Mini Project 2

Matthew Lynn

Section 1

* 1. We first explore the data given.

Notice that gleason is ordered from 6 to 8. This measures the grade of the disease. It can be factored into categorical variables but for simplicity we leave it quantitative.

> str(colon)

'data.frame': 97 obs. of 9 variables:

$ subject : int 1 2 3 4 5 6 7 8 9 10 ...

$ psa : num 0.651 0.852 0.852 0.852 1.448 ...

$ cancervol: num 0.56 0.372 0.601 0.301 2.117 ...

$ weight : num 16 27.7 14.7 26.6 30.9 ...

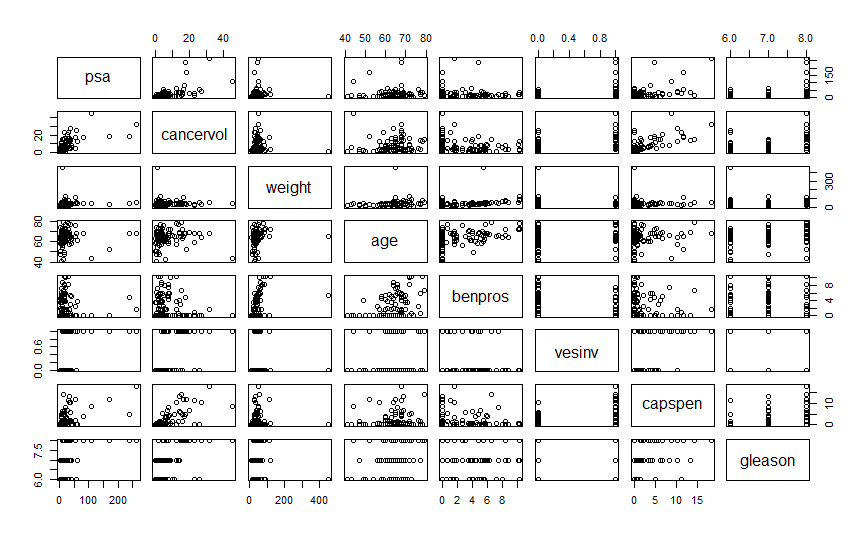
$ age : int 50 58 74 58 62 50 64 58 47 63 ...

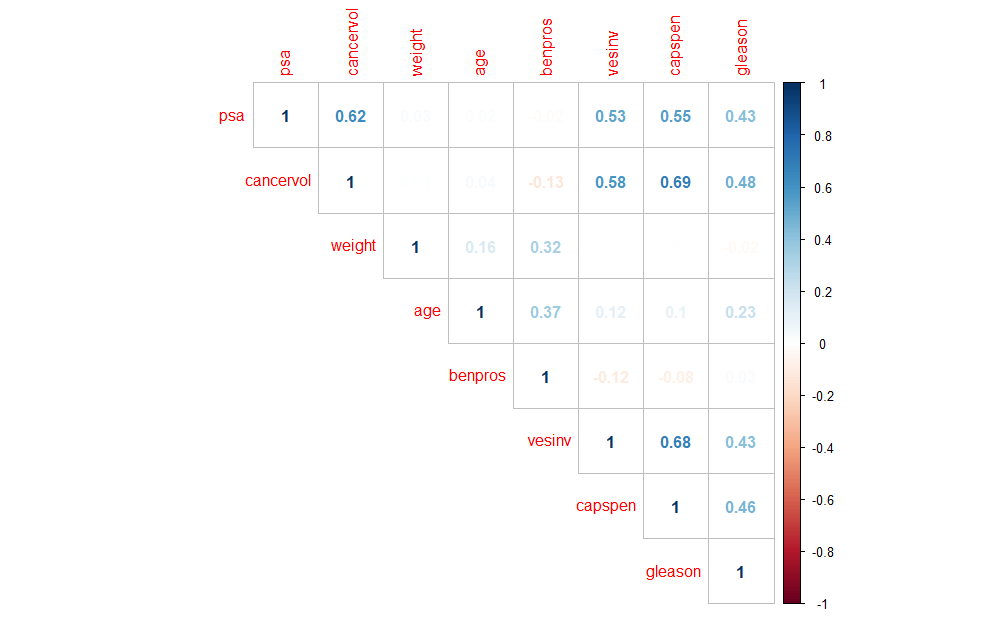
$ benpros : num 0 0 0 0 0 ...

$ vesinv : int 0 0 0 0 0 0 0 0 0 0 ...

$ capspen : num 0 0 0 0 0 0 0 0 0 0 ...

$ gleason : int 6 7 7 6 6 6 6 6 7 6 ...

We would like vesinv to be categorical, however we keep it as integer to run the correlation matrix functions and will consider changing it later if needed. We now look to the scatterplots to see if we can spot any obvious trends among different variables. We then slap a colorful correlation matrix down to help our eyes. A normal correlation matrix is output is included as well. Note that subject is excluded below.

> round(cor(colon), 2)

subject psa cancervol weight age benpros vesinv capspen gleason

subject 1.00 0.60 0.62 0.11 0.20 0.17 0.57 0.48 0.54

psa 0.60 1.00 0.62 0.03 0.02 -0.02 0.53 0.55 0.43

cancervol 0.62 0.62 1.00 0.01 0.04 -0.13 0.58 0.69 0.48

weight 0.11 0.03 0.01 1.00 0.16 0.32 0.00 0.00 -0.02

age 0.20 0.02 0.04 0.16 1.00 0.37 0.12 0.10 0.23

benpros 0.17 -0.02 -0.13 0.32 0.37 1.00 -0.12 -0.08 0.03

vesinv 0.57 0.53 0.58 0.00 0.12 -0.12 1.00 0.68 0.43

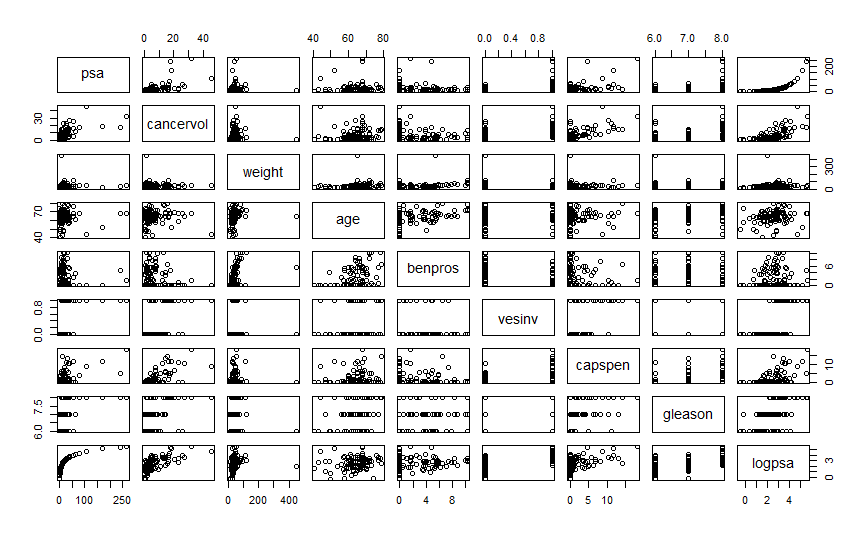
capspen 0.48 0.55 0.69 0.00 0.10 -0.08 0.68 1.00 0.46

gleason 0.54 0.43 0.48 -0.02 0.23 0.03 0.43 0.46 1.00

Right off the bat we see that

* psa and cancervol are 0.62 correlated.
* Cancervol and capspen are 0.69 correlated
* Vesinv and capspen are 0.68

These variables are highly correlated and may prove useful in further analysis.

1. We would like to see if psa by itself is a good response variable or if it needs to be transformed. Below is a scatterplot of log(psa) to be compared with the above.

Earlier, only cancervol showed any obvious trend for psa. Here, we see cancervol, benpros, and capspen. We could also use the means of vesinv and gleason as their midpoints seem to follow a positive trend. This clearly shows that log(psa) is a much better response variable. We can also show a comparison of residual plots using the plot() function. We compare the following models. Results on next page.

* response\_psa = lm(psa~.-logpsa-subject, data = colon)
* response\_logpsa = lm(logpsa~.-psa-subject, data = colon)

A shapiro test on the residuals of response\_psa rejects the null which states normality, but a shapiro test on the residuals of response\_logpsa results in accepting the null and thus shows that log(psa) *is* normally distributed. Below is a snippet of the code and results.

> shapiro.test(residuals(response\_psa))

Shapiro-Wilk normality test

data: residuals(response\_psa)

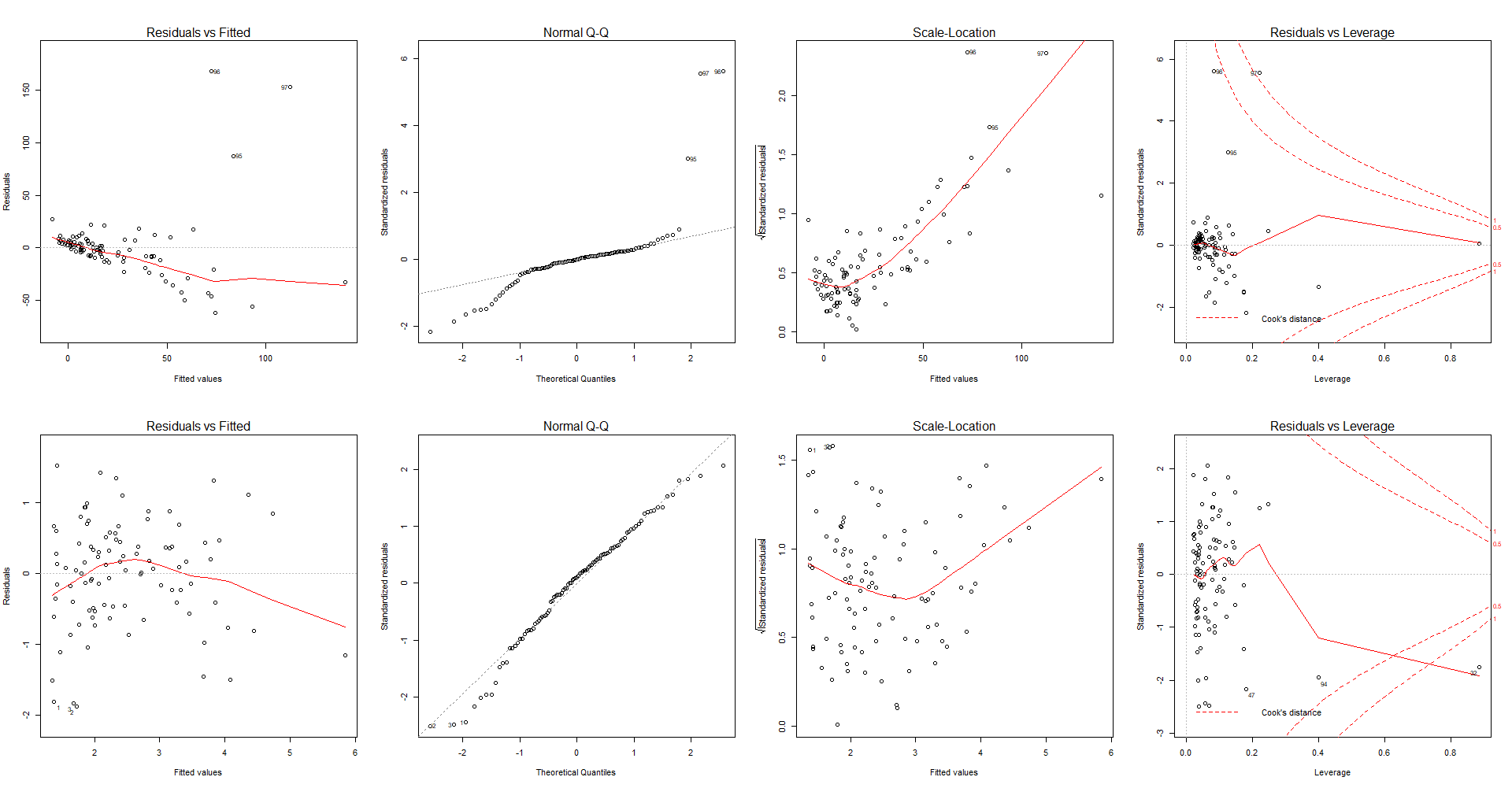
W = 0.65848, p-value = 1.075e-13

> shapiro.test(residuals((response\_logpsa)))

Shapiro-Wilk normality test

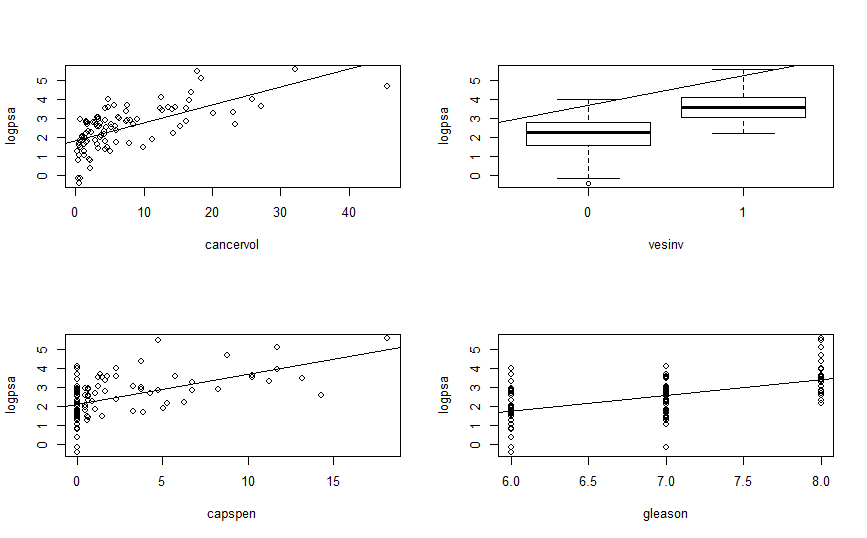
data: residuals((response\_logpsa))

W = 0.9817, p-value = 0.1954

The top row is the psa response model. The residual plot is no good, a clear trend with non-constant variance makes for a poor model. Additionally, the QQ plot is unacceptable which the shapiro test has already shown.

The bottom row is the logpsa model. The residual plot is much more spread out and cleaner looking. The QQ plot reveals that our model is normally distributed.

It should be noted that in general we look to the correlation matrix for ideas about our response, and as seen earlier many variables correlate with psa to begin building a model around psa being the response. This same method can be applied to other variables if we so desire.

1. We now fit a SLR model for each predictor to the response log(psa) except for subject (id variable) and psa. From here forward we use vesinv as a factored predictor. Only the significant SLR models are graphed below.

Above, each plot has the regression line to show how well it fitted.

1. Now we tackle some multiple regression models, namely one which contains all variables (except subject, psa) and a reduced model that only contains the significant predictors based on the summary of the full model. The results are as follows:

|  |  |
| --- | --- |
| Full model | Reduced model |
| lm(logpsa~.-subject-psa, data = colon) | lm(logpsa~cancervol+benpros+vesinv+gleason,data=colon) |

Coefficients:

Estimate Std. Error t value Pr(>|t|)

(Intercept) -0.685796 0.998754 -0.687 0.49409

cancervol 0.069454 0.014624 4.749 7.77e-06 \*\*\*

weight 0.001380 0.001822 0.757 0.45079

age -0.002799 0.011724 -0.239 0.81186

benpros 0.087470 0.029605 2.955 0.00401 \*\*

vesinv1 0.782623 0.268339 2.917 0.00448 \*\*

capspen -0.026521 0.032860 -0.807 0.42177

gleason 0.358153 0.127976 2.799 0.00629 \*\*

Analysis of Variance Table

Model 1: logpsa ~ cancervol + benpros + vesinv + gleason

Model 2: logpsa ~ (subject + psa + cancervol + weight + age + benpros +

vesinv + capspen + gleason) - subject - psa

Res.Df RSS Df Sum of Sq F Pr(>F)

1 92 53.229

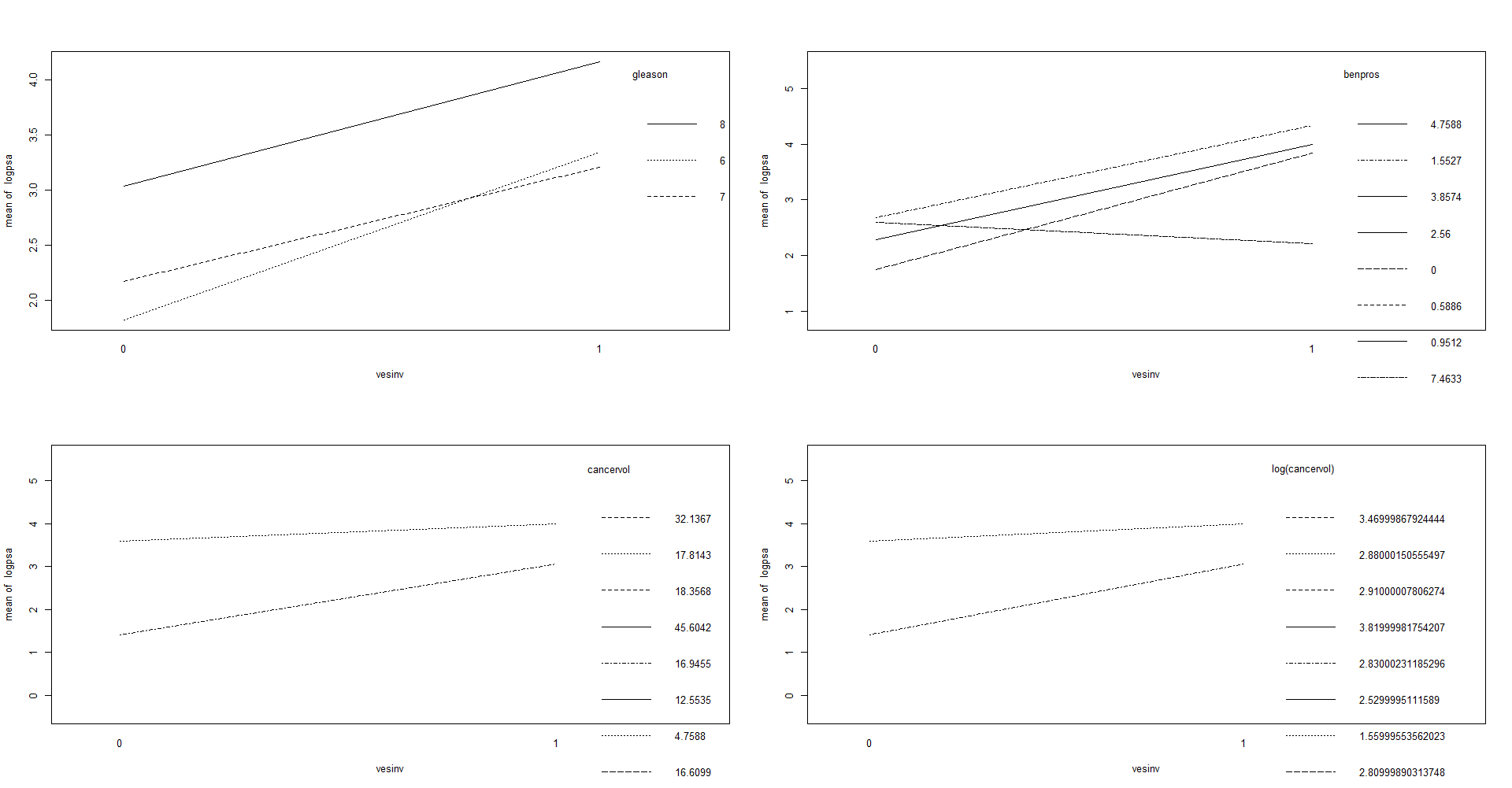
2 89 52.477 3 0.75232 0.4253 0.7353

We reject the null hypothesis for βj=0 on cancervol, benpros, vesinv (1), and gleason.

Thus, we drop weight, age, and capspen. We double check with anova.

We accept the null hypothesis for partial F-test saying models are equal.

Thus, the reduced model is better.

1. We want to build a reasonably good model. So, we check if interactions between our current predictors and vesinv, as well as other combos are worth adding to the model. After, we check if transforms of our quantitative variables help boost our adjusted R2.

We see that some of the interaction plots above are mostly parallel which hints that it won’t be useful for our model. On the other hand, lines that diverge away from each other or intersect may be of use. We test using no interactions as our reduced model vs. the addition of each of the above interactions. It turns out that benpros:vesinv and cancervol:vesinv may be useful based on the partial F-test. In the next part we ultimately check the addition of cancervol:vesinv to the model with benpros:vesinv and notice that adjusted-R2 goes down from 0.631 to 0.6248. A decrease when adding a seemingly significant interaction! A partial F-test clearly shows that the addition of the interaction term cancervol:vesinv is not needed.

Alternatively, we use our current model with cancervol, benpros, vesinv, and gleason to fully fit a new model with all possible interactions. We then use partial F-test to see if any of the new terms are useful:

Analysis of Variance Table

Model 1: logpsa ~ cancervol + benpros + vesinv + gleason

Model 2: logpsa ~ cancervol \* benpros \* vesinv \* gleason

Res.Df RSS Df Sum of Sq F Pr(>F)

1 92 53.229

2 81 44.595 11 8.6335 1.4256 0.1776

Since our p-value is high we can drop all interactions and keep our much simpler model.

This is much more convenient than the previous convention.

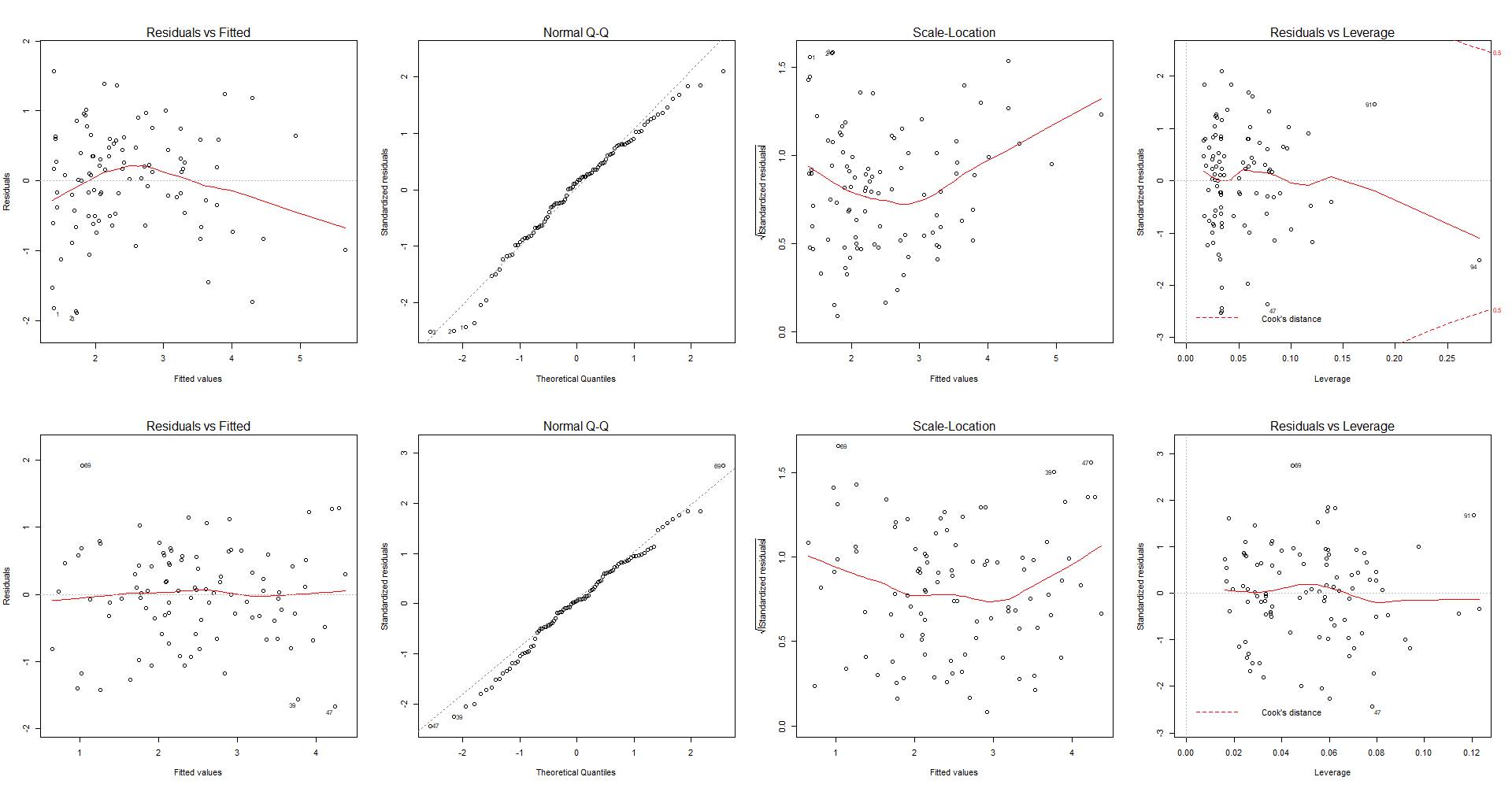
Now we look back to the scatterplot (reduced, shown in code) to see if we can visually mark what transformations we should apply to the predictors if needed.

We pair our visual aid with the shapiro test for normality and a normal density graph.

Many different transformations were tried, we leave most out for simplicity.

We got above .60 adjusted-R2 using a 5th degree polynomial on capspen paired with a log(cancervol), gleason, and vesinv. This is a rather complicated model, so we keep it out as well.

Ultimately, we test two models as we did with response and compare residual plots.

* untransformed = lm(logpsa~cancervol+benpros+vesinv+gleason, data = colon)
* logtransform = lm(logpsa~log(cancervol)+benpros+vesinv+gleason, data = colon)

Now, the top row is the previous model with no transformed predictors. The residual plot was an improvement on the untransformed response model but it still wasn’t as spread out as we would like.

The bottom row is our log transform for cancervol with all else remaining the same. The residual plot is much nicer here. We continue with the new model.

Thus, our final model is:

Log(psa) = log(cancervol) + benpros + vesinv + gleason

Which results in an adjusted-R2 of 0.6138.

* + - 1. Below is a snippet of our model summary

1. Coefficients:
2. Estimate Std. Error t value Pr(>|t|)
3. (Intercept) -0.31633 0.76947 -0.411 0.68195
4. log(cancervol) 0.50503 0.07957 6.347 8.17e-09 \*\*\*
5. benpros 0.06419 0.02448 2.622 0.01024 \*
6. vesinv1 0.65881 0.21754 3.028 0.00319 \*\*
7. gleason 0.26294 0.11804 2.228 0.02835 \*

YLog(psa) = β0 + β1xlog(cancervol) + β2xbenpros + β3xvesinv(1) + β4xgleason

Note that β0 is the baseline for vesinv(0)

YLog(psa) = -0.31633 + 0.50503xlog(cancervol) + 0.06419xbenpros + 0.65881xvesinv(1) + 0.26294xgleason

* + - 1. Finally, we would like to use the final model to predict the PSA level for a patient whose quantitative predictors are at the sample means of the variables and qualitative predictors are at the most frequent category. Simply put we use the following predict function after defining the means of our quantitative predictors and running a table on our vesinv variable to see that 0 is counted most frequently.

> table(colon$vesinv) # we use 0

0 1

76 21

> a = mean(log(cancervol))

> b = mean(benpros)

> d = mean(gleason)

>

> predict(final\_model, data.frame(cancervol=a, benpros=b, vesinv=as.factor(0), gleason=d))

1

1.80585

Our prediction comes out to be 1.80585. This number can also be obtained by manually putting in the mean values of our quantitative predictors in the corresponding x variables in the previous equation and inputting the most frequent category in the qualitative predictor into the corresponding x variable.

Section 2

# Mini Project 2

library(corrplot)

library(ggplot2)

library(GGally)

library(rcompanion)

# part a, data exploration

colon = read.csv("prostate\_cancer.csv", header = T)

head(colon)

str(colon)

summary(colon)

pairs(subset(colon, select = -subject))

round(cor(subset(colon, select = -subject)), 2)

corrplot(cor(subset(colon, select = -subject)), method = "number", type = "upper")

attach(colon)

# part b

# check if psa is a good response

par(mfrow = c(2,4))

colon$logpsa = log(colon$psa)

attach(colon)

pairs(subset(colon, select = -subject))

response\_psa = lm(psa~.-logpsa-subject, data = colon)

response\_logpsa = lm(logpsa~.-psa-subject, data = colon)

plot(response\_psa)

plot(response\_logpsa)

shapiro.test(residuals(response\_psa))

shapiro.test(residuals(response\_logpsa))

# super cool scatterplot, density plot, and corr matrix

# can be used to gain some info by comparing logpsa to psa

ggpairs(data=subset(colon, select = -subject), title="colon data")

# part c

# fit a million SLR's...

colon$vesinv = as.factor(colon$vesinv)

attach(colon)

fitcv = lm(logpsa ~ cancervol)

fitwt = lm(logpsa ~ weight)

fitage = lm(logpsa ~ age)

fitbp = lm(logpsa ~ benpros)

fitvi = lm(logpsa ~ vesinv)

fitcp = lm(logpsa ~ capspen)

fitgle = lm(logpsa ~ gleason)

summary(fitcv) # significant

summary(fitwt)

summary(fitage)

summary(fitbp)

summary(fitvi) # significant

summary(fitcp) # significant

summary(fitgle) # significant

par(mfrow = c(2,2))

plot(logpsa~cancervol)

abline(fitcv)

plot(logpsa~vesinv)

abline(fitvi)

plot(logpsa~capspen)

abline(fitcp)

plot(logpsa~gleason)

abline(fitgle)

# part d

# check all varialbes in model and check null hypothesis

fitall = lm(logpsa~.-subject-psa, data = colon)

summary(fitall)

fitless = lm(logpsa~cancervol+benpros+vesinv+gleason, data = colon)

summary(fitless)

# we accept the null hypothesis for saying models are equal

# thus the fitless model is better

anova(fitless,fitall)

# part e

# lets try some interactions

par(mfrow = c(2,2))

interaction.plot(vesinv, gleason, logpsa, fun = mean)

interaction.plot(vesinv, benpros, logpsa, fun = mean)

interaction.plot(vesinv, cancervol, logpsa, fun = mean)

interaction.plot(vesinv, log(cancervol), logpsa, fun = mean)

# test the interactions below

fit\_interactions = lm(logpsa~cancervol+benpros+vesinv\*gleason, data = colon)

summary(fit\_interactions)

# to double check we use anova

anova(fitless, fit\_interactions)

# we accept the null

# perhaps a faster way of checking interactions

# fit a full model of all possible interactions

fit\_interactions\_full = lm(logpsa~ cancervol\*benpros\*vesinv\*gleason, data = colon)

anova(fitless, fit\_interactions\_full)

# lets try some transformations now

ggpairs(data=subset(colon, select = -c(subject, psa, age, weight)), title="colon data")

# change out variables below to test normality

shapiro.test(capspen)

shapiro.test(capspen^(1/3))

par(mfrow = c(3,2))

plotNormalHistogram(cancervol, main="Untransformed")

plotNormalHistogram(log(cancervol), main="log transformation")

plotNormalHistogram(capspen^2, main="square transformation")

plotNormalHistogram(capspen^3, main="cube transformation")

plotNormalHistogram(sqrt(capspen), main="sqrt transformation")

plotNormalHistogram(capspen^(1/3), main="cbrt transformation")

par(mfrow = c(2,1))

# plot model with cancervol as normal and log(cancervol) in another model

par(mfrow = c(2,4))

untransformed = lm(logpsa~cancervol+benpros+vesinv+gleason, data = colon)

logtransform = lm(logpsa~log(cancervol)+benpros+vesinv+gleason, data = colon)

plot(untransformed)

plot(logtransform)

# # from above we play with different combos below

# summary((lm(logpsa ~ log(cancervol)+benpros+vesinv+gleason+benpros:vesinv)))

# full\_final\_fit = lm(logpsa ~ log(cancervol)+benpros\*vesinv+gleason)

# summary(full\_final\_fit)

# final\_fit = lm(logpsa ~ log(cancervol)+benpros+vesinv+gleason)

# summary(final\_fit)

# # final test to check if we should include benpros:vesinv interaction

# anova(final\_fit,full\_final\_fit)

# part f

# final model is logtransformed

# we set it as final\_model for clarity

final\_model = logtransform

summary(final\_model)

# part g

# we want to predict with mean(quantitatives) and max(count(qualitatives))

table(colon$vesinv) # we use 0

a = mean(log(cancervol))

b = mean(benpros)

d = mean(gleason)

predict(final\_model, data.frame(cancervol=a, benpros=b, vesinv=as.factor(0), gleason=d))