St 503 HW 4

Robin Baldeo

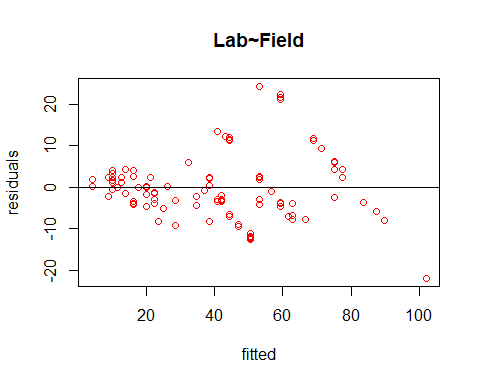
### Question 8.1

#### (a)

mod<- lm(Lab~Field, data = pipeline)  
summary(mod)

##   
## Call:  
## lm(formula = Lab ~ Field, data = pipeline)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -21.985 -4.072 -1.431 2.504 24.334   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.96750 1.57479 -1.249 0.214   
## Field 1.22297 0.04107 29.778 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 7.865 on 105 degrees of freedom  
## Multiple R-squared: 0.8941, Adjusted R-squared: 0.8931   
## F-statistic: 886.7 on 1 and 105 DF, p-value: < 2.2e-16

plot(mod$fitted.values, mod$residuals,xlab = "fitted", ylab = "residuals" ,col = "Red", main = "Lab~Field")  
abline(h=0)



From the plot we see that the constant variance assumption has been violated because of the clear mega phone shape.

#### (b)

i <- order(pipeline$Field)   
npipe <- pipeline[i,]   
ff <- gl(12,9)[-108]   
meanfield <- unlist(lapply(split(npipe$Field,ff),mean))   
varlab <- unlist(lapply(split(npipe$Lab,ff),var))  
  
#not removing the last obs from varlab and meanfield  
modb<-lm(I(log(varlab))~I(log(meanfield)))  
summary(modb)

##   
## Call:  
## lm(formula = I(log(varlab)) ~ I(log(meanfield)))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.2038 -0.6729 0.1656 0.7205 1.1891   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.3538 1.5715 -0.225 0.8264   
## I(log(meanfield)) 1.1244 0.4617 2.435 0.0351 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.018 on 10 degrees of freedom  
## Multiple R-squared: 0.3723, Adjusted R-squared: 0.3095   
## F-statistic: 5.931 on 1 and 10 DF, p-value: 0.03513

#removing the last obs from both varlab and meanfield  
modb<-lm(I(log(varlab[-length(varlab)]))~I(log(meanfield[-length(meanfield)])))  
summary(modb)

##   
## Call:  
## lm(formula = I(log(varlab[-length(varlab)])) ~ I(log(meanfield[-length(meanfield)])))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.00477 -0.42268 0.05989 0.37854 0.93815   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) -1.9352 1.0929 -1.771 0.110403  
## I(log(meanfield[-length(meanfield)])) 1.6707 0.3296 5.070 0.000672  
##   
## (Intercept)   
## I(log(meanfield[-length(meanfield)])) \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.657 on 9 degrees of freedom  
## Multiple R-squared: 0.7406, Adjusted R-squared: 0.7118   
## F-statistic: 25.7 on 1 and 9 DF, p-value: 0.0006723

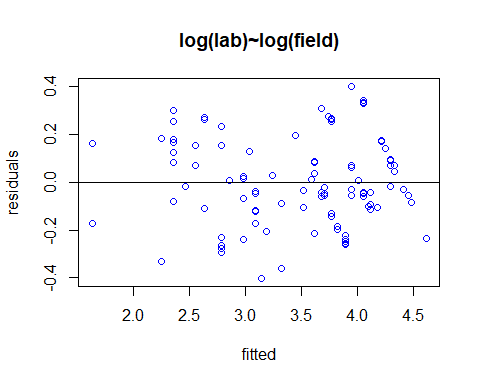
After removing the last points as recommended by the book, the a0 is -1.9351674 and a1 is 1.6707234

#### (c)

# applying log to both lab and field appears to correct the non cocnstant varaince voilation  
modc<- lm(I(log(Lab))~I(log(Field)), data = pipeline)  
summary(modc)

##   
## Call:  
## lm(formula = I(log(Lab)) ~ I(log(Field)), data = pipeline)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.40212 -0.11853 -0.03092 0.13424 0.40209   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.06849 0.09305 -0.736 0.463   
## I(log(Field)) 1.05483 0.02743 38.457 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1837 on 105 degrees of freedom  
## Multiple R-squared: 0.9337, Adjusted R-squared: 0.9331   
## F-statistic: 1479 on 1 and 105 DF, p-value: < 2.2e-16

#plot of the log(lab)~log(field)  
plot(modc$fitted.values, modc$residuals,xlab = "fitted", ylab = "residuals" ,col = "blue", main="log(lab)~log(field)")  
abline(h=0)



Applying log to both the predictor and response appears to correct the volition of constant variance.

### Question 8.3

#head(salmonella)  
  
mod\_s<-lm(colonies~log(dose +1), data=salmonella)  
summary(mod\_s)

##   
## Call:  
## lm(formula = colonies ~ log(dose + 1), data = salmonella)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -16.376 -6.882 -1.509 5.400 29.119   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 19.823 5.064 3.915 0.00123 \*\*  
## log(dose + 1) 2.396 1.128 2.125 0.04955 \*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 10.84 on 16 degrees of freedom  
## Multiple R-squared: 0.2201, Adjusted R-squared: 0.1713   
## F-statistic: 4.514 on 1 and 16 DF, p-value: 0.04955

mod\_c<-lm(colonies~factor(log(dose +1)), data=salmonella)  
summary(mod\_c)

##   
## Call:  
## lm(formula = colonies ~ factor(log(dose + 1)), data = salmonella)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -15.667 -3.917 -0.500 3.417 17.333   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 21.667 5.506 3.935 0.00198  
## factor(log(dose + 1))2.39789527279837 -3.333 7.787 -0.428 0.67617  
## factor(log(dose + 1))3.52636052461616 3.333 7.787 0.428 0.67617  
## factor(log(dose + 1))4.61512051684126 21.000 7.787 2.697 0.01942  
## factor(log(dose + 1))5.8111409929767 15.667 7.787 2.012 0.06722  
## factor(log(dose + 1))6.90875477931522 8.000 7.787 1.027 0.32449  
##   
## (Intercept) \*\*  
## factor(log(dose + 1))2.39789527279837   
## factor(log(dose + 1))3.52636052461616   
## factor(log(dose + 1))4.61512051684126 \*   
## factor(log(dose + 1))5.8111409929767 .   
## factor(log(dose + 1))6.90875477931522   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 9.536 on 12 degrees of freedom  
## Multiple R-squared: 0.5475, Adjusted R-squared: 0.359   
## F-statistic: 2.904 on 5 and 12 DF, p-value: 0.06047

anova(mod\_s, mod\_c)

## Analysis of Variance Table  
##   
## Model 1: colonies ~ log(dose + 1)  
## Model 2: colonies ~ factor(log(dose + 1))  
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 16 1881.1   
## 2 12 1091.3 4 789.73 2.1709 0.1342

Based on the p-value > .05 from the ANOVA test we fail to reject the null. As a result, a linear model is satisfactory.

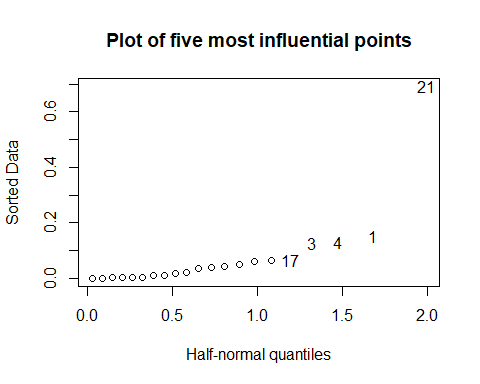
### question 8.5

#### (a)

#a  
#least square method  
  
ls<- lm(stack.loss~., data = stackloss)  
summary(ls)

##   
## Call:  
## lm(formula = stack.loss ~ ., data = stackloss)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -7.2377 -1.7117 -0.4551 2.3614 5.6978   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -39.9197 11.8960 -3.356 0.00375 \*\*   
## Air.Flow 0.7156 0.1349 5.307 5.8e-05 \*\*\*  
## Water.Temp 1.2953 0.3680 3.520 0.00263 \*\*   
## Acid.Conc. -0.1521 0.1563 -0.973 0.34405   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 3.243 on 17 degrees of freedom  
## Multiple R-squared: 0.9136, Adjusted R-squared: 0.8983   
## F-statistic: 59.9 on 3 and 17 DF, p-value: 3.016e-09

#Checking for influential points, then looking at the 5 most influential points  
halfnorm(cooks.distance(ls), 5, names(cooks.distance(ls)), main="Plot of five most influential points")



#### (b)

#lad method  
  
lad<- rq(stack.loss~., data = stackloss)  
summary(lad)

##   
## Call: rq(formula = stack.loss ~ ., data = stackloss)  
##   
## tau: [1] 0.5  
##   
## Coefficients:  
## coefficients lower bd upper bd   
## (Intercept) -39.68986 -41.61973 -29.67754  
## Air.Flow 0.83188 0.51278 1.14117  
## Water.Temp 0.57391 0.32182 1.41090  
## Acid.Conc. -0.06087 -0.21348 -0.02891

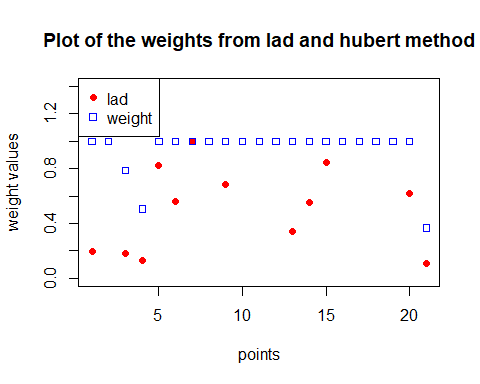
#### (c)

#hubert method  
  
hub<- rlm(stack.loss~., data = stackloss)  
summary(hub)

##   
## Call: rlm(formula = stack.loss ~ ., data = stackloss)  
## Residuals:  
## Min 1Q Median 3Q Max   
## -8.91753 -1.73127 0.06187 1.54306 6.50163   
##   
## Coefficients:  
## Value Std. Error t value   
## (Intercept) -41.0265 9.8073 -4.1832  
## Air.Flow 0.8294 0.1112 7.4597  
## Water.Temp 0.9261 0.3034 3.0524  
## Acid.Conc. -0.1278 0.1289 -0.9922  
##   
## Residual standard error: 2.441 on 17 degrees of freedom

#### (d)

#additional part added  
# lad weights  
w\_lad<- 1/abs(lad$residuals)  
  
#hubert weight  
w\_hub<- hub$w  
  
#plot of the weights of lad and hubert method  
plot(c(1:21), w\_lad, ylim = c(0, 1.4), pch = 19, col = "red", xlab = "points", ylab = "weight values", main= "Plot of the weights from lad and hubert method")  
lines(c(1:21), w\_hub, col="blue", type ="o", lty = 0 , pch = 22)  
nam<- c("red", "blue")  
legend("topleft", legend = c("lad", "weight"),col=nam, pch=c(19,22))



From the halfnorm plot in part a, we see that the top 5 influential points are 21, 1,4,3, and 17. Now with the overlay plot of the weights of LAD and Hubert, we see the weight of most of these points are reduced in relation to the other points. With LAD, 1,3,4,and 21 weights are reduced. While with the Hubert 3,4, and 21 are assigned a lower weight. Overall these two methods prove that they are not as sensitive to the influence points as with least square and are therefore robust.

#### Comparison

Compare the results. Now use diagnostic methods to detect any outliers or influential points. Remove these points and then use least squares. Compare the results.

#based on halfnorm point 21, 1 , 4, 3 and 17 appear to exert the top five influence on the model.  
#removing these points  
  
#  
stackloss\_n<- stackloss[c(-21,-1, -4, -3,-17),]  
  
#  
  
ls\_n<- lm(stack.loss~., data = stackloss\_n)  
summary(ls\_n)

##   
## Call:  
## lm(formula = stack.loss ~ ., data = stackloss\_n)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.5776 -0.7936 0.1845 0.7313 1.8323   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -36.18166 6.22963 -5.808 8.37e-05 \*\*\*  
## Air.Flow 0.79440 0.07030 11.301 9.41e-08 \*\*\*  
## Water.Temp 0.58944 0.17460 3.376 0.00551 \*\*   
## Acid.Conc. -0.08446 0.07836 -1.078 0.30226   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.296 on 12 degrees of freedom  
## Multiple R-squared: 0.9739, Adjusted R-squared: 0.9674   
## F-statistic: 149.2 on 3 and 12 DF, p-value: 9.202e-10

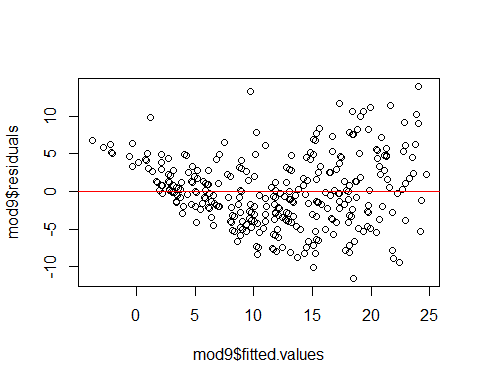
When I remove the top 5 influence points 21,1,4,3, and 17. Our new modified least square parameter coefficients resemble the LAD method . This is not a surprise because the lad method down weighted 4 of the 5 influence points detected by the halfnorm. However, with Hubert method, since 3 or the 5 points are down weighted our estimate coefficients were not a close match, but when I remove just the 3 points down weighted by Hubert our coefficients were every close. This also confirms that LAD and Hubert are not sensitive to influence points as the least square.

### question 9.3

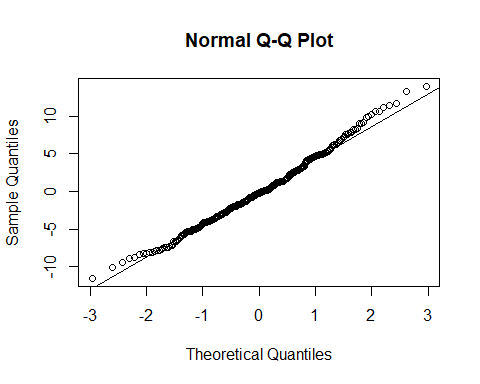
#question 9.3  
  
#head(ozone)  
  
mod9<- lm(O3~temp+humidity+ibh, data= ozone)  
  
summary(mod9)

##   
## Call:  
## lm(formula = O3 ~ temp + humidity + ibh, data = ozone)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -11.5291 -3.0137 -0.2249 2.8239 13.9303   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -1.049e+01 1.616e+00 -6.492 3.16e-10 \*\*\*  
## temp 3.296e-01 2.109e-02 15.626 < 2e-16 \*\*\*  
## humidity 7.738e-02 1.339e-02 5.777 1.77e-08 \*\*\*  
## ibh -1.004e-03 1.639e-04 -6.130 2.54e-09 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 4.524 on 326 degrees of freedom  
## Multiple R-squared: 0.684, Adjusted R-squared: 0.6811   
## F-statistic: 235.2 on 3 and 326 DF, p-value: < 2.2e-16

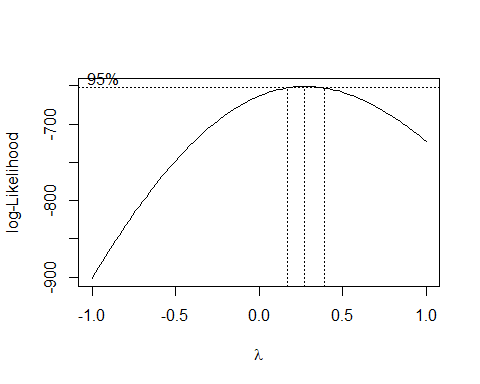
plot(mod9$fitted.values, mod9$residuals)  
abline(h=0, col= "red")



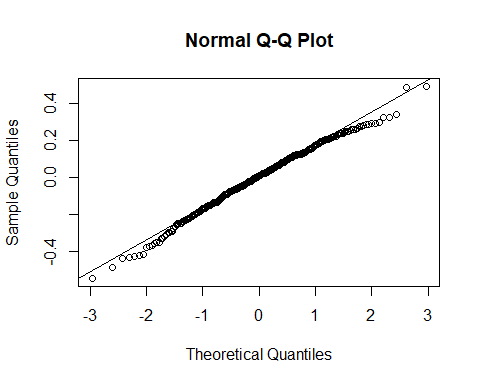
#the constant variance assumption appears violated  
  
  
qqnorm(mod9$residuals)  
qqline(mod9$residuals)



boxcox(mod9, plotit = T, lambda=seq(-1,1, by = .1))



#using lambda of of .25 from box cox in response transformation   
mod9\_<- lm(I(O3\*\*.25)~temp+humidity+ibh, data= ozone)  
  
  
qqnorm(mod9\_$residuals)  
qqline(as.data.frame(mod9\_$residuals))

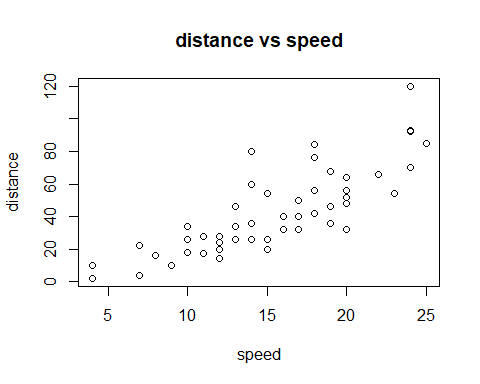


Using the value of .25 from the box cox, we apply this value to the response and based on the new qqplot the problem of long tail appears corrected.

### question 9.8

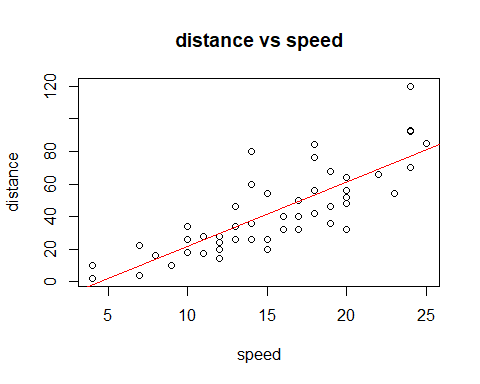
#### (a)

#a  
plot(cars$speed, cars$dist, xlab = "speed", ylab = "distance", main = "distance vs speed")



#### (b)

#b  
plot(cars$speed, cars$dist, xlab = "speed", ylab = "distance", main = "distance vs speed")  
mod8<- lm(dist~speed, data = cars)  
abline(mod8, col = "red")



summary(mod8)

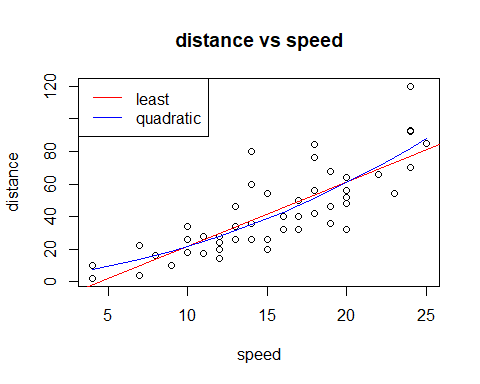
##   
## Call:  
## lm(formula = dist ~ speed, data = cars)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -29.069 -9.525 -2.272 9.215 43.201   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -17.5791 6.7584 -2.601 0.0123 \*   
## speed 3.9324 0.4155 9.464 1.49e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 15.38 on 48 degrees of freedom  
## Multiple R-squared: 0.6511, Adjusted R-squared: 0.6438   
## F-statistic: 89.57 on 1 and 48 DF, p-value: 1.49e-12

#### (c)

#c  
plot(cars$speed, cars$dist, xlab = "speed", ylab = "distance", main = "distance vs speed")  
  
mod8q<- lm(dist~poly(speed, 2), data = cars)  
summary(mod8q)

##   
## Call:  
## lm(formula = dist ~ poly(speed, 2), data = cars)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -28.720 -9.184 -3.188 4.628 45.152   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 42.980 2.146 20.026 < 2e-16 \*\*\*  
## poly(speed, 2)1 145.552 15.176 9.591 1.21e-12 \*\*\*  
## poly(speed, 2)2 22.996 15.176 1.515 0.136   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 15.18 on 47 degrees of freedom  
## Multiple R-squared: 0.6673, Adjusted R-squared: 0.6532   
## F-statistic: 47.14 on 2 and 47 DF, p-value: 5.852e-12

abline(mod8, col = "red")  
lines(cars$speed, mod8q$fitted.values, col= "blue")  
legend("topleft", legend = c("least", "quadratic"),col=c("red", "blue"), lty=c(1,1))

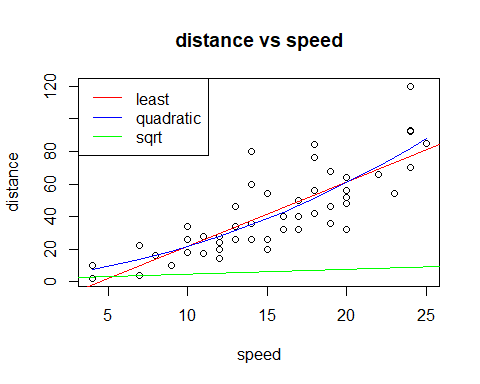


#### (d)

#d  
plot(cars$speed, cars$dist, xlab = "speed", ylab = "distance", main = "distance vs speed")  
mod8s<- lm(I(sqrt(dist))~speed, data = cars)  
summary(mod8s)

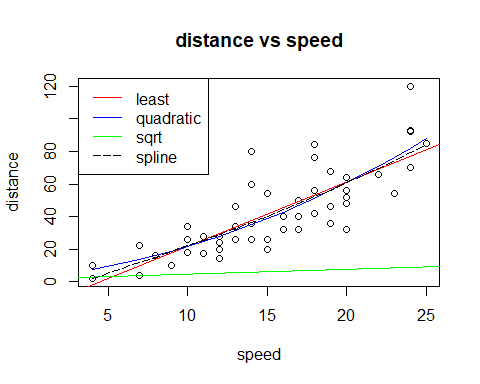
##   
## Call:  
## lm(formula = I(sqrt(dist)) ~ speed, data = cars)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.0684 -0.6983 -0.1799 0.5909 3.1534   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.27705 0.48444 2.636 0.0113 \*   
## speed 0.32241 0.02978 10.825 1.77e-14 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1.102 on 48 degrees of freedom  
## Multiple R-squared: 0.7094, Adjusted R-squared: 0.7034   
## F-statistic: 117.2 on 1 and 48 DF, p-value: 1.773e-14

abline(mod8, col = "red")  
lines(cars$speed, mod8q$fitted.values, col= "blue")  
abline(mod8s, col = "green")  
legend("topleft", legend = c("least", "quadratic", "sqrt"),col=c("red", "blue", "green"), lty=c(1,1, 1))



#### (e)

#e  
  
plot(cars$speed, cars$dist, xlab = "speed", ylab = "distance", main = "distance vs speed")  
abline(mod8, col = "red")  
lines(cars$speed, mod8q$fitted.values, col= "blue")  
abline(mod8s, col = "green")  
lines(smooth.spline(cars$speed, cars$dist), col = "black" , lty = 5)  
legend("topleft", legend = c("least", "quadratic", "sqrt", "spline"),col=c("red", "blue", "green", "black"), lty=c(1,1, 1, 5))



The spline appears to be a good fit. I think the spline and the quadratic are great fits to the data. Then followed by the least square method without any transformation(redline). From the plot, the sqrt of the response appears not to be good.