Regress to markdown

Andrés L. Parrado, Sumedha Jalote, Ishita Batra, Krishanu Chakraborty

April 24th, 2019

Section 1

To start off

Disclaimer

Disclaimer

- Creative Commons Attribution-NonCommercial-NoDerivs 3.0 License
- R for Data Science

Disclaimer

- Creative Commons Attribution-NonCommercial-NoDerivs 3.0 License
- R for Data Science
- R for Stata Users

Disclaimer

- Creative Commons Attribution-NonCommercial-NoDerivs 3.0 License
- R for Data Science
- R for Stata Users
- R Markdown: The definitive guide

Disclaimer

- Creative Commons Attribution-NonCommercial-NoDerivs 3.0 License
- R for Data Science
- R for Stata Users
- R Markdown: The definitive guide
- Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables. R package version 5.2.2.

Disclaimer

- Creative Commons Attribution-NonCommercial-NoDerivs 3.0 License
- R for Data Science
- R for Stata Users
- R Markdown: The definitive guide
- Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables. R package version 5.2.2.
- Jacson, Simon (2016) Some examples from here

Disclaimer

- Creative Commons Attribution-NonCommercial-NoDerivs 3.0 License
- R for Data Science
- R for Stata Users
- R Markdown: The definitive guide
- Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables. R package version 5.2.2.
- Jacson, Simon (2016) Some examples from here
- Fox, J. and Weisberg, S. (2011) An R Companion to Applied Regression, Second Edition, Sage.

Disclaimer

- Creative Commons Attribution-NonCommercial-NoDerivs 3.0 License
- R for Data Science
- R for Stata Users
- R Markdown: The definitive guide
- Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables. R package version 5.2.2.
- Jacson, Simon (2016) Some examples from here
- Fox, J. and Weisberg, S. (2011) An R Companion to Applied Regression, Second Edition, Sage.
- The World Wide Web

Section 2

Basics of R Markdown

What is R Markdown?

What is R Markdown?

- The document format “R Markdown” was first introduced in the `knitr` package (Xie 2015, 2019b) in early 2012.

What is R Markdown?

- The document format "R Markdown" was first introduced in the `knitr` package (Xie 2015, 2019b) in early 2012.
- Idea was to embed code chunks (of R or other languages) in Markdown documents.

What is R Markdown?

- The document format "R Markdown" was first introduced in the `knitr` package (Xie 2015, 2019b) in early 2012.
- Idea was to embed code chunks (of R or other languages) in Markdown documents.
- In fact, `knitr` supported several authoring languages from the beginning in addition to Markdown, including LaTeX, HTML, AsciiDoc, reStructuredText, and Textile.

What can R Markdown do?

The `rmarkdown package` (J. Allaire, Xie, McPherson, et al. 2019) was first created in early 2014. At this point, there are a large number of tasks that you could do with R Markdown:

- Compile a single R Markdown document to a report in different formats, such as PDF, HTML, or Word.

What can R Markdown do?

The `rmarkdown package` (J. Allaire, Xie, McPherson, et al. 2019) was first created in early 2014. At this point, there are a large number of tasks that you could do with R Markdown:

- Compile a single R Markdown document to a report in different formats, such as PDF, HTML, or Word.
- Create notebooks in which you can directly run code chunks interactively.

What can R Markdown do?

The `rmarkdown package` (J. Allaire, Xie, McPherson, et al. 2019) was first created in early 2014. At this point, there are a large number of tasks that you could do with R Markdown:

- Compile a single R Markdown document to a report in different formats, such as PDF, HTML, or Word.
- Create notebooks in which you can directly run code chunks interactively.
- Make slides for presentations (HTML5, LaTeX Beamer, or PowerPoint).

What can R Markdown do?

The `rmarkdown package` (J. Allaire, Xie, McPherson, et al. 2019) was first created in early 2014. At this point, there are a large number of tasks that you could do with R Markdown:

- Compile a single R Markdown document to a report in different formats, such as PDF, HTML, or Word.
- Create notebooks in which you can directly run code chunks interactively.
- Make slides for presentations (HTML5, LaTeX Beamer, or PowerPoint).
- Produce dashboards and build interactive applications based on Shiny.

What can R Markdown do?

The `rmarkdown package` (J. Allaire, Xie, McPherson, et al. 2019) was first created in early 2014. At this point, there are a large number of tasks that you could do with R Markdown:

- Compile a single R Markdown document to a report in different formats, such as PDF, HTML, or Word.
- Create notebooks in which you can directly run code chunks interactively.
- Make slides for presentations (HTML5, LaTeX Beamer, or PowerPoint).
- Produce dashboards and build interactive applications based on Shiny.
- Write journal articles and author books of multiple chapters, websites and blogs.

Install required packages - rmarkdown

Install the rmarkdown package in R

```
# Install from CRAN

install.packages("rmarkdown")

# Or if you want to test the development
# version, install from GitHub

install.packages("devtools")
devtools::install_github("rstudio/rmarkdown")
```

Install required packages - tinytex

If you want to generate PDF output, you will need to install LaTeX. For R Markdown users who have not installed LaTeX before, we recommend that you install TinyTeX:

```
install.packages("tinytex")
tinytex::install_tinytex()  # install TinyTeX
```

Start your own Markdown document

Hands on!

Markdown Syntax

- Inline formatting

Markdown Syntax

- Inline formatting
- Block-level elements

R code chunks and inline R code

- Figures

R code chunks and inline R code

- Figures
- Tables

Chunk options - You have got the power!

- `eval`

Chunk options - You have got the power!

- `eval`
- `echo`

Chunk options - You have got the power!

- `eval`
- `echo`
- `results`

Chunk options - You have got the power!

- `eval`
- `echo`
- `results`
- `collapse`

Chunk options - You have got the power!

- `eval`
- `echo`
- `results`
- `collapse`
- `warning`, `message`, and `error`

Chunk options - You have got the power!

- `eval`
- `echo`
- `results`
- `collapse`
- `warning`, `message`, and `error`
- `include`

Chunk options - You have got the power!

- `eval`
- `echo`
- `results`
- `collapse`
- `warning`, `message`, and `error`
- `include`
- `cache`

Chunk options - You have got the power!

- `eval`
- `echo`
- `results`
- `collapse`
- `warning`, `message`, and `error`
- `include`
- `cache`
- `fig.width` and `fig.height`

Chunk options - You have got the power!

- `eval`
- `echo`
- `results`
- `collapse`
- `warning`, `message`, and `error`
- `include`
- `cache`
- `fig.width` and `fig.height`
- `fig.align`

Chunk options - You have got the power!

- `eval`
- `echo`
- `results`
- `collapse`
- `warning`, `message`, and `error`
- `include`
- `cache`
- `fig.width` and `fig.height`
- `fig.align`
- `fig.cap`

Chunk options - You have got the power!

- `eval`
- `echo`
- `results`
- `collapse`
- `warning`, `message`, and `error`
- `include`
- `cache`
- `fig.width` and `fig.height`
- `fig.align`
- `fig.cap`
- `dev`

Chunk options - You have got the power!

- `eval`
- `echo`
- `results`
- `collapse`
- `warning`, `message`, and `error`
- `include`
- `cache`
- `fig.width` and `fig.height`
- `fig.align`
- `fig.cap`
- `dev`
- `child`

Comparing means with a t-test

```
library(haven)
library(stargazer)
library(tidyverse)

boot_camp <- read_dta("Stata files/jpal_tracking_bootcamp.dta")
t_test_results <- t.test(endline_score ~
    tracking, data = boot_camp)
t_test_results
```

```
##
##  Welch Two Sample t-test
##
## data:  endline_score by tracking
## t = -6.0602, df = 5469.2, p-value = 1.45e-09
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.442331 -1.248427
## sample estimates:
## mean in group 0 mean in group 1
##        18.91380        20.75918
```

Comparing means with regression (1/2)

```
reg_results <- lm(endline_score ~ tracking,
    data = boot_camp)
summary(reg_results)
```

```
##
## Call:
## lm(formula = endline_score ~ tracking, data = boot_camp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -20.759  -9.059  -1.300   9.741  26.901
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  18.9138     0.2186   86.53  < 2e-16 ***
## tracking      1.8454     0.3045    6.06 1.46e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 11.28 on 5487 degrees of freedom
## Multiple R-squared:  0.006647,   Adjusted R-squared:  0.006466
## F-statistic: 36.72 on 1 and 5487 DF,  p-value: 1.456e-09
```

Comparing means with regression (2/2)

```
stargazer(reg_results, header = FALSE, title = "Difference in Means with Regression")
```

Table 1: Difference in Means with Regression

	Dependent variable:
	endline_score
tracking	1.845***
	(0.305)
Constant	18.914***
	(0.219)
Observations	5,489
R^2	0.007
Adjusted R^2	0.006
Residual Std. Error	11.276 (df = 5487)
F Statistic	36.718*** (df = 1; 5487)
Note:	*p<0.1; **p<0.05; ***p<0.01

Section 3

Regressions in R

1. Linear regression

- Let us load the dataset Prestige first

```
library(haven)
library(tidyverse)
Prestige_dataset <- read_dta("Stata files/Prestige_dataset.dta")
```

1. Linear regression - try it yourself!

- We will focus on linear regressions -

$$E(y) = \alpha + \beta x$$

1. Linear regression - try it yourself!

- We will focus on linear regressions -

$$E(y) = \alpha + \beta x$$

- Remember the grammar: `lm(y ~ x_1 + x_2 + ..., data = dataset)` where `lm` refers to linear model

1. Linear regression - try it yourself!

- We will focus on linear regressions -

$$E(y) = \alpha + \beta x$$

- Remember the grammar: `lm(y ~ x_1 + x_2 + ..., data = dataset)` where `lm` refers to linear model
- But what does the ~ notation mean?

1. Linear regression - try it yourself!

```
regression_model_a <- lm(prestige ~ education +
    log2(income) + women, data = Prestige_dataset)
summary(regression_model_a)
```

```
##
## Call:
## lm(formula = prestige ~ education + log2(income) + women, data = Prestige_dataset)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.364  -4.429  -0.101   4.316  19.179
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -110.9658    14.8429  -7.476 3.27e-11 ***
## education       3.7305     0.3544  10.527  < 2e-16 ***
## log2(income)    9.3147     1.3265   7.022 2.90e-10 ***
## women           0.0469     0.0299   1.568     0.12
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.093 on 98 degrees of freedom
## Multiple R-squared:  0.8351, Adjusted R-squared:   0.83
## F-statistic: 165.4 on 3 and 98 DF,  p-value: < 2.2e-16
```

Install Stargazer

Now, let's install `stargazer`

```
install.packages("stargazer")
library(stargazer)
```

Table

```
library(stargazer)
stargazer(regression_model_a, type = "latex",
    title = "Results", header = FALSE)
```

Table 2: Results

	Dependent variable:
	prestige
education	3.731***
	(0.354)
log2(income)	9.315***
	(1.327)
women	0.047
	(0.030)
Constant	−110.966***
	(14.843)
Observations	102
R^2	0.835
Adjusted R^2	0.830
Residual Std. Error	7.093 (df = 98)
F Statistic	165.428*** (df = 3; 98)
Note:	*p<0.1; **p<0.05; ***p<0.01

Table for multiple models

```
regression_model_a_2 <- lm(prestige ~ education +
    log2(income), data = Prestige_dataset)
summary(regression_model_a_2, title = "Results for multiple models")
```

```
##
## Call:
## lm(formula = prestige ~ education + log2(income), data = Prestige_dataset)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.0346  -4.5657  -0.1857   4.0577  18.1270
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -95.1940    10.9979  -8.656 9.27e-14 ***
## education      4.0020     0.3115  12.846  < 2e-16 ***
## log2(income)   7.9278     0.9961   7.959 2.94e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.145 on 99 degrees of freedom
## Multiple R-squared:  0.831,  Adjusted R-squared:  0.8275
## F-statistic: 243.3 on 2 and 99 DF,  p-value: < 2.2e-16
```

Table with Stargazer

```
stargazer(regression_model_a, regression_model_a_2,
    title = "Results", type = "latex", header = FALSE)
```

Table 3: Results

	Dependent variable:	
	prestige	
	(1)	(2)
education	3.731***	4.002***
	(0.354)	(0.312)
log2(income)	9.315***	7.928***
	(1.327)	(0.996)
women	0.047	
	(0.030)	
Constant	−110.966***	−95.194***
	(14.843)	(10.998)
Observations	102	102
R^2	0.835	0.831
Adjusted R^2	0.830	0.828
Residual Std. Error	7.093 (df = 98)	7.145 (df = 99)
F Statistic	165.428*** (df = 3; 98)	243.323*** (df = 2; 99)
Note:		*p<0.1; **p<0.05; ***p<0.01

2. Linear regresssion (heteroskdasticity - robust standard error)

- What those are is beyond the scope of this course

2. Linear regresssion (heteroskdasticity - robust standard error)

- What those are is beyond the scope of this course
- Have to install `sandwich`, which computes robust covariance matrix estimators

2. Linear regresssion (heteroskdasticity - robust standard error)

- What those are is beyond the scope of this course
- Have to install `sandwich`, which computes robust covariance matrix estimators
- Also need to use that information in a linear model. Have to install `lmtest`

2. Linear regresssion (heteroskdasticity - robust standard error)

```
library(lmtest)
library(sandwich)

regression_model_a$robse <- vcovHC(regression_model_a,
    type = "HC1")
coeftest(regression_model_a, regression_model_a$robse)
```

```
##
## t test of coefficients:
##
##                 Estimate  Std. Error t value  Pr(>|t|)
## (Intercept)  -110.965824   15.275221 -7.2644 9.074e-11 ***
## education       3.730508    0.388808  9.5947 9.176e-16 ***
## log2(income)    9.314666    1.382326  6.7384 1.107e-09 ***
## women           0.046895    0.031484  1.4895    0.1396
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2. Linear regresssion (heteroskdasticity - robust standard error)

```
stargazer(coeftest(regression_model_a, regression_model_a$robse),
    type = "latex", header = FALSE, title = "Heteroskedasticity")
```

Table 4: Heteroskedasticity

	Dependent variable:
education	3.731^{***}
	(0.389)
log2(income)	9.315^{***}
	(1.382)
women	0.047
	(0.031)
Constant	-110.966^{***}
	(15.275)
Note:	*p$<$0.1; **p$<$0.05; ***p$<$0.01

3.Predicted value/ residuals

```
prestige_hat <- fitted(regression_model_a)
as.data.frame(prestige_hat)
```

prestige_hat
65.07260
71.50702
60.16243
54.21544
65.55434
72.70790
67.72890
75.20712
68.75371
68.77237
52.02945
54.37693
62.81355
66.44428
64.60173
62.25138
80.68558
62.59103
70.66091
56.90504
76.27680
59.86614
68.31387
85.31677
77.51847
75.52331

```
prestige_resid <- residuals(regression_model_a)
as.data.frame(prestige_resid)
```

prestige_resid
3.7274014
-2.4070193
3.2375678
2.5845601
7.9456572
4.8921019
4.8711040
2.8928824
4.3462926
0.0276314
9.9705459
5.6230675
-9.0135457
-4.2442830
10.2982666
-7.1513830
1.6144218
-4.4910309
-12.3609075
15.8949567
8.3231955
-0.2661392
-2.2138734
1.8832315
-10.8184657
-7.1233058
11.2998577
-2.5632191
13.6789974
-1.9729288

4. Dummy regressions with no interactions (analysis of covariance, fixed effects)

```
library(tidyverse)
Prestige_dataset$type <- as.factor(Prestige_dataset$type) %>%
    factor(labels = c("bc", "wc", "prof"))
regression_model_c <- lm(prestige ~ education +
    log2(income) + type, data = Prestige_dataset)
```

```r
library(stargazer)
stargazer(regression_model_c, type = "latex",
    header = FALSE, title = "Dummy regressions with no interactions")
```

Table 7: Dummy regressions with no interactions

	Dependent variable:
	prestige
education	3.284***
	(0.608)
log2(income)	7.269***
	(1.190)
typewc	6.751*
	(3.618)
typeprof	−1.439
	(2.378)
Constant	−81.202***
	(13.743)
Observations	98
R^2	0.855
Adjusted R^2	0.849
Residual Std. Error	6.637 (df = 93)
F Statistic	137.643*** (df = 4; 93)
Note:	*p<0.1; **p<0.05; ***p<0.01

5. Dummy regressions with interactions

```
regression_model_d <- lm(prestige ~ type *
    (education + log2(income)), data = Prestige_dataset)
```

Table 8: Dummy regressions with interactions

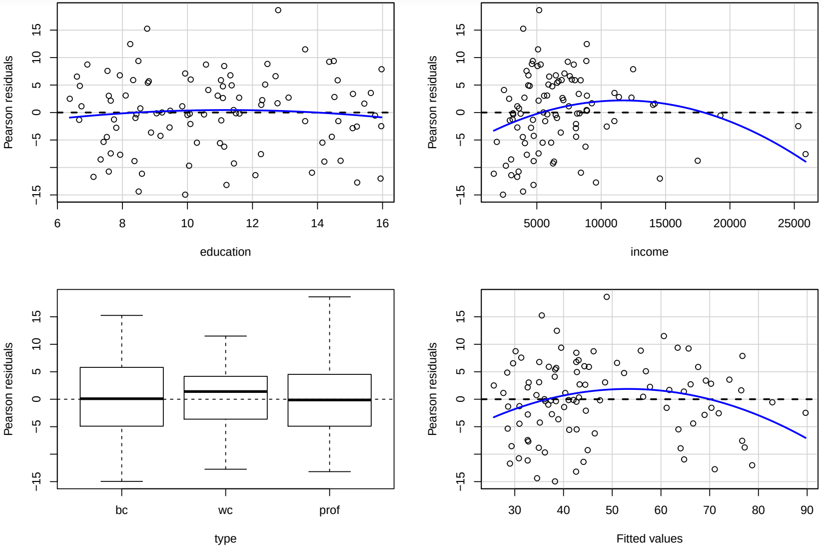
	Dependent variable:
	prestige
typewc	85.160***
	(31.181)
typeprof	30.241
	(37.979)
education	2.336**
	(0.928)
log2(income)	11.078***
	(1.806)
typewc:education	0.697
	(1.290)
typeprof:education	3.640**
	(1.759)
typewc:log2(income)	−6.536**
	(2.617)
typeprof:log2(income)	−5.653*
	(3.052)
Constant	−120.046***
	(20.158)
Observations	98
R^2	0.871
Adjusted R^2	0.859
Residual Std. Error	6.409 (df = 89)
F Statistic	75.147*** (df = 8; 89)
Note:	*p<0.1; **p<0.05; ***p<0.01

Section 4

For the curious - Regression Diagnostics

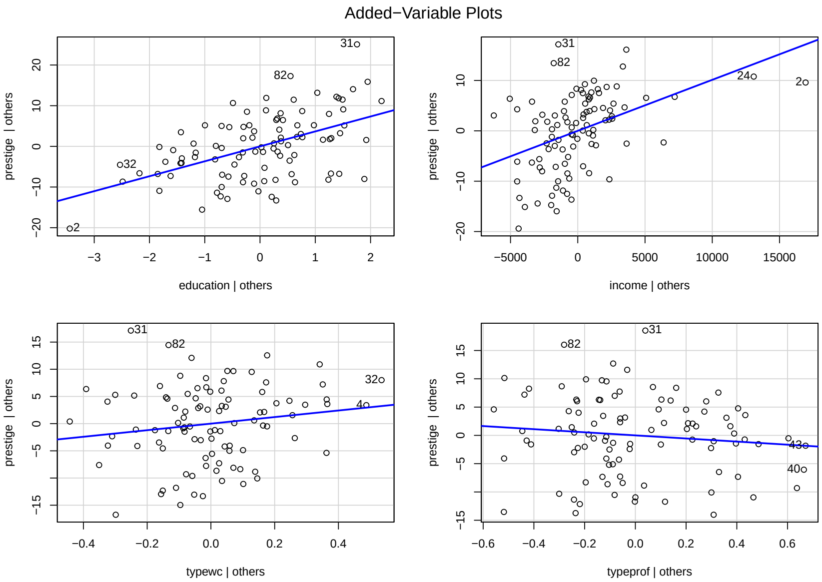
6. Diagnostics for linear regression (residual plots)

```
library(car)
regression_model_e <- lm(prestige ~ education +
    income + type, data = Prestige_dataset)
```

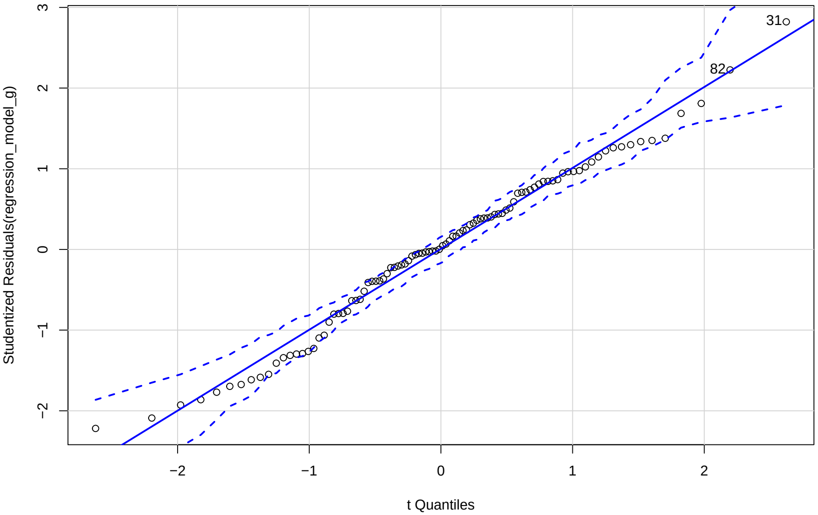



```
##              Test stat Pr(>|Test stat|)
## education    -0.6836          0.495942
## income       -2.8865          0.004854 **
## type
## Tukey test   -2.6104          0.009043 **
## ---
```

7. Influential variable (added variable plot)



8. Outliers (QQ plots)

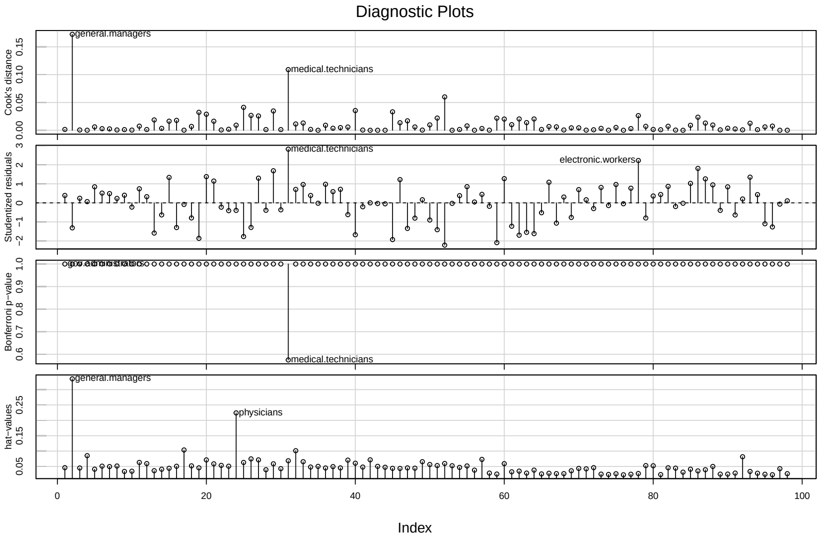



```
## [1] 31 82
```

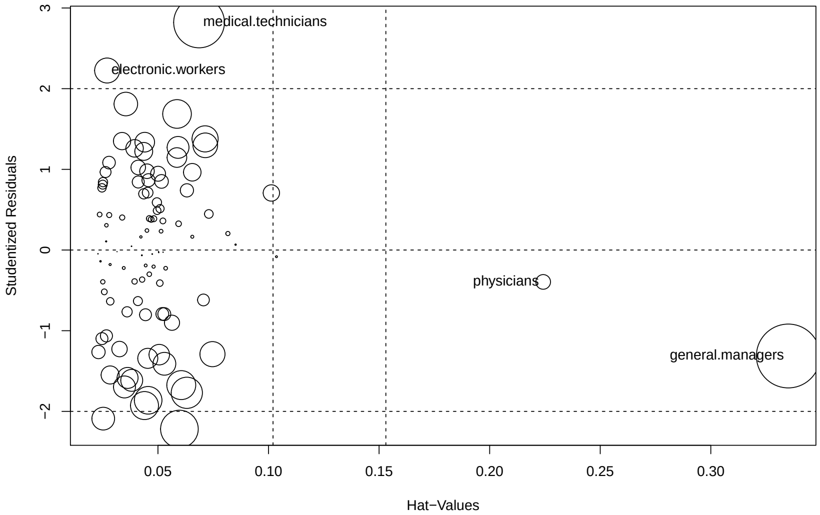
9. Outliers - Bonferonni test

```
## No Studentized residuals with Bonferonni p < 0.05
## Largest |rstudent|:
##                      rstudent unadjusted p-value Bonferonni p
## medical.technicians 2.821091          0.0058632      0.57459
```

10. High Leverage (hat) points

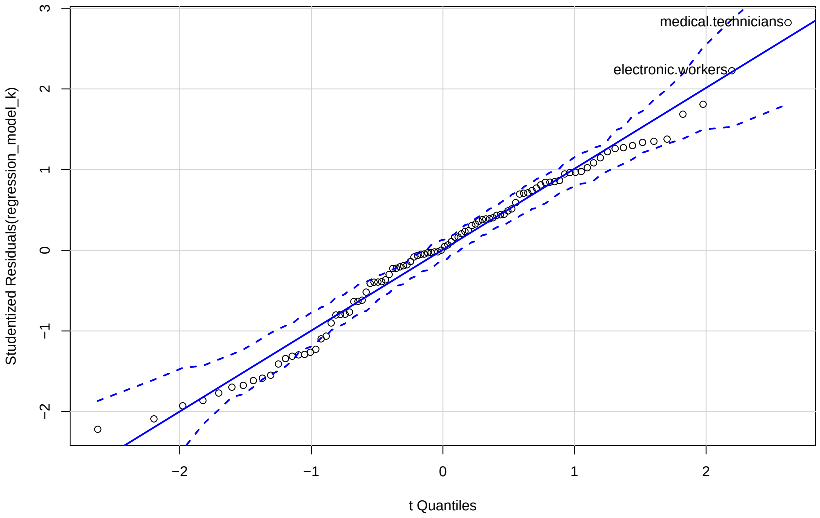


11. Influence Plots



	StudRes	Hat	CookD
general.managers	-1.3134574	0.3350448	0.1725040
physicians	-0.3953204	0.2242031	0.0091155

12. Testing for normality




```
## medical.technicians  electronic.workers
##                  31                  82
```

13. Testing for heteroskedasticity

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.09830307, Df = 1, p = 0.75388
```

14. Testing for multicollineraity

```
##                GVIF Df GVIF^(1/(2*Df))
## education 5.973932  1        2.444163
## income    1.681325  1        1.296659
## type      6.102131  2        1.571703
```

15. Cluster robust standard error

```r
library(car)
library(lmtest)
library(multiwayvcov)

# Need to remove missing before
# clustering
Prestige_dataset_na = na.omit(Prestige_dataset)

# Regular regression using lm()
regression_model_n <- lm(prestige ~ education +
    log2(income) + women, data = Prestige_dataset_na)

# Cluster standard errors by 'type'
regression_model_n$clse <- cluster.vcov(regression_model_n,
    Prestige_dataset_na$type)
clusterred <- coeftest(regression_model_n,
    regression_model_n$clse)
```

```r
stargazer(clusterred, header = FALSE, title = "Clustered standard error",
    no.space = TRUE)
```

Table 10: Clustered standard error

	Dependent variable:
education	3.594***
	(1.003)
log2(income)	10.817**
	(4.407)
women	0.065
	(0.068)
Constant	−129.168***
	(47.025)
Note:	*p<0.1; **p<0.05; ***p<0.01