Lab 9: Fitting Models to Data

Name: Rufus Petrie

This week's agenda: exploratory data analysis, cleaning data, fitting linear/logistic models, and using associated utility functions.

Prostate cancer data

Recall the data set on 97 men who have prostate cancer (from the book The Elements of Statistical Learning). Reading it into our R session:

```
pros.df =
  read.table("http://www.stat.cmu.edu/~ryantibs/statcomp/data/pros.dat")
dim(pros.df)
```

```
## [1] 97  9
```

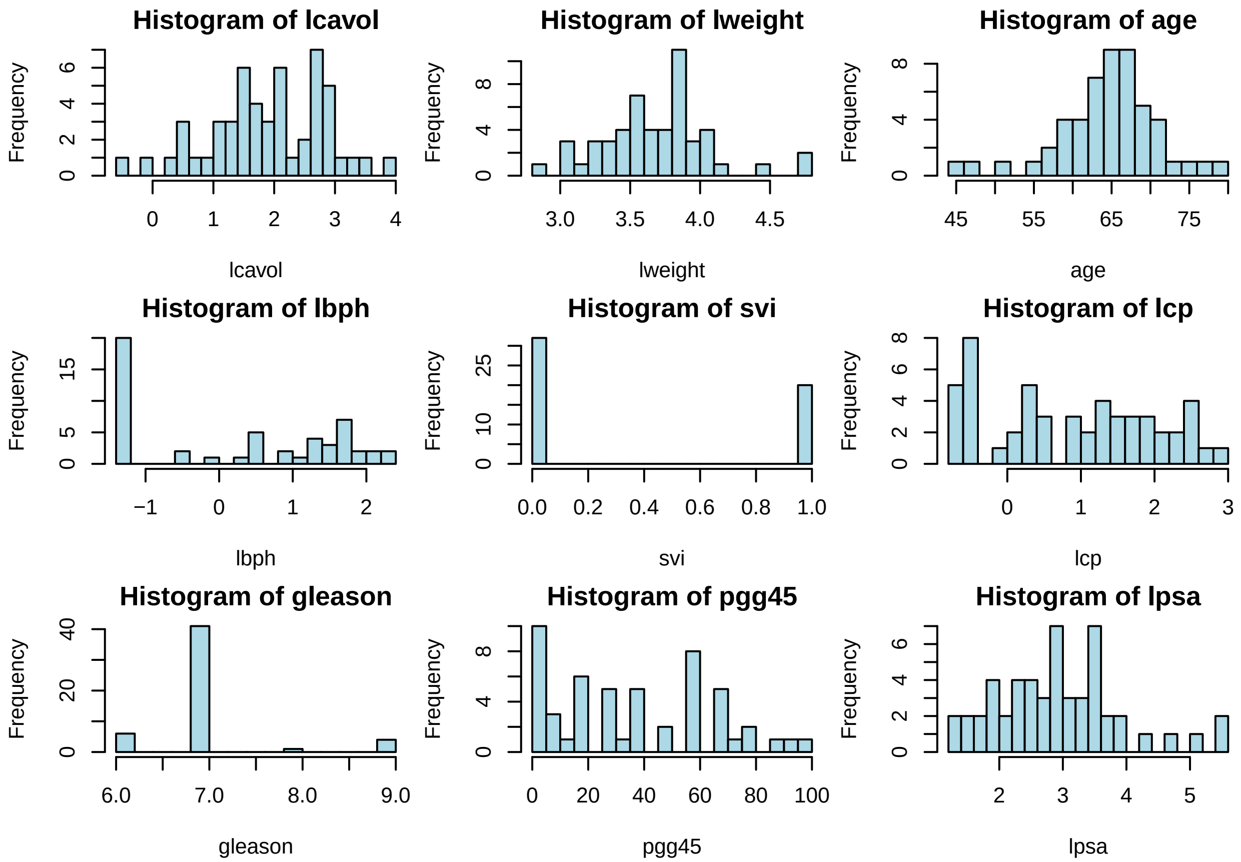
```
head(pros.df, 3)
```

```
##       lcavol  lweight age      lbph svi       lcp gleason pgg45       lpsa
## 1 -0.5798185 2.769459  50 -1.386294   0 -1.386294       6     0 -0.4307829
## 2 -0.9942523 3.319626  58 -1.386294   0 -1.386294       6     0 -0.1625189
## 3 -0.5108256 2.691243  74 -1.386294   0 -1.386294       7    20 -0.1625189
```

Simple exploration and linear modeling

- **1a.** Define `pros.df.subset` to be the subset of observations (rows) of the prostate data set such the `lcp` measurement is greater than the minimum value (the minimum value happens to be `log(0.25)`, but you should not hardcode this value and should work it out from the data). As in lecture, plot histograms of all of the variables in `pros.df.subset`. Comment on any differences you see between these distributions and the ones in lecture.

```
pros.df.subset <- pros.df[pros.df$lcp > min(pros.df$lcp),]
par(mfrow=c(3,3), mar=c(4,4,2,0.5))
for (j in 1:ncol(pros.df.subset)) {
  hist(pros.df.subset[,j], xlab=colnames(pros.df.subset)[j],
       main=paste("Histogram of", colnames(pros.df.subset)[j]),
       col="lightblue", breaks=20)
}
```



Most of the histograms look the same except for lcp, pgg45, and lpsa. lcp and pgg45 no longer have large bars are zero because we removed the observations with the minimum lcp score. Additionally, lpsa looks closer to a uniform distribution than a bell curve now.

- **1b.** Also as in lecture, compute and display correlations between all pairs of variables in `pros.df.subset`. Report the two highest correlations between pairs of (distinct) variables, and also report the names of the associated variables. Are these different from answers that were computed on the full data set?

```
pros.cor = cor(pros.df.subset)
round(pros.cor,3)
```

```
##          lcavol lweight    age   lbph    svi    lcp gleason pgg45   lpsa
## lcavol    1.000   0.221  0.013 -0.241  0.568  0.805   0.285 0.317  0.624
## lweight   0.221   1.000  0.398  0.338  0.145  0.113   0.001 0.058  0.243
## age       0.013   0.398  1.000  0.311  0.086 -0.025   0.173 0.266 -0.059
## lbph     -0.241   0.338  0.311  1.000 -0.239 -0.179   0.041 0.065 -0.063
## svi       0.568   0.145  0.086 -0.239  1.000  0.625   0.170 0.325  0.601
## lcp       0.805   0.113 -0.025 -0.179  0.625  1.000   0.367 0.425  0.530
## gleason   0.285   0.001  0.173  0.041  0.170  0.367   1.000 0.618  0.152
## pgg45     0.317   0.058  0.266  0.065  0.325  0.425   0.618 1.000  0.268
## lpsa      0.624   0.243 -0.059 -0.063  0.601  0.530   0.152 0.268  1.000
```

```
# Note: set cor lower triangle/diagonal to zero so we can ignore autocorrelations
pros.cor[lower.tri(pros.cor,diag=TRUE)] = 0
pros.cor.sorted = sort(abs(pros.cor),decreasing=T)

# Top 2 correlations
pros.cor.sorted[1:2]
```

```
## [1] 0.8049728 0.6246382
```

```
# Top correlation
vars.big.cor = arrayInd(which(abs(pros.cor)==pros.cor.sorted[1]),
                        dim(pros.cor))
colnames(pros.df)[vars.big.cor]
```

```
## [1] "lcavol" "lcp"
```

```
# 2ns highest correlation
vars.big.cor = arrayInd(which(abs(pros.cor)==pros.cor.sorted[2]),
                        dim(pros.cor))
colnames(pros.df)[vars.big.cor]
```

```
## [1] "svi" "lcp"
```

Gleason and pgg45 no longer have the highest correlation. The lcavol/lpsa correlation is now highest, and svi/lcp is now the second highest.

- **1c.** Compute, using `lm()`, a linear regression model of `lpsa` (log PSA score) on `lcavol` (log cancer volume). Do this twice: once with the full data set, `pros.df`, and once with the subsetted data, `pros.df.subset`. Save the results as `pros.lm.` and `pros.subset.lm`, respectively. Using `coef()`, display the coefficients (intercept and slope) from each linear regression. Are they different?

```
pros.lm <- lm(lpsa ~ lcavol, data = pros.df)
pros.subset.lm <- lm(lpsa ~ lcavol, data = pros.df.subset)
coef(pros.lm)
```

```
## (Intercept)      lcavol
##   1.5072975   0.7193204
```

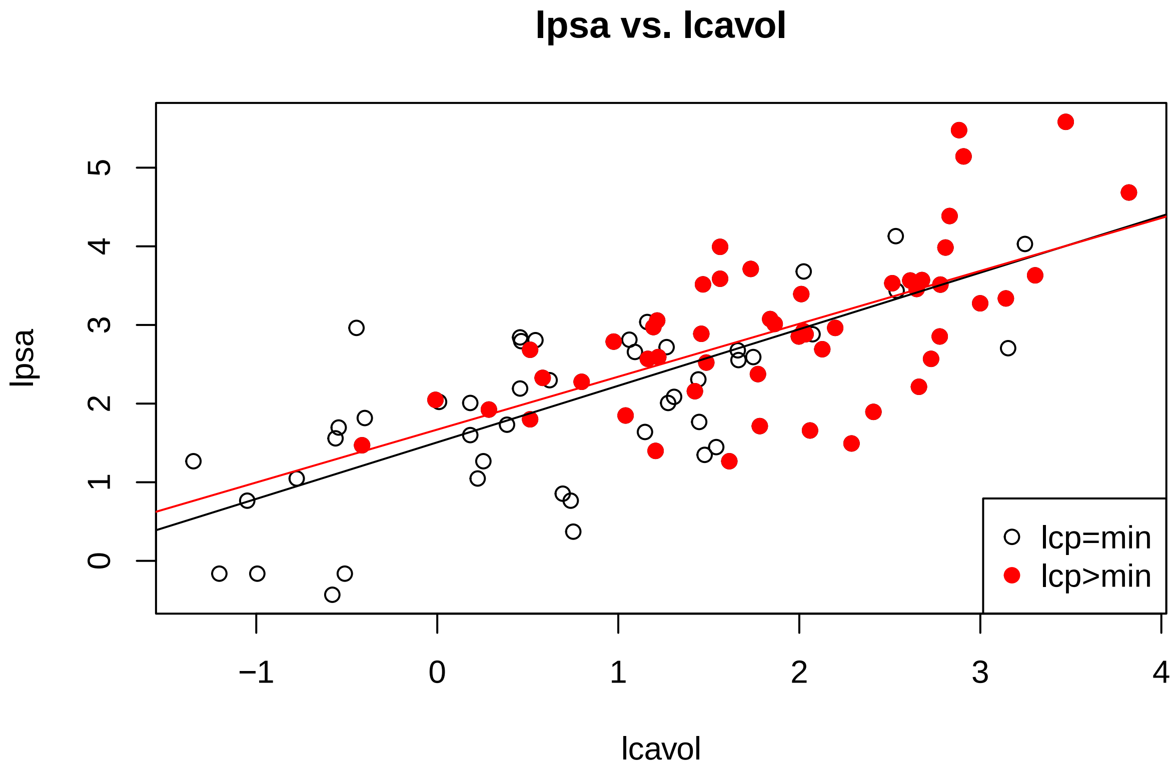
```
coef(pros.subset.lm)
```

```
## (Intercept)      lcavol
##   1.6695707   0.6725807
```

lcavol has a lower coefficient in the second model.

- **1d.** Let's produce a visualization to help us figure out how different these regression lines really are. Plot `lpsa` versus `lcavol`, using the full set of observations, in `pros.df`. Label the axes appropriately. Then, mark the observations in `pros.df.subset` by small filled red circles. Add a thick black line to your plot, displaying the fitted regression line from `pros.lm`. Add a thick red line, displaying the fitted regression line from `pros.subset.lm`. Add a legend that explains the color coding.

```
plot(pros.df$lcavol, pros.df$lpsa,
       main="lpsa vs. lcavol", xlab="lcavol", ylab="lpsa")
points(pros.df.subset$lcavol, pros.df.subset$lpsa, col="red", pch=19)
abline(a =1.5072975, b =,0.7193204, lty=1)
abline(a =1.6695707, b =,0.6725807, lty=1, col="red")
legend("bottomright", legend=c("lcp=min", "lcp>min"), col=c("black", "red"), pch=c(21, 19))
```



- **1e.** Compute again a linear regression of `lpsa` on `lcavol`, but now on two different subsets of the data: the first consisting of patients with SVI, and the second consistent of patients without SVI. Display the resulting coefficients (intercept and slope) from each model, and produce a plot just like the one in the last question, to visualize the different regression lines on top of the data. Do these two regression lines differ, and in what way?

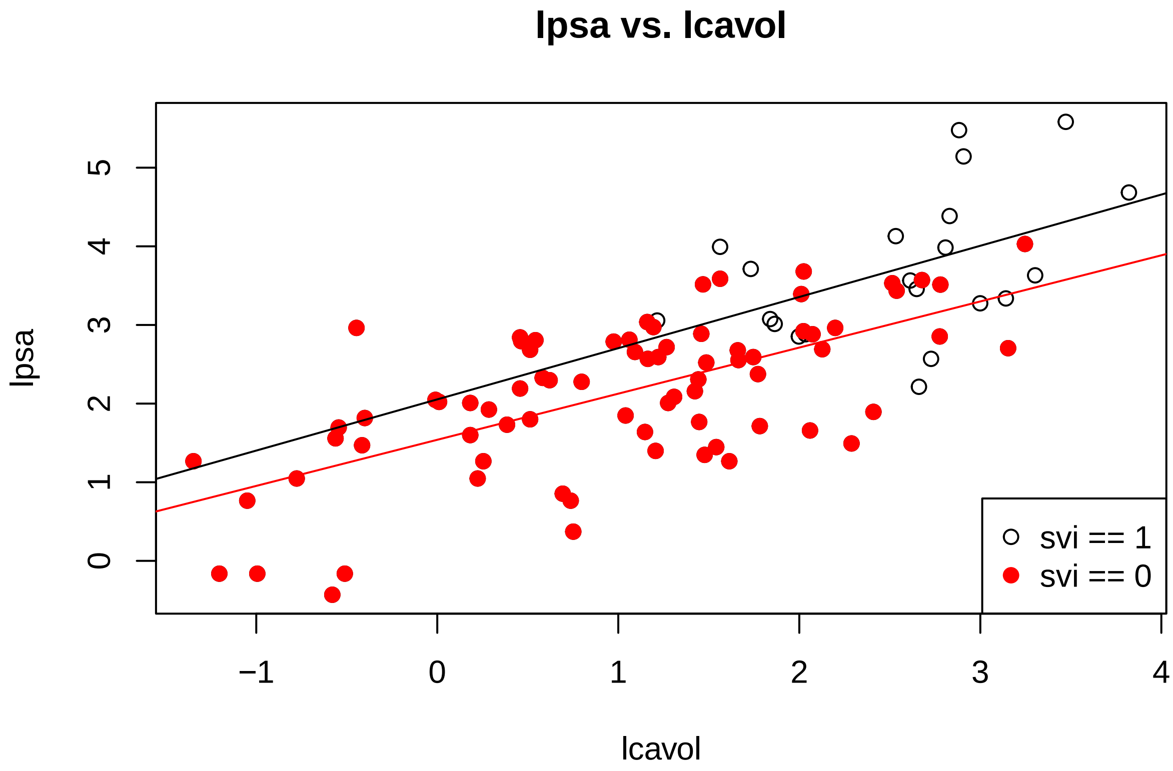
```
svi.lm <- lm(lpsa ~ lcavol, data = pros.df[pros.df$svi==1,])
nosvi.lm <- lm(lpsa ~ lcavol, data = pros.df[pros.df$svi==0,])
coef(svi.lm)
```

```
## (Intercept)      lcavol
##    2.053552    0.651189
```

```
coef(nosvi.lm)
```

```
## (Intercept)      lcavol
##    1.539704    0.586396
```

```
plot(pros.df$lcavol, pros.df$lpsa,
       main="lpsa vs. lcavol", xlab="lcavol", ylab="lpsa")
points(pros.df[pros.df$svi==0,]$lcavol, pros.df[pros.df$svi==0,]$lpsa, col="red", pch=19)
abline(a =2.053552, b =0.651189, lty=1)
abline(a =1.539704, b =0.586396, lty=1, col="red")
legend("bottomright", legend=c("svi == 1", "svi == 0"), col=c("black", "red"), pch=c(21, 19))
```



From the regression lines, it appears that patients with svi tend to have a higher lpsa on average. The svi patients both have a larger intercept and slope, and this difference appears larger than the one for the different groups split up by lcp. We should expect this because for the first set of regressions, we ran a regression on the whole data and then a subset, but for these regressions, we ran regressions on each of the subsets.

Reading in, exploring wage data

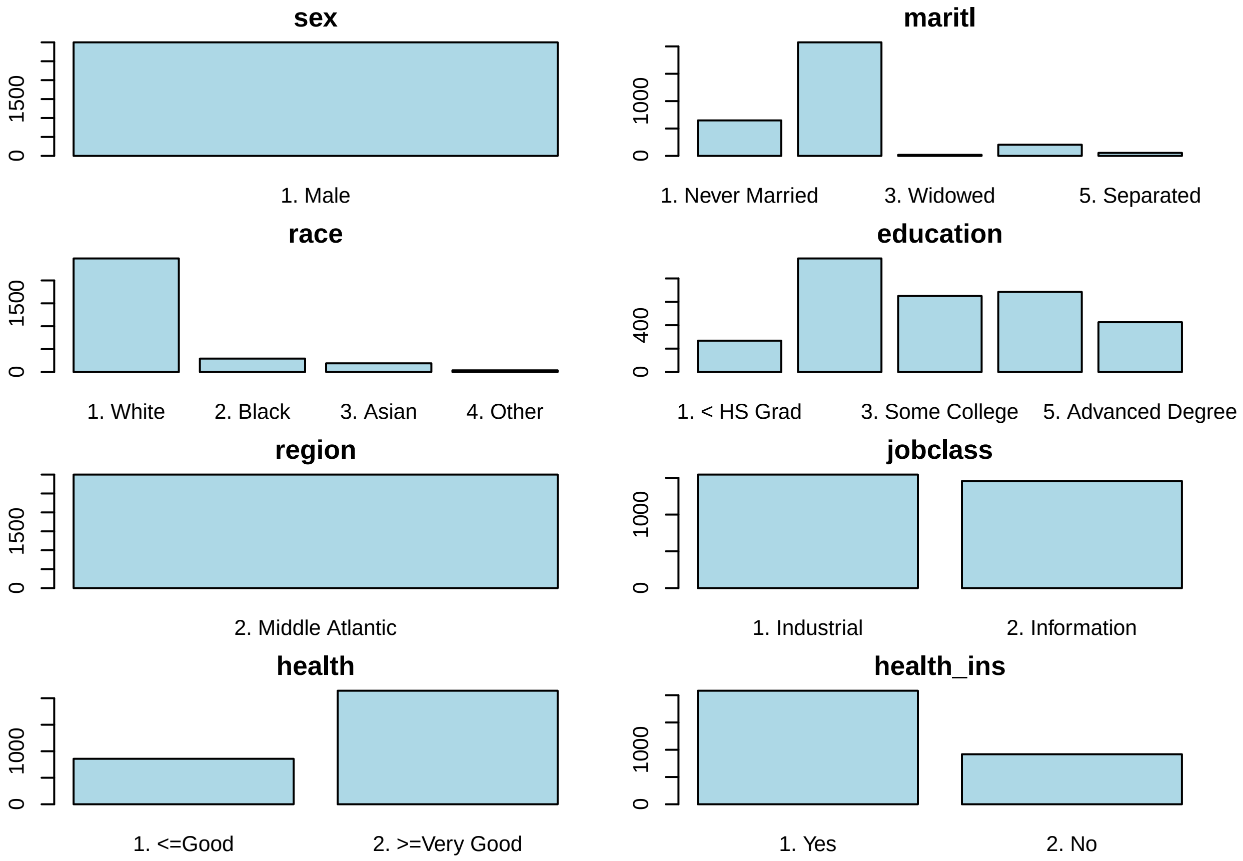
- **2a.** A data table of dimension 3000 x 11, containing demographic and economic variables measured on individuals living in the mid-Atlantic region, is up at http://www.stat.cmu.edu/~ryantibs/statcomp/data/wage.csv. (This has been adapted from the book An Introduction to Statistical Learning.) Load this data table into your R session with `read.csv()` and save the resulting data frame as `wage.df`. Check that `wage.df` has the right dimensions, and display its first 3 rows. Hint: the first several lines of the linked file just explain the nature of the data; open up the file (either directly in your web browser or after you download it to your computer), and count how many lines must be skipped before getting to the data; then use an appropriate setting for the `skip` argument to `read.csv()`.

```
wage.df <- read.csv("http://www.stat.cmu.edu/~ryantibs/statcomp/data/wage.csv",
                    skip=16, stringsAsFactors = TRUE)
head(wage.df)
```

```
##        year age     sex           maritl     race       education
## 231655 2006  18 1. Male 1. Never Married 1. White    1. < HS Grad
## 86582  2004  24 1. Male 1. Never Married 1. White 4. College Grad
## 161300 2003  45 1. Male       2. Married 1. White 3. Some College
## 155159 2003  43 1. Male       2. Married 3. Asian 4. College Grad
## 11443  2005  50 1. Male      4. Divorced 1. White      2. HS Grad
## 376662 2008  54 1. Male       2. Married 1. White 4. College Grad
##                    region       jobclass         health health_ins      wage
## 231655 2. Middle Atlantic  1. Industrial      1. <=Good      2. No  75.04315
## 86582  2. Middle Atlantic 2. Information 2. >=Very Good      2. No  70.47602
## 161300 2. Middle Atlantic  1. Industrial      1. <=Good     1. Yes 130.98218
## 155159 2. Middle Atlantic 2. Information 2. >=Very Good     1. Yes 154.68529
## 11443  2. Middle Atlantic 2. Information      1. <=Good     1. Yes  75.04315
## 376662 2. Middle Atlantic 2. Information 2. >=Very Good     1. Yes 127.11574
```

- **2b.** Identify all of the factor variables in `wage.df`, set up a plotting grid of appropriate dimensions, and then plot each of these factor variables, with appropriate titles. What do you notice about the distributions?

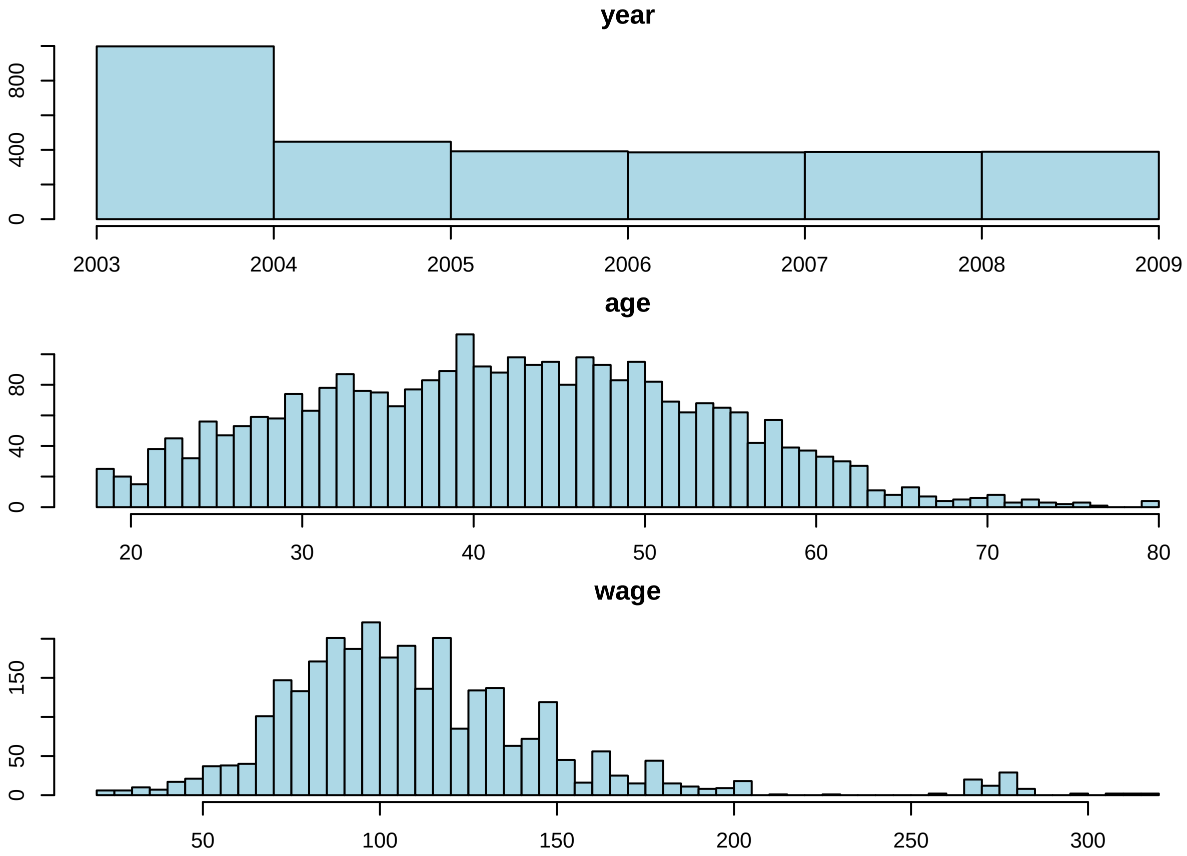
```r
vars <- colnames(wage.df)[-c(1,2,11)]
par(mfrow=c(4,2), mar=c(2,2,2,2))
for (j in 1:length(vars)) {
  plot(wage.df[,vars[j]],
       main= vars[j],
       col="lightblue")
}
```



The sample appears to be entirely men from the mid-Atlantic region.

- **2c.** Identify all of the numeric variables in `wage.df`, set up a plotting grid of appropriate dimensions, and then plot histograms of each these numeric variables, with appropriate titles and x-axis labels. What do you notice about the distributions? In particular, what do you notice about the distribution of the `wage` column? Does it appear to be unimodal (having a single mode)? Does what you see make sense?

```
vars <- c("year", "age", "wage")
breaks <- c(5, 50, 50)
par(mfrow=c(3,1), mar=c(2,2,2,2))
for (j in 1:length(vars)) {
  hist(wage.df[,vars[j]],
       main= vars[j],
       col="lightblue",
       breaks=breaks[j])
}
```



Obviously it will depend on how many breaks you use for the histograms, but wage appears to be unimodal.

Wage linear regression modeling

- **3a.** Fit a linear regression model, using `lm()`, with response variable `wage` and predictor variables `year` and `age`, using the `wage.df` data frame. Call the result `wage.lm`. Display the coefficient estimates, using `coef()`, for `year` and `age`. Do they have the signs you would expect, i.e., can you explain their signs? Display a summary, using `summary()`, of this linear model. Report the standard errors and p-values associated with the coefficient estimates for `year` and `age`. Do both of these predictors appear to be significant, based on their p-values?

```
wage.lm <- lm(wage ~ year + age, data=wage.df)
coef(wage.lm)
```

```
##   (Intercept)          year           age 
## -2318.5309186     1.1968236     0.6992032
```

```
summary(wage.lm)
```

```
## 
## Call:
## lm(formula = wage ~ year + age, data = wage.df)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -96.766 -25.081  -6.108  16.838 209.053 
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -2318.5309   739.1385  -3.137  0.00172 ** 
## year            1.1968     0.3685   3.247  0.00118 ** 
## age             0.6992     0.0647  10.808  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 40.86 on 2997 degrees of freedom
## Multiple R-squared:  0.04165,    Adjusted R-squared:  0.04101 
## F-statistic: 65.12 on 2 and 2997 DF,  p-value: < 2.2e-16
```

Both age and wage have positive signs, which makes sense because age is correlated with experience which correlates with wage, and year is correlated with inflation which is also correlated with wage. Both of these predictors have low p-values, so I would conclude that they're statistically significant.

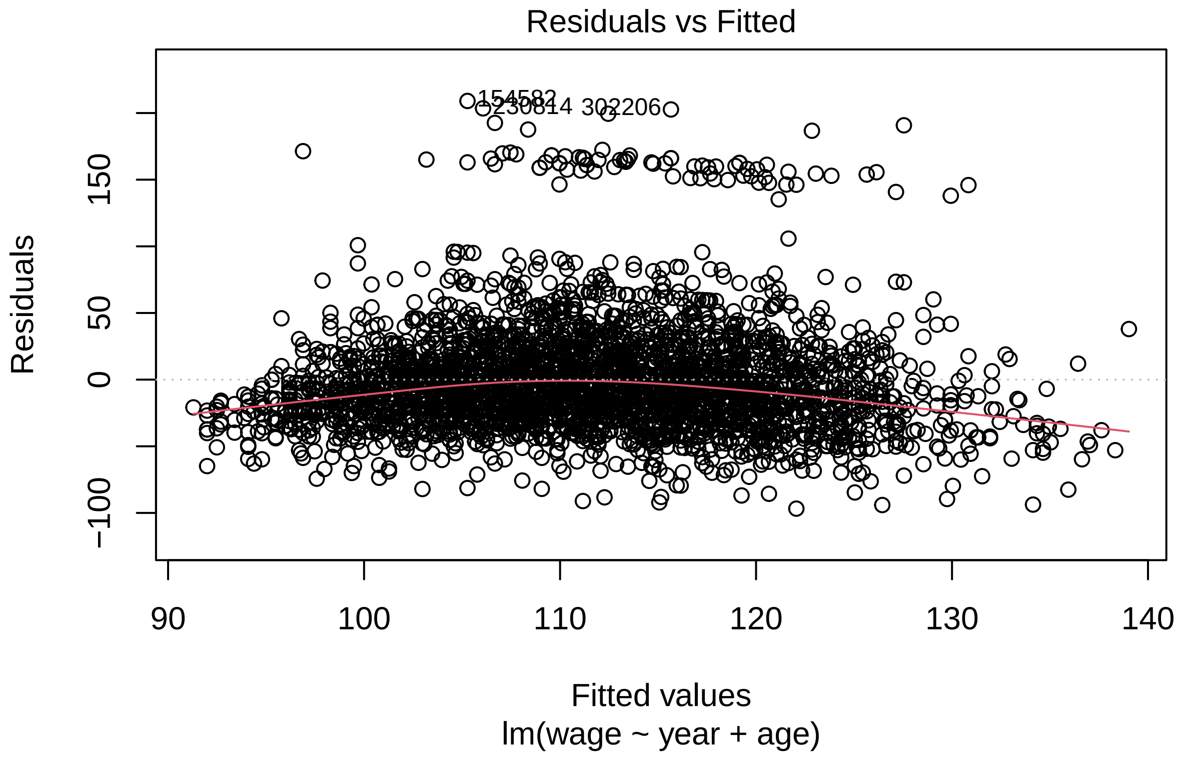
- **3b.** Save the standard errors of `year` and `age` into a vector called `wage.se`, and print it out to the console. Don't just type the values in you see from `summary()`; you need to determine these values programmatically. Hint: define `wage.sum` to be the result of calling `summary()` on `wage.lm`; then figure out what kind of R object `wage.sum` is, and how you can extract the standard errors.

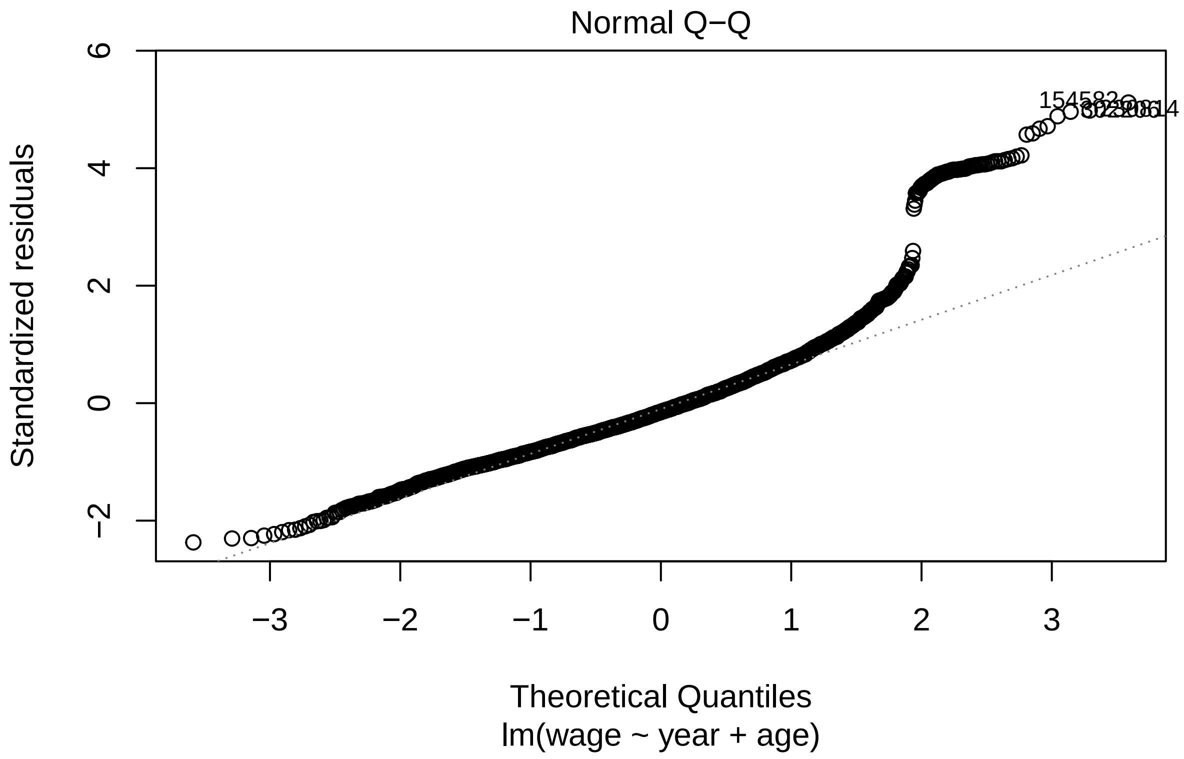
```
wage.sum <- summary(wage.lm)
wage.sum$coefficients[,"Std. Error"]
```

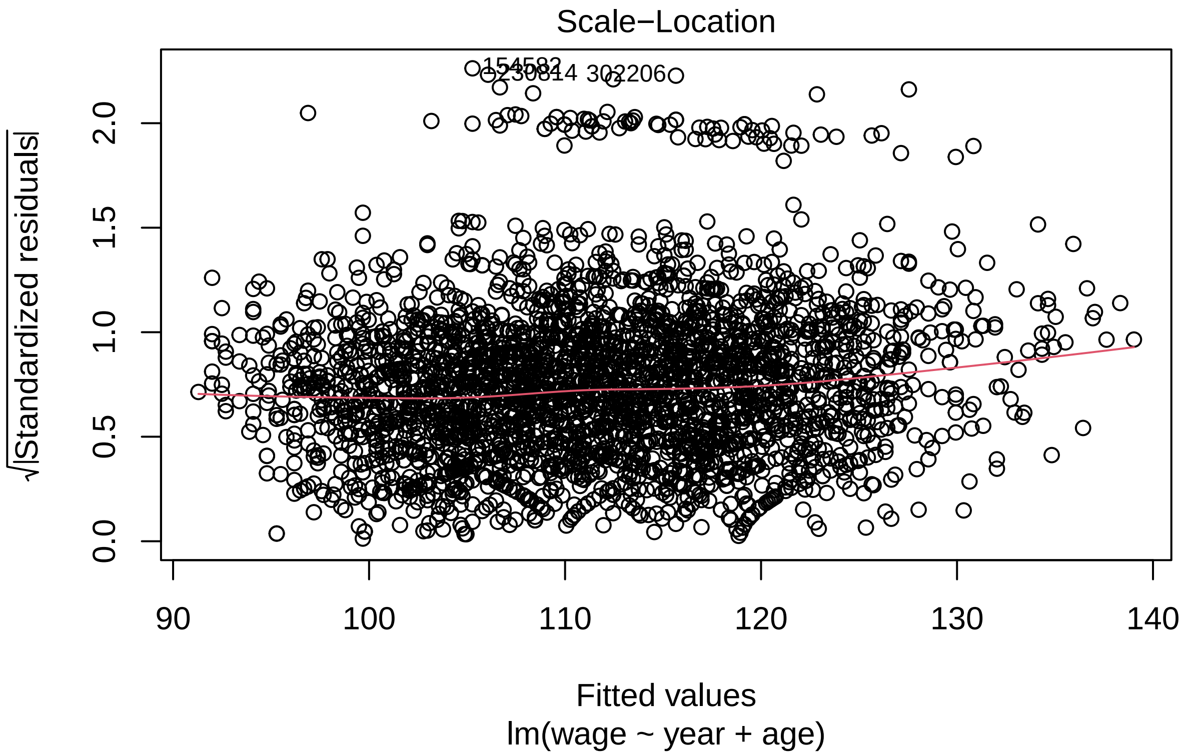
```
##  (Intercept)         year          age 
## 739.13853034   0.36855211   0.06469607
```

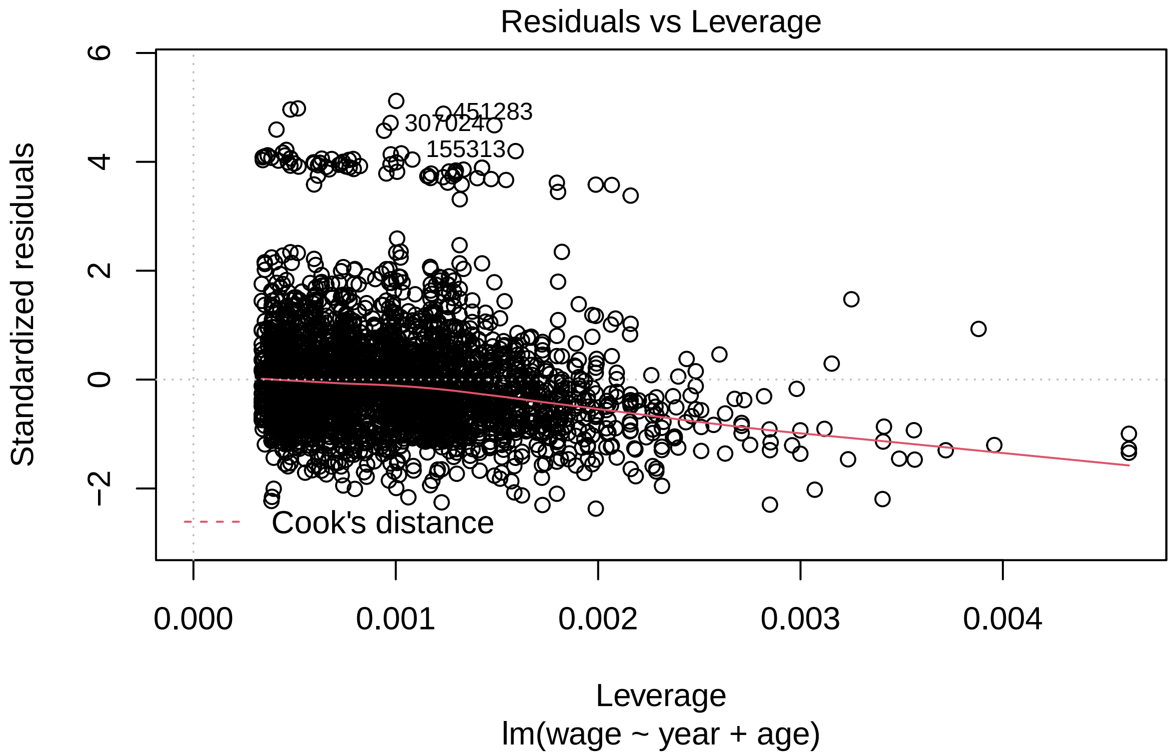
- **3c.** Plot diagnostics of the linear model fit in the previous question, using `plot()` on `wage.lm`. Look at the "Residuals vs Fitted", "Scale-Location", and "Residuals vs Leverage" plots—are there any groups of points away from the main bulk of points along the x-axis? Look at the "Normal Q-Q" plot—do the standardized residuals lie along the line $y = x$? Note: don't worry too if you're generally unsure how to interpret these diagnostic plots; you'll learn a lot more in your Modern Regression 36-401 course; for now, you can just answer the questions we asked. **Challenge**: what is causing the discrepancies you are (should be) seeing in these plots? Hint: look back at the histogram of the `wage` column you plotted above.

```
plot(wage.lm)
```


Residuals vs Fitted
Residuals
150
50
0
−100
90
100
110
120
130
140
154582
230814
302206
Fitted values
lm(wage ~ year + age)


Normal Q−Q
Standardized residuals
6
4
2
0
−2
−3
−2
−1
0
1
2
3
154582
302206
230814
Theoretical Quantiles
lm(wage ~ year + age)


Scale−Location
154582
230814
302206
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
90
100
110
120
130
140
Fitted values
lm(wage ~ year + age)



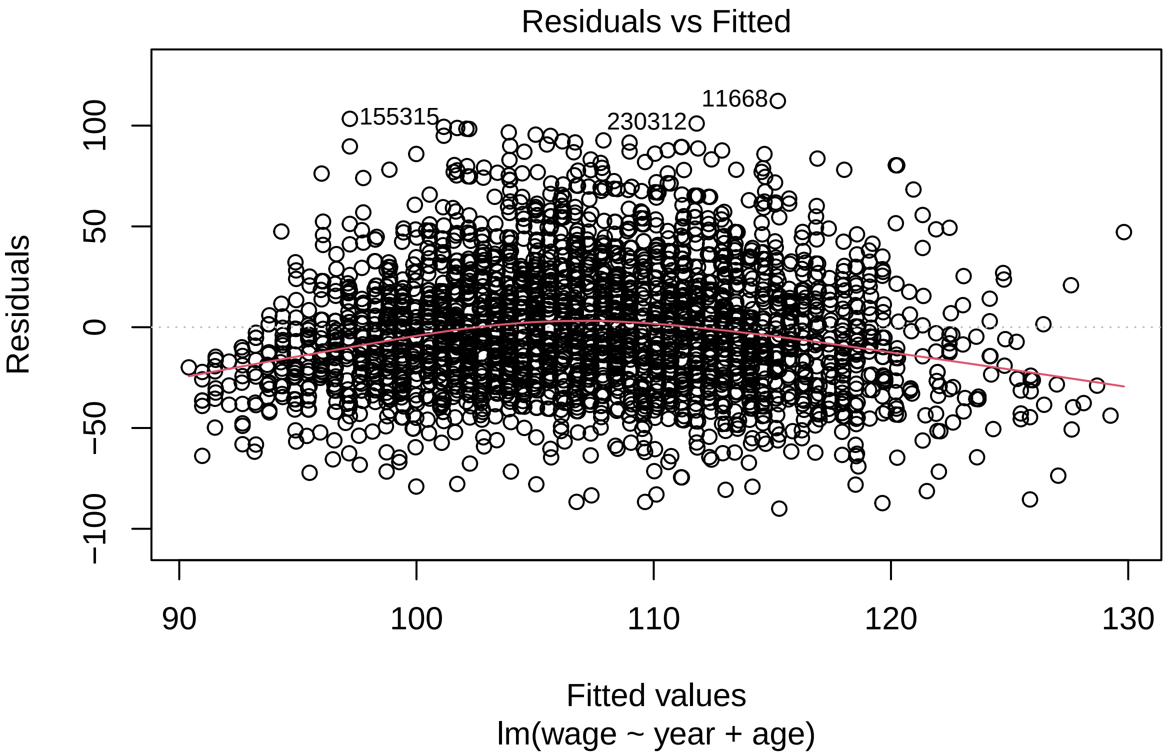
The standard errors look like they're normally distributed apart from a relatively small number of outliers. This group corresponds to people who have abnormally large wages, and age/year don't necessarily correlate well with those observations.

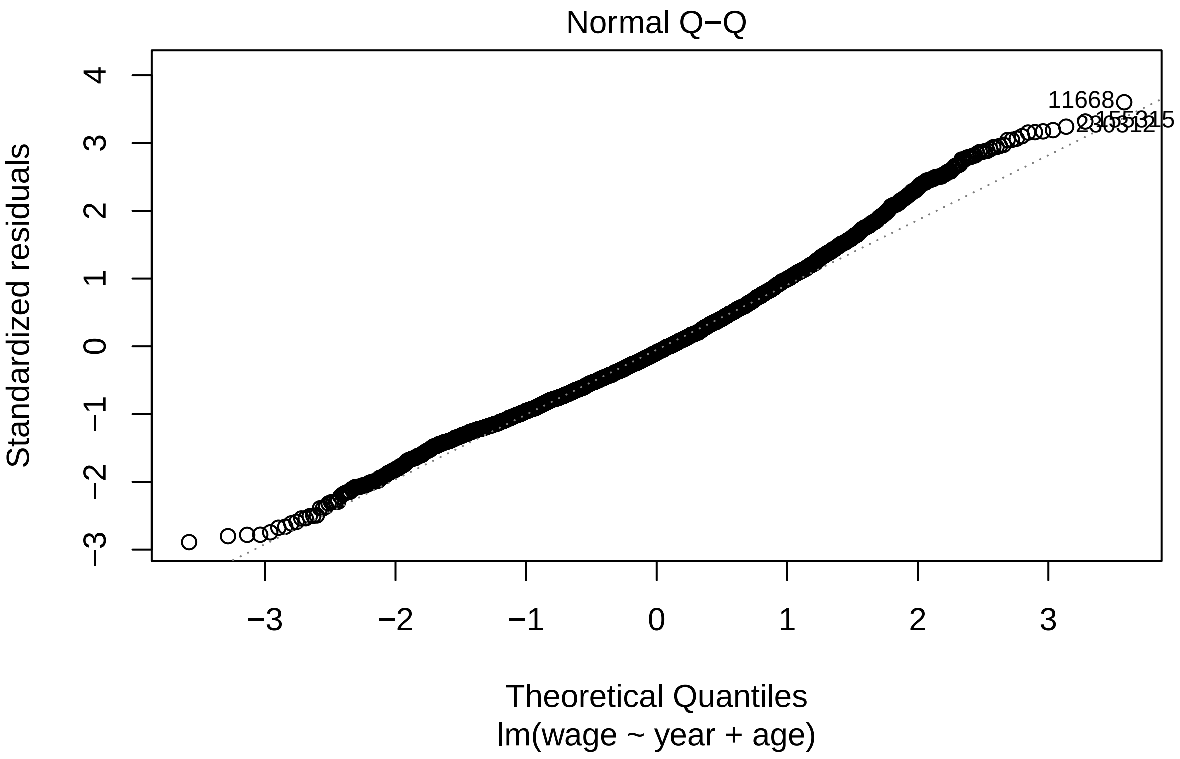
- **3d.** Refit a linear regression model with response variable `wage` and predictor variables `year` and `age`, but this time only using observations in the `wage.df` data frame for which the `wage` variable is less than or equal to 250 (note, this is measured in thousands of dollars!). Call the result `wage.lm.lt250`. Display a summary, reporting the coefficient estimates of `year` and `age`, their standard errors, and associated p-values. Are these coefficients different than before? Are the predictors `year` and `age` still significant? Finally, plot diagnostics. Do the "Residuals vs Fitted", "Normal Q-Q", "Scale-location", and "Residuals vs Leverage" plots still have the same problems as before?

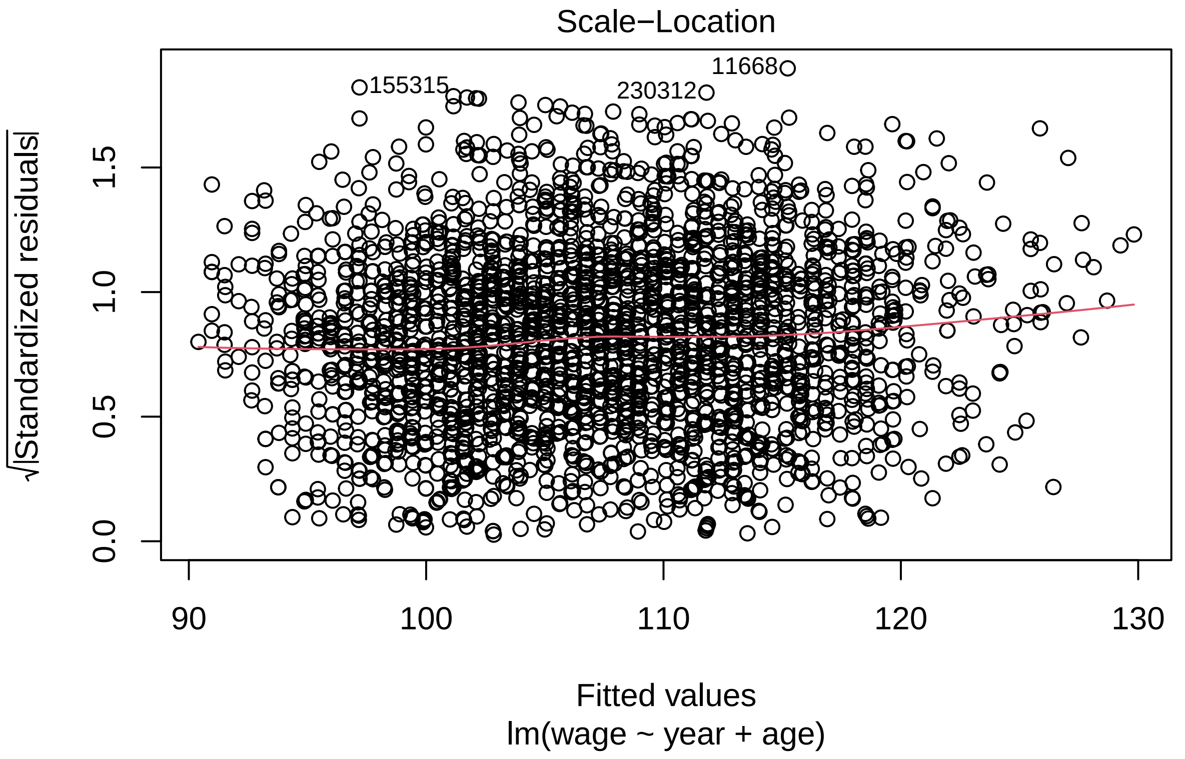
```
wage.lm.lt250 <- lm(wage ~ year + age, data = wage.df[wage.df$wage<250,])
summary(wage.lm.lt250)
```

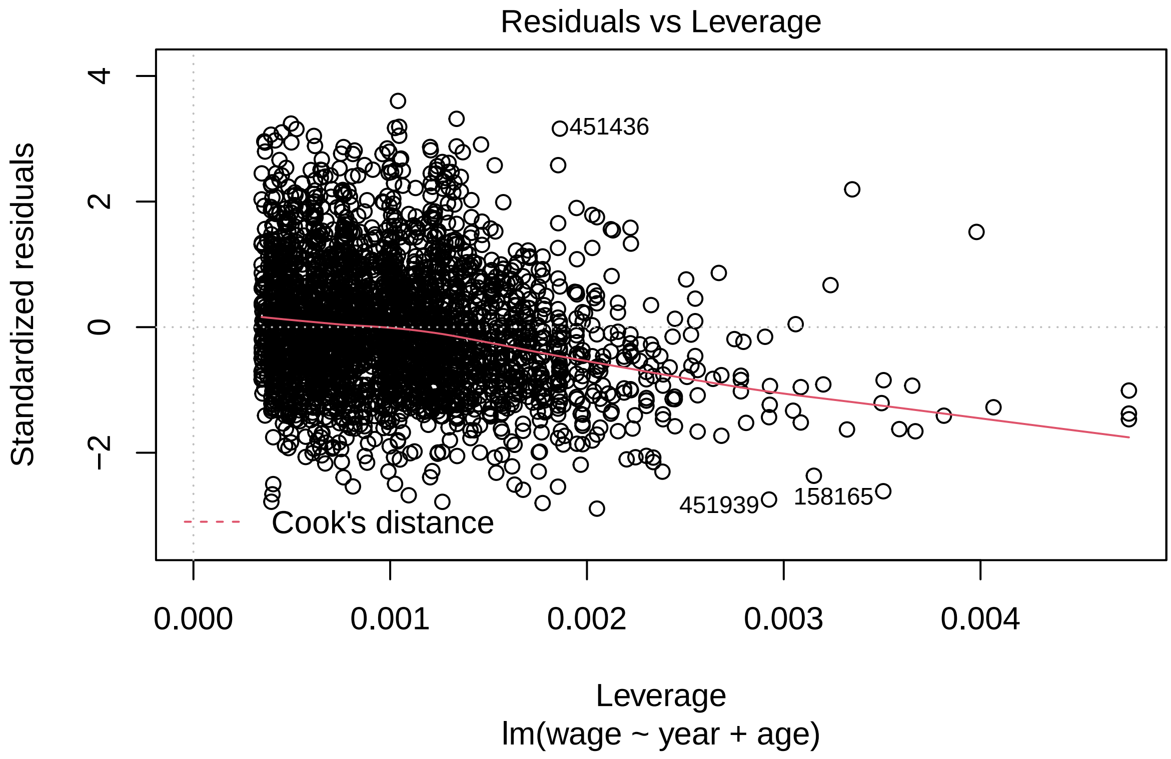
```
##
## Call:
## lm(formula = wage ~ year + age, data = wage.df[wage.df$wage <
##     250, ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -90.001 -21.690  -2.905  18.518 112.226
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.126e+03  5.715e+02  -3.721 0.000203 ***
## year         1.102e+00  2.850e-01   3.866 0.000113 ***
## age          5.656e-01  4.986e-02  11.345  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 31.17 on 2918 degrees of freedom
## Multiple R-squared:  0.04812,    Adjusted R-squared:  0.04747
## F-statistic: 73.75 on 2 and 2918 DF,  p-value: < 2.2e-16
```

```
plot(wage.lm.lt250)
```


Residuals vs Fitted
Residuals
100
50
0
−50
−100
90
100
110
120
130
Fitted values
lm(wage ~ year + age)
155315
230312
11668


Normal Q−Q
Standardized residuals
4
3
2
1
0
−1
−2
−3
−3
−2
−1
0
1
2
3
Theoretical Quantiles
lm(wage ~ year + age)
11668
155315
230312


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
90
100
110
120
130
Fitted values
lm(wage ~ year + age)
155315
230312
11668



The coefficients from age and year are slightly smaller than the ones from before, but they have smaller p-values; this indicates that there's a stronger statistical relationship between wage and these variables in the second model. This likely happened because the large wage outliers from before occurred in later years and among older individuals. This time around, there aren't many points outside of the main groupings in the plots, and the q-q plot is much closer to being linear.

- **3e.** Use your fitted linear model `wage.lm.lt250` to predict: (a) what a 30 year old person should be making this year; (b) what President Trump should be making this year; (c) what you should be making 5 years from now. Comment on the results—which do you think is the most accurate prediction?

```
wage.new = data.frame(age=c(30, 75, 31), year=c(2021, 2021, 2026))
wage.pred = predict(wage.lm.lt250, newdata=wage.new)
wage.pred
```

```
##        1        2        3
## 117.0212 142.4743 123.0949
```

Of the three results, the first prediction is probably the most accurate. The prediction for me includes both my increase in age and the increase in time, so it has two potential sources of error. Donald Trump is an exceptional person, and he probably belongs to the category of people that we excluded when dropping wages over $250, so his prediction probably isn't accurate either. Therefore, the first prediction is probably the best.

Wage logistic regression modeling

- **4a.** Fit a logistic regression model, using `glm()` with `family="binomial"`, with the response variable being the indicator that `wage` is larger than 250, and the predictor variables being `year` and `age`. Call the result wage.glm. Note: you can set this up in two different ways: (i) you can manually define a new column (say) wage.high in the wage.df data frame to be the indicator that the wage column is larger than 250; or (ii) you can define an indicator variable "on-the-fly" in the call to glm() with an appropriate usage of I(). Display a summary, reporting the coefficient estimates for year and age, their standard errors, and associated p-values. Are the predictors year and age both significant?

```
wage.df$wage.high <- wage.df$wage>250
wage.glm <- glm(wage.high ~ age + year, family="binomial", data=wage.df)
summary(wage.glm)
```

```
##
## Call:
## glm(formula = wage.high ~ age + year, family = "binomial", data = wage.df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.4039  -0.2542  -0.2198  -0.1899   2.9302
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -48.808462 112.375762  -0.434 0.664047
## age           0.032779   0.009817   3.339 0.000841 ***
## year          0.021806   0.056033   0.389 0.697155
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 730.53  on 2999  degrees of freedom
## Residual deviance: 719.08  on 2997  degrees of freedom
## AIC: 725.08
##
## Number of Fisher Scoring iterations: 6
```

Only age is a significant predictor of having high wage in this model.

- **4b.** Refit a logistic regression model with the same response variable as in the last question, but now with predictors year, age, and education. Note that the third predictor is stored as a factor variable, which we call a **categorical variable** (rather than a continuous variable, like the first two predictors) in the context of regression modeling. Display a summary. What do you notice about the predictor education: how many coefficients are associated with it in the end? **Challenge**: can you explain why the number of coefficients associated with education makes sense?

```
wage.glm <- glm(wage.high ~ age + year + education, family="binomial", data=wage.df)
summary(wage.glm)
```

```
##
## Call:
## glm(formula = wage.high ~ age + year + education, family = "binomial",
##     data = wage.df)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6917  -0.2450  -0.1325  -0.0954   3.3037
##
## Coefficients:
```

```
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -39.113218 662.234193  -0.059   0.9529
## age                           0.026768   0.010791   2.481   0.0131 *
## year                          0.009165   0.057277   0.160   0.8729
## education2. HS Grad          14.283263 652.196203   0.022   0.9825
## education3. Some College     15.068105 652.196160   0.023   0.9816
## education4. College Grad     16.137982 652.196085   0.025   0.9803
## education5. Advanced Degree  17.356990 652.196068   0.027   0.9788
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 730.53  on 2999  degrees of freedom
## Residual deviance: 615.76  on 2993  degrees of freedom
## AIC: 629.76
##
## Number of Fisher Scoring iterations: 18
```

There are four coefficients associated with education in the regression summary. This is because it's a factor variable with 5 possible values, so the value of having <HS education is coded into the intercept.

- **4c.** In general, one must be careful fitting a logistic regression model on categorial predictors. In order for logistic regression to make sense, for each level of the categorical predictor, we should have observations at this level for which the response is 0, and observations at this level for which the response is 1. In the context of our problem, this means that for each level of the `education` variable, we should have people at this education level that have a wage less than or equal to 250, and also people at this education level that have a wage above 250. Which levels of `education` fail to meet this criterion? Let's call these levels "incomplete", and the other levels "complete".

```
table(wage.df[wage.df$wage.high==0, "education"])
```

```
##
##       1. < HS Grad         2. HS Grad    3. Some College    4. College Grad
##                268                966                643                663
## 5. Advanced Degree
##                381
```

```
table(wage.df[wage.df$wage.high==1, "education"])
```

```
##
##       1. < HS Grad         2. HS Grad    3. Some College    4. College Grad
##                  0                  5                  7                 22
## 5. Advanced Degree
##                 45
```

Each education level apart from "< HS Grad" has people in both the low and high wage categories.

- **4d.** Refit the logistic regression model as in Q4b, with the same response and predictors, but now throwing out all data in `wage.df` that corresponds to the incomplete education levels (equivalently, using only the data from the complete education levels). Display a summary, and comment on the differences seen to the summary for the logistic regression model fitted in Q4b. Did any predictors become more significant, according to their p-values?

```
wage.glm <- glm(wage.high ~ age + year + education, family="binomial",
                data=wage.df[wage.df$education!="1. < HS Grad",])
summary(wage.glm)
```

```
##
## Call:
## glm(formula = wage.high ~ age + year + education, family = "binomial",
##     data = wage.df[wage.df$education != "1. < HS Grad", ])
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6917  -0.2547  -0.1435  -0.1055   3.3037
##
## Coefficients:
##                               Estimate Std. Error z value Pr(>|z|)
## (Intercept)                 -24.829955 114.867312  -0.216 0.828861
## age                           0.026768   0.010791   2.481 0.013118 *
## year                          0.009165   0.057277   0.160 0.872866
## education3. Some College      0.784841   0.588306   1.334 0.182181
## education4. College Grad      1.854719   0.498254   3.722 0.000197 ***
## education5. Advanced Degree   3.073726   0.475777   6.460 1.04e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 715.54  on 2731  degrees of freedom
## Residual deviance: 615.76  on 2726  degrees of freedom
## AIC: 627.76
##
## Number of Fisher Scoring iterations: 8
```

The college graduate and advanced degree coefficients are now significant.

Wage generalized additive modeling (optional)

- **5a.** Install the gam package, if you haven't already, and load it into your R session with library(gam). Fit a generalized additive model, using gam() with family="binomial", with the response variable being the indicator that wage is larger than 250, and the predictor variables being year, age, and education; as in the last question, only use observations in wage.df corresponding to the complete education levels. Also, in the call to gam(), allow for age to have a nonlinear effect by using s() (leave year and education alone, and they will have the default—linear effects). Call the result wage.gam. Display a summary with summary(). Is the age variable more or less significant, in terms of its p-value, to what you saw in the logistic regression model fitted in the last question? Also, plot the fitted effect for each predictor, using plot(). Comment on the plots—does the fitted effect make sense to you? In particular, is there a strong nonlinearity associated with the effect of age, and does this make sense?

```
library(gam)
```

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

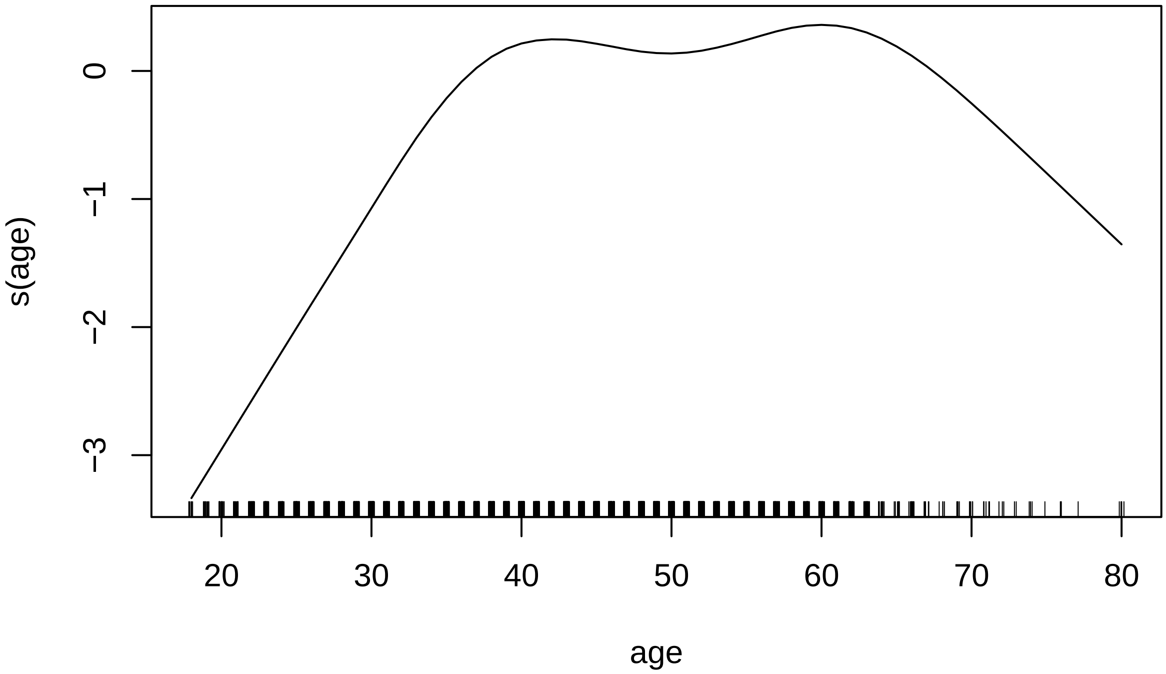
```
## Loading required package: foreach
```

```
## Loaded gam 1.20
```

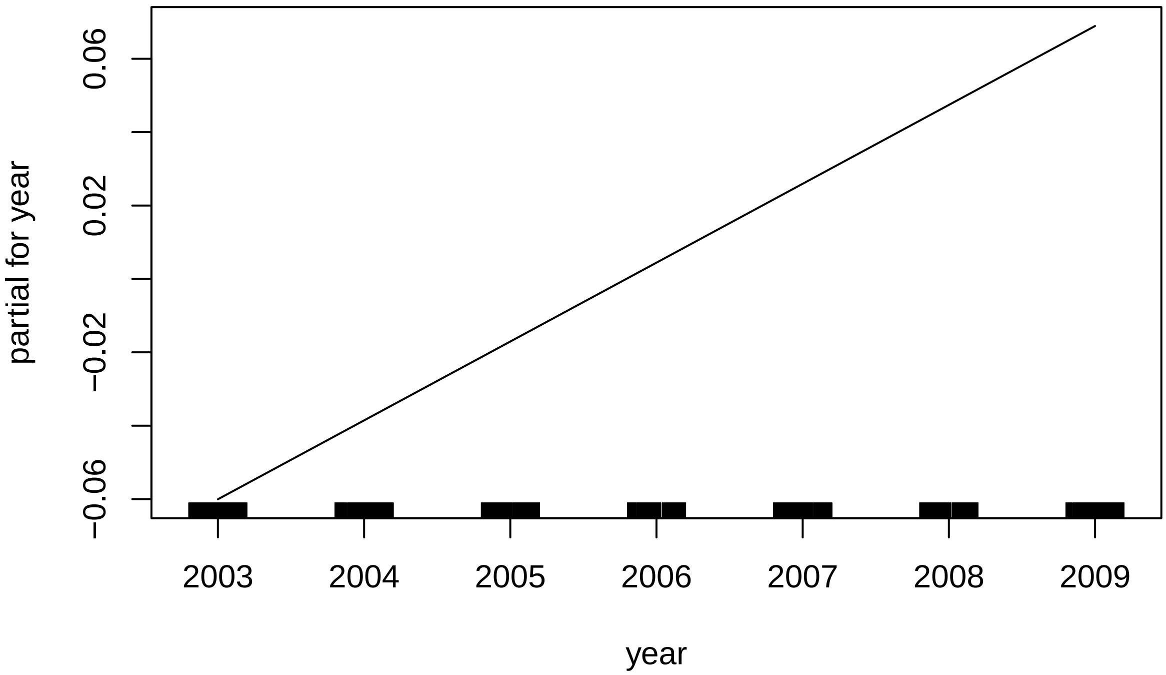
```
wage.gam <- gam(wage.high ~ s(age) + year + education, family = "binomial",
                data=wage.df[wage.df$education!="1. < HS Grad",])
summary(wage.gam)
```

```
##
## Call: gam(formula = wage.high ~ s(age) + year + education, family = "binomial",
##     data = wage.df[wage.df$education != "1. < HS Grad", ])
## Deviance Residuals:
##     Min      1Q  Median      3Q     Max
## -0.5625 -0.2786 -0.1440 -0.1099  3.2430
##
## (Dispersion Parameter for binomial family taken to be 1)
##
##     Null Deviance: 715.5412 on 2731 degrees of freedom
## Residual Deviance: 603.7774 on 2723 degrees of freedom
## AIC: 621.7775
##
## Number of Local Scoring Iterations: NA
##
## Anova for Parametric Effects
##             Df  Sum Sq Mean Sq F value    Pr(>F)
## s(age)       1    4.33  4.3310  4.8823   0.02722 *
## year         1    0.33  0.3300  0.3720   0.54198
## education    3   66.31 22.1046 24.9184 6.678e-16 ***
## Residuals 2723 2415.52  0.8871
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Anova for Nonparametric Effects
##             Npar Df Npar Chisq  P(Chi)
## (Intercept)
## s(age)            3     10.001 0.01856 *
## year
## education
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

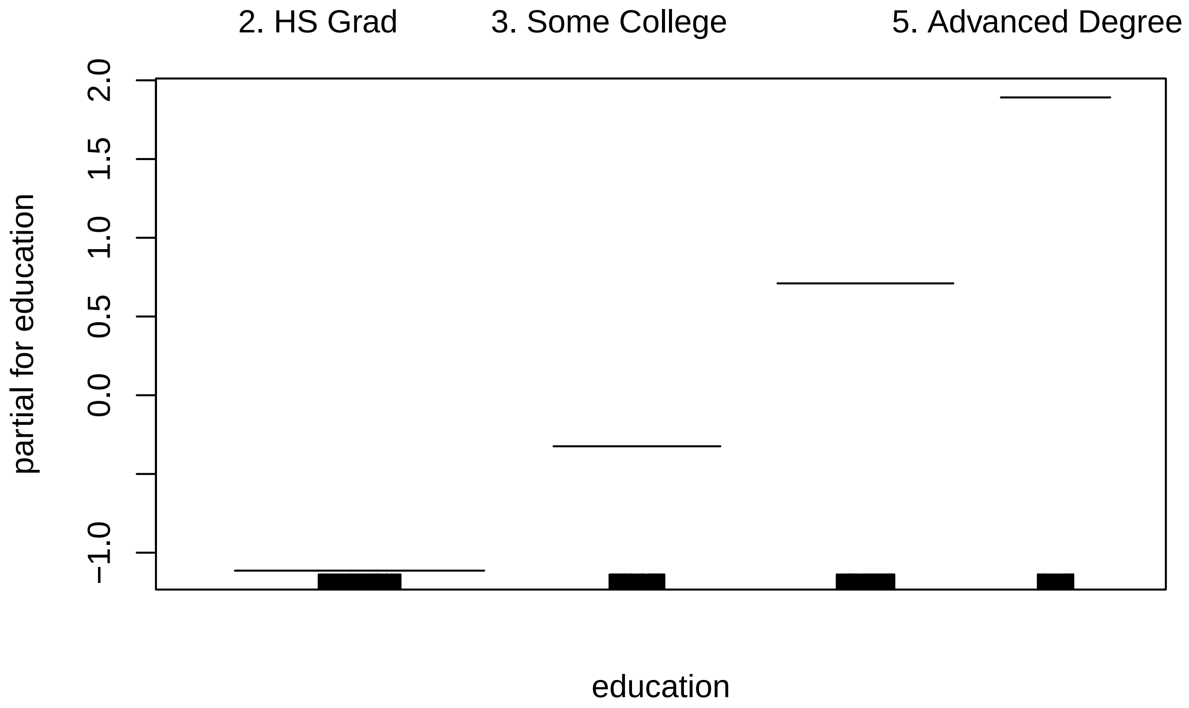
```
plot(wage.gam)
```



s(age)
0
−1
−2
−3
20
30
40
50
60
70
80
age



partial for year
0.06
0.02
−0.02
−0.06
2003
2004
2005
2006
2007
2008
2009
year



The age variable is less significant in this model. We observe fairly linear effects of age and year on the likelihood of wage.high, but age is a downward facing parabola. This makes sense because age will increase experience and therefore wage, but people will eventually start to retire, which will pull the likelihood of having high wage back down.

- **5b.** Using `wage.gam`, predict the probability that a 30 year old person, who earned a Ph.D., will make over $250,000 in 2018.

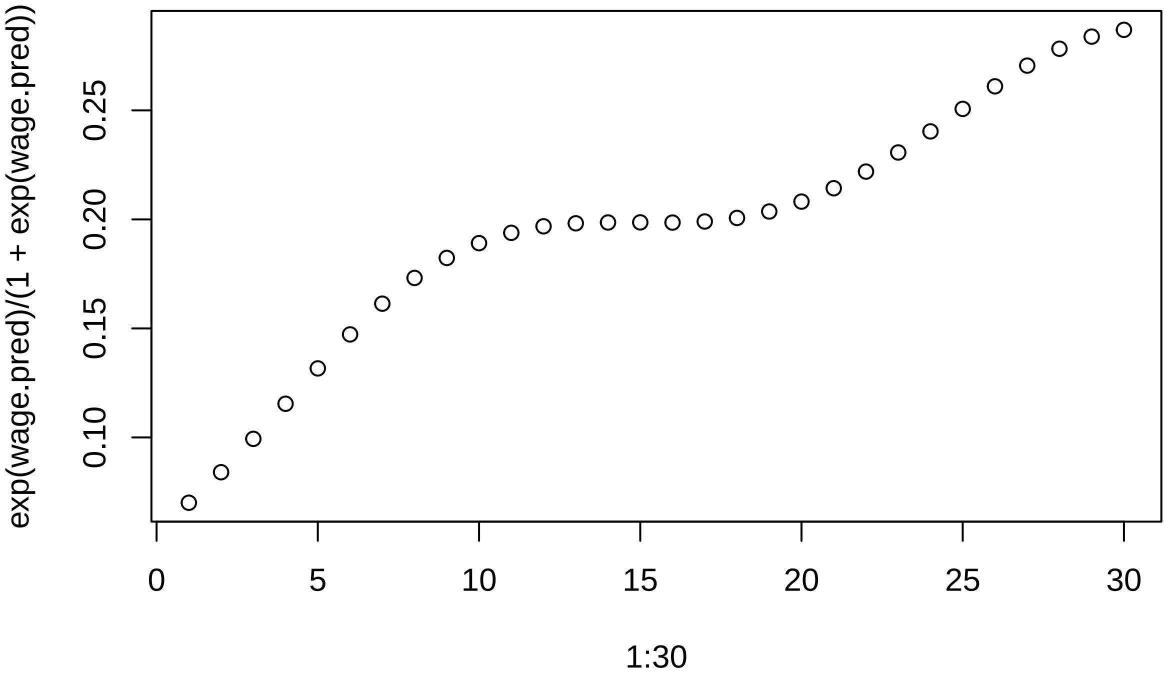
```
wage.new = data.frame(age=30, year=2018, education="5. Advanced Degree")
wage.pred = predict(wage.gam, newdata=wage.new)
exp(wage.pred) / (1+exp(wage.pred))
```

```
##          1
## 0.04737011
```

There's about a 4.7% that a person fitting this description will have high wage in 2018.

- **5c.** For a 32 year old person who earned a Ph.D., how long does he/she have to wait until there is a predicted probability of at least 20% that he/she makes over $250,000 in that year? Plot his/her probability of earning at least $250,000 over the future years—is this strictly increasing?

```
wage.new = data.frame(age=32:61, year=2020:2049,
                      education=rep(c("5. Advanced Degree"), times=30))
wage.pred = predict(wage.gam, newdata=wage.new)
plot(1:30, exp(wage.pred)/(1+exp(wage.pred)))
```



This person has to wait about 20 years until his probability of earing a high wage is over 20%. This probability is not strictly increasing and resembles an s-curve.