# Project Work

**Cande Torres and Kashvi Ajitsaria**

```
library(ISLR)
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.0 ──
```

```
## ✓ ggplot2 3.3.3     ✓ purrr   0.3.4
## ✓ tibble  3.1.0     ✓ dplyr   1.0.5
## ✓ tidyr   1.1.3     ✓ stringr 1.4.0
## ✓ readr   1.4.0     ✓ forcats 0.5.1
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidyr)
library(readr)
library(ggplot2)
library(dplyr)
library(forcats)
library(carData)
library(class)
library(devtools)
```

```
## Loading required package: usethis
```

```
library(gam)
```

```
## Loading required package: splines
```

```
## Loading required package: foreach
```

```
##
## Attaching package: 'foreach'
```

```
## The following objects are masked from 'package:purrr':
##
##     accumulate, when
```

```
## Loaded gam 1.20
```

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loaded glmnet 4.1-1
```

```
library(leaps)
library(methods)
library(openxlsx)
library(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':
##
##     discard
```

```
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
library(splines)
library(stats)
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```
library(broom)
```

```
#library(readr)
#data2019 <- read_csv("STATS_project_data2019.csv")
#View(data2019)
```

```
library(readxl)
data2019 <- read_excel("STATS_project_data2019.xlsx")
```

```
# Remove the variables we don't want
data2019 <- data2019 %>%
    select(-ANO4, -TRIMESTRE,-COMPONENTE,-CODUSU,-NRO_HOGAR,-Birth_Loc_Specific, -Loc
ation_5y_Specific,-Read_write_recode,-Schooling_recode,-NIVEL_ED_recoded,-ESTADO,-CAT
_INAC,-Income_job,-Income_scholarship, -Income_no_work, -IPCF,-NIVEL_ED,-Schooling,-R
ead_write)
```

```
#Recode Vars taken as integer as categorical
data2019$CAT_OCUP <- as.factor(data2019$CAT_OCUP)
data2019$Sex <- as.factor(data2019$Sex)
data2019$REGION <- as.factor(data2019$REGION)
data2019$Relative_Rel <- as.factor(data2019$Relative_Rel)
data2019$Civil_State <- as.factor(data2019$Civil_State)
#data2019$Schooling <- as.factor(data2019$Schooling)
#data2019$NIVEL_ED <- as.factor(data2019$NIVEL_ED)
data2019$Highest_level <- as.factor(data2019$Highest_level)
data2019$Type_of_school <- as.factor(data2019$Type_of_school)
data2019$`finished?` <- as.factor(data2019$`finished?`)
data2019$Birth_Location <- as.factor(data2019$Birth_Location)
data2019$Ownership <- as.factor(data2019$Ownership)
data2019$House_Type <- as.factor(data2019$House_Type)
data2019$Studio <- as.factor(data2019$Studio)
data2019$IH_II_01 <- as.factor(data2019$IH_II_01)
data2019$IH_II_02 <- as.factor(data2019$IH_II_02)
data2019$IP_III_04 <- as.factor(data2019$IP_III_04)
data2019$IP_III_05 <- as.factor(data2019$IP_III_05)
data2019$IP_III_06 <- as.factor(data2019$IP_III_06)
```

```
# Rename column where names is code
names(data2019)[names(data2019) == "IH_II_01"] <- "Computer_house"
names(data2019)[names(data2019) == "IH_II_02"] <- "Internet_house"
names(data2019)[names(data2019) == "IP_III_04"] <- "Internet_use"
names(data2019)[names(data2019) == "IP_III_05"] <- "Computer_use"
names(data2019)[names(data2019) == "IP_III_06"] <- "Cellphone_use"
```

# Initial investigation 1: ignoring nonlinearity (for now)

Use ordinary least squares (OLS) regression, forward and/or backward selection, and LASSO to build initial models for your quantitative outcome as a function of the predictors of interest. (As part of data cleaning, exclude any variables that you don't want to consider as predictors.)

- These models should not include any transformations to deal with nonlinearity. You'll explore this in the next investigation.
- Note: If you have highly collinear/redundant variables, you might see the message "Reordering variables and trying again" and associated `warning()` s about linear dependencies being found. Sometimes stepwise selection is able to handle the collinearity/redundancy by modifying the order of the variables tried. If collinearity/redundancy cannot be handled and causes an error, try reducing `nvmax` .

##Note It is noteworthy the fact that we started fitting the models with individual's income as outcome variable in the natural scale. Only when we got to the residuals plot we realized we needed to log transform the variable to account for non-continuous distribution of residuals or heteroskedasticity. We log transformed the variable and re-traced the process but for the sake of interpretability, we are stiking with the natural scale for now.

That's completely fine.

### OLS

*Note: We kept the original OLS models to compare residual plots bellow

```
mod1 <- lm(Income_individual ~ Literacy_Index+Sex+Birth_Location+CAT_OCUP+Civil_State
+Highest_level+Cellphone_use+Internet_use+ Computer_use+ Age, data = data2019)
summary(mod1)
```

```
##
## Call:
## lm(formula = Income_individual ~ Literacy_Index + Sex + Birth_Location +
##     CAT_OCUP + Civil_State + Highest_level + Cellphone_use +
##     Internet_use + Computer_use + Age, data = data2019)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
##  -73464   -8292   -1112    5111 1642734
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -3560.759   1348.796  -2.640 0.008294 **
## Literacy_Index    2047.797    227.197   9.013  < 2e-16 ***
## Sex2             -4561.713    200.031 -22.805  < 2e-16 ***
## Birth_Location2   1268.776    336.442   3.771 0.000163 ***
## Birth_Location3   2341.698    288.152   8.127 4.52e-16 ***
## Birth_Location4   -465.529    595.578  -0.782 0.434429
## Birth_Location5  -2075.497    942.160  -2.203 0.027605 *
## Birth_Location9  -5622.637   6125.785  -0.918 0.358695
## CAT_OCUP1        17333.102    702.696  24.667  < 2e-16 ***
## CAT_OCUP2         5822.716    331.584  17.560  < 2e-16 ***
## CAT_OCUP3        13199.429    229.878  57.419  < 2e-16 ***
## CAT_OCUP4        -3394.074   1685.685  -2.013 0.044071 *
## CAT_OCUP9        -2725.913  21367.652  -0.128 0.898488
## Civil_State2      1176.670    306.090   3.844 0.000121 ***
## Civil_State3      2322.998    437.821   5.306 1.13e-07 ***
## Civil_State4      3871.796    524.968   7.375 1.67e-13 ***
## Civil_State5     -3608.130    291.161 -12.392  < 2e-16 ***
## Civil_State9     -2558.198  12777.654  -0.200 0.841318
## Highest_level1    2286.799   3439.338   0.665 0.506121
## Highest_level2   -4622.205   1495.991  -3.090 0.002005 **
```

```
## Highest_level3   -3559.696   1764.270  -2.018 0.043632 *  
## Highest_level4   -6258.179   1763.005  -3.550 0.000386 ***
## Highest_level5   -4557.700   1963.588  -2.321 0.020285 *  
## Highest_level6   -6501.692   2134.393  -3.046 0.002319 ** 
## Highest_level7   -3082.126   2115.504  -1.457 0.145144    
## Highest_level8   22430.338   2446.080   9.170  < 2e-16 ***
## Highest_level9    3140.726   1898.742   1.654 0.098112 .  
## Highest_level99 -16965.338   9336.962  -1.817 0.069222 .  
## Cellphone_use2     369.912    375.402   0.985 0.324445    
## Cellphone_use9    1985.503  12296.952   0.161 0.871729    
## Internet_use2    -2208.172    332.085  -6.649 2.97e-11 ***
## Internet_use9    -5744.849   7277.502  -0.789 0.429883    
## Computer_use2    -3221.570    226.905 -14.198  < 2e-16 ***
## Computer_use9    -3343.862   4459.193  -0.750 0.453330    
## Age                277.369      7.586  36.561  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 21370 on 49944 degrees of freedom
##   (23 observations deleted due to missingness)
## Multiple R-squared:  0.2363, Adjusted R-squared:  0.2358 
## F-statistic: 454.5 on 34 and 49944 DF,  p-value: < 2.2e-16
```

```
mod1_output <- broom::augment(mod1, newdata = data2019)
head(mod1_output)
```

```
## # A tibble: 6 x 34
##   Ind_Interview REGION MAS_500 AGLOMERADO Relative_Rel Sex     Age Civil_State
##           <dbl> <fct>  <chr>        <dbl> <fct>        <fct> <dbl> <fct>      
## 1             1 43     S                2 1            1        44 3          
## 2             1 43     S                2 1            1        59 2          
## 3             1 43     S                2 2            2        62 2          
## 4             1 43     S                2 3            1        26 5          
## 5             1 43     S                2 3            1        23 5          
## 6             1 43     S                2 1            2        26 5          
## # … with 26 more variables: Type_of_school <fct>, Highest_level <fct>,
## #   finished? <fct>, last_yr_approved <chr>, Birth_Location <fct>,
## #   Location_5y <dbl>, CAT_OCUP <fct>, JOB_N <dbl>, Income_individual <dbl>,
## #   ITF <dbl>, person_id <dbl>, Internet_use <fct>, Computer_use <fct>,
## #   Cellphone_use <fct>, House_Type <fct>, Room_N <dbl>, Ownership <fct>,
## #   Self_Room <dbl>, Self_Room_Sleep <dbl>, Studio <fct>, Studio_N <dbl>,
## #   Computer_house <fct>, Internet_house <fct>, Literacy_Index <dbl>,
## #   .fitted <dbl>, .resid <dbl>
```

### *Subset Selection: Backward stepwise selection*

```
set.seed(25)

back_step_mod <- train(
    Income_individual~ Literacy_Index+Sex+Birth_Location+CAT_OCUP+Civil_State+Highest
_level+Cellphone_use+Internet_use+ Computer_use+Age,
    data = data2019,
    method = "leapBackward",
    tuneGrid = data.frame(nvmax = 1:10),
    trControl = trainControl(method = "cv", number = 10, selectionFunction = "oneSE")
,
    metric = "MAE",
    na.action = na.omit
)
```

```
## Warning in leaps.setup(x, y, wt = weights, nbest = nbest, nvmax = nvmax, : 1
## linear dependencies found
```

```
## Reordering variables and trying again:
```

```
summary(back_step_mod)
```

```
## Subset selection object
## 34 Variables  (and intercept)
##                  Forced in Forced out
## Literacy_Index      FALSE      FALSE
## Sex2                FALSE      FALSE
## Birth_Location2     FALSE      FALSE
## Birth_Location3     FALSE      FALSE
## Birth_Location4     FALSE      FALSE
## Birth_Location5     FALSE      FALSE
## Birth_Location9     FALSE      FALSE
## CAT_OCUP1           FALSE      FALSE
## CAT_OCUP2           FALSE      FALSE
## CAT_OCUP3           FALSE      FALSE
## CAT_OCUP4           FALSE      FALSE
## CAT_OCUP9           FALSE      FALSE
## Civil_State2        FALSE      FALSE
## Civil_State3        FALSE      FALSE
## Civil_State4        FALSE      FALSE
## Civil_State5        FALSE      FALSE
## Civil_State9        FALSE      FALSE
## Highest_level1      FALSE      FALSE
## Highest_level2      FALSE      FALSE
## Highest_level3      FALSE      FALSE
```

```
## Highest_level4      FALSE      FALSE
## Highest_level5      FALSE      FALSE
## Highest_level6      FALSE      FALSE
## Highest_level7      FALSE      FALSE
## Highest_level8      FALSE      FALSE
## Highest_level9      FALSE      FALSE
## Highest_level99     FALSE      FALSE
## Cellphone_use2      FALSE      FALSE
## Cellphone_use9      FALSE      FALSE
## Internet_use2       FALSE      FALSE
## Internet_use9       FALSE      FALSE
## Computer_use2       FALSE      FALSE
## Computer_use9       FALSE      FALSE
## Age                 FALSE      FALSE
## 1 subsets of each size up to 7
## Selection Algorithm: backward
##          Literacy_Index Sex2 Birth_Location2 Birth_Location3 Birth_Location4
## 1  ( 1 ) " "            " "  " "             " "             " "
## 2  ( 1 ) " "            " "  " "             " "             " "
## 3  ( 1 ) "*"            " "  " "             " "             " "
## 4  ( 1 ) "*"            " "  " "             " "             " "
## 5  ( 1 ) "*"            " "  " "             " "             " "
## 6  ( 1 ) "*"            "*"  " "             " "             " "
## 7  ( 1 ) "*"            "*"  " "             " "             " "
##          Birth_Location5 Birth_Location9 CAT_OCUP1 CAT_OCUP2 CAT_OCUP3
## 1  ( 1 ) " "             " "             " "       " "       "*"
## 2  ( 1 ) " "             " "             " "       " "       "*"
## 3  ( 1 ) " "             " "             " "       " "       "*"
## 4  ( 1 ) " "             " "             "*"       " "       "*"
## 5  ( 1 ) " "             " "             "*"       " "       "*"
## 6  ( 1 ) " "             " "             "*"       " "       "*"
## 7  ( 1 ) " "             " "             "*"       "*"       "*"
##          CAT_OCUP4 CAT_OCUP9 Civil_State2 Civil_State3 Civil_State4
## 1  ( 1 ) " "       " "       " "          " "          " "
## 2  ( 1 ) " "       " "       " "          " "          " "
## 3  ( 1 ) " "       " "       " "          " "          " "
## 4  ( 1 ) " "       " "       " "          " "          " "
## 5  ( 1 ) " "       " "       " "          " "          " "
## 6  ( 1 ) " "       " "       " "          " "          " "
## 7  ( 1 ) " "       " "       " "          " "          " "
##          Civil_State5 Civil_State9 Highest_level1 Highest_level2 Highest_level3
## 1  ( 1 ) " "          " "          " "            " "            " "
## 2  ( 1 ) " "          " "          " "            " "            " "
## 3  ( 1 ) " "          " "          " "            " "            " "
## 4  ( 1 ) " "          " "          " "            " "            " "
## 5  ( 1 ) " "          " "          " "            " "            " "
## 6  ( 1 ) " "          " "          " "            " "            " "
```

```
## 7  ( 1 ) " "           " "           " "             " "             " "
##          Highest_level4 Highest_level5 Highest_level6 Highest_level7
## 1  ( 1 ) " "            " "            " "            " "
## 2  ( 1 ) " "            " "            " "            " "
## 3  ( 1 ) " "            " "            " "            " "
## 4  ( 1 ) " "            " "            " "            " "
## 5  ( 1 ) " "            " "            " "            " "
## 6  ( 1 ) " "            " "            " "            " "
## 7  ( 1 ) " "            " "            " "            " "
##          Highest_level8 Highest_level9 Highest_level99 Cellphone_use2
## 1  ( 1 ) " "            " "            " "             " "
## 2  ( 1 ) " "            " "            " "             " "
## 3  ( 1 ) " "            " "            " "             " "
## 4  ( 1 ) " "            " "            " "             " "
## 5  ( 1 ) "*"            " "            " "             " "
## 6  ( 1 ) "*"            " "            " "             " "
## 7  ( 1 ) "*"            " "            " "             " "
##          Cellphone_use9 Internet_use2 Internet_use9 Computer_use2 Computer_use9
## 1  ( 1 ) " "            " "           " "           " "           " "
## 2  ( 1 ) " "            " "           " "           " "           " "
## 3  ( 1 ) " "            " "           " "           " "           " "
## 4  ( 1 ) " "            " "           " "           " "           " "
## 5  ( 1 ) " "            " "           " "           " "           " "
## 6  ( 1 ) " "            " "           " "           " "           " "
## 7  ( 1 ) " "            " "           " "           " "           " "
##          Age
## 1  ( 1 ) " "
## 2  ( 1 ) "*"
## 3  ( 1 ) "*"
## 4  ( 1 ) "*"
## 5  ( 1 ) "*"
## 6  ( 1 ) "*"
## 7  ( 1 ) "*"
```

```
coef(back_step_mod$finalModel, id = back_step_mod$bestTune$nvmax)
```

```
##    (Intercept) Literacy_Index           Sex2      CAT_OCUP1      CAT_OCUP2
##    -17390.0283      2466.8234     -4155.8548     19185.0613      6416.4241
##      CAT_OCUP3 Highest_level8            Age
##     13965.8905     28848.0022       351.3907
```

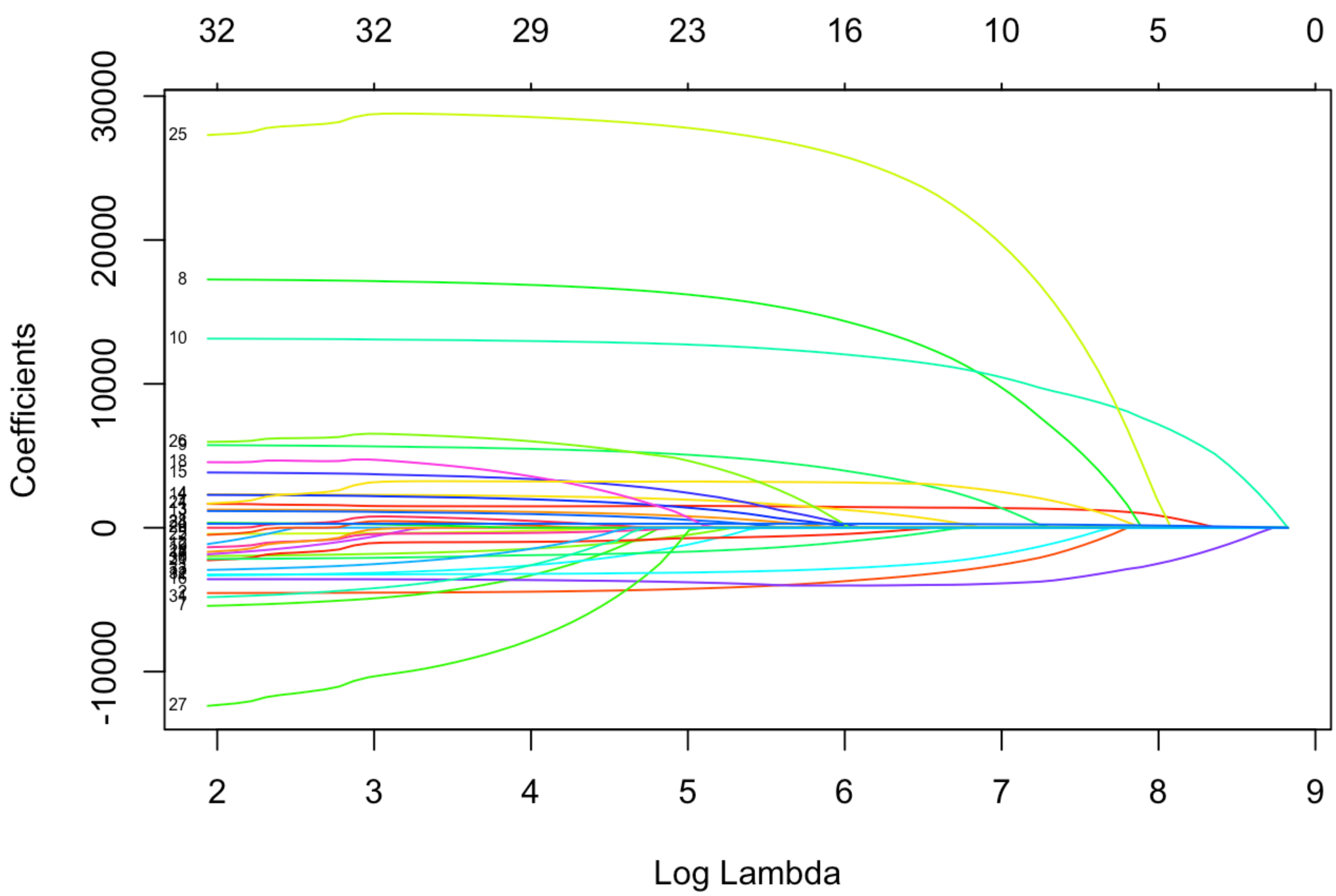
### LASSO

*Note to self: You can also look at the two forms of penalized models with this tuneGrid: ridge regression and lasso regression. alpha = 0 is pure ridge regression, and alpha = 1 is pure lasso regression. You can fit a mixture of the two models (i.e. an elastic net) using an alpha between 0 and 1. For example, alpha = 0.05* *would be 95% ridge regression and 5% lasso regression.* Note to self: According to the internet, the sequence is the amount of lambda values we explore, starting from a particular number, finishing in a different one. The jumps/steps (amount of models we fit) is defined by length.out and it usually jumps with multiples of 10.

```
set.seed(253)
lasso_mod<- train(
    Income_individual ~ Literacy_Index+Sex+Birth_Location+CAT_OCUP+Civil_State+Highes
t_level+Cellphone_use+Internet_use+ Computer_use+Age,
    data = data2019,
    method = "glmnet",
    trControl = trainControl(method = "cv", number = 10, selectionFunction = "oneSE")
,
    tuneGrid = data.frame(alpha = 1, lambda = seq(0,1000, length.out = 100)),
    metric = "MAE",
    na.action = na.omit
)
```

##Examine

```
# Plot coefficient paths as a function of lambda
plot(lasso_mod$finalModel, xvar = "lambda", label = TRUE, col = rainbow(20))
```




```
# Codebook for which variables the numbers correspond to
rownames(lasso_mod$finalModel$beta)
```

```
##  [1] "Literacy_Index"   "Sex2"             "Birth_Location2" "Birth_Location3"
##  [5] "Birth_Location4"  "Birth_Location5"  "Birth_Location9" "CAT_OCUP1"
##  [9] "CAT_OCUP2"        "CAT_OCUP3"        "CAT_OCUP4"       "CAT_OCUP9"
## [13] "Civil_State2"     "Civil_State3"     "Civil_State4"    "Civil_State5"
## [17] "Civil_State9"     "Highest_level1"   "Highest_level2"  "Highest_level3"
## [21] "Highest_level4"   "Highest_level5"   "Highest_level6"  "Highest_level7"
## [25] "Highest_level8"   "Highest_level9"   "Highest_level99" "Cellphone_use2"
## [29] "Cellphone_use9"   "Internet_use2"    "Internet_use9"   "Computer_use2"
## [33] "Computer_use9"    "Age"
```

**PUT ANY RELEVANT TEXT/RESPONSES/INTERPRETATIONS HERE** Interpretation OLS: -The average value of Individual Income when all the predictors are set to 0 corresponds to the intercept ARG$8056.2. -Then, we can say that for one unit increase in the the Literacy index we associate an increase of ARG$1514.6 in individual's income, holding all other predictors constant. Interpretation for Stepwise Selection: -The best model choosen with oneSE is the one with 7 predictors. In the one stage, the predictor remaining is

CAT_OCUP3, which corresponds to the category employee/worker. Whether a person belongs or not to that category seems to impact their income greatly, which makes sense. The second most important predictor seems to be Age. The third is Literacy, as I suspected, and the fourth is CAT_OCUP1 which signals whether a person has a leadership role in their job or not. Additionally, the model picked up Highest_level8 as the fifth predictor, which corresponds to having attended to Graduate School (Masters, MBA, PhD). In general for backward step-wise selection, variables removed first could be viewed as the least important predictors, and variable that remain until the end could be viewed as the most important predictors. Interpretation for Lasso: From just examining the plot above we can see that some of the \

Estimate test performance of the models from these different methods. Report and interpret (with units) these estimates along with a measure of uncertainty in the estimate (SD is most readily available from `caret`).

- Compare estimated test performance across methods. Which method(s) might you prefer?

Test Metrics For OLS

Be very careful! These are *training* estimates of performance! You should use caret_mod$results to obtain the cross-validated estimates of test performance. We use the following code to **graphically** explore training set residuals (because we don't have a test set):
data %>%
  mutate(pred = predict(…), residuals = Y - pred)

You should still use caret to fit the OLS model (method = "lm") and look at $results to get estimates of test performance.

```
# Your code
mean(abs(mod1_output$.resid),na.rm=TRUE)
```

```
## [1] 10812.28
```

```
mean(mod1_output$.resid^2,na.rm=TRUE)
```

```
## [1] 456170460
```

Test Metrics for Stepwise Selection

```
# Look at accuracy/error metrics for the different subset sizes
# If you want to sort the table of results, use arrange() from dplyr
back_step_mod$results %>% arrange(MAE)
```

```
##    nvmax     RMSE   Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 1     10 21234.72 0.23416517 11042.38 4116.832 0.06153874 694.4541
## 2      9 21261.29 0.23218020 11089.10 4118.770 0.06253801 712.8859
## 3      8 21310.35 0.22856637 11130.06 4116.526 0.06126135 701.9552
## 4      7 21373.63 0.22381625 11192.71 4098.222 0.05895647 679.9858
## 5      6 21478.07 0.21599135 11305.27 4094.588 0.05893365 704.3734
## 6      5 21656.98 0.20287251 11400.23 4117.168 0.05790163 684.0452
## 7      4 21774.16 0.19398937 11483.82 4094.982 0.05395613 654.8115
## 8      2 22275.11 0.15537183 11509.24 4056.190 0.04510042 683.1004
## 9      3 21879.39 0.18593402 11607.97 4096.273 0.05188570 653.6835
## 10     1 23175.36 0.08371103 13192.98 3954.987 0.02081807 169.9558
```

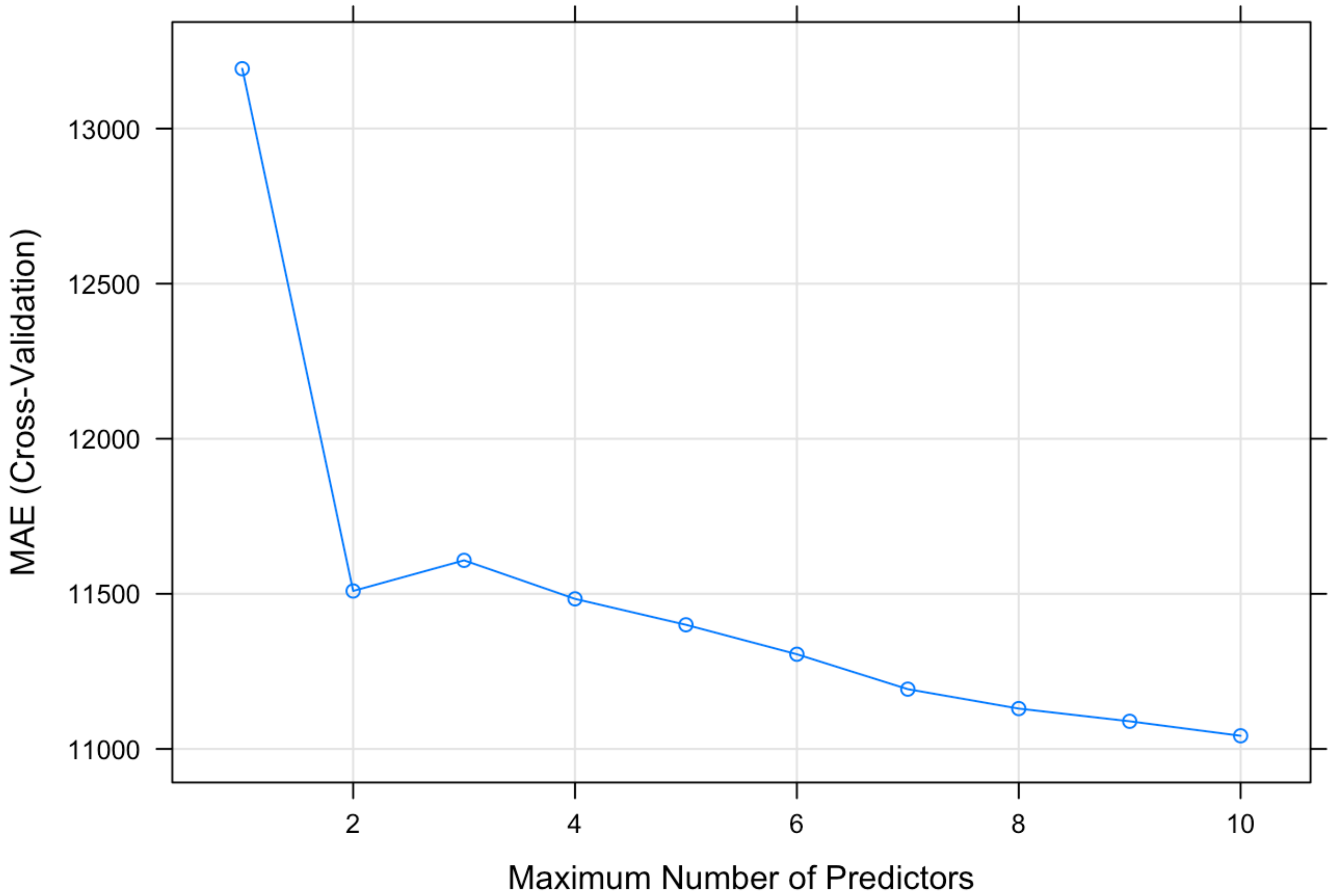
```
# What tuning parameter gave the best performance?
# i.e. What subset size gave the best model?
back_step_mod$bestTune
```

```
##   nvmax
## 7     7
```

```
# Obtain the coefficients for the best model
coef(back_step_mod$finalModel, id = back_step_mod$bestTune$nvmax)
```

```
##    (Intercept) Literacy_Index           Sex2      CAT_OCUP1      CAT_OCUP2
##    -17390.0283      2466.8234     -4155.8548     19185.0613      6416.4241
##      CAT_OCUP3 Highest_level8            Age
##     13965.8905     28848.0022       351.3907
```

```
# Plot metrics for each model in the sequence
plot(back_step_mod)
```




Test Metrics for LASSO

```
lasso_mod$results %>% arrange(MAE)
```

```
##    alpha    lambda     RMSE  Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 1      1 353.53535 21136.12 0.2442148 10734.64 3748.399 0.04413983 140.0113
## 2      1 363.63636 21138.36 0.2441160 10734.68 3749.031 0.04414769 140.2292
## 3      1 343.43434 21133.91 0.2443123 10734.75 3747.767 0.04413147 139.8606
## 4      1 373.73737 21140.68 0.2440137 10734.86 3749.654 0.04415511 140.4286
## 5      1 333.33333 21131.69 0.2444124 10734.98 3747.134 0.04412345 139.6634
## 6      1 383.83838 21143.00 0.2439128 10735.12 3750.263 0.04416282 140.6718
## 7      1 393.93939 21145.28 0.2438163 10735.38 3750.803 0.04416733 140.9472
## 8      1 323.23232 21129.54 0.2445090 10735.49 3746.491 0.04411487 139.3761
## 9      1 404.04040 21147.62 0.2437167 10735.78 3751.333 0.04417125 141.2440
## 10     1 313.13131 21127.46 0.2446018 10736.14 3745.838 0.04410584 139.0979
## 11     1 414.14141 21150.03 0.2436140 10736.40 3751.849 0.04417440 141.4737
## 12     1 424.24242 21152.06 0.2435438 10736.60 3752.435 0.04418775 141.6936
## 13     1 434.34343 21153.96 0.2434852 10736.68 3753.003 0.04420012 141.9160
## 14     1 444.44444 21155.91 0.2434246 10736.86 3753.564 0.04421229 142.1507
```

```
## 15       1  303.03030 21125.46 0.2446905 10736.93 3745.179 0.04409638 138.8288
## 16       1  454.54545 21157.91 0.2433622 10737.11 3754.114 0.04422410 142.3883
## 17       1  464.64646 21159.85 0.2433065 10737.25 3754.693 0.04424061 142.6311
## 18       1  474.74747 21161.82 0.2432506 10737.43 3755.276 0.04425826 142.9081
## 19       1  484.84848 21163.83 0.2431930 10737.78 3755.852 0.04427573 143.1185
## 20       1  292.92929 21123.53 0.2447756 10737.89 3744.510 0.04408648 138.6221
## 21       1  494.94949 21165.90 0.2431336 10738.33 3756.421 0.04429304 143.2672
## 22       1  282.82828 21121.64 0.2448593 10738.91 3743.835 0.04407632 138.3943
## 23       1  505.05051 21168.00 0.2430724 10739.00 3756.984 0.04431001 143.4385
## 24       1  515.15152 21170.16 0.2430093 10739.73 3757.540 0.04432670 143.6462
## 25       1  272.72727 21119.79 0.2449420 10740.06 3743.156 0.04406618 138.1469
## 26       1  525.25253 21172.36 0.2429443 10740.52 3758.089 0.04434332 143.8561
## 27       1  262.62626 21118.00 0.2450216 10741.38 3742.466 0.04405516 137.8853
## 28       1  535.35354 21174.60 0.2428775 10741.40 3758.632 0.04435975 144.1153
## 29       1  545.45455 21176.89 0.2428090 10742.37 3759.169 0.04437601 144.3588
## 30       1  252.52525 21116.18 0.2451059 10742.75 3741.759 0.04404210 137.4931
## 31       1  555.55556 21179.23 0.2427386 10743.49 3759.701 0.04439219 144.5644
## 32       1  242.42424 21114.42 0.2451873 10744.22 3741.041 0.04402866 137.1019
## 33       1  565.65657 21181.60 0.2426666 10744.76 3760.225 0.04440819 144.7521
## 34       1  232.32323 21112.71 0.2452670 10745.75 3740.322 0.04401642 136.7810
## 35       1  575.75758 21184.02 0.2425928 10746.14 3760.742 0.04442401 144.9776
## 36       1  222.22222 21111.06 0.2453438 10747.38 3739.608 0.04400567 136.5231
## 37       1  585.85859 21186.48 0.2425171 10747.62 3761.254 0.04443963 145.2159
## 38       1  212.12121 21109.49 0.2454171 10749.12 3738.896 0.04399544 136.2439
## 39       1  595.95960 21188.99 0.2424394 10749.18 3761.759 0.04445506 145.4468
## 40       1  606.06061 21191.54 0.2423600 10750.79 3762.257 0.04447014 145.6768
## 41       1  202.02020 21107.99 0.2454872 10750.98 3738.180 0.04398523 135.9497
## 42       1  616.16162 21194.14 0.2422786 10752.46 3762.750 0.04448493 145.9111
## 43       1  191.91919 21106.57 0.2455530 10753.03 3737.440 0.04397329 135.6391
## 44       1  626.26263 21196.78 0.2421955 10754.22 3763.239 0.04449942 146.1270
## 45       1  181.81818 21105.25 0.2456138 10755.34 3736.681 0.04395985 135.3046
## 46       1  636.36364 21199.47 0.2421104 10756.07 3763.721 0.04451370 146.3418
## 47       1  171.71717 21104.02 0.2456706 10757.86 3735.929 0.04394591 135.0323
## 48       1  646.46465 21202.20 0.2420233 10757.98 3764.197 0.04452778 146.5788
## 49       1  656.56566 21204.97 0.2419342 10759.93 3764.668 0.04454164 146.8141
## 50       1  161.61616 21102.88 0.2457224 10760.54 3735.127 0.04392965 134.7558
## 51       1  666.66667 21207.76 0.2418452 10761.86 3765.131 0.04455559 146.9826
## 52       1  151.51515 21101.82 0.2457692 10763.41 3734.231 0.04390884 134.4986
## 53       1  676.76768 21210.56 0.2417576 10763.70 3765.586 0.04456910 147.1666
## 54       1  686.86869 21213.40 0.2416686 10765.57 3766.036 0.04458203 147.3651
## 55       1  141.41414 21100.85 0.2458119 10766.48 3733.353 0.04389021 134.2319
## 56       1  696.96970 21216.27 0.2415777 10767.51 3766.480 0.04459473 147.5634
## 57       1  707.07071 21219.20 0.2414847 10769.49 3766.918 0.04460720 147.7523
## 58       1  131.31313 21099.95 0.2458528 10769.70 3732.435 0.04387021 133.9238
## 59       1  717.17172 21222.16 0.2413896 10771.52 3767.350 0.04461943 147.9402
## 60       1  121.21212 21099.10 0.2458914 10773.01 3731.484 0.04384837 133.5850
## 61       1  727.27273 21225.17 0.2412927 10773.61 3767.777 0.04463162 148.1576
```

```
## 62      1  737.37374 21228.18 0.2411970 10775.55 3768.217 0.04464550 148.4159
## 63      1  111.11111 21098.35 0.2459254 10776.49 3730.526 0.04382536 133.2271
## 64      1  747.47475 21231.22 0.2410998 10777.50 3768.649 0.04465889 148.6806
## 65      1  757.57576 21234.31 0.2410005 10779.51 3769.074 0.04467207 148.9345
## 66      1  101.01010 21097.65 0.2459580 10780.18 3729.551 0.04380269 132.8216
## 67      1  767.67677 21237.44 0.2408991 10781.57 3769.494 0.04468503 149.1798
## 68      1  777.77778 21240.61 0.2407957 10783.69 3769.908 0.04469776 149.4110
## 69      1   90.90909 21097.03 0.2459875 10784.08 3728.550 0.04377945 132.5066
## 70      1  787.87879 21243.83 0.2406901 10785.86 3770.316 0.04471026 149.6487
## 71      1  797.97980 21247.08 0.2405828 10788.10 3770.719 0.04472267 149.8873
## 72      1   80.80808 21096.50 0.2460122 10788.23 3727.536 0.04375539 132.2185
## 73      1  808.08081 21250.31 0.2404797 10790.33 3771.144 0.04473791 150.1191
## 74      1   70.70707 21096.09 0.2460316 10792.53 3726.474 0.04372922 131.9100
## 75      1  818.18182 21253.55 0.2403768 10792.59 3771.565 0.04475265 150.3218
## 76      1  828.28283 21256.84 0.2402719 10794.90 3771.980 0.04476718 150.5345
## 77      1   60.60606 21095.76 0.2460476 10796.97 3725.414 0.04370419 131.6378
## 78      1  838.38384 21260.16 0.2401649 10797.30 3772.390 0.04478151 150.7451
## 79      1  848.48485 21263.53 0.2400559 10799.77 3772.794 0.04479564 150.9567
## 80      1   50.50505 21095.57 0.2460561 10801.62 3724.332 0.04368035 131.4656
## 81      1  858.58586 21266.94 0.2399447 10802.31 3773.193 0.04480956 151.1812
## 82      1  868.68687 21270.39 0.2398315 10804.92 3773.586 0.04482328 151.4501
## 83      1   40.40404 21095.50 0.2460598 10806.24 3723.251 0.04365903 131.3424
## 84      1  878.78788 21273.81 0.2397225 10807.54 3773.955 0.04483603 151.6915
## 85      1  888.88889 21277.20 0.2396173 10810.21 3774.331 0.04485028 151.9017
## 86      1   30.30303 21095.55 0.2460568 10810.89 3722.146 0.04363796 131.1841
## 87      1  898.98990 21280.61 0.2395117 10812.95 3774.700 0.04486331 152.1072
## 88      1   20.20202 21095.75 0.2460455 10815.20 3720.997 0.04361708 131.0584
## 89      1  909.09091 21284.06 0.2394042 10815.81 3775.064 0.04487612 152.2947
## 90      1  919.19192 21287.55 0.2392946 10818.75 3775.422 0.04488872 152.5258
## 91      1   10.10101 21095.59 0.2460656 10819.79 3719.913 0.04360682 131.3089
## 92      1    0.00000 21095.60 0.2460789 10821.63 3719.620 0.04361118 131.4670
## 93      1  929.29293 21291.07 0.2391830 10821.74 3775.775 0.04490108 152.7773
## 94      1  939.39394 21294.64 0.2390693 10824.85 3776.122 0.04491322 153.0055
## 95      1  949.49495 21298.25 0.2389535 10828.10 3776.464 0.04492513 153.2316
## 96      1  959.59596 21301.89 0.2388363 10831.49 3776.799 0.04493677 153.4453
## 97      1  969.69697 21305.48 0.2387245 10834.88 3777.147 0.04495154 153.7328
## 98      1  979.79798 21309.08 0.2386141 10838.29 3777.457 0.04496357 154.0283
## 99      1  989.89899 21312.70 0.2385027 10841.79 3777.764 0.04497493 154.3151
## 100     1 1000.00000 21316.36 0.2383892 10845.43 3778.066 0.04498608 154.5771
```

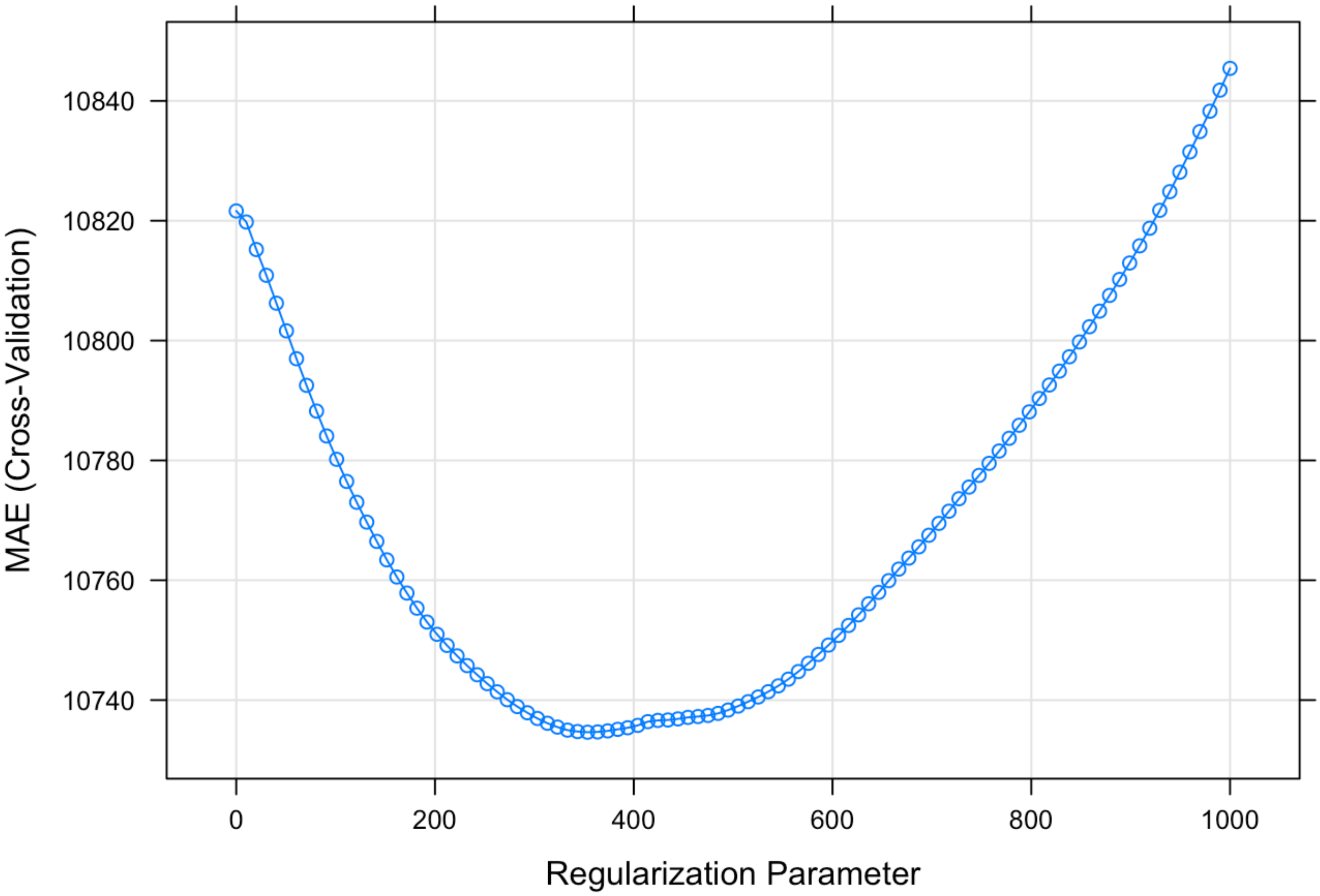
```
# Identify which tuning parameter (lambda) is "best"
lasso_mod$bestTune
```

```
##    alpha   lambda
## 75     1 747.4747
```

```
#Same thing coded differently, we look at the coefficients for the best lambda model
#coef(lasso_mod_log$finalModel, 747)
coef(lasso_mod$finalModel, lasso_mod$bestTune$lambda)
```

```
## 35 x 1 sparse Matrix of class "dgCMatrix"
##                           1
## (Intercept)     -3936.9274
## Literacy_Index   1396.6896
## Sex2            -3123.5300
## Birth_Location2      .
## Birth_Location3   500.1259
## Birth_Location4      .
## Birth_Location5      .
## Birth_Location9      .
## CAT_OCUP1       12093.0945
## CAT_OCUP2        2685.7548
## CAT_OCUP3       11279.9739
## CAT_OCUP4            .
## CAT_OCUP9            .
## Civil_State2         .
## Civil_State3         .
## Civil_State4         .
## Civil_State5    -3970.3985
## Civil_State9         .
## Highest_level1       .
## Highest_level2       .
## Highest_level3       .
## Highest_level4       .
## Highest_level5       .
## Highest_level6       .
## Highest_level7   2973.7376
## Highest_level8  22923.2886
## Highest_level9       .
## Highest_level99      .
## Cellphone_use2       .
## Cellphone_use9       .
## Internet_use2    -195.3898
## Internet_use9        .
## Computer_use2   -2392.9450
## Computer_use9        .
## Age               259.7562
```

```
# Plot a summary of the performance of the different models
plot(lasso_mod)
```




**PUT ANY RELEVANT TEXT/RESPONSES/INTERPRETATIONS HERE**

How do the error metrics compare? OLS: The MAE for the 10-variable OLS is 10812.28.This means that on average this OLS model is off in its percentage predictions about a person's income by about 10812 pesos.

Step-wise: The MAE for the 10-variable step-wise model is 11042.38. This means that on average this OLS model is off in its percentage predictions about a person's income by about 11042.38 pesos.

Lasso: Based on the information in `lasso_mod$results` , we can see that the lowest estimated test MAE (10734.64) shows up for lambda = 353.5. However, the algorithm decided to set the "best" $\lambda = 747$ model with a MAE of 10777.50 pesos. This means that on average this LASSO model is off in its percentage predictions about a person's income by about 10778 pesos. \

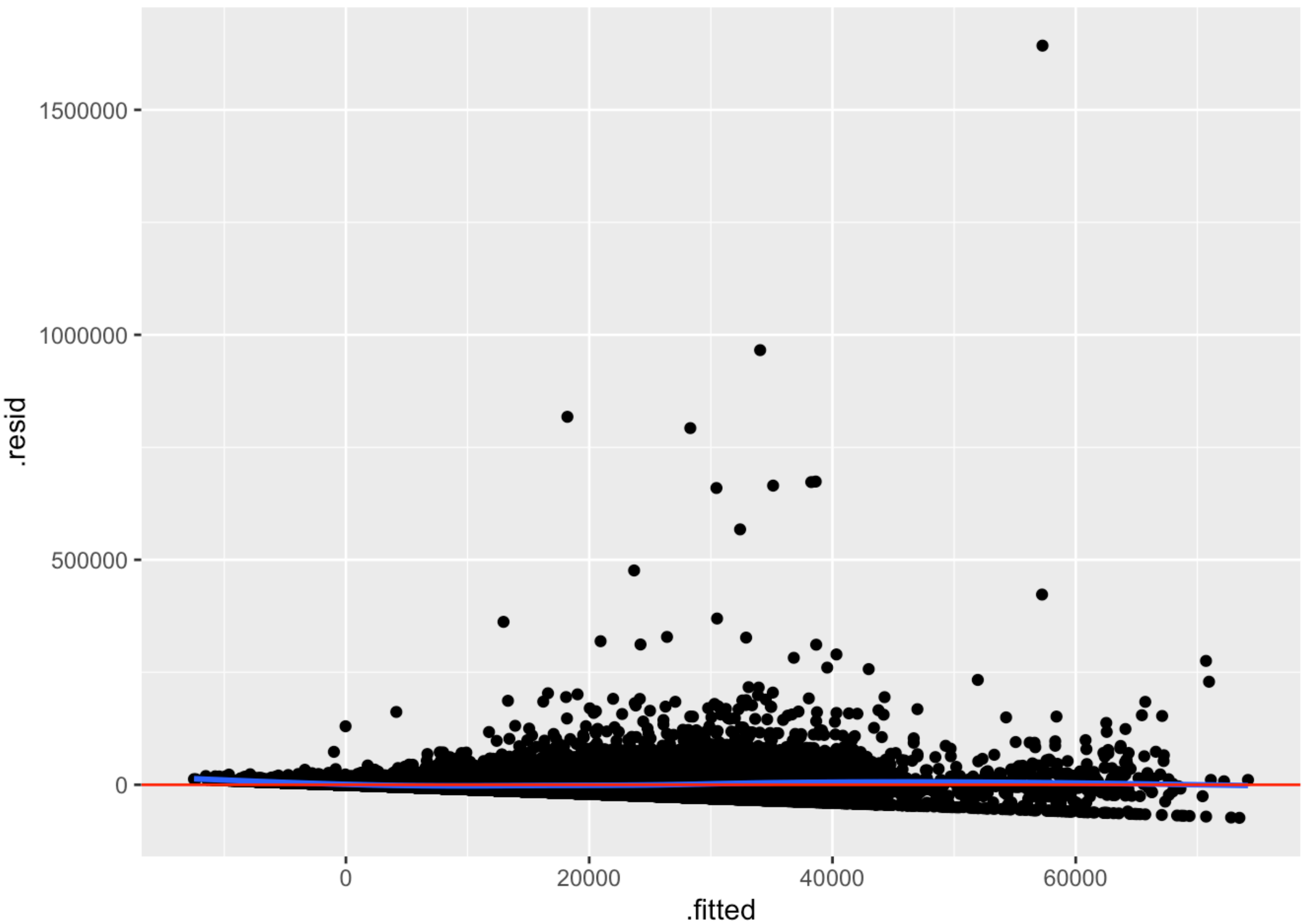
Use residual plots to evaluate whether some quantitative predictors might be better modeled with nonlinear relationships.

```
ggplot(mod1_output, aes(x = .fitted, y = .resid)) +
    geom_point() +
    geom_smooth() +
    geom_hline(yintercept = 0, color = "red")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

```
## Warning: Removed 23 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 23 rows containing missing values (geom_point).
```

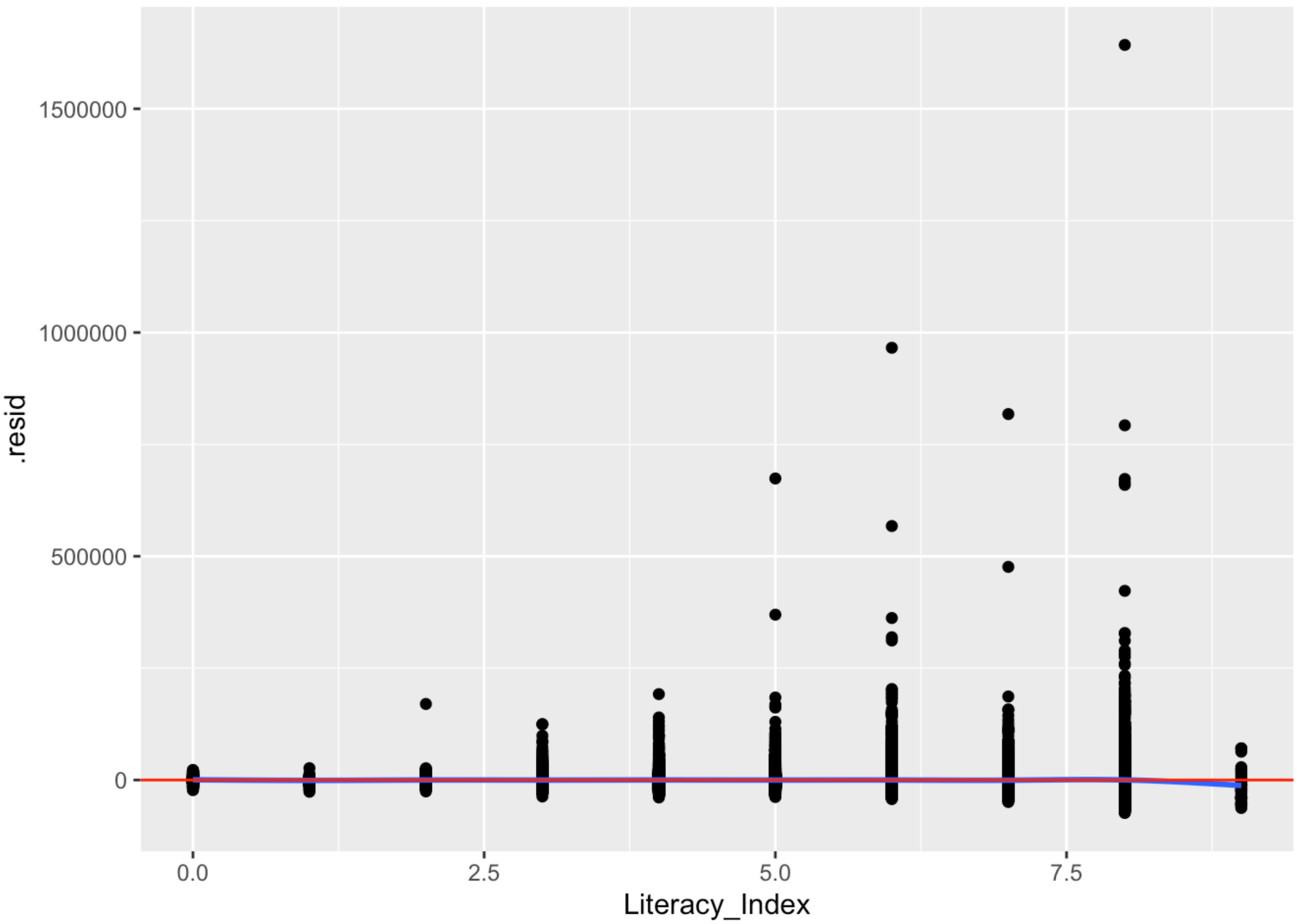



```
# Residuals vs. Literacy index
ggplot(mod1_output, aes(x = Literacy_Index, y = .resid)) +
    geom_point() +
    geom_smooth() +
    geom_hline(yintercept = 0, color = "red")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

```
## Warning: Removed 23 rows containing non-finite values (stat_smooth).

## Warning: Removed 23 rows containing missing values (geom_point).
```

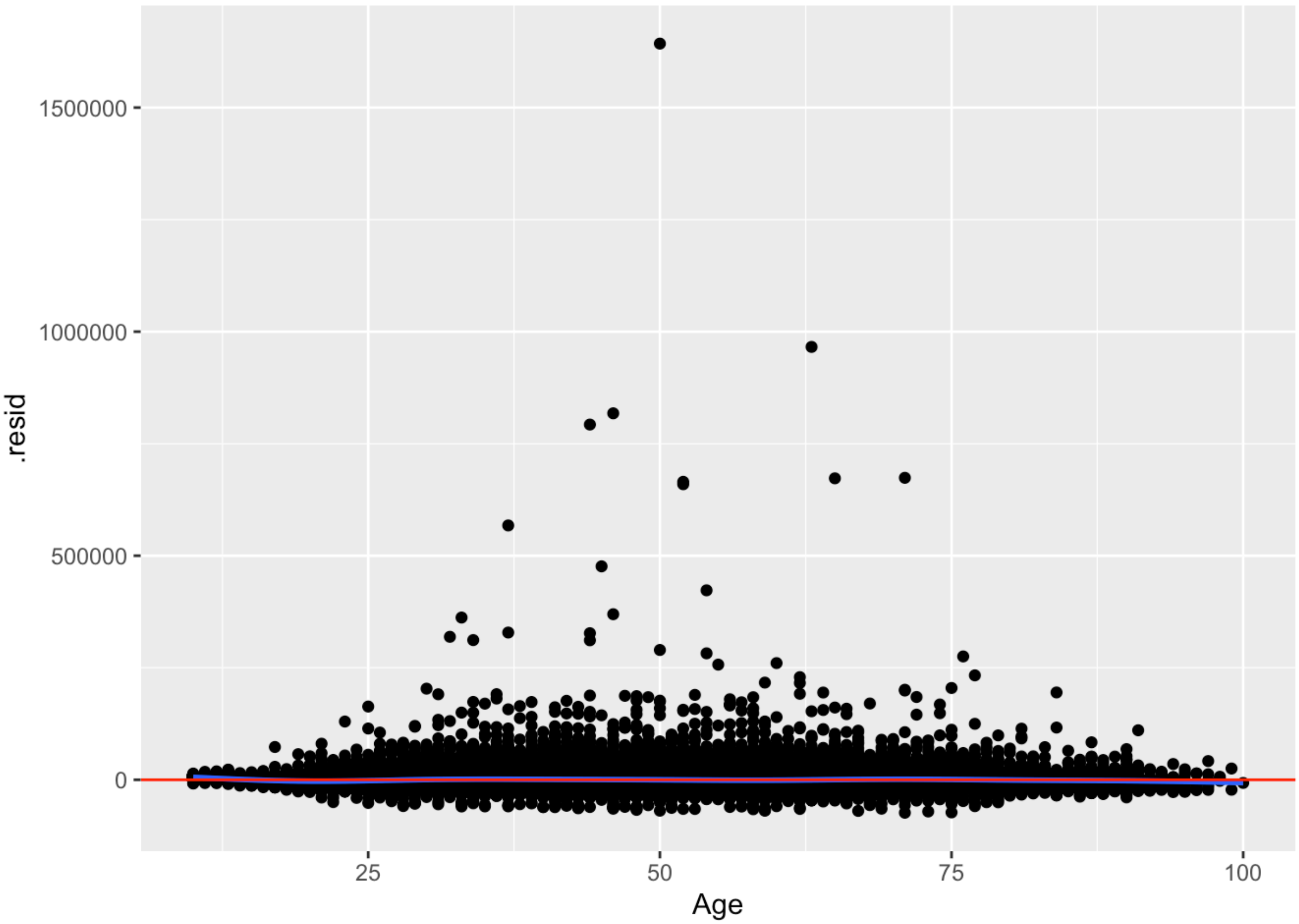



```
# Residuals vs. Age
ggplot(mod1_output, aes(x = Age, y = .resid)) +
    geom_point() +
    geom_smooth() +
    geom_hline(yintercept = 0, color = "red")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

```
## Warning: Removed 23 rows containing non-finite values (stat_smooth).

## Warning: Removed 23 rows containing missing values (geom_point).
```




**PUT ANY RELEVANT TEXT/RESPONSES/INTERPRETATIONS HERE** Considering the data used only has 2 quantitative predictors, our plots show the residuals vs. the Literacy_Index and Age predictors. The Age distributions look okay, but the Literacy index may indicate the need for a log transform because the distribution of the residuals towards the right seems to hit at heteroskedasticity. We should log transform the outcome Individual Income to see how that affects all our investigation, not only the plot It is worth noticing that even when Literacy is numerical, it does have a delimited set of values, which is depicted in the plot as the vertical columns of residuals. Even when no transformation seems necessary, the warning sign mentions the use of the gam method to plot it which is a good sign that we might want to try non-linear investigations. \

```
Data2019 <- data2019 %>% mutate(log_income = log(Income_individual+1))
```

```
## Warning in log(Income_individual + 1): NaNs produced
```

It looks like NAs were still produced. I'd recommend taking a look at the log_income variable to see if anything is amiss.

```
data_new <- Data2019                                        # Duplicate data
data_new[is.na(data_new) | data_new == "Inf"] <- NA # Replace NaN & Inf with NA
data_new[is.na(data_new) | data_new == "NaN"] <- NA
```

```
mod1_log <- lm(log_income ~ Literacy_Index+Sex+Birth_Location+CAT_OCUP+Civil_State+Hi
ghest_level+Cellphone_use+Internet_use+ Computer_use+Age,data = data_new, na.action =
na.omit)
summary(mod1_log)
```

```
##
## Call:
## lm(formula = log_income ~ Literacy_Index + Sex + Birth_Location +
##     CAT_OCUP + Civil_State + Highest_level + Cellphone_use +
##     Internet_use + Computer_use + Age, data = data_new, na.action = na.omit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.2478  -1.1983  -0.0168   1.6433  10.1126
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -0.102346   0.180035  -0.568 0.569713
## Literacy_Index  -0.066566   0.030651  -2.172 0.029881 *
## Sex2            -0.123641   0.026935  -4.590 4.44e-06 ***
## Birth_Location2  0.505625   0.045410  11.135  < 2e-16 ***
## Birth_Location3  0.237649   0.039055   6.085 1.17e-09 ***
## Birth_Location4 -0.176912   0.081143  -2.180 0.029243 *
## Birth_Location5 -0.125105   0.131129  -0.954 0.340060
## Birth_Location9  0.636839   1.232836   0.517 0.605463
## CAT_OCUP1        4.657846   0.103331  45.077  < 2e-16 ***
## CAT_OCUP2        4.270733   0.045943  92.958  < 2e-16 ***
## CAT_OCUP3        5.091827   0.030913 164.717  < 2e-16 ***
## CAT_OCUP4       -0.582917   0.218782  -2.664 0.007716 **
## CAT_OCUP9       -4.022862   2.755771  -1.460 0.144353
## Civil_State2    -0.628923   0.041561 -15.133  < 2e-16 ***
## Civil_State3    -0.241102   0.059882  -4.026 5.68e-05 ***
## Civil_State4     0.320384   0.071176   4.501 6.77e-06 ***
## Civil_State5    -0.798903   0.039320 -20.318  < 2e-16 ***
## Civil_State9     2.095173   4.088108   0.513 0.608301
## Highest_level1   0.124616   0.459121   0.271 0.786067
## Highest_level2   0.316609   0.200285   1.581 0.113932
## Highest_level3   0.943204   0.235046   4.013 6.01e-05 ***
## Highest_level4   0.578385   0.236608   2.444 0.014510 *
## Highest_level5   1.217729   0.262021   4.647 3.37e-06 ***
## Highest_level6   1.269573   0.287083   4.422 9.79e-06 ***
## Highest_level7   1.507984   0.284451   5.301 1.15e-07 ***
## Highest_level8   1.949667   0.335503   5.811 6.24e-09 ***
## Highest_level9   2.941281   0.251068  11.715  < 2e-16 ***
## Highest_level99  0.139335   2.766755   0.050 0.959835
## Cellphone_use2  -0.184407   0.049750  -3.707 0.000210 ***
```

```
## Cellphone_use9   2.465357   1.721898   1.432 0.152217    
## Internet_use2    0.169168   0.044558   3.797 0.000147 ***
## Internet_use9   -2.261993   1.144391  -1.977 0.048094 *  
## Computer_use2    0.123892   0.030536   4.057 4.97e-05 ***
## Computer_use9   -0.206686   0.627158  -0.330 0.741734    
## Age              0.112973   0.001027 109.967  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.755 on 45852 degrees of freedom
##   (4115 observations deleted due to missingness)
## Multiple R-squared:  0.6396, Adjusted R-squared:  0.6393 
## F-statistic:  2393 on 34 and 45852 DF,  p-value: < 2.2e-16
```

```
mod1_log_output <- broom::augment(mod1_log, newdata = data_new)
head(mod1_log_output)
```

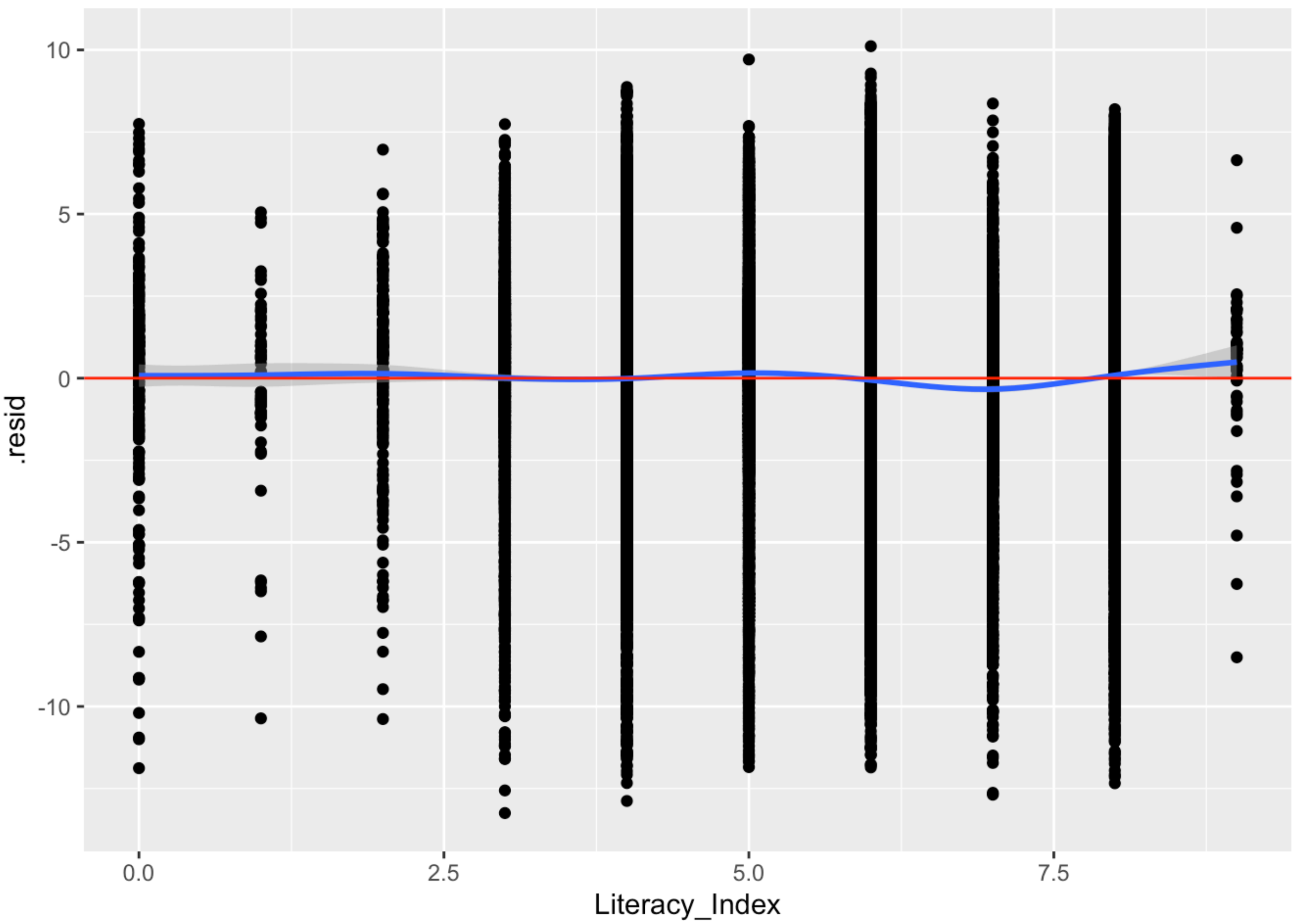
```
## # A tibble: 6 x 35
##   Ind_Interview REGION MAS_500 AGLOMERADO Relative_Rel Sex     Age Civil_State
##           <dbl> <fct>  <chr>        <dbl> <fct>        <fct> <dbl> <fct>      
## 1             1 43     S                2 1            1        44 3          
## 2             1 43     S                2 1            1        59 2          
## 3             1 43     S                2 2            2        62 2          
## 4             1 43     S                2 3            1        26 5          
## 5             1 43     S                2 3            1        23 5          
## 6             1 43     S                2 1            2        26 5          
## # … with 27 more variables: Type_of_school <fct>, Highest_level <fct>,
## #   finished? <fct>, last_yr_approved <chr>, Birth_Location <fct>,
## #   Location_5y <dbl>, CAT_OCUP <fct>, JOB_N <dbl>, Income_individual <dbl>,
## #   ITF <dbl>, person_id <dbl>, Internet_use <fct>, Computer_use <fct>,
## #   Cellphone_use <fct>, House_Type <fct>, Room_N <dbl>, Ownership <fct>,
## #   Self_Room <dbl>, Self_Room_Sleep <dbl>, Studio <fct>, Studio_N <dbl>,
## #   Computer_house <fct>, Internet_house <fct>, Literacy_Index <dbl>,
## #   log_income <dbl>, .fitted <dbl>, .resid <dbl>
```

```
# Residuals vs. Literacy index
ggplot(mod1_log_output, aes(x = Literacy_Index, y = .resid)) +
    geom_point() +
    geom_smooth() +
    geom_hline(yintercept = 0, color = "red")
```

```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```

```
## Warning: Removed 4115 rows containing non-finite values (stat_smooth).
```

```
## Warning: Removed 4115 rows containing missing values (geom_point).
```




The log transformation helped to fix the non-constant distribution of residuals but there is a trade-off on interpretability. We are staying with natural scales in investigation #1.

Compare insights from variable importance analyses from the different methods (stepwise and LASSO, but not OLS). Are there variables for which the methods reach consensus? What insights are expected? Surprising?

- Note that if some (but not all) of the indicator terms for a categorical predictor are selected in the final models, the whole predictor should be treated as selected.

**LASSO**

```
# Create a boolean matrix (predictors x lambdas) of variable exclusion
bool_predictor_exclude <- lasso_mod$finalModel$beta==0

# Loop over each variable
var_imp <- sapply(seq_len(nrow(bool_predictor_exclude)), function(row) {
    # Extract coefficient path (sorted from highest to lowest lambda)
    this_coeff_path <- bool_predictor_exclude[row,]
    # Compute and return the # of lambdas until this variable is out forever
    ncol(bool_predictor_exclude)-which.min(this_coeff_path)+1
})

# Create a dataset of this information and sort
var_imp_data <- tibble(
    var_name = rownames(bool_predictor_exclude),
    var_imp = var_imp
)
var_imp_data %>% arrange(desc(var_imp))
```

```
## # A tibble: 34 x 2
##    var_name       var_imp
##    <chr>            <dbl>
##  1 Cellphone_use9      75
##  2 CAT_OCUP3           74
##  3 Age                 74
##  4 Civil_State5        73
##  5 Literacy_Index      70
##  6 Highest_level8      66
##  7 CAT_OCUP1           65
##  8 Sex2                64
##  9 Highest_level7      64
## 10 Computer_use2       62
## # … with 24 more rows
```

**PUT ANY RELEVANT TEXT/RESPONSES/INTERPRETATIONS HERE**

\ In this case, the more persistent variables for LASSO are the same selected in backward stepwise selection except for the most persistent predictor for this model, which seems to be the Cellphone_use9 variable, which is not very relevant considering that it is NA's for "Use of cellphone in the last three months". It is a bit surprising considering we always talk about how backward and forward subset selection may not arrive to the same "best" model because they don't compute all the possibilities but generally they do their job. Leaving aside the Cellphone_use9, from the analysis we can see that Lasso method does coincide that the most important predictors are CAT_OCUP3, which corresponds to the category employee/worker, Civil_State5, which corresponds to being single and the Literacy index. Additionally we see the lasso model picked up Highest_level8, which corresponds to having attended to Graduate School (Masters, MBA, PhD), in the same way subset selection did.

# Investigation 2: Accounting for nonlinearity

Update your stepwise selection model(s) and LASSO model to use natural splines for the quantitative predictors.

- You'll need to update the model formula from `y ~ .` to something like `y ~ cat_var1 + ns(quant_var1, df) + ....`
- It's recommended to use few knots (e.g., 2 knots = 3 degrees of freedom).
- Note that `ns(x,3)` replaces `x` with 3 transformations of `x`. Keep this in mind when setting `nvmax` in stepwise selection.

**Stepwise**

```
set.seed(253)

spline_back_step_mod2 <- train(
    Income_individual~ ns(Literacy_Index,3)+Sex+Birth_Location+CAT_OCUP+Civil_State+H
ighest_level+Cellphone_use+Internet_use+ Computer_use+ns(Age,3),
    data = data2019,
    method = "leapBackward",
    tuneGrid = data.frame(nvmax = 1:14), #nvmax=number or vars.
    trControl = trainControl(method = "cv", number = 10),
    metric = "MAE",
    na.action = na.omit
)
```

```
## Warning in leaps.setup(x, y, wt = weights, nbest = nbest, nvmax = nvmax, : 1
## linear dependencies found
```

```
## Reordering variables and trying again:
```

```
summary(spline_back_step_mod2)
```

```
## Subset selection object
## 38 Variables  (and intercept)
##                        Forced in Forced out
## ns(Literacy_Index, 3)1     FALSE      FALSE
## ns(Literacy_Index, 3)2     FALSE      FALSE
## ns(Literacy_Index, 3)3     FALSE      FALSE
## Sex2                       FALSE      FALSE
## Birth_Location2            FALSE      FALSE
## Birth_Location3            FALSE      FALSE
## Birth_Location4            FALSE      FALSE
## Birth_Location5            FALSE      FALSE
```

```
## Birth_Location9            FALSE      FALSE
## CAT_OCUP1                  FALSE      FALSE
## CAT_OCUP2                  FALSE      FALSE
## CAT_OCUP3                  FALSE      FALSE
## CAT_OCUP4                  FALSE      FALSE
## CAT_OCUP9                  FALSE      FALSE
## Civil_State2               FALSE      FALSE
## Civil_State3               FALSE      FALSE
## Civil_State4               FALSE      FALSE
## Civil_State5               FALSE      FALSE
## Civil_State9               FALSE      FALSE
## Highest_level1             FALSE      FALSE
## Highest_level2             FALSE      FALSE
## Highest_level3             FALSE      FALSE
## Highest_level4             FALSE      FALSE
## Highest_level5             FALSE      FALSE
## Highest_level6             FALSE      FALSE
## Highest_level7             FALSE      FALSE
## Highest_level8             FALSE      FALSE
## Highest_level9             FALSE      FALSE
## Highest_level99            FALSE      FALSE
## Cellphone_use2             FALSE      FALSE
## Cellphone_use9             FALSE      FALSE
## Internet_use2              FALSE      FALSE
## Internet_use9              FALSE      FALSE
## Computer_use2              FALSE      FALSE
## Computer_use9              FALSE      FALSE
## ns(Age, 3)1                FALSE      FALSE
## ns(Age, 3)2                FALSE      FALSE
## ns(Age, 3)3                FALSE      FALSE
## 1 subsets of each size up to 14
## Selection Algorithm: backward
##           ns(Literacy_Index, 3)1 ns(Literacy_Index, 3)2 ns(Literacy_Index, 3)3
## 1  ( 1 )  " "                    " "                    " "
## 2  ( 1 )  " "                    " "                    " "
## 3  ( 1 )  " "                    " "                    "*"
## 4  ( 1 )  " "                    " "                    "*"
## 5  ( 1 )  " "                    " "                    "*"
## 6  ( 1 )  " "                    " "                    "*"
## 7  ( 1 )  " "                    " "                    "*"
## 8  ( 1 )  " "                    " "                    "*"
## 9  ( 1 )  " "                    " "                    "*"
## 10  ( 1 ) " "                    " "                    "*"
## 11  ( 1 ) " "                    " "                    "*"
## 12  ( 1 ) " "                    " "                    "*"
## 13  ( 1 ) " "                    " "                    "*"
## 14  ( 1 ) " "                    " "                    "*"
```

```
##           Sex2 Birth_Location2 Birth_Location3 Birth_Location4 Birth_Location5
## 1  ( 1 )  " "  " "             " "             " "             " "            
## 2  ( 1 )  " "  " "             " "             " "             " "            
## 3  ( 1 )  " "  " "             " "             " "             " "            
## 4  ( 1 )  " "  " "             " "             " "             " "            
## 5  ( 1 )  "*"  " "             " "             " "             " "            
## 6  ( 1 )  "*"  " "             " "             " "             " "            
## 7  ( 1 )  "*"  " "             " "             " "             " "            
## 8  ( 1 )  "*"  " "             " "             " "             " "            
## 9  ( 1 )  "*"  " "             " "             " "             " "            
## 10  ( 1 ) "*"  " "             " "             " "             " "            
## 11  ( 1 ) "*"  " "             " "             " "             " "            
## 12  ( 1 ) "*"  " "             " "             " "             " "            
## 13  ( 1 ) "*"  " "             " "             " "             " "            
## 14  ( 1 ) "*"  " "             "*"             " "             " "            
##           Birth_Location9 CAT_OCUP1 CAT_OCUP2 CAT_OCUP3 CAT_OCUP4 CAT_OCUP9
## 1  ( 1 )  " "             " "       " "       "*"       " "       " "      
## 2  ( 1 )  " "             " "       " "       "*"       " "       " "      
## 3  ( 1 )  " "             " "       " "       "*"       " "       " "      
## 4  ( 1 )  " "             "*"       " "       "*"       " "       " "      
## 5  ( 1 )  " "             "*"       " "       "*"       " "       " "      
## 6  ( 1 )  " "             "*"       " "       "*"       " "       " "      
## 7  ( 1 )  " "             "*"       " "       "*"       " "       " "      
## 8  ( 1 )  " "             "*"       " "       "*"       " "       " "      
## 9  ( 1 )  " "             "*"       "*"       "*"       " "       " "      
## 10  ( 1 ) " "             "*"       "*"       "*"       " "       " "      
## 11  ( 1 ) " "             "*"       "*"       "*"       " "       " "      
## 12  ( 1 ) " "             "*"       "*"       "*"       " "       " "      
## 13  ( 1 ) " "             "*"       "*"       "*"       " "       " "      
## 14  ( 1 ) " "             "*"       "*"       "*"       " "       " "      
##           Civil_State2 Civil_State3 Civil_State4 Civil_State5 Civil_State9
## 1  ( 1 )  " "          " "          " "          " "          " "         
## 2  ( 1 )  " "          " "          " "          " "          " "         
## 3  ( 1 )  " "          " "          " "          " "          " "         
## 4  ( 1 )  " "          " "          " "          " "          " "         
## 5  ( 1 )  " "          " "          " "          " "          " "         
## 6  ( 1 )  " "          " "          " "          " "          " "         
## 7  ( 1 )  " "          " "          " "          " "          " "         
## 8  ( 1 )  " "          " "          " "          " "          " "         
## 9  ( 1 )  " "          " "          " "          " "          " "         
## 10  ( 1 ) " "          " "          " "          " "          " "         
## 11  ( 1 ) " "          " "          " "          "*"          " "         
## 12  ( 1 ) " "          " "          " "          "*"          " "         
## 13  ( 1 ) " "          " "          " "          "*"          " "         
## 14  ( 1 ) " "          " "          " "          "*"          " "         
##           Highest_level1 Highest_level2 Highest_level3 Highest_level4
## 1  ( 1 )  " "            " "            " "            " "           
```

```
## 2  ( 1 )  " "            " "            " "            " "           
## 3  ( 1 )  " "            " "            " "            " "           
## 4  ( 1 )  " "            " "            " "            " "           
## 5  ( 1 )  " "            " "            " "            " "           
## 6  ( 1 )  " "            " "            " "            " "           
## 7  ( 1 )  " "            " "            " "            " "           
## 8  ( 1 )  " "            " "            " "            " "           
## 9  ( 1 )  " "            " "            " "            " "           
## 10  ( 1 ) " "            " "            " "            " "           
## 11  ( 1 ) " "            " "            " "            " "           
## 12  ( 1 ) " "            " "            " "            " "           
## 13  ( 1 ) " "            " "            " "            " "           
## 14  ( 1 ) " "            " "            " "            " "           
##           Highest_level5 Highest_level6 Highest_level7 Highest_level8
## 1  ( 1 )  " "            " "            " "            " "           
## 2  ( 1 )  " "            " "            " "            " "           
## 3  ( 1 )  " "            " "            " "            " "           
## 4  ( 1 )  " "            " "            " "            " "           
## 5  ( 1 )  " "            " "            " "            " "           
## 6  ( 1 )  " "            " "            " "            "*"           
## 7  ( 1 )  " "            " "            " "            "*"           
## 8  ( 1 )  " "            " "            " "            "*"           
## 9  ( 1 )  " "            " "            " "            "*"           
## 10  ( 1 ) " "            " "            " "            "*"           
## 11  ( 1 ) " "            " "            " "            "*"           
## 12  ( 1 ) " "            "*"            " "            "*"           
## 13  ( 1 ) " "            "*"            " "            "*"           
## 14  ( 1 ) " "            "*"            " "            "*"           
##           Highest_level9 Highest_level99 Cellphone_use2 Cellphone_use9
## 1  ( 1 )  " "            " "              " "            " "          
## 2  ( 1 )  " "            " "              " "            " "          
## 3  ( 1 )  " "            " "              " "            " "          
## 4  ( 1 )  " "            " "              " "            " "          
## 5  ( 1 )  " "            " "              " "            " "          
## 6  ( 1 )  " "            " "              " "            " "          
## 7  ( 1 )  " "            " "              " "            " "          
## 8  ( 1 )  " "            " "              " "            " "          
## 9  ( 1 )  " "            " "              " "            " "          
## 10  ( 1 ) " "            " "              " "            " "          
## 11  ( 1 ) " "            " "              " "            " "          
## 12  ( 1 ) " "            " "              " "            " "          
## 13  ( 1 ) " "            " "              " "            " "          
## 14  ( 1 ) " "            " "              " "            " "          
##           Internet_use2 Internet_use9 Computer_use2 Computer_use9 ns(Age, 3)1
## 1  ( 1 )  " "           " "           " "           " "           " "        
## 2  ( 1 )  " "           " "           " "           " "           "*"        
## 3  ( 1 )  " "           " "           " "           " "           "*"        
```

```
## 4  ( 1 )  " "           " "           " "           " "           "*"
## 5  ( 1 )  " "           " "           " "           " "           "*"
## 6  ( 1 )  " "           " "           " "           " "           "*"
## 7  ( 1 )  " "           " "           " "           " "           "*"
## 8  ( 1 )  " "           " "           " "           " "           "*"
## 9  ( 1 )  " "           " "           " "           " "           "*"
## 10  ( 1 ) " "           " "           "*"           " "           "*"
## 11  ( 1 ) " "           " "           "*"           " "           "*"
## 12  ( 1 ) " "           " "           "*"           " "           "*"
## 13  ( 1 ) "*"           " "           "*"           " "           "*"
## 14  ( 1 ) "*"           " "           "*"           " "           "*"
##           ns(Age, 3)2 ns(Age, 3)3
## 1  ( 1 )  " "         " "
## 2  ( 1 )  " "         " "
## 3  ( 1 )  " "         " "
## 4  ( 1 )  " "         " "
## 5  ( 1 )  " "         " "
## 6  ( 1 )  " "         " "
## 7  ( 1 )  "*"         " "
## 8  ( 1 )  "*"         "*"
## 9  ( 1 )  "*"         "*"
## 10  ( 1 ) "*"         "*"
## 11  ( 1 ) "*"         "*"
## 12  ( 1 ) "*"         "*"
## 13  ( 1 ) "*"         "*"
## 14  ( 1 ) "*"         "*"
```

```
# Look at accuracy/error metrics for the different subset sizes
# If you want to sort the table of results, use arrange() from dplyr
spline_back_step_mod2$results %>% arrange(MAE)
```

```
##    nvmax     RMSE   Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 1     14 21165.22 0.24107381 10809.47 3724.964 0.04384898 131.0383
## 2     13 21196.66 0.23884205 10821.67 3733.309 0.04507788 155.8003
## 3     12 21205.57 0.23817622 10824.09 3732.406 0.04490156 153.6167
## 4     11 21222.37 0.23692563 10834.52 3724.088 0.04425554 151.7327
## 5     10 21300.01 0.23118036 10894.72 3713.738 0.04292123 148.9044
## 6      9 21354.86 0.22710708 10903.06 3710.179 0.04174570 137.6669
## 7      8 21438.07 0.22100174 11075.57 3718.096 0.04197517 178.0557
## 8      7 21577.65 0.21086429 11190.32 3779.432 0.04513555 200.0228
## 9      6 21670.44 0.20389066 11294.99 3762.111 0.04313215 171.6772
## 10     4 21910.07 0.18577803 11434.75 3783.104 0.04575279 495.5613
## 11     5 21822.13 0.19241715 11450.29 3779.249 0.04741749 479.5312
## 12     3 22031.99 0.17647248 11639.67 3772.488 0.04423440 450.7492
## 13     2 22461.53 0.14344219 11996.36 3770.294 0.04006222 379.2329
## 14     1 23219.11 0.08284268 13193.21 3657.685 0.02120588 180.4436
```

```
spline_back_step_mod2$bestTune
```

```
##    nvmax
## 14    14
```

```
# Obtain the coefficients for the best model
coef(spline_back_step_mod2$finalModel, id = spline_back_step_mod2$bestTune$nvmax)
```

```
##            (Intercept) ns(Literacy_Index, 3)3                   Sex2
##              10286.720              14013.732              -4308.312
##        Birth_Location3              CAT_OCUP1              CAT_OCUP2
##               2030.464              17596.431               6182.729
##              CAT_OCUP3           Civil_State5         Highest_level6
##              13801.351              -4526.323              -3234.324
##         Highest_level8          Internet_use2          Computer_use2
##              24420.379              -2376.977              -3280.910
##            ns(Age, 3)1            ns(Age, 3)2            ns(Age, 3)3
##              13810.487              19185.984              21795.493
```

### LASSO

```
set.seed(253)
spline_lasso_mod2 <- train(
    Income_individual ~ ns(Literacy_Index,3)+Sex+Birth_Location+CAT_OCUP+Civil_State+
Highest_level+Cellphone_use+Internet_use+ Computer_use+ns(Age,3),
    data = data2019,
    method = "glmnet",
    trControl = trainControl(method = "cv", number = 10, selectionFunction = "oneSE")
,
    tuneGrid = data.frame(alpha = 1, lambda = seq(0,1000, length.out = 100)),#Ask Les
lie about length.out
    metric = "MAE",
    na.action = na.omit
)
```

```
# Identify which tuning parameter (lambda) is "best"
spline_lasso_mod2$bestTune
```

```
##    alpha   lambda
## 54     1 535.3535
```

```
coef(spline_lasso_mod2$finalModel, lasso_mod$bestTune$lambda)
```

```
## 39 x 1 sparse Matrix of class "dgCMatrix"
##                                  1
## (Intercept)            11775.7161
## ns(Literacy_Index, 3)1     .
## ns(Literacy_Index, 3)2     .
## ns(Literacy_Index, 3)3  9487.5276
## Sex2                   -3319.9355
## Birth_Location2            .
## Birth_Location3          674.0065
## Birth_Location4            .
## Birth_Location5            .
## Birth_Location9            .
## CAT_OCUP1              11278.0200
## CAT_OCUP2               2017.8451
## CAT_OCUP3              10699.1916
## CAT_OCUP4                  .
## CAT_OCUP9                  .
## Civil_State2               .
## Civil_State3               .
## Civil_State4            1951.8274
## Civil_State5           -5057.9354
## Civil_State9               .
## Highest_level1             .
## Highest_level2          -136.9092
## Highest_level3             .
## Highest_level4             .
## Highest_level5             .
## Highest_level6             .
## Highest_level7          2166.9123
## Highest_level8         21113.3501
## Highest_level9             .
## Highest_level99            .
## Cellphone_use2             .
## Cellphone_use9             .
## Internet_use2              .
## Internet_use9              .
## Computer_use2          -1847.6912
## Computer_use9              .
## ns(Age, 3)1            16152.1577
## ns(Age, 3)2             5921.7745
## ns(Age, 3)3             2078.5224
```

```r
# Create a boolean matrix (predictors x lambdas) of variable exclusion
bool_predictor_exclude <- spline_lasso_mod2$finalModel$beta==0

# Loop over each variable
var_imp <- sapply(seq_len(nrow(bool_predictor_exclude)), function(row) {
    # Extract coefficient path (sorted from highest to lowest lambda)
    this_coeff_path <- bool_predictor_exclude[row,]
    # Compute and return the # of lambdas until this variable is out forever
    ncol(bool_predictor_exclude)-which.min(this_coeff_path)+1
})

# Create a dataset of this information and sort
var_imp_data <- tibble(
    var_name = rownames(bool_predictor_exclude),
    var_imp = var_imp
)
var_imp_data %>% arrange(desc(var_imp))
```

```
## # A tibble: 38 x 2
##    var_name               var_imp
##    <chr>                    <dbl>
##  1 Cellphone_use9              74
##  2 CAT_OCUP3                   73
##  3 Civil_State5                73
##  4 ns(Age, 3)1                 71
##  5 ns(Literacy_Index, 3)3      70
##  6 Highest_level8              65
##  7 ns(Age, 3)2                 64
##  8 Sex2                        63
##  9 CAT_OCUP1                   63
## 10 Highest_level7              60
## # … with 28 more rows
```

Compare insights from variable importance analyses here and the corresponding results from Investigation 1. Now after having accounted for nonlinearity, have the most relevant predictors changed?

- Note that if some (but not all) of the spline terms are selected in the final models, the whole predictor should be treated as selected.

**PUT ANY RELEVANT TEXT/RESPONSES/INTERPRETATIONS HERE**

Using splines transformations for quantitative variables showed us that they remain among the most important predictors when we account for non-linearity in both methods. When one transformation is selected, it accounts for the importance of the whole variable. We can further the exploration using the GAMs \

Fit a GAM using LOESS terms using the set of variables deemed to be most relevant based on your investigations so far.

```
# Your code
set.seed(253)
gam_mod <- train(
    Income_individual ~ Literacy_Index+Sex+Birth_Location+CAT_OCUP+Civil_State+Highes
t_level+Cellphone_use
    +Internet_use+Computer_use+Age,
    data = data2019,
    method = "gamLoess",
    tuneGrid = data.frame(degree = 1, span = seq(0.1, 0.9, by = 0.1)),
    trControl = trainControl(method = "cv", number = 8, selectionFunction = "best"),
    metric = "MAE",
    na.action = na.omit
)
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.1, degree = 1)"]], z, w, span = 0.1, :
## eval 100
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.1, degree = 1)"]], z, w, span = 0.1, :
## upperlimit 99.445
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.1, degree = 1)"]], z, w, span = 0.1, :
## extrapolation not allowed with blending
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.2, degree = 1)"]], z, w, span = 0.2, :
## eval 100
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.2, degree = 1)"]], z, w, span = 0.2, :
## upperlimit 99.445
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.2, degree = 1)"]], z, w, span = 0.2, :
## extrapolation not allowed with blending
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.3, degree = 1)"]], z, w, span = 0.3, :
## eval 100
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.3, degree = 1)"]], z, w, span = 0.3, :
## upperlimit 99.445
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.3, degree = 1)"]], z, w, span = 0.3, :
## extrapolation not allowed with blending
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.4, degree = 1)"]], z, w, span = 0.4, :
## eval 100
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.4, degree = 1)"]], z, w, span = 0.4, :
## upperlimit 99.445
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.4, degree = 1)"]], z, w, span = 0.4, :
## extrapolation not allowed with blending
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.5, degree = 1)"]], z, w, span = 0.5, :
## eval 100
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.5, degree = 1)"]], z, w, span = 0.5, :
## upperlimit 99.445
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.5, degree = 1)"]], z, w, span = 0.5, :
## extrapolation not allowed with blending
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.6, degree = 1)"]], z, w, span = 0.6, :
## eval 100
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.6, degree = 1)"]], z, w, span = 0.6, :
## upperlimit 99.445
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.6, degree = 1)"]], z, w, span = 0.6, :
## extrapolation not allowed with blending
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.7, degree = 1)"]], z, w, span = 0.7, :
## eval 100
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.7, degree = 1)"]], z, w, span = 0.7, :
## upperlimit 99.445
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.7, degree = 1)"]], z, w, span = 0.7, :
## extrapolation not allowed with blending
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.8, degree = 1)"]], z, w, span = 0.8, :
## eval 100
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.8, degree = 1)"]], z, w, span = 0.8, :
## upperlimit 99.445
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.8, degree = 1)"]], z, w, span = 0.8, :
## extrapolation not allowed with blending
```

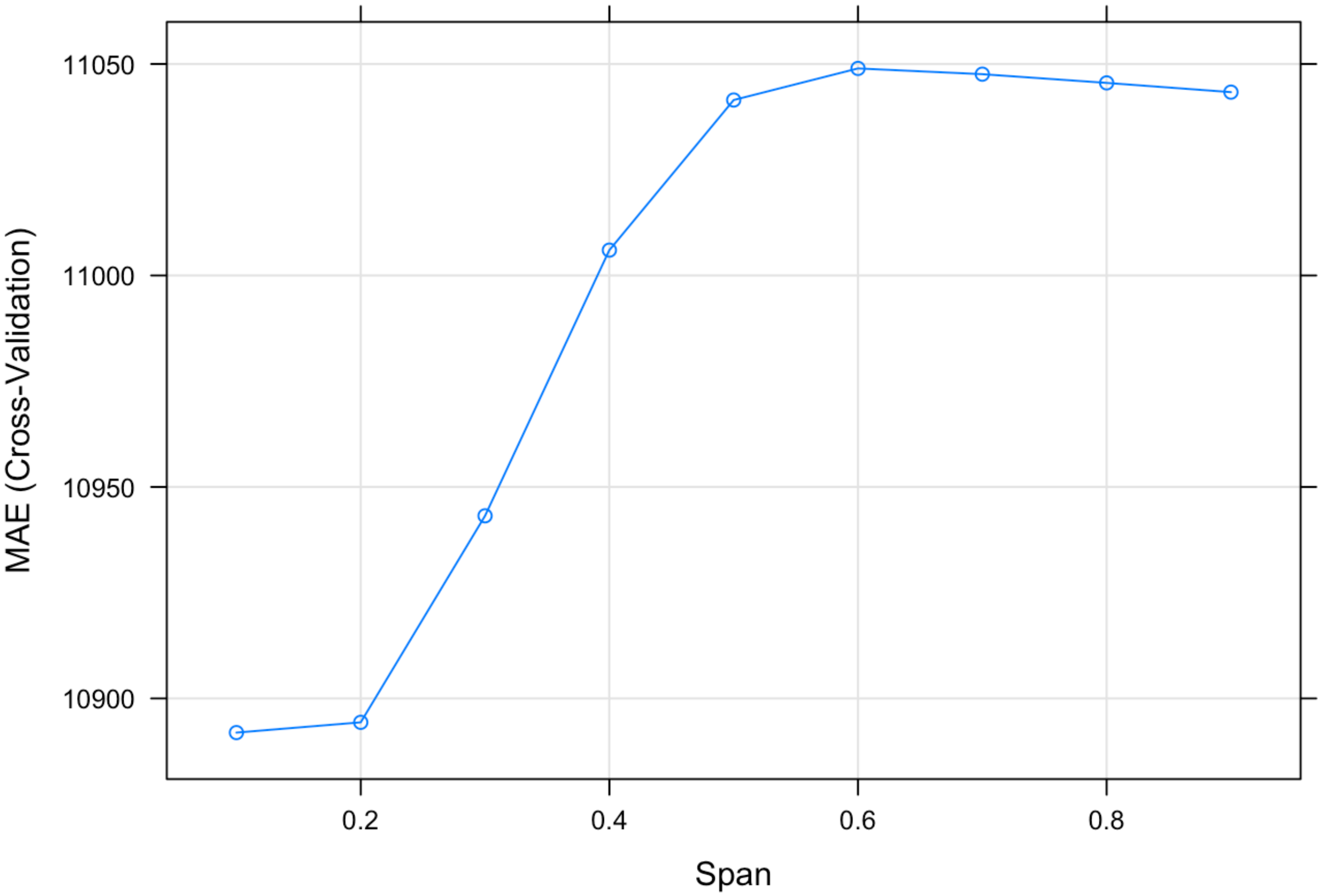
```
## Warning in gam.lo(data[["lo(Age, span = 0.9, degree = 1)"]], z, w, span = 0.9, :
## eval 100
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.9, degree = 1)"]], z, w, span = 0.9, :
## upperlimit 99.445
```

```
## Warning in gam.lo(data[["lo(Age, span = 0.9, degree = 1)"]], z, w, span = 0.9, :
## extrapolation not allowed with blending
```

Cellphone_use9 75
CAT_OCUP3 74
Age 74
Civil_State5 73
Literacy_Index 70
Highest_level8 66
CAT_OCUP1 65
Sex2

```
plot(gam_mod)
```



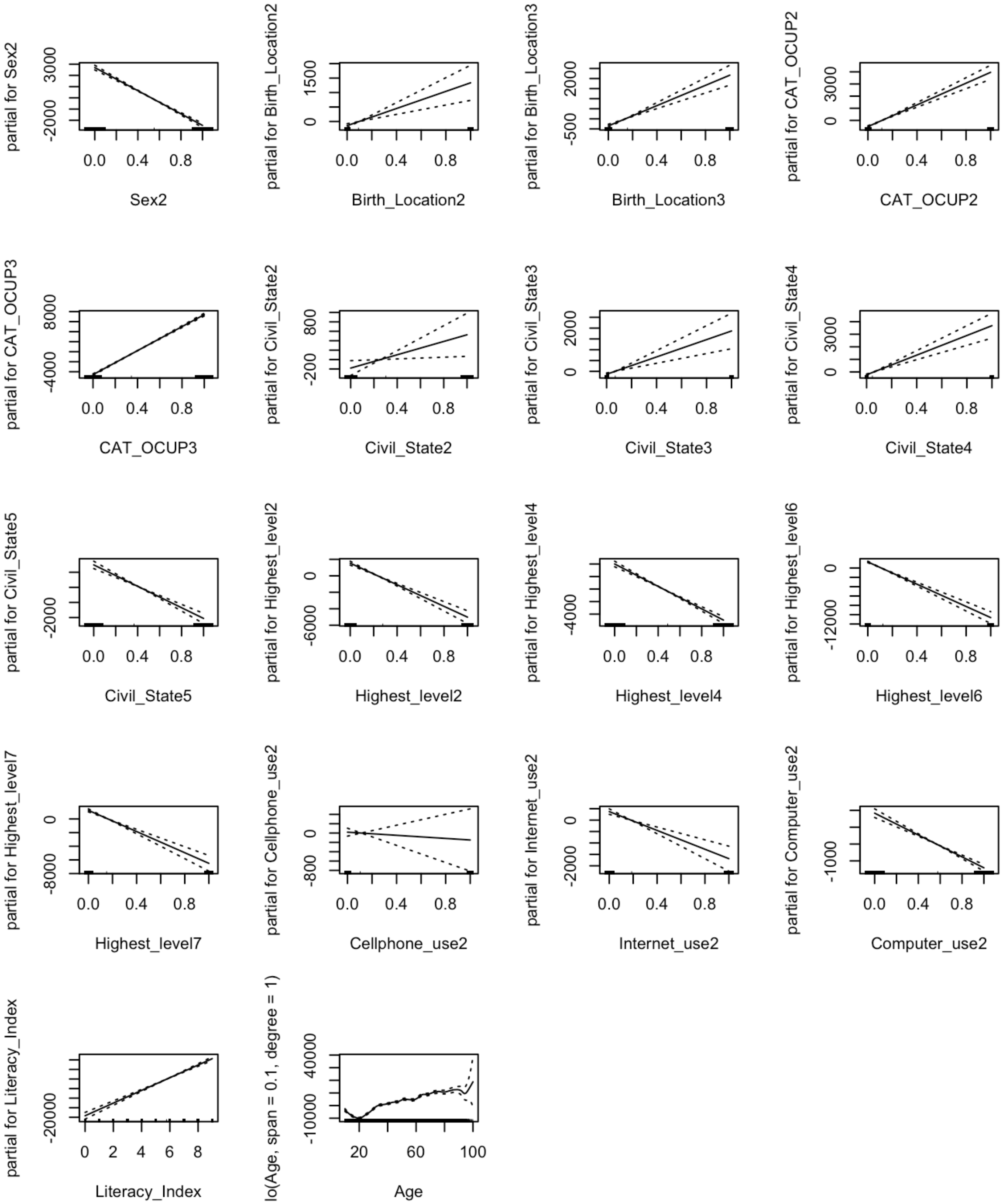

```
gam_mod$bestTune
```

```
##   span degree
## 1  0.1      1
```

```
gam_mod$results %>%
    filter(span==gam_mod$bestTune$span)
```

```
##   degree span     RMSE Rsquared      MAE   RMSESD RsquaredSD    MAESD
## 1      1  0.1 21320.16 0.234746 10891.92 2969.295 0.04393753 156.0739
```

```
par(mfrow = c(3,4)) # Sets up a grid of plots
plot(gam_mod$finalModel, se = TRUE) # Dashed lines are +/- 2 SEs
```


partial for Sex2
3000
-2000
0.0 0.4 0.8
Sex2
partial for Birth_Location2
1500
0
0.0 0.4 0.8
Birth_Location2
partial for Birth_Location3
2000
-500
0.0 0.4 0.8
Birth_Location3
partial for CAT_OCUP2
3000
0
0.0 0.4 0.8
CAT_OCUP2
partial for CAT_OCUP3
8000
-4000
0.0 0.4 0.8
CAT_OCUP3
partial for Civil_State2
800
-200
0.0 0.4 0.8
Civil_State2
partial for Civil_State3
2000
0
0.0 0.4 0.8
Civil_State3
partial for Civil_State4
3000
0
0.0 0.4 0.8
Civil_State4
partial for Civil_State5
-2000
0.0 0.4 0.8
Civil_State5
partial for Highest_level2
0
-6000
0.0 0.4 0.8
Highest_level2
partial for Highest_level4
-4000
0.0 0.4 0.8
Highest_level4
partial for Highest_level6
0
-12000
0.0 0.4 0.8
Highest_level6
partial for Highest_level7
0
-8000
0.0 0.4 0.8
Highest_level7
partial for Cellphone_use2
0
-800
0.0 0.4 0.8
Cellphone_use2
partial for Internet_use2
0
-2000
0.0 0.4 0.8
Internet_use2
partial for Computer_use2
-1000
0.0 0.4 0.8
Computer_use2
partial for Literacy_Index
-20000
0 2 4 6 8
Literacy_Index
lo(Age, span = 0.1, degree = 1)
40000
-10000
20 60 100
Age

**PUT ANY RELEVANT TEXT/RESPONSES/INTERPRETATIONS HERE**

- How does test performance of the GAM compare to other models you explored?

The MAE of the GAM is 10891.92. This means that on average this model is off in its percentage predictions about a person's income by about 10892 pesos.

- Do you gain any insights from the GAM output plots for each predictor?

Since most of the variables are categorical (indicator) variables, we observe linear plots. We can see that among individuals that are same in all other characteristics, those of Sex2 and Civil_State5 have lower average income than others. On the other hand, among individuals that are same in all other characteristics, those of CAT_OCUP 2&3 and Birth_Location 2&3 have higher average income than others. \

# Summarize investigations

Decide on an overall best model based on your investigations so far. To do this, make clear your analysis goals. Predictive accuracy? Interpretability? A combination of both?

Our analysis goal is to accurately predict the income of individuals based on different factors, which we believe will ultimately help us predict their digital access and consequent level of education disruption during COVID. Of course, we are hoping that these predicts will be interpretable as well. The overall best model so far for this is the Lasso, which had a MAE of 10777, the lowest so far.

\

Ok, but perhaps in considering the MAESD there aren't huge differences between the models. Simplicity and variable importance investigations might help distinguish preferred models in this case.

# Societal impact

Are there any harms that may come from your analyses and/or how the data were collected? What cautions do you want to keep in mind when communicating your work?

It is important that we communicate our analyses in a nuanced and non-exaggerated way, based on the relatively significant level of error we have faced so far. If not, this could misguide development programs working in education and digital access. This particular set of analyses, which focuses on income as an outcome of interest, has to effectively address intersectionality and the connection between different social dynamics, in order to ensure holistic development efforts, which do not discriminate against a certain group of people based on pre-existing conditions. If educational and digital access related programmes are not carried out effectively - based on solid evidence - then they could impact the learning, growth and consequently life of many people, causing more harm than good. Thus, it is important to exercise the utmost caution to ensure accuracy and mindfulness of analyses.