Name - NetID

Homework 9

Homework Instructions

For questions that require code, please create a code chunk directly below the question and type your code there. Your knitted pdf will show both your code and your output. You are encouraged to knit your file as you work to check that your coding and formatting is done so appropriately.

For written responses or multiple choice questions, please bold your (selected) answer.

Grading Details

All questions will be graded full credit (1 point), half credit (0.5 point) or no credit (0 points).

Full credit responses should have the correct response and appropriate code (if applicable). Half credit responses will have a reasonable attempt (typically no more than one small error or oversight), and no credit responses will be either non-attempts or attempts with significant errors.

Exercise 1

```r
# preamble
gen_data_1 = function(sample_size = 50, seed = 420) {
  set.seed(seed)
  x = runif(n = sample_size, min = 0, max = 3)
  y = 2 + 3 * x + rnorm(n = sample_size)
  data.frame(x = x, y = y)
}

gen_data_2 = function(sample_size = 50, seed = 420) {
  set.seed(seed)
  x = runif(n = sample_size, min = 0, max = 3)
  y = 2 + 3 * x + rf(n = sample_size, df1 = 4, df2 = 30)
  data.frame(x = x, y = y)
}

data_1 = gen_data_1()
data_2 = gen_data_2()
```

```r
# starter
data_1
data_2
```

The above code block has access to two data frames named `data_1` and `data_2`, both with variables variables `y` and `x`. Here, we use `y` as the response.

Fit a simple linear regression to both datasets (no summaries needed for now).

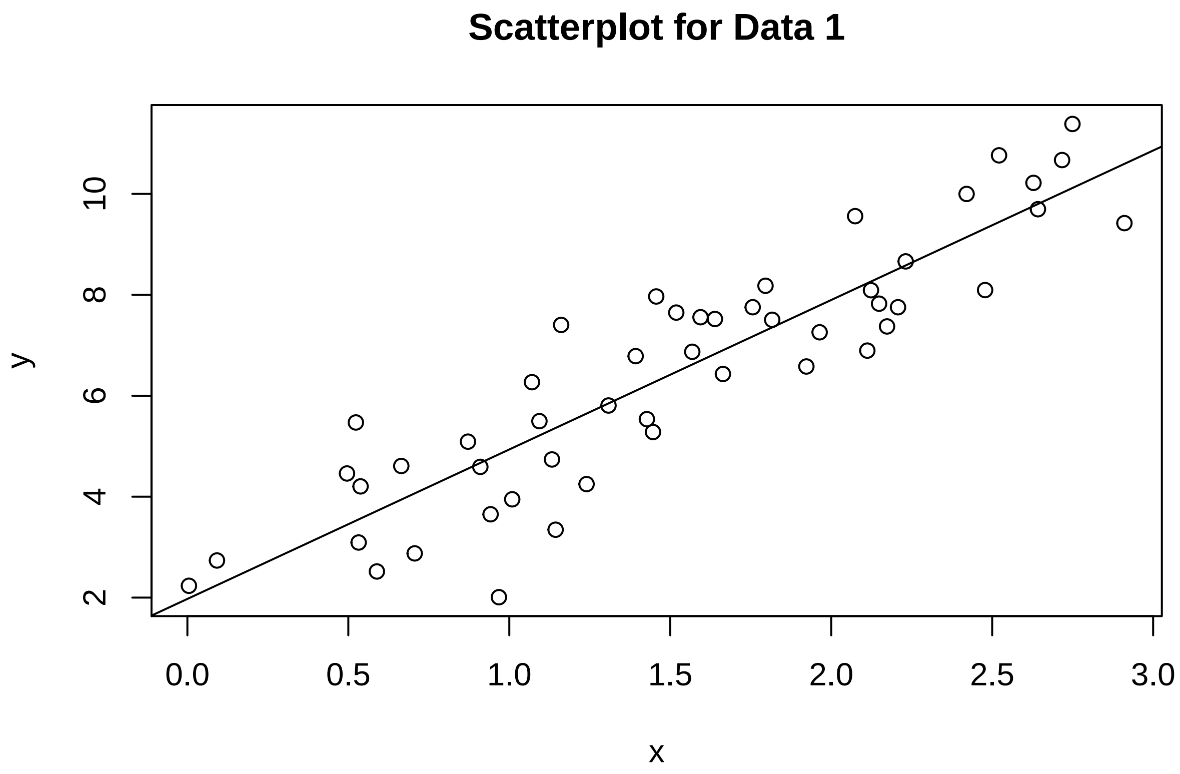
Then create a scatterplot to compare x and y for both sets of data. Add lines of best fit (based on the model you created) for each.

Include a title for each. All other formatting optional.

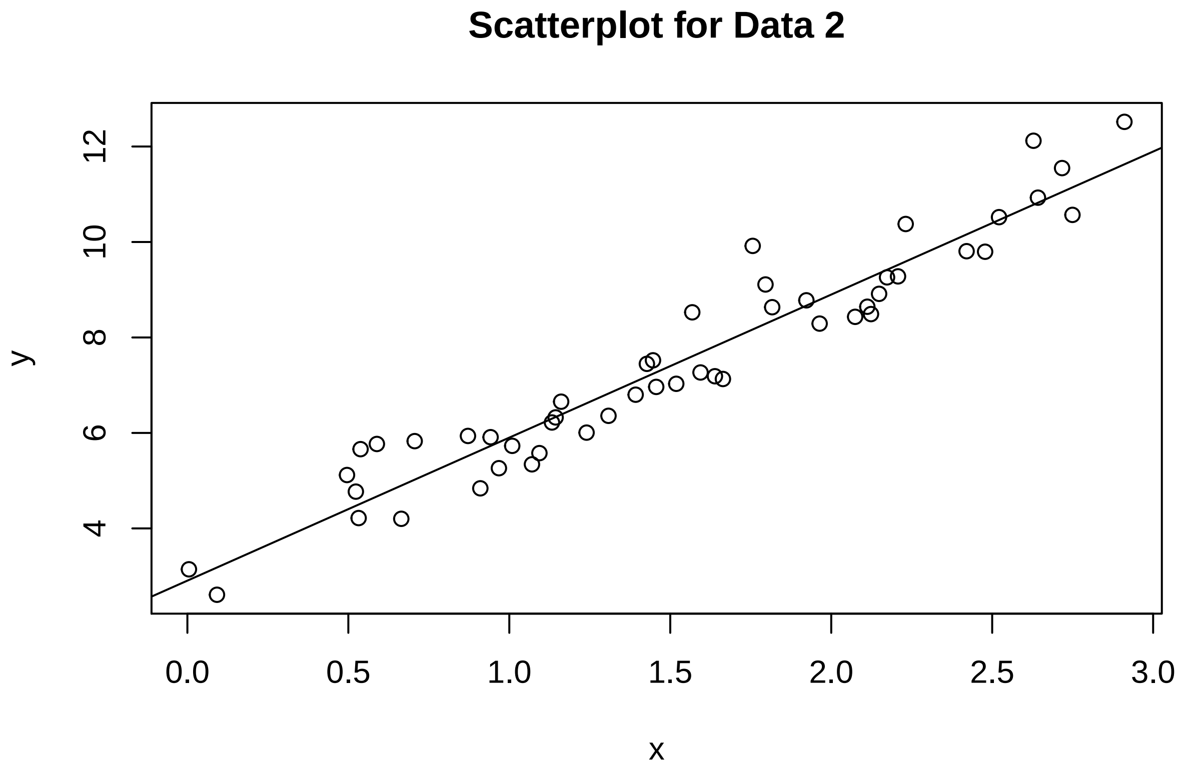
Note: For this homework, you do not need to use ggplot2 for any graphs (unless you want to!). We will be focused on simply checking model diagnostics, which means base R plots are fine. ggplot2 is best when making fancier and cleaner plots for purposes of presentation.

```
#solution
fit_1 = lm(y ~ x, data = data_1)
fit_2 = lm(y ~ x, data = data_2)

plot(y ~ x, data_1, main = "Scatterplot for Data 1")
abline(fit_1)
```



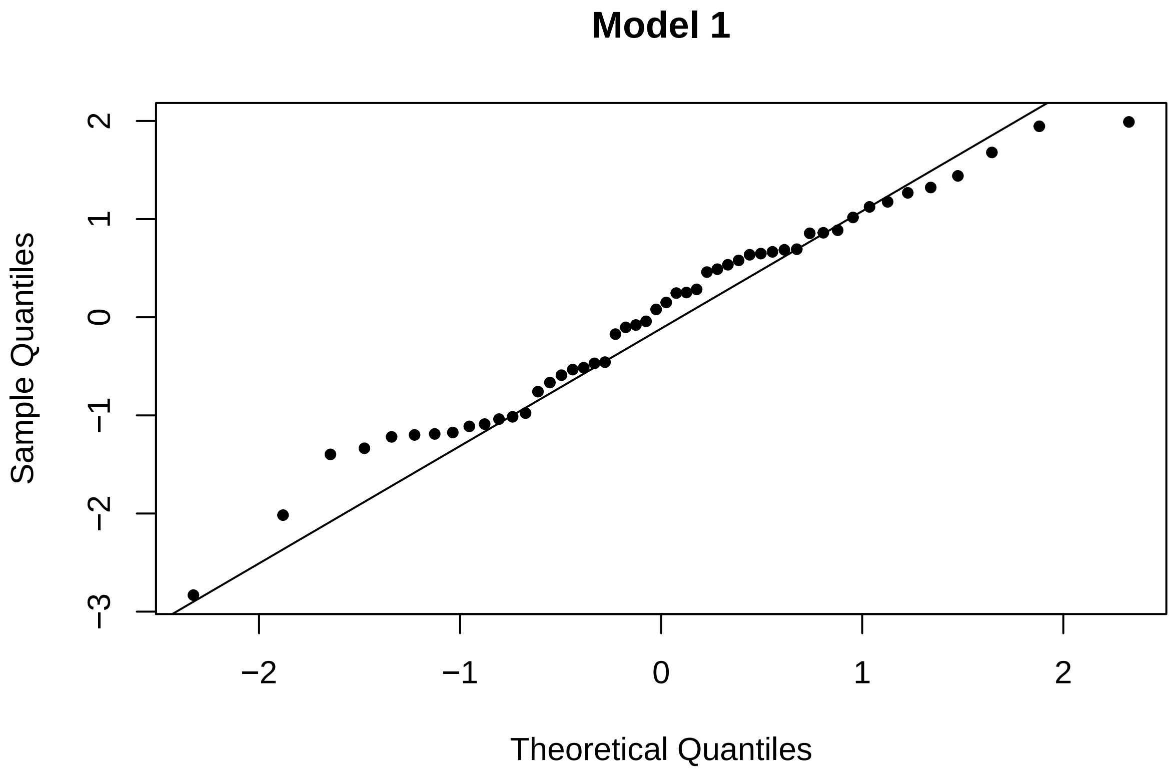
```
plot(y ~ x, data_2, main = "Scatterplot for Data 2")
abline(fit_2)
```



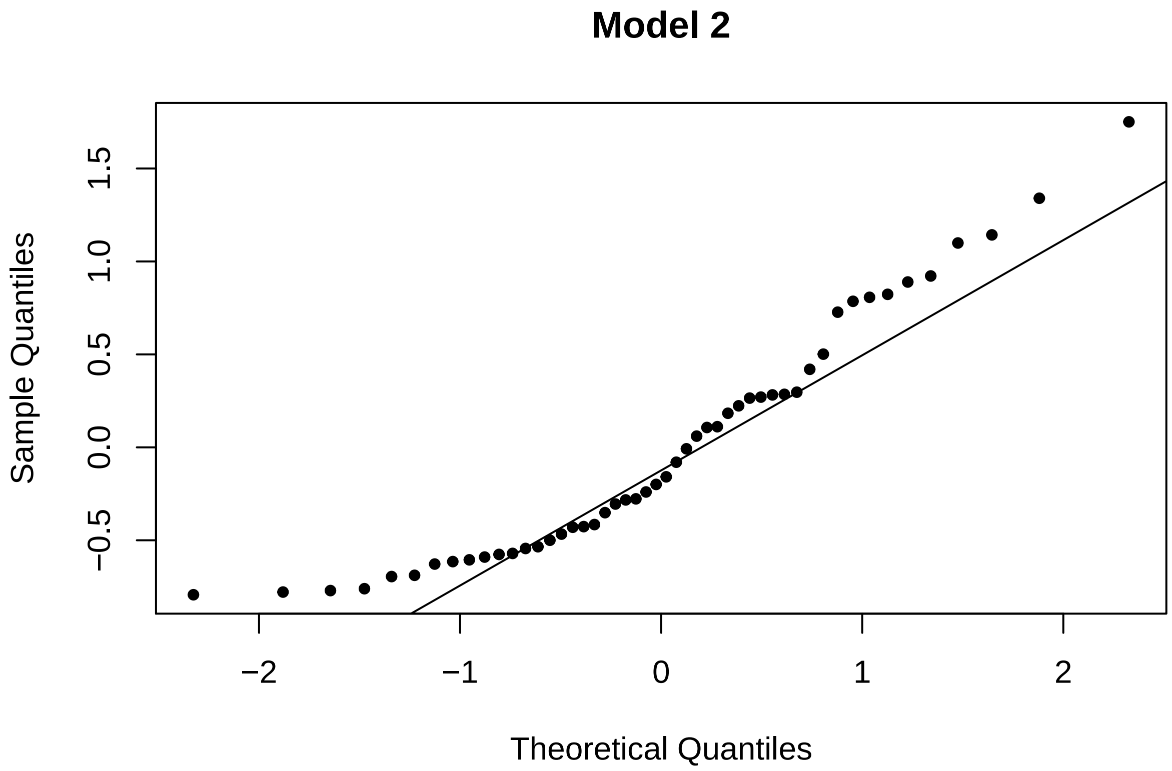
Exercise 2

Continue using the data from Exercise 1. For both fitted regressions, create a Normal Q-Q Plot. Include titles and axes labels. All other formatting optional.

```
# solution
qqnorm(resid(fit_1), pch = 20, main = "Model 1")
qqline(resid(fit_1))
```



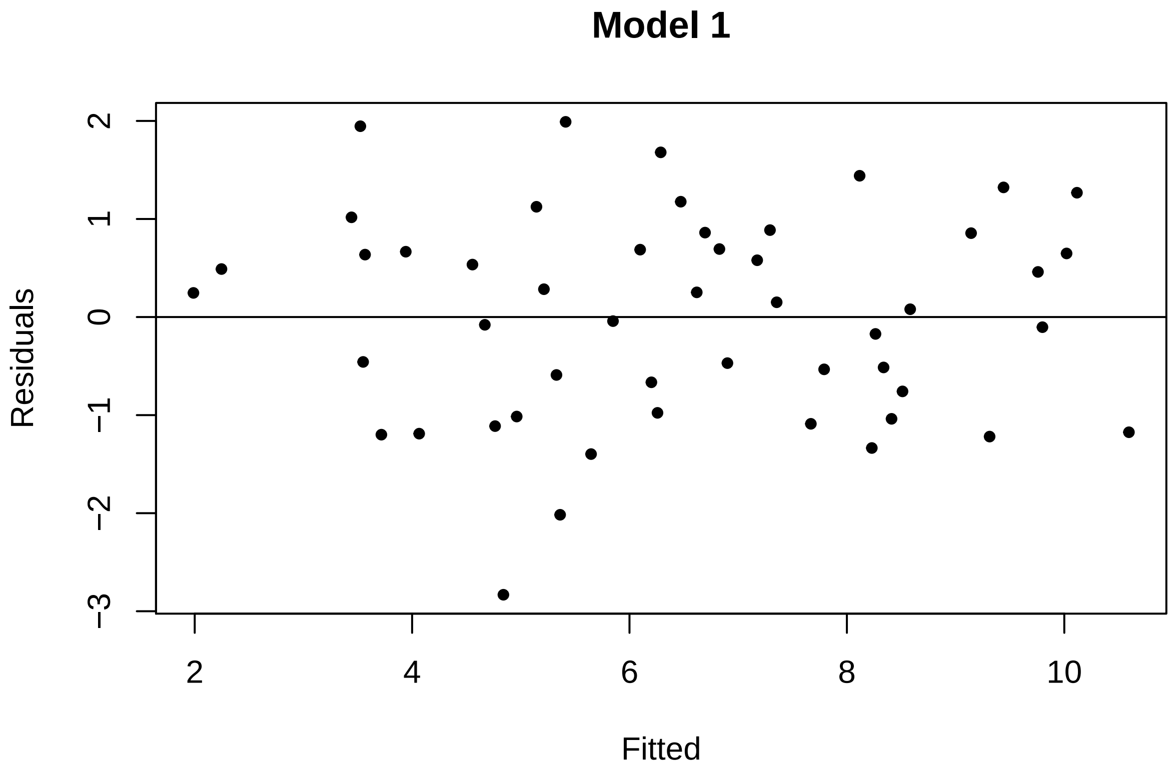
```
qqnorm(resid(fit_2), pch = 20, main = "Model 2")
qqline(resid(fit_2))
```



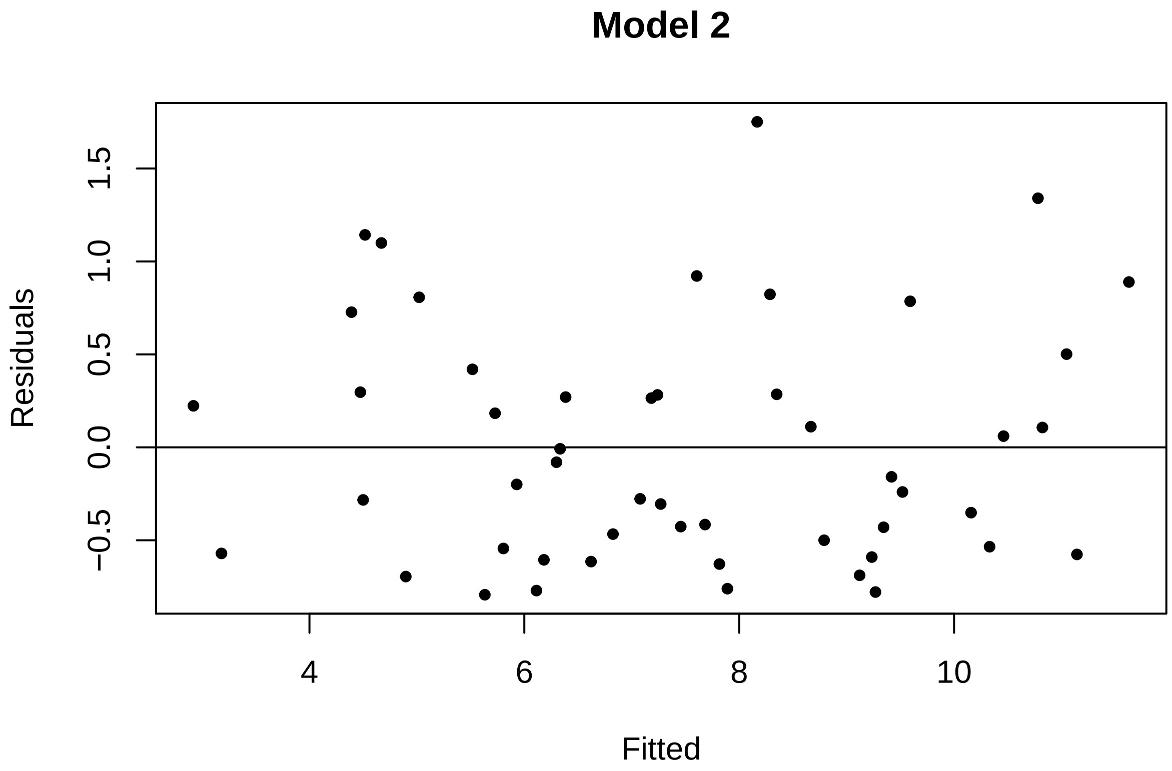
Exercise 3

Continue using the data from Exercise 1. For both fitted regressions, create Fitted versus Residuals plot. Include titles and axes labels. All other formatting optional.

```r
# solution
plot(fitted(fit_1), resid(fit_1), pch = 20, xlab = "Fitted", ylab = "Residuals", main = "Model 1")
abline(h = 0)
```



```
plot(fitted(fit_2), resid(fit_2), pch = 20, xlab = "Fitted", ylab = "Residuals", main = "Model 2")
abline(h = 0)
```



Exercise 4

Continue using the data created from Exercise 1. Run a Shapiro-Wilk test on both model residuals and report the p-values from each.

```
#solution
shapiro.test(resid(fit_1))$p.value
```

```
## [1] 0.4531397
```

```
shapiro.test(resid(fit_2))$p.value
```

```
## [1] 0.002561988
```

Exercise 5

Continue using the data from Exercise 1. Use a Breusch-Pagan Test to check for heteroscedasticity.

Note, you may need to install the `lmtest` package first. Then delete or # that line before trying to knit the file.

```
# solution
#install.packages("lmtest")
library(lmtest)
```

```
## Warning: package 'lmtest' was built under R version 4.0.4

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.0.4

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
bptest(fit_1)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  fit_1
## BP = 0.7475, df = 1, p-value = 0.3873
```

```
bptest(fit_2)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  fit_2
## BP = 0.11121, df = 1, p-value = 0.7388
```

Exercise 6

Continue using the data from Exercise 1. After reviewing all of the diagnostic results above, which conclusions seem appropriate? One or more statements may be true! **Bold** your selections.

- The normality assumption for data_1 is suspect
- **The normality assumption for data_2 is suspect**
- The constant variance assumption for data_1 is suspect
- The constant variance assumption for data_2 is suspect

Exercise 7

For exercises 7 - 12, use the `LifeCycleSavings` dataset which is built into `R`.

```
#starter
```

Fit a multiple linear regression model with `sr` as the response and the remaining variables as predictors. What proportion of observations have a standardized residual less than 2 in magnitude?

Hint: Remember that some standardized residuals are negative and some are positive. Take the absolute value!

Hint: Remember you can take the `mean` of a vector of logicals (TRUE and FALSE values) to calculate the proportion that are true!

```r
# solution
mod = lm(sr ~ ., data = LifeCycleSavings)
mean(abs(rstandard(mod)) < 2)
```

```
## [1] 0.96
```

Exercise 8

Continue using the model fit in Exercise 7. Note that each observation is about a particular country. Which country (observation) has the standardized residual with the largest magnitude?

Hint: The country name is a rowname, so you can use the function `names` to get that entry!

```r
# solution
mod = lm(sr ~ ., data = LifeCycleSavings)
names(which.max(abs(rstandard(mod))))
```

```
## [1] "Zambia"
```

Acceptable solution inputs: "Zambia", Zambia, zambia

Exercise 9

Continue using the model fit in Exercise 7. How many observations have "high" leverage? Use twice the average leverage as the cutoff for "high."

Hint: Use `sum` to count the number of `TRUE` entries in a vector of logicals.

```r
# solution
mod = lm(sr ~ ., data = LifeCycleSavings)
sum(hatvalues(mod) > 2 * mean(hatvalues(mod)))
```

```
## [1] 4
```

Exercise 10

Continue using the model fit in Exercise 4. Which country (observation) has the largest leverage?

Hint: `which.max` may be helpful!

Hint: This question is asking for the Country name, which is a rowname. You can extract that rowname by using the function `names` to extract simply the rowname.

```r
# solution
mod = lm(sr ~ ., data = LifeCycleSavings)
which.max(hatvalues(mod))
```

```
## Libya
##    49
```

- Acceptable solution inputs: "Libya", Libya, libya

Exercise 11

Continue using the model fit in Exercise 7. Report the largest Cook's Distance for observations in this dataset.

```r
# solution
mod = lm(sr ~ ., data = LifeCycleSavings)
max(cooks.distance(mod))
```

```
## [1] 0.2680704
```

Exercise 12

Continue using the model fit in Exercise 7. We wish to exclude observations from the model that are overly influential (in general practice, this should involve more contextual thought before simply ignoring the data, but we will skip that for sake of the coding practice!)

Use $\frac{4}{n}$ as the cutoff for labeling an observation "influential."

Then refit a new model to this data that excludes these influential observations. Report the summary of this new model.

```r
# solution
keep = cooks.distance(mod) < 4 / length(resid(mod))
new_mod = lm(sr ~ ., data = LifeCycleSavings, subset = keep)
summary(new_mod)
```

```
##
## Call:
## lm(formula = sr ~ ., data = LifeCycleSavings, subset = keep)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.4552 -2.5129 -0.1117  1.7477  6.6646
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) 19.8321854  7.8341981   2.531   0.0152 *
## pop15       -0.3047007  0.1513371  -2.013   0.0505 .
## pop75       -0.3030249  1.1135478  -0.272   0.7869
## dpi         -0.0005535  0.0008469  -0.654   0.5170
## ddpi         0.4137823  0.2526006   1.638   0.1089
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 3.441 on 42 degrees of freedom
## Multiple R-squared:  0.3503, Adjusted R-squared:  0.2885
## F-statistic: 5.662 on 4 and 42 DF,  p-value: 0.0009742
```