The purpose of this section is to estimate the returns to education using `R`. There is nothing valid about the results found in this section; but the empirical application gives us a chance to explore categorical dummies and the `ggplot` package. First, as always, we load the required libraries.

```
library(foreign)
library(ggplot2)
library(xtable)
```

We can then read the wage data directly from the online repository for the supplementary data sets for the Wooldridge (2002) text. You will need an internet connection. We only need the `wage`, `educ`, and `age` variables, and we omit all observations with missing observations.

```
f <- "http://fmwww.bc.edu/ec-p/data/wooldridge/wage2.dta"
data <- read.dta(f)
data <- data[ , c("wage", "educ", "age")]
data <- na.omit(data)
```

A quick visualization reveals the distribution of wages in the data set:

```
hist(data$wage, xlab = "wage", main = "", col = "grey", border = "white")
```

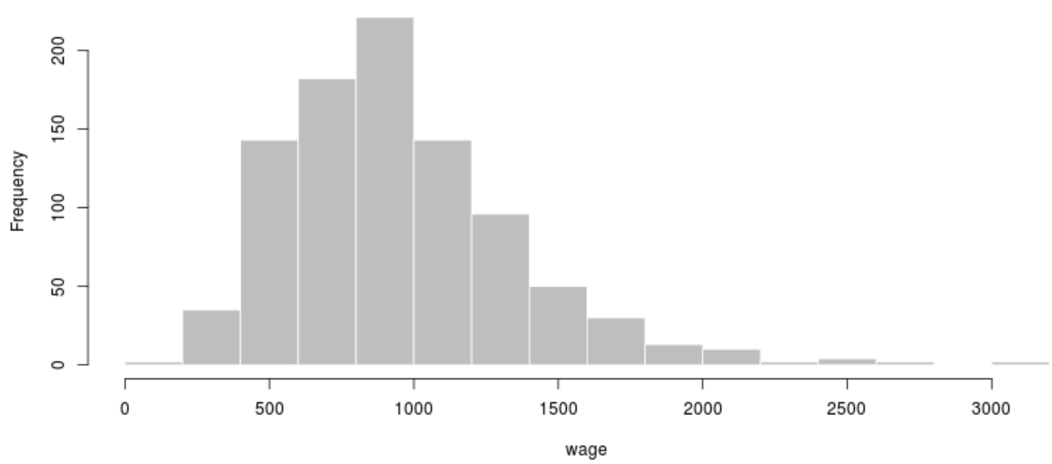

Figure 1: Wage histogram

Roughly following page 38 of the lecture notes, we create a rough measure of educational attainment from the `educ` variable.

```
e1 <- ifelse(data$educ %in% 1:12, 1, 0)
e2 <- ifelse(data$educ %in% 13:14, 1, 0)
e3 <- ifelse(data$educ %in% 15:16, 1, 0)
e4 <- ifelse(data$educ %in% 17:18, 1, 0)
```

The categorical education variables sum to one, and the `lm()` function will force-drop one of the variables. Note that the intercept in this regression reflects the mean wage of the `e4` class. The other coefficients reflect the relative wages of the other three classes.

```
coef(summary(m1 <- lm(wage ~ 1 + e1 + e2 + e3 + e4, data = data)))
```

```
              Estimate Std. Error   t value      Pr(>|t|)
(Intercept) 1196.95876   38.93314 30.743953 7.969254e-144
e1          -350.46396   42.67867 -8.211689  7.239709e-16
e2          -229.97111   49.22796 -4.671555  3.429280e-06
e3           -90.50748   47.64240 -1.899726  5.777793e-02
```

Suppose we want to estimate the premium on education, relative to the least educated class. We can then specify the following regression and print the output directly to LaTeX using the `xtable` package[1]:

```
xtable(m2 <- lm(wage ~ 1 + e2 + e3 + e4, data = data))
```

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	846.4948	17.4837	48.42	0.0000
e2	120.4929	34.8322	3.46	0.0006
e3	259.9565	32.5528	7.99	0.0000
e4	350.4640	42.6787	8.21	0.0000

None that the coefficients, now, indicate the premium over the base level of education for all subsequent levels. Consider, for example, the premium on the `e4` class. The average wage for people in this class, the class with the highest educational attainment levels, is found by:

```
mean(data[e4 == 1, c("wage")])
```

```
[1] 1196.959
```

This is equivalent to adding the coefficient on `e4` to the intercept from the well-specified regression above. Specifically:

```
b <- m2$coefficients
b[["(Intercept)"]] + b[["e4"]]
```

```
[1] 1196.959
```

This equality only holds because there are no other covariates in the regression. If we condition on age, for example, then the simple addition does not yield an average wage. For illustration, consider the previous regression with `age` and squared `age` as cofactors. Note also the manner by which the `lm()` function accepts a nested function to specify squared `age` within the line:

```
coef(summary(lm(wage ~ 1 + e2 + e3 + e4 + age + I(age^2), data = data)))
```

```
                Estimate  Std. Error     t value     Pr(>|t|)
(Intercept) -148.0041478 1660.163718 -0.08915033 9.289817e-01
e2           142.0919504   34.603321  4.10630965 4.374401e-05
e3           276.4462130   32.327306  8.55147690 4.954754e-17
e4           329.3717601   42.427654  7.76313861 2.181854e-14
age           38.4978013  100.467589  0.38318628 7.016693e-01
I(age^2)      -0.2572671    1.508597 -0.17053405 8.646273e-01
```

It looks like age has a positive but diminishing effect on wage. This makes sense, maybe, but the coefficients are not significantly different from zero. Why might this be the case? This is where some non-parametric graphing comes in handy.

```
(g <- ggplot(data, aes(x=age, y=wage)) + geom_smooth(method="loess", size=1.5))
```

We use the `ggplot2` package instead of the base `R` plotting facilities. The plots reveal a reasonable relationship between wage and age, but there is a significant amount of variation in wage, relative to the short time frame of age.

```
(g <- g + geom_point())
```

[1] Note that this is a little superfluous, but it's worth examining the different ways to export tables.

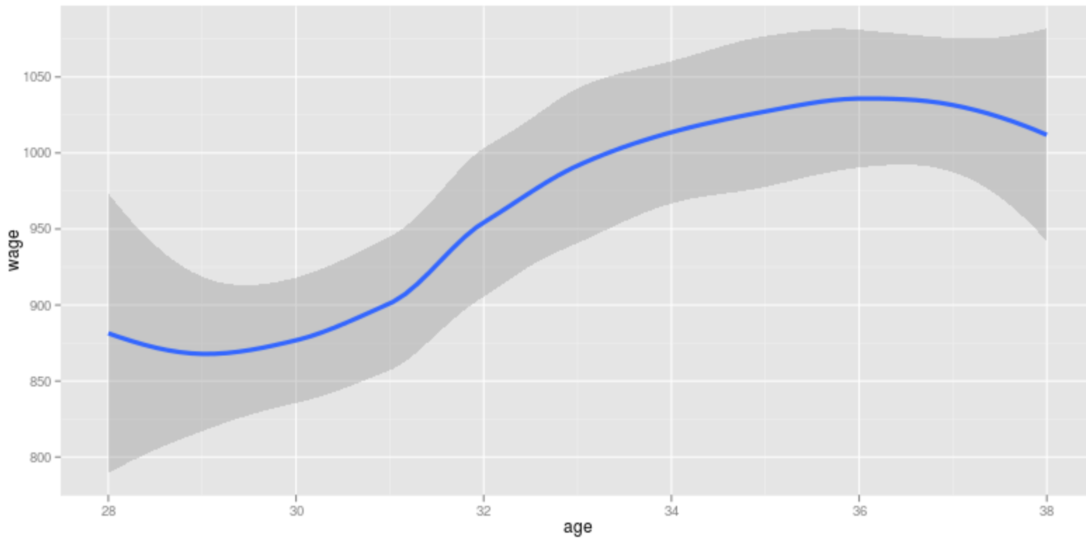

Figure 2: Smoothed line

Maximum Likelihood

Suppose that I tell you that $\mathbf{X}$ is a random vector, where each $\mathbf{X}_i$ is generated from a common density function $\theta/(\theta + \mathbf{X}_i)^2$. We want to estimate θ by maximum likelihood, given the data set `mle.txt` from Freedman (2009). It can be shown that the log-likelihood function is given by the following, where n is the number of observations:

$$\mathscr{L}_n(\theta) = n \log \theta - 2 \sum_{i=1}^{n} \log(\theta + \mathbf{X}_i)$$

First, we will plot the log-likelihood function as a function of θ, and then find the maximum with `optimize`.

```
data <- read.csv("mle.txt", header = FALSE)

logLik <- function(theta, X = data) {
  n <<- nrow(data)
  n * log(theta) - 2 * sum(log(theta + X))
}
```

To maximize this function with respect to θ, we don't have to do any math. And in fact, for this function, there is no explicit function for the maximum likelihood estimate, and we have to find the estimate through numerical optimization.

```
suppressWarnings(opt <- optimize(logLik, interval=c(-100, 100), maximum=TRUE))
(theta.hat <- opt$maximum)
```

```
 [1] 22.50976
```

We can compute the asymptotic variance in a variety of ways, but perhaps the most direct is $[-\mathscr{L}_n''(\hat{\theta})]^{-1}$:

```
dd.logLik <- function(theta, X = data) {
  -1 * (n / theta^2) + 2 * sum(1 / (theta + X)^2)
}

(asy.var <- -1 / dd.logLik(theta.hat))
```

```
 [1] 30.12326
```

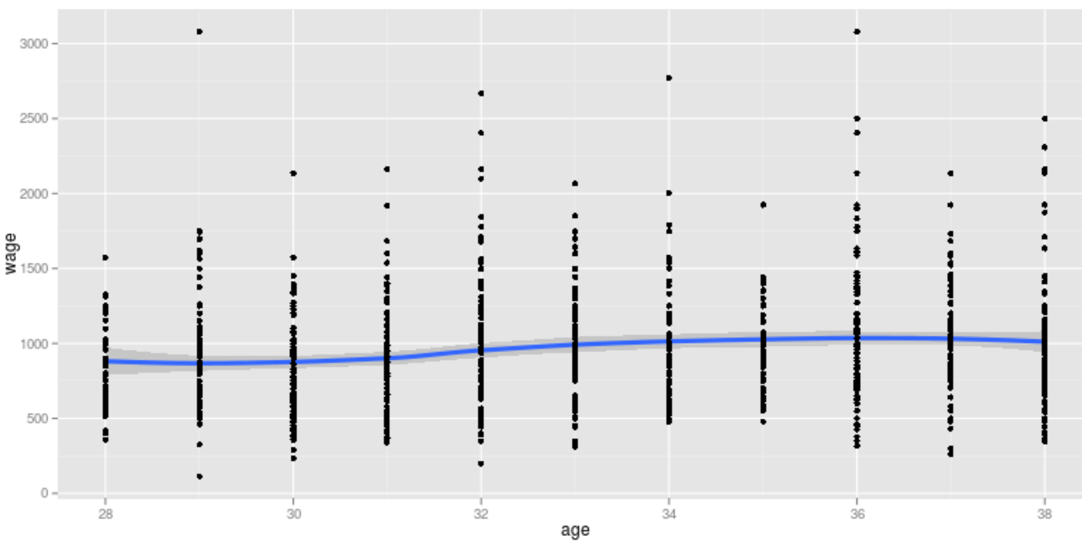

Figure 3: Smoothed line with points