Stock and Watson Chapter 8: Replication

Econ 440 - Introduction to Econometrics

Patrick Toche, ptoche@fullerton.edu

17 April 2022

Replication

Replicate regression results from James H. Stock and Mark W. Watson, Introduction to econometrics, Pearson, 4th Edition, Chapter 8. The data used is available in the Stata format $caschool.dta$ and in the Excel format $caschool.xlsx$.

Load dataset

Let's load the Stata dataset using the haven library.

```
library(haven)
df <- read_dta("caschool.dta")
head(df)
```

```
## # A tibble: 6 x 18
##   observat dist_cod county  district gr_span enrl_tot teachers calw_pct meal_pct
##      <dbl>    <dbl> <chr>   <chr>    <chr>      <dbl>    <dbl>    <dbl>    <dbl>
## 1        1    75119 Alameda Sunol G~ KK-08        195    10.9     0.510     2.04
## 2        2    61499 Butte   Manzani~ KK-08        240    11.1    15.4      47.9
## 3        3    61549 Butte   Thermal~ KK-08       1550    82.9    55.0      76.3
## 4        4    61457 Butte   Golden ~ KK-08        243    14      36.5      77.0
## 5        5    61523 Butte   Palermo~ KK-08       1335    71.5    33.1      78.4
## 6        6    62042 Fresno  Burrel ~ KK-08        137     6.40   12.3      87.0
## # ... with 9 more variables: computer <dbl>, testscr <dbl>, comp_stu <dbl>,
## #   expn_stu <dbl>, str <dbl>, avginc <dbl>, el_pct <dbl>, read_scr <dbl>,
## #   math_scr <dbl>
```

You can also use the Excel file of course

```
library(readxl)
df <- read_xlsx("caschool.xlsx", trim_ws=TRUE)
```

Data cleaning

The variable names are inconsistent with the textbook and other versions of the dataset used in other exercises, so let's rename them:

```
names(df)[names(df) == "avginc"] <- "income"
names(df)[names(df) == "testscr"] <- "testscore"
```

Linear Regression of Test Score on District Income

The simple linear regression.

```
m1 <- lm(testscore ~ income, data=df)
summary(m1)
```

```
##
## Call:
## lm(formula = testscore ~ income, data = df)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -39.57  -8.80   0.60   9.03  32.53
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 625.3836     1.5324   408.1   <2e-16 ***
## income        1.8785     0.0905    20.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.4 on 418 degrees of freedom
## Multiple R-squared:  0.508,  Adjusted R-squared:  0.506
## F-statistic:  431 on 1 and 418 DF,  p-value: <2e-16
```

Tidy up with broom

Once we have estimated a model, it is convenient to use the `broom` library to tidy things up:

```
library(broom)
tidy(m1)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   625.      1.53       408.  0
## 2 income          1.88    0.0905      20.8 2.75e-66
```

Nonlinear Regression of Test Score on District Income

The quadratic regression.

To run a nonlinear regression on var, we can use the $I()$ wrapper:

```
lm(testscore ~ income + I(income^2), data=df)
```

or use the more versatile $poly(var)$:

```
m2 <- lm(testscore ~ poly(income,2,raw=TRUE), data=df)
tidy(m2)
```

```
## # A tibble: 3 x 5
##   term                          estimate std.error statistic  p.value
##   <chr>                            <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)                     607.       3.05       199.   0
## 2 poly(income, 2, raw = TRUE)1    3.85       0.304       12.7  2.69e-31
## 3 poly(income, 2, raw = TRUE)2   -0.0423     0.00626     -6.76 4.71e-11
```

but note that we set the raw=TRUE option: "if true, use raw and not orthogonal polynomials.''

Nonlinear Regression of Test Score on District Income

The cubic regression.

```
m3 <- lm(testscore ~ poly(income,3,raw=TRUE), data=df)
tidy(m3)
```

```
## # A tibble: 4 x 5
##   term                          estimate std.error statistic   p.value
##   <chr>                            <dbl>     <dbl>     <dbl>     <dbl>
## 1 (Intercept)                   600.        5.83        103.   4.61e-298
## 2 poly(income, 3, raw = TRUE)1    5.02      0.859         5.84 1.06e-  8
## 3 poly(income, 3, raw = TRUE)2   -0.0958    0.0374       -2.56 1.07e-  2
## 4 poly(income, 3, raw = TRUE)3    0.000685  0.000472      1.45 1.47e-  1
```

Linear-Log Regression

The linear-log regression.

```
m4 <- lm(testscore ~ log(income), data=df)
tidy(m4)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)    558.       4.20     133.  0
## 2 log(income)     36.4      1.57      23.2 4.77e-77
```

Log-Linear Regression

The linear-log regression.

```
m5 <- lm(log(testscore) ~ income, data=df)
tidy(m5)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)  6.44     0.00236     2724.  0
## 2 income       0.00284  0.000140      20.4 1.41e-64
```

Log-Log Regression

The log-log regression.

```
m6 <- lm(log(testscore) ~ log(income), data=df)
tidy(m6)
```

```
## # A tibble: 2 x 5
##   term        estimate std.error statistic  p.value
##   <chr>          <dbl>     <dbl>     <dbl>    <dbl>
## 1 (Intercept)   6.34     0.00645     982.  0
## 2 log(income)   0.0554   0.00241      23.0 4.52e-76
```