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

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)
```

Question 1

We will also calculate a modified race variable

```
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
```

We shall create a variable called `white` which re-groups this into two categories, white and non-white.

```
data_USoc <- data_USoc %>%
    mutate(white = fct_recode(race,
    "non-white"  = "mixed",    # new level = old level
    "non-white"  = "asian",
    "non-white"  = "black",
    "non-white"  = "other",
    "white"    = "white"))
```

When you define a factor variable, then there will be a base category. You find out what the base theory is by checking the `levels`of the factor variable.

```
levels(data_USoc$white)
```

```
## [1] "white"     "non-white"
```

The first category is the base category, here white. As this is a binary factor variable you could think about this as a dummy variable which takes value 0 for the base category (here white) and 1 for the other category (non-white). As we labeled our new variable `white` it is convention to associate the value 1 with observations which are actually white. So we need to swap the levels:

```
data_USoc$white <- factor(data_USoc$white,levels(data_USoc$white)[c(2,1)])
levels(data_USoc$white)
```

```
## [1] "non-white" "white"
```

It is the very last part pf this command (`c(2,1)`) which ensures that the new first level is the one that previously was second. There is no need to remember such a command, I had to google" "r change level order of factor" in order to remember how to do that.

The pay information (`paygu`) is provided as a measure of the (usual) gross pay per month. As workers work for varying number of days we will calculate a hourly payrate to facilitate a compparison. We shall also adjust for increasing price levels (as measured by `cpi`). We call this variable `hrpay` and also calculate the natural log of this variable (`lnhrpay`). The `mutate` function creates a new variable.

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

Let's estimate a regression, allowing average hourly pay to differ between whites and non-whites

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

```
##
## ============================================================
##                               Dependent variable:
##                      ---------------------------------------
##                                     lnhrpay
## ------------------------------------------------------------
## whitewhite                         0.053***
##                                     (0.007)
##
## Constant                           2.240***
##                                    (0.006)
##
## ----------------------------------------------------------
## Observations                       57,915
## R2                                  0.001
## Adjusted R2                         0.001
## Residual Std. Error          0.631 (df = 57913)
## F Statistic               59.331*** (df = 1; 57913)
## ==========================================================
## Note:                          *p<0.1; **p<0.05; ***p<0.01
##                        Robust standard errors in parenthesis
```

Let say we want to calculate the actual percentage points raw differential between white and non-white respondents, which is a function of the slope coefficient:

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

```
## whitewhite
##    5.42403
```

You can see that, as this value is fairly small, the value of the raw differential is very close to the value of the actual parameter. Only when this parameter becomes larger is it necessary to apply the above formula. So, white respondents, on average, earn about 5.5% higher hourly pay.

Question 2

Now we calculate inflation adjusted figures for gross monthly pay (paygu) and scale it to annual pay (annualpay).

```
data_USoc <- data_USoc %>%
              mutate(annualpay = paygu*12/(cpi/100))
summary(data_USoc$annualpay)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
##      0.81   8695.65  15551.84  18761.59  24665.56 162454.87     74056
```

Now we look at estimating the following model

$$anualpay = \beta_0 + \beta_1\ age + v$$

```
mod1 <- lm(annualpay~age,data = data_USoc,subset=(degree == "first degree"))
mod2 <- lm(annualpay~age,data = data_USoc,subset=(degree == "no degree"))
stargazer(mod1,mod2, type = "text",
          column.labels = c("first degree","without degree"),omit.stat = "all")
```

```
##
## =====================================
##              Dependent variable:
##          ----------------------------
##                    annualpay
##          first degree  without degree
##              (1)            (2)
## -------------------------------------
## age       270.314***     142.448***
##            (14.932)       (4.284)
##
```

```
## Constant 14,365.930***  9,437.297***
##            (607.774)     (185.646)
##
## =====================================
## =====================================
## Note:     *p<0.1; **p<0.05; ***p<0.01
```

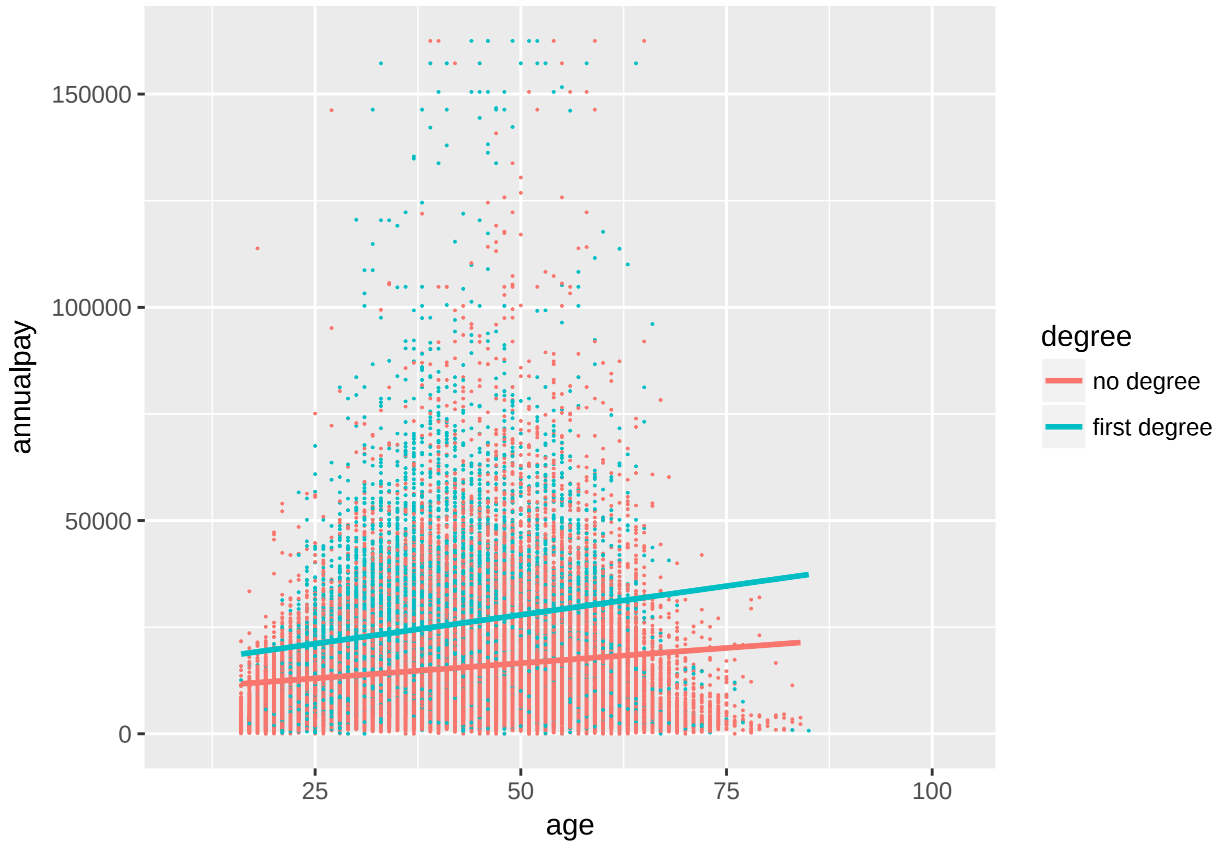
Now we use these models to predict annual wages at the age of 20 and 65 depending on whether someone has a degree or not.

```
# with degree, use mod1
y20_1 <- mod1$coefficients[1]+mod1$coefficients[2]*20
y65_1 <- mod1$coefficients[1]+mod1$coefficients[2]*65

# without degree, use mod2
y20_0 <- mod2$coefficients[1]+mod2$coefficients[2]*20
y65_0 <- mod2$coefficients[1]+mod2$coefficients[2]*65
```

Let's create a plot with the two fitted regression lines. But before we do that we create a temporary dataset which only containsn obs with the two `degree` outcomes we consider here, i.e. "no degree" and "first degree".

```
temp <- data_USoc %>% filter(degree %in% c("no degree", "first degree"))
ggplot(temp, aes(x=age,y=annualpay,color = degree))+
  geom_point(size=0.01) +
  geom_smooth(method = "lm", se = FALSE)
```



To calculate the lifetime earnings premium for a graduate (earnings between 20 and 65) we calculate the lifetime earnings of a graduate and substract the lifetime earnings of a non-graduate.

```r
L1 = (65-20)*(y20_1+y65_1)/2
L0 = (65-20)*(y20_0+y65_0)/2
D = L1 - L0
unname(L1)   # If you don't use unname then
```

```
## [1] 1163442
```

```r
unname(L0)   # all these values will inherit the
```

```
## [1] 697109.9
```

```r
unname(D)    # name "intercept" from coefficients
```

```
## [1] 466332
```

So this almost half a million pounds over a lifetime. Is that the value of a degree?

Graphically this is the difference between the two fitted regression lines between the ages 20 and 65.