QCB 508 – Week 12

John D. Storey

Spring 2020

Contents

HD Latent Variable Models

Definition

Latent variables (or hidden variables) are random variables that are present in the underlying probabilistic model of the data, but they are unobserved.

In high-dimensional data, there may be latent variables present that affect many variables simultaneously.

These are latent variables that induce **systematic variation**. A topic of much interest is how to estimate these and incorporate them into further HD inference procedures.

Model

Suppose we have observed data $\boldsymbol{Y}_{m \times n}$ of m variables with n observations each. Suppose there are r latent variables contained in the r rows of $\boldsymbol{Z}_{r \times n}$ where

$$\mathrm{E}\left[\boldsymbol{Y}_{m \times n} \mid \boldsymbol{Z}_{r \times n}\right] = \boldsymbol{\Phi}_{m \times r} \boldsymbol{Z}_{r \times n}.$$

Let's also assume that $m \gg n > r$. The latent variables $\boldsymbol{Z}$ induce systematic variation in variable $\boldsymbol{y}_i$ parameterized by $\boldsymbol{\phi}_i$ for $i = 1, 2, \ldots, m$.

Estimation

There exist methods for estimating the row space of $\boldsymbol{Z}$ with probability 1 as $m \to \infty$ for a fixed n in two scenarios.

Leek (2011) shows how to do this when $\boldsymbol{y}_i | \boldsymbol{Z} \sim \mathrm{MVN}(\boldsymbol{\phi}_i \boldsymbol{Z}, \sigma_i^2 \boldsymbol{I})$, and the $\boldsymbol{y}_i | \boldsymbol{Z}$ are jointly independent.

Chen and Storey (2015) show how to do this when the $\boldsymbol{y}_i | \boldsymbol{Z}$ are distributed according to a single parameter exponential family distribution with mean $\boldsymbol{\phi}_i \boldsymbol{Z}$, and the $\boldsymbol{y}_i | \boldsymbol{Z}$ are jointly independent.

Jackstraw

Suppose we have a reasonable method for estimating $\boldsymbol{Z}$ in the model

$$\mathrm{E}\left[\boldsymbol{Y} \mid \boldsymbol{Z}\right] = \boldsymbol{\Phi} \boldsymbol{Z}.$$

The **jackstraw** method allows us to perform hypothesis tests of the form

$$H_0 : \boldsymbol{\phi}_i = \boldsymbol{0} \text{ vs } H_1 : \boldsymbol{\phi}_i \neq \boldsymbol{0}.$$

We can also perform this hypothesis test on any subset of the columns of $\boldsymbol{\Phi}$.

This is a challening problem because we have to "double dip" in the data $\boldsymbol{Y}$, first to estimate $\boldsymbol{Z}$, and second to perform significance tests on $\boldsymbol{\Phi}$.

Procedure

The first step is to form estimate $\hat{\boldsymbol{Z}}$ and then test statistic t_i that performs the hypothesis test for each $\boldsymbol{\phi}_i$ from $\boldsymbol{y}_i$ and $\hat{\boldsymbol{Z}}$ ($i = 1, \ldots, m$). Assume that the larger t_i is, the more evidence there is against the null hypothesis in favor of the alternative.

Next we randomly select s rows of $\boldsymbol{Y}$ and permute them to create data set $\boldsymbol{Y}^0$. Let this set of s variables be indexed by $\mathcal{S}$. This breaks the relationship between $\boldsymbol{y}_i$ and $\boldsymbol{Z}$, thereby inducing a true H_0, for each $i \in \mathcal{S}$.

We estimate $\hat{\boldsymbol{Z}}^0$ from $\boldsymbol{Y}^0$ and again obtain test statistics t_i^0. Specifically, the test statistics t_i^0 for $i \in \mathcal{S}$ are saved as draws from the null distribution.

We repeat permutation procedure B times, and then utilize all saved sB permutation null statistics to calculate empirical p-values:

$$p_i = \frac{1}{sB} \sum_{b=1}^{B} \sum_{k \in \mathcal{S}_b} 1 \left(t_k^{0b} \geq t_i \right).$$

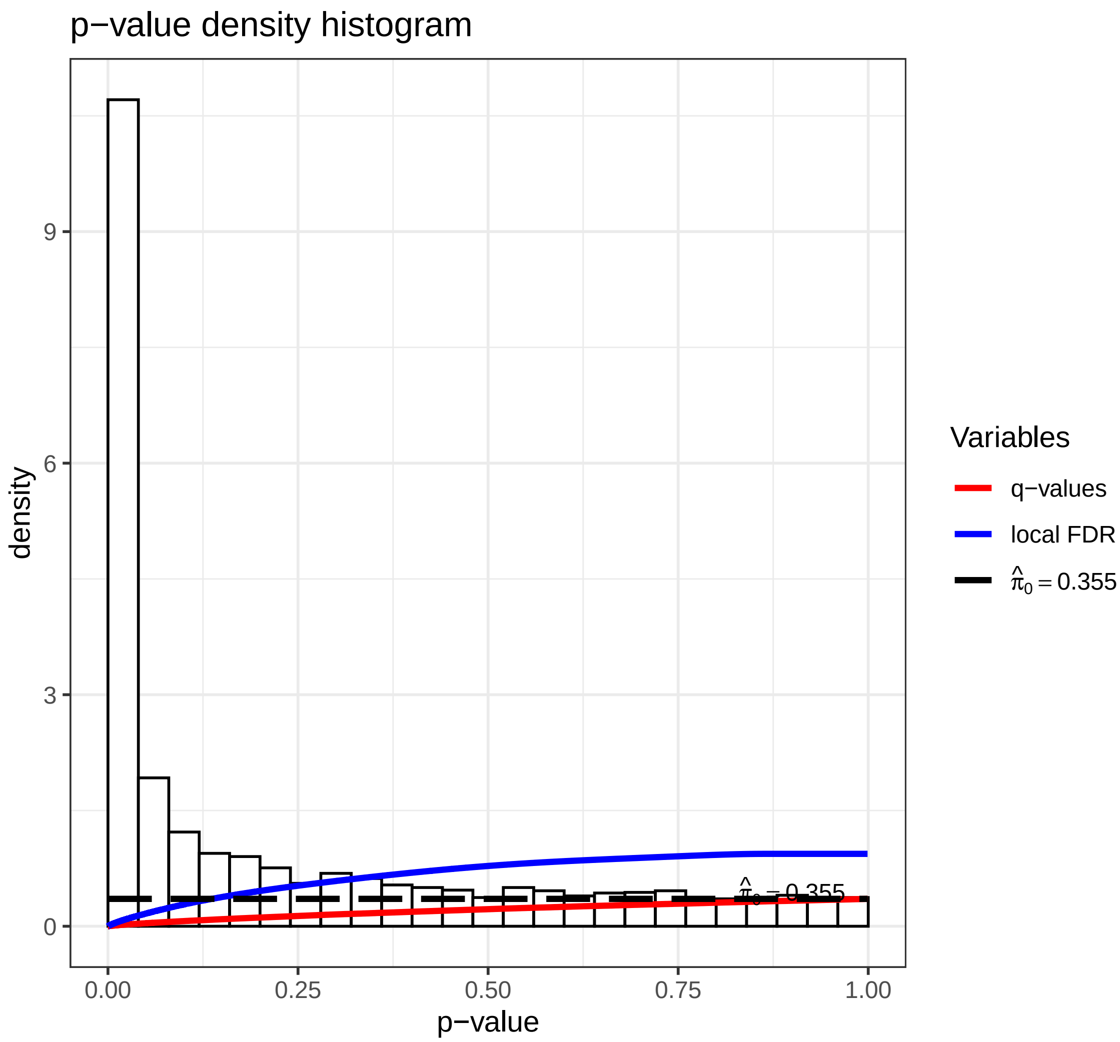
Example: Yeast Cell Cycle

Recall the yeast cell cycle data from earlier. We will test which genes have expression significantly associated with PC1 and PC2 since these both capture cell cycle regulation.

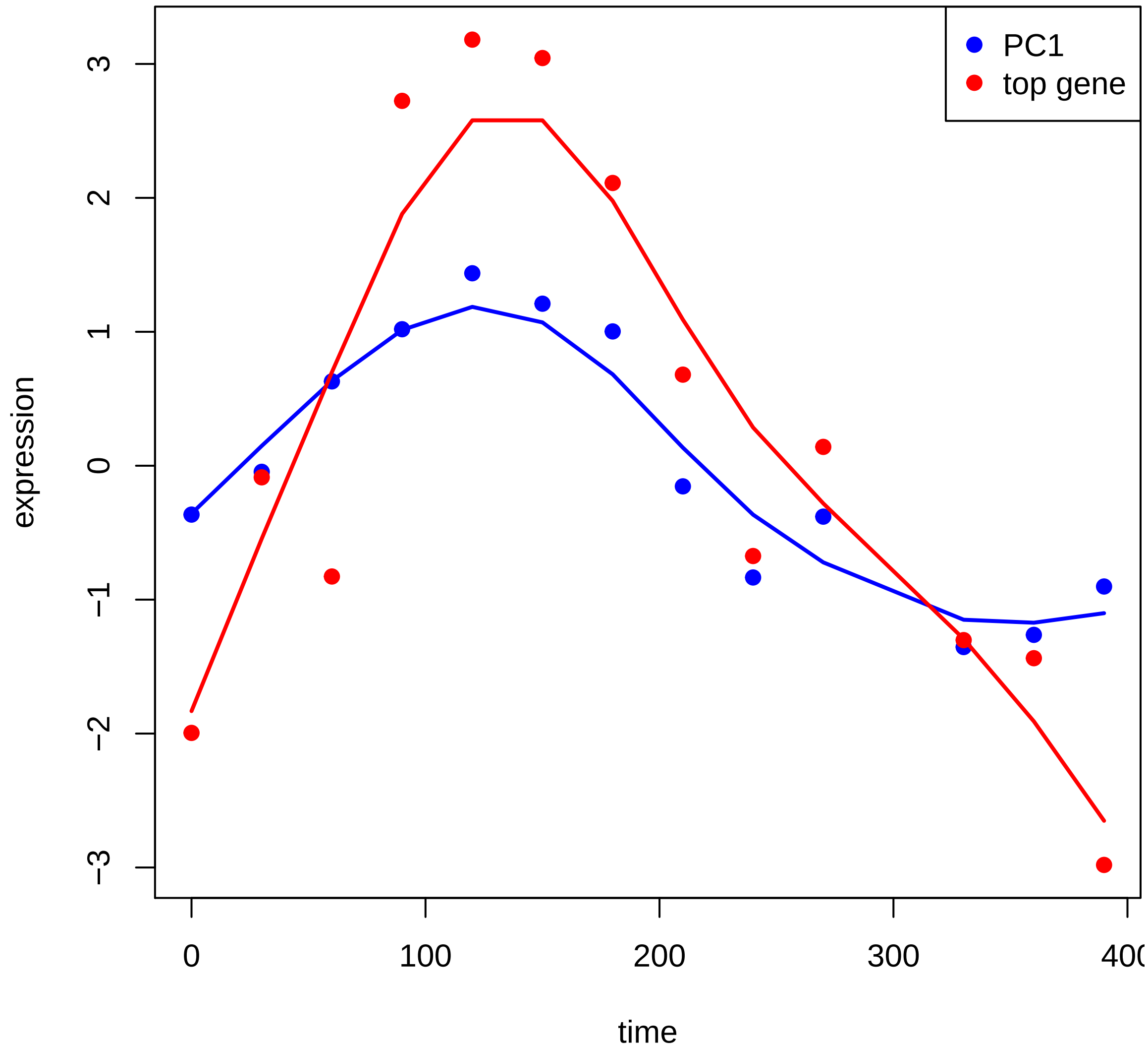
```
> library(jackstraw)
> load("./data/spellman.RData")
> time
 [1]   0  30  60  90 120 150 180 210 240 270 330 360 390
> dim(gene_expression)
[1] 5981   13
> dat <- t(scale(t(gene_expression), center=TRUE, scale=FALSE))
```

Test for associations between PC1 and each gene, conditioning on PC1 and PC2 being relevant sources of systematic variation.

```
> jsobj <- jackstraw_pca(dat, r1=1, r=2, B=500, s=50, verbose=FALSE)
> jsobj$p.value %>% qvalue() %>% hist()
```

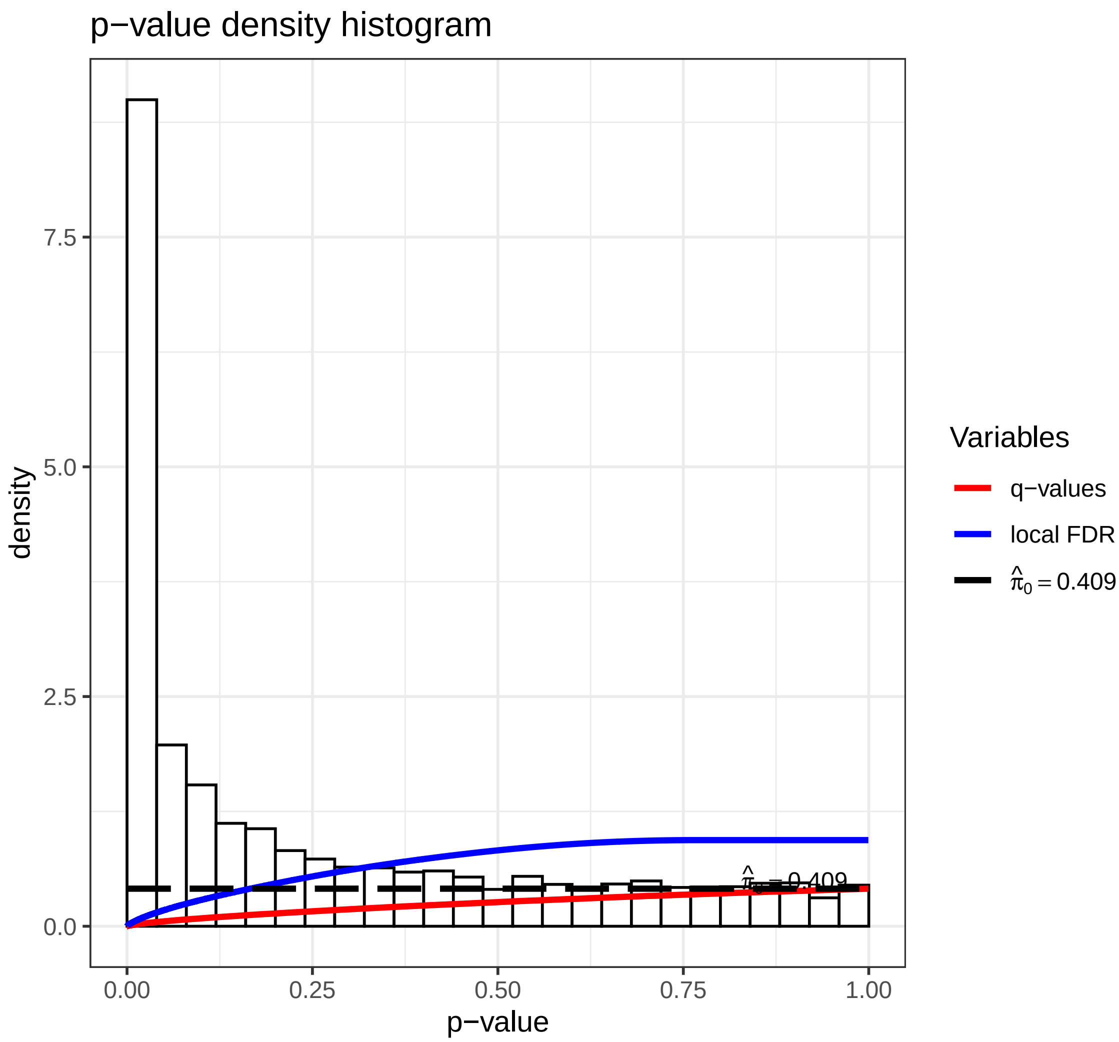


This is the most significant gene plotted with PC1.

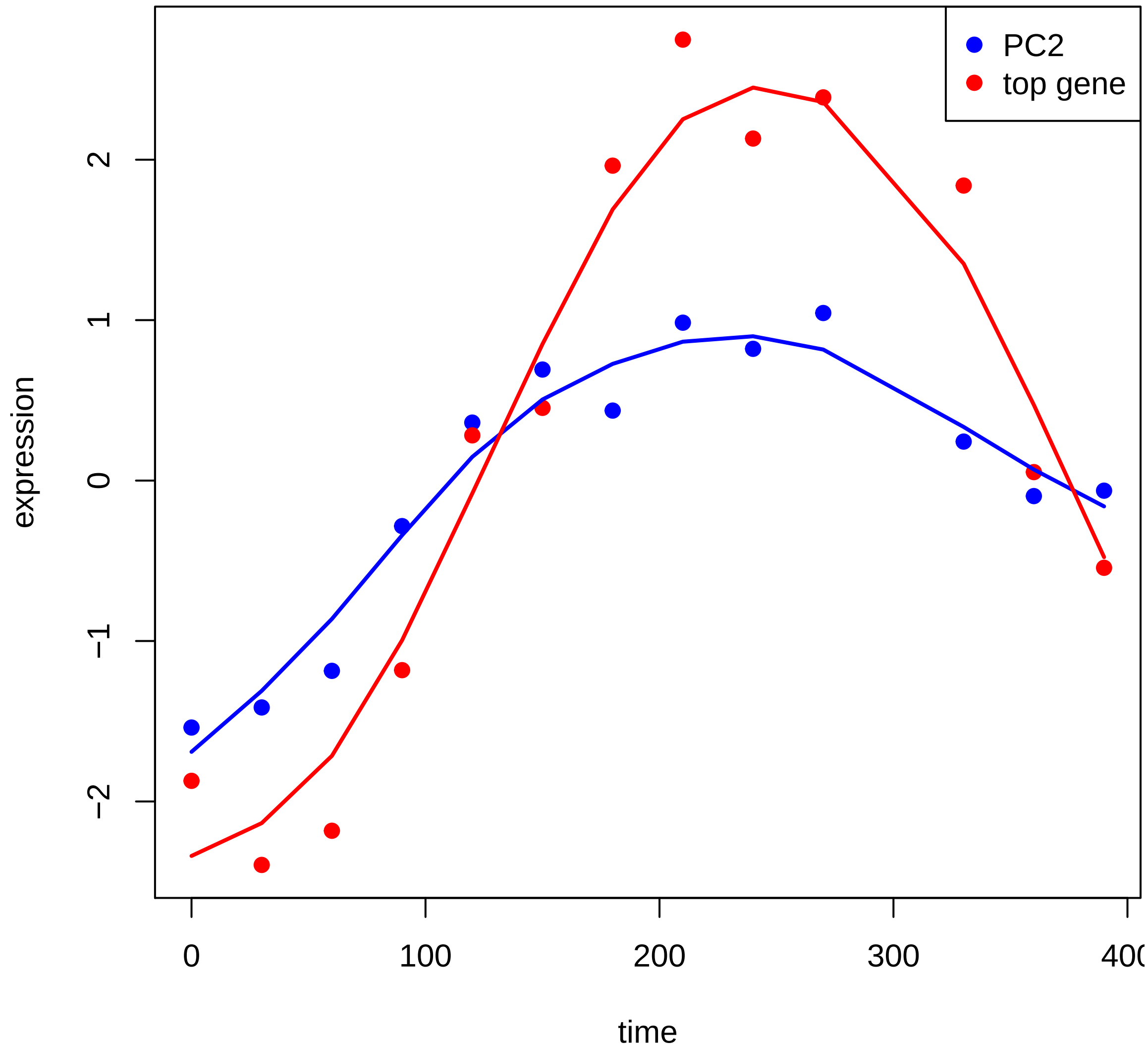


Test for associations between PC2 and each gene, conditioning on PC1 and PC2 being relevant sources of systematic variation.

```
> jsobj <- jackstraw_pca(dat, r1=2, r=2, B=500, s=50, verbose=FALSE)
> jsobj$p.value %>% qvalue() %>% hist()
```



This is the most significant gene plotted with PC2.



Surrogate Variable Analysis

The **surrogate variable analysis** (SVA) model combines the many responses model with the latent variable model introduced above:

$$\boldsymbol{Y}_{m \times n} = \boldsymbol{B}_{m \times d} \boldsymbol{X}_{d \times n} + \boldsymbol{\Phi}_{m \times r} \boldsymbol{Z}_{r \times n} + \boldsymbol{E}_{m \times n}$$

where $m \gg n > d + r$.

Here, only $\boldsymbol{Y}$ and $\boldsymbol{X}$ are observed, so we must combine many regressors model fitting techniques with latent variable estimation.

The variables $\boldsymbol{Z}$ are called **surrogate variables** for what would be a complete model of all systematic variation.

Procedure

The main challenge is that the row spaces of $\boldsymbol{X}$ and $\boldsymbol{Z}$ may overlap. Even when $\boldsymbol{X}$ is the result of a randomized experiment, there will be a high probability that the row spaces of $\boldsymbol{X}$ and $\boldsymbol{Z}$ have some overlap.

Therefore, one cannot simply estimate $\boldsymbol{Z}$ by applying a latent variable esitmation method on the residuals $\boldsymbol{Y} - \hat{\boldsymbol{B}}\boldsymbol{X}$ or on the observed response data $\boldsymbol{Y}$. In the former case, we will only estimate $\boldsymbol{Z}$ in the space orthogonal to $\hat{\boldsymbol{B}}\boldsymbol{X}$. In the latter case, the estimate of $\boldsymbol{Z}$ may modify the signal we can estimate in $\boldsymbol{B}\boldsymbol{X}$.

A recent method, takes an EM approach to esitmating $\boldsymbol{Z}$ in the model

$$\boldsymbol{Y}_{m \times n} = \boldsymbol{B}_{m \times d}\boldsymbol{X}_{d \times n} + \boldsymbol{\Phi}_{m \times r}\boldsymbol{Z}_{r \times n} + \boldsymbol{E}_{m \times n}.$$

It is shown to be necessary to penalize the likelihood in the estimation of $\boldsymbol{B}$ — i.e., form shrinkage estimates of $\boldsymbol{B}$ — in order to properly balance the row spaces of $\boldsymbol{X}$ and $\boldsymbol{Z}$.

The regularized EM algorithm, called **cross-dimensonal inference** (CDI) iterates between

1. Estimate $\boldsymbol{Z}$ from $\boldsymbol{Y} - \hat{\boldsymbol{B}}^{\text{Reg}}\boldsymbol{X}$
2. Estimate $\boldsymbol{B}$ from $\boldsymbol{Y} - \hat{\boldsymbol{\Phi}}\hat{\boldsymbol{Z}}$

where $\hat{\boldsymbol{B}}^{\text{Reg}}$ is a regularized or shrunken estimate of $\boldsymbol{B}$.

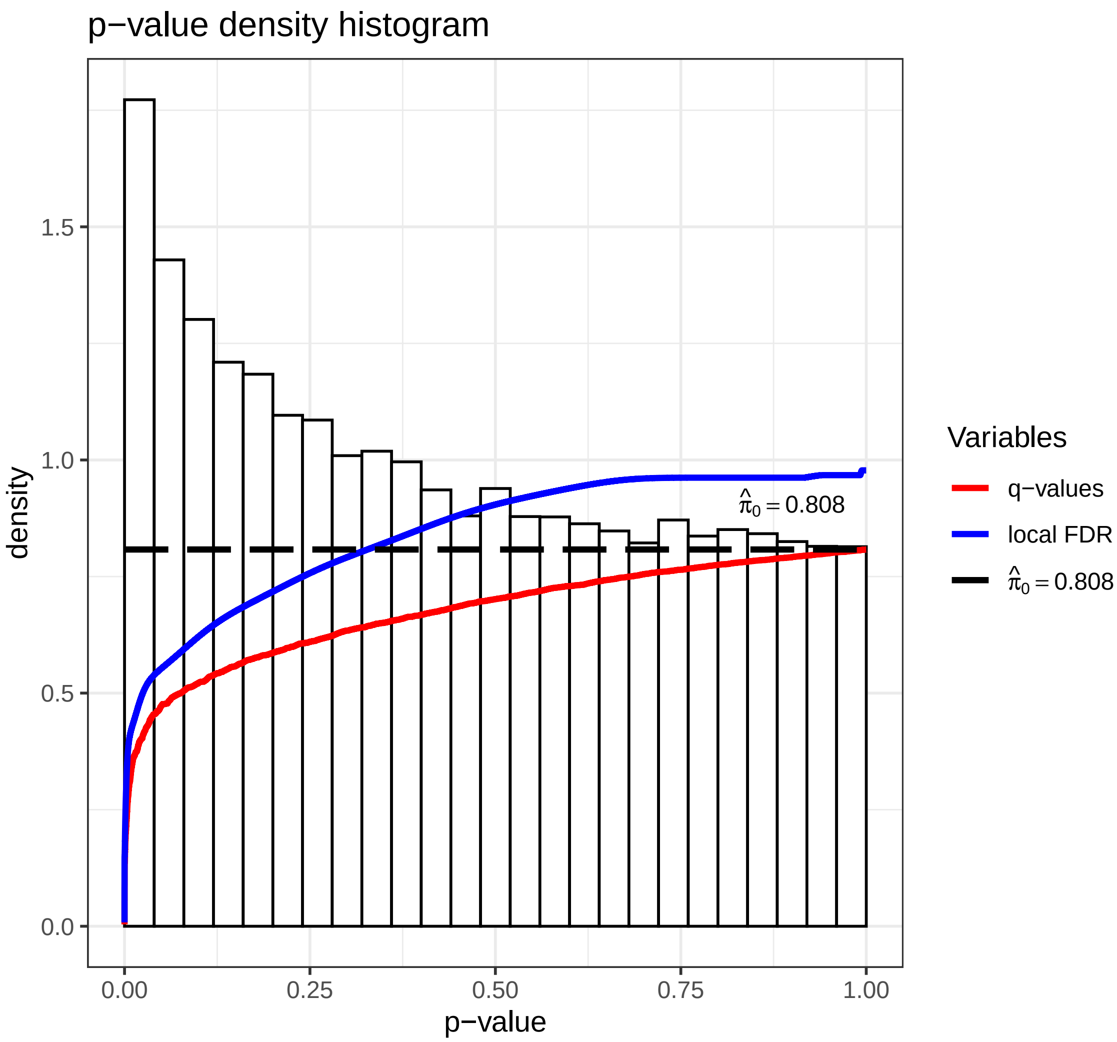
It can be shown that when the regularization can be represented by a prior distribution on $\boldsymbol{B}$ then this algorithm achieves the MAP.

Example: Kidney Expr by Age

In Storey et al. (2005), we considered a study where kidney samples were obtained on individuals across a range of ages. The goal was to identify genes with expression associated with age.

```
> library(edge)
> library(splines)
> load("./data/kidney.RData")
> age <- kidcov$age
> sex <- kidcov$sex
> dim(kidexpr)
[1] 34061    72
> cov <- data.frame(sex = sex, age = age)
> null_model <- ~sex
> full_model <- ~sex + ns(age, df = 3)
```

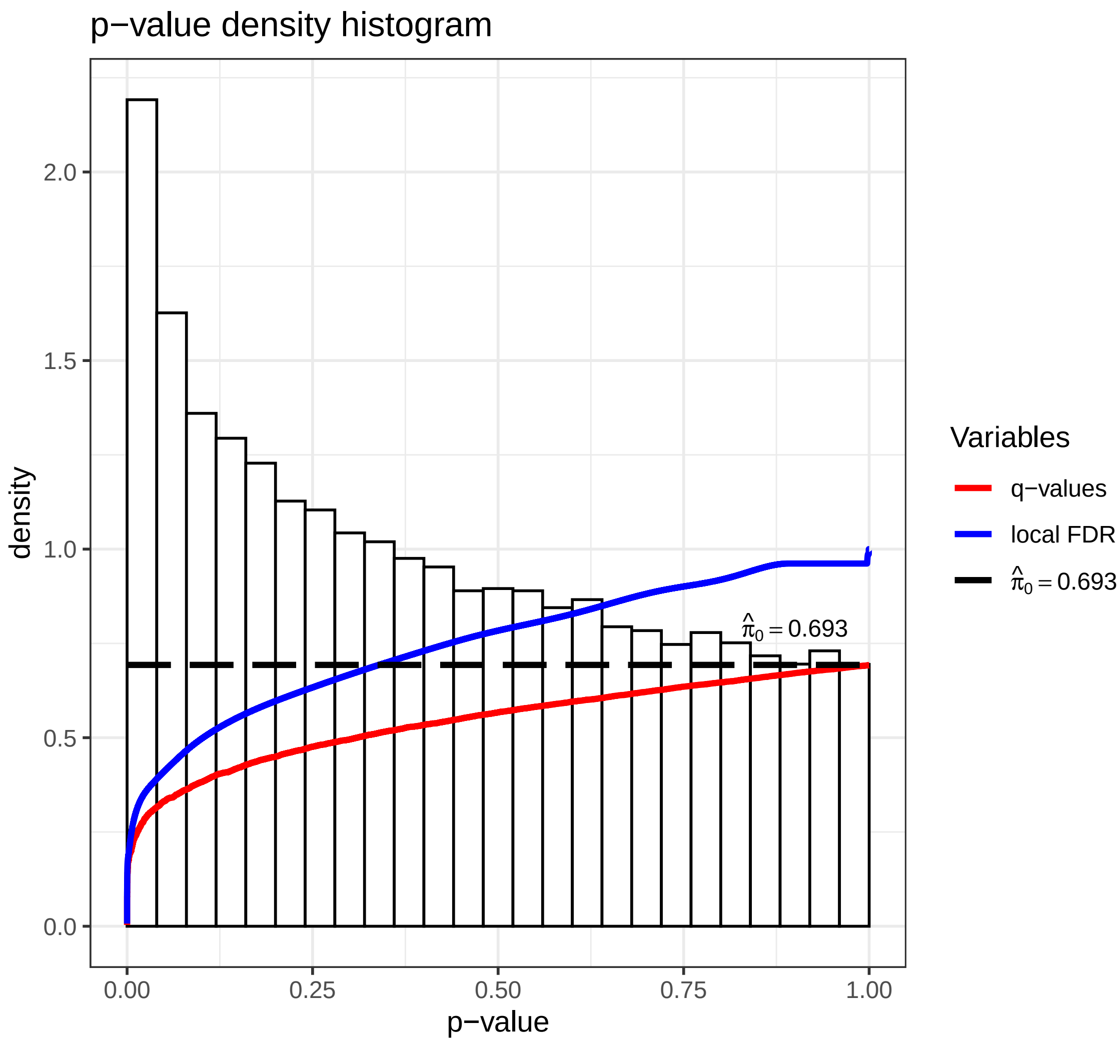
```
> de_obj <- build_models(data = kidexpr, cov = cov,
+                         null.model = null_model,
+                         full.model = full_model)
> de_lrt <- lrt(de_obj, nullDistn = "bootstrap", bs.its = 100, verbose=FALSE)
> qobj1 <- qvalueObj(de_lrt)
> hist(qobj1)
```



Now that we have completed a standard generalized LRT, let's estimate $\boldsymbol{Z}$ (the surrogate variables) using the `sva` package as accessed via the `edge` package.

```
> dim(nullMatrix(de_obj))
[1] 72  2
> de_sva <- apply_sva(de_obj, n.sv=4, method="irw", B=10)
Number of significant surrogate variables is:  4
Iteration (out of 10 ):1  2  3  4  5  6  7  8  9  10
> dim(nullMatrix(de_sva))
[1] 72  6
> de_svalrt <- lrt(de_sva, nullDistn = "bootstrap", bs.its = 100, verbose=FALSE)
```

```
> qobj2 <- qvalueObj(de_svalrt)
> hist(qobj2)
```



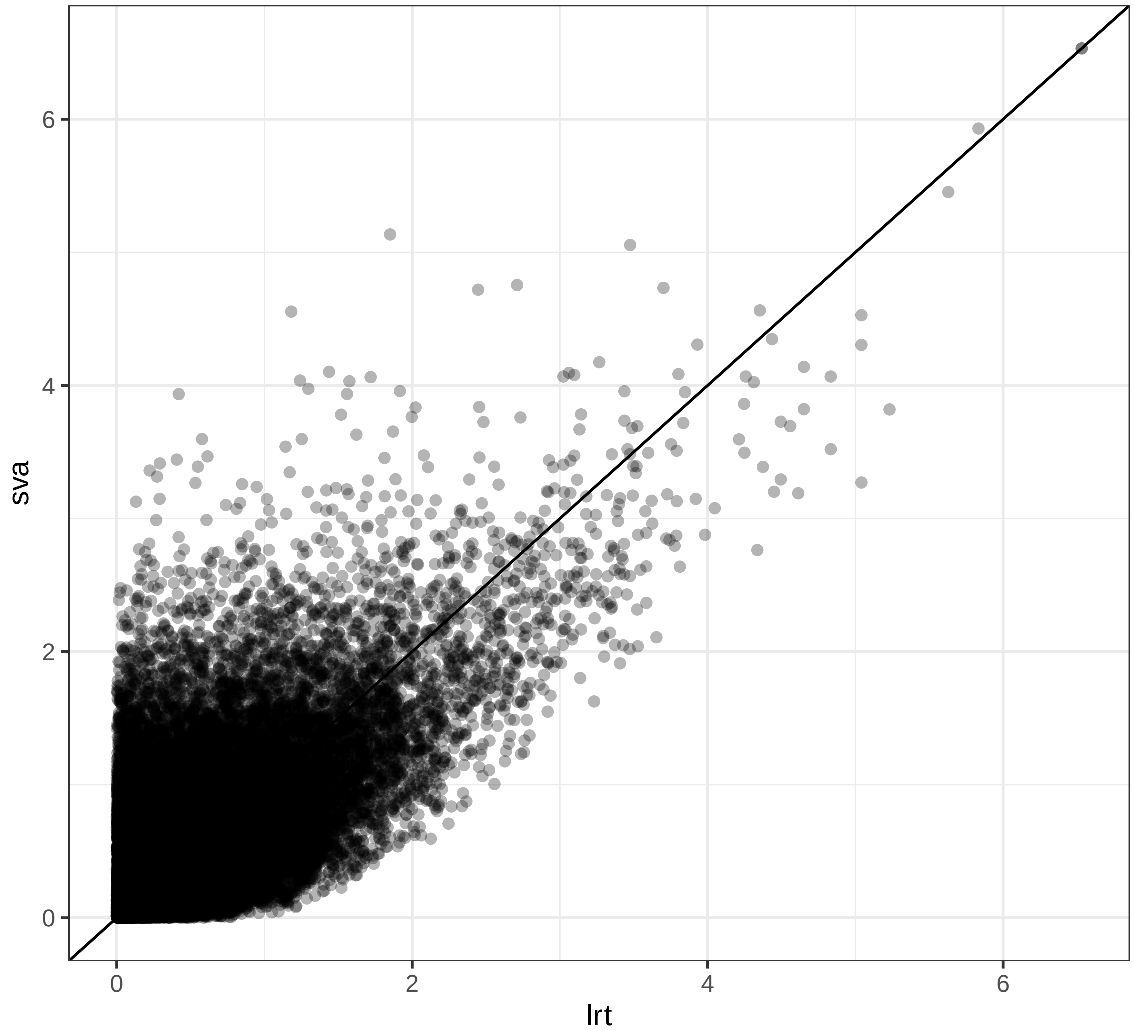

```
> summary(qobj1)

Call:
qvalue(p = pval)

pi0:     0.8081212

Cumulative number of significant calls:

          <1e-04 <0.001 <0.01 <0.025 <0.05 <0.1    <1
p-value       27    161   798   1676  2906 5271 34061
q-value        0      0     2      4    10   27 34061
local FDR      0      0     2      2     5   18 34061
```

```
> summary(qobj2)

Call:
qvalue(p = pval)

pi0:     0.6925105
```

```
Cumulative number of significant calls:

          <1e-04 <0.001 <0.01 <0.025 <0.05 <0.1    <1
p-value       28    151  1001   2051  3549 6168 34061
q-value        0      0     3      4     6   51 34061
local FDR      0      0     2      2     3   28 34053
```

P-values from two analyses are fairly different.

```
> data.frame(lrt=-log10(qobj1$pval), sva=-log10(qobj2$pval)) %>%
+   ggplot() + geom_point(aes(x=lrt, y=sva), alpha=0.3) + geom_abline()
```



Causality

Acknowledgement

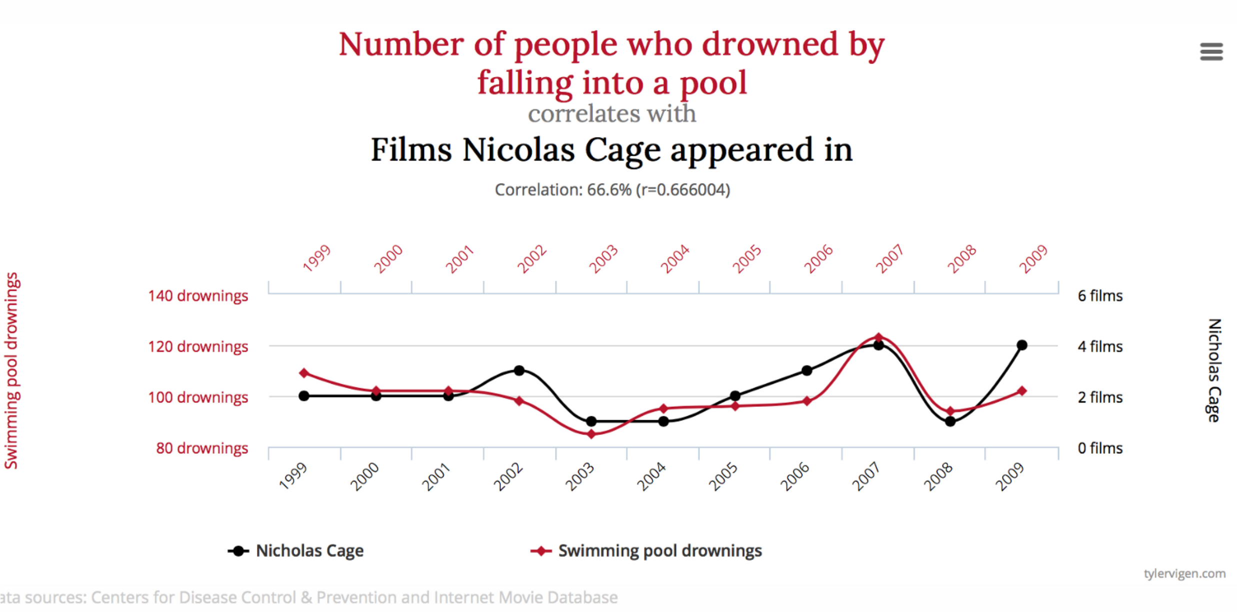
These section is partially based on slides by Irineo Cabreros.

Definition

Informally, we might say X is causal for Y if a change in X influences a change in Y.

However, a formal, statistically rigorous definition is challenging and controversial. We will consider one such framework here, called the **potential outcomes** framework.

Correlation Is Not Causation



From http://tylervigen.com/spurious-correlations.

Reasons For Nonzero Correlation

- Spurious correlation: $\text{Cor}(X, Y) = 0$, however observed $r_{x,y} \neq 0$
- X causes Y: $X \rightarrow Y$
- Y causes X: $Y \rightarrow X$
- X and Y are confounded by Z: $X \leftarrow Z \rightarrow Y$

Potential Outcomes

For each observed unit, four random variables are drawn:

$$(X, Y_0, Y_1, Y)$$

X and Y are observed, Y_0 and Y_1 are **potential outcomes**.

These random variables are related to each other:

$$Y = Y_0 1(X = 0) + Y_1 1(X = 1)$$

Causal Quantities of Interest

Causal effect (CE):

$$\text{CE} = Y_1 - Y_0$$

Average (expected) causal effect (ACE):

$$\text{ACE} = \text{E}[Y_1] - \text{E}[Y_0]$$

Estimable Quantities

"Regression" of Y on X in statistics often refers to modeling $\text{E}[Y|X]$.

Regression effect (RE):

$$\text{RE} = [Y|X = 1] - [Y|X = 0]$$

Average regression effect (ARE):

$$\text{ECE} = \text{E}[Y|X = 1] - \text{E}[Y|X = 0]$$

These are *not* causal quantities.

Causal Inference: Fundamental Challenge

Suppose the following five configurations are equally likley.

X	Y_0	Y_1	Y
0	0	0	0
1	0	1	1
1	1	1	1
0	1	1	1
0	1	0	1

$\text{ACE} = \frac{1}{5}(0+1+1+1+0) - \frac{1}{5}(0+0+1+1+1) = 0$

$\text{ARE} = \frac{1}{2}(1+1) - \frac{1}{3}(0+1+1) = \frac{1}{3}$

Randomization

From Greenberg (2018) *The Omega Principle: Seafood and the Quest for a Long Life and a Healthier Planet*:

"In 1747 the physician James Lind, sailing aboard a British warship, divided a group of 12 sailors suffering from scurvy into six groups of two. All ate the same diet, but each pair was given a different supplemental potion: one pair got a quart of cider, another an elixir of sulfuric acid, another six spoonfuls of vineger, another a pint of seawater, still another a spicy paste together with barley water, and finally the lucky last—two oranges and a lemon."

ACE Equals ARE Under Randomization

Suppose X is decided by a physical coin toss, which can be assumed independent of all potential outcome random variables.

$$
\begin{aligned}
\text{ARE} &\equiv \text{E}[Y|X=1] - \text{E}[Y|X=0] \\
&= \text{E}[Y_0 1(X=0) + Y_1 1(X=1)|X=1] - \\
&\qquad \text{E}[Y_0 1(X=0) + Y_1 1(X=1)|X=0] \\
&= \text{E}[Y_1|X=1] - \text{E}[Y_0|X=0] \\
&= \text{E}[Y_1] - \text{E}[Y_0] \\
&= \text{ACE}
\end{aligned}
$$

Under this set-up, it can be shown that $\text{Cor}(X, Y) \neq 0$ implies $\text{E}[Y|X=1] - \text{E}[Y|X=0] \neq 0$.

So in this case and with randomization of X, it follows that a non-zero population correlation implies X is causal for Y under the potential outcomes model.

Summary of QCB 408 / 508

What Did We Do?

- Utilized R
- Random variables
- Probability models
- Likelihood based inference: frequentist and Bayesian
- Specialized frequentist inference
- Numerical methods for inference
- Statistical modeling
- High-dimensional inference and modeling
- Causality

R

Advanced R, Wickham

R Packages, Wickham

Introductory Statistics with R, Dalgaard

R Cookbook, Teetor

Visualization

R Graphics Cookbook, Chang

Visualizing Data, Cleveland

The Visual Display of Quantitative Information, Tufte

Modeling

Statistical Models: Theory and Practice, Freedman

Nonparametric Regression and Generalized Linear Models: A roughness penalty approach, Green and Silverman

Bayesian Data Analysis, Gelman et al.

Statistical Inference

All of Statistics, Wasserman

Statistical Inference, Casella and Berger

An Introduction to the Bootstrap, Efron and Tibshirani

A First Course in Bayesian Statistical Methods, Hoff

Machine Learning

An Introduction to Statistical Learning: with Applications in R, James et al.

Elements of Statistical Learning, Hastie, Tibshirani, and Friedman

Machine Learning: A Probabilistic Perspective, Murphy

Pattern Recognition and Machine Learning, Bishop

Extras

Source

License

Source Code

Session Information

```
> sessionInfo()
R version 3.6.0 (2019-04-26)
Platform: x86_64-apple-darwin15.6.0 (64-bit)
Running under: macOS  10.15.3

Matrix products: default
BLAS:   /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods
[7] base

other attached packages:
 [1] jackstraw_1.3   qvalue_2.15.0   MASS_7.3-51.5
 [4] broom_0.5.2     forcats_0.5.0   stringr_1.4.0
 [7] dplyr_0.8.4     purrr_0.3.3     readr_1.3.1
[10] tidyr_1.0.2     tibble_2.1.3    ggplot2_3.2.1
[13] tidyverse_1.3.0 knitr_1.28

loaded via a namespace (and not attached):
 [1] rsvd_1.0.3       Rcpp_1.0.3       lfa_1.12.0
 [4] lubridate_1.7.4  lattice_0.20-40  corpcor_1.6.9
 [7] gtools_3.8.1     assertthat_0.2.1 digest_0.6.25
```

```
[10] gmp_0.5-13.6     R6_2.4.1         cellranger_1.1.0
[13] plyr_1.8.5       backports_1.1.5  reprex_0.3.0
[16] evaluate_0.14    httr_1.4.1       pillar_1.4.3
[19] rlang_0.4.5      lazyeval_0.2.2   readxl_1.3.1
[22] irlba_2.3.3      rstudioapi_0.11  Matrix_1.2-18
[25] rmarkdown_2.1    labeling_0.3     splines_3.6.0
[28] ClusterR_1.2.1   munsell_0.5.0    compiler_3.6.0
[31] modelr_0.1.6     xfun_0.12        pkgconfig_2.0.3
[34] htmltools_0.4.0  tidyselect_1.0.0 fansi_0.4.1
[37] crayon_1.3.4     dbplyr_1.4.2     withr_2.1.2
[40] grid_3.6.0       nlme_3.1-144     jsonlite_1.6.1
[43] gtable_0.3.0     lifecycle_0.1.0  DBI_1.1.0
[46] magrittr_1.5     scales_1.1.0     cli_2.0.2
[49] stringi_1.4.6    farver_2.0.3     reshape2_1.4.3
[52] fs_1.3.1         xml2_1.2.2       generics_0.0.2
[55] vctrs_0.2.3      tools_3.6.0      glue_1.3.1
[58] hms_0.5.3        parallel_3.6.0   yaml_2.2.1
[61] colorspace_1.4-1 cluster_2.1.0    rvest_0.3.5
[64] haven_2.2.0
```