Lab 2

Ananya Bhaktaram

2025-09-16

Part A

Question 1:

You will explore the autocorrelation function within three hypothetical studies.

Each of the three hypothetical studies was generated assuming the following:

$$\begin{pmatrix} Y_{i0} \\ Y_{i1} \\ Y_{i2} \\ Y_{i3} \\ Y_{i4} \end{pmatrix} \sim MVN\left(\begin{pmatrix} 35 \\ 38 \\ 41 \\ 44 \\ 47 \end{pmatrix}, \begin{pmatrix} 100 & ? & ? & ? & ? \\ ? & 100 & ? & ? & ? \\ ? & ? & 100 & ? & ? \\ ? & ? & ? & 100 & ? \\ ? & ? & ? & ? & 100 \end{pmatrix}\right)$$

Your goal is to fill in the “?” within the variance specification for this k-variate normal distribution (k = 5). To fill in the “?’ you will be identifying the parametric model that defines the within subject correlation.

Go to the Courseplus site and find data from the three hypothetical studies: “autocor1.csv”, “autocor2.csv” and “autocor3.csv”.

Fill in the following table:

```
# Set working directory
setwd("C:/Users/anany/OneDrive/Hopkins/PhD/Year 3/Multilevel Stats I/Multi Stats Labs")

# Load relevant libraries
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.2     ✔ tibble    3.3.0
## ✔ lubridate 1.9.4     ✔ tidyr     1.3.1
## ✔ purrr     1.0.4
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
library(here)
```

```
## here() starts at C:/Users/anany/OneDrive/Hopkins/PhD/Year 3/Multilevel Stats I/Multi Stats La
bs
```

```
library(nlme)
```

```
##
## Attaching package: 'nlme'
##
## The following object is masked from 'package:dplyr':
##
##     collapse
```

```
library(mvtnorm)
library(ggplot2)

# Read in data
autocor1 <- read_csv("Lab 2/autocor1.csv")
```

```
## Rows: 500 Columns: 3
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (3): id, time, y
##
## 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.
```

```
autocor2 <- read_csv("Lab 2/autocor2.csv")
```

```
## Rows: 500 Columns: 3
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (3): id, time, y
##
## 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.
```

```
autocor3 <- read_csv("Lab 2/autocor3.csv")
```

```
## Rows: 500 Columns: 3
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## dbl (3): id, time, y
##
## 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.
```

```
# Explore data sets
head(autocor1); head(autocor2); head(autocor3)
```

```
## # A tibble: 6 × 3
##      id  time     y
##   <dbl> <dbl> <dbl>
## 1     1     0  39.0
## 2     1     1  37.1
## 3     1     2  40.7
## 4     1     3  35.6
## 5     1     4  50.1
## 6     2     0  41.8
```

```
## # A tibble: 6 × 3
##      id  time     y
##   <dbl> <dbl> <dbl>
## 1     1     0  22.1
## 2     1     1  24.1
## 3     1     2  21.2
## 4     1     3  29.5
## 5     1     4  27.0
## 6     2     0  46.4
```

```
## # A tibble: 6 × 3
##      id  time     y
##   <dbl> <dbl> <dbl>
## 1     1     0  24.7
## 2     1     1  30.3
## 3     1     2  41.9
## 4     1     3  48.7
## 5     1     4  56.8
## 6     2     0  41.6
```

Hypothetical Study 1

```
# Run ACF function
fit1 <- gls(y ~ as.factor(time), data = autocor1)
ACF(fit1, form= ~ 1|id)
```

```
##   lag       ACF
## 1   0 1.0000000
## 2   1 0.8562632
## 3   2 0.8201598
## 4   3 0.7414081
## 5   4 0.7309042
```

Hypothetical Study 2

```
# Run ACF function
fit2 <- gls(y ~ as.factor(time), data = autocor2)
ACF(fit2, form= ~ 1|id)
```

```
##   lag       ACF
## 1   0 1.0000000
## 2   1 0.8253296
## 3   2 0.8364482
## 4   3 0.7888892
## 5   4 0.8197277
```

Hypothetical Study 3

```
# Run ACF function
fit3 <- gls(y ~ as.factor(time), data = autocor3)
ACF(fit3, form= ~ 1|id)
```

```
##   lag       ACF
## 1   0 1.0000000
## 2   1 0.7787847
## 3   2 0.5861138
## 4   3 0.3825778
## 5   4 0.2686456
```

ACF Summary Table

```
library(knitr)
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
acf_table <- data.frame(
  Lag = c(1:4, "Parametric Model:"),
  Study1 = c(0.856, 0.820, 0.741, 0.731, "Toeplitz"),
  Study2 = c(0.825, 0.836, 0.788, 0.819, "Exchangeable"),
  Study3 = c(0.779, 0.586, 0.383, 0.269, "Autoregressive")
)

kable(acf_table,
      col.names = c("Lag", "Sample autocorrelation function",
                    "Sample autocorrelation function",
                    "Sample autocorrelation function")) %>%
  kable_styling(bootstrap_options = c("striped", "hover")) %>%
  add_header_above(c(" " = 1,
                     "Hypothetical Study 1" = 1,
                     "Hypothetical Study 2" = 1,
                     "Hypothetical Study 3" = 1))
```

	Hypothetical Study 1	**Hypothetical Study 2**	**Hypothetical Study 3**
Lag	**Sample autocorrelation function**	**Sample autocorrelation function**	**Sample autocorrelation function**
1	0.856	0.825	0.779
2	0.82	0.836	0.586
3	0.741	0.788	0.383
4	0.731	0.819	0.269
Parametric Model:	Toeplitz	Exchangeable	Autoregressive

Question 2:

For each of the three hypothetical studies, estimate the monthly improvement in SF-36 mental health scores using both ordinary least squares (OLS) and weighted least squares (WLS) with an unstructured variance model; treat time as a linear variable. Fill in the table below:

Estimating OLS & WLS for Hypothetical Study 1

```
# OLS model (treats time as linear)
fit.ols1 <- lm(y ~ time, data = autocor1); summary(fit.ols1)
```

```
##
## Call:
## lm(formula = y ~ time, data = autocor1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -26.073  -6.944  -0.387   6.856  35.216
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.3334     0.8055  45.107   <2e-16 ***
## time          2.8301     0.3288   8.606   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.4 on 498 degrees of freedom
## Multiple R-squared:  0.1295, Adjusted R-squared:  0.1277
## F-statistic: 74.07 on 1 and 498 DF,  p-value: < 2.2e-16
```

```
# WLS with unstructured variance model
fit.wls1 <- gls(y ~ time,
                correlation = corSymm(form = ~1|id),
                weights=varIdent(form= ~1|as.factor(time)),
                data= autocor1, method="ML");
coef(summary(fit.wls1))
```

```
##                 Value Std.Error  t-value       p-value
## (Intercept) 36.093440 0.9523073 37.90104 7.456605e-149
## time         2.877421 0.1753376 16.41075  1.073654e-48
```

Estimating OLS & WLS for Hypothetical Study 2

```
# OLS model (treats time as linear)
fit.ols2 <- lm(y ~ time, data = autocor2); summary(fit.ols2)
```

```
##
## Call:
## lm(formula = y ~ time, data = autocor2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -25.886  -6.252  -0.955   6.240  33.702
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  34.2026     0.7277  47.001   <2e-16 ***
## time          2.9347     0.2971   9.878   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.395 on 498 degrees of freedom
## Multiple R-squared:  0.1638, Adjusted R-squared:  0.1622
## F-statistic: 97.58 on 1 and 498 DF,  p-value: < 2.2e-16
```

```
# WLS with unstructured variance model
fit.wls2 <- gls(y ~ time,
                correlation = corSymm(form = ~1|id),
                weights=varIdent(form= ~1|as.factor(time)),
                data= autocor2, method="ML");
coef(summary(fit.wls2))
```

```
##                 Value Std.Error  t-value       p-value
## (Intercept) 34.336939 0.8516651 40.31742 6.099062e-159
## time         2.932624 0.1232371 23.79661  3.653369e-84
```

Estimating OLS and WLS for Hypothetical Study 3

```
# OLS model (treats time as linear)
fit.ols3 <- lm(y ~ time, data = autocor3); summary(fit.ols3)
```

```
##
## Call:
## lm(formula = y ~ time, data = autocor3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.3070  -5.6904   0.3031   6.0282  28.0214
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.4805     0.6963  52.391   <2e-16 ***
## time          2.8237     0.2843   9.933   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.989 on 498 degrees of freedom
## Multiple R-squared:  0.1654, Adjusted R-squared:  0.1637
## F-statistic: 98.67 on 1 and 498 DF,  p-value: < 2.2e-16
```

```
# WLS with unstructured variance model
fit.wls3 <- gls(y ~ time,
                correlation = corSymm(form = ~1|id),
                weights=varIdent(form= ~1|as.factor(time)),
                data= autocor3, method="ML");
coef(summary(fit.wls3))
```

```
##                 Value Std.Error  t-value       p-value
## (Intercept) 36.405771 0.9226540 39.45766 2.177413e-155
## time         2.825801 0.2736478 10.32642  8.795040e-23
```

OLS & WLS Summary Table

```
sf36_table <- data.frame(
  Data = c("autocor1", "autocor2", "autocor3"),
  OLS_Coef = c(2.830, 2.934, 2.823),
  WLS_Coef = c(2.877, 2.933, 2.825),
  OLS_SE = c(0.329, 0.297, 0.284),
  WLS_SE = c(0.175, 0.123, 0.274)
)

kable(sf36_table,
      col.names = c("Data", "OLS", "WLS", "OLS", "WLS"),
      caption = "Monthly improvement in SF-36") %>%
  kable_styling(bootstrap_options = c("striped", "hover")) %>%
  add_header_above(c(" " = 1, "Coefficient" = 2, "SE" = 2))
```

Monthly improvement in SF-36

	Coefficient		SE	
Data	**OLS**	**WLS**	**OLS**	**WLS**
autocor1	2.830	2.877	0.329	0.175
autocor2	2.934	2.933	0.297	0.123
autocor3	2.823	2.825	0.284	0.274

What is the true monthly improvement in SF-36 mental health scores?

Using the information provided by the model specification at the beginning of the lab the true monthly improvement in SF-36 scores is $(47-45)/(4-0) = 3$ points per month increase in SF-36 mental health scores after ICU discharge.

Do the estimated monthly improvement in SF-36 mental health scores differ across the two statistical methods? If so, why?

No, both the OLS and WLS produced similar coefficients across all three hypothetical studies. This shows that there is a general increasing trend between 2.8-2.9 in SF-36 mental health scores at each monthly follow-up after being discharged from the ICU.

Do the standard errors for the estimated monthly improvement in SF-36 mental health scores differ across the two statistical methods? If so, why?

Yes, the WLS is more accurate than the OLS because it produced smaller standard errors, given that the model re-weights the observations to obtain a new regression that incorporates unbiased estimater coefficients alongside uncorrelated residuals.

Part B

Question 3:

Above, you compared the estimated monthly improvement in SF-36 mental health scores generated from the OLS and WLS procedures from a single study of 100 patients. Here, we will explore the repeated sampling behavior of the estimated monthly improvement in SF-36 mental health scores assuming various models for the within subject variance across increasing sample sizes. You will identify important patterns in the behavior of the estimates based on the specified model for the variance. NOTE: We are exploring the properties of WLS within the context of no missing data; we will consider missing data in more detail later in the course.

SIMULATION STUDY: Please DO NOT try to run this simulation study on your laptop; I ran the simulation in R on the Department of Biostatistics computing cluster using 25 parallel processing cores and it took a couple of hours to complete.

I generated 10,000 simulated studies each for m = 10, 25, 100, 500 and 1000 patients. The patients were sampled from a population of patients whose data follows the k-variate normal distribution (k = 5) below:

$$\begin{pmatrix} Y_{i0} \\ Y_{i1} \\ Y_{i2} \\ Y_{i3} \\ Y_{i4} \end{pmatrix} \sim MVN\left(\begin{pmatrix} 35 \\ 38 \\ 41 \\ 44 \\ 47 \end{pmatrix}, V_i = \begin{pmatrix} 100 & 100\times\rho & 100\times\rho^2 & 100\times\rho^3 & 100\times\rho^4 \\ 100\times\rho & 100 & 100\times\rho & 100\times\rho^2 & 100\times\rho^3 \\ 100\times\rho^2 & 100\times\rho & 100 & 100\times\rho & 100\times\rho^2 \\ 100\times\rho^3 & 100\times\rho^2 & 100\times\rho & 100 & 100\times\rho \\ 100\times\rho^4 & 100\times\rho^3 & 100\times\rho^2 & 100\times\rho & 100 \end{pmatrix}\right)$$

Therefore, the patients are sampled from a population where the monthly improvement in the SF-36 mental health scores is 3 units, the variance of the SF-36 mental health scores at any time is 100 and the correlation between any two SF-36 mental health scores is given by $Corr\left(Y_{ij}, Y_{ik}\right) = \rho^{|j-k|}$ and ρ = 0.9; the AR1 model.

In each of the 10,000 simulated studies, I estimated the monthly improvement in SF-36 mental health scores using the correct model for the mean (linear function of month) and the following models for the within subject variance:

```
WLS – V known:  I provided the correct information for Vi; i.e. $Var\left(Y_{ij}\right)=100\ $ and $Corr\left(Y_{ij},Y_{ik}\right)={0.9}^{|j-k|}$
WLS $– V estimated:  I assumed the correct model for Vi but I estimated the required parameters within each of the simulated studies
WLS – V unstructured:  I did not assume a model for Vi so estimated 5 variance and 10 correlation parameters within each of the simulated studies.
OLS:  I assumed the SF-36 mental health scores from the same subject were uncorrelated and that the variance of the SF-36 mental health scores was the same at all the measurement times and estimated the variance within each of the simulated studies.
```

The table below displays the bias and variance of the 10,000 estimated monthly improvements in SF-36 mental health scores based on different sample sizes and the models i. through iv. The bias is defined as the average of the 10,000 estimated monthly improvements in SF-36 mental health scores over the simulated studies minus 3 (the true monthly improvement).

Provided Summary Table

```
simulation_table <- data.frame(
  Sample_size = c(10, 25, 100, 500, 1000),
  Bias_WLS_known = c(-0.005, -0.004, -0.002, -0.0005, -0.0006),
  Bias_WLS_estimated = c(-0.005, -0.004, -0.002, -0.0005, -0.0007),
  Bias_WLS_unstructured = c(-0.007, -0.003, -0.002, -0.0004, -0.0007),
  Bias_OLS = c(-0.003, -0.003, -0.002, -0.0003, -0.0005),
  Variance_WLS_known = c(0.422, 0.172, 0.0430, 0.00883, 0.00439),
  Variance_WLS_estimated = c(0.422, 0.172, 0.0430, 0.00883, 0.00439),
  Variance_WLS_unstructured = c(0.678, 0.200, 0.0443, 0.00888, 0.00440),
  Variance_OLS = c(0.442, 0.181, 0.0451, 0.00925, 0.00458)
)

# Create the table with proper headers
kable(simulation_table,
      col.names = c("Sample size (m)",
                    "WLS – V known", "WLS – V estimated", "WLS - unstructured", "OLS",
                    "WLS – V known", "WLS – V estimated", "WLS - unstructured", "OLS"),
      caption = "Simulation Results: Bias and Variance") %>%
  kable_styling(bootstrap_options = c("striped", "hover")) %>%
  add_header_above(c(" " = 1, "Bias" = 4, "Variance" = 4))
```

Simulation Results: Bias and Variance

	Bias				Variance			
Sample size (m)	**WLS – V known**	**WLS – V estimated**	**WLS - unstructured**	**OLS**	**WLS – V known**	**WLS – V estimated**	**WLS - unstructured**	**OLS**
10	-5e-03	-5e-03	-7e-03	-3e-03	0.42200	0.42200	0.67800	0.44200
25	-4e-03	-4e-03	-3e-03	-3e-03	0.17200	0.17200	0.20000	0.18100
100	-2e-03	-2e-03	-2e-03	-2e-03	0.04300	0.04300	0.04430	0.04510
500	-5e-04	-5e-04	-4e-04	-3e-04	0.00883	0.00883	0.00888	0.00925
1000	-6e-04	-7e-04	-7e-04	-5e-04	0.00439	0.00439	0.00440	0.00458

Based on the results in the table answer the following questions:

a. For a fixed sample size, how does the bias compare across the models specified by the within subject variance?

While variance specification can dramatically effect the variance of estimates, which can be seen in the right half of the summary table, but for a fixed sample size, the choice of within-subject variance specification has minimal impact on bias. This can be seen because all four methods produced nearly unbiased estimates of the monthly improvement parameter.

b. Regardless of the model selected for the within subject variance, how does the bias change as the sample size goes from small (m = 10) to large (m = 1000)?

The bias consistently decreases toward zero as the sample size increases from small(m=10) to large(m=1000). This tells us that all four methods begin to converge toward the true parametric value as the sample size increases, independent of the within-subject variance.

c. Compare the variance across WLS – V known to WLS – V estimated for fixed sample sizes.

Across all sample sizes, the WLS with known variance and WLS with estimated variance models produced identical variances. This indicates that uncertainty from estimating the variance-covariance structure did not result in increased variability in the coefficient estimates. V is deined by two values, the common variance, and the autoregressive correlation

d. For small sample sizes (m = 10 and m = 25), compare the variance for WLS – V known and WLS – V estimated to WLS – unstructured? If there are differences, what do you think drives the differences?

There is higher variance in the WLS-unstructured approach, when compared to the structured approaches (WLS-V known and WLS-V estimated). At sample size (m=10) the WLS-V known and WLS-V estimated is 0.422, while the WLS-unstructured is 0.678. This is a 61% higher variance at the smaller sample size. When

the sample size is increased to m=25, the estimated variance for the structured models is 0.172, while the variance for WLS-unstructured is 0.200. At the larger sample size the difference in variance has shrunk to only being 16% higher. This is because when sample sizes are relatively small to the number of correlation parameters being estimated, the unstructured approach suffers from reduced statistical efficiency. As the sample size increases the difference decreases because there is more data per parameter to estimate– improving statistical efficiency.

e. For larger sample sizes (m = 100 to m = 1000), compare the variance for WLS – V known and WLS – V estimated to WLS – unstructured? If there are differences, what do you think drives the differences?

At larger sample sizes (m=100 to m=1000) the differences between the structured and unstructured approaches becomes negligible because there is more data per parameter which results in convergence and the production of nearly identical values from both the structured and unstructured approaches. At m=100 the unstructured only has 3% higher variance (0.0443 vs 0.0430). At m=500 the difference is reduced to 0.6% higher variance in the unstructured (0.00888 vs 0.00883). Finally, with a sample size of m=1000, the unstructured only has a 0.2% higher variance (0.00440 vs 0.00439).

f. For each of the sample sizes considered, compute the relative efficiency of estimating the monthly improvement assuming the correct variance model to assuming independence.

$$\frac{\text{Var}_{OLS}}{\text{Var}_{V-estimated}} = \text{Relative Efficiency}$$

```
# Relative Efficiency at m=10
RE_10 = (0.422/0.422)

cat("The relative efficiency at sample size m =10 is:", round(RE_10,3))
```

```
## The relative efficiency at sample size m =10 is: 1
```

```
# Relative Efficiency at m=25
RE_25 = (0.181/0.172)

cat("The relative efficiency at sample size m =25 is:", round(RE_25,3))
```

```
## The relative efficiency at sample size m =25 is: 1.052
```

```
# Relative Efficiency at m=100
RE_100 = (0.0451/0.0439)

cat("The relative efficiency at sample size m =100 is:", round(RE_100,3))
```

```
## The relative efficiency at sample size m =100 is: 1.027
```

```
# Relative Efficiency at m=500
RE_500 = (0.00925/0.00883)

cat("The relative efficiency at sample size m =500 is:", round(RE_500,3))
```

```
## The relative efficiency at sample size m =500 is: 1.048
```

```
# Relative Efficiency at m=1000
RE_1000 = (0.00458/0.00439)

cat("The relative efficiency at sample size m =1000 is:", round(RE_1000,3))
```

```
## The relative efficiency at sample size m =1000 is: 1.043
```

General properties of WLS and frequently asked questions:

I. Regardless of how we specify the model for V, the WLS procedure produces an unbiased estimate of the monthly improvement in SF-36 mental health scores.

II. However, specifying the wrong model for V can have an impact on your inference! EXAMPLE: the OLS estimator produces an estimate of the variance that is 5% too large!

III. Why not always use the unstructured approach? ANSWER: in small samples, you found that the variance for the WLS- unstructured was inflated and what defines “small samples” will change depending on the complexity of the mean model and n (the number of observations within a subject). In addition, sometimes there are unexpected dependencies in the empirical estimate of V using the unstructured approach so the model doesn’t fit (i.e. we are unable to invert the estimated V).

IV. Why not always use OLS? You could if your goal is only estimation (i.e. interest is in prediction) not inference (hypothesis testing, confidence intervals). If you are interested in inference, then you can get the wrong answer!

V. Our simulation study focused on data generated from an underlying multivariate normal distribution. After we estimate the monthly improvement in SF-36 mental health scores, we want to estimate a confidence interval. Our confidence interval methods rely on the assumption that the slope estimate is normally distributed. Even if the data are not normally distributed, the normality of the slope estimates hold in large samples, due to the central limit theorem.