DH 302 Spring 2025 Assignment 01

The assignment is based on Lecture 2, Lecture 3, Lecture 4, Lecture 5, and Lecture 6. Due at 11:59PM (IST), Monday 3^{rd} February, 2025 via Gradescope. Total points = 100

Mradul Sonkar (23B0980)

Instructions

Submit your solutions via gradescope by 11:59 PM (IST) Monday, 3^{rd} February 2025. In-person submissions will not be entertained. Please upload a single PDF file. Late submissions are allowed with a 10% per day penalty. You can raise your questions related to the assignment on Piazza - please tag these as `assignment_01`.

- For theory questions, you can either write your response for in latex or put a screenshot/picture of your handwritten response in the appropriate section. To embed scanned images, use this format: `![question1](/path/to/question1.png)` where `/path/to/question1.png` is the local path (on your laptop) to your scanned (handwritten) response to question1.
- If you are writing the solutions for theory questions by hand please use a pen. Pencil submissions are difficult to read when scanned. You will have to take a scan for each such answer and embed it in this document.
- Your final submission has to be the PDF that comes from this template - one single pdf. No Exceptions.
- Please mention the name(s) of people you collaborated with and what exactly you discussed.

Making your submission: You can download the submission template from here. Open the template in Rstudio (you will need to ensure Quarto is installed). Once you are done with your answers, use the "render" (arrow like) button on the toolbar to create a pdf. Only pdf submissions are allowed.

Question 01 [10 points]

From L to LL to ML: Let $X = (X_1, X_2, \ldots, X_n)$ be independent and identically distributed (IID) observations coming from a poisson distribution with (unknown) parameter λ. Derive a likelihood estimate $\hat{lambda}$ for the n observations.

Solution:

The Likelihood function for poisson distribution can be written as

$$L(\lambda) = \prod_{i=1}^{n} \frac{\lambda^{X_i} e^{-\lambda}}{X_i!}$$

We will use log likelihood function to calculate the value at which the likelihood function attains its maximum value.

$$\begin{aligned} log(L(\lambda)) &= log(\prod_{i=1}^{n} \frac{\lambda^{X_i} e^{-\lambda}}{X_i!}) \\ &= \sum_{i=1}^{n} log(\frac{\lambda^{X_i} e^{-\lambda}}{X_i!}) \\ &= \sum_{X_i}^{n} (X_i log(\lambda) - \lambda - log(X_i!)) \end{aligned}$$

Now let us differentiate the log likelihood function to find the estimation $\hat{\lambda}$ of λ:

$$\begin{aligned} \frac{d}{d\lambda} log(L(\lambda)) &= \frac{d}{d\lambda} \sum_{X_i}^{n} (X_i log(\lambda) - \lambda - log(X_i!)) \\ &= \sum_{i=1}^{n} (\frac{X_i}{\lambda} - 1) \end{aligned}$$

Putting the derivative equal to 0 gives us that:

$$\hat{\lambda} = \sum_{i=1}^{n} \frac{X_i}{N}$$

We can verify that this value is indeed correct by calculating the double derivative of log likelihood function at $\hat{\lambda}$, which comes out to be $-\frac{N^2}{\sum_{i=1}^{n} X_i}$. A negative value suggests that function attains it's maximum at $\hat{\lambda}$.

Question 02 [10 points]

Climbing the L landscape: Simulating a poisson random variable in R is very easy. You can use the `rpois()` function to simulate. Write R code to define a function to output the log likelihood function of a poisson random variable. Additionally write R code to show the maximum likelihood estimate.

Solution:

```
logLikelihood <- function (dataVector, lambda) {
  res <- 0
  for (xi in dataVector) {
    res <- res + xi * log(lambda) - lambda - log(factorial(xi))
  }
  return (res)
}
```

The maximum likelihood estimate is just the mean of observed values.

```
mleEstimate <- function (dataVector) {
  res <- 0
  for (xi in dataVector) {
    res <- res + xi
  }
  return (res/length(dataVector))
}
```

Question 03 [10 points]

Limiting distributions magic: The PMF of binomial random variable is given by $P(X = x) = \binom{N}{x} p^x (1-p)^{N-x}$). A poisson random variable on the other hand is given by $P(Y = y) = \lambda^y \frac{e^{-y}}{y!}$. In the class, we dervied a normal approximation to the poisson. Based on the below zoo of statistical models, it is possible to approximate a binomial model as a poisson. Write down the steps involved for arriving at this approximation.

Solution:

Let us denote the PDF of Binomial Random Variable by $P(n, \theta, x)$. We know that the PDF can be expressed as follows:

$$P(n, \theta, x) = \binom{n}{x} \theta^x (1-\theta)^{n-x}$$

We will begin by assuming that $\theta = \frac{\lambda}{n}$. Replacing the value of θ gives us the following expression:

$$\begin{aligned}
P(n, \frac{\lambda}{n}, x) &= \binom{n}{x} (\frac{\lambda}{n})^x (1-\frac{\lambda}{n})^{n-x} \\
&= (\frac{n!}{x!(n-x)!})(\frac{\lambda^x}{n^x})(1-\frac{\lambda}{n})^n (1-\frac{\lambda}{n})^{-x} \\
&= (\frac{n(n-1)(n-2)\dots(n-x+1)}{x!})(\frac{\lambda^x}{n^x})(1-\frac{\lambda}{n})^n (1-\frac{\lambda}{n})^{-x} \\
&= (\frac{\lambda^x}{x!})[1(1-\frac{1}{n})(1-\frac{2}{n})\dots(1-\frac{x-1}{n})](1-\frac{\lambda}{n})^n (1-\frac{\lambda}{n})^{-x}
\end{aligned}$$

As the value of n approaches ∞, the Binomial distribution tends to Poisson distribution. We will calculate the limit of the above expression as n approaches ∞.

$$\begin{aligned}
Limit &= \lim_{n\to\infty} (\frac{\lambda^x}{x!})[1(1-\frac{1}{n})(1-\frac{2}{n})\dots(1-\frac{x-1}{n})](1-\frac{\lambda}{n})^n (1-\frac{\lambda}{n})^{-x} \\
&= (\frac{\lambda^x}{x!})((1)(1)\dots(1))e^{-\lambda}(1) \\
&= (\frac{\lambda^x}{x!})\ e^{-\lambda}
\end{aligned}$$

Hence, the Binomial distribution can be approximated as Poisson if the value of n is very large.

Question 04 [30 points]

Hirotogu's IC:. Japanese statistician Hirotugu Akaike, formulated a criterion famously called the "Akaike Information Criterion (AIC)". AIC provides an estimate of the relative "quality of

fit" of a model. Given a collection of models, AIC estimates the quality of each model relative to other models. A lower AIC value indicates better fit (think why).

We did not get to cover this in class, but it is a relatively easy concept to understand. AIC is given by:

$\text{AIC} = 2k - 2\log(\hat{L})$,

where k= Estimated number of parameters and $\hat{L}$= Maximum likelihood of the model.

Your goal in the problem is to estimate the best fit for modeling the number of deaths due to abortion in the year 2022. The data is available here. The code below tries to simulate a poisson and then figures out the best model making use of the `fitdistrplus` package. It also shows how the predictions of the three models looks like and figures out the best model using the AIC.

Now write code to show the best fit model among (only) poisson/normal/gamma for deaths due to abortion in 2009-2019 and then in 2019-2020. Since you need to write similar code for the two data frames, it might be worth writing a function to do this, something like:

```
DoFit <- function(x) {
  # Make your life easy by doing all standard operations
  # for fitting in this function

  # death counts
  death_counts <- x$abortion_deaths
  monyear <- x$monyear
  df.x <- data.frame(x = death_counts)

  # Fit normal
  fit_norm <- fitdistr(death_counts, densfun = "normal")
  norm.pdf <- dnorm(death_counts,
                mean = fit_norm$estimate[["mean"]],
                sd = fit_norm$estimate[["sd"]]
              )
  df.fit.norm <- data.frame(x = death_counts,
                            probability = norm.pdf)


  # fit poisson
  fit_poisson <- fitdistr(death_counts,
                          densfun = "poisson")
  pois.pdf <- dpois(death_counts,
                    lambda = fit_poisson$estimate)
  df.fit.pois <- data.frame(x = death_counts,
```

```
                          probability = pois.pdf)


# fit gamma
fit_gamma <- fitdistr(death_counts,
                      densfun = "gamma")
gamma.pdf <- dgamma(death_counts,
                    shape = fit_gamma$estimate[["shape"]],
                    rate = fit_gamma$estimate[["rate"]]
)
df.fit.gamma <- data.frame(x = death_counts,
                           probability = gamma.pdf)



# merging the three dataframes to get one dataframe
df.merged <- bind_rows(list(Normal = df.fit.norm,
                            Poisson = df.fit.pois,
                            Gamma = df.fit.gamma),
                       .id = "Type")



# plotting the three models
combined_plot <- ggplot(df.x, aes(x = x)) +

   geom_histogram(aes(
     y = after_stat(!!str2lang("density"))),
     binwidth = 6,
     fill = "gray",
     color = "black") +

  geom_point(data = df.merged, aes(x = x,
                                   y = probability,
                                   color = Type),
             size = 0.5) +

  geom_line(data = df.merged, aes(x = x,
                                  y = probability,
                                  color = Type),
            linewidth = 0.5) +
```

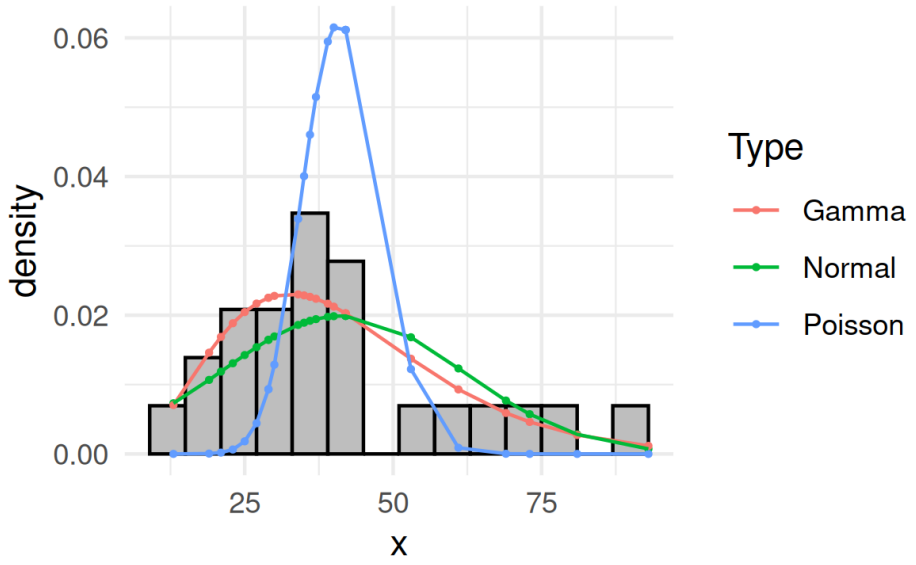
```
    theme_minimal()


  # aic <- AIC() ...

  aic_values <- c(
      Normal = AIC(fit_norm),
      Poisson = AIC(fit_poisson),
      Gamma = AIC(fit_gamma)
    )

    return (list(plot=combined_plot, aic=aic_values))
}
```

```
df_abortions <- read_tsv("India_abortion_deaths.tsv")
```

```
Rows: 180 Columns: 4
-- Column specification --------------------------------------------------------
Delimiter: "\t"
chr (2): monyear, month
dbl (2): year, abortion_deaths

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
deaths_2009_2010 <- df_abortions %>% filter(
  year %in% c(2009, 2010))
deaths_2019_2020 <- df_abortions %>% filter(
  year %in% c(2019, 2020))

# YOUR CODE HERE
fit_2009_2010 <- DoFit(deaths_2009_2010)
fit_2019_2020 <- DoFit(deaths_2019_2020)
```

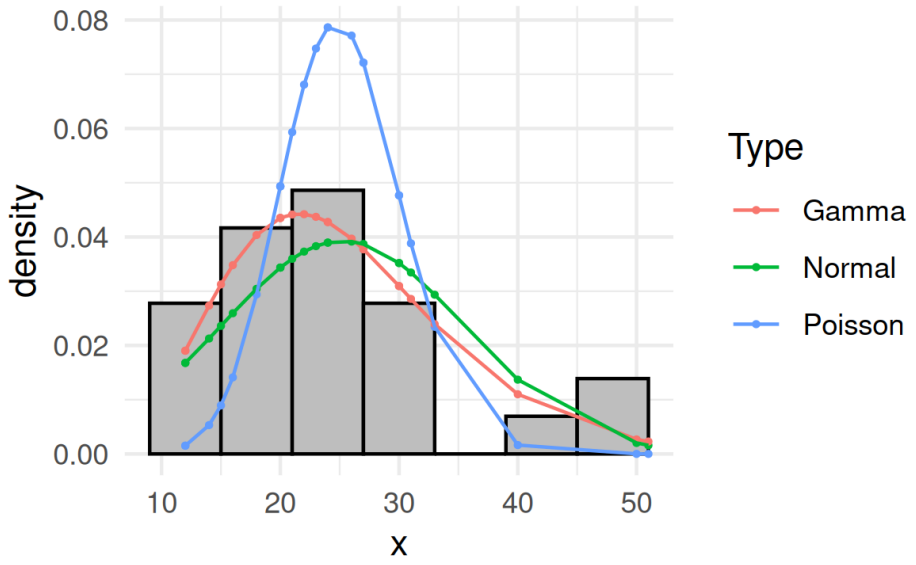
```
# !! DO NOT EDIT/REMOVE !!
fit_2009_2010$plot
```




```
# !! DO NOT EDIT/REMOVE !!
fit_2009_2010$aic
```

```
  Normal  Poisson    Gamma
216.0031 348.6892 209.9672
```

```
# !! DO NOT EDIT/REMOVE !!
fit_2019_2020$plot
```




```
# !! DO NOT EDIT/REMOVE !!
fit_2019_2020$aic
```

```
  Normal  Poisson    Gamma
183.4056 213.0916 178.3454
```

What is the most appropriate model for fitting 2009-2010 deaths and for 2019-2020 deaths?

- From the above computations, the best distribution that fits the data for *both* the years is **Gamma Distribution**. Also, Normal distribution is also doing a good job in fitting the data.

Question 05 [20 points]

You, the policy consultant: The government is interested to know if the distribution of deaths across the twelve months has changed in 10 years from 2009 to 2019. Formulate a null hypothesis to provide recomeondation to the government. Also write R code to test your hypothesis at a p-value threshold of 0.01.

A starter code to arrive at your data frame of interest is below:

```
df_abortions <- read_tsv("India_abortion_deaths.tsv")
```

```
Rows: 180 Columns: 4
-- Column specification --------------------------------------------------------
Delimiter: "\t"
chr (2): monyear, month
dbl (2): year, abortion_deaths

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
deaths_2009 <- df_abortions %>% filter(year %in% c(2009))
deaths_2019 <- df_abortions %>% filter(year %in% c(2019))
df_2009_and_2019 <- bind_rows(list(deaths_2009, deaths_2019))
df_wide <- pivot_wider(
  df_2009_and_2019 %>% dplyr::select(
  month, year, abortion_deaths),
  names_from = year, values_from = abortion_deaths
)
```

- **Null Hypothesis** (H_0) : The distribution of deaths across the years 2009 and 2019 is the same.
- **Alternative Hypothesis** (H_a) : The distribution of deaths across the years 2009 and 2019 has changed.

  We can perform **Chi-Square test** to find whether the distribution of deaths has changed from 2009 to 2019.

```r
# generating row totals for later use
df_wide <- df_wide %>% mutate(row_total = `2009` + `2019`)

# generating column totals for later use
col_totals <- colSums(df_wide[, c("2009", "2019")])

# total number of deaths in 2009 + 2019
total_deaths <- sum(col_totals)

# expected deaths
df_expected <- df_wide %>% mutate(
    expected_2009 = row_total * col_totals["2009"] / total_deaths,
    expected_2019 = row_total * col_totals["2019"] / total_deaths
)

# chi test statistic computation
```

```r
df_expected <- df_expected %>% mutate(
    chi_2009 = ((`2009` - expected_2009)^2) / expected_2009,
    chi_2019 = ((`2019` - expected_2019)^2) / expected_2019
)

chi_square_stat <- sum(df_expected$chi_2009 + df_expected$chi_2019)

# computing p-value
p_value <- pchisq(chi_square_stat, df = 11, lower.tail = FALSE)

print(chi_square_stat)
```

```
[1] 61.88155
```

```r
print(p_value)
```

```
[1] 4.1382e-09
```

- Since the p-value comes out to be very much less than significance level $\alpha = 0.01$, we can **reject** the null hypothesis, and claim that there is a distribution change from 2009 to 2019.

  For this question, I have taken reference from here.

Question 06 [10 points]

Scale me away: Given random variable $X \sim \mathcal{N}(0, 1)$ and Y poisson random variable with mean $\lambda = 5$, derive the the mean and variance of random variables $X' = 4X$ and $Y' = 5Y$. Write R code to draw the PMF of X' and Y' on For each random variable, your code should produce only one plot with different colors indicating X and X' in one plot and Y and Y' in other plot. Range of these random variables are different, use your best judgement to figure out what the x axis should be in each case

HINT: The continuous case of gaussian will require you to plot a PDF (not a PMF)

Derivation of parameters of X' and Y':

$$\begin{aligned} E(X') &= E(4 \times X) \\ &= 4 \times E(X) \\ &= 4 \times 0 \\ &= 0 \end{aligned}$$

$$
\begin{aligned}
Var(X') &= Var(4 \times X) \\
&= 4^2 \times E(X) \\
&= 4^2 \times 1 \\
&= 16
\end{aligned}
$$

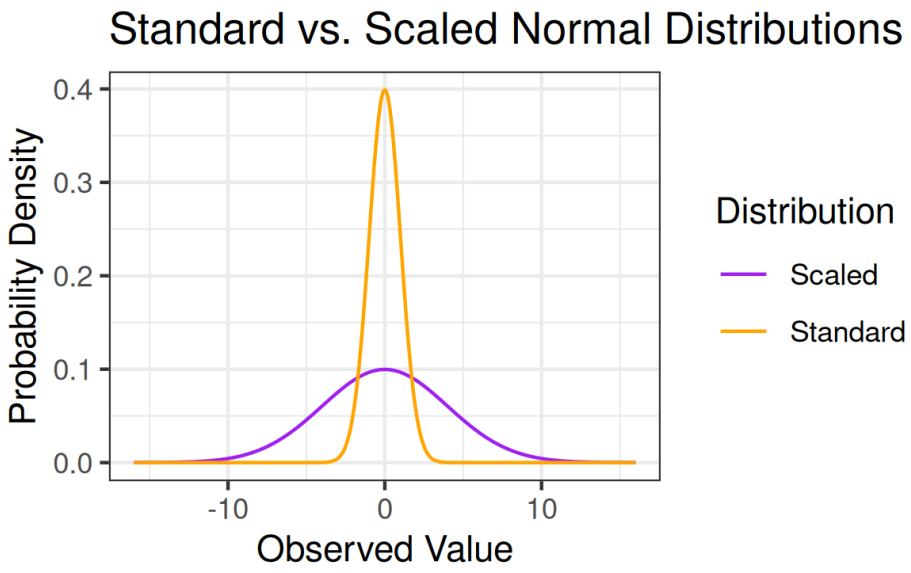
$$
\begin{aligned}
E(Y') &= E(5 \times Y) \\
&= 5 \times E(X) \\
&= 5 \times 5 \\
&= 25
\end{aligned}
$$

The result used in above derivations is ***linearity*** of expectation operator, stated here.

Therefore, $X' \sim N(0, 16)$ and $Y' \sim Pois(25)$.

```
# YOUR CODE AND PLOT HERE FOR GAUSSIAN

mu_standard = 0
sigma_standard = 1

mu_scaled = 0
sigma_scaled = 4

# generating the data for standard and scaled normals
x_standard <- seq(mu_standard - 16 * sigma_standard,
                  mu_standard + 16 * sigma_standard,
                  length.out = 1000)
x_scaled   <- seq(mu_scaled - 4 * sigma_scaled,
                  mu_scaled + 4 * sigma_scaled,
                  length.out = 1000)

# generating the pdf and dataframe for standard normal distribution
pdf_standard <- dnorm(x_standard,
                      mean = mu_standard,
                      sd = sigma_standard)
df_standard <- data.frame(observed = x_standard,
                          probability = pdf_standard)
df_standard$Distribution <- "Standard"
```

```r
# generating the pdf and dataframe for scaled normal distribution
pdf_scaled <- dnorm(x_scaled,
                    mean = mu_scaled,
                    sd = sigma_scaled)

df_scaled <- data.frame(observed = x_scaled,
                        probability = pdf_scaled)

df_scaled$Distribution <- "Scaled"


# combining the dataframes
df_combined <- rbind(df_standard, df_scaled)

# plotting
ggplot(df_combined, aes(x = observed,
                        y = probability,
                        color = Distribution)) +

  geom_line() +

  labs(title = "Standard vs. Scaled Normal Distributions",
       x = "Observed Value",
       y = "Probability Density") +

  scale_color_manual(values = c("Standard" = "orange", "Scaled" = "purple")) +

  theme_bw()
```

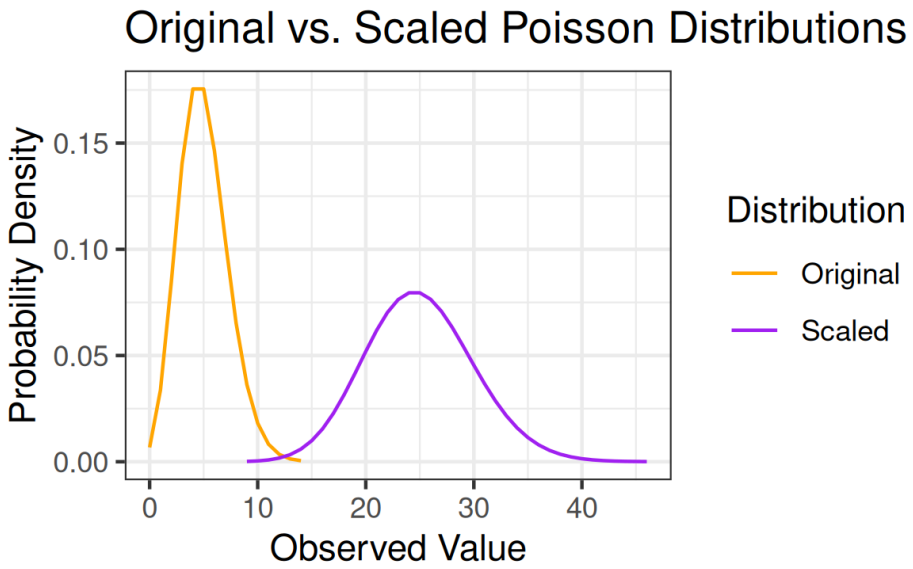


NOTE: The way I calculated the appropriate range is as follows:

The maximum number for which I am plotting X' is $x = \mu' + 4\sigma'$, which equals $x = \mu + 16\sigma'$. Hence, the range of first plot should be $[\mu - 16\sigma', \mu + 16\sigma']$.

```
# YOUR CODE AND PLOT HERE FOR POISSON

lambda_standard = 5
lambda_scaled = 25

# generating the data for original and scaled poisson distributions
x_standard <- rpois(n = 10000, lambda = lambda_standard)
x_scaled   <- rpois(n = 10000, lambda = lambda_scaled)

# generating the pdf and dataframe for original poisson distribution
pmf_standard <- dpois(x_standard, lambda = lambda_standard)
df_standard <- data.frame(x = x_standard, probability = pmf_standard)
df_standard$Distribution <- "Original"

# generating the pdf and dataframe for scaled poisson distribution
pmf_scaled <- dpois(x_scaled, lambda = lambda_scaled)
df_scaled <- data.frame(x = x_scaled, probability = pmf_scaled)
df_scaled$Distribution <- "Scaled"
```

```r
# combining the dataframes
df_combined <- rbind(df_standard, df_scaled)

# plotting
ggplot(df_combined, aes(x = x, y = probability, color = Distribution)) +
  geom_line() +
  labs(title = "Original vs. Scaled Poisson Distributions",
       x = "Observed Value",
       y = "Probability Density") +
  scale_color_manual(values = c("Original" = "orange", "Scaled" = "purple")) +
  theme_bw()
```



Question 07 [10 points]

Sherlocking the NEET fit: National-eligibility-cum-entrance-test has been in the news (formerly known as All India Pre-medical test or AIPMT) is a nation wide entrance exam conducted by the National Testing Agency for admission into the undergraduate (MBBS) medical program throughout the country.

NEET-UG-2024 was in the news even before the (original) results came out on June 4th. On 5th May 2024, the day of the exams, students complained of the paper having been leaked before the exam. NTA initially denied any leaks, but the Bihar police arrested a few peple involved with paper leaks in the next few days. WikiPedia has a good summary of events:

The matter ultimately reached the Supreme Court (SC). The SC ordered NTA to publish the entire center-wise records of the NEET-UG-2024 exam. NTA obliged, making the records available in PDF format here.

After painstakingly scraping 4750 PDFs (a potential assignment exercise for the future), the data is available for you in a simple format: CSV. For 2024, I was able to parse the data in a simple csv format for all the candidates who took the exam. For 2023, the raw data has not been published but I was able to parse out the relative frequency distribution using dome digitization tools (Which allow you to obtain raw data given a frequency/bar diagram).

Use the file `NTA_all_marks_2024.csv` available here to answer: What is the distribution (normal/non-normal/poisson/something else) for the marks obtained by students in 2024? A brief description of what the columns in each file represent:

- `Srlno:` Serial number of the candidate (unique for each center, but you should ideally not need this)
- `Marks:` Marks obtained in NEET-UG-2024 by the candidate
- `Center_number:` A unique center id (you do not need this for this analysis)

Write R code to determine what is the best statistical model for modeling 2024 marks. Is it poisson, normal, log normal (not covered in class, but not hard to understand), poisson, gamma, or something else? Note that marks can be negative because of negative marking but you are allowed to add offsets as long as you declare them. You do not need to show the plot (it might be too slow for these many entries), but just the AIC values.

```
df <- read_csv(file = "./NTA_all_marks_2024.csv")
```

```
Warning: One or more parsing issues, call `problems()` on your data frame for details,
e.g.:
  dat <- vroom(...)
  problems(dat)
```

```
Rows: 2315409 Columns: 3
-- Column specification --------------------------------------------------------
Delimiter: ","
dbl (3): Srlno, Marks, center_number

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df <- df %>% filter(!is.na(Marks))
```

```r
marks <- df$Marks

# fitting gamma distribution was causing an error
# because of negative values, so i am adding an
# offset to all the marks

offset <- abs(min(marks)) + 1
shifted_marks <- marks + offset

fit_norm <- fitdistr(shifted_marks, densfun = "normal")
fit_pois <- fitdistr(shifted_marks, densfun = "poisson")
fit_gamma <- fitdistr(shifted_marks, densfun = "gamma")
```

```
Warning in densfun(x, parm[1], parm[2], ...): NaNs produced
```

```r
aic_values <- c(
      Normal = AIC(fit_norm),
      Poisson = AIC(fit_pois),
      Gamma = AIC(fit_gamma)
  )

aic_values
```

```
   Normal   Poisson     Gamma
 30249909 166936665  29715100
```

From the above AIC values, it is clear that the marks of students follows a **Gamma distribution** from among Normal, Gamma and Poisson.

Bonus question 08 [15 points]

Now that you have finished the assignment, it's time for a bonus problem.

While we do not have the exact scores since the data was extracted from the fig in the file `2023_2024_score_bins.csv` available here. Here is description of the fields:

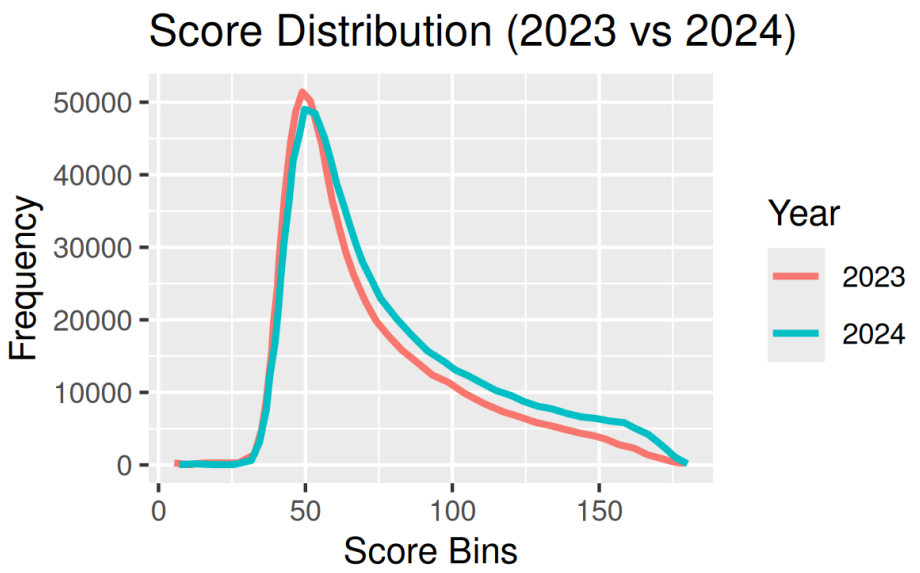
- `Score_bin`: the binned score bracket (these are comparable across 2023 and 2024)
- `frequency`: approximate number of students who obtained marks in a particular score bin

If you had to give a verdict on whether or not cheating was widespread, how would you use the above data to come to any recommendation for the NTA?

1. I am plotting the data for both the years to look whether there is any apparent difference in the distributions.

```
data <- read.csv("./NTA_2023_2024_score_bins.csv")

df_2023 <- data[data$year == 2023,]
df_2024 <- data[data$year == 2024,]

df_2023$Year <- "2023"
df_2024$Year <- "2024"
df_combined <- rbind(df_2023, df_2024)

ggplot(df_combined, aes(x = score_bin, y = frequency, fill = Year, color = Year)) +
  geom_line(size = 1) +
  labs(x = "Score Bins", y = "Frequency", title = "Score Distribution (2023 vs 2024)")
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```



- It seems that, above a certain threshold marks, the number of students scoring a certain amount of marks in 2024 are greater than the number of students scoring the same amount of marks. Also at higher marks, the frequency is unusually high.

2. Lets try to perform a **Chi-Square** test to check whether there is a distribution change from 2023 to 2024.

   Null Hypothesis: The marks in year 2023 and 2024 follows the same distribution.

   Alternate Hypothesis: There is a distribution change from the year 2023 to 2024.

```
score_bins_2023 = data$score_bin[data$year == 2023]
score_bins_2024 = data$score_bin[data$year == 2024]

freq_2023 = data$frequency[data$year == 2023]
freq_2024 = data$frequency[data$year == 2024]

# rebining the data
common_bins <- sort(unique(c(score_bins_2023, score_bins_2024)))

freq_2023_rebinned <- table(cut(score_bins_2023,
                                breaks = common_bins,
                                include.lowest = TRUE,
                                right = FALSE))

freq_2024_rebinned <- table(cut(score_bins_2024,
                                breaks = common_bins,
                                include.lowest = TRUE,
                                right = FALSE))

# this time I am directly using the inbuilt function for Chi-Square test
chi_sq_test <- chisq.test(freq_2023_rebinned, freq_2024_rebinned)
chi_sq_test
```

```
    Pearson's Chi-squared test with Yates' continuity correction

data:  freq_2023_rebinned and freq_2024_rebinned
X-squared = 73.374, df = 1, p-value < 2.2e-16
```

- The above test gives the $p-value$ of the test statistic to be *very small*. Hence we **reject** the null hypothesis.

We conclude that there are traces of cheating in the year 2024.