Lab 6

Elizabeth McHugh

11:59PM April 15, 2021

#Visualization with the package ggplot2

I highly recommend using the ggplot cheat sheet as a reference resource. You will see questions that say "Create the best-looking plot". Among other things you may choose to do, remember to label the axes using real English, provide a title and subtitle. You may want to pick a theme and color scheme that you like and keep that constant throughout this lab. The default is fine if you are running short of time.

Load up the `GSSvocab` dataset in package `carData` as `X` and drop all observations with missing measurements.

```
pacman::p_load(carData)

data(GSSvocab)

GSSvocab = na.omit(GSSvocab)
```

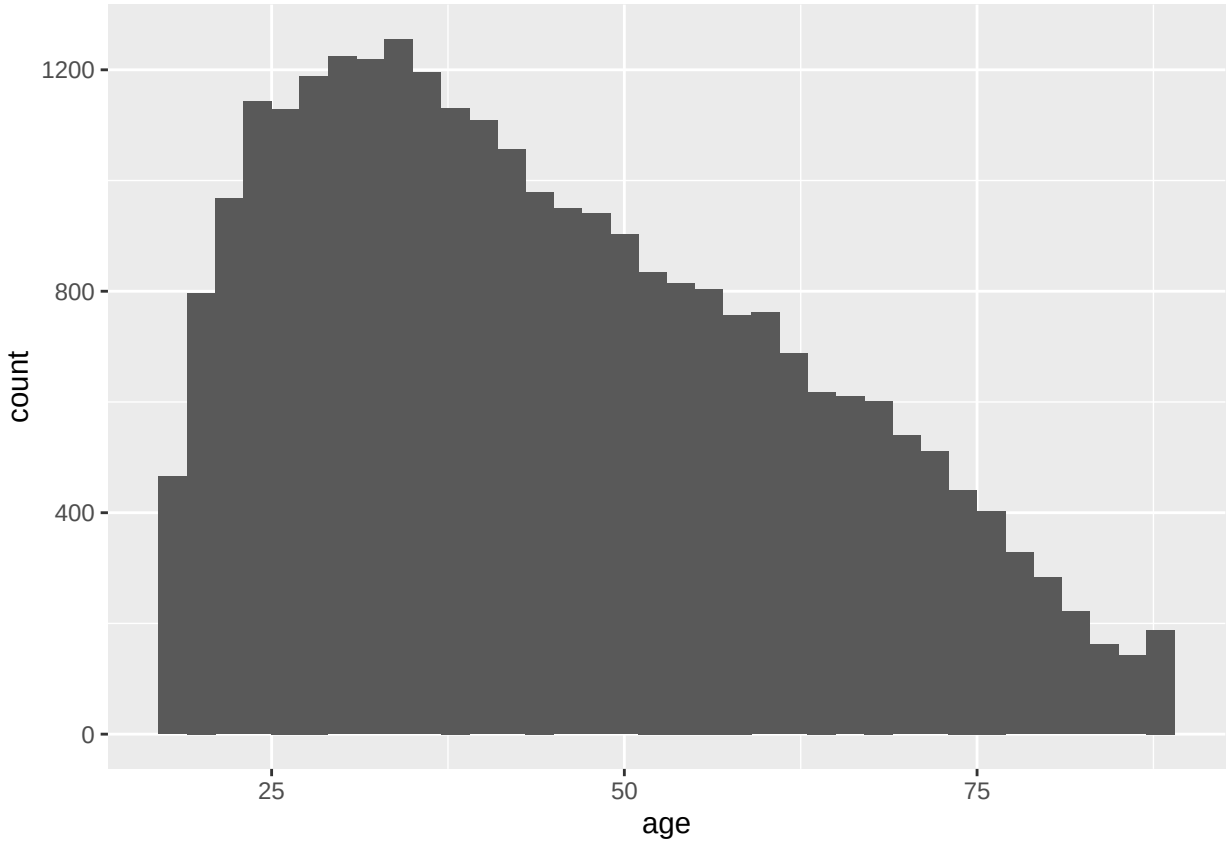
Briefly summarize the documentation on this dataset. What is the data type of each variable? What do you think is the response variable the collectors of this data had in mind?

#TO-DO Summarize the data set information in a paragraph!!!!! (Come back to do.)

Create two different plots and identify the best-looking plot you can to examine the `age` variable. Save the best looking plot as an appropriately-named PDF.

```
pacman::p_load(ggplot2)

ggplot(GSSvocab) +
  aes(x = age)  +
  geom_histogram(binwidth = 2)
```

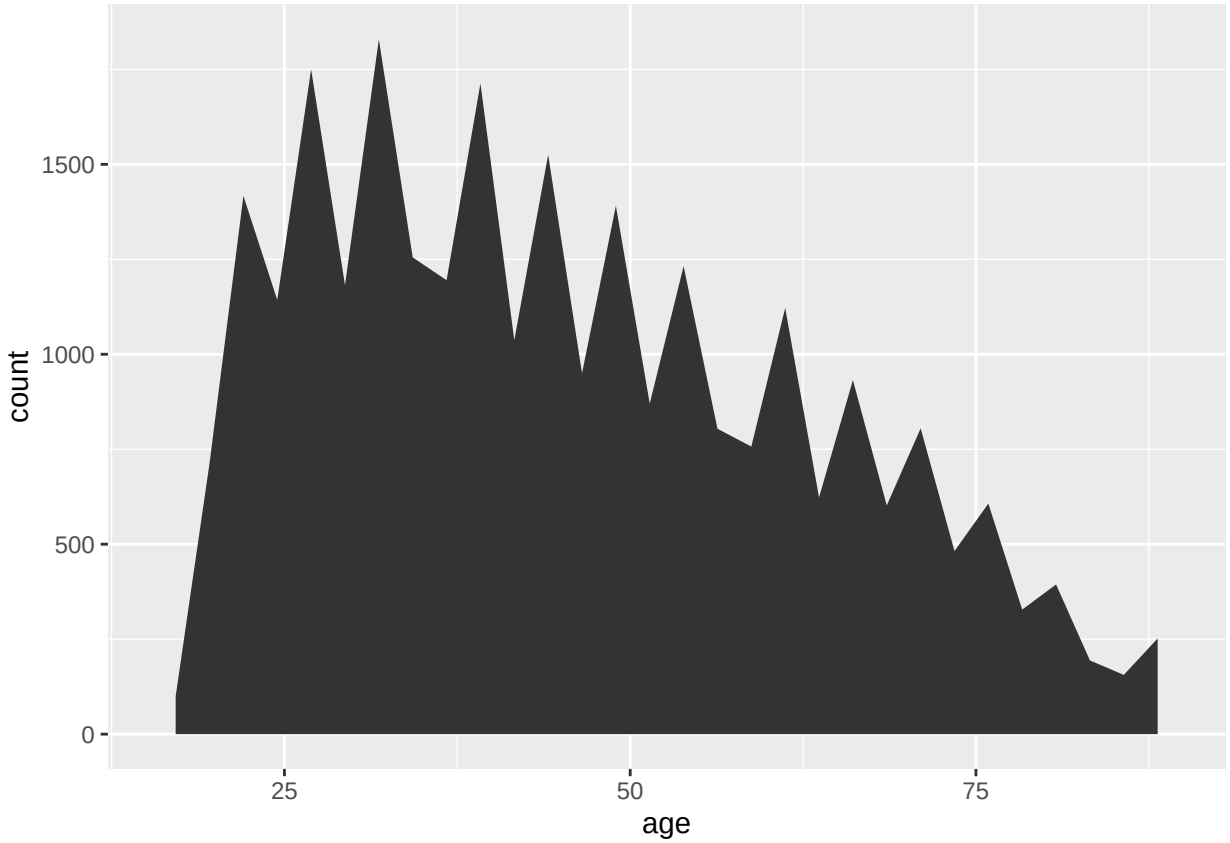



```
ggsave("vocabbyage.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

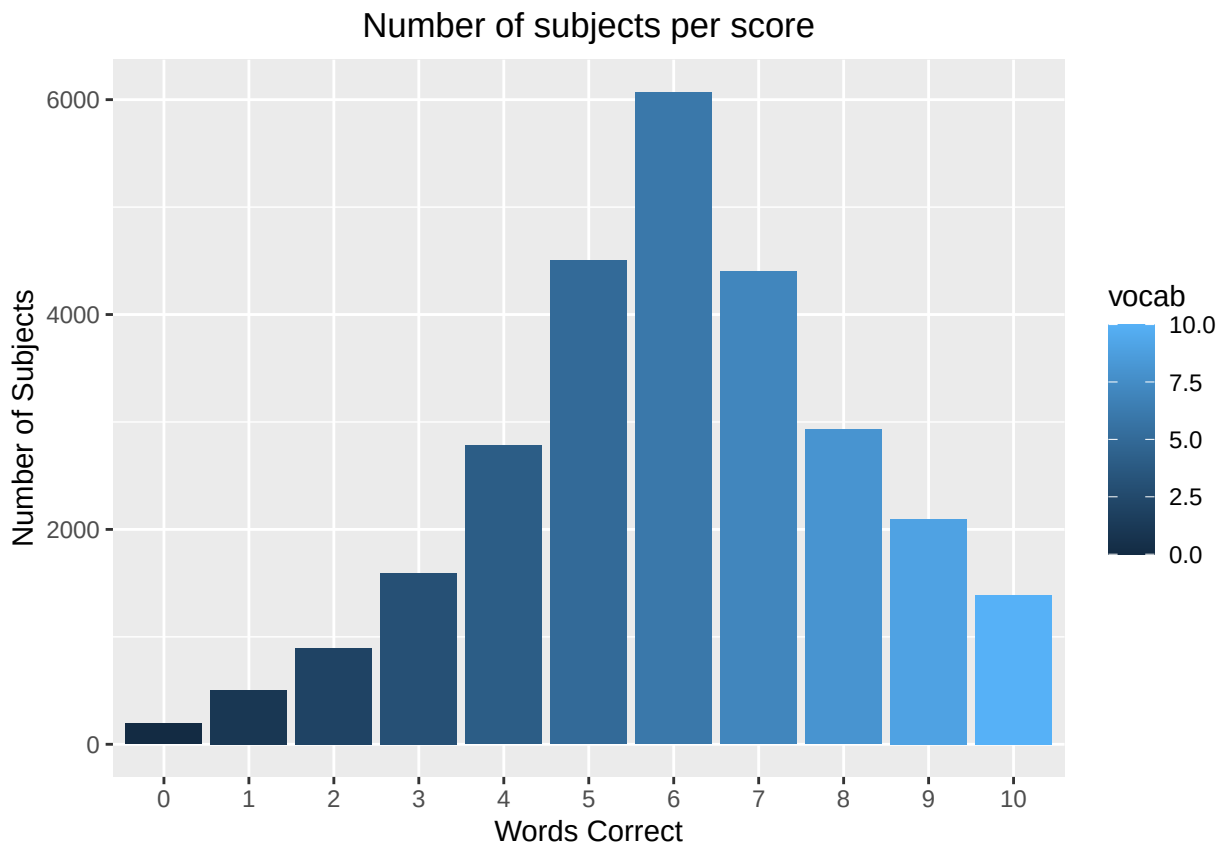
```
ggplot(GSSvocab) +
  aes(x = age) +
  geom_area(stat = "bin")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Create two different plots and identify the best looking plot you can to examine the `vocab` variable. Save the best looking plot as an appropriately-named PDF.

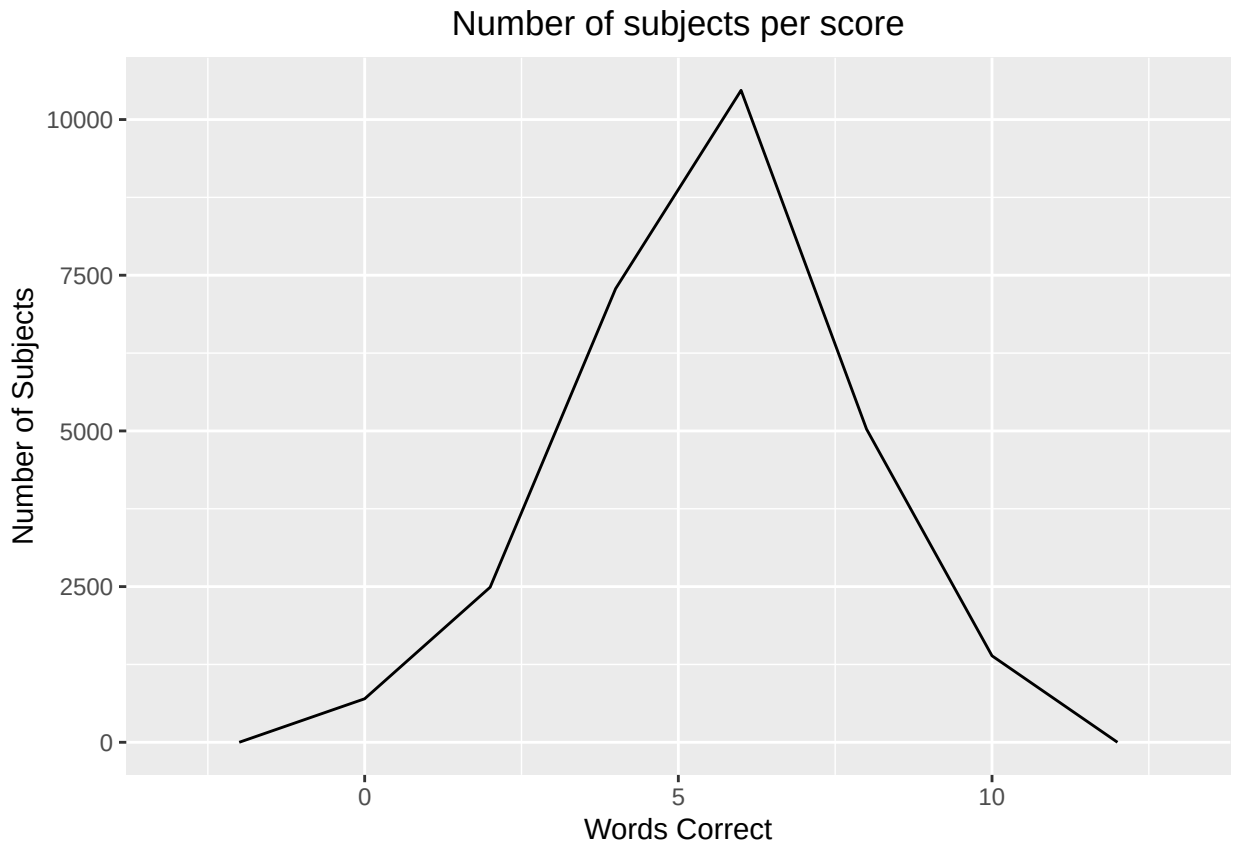
```
ggplot(GSSvocab) +
  aes(x = factor(vocab))+
  geom_bar(aes(fill = vocab)) +
   labs(title = "Number of subjects per score", x = "Words Correct", y = "Number of Subjects") +
  theme(plot.title = element_text(hjust = 0.5))
```




```
ggsave("vocab_plot.pdf")
```

```
## Saving 6.5 x 4.5 in image
```

```
ggplot(GSSvocab) +
  aes(x = vocab) +
  geom_freqpoly(binwidth = 2) +
  labs(title = "Number of subjects per score", x = "Words Correct", y = "Number of Subjects") +
  theme(plot.title = element_text(hjust = 0.5))
```



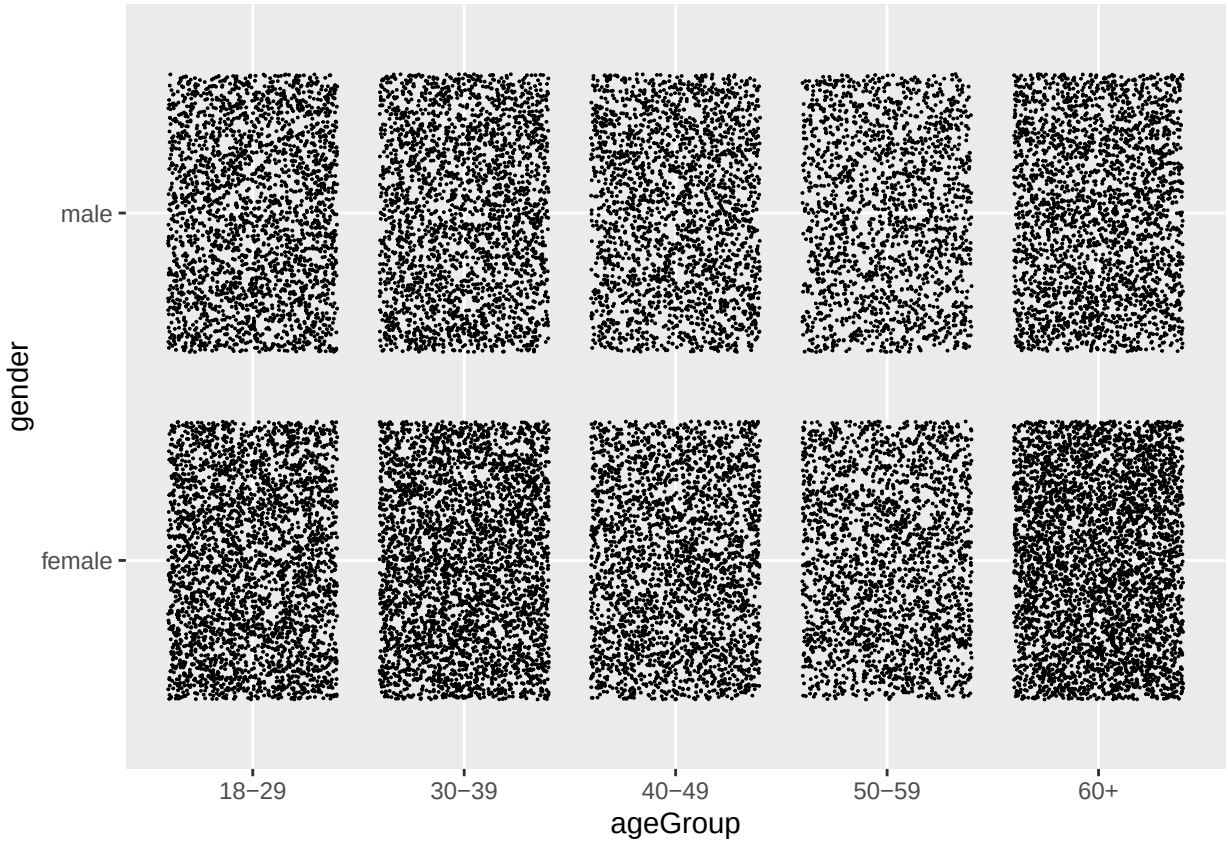

```
?GSSvocab
```

```
## starting httpd help server ...
```

```
##  done
```

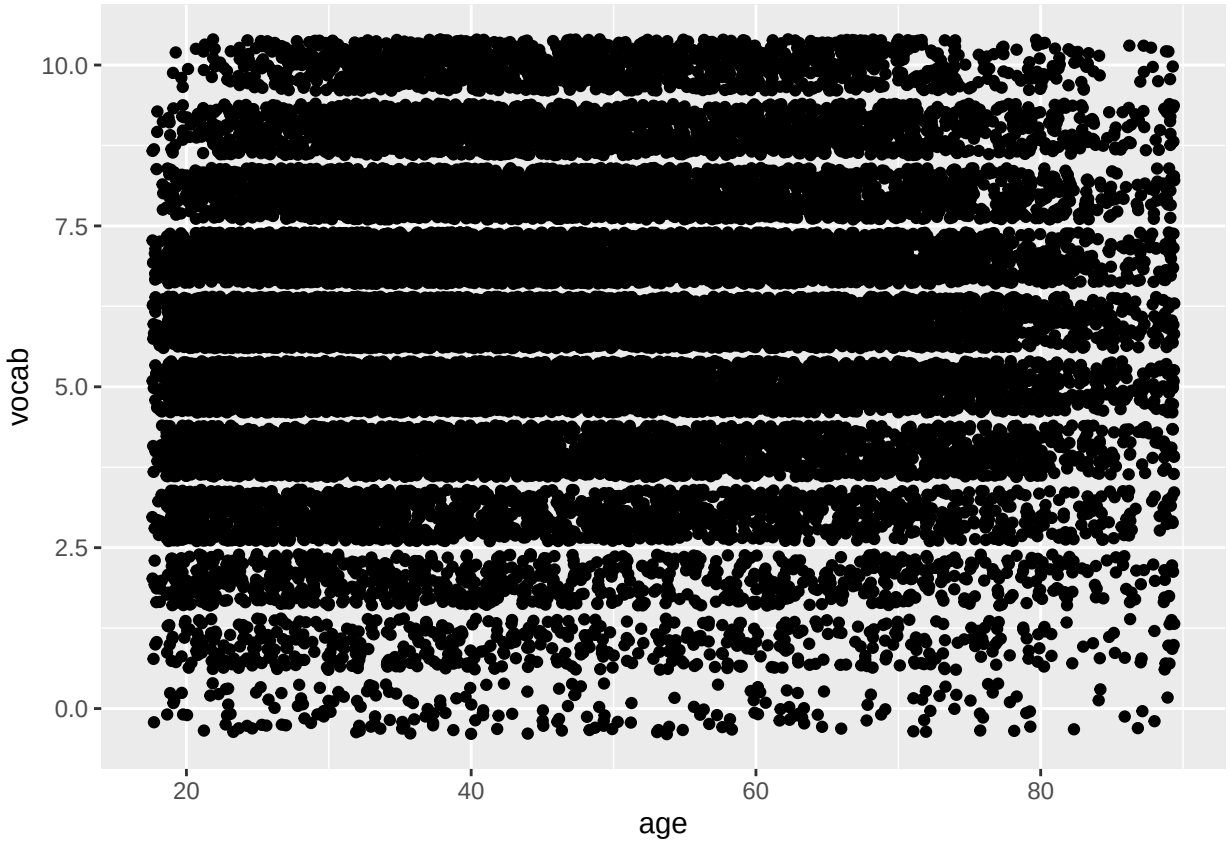
Create the best-looking plot you can to examine the `ageGroup` variable by `gender`. Does there appear to be an association? There are many ways to do this.

```
ggplot(GSSvocab) +
  aes(x = ageGroup, y = gender)   +
  geom_jitter(size = .01, shape = 1)
```



Create the best-looking plot you can to examine the `vocab` variable by `age`. Does there appear to be an association?

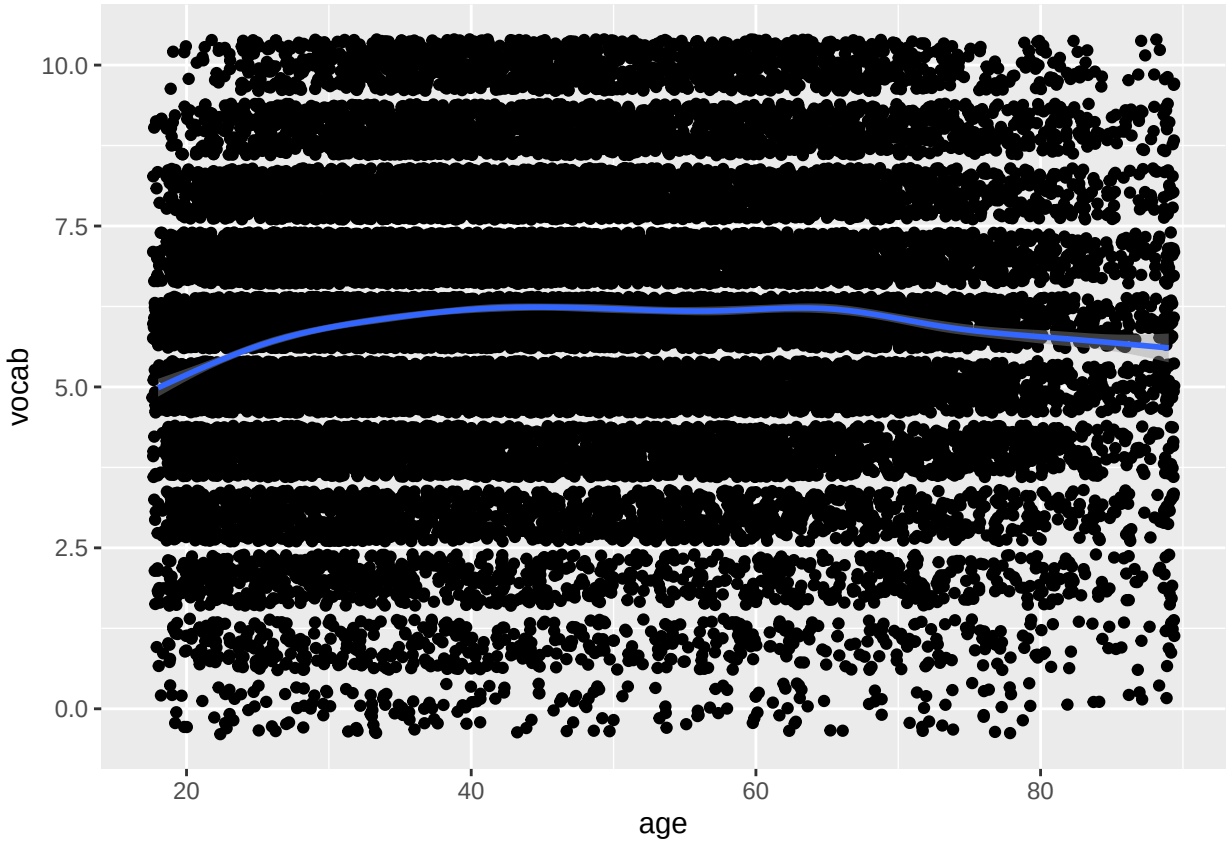
```
ggplot(GSSvocab) +
  aes(x = age, y = vocab)   +
  geom_jitter()
```



Add an estimate of $f(x)$ using the smoothing geometry to the previous plot. Does there appear to be an association now?

```
ggplot(GSSvocab) +
  aes(x = age, y = vocab)   +
  geom_jitter() +
  geom_smooth()
```

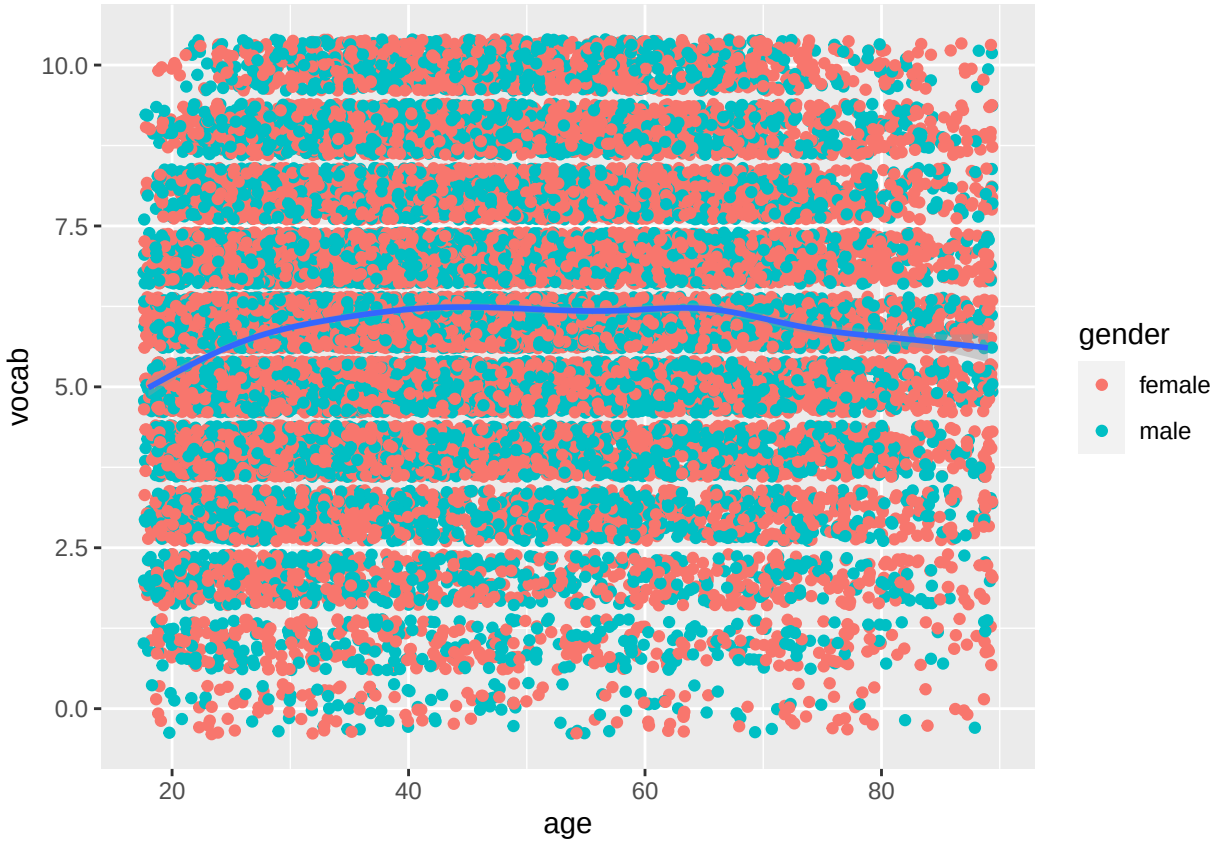
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



Using the plot from the previous question, create the best looking plot overloading with variable `gender`. Does there appear to be an interaction of `gender` and `age`?

```
ggplot(GSSvocab) +
  aes(x = age, y = vocab)   +
  geom_jitter(aes(col = gender)) +
  geom_smooth()
```

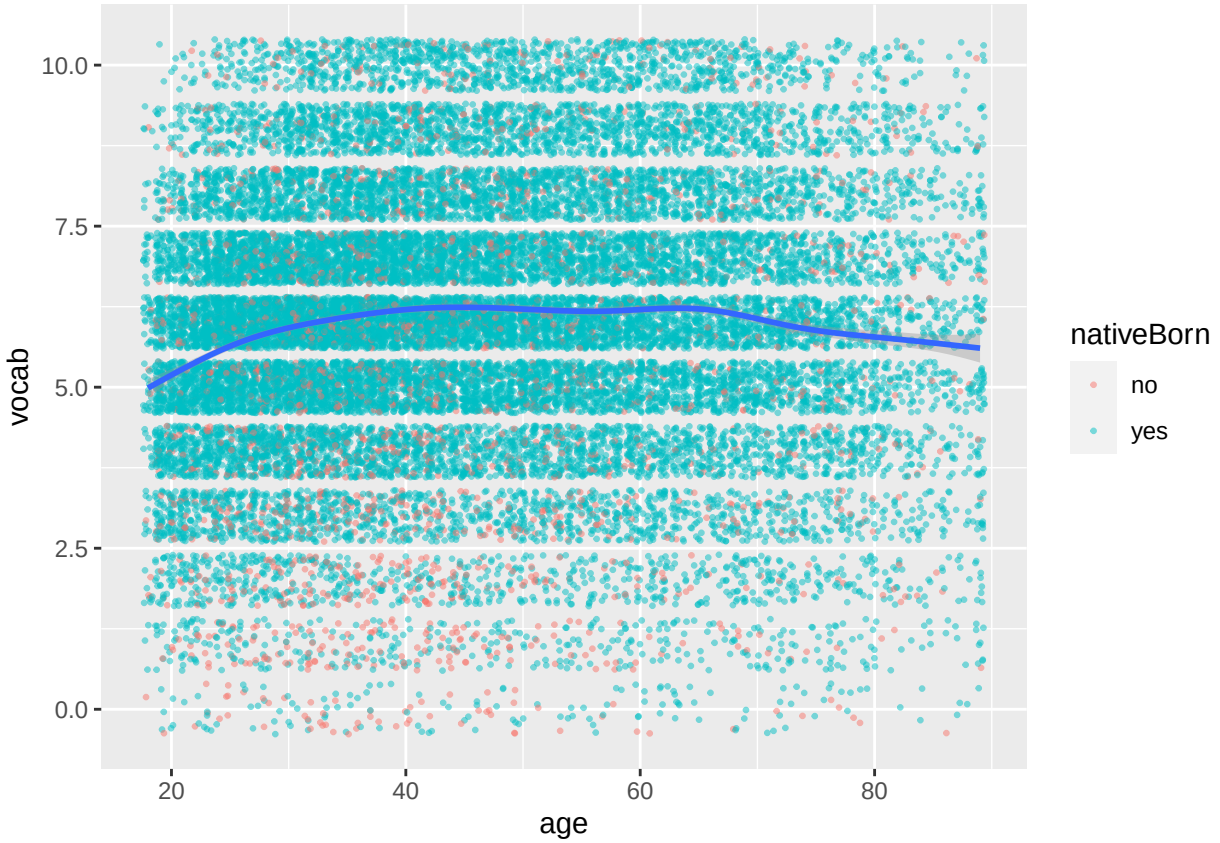
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```



Using the plot from the previous question, create the best looking plot overloading with variable `nativeBorn`. Does there appear to be an interaction of `nativeBorn` and `age`?

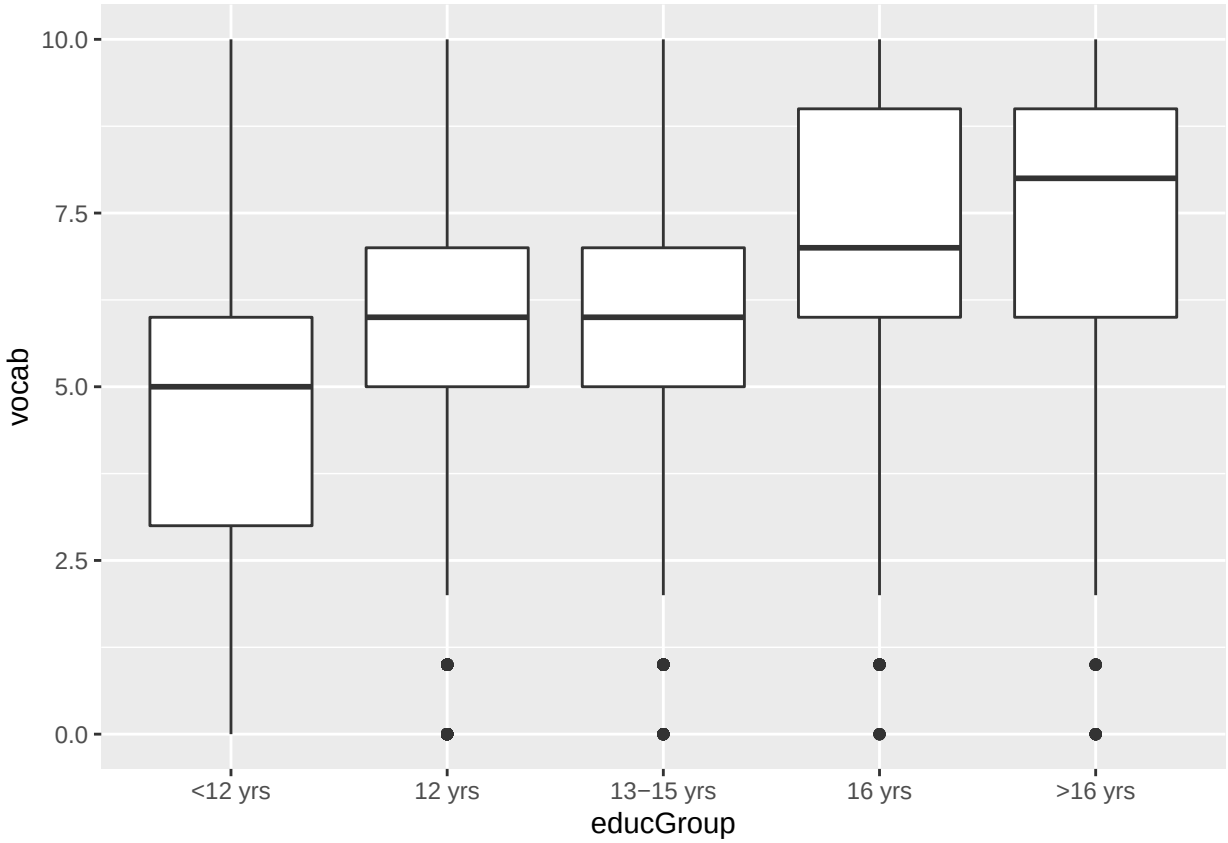
```
ggplot(GSSvocab) +
  aes(x = age, y = vocab)   +
  geom_jitter(aes(col = nativeBorn), size = .5, alpha = .5) +
  geom_smooth()
```

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

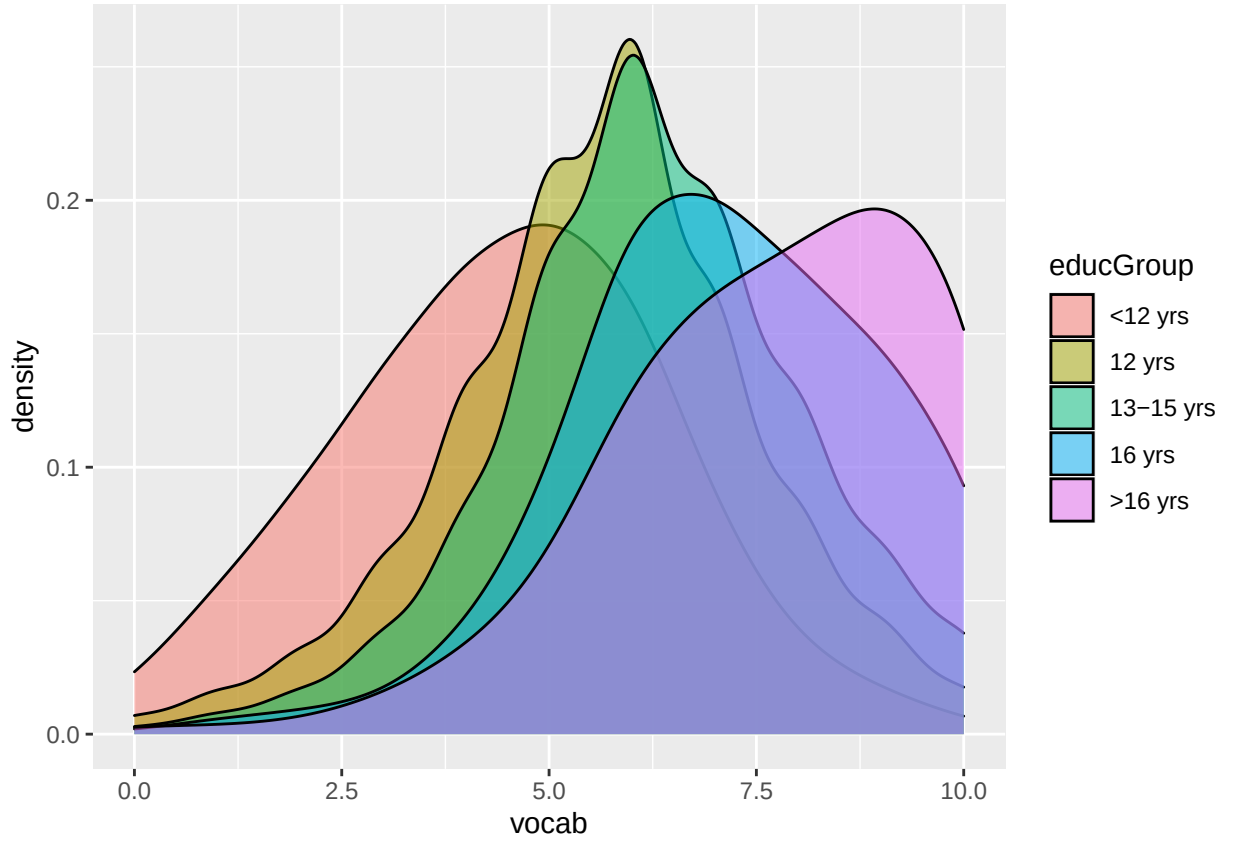


Create two different plots and identify the best-looking plot you can to examine the `vocab` variable by `educGroup`. Does there appear to be an association?

```
ggplot(GSSvocab) +
  aes(x = educGroup, y = vocab)   +
  geom_boxplot()
```



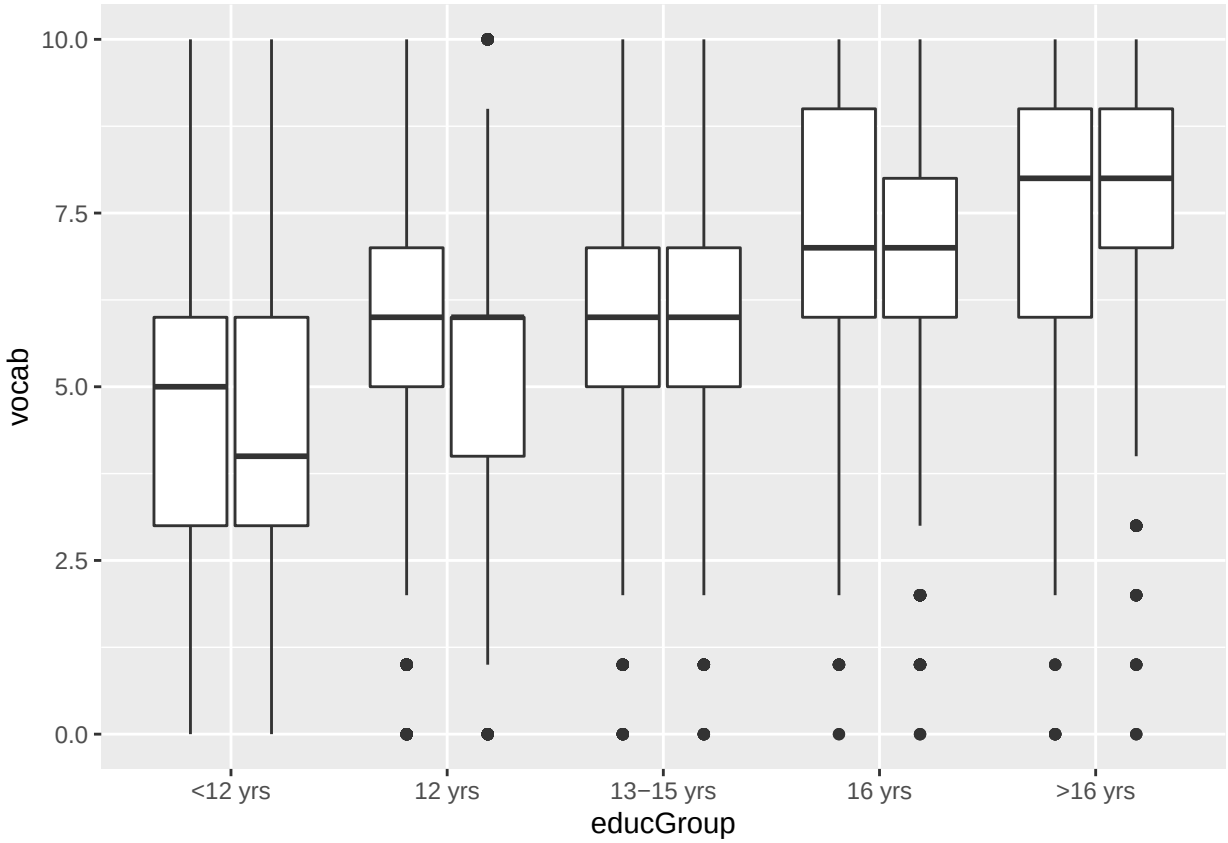

```
ggplot(GSSvocab) +
  aes(x = vocab) +
  geom_density(aes(fill = educGroup), adjust = 2, alpha = .5)
```



Using the best-looking plot from the previous question, create the best looking plot overloading with variable `gender`. Does there appear to be an interaction of `gender` and `educGroup`?

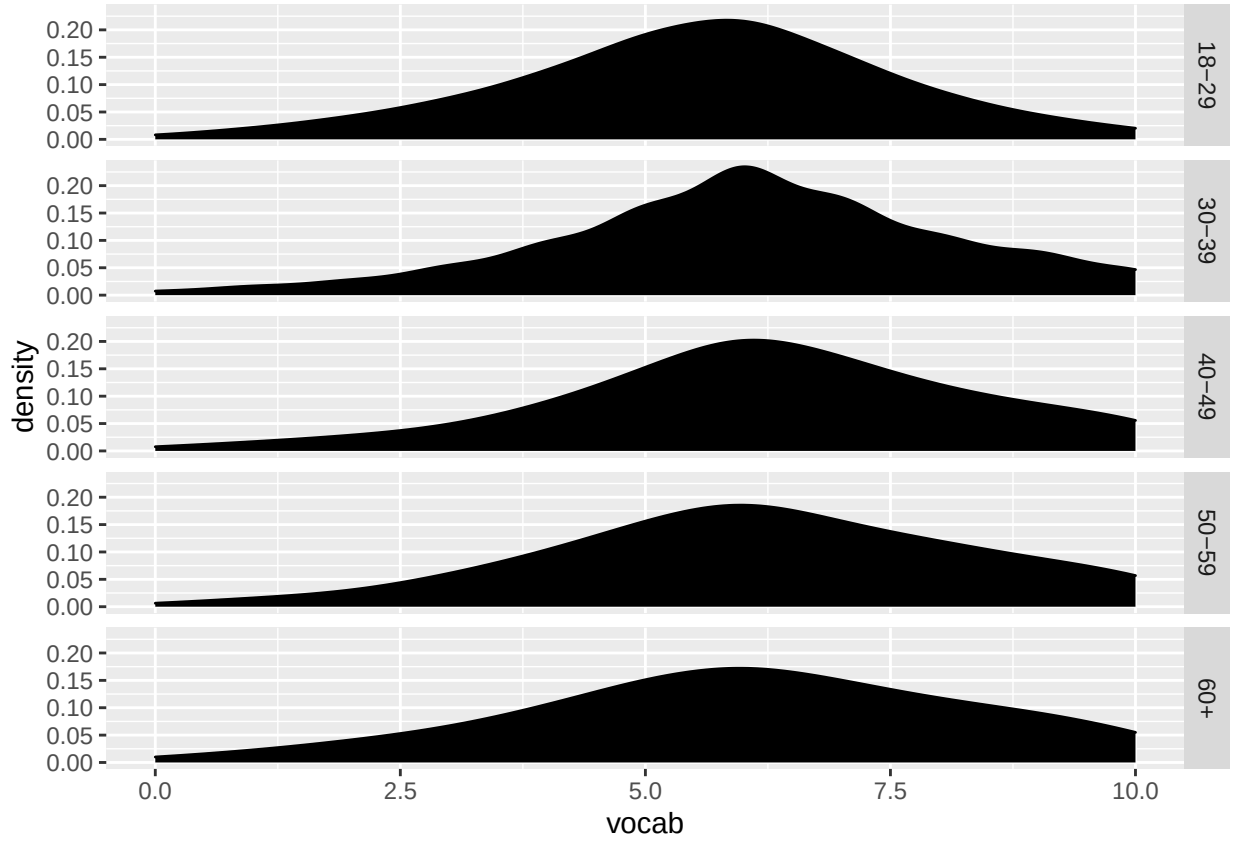
```
ggplot(GSSvocab) +
  aes(x = educGroup, y = vocab)   +
  geom_boxplot(aes(cols = gender))
```

```
## Warning: Ignoring unknown aesthetics: cols
```

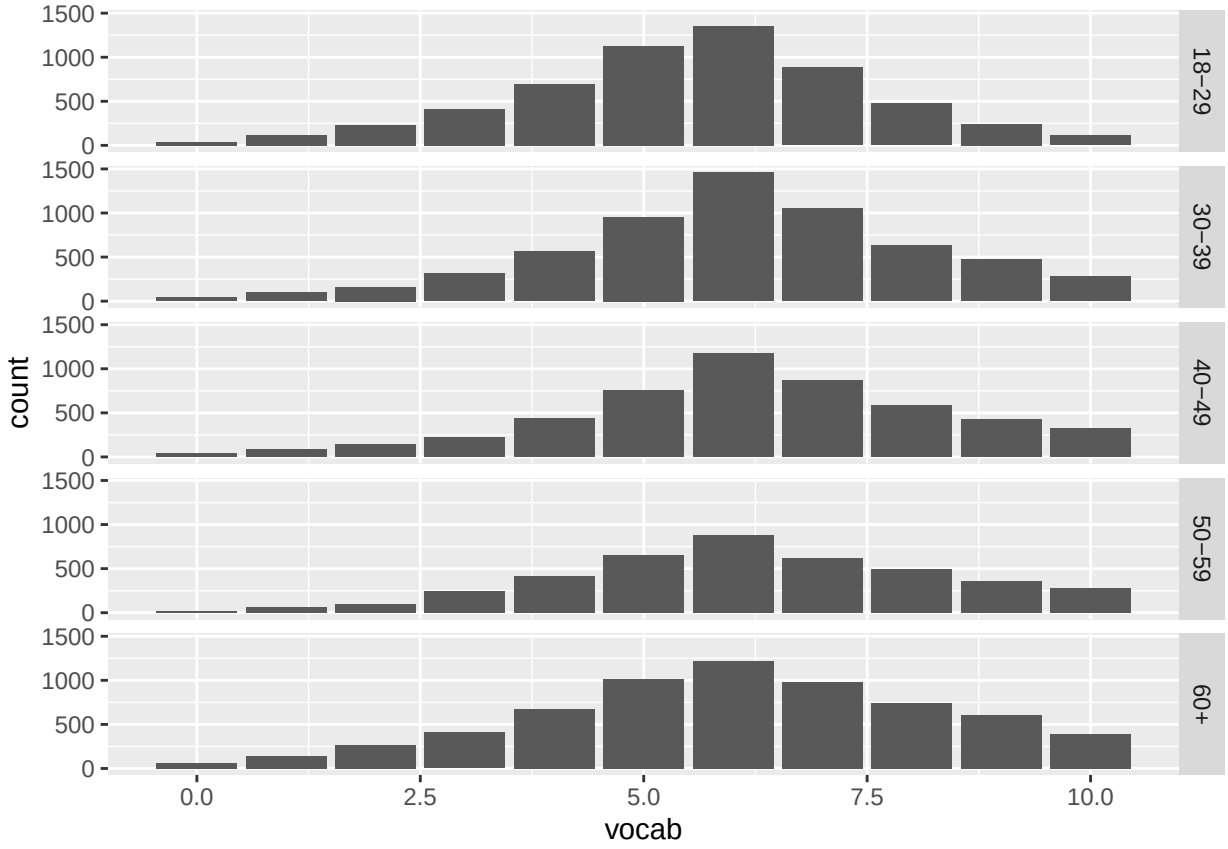


Using facets, examine the relationship between `vocab` and `ageGroup`. Are we getting dumber?

```
ggplot(GSSvocab) +
  aes(x = vocab) +
  geom_density(adjust = 2, fill = "black") +
  facet_grid(ageGroup~.)
```




```
ggplot(GSSvocab) +
  aes(x = vocab) +
  geom_bar() +
  facet_grid(ageGroup~.)
```



Probability Estimation and Model Selection

Load up the `adult` in the package `ucidata` dataset and remove missingness and the variable `fnlwgt`:

```
pacman::p_load_gh("coatless/ucidata")
data(adult)
adult = na.omit(adult) #kill any observations with missingness
adult$fnlwgt = NULL
```

Cast income to binary where 1 is the `>50K` level.

```
adult$income = ifelse(adult$income == ">50K", 1, 0)
```

We are going to do some dataset cleanup now. But in every cleanup job, there's always more to clean! So don't expect this cleanup to be perfect.

Firstly, a couple of small things. In variable `marital_status` collapse the levels `Married-AF-spouse` (armed force marriage) and `Married-civ-spouse` (civilian marriage) together into one level called `Married`. Then in variable `education` collapse the levels `1st-4th` and `Preschool` together into a level called `<=4th`.

```
adult$marital_status = as.character(adult$marital_status)
adult$marital_status = ifelse(adult$marital_status == "Married-AF-spouse" | adult$marital_status == "Ma
adult$marital_status = as.factor(adult$marital_status)
```

```
adult$education = as.character(adult$education)
adult$education = ifelse(adult$education == "1st-4th" | adult$education == "Preschool", "<=4th", adult$
adult$education = as.factor(adult$education)
```

Create a model matrix `Xmm` (for this prediction task on just the raw features) and show that it is *not* full rank (i.e. the result of `ncol` is greater than the result of `Matrix::rankMatrix`).

```
Xmm = model.matrix(income~., adult)
ncol(Xmm)
```

```
## [1] 95
```

```
Matrix::rankMatrix(Xmm)
```

```
## [1] 94
## attr(,"method")
## [1] "tolNorm2"
## attr(,"useGrad")
## [1] FALSE
## attr(,"tol")
## [1] 6.697087e-12
```

Now tabulate and sort the variable `native_country`.

```
tab = sort(table(adult$native_country))

tab
```

```
##
##         Holand-Netherlands                   Scotland
##                          1                         11
##                   Honduras                    Hungary
##                         12                         13
## Outlying-US(Guam-USVI-etc)                  Yugoslavia
##                         14                         16
##                       Laos                   Thailand
##                         17                         17
##                   Cambodia              Trinadad&Tobago
##                         18                         18
##                       Hong                    Ireland
##                         19                         24
##                    Ecuador                     France
##                         27                         27
##                     Greece                       Peru
##                         29                         30
##                  Nicaragua                   Portugal
##                         33                         34
##                      Haiti                       Iran
##                         42                         42
##                     Taiwan                   Columbia
##                         42                         56
```

```
##                      Poland                       Japan
##                          56                          59
##                   Guatemala                     Vietnam
##                          63                          64
##          Dominican-Republic                       China
##                          67                          68
##                       Italy                       South
##                          68                          71
##                     Jamaica                     England
##                          80                          86
##                        Cuba                 El-Salvador
##                          92                         100
##                       India                      Canada
##                         100                         107
##                  Puerto-Rico                     Germany
##                         109                         128
##                  Philippines                      Mexico
##                         188                         610
##                United-States
##                         27503
```

Do you see rare levels in this variable? Explain why this may be a problem.

Of course! (Holand-Netherlands, for example.) This may be a problem in predictive abilities of a model, since a very small sample size may be no true reflection of a population as a whole. (Stats 101.)

Collapse all levels that have less than 50 observations into a new level called `other`. This is a very common data science trick that will make your life much easier. If you can't hope to model rare levels, just give up and do something practical! I would recommend first casting the variable to type "character" and then do the level reduction and then recasting back to type `factor`. Tabulate and sort the variable `native_country` to make sure you did it right.

```
adult$native_country = as.character(adult$native_country)
adult$native_country = ifelse(adult$native_country %in% names(tab[tab < 50]), "Other", adult$native_cou

adult$native_country = as.factor(adult$native_country)
```

We're still not done getting this data down to full rank. Take a look at the model matrix just for `workclass` and `occupation`. Is it full rank?

```
Xmm = model.matrix(income~workclass + occupation, adult)
ncol(Xmm)
```

```
## [1] 21
```

```
Matrix::rankMatrix(Xmm)
```

```
## [1] 20
## attr(,"method")
## [1] "tolNorm2"
## attr(,"useGrad")
## [1] FALSE
## attr(,"tol")
## [1] 6.697087e-12
```

These variables are similar and they probably should be interacted anyway eventually. Let's combine them into one factor. Create a character variable named `worktype` that is the result of concatenating `occupation` and `workclass` together with a ":" in between. Use the `paste` function with the `sep` argument (this casts automatically to type `character`). Then tabulate its levels and sort.

```
adult$worktype = paste(adult$occupation, adult$workclass, sep = " : ")
head(adult)
```

```
##   age        workclass education education_num        marital_status
## 1  50 Self-emp-not-inc Bachelors            13               Married
## 2  38          Private   HS-grad             9              Divorced
## 3  53          Private      11th             7               Married
## 4  28          Private Bachelors            13               Married
## 5  37          Private   Masters            14               Married
## 6  49          Private       9th             5 Married-spouse-absent
##          occupation  relationship  race    sex capital_gain capital_loss
## 1   Exec-managerial       Husband White   Male            0            0
## 2 Handlers-cleaners Not-in-family White   Male            0            0
## 3 Handlers-cleaners       Husband Black   Male            0            0
## 4    Prof-specialty          Wife Black Female            0            0
## 5   Exec-managerial          Wife White Female            0            0
## 6     Other-service Not-in-family Black Female            0            0
##   hours_per_week native_country income                          worktype
## 1             13  United-States      0 Exec-managerial : Self-emp-not-inc
## 2             40  United-States      0        Handlers-cleaners : Private
## 3             40  United-States      0        Handlers-cleaners : Private
## 4             40           Cuba      0           Prof-specialty : Private
## 5             40  United-States      0          Exec-managerial : Private
## 6             16        Jamaica      0            Other-service : Private
```

Like the `native_country` exercise, there are a lot of rare levels. Collapse levels with less than 100 observations to type `other` and then cast this variable `worktype` as type `factor`. Recheck the tabulation to ensure you did this correct.

```
tab = sort(table(adult$worktype))
tab
```

```
##
##           Craft-repair : Without-pay      Handlers-cleaners : Without-pay
##                                     1                                    1
##      Machine-op-inspct : Without-pay          Other-service : Without-pay
##                                     1                                    1
##       Transport-moving : Without-pay     Handlers-cleaners : Self-emp-inc
##                                     1                                    2
##           Adm-clerical : Without-pay          Tech-support : Self-emp-inc
##                                     3                                    3
##       Protective-serv : Self-emp-inc        Farming-fishing : Without-pay
##                                     5                                    6
##   Protective-serv : Self-emp-not-inc                   Sales : Local-gov
##                                     6                                    7
##        Farming-fishing : Federal-gov           Armed-Forces : Federal-gov
##                                     8                                    9
##        Handlers-cleaners : State-gov     Machine-op-inspct : Self-emp-inc
```

```
##                                   9                                   10
##        Machine-op-inspct : Local-gov                 Sales : State-gov
##                                  11                                   11
##        Machine-op-inspct : State-gov     Machine-op-inspct : Federal-gov
##                                  13                                   14
##                 Sales : Federal-gov         Farming-fishing : State-gov
##                                  14                                   15
## Handlers-cleaners : Self-emp-not-inc     Handlers-cleaners : Federal-gov
##                                  15                                   22
##       Transport-moving : Federal-gov     Tech-support : Self-emp-not-inc
##                                  24                                   26
##      Transport-moving : Self-emp-inc        Other-service : Self-emp-inc
##                                  26                                   27
##        Protective-serv : Federal-gov         Adm-clerical : Self-emp-inc
##                                  27                                   28
##          Farming-fishing : Local-gov         Other-service : Federal-gov
##                                  29                                   34
## Machine-op-inspct : Self-emp-not-inc           Tech-support : Local-gov
##                                  35                                   38
##         Transport-moving : State-gov       Handlers-cleaners : Local-gov
##                                  41                                   46
##      Adm-clerical : Self-emp-not-inc      Farming-fishing : Self-emp-inc
##                                  49                                   51
##             Craft-repair : State-gov           Tech-support : State-gov
##                                  55                                   56
##           Craft-repair : Federal-gov         Tech-support : Federal-gov
##                                  63                                   66
##          Craft-repair : Self-emp-inc        Transport-moving : Local-gov
##                                  99                                  115
##          Protective-serv : State-gov  Transport-moving : Self-emp-not-inc
##                                 116                                  118
##            Other-service : State-gov            Craft-repair : Local-gov
##                                 123                                  143
##            Priv-house-serv : Private       Prof-specialty : Self-emp-inc
##                                 143                                  157
##         Prof-specialty : Federal-gov    Other-service : Self-emp-not-inc
##                                 167                                  173
##        Exec-managerial : Federal-gov         Exec-managerial : State-gov
##                                 179                                  186
##            Protective-serv : Private          Other-service : Local-gov
##                                 186                                  189
##          Exec-managerial : Local-gov           Adm-clerical : State-gov
##                                 212                                  250
##             Adm-clerical : Local-gov               Sales : Self-emp-inc
##                                 281                                  281
##          Protective-serv : Local-gov          Adm-clerical : Federal-gov
##                                 304                                  316
##    Prof-specialty : Self-emp-not-inc           Sales : Self-emp-not-inc
##                                 365                                  376
##   Exec-managerial : Self-emp-not-inc      Exec-managerial : Self-emp-inc
##                                 383                                  385
##           Prof-specialty : State-gov   Farming-fishing : Self-emp-not-inc
##                                 403                                  430
##            Farming-fishing : Private     Craft-repair : Self-emp-not-inc
```

```
##                                 450                                 523
##           Prof-specialty : Local-gov              Tech-support : Private
##                                 692                                 723
##           Transport-moving : Private          Handlers-cleaners : Private
##                                1247                                1255
##          Machine-op-inspct : Private             Prof-specialty : Private
##                                1882                                2254
##            Exec-managerial : Private              Other-service : Private
##                                2647                                2665
##               Adm-clerical : Private                       Sales : Private
##                                2793                                2895
##               Craft-repair : Private
##                                3146
```

```
adult$worktype = ifelse(adult$worktype %in% names(tab[tab < 100]), "Other", adult$worktype)

tab = sort(table(adult$worktype))
tab
```

```
##
##        Transport-moving : Local-gov         Protective-serv : State-gov
##                                 115                                 116
## Transport-moving : Self-emp-not-inc           Other-service : State-gov
##                                 118                                 123
##            Craft-repair : Local-gov           Priv-house-serv : Private
##                                 143                                 143
##       Prof-specialty : Self-emp-inc        Prof-specialty : Federal-gov
##                                 157                                 167
##    Other-service : Self-emp-not-inc       Exec-managerial : Federal-gov
##                                 173                                 179
##         Exec-managerial : State-gov           Protective-serv : Private
##                                 186                                 186
##           Other-service : Local-gov         Exec-managerial : Local-gov
##                                 189                                 212
##            Adm-clerical : State-gov            Adm-clerical : Local-gov
##                                 250                                 281
##                Sales : Self-emp-inc         Protective-serv : Local-gov
##                                 281                                 304
##          Adm-clerical : Federal-gov   Prof-specialty : Self-emp-not-inc
##                                 316                                 365
##            Sales : Self-emp-not-inc  Exec-managerial : Self-emp-not-inc
##                                 376                                 383
##      Exec-managerial : Self-emp-inc          Prof-specialty : State-gov
##                                 385                                 403
##  Farming-fishing : Self-emp-not-inc           Farming-fishing : Private
##                                 430                                 450
##     Craft-repair : Self-emp-not-inc          Prof-specialty : Local-gov
##                                 523                                 692
##              Tech-support : Private                               Other
##                                 723                                1008
##           Transport-moving : Private         Handlers-cleaners : Private
##                                1247                                1255
##          Machine-op-inspct : Private             Prof-specialty : Private
##                                1882                                2254
```

```
##           Exec-managerial : Private             Other-service : Private
##                                2647                                2665
##              Adm-clerical : Private                    Sales : Private
##                                2793                                2895
##              Craft-repair : Private
##                                3146
```

To do at home: merge the two variables `relationship` and `marital_status` together in a similar way to what we did here.

```
adult$worktype = paste(adult$relationship, adult$marital_status, sep = " : ")
head(adult)
```

```
##   age        workclass education education_num        marital_status
## 1  50 Self-emp-not-inc Bachelors            13               Married
## 2  38          Private   HS-grad             9              Divorced
## 3  53          Private      11th             7               Married
## 4  28          Private Bachelors            13               Married
## 5  37          Private   Masters            14               Married
## 6  49          Private       9th             5 Married-spouse-absent
##          occupation  relationship  race    sex capital_gain capital_loss
## 1   Exec-managerial       Husband White   Male            0            0
## 2 Handlers-cleaners Not-in-family White   Male            0            0
## 3 Handlers-cleaners       Husband Black   Male            0            0
## 4    Prof-specialty          Wife Black Female            0            0
## 5   Exec-managerial          Wife White Female            0            0
## 6     Other-service Not-in-family Black Female            0            0
##   hours_per_week native_country income                           worktype
## 1             13  United-States      0                 Husband : Married
## 2             40  United-States      0           Not-in-family : Divorced
## 3             40  United-States      0                 Husband : Married
## 4             40           Cuba      0                    Wife : Married
## 5             40  United-States      0                    Wife : Married
## 6             16        Jamaica      0 Not-in-family : Married-spouse-absent
```

```
tab = sort(table(adult$worktype))
tab
```

```
##
##                   Own-child : Widowed               Not-in-family : Married
##                                    12                                    14
## Other-relative : Married-spouse-absent             Other-relative : Widowed
##                                    26                                    40
##      Own-child : Married-spouse-absent           Other-relative : Separated
##                                    43                                    53
##                   Own-child : Married                 Own-child : Separated
##                                    84                                    90
##             Other-relative : Divorced              Other-relative : Married
##                                   103                                   119
##      Unmarried : Married-spouse-absent  Not-in-family : Married-spouse-absent
##                                   120                                   181
##                  Own-child : Divorced                   Unmarried : Widowed
##                                   308                                   343
```

```
##              Not-in-family : Separated                  Unmarried : Separated
##                                    383                                     413
##                Not-in-family : Widowed         Other-relative : Never-married
##                                    432                                     548
##              Unmarried : Never-married                          Wife : Married
##                                    801                                    1406
##                   Unmarried : Divorced               Not-in-family : Divorced
##                                   1535                                    2268
##              Own-child : Never-married          Not-in-family : Never-married
##                                   3929                                    4447
##                      Husband : Married
##                                  12463
```

We are finally ready to fit some probability estimation models for `income`! In lecture 16 we spoke about model selection using a cross-validation procedure. Let's build this up step by step. First, split the dataset into `Xtrain`, `ytrain`, `Xtest`, `ytest` using K=5.

```
set.seed(1984)
K = 5
test_prop = 1 / K
train_indices = sample(1 : nrow(adult), round((1 - test_prop) * nrow(adult)))
adult_train = adult[train_indices, ]
y_train = adult_train$income
X_train = adult_train
X_train$income = NULL
test_indices = setdiff(1 : nrow(adult), train_indices)
adult_test = adult[test_indices, ]
y_test = adult_test$income
X_test = adult_test
X_test$income = NULL
```

Create the following four models on the training data in a `list` objected named `prob_est_mods`: logit, probit, cloglog and cauchit (which we didn't do in class but might as well). For the linear component within the link function, just use the vanilla raw features using the `formula` object `vanilla`. Each model's key in the list is its link function name + "-vanilla". One for loop should do the trick here.

```
link_functions = c("logit", "probit", "cloglog", "cauchit")
vanilla = income ~ .
prob_est_mods = list()

for(link_function in link_functions){
  prob_est_mods[[paste(link_function, "vanilla", sep = "-")]] = glm(vanilla, adult_train, family = bino
}
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
## Warning: glm.fit: algorithm did not converge
```

Now let's get fancier. Let's do some variable transforms. Add `log_capital_loss` derived from `capital_loss` and `log_capital_gain` derived from `capital_gain`. Since there are zeroes here, use log_x = log(1 + x) instead of log_x = log(x). That's always a neat trick. Just add them directly to the data frame so they'll be picked up with the `.` inside of a formula.

```
adult$log_capital_loss = log(1 + adult$capital_loss)
adult$log_capital_gain = log(1 + adult$capital_gain)

head(adult, 10)
```

```
##    age        workclass    education education_num        marital_status
## 1   50 Self-emp-not-inc    Bachelors            13               Married
## 2   38          Private      HS-grad             9              Divorced
## 3   53          Private         11th             7               Married
## 4   28          Private    Bachelors            13               Married
## 5   37          Private      Masters            14               Married
## 6   49          Private          9th             5 Married-spouse-absent
## 7   52 Self-emp-not-inc      HS-grad             9               Married
## 8   31          Private      Masters            14         Never-married
## 9   42          Private    Bachelors            13               Married
## 10  37          Private Some-college            10               Married
##           occupation  relationship  race    sex capital_gain capital_loss
## 1    Exec-managerial       Husband White   Male            0            0
## 2  Handlers-cleaners Not-in-family White   Male            0            0
## 3  Handlers-cleaners       Husband Black   Male            0            0
## 4     Prof-specialty          Wife Black Female            0            0
## 5    Exec-managerial          Wife White Female            0            0
## 6      Other-service Not-in-family Black Female            0            0
## 7    Exec-managerial       Husband White   Male            0            0
## 8     Prof-specialty Not-in-family White Female        14084            0
## 9    Exec-managerial       Husband White   Male         5178            0
## 10   Exec-managerial       Husband Black   Male            0            0
##    hours_per_week native_country income                           worktype
## 1              13  United-States      0                 Husband : Married
## 2              40  United-States      0            Not-in-family : Divorced
## 3              40  United-States      0                 Husband : Married
## 4              40           Cuba      0                    Wife : Married
## 5              40  United-States      0                    Wife : Married
## 6              16        Jamaica      0 Not-in-family : Married-spouse-absent
## 7              45  United-States      1                 Husband : Married
## 8              50  United-States      1       Not-in-family : Never-married
## 9              40  United-States      1                 Husband : Married
## 10             80  United-States      1                 Husband : Married
##    log_capital_loss log_capital_gain
## 1                 0        0.000000
## 2                 0        0.000000
## 3                 0        0.000000
```

```
## 4                  0         0.000000
## 5                  0         0.000000
## 6                  0         0.000000
## 7                  0         0.000000
## 8                  0         9.552866
## 9                  0         8.552367
## 10                 0         0.000000
```

Create a density plot that shows the age distribution by `income`.

??? I'm not sure what to do here ???

What do you see? Is this expected using common sense?

I don't see anything. This chunk confused me. However, if I could make the plot, I would expect to see that the density of income is lower at younger ages, grows through 50's or 60's, and then becomes more sparse as age continues to increase. (That is, I would expect to see the density greater in the middle, and more sparse on either side.)

Now let's fit the same models with all link functions on a formula called `age_interactions` that uses interactions for `age` with all of the variables. Add all these models to the `prob_est_mods` list.

```
link_functions = c("logit", "probit", "cloglog", "cauchit")
age_interactions = age ~ .

for(link_function in link_functions){
  prob_est_mods[[paste(link_function, "age_interactions", sep = "-")]] = glm(age_interactions, adult_tra
}

##This does not work, as in the case of "vanilla", and I have no idea what the error is referring to. :
```

Create a function called `brier_score` that takes in a probability estimation model, a dataframe `X` and its responses `y` and then calculates the brier score.

```
brier_score = function(prob_est_mod, X, y){
  p_hat = predict(prob_est_mod, X, type = "response")
  mean(-(y - p_hat)^2)
}
```

Now, calculate the in-sample Brier scores for all models. You can use the function `lapply` to iterate over the list and pass in in the function `brier_score`.

```
lapply(prob_est_mods, brier_score, X_train, y_train)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## $`logit-vanilla`
## [1] -0.1038628
##
## $`probit-vanilla`
## [1] -0.6725724
##
## $`cloglog-vanilla`
## [1] -0.5631398
##
## $`cauchit-vanilla`
## [1] -0.1055769
```

```
##rank-deficient... need to look over this better
```

Now, calculate the out-of-sample Brier scores for all models. You can use the function `lapply` to iterate over the list and pass in the function `brier_score`.

```
lapply(prob_est_mods, brier_score, X_test, y_test)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading

## $`logit-vanilla`
## [1] -0.1034953
##
## $`probit-vanilla`
## [1] -0.6678133
##
## $`cloglog-vanilla`
## [1] -0.5590186
##
## $`cauchit-vanilla`
## [1] -0.1046053
```

Which model wins in sample and which wins out of sample? Do you expect these results? Explain.

In both in sample and out of sample (based on these rank deficient computations), logit is the winner. I suppose this is somewhat expected, as in class it was discussed that logit is usually used more, but this is very poor justification.

What is wrong with this model selection procedure? There are a few things wrong.

For one, here particularly, the fit is apparently rank-deficient, as that is what the error tells me, and according to the error, the prediction from a rank deficient fit may be misleading.

Run all the models again. This time do three splits: subtrain, select and test. After selecting the best model, provide a true oos Brier score for the winning model.

```
#TO-DO
## Work on me when you have more time.
```