HW4_Part2

Bulun Te

2024-04-06

Problem 2

loading data

```
setwd(here::here())
library(dplyr)
```

```
##
##    'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(readxl)

sheet1 = read_excel("ScreenTime-hw3Q3.xlsx",sheet=1)
sheet2 = read_excel("ScreenTime-hw3Q3.xlsx",sheet=2)

sheet1$pseudo_id <- as.character(sheet1$pseudo_id)
sheet2$pseudo_id <- as.character(sheet2$pseudo_id)

merged_data <- left_join(sheet1, sheet2, by = "pseudo_id")
selected_data <- merged_data %>% filter(Treatment == "B")
pesudo_id_list = unique(selected_data$pseudo_id)

rbind(merged_data%>%head(),merged_data%>%tail()) %>% knitr::kable()
```

Day	Tot.Scr.Time	Tot.Soc.Time	Pickups	pseudo_id	time	Phase	Treatment	sex	age	pets	siblings
Fr	99	11	158	1	1	Baseline	A	1	26	0	1
Sa	83	15	94	1	2	Baseline	A	1	26	0	1
Su	135	12	145	1	3	Baseline	A	1	26	0	1

Day	Tot.Scr.Time	Tot.Soc.Time	Pickups	pseudo_id	time	Phase	Treatment	sex	age	pets	siblings
Mo	95	30	177	1	4	Baseline	A	1	26	0	1
Tu	153	16	142	1	5	Baseline	A	1	26	0	1
We	106	19	162	1	6	Baseline	A	1	26	0	1
We	408	143	153	24	13	Baseline	A	0	21	0	3
Th	295	163	131	24	14	Baseline	A	0	21	0	3
Fr	268	90	188	24	15	Treatment	A	0	21	0	3
Sa	270	129	140	24	16	Treatment	A	0	21	0	3
Su	235	125	126	24	17	Treatment	A	0	21	0	3
Mo	386	167	138	24	18	Treatment	A	0	21	0	3

(a)

```r
# Create an empty list to store the model results
model_results <- list()

# Loop through pseudo_id 1 to 8
for (id in pesudo_id_list) {
  # Subset the data for each pseudo_id, create lag-1 variable for Pickups, and create weekday dummy var
  subset_data <- merged_data %>%
    filter(pseudo_id == id) %>%
    mutate(
      lag1_Pickups = c(NA, Pickups[-n()]),
      weekday = ifelse(Day %in% c("Mo", "Tu", "We", "Th", "Fr"), 1, 0),
      B = ifelse(Phase == "Treatment", 1, 0)
    )

  # Fit the Poisson regression model with offset
  model <- glm(Pickups ~ log(lag1_Pickups) + B + weekday,
               family = poisson(link = "log"),
               offset = log(Tot.Scr.Time),
               data = subset_data)

  # Store the model results in the list
  model_results[[id]] <- model
}

# Create a summary table with estimates and standard errors
summary_table <- tibble(
  id = pesudo_id_list,
  beta0_est = sapply(model_results, function(x) coef(x)[1]),
  beta0_se = sapply(model_results, function(x) summary(x)$coefficients[1, 2]),
  beta1_est = sapply(model_results, function(x) coef(x)[2]),
  beta1_se = sapply(model_results, function(x) summary(x)$coefficients[2, 2]),
  beta2_est = sapply(model_results, function(x) coef(x)[3]),
  beta2_se = sapply(model_results, function(x) summary(x)$coefficients[3, 2]),
  beta3_est = sapply(model_results, function(x) coef(x)[4]),
  beta3_se = sapply(model_results, function(x) summary(x)$coefficients[4, 2])
)

# Print the summary table
```

```
summary_table %>% knitr::kable()
```

id	beta0_est	beta0_se	beta1_est	beta1_se	beta2_est	beta2_se	beta3_est	beta3_se
2	-3.1157167	0.4026836	0.2971298	0.0860509	0.5786629	0.0530414	0.2226030	0.0521819
3	-2.9194709	0.5419778	0.2738430	0.1185552	0.1422284	0.0677840	0.2351549	0.0562976
4	-1.2932423	0.4268217	0.0514133	0.0887010	0.1078503	0.0476339	0.6845209	0.0488738
5	-3.0854735	0.4724012	0.3195853	0.0999752	0.6046964	0.0529061	0.3061989	0.0483575
8	-4.1285556	0.4154342	0.4065129	0.0988193	0.9627066	0.0715000	0.4867084	0.0731790
15	-1.5347885	0.5827898	-0.0413302	0.1293074	0.1189347	0.0681475	0.3410277	0.0640200
16	-1.6063364	0.3828135	0.1556429	0.0839278	0.3465099	0.0572625	0.0209523	0.0566560
18	0.3764456	0.3463724	-0.2182515	0.0727400	0.5211998	0.0510201	-0.0139274	0.0531304

(b)

```
meta_learning <- function(summary_table) {
  # Extract the number of users
  K <- nrow(summary_table)

  # Compute the weights for each beta coefficient and each user
  weights_beta0 <- 1 / summary_table$beta0_se^2
  weights_beta1 <- 1 / summary_table$beta1_se^2
  weights_beta2 <- 1 / summary_table$beta2_se^2
  weights_beta3 <- 1 / summary_table$beta3_se^2

  # Compute the meta-estimate for each beta coefficient
  beta0_meta <- sum(summary_table$beta0_est * weights_beta0) / sum(weights_beta0)
  beta1_meta <- sum(summary_table$beta1_est * weights_beta1) / sum(weights_beta1)
  beta2_meta <- sum(summary_table$beta2_est * weights_beta2) / sum(weights_beta2)
  beta3_meta <- sum(summary_table$beta3_est * weights_beta3) / sum(weights_beta3)

  # Compute the variance of the meta-estimate for each beta coefficient
  beta0_meta_var <- 1 / sum(weights_beta0)
  beta1_meta_var <- 1 / sum(weights_beta1)
  beta2_meta_var <- 1 / sum(weights_beta2)
  beta3_meta_var <- 1 / sum(weights_beta3)

  # Compute the standard error of the meta-estimate for each beta coefficient
  beta0_meta_se <- sqrt(beta0_meta_var)
  beta1_meta_se <- sqrt(beta1_meta_var)
  beta2_meta_se <- sqrt(beta2_meta_var)
  beta3_meta_se <- sqrt(beta3_meta_var)

  meta_summary_table <- tibble(
    beta0_est = beta0_meta,
    beta0_se = beta0_meta_se,
    beta1_est = beta1_meta,
    beta1_se = beta1_meta_se,
    beta2_est = beta2_meta,
    beta2_se = beta2_meta_se,
    beta3_est = beta3_meta,
    beta3_se = beta3_meta_se
  )

  # Return the meta-learning summary table
  return(meta_summary_table)
}

# Use the summary table from the previous code
meta_results <- meta_learning(summary_table)

# Print the meta-estimates and their standard errors
meta_results %>% knitr::kable()
```

beta0_est	beta0_se	beta1_est	beta1_se	beta2_est	beta2_se	beta3_est	beta3_se
-1.987676	0.1517261	0.1274914	0.0328528	0.4127761	0.0201164	0.2853281	0.0195159

(c)

testing significany of the Treatment (denoted as B) at level 0.05

```
# Compute the z-statistic for the Treatment effect
z_stat <- (meta_results$beta2_est - 0) / meta_results$beta2_se
# Compute the p-value for the Treatment effect
p_value <- 2 * (1 - pnorm(abs(z_stat)))
# Print the z-statistic and p-value for the Treatment effect
print(paste("z-statistic:", z_stat))
```

```
## [1] "z-statistic: 20.5193624434363"
```

```
print(paste("p-value:", p_value))
```

```
## [1] "p-value: 0"
```

Based on the test statistics, under 5% significancy level, we can reject the null hypothesis that the treatment effect is zero.

(d)

`Advantages`:

Flexibility and computational efficiency: Meta-learning is more flexible and computationally easier than federated learning. It can handle nonlinear regression models, such as GLMs and Cox PH model, where closed-form expressions for estimation and inference are not available. In contrast, federated learning may lack flexibility and can be computationally expensive when dealing with nonlinear models that require iterative algorithms for parameter estimation.

Minimal data sharing: Meta-learning requires minimal data sharing between local sites and the central server. Only summary statistics (point estimates and their variances or standard errors) need to be shared, regardless of the statistical problem. This is advantageous for privacy and security concerns. In federated learning, the communication between local sites and the central platform can be expensive, especially when iterative updates of summary statistics are required.

`Disadvantages`:

Assumption of independent samples: Meta-learning assumes that the samples from different study sites are independent. This assumption may not always hold in practice, and violations of this assumption can lead to biased or incorrect results. Federated learning, on the other hand, does not necessarily require the assumption of independent samples across sites.

Assumption of homogeneous target model parameters: Meta-learning assumes that the target model parameters are homogeneous across study sites. However, this assumption may be violated in some situations, leading to biased or suboptimal results. In federated learning, the assumption of model homogeneity is also strong, but it can be relaxed by allowing for site-specific model parameters or by using techniques like model averaging.

Problem 3

(a)

load and merge the data

```r
sheet1 = read_excel("ScreenTime-hw3Q3.xlsx",sheet=1)
sheet2 = read_excel("ScreenTime-hw3Q3.xlsx",sheet=2)

sheet1$pseudo_id <- as.character(sheet1$pseudo_id)
sheet2$pseudo_id <- as.character(sheet2$pseudo_id)

merged_data <- left_join(sheet1, sheet2, by = "pseudo_id")

rbind(merged_data%>%head(),merged_data%>%tail()) %>% knitr::kable()
```

Day	Tot.Scr.Time	Tot.Soc.Time	Pickups	pseudo_id	time	Phase	Treatment	sex	age	pets	siblings
Fr	99	11	158	1	1	Baseline	A	1	26	0	1
Sa	83	15	94	1	2	Baseline	A	1	26	0	1
Su	135	12	145	1	3	Baseline	A	1	26	0	1
Mo	95	30	177	1	4	Baseline	A	1	26	0	1
Tu	153	16	142	1	5	Baseline	A	1	26	0	1
We	106	19	162	1	6	Baseline	A	1	26	0	1
We	408	143	153	24	13	Baseline	A	0	21	0	3
Th	295	163	131	24	14	Baseline	A	0	21	0	3
Fr	268	90	188	24	15	Treatment	A	0	21	0	3
Sa	270	129	140	24	16	Treatment	A	0	21	0	3
Su	235	125	126	24	17	Treatment	A	0	21	0	3
Mo	386	167	138	24	18	Treatment	A	0	21	0	3

```r
selected_data_A = merged_data %>% filter(Treatment == "A") %>% mutate(
  lag1_Pickups = c(NA, Pickups[-n()]),
  weekday = ifelse(Day %in% c("Mo", "Tu", "We", "Th", "Fr"), 1, 0),
  A = ifelse(Phase == "Treatment", 1, 0)
)
```

```r
model_a <- glm(
  Pickups ~ log(lag1_Pickups)+A+weekday+sex+age+pets+siblings,
  family = poisson(link = "log"),
  offset = log(Tot.Scr.Time),
  data = selected_data_A
)

summary(model_a)$coefficients[,c(1,2)] %>% data.frame() -> model_a_summary



selected_data_B = merged_data %>% filter(Treatment == "B") %>% mutate(
  lag1_Pickups = c(NA, Pickups[-n()]),
  weekday = ifelse(Day %in% c("Mo", "Tu", "We", "Th", "Fr"), 1, 0),
  B = ifelse(Phase == "Treatment", 1, 0)
)

model_b <- glm(
  Pickups ~ log(lag1_Pickups)+B+weekday+sex+age+pets+siblings,
  family = poisson(link = "log"),
  offset = log(Tot.Scr.Time),
  data = selected_data_B
)

summary(model_b)$coefficients[,c(1,2)] %>% data.frame() -> model_b_summary


sqrt((1/model_a_summary[2]^2 + 1/model_b_summary[2]^2)^(-1)) -> meta_se
((model_a_summary[1]/model_a_summary[2]^2+
    model_b_summary[1]/model_b_summary[2]^2)*meta_se^2) -> meta_est

cbind(meta_est,meta_se) -> meta_summary_table

meta_summary_table = meta_summary_table %>% mutate(
  z = unlist(meta_est/meta_se),
  p = 2*(1-pnorm(abs(z)))
)

meta_summary_table %>% knitr::kable()
```

	Estimate	Std..Error	z	p
(Intercept)	-2.5587981	0.1874143	-13.653162	0e+00
log(lag1_Pickups)	0.0856882	0.0161118	5.318361	1e-07
A	0.2823478	0.0147468	19.146361	0e+00
weekday	0.2189670	0.0127560	17.165798	0e+00
sex	0.1142330	0.0185326	6.163883	0e+00
age	0.0493835	0.0066101	7.470860	0e+00
pets	-0.2699536	0.0182349	-14.804201	0e+00
siblings	0.1663715	0.0070520	23.592228	0e+00

(b)

```r
# compute the chisq statistics

chisq_stat <- ((meta_est[3,1] - 0) / meta_se[3,1])^2

# compute the p-value
p_value <- 1 - pchisq(chisq_stat, df = 1)

# print the chisq statistics and p-value

print(paste("chisq-statistic:", chisq_stat))
```

```
## [1] "chisq-statistic: 366.58315104207"
```

```r
print(paste("p-value:", p_value))
```

```
## [1] "p-value: 0"
```

Based on the meta learning statistics, under 5% significancy level, reject the null hypothesis. The intervention has a significant effect on the daily number of pickups in comparison to the pre-intervention baseline screen activity.

(c)

```r
selected_data_C <- merged_data %>%
  filter(Treatment != "P") %>% mutate(
  lag1_Pickups = c(NA, Pickups[-n()]),
  weekday = ifelse(Day %in% c("Mo", "Tu", "We", "Th", "Fr"), 1, 0),
  R = ifelse(Phase == "Treatment", 1, 0)
)


model_c <- glm(
  Pickups ~ log(lag1_Pickups)+R+weekday+sex+age+pets+siblings,
  family = poisson(link = "log"),
  offset = log(Tot.Scr.Time),
  data = selected_data_C
)

summary(model_c)$coefficients %>% data.frame() %>% knitr::kable()
```

	Estimate	Std..Error	z.value	Pr...z..
(Intercept)	-4.9041282	0.1331577	-36.82947	0
log(lag1_Pickups)	0.3821349	0.0150289	25.42673	0
R	0.2864520	0.0146019	19.61750	0
weekday	0.1705129	0.0126181	13.51332	0
sex	0.5928169	0.0158322	37.44376	0

	Estimate	Std..Error	z.value	Pr...z..
age	0.0712765	0.0047079	15.13963	0
pets	-0.3028059	0.0169734	-17.84006	0
siblings	0.2429020	0.0060189	40.35653	0

(d)

Based on the meta learning estimation and centralized model estimation, we could see that the estimation of coefficients and standard error by the meta learning are different from the version of centralized model estimation on all of the covariates. However, the staistical significance of coefficients are consistent between the two methods.