R-work for Online Assessment

Instructions

You should go through the code below and complete it. Keep the completed code and all the resulting output. With that information you will be able to answer the questions you will see in the online quiz. Every student will see a slightly different collection of questions (as we will draw 10 questions from a pool of questions with about 20 questions).

The type of questions you will see will be of four types.

1) Questions that merely ask you to report output from your analysis.
2) Some questions will ask you about R code. For instance you will see a lot of gaps (`XXXX`) in the code and questions may ask you how you had to complete the code to make the code work. Other questions may ask you what output a particular bit of code would produce. If you want to practice these sorts of questions you could practice on Datacamp.
3) The third type of questions will ask you understanding questions like "What is the meaning of an estimated coefficient?" "Is a particular coefficient statistically significant?"
4) We may be asked questions on general programming issues, like what is the meaning of an error message or how would you search for a particular piece of information.

Preparing your workfile

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

```
library(tidyverse)    # for almost all data handling tasks
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
source("stargazer_HC.r")  # includes the robust regression display
```

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

```r
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.

```r
data_USoc$region <- XXXX
data_USoc$male <- XXXX
data_USoc$degree <- XXXX
data_USoc$race <- XXXX
```

```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)
```

As we defined the `male` variable as a factor it has levels `male` and `female` (check `levels(data_USoc$male)` to confirm). It would be better to relabel the variable to `gender`.

```r
names(data_USoc)[names(data_USoc) == "male"] <- "gender"
```

The pay information (`paygu`) is provided as a measure of the (usual) gross pay per month. As workers work for varying numbers of hours per week (`jbhrs`) we divide the monthly pay by the approximate monthly hours (`4*jbhrs`). We shall also adjust for increasing price levels (as measured by `cpi`). These two adjustments leave us with an inflation adjusted hourly wage. We call this variable `hrpay` and also calculate the natural log of this variable (`lnhrpay`).

```r
data_USoc <- data_USoc %>%
              XXXX(hrpay = XXXX) %>%
              XXXX(lnhrpay = XXXX)
```

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

As we wanted to save these additional variables we assign the result of the operation to `data_USoc`.

We also want to use a measure of annual pay (`paygu*12/(cpi/100))`) and add this variable (`annualpay`) to the dataframe (`data_USoc`).

```r
data_USoc <- XXXX %>%
              XXXX
```

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

Here we will evaluate the relation between `lnhrpay` and `age`. Let's first summarise all numerical variables in our dataset, using the `stargazer` function.

```r
XXXX
```

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

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Thu, Feb 28, 2019 - 15:23:31

You should find, for instance, that the mean value of the unemployment rate (`urate`) is 7.955 and the standard

Table 1:

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
hrpay	58,960	12.268	45.140	0.0004	6.612	14.518	7,104.150
lnhrpay	58,960	2.283	0.631	−7.816	1.889	2.675	8.868
annualpay	59,216	18,761.600	15,185.830	0.813	8,695.652	24,665.560	162,454.900

deviation for the `age` variable is 18.295.

For later purposes we will also need variables $age^2/100$ and $log(age)$. We now need to create these variables (`agesq` and `lnage`) and add them to the `data_USoc` dataframe.

```
XXXX
mean(data_USoc$lnage)
```

```
data_USoc <- data_USoc %>%
                mutate(agesq = age*age/100) %>%
                mutate(lnage = log(age))
mean(data_USoc$lnage)
```

[1] 3.744307

You should find the mean of `lnage` to be 3.744307.

Data cleaning

We will now remove unusable observations from our `data_USoc` dataframe, in particular those observations which have missing (`NA`) data for `lnhrpay`. In addition to that we will also remove observations from males which are 66 years or older and females whose age is 61 years or older.

```
data_USoc <- data_USoc %>%
                XXXX
```

```
data_USoc <- data_USoc %>%
                filter(!is.na(lnhrpay)) %>%
                filter((gender == "male" & age < 66) | (gender == "female" & age < 61) )
```

You should end up with 56778 observations.

Estimate regression models - Version 1

We shall estimate the following regression models (`mod1`)

$$lnhrpay = \beta_0 + \beta_1 \; age + \beta_2 \; agesq + u$$

and (`mod2`)

$$lnhrpay = \alpha_0 + \alpha_1 \; lnage + u$$

```
mod1 <- lm(XXXX ~ XXXX+XXXX, data = data_USoc)
mod2 <- lm(XXXX)
stargazer_HC(mod1,mod2)
```

```
mod1 <- lm(lnhrpay ~ age+agesq, data = data_USoc)
mod2 <- lm(lnhrpay ~ lnage, data = data_USoc)
stargazer_HC(mod1,mod2,type_out="latex")
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Thu, Feb 28, 2019 - 15:23:33

Table 2:

	Dependent variable:	
	lnhrpay	
	(1)	(2)
age	0.087^{***}	
	(0.001)	
agesq	-0.096^{***}	
	(0.002)	
lnage		0.485^{***}
		(0.008)
Constant	0.485^{***}	0.531^{***}
	(0.026)	(0.028)
Observations	56,778	56,778
R^2	0.098	0.066
Adjusted R^2	0.098	0.066
Residual Std. Error	0.594 (df = 56775)	0.605 (df = 56776)
F Statistic	$3,101.185^{***}$ (df = 2; 56775)	$3,984.399^{***}$ (df = 1; 56776)
Note:		*p<0.1; **p<0.05; ***p<0.01
		Robust standard errors in parenthesis

You have done this correctly if you find that that your estimated constant for `mod1` is 0.560.

These are two ways of allowing `age` to be correlated with `lnhrpay`. Both are what we call parametric, either a quadratic relation or a logarithmic relation.

As we have lots of data we can do this in a more flexible way as well. We could basically produce a dummy variable for every age (ages are reported in full years only) an then include a dummy variable for each age. TO do this we will first have to create an age variable which treats age as a categorical, or in R terms factor variable. We shall call this `age_f`.

```
data_USoc <- data_USoc %>% mutate(age_f = as.factor(age))
```

With `age_f` being a factor variable, it is now straightforward to include thsi factor variable into a regresison. We can either include a constant (`lnhrpay ~ age_f`) which will then use age = 16 as a base category, or we can estimate the model without a constant (`lnhrpay ~ age_f - 1`) in which case all age categories enter separately.

```
mod3 <- lm(lnhrpay ~ age_f, data = data_USoc)
mod4 <- lm(lnhrpay ~ age_f -1, data = data_USoc)
stargazer_HC(mod3,mod4)
```

```
mod3 <- lm(lnhrpay ~ age_f, data = data_USoc)
mod4 <- lm(lnhrpay ~ age_f -1, data = data_USoc)
stargazer_HC(mod3,mod4,type_out="latex")
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Thu, Feb 28, 2019 - 15:23:36

Let's compare the fitted values for `mod1`, `mod2` and `mod4`. First we add the predicted values to the dataframe. There are several ways to achieve this and I recommend you ask Dr. Google. Think carefully about the search terms.

```
data_USoc$pred_mod1 <- XXXX
data_USoc$pred_mod2 <- XXXX
data_USoc$pred_mod4 <- XXXX
```

```
data_USoc$pred_mod1 <- mod1$fitted.values
data_USoc$pred_mod2 <- mod2$fitted.values
data_USoc$pred_mod4 <- mod4$fitted.values
```

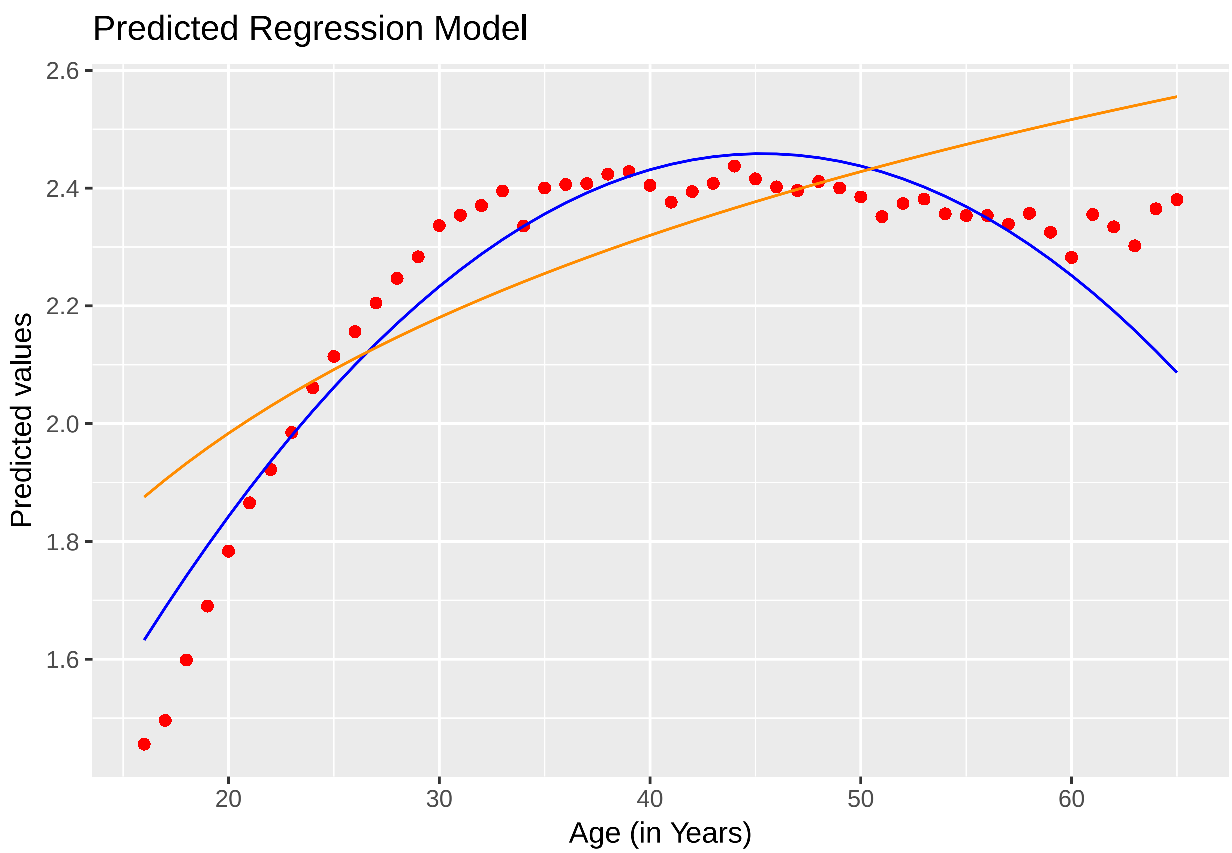
Now we plot the predicted values for the three specifications. You should also change the Axis labels to "Predicted values" for the vertical axis, "Age (in Years)" for the horizontal axis and add a title ("Predicted Regression Model") to your picture. If you google you should find the appropriate commands. Think carefully about the search terms.

```
ggplot(data_USoc, aes(x=age,y=pred_mod4)) +
  geom_point(color = "red") +
  geom_line(aes(y=pred_mod1),color = "blue") +
  geom_line(aes(y=pred_mod2),color = "darkorange") +
  XXXX +    # add code to give your plot a title
  XXXX      # add code to change the axis labels
```

```
ggplot(data_USoc, aes(x=age,y=pred_mod4)) +
  geom_point(color = "red") +
  geom_line(aes(y=pred_mod1),color = "blue") +
  geom_line(aes(y=pred_mod2),color = "darkorange") +
  ggtitle("Predicted Regression Model") +
  ylab("Predicted values") +
  xlab("Age (in Years)")
```

Table 3:

	Dependent variable:	
	lnhrpay	
	(1)	(2)
age_f16		1.456***
		(0.033)
age_f17	0.040	1.496***
	(0.041)	(0.024)
age_f18	0.143***	1.599***
	(0.038)	(0.019)
age_f19	0.234***	1.690***
	(0.039)	(0.021)
age_f20	0.328***	1.783***
	(0.038)	(0.019)
age_f21	0.410***	1.866***
	(0.038)	(0.018)
age_f22	0.466***	1.922***
	(0.037)	(0.017)
age_f23	0.529***	1.985***
	(0.036)	(0.014)
age_f24	0.605***	2.061***
	(0.036)	(0.012)
age_f25	0.658***	2.114***
	(0.037)	(0.015)
age_f26	0.701***	2.156***
	(0.036)	(0.015)
age_f27	0.749***	2.205***
	(0.036)	(0.014)
age_f28	0.791***	2.247***
	(0.036)	(0.015)
age_f29	0.828***	2.283***
	(0.037)	(0.016)
age_f30	0.881***	2.336***
	(0.036)	(0.014)
age_f31	0.898***	2.354***
	(0.037)	(0.015)
age_f32	0.915***	2.370***
	(0.036)	(0.014)
age_f33	0.939***	2.395***
	(0.037)	(0.015)



The fit of `mod4` is the most flexible specification as it uses a coefficient for each year. Specifications `mod1` and `mod2` models model the relationship between `age` and `lnhrpay` with one and two parameters respectively.

Estimate regression models 2

Now we will estimate a quadratic model for `annualpay` (`annualpay ~ age + agesq`) on a subsets of data in order to compare these. When you know that you will be working with different subsets of data, the best way of doing that in R is to create a new factor variale (here `subset_ind`) which allows you to separate the data accordingly.

We will create two subgroups: 1) Males with a first degree and 2) Males with no degree. You may want to check the values of the `degree` variable in order to define these correctly.

```
data_USoc$subset_ind <- "none"   # default group
data_USoc$subset_ind[data_USoc$gender == "male" & data_USoc$degree == "first degree"] <- "Male with firs
data_USoc$subset_ind[XXXX] <- "Male without first degree"  # select all males with no degree
data_USoc$subset_ind <- as.factor(data_USoc$subset_ind)
data_USoc %>% count(subset_ind)
```

```
data_USoc$subset_ind <- "none"   # default group
data_USoc$subset_ind[data_USoc$gender == "male" & data_USoc$degree == "first degree"] <- "Male with firs
data_USoc$subset_ind[data_USoc$gender == "male" & data_USoc$degree == "no degree"] <- "Male without firs
data_USoc$subset_ind <- as.factor(data_USoc$subset_ind)
data_USoc %>% count(subset_ind)
```

A tibble: 3 x 2

subset_ind n 1 Male with first degree 4464 2 Male without first degree 17626 3 none 34688

We will want to save the model predictions and for this purpose we pre-define a variable in which we will save the predictions.

```
data_USoc$pred_mod5 <- 0     # set the prediction to 0 by default
```

Now we estimate the model for the male with first degree subgroup. Note that the `lm` function accepts a `subset` argument which allows you to select a subset of observations, such as the group of all males with first degree.

```
mod5_md <- lm(XXXX ~ XXXX + XXXX, data = XXXX, subset = (subset_ind == XXXX))
stargazer_HC(XXXX)
data_USoc$pred_mod5[data_USoc$subset_ind==XXXX] <- mod5_md$fitted.values
```

```
mod5_md <- lm(annualpay ~ age + agesq, data = data_USoc, subset = (subset_ind == "Male with first degre
stargazer_HC(mod5_md,type_out="latex")
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Thu, Feb 28, 2019 - 15:23:40

Table 4:

	Dependent variable:
	annualpay
age	3,924.330***
	(187.780)
agesq	−4,251.222***
	(225.948)
Constant	−53,414.740***
	(3,703.243)
Observations	4,464
R^2	0.121
Adjusted R^2	0.121
Residual Std. Error	19,324.540 (df = 4461)
F Statistic	308.456*** (df = 2; 4461)
Note:	*p<0.1; **p<0.05; ***p<0.01
	Robust standard errors in parenthesis

```
data_USoc$pred_mod5[data_USoc$subset_ind=="Male with first degree"] <- mod5_md$fitted.values
```

Now we repeat the same just for the group of males with no degree

```
mod5_mnd <- XXXX
stargazer_HC(XXXX)
data_USoc$pred_mod5[XXXX] <- mod5_mnd$fitted.values
```

```
mod5_mnd <- lm(annualpay ~ age + agesq, data = data_USoc, subset = (subset_ind == "Male without first d
stargazer_HC(mod5_mnd,type_out="latex")
```

% Table created by stargazer v.5.2.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu
% Date and time: Thu, Feb 28, 2019 - 15:23:40

Table 5:

	Dependent variable:
	annualpay
age	1,949.206***
	(45.437)
agesq	−2,099.132***
	(56.181)
Constant	−21,087.680***
	(858.212)
Observations	17,626
R^2	0.137
Adjusted R^2	0.137
Residual Std. Error	12,592.500 (df = 17623)
F Statistic	1,397.954*** (df = 2; 17623)
Note:	*p<0.1; **p<0.05; ***p<0.01
	Robust standard errors in parenthesis

```
data_USoc$pred_mod5[data_USoc$subset_ind=="Male without first degree"] <- mod5_mnd$fitted.values
```

Now we plot the predicted values for the two specifications.

```
ggplot(data_USoc, aes(x=age,y=pred_mod5,color = subset_ind)) +
  geom_line() +
  ggtitle("Predicted Regression Model - Model 5") +
  ylab("Predicted values") +
  xlab("Age")
```

You will see that we have the “none” category plotted as well (of course we didn’t estimate this). You could remove these data before plotting

```
data_temp <- data_USoc %>% filter(subset_ind != "none")  # remove observations with subset_ind == "none
ggplot(data_temp, aes(x=age,y=pred_mod5,color = subset_ind)) +
  geom_line() +
  ggtitle("Predicted Regression Model - Model 5") +
  ylab("Predicted values") +
  xlab("Age")
```

END OF INSTRUCTIONS