

Regression Model Comparison

Alejandro Arellano

2023-03-28

General Linear Model 1.2

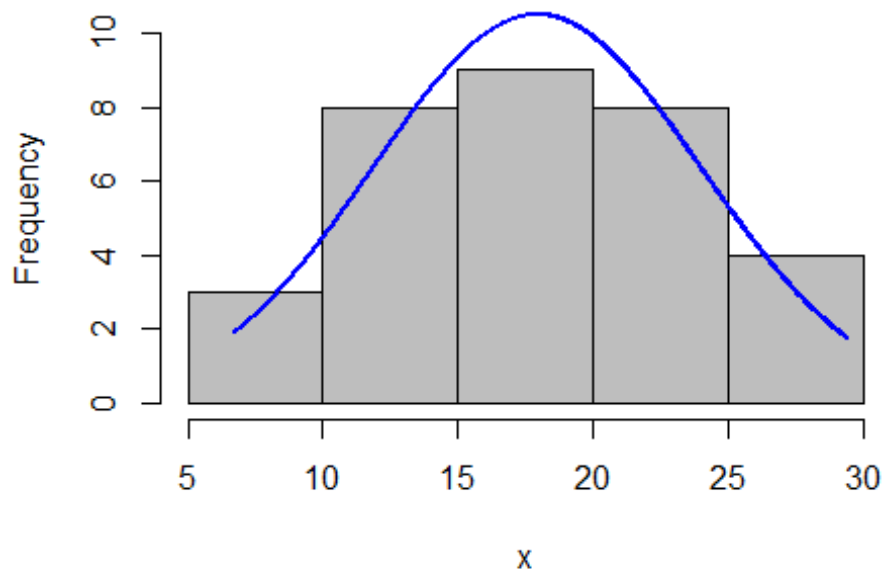
(a) Verify normality of the response variable, then fit the Linear regression model to the data. State the fitted model. Give estimates for all parameters.

#Importing data for 1.2

```
WeightLoss<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/  
Exercise Files/Exercise1.2Data.csv", header = TRUE, sep =  
",")
```

#Checking for normality

```
plotNormalHistogram(WeightLoss$EWL)
```



```
shapiro.test(WeightLoss$EWL)
```

```
##
```

```
## Shapiro-Wilk normality test
```

```
##
## data: WeightLoss$EWL
## W = 0.97424, p-value = 0.6234

#Reference Levels
drug.rel<- relevel(as.factor(WeightLoss$drug), ref="A")
gender.rel<- relevel(as.factor(WeightLoss$gender), ref="M")

#Creating general linear model
summary(fitted.model<- glm(EWL ~ drug.rel + age + gender.rel, data =
WeightLoss, family=gaussian(link=identity)))

##
## Call:
## glm(formula = EWL ~ drug.rel + age + gender.rel, family = gaussian(link =
identity),
##      data = WeightLoss)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -12.6861  -2.8772   0.0282   3.0932  10.2343
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.2146     5.6981   1.617   0.1171
## drug.relB     4.8103     1.9988   2.407   0.0229 *
## age           0.1102     0.1140   0.967   0.3420
## gender.relF   2.7235     1.9952   1.365   0.1831
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 31.44133)
##
##      Null deviance: 1144.12  on 31  degrees of freedom
## Residual deviance:  880.36  on 28  degrees of freedom
## AIC: 206.88
##
## Number of Fisher Scoring iterations: 2

#To show estimated sigma
sigma(fitted.model)

## [1] 5.607257
```

(b) Which regression coefficients turn out to be significant at the 5%? Discuss goodness of fit of the #model.

Drug B is the only significant predictor in the model at the 5% significance level since the corresponding p-value is the only one under 0.05.

```
#Checking to see if the model fits using p-value and deviance test
null.model<- glm(EWL ~ 1, data=WeightLoss, family=gaussian(link=identity))
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 8.386158 (df=2)

print(p.value<- pchisq(deviance, df=3, lower.tail=FALSE))

## 'log Lik.' 0.03867005 (df=2)
```

- (c) Is one of the drugs more efficient for weight loss than the other? Interpret all estimated significant coefficients.

The estimated average EWL for subjects taking drug B is 4.8103 percent higher than that for subjects taking drug A, keeping all the other predictors fixed. It means that drug B is more efficient than drug A.

- (d) According to the model, what is the predicted percent decrease in excess body weight for a 35-year old male who is taking drug A?

The predicted percent decrease in excess body weight for a 35-year old male who is taking drug A is computed by hand as: $EWL_{\hat{A}} = 9.2146 + 0.1102 \cdot 35 = 13.0716$.

```
#Finding the predicted values using the model
print(predict(fitted.model, data.frame(drug.rel="A", age=35, gender.rel="M")))

##          1
## 13.07178
```

General Linear Model 1.3

(a) Verify normality of the response variable, then fit the linear regression model to the data. State the fitted model. Give estimates for all parameters.

#Importing data for 1.3

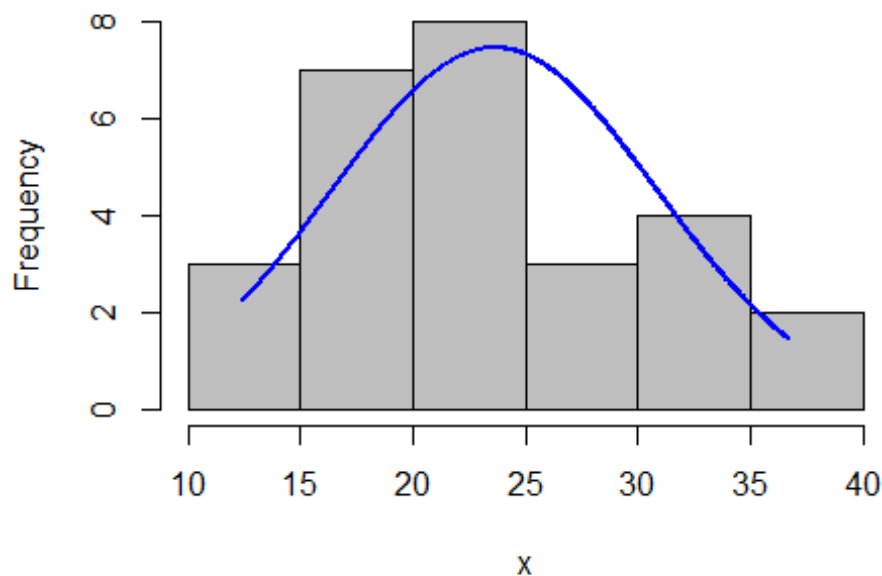
```
CarSales<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/Exercise Files/Exercise1.3Data.csv", header = TRUE, sep = ",")
```

#Rescaling price

```
priceK<- CarSales$price/1000
```

#running normality check

```
plotNormalHistogram(priceK)
```



#Running Shapiro test

```
shapiro.test(priceK)
```

```
##
```

```
## Shapiro-Wilk normality test
```

```
##
```

```
## data: priceK
```

```
## W = 0.95482, p-value = 0.28
```

#Creating reference Levels

```
bodystyle.rel<- relevel(as.factor(CarSales$bodystyle), ref="hatchback")
```

```
country.rel<- relevel(as.factor(CarSales$country), ref="Japan")
```

```
leather.rel<- relevel(as.factor(CarSales$leather), ref="no")
```

#Creating general linear model

```
summary(fitted.model<- glm(priceK ~ bodystyle.rel + country.rel + hwy + doors +  
+ leather.rel, data=CarSales, family=gaussian(link=identity)))
```

```
##
```

```
## Call:
```

```
## glm(formula = priceK ~ bodystyle.rel + country.rel + hwy + doors +
```

```
## leather.rel, family = gaussian(link = identity), data = CarSales)
```

```
##
```

```
## Deviance Residuals:
```

```
##      Min       1Q   Median       3Q      Max
```

```
## -4.4438 -2.4342  0.8821  1.6428  7.2350
```

```
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    5.1353     5.5909   0.919  0.36986
## bodystyle.relcoupe  2.2698     2.0070   1.131  0.27216
## bodystyle.relsedan  6.4107     1.8450   3.475  0.00254 **
## country.relGermany  3.1959     2.0098   1.590  0.12829
## country.relUSA      3.2128     1.8812   1.708  0.10394
## hwy               0.1305     0.1332   0.980  0.33937
## doors             1.5554     0.7904   1.968  0.06384 .
## leather.relyes     12.1757     1.9332   6.298 4.79e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 12.1319)
##
##      Null deviance: 1357.08  on 26  degrees of freedom
## Residual deviance:  230.51  on 19  degrees of freedom
## AIC: 152.52
##
## Number of Fisher Scoring iterations: 2

#To show estimated sigma
sigma(fitted.model)

## [1] 3.483088
```

(b) How good is the model fit? Discuss significance of the regression coefficients.

The p-value in the deviance test is way below 0.05, indicating a good model fit. #Significant variables are sedan body style and leather interior.

```
#Checking to see if the model fits using p-value and deviance test
null.model<- glm(priceK ~ 1, data=CarSales, family=gaussian(link=identity))
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 47.86586 (df=2)

print(p.value<- pchisq(deviance, df=7, lower.tail = FALSE))

## 'log Lik.' 3.78218e-08 (df=2)
```

(c) Interpret the estimates of those regression coefficients that differ significantly from zero.

As estimated, sedan costs on average \$6,410.70 more than a hatchback, under all other equal conditions. The estimated average price of a car with leather interior is \$12,175.70 larger compared to a car without leather interior.

d) What is the predicted price of a sedan made in USA that has 4 doors, leather seats, and runs 30 mpg on highway? The predicted price of a sedan that is made in USA, has 4 doors, leather seats, and runs 30 mpg on highway is calculated as: $price_{\hat{}} =$

$\$1,000(5.1353 + 6.4107 + 3.2128 + 0.1305 \cdot 30 + 1.5554 \cdot 4 + 12.1757) =$
 $\$37,071.10.$

```
#Finding the predicted values using the model using fitted model for prediction
prediction<- (predict(fitted.model, data.frame(bodystyle.rel="sedan", country
.rel
="USA", hwy=30, doors=4, leather.rel="yes"))))
print(prediction*1000)

##          1
## 37071.14
```

#Box Cox 2.1

#(a) Is the decrease in BMI percentile (preBMI-postBMI) normally distributed? Plot a histogram and test for normality of the distribution.

#Importing data for 2.1

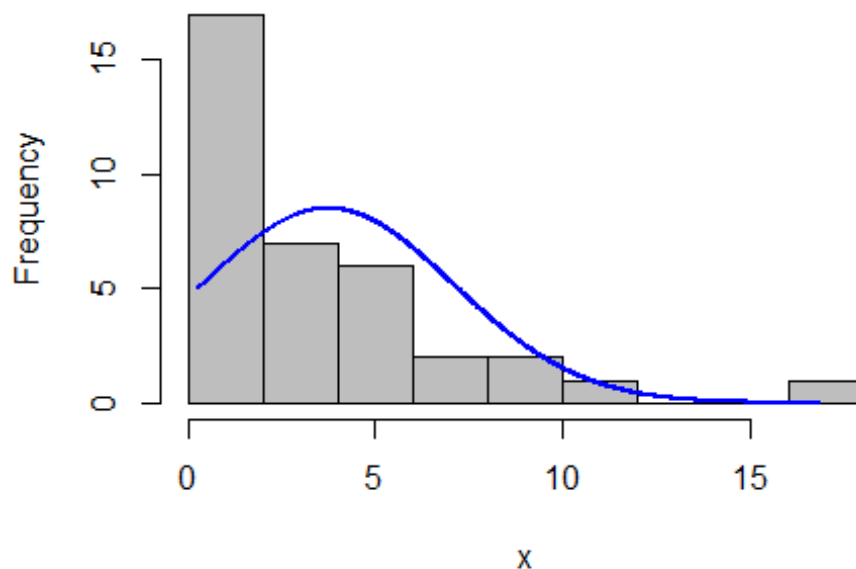
```
Obesity<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/Exercise Files/Exercise2.1Data.csv",header=TRUE, sep=",")
```

#Finding the difference in BMI

```
BMIdiff<- Obesity$preBMI-Obesity$postBMI
```

#Checking for normality

```
plotNormalHistogram(BMIdiff)
```



```
#Shapiro Test for normality
```

```
shapiro.test(BMIdiff)
```

```
##
```

```
## Shapiro-Wilk normality test
```

```
##
```

```
## data: BMIdiff
```

```
## W = 0.79159, p-value = 1.114e-05
```

#(b) Find the optimal Lambda for Box-Cox transformation. Transform the change in BMI percentile (find the appropriate transformation in Table 2.1), and show that the transformed variable is normally distributed. Plot the histogram and do a formal testing.

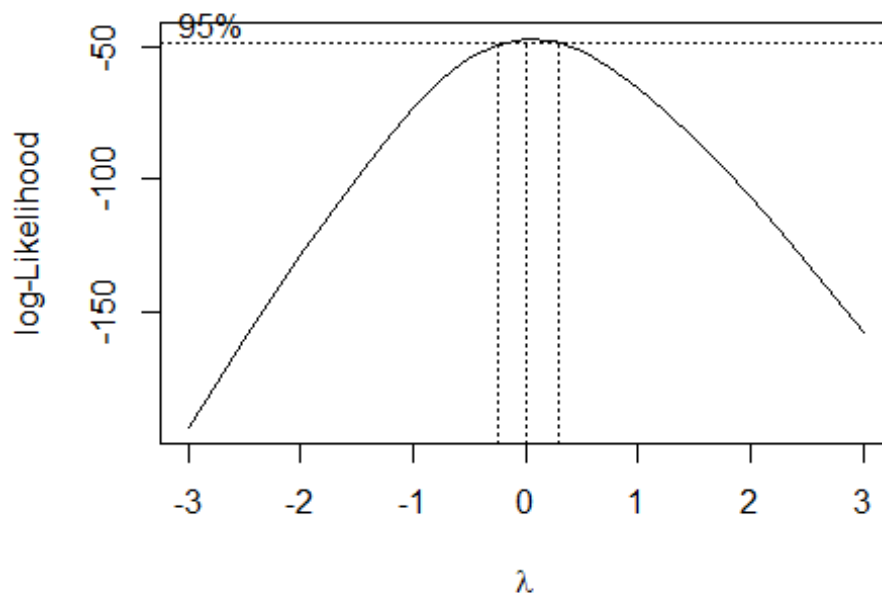
```
#Finding indicator variables
```

```
female<- relevel(as.factor(Obesity$gender), ref="M")
```

```
control<- relevel(as.factor(Obesity$group), ref="Tx")
```

```
#Finding lambda for box-cox
```

```
BoxCox.fit<- boxcox(BMIdiff ~ age + female + control, data=Obesity, lambda =  
seq(-3,3,1/4), interp = FALSE)
```



```
BoxCox.data<- data.frame(BoxCox.fit$x, BoxCox.fit$y)
```

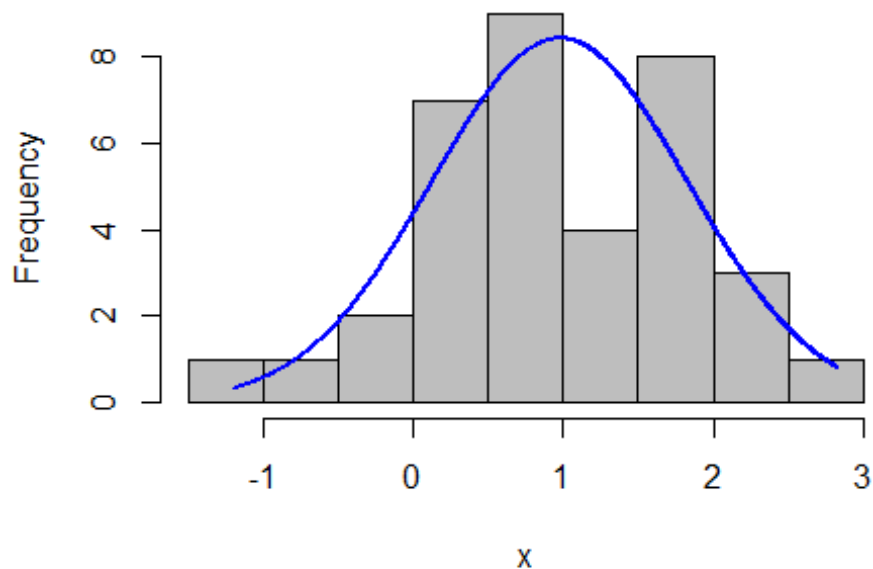
```
ordered.data<- BoxCox.data[with(BoxCox.data, order(-BoxCox.fit.y)),]
```

```
ordered.data[1,]
```

```
##      BoxCox.fit.x BoxCox.fit.y
## 13           0    -46.71658

#Applying the box-cox transformation with lambda from above
BMIdiff.tr<- log(BMIdiff)

#Running normality test again but on the transformed variables
plotNormalHistogram(BMIdiff.tr)
```



```
shapiro.test(BMIdiff.tr)

##
##  Shapiro-Wilk normality test
##
## data:  BMIdiff.tr
## W = 0.9877, p-value = 0.9532

#(c) Fit the general regression model to the Box-Cox transformed change in BM
I percentile. Does this model have a good fit?

#Creating general linear model but on the transformed response
summary(fitted.model<- glm(BMIdiff.tr ~ age + female + control, data=Obesity,
family=gaussian(link=identity)))

##
## Call:
## glm(formula = BMIdiff.tr ~ age + female + control, family = gaussian(link
```



```

= identity),
##      data = Obesity)
##
## Deviance Residuals:
##      Min        1Q      Median        3Q        Max
## -1.31141  -0.41700  -0.08094   0.41958   1.47509
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.14375    0.45959   2.489 0.018218 *
## age          0.05008    0.03651   1.372 0.179731
## femaleF      -0.49862    0.21708  -2.297 0.028317 *
## controlCx    -0.93835    0.22307  -4.207 0.000195 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.4187965)
##
##      Null deviance: 25.432  on 35  degrees of freedom
## Residual deviance: 13.401  on 32  degrees of freedom
## AIC: 76.59
##
## Number of Fisher Scoring iterations: 2

#Estimated sigma
sigma(fitted.model)

## [1] 0.6471448

#Checking again to see if the model fits after transformation
null.model<- glm(BMIdiff.tr ~ 1, family=gaussian(link=identity))

#Printing deviance test for fit
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 23.06361 (df=2)

#Printing p-value for fit
print(p.value<- pchisq(deviance, df=3, lower.tail = FALSE))

## 'log Lik.' 3.916872e-05 (df=2)

#using fitted model for prediction
pred.BMIdiff.tr<- predict(fitted.model, data.frame(female="F", age=9,
control="Cx"))
print(exp(pred.BMIdiff.tr))

##      1
## 1.170609

#Box Cox 2.2

```

#(a) Construct a histogram of the score. Does the distribution look normal? Perform the test for normality. Draw conclusion.

#Importing data for 2.2

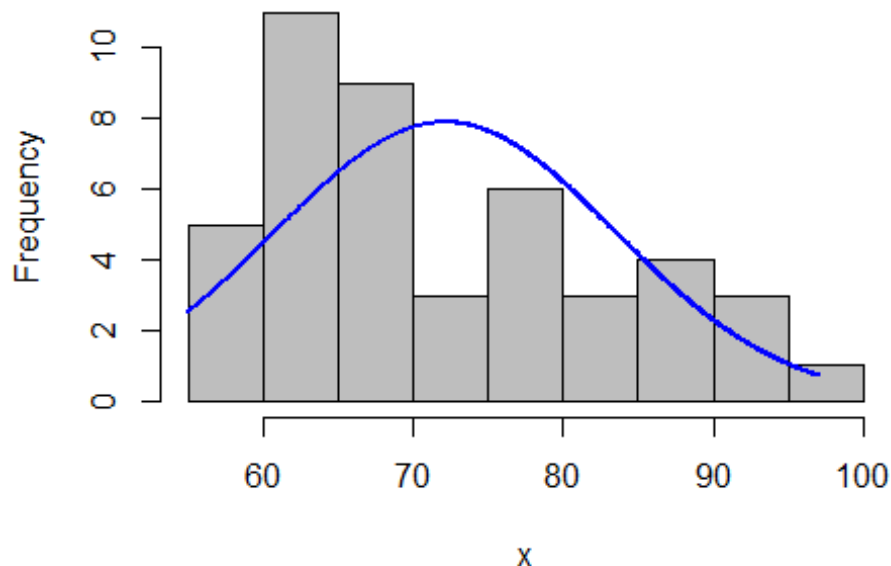
```
QIdata<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/Exercise Files/Exercise2.2Data.csv",header=TRUE, sep=",")
```

#Finding the difference in BMI

```
BMIdiff<- QIdata$preBMI-QIdata$postBMI
```

#Checking for normality

```
plotNormalHistogram(QIdata$score)
```



#Shapiro Test for normality

```
shapiro.test(QIdata$score)
```

```
##
```

```
## Shapiro-Wilk normality test
```

```
##
```

```
## data: QIdata$score
```

```
## W = 0.94357, p-value = 0.02913
```

#(b) Transform the score variable using a meaningful Box-Cox transformation and assure that it is now normally distributed by plotting the histogram and doing normality testing.

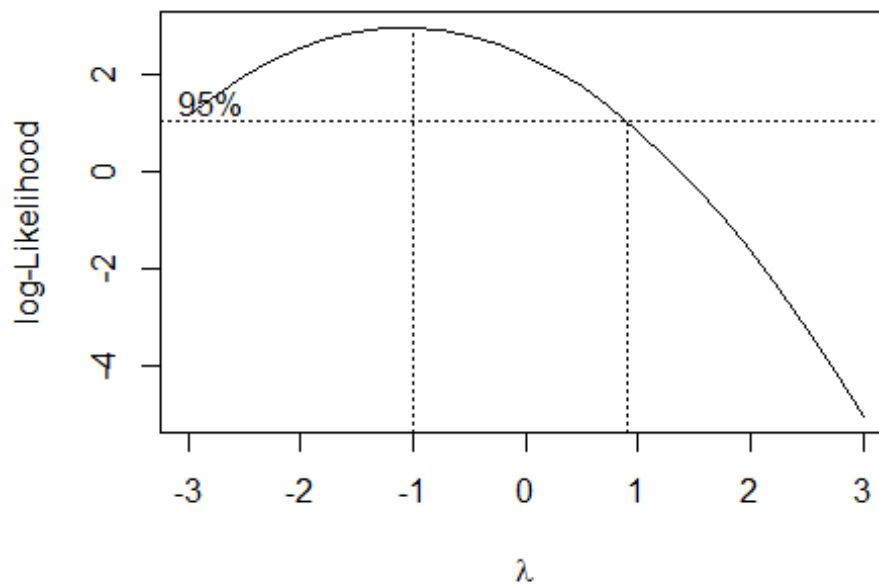
#Finding indicator variables and rescaling

```

desgn.rel<- relevel(as.factor(QIdata$desgn), ref="staff")
priorQI.rel<- relevel(as.factor(QIdata$priorQI), ref="no")
score<- QIdata$score/100

#Finding lambda for box-cox
BoxCox.fit<- boxcox(score ~ desgn.rel + wrkyrs + priorQI.rel,
data=QIdata, lambda=seq(-3,3,1/4), interp = FALSE)

```



```

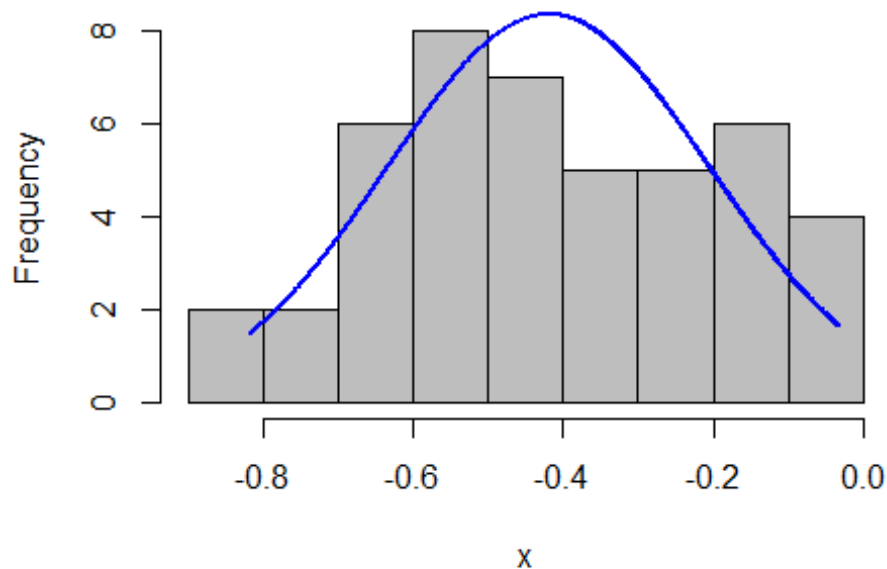
BoxCox.data<- data.frame(BoxCox.fit$x, BoxCox.fit$y)
ordered.data<- BoxCox.data[with(BoxCox.data, order(-BoxCox.fit.y)),]
ordered.data[1,]

##   BoxCox.fit.x BoxCox.fit.y
## 9           -1      2.940242

#Applying the box-cox transformation with lambda from above
score.tr<- 1-(1/score)

#Running normality test again but on the transformed variables
plotNormalHistogram(score.tr)

```



#Shapiro test for transformed variables

```
shapiro.test(score.tr)
```

```
##
```

```
## Shapiro-Wilk normality test
```

```
##
```

```
## data: score.tr
```

```
## W = 0.96606, p-value = 0.2073
```

#(c) Run the general linear regression model on the transformed score. What predictors are significant at the 0.05 level?

#Creating general linear model but on the transformed response

```
summary(fitted.model<- glm(score.tr ~ desgn.rel + wrkyrs + priorQI.rel,
data=QIdata, family=gaussian(link = identity)))
```

```
##
```

```
## Call:
```

```
## glm(formula = score.tr ~ desgn.rel + wrkyrs + priorQI.rel, family = gaussian(link = identity),
```

```
## data = QIdata)
```

```
##
```

```
## Deviance Residuals:
```

```
##      Min       1Q   Median       3Q      Max
```

```
## -0.39192 -0.15611 -0.00556  0.14372  0.36543
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.609290   0.094150  -6.471 1.03e-07 ***
## desgn.reldoctor  0.179873   0.091507   1.966  0.0563 .
## desgn.relnurse   0.212200   0.087173   2.434  0.0195 *
## wrkysrs         0.000243   0.004086   0.059  0.9529
## priorQI.relyes   0.077263   0.068323   1.131  0.2649
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.04317095)
##
## Null deviance: 2.0290  on 44  degrees of freedom
## Residual deviance: 1.7268  on 40  degrees of freedom
## AIC: -7.0122
##
## Number of Fisher Scoring iterations: 2

#Estimated sigma
sigma(fitted.model)

## [1] 0.2077762
```

- (d) Interpret the estimates of the significant beta coefficients. Does the model fit the data well? Conduct the chi-squared deviance test.

The estimated mean inverse transformed score for nurses is 0.2122 points above that for staff, and for doctors it is 0.1799 points above that for staff.

The model doesn't really fit the data well, as seen from the large p-value of the deviance test given below.

```
#checking model fit
null.model<- glm(score.tr ~ 1, family = gaussian(link=identity))

#Running deviance test for model fit
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 7.256887 (df=2)

#Running p-value for model fit
print(p.value<- pchisq( deviance, df=4, lower.tail = FALSE))

## 'log Lik.' 0.1229198 (df=2)
```

- (e) Predict the score for a nurse who has worked at the center for seven years and who had previously been a co-PI on a grant that involved quality assurance

We calculate $score_{\hat{}} = 100 \cdot (1 - (0.6093 + 0.2122 + 0.0002 \cdot 7 + 0.0773))_{\hat{}} = 75.84951$.

```
#Predicting values using the fitted model
pscore.tr<- predict(fitted.model, data.frame(desgn.rel="nurse", wrkysrs=7,
```

```
priorQI.rel="yes"))
print(100/(1-pscore.tr))
```

```
##          1
## 75.86528
```

#Gamma 2.6

#This problem uses the same data as 2.1 (a) Fit the gamma regression model with the log link function. Write down the fitted model. Check its goodness of fit.

#Importing data for 2.6 (this uses the same data as 2.1)

```
ObesityGamma<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/Exercise Files/Exercise2.1Data.csv",header=TRUE, sep=",")
```

#Finding the difference in BMI

```
BMIdiff<- ObesityGamma$preBMI-Obesity$postBMI
```

#Creating the gamma model

```
summary(fitted.model<- glm(BMIdiff ~ gender + age + group, data=ObesityGamma, family=Gamma(link=log)))
```

```
##
```

```
## Call:
```

```
## glm(formula = BMIdiff ~ gender + age + group, family = Gamma(link = log), data = ObesityGamma)
```

```
##
```

```
## Deviance Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -1.2258  -0.5318  -0.2426   0.3200   1.5279
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.04415    0.45077  -0.098 0.922581
## genderM      0.38624    0.22749   1.698 0.099250 .
## age          0.04703    0.03826   1.229 0.227963
## groupTx      0.98701    0.23376   4.222 0.000187 ***
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## (Dispersion parameter for Gamma family taken to be 0.4599244)
```

```
##
```

```
## Null deviance: 24.361 on 35 degrees of freedom
```

```
## Residual deviance: 13.213 on 32 degrees of freedom
```

```
## AIC: 148.76
```

```
##
```

```
## Number of Fisher Scoring iterations: 7
```

#Checking the fit of the model

```
null.model<- glm(BMIdiff ~ 1, data=ObesityGamma, family=Gamma(link=log))
```

```

#Printing deviance test for fit
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 23.83593 (df=2)

#Printing p-value for fit
print(p.value<- pchisq(deviance,3,lower.tail = FALSE))

## 'log Lik.' 2.70297e-05 (df=2)

```

- (b) What variables are significant predictors in this model? Use the 5% significance level. Only group is a significant predictor.
- (c) Interpret estimated significant regression coefficients. Estimated average decrease in BMI percentile for patients in the intervention group is $\exp(0.9870) \cdot 100\% = 268.32\%$ of that for patients in the control group.
- (d) Predict change in BMI percentile for a 9-year old girl in the control group. Compare the prediction with the one obtained in Exercise 2.1.

We calculate the predicted value as follows: $BMI_{diff} = \exp(-0.0442 + 0.0470 \cdot 9) = 1.4605$.

```

#Using the gamma model to predict values
print(predict(fitted.model, data.frame(gender="F", age=9, group="Cx"),
type="response"))

##          1
## 1.461058

```

#Gamma 2.7

#This problem uses the same data as 2.2 (a) Fit the gamma regression model with the log link function. Write down the fitted model. Check its goodness of fit.

```

#Importing data for 2.7 (this uses the same data as 2.2)
QIGamma<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/Exercise Files/Exercise2.2Data.csv",header=TRUE, sep=",")

#Rescaling and relevel
design.rel<- relevel(as.factor(QIGamma$design), ref="staff")
priorQI.rel<- relevel(as.factor(QIGamma$priorQI), ref="no")
score<- QIGamma$score/100

#Creating the gamma model
summary(fitted.model<- glm(score ~ design.rel + wrkys + priorQI.rel,
data=QIGamma, family=Gamma(link=log)))

```

```
##
## Call:
## glm(formula = score ~ desgn.rel + wrkyrs + priorQI.rel, family = Gamma(link = log),
##      data = QIGamma)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.25489  -0.12255  -0.01360   0.08738   0.29224
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.1427297   0.0686964   60.305  <2e-16 ***
## desgn.reldoctor  0.1339899   0.0667675    2.007   0.0516 .
## desgn.relnurse   0.1540443   0.0636050    2.422   0.0201 *
## wrkyrs          -0.0002455   0.0029813   -0.082   0.9348
## priorQI.relyes   0.0532444   0.0498513    1.068   0.2919
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 0.02298337)
##
## Null deviance: 1.06477  on 44  degrees of freedom
## Residual deviance: 0.90565  on 40  degrees of freedom
## AIC: 348.1
##
## Number of Fisher Scoring iterations: 4

#Checking the fit of the model
null.model<- glm(score ~ 1, data=QIGamma, family=Gamma(link=log))

#Printing deviance test for fit
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 7.310283 (df=2)

#Printing p-value for fit
print(p.value<- pchisq(deviance,4,lower.tail=FALSE))

## 'log Lik.' 0.1203719 (df=2)
```

- (b) Discuss significance of the beta coefficients. Interpret the estimated significant coefficients.

Indicators of doctor and nurse are significant at the 5% level. The estimated mean score for doctors is $\exp(0.1340) \cdot 100\% = 114.34\%$ of that for staff. The estimated mean score for nurses is $\exp(0.1540) \cdot 100\% = 116.65\%$ of that for staff.

- (c) Predict the score for a nurse who has worked at the center for seven years and who had previously been a co-PI on a grant that involved quality assurance component. Compare that predicted score to the one obtained in Exercise 2.2.

The predicted value is calculated as: $score = 100 \cdot \exp(-0.4624 + 0.1540 - 0.0002 \cdot 7 + 0.0532) = 77.3678$. The predicted score in Exercise 2.2 is 75.8653 which is smaller than what we predict here.

#Using the gamma model to predict values

```
print(predict(fitted.model, data.frame(design.rel="nurse", wrkys=7,
priorQI.rel="yes"), type="response"))
```

```
##          1
## 77.34687
```

#Logistic 3.2

#(a) Fit a binary logistic model. Write down the fitted model. Discuss significance of predictor variables, and goodness of fit of the model. Use $\alpha = 0.05$.

#Importing data for 3.2

```
Psoriasis<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/Exercise Files/Exercise3.2Data.csv",header=TRUE, sep=",")
```

#Releveling

```
gender.rel<- relevel(as.factor(Psoriasis$gender), ref="F")
medication.rel<- relevel(as.factor(Psoriasis$medication), ref="B")
```

#Creating the logistic model

```
summary(fitted.model<- glm(relief ~ gender.rel + age + medication.rel,
data=Psoriasis,family=binomial(link=logit)))
```

```
##
## Call:
## glm(formula = relief ~ gender.rel + age + medication.rel, family = binomial(link = logit),
##      data = Psoriasis)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.9823  -0.3628   0.1974   0.5294   1.6865
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -6.79921    3.11430  -2.183   0.0290 *
## gender.relM     3.17132    1.47097   2.156   0.0311 *
## age            0.17131    0.08691   1.971   0.0487 *
## medication.relA 3.81641    1.54617   2.468   0.0136 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```
##      Null deviance: 36.652  on 29  degrees of freedom
## Residual deviance: 21.008  on 26  degrees of freedom
## AIC: 29.008
##
## Number of Fisher Scoring iterations: 6

#Calculating and printing AICC
p<-4
n<-30
print(AICC<- -2*logLik(fitted.model)+2*p*n/(n-p-1))

## 'log Lik.' 30.60842 (df=4)

#Calculating and printing BIC
BIC(fitted.model)

## [1] 34.6132

#Checking to see if this model fits
null.model<- glm(relief ~ 1, data=Psoriasis, family=binomial(link=logit))

#Deviance test for fit
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 15.64344 (df=1)

#P-value for fit
print(p.value<- pchisq(deviance,3,lower.tail = FALSE))

## 'log Lik.' 0.001341726 (df=1)
```

(b) Give interpretation of the estimated significant regression coefficients.

The estimated odds in favor of relief from psoriasis for male patients are $\exp(3.1713) \cdot 100\% = 2,383.85\%$ of those for female patients. As age increases by one year, the estimated odds increase by $(\exp(0.1713) - 1) \cdot 100\% = 18.68\%$. The estimated odds for patients taking medication A are $\exp(3.8164) \cdot 100\% = 4,544.03\%$ of those for patients taking medication B.

(c) Find the predicted probability of relief from psoriasis for a 50-year old woman who is administered the medication A treatment.

#Answer should be $P_0(\text{relief})=0.99625$

```
#Calculating predictions using the model
print(predict(fitted.model, data.frame(gender.rel="F", age=50,
medication.rel="A"), type="response"))

##      1
## 0.9962508

#Logistic 3.3
```

#(a) Fit a binary logistic model to the data. What predictors turn out to be significant at the 5% level? How good is the fit of the model?

#Importing data for 3.3

```
Novel<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/Exercise Files/Exercise3.3Data.csv",stringsAsFactors=TRUE,header=TRUE, sep=",")
```

Novel

##	success	cover	methods	novels	years
## 1	yes	yes	one	many	18
## 2	no	no	one	first	7
## 3	no	yes	none	several	10
## 4	yes	yes	many	many	6
## 5	no	yes	none	several	1
## 6	no	no	one	several	1
## 7	no	no	one	first	11
## 8	yes	no	one	several	19
## 9	yes	yes	none	first	5
## 10	no	no	none	many	2
## 11	no	no	one	several	10
## 12	no	no	many	many	9
## 13	yes	no	many	several	6
## 14	yes	yes	many	many	8
## 15	no	no	one	several	12
## 16	no	no	none	many	2
## 17	yes	no	none	several	17
## 18	yes	yes	many	first	10
## 19	yes	no	none	several	7
## 20	no	no	one	first	12
## 21	no	yes	none	several	7
## 22	no	yes	none	many	4
## 23	no	no	one	several	9
## 24	yes	no	many	several	13
## 25	yes	yes	none	first	6
## 26	no	no	none	many	2
## 27	yes	yes	one	several	7
## 28	yes	yes	many	many	17
## 29	yes	yes	many	first	18
## 30	yes	yes	one	several	17
## 31	no	yes	none	several	9
## 32	no	no	one	several	11
## 33	yes	yes	many	first	17
## 34	no	no	many	many	1
## 35	no	no	many	many	6
## 36	no	yes	none	several	1
## 37	yes	yes	many	first	6
## 38	yes	yes	one	many	4
## 39	no	yes	none	many	7
## 40	no	no	one	first	12

```

## 41      no      no      one several      7
## 42     yes     yes      one several      9
## 43      no      no      one several      8
## 44      no      no      one several      2

#ReLeveling
cover.rel<- relevel(as.factor(Novel$cover), ref="no")
methods.rel<- relevel(as.factor(Novel$methods), ref="none")
novels.rel<- relevel(as.factor(Novel$novels), ref="many")

Novel <- Novel %>% mutate_each(funs(as.factor), c(cover, methods, novels))

## Warning: `funs()` was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()` :
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

## Warning: `mutate_each_()` was deprecated in dplyr 0.7.0.
## Please use `across()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

#Creating the Logistic model
summary(fitted.model<- glm(success ~ cover.rel + methods.rel + novels.rel
+ years, data=Novel,family=binomial(link=logit)))

##
## Call:
## glm(formula = success ~ cover.rel + methods.rel + novels.rel +
##      years, family = binomial(link = logit), data = Novel)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.5998  -0.5951  -0.1052   0.3947   2.6358
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -6.8762     2.3656  -2.907  0.00365 **
## cover.relyes     3.5238     1.3120   2.686  0.00724 **
## methods.relmny     3.9286     1.7917   2.193  0.02834 *

```

```

## methods.relone      0.7008      1.1914      0.588      0.55642
## novels.relfirst     1.8768      1.5784      1.189      0.23442
## novels.relseveral   1.3992      1.3254      1.056      0.29113
## years               0.2907      0.1269      2.291      0.02198 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 60.176  on 43  degrees of freedom
## Residual deviance: 32.071  on 37  degrees of freedom
## AIC: 46.071
##
## Number of Fisher Scoring iterations: 6

#Calculating and printing AICC
p<-7
n<-44
print(AICC<- -2*logLik(fitted.model)+2*p*n/(n-p-1))

## 'log Lik.' 49.18253 (df=7)

#Calculating and printing BIC
BIC(fitted.model)

## [1] 58.56075

#Checking to see if this model fits
null.model<- glm(success ~ 1, data=Novel, family=binomial(link=logit))

#Deviance test for fit
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 28.10479 (df=1)

#P-value for fit
print(p.value<- pchisq(deviance,6,lower.tail = FALSE))

## 'log Lik.' 8.97874e-05 (df=1)

```

(b) Give interpretation of the estimated significant beta coefficients.

The estimated odds in favor of financial success for a novel with catchy cover are $\exp(3.5238) \cdot 100\% = 3,391.31\%$ of those for a novel without a catchy cover. The estimated odds for a publisher with many promotional methods are $\exp(3.9286) \cdot 100\% = 5,083.58\%$ of those for a publisher with no promotional methods. For every additional year a publisher was in business prior to publication of a novel, the estimated odds in favor of financial success of the novel increase by $(\exp(0.2907) - 1) \cdot 100\% = 33.74\%$.

(c) Suppose a newly established publishing house prints a novel by some previously unknown author, and doesn't advertise the publication. Find the estimated

probability that this novel is successful financially, if it has an extremely catchy cover.

Answer should be $P_0(\text{success})=0.18609$

```
#Calculating predictions using the model
print(predict(fitted.model, data.frame(cover.rel="yes", methods.rel="none",
novels.rel="first", years=0), type="response"))
```

```
##          1
## 0.186086
```

#Probit 3.4

#(a) Run the binary logistic model, regressing on all the predictors. Identify variables that are significant predictors of loan default at the 5% level of significance. Analyze the model fit.

```
#Importing data for 3.4
```

```
LoanProbit<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/
Exercise Files/Exercise3.4Data.csv",stringsAsFactors=TRUE,header=TRUE, sep=",
")
```

```
#Creating probit model
```

```
summary(fitted.model<- glm(default ~ LTV + age + income, data=LoanProbit,
family=binomial(link=probit)))
```

```
##
## Call:
## glm(formula = default ~ LTV + age + income, family = binomial(link = probit),
##      data = LoanProbit)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.5284  -0.6871  -0.1290   0.5551   1.9530
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.60587    2.34383  -0.685   0.4933
## LTV          0.06200    0.02853   2.173   0.0297 *
## age         -0.09865    0.04121  -2.394   0.0167 *
## incomelow    0.63924    0.59365   1.077   0.2816
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 45.004  on 34  degrees of freedom
## Residual deviance: 28.173  on 31  degrees of freedom
## AIC: 36.173
```

```
##
## Number of Fisher Scoring iterations: 7

#Calculating and printing AICC
p<-4
n<-35
print(AICC<- -2*logLik(fitted.model)+2*p*n/(n-p-1))

## 'log Lik.' 37.50665 (df=4)

#Calculating and printing BIC
BIC(fitted.model)

## [1] 42.39471

#Checking to see if this model fits
null.model<- glm(default ~ 1, data=LoanProbit, family=binomial(link=probit))

#Deviance test for fit
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 16.83055 (df=1)

#P-value for fit
print(p.value<- pchisq(deviance,3,lower.tail = FALSE))

## 'log Lik.' 0.0007657748 (df=1)
```

- (b) Interpret the estimated significant beta coefficients. What is your suggestion in order for the bank to decrease the default rate of home equity loans?

As loan-to-value ratio increases by one, the estimated odds in favor of default increase by $(\exp(0.1059) - 1) \cdot 100\% = 11.17\%$. As the age of a client increases by one year, the estimated odds in favor of default change by $(\exp(-0.1616) - 1) \cdot 100\% = -14.92\%$, that is, decrease by 14.92%. To decrease the default rate, the bank might want to give loans with smaller loan-to-value ratio, and/or give loans to older clients.

- (c) Give a point estimate for the probability of loan default if LTV ratio is 50%, and the borrower is a 50-year old men with high income.

Answer should be $P_0(\text{success})=0.0030374$

```
#Calculating predictions using the model
print(predict(fitted.model, data.frame(LTV=50, age=50, income="high"),
type="response"))

##          1
## 0.0002923167
```

#Probit 3.5

#(a) Model the probability of being a cardiac patient via the binary logistic regression. Write the fitted model explicitly. Discuss the goodness of fit of

the model and significance of the regression coefficients. Assume $\alpha = 0.01$ for all tests.

#Importing data for 3.5

```
CardiacProbit<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/Exercise Files/Exercise3.5Data.csv",stringsAsFactors=TRUE,header=TRUE, sep=",")
```

#Creating probit model

```
summary(fitted.model<- glm(group ~ A + W, data=CardiacProbit, family=binomial(link=probit)))
```

```
##
```

```
## Call:
```

```
## glm(formula = group ~ A + W, family = binomial(link = probit),  
##      data = CardiacProbit)
```

```
##
```

```
## Deviance Residuals:
```

```
##      Min        1Q      Median        3Q        Max  
## -2.2203  -0.9519  -0.1981   1.1270   1.6154
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error z value Pr(>|z|)  
## (Intercept) -0.62995    0.53497  -1.178   0.2390  
## A           0.24895    0.11125   2.238   0.0252 *  
## W           0.01047    0.07694   0.136   0.8918
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
```

```
##      Null deviance: 66.542  on 47  degrees of freedom
```

```
## Residual deviance: 58.302  on 45  degrees of freedom
```

```
## AIC: 64.302
```

```
##
```

```
## Number of Fisher Scoring iterations: 4
```

#Calculating and printing AICC

```
p<-3
```

```
n<-48
```

```
print(AICC<- -2*logLik(fitted.model)+2*p*n/(n-p-1))
```

```
## 'log Lik.' 64.84727 (df=3)
```

#Calculating and printing BIC

```
BIC(fitted.model)
```

```
## [1] 69.91541
```

#Checking to see if this model fits

```
null.model<- glm(group ~ 1, data=CardiacProbit, family=binomial(link=probit))
```



```

#Deviance test for fit
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 8.240317 (df=1)

#P-value for fit
print(p.value<- pchisq(deviance,2,lower.tail = FALSE))

## 'log Lik.' 0.01624194 (df=1)

#using fitted model for prediction
print(predict(fitted.model, data.frame(A=2, W=4), type="response"))

##          1
## 0.4640763

```

#Comp Log-Log 3.4

```

#Importing data for 3.4
LoanComp<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/Exercise Files/Exercise3.4Data.csv",stringsAsFactors=TRUE,header=TRUE, sep=",")

#Creating comp Log Log
summary(fitted.model<- glm(default ~ LTV + age + income, data=LoanComp,
family=binomial(link=cloglog)))

##
## Call:
## glm(formula = default ~ LTV + age + income, family = binomial(link = cloglog),
##      data = LoanComp)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -1.4972  -0.6653  -0.2425   0.4326   1.9708
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.68156    3.27613  -0.819   0.4131
## LTV          0.07896    0.04114   1.919   0.0550 .
## age         -0.12254    0.05111  -2.398   0.0165 *
## incomelow    0.89073    0.72016   1.237   0.2161
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 45.004  on 34  degrees of freedom
## Residual deviance: 28.436  on 31  degrees of freedom
## AIC: 36.436

```

```
##
## Number of Fisher Scoring iterations: 7

#Calculating and printing AICC
p<-4
n<-35
print(AICC<- -2*logLik(fitted.model)+2*p*n/(n-p-1))

## 'log Lik.' 37.7691 (df=4)

#Calculating and printing BIC
BIC(fitted.model)

## [1] 42.65716

#Checking if the model fits
null.model<- glm(default ~ 1, data=LoanComp, family=binomial(link=cloglog))

#Deviance test for model fit
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 16.56811 (df=1)

#P-value for model fit
print(p.value<- pchisq(deviance,3,lower.tail = FALSE))

## 'log Lik.' 0.0008670163 (df=1)

#Predicting values using the fitted model
print(predict(fitted.model, data.frame(LTV=50, age=50, income="high"),
type="response"))

##          1
## 0.007713725
```

##Comp Log-Log 3.5

```
#Importing data for 3.5
CardiacComp<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410
/Exercise Files/Exercise3.5Data.csv",stringsAsFactors=TRUE,header=TRUE, sep="
,")

#Creating comp log log
summary(fitted.model<- glm(group ~ A + W, data=CardiacComp,
family=binomial(link=cloglog)))

##
## Call:
## glm(formula = group ~ A + W, family = binomial(link = cloglog),
##      data = CardiacComp)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
```

```

## -2.1991 -0.9736 -0.2271 1.1681 1.5382
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.03922    0.62874  -1.653  0.0984 .
## A           0.23818    0.10916   2.182  0.0291 *
## W           0.01674    0.09199   0.182  0.8556
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##      Null deviance: 66.542  on 47  degrees of freedom
## Residual deviance: 59.047  on 45  degrees of freedom
## AIC: 65.047
##
## Number of Fisher Scoring iterations: 8

#Calculating and printing AICC
p<-3
n<-48
print(AICC<- -2*logLik(fitted.model)+2*p*n/(n-p-1))

## 'log Lik.' 65.59256 (df=3)

#Calculating and printing BIC
BIC(fitted.model)

## [1] 70.66071

#Checking if the model fits
null.model<- glm(group ~ 1, data=CardiacComp, family=binomial(link=cloglog))

#Deviance test for model fit
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 7.495025 (df=1)

#P-value for model fit
print(p.value<- pchisq(deviance,2,lower.tail = FALSE))

## 'log Lik.' 0.02357632 (df=1)

#Predicting values using the fitted model
print(predict(fitted.model, data.frame(A=2, W=4), type="response"))

##      1
## 0.45612

```

Cumulative Logistic: 4.1

#Cumulative Logistic: 4.1

#First we import the data and fit the cumlogit model

```
Admission <- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/  
Exercise Files/Exercise4.1Data.csv", header = TRUE, sep = ",")
```

Admission

##		GPA	GMAT	status
##	1	2.96	596	admit
##	2	3.14	473	admit
##	3	3.22	482	admit
##	4	3.29	527	admit
##	5	3.69	505	admit
##	6	2.46	693	admit
##	7	3.03	626	admit
##	8	3.19	663	admit
##	9	3.63	447	admit
##	10	3.59	588	admit
##	11	3.30	563	admit
##	12	3.78	591	admit
##	13	3.44	692	admit
##	14	3.48	528	admit
##	15	3.47	552	admit
##	16	3.35	520	admit
##	17	2.89	543	admit
##	18	2.28	523	admit
##	19	3.21	530	admit
##	20	3.58	564	admit
##	21	3.33	565	admit
##	22	2.80	444	border
##	23	3.13	416	border
##	24	2.89	431	border
##	25	3.01	471	border
##	26	2.91	446	border
##	27	2.75	546	border
##	28	2.73	467	border
##	29	3.12	463	border
##	30	3.08	440	notadmit
##	31	3.01	453	notadmit
##	32	3.03	414	notadmit
##	33	3.04	446	notadmit
##	34	2.89	485	notadmit
##	35	2.79	490	notadmit
##	36	2.54	446	notadmit
##	37	2.43	425	notadmit
##	38	2.20	474	notadmit
##	39	3.36	531	notadmit
##	40	2.57	542	notadmit

```

## 41 2.36 482 notadmit
## 42 3.66 420 notadmit

#Now we the predictor GMAT and fit the model
GMAT_rescaled<- Admission$GMAT/100
summary(fitted.model<- clm(as.factor(status) ~ GPA + GMAT_rescaled, data=Admission,
link="logit"))

## formula: as.factor(status) ~ GPA + GMAT_rescaled
## data: Admission
##
## link threshold nobs logLik AIC niter max.grad cond.H
## logit flexible 42 -27.54 63.09 5(0) 2.17e-07 1.6e+04
##
## Coefficients:
## Estimate Std. Error z value Pr(>|z|)
## GPA -3.1194 1.1913 -2.618 0.008833 **
## GMAT_rescaled -2.7755 0.8406 -3.302 0.000961 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
## Estimate Std. Error z value
## admit|border -23.390 5.857 -3.994
## border|notadmit -21.853 5.670 -3.854

#Computations AICC and BIC
p<- 4
n<- 42
print(AICC<- -2*logLik(fitted.model)+2*p*n/(n-p-1))

## 'log Lik.' 64.16976 (df=4)

BIC(fitted.model)

## [1] 70.03935

#Here we check to see if the model fits using a deviance test
null.model<- clm(as.factor(status) ~ 1, data=Admission, link="logit")
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 31.04588 (df=2)

print(p.value<- pchisq(deviance, df=2, lower.tail = FALSE))

## 'log Lik.' 1.813312e-07 (df=2)

#Now we will use the fitted model to predict the probabilities of each admission status for someone with 3.1 GPA and GMAT score of 550
#print(predict(fitted.model, data.frame(GPA=3.1, GMAT_rescaled=5.50), type="prob"))

```

Cumulative Logistic: 4.2

#First we import the data and fit the cumLogit model

```
TVService <- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/  
Exercise Files/Exercise4.2Data.csv", header = TRUE, sep = ",")
```

TVService

##	subscribed	magazine	resolved	satisf
## 1	5	yes	no	5
## 2	49	yes	no	5
## 3	56	no	no	3
## 4	13	yes	yes	5
## 5	27	no	yes	4
## 6	41	yes	yes	5
## 7	2	yes	yes	5
## 8	64	yes	yes	4
## 9	88	yes	yes	4
## 10	43	yes	yes	4
## 11	94	yes	no	4
## 12	8	no	no	1
## 13	9	yes	no	2
## 14	68	yes	no	4
## 15	5	no	yes	2
## 16	108	no	yes	3
## 17	21	yes	yes	4
## 18	25	yes	no	3
## 19	2	no	yes	4
## 20	11	no	no	2
## 21	98	yes	yes	5
## 22	11	no	yes	5
## 23	46	no	no	4
## 24	7	no	no	3
## 25	7	no	yes	5
## 26	9	yes	yes	5
## 27	17	no	no	2
## 28	8	no	yes	2
## 29	9	no	yes	1
## 30	95	no	no	4
## 31	60	no	yes	3
## 32	80	no	yes	4
## 33	2	yes	no	3
## 34	33	yes	yes	4
## 35	5	yes	no	3
## 36	7	no	no	1

#Making the response satisf a categorical variable

```
satisf_categorical<- as.factor(TVService$satisf)
```

#Specifying reference categories and making them categorical variables

```
magazine_releveled<- relevel(as.factor(TVService$magazine), ref="yes")
```

```
resolved_releveled<- relevel(as.factor(TVService$resolved), ref="yes")
```

#Fitting the model

```
summary(fitted.model<- clm(satisf_categorical ~ subscribed + magazine_relevel  
ed  
+ resolved_reveled, data=TVService, link="logit"))
```

```
## formula:
```

```
## satisf_categorical ~ subscribed + magazine_reveled + resolved_reveled
```

```
## data: TVService
```

```
##
```

```
## link threshold nobs logLik AIC niter max.grad cond.H
```

```
## logit flexible 36 -47.05 108.10 5(0) 8.00e-10 6.2e+04
```

```
##
```

```
## Coefficients:
```

```
## Estimate Std. Error z value Pr(>|z|)
```

```
## subscribed 0.010516 0.009686 1.086 0.27760
```

```
## magazine_reveledno -1.917509 0.677141 -2.832 0.00463 **
```

```
## resolved_reveledno -1.428832 0.655864 -2.179 0.02937 *
```

```
## ---
```

```
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## Threshold coefficients:
```

```
## Estimate Std. Error z value
```

```
## 1|2 -4.2359 0.9678 -4.377
```

```
## 2|3 -2.8298 0.8126 -3.483
```

```
## 3|4 -1.5740 0.7442 -2.115
```

```
## 4|5 0.3350 0.6708 0.499
```

#Computations for AICC and BIC

```
p<-7
```

```
n<-36
```

```
print(AICC<- -2*logLik(fitted.model)+2*p*n/(n-p-1))
```

```
## 'log Lik.' 112.0974 (df=7)
```

```
BIC(fitted.model)
```

```
## [1] 119.182
```

#Checking to see how well the model fits using a deviance test

```
null.model<- clm(satisf_categorical ~ 1, data=TVService, link="logit")
```

```
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))
```

```
## 'log Lik.' 14.7994 (df=4)
```

```
print(p.value<- pchisq(deviance, df=3, lower.tail = FALSE))
```

```
## 'log Lik.' 0.001996357 (df=4)
```

#Predicting P probabilities of each of the five levels of the satisfaction score for a caller who had been subscribed for 3 months, doesn't receive the magazine, and whose issue was resolved over the phone

```
print(predict(fitted.model, data.frame(subscribed=3, magazine_relevel="no",
resolved_relevel="yes"), type="prob"))

## $fit
##           1           2           3           4           5
## 1 0.08706729 0.1930496 0.2972414 0.3247592 0.09788253
```

Cumulative Probit: 4.1

#We start by importing the data

```
Admission <- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/
Exercise Files/Exercise4.1Data.csv", header = TRUE, sep = ",")
```

#Here we rescale the predictor GMAT and fit the cumprobit model

```
GMAT_rescaled<- Admission$GMAT/100
```

```
summary(fitted.model<- clm(as.factor(status) ~ GPA + GMAT_rescaled, data=Admission,
link="probit"))
```

```
## formula: as.factor(status) ~ GPA + GMAT_rescaled
## data: Admission
##
## link threshold nobs logLik AIC niter max.grad cond.H
## probit flexible 42 -27.59 63.19 6(0) 1.04e-13 1.3e+04
##
## Coefficients:
## Estimate Std. Error z value Pr(>|z|)
## GPA -1.7356 0.6222 -2.790 0.005276 **
## GMAT_rescaled -1.6546 0.4573 -3.618 0.000297 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
## Estimate Std. Error z value
## admit|border -13.603 3.078 -4.42
## border|notadmit -12.724 3.001 -4.24
```

Admission

```
## GPA GMAT status
## 1 2.96 596 admit
## 2 3.14 473 admit
## 3 3.22 482 admit
## 4 3.29 527 admit
## 5 3.69 505 admit
## 6 2.46 693 admit
## 7 3.03 626 admit
## 8 3.19 663 admit
## 9 3.63 447 admit
## 10 3.59 588 admit
```



```
## 11 3.30 563 admit
## 12 3.78 591 admit
## 13 3.44 692 admit
## 14 3.48 528 admit
## 15 3.47 552 admit
## 16 3.35 520 admit
## 17 2.89 543 admit
## 18 2.28 523 admit
## 19 3.21 530 admit
## 20 3.58 564 admit
## 21 3.33 565 admit
## 22 2.80 444 border
## 23 3.13 416 border
## 24 2.89 431 border
## 25 3.01 471 border
## 26 2.91 446 border
## 27 2.75 546 border
## 28 2.73 467 border
## 29 3.12 463 border
## 30 3.08 440 notadmit
## 31 3.01 453 notadmit
## 32 3.03 414 notadmit
## 33 3.04 446 notadmit
## 34 2.89 485 notadmit
## 35 2.79 490 notadmit
## 36 2.54 446 notadmit
## 37 2.43 425 notadmit
## 38 2.20 474 notadmit
## 39 3.36 531 notadmit
## 40 2.57 542 notadmit
## 41 2.36 482 notadmit
## 42 3.66 420 notadmit
```

#Computations for AICC and BIC

```
p<- 4
n<- 42
print(AICC<- -2*logLik(fitted.model)+2*p*n/(n-p-1))
```

```
## 'log Lik.' 64.26707 (df=4)
```

```
BIC(fitted.model)
```

```
## [1] 70.13666
```

#Now we check to see if the model fits using a deviance test

```
null.model<- clm(as.factor(status) ~ 1, data=Admission, link="probit")
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))
```

```
## 'log Lik.' 30.94857 (df=2)
```

```
print(p.value<- pchisq(deviance, df=2, lower.tail = FALSE))
```

```
## 'log Lik.' 1.903718e-07 (df=2)
```

```
#Now we will use the fitted model to predict the probabilities of each admission status for someone with 3.1 GPA and GMAT score of 550  
#print(predict(fitted.model, data.frame(GPA=3.1, GMAT_rescaled=5.50), type="probit"))
```

Cumulative Probit 4.2

```
#First we import the data and fit the cumprobit model
```

```
TVService <- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/Exercise Files/Exercise4.2Data.csv", header = TRUE, sep = ",")
```

```
#Making the response satisfy a categorical variable  
satisf_categorical<- as.factor(TVService$satisf)
```

```
#Specifying reference categories and making them categorical variables  
magazine_releveled<- relevel(as.factor(TVService$magazine), ref="yes")  
resolved_releveled<- relevel(as.factor(TVService$resolved), ref="yes")
```

```
#Fitting the model
```

```
summary(fitted.model<- clm(satisf_categorical ~ subscribed + magazine_releveled + resolved_releveled,  
data=TVService, link="probit"))
```

```
## formula:
```

```
## satisf_categorical ~ subscribed + magazine_releveled + resolved_releveled
```

```
## data: TVService
```

```
##
```

```
## link threshold nobs logLik AIC niter max.grad cond.H
```

```
## probit flexible 36 -46.69 107.39 4(0) 1.70e-09 5.5e+04
```

```
##
```

```
## Coefficients:
```

```
## Estimate Std. Error z value Pr(>|z|)
```

```
## subscribed 0.006061 0.005718 1.060 0.28919
```

```
## magazine_releveledno -1.202678 0.388476 -3.096 0.00196 **
```

```
## resolved_releveledno -0.880893 0.378313 -2.328 0.01989 *
```

```
## ---
```

```
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## Threshold coefficients:
```

```
## Estimate Std. Error z value
```

```
## 1|2 -2.6005 0.5384 -4.830
```

```
## 2|3 -1.7878 0.4709 -3.796
```

```
## 3|4 -1.0295 0.4411 -2.334
```

```
## 4|5 0.1114 0.3933 0.283
```

```
#Computations for AICC and BIC
```

```
p<-7
```

```

n<-36
print(AICC<- -2*logLik(fitted.model)+2*p*n/(n-p-1))

## 'log Lik.' 111.3854 (df=7)

BIC(fitted.model)

## [1] 118.47

#Checking to see how well the model fits using a deviance test
null.model<- clm(satisf_categorical ~ 1, data=TVService, link="probit")
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 15.51138 (df=4)

print(p.value<- pchisq(deviance, df=3, lower.tail = FALSE))

## 'log Lik.' 0.001427906 (df=4)

#Predicting P probabilities of each of the five levels of the satisfaction score for a caller who had been subscribed for 3 months, doesn't receive the magazine, and whose issue was resolved over the phone
print(predict(fitted.model, data.frame(subscribed=3, magazine_relevelled="no", resolved_relevelled="yes"), type="prob"))

## $fit
##           1           2           3           4           5
## 1 0.0783845 0.1947673 0.2884392 0.3408959 0.09751309

```

Cumulative Complementary Log Log: 4.1

```

#We start by importing the data
Admission <- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/Exercise Files/Exercise4.1Data.csv", header = TRUE, sep =",")

#Here we rescale the predictor GMAT and fit the cumulative cloglog model
GMAT_rescaled<- Admission$GMAT/100
summary(fitted.model<- clm(as.factor(status) ~ GPA + GMAT_rescaled, data=Admission, link="cloglog"))

## formula: as.factor(status) ~ GPA + GMAT_rescaled
## data: Admission
##
## link threshold nobs logLik AIC niter max.grad cond.H
## cloglog flexible 42 -29.50 66.99 5(0) 5.38e-07 1.3e+04
##
## Coefficients:
## Estimate Std. Error z value Pr(>|z|)
## GPA -1.6479 0.6316 -2.609 0.009081 **
## GMAT_rescaled -1.5190 0.4303 -3.530 0.000415 ***
## ---

```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##           Estimate Std. Error z value
## admit|border    -13.137      3.059  -4.294
## border|notadmit  -12.265      2.989  -4.104

#Computations for AICC and BIC
p<- 4
n<- 42
print(AICC<- -2*logLik(fitted.model)+2*p*n/(n-p-1))

## 'log Lik.' 68.07119 (df=4)

BIC(fitted.model)

## [1] 73.94079

#Now we check to see if the model fits using a deviance test
null.model<- clm(as.factor(status) ~ 1, data=Admission, link="cloglog")
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 27.14445 (df=2)

print(p.value<- pchisq(deviance, df=2, lower.tail = FALSE))

## 'log Lik.' 1.275435e-06 (df=2)

#Now we will use the fitted model to predict the probabilities of each admission status for someone with 3.1 GPA and GMAT score of 550
#print(predict(fitted.model, data.frame(GPA=3.1, GMAT_rescaled=5.50), type="prob"))
```

Cumulative Complementary Log Log: 4.2

```
#First we import the data and fit the cum CLogLog model
TVService <- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/Exercise Files/Exercise4.2Data.csv", header = TRUE, sep =",")

#Making the response satisfy a categorical variable and fitting the model
satisf_categorical<- as.factor(TVService$satisf)
summary(fitted.model<- clm(satisf_categorical ~ subscribed + magazine + resolved,
data=TVService, link="cloglog"))

## formula: satisf_categorical ~ subscribed + magazine + resolved
## data:      TVService
##
## link      threshold nobs logLik AIC      niter max.grad cond.H
## cloglog flexible  36   -48.37 110.73 6(0)  4.42e-12 5.2e+04
##
## Coefficients:
```

```
##           Estimate Std. Error z value Pr(>|z|)
## subscribed  0.002997  0.006339  0.473  0.63641
## magazineyes 1.142173  0.422566  2.703  0.00687 **
## resolvedyes 1.009893  0.430080  2.348  0.01887 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Threshold coefficients:
##           Estimate Std. Error z value
## 1|2  -1.4523      0.6499  -2.235
## 2|3  -0.3188      0.4857  -0.656
## 3|4   0.5528      0.4616   1.198
## 4|5   1.7226      0.4941   3.487

#Computations for AICC and BIC
p<-7
n<-36
print(AICC<- -2*logLik(fitted.model)+2*p*n/(n-p-1))

## 'log Lik.' 114.7302 (df=7)

BIC(fitted.model)

## [1] 121.8149

#Checking to see how well the model fits using a deviance test
null.model<- clm(satisf_categorical ~ 1, data=TVService, link="cloglog")
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 12.16653 (df=4)

print(p.value<- pchisq(deviance, df=3, lower.tail = FALSE))

## 'log Lik.' 0.006833923 (df=4)

#Predicting P probabilities of each of the five levels of the satisfaction score for a caller who had been subscribed for 3 months, doesn't receive the magazine, and whose issue was resolved over the phone
print(predict(fitted.model, data.frame(subscribed=3, magazine="no", resolved="yes"), type="prob"))

## $fit
##           1           2           3           4           5
## 1 0.0810142 0.1498225 0.2352053 0.4014798 0.1324782
```

Generalized Logit Model (for nominal response) 4.4

```
#First we import the data and fitting the generalized probit model
Forecast <- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/Exercise Files/Exercise4.4Data.csv", header = TRUE, sep = ",")

#Specifying reference categories and changing them into categorical variables
```

```

outcome_releveled <- relevel(as.factor(Forecast$outcome), ref="C")
water_releveled <- relevel(as.factor(Forecast$water), ref="yes")

#Running the model and checking to see how well it fits using a deviance test
summary(fitted.model<- multinom(outcome_releveled ~ elevation + winddir + wi
ndspeed +
water_releveled , data=Forecast))

## # weights:  18 (10 variable)
## initial  value 32.958369
## iter   10 value 17.771391
## iter   20 value 17.356931
## final   value 17.356923
## converged

## Call:
## multinom(formula = outcome_releveled ~ elevation + winddir +
##          windspeed + water_releveled, data = Forecast)
##
## Coefficients:
## (Intercept)  elevation      winddir windspeed water_releveledno
## FA   -7.764088  0.001089404  0.025033087  0.1116808          0.22092498
## FD   -5.218338  0.002986672  0.007571138  0.1421470          0.04887474
##
## Std. Errors:
## (Intercept)  elevation      winddir windspeed water_releveledno
## FA    3.582384  0.001953411  0.012024957  0.1756380          1.340582
## FD    3.049625  0.001807177  0.008640188  0.1962268          1.585012
##
## Residual Deviance: 34.71385
## AIC: 54.71385

#Deviance test
null.model<- multinom(outcome_releveled ~ 1, data=Forecast)

## # weights:  6 (2 variable)
## initial  value 32.958369
## final   value 30.288413
## converged

print(deviance<- deviance(null.model)-deviance(fitted.model))

## [1] 25.86298

print(p.value<- pchisq(deviance, df=8, lower.tail = FALSE))

## [1] 0.001108523

#Using the fitted model to predict the probabilities for an airport at 2000 f
eet, away from water, winf at 5 knots eastward.
print(predict(fitted.model, data.frame(elevation=2000, winddir=90, windspeed=

```

```
5,
water_releveled = "no"), type="prob"))

##           C           FA           FD
## 0.09933447 0.00773318 0.89293235
```

Generalized Logit Model (for nominal response) 4.5

```
#First we import the data and fitting the generalized probit model
AnkleCondition <- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT
410/Exercise Files/Exercise4.5Data.csv", header = TRUE, sep = ",")

#Specifying reference categories and changing them into categorical variables
condition_releveled<- relevel(as.factor(AnkleCondition$condition), ref="sprai
ned")
gender_releveled<- relevel(as.factor(AnkleCondition$gender), ref="female")

#Running the model and checking to see how well it fits using a deviance test
summary(fitted.model<- multinom(condition_releveled ~ age + gender_releveled,
data=AnkleCondition))

## # weights:  12 (6 variable)
## initial  value 27.465307
## iter   10 value 21.077309
## final   value 21.066688
## converged

## Call:
## multinom(formula = condition_releveled ~ age + gender_releveled,
##          data = AnkleCondition)
##
## Coefficients:
##          (Intercept)          age gender_releveledmale
## broken      -4.6658881   0.21077974           4.195560
## torn         0.3352406  -0.04544743           1.556776
##
## Std. Errors:
##          (Intercept)          age gender_releveledmale
## broken       3.162765  0.2023320           1.655647
## torn         1.679667  0.1379255           1.273358
##
## Residual Deviance: 42.13338
## AIC: 54.13338

#Deviance test
summary(null.model<- multinom(condition_releveled ~ 1, data=AnkleCondition))

## # weights:  6 (2 variable)
## initial  value 27.465307
```

```
## final value 27.300482
## converged

## Call:
## multinom(formula = condition_reveled ~ 1, data = AnkleCondition)
##
## Coefficients:
##             (Intercept)
## broken      0.2513144
## torn       0.2513144
##
## Std. Errors:
##             (Intercept)
## broken      0.5039526
## torn       0.5039526
##
## Residual Deviance: 54.60096
## AIC: 58.60096

print(deviance<- deviance(null.model)-deviance(fitted.model))

## [1] 12.46759

print(p.value<- pchisq(deviance, df=4, lower.tail = FALSE))

## [1] 0.01419266

#We will now use the fitted model to predict the probability of each type of injury for a 9 year old girl.
print(predict(fitted.model, data.frame(age=9, gender_reveled="female"), type="prob"))

##      sprained      broken      torn
## 0.50210809 0.03149866 0.46639325
```

Poisson 5.1

```
#First we begin by importing the data
DefectiveData <- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/Exercise Files/Exercise5.1Data.csv", header = TRUE, sep =",")

#Releveling shift and setting day as the reference
shift_reveled<- relevel(as.factor(DefectiveData$shift), ref="day")

#Fitting the model
summary(fitted.model<- glm(ndefectives ~ experience + shift_reveled,
data=DefectiveData, family=poisson(link=log)))

##
## Call:
## glm(formula = ndefectives ~ experience + shift_reveled, family = poisson
```



```

(link = log),
##      data = DefectiveData)
##
## Deviance Residuals:
##      Min        1Q      Median        3Q        Max
## -2.05694  -0.56800   0.05144   0.47579   2.04325
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      0.35714    0.33734   1.059  0.28973
## experience        0.03552    0.04709   0.754  0.45066
## shift_releveledevening 0.40813    0.31985   1.276  0.20195
## shift_releveledmorning 0.10090    0.33419   0.302  0.76270
## shift_relevelednight  0.90671    0.30630   2.960  0.00307 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 39.019  on 35  degrees of freedom
## Residual deviance: 27.731  on 31  degrees of freedom
## AIC: 128.62
##
## Number of Fisher Scoring iterations: 5

#Checking to see how well the model fits with a deviance test
null.model<- glm(ndefectives ~ 1, data=DefectiveData, family=poisson(link=log
))
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 11.28837 (df=1)

print(p.value<- pchisq(deviance, df=4, lower.tail = FALSE))

## 'log Lik.' 0.02350729 (df=1)

#Using the model to predict the number of defective items produced during a n
ight shift by an operator with six months of experience
print(predict(fitted.model, data.frame(experience=6, shift_releveled="night")
,
type="response"))

##      1
## 4.379627

```

Poisson 5.2

```

#First we begin by importing the data
CarInsuranceData <- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/ST
AT 410/Exercise Files/Exercise5.2Data.csv", header = TRUE, sep = ",")

```

```

#Releveling gender and setting day as the reference
gender_reveled<- relevel(as.factor(CarInsuranceData$gender), ref="F")

#Fitting the model
summary(fitted.model<- glm(naccidents ~ gender_reveled + age + miles,
data=CarInsuranceData, family=poisson(link=log)))

##
## Call:
## glm(formula = naccidents ~ gender_reveled + age + miles, family = poisson(link = log),
##      data = CarInsuranceData)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.63470  -0.83300   0.00054   0.67182   1.70119
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.449155   0.370804   1.211    0.226
## gender_reveledM 0.218899   0.160938   1.360    0.174
## age            0.017103   0.006726   2.543    0.011 *
## miles          -0.001283   0.001781  -0.721    0.471
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##      Null deviance: 70.348  on 47  degrees of freedom
## Residual deviance: 61.761  on 44  degrees of freedom
## AIC: 202.64
##
## Number of Fisher Scoring iterations: 5

#Checking to see how well the model fits with a deviance test
null.model<- glm(naccidents ~ 1, data=CarInsuranceData, family=poisson(link=log))
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 8.587217 (df=1)

print(p.value<- pchisq(deviance, df=3, lower.tail = FALSE))

## 'log Lik.' 0.03531361 (df=1)

#Using the model to predict the value of the total number of auto accidents caused by a 35-year-old woman who has driven a total of one hundred thousand miles
print(predict(fitted.model, data.frame(gender_reveled="F", age=35, miles=100000),
type="response"))

```

```
##          1
## 2.507913
```

Zero-Truncated Poisson 5.4

#First we begin by importing the data which is taken from 5.1

```
DefectiveData <- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT
410/Exercise Files/Exercise5.1Data.csv", header = TRUE, sep = ",")
```

#Getting rid of zeros in data set and fitting the model

```
DefectiveData<- DefectiveData[which(DefectiveData$ndefectives != 0),]
```

#Fitting the model

```
summary(fitted.model<- vglm(ndefectives ~ experience + shift,
data=DefectiveData, family = pospoisson()))
```

```
##
```

```
## Call:
```

```
## vglm(formula = ndefectives ~ experience + shift, family = pospoisson(),
##      data = DefectiveData)
```

```
##
```

```
## Coefficients:
```

```
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.19203    0.44108   0.435  0.66330
## experience   0.03355    0.05606   0.598  0.54951
## shiftevening 0.47641    0.40615   1.173  0.24080
## shiftmorning 0.36479    0.42558   0.857  0.39135
## shiftnight   1.06314    0.38115   2.789  0.00528 **
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## Name of linear predictor: loglink(lambda)
```

```
##
```

```
## Log-likelihood: -49.6747 on 28 degrees of freedom
```

```
##
```

```
## Number of Fisher scoring iterations: 4
```

```
##
```

```
## No Hauck-Donner effect found in any of the estimates
```

#Using a deviance test to check the fit of the model

```
null.model<- vglm(ndefectives ~ 1, data=DefectiveData, family = pospoisson())
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))
```

```
## [1] 9.904659
```

```
print((p.value<- pchisq(deviance, df=4, lower.tail = FALSE)))
```

```
## [1] 0.04206469
```

#Using the model to predict the number of defective items produced during a night shift by an operator with six months of experience.

```
print(predict(fitted.model, data.frame(experience=6, shift="night"),
type="response"))

##           [,1]
## [1,] 4.350423
```

Zero-Truncated Poisson 5.5

```
#First we begin by importing the data which is taken from 5.2
CarInsuranceData <- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/ST
AT 410/Exercise Files/Exercise5.2Data.csv", header = TRUE, sep = ",")

#Getting rid of zeros in data set and fitting the model
CarInsuranceData<- CarInsuranceData[which(CarInsuranceData$naccidents !=0),]

#Fitting the model
summary(fitted.model<- vglm(naccidents ~ gender + age + miles,
data=CarInsuranceData, family = pospoisson()))

##
## Call:
## vglm(formula = naccidents ~ gender + age + miles, family = pospoisson(),
##      data = CarInsuranceData)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.4869342  0.3981630   1.223   0.2213
## genderM      0.0826963  0.1705286   0.485   0.6277
## age          0.0157404  0.0072558   2.169   0.0301 *
## miles        -0.0002013  0.0019055  -0.106   0.9159
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Name of linear predictor: loglink(lambda)
##
## Log-likelihood: -84.0799 on 40 degrees of freedom
##
## Number of Fisher scoring iterations: 4
##
## No Hauck-Donner effect found in any of the estimates

#Using a deviance test to check the fit of the model
null.model<- vglm(naccidents ~ 1, data=CarInsuranceData, family = pospoisson(
))
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## [1] 5.087076

print(p.value<- pchisq(deviance, df=3, lower.tail = FALSE))

## [1] 0.1655309
```

```

#Using the model to predict the total number of auto accidents caused by a 35
-year-old woman who has driven a total of one hundred thousand miles
print(predict(fitted.model, data.frame(gender="F", age=35, miles=100),
type="response"))

##           [,1]
## [1,] 2.952453

```

Zero-Inflated Poisson 5.8

```

#First we begin by importing the data
Races<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/Exercise Files/Exercise5.8Data.csv", header = TRUE, sep = ",")

#Setting reference levels and turning run and gender into categorical variables
run_releveled<- relevel(as.factor(Races$run), ref="5K")
gender_relevled<- relevel(as.factor(Races$gender), ref="M")

#Fitting the model
summary(fitted.model<- zeroinfl(nraces ~ gender_relevled + age + run_releveled | pace,
data=Races))

##
## Call:
## zeroinfl(formula = nraces ~ gender_relevled + age + run_releveled | pace,
##      data = Races)
##
## Pearson residuals:
##      Min      1Q  Median      3Q      Max
## -1.4548 -0.7087 -0.2621  0.4742  2.1409
##
## Count model coefficients (poisson with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    1.63134    0.56313   2.897 0.003769 **
## gender_relevledF 1.02296    0.30360   3.369 0.000753 ***
## age           -0.04431    0.01478  -2.998 0.002719 **
## run_releveled10K  0.18539    0.36080   0.514 0.607359
## run_releveledFull 0.75468    0.32369   2.331 0.019727 *
##
## Zero-inflation model coefficients (binomial with logit link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -8.7064     3.9941  -2.18  0.0293 *
## pace         0.7255     0.3628   2.00  0.0455 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 15
## Log-likelihood: -51.02 on 7 Df

```

```

#Using a deviance test to check the fit of the model
null.model<- zeroinfl(nraces ~ 1, data=Races)
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 19.54518 (df=2)

print(p.value<- pchisq(deviance, df=5, lower.tail = FALSE))

## 'log Lik.' 0.001520639 (df=2)

#Using the fitted model to predict the number of races in the past four months for a female runner, aged 45, who ran at an average pace of 10 minutes per mile, if she ran 10K
print(predict(fitted.model, data.frame(gender_relevled="F", run_releveled="10K", age=45, pace=10)))

##          1
## 1.88771

```

Zero-Inflated Poisson 5.9

```

#First we begin by importing the data
Reading <- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/Exercise Files/Exercise5.9Data.csv", header = TRUE, sep =",")

#Setting reference levels and turning hw and gender into categorical variables
homework_releveled<- relevel(as.factor(Reading$hw), ref="no")
gender_relevled<- relevel(as.factor(Reading$gender), ref="M")

#Fitting the model
summary(fitted.model<- zeroinfl(nbooks ~ homework_releveled + gender_relevled | grade, data=Reading))

##
## Call:
## zeroinfl(formula = nbooks ~ homework_releveled + gender_relevled | grade, data = Reading)
##
## Pearson residuals:
##      Min      1Q   Median      3Q      Max
## -1.22436 -0.71216  0.01381  0.81085  1.68542
##
## Count model coefficients (poisson with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.63085    0.25643   2.460   0.0139 *
## homework_releveledyes 0.07149    0.22003   0.325   0.7453
## gender_relevledF    0.49764    0.24547   2.027   0.0426 *
##

```

```
## Zero-inflation model coefficients (binomial with logit link):
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.4649      1.0651   1.375   0.1690
## grade        -1.2981      0.5616  -2.312   0.0208 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 10
## Log-likelihood: -82.49 on 5 Df

#Using a deviance test to check the fit of the model
null.model<- zeroinfl(nbooks ~ 1, data=Reading)
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 11.00896 (df=2)

print(p.value<- pchisq(deviance, df=3, lower.tail = FALSE))

## 'log Lik.' 0.01167754 (df=2)

#Using the model to predict the number of books read by a second-grade girl f
or whom the reading is part of the homework
print(predict(fitted.model, data.frame(grade=2, gender_relevled="F", homework
_releveled="yes"))))

##           1
## 2.51027
```

Hurdle Poisson 5.12

```
#First we begin by importing the data
Librarianship <- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT
410/Exercise Files/Exercise5.12Data.csv", header = TRUE, sep = ",")

#Fitting the model
summary(fitted.model<- hurdle(ncomps ~ nbooks + njrnls | budget,
data=Librarianship, dist="poisson", zero.dist="binomial", link="logit"))

##
## Call:
## hurdle(formula = ncomps ~ nbooks + njrnls | budget, data = Librarianship,
##       dist = "poisson", zero.dist = "binomial", link = "logit")
##
## Pearson residuals:
##      Min      1Q  Median      3Q      Max
## -3.3994 -1.2434 -0.7700  0.9434  6.9410
##
## Count model coefficients (truncated poisson with log link):
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.733815    0.140691  19.431  <2e-16 ***
## nbooks       -0.025741    0.010592  -2.430   0.0151 *
```

```
## njrnls      0.001832  0.002425  0.755  0.4500
## Zero hurdle model coefficients (binomial with logit link):
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.4086      0.6625  -0.617  0.5374
## budget      0.4490      0.2174   2.065  0.0389 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 9
## Log-likelihood: -155.5 on 5 Df

#Using deviance test to see how well the model fits
null.model<- hurdle(ncomps ~ 1, data=Librarianship, dist="poisson",
zero.dist="binomial", link="logit")
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 19.21102 (df=2)

print(p.value<- pchisq(deviance, df=3, lower.tail = FALSE))

## 'log Lik.' 0.0002472598 (df=2)

#Using the fitted model to predict the number of computers in a library with
10,000 books, 25 periodicals, and an annual budget of $15 per student
print(predict(fitted.model, data.frame(nbooks=10, njrnls=25, budget=15)))

##           1
## 12.43384
```

Hurdle Poisson 5.13

```
#First we begin by importing the data
Medication<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/
Exercise Files/Exercise5.13Data.csv", header = TRUE, sep = ",")

#Releveling gender and making it a categorical variable
gender_reveled<- relevel(as.factor(Medication$gender), ref="M")

#Fitting the model
summary(fitted.model<- hurdle(ndaysnoms ~ gender_reveled + age | nothermeds,
data=Medication, dist="poisson", zero.dist = "binomial", link="logit"))

##
## Call:
## hurdle(formula = ndaysnoms ~ gender_reveled + age | nothermeds, data =
Medication,
##       dist = "poisson", zero.dist = "binomial", link = "logit")
##
## Pearson residuals:
##      Min      1Q  Median      3Q      Max
```



```
## -1.5558 -0.7444 0.1306 0.6857 2.0317
##
## Count model coefficients (truncated poisson with log link):
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.02120    1.48329  -0.688   0.4912
## gender_releveledF 0.72882    0.30489   2.390   0.0168 *
## age           0.02155    0.01992   1.082   0.2794
## Zero hurdle model coefficients (binomial with logit link):
##           Estimate Std. Error z value Pr(>|z|)
## (Intercept)    3.0554     1.0511   2.907   0.00365 **
## nothermeds     -0.2288     0.1122  -2.039   0.04144 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 9
## Log-likelihood: -52.23 on 5 Df

#Using deviance test to see how well the model fits
null.model<- hurdle(ndaysnomeds ~ 1, data=Medication, dist="poisson",
zero.dist="binomial", link="logit")
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 12.7902 (df=2)

print(p.value<- pchisq(deviance, df=3, lower.tail = FALSE))

## 'log Lik.' 0.005113002 (df=2)

#Using the fitted model to predict the number of computers in a library with
10,000 books, 25 periodicals, and an annual budget of $15 per student
print(predict(fitted.model, data.frame(gender_releveled="M", age=78, notherme
ds=0)))

##           1
## 2.158935
```

#Negative Binomial 6.2

```
#First we begin by importing the data
BioMussels<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/
Exercise Files/Exercise6.2Data.csv", header = TRUE, sep = ",")

#Specifying the reference level and making feeding level a categorical variab
le
FeedingLevel_releveled<- relevel(as.factor(BioMussels$feeding.level), ref="hi
gh")

#Fitting the model
summary(fitted.model<- glm.nb(ndeath.mussels ~ max.temp + min.temp + FeedingLe
vel_releveled, data=BioMussels))
```

```
##
## Call:
## glm.nb(formula = ndead.mussels ~ max.temp + min.temp + FeedingLevel_releve
led,
##      data = BioMussels, init.theta = 10.63638151, link = log)
##
## Deviance Residuals:
##      Min        1Q      Median        3Q        Max
## -1.78262  -0.93015  -0.05483   0.43714   1.68131
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -11.45070     3.98601  -2.873  0.00407 **
## max.temp         0.03026     0.02222   1.361  0.17340
## min.temp         0.14185     0.06925   2.048  0.04052 *
## FeedingLevel_releveledlow  1.77858     0.48825   3.643  0.00027 ***
## FeedingLevel_releveledmed  1.41255     0.48347   2.922  0.00348 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(10.6364) family taken to be 1)
##
##      Null deviance: 75.013  on 23  degrees of freedom
## Residual deviance: 23.855  on 19  degrees of freedom
## AIC: 104.34
##
## Number of Fisher Scoring iterations: 1
##
##
##              Theta: 10.6
##            Std. Err.: 12.0
##
## 2 x log-likelihood: -92.338

#Using deviance test to see how well the model fits
null.model<- glm.nb(ndeath.mussels ~ 1, data = BioMussels)
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 27.27491 (df=2)

print(p.value<- pchisq(deviance, df=4, lower.tail = FALSE))

## 'log Lik.' 1.749015e-05 (df=2)

#Using the model to predict the number of dead mussels that were fed a high l
evel of food, and were located in an area with a maximum temperature of 75 de
grees and minimum temperature of 60 degrees.
print(predict(fitted.model, data.frame(max.temp=75, min.temp=60,
FeedingLevel_releveled="high"), type="response"))
```

```
##          1
## 0.5113223
```

#Negative Binomial 6.3

#First we begin by importing the data

```
HSAallowance<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410
/Exercise Files/Exercise6.3Data.csv", header = TRUE, sep = ",")
```

#Specifying the reference level and making gender and job categorical variables

```
gender_relevelled<- relevel(as.factor(HSAallowance$gender), ref="M")
job_relevelled<- relevel(as.factor(HSAallowance$job), ref="yes")
```

#Fitting the model

```
summary(fitted.model<- glm.nb(allowance ~ age + gender_relevelled + job_relevelled,
data=HSAallowance))
```

```
##
```

```
## Call:
```

```
## glm.nb(formula = allowance ~ age + gender_relevelled + job_relevelled,
##       data = HSAallowance, init.theta = 3.29253741, link = log)
```

```
##
```

```
## Deviance Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -2.5167  -0.8701  -0.2815   0.1905   2.5725
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.1297     2.3982  -0.054  0.95688
## age             0.0359     0.1387   0.259  0.79570
## gender_relevelledF  0.4523     0.3067   1.475  0.14033
## job_relevelledno  1.2630     0.3937   3.208  0.00134 **
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## (Dispersion parameter for Negative Binomial(3.2925) family taken to be 1)
```

```
##
```

```
##      Null deviance: 48.685  on 29  degrees of freedom
```

```
## Residual deviance: 34.148  on 26  degrees of freedom
```

```
## AIC: 150.68
```

```
##
```

```
## Number of Fisher Scoring iterations: 1
```

```
##
```

```
##
```

```
##              Theta:  3.29
```

```
##            Std. Err.:  1.67
```

```
##
```

```
## 2 x log-likelihood:  -140.68
```

```

#Using deviance test to see how well the model fits
null.model<- glm.nb(allowance ~ 1, data=HSAallowance)
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 11.87264 (df=2)

print(p.value<- pchisq(deviance, df=3, lower.tail = FALSE))

## 'log Lik.' 0.00783244 (df=2)

#Using the model to predict the amount of weekly allowance for a male student
, age 16, who hasn't held a summer job.
print(predict(fitted.model, data.frame(age=16, gender_reveled="M", job_rele
veled="no"),
type="response"))

##          1
## 5.516843

```

#Zero-Truncated Negative Binomial 6.4

```

#First we begin by importing the data
StateParks<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/
Exercise Files/Exercise6.4Data.csv", header = TRUE, sep = ",")

#Specifying the reference level and making camped a categorical variable
camped_reveled<- relevel(as.factor(StateParks$camped), ref="yes")

#Fitting the model
summary(fitted.model<- vglm(nkayaks ~ partysize + routelength + camped_releve
led,
data=StateParks, family=posnegbinomial()))

##
## Call:
## vglm(formula = nkayaks ~ partysize + routelength + camped_reveled,
##      family = posnegbinomial(), data = StateParks)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept):1    -0.85770    0.55260  -1.552   0.121
## (Intercept):2     2.19308    1.15141     NA     NA
## partysize         0.18331    0.03440   5.329 9.89e-08 ***
## routelength       0.02644    0.03582   0.738   0.460
## camped_reveledno  0.27304    0.29855   0.915   0.360
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: loglink(munb), loglink(size)
##
## Log-likelihood: -48.2332 on 55 degrees of freedom

```

```
##
## Number of Fisher scoring iterations: 13
##
## Warning: Hauck-Donner effect detected in the following estimate(s):
## '(Intercept):2'

#Using deviance test to see how well the model fits
null.model<- vglm(nkayaks ~ 1, data=StateParks, family=posnegbinomial())
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## [1] 24.63644

print(p.value<- pchisq(deviance, df=3, lower.tail = FALSE))

## [1] 1.839315e-05

#Using the model to predict the number of rented kayaks for a party of 5 people who plan to take a 6-hour route and to camp overnight.
print(predict(fitted.model, data.frame(partysize=5, routelength=6, camped_releveled="yes"), type="response"))

##      [,1]
## 1 1.807214
```

#Zero-Truncated Negative Binomial 6.5

```
#First we begin by importing the data
Youtube<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/Exercise Files/Exercise6.5Data.csv", header = TRUE, sep = ",")

#Specifying the reference level and making type a categorical variable
type_releveled<- relevel(as.factor(Youtube$type), ref="fashion")

#Fitting the model
summary(fitted.model<- vglm(nnewvideos ~ nvideos + nsubscr + nviews + type_releveled,
data=Youtube, family = posnegbinomial()))

##
## Call:
## vglm(formula = nnewvideos ~ nvideos + nsubscr + nviews + type_releveled,
##      family = posnegbinomial(), data = Youtube)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept):1      0.3625437   0.3899071   0.930   0.35246
## (Intercept):2      0.7660984   0.4623516   1.657   0.09753 .
## nvideos            0.0049484   0.0016068   3.080   0.00207 **
## nsubscr            -0.0207244   0.0094039  -2.204   0.02754 *
## nviews             0.0009554   0.0010078   0.948   0.34314
## type_releveledcomedy 0.3336973   0.5483930   0.609   0.54286
```

```

## type_reveledlifeadvice  0.5772579  0.4939778  1.169  0.24257
## type_revelednews       1.5310712  0.5050049  3.032  0.00243 **
## type_reveledproducts   0.3359591  0.4596850  0.731  0.46487
## type_reveledscience    0.0101134  0.6663842  0.015  0.98789
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: loglink(munb), loglink(size)
##
## Log-likelihood: -95.9302 on 70 degrees of freedom
##
## Number of Fisher scoring iterations: 8
##
## No Hauck-Donner effect found in any of the estimates

#Using deviance test to see how well the model fits
null.model<- vglm(nnewvideos ~ 1, data = Youtube, family=posnegbinomial())
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## [1] 20.89688

print(p.value<- pchisq(deviance, df=8, lower.tail = FALSE))

## [1] 0.007426463

#Using the model to predict the number of new videos for a vlogger who posted a total
#of 87 videos on popular science, has 50,000 subscribers, and 254,000 views.
print(predict(fitted.model, data.frame(nvideos=87, nsubscr=50, nviews=254,
type_reveled="science"), type="response"))

##      [,1]
## 1 1.793339

#Zero-Inflated Negative Binomial 6.6

#First we begin by importing the data
NegBiInsurance<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT
410/Exercise Files/Exercise6.6Data.csv", header = TRUE, sep =",")

#Specifying the reference level and making gender a categorical variable
gender_reveled<- relevel(as.factor(NegBiInsurance$gender), ref="M")

#Fitting the model
summary(fitted.model<- zeroinfl(nclaimspast5ys ~ age + gender_reveled |
nclaimsprev5ys, data=NegBiInsurance, dist="negbin"))

##
## Call:
## zeroinfl(formula = nclaimspast5ys ~ age + gender_reveled | nclaimsprev5y
s,
##      data = NegBiInsurance, dist = "negbin")

```

```
##
## Pearson residuals:
##      Min      1Q  Median      3Q      Max
## -1.6473 -0.8728 -0.1159  0.5660  2.2521
##
## Count model coefficients (negbin with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.108008   0.536028  -0.201  0.84031
## age           0.029678   0.009213   3.221  0.00128 **
## gender_releveledF 0.127863  0.184205   0.694  0.48760
## Log(theta)    3.475375   2.344593   1.482  0.13826
##
## Zero-inflation model coefficients (binomial with logit link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.8826    0.7204   1.225  0.2205
## nclaimsprev5ys -1.3297    0.5397  -2.464  0.0138 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 32.3099
## Number of iterations in BFGS optimization: 10
## Log-likelihood: -82.02 on 6 Df

#Using deviance test to see how well the model fits
null.model<- zeroinfl(nclaimsplast5ys ~ 1, data=NegBiInsurance, dist="negbin")
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 20.2445 (df=3)

#Using the model to predict the number number of claims made in the past five years by a 55-year-old female
#policyholder who has made no claims in the previous five years.
print(predict(fitted.model, data.frame(nclaimsprev5ys=0, age=55,
gender_releveled="F")))

##      1
## 1.526963
```

#Zero-Inflated Negative Binomial 6.7

```
#First we begin by importing the data
Dentist<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/Exercise Files/Exercise6.7Data.csv", header = TRUE, sep =",")

#Specifying the reference level and making gender and oralhygiene categorical variables
gender_releveled<- relevel(as.factor(Dentist$gender), ref="F")
oralhygiene_releveled<- relevel(as.factor(Dentist$oralhygiene), ref="med")

#Fitting the model
summary(fitted.model<- zeroinfl(DMFTindex ~ gender_releveled + oralhygiene_re
```

```

leveled | age,
data=Dentist, dist="negbin"))

##
## Call:
## zeroinfl(formula = DMFTindex ~ gender_reveled + oralhygiene_reveled |
##   age, data = Dentist, dist = "negbin")
##
## Pearson residuals:
##      Min      1Q  Median      3Q      Max
## -1.0646 -0.7071 -0.1889  0.4202  1.9276
##
## Count model coefficients (negbin with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      1.2352     0.2762   4.472 7.75e-06 ***
## gender_reveledM      0.6655     0.3130   2.126  0.0335 *
## oralhygiene_reveledhigh 0.2468     0.3623   0.681  0.4957
## oralhygiene_reveledlow 0.1577     0.4125   0.382  0.7022
## Log(theta)        0.8514     0.4703   1.810  0.0702 .
##
## Zero-inflation model coefficients (binomial with logit link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.10729     1.42547   1.478  0.1393
## age         -0.10126     0.05073  -1.996  0.0459 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta = 2.3429
## Number of iterations in BFGS optimization: 13
## Log-likelihood: -88.86 on 7 Df

#Using deviance test to see how well the model fits
null.model<- zeroinfl(DMFTindex ~ 1, data=Dentist, dist="negbin")
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 12.76369 (df=3)

print(p.value<- pchisq(deviance, df=4, lower.tail = FALSE))

## 'log Lik.' 0.01249009 (df=3)

#Using the model to find the predicted value of the DMFT index for a man, age
d 28, with a high level of oral hygiene.
print(predict(fitted.model, data.frame(age=28, gender_reveled="M",
oralhygiene_reveled="high")))

##      1
## 5.77492

```

#Hurdle Negative Binomial 6.8


```

#First we begin by importing the data which is taken from 6.6
HNBInsurance<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 41
0/Exercise Files/Exercise6.6Data.csv", header = TRUE, sep = ",")

#Specifying the reference level and making gender a categorical variable
gender_releveled<- relevel(as.factor(HNBInsurance$gender), ref="M")

#Fitting the model
summary(fitted.model<- hurdle(nclaimspast5ys ~ age + gender_releveled | nclai
msprev5ys,
data=HNBInsurance, dist="negbin", zero.dist = "binomial", link="logit"))

##
## Call:
## hurdle(formula = nclaimspast5ys ~ age + gender_releveled | nclaimsprev5ys,
##       data = HNBInsurance, dist = "negbin", zero.dist = "binomial", link = "
logit")
##
## Pearson residuals:
##      Min      1Q   Median      3Q      Max
## -1.58030 -0.89629 -0.09889  0.56517  2.24100
##
## Count model coefficients (truncated negbin with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.187878   0.573901  -0.327  0.74339
## age           0.030928   0.009815   3.151  0.00163 **
## gender_releveledF 0.125341   0.193554   0.648  0.51726
## Log(theta)    3.111167   1.872968   1.661  0.09670 .
## Zero hurdle model coefficients (binomial with logit link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -0.8814     0.7050  -1.250  0.2112
## nclaimsprev5ys  1.2816     0.5071   2.527  0.0115 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta: count = 22.4472
## Number of iterations in BFGS optimization: 18
## Log-likelihood: -81.69 on 6 Df

#Using deviance test to see how well the model fits
null.model<- hurdle(nclaimspast5ys ~ 1, data=HNBInsurance, dist="negbin",
zero.dist="binomial", link="logit")
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 20.90545 (df=3)

print(p.value<- pchisq(deviance, df=3, lower.tail = FALSE))

## 'log Lik.' 0.0001101446 (df=3)

```

#Using the model to find the predicted number of claims made in the past five years by a 55-year-old female policyholder who has made no claims in the previous five years

```
print(predict(fitted.model, data.frame(nclaimsprev5ys=0, age=55,
gender_releveled="F")))
```

```
##          1
## 1.522423
```

#Hurdle Negative Binomial 6.9

#First we begin by importing the data

```
SportsMed<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/Exercise Files/Exercise6.9Data.csv", header = TRUE, sep = ",")
```

#Specifying the reference level and making gender a categorical variable

```
gender_releveled<- relevel(as.factor(SportsMed$gender), ref="F")
```

#Fitting the model

```
summary(fitted.model<- hurdle(ngameinjuries ~ gender_releveled +
nsports|npracticeinjuries, data=SportsMed, dist="negbin",
zero.dist="binomial", link="logit"))
```

```
##
## Call:
## hurdle(formula = ngameinjuries ~ gender_releveled + nsports | npracticeinjuries,
## data = SportsMed, dist = "negbin", zero.dist = "binomial", link = "logit")
##
## Pearson residuals:
##      Min       1Q   Median       3Q      Max
## -1.2907 -0.8023 -0.2172  0.4246  2.4498
##
## Count model coefficients (truncated negbin with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.72465    0.47606   1.522  0.12796
## gender_releveledM 0.88730    0.32510   2.729  0.00635 **
## nsports         0.08047    0.19854   0.405  0.68525
## Log(theta)      1.76429    0.89322   1.975  0.04825 *
## Zero hurdle model coefficients (binomial with logit link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.1574    0.8030  -1.441  0.1495
## npracticeinjuries 1.3498    0.5401   2.499  0.0124 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Theta: count = 5.8374
## Number of iterations in BFGS optimization: 17
## Log-likelihood: -59.46 on 6 Df
```

```

#Using deviance test to see how well the model fits
null.model<- hurdle(ngameinjuries ~ 1, data=SportsMed, dist="negbin",
zero.dist="binomial", link="logit")
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 21.23368 (df=3)

print(p.value<- pchisq(deviance, df=3, lower.tail = FALSE))

## 'log Lik.' 9.414351e-05 (df=3)

#Using the model to find the predicted number of injuries for a male athlete
who throughout his college years has participated in two sports, and who has
received one minor injury during practice games.
print(predict(fitted.model, data.frame(gender_releveled="M", nsports=2,
npracticeinjuries=1)))

##          1
## 3.282217

```

#Beta 7.2

```

Birds <- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/Exercise Files/Exercise7.2Data.csv", header = TRUE, sep = ",")

#Rescaling predictions according to the size of their observations and creating response
mass<- Birds$mass/1000
wingspan<- Birds$wingspan/100
distance<- Birds$distance/1000
propsuccess<- Birds$nmigrated/Birds$nringed

#Fitting the model
summary(fitted.model<- betareg(propsuccess ~ mass + wingspan + distance,
link="logit"))

##
## Call:
## betareg(formula = propsuccess ~ mass + wingspan + distance, link = "logit"
)
##
## Standardized weighted residuals 2:
##      Min      1Q  Median      3Q      Max
## -2.4465 -0.7549  0.0988  0.5438  2.3560
##
## Coefficients (mean model with logit link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.78304    0.78201   3.559 0.000373 ***
## mass        -0.01502    0.41088  -0.037 0.970848
## wingspan     0.04487    0.93935   0.048 0.961901
## distance    -1.31857    0.42237  -3.122 0.001797 **

```

```
##
## Phi coefficients (precision model with identity link):
##      Estimate Std. Error z value Pr(>|z|)
## (phi)    4.173      1.267   3.293 0.000991 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Type of estimator: ML (maximum likelihood)
## Log-likelihood: 6.406 on 5 Df
## Pseudo R-squared: 0.3677
## Number of iterations: 12 (BFGS) + 5 (Fisher scoring)

#Checking to see how well the model fits
null.model<- betareg(propsuccess ~ 1, link="logit")
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 8.55842 (df=2)

print(p.value<- pchisq(deviance, df=3, lower.tail = FALSE))

## 'log Lik.' 0.03577627 (df=2)

#Using the model to predict the number of birds that successfully reach the w
inter grounds for a flock of 70 birds with an average mass of 600 grams, an a
verage wingspan of 65 centimeters, that travel a distance of 1650 kilometers.
print(70*predict(fitted.model, data.frame(mass=.6, wingspan=.65, distance=1.6
5,
nringed=70)))

##      1
## 45.63551

#Beta 7.3

Hospitals <- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/
Exercise Files/Exercise7.3Data.csv", header = TRUE, sep = ",")

#Setting reference levels, changing to numeric, and creating response
prophospitalized<- Hospitals$perhospitalized/100
location.rel<- relevel(as.factor(Hospitals$location), ref="urban")
type.rel<- relevel(as.factor(Hospitals$type), ref="private")

#Fitting the model
summary(fitted.model<- betareg(prophospitalized ~ location.rel
+ type.rel + nbeds, data=Hospitals, link="logit"))

##
## Call:
## betareg(formula = prophospitalized ~ location.rel + type.rel + nbeds,
##      data = Hospitals, link = "logit")
##
## Standardized weighted residuals 2:
```

```
##      Min      1Q  Median      3Q      Max
## -2.9504 -0.7758  0.1150  0.9524  1.6070
##
## Coefficients (mean model with logit link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -1.6735254   0.2787233   -6.004 1.92e-09 ***
## location.relrural  0.5632707   0.2494671    2.258  0.02395 *
## type.relpublic    0.0116530   0.2511991    0.046  0.96300
## nbeds           0.0021170   0.0007219    2.933  0.00336 **
##
## Phi coefficients (precision model with identity link):
##              Estimate Std. Error z value Pr(>|z|)
## (phi)         9.808      2.462    3.984 6.78e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Type of estimator: ML (maximum likelihood)
## Log-likelihood: 20.79 on 5 Df
## Pseudo R-squared: 0.2884
## Number of iterations: 19 (BFGS) + 2 (Fisher scoring)

#Checking to see how well the model fits
null.model<- betareg(prophospitalized ~ 1, data=Hospitals, link="logit")
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 11.00244 (df=2)

print(p.value<- pchisq(deviance, df=3, lower.tail = FALSE))

## 'log Lik.' 0.01171267 (df=2)

#Using the model to predict the predicted proportion of hospitalized ER patients for a rural public hospital with 50 beds.
print(predict(fitted.model, data.frame(location.rel="rural", type.rel="public",
nbeds=50)))

##      1
## 0.270369

#Zero Inflated Beta 7.5

RealEstate <- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410
/Exercise Files/Exercise7.5Data.csv", header = TRUE, sep = ",")

#Recalcing predictors
RealEstate$propsold<- RealEstate$percsold/100
RealEstate$avgprice.res<- RealEstate$avgprice/100
RealEstate$nhouses.res<- RealEstate$nhouses/100

#Fitting the model
```

```

summary(fitted.model<- gamlss(propsold ~ avgprice.res + nhouses.res,
mu.link="logit", nu.formula = ~ age, nu.link="logit", data=RealEstate,
family = BEZI))

## GAMLSS-RS iteration 1: Global Deviance = 12.5187
## GAMLSS-RS iteration 2: Global Deviance = 11.0988
## GAMLSS-RS iteration 3: Global Deviance = 11.0973
## GAMLSS-RS iteration 4: Global Deviance = 11.0973
## *****
## Family:  c("BEZI", "Zero Inflated Beta")
##
## Call:  gamlss(formula = propsold ~ avgprice.res + nhouses.res,
##      nu.formula = ~age, family = BEZI, data = RealEstate,
##      mu.link = "logit", nu.link = "logit")
##
## Fitting method: RS()
##
## -----
## Mu link function:  logit
## Mu Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.16393    0.55373   2.102   0.0462 *
## avgprice.res  -0.20024    0.09321  -2.148   0.0420 *
## nhouses.res   -0.16791    0.24421  -0.688   0.4983
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -----
## Sigma link function:  log
## Sigma Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.703      0.273   6.238 1.9e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -----
## Nu link function:  logit
## Nu Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.51609    1.55696  -2.901  0.00785 **
## age          0.12468    0.05093   2.448  0.02205 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -----
## No. of observations in the fit:  30
## Degrees of Freedom for the fit:  6
##      Residual Deg. of Freedom:  24
##                      at cycle:  4
##

```

```
## Global Deviance:      11.09732
##           AIC:        23.09732
##           SBC:        31.50451
## *****

#Checking to see how well the model fits
null.model<- gamlss(propsold ~ 1, mu.link="logit", nu.formula= ~ 1,
nu.link="logit", data=RealEstate, family=BEZI)

## GAMLSS-RS iteration 1: Global Deviance = 25.2804
## GAMLSS-RS iteration 2: Global Deviance = 25.2182
## GAMLSS-RS iteration 3: Global Deviance = 25.2182

print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 14.12085 (df=3)

print(p.value<- pchisq(deviance, df=3, lower.tail = FALSE))

## 'log Lik.' 0.002745187 (df=3)

#Using the model to predict the percent houses sold for a subdivision with 30
0 houses, built 50 years ago, and where houses are sold, on average, for $450
,000
param.pred<- predictAll(fitted.model, newdata=data.frame(avgprice.res=4.5,
nhouses.res=3, age=50), type="response")
print((1-param.pred$nu)*param.pred$mu)

## [1] 0.06694997
```

#Zero Inflated Beta 7.6

```
MartialArts <- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 41
0/Exercise Files/Exercise7.6Data.csv", header = TRUE, sep = ",")

#Creating response variable
MartialArts$propfirst<- MartialArts$nfirstplaces/MartialArts$ntrophies

#Fitting the model
summary(fitted.model<- gamlss(propfirst ~ nyears + nblackbelts, mu.link="logi
t",
nu.formula = ~ npupils, nu.link="logit", data=MartialArts, family=BEZI))

## GAMLSS-RS iteration 1: Global Deviance = -0.7884
## GAMLSS-RS iteration 2: Global Deviance = -11.4115
## GAMLSS-RS iteration 3: Global Deviance = -11.578
## GAMLSS-RS iteration 4: Global Deviance = -11.5783
## *****
## Family:  c("BEZI", "Zero Inflated Beta")
##
## Call:  gamlss(formula = propfirst ~ nyears + nblackbelts,
##      nu.formula = ~npupils, family = BEZI, data = MartialArts,
```

```

##      mu.link = "logit", nu.link = "logit")
##
## Fitting method: RS()
##
## -----
## Mu link function:  logit
## Mu Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.94446    0.32335  -6.014 7.03e-06 ***
## nyears       0.12740    0.03938   3.235 0.00415 **
## nblackbelts  0.22669    0.10602   2.138 0.04502 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -----
## Sigma link function:  log
## Sigma Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.7057     0.3077   8.795 2.62e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -----
## Nu link function:  logit
## Nu Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.99552     1.79441   1.669   0.111
## npupils       -0.07133    0.03452  -2.066   0.052 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -----
## No. of observations in the fit:  26
## Degrees of Freedom for the fit:   6
##      Residual Deg. of Freedom:  20
##                      at cycle:   4
##
## Global Deviance:      -11.57825
##           AIC:         0.4217455
##           SBC:         7.970325
## *****

#Checking to see how well the model fits
null.model<- gamlss(propfirst ~ 1, mu.link="logit", nu.formula = ~ 1,
nu.link="logit", data=MartialArts, family=BEZI)

## GAMLSS-RS iteration 1: Global Deviance = 17.6774
## GAMLSS-RS iteration 2: Global Deviance = 15.8111
## GAMLSS-RS iteration 3: Global Deviance = 15.8073
## GAMLSS-RS iteration 4: Global Deviance = 15.8073

```



```

print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 27.38556 (df=3)

print(p.value<- pchisq(deviance, df=3, lower.tail = FALSE))

## 'log Lik.' 4.887314e-06 (df=3)

#Using the model to predict the proportion of first-place trophies won by a studio that has been around for 10 years, has 85 students and three black-belt instructors.
param.pred<- predictAll(fitted.model, newdata=data.frame(nyears=10,
nblackbelts=3, npupils=85), type="response")
print((1-param.pred$nu)*param.pred$mu)

## [1] 0.4800583

#One Inflated Beta 7.7

ParksAndRec <- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/Exercise Files/Exercise7.7Data.csv", header = TRUE, sep = ",")

#Creating response variable
ParksAndRec$propsurvived<- ParksAndRec$nsurvived/ParksAndRec$npplanted

#Fitting the model
summary(fitted.model<- gamlss(propsurvived ~ pestcontrol + fertilization,
mu.link="logit", nu.formula = ~ precipitation + windspeed, nu.link="logit",
data=ParksAndRec, family=BEOI))

## GAMLSS-RS iteration 1: Global Deviance = 8.5599
## GAMLSS-RS iteration 2: Global Deviance = 3.9565
## GAMLSS-RS iteration 3: Global Deviance = 3.7543
## GAMLSS-RS iteration 4: Global Deviance = 3.7504
## GAMLSS-RS iteration 5: Global Deviance = 3.7503
## *****
## Family: c("BEOI", "One Inflated Beta")
##
## Call: gamlss(formula = propsurvived ~ pestcontrol + fertilization,
## nu.formula = ~precipitation + windspeed, family = BEOI,
## data = ParksAndRec, mu.link = "logit", nu.link = "logit")
##
## Fitting method: RS()
##
## -----
## Mu link function: logit
## Mu Coefficients:
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.1080 0.5642 -1.964 0.0643 .
## pestcontrol 0.3233 0.1364 2.371 0.0285 *
## fertilization 0.7327 0.2821 2.597 0.0177 *

```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -----
## Sigma link function:  log
## Sigma Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.6048      0.3015   5.323 3.88e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -----
## Nu link function:  logit
## Nu Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.91004    4.29740   1.608  0.1243
## precipitation -0.04206    0.09075  -0.463  0.6483
## windspeed     -0.78224    0.37748  -2.072  0.0521 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -----
## No. of observations in the fit:  26
## Degrees of Freedom for the fit:  7
##      Residual Deg. of Freedom:  19
##                      at cycle:  5
##
## Global Deviance:      3.750299
##           AIC:        17.7503
##           SBC:        26.55697
## *****
```

#Checking to see how well the model fits

```
null.model<- gamlss(propsurvived ~ 1, mu.link="logit", nu.formula=~1,
nu.link="logit", data=ParksAndRec, family=BEOI)
```

```
## GAMLSS-RS iteration 1: Global Deviance = 23.3027
## GAMLSS-RS iteration 2: Global Deviance = 22.2044
## GAMLSS-RS iteration 3: Global Deviance = 22.1958
## GAMLSS-RS iteration 4: Global Deviance = 22.1958
```

```
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))
```

```
## 'log Lik.' 18.4455 (df=3)
```

```
print(p.value<- pchisq(deviance, df=4, lower.tail = FALSE))
```

```
## 'log Lik.' 0.001009668 (df=3)
```

#Using the model to predict the predicted proportion of trees that would survive for two years

```
param1<- predictAll(fitted.model, newdata = data.frame(pestcontrol=0,
fertilization=0, precipitation=2, windspeed=12.5), type="response")
param2<- predictAll(fitted.model, newdata = data.frame(pestcontrol=0,
fertilization=0, precipitation=25, windspeed=6), type="response")
print(param1$nu+(1-param1$nu)*param1$mu)
```

```
## [1] 0.2855535
```

```
print(param2$nu+(1-param2$nu)*param2$mu)
```

```
## [1] 0.8212778
```

#one Inflated Beta 7.8

```
LuggageSales <- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 4
10/Exercise Files/Exercise7.8Data.csv", header = TRUE, sep =",")
```

#Setting reference level and making gender numeric

```
LuggageSales$gender.rel<- relevel(as.factor(LuggageSales$gender), ref="M")
```

#Fitting the model

```
summary(fitted.model<- gamlss(propsales ~ gender.rel + bonus, mu.link="logit"
,
nu.formula = ~ expyr, nu.link="logit", data=LuggageSales, family=BEOI))
```

```
## GAMLSS-RS iteration 1: Global Deviance = -1.75
```

```
## GAMLSS-RS iteration 2: Global Deviance = -19.2634
```

```
## GAMLSS-RS iteration 3: Global Deviance = -22.4945
```

```
## GAMLSS-RS iteration 4: Global Deviance = -22.5962
```

```
## GAMLSS-RS iteration 5: Global Deviance = -22.5971
```

```
## *****
```

```
## Family: c("BEOI", "One Inflated Beta")
```

```
##
```

```
## Call: gamlss(formula = propsales ~ gender.rel + bonus, nu.formula = ~expyr,
## family = BEOI, data = LuggageSales, mu.link = "logit",
## nu.link = "logit")
```

```
##
```

```
## Fitting method: RS()
```

```
##
```

```
## -----
```

```
##
```

```
## Mu link function: logit
```

```
## Mu Coefficients:
```

```
## Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.3532 0.4533 -0.779 0.445044
## gender.relF -0.7023 0.2599 -2.702 0.013712 *
## bonus 1.8137 0.4038 4.492 0.000223 ***
```

```
## ---
```

```
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## -----
```

```
## Sigma link function: log
## Sigma Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.9540      0.3228   9.151 1.38e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -----
## Nu link function: logit
## Nu Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.7949      1.2964  -2.927  0.00833 **
## expyr        0.4619      0.1933   2.389  0.02684 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -----
## No. of observations in the fit:  26
## Degrees of Freedom for the fit:  6
##      Residual Deg. of Freedom:  20
##                      at cycle:  5
##
## Global Deviance:      -22.59715
##           AIC:        -10.59715
##           SBC:        -3.048571
## *****
```

#Checking to see how well the model fits

```
null.model<- gamlss(propsales ~ 1, mu.link="logit", nu.formula = ~ 1,
nu.link="logit", data=LuggageSales, family=BEOI)
```

```
## GAMLSS-RS iteration 1: Global Deviance = 17.0857
## GAMLSS-RS iteration 2: Global Deviance = 10.0505
## GAMLSS-RS iteration 3: Global Deviance = 9.5774
## GAMLSS-RS iteration 4: Global Deviance = 9.5682
## GAMLSS-RS iteration 5: Global Deviance = 9.568
```

```
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))
```

```
## 'log Lik.' 32.16519 (df=3)
```

```
print(p.value<- pchisq(deviance, df=3, lower.tail = FALSE))
```

```
## 'log Lik.' 4.830298e-07 (df=3)
```

#Using the model to predict the proportion of completed sales for a salesman with 3 years of work experience and who received \$1,500 in bonuses the previous year.

```
param<- predictAll(fitted.model, newdata = data.frame(expyr=3, gender.rel="M"
,
```

```
bonus=1.5), type="response")
print(param$nu+(1-param$nu)*param$mu)
```

```
## [1] 0.9213704
```

```
#Zero-One Inflated Beta 7.9
```

```
AgLab <- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/Exercise Files/Exercise7.9Data.csv", header = TRUE, sep = ",")
```

```
#Rescaling
```

```
AgLab$altitudeK<- AgLab$altitude/1000
```

```
#Fitting the model
```

```
summary(fitted.model<- gamlss(germrate ~ altitudeK, mu.link="logit", nu.formula
= ~ EC, nu.link="log", tau.formula = ~ soiltemp, tau.link="log", data=AgLab,
family=BEINF))
```

```
## GAMLSS-RS iteration 1: Global Deviance = 18.173
```

```
## GAMLSS-RS iteration 2: Global Deviance = 17.6438
```

```
## GAMLSS-RS iteration 3: Global Deviance = 17.639
```

```
## GAMLSS-RS iteration 4: Global Deviance = 17.639
```

```
## *****
```

```
## Family: c("BEINF", "Beta Inflated")
```

```
##
```

```
## Call: gamlss(formula = germrate ~ altitudeK, nu.formula = ~EC,
```

```
## tau.formula = ~soiltemp, family = BEINF, data = AgLab,
```

```
## mu.link = "logit", nu.link = "log", tau.link = "log")
```

```
##
```

```
## Fitting method: RS()
```

```
##
```

```
## -----
```

```
## Mu link function: logit
```

```
## Mu Coefficients:
```

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

```
## (Intercept) 1.6631 0.4713 3.529 0.00224 **
```

```
## altitudeK -0.4790 0.1262 -3.796 0.00122 **
```

```
## ---
```

```
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## -----
```

```
## Sigma link function: logit
```

```
## Sigma Coefficients:
```

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

```
## (Intercept) -0.5594 0.2370 -2.361 0.0291 *
```

```
## ---
```

```
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## -----
```

```
## Nu link function: log
```

```
## Nu Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) -13.716      6.161  -2.226  0.0383 *
## EC           5.863      2.579   2.273  0.0348 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -----
## Tau link function: log
## Tau Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) -24.8615    12.9583  -1.919  0.0702 .
## soiltemp      0.3599     0.1932   1.863  0.0780 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -----
## No. of observations in the fit: 26
## Degrees of Freedom for the fit: 7
##      Residual Deg. of Freedom: 19
##                      at cycle: 4
##
## Global Deviance:    17.63899
##           AIC:      31.63899
##           SBC:      40.44567
## *****
```

#Checking to see how well the model fits

```
null.model<- gamlss(germrate ~ 1, mu.link="logit", nu.formula = ~ 1,
nu.link="log", tau.formula = ~ 1, tau.link="log", data=AgLab, family=BEINF)
```

```
## GAMLSS-RS iteration 1: Global Deviance = 49.2481
## GAMLSS-RS iteration 2: Global Deviance = 49.2279
## GAMLSS-RS iteration 3: Global Deviance = 49.2278
```

```
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))
```

```
## 'log Lik.' 31.5888 (df=4)
```

```
print(p.value<- pchisq(deviance, df=3, lower.tail = FALSE))
```

```
## 'log Lik.' 6.389023e-07 (df=4)
```

#Using the fitted model to predict the germination rate for a plot with EC of 1.5 mS/cm2, soil temperature of 68F, and altitude of 950 feet.

```
param<- predictAll(fitted.model, newdata = data.frame(EC=1.5, soiltemp=68,
altitudeK=0.95), type="response")
print((param$tau+param$mu)/(1+param$nu+param$tau))
```

```
## [1] 0.8593549
```

#Zero-One Inflated Beta 7.10

```
CollegeFootball <- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STA  
T 410/Exercise Files/Exercise7.10Data.csv", header = TRUE, sep = ",")
```

```
#Fitting the model
```

```
summary(fitted.model<- gamlss(propgames ~ vertical + bench, mu.link="logit",  
nu.formula = ~ broad, nu.link="log",tau.formula = ~ BMI + fortyyd, tau.link  
="log", data=CollegeFootball, family=BEINF))
```

```
## GAMLSS-RS iteration 1: Global Deviance = 15.291  
## GAMLSS-RS iteration 2: Global Deviance = 13.6346  
## GAMLSS-RS iteration 3: Global Deviance = 13.6244  
## GAMLSS-RS iteration 4: Global Deviance = 13.6244  
## *****  
## Family: c("BEINF", "Beta Inflated")  
##  
## Call: gamlss(formula = propgames ~ vertical + bench, nu.formula = ~broad,  
## tau.formula = ~BMI + fortyyd, family = BEINF, data = CollegeFootball,  
## mu.link = "logit", nu.link = "log", tau.link = "log")  
##  
## Fitting method: RS()  
##  
## -----  
## Mu link function: logit  
## Mu Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -6.33600 2.44008 -2.597 0.01395 *  
## vertical 0.21407 0.06938 3.086 0.00409 **  
## bench -0.01399 0.03066 -0.456 0.65117  
## ---  
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## -----  
## Sigma link function: logit  
## Sigma Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -0.7429 0.1665 -4.463 8.9e-05 ***  
## ---  
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## -----  
## Nu link function: log  
## Nu Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 45.5488 18.5894 2.450 0.0197 *  
## broad -0.4148 0.1652 -2.511 0.0171 *  
## ---  
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## -----  
## Tau link function: log
```

```
## Tau Coefficients:
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) -27.0077    26.3530  -1.025   0.3129
## BMI         -1.0664     0.4808  -2.218   0.0336 *
## fortyyd      12.4225     5.7375   2.165   0.0377 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## -----
## No. of observations in the fit: 42
## Degrees of Freedom for the fit: 9
##      Residual Deg. of Freedom: 33
##              at cycle: 4
##
## Global Deviance:    13.62437
##           AIC:      31.62437
##           SBC:      47.26339
## *****
```

#Checking to see how well the model fits

```
null.model<- gamlss(propgames ~ 1, mu.link="logit", nu.formula = ~ 1,
nu.link="log", tau.formula = ~ 1, tau.link="log", data=CollegeFootball,
family= BEINF)
```

```
## GAMLSS-RS iteration 1: Global Deviance = 47.9109
## GAMLSS-RS iteration 2: Global Deviance = 47.2882
## GAMLSS-RS iteration 3: Global Deviance = 47.2861
## GAMLSS-RS iteration 4: Global Deviance = 47.2861
```

```
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))
```

```
## 'log Lik.' 33.66176 (df=4)
```

```
print(p.value<- pchisq(deviance, df=5, lower.tail = FALSE))
```

```
## 'log Lik.' 2.780153e-06 (df=4)
```

#sing the fitted model to predict the proportion of games that a new player will play, if his BMI 27.8 kg/m2, forty-dash run is 4.67 seconds, the vertical jump is 32 inches, broad jump is 117 inches, and bench press is 16 repetition s.

```
param<- predictAll(fitted.model, newdata = data.frame(BMI=27.8, fortyyd=4.67,
vertical=32, broad=117, bench=16), type="response")
print((param$tau+param$mu)/(1+param$nu+param$tau))
```

```
## [1] 0.9032484
```

#Random Slope and Intercept for Normal Response 8.2

```
DepartmentStore<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT
410/Exercise Files/Exercise8.2Data.csv", header = TRUE, sep =",")
```



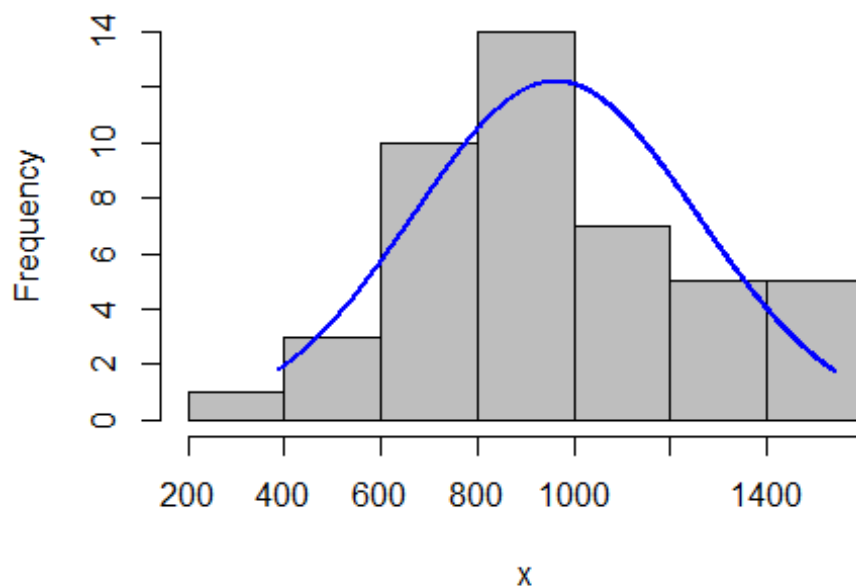
```

#Creating a Long form data set
LongForm<- melt(DepartmentStore, id.vars=c("id", "totalyears", "status"),
variable.name="bonus.year", value.name="bonus")

#Creating response variable
year<- ifelse(LongForm$bonus.year=="bonus18",1.8,
ifelse(LongForm$bonus.year=="bonus19",1.9,2.0))

#Checking normality of reponse variable
plotNormalHistogram(LongForm$bonus)

```



```

shapiro.test(LongForm$bonus)

##
##  Shapiro-Wilk normality test
##
## data:  LongForm$bonus
## W = 0.96686, p-value = 0.222

#Creating reference level for status
status.rel<- relevel(as.factor(LongForm$status), ref="part")

#fitting the model
summary(fitted.model<- lme(bonus ~ totalyears + status.rel + year,
random = ~ 1 + year | id, data=LongForm))

```

```

## Linear mixed-effects model fit by REML
##   Data: LongForm
##       AIC      BIC    logLik
##  529.3133 543.0219 -256.6567
##
## Random effects:
##   Formula: ~1 + year | id
##   Structure: General positive-definite, Log-Cholesky parametrization
##              StdDev     Corr
## (Intercept) 1499.85877 (Intr)
## year        797.95782 -0.998
## Residual    67.88526
##
## Fixed effects: bonus ~ totalyears + status.reel + year
##              Value Std.Error DF   t-value p-value
## (Intercept) -2246.1138  457.2023 29 -4.912735  0.000
## totalyears   58.8982   8.2368 12  7.150631  0.000
## status.reelfull 53.9643  63.3473 12  0.851879  0.411
## year        1394.6667  240.4381 29  5.800523  0.000
## Correlation:
##              (Intr) ttlyrs stts.r
## totalyears   -0.111
## status.reelfull 0.013 -0.600
## year        -0.990  0.000  0.000
##
## Standardized Within-Group Residuals:
##              Min      Q1      Med      Q3      Max
## -1.55954603 -0.32012837 -0.04131555  0.31523299  1.89915223
##
## Number of Observations: 45
## Number of Groups: 15

intervals(fitted.model)

## Approximate 95% confidence intervals
##
## Fixed effects:
##              lower      est.      upper
## (Intercept) -3181.19760 -2246.11380 -1311.03001
## totalyears   40.95181   58.89822   76.84464
## status.reelfull -84.05771   53.96428   191.98628
## year        902.91556  1394.66667  1886.41777
## attr(,"label")
## [1] "Fixed effects:"
##
## Random Effects:
##   Level: id
##              lower      est.      upper
## sd((Intercept)) 891.8860755 1499.8587688 2522.2686935
## sd(year)        476.3219559  797.9578238 1336.7779518

```

```
## cor((Intercept),year)  -0.9995787  -0.9983042  -0.9931881
##
## Within-group standard error:
##   lower    est.    upper
## 47.49499 67.88526 97.02937

#Checking to see how well the model fits
null.model<- glm(bonus ~ totalyears + status.rel + year, data=LongForm)
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 45.81206 (df=5)

print(p.value<- pchisq(deviance, df=3, lower.tail = FALSE))

## 'log Lik.' 6.218073e-10 (df=5)

#Using fitted model to predict the predicted bonus in 2021 for a full-time employee who has been with the company for 7 years
#print(predict(fitted.model, data.frame(status.rel="full", totalyears=7, year=2.1), level=0))
```

#Random Slope and Intercept for Normal Response 8.3

```
Orthopedic<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/Exercise Files/Exercise8.3Data.csv", header = TRUE, sep =",")

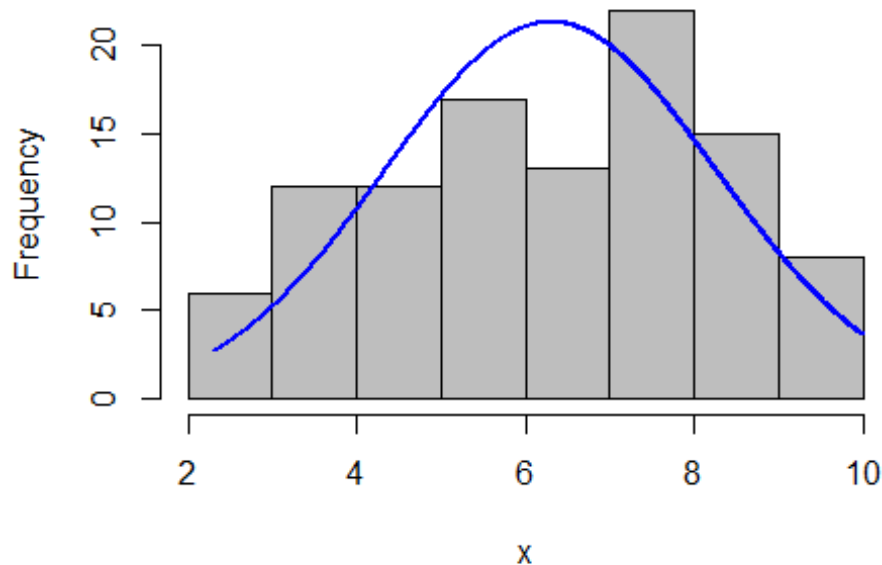
#Creating a Long form data set
data1<- melt(Orthopedic[,c("id","gender","age","doctor","length1","length2","length3")],id.vars=c("id","gender","age","doctor"),variable.name="length.visit",value.name="length")
data2<- melt(Orthopedic[,c("score1","score2","score3")],variable.name="score.visit", value.name="score")

## No id variables; using all as measure variables

Longform<- cbind(data1,data2)

#Creating response variable
visit<- ifelse(Longform$score.visit=="score1",1,ifelse(Longform$score.visit=="score2",2,3))

#Checking normality of reponse variable
plotNormalHistogram(Longform$score)
```



```
shapiro.test(Longform$score)

##
##  Shapiro-Wilk normality test
##
## data:  Longform$score
## W = 0.97598, p-value = 0.05332

#Creating reference levels
gender.rel<- relevel(as.factor(Longform$gender), ref="M")
doctor.rel<- relevel(as.factor(Longform$doctor), ref="B")

#fitting the model
summary(fitted.model<- lme(score ~ gender.rel + age + doctor.rel
+ length + visit, random = ~ 1 + visit | id, data=Longform))

## Linear mixed-effects model fit by REML
## Data: Longform
##      AIC      BIC    logLik
## 271.0585 297.0097 -125.5293
##
## Random effects:
## Formula: ~1 + visit | id
## Structure: General positive-definite, Log-Cholesky parametrization
##              StdDev    Corr
## (Intercept) 1.5090310 (Intr)
## visit       0.3709390 -0.245
```

```

## Residual    0.2852501
##
## Fixed effects:  score ~ gender.rel + age + doctor.rel + length + visit
##               Value Std.Error DF   t-value p-value
## (Intercept)  5.221580 2.5707920 68   2.031117  0.0462
## gender.relF   2.084700 0.5198854 31   4.009921  0.0004
## age          0.002997 0.0385434 31   0.077755  0.9385
## doctor.relA   0.158469 0.5084557 31   0.311668  0.7574
## length       -0.010510 0.0047844 68  -2.196820  0.0314
## visit        -0.046100 0.0714364 68  -0.645336  0.5209
## Correlation:
##           (Intr) gndr.F age    dctr.A length
## gender.relF  0.033
## age         -0.979 -0.166
## doctor.relA -0.213  0.086  0.095
## length      -0.083  0.114  0.014  0.042
## visit       -0.030 -0.005 -0.001 -0.002 -0.043
##
## Standardized Within-Group Residuals:
##           Min           Q1           Med           Q3           Max
## -1.89038162 -0.46995687  0.01261292  0.40384503  1.67790854
##
## Number of Observations: 105
## Number of Groups: 35

intervals(fitted.model)

## Approximate 95% confidence intervals
##
## Fixed effects:
##           lower           est.           upper
## (Intercept)  0.09164411  5.221579728 10.3515153426
## gender.relF   1.02438617  2.084699512  3.1450128591
## age          -0.07561292  0.002996939  0.0816068002
## doctor.relA  -0.87853313  0.158469127  1.1954713812
## length       -0.02005743 -0.010510384 -0.0009633373
## visit        -0.18864960 -0.046100468  0.0964486669
## attr(,"label")
## [1] "Fixed effects:"
##
## Random Effects:
## Level: id
##           lower           est.           upper
## sd((Intercept))  1.1473569  1.5090310  1.9847134
## sd(visit)        0.2705064  0.3709390  0.5086597
## cor((Intercept),visit) -0.5845323 -0.2446523  0.1682708
##
## Within-group standard error:
##           lower           est.           upper
## 0.2246113  0.2852501  0.3622595

```

```

#Checking to see how well the model fits
null.model<- glm(score ~ gender.rel + age + doctor.rel + length + visit,
data=Longform)
print(deviance<- -2*(logLik(null.model)-logLik(fitted.model)))

## 'log Lik.' 72.63075 (df=7)

print(p.value<- pchisq(deviance, df=3, lower.tail = FALSE))

## 'log Lik.' 1.166255e-15 (df=7)

#Using fitted model to predict the quality of service score that would be given by a 55-year-old male patient on his fourth visit to Dr. A with a 30-minute appointment.
#print(predict(fitted.model, data.frame(id=136,gender.rel="M",age=55,doctor.rel="A",length=30,visit=4),level=0))

```

#Random Slope and Intercept with Variance Structure for Error 8.6

```

#Uses data from 8.2
DepartmentStore<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/Exercise Files/Exercise8.2Data.csv", header = TRUE, sep =",")

#Creating Longform data
Longform<- melt(DepartmentStore, id.vars=c("id","totalyears","status"),
variable.name="bonus.year", value.name="bonus")

#Creating variables
year<- ifelse(Longform$bonus.year=="bonus18",1.8,
ifelse(Longform$bonus.year=="bonus19",1.9,2.0))

#Rescaling and reference
status.rel<- relevel(as.factor(Longform$status), ref="part")

#Fitting random slope and intercept model with unstructured covariance matrix of error terms
summary(un.fitted.model<- lme(bonus ~ totalyears + status.rel + year,
random = ~ 1 + year | id, data=Longform, correlation=corSymm(),
weights=varIdent(form = ~ id | year)))

## Linear mixed-effects model fit by REML
##   Data: Longform
##       AIC      BIC    logLik
##   529.393 551.6694 -251.6965
##
## Random effects:
##   Formula: ~1 + year | id
##   Structure: General positive-definite, Log-Cholesky parametrization
##              StdDev   Corr
## (Intercept) 127.86136 (Intr)
##   year       72.16169 -0.815

```

```

## Residual    108.15388
##
## Correlation Structure: General
## Formula: ~1 | id
## Parameter estimate(s):
## Correlation:
## 1      2
## 2 -0.247
## 3 -0.616  0.735
## Variance function:
## Structure: Different standard deviations per stratum
## Formula: ~id | year
## Parameter estimates:
##      1.8      1.9      2
## 1.0000000 0.8826214 0.8955668
## Fixed effects:  bonus ~ totalyears + status.rel + year
##
##              Value Std.Error DF   t-value p-value
## (Intercept)  -2032.3403  338.3799 29  -6.006090  0.0000
## totalyears      61.1977   7.9297 12   7.717549  0.0000
## status.relfull  40.7302   60.9854 12   0.667868  0.5169
## year          1272.6088  174.2924 29   7.301573  0.0000
## Correlation:
##              (Intr) ttlyrs stts.r
## totalyears    -0.145
## status.relfull  0.017 -0.600
## year          -0.983  0.000  0.000
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -2.02149053 -0.42338681 -0.05195106  0.66991469  2.11525768
##
## Number of Observations: 45
## Number of Groups: 15

#AICC
n<- 45
p<- 14
print(AICC<- -2*logLik(un.fitted.model)+2*p*n/(n-p-1))

## 'log Lik.' 545.393 (df=13)

#Fitting random slope and intercept model with Toeplitz covariance matrix of
error terms
summary(toep.fitted.model<- lme(bonus ~ totalyears + status.rel + year,
random = ~ 1 + year| id, data=Longform,
correlation=corARMA(form = ~ 1 | id, p=1, q=1)))

## Linear mixed-effects model fit by REML
## Data: Longform
##      AIC      BIC    logLik
## 532.5445 549.6802 -256.2722

```

```

##
## Random effects:
## Formula: ~1 + year | id
## Structure: General positive-definite, Log-Cholesky parametrization
##           StdDev      Corr
## (Intercept) 4.092560e-02 (Intr)
## year        2.105377e-05 0
## Residual    1.299681e+02
##
## Correlation Structure: ARMA(1,1)
## Formula: ~1 | id
## Parameter estimate(s):
##      Phi1      Theta1
## 0.09194164 0.99999473
## Fixed effects: bonus ~ totalyears + status.rel + year
##           Value Std.Error DF   t-value p-value
## (Intercept) -2263.5843  443.8512 29 -5.099872  0.0000
## totalyears    60.9362    8.0566 12  7.563516  0.0000
## status.relfull  42.5230   61.9615 12  0.686280  0.5056
## year         1394.6667  231.2558 29  6.030839  0.0000
## Correlation:
##           (Intr) ttlyrs stts.r
## totalyears   -0.112
## status.relfull 0.013 -0.600
## year         -0.990  0.000  0.000
##
## Standardized Within-Group Residuals:
##           Min      Q1      Med      Q3      Max
## -1.7345232 -0.7151969 -0.0438371  0.8018594  1.9381448
##
## Number of Observations: 45
## Number of Groups: 15

#AICC
p<-11
print(AICC<- -2*logLik(toep.fitted.model)+2*p*n/(n-p-1))

## 'log Lik.' 542.5445 (df=10)

#Fitting random slope and intercept model with spatial power covariance matrix of error terms
summary(sppow.fitted.model<- lme(bonus ~ totalyears + status.rel + year,
random = ~ 1 + year | id, data=Longform,
correlation=corCAR1(form = ~ 1 | id)))

## Linear mixed-effects model fit by REML
## Data: Longform
##      AIC      BIC    logLik
## 532.8405 548.2627 -257.4203
##
## Random effects:

```



```

## Formula: ~1 + year | id
## Structure: General positive-definite, Log-Cholesky parametrization
##           StdDev      Corr
## (Intercept) 1.682603e-02 (Intr)
## year        3.010918e-03 0
## Residual    1.309026e+02
##
## Correlation Structure: Continuous AR(1)
## Formula: ~1 | id
## Parameter estimate(s):
##      Phi
## 0.5467992
## Fixed effects: bonus ~ totalyears + status.rel + year
##           Value Std.Error DF   t-value p-value
## (Intercept) -2253.4538  386.4705 29 -5.830856  0.0000
## totalyears    59.8379   8.9004 12  6.723065  0.0000
## status.relfull 48.2277  68.4510 12  0.704558  0.4945
## year        1394.6667 200.1013 29  6.969804  0.0000
## Correlation:
##           (Intr) ttlyrs stts.r
## totalyears -0.142
## status.relfull 0.017 -0.600
## year        -0.984  0.000  0.000
##
## Standardized Within-Group Residuals:
##           Min      Q1      Med      Q3      Max
## -1.7491915 -0.7551994 -0.0889853  0.7951945  1.8956271
##
## Number of Observations: 45
## Number of Groups: 15

#AICC
p<- 10
print(AICC<- -2*logLik(sppow.fitted.model)+2*p*n/(n-p-1))

## 'log Lik.' 541.3111 (df=9)

#Fitting random intercept-only model with spatial power covariance matrix of
error terms
summary(sppowint.fitted.model<- lme(bonus ~ totalyears + status.rel + year,
random = ~ 1 | id, data=Longform,
correlation=corCAR1(form = ~ 1 | id)))

## Linear mixed-effects model fit by REML
## Data: Longform
##           AIC      BIC    logLik
## 528.8405 540.8355 -257.4203
##
## Random effects:
## Formula: ~1 | id
##           (Intercept) Residual

```

```

## StdDev:  0.01464535 130.9026
##
## Correlation Structure: Continuous AR(1)
## Formula: ~1 | id
## Parameter estimate(s):
##      Phi
## 0.5467992
## Fixed effects:  bonus ~ totalyears + status.rel + year
##               Value Std.Error DF   t-value p-value
## (Intercept)  -2253.4538  386.4705  29 -5.830856  0.0000
## totalyears      59.8379   8.9004  12  6.723065  0.0000
## status.rell    48.2277  68.4510  12  0.704558  0.4945
## year          1394.6667 200.1013  29  6.969804  0.0000
## Correlation:
##           (Intr) ttlyrs stts.r
## totalyears  -0.142
## status.rell  0.017 -0.600
## year        -0.984  0.000  0.000
##
## Standardized Within-Group Residuals:
##           Min           Q1           Med           Q3           Max
## -1.74919151 -0.75519942 -0.08898529  0.79519451  1.89562705
##
## Number of Observations: 45
## Number of Groups: 15

#AICC
p<- 8
print(AICC<- -2*logLik(sppowint.fitted.model)+2*p*n/(n-p-1))

## 'log Lik.' 534.8405 (df=7)

#Fitting random slope and intercept model with autoregressive covariance matrix of error terms
summary(ar.fitted.model<- lme(bonus ~ totalyears + status.rel + year,
random = ~ 1 + year | id, data=Longform,
correlation=corAR1(form = ~ 1 | id)))

## Linear mixed-effects model fit by REML
## Data: Longform
##      AIC      BIC    logLik
## 532.8405 548.2627 -257.4203
##
## Random effects:
## Formula: ~1 + year | id
## Structure: General positive-definite, Log-Cholesky parametrization
##           StdDev      Corr
## (Intercept) 0.02036143 (Intr)
## year         0.00073145 0
## Residual    130.90256777
##

```

```

## Correlation Structure: AR(1)
## Formula: ~1 | id
## Parameter estimate(s):
##      Phi
## 0.5467992
## Fixed effects:  bonus ~ totalyears + status.rel + year
##               Value Std.Error DF   t-value p-value
## (Intercept)  -2253.4538  386.4705 29  -5.830856  0.0000
## totalyears     59.8379   8.9004 12   6.723065  0.0000
## status.rell    48.2277  68.4510 12   0.704558  0.4945
## year          1394.6667 200.1013 29   6.969804  0.0000
## Correlation:
##      (Intr) ttlyrs stts.r
## totalyears  -0.142
## status.rell  0.017 -0.600
## year        -0.984  0.000  0.000
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -1.7491915 -0.7551994 -0.0889853  0.7951945  1.8956271
##
## Number of Observations: 45
## Number of Groups: 15

#AICC
p<- 10
print(AICC<- -2*logLik(ar.fitted.model)+2*p*n/(n-p-1))

## 'log Lik.' 541.3111 (df=9)

#Fitting random slope and intercept model with compound symmetric covariance
matrix of error terms
summary(cs.fitted.model<- lme(bonus ~ totalyears + status.rel + year,
random = ~ 1 + year | id, data=Longform,
correlation=corCompSymm(form = ~ 1 | id)))

## Linear mixed-effects model fit by REML
## Data: Longform
##      AIC      BIC    logLik
## 531.3133 546.7355 -256.6567
##
## Random effects:
## Formula: ~1 + year | id
## Structure: General positive-definite, Log-Cholesky parametrization
##      StdDev    Corr
## (Intercept) 1499.84438 (Intr)
## year        797.95695 -0.998
## Residual    68.15637
##
## Correlation Structure: Compound symmetry
## Formula: ~1 | id

```

```

## Parameter estimate(s):
##      Rho
## 0.007941571
## Fixed effects:  bonus ~ totalyears + status.rel + year
##               Value Std.Error DF   t-value p-value
## (Intercept)  -2246.1138  457.2018 29  -4.912741  0.000
## totalyears     58.8982    8.2368 12   7.150613  0.000
## status.rell    53.9643   63.3475 12   0.851878  0.411
## year          1394.6667  240.4378 29   5.800529  0.000
## Correlation:
##               (Intr) ttlyrs stts.r
## totalyears    -0.111
## status.rellfull 0.013 -0.600
## year          -0.990  0.000  0.000
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -1.55566258 -0.31195060 -0.04880253  0.30786515  1.89838030
##
## Number of Observations: 45
## Number of Groups: 15

#AICC
p<- 10
print(AICC<- -2*logLik(cs.fitted.model)+2*p*n/(n-p-1))

## 'log Lik.' 539.7839 (df=9)

#Fitting random slope and intercept model with independent covariance matrix
of error terms
summary(ind.fitted.model<- lme(bonus ~ totalyears + status.rel+ year,
random = ~ 1 + year | id, data=Longform))

## Linear mixed-effects model fit by REML
##   Data: Longform
##      AIC      BIC    logLik
## 529.3133 543.0219 -256.6567
##
## Random effects:
## Formula: ~1 + year | id
## Structure: General positive-definite, Log-Cholesky parametrization
##              StdDev      Corr
## (Intercept) 1499.85877 (Intr)
## year          797.95782 -0.998
## Residual      67.88526
##
## Fixed effects:  bonus ~ totalyears + status.rel + year
##               Value Std.Error DF   t-value p-value
## (Intercept)  -2246.1138  457.2023 29  -4.912735  0.000
## totalyears     58.8982    8.2368 12   7.150631  0.000
## status.rellfull 53.9643   63.3473 12   0.851879  0.411

```

```

## year          1394.6667  240.4381 29  5.800523  0.000
## Correlation:
##              (Intr) ttlyrs stts.r
## totalyears   -0.111
## status.relfull 0.013 -0.600
## year         -0.990  0.000  0.000
##
## Standardized Within-Group Residuals:
##           Min           Q1           Med           Q3           Max
## -1.55954603 -0.32012837 -0.04131555  0.31523299  1.89915223
##
## Number of Observations: 45
## Number of Groups: 15

#AICC
p<- 8
print(AICC<- -2*logLik(ind.fitted.model)+2*p*n/(n-p-1))

## 'log Lik.' 533.3133 (df=8)

#Fitting random intercept-only model with spatial power covariance matrix of
error terms
summary(sppowint.fitted.model<- lme(bonus ~ totalyears + status.rel + year, r
andom = ~ 1 | id, data=Longform, correlation=corCAR1(form = ~ 1 | id)))

## Linear mixed-effects model fit by REML
##   Data: Longform
##       AIC       BIC    logLik
##  528.8405 540.8355 -257.4203
##
## Random effects:
## Formula: ~1 | id
##          (Intercept) Residual
## StdDev:  0.01464535 130.9026
##
## Correlation Structure: Continuous AR(1)
## Formula: ~1 | id
## Parameter estimate(s):
##      Phi
## 0.5467992
## Fixed effects:  bonus ~ totalyears + status.rel + year
##              Value Std.Error DF   t-value p-value
## (Intercept) -2253.4538  386.4705 29 -5.830856  0.0000
## totalyears    59.8379   8.9004 12  6.723065  0.0000
## status.relfull  48.2277  68.4510 12  0.704558  0.4945
## year        1394.6667  200.1013 29  6.969804  0.0000
## Correlation:
##              (Intr) ttlyrs stts.r
## totalyears   -0.142
## status.relfull 0.017 -0.600
## year        -0.984  0.000  0.000

```

```
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -1.74919151 -0.75519942 -0.08898529  0.79519451  1.89562705
##
## Number of Observations: 45
## Number of Groups: 15
```

```
intervals(sppowint.fitted.model)
```

```
## Approximate 95% confidence intervals
```

```
##
```

```
## Fixed effects:
```

```
##      lower      est.      upper
## (Intercept) -3043.87467 -2253.45377 -1463.03287
## totalyears   40.44562   59.83791   79.23019
## status.rell  -100.91419   48.22769   197.36956
## year         985.41361  1394.66667  1803.91972
```

```
## attr(,"label")
```

```
## [1] "Fixed effects:"
```

```
##
```

```
## Random Effects:
```

```
## Level: id
```

```
##      lower      est.      upper
## sd((Intercept)) 1.224958e-49 0.01464535 1.750969e+45
```

```
##
```

```
## Correlation structure:
```

```
##      lower      est.      upper
## Phi 0.2728708 0.5467992 0.7950434
```

```
## attr(,"label")
```

```
## [1] "Correlation structure:"
```

```
##
```

```
## Within-group standard error:
```

```
##      lower      est.      upper
## 99.34883 130.90257 172.47795
```

```
getVarCov(sppowint.fitted.model, type="conditional")
```

```
## id 1
```

```
## Conditional variance covariance matrix
```

```
##      1      2      3
```

```
## 1 17135.0 9369.7 5123.3
```

```
## 2 9369.7 17135.0 9369.7
```

```
## 3 5123.3 9369.7 17135.0
```

```
## Standard Deviations: 130.9 130.9 130.9
```

#Using the fitted model to predict the predicted bonus in 2021 for a full-time employee who has been with the company for 7 years

```
#print(predict(sppowint.fitted.model, data.frame(totalyears=7, status.rel="full", year=2.1), level=0))
```

#Random Slope and Intercept with Variance Structure for Error 8.7

```
#Uses data from 8.3
Orthopedic<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/
Exercise Files/Exercise8.3Data.csv", header = TRUE, sep =",")

#Creating Longform data
data1<- melt(Orthopedic[,c("id","gender","age","doctor", "length1",
"length2", "length3")], id.vars=c("id","gender","age","doctor"), variable.nam
e
="length.visit",value.name="length")
data2<- melt(Orthopedic[,c("score1","score2","score3")], variable.name
="score.visit", value.name="score")

## No id variables; using all as measure variables
Longform<- cbind(data1,data2)

#Creating variable for time
visit<- ifelse(Longform$score.visit=="score1",1,
ifelse(Longform$score.visit=="score2",2,3))

#Refernce Levels
gender.rel<- relevel(as.factor(Longform$gender), ref="M")
doctor.rel<- relevel(as.factor(Longform$doctor), ref="B")

#Fitting the model with untructured covariance matrix of error terms

#summary(un.fitted.model<- lme(score ~ gender.rel + age + doctor.rel + Length
+
#visit, random = ~ 1 + visit | id, control= lmeControl(opt="optim"),
#data=Longform, correlation=corSymm(), weights=varIdent(form=~id|Length)))
#DOESN'T CONVERGE

#fitting random intercept-only model with
#unstructured covariance matrix of error terms

#summary(un.fitted.model<- lme(score ~ gender.rel + age + doctor.rel + Length
+
#visit, random = ~ 1 | id, control= lmeControl(opt="optim"),
#data=Longform,correlation=corSymm(), weights=varIdent(form=~id|Length)))
#DOESN'T CONVERGE

#Fitting random slope and intercept model with Toeplitz covariance matrix of
error terms
summary(toep.fitted.model<- lme(score ~ gender.rel + age + doctor.rel + lengt
h +
visit, random = ~ 1 + visit | id, data=Longform,
correlation = corARMA(form = ~ 1 | id, p=1, q=1)))
```

```

## Linear mixed-effects model fit by REML
##   Data: Longform
##       AIC      BIC    logLik
##  274.9041 306.0456 -125.4521
##
## Random effects:
##   Formula: ~1 + visit | id
##   Structure: General positive-definite, Log-Cholesky parametrization
##              StdDev    Corr
## (Intercept) 1.5680940 (Intr)
## visit       0.4221922 -0.33
## Residual    0.1746268
##
## Correlation Structure: ARMA(1,1)
##   Formula: ~1 | id
##   Parameter estimate(s):
##       Phi1    Theta1
## -0.9999999  0.7063602
## Fixed effects: score ~ gender.rel + age + doctor.rel + length + visit
##              Value Std.Error DF   t-value p-value
## (Intercept)  5.032941 2.5636381 68   1.963203  0.0537
## gender.relF  2.065154 0.5186892 31   3.981486  0.0004
## age          0.005701 0.0384599 31   0.148241  0.8831
## doctor.relA  0.195455 0.5074278 31   0.385188  0.7027
## length       -0.010430 0.0047788 68  -2.182628  0.0325
## visit        -0.046152 0.0714296 68  -0.646117  0.5204
## Correlation:
##              (Intr) gnдр.F age    dctr.A length
## gender.relF  0.034
## age          -0.979 -0.167
## doctor.relA -0.213  0.086  0.095
## length       -0.075  0.112  0.005  0.043
## visit        -0.031 -0.005  0.000 -0.002 -0.043
##
## Standardized Within-Group Residuals:
##              Min          Q1          Med          Q3          Max
## -2.27827542 -0.43564923 -0.01232409  0.47950177  2.27827475
##
## Number of Observations: 105
## Number of Groups: 35

#AICC
n<-105
p<- 12
print(AICC<- -2*logLik(toep.fitted.model)+2*p*n/(n-p-1))

## 'log Lik.' 278.2954 (df=12)

#fitting random slope and intercept model with spatial power covariance matrix of error terms

```



```

summary(sppow.fitted.model<- lme(score ~ gender.rel + age + doctor.rel + length +
visit, random = ~ 1 + visit | id, data=Longform,
correlation=corCAR1(form = ~ 1 | id)))

## Linear mixed-effects model fit by REML
## Data: Longform
##      AIC      BIC    logLik
## 273.0585 301.6048 -125.5293
##
## Random effects:
## Formula: ~1 + visit | id
## Structure: General positive-definite, Log-Cholesky parametrization
##              StdDev   Corr
## (Intercept) 1.509031 (Intr)
## visit       0.370939 -0.245
## Residual    0.285250
##
## Correlation Structure: Continuous AR(1)
## Formula: ~1 | id
## Parameter estimate(s):
##      Phi
## 1.33668e-07
## Fixed effects: score ~ gender.rel + age + doctor.rel + length + visit
##              Value Std.Error DF   t-value p-value
## (Intercept)  5.221582 2.5707924 68   2.031118  0.0462
## gender.relF  2.084699 0.5198855 31   4.009920  0.0004
## age          0.002997 0.0385434 31   0.077754  0.9385
## doctor.relA  0.158469 0.5084558 31   0.311667  0.7574
## length      -0.010510 0.0047844 68  -2.196820  0.0314
## visit       -0.046100 0.0714364 68  -0.645336  0.5209
## Correlation:
##              (Intr) gndr.F age    dctr.A length
## gender.relF  0.033
## age         -0.979 -0.166
## doctor.relA -0.213  0.086  0.095
## length      -0.083  0.114  0.014  0.042
## visit       -0.030 -0.005 -0.001 -0.002 -0.043
##
## Standardized Within-Group Residuals:
##              Min          Q1          Med          Q3          Max
## -1.89038180 -0.46995700  0.01261281  0.40384486  1.67790864
##
## Number of Observations: 105
## Number of Groups: 35

#AICC
p<- 11
print(AICC<- -2*logLik(sppow.fitted.model)+2*p*n/(n-p-1))

```

```
## 'log Lik.' 275.8972 (df=11)

#Fitting random slope and intercept model with autoregressive covariance matrix of error terms
summary(ar.fitted.model<- lme(score ~ gender.rel + age + doctor.rel + length
+
visit, random = ~ 1 + visit| id, data=Longform,
correlation=corAR1(form = ~ 1 | id)))

## Linear mixed-effects model fit by REML
##   Data: Longform
##           AIC      BIC    logLik
##   272.9041 301.4504 -125.4521
##
## Random effects:
## Formula: ~1 + visit | id
## Structure: General positive-definite, Log-Cholesky parametrization
##           StdDev   Corr
## (Intercept) 1.5680951 (Intr)
## visit        0.4221926 -0.33
## Residual     0.1746265
##
## Correlation Structure: AR(1)
## Formula: ~1 | id
## Parameter estimate(s):
## Phi
## -1
## Fixed effects:  score ~ gender.rel + age + doctor.rel + length + visit
##               Value Std.Error DF   t-value p-value
## (Intercept)  5.032941 2.5636400 68   1.963201  0.0537
## gender.relF   2.065154 0.5186896 31   3.981483  0.0004
## age           0.005701 0.0384599 31   0.148241  0.8831
## doctor.relA   0.195455 0.5074282 31   0.385187  0.7027
## length       -0.010430 0.0047788 68  -2.182628  0.0325
## visit        -0.046152 0.0714297 68  -0.646116  0.5204
## Correlation:
##           (Intr) gndr.F age    dctr.A length
## gender.relF  0.034
## age          -0.979 -0.167
## doctor.relA -0.213  0.086  0.095
## length       -0.075  0.112  0.005  0.043
## visit        -0.031 -0.005  0.000 -0.002 -0.043
##
## Standardized Within-Group Residuals:
##           Min           Q1           Med           Q3           Max
## -2.27827948 -0.43564986 -0.01232358  0.47950317  2.27827946
##
## Number of Observations: 105
## Number of Groups: 35
```

```

#AICC
p<- 11
print(AICC<- -2*logLik(ar.fitted.model)+2*p*n/(n-p-1))

## 'log Lik.' 275.7428 (df=11)

#Fitting random intercept-only model with autoregressive covariance matrix of
error terms
summary(arint.fitted.model<- lme(score ~ gender.rel + age + doctor.rel + leng
th +
visit, random = ~ 1 | id, data=Longform, correlation=corAR1(form = ~ 1 |
id)))

## Linear mixed-effects model fit by REML
##   Data: Longform
##       AIC      BIC    logLik
##   272.612 295.9681 -127.306
##
## Random effects:
## Formula: ~1 | id
##      (Intercept) Residual
## StdDev: 0.0005791566  1.57967
##
## Correlation Structure: AR(1)
## Formula: ~1 | id
## Parameter estimate(s):
##      Phi
## 0.9396084
## Fixed effects:  score ~ gender.rel + age + doctor.rel + length + visit
##              Value Std.Error DF   t-value p-value
## (Intercept)  4.686268 2.6627508 68   1.759935  0.0829
## gender.relF  2.229715 0.5388793 31   4.137689  0.0002
## age          0.010122 0.0398445 31   0.254043  0.8011
## doctor.relA  0.167042 0.5250427 31   0.318150  0.7525
## length       -0.011298 0.0051494 68  -2.194026  0.0317
## visit        -0.045594 0.0647041 68  -0.704658  0.4834
## Correlation:
##      (Intr) gndr.F age    dctr.A length
## gender.relF  0.028
## age          -0.978 -0.163
## doctor.relA -0.211  0.083  0.095
## length       -0.101  0.138  0.030  0.016
## visit        -0.043 -0.007 -0.002 -0.001 -0.051
##
## Standardized Within-Group Residuals:
##      Min      Q1      Med      Q3      Max
## -2.23484469 -0.65331876  0.02324913  0.52261910  2.69537024
##
## Number of Observations: 105
## Number of Groups: 35

```

```

getVarCov(arint.fitted.model, type="conditional")

## id 101
## Conditional variance covariance matrix
##      1      2      3
## 1 2.4954 2.3447 2.2031
## 2 2.3447 2.4954 2.3447
## 3 2.2031 2.3447 2.4954
## Standard Deviations: 1.5797 1.5797 1.5797

#AICC
p<- 9
print(AICC<- -2*logLik(arint.fitted.model)+2*p*n/(n-p-1))

## 'log Lik.' 274.5067 (df=9)

#Fitting random slope and intercept model with compound symmetric covariance matrix of error terms
summary(cs.fitted.model<- lme(score ~ gender.rel + age + doctor.rel + length +
visit, random = ~ 1 + visit | id, data=Longform,
correlation=corCompSymm(form = ~ 1 | id)))

## Linear mixed-effects model fit by REML
## Data: Longform
##      AIC      BIC    logLik
## 273.0585 301.6048 -125.5293
##
## Random effects:
## Formula: ~1 + visit | id
## Structure: General positive-definite, Log-Cholesky parametrization
##           StdDev   Corr
## (Intercept) 1.5089867 (Intr)
## visit       0.3709369 -0.245
## Residual    0.2854340
##
## Correlation Structure: Compound symmetry
## Formula: ~1 | id
## Parameter estimate(s):
##           Rho
## 0.001279486
## Fixed effects: score ~ gender.rel + age + doctor.rel + length + visit
##           Value Std.Error DF   t-value p-value
## (Intercept)  5.221633 2.5707883 68   2.031141  0.0462
## gender.relF  2.084693 0.5198847 31   4.009914  0.0004
## age          0.002996 0.0385434 31   0.077737  0.9385
## doctor.relA  0.158465 0.5084550 31   0.311659  0.7574
## length      -0.010510 0.0047844 68  -2.196814  0.0314
## visit       -0.046100 0.0714362 68  -0.645338  0.5209
## Correlation:
##           (Intr) gndr.F age    dctr.A length

```

```

## gender.relF  0.033
## age          -0.979 -0.166
## doctor.relA -0.213  0.086  0.095
## length       -0.083  0.114  0.014  0.042
## visit        -0.030 -0.005 -0.001 -0.002 -0.043
##
## Standardized Within-Group Residuals:
##           Min           Q1           Med           Q3           Max
## -1.88928894 -0.46957901  0.01260654  0.40373339  1.67698899
##
## Number of Observations: 105
## Number of Groups: 35

#AICC
p<- 11
print(AICC<- -2*logLik(cs.fitted.model)+2*p*n/(n-p-1))

## 'log Lik.' 275.8972 (df=11)

#Fitting random slope and intercept model with independent covariance matrix
of error terms
summary(ind.fitted.model<- lme(score ~ gender.rel + age + doctor.rel + length
+
visit, random = ~ 1 + visit | id, data=Longform))

## Linear mixed-effects model fit by REML
##   Data: Longform
##       AIC       BIC    logLik
##  271.0585 297.0097 -125.5293
##
## Random effects:
##  Formula: ~1 + visit | id
##  Structure: General positive-definite, Log-Cholesky parametrization
##              StdDev    Corr
## (Intercept) 1.5090310 (Intr)
## visit        0.3709390 -0.245
## Residual     0.2852501
##
## Fixed effects:  score ~ gender.rel + age + doctor.rel + length + visit
##              Value Std.Error DF   t-value p-value
## (Intercept)  5.221580 2.5707920 68   2.031117  0.0462
## gender.relF  2.084700 0.5198854 31   4.009921  0.0004
## age          0.002997 0.0385434 31   0.077755  0.9385
## doctor.relA  0.158469 0.5084557 31   0.311668  0.7574
## length       -0.010510 0.0047844 68  -2.196820  0.0314
## visit        -0.046100 0.0714364 68  -0.645336  0.5209
## Correlation:
##              (Intr) gndr.F age    dctr.A length
## gender.relF  0.033
## age          -0.979 -0.166
## doctor.relA -0.213  0.086  0.095

```

```

## length      -0.083  0.114  0.014  0.042
## visit       -0.030 -0.005 -0.001 -0.002 -0.043
##
## Standardized Within-Group Residuals:
##           Min           Q1           Med           Q3           Max
## -1.89038162 -0.46995687  0.01261292  0.40384503  1.67790854
##
## Number of Observations: 105
## Number of Groups: 35

#AICC
p<- 10
print(AICC<- -2*logLik(ind.fitted.model)+2*p*n/(n-p-1))

## 'log Lik.' 273.3989 (df=10)

#Fitting random intercept-only model with autoregressive covariance matrix of
error terms
summary(arint.fitted.model<- lme(score ~ gender.rel + age + doctor.rel + leng
th + visit, random = ~ 1 | id,
                                data=Longform, correlation=corAR1(form = ~ 1
|id)))

## Linear mixed-effects model fit by REML
##   Data: Longform
##       AIC       BIC    logLik
##  272.612 295.9681 -127.306
##
## Random effects:
## Formula: ~1 | id
##          (Intercept) Residual
## StdDev: 0.0005791566  1.57967
##
## Correlation Structure: AR(1)
## Formula: ~1 | id
## Parameter estimate(s):
##      Phi
## 0.9396084
## Fixed effects:  score ~ gender.rel + age + doctor.rel + length + visit
##              Value Std.Error DF   t-value p-value
## (Intercept)  4.686268 2.6627508 68  1.759935  0.0829
## gender.relF  2.229715 0.5388793 31  4.137689  0.0002
## age          0.010122 0.0398445 31  0.254043  0.8011
## doctor.relA  0.167042 0.5250427 31  0.318150  0.7525
## length       -0.011298 0.0051494 68 -2.194026  0.0317
## visit        -0.045594 0.0647041 68 -0.704658  0.4834
## Correlation:
##          (Intr) gndr.F age    dctr.A length
## gender.relF  0.028
## age          -0.978 -0.163
## doctor.relA -0.211  0.083  0.095

```

```

## length      -0.101  0.138  0.030  0.016
## visit       -0.043 -0.007 -0.002 -0.001 -0.051
##
## Standardized Within-Group Residuals:
##           Min           Q1           Med           Q3           Max
## -2.23484469 -0.65331876  0.02324913  0.52261910  2.69537024
##
## Number of Observations: 105
## Number of Groups: 35

intervals(arint.fitted.model)

## Approximate 95% confidence intervals
##
## Fixed effects:
##           lower           est.           upper
## (Intercept) -0.62716841  4.68626801  9.999704438
## gender.relF  1.13066337  2.22971487  3.328766360
## age         -0.07114118  0.01012220  0.091385587
## doctor.relA -0.90378924  0.16704250  1.237874234
## length      -0.02157322 -0.01129783 -0.001022444
## visit       -0.17470925 -0.04559425  0.083520749
## attr(,"label")
## [1] "Fixed effects:"
##
## Random Effects:
## Level: id
##           lower           est.           upper
## sd((Intercept)) 7.847905e-63 0.0005791566 4.274037e+55
##
## Correlation structure:
##           lower           est.           upper
## Phi 0.8917995 0.9396084 0.9666649
## attr(,"label")
## [1] "Correlation structure:"
##
## Within-group standard error:
##           lower           est.           upper
## 1.241087 1.579670 2.010622

getVarCov(arint.fitted.model, type="conditional")

## id 101
## Conditional variance covariance matrix
##           1           2           3
## 1 2.4954 2.3447 2.2031
## 2 2.3447 2.4954 2.3447
## 3 2.2031 2.3447 2.4954
## Standard Deviations: 1.5797 1.5797 1.5797

```

#Using the fitted model to predict the quality of service score that would be given by a 55-year-old male patient on his fourth visit to Dr. A with a 30-minute appointment.

```
#print(predict(arint.fitted.model, data.frame(id=136, gender.rel="M", age=55,
doctor.rel="A", length=30, visit=4), level=0))
```

#Generalized Estimating Equations with Correlation Matrix 8.10

#Uses data from 8.2

```
DepartmentStore<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT
410/Exercise Files/Exercise8.2Data.csv", header = TRUE, sep = ",")
```

#Creating Longform dataset

```
Longform<- melt(DepartmentStore, id.vars=c("id","totalyears", "status"),
variable.name="bonus.year", value.name="bonus")
```

#Sorting data by id

```
Longform<- Longform[order(Longform$id),]
```

#Creating variable for time

```
year<- ifelse(Longform$bonus.year=="bonus18",1.8,
ifelse(Longform$bonus.year=="bonus19",1.9,2.0))
```

#Creating reference level

```
status.rel<- relevel(as.factor(Longform$status), ref="part")
```

#Fitting GEE model with unstructured working correlation matrix

```
summary(un.fitted.model<- geeglm(bonus ~ totalyears + status.rel + year,
data=Longform, id=id, family=gaussian(link="identity"),
corstr="unstructured"))
```

```
##
```

```
## Call:
```

```
## geeglm(formula = bonus ~ totalyears + status.rel + year, family = gaussian
(link = "identity"),
```

```
## data = Longform, id = id, corstr = "unstructured")
```

```
##
```

```
## Coefficients:
```

```
## Estimate Std.terr Wald Pr(>|W|)
## (Intercept) -2063.298 309.562 44.425 2.64e-11 ***
## totalyears 61.041 7.806 61.146 5.33e-15 ***
## status.rellull 41.799 51.598 0.656 0.418
## year 1289.096 169.764 57.661 3.12e-14 ***
```

```
## ---
```

```
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## Correlation structure = unstructured
```

```
## Estimated Scale Parameters:
```

```
##
```



```

##              Estimate Std.err
## (Intercept)   14719    2717
##   Link = identity
##
## Estimated Correlation Parameters:
##              Estimate Std.err
## alpha.1:2    0.13954   0.2692
## alpha.1:3   -0.08909   0.2694
## alpha.2:3    0.80758   0.1511
## Number of clusters:   15   Maximum cluster size: 3

QIC(un.fitted.model)

##           QIC          QICu  Quasi Lik          CIC      params          QICC
##  6.624e+05  6.624e+05 -3.312e+05  5.188e+00  4.000e+00  6.624e+05

#Fitting GEE model with autoregressive working correlation matrix
summary(ar.fitted.model<- geeglm(bonus ~ totalyears + status.rel + year,
data=Longform, id=id, family=gaussian(link="identity"), corstr="ar1"))

##
## Call:
## geeglm(formula = bonus ~ totalyears + status.rel + year, family = gaussian
(link = "identity"),
##   data = Longform, id = id, corstr = "ar1")
##
## Coefficients:
##              Estimate Std.err Wald Pr(>|W|)
## (Intercept)   -2251.24  436.52 26.60  2.5e-07 ***
## totalyears         59.60    8.15 53.41  2.7e-13 ***
## status.relfull    49.47   48.30  1.05    0.31
## year            1394.67  232.29 36.05  1.9e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = ar1
## Estimated Scale Parameters:
##
##              Estimate Std.err
## (Intercept)   14580    2523
##   Link = identity
##
## Estimated Correlation Parameters:
##              Estimate Std.err
## alpha    0.383    0.163
## Number of clusters:   15   Maximum cluster size: 3

QIC(ar.fitted.model)

##           QIC          QICu  Quasi Lik          CIC      params          QICC
##  6.56e+05  6.56e+05 -3.28e+05  5.89e+00  4.00e+00  6.56e+05

```

```

#Fitting GEE model with compound symmetric working correlation matrix
summary(cs.fitted.model<- geeglm(bonus ~ totalyears + status.rel + year,
data=Longform, id=id, family=gaussian(link="identity"),
corstr="exchangeable"))

##
## Call:
## geeglm(formula = bonus ~ totalyears + status.rel + year, family = gaussian
(link = "identity"),
## data = Longform, id = id, corstr = "exchangeable")
##
## Coefficients:
## Estimate Std.err Wald Pr(>|W|)
## (Intercept) -2246.99 434.47 26.75 2.3e-07 ***
## totalyears 59.14 8.27 51.14 8.6e-13 ***
## status.rell 51.87 48.01 1.17 0.28
## year 1394.67 232.29 36.05 1.9e-09 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = exchangeable
## Estimated Scale Parameters:
## Estimate Std.err
## (Intercept) 14576 2518
## Link = identity
##
## Estimated Correlation Parameters:
## Estimate Std.err
## alpha 0.287 0.159
## Number of clusters: 15 Maximum cluster size: 3

QIC(cs.fitted.model)

## QIC QICu Quasi Lik CIC params QICC
## 6.56e+05 6.56e+05 -3.28e+05 6.05e+00 4.00e+00 6.56e+05

#Fitting GEE model with independent working correlation matrix
summary(ind.fitted.model<- geeglm(bonus ~ totalyears + status.rel + year,
data=Longform, id=id, family=gaussian(link="identity"),
corstr="independence"))

##
## Call:
## geeglm(formula = bonus ~ totalyears + status.rel + year, family = gaussian
(link = "identity"),
## data = Longform, id = id, corstr = "independence")
##
## Coefficients:
## Estimate Std.err Wald Pr(>|W|)
## (Intercept) -2246.99 434.47 26.75 2.3e-07 ***

```

```

## totalyears      59.14      8.27 51.14  8.6e-13 ***
## status.rellfull 51.87     48.01  1.17    0.28
## year           1394.67    232.29 36.05   1.9e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = independence
## Estimated Scale Parameters:
##
##              Estimate Std.err
## (Intercept)   14576     2518
## Number of clusters: 15 Maximum cluster size: 3

QIC(ind.fitted.model)

##          QIC      QICu Quasi Lik      CIC      params      QICC
## 6.56e+05 6.56e+05 -3.28e+05 6.05e+00 4.00e+00 6.56e+05

summary(un.fitted.model<- geeglm(bonus ~ totalyears+status.rel + year,
data=Longform, id=id, family=gaussian(link="identity"),
corstr="unstructured"))

##
## Call:
## geeglm(formula = bonus ~ totalyears + status.rel + year, family = gaussian
(link = "identity"),
## data = Longform, id = id, corstr = "unstructured")
##
## Coefficients:
##              Estimate Std.err Wald Pr(>|W|)
## (Intercept)  -2063.30   309.56 44.43 2.6e-11 ***
## totalyears      61.04     7.81 61.15 5.3e-15 ***
## status.rellfull 41.80     51.60  0.66    0.42
## year          1289.10   169.76 57.66 3.1e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = unstructured
## Estimated Scale Parameters:
##
##              Estimate Std.err
## (Intercept)   14719     2717
## Link = identity
##
## Estimated Correlation Parameters:
##              Estimate Std.err
## alpha.1:2    0.1395    0.269
## alpha.1:3   -0.0891    0.269
## alpha.2:3    0.8076    0.151
## Number of clusters: 15 Maximum cluster size: 3

```

##Using the fitted model to predict the predicted bonus in 2021 for a full-time employee who has been with the company for 7 years

```
print(predict(un.fitted.model, data.frame(totalyears=7, status.rel="full",
year=2.1)))
```

```
##      1
```

```
## 1113
```

#Generalized Estimating Equations with Correlation Matrix 8.11

#Uses data from 8.3

```
Orthopedic<- read.csv(file="C:/Users/alex0/Desktop/School/Fall 2021/STAT 410/
Exercise Files/Exercise8.3Data.csv", header = TRUE, sep =",")
```

#creating longform dataset

```
data1<- melt(Orthopedic[,c("id","gender","age","doctor","length1",
"length2","length3")], id.vars=c("id","gender","age","doctor"),
variable.name="length.visit",value.name="length")
```

```
data2<- melt(Orthopedic[,c("score1","score2","score3")],
variable.name="score.visit", value.name="score")
```

No id variables; using all as measure variables

```
Longform<- cbind(data1,data2)
```

#sorting data by id

```
Longform<- Longform[order(Longform$id),]
```

#creating variable for time

```
visit<- ifelse(Longform$score.visit=="score1",1,
ifelse(Longform$score.visit=="score2",2,3))
```

#specifying reference levels

```
gender.rel<- relevel(as.factor(Longform$gender), ref="M")
```

```
doctor.rel<- relevel(as.factor(Longform$doctor), ref="B")
```

#fitting GEE model with unstructured working correlation matrix

```
summary(un.fitted.model<- geeglm(score ~ gender.rel + age + doctor.rel
+ length + visit, data=Longform, id=id, family=gaussian(link="identity"),
corstr="unstructured"))
```

```
##
```

```
## Call:
```

```
## geeglm(formula = score ~ gender.rel + age + doctor.rel + length +
##       visit, family = gaussian(link = "identity"), data = Longform,
##       id = id, corstr = "unstructured")
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std.err Wald Pr(>|W|)
## (Intercept)  4.68494   1.88506   6.18   0.013 *
## gender.relF   2.12954   0.52062  16.73  4.3e-05 ***
```

```

## age          0.01032  0.03105  0.11    0.740
## doctor.relA  0.21578  0.48814  0.20    0.658
## length       -0.01046  0.00558  3.52    0.061 .
## visit        -0.04797  0.07809  0.38    0.539
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = unstructured
## Estimated Scale Parameters:
##
##             Estimate Std.err
## (Intercept)    2.18    0.525
## Link = identity
##
## Estimated Correlation Parameters:
##             Estimate Std.err
## alpha.1:2      0.858  0.0522
## alpha.1:3      0.874  0.0429
## alpha.2:3      0.977  0.0476
## Number of clusters: 35 Maximum cluster size: 3

QIC(un.fitted.model)

##          QIC          QICu Quasi Lik          CIC      params          QICC
##        266.4         241.3     -114.7         18.5         6.0         273.6

#fitting GEE model with autoregressive working correlation matrix
summary(ar.fitted.model<- geeglm(score ~ gender.rel + age + doctor.rel
+ length + visit, data=Longform, id=id, family=gaussian(link="identity"),
corstr="ar1"))

##
## Call:
## geeglm(formula = score ~ gender.rel + age + doctor.rel + length +
##       visit, family = gaussian(link = "identity"), data = Longform,
##       id = id, corstr = "ar1")
##
## Coefficients:
##             Estimate Std.err Wald Pr(>|W|)
## (Intercept)  4.73590  1.90238  6.20  0.0128 *
## gender.relF  2.21330  0.54843 16.29 5.4e-05 ***
## age          0.00998  0.03141  0.10  0.7507
## doctor.relA  0.16670  0.49337  0.11  0.7355
## length       -0.01238  0.00480  6.67  0.0098 **
## visit        -0.04490  0.07126  0.40  0.5287
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = ar1
## Estimated Scale Parameters:
##

```

```

##              Estimate Std.err
## (Intercept)    2.14    0.477
##   Link = identity
##
## Estimated Correlation Parameters:
##           Estimate Std.err
## alpha      0.926    0.0201
## Number of clusters:   35   Maximum cluster size: 3

QIC(ar.fitted.model)

##           QIC      QICu Quasi Lik      CIC      params      QICC
##          262.5      236.9     -112.4      18.8         6.0      266.6

#fitting GEE model with compound symmetric working correlation matrix
summary(cs.fitted.model<- geeglm(score ~ gender.rel + age + doctor.rel
+ length + visit, data=Longform, id=id, family=gaussian(link="identity"),
corstr="exchangeable"))

##
## Call:
## geeglm(formula = score ~ gender.rel + age + doctor.rel + length +
##       visit, family = gaussian(link = "identity"), data = Longform,
##       id = id, corstr = "exchangeable")
##
## Coefficients:
##              Estimate Std.err Wald Pr(>|W|)
## (Intercept)  4.46350  1.90511  5.49   0.019 *
## gender.relF  2.15436  0.53112 16.45   5e-05 ***
## age          0.01480  0.03135  0.22   0.637
## doctor.relA  0.22814  0.48751  0.22   0.640
## length      -0.01398  0.00624  5.02   0.025 *
## visit       -0.04387  0.07187  0.37   0.542
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = exchangeable
## Estimated Scale Parameters:
##
##              Estimate Std.err
## (Intercept)    2.11    0.473
##   Link = identity
##
## Estimated Correlation Parameters:
##           Estimate Std.err
## alpha      0.898    0.0253
## Number of clusters:   35   Maximum cluster size: 3

QIC(cs.fitted.model)

```

```
##          QIC          QICu Quasi Lik          CIC          params          QICC
##          258.1          233.1          -110.6          18.5          6.0          262.2

#fitting GEE model with independent working correlation matrix
summary(ind.fitted.model<- geeglm(score ~ gender.rel + age + doctor.rel
+ length + visit, data=Longform, id=id, family=gaussian(link="identity"),
corstr="independence"))

##
## Call:
## geeglm(formula = score ~ gender.rel + age + doctor.rel + length +
##       visit, family = gaussian(link = "identity"), data = Longform,
##       id = id, corstr = "independence")
##
## Coefficients:
##              Estimate Std.err Wald Pr(>|W|)
## (Intercept)  7.92624  1.39542 32.26  1.3e-08 ***
## gender.relF  1.04380  0.44479  5.51   0.019 *
## age          0.00670  0.01984  0.11   0.736
## doctor.relA  0.07969  0.31080  0.07   0.798
## length      -0.09331  0.01397 44.62  2.4e-11 ***
## visit        0.00713  0.11386  0.00   0.950
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = independence
## Estimated Scale Parameters:
##
##              Estimate Std.err
## (Intercept)    1.28    0.189
## Number of clusters: 35 Maximum cluster size: 3

QIC(ind.fitted.model)

##          QIC          QICu Quasi Lik          CIC          params          QICC
##          153.30          146.13          -67.07          9.58          6.00          156.30

#fitting GEE model with independent working correlation matrix
summary(un.fitted.model<- geeglm(score ~ gender.rel + age + doctor.rel
+ length + visit, data=Longform, id=id, family=gaussian(link="identity"),
corstr="unstructured"))

##
## Call:
## geeglm(formula = score ~ gender.rel + age + doctor.rel + length +
##       visit, family = gaussian(link = "identity"), data = Longform,
##       id = id, corstr = "unstructured")
##
## Coefficients:
##              Estimate Std.err Wald Pr(>|W|)
## (Intercept)  4.68494  1.88506  6.18   0.013 *
```

```

## gender.relF  2.12954  0.52062 16.73  4.3e-05 ***
## age          0.01032  0.03105  0.11   0.740
## doctor.relA  0.21578  0.48814  0.20   0.658
## length       -0.01046  0.00558  3.52   0.061 .
## visit        -0.04797  0.07809  0.38   0.539
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation structure = unstructured
## Estimated Scale Parameters:
##
##             Estimate Std.err
## (Intercept)    2.18   0.525
## Link = identity
##
## Estimated Correlation Parameters:
##             Estimate Std.err
## alpha.1:2      0.858  0.0522
## alpha.1:3      0.874  0.0429
## alpha.2:3      0.977  0.0476
## Number of clusters:  35 Maximum cluster size: 3

#Using the fitted model to predict the quality of service score that would be
given by a 55-year-old male patient on his fourth visit to Dr. A with a 30-mi
nute appointment.
print(predict(un.fitted.model, data.frame(id=136, gender.rel="M", age=55,
doctor.rel="A",length=30, visit=4)))

##      1
## 4.96

```