Problem Set 04: Linear Regression

Your Name

Last modified on August 24, 2024 10:22:37 Eastern Daylight Time

Background

For this problem set you will first run through an example of a simple linear regression, answering a few questions on the way. Then you will work through a regression analysis independently. Knit this file…and you can read through all the instructions.

We will look at some demographic data from the `fivethirtyeight` package recorded for 48 voting areas in the US states just after the 2016 presidential election. We will investigate what variables within those regions might be tied to the percentage of US voters that supported Donald Trump, and in turn, which variables might be useful to predict Trump support in other regions (i.e. to a wider US population).

Setup

Load Packages

We will read the data in with the `readr` package, explore the data using the `dplyr` package and visualize the data using the `ggplot2` package. The `moderndive` package includes some nice functions to show regression model outputs.

R Code

```r
library(dplyr)
library(ggplot2)
library(readr)
library(moderndive)
```

The Data

The following uses the function `read_csv()` to read a `*.CSV` file of the data from where it is published on the web.

R Code

```
url <- "https://docs.google.com/spreadsheets/d/e/2PACX-1vT8qHdvTPaRc62hU94ShBcSh04HP3c11
if(!dir.exists("./data/")){
  dir.create("./data/")
}
if(!file.exists("./data/trump.csv")){
    download.file(url, destfile = "./data/trump.csv")}
trump <- read_csv("./data/trump.csv")
```

Take a moment to look at the data in the viewer or by using `glimpse()`.

```
glimpse(trump)
```

```
Rows: 48
Columns: 4
$ hs_ed         <dbl> 90.4, 80.6, 91.0, 89.0, 89.0, 84.7, 89.7, 86.4, 82.8, 84~
$ poverty       <dbl> 7, 9, 10, 8, 6, 10, 9, 7, 10, 8, 6, 10, 8, 7, 7, 12, 5, ~
$ non_white     <dbl> 81, 61, 6, 27, 50, 42, 31, 37, 62, 28, 30, 26, 37, 44, 3~
$ trump_support <dbl> 30, 33, 33, 34, 35, 37, 38, 39, 40, 40, 41, 41, 42, 42, ~
```

The explanatory variables include:

- `hs_ed` - the percentage of the adults in the region with a high school education.
- `poverty`- the percentage of the "white" households in the region in poverty.
- `non_white`- the percentage of humans in a region that identify as a person of color.

The outcome variable `trump_support` is the percentage of votes for Trump in 2016 in each region.

Observe that all percentages are expressed as values between 0 and 100, and not 0 and 1.

An Example/Demo

Visualization

We will start by investigating the relationship between white poverty levels and support for Trump.

We'll do this by creating a scatterplot with `trump_support` as the outcome variable on the y-axis and `poverty` as the explanatory variable on the x-axis. Note the use of the `geom_smooth()` function, that tells `R` to add a regression line. While the points do scatter/vary around the blue regression line, of all possible lines we can draw in this point of clouds, the blue line is the "best-fitting" line in that in minimizes the sum of the squared residuals.

R Code

```r
ggplot(data = trump, aes(y = trump_support, x = poverty)) +
  geom_point() +
  geom_smooth(method = "lm", se = FALSE) +
  labs(x = "Percentage of white households in poverty",
       y = "Percentage of voters supporting Trump",
       title = "Trump support and white poverty in the US") +
  theme_bw() +
  theme(plot.title = element_text(hjust = 0.5))
```

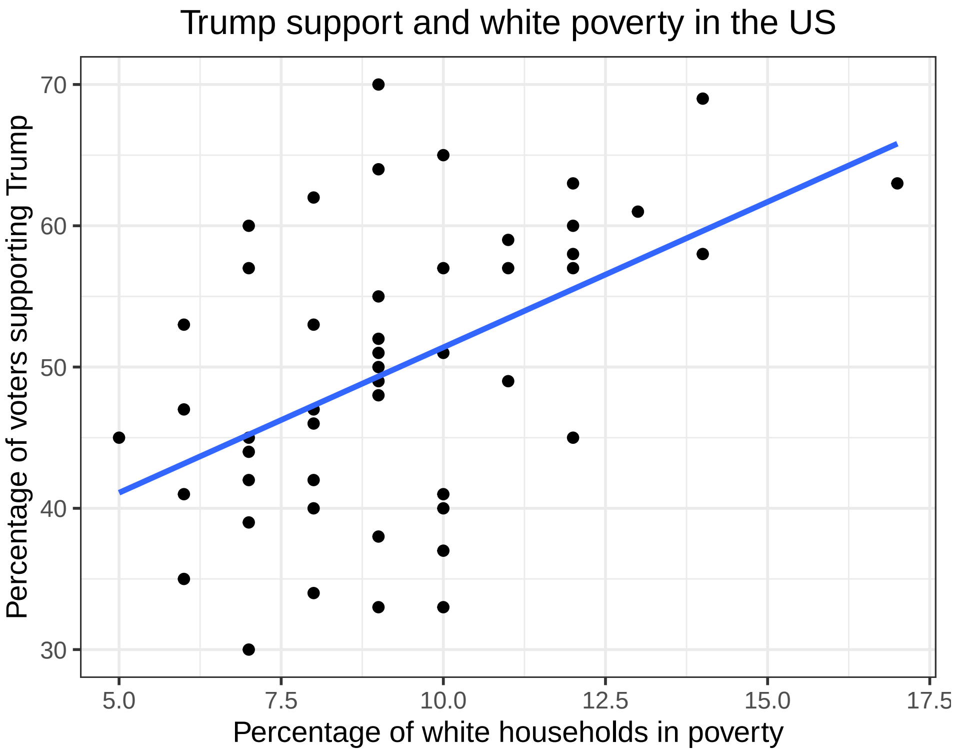


Figure 1: Percentage of voters supporting Trump versus the percentage of white households in poverty.

Problem 1

Does the relationship in Figure 1 appear to be positive or negative? Does the relationship in Figure 1 look reasonably linear?

Problem 1 Answers

- Delete this and put your text answer here.

The Correlation Coefficient (r)

We can numerically quantify the strength of the linear relationship between the two variables with the correlation coefficient. The following tells `R` to `summarize()` the correlation coefficient between the numerical variables `poverty` and `trump_support`. Note that the correlation coefficient only exists for pairs of numerical variables.

R Code

```
trump |>
  summarise(r = cor(trump_support, poverty))
```

```
# A tibble: 1 x 1
      r
  <dbl>
1 0.486
```

```
# Or
trump |> get_correlation(trump_support ~ poverty)
```

```
# A tibble: 1 x 1
    cor
  <dbl>
1 0.486
```

Running a Linear Regression Model

In `R` we can fit a linear regression model (a regression line), like so:

Note

```
poverty_mod <- lm(trump_support ~ poverty, data = trump)
```

Note that:

- the function `lm()` is short for "linear model"
- the first argument is a *formula* in the form `y ~ x` or in other words `outcome variable ~ explantory variable`.
- the second argument is the data frame in which the outcome and explanatory variables can be found.
- we **SAVED THE MODEL RESULTS** as an object called `poverty_mod`

This object `poverty_mod` contains all of the information we need about the linear model that was just fit and we'll be accessing this information again later.

Get the Regression Table

The `get_regression_table()` function from the `moderndive` package will output a regression table. Let's focus on the value in the second column: an estimate for 1) an intercept, and 2) a slope for the `poverty` variable. We'll revisit what the other columns mean in a future problem set.

R Code

```r
get_regression_table(poverty_mod)
```

```
# A tibble: 2 x 7
  term      estimate std_error statistic p_value lower_ci upper_ci
  <chr>        <dbl>     <dbl>     <dbl>   <dbl>    <dbl>    <dbl>
1 intercept    30.8      5.22       5.90       0   20.3      41.3
2 poverty       2.06     0.545      3.78       0    0.961     3.16
```

```r
kable(get_regression_table(poverty_mod))
```

term	estimate	std_error	statistic	p_value	lower_ci	upper_ci
intercept	30.806	5.223	5.898	0	20.293	41.320
poverty	2.059	0.545	3.776	0	0.961	3.157

```r
# I prefer
kable(summary(poverty_mod)$coef)
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	30.806367	5.2230395	5.898168	4.00e-07
poverty	2.059089	0.5453428	3.775769	4.56e-04

We can interpret the `intercept` and `poverty` slope like so:

- When the poverty level is 0, the predicted average Trump support is 30.81%.
- For every increase in poverty level of 1 percentage point, there is an **associated increase** in Trump support of 2.06 percentage points.

Revisiting Figure 1 in Figure 2, we can see that the best-fit line hits the y axis at 30.81 (if we extend it). This is the intercept…the y value at which poverty = 0 (note, a value that is not close to the range of values for "percentage of white households in poverty").

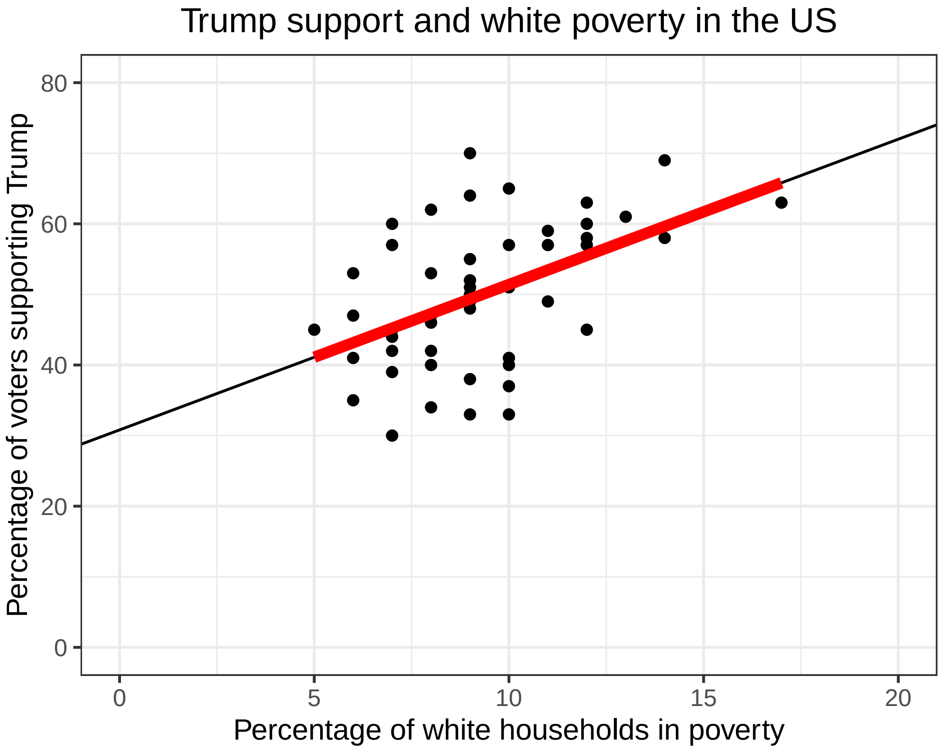


Figure 2: Trump support and white poverty in the US

Problem 2

We found a positive correlation coefficient between `trump_support` and `poverty`. Is it reasonable for us to conclude that social policies that increase white poverty will **cause** an increase in Trump support? Explain why or why not?

Problem 2 Answers

- Delete this and put your text answer here.

Making Predictions

Based on the `R` output of our model, the following is our least squares regression line for the linear model:

$$\widehat{\text{trump\_support}} = 30.8064 + 2.0591 \times \text{poverty}$$

We can use the least squares regression line of the `trump_support` versus `poverty` relationship (See Figure 2) to **visually** make predictions. For instance at 15% white poverty, the line shows a value of just over 60% Trump support.

To get a **more accurate** prediction, we could actually plug 15% into the regression equation like so:

R Code

```r
y_hat = 30.8064 + 2.0591 * 15
y_hat
```

```
[1] 61.6929
```

```r
# Or better yet
predict(poverty_mod, newdata = data.frame(poverty = 15))
```

```
       1
61.69269
```

Problem 3

What percent of Trump support would you expect at a value of 6% white poverty?

Problem 3 Answers

```
# Type your code and comments inside the code chunk
```

- Delete this and put your text answer here.

Problem 4

Do you think it is a good idea to predict Trump support at 85% white poverty, based on this regression equation? Explain your reasoning. Look at Figure 2 carefully before answering the question.

Problem 4 Answers

- Delete this and put your text answer here.

Residuals

Recall that model residuals are the difference between the **observed values in your data set** and the **values predicted by the line**:

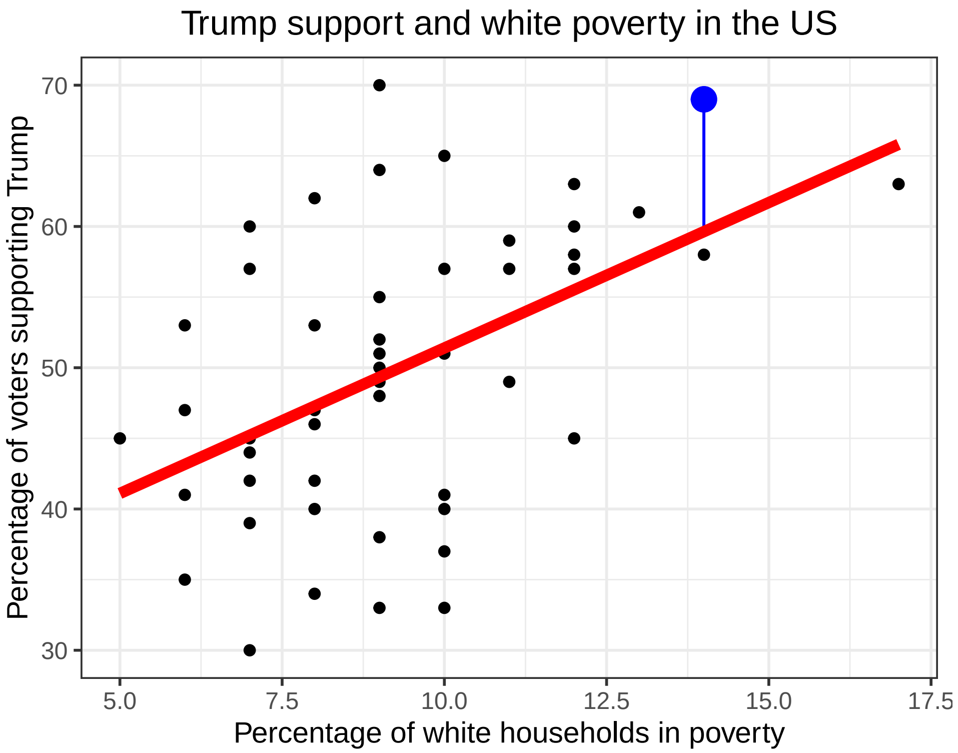
$$\text{residual} = y - \hat{y}$$

For instance, below, one data point is highlighted in blue…the residual is the difference between the y value of the **data point** (here 69), and the y value **predicted** by the line (roughly 59). Here the residual is roughly 10 ($69 - 59 = 10$). The regression equation has under-estimated Trump support, compared to this data point.

The function `get_regression_points()` provides the **fitted** also known as **predicted** value for every data point, and the **residual** for every data point. The first row in the output is the first data point…you see that Trump support was 30%, white poverty was 7%, the regression equation predicted 45.22% Trump support, and the residual was $-15.22 = (30 - 45.22)$.

R Code

```r
results <- get_regression_points(poverty_mod)
kable(head(results))
```



ID	trump_support	poverty	trump_support_hat	residual
1	30	7	45.220	-15.220
2	33	9	49.338	-16.338
3	33	10	51.397	-18.397
4	34	8	47.279	-13.279
5	35	6	43.161	-8.161
6	37	10	51.397	-14.397

Put your Skills to Practice Independently!

Use the same `trump` data set for the following questions:

Problem 5

Generate a scatterplot with a best-fitting line with `non_white` as the explanatory variable, and `trump_support` as the response. Be sure to include an informative title and axis labels for your plot. This will help contextualize the plot.

Problem 5 Answers

```
# Type your code and comments inside the code chunk
```

Problem 6

Do you expect the correlation coefficient (for `non_white` and `trump_support`) to be positive or negative? Write a code chunk testing your prediction (it is OK if your expectation was wrong!).

Problem 6 Answers

- Delete this and put your text answer here.

```
# Type your code and comments inside the code chunk
```

Problem 7

Run a linear regression using `non_white` as the **explanatory** variable, and `trump_support` as the **outcome** variable. Store your linear model in the object `nw_mod`. Use the `get_regression_table()` function on `nw_mod` and interpret the intercept and slope estimates.

Problem 7 Answers

```
# Type your code and comments inside the code chunk
```

- Delete this and put your text answer here.
- Delete this and put your text answer here.

Problem 8

Make a numerical prediction for the level of trump support for a region that has 70% of humans that identify as a person of color. In other words, use **math** not a visual prediction.

Problem 8 Answers

```
# Type your code and comments inside the code chunk
```

- Delete this and put your text answer here.

Problem 9

Based on the evidence you have so far (scatterplots and correlation coefficients), which of the explanatory variables we have considered (`non_white` or `poverty`) seems to be a better explanatory variable of Trump support? Explain.

Problem 9 Answers

```
# Type your code and comments inside the code chunk
```

- Delete this and put your text answer here.

Problem 10

If Representative Ocasio-Cortez saw the regression line (`nw_mod`) and not the actual data:
A. What would her prediction of Trump support be for a region in which 61% of the people identify as non-white?
B. Would her prediction be an overestimate or an underestimate (compared to the observed data), and by how much?
C. In other words, what is the residual for this prediction?

Problem 10 Answers

```
# Type your code and comments inside the code chunk
```

A. Delete this and put your text answer here.

```
# Type your code and comments inside the code chunk
```

B. Delete this and put your text answer here.
C. Delete this and put your text answer here.

Turning in Your Work

You will need to make sure you commit and push all of your changes to the github education repository where you obtained the lab.

> Tip
>
> - Make sure you **render a final copy with all your changes** and work.
> - Look at your final html file to make sure it contains the work you expect and is formatted properly.

Logging out of the Server

There are many statistics classes and students using the Server. To keep the server running as fast as possible, it is best to sign out when you are done. To do so, follow all the same steps for closing Quarto document:

> Tip
>
> - Save all your work.
> - Click on the orange button in the far right corner of the screen to quit `R`
> - Choose **don't save** for the **Workspace image**
> - When the browser refreshes, you can click on the sign out next to your name in the top right.
> - You are signed out.

```r
sessionInfo()
```

```
R version 4.4.1 (2024-06-14)
Platform: x86_64-redhat-linux-gnu
Running under: Red Hat Enterprise Linux 9.4 (Plow)

Matrix products: default
BLAS/LAPACK: FlexiBLAS OPENBLAS-OPENMP;  LAPACK version 3.9.0

locale:
 [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
 [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
 [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
 [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
 [9] LC_ADDRESS=C               LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C

time zone: America/New_York
tzcode source: system (glibc)

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] moderndive_0.6.1 readr_2.1.5      ggplot2_3.5.1    dplyr_1.1.4
[5] knitr_1.48

loaded via a namespace (and not attached):
 [1] utf8_1.2.4          generics_0.1.3      tidyr_1.3.1
 [4] lattice_0.22-6      stringi_1.8.4       hms_1.1.3
 [7] digest_0.6.36       magrittr_2.0.3      evaluate_0.24.0
[10] grid_4.4.1          timechange_0.3.0    fastmap_1.2.0
[13] Matrix_1.7-0        operator.tools_1.6.3 jsonlite_1.8.8
[16] backports_1.5.0     tinytex_0.52        mgcv_1.9-1
[19] purrr_1.0.2         fansi_1.0.6         scales_1.3.0
[22] infer_1.0.7         cli_3.6.3           rlang_1.1.4
[25] crayon_1.5.3        splines_4.4.1       bit64_4.0.5
[28] munsell_0.5.1       withr_3.0.1         yaml_2.3.10
[31] tools_4.4.1         parallel_4.4.1      tzdb_0.4.0
[34] colorspace_2.1-1    broom_1.0.6         vctrs_0.6.5
[37] R6_2.5.1            lifecycle_1.0.4     lubridate_1.9.3
[40] snakecase_0.11.1    stringr_1.5.1       bit_4.0.5
[43] vroom_1.6.5         janitor_2.2.0       pkgconfig_2.0.3
[46] pillar_1.9.0        gtable_0.3.5        glue_1.7.0
[49] xfun_0.47           tibble_3.2.1        tidyselect_1.2.1
[52] rstudioapi_0.16.0   farver_2.1.2        nlme_3.1-166
[55] htmltools_0.5.8.1   labeling_0.4.3      rmarkdown_2.28
[58] formula.tools_1.7.1 compiler_4.4.1
```