Z-score estimation of summary statistics based on LD

2017/06/05

Yosuke Tanigawa (ytanigaw@stanford.edu)
M. Rivas Lab (rotation student) | Biomedical Informatics Ph.D. Program

Model

Notation

- $X \in \mathcal{R}^{N \times M}$: genotype. We assume X is normalized such that is has zero-mean and unit variance.
- $Y \in \mathcal{R}^{N \times 1}$: traits. We assume Y is normalized such that is has zero-mean and unit variance.
- $\beta \in \mathcal{R}^{M}$: effect size.
- N: number of individuals.
- M: number of SNP markers.
- $V \in \mathcal{R}^{M \times M}$: LD matrix. This can be found by $V = X^T X$.

Model

Linear regression model

Our regression model is:

$$Y = X\beta + \varepsilon, \quad \varepsilon \sim N(0, I) \tag{1}$$

Least square and marginal effect model

Least square

$$\nabla_\beta (Y - X\beta)^T (Y - X\beta) = -X^T (Y - X\beta) = 0 \tag{2}$$

$$\hat{\beta} = (X^T X)^{-1} X^T Y = V^{-1} X^T Y; \quad \text{Var}\left[\hat{\beta}\right] = \sigma_j^2 V^{-1} \tag{3}$$

where, σ_j^2 is residual variance.

Marginal effect

$$\hat{\beta}_M = D^{-1} X^T Y; \quad \text{Var}\left[\hat{\beta}_M\right] = \sigma_M^2 D^{-1} \tag{4}$$

where, D is the diagonal matrix of V.

The relationship between two models

Since we have $V\hat{\beta} = X^T Y = D\hat{\beta}_M$, we have

$$\hat{\beta} = V^{-1} D \hat{\beta}_M \tag{5}$$

Z-score

We define z-score:

$$Z := \frac{\hat{\beta}_M}{\sqrt{\mathrm{Var}\left[\hat{\beta}_M\right]}} = \frac{X^T Y}{\sqrt{N}} \tag{6}$$

We assume

$$Z \sim N(0, V) \tag{7}$$

Imputation of Z-scores

Let's consider to divide Z into two blocks:

1. Z_t: Z-score for typed SNPs
2. Z_i: Z-score for untyped SNPs

i.e.

$$Z^T = (Z_t^T \quad Z_i^T); \quad V = \begin{pmatrix} V_{tt} & V_{ti} \\ V_{it} & V_{ii} \end{pmatrix} \tag{8}$$

Since we modeled the Z-scores as multi-variate normal, the conditional distribution $p(Z_i \mid Z_t)$ is also normal:

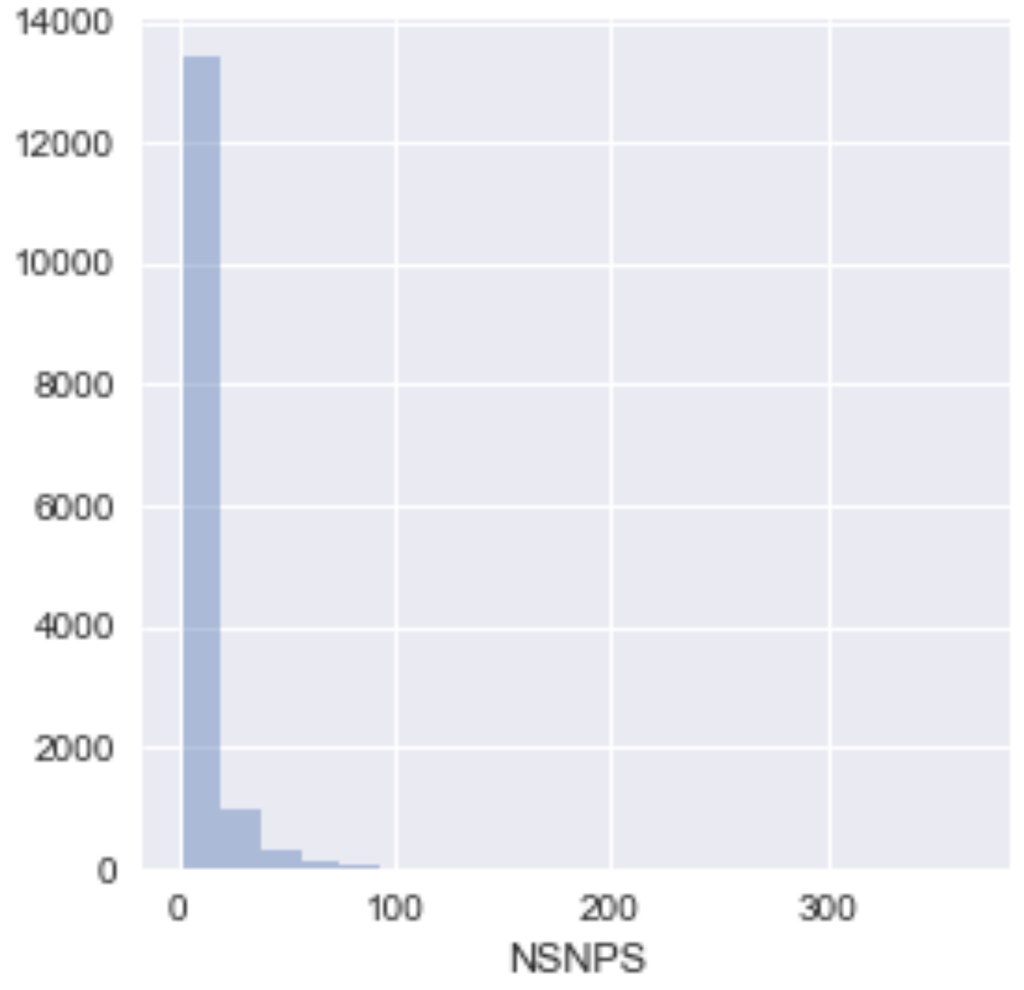
$$Z_i \mid Z_t \sim N(V_{it} Vtt^{-1} Z_t, V_{ii} - VitV_{tt}^{-1} V_{ti}) \tag{9}$$

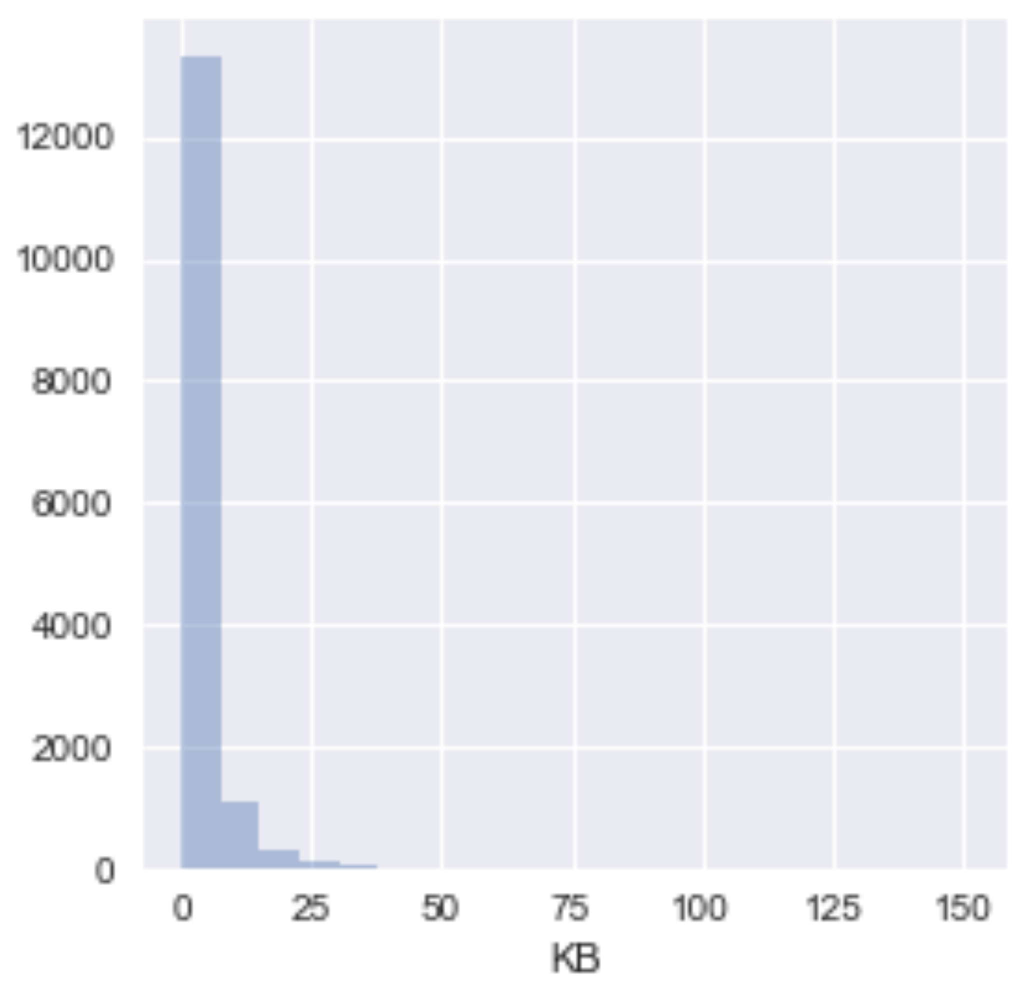
Dataset description

- Genotype info:
  - UKBB (with population stratification): 112,338 individuals
  - Focusing on chromosome 20
- LD block (plink)
  - --blocks no-pheno-req
  - --blocks-max-kb 1000
  - --blocks-min-maf .05
- GWAS summary statistics
  - ADD, age, sex, C1-C4 (first 4 components)
  - Focusing on ADD (additive effects)

LD block structure on chromosome 20

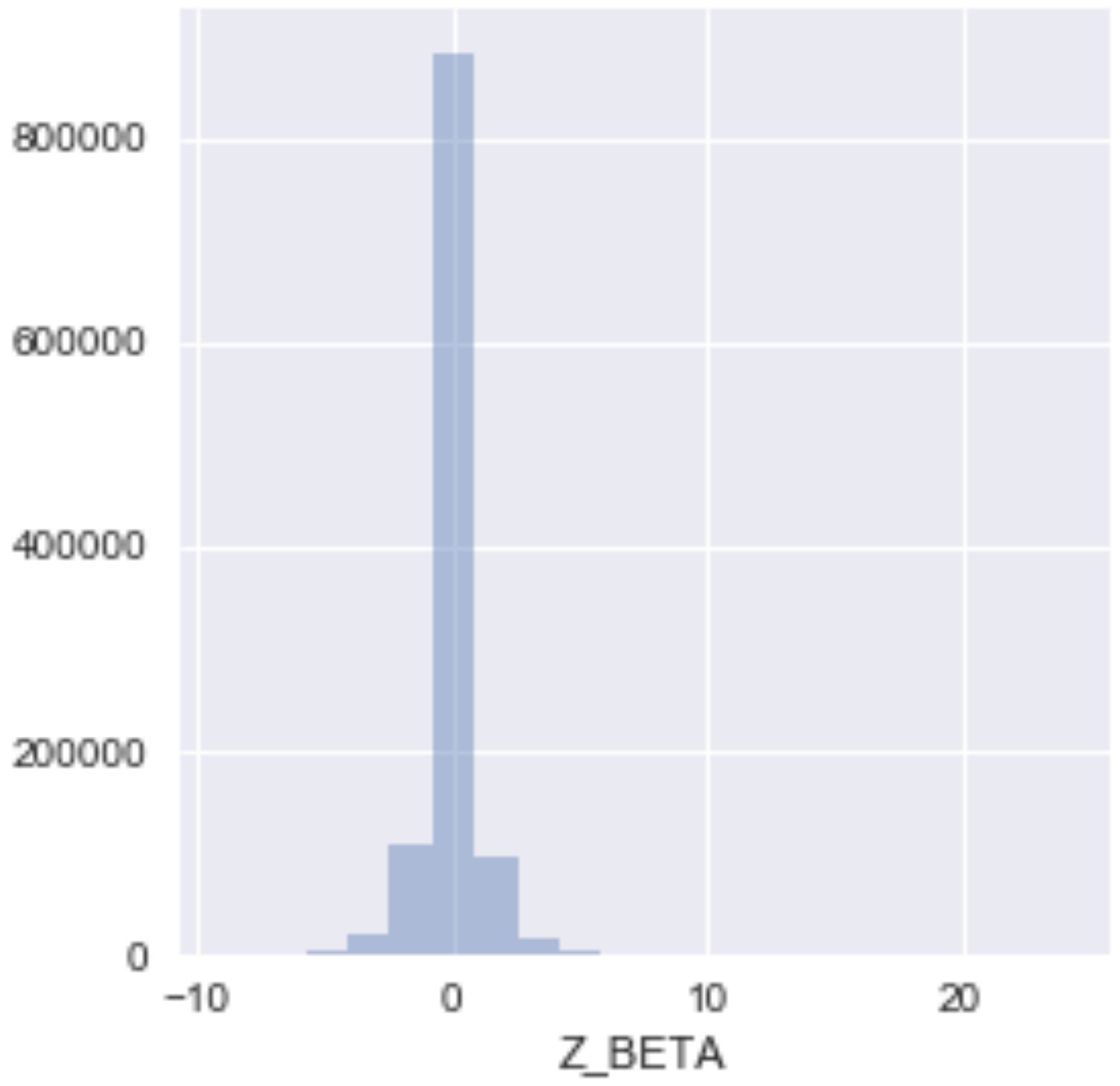
- (left) Number of SNPs in a LD block (median 4.0)
  - Note: MAF 5%
- (right) Size of LD block (median 1.1865)

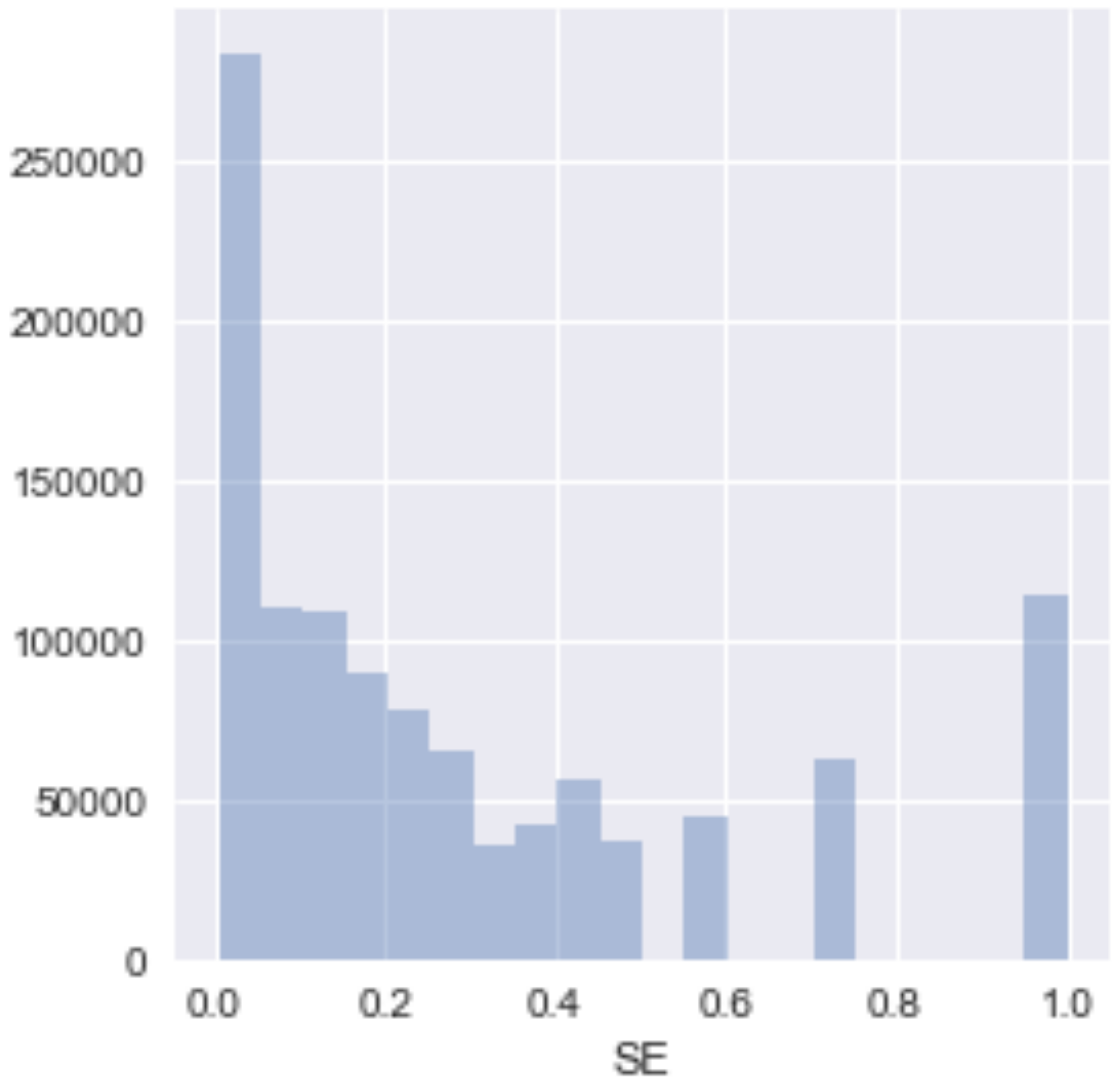




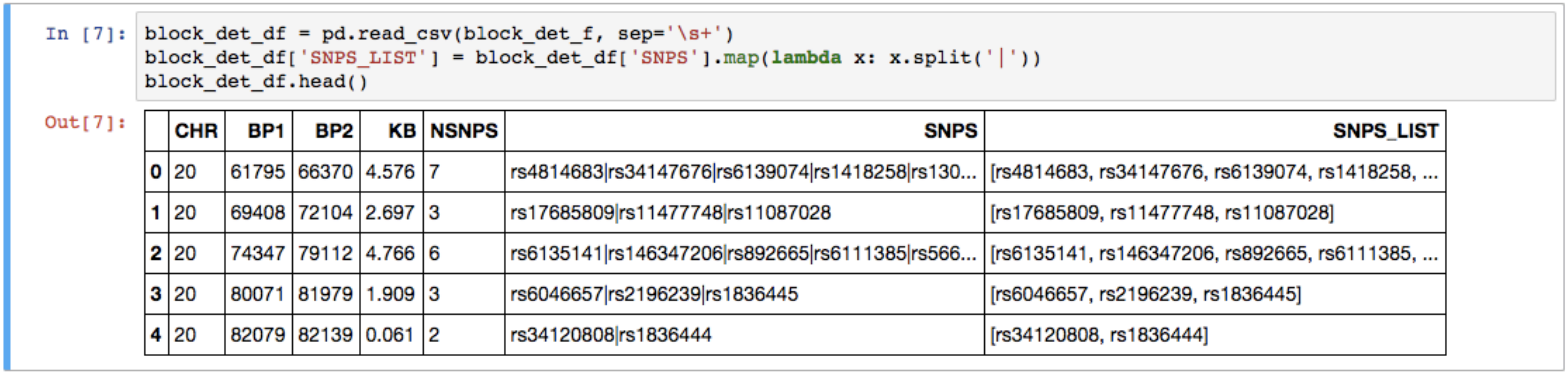
Z-score distribution

- Zero-mean and unit-variance normalization for Z-score





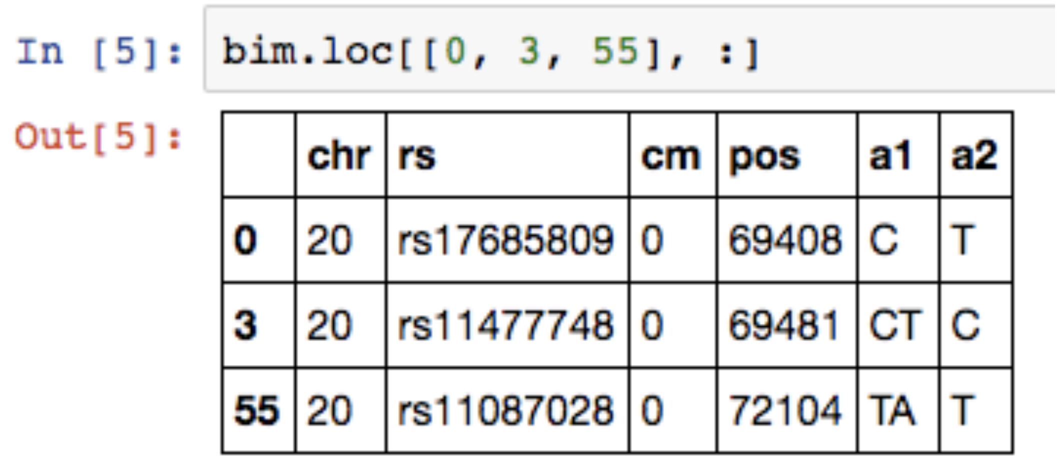
Example LD block: chr20:69408-72104



In [7]:
```
block_det_df = pd.read_csv(block_det_f, sep='\s+')
block_det_df['SNPS_LIST'] = block_det_df['SNPS'].map(lambda x: x.split('|'))
block_det_df.head()
```

Out[7]:

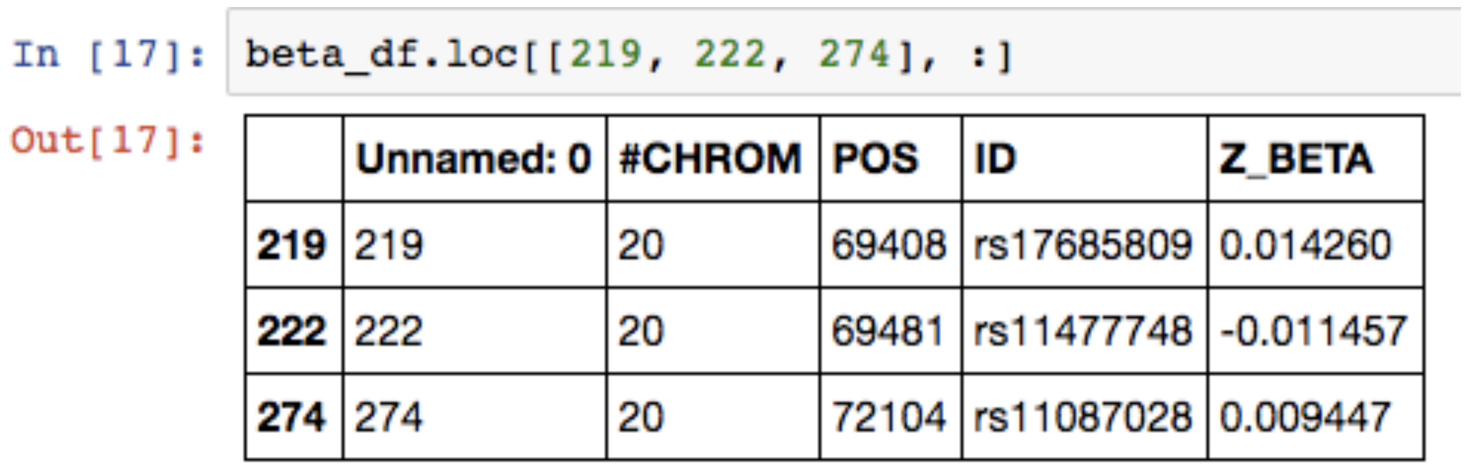
	CHR	BP1	BP2	KB	NSNPS	SNPS	SNPS_LIST
0	20	61795	66370	4.576	7	rs4814683|rs34147676|rs6139074|rs1418258|rs130...	[rs4814683, rs34147676, rs6139074, rs1418258, ...
1	20	69408	72104	2.697	3	rs17685809|rs11477748|rs11087028	[rs17685809, rs11477748, rs11087028]
2	20	74347	79112	4.766	6	rs6135141|rs146347206|rs892665|rs6111385|rs566...	[rs6135141, rs146347206, rs892665, rs6111385, ...
3	20	80071	81979	1.909	3	rs6046657|rs2196239|rs1836445	[rs6046657, rs2196239, rs1836445]
4	20	82079	82139	0.061	2	rs34120808|rs1836444	[rs34120808, rs1836444]



In [5]:
```
bim.loc[[0, 3, 55], :]
```

Out[5]:

	chr	rs	cm	pos	a1	a2
0	20	rs17685809	0	69408	C	T
3	20	rs11477748	0	69481	CT	C
55	20	rs11087028	0	72104	TA	T

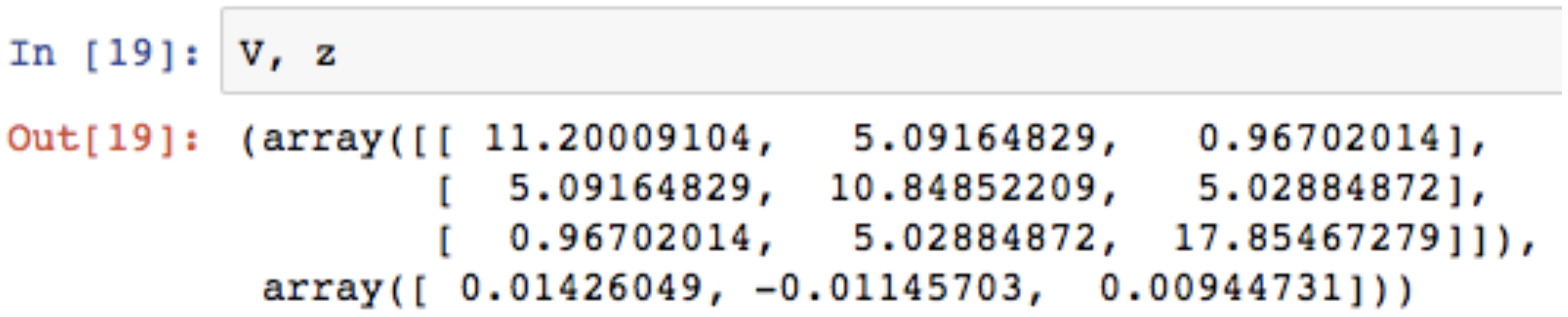


In [17]:
```
beta_df.loc[[219, 222, 274], :]
```

Out[17]:

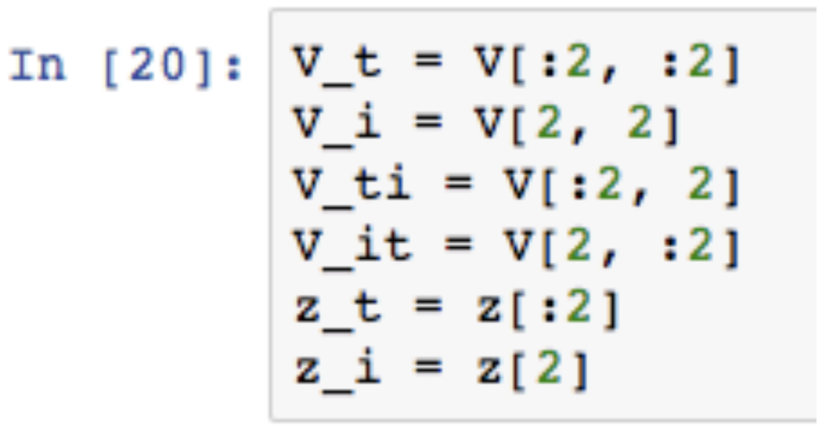
	Unnamed: 0	#CHROM	POS	ID	Z_BETA
219	219	20	69408	rs17685809	0.014260
222	222	20	69481	rs11477748	-0.011457
274	274	20	72104	rs11087028	0.009447

Example of Z-score imputation
2 typed SNPs + 1 untyped SNP in the LD block

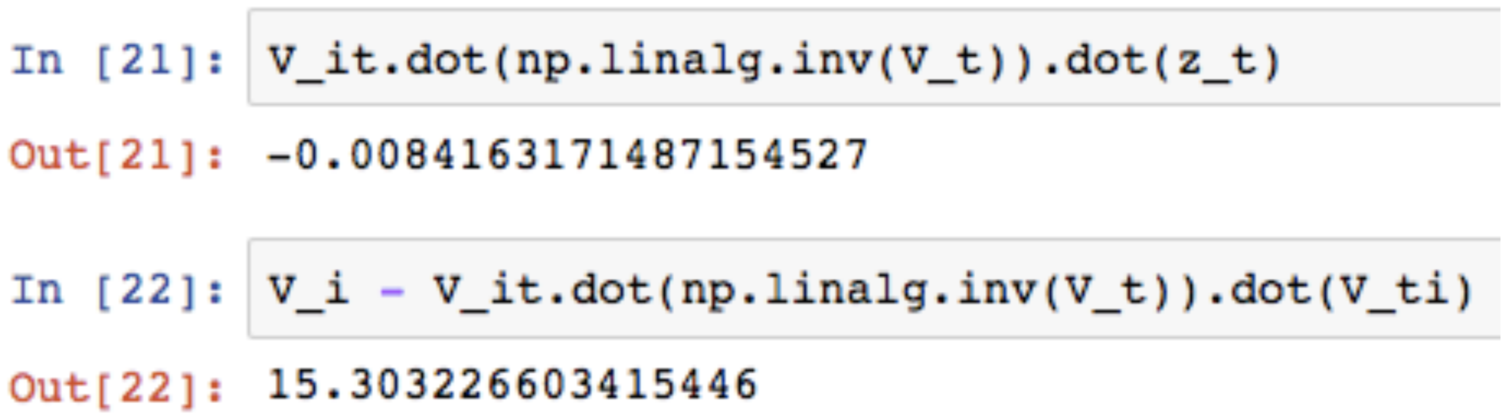


```
In [19]: V, z

Out[19]: (array([[ 11.20009104,   5.09164829,   0.96702014],
                 [  5.09164829,  10.84852209,   5.02884872],
                 [  0.96702014,   5.02884872,  17.85467279]]),
          array([ 0.01426049, -0.01145703,  0.00944731]))
```



```
In [20]: V_t = V[:2, :2]
         V_i = V[2, 2]
         V_ti = V[:2, 2]
         V_it = V[2, :2]
         z_t = z[:2]
         z_i = z[2]
```



```
In [21]: V_it.dot(np.linalg.inv(V_t)).dot(z_t)

Out[21]: -0.0084163171487154527

In [22]: V_i - V_it.dot(np.linalg.inv(V_t)).dot(V_ti)

Out[22]: 15.303226603415446
```

Current approach is not scalable

- plink --r2 does not provide full output for LD matrix
  - --ld-window-r2 does not work
- We need to normalize X (genotype)
- Current platform: pgenlib + python
  - Accessing on the raw data

- External validation set?