# Self-test
# School on Methods of Data Analysis

winter 2019/2020, Myanmar
Last updated: June 18, 2019

Lecturers:
Yossi Ashkenazy (Ben Gurion University), Hezi Gildor (The Hebrew University)
Eli Tziperman (Harvard University), Peter Huybers (Harvard University)

The school is intended for advanced graduate students and postdoctoral researchers from physics, math, earth sciences, engineering, etc. Topics include Probability and statistics; Time series and Fourier analysis; Regression analysis; Principal Component Analysis; Clustering; Classification and Neural networks.

This collection of review problems covers a few subjects you should have seen in previous courses and which will be used in this school, allowing you to decide if you are prepared for this school. Please show all steps in all calculations explicitly. See relevant python commands at end of this document[1]

Scan your solution to a pdf file (e.g., using https://www.thegrizzlylabs.com/genius-scan) and upload with your winter school application form.

1. **Scalars, vectors and matrices:** transpose, addition and multiplication. In this question and following ones, denote

$$\mathsf{A}=\begin{pmatrix}-2 & 3 & -5\\ 3 & 1 & 3\\ -2 & 2 & 3\end{pmatrix},\ \mathsf{B}=\begin{pmatrix}-1+2i & -6.5+1i & -3\\ 2 & 7+6i & 3+3i\\ 2 & -9 & -5\end{pmatrix},\ \mathsf{P}=\begin{pmatrix}-2 & 3\\ 2 & 1\end{pmatrix},$$
$$\mathbf{a}=\begin{pmatrix}2\\ 3-i\\ -1\end{pmatrix},\ \mathbf{b}=\begin{pmatrix}2\\ -1\\ 3\end{pmatrix},\ z_1=-2+3i,\ z_2=5-3i.$$

   Calculate and write the matrix product of two matrices $\mathsf{A}^T\mathsf{A}$, product of a matrix and a vector $\mathsf{A}\mathbf{b}$, scalar products of two vectors $\mathbf{a}^T\mathbf{b}$, $\mathbf{a}^\dagger\mathbf{a}$, $\mathbf{b}^T\mathbf{b}$, and the vector norm $|\mathbf{b}|$, where $^T$ and $^\dagger$ represent transpose and conjugate transpose, correspondingly.

2. **Linear equations, Gaussian elimination & back substitution:** solve $\mathsf{A}\mathbf{x}=\mathbf{b}$ using Gaussian elimination to bring the equation to an upper triangular form $\mathsf{A}'\mathbf{x}=\mathbf{b}'$,

$$\begin{pmatrix}a'_{11} & a'_{12} & a'_{13}\\ 0 & a'_{22} & a'_{23}\\ 0 & 0 & a'_{33}\end{pmatrix}\begin{pmatrix}x_1\\ x_2\\ x_3\end{pmatrix}=\begin{pmatrix}b'_1\\ b'_2\\ b'_3\end{pmatrix}$$

   and then using back substitution to find all elements of $\mathbf{x}$.

3. **Determinants, linear independence of vectors:** (i) Calculate $\det(\mathsf{A})$ using the cofactors of $A$. (ii) Show that $\det(\mathsf{B})=0$ in python using `np.linalg.det(B)`, conclude that the of columns of $B$ are linearly dependent and show explicitly that one column may be expressed as a linear combination of the other two.

4. **Matrix inversion, regular vs singular matrices:** calculate $\mathsf{A}^{-1}$ (A) Using the matrix of cofactors, (B) by performing row operations in parallel on $\mathsf{A}$ and on the identity matrix. Show explicitly that $\mathsf{A}^{-1}\mathsf{A} = \mathsf{I}$. Does $\mathsf{B}^{-1}$ exist? Why?

5. **Complex numbers:** calculate the following quantities,

   (a) $z_1 z_2$, $z_1^\dagger z_2$, (b) $|z_1| - |z_2|$,
   (c) $(e^{2z_1})^\dagger - 2e^{2z_2}$, $e^{2z_1}2e^{2z_2}$, (d) Write $z_1 = re^{i\theta}$ and find $r, \theta$.

6. **Homogeneous linear equations:** (i) What conditions does a matrix $\mathsf{A}$ need to satisfy for there to be a nonzero vector $\mathbf{x}$ for which $\mathsf{A}\mathbf{x} = \mathbf{0}$? (ii) Find the general solution $\mathbf{x}$ to $\mathsf{B}\mathbf{x} = \mathbf{0}$ for the matrix defined above.

7. **Eigenvalues and eigenvectors:** (i) Write the characteristic equation of $\mathsf{P}$, $\det(\mathsf{P} - \lambda\mathsf{I}) = 0$, and solve it for the eigenvalues $\lambda_i$. Calculate the eigenvectors $\mathbf{e}_i$ (by hand) by solving $\mathsf{P}\mathbf{e}_i = \lambda_i \mathbf{e}_i$ and requiring that $|\mathbf{e}_i|^2 = e_{i,1}^2 + e_{i,2}^2 = 1$ where $e_{i,1}$ is the first element of the eigenvector. (ii) Find the eigenvalues and eigenvectors of $\mathsf{B}$ in python using `D,V=scipy.linalg.eig(B)`, verify by hand-calculation that the first eigenvector (first column of $V$, `V[:,0]` in python) and eigenvalue (first element of `diag(D)`) indeed satisfy $\mathsf{B}\mathbf{e}_1 = \lambda_1 \mathbf{e}_1$. (iii) Find the eigenvalues of $\mathsf{A}$ using `\verb`D,V=linalg.eig(A);—, and then solve by hand for its eigenvector that corresponds to the eigenvalue that is largest in absolute value.

8. **Matrix diagonalization:** Let $\mathsf{U}$ be a matrix whose columns are the eigenvectors of $\mathsf{P}$ calculated above, calculate $\mathsf{U}^{-1}\mathsf{P}\mathsf{U}$ and show it is diagonal and that the values along the diagonal are the eigenvalues of $\mathsf{P}$ also calculated above.

9. **Gram-Schmidt orthogonalization:** starting from the three columns of $\mathsf{A}$ taken as vectors $(\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$, apply the Gram-Schmidt orthogonalization process to find three orthonormal vectors $(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3)$ satisfying $\mathbf{y}_i^T \mathbf{y}_j = \delta_{ij}$ where $\delta_{ij}$ is the Kronecker delta.

10. **Gradient:** Given a column vector $\mathbf{x} = (x, y)^T$, (i) write the expression for $J(x, y) = \mathbf{x}^T(\mathsf{P}^T\mathsf{P})\mathbf{x}$. (ii) Write the gradient of this function, $\nabla J(x, y)$. Evaluate the gradient at the point $\mathbf{x}_0 = (x, y) = (1, -2)$. (iii) What do the direction and magnitude of the gradient represent in terms of the geometric properties of $J(x, y)$ as a function of two variables? (iv) Calculate the two numbers $J(\mathbf{x}) \pm \delta\mathbf{x} \cdot \nabla J$, where $\delta\mathbf{x}$ is a small vector in the direction of $\nabla J$, $\delta\mathbf{x} = 0.001\nabla J$, all evaluated at $\mathbf{x} = \mathbf{x}_0$; explain the results.

# Python commands

1 The python commands below assume you have installed anaconda python version 3.7, started the spyder interface. First give the following commands: `import numpy as np; from numpy import linalg; import scipy as scipy; import matplotlib.pyplot as plt; import matplotlib;` and then follow the instructions below.

You can input matrix $A$, column vector $b$ and row vector $c$ into python in the form
`A=np.array([[a11,a12,a13],[a21,a22,a23],[a31,a32,a33]];`
`b=np.array([[b1],[b2],[b3]]); c=np.array([[c1,c2,c3]]);`.