
## [1] 9
```

Third, we create a function that takes the birth dataframe and temperature dataframe as inputs and generate extreme cold days exposure for each pregancy in the birth dateaframe using apply with an anonymous function.

```r
# Given two dataframes with birth data and temperature data, find cold exposure
ffi_birth_extreme_exposure <- function(df_birth_data, df_fahrenheit){
  # apply row by row, anonymous function
  # see: https://fanwangecon.github.io/R4Econ/function/noloop/htmlpdfr/fs_apply.html#122_anonymous_func
  mt_birth_cold <- apply(df_birth_data, 1, function(row) {

      id <- row[1]
      it_date_conception <- row[2]
      it_date_birth <- row[3]
      it_preterm <- row[11]

      it_extreme_cold_days <- ffi_extreme_cold_percent_gestation(
        df_fahrenheit, it_date_conception, it_date_birth)

      mt_all_res <- cbind(id, it_date_conception, it_date_birth,
                          it_preterm, it_extreme_cold_days)

      return(mt_all_res)
    })

  # Column Names
  ar_st_varnames <- c('id', 'survey_date_conception', 'survey_date_birth',
          'preterm', 'days_extreme_cold')

  # Combine to tibble, add name col1, col2, etc.
  tb_birth_cold <- as_tibble(t(mt_birth_cold)) %>%
    rename_all(~c(ar_st_varnames)) %>%
    mutate(days_extreme_cold_percent = days_extreme_cold/(survey_date_birth-survey_date_conception))

  # Show Results
  # kable(tb_birth_cold[1:20,]) %>% kable_styling_fc()
  return(tb_birth_cold)
}
```

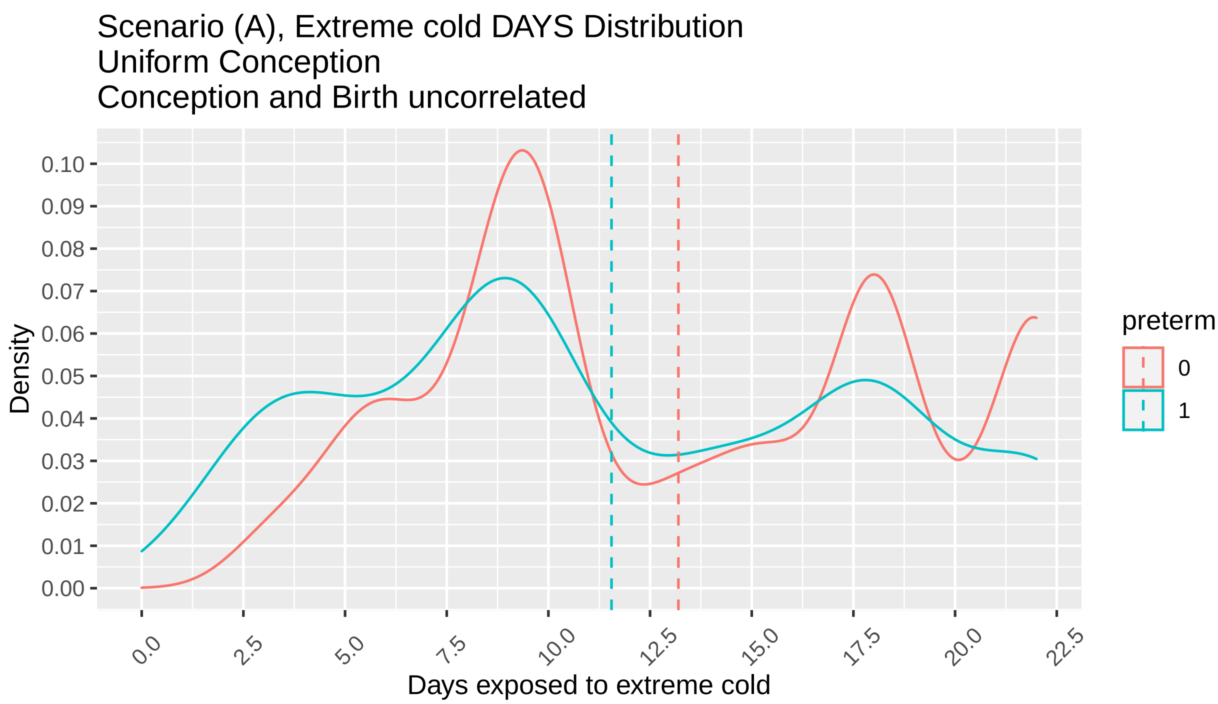
Fourth, we generate exposure cold exposures for the datasets we generated using the same temperature dataframe by different birth/conception dataframes created under different assumptions. We use the *ffi_birth_extreme_exposure* function just created.

```
# Scenario (A)
tb_birth_cold_rand_cor0 <- ffi_birth_extreme_exposure(df_birth_data_rand_cor0, df_fahrenheit)
# Scenario (B)
tb_birth_cold_CFeb_cor0 <- ffi_birth_extreme_exposure(df_birth_data_CFeb_cor0, df_fahrenheit)
```

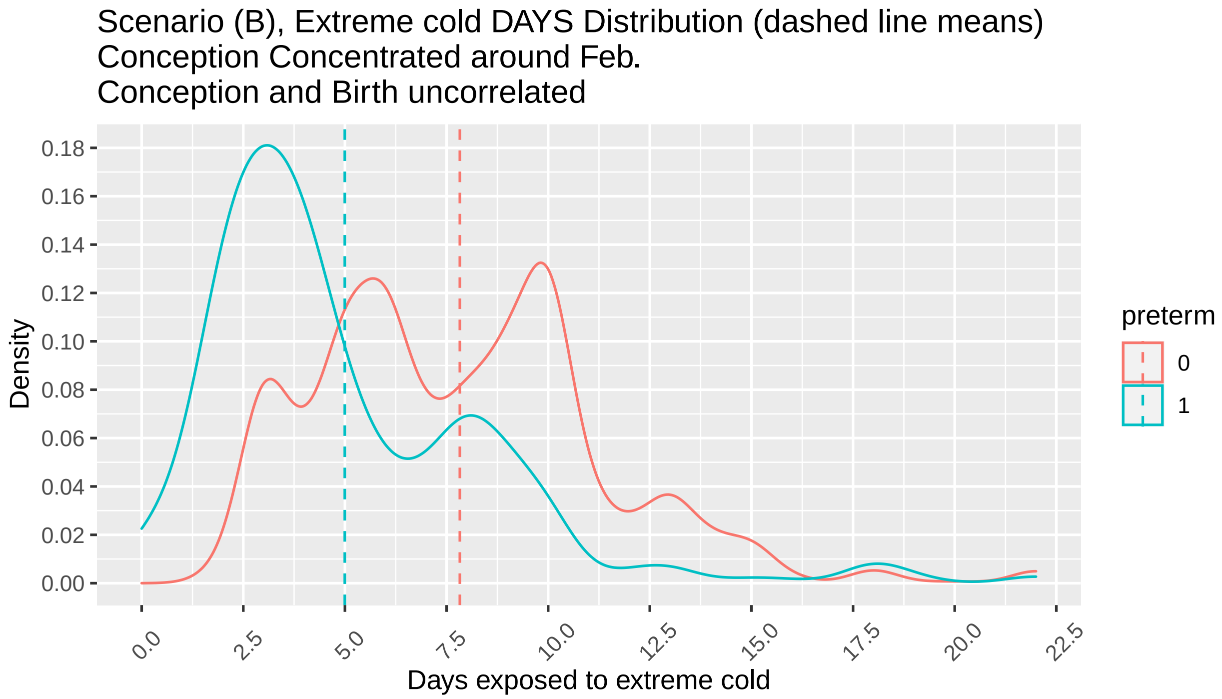
1.5.3 Visualize Pre-term, Full-term and Extreme Temperature Exposures

Using the data generated prior, first, we generate distribution of days exposed to extreme cold temperature.

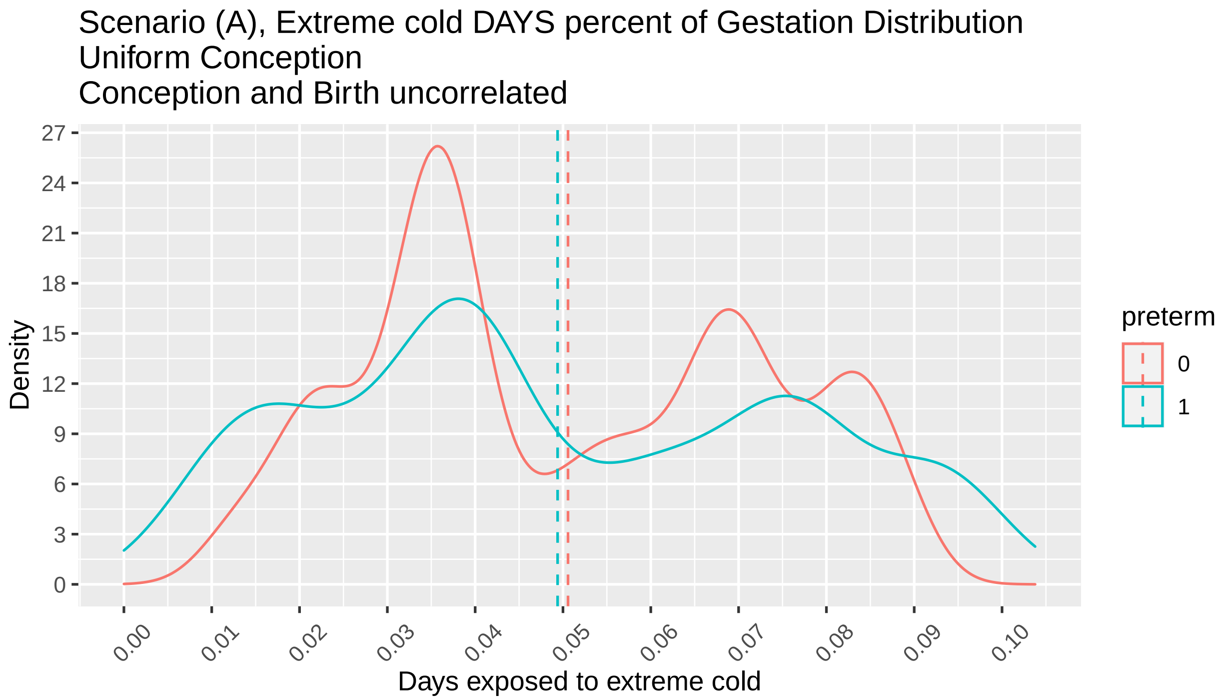
```
# summarize
str_stats_group <- 'allperc'
ar_perc <- c(0.05, 0.25, 0.5, 0.75, 0.95)

# For tb_birth_cold_rand_cor0
ls_summ_by_group <- REconTools::ff_summ_bygroup(
  tb_birth_cold_rand_cor0, c('preterm'),
  'days_extreme_cold', str_stats_group, ar_perc)
df_table_grp_stats_rand_cor0 <- ls_summ_by_group$df_table_grp_stats
print(df_table_grp_stats_rand_cor0)
# Visualize
plt_rand_cor0_level <- tb_birth_cold_rand_cor0 %>%
  mutate(preterm = factor(preterm)) %>%
  group_by(preterm) %>% mutate(days_extreme_cold_mean = mean(days_extreme_cold)) %>% ungroup() %>%
  ggplot(aes(x=days_extreme_cold, color=preterm)) +
  geom_density() +
  geom_vline(aes(xintercept=days_extreme_cold_mean, color=preterm), linetype="dashed") +
  labs(
    title = paste0('Scenario (A), Extreme cold DAYS Distribution\n',
                   'Uniform Conception\n',
                   'Conception and Birth uncorrelated'),
    x = 'Days exposed to extreme cold',
    y = 'Density'
    ) +
  scale_x_continuous(n.breaks = 10) +
  scale_y_continuous(n.breaks = 10) +
  theme(
    axis.text.x = element_text(angle = 45, vjust = 0.1, hjust = 0.1)
  )
print(plt_rand_cor0_level)
```



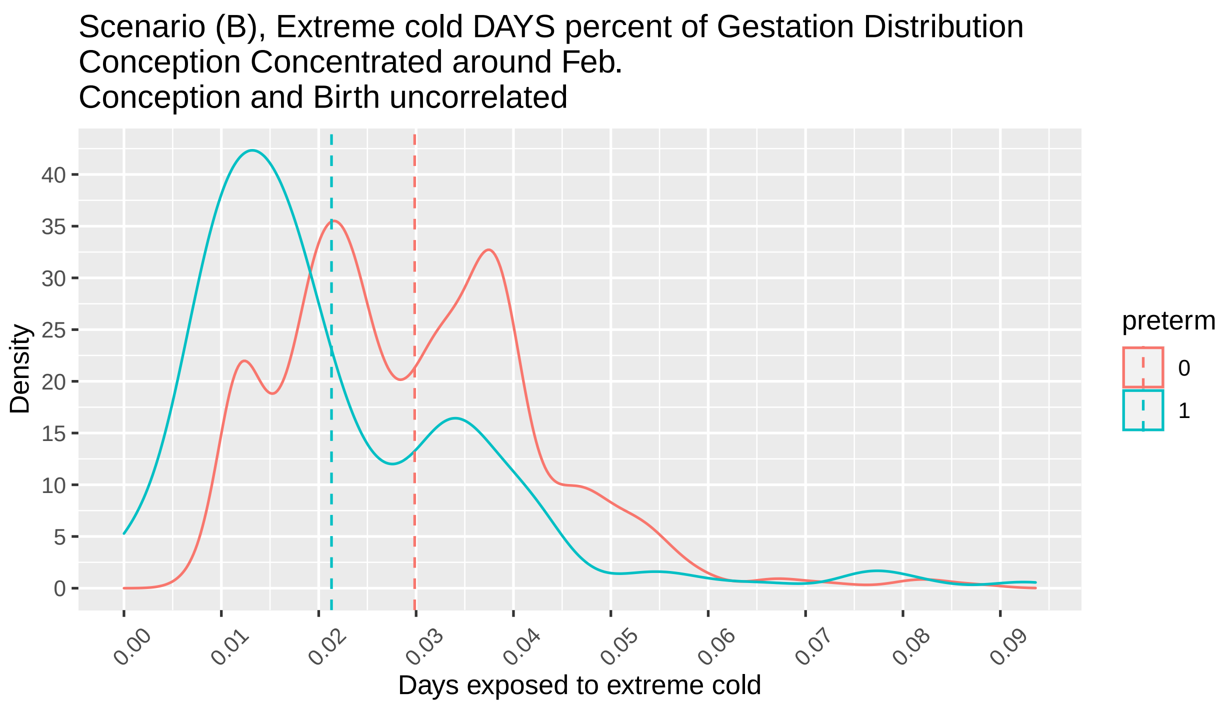

```
# For tb_birth_cold_CFeb_cor0
ls_summ_by_group <- REconTools::ff_summ_bygroup(
  tb_birth_cold_CFeb_cor0, c('preterm'),
  'days_extreme_cold', str_stats_group, ar_perc)
df_table_grp_stats_CFeb_cor0 <- ls_summ_by_group$df_table_grp_stats
print(df_table_grp_stats_CFeb_cor0)
# Visualize
plt_CFeb_cor0_level <- tb_birth_cold_CFeb_cor0 %>%
  mutate(preterm = factor(preterm)) %>%
  group_by(preterm) %>% mutate(days_extreme_cold_mean = mean(days_extreme_cold)) %>% ungroup() %>%
  ggplot(aes(x=days_extreme_cold, color=preterm)) +
  geom_density() +
  geom_vline(aes(xintercept=days_extreme_cold_mean, color=preterm), linetype="dashed") +
  labs(
    title = paste0('Scenario (B), Extreme cold DAYS Distribution (dashed line means)\n',
                   'Conception Concentrated around Feb.\n',
                   'Conception and Birth uncorrelated'),
    x = 'Days exposed to extreme cold',
    y = 'Density'
    ) +
  scale_x_continuous(n.breaks = 10) +
  scale_y_continuous(n.breaks = 10) +
  theme(
    axis.text.x = element_text(angle = 45, vjust = 0.1, hjust = 0.1)
  )
print(plt_CFeb_cor0_level)
```




```r
# summarize
str_stats_group <- 'allperc'
ar_perc <- c(0.05, 0.25, 0.5, 0.75, 0.95)

# For tb_birth_cold_rand_cor0
ls_summ_by_group <- REconTools::ff_summ_bygroup(
  tb_birth_cold_rand_cor0, c('preterm'),
  'days_extreme_cold_percent', str_stats_group, ar_perc)
df_table_grp_stats_rand_cor0 <- ls_summ_by_group$df_table_grp_stats
print(df_table_grp_stats_rand_cor0)
# Visualize
plt_rand_cor0_level <- tb_birth_cold_rand_cor0 %>%
  mutate(preterm = factor(preterm)) %>%
  group_by(preterm) %>% mutate(days_extreme_cold_percent_mean = mean(days_extreme_cold_percent)) %>% ung
  ggplot(aes(x=days_extreme_cold_percent, color=preterm)) +
  geom_density() +
  geom_vline(aes(xintercept=days_extreme_cold_percent_mean, color=preterm), linetype="dashed") +
  labs(
    title = paste0('Scenario (A), Extreme cold DAYS percent of Gestation Distribution\n',
                   'Uniform Conception\n',
                   'Conception and Birth uncorrelated'),
    x = 'Days exposed to extreme cold',
    y = 'Density'
    ) +
  scale_x_continuous(n.breaks = 10) +
  scale_y_continuous(n.breaks = 10) +
  theme(
    axis.text.x = element_text(angle = 45, vjust = 0.1, hjust = 0.1)
  )
print(plt_rand_cor0_level)
```




```
# For tb_birth_cold_CFeb_cor0
ls_summ_by_group <- REconTools::ff_summ_bygroup(
  tb_birth_cold_CFeb_cor0, c('preterm'),
  'days_extreme_cold_percent', str_stats_group, ar_perc)
df_table_grp_stats_CFeb_cor0 <- ls_summ_by_group$df_table_grp_stats
print(df_table_grp_stats_CFeb_cor0)
# Visualize
plt_CFeb_cor0_level <- tb_birth_cold_CFeb_cor0 %>%
  mutate(preterm = factor(preterm)) %>%
  group_by(preterm) %>% mutate(days_extreme_cold_percent_mean = mean(days_extreme_cold_percent)) %>% ung
  ggplot(aes(x=days_extreme_cold_percent, color=preterm)) +
  geom_density() +
  geom_vline(aes(xintercept=days_extreme_cold_percent_mean, color=preterm), linetype="dashed") +
  labs(
    title = paste0('Scenario (B), Extreme cold DAYS percent of Gestation Distribution\n',
                   'Conception Concentrated around Feb.\n',
                   'Conception and Birth uncorrelated'),
    x = 'Days exposed to extreme cold',
    y = 'Density'
    ) +
  scale_x_continuous(n.breaks = 10) +
  scale_y_continuous(n.breaks = 10) +
  theme(
    axis.text.x = element_text(angle = 45, vjust = 0.1, hjust = 0.1)
  )
print(plt_CFeb_cor0_level)
```



Scenario (B), Extreme cold DAYS percent of Gestation Distribution
Conception Concentrated around Feb.
Conception and Birth uncorrelated
Density
0
5
10
15
20
25
30
35
40
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.09
Days exposed to extreme cold
preterm
0
1

conception_calendar_week	conception_prob
1	0.0127130
2	0.0176571
3	0.0139137
4	0.0186053
5	0.0192192
6	0.0105366
7	0.0157063
8	0.0201894
9	0.0184492
10	0.0200423
11	0.0283162
12	0.0270358
13	0.0325643
14	0.0336904
15	0.0295436
16	0.0361683
17	0.0271255
18	0.0218356
19	0.0213639
20	0.0247214
21	0.0219399
22	0.0185480
23	0.0171439
24	0.0201420
25	0.0165473
26	0.0169517
27	0.0152796
28	0.0157608
29	0.0127605
30	0.0114004
31	0.0195570
32	0.0191672
33	0.0174747
34	0.0191898
35	0.0126323
36	0.0185624
37	0.0230986
38	0.0195683
39	0.0220649
40	0.0219644
41	0.0208423
42	0.0223014
43	0.0204080
44	0.0179026
45	0.0139712
46	0.0125547
47	0.0127300
48	0.0146663
49	0.0125517
50	0.0183537
51	0.0103274
52	0.0142397