Regression

Isabelle Villegas

2022-09-26

Linear Regression

How it works

In Linear Regression, our data is made up of predictor values, x, and target values y. Where we want to find the relationship between x and y. Added variables such as w and b, determine the slope of the line, w, and the intercept, b.

Strengths

- Easy to implement, used to find the relationship between two variables well, over fitting can be amended

Weaknesses

- prone to underfiting, sensitive to noise, assumes that the variables are independent

Reading in the csv file, with data on income of people from different countries. This data set can be used in different use-cases.

The data set can be found here: https://www.kaggle.com/datasets/mukeshmanral/income-data

```
data <- read.csv("income_threshold.csv")
summary(data)
```

```
##       age          workclass             fnlwgt          education
##  Min.   :17.00   Length:30162       Min.   :  13769   Length:30162
##  1st Qu.:28.00   Class :character   1st Qu.: 117627   Class :character
##  Median :37.00   Mode  :character   Median : 178425   Mode  :character
##  Mean   :38.44                      Mean   : 189794
##  3rd Qu.:47.00                      3rd Qu.: 237628
##  Max.   :90.00                      Max.   :1484705
##  education.num   maritalStatus       occupation        relationship
##  Min.   : 1.00   Length:30162       Length:30162       Length:30162
##  1st Qu.: 9.00   Class :character   Class :character   Class :character
##  Median :10.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   :10.12
##  3rd Qu.:13.00
##  Max.   :16.00
##      race               sex             capitalGain     capitalLoss
##  Length:30162       Length:30162       Min.   :    0   Min.   :   0.00
```

```
##  Class :character   Class :character   1st Qu.:    0   1st Qu.:   0.00
##  Mode  :character   Mode  :character   Median :    0   Median :   0.00
##                                        Mean   : 1092   Mean   :  88.37
##                                        3rd Qu.:    0   3rd Qu.:   0.00
##                                        Max.   :99999   Max.   :4356.00
##   hoursPerWeek   nativeCountry         income
##  Min.   : 1.00   Length:30162       Length:30162
##  1st Qu.:40.00   Class :character   Class :character
##  Median :40.00   Mode  :character   Mode  :character
##  Mean   :40.93
##  3rd Qu.:45.00
##  Max.   :99.00
```

a. Divide into 80/20 train/test

Calculating where in the data set it needs to be split for an 80/20 training and test set and then creating the training set from the first element to the split-th element

```
split <- round(nrow(data)*0.8)
training <- data[1:split, ]
```

Creating the test data set going from the split point + 1 all the way to the end of the data set

```
test <- data[(split+1):nrow(data),]
```

b. Use at least 5 R functions for data exploration, using the training data

```
mean(training$fnlwgt)
```

```
## [1] 189499.5
```

```
max(training$fnlwgt)
```

```
## [1] 1484705
```

```
min(training$fnlwgt)
```

```
## [1] 18827
```

```
median(training$fnlwgt)
```

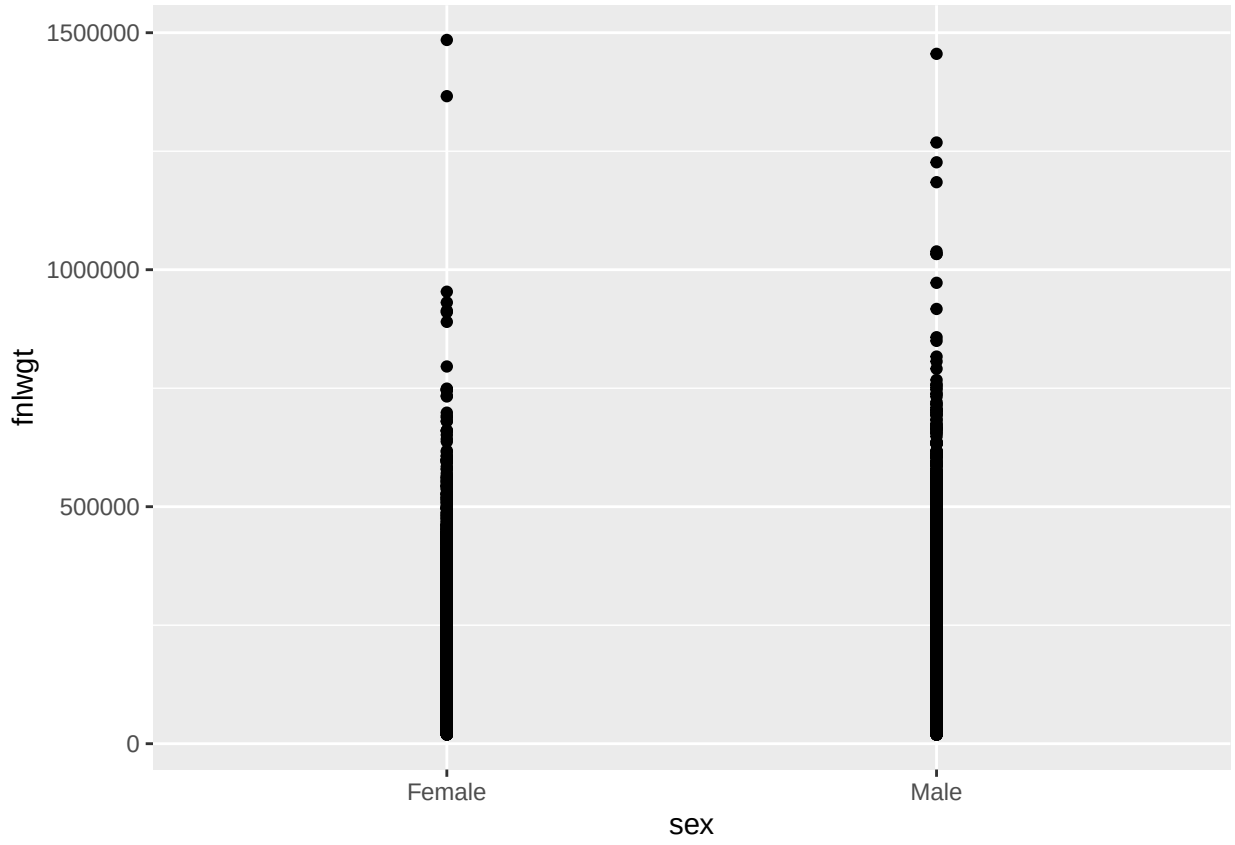
```
## [1] 178138
```

```
sum(training$age)
```

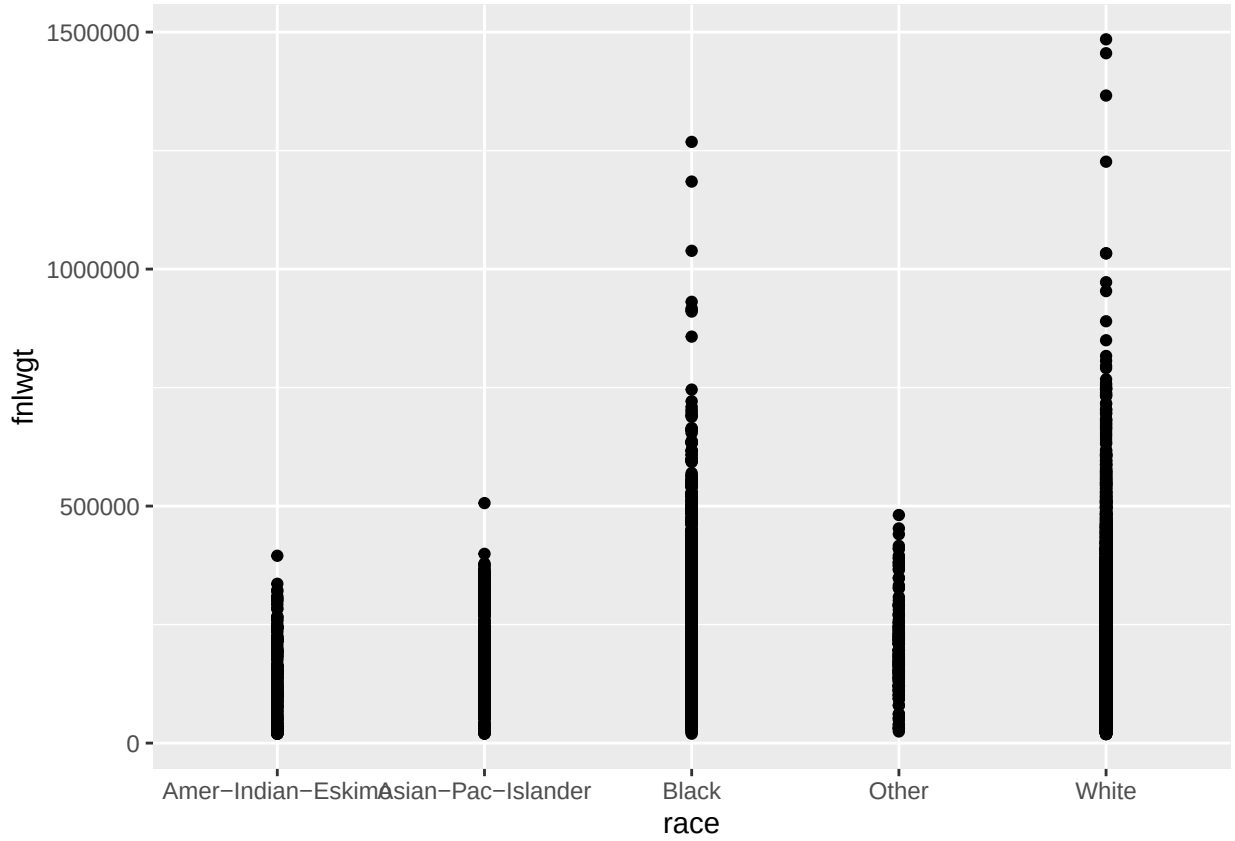
```
## [1] 933110
```

c. Create at least 2 informative graphs, using the training data

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```




```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

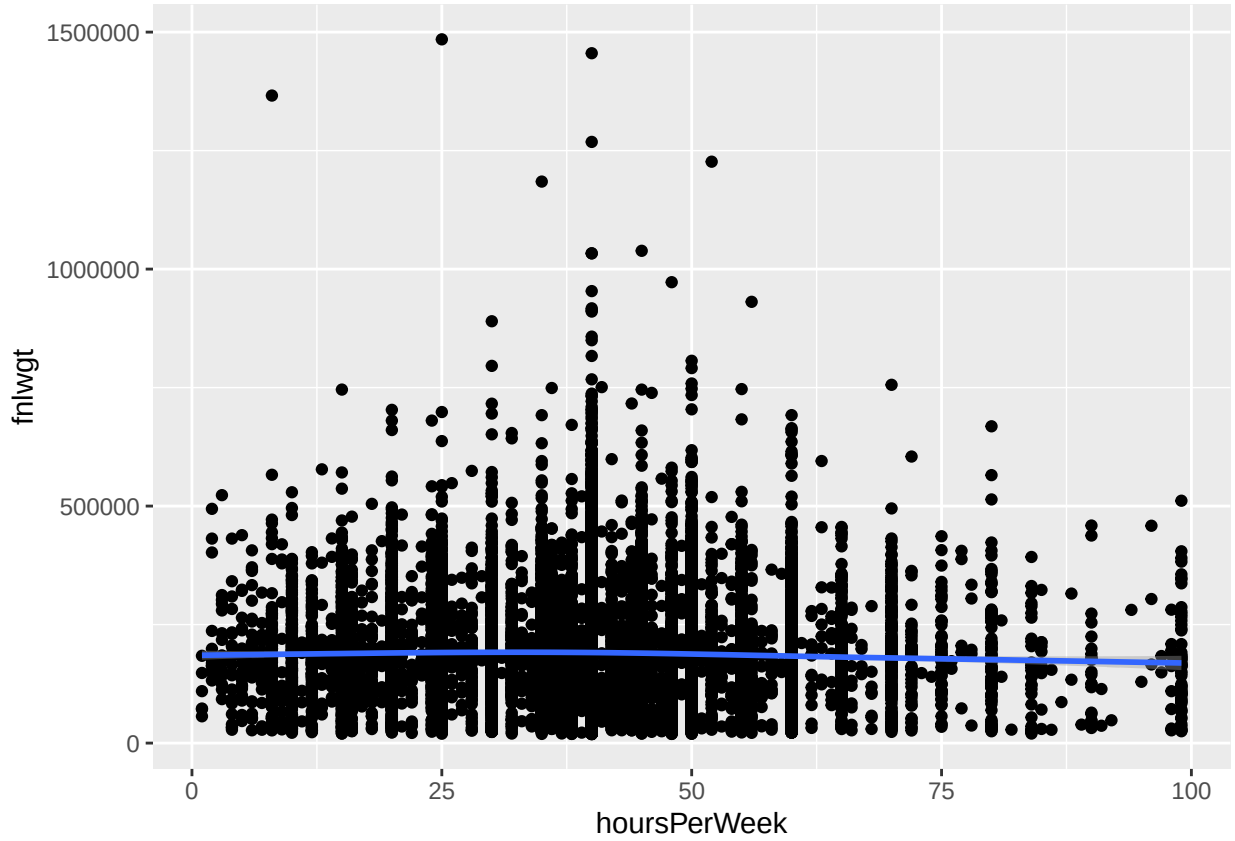


d. Build a simple linear regression model (one predictor) and output the summary. Write athorough explanation of the information in the model summary.

Here we have a plot that plots hours per week worked and income, the linear regression shows that the RSE is 105200 while the R-squared value is .0003934 which is honestly really bad and shows that there is little correlation between hours per week that a person worked and their income. The R-squared value is far from 1 which shows that it has a very low correlation.

```
ggplot(training, aes(x = hoursPerWeek, y = fnlwgt)) +
  geom_point() +
  stat_smooth()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

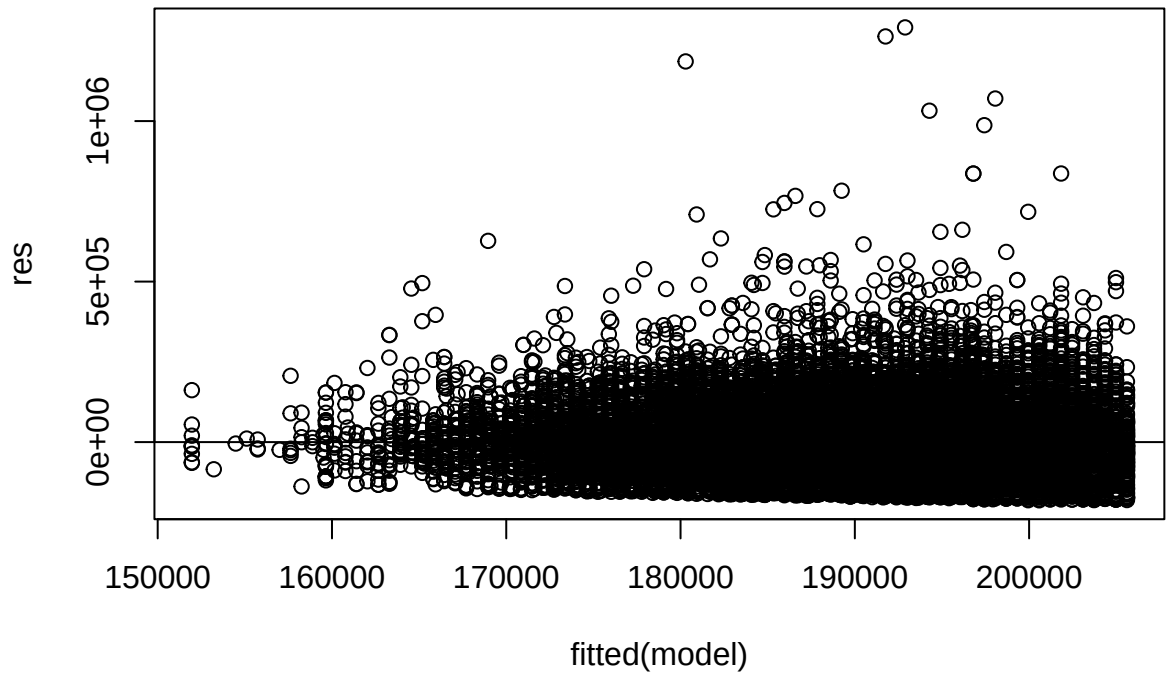



```
lm1 <- lm(fnlwgt~hoursPerWeek, data = training)
summary(lm1)
```

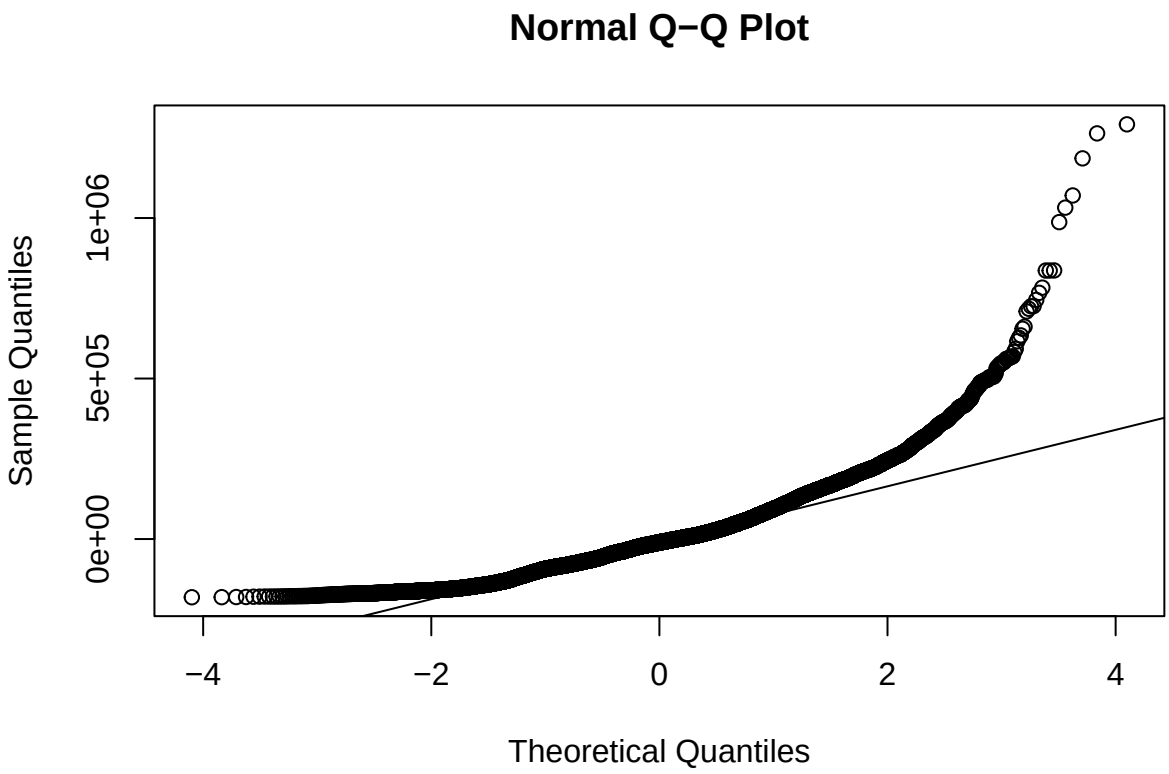
```
##
## Call:
## lm(formula = fnlwgt ~ hoursPerWeek, data = training)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -174129  -71888  -11118   47658 1292408
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  196661.96    2420.85  81.237  < 2e-16 ***
## hoursPerWeek   -174.60      56.66  -3.082  0.00206 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 105200 on 24128 degrees of freedom
## Multiple R-squared:  0.0003934,  Adjusted R-squared:  0.000352
## F-statistic: 9.496 on 1 and 24128 DF,  p-value: 0.002061
```

e. Plot the residuals and write a thorough explanation of what the residual plot tells you.

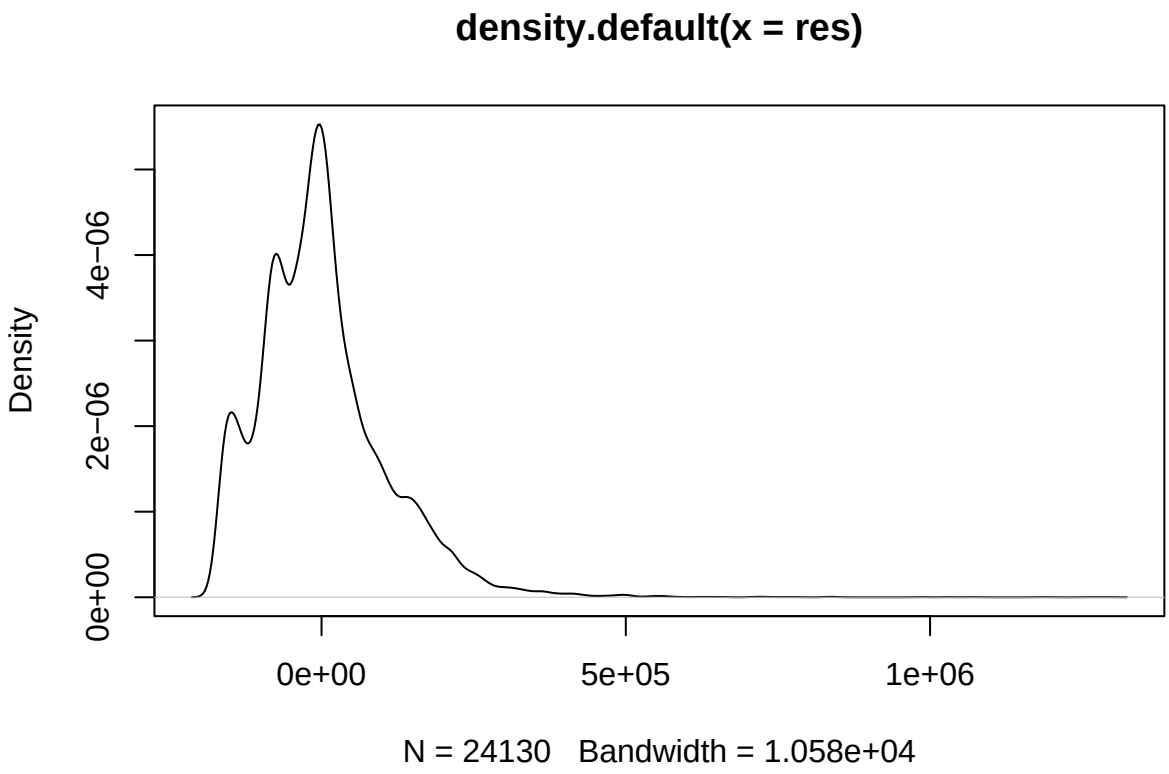
```
#fit a regression model
model <- lm(fnlwgt~age+sex, data=training)

#get list of residuals
res <- resid(model)

#produce residual vs. fitted plot
plot(fitted(model), res)

#add a horizontal line at 0
abline(0,0)
```




```
#create Q-Q plot for residuals
qqnorm(res)

#add a straight diagonal line to the plot
qqline(res)
```



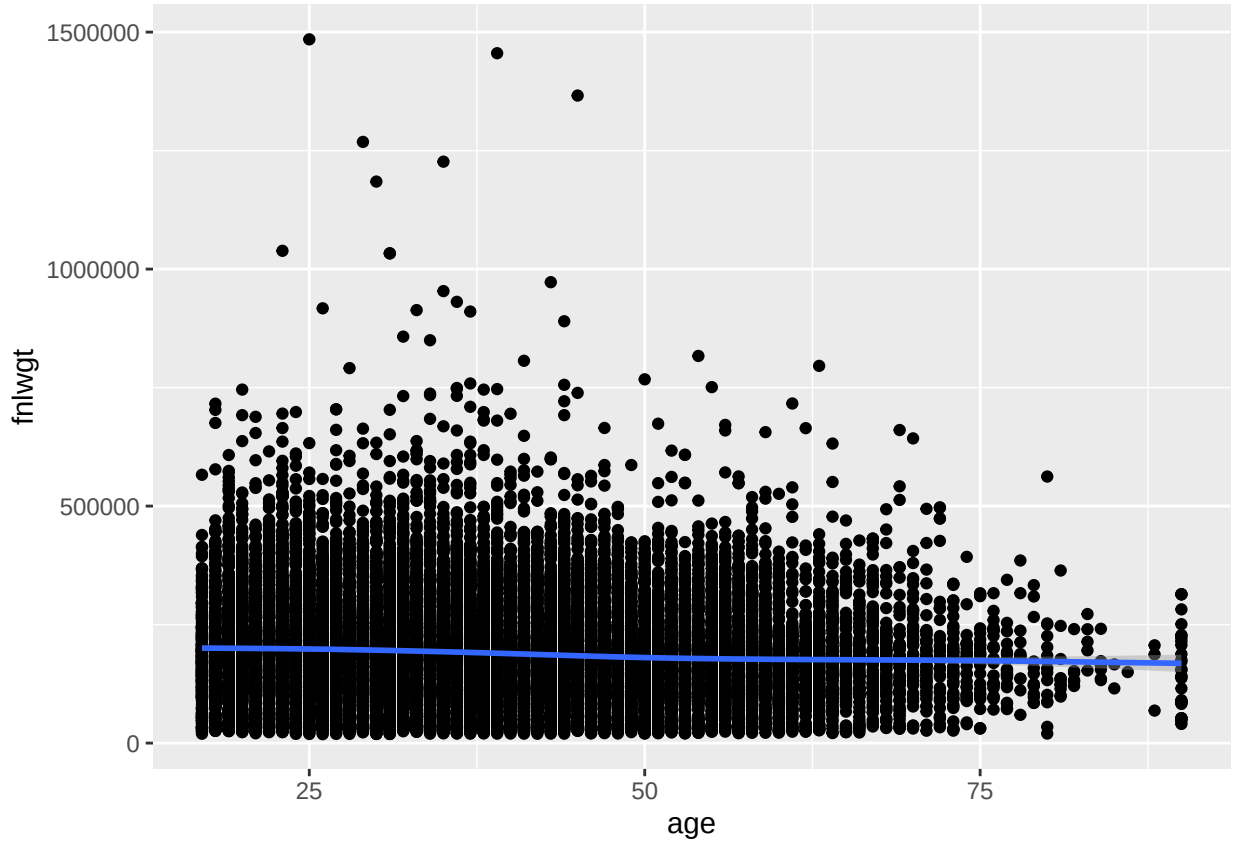

```
#Create density plot of residuals
plot(density(res))
```



f. Build a multiple linear regression model (multiple predictors), output the summary and residual plots.

```
ggplot(training, aes(x = age, y = fnlwgt)) +
  geom_point() +
  stat_smooth()
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



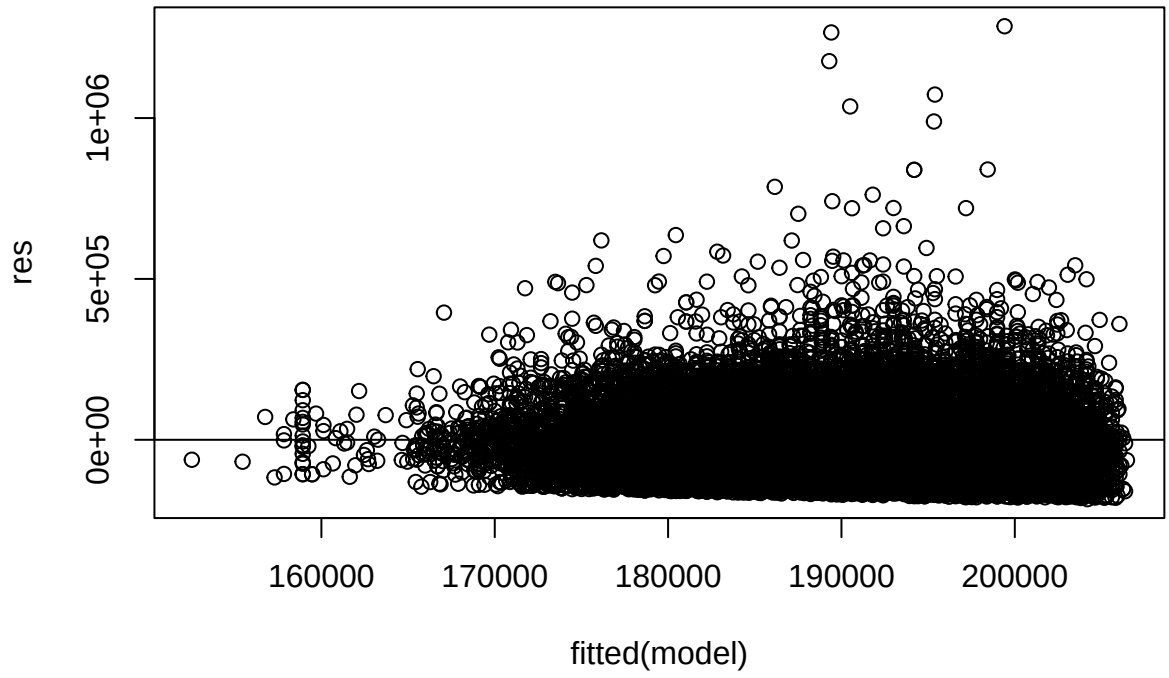

```
lm1 <- lm(fnlwgt~age, data = training)
lm1
```

```
##
## Call:
## lm(formula = fnlwgt ~ age, data = training)
##
## Coefficients:
## (Intercept)          age
##    212997.7       -607.7
```

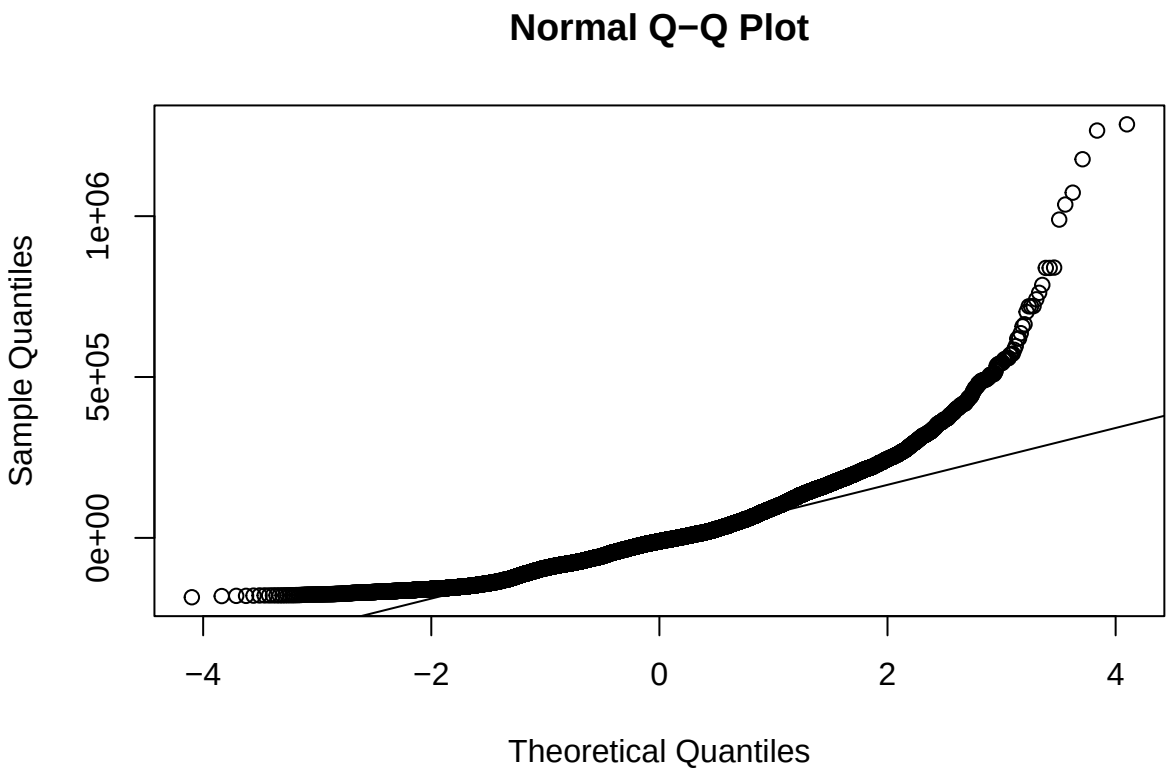
g. Build a third linear regression model using a different combination of predictors, interaction effects, polynomial regression, or any combination to try to improve the results. Output the summary and residual plots.

```
#fit a regression model
model <- lm(fnlwgt~hoursPerWeek+age, data=training)

#get list of residuals
res <- resid(model)

#produce residual vs. fitted plot
plot(fitted(model), res)
```

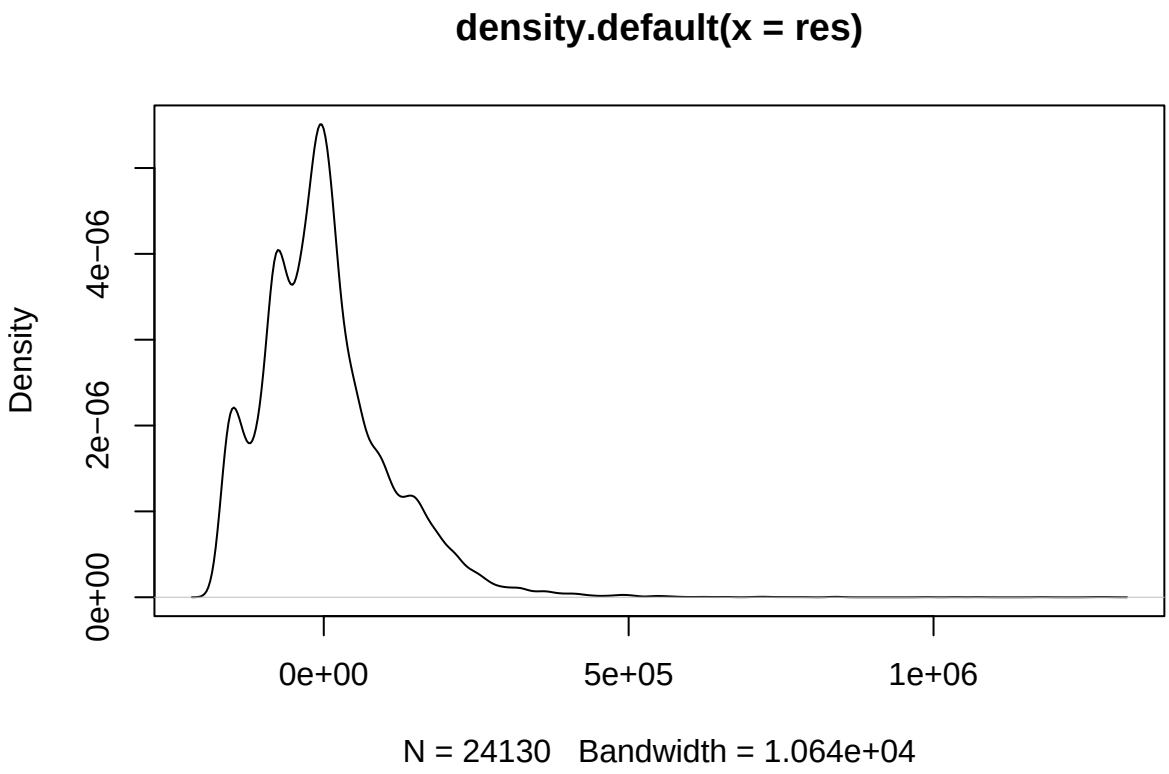
```
#add a horizontal line at 0
abline(0,0)
```




```
#create Q-Q plot for residuals
qqnorm(res)

#add a straight diagonal line to the plot
qqline(res)
```




```
#Create density plot of residuals
plot(density(res))
```


density.default(x = res)
Density
0e+00
2e−06
4e−06
0e+00
5e+05
1e+06
N = 24130 Bandwidth = 1.064e+04