exploring_estimateates

Preliminaries

Loading libraries

```
suppressPackageStartupMessages({
    library(tidyverse)
    library(readxl)
    library(plot3D)
})
```

```
Warning: package 'plot3D' was built under R version 4.4.2
```

Loading data, adjusted visual settings, renaming columns

```
fermi_data <- read_excel("Fermi-Data.xlsx")
fermi_data <- data.frame(fermi_data)

options(scipen = 999)
fermi_data <- fermi_data %>%
    rename(
        Answers = Correct.answers,
        Timothy_estimate = Timothy.Answer,
        Timothy_average = Timothy.s.guess,
        Markus_estimate = Markus.Answer,
        Markus_average = Markus.s.guess,
        Marvin_estimate = Marvin.Answer,
        Marvin_average = Marvin.s.guess,
        Adrian_estimate = Adrian.Answer,
        Adrian_average = Adrian.s.guess,
    )
```

chat gpt estimates that one million chickens were shredded in germany last year. Even with the ban, this is just a guess of course. run this cell if you want to use 1 million instead of 0 as the value.

```
# fermi_data[2, 2] <- 1000000
```

Function to calculate OOM error

```
calc_error <- function(actual, forecast) {
    abs(log10(max(0.01, actual)) - log10(max(0.01, forecast)))
}
```

Setting up dataframes and calculating errors

Add columns for the two mean values

```
fermi_data$Mean_of_estimates <- rowMeans(fermi_data[, c(3, 5, 7, 9)], na.rm = TRUE)
fermi_data$Mean_of_averages <- rowMeans(fermi_data[, c(4, 6, 8, 10)], na.rm = TRUE)
```

create results dataframe, where the errors will be stored, and norm_results where the normalized errors will be stored

```
people <- c("Timothy", "Markus", "Marvin", "Adrian", "Mean")
types <- c("_est", "_ave")

results <- data.frame(matrix(ncol = 10, nrow = 10))
names(results) <- paste0(rep(people, each = 2), rep(types, times = 5), "_er")

norm_results <- data.frame(matrix(ncol = 10, nrow = 10))
names(norm_results) <- paste0(rep(people, each = 2), rep(types, times = 5), "_nor_er")
```

calculate the errors and store them in results

```
for (i in 1:10) {
    for (j in 1:10) {
        results[i, j] <- calc_error(fermi_data$Answers[i], fermi_data[i, j + 2])
    }
}
```

calculate the normalized errors and store them in norm_results

```r
for (i in 1:10) {
    sum_row_estimates <- sum(results[i, c(1, 3, 5, 7)], na.rm = TRUE)
    sum_row_averages <- sum(results[i, c(2, 4, 6, 8)], na.rm = TRUE)
    for (j in c(1, 3, 5, 7, 9)) {
        norm_results[i, j] <- results[i, j] / sum_row_estimates
    }
    for (j in c(2, 4, 6, 8, 10)) {
        norm_results[i, j] <- results[i, j] / sum_row_averages
    }
}
```

Add collumn to the left of norm_results and results with description of each row

```r
new_col <- rep(NA, nrow(results))

results <- cbind(new_col, results)
norm_results <- cbind(new_col, norm_results)
names(results)[1] <- "Row_description"
names(norm_results)[1] <- "Row_description"
results$Row_description <- fermi_data$Question
norm_results$Row_description <- fermi_data$Question
```

add row to results and norm_results with column sums

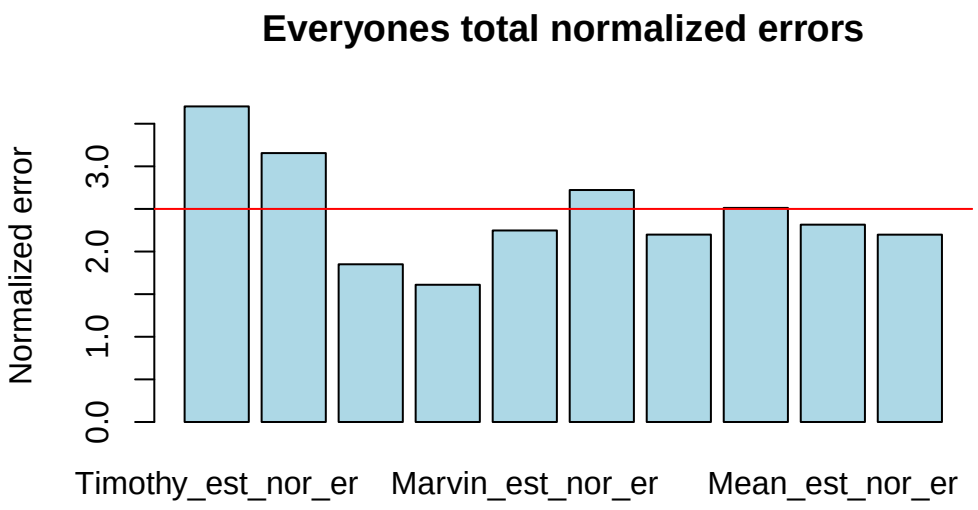
```r
for (j in 2:11) {
    results[11, j] <- sum(results[1:10, j])
    norm_results[11, j] <- sum(norm_results[1:10, j])
}

norm_results[11, 1] <- "Total_norm_error"
results[11, 1] <- "Total_error"
```

Create Plots

plot normalized errors

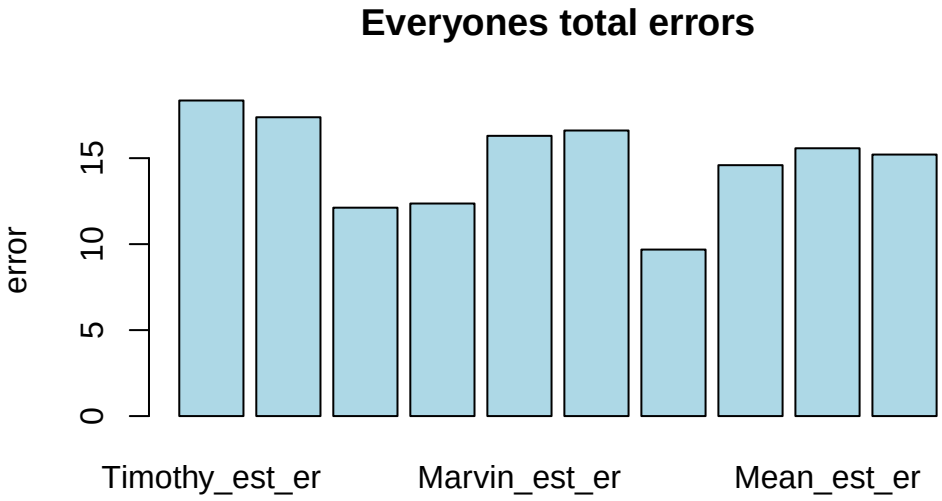
```r
barplot(
    as.numeric(norm_results[11, 2:11]),
    names.arg = names(norm_results)[2:11],
    col = "lightblue",
```

```
    main = "Everyones total normalized errors",
    ylab = "Normalized error",
)
abline(h = 2.5, col = "red")
```



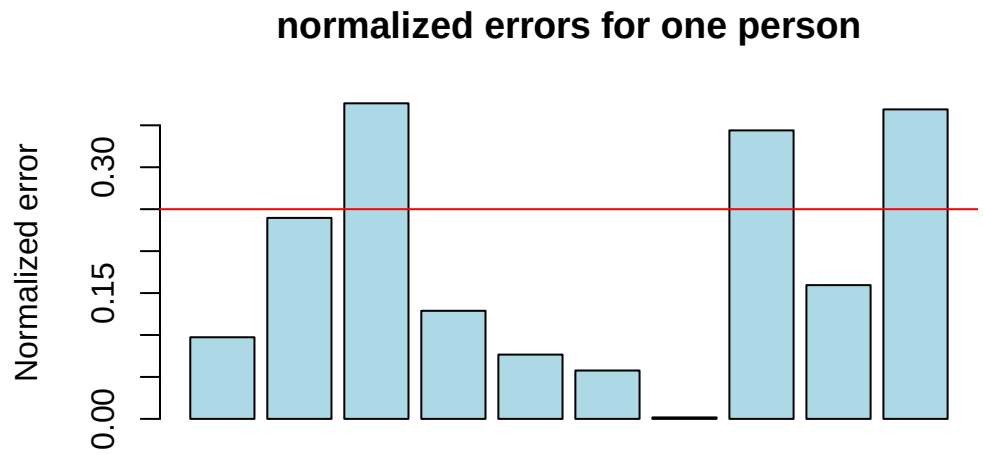
plot non normalized errors

```
barplot(
    as.numeric(results[11, 2:11]),
    names.arg = names(results)[2:11],
    col = "lightblue",
    main = "Everyones total errors",
    ylab = "error",
)
```



plot normalized errors for one person

```
# 2 = Timothy
# 4 = Markus
# 6 = Marvin
# 8 = Adrian
person <- 4

barplot(
    as.numeric(norm_results[1:10, person]),
    names.arg = norm_results$Row_description[1:10],
    col = "lightblue",
    main = "normalized errors for one person",
    ylab = "Normalized error",
)
# The line represents the average normalized error
abline(h = 0.25, col = "red")
```

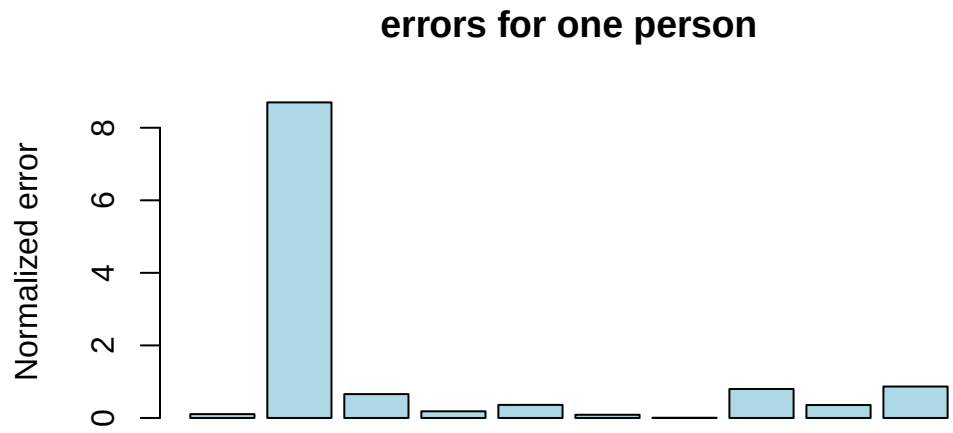


:hickens in factory farms in DE now 8. Cost of an EAGxBerlin 2023

plot non normalized errors for one person

```
# 2 = Timothy
# 4 = Markus
# 6 = Marvin
# 8 = Adrian
person <- 4

barplot(
    as.numeric(results[1:10, person]),
```

```
    names.arg = results$Row_description[1:10],
    col = "lightblue",
    main = "errors for one person",
    ylab = "Normalized error",
)
```



:hickens in factory farms in DE now 8. Cost of an EAGxBerlin 2023

create 3d barplot of normalized errors

```
z <- as.matrix(norm_results[1:10, c(2, 4, 6, 8)])

hist3D(
    x = rep(1:10),
    y = rep(1:4),
    z = z,
    col = "lightblue",
    border = "black",
    xlab = "Questions",
    ylab = "People",
    zlab = "Normalized Error",
    main = "3D Barplot of Normalized Errors"
)
```

3D Barplot of Normalized Errors