STAT4520 HW2

Anton Yang

2024-09-14

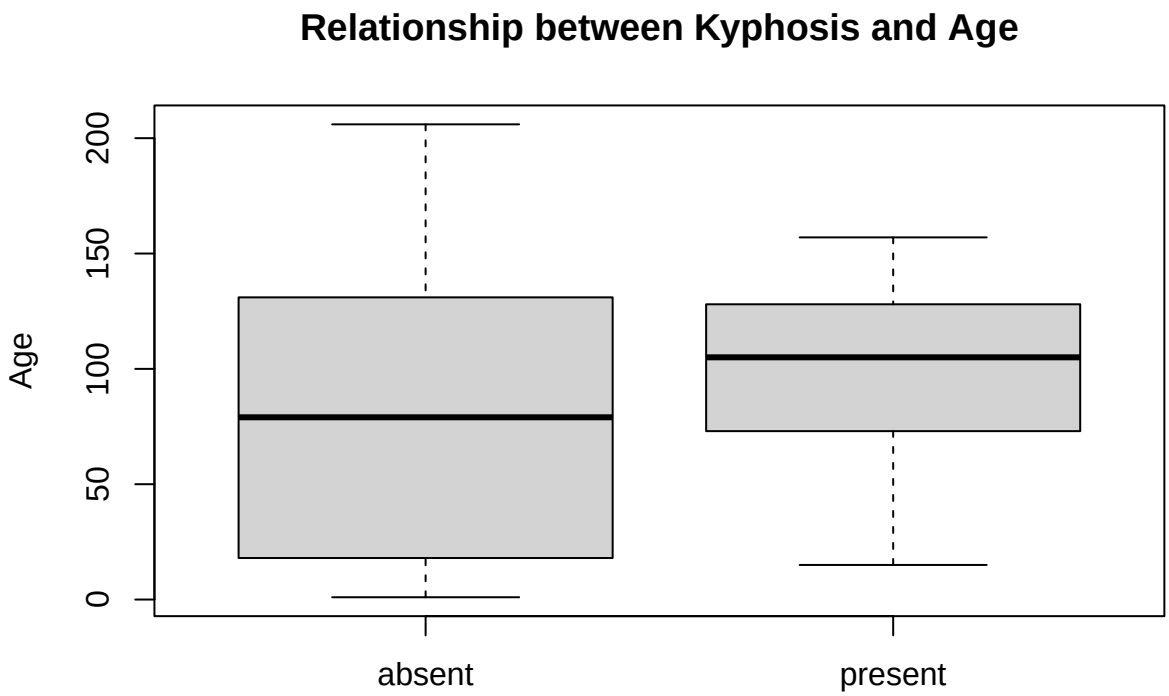
Problem 1 - Binary Response

```
set.seed(123)
library(MASS)
library(faraway)
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
```

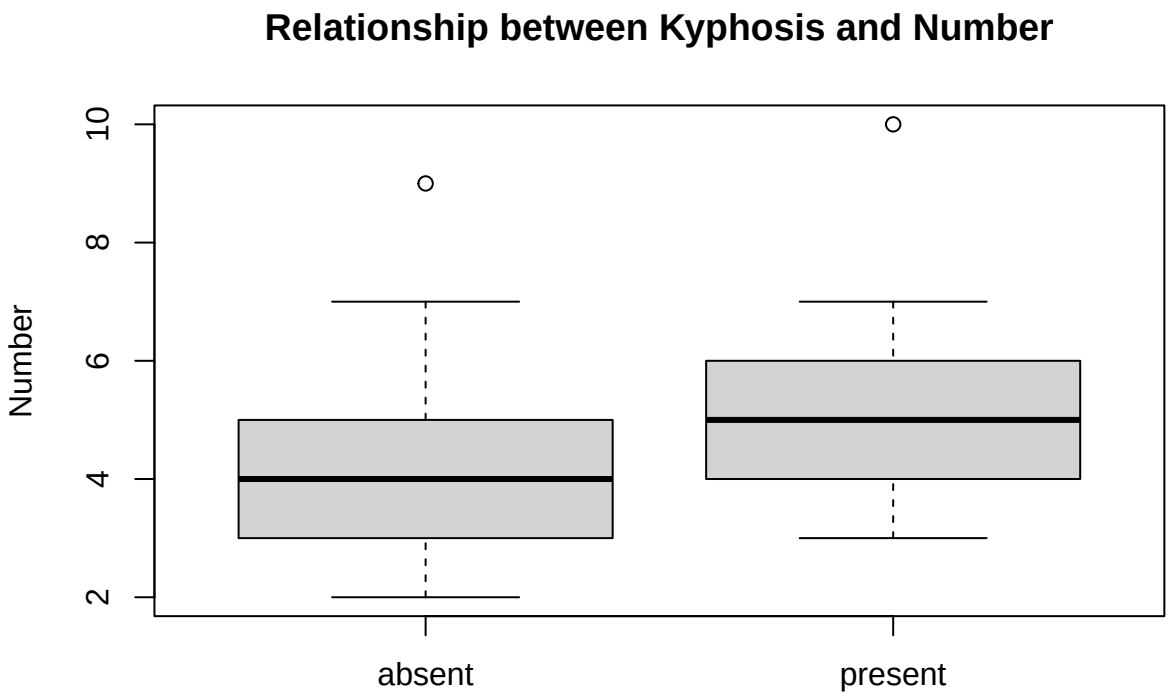
```
library(ggplot2)
library(tidyr)
data(kyphosis, package = "rpart")
```

```
plot(kyphosis$Kyphosis,kyphosis$Age, main = "Relationship between Kyphosis and Age",
xlab = "", ylab = "Age")
```



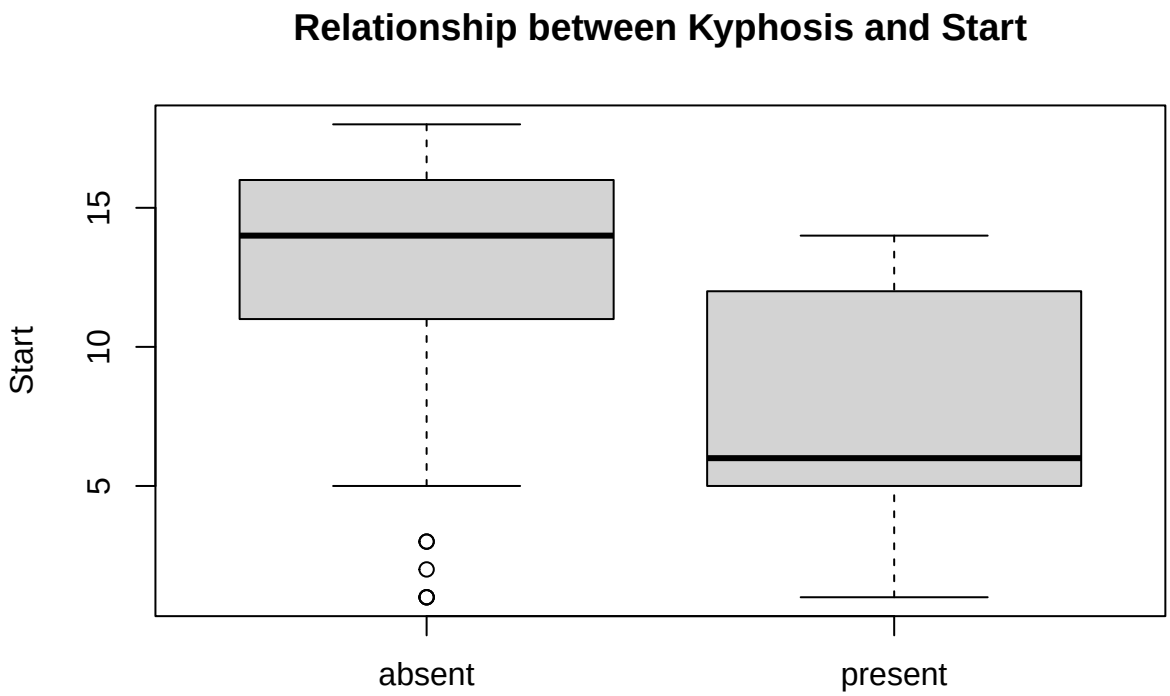
We can see that there is a clear difference in median age between the state of Kyphosis. We can see that Kyphosis typically is present with older children. We can also see that that the quartile is wider for absent Kyphosis and 75% of the children has Kyphosis is around the age of 70 - 80 months.

```
plot(kyphosis$Kyphosis,kyphosis$Number, main = "Relationship between Kyphosis and Number",
xlab = "", ylab = "Number")
```



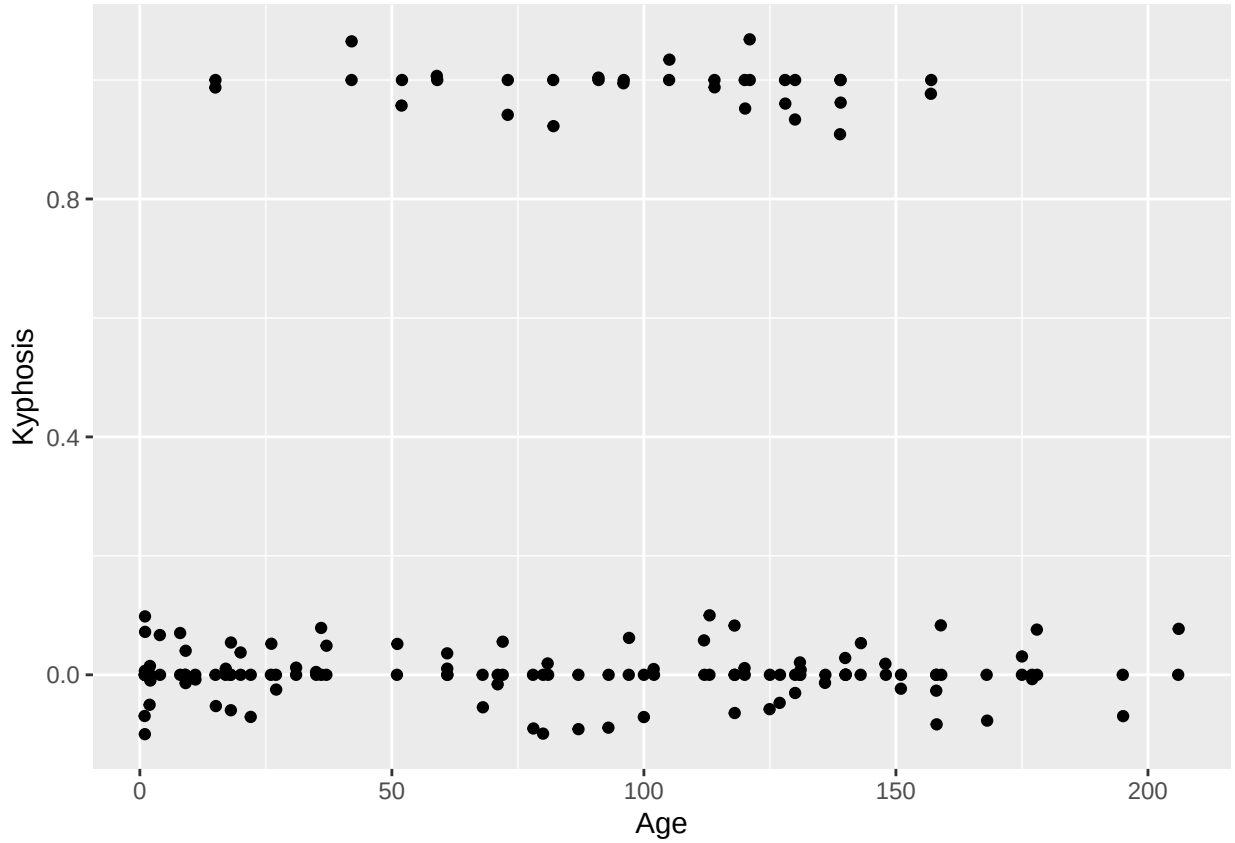
We can see that that there's a clear difference between the median of the number of vertebrae with absent and present Kyphosis. We can see that the width of quartile is around the same and there's a clear difference between the lower quartile and the upper quartile with absent and present Kyphosis.

```
plot(kyphosis$Kyphosis,kyphosis$Start, main = "Relationship between Kyphosis and Start",
xlab = "", ylab = "Start")
```

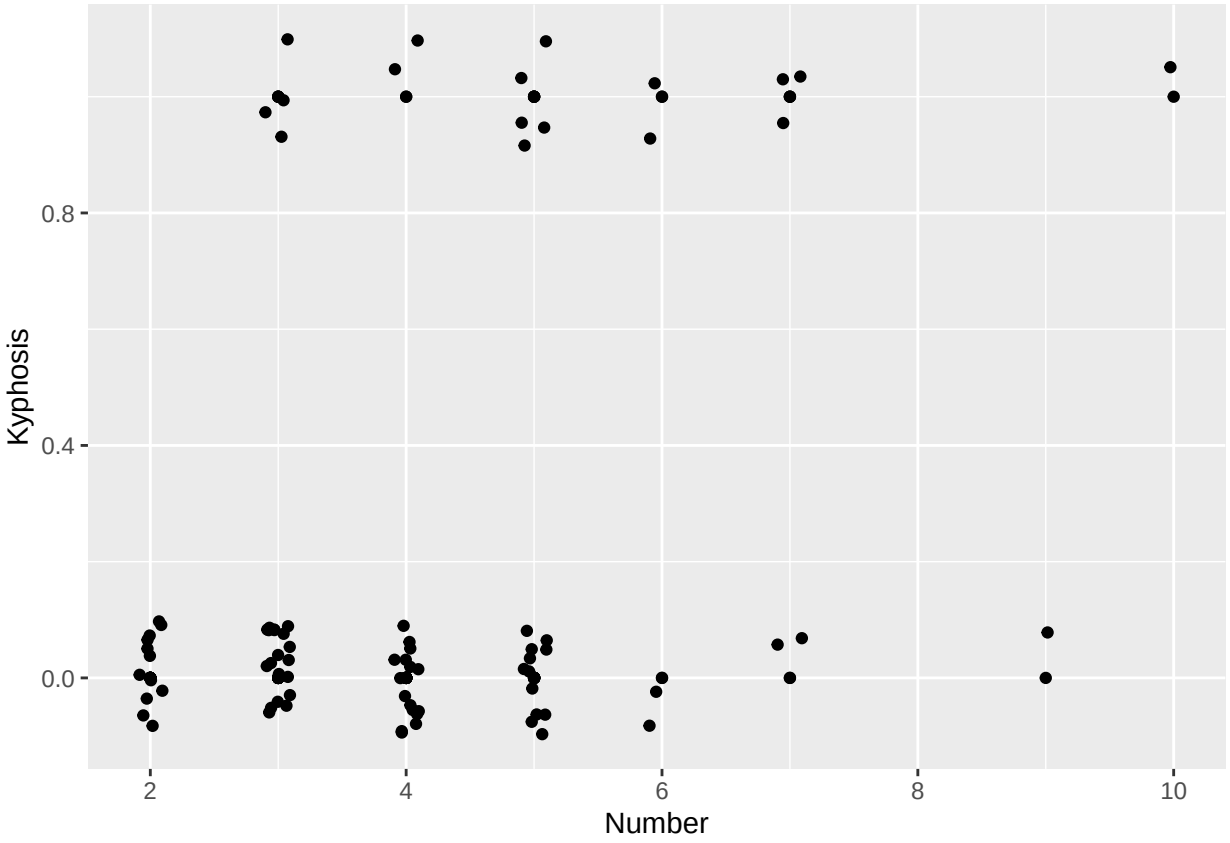


We can see that there's a huge difference between the median of the number of the first (topmost) vertebra operated on with absent and present Kyphosis. We can see that see that with absent Kyphosis, the median Start 13-14, and with present, the median Start is around 6-7.

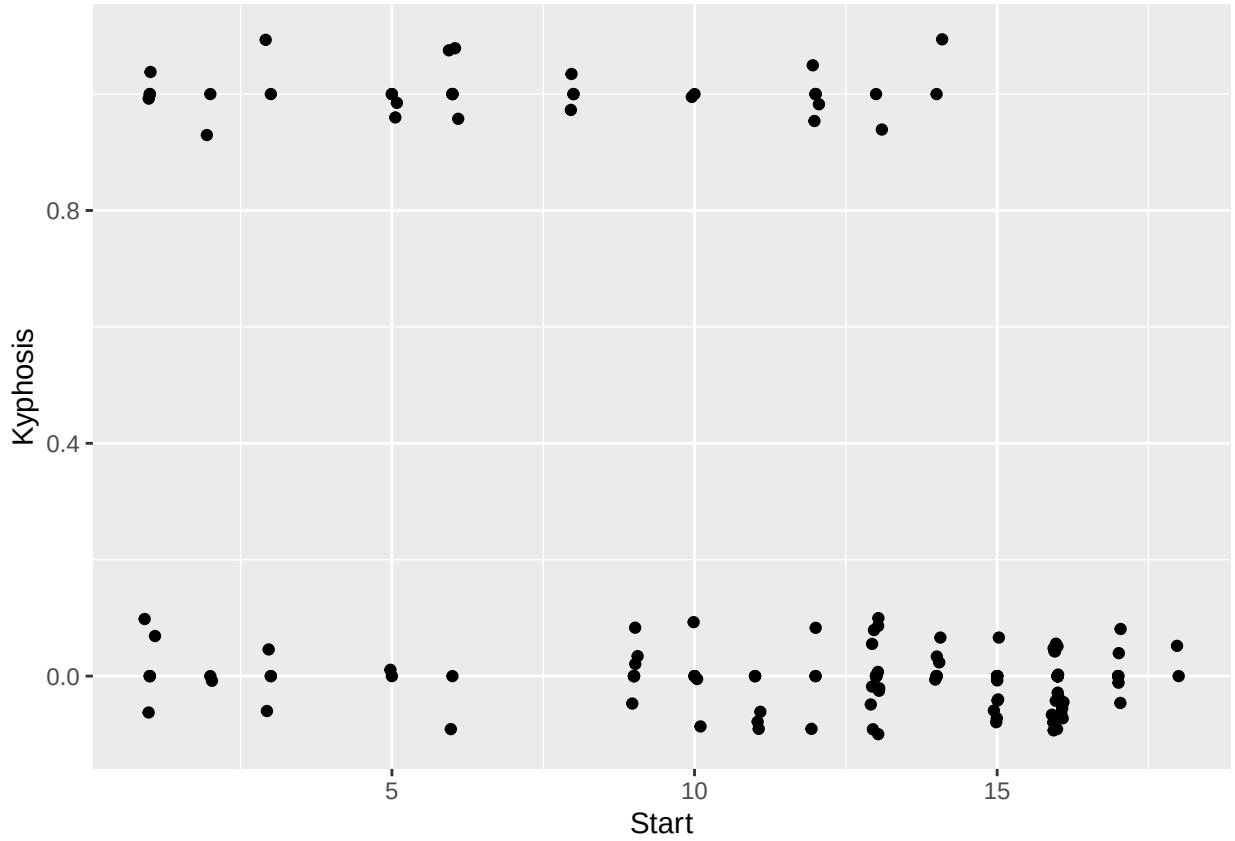
```
kyphosis$Kyphosis<-ifelse(kyphosis$Kyphosis == "absent", 0, 1)

ggplot(kyphosis, aes(x=Age, y= Kyphosis))+
  geom_point()+
  geom_jitter(width = 0.1, height = 0.1)
```




```
ggplot(kyphosis, aes(x=Number, y= Kyphosis))+
  geom_point()+
  geom_jitter(width = 0.1, height = 0.1)
```




```
ggplot(kyphosis, aes(x=Start, y= Kyphosis))+
  geom_point()+
  geom_jitter(width = 0.1, height = 0.1)
```



Based on these three scatterplots, the age predictor seems to indicate higher ages increases likelihood. However, it doesn't look strongly correlated while the number seems to indicate higher numbers increases likelihood. The Start predictor seems to indicate lower numbers increases likelihood.

```
model1<-glm(Kyphosis ~ Age + Number + Start, data = kyphosis, family = "binomial")
summary(model1)
```

```
##
## Call:
## glm(formula = Kyphosis ~ Age + Number + Start, family = "binomial",
##     data = kyphosis)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3124  -0.5484  -0.3632  -0.1659   2.1613
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.036934   1.449575  -1.405  0.15996
## Age          0.010930   0.006446   1.696  0.08996 .
## Number       0.410601   0.224861   1.826  0.06785 .
## Start       -0.206510   0.067699  -3.050  0.00229 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 83.234  on 80  degrees of freedom
## Residual deviance: 61.380  on 77  degrees of freedom
## AIC: 69.38
##
## Number of Fisher Scoring iterations: 5
```

Now we want to check with the F-test. Since there are not many variables in the data, we can check all the possibility of a reduced model. Therefore, we'll check all the reduced model with the F-test.

```
reduced_model1<-glm(Kyphosis ~ Start, data = kyphosis, family = "binomial")

reduced_model2<-glm(Kyphosis ~ Age, data = kyphosis, family = "binomial")

reduced_model3<-glm(Kyphosis ~ Number, data = kyphosis, family = "binomial")

reduced_model4<-glm(Kyphosis ~ Age + Number, data = kyphosis, family = "binomial")

reduced_model5<-glm(Kyphosis ~ Age + Start, data = kyphosis, family = "binomial")

reduced_model6<-glm(Kyphosis ~ Number + Start, data = kyphosis, family = "binomial")

anova(reduced_model1, model1, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: Kyphosis ~ Start
## Model 2: Kyphosis ~ Age + Number + Start
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        79     68.072
## 2        77     61.380  2   6.6923  0.03522 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(reduced_model2, model1, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: Kyphosis ~ Age
## Model 2: Kyphosis ~ Age + Number + Start
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1        79     81.932
## 2        77     61.380  2   20.553 3.444e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(reduced_model3, model1, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: Kyphosis ~ Number
## Model 2: Kyphosis ~ Age + Number + Start
```

```
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        79     73.357
## 2        77     61.380  2   11.977 0.002507 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(reduced_model4, model1, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: Kyphosis ~ Age + Number
## Model 2: Kyphosis ~ Age + Number + Start
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        78     71.627
## 2        77     61.380  1   10.247 0.001369 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(reduced_model5, model1, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: Kyphosis ~ Age + Start
## Model 2: Kyphosis ~ Age + Number + Start
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        78     65.299
## 2        77     61.380  1   3.9191  0.04774 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(reduced_model6, model1, test = "Chi")
```

```
## Analysis of Deviance Table
##
## Model 1: Kyphosis ~ Number + Start
## Model 2: Kyphosis ~ Age + Number + Start
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1        78     64.536
## 2        77     61.380  1   3.1565  0.07562 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Our Null Hypothesis H_0 : the smaller model is correct and our Alternative Hypothesis H_1 : at least one variable is related to response. After looking through all the F-test, we can see that the p-value is higher than 0.05 for the reduced model with variables Number and Start, so we'll fail to reject the Null Hypothesis, which means the reduced model is correct. Therefore, based on F-Test, the model with variables Number and Start is the best model.

$log(\frac{p_{present}}{1-p_{present}}) = \beta_0 + \beta_2 x_{Number} + \beta_3 x_{Start}$

```r
beta <- coef(reduced_model6)
exp(beta)
```

```
## (Intercept)      Number       Start
##   0.3573988   1.4296819   0.8311465
```

From the model, the odds of Kyphosis increases by 42.96819% with each additional number of vertebae involved, and decreases by 16.688535% for each addition number of the first (topmost) vertebra operated on.

```r
confint(reduced_model6)
```

```
## Waiting for profiling to be done...
```

```
##                   2.5 %      97.5 %
## (Intercept) -3.54539710  1.33695790
## Number      -0.01467272  0.78433678
## Start       -0.31799517 -0.06267191
```

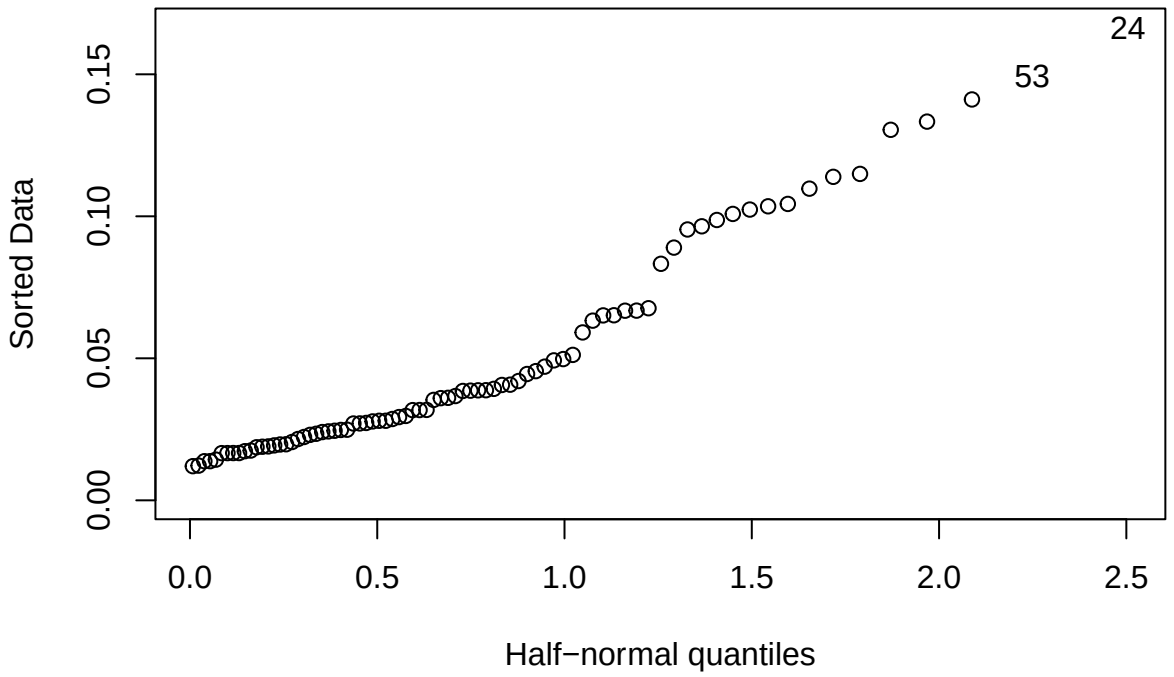
```r
backward_model<-step(model1, direction = "backward", trace = FALSE)
summary(backward_model)
```

```
##
## Call:
## glm(formula = Kyphosis ~ Age + Number + Start, family = "binomial",
##     data = kyphosis)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3124  -0.5484  -0.3632  -0.1659   2.1613
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.036934   1.449575  -1.405  0.15996
## Age          0.010930   0.006446   1.696  0.08996 .
## Number       0.410601   0.224861   1.826  0.06785 .
## Start       -0.206510   0.067699  -3.050  0.00229 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 83.234  on 80  degrees of freedom
## Residual deviance: 61.380  on 77  degrees of freedom
## AIC: 69.38
##
## Number of Fisher Scoring iterations: 5
```

We can see that according to the backward selection model, it is the same as the original model.

$log(\frac{p_{present}}{1-p_{present}}) = \beta_0 + \beta_1 x_{Age} + \beta_2 x_{Number} + \beta_3 x_{Start}$

```
halfnorm(hatvalues(model1))
```

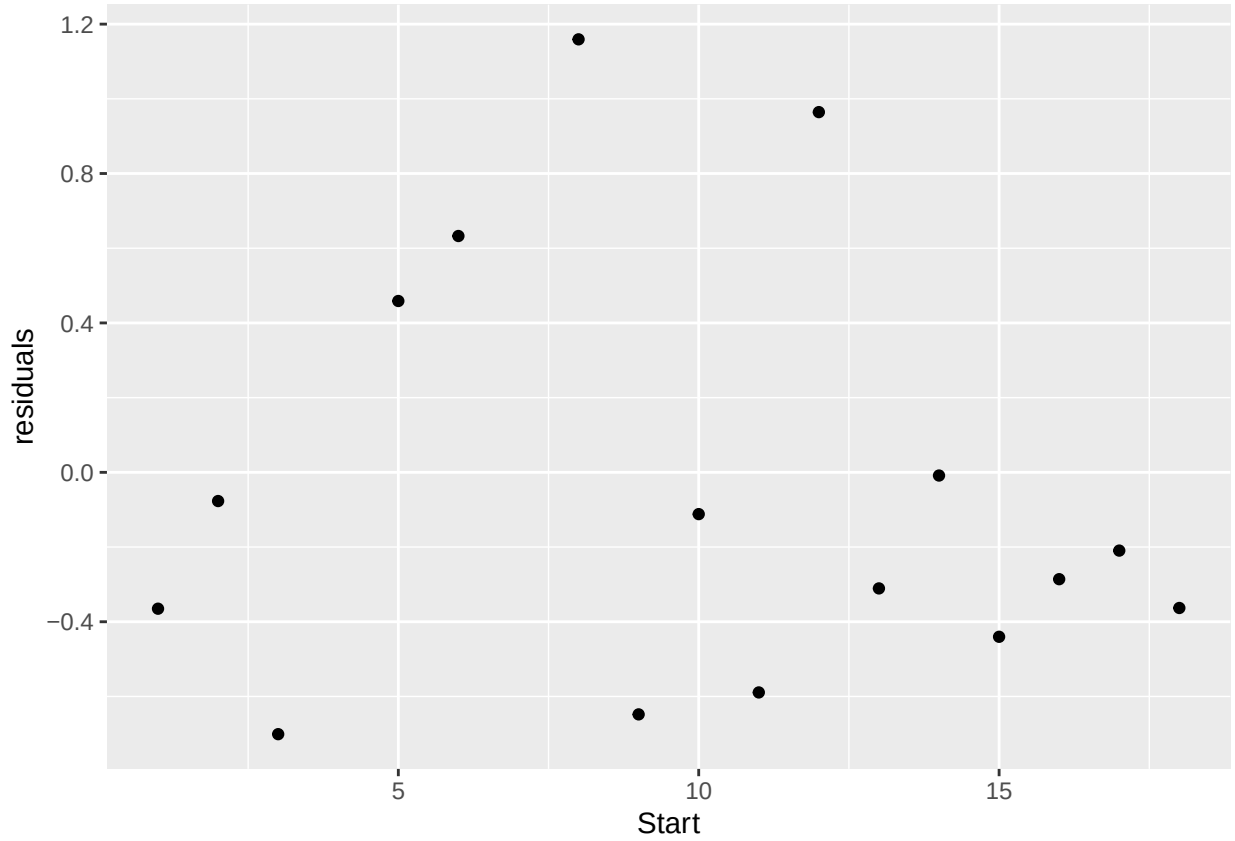



```
filter(kyphosis, hatvalues(model1) > 0.145) %>% select(Age, Number, Start, Kyphosis)
```

```
##   Age Number Start Kyphosis
## 1 131      2     3        0
## 2 139     10     6        1
```

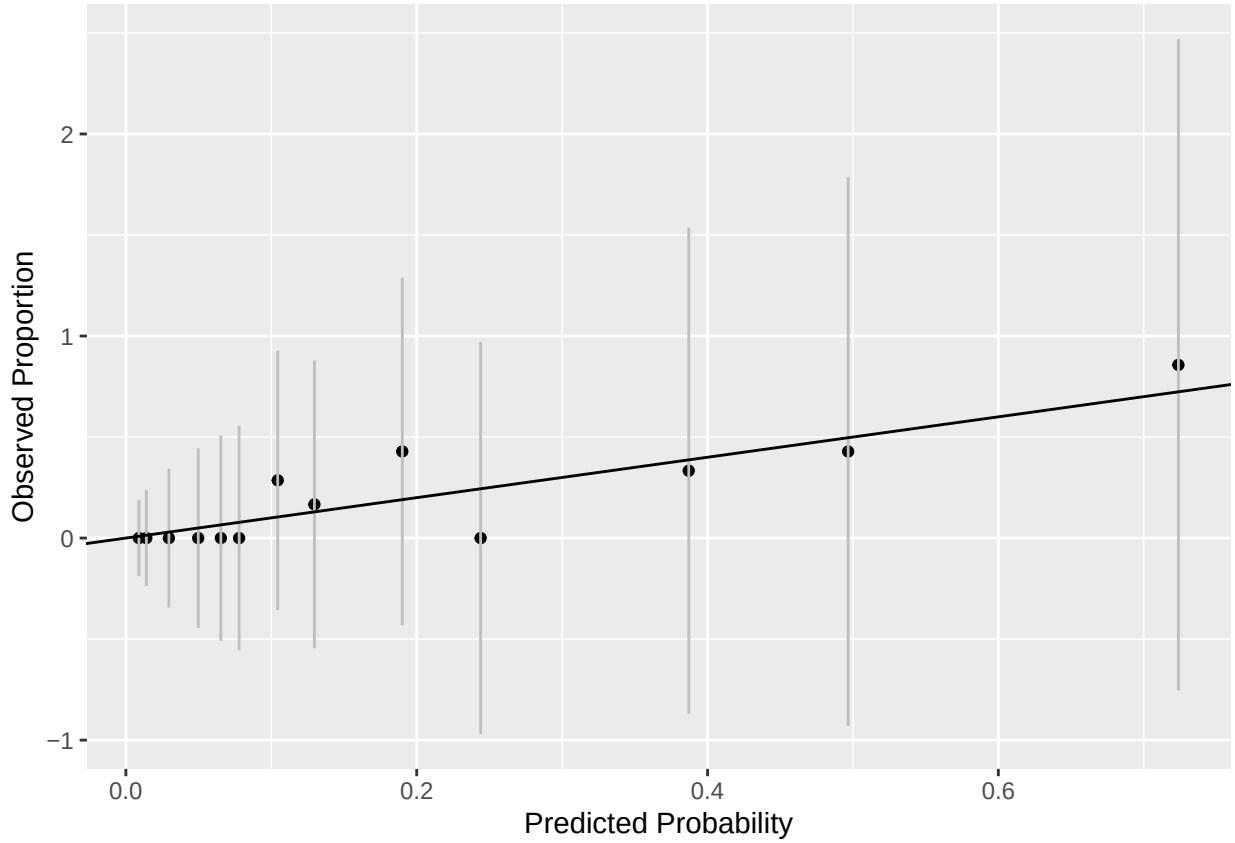
The leverage is a measure of how far away an observation's independent variable values are from the other observations. We can see that there are the two children age 131 and 139 months. For the child age 131 months has a very low number of Start, and child age 139 months has a very high Number and low Start. However, these two points do not seem particularly extreme, so we are not so concerned about these two observations.

```
kyphosis2<-mutate(kyphosis, residuals = residuals(model1), linpred = predict(model1))
gdf<-group_by(kyphosis2, Start)
diagdf<-summarise(gdf, residuals = mean(residuals))
ggplot(diagdf, aes(x = Start, y = residuals))+
  geom_point()
```



We can see that the residuals is pretty random. Although we can see that the Start value between 5 and 10 has a really high residual. We can see that at Start value equal 8, the residuals is around 1.2. However, it's not huge enough to be a concern.

```
linPred <- predict(model1)
kDataM <- mutate(kyphosis, predProb = predict(model1, type = "response"))
gDf <- group_by(kDataM, cut(linPred, breaks = unique(quantile(linPred, (0:12)/12))))

hlDf <- summarise(gDf, y= sum(Kyphosis), pPred=mean(predProb), count = n())

hlDf <- mutate(hlDf, se.fit=sqrt(pPred * (1-(pPred)/count)))


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



The plot shows 95% confidence intervals and our line fits through all of them, so this is a sign that it is a good fit.

```
linpred <- predict(model1)
data <- mutate(kyphosis, predProb = predict(model1, type = "response"))
gDf <- group_by(data, cut(linpred, breaks = unique(quantile(linpred, (0:12)/12))))

hldf <- summarise(gDf,
                  y = sum(Kyphosis),
                  ppred = mean(predProb),
                  count = n())

hldf<-mutate(hldf, se.fit = sqrt(ppred*(1-(ppred))/count))

hlStat<- with(hldf, sum( ( y- count* ppred)^2/(count*ppred*(1-ppred))))

hlStat
```

```
## [1] 9.887868
```

```
1-pchisq(hlStat, nrow(hldf)-1)
```

```
## [1] 0.6257973
```

Our Null Hypothesis H_0 : the data is poor fit for logistic regression and our Alternative Hypothesis H_1 : the data is not a poor fit for logistic regression. Therefore, we can see that this data produces a p-value of 0.6257973, and this indicates that there's no lack of fit.

```
data<-mutate(data, Predicted = ifelse(predProb < 0.5, 0, 1))
table(predicted = data$Predicted, data$Kyphosis)
```

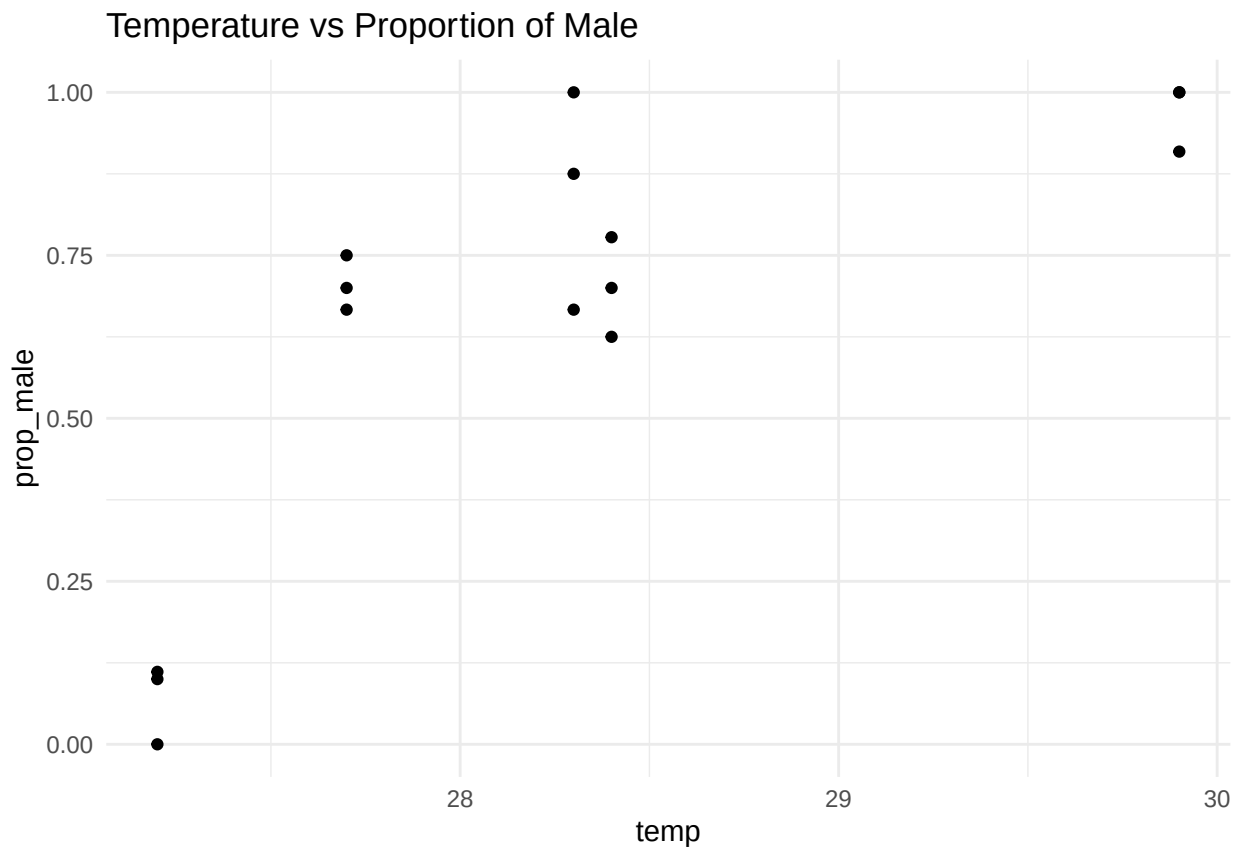
```
##
## predicted  0  1
##         0 61 10
##         1  3  7
```

Overall, the model is around 84% accurate. We can see that this model is poor in predicting children with Kyphosis as we see that it got more than half of it wrong. It did well for predicting absent Kyphosis with a misclassification rate of 0.046875, but for the present Kyphosis, it has a misclassification rate of 0.588235. We can see that from the data, the model tend to predict 1 when Start is low. We can see that when Start is like 10 or 17, it predicts 0, but when Start is 6, it predicts 1. This means that it is sensitive to the variable Start, as this heavily influence the prediction of the model. We can see that variables age and number doesn't seem to have an apparent sign that affect the model's prediction. The sensitivity metric is about 41% and specificity metric is about 95%.

Problem 2 - Binomial Responses

```
data(turtle, package = "faraway")

turtle$prop_male<-ifelse(turtle$female == 0, 1, (turtle$male)/(turtle$male + turtle$female))
```

```
ggplot(turtle, aes(x = temp, y = prop_male))+
  geom_point()+
  labs(title = "Temperature vs Proportion of Male")+
  theme_minimal()
```



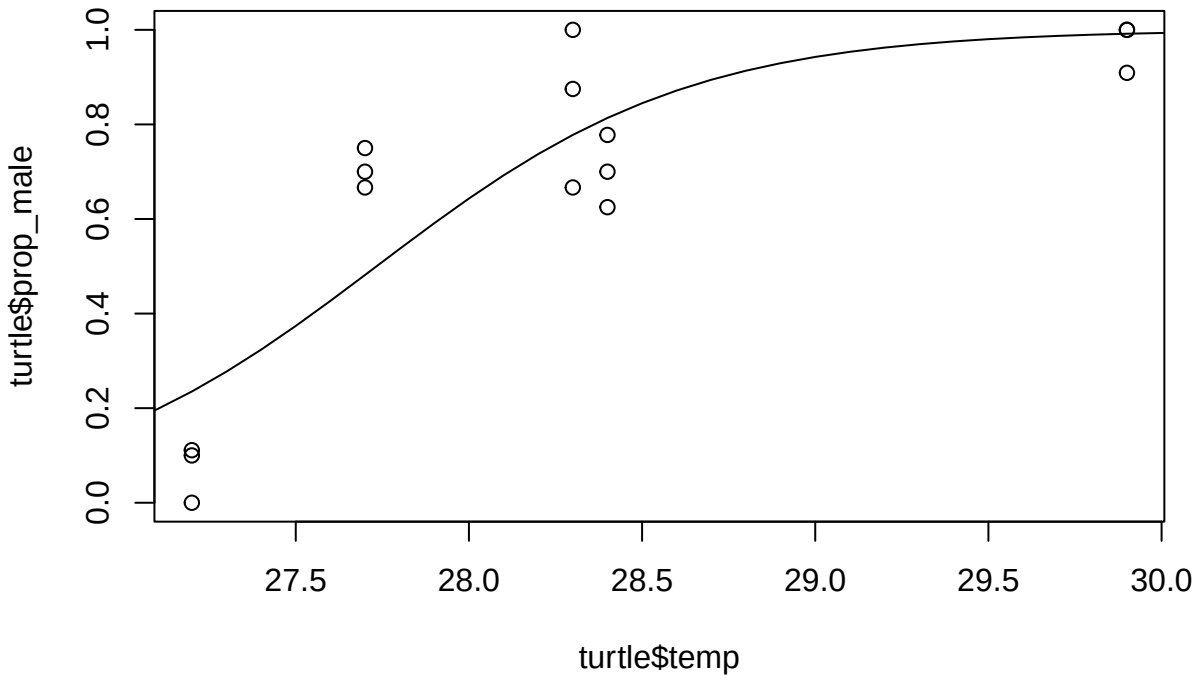
We can see that there's a clear relationship where as the temperature increases, the proportion of male turtle increases.

```
model1<-glm(cbind(male, female) ~ temp, data = turtle, family = "binomial"(link = "logit"))
summary(model1)
```

```
##
## Call:
## glm(formula = cbind(male, female) ~ temp, family = binomial(link = "logit"),
##     data = turtle)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0721  -1.0292  -0.2714   0.8087   2.5550
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -61.3183    12.0224  -5.100 3.39e-07 ***
## temp          2.2110     0.4309   5.132 2.87e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 74.508  on 14  degrees of freedom
## Residual deviance: 24.942  on 13  degrees of freedom
```

```
## AIC: 53.836
##
## Number of Fisher Scoring iterations: 5
```

```
x<-seq(27, 30.2, 0.1)
plot(turtle$temp, turtle$prop_male)
lines(x, ilogit(-61.3183 + 2.2110*x))
```




```
deviance(model1)
```

```
## [1] 24.94249
```

```
df.residual(model1)
```

```
## [1] 13
```

```
pchisq(deviance(model1), df.residual(model1), lower = FALSE)
```

```
## [1] 0.02348863
```

We fit the model with male_prop as target variable and temp as the predictor. We use logit as the link function.

$log(\frac{p_{male}}{1-p_{male}}) = \beta_0 + \beta_1 x_{temp}$

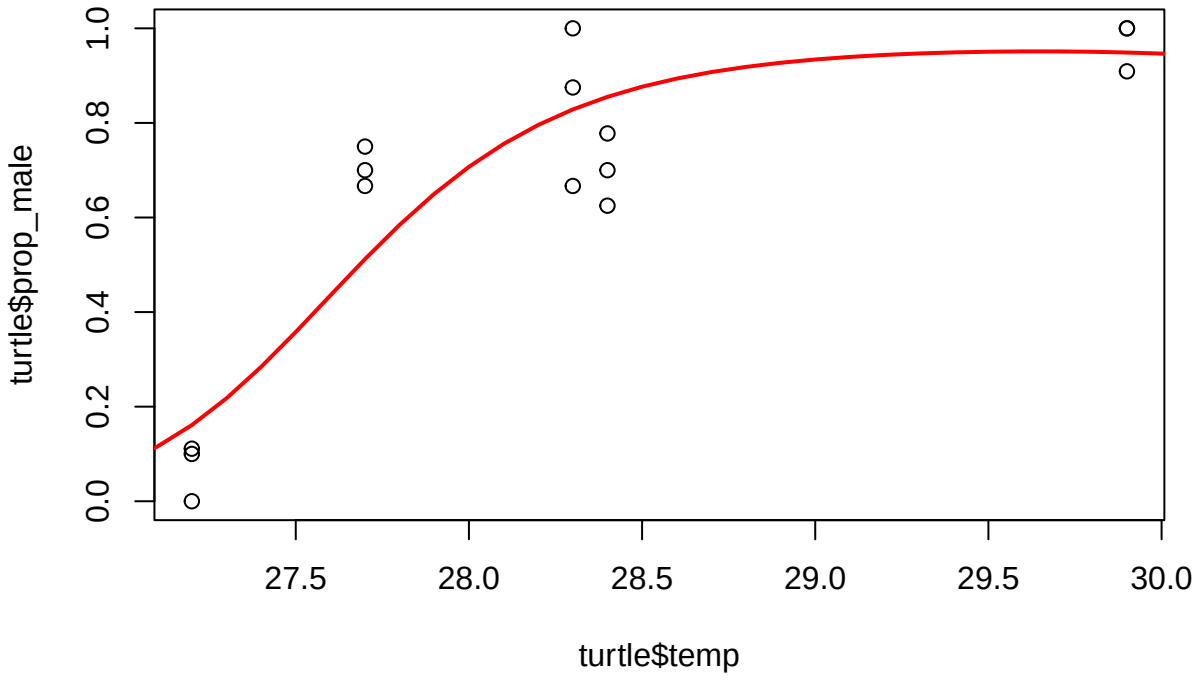
In addition, we can see that the p-value is less than 0.05, so we can conclude that this model does not fit sufficiently well.

The data is space, where there are only 15 observations, and there is essentially one predictor (male_prob). According to the plot, there are no strong outliers. We can clearly see that there is a fit according to the plot, but it could be improved.

```
model2<-glm(cbind(male, female) ~ temp + I(temp^2), data = turtle, family = "binomial"(link = "logit"))
summary(model2)
```

```
##
## Call:
## glm(formula = cbind(male, female) ~ temp + I(temp^2), family = binomial(link = "logit"),
##     data = turtle)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6703  -0.8875  -0.4194   0.9481   2.2198
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -677.5950   268.7984  -2.521   0.0117 *
## temp          45.9173    18.9169   2.427   0.0152 *
## I(temp^2)     -0.7745     0.3327  -2.328   0.0199 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 74.508  on 14  degrees of freedom
## Residual deviance: 20.256  on 12  degrees of freedom
## AIC: 51.15
##
## Number of Fisher Scoring iterations: 4
```

```
plot(turtle$temp, turtle$prop_male)
lines(x, ilogit(-677.595 + (45.9173 * x) - (0.7745 * x^2)), col = "red", lwd = 2)
```




```
deviance(model2)
```

```
## [1] 20.25621
```

```
df.residual(model2)
```

```
## [1] 12
```

```
pchisq(deviance(model2), df.residual(model2), lower = FALSE)
```

```
## [1] 0.06239194
```

Now, we have the quadratic model, which we add additional term of temp squared.

$log(\frac{p_{male}}{1-p_{male}}) = \beta_0 + \beta_1 x_{temp} + \beta_2 x_{temp}^2$

We can see that the deviance for this model is better with p-value of 0.06239194. Since the p-value is greater than 0.05, we can conclude that it fits well. We can also see that the temp^2 predictor has a 0.0199 p-value, so it is a significant predictor.

```
model3<-glm(cbind(male, female) ~ temp + I(temp^2), data = turtle, family = "binomial"(link = "inverse")
summary(model3)
```

```
##
## Call:
## glm(formula = cbind(male, female) ~ temp + I(temp^2), family = binomial(link = "inverse"),
##     data = turtle)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4487  -1.0707  -0.2828   0.9452   2.7199
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 482.3411   190.4478   2.533   0.0113 *
## temp        -32.9121    13.1027  -2.512   0.0120 *
## I(temp^2)     0.5624     0.2252   2.497   0.0125 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 74.508  on 14  degrees of freedom
## Residual deviance: 29.703  on 12  degrees of freedom
## AIC: 60.596
##
## Number of Fisher Scoring iterations: 11
```

```
deviance(model3)
```

```
## [1] 29.7026
```

```
df.residual(model3)
```

```
## [1] 12
```

```
pchisq(deviance(model3), df.residual(model3), lower = FALSE)
```

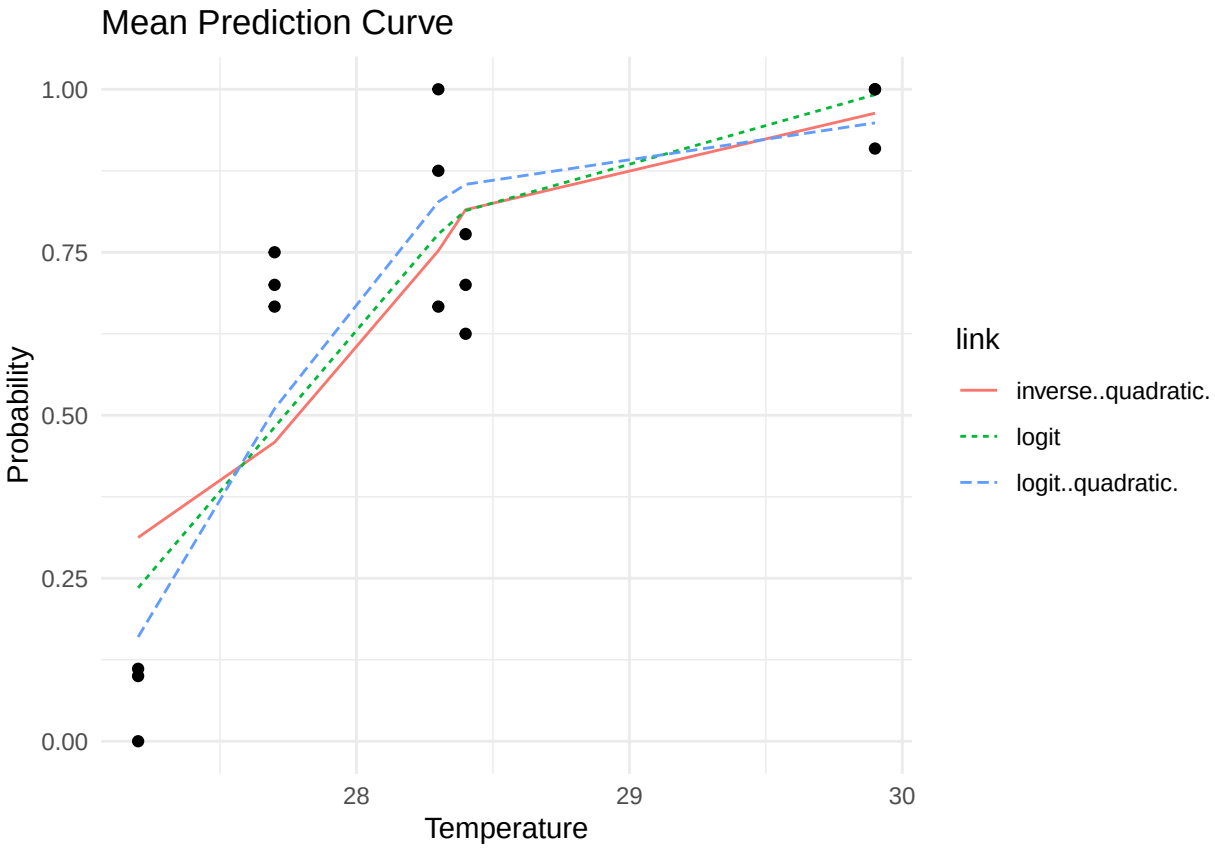
```
## [1] 0.003095006
```

Now, we'll build a model with the link function of inverse. $\frac{1}{p_{male}} = \beta_0 + \beta_1 x_{temp} + \beta_2 x_{temp}^2$

We can see that this model is significantly worse that the link function logit. With the p-value significantly lower the 0.05, it indicates that the model does not fit well.

```
temp_seq<-c(27.2, 27.2, 27.2, 27.7, 27.7, 27.7, 28.3, 28.3, 28.3, 28.4, 28.4, 28.4, 29.9, 29.9, 29.9)
predval<-sapply(list(model1, model2, model3), function(m) predict(m, data.frame(temp = temp_seq), type =

colnames(predval)<-c("logit", "logit (quadratic)", "inverse (quadratic)")

predval<-data.frame(temp_seq, predval)

mpv<-gather(data = predval, key = link, value = probability, -temp_seq)

ggplot() +
```

```
  geom_line(data = mpv, aes(x = temp_seq, y = probability, linetype = link, color = link)) +
  labs(x = "Temperature", y = "Probability", title = "Mean Prediction Curve") +
  geom_point(data = turtle, aes(x = temp, y = prop_male), color = "black")+
  theme_minimal()
```



We can see that logit and logit (quadratic) is pretty similar except that logic (quadratic) is better at capturing data that is really low and high. The inverse clearly is not as good as the logit. It is unable to capture the low data at like temperature 27.

Now, we want to write the model with two parameters, β_0 and β_1, and find their maximum likelihood estimates. The binomial likelihood (log-likelihood) for original logit model is:

$\ell(\beta) = \sum_{i=1}^{n}[y_i\eta_i \log(1 + e^{\eta_i})\binom{m_i}{y_i}]$

Since we only have 81 data points and 2 predictors, we can rewrite it as:

$\ell(\beta) = \sum_{i=1}^{81}[y_i\eta_i \log(1 + e^{\eta_i})\binom{m_i}{y_i}]$

with $\eta_i = \log(\frac{p_i}{1-p_i})$.

Therefore, we can rewrite it as: $\ell(\beta) = \sum_{i=1}^{81}[y_i \log(p_i) + (m_i - y_i)\log(1 - p_i) + \log\binom{m_i}{y_i}]$