Analysis of Factors Related to Tumor Penetration of Prostatic Capsule

Andrew Bates

November 15, 2018

Executive Summary

In this paper we investigate various prostate exam results and their relationship with tumor pentration of the prostatic capsule. A logistic regression model is constructed based on a subset of data collected by the Ohio State University Comprehensive Cancer Center in order to quantify the relationship. The components of the model are the results of a digital rectal exam, total Gleason score, and prostate specific antigen value. The most significant factor is whether a nodule was detected in a digital rectal exam. Presence of a nodule is associated with up to a 4.7 times increase in odds of tumor pentration compared to absence of a nodule. Provided the other variables are fixed, a one unit increase in Gleason score is associated with a 2.7 times increase in odds of tumor penetration. Prostate specific antigen value is related to tumor penetration via a 3% increase in odds of penetration for each additional mg/ml increase. We also assess the potential of using the model as an overall diagnostic tool that combines the results of a digital rectal exam, prostate specific antigen value and Gleason score into a single measure of risk, rather than considering each measure individually.

1 Introduction

Prostate cancer is one of the most common cancers among males. Early detection is crucial because treatment can often be successful if the cancer is found early enough to where a spread to other parts of the body have not occured. In this paper we investigate the relationship between the results of various prostate related exams and whether tumor penetration of the prostatic capsule has occured. The primary purpose of this analysis is to determine if there are any factors that have a particularly influential relationship with prostate capsule penetration. A secondary aim is to develop a method to use the model resulting from the primary analysis as a diagnostic aid that merges the results of several exams into a single risk measure instead of considering each result independently.

2 Methods

In this analysis we examine a subset of data collected by the Ohio State University Comprehensive Cancer Center as part of a study to determine the potential of standard exam results to predict whether a tumor will penetrate the prostatic capsule. Out of 380 patients, 153 have experienced capsule penetration and 227 have not. The data set contains six explanatory variables: the patients age and race, results of a digital rectal exam (DRE), whether capsular involvement was detected, prostate specific antigen (PSA) value, and total Gleason score. Also included in the data set was an identification code which is not used in the analysis as it simply identifies the patient. For a detailed description of each variable see Table 1. There are two continuous variables, PSA and age, and the remaining variables are categorical. Observe that race is recorded as only Black or White. This may be the reason why three observations contain missing values for race. Perhaps

three of the patients were neither Black nor White. However, without access to the details of the study we can only speculate. Furthermore, the other variables in these observations do not point to any particular reason as to why race is not recorded. Because of this, we decided to omit these three observations. The data set used for analysis then, consists of 377 observations. Of these, 151 patients have experienced tumor penetration of the prostatic capsule, and 226 have not.

Table 1: Detailed description of variables included in the data set (except id). The two continuous variables are age and prostate specific antigen value. The remaining variables are categorical.

Name	**Description**	**Details**
penetrate	Tumor penetration of prostatic capsule?	Yes, no
age	Patient age	Years
race	Patient race	Black, White
dre	Results of digital rectal exam	No nodule, unilobar left, unilobar right, bilobar
caps	Detection of capsular involvement?	Yes, no
psa	Prostate specific antigen value	mg/ml
gleason	Total Gleason score	0 - 10

To investigate the relationship between the covariates and tumor penetration status, we use a logistic regression model. The response is especially well balanced with 40% of the observations having capsular penetration and 60% not having penetration. As such, procedures designed to assist with class imbalances, such as downsampling, are not considered in this analysis. In addition to the explanatory variables included in the data set, all two-way interaction terms are examined. Using this as a starting point, the model is chosen via a backwards elimination procedure using Akaike Information Criteria (AIC) as the model selection criterion. In the model constructed with this procedure age and race are not significant at the 0.05 level so they are subsequently removed along with any associated interaction terms.

Diagnostics of the resulting model are performed via explanatory variable patterns and a Hosmer-Lemeshow goodness-of-fit test. The strength of the relationship between the covariates and tumor penetration of the prostatic capsule is measured by the odds ratio. The predictive power of the model is evaluated by the area under the receiver operating characteristic curve. A method based on predictions from the model is proposed as a tumor penetration diagnostic tool to assist physicians in determining if further investigation of a given patient is warranted. The analysis is conducted using the statistical software R and the report is written using the R Markdown package. The R code is available in the appendix. All other materials used in this analysis are available online at `https://github.com/asbates/stat696/tree/master/reports/prostate-cancer`.

3 Analysis

3.1 Exploratory Analysis

We begin by inspecting tables of the categorical variables against whether tumor penetration has occured. Rather than examing contingency tables of counts, we find it more informative to view tables marginalized by the covariates. This provides more context by allowing us to see, for example, the proportion of Black patients who had tumor penetration and the proportion who did not. Table 2 displays the proportion of patients who did and did not experience tumor penetration of the

prostatic capsule grouped by the results of a digital rectal exam. It shows that the majority of patients who do not have a nodule have not experience tumor penetration. For those patients who did have a nodule, the rate of tumor penetration is reversed. The most pronounced difference is for patients who had a bilobar nodule with 65% experiencing tumor penetration. The relationship we see here tells us that DRE results will likely be a valuable predictor.

Table 2: Results of digital rectal exam vs. whether tumor pentration of prostatic capsule occured, marginalized by the digital rectal exam results.

	DRE Result			
	no nodule	unilobar left	unilobar right	bilobar
No Penetration	80.8	63.4	47.4	34.6
Penetration	19.2	36.6	52.6	65.4

The relationship for the remaining categorical variables can be found in Tables 5 to 7 in the appendix. For detection of capsular involvment and Gleason score, we find a similar story as DRE results. For patients where capsular involvement was detected, tumor penetration occured at more than twice the rate than patients where involvement was not detected (35% vs. 75%). As Gleason score increases from zero to nine, the tumor penetration rate increases from zero to 92%. Where we do not see a significant change in the rate of tumor penetration is with race. Both Black and White patients have a tumor penetration rate of approximately 40%.

For the numeric variables, we examine summary statistics and plots of age and PSA grouped by whether prostatic capusle penetration has occured. Summary statistics and a box plot for age can be found in Table 8 and Figure 3 in the appendix. There is virtually no difference in the age distribution of patients who have experience tumor penetration and those who have not. As such, we do not expect it to make a significant contribution to our model. On the other hand, PSA value seems like it may be an important predictor. Figure 1 is a histogram of PSA value where color indicates tumor penetration of the prostatic capsule. The distribution has a similar shape for low PSA values however, for PSA values above 67 mg/ml there are no cases where tumor penetration has occured.

3.2 Modeling

To model the relationship between tumor penetration and the predictor variables, we use a logistic regression model. As most of the covariates are tests covering different aspects of of the prostate, it is not hard to imagine there some of them connected in their relationship with tumor penetration. For this reason, the initial model is fit using all first-order interaction terms. This serves as the starting point for model selection which is done via stepwise regression with AIC as the selection criteria. The model chosen via the stepwise procedure includes race, age, DRE results, PSA value, and Gleason score as well as age interacted with DRE result and race interacted with PSA value. A detailed summary can be found in Table 10. We note here that the AIC is 391 and the p-values for age and race are quite high at 0.11 and 0.99, respectively.

Since some of the p-values of the model obtained via stepwise regression are large, we remove these terms and refit our model. Specifically, we remove age and race along with interaction terms associated with them. We can not justify keeping variables in interactions that themselves are not included in the model. Note that for this model, removing the relevant interaction terms actually

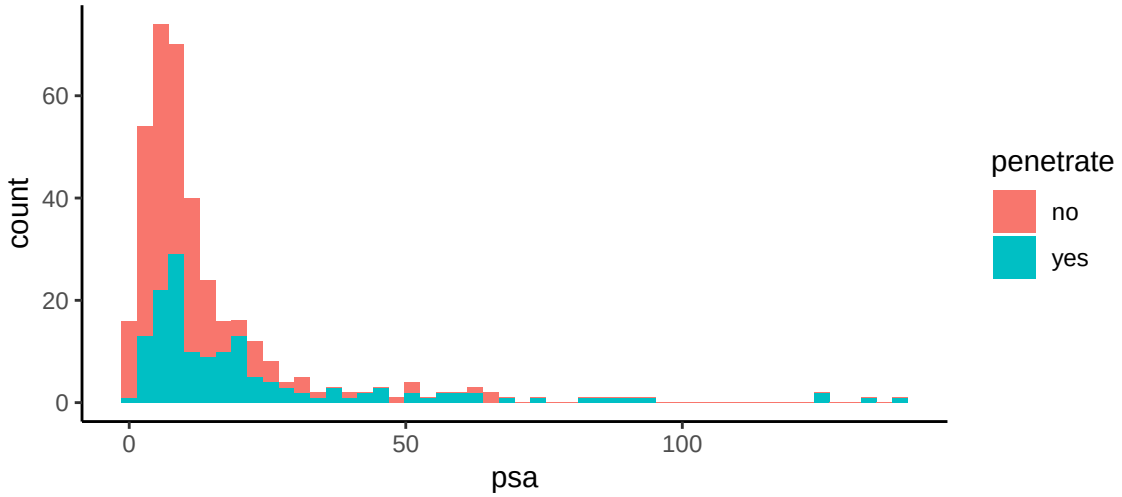

Figure 1: Histogram of prostate specific antigen value colored by whether tumor penetration of prostatic capsule occured.

means removing all interaction terms. Once this is done, we are left with a model that relates tumor penetration to just three variables: DRE results, PSA value, and Gleason score.

3.3 Diagnostics

To assess the model we use diagnostic tools on explanatory variable patterns[1] (EVP). We collapse the orginal data into groups where each group shares the same covariate values. Recall that one of the variables, PSA, is continous so we first cut the values into nonoverlapping bins. Then we fit the model to the EVPs using the number of original observations in each EVP as weights. Standardized Pearson residuals, Cook's distance, and leverage from the EVP model are what we use to perform our diagnostics. A plot of the residuals vs. fitted probabilites is shown in Figure 2. The dotted horizontal lines indicate the standard cut off values of -3, -2, 2, and 3. There are no discernible patterns in the residuals and they are scattered roughly equally about zero. For larger fitted probabilities there is a bit more variability in the residuals but nothing too concerning. One EVP (lower right hand corner of plot) is slightly smaller than the lowest cut off bound, having a value of -3.4. This is only a minor violation and the cut off values are simply rules-of-thumb so we consider this EVP satisfactory.

Further diagnostic plots can be found in Figure 4 in the appendix. Only one other EVP is of any concern as its Cook's distance and leverage values are both outside the standard cut off values. However, this EVP is also only faintly outside the boundaries[2]. Twenty two of the original observations are contained in this EVP. Given that the Cook's distance and leverage for this EVP are just barely outside their respective cut off, this is not enough justification to remove this many data points.

As a final diagnostic measure we perform a Homer-Lemeshow goodness-of-fit test using the model fit to the EVPs. The EVP data is separated into bins according ot quantiles of the fitted values. Then observed and expected counts are computed for each bin and a χ^2 test is performed. This test

[1] Also called covariate patterns.

[2] Cook's distance = 0.06, leverage = 0.3 with cutoff values of 0.05 and 0.28.

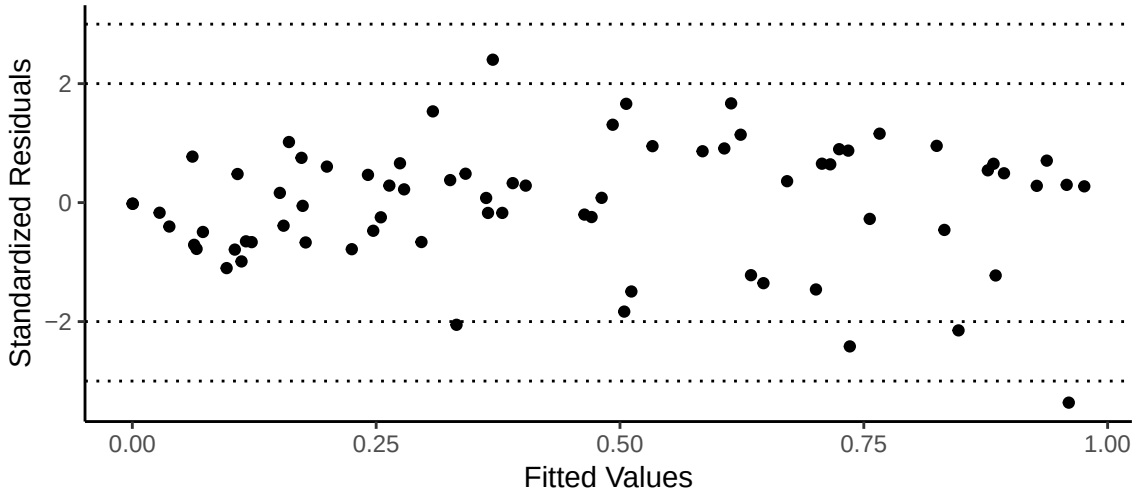

Figure 2: Pearson residuals vs. fitted values via Explanatory Variable Patterns (74). The horizontal reference lines indicate the usual residual cutoffs of +-2 and +- 3.

is an overall assessment of how well the model fits the data. The obtained test statistic is 3.9 with a corresponding p-value of 0.42. The p-value may seem unfruitful at first as we are typically looking for small values. However in the situation at hand this is not the case. The null hypothesis in this test is that the model fits the data well so we do *not* want to reject the null in our situation. So based on this test our model seems to fit the data well and we can move on to inferences.

3.4 Inferences

Table 3 contains a summary of the final model used in this analysis. Included in the table are estimates for the coefficients with their associated standard errors and p-values. Also include are the odds ratio for each term and 95% confidence intervals for the odds ratio. The largest odds ratio corresponds to a digital rectal exam where a unilobar nodule is found on the patient's ride side. The odds of experiencing tumor penetration of the prostatic capsule for such patients is 4.73 times more than the odds for patients which did not have a nodule present. Interestingly, the odds for patients with a unilobar nodule on the left are about half the odds of tumor pentration for patients with either a nodule on the right or a bilobar nodule. The odds ratio for Gleason score indicates that each unit increase in score corresponds to increase in odds of 2.71 times, if all other variables are held constant. Since the odds ratio for PSA is smaller than two, it has a slightly different interpretation. One additional mg/ml of PSA is associated with a 3% increase in odds when the remaining variables are unchanged.

In addition to making inferences on the odds ratios of each of the variables in our model, we can assess the models potential predictive ability. The primary goal of this analysis is to determine which factors have a particularly strong influence on tumor penetration. This is not a prediction problem per se and so we did not follow the usual paradigm of splitting our data into a training and testing set. This means that the predictions we consider here are in sample predictions and will no doubt have much better performance than what would actually be seen in practice. Nonetheless, we can still get some sense of how well the model performs at predicting tumor penetration. To that end, we plotted a receiver operating characteristic (ROC) curve which we leave in the appendix

Table 3: Summary for final model fitting tumor penetration of prostatic capsule with covariates digital rectal exam result, prostate specific antigen value, and total Gleason score. Coefficient estimates with standard errors and p-values are provided along with odds ratios and 95% confidence intervals for odd ratios. AIC for the model is 393.

Term	**Estimate**	**SE**	**p-value**	**Odds ratio**	**OR 95% CI**
Intercept	-8.14	1.06	0.000	0.00	(0.00 , 0.00)
DRE: unilobar left	0.77	0.36	0.030	2.17	(1.09 , 4.43)
DRE: unilobar right	1.55	0.37	0.000	4.73	(2.32 , 10.00)
DRE: bilobar nodule	1.43	0.45	0.001	4.18	(1.75 , 10.25)
Prostate specific antigen value	0.03	0.01	0.004	1.03	(1.01 , 1.05)
Gleason score	1.00	0.16	0.000	2.71	(2.00 , 3.76)

(Figure 5) for the curious reader. The curve indicates that our model performs significantly better than the default comparison of randomly assigning whether tumor penetration will occur. The area under the ROC curve (AUC) for our final model was 0.82 which is quite good. The ROC curve and AUC indicate that the model developed here has potential as an overall diagnostic tool that combines the results of a digital rectal exam, prostate specific antigen value, and Gleason score into a single measure of risk, rather than considering each measure individually.

We now illustrate how our model can be used as the aformentioned diagnostic tool. Let us suppose that we have three patients who have had a digitial rectal exam, have a PSA test done, and have received a Gleason score. Of course in reality we would have this information available but for the sake of demonstration we will use hypothetical values. The predictions are easy to obtain with software so we will focus on what to do once we have the predictions. Table 4 presents the supposed values as well as the probability of tumor penetration obtained from the model. Doctors would likely be satisfied with the first two patients whose predicted probability of tumor penetration are quite low. However, further investigation would likely be desired for the third patient how has a reasonably high chance of tumor penetration according to the model.

But is this probability really that high? What if it was 0.85 instead of 0.88? How would we determine what is considered 'high' and what is considered 'low'? One way we can do this is by considering the false positive rate and the false negative rate. We start by setting a cutoff value, say 0.85, for which values above we say that tumor penetration has occured and otherwise we say that it has not. Patients with scores above the cutoff would warrant a deeper investigation while patients below the cutoff would proceed with a regular prostate exam schedule. A false postive occurs when we say tumor pentration has occured when it actually has not. Similarly, a false negative is when we claim that tumor penetration has not occured when it actually has. Both false positives and false negatives are inevitable so the game here is to find an appropriate balance between the two and determine which one is more important. For example, we may want to minimize the false negative rate because this means a patient walks away thinking they have not experienced tumor penetration when in reality they have. Ultimately, a team of physicians would need to weigh the costs and benefits of an incorrect prediction from which an appropriate cutoff value can be determined.

Table 4: Hypothetical values for DRE result, PSA value, and Gleason score along with predicted probability of tumor pentration of prostatic capsule obtained from the model.

DRE result	PSA value	Gleason score	Predicted value
no nodule	15.26	6	0.15
unilobar right	8.70	4	0.09
no nodule	80.42	8	0.88

4 Conclusion

In this analysis we developed a logistic regression model to aid in the understanding of how various factors relate to tumor penetration of the prostatic capsule. Along with this we compared the strength of the relationship by examing the odds ratio for each factor. The patient's age and race were not included in the model from which we can infer that these have a weak relationship with tumor penetration, if any. The most significant factor was whether a nodule was detected in a digital rectal exam. Presence of a unilobar nodule on the right relates to a 4.7 times increase in odds than for patients with no nodule detected. An increase in Gleason score by one is associated with a 2.7 times increase in odds provided the remaining covariates are fixed. For a one mg/ml increase in prostate specific antigen value, we can expect a 3% increase in odds if the other variables remain constant. The predictive power of the model was assessed via AUC which was 0.82. This led us to believe the model has the potential to be used as a tool to help determine if a given patient has experienced tummor penetration. We also discussed how the model can be used in practice for this purpose.

We should note that the analyis performed here does have its limitations. The most obvious of these limitations is the population that the data represents. Although race and age were not included in the model, they do limit our inferences because they were quite limited in scope. Only Black and White patients were included in this data set and 75% of the patients were over 62 years of age. One needs to be careful not to extrapolate outside this population.

Another aspect of this analysis that should be handled with care is using the model developed to predict tumor penetration. The predictive capability assessment of this analysis was done on data that was used to fit the model. These predictions will no doubt be an overestimate of the ability of predictions on previously unseen data. If it is desired to use this model as a primary method to determine if a patient has experienced tumor penetration, a different approach should be followed than was done in this analysis such as splitting the data into a training and testing set to estimate its out of sample predictive performance. One would likely also want to determine a cutoff point as this model outputs probabilities and not yes-no decisions. This has its own inherent difficulties. Experts would need to determine an acceptable level of false postives and false negatives and an apropriate balance between the two.

A Supplementary Tables and Figures

Table 5: Race vs. whether tumor penetration of prostatic capsule has occured, marginalized by race.

	Race	
	white	black
No Penetration	59.8	61.1
Penetration	40.2	38.9

Table 6: Detection of capsular involvement vs. whether tumor pentration of prostatic capsule occured, marginalized by capsular involvment.

	Capsular Involvement	
	no	yes
No Penetration	64.1	25.0
Tumor Penetration	35.9	75.0

Table 7: Total Gleason score vs. whether tumor penetration of prostatic capsule occured, marginalized by Gleason score.

	Gleason Score						
	0	4	5	6	7	8	9
No Penetration	100.0	100.0	91.0	72.5	43.3	20.7	7.7
Tumor Penetration	0.0	0.0	9.0	27.5	56.7	79.3	92.3

Table 8: Summary statistics of patient age grouped by whether tumor penetration of prostatic capsule occured.

	min	**median**	**mean**	**max**
No Penetration	50.0	67.0	66.3	79.0
Tumor Penetration	47.0	66.0	65.7	79.0

Table 9: Summary statistics for prostate specific antigen value grouped by whether tumor penetration of prostatic capsule occured.

	min	**median**	**mean**	**max**
No Penetration	0.3	7.5	10.0	66.7
Tumor Penetration	1.4	12.9	23.1	139.7

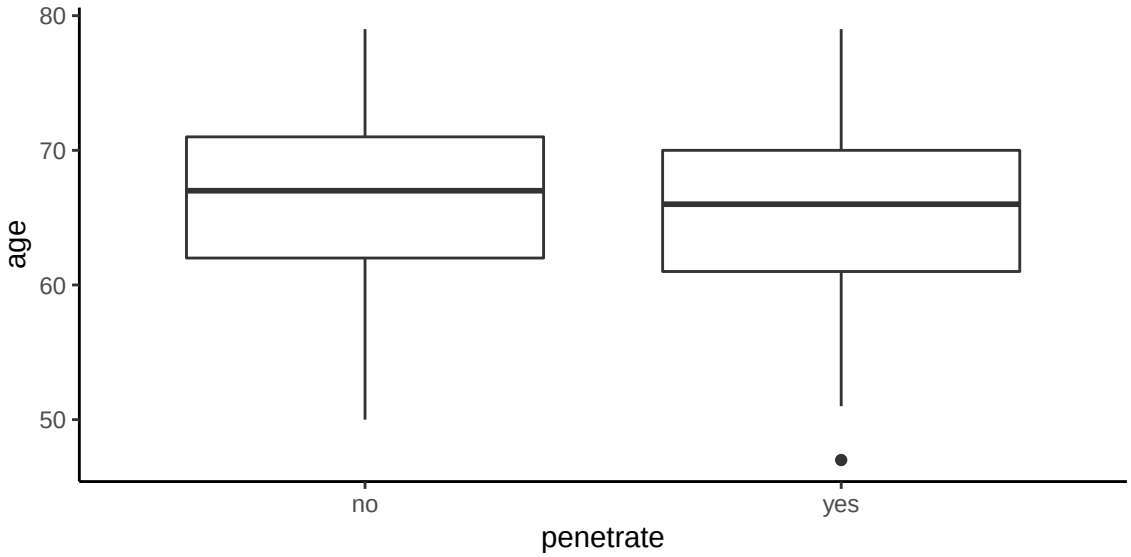

Figure 3: Boxplot of age vs. whether tumor penetration of prostatic capsule occured.

Table 10: Summary of best stepwise fit of tumor penetration of prostatic capsule against age, race, digital rectal exam result, prostate specific antigen value, total Gleason score and the interaction terms age * dre results and race * psa. AIC is 391.

Term	**Estimate**	**SE**	**p-value**
(Intercept)	-28.15	12.36	0.023
age	0.29	0.18	0.107
raceblack	-0.01	0.66	0.988
dreunilobar left	7.30	4.01	0.069
dreunilobar right	-0.29	4.20	0.944
drebilobar	2.67	5.29	0.614
psa	0.04	0.01	0.001
gleason	3.94	1.84	0.032
age:dreunilobar left	-0.10	0.06	0.095
age:dreunilobar right	0.03	0.06	0.646
age:drebilobar	-0.02	0.08	0.821
age:gleason	-0.04	0.03	0.110
raceblack:psa	-0.03	0.02	0.134

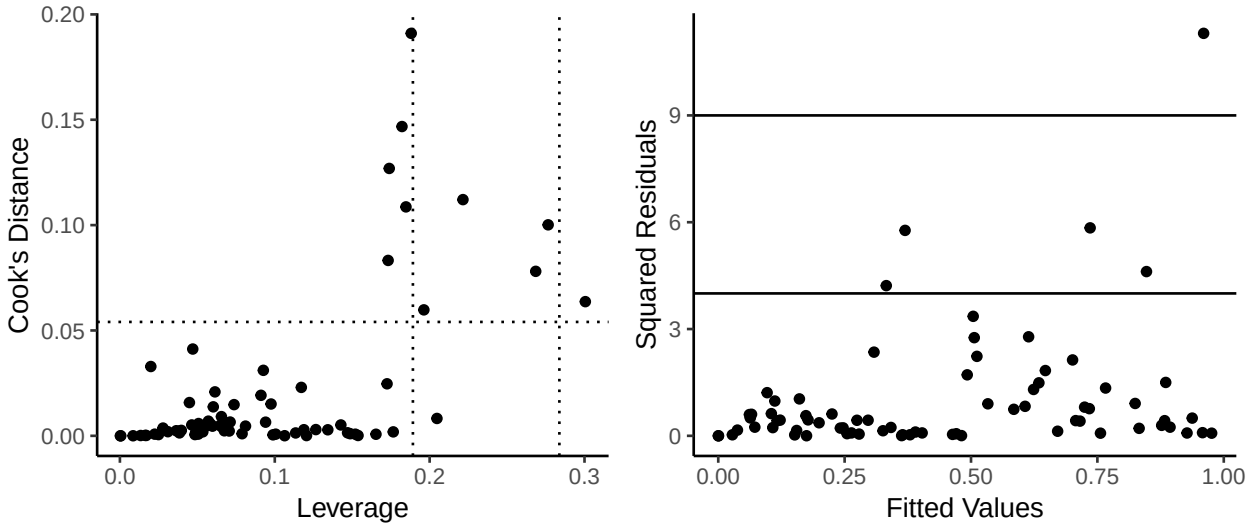

Figure 4: Diagnostic plots for the model fit on explanatory variable patterns. Left: Cook's distance vs. leverage. Right: squared Pearson standardized residuals vs. fitted probabilities

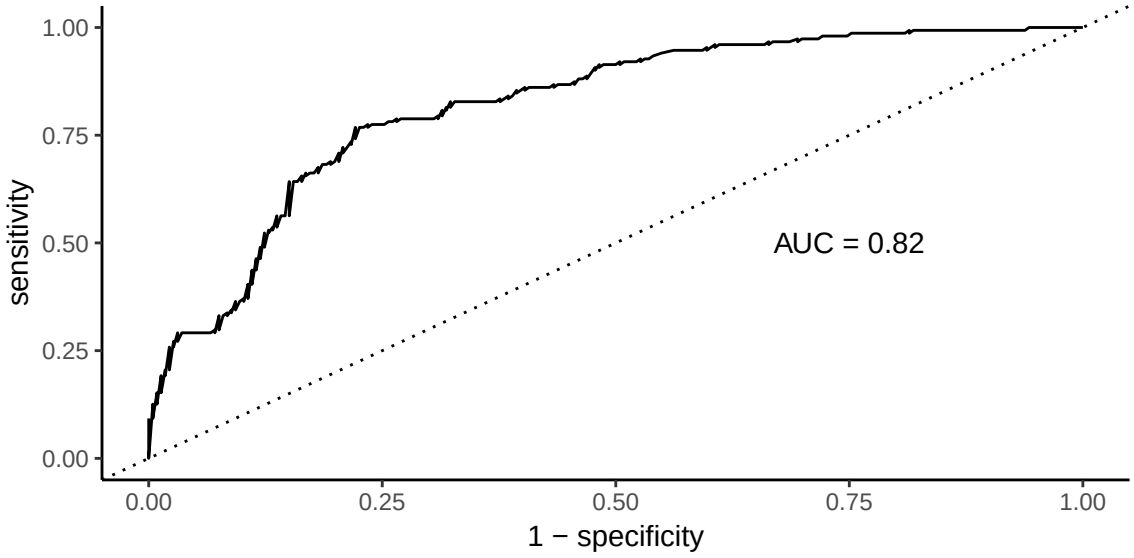

Figure 5: Reciever Operating Characteristic curve for final model fitting tumor penetration of prostatic capsule with covariates digital rectal exam result, prostate specific antigen value, and total Gleason score. The dotted line is a reference line indicating what the curve would be if we randomly decided whether tumor penetration occured.

B R Code

```
library(here)  # handles directories
library(readr)  # read in data
library(MASS) # stepwise regression
library(dplyr)  # manipulate data
library(forcats) # handle factors
library(ggplot2)  # plotting
library(broom) # tidy model output
library(pROC) # assess predictions

# these packages were used for formatting tables and figures
#  in the report but are not necessary to perform the analysis
library(knitr)  # table formatting
library(kableExtra) # table formatting
library(gridExtra) # matrix of ggplot's

prostate_raw <- read_tsv(here("data", "prostate.txt"))

# ====================================
# ======== SETUP/CLEANING =============
# ====================================

dim(prostate)
names(prostate)
str(prostate)

# remove id, rename
prostate <- prostate_raw %>%
  select(-id)

prostate <- prostate %>%
  rename(
    penetrate = capsule,
    dre = dpros,
    caps = dcaps
  )

# re-encode variables
prostate <- prostate %>%
  mutate(
    penetrate = as.factor(penetrate),
    race = as.factor(race),
    dre = as.factor(dre),
    caps = as.factor(caps)
  ) %>%
  mutate(
```

```
    penetrate = fct_recode(penetrate, yes = "1", no = "0"),
    race = fct_recode(race, white = "1", black = "2"),
    caps = fct_recode(caps, no = "1", yes = "2"),
    dre = fct_recode(dre,
                     "no nodule" = "1",
                     "unilobar left" = "2",
                     "unilobar right" = "3",
                     "bilobar" = "4")
  )

str(prostate)

# how many have experience capsule penetration?
prostate %>%
  group_by(penetrate) %>%
  summarise(n = n())

# any missing values?
apply(prostate, 2, function(x) sum(is.na(x)) )

# race has 3 missing values
summary(prostate)

prostate %>%
  filter(is.na(race))

# remove missing observations
prostate <- prostate %>%
  filter(!is.na(race))

dim(prostate)
prostate %>%
  group_by(penetrate) %>%
  summarise(n = n())

# ======================================
# ============== EDA ==================
# ======================================

# ----- categorical predictors -----
race_tab <- table(`Capsule Penetration` = prostate$penetrate,
                  Race = prostate$race)
dre_tab <- table(`Capsule Penetration` = prostate$penetrate,
                 `DRE Result` = prostate$dre)
caps_tab <- table(`Capsule Penetration` = prostate$penetrate,
                  `Capsular Involvement` = prostate$caps)
gleason_tab <- table(`Capsule Penetration` = prostate$penetrate,
```

```
                      `Gleason Score` = prostate$gleason)

race_tab
dre_tab
caps_tab
gleason_tab

# proportion tables marginalized by covariate
prop.table(race_tab, margin = 2) * 100
prop.table(dre_tab, margin = 2) * 100
prop.table(caps_tab, margin = 2) * 100
prop.table(gleason_tab, margin = 2) * 100

# ------------ numeric predictors --------------

# ---- age -----

# sumarize age by prostate
age_sum <- prostate %>%
  group_by(penetrate) %>%
  summarise(
    min = min(age),
    median = median(age),
    mean = mean(age),
    max = max(age)
  )

# bar chart of age
ggplot(prostate, aes(x = age)) +
  geom_bar()

# penetrate vs. age bar plot
ggplot(prostate, aes(x = age, fill = penetrate)) +
  geom_bar(position = "dodge") +
  labs(fill = "Capsule Penetration")

# boxplot - age vs. penetrate
ggplot(prostate, aes(x = penetrate, y = age)) +
  geom_boxplot()

# ----- psa ----
# summary stats for psa by prostate
prostate %>%
  group_by(penetrate) %>%
  summarise(
    min = min(psa),
    median = median(psa),
```

```
    mean = mean(psa),
    max = max(psa)
  )

# histogram of psa
ggplot(prostate, aes(x = psa)) +
  geom_histogram(bins = 50)

# histogram of psa vs. penetrate
ggplot(prostate, aes(x = psa, fill = penetrate)) +
  geom_histogram(bins = 50)

# boxplot - psa vs. penetrate
ggplot(prostate, aes(x = penetrate, y = psa)) +
  geom_boxplot()

# ====================================
# ========== MODELING ================
# ====================================

# fit a 'full' model with all first-order interactions
full_fit <- glm(penetrate ~.*.,
                data = prostate,
                family = binomial(link = "logit"))

# 'best' fit by stepwise regression
best_step <- stepAIC(full_fit, data = prostate)

best_step
summary(best_step)

# or via the broom package
tidy(best_step)

# if we remove variables from best_step according to p-value
# we are left with penetrate ~ dre + psa + gleason

# ====================================
# ========== DIAGNOSTICS =============
# ====================================

# functions used for residual analysis
one_fourth_root=function(x){
  x^(0.25)
}
source(here("reports", "prostate-cancer","code","examine.logistic.reg.R"))
```

```r
# note: we're using the 'raw' data here with the default encoding
pros_diag <- prostate_raw %>%
  rename(
    penetrate = capsule,
    dre = dpros,
    caps = dcaps
  )

# Create EVPs by binning continuous covariates
g <- 5 # number of categories
psa_interval = cut(pros_diag$psa,
                   quantile(pros_diag$psa, 0:g/g), include.lowest = TRUE)

w <- aggregate(penetrate ~ psa_interval + gleason + dre,
               data = pros_diag,
               FUN = sum)
n <- aggregate(penetrate ~ psa_interval + gleason + dre,
               data = pros_diag,
               FUN = length)
wn <- data.frame(
  w,
  trials = n$penetrate,
  prop = round(w$penetrate / n$penetrate, 2)
)

dim(wn)  # 74 EVPs

diag_fit <- glm(penetrate/trials ~ psa_interval + gleason + dre,
           data = wn,
           family = binomial(link = "logit"),
           weights = trials)

# with identify.points = TRUE we can click points
#  and see their location in the data frame
# In residuals vs. fitted, we see EVP 73 has a small residual (< 3)
# In Cook's distance vs. leverage, we find EVP 25 with large Cook's
#  distance and loarge leverage
# In delta X^2 (squared residuals) vs. fitted (size prop. to trials)
#   we find EVP 73 with high delta X^2 (>9)
# In delta X^2 vs. fitted (size prop. to Cook's distance)
#  we find EVP 73 with high delta x^2 (>9)
# there are a few other EVPs that are outside some cutoffs
#  but these two are the worst
examine <- examine.logistic.reg(diag_fit,
                                identify.points = FALSE,
                                scale.n = one_fourth_root,
                                scale.cookd = sqrt)
```

```r
wn_diag <- data.frame(
  wn,
  pi_hat = round(examine$pi.hat, 2),
  std_res = round(examine$stand.resid, 2),
  cooks_d = round(examine$cookd, 2),
  h = round(examine$h, 2)
)

p <- length(diag_fit$coefficients)

# locate points of interest
which_look_at <-
  abs(wn_diag$std_res) > 2 |
  wn_diag$cooks_d > 4 / nrow(wn) |
  wn_diag$h > 3*p / nrow(wn)
look_at <- wn_diag[which_look_at, ]

# look at points of interest
look_at

dim(look_at)  # 12 EVPs
sum(look_at$trials)  # 114 points

# look at only the points identified above
wn_diag[c(25, 73), ]

# what are the actual values for the cutoffs
#  of cooks d and leverage h?
4/nrow(wn)
2 * p / nrow(wn)
3 * p / nrow(wn)

# do any EVPs violate all criteria?
# no, they don't
which_violate_all <-
  abs(wn_diag$std_res) > 2 &
  wn_diag$cooks_d > 4 / nrow(wn) &
  wn_diag$h > 3*p / nrow(wn)
any(which_violate_all)

# EVP 25 contains 22 observations
#  its Cook's distance and leverage are above the cutoffs
#  however, only slightly on both accounts
#  not enough for us to justify removing this many points
# EVP 73 has 2 observations
#  it is above the residual cutoff thus above the delta X^2 cutoff
#  again, this is only a minor violation
```

```r
#  not quite strong enough for us to consider removing.

# as for the remaining EVPs
# the ones that do violate the cutoffs do so only slightly
# additionally, they mostly surpass one cutoff, not two
# lastly, almost all of them contain more than 6 original points, up to 14
# we will proceed with the model as is

# note: in the analysis we used the examine.logistic.reg function
# because of it's interactivity
# the plots in the report were made with the following code

# add relavent values (cooks d, etc.) to data
aug_diag <- augment(diag_fit,
                    data = wn,
                    type.predict = "response")

# for some reason, augment() is only returning deviance residuals
aug_diag <- aug_diag %>%
  mutate(
    .std.resid = rstandard(diag_fit, type = "pearson")
  )

# make diagnostic plots.
# same plots as examine.logistic.reg but using ggplot2

ggplot(aug_diag, aes(x = .fitted, y = .std.resid)) +
  geom_point() +
  geom_hline(yintercept = 2, linetype = "dotted") +
  geom_hline(yintercept = 3, linetype = "dotted") +
  geom_hline(yintercept = -2, linetype = "dotted") +
  geom_hline(yintercept = -3, linetype = "dotted")

cook_cutoff <- 4 / nrow(wn)
h_cutoff2 <- 2 * p / nrow(wn)
h_cutoff3 <- 3 * p / nrow(wn)

ggplot(aug_diag, aes(x = .hat, y = .cooksd)) +
  geom_point() +
  geom_vline(xintercept = h_cutoff2, linetype = "dotted") +
  geom_vline(xintercept = h_cutoff3, linetype = "dotted") +
  geom_hline(yintercept = cook_cutoff, linetype = "dotted") +
  ylab("Cook's Distance") +
  xlab("Leverage") +
  theme_classic()

ggplot(aug_diag, aes(x = .fitted, y = .std.resid^2)) +
```

```r
  geom_point() +
  geom_hline(yintercept = 4) +
  geom_hline(yintercept = 9) +
  ylab("Squared Residuals") +
  xlab("Fitted Values") +
  theme_classic()

# Hosmer-Lemeshow goodness-of-fit test

source(here("reports", "prostate-cancer", "code", "HLTest.R"))

# 6 groups used b/c that's minimum number such that expected counts are
#  greater than 5
hl_test <- HLTest(diag_fit, 6)
hl_test

# contingency table of observed vs. expected
cbind(hl_test$observed, hl_test$expect)

# ====================================
# ========== INFERENCE ===============
# ====================================

final_fit <- glm(penetrate ~ dre + psa + gleason,
                 data = prostate,
                 family = binomial(link = "logit"))

# summary and confidence intervals for final fit
final_fit
summary(final_fit)
confint(final_fit)

# also via the broom package
tidy(final_fit, conf.int = TRUE)


# ---- predictive power ----

fit_roc <- roc(prostate$penetrate, predict(final_fit, type = "response"))
plot(fit_roc, legacy.axes = TRUE, print.auc = TRUE)

# or using ggplot
data.frame(
  sensitivity = fit_roc$sensitivities,
  specificity = fit_roc$specificities
) %>%
  ggplot(aes(x = 1 - specificity, y = sensitivity)) +
```

```r
  geom_line() +
  geom_abline(slope = 1, intercept = 0) +
  annotate("text", label = paste0("AUC = ", round(fit_roc$auc, 2)),
           x = 0.75, y = 0.5)

# --- future predictions ---

# hypothetical new data
new_data <- data.frame(
  dre = c("no nodule", "unilobar right", "no nodule"),
  psa = c(15.26, 8.70, 80.42),
  gleason = c(6, 4, 8)
)

predict(final_fit, newdata = new_data, type = "response")

# or via broom
augment(final_fit, newdata = new_data, type.predict = "response")

# ================================
# ======= ADDITIONAL FUNCTIONS =====
# ================================

##############################################################################################
# NAME: Chris Bilder                                                                         #
# DATE: 7-30-13                                                                              #
# UPDATE:                                                                                    #
# Purpose: Automate part of the logistic regression diagnostic                               #
#     procedures into one function.                                                          #
#                                                                                            #
# NOTES:                                                                                     #
# 1) This function needs to be run just once before it is used. Below is an                  #
#   example of how to use the function the first time:                                       #
#                                                                                            #
#   source(file = "C:\\Chris\\examine.logistic.model.R")                                     #
#   mod.fit1 <- glm(formula = y/n ~ x, data = EVP.form, weight = n,                          #
#    family = binomial(link = logit))                                                        #
#   examine.logistic.reg(mod.fit1)                                                           #
#   mod.fit2 <- glm(formula = y/n ~ x + I(x^2), data = EVP.form, weight = n,                 #
#    family = binomial(link = logit))                                                        #
#   examine.logistic.reg(mod.fit2)                                                           #
#                                                                                            #
# 2) Arguments:                                                                              #
#   mod.fit.obj = model fit object from glm()                                                #
#   identify.points = identify points on the plot by mouse clicks                            #
#   bubble = produce plots where plotting point is proportional in size                      #
#    to a third dimension (number of trials or Cook's D)                                     #
```

```
#   scale.n, scale.cookd = scaling to use with bubble size; the default is     #
#    the original numerical scale of the quantity; it can be helpful to use    #
#    a transformation, like sqrt, when there is a large difference in          #
#    numerical values.                                                         #
#   pearson.dev = specifies whether "Pearson" or "deviance" based residuals    #
#    should be given in the plots                                              #
################################################################################

examine.logistic.reg <- function(mod.fit.obj = mod.fit,
                                 identify.points = TRUE, bubble = TRUE,
                                 scale.n = I,
                                 scale.cookd = I,
                                 pearson.dev = "Pearson"){

 pearson <- residuals(mod.fit.obj, type = "pearson")  #Pearson residuals
 # Standardized Pearson residuals
 stand.resid <- rstandard(model = mod.fit.obj, type = "pearson")
 deltaXsq <- stand.resid^2
 pred <- mod.fit.obj$fitted.values
 n <- mod.fit.obj$prior.weights   # Number of observations per EVP
 df <- mod.fit.obj$df.residual
 cookd <- cooks.distance(mod.fit.obj)
 h <- hatvalues(mod.fit.obj)
 dev.res <- residuals(mod.fit.obj, type = "deviance")
 # Standardized deviance residuals
 stand.dev.resid <- rstandard(model = mod.fit.obj, type = "deviance")
 deltaD <- dev.res^2 + h*stand.resid^2
 pear.stat <- sum(pearson^2)
 dev <- mod.fit.obj$deviance
 p <- length(mod.fit.obj$coefficients)

 # Type of residuals to include on plots
 resid.plot11 <- stand.resid
 resid.plot21 <- deltaXsq
 plot.label11 <- "Pearson"
 plot.label21 <- "Delta X^2"
 if (pearson.dev == "deviance") {
  resid.plot11 <- stand.dev.resid
  resid.plot21 <- deltaD
  plot.label11 <- "deviance"
  plot.label21 <- "Delta D"
 }

 ###############################################################################
 # Four plots

  # Open a new plotting window
```

```
#x11(width = 8, height = 6, pointsize = 12)
# Divide the plot into three rows and two columns.
# The last row is only 1cm in height to make sure
#  there is some room for the printed GOF statistics
layout(mat = matrix(c(1,2,3,4,5,5), byrow = TRUE, ncol = 2),
       height = c(1,1,lcm(1)))
# layout.show(5)


# Standardized residual vs predicted prob.
plot(x = pred, y = resid.plot11, xlab = "Estimated probabilities",
     ylab = "Standardized residuals",
 main = paste("Standardized", plot.label11, "residuals vs. est. prob."),
 ylim = c(min(-3, stand.resid), max(3, stand.resid)))
abline(h = c(-3, -2, 0, 2, 3), lty = "dotted", col = "blue")
if(identify.points == TRUE) {
 # labels(pred) uses the row names from the original data set
 #  This can be helpful, rather than the default of 1:n in identify(),
 #  when observations have been removed from the data set
 #  (i.e., the same row names will be used
 #  as with the original data set)
 identify(x = pred, y = resid.plot11, labels = labels(pred))
}

order.pred <- order(pred)
smooth.stand <- loess(formula = resid.plot11 ~ pred, weights = n)
lines(x = pred[order.pred], y = predict(smooth.stand)[order.pred],
      lty = "solid", col = "red")
# The ordering of pred leads to one line drawn across the plot.
# Otherwise, multiple lines will be drawn
#  between each pred and predict() pair, which may cause a
#  zig-zag-like pattern of lines

# Very similar way to get the loess model plotted
# smooth.stand <- loess(formula = resid.plot11 ~ pred, weights = n)
# x.axis <- seq(from = min(pred), to = max(pred),
#    by = (max(pred) - min(pred))/100)
# pred.data<-predict(object = smooth.stand, newdata =
# data.frame(pred = x.axis))
# lines(x = x.axis, y = pred.data, lty = "solid", col = "red")

# Cook's distance vs. leverage
plot(x = h, y = cookd, ylim = c(0, max(4/length(cookd), cookd)),
     xlim = c(0, max(3*p/length(h), h)),
 xlab = "Leverage (hat matrix diagonal)", ylab = "Cook's distance",
 main = "Cook's distance vs. leverage")
abline(h = c(4/length(cookd), 1), lty = "dotted")
```

```
abline(v = c(2*p/length(h), 3*p/length(h)), lty = "dotted")
if(identify.points == TRUE) {
 identify(x = h, y = cookd, labels = labels(h))
}

# Delta X^2 or D vs. predicted prob. with plotting point proportional to n
if(bubble == TRUE) {
 symbols(x = pred, y = resid.plot21, xlab = "Estimated probabilities",
         circles = scale.n(n),
  ylab = plot.label21,
  main = paste(
    plot.label21,
    "vs. est. prob. \n with plot point proportional to number of trials"),
  inches = 0.1, ylim = c(0, max(9, resid.plot21)))
 abline(h = c(4, 9), lty = "dotted", col = "blue")
 if(identify.points == TRUE) {
  identify(x = pred, y = resid.plot21, labels = labels(pred))
 } }
else {
 plot(x = pred, y = resid.plot21, xlab = "Estimated probabilities",
      ylab = plot.label21,
  main = paste(plot.label21, "vs. est. prob."),
  ylim = c(0, max(9, resid.plot21)))
 abline(h = c(4, 9), lty = "dotted", col = "blue")
 if(identify.points == TRUE) {
  identify(x = pred, y = resid.plot21)
} }


# Print deviance/df on plot
dev.df <- dev/df
gof.threshold <- round(c(1 + 2*sqrt(2/df), 1 + 3*sqrt(2/df)), 2)
mtext(text = paste("Deviance/df = ", round(dev.df, 2),
                   "; GOF thresholds: 2 SD = ",
 round(gof.threshold[1], 2), ", 3 SD = ",
 round(gof.threshold[2], 2), sep = ""),
 side = 1, line = 6, cex = 1.0, adj = 0)


# Delta X^2 or D vs. predicted prob.
# with plotting point proportional to Cook's distance
if(bubble == TRUE) {
 symbols(x = pred, y = resid.plot21, circles = scale.cookd(cookd),
         xlab = "Estimated probabilities",
  ylab = plot.label21,
  main = paste(
    plot.label21,
```

```
      "vs. est. prob. \n with plot point proportional to Cook's distance"),
    inches = 0.1, ylim = c(0, max(9, resid.plot21)))
   abline(h = c(4, 9), lty = "dotted", col = "blue")
   if(identify.points == TRUE) {
    identify(x = pred, y = resid.plot21, labels = labels(pred))
   } }
  else {
   # Empty plot because the last plot would be exactly the same
   plot(x = c(0, 1), y = c(0,1), type = "n", axes = FALSE, xlab = " ",
        ylab = " ")
  }

  # Return to normal layout
  layout(mat = 1)

  # Information is stored in the object, but not printed unless requested
  invisible(list(pearson = pearson, stand.resid = stand.resid,
                 stand.dev.resid = stand.dev.resid,
   deltaXsq = deltaXsq, deltaD = deltaD, cookd = cookd,
   pear.stat = pear.stat, dev = dev, dev.df = dev.df,
   gof.threshold = gof.threshold, pi.hat = pred, h = h))
}

HLTest = function(obj, g) {

 # first, check to see if we fed in the right kind of object

 stopifnot(family(obj)$family == "binomial" && family(obj)$link == "logit")

 y = obj$model[[1]]

 trials = rep(1, times = nrow(obj$model))

 if(any(colnames(obj$model) == "(weights)"))

  trials <- obj$model[[ncol(obj$model)]]

 # the double bracket (above) gets the index of items within an object

 if (is.factor(y))

  y = as.numeric(y) == 2  # Converts 1-2 factor levels to logical 0/1 values

 yhat = obj$fitted.values
 # Creates factor with levels 1,2,...,g
 interval = cut(yhat, quantile(yhat, 0:g/g), include.lowest = TRUE)
```

```
 Y1 <- trials*y

 Y0 <- trials - Y1

 Y1hat <- trials*yhat

 Y0hat <- trials - Y1hat

 obs = xtabs(formula = cbind(Y0, Y1) ~ interval)

 expect = xtabs(formula = cbind(Y0hat, Y1hat) ~ interval)

 if (any(expect < 5))

  warning("Some expected counts are less than 5. Use smaller number of groups")

 pear <- (obs - expect)/sqrt(expect)

 chisq = sum(pear^2)

 P = 1 - pchisq(chisq, g - 2)

 # by returning an object of class "htest", the function will perform like the

 # built-in hypothesis tests

 return(structure(list(

  method = c(paste("Hosmer and Lemeshow goodness-of-fit test with", g, "bins",
                   sep = " ")),

  data.name = deparse(substitute(obj)),

  statistic = c(X2 = chisq),

  parameter = c(df = g-2),

  p.value = P,

  pear.resid = pear,

  expect = expect,

  observed = obs

 ), class = 'htest'))
}
```

References

Auguie, Baptiste. 2017. *GridExtra: Miscellaneous Functions for "Grid" Graphics*. https://CRAN.R-project.org/package=gridExtra.

Mayo Clinic. 2018. "Prostate Cancer." https://www.mayoclinic.org/diseases-conditions/prostate-cancer/symptoms-causes/syc-20353087.

Müller, Kirill. 2017. *Here: A Simpler Way to Find Your Files*. https://CRAN.R-project.org/package=here.

———. 2018. *Bindrcpp: An 'Rcpp' Interface to Active Bindings*. https://CRAN.R-project.org/package=bindrcpp.

R Core Team. 2018. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Ripley, Brian. 2018. *MASS: Support Functions and Datasets for Venables and Ripley's Mass*. https://CRAN.R-project.org/package=MASS.

Robin, Xavier, Natacha Turck, Alexandre Hainard, Natalia Tiberti, Frédérique Lisacek, Jean-Charles Sanchez, and Markus Müller. 2018. *PROC: Display and Analyze Roc Curves*. https://CRAN.R-project.org/package=pROC.

Robinson, David, and Alex Hayes. 2018. *Broom: Convert Statistical Analysis Objects into Tidy Tibbles*. https://CRAN.R-project.org/package=broom.

Wickham, Hadley. 2018. *Forcats: Tools for Working with Categorical Variables (Factors)*. https://CRAN.R-project.org/package=forcats.

Wickham, Hadley, Winston Chang, Lionel Henry, Thomas Lin Pedersen, Kohske Takahashi, Claus Wilke, and Kara Woo. 2018. *Ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics*. https://CRAN.R-project.org/package=ggplot2.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2018. *Dplyr: A Grammar of Data Manipulation*. https://CRAN.R-project.org/package=dplyr.

Wickham, Hadley, Jim Hester, and Romain Francois. 2017. *Readr: Read Rectangular Text Data*. https://CRAN.R-project.org/package=readr.

Xie, Yihui. 2018a. *Bookdown: Authoring Books and Technical Documents with R Markdown*. https://CRAN.R-project.org/package=bookdown.

———. 2018b. *Knitr: A General-Purpose Package for Dynamic Report Generation in R*. https://CRAN.R-project.org/package=knitr.

Zhu, Hao. 2018. *KableExtra: Construct Complex Table with 'Kable' and Pipe Syntax*. https://CRAN.R-project.org/package=kableExtra.