Rmarkdown for Data Analysis: A Refresher

Haohan Chen[*]

February 2, 2018

This is a refresher of some common Rmarkdown operations, to help you editing your homework and papers more efficiently for PS733. I demonstrate it by writing a mini data analysis report on a toy dataset.

Rmarkdown Setup

```
  # enable setting font size of code chunk
  def.chunk.hook  <- knitr::knit_hooks$get("chunk")
  knitr::knit_hooks$set(chunk = function(x, options) {
    x <- def.chunk.hook(x, options)
    ifelse(options$size != "normalsize",
           paste0("\\", options$size,"\n\n", x, "\n\n \\normalsize"), x)
  })

  # knitr options
  knitr::opts_chunk$set(echo = TRUE, message = FALSE, warning = FALSE,
                        results = "hold",
                        fig.path = "figures/", size = "small")
  # Explanation in the following chunk -- when fontsize is reduced!
```

```
#-----------------------------
# About the Header
#-----------------------------
  # keep_tex: output the tex file (so that you can directly use the tex code generated)
  # fig_caption: show caption of figures. true by default.
  # citation_package: use latex natbib citation package for bibliography. recommended!
  # header_includes: include other command the document's preamble.
  #                  mostly used it to call more LaTeX packages.
#-----------------------------
# Global options
#-----------------------------
  # About the font size mess
    # A trick that enables you to custimize the fontsize of code in the chunk.
    # I have to do this because Rmarkdown does not directly support setting

  # About knitr::opt_chunk
    # echo: show code
    # message, warning: show system generated info (e.g. progress bar)
    # results = "hold". hold output of results till the end of chunk (invalid for fig)
    # fig.path: set a path to store figures generate. can reuse them elsewhere.
    #           without this, no fig will be saved.
    # size: font size of *code in the chunks* (not your main text, which is set
    #       in the header "fontsize: 11pt". options of size include "small", "tiny",
    #       "normalisize", "huge"...

  # Also, Create a directory to save your tables (used later)
  dir.create("tables")
  # Output type of this file is LaTex (a param for later)
  out_type = "latex"
```

[*]Political Science Department, Duke University. haohan.chen@duke.edu

Packages and Dataset Setup

```
#-----------------------------
# load/install required packages
#-----------------------------
# Names of all packages used
  pkgs <- c("dplyr", "ggplot2", "xtable", "stargazer", "PerformanceAnalytics", "cowplot")
# A function to load all above packages. Install if they have not been installed.
  usePackage <- function(p){
    for (pkg in p){
      if (!is.element(pkg, installed.packages()[,1]))
        install.packages(pkg, dep = TRUE, repos = "https://cloud.r-project.org/")
      require(pkg, character.only = TRUE)
    }
  }
  usePackage(pkgs)
```

```
#-----------------------------
# load your data
#-----------------------------
  # Load your dataset of interest.
  # Below is an example economic dataset coming with R
    data("longley")
      # J. W. Longley (1967) An appraisal of least-squares programs from
      # the point of view of the user.
      # Journal of the American Statistical Association 62, 819-841.
  # Just to mess up the dataset by a bit
    names(longley) <- c("gnp.def", "gnp", "unemp", "force", "pop", "yr", "emp")
```

Exploratory Data Analysis

Table 2 shows the descriptive statistics. Figure 1 is the Correlation Matrix. Figure 2 shows the relationship between GNP and the size of armed force using the default `plot` function. Figure 3 is the same plot using `ggplot`.

Table

```
#----------------------------
# Table of summary statistics
#----------------------------
  # Summary statistics
    summary(longley)
    # Not pretty. We can do better!
```

```
##     gnp.def           gnp            unemp           force
##  Min.   : 83.00   Min.   :234.3   Min.   :187.0   Min.   :145.6
##  1st Qu.: 94.53   1st Qu.:317.9   1st Qu.:234.8   1st Qu.:229.8
##  Median :100.60   Median :381.4   Median :314.4   Median :271.8
##  Mean   :101.68   Mean   :387.7   Mean   :319.3   Mean   :260.7
##  3rd Qu.:111.25   3rd Qu.:454.1   3rd Qu.:384.2   3rd Qu.:306.1
##  Max.   :116.90   Max.   :554.9   Max.   :480.6   Max.   :359.4
##       pop              yr            emp
##  Min.   :107.6   Min.   :1947   Min.   :60.17
##  1st Qu.:111.8   1st Qu.:1951   1st Qu.:62.71
##  Median :116.8   Median :1954   Median :65.50
##  Mean   :117.4   Mean   :1954   Mean   :65.32
##  3rd Qu.:122.3   3rd Qu.:1958   3rd Qu.:68.29
##  Max.   :130.1   Max.   :1962   Max.   :70.55
```

```
#-------------------------------------
# Table of summary statistics (con'd)
#-------------------------------------
  # Produce a LaTeX summary stats table (can also be HTML)
  stargazer(longley, title = "Descriptive Statistics",
      mean.sd = TRUE, median = TRUE, iqr = TRUE, min.max = TRUE,
      header = FALSE, label = "tab:desc", type = out_type)
```

Table 1: Descriptive Statistics

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Median	Pctl(75)	Max
gnp.def	16	101.681	10.792	83	94.5	100.6	111.2	117
gnp	16	387.698	99.395	234.289	317.881	381.427	454.085	554.894
unemp	16	319.331	93.446	187.000	234.825	314.350	384.250	480.600
force	16	260.669	69.592	146	229.8	271.8	306.1	359
pop	16	117.424	6.956	107.608	111.788	116.803	122.304	130.081
yr	16	1,954.500	4.761	1,947	1,950.8	1,954.5	1,958.2	1,962
emp	16	65.317	3.512	60.171	62.712	65.504	68.291	70.551

```
    # Will come back to Stargazer soon.
```

Tip: Save your Table

I recommend saving your table in a separate `.tex` file for convenient re-use.

```
  # Instead of directly output your outcome.
  # Saving the output is a better strategy. Think about why.
```

```r
desc_tab <- capture.output(
  stargazer(longley, title = "Descriptive Statistics",
      mean.sd = TRUE, median = TRUE, iqr = TRUE, min.max = TRUE,
      header = FALSE, label = "tab:desc", type = out_type)
  )
# Save it to a folder for tables (created earlier)
writeLines(desc_tab, "tables/descriptive.tex")
```

Table 2: Descriptive Statistics

Statistic	N	Mean	St. Dev.	Min	Pctl(25)	Median	Pctl(75)	Max
gnp.def	16	101.681	10.792	83	94.5	100.6	111.2	117
gnp	16	387.698	99.395	234.289	317.881	381.427	454.085	554.894
unemp	16	319.331	93.446	187.000	234.825	314.350	384.250	480.600
force	16	260.669	69.592	146	229.8	271.8	306.1	359
pop	16	117.424	6.956	107.608	111.788	116.803	122.304	130.081
yr	16	1,954.500	4.761	1,947	1,950.8	1,954.5	1,958.2	1,962
emp	16	65.317	3.512	60.171	62.712	65.504	68.291	70.551

Correlcation Matrix

```
#---------------------------
# Correlation Matrix
#---------------------------
  PerformanceAnalytics::chart.Correlation(longley)
```

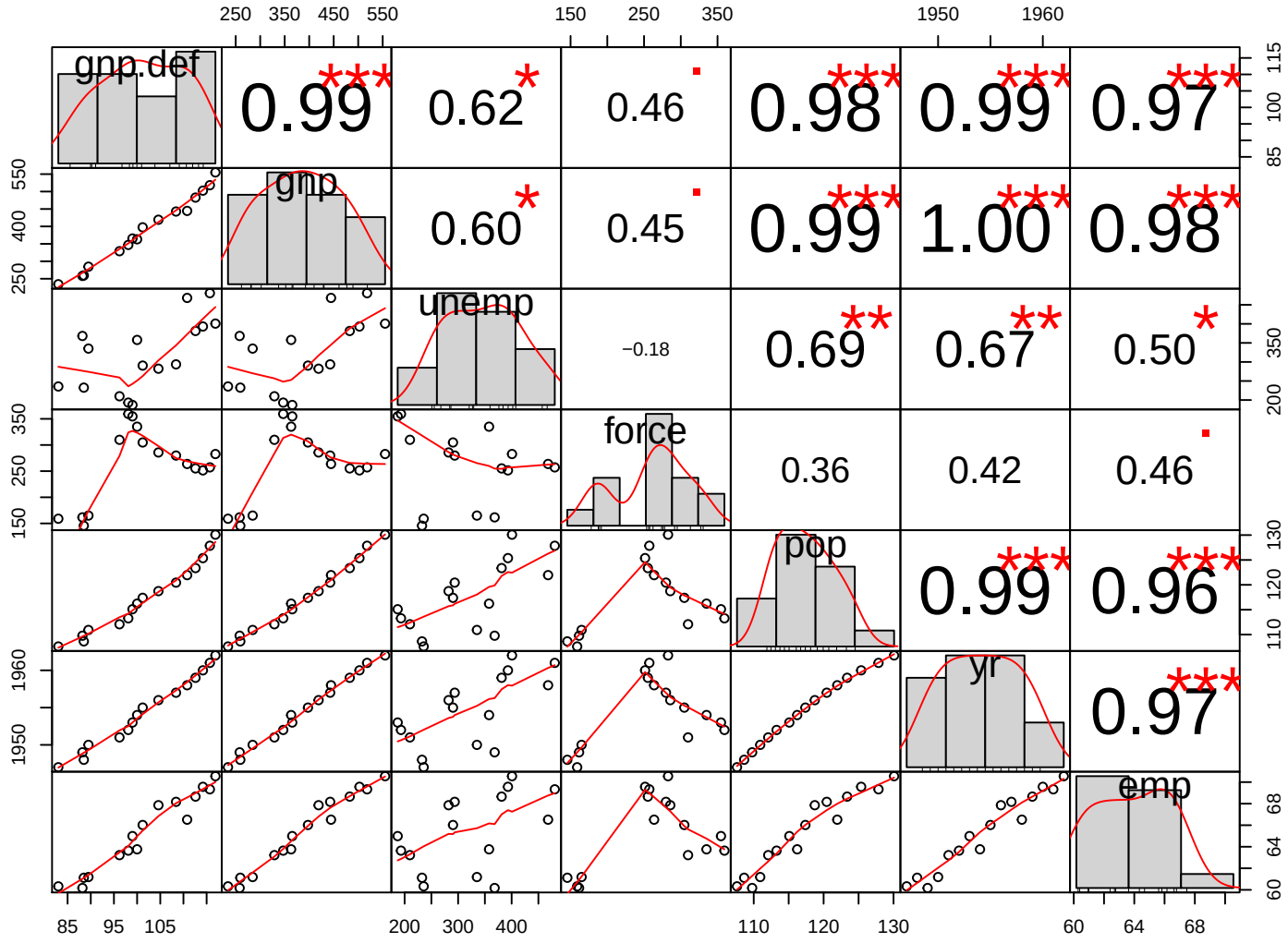

Figure 1: Correlation Matrix

```
  # By far my favorite, better than other fancy stuff.
  # Perfect for continuous variables
```

Correlation Plots (and their arrangement)

```
par(mfrow = c(1, 2)) # 2 figures in a row
plot(longley$gnp, longley$force, xlab = "GNP", ylab = "Size of Armed Force", main = "GNP")
plot(log(longley$gnp), longley$force, xlab = "log(GNP)", ylab = "Size of Armed Force",
     main = "log(GNP)")
```

```
# The cowplot package: https://cran.r-project.org/web/packages/cowplot/vignettes/plot_grid.html
fig_cor1 <- ggplot(longley, aes(x = gnp, y = force)) + geom_point() +
  geom_smooth(method = "loess") + xlab("GNP") + ylab("Size of Armed Force") +
  ggtitle("GNP")
```

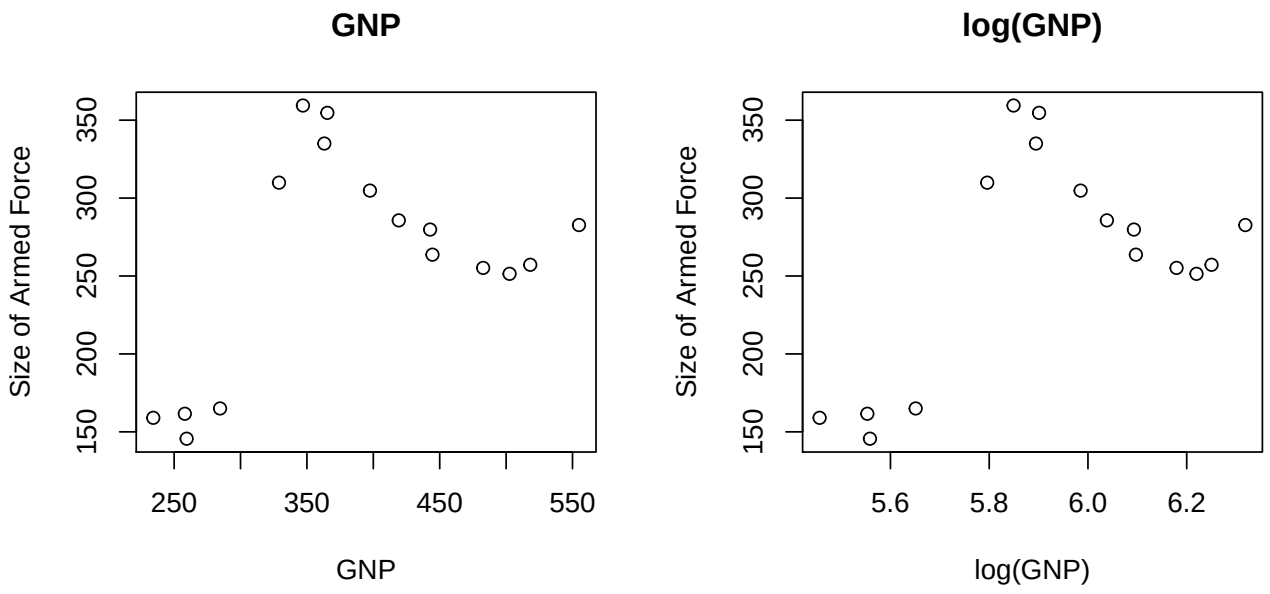

Figure 2: Size of Armed Force and GNP (default)

```
fig_cor2 <- ggplot(longley, aes(x = log(gnp), y = force)) + geom_point() +
  geom_smooth(method = "loess") + xlab("GNP") + ylab("Size of Armed Force") +
  ggtitle("log(GNP)")
plot_grid(fig_cor1, fig_cor2, ncol = 2)
```

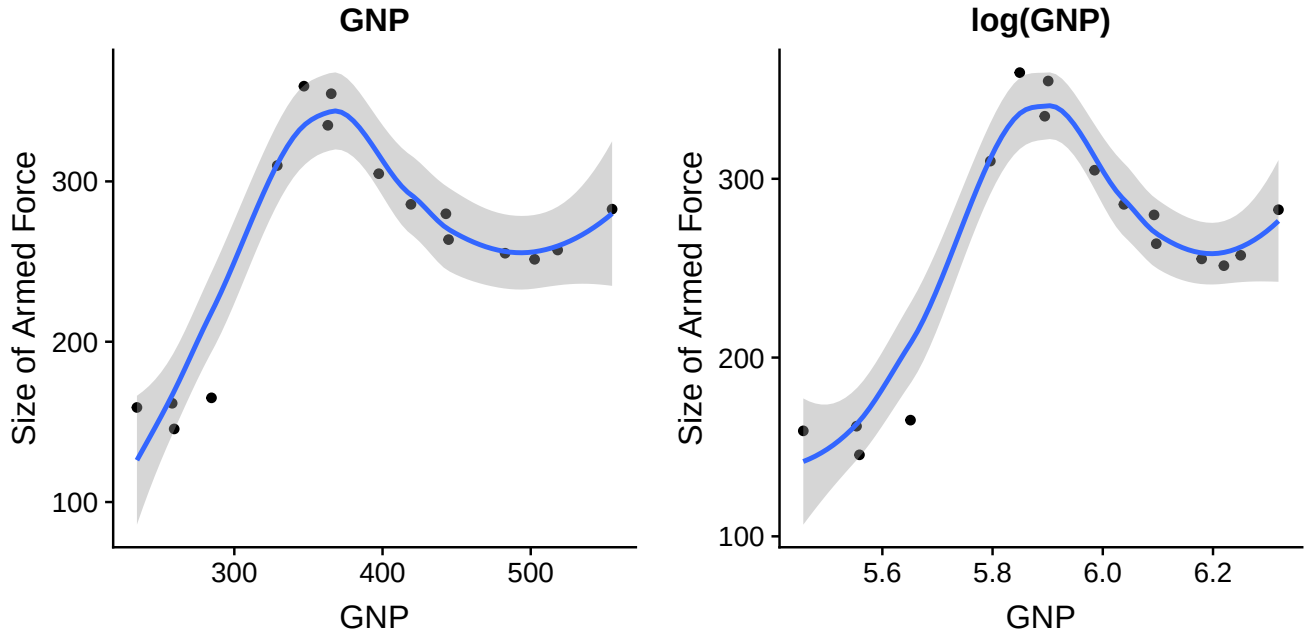

Figure 3: Size of Armed Force and GNP (ggplot)

Models

Clearly state your model and the assumption of the model.

(Alignment Style 1:)

$$\text{Model 1:} \quad \text{Armed Force}_i = \beta_0 + \beta_1 \text{Unemployment}_i + \beta_2 \text{GNP}_i + \epsilon_i$$
$$\text{Model 2:} \quad \text{Armed Force}_i = \beta_0 + \beta_1 \text{Unemployment}_i + \beta_2 \text{GNP}_i + \beta_3 \text{Population}_i + \epsilon_i$$
$$\text{Model 3:} \quad \text{Armed Force}_i = \beta_0 + \beta_1 \text{Unemployment}_i + \beta_2 \text{GNP}_i + \beta_3 \text{GNP}_i^2 + \beta_4 \text{Population}_i + \epsilon_i$$
$$\text{Model 4:} \quad \text{Armed Force}_i = \beta_0 + \beta_1 \text{Unemployment}_i + \beta_2 \text{GNP}_i + \beta_3 \text{GNP}_i^2 + \beta_4 \text{Population}_i + \beta_5 \text{Year}_i + \epsilon_i$$
$$\text{For all models, I assume } \epsilon \sim N(0, \sigma^2)$$

(Alignment Style 2:)

$$\begin{aligned}
\text{Model 1: Armed Force}_i &= \beta_0 + \beta_1 \text{Unemployment}_i + \beta_2 \text{GNP}_i + \epsilon_i \\
\text{Model 2: Armed Force}_i &= \beta_0 + \beta_1 \text{Unemployment}_i + \beta_2 \text{GNP}_i + \beta_3 \text{Population}_i + \epsilon_i \\
\text{Model 3: Armed Force}_i &= \beta_0 + \beta_1 \text{Unemployment}_i + \beta_2 \text{GNP}_i + \beta_3 \text{GNP}_i^2 + \beta_4 \text{Population}_i + \epsilon_i \\
\text{Model 4: Armed Force}_i &= \beta_0 + \beta_1 \text{Unemployment}_i + \beta_2 \text{GNP}_i + \beta_3 \text{GNP}_i^2 + \beta_4 \text{Population}_i + \beta_5 \text{Year}_i + \epsilon_i
\end{aligned}$$
$$\text{For all models, I assume } \epsilon \sim N(0, \sigma^2)$$

```r
#---------------------------
# Fit models
#---------------------------
  # Tips: store a group of model in a list
  # Benefits: convenient management!
  fit_models <- function(d){
    m <- list()
    m[["Baseline"]] <- glm(force ~ unemp + gnp, data = d, family = gaussian)
    m[["Population"]] <- glm(force ~ unemp + gnp + pop, data = d, family = gaussian)
    m[["Quad Population"]] <- glm(force ~ unemp + gnp + I(gnp^2) + pop, data = d,
                                  family = gaussian)
    m[["Year"]] <- glm(force ~ unemp + gnp + I(gnp^2) + pop + yr, data = d,
                       family = gaussian)
    m
  }

  m <- fit_models(longley)
```

Results (Tables)

Table 3 reports all models with no labels. Table 4 reports part of the models. Table 5 label the variables, reset the number of digits to report etc.

```
# Stargazer Quick Reference: https://www.jakeruss.com/cheatsheets/stargazer/
# Alternative: xtable. More flexible, but harder to code.
    # https://cran.r-project.org/web/packages/xtable/vignettes/xtableGallery.pdf

#-------------------------------------
# Show regression results with tables
#-------------------------------------
  # Print all models
    stargazer(m, label = "tab:arm1",
              title =
                "(All Models) Economic Determinants of the Size of Armed Force",
              header = FALSE, type = out_type)
```

Table 3: (All Models) Economic Determinants of the Size of Armed Force

	Dependent variable:			
	force			
	(1)	(2)	(3)	(4)
unemp	−0.525**	−0.227	−0.825***	−0.398
	(0.181)	(0.317)	(0.255)	(0.428)
gnp	0.611***	2.448	2.101*	7.317
	(0.170)	(1.628)	(1.075)	(4.377)
I(gnp^2)			−0.007***	−0.010***
			(0.002)	(0.003)
pop		−28.928	59.152*	79.852**
		(25.485)	(27.333)	(31.599)
yr				−93.220
				(75.935)
Constant	191.458***	2,780.931	−6,123.924**	171,987.500
	(56.948)	(2,281.964)	(2,648.690)	(145,108.600)
Observations	16	16	16	16
Log Likelihood	−85.292	−84.476	−77.097	−75.974
Akaike Inf. Crit.	176.584	176.952	164.193	163.947
Note:				*p<0.1; **p<0.05; ***p<0.01

```
    # If you input a list of models, it will report them all in one table.
    # Remember to add label and title to your table.
    # A table of ambiguous meaning is not worth reporting

  # Print a subset of models
    stargazer(m[["Baseline"]], m[["Population"]], label = "tab:arm2",
              title =
                "(Baseline and Population) Economic Determinants of the Size of Armed Force",
```

```
          header = FALSE, type = out_type)
```

Table 4: (Baseline and Population) Economic Determinants of the Size of Armed Force

	Dependent variable:	
	force	
	(1)	(2)
unemp	−0.525**	−0.227
	(0.181)	(0.317)
gnp	0.611***	2.448
	(0.170)	(1.628)
pop		−28.928
		(25.485)
Constant	191.458***	2,780.931
	(56.948)	(2,281.964)
Observations	16	16
Log Likelihood	−85.292	−84.476
Akaike Inf. Crit.	176.584	176.952
Note:	*p<0.1; **p<0.05; ***p<0.01	

```
# Label your Table (Essential!!!)
  stargazer(m, label = "tab:arm3",
            title = "(Labeled) Economic Determinants of the Size of Armed Force",
            covariate.labels = c("Unemployment", "GNP",
                                 "GNP sq", "Population", "Year"),
              # Mind the order... Better Strategy is asigning meaningful var names
              # in the dataset. Will end up saving your time!
            dep.var.labels = "Size of Armed Force",
            digits = 2,
            ci = TRUE,
            star.cutoffs = NA, # don't show stars
            notes = "Source of Data: Longley (1967)",
            font.size = "footnotesize", # Font size
            header = FALSE, type = out_type
            )
```

Table 5: (Labeled) Economic Determinants of the Size of Armed Force

	Dependent variable:			
	Size of Armed Force			
	(1)	(2)	(3)	(4)
Unemployment	−0.52	−0.23	−0.82	−0.40
	(−0.88, −0.17)	(−0.85, 0.39)	(−1.32, −0.32)	(−1.24, 0.44)
GNP	0.61	2.45	2.10	7.32
	(0.28, 0.94)	(−0.74, 5.64)	(−0.01, 4.21)	(−1.26, 15.89)
GNP sq			−0.01	−0.01
			(−0.01, −0.004)	(−0.02, −0.004)
Population		−28.93	59.15	79.85
		(−78.88, 21.02)	(5.58, 112.72)	(17.92, 141.79)
Year				−93.22
				(−242.05, 55.61)
Constant	191.46	2,780.93	−6,123.92	171,987.50
	(79.84, 303.07)	(−1,691.64, 7,253.50)	(−11,315.26, −932.59)	(−112,420.10, 456,395.10)
Observations	16	16	16	16
Log Likelihood	−85.29	−84.48	−77.10	−75.97
Akaike Inf. Crit.	176.58	176.95	164.19	163.95

Note: NA

Source of Data: Longley (1967)

Regression Results (Graphs)

This part is muted by default because of the long time it takes to install the package.

Figure **??** and **??** plots the coefficients with unscaled predictors. Figure **??** and **??** plot the coefficients with scaled predictors. The interpretation of the coefficients are different after scaling. Before scaling: "one unit change of `variable x` is associated with `coef` units change of the dependent variables". After scaling: "one standard deviation change of `variable x` is associated with `coef` units change of the dependent variables".

In addition, Figure **??** shows how the dependent variable changes in response to all four independent variables in our `Year` Model.

ALL THESE FIGURES ARE TERRIBLE, BECAUSE THEY ARE NOT CORRECTLY LABELED!!
So I would still recommend using `ggplot` for your visualization, where you enjoy higher flexibility.

```
# Show the code in the appdx, but do not run them again.
knitr::opts_chunk$set(eval = FALSE)
```

```
# Experimenting a good visualization package.
# Takes a long time to install. And there's been error report
# Save your other R files. When you are ready to try, set the above
# eval = TRUE
usePackage("sjPlot")

#----------------------------
# Visualize Regression Results
#----------------------------
  # sjPlot: http://www.strengejacke.de/sjPlot/;
  #         https://cran.r-project.org/web/packages/sjPlot/vignettes/sjpglm.html
  #         https://github.com/strengejacke/sjPlot
  # Pros: Can work with models generated by different packages
  #       lmer, glm, glmm, glmer, etc.
  #       it's plot appears much better than many default plot functions.
  # Cons: Not flexible compared to ggplot. Package under development. Use with caution.

  #------------------------------
  # Coefficients (original data)
  #------------------------------
    # Plotting one model
    sjPlot::plot_model(m[["Year"]],
                       title = "Economic Determinants of Armed Force Size I")
```

```
    # Comparing multiple models
    sjPlot::plot_models(m, title = "Economic Determinants of Armed Force Size II")
    # Note. Scale problem --> can't really see CI of most variables.

    # Solution: scale independentvariables before input. BUT DON'T SCALE THE DVs!
```

```
  #-----------------------------------------------
  # Coefficients (scaled Independent variables)
  #-----------------------------------------------
    longley_s <- longley %>% mutate(gnp = scale(gnp), unemp = scale(unemp),
                                    pop = scale(pop), yr = scale(yr))
    m <- NULL
    m <- fit_models(longley_s)
    # Plotting one model
    sjPlot::plot_model(m[["Year"]],
                       title = "Economic Determinants of Armed Force Size I (Scaled)")
```

```
    # Comparing multiple models
    sjPlot::plot_models(m, title = "Economic Determinants of Armed Force Size II (Scaled)")
    # Something is wrong with this command. Still need a fix.
```

```r
#-----------------------------------------------
# Other Plots
#-----------------------------------------------
  sjPlot::plot_model(m[["Year"]], type = "slope")
    # Again. package under developement.
    # Don't trust it too much. Esp. the Variance.
    # Good for perliminary analysis
```

Discussion

All results are summarized in Table 5... bla bla bla

Citation

Bla bla bla [Johnston et al., 2014].

Beramendi and Anderson [2008, p.234] argue that...

Existing studies find evidence that bla bla bla [see Stegmueller, 2013, Bell and Jones, 2015, for detailed explanation]...

References

Andrew Bell and Kelvyn Jones. Explaining fixed effects: Random effects modeling of time-series cross-sectional and panel data. *Political Science Research and Methods*, 3(1):133–153, 2015.

Pablo Beramendi and Christopher J Anderson. *Democracy, Inequality, and Representation in Comparative Perspective*. Russell Sage Foundation, 2008.

Christopher D Johnston, D Sunshine Hillygus, and Brandon L Bartels. Ideology, the affordable care act ruling, and supreme court legitimacy. *Public Opinion Quarterly*, 78(4):963–973, 2014.

Daniel Stegmueller. How many countries for multilevel modeling? a comparison of frequentist and bayesian approaches. *American Journal of Political Science*, 57(3):748–761, 2013.

Others (Analytical Graphs, Game Trees...)

`Rmarkdown` allows you to use all LaTex packages (put `header_includes: \usepackage{} in in the header at the start of the document`). For example, you can plot analytical graphs (functions, game trees etc.) with the `TikZ` packages. See more examples here:
`http://www.sfu.ca/~haiyunc/notes/Game_Trees_with_TikZ.pdf`;
`https://sites.google.com/site/kochiuyu/Tikz`.

Appendix (Code)

For readability, you may suppress your code within your text, and put them all into the appendix. You can re-use a chunk of code by calling `ref.label=(chunck_name)`. When you reuse a chunk, you may want to avoid running again by setting `eval=FALSE`. Again, you can set these up as a global option with the `knitr::opts_chunck` command.

```
# Show the code in the appdx, but do not run them again.
knitr::opts_chunk$set(echo = TRUE, eval = FALSE)
```

Loading the Data

```
#-----------------------------
# load your data
#-----------------------------
  # Load your dataset of interest.
  # Below is an example economic dataset coming with R
    data("longley")
      # J. W. Longley (1967) An appraisal of least-squares programs from
      # the point of view of the user.
      # Journal of the American Statistical Association 62, 819-841.
  # Just to mess up the dataset by a bit
    names(longley) <- c("gnp.def", "gnp", "unemp", "force", "pop", "yr", "emp")
```

Generating a Correlation Matrix

```
#----------------------------
# Correlation Matrix
#----------------------------
  PerformanceAnalytics::chart.Correlation(longley)
  # By far my favorite, better than other fancy stuff.
  # Perfect for continuous variables
```

Fitting Models

```
#---------------------------
# Fit models
#---------------------------
  # Tips: store a group of model in a list
  # Benefits: convenient management!
  fit_models <- function(d){
    m <- list()
    m[["Baseline"]] <- glm(force ~ unemp + gnp, data = d, family = gaussian)
    m[["Population"]] <- glm(force ~ unemp + gnp + pop, data = d, family = gaussian)
    m[["Quad Population"]] <- glm(force ~ unemp + gnp + I(gnp^2) + pop, data = d,
                                  family = gaussian)
```

```
    m[["Year"]] <- glm(force ~ unemp + gnp + I(gnp^2) + pop + yr, data = d,
                       family = gaussian)
    m
  }

  m <- fit_models(longley)
```

Presenting Results in Tables

```
# Stargazer Quick Reference: https://www.jakeruss.com/cheatsheets/stargazer/
# Alternative: xtable. More flexible, but harder to code.
    # https://cran.r-project.org/web/packages/xtable/vignettes/xtableGallery.pdf

#-------------------------------------
# Show regression results with tables
#-------------------------------------
  # Print all models
    stargazer(m, label = "tab:arm1",
              title =
                "(All Models) Economic Determinants of the Size of Armed Force",
              header = FALSE, type = out_type)
      # If you input a list of models, it will report them all in one table.
      # Remember to add label and title to your table.
      # A table of ambiguous meaning is not worth reporting

  # Print a subset of models
    stargazer(m[["Baseline"]], m[["Population"]], label = "tab:arm2",
              title =
                "(Baseline and Population) Economic Determinants of the Size of Armed Force",
              header = FALSE, type = out_type)

  # Label your Table (Essential!!!)
    stargazer(m, label = "tab:arm3",
              title = "(Labeled) Economic Determinants of the Size of Armed Force",
              covariate.labels = c("Unemployment", "GNP",
                                   "GNP sq", "Population", "Year"),
                # Mind the order... Better Strategy is asigning meaningful var names
                # in the dataset. Will end up saving your time!
              dep.var.labels = "Size of Armed Force",
              digits = 2,
              ci = TRUE,
              star.cutoffs = NA, # don't show stars
              notes = "Source of Data: Longley (1967)",
              font.size = "footnotesize", # Font size
              header = FALSE, type = out_type
              )
```