Workshop on GWAS

misc. topics, PCA and overfitting

Prof. Lei Sun

Department of Statistical Sciences, FAS

Division of Biostatistics, DLSPH

University of Toronto

24 June, 2021

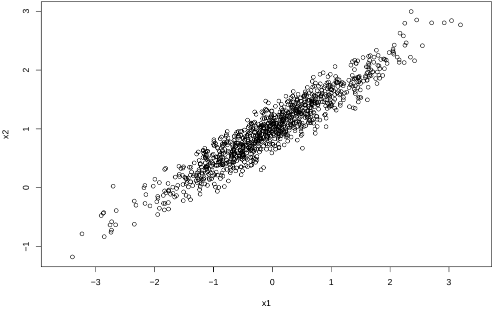
At the end of this lecture: some **basic** understanding of

- Unsupervised learning, e.g. principle component analysis (PCA)
- Supervised learning, e.g. regression and model fit
- Overfitting and model fit vs. prediction

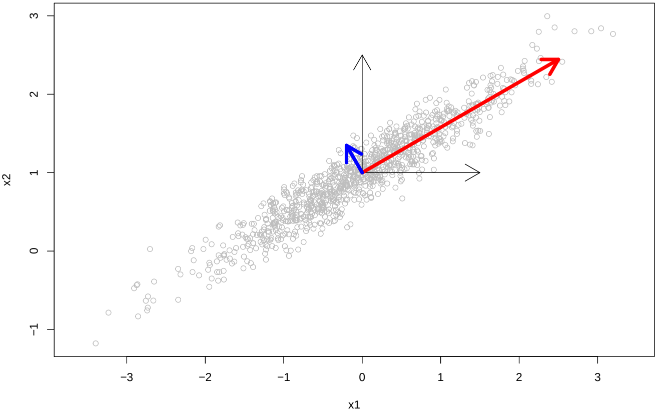
Principle Component Analysis

PCA is commonly used for dimensionality reduction by projecting each data point onto only the first few principal components to obtain lower-dimensional data while preserving as much of the data's variation as possible.

Do we need to report two values/dimensions for each observation?



Dimension reduction via PCA



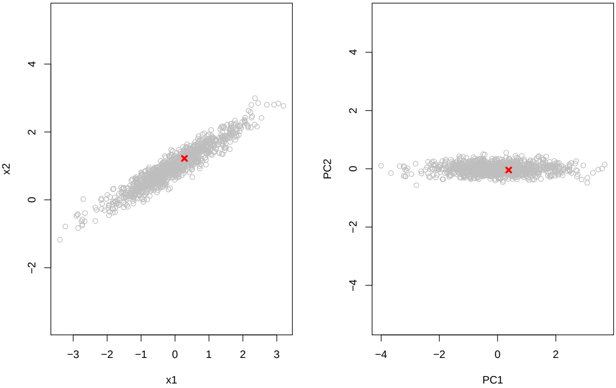
For each observation, need to report two values, (x1, x2), to capture the data along the x1 and x2 directions.

Along the second principle component direction, can assume the values approximately zero for all observations.

The first principle component captures most of the variation in the data: Values vary a lot along this direction.

Contrasting X2 vs. X1 with PC1 vs. PC2

```
x=cbind(x1,x2) # the input data n-by-2 matrix
mypca=prcomp(x) # run the PCA using the prcomp function
par(mfrow=c(1,2))
plot(x1,x2,asp=1,col="grey"); points(x[2,1], x[2,2], pch=4,col="red",lwd=3)
plot(mypca$x[,1],mypca$x[,2],xlab="PC1",ylab="PC2",asp=1,col="grey");
points(mypca$x[2,1],mypca$x[2,2], pch=4, col="red",lwd=3)
```



```
print(x[1:5,]) # original (x1, x2) values/coordinates of the first 5 observations
```

```
##              x1          x2
## [1,] -1.2070657  0.06203358
## [2,]  0.2774292  1.22046720
## [3,]  1.0844412  1.31827336
## [4,] -2.3456977 -0.22721506
## [5,]  0.4291247  1.38834561
```

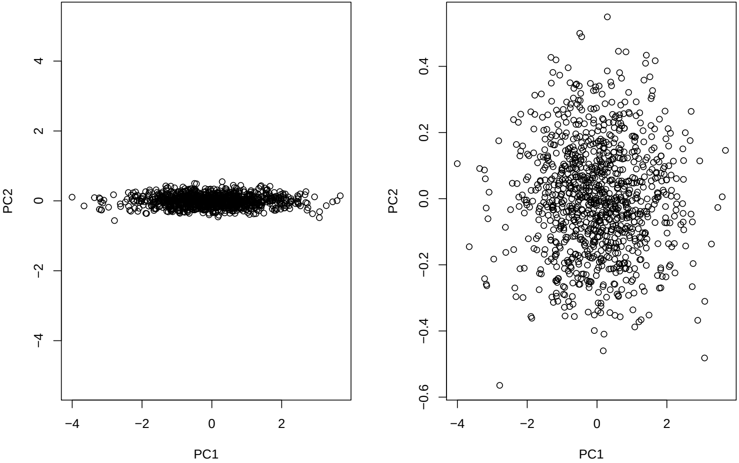
```
print(mypca$x[1:5,]) # the corresponding (pc1, pc2) values/coordinates
```

```
##             PC1         PC2
## [1,] -1.4891846  0.18002188
## [2,]  0.3807161 -0.04170974
## [3,]  1.1216460  0.29274084
## [4,] -2.6129422 -0.16249298
## [5,]  0.5974415 -0.10670807
```

```
summary(mypca) # the `importance' of pc
```

```
## Importance of components:
##                           PC1     PC2
## Standard deviation     1.1616 0.16822
## Proportion of Variance 0.9795 0.02054
## Cumulative Proportion  0.9795 1.00000
```

Many data visulization ‘tricks’ and pitfalls, e.g. the scale!



What's needed to understand the prcomp( ) blackbox?

Mathematics, Mathematics and Mathematics!

Also pay attention to the **Pitfalls and Limitations** in Principle Component Analysis

e.g. correlated samples, correlated SNPs, centering and standardization, and missing data?

Recall the 1000 Genomes project data

```
# mydata.ERAP2=data.table::fread("http://www.utstat.toronto.edu/sun/data/GWAS-workshop-sample-dataset-ERAP2
# Locally if you have already downloaded the data to your working directory
mydata.ERAP2=read.table("GWAS-workshop-sample-dataset-ERAP2.txt",header=T)
colnames(mydata.ERAP2);  nsnps=100
```

```
##   [1] "FID"             "IID"             "PID"             "MID"
##   [5] "SEX"             "PHENO"           "POP"             "SNP1.5618704"
##   [9] "SNP1.57815437"   "SNP1.64302980"   "SNP1.104336159"  "SNP1.151435036"
##  [13] "SNP1.158018135"  "SNP1.173714419"  "rs2782524"       "SNP2.23882292"
##  [17] "SNP2.60375263"   "rs10192914"      "SNP2.112710623"  "rs999891"
##  [21] "SNP2.211265378"  "SNP3.19002158"   "SNP3.61162380"   "SNP3.64305918"
##  [25] "SNP3.71108737"   "SNP3.125827295"  "SNP3.127964403"  "SNP3.157693188"
##  [29] "SNP3.178358711"  "SNP3.184470209"  "rs1518872"       "SNP4.48537429"
##  [33] "SNP4.67185059"   "SNP4.76791598"   "SNP4.121158599"  "SNP4.129152543"
##  [37] "rs10007083"      "SNP4.162915830"  "SNP4.167449697"  "SNP4.173723483"
##  [41] "SNP5.33837406"   "rs10054860"      "rs1056893"       "rs4360063"
##  [45] "rs10044354"      "SNP5.106184146"  "rs40588"         "SNP6.24967500"
##  [49] "SNP6.37032449"   "rs2746304"       "SNP6.69359962"   "SNP6.94678620"
##  [53] "SNP6.108159912"  "rs1006932"       "SNP7.6655522"    "SNP7.8339546"
##  [57] "SNP7.52236127"   "rs10282724"      "SNP7.126907925"  "SNP8.97403295"
##  [61] "SNP9.12237310"   "SNP9.27159704"   "rs7864801"       "SNP9.109943261"
##  [65] "SNP10.15114789"  "rs7912144"       "SNP10.28792948"  "rs7913102"
##  [69] "rs9415825"       "SNP10.80281274"  "rs2020163"       "SNP11.89516883"
##  [73] "SNP11.95415516"  "SNP12.23576923"  "SNP12.47113506"  "rs10220224"
##  [77] "rs9582475"       "rs7339421"       "SNP13.104488021" "rs4772700"
##  [81] "SNP14.27762092"  "rs2737721"       "SNP14.41252775"  "rs1813500"
##  [85] "SNP14.50513718"  "SNP14.89810169"  "rs11845053"      "SNP15.91733944"
##  [89] "SNP16.8669923"   "SNP16.20312407"  "SNP16.57425543"  "SNP16.77578033"
##  [93] "SNP17.60824552"  "SNP17.67165654"  "rs733383"        "rs11870893"
##  [97] "SNP18.549352"    "SNP18.46491800"  "SNP18.66700491"  "SNP19.2446724"
## [101] "SNP19.6387304"   "rs427366"        "SNP20.4494424"   "rs227134"
## [105] "rs4134385"       "SNP22.47353519"  "SNP23.99734210"
```

```r
# a sneak peak at the data; male=1 and female=2 by convention
# CEU: Utah residents with Northern and Western European ancestry from the CEPH collection
# YRI: Yoruba in Ibadan, Nigeria
mydata.ERAP2[c(1:3,103:105,193:195),1:9]
```

```
##       FID     IID PID MID SEX    PHENO POP SNP1.5618704 SNP1.57815437
## 1    1328 NA06984   0   0   1 11.49810 CEU           AA            AA
## 2    1328 NA06989   0   0   2 10.67960 CEU           AA            AA
## 3    1330 NA12340   0   0   1 10.54530 CEU           AA            AA
## 103 13291 NA07435   0   0   1 11.55000 CEU           AA            AA
## 104 13292 NA07051   0   0   1 10.68810 CEU           AA            AA
## 105  Y001 NA18486   0   0   1 10.65360 YRI           CA            GA
## 193  Y116 NA19236   0   0   1 10.84090 YRI           CA            GG
## 194  Y120 NA19247   0   0   2  8.57924 YRI           AA            AA
## 195  Y120 NA19248   0   0   1 11.95420 YRI           CC            AA
```

```r
# Our data matrix is 195 by 107 in dimension
c(length(mydata.ERAP2[,1]),length(mydata.ERAP2[1,]))
```

```
## [1] 195 107
```

```r
# the total number of individuals from each population
c(sum(mydata.ERAP2$POP=="CEU"), sum(mydata.ERAP2$POP=="YRI"))
```

```
## [1] 104  91
```

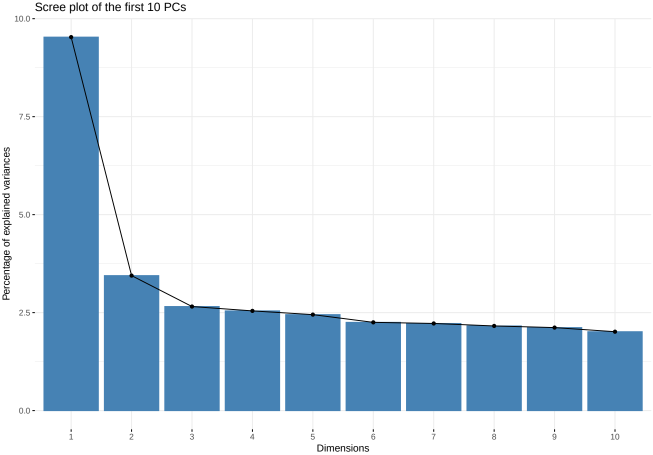
Run PCA on the genotype data, 195 by 100 matrix

```
# the genotype data matrix initial value
x=matrix(-9,nrow=length(mydata.ERAP2[,1]),ncol=100)

for(j in 1:nsnps)
  # use 0, 1 and 2 for the genotype coding; did not deal with missing data issue
  x[,j]=as.numeric(mydata.ERAP2[,(j+7)])-1

# run the PCA
mypca=prcomp(x,center=TRUE,scale=TRUE)
```

```
## Importance of components:
##                           PC1     PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     3.0870 1.85599 1.62960 1.59515 1.56497 1.50073 1.49119
## Proportion of Variance 0.0953 0.03445 0.02656 0.02545 0.02449 0.02252 0.02224
## Cumulative Proportion  0.0953 0.12975 0.15630 0.18175 0.20624 0.22876 0.25100
##                           PC8     PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation     1.4698 1.45547 1.41892 1.40926 1.38144 1.36175 1.35255
## Proportion of Variance 0.0216 0.02118 0.02013 0.01986 0.01908 0.01854 0.01829
## Cumulative Proportion  0.2726 0.29378 0.31392 0.33378 0.35286 0.37140 0.38970
##                           PC15    PC16    PC17    PC18    PC19    PC20    PC21
## Standard deviation     1.33469 1.31119 1.29896 1.27800 1.26986 1.26326 1.24644
## Proportion of Variance 0.01781 0.01719 0.01687 0.01633 0.01613 0.01596 0.01554
## Cumulative Proportion  0.40751 0.42470 0.44158 0.45791 0.47403 0.48999 0.50553
##                           PC22   PC23    PC24    PC25    PC26    PC27    PC28
## Standard deviation     1.23557 1.2000 1.18471 1.16706 1.14679 1.13750 1.12512
## Proportion of Variance 0.01527 0.0144 0.01404 0.01362 0.01315 0.01294 0.01266
## Cumulative Proportion  0.52080 0.5352 0.54923 0.56285 0.57600 0.58894 0.60160
##                          PC29    PC30    PC31    PC32    PC33    PC34   PC35
## Standard deviation     1.1046 1.10220 1.08920 1.07559 1.06678 1.05648 1.0486
## Proportion of Variance 0.0122 0.01215 0.01186 0.01157 0.01138 0.01116 0.0110
## Cumulative Proportion  0.6138 0.62595 0.63781 0.64938 0.66076 0.67192 0.6829
##                           PC36    PC37    PC38    PC39   PC40    PC41    PC42
## Standard deviation     1.04167 1.03480 1.02044 1.00128 0.9901 0.98304 0.97202
## Proportion of Variance 0.01085 0.01071 0.01041 0.01003 0.0098 0.00966 0.00945
## Cumulative Proportion  0.69377 0.70448 0.71489 0.72492 0.7347 0.74438 0.75383
##                           PC43    PC44   PC45    PC46    PC47    PC48    PC49
## Standard deviation     0.95648 0.95078 0.9382 0.92611 0.91205 0.90302 0.89347
## Proportion of Variance 0.00915 0.00904 0.0088 0.00858 0.00832 0.00815 0.00798
## Cumulative Proportion  0.76298 0.77202 0.7808 0.78940 0.79772 0.80587 0.81385
##                           PC50    PC51    PC52    PC53    PC54    PC55   PC56
## Standard deviation     0.87933 0.86394 0.85564 0.85045 0.84138 0.82631 0.8124
## Proportion of Variance 0.00773 0.00746 0.00732 0.00723 0.00708 0.00683 0.0066
## Cumulative Proportion  0.82159 0.82905 0.83637 0.84360 0.85068 0.85751 0.8641
##                           PC57   PC58    PC59    PC60    PC61    PC62    PC63
## Standard deviation     0.79167 0.7809 0.76298 0.75329 0.73600 0.72268 0.71043
## Proportion of Variance 0.00627 0.0061 0.00582 0.00567 0.00542 0.00522 0.00505
## Cumulative Proportion  0.87038 0.8765 0.88230 0.88797 0.89339 0.89861 0.90366
##                           PC64    PC65    PC66    PC67    PC68   PC69    PC70
## Standard deviation     0.70489 0.69914 0.68280 0.66795 0.65155 0.6403 0.62749
```

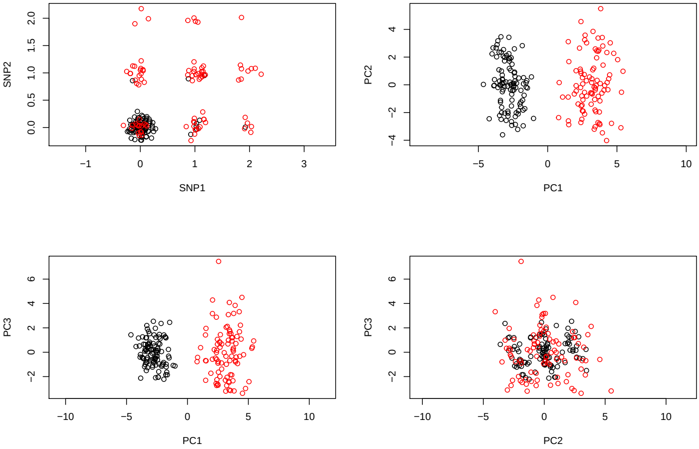
The importance of each PC, Scree plot

```
fviz_eig(mypca,main="Scree plot of the first 10 PCs")
```



Top PCs can seperate populations, unlike the individual SNPs, e.g.

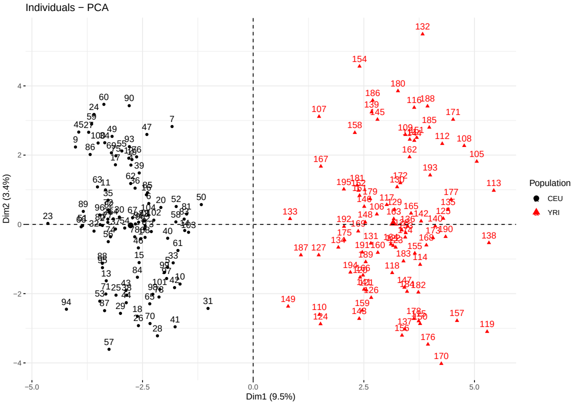
```
par(mfrow=c(2,2))
plot((x[,1]+rnorm(195,0,0.1)),(x[,2]+rnorm(195,0,0.1)),xlab="SNP1",ylab="SNP2",asp=1,col=mydata.ERAP2$POP)
#plot((x[,1]),(x[,2]),xlab="SNP1",ylab="SNP2",asp=1,col=mydata.ERAP2$POP)
plot(mypca$x[,1],mypca$x[,2],xlab="PC1",ylab="PC2",asp=1,col=mydata.ERAP2$POP)
plot(mypca$x[,1],mypca$x[,3],xlab="PC1",ylab="PC3",asp=1,col=mydata.ERAP2$POP)
plot(mypca$x[,2],mypca$x[,3],xlab="PC2",ylab="PC3",asp=1,col=mydata.ERAP2$POP)
```



Quiz: why adding norm(195, 0, 0.1) to the 0, 1 and 2 genotype codings of SNP1 and SNP2?

A closer look at PC2 vs. PC1

```
fviz_pca_ind(mypca,col.ind=mydata.ERAP2$POP,legend.title="Population",palette=c("black","red"))
```



This is an example of unsupervised learning: We have learned the population information of these individuals using their genetic data alone (X) without using the labeled POP data (Y).

What is supervised learning then? Regression analysis is a form of supervised learning!

Supervised learning is the machine learning task of learning a function that maps an input [X] to an output [Y] based on example input-output pairs.

Regression analysis is a set of statistical processes for estimating the relationships between a dependent variable [Y] (often called the 'outcome variable') and one or more independent variables [X] (often called 'predictors', 'covariates', or 'features').

Overfitting

In statistics, overfitting is "the production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably".

Overfitting, a $n = 2$ example

```
mydata.ERAP2[c(1,105,106,195),1:9]
```

```
##      FID     IID PID MID SEX   PHENO POP SNP1.5618704 SNP1.57815437
## 1   1328 NA06984   0   0   1 11.4981 CEU           AA            AA
## 105 Y001 NA18486   0   0   1 10.6536 YRI           CA            GA
## 106 Y001 NA18488   0   0   2 10.3231 YRI           AA            GA
## 195 Y120 NA19248   0   0   1 11.9542 YRI           CC            AA
```

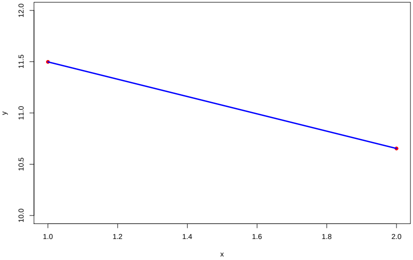
Two observations: always a perfect regression line!

```
x=as.numeric(mydata.ERAP2[c(1,105),]$POP); y=mydata.ERAP2[c(1,105),]$PHENO;  x;y
```

```
## [1] 1 2
```

```
## [1] 11.4981 10.6536
```

```
plot(x,y,pch=19,col="red",ylim=c(10,12))
lines(x,fitted(lm(y~x)),col="blue",lwd=3) # fitted regression line
```



Another $n = 2$ example and a perfect regression line

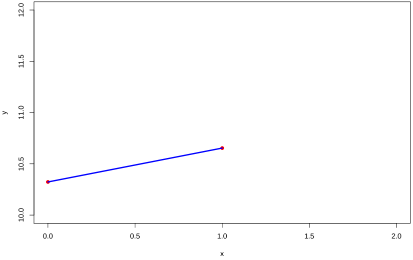
```
mydata.ERAP2[c(105,106),]$SNP1.5618704
```

```
## [1] CA AA
## Levels: AA CA CC
```

```
x=as.numeric(mydata.ERAP2[c(105,106),]$SNP1.5618704)-1 # count the number of copies of allele C
y=mydata.ERAP2[c(105,106),]$PHENO; x;y
```

```
## [1] 1 0

## [1] 10.6536 10.3231
```

```
plot(x,y,pch=19,col="red",xlim=c(0,2),ylim=c(10,12))
lines(x,fitted(lm(y~x)),col="blue",lwd=3) # fitted regression line
```



How about $n = 3$? A ploynomial regression then delivers the perfect fit!

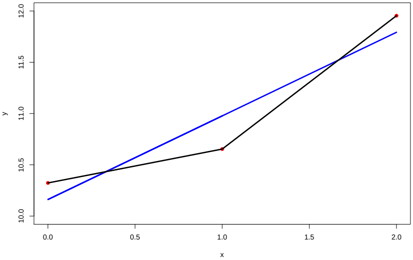
```
mydata.ERAP2[c(105,106,195),]$SNP1.5618704
```

```
## [1] CA AA CC
## Levels: AA CA CC
```

```
x=as.numeric(mydata.ERAP2[c(105,106,195),]$SNP1.5618704)-1 # count the number of copies of allele C
y=mydata.ERAP2[c(105,106,195),]$PHENO; x;y
```

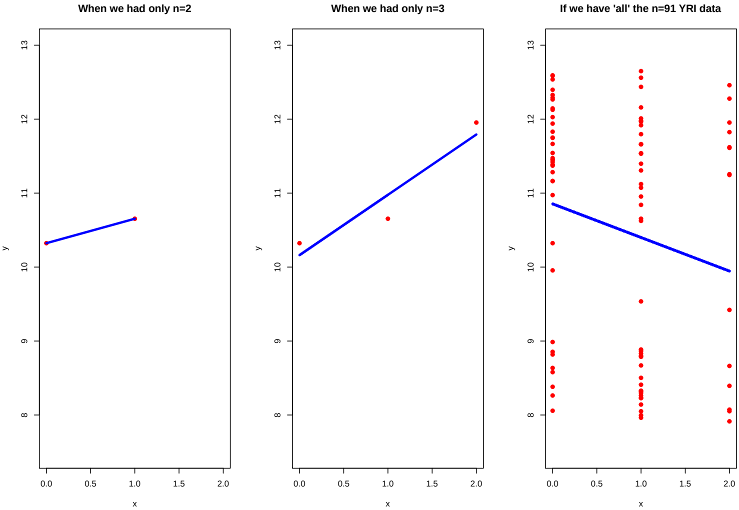
```
## [1] 1 0 2

## [1] 10.6536 10.3231 11.9542
```

```
plot(x,y,pch=19,col="red",ylim=c(10,12))
lines(x,fitted(lm(y~x)),col="blue",lwd=3) # fitted linear regression line

myfit=lm(y~x+I(x^2)) # including x-squared; i.e. polynomial regression fit=lm(y~poly(x,2))
lines(sort(x),myfit$fitted.values[order(x)],col="black",lwd=3)
```



‘Large’ *n* is important!

Continue the study of association between X=SNP1.5618704 and Y=trait (gene expression of *ERAP2*) in the YRI sample:



Large *n* is in the context of the number of predictors

Why didn't we use ALL 100 SNPs simultaneously to fit a multivariate (multiple predictors) regression model?

```r
# the genotype data matrix for the nsnps=100 SNPs
x=matrix(-9,nrow=length(mydata.ERAP2[,1]),ncol=nsnps)
for(j in 1:nsnps)
  x[,j]=as.numeric(mydata.ERAP2[,(j+7)])-1

# Use only the 91 individuals from the YRI population (population homogeneity)
sample.index=(mydata.ERAP2$POP=="YRI")
x=x[sample.index,]
y=mydata.ERAP2[sample.index,]$PHENO

# When call lm(y~x)),
# x is a 91 by 100 matrix for the 91 individuals and
# their genotypes of the 100 SNPs, i.e. p=91 predictors (dimensions)
c(length(x[,1]),length(x[1,]))
```

```
## [1]  91 100
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
## ALL 91 residuals are 0: no residual degrees of freedom!
##
## Coefficients: (10 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -255.3779         NA      NA       NA
## x1            -1.8079         NA      NA       NA
## x2            -8.5452         NA      NA       NA
## x3            -3.3781         NA      NA       NA
## x4             6.3520         NA      NA       NA
## x5             1.5279         NA      NA       NA
## x6             0.8207         NA      NA       NA
## x7             4.7440         NA      NA       NA
## x8            -5.3956         NA      NA       NA
## x9           -28.9569         NA      NA       NA
## x10           -3.5585         NA      NA       NA
## x11           11.4246         NA      NA       NA
## x12          -24.5252         NA      NA       NA
## x13            2.5082         NA      NA       NA
## x14           -6.8579         NA      NA       NA
## x15           15.5542         NA      NA       NA
## x16          -12.0797         NA      NA       NA
## x17           -1.9117         NA      NA       NA
## x18           17.1463         NA      NA       NA
## x19            3.5276         NA      NA       NA
## x20           39.9265         NA      NA       NA
## x21           30.8853         NA      NA       NA
## x22           25.4466         NA      NA       NA
## x23            7.0048         NA      NA       NA
## x24            7.2786         NA      NA       NA
## x25            4.0077         NA      NA       NA
## x26           14.2465         NA      NA       NA
## x27           13.0897         NA      NA       NA
## x28          -11.1000         NA      NA       NA
## x29           -0.8119         NA      NA       NA
```

Looking ahead: Statistical techinques for the $n << p$ issue

Regularized least squares, e.g. Elastic net regularization and Least Absolute Shrinkage and Selection Operator (LASSO)

In essence, in addition to minimizing the difference between y_i and $\hat{y}_i$, e.g.

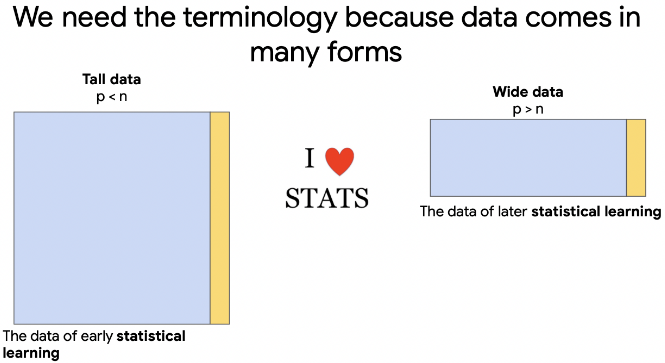
$$\text{minimizing } \sum_i (y_i - \hat{y}_i)^2 \text{ or } \sum_i |y_i - \hat{y}_i|,$$

also considering how many predictors (and their estimated importance) used to obtain the fitted value $\hat{y}_i$ (e.g. $= \sum_j \hat{\beta}_j x_j$).

That is, also minimizing, e.g. the number of non-zero $\hat{\beta}_j$, or

$$\text{also minimizing } \sum_j \hat{\beta}_j^2 \text{ or } \sum_j |\hat{\beta}_j|.$$

A graphic display of the $n << p$ issue[1]



We need the terminology because data comes in many forms

Tall data

p < n

The data of early **statistical learning**

I ♥ STATS

Wide data

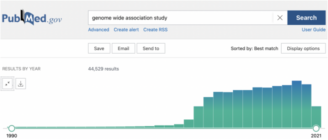
p > n

The data of later **statistical learning**

With vs. without the yellow column (Y): supervised vs. unsupervised learning

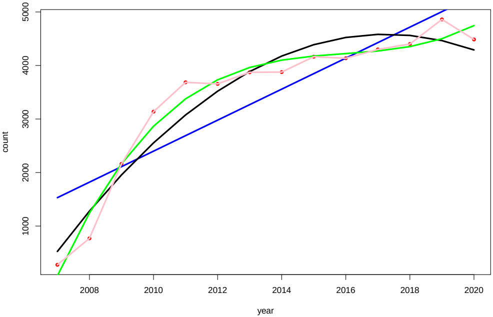
[1] Prof. Jessica Gronsbell, *develop statistical methods that bridge classical theory with modern machine learning tools in an effort to extract reliable insights from large observational health data sets such as electronic health records.*

Model fitting $\neq$ Prediction e.g. the growth of GWAS research


genome wide association study

44,529 results

```
# Search query: polygenic risk score on March 17, 2021
year=c(2007,2008,2009,2010,2011,2012,2013,2014,2015,2016,2017,2018,2019,2020)
count=c(276,770,2162,3137,3686,3656,3873,3877,4164,4135,4300,4401,4861,4486)
plot(year,count,col="red",pch=16)
lines(year,fitted(lm(count~year)),col="blue",lwd=3)
lines(year,fitted(lm(count~poly(year,2))),col="black",lwd=3)
lines(year,fitted(lm(count~poly(year,3))),col="green",lwd=3)
lines(year,fitted(lm(count~poly(year,13))),col="pink",lwd=3)
```



Parting messages told by examples

Women will sprint faster than men in the year 2156?

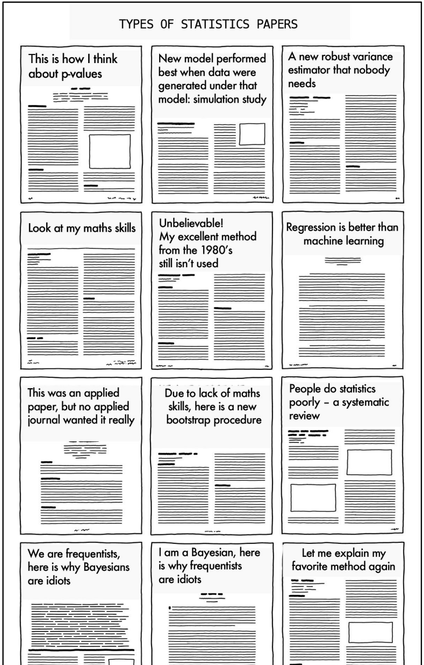
What is wrong with this correlation?

Machine Learning Faces a Reckoning in Health Research

Quiz: what is the statistical keyword here that summarizes the issue discussed below?

For example, 16 of the 62 studies used a dataset of images of children's lungs as the healthy control—without mentioning it in the methodology—then tested the algorithms on images from adults with COVID-19, essentially training the model to tell the difference between children and adults, not healthy versus infected. Additionally, some models were trained on datasets too small to be effective or did not specify where the data came from.

A bit humor does not hurt


TYPES OF STATISTICS PAPERS
This is how I think about p-values
New model performed best when data were generated under that model: simulation study
A new robust variance estimator that nobody needs
Look at my maths skills
Unbelievable! My excellent method from the 1980's still isn't used
Regression is better than machine learning
This was an applied paper, but no applied journal wanted it really
Due to lack of maths skills, here is a new bootstrap procedure
People do statistics poorly – a systematic review
We are frequentists, here is why Bayesians are idiots
I am a Bayesian, here is why frequentists are idiots
Let me explain my favorite method again