Lab03 - Multiple Regression and Transformations

Your Name Here

Country Data

We have data on 157 countries, with the following variables recorded (the data were assembled in 2012, so all values are for the countries as of that year):

- `region`: Region of the world: Africa, Asia, Caribbean, Europe, Latin Amer, North America, NorthAtlantic, Oceania.
- `group`: A factor with levels oecd for countries that are members of the OECD, the Organization for Economic Co-operation and Development, as of May 2012, africa for countries on the African continent, and other for all other countries. No OECD countries are located in Africa.
- `fertility`: Total fertility rate, number of children per woman.
- `ppgdp`: Per capita gross domestic product in US dollars.
- `lifeExpF`: Female life expectancy, years.
- `pctUrban`: Percent urban.
- `infantMortality`: Infant deaths by age 1 year per 1000 live births

Let's develop models to predict `infantMortality` as a function of the other variables.

The code below reads in the data and sets up a split into 10 cross-validation folds for you to use below. I'm doing this so that everyone is working with the same cross-validation folds.

```
countries <- read_csv("data/countries.csv")
```

```
## Parsed with column specification:
## cols(
##   region = col_character(),
##   group = col_character(),
##   fertility = col_double(),
##   ppgdp = col_double(),
##   lifeExpF = col_double(),
##   pctUrban = col_double(),
##   infantMortality = col_double()
## )
```

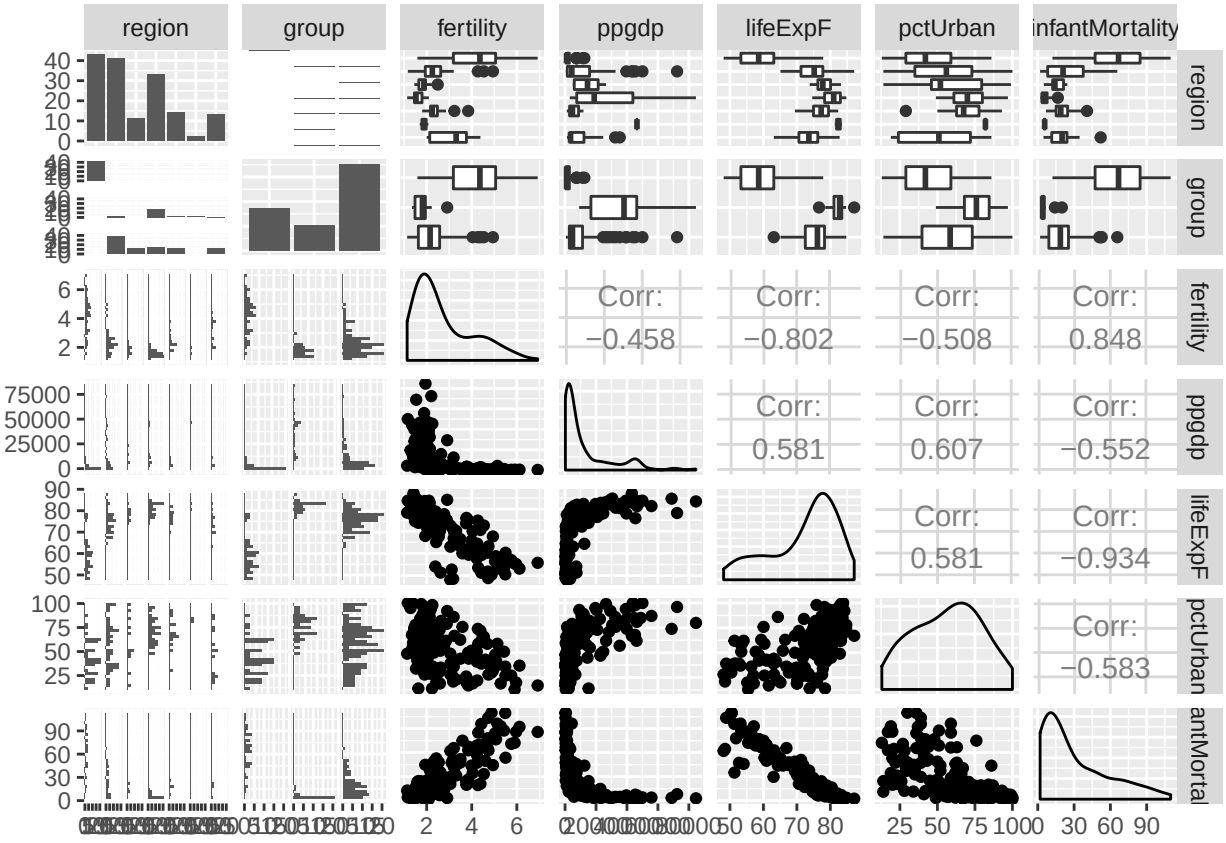
```
set.seed(98448)
val_folds <- caret::createFolds(countries$infantMortality, k = 10)
```

1. Make some exploratory plots of the data.

```
ggpairs(countries)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



2. Fit a model that includes `region` and `group` as explanatory variables and take a look at the summary output. What is going on? (I honestly did not plan for this issue to arise with this data set, it just did.)

```
lm_fit <- lm(infantMortality ~ region + group, data = countries)
summary(lm_fit)
```

```
##
## Call:
## lm(formula = infantMortality ~ region + group, data = countries)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -52.114  -7.570   0.077   6.732  45.592
##
## Coefficients: (1 not defined because of singularities)
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)             64.226      2.566  25.028  < 2e-16 ***
## regionAsia             -39.373      3.688 -10.675  < 2e-16 ***
## regionCaribbean        -49.054      5.686  -8.628 8.68e-15 ***
## regionEurope           -53.085      4.622 -11.486  < 2e-16 ***
## regionLatin Amer       -42.426      5.219  -8.130 1.54e-13 ***
## regionNorth America    -49.468     13.000  -3.805 0.000206 ***
## regionOceania          -42.834      5.338  -8.025 2.79e-13 ***
```

```
## groupoecd             -9.065      4.563  -1.987 0.048805 *
## groupother                NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.83 on 149 degrees of freedom
## Multiple R-squared:  0.6477, Adjusted R-squared:  0.6311
## F-statistic: 39.13 on 7 and 149 DF,  p-value: < 2.2e-16
```

The NA estimate is a sign of multicollinearity. Looking back at the variable descriptions, we see that there is an "Africa" level for region and an "africa" level for the group. Those variables contain duplicate information in that case.

```
group_africa_inds <- which(countries$group == "africa")
region_africa_inds <- which(countries$region == "Africa")
identical(group_africa_inds, region_africa_inds)
```

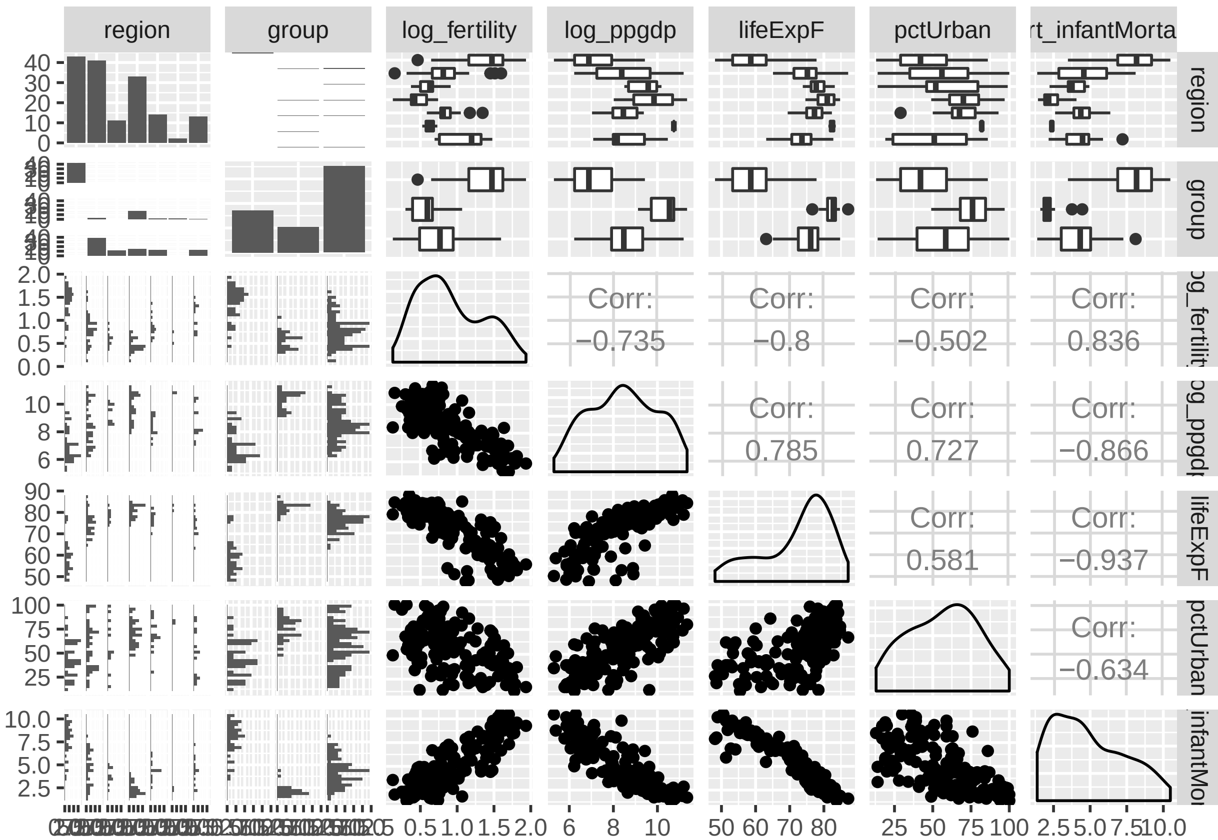
```
## [1] TRUE
```

3. Develop a predictive model by transforming the response and/or quantitative explanatory variables so that the associations between the transformed variables are approximately linear. Include just one of `region` and `group` in your model.

(a) As a challenge, see if you can make a successful guess about the transformations to use on the first try. Make plots of your transformed variables to see how you're doing. You should feel pretty good about your selected transformations before fitting any models. You only need to keep your final selections for transformations, no need to keep any intermediate steps.

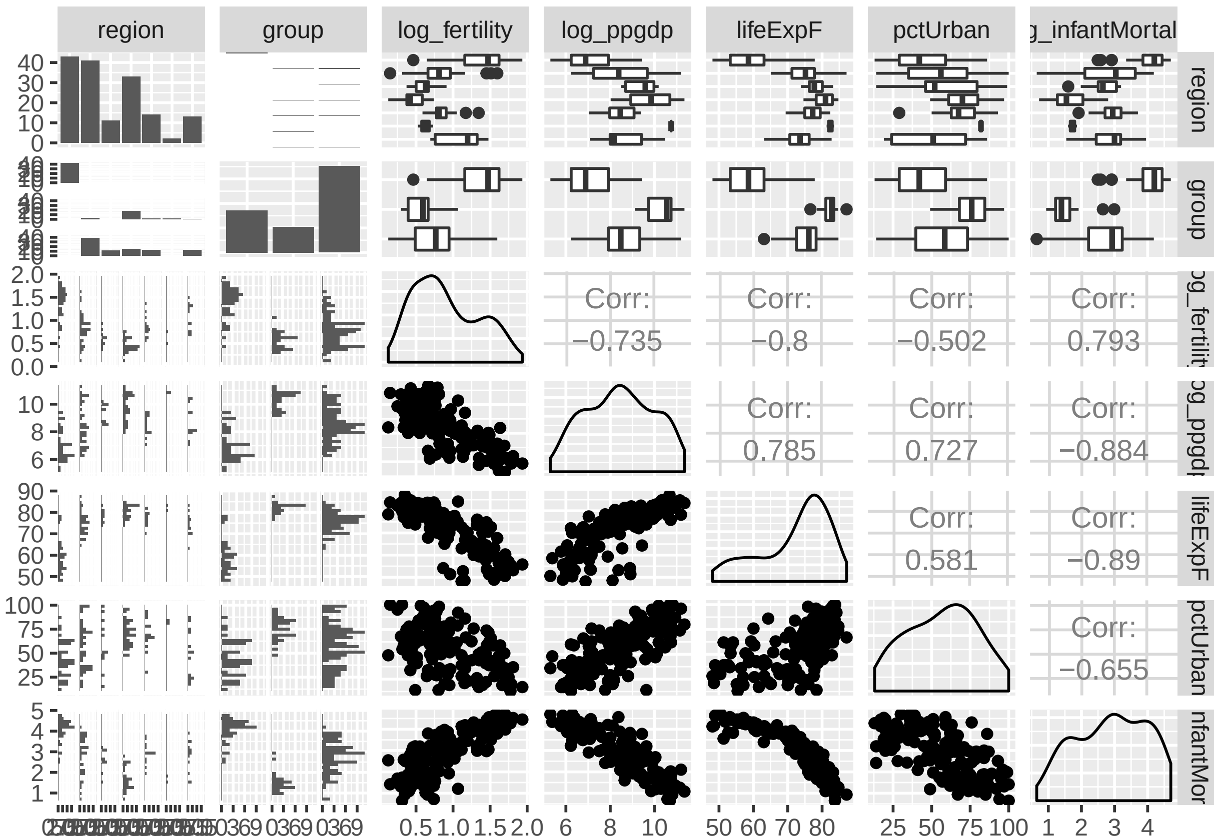
```r
countries_transformed <- countries %>%
  transmute(
    region = region,
    group = group,
    log_fertility = log(fertility),
    log_ppgdp = log(ppgdp),
    lifeExpF = lifeExpF,
    pctUrban = pctUrban,
    sqrt_infantMortality = sqrt(infantMortality)
  )
ggpairs(countries_transformed)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



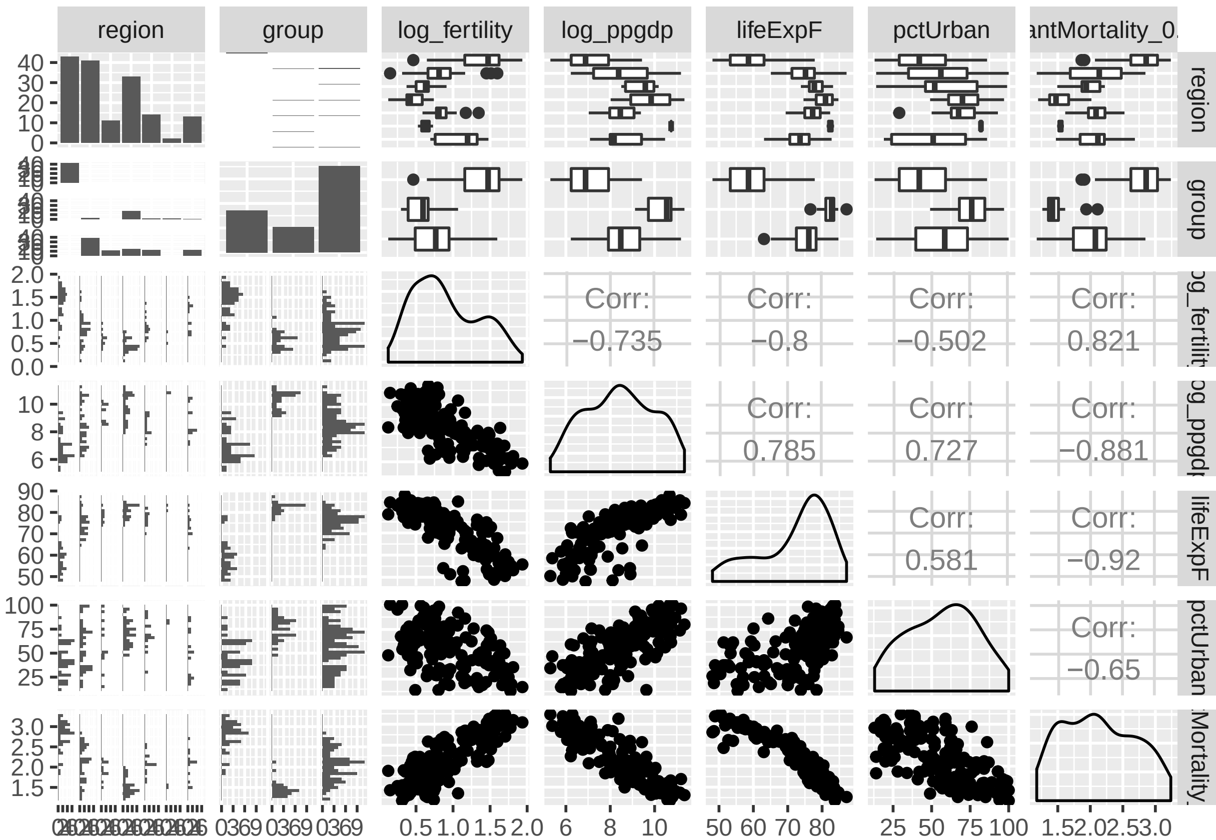
```
countries_transformed <- countries %>%
  transmute(
    region = region,
    group = group,
    log_fertility = log(fertility),
    log_ppgdp = log(ppgdp),
    lifeExpF = lifeExpF,
    pctUrban = pctUrban,
    log_infantMortality = log(infantMortality)
  )
ggpairs(countries_transformed)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



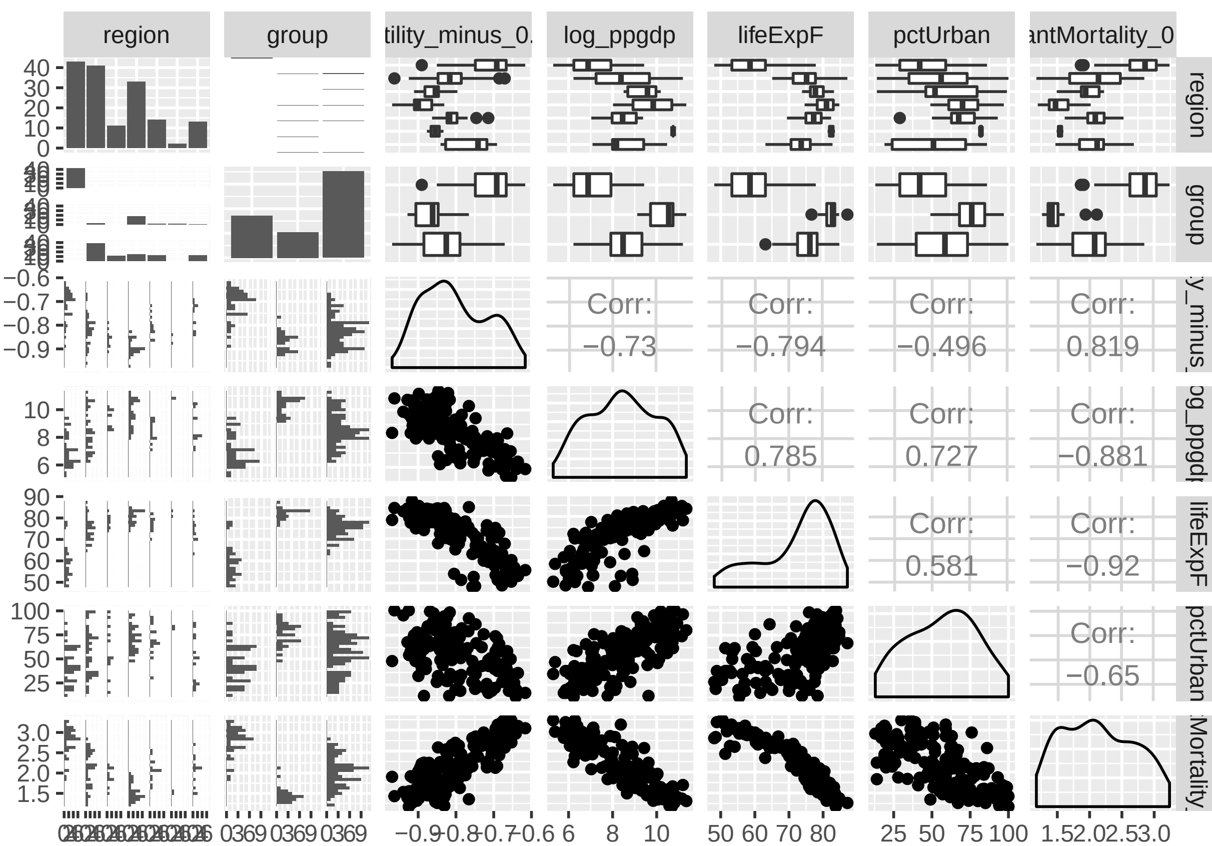
```r
countries_transformed <- countries %>%
  transmute(
    region = region,
    group = group,
    log_fertility = log(fertility),
    log_ppgdp = log(ppgdp),
    lifeExpF = lifeExpF,
    pctUrban = pctUrban,
    infantMortality_0.25 = infantMortality^0.25
  )
ggpairs(countries_transformed)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
countries_transformed <- countries %>%
  transmute(
    region = region,
    group = group,
    fertility_minus_0.25 = -1/(fertility^0.25),
    log_ppgdp = log(ppgdp),
    lifeExpF = lifeExpF,
    pctUrban = pctUrban,
    infantMortality_0.25 = infantMortality^0.25
  )
ggpairs(countries_transformed)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

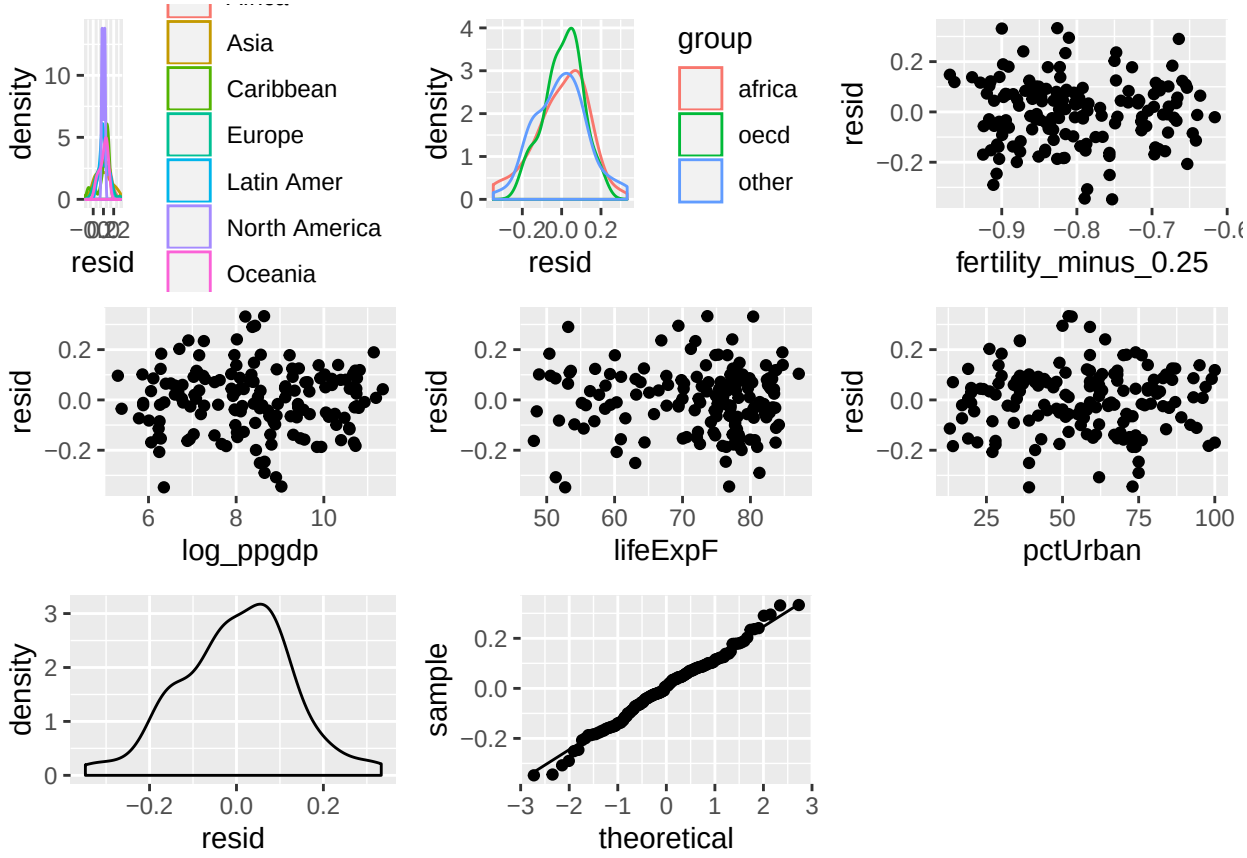


- I don't think additional transformations of infant mortality can simultaneously improve the relationships with fertility and pctUrban; the heteroskedasticity is kindof going in opposite directions in those two plots. But it's not too severe in either plot at this point.
- There is a non-linear relationship between lifeExpF and my transformed infant mortality variable, but

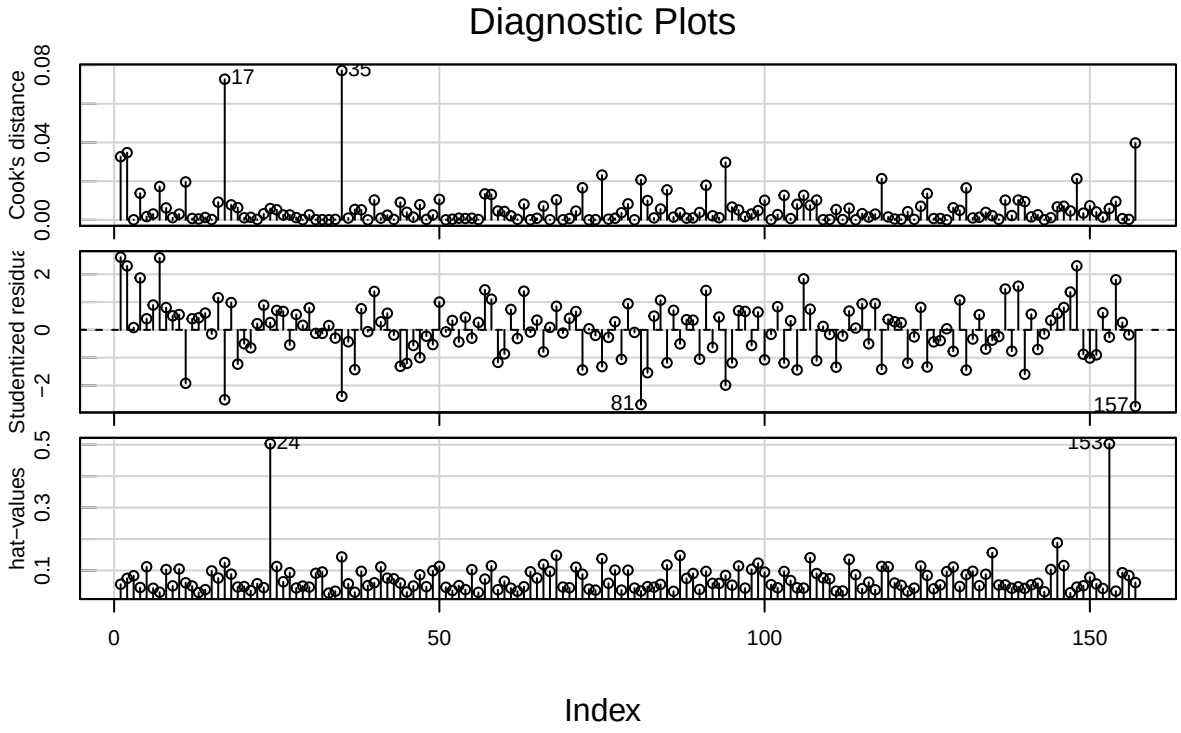
the variance of the response is fairly constant across the range of values for lifeExpF. I will handle that non-linearity with a polynomial term in lifeExpF.

(b) Fit a model to your transformed data and create a set of diagnostic plots. These should include (i) scatter plots of the residuals vs. each quantitative explanatory variable in the data set (whether or not you included it in your model); (ii) a density plot or histogram of the residuals; (iii) a QQ plot of the residuals; and (iv) diagnostic plots of Cook's distance, studentized residuals, and leverage. (See my solutions to lab 2 for examples of how to make these plots.) If you see any serious issues, go back to step (a) and try additional transformations.

```
lm_fit <- lm(infantMortality_0.25 ~ region + fertility_minus_0.25 + log_ppgdp + poly(lifeExpF, 2) + pctU

countries_transformed <- countries_transformed %>%
  mutate(
    resid = residuals(lm_fit)
  )

p1 <- ggplot(data = countries_transformed, mapping = aes(x = resid, color = region)) +
  geom_density()

p2 <- ggplot(data = countries_transformed, mapping = aes(x = resid, color = group)) +
  geom_density()

p3 <- ggplot(data = countries_transformed, mapping = aes(x = fertility_minus_0.25, y = resid)) +
  geom_point()

p4 <- ggplot(data = countries_transformed, mapping = aes(x = log_ppgdp, y = resid)) +
  geom_point()

p5 <- ggplot(data = countries_transformed, mapping = aes(x = lifeExpF, y = resid)) +
  geom_point()

p6 <- ggplot(data = countries_transformed, mapping = aes(x = pctUrban, y = resid)) +
  geom_point()

p7 <- ggplot(data = countries_transformed, mapping = aes(x = resid)) +
  geom_density()

p8 <- ggplot(data = countries_transformed, mapping = aes(sample = resid)) +
  geom_qq() +
  geom_qq_line()

grid.arrange(p1, p2, p3, p4, p5, p6, p7, p8)
```




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

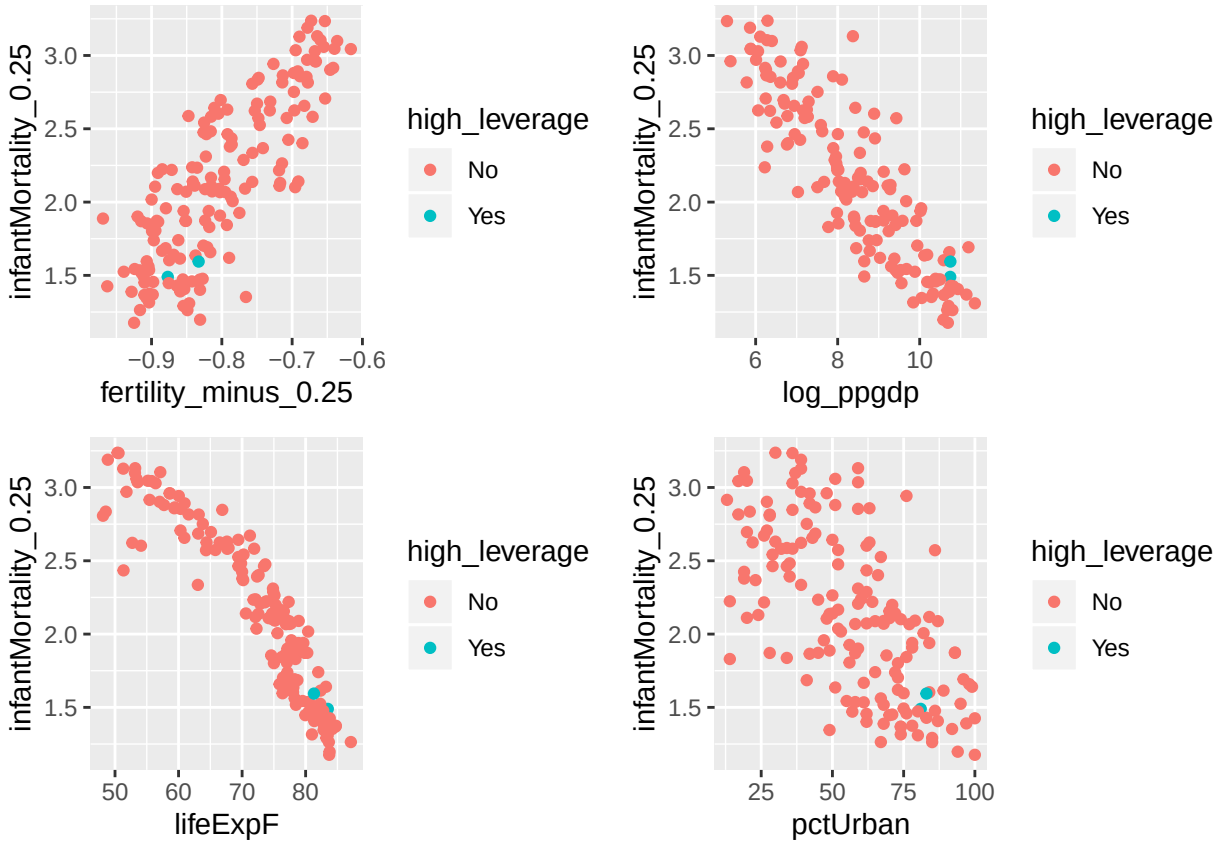


```r
2 * length(coef(lm_fit)) / nrow(countries_transformed) # threshold for when we have to worry about leve
```

```
## [1] 0.1528662
```

Observations 24 and 153 show high leverage. Let's look at plots to see if we're worried:

```r
countries_transformed$high_leverage <- "No"
countries_transformed$high_leverage[c(24, 153)] <- "Yes"

p1 <- ggplot(data = countries_transformed, mapping = aes(x = fertility_minus_0.25, y = infantMortality_
  geom_point()

p2 <- ggplot(data = countries_transformed, mapping = aes(x = log_ppgdp, y = infantMortality_0.25, color
  geom_point()

p3 <- ggplot(data = countries_transformed, mapping = aes(x = lifeExpF, y = infantMortality_0.25, color =
  geom_point()

p4 <- ggplot(data = countries_transformed, mapping = aes(x = pctUrban, y = infantMortality_0.25, color =
  geom_point()

grid.arrange(p1, p2, p3, p4)
```



Not particularly worried.

```r
lm_fit_no_high_leverage <- lm(infantMortality_0.25 ~ region + fertility_minus_0.25 + log_ppgdp + poly(li

summary(lm_fit)
```

```
##
## Call:
## lm(formula = infantMortality_0.25 ~ region + fertility_minus_0.25 +
##     log_ppgdp + poly(lifeExpF, 2) + pctUrban, data = countries_transformed)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.34767 -0.08204  0.00823  0.08393  0.33315
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            3.3557045  0.2023757  16.582  < 2e-16 ***
## regionAsia             0.0060007  0.0445966   0.135   0.8931
## regionCaribbean        0.1099764  0.0609902   1.803   0.0734 .
## regionEurope          -0.0982565  0.0554680  -1.771   0.0786 .
## regionLatin Amer       0.1088302  0.0550768   1.976   0.0501 .
## regionNorth America    0.0722279  0.1071672   0.674   0.5014
## regionOceania         -0.0944230  0.0540163  -1.748   0.0826 .
## fertility_minus_0.25   0.6085707  0.2445627   2.488   0.0140 *
## log_ppgdp             -0.0800790  0.0164913  -4.856 3.07e-06 ***
## poly(lifeExpF, 2)1    -4.4699095  0.3373754 -13.249  < 2e-16 ***
## poly(lifeExpF, 2)2    -0.9294674  0.1695197  -5.483 1.81e-07 ***
## pctUrban              -0.0007676  0.0007408  -1.036   0.3019
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1331 on 145 degrees of freedom
## Multiple R-squared:  0.947,  Adjusted R-squared:  0.943
## F-statistic: 235.5 on 11 and 145 DF,  p-value: < 2.2e-16
```

```
summary(lm_fit_no_high_leverage)
```

```
##
## Call:
## lm(formula = infantMortality_0.25 ~ region + fertility_minus_0.25 +
##     log_ppgdp + poly(lifeExpF, 2) + pctUrban, data = countries_transformed %>%
##     filter(high_leverage == "No"))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.34735 -0.08338  0.00783  0.08496  0.33303
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            3.3619122  0.2027408  16.582  < 2e-16 ***
## regionAsia             0.0062281  0.0447488   0.139   0.8895
## regionCaribbean        0.1103042  0.0611997   1.802   0.0736 .
## regionEurope          -0.0977533  0.0556799  -1.756   0.0813 .
## regionLatin Amer       0.1091036  0.0552643   1.974   0.0503 .
## regionOceania         -0.0944306  0.0541906  -1.743   0.0835 .
## fertility_minus_0.25   0.6114526  0.2455967   2.490   0.0139 *
## log_ppgdp             -0.0798965  0.0165591  -4.825 3.53e-06 ***
## poly(lifeExpF, 2)1    -4.4284179  0.3355929 -13.196  < 2e-16 ***
## poly(lifeExpF, 2)2    -0.9250190  0.1690619  -5.471 1.93e-07 ***
## pctUrban              -0.0007665  0.0007432  -1.031   0.3041
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1336 on 144 degrees of freedom
## Multiple R-squared:  0.9462, Adjusted R-squared:  0.9425
## F-statistic: 253.3 on 10 and 144 DF,  p-value: < 2.2e-16
```

There are essentially no differences between the model fits with and without the high leverage observations. Not worried at all.

(c) Take a look at the summary output for your chosen model. Which variables would hypothesis tests suggest have a strong relationship with infant mortality rates?

```
summary(lm_fit)
```

```
##
## Call:
## lm(formula = infantMortality_0.25 ~ region + fertility_minus_0.25 +
##     log_ppgdp + poly(lifeExpF, 2) + pctUrban, data = countries_transformed)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.34767 -0.08204  0.00823  0.08393  0.33315
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)            3.3557045  0.2023757  16.582  < 2e-16 ***
## regionAsia             0.0060007  0.0445966   0.135   0.8931
## regionCaribbean        0.1099764  0.0609902   1.803   0.0734 .
## regionEurope          -0.0982565  0.0554680  -1.771   0.0786 .
## regionLatin Amer       0.1088302  0.0550768   1.976   0.0501 .
## regionNorth America    0.0722279  0.1071672   0.674   0.5014
## regionOceania         -0.0944230  0.0540163  -1.748   0.0826 .
## fertility_minus_0.25   0.6085707  0.2445627   2.488   0.0140 *
## log_ppgdp             -0.0800790  0.0164913  -4.856 3.07e-06 ***
## poly(lifeExpF, 2)1    -4.4699095  0.3373754 -13.249  < 2e-16 ***
## poly(lifeExpF, 2)2    -0.9294674  0.1695197  -5.483 1.81e-07 ***
## pctUrban              -0.0007676  0.0007408  -1.036   0.3019
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1331 on 145 degrees of freedom
## Multiple R-squared:  0.947,  Adjusted R-squared:  0.943
## F-statistic: 235.5 on 11 and 145 DF,  p-value: < 2.2e-16
```

There is fairly strong evidence of an association between fertility, ppgdp, and lifeExpF and infant mortality rates. We have to conduct an F test to investigate region:

```
reduced_fit <- lm(infantMortality_0.25 ~ fertility_minus_0.25 + log_ppgdp + poly(lifeExpF, 2) + pctUrban

anova(reduced_fit, lm_fit)
```

```
## Analysis of Variance Table
##
## Model 1: infantMortality_0.25 ~ fertility_minus_0.25 + log_ppgdp + poly(lifeExpF,
##     2) + pctUrban
```

```
## Model 2: infantMortality_0.25 ~ region + fertility_minus_0.25 + log_ppgdp +
##     poly(lifeExpF, 2) + pctUrban
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    151 3.2517
## 2    145 2.5699  6   0.68183 6.4118 5.194e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Yep, a hypothesis test says region is important too. Basically the only variable a hypothesis test says we could drop is pctUrban.

(d) Obtain cross-validated MSE for your selected model using the split into validation folds I set up above. Be careful! You will need to apply your selected transformations to the relevant explanatory variables, use your model to generate predictions for the transformed y, then transform those predictions back to the original scale for the response before calculating the residuals. If you want, see if you can find a model with better cross-validated performance by fine-tuning your model. (You can also skip this fine-tuning step if you want.)

```
val_mse <- rep(NA, 10)

for(i in seq_len(10)) {
  countries_transformed_train <- countries_transformed %>% slice(-val_folds[[i]])
  countries_transformed_val <- countries_transformed %>% slice(val_folds[[i]])
  countries_val <- countries %>% slice(val_folds[[i]])

  lm_fit <- lm(infantMortality_0.25 ~ region + fertility_minus_0.25 + log_ppgdp + poly(lifeExpF, 2) + p

  y_hat_trans <- predict(lm_fit, newdata = countries_transformed_val)
  val_mse[i] <- mean((countries_val$infantMortality - y_hat_trans^4)^2)
}
mean(val_mse)
```

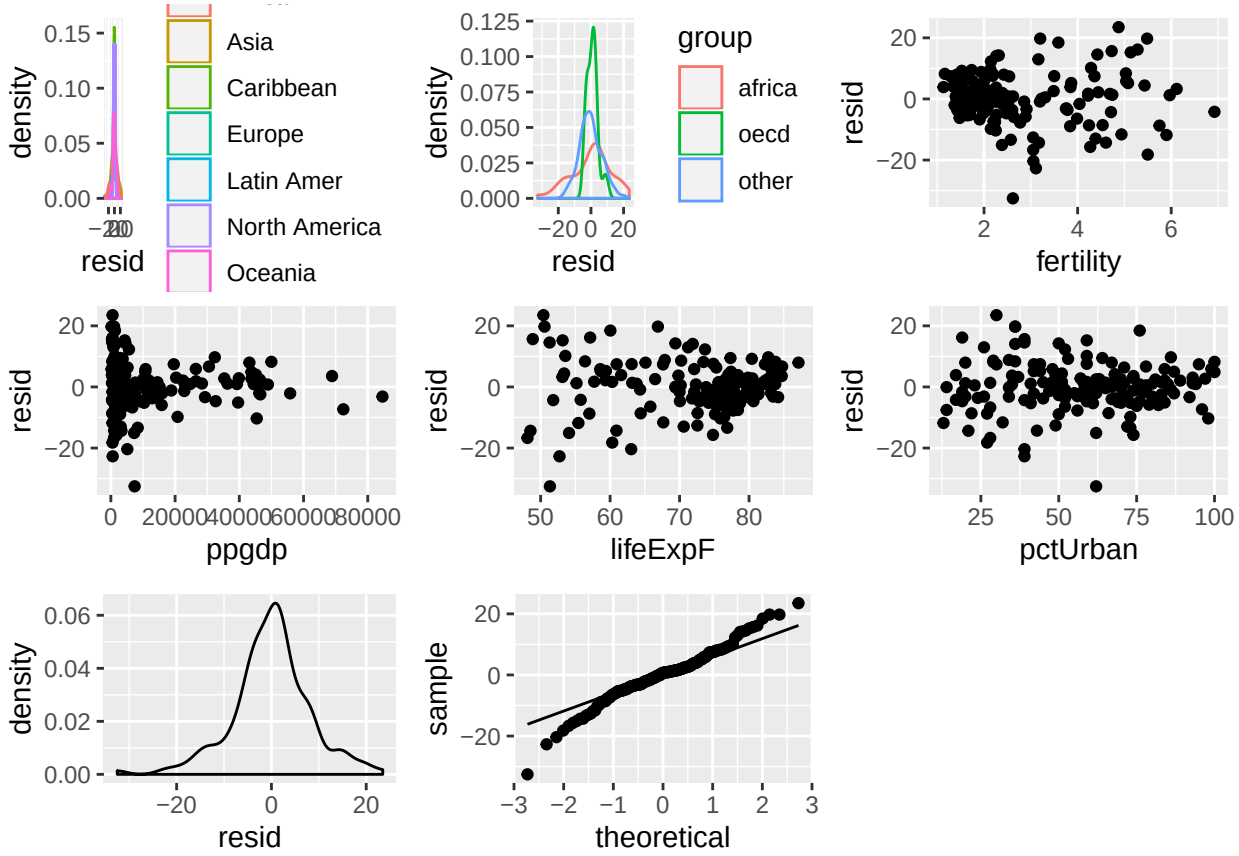
```
## [1] 55.51407
```

4. Develop a predictive model by including polynomial terms in any explanatory variables that have a non-linear relationship with the response. You should not use any transformations of the explanatory or response variables for this model.

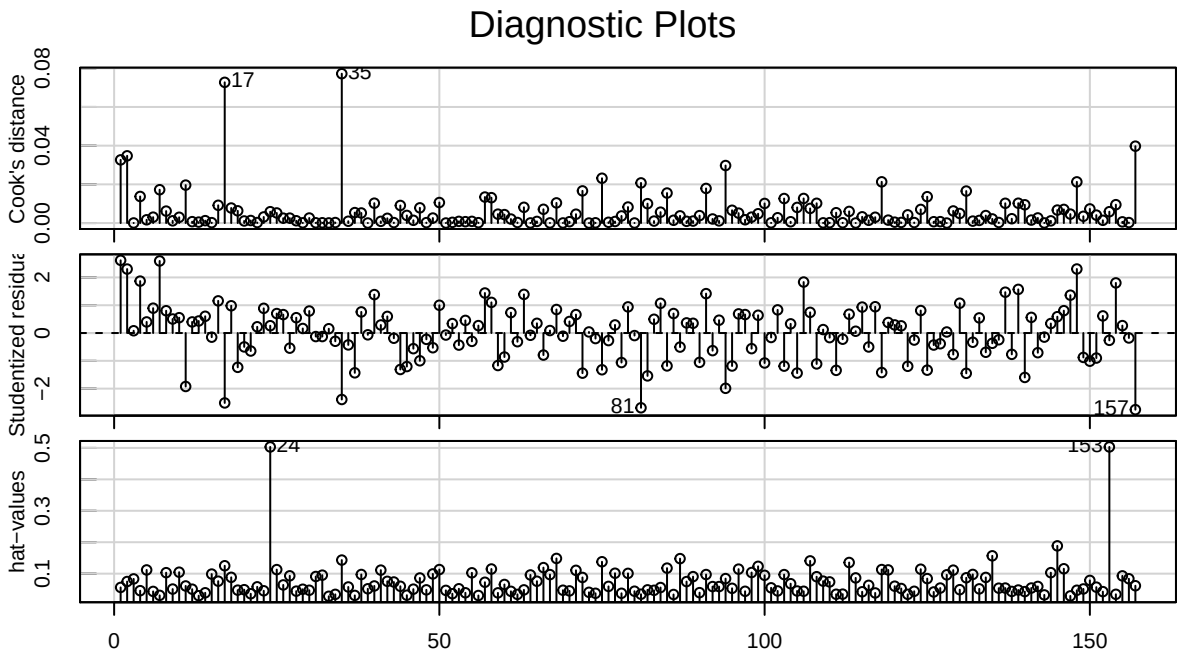
(a) Fit your selected model to the data and create the same diagnostic plots you made for your model in part 3 (b). You will not be able to solve problems with non-constant variance of the residuals (heteroskedasticity), but the residual diagnostic plots should show no signs of non-linearities that are not captured in your model.

```
lm_fit2 <- lm(infantMortality ~ region + fertility + poly(ppgdp, 2) + lifeExpF + poly(pctUrban, 2), data

countries <- countries %>%
  mutate(
    resid = residuals(lm_fit2)
  )

p1 <- ggplot(data = countries, mapping = aes(x = resid, color = region)) +
  geom_density()

p2 <- ggplot(data = countries, mapping = aes(x = resid, color = group)) +
```

```
  geom_density()

p3 <- ggplot(data = countries, mapping = aes(x = fertility, y = resid)) +
  geom_point()

p4 <- ggplot(data = countries, mapping = aes(x = ppgdp, y = resid)) +
  geom_point()

p5 <- ggplot(data = countries, mapping = aes(x = lifeExpF, y = resid)) +
  geom_point()

p6 <- ggplot(data = countries, mapping = aes(x = pctUrban, y = resid)) +
  geom_point()

p7 <- ggplot(data = countries, mapping = aes(x = resid)) +
  geom_density()

p8 <- ggplot(data = countries, mapping = aes(sample = resid)) +
  geom_qq() +
  geom_qq_line()

grid.arrange(p1, p2, p3, p4, p5, p6, p7, p8)
```




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




```
2 * length(coef(lm_fit)) / nrow(countries_transformed) # threshold for when we have to worry about leve
```

```
## [1] 0.1528662
```

There are clear problems, but this is about as good as you can do without using transformations.

(b) Take a look at the summary output for your chosen model. Which variables would hypothesis tests suggest have a strong relationship with infant mortality rates?

```
summary(lm_fit2)
```

```
##
## Call:
## lm(formula = infantMortality ~ region + fertility + poly(ppgdp,
##     2) + lifeExpF + poly(pctUrban, 2), data = countries)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -32.537  -3.962   0.644   4.032  23.495
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          147.3873    12.3746  11.910  < 2e-16 ***
## regionAsia             1.8586     2.8798   0.645   0.5197
## regionCaribbean        3.1120     3.9546   0.787   0.4326
## regionEurope           0.7000     3.5221   0.199   0.8427
## regionLatin Amer       3.2408     3.6701   0.883   0.3787
## regionNorth America    2.3606     6.9940   0.338   0.7362
## regionOceania         -8.1914     3.3855  -2.420   0.0168 *
## fertility              6.5265     0.9865   6.616 6.76e-10 ***
## poly(ppgdp, 2)1        4.4849    15.0253   0.298   0.7658
## poly(ppgdp, 2)2        4.0588    10.4139   0.390   0.6973
```

```
## lifeExpF             -1.8802     0.1657 -11.345  < 2e-16 ***
## poly(pctUrban, 2)1  -21.1090    12.9306  -1.632   0.1048
## poly(pctUrban, 2)2   -4.1743     9.7408  -0.429   0.6689
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.581 on 144 degrees of freedom
## Multiple R-squared:  0.9115, Adjusted R-squared:  0.9041
## F-statistic: 123.5 on 12 and 144 DF,  p-value: < 2.2e-16
```

Neither of the terms for ppgdp are showing up as statistically significant according to individual t tests. Again, we really need an F test:

```
lm_fit2a <- lm(infantMortality ~ region + fertility + lifeExpF + poly(pctUrban, 2), data = countries)
anova(lm_fit2a, lm_fit2)
```

```
## Analysis of Variance Table
##
## Model 1: infantMortality ~ region + fertility + lifeExpF + poly(pctUrban,
##     2)
## Model 2: infantMortality ~ region + fertility + poly(ppgdp, 2) + lifeExpF +
##     poly(pctUrban, 2)
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1    146 10632
## 2    144 10602  2    29.508 0.2004 0.8186
```

- The F test does not indicate strong evidence of a relationship between ppgdp and infant mortality rates, after accounting for the effects of region, fertility, lifeExpF, and pctUrban.
- But note we can't trust this F test! The conditions are not satisfied and F tests are particularly sensitive to the conditions.

(c) Obtain cross-validated MSE for your selected model using the split into validation folds I set up above. There is much less work to do than there was in question 3 since you have not done any transformations this time. If you want, see if you can find a model with better cross-validated performance by fine-tuning your model. (You can also skip this fine-tuning step if you want.)

```
val_mse2 <- rep(NA, 10)

for(i in seq_len(10)) {
  countries_train <- countries %>% slice(-val_folds[[i]])
  countries_val <- countries %>% slice(val_folds[[i]])

  lm_fit <- lm(infantMortality ~ region + fertility + poly(ppgdp, 2) + lifeExpF + poly(pctUrban, 2), da

  y_hat <- predict(lm_fit, newdata = countries_val)
  val_mse2[i] <- mean((countries_val$infantMortality - y_hat)^2)
}
mean(val_mse2)
```

```
## [1] 67.877
```

5. Are your hypothesis test results consistent between the models you developed in parts 3 and 4? Which model would you prefer if you had to conduct some hypothesis tests about which variables have a strong association with infant mortality rates? Why?

No. My model with transformations provided strong evidence of a relationship between ppgdp and infant mortality rates after accounting for the other covariates, but the model without transformations did not.

The hypothesis tests from the model without transformations cannot be trusted because the conditions for inference were not satisfied.

6. Which of your models above would you prefer if you had to make a prediction for the infant mortality rate in a test set country? Why?

The model with transformations had a cross-validated MSE of 55.5, and the model without transformations had a cross-validated MSE of 67.9. We prefer the model with lower cross-validated MSE.

Here's a summary across all the models people submitted:

My model:

```
lm(infantMortality_0.25 ~ region + fertility_minus_0.25 + log_ppgdp + poly(lifeExpF, 2) + pctUrban, data
```

Cross-validated MSE: 55.5

Group 1:

```
lm(infantMortality_log ~ fertility_log + ppgdp_log + pctUrban_sq, data = train_countries)
```

Cross-validated MSE: 223.1

Group 2:

```
lm(sqrtMort ~ lgFert + lgGdp + lifeExpF, data = train)
```

Cross-validated MSE: 66.7

Group 3:

```
lm(log(infantMortality) ~ log(fertility) + group,
             data = train_countries)
```

Cross-validated MSE: 171.9