Model Exercises

Ai Yukino

HW link

Import packages

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(modelr)
```

Exercise 1

Familiarize yourself with the `heights` data set provided with the `modelr` package.

Solution

```
data(heights)
heights
```

```
## # A tibble: 7,006 x 8
##    income height weight   age marital  sex    education  afqt
##     <int>  <dbl>  <int> <int> <fct>    <fct>      <int> <dbl>
##  1  19000     60    155    53 married  female        13  6.84
##  2  35000     70    156    51 married  female        10 49.4
##  3 105000     65    195    52 married  male          16 99.4
##  4  40000     63    197    54 married  female        14 44.0
##  5  75000     66    190    49 married  male          14 59.7
##  6 102000     68    200    49 divorced female        18 98.8
##  7      0     74    225    48 married  male          16 82.3
##  8  70000     64    160    54 divorced female        12 50.3
##  9  60000     69    162    55 divorced male          12 89.7
## 10 150000     69    194    54 divorced male          13 96.0
## # ... with 6,996 more rows
```

```
# ?heights
```

Exercise 2

Create a list of formulas for modeling income with:

- `height`
- `height` · `weight`
- linear combination of all variables

Solution

```
concat_col <- paste(colnames(heights)[-1], collapse=" + ")

formulas <- paste0("income ~ ", c("height", "height * weight", concat_col))

formulas
```

```
## [1] "income ~ height"
## [2] "income ~ height * weight"
## [3] "income ~ height + weight + age + marital + sex + education + afqt"
```

Exercise 3

From the data, remove rows containing NA's. Fit the linear model with the formulas from exercise 2.

Solution

```
heights <- heights %>%
  drop_na()
```

```
model_height <- lm(formula = formulas[1], data = heights)
model_height_times_weight <- lm(formula = formulas[2], data = heights)
model_all <- lm(formula = formulas[3], data = heights)
```

```
summary(model_height)
```

```
##
## Call:
## lm(formula = formulas[1], data = heights)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -92970 -31753 -11225  14620 320574
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -161639.1    11215.0  -14.41   <2e-16 ***
## height         3031.1      166.8   18.18   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 55500 on 6643 degrees of freedom
## Multiple R-squared:  0.04737,    Adjusted R-squared:  0.04723
## F-statistic: 330.3 on 1 and 6643 DF,  p-value: < 2.2e-16
```

```
summary(model_height_times_weight)
```

```
##
## Call:
## lm(formula = formulas[2], data = heights)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -100812  -31099  -11073   14835  322415
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.164e+05  4.652e+04  -4.652 3.36e-06 ***
## height         4.079e+03  7.000e+02   5.827 5.90e-09 ***
## weight         1.393e+02  2.369e+02   0.588    0.557
## height:weight -3.286e+00  3.510e+00  -0.936    0.349
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 55420 on 6641 degrees of freedom
## Multiple R-squared:  0.0507, Adjusted R-squared:  0.05028
## F-statistic: 118.2 on 3 and 6641 DF,  p-value: < 2.2e-16
```

```r
summary(model_all)
```

```
##
## Call:
## lm(formula = formulas[3], data = heights)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -115521  -25139   -5477   14904  326890
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -44409.17   20565.27  -2.159  0.03085 *
## height              293.26     227.77   1.288  0.19796
## weight              -22.62      15.41  -1.468  0.14227
## age                -401.81     270.53  -1.485  0.13753
## maritalmarried    14204.65    1754.67   8.095 6.74e-16 ***
## maritalseparated   3364.49    3055.37   1.101  0.27086
## maritaldivorced    5586.83    1990.67   2.807  0.00502 **
## maritalwidowed    10663.36    4290.03   2.486  0.01296 *
## sexfemale        -24815.77    1744.56 -14.225  < 2e-16 ***
## education          5944.87     289.14  20.561  < 2e-16 ***
## afqt                389.42      26.52  14.685  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 49100 on 6634 degrees of freedom
## Multiple R-squared:  0.2556, Adjusted R-squared:  0.2545
## F-statistic: 227.8 on 10 and 6634 DF,  p-value: < 2.2e-16
```

Exercise 4

For each fit, calculate RMSE.

Solution

```
rmse(model_height, heights)
```

```
## [1] 55496.35
```

```
rmse(model_height_times_weight, heights)
```

```
## [1] 55399.18
```

```
rmse(model_all, heights)
```

```
## [1] 49056.82
```

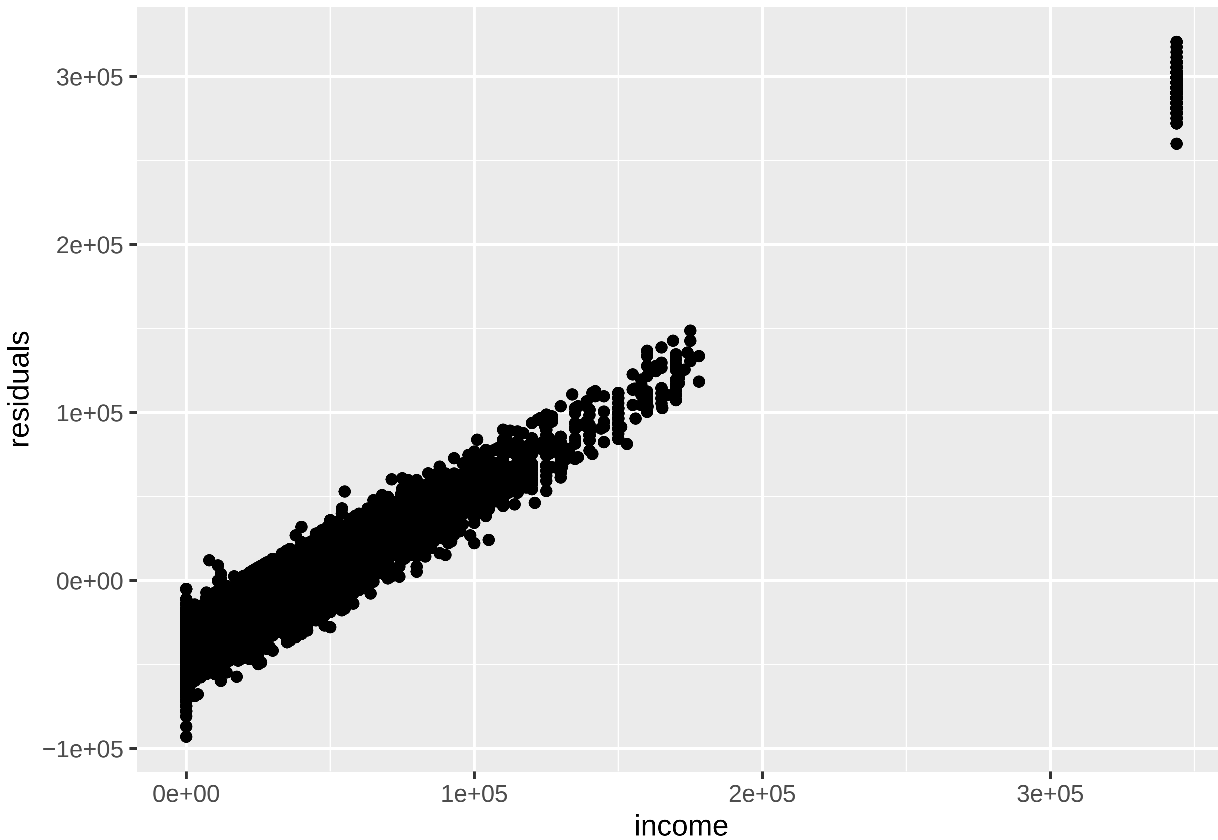
Exercise 5

For each model, add residuals to the data and plot their distribution. (Hint: use `lift_dl()`.)

Solution

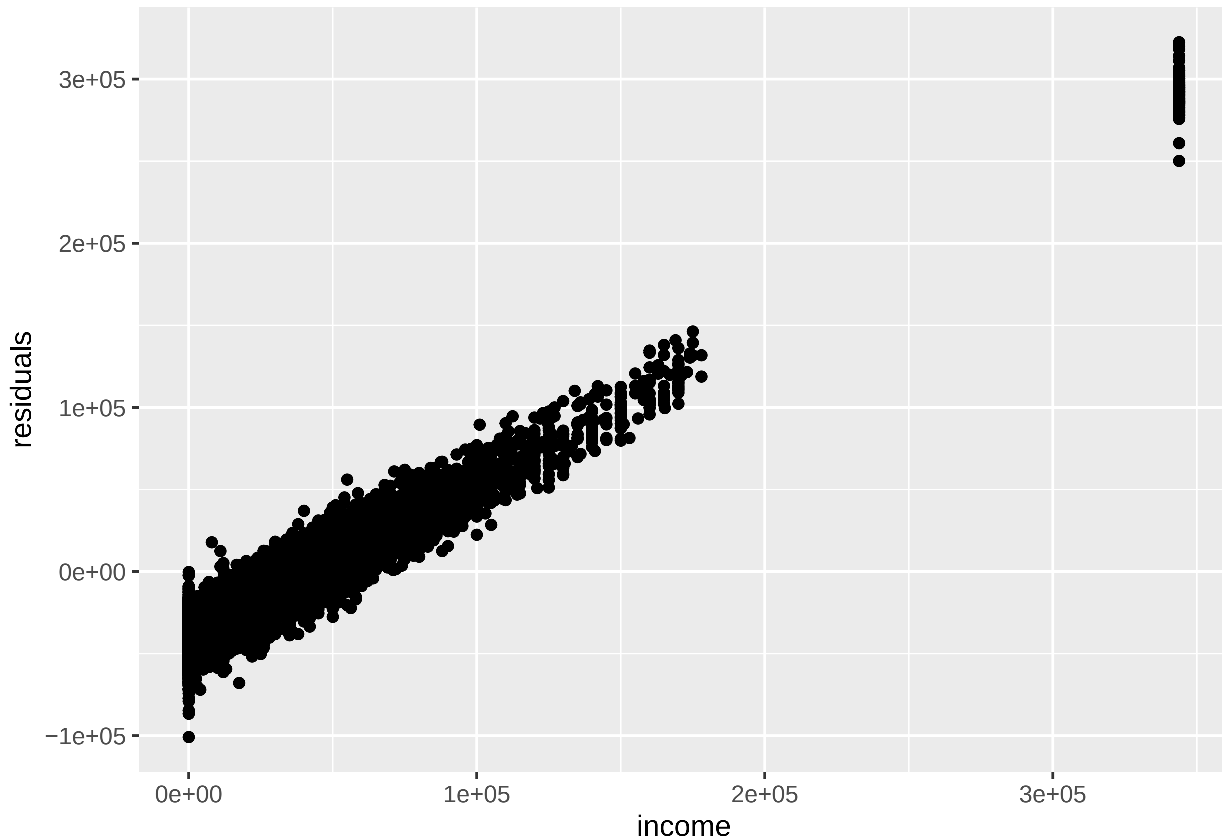
```r
residuals <- resid(model_height)

ggplot(data = heights, aes(x = income, y = residuals)) +
  geom_point()
```



```r
residuals <- resid(model_height_times_weight)

ggplot(data = heights, aes(x = income, y = residuals)) +
  geom_point()
```



```r
residuals <- resid(model_all)

ggplot(data = heights, aes(x = income, y = residuals)) +
  geom_point()
```