Mini-Project 2

STAT-340 Applied Regression Methods

Yerim Oh

Load Packages

The following R code loads packages needed in this assignment.

Problem 1: Validation of model assumptions

"Overall mortality indicators (e.g., life expectancy at birth or survival to a given age) are important indicators of health status in a country. Because data on the incidence and prevalence of diseases are frequently unavailable, mortality rates are often used to identify vulnerable populations. And they are among the indicators most frequently used to compare socioeconomic development across countries.

The data in this problem is from World Health Organization (WHO) and the United Nations Population Division and contain the following variables for 170 countries:

- `LifeExpectancy2015`: the life expectancy at birth in the year 2015 (number of years a newborn infant would live if prevailing patterns of mortality at the time of its birth were to stay the same throughout its life).
- `IncomeGroup`: the country's income group (low, lower-middle, upper-middle, and high)
- `Polio`: Polio immunization coverage among 1-year-olds (%)
- `IncomeCompositionResources`: Human Development Index in terms of income composition of resources (index ranging from 0 to 1)
- `Schooling`: Number of years of Schooling (years)

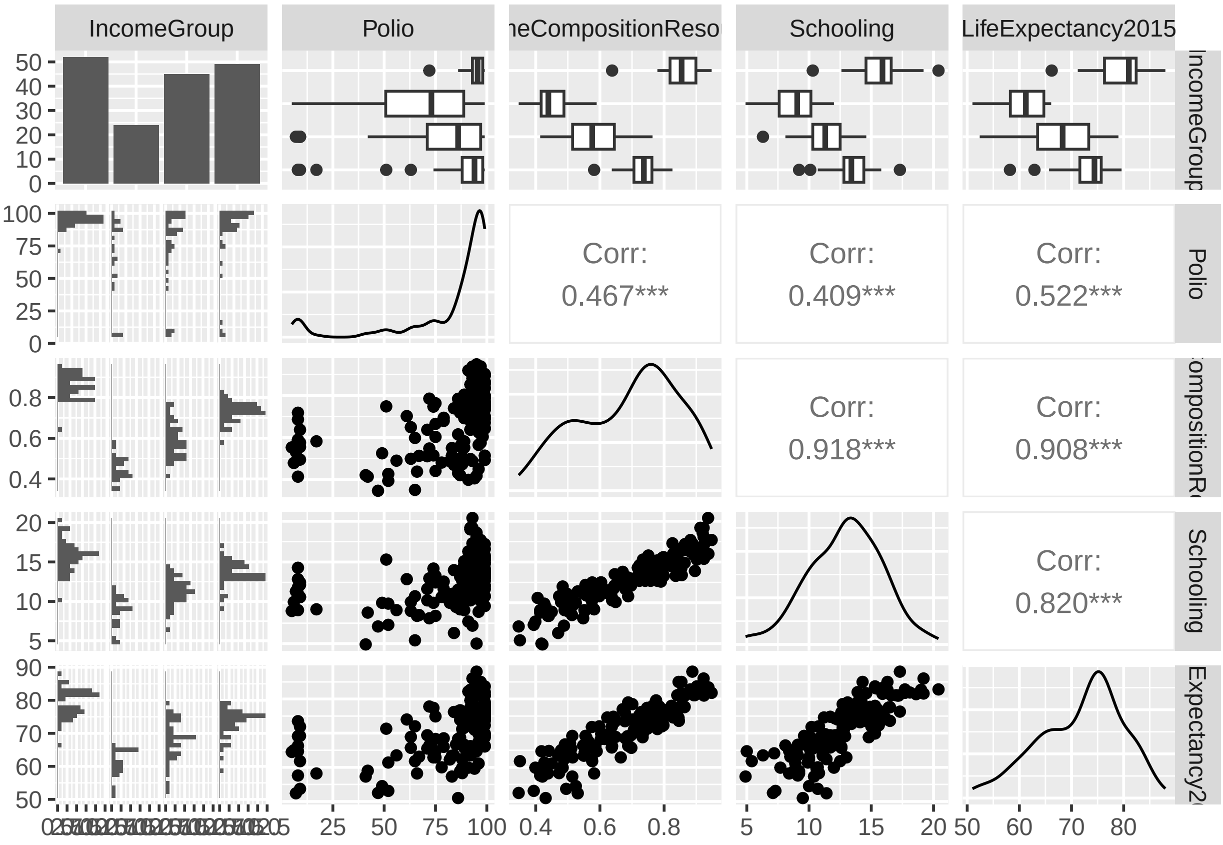
We treat `LifeExpectancy2015` as the response variable and the reminder variables as the explanatory variables.

Description: https://data.worldbank.org/indicator/SP.DYN.LE00.IN

Data sources: https://www.kaggle.com/datasets/kumarajarshi/life-expectancy-who/data and https://data.worldbank.org/indicator/SP.DYN.LE00.IN

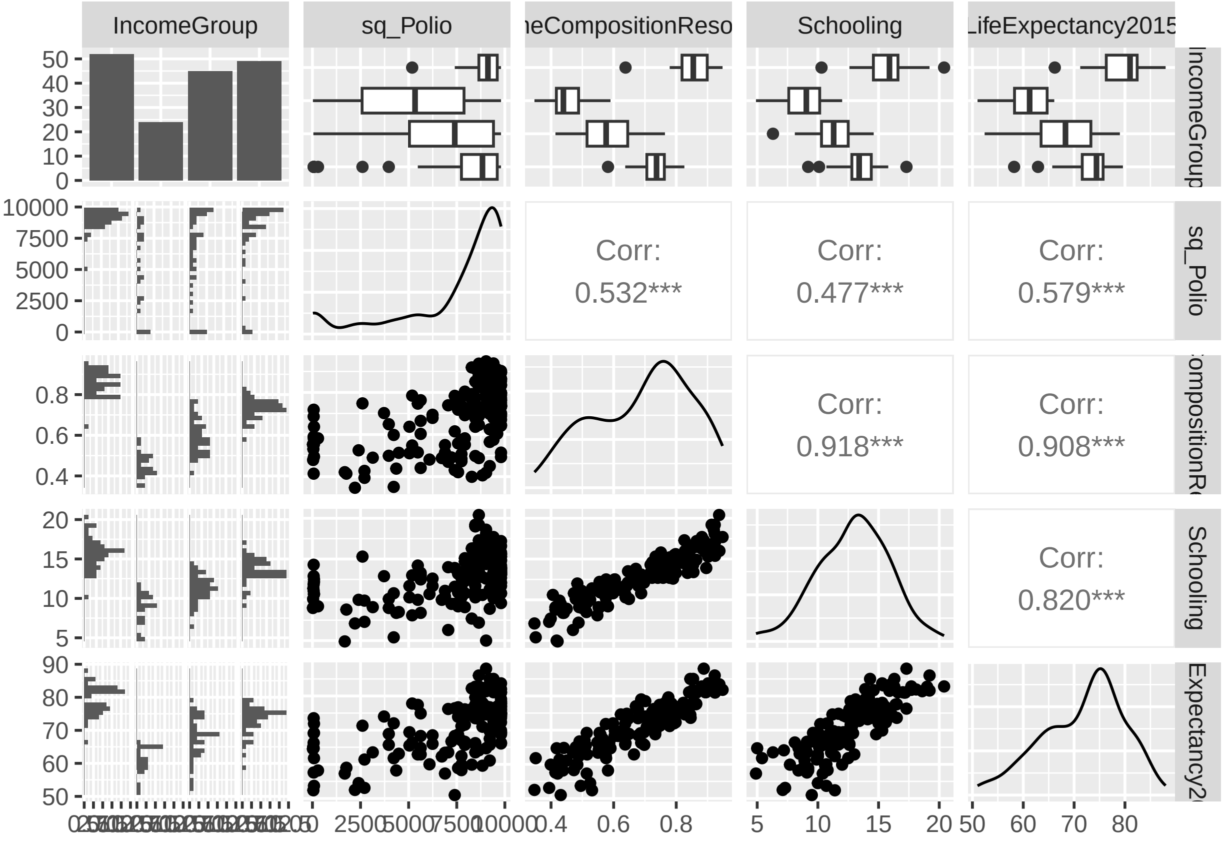
a) Make a pair plot all the variables.

```
ggpairs(LifeExp)
```



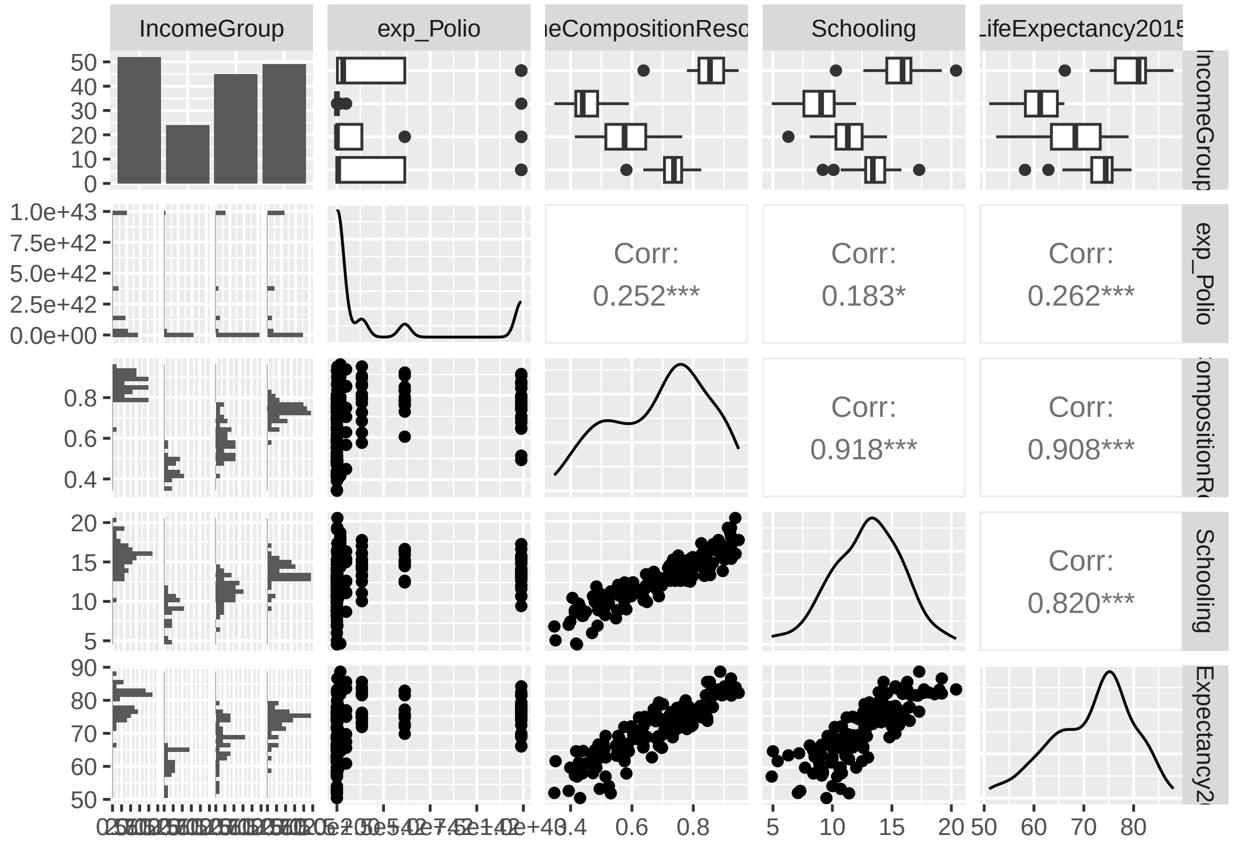
b) Develop a predictive model for the response variable `LifeExpectancy2015` by transforming the response and/or quantitative explanatory variables so that the associations between the transformed variables are approximately linear and the variance is approximately constant.

```r
LifeExp_transformed <- LifeExp %>%
  transmute(IncomeGroup = IncomeGroup,
            sq_Polio = Polio^2,
            IncomeCompositionResources = IncomeCompositionResources,
            Schooling = Schooling,
            LifeExpectancy2015 = LifeExpectancy2015)

ggpairs(LifeExp_transformed)
```



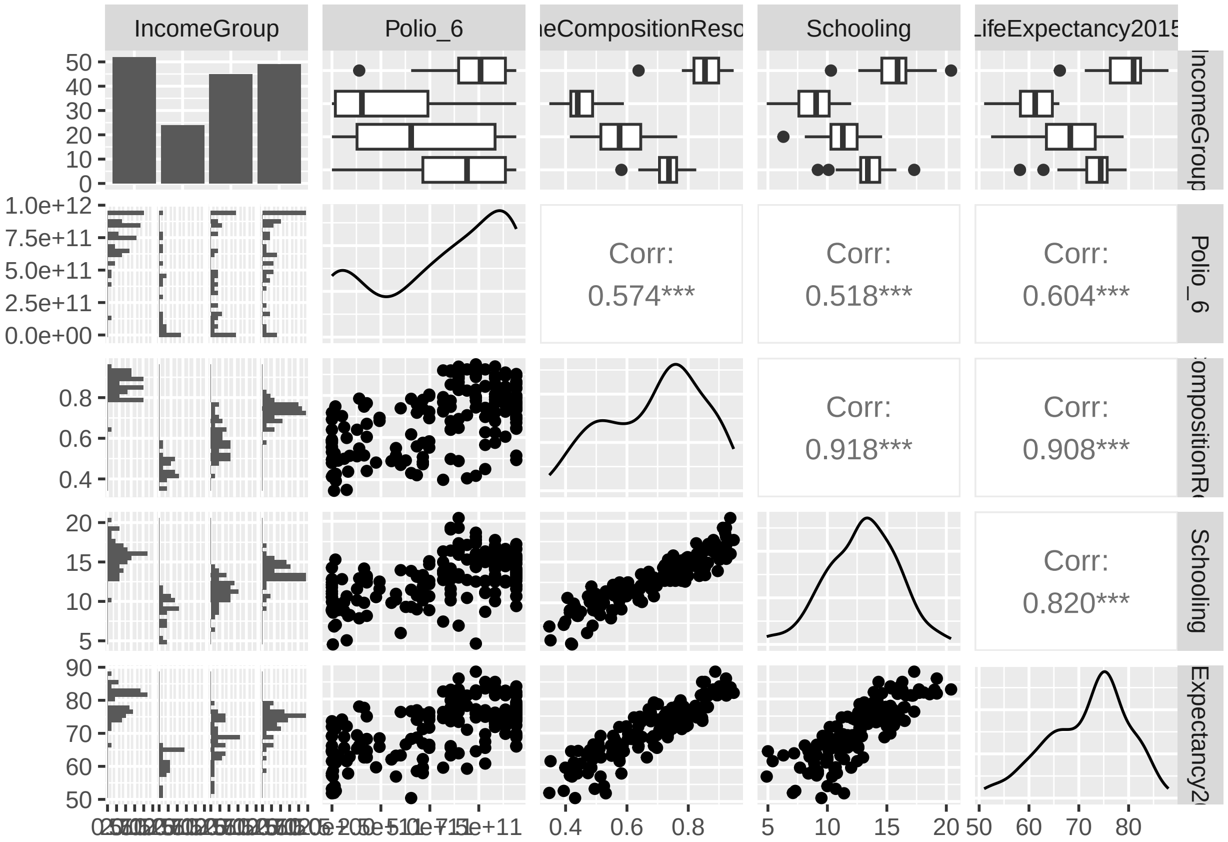
Squared `Polio` is still left skewed and has low correlation and inconsistant variance with the response variable.

```r
LifeExp_transformed <- LifeExp %>%
  transmute(IncomeGroup = IncomeGroup,
            exp_Polio = exp(Polio),
            IncomeCompositionResources = IncomeCompositionResources,
            Schooling = Schooling,
            LifeExpectancy2015 = LifeExpectancy2015)

ggpairs(LifeExp_transformed)
```



Now, exp_Polio is right skewed and still not consistent.

```
LifeExp_transformed <- LifeExp %>%
  transmute(IncomeGroup = IncomeGroup,
            Polio_6 = Polio^6,
            IncomeCompositionResources = IncomeCompositionResources,
            Schooling = Schooling,
            LifeExpectancy2015 = LifeExpectancy2015)

ggpairs(LifeExp_transformed)
```



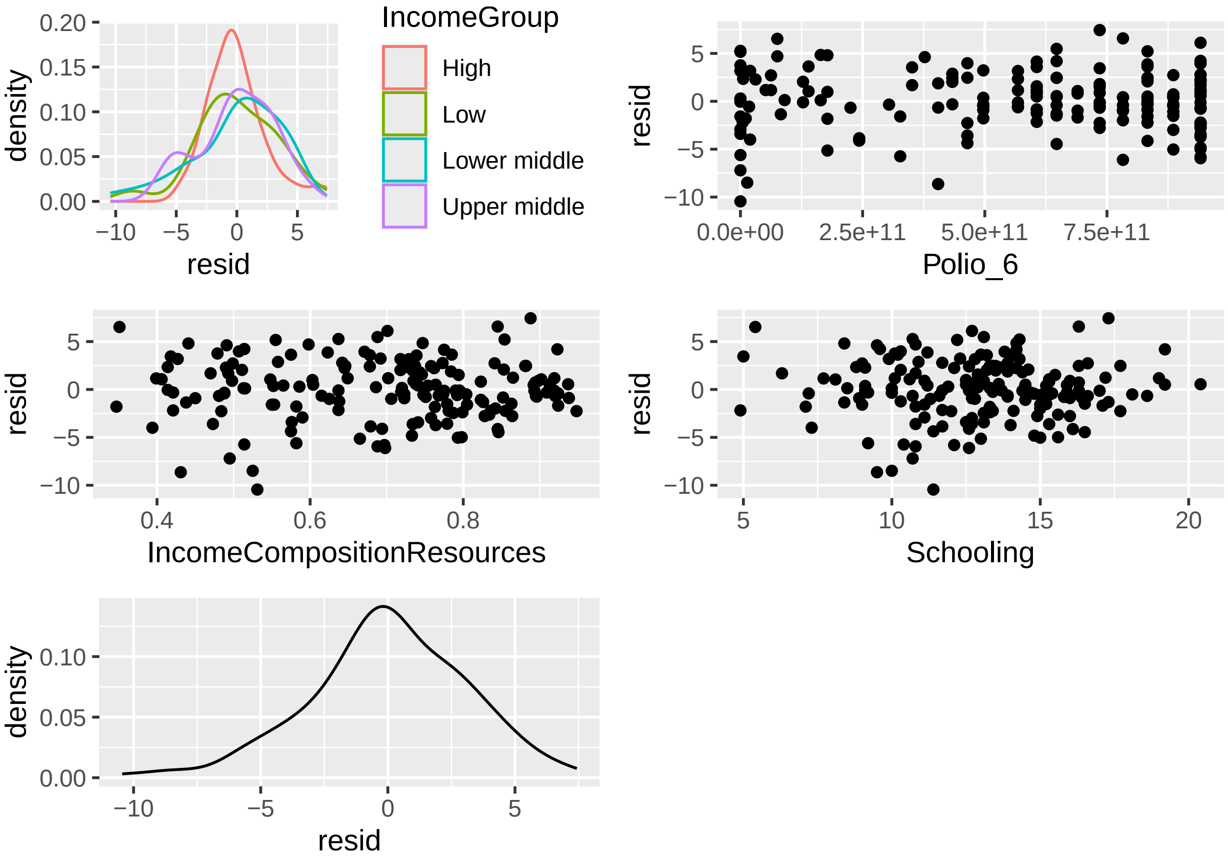
- I do not think additional transformations of `Polio` can simultaneously improve the relationships with `LifeExpectancy2015`. Although it is still left skewed, this transformation of the variable is more correlated than other transformations before and the variance is fairly constant across the range of values for the transformed `LifeExpextancy2015`.
- There seems to have a correlation between `IncomeCompositionResources` and `Schooling` which is really high. This could separate out the individual effects of collinear variables on the response.

c) Fit a model to your transformed data, and create a set of diagnostic plots, which should include plots (i) to (iv).

```r
# fit a model to the transformed data
lm_fit <- lm(LifeExpectancy2015 ~
               IncomeGroup + Polio_6 + IncomeCompositionResources + Schooling,
             data = LifeExp_transformed)

LifeExp_transformed <- LifeExp_transformed %>%
  mutate(resid = residuals(lm_fit))
```

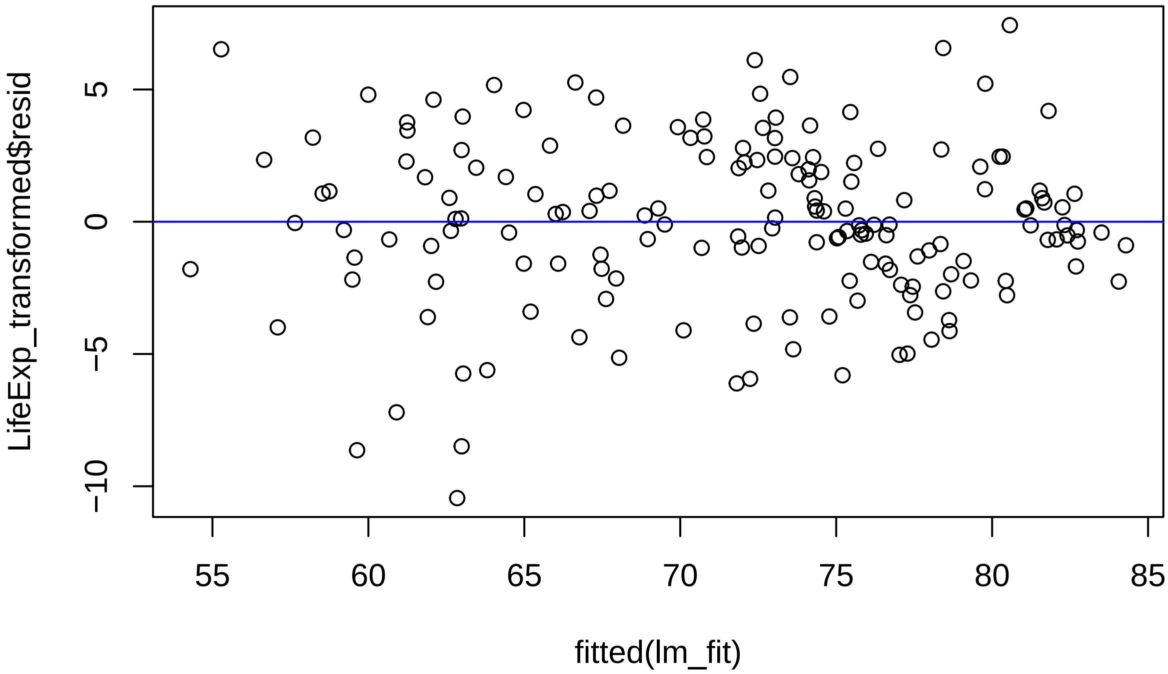
```r
# scatter plots of the residuals vs. each quantitative explanatory variable
sp1 <- ggplot(data = LifeExp_transformed,
              mapping = aes(x = resid, color = IncomeGroup))+
  geom_density()

sp2 <- ggplot(data = LifeExp_transformed,
              mapping = aes(x = Polio_6, y = resid))+
  geom_point()

sp3 <- ggplot(data = LifeExp_transformed,
              mapping = aes(x = IncomeCompositionResources, y = resid))+
  geom_point()

sp4 <- ggplot(data = LifeExp_transformed,
              mapping = aes(x = Schooling, y = resid))+
  geom_point()

sp5 <- ggplot(data = LifeExp_transformed, mapping = aes(x = resid))+
  geom_density()

grid.arrange(sp1, sp2, sp3, sp4, sp5, ncol=2)
```

(i) scatter plots of the residuals vs. each quantitative explanatory variable in the data set



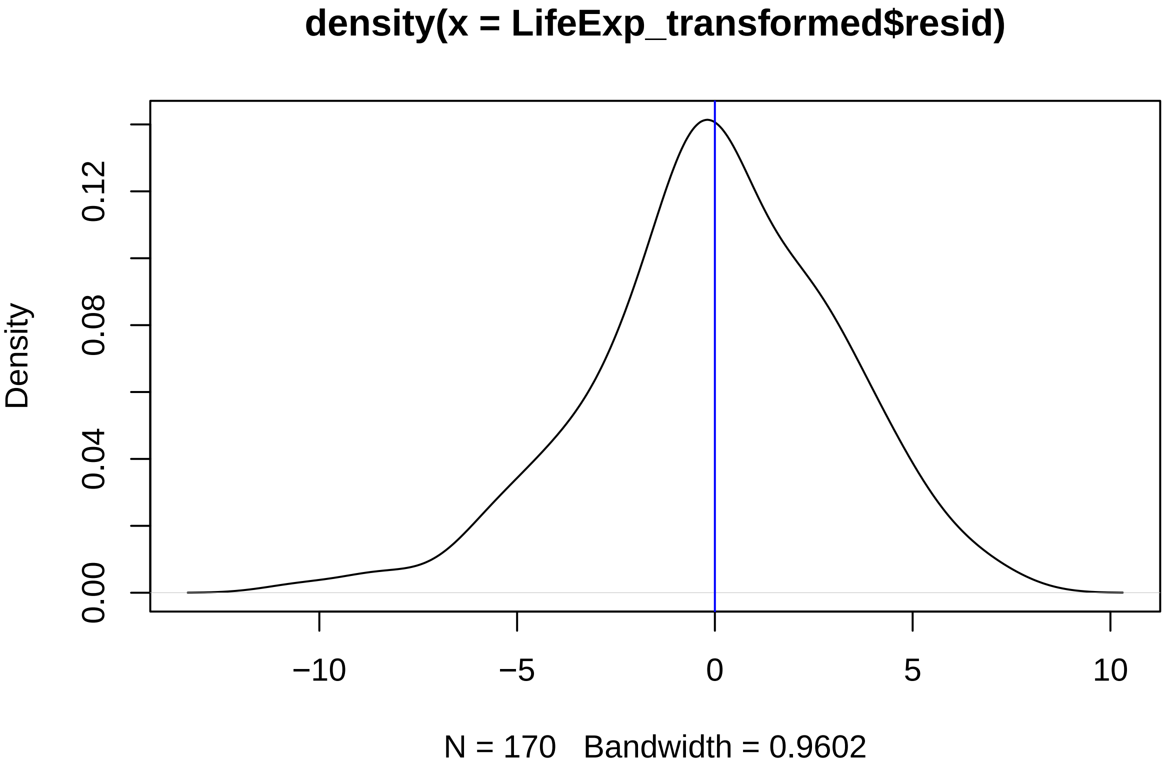
```
plot(fitted(lm_fit), LifeExp_transformed$resid)
abline(0,0, col="blue")
```

(ii) a scatter plot of the residuals against the fitted values



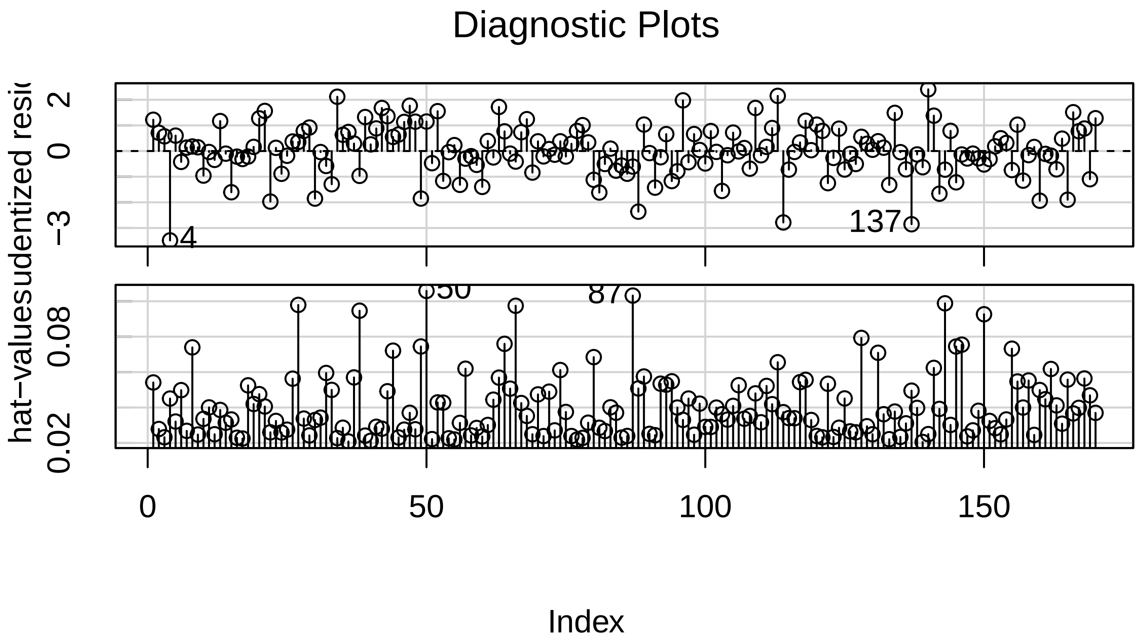
```r
plot(density(LifeExp_transformed$resid))
abline(v=0, col="blue")
```

(iii) a density plot or histogram of the residuals; and



```r
car::influenceIndexPlot(lm_fit,
                        vars = c("Studentized", "hat"))
```

(iv) diagnostic plots of studentized residuals, and leverage.

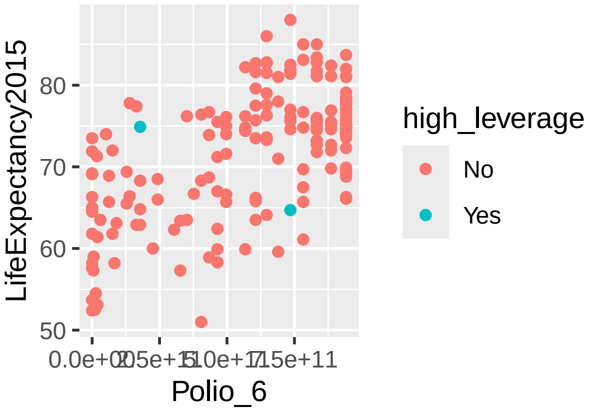


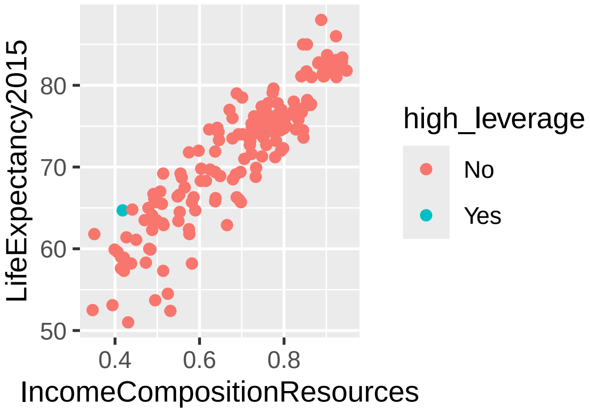
```r
# leverage
2 * length(coef(lm_fit)) / nrow(LifeExp_transformed)
```

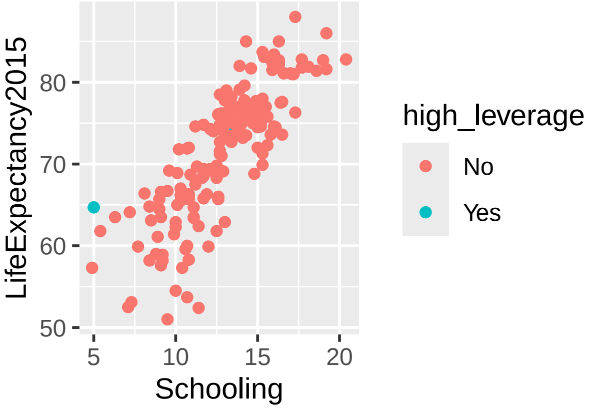
```
## [1] 0.08235294
```

Observation 50 and 87 show high leverage rates.

```r
LifeExp_transformed$high_leverage <- "No"
LifeExp_transformed$high_leverage[c(50, 87)] <- "Yes"

p1 <- ggplot(data = LifeExp_transformed,
             mapping = aes(x = Polio_6, y = LifeExpectancy2015,
                           color = high_leverage)) +
  geom_point()

p2 <- ggplot(data = LifeExp_transformed,
             mapping = aes(x = IncomeCompositionResources, y = LifeExpectancy2015,
                           color = high_leverage)) +
  geom_point()

p3 <- ggplot(data = LifeExp_transformed,
             mapping = aes(x = Schooling, y = LifeExpectancy2015,
                           color = high_leverage)) +
  geom_point()

grid.arrange(p1, p2, p3, ncol=2)
```







Not particularly worried.

```
lm_fit_no_high_leverage <- lm(LifeExpectancy2015 ~
                                IncomeGroup + Polio_6 +
                                IncomeCompositionResources + Schooling,
                              data = LifeExp_transformed %>%
                                filter(high_leverage == "No"))
summary(lm_fit)
```

```
##
## Call:
## lm(formula = LifeExpectancy2015 ~ IncomeGroup + Polio_6 + IncomeCompositionResources +
##     Schooling, data = LifeExp_transformed)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.4478  -1.7571  -0.1062   2.2361   7.4317
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                3.067e+01  3.417e+00   8.974 6.59e-16 ***
## IncomeGroupLow             4.990e+00  1.790e+00   2.787  0.00595 **
## IncomeGroupLower middle    3.886e+00  1.242e+00   3.128  0.00209 **
## IncomeGroupUpper middle    1.030e+00  8.023e-01   1.284  0.20092
## Polio_6                    2.712e-12  9.152e-13   2.963  0.00350 **
## IncomeCompositionResources 6.046e+01  6.060e+00   9.977  < 2e-16 ***
## Schooling                 -3.343e-01  2.132e-01  -1.568  0.11877
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.171 on 163 degrees of freedom
## Multiple R-squared:  0.8462, Adjusted R-squared:  0.8406
## F-statistic: 149.5 on 6 and 163 DF,  p-value: < 2.2e-16
```

```
summary(lm_fit_no_high_leverage)
```

```
##
## Call:
## lm(formula = LifeExpectancy2015 ~ IncomeGroup + Polio_6 + IncomeCompositionResources +
##     Schooling, data = LifeExp_transformed %>% filter(high_leverage ==
##     "No"))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.5898  -1.6749   0.0297   2.2510   7.3562
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 2.997e+01  3.535e+00   8.476 1.41e-14 ***
## IncomeGroupLow              5.114e+00  1.843e+00   2.775  0.00617 **
## IncomeGroupLower middle     4.116e+00  1.292e+00   3.187  0.00173 **
## IncomeGroupUpper middle     1.124e+00  8.134e-01   1.381  0.16906
## Polio_6                     2.467e-12  9.384e-13   2.629  0.00940 **
## IncomeCompositionResources  6.079e+01  6.301e+00   9.647  < 2e-16 ***
## Schooling                  -2.957e-01  2.196e-01  -1.346  0.18008
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.173 on 161 degrees of freedom
## Multiple R-squared:  0.847,  Adjusted R-squared:  0.8413
## F-statistic: 148.5 on 6 and 161 DF,  p-value: < 2.2e-16
```

There are no high differences between the model fits with and without the high leverage observations. Nothing to worry about.

If you see any serious issues, go back to step b) and try additional transformations.

d) Take a look at the summary output for your chosen model. Which variables would hypothesis tests suggest have a strong relationship with `LifeExpectancy2015`, the life expectancy at birth in 2015? Is this aligned with your interpretation of the pair plot of the (possibly transformed) variables? If not, what do you think might be going on?

```
summary(lm_fit)
```

```
##
## Call:
## lm(formula = LifeExpectancy2015 ~ IncomeGroup + Polio_6 + IncomeCompositionResources +
##     Schooling, data = LifeExp_transformed)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.4478  -1.7571  -0.1062   2.2361   7.4317
##
```

```
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                  3.067e+01  3.417e+00   8.974 6.59e-16 ***
## IncomeGroupLow               4.990e+00  1.790e+00   2.787  0.00595 ** 
## IncomeGroupLower middle      3.886e+00  1.242e+00   3.128  0.00209 ** 
## IncomeGroupUpper middle      1.030e+00  8.023e-01   1.284  0.20092    
## Polio_6                      2.712e-12  9.152e-13   2.963  0.00350 ** 
## IncomeCompositionResources   6.046e+01  6.060e+00   9.977  < 2e-16 ***
## Schooling                   -3.343e-01  2.132e-01  -1.568  0.11877    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 3.171 on 163 degrees of freedom
## Multiple R-squared:  0.8462, Adjusted R-squared:  0.8406 
## F-statistic: 149.5 on 6 and 163 DF,  p-value: < 2.2e-16
```

There is fairly strong evidence of an association between `Polio_6`, `IncomeCompositionResources` and `sq_LifeExpectancy2015`. We have to conduct an F test to investigate `IncomeGroup`:

```
reduced_fit <- lm(LifeExpectancy2015 ~ Polio_6 +
                    IncomeCompositionResources +
                    Schooling,
                  data = LifeExp_transformed)

anova(reduced_fit, lm_fit)
```

```
## Analysis of Variance Table
## 
## Model 1: LifeExpectancy2015 ~ Polio_6 + IncomeCompositionResources + Schooling
## Model 2: LifeExpectancy2015 ~ IncomeGroup + Polio_6 + IncomeCompositionResources + 
##     Schooling
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)   
## 1    166 1759.2                                
## 2    163 1638.6  3    120.65 4.0007 0.008821 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

A hypothesis test says `IncomeGroup` is important too. Therefore, the only variable a hypothesis test says we could drop is `Schooling`.

e) Check for multicollinearity among the explanatory variables. If you detect multicollinearity, please discuss how you would address it.

There was a fairly strong evidence of multicollinearity between `IncomeCompositionResources` and `Schooling` when we looked at the pair plots.

```
vif(lm_fit)
```

```
##                                 GVIF Df GVIF^(1/(2*Df))
## IncomeGroup                 6.273654  3        1.358064
## Polio_6                     1.508758  1        1.228315
## IncomeCompositionResources 14.679236  1        3.831349
## Schooling                   6.575440  1        2.564262
```

When the vif is higher than 5, there is severe correlation between a given predictor variable and other predictor variables in the model. This means that all of the explanatory variables except `Polio_6` is correlated with other predictor variables in the model.

Since the high multicollinearity would make the result we got unreliable, we should drop the variable with the highest VIF, which is `IncomeCompositionResources` although it has the highest correlation with the explanatory variable.

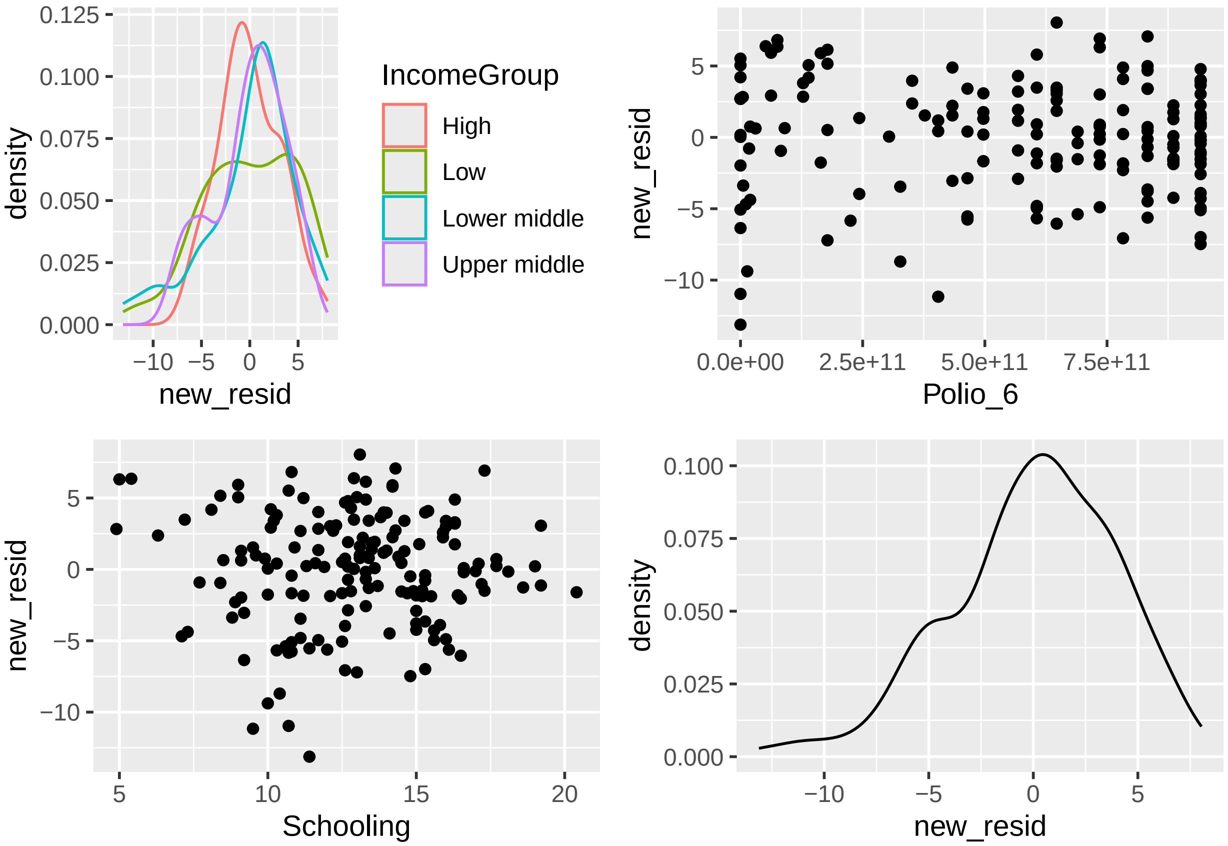
New Model

new model without `IncomeCompositionResources`

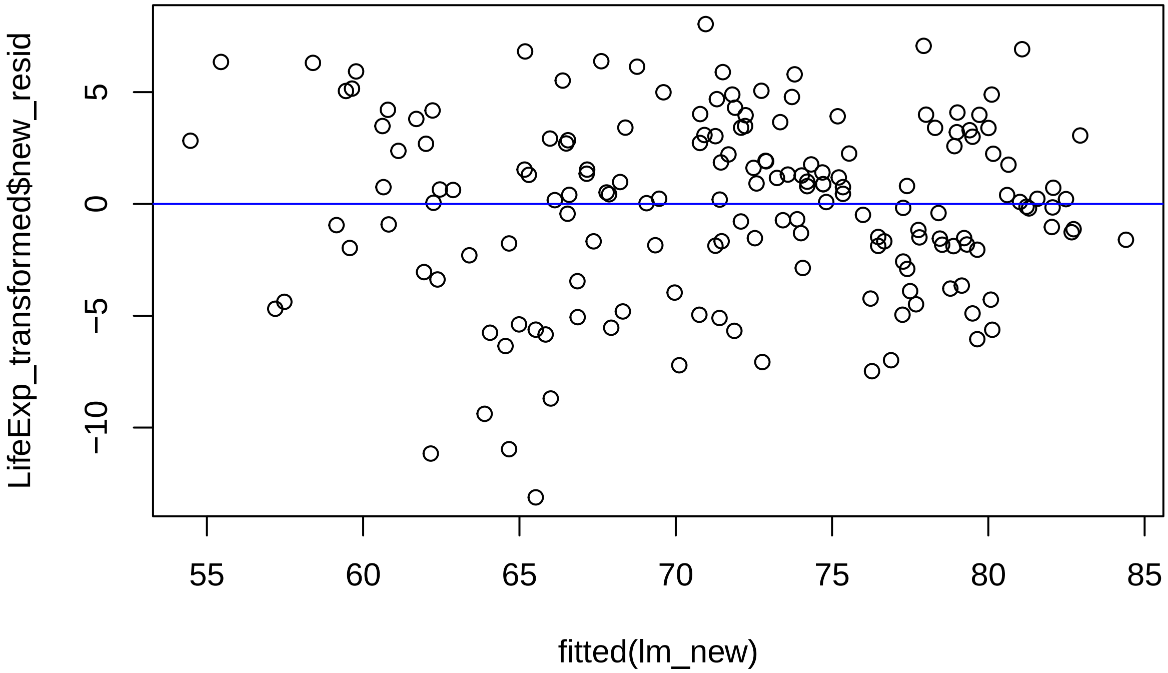
```
lm_new <- lm(LifeExpectancy2015 ~
               IncomeGroup + Polio_6 + Schooling,
             data = LifeExp_transformed)
LifeExp_transformed <- LifeExp_transformed %>%
  mutate(new_resid = residuals(lm_new))
```

Plots

```
# scatter plot residuals vs. each quantitative explanatory variable
sp1 <- ggplot(data = LifeExp_transformed,
              mapping = aes(x = new_resid, color = IncomeGroup))+
  geom_density()

sp2 <- ggplot(data = LifeExp_transformed,
              mapping = aes(x = Polio_6, y = new_resid))+
  geom_point()

sp3 <- ggplot(data = LifeExp_transformed,
              mapping = aes(x = Schooling, y = new_resid))+
  geom_point()

sp4 <- ggplot(data = LifeExp_transformed, mapping = aes(x = new_resid))+
  geom_density()

grid.arrange(sp1, sp2, sp3, sp4, ncol=2)
```

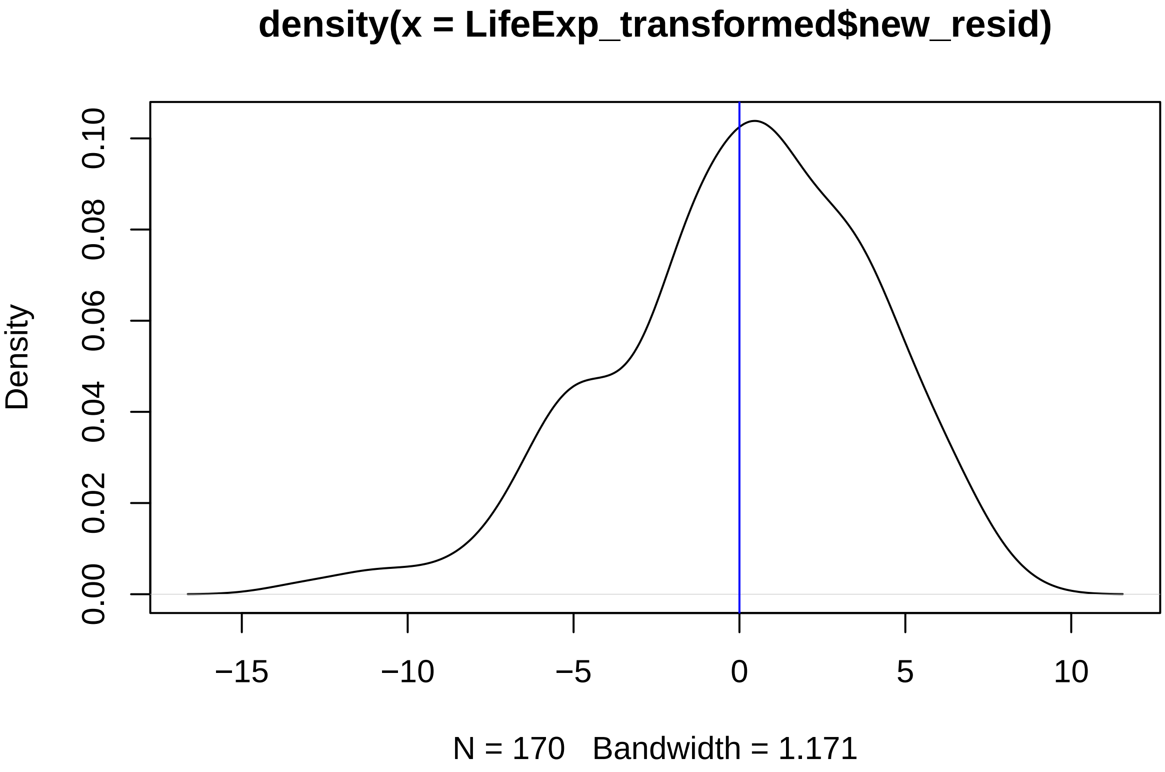


```r
# scatter plot against the fitted value
plot(fitted(lm_new), LifeExp_transformed$new_resid)
abline(0,0, col="blue")
```

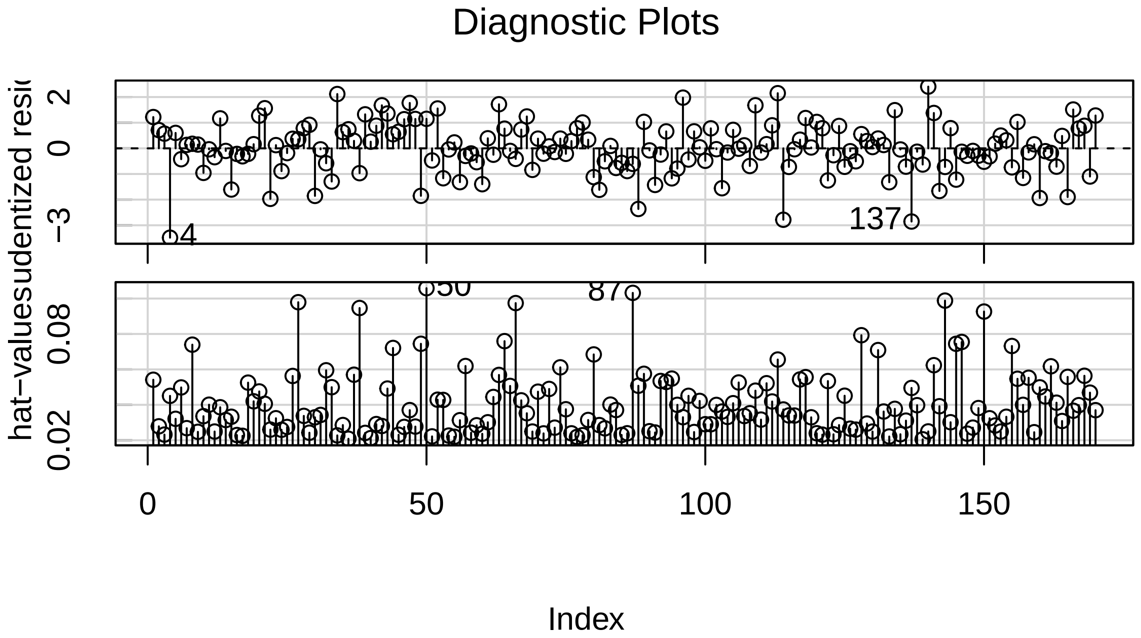


```r
# density plot of the residuals
plot(density(LifeExp_transformed$new_resid))
```

```
abline(v=0, col="blue")
```



```
# diagnostic plots of studentized residuals, and leverage
car::influenceIndexPlot(lm_fit,
                        vars = c("Studentized", "hat"))
```



```
# leverage
2 * length(coef(lm_fit)) / nrow(LifeExp_transformed)
```

```
## [1] 0.08235294
```

Observation 50 and 87 show high leverage rates (these are the observations that is already picked from the previous model)

```r
lm_new_no_high_leverage <- lm(LifeExpectancy2015 ~
                                IncomeGroup + Polio_6 + Schooling,
                              data = LifeExp_transformed %>%
                                filter(high_leverage == "No"))
summary(lm_new)
```

```
##
## Call:
## lm(formula = LifeExpectancy2015 ~ IncomeGroup + Polio_6 + Schooling,
##     data = LifeExp_transformed)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.1201  -1.8831   0.4027   2.9846   8.0428
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              5.616e+01  2.871e+00  19.559  < 2e-16 ***
## IncomeGroupLow          -7.688e+00  1.596e+00  -4.818 3.28e-06 ***
## IncomeGroupLower middle -4.543e+00  1.153e+00  -3.942 0.000119 ***
## IncomeGroupUpper middle -2.827e+00  8.895e-01  -3.178 0.001773 **
## Polio_6                  5.199e-12  1.114e-12   4.666 6.34e-06 ***
## Schooling                1.219e+00  1.842e-01   6.621 4.84e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.012 on 164 degrees of freedom
## Multiple R-squared:  0.7523, Adjusted R-squared:  0.7447
## F-statistic: 99.61 on 5 and 164 DF,  p-value: < 2.2e-16
```

```r
summary(lm_new_no_high_leverage)
```

```
##
## Call:
## lm(formula = LifeExpectancy2015 ~ IncomeGroup + Polio_6 + Schooling,
##     data = LifeExp_transformed %>% filter(high_leverage == "No"))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.0520  -1.8836   0.3808   2.9390   8.1500
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              5.577e+01  2.895e+00  19.261  < 2e-16 ***
## IncomeGroupLow          -7.817e+00  1.584e+00  -4.936 1.96e-06 ***
## IncomeGroupLower middle -4.582e+00  1.158e+00  -3.957 0.000113 ***
## IncomeGroupUpper middle -2.776e+00  8.840e-01  -3.140 0.002007 **
## Polio_6                  5.055e-12  1.126e-12   4.489 1.35e-05 ***
## Schooling                1.251e+00  1.879e-01   6.658 4.09e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 3.974 on 162 degrees of freedom
## Multiple R-squared:  0.7585, Adjusted R-squared:  0.751
## F-statistic: 101.8 on 5 and 162 DF,  p-value: < 2.2e-16
```

There are no high differences between the model fits with and without the high leverage observations. Nothing to worry about.

Summary output for the new model

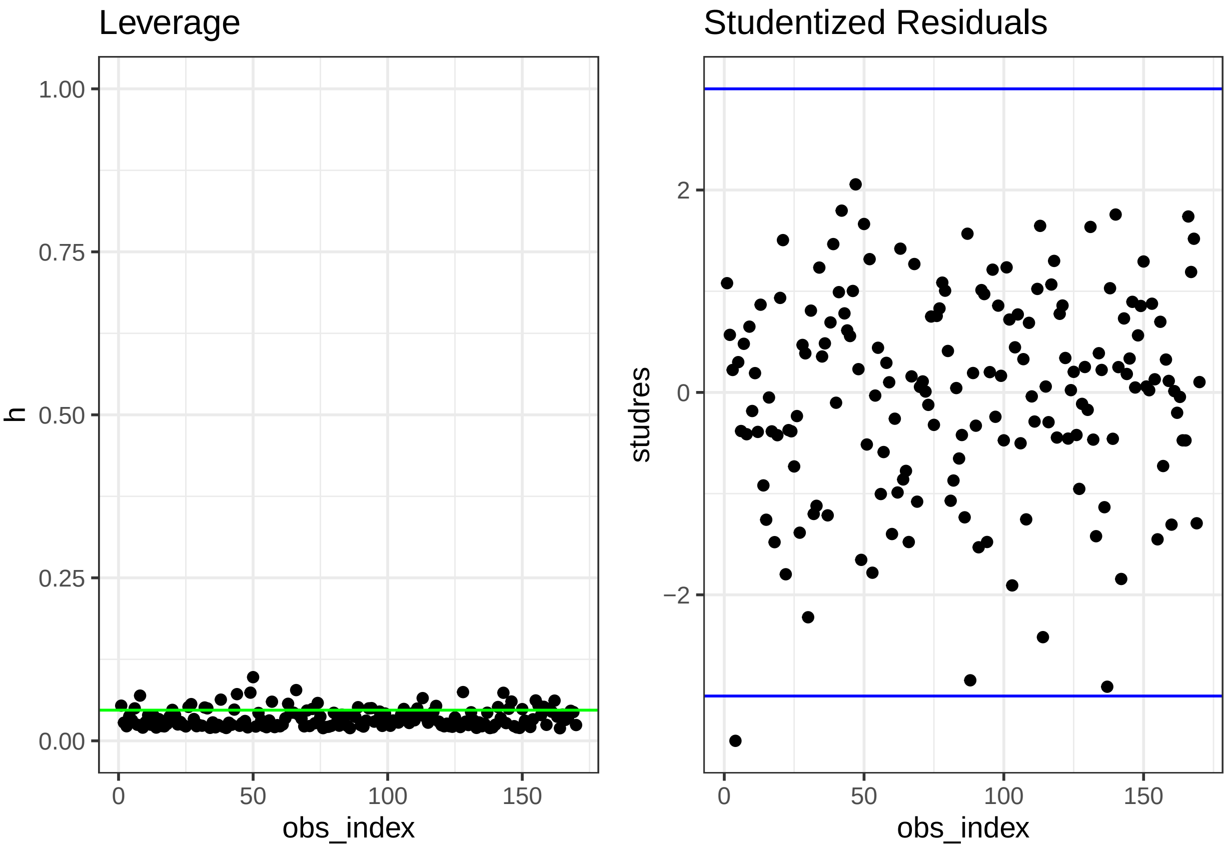
```
summary(lm_new)
```

```
##
## Call:
## lm(formula = LifeExpectancy2015 ~ IncomeGroup + Polio_6 + Schooling,
##     data = LifeExp_transformed)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.1201  -1.8831   0.4027   2.9846   8.0428
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              5.616e+01  2.871e+00  19.559  < 2e-16 ***
## IncomeGroupLow          -7.688e+00  1.596e+00  -4.818 3.28e-06 ***
## IncomeGroupLower middle -4.543e+00  1.153e+00  -3.942 0.000119 ***
## IncomeGroupUpper middle -2.827e+00  8.895e-01  -3.178 0.001773 **
## Polio_6                  5.199e-12  1.114e-12   4.666 6.34e-06 ***
## Schooling                1.219e+00  1.842e-01   6.621 4.84e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.012 on 164 degrees of freedom
## Multiple R-squared:  0.7523, Adjusted R-squared:  0.7447
## F-statistic: 99.61 on 5 and 164 DF,  p-value: < 2.2e-16
```

There is fairly strong evidence of an association between all explanatory variables, `IncomeGroup`, `Polio_6`, and `Schooling`, with the response variable, `LifeExpectancy2015`. So, this will be the final model we are going to observe.

f) Are there any outliers in the data? If so, are they considered influential, i.e. are the results of your statistical analysis significantly affected by the outliers, if any? How did you make this determination?

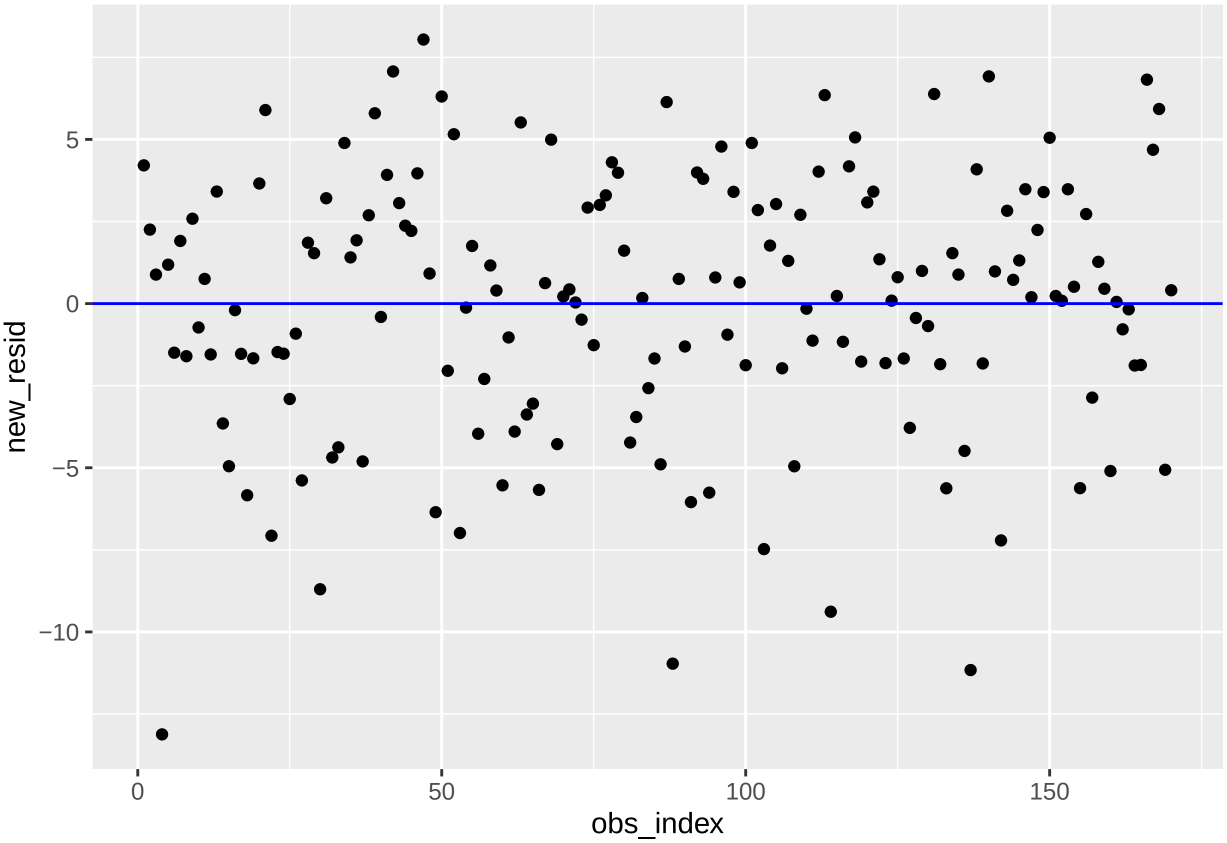
```
# calculate leverage and studentized residuals
LifeExp_transformed <- LifeExp_transformed %>%
  mutate(obs_index = row_number(),
         h = hatvalues(lm_new), # ask if need to exclude schooling and make new model
         studres = rstudent(lm_new))

lev_p <- ggplot(data = LifeExp_transformed,
                mapping = aes(x = obs_index, y = h)) +
  geom_point() +
  geom_hline(yintercept = 2 * 4 / nrow(LifeExp_transformed), col = "green") +
  ylim(0, 1) +
  ggtitle("Leverage") +
  theme_bw()

studres_p <- ggplot(data = LifeExp_transformed,
                    mapping = aes(x = obs_index, y = studres)) +
  geom_point() +
  geom_hline(yintercept = 3, col = "blue") +
  geom_hline(yintercept = -3, col = "blue") +
  ggtitle("Studentized Residuals") +
  theme_bw()

grid.arrange(lev_p, studres_p, ncol = 2)
```



By observing lot of observations above $2(p+1)/n$ line, there are a lot of potential outliers. Since there is one observation below -3 in the studentized residual plot, there is an influential outlier.

g) Plot the ordered residuals. Is there autocorrelation? Summarize and explain your findings.

```
ggplot(data = LifeExp_transformed,
              mapping = aes(x = obs_index, y = new_resid))+
  geom_point() +
  geom_hline(yintercept = 0, col = "blue")
```



Since we cannot observe the errors related to each other or see bunch of errors gathered, we can conclude that there is no autocorrelation.

Problem 2: Polynomial Regression

The following R code loads in a data set with measurements of the tensile strength of paper (`tensile`, in units of pounds per square inch), and the percent of hardwood in the batch of pulp that was used to produce the paper (`hardwood`), for 19 different samples of paper with different percent hardwoods.

References: R package for "Basic Statistics and Data Analysis" by Alan T. Arnholt: https://alanarnholt.github.io/BSDA/

(a) Fit and summarize polynomial regression models of degree 2, and 3.

For each of these candidate models, please produce:

- Output from the `summary` function that you could use to conduct relevant hypothesis tests
- A scatter plot of the data with the estimated curve overlaid on top. The estimated curve can be produced by adding this code to your ggplot:

  ```
  geom_smooth(method = "lm", formula = y ~ poly(x, 2, raw = TRUE))
  ```

where "2" can be replaced by the degree of the polynomial

- A plot of the residuals vs. fitted values
- The residual sum of squares (RSS). (You should also know how to find the R^2 and residual standard error (RSE) in the `summary` output.)

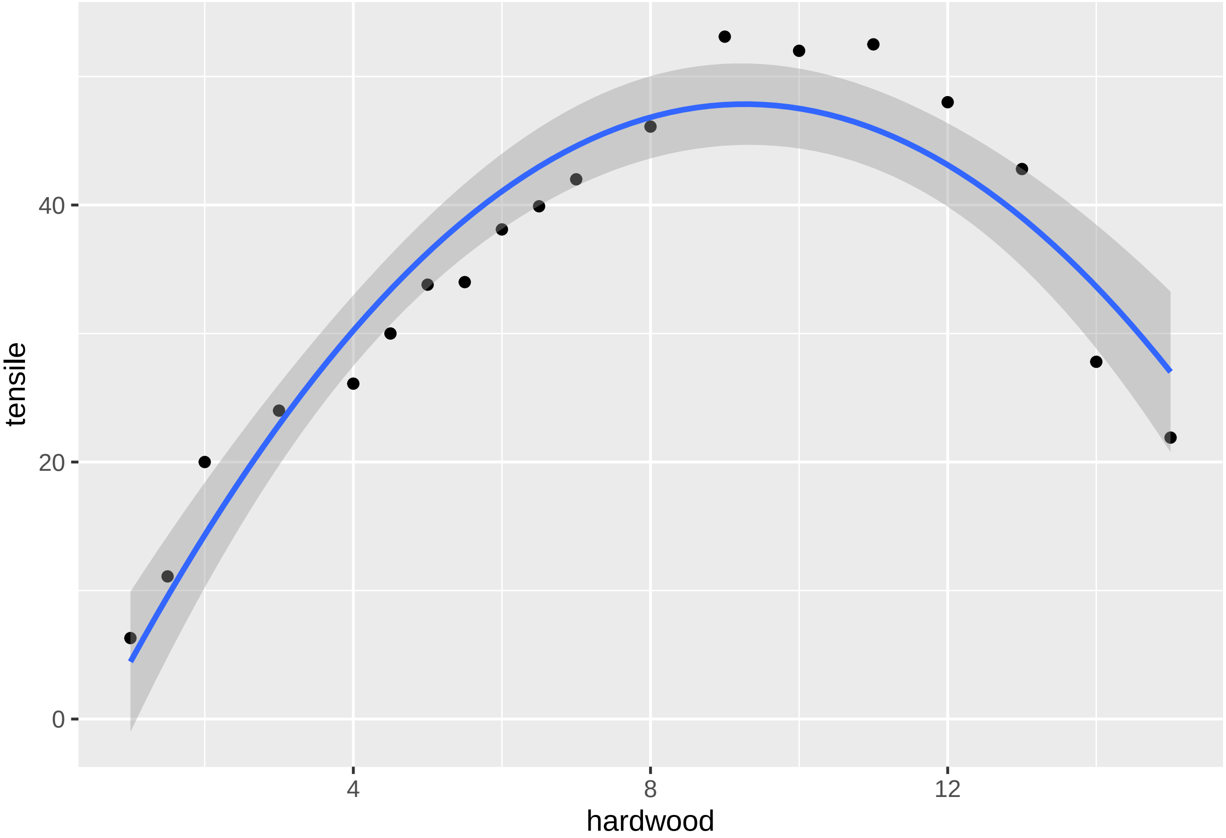
$$RSE = \sqrt{\frac{1}{n-p-1}RSS} = \sqrt{\frac{1}{19-2-1}RSS}RSS = RSE^2 \cdot (n-p-1)$$

```r
model_d2 <- lm(tensile ~ poly(hardwood, 2, raw = TRUE),
               data = paper)

# summary of the model
summary(model_d2)
```

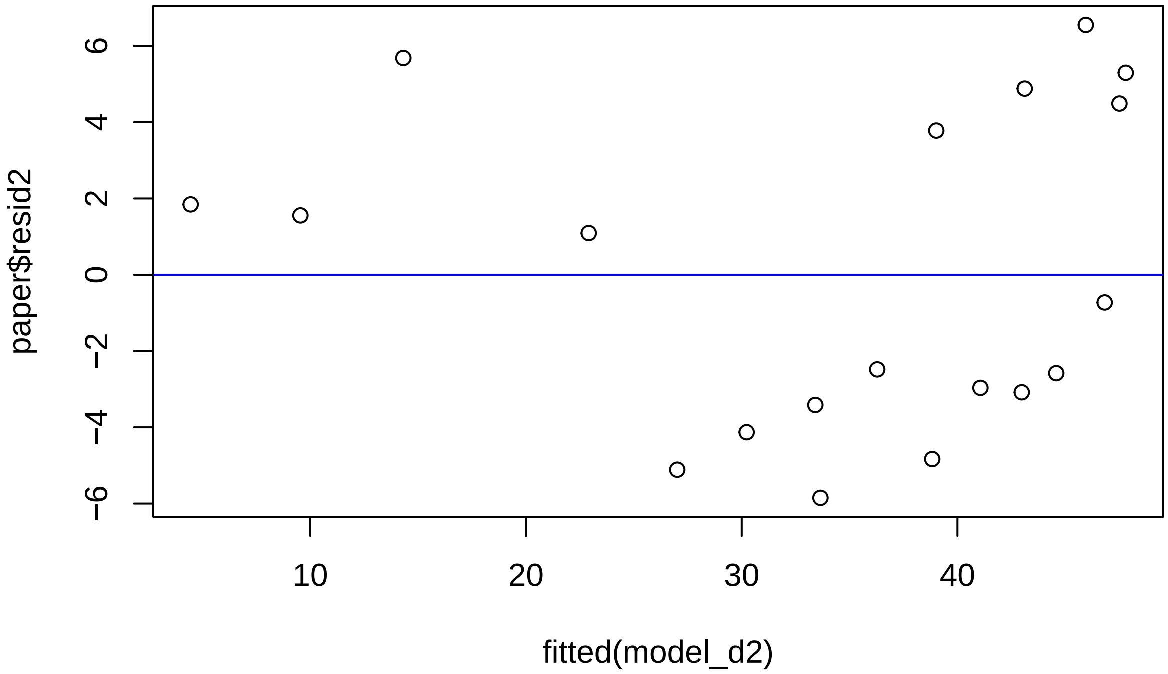
Polynomial regression model of degree 2

```
##
## Call:
## lm(formula = tensile ~ poly(hardwood, 2, raw = TRUE), data = paper)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.8503 -3.2482 -0.7267  4.1350  6.5506
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                    -6.67419    3.39971  -1.963   0.0673 .
## poly(hardwood, 2, raw = TRUE)1 11.76401    1.00278  11.731 2.85e-09 ***
## poly(hardwood, 2, raw = TRUE)2 -0.63455    0.06179 -10.270 1.89e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.42 on 16 degrees of freedom
## Multiple R-squared:  0.9085, Adjusted R-squared:  0.8971
## F-statistic: 79.43 on 2 and 16 DF,  p-value: 4.912e-09
```

```
# scatter plot of the data with the estimated curve overlaid on top
ggplot(data = paper, mapping = aes(x = hardwood, y = tensile))+
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ poly(x, 2, raw = TRUE))
```



```
# plot of the residuals vs. fitted values
paper <- paper %>%
  mutate(resid2 = residuals(model_d2))
plot(fitted(model_d2), paper$resid2)
abline(0,0, col="blue")
```



```
# residual sum of squares (RSS)
(RSS <- 4.42^2 * 16)
```

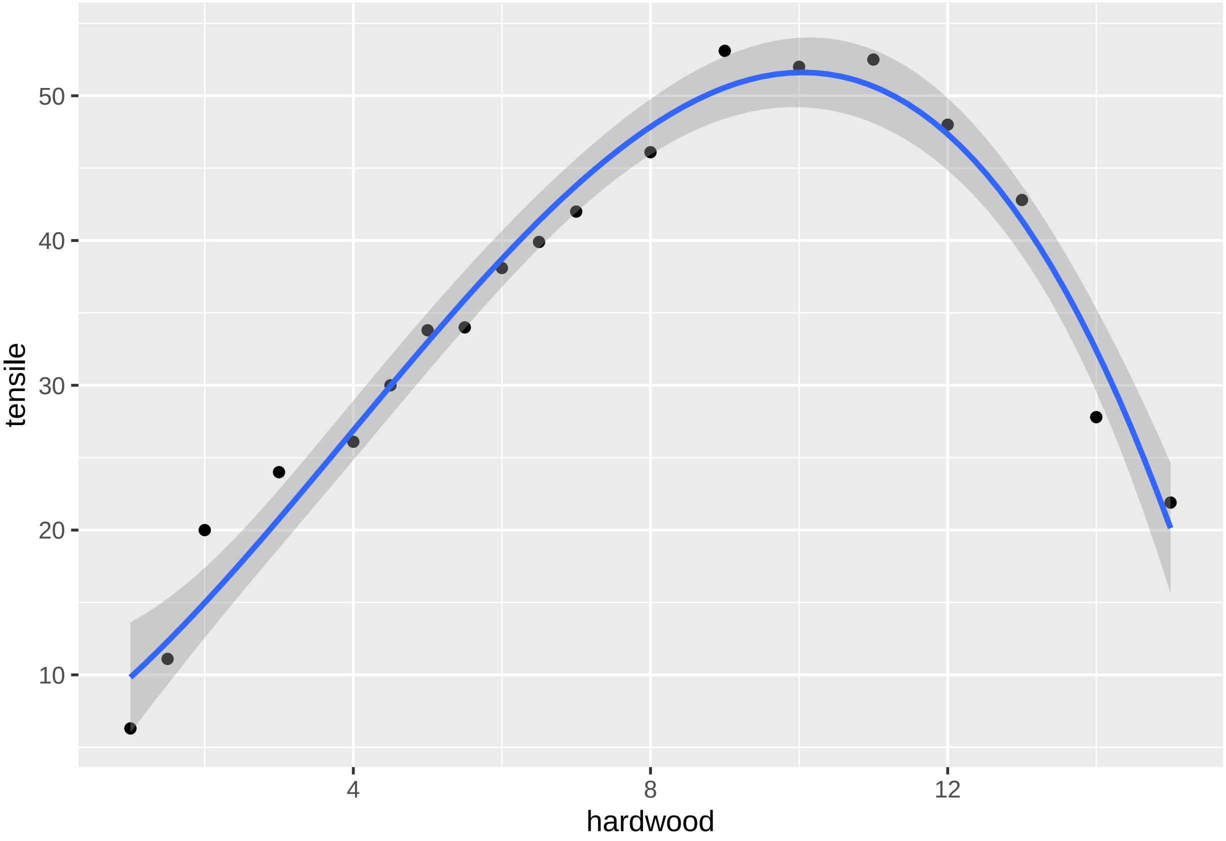
```
## [1] 312.5824
```

```

model_d3 <- lm(tensile ~ poly(hardwood, 3, raw = TRUE),
               data = paper)

# summary of the model
summary(model_d3)
```

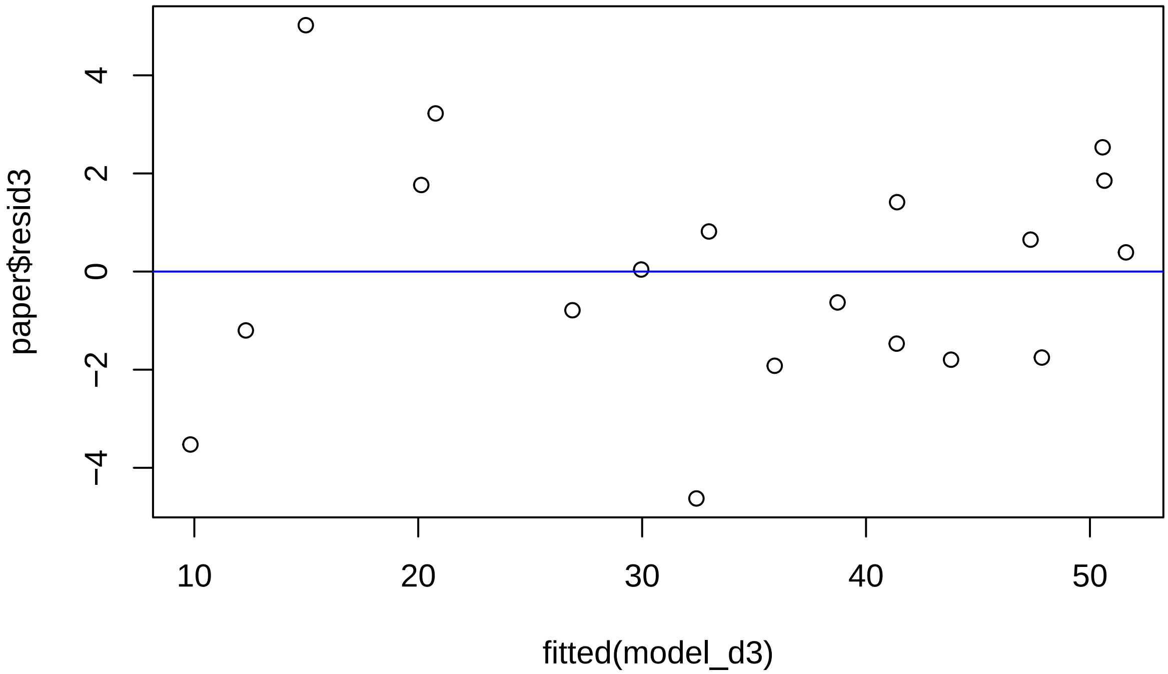
Polynomial regression model of degree 3

```
##
## Call:
## lm(formula = tensile ~ poly(hardwood, 3, raw = TRUE), data = paper)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.6250 -1.6109  0.0413  1.5892  5.0216
##
## Coefficients:
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     5.648395   2.954663   1.912   0.0752 .
## poly(hardwood, 3, raw = TRUE)1  3.578489   1.565854   2.285   0.0373 *
## poly(hardwood, 3, raw = TRUE)2  0.653635   0.231330   2.826   0.0128 *
## poly(hardwood, 3, raw = TRUE)3 -0.055188   0.009789  -5.638 4.72e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.585 on 15 degrees of freedom
## Multiple R-squared:  0.9707, Adjusted R-squared:  0.9648
## F-statistic: 165.4 on 3 and 15 DF,  p-value: 1.025e-11
```

```
# scatter plot of the data with the estimated curve overlaid on top
ggplot(data = paper, mapping = aes(x = hardwood, y = tensile))+
  geom_point() +
  geom_smooth(method = "lm", formula = y ~ poly(x, 3, raw = TRUE))
```



```
# plot of the residuals vs. fitted values
paper <- paper %>%
  mutate(resid3 = residuals(model_d3))
plot(fitted(model_d3), paper$resid3)
abline(0,0, col="blue")
```



```r
# residual sum of squares (RSS)
(RSS <- 2.585^2 * 15)
```

```
## [1] 100.2334
```

(b) Based on your results above, which model do you prefer?

Since we want the RSS to be as small as possible, we would prefer the model of degree 3. Also, the estimated curve of this model overlay more of the observations than the model of degree 2.

(c) Extract the model matrix from your degree 2 polynomial fit, and use it to find the coefficient estimates $\hat{\beta}$ and the fitted values $\hat{y}$ through direct matrix manipulations.

```r
X <- model.matrix(model_d2)
y <- matrix(paper$tensile)
(beta_hat <- solve( t(X) %*% X) %*% t(X) %*% y) # beta_hat = (X'X)^-1 X'y
```

```
##                                   [,1]
## (Intercept)                 -6.6741916
## poly(hardwood, 2, raw = TRUE)1 11.7640057
## poly(hardwood, 2, raw = TRUE)2 -0.6345492
```

```r
(y_hat <- X %*% beta_hat) # y_hat = X * beta_hat
```

```
##         [,1]
## 1   4.455265
## 2   9.544081
## 3  14.315623
## 4  22.906883
## 5  30.229044
## 6  33.414213
## 7  36.282107
## 8  38.832727
## 9  41.066072
## 10 42.982143
## 11 44.580939
## 12 46.826707
## 13 47.803377
## 14 47.510948
## 15 45.949421
## 16 43.118796
## 17 39.019072
## 18 33.650250
## 19 27.012330
```