* **Summary statistics**

Example 1: #Given a matrix or data.frame x, t() returns the transpose of x.

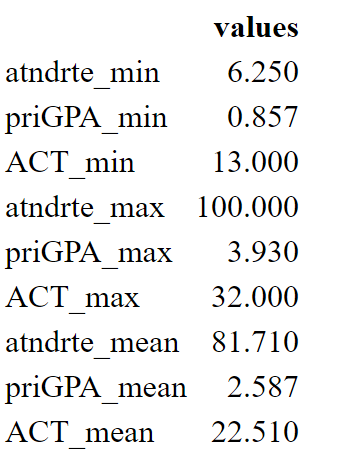
attend %>%

**select**(atndrte,priGPA,ACT)%>%

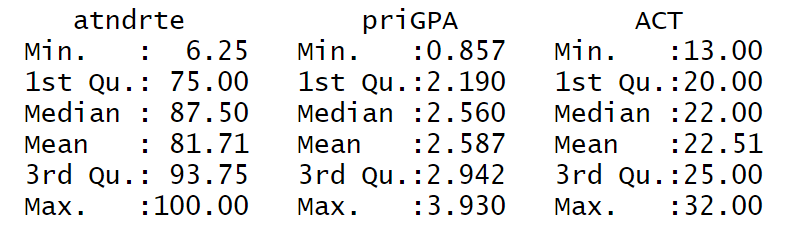
**summarise\_all**(list(~min(.),~max(.),~mean(.)),na.rm=TRUE)%>%

**t()**%>%

kable(,col.names = c("values"),digits=3)



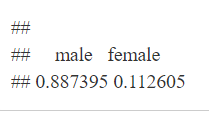
summary(attend[,c("atndrte","priGPA","ACT")])



* Show the **proportion** of people that reside in a standard metropolitan statistical area

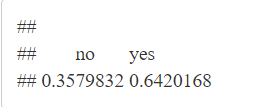
smsa <- factor(PSID1982$smsa,labels = c("no","yes"))

**prop.table**(table(smsa))



* Add a **cross table**

kable(prop.table(table(PSID1982$gender,PSID1982$smsa)),digits = 3)



htv %>% select(educ, motheduc,fatheduc) %>%

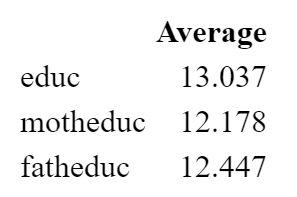
summarise\_all(mean) %>%

t()%>%

data.frame() %>%

setNames(., c("Average"))%>%

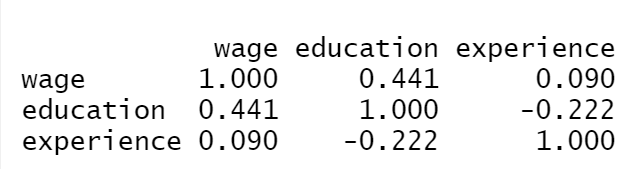
kable(digits=3)



* The following table looks at the **correlation** between wages, education and experience

cortabe<- **cor**(PSID1982[,c("wage","education","experience")])

round(cortabe, 3)



Or use

cormat<-**select**(gpa1, colGPA, hsGPA, ACT, x, z ) %>% **cor**()

round(cormat, 3),

where gpa1 is the dataframe

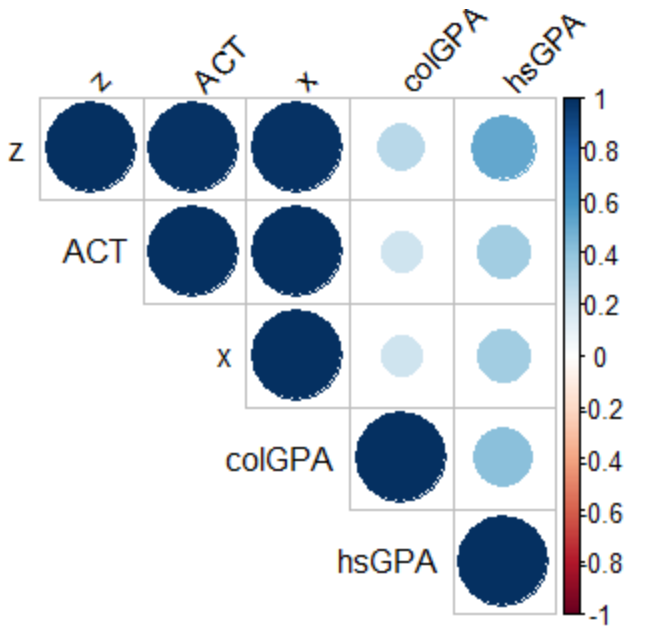
* Other correlation graphs

#library(corrplot)

#library(PerformanceAnalytics)

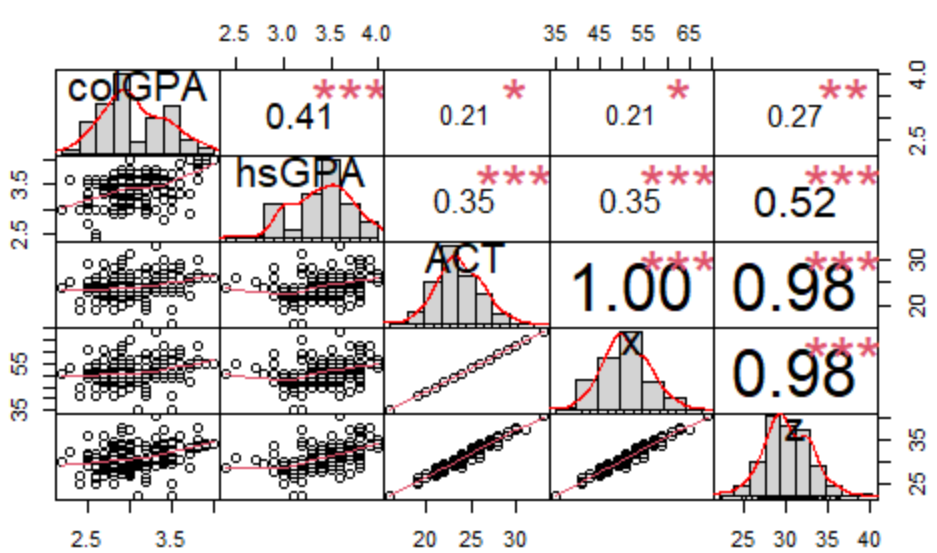
cormat<-**select**(gpa1, colGPA, hsGPA, ACT, x, z ) %>% **cor**()

**corrplot**(cormat, type = "upper", order = "hclust",tl.col = "black", tl.srt = 45)



dat <- select(gpa1, colGPA, hsGPA, ACT, x, z ) #gpa1 is the dataframe

**chart.Correlation**(dat, histogram = TRUE, pch = 19)



* Regression on a subset of the dataset

mfr <- lm(atndrte ~ priGPA+ACT, **subset**(attend,frosh==1))

* **Plot regression line**

ggplot(PSID1982 , aes(x=education, y=wage))+

geom\_point(color="red", alpha=.5 , position=position\_jitter(w=0.2))+

**geom\_smooth(method='lm')**+

ggtitle('Wage vs Educ') +

xlab('Education') +

ylab('Wage')

* **Plot regression line, grouped by gender**

ggplot(PSID1982 , aes(x=education,y=wage))+

geom\_point(color="red", alpha=.5)+

geom\_smooth( )+

**facet\_grid(~gender**, scales="free")+ ggtitle('Wage vs Educ') +

xlab('Education') +

ylab('Wage')

* Plot data points and a regression line of your choice

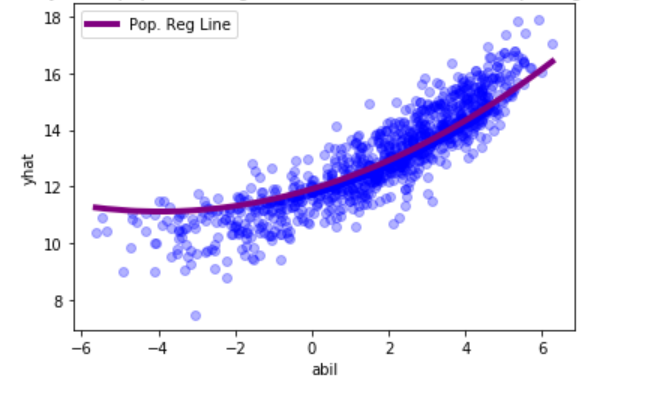
dat<-data.frame(educhat=predict(model3), abil=htv$abil)

plot<-ggplot(dat,aes(abil,educhat))+

geom\_point(col=4, alpha=.5)+

**stat\_function**(fun=function(abil){coef(model3)[1]+ coef(model3)[2]\*12.18 + coef(model3)[3]\*12.45+ coef(model3)[4]\*abil + coef(model3)[5]\*(abil^2)}, geom="line", col=2)+

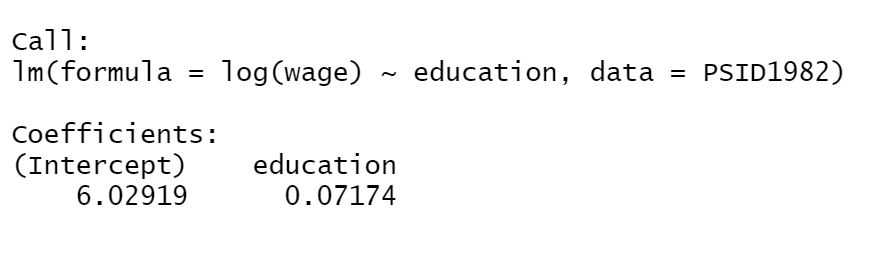
ggtitle(paste("Predicted Education and Ability"))



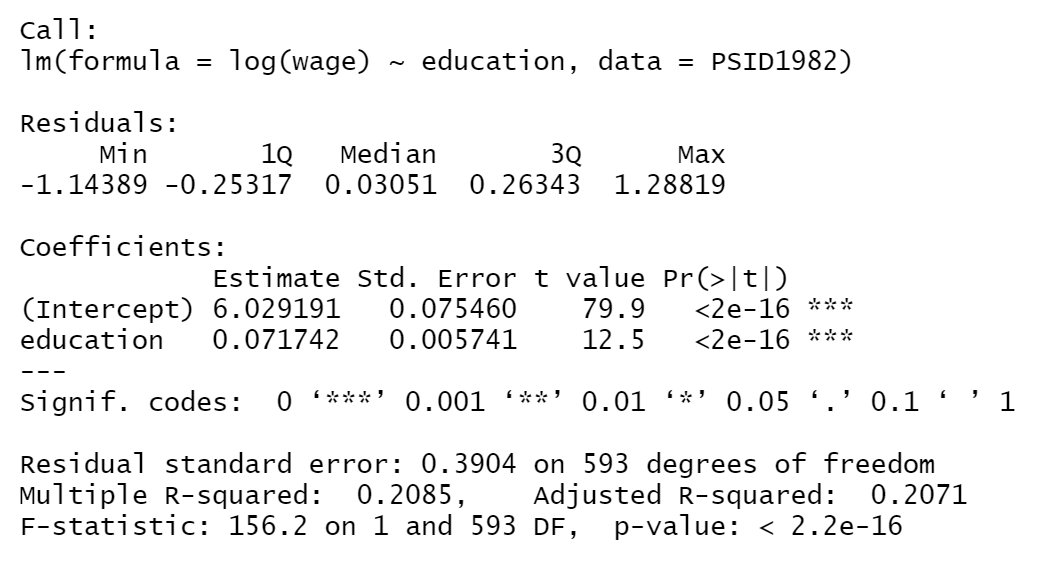
* Use the **lm()** command to estimate

#Estimate using log(wage) = B0 +B1educ + u

(regre<-***lm(formula = log(wage)~education, data=PSID1982)***)



summary(regre)



* Using Function **minimization** to calculate coefficients

# Optimizing the function SSR to find the parameters that minimize it. Call the object result. The vector par=c(b0,b1) will be replaced but the correct values of b0 and b1 that minimize your function

#*Define a function*

min.SSR <- function(data,par){sum((y-par[1]-(par[2]\*x))^2)}

# *Retrieves the vector of parameters from the results object*, and rounds them to the 3 decimals

result<-optim(par=c(b0,b1) , fn=min.SSR, data=PSID1982)

* The **lm()** command and its useful **features**

mydata<-data.frame(x,y)

fit <- **lm**(formula = y ~ x , data=mydata)

summary(fit) # standard regression output

* Useful functions

coef(fit) # (or coefficients()) extracting the *regression coefficients*

confint(fit, level=0.95) # *Confidence Intervals* for model parameters

fitted(fit) # *predicted values* estimated y

residuals(fit) # (or resid()) *extracting residuals*

predict(fit) # *predictions* for new data

deviance(fit) # *residual sum of squares*

anova(fit) # *anova table*

vcov(fit) # *covariance matrix for model parameters*

influence(fit) # regression diagnostics

nobs(fit) # *number of observations*

# estimate the model and save it into an object m1

m1<-lm(wage~educ, wage1)

m1$coeff # coefficients

mean(**m1$fitted.values**) # average of fitted values

m2<-lm(log(wage)~educ, wage1)

# **using the function to extract the coefficients create b\_0 from m2**

(b\_0=**coef**(m2)[1])

(b\_0\_alt<-summary(m2)$coefficients[1,1])

Output:

1614437964(1)

# **calculate the values of y\_hat** and the residuals **u\_hat**

*y\_hat*<-**fitted**(m2)

*u\_hat*<-**resid**(m2)

#Show how sum(u\_hat) =0.

(sum\_u\_hat=round(sum(u\_hat),3))

#Show how cov(x, u\_hat)~=0

(covx\_u\_hat=round((cov(wage1$educ, u\_hat)),3))