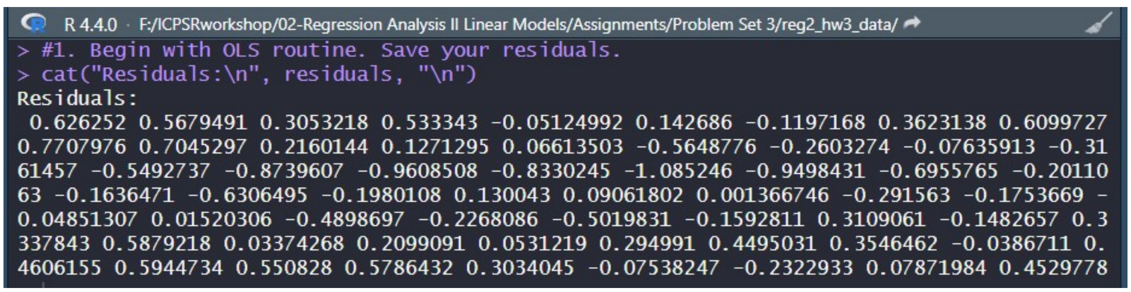
ICPSR Regression II - Problem Set 3

Ziyuan Gao

August 2024

1 Autocorrelation

1.1



```
R 4.4.0 · F:/ICPSRworkshop/02-Regression Analysis II Linear Models/Assignments/Problem Set 3/reg2_hw3_data/
> #1. Begin with OLS routine. Save your residuals.
> cat("Residuals:\n", residuals, "\n")
Residuals:
 0.626252 0.5679491 0.3053218 0.533343 -0.05124992 0.142686 -0.1197168 0.3623138 0.6099727
0.7707976 0.7045297 0.2160144 0.1271295 0.06613503 -0.5648776 -0.2603274 -0.07635913 -0.31
61457 -0.5492737 -0.8739607 -0.9608508 -0.8330245 -1.085246 -0.9498431 -0.6955765 -0.20110
63 -0.1636471 -0.6306495 -0.1980108 0.130043 0.09061802 0.001366746 -0.291563 -0.1753669 -
0.04851307 0.01520306 -0.4898697 -0.2268086 -0.5019831 -0.1592811 0.3109061 -0.1482657 0.3
337843 0.5879218 0.03374268 0.2099091 0.0531219 0.294991 0.4495031 0.3546462 -0.0386711 0.
4606155 0.5944734 0.550828 0.5786432 0.3034045 -0.07538247 -0.2322933 0.07871984 0.4529778
```

Figure 1: Residuals from OLS routine

1.2

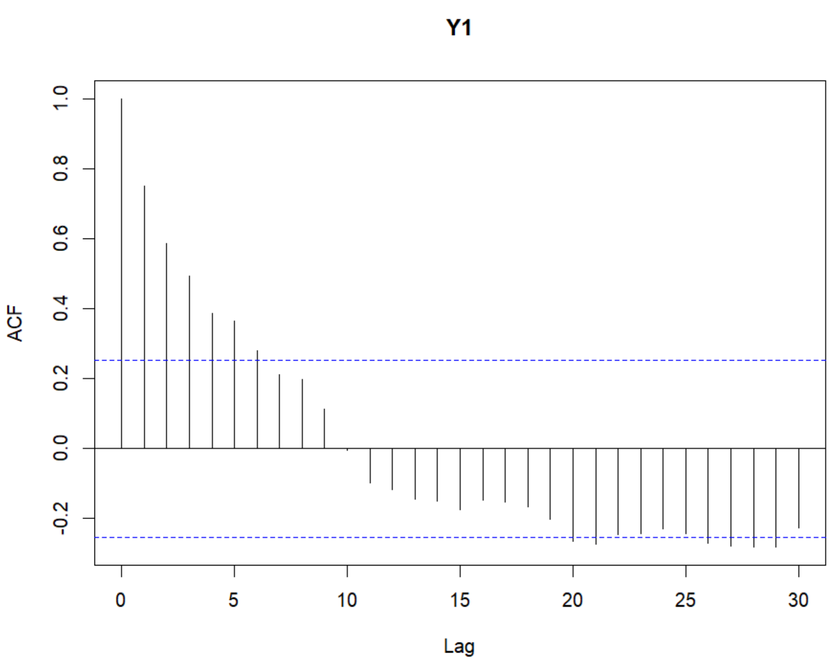


Figure 2: ACF correlogram

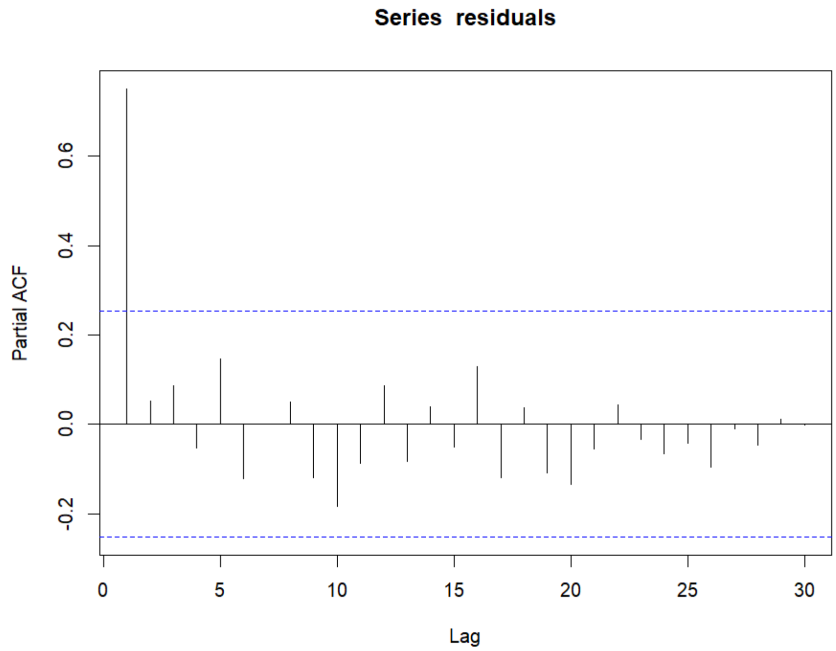


Figure 3: PACF correlogram

1.3

```
        Box-Ljung test

data:  residuals
X-squared = 104.67, df = 10, p-value < 2.2e-16
```

Figure 4: Box-Ljung test

The null hypothesis of the Box-Ljung test is that there is no autocorrelation up to lag k in the residals. Given the p-value is 2.2e-16, it indicates to reject the null hypothesis and the strong evidence of autocorrelation.

1.4

```r
########################## Transforming ##########################
#4. Transforming the data with AR1 error process
lagged_X2 <- c(NA, diff(data$X2))
data$X2tr <- data$X2 - lagged_X2
data$X2tr[is.na(data$X2tr)] <- data$X2tr[2]

x <- as.matrix(data[, c('X2tr', 'X3', 'X4')])


df <- nrow(t) - ncol(x) - 1
t <- as.matrix(data['Y1'])
t

transformed_SS_tot <- calculate_SS_tot(t)
transformed_df <- nrow(t) - ncol(x) - 1
transformed_beta_hat <- calculate_beta_hat(t, x)
transformed_residuals <- calculate_residual(t,x,transformed_beta_hat)
transformed_SSR <- calculate_SSR(transformed_residuals)
transformed_SEE <- calculate_SEE(transformed_SSR,transformed_df)
transformed_var_b <- calculate_var_b(transformed_SSR,x)
transformed_t_beta <- calculate_t_beta(transformed_beta_hat,transformed_var_b)
transformed_r2 <- calculate_r2(transformed_SSR,transformed_SS_tot)
transformed_adjusted_r2 <- calculate_adjusted_r2(transformed_r2, nrow(t), ncol(x))
```

Figure 5: data transformation in AR1 error process

1.5

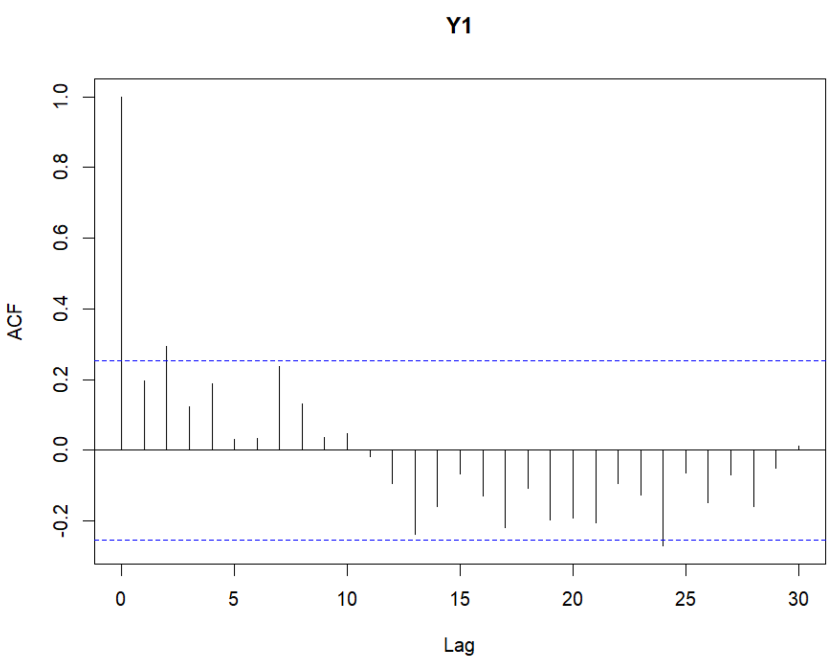


Figure 6: transformed ACF correlogram

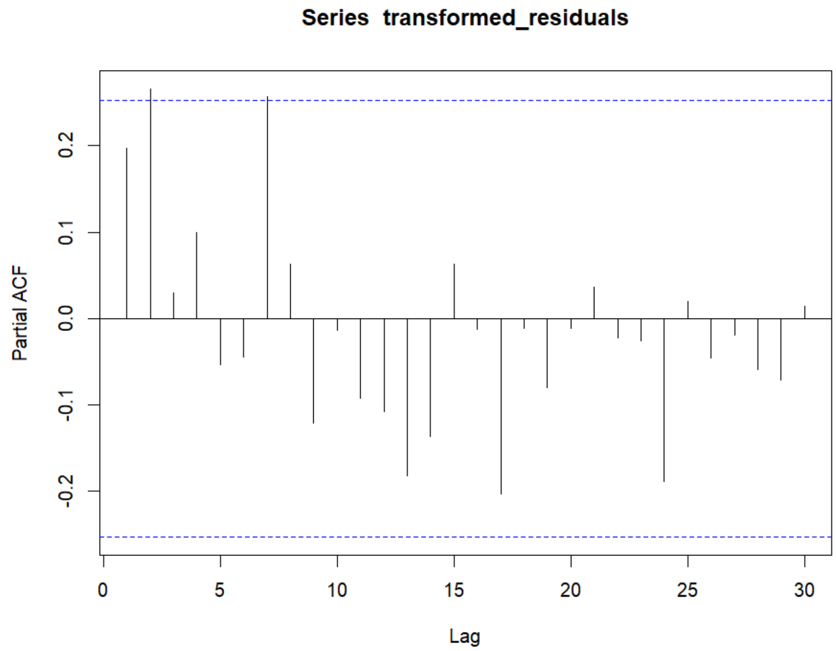


Figure 7: transformed PACF correlogram

```
        Box-Ljung test

data:  transformed_residuals
X-squared = 16.86, df = 10, p-value = 0.07752
```

Figure 8: transformed Box-Ljung test

From transformed ACF and PACF correlogram, it can be seen that auto correlation has been handled within the significance threshold. The transformed p-value 0.07752 indicates that I failed to reject the null hypothesis, that is there is no autocorrelation after data transformation.

2 Heteroskedasticity

2.1

```
R 4.4.0 · F:/ICPSRworkshop/02-Regression Analysis II Linear Models/Assignments/Problem Set 3/reg2_hw3_data/
> #1. Begin with OLS routine. Save your residuals.
> cat("Residuals:\n", residuals, "\n")
Residuals:
 0.626252 0.5679491 0.3053218 0.533343 -0.05124992 0.142686 -0.1197168 0.3623138 0.6099727
0.7707976 0.7045297 0.2160144 0.1271295 0.06613503 -0.5648776 -0.2603274 -0.07635913 -0.31
61457 -0.5492737 -0.8739607 -0.9608508 -0.8330245 -1.085246 -0.9498431 -0.6955765 -0.20110
63 -0.1636471 -0.6306495 -0.1980108 0.130043 0.09061802 0.001366746 -0.291563 -0.1753669 -
0.04851307 0.01520306 -0.4898697 -0.2268086 -0.5019831 -0.1592811 0.3109061 -0.1482657 0.3
337843 0.5879218 0.03374268 0.2099091 0.0531219 0.294991 0.4495031 0.3546462 -0.0386711 0.
4606155 0.5944734 0.550828 0.5786432 0.3034045 -0.07538247 -0.2322933 0.07871984 0.4529778
```

Figure 9: Residuals from OLS routine

2.2

```
> bptest(model_heter)

        studentized Breusch-Pagan test

data:  model_heter
BP = 13.215, df = 3, p-value = 0.004194
```

Figure 10: Breusch-Pagan Test result

Clearly the p-value (0.004194) from the Breusch-Pagan test is far less than 0.05, I reject the null hypothesis, indicating heteroskedasticity.

```
R 4.4.0 · F:/ICPSRworkshop/02-Regression Analysis II Linear Models/Assignments/Problem Set 3/reg2_hw3
> aux_model_X2 <- lm(residuals_sq ~ X2, data = data_heter)
> summary(aux_model_X2)

Call:
lm(formula = residuals_sq ~ X2, data = data_heter)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.6915 -0.3240 -0.1085  0.3266  1.3576 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -0.4063     0.2617  -1.553 0.125949    
X2            0.3865     0.1097   3.525 0.000834 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.434 on 58 degrees of freedom
Multiple R-squared:  0.1764,    Adjusted R-squared:  0.1622 
F-statistic: 12.43 on 1 and 58 DF,  p-value: 0.0008341
```

Figure 11: glejser test-X2

```
R 4.4.0 · F:/ICPSRworkshop/02-Regression Analysis II Linear Models/Assignments/Problem Set 3/reg2_hw3
> aux_model_X3 <- lm(residuals_sq ~ X3, data = data_heter)
> summary(aux_model_X3)

Call:
lm(formula = residuals_sq ~ X3, data = data_heter)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.5770 -0.3661 -0.1158  0.2424  1.3446 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)   
(Intercept)   0.9385     0.3260   2.879  0.00558 **
X3           -0.1911     0.1379  -1.385  0.17121   
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4705 on 58 degrees of freedom
Multiple R-squared:  0.03204,	Adjusted R-squared:  0.01535 
F-statistic:  1.92 on 1 and 58 DF,  p-value: 0.1712
```

Figure 12: glejser test-X3

```
R 4.4.0 · F:/ICPSRworkshop/02-Regression Analysis II Linear Models/Assignments/Problem Set 3/reg2_hw3
> aux_model_X4 <- lm(residuals_sq ~ X4, data = data_heter)
> summary(aux_model_X4)

Call:
lm(formula = residuals_sq ~ X4, data = data_heter)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.5282 -0.3917 -0.1136  0.3131  1.4600 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept)   0.7362     0.3611   2.039    0.046 *
X4           -0.1004     0.1480  -0.679    0.500  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4764 on 58 degrees of freedom
Multiple R-squared:  0.007875,	Adjusted R-squared:  -0.00923 
F-statistic: 0.4604 on 1 and 58 DF,  p-value: 0.5001
```

Figure 13: glejser test-X4

I used the Glesjer test to test each predictor individually. Clearly, for X2, a p-value of 0.0008341 strongly indicates heteroskedasticity. While for X3 (p-value 0.1712) and X4 (0.5001) not indicating heteroskedasticity. In conclusion, X2 is the offending variable.

2.3

```r
# Transform the predictor matrix
# The addition of 1 is a key part of stabilizing the variance of the residuals.
data_heter$Y4_transformed <- residuals(model_heter) / sqrt(1 + residuals_sq)
data_heter$X2_transformed <- data_heter$X2 / sqrt(1 + residuals_sq)
data_heter$X3_transformed <- data_heter$X3 / sqrt(1 + residuals_sq)
data_heter$X4_transformed <- data_heter$X4 / sqrt(1 + residuals_sq)

library(nlme)
# Fit GLS model with a variance structure
gls_model <- gls(Y4_transformed ~ X2_transformed + X3_transformed + X4_transformed,
                 data = data_heter,
                 weights = varPower(form = ~ X2_transformed))  # or other appropriate variance
summary(gls_model)
# Check residuals of the GLS model
gls_residuals <- residuals(gls_model)
plot(gls_residuals, main = "Residuals of GLS Model")
acf(gls_residuals, main = "ACF of Residuals")
# Compare GLS and OLS results
summary(model_heter)  # OLS results
summary(gls_model)    # GLS results
```

Figure 14: transformation and re-estimation code

```
R 4.4.0 · F:/ICPSRworkshop/02-Regression Analysis II Linear Models/Assignments/Problem Set 3/reg2_
> # Compare GLS and OLS results
> summary(model_heter)  # OLS results

Call:
lm(formula = Y4 ~ X2 + X3 + X4, data = data_heter)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.39672 -0.48942  0.08721  0.62815  1.03125 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   2.5734     0.7875   3.268  0.00185 ** 
X2            2.1051     0.1851  11.375 3.49e-16 ***
X3            1.3620     0.2229   6.110 1.01e-07 ***
X4           -2.2116     0.2362  -9.365 4.67e-13 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7281 on 56 degrees of freedom
Multiple R-squared:  0.8209,	Adjusted R-squared:  0.8113 
F-statistic: 85.57 on 3 and 56 DF,  p-value: < 2.2e-16
```

Figure 15: OLS summary

```
R 4.4.0 · F:/ICPSRworkshop/02-Regression Analysis II Linear Models/Assignments/Problem Set 3/reg2_hw3_data/
> summary(gls_model)    # GLS results
Generalized least squares fit by REML
  Model: Y4_transformed ~ X2_transformed + X3_transformed + X4_transformed
  Data: data_heter
       AIC      BIC    logLik
  110.9424 123.0945 -49.47122

Variance function:
 Structure: Power of variance covariate
 Formula: ~X2_transformed
 Parameter estimates:
     power
-0.2999196

Coefficients:
                    Value Std.Error    t-value p-value
(Intercept)    -0.3490972 0.4015578 -0.8693571  0.3884
X2_transformed  0.1597106 0.1788021  0.8932252  0.3756
X3_transformed -0.0617760 0.1813431 -0.3406578  0.7346
X4_transformed  0.0876563 0.1798497  0.4873866  0.6279

 Correlation:
               (Intr) X2_trn X3_trn
X2_transformed -0.640
X3_transformed -0.073 -0.287
X4_transformed -0.367  0.009 -0.622

Standardized residuals:
        Min          Q1         Med          Q3         Max
-1.56762119 -0.87768709  0.05364205  0.95048461  1.40253507

Residual standard error: 0.6418487
Degrees of freedom: 60 total; 56 residual
>
```

Figure 16: GLS summary

```
R 4.4.0 · F:/ICPSRworkshop/02-Regression Analysis II Linear Models/Assignments/Problem Set 3/reg2_hw3
> summary(aux_model_X2)

Call:
lm(formula = residuals_gls ~ X2_transformed, data = data_heter)

Residuals:
     Min       1Q   Median       3Q      Max
-0.33667 -0.15515 -0.00459  0.15595  0.39843

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)     0.44548    0.11911   3.740 0.000424 ***
X2_transformed -0.09313    0.05995  -1.554 0.125723
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1912 on 58 degrees of freedom
Multiple R-squared:  0.03995,	Adjusted R-squared:  0.0234
F-statistic: 2.414 on 1 and 58 DF,  p-value: 0.1257
```

Figure 17: glejser test-X2 transformed

Please noted In data transformation I added 1 to the residuals squared, this is crucial step to ensure the denominator is never zero and always positive, preventing division by very small numbers that could lead to instability in the transformed values.

X2: a p-value of 0.12571 indicates heteroskedasticity has been solved.

Residual Standard Error: The GLS model shows a slightly lower residual standard error (0.6418) compared to the OLS model (0.7281), indicating a better fit after adjusting for heteroscedasticity.

AIC/BIC: The GLS model has lower AIC and BIC values compared to the OLS model. Lower AIC and BIC suggest that the GLS model perform better after accounting for heteroscedasticity. The transformation and the variance modeling applied in GLS appear to have addressed the heteroscedasticity issue,

3 Multicollinearity

```
############################################
##############  QUESTION 3  ###############
############################################

dataset_Q3 <- read.csv("Singular.csv")
head(dataset_Q3)
y <- as.matrix(dataset_Q3['y'])
x <- as.matrix(dataset_Q3[, c('X2','X3','X4','X5','X6','X7')])

#1. fit the model
model_full <- lm(y ~ X2 + X3 + X4 + X5 + X6 + X7, data = dataset_Q3)
summary(model_full)

#2. use the Variance Inflation Factor (VIF) to diagnose multicollinearity
vif_value <- vif(model_full)
vif_value

#Option 1 discard independent variables of multicollinearity
model_discard <- lm(y ~  X3 + X4 + X5, data = dataset_Q3)
summary(model_discard)
vif_value_discard <- vif(model_discard)
vif_value_discard

anova(model_discard, model_full)
AIC(model_discard, model_full)
BIC(model_discard, model_full)


#Option 2 PCA
# Perform PCA
pca <- prcomp(dataset_Q3[, c('X2', 'X3', 'X4', 'X5', 'X6', 'X7')], scale. = TRUE)
pca_scores <- pca$x

# Fit a model using the principal components
model_pca <- lm(y ~ pca_scores[, 1:3], data = data.frame(y = dataset_Q3$y, pca_scores))
summary(model_pca)
anova(model_pca, model_full)
AIC(model_pca, model_full)
BIC(model_pca, model_full)
```

Figure 18: Q3 code

Workflow

1. Fit the model.

2. Diagnose multicollinearity using Variance Inflation Factor (VIF). It can be seen there is strong evidence of multicollinearity issue in variable X2, X6, X7 as they have high vif value.

```
> vif_value
       X2        X3        X4        X5        X6        X7
13.193633  1.310131  1.044735  2.115655 22.477070 11.770261
```

Figure 19: VIF

3. To deal with collinear explanatory variables, I tried two options:
   - Option 1: Discard independent variables exhibiting multicollinearity.
   - Option 2: Apply Principal Component Analysis (PCA).
4. After transforming the data, I compared the results with the original model using ANOVA, AIC, and BIC to determine the best model with the best estimates.

Option 1: Discard independent variables exhibiting multicollinearity.

From ANOVA test, The null hypothesis for the F-test is that the new model improve the model fit. Howerver, the extremely small p-value 2.2e-16 leads us to reject this null hypothesis, indicating that discarding high multicollineared variables X1 X3 X4 performs worse on the model fit. Also, from AIC and BIC, the new model discarding high multicollineared variables has both a much higher AIC (860.5856) and a higher BIC (870.1458) compared to the original model with AIC (758.2161) and BIC(773.5123), indicating that the new model performed much worse then the original one. So discarding variables is not good operation.

```
> vif_value_discard
      X3       X4       X5
1.003330 1.003071 1.001454
> anova(model_discard, model_full)
Analysis of Variance Table

Model 1: y ~ X3 + X4 + X5
Model 2: y ~ X2 + X3 + X4 + X5 + X6 + X7
  Res.Df      RSS Df Sum of Sq      F    Pr(>F)
1     46 71546551
2     43  8190352  3  63356199 110.88 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> AIC(model_discard, model_full)
              df      AIC
model_discard  5 860.5856
model_full     8 758.2161
> BIC(model_discard, model_full)
              df      BIC
model_discard  5 870.1458
model_full     8 773.5123
```

Figure 20: ANOVA AIC BIC for op1

Option 2: Apply Principal Component Analysis (PCA)

From ANOVA test, The null hypothesis for the F-test is that the new model improves the model fit. After PCA, the p-value 0.3457 fail to reject this null hypothesis, indicating that PCA improve the model fit. From AIC and BIC, the new model after PCA has both a lower AIC (757.2622) and a lower

BIC (770.6464) compared to the original model with AIC (758.2161) and BIC(773.5123), indicating it performs better then the original model. Therefore, we pick PCA for data transformation leading to a good fit.

```
> anova(model_pca, model_full)
Analysis of Variance Table

Model 1: y ~ pca_scores[, 1:5]
Model 2: y ~ X2 + X3 + X4 + X5 + X6 + X7
  Res.Df     RSS Df Sum of Sq      F Pr(>F)
1     44 8363524
2     43 8190352  1    173172 0.9092 0.3457
> AIC(model_pca, model_full)
           df      AIC
model_pca   7 757.2622
model_full  8 758.2161
> BIC(model_pca, model_full)
           df      BIC
model_pca   7 770.6464
model_full  8 773.5123
```

Figure 21: ANOVA AIC BIC for op2

4 Model Specification

4.1

I build two model. Model 1 is Life expectancy - People/TV. Model 2 is Life expectancy - People/TV + People/ physician. Clearly Model 1 is nested in Model 2.

From ANOVA test, The null hypothesis for the F-test is that the nested model/Model 1 fits better then Model 2. The p-value 0.043 reject this null hypothesis, indicating that Model 2 performs better.

From AIC and BIC, Model 2 has both a lower AIC (260.1059) and a lower BIC (266.8614) compared to Model 1 (262.5792) and BIC(267.6459), indicating it performs better then the nested model. Therefore, we pick Model 2.

```

###########################################
##############  QUESTION 4  ###############
###########################################
library(dplyr)
library(tibble)
dataset_Q4 <- read.csv("TVlaVie.csv", header = FALSE, skip = 3)
dataset_Q4 <- dataset_Q4 %>% filter(rowSums(is.na(.))==0)
dataset_Q4
head(dataset_Q4)

dataset_Q4$Life.Expectancy
model1 <- lm(V2 ~ V3, data = dataset_Q4)
model2 <- lm(V2 ~ V3 + V4, data = dataset_Q4)
anova(model1, model2)
AIC(model1, model2)
BIC(model1, model2)

```

Figure 22: comparision of two nested models code

```
R 4.4.0 · F:/ICPSRworkshop/02-Regression Analysis II Linear Models/Assignments/Problem Set 3/reg2_hw3_data/
> anova(model1, model2)
Analysis of Variance Table

Model 1: V2 ~ V3
Model 2: V2 ~ V3 + V4
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     38 1430.1
2     37 1278.8  1    151.31 4.3781 0.04332 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
> AIC(model1, model2)
       df      AIC
model1  3 262.5792
model2  4 260.1059
> BIC(model1, model2)
       df      BIC
model1  3 267.6459
model2  4 266.8614
```

Figure 23: ANOVA AIC BIC

4.2

I build two model. Model A is y - X3 + X4. Model B is y - X3 + X5. Model C is y - X5 + X6 + X7. Clearly Model B should be “overlapping non-nested” with Model A, and Model C should be “strictly non-nested” with respect to Model A.

From AIC and BIC, Model C has both the lowest AIC (848.0703) and the lowest BIC (857.6305) compared to Model A and Model B, indicating it performs best as it achieves a better trade-off between fit and complexity.

```
############################################
#############  QUESTION 4-2  ##############
############################################
head(dataset_Q3)
modelA <- lm(y ~ X3 + X4, data = dataset_Q3)
modelB <- lm(y ~ X3 + X5, data = dataset_Q3)
modelC <- lm(y ~ X5 + X6 + X7, data = dataset_Q3)
# Calculate AIC
aicA <- AIC(modelA)
aicB <- AIC(modelB)
aicC <- AIC(modelC)

# Calculate BIC
bicA <- BIC(modelA)
bicB <- BIC(modelB)
bicC <- BIC(modelC)

# Print AIC and BIC
print(c(AIC_A = aicA, AIC_B = aicB, AIC_C = aicC))
print(c(BIC_A = bicA, BIC_B = bicB, BIC_C = bicC))
```

Figure 24: comparision of three non-nested models code

```
> print(c(AIC_A = aicA, AIC_B = aicB, AIC_C = aicC))
   AIC_A    AIC_B    AIC_C 
862.8973 875.4680 848.0703 
> print(c(BIC_A = bicA, BIC_B = bicB, BIC_C = bicC))
   BIC_A    BIC_B    BIC_C 
870.5454 883.1161 857.6305 
```

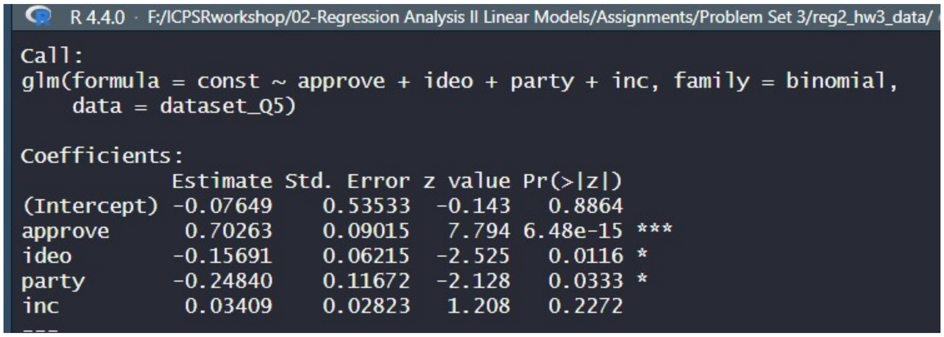
Figure 25: AIC BIC

5 Logistic Regression and GLM

```
############################################
##############  QUESTION 5  ###############
############################################
dataset_Q5 <- read.csv("eo_survey.csv")
head(dataset_Q5)

# transform party from categorical to numerical
dataset_Q5$party_new <- as.numeric(factor(dataset_Q5$party,
                                          levels = c("Democratic Party",
                                                     "Republican Party",
                                                     "No Party, Independent, Decline to state")))
dataset_Q5$party <- dataset_Q5$party_new
dataset_Q5 <- dataset_Q5 %>% select(-party_new)
head(dataset_Q5)

# build the logit model
model_logit <- glm(const ~ approve + ideo + party + inc, data = dataset_Q5, family = binomial)
summary(model_logit)

# Predicted probabilities for the first two observations
predicted_probability <- predict(model_logit, type = "response")[1:2]
predicted_probability

# MANUALLY calculate predicted probabilities
calculate_pred_probability <- function(approve,ideo,party,inc){
  X_beta <- -0.07649+(0.70263*approve)+(-0.15691*ideo)+(-0.24840*party)+(0.03409*inc)
  pred_probability <- exp(X_beta)/(1+exp(X_beta))
  return(pred_probability)
}

# MANUALLY calculate the first observations
pred_probability_first <- calculate_pred_probability(1,6,2,7)
pred_probability_first

# MANUALLY calculate the first observations
pred_probability_second <- calculate_pred_probability(3,2,1,11)
pred_probability_second

```

Figure 26: Q5 code

5.1



```
R 4.4.0 · F:/ICPSRworkshop/02-Regression Analysis II Linear Models/Assignments/Problem Set 3/reg2_hw3_data/
Call:
glm(formula = const ~ approve + ideo + party + inc, family = binomial,
    data = dataset_Q5)

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.07649    0.53533  -0.143   0.8864
approve      0.70263    0.09015   7.794 6.48e-15 ***
ideo        -0.15691    0.06215  -2.525   0.0116 *
party       -0.24840    0.11672  -2.128   0.0333 *
inc          0.03409    0.02823   1.208   0.2272
---
```

Figure 27: sign and significance of the variables

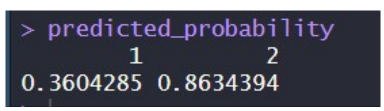
Obama approval (approve): The coefficient for Obama approval is positive and highly significant (estimate = 0.70263, p = 6.48e-15). This suggests that as respondents' approval of Obama increases, they are significantly more likely to believe that executive orders are constitutional.

Ideology (ideo): The ideology coefficient is negative and significant (estimate = -0.15691, p = 0.0116). This indicates that as respondents' political ideology becomes more conservative, the likelihood of believing that executive orders are constitutional decreases.

Party (party): The coefficient is negative and significant (estimate = -0.24840, p = 0.0333). This means that respondents identifying as Republican or Independent are significantly less likely to believe that executive orders are constitutional compared to Democrats.

Income (inc): The income coefficient is positive but not statistically significant (estimate = 0.03409, p = 0.2272). This indicates that income does not have a significant effect on the belief in the constitutionality .

5.2



```
> predicted_probability
        1         2
0.3604285 0.8634394
```

Figure 28: predicted probability for the first two observations

5.3

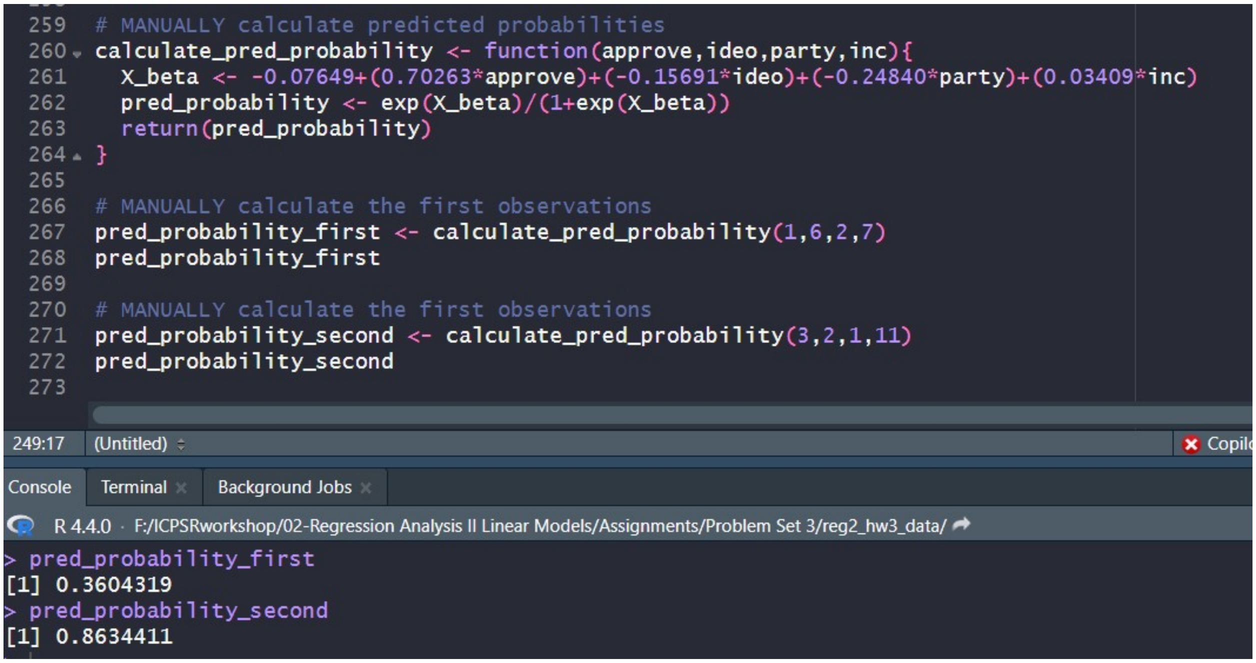


Figure 29: hand-calculate for the predicted probablity

Question 5 Bonus

I tried my best but the plot looks very weird... I attached my code and plots below and do appreciate if I could have any feedback on it!

```r
# Predicted Probability Plot
library(ggplot2)
library(tidyverse)
library(dplyr)

dataset_Q5$pred_prob <- predict(model_logit, dataset_Q5, type = "response")

dataset_Q5_plot <- dataset_Q5 %>%
  mutate(predlow = plogis(const - (1.96 * pred_prob)),
         pred = plogis(const),
         predhigh = plogis(const + (1.96 * pred_prob)))

dataset_Q5_plot

#####
# Predicted probability plot in ggplot2.
dataset_Q5_plot %>%
  ggplot(aes(x =  ideo, y = pred_prob)) +
  geom_ribbon(aes(ymin = predlow,
                  ymax = predhigh,
                  fill =  const), alpha = 0.3) +
  geom_line(aes(colour = const), size = 1) +
  labs(y = "Predicted Probability",
       x = "Ideology",
       title = "Predicted Probability by Ideology",
       subtitle = "Holding other variables constant",
       color = "Constitutionality",
       fill = "Constitutionality") +
  scale_color_manual(values = c("#3368d8", "#33a2d8"),
                     labels = c("Unconstitutional", "Constitutional")) +
  scale_fill_manual(values = c("#3368d8", "#33a2d8"),
                    labels = c("Unconstitutional", "Constitutional")) +
  guides(color = guide_legend(reverse = TRUE)) +
  guides(fill = guide_legend(reverse = TRUE)) +
  theme_minimal()

```

Figure 30: Buggy code. I do appreciate if I could have any feedback on it!

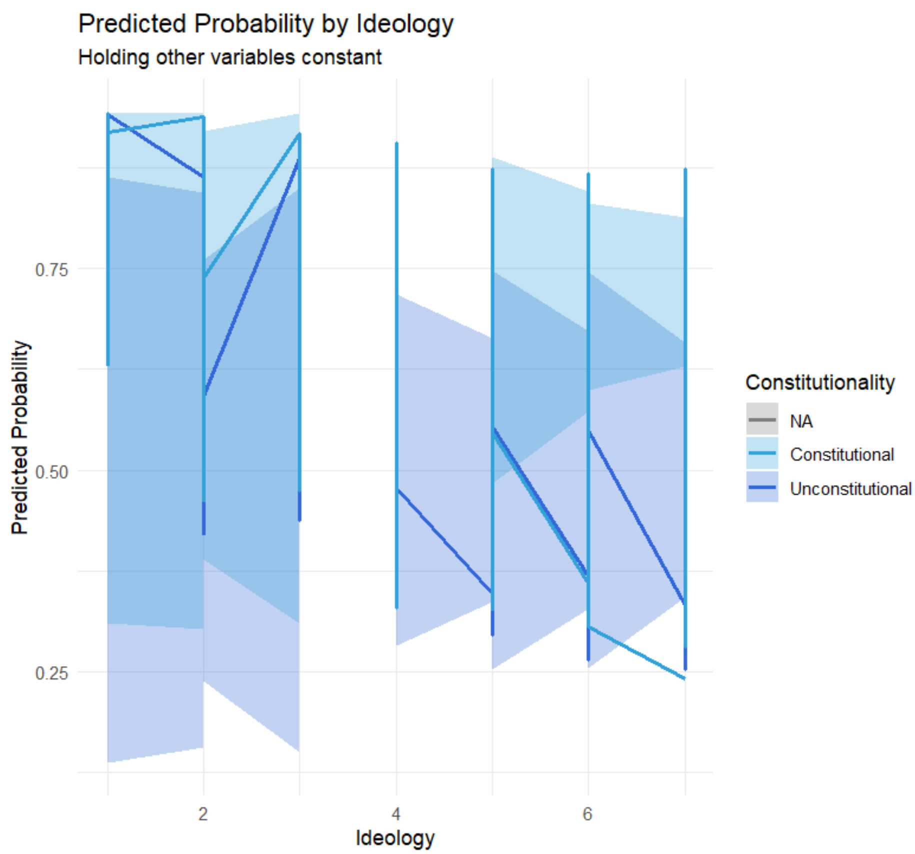


Figure 31: Buggy plot. I tried my best but the plot looks very weird... I do appreciate if I could have any feedback on it!