

```
rm(list = ls())
tinytex::install_tinytex()
#install.packages("stargazer")
library(stargazer)
```

```
## Warning: package 'stargazer' was built under R version 4.1.2
```

```
##
## Please cite as:
```

```
## Hlavac, Marek (2022). stargazer: Well-Formatted Regression and Summary Statistics Tables.
```

```
## R package version 5.2.3. https://CRAN.R-project.org/package=stargazer
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.3
```

```
## -- Attaching packages ----- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5      v purrr  0.3.4
## v tibble  3.1.6      v dplyr  1.0.8
## v tidyr   1.2.0      v stringr 1.4.0
## v readr   2.1.2      v forcats 0.5.1
```

```
## Warning: package 'tibble' was built under R version 4.1.3
```

```
## Warning: package 'tidyr' was built under R version 4.1.3
```

```
## Warning: package 'readr' was built under R version 4.1.3
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
## -- Conflicts ----- tidyverse_conflicts() --
```

```
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
dataset <- read.csv("./data/dataset.csv")
```

```
test_data <- dataset %>%
  select(E.Government.Index, ps_2020, FDI, urban, young_population, pop_dens, GDP_per,
         corruption, rule_of_law, accountability, effectiveness, reg_qual) %>%
  mutate(GDP_per = log(GDP_per))
```

```
coef_p_table <- array(data = NA, dim = c(ncol(test_data)-2, 4),
  dimnames = list(colnames(test_data)[3:ncol(test_data)],
    c("coeff for E-GOV", "P for EGOV", "coeff for PS", "P for PS")))
```

```
i = 1
for (i in 1: ncol(test_data)-2){
```

```

formula_1 <- E.Government.Index ~ test_data[,i + 2]
formula_2 <- ps_2020 ~ test_data[,i + 2]

temp_1 <- lm(formula_1, data = test_data)
temp_2 <- lm(formula_2, data = test_data)

coef_p_table[i,1:2] <- round(summary(temp_1)[["coefficients"]][2,c(1,4)],3)
coef_p_table[i,3:4] <- round(summary(temp_2)[["coefficients"]][2,c(1,4)],3)
}

```

```

## Warning in summary.lm(temp_1): essentially perfect fit: summary may be
## unreliable

```

```

## Warning in summary.lm(temp_2): essentially perfect fit: summary may be
## unreliable

```

```
coef_p_table
```

	coeff for E-GOV P for EGOV	coeff for PS P for PS
## FDI	0.154	0.000
## urban	0.183	0.000
## young_population	0.022	0.000
## pop_dens	0.000	0.193
## GDP_per	0.410	0.000
## corruption	0.120	0.000
## rule_of_law	0.184	0.000
## accountability	0.005	0.000
## effectiveness	0.123	0.000
## reg_qual	0.000	0.000

```
cor(test_data,use="complete.obs")
```

	E.Government.Index	ps_2020	FDI	urban
## E.Government.Index	1.0000000	0.6038815	0.7157513	0.8416965
## ps_2020	0.6038815	1.0000000	0.5437639	0.6064037
## FDI	0.7157513	0.5437639	1.0000000	0.9184420
## urban	0.8416965	0.6064037	0.9184420	1.0000000
## young_population	0.6417140	0.2660908	0.3598705	0.4959605
## pop_dens	0.1175506	0.1243187	0.1837093	0.2270246
## GDP_per	0.6199253	0.3358956	0.4396052	0.5305982
## corruption	0.6046215	0.4625480	0.7901466	0.7565814
## rule_of_law	0.8258633	0.5878983	0.8893027	0.9523239
## accountability	0.6424275	0.3371311	0.5155426	0.5624266
## effectiveness	0.6163238	0.5617877	0.7841821	0.7336525
## reg_qual	0.6520410	0.4529591	0.7989212	0.8014344
##	young_population	pop_dens	GDP_per	corruption
## E.Government.Index	0.6417140	0.1175506	0.6199252	0.6046215
## ps_2020	0.2660908	0.1243187	0.3358956	0.4625480
## FDI	0.3598705	0.1837093	0.4396052	0.7901466
## urban	0.4959605	0.2270246	0.5305982	0.7565814

```
## young_population      1.0000000  0.22821501 0.49396182  0.4002779
## pop_dens              0.2282150  1.00000000 0.09283174  0.1722967
## GDP_per               0.4939618  0.09283174 1.00000000  0.4136752
## corruption            0.4002779  0.17229667 0.41367517  1.0000000
## rule_of_law           0.4397863  0.21226005 0.48826261  0.7365798
## accountability       0.4647149  0.13075057 0.38754600  0.4035219
## effectiveness        0.2016453 -0.01124569 0.38969421  0.7470431
## reg_qual              0.3173730  0.19959219 0.30393257  0.5921368
##               rule_of_law accountability effectiveness reg_qual
## E.Government.Index  0.8258633      0.6424275      0.61632377 0.6520410
## ps_2020             0.5878983      0.3371311      0.56178769 0.4529591
## FDI                 0.8893027      0.5155426      0.78418208 0.7989212
## urban              0.9523239      0.5624266      0.73365250 0.8014344
## young_population    0.4397863      0.4647149      0.20164530 0.3173730
## pop_dens            0.2122600      0.1307506     -0.01124569 0.1995922
## GDP_per            0.4882626      0.3875460      0.38969421 0.3039326
## corruption          0.7365798      0.4035219      0.74704311 0.5921368
## rule_of_law         1.0000000      0.5633732      0.76457172 0.7918097
## accountability     0.5633732      1.0000000      0.41799853 0.5510825
## effectiveness      0.7645717      0.4179985      1.00000000 0.6236953
## reg_qual            0.7918097      0.5510825      0.62369531 1.0000000
```

```
formula_1 <- E.Government.Index ~ ps_2020
formula_2 <- E.Government.Index ~ ps_2020 + log(GDP_per)
formula_3 <- E.Government.Index ~ ps_2020 + log(GDP_per) + urban + young_population

reg_1 <- lm(formula_1, data = dataset)
reg_2 <- lm(formula_2, data = dataset)
reg_3 <- lm(formula_3, data = dataset)
```

$$E_Gov_i = \alpha + \beta * party_strength_i + \sum \beta_j * Cov_i + \epsilon_i$$

% Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac at gmail.com % Date and time: Fri, May 13, 2022 - 3:15:38 PM

Table 1:

	<i>Dependent variable:</i>		
	E.Government.Index		
	(1)	(2)	(3)
Party Strength	0.169*** (0.025)	0.115*** (0.021)	0.051*** (0.015)
GDP per capita(log)		0.377*** (0.041)	0.119*** (0.033)
Urban Population Ratio			0.112*** (0.012)
Young Population Ratio			0.009*** (0.002)
Constant	0.604*** (0.015)	-1.121*** (0.192)	-0.496*** (0.155)
Observations	170	135	135
R ²	0.209	0.513	0.801
Adjusted R ²	0.205	0.506	0.795
Residual Std. Error	0.200 (df = 168)	0.140 (df = 132)	0.090 (df = 130)
F Statistic	44.501*** (df = 1; 168)	69.520*** (df = 2; 132)	130.660*** (df = 4; 130)

Note:

*p<0.1; **p<0.05; ***p<0.01