PE Voetbal

Data advanced

Tommie Bergström 2TINm

Casper Naudts 2TINm

Brecht Van Camp 2TINm

Inhoud

[Opgave 1 a: 3](#_Toc6567106)

[1. Beschrijf hoe je deze dataset ingevoerd hebt in Python 3](#_Toc6567107)

[2. Genereer de geboortedatum 3](#_Toc6567108)

[3. Genereer de kolom inzet met 3 mogelijke categorieën 3](#_Toc6567109)

[4. Maak in python een spreiding wolk 3](#_Toc6567110)

[5. Maak in python een staafdiagram van het aantal gemaakte goals per positie 4](#_Toc6567111)

[6. Bereken het gemiddelde en de modus van kolom D 4](#_Toc6567112)

[7. Bereken mbv Python kwartiel 1 en de standaardafwijking van kolom G 4](#_Toc6567113)

[8. Is er een verband te zijn tussen positie op het veld en aantal gemaakte goals? 4](#_Toc6567114)

[9. Maak een cirkeldiagram op basis van inzet 5](#_Toc6567115)

[10. Vergelijk de posities linkervleugel - rechtervleugel - piloot wat betreft het aantal gemaakte goalen mbv een boxplot en bespreek 5](#_Toc6567116)

[11. Welk soort gegevens zijn 'aantal gemaakte goalen', 'inzet' en 'gewicht' 5](#_Toc6567117)

[Pluralsight course 6](#_Toc6567118)

[Slechte data representatie 7](#_Toc6567119)

[1. Artikel 1 7](#_Toc6567120)

[2. Artikel 2 7](#_Toc6567121)

[3. Conclusie 7](#_Toc6567122)

# Opgave 1 a

## Invoeren van de dataset in Python

Het Excel bestand wordt eerst omgevormd naar een DataFrame door de read\_excel functie in de python-library “pandas” te gebruiken. Dit DataFrame wordt dan omgevormd naar een array van person objecten. In een person object worden de velden: naam, voornaam, positie, aantal gemaakte goals, geboortedatum, inzet, gezicht en lengte bijgehouden.

## Genereren van de geboortedatum

Omdat de geboortedatum niet in het bronbestand staat, moet hij gegenereerd worden. Alle spelers zijn geboren in 2011. Voor het genereren van de geboortedatums wordt een klasse dataGenerator geschreven met enkele statische functies, zoals genereer geboortedatum. Bij de startdatum (01/01/2011) wordt een willekeurig getal tussen 0 en 1 vermenigvuldigd met de einddatum (31/12/2011) min de begindatum opgeteld.

01/01/2011 + willekeurig getal \* (31/12/2011 - 01/01/2011)

Dit wordt herhaald voor elke persoon in de array van person objecten. Deze geboortedatum wordt verder ingedeeld in vier categorieën.

|  |  |
| --- | --- |
| Categorie 1 | Geboren in januari - februari - maart |
| Categorie 2 | Geboren in april - mei - juni |
| Categorie 3 | Geboren in juli - augustus - september |
| Categorie 4 | Geboren in oktober - november - december |

## Genereren van de kolom inzet

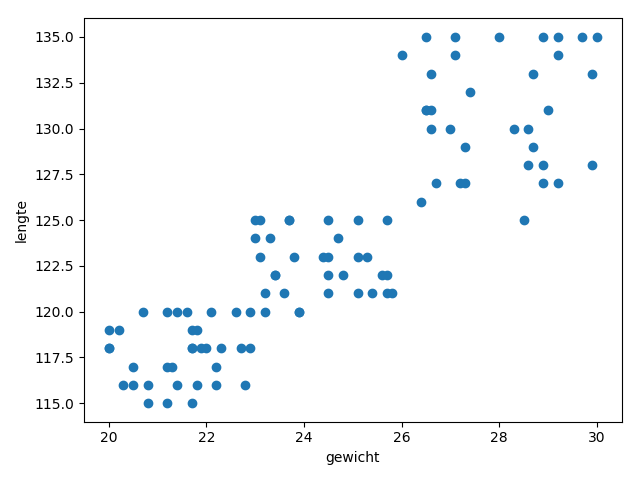
De drie categorieën van inzet worden op de volgende manier bepaald op basis van de geboortedatum.

|  |  |  |
| --- | --- | --- |
| Zeer goed | Goed | Matig |
| Geboren in januari - februari - maart | Geboren in april - mei - juni of Geboren in juli - augustus - september | Geboren in oktober - november - december |

In de klasse dataGenerator is er een statische methode gemaakt om inzet te genereren genaamd generateInzet. Hierin wordt de inzet van de speler berekend ten opzichte van de geboortedatum. Zo wordt er gekeken in welke categorie een speler thuishoort en deze waarde wordt daarna opgeslagen bij de speler.

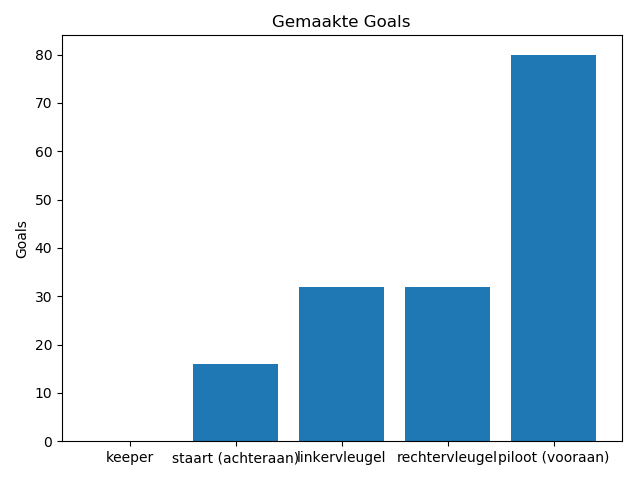
## Weergeven in een spreidingswolk van lengte en gewicht van spelers

In de klasse plot zit een statische functie plotPerson waar een array van persoonsObjecten aan meegeven wordt. Deze persoonsObjecten worden in puntje 1 besproken. Bij het plotten van de puntenwolk is er een 2d-Array met een array voor X en een voor Y-coördinaten. Op deze array wordt op de x-coördinaat gewicht van de spelers toegevoegd en op de y-coördinaat de lengte. Deze 2d-Array wordt dan meegegeven aan de functie Scatter in de Phyton-library “matplotlib.pyplot”. Aan deze library geven we nog twee statische variabelen mee: X-label en Y-Label. X-label krijgt de naam van de x-as en y-label die van de y-as. Hierna wordt de functie show opgeroepen en zal de puntenwolk weergegeven worden.



## Weergeven in een staafdiagram van aantal gemaakte goals per positie

In de klasse plot bestaat er de statische methode “plot\_graph\_five” waarmee een staafdiagram wordt gegenereerd. Om een staafdiagram te plotten moet de functie bar uit de Phyton-library “matplotlib.pyplot” gebruikt worden. Deze functie vraagt als argument een positie en de weer te geven waarde. De positie is een numerieke waarde beginnend bij 0 en elke lijn heeft zijn bijhorende index. Om de lijnen een naam te geven wordt de functie “xticks” gebruikt met twee argumenten: een array met posities en een bijhorende array met namen voor die posities. Om de Y-as een naam te geven wordt de functie “Y-label” gebruikt, in dit geval met argument goals. Om de grafiek een titel te geven gebruiken we de functie “titel” met als argument de string “gemaakte goals”. Wanneer de functie “show” opgeroepen wordt verschijnt de grafiek.



## Berekenen van gemiddelde en modus van gemaakte goals per positie

Om het gemiddeld aantal goals per positie te berekenen wordt een statische functie “calculate\_avarage\_goals\_per\_position” met als argument een array van personenobjecten in de klasse CalculateData gemaakt. Deze gebruikt de Phyton-lybrary “statistics”. Om het gemiddelde te berekenen wordt er eerst een object gemaakt met vijf variabelen die geïnitialiseerd worden met een array. Deze variabelen stellen de posities voor. Hierna wordt er door de array van persoonsobjecten geïtereerd om zo het aantal gemaakte goals in de juiste categorie toe te voegen. Als alle gemaakte goals bij de juiste categorie zijn toegevoegd wordt er geïtereerd door de posities. Hiervan wordt dan het gemiddelde berekend. De modus wordt op een gelijkaardige manier berekend.

|  |  |  |
| --- | --- | --- |
| Positie | Gemiddelde | Modus |
| Keeper | 0 | 0 |
| Staart (achteraan) | 0.8 | 0 |
| Linkervleugel | 1.6 | 1 |
| Rechtervleugel | 1.6 | 1 |
| Piloot (vooraan) | 4 | 4 |

## Berekenen van kwartiel 1 en standaardafwijking van gewicht

De gemaakte functie “calculate\_quartil\_of\_weight” heeft als argumenten een array van persoonobjecten en de quartil-value. In deze functie wordt de Phyton-lybrary “numpy” gebruikt. Hierna wordt de array weights geïnitialiseerd. Daarna wordt er geïtereerd over de array van persoonsobjecten en wordt de property gewicht toegevoegd aan de array weights. Met de numpy-functie “quantile” wordt het kwartiel berekend. Deze functie heeft als parmeters de array weights en de quartile\_value. Om de standaardafwijking te berekening wordt ongeveer dezelfde code als kwartielberekenen gebruikt. Het enige verschil is dat er voor deze geen kwartielwaarde nodig is en dat op het einde de functie “std” met als argumenten de array van weights gebruikt wordt in plaats van quantile.

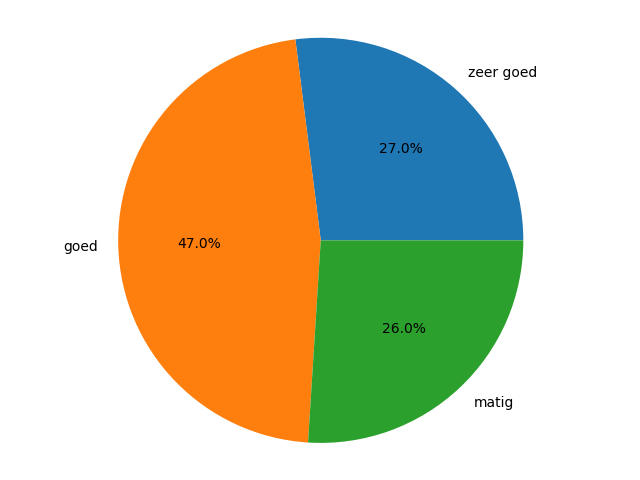
|  |  |  |
| --- | --- | --- |
|  | Kwartiel 1 | Standaardafwijking |
| Gewicht | 22.075 | 2.869 |

## Verband tussen positie op het veld en aantal gemaakte goals

In het staafdiagram van punt 5 is duidelijk te zien dat spelers met positie piloot veruit meer goals maken dan spelers op andere posities. Dit wordt verklaard door het feit dat zij dichter bij de goal van de tegenstander staan. Spelers op de linker en rechtervleugel maken evenveel goals maar minder dan piloot spelers omwille van hun centrale positie. Staartspelers maken er nog minder omdat zij helemaal achteraan spelen op het veld. De keepers maken geen goals, dit is te wijten aan het feit dat ze normaal niet uit hun eigen doelgebied komen.

## Inzet weergeven in een cirkeldiagram

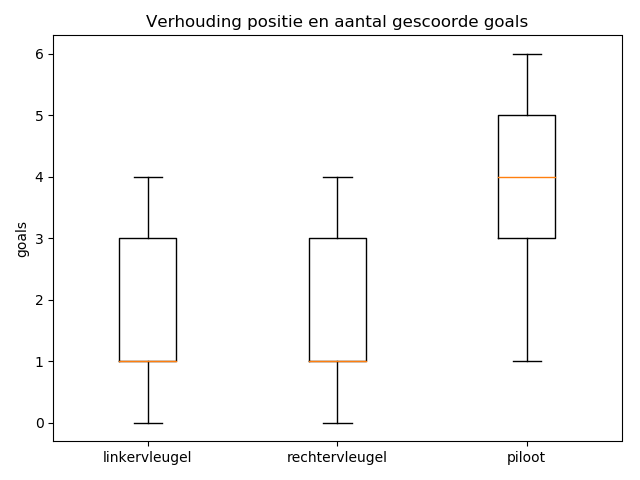
Voor het plotten van een cirkeldiagram hebben we de functie “plot\_pie\_chart\_inzet” gemaakt, met als argument een array van persoonsobjecten in de klasse plot. Deze functie maakt gebruik van de Python-library “matplotlib.phyplot”. Er wordt een array van labels gemaakt met als waarde goed, zeer goed en matig. Een array wordt geïnitialiseerd met drie posities waarvan elke positie gelijk is aan 0. Hierna wordt er geïtereerd door de array van persoonsobjecten en wordt de relevante waarde van inzet met één verhoogd. De functie “pie” uit de library “matplotlib.phyplot” wordt gebruikt, met als argumenten de array van values, labels en autopct=’%1.1f%%’. Dit laatste argument zorgt ervoor dat de effectieve waarde wordt weergegeven in de pie. Hierna wordt de functie “ax” met als argument equal opgeroepen. Door de show functie op te roepen wordt de pie-chart weergegeven



## Vergelijken van goals per positie met een boxplot

Om een boxplot te plotten wordt er een functie gemaakt genaamd “plot\_boxplot\_goals\_per\_position” met als argument een array van persoonsobjecten in de klasse plot. In deze functie wordt er gebruik gemaakt van de phyton-library “matplotlib.phyplot”. Hierin wordt een array van posities geïnitialiseerd met als waarde linker-vleugel, rechter-vleugel en piloot. Er wordt een object geïnitialiseerd in de variabele groups met als variabele linker-vleugel, rechter-vleugel en piloot, telkens geïnitialiseerd als een lege array. Hierna wordt er geïtereerd over de array van persoonsobjecten. Er wordt eerst gekeken of de persoon op 1 van de te vergelijken posities speelt. Als dit het geval is wordt het persoonsobject toegevoegd aan de array van de bijhorende positie in de variabele groups. De naam van de grafiek wordt ingesteld als “verhouding, positie en gescoorde goals”. Dan wordt de array “super\_data” geïnitialiseerd. Er wordt geïtereerd over de groupsarray om zo een nieuwe array genaamd “data” te initialiseren en er wordt over de bijhorende spelers van die groep geïtereerd. Het aantal gemaakte goals van elke speler wordt aan de data array toegevoegd en deze wordt hierop toegevoegd aan de super\_data array. De functie boxplot wordt opgeroepen uit de Python library “matplotlib.phyplot” met als argumenten de array superdata en de array positions als labels. De functie “show” wordt opgeroepen om de boxplot te tonen.

Wat we uit deze boxplot kunnen afleiden is dat de linker en rechtervleugel evengoed spelen en dat de pilootpositie duidelijk meer goals maakt wat zeer logisch is omdat deze meer vooraan op het veld staan.



## Welk soort gegevens zijn 'aantal gemaakte goals', 'inzet' en 'gewicht'?

* Aantal gemaakte goals: kwantitatief en discreet omwille van hun gehele waarde.
* Gewicht: kwantitatief en continu omwille van de decimale waarde van de getallen.
* Inzet: kwalitatieve gegevens en ordinaal omwille van de duidelijke rangschikking in de gegevens.

# Opgave 1 b:

# Pluralsight course

We hebben de video course “Doing Data Science with Python” (<https://app.pluralsight.com/library/courses/python-data-science/table-of-contents>) gebruikt om te leren hoe een aantal grafieken uitgetekend moeten worden. Deze video course werd niet super veel gebruikt omdat een groepslid al heel wat ervaring met Python heeft en dus al veel kennis had van benodigde libraries. Het interessantste hoofdstuk in deze video course was voornamelijk “Exploring and Processing Data - Part 1” omdat het hier echt specifiek ging over het weergeven van data. Iets wat toch redelijk essentieel is voor deze opdracht. Vooral de video over de boxplot hielp omdat het niet helemaal duidelijk was welke argumenten de functie nodig had. Maar de video bracht veel duidelijkheid op dat vlak. Wij raden dus niet aan om deze hele course te bekijken, maar enkel het besproken hoofdstuk omdat je anders overladen zal worden met onnodige info.

# Opgave 2

# Slechte data representatie

## Artikel 1: migratiepact mag regering doen vallen

Het eerste artikel is gebaseerd op een online bevraging bij ongeveer 2500 Belgen, waarvan zo’n 1000 Vlamingen. Hoewel de steekproef voldoende groot is om representatief te kunnen zijn, kunnen er vragen gesteld worden bij de gevolgde methode. Bij een online bevraging bereikt men immers niet alle groepen van de bevolking. Vooral ouderen en misschien ook lager opgeleiden zullen ondervertegenwoordigd zijn in de steekproef. Ook lijkt het aandeel Vlamingen in de steekproef te laag vergeleken met het aandeel in de bevolking. Vlaanderen vertegenwoordigd maar ongeveer 40% van de steekproef, terwijl 57% van de Belgen in Vlaanderen woont.

Het is dus waarschijnlijk dat er een relatief hoge foutenmarge zit op de gepresenteerde resultaten, waardoor voorzichtigheid nodig is bij hun interpretatie. Zo denkt, volgens de bevraging, 33% van de Vlamingen dat de regering mocht vallen over het migratiepact. In Wallonië en Brussel waren dat respectievelijk 34% en 35%. Deze verschillen lijken te klein tegenover de foutenmarge om te kunnen stellen dat in Brussel en Franstalig België meer mensen vinden dat de regering mocht vallen over het migratiepact.

Wanneer het artikel ook nog meer gedetailleerde cijfers heeft per politieke voorkeur, worden de resultaten nog onbetrouwbaarder.

<https://www.nieuwsblad.be/cnt/dmf20181205_04011918>

## Artikel 2: charcuterie is zo kankerverwekkend als asbest, diesel en tabak

In deze open brief legt Sam Proesmans uit waarom hij geen dokter meer wil zijn. Hij vindt dat de geneeskunde te veel de symptomen behandelt in plaats van het onderliggend gedrag te wijzigen. Hij geeft als voorbeeld dat darmkanker veroorzaakt wordt door ongezonde voeding. Om zijn argument kracht bij te zetten stelt hij: “Geen misplaatste grap, charcuterie staat in dezelfde categorie als asbest, diesel en tabak: bewezen kankerverwekkend.”

Daarmee geeft hij de indruk dat charcuterie even kankerverwekkend is als asbest, diesel en tabak. Dit is echter niet correct, zoals blijkt uit een gedetailleerde lezing van het artikel waar hij naar verwijst.[[1]](#footnote-2) Dit artikel stelt dat de Wereldgezondheidsorganisatie inderdaad oordeelt dat het bewezen is dat charcuterie kanker kan veroorzaken, net als onder andere asbest, diesel en tabak. Maar, wat Sam Proesmans niet vermeldt is dat de kans dat je kanker krijgt van charcuterie veel en veel lager ligt dan van deze andere kankerverwekkende producten. Terwijl het dagelijks eten van 50 gram charcuterie de kans op kanker met 18% verhoogt, verhoogt roken de kans op kanker met een factor 10 tot 20.[[2]](#footnote-3) Hetzelfde fact-check artikel in de Standaard stelt ook dat het eten van charcuterie naar schatting jaarlijks 34 000 kankerdoden veroorzaakt, vergeleken met de ongeveer 1 miljoen kankerdoden die te wijten zijn aan tabak. De risico’s van het eten van charcuterie kunnen dus niet op dezelfde lijn gezet worden als die van zwaar kankerverwekkende stoffen als asbest, diesel en tabak.

<https://www.knack.be/nieuws/belgie/na-13-jaar-ben-ik-van-plan-om-de-geneeskunde-vaarwel-te-zeggen-ik-wil-geen-dokter-meer-zijn/article-opinion-1451199.html>

## Conclusie

Uit deze twee artikelen kunnen we besluiten dat we bij de interpretatie van data in de pers steeds voorzichtig moeten zijn en de onderliggende gegevens moeten bekijken. Zo kunnen er problemen zijn met de manier waarop de data verzameld worden (artikel 1), met hun interpretatie (artikel 1), of met de manier waarop ze voorgesteld worden (artikel 2).

1. <https://www.theguardian.com/society/2015/oct/26/bacon-ham-sausages-processed-meats-cancer-risk-smoking-says-who> [↑](#footnote-ref-2)
2. <http://www.standaard.be/cnt/dmf20190414_04325159> [↑](#footnote-ref-3)