STA5075: Practical 16

Jessica Stow (STWJES003@myuct.ac.za)

2025-02-13

Part 1: Simple linear regression

a)

Download and import the the "tobacco.txt"

```
tobacco <- read.delim("tobacco.txt")
summary(tobacco)
```

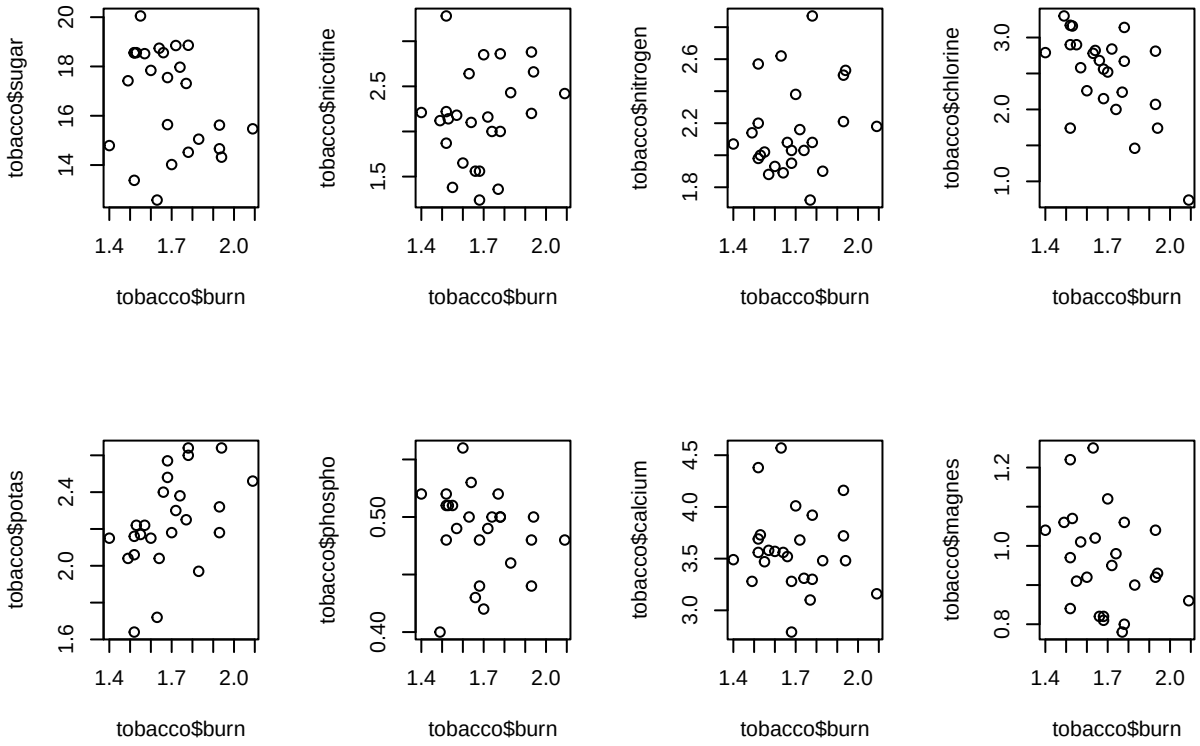
```
##       burn           sugar          nicotine        nitrogen
##  Min.   :1.400   Min.   :12.58   Min.   :1.240   Min.   :1.720
##  1st Qu.:1.550   1st Qu.:14.79   1st Qu.:1.870   1st Qu.:1.980
##  Median :1.680   Median :17.42   Median :2.160   Median :2.080
##  Mean   :1.688   Mean   :16.70   Mean   :2.159   Mean   :2.157
##  3rd Qu.:1.780   3rd Qu.:18.56   3rd Qu.:2.430   3rd Qu.:2.210
##  Max.   :2.090   Max.   :20.05   Max.   :3.280   Max.   :2.870
##     chlorine         potas          phospho          calcium
##  Min.   :0.740   Min.   :1.640   Min.   :0.4000   Min.   :2.790
##  1st Qu.:2.150   1st Qu.:2.150   1st Qu.:0.4800   1st Qu.:3.310
##  Median :2.670   Median :2.220   Median :0.5000   Median :3.560
##  Mean   :2.481   Mean   :2.238   Mean   :0.4868   Mean   :3.592
##  3rd Qu.:2.840   3rd Qu.:2.400   3rd Qu.:0.5100   3rd Qu.:3.720
##  Max.   :3.300   Max.   :2.640   Max.   :0.5600   Max.   :4.570
##      magnes
##  Min.   :0.780
##  1st Qu.:0.860
##  Median :0.950
##  Mean   :0.964
##  3rd Qu.:1.040
##  Max.   :1.250
```

The response variable burn measures the rate of cigarette burn in inches per 1000 seconds while the explanatory variables are percentages of total nitrogen, chlorine, potassium, phosphorous, calcium and magnesium. It is assumed that the true relationship between the burn rate and the explanatory variables is linear.

Explore the data, in particular the relationships between the response and each of the explanatory variables. Show some evidence of your exploratory data analysis and display the relationships on a single plot.

```
par(mfrow = c(2,4))

## Relationship between burn and sugar
```

```r
plot(tobacco$burn, tobacco$sugar)

## Relationship between burn and nicotine
plot(tobacco$burn, tobacco$nicotine)

## Relationship between burn and nitrogen
plot(tobacco$burn, tobacco$nitrogen)

## Relationship between burn and chlorine
plot(tobacco$burn, tobacco$chlorine)

## Relationship between burn and potas
plot(tobacco$burn, tobacco$potas)

## Relationship between burn and phospho
plot(tobacco$burn, tobacco$phospho)

## Relationship between burn and calcium
plot(tobacco$burn, tobacco$calcium)

## Relationship between burn and magnes
plot(tobacco$burn, tobacco$magnes)
```



It appears that the relationship between burn rate and chlorine is strong (and negative), The relationship between burn rate and potassium seems to also be strong (and positive). For the remaining variables, no obvious linear relationship with burn rate is evident.

Build a correlation matrix that displays the correlations between the response and each of the explanatory variables.

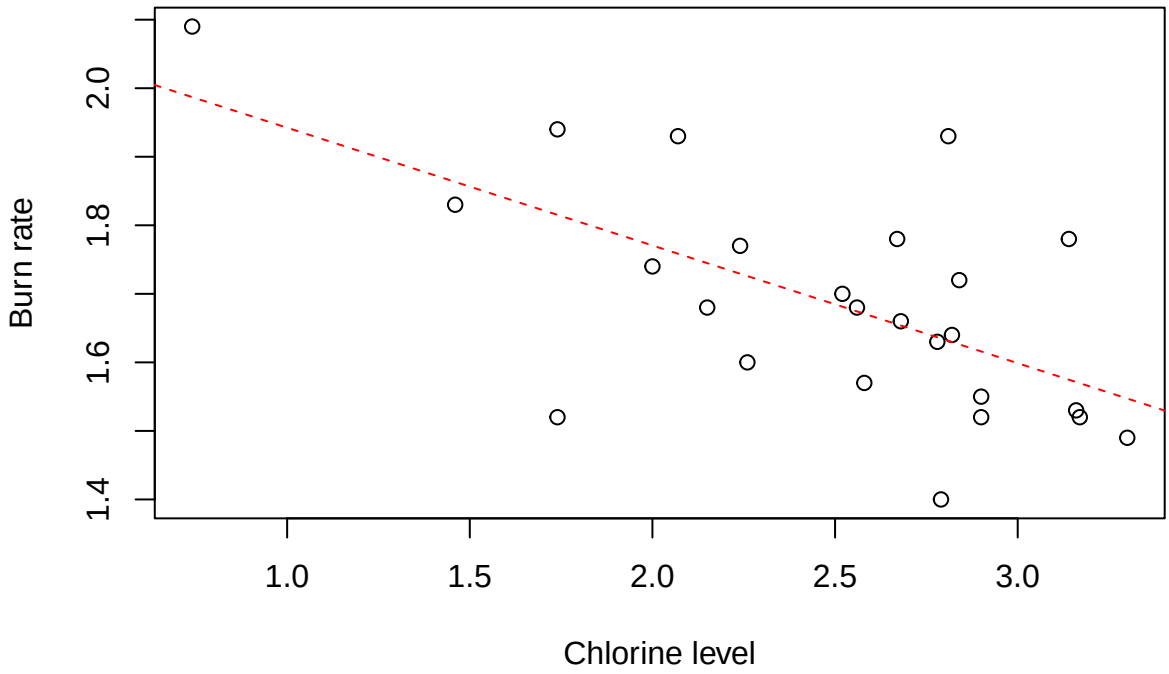
```
# Correlation matrix
cor(tobacco)
```

```
##                burn      sugar    nicotine     nitrogen    chlorine
## burn      1.0000000 -0.3049965  0.20876900  0.226000207 -0.62343918
## sugar    -0.3049965  1.0000000 -0.68411541 -0.688565251  0.44352982
## nicotine  0.2087690 -0.6841154  1.00000000  0.767802334 -0.27330343
## nitrogen  0.2260002 -0.6885653  0.76780233  1.000000000 -0.08936414
## chlorine -0.6234392  0.4435298 -0.27330343 -0.089364140  1.00000000
## potas     0.4865503  0.1947681 -0.29440378 -0.007483429 -0.09904219
## phospho  -0.1752969  0.1782754 -0.02680349 -0.093028344 -0.05580311
## calcium  -0.1212625 -0.4620981  0.68009707  0.627170771  0.13825072
## magnes   -0.3134037 -0.5176612  0.73631729  0.604036129  0.11833719
##                potas     phospho     calcium      magnes
## burn      0.486550345 -0.17529690 -0.12126250 -0.31340373
## sugar     0.194768075  0.17827541 -0.46209806 -0.51766120
## nicotine -0.294403785 -0.02680349  0.68009707  0.73631729
## nitrogen -0.007483429 -0.09302834  0.62717077  0.60403613
## chlorine -0.099042191 -0.05580311  0.13825072  0.11833719
## potas     1.000000000 -0.11192376 -0.59246374 -0.61185069
## phospho  -0.111923762  1.00000000  0.08268827  0.06812773
## calcium  -0.592463735  0.08268827  1.00000000  0.76380091
## magnes   -0.611850688  0.06812773  0.76380091  1.00000000
```

b)

Fit a simple linear regression model using lm() and print the model output. Choose the most pertinent explanatory variable to model the response variable, providing a motivation for your choice from the output in (a).

```r
# fit linear regression model
lm.chlorine <- lm(burn ~ chlorine, data = tobacco)

# plot the relationship to visualise
plot(x = tobacco$chlorine,
     y = tobacco$burn,
     ylab = "Burn rate",
     xlab = "Chlorine level")
abline(lm.chlorine, col="red", lty=2)
```



I plotted chlorine (explanatory variable) against burn (response variable). The reason I chose chlorine is because it has the highest (absolute) correlation coefficient (correlation coefficient = -0.6234392), indicating a strong negative linear relationship. This makes it the most suitable predictor for our linear model, as variables with higher correlation tend to contribute more effectively to explaining variability in the response variable.

c)

From first principles, create a function to fit a simple linear regression model, and find the least squares estimates using optim(). Compare the results to those found in (b).

```
# use chlorine because it has the strongest correlation (neg corr)

# here we optimising for intercept and coefficients
regression_f <- function(betav, x, y){
  # x = explanatory variable
  # y = response variable
  # betav = c(intercept, slope)
  intercept <- betav[1]
  slope <- betav[2]
  fitted_values <- intercept + slope*x
  residuals <- y - fitted_values
  SSE <- sum(residuals^2) # a loss function return (SSE)
  return(SSE) # we are optimising for lowest SSE
}

olsfit <- optim(par = c(10,10),
                fn = regression_f,
                y = tobacco$burn,
                x = tobacco$chlorine)

olsfit$par[1] # intercept = 2.111097
```

```
## [1] 2.111097
```

```
olsfit$par[2] # slope = -0.1702047
```

```
## [1] -0.1702047
```

We used chlorine as the predictor since it has the strongest correlation of all the variables. We optimised for the lowest SSE. Our predicted outputs using least squares estimates using `optim()` were as follows:

- intercept: 2.111097
- slope: -0.1702047

These estimates matched the estimates given in the `lm()` function.

d)

Use optim() o maximise the [given] loglikelihood. Once again, assume that you only have one explanatory variable. Take note that you now also have to estimate sigma and that it has to be positive. Read up on the use of optim() and how to set constraints on parameters. You might also consider transforming sigma to some other variable.

```
int.par <- c(0.5, 0.5, 0.5)
n <- nrow(tobacco)

likelihood_f <- function(int.par, x, y){
  var <- int.par[1]
  intercept <- int.par[2]
  slope <- int.par[3]
  ll <- -0.5*n*log(2*pi)-0.5*n*log(var)-0.5*(1/var)*sum((y-intercept-slope*x)^2)
  return(-ll) # must use negative log-likelihood
}

# use optim() to optimise the max log likelihood
optim(par = int.par,
      fn = likelihood_f,
      y = tobacco$burn,
      x = tobacco$chlorine,
      method = "L-BFGS-B",
      lower = c(0.0001, -Inf, -Inf), # setting constraints, variance must be > 0
      upper = c(Inf, Inf, Inf))
```

```
## $par
## [1]  0.01639906  2.11414937 -0.17178162
##
## $value
## [1] -15.9387
##
## $counts
## function gradient
##       71       71
##
## $convergence
## [1] 0
##
## $message
## [1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

Part 2: Multiple Regression in R

Using the tobacco data set, let's fit a multiple regression model.

a)

Using output from Part 1, fit a simple linear model using the most appropriate predictor.

```
lm <- lm(tobacco$burn ~ tobacco$chlorine)
```

Take note of the correlation between the response variable and the fitted values of the regression. The square of this amount is known as the coefficient of determination ('Multiple R-squared' or simply R^2).

```
cor(tobacco$burn,tobacco$chlorine) # correaltion = -0.6234392
```

```
## [1] -0.6234392
```

```
cor(tobacco$burn,tobacco$chlorine)^2 # Multiple R squared = 0.3886764
```

```
## [1] 0.3886764
```

Explain what this quantity represents and why a larger R^2 is preferred.

- R squared (the coefficient of determination) is the amount of variation in the dependent variable that can be explained by the independent variables. It is measured on a 0 to 1 (0% to 100%) scale nad is a proportion of the amount of variance that can be explained by the model divided by the total variable.
- A larger value of R squared is preferred since it indicates that the model explains a greater proportion of the variability in the dependent variable, meaning the independent variables collectively provide a better fit to the data. (However, a very high R-squared value may also indicate over-fitting, especially in complex models, so it should be interpreted alongside other model evaluation metrics.)

Use the summary output and take note of the R2 amount. Comment on the significance of the beta parameters and whether regression analysis should be undertaken on this data.

The R squared value is 0.3887, meaning that 38.87% of the variance of the burn variable can be explained by the chlorine variable.

The beta coefficient is statistically significant, with a p-value of less than 0.001, suggesting that chlorine is a meaningful predictor in this linear model, and that regression analysis is appropriate.

```
summary(lm)
```

```
##
## Call:
## lm(formula = tobacco$burn ~ tobacco$chlorine)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.295267 -0.064830 -0.006598  0.093709  0.298555
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        2.11419    0.11460  18.449 2.79e-15 ***
## tobacco$chlorine -0.17180    0.04493  -3.824  0.00087 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.1333 on 23 degrees of freedom
## Multiple R-squared:  0.3887, Adjusted R-squared:  0.3621
## F-statistic: 14.62 on 1 and 23 DF,  p-value: 0.00087
```

```
# Multiple R-squared:  0.3887
# beta parameters are significant (p-value: 0.00087) so regression analysis can be undertaken
```

b)

Add the remaining variables to the regression model one at a time. Add the variable with the next largest (absolute) correlation coefficient between with the response variable and so forth. Take note of the correlation coefficient between the fitted values of the response variable for the new regression equations and the response variable, the R^2 value and the significance of the beta parameters.

```
cor(tobacco)[,1] # view correlation coefficients between dependent (burn) and independent (remaining) v
```

```
##       burn      sugar   nicotine   nitrogen   chlorine      potas    phospho
##  1.0000000 -0.3049965  0.2087690  0.2260002 -0.6234392  0.4865503 -0.1752969
##    calcium     magnes
## -0.1212625 -0.3134037
```

```
lm2 <- lm(burn ~ chlorine + potas, data = tobacco)
summary(lm2)
```

```
##
## Call:
## lm(formula = burn ~ chlorine + potas, data = tobacco)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.213975 -0.079165 -0.007944  0.061053  0.310827
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.45868    0.23584   6.185 3.17e-06 ***
## chlorine    -0.16009    0.03867  -4.139 0.000429 ***
## potas        0.27997    0.09159   3.057 0.005779 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1142 on 22 degrees of freedom
## Multiple R-squared:  0.5709, Adjusted R-squared:  0.5319
## F-statistic: 14.64 on 2 and 22 DF,  p-value: 9.076e-05
```

```
lm3 <- lm(burn ~ chlorine + potas + magnes, data = tobacco)
summary(lm3)
```

```
##
## Call:
## lm(formula = burn ~ chlorine + potas + magnes, data = tobacco)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.215720 -0.073394 -0.005841  0.061671  0.313299
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.39793    0.45336   3.083  0.00563 **
## chlorine    -0.16055    0.03967  -4.047  0.00058 ***
## potas        0.29131    0.11793   2.470  0.02216 *
## magnes       0.03789    0.23941   0.158  0.87576
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1168 on 21 degrees of freedom
## Multiple R-squared:  0.5714, Adjusted R-squared:  0.5102
## F-statistic: 9.334 on 3 and 21 DF,  p-value: 0.0004039
```

```
lm4 <- lm(burn ~ chlorine + potas + magnes + sugar, data = tobacco)
summary(lm4)
```

```
##
## Call:
## lm(formula = burn ~ chlorine + potas + magnes + sugar, data = tobacco)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.251119 -0.054197 -0.009521  0.081827  0.266937
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.94662    0.62725   3.103   0.0056 **
## chlorine    -0.12407    0.04886  -2.540   0.0195 *
## potas        0.26179    0.11877   2.204   0.0394 *
## magnes      -0.19800    0.30256  -0.654   0.5203
## sugar       -0.02071    0.01659  -1.248   0.2264
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1153 on 20 degrees of freedom
## Multiple R-squared:  0.6024, Adjusted R-squared:  0.5229
## F-statistic: 7.576 on 4 and 20 DF,  p-value: 0.0006934
```

```
lm5 <- lm(burn ~ chlorine + potas + magnes + sugar + nitrogen, data = tobacco)
summary(lm5)
```

```
##
## Call:
## lm(formula = burn ~ chlorine + potas + magnes + sugar + nitrogen,
##     data = tobacco)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.190664 -0.067083 -0.002777  0.070342  0.255404
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.807351   0.623535   2.899  0.00921 **
## chlorine    -0.134664   0.048535  -2.775  0.01207 *
```

```
## potas        0.151891   0.142203   1.068  0.29885
## magnes      -0.484744   0.365169  -1.327  0.20009
## sugar       -0.006661   0.019323  -0.345  0.73409
## nitrogen     0.210198   0.156151   1.346  0.19410
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1131 on 19 degrees of freedom
## Multiple R-squared:  0.637,  Adjusted R-squared:  0.5415
## F-statistic: 6.669 on 5 and 19 DF,  p-value: 0.0009617
```

```
lm6 <- lm(burn ~ chlorine + potas + magnes + sugar + nitrogen + nicotine, data = tobacco)
summary(lm6)
```

```
##
## Call:
## lm(formula = burn ~ chlorine + potas + magnes + sugar + nitrogen +
##     nicotine, data = tobacco)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.199445 -0.041386  0.009105  0.063153  0.231525
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.951095   0.614432   3.175  0.00524 **
## chlorine    -0.098864   0.053292  -1.855  0.08003 .
## potas        0.159176   0.138374   1.150  0.26506
## magnes      -0.760281   0.402960  -1.887  0.07543 .
## sugar       -0.007739   0.018805  -0.412  0.68556
## nitrogen     0.091219   0.172690   0.528  0.60380
## nicotine     0.134966   0.093299   1.447  0.16520
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1099 on 18 degrees of freedom
## Multiple R-squared:  0.6748, Adjusted R-squared:  0.5664
## F-statistic: 6.226 on 6 and 18 DF,  p-value: 0.00112
```

```
lm7 <- lm(burn ~ chlorine + potas + magnes + sugar + nitrogen + nicotine + phospho, data = tobacco)
summary(lm7)
```

```
##
## Call:
## lm(formula = burn ~ chlorine + potas + magnes + sugar + nitrogen +
##     nicotine + phospho, data = tobacco)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.176215 -0.046195 -0.004722  0.059891  0.220846
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.07313    0.64884   3.195   0.0053 **
## chlorine    -0.10990    0.05646  -1.946   0.0683 .
```

```
## potas        0.15537    0.14058   1.105   0.2845
## magnes      -0.70384    0.41734  -1.687   0.1100
## sugar       -0.00335    0.02014  -0.166   0.8699
## nitrogen     0.10030    0.17581   0.571   0.5758
## nicotine     0.12843    0.09520   1.349   0.1950
## phospho     -0.45047    0.65968  -0.683   0.5039
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1116 on 17 degrees of freedom
## Multiple R-squared:  0.6835, Adjusted R-squared:  0.5532
## F-statistic: 5.245 on 7 and 17 DF,  p-value: 0.002489
```

```
lm8 <- lm(burn ~ chlorine + potas + magnes + sugar + nitrogen + nicotine + phospho + calcium, data = tob
summary(lm8)
```

```
##
## Call:
## lm(formula = burn ~ chlorine + potas + magnes + sugar + nitrogen +
##     nicotine + phospho + calcium, data = tobacco)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.149403 -0.048228  0.001326  0.047505  0.137921
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.563675   0.712327   2.195   0.0432 *
## chlorine    -0.134570   0.056919  -2.364   0.0310 *
## potas        0.305924   0.168749   1.813   0.0887 .
## magnes      -0.645189   0.404667  -1.594   0.1304
## sugar       -0.004242   0.019449  -0.218   0.8301
## nitrogen    -0.032790   0.191465  -0.171   0.8662
## nicotine     0.091315   0.095146   0.960   0.3515
## phospho     -0.623569   0.647028  -0.964   0.3495
## calcium      0.179185   0.119419   1.500   0.1530
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1077 on 16 degrees of freedom
## Multiple R-squared:  0.7226, Adjusted R-squared:  0.5838
## F-statistic: 5.209 on 8 and 16 DF,  p-value: 0.002488
```

Fit the full model. i.e. the model that includes all of the explanatory variables. Comment on the estimation results.

```
full_lm <- lm(tobacco$burn ~ ., data = tobacco)
summary(full_lm)
```

```
##
## Call:
## lm(formula = tobacco$burn ~ ., data = tobacco)
##
```

```
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.149403 -0.048228  0.001326  0.047505  0.137921
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.563675   0.712327   2.195   0.0432 *
## sugar       -0.004242   0.019449  -0.218   0.8301
## nicotine     0.091315   0.095146   0.960   0.3515
## nitrogen    -0.032790   0.191465  -0.171   0.8662
## chlorine    -0.134570   0.056919  -2.364   0.0310 *
## potas        0.305924   0.168749   1.813   0.0887 .
## phospho     -0.623569   0.647028  -0.964   0.3495
## calcium      0.179185   0.119419   1.500   0.1530
## magnes      -0.645189   0.404667  -1.594   0.1304
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1077 on 16 degrees of freedom
## Multiple R-squared:  0.7226, Adjusted R-squared:  0.5838
## F-statistic: 5.209 on 8 and 16 DF,  p-value: 0.002488
```

Suggest a suitable final model/s. Explain your choice. Are all of the beta coefficients significant? Is the F statistic significant?

Once you have chosen a suitable model you now have to check whether or not the assumptions made at the start of the estimation phase are satisfied. i.e. Are the residuals normally distributed? What is mean of the residual series? Are the residuals homoscedastic? Are the residuals independent?

Plot the histogram of the estimated residuals. Do they look normally distributed? (don't always trust your eye). Use the car package and plot the QQ plot of the estimated residuals. (A QQ plot can be produced by first order- ing the estimated residuals in ascending order and then using a standard normal to calculate the quantiles associated with the ordered residuals. If the residuals are normally distributed the plotted values should lie close to a straight line.)

Formal tests of normality can be undertaken by using the ks.test and the shapiro.test function. Use these functions to test whether the residuals are normally distributed.

Write a function to fit a multiple regression model from first principles. Use this function and optim() to find the MLE estimates for the coeffi- cients of your chosen model in (e).

```r
# initial parameter values:
B <- c(1, 0.1, -0.1, 0.1)
n <- nrow(tobacco)

n <- nrow(tobacco)

# Create X and Y matrixes. Note you will need a column of 1s in X to represent the intercept!
X <- data.frame(intercept = 1, tobacco[c("chlorine", "potas", "magnes")])
```

```r
X <- as.matrix(X)

Y <- tobacco$burn

# multiple linear regression
mlr_f <- function(B, X, Y){
  SSE <- t(Y-X%*%B) %*% (Y-X%*%B) # need to use matrix multiplication # a loss function return (SSE)
  return(SSE)
}

olsfit <- optim(par = B,
                fn = mlr_f,
                X = X,
                Y = Y)

olsfit$par
```

```
## [1]  1.39763080 -0.16045346  0.29135705  0.03785981
```

```r
# lm
lm(burn ~ chlorine, data = tobacco)
```

```
##
## Call:
## lm(formula = burn ~ chlorine, data = tobacco)
##
## Coefficients:
## (Intercept)     chlorine
##      2.1142      -0.1718
```