Mini-Lesson 1

Sumner Perera

2025-01-30

2: Obtaining the Data

Completed - data downloaded

3. Preparing the Data

1. Check
2.

```
## load packages
import pandas as pd
import numpy as np
```

```
## open up the data
path = r"C:\Users\12019\OneDrive - The University of
↪ Chicago\Documents\GitHub\Machine-Learning\mini_lesson_1\usa_00001.csv"

edu = pd.read_csv(path)
```

a.

```
## open up the crosswalk
path2 = r"C:\Users\12019\OneDrive - The University of
↪ Chicago\Documents\GitHub\Machine-Learning\mini_lesson_1\PPHA_30545_MP01-Crosswalk.csv"

crosswalk = pd.read_csv(path2)
crosswalk.rename(columns={"educd": "EDUCD", "educdc": "EDUCDC"}, inplace =
↪ True)
```

```
## create new column using crosswalk
edu_merged = edu.merge(crosswalk, on = "EDUCD", how = "left")
```

b.

```
## create dummy variables
### hsdip for EDUCDC values 12-15
hs=[12,13,14,15]
edu_merged['hsdip'] = np.where(edu_merged['EDUCDC'].isin(hs), 1, 0)

### coldip for EDUCDC values 16 or greater
edu_merged['coldip'] = np.where(edu_merged['EDUCDC']>= 16, 1, 0)

### white for RACE = 1, black for RACE = 2, hispanic for HISPAN = 1, 2, 3, 4
edu_merged['white'] = np.where(edu_merged['RACE'] == 1, 1, 0)
edu_merged['black'] = np.where(edu_merged['RACE'] == 2, 1, 0)
hisp=[1,2,3,4]
edu_merged['hispanic'] = np.where(edu_merged['HISPAN'].isin(hisp), 1, 0)

## married for MARST = 1 or 2
mar=[1,2]
edu_merged['married'] = np.where(edu_merged['MARST'].isin(mar), 1, 0)

## female for SEX = 2
edu_merged['female'] = np.where(edu_merged['SEX'] == 2, 1, 0)

## vet for VETSTAT=2
edu_merged['vet'] = np.where(edu_merged['VETSTAT'] == 2, 1, 0)
```

c.

```
## create the interaction term between both of the education dummies and the
↪ continuous
edu_merged['interact'] =
↪ edu_merged['hsdip']*edu_merged['coldip']*edu_merged['EDUCDC']
```

d.

```
## age squared var
edu_merged['age_sq']= np.power(edu_merged['AGE'],2)

## drop any observations where incwage<=0.
edu_merged_clean = edu_merged.loc[edu_merged['INCWAGE']>0]

## create new var that's the ln of INCWAGE
edu_merged_clean['lnincwage'] = np.log(edu_merged['INCWAGE'])
```

```
c:\Users\12019\OneDrive - The University of Chicago\Documents\GitHub\.venv\Lib\site-packages\
  result = getattr(ufunc, method)(*inputs, **kwargs)
C:\Users\12019\AppData\Local\Temp\ipykernel_16500\2885664428.py:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide
  edu_merged_clean['lnincwage'] = np.log(edu_merged['INCWAGE'])
```

4. Data Analysis Questions

1.

```
## descriptive stats for multiple variables
vars = ['YEAR', 'INCWAGE', 'lnincwage', 'EDUCDC', 'female', 'AGE', 'age_sq',
↪  'white', 'black', 'hispanic', 'married', 'NCHILD', 'vet', 'hsdip',
↪  'coldip', 'interact']

for title in vars:
  print(f'Descriptive stats for {title}')
  print(edu_merged_clean[title].describe())
```

```
Descriptive stats for YEAR
count    8683.0
mean     2023.0
std         0.0
min      2023.0
25%      2023.0
50%      2023.0
75%      2023.0
max      2023.0
```

```
Name: YEAR, dtype: float64
Descriptive stats for INCWAGE
count      8683.000000
mean      69000.380053
std       77823.703146
min          40.000000
25%       26800.000000
50%       50000.000000
75%       85000.000000
max      870000.000000
Name: INCWAGE, dtype: float64
Descriptive stats for lnincwage
count    8683.000000
mean       10.678624
std         1.072527
min         3.688879
25%        10.196150
50%        10.819778
75%        11.350407
max        13.676248
Name: lnincwage, dtype: float64
Descriptive stats for EDUCDC
count    8683.000000
mean       14.276978
std         3.047631
min         0.000000
25%        12.000000
50%        14.000000
75%        16.000000
max        22.000000
Name: EDUCDC, dtype: float64
Descriptive stats for female
count    8683.000000
mean        0.482207
std         0.499712
min         0.000000
25%         0.000000
50%         0.000000
75%         1.000000
max         1.000000
Name: female, dtype: float64
Descriptive stats for AGE
count    8683.000000
```

```
mean       41.626396
std        13.361937
min        18.000000
25%        30.000000
50%        42.000000
75%        53.000000
max        65.000000
Name: AGE, dtype: float64
Descriptive stats for age_sq
count    8683.000000
mean     1911.277669
std      1122.303850
min       324.000000
25%       900.000000
50%      1764.000000
75%      2809.000000
max      4225.000000
Name: age_sq, dtype: float64
Descriptive stats for white
count    8683.000000
mean        0.671427
std         0.469721
min         0.000000
25%         0.000000
50%         1.000000
75%         1.000000
max         1.000000
Name: white, dtype: float64
Descriptive stats for black
count    8683.000000
mean        0.075665
std         0.264477
min         0.000000
25%         0.000000
50%         0.000000
75%         0.000000
max         1.000000
Name: black, dtype: float64
Descriptive stats for hispanic
count    8683.000000
mean        0.163192
std         0.369562
min         0.000000
```

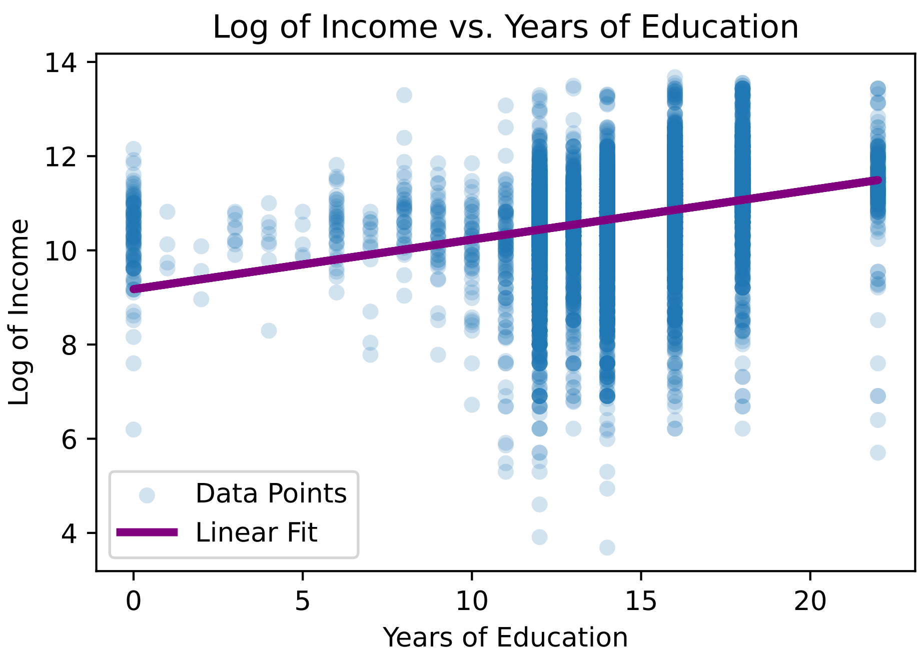
```
25%          0.000000
50%          0.000000
75%          0.000000
max          1.000000
Name: hispanic, dtype: float64
Descriptive stats for married
count    8683.000000
mean        0.547507
std         0.497767
min         0.000000
25%         0.000000
50%         1.000000
75%         1.000000
max         1.000000
Name: married, dtype: float64
Descriptive stats for NCHILD
count    8683.000000
mean        0.768053
std         1.093575
min         0.000000
25%         0.000000
50%         0.000000
75%         1.000000
max         8.000000
Name: NCHILD, dtype: float64
Descriptive stats for vet
count    8683.000000
mean        0.039387
std         0.194526
min         0.000000
25%         0.000000
50%         0.000000
75%         0.000000
max         1.000000
Name: vet, dtype: float64
Descriptive stats for hsdip
count    8683.000000
mean        0.539099
std         0.498498
min         0.000000
25%         0.000000
50%         1.000000
75%         1.000000
```

```
max          1.000000
Name: hsdip, dtype: float64
Descriptive stats for coldip
count    8683.000000
mean        0.412070
std         0.492236
min         0.000000
25%         0.000000
50%         0.000000
75%         1.000000
max         1.000000
Name: coldip, dtype: float64
Descriptive stats for interact
count    8683.0
mean        0.0
std         0.0
min         0.0
25%         0.0
50%         0.0
75%         0.0
max         0.0
Name: interact, dtype: float64
```

2.

```
## scatter plot of lnincwage and educdc
import matplotlib.pyplot as plt
import seaborn as sns

## create linear fit
from sklearn.linear_model import LinearRegression as lm
y = edu_merged_clean[['lnincwage']].values
X = edu_merged_clean[['EDUCDC' ]].values
y_pred = lm().fit(X, y).predict(X)

## create plot with linear fit
fig,ax = plt.subplots()
ax.scatter(edu_merged_clean['EDUCDC'], edu_merged_clean['lnincwage'],
 ↪  alpha=0.2, s=25, label='Data Points')
ax.plot(X, y_pred, color="purple", linewidth=3, label='Linear Fit')

## set title, legend, labels
ax.set_xlabel("Years of Education")
```

```
ax.set_ylabel("Log of Income")
ax.set_title('Log of Income vs. Years of Education')
ax.legend()

## show plot
plt.show()
```



3.

```
## load libraries
import statsmodels.formula.api as smf

## estimate given model
regression = smf.ols('lnincwage ~ EDUCDC + female + AGE + age_sq + white +
↪  black + hispanic + married + NCHILD + vet', data =
↪  edu_merged_clean).fit()
print(regression.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              lnincwage   R-squared:                       0.294
```

```
Model:                            OLS   Adj. R-squared:                  0.293
Method:                 Least Squares   F-statistic:                     360.7
Date:                Fri, 31 Jan 2025   Prob (F-statistic):               0.00
Time:                        01:41:47   Log-Likelihood:                -11418.
No. Observations:                8683   AIC:                         2.286e+04
Df Residuals:                    8672   BIC:                         2.294e+04
Df Model:                          10
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      6.3154      0.113     56.063      0.000       6.095       6.536
EDUCDC         0.0889      0.003     26.606      0.000       0.082       0.095
female        -0.4297      0.020    -21.793      0.000      -0.468      -0.391
AGE            0.1465      0.006     26.355      0.000       0.136       0.157
age_sq        -0.0015   6.55e-05    -23.041      0.000      -0.002      -0.001
white          0.0096      0.027      0.362      0.718      -0.043       0.062
black         -0.1984      0.043     -4.621      0.000      -0.283      -0.114
hispanic      -0.0883      0.032     -2.762      0.006      -0.151      -0.026
married        0.2116      0.023      9.312      0.000       0.167       0.256
NCHILD     -5.364e-05      0.010     -0.005      0.996      -0.020       0.020
vet           -0.1387      0.051     -2.744      0.006      -0.238      -0.040
==============================================================================
Omnibus:                     2210.672   Durbin-Watson:                   1.858
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             8303.702
Skew:                          -1.233   Prob(JB):                         0.00
Kurtosis:                       7.108   Cond. No.                     2.60e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.6e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

a. According to the R^2 value, the model explains 29% of the variation in log wages.

b. An additional year of education gives an increase of 8.9% in income, holding all else constant. This is statistically significant at the 5% significance level with a p value that is less than 0.05 but practically this doesn't necessarily hold true for all levels of education.

For example, no children work thus a change from 2 to 3 years of education (which would signify a toddler) then their wage should not increase but rather stay static at 0. This would be the case all the way until an individual reaches high school which is 12 years of schooling at which point they are able to legally get a job and then additional years of schooling would impact their earnings.

c. To figure out the age that gives the largest increase in % of age, take the derivative of the regression with respect to age and solve for the variable. This gives an age of 49 that yields the highest % increase in wage.

(see calculations at end of PDF)

d. The model predicts that men will have higher wages, all else equal, as indicated by the negative value of the female coefficient of -0.4297. We might observe this pattern in the data because there might be bias against women to pay them less than their male counterparts.

e. Holding all else equal, being white is associated with a 0.96% increase in wage. This value is not significant however because the p value is larger than the threshold significance level of 0.05.

Being black, holding all else equal, is associated with a 19.8% decrease in wages and this value is significant because the p value is smaller than 0.05.

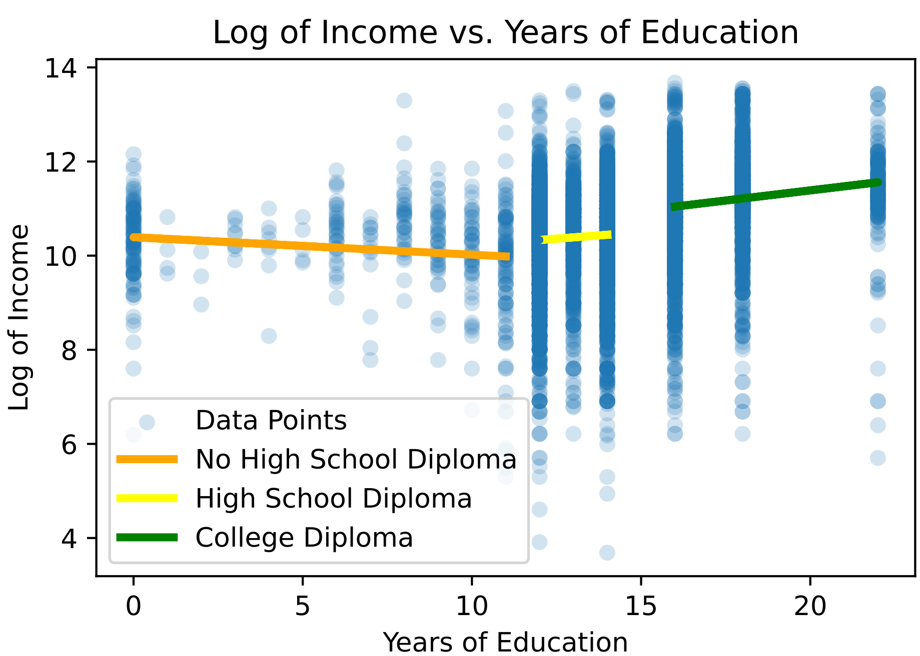
4.

```
## subset no high school (hsdip=0 AND coldip=0)
no_hsd = edu_merged_clean[(edu_merged_clean['hsdip'] == 0) &
 ↪  (edu_merged_clean['coldip'] ==0)]

## fit the linear regression prediction of lnincwage vs education for hsdip=0
y_nhs = no_hsd[['lnincwage']].values
X_nhs = no_hsd[['EDUCDC']].values
y_pred_nhs = lm().fit(X_nhs, y_nhs).predict(X_nhs)
```

```
## subset high school diploma (hsdip=1)
hsd = edu_merged_clean[edu_merged_clean['hsdip'] == 1]

## fit the linear regression prediction of lnincwage vs education for hsdip=0
y1 = hsd[['lnincwage']].values
X1 = hsd[['EDUCDC' ]].values
y_pred_hsd = lm().fit(X1, y1).predict(X1)
```

```
## subset high school diploma (coldip=1)
col = edu_merged_clean[edu_merged_clean['coldip'] == 1]

## fit the linear regression prediction of lnincwage vs education for hsdip=0
y2 = col[['lnincwage']].values
X2 = col[['EDUCDC' ]].values
y_pred_col = lm().fit(X2, y2).predict(X2)
```

```
## graph the plot with the three separate trendlines
fig,ax = plt.subplots()
ax.scatter(edu_merged_clean['EDUCDC'], edu_merged_clean['lnincwage'],
 ↪  alpha=0.2, s=25, label='Data Points')
ax.plot(X_nhs, y_pred_nhs, color="orange", linewidth=3, label='No High School
 ↪  Diploma')
ax.plot(X1, y_pred_hsd, color="yellow", linewidth=3, label='High School
 ↪  Diploma')
ax.plot(X2, y_pred_col, color="green", linewidth=3, label='College Diploma')

ax.set_xlabel("Years of Education")
ax.set_ylabel("Log of Income")
ax.set_title('Log of Income vs. Years of Education')
ax.legend()
```



Question 5

a. The equation is $ln(incwage) = \beta_0 + \beta_1 hsdip + \beta_2 coldip + \beta_3 female + \beta_4 age + \beta_5 age^2 + \beta_6 white + \beta_7 black + \beta_8 hispanic + \beta_9 married + \beta_{10} nchild + \beta_{11} vet + \varepsilon$ I think this is the best possible way to explain the data because it allows for conditioning based on the different education levels which in turn are different combinations of the variables hsdip and coldip.

b.

```
## estimate this model
regression1 = smf.ols('lnincwage ~ hsdip + coldip + female + AGE + age_sq +
↪  white + black + hispanic + married + NCHILD + vet', data =
↪  edu_merged_clean).fit()
print(regression1.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:              lnincwage   R-squared:                       0.314
Model:                            OLS   Adj. R-squared:                  0.313
Method:                 Least Squares   F-statistic:                     360.2
Date:                Fri, 31 Jan 2025   Prob (F-statistic):               0.00
Time:                        01:41:48   Log-Likelihood:                -11294.
No. Observations:                8683   AIC:                         2.261e+04
Df Residuals:                    8671   BIC:                         2.270e+04
Df Model:                          11
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      7.2375      0.114     63.699      0.000       7.015       7.460
hsdip          0.2831      0.046      6.155      0.000       0.193       0.373
coldip         0.8867      0.047     18.718      0.000       0.794       0.980
female        -0.4317      0.019    -22.220      0.000      -0.470      -0.394
AGE            0.1371      0.006     24.872      0.000       0.126       0.148
age_sq        -0.0014   6.49e-05    -21.520      0.000      -0.002      -0.001
white          0.0288      0.026      1.093      0.274      -0.023       0.080
black         -0.1643      0.042     -3.876      0.000      -0.247      -0.081
hispanic      -0.0854      0.031     -2.714      0.007      -0.147      -0.024
married        0.1882      0.022      8.385      0.000       0.144       0.232
NCHILD         0.0053      0.010      0.526      0.599      -0.014       0.025
vet           -0.0985      0.050     -1.974      0.048      -0.196      -0.001
```

```
==============================================================================
Omnibus:                     2281.164   Durbin-Watson:                   1.868
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             8620.565
Skew:                          -1.271   Prob(JB):                         0.00
Kurtosis:                       7.167   Cond. No.                     2.72e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.72e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

c.

```
## predict for 22 yr old female (who is neither white, black, nor Hispanic,
 ↪  is not married, has no children, and is not a veteran) with a high school
 ↪  diploma
hs_22_f = edu_merged_clean.loc[(edu_merged_clean['AGE'] == 22) &
 ↪  (edu_merged_clean['female'] == 1) & (edu_merged_clean['hsdip'] == 1) &
 ↪  (edu_merged_clean['coldip'] == 0)]
pred1 = regression1.get_prediction(hs_22_f)
pred1.summary_frame(alpha=0.05)[:1]

import math
math.exp(9.428)

## predict but with college diploma
col_22_f = edu_merged_clean.loc[(edu_merged_clean['AGE'] == 22) &
 ↪  (edu_merged_clean['female'] == 1) & (edu_merged_clean['hsdip'] == 0) &
 ↪  (edu_merged_clean['coldip'] == 1)]
pred2 = regression1.get_prediction(col_22_f)
pred2.summary_frame(alpha=0.05)[:1]

math.exp(9.946)
```

```
20868.580886763528
```

For a 22 year old female (who is neither white, black, nor Hispanic, is not married, has no children, and is not a veteran) with a high school diploma then their expected wages, will be e^9.428 which is $12,431. For this same individual with a college degree their wage is equal to e^9.946 which is $20,868.

d. Looking a the model, there is a coefficient of 0.8867 for the college diploma term which means that holding all else equal there is on average about an 89% increase in wages for a person with a college degree compared to someone who does not have one. This is statistically significant at the 5% significance level.

e. Given the evidence, I would advise the President to pursue legislation to expand access to college education since it seems to have a large and significant effect on wages even when holding other variables like gender, age, race, married status, and number of children fixed.

f. This new model explains 31% of the variation that's observed in log income which is slightly higher than the previous model. The previous model explained 29.3%.

g. I'm pretty confident in my predictions because the increase in log wages is significant for individuals with a degree and the model explains over 30% of the variation in log wages that is observed in the data. This is pretty good considering the complexity of the relationship that we are trying to explain and the use of various predictors to try and control for other potential influences.

Question 6

```
## library
from sklearn.preprocessing import SplineTransformer
from sklearn.linear_model import LinearRegression

## prepare variables
X_age = edu_merged_clean[['AGE']]
X_hsdip = edu_merged_clean[['hsdip']]
X_coldip = edu_merged_clean[['coldip']]
X_fem = edu_merged_clean[['female']]
X_white = edu_merged_clean[['white']]
X_black = edu_merged_clean[['black']]
X_hispanic = edu_merged_clean[['hispanic']]
X_married = edu_merged_clean[['married']]
X_nchild = edu_merged_clean[['NCHILD']]
X_vet = edu_merged_clean[['vet']]
y = edu_merged_clean['lnincwage']

## knots for the age variable
knots = np.array([18, 65]).reshape(-1, 1)

## spline transformer
```

```
spline_transformer = SplineTransformer(degree=3, knots = knots, include_bias
 ↪  = False)

## transofrm age into spline functions
X_splines = spline_transformer.fit_transform(X_age)

## combine with controls
X = np.hstack([X_splines, X_hsdip, X_coldip, X_fem, X_white, X_black,
 ↪  X_hispanic, X_married, X_nchild, X_vet])

## fit linear reg model
final = LinearRegression()
final.fit(X,y)

## print intercept
print('Intercept:', final.intercept_)

## combine coefficient names
spline_columns = [f"spline_{i}" for i in range(X_splines.shape[1])]
all_columns = spline_columns + ['hsdip'] + ['coldip'] + ['female'] +
 ↪  ['white'] + ['black'] + ['hispanic'] + ['married'] + ['NCHILD'] + ['vet']
coefficients = pd.Series(final.coef_, index=all_columns)
print("Coefficients:")
print(coefficients)

## get adj r^2
# Calculate R^2
r_squared = final.score(X, y)

n = len(y)
p = X_splines.shape[1]
adjusted_r_squared = 1 - ((1 - r_squared) * (n - 1)) / (n - p - 1)
print("Adjusted R-squared:", adjusted_r_squared)
```

```
Intercept: 14.331135251571602
Coefficients:
spline_0   -19.476711
spline_1    -1.780260
spline_2    -5.272285
hsdip        0.273681
coldip       0.854378
```

```
female      -0.428941
white        0.024746
black       -0.166548
hispanic    -0.093290
married      0.181026
NCHILD       0.005171
vet         -0.089764
dtype: float64
Adjusted R-squared: 0.32122736547279296
```

The adjusted R^2 for this model is 0.321

b. This adjusted R^2 is different from the previous model which was 0.313. These are different because the spline is a different model that allows for more flexibility than a regular linear regression so it would make sense that it's ability to explain variation in the y value of the data changes.

c. skip

d. Given the previous spline models with knots at 24 and 55, the values of the predictions for a female with a college diploma would be different because with a spline you are essentially creating mini models that you then stich together (at the knots) smoothly. This allows for more prediction accuracy because the splines behave like linear regression functions but have more flexibility, and therefore can better explain the variation in the data better than a regular OLS model. This is what we saw in the slightly higher adjusted R^2 value.

3c). $\frac{d}{dage}\ln(wage) = 0.1465 - 0.003age$

$0.1465 - 0.003age = 0$

$0.003age = 0.1465$

$age = \frac{0.1465}{0.003}$

$= 48.8\overline{3}$