SCRIPT `MOD1_2A`: A SIMULATED REGRESSION MODEL

Set basic R-options upfront and load all required R packages:

1. Model Building

1.1. **Variable definition.**

dependent variable: hours of studying for a VT undergraduate student per week last semester.

explanatory variables:

- *credits*: number of credits taken by student (assume between 6 and 15)
- *work*: weekly hours of work (assume between 1 and 20)
- *SAT*: total high school SAT score (between 600 and 2400)
- *upper*: an indicator variable taking a value of ”0” if student is a freshman or sophomore, ”1” if junior or senior.

1.2. **Create Explanatory Variables.**

credits:

Range from 6 to 15, but most students will take something in the 9-12 range; thus, when we assign random credit figures to students we want to have more values in the 9-12 range. We can acomplish this either by assigning a specific distribution to ”credits” (such as ”Poisson” or ”Negative Binomial”), or , simpler, pre-defining underlying population probabilities for each value and then use ”sample” to draw from the resulting weighted density.

```
R> n<-5000 #number of draws
R> pvec<-c(0.05, 0.05, 0.1, 0.15, 0.15, 0.15, 0.15, 0.1, 0.05, 0.05)
R> #  so "6" gets probability 0.05, same for "7", "8" gets 0.1, etc. These
R> # prob's add to 1 - check!
R> credits=sample(6:15,n,replace=TRUE,prob=pvec)
```

work:
Assume students are slightly more likely to work between 11 and 20 hours than between 1 and 10 hours. Draw a random sample and plot it.

```
R> pvec<-c(rep(0.04,10),rep(0.06,10))
R> #stack a set of 10 "0.04's" over a set of 10 "0.06"'s
R> work=sample(1:20,n,replace=TRUE,prob=pvec)
```

score:
Here we simply choose a normal distribution with mean 1500 and std 200, truncated to the 600, 2400 interval. Draw and create a histogram.

```
R> SAT<-rtnorm(n, mean=1500, sd=200, lower=600, upper=2400)
```

upper: let's assume there are about 40% of juniors and seniors, and 60% of freshmen and sophomores in the UNR population. We can use a Bernoulli (or basic Binomial) density to obtain the corresponding draws. Show in a barplot.

```
R> upper=rbinom(n,1,0.4);
```

Now plot all four graphs into a single "multi-figure":

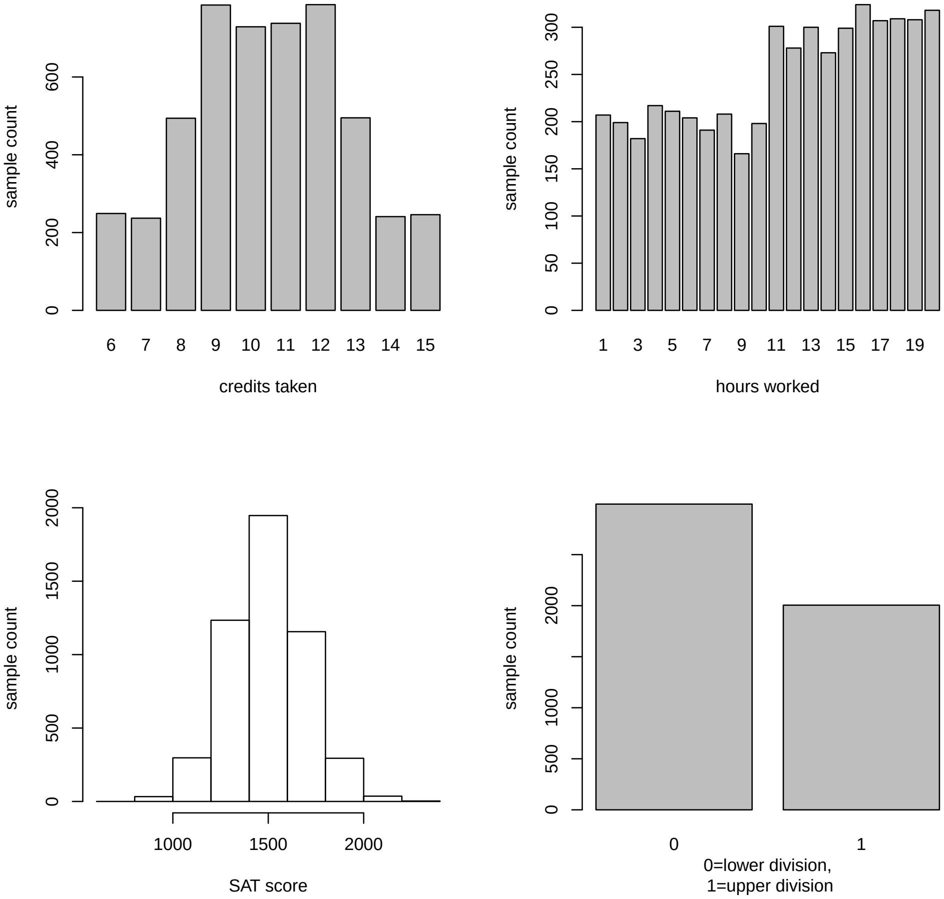

Figure 1. Overview of student attributes

1.3. **Create coefficients, error term, and dependent variable.**

Coefficients

These will describe the marginal effect of each regressor (=explanatory variable) on the outcome.

```
R> bconst<- 1 #intercept term
R> bcredits<- 0.02 #more credits, more study time
R> bwork<- -0.02 #more work, less study
```

```
R> bSAT<- 0.01 #higher SAT = more ability / motivation = more study
R> bupper<- 0.5
R> #see the finish line / gain experience =
R> # more study (you could argue about this..)
```

error term:
This includes all the stuff we don’t observe that - for better or for worse - drives study time (social activities, sports, health, sleep habits,etc). Assume this error is normally distributed with expectation 0 and std. 1.5. Draw n of these - one for each observation.

```
R> eps<-rnorm(n,0,1.5)
```

Dependent variable: Assuming a CLRM, we have

$$y_i = \mathbf{x}_i' \boldsymbol{\beta} + \epsilon_i \tag{1}$$

Implementation in R:

```
R> y<-bconst+bcredits*credits+bwork*work+bSAT*SAT+bupper*upper+eps;
R> # or, more elegantly:
R> bvec=c(bconst,bcredits,bwork,bSAT,bupper)
R> X<-cbind(rep(1,n),credits,work,SAT,upper);
R> y<-X %*% bvec+eps
R> k<-ncol(X)
```

Show descriptive stats to make sure we get a reasonable range for ”study time”, our outcome variable; else tweek the parameters.

```
R> tt<-data.frame(col1=c("credits","work","SAT","upper","study time"),
                 col2=c(mean(credits),mean(work),mean(SAT),mean(upper),mean(y)),
                 col3=c(sd(credits),sd(work),sd(SAT),sd(upper),sd(y)),
                 col4=c(min(credits),min(work),min(SAT),min(upper),min(y)),
                 col5=c(max(credits),max(work),max(SAT),max(upper),max(y)))
R> colnames(tt)<-c("variable","mean","std","min","max")

R> print(xtable(tt,caption="summary statistics for regressors"),
 include.rownames=FALSE,
 latex.environment="center", caption.placement="top",table.placement="!h")
```

Table 1. summary statistics for regressors

variable	mean	std	min	max
credits	10.50	2.28	6.00	15.00
work	11.56	5.74	1.00	20.00
SAT	1498.40	200.63	838.52	2250.34
upper	0.40	0.49	0.00	1.00
study time	16.15	2.51	6.96	24.74

```
R> #get rid of row counters, and center over decimal)
```

NOTE: When we created the explanatory variables, we implicitly assumed:

(1) *independence across observations*: one student's outcome is independent of another's.
(2) *orthogonality of regressors*: somebody's SAT score has nothing to do with that person's weekly workload or vice versa, work has nothing to do with "upper" or vice versa, etc.

In reality, both are likely violated. We will learn about the implications of this shortly.

2. Estimation

Let's first omit a few regressors. This will move more "stuff" into the error term. Since our artificial variables should have very low correlation with each other, the effect of this omission will primarily manifest itself in a loss in precision, i.e. higher standard errors and lower t-values.

```
R> X<-cbind(rep(1,n),credits,work)
R> k<-ncol(X)
R> bols<-solve((t(X)) %*% X) %*% (t(X) %*% y)# compute OLS estimator
R>  e<-y-X%*%bols # Get residuals.
R> s2<-(t(e)%*%e)/(n-k) #get the regression error (estimated variance of "eps").
R> Vb<-s2[1,1]*solve((t(X))%*%X)
R> # get the estimated variance-covariance matrix of bols
R> se=sqrt(diag(Vb)) # get the standard erros for your coefficients;
R> tval=bols/se # get your t-values.
```

Display results in a nice table:

```
R> tt<-data.frame(col1=c("constant","credits","work"),
                 col2=bvec[1:3],
                 col3=bols,
                 col4=se,
                 col5=tval)
R> colnames(tt)<-c("variable","true value","estimate","s.e.","t")

R> ttx<- xtable(tt,caption="OLS output, Model 1 (incomplete)")
R> digits(ttx)<-3   #decimals to be shown for each column
R> print(ttx,include.rownames=FALSE,
 latex.environment="center", caption.placement="top",table.placement="!h")
```

Table 2. OLS output, Model 1 (incomplete)

variable	true value	estimate	s.e.	t
constant	1.000	16.208	0.182	88.891
credits	0.020	0.027	0.016	1.756
work	-0.020	-0.030	0.006	-4.836

```
R> #get rid of row counters, and center over decimal)
```

The estimated standard deviation of the regression error is 2.503.
Now for the full (correct) model:

```
R> X<-cbind(rep(1,n),credits,work,SAT,upper);
R> k<-ncol(X)
R> bols<-solve((t(X)) %*% X) %*% (t(X) %*% y);# compute OLS estimator
R>  e<-y-X%*%bols # Get residuals.
R> s2<-(t(e)%*%e)/(n-k) #get the regression error (estimated variance of "eps").
R> Vb<-s2[1,1]*solve((t(X))%*%X) # get the estimated variance-cov. matrix of bols
R> se=sqrt(diag(Vb)) # get the standard erros for your coefficients;
R> tval=bols/se # get your t-values.
```

```
R> tt<-data.frame(col1=c("constant","credits","work","SAT","upper"),
             col2=bvec,
             col3=bols,
             col4=se,
             col5=tval)
R> colnames(tt)<-c("variable","true value","estimate","s.e.","t")

R> ttx<- xtable(tt,caption="OLS output, Model 2 (correct)")
R> digits(ttx)<-3   #decimals to be shown for each column
R> print(ttx,include.rownames=FALSE,
 latex.environment="center", caption.placement="top",table.placement="!h")
```

Table 3. OLS output, Model 2 (correct)

variable	true value	estimate	s.e.	t
constant	1.000	1.116	0.192	5.803
credits	0.020	0.019	0.009	2.034
work	-0.020	-0.019	0.004	-5.274
SAT	0.010	0.010	0.000	94.313
upper	0.500	0.451	0.043	10.471

The estimated standard deviation of the regression error is 1.493.

```
R> proc.time()-tic
   user  system elapsed
   2.33    0.10    2.41
```