The Current Population Survey (CPS) is the U.S. Government's monthly survey of employment and labor force participation (find them at `nber.org`). Jobs and households are complex things to describe with numbers, and they evolve with time. Hence, unfortunately, the survey results are complex and evolve with time. The version I worked with (March 2018) included 351,459 records, each 1076 bytes long encoding more than 700 variables. For example, a 39 year-old man reports he is paid by the hour at a rate of \$0.00/hour, works 45 hours/week, and grosses \$1250/week every week. (About 4,000 of the nearly 19,000 employees report they are paid by the hour at a rate of \$0.00/hour.) A 33 year-old women with a masters degree works 2 hours/week at \$50.00/hour, for an annual income of a few thousand dollars, but lives well above the defined poverty line. I'm sure that those 700 variables would explain these anomalies, if I really cared to understand the details. (Understanding the data really is Job One for a data analyst, but frankly I was just looking for an interesting dataset. Don't follow my carelessly lazy example in real life.) I made my life easier by stealing digested CPS results from the book: *Econometric Data Science: A Predictive Modeling Approach* by Francis Diebold at UPenn.edu. However, only three years were included: 1995, 2004, 2012, so I decided to augment those by digesting 2018. Diebold's data is a much simplified subset of the full CPS with just the following variables:

`FEMALE` — int: really a logical; 1 iff female, 0 otherwise

`NONWHITE` — int: really a logical; 0 iff race=white, 1 otherwise

`UNION` — int: really a logical; 1 iff union member, 0 otherwise

`EDUC` — int: approximate years of education; e.g., B.A.=16, Ph.D.=20

`AGE` — int: in years

`EXPER` — int: calculated experience (not part of CPS): `AGE-EDUC-6`

`WAGE` — num: unit: \$/hour, estimated using other CPS data

`LNWAGE` — num: natural log of `WAGE`

Since I created the 2018 file I decided to add additional variables from CPS that interested me. Note that `MJIND, MJOCC, GEDIV` are actually unordered integer categories; They will need to be converted to factors before use (see below), The file `cps18.txt` defines the integer code as text.

`VET` — logi: True if a U.S. veteran

`PRCITSHP` — logi: True if not a U.S. citizen

`MJIND` — int: the category of the industry employing the individual (1:13)

`MJOCC` — int: the category of the employee's occupation (1:10)

`NOEMP` — int: ordered categories of the employer's size by number of employees (0:6)

`GEDIV` — int: census code for location of job (1:9)

`PTOT` — int: ordered categories of total personal income (1:41)

`PERLIS` — int: ordered poverty categories: 1=below poverty level, 4=above 1.5× poverty level (1:4)

`HEA` — int: ordered categories of increasing unhealth (1:5)

1. For each of the four datasets (1995, 2004, 2012, 2018) make a linear model, e.g.:

```
d18a=lm(LNWAGE~FEMALE+NONWHITE+UNION+poly(EDUC,2)+poly(AGE,2),data=D18)
summary(d18a)
```

The datasets have filenames of the form `output`xx`_update.csv` where xx=95, 04, 12, 18; You will need to load each individual dataset into R, e.g., `D18=read.csv("output18_update.csv")`

Make a nice table showing how the estimated coefficient of `FEMALE, NONWHITE, UNION` changed over the four years; also include the adjusted R^2 for each fit. (I would do this by copy & paste into a spreadsheet like `gnumerics`.) Note: since we are fitting log(`WAGE`) the coefficients of `FEMALE, NONWHITE, UNION` essentially show the fractional change in wages associated with each variable. Unless we specify `poly(EDUC,2,raw=TRUE)` the coefficients of the `EDUC, AGE` poly lack immediate interpretation, so don't bother to report those.

Find the `mean` of `FEMALE, NONWHITE, UNION, EDUC, AGE, WAGE` for each dataset. Include these values in the above table. Note that the mean of the first three variables would be the fraction of the surveyed population in each class. Report: which quantities in the table show consistent change (monotonic increase or decrease)? Report: which quantities show nearly consistent change (at most one exceptional year)?

The *residual* is the difference between the actual and the model (actual–model). The reported `Residual standard error` is essentially the standard deviation of those residuals, and hence shows approximately the range of deviation between model and reality. Your fits should have displayed `Residual standard error` of about 0.5, which means that the actual `LNWAGE` is approximately the model ± 0.5, which in turn means the actual `WAGE` is approximately $\exp(\text{model} \pm 0.5)$, or $\exp(\text{model}) \overset{\times}{\div} \exp(0.5)$, where $\exp(0.5) \approx 1.65$, so the calculated `WAGE` might be high by 65% or low by 40%. This relatively large scatter is demonstrated by the smallish $R^2 \approx .3$, which verbally might be reported as "only 30% of the variation is explained by the model".

2. The 2018 dataset includes many more variables; report which are significant:
`out18b=lm(LNWAGE~.-MJIND-MJOCC-GEDIV-PTOT-WAGE,data=D18)`
The first three variables are excluded (the minus sign) as they include a large number of categories in no particular order; `PTOT` is excluded because it is (in theory) essentially a version of (i.e., highly colinear to) `WAGE`. Note that `EXPER` ends up being excluded: since `EXPER` is just a linear combination of `AGE` and `EDUC`, it will be automatically undefined because of the resulting singularity. For each of the following situations report if wage increase or decrease is associated with the variable and if the association was found to be significant (defined here as $p < .05$). (A) You are a vet, (B) You are not a U.S. citizen, (C) You work for a large company, (D) You live well above the poverty line, (E) Your health is poor. You will need to consult `cps18.txt` to decode the direction of these ordered categorical variables.

3. We investigate next the male/female wage differential in various industries (13 options), occupations (10 options), and geographic locations (9 options). (Some of these options hugely reduce the number of samples, for example, just 10 female wages are reported in the mining industry.) We can cover all of these options quickly by defining functions which do the t-tests on subsets of `D18`:

```
goI=function(i){return(t.test(WAGE~FEMALE,data=D18,subset=(MJIND==i))$p.value)}
sapply(1:13,goI)
goO=function(i){return(t.test(WAGE~FEMALE,data=D18,subset=(MJOCC==i))$p.value)}
sapply(1:10,goO)
goG=function(i){return(t.test(WAGE~FEMALE,data=D18,subset=(GEDIV==i))$p.value)}
sapply(1:9,goG)
```

You should find that the majority of industries, occupations and all but one geographic region show male/female wage differences. Q: Report the names of the *NOT* significant categories, e.g., which region did *not* show a significant male/female wage difference?

4. Several of the industries/occupations show significant education and age differences between the sexes, and we know those are also important features affecting wages. So we'll try to 'control' for those variables by putting nearly everything into the regression formula. (Note: another approach would be to use KNN: find the average wage of men who nearly match the characteristics of each

woman.) MJIND, MJOCC, GEDIV are currently encoded as integers but are unordered categories. We need to convert them to factors.

```
D18a=D18
D18a$MJIND=as.factor(D18a$MJIND)
D18a$MJOCC=as.factor(D18a$MJOCC)
D18a$GEDIV=as.factor(D18a$GEDIV)

a18=lm(LNWAGE~FEMALE*MJOCC+FEMALE*MJIND+FEMALE*GEDIV+NONWHITE+UNION+poly(AGE,2)+
                                                        poly(EDUC,2),data=D18a)
```

Note that the base class of factors is included in the Intercept. For the following assume that the poly of AGE and EDUC produce numerical value zero, What is the model value if: (A) nonwhite, female union member and MJIND11, MJOCC4, GEDIV9 (B) white, male, non-union member, and MJIND1, MJOCC1, GEDIV1, (C) white, female non-union member and MJIND1, MJOCC1, GEDIV1.

Note that the listed MJOCC coefficients are generally significant and negative. What does that mean (negative compared to what)? Provide the name of the MJIND that provides the biggest wages for males. Generally the interaction terms with females are not significant. If a female wanted an occupation where she would have a significant wage boost compared to males, which occupation should she choose? What set of options should a female select to maximize her wages? What is the resulting regression model value (still assuming the poly of AGE and EDUC produce numerical value zero). What set of options should a male select to maximize his wages? What is the resulting regression model value.

5. The current chapter covers error estimation using various sampling techniques, so we should practice some of that, even though this dataset will not prove to be a great example. Clearly EDUC and AGE have nonlinear effects which we have modeled using second order poly. What order of poly is best for AGE? In order to see some effect, I've decided to use the (smaller) D95 dataset, and I even go to the unusual step of selecting a training set *smaller* than the testing set, both choices designed to minimize the constraints on training so it can roam into over-fitting.

```
D95a=D95

index=sample(nrow(D95a),floor(nrow(D95a)/3))
D95train=D95a[index,]
D95test=D95a[-index,]
Etest=rep(0,10)
Etrain=rep(0,10)
for (j in 1:10){
d95train=lm(LNWAGE~FEMALE+NONWHITE+UNION+poly(EDUC,2)+poly(AGE,j),data=D95train)
Etrain[j]=sd(d95train$residuals)
out=predict(d95train,D95test)
Etest[j]=sd(out-D95test$LNWAGE)
}
Etrain
Etest
```

Note the initial creation of the vectors Etrain & Etest to store error values about to be created inside the loop: for (j in 1:10). Note the upper-case D and lower-case d.

You can run this set of commands a few times; different index should result is slightly different outcomes. The expected outcome: Etrain shows monotone decreasing behavior, while Etest generally follows Etrain, until over-fitting causes Etest to increase even as Etrain shows continuous improvement. The effect is muted in this dataset as most of the variation is unrelated to AGE and, even with the reduced data set, there are many more datapoints than adjustible parameters.

6. We can try k-fold cross validation; we switch back to the larger `D18a` dataset. The following code tries 10×10 `poly` for `EDUC` and `AGE` and uses $k = 10$ fold cross validation. Consider that this is 1000 fits to a data.frame with more than 10,000 rows. Give it a minute to complete.

```
library(boot)
delta2=matrix(rep(0,100),ncol=10)
aic=matrix(rep(0,100),ncol=10)
for (i in 1:10){
for (j in 1:10){
g18a=glm(LNWAGE~FEMALE+NONWHITE+UNION+poly(EDUC,i)+poly(AGE,j),data=D18a)
delta2[i,j]=cv.glm(D18a,g18a,K=10)$delta[2]
aic[i,j]=g18a$aic
}
}
delta2
aic
```

Notice the (nested) double loop structure. The first two lines initialize two 10×10 matrices to hold the results. `delta2` is a fit-error estimate (mean *square*; take the `sqrt` of it to get something like the 'residual standard error'); `aic` should be familiar to you. Notice that even though we are making linear models we are using `glm` not `lm`; `lm` lacks a nice interface with cross validation. Note that the new action is this code is the `cv.glm` function. Q: report the (i, j) with the lowest `delta2`, but frankly it's rather constant.

The results of this are perhaps disappointing: there is no sign of over-fitting (i.e., a worsening fit with greater flexibility)—the scatter is mostly unrelated to `AGE` & `EDUC` and 10,000 datapoints limits the ability to over-fit using those paramters. Nevertheless there can be no doubt that overfitting is occurring as `EDUC` is actually an ordered category with just 12 categories and we're fitting it to a 10^{th} degree poly.

With multi-variate problems it is difficult to 'see' what the fit function looks like. In this problem most of our variables are factors (and so just change the intercept). If we put together a data.frame where those factors are fixed, and `EDUC` has a continuous run, we can plot those results. We can also plot as points the prediction for the 12 actual category values of `EDUC`. I'm going to include with those points three values for `AGE`: 20, 40, 60. You will see that there is little difference between 40 & 60, while 20 is distinctly less.

```
ed=sort(unique(D18$EDUC))
points=data.frame(ed,rep(F,36),rep(F,36),rep(F,36),c(rep(20,12),rep(40,12),rep(60,12)))
colnames(points)=c("EDUC","FEMALE","NONWHITE","UNION","AGE")
lnwp=predict(g18a,newdata=points)
ed=seq(0,20,length.out=200)
line=data.frame(ed,rep(F,200),rep(F,200),rep(F,200),rep(40,200))
colnames(line)=c("EDUC","FEMALE","NONWHITE","UNION","AGE")
lnwl=predict(g18a,newdata=line)
plot(points$EDUC,lnwp)
lines(line$EDUC,lnwl,add=T)
```

See the clear over-fitting! The fit is running wild at `EDUC` values that are not in the dataset. If we return to `poly(,3)` models more rational behavior is seen:

```
g18b=glm(LNWAGE~FEMALE+NONWHITE+UNION+poly(EDUC,3)+poly(AGE,3),data=D18)

lnwp=predict(g18b,newdata=points)
lnwl=predict(g18b,newdata=line)
plot(points$EDUC,lnwp)
lines(line$EDUC,lnwl)
```

Q: What `EDUC` yields the lowest wages?

To see effective use of cross validation we need fewer data points so flexibility has some freedom to flex. Let's return to `yacht_hydrodynamics.csv` (also not an ideal example as its really gridded data) and mess with the polynomials related to boat speed:

```
df=read.csv("yacht_hydrodynamics.csv")
lin3=lm(RR ~ poly(Fr,6)+P.C*Fr+L.D*Fr+L.B*Fr+I(P.C*Fr^6)+I(L.B*Fr^6)+I(L.D*Fr^6)+
                                                               I(CoB*Fr^6), data=df)
delta2=matrix(rep(0,100),ncol=10)
aic=matrix(rep(0,100),ncol=10)
for (i in 1:10){
for (j in 1:10){
gYHa=glm(RR ~ poly(Fr,i)+P.C*Fr+L.D*Fr+L.B*Fr+I(P.C*Fr^j)+I(L.B*Fr^j)+I(L.D*Fr^j)+
                                                               I(CoB*Fr^j), data=df)
delta2[i,j]=cv.glm(df,gYHa,K=10)$delta[2]
aic[i,j]=gYHa$aic
}
}
delta2
aic
```

Warnings will be generated, but it looks like AIC likes $(i,j)=(7,6)$ and `delta2` likes $(6,6)$ (results will vary for these random selections). Q: what is your `delta2` at $(6,6)$?

LOOCV would be crazy with 10,000 datapoints; we can try it with `yacht_hydrodynamics`:

```
for (i in 1:10){
for (j in 1:10){
gYHb=glm(RR ~ poly(Fr,i)+P.C*Fr+L.D*Fr+L.B*Fr+I(P.C*Fr^j)+I(L.B*Fr^j)+I(L.D*Fr^j)+
I(CoB*Fr^j), data=df)
delta2[i,j]=cv.glm(df,gYHb)$delta[2]
aic[i,j]=gYHb$aic
}
}
delta2
aic
```

The code is identical to the k-fold cross validation except the `K=10` has been dropped. This will do approximately a third of a million fits, and so takes much longer than k-fold CV. `aic` should be exactly the same (as it's produced from the fit not the CV process); For me `delta2` pointed to $(7,6)$ with a broad minimum. Q: what was your `delta2` at $(6,6)$?

For a bootstrap example, return to the simple CPS fits; note that the standard error is available so there is no particular reason to get it via bootstrap, but that's the plan.

```
d95a=lm(LNWAGE~FEMALE+NONWHITE+UNION+poly(EDUC,2)+poly(AGE,2),data=D95)
summary(d95a)
```

Take a look at the output of `sample(5,10,replace=T)`: you get 10 samples of the integers 1:5 with some repeated. We aim to make a random subset of the rows of a data.frame by such a sample of `1:nrow`; in the `D95` data.frame there are 1323 rows, so we'll use `sample(1323,1323,replace=T)`, e.g.,

```
lm(LNWAGE~FEMALE+NONWHITE+UNION+poly(EDUC,2)+poly(AGE,2),data=D95,
                                    subset=sample(1323,1323,replace=T))
```

If you run this command several times (up arrow) you will get slightly different results because different subset will be processed. The `boot` function will itself make the `sample( , ,replace=T)` and pass the result as the second argument to a boot function you define. (The first argument of your boot function must be the data.frame.)

```
bootit=function(d.f,index){
return(coef(lm(LNWAGE~FEMALE+NONWHITE+UNION+poly(EDUC,2)+poly(AGE,2),data=d.f,
subset=index)))
}

boot(D95,bootit,1000)
```

The `boot` takes 1000 samples of your rows, and reports statistics on the (vector) output of the boot function you defined. For each coefficient report the value and error from `boot` and `lm`... make a nice table.