Homework 7

Eric Zou

Honor code:

"I have neither given nor received unauthorized assistance on this assignment." Type your initials here.

I receive help from " " and give help to " ".

Problem 1

Analyze the relationship between TV advertising budget and sales using the provided dataset and visual tools in R.

Steps:

1.Import the dataset from the following URL using R: "https://www.statlearning.com/s/Advertising.csv". (Use read.csv function from `tidyverse` package) 2. Designate 'sales' as your response variable and 'TV' as the predictor variable. 3. Utilize the lm() function in R to establish a linear regression model. 4. From the outcome of the linear regression, deduce the intercept and slope values. 5. Utilizing the ggplot2 package, create a visual representation: Plot a scatter plot for the data points. Overlay the scatter plot with the linear regression line you derived.

```
library(tidyverse)
```

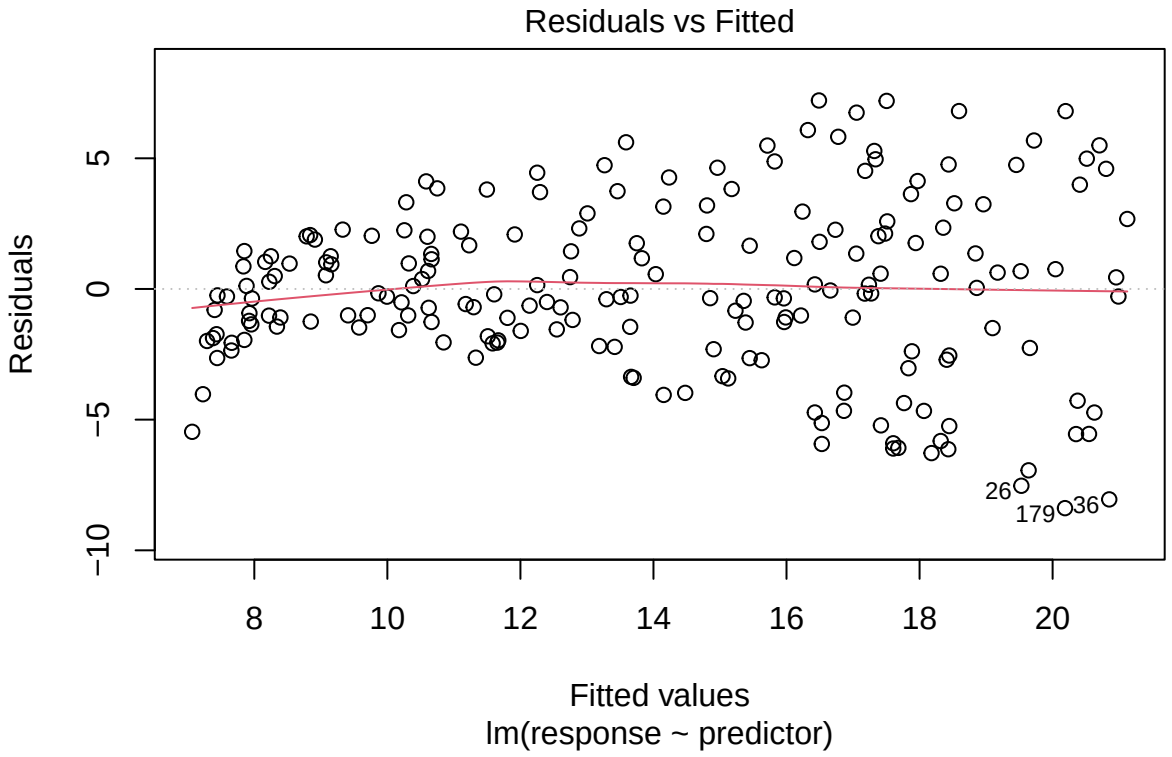
```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

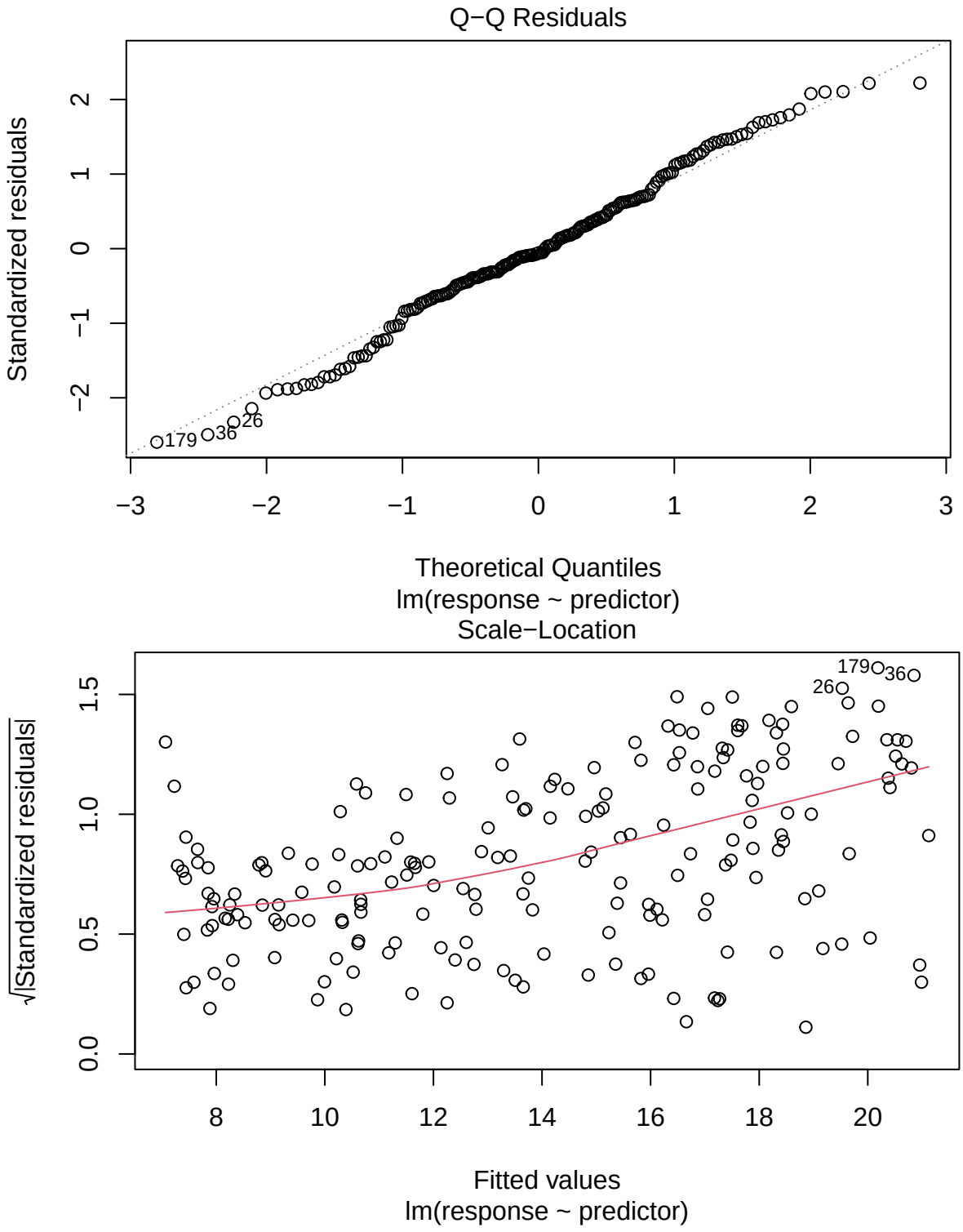
```
df = read.csv("https://www.statlearning.com/s/Advertising.csv")
response = df$sales
predictor = df$TV
fit = lm(response~predictor)
summary(fit)
```

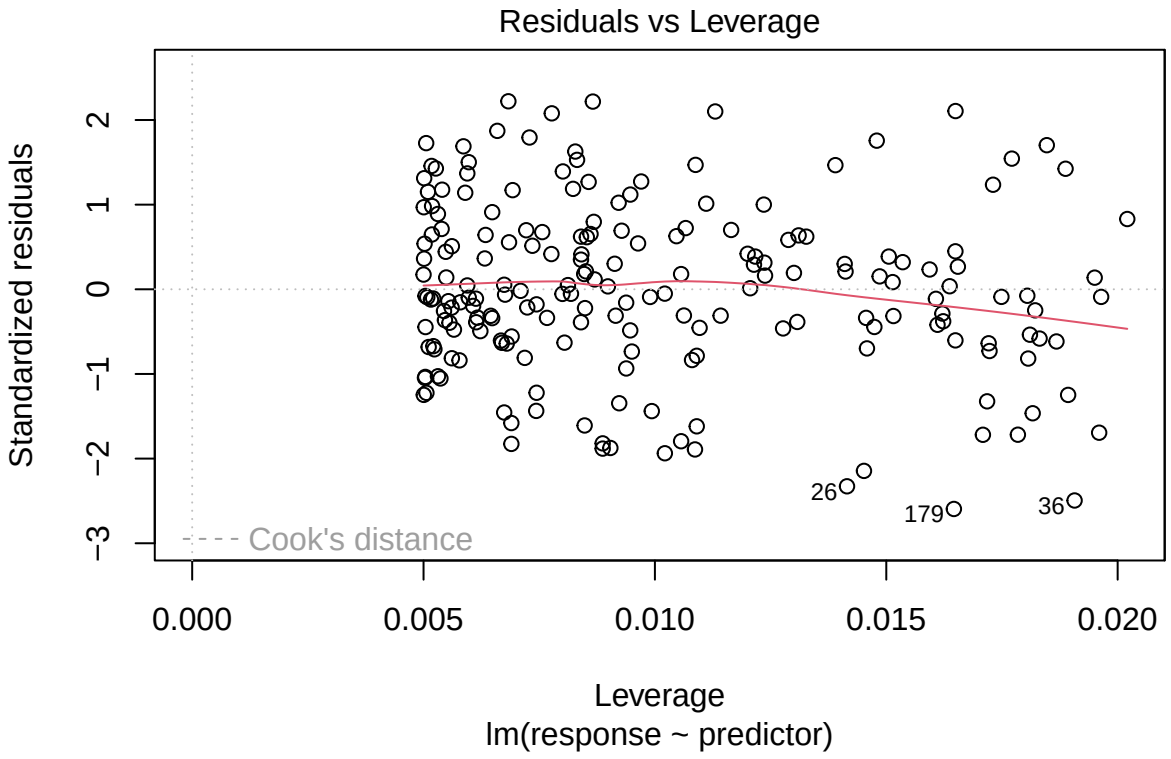
```
##
## Call:
## lm(formula = response ~ predictor)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.3860 -1.9545 -0.1913  2.0671  7.2124
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 7.032594   0.457843   15.36   <2e-16 ***
## predictor   0.047537   0.002691   17.67   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.259 on 198 degrees of freedom
## Multiple R-squared:  0.6119, Adjusted R-squared:  0.6099
## F-statistic: 312.1 on 1 and 198 DF,  p-value: < 2.2e-16
```

```
plot(fit)
```




Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(response ~ predictor)
179
36
26
Scale−Location
√|Standardized residuals|
Fitted values
lm(response ~ predictor)



Problem 2

Here is the summary for `fit = lm(sales~TV)`:

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 7.032594   0.457843   15.36   <2e-16 ***
TV          0.047537   0.002691   17.67   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.259 on 198 degrees of freedom
```

1. Calculate the least square estimate of intercept and slope by your own code (using the partial derivative or projection method).

```
x = df$TV
y = df$sales
xbar = mean(x)
ybar = mean(y)
beta1hat = sum((x-xbar)*(y-ybar))/sum((x-xbar)^2)
beta0hat = ybar - beta1hat*xbar
print(c(beta0hat, beta1hat))
```

```
## [1] 7.03259355 0.04753664
```

2. Show how to calculate the `t value` in the summary table and explain the pvalues (in the last column) Hint: specify the null hypothesis and explain whether there is enough evidence to reject the null hypothesis or make the discovery.

T is calculated by dividing the beta value by it's standard error. P value is the chance that, assuming the null hypothesis is true, testing results will be as extreme as actual results.

3. Calculate the RSS by your own code.

```
rss = sum(response-predictor*beta1hat-beta0hat)^2
rss
```

```
## [1] 1.232595e-26
```

Problem 3

Load the `mpg` data in `tidyverse` package. Fit the simple linear regression with `hwy` as response and `cty` as predictor. Calculate the standard error for each $\hat{y}_i$ and visualize the result by adding the fitted line and confidence band to the scatter plot.

```
df = mpg
x = df$hwy
y = df$cty
xbar = mean(x)
ybar = mean(y)

mod = lm(mpg$hwy~mpg$cty)

beta0hat = 0.8920411 #from model ^^
beta1hat = 1.3374556

dmat = cbind(rep(1,234), x)
P_1x = dmat %*% solve(t(dmat) %*% dmat) %*% t(dmat)
sigmahat = sqrt(sum((y- beta0hat-beta1hat*x)^2/(length(x)-2)))
se_beta1 = sqrt(sigmahat^2/ sum((x-xbar)^2))
c(beta1hat-2*se_beta1, beta1hat+2*se_beta1)
```

```
## [1] 0.9857431 1.6891681
```

```
se_beta0 = sqrt(1/length(x) + xbar^2/sum((x-xbar)^2)) *sigmahat
c(beta0hat-2*se_beta0, beta0hat+2*se_beta0)
```

```
## [1] -7.612915  9.396997
```

```
se_yhat = sigmahat * sqrt(diag(P_1x))

se_yhat #standard error
```

```
##   [1] 1.431020 1.431020 1.690940 1.556478 1.137764 1.137764 1.218097 1.137764
##   [9] 1.080326 1.317142 1.218097 1.080326 1.080326 1.080326 1.080326 1.049549
##  [17] 1.080326 1.047784 1.207417 1.815180 1.207417 1.540947 1.540947 1.137764
##  [25] 1.047784 1.137764 1.080326 1.049549 1.304438 1.961590 1.815180 1.540947
##  [33] 1.218097 1.556478 1.137764 1.431020 1.137764 1.049549 1.049549 1.075176
##  [41] 1.075176 1.049549 1.049549 1.540947 1.075176 1.129603 1.047784 1.047784
##  [49] 1.304438 1.416726 1.540947 1.540947 1.304438 1.304438 2.267003 1.540947
##  [57] 1.815180 1.540947 1.540947 2.267003 1.540947 1.674447 1.416726 1.815180
##  [65] 1.674447 2.267003 1.540947 1.540947 1.674447 2.267003 1.815180 1.674447
##  [73] 1.540947 1.815180 1.540947 1.540947 1.416726 1.540947 1.304438 1.540947
##  [81] 1.304438 1.304438 1.540947 1.540947 1.540947 1.674447 1.674447 1.540947
##  [89] 1.815180 1.540947 1.137764 1.080326 1.137764 1.049549 1.129603 1.075176
##  [97] 1.047784 1.075176 1.207417 1.979430 1.832426 1.832426 1.431020 1.832426
## [105] 2.130810 2.443424 2.443424 1.431020 1.137764 1.218097 1.556478 1.690940
## [113] 1.137764 1.137764 1.317142 1.137764 1.431020 1.317142 1.218097 1.049549
## [121] 1.049549 1.049549 1.075176 1.304438 1.207417 1.540947 2.267003 1.304438
```

```
## [129] 1.416726 1.961590 1.815180 1.416726 1.416726 1.815180 1.540947 1.674447
## [137] 1.416726 1.540947 1.304438 1.304438 1.540947 1.431020 1.218097 1.690940
## [145] 1.832426 1.218097 1.137764 1.137764 1.080326 1.080326 1.540947 1.540947
## [153] 1.207417 1.416726 1.137764 1.137764 1.218097 1.317142 1.080326 1.080326
## [161] 1.049549 1.218097 1.080326 1.137764 1.047784 1.137764 1.137764 1.137764
## [169] 1.137764 1.080326 1.218097 1.080326 1.218097 1.207417 1.207417 1.304438
## [177] 1.540947 1.207417 1.540947 1.431020 1.218097 1.690940 1.690940 1.137764
## [185] 1.137764 1.317142 1.218097 1.431020 1.690940 1.690940 1.137764 1.137764
## [193] 1.218097 1.556478 1.979430 2.285697 2.603475 2.285697 1.815180 1.416726
## [201] 1.207417 1.207417 1.075176 1.540947 1.304438 1.416726 1.207417 1.431020
## [209] 1.137764 1.431020 1.431020 1.049549 3.763540 1.431020 1.137764 1.431020
## [217] 1.431020 1.431020 1.431020 1.047784 1.049549 3.763540 3.260006 1.431020
## [225] 1.137764 1.317142 1.431020 1.431020 1.431020 1.317142 1.431020 1.137764
## [233] 1.137764 1.137764
```

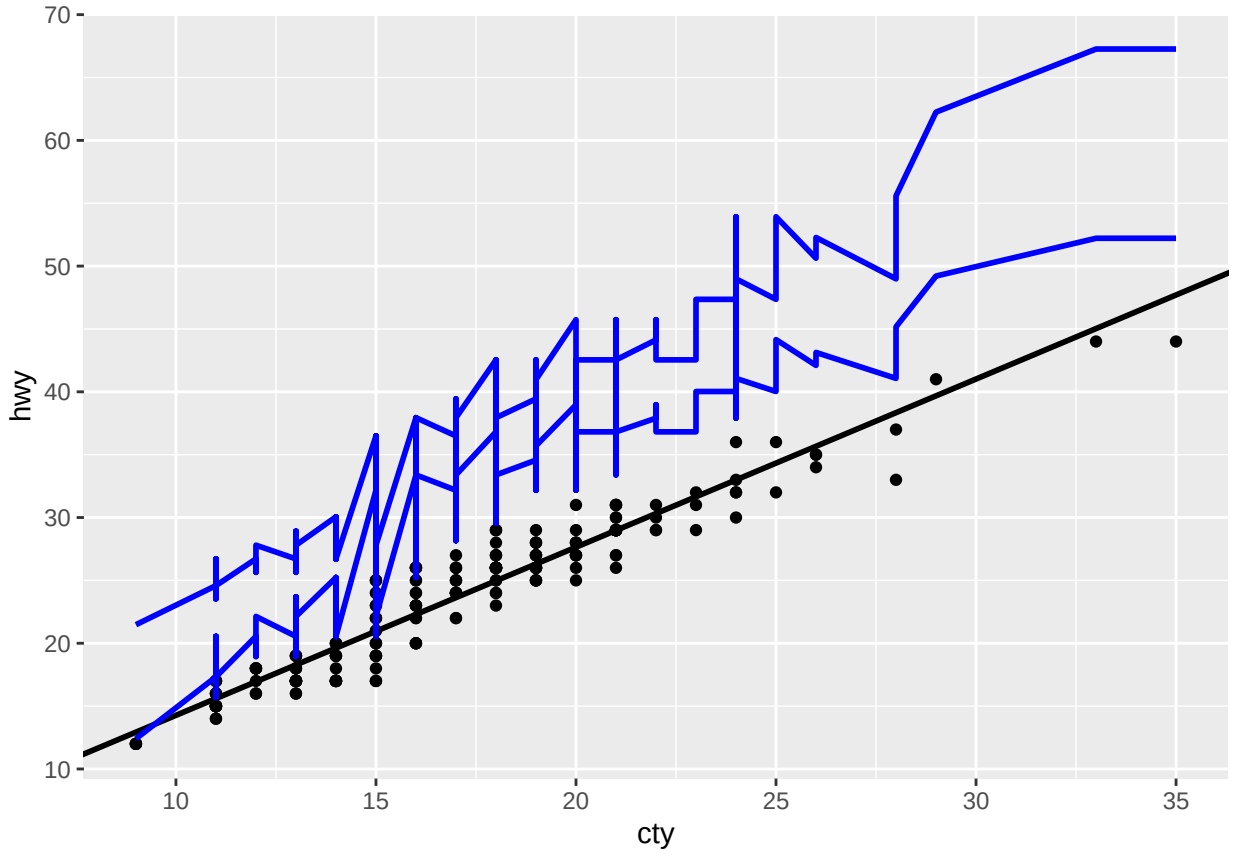
```
y_hat_proj = P_1x %*% y
y_hat = beta0hat + beta1hat * x
print(tibble(y_hat,y_hat_proj))
```

```
## # A tibble: 234 x 2
##    y_hat y_hat_proj[,1]
##    <dbl>          <dbl>
##  1  39.7           20.7
##  2  39.7           20.7
##  3  42.4           22.0
##  4  41.0           21.3
##  5  35.7           18.6
##  6  35.7           18.6
##  7  37.0           19.3
##  8  35.7           18.6
##  9  34.3           17.9
## 10  38.3           20.0
## # i 224 more rows
```

```
p = ggplot(mpg, aes(x=cty, y=hwy)) + geom_point()
p = p + geom_abline(slope=beta1hat, intercept = beta0hat, size=1)
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
p = p + geom_line(aes(y=y_hat-2*se_yhat),color="blue", size = 1)
p = p + geom_line(aes(y=y_hat+2*se_yhat),color="blue", size = 1)
p
```


70
60
50
40
30
20
10
hwy
10
15
20
25
30
35
cty