HW1

Austin Brewer

9/9/2023

```
#1)
set.seed(1)
y = sample(1:1000, 200, replace=TRUE)
x1 = sample(1:2, 200, replace=TRUE)
x2 = sample(1:1000, 200, replace=TRUE)
df1 = data.frame(y,x1,x2)

mod1 = lm(y~x1+x2,data=df1)
#True values of B0,B1, and B2
coef(mod1)
```

```
##   (Intercept)            x1            x2
## 464.473897221  30.587157308  -0.003059261
```

```
#Variance
sqrt(deviance(mod1)/df.residual(mod1))
```

```
## [1] 286.045
```

```
x3 = sample(1:2, 200, replace=TRUE)
x4 = sample(1:1000, 200, replace=TRUE)
x5 = sample(1:1000, 200, replace=TRUE)

mod2 = lm(y~x1+x2+x3+x4+x5,data=df1)
#True values of B0,B1, and B2
coef(mod2)
```

```
##   (Intercept)            x1            x2            x3            x4
## 475.458848741  35.232155387   0.002957104 -36.473963063   0.140611159
##            x5
##  -0.069312094
```

```
#Variance
sqrt(deviance(mod2)/df.residual(mod2))
```

```
## [1] 284.6873
```

```
#Testing to see which predictors are significant and building a model
accordingly.
summary(mod2)
```

```
##
## Call:
```

```
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5, data = df1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -561.89 -221.27  -23.07  219.88  558.67
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 475.458849 106.148603   4.479 1.28e-05 ***
## x1           35.232155  40.772344   0.864    0.389
## x2            0.002957   0.072192   0.041    0.967
## x3          -36.473963  40.520483  -0.900    0.369
## x4            0.140611   0.073423   1.915    0.057 .
## x5           -0.069312   0.071875  -0.964    0.336
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 284.7 on 194 degrees of freedom
## Multiple R-squared:  0.02736,    Adjusted R-squared:  0.002295
## F-statistic: 1.092 on 5 and 194 DF,  p-value: 0.3664
```

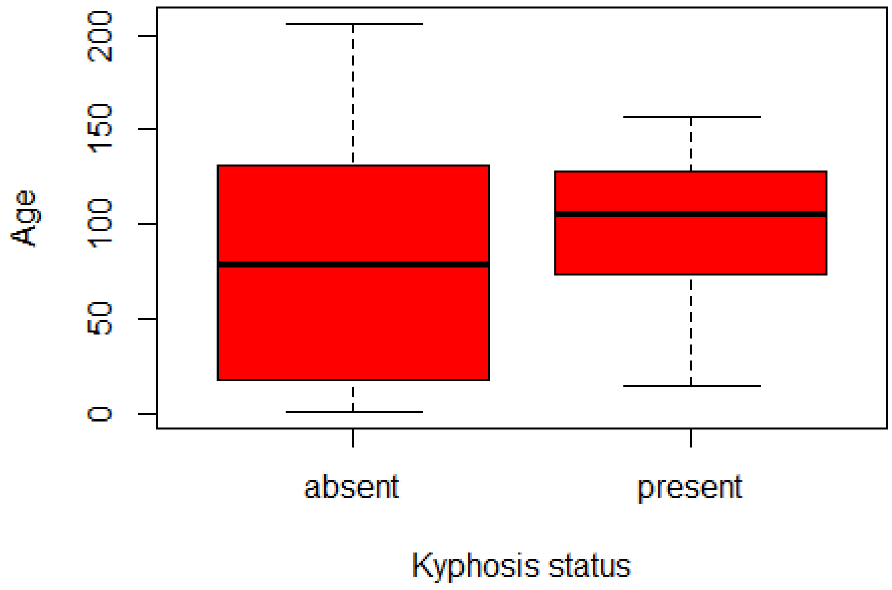
```
#According to the summary of the 2nd model, the only statistically
significant predictor is x4.

mod3 = lm(y~x4, data=df1)


#Testing the model with the only predictor that was significantly
significant.
anova(mod3,mod2)
```

```
## Analysis of Variance Table
##
## Model 1: y ~ x4
## Model 2: y ~ x1 + x2 + x3 + x4 + x5
##   Res.Df      RSS Df Sum of Sq      F Pr(>F)
## 1    198 15933511
## 2    194 15723094  4    210416 0.6491 0.6282
```

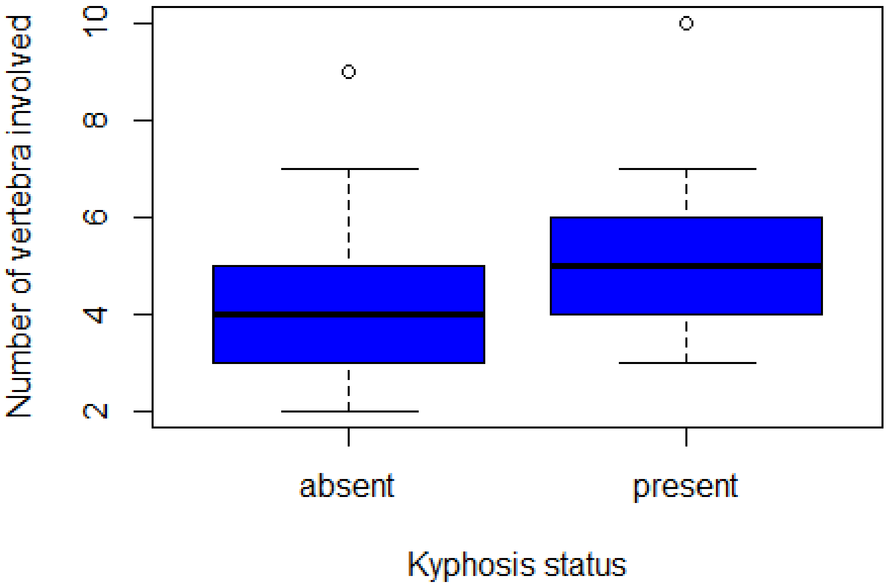
```
#According to the anova tests, the original model is superior in both cases
as we fail to reject the null hypotheses (which is the original model is
superior).


AICmod = step(mod2, trace=FALSE)
AICmod
```

```
##
## Call:
## lm(formula = y ~ x4, data = df1)
##
```

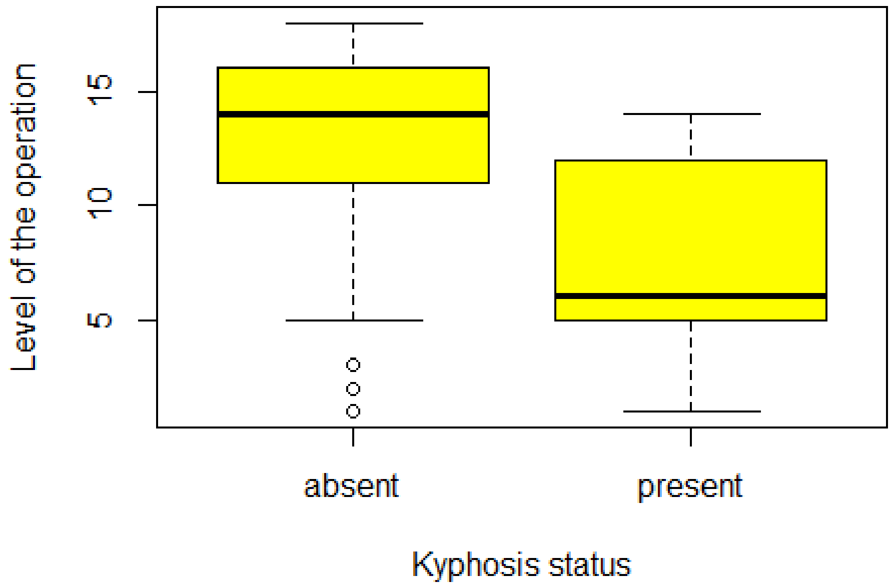
```
## Coefficients:
## (Intercept)           x4
##    450.8393       0.1226

confint(mod3)

##                    2.5 %      97.5 %
## (Intercept) 372.88658711 528.7920592
## x4           -0.01980993   0.2649716

#I did arrive at the same model as before. X4 is the only significant
predictor, but it is unclear whether the impact on the response is positive
or negative. Although the majority of the confidence interval is positive, a
small portion of it is negative which indicates a small change of a negative
affect on the response.

#2)
data(kyphosis, package = "rpart")

a)
plot(kyphosis$Kyphosis, kyphosis$Age, xlab="Kyphosis status", ylab="Age",
col="red")
```



```
#The median age for the group that has kyphosis appears to be higher than
that of the group without. The kyphosis group has a much smaller variance as
well.
plot(kyphosis$Kyphosis, kyphosis$Number, xlab="Kyphosis status", ylab="Number
of vertebra involved", col="blue")
```



```
#The number of vertebra is higher on average for those with kyphosis than without. The variance for both groups this time is very similar.
plot(kyphosis$Kyphosis, kyphosis$Start, xlab="Kyphosis status", ylab="Level of the operation", col="yellow")
```



```
#The level of operation is lower for those with kyphosis than without. The
variance for the kyphosis group is greater, and the non-kyphosis group
appears to have 3 outliers (that would only make the mean greater if they
were removed).

b)
kyphosis$Kyphosis = as.numeric(kyphosis$Kyphosis)-1
gmod = glm(kyphosis$Kyphosis~kyphosis$Age+kyphosis$Number+kyphosis$Start,
data=kyphosis)
summary(gmod)

##
## Call:
## glm(formula = kyphosis$Kyphosis ~ kyphosis$Age + kyphosis$Number +
##     kyphosis$Start, data = kyphosis)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.79440  -0.22356  -0.08478   0.10205   0.84768
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.2612198  0.1934124   1.351  0.18078
## kyphosis$Age     0.0010657  0.0006937   1.536  0.12858
## kyphosis$Number  0.0525555  0.0274522   1.914  0.05928 .
## kyphosis$Start  -0.0307392  0.0091166  -3.372  0.00117 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for gaussian family taken to be 0.1295296)
## 
##     Null deviance: 13.4321  on 80  degrees of freedom
## Residual deviance:  9.9738  on 77  degrees of freedom
## AIC: 70.214
## 
## Number of Fisher Scoring iterations: 2
```

```
#The significant variables are number and start.
gmod2 = glm(kyphosis$Kyphosis~kyphosis$Number+kyphosis$Start, data=kyphosis)
summary(gmod2)
```

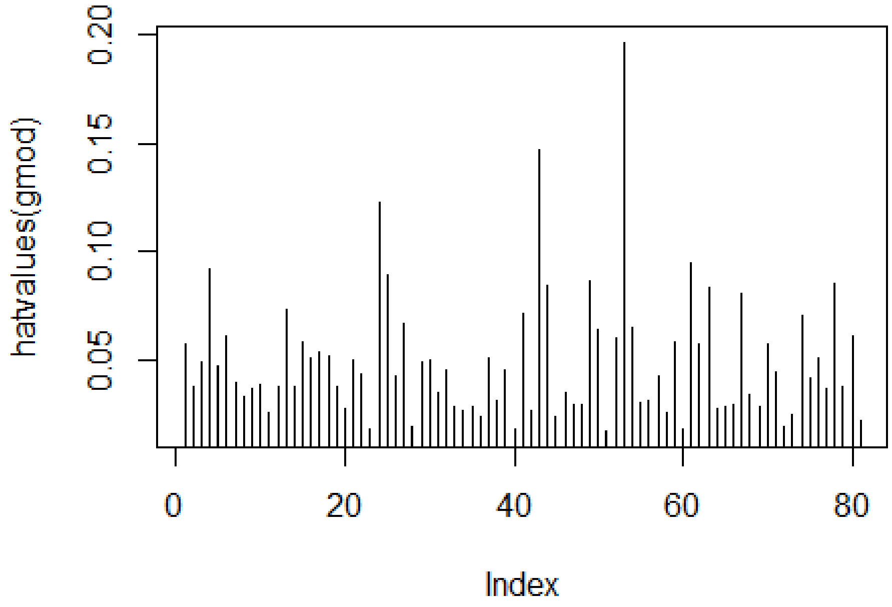
```
## 
## Call:
## glm(formula = kyphosis$Kyphosis ~ kyphosis$Number + kyphosis$Start, 
##     data = kyphosis)
## 
## Deviance Residuals: 
##      Min        1Q    Median        3Q       Max  
## -0.72631  -0.21507  -0.07927   0.06352   0.89078  
## 
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)   
## (Intercept)      0.339854   0.188135   1.806  0.07471 . 
## kyphosis$Number  0.052924   0.027689   1.911  0.05963 . 
## kyphosis$Start  -0.029954   0.009181  -3.263  0.00164 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for gaussian family taken to be 0.1317881)
## 
##     Null deviance: 13.432  on 80  degrees of freedom
## Residual deviance: 10.279  on 78  degrees of freedom
## AIC: 70.66
## 
## Number of Fisher Scoring iterations: 2
```

```
#c)
exp(coef(gmod2))
```

```
##     (Intercept) kyphosis$Number  kyphosis$Start 
##       1.4047430       1.0543495       0.9704901 
```

```
#As the number of vertebra involved increases by 1 unit, the probability of
kyphosis being present increases by 5.4%.
#AS the level of the operation increases by 1 unit, the probability of
kyphosis being present decreases by 2.96%.
```

```
d)
confint(gmod2)

## Waiting for profiling to be done...

##                        2.5 %      97.5 %
## (Intercept)     -0.028884183  0.70859292
## kyphosis$Number -0.001346278  0.10719423
## kyphosis$Start  -0.047949018 -0.01195918

#The confidence intervals for the parameters back up the interpretations for
2c. While I can't confirm that the number of vertebra will increase the
probability of kyphosis (since the lower bound is negative) the vast majority
of the CI is positive. I can however confirm that the level of operation will
decrease the probability of kyphosis being present with both bounds being
negative.

e)
AICgmod = step(gmod, trace=FALSE, direction="backward")
AICgmod

##
## Call:  glm(formula = kyphosis$Kyphosis ~ kyphosis$Age + kyphosis$Number +
##     kyphosis$Start, data = kyphosis)
##
## Coefficients:
##     (Intercept)     kyphosis$Age  kyphosis$Number   kyphosis$Start
##        0.261220         0.001066         0.052556        -0.030739
##
## Degrees of Freedom: 80 Total (i.e. Null);  77 Residual
## Null Deviance:       13.43
## Residual Deviance: 9.974     AIC: 70.21

#The model is different from the model in part b as it now includes the age
variable with the other two.
#The model is y = 0.261 + 0.001066x1 + 0.052556x2 - 0.030739x3

f)
#I was having trouble with the faraway package so I did this instead.
plot(hatvalues(gmod), type="h")
```



```
#After looking at the plot, it seems like 0.15 is a reasonable cutoff for
leverage points.
lev = as.data.frame(hatvalues(gmod))
points = c()
rowNum = c()
#This loop will go through every hat value in the data, and then it will pull
out the hat value itself and the row index
for(x in 1:nrow(lev)){
  if(lev[x,1]>0.15){
    points = c(points, lev[x,1])
    rowNum = c(rowNum, x)
  }
}


print(points)

## [1] 0.1962518

print(rowNum)

## [1] 53

#Since the row is 53, I'll pull out row 53 from the original dataset.
print(kyphosis[53,])
```

```
##    Kyphosis Age Number Start
## 53        1 139     10     6
```

```
#It seems that the leverage point is driven by the number of vertebrae
involved in the operation and is also elevated by the level of the operation.
Only one other operation had more than 7 vertebrae involved, while that other
operation, (which had 9 involved), only had a level of operation of 3 while
this one had a level of 6.

g1)
ky = kyphosis
ky = mutate(ky, probs=predict(gmod, type="response"))
gky = ky %>% group_by(Start)
gky = gky %>% summarise(mProbs=mean(probs), count=n(), yes=sum(Kyphosis==1))
gky = mutate(gky, se=sqrt(mProbs*(1-mProbs)/count))
```

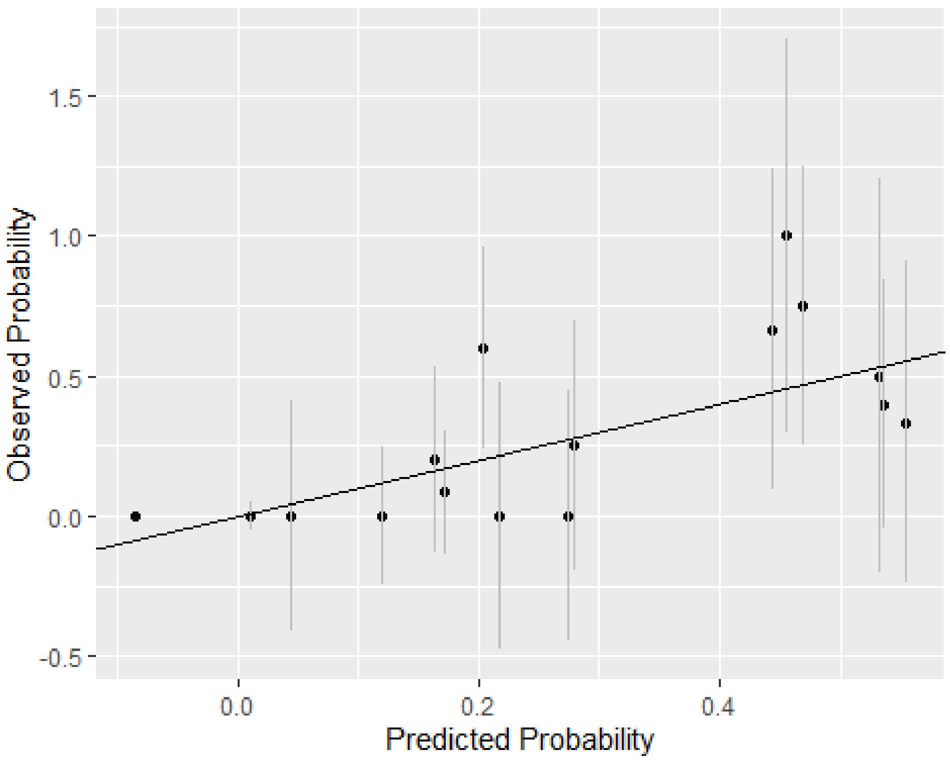
```
## Warning in sqrt(mProbs * (1 - mProbs)/count): NaNs produced
```

```
gky
```

```
## # A tibble: 16 x 5
##    Start  mProbs count   yes      se
##    <int>   <dbl> <int> <int>   <dbl>
##  1     1  0.536      5     2   0.223
##  2     2  0.533      2     1   0.353
##  3     3  0.555      3     1   0.287
##  4     5  0.443      3     2   0.287
##  5     6  0.469      4     3   0.250
##  6     8  0.455      2     2   0.352
##  7     9  0.274      4     0   0.223
##  8    10  0.279      4     1   0.224
##  9    11  0.216      3     0   0.238
## 10    12  0.203      5     3   0.180
## 11    13  0.172     12     1   0.109
## 12    14  0.164      5     1   0.166
## 13    15  0.119      7     0   0.122
## 14    16  0.0102    17     0   0.0244
## 15    17 -0.0851     4     0 NaN
## 16    18  0.0446     1     0   0.206
```

```
ggplot(gky, aes(x=mProbs, y=yes/count, ymin=yes/count-2*se,
ymax=yes/count+2*se)) +
  geom_point() + geom_linerange(color=grey(0.75)) +
geom_abline(intercept=0,slope=1) +
  xlab("Predicted Probability") + ylab("Observed Probability")
```

```
## Warning: Removed 1 rows containing missing values (geom_segment).
```



```
#It seems that the predicted probabilities generally follow a 1 to 1 linear
correlation with the observed probabilities. There is one point with a
negative predicted probabilility that also lacks a standard error. I don't
know what is causing this as there are other groups with similar data to it.

g2)
hl = with(gky, sum((yes-count*mProbs)^2/(count*mProbs*(1-mProbs))))
c(hl, nrow(gky))

## [1] 14.01485 16.00000

1-pchisq(14.01485, 16-1)

## [1] 0.5244032

#According to the test, there is no lack of fit.

g3)
ky = mutate(ky, predout=ifelse(probs < 0.5, "no", "yes"))
xtabs( ~ Kyphosis + predout, ky)

##         predout
## Kyphosis no yes
##        0 61   3
##        1 11   6

#Specificity: 61/64=(95.3%) - This means that 95.3% of patients predicted to
not have kyphosis will not get it.
```

```
#Sensitivity: 11/17=(64.7%) - This means that 64.7% of patients predicted to have kyphosis will get it.
```