```
# 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))
```