SF1930 Projekt

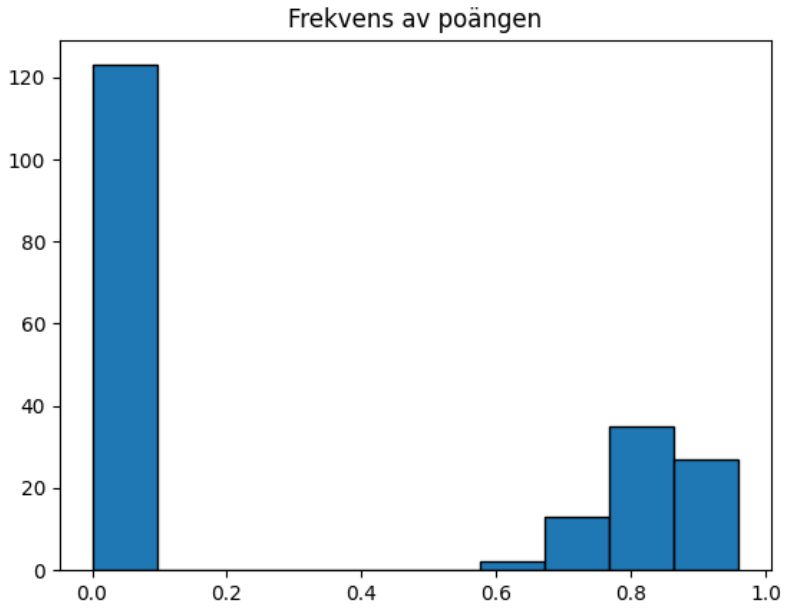
Jim Ogenstad

11 oktober 2024

Uppgift 1. Första insikterna

Uppgift (a)

Med python kan vi generera följande histogram:



Vi ser att en stor andel tävlande fick noll poäng medan resten fick värden runt omkring 0.8.

Uppgift (b)

Vi vill skatta en bernoulli fördelad variabel. Låt oss då först göra det allmänt för att sedan kunna implementera det i python. Vi har täthetsfunktionen för en bernoulli fördelad stokastisk variabel:

$$P_{\mathbf{X}}(x) = \begin{cases} p & x = 1 \\ 1 - p & x = 0 \end{cases}$$

Vi får då likelihoodfunktionen givet lite data $x_1, \ldots, x_n$ som $L(p|x_1, \ldots, x_n) = f_{\mathbf{X}}(x_1, \ldots, x_n|p)$. Eftersom alla trick antas vara oberoende så har vi att den simultana täthetsfunktionen bara är produkten av alla enskilda täthetsfunktioner (där alla är likadana dessutom). Vi har $f_{\mathbf{X}}(x_1, \ldots, x_n|p) = \Pi_{i=1}^{n} f_{X_i}(x_i)$. Om vi vet hur många x_i som är 1 och hur många som är (0), vilket vi gör, då kan vi skriva detta som $p^k(1-p)^{n-k}$ där $1 \leq k \leq n$ är antalet lyckade trick. Vi tar logaritmen av funktionen vilket ger $k\log(p) + (n-k)\log(1-p)$. Vi kan derivera detta och få $k/p - (n-k)/(1-p)$. För att vi ska ha en extrempunkt så måste denna derivata vara noll. Vi får då att $k(1-p) = p(n-k) \iff p = k/n$. Detta blir vår skattning. Vi vet att det är en maxpunkt då funktionsvärden till vänster som $p = 0$ samt till höger som $p = 1$ båda är lägre. För ett stickprov av en bernoulli fördelad variabel kan vi alltså skatta p som k/n. Jag har gjort dessa beräkningar i python och printat resultaten. Här kommer svaren nedskrivna i en tabell:

Namn	**Skatting av p**
Decenzo	0.4375
Foy	0.5
Gustavo	0.4
Hoban	0.4
Hoefler	0.4375
Jordan	0.4
Midler	0.3333
Mota	0.25
Oliveira	0.4167
O'neill	0.25
Papa	0.4375
Ribeiro C	0.25
Shirai	0.4

Uppgift (c)

Väntevärdet av en bernoullifördelad variabel är ju bara chansen att lyckas, dvs p. För att bestämma vilka som har bäst chans att lyckas med ett givet trick så ser vi bara på de högsta tabellvärdena. Högst p har Foy och efter det har Hoefler, Papa och Decenzo en delad andra plats.

Uppgift (d)

Vi tittar först på hur vi skattar α och β för en allmän mängd data. Vi gör detta med momentmetoden (av anledningen att vi inte vill behöva derivera täthetsfunktionen). Momentmetoden ger

$$m_1(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} x_i = E[X] = \frac{\alpha}{\alpha+\beta}.$$

Väntevärdet för en betafördelning är taget från formelbladet. Sen har vi det andra stickprovsmomentet

$$m_2(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} x_i = E[X^2] = \mathrm{Var}[X] + E[X]^2 \iff$$

$$\iff m_2(\mathbf{x}) - m_1(\mathbf{x})^2 = \mathrm{Var}[X] = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} =$$

$$= m_1(\mathbf{x})\frac{\beta}{(\alpha+\beta)(\alpha+\beta+1)} \iff \frac{m_2(\mathbf{x}) - m_1(\mathbf{x})^2}{m_1(\mathbf{x})} = \frac{\beta}{(\alpha+\beta)(\alpha+\beta+1)}.$$

Vi kommer ju alltid kunna bestämma de första stickprovsmomenten utifrån datan, därför vill vi bara kunna uttrycka α och β i termer av dessa stickprovsmoment. Vi tittar först på den första likheten och får

$$m_1(\mathbf{x}) = \frac{\alpha}{\alpha+\beta} \iff \beta = \frac{\alpha(1-m_1(\mathbf{x}))}{m_1(\mathbf{x})}.$$

Med detta kan vi substutuera β in i den andra likheten för att sedan försöka lösa ut α. Låt oss för tillfället skriva $m_1 = m_1(\mathbf{x})$ och $m_2 = m_2(\mathbf{x})$ bara för att all algerbra ska bli en aning smidigare. Då har vi

$$\frac{m_2 - m_1^2}{m_1} = \frac{\alpha(1-m_1)}{m_1\left(\alpha + \frac{\alpha(1-m_1)}{m_1}\right)\left(\alpha + 1 + \frac{\alpha(1-m_1)}{m_1}\right)}.$$

Vi ser att vi kan bryta ut en faktor α i nämnaren och därmed tar detta α ut det som står i täljaren. Vidare kan vi i båda faktorerna i nämnaren skriva summorna på gemensamt bråkstreck, vilket ger

$$= \frac{1-m_1}{(m_1+1-m_1)\left(\frac{\alpha m_1 + m_1 + \alpha - \alpha m_1}{m_1}\right)} = \frac{m_1(1-m_1)}{m_1+\alpha}.$$

Målet är som sagt att isolera α så nästa steg blir att multiplicera båda led med $m_1 + \alpha$. Vi får

$$\frac{(m_1+\alpha)(m_2-m_1^2)}{m_1} = m_1(1-m_1) \iff \alpha = \frac{m_1^2(1-m_1)}{m_2-m_1^2} - m_1.$$

Vi har alltså lyckats skatta α och med det kan vi skatta β genom

$$\beta = \frac{\alpha(1-m_1)}{m_1}.$$

I python koden tittar vi på den data som motsvarar lyckade trick, beräknar m_1 och m_2 och får då fram skattade värden för α och β. Vi får tabellen:

Namn	Skattning av α	Skattning av β
Decenzo	24.4559	6.1369
Foy	51.5895	10.6801
Gustavo	70.6446	21.8094
Hoban	107.6989	17.1145
Hoefler	32.5226	12.0411
Jordan	23.0630	4.1880
Midler	43.7232	12.8524
Mota	24.0067	8.6045
Oliveira	68.8786	22.4403
O'neill	1252.6663	275.9047
Papa	22.3148	8.0739
Ribeiro C	194.7805	72.0334
Shirai	20.5827	2.6067

Uppgift (e)

Rimligen använder vi de skattade värdena av α och β för att se vilken skatebordare som motsvarar den högsta kvoten $\alpha/(\alpha+\beta)$ eftersom denna kvot representerar väntevärdet.

Namn	**Betyg**
Shirai	0.8876
Hoban	0.8629
Jordan	0.8463
Foy	0.8285
O'Neill	0.8195
Decenzo	0.7994
Midler	0.7728
Gustavo	0.7641
Oliveira	0.7543
Mota	0.7361
Papa	0.7343
Ribeiro C	0.7300
Hoefler	0.7298

Vi ser att om tricket landas är Shirai, Hoban och Jordan i förväntan bäst betyg. Eftersom totalbetyget avser de bästa två tricken av fyra tänker jag att ett bra betyg givet att tricket sitter väger tyngre än en lite högre chans att ett trick faktiskt sitter. Därav tänker jag nog att Shirai, Hoban och Jordan är mest

troliga att få det högsta totalbetyget i förväntan. Foy kanske också bör nämnas eftersom han är överlägset bäst på att landa trick och då han knappt gör sämre trick än de i top 3.

Uppgift 2. En modell för bra run betyg

Uppgift (a)

Vi ser att g avbildar intervallet $(0, 1)$ på hela $\mathbb{R}$ först genom att inse att g är kontinuerlig på dess definitionsmängd. Sedan kan vi se att funktionens värden sträcker sig mot negativ oändlighet genom gränsvärdet $\lim_{x\to 0} \log(x/(1-x)) = \lim_{x\to 0} \log(x) - \log(1-x) = \lim_{x\to 0} \log(x) - 0 \to -\infty$, eftersom logaritmfunktionen närmar sig negativ oändlighet då $x \to 0$. Sen har vi gränsvärdet $\lim_{x\to 1} \log(x) - \log(1-x) = \lim_{x\to 1} 0 - \log(1-x) \to \infty$. Vi subtraherar med något som närmar sig negativ oändlighet, alltså får vi att g närmar sig oändlighet. Eftersom g är kontinuerlig och antar värden från negativ oändlighet till oändlighet så antar g alla värden i hela $\mathbb{R}$. Eftersom funktionen är strikt växande är den bijektiv och då inverterbar, vi hittar inversen genom

$$y = \log(x/(1-x)) \iff e^y = \frac{x}{1-x} \iff e^y(1-x) = x$$

$$\iff e^y = x(e^y + 1) \iff x = \frac{e^y}{e^y + 1}, g^{-1}(y) = \frac{e^y}{e^y + 1}$$

Vi kan också dubbelkolla och se att både täljare och nämnare är alltid positiva, samt nämnaren är alltid större än täljaren. Därmed avbildas y på intervallet $(0, 1)$.

Uppgift (b)

Jag beräknade kovariansmatrisen genom att först transformera datan med g och sen ta np.cov och fick följande:

$$\begin{pmatrix} 1.2008 & -0.20495 \\ -0.20495 & 2.2628 \end{pmatrix}$$

Vi kan läsa av kovariansen i (1, 2) eller (2, 1). Kovariansen är en storleksordning mindre än den vanliga variansen och därmed är det inte alls omöjligt att R_1 och R_2 är oberoende. Vi kan ju ändå inte vänta oss kovarians noll om de är oberoende eftersom vi rimligen borde få variationer i datan som får det att verka som att det finns en kovarians. Kort sagt kan vi gissa att de är oberoende eftersom kovariansen är låg. Sedan skulle vi också kunna argumentera rent logiskt att kovariansen mer troligt borde vara positiv om något, eftersom ett bra resultat i en run bör indikera någon slags bra dagsform som också hjälper i run 2 (med det är lite spekulativt kanske).

Uppgift (c)

Vi vill ta fram ett hypotestest för hypotesen H_0: R_1 och R_2 är oberoende. Vi ser i boken att vi kan ta fram korrelationskoeficienten ρ och använda faktumet att

$$T(\mathbf{X}) = \sqrt{n-2}\frac{\rho}{\sqrt{1-\rho^2}}$$

är en t-fördelad variabel. Vi kan ta fram korrelationskoeficienten med de skattade värdena genom definitionen av korrelation. Vidare har vi $n = 50$, då vi har 50 runbetyg av varje sort totalt. Vi förkastar H_0 om $|T(\mathbf{x})| \geq c$. Om vi låter signifikansnivån $\alpha = 0.05$ får vi värdet på c enligt $1 - \alpha/2$ kvantilen för en t_{48} fördelning, vilket vi beräknar med scipy till ungefär 2.01. Samtidigt får vi $|T(\mathbf{x})| = 0.87$. Detta betyder att vi inte kan förkasta vår nollhypotes. Resultatet var inte särskilt signifikant, vi får samma resultat på testet även med ett överdrivet högt $\alpha = 0.25$. Kort sagt, testet beslutar att behålla nollhypotesen.

Uppgift (d)

Vi varje åkare har vi $R_1 = N_1$, $R_2 = \lambda R_1 + N_2 = \lambda N_1 + N_2$. Vi har alltså en transformation av variabler från $\mathbf{N}$, till $\mathbf{R}$. Transformationen ges av

$$\begin{bmatrix} R_1 \\ R_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \lambda & 1 \end{bmatrix} \begin{bmatrix} N_1 \\ N_2 \end{bmatrix}.$$

Vi har då att väntevärdet ges av $E[\mathbf{R}] = [\theta_1, \lambda\theta_1 + \theta_2]^T$. Sedan har vi kovariansmatrisen enligt

$$\begin{bmatrix} 1 & 0 \\ \lambda & 1 \end{bmatrix} \begin{bmatrix} \omega_1^2 & 0 \\ 0 & \omega_2^2 \end{bmatrix} \begin{bmatrix} 1 & \lambda \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \omega_1^2 & \lambda\omega_1^2 \\ \lambda\omega_1^2 & \lambda^2\omega_1^2 + \omega_2^2 \end{bmatrix}.$$

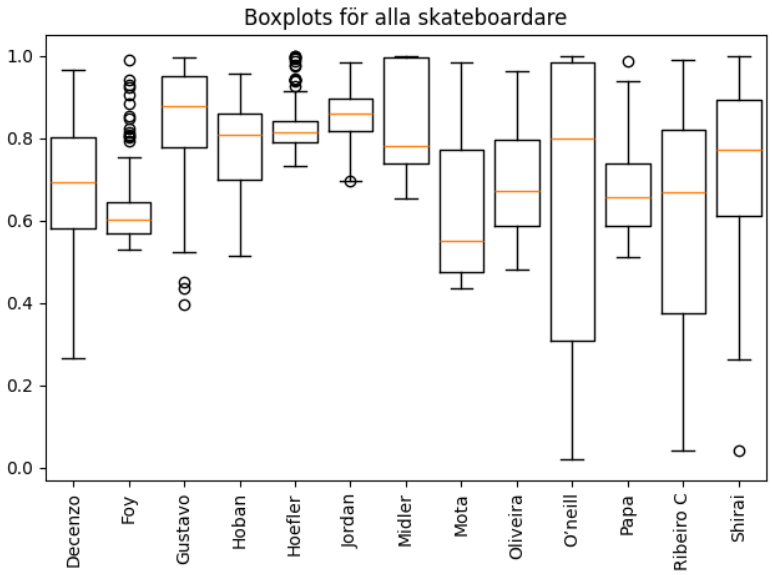
Med detta kan vi skatta alla parametrar med hjälp av stickprovsväntevärden, stickprovsvarianser och stickprovskovarianser. Först och främst kan vi låta θ_1 vara stickprovsväntevärdet för R_1. Sen låter vi ω_1^2 vara stickprovsvariansen för R_1. Vidare låter vi $\lambda\omega_1^2$ vara stickprovskovariansen, så vi låter λ vara kvoten mellan stickprovskovariansen och det redan skattade ω_1^2. Med detta kan vi använda ω_2^2 = stickprovsvariansen för $R_2 - \lambda^2\omega_1^2$. Sist skattar vi θ_2 genom att ta stickprovsmedelvärdet av R_2 och subtrahera $\lambda\theta_1$. Vi utför dessa beräkningar i python och får:

Namn	θ_1	θ_2	λ	ω_1^2	ω_2^2
Decenzo	0.3563	0.4489	0.2962	1.1041	0.5395
Foy	-0.7459	0.1445	-0.1322	1.6124	0.0324
Gustavo	0.1256	0.8015	-0.8233	1.9763	1.2518
Hoban	0.1290	1.2649	-0.4276	0.4702	0.8330
Hoefler	1.3679	3.6921	-2.9730	0.1993	2.5865
Jordan	0.7325	1.7274	-0.0288	1.0821	0.3951
Midler	1.1228	1.8761	-2.5007	0.2626	8.4439
Mota	0.0991	-0.2687	-0.1646	1.3764	0.0437
Oliveira	0.5261	0.3143	-0.2797	0.9630	0.3881
O'neill	0.2034	-0.8897	0.0064	4.2504	3.1473
Papa	0.2971	0.1343	-1.3013	0.8879	0.1297
Ribeiro C	-0.0236	0.3069	0.6951	2.4129	0.0379
Shirai	0.9339	-1.0980	1.3255	0.8570	2.7945

Negativa värden på λ indikerar att resultatet i run1 har negativ påverkan på resultaten av run2. Lite oväntat kanske, men det fanns inte så mycket data. Man kanske spelar mer säkert om man presterar dåligt i run 1 medan man måste riskera något om man ska kunna få en bättre run 2 om run 1 redan var bra (återigen ganska spekulativt).

Uppgift (e)

Här är alla låddiagram genererade av python:



Vi tittar på låddiagrammen och försöker avgöra vilka som har störst chans att få mest poäng. En hög median samt andra kvartiler borde rimligen innebära goda chanser att vinna. Så Gustavo har nog en bra chans. Det har även O'neill och Jordan.

Uppgift (f)

Vi får följande tabell över stickprovsmedelvärden av de genererade punkterna:

Vi ser högst stickprovsmedelvärde för Jordan, Midler, Gustavo och Hoefler. Rimligen är dessa fyra mest sannolika att vinna tävlingen i förväntan.

Baserat på tidigare trickbetyg och runbetyg kan vi säga att de åkare som inte har befunnit sig i top 3 till någon av kategorierna är osannolika vinnare. Dessa är: Foy, Decenzo, Oliveira, Mota, Papa och Ribeiro C. Jag har alltså bara tagit bort de med bra trick i förväntan, bra run i förväntan och bra låddiagram.

Name	Stickprovsmedelvärde
Decenzo	0.6656
Foy	0.6164
Gustavo	0.8396
Hoban	0.7573
Hoefler	0.8293
Jordan	0.8468
Midler	0.8389
Mota	0.6097
Oliveira	0.6942
O'neill	0.6817
Papa	0.6787
Ribeiro C	0.6247
Shirai	0.7759

Uppgift 3
En (helt frekventisk) modell för trickbetyg

Uppgift (a)

Rimligen plockar vi bort alla trick vars poäng var 0, transformerar resten med g och låter θ vara stickprovsmedelvärdet, samt ω^2 vara stickprovsvariansen. Värdena från python är här:

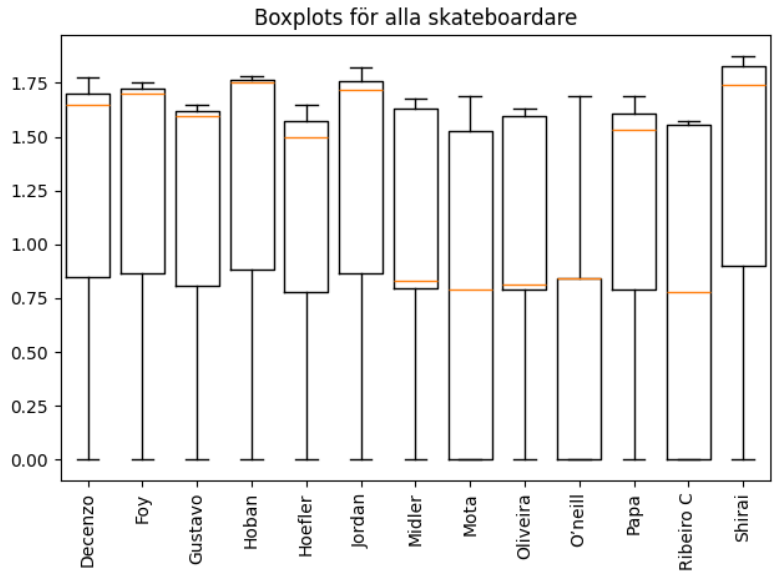
Namn	θ	ω^2
Decenzo	1.6493	0.2657
Foy	1.7806	0.1322
Gustavo	1.4179	0.0801
Hoban	1.9973	0.0756
Hoefler	1.2812	0.1477
Jordan	1.9245	0.2019
Midler	1.4697	0.1170
Mota	1.3222	0.2036
Oliveira	1.3667	0.0600
O'neill	1.6849	0.0049
Papa	1.3157	0.2128
Ribeiro C	1.2524	0.0219
Shirai	2.3114	0.3407

För att sen simmulera värden av T_i kan vi använda de skattade väredena av

ρ från uppgift 1.

Uppgift (b)

Vi implementerar simulering av T i python genom att simmulera en bernoulli fördelning och sen ersätta ettor med en $N(\theta, \omega^2)$ fördelning och utföra g^{-1} på dessa. Sedan beräknar vi $T = Z_4 + Z_3$.



Vi tittar på kvartilerna och försöker hitta vilka som ofta får högst poäng. Jag tycker att Foy, Hoban, Jordan och Shirai sticker ut med sina höga medianer. Därför verkar dessa vara troligast att få bästa tricksumma för deras två bästa trick.

Uppgift (c)

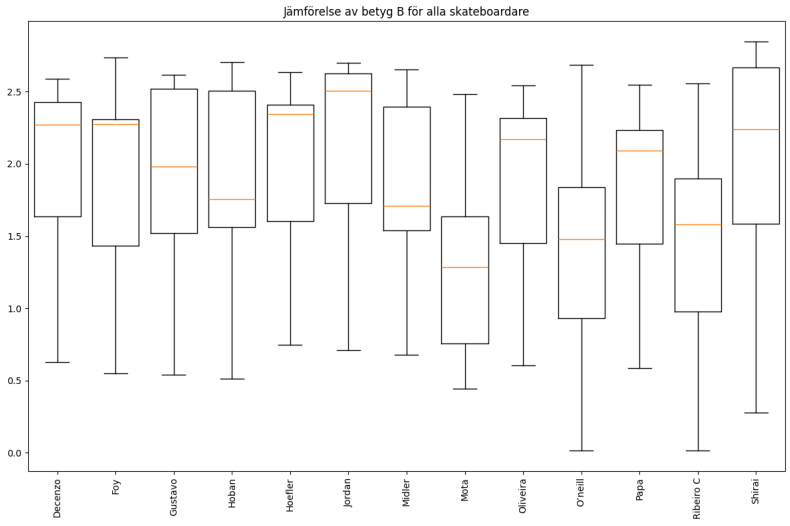
Vi beräknar lätt stickprovsmedelvärdena med np.mean, här är en sorterad tabell:

Name	Stickprovsmedelsvärde
Foy	1.3645
Hoban	1.3051
Shirai	1.3048
Decenzo	1.2893
Jordan	1.2536
Papa	1.2036
Gustavo	1.2351
Hoefler	1.1561
Oliveira	1.0777
Midler	1.0268
Ribeiro C	0.7784
Mota	0.7633
O'neill	0.7424

Utifrån denna kan vi säga att de mest troliga vinnarna är Foy, Hoban och Shirai.

Uppgift (d)

Vi kopierar bara in koden från uppgift 2e och adderar trickvektorn med runvektorn. Vi får då följande låddiagram:



Vi tittar på diagramen. De bästa är nog Jordan, Hoefler och Shirai.

Uppgift (e)

Stickprovsmedelvärdena i en sorterad tabell:

Name	Stickprovsmedelvärde
Jordan	2.1124
Decenzo	2.0612
Shirai	2.0442
Hoefler	1.9988
Foy	1.9451
Gustavo	1.9400
Hoban	1.8747
Oliveira	1.8716
Papa	1.8311
Midler	1.8518
O'neill	1.4546
Ribeiro C	1.4513
Mota	1.2913

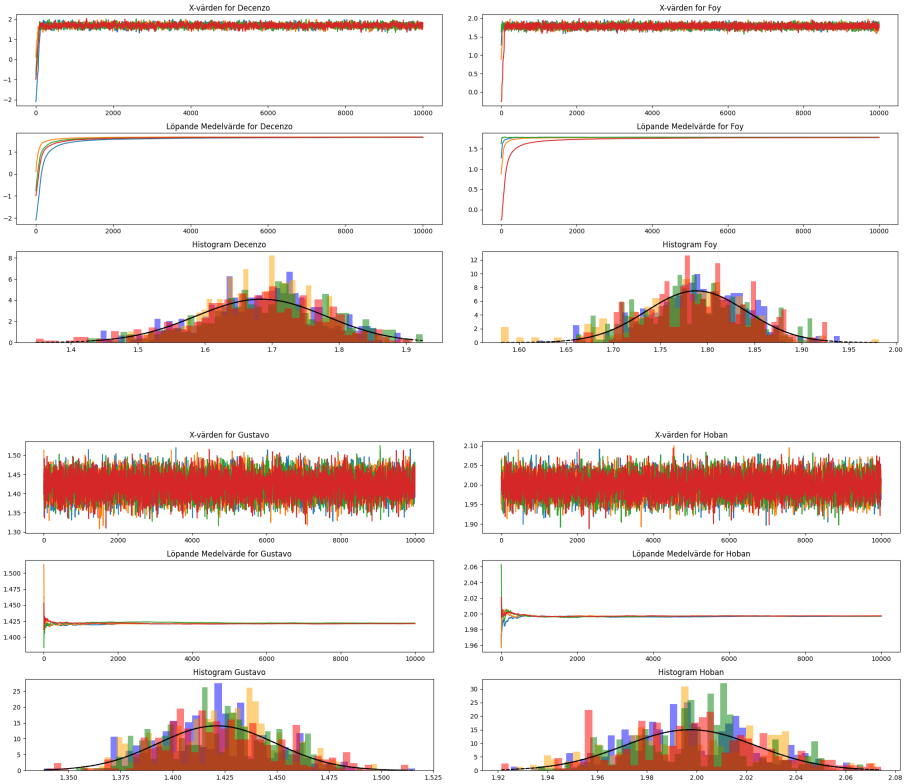
Bästa värden för Shirai, Jordan, Decenzo och Hoefler. Dessa fyra är mest sannolika att vinna i förväntan eftersom de i förväntan bör få högst poäng B. Dock varierar ordningen mellan dessa fyra för olika körningar av programmet.

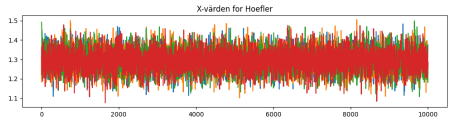
Uppgift 4. En smak av Bayesianska metoder.

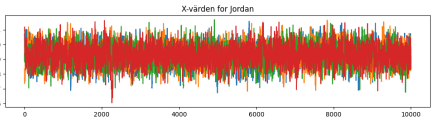
Uppgift (a)

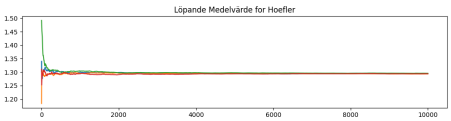
Enligt Bayes skatt är aposteriorifördelningen för Θ proportionell mot aprioriförde lningen gånger den simultana datafördelningen (simultan för alla olika datapunkter, dvs trick). Vi vill generera ett approximativt stickprov från denna fördelning och det gör vi med metropolisalgoritmen. Jag genererade först ett initialvärde med en normalfördelning med parametrar från uppgift 3a. Sedan implementeras förslagsfunktionen som värdet från den förra iterationen plus en uniformt fördelad stockastisk variabel som antar ett värde mellan $-0,1$ och $0,1$. Vi får då en kedja med värden som konvergerar mot att vara fördelade som aposteriorifördelningen vi söker. Vi kan då ta ut de sista 500 värdena

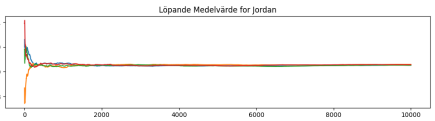
ur denna kedja och låta dessa utgöra stickprovet för Θ. Sedan kan vi plotta dessa 500 värden i ett histogram. Här kommer plottar för varje skateboardare med värdena längst fyra kedjor, dess löpande medelvärde samt ett histogram för den genererade datan, tillsammans med en plot av den faktiska aposteriorifördelningsfunktionen. Denna funktion har normaliserat genom att hitta normaliseringsfaktorn med hjälp av trapetsregeln. Som vi kan se på följande plottar så konvergerar alla kedjor på ett önskvärt sätt och histogrammen förhåller sig till fördelningsfunktionerna på ett trovärdigt sätt.

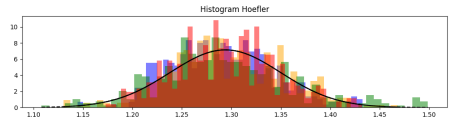


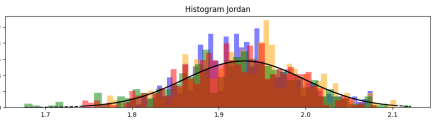

X-värden for Hoefler

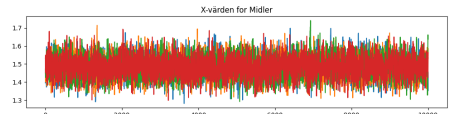

X-värden for Jordan


Löpande Medelvärde for Hoefler

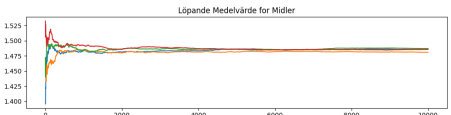

Löpande Medelvärde for Jordan

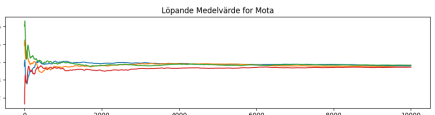

Histogram Hoefler

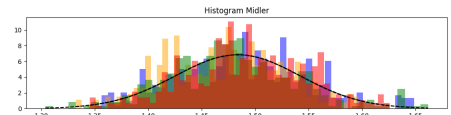

Histogram Jordan

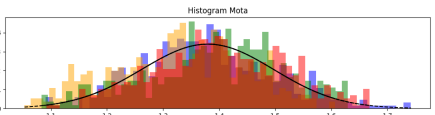

X-värden for Midler


X-värden for Mota

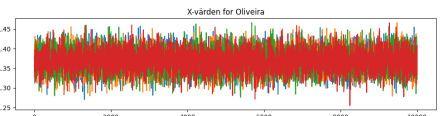

Löpande Medelvärde for Midler

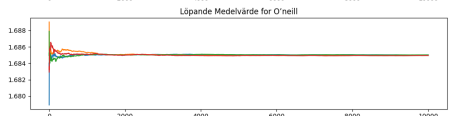

Löpande Medelvärde for Mota

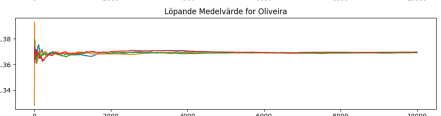

Histogram Midler

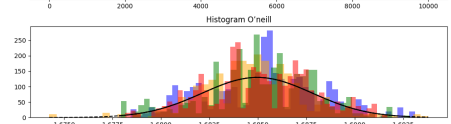

Histogram Mota

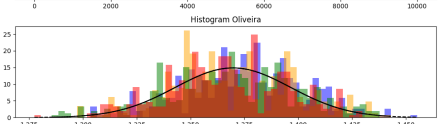

X-värden for O'neill

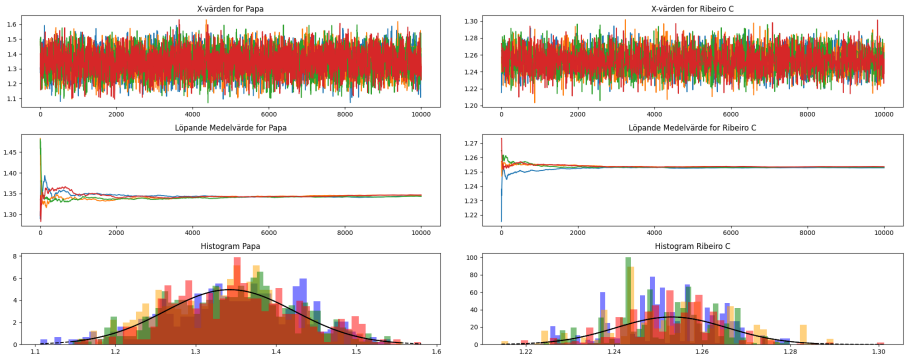

X-värden for Oliveira

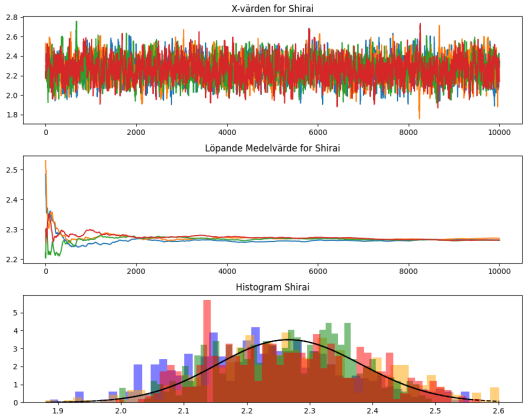

Löpande Medelvärde for O'neill


Löpande Medelvärde for Oliveira


Histogram O'neill


Histogram Oliveira



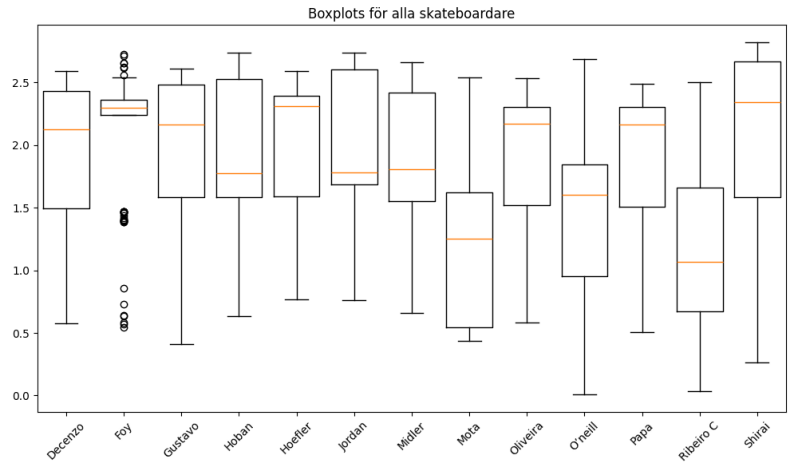


Det visade sig att O´neill hade väldigt svårt att uppdatera sina värden, så jag dividerade skalan med 10 för att få fram en rimlig simulering för honom. Jag kan också kommentera att vissa av histogrammen inte verkar passa riktigt lika bra som man hade kunnat hoppas. Dock kan jag se att om jag genererar ett stickprov med 5000 värden istället, så passar histogrammet väldigt bra med funktionen. Vi kan därför påstå att imperfektionen bara beror på låg mängd data snarare än ett systematiskt fel.

Uppgift (b)

Givet värden av Θ från aposteriorifördelningen kan vi generera nya trickbetyg med hjälp av den givna datafördelningen där vi låter μ motsvara de simulerade värdena av Θ. Med 400 genererade trickbetyg kan vi gå över till precis samma procedur som i uppgift 3d där vi simulerade en bernoullifördelning för att bestämma hur många värden vi satte till noll medan vi utförde g^{-1} på

resterande värden. Sedan grupperade vi betygen i grupper om fyra, valde ut de två största och adderade dessa med det frekvensistiskt skattade runbetyget. Detta gav 100 B-betyg per skateboardare och låddiagrammen nedan.



Lite svårt att konkret säga hur låddiagramen skiljer sig, men vi kan säga att det helt klart finns stora likheter. De bästa tre verkar vara Shirai, Hoefler och Foy. När det är så jämt kommer vi dock få stora skillnader mellan olika simuleringsomgångar.

Uppgift (c)

De sorterade stickprovsmedelvärdena ges av:

Skateboardare	**Stickprovsmedelvärde**
Foy	2.0809
Shirai	2.0356
Hoefler	1.9549
Hoban	1.9488
Gustavo	1.9486
Jordan	1.9410
Oliveira	1.9331
Decenzo	1.9091
Papa	1.8886
Midler	1.8465
O'neill	1.4643
Mota	1.2163
Ribeiro C	1.1519

Vi ser högst stickprovsmedelvärde för Foy, Shirai och Hoefler. Vi ser viss skillnad från resultaten från tidigare. Vi kan notera att det är väldigt jämnt (och Foy får inte så högt värde i andra körningar), så resultaten ger knappast några garantier. Om jag kör pythonprogrammet flera gånger får jag inte samma svar. Vi kan väl säga att utifrån den sista tabellen så är de med stickprovsmedelvärde över 1,94 något mer sannolika att vinna än resten av gruppen. Kanske att Shirai är den allra största favoriten eftersom han toppar listan på flest körningar.