Predicting with regression, multiple covariates

Jeffrey Leek
Johns Hopkins Bloomberg School of Public Health

Example: predicting wages

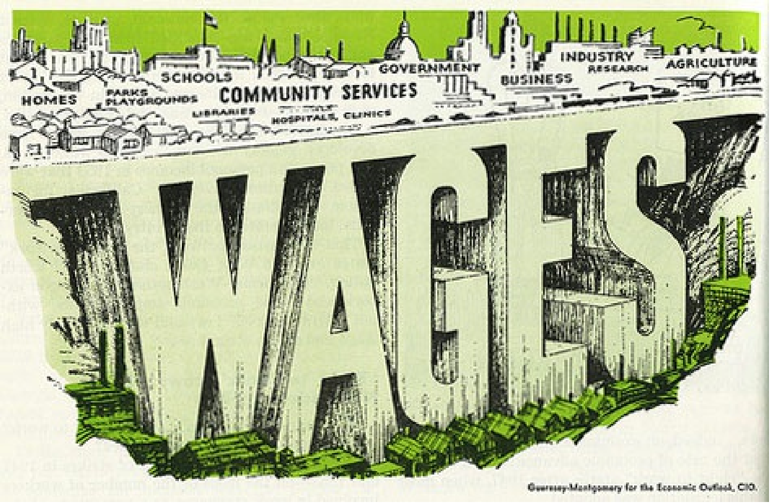


Image Credit http://www.cahs-media.org/the-high-cost-of-low-wages

Data from: ISLR package from the book: Introduction to statistical learning

Example: Wage data

```
library(ISLR); library(ggplot2); library(caret);
data(Wage); Wage <- subset(Wage,select=-c(logwage))
summary(Wage)
```

```
      year           age              sex                      maritl          race     
 Min.   :2003   Min.   :18.0   1. Male  :3000   1. Never Married: 648   1. White:2480  
 1st Qu.:2004   1st Qu.:33.8   2. Female:   0   2. Married      :2074   2. Black: 293  
 Median :2006   Median :42.0                    3. Widowed      :  19   3. Asian: 190  
 Mean   :2006   Mean   :42.4                    4. Divorced     : 204   4. Other:  37  
 3rd Qu.:2008   3rd Qu.:51.0                    5. Separated    :  55                  
 Max.   :2009   Max.   :80.0                                                           

              education                     region               jobclass               health    
 1. < HS Grad       :268   2. Middle Atlantic   :3000   1. Industrial :1544   1. <=Good     : 858  
 2. HS Grad         :971   1. New England       :   0   2. Information:1456   2. >=Very Good:2142  
 3. Some College    :650   3. East North Central:   0                                            
 4. College Grad    :685   4. West North Central:   0                                            
 5. Advanced Degree:426   5. South Atlantic    :   0                                            
                           6. East South Central:   0                                            
                           (Other)              :   0                                            
```

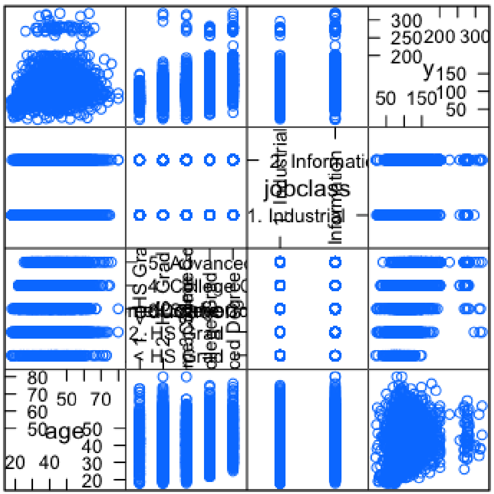
Get training/test sets

```
inTrain <- createDataPartition(y=Wage$wage,
                              p=0.7, list=FALSE)
training <- Wage[inTrain,]; testing <- Wage[-inTrain,]
dim(training); dim(testing)
```

```
[1] 898  12
```

Feature plot

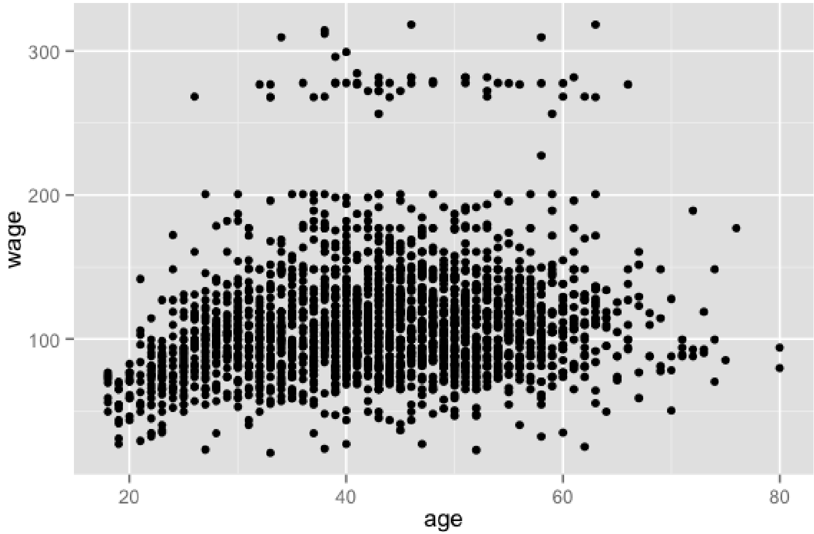
```
featurePlot(x=training[,c("age","education","jobclass")],
            y = training$wage,
            plot="pairs")
```



Scatter Plot Matrix

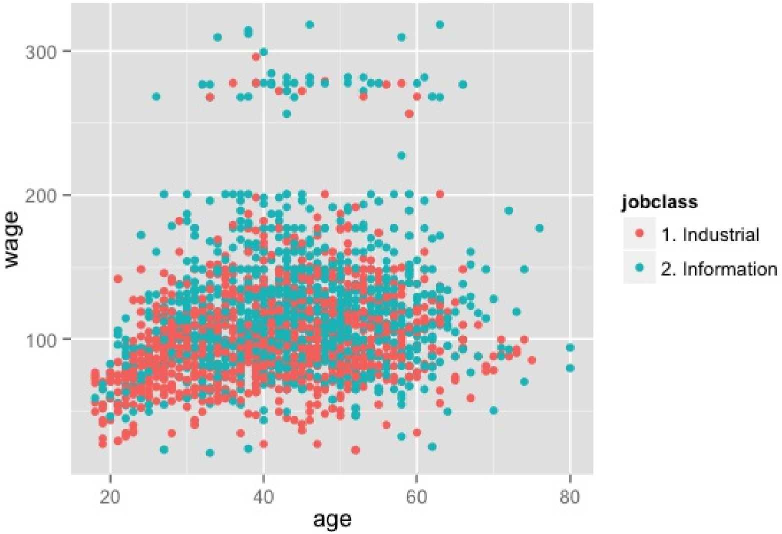
Plot age versus wage

```
qplot(age,wage,data=training)
```



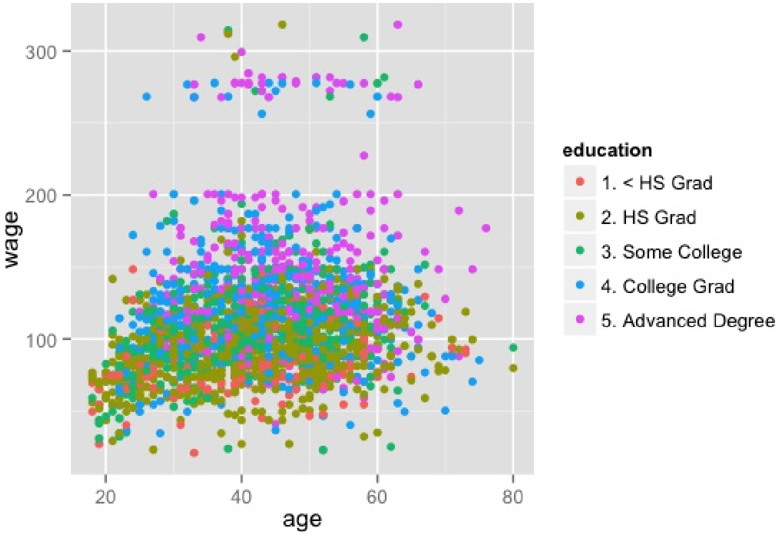
Plot age versus wage colour by jobclass

```
qplot(age,wage,colour=jobclass,data=training)
```



Plot age versus wage colour by education

```
qplot(age,wage,colour=education,data=training)
```



Fit a linear model

$$ED_i = b_0 + b_1 age + b_2 I(Jobclass_i = "Information") + \sum_{k=1}^{4} \gamma_k I(education_i = levelk)$$

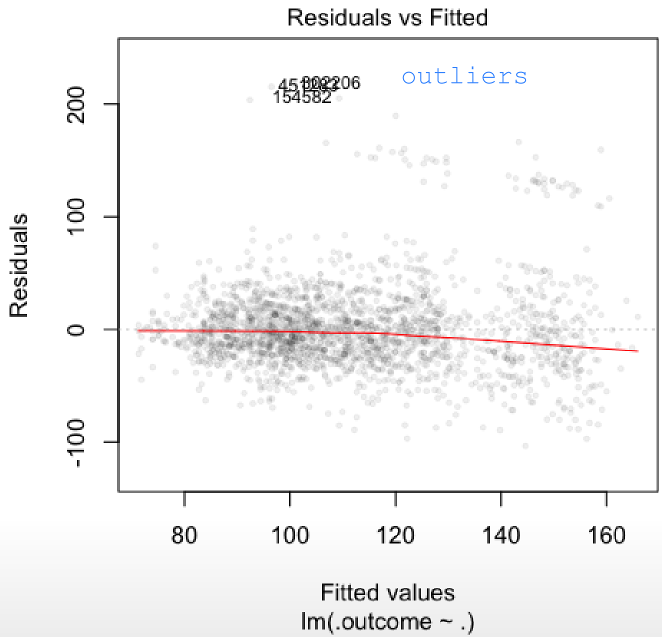
since jobclass and educ. are factors, R automatically creates the corresponding dummy vars

```
modFit<- train(wage ~ age + jobclass + education,
               method = "lm",data=training)
finMod <- modFit$finalModel
print(modFit)
```

```
Linear Regression

2102 samples
  11 predictors  10 predictors because of the dummy variables

No pre-processing
Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 2102, 2102, 2102, 2102, 2102, 2102, ...

Resampling results
```

Diagnostics

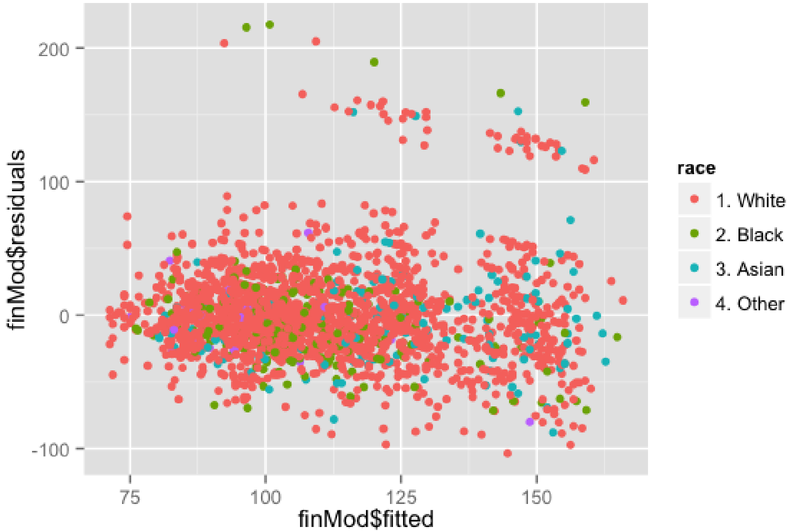
```
plot(finMod,1,pch=19,cex=0.5,col="#00000010")
```



Predictor line near 0: good

Color by variables not used in the model

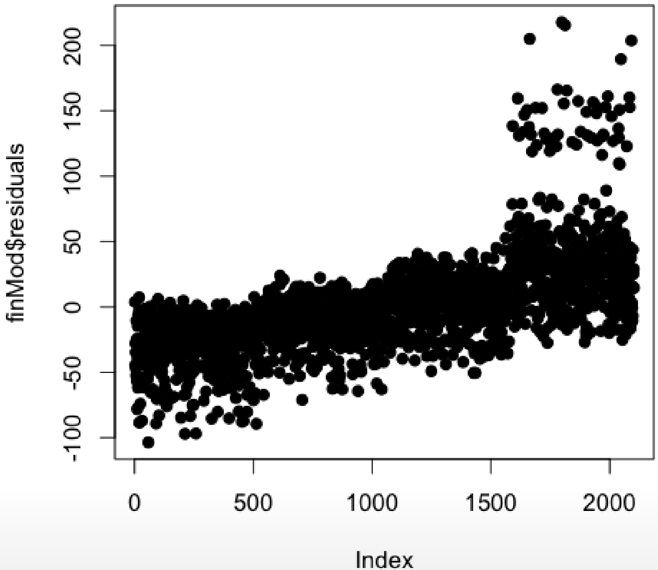
```
qplot(finMod$fitted,finMod$residuals,colour=race,data=training)
```



Plot by index

the index is just the row number of the dataset

```
plot(finMod$residuals,pch=19)
```

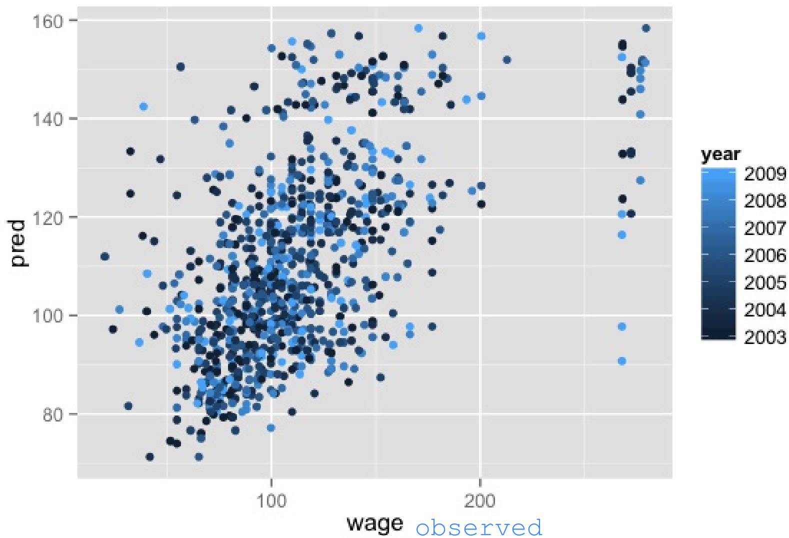


hm…outliers are all at the bottom of the table.
And row index correlates with residuals… strange

So probably the table was sorted by some continuous variable like eg. age!

Predicted versus truth in test set

```
pred <- predict(modFit, testing)
qplot(wage,pred,colour=year,data=testing)
```

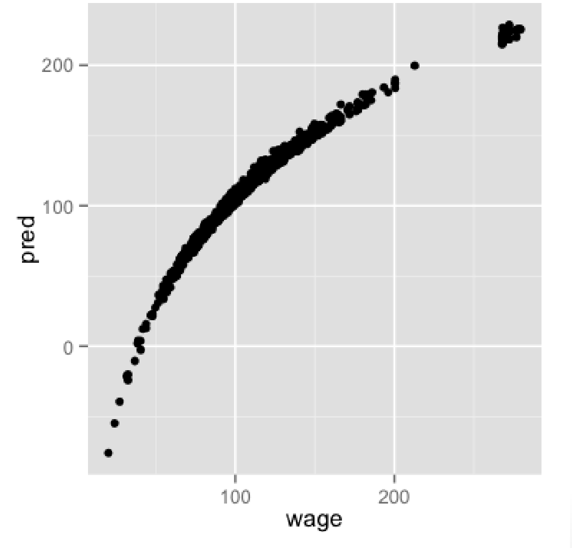


observed

but now if we find some trend here, looking at the test set, we can't use this information to update the model anymore!

If you want to use all covariates

```
modFitAll<- train(wage ~ .,data=training,method="lm")
pred <- predict(modFitAll, testing)
qplot(wage,pred,data=testing)
```



Notes and further reading

- Often useful in combination with other models
- Elements of statistical learning
- Modern applied statistics with S
- Introduction to statistical learning