P53 Data Analysis

Alyssa Rogers-Armstrong

February 2024

```
Data = read.csv("p53.csv")
View(Data)
library(ggplot2)
library(magrittr)
library(dplyr)
```

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

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

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

```
library(readr)
library(dplyr)
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.
```

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
library(plotROC)
```

```
##
## Attaching package: 'plotROC'
```

```
## The following object is masked from 'package:pROC':
##
##     ggroc
```

```
library(caret)
```

```
## Loading required package: lattice
```

Exploratory Analysis

```
# Display the structure of the data
str(Data)
```

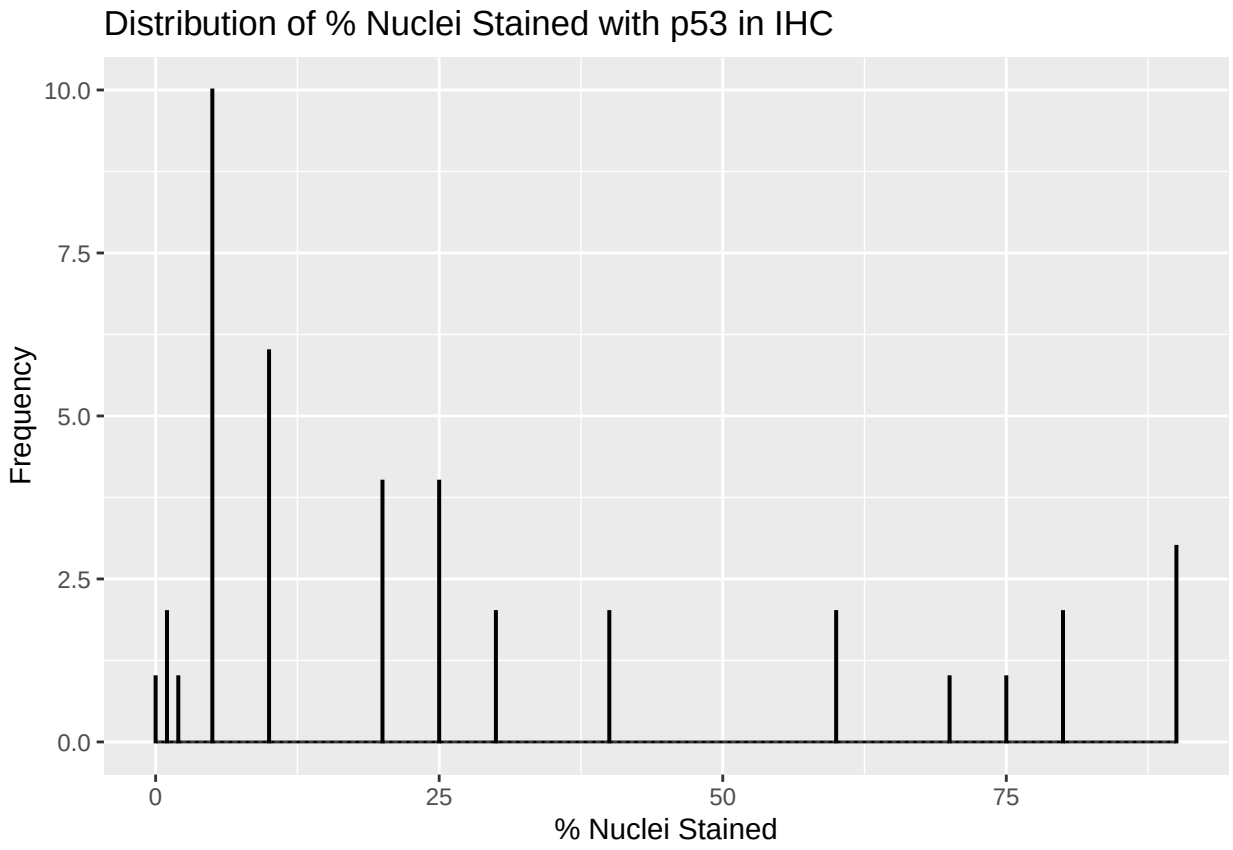
```
## 'data.frame':    41 obs. of  8 variables:
##  $ Sample                                    : chr  "CGB 1" "CGB 2" "CGB 4" "CGB 5" ...
##  $ proportion.of.nuclei.stained.with.p53.in.IHC: num  0.9 0.05 0.1 0.25 0.3 0.05 0.02 0.8 0.05 0.05
##  $ Mutation                                  : chr  "H179R" "no mutation" "no mutation" "no mutati
##  $ IHC.P53.Status                            : chr  "Positive" "Negative" "Positive" "Positive" ..
##  $ TP53.Sanger.Status                        : chr  "Positive" "Negative" "Negative" "Negative" ..
##  $ TP53.Sanger.Status..0.1.                  : int  1 0 0 0 0 0 0 1 0 0 ...
##  $ Comments                                  : chr  "" "" "" "" ...
##  $ Percent.Nuclei.Stained.in.p53.IHC         : int  90 5 10 25 30 5 2 80 5 5 ...
```

```
# Summary statistics
summary(Data)
```

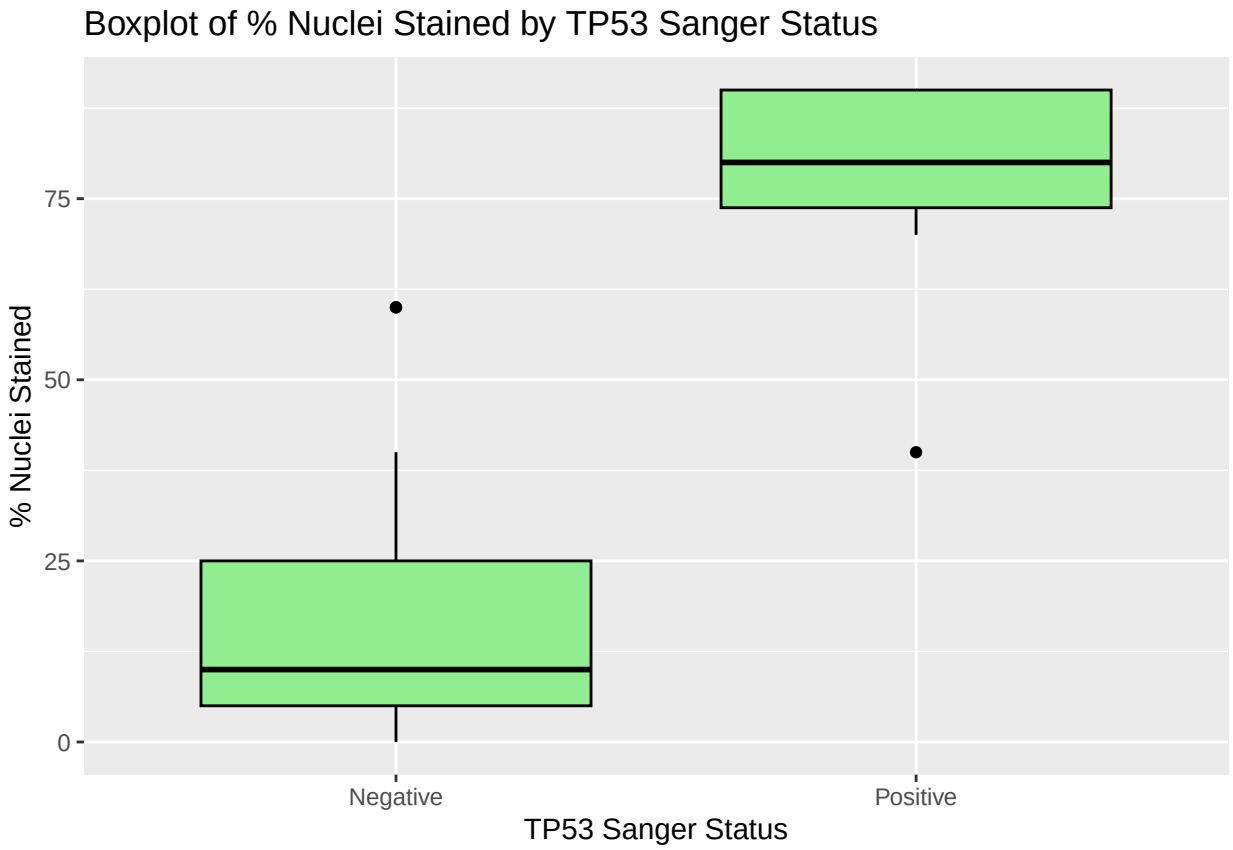
```
##     Sample          proportion.of.nuclei.stained.with.p53.in.IHC
##  Length:41          Min.   :0.0000
##  Class :character   1st Qu.:0.0500
##  Mode  :character   Median :0.2000
##                     Mean   :0.2754
##                     3rd Qu.:0.4000
##                     Max.   :0.9000
##    Mutation         IHC.P53.Status     TP53.Sanger.Status
##  Length:41          Length:41          Length:41
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##  TP53.Sanger.Status..0.1.   Comments         Percent.Nuclei.Stained.in.p53.IHC
##  Min.   :0.0000           Length:41          Min.   : 0.00
##  1st Qu.:0.0000           Class :character   1st Qu.: 5.00
##  Median :0.0000           Mode  :character   Median :20.00
```

```
##  Mean   :0.1951                          Mean   :27.54
##  3rd Qu.:0.0000                          3rd Qu.:40.00
##  Max.   :1.0000                          Max.   :90.00
```

```r
# Histogram of the percentage of nuclei stained with p53 in IHC
ggplot(Data, aes(x = Percent.Nuclei.Stained.in.p53.IHC)) +
  geom_histogram(binwidth = 0.1, fill = "skyblue", color = "black") +
  labs(title = "Distribution of % Nuclei Stained with p53 in IHC",
       x = "% Nuclei Stained", y = "Frequency")
```




```r
# Boxplot of % Nuclei Stained by TP53 Sanger Status
ggplot(Data, aes(x = as.factor(TP53.Sanger.Status), y = Percent.Nuclei.Stained.in.p53.IHC)) +
  geom_boxplot(fill = "lightgreen", color = "black") +
  labs(title = "Boxplot of % Nuclei Stained by TP53 Sanger Status",
       x = "TP53 Sanger Status", y = "% Nuclei Stained")
```



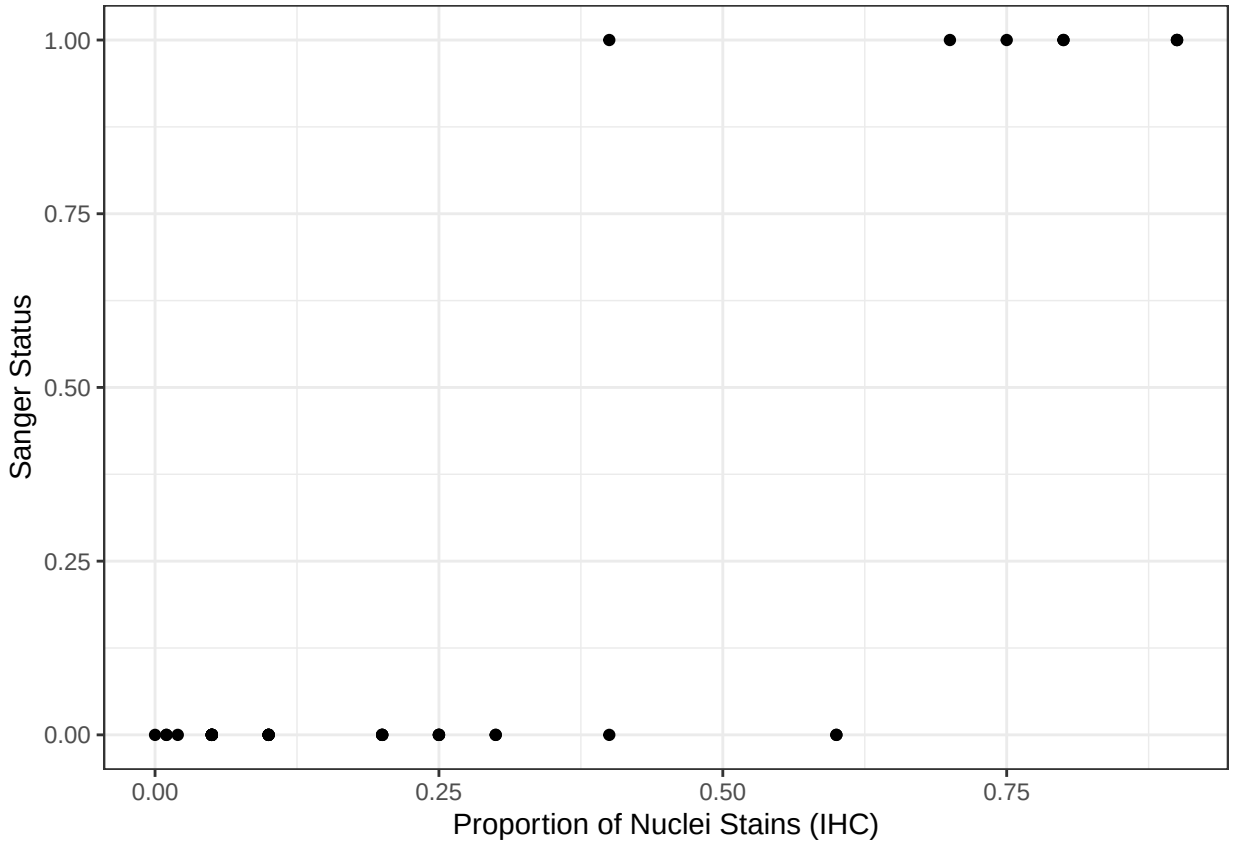

```
# Outlier detection
outliers <- boxplot.stats(Data$Percent.Nuclei.Stained.in.p53.IHC)$out
print(outliers)
```

```
## integer(0)
```

```
# Missing values
missing_values <- colSums(is.na(Data))
print(missing_values)
```

```
##                                     Sample
##                                          0
## proportion.of.nuclei.stained.with.p53.in.IHC
##                                          0
##                                   Mutation
##                                          0
##                              IHC.P53.Status
##                                          0
##                          TP53.Sanger.Status
##                                          0
##                     TP53.Sanger.Status..0.1.
##                                          0
##                                   Comments
##                                          0
##           Percent.Nuclei.Stained.in.p53.IHC
##                                          0
```

```r
# Scatterplot
ggplot(Data, aes(x=proportion.of.nuclei.stained.with.p53.in.IHC, y=TP53.Sanger.Status..0.1.))+geom_poin
  labs(x="Proportion of Nuclei Stains (IHC)", y="Sanger Status")
```



Statistical Analysis

Creating Factors:

```r
Data$TP53.Sanger.Status = as.factor(Data$TP53.Sanger.Status)
Data$TP53.Sanger.Status..0.1. = as.factor(Data$TP53.Sanger.Status..0.1.)
Data$IHC.P53.Status = as.factor(Data$IHC.P53.Status)
```

Fit the Model:

```r
# Fit the logistic regression model
logmodel <- glm(TP53.Sanger.Status..0.1. ~ Percent.Nuclei.Stained.in.p53.IHC, data=Data, family = binom
summary(logmodel)
```

```
##
## Call:
## glm(formula = TP53.Sanger.Status..0.1. ~ Percent.Nuclei.Stained.in.p53.IHC,
##     family = binomial, data = Data)
```

```
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                      -7.06358    2.68180  -2.634  0.00844 **
## Percent.Nuclei.Stained.in.p53.IHC  0.12120    0.04439   2.730  0.00633 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 40.4723  on 40  degrees of freedom
## Residual deviance:  9.5604  on 39  degrees of freedom
## AIC: 13.56
##
## Number of Fisher Scoring iterations: 7
```

- A one unit increase in Percent of Nuclei Stained is associated with an average increase of 0.12120 in the log odds of TP53 Sanger Status.

Calculating McFadden's R-Square

```
# Calculate McFadden's R-Squared

pscl::pR2(logmodel)["McFadden"]
```

```
## fitting null model for pseudo-r2

##  McFadden
## 0.7637797
```

- We can compute a metric known as McFadden's R2, which ranges from 0 to just under 1. Values close to 0 indicate that the model has no predictive power. In practice, values over 0.40 indicate that a model fits the data very well.
- McFadden = 0.7637797
- A value of 0.7637797 is very high for McFadden's R2, which indicates that our model fits the data very well and has high predictive power.

Odds Ratio

- Model Interpretation: Percent Nuclei Stained (p = 0.00633) is significantly associated with the TP53 Sanger Status at the 0.05 level.
- Odds Ratio Interpretation: For each additional percentage of P53 nuclei stained in IHC, there is a 12.8% increased odds of a TP53 mutation.

```
or_logmodel = exp(logmodel$coefficients)
ci_logmodel = exp(confint(logmodel))
```

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

```
orci_logmodel = cbind(or_logmodel, ci_logmodel)
orci_logmodel
```

```
##                                    or_logmodel        2.5 %     97.5 %
## (Intercept)                       0.0008557107 1.495201e-07 0.02854974
## Percent.Nuclei.Stained.in.p53.IHC 1.1288505669 1.061685e+00 1.29729511
```

Confusion Matrix at 10% Threshold

- This is telling us that at the 10% threshold, there are a lot of false positives. This would waste a lot of resources by further testing these patients when they are true negatives.

```
confusionMatrix(Data$IHC.P53.Status, reference=Data$TP53.Sanger.Status, positive='Positive')
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Negative Positive
##   Negative       14        0
##   Positive       19        8
##
##                Accuracy : 0.5366
##                  95% CI : (0.3742, 0.6934)
##     No Information Rate : 0.8049
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2233
##
##  Mcnemar's Test P-Value : 3.636e-05
##
##             Sensitivity : 1.0000
##             Specificity : 0.4242
##          Pos Pred Value : 0.2963
##          Neg Pred Value : 1.0000
##              Prevalence : 0.1951
##          Detection Rate : 0.1951
##    Detection Prevalence : 0.6585
##       Balanced Accuracy : 0.7121
##
##        'Positive' Class : Positive
##
```

Optimal Cutpoint

```
library(cutpointr)
```

```
##
## Attaching package: 'cutpointr'
```

```
## The following objects are masked from 'package:caret':
##
##     precision, recall, sensitivity, specificity

## The following objects are masked from 'package:pROC':
##
##     auc, roc
```

```
optimal = cutpointr(Data, Percent.Nuclei.Stained.in.p53.IHC, TP53.Sanger.Status..0.1., method = maximiz
```

```
## Assuming the positive class is 1

## Assuming the positive class has higher x values
```

```
summary(optimal)
```

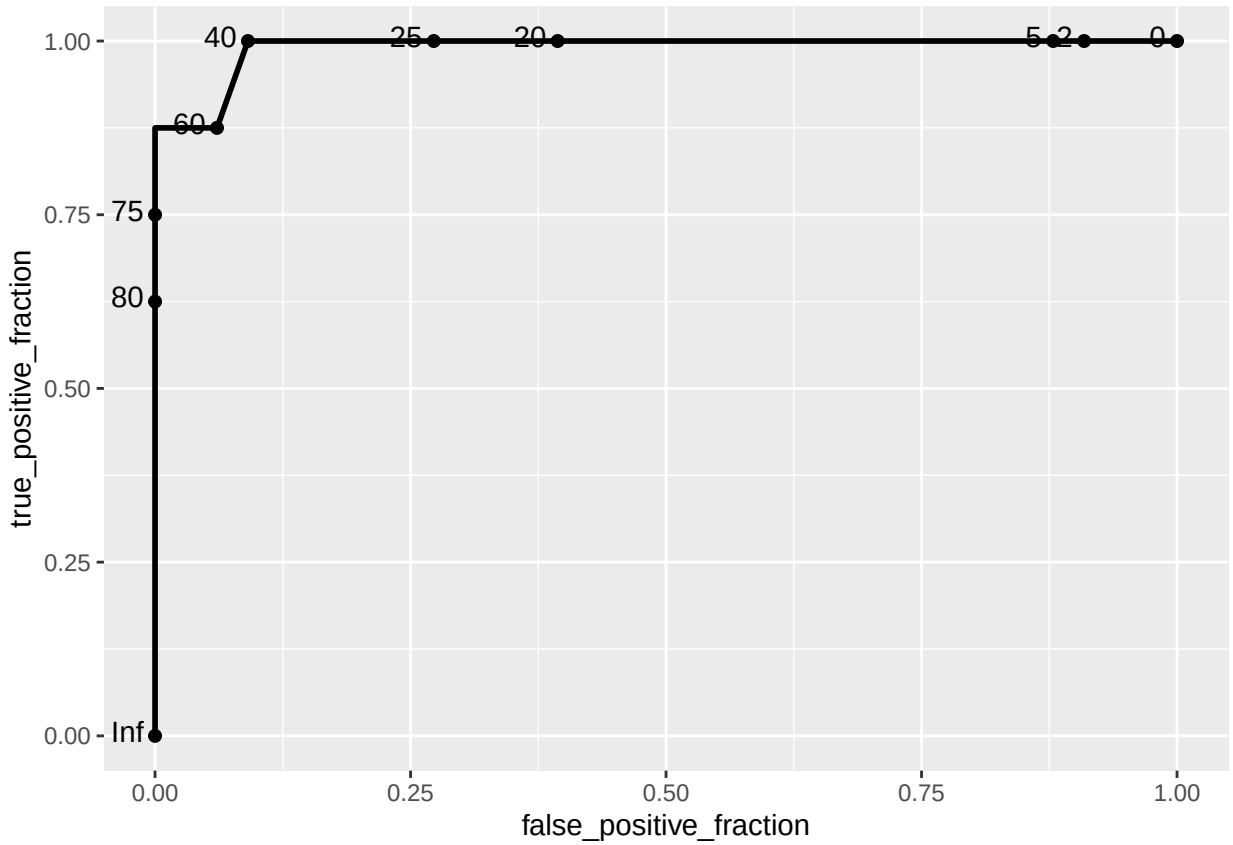
```
## Method: maximize_metric
## Predictor: Percent.Nuclei.Stained.in.p53.IHC
## Outcome: TP53.Sanger.Status..0.1.
## Direction: >=
##
##     AUC  n n_pos n_neg
##  0.9905 41     8    33
##
##  optimal_cutpoint sum_sens_spec    acc sensitivity specificity tp fn fp tn
##                40        1.9091 0.9268           1      0.9091  8  0  3 30
##
## Predictor summary:
##     Data Min.   5% 1st Qu. Median     Mean 3rd Qu. 95% Max.       SD NAs
##  Overall    0  1.0    5.00     20 27.53659      40  90   90 29.02249   0
##        0    0  1.0    5.00     10 15.57576      25  48   60 15.36032   0
##        1   40 50.5   73.75     80 76.87500      90  90   90 16.67708   0
```

ROC Plot

```
# Create ROC curve

basic.roc = ggplot(Data, aes(d=TP53.Sanger.Status, m=Percent.Nuclei.Stained.in.p53.IHC))+geom_roc()
basic.roc
```

```
## Warning in verify_d(data$d): D not labeled 0/1, assuming Negative = 0 and
## Positive = 1!
```


1.00
0.75
0.50
0.25
0.00
true_positive_fraction
40
25
20
5
2
0
60
75
80
Inf
0.00
0.25
0.50
0.75
1.00
false_positive_fraction