A
density of a solution
B
acidity of the solution
C
treatment time
D
temperature

On each row of **experiment.xslx** (check file pane), those four factors were set to certain levels, 1000 batches at those levels were producted, and the percentage of unacceptable batches out of 1000 were reported in column **P**. We would like to understand the relation between production factors and the quality of product.

```
library(magrittr)
library(tidyverse)
library(modelr)
library(pander)
library(leaps)
library(ISLR)
```

Exploration [suggested time: 10 minutes]

1. Read data, (use readxl package) print the first six rows, and explore the relations between variables (graphical and tabular summaries). For full credit, you should comment on the plots and tables.

We can fit both linear and binomial regression models to the that experiment data. In the former, we overlook the binomial distribution inherent to the response, but gain an extra variance parameter. The variance will on the other be constant across cases, which we know is not true for binomial distribution. So both approaches have pros and cons. Your job is to explore both models and their variants and choose the best one.

```
library(readxl)
d <- read_excel("experiment.xlsx")
d %>% head() %>% pander(caption="Here is a glimpse of data that we will an
alyze.")
```

Here is a glimpse of data that we will analyze.

A	B	C	D	P

A	B	C	D	P
1.125	3.5	40	150	14
1.125	3.5	50	120	13.5
1.125	3.5	30	135	18.3
1.125	3	40	120	17.4
1.125	3	50	135	16.3
1.125	3	30	150	13.9

```
summary(d)
```

```
##        A                B             C            D             P          
##  Min.   :1.121   Min.   :3.0   Min.   :30   Min.   :120   Min.   :10.30  
##  1st Qu.:1.121   1st Qu.:3.0   1st Qu.:30   1st Qu.:120   1st Qu.:11.85  
##  Median :1.123   Median :3.5   Median :40   Median :135   Median :13.40  
##  Mean   :1.123   Mean   :3.5   Mean   :40   Mean   :135   Mean   :14.03  
##  3rd Qu.:1.125   3rd Qu.:4.0   3rd Qu.:50   3rd Qu.:150   3rd Qu.:15.80  
##  Max.   :1.125   Max.   :4.0   Max.   :50   Max.   :150   Max.   :21.00
```

We can say that all five variables are continuous variables.

```
quant <- d %>% select_if(is.numeric)
names(quant)
```

```
## [1] "A" "B" "C" "D" "P"
```

```
cor(d)
```

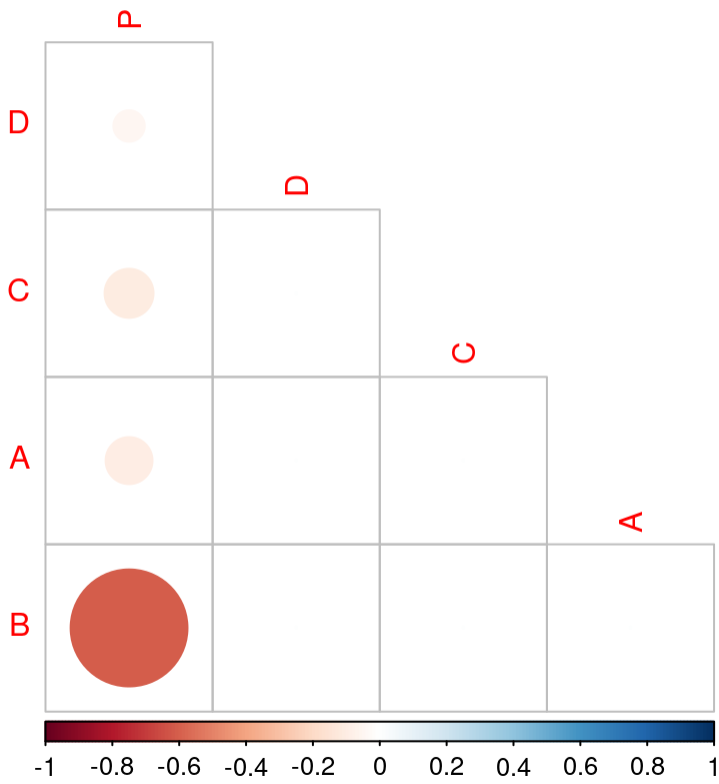
```
##             A          B          C           D           P
## A  1.00000000  0.0000000  0.0000000  0.00000000 -0.09713451
## B  0.00000000  1.0000000  0.0000000  0.00000000 -0.60310380
## C  0.00000000  0.0000000  1.0000000  0.00000000 -0.10583312
## D  0.00000000  0.0000000  0.0000000  1.00000000 -0.04349306
## P -0.09713451 -0.6031038 -0.1058331 -0.04349306  1.00000000
```

The table above suggests that only predictor variables are not correlated with each other. B (acidity of the solution) is the only one that is correlated with P (percentage of unacceptable batches) to a certain extent. Correlation coefficient is -0.6.

```
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```
d %>%
  select_if(is.numeric) %>%
  cor(use = "pairwise.complete.obs") %>%
  corrplot(type = "lower", diag = FALSE, order = "hclust")
```



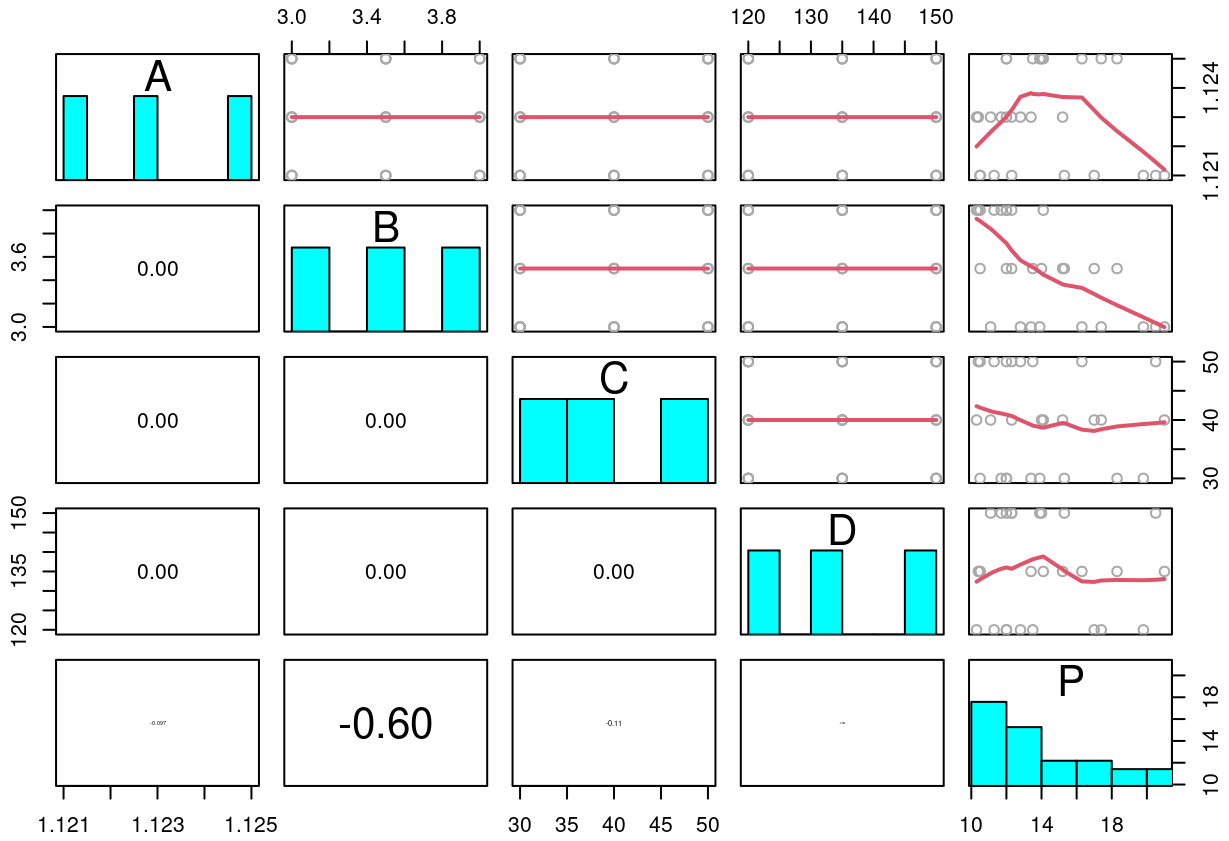
The above graph verifies our statement. Except B and P, no other variables are correlated with each other. And correlation between B and P is also not very strong (60% correlation). One can only say that there is very weak correlation between the pairs (A,P), (C,P) and (D,P). (You see above that color is very transparent.)

Let us go into more detail and look at histograms and distributions.

```
panel.hist <- function(x, ...)
{
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(usr[1:2], 0, 1.5) )
    h <- hist(x, plot = FALSE)
    breaks <- h$breaks; nB <- length(breaks)
    y <- h$counts; y <- y/max(y)
    rect(breaks[-nB], 0, breaks[-1], y, col = "cyan", ...)
}

panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...)
{
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    rr <- cor(x, y)
    r <- abs(rr)
    txt <- format(c(rr, 0.123456789), digits = digits)[1]
    txt <- paste0(prefix, txt)
    if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
    text(0.5, 0.5, txt, cex = cex.cor * r)
}

d %>%
    pairs(diag.panel = panel.hist,
        lower.panel = panel.cor,
        upper.panel = function (...) panel.smooth(..., lwd = 2, col = "dar
kgray"))
```



Finding 1

By looking at the above set of plots and the data itself, we see that, although A, B, C and D are continuous variables, their ranges are very limited. In fact they can be treated as factors, but in this case I will continue with assuming they are discrete-continuous random variables.

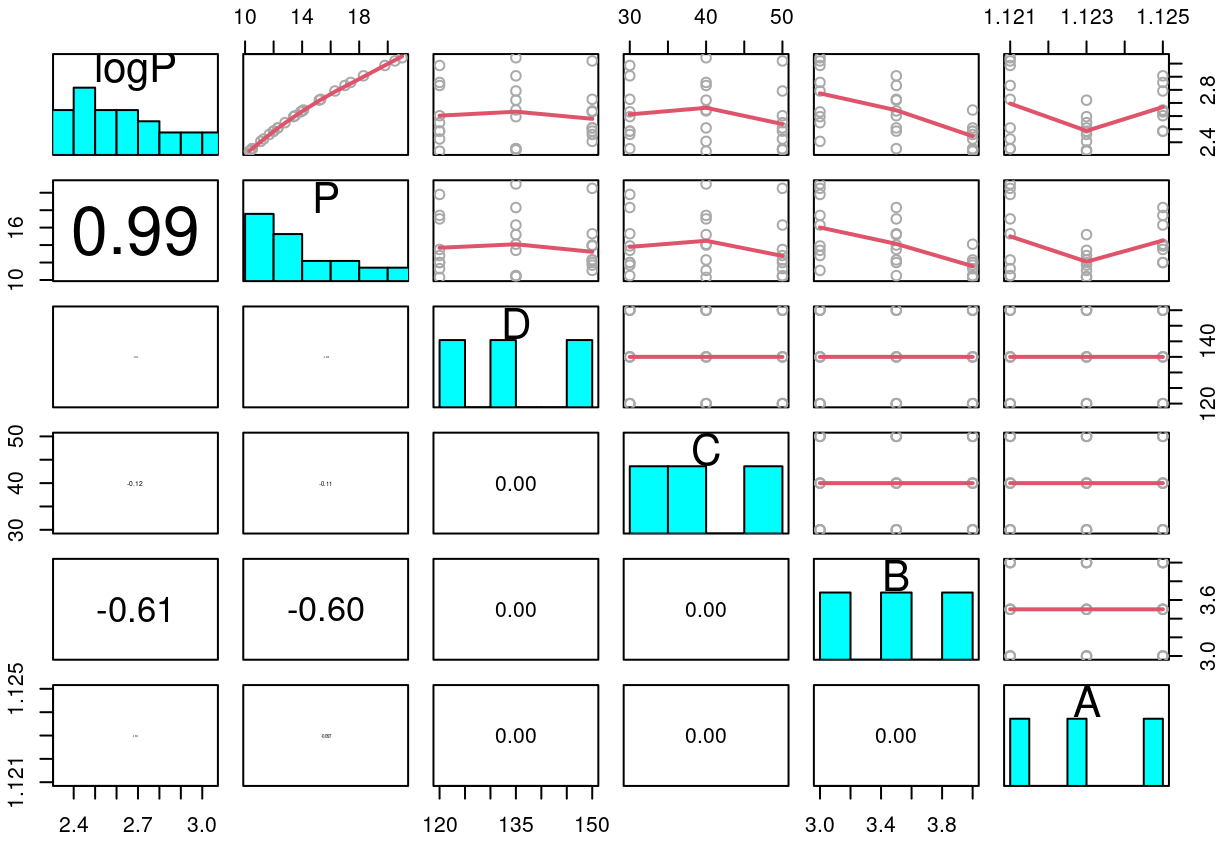
Finding 2

When we look at the bottom-right plot, we seen that our response variable *P* is not normally distributed and it is highly right-skewed. We should fix this problem since many of the statistical methods we might perform assume that response variable, as well as the residual errors, are normally distributed. Let us try log transformation to fix this issue. Also, I'm taking P to first plot so that I can look at it more easily.

```
panel.hist <- function(x, ...)
{
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(usr[1:2], 0, 1.5) )
    h <- hist(x, plot = FALSE)
    breaks <- h$breaks; nB <- length(breaks)
    y <- h$counts; y <- y/max(y)
    rect(breaks[-nB], 0, breaks[-1], y, col = "cyan", ...)
}

panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...)
{
    usr <- par("usr"); on.exit(par(usr))
    par(usr = c(0, 1, 0, 1))
    rr <- cor(x, y)
    r <- abs(rr)
    txt <- format(c(rr, 0.123456789), digits = digits)[1]
    txt <- paste0(prefix, txt)
    if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
    text(0.5, 0.5, txt, cex = cex.cor * r)
}

d %>%
  mutate(logP = log(P)) %>%
          rev() %>%
    pairs(diag.panel = panel.hist,
        lower.panel = panel.cor,
        upper.panel = function (...) panel.smooth(..., lwd = 2, col = "dar
kgray"))
```

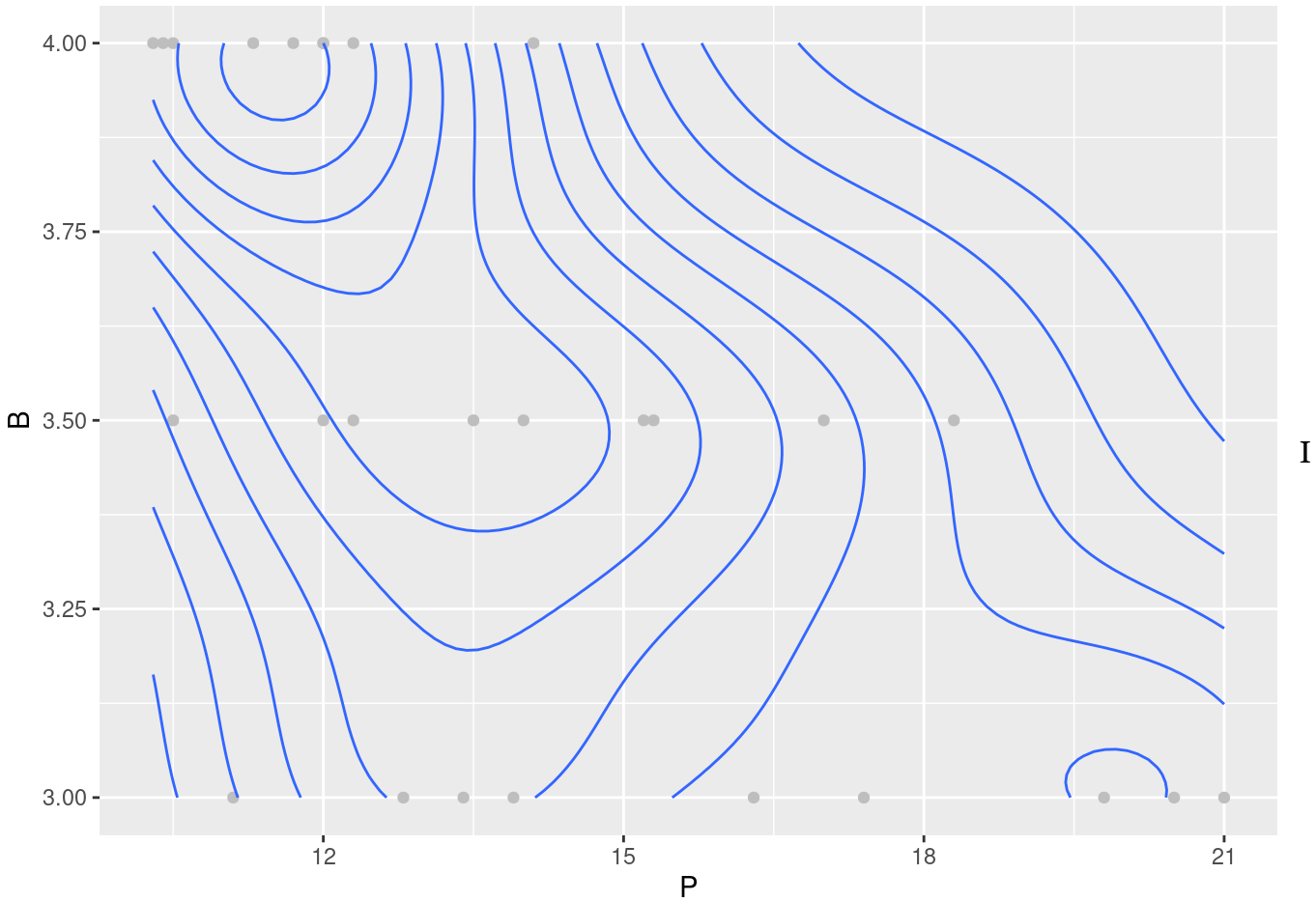


Finding 3

Log transformation really did well and put *P* 's distribution into a symmetric form. That's why I will not try Box-Cox or any other transformation technique further.

Let us look at another plot type, which is density plot:

```
d %>%
  ggplot(aes(P,B)) +
  geom_point(col="gray") +
  geom_density_2d()
```



I

couldn't extract any information by looking at this graph. I also tried with aes(P,A),aes(P,C),aes(P,D), but they didn't help either.

I will continue with the current setting.

Linear regression [suggested time: 45 minutes]

2. Propose a linear regression model for the percentage of unacceptable batches (or its preferred transformation) and predictors. Fit the model to data and see if it explains variation well. You may need to iterate several times until you find a satisfactory model.

```
d2 <- d %>% mutate(P=log(P)) #applying log transformation
linmod <- lm(P~., data=d2) #regressing all variable against log(P)
summary(linmod)
```

```
##
## Call:
## lm(formula = P ~ ., data = d2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.36036 -0.14399  0.01731  0.09264  0.27243
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.3152793 24.2485217   0.425  0.67468
## A           -5.6857670 21.5876078  -0.263  0.79471
## B           -0.3162772  0.0863504  -3.663  0.00137 **
## C           -0.0030675  0.0043175  -0.710  0.48488
## D           -0.0006089  0.0028783  -0.212  0.83442
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1832 on 22 degrees of freedom
## Multiple R-squared:  0.3895, Adjusted R-squared:  0.2785
## F-statistic: 3.509 on 4 and 22 DF,  p-value: 0.02321
```

Before looking at diagnostic plots, we can say that only significant variable (at a significance level of 99%) is B. Other variables' p-values are high indicating that they are not significant. The model's R^2 is not very good, it is 39%. We should improve the model.

For this purpose, I will analyze effects plots and diagnostic plots of the regression model.

```
library(effects)
```

```
## Loading required package: carData
```

```
## lattice theme set by effectsTheme()
## See ?effectsTheme for details.
```

```
allEffects(linmod, partial.residuals = TRUE) %>% plot() #sorry for the err
or below, anyway it shows plots
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## pseudoinverse used at 1.121
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## neighborhood radius 0.00202
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## There are other near singularities as well. 4.0804e-06
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## pseudoinverse used at 1.121
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## neighborhood radius 0.00202
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## There are other near singularities as well. 4.0804e-06
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## pseudoinverse used at 1.121
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## neighborhood radius 0.00202
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## There are other near singularities as well. 4.0804e-06
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## pseudoinverse used at 1.121
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## neighborhood radius 0.00202
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## There are other near singularities as well. 4.0804e-06
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## pseudoinverse used at 1.121
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## neighborhood radius 0.00202
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## There are other near singularities as well. 4.0804e-06
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## pseudoinverse used at 2.995
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## neighborhood radius 0.505
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## There are other near singularities as well. 0.25502
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## pseudoinverse used at 2.995
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## neighborhood radius 0.505
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## There are other near singularities as well. 0.25502
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## pseudoinverse used at 2.995
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## neighborhood radius 0.505
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## There are other near singularities as well. 0.25502
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## pseudoinverse used at 2.995
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## neighborhood radius 0.505
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## There are other near singularities as well. 0.25502
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## pseudoinverse used at 2.995
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## neighborhood radius 0.505
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## There are other near singularities as well. 0.25502
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## pseudoinverse used at 29.9
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## neighborhood radius 10.1
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## There are other near singularities as well. 102.01
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## pseudoinverse used at 29.9
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## neighborhood radius 10.1
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## There are other near singularities as well. 102.01
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## pseudoinverse used at 29.9
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## neighborhood radius 10.1
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## There are other near singularities as well. 102.01
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## pseudoinverse used at 29.9
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## neighborhood radius 10.1
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## There are other near singularities as well. 102.01
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## pseudoinverse used at 29.9
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## neighborhood radius 10.1
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## There are other near singularities as well. 102.01
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## pseudoinverse used at 119.85
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## neighborhood radius 15.15
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## There are other near singularities as well. 229.52
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## pseudoinverse used at 119.85
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## neighborhood radius 15.15
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## There are other near singularities as well. 229.52
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## pseudoinverse used at 119.85
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## neighborhood radius 15.15
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## There are other near singularities as well. 229.52
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## pseudoinverse used at 119.85
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## neighborhood radius 15.15
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## reciprocal condition number 0
```

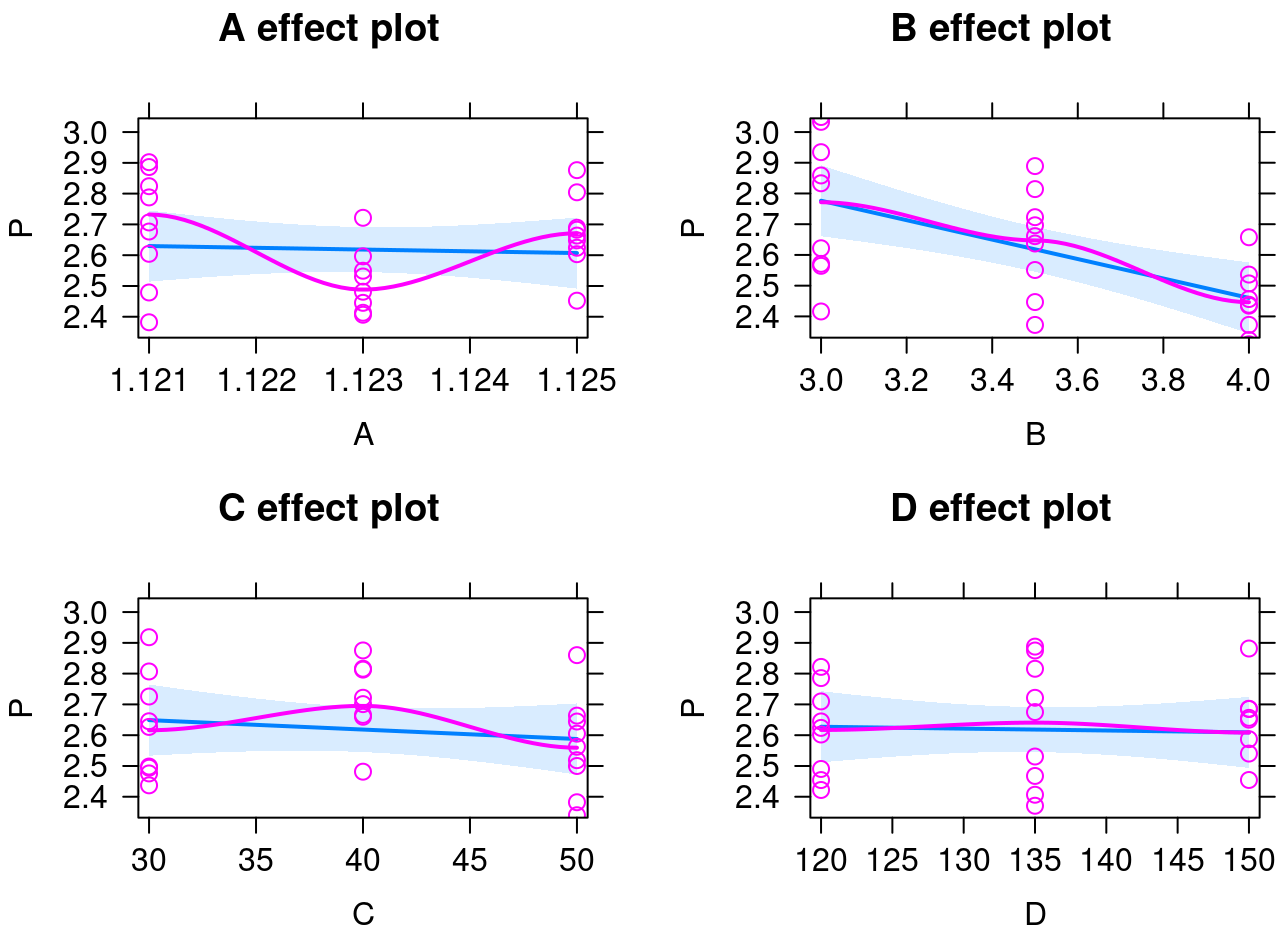
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## There are other near singularities as well. 229.52
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## pseudoinverse used at 119.85
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## neighborhood radius 15.15
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## reciprocal condition number 0
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric = FAL
SE, :
## There are other near singularities as well. 229.52
```



Finding 4

It seems like the predictor *D* has nothing to do with our response variable *P*. Its explanatory power is very low. We should remove it. (after checking interactions!)

Finding 5

Top-left plot suggests that we may use a quadratic term for the predictor *A*, as one can detect the U-shape in its effect plot. This is also valid for C.

Finding 6

It is clear that as *B* increases *P* decreases. This is also verified by its negative coefficient.

```
linmod2 <- lm(log(P)~. + poly(A,2)+ poly(C,2), data=d2) #adding quadratic
terms for A and C
summary(linmod2)
```

```
##
## Call:
## lm(formula = log(P) ~ . + poly(A, 2) + poly(C, 2), data = d2)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.110081 -0.021915 -0.006611  0.034123  0.083778
##
## Coefficients: (2 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.7608278  7.7205243   0.358 0.724392
## A           -1.1661782  6.8733118  -0.170 0.866976
## B           -0.1191192  0.0274932  -4.333 0.000323 ***
## C           -0.0011943  0.0013747  -0.869 0.395268
## D           -0.0002014  0.0009164  -0.220 0.828258
## poly(A, 2)1         NA         NA      NA       NA
## poly(A, 2)2  0.1783472  0.0583220   3.058 0.006208 **
## poly(C, 2)1         NA         NA      NA       NA
## poly(C, 2)2 -0.0654453  0.0583220  -1.122 0.275096
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05832 on 20 degrees of freedom
## Multiple R-squared:  0.6017, Adjusted R-squared:  0.4822
## F-statistic: 5.036 on 6 and 20 DF,  p-value: 0.002706
```

Finding 6

Summary result showed us that quadratic term for the predictor *A* worked well, and our model's R^2 is increased from 39% to 60%. (Adjusted R^2 is still low!(48%)). On the other hand, quadratic term of *C* did not provide us any additional explanatory power. We don't need it.

We should improve model further. I will look at interactions. Additionally, I am adding quadratic term for the predictor *B*. (its effect plot gives us a hint)

```
linmod3 <- lm(P~. + poly(A,2)*D+D*poly(B,2), data=d2) #adding interaction
between D and others.
summary(linmod3)
```

```
##
## Call:
## lm(formula = P ~ . + poly(A, 2) * D + D * poly(B, 2), data = d2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.33585 -0.06621 -0.02204  0.06709  0.25433
##
## Coefficients: (2 not defined because of singularities)
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -7.206e+01  2.526e+02  -0.285    0.779
## A              7.021e+01  2.249e+02   0.312    0.759
## B             -1.133e+00  8.997e-01  -1.259    0.226
## C             -3.067e-03  4.065e-03  -0.755    0.461
## D             -6.089e-04  2.710e-03  -0.225    0.825
## poly(A, 2)1           NA         NA      NA       NA
## poly(A, 2)2    8.302e-01  1.909e+00   0.435    0.669
## poly(B, 2)1           NA         NA      NA       NA
## poly(B, 2)2   -2.878e-02  1.909e+00  -0.015    0.988
## D:poly(A, 2)1 -4.770e-03  1.408e-02  -0.339    0.739
## D:poly(A, 2)2 -2.631e-03  1.408e-02  -0.187    0.854
## D:poly(B, 2)1  1.284e-02  1.408e-02   0.912    0.375
## D:poly(B, 2)2 -4.286e-04  1.408e-02  -0.030    0.976
##
## Residual standard error: 0.1724 on 16 degrees of freedom
## Multiple R-squared:  0.6065, Adjusted R-squared:  0.3605
## F-statistic: 2.466 on 10 and 16 DF,  p-value: 0.05204
```

Our model significantly got worse. We shouldn't try this.

Continuing with the existing model, i.e. *linmod2*. I am applying step() function to perform stepwise selection.

```
linmod2 %>% step()
```

```
## Start:  AIC=-147.56
## log(P) ~ A + B + C + D + poly(A, 2) + poly(C, 2)
##
##
## Step:  AIC=-147.56
## log(P) ~ A + B + D + poly(A, 2) + poly(C, 2)
##
##
## Step:  AIC=-147.56
## log(P) ~ B + D + poly(A, 2) + poly(C, 2)
##
##              Df Sum of Sq      RSS     AIC
## - D           1  0.000164 0.068193 -149.49
## - poly(C, 2)  2  0.006850 0.074880 -148.97
## <none>                    0.068029 -147.56
## - poly(A, 2)  2  0.031906 0.099935 -141.18
## - B           1  0.063852 0.131881 -131.69
##
## Step:  AIC=-149.49
## log(P) ~ B + poly(A, 2) + poly(C, 2)
##
##              Df Sum of Sq      RSS     AIC
## - poly(C, 2)  2  0.006850 0.075044 -150.91
## <none>                    0.068193 -149.49
## - poly(A, 2)  2  0.031906 0.100099 -143.13
## - B           1  0.063852 0.132046 -133.65
##
## Step:  AIC=-150.91
## log(P) ~ B + poly(A, 2)
##
##              Df Sum of Sq      RSS     AIC
## <none>                    0.075044 -150.91
## - poly(A, 2)  2  0.031906 0.106950 -145.34
## - B           1  0.063852 0.138896 -136.29
```

```
##
## Call:
## lm(formula = log(P) ~ B + poly(A, 2), data = d2)
##
## Coefficients:
## (Intercept)            B  poly(A, 2)1  poly(A, 2)2
##    1.376245    -0.119119    -0.009895     0.178347
```

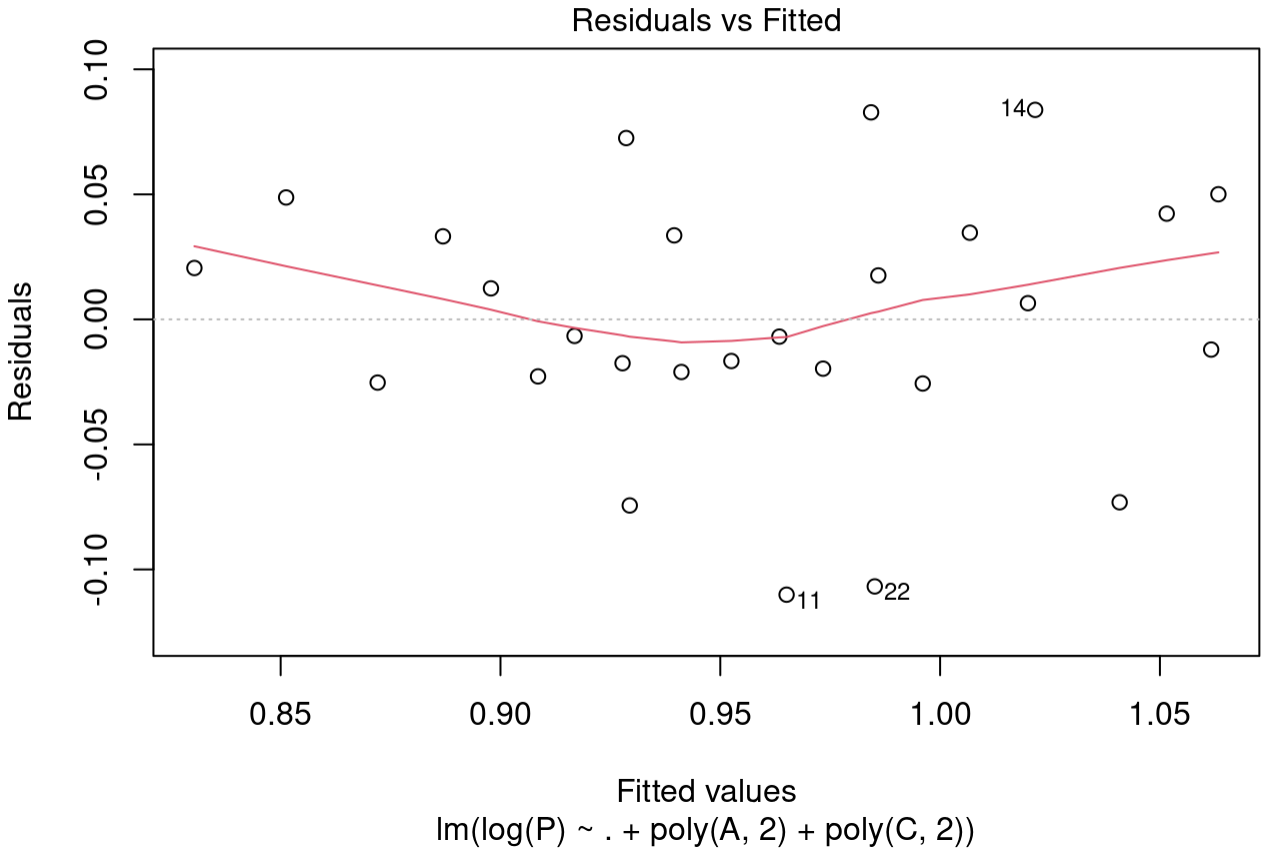
Finding 7

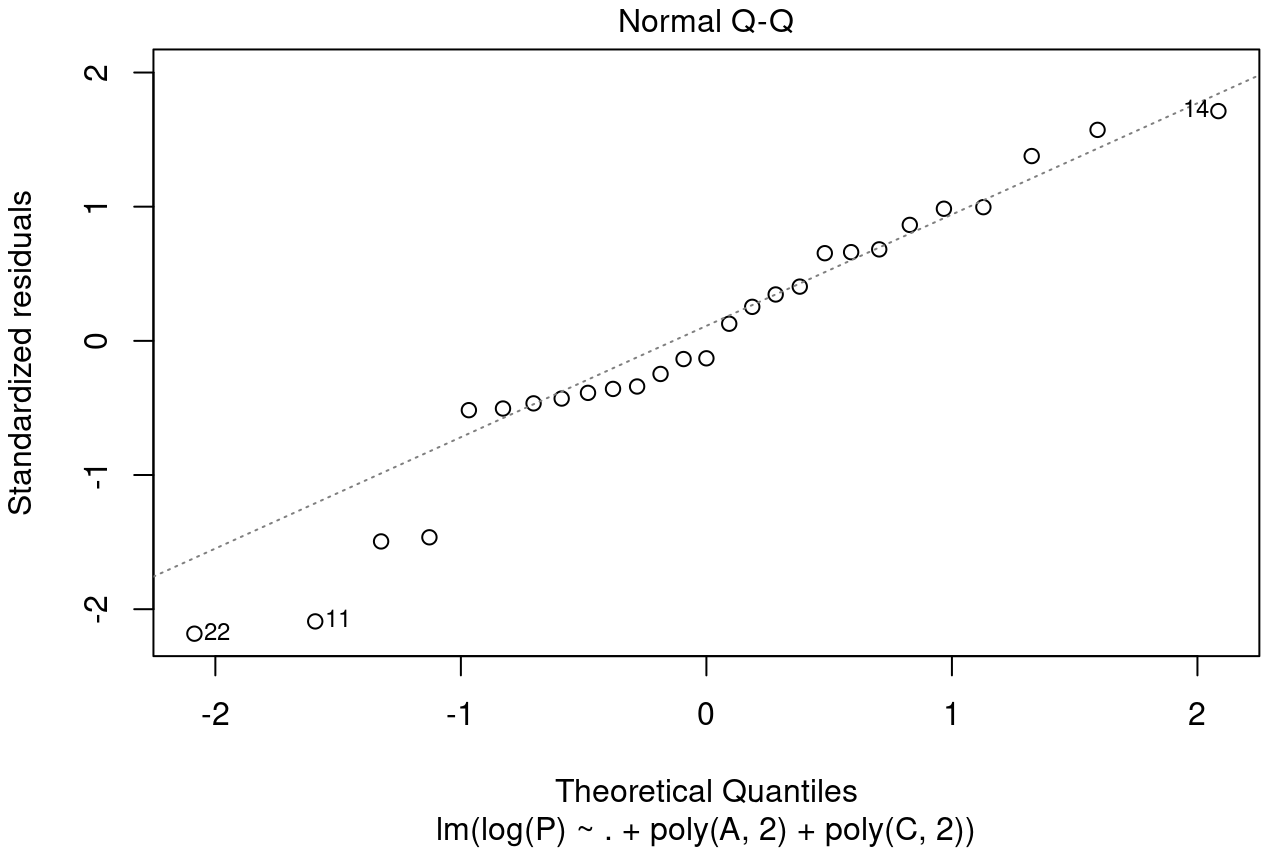
Step() function performed stepwise selection with respect to AIC values and suggests the model with B and A^2 as predictors.

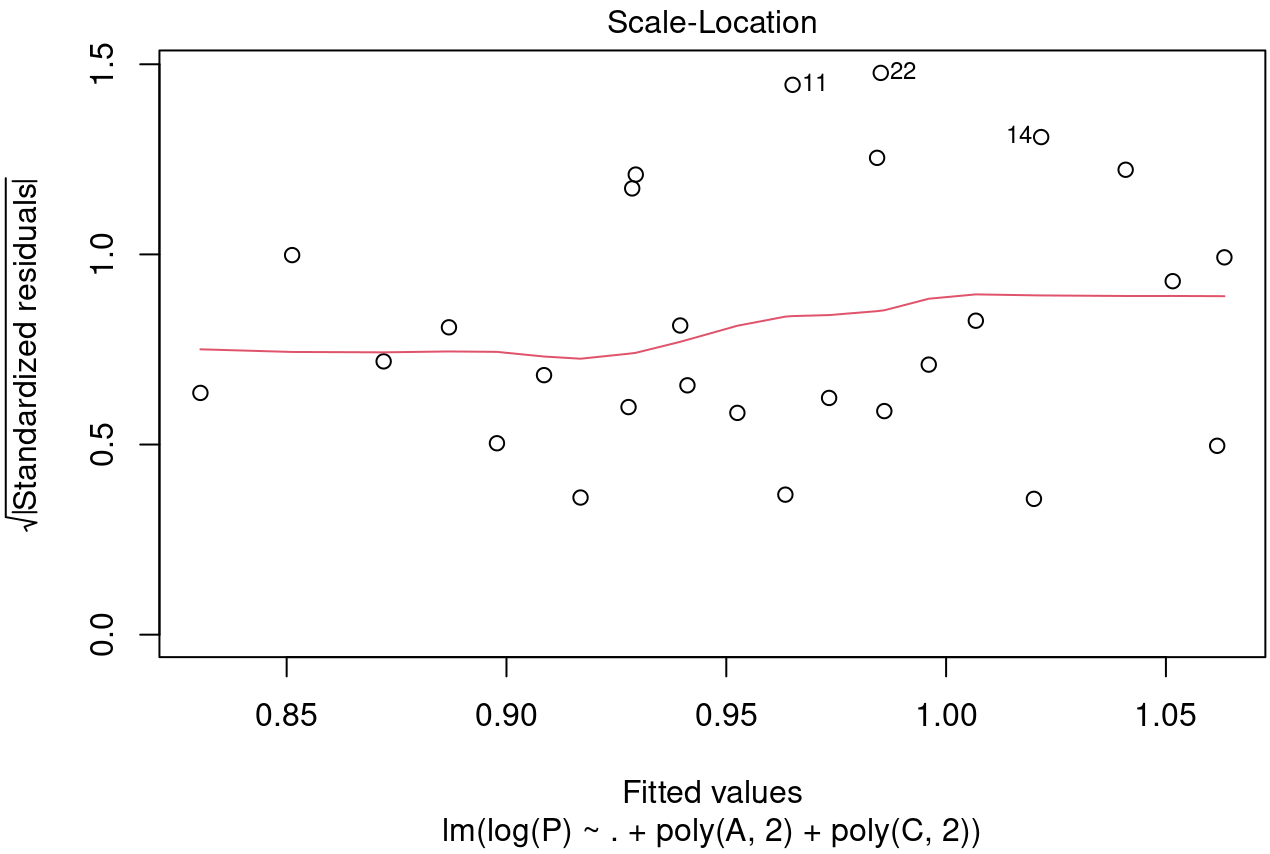
I will continue with this model (linmod2). I couldn't improve it further.

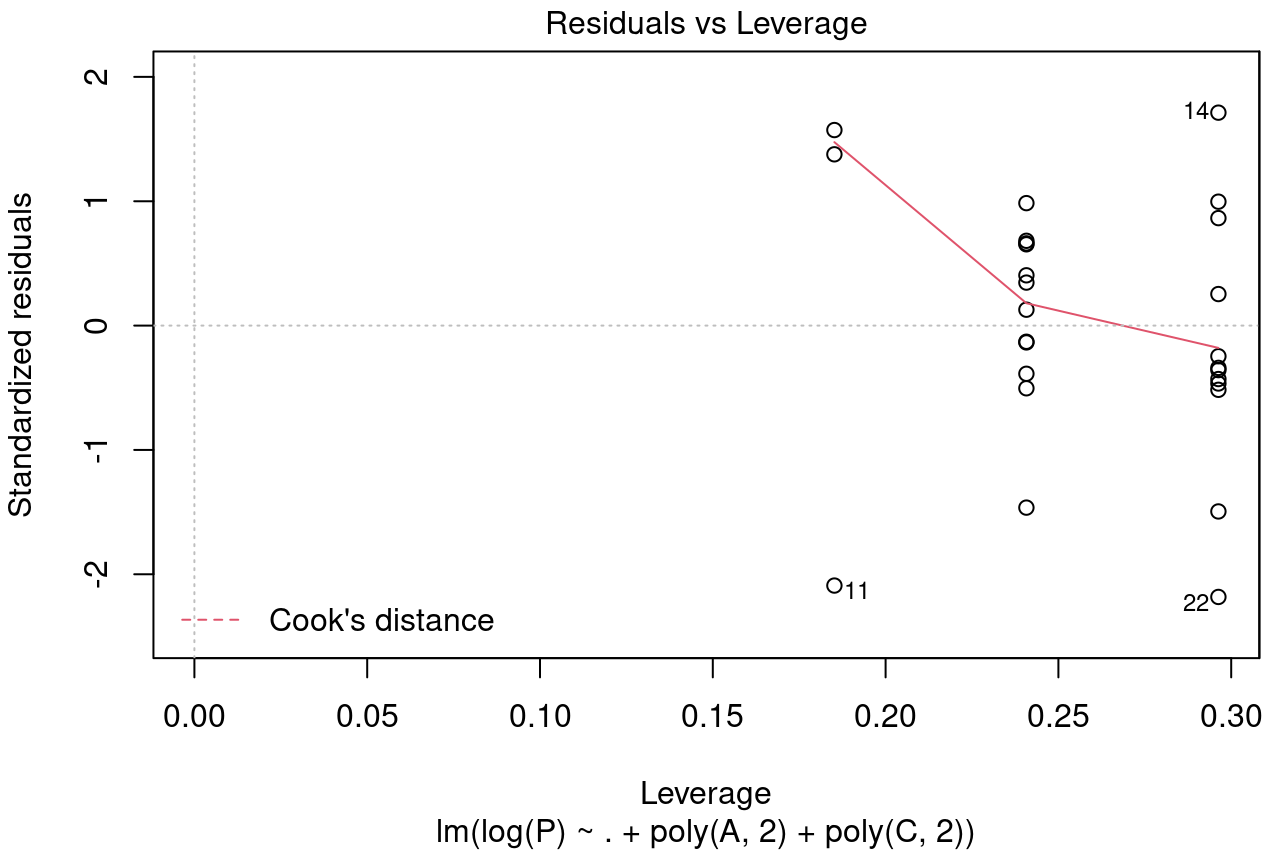
Let's check its diagnostic plots:

```
plot(linmod2)
```




Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(log(P) ~ . + poly(A, 2) + poly(C, 2))
14
22
11


Scale-Location
√|Standardized residuals|
Fitted values
lm(log(P) ~ . + poly(A, 2) + poly(C, 2))
11
22
14



There is pattern in the Residual vs. fitted plot, this implies that there is correlation between error terms. Unfortunately log transformation didnt help to fix this issue.

On the other hand, Normal Q-Q plot tells us that errors more or less follow normal distribution, as they lie on the straight line.

Scale location does not show us any pattern, which means the model is OK. I will revisit Leverage plot in later questions.

3. How good does your model fit to data? Explain.

To answer this question we may look at R^2 and/or RSE/mean(P).

```
names(summary(linmod2))
```

```
##  [1] "call"          "terms"         "residuals"     "coefficients"
##  [5] "aliased"       "sigma"         "df"            "r.squared"
##  [9] "adj.r.squared" "fstatistic"    "cov.unscaled"
```

```
summary(linmod2)$r.squared
```

```
## [1] 0.6017073
```

R^2 is 60%. This means that 60% of the variation in the response variable is explained by the predictor variables. 60% is not a perfect R^2 value but it is not the worst either. Let us also look at RSE/mean(P).

```
(summary(linmod2)$sigma)/mean(d2$P)
```

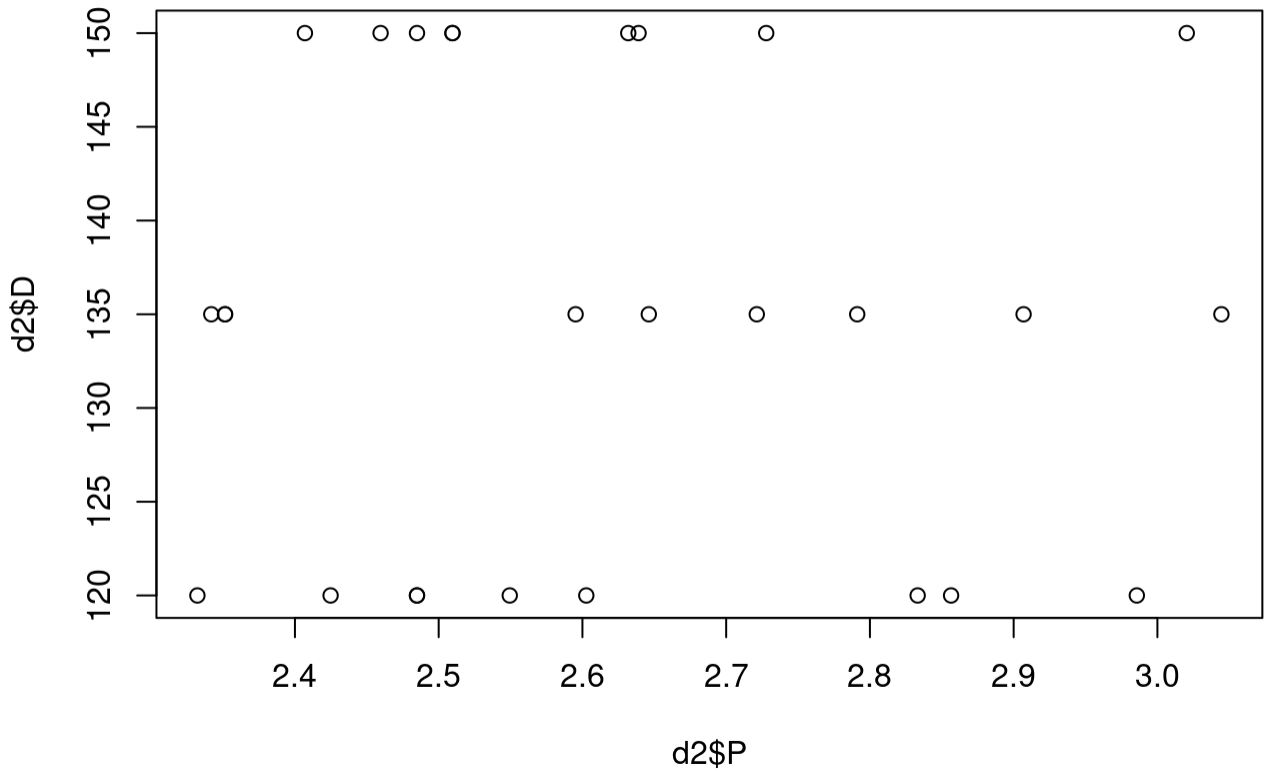
```
## [1] 0.02227479
```

We want this error percentage to be as low as possible. It is 5% which means the model gives somewhat accurate results.

4. How does the percentage of unacceptable products change with temperature (D) ? Does the effect of temperature (D) change with the other factors?

To answer this question we may look at plot below.

```
plot(d2$P,d2$D)
```



Plot tells us nothing. This is actually what we expected because previously I concluded that D is

not a significant variable. Nonetheless, I want to check its marginal relation (in isolation of other variables) anyway.

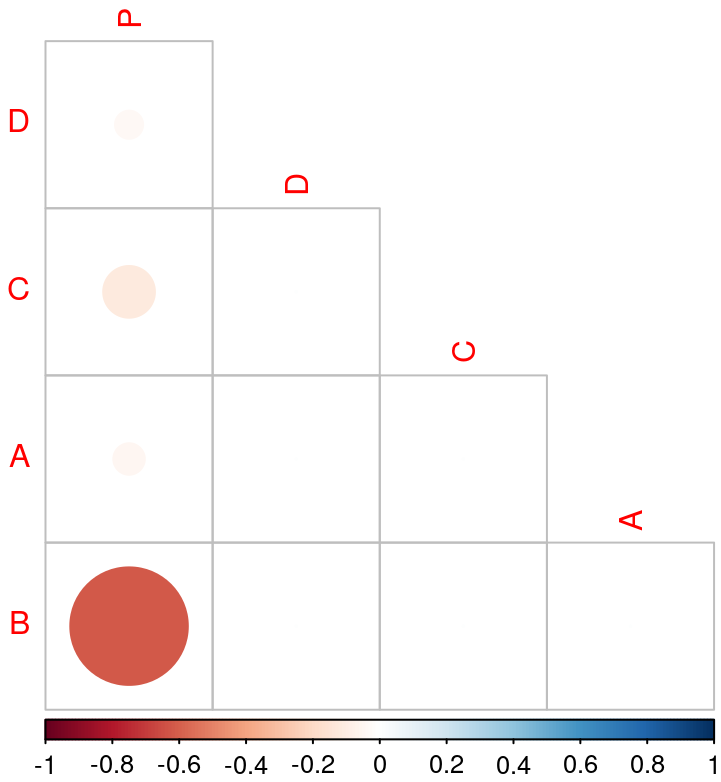
```
linmod.onlyD <- lm(P~D,data=d2)
summary(linmod.onlyD)
```

```
##
## Call:
## lm(formula = P ~ D, data = d2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.29529 -0.14605 -0.02474  0.14578  0.42623
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.7004947  0.4681398   5.769 5.18e-06 ***
## D           -0.0006089  0.0034535  -0.176    0.861
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2198 on 25 degrees of freedom
## Multiple R-squared:  0.001242,   Adjusted R-squared:  -0.03871
## F-statistic: 0.03108 on 1 and 25 DF,  p-value: 0.8615
```

The p-value is more than 5% or 10%. This means that we are unable to reject the null hypothesis of “H_0:D’s beta=0, i.e. it is not significant for the model”. We conclude that D cannot be used to explain the variation in P.

To answer the latter question, one can look at the correlation graph below;

```
d2 %>%
  select_if(is.numeric) %>%
  cor(use = "pairwise.complete.obs") %>%
  corrplot(type = "lower", diag = FALSE, order = "hclust")
```



Finding 8

D is not correlated with any of the variables. I also checked the interaction between them in linmod3 previously. One can confidently say that D is not useful for our analysis.

5. Are there influential points? If there are, how does your model change when you remove them (I am not suggesting that you should remove them, but I just want to know how significant they are)?

By looking at the leverage plot we can see 22 and 14 have Cook's Distance of 0.3. They are influential. Let us check what happens if we remove them.

```
linmod4 <- linmod2 %>% update(subset = -c(22, 14))
linmod4 %>% summary()
```

```
##
## Call:
## lm(formula = log(P) ~ . + poly(A, 2) + poly(C, 2), data = d2,
##     subset = -c(22, 14))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.085564 -0.026359  0.008072  0.028475  0.084138
##
## Coefficients: (2 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.703e-01  6.817e+00  -0.069 0.945757
## A            1.729e+00  6.073e+00   0.285 0.779091
## B           -1.232e-01  2.536e-02  -4.858 0.000126 ***
## C           -1.773e-03  1.215e-03  -1.460 0.161513
## D           -6.613e-05  8.452e-04  -0.078 0.938499
## poly(A, 2)1         NA         NA      NA       NA
## poly(A, 2)2  1.259e-01  5.252e-02   2.396 0.027651 *
## poly(C, 2)1         NA         NA      NA       NA
## poly(C, 2)2 -1.179e-01  5.252e-02  -2.245 0.037536 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04959 on 18 degrees of freedom
## Multiple R-squared:  0.6899, Adjusted R-squared:  0.5865
## F-statistic: 6.674 on 6 and 18 DF,  p-value: 0.0007593
```

Our R^2 is increased to 69%. We improved the model. Also quadratic term of C became significant. That means those observations (22 and 14) apply strong force to the regression line.

Binomial regression [suggested time: 45 mimutes]

Note the questions are basically the same as those for your linear regression model. However, because models are different, you may arrive at different conclusions.

6. Now propose a binomial regression model for the percentage of unacceptable batches and predictors. Fit the model to data and see if it explains variation well. You may need to iterate several times until you find a satisfactory model.

Interpretation

Firstly we need to add a binary outcome data to data to perform logistic regression. I do it as follows;

```
mean(d$P)
```

```
## [1] 14.03333
```

```
d3<- d%>%mutate(Perc.above.average=as.factor(ifelse(P>=mean(P),"Yes","N
o"))) %>% select(-P)
d3
```

```
## # A tibble: 27 x 5
##        A     B     C     D Perc.above.average
##    <dbl> <dbl> <dbl> <dbl> <fct>
##  1  1.12   3.5    40   150 No
##  2  1.12   3.5    50   120 No
##  3  1.12   3.5    30   135 Yes
##  4  1.12   3      40   120 Yes
##  5  1.12   3      50   135 Yes
##  6  1.12   3      30   150 No
##  7  1.12   4      40   135 Yes
##  8  1.12   4      50   150 No
##  9  1.12   4      30   120 No
## 10  1.12   3.5    40   120 Yes
## # … with 17 more rows
```

```
res_logit <- glm(Perc.above.average ~ ., data = d3, family = binomial(link
= logit))
res_null <- glm(Perc.above.average ~1,data=d3, family=binomial)
summary(res_logit)
```

```
##
## Call:
## glm(formula = Perc.above.average ~ ., family = binomial(link = logit),
##     data = d3)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.5006  -0.7791  -0.5718   1.0578   2.0726
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  176.94275  304.05177   0.582   0.5606
## A           -147.95770  270.01225  -0.548   0.5837
## B             -2.16577    1.14987  -1.883   0.0596 .
## C             -0.02959    0.05400  -0.548   0.5837
## D             -0.01976    0.03606  -0.548   0.5838
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 35.594  on 26  degrees of freedom
## Residual deviance: 30.747  on 22  degrees of freedom
## AIC: 40.747
##
## Number of Fisher Scoring iterations: 4
```

Finding 9

Logistic regression also suggests that B is the only significant variable.

```
res_logit2 <- glm(Perc.above.average ~ B, data = d3, family = binomial(lin
k = logit))
summary(res_logit2)
```

```
##
## Call:
## glm(formula = Perc.above.average ~ B, family = binomial(link = logit),
##     data = d3)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.3599  -0.9264  -0.5882   1.0053   1.9182
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    6.677      3.845   1.736   0.0825 .
## B             -2.086      1.120  -1.863   0.0625 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 35.594  on 26  degrees of freedom
## Residual deviance: 31.612  on 25  degrees of freedom
## AIC: 35.612
##
## Number of Fisher Scoring iterations: 4
```

Let us perform analysis of “deviance” to check whether the model is different than the null model.

```
anova(res_logit2,res_null)
```

```
## Analysis of Deviance Table
##
## Model 1: Perc.above.average ~ B
## Model 2: Perc.above.average ~ 1
##   Resid. Df Resid. Dev Df Deviance
## 1        25     31.612
## 2        26     35.594 -1  -3.9823
```

We can reject the “H_o:Proposed model and null model and equivalent” and conclude that our model is better than the null model.

```
res_logit2 %>% step()
```

```
## Start:  AIC=35.61
## Perc.above.average ~ B
##
##        Df Deviance    AIC
## <none>      31.612 35.612
## - B     1   35.594 37.594
```

```
##
## Call:  glm(formula = Perc.above.average ~ B, family = binomial(link = l
ogit),
##     data = d3)
##
## Coefficients:
## (Intercept)            B
##       6.677       -2.086
##
## Degrees of Freedom: 26 Total (i.e. Null);  25 Residual
## Null Deviance:       35.59
## Residual Deviance: 31.61     AIC: 35.61
```

Finding 10

I again applied step() function and it suggests us to use model with B as the only predictor.

7. How good does your model fit to data? Explain.

To check goodness of fit in logistic regression models, we use Hosmer-Lemeshow (H-L) test.

```
number_of_intervals <- 3
library(modelr)
d3 %>%
  add_predictions(res_logit2, var = "pred", type = "response") %>%
  mutate(pred_cut = cut(pred, breaks = c(0, quantile(pred, prob = seq(numb
er_of_intervals-1)/number_of_intervals), 1))) %>%
  group_by(pred_cut) %>%
  summarize(N = n(),
            S = sum(Perc.above.average == "Yes"),
            pbar = median(pred), .groups = "drop") %>%
  mutate(mean = N*pbar, sd = sqrt(mean*(1-pbar)),
         Z = (S - mean)/sd) %>% arrange(pbar) %>%
  summarize(HL = sum(Z^2), .groups = "drop") %>%
  extract2("HL") -> HL

cat("HL pvalue", pchisq(HL, df = number_of_intervals - 1, lower.tail = FAL
SE), "\n")
```

```
## HL pvalue 0.740556
```

Finding 11

P-value of the H-L test is really high. We can not reject the significance of the proposed model. So we can say that model that we propose (res_logit2) works well.

Let us just perform H-L test with another number of intervals to make sure our model is correct.

```
number_of_intervals <- 4
d3 %>%
  add_predictions(res_logit2, var = "pred", type = "response") %>%
  mutate(pred_cut = cut(pred, breaks = c(0, quantile(pred, prob = seq(numb
er_of_intervals-1)/number_of_intervals), 1))) %>%
  group_by(pred_cut) %>%
  summarize(N = n(),
            S = sum(Perc.above.average == "Yes"),
            pbar = median(pred), .groups = "drop") %>%
  mutate(mean = N*pbar, sd = sqrt(mean*(1-pbar)),
         Z = (S - mean)/sd) %>% arrange(pbar) %>%
  summarize(HL = sum(Z^2), .groups = "drop") %>%
  extract2("HL") -> HL

cat("HL pvalue", pchisq(HL, df = number_of_intervals - 1, lower.tail = FAL
SE), "\n")
```

```
## HL pvalue 0.8962703
```

Again, we can not reject the significance of the proposed model. We can say that model that we propose (res_logit2) works well.

8. How does the percentage of unacceptable products change with temperature (D)? Does the effect of temperature (D) change with the other factors?

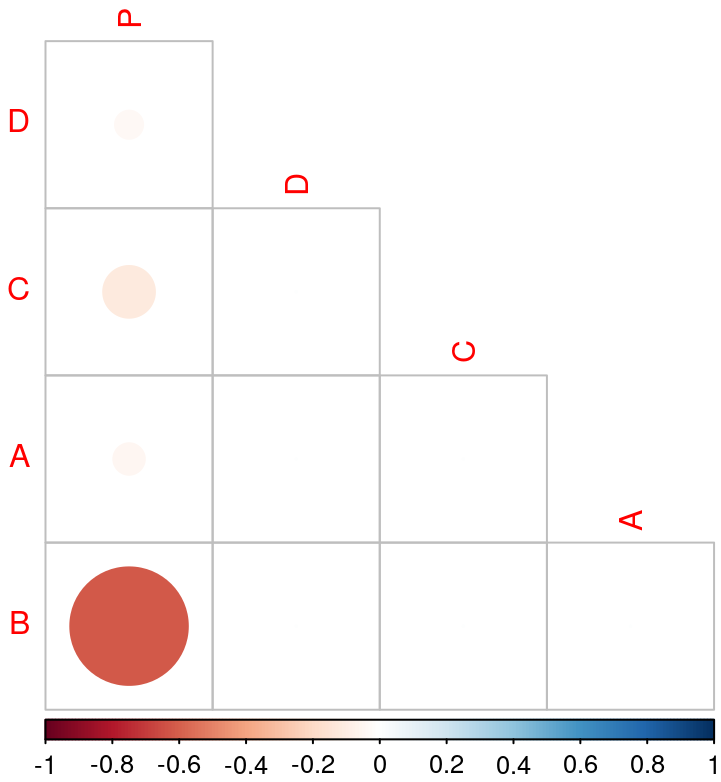
```
logistic.with.onlyD <- glm(Perc.above.average ~ D, data = d3, family = bin
omial(link = logit))
summary(linmod.onlyD)
```

```
##
## Call:
## lm(formula = P ~ D, data = d2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.29529 -0.14605 -0.02474  0.14578  0.42623
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.7004947  0.4681398   5.769 5.18e-06 ***
## D           -0.0006089  0.0034535  -0.176    0.861
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2198 on 25 degrees of freedom
## Multiple R-squared:  0.001242,   Adjusted R-squared:  -0.03871
## F-statistic: 0.03108 on 1 and 25 DF,  p-value: 0.8615
```

Finding 12

Percentage of unacceptable products does not change with D as the variable is not significant again(p-value>0.05).

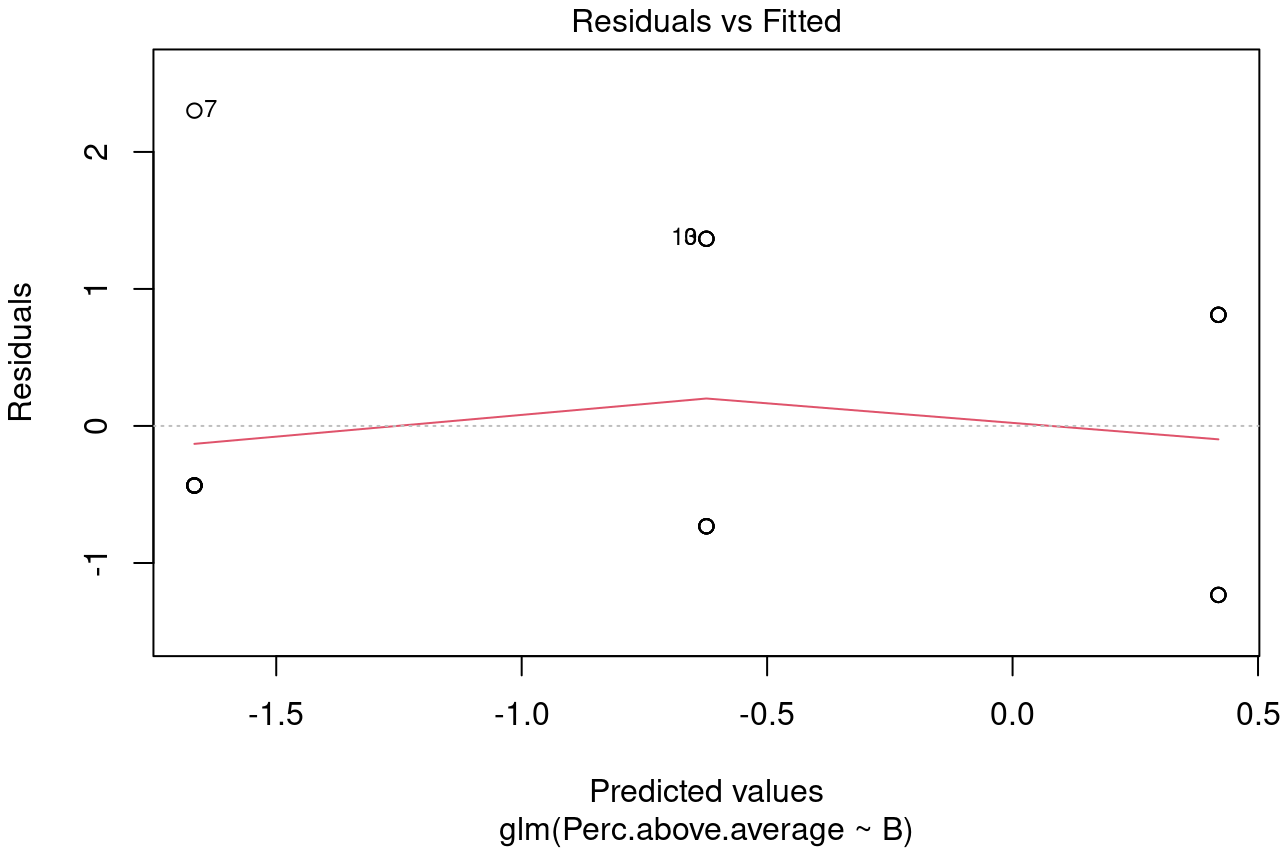
```
d2 %>%
  select_if(is.numeric) %>%
  cor(use = "pairwise.complete.obs") %>%
  corrplot(type = "lower", diag = FALSE, order = "hclust")
```

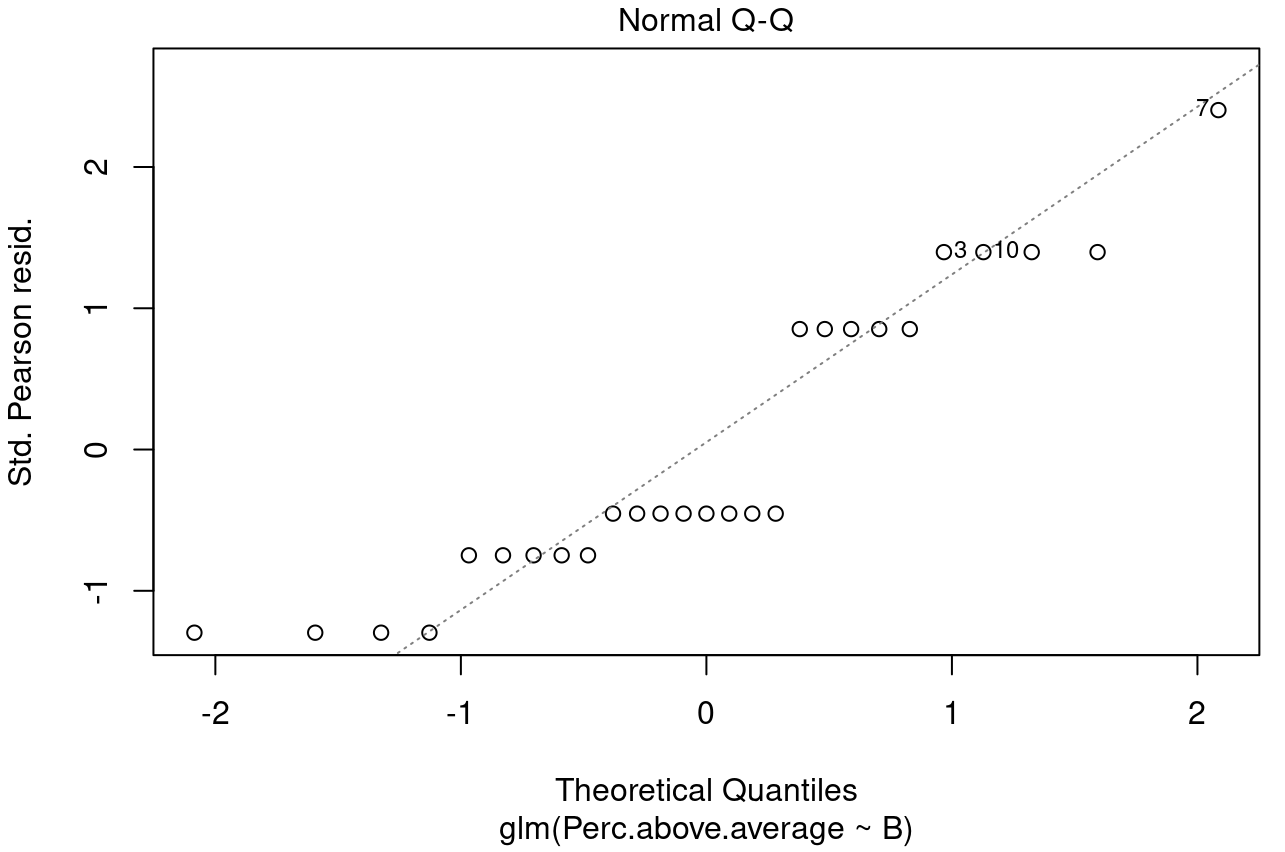


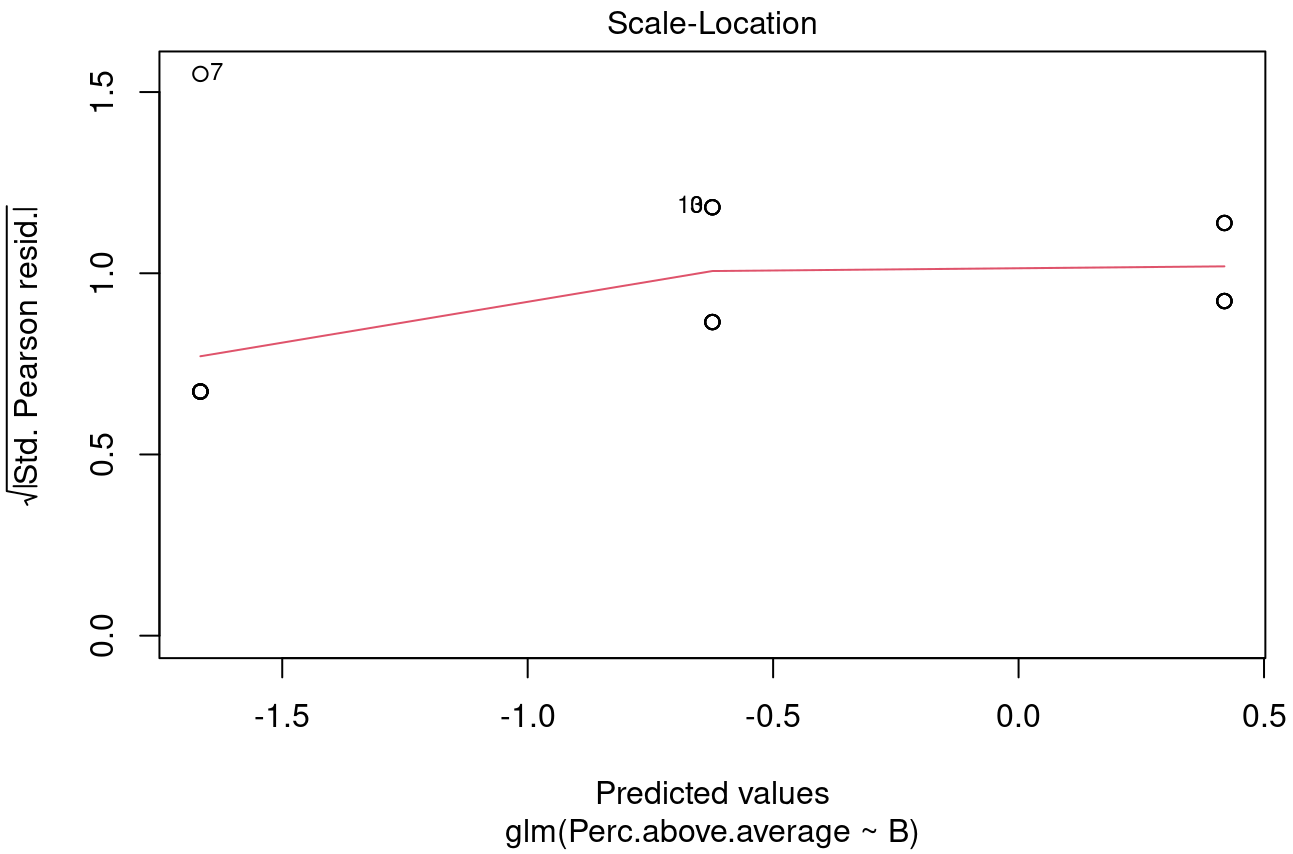
Again we conclude that D has nothing to do with A,B,C and P.

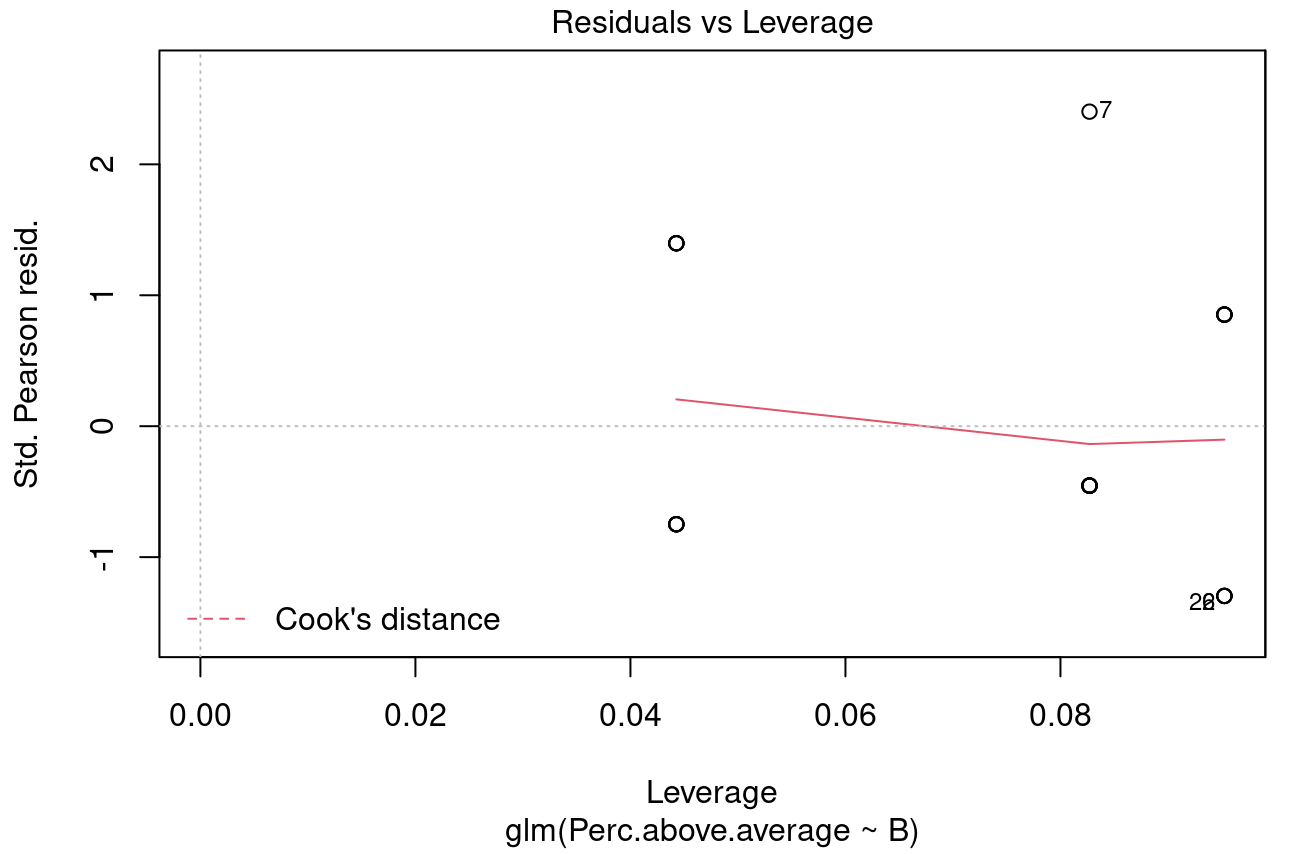
9. Are there influential points? If there are, how does your model change when you remove them (I am not suggesting that you should remove them, but I just want to know how significant they are)?

```
plot(res_logit2)
```


Residuals vs Fitted
Residuals
Predicted values
glm(Perc.above.average ~ B)
7
10
2
1
0
-1
-1.5
-1.0
-0.5
0.0
0.5


Normal Q-Q
Std. Pearson resid.
Theoretical Quantiles
glm(Perc.above.average ~ B)
7
3
10
2
1
0
-1
-2
-1
0
1
2


Scale-Location
√|Std. Pearson resid.|
0.0
0.5
1.0
1.5
-1.5
-1.0
-0.5
0.0
0.5
Predicted values
glm(Perc.above.average ~ B)


Residuals vs Leverage
Std. Pearson resid.
-1
0
1
2
Cook's distance
0.00
0.02
0.04
0.06
0.08
Leverage
glm(Perc.above.average ~ B)

7 and 28 are influential observations that have high leverage and Cook's distance.

Synthesis [suggested time: 20 minutes]

10. Which model is better? Your best linear model or your best binomial model? Show your work.

I want to look at Test MSE of the linear model and create a confusion matrix for the logistic regression to decide for this. First let us split data set into training data set and test data set. This will make MSE calculation for the linear model more accurate.

```
trindexes <- (1:nrow(d) %>% sample())[1:round(3/4*nrow(d))]
d.training <- d %>% slice(trindexes)
d.test <- d %>% slice(setdiff(1:nrow(d),trindexes))
```

```
mse.linear <- mean(d.test$P-(exp(predict(linmod2,newdata=d.test))^2)) #exp
onentiating back to find predict real P value
```

```
## Warning in predict.lm(linmod2, newdata = d.test): prediction from a ran
k-
## deficient fit may be misleading
```

```
mse.linear
```

```
## [1] 8.098517
```

MSE for the linear model is 8.30.

For logistic regression;

```
glm.probs=predict(res_logit2,type="response")
glm.pred=rep("No",nrow(d))
glm.pred[glm.probs >.5]="Yes" # I am using 0.5 as threshold, there is no t
ime left for ROC/AUC work.

table(glm.pred,d3$Perc.above.average)
```

```
##
## glm.pred No Yes
##      No  13   5
##      Yes  4   5
```

```
mean(glm.pred==d3$Perc.above.average)
```

```
## [1] 0.6666667
```

Logistic model gives 66% accuracy. If I had time, I would study its ROC and AUC to improve its threshold and achieve a better model.

Final interpretation

It was really hard to compare a linear model and a logistic model because their dynamics, for example, how they define the response variable, are different. It is better to use logistic regression when the response is binary outcome. Other than that, I think this decision is up to the analyst.
