# STAT4520 HW3

Anton Yang

2024-09-29

## Problem 1

```r
library(faraway)
library(ggplot2)
library(pscl)
```

```
## Classes and Methods for R originally developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University (2002-2015),
## by and under the direction of Simon Jackman.
## hurdle and zeroinfl functions by Achim Zeileis.
```

```r
library(nnet)
library(MASS)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

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

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

```r
set.seed(123)
lambda<-3
pi<-0.5
n<-100

zip_data<-ifelse(rbinom(n, 1, pi) == 1, 0, rpois(n, lambda))
```

```r
mean<-mean(zip_data)
variance<-var(zip_data)

cat("Mean:", mean, "\n")
```

```
## Mean: 1.67
```

```r
cat("Variance:", variance, "\n")
```

```
## Variance: 3.758687
```

```r
glm_model<-glm(zip_data ~ 1, family = poisson)
sumary(glm_model)
```

```
##             Estimate Std. Error z value  Pr(>|z|)
## (Intercept) 0.512824   0.077379  6.6274 3.417e-11
##
## n = 100 p = 1
## Deviance = 253.68376 Null Deviance = 253.68376 (Difference = 0.00000)
```

```r
pchisq(260.19280, 99, lower.tail = F )
```

```
## [1] 2.010829e-16
```

For this model, we will choose $\lambda = 3$ and $\pi = 0.5$. We'll generate 100 data and we can see that the mean is 1.67 and the variance is 3.758687. This suggests that that this is an overdispersion model with the variance being higher than the mean.

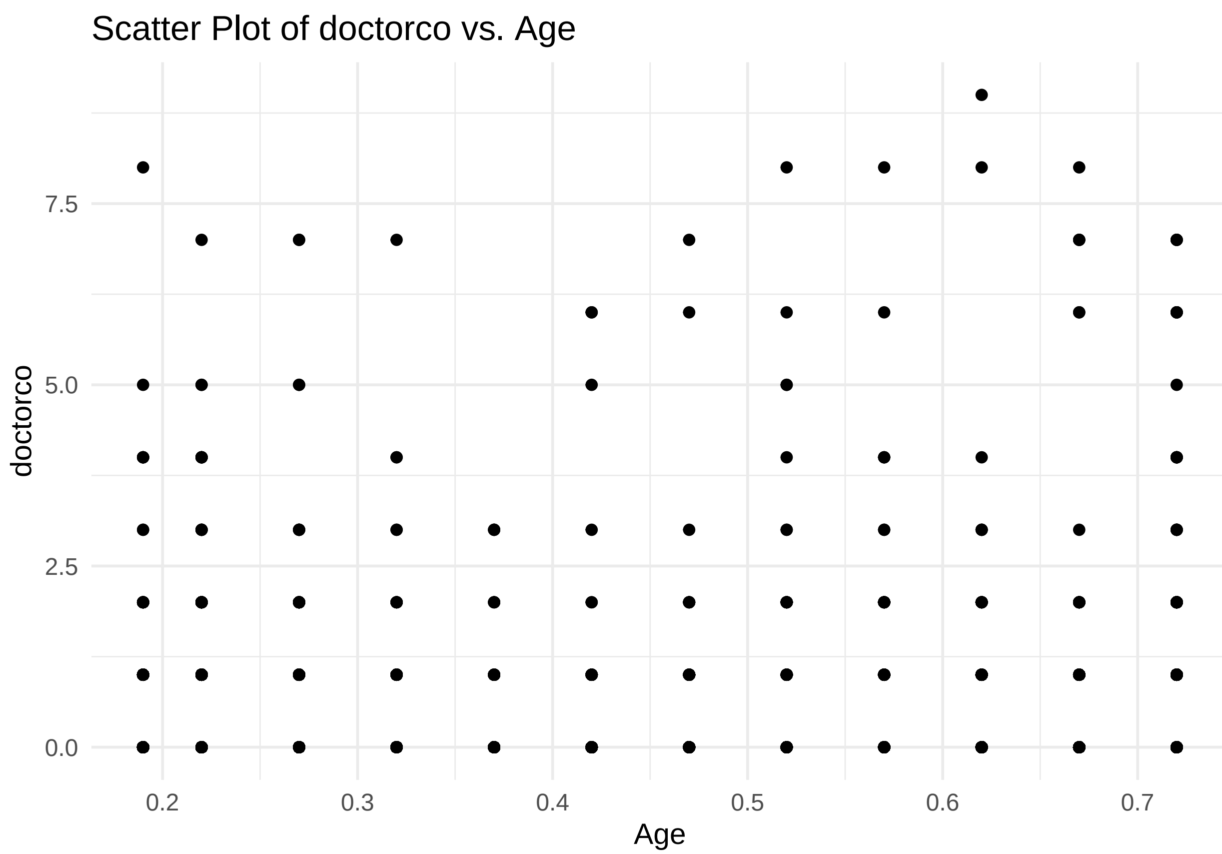
After to constructing the model, we can see that the estimated coefficient is 0.512824. We conducted a goodness of it and is produces a p-value of 2.010829e-16. This suggests that the standard Poisson model doesn't provide a good fit. This means that there's evidence that the Poisson GLM was the wrong model, and we can experiment with the Zero Inflated or Hurdle Model.

## Problem 2

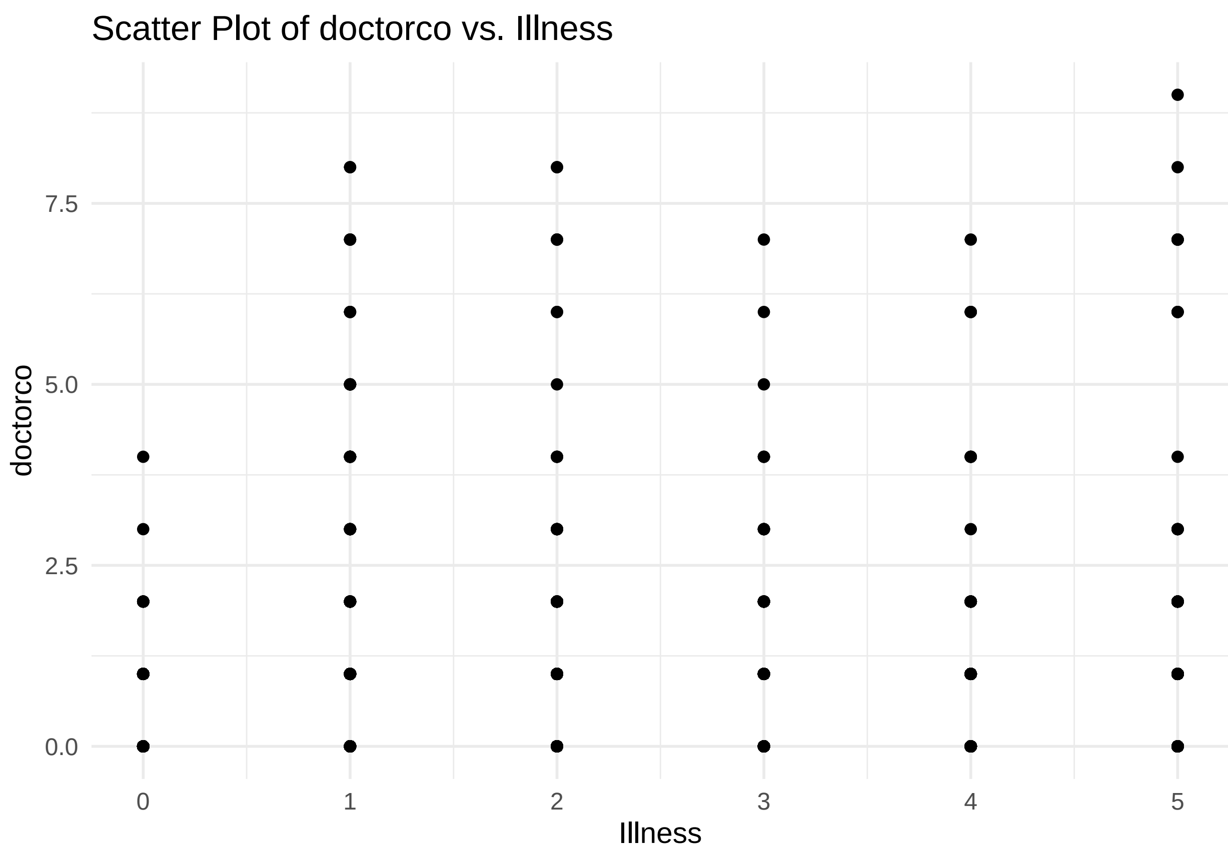
```r
data(dvisits, package = "faraway")
data<-dvisits

par(mfrow = c(1,2))

ggplot(data, aes(x = age, y = doctorco)) +
  geom_point() +
  labs(title = "Scatter Plot of doctorco vs. Age",
       x = "Age",
       y = "doctorco") +
  theme_minimal()
```



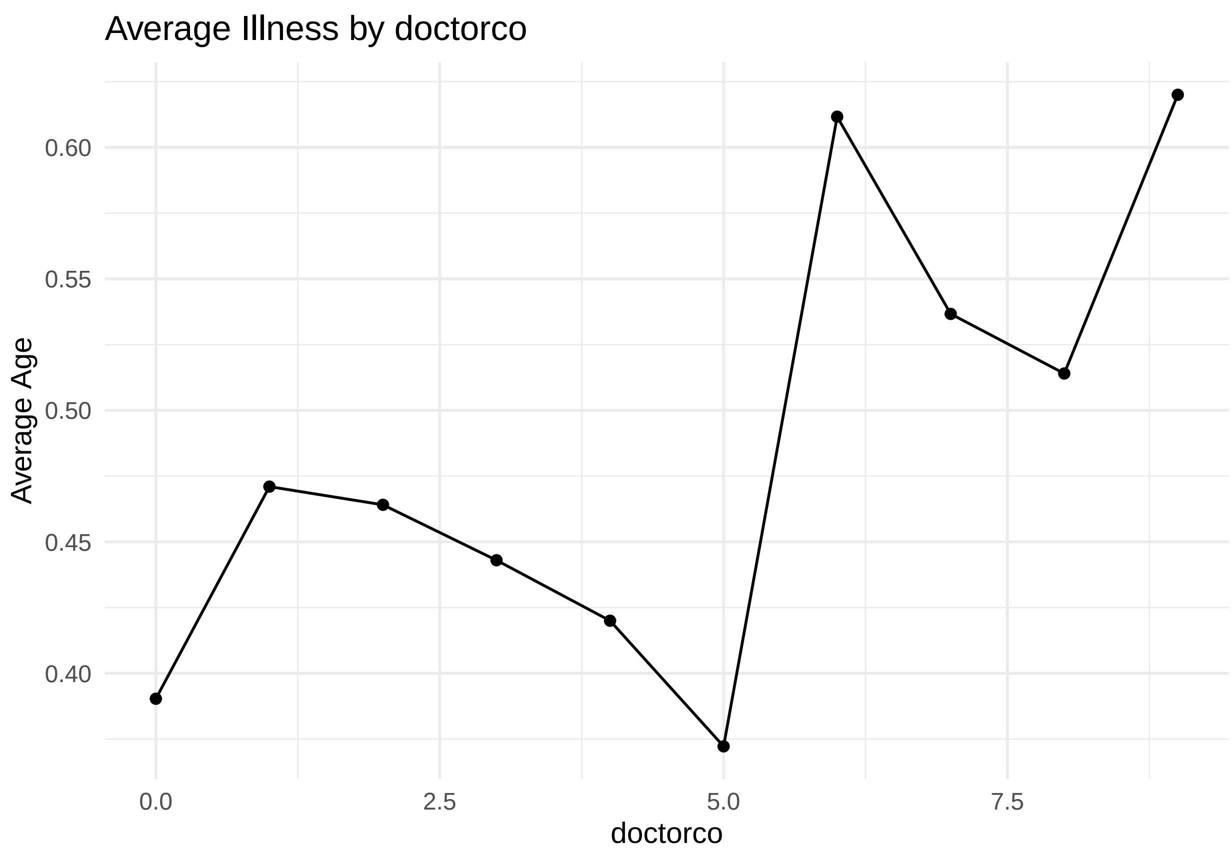

```
ggplot(data, aes(x = illness, y = doctorco)) +
  geom_point() +
  labs(title = "Scatter Plot of doctorco vs. Illness",
       x = "Illness",
       y = "doctorco") +
  theme_minimal()
```



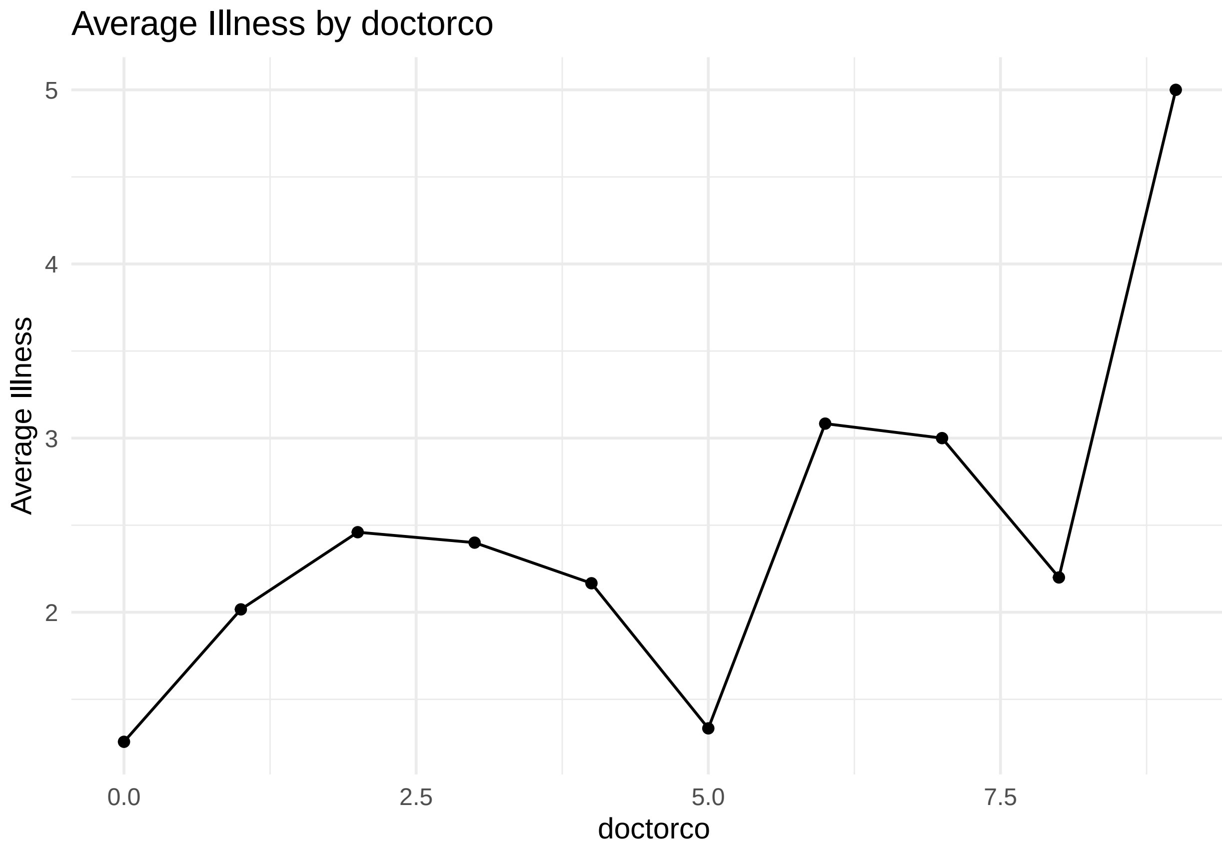

```r
avg_age <- data %>%
  group_by(doctorco) %>%
  summarize(avg_age = mean(age, na.rm = TRUE))

avg_illness <- data %>%
  group_by(doctorco) %>%
  summarize(avg_illness = mean(illness, na.rm = TRUE))

ggplot(avg_age, aes(x = doctorco, y = avg_age)) +
  geom_point() +
  geom_line() +
  labs(title = "Average Illness by doctorco",
       x = "doctorco",
       y = "Average Age") +
  theme_minimal()
```




```
ggplot(avg_illness, aes(x = doctorco, y = avg_illness)) +
  geom_point() +
  geom_line() +  # Optional: connect points with lines
  labs(title = "Average Illness by doctorco",
       x = " doctorco",
       y = "Average Illness") +
  theme_minimal()
```




We can see that there's a little correlation as the age higher there's higher number of doctorco (number of consultations with a doctor or specian the past 2 weeks). We can also see that there's a little correlation as the number of illness increases, doctorco increases. We can also see that both average age and illness increase as the doctorco increases.

```
model<-glm(doctorco ~ sex + age + agesq + income + levyplus + freepoor + freerepa + illness + actdays +
sumary(model)
```

```
##              Estimate Std. Error  z value  Pr(>|z|)
## (Intercept) -2.223848   0.189816 -11.7158 < 2.2e-16
## sex          0.156882   0.056137   2.7946  0.005196
## age          1.056299   1.000780   1.0555  0.291208
## agesq       -0.848704   1.077784  -0.7875  0.431017
## income      -0.205321   0.088379  -2.3232  0.020170
## levyplus     0.123185   0.071640   1.7195  0.085521
## freepoor    -0.440061   0.179811  -2.4473  0.014391
## freerepa     0.079798   0.092060   0.8668  0.386048
## illness      0.186948   0.018281  10.2266 < 2.2e-16
## actdays      0.126847   0.005034  25.1981 < 2.2e-16
## hscore       0.030081   0.010099   2.9785  0.002897
## chcond1      0.114085   0.066640   1.7120  0.086901
## chcond2      0.141158   0.083145   1.6977  0.089558
##
## n = 5190 p = 13
## Deviance = 4379.51510 Null Deviance = 5634.82111 (Difference = 1255.30602)
```

```r
pchisq(4379.51510, 5177, lower.tail = F)
```

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

We can see that according to the model, not many variables are significant. According to the goodness of fit, this model provides a good fit with a p-value of 1.

```r
AICModel<-step(model, trace = 0)
sumary(AICModel)
```

```
##               Estimate Std. Error  z value  Pr(>|z|)
## (Intercept) -2.0890635  0.1008113 -20.7225 < 2.2e-16
## sex          0.1620000  0.0558237   2.9020  0.003708
## age          0.3551307  0.1431956   2.4800  0.013137
## income      -0.1998064  0.0843284  -2.3694  0.017818
## levyplus     0.0836885  0.0535438   1.5630  0.118054
## freepoor    -0.4695963  0.1763601  -2.6627  0.007751
## illness      0.1861008  0.0182604  10.1915 < 2.2e-16
## actdays      0.1266107  0.0050288  25.1769 < 2.2e-16
## hscore       0.0311156  0.0100649   3.0915  0.001991
## chcond1      0.1211005  0.0663894   1.8241  0.068138
## chcond2      0.1588936  0.0817616   1.9434  0.051971
##
## n = 5190 p = 11
## Deviance = 4380.96103 Null Deviance = 5634.82111 (Difference = 1253.86009)
```

```r
pchisq(4380.96103, 5190-11, lower.tail = FALSE)
```

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

According to the AIC, we have $\log(\mu_i) = -2.089063 + 0.162000x_{sex} + 0.355131x_{age} - 0.199806x_{income} + 0.083689x_{levyplus} - 0.469596x_{freepoor} + 0.186101x_{illness} + 0.126611x_{actdays} + 0.031116x_{hscore} + 0.121100_{chcond1} + 0.158894x_{chcond2}$. We can see that the model is a very good fit with a p-value of 1. We can see that illness is really significant, with a p-value near to 0, to the model and has a coefficient of 0.1861008. This means that every increase in illnesses in past 2 weeks will increase the prediction of doctorco by a factor $e^{0.1861008}$. We can also see that actdays is also really significant. It has a coefficient of 0.1266107, which means every increase in actdays increases the prediction of doctorco by a factor $e^{0.1266107}$. Lastly, hscore is significant but not as much as illness and actdays. hscore has a coefficient of 0.0311156, which means that every increase in hscore will increase the prediction of doctorco by a factor $e^{0.0311156}$.

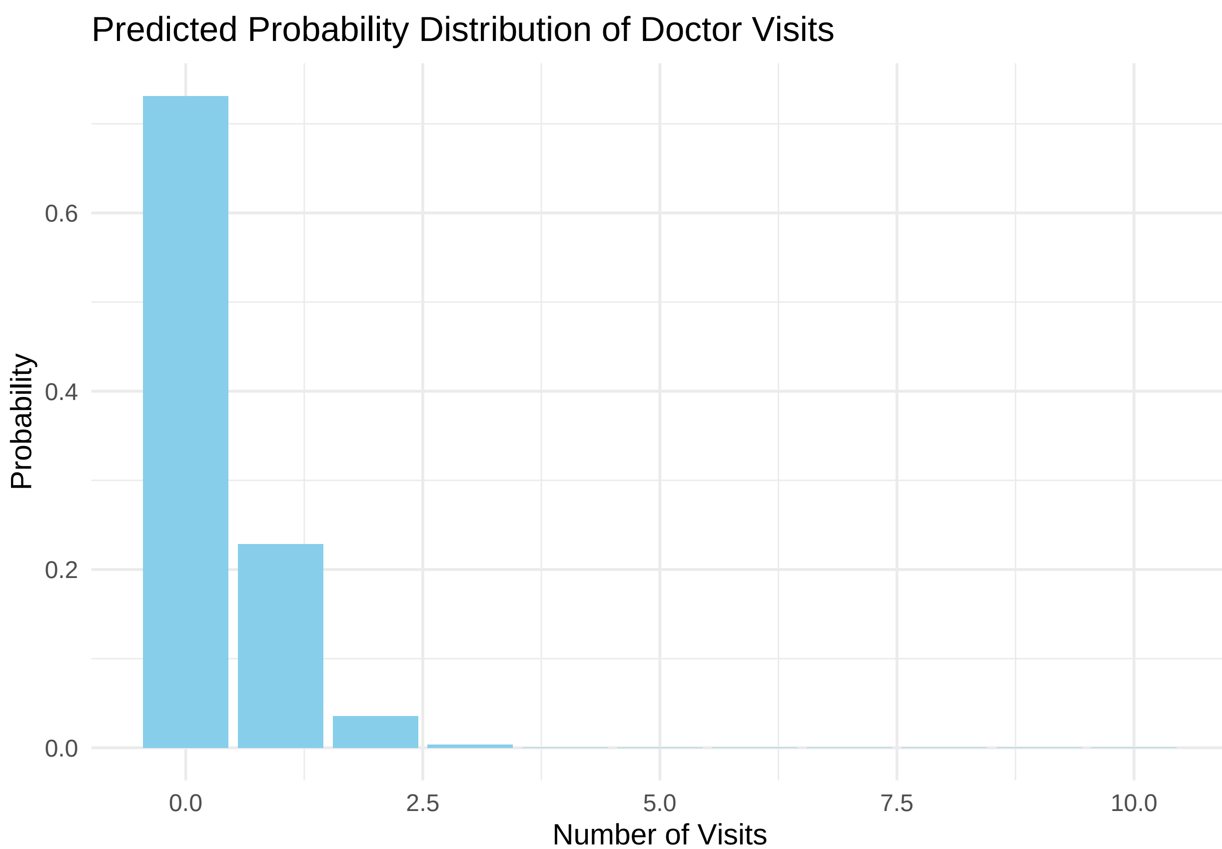
```r
first_person <- data[1,]
log_lambda<-predict(AICModel, newdata = first_person, type = "link")
lambda<-exp(log_lambda)

k<-10
probabilities <- dpois(0:k, lambda)
prob_df <- data.frame(Visits = 0:k, Probability = probabilities)

print(prob_df)
```

```
##    Visits  Probability
## 1       0 7.313832e-01
## 2       1 2.287896e-01
## 3       2 3.578472e-02
## 4       3 3.731365e-03
## 5       4 2.918092e-04
## 6       5 1.825662e-05
## 7       6 9.518321e-07
## 8       7 4.253570e-08
## 9       8 1.663240e-09
## 10      9 5.781010e-11
## 11     10 1.808402e-12
```

```
ggplot(prob_df, aes(x = Visits, y = Probability)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Predicted Probability Distribution of Doctor Visits",
       x = "Number of Visits",
       y = "Probability") +
  theme_minimal()
```



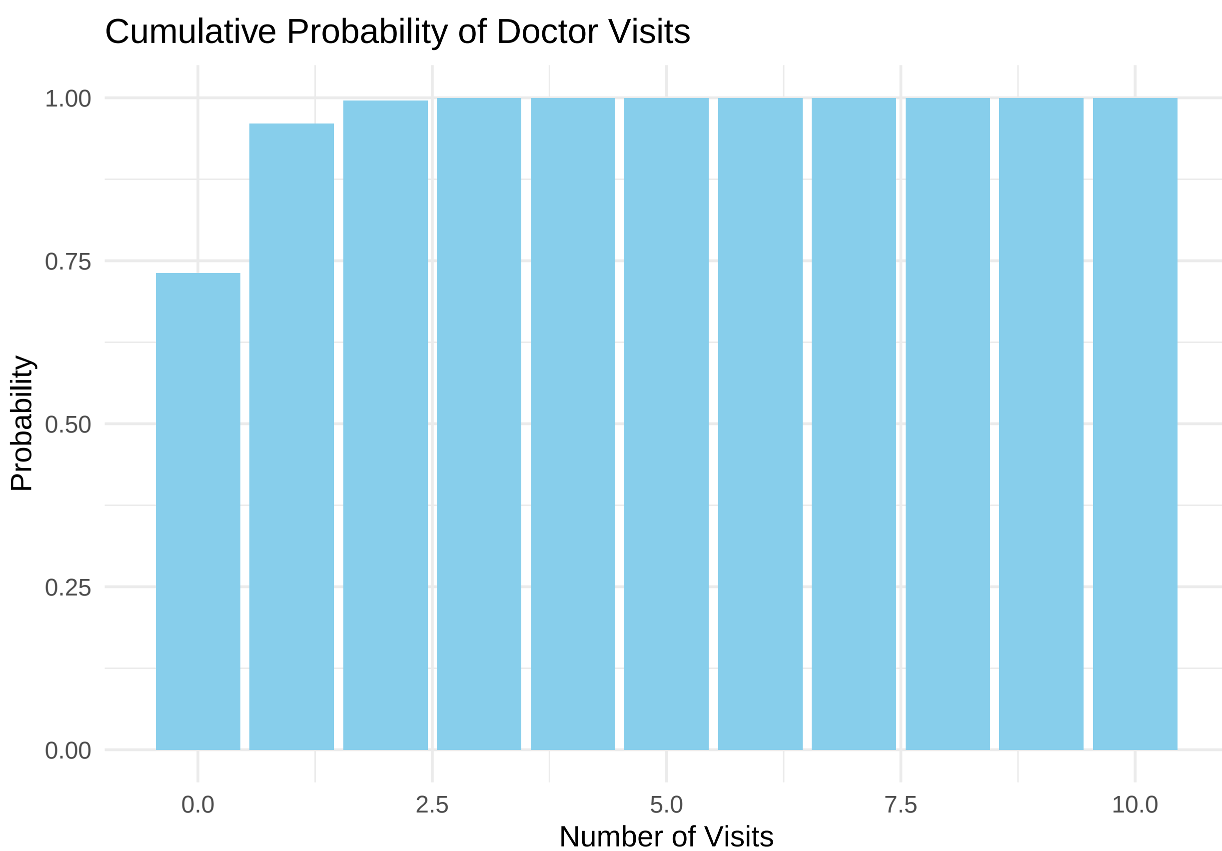

We can see that the first person visits 0 doctors has a probability of about 70%. We can see that there's a significant decrease in probability as number of doctor visits increase.

```
prob_df$Cumulative_Probability <- cumsum(prob_df$Probability)

print(prob_df)
```

```
##    Visits  Probability Cumulative_Probability
## 1       0 7.313832e-01              0.7313832
## 2       1 2.287896e-01              0.9601729
## 3       2 3.578472e-02              0.9959576
## 4       3 3.731365e-03              0.9996889
## 5       4 2.918092e-04              0.9999807
## 6       5 1.825662e-05              0.9999990
## 7       6 9.518321e-07              1.0000000
## 8       7 4.253570e-08              1.0000000
## 9       8 1.663240e-09              1.0000000
## 10      9 5.781010e-11              1.0000000
## 11     10 1.808402e-12              1.0000000
```

```
ggplot(prob_df, aes(x = Visits, y = Cumulative_Probability))+
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Cumulative Probability of Doctor Visits",
       x = "Number of Visits",
       y = "Probability") +
  theme_minimal()
```




We can see that the probability cumulates to 1 when the number of visit is 3. This means that there's a low probability that the first person visits 3 or more doctors.

```
table(data$doctorco)
```

```
##
```

```
##    0    1    2    3    4    5    6    7    8    9
## 4141  782  174   30   24    9   12   12    5    1
```

```
predicted_counts<-predict(AICModel, type = "response")
expected_freq <- table(factor(round(predicted_counts), levels = 0:max(data$doctorco)))
print(expected_freq)
```

```
##
##    0    1    2    3    4    5    6    7    8    9
## 4756  302   82   34   15    1    0    0    0    0
```

We can see from the that there are excessive number of 0's, and in fact, majority of the people has 0 doctor visits. Therefore, it is a worth fitting a Zero-Inflated Model count model in this case.

```
modz <- zeroinfl(doctorco ~ sex + age + agesq + income + levyplus + freepoor + freerepa + illness + actd
summary(modz)
```

```
##
## Call:
## zeroinfl(formula = doctorco ~ sex + age + agesq + income + levyplus +
##     freepoor + freerepa + illness + actdays + hscore + chcond1 + chcond2,
##     data = data)
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -1.6470 -0.4518 -0.2878 -0.1923 10.9964
##
## Count model coefficients (poisson with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.050368   0.255053  -4.118 3.82e-05 ***
## sex         -0.026992   0.071544  -0.377  0.70597
## age          3.128345   1.297091   2.412  0.01587 *
## agesq       -3.409196   1.373538  -2.482  0.01306 *
## income      -0.294996   0.112956  -2.612  0.00901 **
## levyplus    -0.033769   0.096469  -0.350  0.72630
## freepoor    -0.376987   0.238963  -1.578  0.11466
## freerepa    -0.215258   0.117189  -1.837  0.06623 .
## illness      0.048611   0.024571   1.978  0.04789 *
## actdays      0.082649   0.005928  13.943  < 2e-16 ***
## hscore       0.017844   0.011300   1.579  0.11430
## chcond1     -0.013380   0.092386  -0.145  0.88485
## chcond2     -0.034093   0.102785  -0.332  0.74012
##
## Zero-inflation model coefficients (binomial with logit link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.78631    0.57197   1.375 0.169216
## sex          -0.48841    0.17147  -2.848 0.004396 **
## age          10.49611    3.27099   3.209 0.001333 **
## agesq       -13.33742    3.68990  -3.615 0.000301 ***
## income       -0.43669    0.26450  -1.651 0.098735 .
## levyplus     -0.43318    0.19673  -2.202 0.027668 *
## freepoor      0.30806    0.50782   0.607 0.544092
```

```
## freerepa     -1.14905    0.30497  -3.768 0.000165 ***
## illness      -0.41581    0.08074  -5.150 2.61e-07 ***
## actdays      -1.25603    0.23809  -5.275 1.32e-07 ***
## hscore       -0.09743    0.03854  -2.528 0.011477 *
## chcond1      -0.12717    0.19908  -0.639 0.522951
## chcond2      -0.60379    0.30604  -1.973 0.048503 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 50
## Log-likelihood: -3174 on 26 Df
```

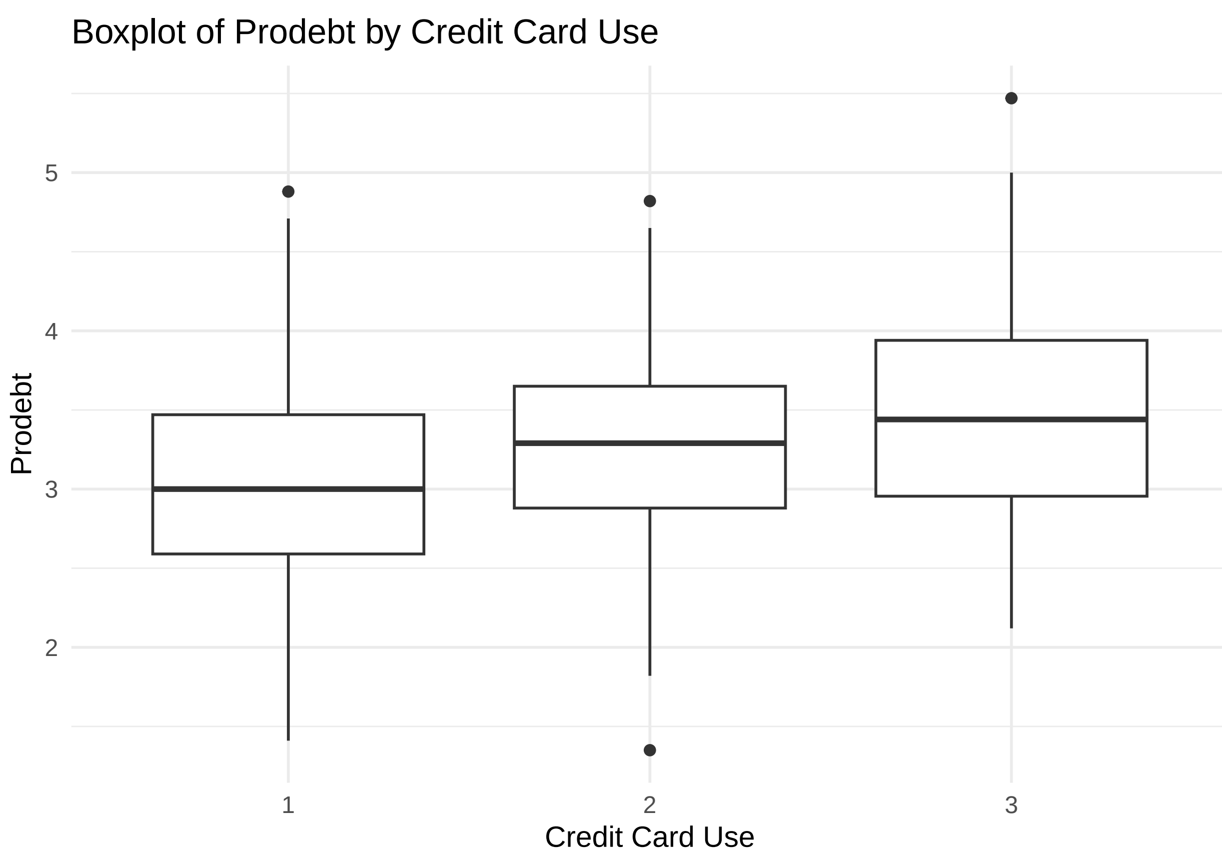
```r
AICmodz <- step(modz, trace = 0)
summary(AICmodz)
```

```
##
## Call:
## zeroinfl(formula = doctorco ~ sex + age + agesq + income + levyplus +
##     freepoor + freerepa + illness + actdays + hscore + chcond2, data = data)
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -1.6484 -0.4560 -0.2874 -0.1930 10.9208
##
## Count model coefficients (poisson with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.047727   0.254645  -4.114 3.88e-05 ***
## sex         -0.026812   0.071328  -0.376  0.70699
## age          3.099613   1.291212   2.401  0.01637 *
## agesq       -3.397356   1.372129  -2.476  0.01329 *
## income      -0.295606   0.112920  -2.618  0.00885 **
## levyplus    -0.034183   0.096067  -0.356  0.72197
## freepoor    -0.383930   0.238268  -1.611  0.10711
## freerepa    -0.216486   0.116570  -1.857  0.06329 .
## illness      0.048187   0.024373   1.977  0.04804 *
## actdays      0.082716   0.005921  13.969  < 2e-16 ***
## hscore       0.018014   0.011290   1.595  0.11060
## chcond2     -0.025039   0.077809  -0.322  0.74760
##
## Zero-inflation model coefficients (binomial with logit link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.79935    0.57164   1.398 0.162009
## sex          -0.49131    0.17145  -2.866 0.004161 **
## age          10.34931    3.26584   3.169 0.001530 **
## agesq       -13.30682    3.69548  -3.601 0.000317 ***
## income       -0.44036    0.26443  -1.665 0.095844 .
## levyplus     -0.43669    0.19638  -2.224 0.026168 *
## freepoor      0.29062    0.50704   0.573 0.566529
## freerepa     -1.15577    0.30527  -3.786 0.000153 ***
## illness      -0.42882    0.07923  -5.412 6.22e-08 ***
## actdays      -1.25501    0.23701  -5.295 1.19e-07 ***
## hscore       -0.09611    0.03839  -2.503 0.012304 *
## chcond2      -0.53330    0.28213  -1.890 0.058723 .
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 43
## Log-likelihood: -3174 on 24 Df
```

### Problem 3

```
data(debt, package = "faraway")
debt = debt[complete.cases(debt),]
ggplot(debt, aes(x = factor(ccarduse), y = prodebt)) +
  geom_boxplot() +
  labs(x = "Credit Card Use", y = "Prodebt", title = "Boxplot of Prodebt by Credit Card Use") +
  theme_minimal()
```




According to the boxplot, we can clear see that as ccarduse increases, the median prodebt increases. This means higher use of credit cards is positively correlated with the score on a scale of attitudes to debt.We can also see the lower and upper quartile increase when the ccarduse increases.

```
multi_model<-multinom(ccarduse ~ ., data = debt)
```

```
## # weights:  42 (26 variable)
## initial  value 333.978136
## iter  10 value 273.043907
## iter  20 value 253.258031
```

```
## iter  30 value 252.499137
## final  value 252.499062
## converged
```

```r
summary(multi_model)
```

```
## Call:
## multinom(formula = ccarduse ~ ., data = debt)
##
## Coefficients:
##   (Intercept)  incomegp     house   children   singpar      agegp  bankacc
## 2   -7.211297 0.3896068 0.5622512 -0.1524880 0.6410355 -0.0219860 1.604880
## 3  -11.964897 0.5923637 0.1277550 -0.1272691 1.2548392  0.3598495 2.653151
##     bsocacc    manage     cigbuy   xmasbuy   locintrn   prodebt
## 2 0.1244312 0.1157265 -0.8774290 0.9038264 0.07999505 0.4229674
## 3 0.6088511 0.2494865 -0.6747242 0.3676068 0.23939095 0.9277351
##
## Std. Errors:
##   (Intercept)  incomegp     house  children   singpar     agegp   bankacc
## 2    1.897942 0.1361372 0.3023847 0.1640768 0.7834518 0.2013498 0.6781575
## 3    2.257886 0.1469654 0.3196673 0.1747449 0.8479568 0.2191490 1.0863944
##     bsocacc    manage    cigbuy   xmasbuy  locintrn   prodebt
## 2 0.3442335 0.2004385 0.3824691 0.5384967 0.1832802 0.2387975
## 3 0.3759697 0.2187037 0.4092129 0.5353166 0.2045112 0.2598074
##
## Residual Deviance: 504.9981
## AIC: 556.9981
```

```r
AICmulti_model<-step(multi_model, trace = 0, direction = "backward")
```

```r
summary(AICmulti_model)
```

```
## Call:
## multinom(formula = ccarduse ~ incomegp + agegp + bankacc + bsocacc +
##     cigbuy + prodebt, data = debt)
##
## Coefficients:
##   (Intercept)  incomegp     agegp  bankacc   bsocacc     cigbuy   prodebt
## 2   -4.889147 0.4004706 0.1861225 1.613804 0.1642101 -0.9665574 0.3739720
## 3   -8.992223 0.5559330 0.4184312 2.660007 0.7085370 -0.7113458 0.8078491
##
## Std. Errors:
##   (Intercept)  incomegp     agegp   bankacc   bsocacc    cigbuy   prodebt
## 2    1.212734 0.1275318 0.1712133 0.6558334 0.3296689 0.3724049 0.2285155
## 3    1.609499 0.1388008 0.1879976 1.0641077 0.3631084 0.3971586 0.2453915
##
## Residual Deviance: 516.9554
## AIC: 544.9554
```

From the AIC selected model (backward), we can see that kept incomegp, agegp, bankacc, bsocacc, cigbuy, and prodebt. The final model has an AIC of 544.9554 and the original model has an AIC of 556.9981. Since the final model has a smaller AIC, this means that the final model is a better model than the original model.

```r
ordinal_model<-polr(factor(ccarduse) ~ ., data = debt)
summary(ordinal_model)
```

```
##
## Re-fitting to get Hessian

## Call:
## polr(formula = factor(ccarduse) ~ ., data = debt)
##
## Coefficients:
##              Value Std. Error t value
## incomegp  0.47131     0.1061  4.4423
## house     0.11600     0.2324  0.4992
## children -0.07872     0.1250 -0.6296
## singpar   0.88172     0.5971  1.4766
## agegp     0.20568     0.1576  1.3050
## bankacc   2.10270     0.5934  3.5435
## bsocacc   0.47322     0.2671  1.7715
## manage    0.18179     0.1653  1.0998
## cigbuy   -0.73546     0.2981 -2.4674
## xmasbuy   0.47014     0.4130  1.1385
## locintrn  0.11881     0.1424  0.8344
## prodebt   0.61046     0.1822  3.3497
##
## Intercepts:
##     Value   Std. Error t value
## 1|2  7.9694  1.4752     5.4023
## 2|3  9.3944  1.5051     6.2417
##
## Residual Deviance: 511.673
## AIC: 539.673
```

```r
AICordinal_model<-step(ordinal_model, trace = 0, direction = "backward")
summary(AICordinal_model)
```

```
##
## Re-fitting to get Hessian

## Call:
## polr(formula = factor(ccarduse) ~ incomegp + agegp + bankacc +
##     bsocacc + cigbuy + prodebt, data = debt)
##
## Coefficients:
##             Value Std. Error t value
## incomegp  0.4589     0.1007   4.555
## agegp     0.2696     0.1352   1.993
## bankacc   2.0816     0.5753   3.618
## bsocacc   0.5048     0.2591   1.949
## cigbuy   -0.7677     0.2922  -2.627
## prodebt   0.5635     0.1755   3.211
##
## Intercepts:
```

```
##     Value   Std. Error t value
## 1|2  5.9944  0.9961     6.0178
## 2|3  7.3948  1.0276     7.1961
##
## Residual Deviance: 517.5895
## AIC: 533.5895
```

We can see that the ordinal model kept same variables as the multinomial logit model. It kept incomegp, agegp, bankacc, bsocacc, cigbuy, and prodebt. The AIC for the original ordinal model is 539.673, and final ordinal model has an 533.5895, which is lower than original multinomial model, AIC selected multinomial model, and original ordinal model. This means that the best model is AIC final ordinal model.

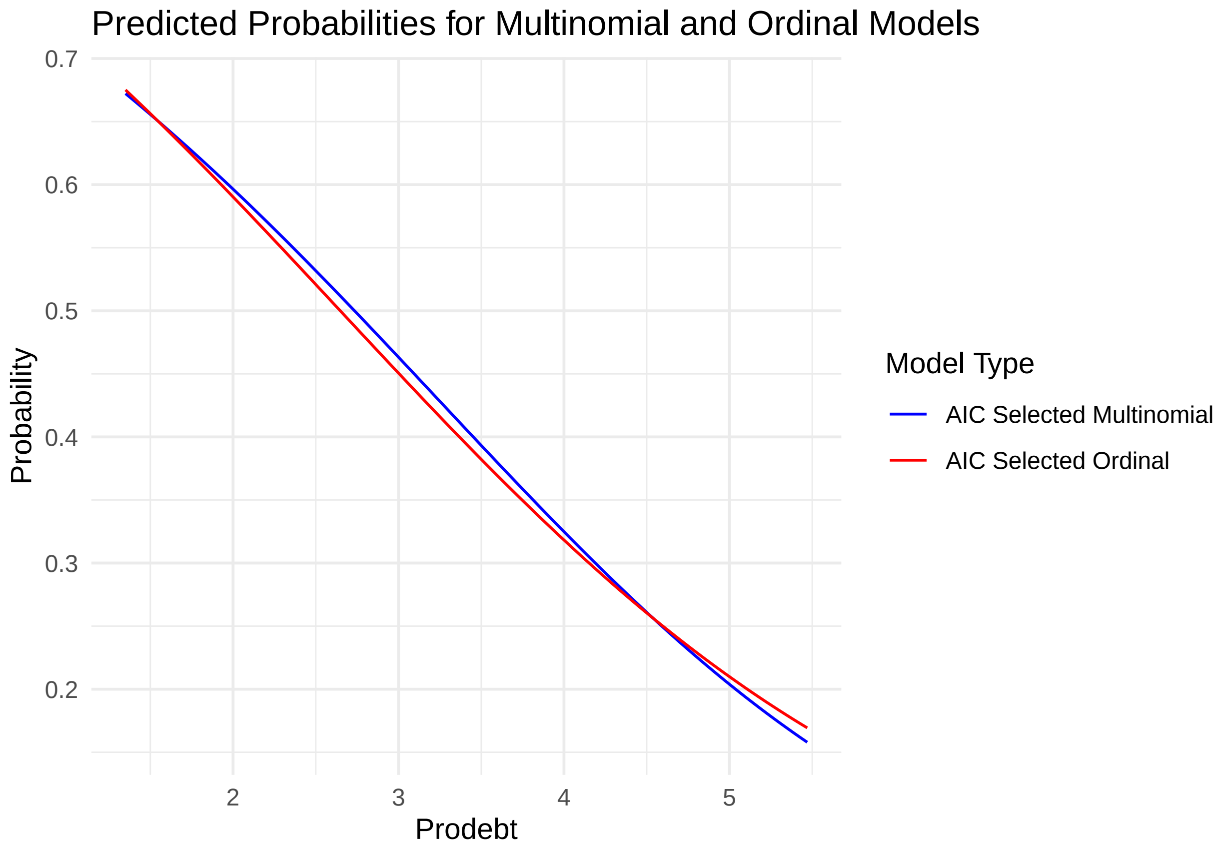
```
predictors <- expand.grid(
  incomegp = round(mean(debt$incomegp)),
  house = round(mean(debt$house)),
  children = round(mean(debt$children)),
  singpar = round(mean(debt$singpar)),
  agegp = round(mean(debt$agegp)),
  bankacc = round(mean(debt$bankacc)),
  bsocacc = round(mean(debt$bsocacc)),
  manage = round(mean(debt$manage)),
  cigbuy = round(mean(debt$cigbuy)),
  xmasbuy = round(mean(debt$xmasbuy)),
  locintrn = round(mean(debt$locintrn)),
  prodebt = seq(min(debt$prodebt), max(debt$prodebt), length.out = 100)
)

AICmulti_model_pred<-predict(AICmulti_model, newdata = predictors, type = "probs")

AICordinal_model_pred<-predict(AICordinal_model, newdata = predictors, type = "probs")

prediction_data <- data.frame(
  prodebt = predictors$prodebt,
  AICmulti_model_pred = AICmulti_model_pred[, 1],
  AICordinal_model_pred = AICordinal_model_pred[, 1]
)

ggplot(prediction_data) +
  geom_line(aes(x = prodebt, y = AICmulti_model_pred, color = "AIC Selected Multinomial")) +
  geom_line(aes(x = prodebt, y = AICordinal_model_pred, color = "AIC Selected Ordinal")) +
  labs(title = "Predicted Probabilities for Multinomial and Ordinal Models",
       x = "Prodebt",
       y = "Probability") +
  theme_minimal() +
  scale_color_manual(values = c("blue", "red", "green", "purple")) +
  guides(color = guide_legend(title = "Model Type"))
```




I set all the predictors to their rounded averages. The plot shows that the multinomial logit model predicts a higher probability than the ordinal model for prodebt values ranging from about 1 to 4. Conversely, when prodebt exceeds 5, the ordinal model predicts a higher probability than the multinomial logit model. This shift shows the different predictive behaviors of the two models based on the levels of prodebt.