ECONOMETRICS IN R

LECTURE 4 INTRO TO R PROGRAMMING

MARIA MONTOYA-AGUIRRE

M1 APE @ PARIS SCHOOL OF ECONOMICS

REVIEW

AGENDA

- REGRESSIONS IN R
  - CONTINUOUS VARIABLES
  - CATEGORICAL AND BINARY VARIABLES
- GETTING TO THE RIGHT MODEL
  - VARIABLE TRANSFORMATION
  - FUNCTIONAL FORM
  - CONTROL VARIABLES
  - INTERACTION TERMS
- INFERENCE
- EXPORTING RESULTS
  - REGRESSION TABLES
  - COEFFICIENT PLOTS
- PRACTICING YOUR SKILLS

REGRESSIONS

We are going to work with a portion of the US Current Population Survey for August 2023. Download it from here. We are going to explore some well-known relationships in **labor economics**.

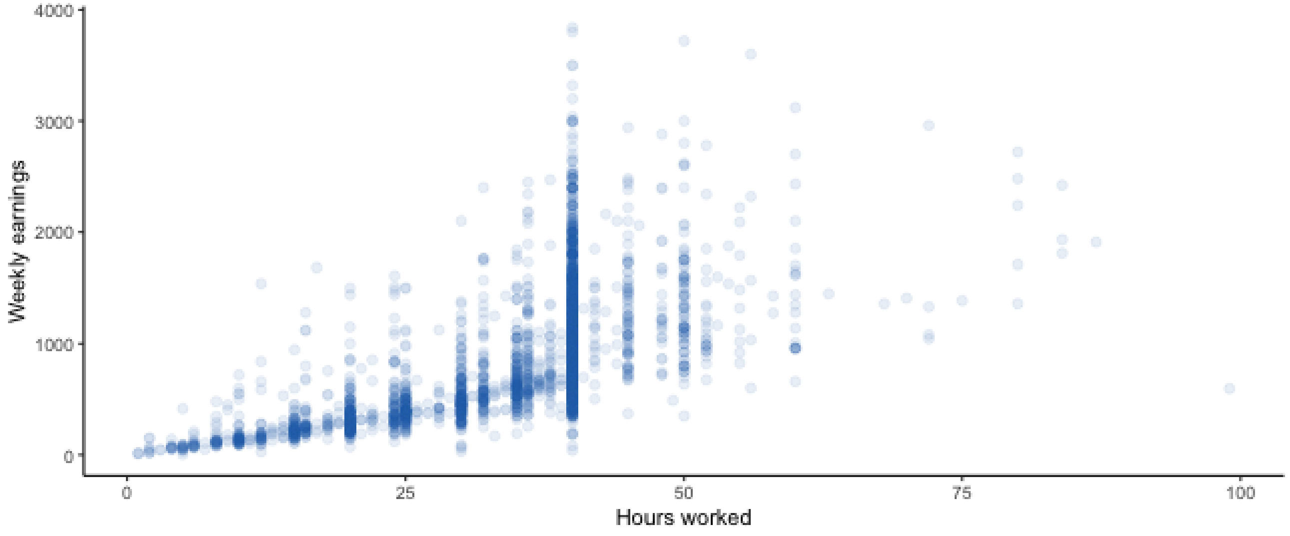
```
cps <- read.csv("../data/04_202308_cps.csv")
cps10 <- read.csv("../data/04_202308_cps10.csv")
str(cps)
```

```
## 'data.frame':    4609 obs. of  13 variables:
##  $ X        : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ region   : int  3 3 3 3 3 3 3 3 3 1 ...
##  $ state    : int  1 1 1 1 1 1 1 1 1 9 ...
##  $ age      : int  22 50 55 52 59 57 22 38 24 16 ...
##  $ sex      : chr  "Male" "Female" "Male" "Male" ...
##  $ maritl   : int  6 1 5 5 1 1 6 1 6 6 ...
##  $ educ     : chr  "High school" "Bachelor's degree" "No high school" "High school" ...
##  $ race     : int  1 1 1 1 1 1 1 2 2 1 ...
##  $ status   : int  1 1 2 1 1 1 1 1 1 1 ...
##  $ work     : int  1 1 2 1 1 1 1 1 1 1 ...
##  $ hours    : int  40 30 40 40 40 40 20 30 40 12 ...
##  $ hourrt   : num  14 42 20 27.6 40 ...
##  $ earnings: num  560 1260 800 1103 1600 ...
```

REGRESSIONS

Let's take a look at the relation between hours worked and earnings (measured for a weekly period).

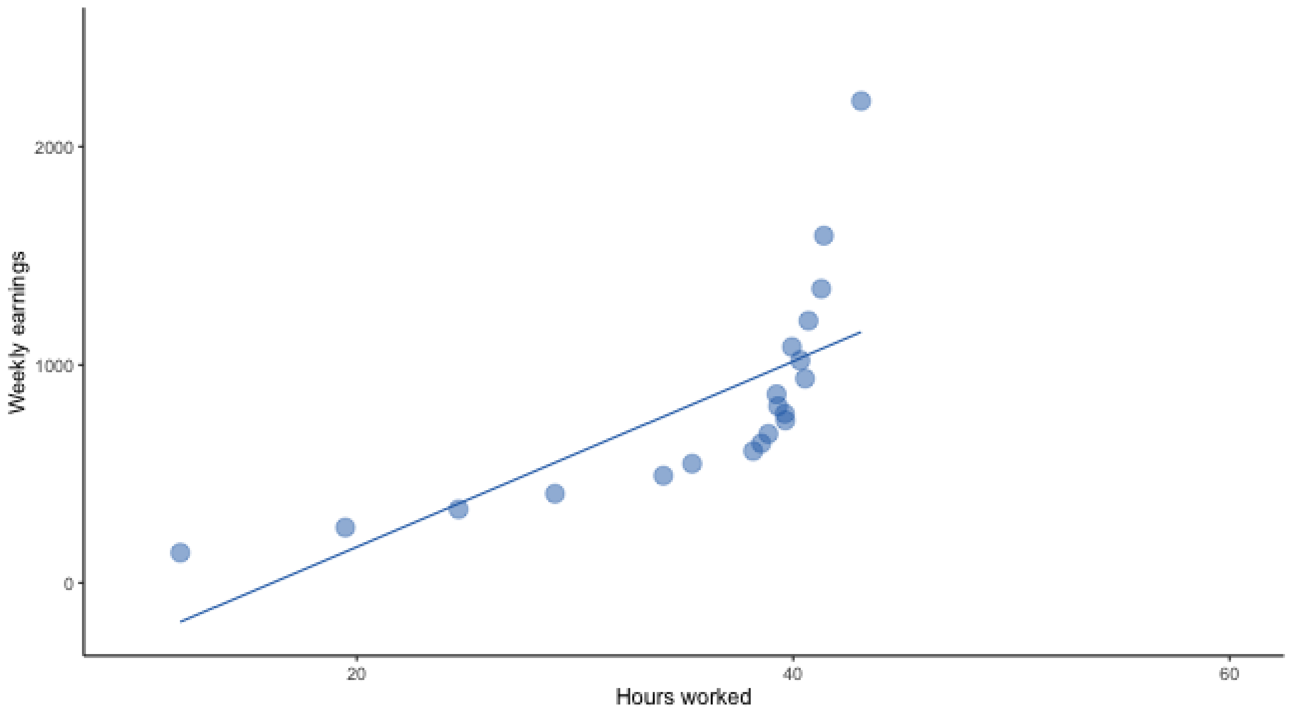
How can we best summarize this relationship?



REGRESSIONS

You are already familiar with **univariate regressions** $y = \alpha + \beta x + \epsilon$.

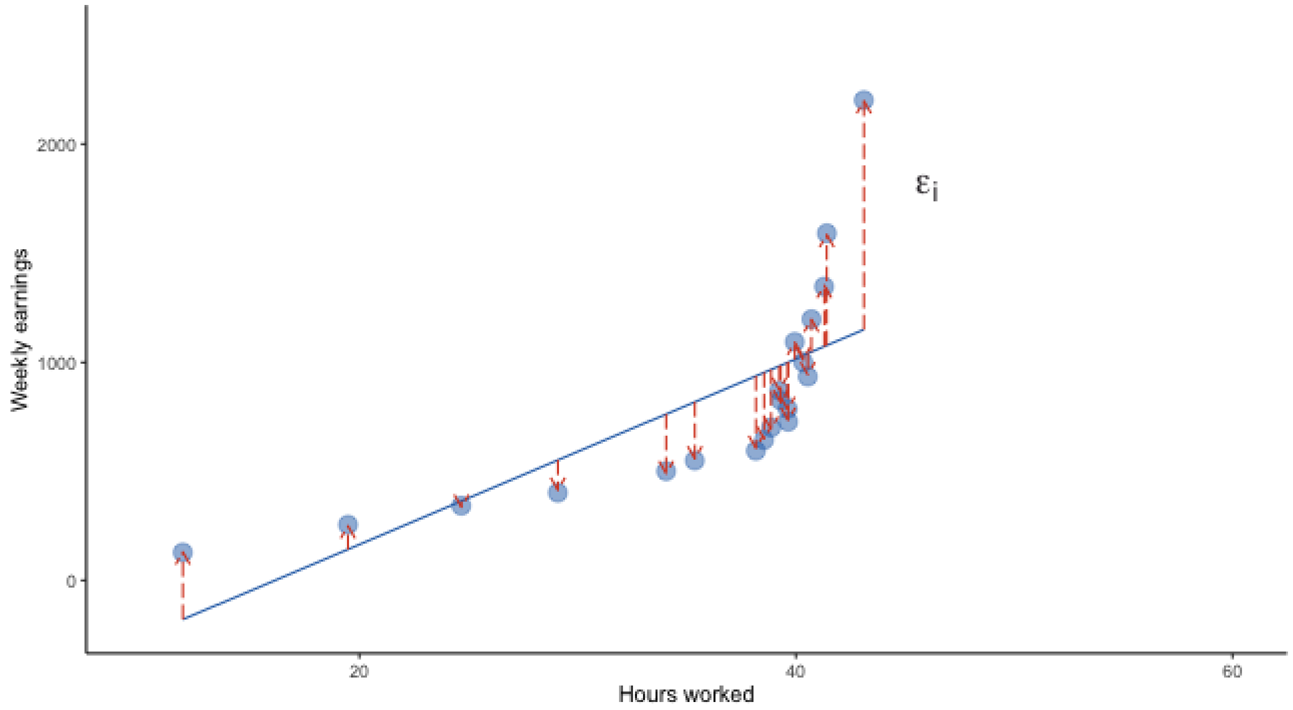
- We are looking for the line $\hat{y}_i = \hat{\alpha} + \hat{\beta} x_i$ that **minimizes the distance** to the data points



REGRESSIONS

You are already familiar with **univariate regressions** $y = \alpha + \beta x + \epsilon$.

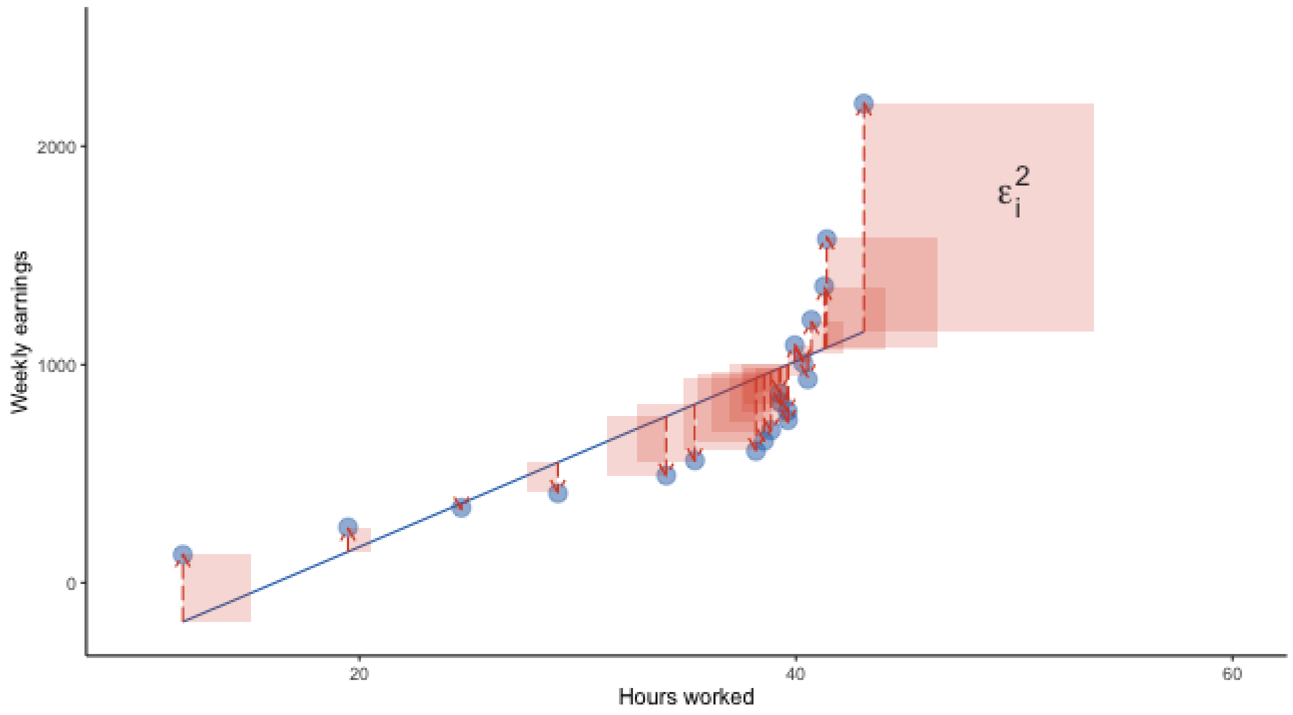
- We are looking foor the line $\hat{y}_i = \hat{\alpha} + \hat{\beta} x_i$ that **minimizes the distance** to the data points



REGRESSIONS

You are already familiar with **univariate regressions** $y = \alpha + \beta x + \epsilon$.

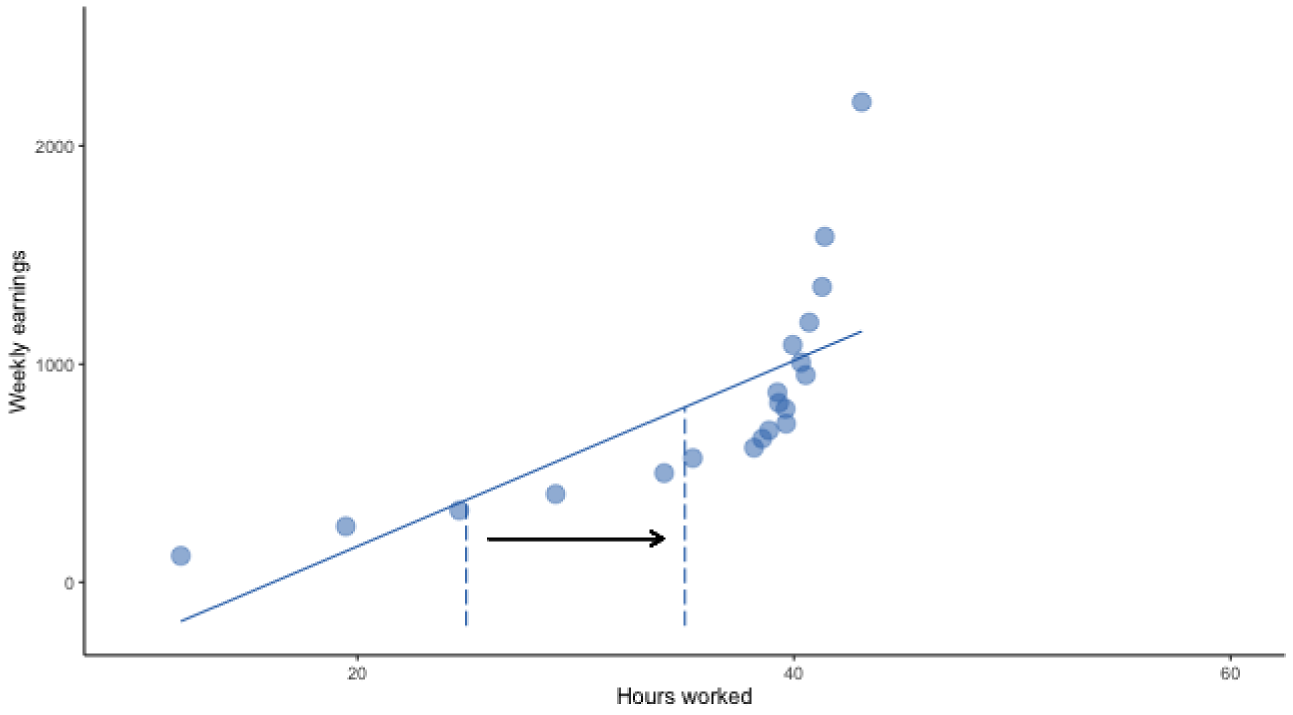
- We are looking foor the line $\hat{y}_i = \hat{\alpha} + \hat{\beta} x_i$ that **minimizes the distance** to the data points: $\sum \epsilon_i^2$



REGRESSIONS

You are already familiar with **univariate regressions** $y = \alpha + \beta x + \epsilon$.

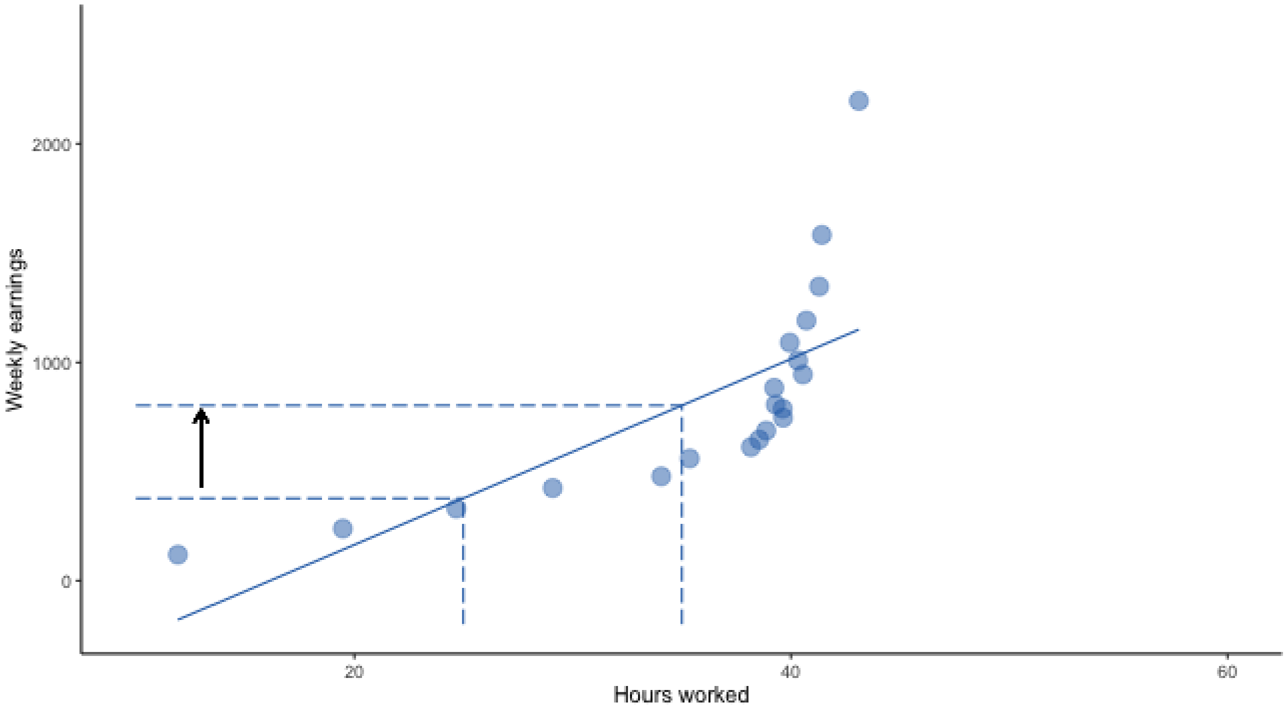
- We are looking for the line $\hat{y}_i = \hat{\alpha} + \hat{\beta} x_i$ that **minimizes the distance** to the data points: $\sum \epsilon_i^2$
  - Such that for a **one unit increase** in x,



REGRESSIONS

You are already familiar with **univariate regressions** $y = \alpha + \beta x + \epsilon$.

- We are looking for the line $\hat{y}_i = \hat{\alpha} + \hat{\beta} x_i$ that **minimizes the distance** to the data points: $\sum \epsilon_i^2$
  - Such that for a **one unit increase** in x, $\hat{\beta}$ indicates the associated **expected change** in y



REGRESSIONS

In R we can estimate a regression model using the `lm()` command (Linear Model), which takes the arguments:

- Formula. Written as `y ~ x`
- Data

```
lm(earnings ~ hours, cps)
```

```
##
## Call:
## lm(formula = earnings ~ hours, data = cps)
##
## Coefficients:
## (Intercept)        hours
##     -198.81        28.91
```

REGRESSIONS

- To get a more complete description of our regression, we use `summary()` on the regression object

```
mod <- lm(earnings ~ hours, cps)
summary(mod)
```

```
##
## Call:
## lm(formula = earnings ~ hours, data = cps)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2066.65  -237.53   -86.26   121.36  2882.47
##
## Coefficients:
##              Estimate Std. Error t value            Pr(>|t|)
## (Intercept) -198.8092    21.9152  -9.072 <0.0000000000000002 ***
## hours         28.9085     0.5911  48.904 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 393.3 on 4607 degrees of freedom
## Multiple R-squared:  0.3417,    Adjusted R-squared:  0.3416
## F-statistic:  2392 on 1 and 4607 DF,  p-value: < 0.00000000000000022
```

REGRESSIONS

- We get the **command** we used, including the **formula**

```
mod <- lm(earnings ~ hours, cps)
summary(mod)
```

```
##
## Call:
## lm(formula = earnings ~ hours, data = cps)
....
```

REGRESSIONS

- A description of the **distribution of residuals**

```
mod <- lm(earnings ~ hours, cps)
summary(mod)
```

```
##
## Call:
## lm(formula = earnings ~ hours, data = cps)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2066.65  -237.53   -86.26   121.36  2882.47
....
```

REGRESSIONS

- **Coefficients** and their **standard error**, **t-value**, and **p-value**

```r
mod <- lm(earnings ~ hours, cps)
summary(mod)
```

```
##
## Call:
## lm(formula = earnings ~ hours, data = cps)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2066.65  -237.53   -86.26   121.36  2882.47
##
## Coefficients:
##              Estimate Std. Error t value            Pr(>|t|)
## (Intercept) -198.8092    21.9152  -9.072 <0.0000000000000002 ***
## hours         28.9085     0.5911  48.904 <0.0000000000000002 ***
....
```

REGRESSIONS

- Significance thresholds and their symbols

```
mod <- lm(earnings ~ hours, cps)
summary(mod)
```

```
##
## Call:
## lm(formula = earnings ~ hours, data = cps)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2066.65  -237.53   -86.26   121.36  2882.47
##
## Coefficients:
##              Estimate Std. Error t value            Pr(>|t|)
## (Intercept) -198.8092    21.9152  -9.072 <0.0000000000000002 ***
## hours         28.9085     0.5911  48.904 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
....
```

REGRESSIONS

- The **residual standard error** $\sqrt{\sum(y_i - \hat{y}_i)^2/\mathrm{df}}$ and their symbols

```
mod <- lm(earnings ~ hours, cps)
summary(mod)
```

```
##
## Call:
## lm(formula = earnings ~ hours, data = cps)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2066.65  -237.53   -86.26   121.36  2882.47
##
## Coefficients:
##              Estimate Std. Error t value            Pr(>|t|)
## (Intercept) -198.8092    21.9152  -9.072 <0.0000000000000002 ***
## hours         28.9085     0.5911  48.904 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 393.3 on 4607 degrees of freedom
....
```

REGRESSIONS

- The R^2 and **adjusted** R^2

```r
mod <- lm(earnings ~ hours, cps)
summary(mod)
```

```
##
## Call:
## lm(formula = earnings ~ hours, data = cps)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2066.65  -237.53   -86.26   121.36  2882.47
##
## Coefficients:
##              Estimate Std. Error t value            Pr(>|t|)
## (Intercept) -198.8092    21.9152  -9.072 <0.0000000000000002 ***
## hours         28.9085     0.5911  48.904 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 393.3 on 4607 degrees of freedom
## Multiple R-squared:  0.3417,    Adjusted R-squared:  0.3416
....
```

REGRESSION RESULTS

- The results of an **F-test** ($H_0 : \beta_k = 0 \forall k$)

```r
mod <- lm(earnings ~ hours, cps)
summary(mod)
```

Formula

Residuals distribution

Coefficients

Significance levels

Residual std. error

R^2

F-test

```
##
## Call:
## lm(formula = earnings ~ hours, data = cps)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2066.65  -237.53   -86.26   121.36  2882.47
##
## Coefficients:
##             Estimate Std. Error t value            Pr(>|t|)
## (Intercept) -198.8092    21.9152  -9.072 <0.0000000000000002 ***
## hours         28.9085     0.5911  48.904 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 393.3 on 4607 degrees of freedom
## Multiple R-squared:  0.3417,    Adjusted R-squared:  0.3416
## F-statistic:  2392 on 1 and 4607 DF,  p-value: < 0.00000000000000022
```

REGRESSION RESULTS

All these elements are easily accessible using the `$` operator. You can find the definition of most elements in the **Value** section of the summary function documentation `?summary.lm()`

- Remember that functions take objects as arguments and produce new objects. **Value** describes the object that is created by a function.

```
sum_mod <- summary(mod)
str(sum_mod, give.attr = F)
```

```
## List of 11
##  $ call         : language lm(formula = earnings ~ hours, data = cps)
##  $ terms        :Classes 'terms', 'formula'  language earnings ~ hours
##  $ residuals    : Named num [1:4609] -398 592 -158 146 642 ...
##  $ coefficients : num [1:2, 1:4] -198.809 28.908 21.915 0.591 -9.072 ...
##  $ aliased      : Named logi [1:2] FALSE FALSE
##  $ sigma        : num 393
##  $ df           : int [1:3] 2 4607 2
##  $ r.squared    : num 0.342
##  $ adj.r.squared: num 0.342
##  $ fstatistic   : Named num [1:3] 2392 1 4607
##  $ cov.unscaled : num [1:2, 1:2] 0.00310512 -0.00008078 -0.00008078 0.00000226
```

REGRESSION RESULTS

Taking the **coefficients**:

```
sum_mod$coefficients
```

```
##                Estimate Std. Error   t value                  Pr(>|t|)
## (Intercept) -198.80923 21.9152104 -9.071746 0.00000000000000000001699063
## hours         28.90846  0.5911213 48.904453 0.00000000000000000000000000
```

We can **subset** this matrix like we would do with a regular `data.frame`:

```
sum_mod$coefficients[2,1]
```

```
## [1] 28.90846
```

```
sum_mod$coefficients[,"Std. Error"]
```

```
## (Intercept)       hours
##  21.9152104   0.5911213
```

We can use this to compute the fitted values $\hat{y} = \alpha + \beta x$

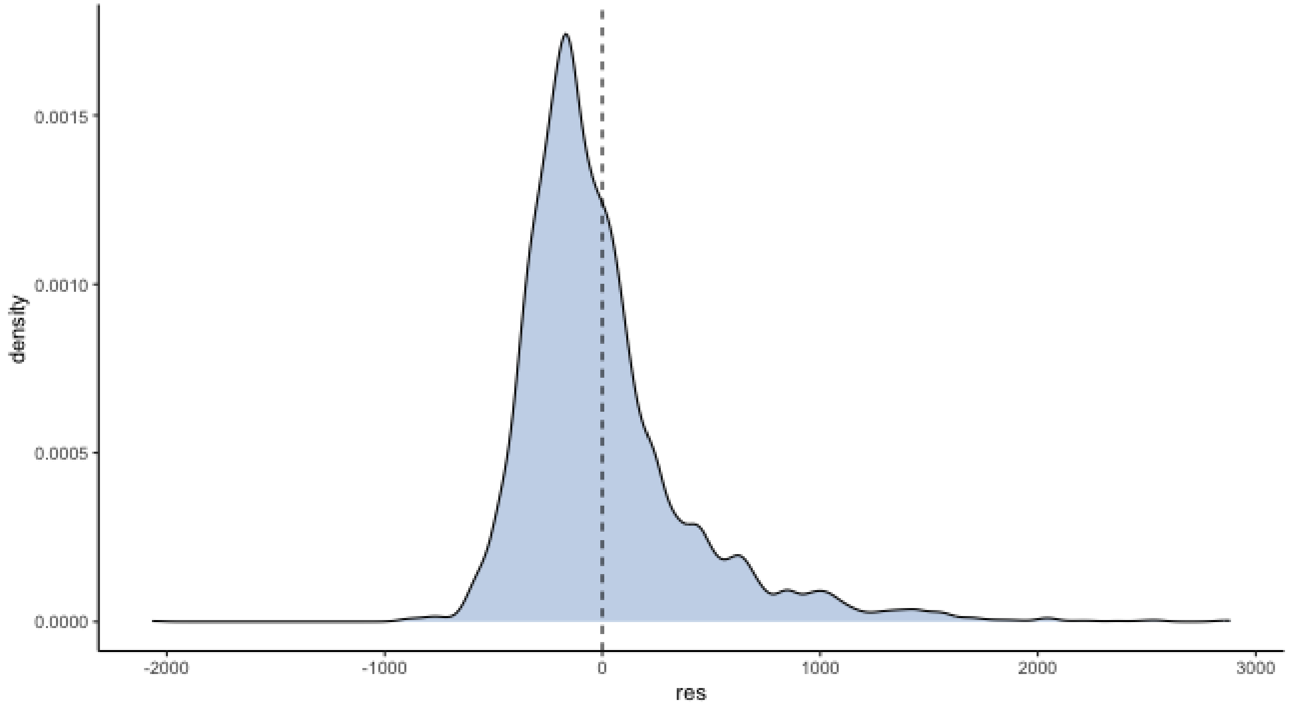
```
alpha <- sum_mod$coefficients[1,1]
beta  <- sum_mod$coefficients[2,1]

cps$hat_earnings <- alpha + (beta * cps$hours)
```

REGRESSION RESULTS

We can easily **plot** the **distribution** of our **residuals**

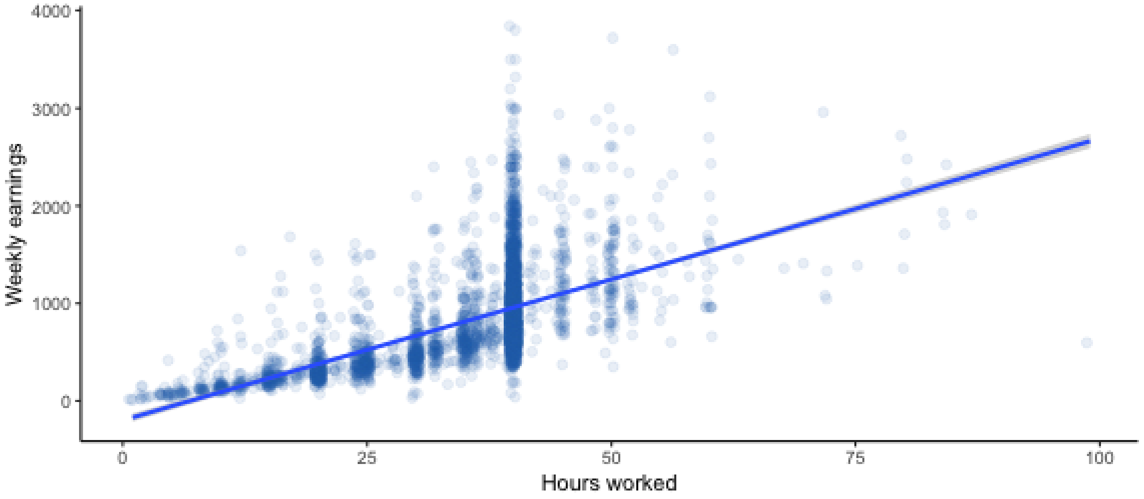
```
data.frame(res = sum_mod$residuals) %>%    # Convert the residuals to a data frame
  ggplot(aes(x = res)) +
  geom_density() +
  geom_vline(xintercept = 0, linetype = "dashed")
```



REGRESSION RESULTS

- `ggplot()` has a dedicated **geometry for fitted values** `geom_smooth()`

```
cps %>%
  ggplot(aes(x = hours, y = earnings)) +
  geom_point() +
  geom_smooth(method = "lm")
```



PRACTICE

Check that `lm()` works fine by computing a regression manually for the model:

$$y = \alpha + \beta x$$

where y is earnings and x is hours worked.

1. Start by creating a variable for $\hat{\beta}$, then for $\hat{\alpha}$, $\hat{y}_i$ and $\hat{\varepsilon}_i$

$$\hat{\beta} = \frac{Cov(x,y)}{Var(x)}; \quad \hat{\alpha} = \bar{y} - \hat{\beta} \times \bar{x}$$

   You will need the `cov()` and `var()` functions

   Review: where do these formulas come from? $E(u) = 0; \quad E(u|x) = 0$

2. Summarise the data to only display α, $\hat{\beta}$ and the R^2

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y}_i)^2}$$

10:00

PRACTICE

1. Start by creating a variable for $\hat{\beta}$, then for $\hat{\alpha}$, $\hat{y}_i$ and $\hat{\varepsilon}_i$

$$\hat{\beta} = \frac{Cov(x,y)}{Var(x)}; \quad \hat{\alpha} = \bar{y} - \hat{\beta} \times \bar{x}$$

```
cps
#
#
#
#
```

```
##      hours  earnings
## 1       40    560.00
## 2       30   1260.00
## 3       40    800.00
## 4       40   1103.20
## 5       40   1600.00
## 6       40    840.00
## 7       20    280.00
## 8       30     30.00
## 9       40    400.00
....
```

PRACTICE

1. Start by creating a variable for $\hat{\beta}$, then for $\hat{\alpha}$, $\hat{y}_i$ and $\hat{\varepsilon}_i$

$$\hat{\beta} = \frac{Cov(x,y)}{Var(x)}; \quad \hat{\alpha} = \bar{y} - \hat{\beta} \times \bar{x}$$

```
cps %>%
  mutate(beta = cov(hours, earnings) / var(hours))
#
#
#
```

```
##        hours earnings     beta
## 1         40   560.00 28.90846
## 2         30  1260.00 28.90846
## 3         40   800.00 28.90846
## 4         40  1103.20 28.90846
## 5         40  1600.00 28.90846
## 6         40   840.00 28.90846
## 7         20   280.00 28.90846
## 8         30    30.00 28.90846
## 9         40   400.00 28.90846
....
```

PRACTICE

1. Start by creating a variable for $\hat{\beta}$, then for $\hat{\alpha}$, $\hat{y}_i$ and $\hat{\varepsilon}_i$

$$\hat{\beta} = \frac{Cov(x,y)}{Var(x)}; \quad \hat{\alpha} = \bar{y} - \hat{\beta} \times \bar{x}$$

```
cps %>%
  mutate(beta = cov(hours, earnings) / var(hours),
         alpha = mean(earnings) - beta * mean(hours))
#
#
```

```
##      hours earnings     beta     alpha
## 1       40   560.00 28.90846 -198.8092
## 2       30  1260.00 28.90846 -198.8092
## 3       40   800.00 28.90846 -198.8092
## 4       40  1103.20 28.90846 -198.8092
## 5       40  1600.00 28.90846 -198.8092
## 6       40   840.00 28.90846 -198.8092
## 7       20   280.00 28.90846 -198.8092
## 8       30    30.00 28.90846 -198.8092
## 9       40   400.00 28.90846 -198.8092
....
```

PRACTICE

1. Start by creating a variable for $\hat{\beta}$, then for $\hat{\alpha}$, $\hat{y}_i$ and $\hat{\varepsilon}_i$

$$\hat{\beta} = \frac{Cov(x,y)}{Var(x)}; \quad \hat{\alpha} = \bar{y} - \hat{\beta} \times \bar{x}$$

```r
cps %>%
  mutate(beta = cov(hours, earnings) / var(hours),
         alpha = mean(earnings) - beta * mean(hours),
         y_hat = alpha + beta * hours)
#
```

```
##      hours earnings     beta     alpha      y_hat
## 1       40   560.00 28.90846 -198.8092  957.529242
## 2       30  1260.00 28.90846 -198.8092  668.444625
## 3       40   800.00 28.90846 -198.8092  957.529242
## 4       40  1103.20 28.90846 -198.8092  957.529242
## 5       40  1600.00 28.90846 -198.8092  957.529242
## 6       40   840.00 28.90846 -198.8092  957.529242
## 7       20   280.00 28.90846 -198.8092  379.360008
## 8       30    30.00 28.90846 -198.8092  668.444625
## 9       40   400.00 28.90846 -198.8092  957.529242
....
```

PRACTICE

1. Start by creating a variable for $\hat{\beta}$, then for $\hat{\alpha}$, $\hat{y}_i$ and $\hat{\varepsilon}_i$

$$\hat{\beta} = \frac{Cov(x, y)}{Var(x)}; \quad \hat{\alpha} = \bar{y} - \hat{\beta} \times \bar{x}$$

```
cps %>%
  mutate(beta = cov(hours, earnings) / var(hours),
         alpha = mean(earnings) - beta * mean(hours),
         y_hat = alpha + beta * hours,
         res = earnings - y_hat)
```

```
##      hours earnings     beta     alpha      y_hat          res
## 1       40   560.00 28.90846 -198.8092 957.529242 -397.5292415
## 2       30  1260.00 28.90846 -198.8092 668.444625  591.5553752
## 3       40   800.00 28.90846 -198.8092 957.529242 -157.5292415
## 4       40  1103.20 28.90846 -198.8092 957.529242  145.6707585
## 5       40  1600.00 28.90846 -198.8092 957.529242  642.4707585
## 6       40   840.00 28.90846 -198.8092 957.529242 -117.5292415
## 7       20   280.00 28.90846 -198.8092 379.360008  -99.3600080
## 8       30    30.00 28.90846 -198.8092 668.444625 -638.4446248
## 9       40   400.00 28.90846 -198.8092 957.529242 -557.5292415
....
```

PRACTICE

1. Summarise the data to only display α, $\hat{\beta}$ and the $R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y}_i)^2}$

```
cps %>%
  mutate(beta = cov(hours, earnings) / var(hours),
         alpha = mean(earnings) - beta * mean(hours),
         y_hat = alpha + beta * hours,
         res = earnings - y_hat) %>%
  summarise(alpha = first(alpha),
            beta = first(beta),
            r2 = 1 - sum(res^2) / sum((earnings - mean(earnings))^2))
```

```
##       alpha     beta        r2
## 1 -198.8092 28.90846 0.3417298
```

```
sum_mod$coefficients[,"Estimate"]
```

```
## (Intercept)       hours
##  -198.80923    28.90846
```

```
sum_mod$r.squared
```

```
## [1] 0.3417298
```

USING BINARY AND CATEGORICAL VARIABLES

- We want to know the **relationship** between people's **sex** and **earnings**

```
cps$sex
```

```
##    [1] "Male"   "Female" "Male"   "Male"   "Male"   "Female" "Male"   "Male"
##    [9] "Male"   "Female" "Female" "Male"   "Female" "Female" "Female" "Male"
##   [17] "Female" "Female" "Female" "Female" "Female" "Female" "Female" "Female"
....
```

Can we just regress earnings on sex even though it's a character variable?

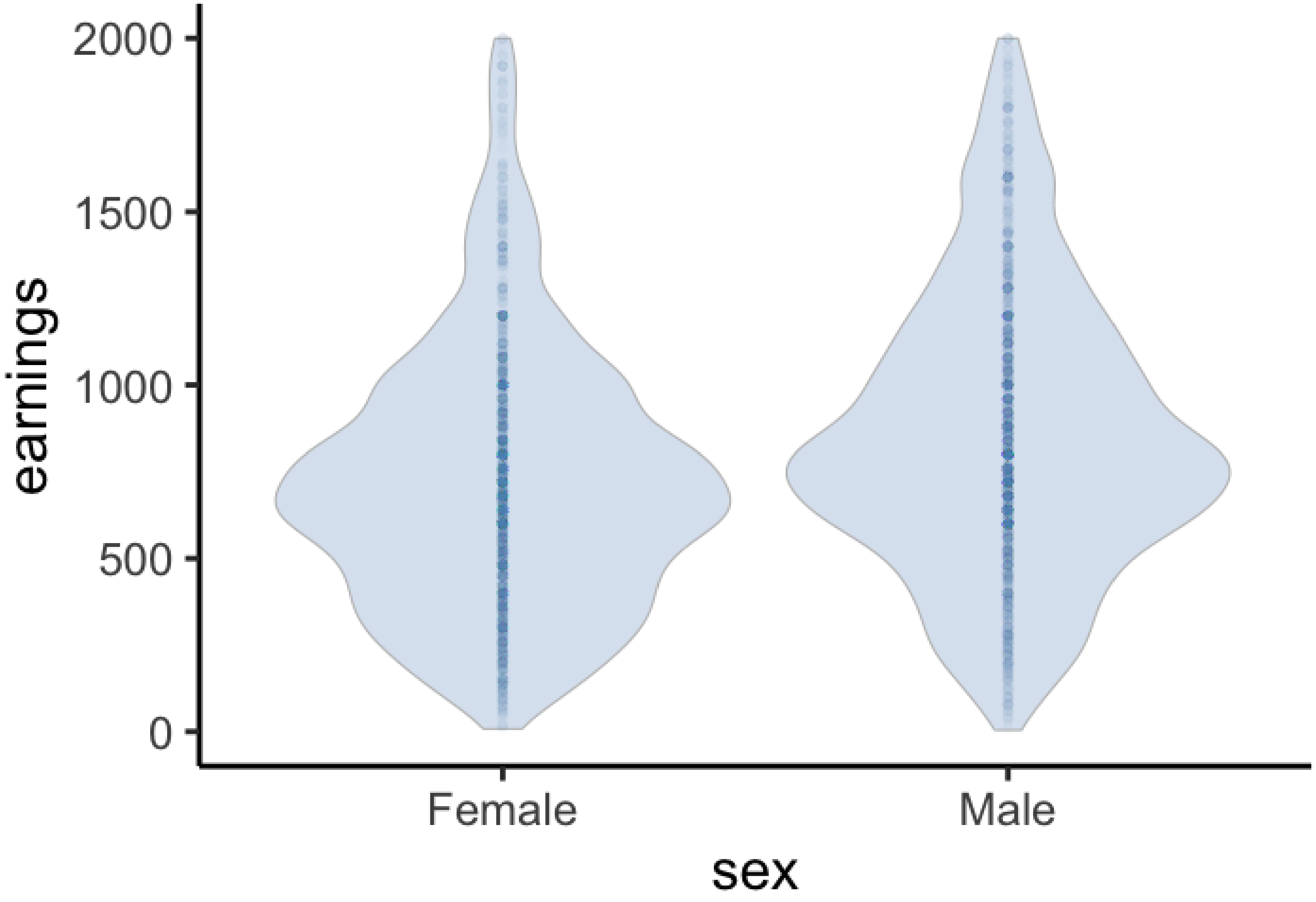
```
lm(earnings ~ sex, data = cps)
```

```
##
## Call:
## lm(formula = earnings ~ sex, data = cps)
##
## Coefficients:
## (Intercept)      sexMale
##       746.3        179.3
```

USING BINARY AND CATEGORICAL VARIABLES

What is going on?

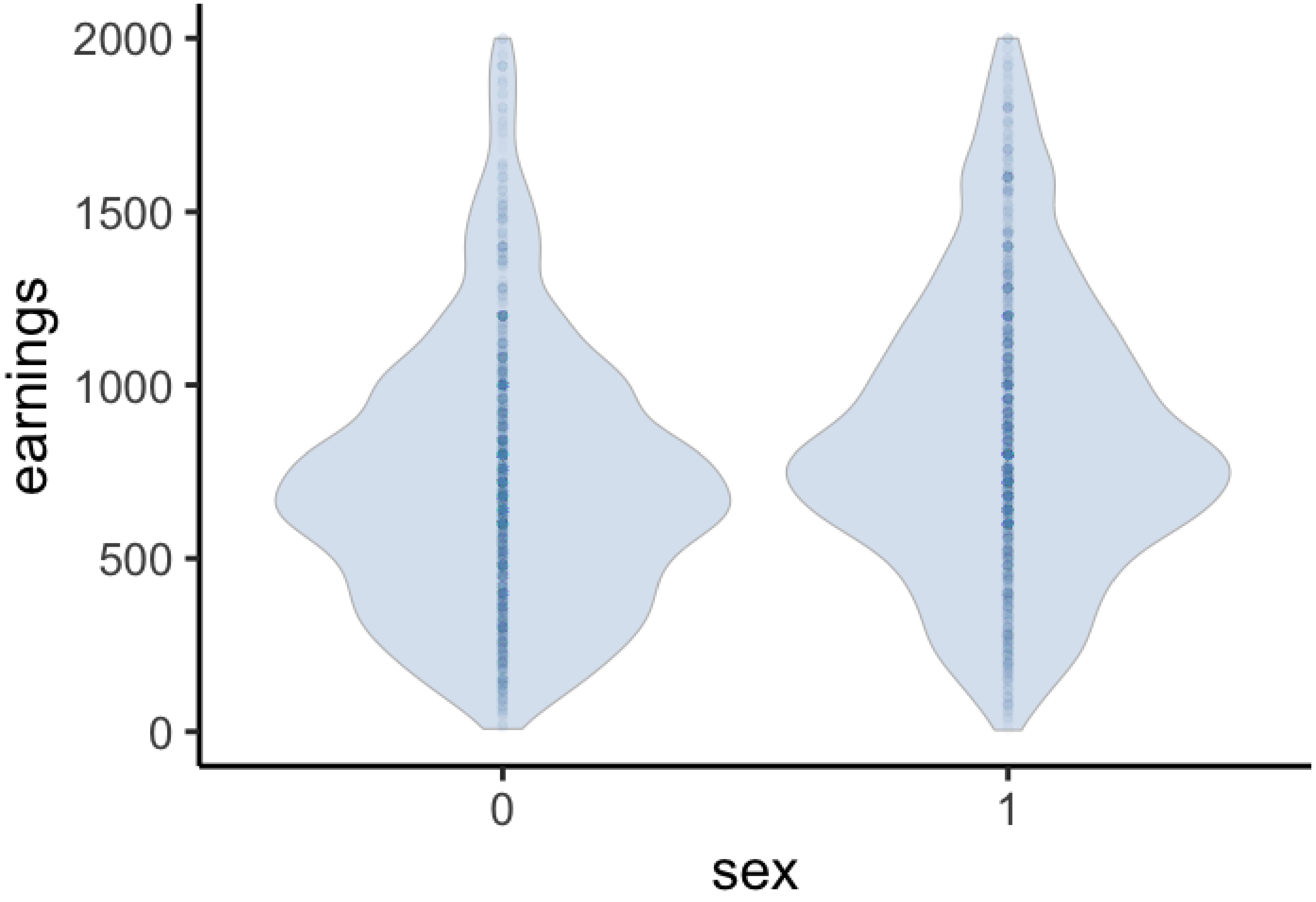
- R implicitely **converts** character variables into **binary** variables



USING BINARY AND CATEGORICAL VARIABLES

What is going on?

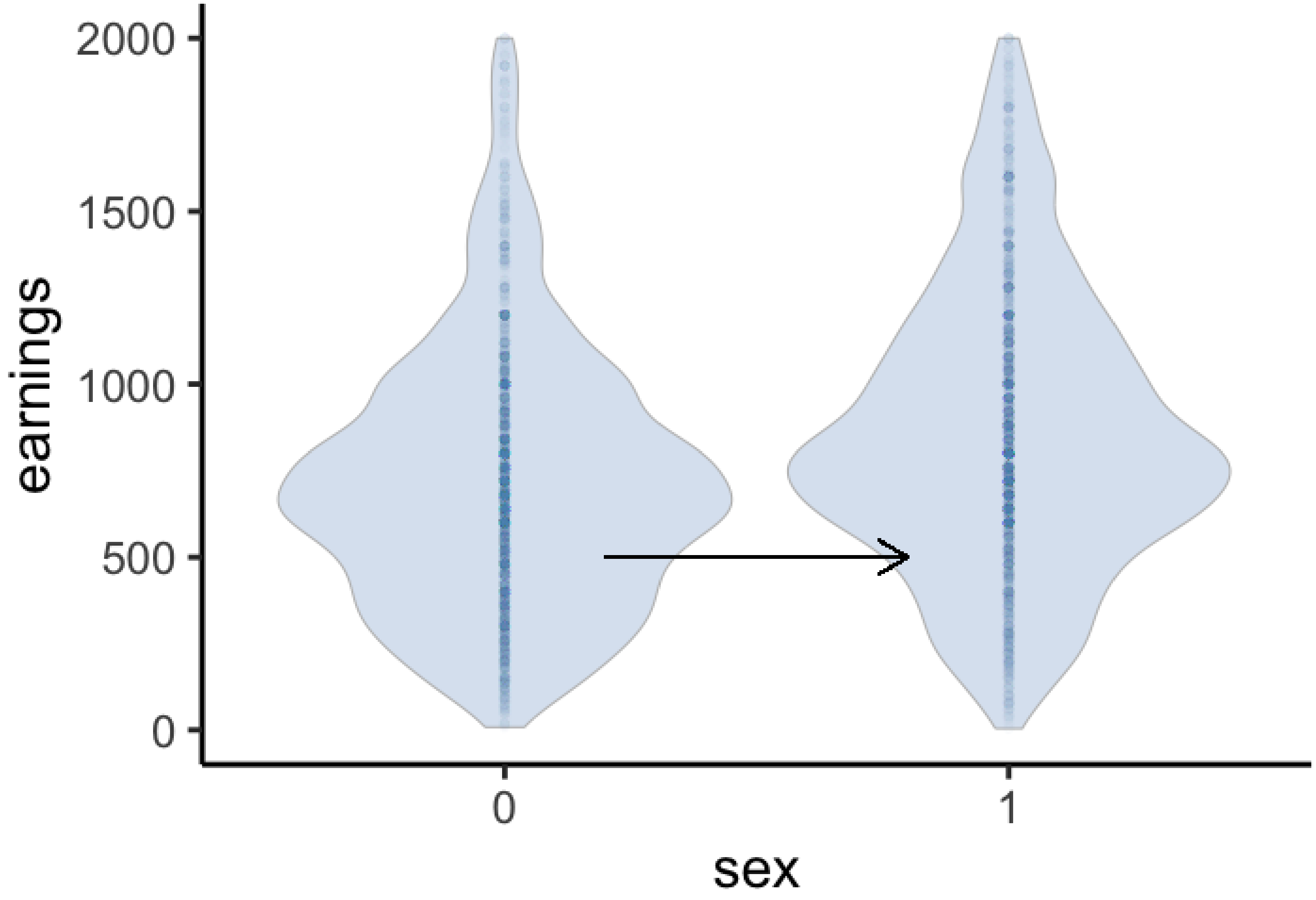
- ... to ressemble the **continuous** case, such that we're looking at a **1-unit increase** in x



USING BINARY AND CATEGORICAL VARIABLES

What is going on?

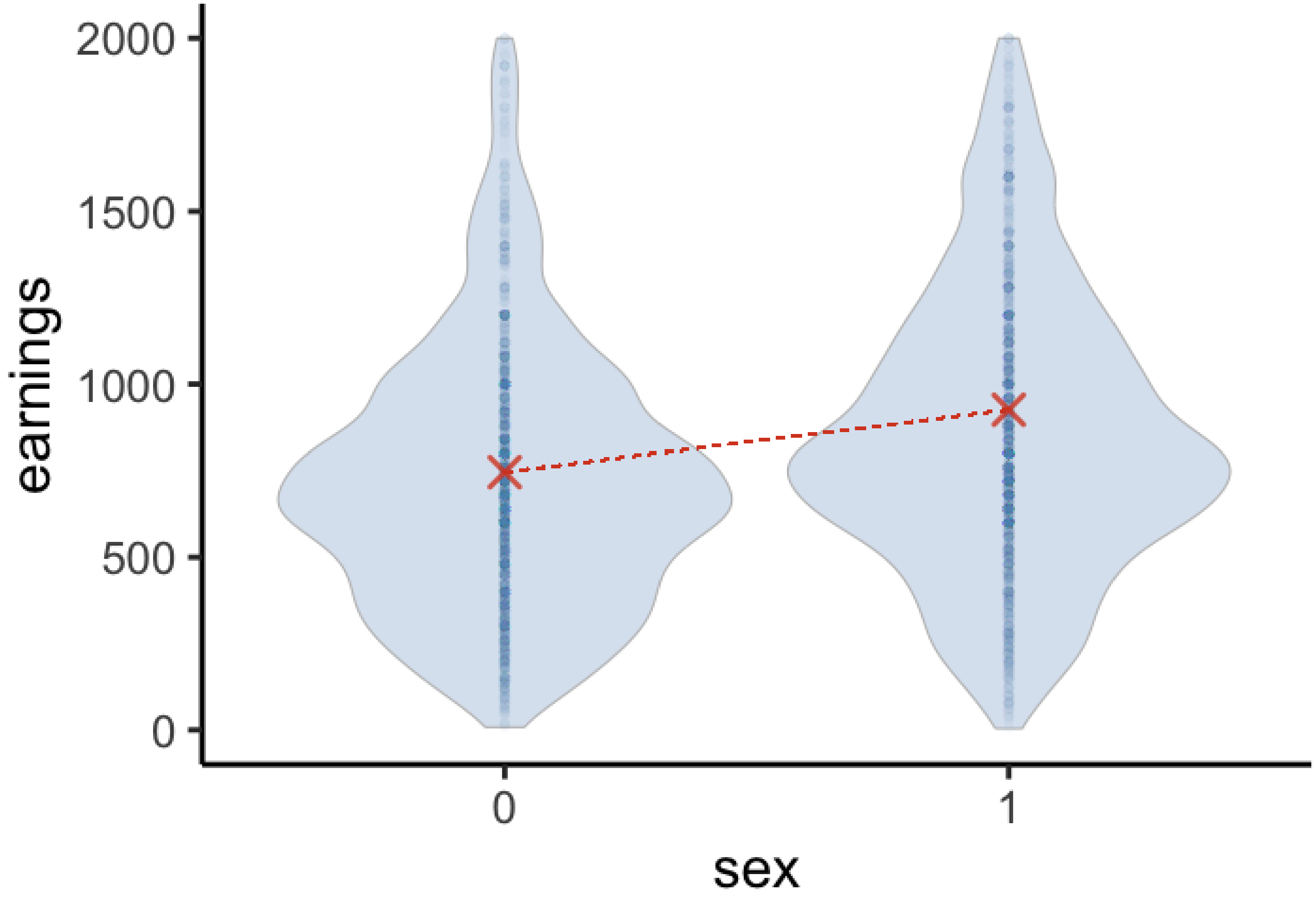
- ... to ressemble the **continuous** case, such that we're looking at a **1-unit increase** in x



USING BINARY AND CATEGORICAL VARIABLES

What is going on?

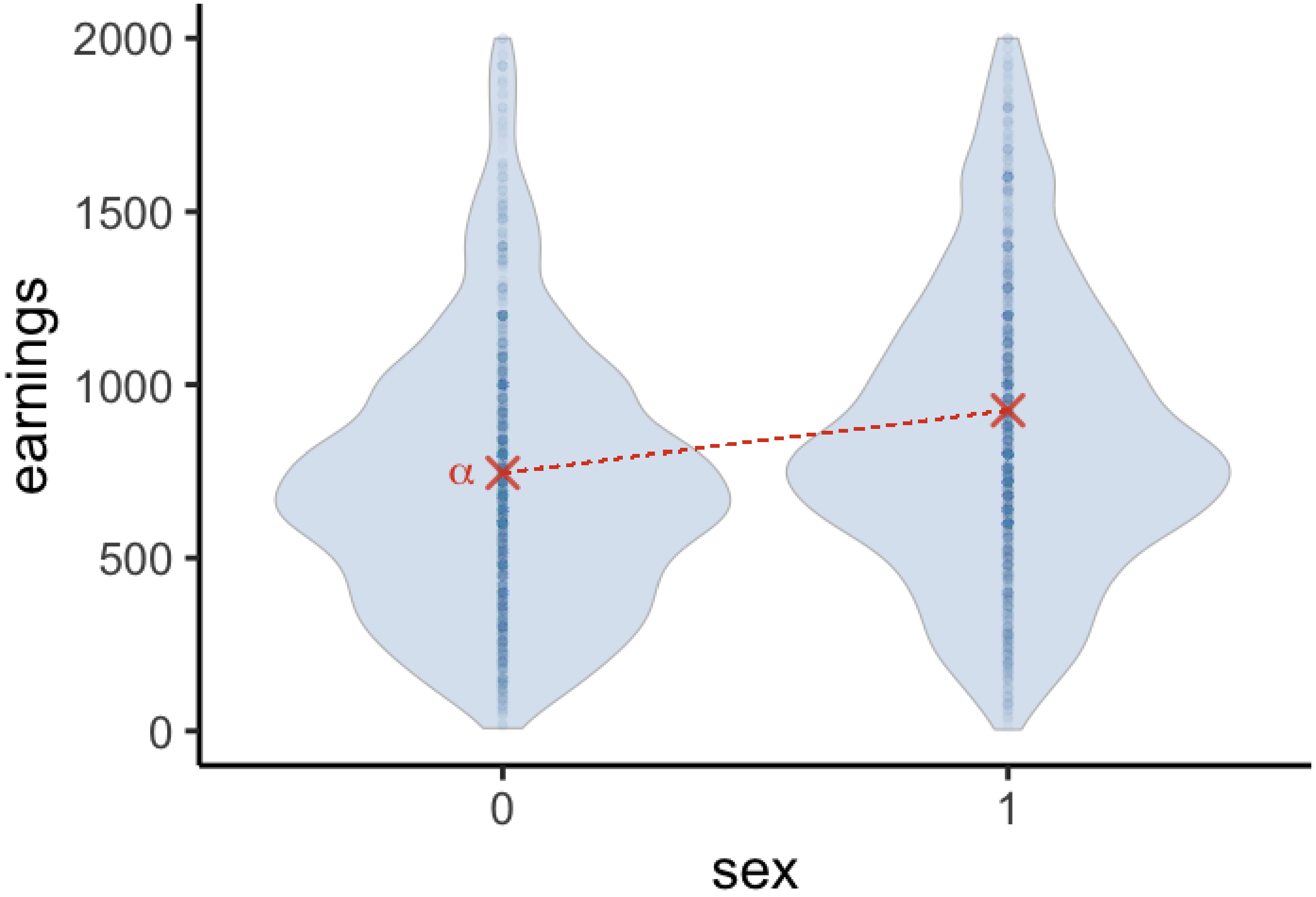
- Also, the **fit** is necessarily going through the **mean of each category**



USING BINARY AND CATEGORICAL VARIABLES

What is going on?

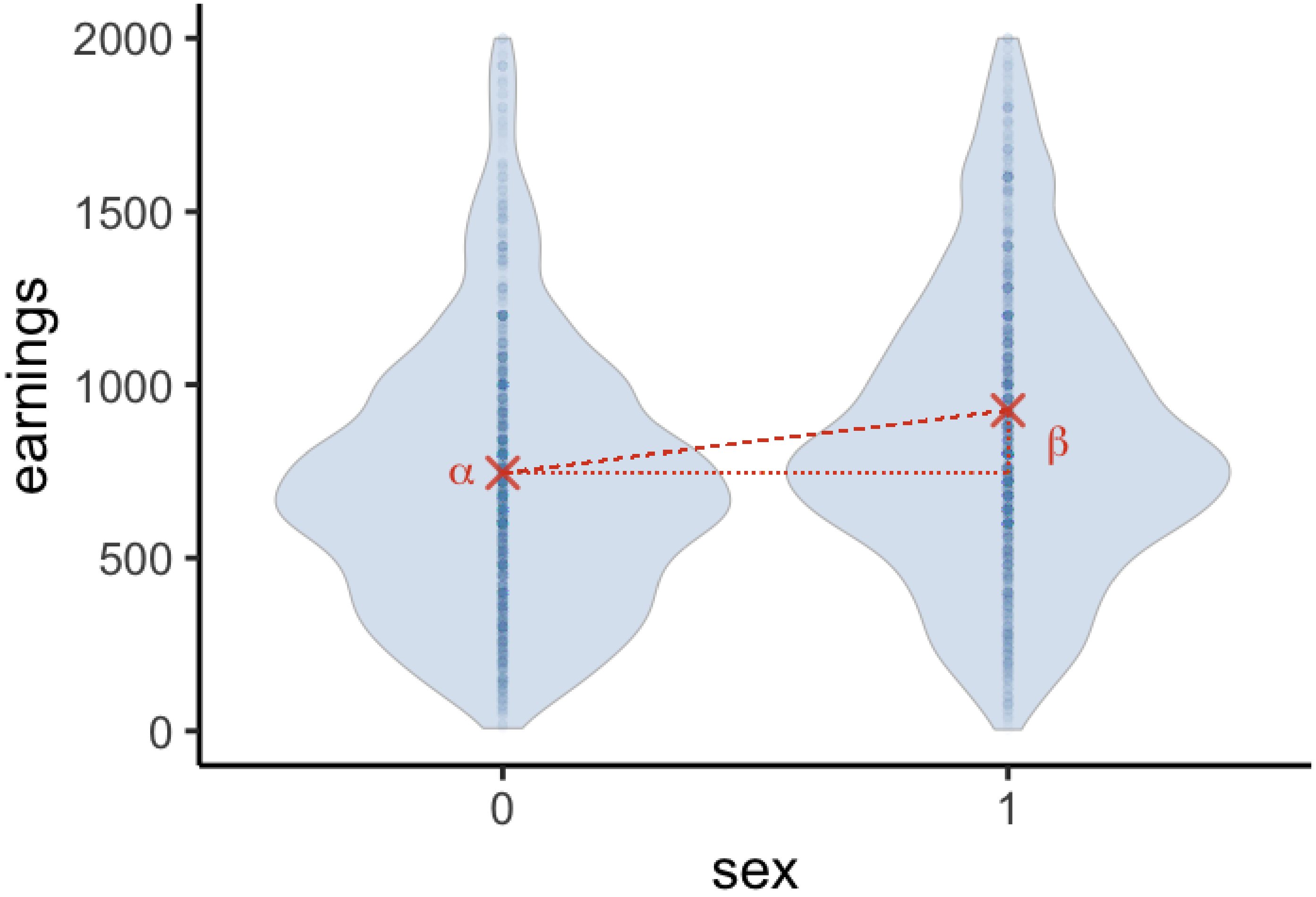
- Such that $\hat{\alpha}$ is the **mean of the reference group**



USING BINARY AND CATEGORICAL VARIABLES

What is going on?

- And $\hat{\beta}$ is the **difference in means**



USING BINARY AND CATEGORICAL VARIABLES

We can verify this easily:

```
cps %>%
  group_by(sex) %>%
  summarise(y_bar = mean(earnings)) %>%
  mutate(dif = y_bar - y_bar[1])
```

```
## # A tibble: 2 × 3
##   sex    y_bar   dif
##   <chr>  <dbl> <dbl>
## 1 Female  746.    0
## 2 Male    926.  179.
```

```
lm(earnings ~ sex, data = cps)
```

```
##
## Call:
## lm(formula = earnings ~ sex, data = cps)
##
## Coefficients:
## (Intercept)      sexMale
##       746.3        179.3
```

USING BINARY AND CATEGORICAL VARIABLES

Let's complicate it by using categorical variables with more than two values. Let's see how **education** relates to **earnings**

```
unique(cps$educ)
```

```
## [1] "High school"        "Bachelor's degree" "No high school"
## [4] "Associate degree"
```

Before, a **2-category** variables was equivalent to 1 **dummy variable**.
now an **n-category** variable is equivalent to **n-1 dummy variables**

sex	male
Female	0
Female	0
Female	0
Male	1
Male	1
Male	1

educ	Bachelor's degree	High school	No high school
High school	0	1	0
Bachelor's degree	1	0	0
No high school	0	0	1
Associate degree	0	0	0

USING BINARY AND CATEGORICAL VARIABLES

Once again the **constant** is the average y for the reference category and the **slopes** are the relative **differences in means**

```
lm(earnings ~ educ, data = cps)
```

```
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)                904.01      19.47   46.43        0
## educBachelor's degree      106.06      24.65    4.30        0
## educHigh school            -97.30      21.58   -4.51        0
## educNo high school        -328.85      28.19  -11.67        0
```

```
cps %>%
  group_by(educ) %>%
  summarise(y_bar = mean(earnings)) %>%
  mutate(dif = y_bar - y_bar[1])
```

```
## # A tibble: 4 × 3
##   educ              y_bar    dif
##   <chr>             <dbl>  <dbl>
## 1 Associate degree   904.      0
## 2 Bachelor's degree 1010.   106.
## 3 High school        807.  -97.3
## 4 No high school     575. -329.
```

Note that R always sort **character** variables by **alphabetical order**

- Does it make sense that our **reference category** is "Associate's degree" ?
- We'd prefer to have "No high school" (the least education) as the **reference category**

USING BINARY AND CATEGORICAL VARIABLES

We need to talk about **factor variables** (another **class** of R objects)

- Variables whose values **indicate** different groups
- They take different values that are **arbitrary group classifiers**

```
## [1] 1
## Levels: 1 2 3 4 5
```

- R understands that the different values **do not mean anything** they are there to differentiate groups only

```r
states * 2
```

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

- We can specify the **levels** and **labels** that they take:

```r
x <- c("Man", "Male", "Man", "Lady", "Female")
## Map from 4 different values to only two levels:
xf <- factor(x, levels = c("Male", "Man" , "Lady",   "Female"),
                labels = c("Male", "Male", "Female", "Female"))
xf
```

```
## [1] Male   Male   Male   Female Female
## Levels: Male Female
```

USING BINARY AND CATEGORICAL VARIABLES

Going back to our education variable, we want to **order it** in a way that makes sense:

```
levels(as.factor(cps$educ)) # Default order is alphabetical
```

```
## [1] "Associate degree"  "Bachelor's degree" "High school"
## [4] "No high school"
```

Let's create a **factor variable** with the right order using `factor()`

```
cps <-
  cps %>%
  mutate(educf = factor(educ, levels = c("No high school", "High school",
                                         "Associate degree", "Bachelor's degree")))
levels(cps$educf)
```

```
## [1] "No high school"    "High school"       "Associate degree"
## [4] "Bachelor's degree"
```

Or **only** change the reference category (first level), using `relevel()`

```
cps <-
  cps %>%
  mutate(educf = relevel(as.factor(educ), ref = "No high school"))

levels(cps$educf)
```

USING BINARY AND CATEGORICAL VARIABLES

```
lm(earnings ~ educf, data = cps)
```

```
##
## Call:
## lm(formula = earnings ~ educf, data = cps)
##
## Coefficients:
##            (Intercept)   educfAssociate degree  educfBachelor's degree
##                  575.2                   328.8                   434.9
##        educfHigh school
##                  231.5
```

We can also modify our `educ` variable directly in the regression call:

```
lm(earnings ~ relevel(as.factor(educ), ref = "No high school"), data = cps)
```

or

```
lm(earnings ~ factor(educ, levels = c("No high school", "High school",
                                      "Associate degree", "Bachelor's degree")),
   data = cps)
```

PRACTICE

1. Load the Current Population Survey data `cps,csv` in case you haven't
2. Regress the earnings on the age variable
3. Redo the same regression afer converting the age variable as a **factor**

04:00

PRACTICE

```
lm(earnings ~ age, data = cps) %>%
  summary()
```

```
##
## Call:
## lm(formula = earnings ~ age, data = cps)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -966.33 -308.32  -75.55  207.35 3012.86
##
## Coefficients:
##               Estimate Std. Error t value
Pr(>|t|)
## (Intercept) 617.8860    19.1940   32.19
<0.0000000000000002 ***
## age           5.5067     0.4534   12.15
<0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*'
0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 477.2 on 4607
degrees of freedom
## Multiple R-squared:  0.03103,    Adjusted R-
```

```
lm(earnings ~ as.factor(age), data = cps) %>%
  summary()
```

```
##
## Call:
## lm(formula = earnings ~ as.factor(age), data
= cps)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -995.53 -283.21  -75.23  191.46 2852.56
##
## Coefficients:
##                   Estimate Std. Error t value
Pr(>|t|)
## (Intercept)        294.167    117.202   2.510
0.012111 *
## as.factor(age)16    -9.190    128.533  -0.072
0.943002
## as.factor(age)17     7.128    126.798   0.056
0.955174
## as.factor(age)18   174.198    126.695   1.375
0.169217
## as.factor(age)19   231.065    124.804   1.851
0.064174 .
```

USING BINARY AND CATEGORICAL VARIABLES

We can also **one hot encode** the data: converting the categorical variable to **several dummies** so that everything is numeric

```
cps <- cps %>%
  mutate(educ_nohs  = as.numeric(educ == "No high school"),
         educ_hs    = as.numeric(educ == "High school"),
         educ_assoc = as.numeric(educ == "Associate degree"),
         educ_bach  = as.numeric(educ == "Bachelor's degree"))
```

```
##                 educ educ_nohs educ_hs educ_assoc educ_bach
## 1        High school         0       1          0         0
## 2     No high school         1       0          0         0
## 3        High school         0       1          0         0
## 4 Bachelor's degree          0       0          0         1
```

Then, we can include them as different variables in the regression using the `+` sign

```
lm(earnings ~ educ_hs + educ_assoc + educ_bach, data = cps)
```

Why do we need to omit one category?

USING BINARY AND CATEGORICAL VARIABLES

Why do we need to omit one category?

- `We are in the` **`multivariate case`**`, so we move from` $\hat{\beta} = \frac{Cov(x,y)}{Var(x)}$ `to` $\hat{\beta} = (\boldsymbol{X'X})^{-1}\boldsymbol{X'}y$

```
y <- as.matrix(cps$earnings)
X <- cps %>%
  mutate(constant = 1) %>%
  select(constant, contains("educ_")) %>%
  as.matrix()
```

```
dim(y)
```

```
## [1] 4609    1
```

y

```
dim(X)
```

```
## [1] 4609    5
```

$\boldsymbol{X}$

```
dim(t(X))
```

```
## [1]    5 4609
```

$\boldsymbol{X'}$

```
dim(t(X) %*% X)
```

```
## [1] 5 5
```

$\boldsymbol{X'X}$

USING BINARY AND CATEGORICAL VARIABLES

Because of perfect **multicollinearity** (no explanatory variable is a perfect linear function of other explanatory variables) it will not be posible to invert $\boldsymbol{X'X}$

```
solve(t(X) %*% X)
```

$(\boldsymbol{X'X})^{-1}$

```
## [1] "Error in solve.default(t(X) %*% X)
system is computationally singular"
```

```
X
```

```
##        constant educ_nohs educ_hs educ_assoc educ_bach
##  [1,]         1         0       1          0         0
##  [2,]         1         0       0          0         1
##  [3,]         1         1       0          0         0
##  [4,]         1         0       1          0         0
##  [5,]         1         0       0          1         0
##  [6,]         1         0       0          1         0
##  [7,]         1         0       1          0         0
##  [8,]         1         0       0          0         1
##  [9,]         1         0       0          0         1
....
```

- constant = educ_nohs + educ_hs + educ_assoc + educ_bach
- educ_bach = 1 - educ_nohs - educ_hs - educ_assoc

USING BINARY AND CATEGORICAL VARIABLES

We need to:

- Remove one category

```
X <- cps %>%
  mutate(constant = 1) %>%
  select(constant, educ_hs,
         educ_assoc, educ_bach) %>%
  as.matrix()

solve(t(X) %*% X) %*% (t(X) %*% y)
```

```
##                  [,1]
## constant   575.1593
## educ_hs    231.5476
## educ_assoc 328.8480
## educ_bach  434.9049
```

```
lm(earnings ~ educ_hs +
              educ_assoc + educ_bach,
   cps)
```

- Or remove the constant

```
X <- cps %>%
  select(educ_nohs, educ_hs,
         educ_assoc, educ_bach) %>%
  as.matrix()

solve(t(X) %*% X) %*% (t(X) %*% y)
```

```
##                  [,1]
## educ_nohs   575.1593
## educ_hs     806.7068
## educ_assoc  904.0072
## educ_bach  1010.0642
```

```
lm(earnings ~ educ_nohs + educ_hs +
              educ_assoc + educ_bach - 1,
   cps)
```

- You can remove the constant in `lm()` by adding `-1` to the formula

USING BINARY AND CATEGORICAL VARIABLES

Actually, if we don't drop anything `lm()` would still work:

```
lm(earnings ~ educ_nohs + educ_hs + educ_assoc + educ_bach,
   cps)
```

```
##
## Call:
## lm(formula = earnings ~ educ_nohs + educ_hs + educ_assoc + educ_bach,
##     data = cps)
##
## Coefficients:
## (Intercept)    educ_nohs      educ_hs   educ_assoc    educ_bach
##      1010.1       -434.9       -203.4       -106.1           NA
```

- It will automatically drop one of the categories **but** it might not be the most adequate reference category
- Even if **multicollinearity** does not break `lm()` you need to be mindful about it
  - Make sure the your explanatory variables are not redundant
  - Multicollinearity will invalidate statistical inference (inflate standard errors)
- Always be **intentional** about your reference categories!

AGENDA

- REGRESSIONS IN R
  - CONTINUOUS VARIABLES
  - CATEGORICAL AND BINARY VARIABLES

- GETTING TO THE RIGHT MODEL
  - VARIABLE TRANSFORMATION
  - FUNCTIONAL FORM
  - CONTROL VARIABLES
  - INTERACTION TERMS

- INFERENCE

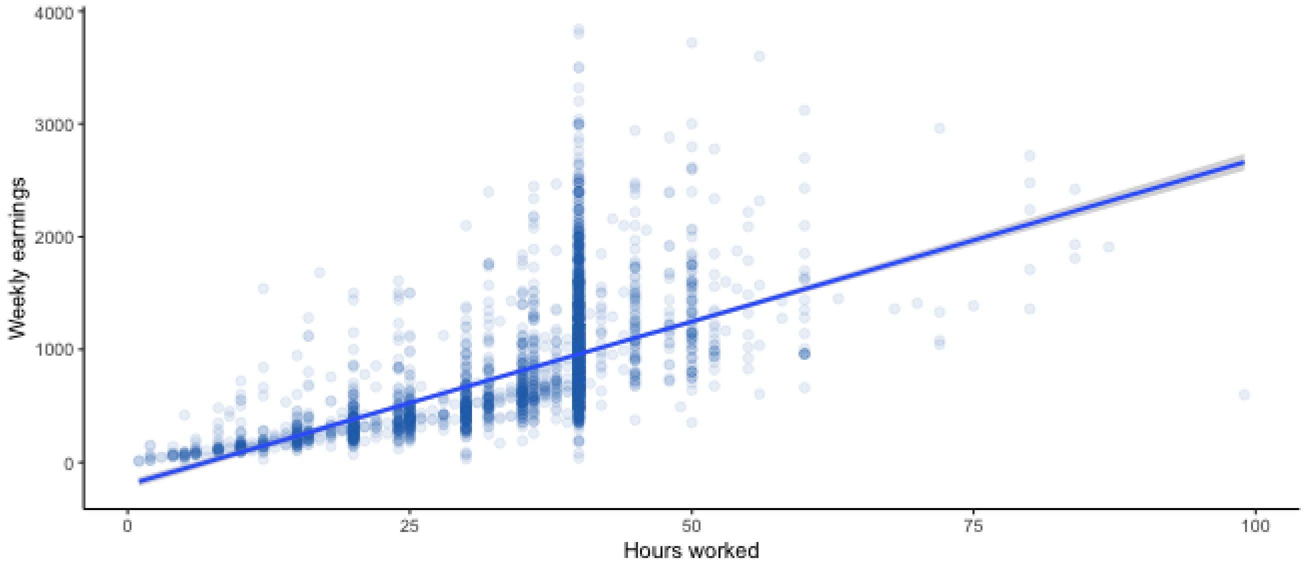
- EXPORTING RESULTS
  - REGRESSION TABLES
  - COEFFICIENT PLOTS

- PRACTICING YOUR SKILLS

GETTING TO THE *RIGHT* MODEL

We want to examine the relationship between earnings and hours worked. We might need to add a couple of things before we are happy with our model:

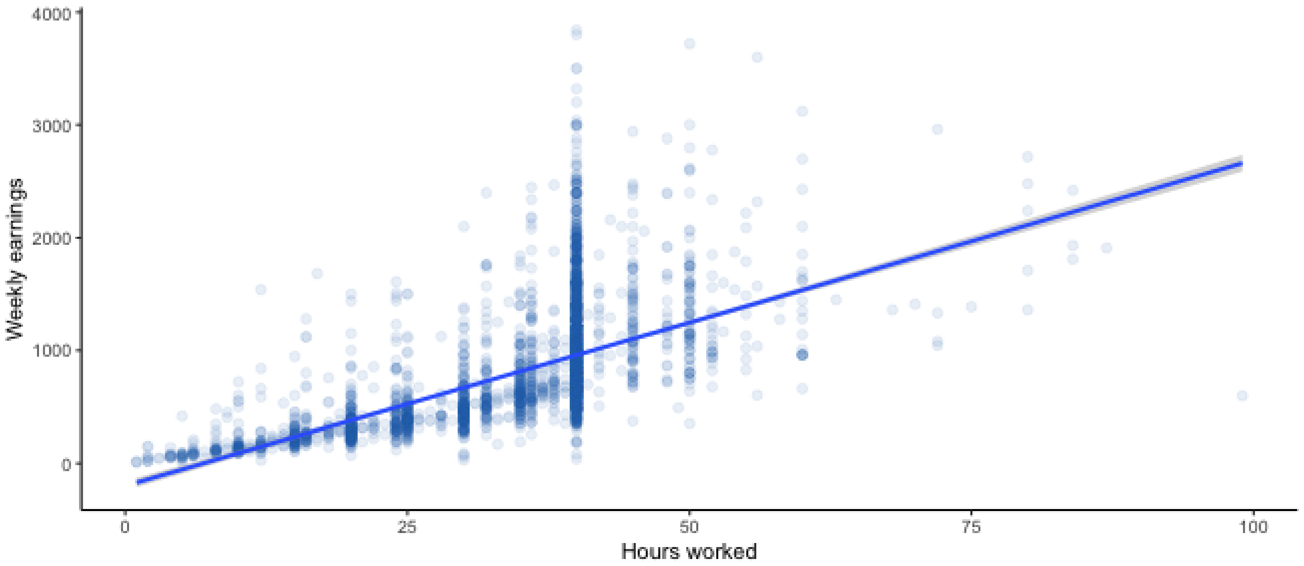
- Variable transformation
- Functional form
- Control variables
- Interaction terms



GETTING TO THE *RIGHT* MODEL

Variable transformation

What we have so far:

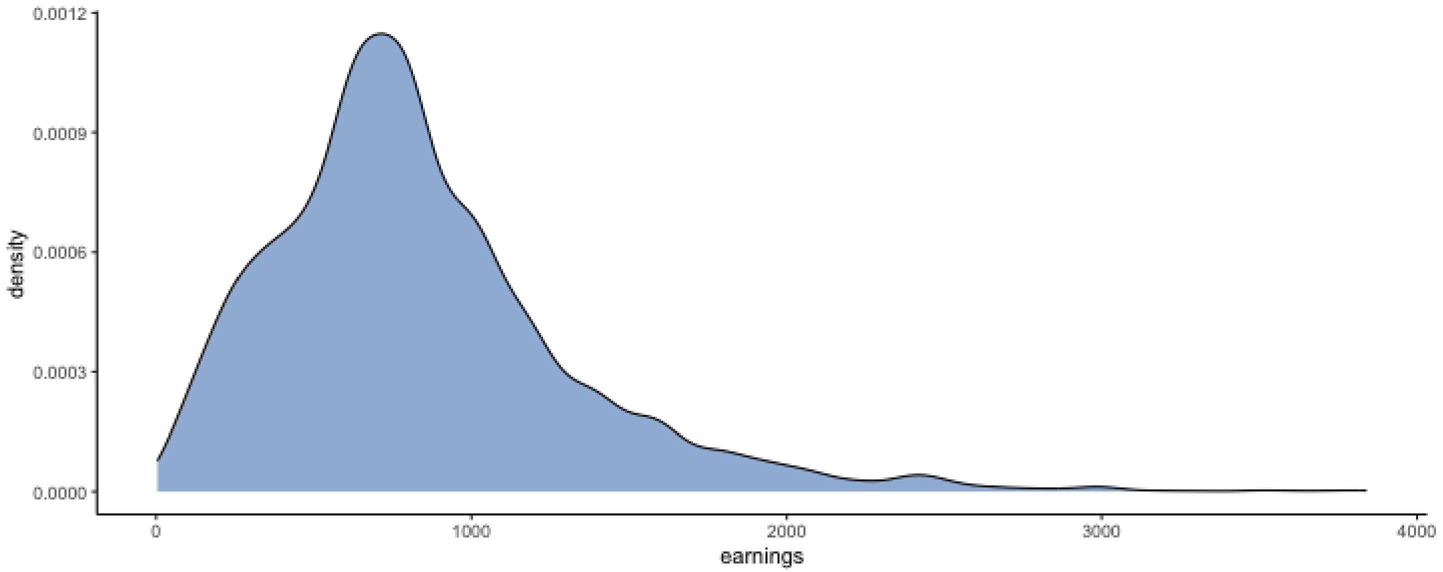


- Earnings are concentrated below $2000
- The regression does not fit well on the **y dimension**

GETTING TO THE *RIGHT* MODEL

Variable transformation

```
ggplot(cps, aes(x = earnings)) +
  geom_density()
```



- Earnings are concentrated below $2000, with fewer and fewer observations after that
- The estimated probability density function has a long right tail

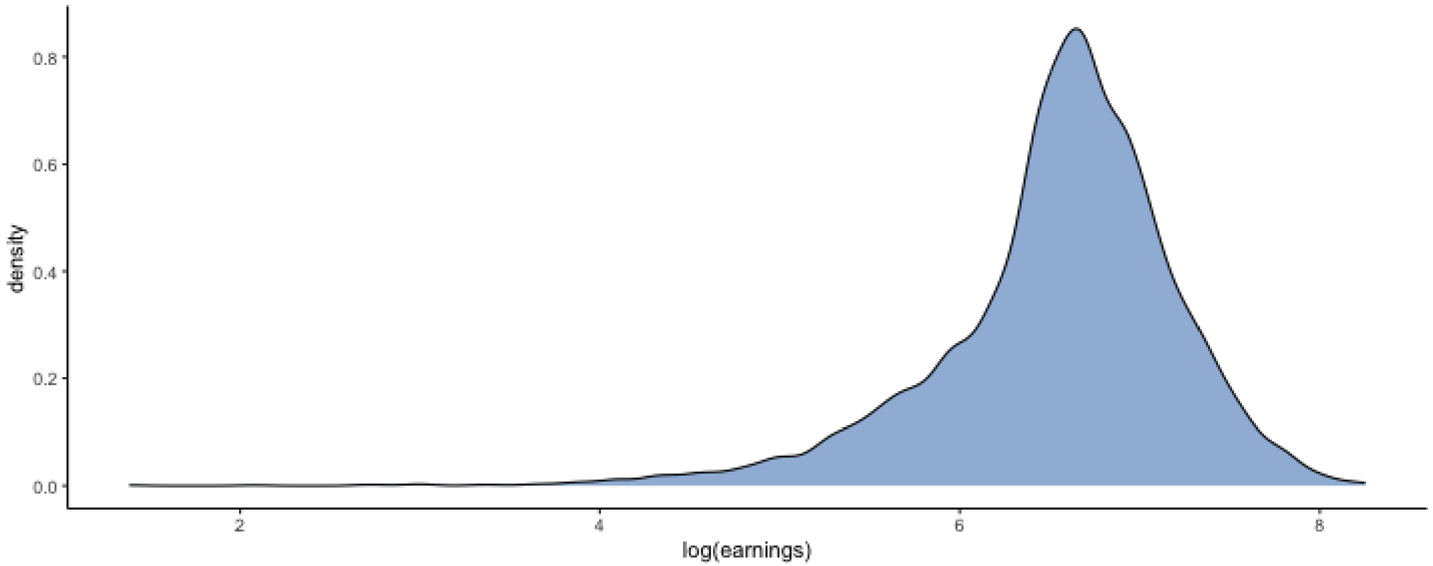
The distribution is likely log-normal!

(when we take its logarithm it looks like a normal distribution)

GETTING TO THE *RIGHT* MODEL

Variable transformation

```
ggplot(cps, aes(x = log(earnings))) +
  geom_density()
```



The log-transformation is very popular in economics:

- Several important economic variables (like wages, income, firm size, GDP) are approximately **log-normally distributed**. A normal distribution has desirable properties for our regression.
- Reduces the impact of outliers
- Allow for convenient interpretations in terms of **percentage changes** of the outcome variable

GETTING TO THE *RIGHT* MODEL

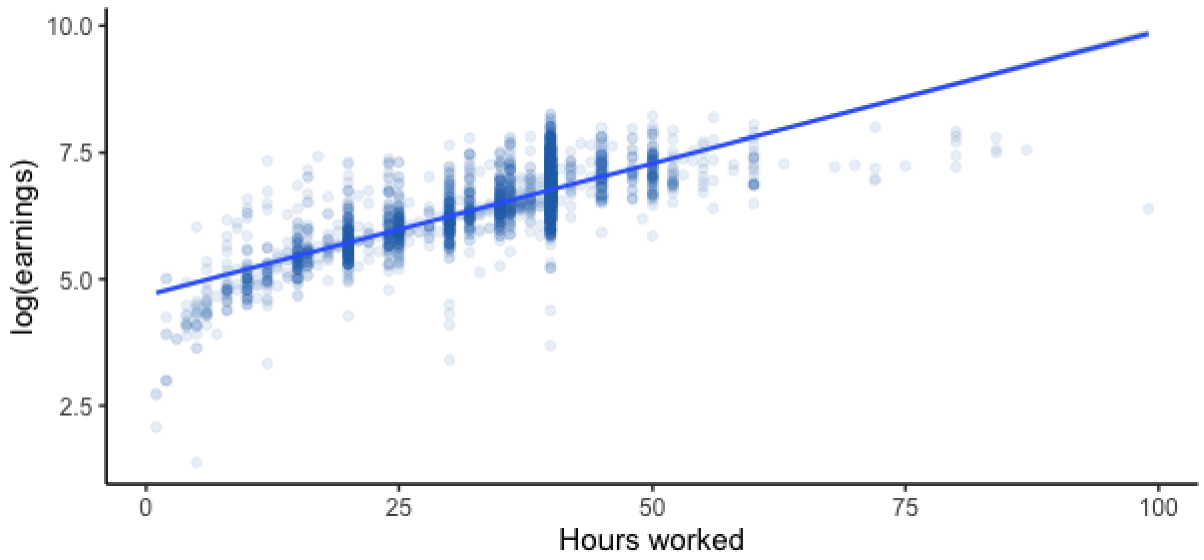
Variable transformation

Specification	Outcome var	Regressor	Interpretation of β	Name
Level-level	y	x	$\Delta y = \beta \Delta x$	Standard
Level-log	y	$log(x)$	$\Delta y = \frac{\beta}{100} \Delta x$	Less common
Log-level	$log(y)$	x	$\% \Delta y = (100\beta) \Delta x$	Semi-elasticity
Log-log	$log(y)$	$log(x)$	$\% \Delta y = \% \Delta \beta x$	Elasticity

GETTING TO THE *RIGHT* MODEL

Variable transformation

```
cps %>%
  ggplot(aes(x = hours, y = log(earnings) )) +
  geom_point() +
  geom_smooth(method = "lm")
```

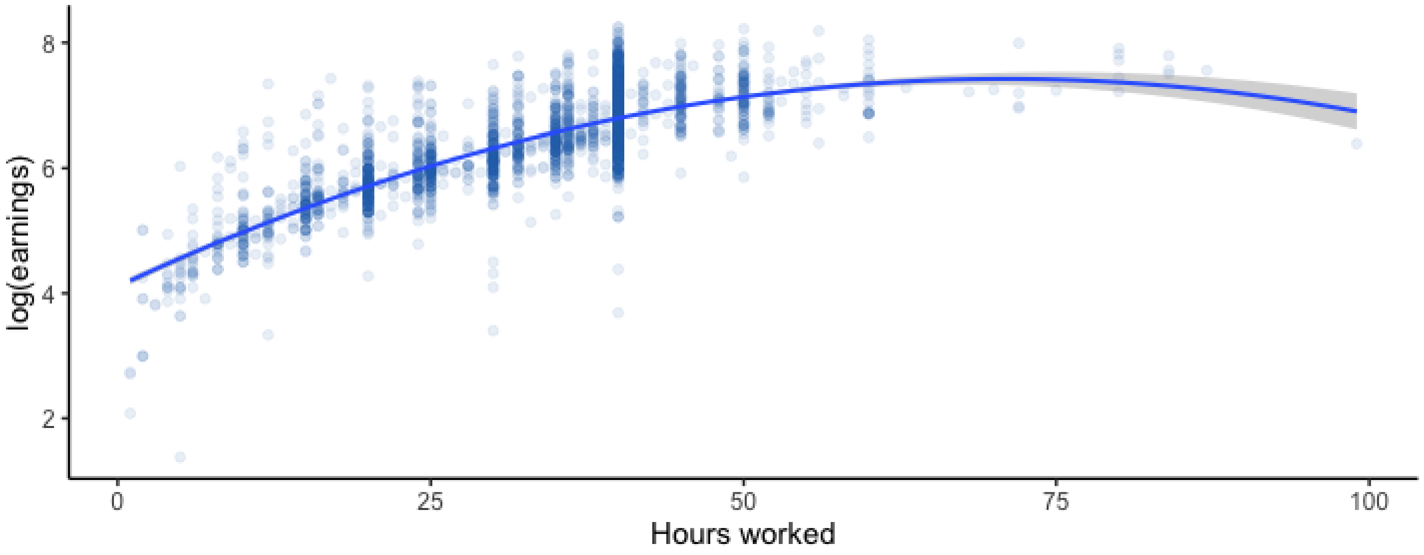


- Better! But the fit looks weird for low and high numbers of hours

GETTING TO THE *RIGHT* MODEL

FUNCTIONAL FORM

```
cps %>%
  ggplot(aes(x = hours, y = log(earnings) )) +
  geom_point() +
  geom_smooth(method = "lm",
              formula = y ~ poly(x, 2))
```



GETTING TO THE *RIGHT* MODEL

FUNCTIONAL FORM

We can rewrite our model:

$$log(Earnings_i) = \alpha + \beta_i Hours_i + \beta_2 Hours_i^2 + \varepsilon_i$$

- Create the necessary variables:

```
cps <- cps %>%
  mutate(logearnings = log(earnings),
         sqhours = hours^2)
```

- Run the new model:

```
lm(logearnings ~ hours + sqhours, cps )
```

```
##
## Call:
## lm(formula = logearnings ~ hours + sqhours, data = cps)
##
## Coefficients:
## (Intercept)        hours      sqhours
##   4.1087277    0.0934110   -0.0006587
```

GETTING TO THE *RIGHT* MODEL

FUNCTIONAL FORM Are we missing something?

- This relationship might be **driven** by **something else**. People who work more hours are different than people who work less hours in ways that also affect earnings. **How is this called?**

For example:

- Men tend to work **full time more often** and **earn more**
- Higher hours are highly correlated with being a man, being a man is also correlated with higher wages

```
cps <-
  cps %>%
  mutate(male =
           as.numeric(sex == "Male"))

lm(hours ~ male, cps)
```

```
## (Intercept)        male
##   34.013288    3.527574
```

We want to control for this in the regression:

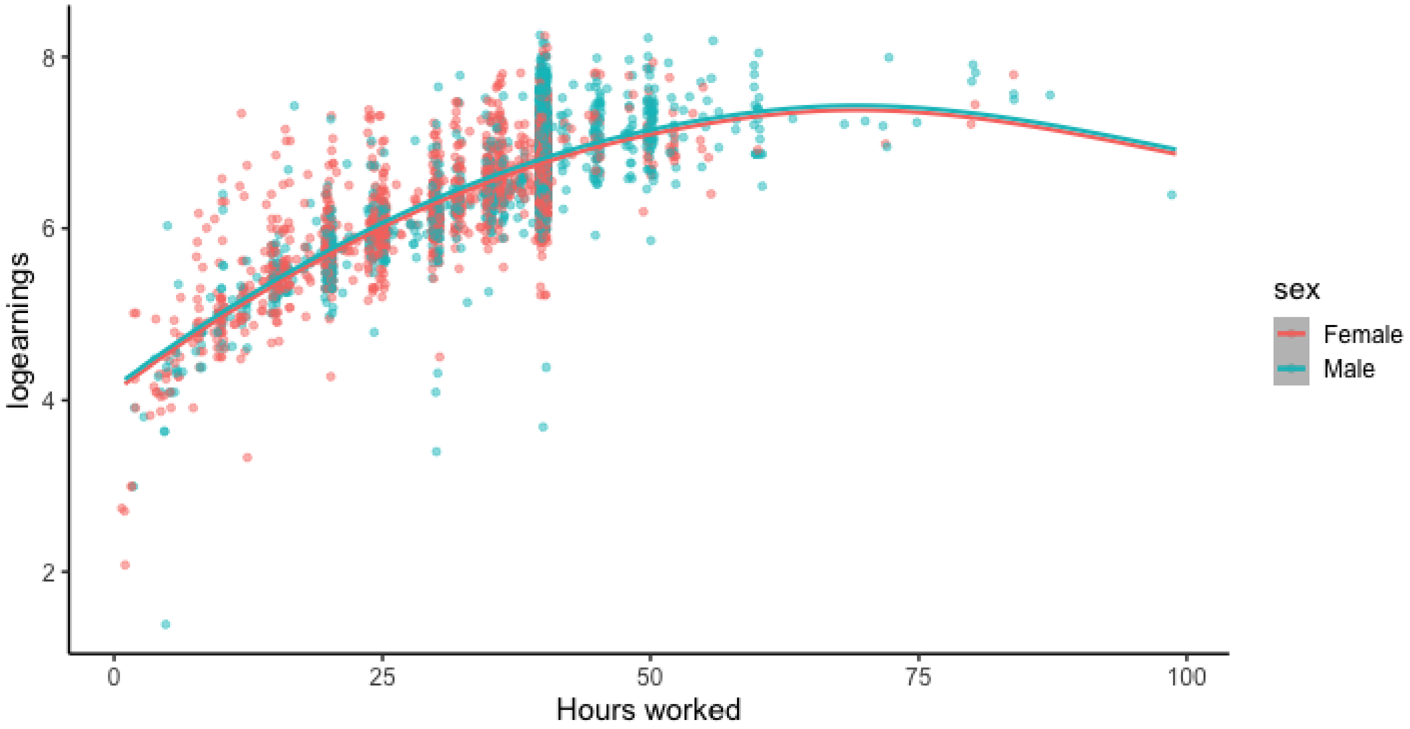
$$log(Earnings_i) = \alpha + \beta_1 Hours_i + \beta_2 Hours_i^2 + \beta_3 Male_i + \varepsilon_i$$

	Male = 0	Male = 1
Intercept	α	$\alpha + \beta_3$
Slope	$\beta_1 + 2\beta_2 Hours$	$\beta_1 + 2\beta_2 Hours$

-> Men and women have the **same slope** but **different intercepts**

GETTING TO THE *RIGHT* MODEL

CONTROL VARIABLES



GETTING TO THE *RIGHT* MODEL

CONTROL VARIABLES

How do the coefficients of both models compare?

```
lm(logearnings ~ hours + sqhours + male, cps)
```

```
##   (Intercept)         hours       sqhours          male
##  4.0945625390  0.0933176308 -0.0006649689  0.0528821933
```

```
lm(logearnings ~ hours + sqhours, cps)
```

```
##   (Intercept)         hours       sqhours
##  4.1087276823  0.0934110198 -0.0006587048
```

The omitted variable bias was not too big in this case

- Beware of violently adding controls (more on this in Econometrics 3 for a deep dive on identification).
- A Crash Course in Good and Bad Controls

GETTING TO THE *RIGHT* MODEL

INTERACTION TERMS

We only allowed the **intercept** to vary by gender. What if we want the **slope** to vary too?

- The relationship between hours and earnings might be heterogeneous between **men** and **women**
- We allow β to vary by gender with an interaction term

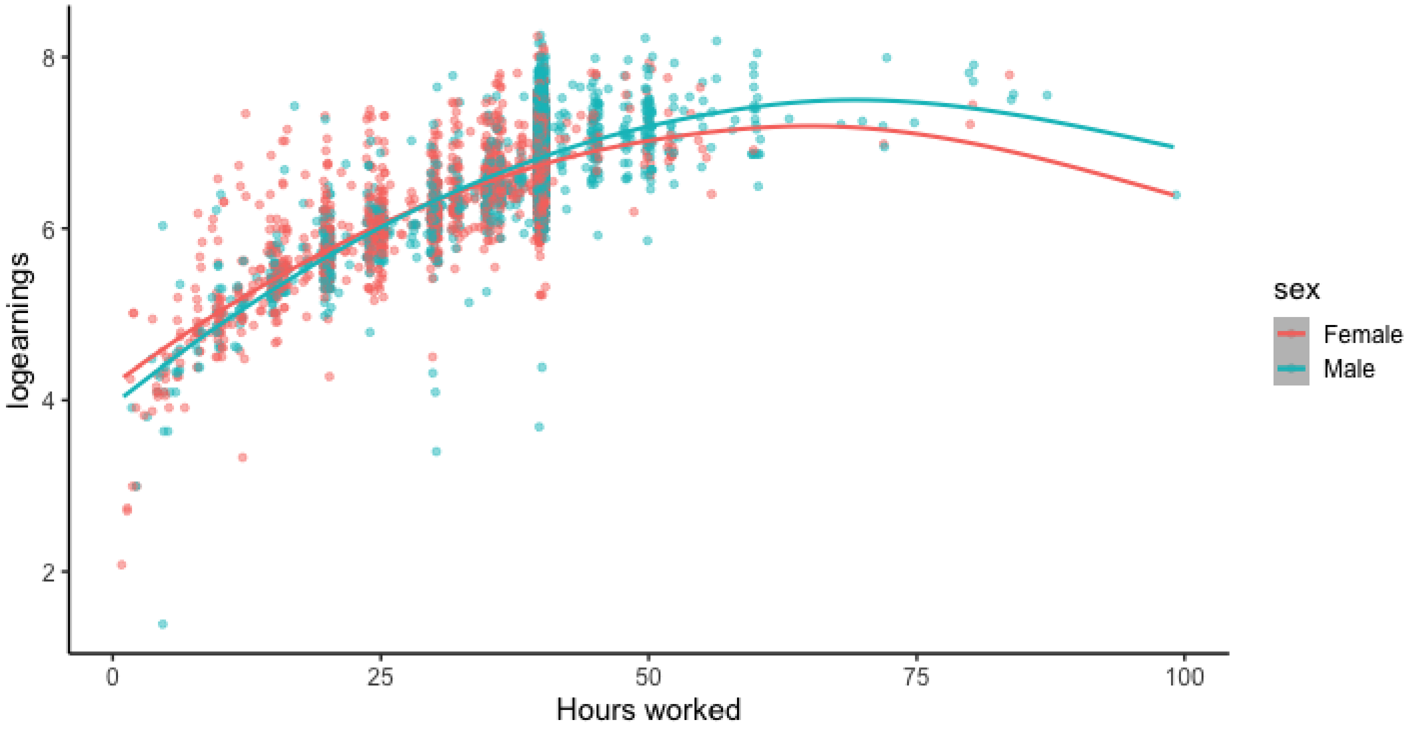
$$log(Earnings_i) = \alpha + \beta_1 Hours_i + \beta_2 Hours_i^2 + \beta_3 Male_i + \beta_4 Hours_i \times Male_i + \varepsilon_i$$

	Male = 0	Male = 1
Intercept	α	$\alpha + \beta_3$
Slope	$\beta_1 + 2\beta_2 Hours$	$\beta_1 + 2\beta_2 Hours + \beta_4$

-> Men and women have **different slopes** and **different intercepts**

GETTING TO THE *RIGHT* MODEL

INTERACTION TERMS



GETTING TO THE *RIGHT* MODEL

INTERACTION TERMS

We specify interaction terms with a `*`

```
mod <- lm(logearnings ~ hours + sqhours + male + male*hours, cps)
results <- summary(mod)$coefficients
```

```
##                 Estimate    Std. Error    t value     Pr(>|t|)
## (Intercept)  4.1714631273 0.0378575672 110.188357 0.0000000000
## hours        0.0930787301 0.0020751060  44.854928 0.0000000000
## sqhours     -0.0007196841 0.0000329061 -21.870832 0.0000000000
## male        -0.2348248075 0.0482515339  -4.866681 0.0000011728
## hours:male   0.0080331447 0.0013030271   6.164987 0.0000000008
```

What can we conclude?

AGENDA

- REGRESSIONS IN R
  - CONTINUOUS VARIABLES
  - CATEGORICAL AND BINARY VARIABLES
- GETTING TO THE RIGHT MODEL
  - VARIABLE TRANSFORMATION
  - FUNCTIONAL FORM
  - CONTROL VARIABLES
  - INTERACTION TERMS
- INFERENCE
- EXPORTING RESULTS
  - REGRESSION TABLES
  - COEFFICIENT PLOTS
- PRACTICING YOUR SKILLS

INFERENCE

HYPOTHESIS TESTING

- Let's assume that we know that $\hat{\beta}_1$ is equal to **.10** using the previous month data. Are we sure that $\hat{\beta}_1$ is actually below .10?
- How can we test whether or not our test **is below from** .10?

We want to use our sample estimates to conclude something about the population parameters!

Concept	Description	Example
Null hypothesis	The hypothes to evaluate	$H_0 : \beta_1 \geq .10$
Alternative hypothesis	The statement if the value differs from the null hypothesis	$H_1 : \beta_1 < .10$
Test statistic	The tool (point estimate statistic formula) we use to decide if we reject or not the null hypothesis	$t = \frac{\hat{\beta} - .10}{s.e.(\hat{\beta})}$
Null distribution	The sampling distribution of the test statistic *assuming* the null H_0 is true	t follows a Student's t-distribution
P-value	The prob. of obtaining a test statistic just as extreme or more extreme than the observed test statistic assuming the null is true. *How surprised am I of oberving t assuming H_0 holds?*	
Significance level	A cutof on the p-value: if the p-value does not fall below α we would “fail to reject H_0	$\alpha = 0.05$

INFERENCE

HYPOTHESIS TESTING

1. State the null and alternative hypotheses $H_0 : \beta_1 \geq .10, \ H_1 : \beta_1 < .10$ 2. Choose a test and significance level

- How many parameters do we have? (one = one-sample test, two = two-sample test)
- Do we know the population variance? (yes = z-test, no = t-test)

   3. Compute the observed test statistic: Computing the t-stat $t = \frac{\hat{\beta} - .10}{s.e.(\hat{\beta})}$

```r
round(results)
```

```
##              Estimate Std. Error   t value Pr(>|t|)
## (Intercept)   4.17146    0.03786 110.18836        0
## hours         0.09308    0.00208  44.85493        0
## sqhours      -0.00072    0.00003 -21.87083        0
## male         -0.23482    0.04825  -4.86668        0
## hours:male    0.00803    0.00130   6.16499        0
```

```r
t <- (results[2,1] - .10) / results[2,2]
t
```

```
## [1] -3.335381
```

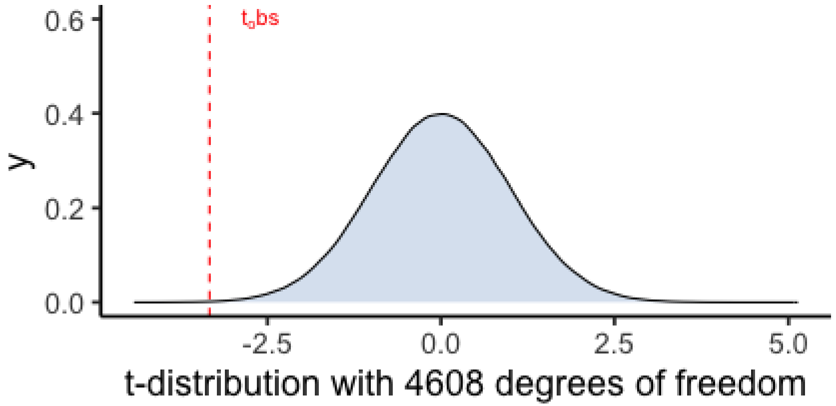
INFERENCE

HYPOTHESIS TESTING

Our observed test statistic provides a measure of “evidence” against the null hypothesis. In particular, we know that under the null hypothesis, the test statistic follows a $t_{df} = t_{n-1}$ = $t_4 608$ distribution

```
df <- nrow(cps) - 1
df
```

```
## [1] 4608
```



- This distribution represents the distribution of sample evidence given that the null is true
- Our observed test statistic (the dashed red line) shows that the event we observed is unlikely to occur if the null ($H_0 : \beta_1 \geq .10$) is true. Now we want to calculate this probability more formally.

INFERENCE

HYPOTHESIS TESTING

4. Compute the p-value The p-value is the probability of getting sample evidence as or more extreme than what we actually observed given that the null hypothesis is actually true.

Since we are working with a "smaller-than" alternative hypothesis: $$\text{p-value} = P(t*{df}* < *t*{obs} | H_0 \text{ is true}) $$

```r
pt(t, df = df, lower.tail = TRUE)
```

```
## [1] 0.0004292814
```

5. Make a statistical decision and interpret the results:

$$\text{if p-value } \leq \alpha \text{ reject } H_0$$

$$\text{if p-value } > \alpha \text{ fail to reject } H_0$$

- Since $\alpha$is the maximum p-value at which we reject H_0, then we are ensuring that there is at most a $100 \times \alpha\%$ chance of making a type I error (reject the null when it is true): a 5% chance of mistakenly deciding that $\beta_1 \geq .10$ when $\beta_1 < .10$

INFERENCE

HYPOTHESIS TESTING

We can use the `linearHypothesis()` function from the `{car}` package to carry out one and two-sample t-tests in R.

- The first argument is the **model**
- The second is the **null hypothesis**

```
## Linear hypothesis test
##
## Hypothesis:
## male = 0
##
## Model 1: restricted model
## Model 2: logearnings ~ hours + sqhours + male + hours * male
##
##   Res.Df    RSS Df Sum of Sq      F      Pr(>F)
## 1   4605 773.82
## 2   4604 769.86  1    3.9604 23.685 0.000001173 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

INFERENCE

HYPOTHESIS TESTING

We can also use it to carry out **F-tests** for joint significance

```
linearHypothesis(
  lm(logearnings ~ hours + sqhours + male + hours*male, cps),
       c("hours = 0", "sqhours = 0", "male = 0", "hours:male = 0"))
```

```
## Linear hypothesis test
##
## Hypothesis:
## hours = 0
## sqhours = 0
## male = 0
## hours:male = 0
##
## Model 1: restricted model
## Model 2: logearnings ~ hours + sqhours + male + hours * male
##
##   Res.Df     RSS Df Sum of Sq    F                Pr(>F)
## 1   4608 2056.79
## 2   4604  769.86  4    1286.9 1924 < 0.00000000000000022 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
mod <- lm(logearnings ~ hours + sqhours + male + hours*male, cps)
```

AGENDA

- REGRESSIONS IN R
  - CONTINUOUS VARIABLES
  - CATEGORICAL AND BINARY VARIABLES
- GETTING TO THE RIGHT MODEL
  - VARIABLE TRANSFORMATION
  - FUNCTIONAL FORM
  - CONTROL VARIABLES
  - INTERACTION TERMS
- INFERENCE
- EXPORTING RESULTS
  - REGRESSION TABLES
  - COEFFICIENT PLOTS
- PRACTICING YOUR SKILLS

EXPORTING RESULTS

Eventually, we want to format our regression results into tables to include in our **papers** and **reports**. In general, they look like this:

Table 2
Labor market states of prime-age men and women.

	Men			Women		
	Participate	Emp/Pop	Unem/Pop	Participate	Emp/Pop	Unem/Pop
Prescrip. Rate	−0.053***	−0.055***	0.003*	−0.010***	−0.011**	0.002
	(0.005)	(0.006)	(0.002)	(0.003)	(0.003)	(0.001)
Demand Shock	0.317**	0.689***	−0.311***	−0.456***	−0.202*	−0.210***
	(0.105)	(0.117)	(0.057)	(0.093)	(0.098)	(0.051)
2000 Particip.	0.637***			0.409***		
	(0.050)			(0.029)		
2000 Emp/Pop		0.517***			0.349***	
		(0.036)			(0.021)	
2000 Unem/Pop			0.263***			0.169***
			(0.027)			(0.021)
R-sqr	0.09	0.11	0.02	0.06	0.06	0.02
N	6424995	6424995	6424995	6641288	6641288	6641288

All regressions include demographic variables, year, and state fixed effects.
Robust standard errors with clustering on coumas.
* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$.

EXPORTING RESULTS

There are many different packages to produce regression tables:
`{stargazer}` `{gtsummary}` `{huxtable}` `{fixest}` `{modelsummary}`

We'll work with the `huxreg()` function from the `{huxtable}` package. Main arguments:

- Many **models**, with a name or automatically numbered
-
-
-
-
-

```
outreg <- huxreg(Baseline = lm(logearnings ~ hours, cps),
                 lm(logearnings ~ hours + sqhours, cps),
                 lm(logearnings ~ hours + sqhours + male, cps),
                 lm(logearnings ~ hours + sqhours + male + male*hours, cps),
#
#
#
#
#
```

EXPORTING RESULTS

There are many different packages to produce regression tables:
`{stargazer}` `{gtsummary}` `{huxtable}` `{fixest}` `{modelsummary}`

We'll work with the `huxreg()` function from the `{huxtable}` package. Main arguments:

- Many **models**, with a name or automatically numbered
- Which **uncertainty statistic** to display (std.error, p.value, conf.low, conf.high)
-
-
-
-

```
outreg <- huxreg(Baseline = lm(logearnings ~ hours, cps),
                 lm(logearnings ~ hours + sqhours, cps),
                 lm(logearnings ~ hours + sqhours + male, cps),
                 lm(logearnings ~ hours + sqhours + male + male*hours, cps),
                 error_format = "({std.error})",
#
#
#
#
```

EXPORTING RESULTS

There are many different packages to produce regression tables:
`{stargazer}` `{gtsummary}` `{huxtable}` `{fixest}` `{modelsummary}`

We'll work with the `huxreg()` function from the `{huxtable}` package. Main arguments:

- Many **models**, with a name or automatically numbered
- Which **uncertainty statistic** to display (std.error, p.value, conf.low, conf.high)
- Where to **place** the uncertainty statistic (below, same, right),
-
-
-

```
outreg <- huxreg(Baseline = lm(logearnings ~ hours, cps),
                 lm(logearnings ~ hours + sqhours, cps),
                 lm(logearnings ~ hours + sqhours + male, cps),
                 lm(logearnings ~ hours + sqhours + male + male*hours, cps),
                 error_format = "({std.error})",
                 error_pos = "below",
#
#
#
```

EXPORTING RESULTS

There are many different packages to produce regression tables:
`{stargazer}` `{gtsummary}` `{huxtable}` `{fixest}` `{modelsummary}`

We'll work with the `huxreg()` function from the `{huxtable}` package. Main arguments:

- Many **models**, with a name or automatically numbered
- Which **uncertainty statistic** to display (std.error, p.value, conf.low, conf.high)
- Where to **place** the uncertainty statistic (below, same, right),
- Which **general statistics** to display (adj.r.squared, df, ...)
-
-

```
outreg <- huxreg(Baseline = lm(logearnings ~ hours, cps),
                lm(logearnings ~ hours + sqhours, cps),
                lm(logearnings ~ hours + sqhours + male, cps),
                lm(logearnings ~ hours + sqhours + male + male*hours, cps),
                error_format = "({std.error})",
                error_pos = "below",
                statistics = c(N = "nobs", R2 = "r.squared"),
#
#
```

EXPORTING RESULTS

There are many different packages to produce regression tables:
`{stargazer}` `{gtsummary}` `{huxtable}` `{fixest}` `{modelsummary}`

We'll work with the `huxreg()` function from the `{huxtable}` package. Main arguments:

- Many **models**, with a name or automatically numbered
- Which **uncertainty statistic** to display (std.error, p.value, conf.low, conf.high)
- Where to **place** the uncertainty statistic (below, same, right),
- Which **general statistics** to display (adj.r.squared, df, ...)
- The designed **significance symbols**
-

```
outreg <- huxreg(Baseline = lm(logearnings ~ hours, cps),
                 lm(logearnings ~ hours + sqhours, cps),
                 lm(logearnings ~ hours + sqhours + male, cps),
                 lm(logearnings ~ hours + sqhours + male + male*hours, cps),
                 error_format = "({std.error})",
                 error_pos = "below",
                 statistics = c(N = "nobs", R2 = "r.squared"),
                 stars = c(`***` = 0.01, `**` = 0.05, `*` = 0.1),
#
```

EXPORTING RESULTS

There are many different packages to produce regression tables:
`{stargazer}` `{gtsummary}` `{huxtable}` `{fixest}` `{modelsummary}`

We'll work with the `huxreg()` function from the `{huxtable}` package. Main arguments:

- Many **models**, with a name or automatically numbered
- Which **uncertainty statistic** to display (std.error, p.value, conf.low, conf.high)
- Where to **place** the uncertainty statistic (below, same, right),
- Which **general statistics** to display (adj.r.squared, df, ...)
- The designed **significance symbols**
- What to write in the **footnote**

```
outreg <- huxreg(Baseline = lm(logearnings ~ hours, cps),
                 lm(logearnings ~ hours + sqhours, cps),
                 lm(logearnings ~ hours + sqhours + male, cps),
                 lm(logearnings ~ hours + sqhours + male + male*hours, cps),
                 error_format = "({std.error})",
                 error_pos = "below",
                 statistics = c(N = "nobs", R2 = "r.squared"),
                 stars = c(`***` = 0.01, `**` = 0.05, `*` = 0.1),
                 note = "Dependent variable: log weekly earnings. {stars}")
```

EXPORTING RESULTS

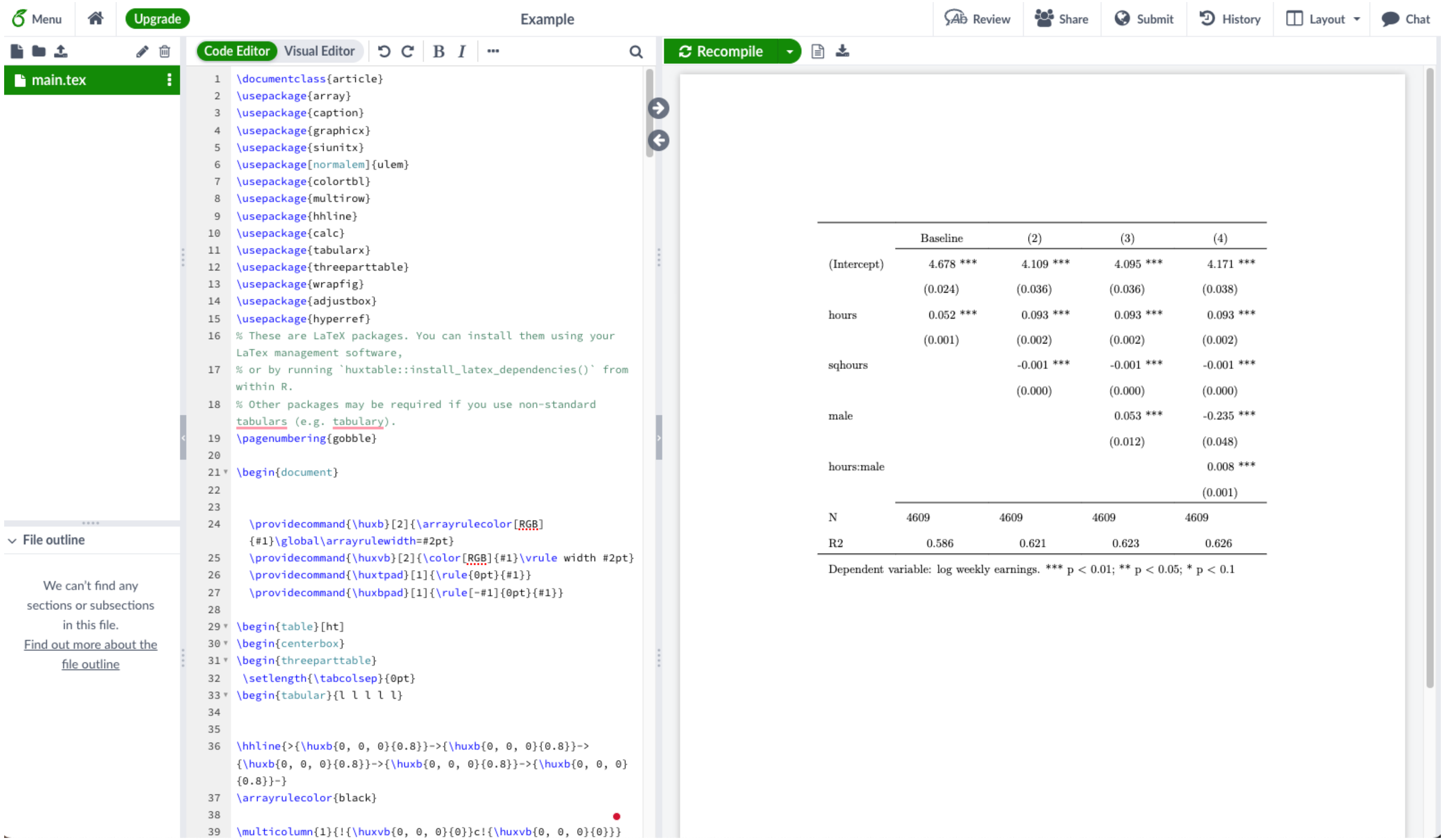
We get somthing like this in the console:

```
─────────────────────────────────────────────────────────────────
               Baseline      (2)           (3)           (4)
          ───────────────────────────────────────────────────────
(Intercept)      4.678 ***     4.109 ***     4.095 ***     4.171 ***
                (0.024)       (0.036)       (0.036)       (0.038)
hours            0.052 ***     0.093 ***     0.093 ***     0.093 ***
                (0.001)       (0.002)       (0.002)       (0.002)
sqhours                       -0.001 ***    -0.001 ***    -0.001 ***
                              (0.000)       (0.000)       (0.000)
male                                         0.053 ***    -0.235 ***
                                            (0.012)       (0.048)
hours:male                                                 0.008 ***
                                                          (0.001)
          ───────────────────────────────────────────────────────
N             4609          4609          4609          4609
R2               0.586         0.621         0.623         0.626
─────────────────────────────────────────────────────────────────
Dependent variable: log weekly earnings.  *** p < 0.01; ** p < 0.05; * p <
0.1
```

- We can export it with the functions: `quick_latex()`, `quick_html()`, `quick_pdf()`, `quick_docx()`, `quick_xlsx()`,

EXPORTING RESULTS

```r
quick_latex(outreg, file = "./inputs/04_regtable.tex")
```




```latex
\documentclass{article}
\usepackage{array}
\usepackage{caption}
\usepackage{graphicx}
\usepackage{siunitx}
\usepackage[normalem]{ulem}
\usepackage{colortbl}
\usepackage{multirow}
\usepackage{hhline}
\usepackage{calc}
\usepackage{tabularx}
\usepackage{threeparttable}
\usepackage{wrapfig}
\usepackage{adjustbox}
\usepackage{hyperref}
% These are LaTeX packages. You can install them using your LaTex management software,
% or by running `huxtable::install_latex_dependencies()` from within R.
% Other packages may be required if you use non-standard tabulars (e.g. tabulary).
\pagenumbering{gobble}

\begin{document}


 \providecommand{\huxb}[2]{\arrayrulecolor[RGB]{#1}\global\arrayrulewidth=#2pt}
 \providecommand{\huxvb}[2]{\color[RGB]{#1}\vrule width #2pt}
 \providecommand{\huxtpad}[1]{\rule{0pt}{#1}}
 \providecommand{\huxbpad}[1]{\rule[-#1]{0pt}{#1}}

\begin{table}[ht]
\begin{centerbox}
\begin{threeparttable}
 \setlength{\tabcolsep}{0pt}
\begin{tabular}{l l l l l}


\hhline{>{\huxb{0, 0, 0}{0.8}}->{\huxb{0, 0, 0}{0.8}}->{\huxb{0, 0, 0}{0.8}}->{\huxb{0, 0, 0}{0.8}}->{\huxb{0, 0, 0}{0.8}}-}
\arrayrulecolor{black}

\multicolumn{1}{!{\huxvb{0, 0, 0}{0}}c!{\huxvb{0, 0, 0}{0}}}
```

	Baseline	(2)	(3)	(4)
(Intercept)	4.678 ***	4.109 ***	4.095 ***	4.171 ***
	(0.024)	(0.036)	(0.036)	(0.038)
hours	0.052 ***	0.093 ***	0.093 ***	0.093 ***
	(0.001)	(0.002)	(0.002)	(0.002)
sqhours		-0.001 ***	-0.001 ***	-0.001 ***
		(0.000)	(0.000)	(0.000)
male			0.053 ***	-0.235 ***
			(0.012)	(0.048)
hours:male				0.008 ***
				(0.001)
N	4609	4609	4609	4609
R2	0.586	0.621	0.623	0.626

Dependent variable: log weekly earnings. *** p < 0.01; ** p < 0.05; * p < 0.1

EXPORTING RESULTS

	Baseline	(2)	(3)	(4)
(Intercept)	4.678 ***	4.109 ***	4.095 ***	4.171 ***
	(0.024)	(0.036)	(0.036)	(0.038)
hours	0.052 ***	0.093 ***	0.093 ***	0.093 ***
	(0.001)	(0.002)	(0.002)	(0.002)
sqhours		-0.001 ***	-0.001 ***	-0.001 ***
		(0.000)	(0.000)	(0.000)
male			0.053 ***	-0.235 ***
			(0.012)	(0.048)
hours:male				0.008 ***
				(0.001)
N	4609	4609	4609	4609
R2	0.586	0.621	0.623	0.626

Dependent variable: log weekly earnings. *** p < 0.01; ** p < 0.05; * p < 0.1

PRACTICE

Use the functions `huxreg()`, `insert_row()` and `merge_cells()` to reproduce this table and export it to html.

	Dependent variable: Log weekly earnings			
	(1)	(2)	(3)	(4)
Hours worked	0.052 ***	0.093 ***	0.093 ***	0.093 ***
	(0.000)	(0.000)	(0.000)	(0.000)
(Hours worked)^2		-0.001 ***	-0.001 ***	-0.001 ***
		(0.000)	(0.000)	(0.000)
Male			0.053 ***	-0.235 ***
			(0.000)	(0.000)
Hours worked x Male				0.008 ***
				(0.000)
Constant	4.678 ***	4.109 ***	4.095 ***	4.171 ***
	(0.000)	(0.000)	(0.000)	(0.000)
N	4609	4609	4609	4609
R2 adj.	0.586	0.621	0.622	0.625

P-values in parentheses. *** p < 0.01; ** p < 0.05; * p < 0.1

10:00

PRACTICE

```
huxreg(lm(logearnings ~ hours, cps),
       lm(logearnings ~ hours + sqhours, cps),
       lm(logearnings ~ hours + sqhours + male, cps),
       lm(logearnings ~ hours + sqhours + male + male*hours, cps),
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
```

PRACTICE

```
huxreg(lm(logearnings ~ hours, cps),
       lm(logearnings ~ hours + sqhours, cps),
       lm(logearnings ~ hours + sqhours + male, cps),
       lm(logearnings ~ hours + sqhours + male + male*hours, cps),
       error_format = "({p.value})",
       error_pos = "below",
#
#
#
#
#
#
#
#
#
#
#
#
#
#
#
```

PRACTICE

```
huxreg(lm(logearnings ~ hours, cps),
       lm(logearnings ~ hours + sqhours, cps),
       lm(logearnings ~ hours + sqhours + male, cps),
       lm(logearnings ~ hours + sqhours + male + male*hours, cps),
       error_format = "({p.value})",
       error_pos = "below",
       statistics = c(N = "nobs", "R2 adj." = "adj.r.squared"),
       stars = c(`***` = 0.01, `**` = 0.05, `*` = 0.1),
#
#
#
#
#
#
#
#
#
#
#
#
#
```

PRACTICE

```
huxreg(lm(logearnings ~ hours, cps),
       lm(logearnings ~ hours + sqhours, cps),
       lm(logearnings ~ hours + sqhours + male, cps),
       lm(logearnings ~ hours + sqhours + male + male*hours, cps),
       error_format = "({p.value})",
       error_pos = "below",
       statistics = c(N = "nobs", "R2 adj." = "adj.r.squared"),
       stars = c(`***` = 0.01, `**` = 0.05, `*` = 0.1),
        coefs = c("Hours worked" = "hours",
                  "(Hours worked)²" = "sqhours",
                  "Male" = "male",
                  "Hours worked x Male" = "hours:male",
                  "Constant" = "(Intercept)"),
#
#
#
#
#
#
#
#
```

PRACTICE

```
huxreg(lm(logearnings ~ hours, cps),
       lm(logearnings ~ hours + sqhours, cps),
       lm(logearnings ~ hours + sqhours + male, cps),
       lm(logearnings ~ hours + sqhours + male + male*hours, cps),
       error_format = "({p.value})",
       error_pos = "below",
       statistics = c(N = "nobs", "R2 adj." = "adj.r.squared"),
       stars = c(`***` = 0.01, `**` = 0.05, `*` = 0.1),
        coefs = c("Hours worked" = "hours",
                  "(Hours worked)²" = "sqhours",
                  "Male" = "male",
                  "Hours worked x Male" = "hours:male",
                  "Constant" = "(Intercept)"),
       note = "P-values in parentheses. {stars}",
       align = "c") %>%
#
#
#
#
#
#
```

PRACTICE

```
huxreg(lm(logearnings ~ hours, cps),
       lm(logearnings ~ hours + sqhours, cps),
       lm(logearnings ~ hours + sqhours + male, cps),
       lm(logearnings ~ hours + sqhours + male + male*hours, cps),
       error_format = "({p.value})",
       error_pos = "below",
       statistics = c(N = "nobs", "R2 adj." = "adj.r.squared"),
       stars = c(`***` = 0.01, `**` = 0.05, `*` = 0.1),
        coefs = c("Hours worked" = "hours",
                  "(Hours worked)²" = "sqhours",
                  "Male" = "male",
                  "Hours worked x Male" = "hours:male",
                  "Constant" = "(Intercept)"),
       note = "P-values in parentheses. {stars}",
       align = "c") %>%
  insert_row(c("", "Dependent variable: Log weekly earnings", rep("", 3))) %>%
#
#
#
#
#
```

PRACTICE

```
huxreg(lm(logearnings ~ hours, cps),
       lm(logearnings ~ hours + sqhours, cps),
       lm(logearnings ~ hours + sqhours + male, cps),
       lm(logearnings ~ hours + sqhours + male + male*hours, cps),
       error_format = "({p.value})",
       error_pos = "below",
       statistics = c(N = "nobs", "R2 adj." = "adj.r.squared"),
       stars = c(`***` = 0.01, `**` = 0.05, `*` = 0.1),
        coefs = c("Hours worked" = "hours",
                  "(Hours worked)²" = "sqhours",
                  "Male" = "male",
                  "Hours worked x Male" = "hours:male",
                  "Constant" = "(Intercept)"),
       note = "P-values in parentheses. {stars}",
       align = "c") %>%
  insert_row(c("", "Dependent variable: Log weekly earnings", rep("", 3))) %>%
  merge_cells(1, 2:5) %>%
#
#
#
```

PRACTICE

```
huxreg(lm(logearnings ~ hours, cps),
       lm(logearnings ~ hours + sqhours, cps),
       lm(logearnings ~ hours + sqhours + male, cps),
       lm(logearnings ~ hours + sqhours + male + male*hours, cps),
       error_format = "({p.value})",
       error_pos = "below",
       statistics = c(N = "nobs", "R2 adj." = "adj.r.squared"),
       stars = c(`***` = 0.01, `**` = 0.05, `*` = 0.1),
        coefs = c("Hours worked" = "hours",
                  "(Hours worked)²" = "sqhours",
                  "Male" = "male",
                  "Hours worked x Male" = "hours:male",
                  "Constant" = "(Intercept)"),
       note = "P-values in parentheses. {stars}",
       align = "c") %>%
  insert_row(c("", "Dependent variable: Log weekly earnings", rep("", 3))) %>%
  merge_cells(1, 2:5) %>%
  set_align(1, 2, "center")
#
#
```

PRACTICE

EXPORTING RESULTS

You can do many more things with `huxreg()` and `huxtable()`. See here for more details.

	(1)	(2)	(3)	(4)
Hours worked	0.052 ***	0.093 ***	0.093 ***	0.093 ***
	(0.000)	(0.000)	(0.000)	(0.000)
(Hours worked)²		-0.001 ***	-0.001 ***	-0.001 ***
		(0.000)	(0.000)	(0.000)
Male			0.053 ***	-0.235 ***
			(0.000)	(0.000)
Hours worked x Male				0.008 ***
				(0.000)
Constant	4.678 ***	4.109 ***	4.095 ***	4.171 ***
	(0.000)	(0.000)	(0.000)	(0.000)
N	4609	4609	4609	4609
R2 adj.	0.586	0.621	0.622	0.625

P-values in parentheses. *** p < 0.01; ** p < 0.05; * p < 0.1

EXPORTING RESULTS

PLOT COEFFICIENTS It might be useful to provide a **graphical representation** of the coefficients

- You can do it with **dplyr**, **gplot** and extracting elements from the regression object with `$`

-We can use a **shortcut** using the `plot_summs()` function from the `{jtools}` package. You can:

- Include one or many **models**
-
-
-
-

```
plot_summs(lm(logearnings ~ hours + sqhours, cps),
           lm(logearnings ~ hours + sqhours + male, cps),
           lm(logearnings ~ hours + sqhours + male + educ_hs + educ_assoc + educ_bach, cps),
#
#
#
#
```

EXPORTING RESULTS

PLOT COEFFICIENTS

It might be useful to provide a **graphical representation** of the coefficients

- You can do it with **dplyr**, **gplot** and extracting elements from the regression object with `$`

-We can use a **shortcut** using the `plot_summs()` function from the `{jtools}` package. You can:

- Include one or many **models**
- Specify **coefficients** to omit from the plot
-
-
-

```
plot_summs(lm(logearnings ~ hours + sqhours, cps),
           lm(logearnings ~ hours + sqhours + male, cps),
           lm(logearnings ~ hours + sqhours + male + educ_hs + educ_assoc + educ_bach, cps),
           omit.coefs = "(Intercept)",
#
#
#
```

EXPORTING RESULTS

PLOT COEFFICIENTS

It might be useful to provide a **graphical representation** of the coefficients

- You can do it with **dplyr**, **gplot** and extracting elements from the regression object with `$`

-We can use a **shortcut** using the `plot_summs()` function from the `{jtools}` package. You can:

- Include one or many **models**
- Specify **coefficients** to omit from the plot
- Change the level of the **confidence intervals**
-
-

```
plot_summs(lm(logearnings ~ hours + sqhours, cps),
           lm(logearnings ~ hours + sqhours + male, cps),
           lm(logearnings ~ hours + sqhours + male + educ_hs + educ_assoc + educ_bach, cps),
           omit.coefs = "(Intercept)",
           ci_level = 0.99,
#
#
```

EXPORTING RESULTS

PLOT COEFFICIENTS

It might be useful to provide a **graphical representation** of the coefficients

- You can do it with **dplyr**, **gplot** and extracting elements from the regression object with `$`

-We can use a **shortcut** using the `plot_summs()` function from the `{jtools}` package. You can:

- Include one or many **models**
- Specify **coefficients** to omit from the plot
- Change the level of the **confidence intervals**
- Choose a **color palette**
-

```
plot_summs(lm(logearnings ~ hours + sqhours, cps),
           lm(logearnings ~ hours + sqhours + male, cps),
           lm(logearnings ~ hours + sqhours + male + educ_hs + educ_assoc + educ_bach, cps),
           omit.coefs = "(Intercept)",
           ci_level = 0.99,
           colors = c( "#296EB4","#D84727","#69995D")) +
#
```

EXPORTING RESULTS

PLOT COEFFICIENTS

It might be useful to provide a **graphical representation** of the coefficients

- You can do it with **dplyr**, **gplot** and extracting elements from the regression object with `$`

-We can use a **shortcut** using the `plot_summs()` function from the `{jtools}` package. You can:

- Include one or many **models**
- Specify **coefficients** to omit from the plot
- Change the level of the **confidence intervals**
- Choose a **color palette**
- Other functions from **ggplot** ...

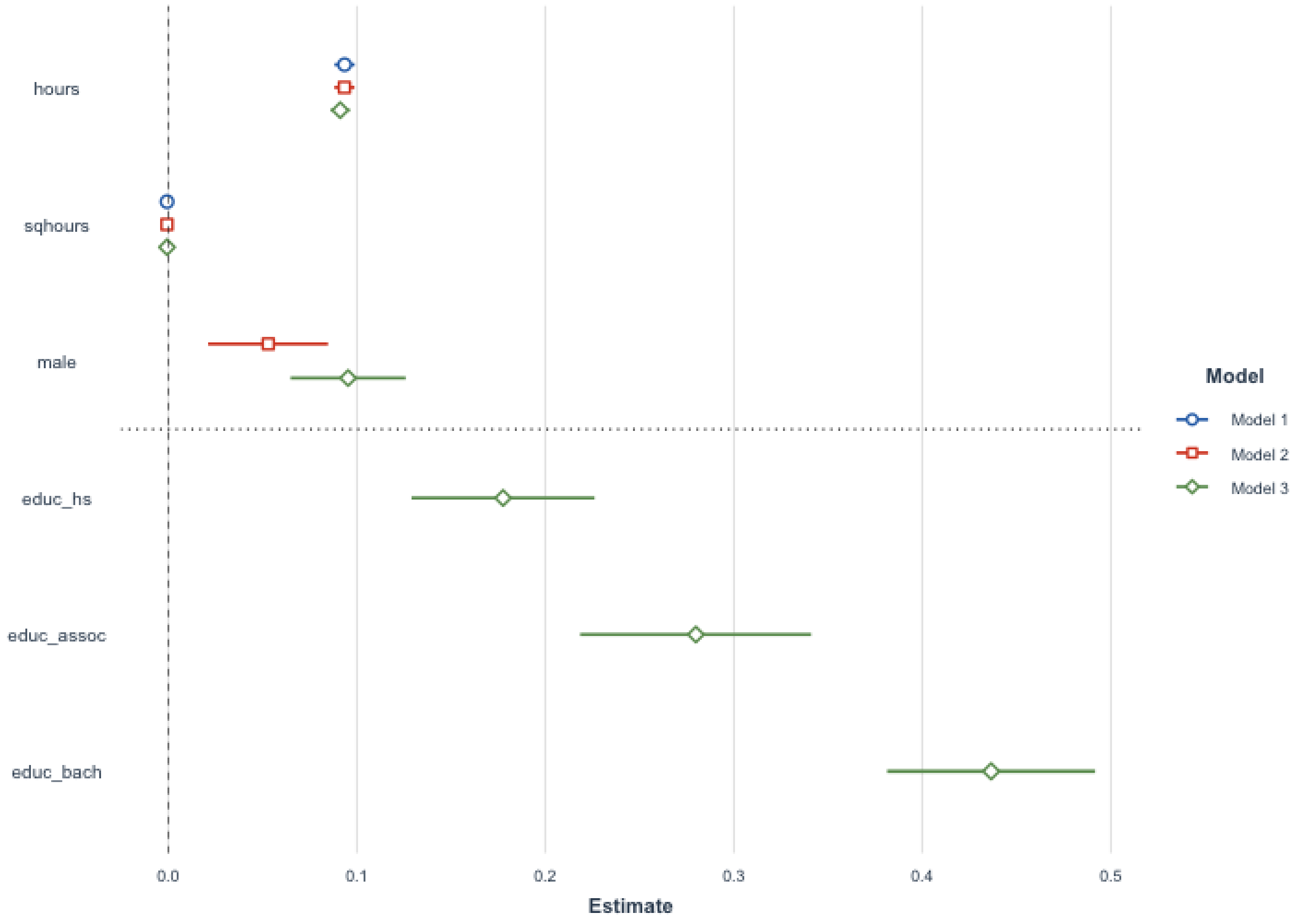
```
plot_summs(lm(logearnings ~ hours + sqhours, cps),
           lm(logearnings ~ hours + sqhours + male, cps),
           lm(logearnings ~ hours + sqhours + male + educ_hs + educ_assoc + educ_bach, cps),
           omit.coefs = "(Intercept)",
           ci_level = 0.99,
           colors = c( "#296EB4","#D84727","#69995D")) +
  geom_hline(yintercept = 3.5, linetype = "dotted")
```

AGENDA

- REGRESSIONS IN R
  - CONTINUOUS VARIABLES
  - CATEGORICAL AND BINARY VARIABLES
- GETTING TO THE RIGHT MODEL
  - VARIABLE TRANSFORMATION
  - FUNCTIONAL FORM
  - CONTROL VARIABLES
  - INTERACTION TERMS
- INFERENCE
- EXPORTING RESULTS
  - REGRESSION TABLES
  - COEFFICIENT PLOTS
- PRACTICING YOUR SKILLS

EXPORTING RESULTS

PLOT COEFFICIENTS



PRACTICING YOUR SKILLS

- How can you work on improve your programming skills throughout the year?
- A research project is too demanding and you're not ready with the basics yet! **Replicate a paper:**- Browse the recent issues (last 3 years) of Economics journals.
    - **General interest:** American Economic Review, The Quarterly Journal of Economics, Econometrica, The Review of Economic Studies, Economic Journal
    - **Fields:** Journal of Political Economy, Journal of Development Economics, Labor Economics, Journal of Public Economics
- Many of the papers now have a replication package with data and code that you can download- We want to ensure reproducibliity and replicability
    - Computational reproducibility: the code runs and produces the same results
    - Replicability: ability to replicate results from scratch
- See here for reports on previous replications by the Institute for Replication and papers that need a replicator.
- Sign-up for an edition of the Replication Games organized by the Institute for Replication.

PRACTICING YOUR SKILLS

Journal of Political Economy > Volume 130, Number 1

PREVIOUS ARTICLE NEXT ARTICLE

Details Figures References Cited by

Cooperative Property Rights and Development: Evidence from Land Reform in El Salvador

Eduardo Montero

Corrections to this article

Erratum: Cooperative Property Rights and Development: Evidence from Land Reform in El Salvador

PDF PDF PLUS Abstract Full Text Supplemental Material

Abstract

In cooperative property rights systems, workers jointly own and manage production, whereas in outside-ownership systems, an owner contracts workers. Despite a rich literature on how the allocation of property rights matters for specialization, efficiency, and equity, little causal evidence exists. During a land reform in El Salvador in 1980, the military government reorganized properties owned by individuals with cumulative landholdings over 500 hectares into cooperatives; properties below this threshold remained as outside-owned properties. Using the discontinuous probability of cooperative formation, I provide evidence on the effects of cooperative property rights relative to outside ownership on specialization, productivity, and worker equity.

Journal of Political Economy
Volume 130, Number 1
January 2022

Article DOI
https://doi.org/10.1086/717042

PRACTICING YOUR SKILLS

Cooperative Property Rights and Development:

Evidence from Land Reform in El Salvador:

A Comment*

Anders Kjelsrud† Andreas Kotsadam‡ Ole Rogeberg§

March, 2023 (updated version, March 10)**

Abstract

Montero (2022) explores a discontinuity in a land reform in El Salvador and reports two main findings. First, relative to outside-owned haciendas operated by contract workers, the productivity of worker-owned cooperatives is higher for staple crops and lower for cash-crop. Second, cooperative property rights increase workers' incomes and compress wage distributions. In this comment, we show that the latter result rests on two mistakes: three-quarters of the observations are duplicates and income inequality is calculated over too few workers to be meaningful. When corrected, the data sources and research design provide no credible evidence regarding the causal effects of ownership structure on income levels and inequality.

PRACTICING YOUR SKILLS

Upcoming Games: Registration Open

Lyon Replication Games: October 24th, 2023 at ENS de Lyon, France. Economics and political science papers to be reproduced/replicated. Local organizer is Mathieu Couttenier.

Stockholm Replication Games: October 26th, 2023 in Stockholm, Sweden. Behavioral science, economics and political science papers to be reproduced/replicated. Local organizer is I4R's co-director Anna Dreber.

Brussels Replication Games: October 27th, 2023 at ULB-ECARES, Belgium. Economics and political science papers to be reproduced/replicated. Local organizers are Paula Gobbi and Joanne Haddad.

Upcoming Games: Registration Closed

None.

Upcoming Games: Registration Not Open Yet

Toronto Replication Games: February 20th 2024 at the University of Toronto, Canada. This is a collaboration with the University of Toronto's Data Sciences Institute. Local organizer is once again Rohan Alexander.

Los Angeles Replication Games: February 28th 2024 at University of California, Los Angeles. Economics and political science papers to be replicated.

Berkeley Replication Games: March 7th 2024 at University of California, Berkeley. Social science papers to be replicated. Local organizers are Edward Miguel and Fernando Fernando Hoces de la Guardia.

Tokyo Replication Games: Date tbd, at the University of Tokyo campus, Japan. Economics and political science papers to be replicated. Local organizers are Yasuyuki Sawada, Chishio Furukawa and Hiroki Kameyama.

Cologne Replication Games: Fall 2023 or Early 2024 (date tbd), Cologne, Germany. Economics

RESOURCES

- Causal Inference: *The Mixtape* by Scott Cunningham.
  Great book on econometrics with a special focus on causal inference (great for us economists!).
  - It has code examples in R, Stata and Python for each chapter.
  - Good for finding the right commands for each method (instrumental variables, regression discontinuity, etc.)

- Mostly Harmless Econometrics by Joshua D. Angrist and Jörn-Steffen Pischke

  - Best book to get "the intuition".
  - Very useful for Econometrics 3

- Introduction to Econometrics with R course by Florian Oswald, Vincent Viers, Jean-Marc Robin, Pierre Villedieu, Gustave Kenedi at Sciences Po

- A crowd-sourced checklist of the top 10 little things that drive us crazy with regression output

  - Things you need to include in your regression tables