ST552 Homework 7

Brian Cervantes Alvarez
March 12, 2024

Problem 1

Part A

Given the generalized regression model:

$$Y_{n\times 1} = X_{n\times p}\beta_{p\times 1} + \varepsilon_{n\times 1}$$

where $Var(\varepsilon) = \sigma^2\Sigma_{n\times n}$ and Σ is known and positive definite.

The least squares estimate of $\hat{\beta}_{OLS}$ is defined as follows:

$$\hat{\beta}_{OLS} = (X^TX)^{-1}X^TY$$

We know that the OLS estimator is unbiased when the expectation of the error terms is zero and the errors are uncorrelated, meaning $E[\varepsilon] = 0$ and $Var(\varepsilon) = \sigma^2 I$.

In our case, since $Var(\varepsilon) = \sigma^2\Sigma$ and $\Sigma \neq I$, the assumption of constant variance and independence is violated. Nevertheless, the expectation of $\hat{\beta}_{OLS}$ is equal to β.

Here is the proof:

$$E[\hat{\beta}_{OLS}] = E[(X^TX)^{-1}X^T(Y)] = E[(X^TX)^{-1}X^T(X\beta + \varepsilon)] = \beta + (X^TX)^{-1}X^TE[\varepsilon]$$

Since $E[\varepsilon] = 0$, this simplifies to:

$$E[\hat{\beta}_{OLS}] = \beta$$

Despite the clear violation of constant variance and independence, the OLS estimate of β remains unbiased.

Part B

The GLS estimator is:

$$\hat{\beta}_{GLS} = (X^T\Sigma^{-1}X)^{-1}X^T\Sigma^{-1}Y$$

To show that $\hat{\beta}_{GLS}$ is unbiased, consider the expectation:

$$\begin{aligned} E[\hat{\beta}_{\text{GLS}}] &= E\left[\left(X^T\Sigma^{-1}X\right)^{-1}X^T\Sigma^{-1}(X\beta+\varepsilon)\right] \\ &= \left(X^T\Sigma^{-1}X\right)^{-1}X^T\Sigma^{-1}X\beta + \left(X^T\Sigma^{-1}X\right)^{-1}X^T\Sigma^{-1}E[\varepsilon] \end{aligned}$$

Since $E[\varepsilon] = 0$, this reduces to:

$$E[\hat{\beta}_{GLS}] = (X^T\Sigma^{-1}X)^{-1}(X^T\Sigma^{-1}X)\beta = I\beta = \beta$$

Hence, $\hat{\beta}_{GLS}$ is an unbiased estimate of β.

Part C

To demonstrate that the least squares estimate of β for the new regression model is $\hat{\beta}_{\text{GLS}} = (X^T\Sigma^{-1}X)^{-1}X^T\Sigma^{-1}Y$, we begin by considering the transformed linear regression model:

$$S^{-1}Y = S^{-1}X\beta + S^{-1}\varepsilon.$$

Let's define the transformations directly:

1. $Y^* = S^{-1}Y$,
2. $X^* = S^{-1}X$, and
3. $\varepsilon^* = S^{-1}\varepsilon$.

$Y^* = X^*\beta + \varepsilon^*$. Next,to minimize the sum of squared residuals we can apply:

$$X^{*T}Y^* = X^{*T}X^*\hat{\beta}_{\text{GLS}}.$$

Plugging in the expressions for X^* and Y^*, we get:

$$(S^{-1}X)^T(S^{-1}Y) = (S^{-1}X)^T(S^{-1}X)\hat{\beta}_{\text{GLS}}.$$

This simplifies to:

$$X^T(S^{-1})^TS^{-1}Y = X^T(S^{-1})^TS^{-1}X\hat{\beta}_{\text{GLS}}.$$

Given that $\Sigma = SS^T$, so $\Sigma^{-1} = (S^{-1})(S^{-1})^T$, we can rewrite the equation as:

$$X^T\Sigma^{-1}Y = X^T\Sigma^{-1}X\hat{\beta}_{\text{GLS}}.$$

Solving this equation for $\hat{\beta}_{\text{GLS}}$, we get:

$$(X^T\Sigma^{-1}X)^{-1}(X^T\Sigma^{-1}Y) = (X^T\Sigma^{-1}X)^{-1}(X^T\Sigma^{-1}X)\hat{\beta}_{\text{GLS}} = (X^T\Sigma^{-1}X)^{-1}(X^T\Sigma^{-1}Y) = I\hat{\beta}_{\text{GLS}}$$

$$\hat{\beta}_{\text{GLS}} = (X^T\Sigma^{-1}X)^{-1}X^T\Sigma^{-1}Y.$$

This demonstrates that the least squares estimate of β for the new regression equation is indeed $\hat{\beta}_{\text{GLS}} = (X^T\Sigma^{-1}X)^{-1}X^T\Sigma^{-1}Y$.

Problem 2

Part A

When we run the `colSums(is.nan(summary(model)$coefficients))` we find out that 60 predictors converge. This demonstates that we cannot fit all the predictors as it's leading to perfect multicollinearity and/or will result in an overfitted model.

```
library(pls)
library(olsrr)
library(glmnet)
library(dplyr)
library(faraway)

data(gasoline)
gasoline$NIR <- unclass(gasoline$NIR)
ds <- as.data.frame(gasoline$NIR)
ds$octane <- gasoline$octane
model <- lm(octane ~ ., data = ds)

colSums(is.nan(summary(model)$coefficients))
```

```
  Estimate Std. Error    t value   Pr(>|t|)
         0         60         60         60
```

```
#summary(model)
```

Part B

Here are the best explanatory variables using the forward selection method: `1208 nm + 1196 nm + 976 nm + 1692 nm + 970 nm + 1206 nm + 1056 nm + 1074 nm + 1098 nm`

```
fSelectionModel <- ols_step_forward_p(model, p_val = 0.05)
summary(fSelectionModel$model)
```

```
Call:
lm(formula = paste(response, "~", paste(preds, collapse = " + ")),
    data = l)

Residuals:
     Min       1Q   Median       3Q      Max
-0.43082 -0.09774  0.00706  0.12572  0.35202

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  100.8111     1.4668  68.726  < 2e-16 ***
`1208 nm`     37.6893    39.1939   0.962 0.340872
`1196 nm`     47.2937     3.3268  14.216  < 2e-16 ***
`976 nm`     230.0237   105.6168   2.178 0.034157 *
`1692 nm`     -3.1588     0.8184  -3.860 0.000326 ***
`970 nm`    -293.9540   124.9296  -2.353 0.022600 *
`1206 nm`   -139.8200    39.3453  -3.554 0.000840 ***
`1056 nm`    194.9291    55.3245   3.523 0.000921 ***
`1074 nm`   -268.4304    83.7947  -3.203 0.002365 **
`1098 nm`    176.9943    76.1575   2.324 0.024226 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1745 on 50 degrees of freedom
Multiple R-squared:  0.989, Adjusted R-squared:  0.987
F-statistic: 498.4 on 9 and 50 DF,  p-value: < 2.2e-16
```

```
selectedVariables <- names(coef(fSelectionModel$model))
print(selectedVariables[-1])
```

```
[1] "`1208 nm`" "`1196 nm`" "`976 nm`"  "`1692 nm`" "`970 nm`"  "`1206 nm`"
[7] "`1056 nm`" "`1074 nm`" "`1098 nm`"
```

Part C

Here are the best explanatory variables using the lasso selection method: 900 nm + 914 nm + 1208 nm + 1210 nm + 1220 nm + 1226 nm + 1230 nm + 1232 nm + 1362 nm + 1364 nm + 1368 nm + 1636 nm + 1638 nm + 1640 nm + 1688 nm + 1692 nm + 1694 nm

```
X <- as.matrix(ds[, -ncol(ds)])
Y <- ds[, ncol(ds)]
cvFit <- cv.glmnet(X, Y, alpha = 1)
minLambda <- cvFit$lambda.min
lassoModel <- glmnet(X, Y, alpha = 1, lambda = minLambda)
lassoCoefs <- coef(lassoModel)
nonZeroCoefs <- which(lassoCoefs != 0)
selectedVars <- colnames(X)[nonZeroCoefs]
print(selectedVars)
```

```
 [1] "900 nm"  "914 nm"  "1208 nm" "1210 nm" "1220 nm" "1226 nm" "1230 nm"
 [8] "1232 nm" "1362 nm" "1364 nm" "1368 nm" "1636 nm" "1638 nm" "1640 nm"
[15] "1688 nm" "1692 nm" "1694 nm"
```

Part D

When comparing linear regression models from forward and lasso selection methods, we can see the differences in their complexity and how they value predictors. The forward selection model is simpler, using fewer but statistically significant variables to predict octane, suggesting a more effective and straightforward approach. In contrast, the lasso model uses more predictors but fails to show their significant impact at the standard level, raising concerns about overfitting and its real-world usefulness.

```
# Forward Selection Model
modelForward <- lm(octane ~ `1208 nm` + `1196 nm` + `976 nm` +
                     `1692 nm` + `970 nm` + `1206 nm` + `1056 nm` +
                     `1074 nm` + `1098 nm`, data = ds)
summary(modelForward)
```

```
Call:
lm(formula = octane ~ `1208 nm` + `1196 nm` + `976 nm` + `1692 nm` +
    `970 nm` + `1206 nm` + `1056 nm` + `1074 nm` + `1098 nm`,
    data = ds)

Residuals:
     Min       1Q   Median       3Q      Max
-0.43082 -0.09774  0.00706  0.12572  0.35202

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  100.8111     1.4668  68.726  < 2e-16 ***
`1208 nm`     37.6893    39.1939   0.962 0.340872
`1196 nm`     47.2937     3.3268  14.216  < 2e-16 ***
`976 nm`     230.0237   105.6168   2.178 0.034157 *
`1692 nm`     -3.1588     0.8184  -3.860 0.000326 ***
`970 nm`    -293.9540   124.9296  -2.353 0.022600 *
`1206 nm`   -139.8200    39.3453  -3.554 0.000840 ***
`1056 nm`    194.9291    55.3245   3.523 0.000921 ***
`1074 nm`   -268.4304    83.7947  -3.203 0.002365 **
`1098 nm`    176.9943    76.1575   2.324 0.024226 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1745 on 50 degrees of freedom
Multiple R-squared:  0.989, Adjusted R-squared:  0.987
F-statistic: 498.4 on 9 and 50 DF,  p-value: < 2.2e-16
```

```r
# Lasso Selection Model
modelLasso <- lm(octane ~ `900 nm` + `914 nm` + `1208 nm` + `1210 nm` +
                   `1220 nm` + `1226 nm` + `1230 nm` + `1232 nm` +
                   `1362 nm` + `1364 nm` + `1368 nm` + `1636 nm` +
                   `1638 nm` + `1640 nm` + `1688 nm` + `1692 nm` +
                   `1694 nm`, data = ds)
summary(modelLasso)
```

```
Call:
lm(formula = octane ~ `900 nm` + `914 nm` + `1208 nm` + `1210 nm` +
    `1220 nm` + `1226 nm` + `1230 nm` + `1232 nm` + `1362 nm` +
    `1364 nm` + `1368 nm` + `1636 nm` + `1638 nm` + `1640 nm` +
    `1688 nm` + `1692 nm` + `1694 nm`, data = ds)

Residuals:
     Min       1Q   Median       3Q      Max
-0.34049 -0.10527 -0.02336  0.12527  0.34352

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  100.6436     5.3982  18.644   <2e-16 ***
`900 nm`     -22.6936    47.6504  -0.476    0.636
`914 nm`      46.2276    45.1587   1.024    0.312
`1208 nm`   -100.1181    78.5195  -1.275    0.209
`1210 nm`     79.5443    90.8384   0.876    0.386
`1220 nm`    -52.9895    74.5246  -0.711    0.481
`1226 nm`     12.9481    71.3293   0.182    0.857
`1230 nm`   -171.9644   105.5422  -1.629    0.111
`1232 nm`    141.6543    98.1367   1.443    0.156
`1362 nm`     23.3611    95.8367   0.244    0.809
`1364 nm`     29.5509    86.0452   0.343    0.733
`1368 nm`      9.7858    71.4332   0.137    0.892
`1636 nm`      6.4870    29.9888   0.216    0.830
`1638 nm`     -2.1613    39.6103  -0.055    0.957
`1640 nm`    -15.2877    29.1802  -0.524    0.603
`1688 nm`     -0.6364     2.2686  -0.281    0.780
`1692 nm`     -1.4102     1.1816  -1.193    0.239
`1694 nm`     -0.4658     1.3421  -0.347    0.730
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1904 on 42 degrees of freedom
Multiple R-squared:  0.989, Adjusted R-squared:  0.9845
```

```
F-statistic: 221.8 on 17 and 42 DF,  p-value: < 2.2e-16
```

Problem 3

Part A

First PC explains $\approx 90\%$ of the variance

```
data(kanga)
kangaClean <- na.omit(kanga)
numericCols <- sapply(kangaClean, is.numeric)
ds <- kangaClean[, numericCols]
# Perform PCA on numeric dataset
pcaResult <- prcomp(ds, center = TRUE, scale = FALSE)
# Calculate explained variance of first principal component
varExplained <- summary(pcaResult)$importance[2, 1] * 100
varExplained
```

```
[1] 90.026
```

Part B

After setting an absolute loading threshold of 0.25, the prominent variables for the first principal component are `basilar.length`, `occipitonasal.length`, `palate.length`, and `mandible.length`.

```
# Loadings for the first principal component
loadingsOne <- pcaResult$rotation[, 1]
loadingsOne
```

```
    basilar.length occipitonasal.length        palate.length
       0.484068280          0.456296136          0.366214204
      palate.width         nasal.length          nasal.width
       0.084357568          0.248098913          0.074647492
    squamosal.depth       lacrymal.width      zygomatic.width
       0.063661339          0.118936859          0.206697648
     orbital.width       .rostral.width      occipital.depth
       0.014288869          0.106458931          0.178177037
       crest.width      foramina.length      mandible.length
      -0.081968043          0.009941189          0.435981938
     mandible.width       mandible.depth         ramus.height
       0.029996792          0.058322138          0.209071437
```

```
prominentVars <- names(loadingsOne[abs(loadingsOne) > 0.25])
prominentVars
```

```
[1] "basilar.length"       "occipitonasal.length" "palate.length"
[4] "mandible.length"
```

Part C

After running PCA with scaling, our new key variables influencing the first principal component include:`basilar.length`, `occipitonasal.length`, `squamosal.depth`, `lacrymal.width`, `orbital.width`, `rostral.width`, `foramina.length`, `mandible.depth`, and `ramus.height`. Again, I used a threshold of 0.25 to select these variables, indicating their significant contribution to the principal component analysis.

```
# Perform PCA on the cleaned dataset with scaling
pcaScaled <- prcomp(ds[,-1], center = TRUE, scale. = TRUE)
varExplained <- summary(pcaScaled)$importance[2, 1] * 100
varExplained
```

```
[1] 67.794
```

```
loadingsOneScaled <- pcaScaled$rotation[, 1]
loadingsOneScaled
```

```
occipitonasal.length        palate.length         palate.width
           0.27676284           0.28569501           0.24488311
         nasal.length          nasal.width      squamosal.depth
           0.23907455           0.23732429           0.24906415
       lacrymal.width      zygomatic.width        orbital.width
           0.27762419           0.26847879           0.07736949
       .rostral.width      occipital.depth         crest.width
           0.26831077           0.28226176          -0.17081968
      foramina.length      mandible.length       mandible.width
           0.06221900           0.28950099           0.21806709
       mandible.depth         ramus.height
           0.24903967           0.27029740
```

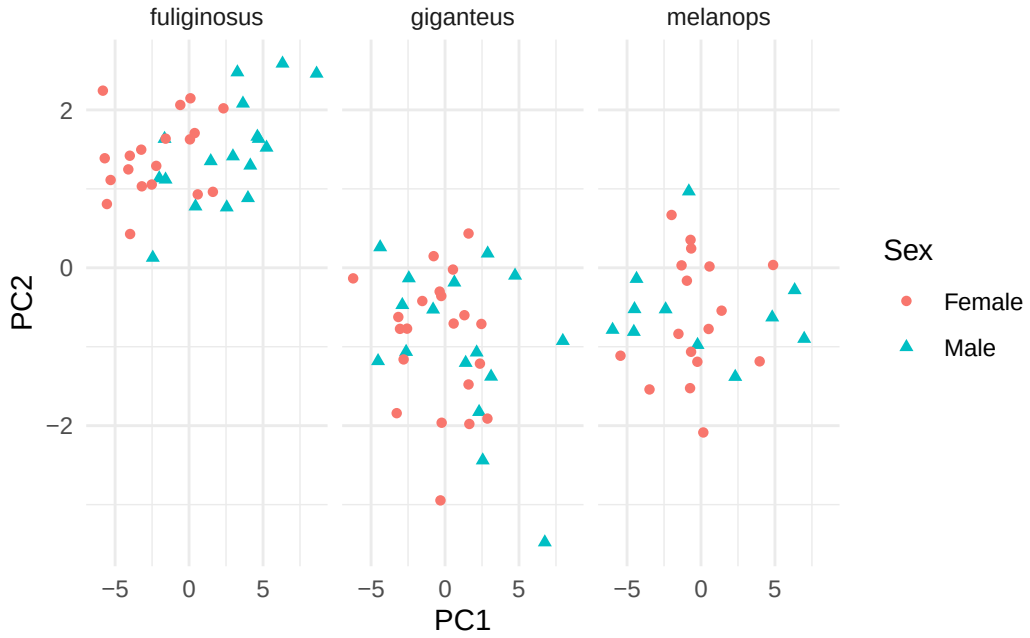
```
prominentVars <- names(loadingsOne[abs(loadingsOneScaled) > 0.25])
prominentVars
```

```
[1] "basilar.length"       "occipitonasal.length" "squamosal.depth"
[4] "lacrymal.width"       "orbital.width"        ".rostral.width"
[7] "foramina.length"      "mandible.depth"       "ramus.height"
```

Part F

In the scatterplots of the three species, PCA 1 vs PCA 2 effectively separates male and female specimens for fuliginosus. However, for giganteus, additional PCA components are needed to clearly distinguish between sexes. Melanops shows partial separation, but adding more components could improve sex determination clarity.

```
library(ggplot2)
pcaDs <- data.frame(PC1 = pcaScaled$x[,1], PC2 = pcaScaled$x[,2],
                    Sex = kangaClean$sex,
                    Species = kangaClean$species)
ggplot(pcaDs, aes(x = PC1, y = PC2, color = Sex, shape = Sex)) +
  geom_point() +
  facet_wrap(~ Species) +
  theme_minimal() +
  labs(shape = "Sex")
```




```
unique(kangaClean$species)
```

```
[1] giganteus   melanops    fuliginosus
Levels: fuliginosus giganteus melanops
```

Problem 4

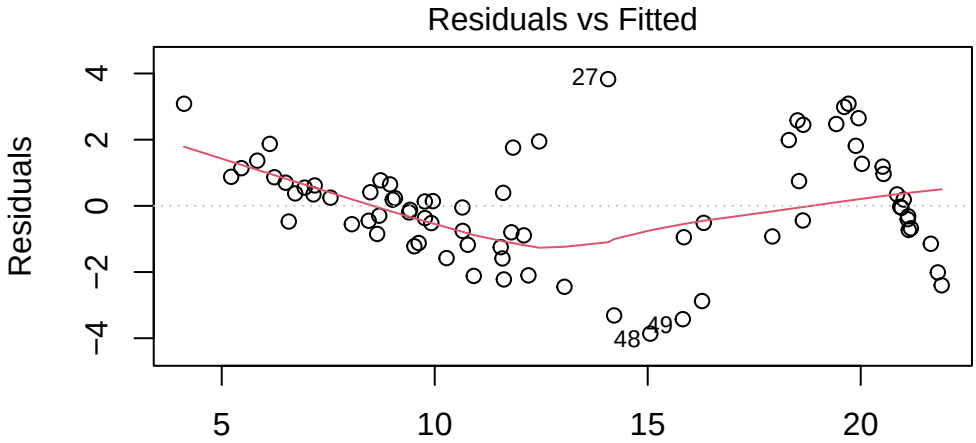
Part A

Given the presence of heteroscedasticity in the residuals and high autocorrelation observed in the ACF plot, the linear regression assumptions are violated, potentially leading to biased parameter estimates and unreliable inference with our highly significant estimators.

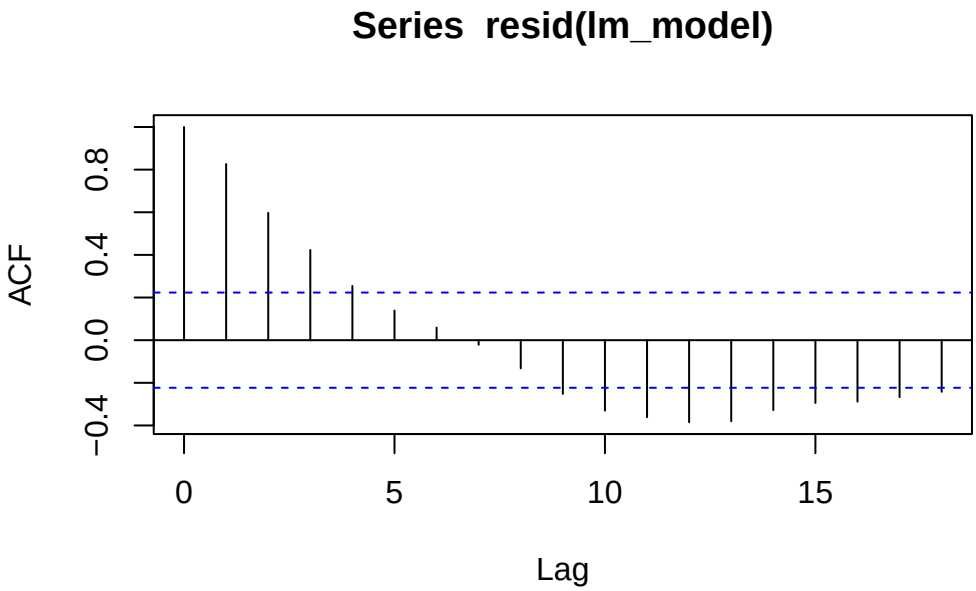
```
data(divusa)
lm_model <- lm(divorce ~ unemployed + femlab + marriage + birth + military,
               data = divusa)
summary(lm_model)
```

```
Call:
lm(formula = divorce ~ unemployed + femlab + marriage + birth +
    military, data = divusa)

Residuals:
    Min      1Q  Median      3Q     Max
-3.8611 -0.8916 -0.0496  0.8650  3.8300

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.48784    3.39378   0.733   0.4659
unemployed  -0.11125    0.05592  -1.989   0.0505 .
femlab       0.38365    0.03059  12.543  < 2e-16 ***
marriage     0.11867    0.02441   4.861 6.77e-06 ***
birth       -0.12996    0.01560  -8.333 4.03e-12 ***
military    -0.02673    0.01425  -1.876   0.0647 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.65 on 71 degrees of freedom
Multiple R-squared:  0.9208,    Adjusted R-squared:  0.9152
F-statistic: 165.1 on 5 and 71 DF,  p-value: < 2.2e-16
```

```
# Residual Plot
plot(lm_model, which = 1)
```




```r
# ACF Plot
acf(resid(lm_model))
```



Part B

In contrast to the lm model, GLS considers both heteroscedasticity and autocorrelation, resulting in more reliable parameter estimates and valid inference. In our case, the GLS model objectively outperforms the lm model by addressing observed issues and strengthening the significance of our findings.

```
library(nlme)
gls_model <- gls(divorce ~ unemployed + femlab + marriage + birth + military,
                 data = divusa, correlation = corAR1(form = ~ 1), method = "ML")
summary(gls_model)
```

```
Generalized least squares fit by maximum likelihood
  Model: divorce ~ unemployed + femlab + marriage + birth + military
  Data: divusa
       AIC      BIC    logLik
  179.9523 198.7027 -81.97613

Correlation Structure: AR(1)
 Formula: ~1
 Parameter estimate(s):
      Phi
0.9715486

Coefficients:
                Value Std.Error   t-value p-value
(Intercept) -7.059682  5.547193 -1.272658  0.2073
unemployed   0.107643  0.045915  2.344395  0.0219
femlab       0.312085  0.095151  3.279878  0.0016
marriage     0.164326  0.022897  7.176766  0.0000
birth       -0.049909  0.022012 -2.267345  0.0264
military     0.017946  0.014271  1.257544  0.2127

 Correlation:
           (Intr) unmply femlab marrig birth
unemployed -0.420
femlab     -0.802  0.240
marriage   -0.516  0.607  0.307
birth      -0.379  0.041  0.066 -0.094
military   -0.036  0.436 -0.311  0.530  0.128

Standardized residuals:
       Min         Q1        Med         Q3        Max
-1.4509327 -0.9760939 -0.6164694  1.1375377  2.1593261

Residual standard error: 2.907664
```

```
Degrees of freedom: 77 total; 71 residual
```

Part C

Given the presence of a year column in the dataset, it suggests a time series structure. The correlation observed in the errors could stem from various factors such as seasonal patterns, trends, or other time-dependent phenomena not fully captured by the predictors. Additionally, external factors that evolve over time might also contribute to the correlation in the errors.