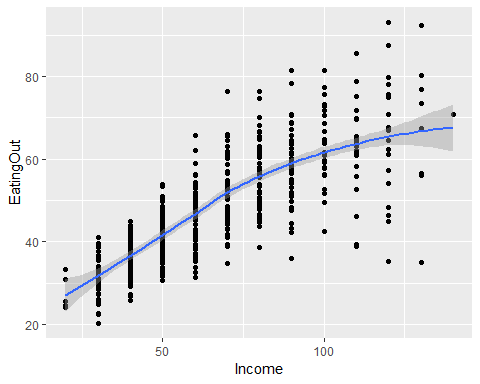
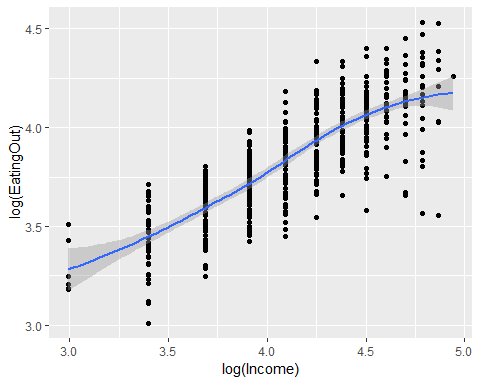
Homework 2

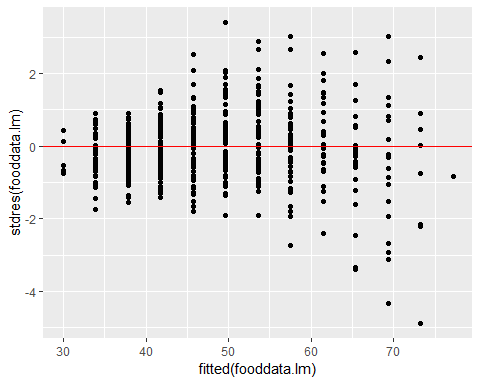
# 1





There appears to be a linear relationship between income and eating out with some changing variance. Looking at a log transform it appears to be more linear, suggesting a heteroskedastic linear model for our fit.

# 2



##   
## studentized Breusch-Pagan test  
##   
## data: fooddata.lm  
## BP = 66.388, df = 1, p-value = 3.704e-16

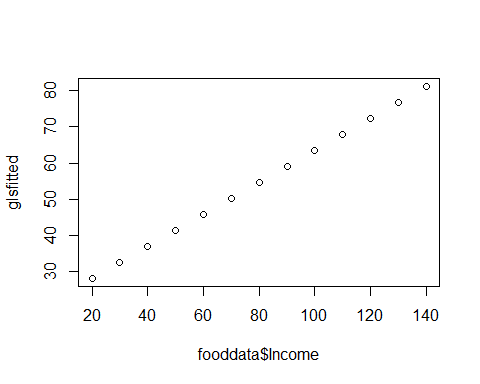
The homoskedastic model we use is y = X \* B + e where y is the number eating Out, x is our explanatory matrix, beta is our coefficients, and e is the residuals with mean 0 and variance sigma-squared x the identity matrix. The plot of fitted values by residuals appears to have expanding variance as the fitted values increase, and this is confirmed by failing the bptest. Thus this homoskedastic model violates the equal variances assumption for a linear model.

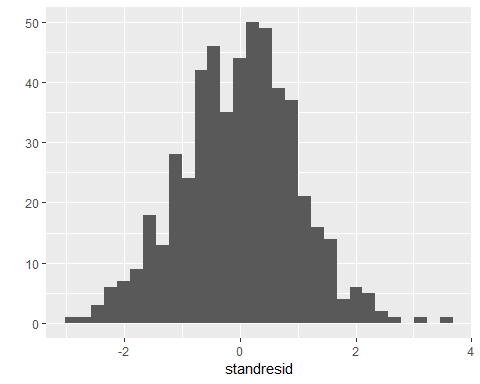
# 3

The heteroskedastic model we use is y is normally distributed with mean X x B, and variance sigma-squared x D; where X is a n x (P + 1) matrix of the explanatory variables, B is the (P + 1) x 1 vector of coefficients explanatory variable, D is n x n matrix with exp(2 *Income*  theta) for its values. For this we use the gls function to create our model. The statistical inference of this model can give an accurate prediction of the effect of income on food expenditure.

# 4

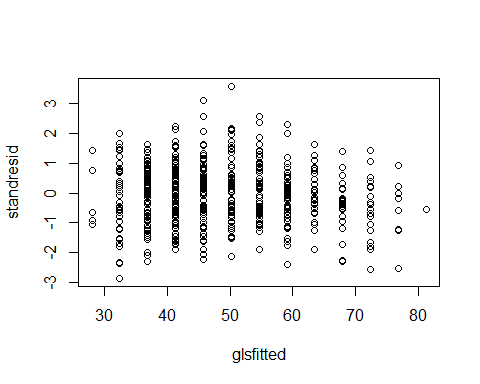
Linearity - the relationship between y and all explanatory variables is linear (only applies to quantitative explanatory variables). Independence - the salary of one person does not impact the salary of another. Normality - the residuals e are normall distributed. Equal variance (but only after normalizing) - the residuals have equal variation across the whole fitted line.





## Kolmogorov-Smirnov test

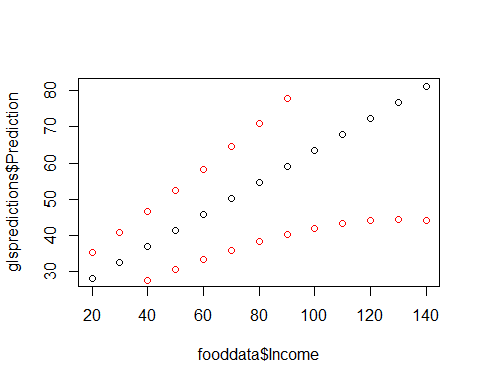
##   
## One-sample Kolmogorov-Smirnov test  
##   
## data: standresid  
## D = 0.027929, p-value = 0.8093  
## alternative hypothesis: two-sided



Linearity - The Income by fitted appears to be linear. Independence - There is no cause to suspect that the same person appears twice in this data set thus keeping independence. Normality - A histogram of the standardized residuals appears to be normal and has a ks test that passes. Equal variance - The plot of fitted values by standardized residuals appears to not have differing variances.

# 5

The mean RPMSE is 7.961076 and the mean of coverage is 0.942682.



# 6

Betahat for income is 0.4430501 with a 95% confidence interval of (0.4165682, 0.4695319). As income goes up by one thousand, the amount of eating out on average increases between 0.4165 and 0.4695 95% of the time. The variance parameters are sigma squared and theta hat, sigma squared has estimate of 7.973177 and 95% confidence interval of (5.702826, 11.147377). This is saying that the value for eating out varies across all levels of income. Theta has an estimate of 0.01358099 and confidence interval (0.01121274, 0.01594923). This is saying that higher income has higher variance, or variability in eating out.

# 7

##   
## Simultaneous Tests for General Linear Hypotheses  
##   
## Fit: gls(model = EatingOut ~ Income, data = fooddata, weights = varExp(form = ~Income),   
## method = "ML")  
##   
## Linear Hypotheses:  
## Estimate Std. Error z value Pr(<z)   
## 1 >= 0.5 0.44305 0.01351 -4.215 1.25e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
## (Adjusted p values reported -- single-step method)

We use the null hypothesis that Beta income = 0.5 and alternative hypothesis that Beta income < 0.5. Running a statistical test for this we get a p-value of 1.25e-05 and reject the null hypothesis and conclude that the economy is not “healthy” for restaurant owners.

# 8

With a desired income of $55000 after graduation I would predict that I would spend $41.34 a week on food not cooked at home with a 95% confidence interval of (30.40232, 52.2805).

Code

#1

fooddata <- read.table("FoodExpenses.txt", header = TRUE)  
ggplot(data=fooddata, aes(x = Income, y = EatingOut)) + geom\_point() + geom\_smooth()

ggplot(data=fooddata, aes(x = log(Income), y = log(EatingOut))) + geom\_point() + geom\_smooth()

#2

fooddata.lm <- lm(EatingOut~Income, data=fooddata)  
qplot(fitted(fooddata.lm), stdres(fooddata.lm), geom="point") +  
 geom\_hline(yintercept=0, color="red")

bptest(fooddata.lm)

#4

fooddata.gls <- gls(EatingOut~Income, data=fooddata, weights=varExp(form=~Income), method="ML")  
betahat <- fooddata.gls$coefficients   
thetahat <- coef(fooddata.gls$modelStruct, unconstrained=FALSE)  
sigma <- fooddata.gls$sigma  
glsfitted <- fitted(fooddata.gls)  
standresid <- resid(object=fooddata.gls, type="pearson")

plot(fooddata$Income, glsfitted)

qplot(standresid, geom="histogram")

ks.test(standresid, "pnorm")  
plot(glsfitted, standresid)

#5

source("predictgls.R")  
n.cv <- 100 #Number of CV studies to run  
n.test <- nrow(fooddata) / 2 #Number of observations in a test set  
n <- nrow(fooddata)  
rpmse <- rep(x=NA, times=n.cv)  
bias <- rep(x=NA, times=n.cv)  
wid <- rep(x=NA, times=n.cv)  
cvg <- rep(x=NA, times=n.cv)  
alpha <- 0.05  
for(cv in 1:n.cv){  
 ## Select test observations  
   
 test.obs <- sample(x=1:n, size=n.test)  
   
 ## Split into test and training sets  
 test.set <- fooddata[test.obs,]  
 train.set <- fooddata[-test.obs,]  
   
 ## Fit a lm() using the training data  
 train.gls <- gls(EatingOut~Income, data=train.set , weights=varExp(form =~Income), method="ML")  
   
 ## Generate predictions for the test set  
 my.preds <- predictgls(train.gls, newdframe=test.set)  
 pred.int.low <- my.preds$Prediction - qt(1-alpha/2, df=nrow(train.set)-length(coef(train.gls)))\*my.preds$SE.pred  
 pred.int.up <- my.preds$Prediction + qt(1-alpha/2, df=nrow(train.set)-length(coef(train.gls)))\*my.preds$SE.pred  
 ## Calculate bias  
 bias[cv] <- mean(test.set[['EatingOut']]-exp(my.preds[,'Prediction']))  
   
 ## Calculate RPMSE  
 rpmse[cv] <- (test.set[['EatingOut']]-my.preds[,'Prediction'])^2 %>% mean() %>% sqrt()  
   
 ## Calculate Coverage  
 cvg[cv] <- mean((test.set$EatingOut > pred.int.low) & (test.set$EatingOut < pred.int.up))  
   
 ## Calculate Width  
 wid[cv] <- (pred.int.up - pred.int.low) %>% mean()  
   
}  
mean(rpmse)

## [1] 7.998038

mean(cvg)

## [1] 0.9420307

glspredictions <-predictgls(fooddata.gls, newdframe = fooddata)  
pred.int.low <- glspredictions$Prediction - qt(1-alpha/2, df=nrow(fooddata)-length(coef(fooddata.gls)))\*glspredictions$SE.pred  
pred.int.up <- glspredictions$Prediction + qt(1-alpha/2, df=nrow(fooddata)-length(coef(fooddata.gls)))\*glspredictions$SE.pred  
plot(fooddata$Income, glspredictions$Prediction)  
points(fooddata$Income, pred.int.low, col = "red")  
points(fooddata$Income, pred.int.up, col = "red")

#6

betahat

confint(fooddata.gls)

sigma^2

food.int <- intervals(fooddata.gls, level=0.95)  
theta <- food.int$varStruct  
theta

sigmasquared <- food.int$sigma ^2  
sigmasquared

#7

a <- matrix(c(0,1), nrow = 1)  
glht(fooddata.gls, linfct=a, rhs=0.5, alternative ="less") %>% summary

#8

my.prediction <- predictgls(fooddata.gls, newdframe = data.frame(Income = 50))  
my.prediction$Prediction

## [1] 41.34141

mypred.int.low <- my.prediction$Prediction - qt(1-alpha/2, df=nrow(fooddata)-length(coef(fooddata.gls)))\*my.prediction$SE.pred  
mypred.int.up <- my.prediction$Prediction + qt(1-alpha/2, df=nrow(fooddata)-length(coef(fooddata.gls)))\*my.prediction$SE.pred  
mypred.int.low

## [1] 30.40232

mypred.int.up

## [1] 52.2805