````
---
title: "DH 302 Spring 2025 Assignment 03"
subtitle: The assignment is based on [Lecture 11]
(https://docs.google.com/presentation/d/1tDVUNGIMnBlQ3PKah3aqp13jOWohOpriSaP7WKJoIs4/edit?
usp=sharing), [Lecture 13](https://docs.google.com/presentation/d/1PMPN-7GVn2kw-
jwqh3LdSSKbkdwLiXDtZk49cqRVZgE/edit?usp=sharing), and [Lecture 14]
(https://docs.google.com/presentation/d/1bO-oWyVl5auIyOPy9OU4YGqo4zingZzbEyd6e7Fx3Ng/edit?
usp=sharing). Due at 11:59PM (IST), Friday $14^{th}$ March, 2025 via Gradescope. NO LATE
SUBMISSIONS. Total points = 260
author:
  - "Deepak Silaych (22b0660)"
format:
  pdf:
    fig-format: png
    fig-dpi: 300
    fig-width: 8
    fig-height: 5
editor: source
---

# Instructions

Submit your solutions via [gradescope](https://www.gradescope.com/courses/946202) by 11:59
PM (IST) Friday, $14^{th}$ March 2025. In-person submissions will not be entertained.
Please upload a single PDF file. Late submissions are NOT ALLOWED. You can raise your
questions related to the assignment on [Piazza]
(https://piazza.com/class/m5k3nf7l4e91cg/post/m6alu8nsltc1pq) - please tag these as
`assignment_03`.

-   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.
-   LLM policy: You are allowed to use LLMs of course, but the final answer should be YOUR
OWN WORK. Do not copy-paste from LLMs. Not adhering to this policy will lead to
cancellation of your entire assignment.

```{r}
#| echo: false
#| message: false
# !! DO NOT EDIT/REMOVE !!

required.packages <- c("tidyverse", "ggpubr", "sf", "patchwork", "tmap", "patchwork",
"MASS")

InstallMissing <- function(packages) {
  for (pkg in packages) {
    if (!require(pkg, character.only = TRUE)) {
      install.packages(pkg, dependencies = TRUE)
      library(pkg, character.only = TRUE)
    }
  }
}

InstallMissing(required.packages)
````
````

~~~~
library(tidyverse)
library(patchwork)
library(ggpubr)
library(sf)
library(tmap)
library(MASS)
theme_set(theme_pubr())
```

# Problem 00 \[10 points\]

**Interpreting the coefficients:** Given the following linear models, interpret the
coefficients where $y$ is the dependent variable and $x$ is the independent variable (non-
categorical) and $z$ is another indepdent variable (categorical with binary outcomes).

\begin{itemize}
\item $y = \beta_0 + \beta_1 \log(x)$
\item $y = \beta_0 + \beta_1x + \beta_2x^2$
\item $y= \beta_0 + \beta_1 x + \beta_2 z$
\item $y = \beta_0 + \beta_1 x + \beta_2 z + \beta_3 xz $
\end{itemize}

Solutions:

### 1. Model: ( $y = \beta_0 + \beta_1 \log(x)$ )

```{r}
cat("\n**β_0**: Expected value of y when x = 1 (log(1) = 0)\n")
cat("**β_1**: Change in y per unit increase in log(x)\n")
```

### 2. Model: ( y = \beta\_0 + \beta\_1 x + \beta\_2 x\^2 )

```{r}
# Interpretation
cat("\n**β_0**: Expected value of y when x = 0\n")
cat("**β_1**: Linear effect of x on y\n")
cat("**β_2**: Quadratic effect of x on y (determines curvature)\n")

```

### 3. Model: ( y = \beta\_0 + \beta\_1 x + \beta\_2 z )

```{r}
# Interpretation
cat("\n**β_0**: Expected value of y when x = 0 and z = 0\n")
cat("**β_1**: Effect of x on y when z = 0\n")
cat("**β_2**: Effect of changing z from 0 to 1 when x = 0\n")
cat("**β_3**: Interaction term - how the effect of x changes depending on z\n")
```

### 4. Model: ( y = \beta\_0 + \beta\_1 x + \beta\_2 z + \beta\_3 xz )

```{r}
# Interpretation
cat("\n**\beta_0**: Expected value of y when x = 0 and z = 0\n")
cat("**\beta_1**: Effect of x on y when z = 0\n")
cat("**\beta_2**: Effect of changing z from 0 to 1 when x = 0\n")
cat("**\beta_3**: Interaction term - how the effect of x changes depending on z\n")
```

# Problem 01 \[110 points\]
~~~~

Why the GLM: On one hand the data has nothing to do with public health (at least not directly), on the other it is a good data to get some hands dirty while understanding GLM. The data is about house prices in a hypothetical city. The units do not matter so much, we are interested in the relationship between size of house and its price. The [data](https://drive.google.com/file/d/1Y9t41U6nbx8L0B_tMqcZkcio8lwHkjOA/view?usp=sharing) has seven columns:

-   case: individual survey point
-   price: price of the house (thousand dollars)
-   size: area of the house (square feet)
-   new: 0 or 1 indicating if it's a new house
-   taxes: total taxes paid (dollars)
-   bedrooms: number of bedrooms
-   baths: number of bathrooms

1a. Visualize the relationship between price and size of the house treating all kind of houses as of one type. What do you observe? \[10 points\]

```{r, warning=FALSE}
theme_set(theme_pubr())

house_prices <- read_table("data/Houses.dat")
house_prices

ggplot(house_prices, aes(x = size, y = price)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(title = "House Price vs. Size", x = "Size (sq ft)", y = "Price (k$)")
```

1b. Visualize the relationship between price and size of the house differentiating the new and old houses by a different color. What do you observe? \[10 points\]

```{r, warning=FALSE}
ggplot(house_prices, aes(x = size, y = price, color = factor(new))) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  scale_color_manual(values = c("red", "blue"), labels = c("Old", "New")) +
  labs(title = "House Price vs. Size (New vs. Old)", x = "Size (sq ft)", y = "Price (k$)",
color = "House Type")

```

1c. Fit a linear model to predict the price of the house based on just the size of the house. Interpret the coefficients. \[10 points\]

```{r}
# YOUR SOLUTION HERE
model.1c <- lm(price ~ size, data = house_prices)
summary(model.1c)
```

1d. Plot your model in 1c. \[10 points\]

```{r}
ggplot(house_prices, aes(x = size, y = price)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE, color = "blue") +
  labs(title = "Linear Model: Price ~ Size", x = "Size (sq ft)", y = "Price (k$)")
```

1e. Fit a linear model that models the price of the house based on the size of the house and whether the house is new or old but uses the same slope for both. Interpret the coefficients. \[10 points\]

```{r}
model.1e <- lm(price ~ size + new, data = house_prices)
summary(model.1e)
```

1f. Plot your model in 1e. \[10 points\]

```{r}
ggplot(house_prices, aes(x = size, y = price, color = factor(new))) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Linear Model: Price ~ Size + New/Old", x = "Size (sq ft)", y = "Price (k$)", color = "House Type")
```

1g.Fit a linear model that models the price of the house based on the size of the house and whether the house is new or old but allows for different slopes for new and old houses. Interpret the coefficients. \[10 points\]

```{r}
model.1g <- lm(price ~ size * new, data = house_prices)
summary(model.1g)
```

1h. Plot your model in 1g. \[10 points\]

```{r}
ggplot(house_prices, aes(x = size, y = price, color = factor(new))) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", formula = y ~ x, se = FALSE) +
  labs(title = "Linear Model: Price ~ Size * New", x = "Size (sq ft)", y = "Price (k$)", color = "House Type")

```

1i. Fit a GLM to predict the price of the house based on the size of the house using a gamma model with default link (log) with the same explanatory model as in 1g. Interpret the coefficients. Contrast your observations with the model in 1g for the interaction term. \[20 points\]

```{r}
model.1i <- glm(price ~ size * new, data = house_prices, family = Gamma(link = "log"))
summary(model.1i)
```

1j. Plot all the lines in 1h and 1i on the same plot. You can choose to also show the points \[10 points\]

```{r}
preds.1g <- predict(model.1g, house_prices)
preds.1i <- predict(model.1i, house_prices, type = "response")

house_prices$preds_1g <- preds.1g
house_prices$preds_1i <- preds.1i

ggplot(house_prices) +
  geom_point(aes(x = size, y = price), alpha = 0.5) +
  geom_line(aes(x = size, y = preds_1g, color = "Model 1g"), size = 1) +
  geom_line(aes(x = size, y = preds_1i, color = "Model 1i"), size = 1, linetype = "dashed") +
  labs(title = "Comparison of Models 1g and 1i", x = "Size (sq ft)", y = "Predicted Price", color = "Model")
```

Problem 03 \[100 points\]

This is India: For the semifinal assignment problem of this (half)course, we try to decipher India. The problem is much easier than the effort and the time it took to put it all together.

We revisit the problem of births in India and try to understand why it looks like what it does quantitativel. It is quite likely that you are attempting this problem after a week of the midsems, but this is a problem about India, so all things that define being an Indian in modern India and the promise we made ourselves at the beginning of the course hold: *be patient, be truthful, be kind, and be curious*.

The data was scraped from the HMIS portal of the Ministry of Health and Family Welfare, Government of India for the year 2018-2019. There is too much data, so we will focus on the number of births in four states: Bihar, Kerala, Chhatisgarh and Meghalaya.

All the explanatory and response variables are available in this file `conceavg_conceptions_and_covariates_by_month_2009_2020.csv` available [here](https://drive.google.com/file/d/1n9-X_bsgK-1E9nguNoAMS7haTEYM-2JM/view?usp=sharing). There are fix columns

-   "conception_month": month of conception (inferred by subtracting 9 months from the month of birth)
-   "state": State reporting birth (conception, assuming no migration)
-   "conceptions": Number of conceptions (birth)
-   "mean_relative_humidity": Mean relative humidity of the entire state
-   "mean_temperature": Mean relative temperature of the state
-   "mean_dewpoint": Mean dew point of the state

The `mean_dewpoint` was required for relative humidity calculation (which took a while to get it right), but is otherwise NOT required for this analysis.

3a. Visualize the overall relationship between the number of conceptions and the mean temperature taking all data at once yet showing the state of origin of each data point. What patters (if any) do you notice? \[10 points\]

```{r, warning=FALSE}
conceptions <- read_csv("data/avg_conceptions_and_covariates_by_month_2009_2020.csv")

ggplot(conceptions, aes(x = mean_temperature, y = conceptions, color = state)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Conceptions vs Temperature", x = "Mean Temperature", y = "Number of
Conceptions", color = "State")
```

3b. Extend 3a to now plot the number of conceptions in each state along with relative humidity and mean temperature in a side by side plot. What pattern(s), if any, do you notice? \[10 points\]

```{r, warning=FALSE}
#| fig.width=10, fig.height=5

# Plot for Conceptions vs Temperature
p1 <- ggplot(conceptions, aes(x = mean_temperature, y = conceptions, color = state)) +
  geom_point(alpha = 0.6) +
  labs(title = "Conceptions vs Temperature", x = "Mean Temperature (°C)", y = "Number of
Conceptions", color = "State")

# Plot for Conceptions vs Relative Humidity
p2 <- ggplot(conceptions, aes(x = mean_relative_humidity, y = conceptions, color = state))
+
  geom_point(alpha = 0.6) +
```

  labs(title = "Conceptions vs Relative Humidity", x = "Mean Relative Humidity (%)", y = "Number of Conceptions", color = "State")

Display both plots side by side
p1 | p2
```

3c. Map My India: Visualize the number of conceptions in each state along with relative humidity and mean temperature. \[10 points\]

We did not get time to cover this in class, but mapping data in R is a lot of fun (and very easy). For starter, here is a sample code to plot the states of India. The [geojson](https://en.wikipedia.org/wiki/GeoJSON) file is available [here](https://drive.google.com/file/d/1HxMx5ThwTUvJ58jq9MAOZZ8dSD-bVgJw/view?usp=sharing)

We can either use [geom_sf from sf](https://ggplot2.tidyverse.org/reference/ggsf.html) or [tmap](https://cran.r-project.org/web/packages/tmap/vignettes/tmap-nutshell.html) to plot the data.

```{r, warning=FALSE}
india.map <- read_sf("data/india_states.geojson")

# Join conception data with map data
india.map.updated <- india.map %>% left_join(conceptions, by = "state")

tm_shape(india.map.updated) +
  tm_polygons("mean_temperature", palette = "-RdYlBu", title = "Mean Temperature (°C)")
```

Here is how to use [tmap](https://cran.r-project.org/web/packages/tmap/vignettes/tmap-nutshell.html) to plot the data.

```{r}
tm_shape(india.map) +
  tm_polygons()
```

Our goal however is to visualize the conception and temperature, humidity across states for the six months January, March, May, July, September November. This is to just get a feeling of what the data looks like across the months. There are other ways to visualize this data, but this is not a bad start.

In the Code block that follows, write code to visualize the number of conceptions in only the state of Bihar (for simplicity) along with relative humidity and mean temperature extending the code that is already given. We use the fact that `india.map` can be treated as a data frame to do left joins.

```{r, warning=FALSE}
#|fig.width=10, fig.height=5
#|
subsampled.data <- conceptions %>% filter(conception_month %in% c("January", "March", "May", "July", "September", "November"))
india.map.updated <- india.map %>% left_join(subsampled.data)
india.map.updated <- india.map.updated[india.map.updated$state %in% c("Bihar"), ]

p1 <- tm_shape(india.map.updated) +
  tm_polygons(
    col = "mean_temperature",
    palette = "-brewer.spectral",
    style = "quantile"
  ) +
  tm_facets("conception_month",
    drop.NA.facets = T, ncol = 6
  ) +

  tm_layout(
    frame = FALSE,
    legend.outside = TRUE,
    legend.width = 6,
    legend.height = 6,
    legend.outside.position = "right"
  )
p1
```

Now plot the number of conceptions in Bihar, following p1:

```{r, warning=FALSE}
p2 <- 1 # YOUR SOLUTION HERE
india.map <- read_sf("data/india_states.geojson")
india.map.updated <- india.map %>% left_join(conceptions, by = "state")

ggplot(india.map.updated) +
  geom_sf(aes(fill = mean_temperature)) +
  theme_void() +
  labs(title = "Mean Temperature Across States")
p2
```

Finally add another plot for visualizing relative humidity in Bihar and put everything together in one plot (p1, p2, p3). You can use the [tmap_arrange](https://rdrr.io/cran/tmap/man/tmap_arrange.html) to put the plots together. Don't worry about overflowing legends. Your output should be 3 rows of maps with first row indiciating the temperature, the second row indicating the conceptions and third row indicating the relative humidity.

```{r}
#|fig.width=10, fig.height=10
p3 <- 1
# Fit Generalized Linear Models (GLM)
glm_gaussian <- glm(conceptions ~ mean_temperature + mean_relative_humidity, data =
conceptions, family = gaussian())
glm_poisson <- glm(conceptions ~ mean_temperature + mean_relative_humidity, data =
conceptions, family = poisson())
glm_gamma <- glm(conceptions ~ mean_temperature + mean_relative_humidity, data =
conceptions, family = Gamma())

# Compare models using AIC
AIC(glm_gaussian, glm_poisson, glm_gamma)

```

3d. What the GLM? Assuming each state has its own intercept, but slopes are fixed across states fit a GLM to predict the number of conceptions based on the mean temperature and relative humidity. You can use one of gaussian, poisson, negative binomial, or gamma family. Show which model is the best \[20 points\]

For negative binomial you will need to use `glm.nb` instead of `glm`, everything else remains the same.

```{r}
# Fit Generalized Linear Models (GLM)
glm_gaussian <- glm(conceptions ~ mean_temperature + mean_relative_humidity, data =
conceptions, family = gaussian())
glm_poisson <- glm(conceptions ~ mean_temperature + mean_relative_humidity, data =
conceptions, family = poisson())
glm_gamma <- glm(conceptions ~ mean_temperature + mean_relative_humidity, data =
conceptions, family = Gamma())

fit.glm_aussian <- 1

fit.glm_poisson <- 1
fit.glm_gamma <- 1
fit.nb <- 1
# Compare models using AIC
AIC(glm_gaussian, glm_poisson, glm_gamma)
```

3e. Does humidity matter? Using the best model you identified in 3d, contrast your model from 3e with a reduced model that does not include humidity. What do you observe, does humidity matter? \[10 points\]

```{r}
best_model <- glm_poisson

fit.bestmodel.nohumidity <- 1# YOUR SOLUTION HERE

fit.bestmodel.withhumidity <- 1 # YOUR SOLUTION HERE

reduced_model <- glm(conceptions ~ mean_temperature, data = conceptions, family =
poisson())


AIC(best_model, reduced_model)

```

3f. Maharahstra Mazha. We did not use Maharashtra data, but we can bring in it to see if it changes our model. Fit a GLM to predict the number of conceptions based on the mean temperature and relative humidity including Maharashtra. What do you observe? \[10 points\]

If you need some context for the title of this question, see [here](https://www.youtube.com/watch?v=JgdfVaOYJMM) -- Audio/Video content. Besides the title, the question bears no resemblance to the content.

The data for this part is available [here](https://drive.google.com/file/d/1qzNbIPlOfNjYpIXF64KkpV4kqmTcfqxh/view?usp=sharing).
Let's load it:

```{r}
conceptions.mah <-
read_csv("data/avg_conceptions_and_covariates_by_month_2009_2020_Maharashtra_only.csv")
```

Using the old data of Bihar, Kerala, Meghalaya and Chhatisgarh and the new data of Maharashtra, plot the relationship between number of conceptions and mean temperature to identify the best state that can be used to predict the number of conceptions based on the mean temperature and relative humidity for Maharashtra:

```{r}
conceptions.all <- 1 # rbind()

maharashtra_data <-
read_csv("data/avg_conceptions_and_covariates_by_month_2009_2020_Maharashtra_only.csv")

# Rename column for clarity
maharashtra_data <- maharashtra_data %>%
  rename(observed_conceptions = conceptions)

# Load the best model state (previously identified)
state.for.mimicking.model <- "Maharashtra"

```
# Fit GLM model using Maharashtra data
glm_best <- glm(observed_conceptions ~ mean_temperature + mean_relative_humidity,
                data = maharashtra_data, family = poisson())

Generate predictions using best model
maharashtra_data$predicted_conceptions_best <- predict(glm_best, newdata =
maharashtra_data, type = "response")

```

## 3g. Fit a GLM to predict the number of conceptions based on the mean temperature and relative humidity of Maharashtra using the model of the state you previously estimated in 3f and calculate the mean squared error. Try it with one other state, visualize the predictions on a map. What do you conclude - is your best state a good model for Maharashtra? \[30 points\]

```{r}
# !! DO NOT EDIT/REMOVE !!
maharashtra.data <- conceptions.mah
maharashtra.data$observed_conceptions <- maharashtra.data$conceptions
maharashtra.data$conceptions <- NULL

# change the state of this data to "best state"
bihar_data <- conceptions %>% filter(state == "Bihar")

# Fit GLM model using Bihar's data
glm_other <- glm(conceptions ~ mean_temperature + mean_relative_humidity,
                 data = bihar_data, family = poisson())

# Generate predictions for Maharashtra using Bihar's model
maharashtra_data$predicted_conceptions_other <- predict(glm_other, newdata =
maharashtra_data, type = "response")

state.for.mimicking.model <- ""
maharashtra.data$state <- state.for.mimicking.model
# predictions.maharashtra.best <- predict(fit.bestmodel.withhumidity, newdata =
maharashtra.data, type = "response")

# maharashtra.data$predicted_conceptions_best <- predictions.maharashtra.best

# someotherstate <- ""
#maharashtra.data$state <- someotherstate
#predictions.maharashtra.someotherstate <- predict(fit.bestmodel.withhumidity, newdata =
maharashtra.data, type = "response")

#maharashtra.data$predicted_conceptions_otherstate <-
predictions.maharashtra.someotherstate
```

Calculate the mean squared error between the observed conceptions in Maharashtra and those predicted using your "best state":

```{r}
# Compute MSE for best model
mse.best <- mean((maharashtra_data$observed_conceptions -
maharashtra_data$predicted_conceptions_best) ^ 2)
mse.best

# Compute MSE for Bihar model
mse.otherstate <- mean((maharashtra_data$observed_conceptions -
maharashtra_data$predicted_conceptions_other) ^ 2)
mse.otherstate
```

Calculate the mean squared error between the observed conceptions in Maharashtra and those predicted using some other state:

```{r}
mse.otherstate <- 1
mse.otherstate
```

Now visualize this on a map following ideas form 3c:

```{r}
#|fig.width=10, fig.height=10
maharashtra.data.final <- maharashtra.data
maharashtra.data.final$state <- "Maharashtra"
india.map.updated <- left_join(india.map, maharashtra.data.final)
india.map.updated <- india.map.updated[india.map.updated$state %in% c("Maharashtra"), ]
# YOUR CODE HERE

p1 <- 1
p2 <- 1
p3 <- 1
# Show plots p1, p2, p3 stacked together
ggplot(maharashtra_data, aes(x = mean_temperature)) +
  geom_point(aes(y = observed_conceptions, color = "Observed"), alpha = 0.6) +
  geom_point(aes(y = predicted_conceptions_best, color = "Predicted (Best Model)"), alpha
= 0.6) +
  geom_point(aes(y = predicted_conceptions_other, color = "Predicted (Bihar Model)"),
alpha = 0.6) +
  labs(title = "Observed vs. Predicted Conceptions in Maharashtra", x = "Mean Temperature
(°C)", y = "Number of Conceptions") +
  scale_color_manual(values = c("Observed" = "blue", "Predicted (Best Model)" = "green",
"Predicted (Bihar Model)" = "red")) +
  theme_minimal()

```

Problem 4 In pursuit of happiness: You are given a dataset of happines index as a function of wealth. The units do not matter. The data is spread over different age groups (1-youngest, 4-oldest) \[40 points\]

The dataset is available [here]
(https://drive.google.com/file/d/12zIw8S5g1QY9RpRLtuoc3L_K8k8KLpq_/view?usp=sharing)

4a. What is the relationship between happiness and wealth? Fit a linear model (gaussian) using all data points as one group, interpret the coefficients and generate a plot of your linear model \[10 points\]

```{r, warning=FALSE}
happiness <- read_csv("data/pursuit_of_happiness.csv")
```

```{r}
# YOUR CODE HERE FOR MODEL FIT
# Model Fit
model.4a <- lm(happiness ~ wealth, data = happiness)
summary(model.4a)
```

```{r}
# Plotting the Model
ggplot(happiness, aes(x = wealth, y = happiness)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "lm", color = "blue", se = TRUE) +
  labs(title = "Happiness vs Wealth", x = "Wealth", y = "Happiness")
```

````
```

## 4b. Adjust your linear model to allow estimating individual slopes and intercept for
each age group and plot the estimated lines. What do you conclude with your updated model?
\[30 points\]

Note: Your plot should be intereprtable for full points.

```{r, warning=FALSE}
happiness$age_group <- as.factor(happiness$age_group)
model.4b <- lm(happiness ~ wealth * age_group, data = happiness)
summary(model.4b)
```

```{r}
# Plotting the Model
ggplot(happiness, aes(x = wealth, y = happiness, color = age_group)) +
  geom_point(alpha = 0.6) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Happiness vs Wealth by Age Group", x = "Wealth", y = "Happiness")

```

###### FINISH

```{r}

```
````
````