MiniProj5

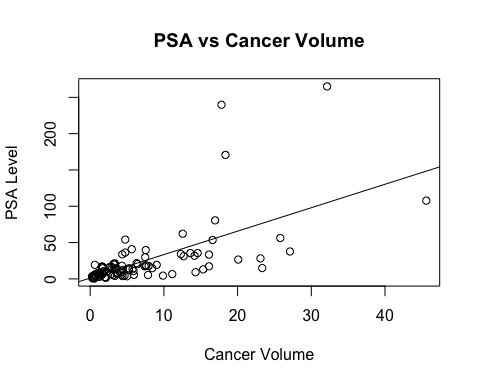
Prashant Prakash

November 29, 2015

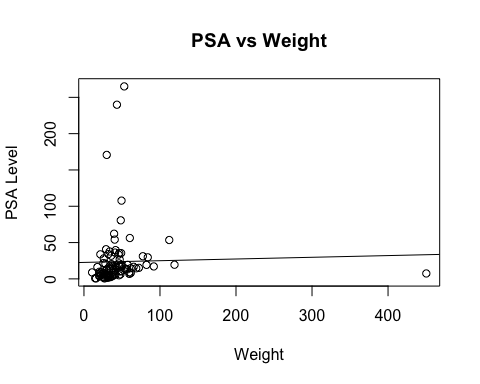
This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

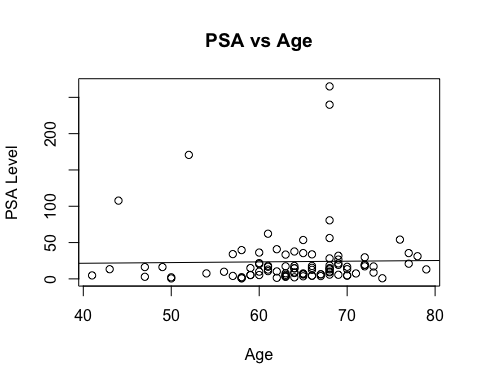
# read data from csv file   
data = read.csv("/Users/Prashant/Downloads/prostate\_cancer.csv",header = TRUE , sep=",")  
  
# Plot scatter plot of PSA level with other variables   
  
# with Cancer volume   
plot(data$cancervol, data$psa, main= "PSA vs Cancer Volume", xlab="Cancer Volume", ylab="PSA Level")  
abline(lm(data$psa~data$cancervol))



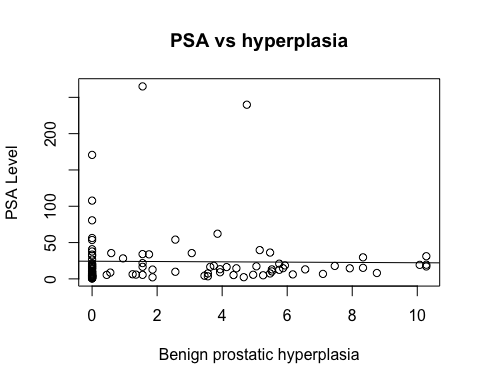
# with weight   
plot(data$weight, data$psa, main= "PSA vs Weight" , xlab="Weight", ylab="PSA Level")  
abline(lm(data$psa~data$weight))



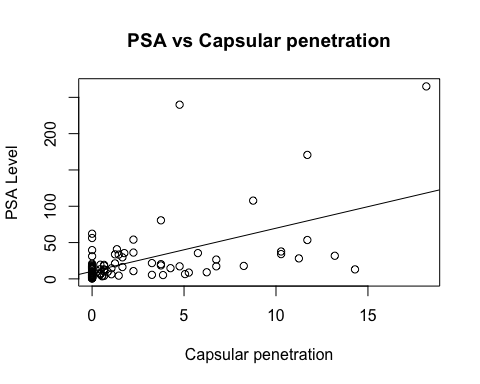
# with Age  
plot(data$age, data$psa, main = "PSA vs Age" , xlab="Age", ylab="PSA Level")  
abline(lm(data$psa~data$age))



# with Benign prostatic hyperplasia  
plot(data$benpros, data$psa, main = "PSA vs hyperplasia", xlab="Benign prostatic hyperplasia", ylab="PSA Level")  
abline(lm(data$psa~data$benpros))



# with Capsular penetration  
plot(data$capspen, data$psa, main="PSA vs Capsular penetration" ,xlab="Capsular penetration", ylab="PSA Level")  
abline(lm(data$psa~data$capspen))



# capturing correaltion coefficient to decide which varible is good for deciding PSA level   
  
cor(data$psa,data$cancervol ,use ="everything", method =c("pearson"))

## [1] 0.6241506

cor(data$psa,data$weight ,use ="everything", method =c("pearson"))

## [1] 0.02621343

cor(data$psa,data$age ,use ="everything", method =c("pearson"))

## [1] 0.01719938

cor(data$psa,data$benpros ,use ="everything", method =c("pearson"))

## [1] -0.01648649

cor(data$psa,data$capspen ,use ="everything", method =c("pearson"))

## [1] 0.5507925

# Also we can check R Square to decide on which can be good for deciding PSA level   
  
psa.reg.canvervol <- lm(data$psa ~ data$cancervol , data = data)  
summary(psa.reg.canvervol)

##   
## Call:  
## lm(formula = data$psa ~ data$cancervol, data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -61.619 -9.023 -1.586 3.151 181.183   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.1249 4.3596 0.258 0.797   
## data$cancervol 3.2299 0.4148 7.786 8.47e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 32.03 on 95 degrees of freedom  
## Multiple R-squared: 0.3896, Adjusted R-squared: 0.3831   
## F-statistic: 60.63 on 1 and 95 DF, p-value: 8.468e-12

psa.reg.weight <- lm(data$psa ~ data$weight , data = data)  
summary(psa.reg.weight)

##   
## Call:  
## lm(formula = data$psa ~ data$weight, data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -25.737 -17.879 -10.167 -2.416 241.167   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 22.66607 5.88626 3.851 0.000214 \*\*\*  
## data$weight 0.02339 0.09152 0.256 0.798824   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 40.98 on 95 degrees of freedom  
## Multiple R-squared: 0.0006871, Adjusted R-squared: -0.009832   
## F-statistic: 0.06532 on 1 and 95 DF, p-value: 0.7988

psa.reg.age <- lm(data$psa ~ data$age , data = data)  
summary(psa.reg.age)

##   
## Call:  
## lm(formula = data$psa ~ data$age, data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -23.833 -17.853 -10.653 -2.886 240.952   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 17.71301 36.12862 0.490 0.625  
## data$age 0.09421 0.56193 0.168 0.867  
##   
## Residual standard error: 40.99 on 95 degrees of freedom  
## Multiple R-squared: 0.0002958, Adjusted R-squared: -0.01023   
## F-statistic: 0.02811 on 1 and 95 DF, p-value: 0.8672

psa.reg.benpros <- lm(data$psa ~ data$benpros , data = data)  
summary(psa.reg.benpros)

##   
## Call:  
## lm(formula = data$psa ~ data$benpros, data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -23.64 -18.12 -10.00 -2.19 241.12   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 24.2924 5.4371 4.468 2.18e-05 \*\*\*  
## data$benpros -0.2218 1.3802 -0.161 0.873   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 40.99 on 95 degrees of freedom  
## Multiple R-squared: 0.0002718, Adjusted R-squared: -0.01025   
## F-statistic: 0.02583 on 1 and 95 DF, p-value: 0.8727

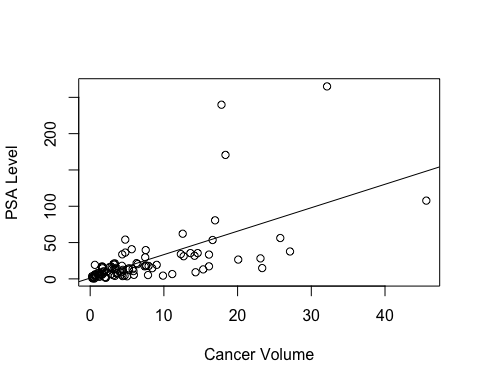
psa.reg.capspen <- lm(data$psa ~ data$capspen , data = data)  
summary(psa.reg.capspen)

##   
## Call:  
## lm(formula = data$psa ~ data$capspen, data = data)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -82.215 -9.547 -4.758 5.037 201.194   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 10.3986 4.0456 2.570 0.0117 \*   
## data$capspen 5.9373 0.9231 6.432 5.06e-09 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 34.22 on 95 degrees of freedom  
## Multiple R-squared: 0.3034, Adjusted R-squared: 0.296   
## F-statistic: 41.37 on 1 and 95 DF, p-value: 5.055e-09

# clearly using correleation it's obvious Cancer volume is the candidate to do a better regression for PSA level   
  
x <- data$cancervol  
y <- data$psa  
  
(psa.reg <- lm(y ~ x))

##   
## Call:  
## lm(formula = y ~ x)  
##   
## Coefficients:  
## (Intercept) x   
## 1.125 3.230

plot(data$cancervol, data$psa, xlab="Cancer Volume", ylab="PSA Level")  
abline(psa.reg)



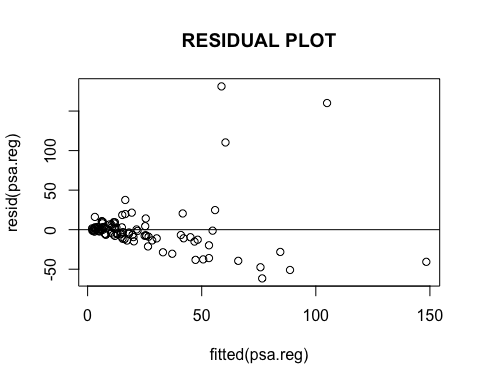
# after plot point out three outliers in the plot   
  
  
  
# Regression Diagnositics on Cancer Volume vs PSA  
  
confint(psa.reg)

## 2.5 % 97.5 %  
## (Intercept) -7.530000 9.779701  
## x 2.406407 4.053461

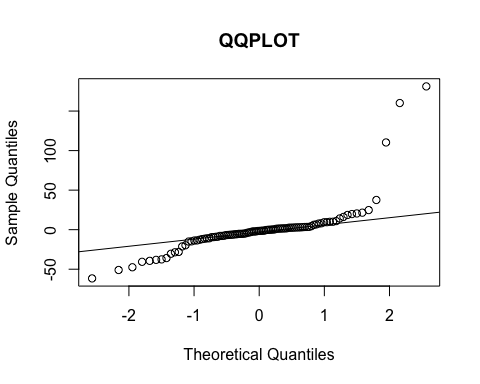
summary(psa.reg)

##   
## Call:  
## lm(formula = y ~ x)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -61.619 -9.023 -1.586 3.151 181.183   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.1249 4.3596 0.258 0.797   
## x 3.2299 0.4148 7.786 8.47e-12 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 32.03 on 95 degrees of freedom  
## Multiple R-squared: 0.3896, Adjusted R-squared: 0.3831   
## F-statistic: 60.63 on 1 and 95 DF, p-value: 8.468e-12

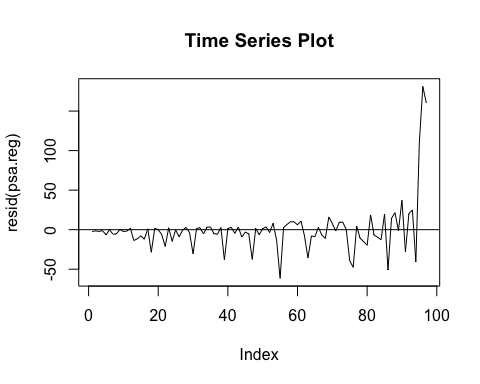
# Check for errors for zero mean and constant variance , normality and independence of residual error   
  
plot(fitted(psa.reg),resid(psa.reg), main="RESIDUAL PLOT")  
abline(h=0)



qqnorm(resid(psa.reg),main="QQPLOT")  
qqline(resid(psa.reg))



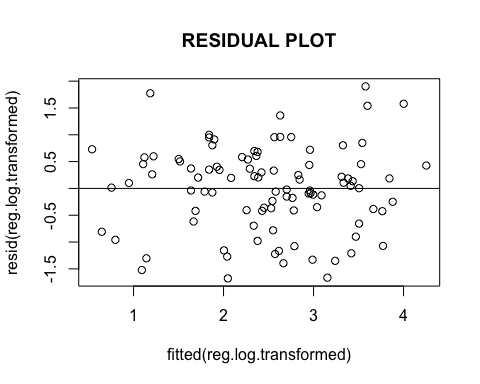
plot(resid(psa.reg), type="l",main="Time Series Plot")  
abline(h=0)



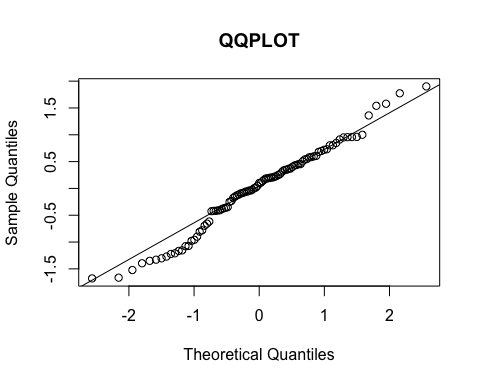
# Do Log transform to provide a better fit   
  
psa.log <- log(data$psa)  
cancervol.log <- log(data$cancervol)  
reg.log.transformed <- lm(psa.log ~ cancervol.log)  
  
summary(reg.log.transformed)

##   
## Call:  
## lm(formula = psa.log ~ cancervol.log)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -1.6778 -0.4187 0.1012 0.5035 1.9022   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.50923 0.12198 12.37 <2e-16 \*\*\*  
## cancervol.log 0.71827 0.06822 10.53 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.7879 on 95 degrees of freedom  
## Multiple R-squared: 0.5385, Adjusted R-squared: 0.5336   
## F-statistic: 110.8 on 1 and 95 DF, p-value: < 2.2e-16

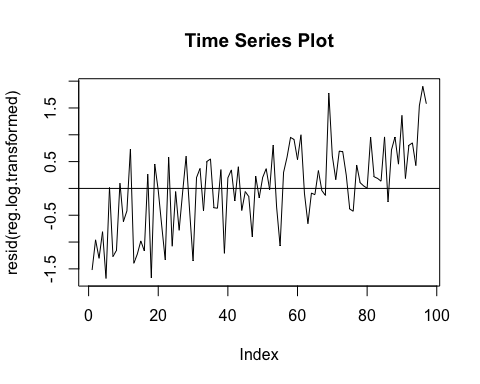
# Check for errors for zero mean and constant variance , normality and independence of residual error   
  
plot(fitted(reg.log.transformed),resid(reg.log.transformed), main="RESIDUAL PLOT")  
abline(h=0)



qqnorm(resid(reg.log.transformed),main="QQPLOT")  
qqline(resid(reg.log.transformed))



plot(resid(reg.log.transformed), type="l",main="Time Series Plot")  
abline(h=0)



# prediction of PSA for mean of cancer volume  
  
x.new <- data.frame(x = median(data$cancervol))  
predict(psa.reg, newdata=x.new)

## 1   
## 14.89438

# prediction of PSA for mean of log transformed cancer volume   
log.predict<- exp(1)^predict(reg.log.transformed,data.frame(cancervol.log=log(median(data$cancervol))))

You can also embed plots, for example:

Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.