Compulsary 1

Tord Sture Stangeland

10/21/2019

Excersise 1:

Question 1:

Both month and day should be treated as categorical data. There are very few unique values of month, and day is just a division of the month. It is important to make these distinctions because certain measurements makes sense on numerical data and not on categorical data.

Question 2:

```
library("psych")
cor(completeData$Ozone, completeData$Solar.R)
corr.test(completeData[,1:4])
```

From running the correlation tests, it shows that solar radiation and wind do not necessarily have a correlation efficient different from 0.

Question 3:

```
windAndSolarLR <- lm(Solar.R ~ Wind)
ozoneAndTempLR <- lm(Ozone ~ Temp)
TempWindOnOzone <- lm(Ozone ~ Temp + Wind)
TempWindSolarROnOzone <- lm(Ozone ~ Temp + Wind + Solar.R)
ozoneOnTemp <- lm(Temp ~ Ozone)
windSolarOzoneOnTemp <- lm(Temp ~ Wind + Solar.R + Ozone)

summary(windAndSolarLR)
summary(ozoneAndTempLR)
summary(TempWindOnOzone)
summary(TempWindsolarROnOzone)
summary(ozoneOnTemp)
summary(windSolarOzoneOnTemp)
```

First SLR: Wind on solar radiation: In this linear regression we get an R squared of 0.01617563 which is really low and thus indicated that the variability explained by the regression is really low. Adjusted R squared does not provide any information as there is only one variable.

Second SLR: Temp on Ozone: In this linear regression there is a higher R^2 as compared to the previous as this models' R^2= 0.4879601. Although still not very high, it tells us that about 49% of the variability in the Ozone can be explained by the model. Here too, adjusted R^2 does not apply.

Third Linear Regression: Multiple linear regression of temp and wind on ozone: For this regression there is a R^2 value of 0.5814 which is even better than temperature alone. However, as this is a multiple linear regression model the important statistic for variability is the adjusted R^2. The adj R^2 is 0.5736 which is just over a percentage worse than unadjusted. The adj R^2 is higher than the R^2 of temp only SLR on Ozone and both Temp and Wind have significant coefficients and so this is a better model for predicting ozone.

Fourth Linear Regression: Multiple linear regression of temp, wind and solar radiation on Ozone: For This regression there is an adj R^2 of 0.5948, which means that nearly 60% of the variability in Ozone can be explained by the three independent variables. The model has a very low p value and so does all of the coefficients. This means that all the independent variables are significantly impacting the model which is also reflected in the small difference between R^2 and adj R^2. The R^2 is 0.6059.

Fifth Linear regression: SLR of ozone on Temp: This model is essentially the same as Temp on ozone. The R^2 is 0.488 and the adj R^2 is not important. I included this model to aid the sixth linear regression which follows.

Sixth Linear Regression: Multiple linear regression of wind, solar radiation and ozone on temperature: Here the R^2 is 0.4999 and the adjusted R^2 is 0.4858 which means that the model explains about 49 % of the variability in temperature. The two variables Wind and Solar Radiation have a high probability which means they aren't significant in this model. If we compare the Adj R^2 of this model to the R^2 of the fifth linear regression above (R^2 0.488) we can see that they are nearly the same (this one is slightly worse) which explains why the p-value of wind and solar radiation is so high.

Question 4:

```
library("lmtest")
library("car")
layout(matrix(c(1,2,3,4),2,2))

# Testing the assumptions of linear regression: Model 1
plot(windAndSolarLR)
```

```
plot(Wind, Solar.R, main = "SLR of wind on solar radiation", col.line = "red")
abline(windAndSolarLR, col="red")
bptest(Solar.R ~ Wind)
dwtest(windAndSolarLR)
shapiro.test(windAndSolarLR)
vif(windAndSolarLR)

# Model 2:
plot(ozoneAndTempLR)
plot(Temp, Ozone, main = "SLR of temp on onzone", col.line = "red")
abline(ozoneAndTempLR, col="red")
bptest(ozoneAndTempLR)
dwtest(ozoneAndTempLR)
shapiro.test(ozoneAndTempLR)

# Model 3:
plot(TempWindOnOzone)
bptest(TempWindOnOzone)
dwtest(TempWindOnOzone)
vif(TempWindOnOzone)

# Model 4:
plot(TempWindSolarROnOzone)
bptest(TempWindSolarROnOzone)
dwtest(TempWindSolarROnOzone)
vif(TempWindSolarROnOzone)

# Model 5:
plot(Ozone, Temp)
abline(ozoneOnTemp, col="red")
plot(ozoneOnTemp)
bptest(ozoneOnTemp)
dwtest(ozoneOnTemp)
vif(ozoneOnTemp)

# Model 6:
plot(windSolarOzoneOnTemp)
```

```
bptest(windSolarOzoneOnTemp)
dwtest(windSolarOzoneOnTemp)
vif(windSolarOzoneOnTemp)
```

Assumptions: Linearity, constant variance, independent error terms, normality of error and no multicollinearity.

Model 1: SLR of Wind on Solar radiation. - Linearity: o From the graph and it does not appear to have a linear relationship. - Constant variance: o Through a bp test p-value of 0.001478 I reject the null hypothesis and conclude that there is heteroskedasticity. - Independent error terms: o Through a Durbin-Watson test p-value of 0.3143 I conclude that the null hypothesis is true and there is autocorrelation. The DW = 1.9133 so the autocorrelation appears to be positive. - Normality of Error: o By visually inspecting a Q-Q plot I determine that the errors may not be normally distributed sufficiently. - No multicollinearity: o Does not really apply here as there is only one independent variable.

Model 2: SLR of Temp on Ozone. - Linearity: o Does not follow a linear relationship. - Constant variance: o By using a bp test I conclude that there is homoskedasticity. The BP resulted in a p-value of 0.2193 - Independent error terms: o DW test indicated that there is autocorrelation with a p-value of 0.2123 and DW = 1.8644 so the autocorrelation is positive. - Normality of error: o The Q-Q plot indicates that the errors are not normally distributed. - No multicollinearity: o Does not apply as there is only one independent variable.

Model 3: MLR of temp and wind on ozone. - Linearity: o Does not follow a linear relationship - Constant variance: o BP test p-value: 0.06147. As I use cutoff off 5% I conclude that there is homoskedasticity. - Independent error terms: o DW p-value: 0.3106 and DW = 1.9214. Therefore there is positive autocorrelation. - Normality of error: o The errors are nearly normally distributed, however near the extremes they deviate quite a bit so I conclude they are not normally distributed. - Multicollinearity: o By using Variance Inflation Factor I got a value of 1.32837 for both variables which indicates that they are
moderately correlated.

Model 4: MLR of temp wind and solar radiation on Ozone. - Linearity: o Does not follow a linear relationship. - Constant variance: o Bp test p-value: 0.1678, so there is homoskedasticity. - Independent error terms: o DW test p-value: 0.3347 and DW = 1.933 so there does appear to be some positive autocorrelation. - Normality of error: o By visually inspecting the QQ plot there seems to not be a normal distribution of errors. - Multicollinearity: o Using VIF I got the values for Temp, Wind and Solar.R of 1.43, 1.33 and 1.1 respectively. There, therefore, appears to be some moderate multicollinearity.

Model 5: SLR of Ozone on temp. - Linearity: o There does not appear to be a linear relationship. Seems more exponential. - Constant variance: o BP test p-value: 0.0637, so there is homoskedasticity. - Independent error terms: o DW test p-value: 5.261e-06 and a DW =1.18 so there appears to be some positive autocorrelation. - Normality of error: o By viewing the QQ plot there does not appear to be normally distributed errors. - Multicollinearity: o Not applicable.

Model 6: MLR of wind, solar radiation and ozone on temp. - Linearity: o There does not appear to be a linear relationship. - Constant variance: o BP test p-value = 0.05501. So I keep my null hypothesis and conclude there is homoskedasticity. - Independent error terms: o DW test p-value =

4.176e-06, DW = 1.17. So there is positive autocorrelation. - Normality of error: o Appears to be somewhat normal from the QQ test, however some deviations near the extremes. - Multicollinearity: o All variables have VIF values between 1 and 2 so there is moderate multicollinearity.

Question 5: NOTE: I did not keep all the code of all the changes I made throughout this section. It was simpler to test things out by chaning single lines rather than to reproduce an entierly new model every time I made a change.

```
model2fix <- lm(Ozone ~ Temp + I(Temp^2))
summary(model2fix)
layout(matrix(c(1,2,3,4),2,2))

plot(model2fix)

model2fix2 <- lm(Ozone ~ + I(Temp^2) + I(log(Wind)))
summary(model2fix2)
bptest(model2fix2)
dwtest(model2fix2)
vif(model2fix2)

model5fix <- lm(I(log(Temp)) ~ I(Ozone^1/4))
layout(matrix(c(1,2,3,4),2,2))
plot(model5fix)
summary(model5fix)
bptest(model5fix)
dwtest(model5fix)
vif(model5fix)
```

Model 1: This model just seems terrible in general.

Model 2: There appears to be an exponential relationship. I tried to add a squared term too to some success. Now the model looks something like this: Ozone = Temp + Temp^2. This solved the linearity issue and increased the R^2 from 0.488 in the first case to adj R^2 of 0.5342 in the second. As I kept trying out new things and testing them I decided to remove the Temp term completely due to the autocorrelation. As I kept trying things out I reproduced a lot of the other models triying to predict ozone (model 2,3,4), with different terms being squared and logged and by solving one problem I created a different one. If I tried to reduce autocorrelation I would increase autocorrelation and if I tried to reduce them both I would cause heteroskedasticity etc.

Model 6: Trying out strategies with model 6 I encountered similar problems as before. I with Temp = Ozone^1/4 with an R^2 of 0.4833, which was the best I was able to do with significant coefficients. This is still a homoscedastic model with little autocorrelation. The lacking normality in errors are probably due to lack of data as changing the dependent with log did not improve the model. There is till a linearity issue though.

Question 6:

```
library("glmnet")

set.seed(21)
x_names = c("Solar.R", "Wind", "Temp")
x_var <- as.matrix(completeData[,])
y_var <- as.matrix(completeData[, "Ozone"])
ss <- floor(0.75 * nrow(x_var))
ti <- sample(seq_len(nrow(x_var)), size=ss)

x_train <- completeData[ti,]
x_test <- completeData[-ti,]

lambda_seq <- 10^seq(2.0, -2, by = -.1)

ridge <- glmnet(as.matrix(x_train[, x_names]), x_train$Ozone, alpha =0, lambda=lambda_seq)
ridge_cv <- cv.glmnet(as.matrix(x_train[, x_names]), x_train$Ozone, alpha = 0, lambda = lambda_seq)

best_lambda <- ridge_cv$lambda.min
best_lambda


best_fit <- ridge_cv$glmnet.fit
head(best_fit)

summary(ridge)

best_ridge <- glmnet(as.matrix(x_train[, x_names]), x_train$Ozone, alpha = 0, lambda = best_lambda)

predictions <- predict(best_ridge, s = best_lambda, newx= as.matrix(x_test[, x_names]))

rss <- sum((x_test$Ozone - predictions)^2)
```

```
tss <- sum((x_test$Ozone - mean(x_test$Ozone))^2)

Rsquared <- 1.0 - rss/tss
Rsquared
meanSquaredError <- sum((x_test$Ozone - predictions)^2)/nrow(predictions)
meanSquaredError
adjRsquared <- 1 - ((1-Rsquared)*(nrow(predictions) - 1)/(nrow(predictions) - length(x_names) - 1))
adjRsquared
```

When using a ridge regression when trying to predict the Ozone using temp, solar radiation and wind I got the model (keeping seed constant): Ozone = -24.49 + 0.05Solar.R – 3.94Wind + 1.25Temp (approximately). I had an R^2 of 0.5698 and an adj R^2 of 0.5160854. I also had a mean squared error of: 307.4756.

Question 7:

```
library("glmnet")

set.seed(21)
x_names = c("Solar.R", "Wind", "Temp")
x_var <- as.matrix(completeData[,])
y_var <- as.matrix(completeData[, "Ozone"])
ss <- floor(0.75 * nrow(x_var))
ti <- sample(seq_len(nrow(x_var)), size=ss)

x_train <- completeData[ti,]
x_test <- completeData[-ti,]

lambda_seq_lasso <- 10^seq(2.0, -2, by = -.1)

lasso <- glmnet(as.matrix(x_train[, x_names]), x_train$Ozone, alpha =1, lambda=lambda_seq_lasso)
lasso_cv <- cv.glmnet(as.matrix(x_train[, x_names]), x_train$Ozone, alpha = 1, lambda = lambda_seq_lasso)

best_lambda_lasso <- lasso_cv$lambda.min
best_lambda_lasso
```

```
best_fit_lasso <- lasso_cv$glmnet.fit
head(best_fit_lasso)

summary(lasso)

best_lasso <- glmnet(as.matrix(x_train[, x_names]), x_train$Ozone, alpha = 1, lambda = best_lambda_lasso)
coef(best_lasso)

predictions_lasso <- predict(best_lasso, s = best_lambda_lasso, newx= as.matrix(x_test[, x_names]))

rss_lasso <- sum((x_test$Ozone - predictions_lasso)^2)
tss_lasso <- sum((x_test$Ozone - mean(x_test$Ozone))^2)

Rsquared_lasso <- 1.0 - rss_lasso/tss_lasso
Rsquared_lasso
meanSquaredError_lasso <- sum((x_test$Ozone - predictions_lasso)^2)/nrow(predictions)
meanSquaredError_lasso
adjRsquared_lasso <- 1 - ((1-Rsquared_lasso)*(nrow(predictions_lasso) - 1)/(nrow(predictions_lasso) - length(x_na
mes) - 1))
adjRsquared_lasso
```

When using a lasso regression to predict Ozone using temp, solar radiation and wind I got the model (keeping seed constant): Ozone = -26.14 + 0.051solarR - 4.31 wind + 1.32 temp (approx.). I also got an R^2 of 0.55 and and adjusted R^2 of .49 and a mean squared error of 320. This dataset does not reflect the benefits of the lasso regression as all the variables in the model are significant. The lasso regression has a tendency to make the coefficients with the largest impact on the model more prominent while reducing the impact of already less significant variables (in terms of magnitude). Since all the variables in this model play an important role in the models predictive ability, using a lasso here reduces the efficiency of the model.

Excersise 2:

Question 1:

```
library("naniar")
```

```
setwd("D:/Documents/Universitetet i Bergen/STATLEARN/Compulsary 1")
# Loading CSV
diadat = read.csv("Diadat.csv")

colNames = c("pregnant","glucose", "pressure", "triceps", "insulin", "mass", "pedigree", "age")

neg <- diadat[diadat[,"diabetes"]=="neg",]
pos <- diadat[diadat[,"diabetes"]=="pos",]

for(i in colNames){
  if (i != "pregnant" || i!="age") {
  diadat[diadat[,"diabetes"]=="pos",][pos[,i] == 0, i] <- mean(pos[,i] != 0 )
  diadat[diadat[,"diabetes"]=="neg",][neg[,i] == 0, i] <- mean(neg[,i] != 0 )
  }
}
diadat$diabetes <- ifelse(diadat$diabetes == "pos", 1, 0)
```

Question 2:

```
seed(21)
ss <- floor(0.75 * nrow(diadat))
ti <- sample(seq_len(nrow(diadat)), size=ss)

train <- diadat[ti,]
test <- diadat[-ti,]
```

Question 3:

```
library("mtcars")
logistic <- glm(train$diabetes ~ ., family="binomial"(link="logit"), data=train)
summary(logistic)

p <- predict(logistic, newdata = subset(test, select = colNames), type="response",family="binomial"(link="logi
t"))
coef(logistic)
```

```
mse <- (sum((p - test$diabetes)^2))/length(test$diabetes)
mse
```

MSE 0.17

Question 4:

```
library("MASS")
lindiscan <- lda(train$diabetes ~ ., data=train)
summary(lindiscan)

p <- predict(lindiscan, test[,colNames])
chanceOfDia <- p$posterior[,2]
chanceOfDia

mse <- (sum((test$diabetes - chanceOfDia)^2))/length(test$diabetes)
mse
```

MSE = 0.1633386

Question 5:

```
library("MASS")

quadiscan <- qda(train$diabetes ~ ., data=train)
summary(quadiscan)

p <- predict(quadiscan, test[,colNames])
chanceOfDia <- p$posterior[,2]
chanceOfDia

mse <- (sum((test$diabetes - chanceOfDia)^2))/length(test$diabetes)
mse
```

MSE = 0.1731801

Question 6:

```
library("class")
knn28 <- knn(train=train[,colNames], test=test[,colNames], cl=train$diabetes, prob = TRUE,k=27)
summary(knn28)
summary(attr(knn28, "prob"))

prob <-attr(knn28, "prob")
p <- predict(knn28)

mse <- (sum((1-prob) - as.numeric(test$diabetes))^2)/length(test$diabetes)
mse
```

MSE = 0.1969379