Statistical Models Report

Part 1 - Time Travel Data

Part (a)

> Create a subject-level data set (492 rows) with the following variables: the above seven predictors, the number of observations for the given subject (up to 18), and his/her number of future thoughts (i.e., number of observations for which future=1). (Hint: The aggregate() function or dplyr::group_by() may be useful.)

```
source("renv/activate.R")
```

```
## - Project '~/statsmod2' loaded. [renv 1.1.5]
```

```
library(dplyr)
library(tidyverse)
library(haven)
library(zeallot)
```

a1. Loading the data

```
path <- "datasources-time-travel/Study\ 1/ETT_ESM_Study1.sav"
print(sprintf("Reading %s", path))
```

```
## [1] "Reading datasources-time-travel/Study 1/ETT_ESM_Study1.sav"
```

```
tib <- haven::read_sav(path)
# colnames(tib) # commented out - Too long output for report
```

a2. Initial cleaning

```
tib_clean <- tib |>
  select(where(~ !all(is.na(.))))

tib_clean |> group_by(Subject)
```

```
## # A tibble: 6,686 x 273
## # Groups:   Subject [492]
##      Subject random randomemo     PID RID
##        <dbl>  <dbl> <chr>       <dbl> <chr>
```

```
##  1   4.16e10      7 2         4.16e10 MLRP~
##  2   4.16e10      5 2         4.16e10 MLRP~
##  3   4.16e10      1 2         4.16e10 MLRP~
##  4   4.16e10      2 2         4.16e10 MLRP~
##  5   4.16e10      7 1         4.16e10 MLRP~
##  6   4.16e10      9 2         4.16e10 MLRP~
##  7   4.16e10      2 1         4.16e10 MLRP~
##  8   4.16e10      4 1         4.16e10 MLRP~
##  9   4.16e10      3 2         4.16e10 MLRP~
## 10   4.16e10      9 2         4.16e10 MLRP~
## # i 6,676 more rows
## # i 268 more variables: DAY <dbl>,
## #   SIG <dbl>, TIME <time>, SMS <chr>,
## #   TimeZone <chr>, RT <time>,
## #   country <chr>, mind_1 <dbl>,
## #   mind_2 <dbl>, mind_3 <dbl>,
## #   mind_4 <dbl>, alone <dbl+lbl>, ...
```

```
length(tib_clean)
```

```
## [1] 273
```

a3. Subject Level

```
subj_level <- tib_clean |>
  group_by(Subject) |>
  summarise(
    age = first(age, na_rm = TRUE),
    sex = first(sex, na_rm = TRUE),
    O = first(O, na_rm = TRUE),
    C = first(C, na_rm = TRUE),
    E = first(E, na_rm = TRUE),
    A = first(A, na_rm = TRUE),
    N = first(N, na_rm = TRUE),
    n_obs = n(),
    n_futures = sum(time_3, na.rm = TRUE)
  ) |>
  drop_na(age, sex, O, C, E, A, N)
```

Use descriptive statistics and graphs to inspect these variables, and describe any missing data.

```
table(subj_level$age, useNA = "ifany")
```

```
##
## 18 19 20 21 22 23 24 25 26 27 28 29 30 31
##  5 32 45 32 33 28 28 34 19 25 17 14 13 18
## 32 33 34 35 36 37 38 39 40 41 42 43 44 45
##  9 16  5  9  9  5  8  5  5  6  5  5  2  4
## 46 47 48 49 50 51 52 53 54 55 56 57 58 59
##  3  8  2  2  3  3  4  3  2  2  2  1  1  1
## 60 65 67
##  2  1  1
```

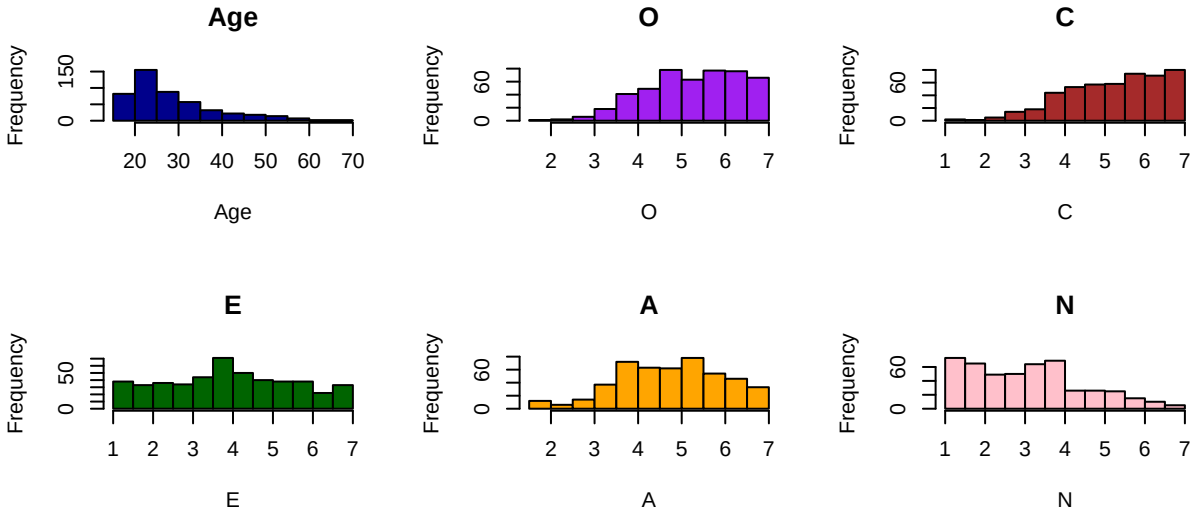
```r
table(subj_level$sex, useNA = "ifany")
```

```
##
##   0   1
## 173 304
```

```r
sapply(
  subj_level[, c("age", "O", "C", "E", "A", "N", "n_obs", "n_futures")],
  function(x) {
    c(
      Mean = mean(x, na.rm = TRUE),
      SD = sd(x, na.rm = TRUE),
      Min = min(x, na.rm = TRUE),
      Max = max(x, na.rm = TRUE)
    )
  }
)
```

```
##            age        O        C        E
## Mean 28.811321 5.506289 5.463312 4.155136
## SD    9.618022 1.072950 1.202333 1.618576
## Min  18.000000 1.500000 1.000000 1.000000
## Max  67.000000 7.000000 7.000000 7.000000
##             A        N    n_obs n_futures
## Mean 4.976939 3.289308 13.79036  4.010482
## SD   1.202017 1.452378  4.67045  2.809403
## Min  1.500000 1.000000  1.00000  0.000000
## Max  7.000000 7.000000 18.00000 15.000000
```

```r
par(mfrow = c(3, 3))
hist(subj_level$age, main = "Age", xlab = "Age", col = "darkblue")
hist(subj_level$O, main = "O", xlab = "O", col = "purple")
hist(subj_level$C, main = "C", xlab = "C", col = "brown")
hist(subj_level$E, main = "E", xlab = "E", col = "darkgreen")
hist(subj_level$A, main = "A", xlab = "A", col = "orange")
hist(subj_level$N, main = "N", xlab = "N", col = "pink")
```



Part (b)

> Using your favorite model selection method, find an appropriate Poisson regression with number of future thoughts as response and a subset of the above seven predictors. Include an offset term to account for the different numbers of observations per person.

Chosen Statistical model

A Poisson regression model is fitted to the count of future events (`n_futures`).

- The expected count is modeled as a linear function of `sex` `O`, `C`, `E`, `A` and, `N`.
- The **log** is the link function for this GLM.
- The **offset** `log(n_obs)` accounts for differing numbers of observations per subject.

Model Equation Let y_i denote the number of future thoughts for subject i, and n_i the number of observations recorded for that subject.

The Poisson regression with offset is:

$$y_i \sim \text{Poisson}(\mu_i), \quad \log(\mu_i) = \log(n_i) + \beta_0 + \beta_1 \text{sex}_i + \beta_2 O_i + \beta_3 C_i + \beta_4 E_i + \beta_5 A_i + \beta_6 N_i$$

$$\begin{aligned} y_i &= n\_futures_i && \text{(count of future thoughts)} \\ n_i &= n\_obs_i && \text{(number of observations)} \\ \beta_0, \ldots, \beta_6 &= \text{coefficients for } \text{age}_i, \text{sex}_i, O_i, C_i, E_i, A_i, N_i \end{aligned}$$

```
model <- glm(n_futures ~ age + sex + O + C + E + A + N,
  data = subj_level,
  family = poisson(link = "log"),
  offset = log(n_obs)
)
summary(model)
```

```
##
## Call:
## glm(formula = n_futures ~ age + sex + O + C + E + A + N, family = poisson(link = "log"),
##     data = subj_level, offset = log(n_obs))
##
## Coefficients:
##              Estimate Std. Error z value
## (Intercept) -1.615464   0.208748  -7.739
## age         -0.002526   0.002411  -1.048
## sex         -0.074190   0.049714  -1.492
## O            0.055107   0.022365   2.464
## C           -0.018758   0.020386  -0.920
## E            0.023194   0.014547   1.594
## A            0.028614   0.021555   1.327
## N            0.019073   0.017770   1.073
##             Pr(>|z|)
## (Intercept)    1e-14 ***
## age           0.2947
## sex           0.1356
```

```
## O             0.0137 *
## C             0.3575
## E             0.1108
## A             0.1844
## N             0.2831
## ---
## Signif. codes:
##   0 '***' 0.001 '**' 0.01 '*' 0.05 '.'
##   0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 820.00  on 476  degrees of freedom
## Residual deviance: 805.14  on 469  degrees of freedom
## AIC: 2198.9
##
## Number of Fisher Scoring iterations: 5
```

Part (c)

> Test for overdispersion in the chosen model, and refit the same model with the "quasipoisson" family. Does this change the model results? Why or why not?

c1. Cheking for overdispersion

For each subject, this model makes the following assumption:

$$\mathbb{E}(count) = \mathbb{V}(count)$$

So we compute the dispersion:

$$\hat{\phi} = \frac{\text{deviance}}{\text{tib}}$$

We check for overdispersion if $\hat{\phi} > 1.5$

```
disp <- deviance(model) / df.residual(model)
if (disp > 1.5) {
  print(sprintf("Overdispersion detected: (%s)", disp))
} else {
  print(sprintf("No Overdispersion detected: (%s)", disp))
}
```

```
## [1] "Overdispersion detected: (1.71671634829954)"
```

c2. Quasi-Poisson

The quasi-Poiison has a more relaxed variance assumption

```
quasi_model <- glm(n_futures ~ sex + O + C + E + A + N,
  data = subj_level,
  family = quasipoisson(link = "log"),
  offset = log(n_obs),
```

```
)

quasi_disp <- summary(quasi_model)$dispersion
if (quasi_disp > 1.5) {
  print(sprintf("Quasi: Overdispersion detected: (%s)", quasi_disp))
} else {
  print(sprintf("Quasi: No Overdispersion detected: (%s)", quasi_disp))
}
```

```
## [1] "Quasi: Overdispersion detected: (1.56061572332126)"
```

Part 2 - Lucia Deberk

Generation of data. Set the total number of shifts to `1029`, where a third of them correspond to the morning shifts, and all the rest to evening/night shifts. Denote the binary indicator of the morning shift by morning. Given morning $= 1$ generate a binary indicator Lucia of whether Lucia was on duty from `Bern(0.4)`, otherwise from `Bern(0.1)`. If you aggregate these two indicators into a 2-by-2 table, you will get something similar to:

```
shifts <- 1029

morning <- sample(c(FALSE, TRUE),
    size = shifts,
    replace = TRUE,
    prob = c(2 / 3, 1 / 3)
)

lucia <- ifelse(
    morning == TRUE,
    rbinom(n = shifts, size = 1, prob = 0.4),
    rbinom(n = shifts, size = 1, prob = 0.1)
) == 1

tb <- tibble(
    morning = morning,
    Lucia = lucia
)

tb |> count(Lucia, morning)
```

```
## # A tibble: 4 x 3
##   Lucia morning     n
##   <lgl> <lgl>   <int>
## 1 FALSE FALSE     614
## 2 FALSE TRUE      195
## 3 TRUE  FALSE      77
## 4 TRUE  TRUE      143
```

```
table(Lucia = tb$Lucia, Morning = tb$morning) |> addmargins()
```

```
##        Morning
```

```
## Lucia   FALSE TRUE  Sum
##   FALSE   614  195  809
##   TRUE     77  143  220
##   Sum     691  338 1029
```

Now, generate the number of incidences (deaths) occurred for each of the combi- nations of Lucia and morning, from the following Poisson regression model with a log link and an offset for the number of shifts

$$log(\frac{\mu(x_1, x_2)}{t(x_1, x_2)}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

Where $X_1 = morning$, X_2=$Lucia$ $t(x_1, x_2)$ equals the number of shifts corresponding to (x_1, x_2) and, with $\beta_0 = -4, \beta_1 = 1.7, \beta_2 = 0$.

```
c(beta_0, beta_1, beta_2) %<-% c(-4.0, 1.7, 0.0)

cell_counts <- tb |> count(morning, Lucia, name = "shift_count")
```

We want to find $\mu(x_1, x_2)$ = Expected number of deaths

$$log(\frac{\mu(x_1, x_2)}{t(x_1, x_2)}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$

$$\implies \frac{\mu(x_1, x_2)}{t(x_1, x_2)} = e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2}$$

$$\implies \mu(x_1, x_2) = e^{\beta_0 + \beta_1 x_1 + \beta_2 x_2} \cdot t(x_1, x_2)$$

```
tb2 <- cell_counts |>
    mutate(mu = shift_count * exp(beta_0 + beta_1 * morning + beta_2 * Lucia))

# Now apply the poission to the $\mu$
tb2 <- tb2 |> mutate(deaths = rpois(n = n(), lambda = mu))
tb2
```

```
## # A tibble: 4 x 5
##   morning Lucia shift_count    mu deaths
##   <lgl>   <lgl>       <int> <dbl>  <int>
## 1 FALSE   FALSE         614 11.2      10
## 2 FALSE   TRUE           77  1.41      1
## 3 TRUE    FALSE         195 19.6      22
## 4 TRUE    TRUE          143 14.3      10
```

```
lucia_marginal <- tb2 |>
    group_by(Lucia) |>
    summarise(
        deaths = sum(deaths),
        shifts = sum(shift_count),
    )
```

(b) **Careful analysis**. Analyze the obtained count data using the Poisson log-linear regression with an intercept and 2 factors $X_1 = morning$ and $X_2 = Lucia$, and include an offset term to account for the different number of shifts between Lucia de Berk and the rest of the nurses. Keep the pvalue corresponding to the effect of X_2.

```
tb2 <- tb2 |>
    mutate(
        morning = as.integer(morning),
        Lucia   = as.integer(Lucia)
    )

y <- tb2$deaths

fit_b <- glm(
    deaths ~ morning + Lucia,
    family = poisson(link = "log"),
    data = tb2,
    offset = log(shift_count)
)


lucia_pvalue <- summary(fit_b)$coefficients["Lucia", "Pr(>|z|)"]
morning_pvalue <- summary(fit_b)$coefficients["morning", "Pr(>|z|)"]
print(sprintf("P values: Morning: %s Lucia: %s", morning_pvalue, lucia_pvalue))
```

```
## [1] "P values: Morning: 1.19155169826394e-07 Lucia: 0.210533519897748"
```

(c) **Less careful analysis.** Aggregate the data over $X_1 = morning$ in order to obtain similar data as the real data set in Lucia de Berk's trial. Analyze the obtained count data using again the Poisson log-linear regression with an intercept, a factor of $X_2 = Lucia$, and also include an offset term to account for the different number of shifts between Lucia de Berk and the rest of the nurses. Keep the pvalue corresponding to the effect of X_2.

```
agg <- tb2 |>
    group_by(Lucia) |>
    summarise(
        deaths = sum(deaths),
        shift_count = sum(shift_count)
    )


fit_c <- glm(
    deaths ~ Lucia,
    family = poisson(link = "log"),
    data = agg,
    offset = log(shift_count)
)

#summary(fit_c)$coefficients["Lucia", "Pr(>|z|)"]
```

(d) **Original analysis.** Under the null hypothesis that there is no difference between Lucia de Berk and other nurses, and under several additional assumptions, with one of them being

that there are no other imoportant factors to be taken into account (aka confounders) (and also under blinded/fair data collection process), the conditional probability (given the total number of incidents and the total number of shifts) of observing the number of incidences

(e) which was observed, or more, 3(7) (.”2)/ (“Ee”) could be calculated using the hypergeometric distribution (see page 235 in Meester et al. (2006)) summed up for all x values which are $\geq x$ and $\leq k$:

$$\frac{\binom{m}{x}\binom{n}{k-x}}{\binom{m+n}{k}}$$

where k is the number of incidents occurred, n is the number of shifts of other nurses, m is the number of shifts of Lucia, x is the number of incidents ocurred during the shifts of Lucia. Keep the obtained pvalue.

Summarize the results of 200 replications by calculating the proportion of times you rejected the null hypothesis that there is no difference between Lucia de Berk and other nurses (i.e., no nurse effect), for each type of the 3 analyses separately. Use the 0.05 significance level for rejection of the null hypothesis.

Breakdown:

n = number of shifts worked by Lucia m = number of shifts worked by Other Nurses $m + n$ = Total shifts k = number of incidents x = number of incidents in Lucia's shift

H_0 = Lucia has the same rate as Other Nurses

$$P(X = x) = \frac{\binom{m}{x}\binom{n}{k-x}}{\binom{m+n}{k}}$$

Explanation:

1. $\binom{m}{x}$: choose which x of Lucia's m shifts had incidents
2. $\binom{n}{k-x}$: choose which $k - x$ of the other nurseses n shifts had incidents
3. divide by $\binom{m+n}{k}$: all ways to place k incidents among all shifts

```
n <- tb2 |>
    filter(Lucia == FALSE) |>
    summarise(shift_count = sum(shift_count)) |>
    pull()

m <- tb2 |>
    filter(Lucia == TRUE) |>
    summarise(shift_count = sum(shift_count)) |>
    pull()

x <- tb2 |>
    filter(Lucia == TRUE) |>
    summarise(deaths = sum(deaths)) |>
    pull()
```

```r
k <- tb2 |>
    summarise(deaths = sum(deaths)) |>
    pull()

cat(n, m, x, k, sep = "\n")
```

```
## 809
## 220
## 11
## 43
```

```r
p_value <- phyper(x - 1, m, n, k, lower.tail = FALSE)

p_value
```

```
## [1] 0.3011673
```

Simulation section

Model Equations Let y_{ij} be the number of deaths where:

- $i = 1$:= morning shift
- $j = 1$:= Lucia de Berk on duty
  (and 0 is inverse)
  and t_{ij} denote the number of shifts.

The model:

$$y_{ij} \sim \text{Poisson}(\mu_{ij}), \quad \log(\mu_{ij}) = \log(t_{ij}) + \beta_0 + \beta_1 \cdot i + \beta_2 \cdot j$$

```r
NUM_SIMULATIONS <- 200
P_THRESH <- 0.05

gen_data <- function(shifts = 1029,
                     beta_0 = -4.0,
                     beta_1 = 1.7,
                     beta_2 = 0.0,
                     mornings_part = 1 / 3) {
    morning <- sample(c(FALSE, TRUE),
        size = shifts,
        replace = TRUE,
        prob = c(2 / 3, 1 / 3)
    )
    lucia <- ifelse(
        morning == TRUE,
        rbinom(n = shifts, size = 1, prob = 0.4),
        rbinom(n = shifts, size = 1, prob = 0.1)
    ) == 1

    data <- tibble(
```

```
        morning = as.integer(morning),
        Lucia = as.integer(lucia)
    )

    data <- data |> count(morning, Lucia, name = "shift_count")

    data <- data |> mutate(mu = shift_count * exp(beta_0 + beta_1 * morning + beta_2 * Lucia))

    # Now apply the poission to the $\mu$
    data <- data |> mutate(deaths = rpois(n = n(), lambda = mu))
    data
}

sim_b <- function(data) {
    fit_b <- glm(
        deaths ~ morning + Lucia,
        family = poisson(link = "log"),
        data = data,
        offset = log(shift_count)
    )

    summary(fit_b)$coefficients["Lucia", "Pr(>|z|)"]
}
sim_c <- function(data) {
    agg <- data |>
        group_by(Lucia) |>
        summarise(
            deaths = sum(deaths),
            shift_count = sum(shift_count)
        )

    fit_c <- glm(
        deaths ~ Lucia,
        family = poisson(link = "log"),
        data = agg,
        offset = log(shift_count)
    )

    summary(fit_c)$coefficients["Lucia", "Pr(>|z|)"]
}

sim_d <- function(data) {
    n <- data |>
        filter(Lucia == FALSE) |>
        summarise(shift_count = sum(shift_count)) |>
        pull()

    m <- data |>
        filter(Lucia == TRUE) |>
        summarise(shift_count = sum(shift_count)) |>
        pull()

    x <- data |>
```

```
        filter(Lucia == TRUE) |>
        summarise(deaths = sum(deaths)) |>
        pull()

    k <- data |>
        summarise(deaths = sum(deaths)) |>
        pull()

    phyper(x - 1, m, n, k, lower.tail = FALSE)
}

counter_b <- 0
counter_c <- 0
counter_d <- 0
# We the number of simlutations that cross our P value Threshhold.
for (i in 1:NUM_SIMULATIONS) {
    d <- gen_data()
    counter_b <- counter_b + as.integer(sim_b(data = d) < P_THRESH)
    counter_c <- counter_c + as.integer(sim_c(data = d) < P_THRESH)
    counter_d <- counter_d + as.integer(sim_d(data = d) < P_THRESH)
}

cat(
    sprintf("Number of pvalues below threshhold (%s)\n", P_THRESH),
    "=====\n",
    sprintf(
        "Simulation b: %s/%s (%s percent)\n",
        counter_b, NUM_SIMULATIONS, (counter_b / NUM_SIMULATIONS) * 100
    ),
    sprintf(
        "Simulation c: %s/%s (%s percent)\n",
        counter_c, NUM_SIMULATIONS, (counter_c / NUM_SIMULATIONS) * 100
    ),
    sprintf(
        "Simulation d: %s/%s (%s percent)\n",
        counter_d, NUM_SIMULATIONS, (counter_d / NUM_SIMULATIONS) * 100
    )
)
```

```
## Number of pvalues below threshhold (0.05)
##  =====
##  Simulation b: 7/200 (3.5 percent)
##  Simulation c: 104/200 (52 percent)
##  Simulation d: 116/200 (58 percent)
```

Interpretation Across 200 simulations, the proportion of rejections at the 0.05 level gives an estimate of Type I error / FP rate under each of the simulation's assumptions:

- Simulation (b): Poisson regression with both factors morning and Lucia gives p-values indicating Lucia does not significantly affects the death rate.
  Simulation b accounts for the effect of morning shift ≈ 5 percent, Showing correct Type I error control - FP occur only by chance, as expected under the null hypothesis.

- Simulation (c): Aggregating over morning shifts shows the effect of Lucia alone.
  Simulation c collapses over morning shifts introducing confounding, causing substantially higher FP rates. This demonstrates how ignoring known confounder can cause an effect even when none exists.

- Simulation (d): Hypergeometric approach calculates the exact p-value under the null hypothesis that Lucia's rate equals other nurses.
  Simulation d assumes incidents are randomly distributed accross all types of shifts resulting and demonstrates both ignoring known confounders and use of a wrong sampling model. This results in the highest FP rate of the three.