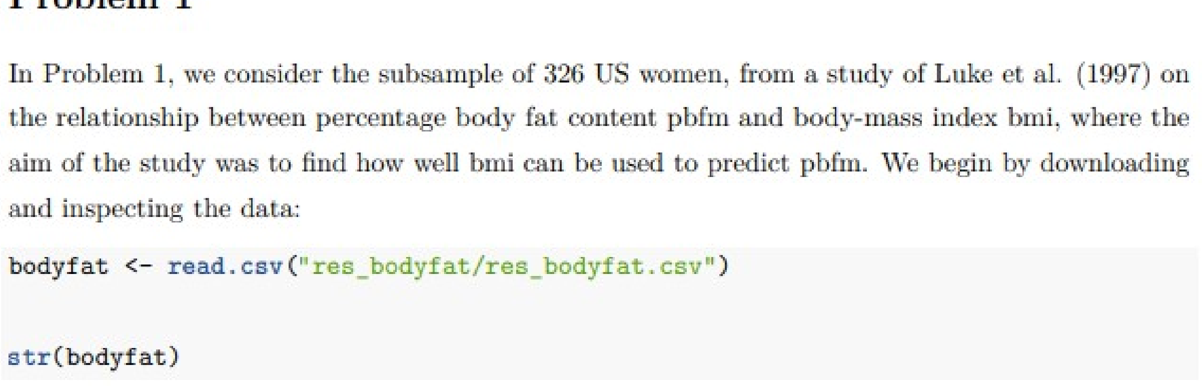
Solution Manual
Mandatory assignment 1
STK2100 Spring 2020



Problem 1

In Problem 1, we consider the subsample of 326 US women, from a study of Luke et al. (1997) on the relationship between percentage body fat content pbfm and body-mass index bmi, where the aim of the study was to find how well bmi can be used to predict pbfm. We begin by downloading and inspecting the data:

```
bodyfat <- read.csv("res_bodyfat/res_bodyfat.csv")


str(bodyfat)
```

```
## 'data.frame':    327 obs. of  3 variables:
##  $ bmi         : num  30.6 30.2 24.7 34.4 21.4 ...
##  $ pbfm        : num  38.8 45.5 34.8 45.7 31.1 ...
##  $ msbp_missing: int  0 0 0 0 0 0 0 0 0 0 ...
```

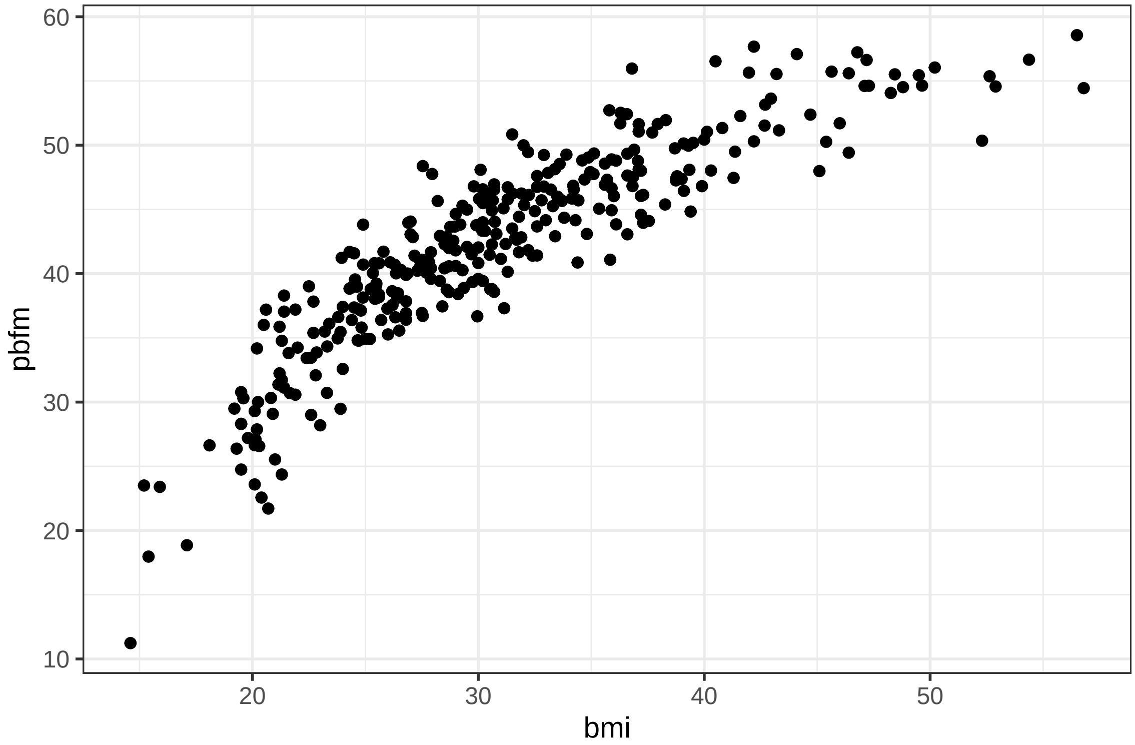
```
summary(bodyfat)
```

```
##       bmi             pbfm        msbp_missing
##  Min.   :14.60   Min.   :11.23   Min.   :0.000000
##  1st Qu.:25.28   1st Qu.:37.83   1st Qu.:0.000000
##  Median :30.10   Median :42.83   Median :0.000000
##  Mean   :30.94   Mean   :42.20   Mean   :0.003058
##  3rd Qu.:35.87   3rd Qu.:47.49   3rd Qu.:0.000000
##  Max.   :56.80   Max.   :58.56   Max.   :1.000000
```

a) Simple linear model

Before building the first model, we simply plot the data in a scatter plot. The plot shows a pattern of positive correlation between bmi and $pbfm$, where higher bmi values correspond to higher $pbfm$ values, and the relationship is somewhat linear. However, we can notice that the points constitute a slightly concave shape, particularly in that increases in bmp for higher values do not seem to lead to as large increases in $pbfm$ as for smaller values.

```
fig1 <- bodyfat %>%
  ggplot() +
  aes(x = bmi, y = pbfm) +
  geom_point() +
  theme_bw()
fig1
```



Then, we fit our first model, which is a simple linear model, and examine the results. As could be expected from the scatter plot, the coefficient for bmi is positive, where the expected increase in pbfm from one unit increase in bmi is 0.885. Furthermore, the intercept is at a $pbfm$ value of 14.828. Both of the coefficients are significant at less than 0.1 percent significance level. Note, however, that the intercept value, is the value of $pbfm$ when bmi is 0 – which will never occur. Thus, the intercept is the $pbfm$ when bmi is at the hypothetical value of 0.

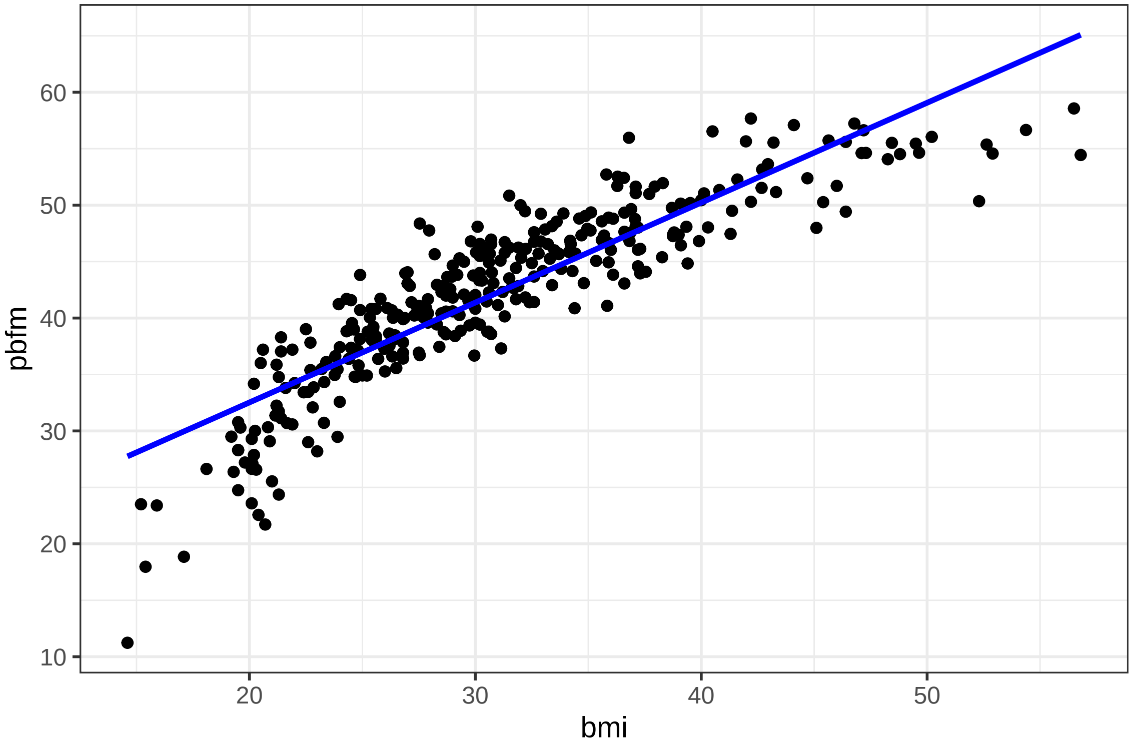
```
mod1 <- bodyfat %>%
  lm(pbfm ~ bmi, data = .)
summary(mod1)
```

```
##
## Call:
## lm(formula = pbfm ~ bmi, data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.5116  -2.0714   0.4083   2.4994   9.1758
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 14.82772    0.82671   17.94   <2e-16 ***
## bmi          0.88481    0.02589   34.17   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.702 on 325 degrees of freedom
## Multiple R-squared:  0.7823, Adjusted R-squared:  0.7816
## F-statistic:  1168 on 1 and 325 DF,  p-value: < 2.2e-16
```

We can examine graphically how well the simple linear model fits the data, see plot below. Overall, it is quite good, however, it does not entirely capture the curvature of the points.

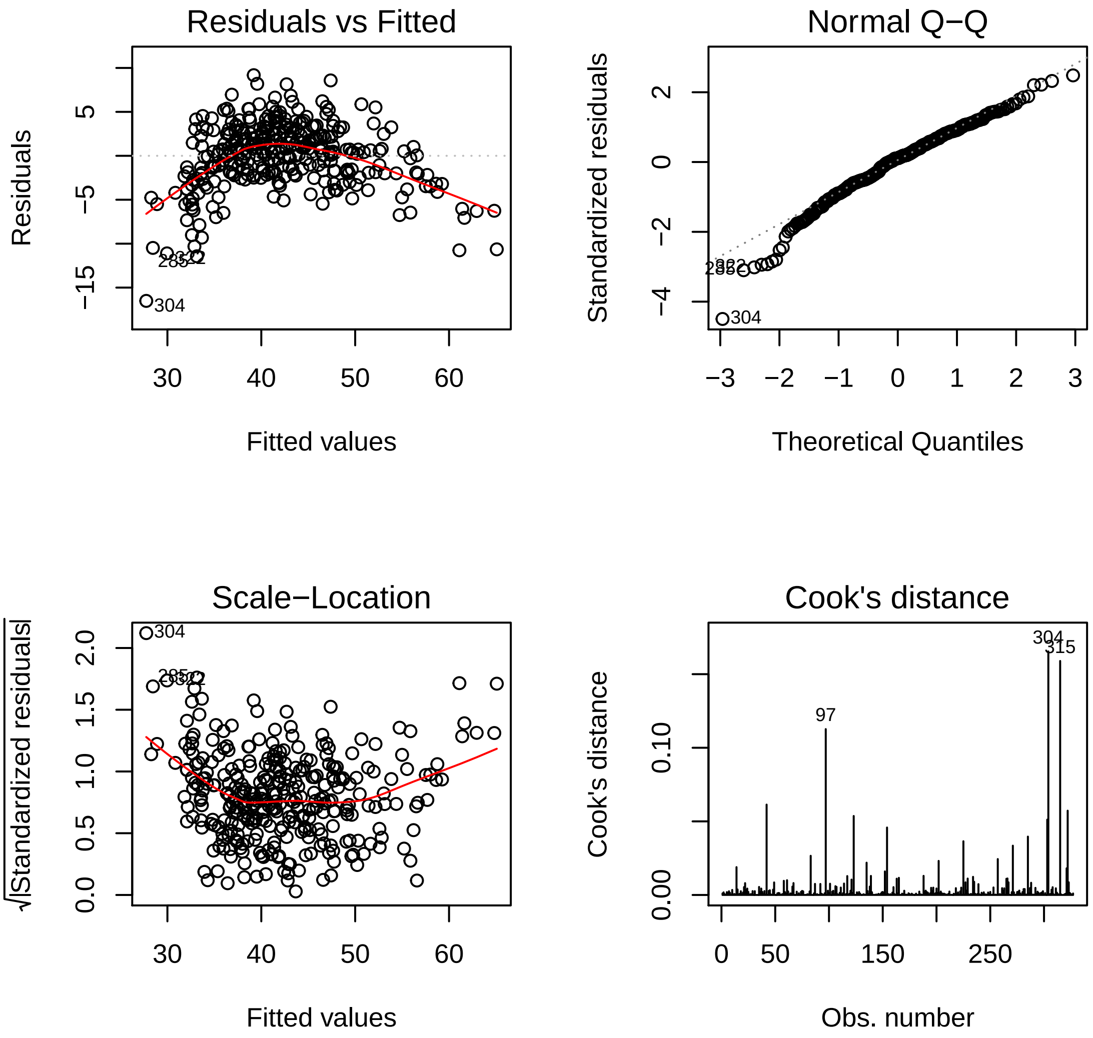
```
bodyfat1 <- bodyfat %>%
  mutate(pred = predict(mod1, bodyfat))

fig1_mod1 <- bodyfat1 %>%
  ggplot() +
  geom_point(aes(x = bmi, y = pbfm)) +
  geom_line(aes(x = bmi, y = pred), size = 1, col = "blue") +
  theme_bw()
fig1_mod1
```



Performing a graphical diagnostic analysis further reveals signs of heteroskedasticity, as shown by the Ascombe plot (top-left plot) below. Under the assumption of constant variance for the error term, the Ascombe plot should not show any clear pattern, but it does so in the plot below. Hence, the assumption about homoskedasticity is violated. The scale-location plot, where the residuals are standardized, also supports this conclusion. Furthermore, the quantile-quantile plot (top-right plot) shows signs of a non-linear distribution of the residuals, especially in lower-left tail. Under the assumption of normally distributed residuals, the theoretical quantiles (shown on the x-axis) should equal the standardized residuals (shown on the y-axis), which is not entirely satisfied in the Q-Q plot below. When inspecting the Cook's distance plot, there does not seem to be any observations that are particularly influential, as the Cook's distance values on the y-axis are well below 0.5.

```r
par(mfrow = c(2, 2))
plot(mod1, which = 1:4, sub.caption = "")
```



b) Logarithmic transformation and quadratic terms

Both a logarithmic and quadratic transformation could improve the model as it would allow for a non-linear relationship between bmi and $pbfm$ (while still preserving the linear model) and thus capture the curvature we can observe in the scatter plot. Logarithmic transformations are often used with intrinsically positive values, which is the case for bmi, and according to Azzalini and Scarpa (2012) often tend to correct heteroskedasticity. Furthermore, it would be relevant to consider the theoretical relationship between bmi and $pbfm$: we assume higher values of bmi to correspond to higher values of $pbfm$. While a logarithmic transformation would capture that relationship, a polynomial of degree 2 would eventually lead to the model predicting decreasing values of $pfmb$ as bmi increases. Hence, a logistic transformation of bmi might be more appropriate than including a quadratic term. We will now build both models and report the results, including diagnostics plots.

Logarithmic explanatory variable

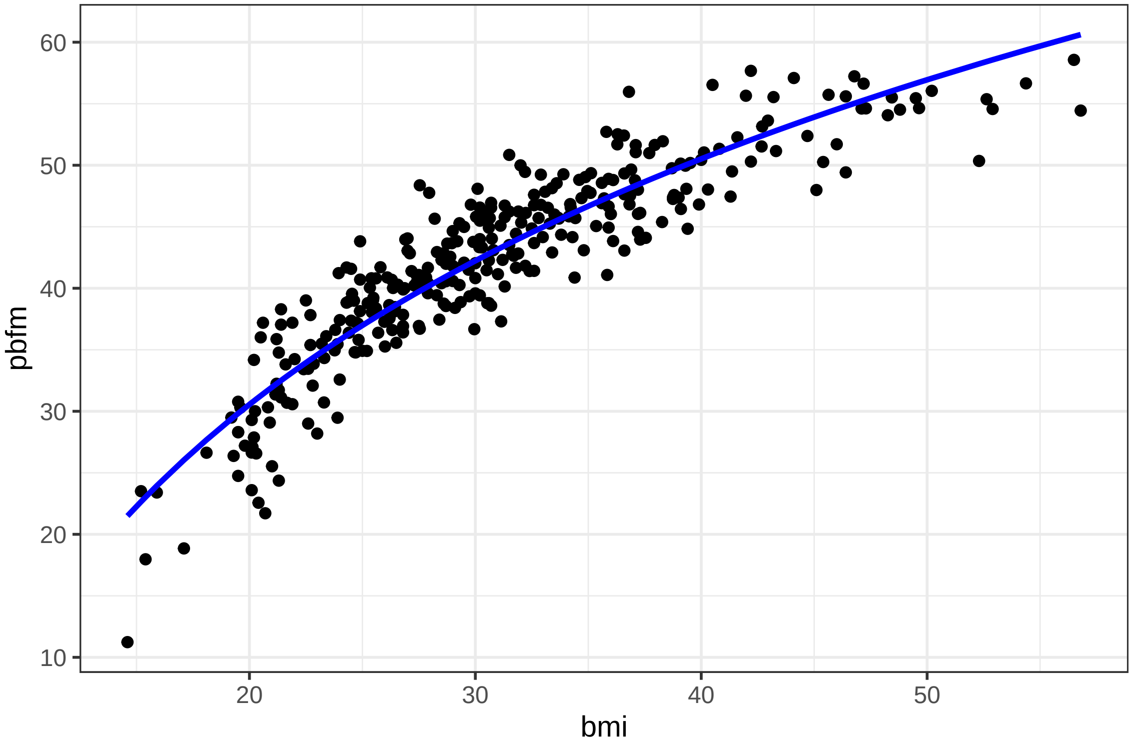
The results of the model with a logarithmic transformation of bmi is shown below. We also plot the model against the true values, to allow us to examine the fit graphically. From the plot we can notice that there is an improvement compared to the simple linear model, but the model seems to slightly overestimate the $pbfm$ value for the highest values of bmi, and perhaps also for the lower values. Regarding the estimated coefficients, we see that a 1 percent increase in bmi results in an increase of about $28.8/100 = 0.288$ in $pbfm$ with this model. As we would never observe a bmi value of 0, the intercept of -55.7 (which is not a possible value for $pbfm$) is again only a hypothetical value. Finally, both coefficients are diffent from zero at a significance level of less than 0.1 percent.

```r
mod2 <- bodyfat %>%
  lm(pbfm ~ log(bmi), data = .)
summary(mod2)
```

```
##
## Call:
## lm(formula = pbfm ~ log(bmi), data = .)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.2548  -2.0453   0.1026   2.1238   8.6029
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -55.7327     2.3337  -23.88   <2e-16 ***
## log(bmi)     28.8031     0.6845   42.08   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.125 on 325 degrees of freedom
## Multiple R-squared:  0.8449, Adjusted R-squared:  0.8444
## F-statistic:  1771 on 1 and 325 DF,  p-value: < 2.2e-16
```

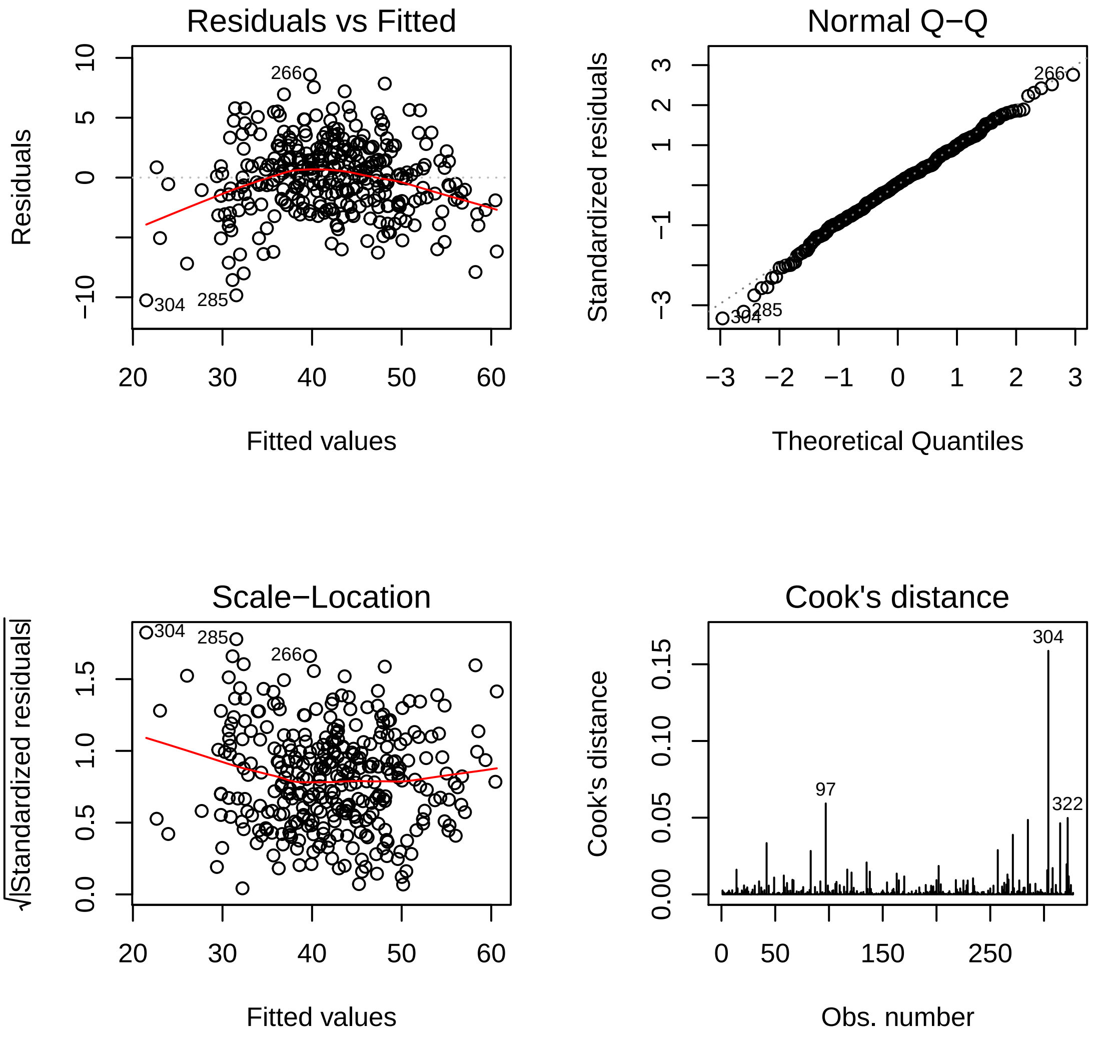
```
bodyfat2 <- bodyfat %>%
  mutate(pred = predict(mod2, bodyfat))

fig1_mod2 <- bodyfat2 %>%
  ggplot() +
  geom_point(aes(x = bmi, y = pbfm)) +
  geom_line(aes(x = bmi, y = pred), size = 1, col = "blue") +
  theme_bw()
fig1_mod2
```



When examining the diagnostics plot below, there seems to be an improvement in normality assumption of the residuals (the Q–Q plot). However, although slightly improved compared to the simple linear model, there are still signs of heteroskedasticity in Ascombe plot. Hence, the logarithmic transformation of *bmi* has not solved the problem of heteroskedasticity.

```r
par(mfrow = c(2, 2))
plot(mod2, which = 1:4, sub.caption = "")
```



Quadratic terms

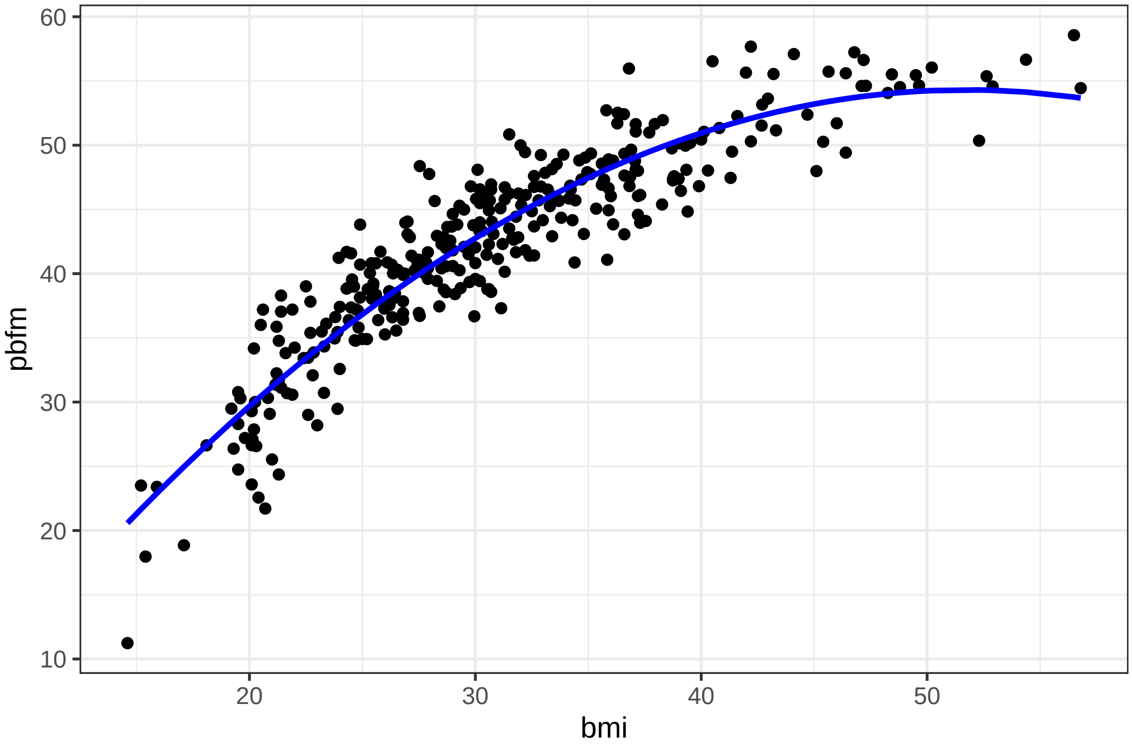
We now turn to expanding the simple linear model with a quadratic term, and the regression results and corresponding model plot are both shown below. First, we can notice that the coefficents of both *bmi* and *bmi*ˆ2 are significant, as is the coefficient of the intercept. From the plot we can observer that, again, there seems to be an improvement in the fit compared to the simple linear model. Compared to the logarithmic model, the quadratic model also seems to perform better in predicting especially high values of *pbfm*. However, as we expected, we can notice how the curve starts to turn downwards after *bmi* values of around 50, predicting slightly decreasing *pbfm* values as *bmi* increases – which theoretically would not make sense. (This effect is due to the coefficient of the quadratic term being negative.)

```r
mod3 <- bodyfat %>%
  lm(pbfm ~ bmi + I(bmi^2), data = .)
summary(mod3)
```

```
##
## Call:
## lm(formula = pbfm ~ bmi + I(bmi^2), data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3403 -1.9246  0.1433  1.8665  8.3780
##
## Coefficients:
##              Estimate Std. Error t value    Pr(>|t|)
## (Intercept) -11.17790    2.16977  -5.152 0.000000449 ***
## bmi           2.53223    0.13229  19.142     < 2e-16 ***
## I(bmi^2)     -0.02448    0.00194 -12.617     < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.036 on 324 degrees of freedom
## Multiple R-squared:  0.854,  Adjusted R-squared:  0.8531
## F-statistic: 947.8 on 2 and 324 DF,  p-value: < 2.2e-16
```

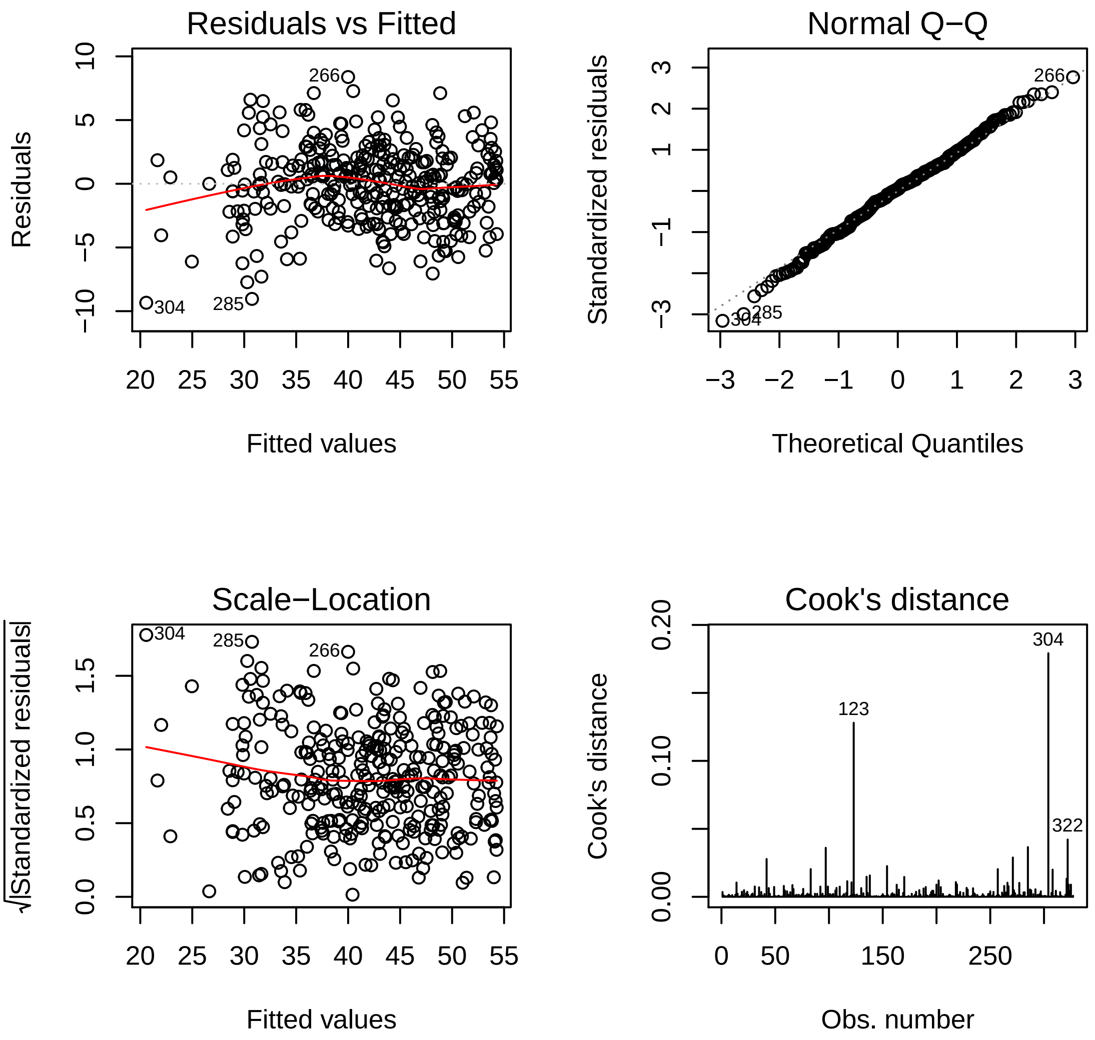
```r
bodyfat3 <- bodyfat %>%
  mutate(pred = predict(mod3, bodyfat))

fig1_mod3 <- bodyfat3 %>%
  ggplot() +
  geom_point(aes(x = bmi, y = pbfm)) +
  geom_line(aes(x = bmi, y = pred), size = 1, col = "blue") +
  theme_bw()
fig1_mod3
```



Examining the diagnostics plots below, there seems to be a further reduction in heteroskedasticity with this model, compared to the logarithmic model. Although there might still be reason to be concerned about heteroskedasticity, its presence is less severe than in the previous models. The Q–Q plot is also more satisfying compared to the simple linear model. Finally, we can notice that some observations are slightly more influential, but the Cook's distance is still well below 0.5 for all observations.

```
par(mfrow = c(2, 2))
plot(mod3, which = 1:4, sub.caption = "")
```



Final note

The choice between a logarithmic transformation and the inclusion of a quadratic term also depends on the range of bmi you want to apply the model on. In the data sample we have here, the values for bmi range from 15 to 57. In this range, a quadratic term appears to fit the observed data slightly better than a logarithmic transformation. However, for bmi values outside this range – especially for particularly high values – predicting $pbfm$ with a quadratic model would not give sensible results (decreasing $pbfm$!). However, one should anyway be careful in applying the model to predict $pbfm$ for bmi data far outside the range used to fit the model, as we do not know if the relationship between $pbfm$ and bmi is the same then. Hence, given that this model is built to predict $pbfm$ for someone with bmi between around 15 and 60, a quadratic term might be preferable to a logarithmic transformation.

c) Cross-validation

To examine if a polynomial of a higher order might be beneficial, we apply cross-validation using the *train* function in the *caret* pacakge. In particular, we set the method to be leave-one-out cross-validation (LOOCV), which implies that n–1 observations are used to fit the model, while the remaining observation is used for testing (Azzalini & Scarpa, 2012). From the plot below, we see that a polynomial of degree 4 appears to yield the lowest mean squared error (MSE).

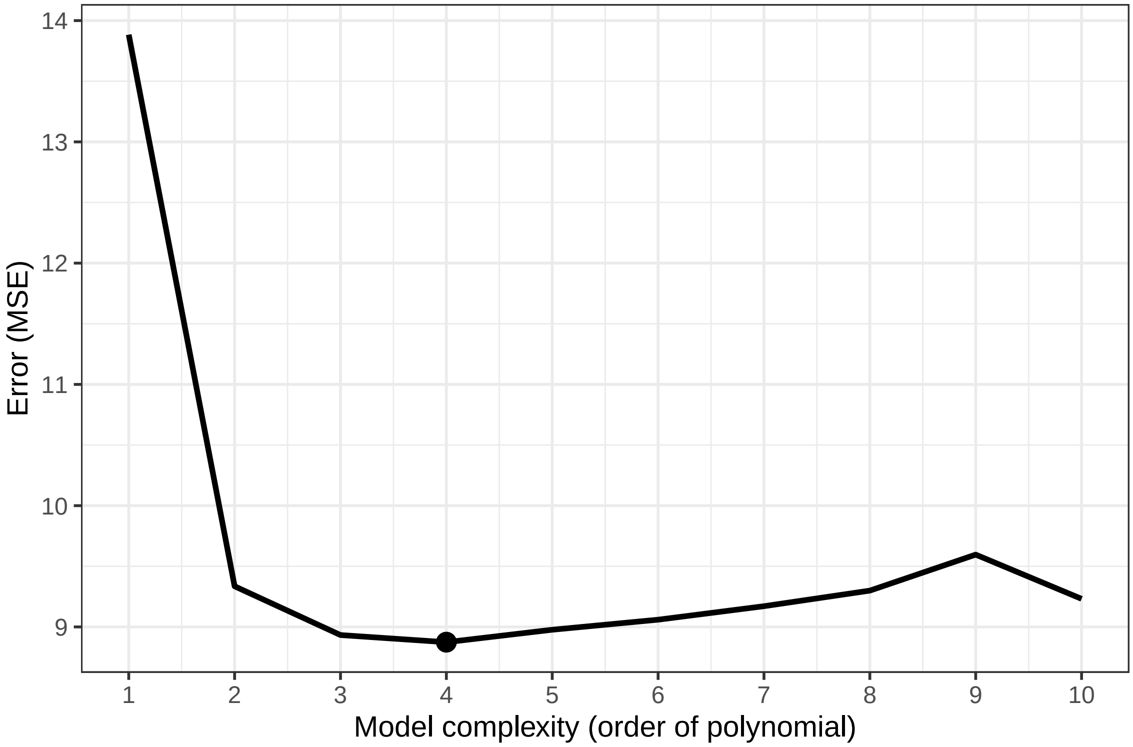
```
train_control <- trainControl(method = "LOOCV")

MSE <- numeric(10)
for (p in 1:10){
  formula <- bquote(pbfm ~ poly(bmi, .(p)))
  model <- train(as.formula(formula),
                 data = bodyfat,
                 method = "lm",
                 trControl = train_control)
  MSE[p] <- model$results$RMSE^2
}
```

```
min_cv <- which(MSE == min(MSE))
min_cv
```

```
## [1] 4
```

```
fig_cv <- ggplot() +
  aes(x = 1:10, y = MSE) +
  geom_line(size = 1) +
  geom_point(aes(x = min_cv, y = MSE[min_cv]), size = 3) +
  scale_x_continuous(breaks = c(1:length(MSE))) +
  theme_bw() +
  labs(x = "Model complexity (order of polynomial)",
       y = "Error (MSE)")
fig_cv
```



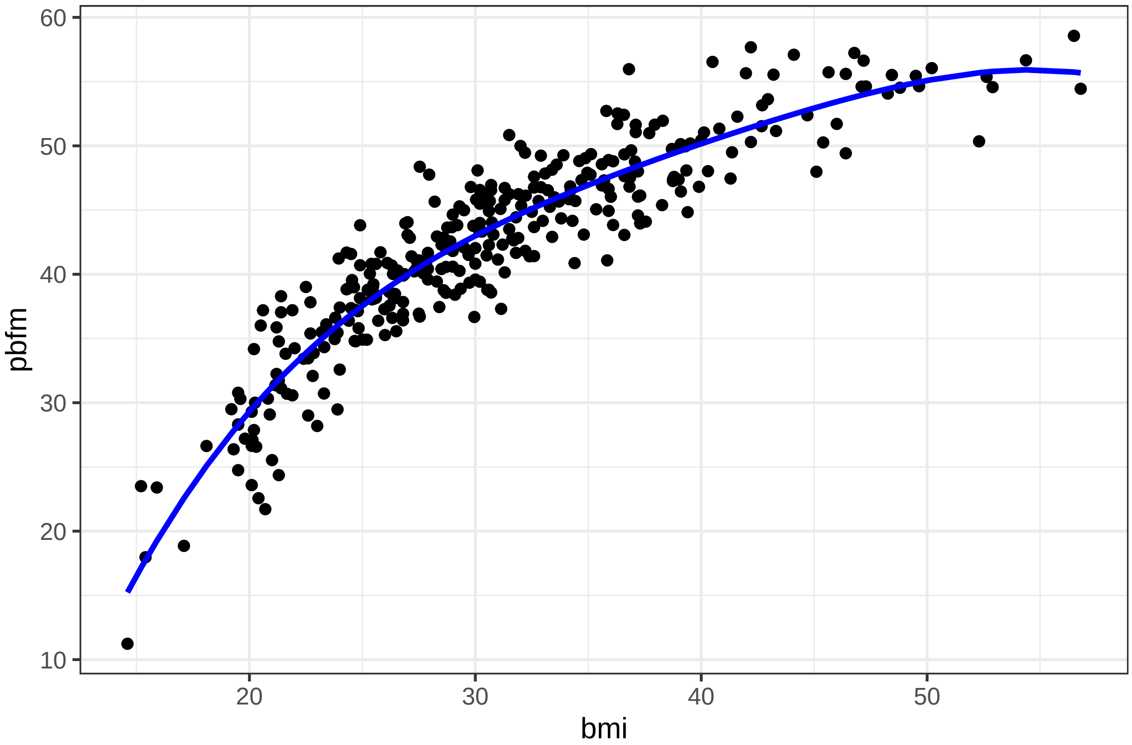
We can also report the related model, which is done below. As before, we also include a plot of the model to be able to examine its fit graphically, in addition to the diagnostics plots.

```
mod4 <- bodyfat %>%
  lm(pbfm ~ poly(bmi, min_cv), data = .)
summary(mod4)
```

```
##
## Call:
## lm(formula = pbfm ~ poly(bmi, min_cv), data = .)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.9670 -1.8763  0.0443  1.8563  7.8093
##
## Coefficients:
##                     Estimate Std. Error t value  Pr(>|t|)
## (Intercept)           42.200      0.163 258.956   < 2e-16 ***
## poly(bmi, min_cv)1   126.529      2.947  42.937   < 2e-16 ***
## poly(bmi, min_cv)2   -38.312      2.947 -13.001   < 2e-16 ***
## poly(bmi, min_cv)3    12.181      2.947   4.134 0.0000456 ***
## poly(bmi, min_cv)4   -6.536      2.947  -2.218    0.0273 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.947 on 322 degrees of freedom
## Multiple R-squared:  0.8634, Adjusted R-squared:  0.8617
## F-statistic: 508.7 on 4 and 322 DF,  p-value: < 2.2e-16
```

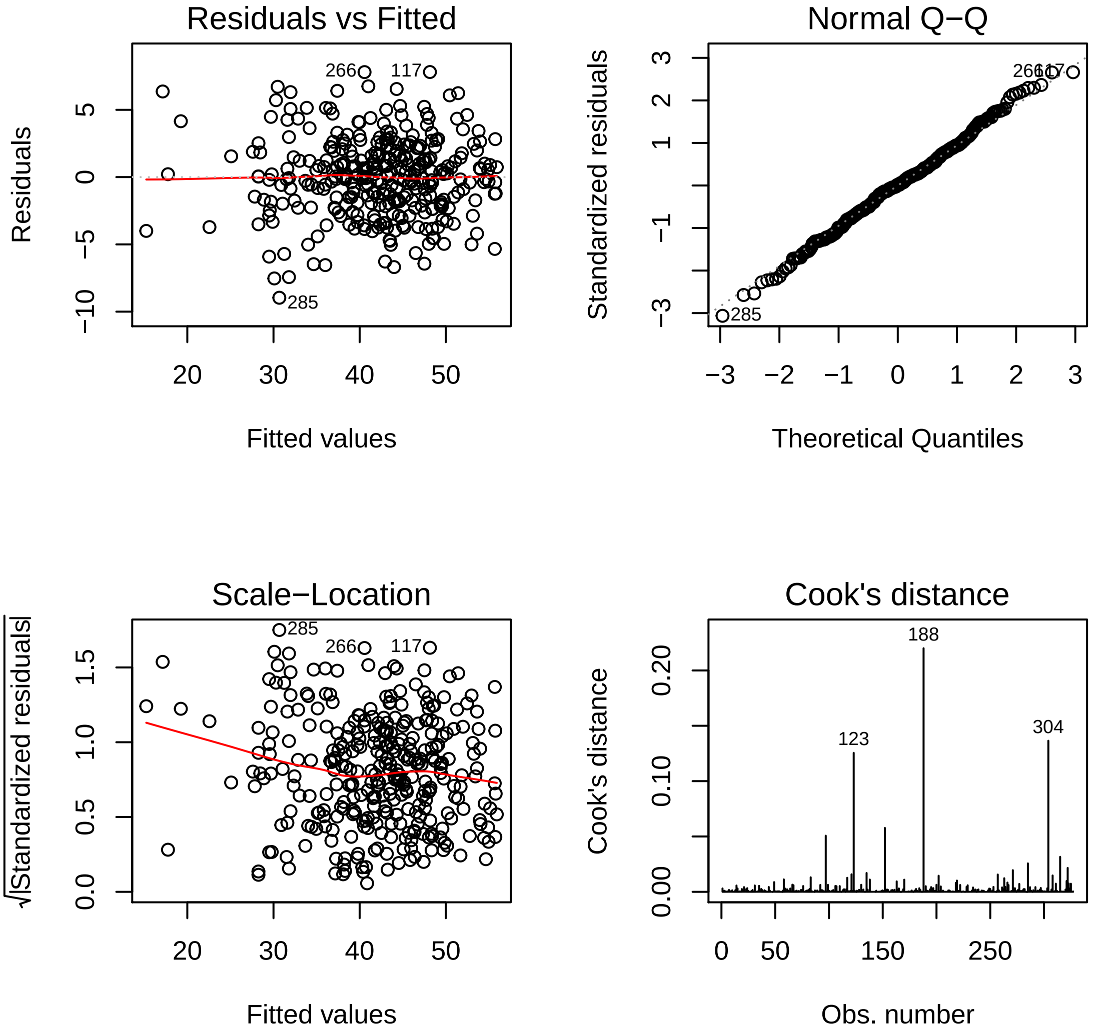
```
bodyfat4 <- bodyfat %>%
  mutate(pred = predict(mod4, bodyfat))

fig1_mod4 <- bodyfat4 %>%
  ggplot() +
  geom_point(aes(x = bmi, y = pbfm)) +
  geom_line(aes(x = bmi, y = pred), size = 1, col = "blue") +
  theme_bw()
fig1_mod4
```



Diagnostics plot:

```
par(mfrow = c(2, 2))
plot(mod4, which = 1:4, sub.caption = "")
```



d) Information criteria

Another technique for model selection is to use an information criterion (IC) which applies a penalty for increasing the number of parameters. Here, both Akaike information criterion (AIC), which uses the penalty $2p$, and Bayesian information criterion (BIC), which uses the penalty $p \log(n)$, is employed, and the results are plotted below. We see that model selection by AIC yields the same result as LOOCV, with AIC being at its lowest with degree 4 of the polynomial. The BIC, on the other hand, penalises model complexity more than both AIC and LOOCV, reflected in BIC being the lowest for the model with degree 3 of the polynomial. According to Hastie et al. (2009), for very large sample sizes AIC tends to choose too complex models, while for smaller sample sizes, BIC tends to choose too simple models due to its penalty on complexity. Therefore, with $N = 327$ observations in our dataset, I would choose the model where IC is lowest based on AIC, that is, the model with degree 4 of the polynomial.

```r
AIC_BIC <- data.frame(p = 1:10,
                      AIC = numeric(10),
                      BIC = numeric(10))

for (p in AIC_BIC$p){
  formula <- bquote(pbfm ~ poly(bmi, .(p)))
  AIC_BIC$AIC[p] <- AIC(lm(as.formula(formula), data = bodyfat))
  AIC_BIC$BIC[p] <- BIC(lm(as.formula(formula), data = bodyfat))
}
```

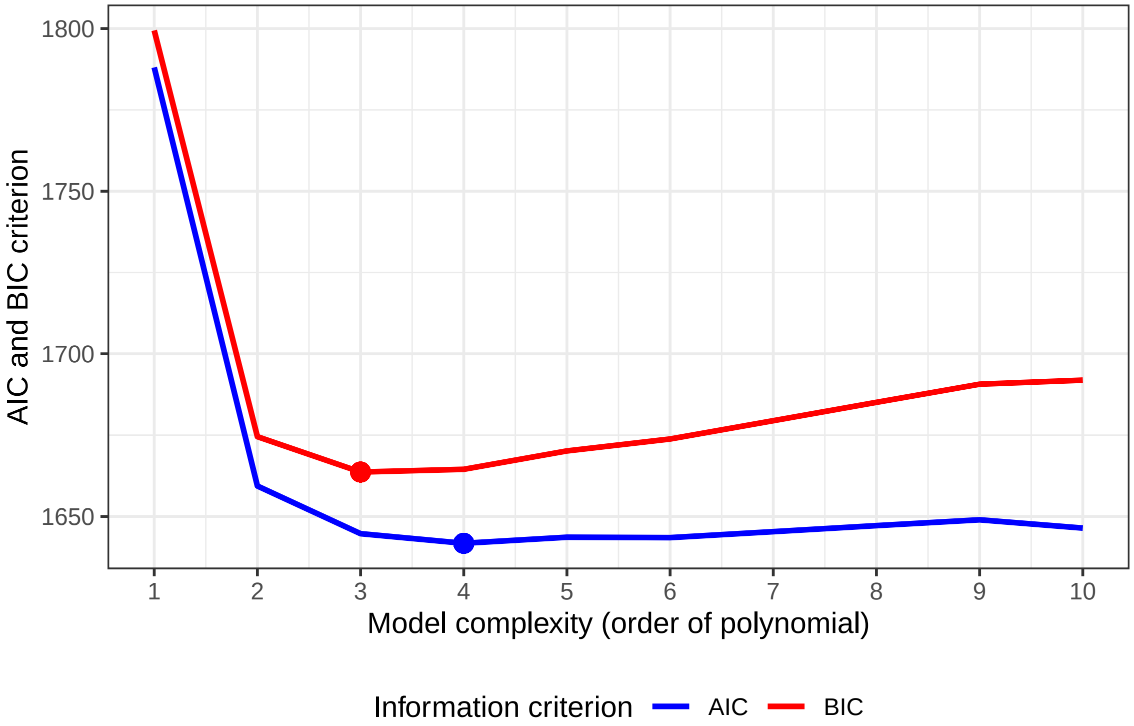
```r
min_AIC <- which(AIC_BIC$AIC == min(AIC_BIC$AIC))
min_AIC
```

```
## [1] 4
```

```r
min_BIC <- which(AIC_BIC$BIC == min(AIC_BIC$BIC))
min_BIC
```

```
## [1] 3
```

```r
fig_AIC_BIC <- AIC_BIC %>%
  gather(`Information criterion`, Value, AIC, BIC) %>%
  ggplot() +
  aes(x = p, y = Value, col = `Information criterion`) +
  geom_line(size = 1) +
  geom_point(aes(x = min_AIC, y = AIC_BIC$AIC[min_AIC]), col = "blue", size = 3) +
  geom_point(aes(x = min_BIC, y = AIC_BIC$BIC[min_BIC]), col = "red", size = 3) +
  scale_x_continuous(breaks = c(1:10)) +
  scale_colour_manual(values = c("blue", "red")) +
  theme_bw() +
  labs(x = "Model complexity (order of polynomial)",
       y = "AIC and BIC criterion") +
  theme(legend.position = "bottom")
fig_AIC_BIC
```


1800
1750
1700
1650
AIC and BIC criterion
1
2
3
4
5
6
7
8
9
10
Model complexity (order of polynomial)
Information criterion
AIC
BIC

Problem 2

In Problem 2, we consider a large population-based case–control study on Oral cancer conducted in the US (Day et al., 1993), from which the data related to the African American population (194 cases, *ccstatus* = 1, and 203 controls, *ccstatus* = 0) have been selected. The aim of the study was to evaluate the risk of Oral cancer based on the variables *drinks* (number of 1oz ethanol-equivalent drinks consumed per week), *sex*, *age* and *cigs* (number of cigarettes smoked per day). We begin by downloading and inspecting the data:

```r
oral_ca <- read.csv("oral_ca/oral_ca.csv")
str(oral_ca)
```

```
## 'data.frame':    397 obs. of  7 variables:
##  $ drinks  : num  11.1 0 48 13 76 ...
##  $ ccstatus: int  1 1 1 1 1 1 1 1 1 1 ...
##  $ cigs    : int  20 6 20 10 40 40 40 20 30 20 ...
##  $ age     : int  52 54 47 39 47 47 37 61 59 36 ...
##  $ sex     : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ M_drinks: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ M_cigs  : int  0 0 0 0 0 0 0 0 0 0 ...
```

```r
summary(oral_ca)
```

```
##      drinks          ccstatus           cigs            age
##  Min.   :  0.00   Min.   :0.0000   Min.   : 0.00   Min.   :21
##  1st Qu.:  1.50   1st Qu.:0.0000   1st Qu.: 3.00   1st Qu.:48
##  Median : 15.75   Median :0.0000   Median :20.00   Median :56
##  Mean   : 31.40   Mean   :0.4887   Mean   :16.36   Mean   :56
##  3rd Qu.: 48.00   3rd Qu.:1.0000   3rd Qu.:20.00   3rd Qu.:65
##  Max.   :140.00   Max.   :1.0000   Max.   :60.00   Max.   :80
##       sex            M_drinks            M_cigs
##  Min.   :0.0000   Min.   :0.000000   Min.   :0.00000
##  1st Qu.:0.0000   1st Qu.:0.000000   1st Qu.:0.00000
##  Median :0.0000   Median :0.000000   Median :0.00000
##  Mean   :0.2771   Mean   :0.005038   Mean   :0.01511
##  3rd Qu.:1.0000   3rd Qu.:0.000000   3rd Qu.:0.00000
##  Max.   :1.0000   Max.   :1.000000   Max.   :1.00000
```

a) Frequencies and probabilities

We begin by looking at the correlation between *ccstatus* and *cigs* as a dichotomized variable. The table below shows the observed frequencies of cases and controls for observations in the non-smokers and smokers groups. We observe that the number of smokers in the dataset is about three times larger than the number of non-smokers, and also that the majority of smokers are in the case group, while the majority of non-smokers are in the control group.

```
oral_ca <- oral_ca %>%
  mutate(cigs.d = ifelse(cigs > 0, "Smokers", "Non-smokers"),
         ccstatus.d = ifelse(ccstatus == 1, "Case", "Control"))

table(oral_ca$cigs.d, oral_ca$ccstatus.d)
```

```
##
##              Case Control
##   Non-smokers   22      69
##   Smokers      172     134
```

We can also compute the estimated probabilities, with their standard errors, of being a case for each of the groups ("Smokers" and "Non-smokers"). The results are shown below. We also include the probability for being a case for the observations in the sample as a whole as this value will be used in pt. (b). The estimated probability of a smoker being a case is approximately 0.562, with a standard error of 0.028. For a non-smoker the corresponding numbers are 0.242 and 0.045. (For the whole sample the estimated probability is 0.489 with a standard error of 0.025.)

```
probs <- oral_ca %>%
  summarise(pi.smoke.hat =
              sum(ccstatus[cigs.d == "Smokers"]) / sum(cigs.d == "Smokers"),
            pi.nonsmoke.hat =
              sum(ccstatus[cigs.d == "Non-smokers"]) / sum(cigs.d == "Non-smokers"),
            pi.common.hat =
              sum(ccstatus) / n(),
            se.pi.smoke.hat =
              sqrt(pi.smoke.hat * (1 - pi.smoke.hat) / sum(cigs.d == "Smokers")),
            se.pi.nonsmoke.hat =
              sqrt(pi.nonsmoke.hat * (1 - pi.nonsmoke.hat) / sum(cigs.d == "Non-smokers")),
            se.pi.common.hat =
              sqrt(pi.common.hat * (1 - pi.common.hat) / n()))
```

```
t(probs)
```

```
##                          [,1]
## pi.smoke.hat       0.56209150
## pi.nonsmoke.hat    0.24175824
## pi.common.hat      0.48866499
## se.pi.smoke.hat    0.02836185
## se.pi.nonsmoke.hat 0.04488216
## se.pi.common.hat   0.02508783
```

b) Test for equal probabilities

Using the estimated probabilities and their standard errors from pt. (a), we compute the likelihood ratio test statistics, w. This test statistics is 2(log(L1)–log(L0)) where L0 is the maximized log-likelihood function under the null hypothesis that the probabilities are equal (i.e. does not depend upon being a smoker or not) and L1 the maximized log-likelihood function without that constraint (Azzalini & Scarpa, 2012). We also compute the p-value of w based on the chi-squared distribution with 1 df. As the p-value is close to zero, we reject the null hypothesis of equal probability.

```
res <- oral_ca %>%
  summarise(llik_pi.smoke.hat_pi.nonsmoke.hat =
              sum(dbinom(ccstatus[cigs.d == "Smokers"], 1, probs$pi.smoke.hat, log = TRUE)) +
              sum(dbinom(ccstatus[cigs.d == "Non-smokers"], 1, probs$pi.nonsmoke.hat, log = TRU
            llik_pi.common.hat = sum(dbinom(ccstatus, 1, probs$pi.common.hat, log = TRUE)),
            w = 2 * (llik_pi.smoke.hat_pi.nonsmoke.hat - llik_pi.common.hat),
            p.value = 1 - pchisq(w, df = 1))
t(res)
```

```
##                                                  [,1]
## llik_pi.smoke.hat_pi.nonsmoke.hat -260.06939029403202
## llik_pi.common.hat                -275.07740682904608
## w                                   30.01603307002813
## p.value                              0.00000004284888
```

c) Linear logistic model with cigarettes (cigs) as dichotomised variable

To fit a linear logistics model, we use the *glm* function, and the results are shown below. As the coefficient of cigs is positive and significant at less than 0.1 percent significance level, being a smoker seems to increase the risk of oral cancer. Specifically, the increase in log-odds from smoking is approximately 1.39. This corresponds to an increase in odds by the exponential of the log-odds, which is approximately 4.028. We can also see this by the calculation below. Furthermore, the intercept coefficient is the log-odds of cancer for the reference group of not being a smoker.

```
mod.c <- glm(ccstatus ~ cigs.d,
             family = "binomial",
             data = oral_ca)
summary(mod.c)
```

```
##
## Call:
## glm(formula = ccstatus ~ cigs.d, family = "binomial", data = oral_ca)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.285  -1.285  -0.744   1.073   1.685
##
## Coefficients:
##                Estimate Std. Error z value    Pr(>|z|)
## (Intercept)     -1.1431     0.2448  -4.669 0.000003033 ***
## cigs.dSmokers    1.3927     0.2706   5.147 0.000000265 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 550.15  on 396  degrees of freedom
## Residual deviance: 520.14  on 395  degrees of freedom
## AIC: 524.14
##
## Number of Fisher Scoring iterations: 4
```

```r
beta <- mod.c$coefficients

pi_values <- data.frame(pi_smoke = exp(beta[1] + beta[2]) / (1 + exp(beta[1] + beta[2])),
                        pi_nonsmoke = exp(beta[1]) / (1 + exp(beta[1])))

log_odds <- pi_values %>%
  mutate(odds_smoke = pi_smoke / (1 - pi_smoke),
         odds_nonsmoke = pi_nonsmoke / (1 - pi_nonsmoke),
         log_odds_smoke = log(odds_smoke),
         log_odds_nonsmoke = log(odds_nonsmoke),
         diff_log_odds = log_odds_smoke - log_odds_nonsmoke,
         diff_odds = exp(diff_log_odds))
t(log_odds)
```

```
##                         [,1]
## pi_smoke           0.5620915
## pi_nonsmoke        0.2417582
## odds_smoke         1.2835821
## odds_nonsmoke      0.3188406
## log_odds_smoke     0.2496547
## log_odds_nonsmoke -1.1430641
## diff_log_odds      1.3927187
## diff_odds          4.0257802
```

d) Linear logistic model with cigarettes (cigs) as a continuous variable

Instead of using a dichotomized variable for *cigs*, we now use the continuous variable in the model. The results are shown below. The coefficient of *cigs* of approximately 0.054 is now the expected change (increase) in log-odds from one additional cigarette smoked per day. Although they are both related to the odds for non-smokers, the coefficient of the intercept changes with respect to pt. (c) because *cigs* is now a continuous variable.

```r
mod.d <- glm(ccstatus ~ cigs,
             family = "binomial",
             data = oral_ca)
summary(mod.d)
```

```
##
## Call:
## glm(formula = ccstatus ~ cigs, family = "binomial", data = oral_ca)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1923  -1.0237  -0.8228   1.1088   1.5796
##
## Coefficients:
##              Estimate Std. Error z value       Pr(>|z|)
## (Intercept) -0.909057   0.171766  -5.292 0.000000120714 ***
## cigs         0.053624   0.008614   6.225 0.000000000481 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 550.15  on 396  degrees of freedom
## Residual deviance: 504.39  on 395  degrees of freedom
## AIC: 508.39
##
## Number of Fisher Scoring iterations: 4
```

e) Linear logistic model including all the explanatory variables

We now include the other three variables (*drinks*, *age* and *sex*) in the model as well, and the results are shown below. Of the three variables, *drinks* and *sex* are significant at 5 percent significance level or less, and both have a positive correlation with the risk of oral cancer. The coefficient of the variable *age* is also positive, but it is not significantly different from zero. Furthermore, we observe that the increase in the log-odds for one more cigarette smoked per day now is 0.035, which is a lower value than in pt. (d) where it was 0.054. The reason for this is that in pt. (d), the model suffered from omitted variable bias. That is, one or more relevant variables correlating both with *ccstatus* and with *cigs* were left out of the model, resulting in the coefficient of *cigs* being estimated to be too high. Finally, we can notice that AIC is 453.8.

```
mod.e <- glm(ccstatus ~ cigs + drinks + age + sex,
             family = "binomial",
             data = oral_ca)
summary(mod.e)
```

```
##
## Call:
## glm(formula = ccstatus ~ cigs + drinks + age + sex, family = "binomial",
##     data = oral_ca)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7185  -0.8589  -0.5832   0.9644   1.9776
##
## Coefficients:
##              Estimate Std. Error z value       Pr(>|z|)
## (Intercept) -1.966071   0.620756  -3.167        0.00154 **
## cigs         0.035480   0.009571   3.707        0.00021 ***
## drinks       0.029623   0.004643   6.380 0.000000000177 ***
## age          0.006529   0.009960   0.656        0.51213
## sex          0.594499   0.272752   2.180        0.02928 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 550.15  on 396  degrees of freedom
## Residual deviance: 443.84  on 392  degrees of freedom
## AIC: 453.84
##
## Number of Fisher Scoring iterations: 5
```

f) Exclude age from the model

The coefficient of age has a positive value, but it is not statistically significant at 5 percent level, thus we do not reject the null hypothesis that the coefficient is zero. Hence, it does not seem to be the case that older people risk more, controlling for the other variables (*cigs*, *drinks* and *sex*). We now fit a new model excluding age, and the results are shown below. Specifically, the coefficients of *cigs*, *drinks* and *sex* are more or less the same as in pt. (e), and AIC is slightly improved to 452.3. As the model without *age* is simpler than the model including *age*, and also improves AIC, I would prefer this simpler model.

```r
mod.f <- glm(ccstatus ~ cigs + drinks + sex,
             family = "binomial",
             data = oral_ca)
summary(mod.f)
```

```
## 
## Call:
## glm(formula = ccstatus ~ cigs + drinks + sex, family = "binomial", 
##     data = oral_ca)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -2.6943  -0.8596  -0.6084   0.9607   1.8857  
## 
## Coefficients:
##              Estimate Std. Error z value        Pr(>|z|)    
## (Intercept) -1.592919   0.238787  -6.671 0.0000000000254 ***
## cigs         0.035536   0.009565   3.715        0.000203 ***
## drinks       0.029498   0.004638   6.360 0.0000000002012 ***
## sex          0.582183   0.271756   2.142        0.032169 *  
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 550.15  on 396  degrees of freedom
## Residual deviance: 444.27  on 393  degrees of freedom
## AIC: 452.27
## 
## Number of Fisher Scoring iterations: 5
```

g) Quadratic term for drinks

We still exclude *age* and instead include a polynomial of degree 2 to model the effect of *drinks*, where the results are shown below. The coefficients of all variables are statistically significant at a 5 percent significance level, and AIC is further improved to 447.6, compared to 452.3. Hence, including a polynomial of degree 2 appears to improve the model.

```
mod.g <- glm(ccstatus ~ cigs + drinks + I(drinks^2) + sex,
             family = "binomial",
             data = oral_ca)
summary(mod.g)
```

```
##
## Call:
## glm(formula = ccstatus ~ cigs + drinks + I(drinks^2) + sex, family = "binomial",
##     data = oral_ca)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2192  -0.8379  -0.5405   0.8756   1.9980
##
## Coefficients:
##                Estimate  Std. Error z value         Pr(>|z|)
## (Intercept) -1.84989123  0.26670134  -6.936 0.00000000000403 ***
## cigs         0.03301049  0.00964170   3.424         0.000618 ***
## drinks       0.05361658  0.01051587   5.099 0.00000034210964 ***
## I(drinks^2) -0.00022953  0.00008419  -2.726         0.006405 **
## sex          0.72564421  0.28540905   2.542         0.011007 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 550.15  on 396  degrees of freedom
## Residual deviance: 437.62  on 392  degrees of freedom
## AIC: 447.62
##
## Number of Fisher Scoring iterations: 4
```

h) Quadratic term for cigarettes (cigs)

If we instead include a quadratic term for *cigs*, we get the results shown below. It appears that including a quadratic term for *cigs* does not improve the model as the coefficient of *cigs*^2 is not significant and AIC is increased to 453.6.

```
mod.h <- glm(ccstatus ~ cigs + I(cigs^2) + drinks + sex,
             family = "binomial",
             data = oral_ca)
summary(mod.h)
```

```
##
## Call:
## glm(formula = ccstatus ~ cigs + I(cigs^2) + drinks + sex, family = "binomial",
##     data = oral_ca)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6870  -0.8764  -0.5808   0.9695   1.9303
##
## Coefficients:
##               Estimate Std. Error z value       Pr(>|z|)
## (Intercept) -1.6943120  0.2697349  -6.281 0.000000000336 ***
## cigs         0.0541177  0.0238775   2.266         0.0234 *
## I(cigs^2)   -0.0004485  0.0005225  -0.858         0.3907
## drinks       0.0289791  0.0046347   6.253 0.000000000404 ***
## sex          0.6019008  0.2742844   2.194         0.0282 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 550.15  on 396  degrees of freedom
## Residual deviance: 443.55  on 392  degrees of freedom
## AIC: 453.55
##
## Number of Fisher Scoring iterations: 5
```

i) Cubic terms

Finally, we can build models including a cubic term for *drinks* and *cigs*, respectively, to see if that improves the fit. However, as the results below show, neither models appear to be an improvement compared to the simpler ones, as the coefficients of the second and third order polynomials are not statistically significant in any of the models. Also, AIC is worse in both models compared to the AIC of the model in pt. (g). Hence, the best model seems to be the one with quadratic effect for drinks.

Cubic term for drinks

```r
mod.i1 <- glm(ccstatus ~ cigs + drinks + I(drinks^2) + I(drinks^3) + sex,
              family = "binomial",
              data = oral_ca)
summary(mod.i1)
```

```
##
## Call:
## glm(formula = ccstatus ~ cigs + drinks + I(drinks^2) + I(drinks^3) +
##     sex, family = "binomial", data = oral_ca)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1993  -0.8382  -0.5367   0.8703   2.0045
##
## Coefficients:
##                  Estimate   Std. Error z value        Pr(>|z|)
## (Intercept) -1.865004213  0.287615254  -6.484 0.0000000000891 ***
## cigs         0.032940949  0.009651358   3.413        0.000642 ***
## drinks       0.056358094  0.021973596   2.565        0.010323 *
## I(drinks^2) -0.000296010  0.000474763  -0.623        0.532963
## I(drinks^3)  0.000000354  0.000002488   0.142        0.886856
## sex          0.732025366  0.289282623   2.530        0.011390 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 550.15  on 396  degrees of freedom
## Residual deviance: 437.60  on 391  degrees of freedom
## AIC: 449.6
##
## Number of Fisher Scoring iterations: 4
```

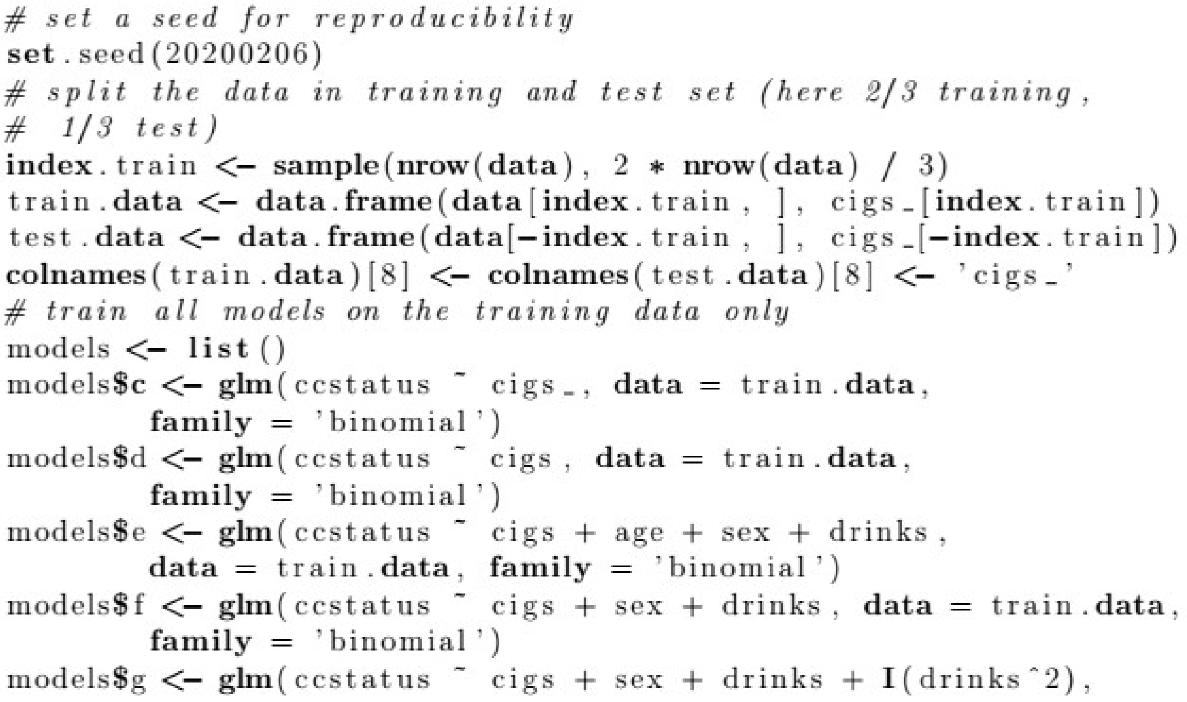
Cubic term for cigs

```
mod.i2 <- glm(ccstatus ~ cigs + I(cigs^2) + I(cigs^3) + drinks + sex,
              family = "binomial",
              data = oral_ca)
summary(mod.i2)
```

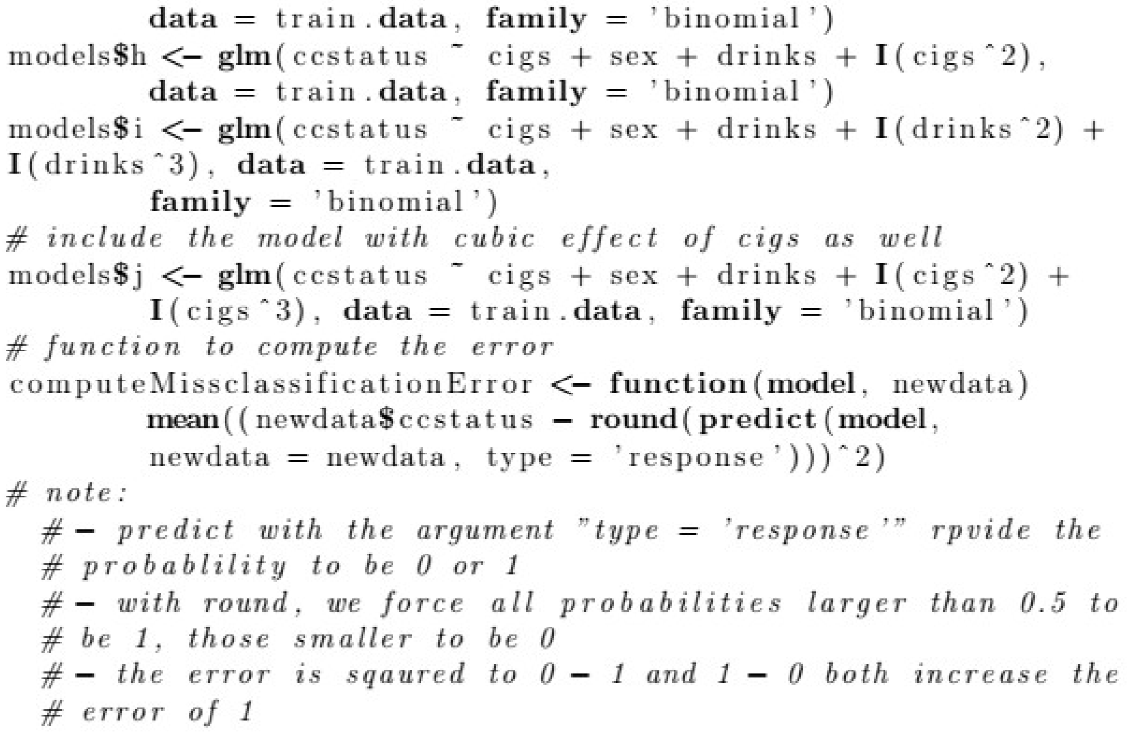
```
##
## Call:
## glm(formula = ccstatus ~ cigs + I(cigs^2) + I(cigs^3) + drinks +
##     sex, family = "binomial", data = oral_ca)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.6567  -0.8871  -0.5658   0.9740   1.9551
##
## Coefficients:
##                 Estimate  Std. Error z value       Pr(>|z|)
## (Intercept) -1.75117731  0.28631554  -6.116 0.000000000958 ***
## cigs         0.07878751  0.04578855   1.721         0.0853 .
## I(cigs^2)   -0.00190087  0.00236132  -0.805         0.4208
## I(cigs^3)    0.00002011  0.00003215   0.626         0.5316
## drinks       0.02896095  0.00464273   6.238 0.000000000443 ***
## sex          0.61435091  0.27523429   2.232         0.0256 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 550.15  on 396  degrees of freedom
```

k) Separate training and test set

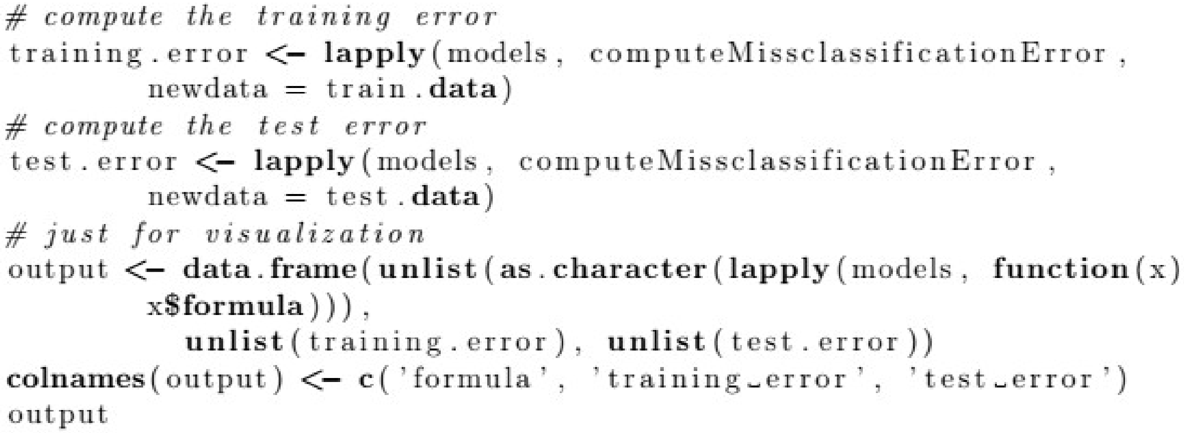
Finally, we examine which model is the best for prediction by splitting the data into a training set and a test set, where we use 2/3 of the data for training and 1/3 for testing. To ensure that results are replicable, we use set.seed. However, it should be noticed that the computed errors of the training and test data for the different models will depend on the split into training and test. If the data were split using a different set.seed value, the results may change.



```
# set a seed for reproducibility
set.seed(20200206)
# split the data in training and test set (here 2/3 training,
# 1/3 test)
index.train <- sample(nrow(data), 2 * nrow(data) / 3)
train.data <- data.frame(data[index.train, ], cigs_[index.train])
test.data <- data.frame(data[-index.train, ], cigs_[-index.train])
colnames(train.data)[8] <- colnames(test.data)[8] <- 'cigs_'
# train all models on the training data only
models <- list()
models$c <- glm(ccstatus ~ cigs_, data = train.data,
        family = 'binomial')
models$d <- glm(ccstatus ~ cigs, data = train.data,
        family = 'binomial')
models$e <- glm(ccstatus ~ cigs + age + sex + drinks,
        data = train.data, family = 'binomial')
models$f <- glm(ccstatus ~ cigs + sex + drinks, data = train.data,
        family = 'binomial')
models$g <- glm(ccstatus ~ cigs + sex + drinks + I(drinks^2),
```



```
        data = train.data, family = 'binomial')
models$h <- glm(ccstatus ~ cigs + sex + drinks + I(cigs^2),
        data = train.data, family = 'binomial')
models$i <- glm(ccstatus ~ cigs + sex + drinks + I(drinks^2) +
I(drinks^3), data = train.data,
        family = 'binomial')
# include the model with cubic effect of cigs as well
models$j <- glm(ccstatus ~ cigs + sex + drinks + I(cigs^2) +
        I(cigs^3), data = train.data, family = 'binomial')
# function to compute the error
computeMissclassificationError <- function(model, newdata)
        mean((newdata$ccstatus - round(predict(model,
        newdata = newdata, type = 'response')))^2)
# note:
  #- predict with the argument "type = 'response'" rpvide the
  # probablility to be 0 or 1
  #- with round, we force all probabilities larger than 0.5 to
  # be 1, those smaller to be 0
  #- the error is sqaured to 0 - 1 and 1 - 0 both increase the
  # error of 1
```



```
# compute the training error
training.error <- lapply(models, computeMissclassificationError,
        newdata = train.data)
# compute the test error
test.error <- lapply(models, computeMissclassificationError,
        newdata = test.data)
# just for visualization
output <- data.frame(unlist(as.character(lapply(models, function(x)
        x$formula))),
          unlist(training.error), unlist(test.error))
colnames(output) <- c('formula', 'training_error', 'test_error')
output
```

Output

```
                                                       formula training error test error
c                                              ccstatus ~ cigs_     0.4090909
0.3609023
d                                              ccstatus ~ cigs     0.3446970  0.3308271
e                             ccstatus ~ cigs + age + sex + drinks     0.2424242  0.2781955
f                                   ccstatus ~ cigs + sex + drinks     0.2651515  0.2781955
g                      ccstatus ~ cigs + sex + drinks + I(drinks^2)     0.2537879  0.2932331
h                        ccstatus ~ cigs + sex + drinks + I(cigs^2)     0.2613636  0.2631579
I        ccstatus ~ cigs + sex + drinks + I(drinks^2) + I(drinks^3)     0.2575758  0.2781955
j            ccstatus ~ cigs + sex + drinks + I(cigs^2) + I(cigs^3)     0.2613636  0.2932331
```

The test error is the lowest for the model from task h), so I choose this model as the best.