OHJELMOINNIN JA TIETOTEKNIIKAN YHDISTÄMINEN TODENNÄKÖISYYS-LASKENTAAN YLÄKOULUSSA

Uhkapelurit, Monty Hall ja COVID-19 vasta-ainetesti

Alina Kulmala

Pro gradu -tutkielma
Syyskuu 2020

MATEMATIIKAN JA TILASTOTIETEEN LAITOS

Turun yliopiston laatujärjestelmän mukaisesti tämän julkaisun alkuperäisyys
on tarkastettu Turnitin OriginalityCheck-järjestelmällä.

TURUN YLIOPISTO
Matematiikan ja tilastotieteen laitos

KULMALA, ALINA: Ohjelmoinnin ja tietotekniikan yhdistäminen todennäköisyyslaskentaan yläkoulussa. Uhkapelurit, Monty Hall ja COVID-19 vasta-ainetesti
Pro gradu -tutkielma, 35 s.
Matematiikka
Syyskuu 2020

Tässä matematiikan opettajalinja pro gradu -tutkielmassa esitellään kolme todennäköisyyslaskennan ongelmaa, joiden ratkaisemisessa voidaan hyödyntää ohjelmointia tai taulukkolaskentaa. Näin saadaan yhdistettyä todennäköisyyslaskennan ja algoritmisen ajattelun oppiminen. Algoritmisen ajattelun kehittyminen ja ohjelmoinnin käyttäminen matematiikan ongelmien ratkaisemiseen on kirjattu peruskoulun opetussuunnitelmaan, joten tämän kaltaisille tehtäväkokonaisuuksille on perusteltu tarve.

Ensimmäinen käsiteltävä ongelma on uhkapelurit (engl. Gambler's ruin), jossa oppilas tekee peliä simuloivan koodin, ja tutustuu sitä kautta ohjelmoinnin perusrakenteisiin. Toisena tehtävänä oppilas tekee ohjelman, jolla voidaan simuloida Monty Hall -peliä useita kierroksia. Näin oppilas pääsee tutkimaan, miten voittotodennäköisyys lähestyy teoreettista arvoa, kun simuloitavat kierrosmäärät kasvavat. Viimeisenä esimerkkinä on verikokeesta tehtävän vasta-ainetestin luotettavuus. Tämä on käsiteltävistä esimerkeistä matemaattisesti haastavin, mutta myös käytännönläheisin.

Tutkielmassa esitetään kaikkien ongelmien matemaattiset taustat, vaikka ne eivät ole peruskoululaiselle välttämättömiä ymmärtää. Niihin tutustuminen on sopivaa ylöspäin eriyttävää materiaalia. Muutenkin tehtäväkokonaisuudet on koottu niin, että eriyttäminen on luontevaa ja sujuvaa.

Tutkielmassa on esimerkkiratkaisut tehtäviin, ja liitteenä oppilasohjeet töihin. Esimerkkiratkaisuissa on käytetty pääosin Scratch-ohjelmointiympäristöä, ja lisäksi taulukkolaskentaohjelmaa ja Geogebraa.

Asiasanat: todennäköisyyslaskenta, ohjelmointi, Scratch, simulointi, algoritminen ajattelu, matematiikka, yläkoulu

Sisältö

1 Johdanto

Jos selaa peruskoulun oppikirjoja, todennäköisyyslaskennan sisällöstä saattaa muodostua kuva, että ydinsisältöä on osata laskea kolikon- ja nopanheittoon, sekä korttipeleihin liittyviä todennäköisyyksiä. Todennäköisyyslaskennan sovelluskenttä on kuitenkin laaja, esimerkiksi tilastotiede, tekoäly ja ydinfysiikka pohjautuvat todennäköisyyksien laskemiseen.

Todennäköisyyslaskenta kuuluu tyypillisesti peruskoulun yhdeksännen luokan sisältöön. Voimassa olevassa opetussuunnitelmassa ei määritellä sisältöä täsmällisemmin, kuin ”lasketaan todennäköisyyksiä”. Opetussuunnitelmaan on kirjattu yleisesti matematiikan opetuksen tavoitteeksi algoritmisen ajattelun kehittyminen ja ohjelmoinnin käyttäminen ongelmien ratkaisemiseen. Lisäksi useassa kohdassa korostetaan yhteyden luomista matematiikan opetuksen ja ympäröivän yhteiskunnan välille ja matematiikan soveltamista. [4.] Näistä lähtökohdista tässä työssä on kehitetty kolme esimerkkitapausta, joiden avulla voidaan opetella todennäköisyyksien laskemista, ja lisäksi ohjelmointia, tietotekniikan taitoja ja soveltamista.

Ensimmäisenä käsiteltävänä esimerkkinä on peli nimeltä uhkapelurit, joka on matemaattiselta kannalta helpoin esimerkki, ja sen kautta opitaan ensisijaisesti ohjelmointia. Pelin asetelma on lähtöisin todennäköisyyslaskennan uranuurtajien Blaise Pascalin ja Pierre Fermat kirjeenvaihdosta vuonna 1656 [7]. Monty Hall -ongelmassa oppilas tutkii pelin voittotodennäköisyyttä ensin simuloimalla, ja miettii sen jälkeen, osaisiko ratkaista saman klassisen todennäköisyyslaskennan keinoin. Verikokeen positiivisen tuloksen luotettavuuden analysoimisessa oppilas pääsee hyödyntämään taulukkolaskentaohjelmaa ja miettimään, mitä saatu tulos tarkoittaa esimerkiksi virusepidemioihin liittyvien joukkotestaamisten kannalta.

Suomessa on tehty linjaus, että algoritmisen ajattelun oppimiselle ja ohjelmoinnille ei osoiteta peruskoulussa erillisiä tunteja, vaan tavoite kuuluu matematiikkaan. On oleellista, että sen lisäksi, että matematiikka tukee ohjelmoinnin oppimista, myös ohjelmointiin käytetty aika tukisi matematiikan oppimista. Todennäköisyyslaskentaan liittyvät simuloinnit ovat yksi hyvä osa-alue, jossa nämä kaksi tieteenalaa yhdistyvät luontevasti.

2 Matemaattinen tausta

2.1 Todennäköisyyslaskennan käsitteitä

Todennäköisyyslaskennan tavoitteena on kehittää satunnaisluonteisten ilmiöiden kuvaamiseen soveltuvia matemaattisia malleja. Todennäköisyyden laskeminen voidaan määritellä numeeriseksi tavaksi mitata varmuutta tai epävarmuutta jollekin tapahtumalle. Todennäköisyys perustuu periaatteessa empiiriseen kokemukseen, ja samoissa olosuhteissa toistettuna todennäköisyyden arvoksi pitäisi saada sama toistettaessa koetta. [5.]

Todennäköisyyslaskennan teorian rakentaminen lähtee liikkeelle *satunnaisilmiön* käsitteestä. Ollakseen satunnaisilmiö, tarkasteltavan ilmiön pitää täyttää kolme ehtoa:

(i) Alkutilasta voidaan päätyä useisiin lopputiloihin. Kolikonheitossa esimerkiksi lopputila on kruuna tai klaava.

(ii) Alkutilassa ei voida ennustaa yksittäiselle kerralle, mikä lopputila toteutuu.

(iii) Vaikka yksittäisen lopputilan ennustaminen on mahdotonta, lopputilojen osuudet esiintyvät säännönmukaisesti, eli lopputilojen suhteelliset frekvenssit voidaan määrittää. Esimerkiksi arpakuution heitossa todennäköisyys jokaiselle silmäluvulle on $\frac{1}{6}$. [3.]

Satunnaisilmiön tulosvaihtoehtoja kutsutaan *alkeistapahtumiksi*, nopanheitossa siis jokainen mahdollinen silmäluku on yksi alkeistapahtuma. Alkeistapahtumat eivät välttämättä ole lukuja, joten tarvitaan funktio, joka liittää reaaliluvun tai reaalilukuvektorin jokaiseen koetulokseen. Tätä funktiota kutsutaan *satunnaismuuttujaksi*, ja sitä merkitään tässä työssä isolla kirjaimella *X, Y,* ... Satunnaismuuttujan arvoja merkitään vastaavilla pienillä kirjaimilla *x, y,* … Tutkittava satunnaiskoe voi olla esimerkiksi neljässä perättäisessä kolikonheitossa saatavien klaavojen lukumäärä. Tällöin satunnaismuuttuja *X* on klaavojen lukumäärä, ja sen mahdollisia arvoja ovat 0, 1, 2, 3 ja 4. Nyt merkintä $X \geq 2$ tarkoittaa tilannetta, jossa heittosarja sisältää vähintään kaksi klaavaa.

Esimerkki satunnaismuuttujasta, joka ei luonnostaan ole reaaliluku, on vaikka suuresta joukosta poimittujen ihmisten silmien väri. Voidaan merkitä $Y = silmien\ väri$, ja koodata tulokset siten, että $y = 1$ tarkoittaa sinistä silmien väriä ja $y = 2$ ruskeaa.

Alkeistapahtumien *s* muodostama joukko muodostaa *otosavaruuden Ω.* Alkeistapahtumat ovat otosavaruuden *alkioita*. Nopanheitossa silmälukujen muodostama otosavaruus on siis $\Omega = \{1, 2, 3, 4, 5, 6\}$. Otosavaruuden osajoukko $A \subset \Omega$ on *tapahtuma*, esimerkki tapahtumasta nopanheitossa on parillinen silmäluku.

Todennäköisyyslaskennassa tavoitteena on määrittää satunnaisilmiön eri tapahtumien todennäköisyydet. *Todennäköisyysmitta P* (englannin sanasta probability) on funktio, joka liittää jokaiseen tapahtumaan luvun väliltä [0,1]. $P(X \in A)$, tai lyhyemmin $P(A)$ on tapahtuman *A* todennäköisyys. [5.]

Jos tapahtuma A sattuu väistämättä, eli jokainen otosavaruuden alkeistapahtuma johtaa tapahtumaan *A*, sanotaan, että *A* on varma tapahtuma, ja sen todennäköisyys $P = 1$. Tapahtuma, joka ei voi sattua, on mahdoton tapahtuma, ja sen todennäköisyys $P = 0$.

2.2 Tilastollinen ja klassinen todennäköisyys

Todennäköisyyslaskennan aksioomien ymmärtäminen ei ole tarpeellista koulumatematiikassa, joten ne jätetään tässä esittelemättä. Koulumatematiikassa todennäköisyys määritellään *tilastollisen* ja *klassisen todennäköisyyden* avulla. Tässä kappaleessa esitellään määritelmät niille, ja tuodaan esiin pedagogisesta näkökulmasta määritelmien puutteet.

Tilastollisessa todennäköisyydessä keskeinen käsite on *suhteellinen frekvenssi*. Ajatuksena on, että sama koe toistetaan *n* kertaa. Tapahtuman *A* suhteellinen frekvenssi lasketaan $p_n(A) = \frac{f_n(A)}{n}$, missä $f_n(A)$ on tapahtuman *A* realisoitumisen frekvenssi, eli realisoitumisten lukumäärä. Tapahtuman *A* tilastollinen todennäköisyys on raja-arvo $P(A) = \lim_{n \to \infty} p_n(A)$.

Matemaattisesta näkökulmasta määritelmä ei ole täsmällinen, sillä koetta ei voida toistaa äärettömästi. Mikään ei myöskään takaa, että suhteellisten frekvenssien jono olisi suppeneva. Käytännön sovelluksissa suhteelliset frekvenssit vaikuttavat usein suppenevan, kun toistojen määrä on suuri, jolloin tapahtuman *A* suhteelliset frekvenssit $p_{n1}(A), p_{n2}(A), \ldots, p_k(A)$ poikkeavat toisistaan vain hyvin vähän. [5.]

Klassisessa todennäköisyydessä alkeistapauksille tehdään lisäksi kaksi oletusta: alkeistapauksia on äärellinen määrä *N*, ja ne ovat kaikki yhtä todennäköisiä. Alkeistapaukset ovat silloin *symmetrisiä*. Tapahtuman $A \subset \Omega$ klassinen todennäköisyys lasketaan tapahtumalle *A* suotuisten alkeistapausten ja kaikkien alkeistapausten lukumäärän suhteena, $P(A) = \frac{card\ (A)}{N}$. [5.]

Klassisen todennäköisyyden määritelmän suurin ongelma on se, että alkeistapausten oletetaan olevan yhtä todennäköisiä, vaikka ollaan vasta määrittelemässä todennäköisyyden käsitettä [5].

2.3 Todennäköisyyden ominaisuuksia

Otetaan käyttöön joukko-opin merkinnät, ja tutkitaan otosavaruuden Ω osajoukkoja A ja *B*. Tapahtuman A *komplementtitapahtuma* on $\bar{A}$, joka tarkoittaa kaikkia niitä otosavaruuden alkeistapahtumia, joissa *A* ei realisoidu, eli $\bar{A} = \Omega \setminus A = \{x \in \Omega : x \notin A\}$.

Määritellään joukkojen A ja B *unioni* eli *yhdiste* niiden alkioiden joukoksi, jotka kuuluvat joukkoon A tai B. Joukkojen yhdiste $A \cup B$ koostuu alkeistapauksista, jotka kuuluvat joko joukkoon A tai B tai molempiin, eli $A \cup B = \{x \in \Omega \mid x \in A \lor x \in B\}$.

Lisäksi otetaan käyttöön käsite joukkojen A ja B *leikkaus* $A \cap B$, jossa vaaditaan, että alkio kuuluu joukkoon A ja joukkoon B, eli $A \cap B = \{x \in S \mid x \in A \land x \in B\}$. Tapahtumat ovat *toisensa poissulkevat*, jos A ja B eivät voi sattua samanaikaisesti. Tällöin $A \cap B = \emptyset$.

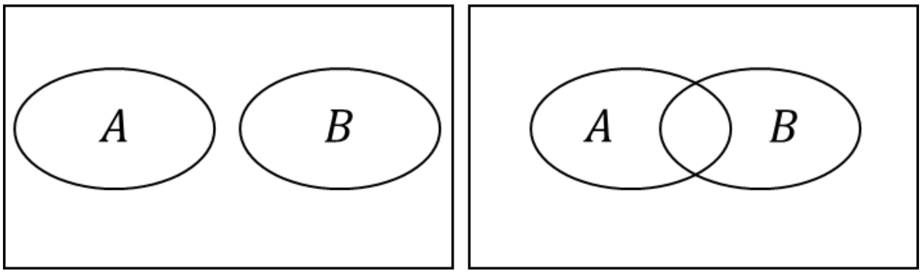
Tutkitaan joukkojen unionin todennäköisyyttä siinä tapauksessa, että tapahtumat ovat toisensa poissulkevia. $P(A \cup B)$ tarkoittaa sitä, millä todennäköisyydellä A tapahtuu tai B tapahtuu, sillä A ja B eivät voi tapahtua samanaikaisesti. Kun $A \cap B = \emptyset$, todennäköisyys sille, että A tapahtuu tai B tapahtuu, on

$$P(A \cup B) = P(A) + P(B) \ (tulos\ 2.3.1).$$

Tulos pätee myös useammille toisensa poissulkeville tapahtumille. Todennäköisyys sille, että ainakin joku tapahtumista toteutuu, on yksittäisten tapahtumien todennäköisyyksien summa, koska tapahtumat eivät voi sattua yhtä aikaa. Tilannetta havainnollistetaan kuvan 1 vasemmanpuoleisessa tilanteessa.

Kun $A \cap B \neq \emptyset$, tapahtumat A ja B voivat sattua myös samanaikaisesti, mikä esitetään kuvan 1 oikeanpuoleisessa tilanteessa. Tällöin on huomattava, ettei lasketa osajoukkojen A ja B leikkauksen todennäköisyyttä kahteen kertaan, eli

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) \ (tulos\ 2.3.2).$$

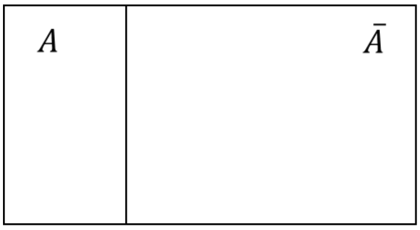


Kuva 1. Joukkojen unionin todennäköisyyteen vaikuttaa se, ovatko joukot toisensa poissulkevia. Todennäköisyys, että A tai B tapahtuu, on yksittäisten tapahtumien todennäköisyyksien summa, kun tapahtumat ovat toisensa poissulkevia. Oikeanpuoleisessa tilanteessa A ja B voivat tapahtua myös samanaikaisesti, eli ne eivät ole toisensa poissulkevia. Tällöin joukkojen unionin todennäköisyys on niiden leikkauksen verran pienempi.

Kun aksioomina on, että $A \cup \bar{A} = \Omega$, $A \cap \bar{A} = \emptyset$ ja $P(\Omega) = 1$, tuloksen 2.3.1 perusteella saadaan määritettyä todennäköisyys osajoukkojen summalle $P(A \cup \bar{A}) = P(A) + P(\bar{A}) = P(\Omega) = 1$. Tästä saadaan komplementti-tapahtuman todennäköisyydeksi

$$P(\bar{A}) = 1 - P(A)\ (tulos\ 2.3.3).$$

Komplementtitapahtuman todennäköisyyden laskeminen on monissa tilanteissa tarpeellista, ja tilannetta selvennetään vielä kuvassa 2. Tapahtuma A voisi olla esimerkiksi se, että nostettaessa pelikortti pakasta, saadaan ässä. Komplementtitapahtuma $\bar{A}$ on se, että nostettu kortti ei ole ässä.



Kuva 2. Tapahtuma A ja komplementtitapahtuma $\bar{A}$ muodostavat koko otos-avaruuden Ω. Tapahtumat A ja $\bar{A}$ ovat toisensa poissulkevia, eli eivät voi tapahtua samanaikaisesti. Tapahtumien A ja $\bar{A}$ yhteenlaskettu todennäköisyys on yksi.

Tapahtumat ovat *riippumattomia*, jos $P(A \cap B) = P(A)P(B)$. Riippumatto-muutta kuvaamaan käytetään merkintää $A \perp B$ ja jos tapahtumien tiedetään olevan toisistaan riippumattomia, määritelmää voi käyttää tapahtumien A ja B leikkauksen todennäköisyyden laskemiseen.

2.4 Ehdollinen todennäköisyys ja Bayesin kaava

Joskus on tilanteita, joissa tiedetään, että otosavaruuden tapahtuma B on toteutunut, ja halutaan määrittää, mikä tämän tieto huomioiden on tapahtuman A todennäköisyys. Tällöin puhutaan ehdollisesta todennäköisyydestä. Esimerkkinä voisi olla, että pelikortin tiedetään olevan kuva-kortti, ja halutaan laskea, millä todennäköisyydellä se on kuningatar. Tapahtuman *A ehdollinen todennäköisyys* siinä tapauksessa, että B on tapahtunut, merkitään $P(A|B)$. Kun $P(B) > 0$, ehdollinen todennäköisyys saadaan laskettua klassisen todennäköisyyden mallin mukaisesti, eli suotuisten alkeistapausten suhteena kaikkiin alkeistapauksiin. Suotuisia alkeistapauksia ovat ne, joissa sekä A että B tapahtuvat eli $A \cup B$, ja kaikkien alkeistapausten joukon muodostavat ne, joissa B tapahtuu. Näin ollen

$$P(A|B) = \frac{P(A \cap B)}{P(B)}\ (tulos\ 2.4.1).$$

Tästä saadaan johdettua toinen tapa laskea leikkauksen todennäköisyys, jossa ei tarvitse tuntea tapahtuman A todennäköisyyttä kuten silloin, jos lasketaan leikkauksen todennäköisyys riippumattomuuden määritelmän avulla. Nyt saadaan

$$P(A \cap B) = P(A|B)P(B)\ (tulos\ 2.4.2).$$

Riippuu tilanteesta, onko ehdollinen todennäköisyys $P(A|B)$ vai $P(B|A)$ helpompi selvittää. Niiden välille voidaan johtaa yhteys, jota kutsutaan Bayesin kaavaksi. Kun tuloksessa 2.4.1 vaihdetaan tapahtumien nimet toisin päin, saadaan $P(B|A) = \frac{P(B \cap A)}{P(A)}$. Leikkaus on vaihdannainen, joten edellisestä voidaan ratkaista $P(A \cap B) = P(B|A)P(A)$. Sijoitetaan tämä tuloksen 2.4.1 osoittajaan ja saadaan

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}\ (tulos\ 2.4.3),$$

mitä kutsutaan Bayesin kaavaksi.

3 Työssä käsiteltävien ongelmien matemaattiset ratkaisut

3.1 Uhkapelurit

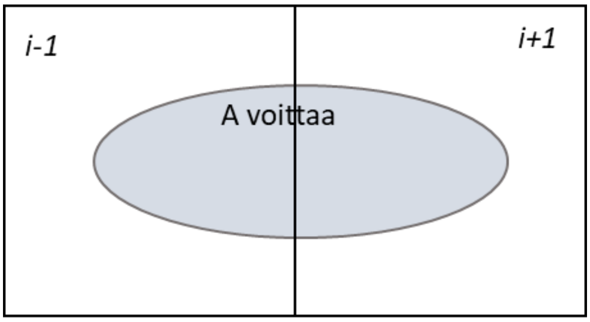
Tässä työssä käytetään nimitystä uhkapelurit ongelmalle, joka tunnetaan englanniksi nimellä Gambler's ruin. Ongelmasta on useita erilaisia versioita, otetaan tässä tarkasteluun peli, jossa alussa rahapotti suuruudeltaan N € on jaettuna kahdelle pelaajalle niin, että ensimmäisellä pelaajalla on i € ja toisella pelaajalla on loput. Määritellään pelaajalle A yksittäisellä kierroksella voittotodennäköisyys p, eli todennäköisyys hänen häviämiselleen on $q = 1 - p$. Jokaisella kierroksella voittaja saa yhden euron häviäjältä. Peliä pelataan siihen asti, kunnes toiselta pelaajalta loppuvat rahat. Kysymys kuuluu, millä todennäköisyydellä P_i pelaaja A voittaa, kun hänellä on alussa rahaa i €. [1.] Alla esitetään matemaattinen ratkaisu ongelmalle, mutta se ei ole mielekäs peruskoululaiselle, joten oppilaan tehtävänä on tehdä koodi, joka simuloi pelin kulkua, ja kertoo, kumpi pelaajista voitti, ja kuinka monta kierrosta peli kesti. Matemaattinen tarkastelu soveltuu ylös päin eriyttäväksi materiaaliksi, tai ainakin oppilaat voivat tutustua lopputulokseen.

Ongelma voidaan kuvata *satunnaiskulkuna* (*random walk*) välillä $0 < i < N$. Piste on aluksi x-akselilla kohdassa i, josta se liikkuu oikealle todennäköisyydellä p ja vasemmalle todennäköisyydellä $q = 1 - p$. Piste kuvaa pelaajan A rahamäärää. Siirtymä on pituudeltaan täsmälleen 1, paikallaan pysyminen ei ole mahdollista. Liike jatkuu toistuvin askelin ja loppuu, kun piste saavuttaa

x-akselin pisteen 0 tai N, eli pelaajan A rahat joko loppuvat, tai hän voittaa kaikki itselleen.

Olkoon R_n pelaajan A rahasumma, kun peliä on pelattu n kierrosta. $\{R_n : n \geq 0\}$ muodostaa satunnaiskulun $R_n = \Delta_1 + \Delta_2 + \ldots \Delta_n, R_0 = i$, missä Δ_n on riippumaton satunnaismuuttuja, jolle $P(\Delta = 1) = p, P(\Delta = -1) = q = 1 - p$.

Ajatellaan todennäköisyyslaskennan perusteiden mukaisesti, että otosavaruus Ω edustaa kaikkia alkeistapauksia, jotka voivat tapahtua, kun peli alkaa kohdasta i. Otosavaruus voidaan jakaa kahteen osaan, eli kaikkiin mahdollisiin lopputuloksiin, kun ensimmäinen tapahtuma on pelaajan A voitto, jolloin satunnaiskulku alkaa suuntaan $i+1$. Vastaavasti toinen osa otosavaruudesta muodostuu lopputuloksista, jotka ovat mahdollisia sen jälkeen, kun pelaaja A on hävinnyt ensimmäisen pelin, jolloin siis satunnaiskulku alkoi suuntaan $i-1$. Näissä molemmissa tapauksissa osa alkeistapauksista johtaa pelaajan A voittoon, ja osa häviöön. [1.] Otos-avaruutta selventää kuva 3.



Kuva 3. Otosavaruus Ω, kun peli aloitetaan kohdasta i. Seuraava askel on suuntaan i+1 todennäköisyydellä p, tai suuntaan i-1 todennäköisyydellä q. Molemmissa suunnissa pelaajan A voitto koko uhkapelissä on mahdollista.

Olkoon P_i todennäköisyys, että pelaaja A voittaa, kun hänen alkusummansa on i €. Tällöin $P_0 = 0$, sillä voittaminen on mahdotonta, jos alussa ei ole yhtään rahaa. Samoin voitto on täysin varma, jos koko potti on alussa pelaajalla A, eli $P_N = 1$. Tavoitteena on muotoilla lauseke todennäköisyydelle P_i, missä $1 \leq i \leq N - 1$. P_i muodostuu kahdesta otosavaruuden osajoukosta, eli $P(ensin\ A\ voittaa\ ja\ lopulta\ voittaa\ koko\ pelin)$ tai $P(ensin\ A\ häviää\ ja\ lopulta\ voittaa\ koko\ pelin)$. Todennäköisyys ensimmäiselle voitolle on p. Tämän jälkeen pelaajan rahasumma on R_{i+1} ja tässä tilanteessa koko pelin voittamisen todennäköisyys on P_{i+1}. Vastaavasti ajatellaan tilanne, jossa pelaaja A häviää ensimmäisen kierroksen. Näiden tapahtumien yhdistetty todennäköisyys voidaan ilmaista lausekkeella $P_i = pP_{i+1} + qP_{i-1}$. [1.]

Yhtälön vasen puoli voidaan ilmaista toisin, kun tiedetään, että $p + q = 1$. Tällöin saadaan $(p + q)P_i = pP_1 + qP_i = pP_{i+1} + qP_{i-1}$. Muokataan yhtälö muotoon $P_{i+1} - P_i = \frac{q}{p}(P_i - P_{i-1})$, missä $1 \leq i \leq N - 1$. Alkuehdon $P_0 = 0$ avulla todennäköisyyden lauseke voidaan ratkaista rekursiivisesti.

$$P_2 - P_1 = \frac{q}{p}(P_1 - P_0) = \frac{q}{p}P_1\,,$$
$$P_3 - P_2 = \frac{q}{p}(P_2 - P_1) = \frac{q}{p}\left(\frac{q}{p}P_1\right) = \left(\frac{q}{p}\right)^2 P_1\,,$$
$$P_{i+1} - P_i = \left(\frac{q}{p}\right)^i P_1, \qquad miss\ddot{a}\ 0 < i < N.$$

Ilmaistaan seuraavaksi $P_{i+1} - P_1$ summalausekkeen avulla

$$P_{i+1} - P_1 = \sum_{k=1}^{i} P_{k+1} - P_k = \sum_{k=1}^{i} \left(\frac{q}{p}\right)^k P_1\,.$$

Täten

$$P_{i+1} = P_1 + P_1 \sum_{k=1}^{i} \left(\frac{q}{p}\right)^k = P_1 \sum_{k=0}^{i} \left(\frac{q}{p}\right)^k.$$

Kyseessä on geometrinen summa, joten

$$P_{i+1} = \begin{cases} P_1 \dfrac{1 - \left(\frac{q}{p}\right)^{i+1}}{1 - \dfrac{q}{p}} & kun\ p \neq q \\ P_1 N & kun\ p = q = 0{,}5. \end{cases} \qquad (tulos\ 3.1.1)$$

Kun nyt valitaan $i = N - 1$, ja käytetään tietoa, että $P_N = 1$, saadaan

$$P_{N-1+1} = P_N = 1 = \begin{cases} P_1 \dfrac{1 - \left(\frac{q}{p}\right)^{N}}{1 - \dfrac{q}{p}} & kun\ p \neq q \\ P_1 N & kun\ p = q = 0{,}5. \end{cases}$$

Tästä voidaan ratkaista lauseke todennäköisyydelle P_1.

$$P_1 = \begin{cases} \dfrac{1 - \dfrac{q}{p}}{1 - \left(\frac{q}{p}\right)^N} & kun\ p \neq q \\ \dfrac{1}{N} & kun\ p = q = 0{,}5. \end{cases} \qquad (tulos\ 3.1.2)$$

Sijoitetaan nyt tulos 3.1.2 tulokseen 3.1.1, ja vaihdetaan indeksin $i+1$ tilalle selvyyden vuoksi i, jolloin saadaan lopullinen lauseke

$$P_i = \begin{cases} \dfrac{1-\left(\frac{q}{p}\right)^i}{1-\left(\frac{q}{p}\right)^N} & kun\ p \neq q \\ \dfrac{i}{N} & kun\ p = q = 0{,}5. \end{cases}$$

Otetaan esimerkki tilanteesta, jossa pelaajan A yksittäisen pelin voittotodennäköisyys p on 0,6. Hänellä on aluksi 3 €, ja koko potti on 10 €. Voittotodennäköisyydeksi P_3 saadaan

$$P_3 = \frac{1-\left(\frac{q}{p}\right)^i}{1-\left(\frac{q}{p}\right)^N} = \frac{1-\left(\frac{0{,}4}{0{,}6}\right)^3}{1-\left(\frac{0{,}4}{0{,}6}\right)^{10}} = 0{,}72.$$

Jos pelaajien voittotodennäköisyydet olisivat yhtä suuret, eli $p = q = 0{,}5$ pelaajan A voittotodennäköisyys samasta lähtötilanteesta olisi merkittävästi huonompi,

$$P_3 = \frac{i}{N} = \frac{3}{10} = 0{,}30.$$

3.2 Monty Hall

3.2.1 Monty Hall -ongelman historia

Matematiikan Monty Hall- ongelma on saanut nimensä amerikkalaisesta 1960-luvulla alkaneesta TV-sarjasta Let's make a deal, jota on esitetty erilaisina versioina vielä 2000-luvullakin. Ohjelma on nostettu korkeille sijoille erilaisissa kautta aikain parhaat TV-kisailut- listoissa. Ohjelman suosio oli suurin kolmena ensimmäisenä vuosikymmenenä, jolloin juontajana toimi Monty Hall. Ideana oli, että kilpailija vietiin kolmen oven eteen, joista yhden takana oli arvokas palkinto, ja kahden muun takana täysin arvottomat tuotteet. Kilpailija valitsi ensin yhden oven, jonka jälkeen juontaja poisti jäljelle jääneistä kahdesta ovesta sen, jossa oli arvoton palkinto. Nyt kilpailija sai päättää, ottaako hän alun perin valitsemansa oven, vai vaihtaako valintansa. Matemaattisesti on selvää, kummalla tavalla toimiminen on järkevämpää.

Tämä peli on hyvä esimerkki matematiikan ongelmasta, jolla voi herättää oppilaiden kiinnostuksen todennäköisyyslaskentaan. Tehtävän käyttö TV-sarjassa tuo sille mielenkiintoisen historian, ja peliä voi pelata luokassakin vaikkapa tikkukaramellilla ja rikkinäisellä sukkaparilla. Kun Monty Hall -ongelma esitetään matematiikan maailmassa, palkintovaihtoehtoina on tyypillisesti tuliterä auto ja kaksi vuohta, joita käytetään esimerkkeinä tässäkin työssä. Tässä työssä laajennetaan Monty Hall- ongelman tarkastelua pelkästä klassisen todennäköisyyslaskennan näkökulmasta ongelman tarkasteluun simulaation avulla.

3.2.2 Monty Hall -ongelma klassinen todennäköisyyden esimerkkinä

Monty Hall-ongelmaa voidaan käsitellä klassisen todennäköisyyslaskennan keinoin, sillä alkeistapauksia on äärellinen määrä, ja arvokkain voitto sijaitsee minkä tahansa oven takana yhtä suurella todennäköisyydellä. Palkintoina olevat kaksi vuohta ja auto ovat toisensa poissulkevia tapahtumia, sillä yhdenkään oven takana ei ole kahta palkintoa.

Monty Hall- ongelmaa on helpoin lähestyä taulukoimalla mahdolliset palkinto-ovien kombinaatiot ja tarkastelemalla rivi kerrallaan, mitä pelaaja voittaa, jos pysyy alkuperäisessä päätöksessään, tai vaihtaa päätöstään sen jälkeen, kun pelin järjestäjä on poistanut yhden oven, jonka takana ei ole palkintoa. Taulukossa 1 esitetään kaikki mahdolliset tilanteet, kun pelaaja valitsee ensimmäisenä päätöksenään ensimmäisen oven. Jos ensimmäinen valinta olisi toinen tai kolmas ovi, tarkastelu olisi täysin identtinen.

Taulukko 1. Pelin kulku kaikissa eri kombinaatioissa, kun pelaaja valitsee ensimmäisessä vaiheessa oven 1. Tarkastelu on samanlainen, jos ensimmäinen valinta olisi joku muu ovi.

Ovi 1	**Ovi 2**	**Ovi 3**	**Palkinto, jos pysytään valinnassa ovi 1**	**Palkinto, jos päätös vaihdetaan**
Auto	Vuohi	Vuohi	Auto	Vuohi
Vuohi	Auto	Vuohi	Vuohi	Auto
Vuohi	Vuohi	Auto	Vuohi	Auto
Auton voittamisen todennäköisyys			$\frac{1}{3}$	$\frac{2}{3}$

Taulukosta 1 voidaan nähdä, että oikea strategia on vaihtaa päätöstä, silloin voiton mahdollisuus on kaksinkertainen pysymisstrategiaan verrattuna.

Vaihtoehtoinen tapa päätyä samaan tulokseen on tarkastella yhdistettyjä todennäköisyyksiä. Jos pelaaja aikoo pysyä alkuperäisessä valinnassaan, todennäköisyys auton voittamiselle on $P(auto) = \frac{suotuisat\ alkeistapaukset}{kaikki\ alkeistapaukset} = \frac{1}{3}$.

Jos taas valittu strategia on vaihtaa päätöstä, voiton todennäköisyys muodostuu yhdistetystä todennäköisyydestä, että pelaaja valitsee ensin vuohen, ja sitten auton. Toisessa vaiheessa auton todennäköisyys on 1, sillä toinen vuohiovi poistetaan. Näin ollen $P(auto) = P(vuohi) \cap P(auto) = \frac{2}{3} \cdot 1 = \frac{2}{3}$.

3.2.3 Monty Hall -ongelma yleistettynä useamman oven tapaukseen

Monty Hall -ongelma voidaan yleistää tilanteeseen, jossa ovia on enemmän. Näissä versioissa kilpailijan ensimmäisen valinnan jälkeen pelin järjestäjä ottaa jäljelle jääneistä ovista pois yhden, jonka takana ei ole palkintoa. Tässä vaiheessa kilpailijan on mahdollista vaihtaa päätöstään, tai pysyä alkuperäisessä. Uuden päätöksen jälkeen järjestäjä ottaa taas yhden oven pois, ja tällä tavoin jatketaan, kunnes jäljellä on vain kaksi ovea. Kun ovia on alun perin n kappaletta, pelaaja tekee $n - 1$ päätöstä pelin kuluessa. Taulukossa 2 tarkastellaan peliä neljän oven tapauksessa. Tarkasteltavana on strategiat, joissa pelaaja pysyy koko pelin ajan alkuperäisessä päätöksessään, tai vaihtaa päätöstään vain viimeisessä arvauksessa. Strategia, jossa pelaaja vaihtaisi päätöstään jokaisella kierroksella johtaa voittoon, jos pelaaja tekee viimeisessä vaiheessa oikean arvauksen, ja aikaisempien kierrosten päätöksillä ei ole mitään merkitystä. Näin ollen satunnaisen strategian voittotodennäköisyys on $\frac{1}{2}$. Samoin kuin kolmen oven tapauksessa, tässäkin riittää tarkastella tilanne, jossa pelaaja valitsee aluksi ensimmäisen oven. Muut valinnat alussa johtavat identtiseen tarkasteluun.

Taulukko 2.Monty Hall -ongelma neljän oven tapauksessa. Oletuksena on, että pelaajan ensimmäinen valinta on ovi 1. Tarkastelu on samanlainen, jos ensimmäinen valinta olisi joku muu ovi.

Ovi 1	**Ovi 2**	**Ovi 3**	**Ovi 4**	**Palkinto, jos pysytään aina valinnassa ovi 1**	**Palkinto, jos päätös vaihdetaan viimeisessä arvauksessa**
Auto	Vuohi	Vuohi	Vuohi	Auto	Vuohi
Vuohi	Auto	Vuohi	Vuohi	Vuohi	Auto
Vuohi	Vuohi	Auto	Vuohi	Vuohi	Auto
Vuohi	Vuohi	Vuohi	Auto	Vuohi	Auto
Auton voittamisen todennäköisyys				$\frac{1}{4}$	$\frac{3}{4}$

Kun pelissä on *n* ovea, voittotodennäköisyys on $\frac{1}{n}$ jos pelaaja pysyy alkuperäisessä päätöksessään. Jos hän pysyy alkuperäisessä päätöksessään, ja vaihtaa sen viimeisessä vaiheessa, kun jäljellä on enää kaksi ovea, voittotodennäköisyys on $\frac{n-1}{n}$. Satunnaisen strategian voittotodennäköisyys on $\frac{1}{2}$ riippumatta ovien määrästä.

3.3 Vasta-ainetesti ehdollisen todennäköisyyden sovellutuksena

Verikokeiden luotettavuuteen ja tulosten tulkitsemiseen liittyy todennäköisyyslaskenta, ja ne ovatkin hyvä sovelluskohde, joita tutkimalla oppilaita voi johdatella ymmärtämään todennäköisyyslaskennan käyttökohteita ja merkitystä. Otetaan esimerkiksi vasta-ainetesti, jolla tutkitaan verinäytteestä, esiintyykö henkilöllä vasta-aineita virusta vastaan. Satunnaisotostutkimuksen tulosten perusteella on tarkoitus pystyä päättelemään, kuinka suurella osalla väestöstä on vasta-aineita, eli kuinka suuri osa väestöstä on sairastanut viruksen aiheuttaman taudin. [1.] Käytännössä kaikki verikoetestit antavat satunnaisesti vääriä tuloksia, ja ne jaetaan kahteen joukkoon, väärät negatiiviset ja väärät positiiviset. Tämä aiheuttaa sen, että todennäköisyyslaskentaa tarvitaan joukkotestauksen tulosten tulkintaan. Koska vuoden 2020 korona-pandemia vaikutti merkittävästi jokaisen koululaisen elämään, käsiteltäväksi esimerkiksi on valittu COVID-19-viruksen vasta-ainetesti.

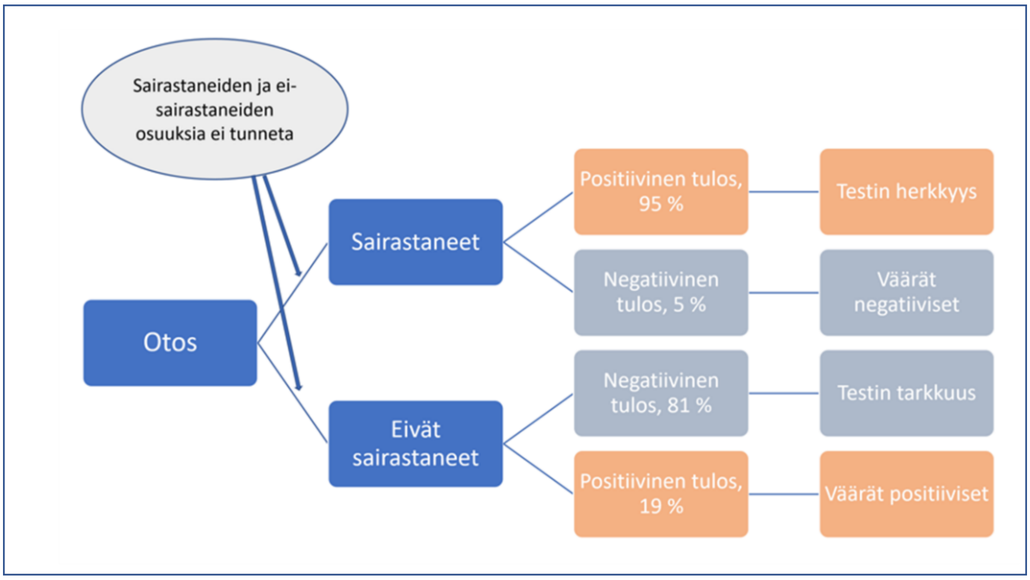
Nimetään ensin tapahtumat A = ”henkilöllä on vasta-aineita” ja B = ”testi on positiivinen”. Lääketieteessä on käytössä termi testin herkkyys, ja se on ehdollisen todennäköisyyden merkinnöillä $P(B|A)$, eli se kertoo, kuinka suurella osalla testattavista testitulos on positiivinen, jos henkilöllä on vasta-aineita. Herkkyysarvo saadaan selville tekemällä testi joukolle ihmisiä, joilla tiedetään olevan vasta-aineita. [1.]

Väärien negatiivisten todennäköisyys saadaan tutkittaessa herkkyyttä. Jos siis verikoe antaa positiivisen tuloksen todennäköisyydellä 95 %, kun otoksessa kaikilla koehenkilöillä on vasta-ainetta, todennäköisyys väärälle negatiiviselle tulokselle on komplementti herkkyydelle, eli 5 %.

Väärä positiivinen tulos tarkoittaa tilannetta, jossa testi on positiivinen, mutta henkilöllä ei ole vasta-ainetta, eli ehdollisen todennäköisyyden avulla ilmaistuna $P(B|\bar{A})$. Tämän arvon komplementtia kutsutaan testin tarkkuudeksi, eli se osuus ei-sairastaneista, joille testi antaa oikeellisesti negatiivisen tuloksen.

Testin valmistaja ilmoittaa käytännössä herkkyys- ja tarkkuusarvot. Suomessa joukkotestauksissa kaikki positiiviset tulokset testataan uudestaan erimerkkisellä testillä, jolloin väärien positiivisten määrää saadaan pienennettyä.

Kuvassa 4 esitetään verikoetestin herkkyyden ja tarkkuuden merkitykset kuvan avulla, ja annetaan esimerkkiarvot Abbottin testistä, jota käytetään Suomessa COVID-19-viruksen vasta-ainetesteissä ensisijaisena testinä [2].



Kuva 4. Verikokeessa negatiivisia ja positiivisia tuloksia saadaan sekä sairastaneiden että ei-sairastaneiden joukosta.

Kuvan 4 avulla voidaan ymmärtää, että vaikka tunnetaan testin antamien positiivisten ja negatiivisten tapausten lukumäärät, tiedon avulla ei pystytä selvittämään, kuinka suuri osa väestöstä on sairastanut viruksen. Positiiviset tulokset tulevat oikeista positiivisista ja vääristä positiivista, eikä yhteislukumäärästä pysty päättelemään, kuinka väestö jakautuu taudin sairastaneisiin ja sairastamattomiin.

Kun halutaan selvittää, kuinka hyvän kuvan väestön satunnainen testaaminen antaa taudin levinneisyydestä, muotoilemme kysymyksen, *mikä on todennäköisyys, että henkilöllä on vasta-ainetta, kun testi antaa positiivisen tuloksen*. Tämä saadaan selville Bayesin kaavan avulla $P(A|B) = \frac{P(B|A)P(A)}{P(B)}$. Nimittäjässä oleva *P(B),* eli positiivisen tuloksen todennäköisyys koostuu oikeista positiivisista ja vääristä positiivisista tuloksista, eli $P(B) = P(A)P(B|A) + P(\bar{A})P(B|\bar{A})$. [1.] Näin saadaan

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(A)P(B|A) + P(\bar{A})P(B|\bar{A})} \; (tulos\ 3.5.1).$$

Tehdään esimerkkitarkastelu kuvassa 4 esitetyillä arvoilla. Vasta-ainetestin herkkyys $P(B|A)$ on 0,95 ja tarkkuuden komplementti $P(B|\bar{A})$ on 0,19. Oletetaan, että taudin levinneisyys *P(A)* on 0,15, mikä tarkoittaa, että sen komplementti, eli tautia sairastamattomien osuus olisi 0,85. Sijoittamalla arvot tulokseen 3.5.1 saadaan laskettua ehdollinen todennäköisyys sille, että henkilöllä on vasta-ainetta veressään, kun hän on saanut positiivisen testituloksen.

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(A)P(B|A) + P(\bar{A})P(B|\bar{A})} = \frac{0{,}95 \cdot 0{,}15}{0{,}15 \cdot 0{,}95 + 0{,}85 \cdot 0{,}19} = 0{,}41$$

Todennäköisyys on siis alle puolet, vaikka testitulos oli positiivinen. Kappaleessa 5.3.2 tarkastellaan tilannetta yksityiskohtaisemmin, erityisesti levinneisyyden vaikutusta tulokseen.

4 Simuloinnit ja mallinnukset todennäköisyyslaskennassa

Tässä luvussa käsitellään lyhyesti tietokoneiden ja simulointien hyödyntämistä todennäköisyyslaskennassa. Yksinkertaisimmillaan simulointi voi olla satunnaiskokeen toistamista tietokoneella, jolloin toistoja saadaan helposti suuri määrä, ja voidaan tutkia, mikä vaikuttaisi olevan suhteellisen frekvenssin raja-arvo eri tapahtumille.

Simulointimalleja muodostetaan kuvaamaan todellisen systeemin käyttäytymistä. Ensin reaalimaailman ilmiöstä tehdään konseptionaalinen havainnemalli, ja sen jälkeen tietokone voidaan laittaa simuloimaan systeemin käyttäytymistä erilaisilla lähtöarvoilla. Monissa ilmiöissä täsmällinen matemaattinen mallintaminen on haastavaa ja vaatii yksinkertaistuksia. Monty Hall -ongelmasta tekee yksinkertaisen ennen kaikkea se, että tapahtumat eivät ole ajasta riippuvia, ja muuttujat ovat diskreettejä. Jos tapahtuman ajankohdalla olisi merkitystä lopputulokselle, simulaatiossa pitäisi hyödyntää differentiaali- tai osittaisdifferentiaali-yhtälöitä. [6.] Tilastollisia malleja hyödynnetään esimerkiksi virusepidemian leviämisen mallintamisessa ja ennustamisessa. Syötteinä mallissa on ilmiöstä kerättyjä todellisia havaintoja, ja mallin tavoitteena on saada kuvattua kaikki mahdolliset tulosvaihtoehdot, ja todennäköisyydet niille. [3.]

Systeemit voidaan jakaa stokastisiin ja deterministisiin systeemeihin, joista stokastisissa systeemeissä satunnaisuudella on merkitystä, ja vastaavasti deterministisissä systeemeissä lopputila on täysin laskettavissa, kun alkuarvot tiedetään [6]. Esimerkki stokastisesta systeemistä on yksittäisen asiakkaan jonotusaika palvelupisteellä, kun asiakkaiden saapumisaikoja ei tunneta etukäteen. Uhkapelurit ja Monty Hall -ongelma ovat stokastisia systeemejä, koska niissä sattumalla on merkitystä lopputulokseen, ja samasta alkuasetelmasta lähdettäessä lopputulokselle on eri vaihtoehtoja.

5 Esimerkkiratkaisut simulointi- ja mallinnustehtäviin

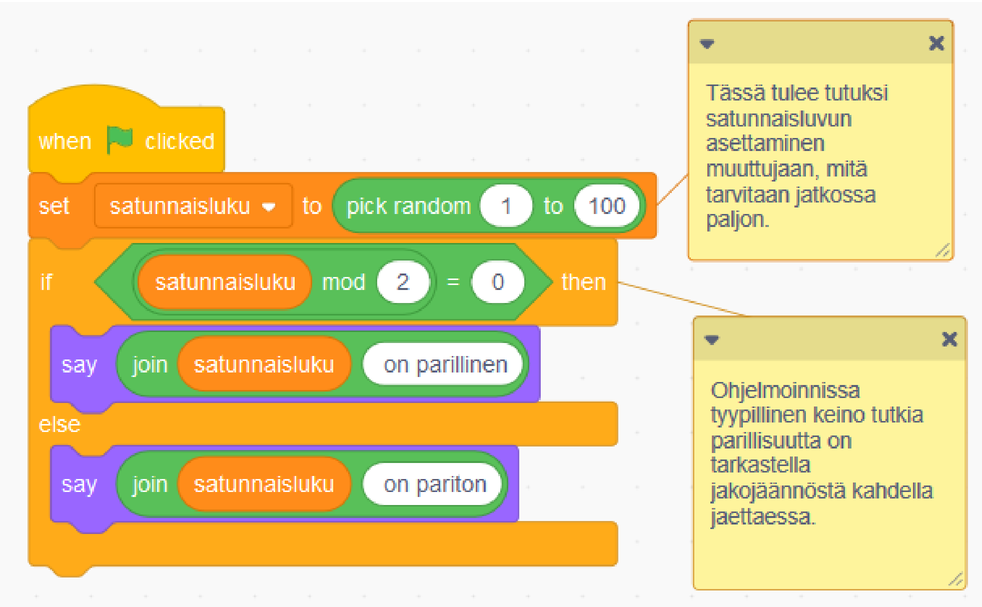
5.1 Uhkapelurit

5.1.1 Scratch ohjelmointiympäristönä

Sekä uhkapelureissa että Monty Hall-ongelmassa simulointi on tehty Scratch-ympäristössä. Scratch perustuu visuaaliseen ohjelmointiin, ja sitä voi käyttää suoraan selaimessa. Visuaalinen ohjelmointi on nopea tapa päästä ohjelmoimaan. Se mahdollistaa, että aloittelijakin pääsee perehtymään ensisijaisesti algoritmiseen rakenteeseen sen sijaan, että ensin pitäisi opetella kielen syntaksi.

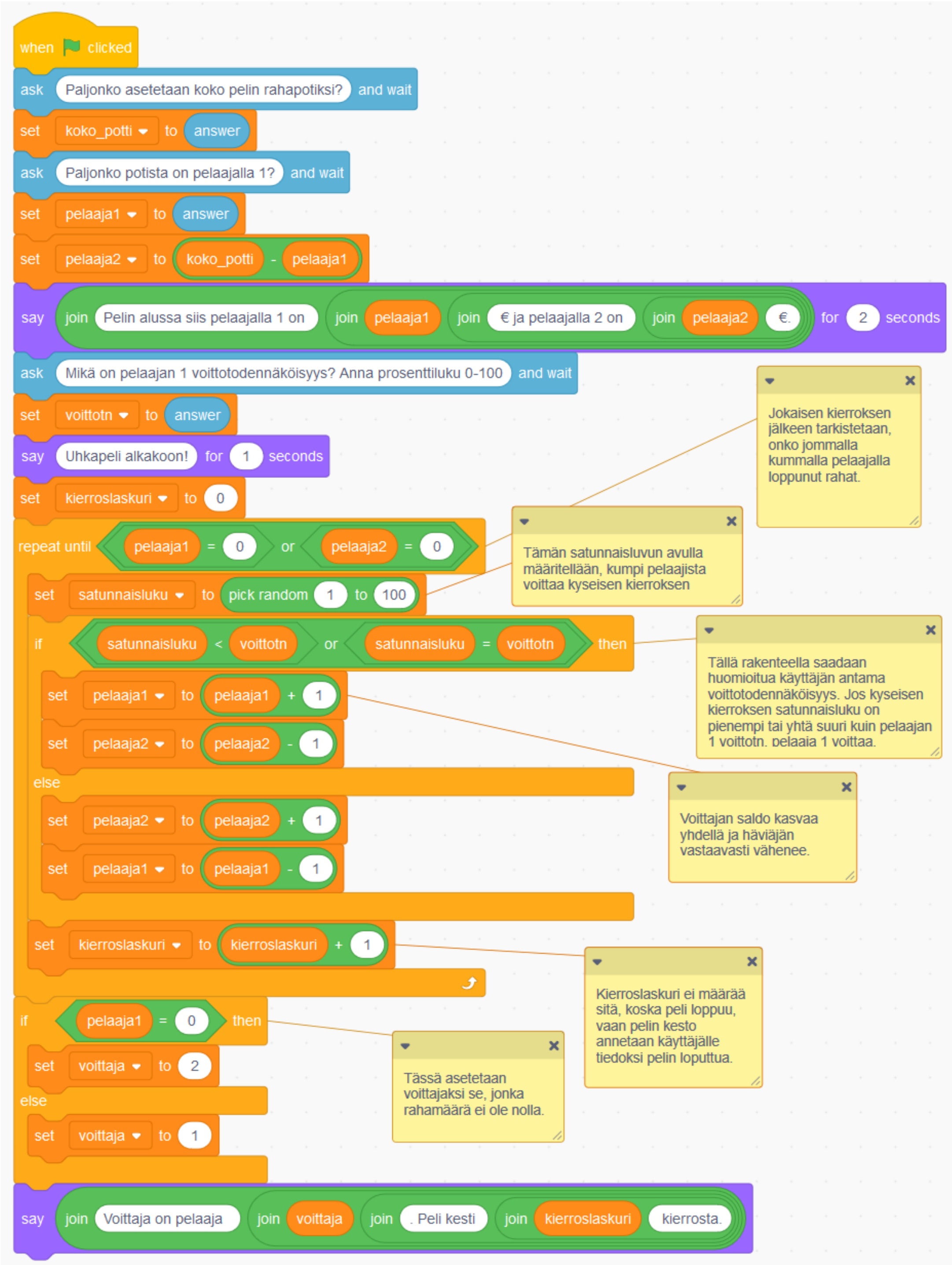
5.1.2 Simulaatio uhkapelistä

Tätä simulaatiota koodatessa oppilas tutustuu muuttujien käyttämiseen, satunnaisluvun hyödyntämiseen, ehtolauseeseen ja silmukkarakenteeseen. Oppilasohjeet on esitetty liitteessä 1. Aivan aluksi, jos oppilaiden kokemukset ohjelmoinnista ovat vähäisiä, voi tehdä yksinkertaisen harjoituksen, jossa arvotaan satunnaisluku väliltä 1-100, ja tutkitaan, onko luku parillinen vai pariton. Esimerkki tästä on kuvassa 5.



Kuva 5. Lyhyt harjoitus, jossa oppilas tutustuu satunnaisluvun hyödyntämiseen ja tutkii, onko arvottu satunnaisluku parillinen vai pariton.

Esimerkki varsinaisesti uhkapelurit-simulaatiosta on kuvassa 6. Haasteellisin osuus lienee se, miten pelaajan voittotodennäköisyyden voi huomioida. Tässä se on ratkaistu siten, että arvotaan satunnaisluku väliltä 1-100. Jos saatu satunnaisluku on pienempi tai yhtä suuri kuin pelaajalle 1 asetettu voittotodennäköisyys, tämä tarkoittaa pelaajan 1 voittoa.



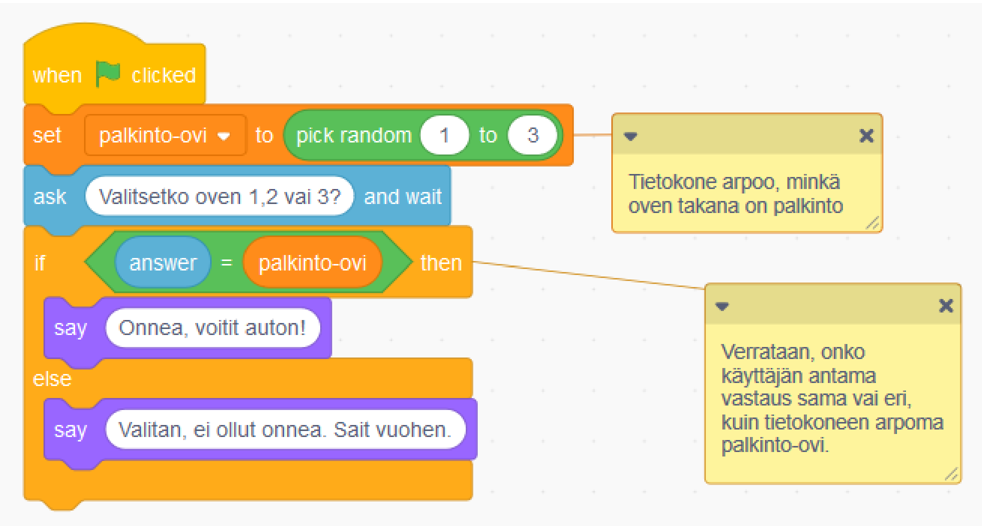
Kuva 6. Esimerkki koodista uhkapelurit-simulaatioon.

5.2 Monty Hall

5.2.1 Kolme ovea, käyttäjä tekee arvauksen

Tässä luvussa esitetään, millaisina paloina Monty Hall- ongelman simuloimisen voi antaa oppilaille, ja mitä todennäköisyyslaskennan käsitteitä kussakin vaiheessa tulee esiin. Oppilasohjeet on esitetty liitteessä 2. Oppilaiden eriyttäminen tapahtuu itsestään, kun nopeammin etenevät oppilaat pääsevät etenemään pidemmälle. Taitavimmat oppilaat voivat edetä itsenäisesti, kun apua tarvitsevien kanssa hankalimmat ohjelmointiin liittyvät seikat voidaan tehdä yhdessä edeten, ja jättää oppilaiden itse mietittäväksi vain valitut, sopivan tasoiset haasteet.

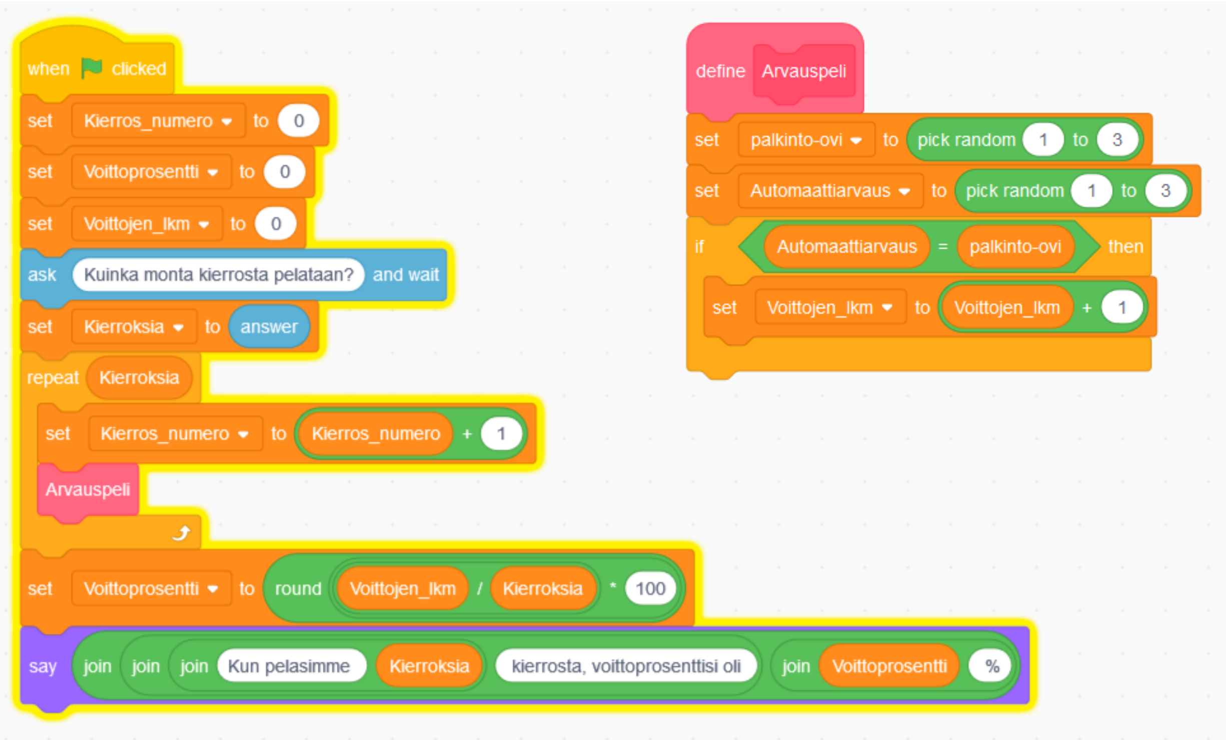
Ensimmäisessä vaiheessa toteutetaan simulointi, jossa ohjelma määrittää satunnaisesti, minkä oven taakse palkinto sijoitetaan. Käyttäjä tekee arvauksen ja kone kertoo, menikö oikein. Oppilas tutustuu satunnaisuuden käsitteeseen, ja ohjelmoinnin puolelta muuttuja-käsitteeseen ja if-lauseisiin. Esimerkki toteutuksesta on kuvassa 7.



Kuva 7. Yksinkertaisin tilanne, jossa tutkitaan, onko käyttäjän antama syöte sama kuin ohjelman arpoma.

5.2.2 Lasketaan voittoprosentti

Toisessa vaiheessa tehdään laskuri, jonka avulla lasketaan usean kierroksen voittoprosentti. Käyttäjä määrittelee, montako kierrosta pelissä on, ja kierrokset pelataan automaattisesti, eli kone simuloi ne. Tässä oppilas tutustuu laskurin käyttämiseen ohjelmoinnissa ja silmukka-rakenteeseen. Esimerkki toteutuksesta on kuvassa 8. Tässä varsinainen arvauspeli on toteutettu aliohjelmana, mutta kuten aina ohjelmoinnissa, toteutuksen voi tehdä monella eri tavalla.



Kuva 8. Koodi, jolla lasketaan voittoprosentti, kun simuloidaan käyttäjän antama kierrosmäärä.

Tässä vaiheessa oppilas pääsee tutkimaan, kuinka monella pelikierroksella voittoprosentti lähestyy teoreettista arvoa. Näin päästään empiirisesti toteamaan, kuinka pienillä toistomäärillä voittoprosentti poikkeaa satunnaisesti teoreettisesta arvosta välillä hyvinkin paljon, mutta toistomäärien kasvaessa riittävästi se saa teoreettista todennäköisyyttä lähellä olevia arvoja [3].

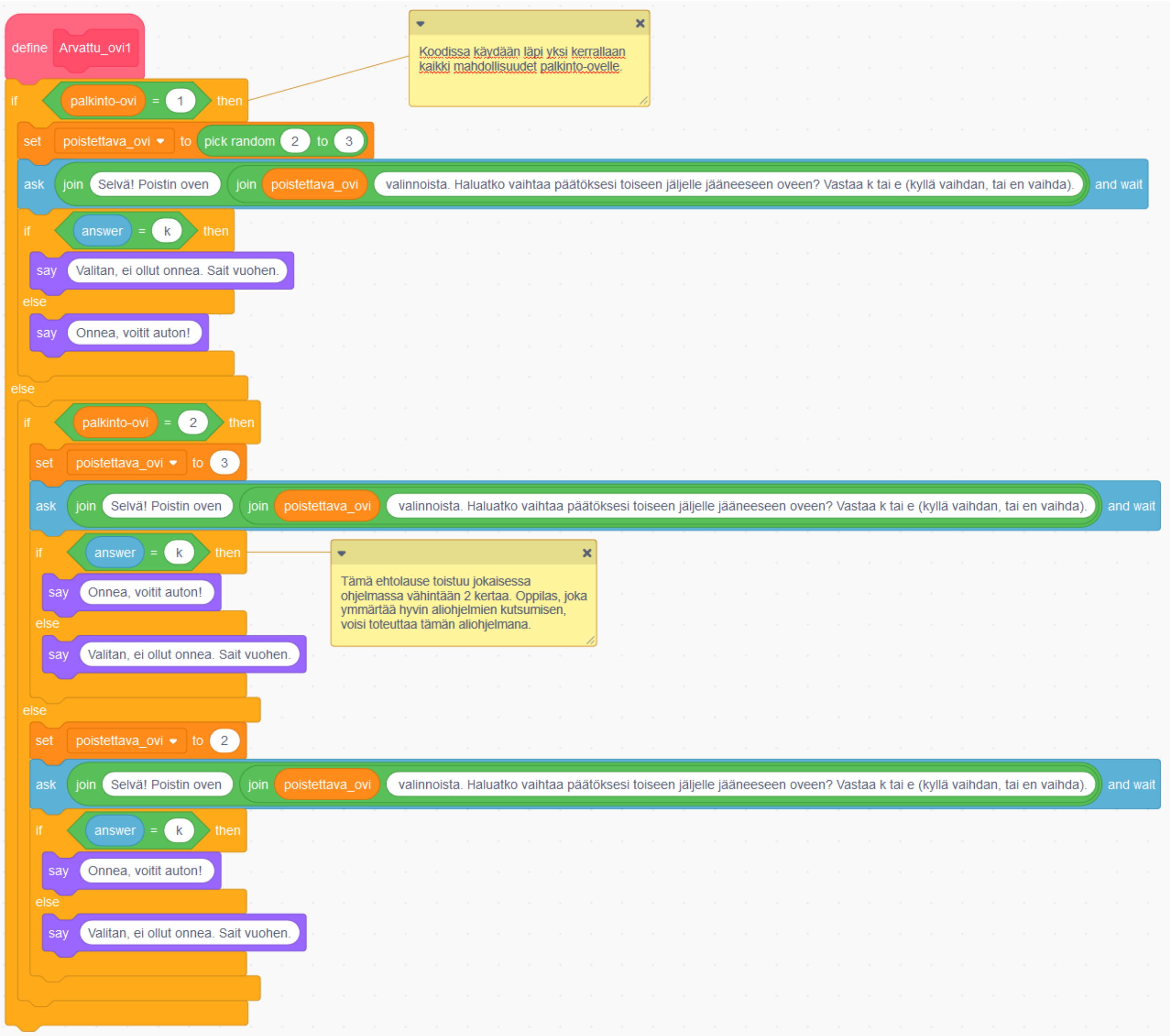
5.2.3 Kokonainen Monty Hall -simulaatio, yksi kierros

Tässä vaiheessa lisätään simulaatioon mahdollisuus vaihtaa ovea, eli peli ottaa pois yhden väärän oven ensimmäisen valinnan jälkeen. Nyt siis käyttäjä pelaa yhden kokonaisen kierroksen Monty Hall -arvauspeliä, tekee pelin aikana kaksi valintaa, ja kuulee lopulta, onko saanut auton vai vuohen.

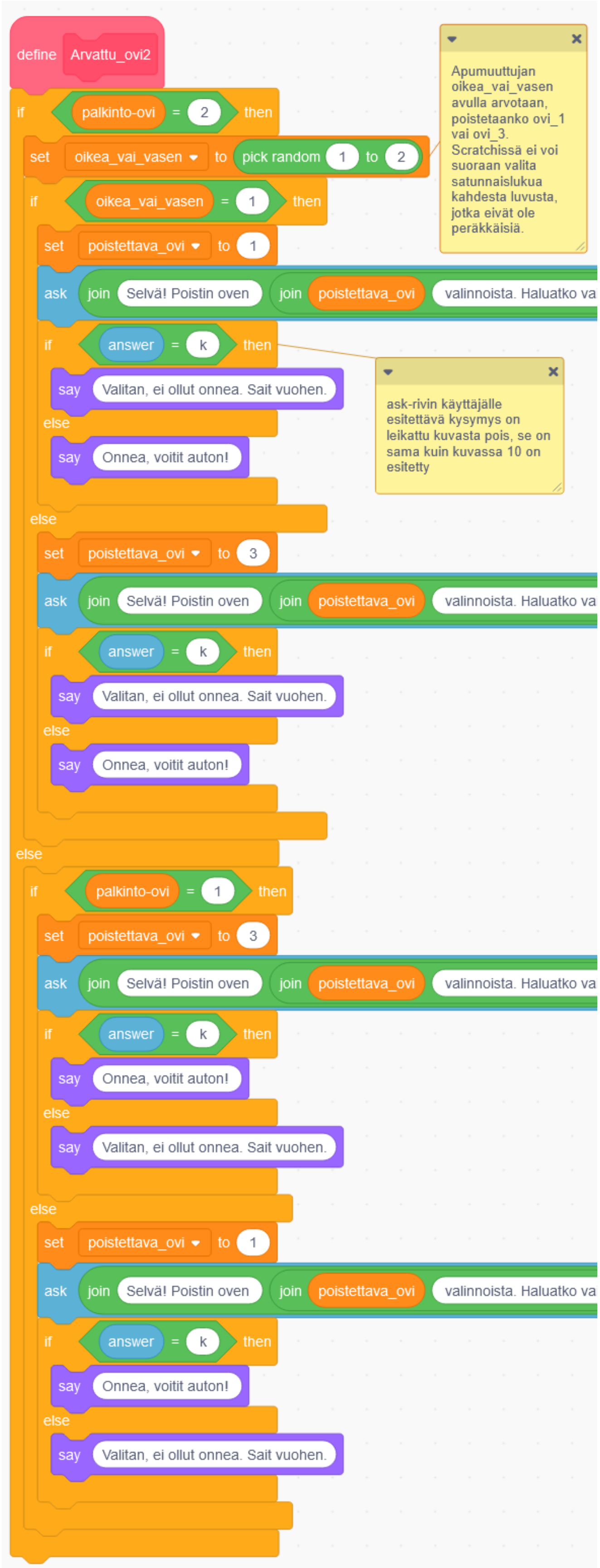
Kuvassa 9 esitetään lyhyt pääohjelma, joka kutsuu käyttäjän valinnasta riippuen jotakin kolmesta aliohjelmasta, jotka ovat lähes identtisiä. Taitava koodari saattaisi osata toteuttaa pelin lyhyemmin, mutta tässä oppilas pääsee toiston kautta oppimaan, miten työskentelee systemaattisesti erilaisten kombinaatioiden parissa.

Kuva 9. Pääohjelma, joka käyttäjän syötteestä riippuen kutsuu tiettyä aliohjelmaa.

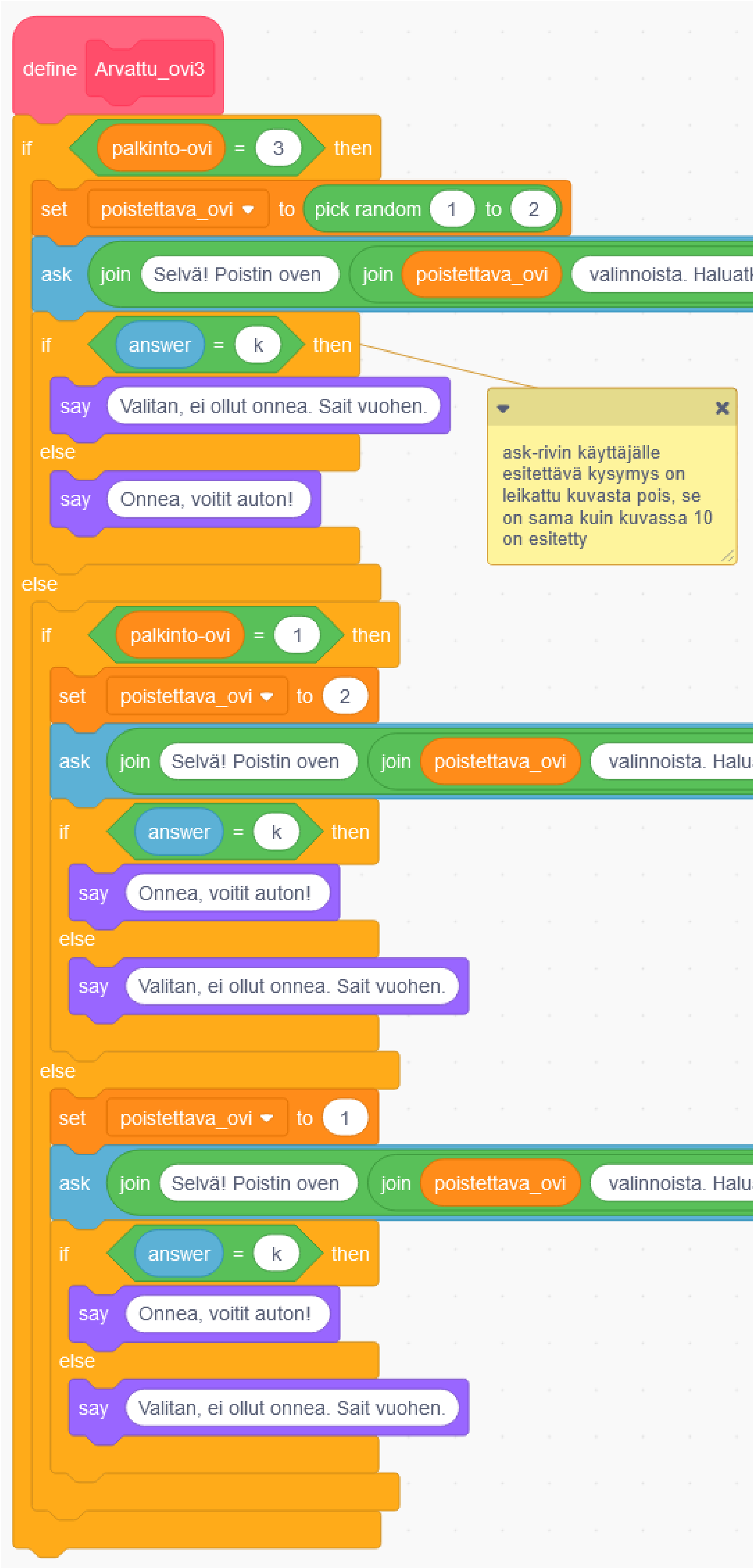
Aliohjelmissa käytetyssä rakenteessa on paljon sisäkkäisiä if-lauseita, joiden toiminnan oppiminen on ohjelmoinnin tärkeimpiä perusteita, vaikkakaan ei tuota lyhyintä mahdollista koodia. Kuvissa 10-12 esitetään aliohjelmat, joista yhtä kutsutaan, riippuen ensimmäisestä oven valinnasta.



Kuva 10. Aliohjelma Arvattu_ovi1.



Kuva 11. Aliohjelma Arvattu_ovi2. Muut kaksi aliohjelmaa ovat lähes samanlaisia, mutta apumuuttujaa oikea_vai_vasen ei niissä tarvita.



Kuva 12. Aliohjelma Arvattu_ovi3.

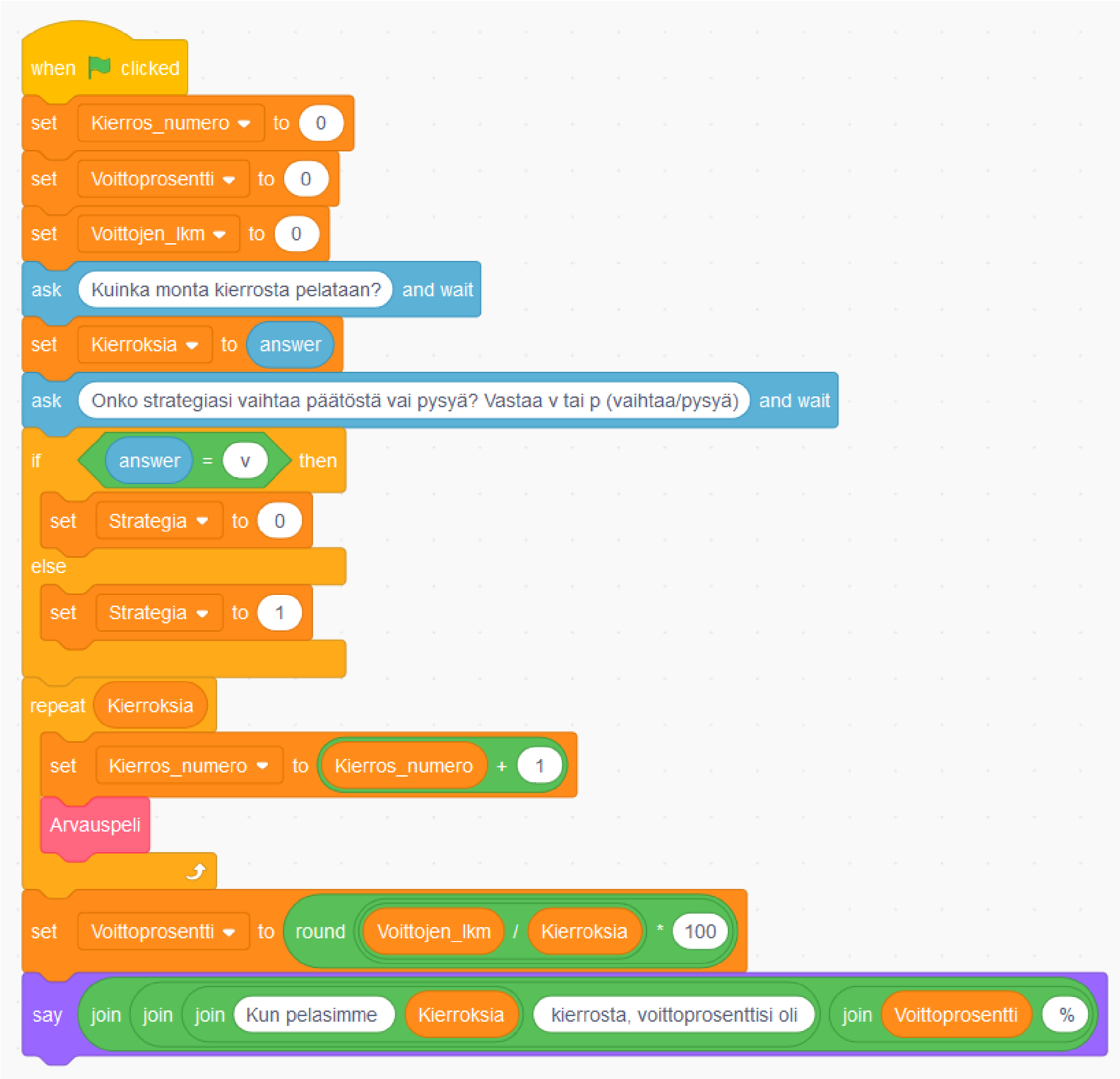
On syytä selventää Arvattu_ovi2 -aliohjelman koodissa esiintyvää muuttujaa oikea_vai_vasen, joka saa sattumanvaraisesti arvon 1 tai 2. Tätä muuttujaa tarvitaan, kun poistettava ovi pitää arpoa ovien numero 1 ja 3 väliltä, sillä Scratchissä pystyy arpomaan satunnaisen lukuarvon vain peräkkäisistä luvuista. Tätä apumuuttujaa käytetään siten, että arvon ollessa 1, poistetaan vasemmanpuoleinen ovi, eli ovi numero 1 ja arvon ollessa 2 valitaan poistettavaksi oikeanpuoleinen ovi, eli ovi numero 3.

5.2.4 Monty Hall -simulaatio, voittoprosentin laskeminen

Edellisen vaiheen, jossa simuloidaan kokonainen peli, voi jättää tekemättä, jos tavoitteet ovat enemmän matematiikan kuin ohjelmoinnin oppimisessa. Seuraavassa vaiheessa simuloidaan automaattisesti useita kierroksia kerrallaan, ja päästään tutkimaan, kuinka suurilla kierrosmäärillä simulaation tulokset vastaavat klassisen todennäköisyyslaskennan avulla tehtyä päätelmää.

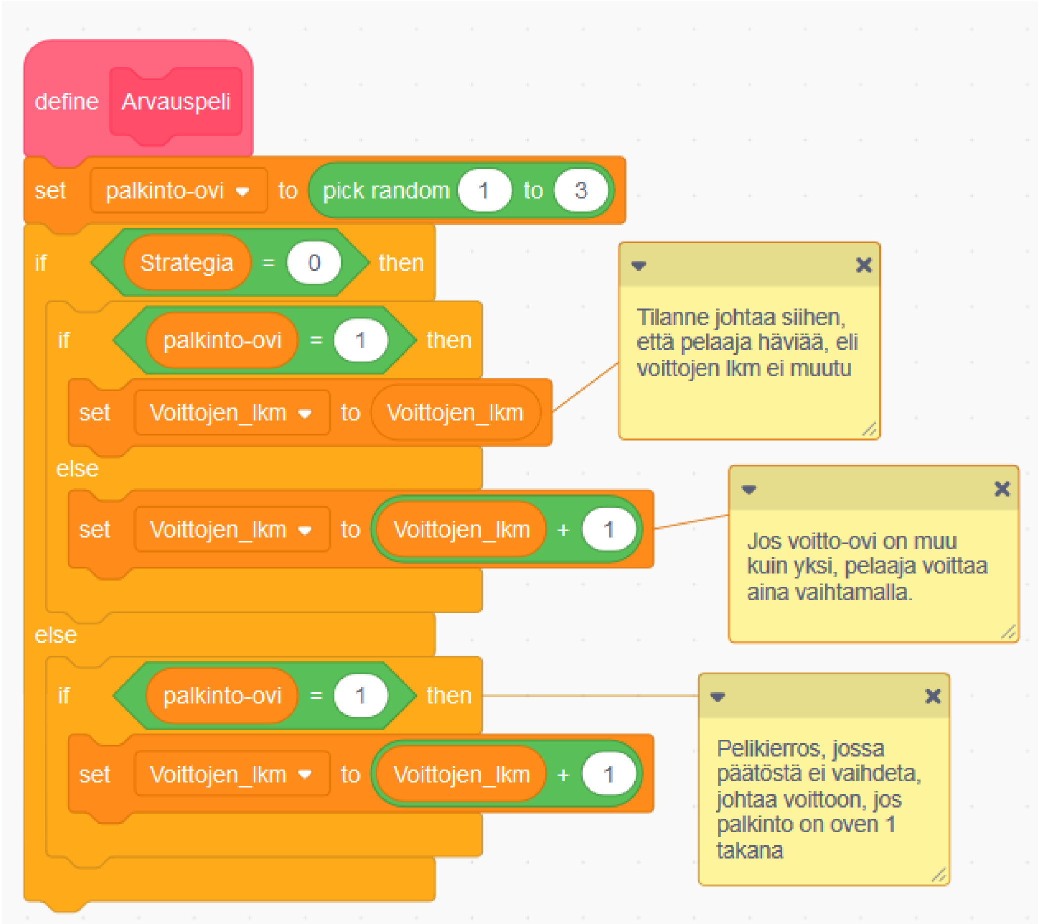
Tässä versiossa on tehty yksinkertaistus, että käyttäjä valitsee aina ensin oven numero 1. Kaikki vaihtoehdot voisi pitää mukana, kuten luvussa 5.2.3, mutta se tekisi koodista vain pidemmän lisäämättä matemaattista arvoa mitenkään.

Kuvassa 13 esitetään pääohjelman koodi peliin, jossa pelaaja valitsee, aikooko pysyä alkuperäisessä päätöksessään joka kierroksella, vai tekeekö vaihdon. Strategiaa ei voi muuttaa kesken simulaation. Käyttäjä saa lopuksi tietoonsa simulaation laskeman voittoprosentin, ja pääsee näin tutkimaan, kuinka paljon kierroksia tarvitaan, jotta voittoprosentti on sama kuin klassisen todennäköisyyden avulla pääteltynä.



Kuva 13. Pääohjelma voittoprosentin laskemiseen pelattaessa Monty Hall -simulaatiota.

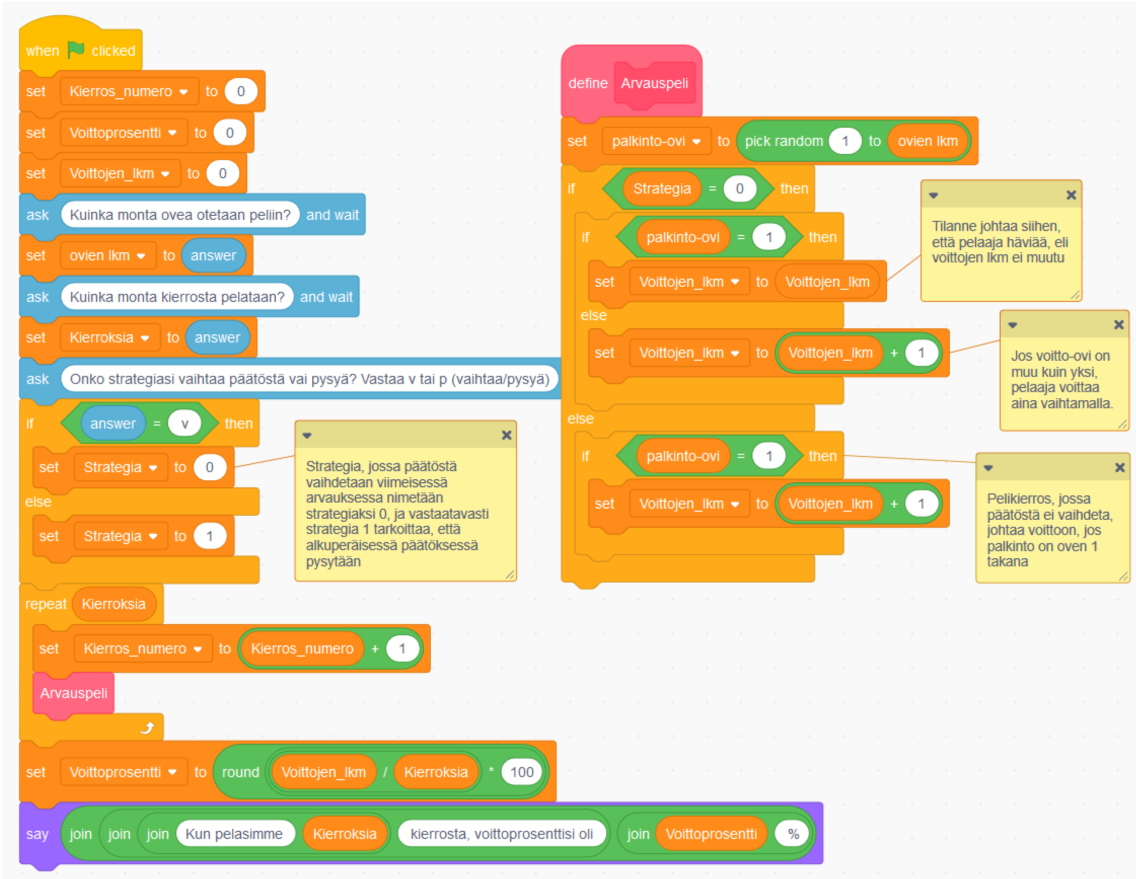
Kuvassa 14 on aliohjelma, jossa simuloidaan varsinaiset pelikierrokset.



Kuva 14. Aliohjelma, jossa simuloidaan pelikierrokset tapauksissa, joissa pelaaja vaihtaa päätöstään (strategia 0), tai pysyy alkuperäisessä päätöksessään (strategia 1).

5.2.5 Monty Hall -pelin voittoprosentti, kun pelaaja määrittelee ovien lukumäärän

Viimeinen vaihe on nopeimmille oppilaille. Käyttäjä valitsee ovien määrän, ja simulaatio laskee voittoprosentin. Käyttäjä valitsee taas alussa, aikooko pysyä alkuperäisessä päätöksessään, vai vaihtaa päätöstä viimeisessä valinnassa. Kuvassa 15 esitetään koodi simulaatioon, jossa pelissä on aluksi käyttäjän valitsema määrä ovia.

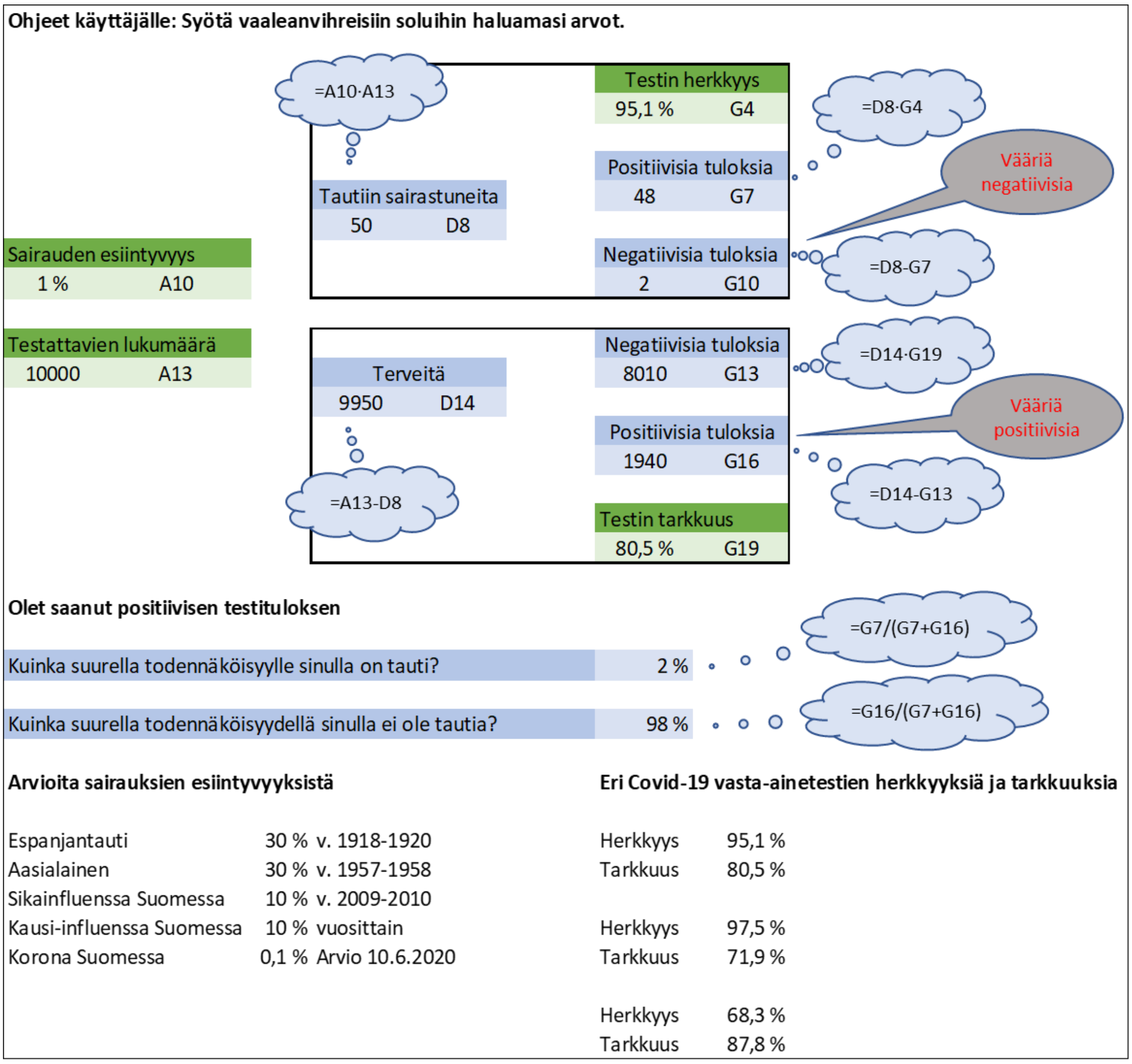


Kuva 15. Simulaatio, jossa käyttäjä valitsee ovien lukumäärän, kierrosten lukumäärän ja pelistrategian.

5.3 COVID-19 vasta-ainetesti

5.3.1 Todennäköisyys tautiin, jos testitulos on positiivinen

Verikokeen tulosten tutkimisen kautta oppilaalle tulee tutuksi ehdollisen todennäköisyyden käsite ja merkinnät. Oppilaan ohjeet tähän kokonaisuuteen ovat liitteessä 3. Tässäkin oppilaita voi eriyttää niin, että kaikki eivät tee jälkimmäistä osiota ollenkaan. Ensimmäisessä vaiheessa hyödynnetään taulukkolaskentaohjelmaa ja tehdään oletuksia testattavien lukumäärästä ja sairauden esiintyvyydestä. Tavoitteena on, että oppilas hahmottaa, kuinka usein testi antaa väärän positiivisen tuloksen. Oppilas oppii hyödyntämään taulukkolaskentaohjelmassa kaavojen kirjoittamista, jolloin hän tiettyjä soluja muuttamalla näkee syötteen muuttamisen vaikutukset lopputulokseen. Kuvassa 16 on esimerkki Excelissä tehdystä laskelmasta. Esitetyt herkkyyden ja tarkkuuden arvot ovat Suomessa käytössä olevien vasta-ainetestien valmistajien ilmoittamia [2].



Kuva 16. Esimerkki taulukosta, jonka oppilas voisi tehdä tutkiakseen, kuinka suurella todennäköisyydellä positiivinen testitulos tarkoittaa, että henkilöllä on vasta-aineita tautiin. Vihreissä soluissa on käyttäjän syötteet, joiden avulla on laskettu sinisiin soluihin arvoja. Arvojen vieressä näkyy solun nimi, jotta lukijalle on voitu selittää käytetyt kaavat ajatuskuplissa.

5.3.2 Bayesin kaavan hyödyntäminen

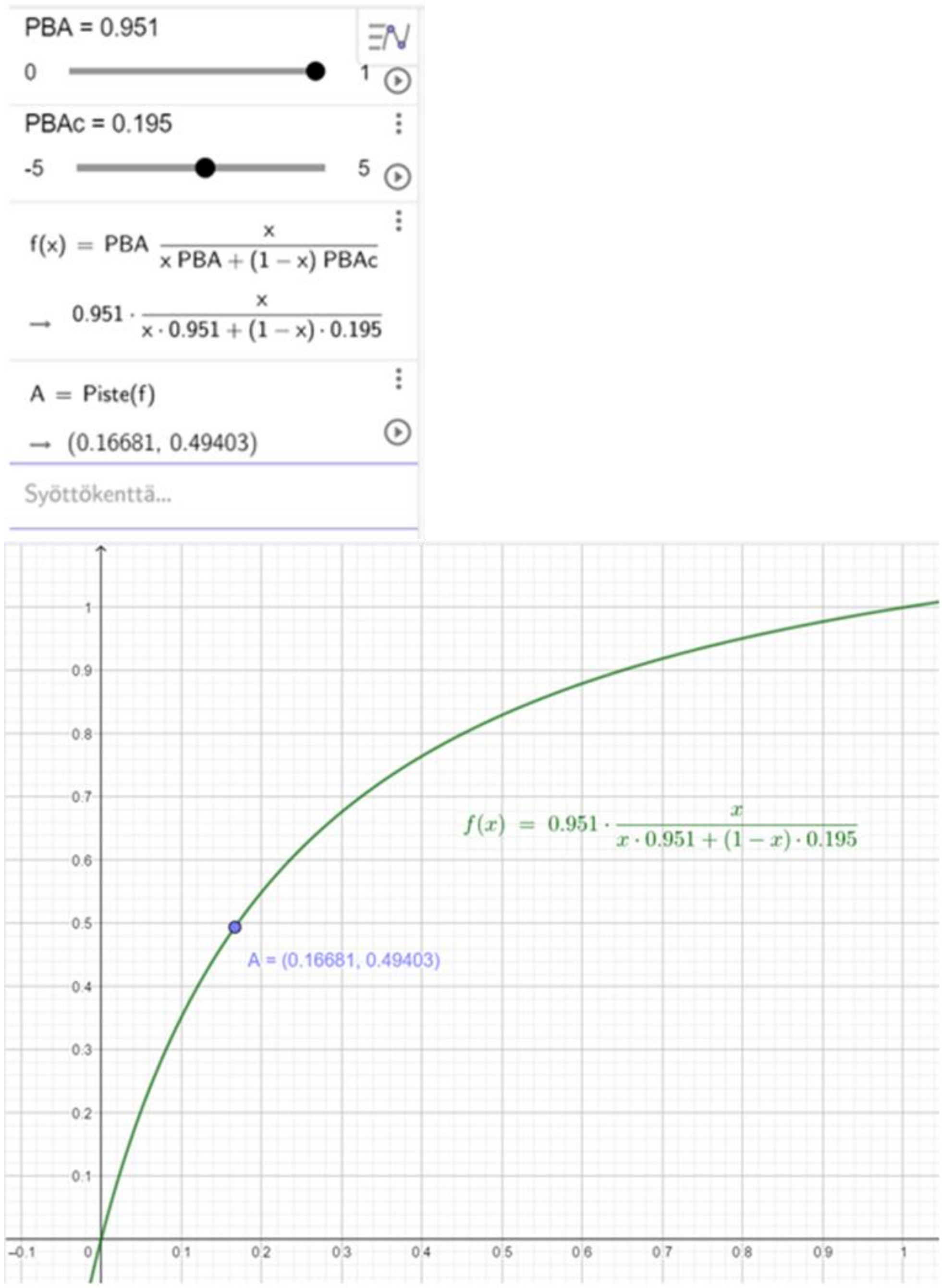
Seuraavassa vaiheessa selvitetään positiivisen testituloksen luotettavuutta, kun lähtötietoina on pelkästään testin herkkyys ja tarkkuus. Nyt ei siis tehdä oletuksia testattavien lukumäärästä. Tämä on mahdollista Bayesin kaavan avulla. Taudin esiintyvyys vaikuttaa yhä tulokseen, mutta voimme tutkia todennäköisyyttä esiintyvyyden funktiona, mikä vahvistaa samalla oppilaan taitoa tulkita kuvaajaa, ja funktio-käsitteen ymmärtämistä.

Tutkimuskysymys muotoillaan nyt toisin: kuinka suurella todennäköisyydellä henkilöllä on tauti, jos hän on saanut positiivisen testituloksen.

Tuloksen 3.5.1 mukaisesti ehdollinen todennäköisyys lasketaan $P(A|B) = \frac{P(B|A) \cdot P(A)}{P(A)P(B|A) + P(\bar{A})P(B|\bar{A})}$.

Kaavasta nähdään, että taudin esiintyvyys väestössä *P(A)* vaikuttaa tulokseen. Sen arvon voi jättää muuttujaksi, mutta on syytä tietää realistinen suuruusluokka. Arvio koronaviruksen sairastaneiden suomalaisten osuudesta 16.6.2020 oli 0,13 % [8]. Ei-sairastaneiden osuus on sairastaneiden komplementti, eli $P(\bar{A}) = 1 - P(A)$.

Valmistaja antaa eräälle Suomessa käytössä olevalle COVID-19-vasta-ainetestille herkkyyden ja tarkkuuden arvoiksi 95,1 % ja 80,5 % [2]. Tarvittavat ehdolliset todennäköisyydet ovat siis $P(B|A) = 0{,}951$, eli testin herkkyys, sekä $P(B|\bar{A}) = 0{,}195$, eli testin tarkkuuden komplementti. Näillä tiedoilla piirretään ehdollisen todennäköisyyden $P(A|B)$ kuvaaja siten, että *P(A)* on funktion muuttuja. Kuvassa 17 esitetään Geogebralla tehty kuvaaja. Kuvaajasta oppilas pystyy tekemään havaintoja ja päätelmiä, miten selkeästi viruksen levinneisyys vaikuttaa siihen, kuinka suurella todennäköisyydellä positiivinen testitulos tarkoittaa, että henkilöllä on vasta-ainetta veressään. Tässä esimerkissä Geogebran merkinnöissä PBA tarkoittaa $P(B|A)$ ja PBAc tarkoittaa $P(B|\bar{A})$.



Kuva 17. Kuvassa y-akselilla on $P(A|B)$, ja muuttujana x-akselilla $P(A)$.

Kuvaajasta nähdään, että jos todennäköisyyden $P(A|B)$ haluttaisiin olevan 0,5, ja käytössä olevan testien tarkkuus ja herkkyys pysyisivät samana, taudin levinneisyyden pitäisi olla noin 17 %. Herkkyys ja tarkkuuden komplementtiarvo ovat liukukytkimien avulla säädeltävissä. Näin oppilas pystyy testaamaan, miten testin parantaminen vaikuttaisi tuloksiin. Vasta-ainetestin tuloksen luotettavuuden matemaattinen tarkastelu osoittaa, että todellisuudessa varmistustestin tekeminen toisen valmistajan testillä on tärkeää, jotta luotettavuutta saadaan paremmaksi.

Viitteet

[1] Isaac, R. *The pleasures of Probability*. Springer, 1995.

[2] Jääskeläinen, A. J. et al. *Performance of six SARS-CoV-2 immunoassays in comparison with microneutralization.* Saatavilla https://www.medrxiv.org/content/10.1101/2020.05.18.20101618v1.full.pdf Viitattu 16.6.2020

[3] Mellin, I. *Todennäköisyyslaskenta,* Aalto-yliopisto, luentomoniste. Saatavilla http://math.aalto.fi/opetus/sovtodb/oppikirja/TodLaskLaskusaannot.pdf

[4] Opetushallitus. Perusopetuksen opetussuunnitelman perusteet 2014. https://www.oph.fi/sites/default/files/documents/perusopetuksen_opetussuunnitelman_perusteet_2014.pdf

[5] Perttula, A., Vattulainen, K. & Suurhasko, T. *Todennäköisyyslaskenta,* Tampereen teknillinen yliopisto, luentomoniste, 2012. Saatavilla http://www.math.tut.fi/courses/MAT-20501/tnlmoniste.pdf

[6] Ross, S. M. *Simulation, fifth edition*. Academic Press, 2013.

[7] Song, S., Song, J. *A note on the history of the gambler's ruin problem*. Communications for statistical applications and methods 2013, 20 (2). Saatavilla https://www.researchgate.net/publication/264021923_A_Note_on_the_History_of_the_Gambler%27s_Ruin_Problem

[8] Terveyden ja hyvinvoinnin laitos. *Tilannekatsaus koronaviruksesta,* https://thl.fi/fi/web/infektiotaudit-ja-rokotukset/ajankohtaista/ajankohtaista-koronaviruksesta-covid-19/tilannekatsaus-koronaviruksesta Viitattu 16.6.2020

Liitteet

Liite 1. Oppilaan ohjeet Uhkapelureihin

Opettajalle: Uhkapelurit soveltuu hyvin ohjelmointiharjoitukseksi. Taitavat oppilaat voivat edetä itsenäisesti, ja muiden kanssa koodia voi käydä tarvittaessa yhdessä läpi, niin että muuttujien käyttö, satunnaisluvun hyödyntäminen, ehtolauseet ja silmukkarakenne tulevat tutuiksi. Malliratkaisujen koodit löytyvät seuraavien linkkien kautta:

Parillinen vai pariton: https://scratch.mit.edu/projects/405253189
Uhkapelurit: https://scratch.mit.edu/projects/402590496

Uhkapelurit

Pelin ideana on, että rahapotti on jaettu kahdelle pelaajalle, ja yksittäisellä pelikierroksella voittaja saa hävinneeltä yhden rahan. Pelikierroksia pelataan niin kauan, kunnes toiselta pelaajalta loppuvat rahat. Kyseessä voisi olla vaikka kolikonheitto, kivi-paperi-sakset-peli, tai juoksukilpailu. Käyttäjä voi määrittää, onko peli tasapuolinen vai ei. Alkusumma ei välttämättä ole jaettu tasan pelaajien kesken, ja voittotodennäköisyys yksittäisellä kierroksella voidaan määrittää miten tahansa.

Tehtävänäsi on tehdä koodi, joka simuloi yllä kuvattua peliä.

1. Avaa https://scratch.mit.edu/ ja luo uusi projekti.
2. Tee lämmittelyharjoituksena ohjelma, joka arpoo satunnaisluvun väliltä 1-100, ja kertoo käyttäjälle, mikä satunnaisluku oli, ja onko se parillinen vai pariton. Kun tutkit luvun parillisuutta, mieti, millä säännöllä voit tehdä sen päässälaskuna.
3. Sitten tehdään varsinainen uhkapelurit-simulaatio. Tee koodi, jossa aluksi kysytään käyttäjältä, kuinka suuri koko pelin rahapotti on, ja miten se on jakautunut pelaajien kesken.
4. Lisäksi käyttäjä saa määritellä, mikä on kummankin pelaajan voittotodennäköisyys.
5. Annetusta alkutilanteesta lähtee käyntiin peli, jonka ohjelmasi simuloi. Peli loppuu, kun toisen pelaajan rahat loppuvat.
6. Lopuksi ohjelma kertoo, kumpi pelaaja voitti, ja kuinka monta kierrosta peli kesti.
7. Jos sinulle jää aikaa, tutustu kaavaan, jolla voidaan laskea voittotodennäköisyys. Tee taulukkolaskentaohjelmaan tiedosto, jossa voit vaihdella lähtöarvoja, ja näet laskennallisen voittotodennäköisyyden.

Liite 2. Oppilaan ohjeet Monty Hall -kokonaisuuteen

Opettajalle: Oppilaan ohjeet on tehty niin, että taitava oppilas selviää niiden avulla itsenäisesti, mutta jos kokemukset ohjelmoinnista ovat vähäisiä, on varmasti tarpeen käydä yhdessä luokan kanssa läpi esimerkkejä uusista rakenteista ja käytänteistä, esimerkiksi if-lauseista, laskurin hyödyntämisestä, silmukkarakenteesta, aliohjelman kutsumisesta, jne. Työstä on tarkoitus palauttaa raportti, mikä voi sopimuksenne mukaan vaikuttaa oppilaan arviointiin ja arvosanaan.

Tavoitteena on, että kaikki oppilaat pystyisivät tekemään kohdat 3 ja 4 vähintään avustettuna. Jos luokalla ei ole kokemusta ohjelmoinnista lainkaan, pelkästään näiden osioiden kautta oppii paljon tärkeitä perusteita. Varsinaisen tuloksen annetulle ongelmalle löytää tekemällä kohdat 3, 4 ja 6. Kohdan 5 mukaan ottamisen voi päättää riippuen käytettävissä olevasta ajasta. Kohta 8 on tarkoitettu lisätehtäväksi nopeimmille oppilaille. Annetut ohjeet eivät ole yksityiskohtaiset tarkoituksella, jotta oppilaalle jää tilaa löytää itse ongelmat, ja ratkaista ne. Luvusta 5.2 löytyy opettajalle vinkkejä jokaisen vaiheen ratkaisuun. Tosin on syytä pitää mielessä, että ohjelmoitaessa samaan lopputulokseen voi päätyä usealla eri tavalla.

Malliratkaisujen koodit löytyvät linkkien kautta.
Osa 1, oven arvaus: https://scratch.mit.edu/projects/403748554
Osa 2, voittoprosentti: https://scratch.mit.edu/projects/400877471
Osa 3, koko kierros: https://scratch.mit.edu/projects/401203741
Osa 4, koko pelin voittoprosentti: https://scratch.mit.edu/projects/401274458
Osa 5, n ovea: https://scratch.mit.edu/projects/402313352

Monty Hall- ongelma

Sinulla on edessäsi kolme ovea, joiden takana on satunnaisessa järjestyksessä palkintoja. Kahden oven takana on vuohi, ja kolmannen oven takana tuliterä auto. Aluksi teet ensimmäisen arvauksesi. Tämän jälkeen pelin vetäjä poistaa toisen jäljelle jääneistä ovista, sellaisen, jossa varmuudella oli vuohi. Tämän jälkeen saat valita, pysytkö alkuperäisessä arvauksessasi, vai vaihdatko valintaasi. Lopulta valitsemasi ovi avataan, ja katsotaan, onnistuitko voittamaan arvokkaan auton.

Tehtävänäsi on selvittää tietokonesimulaation avulla, onko Monty Hall-ongelmassa parempi pelistrategia pysyä alkuperäisessä valinnassa, vai vaihtaa päätöstä sen jälkeen, kun pelin vetäjä on poistanut yhden oven.

Tästä tehtävästä on tarkoitus palauttaa raportti, joka arvioidaan.

1. Kuinka suuri todennäköisyys on voittaa auto mielestäsi?
2. Avaa https://scratch.mit.edu/ ja luo uusi projekti.
3. Ensimmäinen tehtäväsi on tehdä koodi, jossa ohjelma kysyy, valitseeko käyttäjä oven 1, 2 vai 3. Ohjelma valitsee satunnaisesti, miten palkinnot sijoittuvat ovien taakse. Lopuksi se kertoo käyttäjälle, voittoko hän vuohen vai auton. Tallenna tämä projekti jollain nimellä. Liitä kuva koodista raporttiisi.

4. Avaa uusi projekti Scratchiin. Seuraavaksi tarkoituksesi on laittaa tietokone simuloimaan pelikierroksia (samanlaisia, mitä teit kohdassa 3). Kone kysyy ensin käyttäjältä, kuinka monta kierrosta simuloidaan. Sen jälkeen ohjelma ajaa simulaatiot, ja ilmoittaa lopulta, kuinka suuri oli voittoprosentti. Kirjoita raporttiin ainakin kaksi havaintoa simulointien tuloksista, ja lisäksi kuva koodista.
5. Avaa taas uusi projekti. Tässä koodissa on tarkoitus pelata kokonainen kierros Monty Hall-peliä. Aluksi käyttäjä kertoo ensimmäisen arvauksensa. Sen jälkeen kone ottaa pois yhden oven, jonka takana on vuohi, ja kysyy, haluaako pelaaja vaihtaa päätöstään, vai pysyä alkuperäisessä ovessa. Lopuksi kone kertoo, mitä käyttäjän valitseman oven takaa paljastui.

   Tästä koodista tulee melko pitkä. Mieti, voisiko sitä mahdollisesti selkiyttää aliohjelmilla. Mitkä olisivat sopivia kokonaisuuksia aliohjelmaan?

   Liitä koodi taas raporttiin. Kerro lisäksi, mikä oli tämän kokonaisuuden vaikein asia, jonka sait ratkaistua.
6. Avaa seuraava projekti. Nyt on tarkoitus simuloida Monty Hall-peliä useita kierroksia, ja laskea voittoprosentti. Päätetään aluksi, että pelaajan ensimmäinen valinta on aina ovi numero 1. Käyttäjä kertoo, kuinka monta kierrosta simulaatiota ajetaan, ja aikooko hän pysyä päätöksessään, vai vaihtaa arvaustaan. Tämä pelistrategia pysyy siis samana koko simulaation ajan. Vasta uudessa simulaatiossa käyttäjä voi valita toisen strategian.

   Liitä koodi raporttiin. Pohdi, mitä merkitystä tulokseen on sillä, että yksinkertaistimme peliä, ja käyttäjä valitsee aina aluksi oven 1. Kirjoita ainakin kolme havaintoa simulointien tuloksista. Onko tulos sen kaltainen, mitä mietit kohdassa 1?
7. Pystytkö laskemaan voittotodennäköisyyden matemaattisen tarkastelun avulla? Klassisessa todennäköisyydessä vaaditaan, että jokaisen tulosvaihtoehdon todennäköisyyden pitää olla sama (symmetriaehto). Täyttyykö tämä vaatimus?
8. Avaa uusi projekti. Tehdään samanlainen tarkastelu kuin kohdassa 6, mutta nyt käyttäjä ilmoittaa, kuinka monta ovea pelissä on aluksi. Useamman oven tapauksessa peli toimii niin, että pelin vetäjä poistaa aina yhden oven, ja tässä vaiheessa pelaaja saa päättää, pysyykö hän päätöksessään, vai vaihtaako valintaansa. Näin jatketaan, kunnes jäljellä on kaksi ovea, joista pelaajan viimeinen valinta avataan. Voit tehdä tämän tarkastelun matemaattisesti tai simuloimalla, tai molemmilla tavoilla. Voit tehdä peliin yksinkertaistuksia, mutta mainitse raportissasi, mitä ne ovat. Liitä koodi taas raporttiisi.
9. Vastaa lopuksi seuraaviin kysymyksiin:

   Mitä uutta opit ohjelmoinnista?

Mikä oli suurin haaste, joka tuli vastaan? Millä keinoilla sait ratkaistua sen?
Arvioi omaa työskentelyäsi koko projektin aikana.

Liite 3. Oppilaan ohjeet vasta-ainetesti-kokonaisuuteen

Opettajalle: Tässä työssä on selvä yhteys biologiaan. Kannattaa kysellä, olisiko biologian opettajan kanssa mahdollisuus toteuttaa oppiainerajat ylittävä projekti, jolloin biologiassa käsiteltäisiin viruksia ja vasta-aineen muodostumista vereen. Kokonaisuus on hieman haastava peruskoululaiselle. Parhaiten tämä sopisi valinnaisaineryhmään, matematiikka-painotteiselle luokalle, tai motivoituneelle ryhmälle. Annetut ohjeet oppilaille eivät ole kovin yksityiskohtaiset, jotta oppilaan omalle miettimiselle jää riittävästi tilaa. Tehtävä on kuitenkin teorialtaan sen verran monimutkainen, että lisävinkkien antaminen tai yhdessä eteneminen lienee tarpeellista suurelle osalle.

Luvuista 3.3 ja 5.3 löytyy opettajalle vinkkejä oikeisiin vastauksiin. Pienenä huomautuksena, että kohdassa kaksi on hyvä tarkistaa, että oppilas on ymmärtänyt analysoida, tuleeko positiivisia tuloksia enemmän sairastaneiden vai terveiden joukosta, ja osaa selittää tämän. Työstä on tarkoitus palauttaa raportti, mikä voi sopimuksenne mukaan vaikuttaa oppilaan arviointiin ja arvosanaan.

Vasta-ainetestien tulosten luotettavuus

Virusepidemioiden aikana tai niiden jälkeen voi olla hyödyllistä selvittää, kuinka suurella osalla väestöstä löytyy verestä vasta-ainetta virukselle. Tässä työssä tarkastelemme COVID-19-viruksen aiheuttaman korona-pandemian aikana käytössä olleita vasta-ainetestejä.

Tutkittavasta väestöstä otetaan ensin satunnaisotos, johon kuuluville ihmisille tehdään vasta-ainetesti verikokeella. Vasta-ainetesteissä positiivinen tulos tarkoittaa, että tutkittavalla on vasta-ainetta veressään, eli hän on sairastanut taudin. Negatiivinen tulos tarkoittaa, että vasta-ainetta ei ole, ja hän ei ole sairastanut tautia. Testeissä tulee aina virheellisiä tuloksia.

Väärä negatiivinen tulos tarkoittaa, että testi näyttää negatiivista, vaikka henkilöllä on vasta-ainetta. Sitä osuutta, jossa testi näyttää oikein sairastaneen ihmisen tuloksen, kutsutaan testin herkkyydeksi. Herkkyyden arvo on eräässä testissä 95 %. Tällöin siis 5 % on vääriä negatiivisia.

Väärä positiivinen tulos tarkoittaa, että testi näyttää positiivista, vaikka henkilöllä ei ole vasta-ainetta. Sitä arvoa, jolla testi antaa oikean tuloksen henkilölle, jolla ei ole vasta-ainetta, kutsutaan testin tarkkuudeksi. Jos tarkkuuden arvo on 81 % se tarkoittaa, että 19 % saa väärän positiivisen.

Tee työstä raportti, jossa vastaat alla olevissa ohjeissa esitettyihin kysymyksiin.

1. Piirrä ensin puukartta, jossa näkyy, miten otokseen kuuluvat ihmiset kuuluvat terveisiin ja taudin sairastaneisiin. Näistä kahdesta joukosta osalle tulee positiivinen testitulos, ja osalle negatiivinen yllä annettujen herkkyys- ja tarkkuusarvojen mukaisesti.
   Jos tiedät otoksen koon, testin herkkyys- ja tarkkuusarvot, sekä positiivisten ja negatiivisten tulosten lukumäärän kummastakin joukosta, pystytkö näillä tiedoilla päättelemään, kuinka suuri osuus väestöstä on sairastanut taudin?
2. Muotoillaan seuraavaksi kysymys uudella tavalla. Olet saanut positiivisen testituloksen. Kuinka suurella todennäköisyydellä olet sairastanut taudin, ja kuinka suurella todennäköisyydellä et? Lähde selvittämään vastausta taulukkolaskentaohjelman avulla. Voit itse tehdä tarvittavat oletukset. Kokeile, miten lopputulos vaihtuu, kun muutat alkuoletuksiasi. Kirjoita havainnoistasi raporttiin, ja liitä kuva taulukkolaskentaohjelmasta. Kerro selvästi, mitä alkuoletuksia tarvitsit, jotta pystyit tekemään laskelmat.
3. Seuraavaksi pyrimme selvittämään, kuinka suurella todennäköisyydellä henkilöllä on tauti, jos hän on saanut positiivisen testituloksen, mutta erona edelliseen kohtaan teemme vähemmän alkuoletuksia.
   Seuraavassa kaavassa käytetyt merkinnät käydään tunnilla yhdessä läpi, ja samoin se, miten kaava johdetaan.
   Tapahtuma A tarkoittaa, että henkilöllä on vasta-ainetta ja tapahtuma B, että testitulos on positiivinen.
   $$\mathrm{P}(\mathrm{A}|\mathrm{B}) = \frac{\mathrm{P}(\mathrm{A} \cap \mathrm{B})}{\mathrm{P}(\mathrm{B})} = \frac{\mathrm{P}(\mathrm{B}|\mathrm{A})\mathrm{P}(\mathrm{A})}{\mathrm{P}(\mathrm{B})} = \frac{\mathrm{P}(\mathrm{B}|\mathrm{A})\mathrm{P}(\mathrm{A})}{\mathrm{P}(\mathrm{A})\mathrm{P}(\mathrm{B}|\mathrm{A}) + \mathrm{P}(\bar{\mathrm{A}})\mathrm{P}(\mathrm{B}|\bar{\mathrm{A}})}$$
   Tehtävänäsi on piirtää kuvaaja, jossa y-akselilla on $P(A|B)$ ja x-akselilla on $P(A)$. Käytä Geogebraa. Muista miettiä, mitkä ovat järkevät tarkasteluvälit akseleille. Liitä kuvaaja raporttiin, ja selitä lukijalle, mitä kuvasta pystyy näkemään.
4. Vastaa lopuksi seuraaviin kysymyksiin:
   Mitä uutta opit työn aikana?
   Opitko uusia työskentelytapoja? Jos opit, niin mitä, ja miten voit hyödyntää niitä?
   Mikä asia oli vaikein ymmärtää? Mikä auttoi sen selvittämisessä?