Introduction to Regression Analysis 1

Preparing your workfile

We add the basic libraries needed for this week's work:

```
library(tidyverse)     # for almost all data handling tasks
library(readxl)        # to import Excel data
library(ggplot2)       # to produce nice graphiscs
library(stargazer)     # to produce nice results tables
library(haven)         # to import stata file
library(AER)           # access to HS robust standard errors
library(estimatr)      # use robust se
```

Introduction

The data are an extract from the Understanding Society Survey (formerly the British Household Survey Panel).

Data Upload - and understanding data structure

Upload the data, which are saved in a STATA datafile (extension `.dta`). There is a function which loads STATA file. It is called `read_dta` and is supplied by the `haven` package.

```
data_USoc <- read_dta("20222_USoc_extract.dta")
data_USoc <- as.data.frame(data_USoc)     # ensure data frame structure
names(data_USoc)
```

```
##  [1] "pidp"    "age"     "jbhrs"   "paygu"   "wave"    "cpi"     "year"
##  [8] "region"  "urate"   "male"    "race"    "educ"    "degree"  "mfsize9"
```

Let us ensure that categorical variables are stored as `factor` variables. It is easiest to work with these in R.

```
data_USoc$region <- as_factor(data_USoc$region)
data_USoc$male <- as_factor(data_USoc$male)
data_USoc$degree <- as_factor(data_USoc$degree)
data_USoc$race <- as_factor(data_USoc$race)
```

Click on the little table symbol in your environment tab to see the actual data table.

The variable `pidp` contains a unique person identifier and the variable `wave` indicates the wave and `year` the year of observation.

To explain the meaning of these let us just pick out all the observations that pertain to one particular individual (`pidp == 272395767`). The following command does the following in words: "Take `data_USoc` filter/keep all observations which belong to individual pidp == 272395767, then select a list of variables (we don't need to see all 14 variables) and print the result":

```r
data_USoc %>% filter(pidp == 272395767) %>%
              select(c("pidp","male","wave","year","paygu","age","educ")) %>%
              print()
```

```
##        pidp   male wave year    paygu age educ
## 1 272395767 female    1 2009 774.8302  40   11
## 2 272395767 female    2 2010 812.2778  41   11
## 3 272395767 female    3 2011 772.1625  42   11
```

The same person (female) was observed three years in a row (from 2009 to 2011). Their gross monthly income changed, as did, of course, their age, but not their education. This particular person was observed in three consequitive waves. Let's se whether this is a commom pattern.

The code below figures out for how many individuals we have 1, 2 and 3 waves of observations. It is not important to understand that code.

```r
pattern <- data_USoc %>% group_by(pidp) %>%
                      mutate(n_wave = n()) %>%
                      select(pidp,n_wave) %>%
                      unique() %>%
                      group_by(n_wave)%>%
                      summarise(n_pat = n()) %>%
                      print()
```

```
## # A tibble: 3 x 2
##   n_wave n_pat
##    <int> <int>
## 1      1 15099
## 2      2 12450
## 3      3 31091
```

As you can see only just over half of individuals have records for three waves. Let us look at the observations for an individual (`pidp == 2670365`) which only has observations for two waves.

```r
data_USoc %>% filter(pidp == 2670365) %>%
              select(c("pidp","male","wave","year","paygu","age","educ")) %>%
              print()
```

```
##      pidp   male wave year    paygu age educ
## 1 2670365 female    2 2010 230.0000  16   11
## 2 2670365 female    3 2011 346.6667  17   11
```

Some summary statistics

Let's use the `stargazer` function to produce a nice summary table

```r
stargazer(data_USoc, type = "text")
```

```
##
## ===================================================================================================
## Statistic    N         Mean          St. Dev.       Min    Pctl(25)     Pctl(75)          Max
## ---------------------------------------------------------------------------------------------------
## pidp      133,272 839,218,358.000 467,699,610.000 280,165 410,528,927 1,225,328,047 1,639,568,724
## age       133,272     46.172          18.295         9        31           60             103
## jbhrs     64,217      32.594          11.614       0.100    25.000       40.000         97.000
## paygu     59,216     1,823.574       1,475.064     0.083    850.000     2,400.000      15,000.000
```

```
## wave      133,272     1.912          0.818        1        1          3            3
## cpi       133,272    116.790         4.199     110.800   114.500    119.600      126.100
## year      133,272   2,010.453        0.991      2,009     2,010      2,011        2,013
## urate     133,272     7.955          1.311      5.800     6.700      9.100       10.800
## educ      133,041    12.838          2.316     11.000    11.000     15.000       17.000
## mfsize9   58,989     303.135        484.430     1.000    17.000    350.000     1,500.000
## ----------------------------------------------------------------------------------------------------
```

Here are some frequency tables

```
data_USoc %>% count(wave)
```

```
## # A tibble: 3 x 2
##    wave     n
##   <dbl> <int>
## 1     1 50923
## 2     2 43131
## 3     3 39218
```

```
data_USoc %>% count(region)
```

```
## # A tibble: 12 x 2
##    region                       n
##    <fct>                    <int>
##  1 north east                5368
##  2 north west               14095
##  3 yorkshire and the humber 11139
##  4 east midlands            10257
##  5 west midlands            11747
##  6 east of england          11860
##  7 london                   19994
##  8 south east               16461
##  9 south west               10554
## 10 wales                     6676
## 11 scotland                  9321
## 12 northern ireland          5800
```

```
data_USoc %>% count(male)
```

```
## # A tibble: 2 x 2
##   male       n
##   <fct>  <int>
## 1 female 72072
## 2 male   61200
```

```
data_USoc %>% count(year)
```

```
## # A tibble: 5 x 2
##    year     n
##   <dbl> <int>
## 1  2009 25363
## 2  2010 44495
## 3  2011 42213
## 4  2012 20048
## 5  2013  1153
```

```
data_USoc %>% count(race)
```

```
## # A tibble: 6 x 2
##   race      n
##   <fct> <int>
## 1 white 99593
## 2 mixed  2057
## 3 asian 12994
## 4 black  6167
## 5 other  2078
## 6 <NA>  10383
```

```
data_USoc %>% count(educ)
```

```
## # A tibble: 7 x 2
##    educ     n
##   <dbl> <int>
## 1    11 74150
## 2    12  3673
## 3    13 11628
## 4    15 13193
## 5    16 17509
## 6    17 12888
## 7    NA   231
```

```
data_USoc %>% count(degree)
```

```
## # A tibble: 4 x 2
##   degree             n
##   <fct>          <int>
## 1 no degree     102644
## 2 first degree   17509
## 3 higher degree  12888
## 4 <NA>             231
```

```
data_USoc %>% count(mfsize9)
```

```
## # A tibble: 10 x 2
##    mfsize9     n
##      <dbl> <int>
##  1       1  2269
##  2       6  7722
##  3      17  9599
##  4      37  9095
##  5      75  6766
##  6     150  5814
##  7     350  6788
##  8     750  3768
##  9    1500  7168
## 10      NA 74283
```

The pay information (`paygu`) is provided as a measure of the (usual) gross pay per month. We shall adjust for increasing price levels (as measured by `cpi`). We call this variable `hrpay` and also calculate the natural log of this variable (`lnhrpay`). Recall that, in order to change or create a new variable we use the `mutate` function.

```
data_USoc <- data_USoc %>%
              mutate(hrpay = paygu/(jbhrs*4)/(cpi/100)) %>%
              mutate(lnhrpay = log(hrpay))
```

As we want to save these additional variables for further use we assign the result of the operation to `data_USoc`.

Let's check whether the log of the hourly pay differes if we analyse the data by certain criteria.

```
data_USoc %>% group_by(degree) %>%
              summarise(n = sum(!is.na(lnhrpay)),
                        mean = mean(lnhrpay,na.rm=TRUE),
                        sd = sd(lnhrpay,na.rm=TRUE))
```

```
## # A tibble: 4 x 4
##   degree              n  mean    sd
##   <fct>           <int> <dbl> <dbl>
## 1 no degree       40816  2.14 0.581
## 2 first degree    10601  2.57 0.602
## 3 higher degree    7525  2.67 0.637
## 4 <NA>               18  1.90 0.535
```

```
data_USoc %>% group_by(educ) %>%
              summarise(n = sum(!is.na(lnhrpay)),
                        mean = mean(lnhrpay,na.rm=TRUE),
                        sd = sd(lnhrpay,na.rm=TRUE))
```

```
## # A tibble: 7 x 4
##    educ     n  mean    sd
##   <dbl> <int> <dbl> <dbl>
## 1    11 26349  2.09 0.549
## 2    12  1751  2.01 0.593
## 3    13  5651  2.18 0.631
## 4    15  7065  2.33 0.604
## 5    16 10601  2.57 0.602
## 6    17  7525  2.67 0.637
## 7    NA    18  1.90 0.535
```

```
data_USoc %>% group_by(male) %>%
              summarise(n = sum(!is.na(lnhrpay)),
                        mean = mean(lnhrpay,na.rm=TRUE),
                        sd = sd(lnhrpay,na.rm=TRUE))
```

```
## # A tibble: 2 x 4
##   male       n  mean    sd
##   <fct>  <int> <dbl> <dbl>
## 1 female 32609  2.20 0.598
## 2 male   26351  2.38 0.656
```

You can see that difference in the averages for `lnhrpay` is about 0.183.

Testing for differences

The first hypothesis test we may want to implement is to test whether the difference in the raw data is statistically significant (recall this is not the same economically significant difference).

We use the `t.test` function. We could create two subsets of data for `lnhrpay`, one for males and one for females and could then feed these two series into the `t.test` function (`t.test(data_male,data_female, mu = 0)`) but there is a more straightforward way to achieve this. We shall call `t.test(lnhrpay~male, mu=0, data = data_USoc)`. The `lnhrpay~male` is almost like a regression call, the variable we are interested in is lnhrpay but we want to know whether it differs according to male. The other inputs set the data frame (data = data_USoc) and the null hypothesis (mu=0).

```
t.test(lnhrpay~male, mu=0, data = data_USoc)  # testing that mu = 0
```

```
##
##  Welch Two Sample t-test
##
## data:  lnhrpay by male
## t = -35.022, df = 53923, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1932469 -0.1727632
## sample estimates:
## mean in group female   mean in group male
##             2.201202             2.384207
```

So here the p-value is extremely small indicating that the raw differential in log hourly wages between males and females is certainly statistically significant.

A regression - Gender differences

Regression analysis is our bread-and-butter technique and we can obtain the same result with a regression model.

```
mod1 <- lm(lnhrpay~male,data = data_USoc)
cov1 <- vcovHC(mod1, type = "HC1")
robust_se <- sqrt(diag(cov1))
stargazer(mod1,type="text",se=list(NULL, robust_se))
```

```
##
## ================================================
##                         Dependent variable:
##                     ----------------------------
##                               lnhrpay
## ------------------------------------------------
## malemale                      0.183***
##                               (0.005)
##
## Constant                      2.201***
##                               (0.003)
##
## ------------------------------------------------
## Observations                   58,960
## R2                              0.021
## Adjusted R2                     0.021
## Residual Std. Error      0.625 (df = 58958)
## F Statistic         1,251.147*** (df = 1; 58958)
## ================================================
## Note:                *p<0.1; **p<0.05; ***p<0.01
```

In the regression output you can see the variable male but it says malemale which is indication that basically a dummy variable has been included into your regression that takes the value 1 for every male respondent and 0 for everyone else.

The coefficient you can see for that is 0.183 which is exactly the difference between the male and female averages for `lnhrpay`.

If you were to merely calculate the regression model (`mod1 <- lm(lnhrpay~male,data = data_USoc)`) and then show the results with `stargazer`) you would get a very similar regression output, but the standard errors for the parameter estimates would have been calculated on the basis of a homoskedasticity assumption. Without any further details, this is an assumption which on many occasions is breached. However, the consequences are not too worrysome as long as we change how we calculate the standard errors. And that is what the extra lines achieve.

We can actually write a little function which does add the HC robust standard errors to the usual output. It is not so important that you fully understand what is hapening here. Copy and paste the next code chunk into a new script file and save it as `stargazer_HC.r` into your working directory.

```
stargazer_HC <- function(mod, type_in = "text") {
  cov1 <- vcovHC(mod, type = "HC1")
  robust_se <- sqrt(diag(cov1))
  stargazer(mod,type=type_in,se=list(NULL, robust_se))
}
```

Once you have saved `stargazer_HC.r` into your working directory you need to make it accessible to your code by including

```
source("stargazer_HC.r")
```

into your script. It is possibly best to do that right at the top where you load the packages (`library` commands).

Then we call a regresison and if we want robust standard errors we merely call `stargazer_HC` rather than `stargazer`). Added bonus is that you don't have to use the `type = "text"` option any longer as I used it as the default.

```
mod1 <- lm(lnhrpay~male,data = data_USoc)
stargazer_HC(mod1)
```

```
##
## ==========================================================
##                               Dependent variable:
##                          -------------------------------------
##                                      lnhrpay
## ----------------------------------------------------------
## malemale                            0.183***
##                                      (0.005)
##
## Constant                            2.201***
##                                      (0.003)
##
## ----------------------------------------------------------
## Observations                         58,960
## R2                                    0.021
## Adjusted R2                           0.021
## Residual Std. Error          0.625 (df = 58958)
## F Statistic             1,251.147*** (df = 1; 58958)
## ==========================================================
## Note:                          *p<0.1; **p<0.05; ***p<0.01
##                          Robust standard errors in parenthesis
```

Sometimes it is useful to extract the actual estimated coefficient and use it for some subsequent calculation.

On this occasion let us extract the estimated cooefficient for `male`. When we estimated the regression model we saved the output in `mod1`. In fact `mod1` contains a lot of information we may want to use, most notably the coefficient estimates, estimated residuals, fitted values etc. We access these as follows

```
mod1$coefficients
```

```
## (Intercept)    malemale
##   2.2012023   0.1830051
```

```
# mod1$residuals       # uncomment if you want to see these
# mod1$fitted.values
```

And if want a particular coefficient we can get that as follows

```
mod1$coefficients["malemale"]    # or
```

```
##  malemale
## 0.1830051
```

```
# mod1$coefficients[1]
```

where we use the coefficient name we also saw in the regression output.

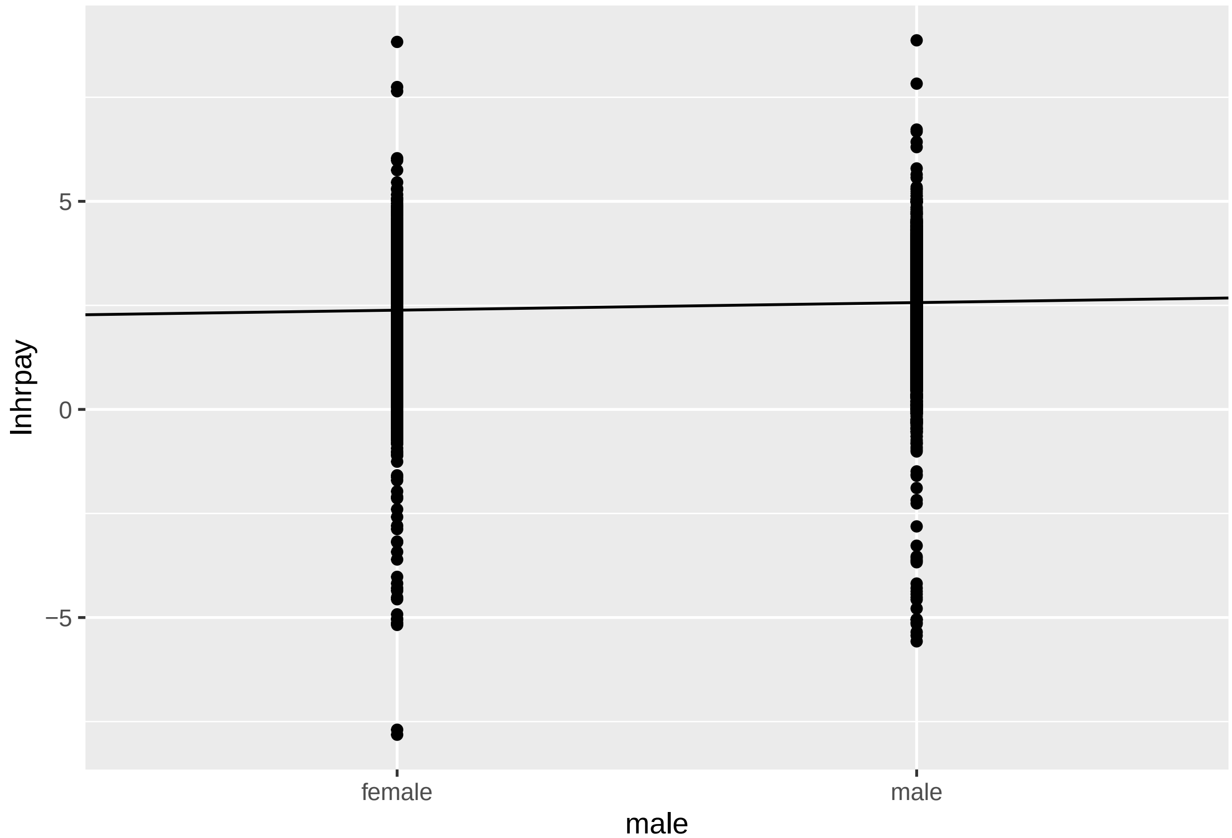
Let say we want to calculate the actual percentage points raw differential which is a function of this coefficient:

```
(exp(mod1$coefficients["malemale"])-1)*100
```

```
## malemale
## 20.08205
```

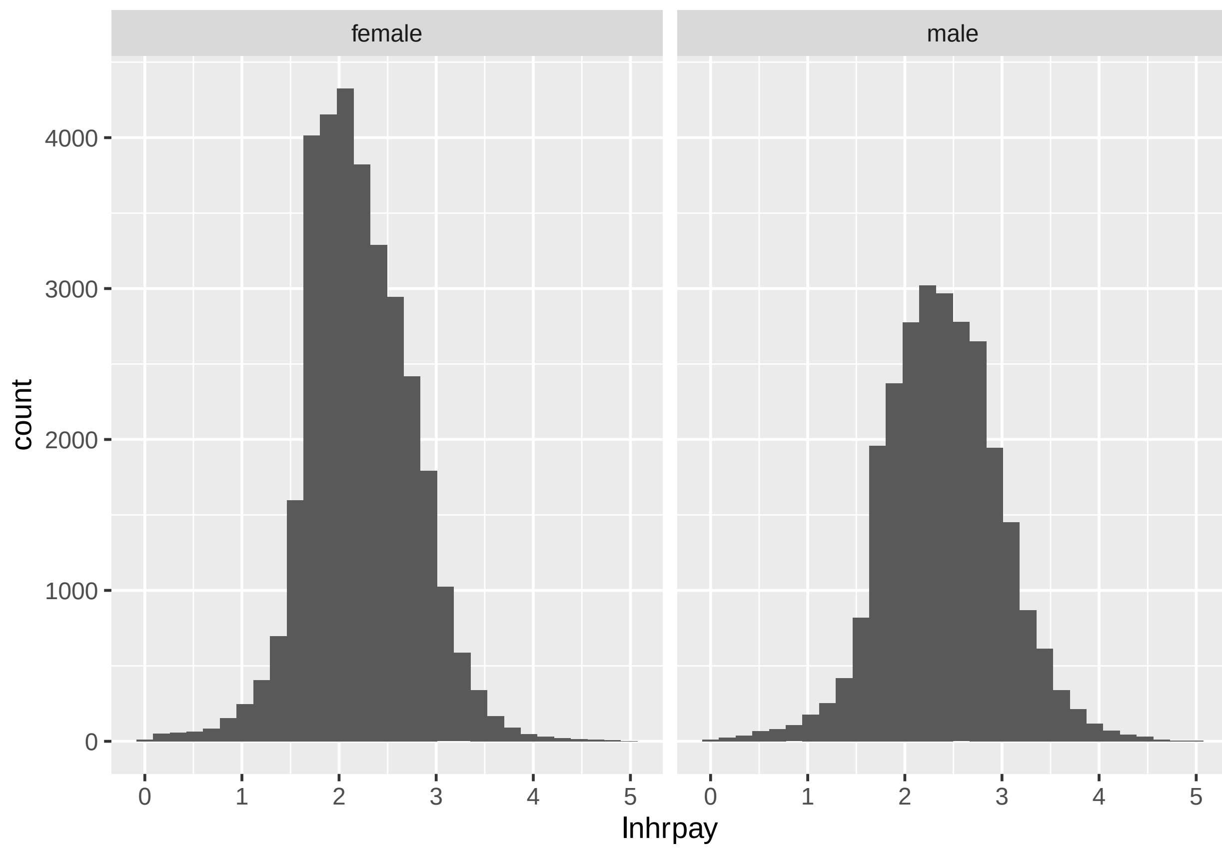
Let's produce a scatter plot of the data on which the above regression is based.

```
ggplot(data_USoc, aes(x = male, y = lnhrpay))  +
  geom_point() +
  geom_abline(intercept = mod1$coefficients[1], slope = mod1$coefficients[2])
```



As you can see this plot has limited informative value as, on the horizontal axis, we only have two possible outcomes, female and male. On each of these we have a wide range of outcomes for `lnhrpay`. We want to find a way to illustrate the distribution of outcomes on the `lnhrpay` scale depending on the value for `male`.

```
ggplot(subset(data_USoc, (lnhrpay>0) & (lnhrpay<5)), aes(x = lnhrpay))  +
  geom_histogram() +
  facet_grid(~male)
```

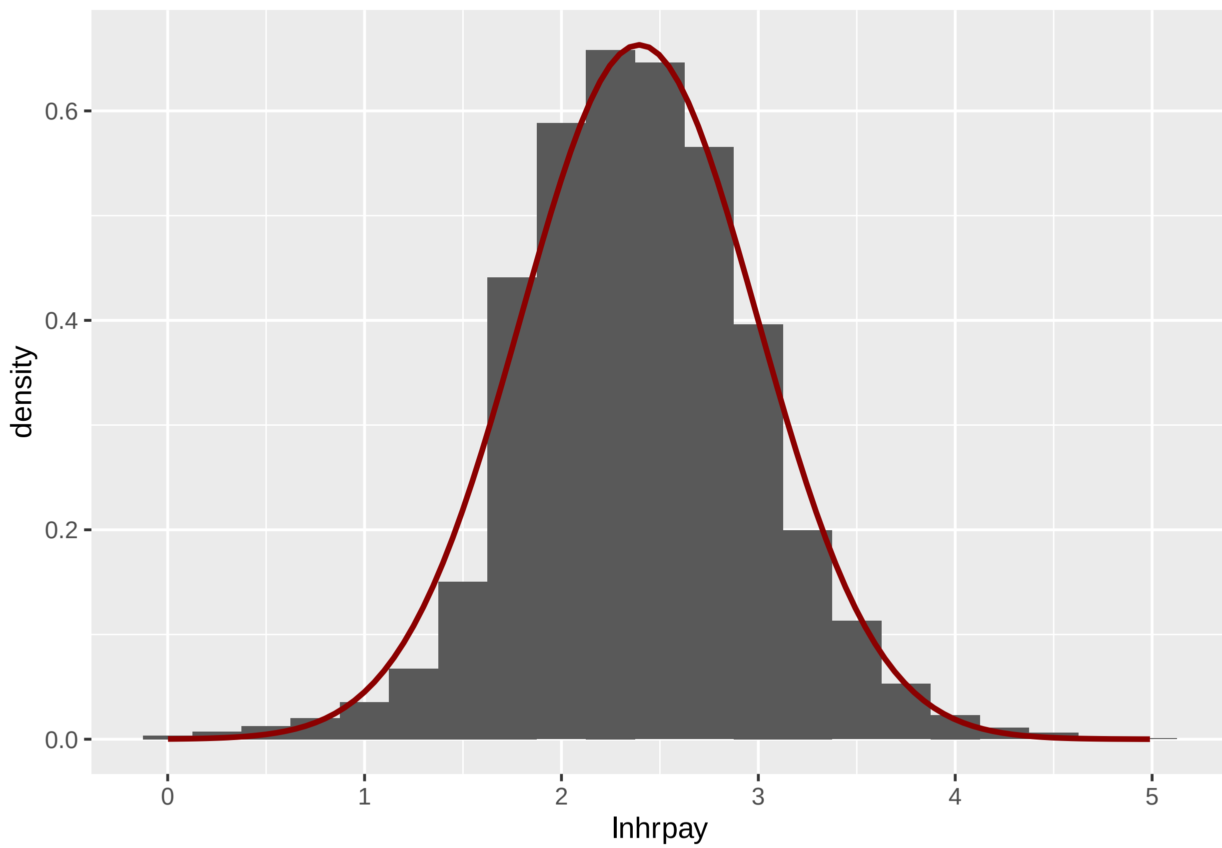


Recall that teh first input into the `ggplot` function is the dataframe we are using. HOwever, in the above command, instead of just using `data_USoc` we use `subset(data_USoc, (lnhrpay>0) & (lnhrpay<5))`. All that does is that we exclude some data, or better we select a subset of data, namely the data which have log hourly wage larger than 0 and smaller than 5. Try yourself how these histograms look if you do include all data (just ude `data_USoc`).

You could, if you wanted to, fit a normal distribution to these (remember the variables are the log hourly pay rates). This is not super straightforward and it turns out that in this case you would want to generate the graphs seperately. I had to google "R ggplot add normal distribution fit to geom_histogram" and came across https://stackoverflow.com/questions/1376967/using-stat-function-and-facet-wrap-together-in-ggplot2-in-r/1379074#1379074 which helped me to adopt a solution. The normal density is added with a `stat_function`:

```
# first create a relevant subset of the data
data_temp <- data_USoc %>%
            filter((lnhrpay>0) & (lnhrpay<5)) %>% # remove very small and large wages
            filter(male == "male")

# now create the plot
ggplot(data_temp,aes(x = lnhrpay))  +
      geom_histogram(binwidth = 0.25,aes(y = ..density..)) +
      stat_function(fun = dnorm, args=list(mean=mean(data_temp$lnhrpay),
                            sd=sd(data_temp$lnhrpay)), color = "darkred", size = 1)
```

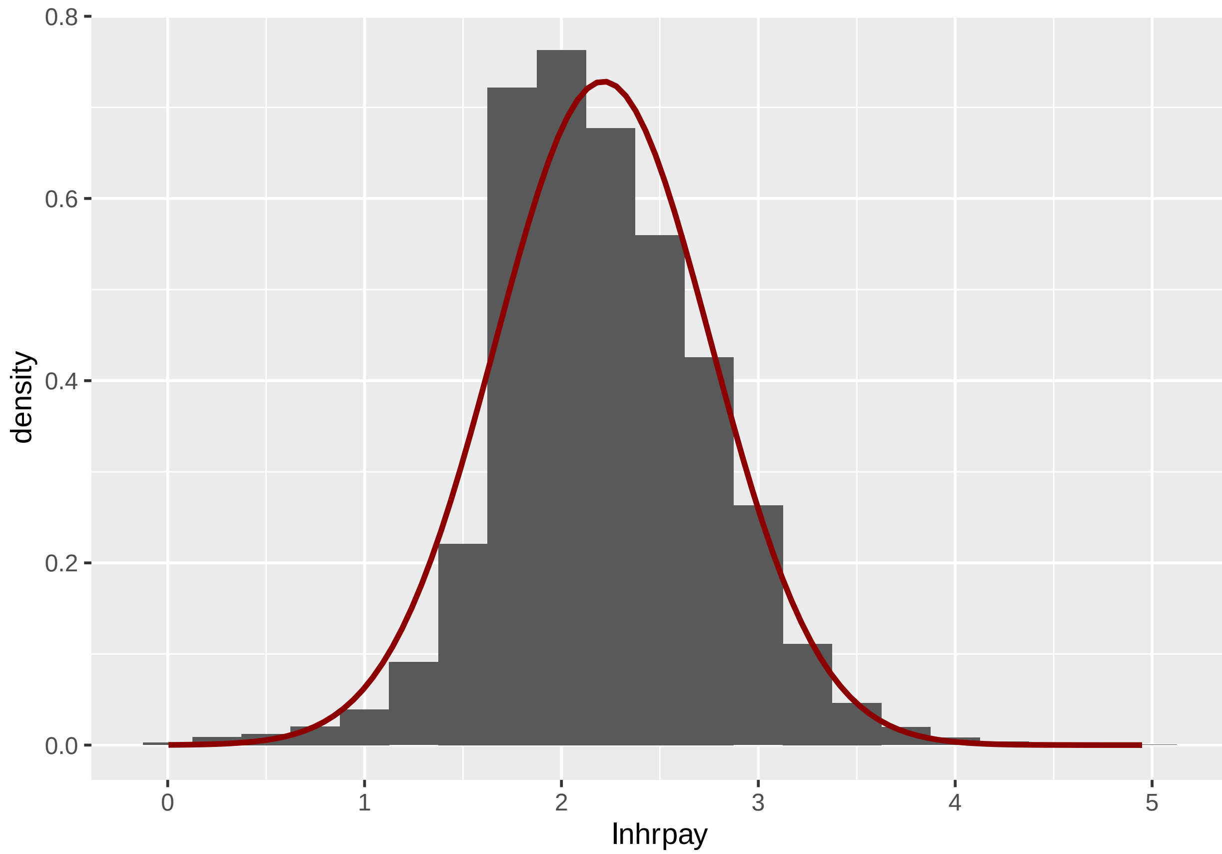



```
# Note: aes(y = ..density..) puts the histogram on a density scale
```

And now we can do the same for females.

```
# first create a relevant subset of the data
data_temp <- data_USoc %>%
            filter((lnhrpay>0) & (lnhrpay<5)) %>%
            filter(male == "female")

# now create the plot
ggplot(data_temp,aes(x = lnhrpay))  +
      geom_histogram(binwidth = 0.25,aes(y = ..density..)) +
      stat_function(fun = dnorm, args=list(mean=mean(data_temp$lnhrpay),
                            sd=sd(data_temp$lnhrpay)), color = "darkred", size = 1)
```



A regression - Education differences

After looking at gender differences we will now look at differences in earnings according to education differences.

Let's first look at the `degree` variable in our dataset.

```
data_USoc %>% count(degree)
```

```
## # A tibble: 4 x 2
##   degree              n
##   <fct>           <int>
## 1 no degree      102644
## 2 first degree    17509
## 3 higher degree   12888
## 4 <NA>              231
```

Let us create a new variable which merely differentiates between having any degree or no degree. So we essentially want to collapse the `first degree` and `higher degree` categories into a `any degree` category. Recall that this is a factor variable and there is a very convenient function (`fct_recode`) which allows you to change the levels. The `mutate(grad = ...)` function creates a new variable with the definition of that variable following the equal sign.

```
data_USoc <- data_USoc %>%
    mutate(grad = fct_recode(degree,
    "no degree"    = "no degree",    # new level = old level
```

```
    "any degree"  = "first degree",
    "any degree" = "higher degree"))
```

And now let's look at the counts of this new variable.

```
data_USoc %>% count(grad)
```

```
## # A tibble: 3 x 2
##   grad             n
##   <fct>        <int>
## 1 no degree  102644
## 2 any degree  30397
## 3 <NA>          231
```

And now we run a regression just as we did for gender differences.

```
mod1 <- lm(lnhrpay~grad,data = data_USoc)
stargazer_HC(mod1)
```

```
##
## =========================================================
##                              Dependent variable:
##                     -------------------------------------
##                                   lnhrpay
## ---------------------------------------------------------
## gradany degree                   0.472***
##                                   (0.005)
##
## Constant                         2.138***
##                                   (0.003)
##
## ---------------------------------------------------------
## Observations                      58,942
## R2                                 0.119
## Adjusted R2                        0.119
## Residual Std. Error          0.592 (df = 58940)
## F Statistic             7,959.665*** (df = 1; 58940)
## =========================================================
## Note:                        *p<0.1; **p<0.05; ***p<0.01
##                     Robust standard errors in parenthesis
```

And let's calculate the actual percentage points raw differential which is a function of the estimated coefficient to the grad variable:

```
(exp(mod1$coefficients["gradany degree"])-1)*100
```

```
## gradany degree
##       60.28478
```

Clearly there is a massive difference. Graduates, on average, earn more than 60% higher hourly wages.

A regression - gender and education differences

We found that, individually, gender and degree status make significant differences to hourly pay. Let's use the full power of regression analysis and see how hourly pay changes as a function of both these factors.

```
mod1 <- lm(lnhrpay~grad+male,data = data_USoc)
stargazer_HC(mod1)
```

```
##
## =========================================================
##                                Dependent variable:
##                     -------------------------------------
##                                      lnhrpay
## ---------------------------------------------------------
## gradany degree                      0.469***
##                                      (0.005)
##
## malemale                            0.176***
##                                      (0.005)
##
## Constant                            2.060***
##                                      (0.004)
##
## ---------------------------------------------------------
## Observations                         58,942
## R2                                    0.138
## Adjusted R2                           0.138
## Residual Std. Error            0.586 (df = 58939)
## F Statistic               4,727.510*** (df = 2; 58939)
## =========================================================
## Note:                         *p<0.1; **p<0.05; ***p<0.01
##                     Robust standard errors in parenthesis
```

Recall that both variables, `grad` and `male` have a base category (female and no degree respectively). It is best to think about this in form of a Table

Name	Gender	Degree	`male`	`grad`
John	male	no	1	0
Maria	female	no	0	0
Jess	female	yes	0	1
Pete	male	yes	1	1

The variable `male` will contribute to the conditional expectation for John and Pete and the `grad` variable will kick in for Jess and Pete. This sort of setup impies that the "effect"" of being male is the same regardless of whether you have a degree or not and also the "effect"" of having a degree is the same regardless of whether you are male or female.

If you want a model setup that does not make that assumption you need to include an interaction term:

```
mod2 <- lm(lnhrpay~grad*male,data = data_USoc)
stargazer_HC(mod2)
```

```
##
## =============================================================
##                                    Dependent variable:
##                         -------------------------------------
##                                          lnhrpay
## -------------------------------------------------------------
## gradany degree                          0.484***
```

```
##                                         (0.007)
##
## malemale                               0.187***
##                                         (0.006)
##
## gradany degree:malemale                -0.034***
##                                         (0.011)
##
## Constant                               2.056***
##                                         (0.004)
##
## ---------------------------------------------------------------
## Observations                            58,942
## R2                                       0.138
## Adjusted R2                              0.138
## Residual Std. Error                0.586 (df = 58938)
## F Statistic                   3,155.706*** (df = 3; 58938)
## ===============================================================
## Note:                               *p<0.1; **p<0.05; ***p<0.01
##                           Robust standard errors in parenthesis
```

The interaction term will take the value 1 only for those who are male and have a degree (in the above table only Pete). This setup allow for the "effect" of a degree to differ between males and females.