MA678 Homework 2

Jing Xu

9/20/2022

11.5

Residuals and predictions: The folder `Pyth` contains outcome y and predictors x_1, x_2 for 40 data points, with a further 20 points with the predictors but no observed outcome. Save the file to your working directory, then read it into R using `read.table()`.

(a)

Use R to fit a linear regression model predicting y from x_1, x_2, using the first 40 data points in the file. Summarize the inferences and check the fit of your model.

```
library(rosdata)
```

```
##
## Attaching package: 'rosdata'

## The following objects are masked from 'package:rstanarm':
##
##     kidiq, roaches, wells

## The following object is masked from 'package:MASS':
##
##     newcomb
```

```
data(pyth)
fit_1<-lm(y ~ x1 + x2,data = pyth[1:40,])
summary(fit_1)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2, data = pyth[1:40, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9585 -0.5865 -0.3356  0.3973  2.8548
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.31513    0.38769   3.392  0.00166 **
```

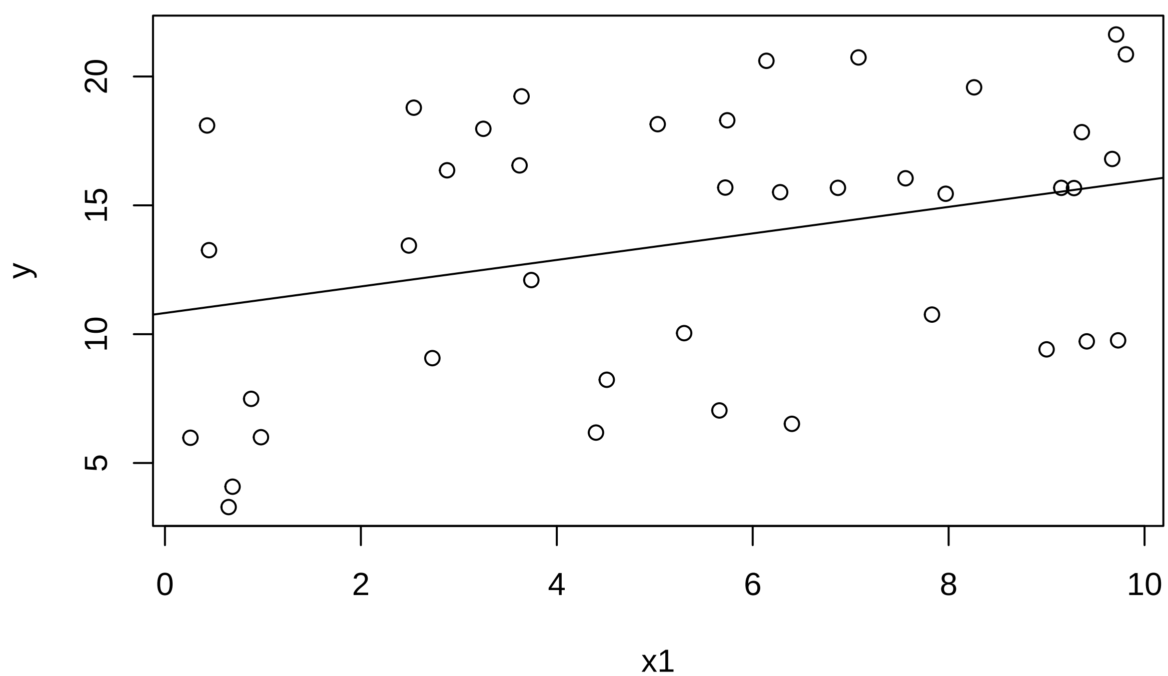
```
## x1           0.51481    0.04590  11.216 1.84e-13 ***
## x2           0.80692    0.02434  33.148  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9 on 37 degrees of freedom
## Multiple R-squared:  0.9724, Adjusted R-squared:  0.9709
## F-statistic: 652.4 on 2 and 37 DF,  p-value: < 2.2e-16
```

```
#both x1 and x2 are significant, R square is also close to 1.
```

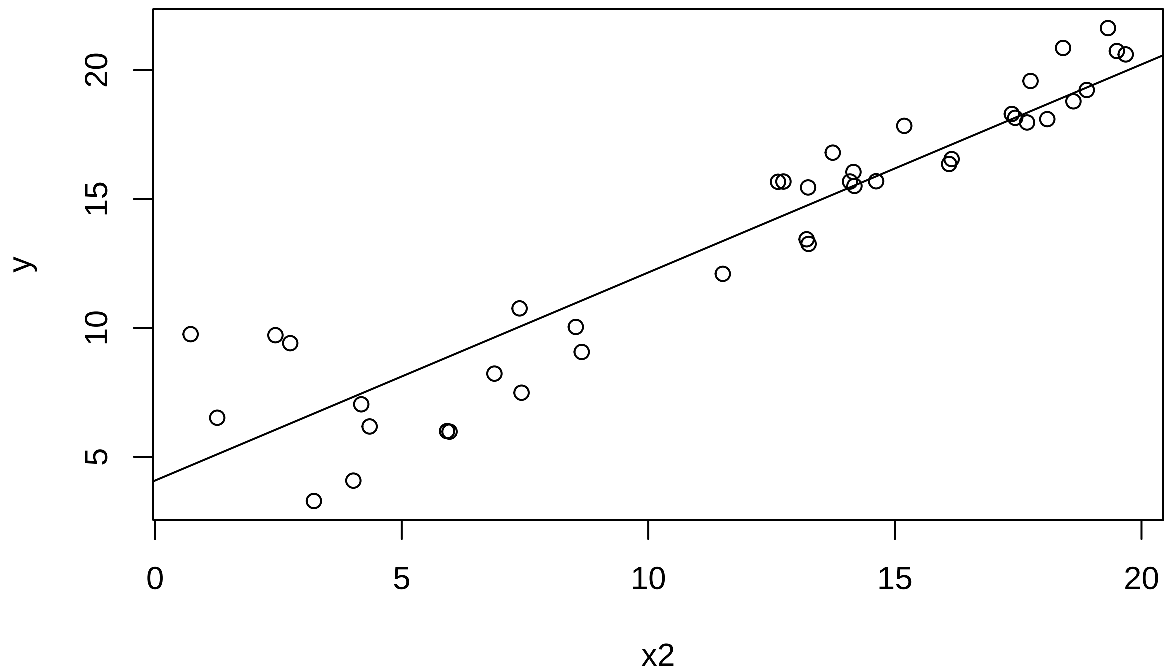
(b)

Display the estimated model graphically as in Figure 10.2

```
plot(pyth$x1[1:40],pyth$y[1:40],xlab = 'x1',ylab = 'y')
abline(coef(fit_1)[1]+coef(fit_1)[3]*mean(pyth$x2[1:40]),coef(fit_1)[2])
```



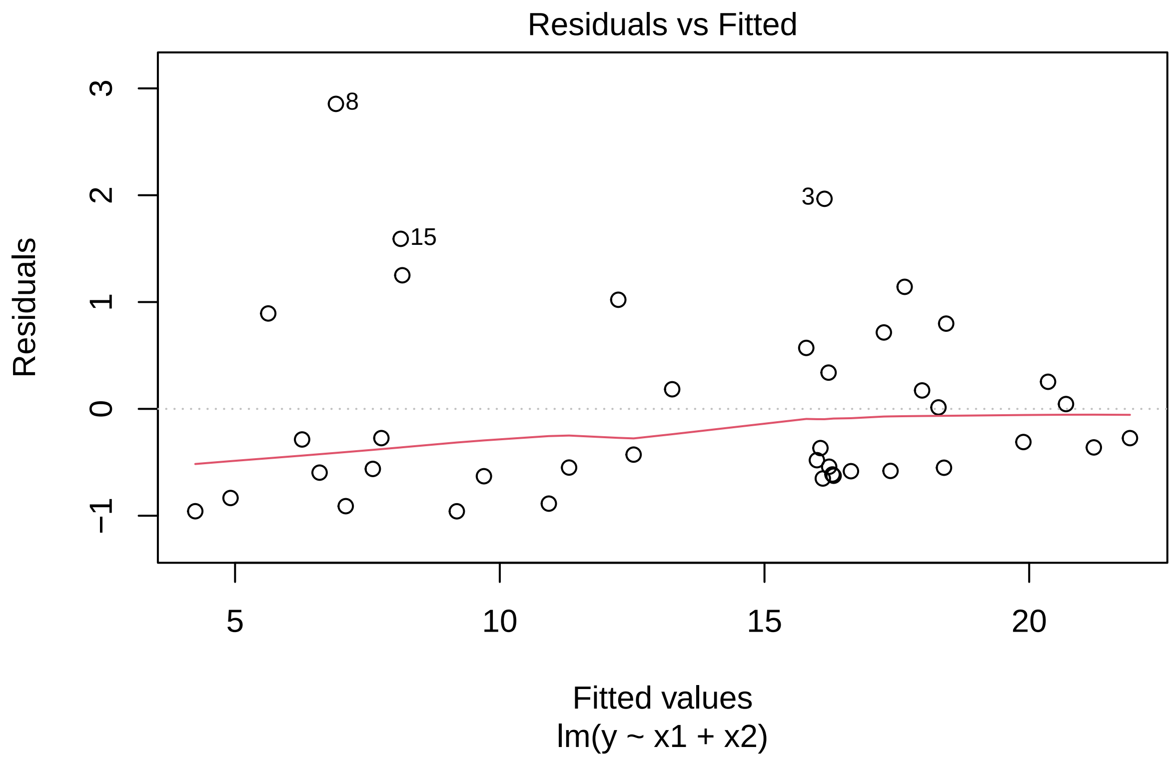

```
plot(pyth$x2[1:40],pyth$y[1:40],xlab = 'x2',ylab = 'y')
abline(coef(fit_1)[1]+coef(fit_1)[2]*mean(pyth$x1[1:40]),coef(fit_1)[3])
```



(c)

Make a residual plot for this model. Do the assumptions appear to be met?

```
plot(fit_1, which = 1)
```



(d)

Make predictions for the remaining 20 data points in the file. How confident do you feel about these predictions?

```
pyth_1 <- data.frame(x1 = pyth$x1[41:60], x2 = pyth$x2[41:60])
predict(fit_1, pyth_1)
```

```
##         1         2         3         4         5         6         7         8
## 14.812484 19.142865  5.916816 10.530475 19.012485 13.398863  4.829144  9.145767
##         9        10        11        12        13        14        15        16
##  5.892489 12.338639 18.908561 16.064649  8.963122 14.972786  5.859744  7.374900
##        17        18        19        20
##  4.535267 15.133280  9.100899 16.084900
```

```
#confidencial result based on a)
```

12.5

Logarithmic transformation and regression: Consider the following regression:

$$\log(\text{weight}) = -3.8 + 2.1 \log(\text{height}) + \text{error},$$

with errors that have standard deviation 0.25. Weights are in pounds and heights are in inches.

(a)

Fill in the blanks: Approximately 68% of the people will have weights within a factor of **0.78**___ and **1.28**___ of their predicted values from the regression.

(b)

Using pen and paper, sketch the regression line and scatterplot of log(weight) versus log(height) that make sense and are consistent with the fitted model. Be sure to label the axes of your graph.

```
data(earnings)
summary(earnings)
```

```
##      height          weight           male             earn
##  Min.   :57.00   Min.   : 80.0   Min.   :0.0000   Min.   :     0
##  1st Qu.:64.00   1st Qu.:130.0   1st Qu.:0.0000   1st Qu.:  6000
##  Median :66.00   Median :150.0   Median :0.0000   Median : 16000
##  Mean   :66.57   Mean   :156.3   Mean   :0.3717   Mean   : 21147
##  3rd Qu.:69.25   3rd Qu.:180.0   3rd Qu.:1.0000   3rd Qu.: 27000
##  Max.   :82.00   Max.   :342.0   Max.   :1.0000   Max.   :400000
##                  NA's   :27
##      earnk         ethnicity           education     mother_education
##  Min.   :  0.00   Length:1816        Min.   : 2.00   Min.   : 3.00
##  1st Qu.:  6.00   Class :character   1st Qu.:12.00   1st Qu.:12.00
##  Median : 16.00   Mode  :character   Median :12.00   Median :13.00
##  Mean   : 21.15                      Mean   :13.24   Mean   :13.61
##  3rd Qu.: 27.00                      3rd Qu.:15.00   3rd Qu.:16.00
##  Max.   :400.00                      Max.   :18.00   Max.   :99.00
##                                      NA's   :2       NA's   :244
##  father_education      walk          exercise        smokenow
##  Min.   : 3.00    Min.   :1.000   Min.   :1.000   Min.   :1.000
##  1st Qu.:12.00    1st Qu.:3.000   1st Qu.:1.000   1st Qu.:1.000
##  Median :13.00    Median :6.000   Median :2.000   Median :2.000
##  Mean   :13.65    Mean   :5.303   Mean   :3.049   Mean   :1.745
##  3rd Qu.:16.00    3rd Qu.:8.000   3rd Qu.:5.000   3rd Qu.:2.000
##  Max.   :99.00    Max.   :8.000   Max.   :7.000   Max.   :2.000
##  NA's   :295                                      NA's   :1
##      tense           angry            age
##  Min.   :0.000   Min.   :0.000   Min.   :18.00
##  1st Qu.:0.000   1st Qu.:0.000   1st Qu.:29.00
##  Median :0.000   Median :0.000   Median :39.00
##  Mean   :1.421   Mean   :1.421   Mean   :42.93
##  3rd Qu.:2.000   3rd Qu.:2.000   3rd Qu.:56.00
##  Max.   :7.000   Max.   :7.000   Max.   :91.00
##  NA's   :1       NA's   :1
```

```
log(57);log(80);log(342);log(66);log(150)
```

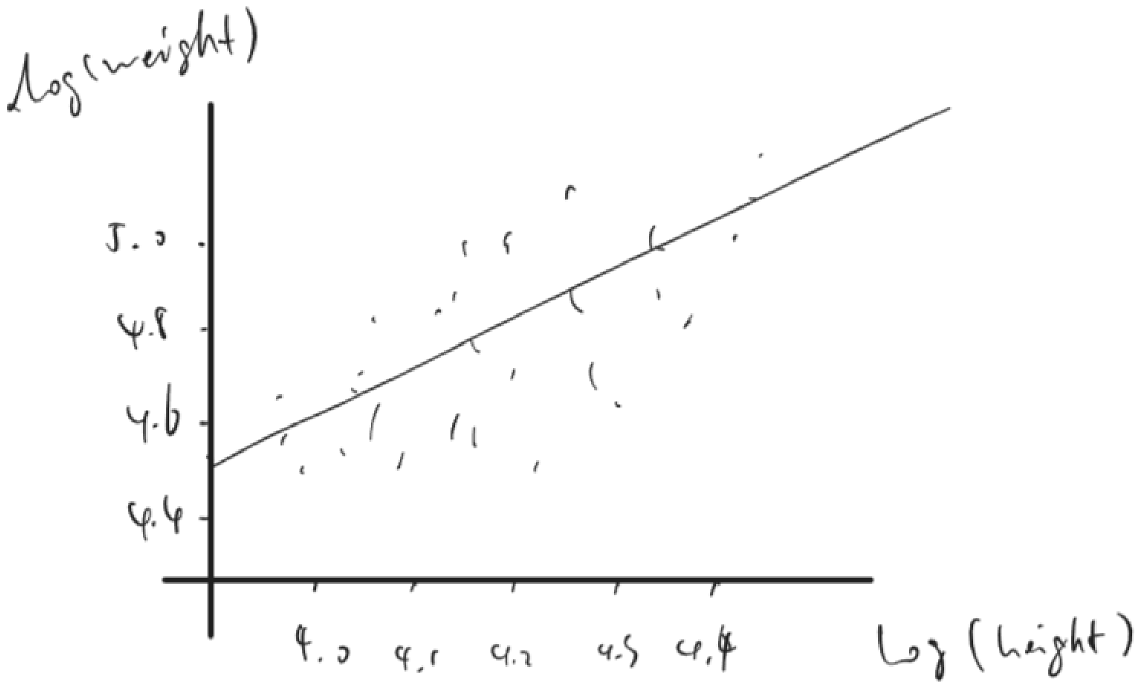
```
## [1] 4.043051
```

```
## [1] 4.382027
```

```
## [1] 5.834811
```

```
## [1] 4.189655
```

```
## [1] 5.010635
```



12.6

Logarithmic transformations: The folder `Pollution` contains mortality rates and various environmental factors from 60 US metropolitan areas. For this exercise we shall model mortality rate given nitric oxides, sulfur dioxide, and hydrocarbons as inputs. this model is an extreme oversimplication, as it combines all sources of mortality and does not adjust for crucial factors such as age and smoking. We use it to illustrate log transformation in regression.

(a)

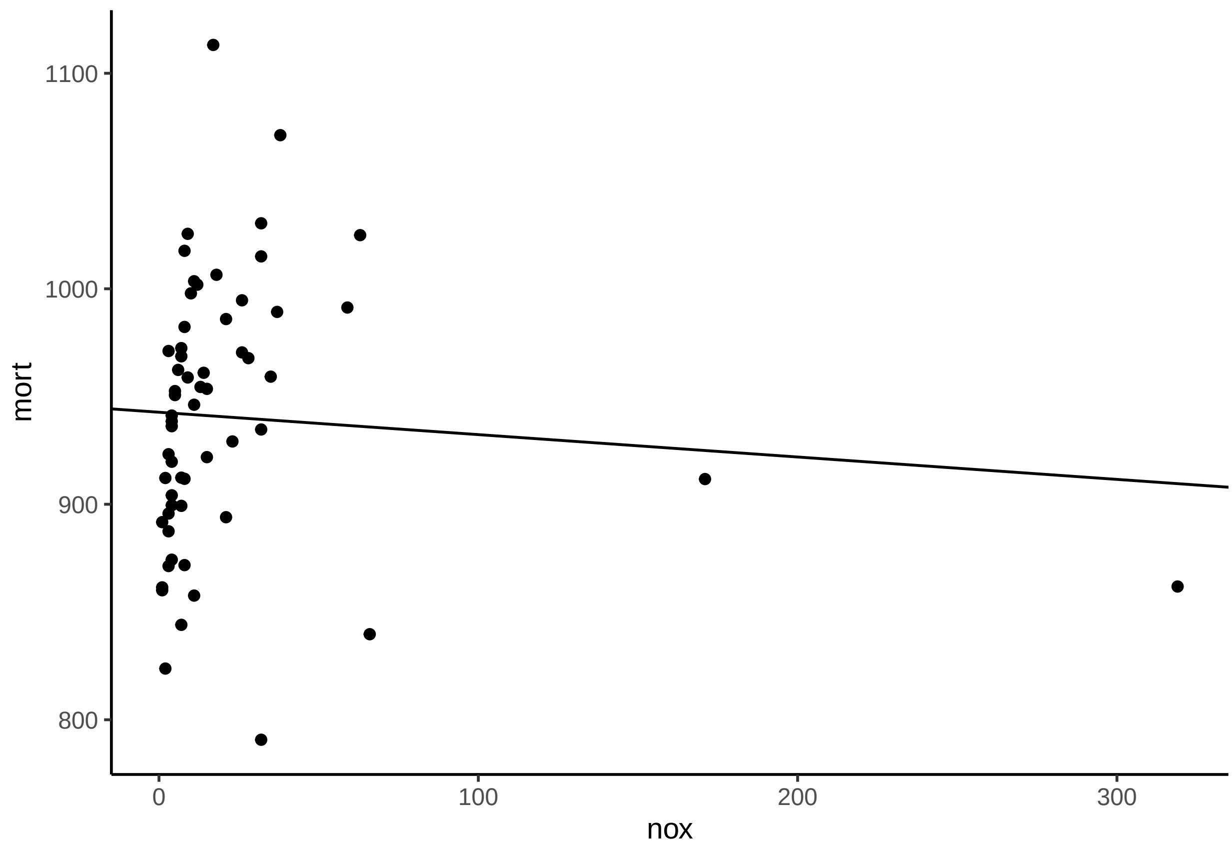
Create a scatterplot of mortality rate versus level of nitric oxides. Do you think linear regression will fit these data well? Fit the regression and evaluate a residual plot from the regression.

```
data(pollution)
head(pollution)
```

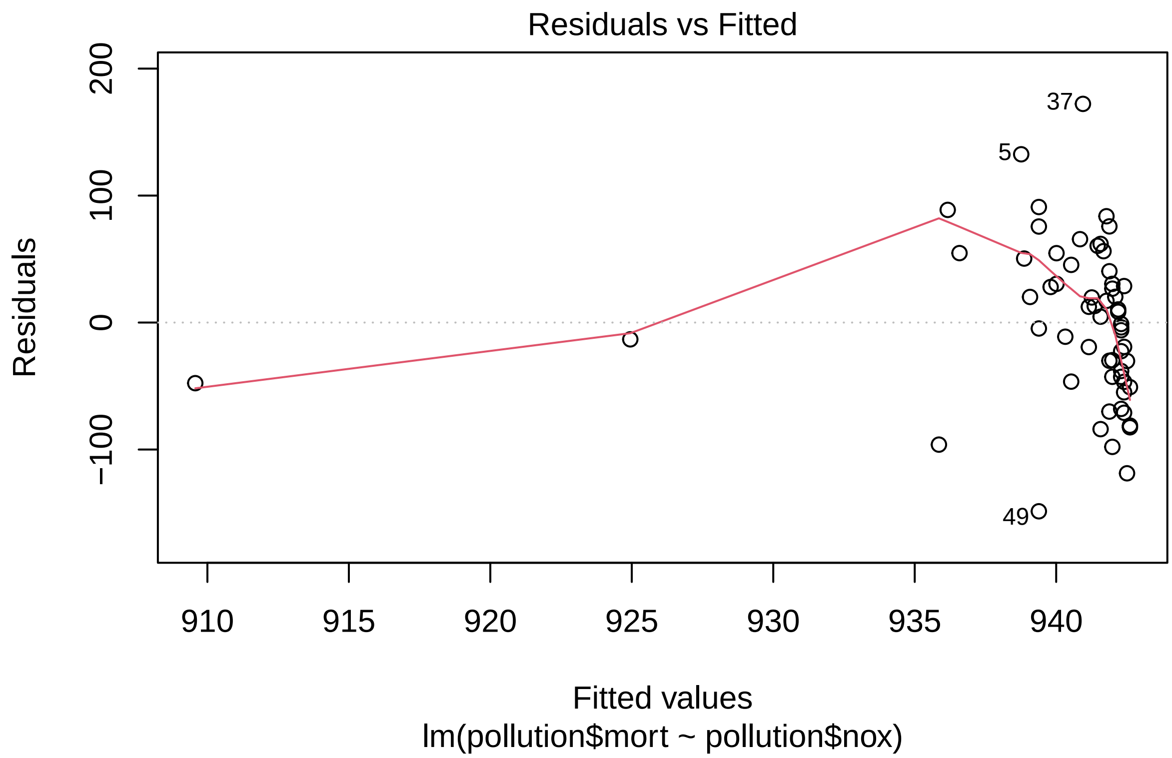
```
##   prec jant jult ovr65 popn educ hous dens nonw wwdrk poor hc nox so2 humid
## 1   36   27   71   8.1 3.34 11.4 81.5 3243  8.8  42.6 11.7 21  15  59    59
## 2   35   23   72  11.1 3.14 11.0 78.8 4281  3.5  50.7 14.4  8  10  39    57
```

```
## 3   44   29   74  10.4 3.21  9.8 81.6 4260  0.8  39.4 12.4  6    6  33    54
## 4   47   45   79   6.5 3.41 11.1 77.5 3125 27.1  50.2 20.6 18   8  24    56
## 5   43   35   77   7.6 3.44  9.6 84.6 6441 24.4  43.7 14.3 43  38 206    55
## 6   53   45   80   7.7 3.45 10.2 66.8 3325 38.5  43.1 25.5 30  32  72    54
##        mort
## 1  921.870
## 2  997.875
## 3  962.354
## 4  982.291
## 5 1071.289
## 6 1030.380
```

```
fit_126 <- lm(pollution$mort ~ pollution$nox)
ggplot(pollution)+
  geom_point(mapping = aes(nox,mort))+
  geom_abline(intercept = coef(fit_126)[1], slope = coef(fit_126)[2]) +
  theme_classic()
```



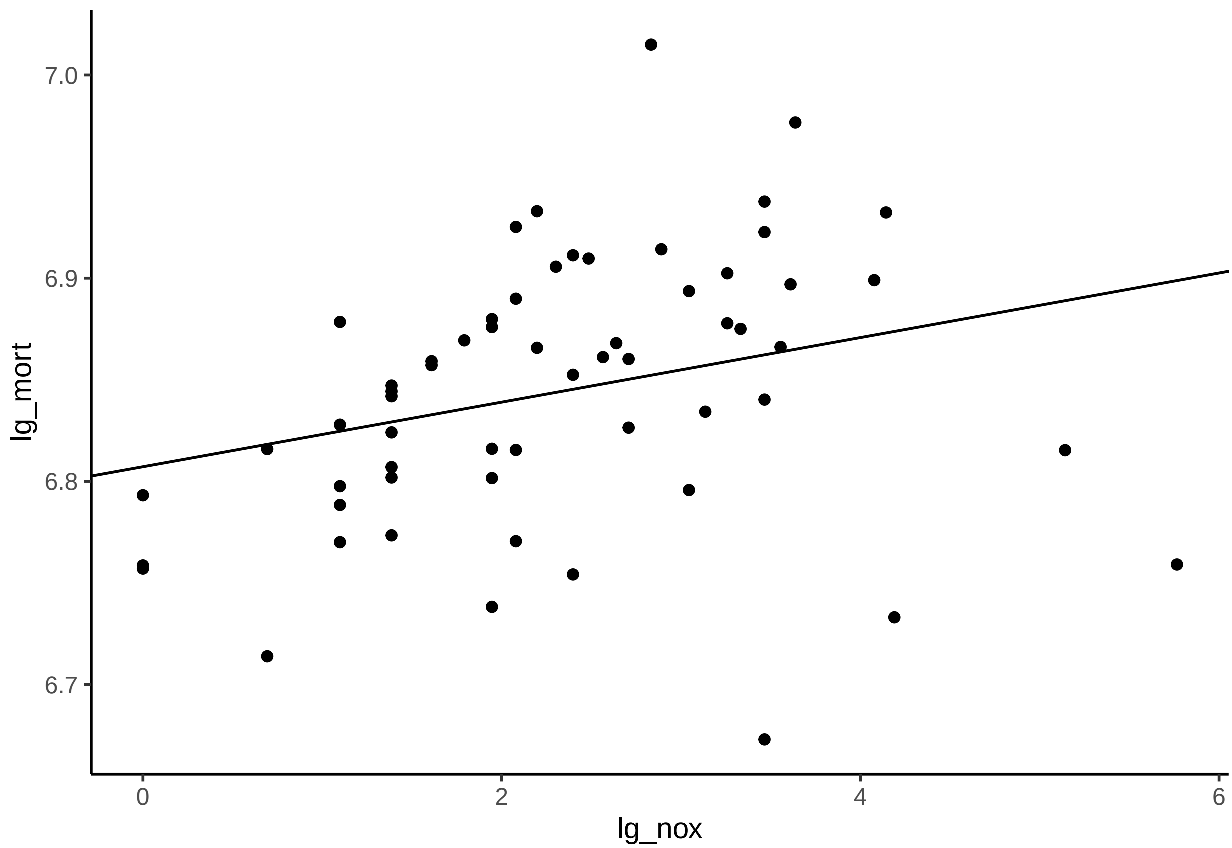

```
plot(fit_126,which = 1)
```



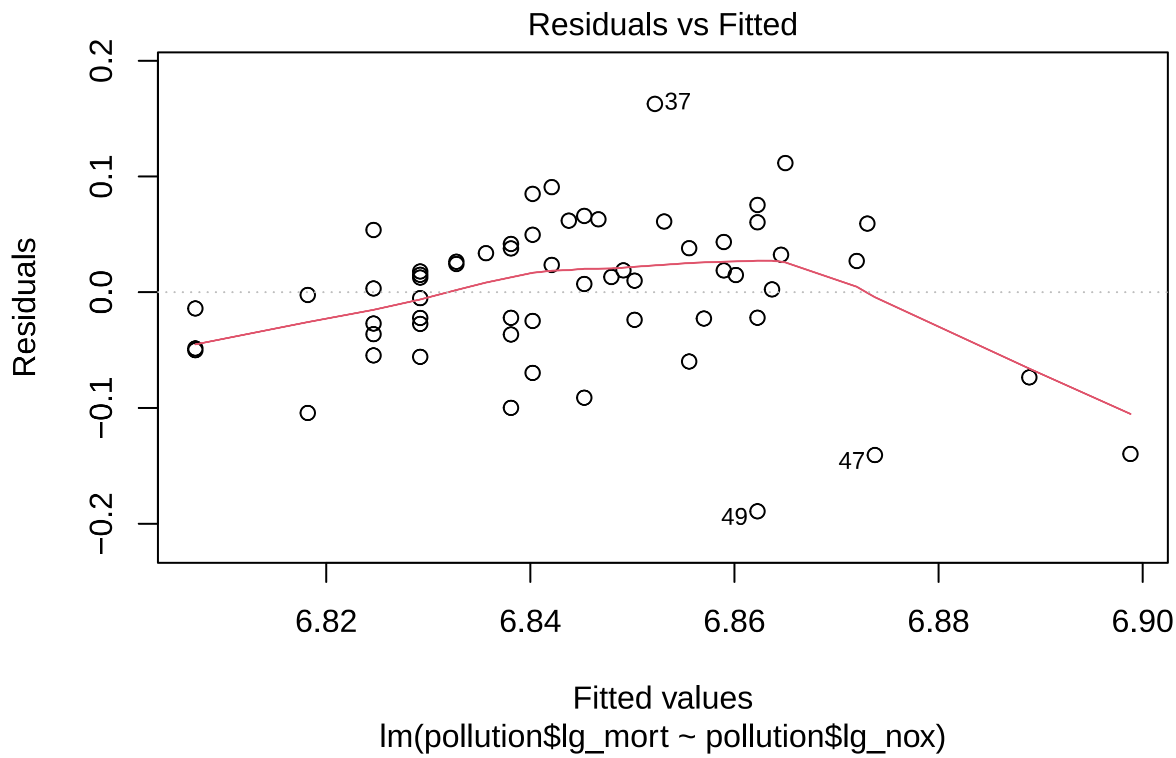
(b)

Find an appropriate reansformation that will result in data more appropriate for linear regression. Fit a regression to the transformed data and evaluate the new residual plot.

```
pollution$lg_mort <- log(pollution$mort)
pollution$lg_nox <- log(pollution$nox)
fit_126b <- lm(pollution$lg_mort ~ pollution$lg_nox)
ggplot(pollution)+
  geom_point(mapping = aes(lg_nox,lg_mort))+
  geom_abline(intercept = coef(fit_126b)[1], slope = coef(fit_126b)[2]) +
  theme_classic()
```




```r
plot(fit_126b,which = 1)
```



(c)

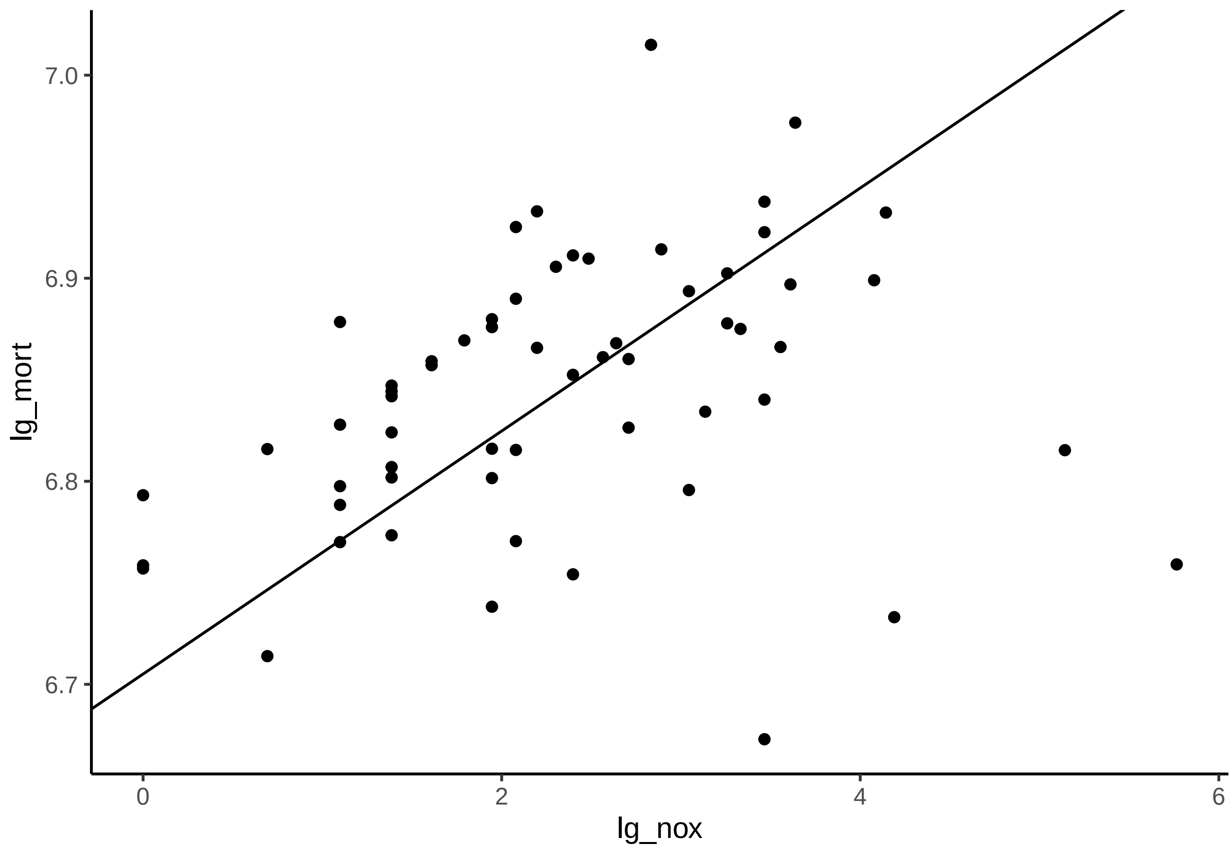
Interpret the slope coefficient from the model you chose in (b)

A 1% increase in height will averagely results in 0.15% increase in weight.

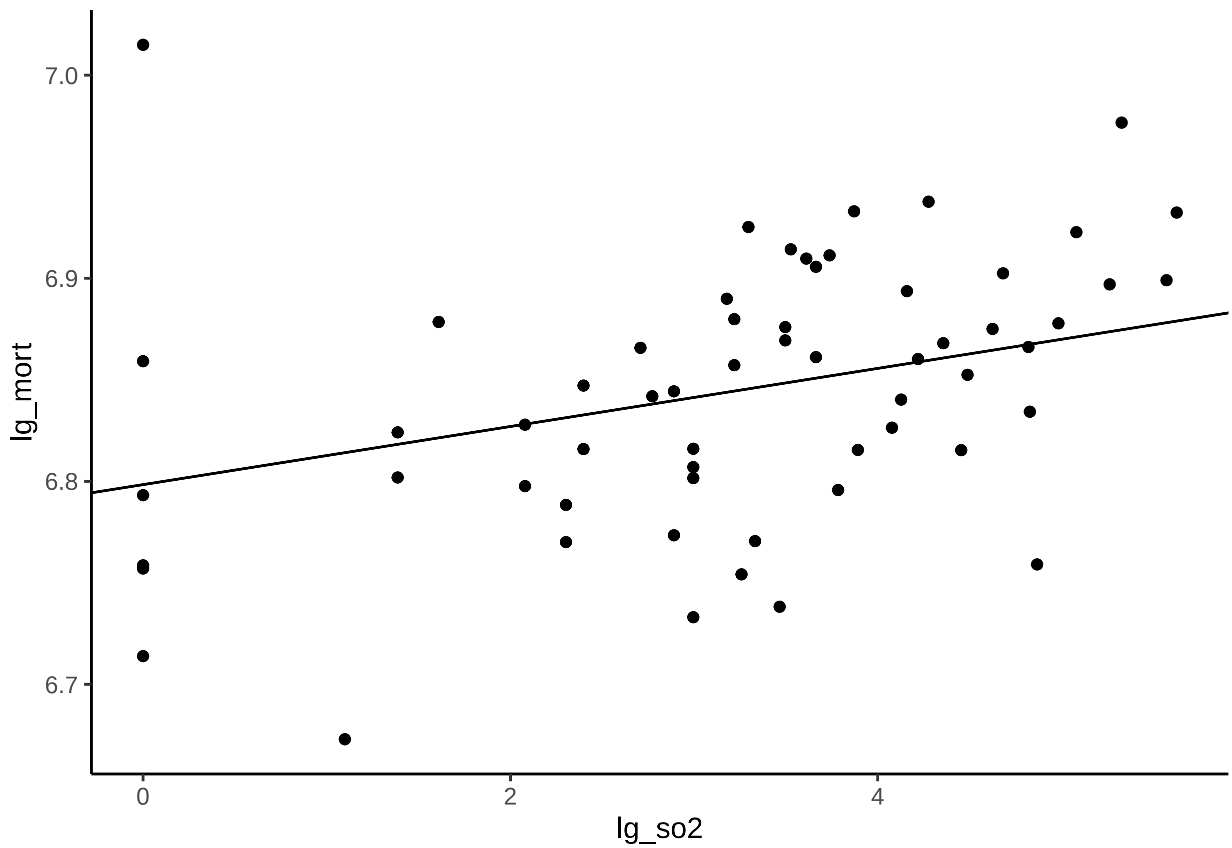
(d)

Now fit a model predicting mortality rate using levels of nitric oxides, sulfur dioxide, and hydrocarbons as inputs. Use appropriate transformation when helpful. Plot the fitted regression model and interpret the coefficients.

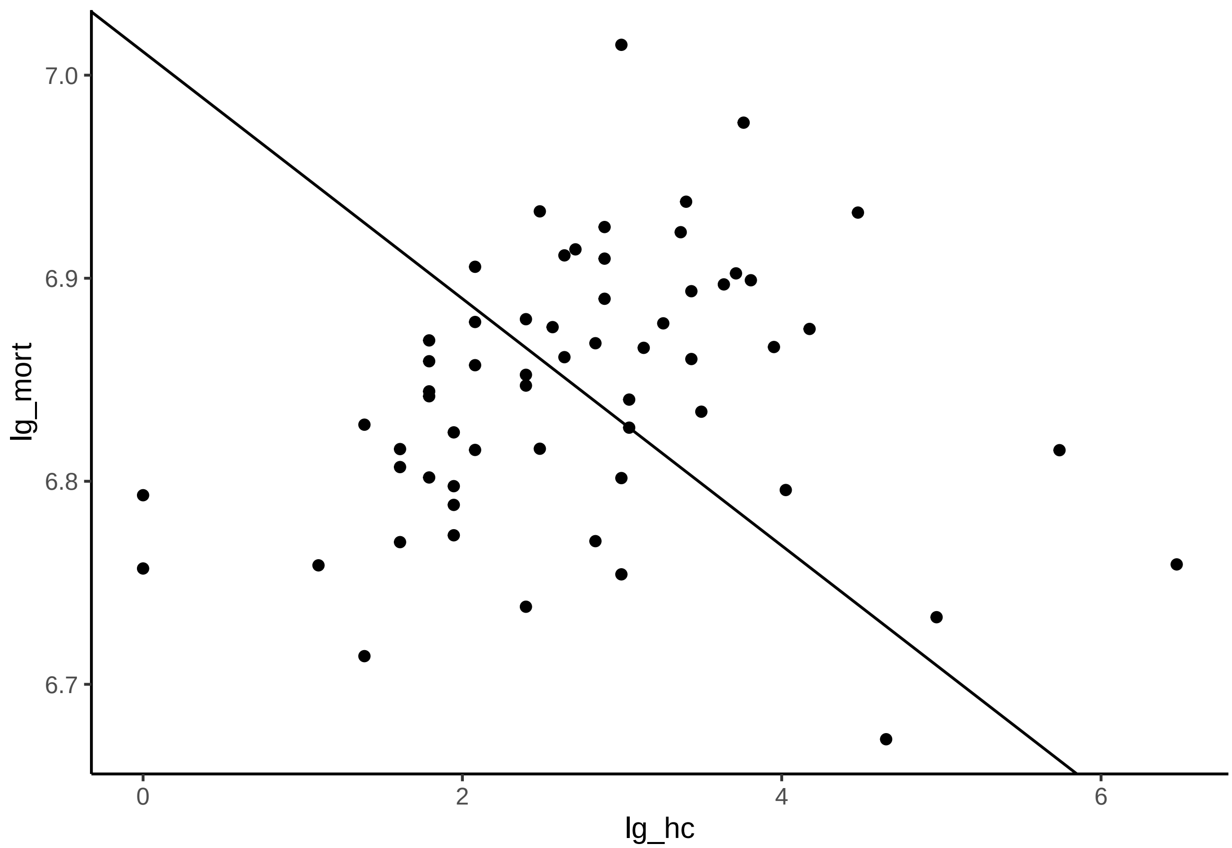
```
# using log() instead
pollution$lg_so2 <- log(pollution$so2)
pollution$lg_hc <- log(pollution$hc)
fit_126d <- lm(pollution$lg_mort ~ pollution$lg_nox + pollution$lg_so2 + pollution$lg_hc)
mean <- c(1, mean(pollution$lg_nox), mean(pollution$lg_so2), mean(pollution$lg_hc))
c <- coef(fit_126d)
ggplot(pollution)+
  geom_point(mapping = aes(lg_nox,lg_mort))+
  geom_abline(intercept = c[1] + mean[3]*c[3] + mean[4]*c[4], slope = c[2]) +
  theme_classic()
```




```
ggplot(pollution)+
  geom_point(mapping = aes(lg_so2,lg_mort))+
  geom_abline(intercept = c[1] + mean[2]*c[2] + mean[4]*c[4], slope = c[3]) +
  theme_classic()
```




```
ggplot(pollution)+
  geom_point(mapping = aes(lg_hc,lg_mort))+
  geom_abline(intercept = c[1] + mean[2]*c[2] + mean[3]*c[3], slope = c[4]) +
  theme_classic()
```



(e)

Cross validate: fit the model you chose above to the first half of the data and then predict for the second half. You used all the data to construct the model in (d), so this is not really cross validation, but it gives a sense of how the steps of cross validation can be implemented.

```
fit_126e <- lm(pollution$lg_mort[1:30] ~ pollution$lg_nox[1:30] + pollution$lg_so2[1:30] + pollution$lg
pred <- data.frame(x1 = pollution$lg_nox[31:60], x2 = pollution$lg_so2[31:60], x3 = pollution$lg_hc[31:
predict(fit_126e, pred)
```

```
##        1        2        3        4        5        6        7        8
## 6.869872 6.873543 6.869462 6.845278 6.895535 6.876079 6.879118 6.816638
##        9       10       11       12       13       14       15       16
## 6.859562 6.847021 6.847985 6.894830 6.892698 6.868271 6.831289 6.802482
##       17       18       19       20       21       22       23       24
## 6.848491 6.849858 6.879542 6.828877 6.802482 6.837923 6.813465 6.831815
##       25       26       27       28       29       30
## 6.787173 6.857074 6.813840 6.859400 6.858474 6.895996
```

12.7

Cross validation comparison of models with different transformations of outcomes: when we compare models with transformed continuous outcomes, we must take into account how the nonlinear transformation warps the continuous outcomes. Follow the procedure used to compare models for the mesquite bushes example on page 202.

(a)

Compare models for earnings and for log(earnings) given height and sex as shown in page 84 and 192. Use `earnk` and `log(earnk)` as outcomes.

```
data(earnings)
fit_127a <- stan_glm(earnk ~ height + male, data = earnings, refresh = 0)
fit_127b <- stan_glm(log(earnk[earnk!=0]) ~ height[earnk!=0] + male[earnk!=0], data = earnings, refresh
#adding if/else due to the missing values
loo(fit_127a)
```

```
## Warning: Found 1 observation(s) with a pareto_k > 0.7. We recommend calling 'loo' again with argumen
```

```
##
## Computed from 4000 by 1816 log-likelihood matrix
##
##          Estimate    SE
## elpd_loo  -8153.8 172.5
## p_loo        29.8  21.9
## looic     16307.6 344.9
## ------
## Monte Carlo SE of elpd_loo is NA.
##
## Pareto k diagnostic values:
##                          Count Pct.    Min. n_eff
## (-Inf, 0.5]   (good)     1815  99.9%   657
##  (0.5, 0.7]   (ok)          0   0.0%   <NA>
##    (0.7, 1]   (bad)         0   0.0%   <NA>
##    (1, Inf)   (very bad)    1   0.1%   7
## See help('pareto-k-diagnostic') for details.
```

```
loo(fit_127b)
```

```
##
## Computed from 4000 by 1629 log-likelihood matrix
##
##          Estimate   SE
## elpd_loo  -2083.5 38.7
## p_loo         4.8  0.4
## looic      4166.9 77.5
## ------
## Monte Carlo SE of elpd_loo is 0.0.
##
## All Pareto k estimates are good (k < 0.5).
## See help('pareto-k-diagnostic') for details.
```

(b)

Compare models from other exercises in this chapter.

```
# all other models are using lm which can not apply leave-one-outvalidation
```

12.8

Log-log transformations: Suppose that, for a certain population of animals, we can predict log weight from log height as follows:

- An animal that is 50 centimeters tall is predicted to weigh 10 kg.
- Every increase of 1% in height corresponds to a predicted increase of 2% in weight.
- The weights of approximately 95% of the animals fall within a factor of 1.1 of predicted values.

(a)

Give the equation of the regression line and the residual standard deviation of the regression.

log(weight)=-5.521461 + 2*log(height) residual standard deviation: 0.05

(b)

Suppose the standard deviation of log weights is 20% in this population. What, then, is the R^2 of the regression model described here?

0.8, 80% of animals is fitted instead of 95%, approximately.

12.9

Linear and logarithmic transformations: For a study of congressional elections, you would like a measure of the relative amount of money raised by each of the two major-party candidates in each district. Suppose that you know the amount of money raised by each candidate; label these dollar values D_i and R_i. You would like to combine these into a single variable that can be included as an input variable into a model predicting vote share for the Democrats. Discuss the advantages and disadvantages of the following measures:

(a)

The simple difference, $D_i - R_i$

easy to interpret, both for coefficient and interception, however, the difference in dollar amount might not fully convey the real difference considering the totals.

(b)

The ratio, D_i/R_i

easy to interpret, but hard to interpret the interception since it can not be zero

(c)

The difference on the logarithmic scale, $\log D_i - \log R_i$

better than a)

(d)

The relative proportion, $D_i/(D_i + R_i)$.

better than b), but has the same weakness(intercept)

12.11

Elasticity: An economist runs a regression examining the relations between the average price of cigarettes, P, and the quantity purchased, Q, across a large sample of counties in the United States, assuming the functional form, $\log Q = \alpha + \beta \log P$. Suppose the estimate for β is 0.3. Interpret this coefficient.

1% increase in average price is averagely associated with an 0.3 increase in the quantity purchased of cigarettes.

12.13

Building regression models: Return to the teaching evaluations data from Exercise 10.6. Fit regression models predicting evaluations given many of the inputs in the dataset. Consider interactions, combinations of predictors, and transformations, as appropriate. Consider several models, discuss in detail the final model that you choose, and also explain why you chose it rather than the others you had considered.

```
data
```

```
## function (..., list = character(), package = NULL, lib.loc = NULL,
##     verbose = getOption("verbose"), envir = .GlobalEnv, overwrite = TRUE)
## {
##     fileExt <- function(x) {
##         db <- grepl("\\.[^.]+\\.(gz|bz2|xz)$", x)
##         ans <- sub(".*\\.", "", x)
##         ans[db] <- sub(".*\\.([^.]+\\.)(gz|bz2|xz)$", "\\1\\2",
##             x[db])
##         ans
##     }
##     my_read_table <- function(...) {
##         lcc <- Sys.getlocale("LC_COLLATE")
##         on.exit(Sys.setlocale("LC_COLLATE", lcc))
##         Sys.setlocale("LC_COLLATE", "C")
##         read.table(...)
##     }
##     stopifnot(is.character(list))
##     names <- c(as.character(substitute(list(...))[-1L]), list)
##     if (!is.null(package)) {
##         if (!is.character(package))
##             stop("'package' must be a character vector or NULL")
##     }
##     paths <- find.package(package, lib.loc, verbose = verbose)
##     if (is.null(lib.loc))
##         paths <- c(path.package(package, TRUE), if (!length(package)) getwd(),
##             paths)
##     paths <- unique(normalizePath(paths[file.exists(paths)]))
##     paths <- paths[dir.exists(file.path(paths, "data"))]
##     dataExts <- tools:::.make_file_exts("data")
```

```
##     if (length(names) == 0L) {
##         db <- matrix(character(), nrow = 0L, ncol = 4L)
##         for (path in paths) {
##             entries <- NULL
##             packageName <- if (file_test("-f", file.path(path,
##                 "DESCRIPTION")))
##                 basename(path)
##             else "."
##             if (file_test("-f", INDEX <- file.path(path, "Meta",
##                 "data.rds"))) {
##                 entries <- readRDS(INDEX)
##             }
##             else {
##                 dataDir <- file.path(path, "data")
##                 entries <- tools::list_files_with_type(dataDir,
##                   "data")
##                 if (length(entries)) {
##                   entries <- unique(tools::file_path_sans_ext(basename(entries)))
##                   entries <- cbind(entries, "")
##                 }
##             }
##             if (NROW(entries)) {
##                 if (is.matrix(entries) && ncol(entries) == 2L)
##                   db <- rbind(db, cbind(packageName, dirname(path),
##                     entries))
##                 else warning(gettextf("data index for package %s is invalid and will be ignored",
##                   sQuote(packageName)), domain = NA, call. = FALSE)
##             }
##         }
##         colnames(db) <- c("Package", "LibPath", "Item", "Title")
##         footer <- if (missing(package))
##             paste0("Use ", sQuote(paste("data(package =", ".packages(all.available = TRUE))")),
##                 "\n", "to list the data sets in all *available* packages.")
##         else NULL
##         y <- list(title = "Data sets", header = NULL, results = db,
##             footer = footer)
##         class(y) <- "packageIQR"
##         return(y)
##     }
##     paths <- file.path(paths, "data")
##     for (name in names) {
##         found <- FALSE
##         for (p in paths) {
##             tmp_env <- if (overwrite)
##                 envir
##             else new.env()
##             if (file_test("-f", file.path(p, "Rdata.rds"))) {
##                 rds <- readRDS(file.path(p, "Rdata.rds"))
##                 if (name %in% names(rds)) {
##                   found <- TRUE
##                   if (verbose)
##                     message(sprintf("name=%s:\t found in Rdata.rds",
##                       name), domain = NA)
##                   thispkg <- sub(".*/([^/]*)/data$", "\\1", p)
```

```
##                   thispkg <- sub("_.*$", "", thispkg)
##                   thispkg <- paste0("package:", thispkg)
##                   objs <- rds[[name]]
##                   lazyLoad(file.path(p, "Rdata"), envir = tmp_env,
##                     filter = function(x) x %in% objs)
##                   break
##                 }
##                 else if (verbose)
##                   message(sprintf("name=%s:\t NOT found in names() of Rdata.rds, i.e.,\n\t%s\n",
##                     name, paste(names(rds), collapse = ",")),
##                     domain = NA)
##             }
##             if (file_test("-f", file.path(p, "Rdata.zip"))) {
##                 warning("zipped data found for package ", sQuote(basename(dirname(p))),
##                   ".\nThat is defunct, so please re-install the package.",
##                   domain = NA)
##                 if (file_test("-f", fp <- file.path(p, "filelist")))
##                   files <- file.path(p, scan(fp, what = "", quiet = TRUE))
##                 else {
##                   warning(gettextf("file 'filelist' is missing for directory %s",
##                     sQuote(p)), domain = NA)
##                   next
##                 }
##             }
##             else {
##                 files <- list.files(p, full.names = TRUE)
##             }
##             files <- files[grep(name, files, fixed = TRUE)]
##             if (length(files) > 1L) {
##                 o <- match(fileExt(files), dataExts, nomatch = 100L)
##                 paths0 <- dirname(files)
##                 paths0 <- factor(paths0, levels = unique(paths0))
##                 files <- files[order(paths0, o)]
##             }
##             if (length(files)) {
##                 for (file in files) {
##                   if (verbose)
##                     message("name=", name, ":\t file= ...", .Platform$file.sep,
##                       basename(file), "::\t", appendLF = FALSE,
##                       domain = NA)
##                   ext <- fileExt(file)
##                   if (basename(file) != paste0(name, ".", ext))
##                     found <- FALSE
##                   else {
##                     found <- TRUE
##                     zfile <- file
##                     zipname <- file.path(dirname(file), "Rdata.zip")
##                     if (file.exists(zipname)) {
##                       Rdatadir <- tempfile("Rdata")
##                       dir.create(Rdatadir, showWarnings = FALSE)
##                       topic <- basename(file)
##                       rc <- .External(C_unzip, zipname, topic,
##                         Rdatadir, FALSE, TRUE, FALSE, FALSE)
##                       if (rc == 0L)
```

```
##                          zfile <- file.path(Rdatadir, topic)
##                      }
##                      if (zfile != file)
##                        on.exit(unlink(zfile))
##                      switch(ext, R = , r = {
##                        library("utils")
##                        sys.source(zfile, chdir = TRUE, envir = tmp_env)
##                      }, RData = , rdata = , rda = load(zfile,
##                        envir = tmp_env), TXT = , txt = , tab = ,
##                        tab.gz = , tab.bz2 = , tab.xz = , txt.gz = ,
##                        txt.bz2 = , txt.xz = assign(name, my_read_table(zfile,
##                          header = TRUE, as.is = FALSE), envir = tmp_env),
##                        CSV = , csv = , csv.gz = , csv.bz2 = ,
##                        csv.xz = assign(name, my_read_table(zfile,
##                          header = TRUE, sep = ";", as.is = FALSE),
##                          envir = tmp_env), found <- FALSE)
##                    }
##                    if (found)
##                      break
##                  }
##                  if (verbose)
##                    message(if (!found)
##                      "*NOT* ", "found", domain = NA)
##              }
##              if (found)
##                  break
##          }
##          if (!found) {
##              warning(gettextf("data set %s not found", sQuote(name)),
##                  domain = NA)
##          }
##          else if (!overwrite) {
##              for (o in ls(envir = tmp_env, all.names = TRUE)) {
##                  if (exists(o, envir = envir, inherits = FALSE))
##                    warning(gettextf("an object named %s already exists and will not be overwritten",
##                      sQuote(o)))
##                  else assign(o, get(o, envir = tmp_env, inherits = FALSE),
##                    envir = envir)
##              }
##              rm(tmp_env)
##          }
##      }
##      invisible(names)
## }
## <bytecode: 0x0000021bfd780928>
## <environment: namespace:utils>
```

```
head(beauty)
```

```
##   eval     beauty female age minority nonenglish lower course_id
## 1  4.3  0.2015666      1  36        1          0     0         3
## 2  4.5 -0.8260813      0  59        0          0     0         0
## 3  3.7 -0.6603327      0  51        0          0     0         4
## 4  4.3 -0.7663125      1  40        0          0     0         2
```

```
## 5  4.4  1.4214450      1  31         0           0     0          0
## 6  4.2  0.5002196      0  62         0           0     0          0
```

```r
fit_1213a <- lm(eval ~ beauty + female + age + female*age,data=beauty)
summary(fit_1213a)
```

```
##
## Call:
## lm(formula = eval ~ beauty + female + age + female * age, data = beauty)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.8890 -0.3582  0.0527  0.3734  1.0346
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.014048   0.173347  23.156  < 2e-16 ***
## beauty       0.142322   0.033133   4.295 2.13e-05 ***
## female       0.348344   0.267425   1.303   0.1934
## age          0.001568   0.003380   0.464   0.6429
## female:age  -0.011887   0.005574  -2.133   0.0335 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5353 on 458 degrees of freedom
## Multiple R-squared:  0.07725,    Adjusted R-squared:  0.06919
## F-statistic: 9.586 on 4 and 458 DF,  p-value: 1.887e-07
```

```r
fit_1213b <- lm(log(eval) ~ beauty + female + age + female*age,data=beauty )
summary(fit_1213b)
```

```
##
## Call:
## lm(formula = log(eval) ~ beauty + female + age + female * age,
##     data = beauty)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.62921 -0.08533  0.02153  0.09891  0.24727
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.3873490  0.0460964  30.097  < 2e-16 ***
## beauty       0.0367841  0.0088108   4.175 3.57e-05 ***
## female       0.0850793  0.0711135   1.196   0.2322
## age          0.0002404  0.0008989   0.267   0.7892
## female:age  -0.0029515  0.0014823  -1.991   0.0471 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1424 on 458 degrees of freedom
## Multiple R-squared:  0.07288,    Adjusted R-squared:  0.06479
## F-statistic: 9.001 on 4 and 458 DF,  p-value: 5.265e-07
```

```
# choosing log(eval) instead.There is no clear difference between the two except for the residuals. Usi
```

12.14

Prediction from a fitted regression: Consider one of the fitted models for mesquite leaves, for example `fit_4`, in Section 12.6. Suppose you wish to use this model to make inferences about the average mesquite yield in a new set of trees whose predictors are in data frame called new_trees. Give R code to obtain an estimate and standard error for this population average. You do not need to make the prediction; just give the code.

```
data(mesquite)
head(mesquite)
```

```
##   obs group diam1 diam2 total_height canopy_height density weight
## 1   1   MCD   1.8  1.15         1.30          1.00       1  401.3
## 2   2   MCD   1.7  1.35         1.35          1.33       1  513.7
## 3   3   MCD   2.8  2.55         2.16          0.60       1 1179.2
## 4   4   MCD   1.3  0.85         1.80          1.20       1  308.0
## 5   5   MCD   3.3  1.90         1.55          1.05       1  855.2
## 6   6   MCD   1.4  1.40         1.20          1.00       1  268.7
```

```
fit_1214 <- stan_glm(formula = weight ~ diam1 + diam2 + canopy_height +
total_height + group + density, data=mesquite, refresh = 0)
```