PSTAT220A Homework 1

2025-10-13

Question 1 (25 pt)

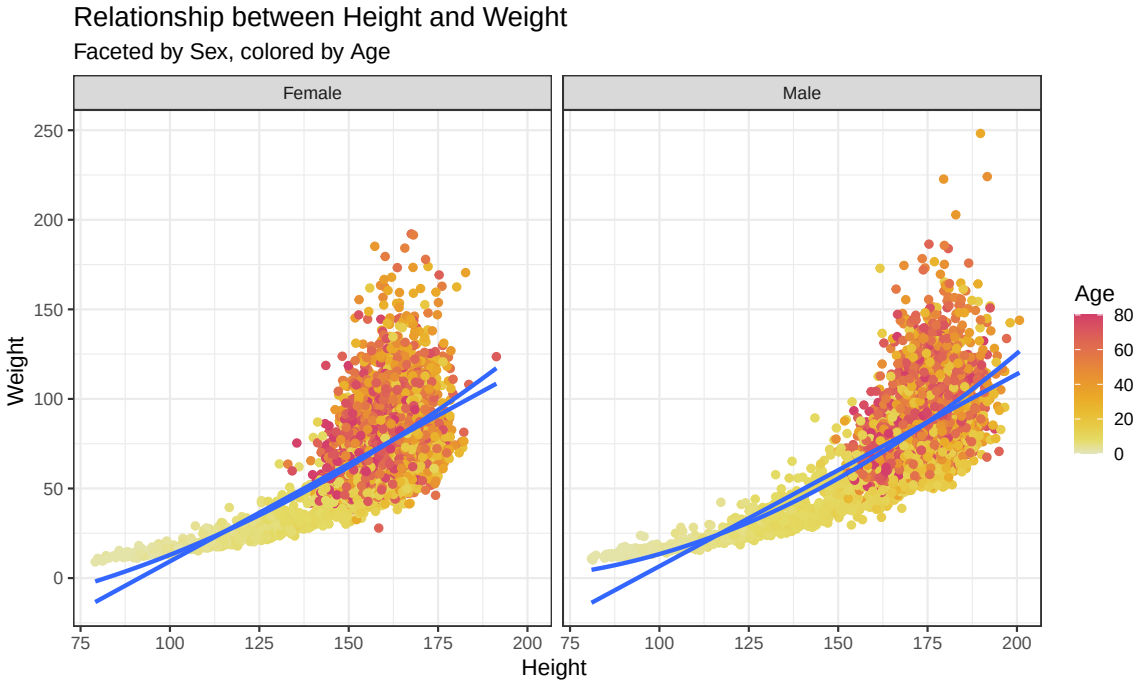
Investigate the relationship between bodyweight, height and age in the data in `height_weight.csv`. Answer the following questions, supporting your answers with appropriate polished graphs.

a. Make a plots comparing height and weight, colored by age, and faceted by gender. Describe how weight varies as a function of height (e.g. is it approximiately linear, quadratic, etc?). Also comment on the variance of weight given age at different ages.

b. Find a transformation of weight that makes the relationship between height and weight more linear. Make the above plots replacing weight with the transformed weight.

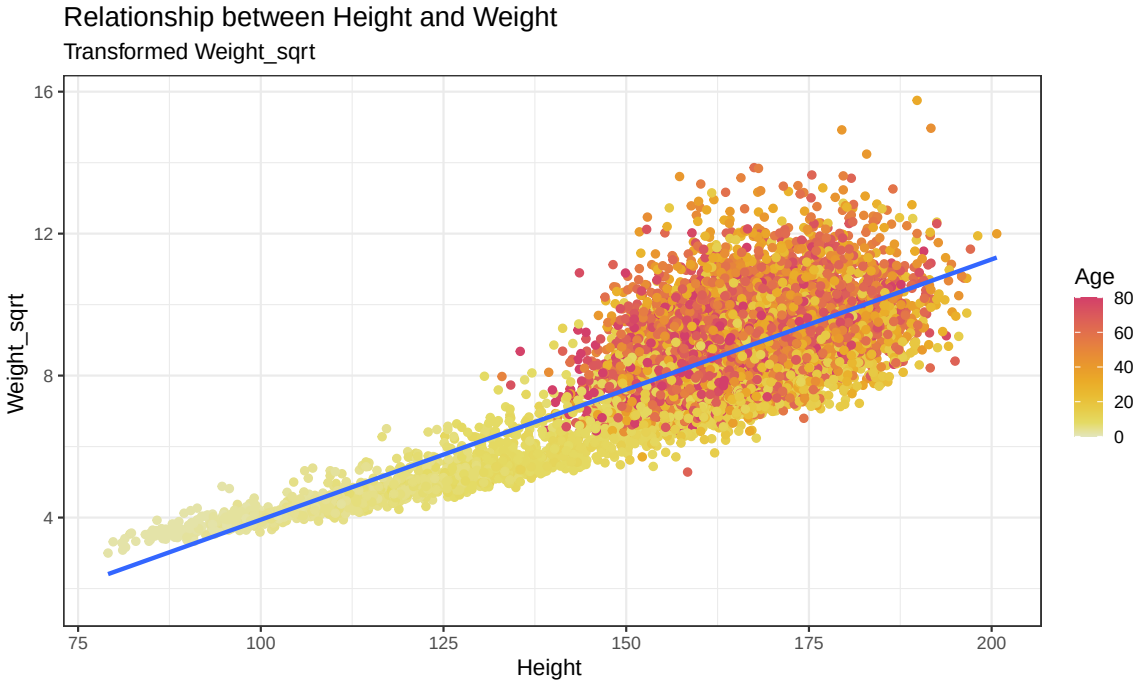
c. At younger ages, the average heights of girls and boys are the same. At what age does the average height of boys diverge from the average height of girls? Support your answer with a clear and well constructed plot. Hint: you may want to use `geom_smooth` to more clearly visualize trends.

```
body_df <- read_csv("height_weight.csv")
# Your R code here
ggplot(body_df, aes(x = height, y = weight, color = age)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE) +
  geom_smooth(method = "lm", formula = y ~ poly(x, 2), se = FALSE) +
  facet_wrap(~sex) +
  scale_color_continuous_sequential(palette = "Heat 2")+
  labs(
    title = "Relationship between Height and Weight",
    subtitle = "Faceted by Sex, colored by Age",
    x = "Height ",
    y = "Weight ",
    color = "Age "
  ) +
  theme_bw(base_size = 14)
```

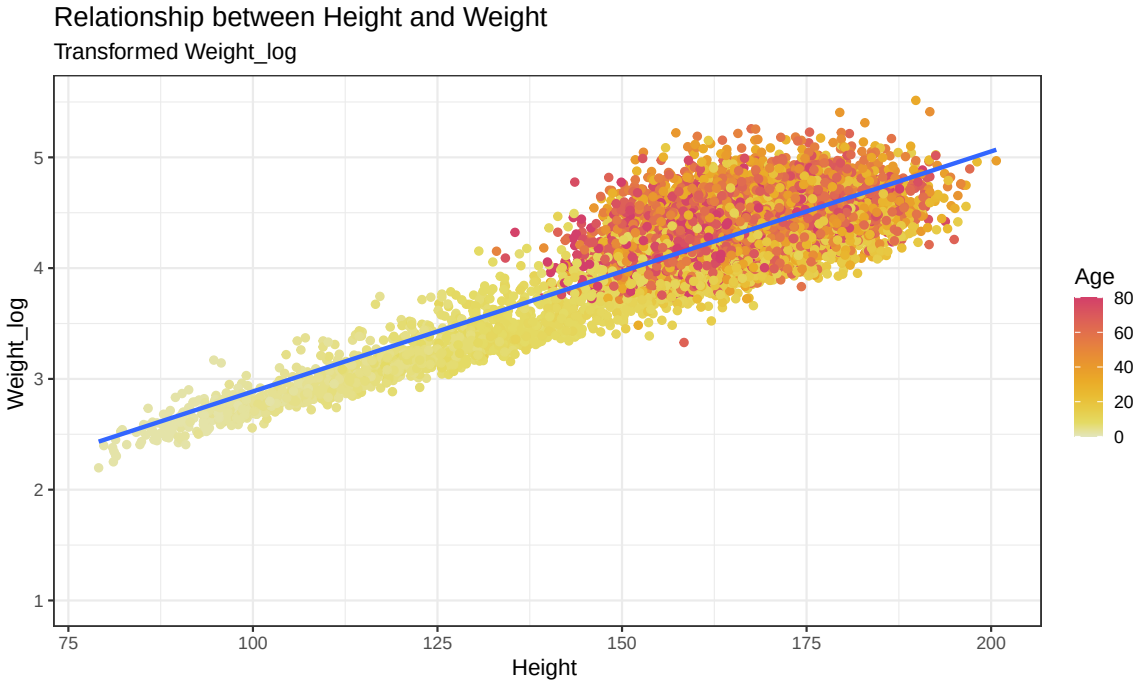


The plot displays two regression lines for each sex, representing both linear and quadratic relationships between age and weight. The quadratic model demonstrates superior fit compared to the linear model for both male and female subjects. Additionally, the analysis reveals heteroscedasticity in the data, as the variance of weight conditional on age is not constant and appears to decrease with increasing age.

```
body_df <- read_csv("height_weight.csv")
body_df_trans <- body_df %>%
  mutate(
    weight_sqrt   = sqrt(weight),
    weight_log    = log(weight),
  )

ggplot(body_df_trans, aes(x = height, y = weight_sqrt, color = age)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE) +
  scale_color_continuous_sequential(palette = "Heat 2")+
  labs(
    title = "Relationship between Height and Weight",
    subtitle = "Transformed Weight_sqrt",
    x = "Height ",
    y = "Weight_sqrt ",
    color = "Age "
  ) +
  theme_bw(base_size = 14)
```

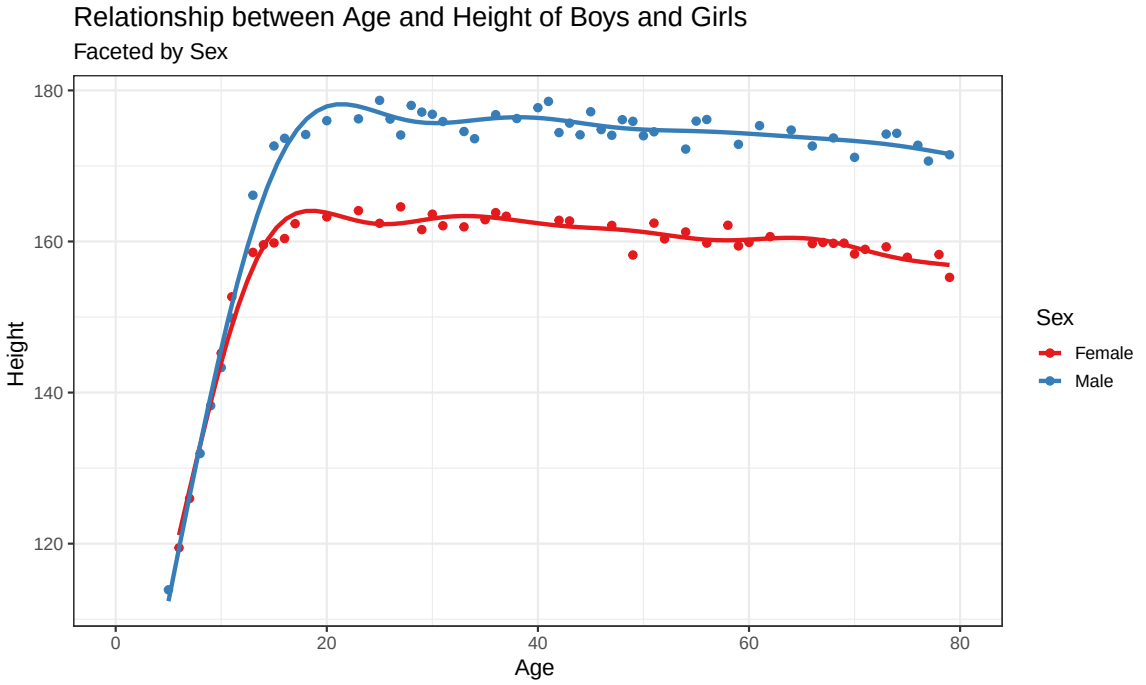



```
ggplot(body_df_trans, aes(x = height, y = weight_log, color = age)) +
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE) +
  scale_color_continuous_sequential(palette = "Heat 2")+
  labs(
    title = "Relationship between Height and Weight",
    subtitle = "Transformed Weight_log",
    x = "Height ",
    y = "Weight_log ",
    color = "Age "
  ) +
  theme_bw(base_size = 14)
```



The first plot transforms the weight to the square root of the weight, and the second plot transforms the weight to the natural logarithm of the weight. Both transformations make the relationship between height and weight more linear. However, the natural logarithm transformation is more linear than the square root transformation. As we can see from the plots, the points are more tightly clustered around the regression line in the natural logarithm transformation.

```
body_df <- read_csv("height_weight.csv")
summary_date <- body_df %>%
  group_by(sex, age) %>%
  summarise(mean_height = mean(height))
ggplot(summary_date, aes(x = age, y = mean_height, color = sex)) +
  geom_point() +
  geom_smooth(method = "gam", formula = y ~ s(x), se = FALSE) +
  scale_color_discrete(palette = "Set1")+
  labs(
    title = "Relationship between Age and Height of Boys and Girls",
    subtitle = "Faceted by Sex",
    x = "Age ",
    y = "Height ",
    color = "Sex "
  ) +
  theme_bw(base_size = 14)
```



As we can see from the plot, the average height of boys and girls is the same at younger ages. However, the average height of boys diverges from the average height of girls at nearly 14 years old.

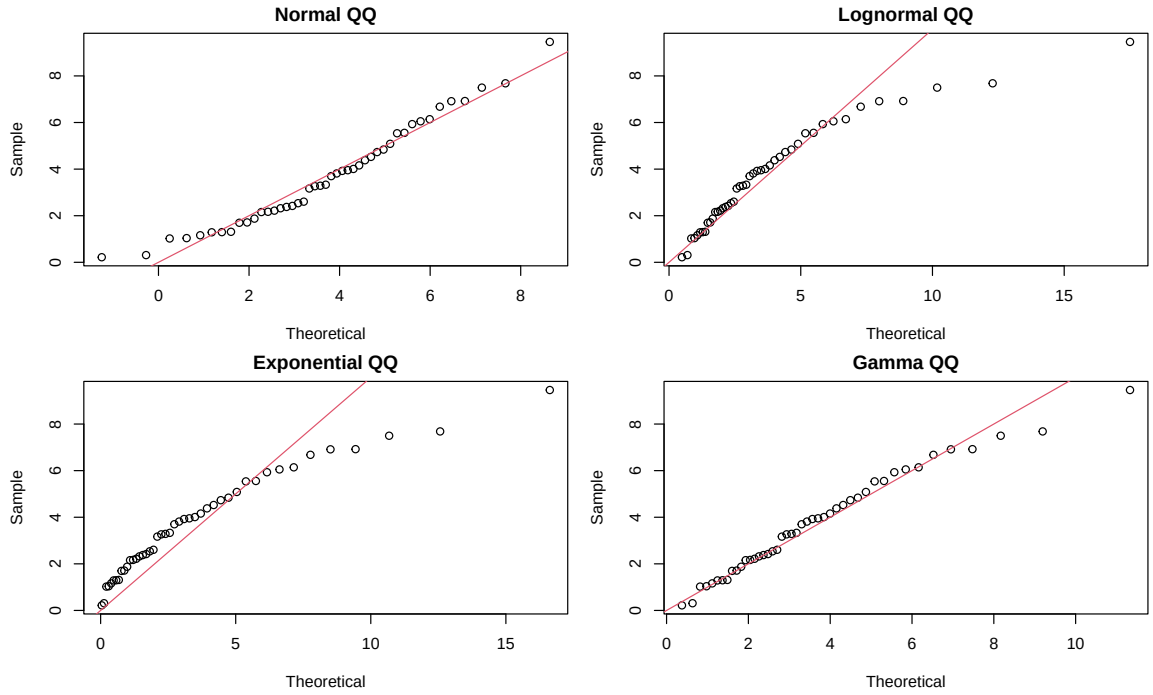
Question 2 (20 pt)

The following data are failure times in hours of 45 transmissions from caterpillar tractors belonging to a particular American company:

```
failure_times <- c(4381, 3953, 2603, 2320, 1161, 3286, 6914, 4007, 3168,
                   2376, 7498, 3923, 9460, 4525, 2168, 1288, 5085, 2217,
                   6922, 218, 1309, 1875, 1023, 1697, 1038, 3699, 6142,
                   4732, 3330, 4159, 2537, 3814, 2157, 7683, 5539, 4839,
                   6052, 2420, 5556, 309, 1295, 3266, 6679, 1711, 5931)
```

Use QQ-plots to examine the applicability of the following models for the probability distribution of failure time: normal, lognormal, exponential and Gamma (hint: check out the function `fitdistr` in the library MASS to fit these distributions. You may rescale the data if you have numerical problems). For the model that fits best (explain how you determine which model fits best), plot the PDF and the kernel density estimate of the data on the same plot.

```r
# Your R code library(MASS)
library(MASS)
x <- failure_times / 1000

fit_norm <- fitdistr(x, "normal")
fit_logn <- fitdistr(x, "lognormal")
fit_exp  <- fitdistr(x, "exponential")
shape0 <- mean(x)^2 / var(x)
rate0  <- mean(x) / var(x)
fit_gam  <- fitdistr(x, "gamma", start = list(shape = shape0, rate = rate0))

op <- par(mfrow=c(2,2), mar=c(4,4,2,1))
n <- length(x); p <- ppoints(n); xs <- sort(x)

qN <- qnorm(p, mean=fit_norm$estimate["mean"], sd=fit_norm$estimate["sd"])
plot(qN, xs, main="Normal QQ", xlab="Theoretical", ylab="Sample"); abline(0,1,col=2)

qLN <- qlnorm(p, meanlog=fit_logn$estimate["meanlog"], sdlog=fit_logn$estimate["sdlog"])
plot(qLN, xs, main="Lognormal QQ", xlab="Theoretical", ylab="Sample"); abline(0,1,col=2)

qE <- qexp(p, rate=fit_exp$estimate["rate"])
plot(qE, xs, main="Exponential QQ", xlab="Theoretical", ylab="Sample"); abline(0,1,col=2)

qG <- qgamma(p, shape=fit_gam$estimate["shape"], rate=fit_gam$estimate["rate"])
plot(qG, xs, main="Gamma QQ", xlab="Theoretical", ylab="Sample"); abline(0,1,col=2)
```




```
par(op)
aic <- c(
  normal     = 2*2 - 2*fit_norm$loglik,
  lognormal  = 2*2 - 2*fit_logn$loglik,
  exponential= 2*1 - 2*fit_exp$loglik,
  gamma      = 2*2 - 2*fit_gam$loglik
)
aic
```

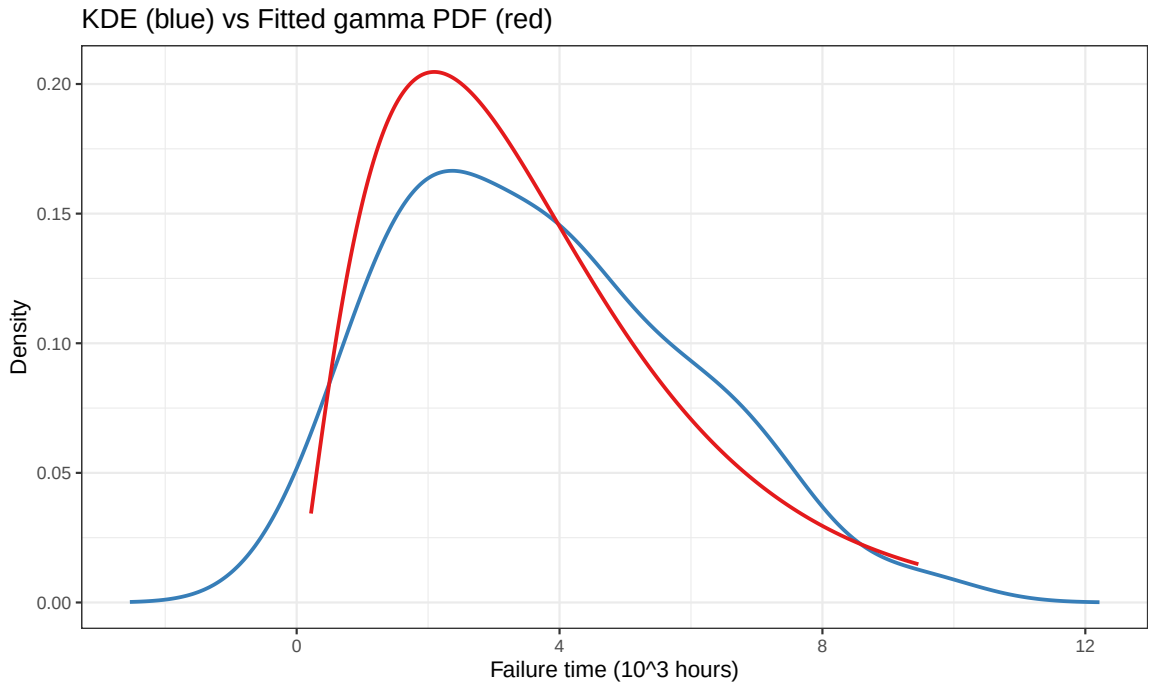
```
     normal   lognormal exponential       gamma
   201.1754    206.2696    209.6228    197.2365
```

```
best <- names(which.min(aic))
best
```

```
[1] "gamma"
```

```
dens <- density(x)
grid <- seq(min(x), max(x), length.out = 400)

pdf_y <- switch(best,
  normal       = dnorm(grid, mean = fit_norm$estimate["mean"], sd = fit_norm$estimate["sd"]),
  lognormal    = dlnorm(grid, meanlog = fit_logn$estimate["meanlog"], sdlog = fit_logn$estim
```

```
  exponential  = dexp(grid, rate = fit_exp$estimate["rate"]),
  gamma        = dgamma(grid, shape = fit_gam$estimate["shape"], rate = fit_gam$estimate["ra
)

ggplot() +
  geom_line(aes(x = dens$x, y = dens$y), color = "#377eb8", linewidth = 1.1) +
  geom_line(aes(x = grid, y = pdf_y), color = "#e41a1c", linewidth = 1.1) +
  labs(
    title = paste("KDE (blue) vs Fitted", best, "PDF (red)"),
    x = "Failure time (10^3 hours)",
    y = "Density"
  ) +
  theme_bw(base_size = 14)
```



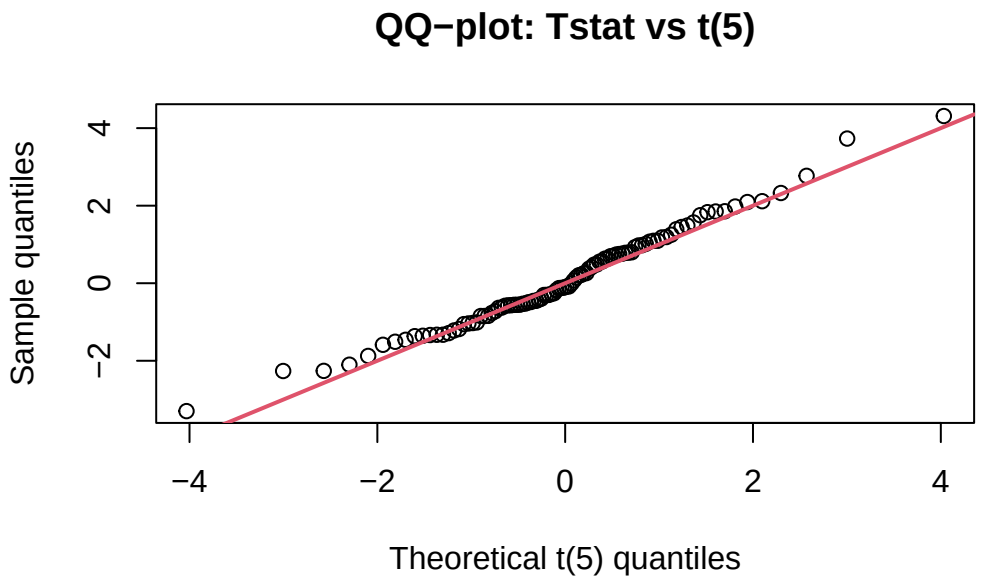
Model choice is based on joint criteria: (i) visual conformity in QQ-plots (closest to the 45-degree line, minimal systematic departures), and (ii) the smallest AIC among Normal, Lognormal, Exponential, and Gamma candidates. As we can see from the AIC values, the gamma distribution fits the best, the distribution. KDE (blue) and fitted Gamma PDF (red) for failure time (10^3 hours). The Gamma model matches the overall shape and right tail well, with a slight deviation near the peak.

Question 3 (15 pt)

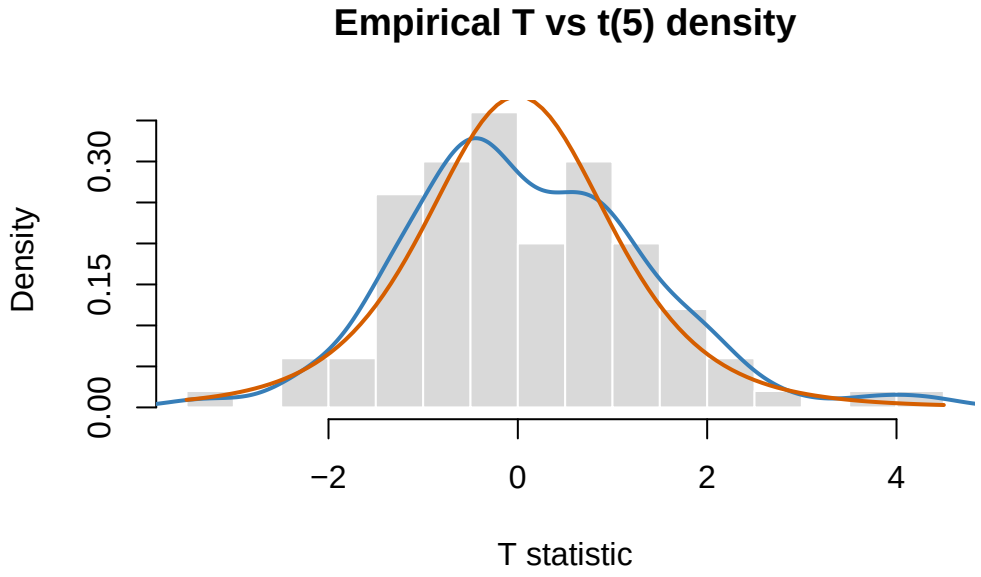
Generate 600 random samples from the normal distribution with mean 10 and standard deviation 5. Divide these 600 samples into 100 groups each with 6 samples. Compute the statistic $(\bar{X} - 10)/\sqrt{S^2/6}$ for each group. What kind of distribution do you expect this statistic to follow?

Using the 100 such statistics verify that the empirical distribution of these statistics actually follows the expected distribution.

```r
# Your R code here
set.seed(604)
n_groups <- 100; n <- 6; mu <- 10; sigma <- 5

x <- rnorm(n_groups * n, mean = mu, sd = sigma)
m <- matrix(x, ncol = n, byrow = TRUE)

xbar <- rowMeans(m)
s2   <- apply(m, 1, var)
Tstat <- (xbar - mu) / sqrt(s2 / n)

df <- n - 1

p <- ppoints(length(Tstat))
theo <- qt(p, df = df)
plot(theo, sort(Tstat),
     xlab = "Theoretical t(5) quantiles", ylab = "Sample quantiles",
     main = "QQ-plot: Tstat vs t(5)")
abline(0, 1, col = 2, lwd = 2)
```




```
hist(Tstat, prob = TRUE, breaks = 20, col = "grey85", border = "white",
     main = "Empirical T vs t(5) density", xlab = "T statistic")
lines(density(Tstat), col = "#377eb8", lwd = 2)
curve(dt(x, df = df), add = TRUE, col = "#D55E00", lwd = 2)
```



```r
ks.test(Tstat, "pt", df = df)
```

```
    Asymptotic one-sample Kolmogorov-Smirnov test

data:  Tstat
D = 0.072616, p-value = 0.6674
alternative hypothesis: two-sided
```

Since we are using the sample standard deviation S to estimate the population standard deviation, the statistic $T = (\bar{X} - 10)/\sqrt{S^2/6}$ follows a t-distribution with $n - 1 = 5$ degrees of freedom.

In this experiment, we generated 600 $\mathrm{N}(10, 5^2)$ samples, split them into 100 groups of six, and computed $\mathrm{T} = (\bar{X} - 10)/\sqrt{S^2/6}$ per group, which theoretically follows a t distribution with 5 degrees of freedom. The QQ-plot of the 100 T values against t5 quantiles lies close to the 45° line with only slight tail deviations, and the histogram/KDE matches the t5 density well; a KS test (D *approx* 0.073, p *approx* 0.667) likewise fails to reject t5. Overall, the empirical evidence is consistent with $\mathrm{T} \sim \mathrm{t}(5)$.

Question 4 (40 pt)

This question considers the analysis of U.S. birth data. The data consists of the numbers of infants born in each month from 2016 through 2023 separated by state and race/ethnicity. The data also includes the number of women age 15-44 in each state and race/ethnicity (the `denom` variables). The prefix in the column variables indicates the race/ethnicity and the suffix indicates whether the variable tracks births or number of women.

```r
birth_data <- read_csv("birth_data-1.csv", show_col_types = FALSE)
head(birth_data)
```

```
# A tibble: 6 x 14
  state  state_code  year month all_births white_births black_births hisp_births
  <chr>       <dbl> <dbl> <chr>      <dbl>        <dbl>        <dbl>       <dbl>
1 Alaba~          1  2016 Janu~       4805         2786         1462         381
2 Alaba~          1  2016 Febr~       4718         2846         1381         349
3 Alaba~          1  2016 March       4825         2839         1430         404
4 Alaba~          1  2016 April       4527         2730         1305         332
5 Alaba~          1  2016 May         4802         2908         1400         362
6 Alaba~          1  2016 June        5047         3077         1450         364
# i 6 more variables: all_denom <dbl>, white_denom <dbl>, black_denom <dbl>,
#   hisp_denom <dbl>, otherrace_births <dbl>, otherrace_denom <dbl>
```

a. Use `pivot_longer` to make the data tidy. Hint: you need to simultaneously pivot on `birth` columns and `denom` columns. To do so read the documentation for `pivot_longer` and use `.value` in the `names_to` argument (see e.g. the last example in the documentation). It might also help to use `names_sep`. After tidying, your data set should have 7 columns: `state`, `state_code`, `year`, `month`, `race`, `births` and `denom`. Print the first 10 rows and last 10 rows of your tidy data.

```
# Your code here
tidy_birth <- birth_data %>%
  pivot_longer(
    cols = ends_with("_births") | ends_with("_denom"),
    names_to = c("race", ".value"),
    names_sep = "_"
  )

cat("First 10 rows:\n")
```

```
First 10 rows:
```

```
head(tidy_birth, 10)
```

```
# A tibble: 10 x 7
   state   state_code  year month    race      births  denom
   <chr>        <dbl> <dbl> <chr>    <chr>      <dbl>  <dbl>
 1 Alabama          1  2016 January  all         4805 952796
 2 Alabama          1  2016 January  white       2786 580142
 3 Alabama          1  2016 January  black       1462 292632
 4 Alabama          1  2016 January  hisp         381  43095
 5 Alabama          1  2016 January  otherrace    176  36927
 6 Alabama          1  2016 February all         4718 952796
 7 Alabama          1  2016 February white       2846 580142
 8 Alabama          1  2016 February black       1381 292632
 9 Alabama          1  2016 February hisp         349  43095
10 Alabama          1  2016 February otherrace    142  36927
```

```
cat("\nLast 10 rows:\n")
```

```
Last 10 rows:
```

```
tail(tidy_birth, 10)
```

```
# A tibble: 10 x 7
   state   state_code  year month    race      births denom
   <chr>        <dbl> <dbl> <chr>    <chr>      <dbl> <dbl>
 1 Wyoming         56  2023 November all          469    NA
 2 Wyoming         56  2023 November white        349    NA
 3 Wyoming         56  2023 November black         NA    NA
 4 Wyoming         56  2023 November hisp          68    NA
 5 Wyoming         56  2023 November otherrace     NA    NA
 6 Wyoming         56  2023 December all            0    NA
 7 Wyoming         56  2023 December white          0    NA
 8 Wyoming         56  2023 December black          0    NA
 9 Wyoming         56  2023 December hisp           0    NA
10 Wyoming         56  2023 December otherrace      0    NA
```

```r
cat("\nData dimensions:", dim(tidy_birth), "\n")
```

```
Data dimensions: 24480 7
```

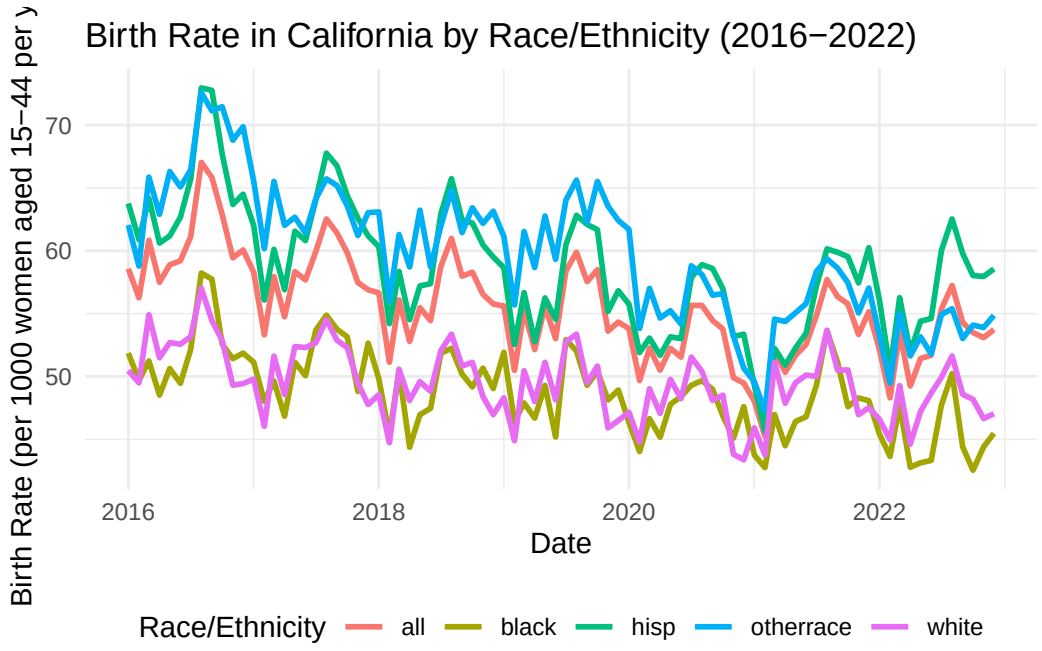
```r
cat("Column names:", names(tidy_birth), "\n")
```

```
Column names: state state_code year month race births denom
```

b. Create two new variable: `date` and `birth_rate`. To create `date`, use the `my` function from the `lubridate` package which takes a string consisting of the month followed by the year and returns the appropriate date object. `birth_rate` should have units of births per 1000 women 15-44 per year (note: don't forget to adjust for the fact that we have monthly data. As a reference, the national fertility rate is about 55 births per 1000 women aged 15-44 per year). Use the tidy data you just computed to plot the birth rate in California vs `date`. You should have lines with distinct colors for the birth rate in each race and overall. Which race/ethnicities tend to have the highest birth rates? Lowest?

```r
## Your code here
library(lubridate)


tidy_birth_clean <- tidy_birth %>%
  filter(!is.na(denom), !is.na(births))

tidy_birth_clean <- tidy_birth_clean %>%
  mutate(
    date = my(paste(month, year)),
```

```
    birth_rate = (births / denom) * 1000 * 12
  )

california_data <- tidy_birth_clean %>%
  filter(state == "California")

ggplot(california_data, aes(x = date, y = birth_rate, color = race)) +
  geom_line(linewidth = 1) +
  labs(
    title = "Birth Rate in California by Race/Ethnicity (2016-2022)",
    x = "Date",
    y = "Birth Rate (per 1000 women aged 15-44 per year)",
    color = "Race/Ethnicity"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```




```
california_summary <- california_data %>%
  group_by(race) %>%
  summarize(avg_birth_rate = mean(birth_rate, na.rm = TRUE)) %>%
  arrange(desc(avg_birth_rate))

print(california_summary)
```

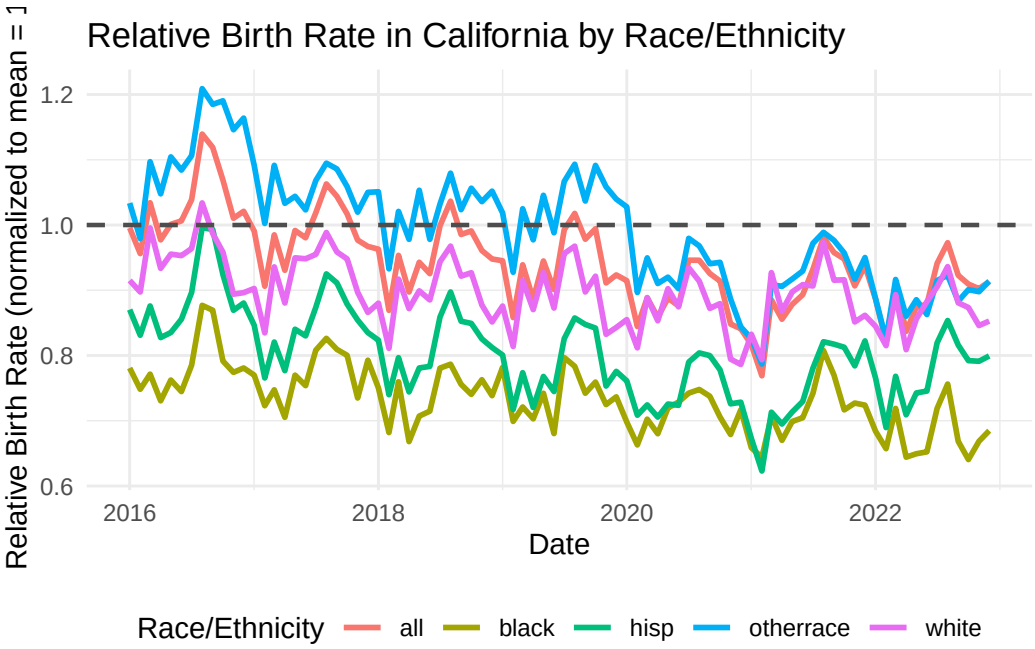
```
# A tibble: 5 x 2
```

```
  race      avg_birth_rate
  <chr>              <dbl>
1 otherrace           59.7
2 hisp                58.7
3 all                 55.5
4 white               49.4
5 black               48.7
```

The highest birth rates are observed in Hispanic populations, followed by other races, while Black populations have the lowest rates, with White populations also at the lower end. This difference is consistent across all races, suggesting a broader pattern rather than a regional variation. The seasonal fluctuations indicate periodic variations in birth rates across months, which could be attributed to seasonal changes in healthcare access or societal norms.

c. Create a new variable using `mutate` which corresponds to the *relative* birth rate for each race category. The relative birth rate should be computed by grouping by race, taking the birth rate and dividing by the mean birth rate for that race over the full range of data and then ungrouping again. Plot the relative birth rates for all races on the same plot. What new observations do you make about the data when plotting the relative birth rate? Are any patterns clearer in this plot than they were in the previous plot?

```r
tidy_birth_relative <- tidy_birth_clean %>%
  group_by(race) %>%
  mutate(relative_birth_rate = birth_rate / mean(birth_rate)) %>%
  ungroup()


california_relative <- tidy_birth_relative %>%
  filter(state == "California")

ggplot(california_relative, aes(x = date, y = relative_birth_rate, color = race)) +
  geom_line(linewidth = 1) +
  geom_hline(yintercept = 1, linetype = "dashed", color = "gray30", linewidth = 0.8) +
  labs(
    title = "Relative Birth Rate in California by Race/Ethnicity",
    x = "Date",
    y = "Relative Birth Rate (normalized to mean = 1)",
    color = "Race/Ethnicity"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```

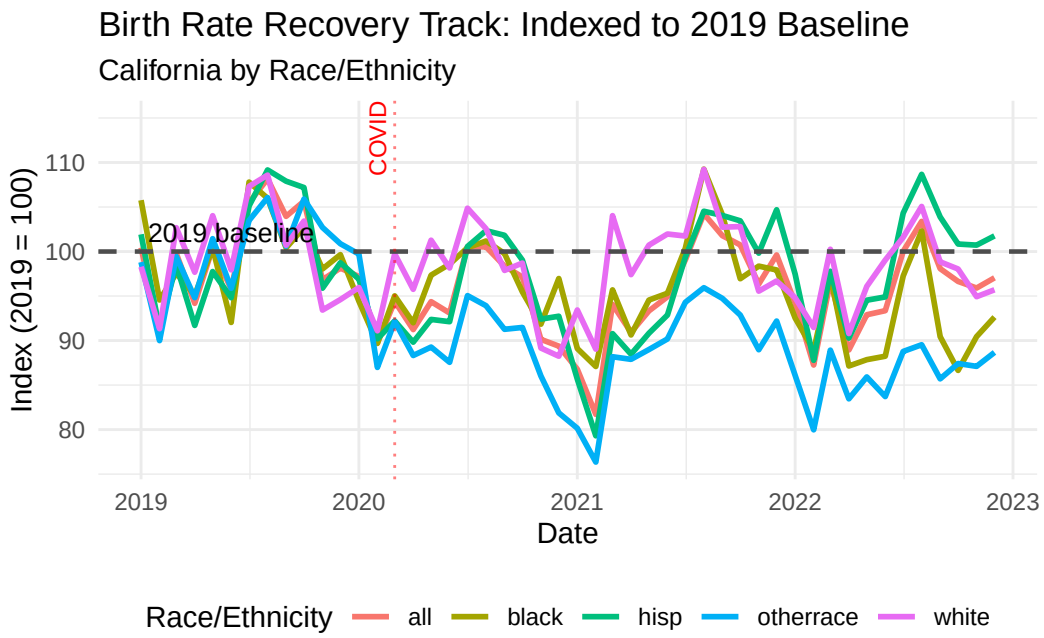


Relative rates make patterns clearer: all groups move together with similar seasonality and a clear drop around 2020 (a common shock, likely COVID-19). Over time many series shift from above 1 in 2016 to below 1 by 2022, indicating a broad decline. The "otherrace" series varies the most, while the others are steadier. Normalization shifts the focus from level differences to proportional changes over time.

d. Make a visualization of your choice that clearly highlights something about the data that was not evident in the previous plots. For example, you can explore seasonality in the trends or variation across race and/or states. Describe what you learned from your plot. The best visualizations will be shared in class (given your approval).

```
recovery_data <- tidy_birth_relative %>%
  filter(state == "California", year >= 2019) %>%
  group_by(race) %>%
  mutate(
    baseline_2019 = mean(birth_rate[year == 2019], na.rm = TRUE),
    recovery_index = birth_rate / baseline_2019 * 100
  ) %>%
  ungroup()

ggplot(recovery_data, aes(x = date, y = recovery_index, color = race)) +
  geom_line(linewidth = 1) +
  geom_hline(yintercept = 100, linetype = "dashed", color = "gray30", linewidth = 0.8) +
  geom_vline(xintercept = as.Date("2020-03-01"), linetype = "dotted",
             color = "red", alpha = 0.5) +
  annotate("text", x = as.Date("2019-06-01"), y = 100,
```

```
          label = "2019 baseline", vjust = -0.5, size = 3.5) +
annotate("text", x = as.Date("2020-03-01"), y = 115,
          label = "COVID-19", angle = 90, vjust = -0.5, size = 3, color = "red") +
labs(
  title = "Birth Rate Recovery Track: Indexed to 2019 Baseline",
  subtitle = "California by Race/Ethnicity",
  x = "Date",
  y = "Index (2019 = 100)",
  color = "Race/Ethnicity"
) +
theme_minimal() +
theme(legend.position = "bottom")
```



The recovery track indexed to 2019 reveals important differences in COVID-19 impact and recovery across racial groups. A sharp drop in early 2021 affected all groups, with some experiencing declines to 75-80% of their 2019 baseline. White and Hispanic populations show gradual recovery, with rates approaching or reaching 100% by late 2022, suggesting resilience to the pandemic shock. In contrast, Black and "otherrace" populations remain persistently below baseline throughout the period, indicating incomplete recovery. Another notable drop occurs in early 2022 across all groups, possibly reflecting delayed pandemic effects or other external factors. This visualization highlights that while the pandemic initially affected all groups, recovery has been uneven, with some racial groups returning to pre-pandemic fertility levels while others continue to experience suppressed birth rates.