Revisit to logistic regression

Example: Prostate Cancer

PROSTATE CANCER DATA SET
SIZE: 380 observations, 9 variables
SOURCE: Hosmer and Lemeshow (2000) Applied Logistic egression: 2nd Edn.

1 Identification Code	1 – 380	ID
2 Tumor Penetration of Prostatic Capsule	0 = No Penetration, 1 = Penetration	CAPSULE
3 Age	Years	AGE
4 Race	1= White, 2 = Black	RACE
5 Results of Digital Rectal Exam	1 = No Nodule 2 = Unilobar Nodule (Left) 3 = Unilobar Nodule (Right) 4 = Bilobar Nodule	DPROS
6 Detection of Capsular Involvement in Rectal Exam	1 = No, 2 = Yes	DCAPS
7 Prostatic Specific Antigen Value	mg/ml	PSA
8 Tumor Volume from Ultrasound	cm3	VOL
9 Total Gleason Score	0 - 10	GLEASON

What factors are related to capsular penetration?

- The **prostate capsule** is the membrane the surrounds the prostate gland
- As prostate cancer advances, the disease may extend into the capsule (extraprostatic extension) or beyond (extracapsular extension) and into the seminal vesicles.
- Capsular penetration is a poor prognostic indicator, which accounts for a reduced survival expectancy and a higher progression rate following radical prostatectomy.
- Let's start with PSA and Gleason score
- Both are well-known factors related to disease severity
- What does a linear regression of capsular penetration on PSA and Gleason mean?

$$Y_i = \beta_0 + \beta_1 PSA + \beta_2 GS + e_i$$

PSA

- **PSA** is the abbreviation for prostate-specific antigen which is an enzyme produced in the epithelial cells of both benign and malignant tissue of the prostate gland.
- The enzyme keeps ejaculatory fluid from congealing after it has been expelled from the body.
- Prostate-specific antigen is used as a tumor marker to determine the presence of prostate cancer because a greater prostatic volume, associated with prostate cancer, produces larger amount of prostate-specific antigen.

http://www.prostate-cancer.com/

Gleason Score

- The prostate cancer **Gleason Score** is the sum of the two Gleason grades.
- After a prostate biopsy, a pathologist examines the samples of prostate cancer cells to see how the patterns, sizes, and shapes are different from healthy prostate cells.
- Cancerous cells that appear similar from healthy prostate are called well-differentiated while cancerous cells that appear very different from healthy prostate cells are called poorly-differentiated.
- The pathologist assigns one Gleason grade to the most common pattern of prostate cancer cells and then assigns a second Gleason grade to the second-most common pattern of prostate cancer cells.
- These two Gleason grades indicate prostate cancer's aggresiveness, which indicates how quickly prostate cancer may extend out of the prostate gland.
- Gleason score = Gleason 1 + Gleason 2

http://www.prostate-cancer.com/

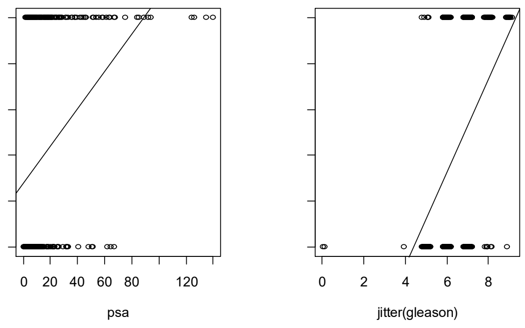
What is Y?

- Y is a binary outcome variable
- Observed data:
  - Y_i = 1 if patient if patient had capsular involvement
  - Y_i = 0 if patient did not have capsular involvement
- But think about the 'binomial distribution'
- The parameter we are modeling is a probability, p
- We'd like to be able to find a model that relates the probability of capsular involvement to covariates

$$P(Y_i = 1) = \beta_0 + \beta_{\cdot} PSA + \beta_2 GS + e_i$$

For a one-unit increase in GS, we expect the probability of capsular penetration to increase by β_2.

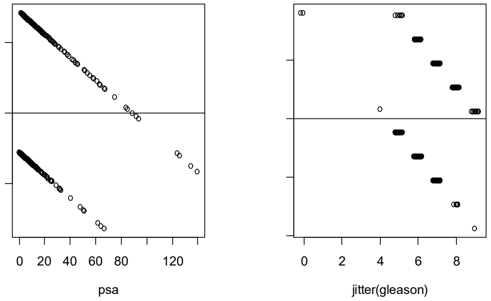
Data exploration?



What are the problems?

- The interpretation does not make sense for a few reasons
- You cannot have P(Y=1) values below 0 or 1
- What about the behavior of residuals?
  - normal?
  - constant variance?

(Based on simple linear regressions)



Why do they have these strange patterns?

Properties of the residuals (with linear regression)

- Nonnormal error terms
  - Each error term can only take one of two values:

$$e_i = 1 - \beta_0 - \beta_1 x_i \quad if\ y_i = 1$$
$$e_i = -\beta_0 - \beta_1 x_i \quad if\ y_i = 0$$

- Nonconstant error variance: the variance depends on X:

$$Var(\hat{p}) = p(1-p)$$
$$\sigma^2 = p(1-p)$$
$$\sigma^2 = (\beta_0 + \beta_1 x_i)(1 - \beta_0 - \beta_1 x_i)$$

Clearly, that does not work!

- A few things to consider
- We’d like to model the ‘probability’ of the event occuring
- Y=1 or 0, but we can conceptualize values in between as probabilities
- We cannot allow probabilities greater than 1 or less than 0

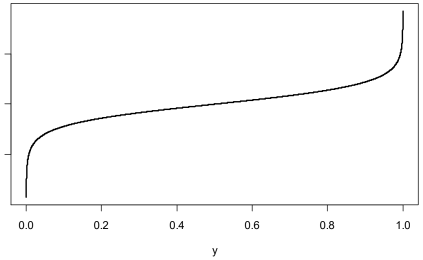
“Link” functions: Y

- Logit link: $\mathrm{logit}(Y) = \log\left(\frac{Y}{1-Y}\right)$
- Probit link: $probit(Y) = \Phi^{-1}(Y)$
- Complementary log-log:

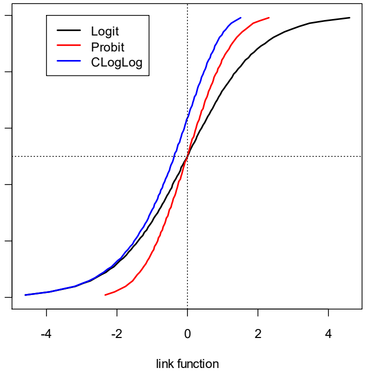
$$c \log\log(Y) = \log[-\log(1-Y)]$$

All have similar property

- They can take any value on the real line for 0 ≤ Y≤ 1
- Consider logit:
  - If Y=0, logit(Y) = log(0) = -Inf
  - If Y=1, logit(Y) = log(Inf) = Inf



All three together



Focus on Logistic Regression

- Logistic regression: uses the logit link
- "Simple" logistic regression model

$$\operatorname{logit}(P(Y=1)) = \log\left(\frac{P(Y=1)}{1-P(Y=1)}\right) = \beta_0 + \beta_1 X$$

- Residuals? They are not normal and we don't expect them to behave that way
- "Y_i are independent Bernoulli random variables with expected values $E(Y_i)$ = p_i"

- All methods use the MLE for estimating parameters.

E(Y_i)

- What is $E(Y_i)$?
  - Let $p_i = P(Y=1)$
  - Then $E(Y_i) = 1*p_i + 0*(1-p_i) = p_i$
  - Hence $E(Y_i) = P(Y=1) = p_i$
- **That will be our notation**

$$\text{logit}(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 X$$

- Now, solve for pi:

p_i

$$\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 X$$

$$\frac{p_i}{1-p_i} = \exp(\beta_0 + \beta_1 X)$$

$$p_i = (1-p_i)\exp(\beta_0 + \beta_1 X)$$

$$p_i = \exp(\beta_0 + \beta_1 X) - p_i \exp(\beta_0 + \beta_1 X)$$

$$p_i + p_i \exp(\beta_0 + \beta_1 X) = \exp(\beta_0 + \beta_1 X)$$

$$p_i(1 + \exp(\beta_0 + \beta_1 X)) = \exp(\beta_0 + \beta_1 X)$$

$$\mathbf{p_i} = \frac{\mathbf{exp(}\ \text{ഞ}_0 + \text{ഞ}_1\mathbf{X)}}{\mathbf{1} + \mathbf{exp(}\ \text{ഞ}_0 + \text{ഞ}_1\mathbf{X)}}$$

Hence, the following are equivalent:

$$p_i = \frac{\exp(\beta_0 + \beta_1 X_i)}{1 + \exp(\beta_0 + \beta_1 X_i)}$$

$$\text{logit}(p_i) = \beta_0 + \beta_1 X_i$$

Fitted values: two types

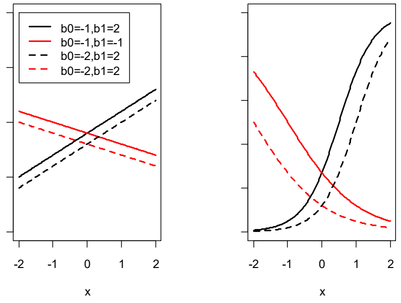
- Linear predictor:

$$\text{logit}(p_i) = \hat{\beta}_0 + \hat{\beta}_1 X_i$$

- Fitted probability:

$$\hat{p}_i = \frac{\exp(\hat{\beta}_0 + \hat{\beta}_1 X_i)}{1 + \exp(\hat{\beta}_0 + \hat{\beta}_1 X_i)}$$

Fitted values



Prostate Cancer Example

- Logistic regression of capsular penetration on PSA and Gleason Score

$$\log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_{.} PSA + \beta_2 GS$$

- Notice that we don’t include the error term
- Implied assumption that the data (i.e. Y) is binary (Bernoulli)

R code

- Regression estimation:

```
glm(y~x1+x2+x3, family=binomial)
glm(y~x1+x2+x3, family=binomial(link=“logit”))
```

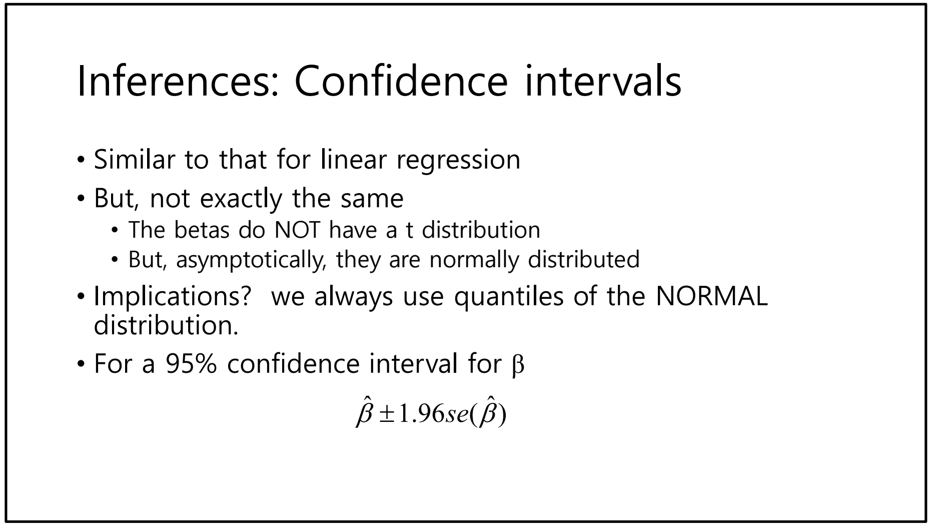
by default, link for binomial family is logit

glm = generalized linear regression

```
> pros1.reg <- glm(cap.inv ~ psa + gleason, family=binomial)
> summary(pros1.reg)

Call:
glm(formula = cap.inv ~ psa + gleason, family = binomial)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.2100  -0.7692  -0.4723   1.0431   2.1398

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.639296   1.011128  -7.555 4.18e-14 ***
psa          0.026677   0.008929   2.988  0.00281 **
gleason      1.059344   0.158327   6.691 2.22e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 512.29  on 379  degrees of freedom
Residual deviance: 404.44  on 377  degrees of freedom
AIC: 410.44

Number of Fisher Scoring iterations: 5
```

Interpreting the output

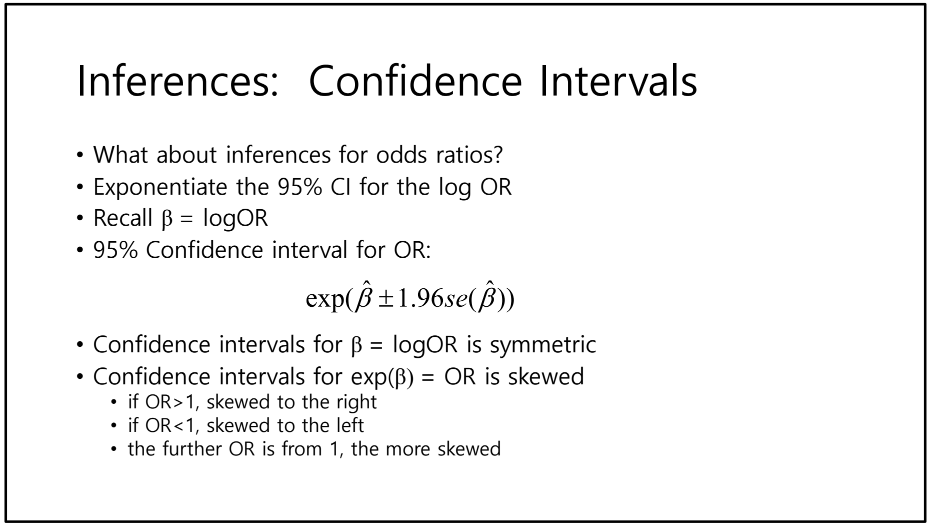
- Beta coefficients
- What do they mean?
  - log-odds ratios
  - example: comparing two men with Gleason scores that are one unit different, the log odds ratio for capsular penetration is 1.06.
- We usually exponentiate them:
  - $\exp(B_2) = \exp(1.06) = 2.88$
  - the odds of capsular penetration for a man with Gleason score of 7 is 2.88 times that of a man with Gleason score of 6
  - The odds ratio for a 1 unit difference in Gleason score is 2.88
- You also need to interpret them as ‘adjusting for PSA’



Inferences: Confidence intervals

- Similar to that for linear regression
- But, not exactly the same
  - The betas do NOT have a t distribution
  - But, asymptotically, they are normally distributed
- Implications? we always use quantiles of the NORMAL distribution.
- For a 95% confidence interval for β

$$\hat{\beta} \pm 1.96 se(\hat{\beta})$$

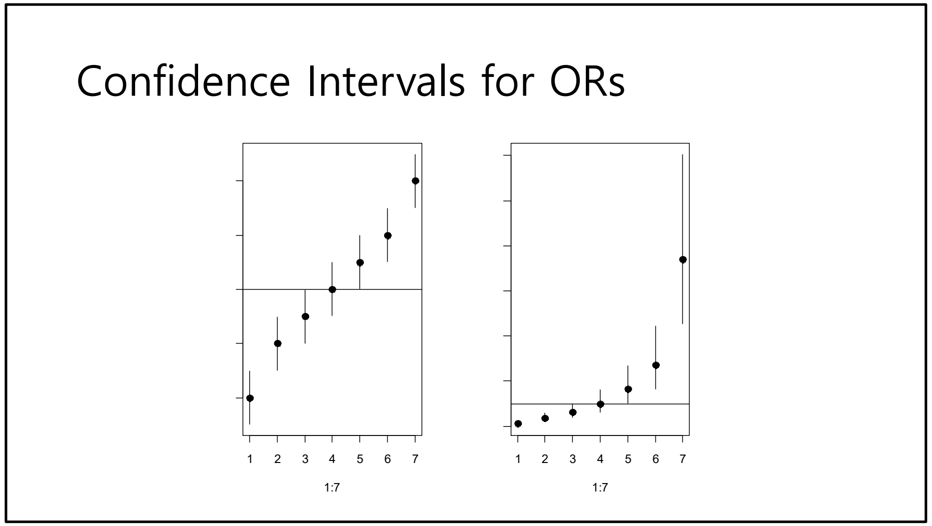


Inferences: Confidence Intervals

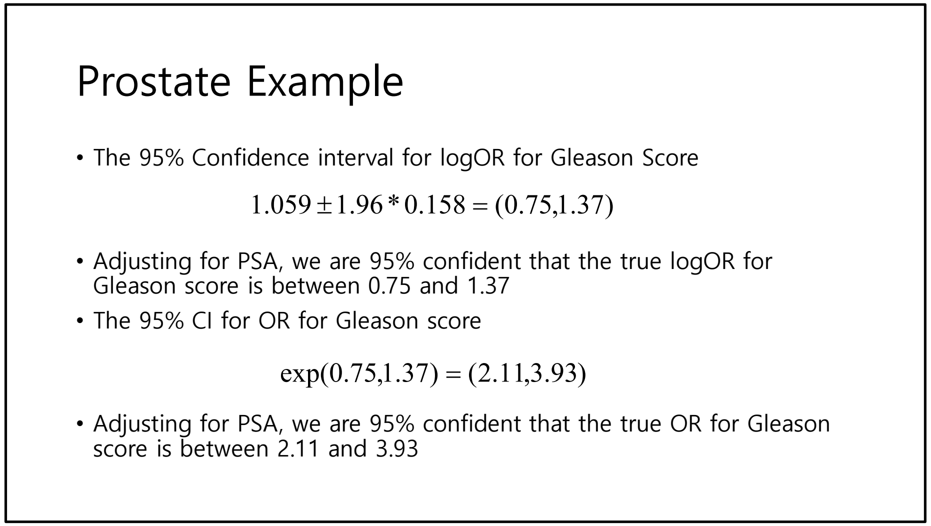
- What about inferences for odds ratios?
- Exponentiate the 95% CI for the log OR
- Recall β = logOR
- 95% Confidence interval for OR:

$$\exp(\hat{\beta} \pm 1.96 se(\hat{\beta}))$$

- Confidence intervals for β = logOR is symmetric
- Confidence intervals for exp(β) = OR is skewed
  - if OR>1, skewed to the right
  - if OR<1, skewed to the left
  - the further OR is from 1, the more skewed



Confidence Intervals for ORs



Prostate Example

- The 95% Confidence interval for logOR for Gleason Score

$$1.059 \pm 1.96 * 0.158 = (0.75, 1.37)$$

- Adjusting for PSA, we are 95% confident that the true logOR for Gleason score is between 0.75 and 1.37
- The 95% CI for OR for Gleason score

$$\exp(0.75, 1.37) = (2.11, 3.93)$$

- Adjusting for PSA, we are 95% confident that the true OR for Gleason score is between 2.11 and 3.93

Inferences: Hypothesis Testing

- Similar to linear regression
- But, we use a Z and not a t for testing signficance

$$\frac{\hat{\beta}}{se(\hat{\beta})} \sim N(0,1) \text{ under Ho} : \beta = 0$$

- Hence, we use -1.96 and 1.96 as thresholds for alpha of 0.05
- Need to worry more about whether or not asymptotics are appropriate (i.e., is sample size large enough?)

Prostate Example

```
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.639296   1.011128  -7.555 4.18e-14 ***
psa          0.026677   0.008929   2.988  0.00281 **
gleason      1.059344   0.158327   6.691 2.22e-11 ***
```

- PSA: p = 0.003
- Gleason: p<0.0001

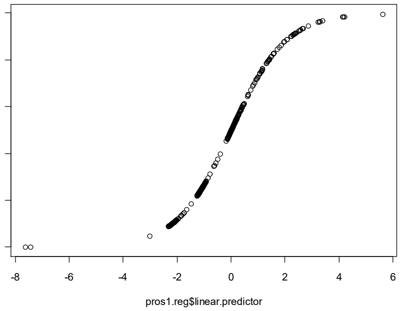
- Both PSA and Gleason are strongly associated with capsular penetration

Fitted estimates

- As mentioned earlier, two types
  - linear predictor
  - fitted probability
- For most inference, the fitted probability will be of more interest

```
> attributes(pros1.reg)
$names
 [1] "coefficients"      "residuals"         "fitted.values"
 [4] "effects"           "R"                 "rank"
 [7] "qr"                "family"            "linear.predictors"
[10] "deviance"          "aic"               "null.deviance"
[13] "iter"              "weights"           "prior.weights"
[16] "df.residual"       "df.null"           "y"
[19] "converged"         "boundary"          "model"
[22] "call"              "formula"           "terms"
[25] "data"              "offset"            "control"
[28] "method"            "contrasts"         "xlevels"
```

Fitted values vs. linear predictor



Estimation

- Recall estimation for linear regression
  - least squares
  - maximum likelihood
- For GLMs, maximum likelihood is used
- There is not a “closed form” solution
- As a result, an iterative (or algorithmic) approach is used
  - Newton-Raphson algorithm
  - Expectation-Maximization (EM) algorithm
- Notice in R output “scoring iterations” is listed

Maximum Likelihood Estimation

- Based on the likelihood function
- Recall the process
  - Write down the likelihood
  - take partial derivatives with respect to the parameters (i.e., β’s)
  - set each partial derivative equal to zero
  - Solve the system of equations for the estimated values of β’s
- The estimation of standard errors is more complicated (recall information matrix?)

Maximum Likelihood Estimation

- With logistic regression (and other generalized linear regression models), you cannot “solve” for the β’s.
- You must then use Newton-Raphson (or other) approach to do the solving.

Likelihood Function for “simple” logistic regression

$$L(p;y)=\prod_{i=1}^{n} p^{y_i}(1-p)^{1-y_i}$$

$$L(\beta_0,\beta_1;y,x)=\prod_{i=1}^{n}\left(\frac{\exp(\beta_0+\beta_1x_i)}{1+\exp(\beta_0+\beta_1x_i)}\right)^{y_i}\left(\frac{1}{\exp(\beta_0+\beta_1x_i)}\right)^{1-y_i}$$

$$=\prod_{i=1}^{n}\frac{\left(\exp(\beta_0+\beta_1x_i)\right)^{y_i}}{1+\exp(\beta_0+\beta_1x_i)}$$

$$\log L(\beta_0,\beta_1;y,x)=\sum_{i=1}^{n} y_i(\beta_0+\beta_1x_i)-\log(1+\exp(\beta_0+\beta_1x_i))$$

Score functions

$$\log L(\beta_0, \beta_1; y, x) = \sum_{i=1}^{n} y_i(\beta_0 + \beta_1 x_i) - \log(1 + \exp(\beta_0 + \beta_1 x_i))$$

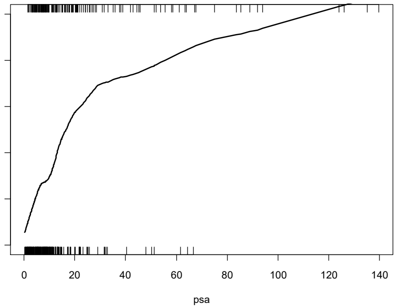
$$\frac{\partial \log L}{\partial \beta_0} = \sum_{i=1}^{n} y_i - \frac{\exp(\beta_0 + \beta_1 x_i)}{1 + \exp(\beta_0 + \beta_1 x_i)}$$

$$\frac{\partial \log L}{\partial \beta_1} = \sum_{i=1}^{n} x_i y_i - \frac{x_i \exp(\beta_0 + \beta_1 x_i)}{1 + \exp(\beta_0 + \beta_1 x_i)}$$

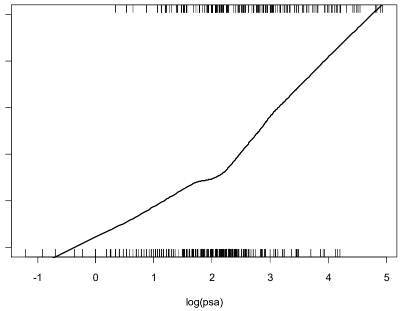
Second derivatives can be obtained to find standard errors and covariances of coefficients.

Data exploration and modeling

- Scatterplots are not helpful on their own
- Lowess smooths may be:



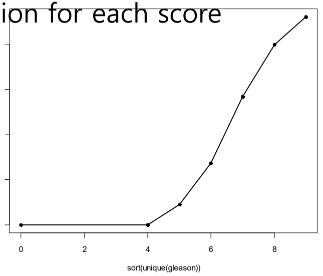
LogPSA



But should it look linear?

Gleason Score

- Smoother?
- Gleason score is categorical
- We can estimate the proportion of capsular penetration for each score



Rcode

```
###########################
smoother1 <- lowess(psa, cap.inv)
plot(psa, cap.inv, type="n")
lines(smoother1, lwd=2)
rug(psa[cap.inv==0], side=1)
rug(psa[cap.inv==1], side=3)

smoother2 <- lowess(log(psa), cap.inv)
plot(log(psa), cap.inv, type="n")
lines(smoother2, lwd=2)
rug(log(psa[cap.inv==0]), side=1)
rug(log(psa[cap.inv==1]), side=3)

###########################
gleason.probs <- table(gleason, cap.inv)/as.vector(table(gleason))
gleason.p <- gleason.probs[,2]
par(mar=c(5,4,1,1))
plot(sort(unique(gleason)), gleason.p, pch=16)
lines(sort(unique(gleason)), gleason.p, lwd=2)
```

Modeling, but also model checking

- These will be useful to compare “raw data” to fitted model
- Smoothers etc can be compared to fitted model
- If the model fits well, you would expect to see good agreement
- Problem?
  - only really works for simple logistic regression
  - cannot generalize to multiple logistic

Revised model

- Try logPSA
- Try categories of Gleason: what makes sense?

```
pros2.reg <- glm(cap.inv ~ log(psa) + factor(gleason), family=binomial)
summary(pros2.reg)

keep <- ifelse(gleason>4,1,0)
data.keep <- data.frame(cap.inv, psa, gleason)[keep==1,]

pros3.reg <- glm(cap.inv ~ log(psa) + factor(gleason), data=data.keep,
        family=binomial)
summary(pros3.reg)

pros4.reg <- glm(cap.inv ~ log(psa) + gleason, data=data.keep,
        family=binomial)
summary(pros4.reg)

pros5.reg <- glm(cap.inv ~ log(psa) + gleason,  family=binomial)
summary(pros5.reg)

##########
median(log(psa))
b <- pros5.reg$coefficients
fit.logpsamed <- b[1] + b[2]*median(log(psa)) + b[3]*c(0:9)
phat <- unlogit(fit.logpsamed)
lines(0:9, phat, col=2, lwd=3)

b <- pros4.reg$coefficients
fit.logpsamed <- b[1] + b[2]*median(log(psa)) + b[3]*c(0:9)
phat <- unlogit(fit.logpsamed)
lines(0:9, phat, col=3, lwd=3, lty=2)
```