Empirical Exercise 1, Stock and Watson Chapter 2

Econ 440 - Introduction to Econometrics

Patrick Toche, ptoche@fullerton.edu

21 March 2022

Remove empty rows and columns: simple and symmetric!

```
df1 <- df1[, colMeans(is.na(df1)) != 1]
df1 <- df1[rowMeans(is.na(df1)) != 1,]
```

We saw other ways to do this using the `Filter()` and `complete.cases()` functions.

Reshape data from wide to long form:

```
library("tidyr")
df2 <- gather(df1, Age, Probability, -AHE)
```

Make Age a numeric or integer value:

```
df2$Age <- as.integer(df2$Age)
```

This is needed after conversion from wide (horizontal) to long (vertical), since the variable `Age` in dataframe `df2` is constructed from the column names in dataframe `df1`. And `colnames` are `strings` (aka `character vectors` in `R`).

Now let's look at our data

```
head(df2)
```

```
## # A tibble: 6 x 3
##     AHE   Age Probability
##   <dbl> <int>       <dbl>
## 1     5    25     0.00298
## 2     6    25     0.00116
## 3     7    25     0.00247
## 4     8    25     0.00240
## 5     9    25     0.00356
## 6    10    25     0.00516
```

(a) Compute the marginal distribution of Age.

This is a situation where data in wide format is convenient! Often data is more convenient in long format.

Sum by column: The marginal distribution of Age

```
colSums(df1[,names(df1) != "AHE"])
```

```
##       25       26       27       28       29       30       31       32
## 0.084890 0.092231 0.085471 0.093393 0.103496 0.104731 0.103932 0.108075
##       33       34
## 0.108802 0.114979
```

Sum by row:

```
rowSums(df1[,names(df1) != "AHE"])
```

```
##  [1] 0.0170797 0.0119195 0.0220219 0.0211498 0.0278363 0.0457155 0.0347409
##  [8] 0.0698452 0.0374300 0.0630133 0.0404099 0.0324878 0.0566902 0.0284178
## [15] 0.0590886 0.0359038 0.0523294 0.0676648 0.0364852 0.0415001 0.0423723
## [22] 0.0486227 0.0312523 0.0214405 0.0177339 0.0085762 0.0097391 0.0044335
## [29] 0.0140999
```

If the data is in long format, we have to work a little harder. Below are several approaches.

With the data in long format:

```
sum(df2[df2$Age == 25,]$Probability)
```

```
## [1] 0.08489
```

```
sum(df2[df2$Age == 26,]$Probability)
```

```
## [1] 0.092231
```

```
sum(df2[df2$Age == 27,]$Probability)
```

```
## [1] 0.085471
```

with a split/apply routine to group by Age

```
sapply(split(df2, df2$Age), function(x) sum(x$Probability))
```

```
##       25       26       27       28       29       30       31       32
## 0.084890 0.092231 0.085471 0.093393 0.103496 0.104731 0.103932 0.108075
##       33       34
## 0.108802 0.114979
```

To see how this works, try this command: `split(df2, df2$Age)`. It returns a list of dataframes. Then `sapply` applies the function we defined as `function(x) sum(x$Probability)` to each dataframe.

or we can always write a loop

```
for (age in 25:34){
    print(sum(df2[df2$Age == age,]$Probability))
}
```

```
## [1] 0.08489
## [1] 0.092231
## [1] 0.085471
## [1] 0.093393
```

```
## [1] 0.1035
## [1] 0.10473
## [1] 0.10393
## [1] 0.10807
## [1] 0.1088
## [1] 0.11498
```

(b) Compute the mean of AHE for each value of Age

Two ways we could do it. There are still other ways we saw in class.

(i) with split/apply:

```
sapply(split(df2, df2$Age), function(x) weighted.mean(x$AHE, x$Probability))
```

```
##     25     26     27     28     29     30     31     32     33     34
## 17.591 18.967 19.705 20.236 21.171 21.785 22.595 23.692 23.349 24.108
```

(ii) with dplyr:

It is convenient to compute the probabilities for each Age and store them together, which is easily done with dplyr:

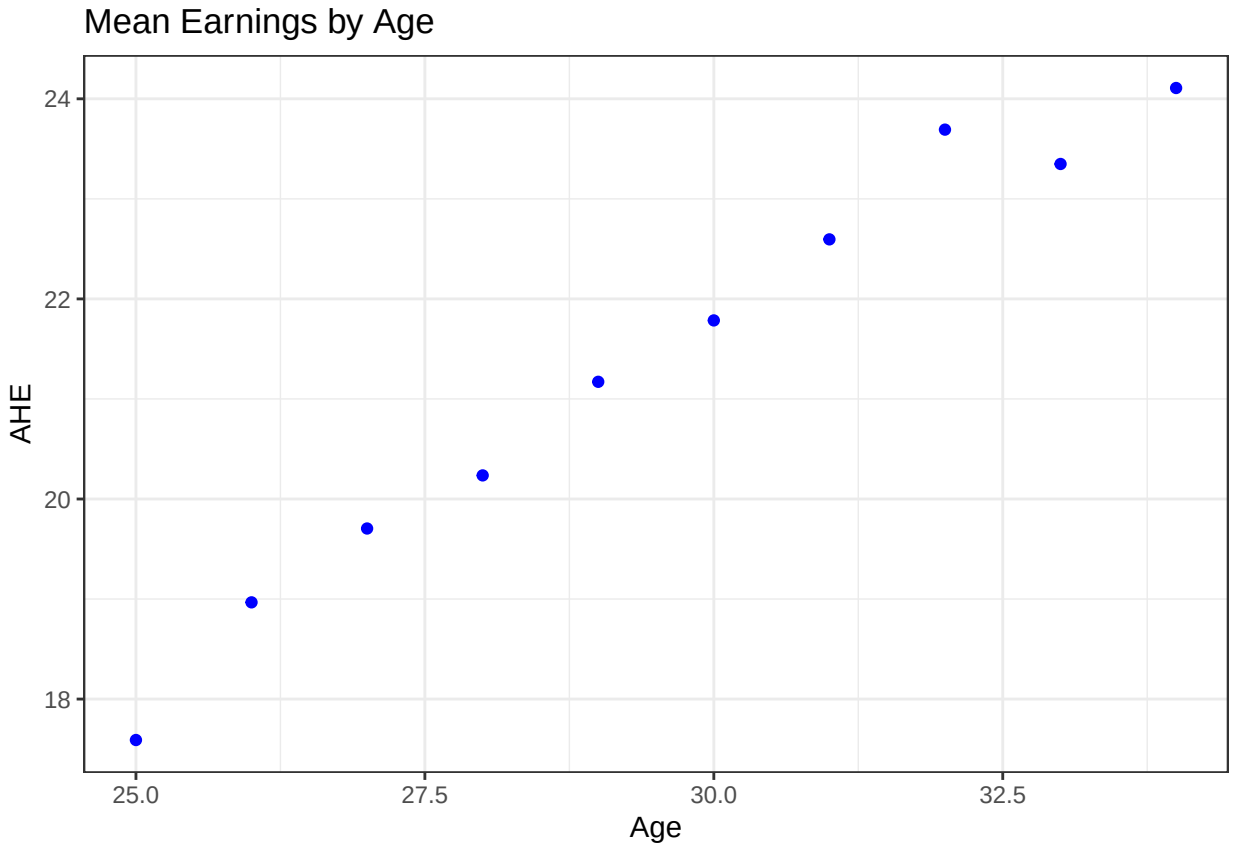
```
df2 %>%
    group_by(Age) %>%
    summarise(AHE = weighted.mean(AHE, Probability),
              Probability = sum(Probability)) -> dm
dm
```

```
## # A tibble: 10 x 3
##      Age   AHE Probability
##    <int> <dbl>       <dbl>
##  1    25  17.6      0.0849
##  2    26  19.0      0.0922
##  3    27  19.7      0.0855
##  4    28  20.2      0.0934
##  5    29  21.2      0.103
##  6    30  21.8      0.105
##  7    31  22.6      0.104
##  8    32  23.7      0.108
##  9    33  23.3      0.109
## 10    34  24.1      0.115
```

(c) Compute and plot the mean of AHE versus Age. Are average hourly earnings and age related? Explain.

Plot of mean hourly income by Age:

```
library("ggplot2")
library("scales")
df <- dm[c("Age", "AHE")]
ggplot(data=df, aes(x=Age, y=AHE)) +
    geom_point(color="blue") +
    ggtitle("Mean Earnings by Age") +
    theme_bw()
```



The correlation between AHE and Age is visible from the scatter plot. An explanation is that wages rise with experience and experience rises with age, in most professions and at most of the range of ages and earnings (there may be exceptions at the extreme bounds of the distribution of age and earnings).

(d) Use the law of iterated expectations to compute the mean of AHE

The mean of AHE for all ages:

```
sum(dm$AHE)
```

```
## [1] 213.2
```

Check that sum(dm$Probability)=1:

```
sum(dm$Probability)
```

```
## [1] 1
```

(e) Compute the variance of AHE.

Define a weighted variance function

One approach is to compute a weighted variance the same way we computed the weighted mean. This is done now. Another approach, shown later, is to exploit the relation:

$$V[x] = E[x^2] - (E[x])^2$$

There is no way to unbias it because the information about the sample size is not available. One simple definition of the weighted variance is the following:

```
weighted.variance <- function(x,w) sum(w*(x-weighted.mean(x,w))^2)/sum(w)
```

We can now use the `weighted.variance` function in the same way we used the `weighted.mean` function.

(i) with split/apply:

```
dv <- sapply(split(df2, df2$Age), function(x) weighted.variance(x$AHE, x$Probability))
# we could make a dataframe
dv <- data.frame(Age = names(dv), Variance = dv)
rownames(dv) <- NULL  # to keep things pretty
dv
```

```
##    Age Variance
## 1   25   95.935
## 2   26  129.049
## 3   27  128.790
## 4   28  142.663
## 5   29  152.159
## 6   30  162.795
## 7   31  169.672
## 8   32  208.637
## 9   33  189.950
## 10  34  197.115
```

(ii) with dplyr:

```
df2 %>%
    group_by(Age) %>%
    summarise(Variance = weighted.variance(AHE, Probability),
              Probability = sum(Probability)) -> dv
dv
```

```
## # A tibble: 10 x 3
##      Age Variance Probability
##    <int>    <dbl>       <dbl>
##  1    25     95.9      0.0849
##  2    26    129.       0.0922
##  3    27    129.       0.0855
##  4    28    143.       0.0934
##  5    29    152.       0.103
##  6    30    163.       0.105
##  7    31    170.       0.104
##  8    32    209.       0.108
##  9    33    190.       0.109
## 10    34    197.       0.115
```

variance for all ages:

```
# merge data for convenience
d1 = merge(dm, dv)
sum(d1$Probability * d1$Variance)/sum(d1$Probability)
```

```
## [1] 160.69
```

expected value of squared earnings

```
weighted.second.moment <- function(x,w) sum(w*x^2)/sum(w)
```

compute expected value of squared earnings by Age

$E[AHE^2|Age]$

(i) with split/apply:

```
sapply(split(df2, df2$Age), function(x) weighted.second.moment(x$AHE, x$Probability))
```

```
##     25     26     27     28     29     30     31     32     33     34
## 405.37 488.79 517.07 552.15 600.39 637.38 680.21 769.95 735.11 778.32
```

(ii) with dplyr:

```
df2 %>%
    group_by(Age) %>%
    summarise(AHE2 = weighted.second.moment(AHE, Probability),
              Probability = sum(Probability)) -> de
de
```

```
## # A tibble: 10 x 3
##      Age  AHE2 Probability
##    <int> <dbl>       <dbl>
##  1    25  405.      0.0849
##  2    26  489.      0.0922
##  3    27  517.      0.0855
##  4    28  552.      0.0934
##  5    29  600.      0.103
##  6    30  637.      0.105
##  7    31  680.      0.104
##  8    32  770.      0.108
##  9    33  735.      0.109
## 10    34  778.      0.115
```

```
# merge data for convenience
d1 <- merge(d1, de)
```

(f) Compute the covariance between AHE and Age.

The piping technique of the `dplyr` package is perhaps the most convenient approach. We could in fact do all the computations of this assignment in one series of pipes, as shown below.

(g) Compute the correlation between AHE and Age.

Now put it all together

```
df2 %>%
    summarise(Age1 = weighted.mean(Age, Probability),
              Age2 = weighted.mean(Age^2, Probability),
              AHE1 = weighted.mean(AHE, Probability),
              AHE2 = weighted.mean(AHE^2, Probability),
              AHEAge = weighted.mean(AHE * Age, Probability),
```

```
              AHEV = AHE2 - AHE1^2,
              AHES = sqrt(AHEV),
              AgeV = Age2 - (Age1)^2,
              AgeS = sqrt(AgeV),
              Cov = AHEAge - Age1*AHE1,
              Cor = Cov/AHES/AgeS) -> d2
d2
```

```
## # A tibble: 1 x 11
##    Age1  Age2  AHE1  AHE2 AHEAge  AHEV  AHES  AgeV  AgeS   Cov   Cor
##   <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  29.8  894.  21.5  628.   646.  165.  12.8  8.14  2.85  5.72 0.156
```

(h) Relate your answers in (f) and (g) to the plot you constructed in (c).

The evidence is clear: The correlation is significant. This can be shown formally for a test of the null of no correlation at the 5% significance level. Under the null, the correlation coefficient has a t-distribution with n-2 degrees of freedom:

```
alpha = 0.05
r = d2$Cor
n = ncol(d2)
t = r/sqrt((n-2)/(1-r^2))
tc = qt(1-alpha/2, n-2)
t < tc
```

```
## [1] TRUE
```