Casus Scripting Blok 1

**Teamleden:** Glenn van der Heijden (2304319), Keanu Molin (1235583), Kyle Lukassen (1641344), Robbin van Kempen (2406543)

**Datum:** 5-11-2024

# Inhoudsopgave

1. Casus / Opdrachtbeschrijving

2. Product Backlog

3. Sprint Backlog

4. Wireframes

5. Python visualisaties

6. Excel visualisaties

7. Code en hulpmiddelen

8. Logboek Scrum Master

9. Testplan en testrealisatie

# 1. Casus / Opdrachtbeschrijving

Vanuit de klant is de wens gekomen om een dashboard te ontwikkelen waarin managementinformatie kan worden ingezien door middel van visualisaties. Waarbij het specifiek om data over de verkopen gaat. Waarmee meer inzicht kan worden verkregen in de verkopen zodat de winstgevendheid van het bedrijf kan worden vergroot. Daarbij heeft de klant specifiek gevraagd naar onderstaande visualisaties:  
- 1D Visualisatie: percentage verkochte pizza's per categorie.  
- 2D Visualisatie (2015): meest verkochte pizzanaam en maat.  
- 2D Visualisatie per categorie: aantal verkochte pizzanamen per categorie.  
- 3D Visualisatie: maandelijkse verkopen per pizzanaam en maat.

# 2. Product Backlog

De product backlog bevat een lijst van alle user stories die nodig zijn om het dashboard voor de pizza webshop te realiseren. Elke user story beschrijft een specifieke functionaliteit en bevat functionele en niet-functionele requirements.

## User Story 1: Als gebruiker van het dashboard wil ik het percentage verkochte pizza's per categorie kunnen zien, zodat ik inzicht heb in de populariteit van elke categorie.

### Functionele requirements:

* Het dashboard moet een 1D visualisatie tonen van de procentuele verdeling van verkochte pizza's per categorie.

### Niet-functionele requirements:

* De visualisatie moet binnen 5 seconden laden.
* De gegevens moeten automatisch worden geüpdatet.

## User Story 2: Als gebruiker van het dashboard wil ik een overzicht zien van de meest verkochte pizzanamen en maten in 2015, zodat ik weet welke combinaties het best verkopen.

### Functionele requirements:

* Het dashboard moet een 2D visualisatie bevatten van pizzanamen en maten, specifiek voor het jaar 2015.

### Niet-functionele requirements:

* De visualisatie moet binnen 5 seconden laden.
* Indien database niet aangemaakt is moet de visualisatie binnen 15 seconden geladen worden.

## User Story 3: Als gebruiker van het dashboard wil ik per categorie de pizzanamen en verkochte aantallen zien, zodat ik kan beoordelen welke soorten het meest populair zijn binnen elke categorie.

### Functionele requirements:

* Het dashboard moet een 2D visualisatie tonen met pizzanamen per categorie en hun verkochte aantallen.

### Niet-functionele requirements:

* De visualisatie moet binnen 5 seconden laden.
* De gegevens moeten automatisch worden geüpdatet.

## User Story 4: Als gebruiker van het dashboard wil ik het aantal verkochte pizza's per maand, per pizzanaam, en per maat kunnen zien om verkooptrends te kunnen analyseren.

### Functionele requirements:

* Het dashboard moet een 3D visualisatie bieden van maandelijkse verkoopgegevens per pizzanaam en maat.

### Niet-functionele requirements:

* De data moet betrouwbaar zijn en een terugkijkperiode van 1 jaar bieden.
* De 3D visualisatie moet soepel werken op alle computers.

## User Story 5: Als gebruiker van het dashboard wil ik de outliers van de visualisaties weg kunnen filteren zodat ik beter inzicht krijg in de belangrijkste data.

### Functionele requirements:

* Het dashboard moet visualisaties zonder extremen (datapunten buiten een opgegeven domein) kunnen weergeven

### Niet-functionele requirements:

* De data moet betrouwbaar zijn, ondanks het wegnemen van de outliers.
* De visualisatie moet binnen 5 seconden laden.

# 3. Sprint Backlog

## Sprint 1: Basis setup en eerste user story implementatie.

* Dataset PizzaSales.xlsx importeren en analyseren.
* Implementeren van de 1D visualisatie (percentage verkochte pizza’s per categorie).
* Basisstructuur van het dashboard opzetten.
* Begin maken met wireframes voor dashboard.

## Sprint 2: Uitbreiden van visualisaties en verfijnen van design.

* Implementeren van de 2D visualisatie van de meest verkochte pizzanamen en maten in 2015.
* Verbeteringen aan wireframes op basis van feedback.
* Testing en validering van data-output.

## Sprint 3: Voltooien van resterende visualisaties.

* Implementeren van de 2D visualisatie van verkochte aantallen per categorie.
* Verbeteringen doorvoeren op basis van klantfeedback.
* Naar aanleiding van feedback: outliers wegfilteren.
* Opzetten van testplan voor alle visualisaties.

## Sprint 4: Finaliseren en testen.

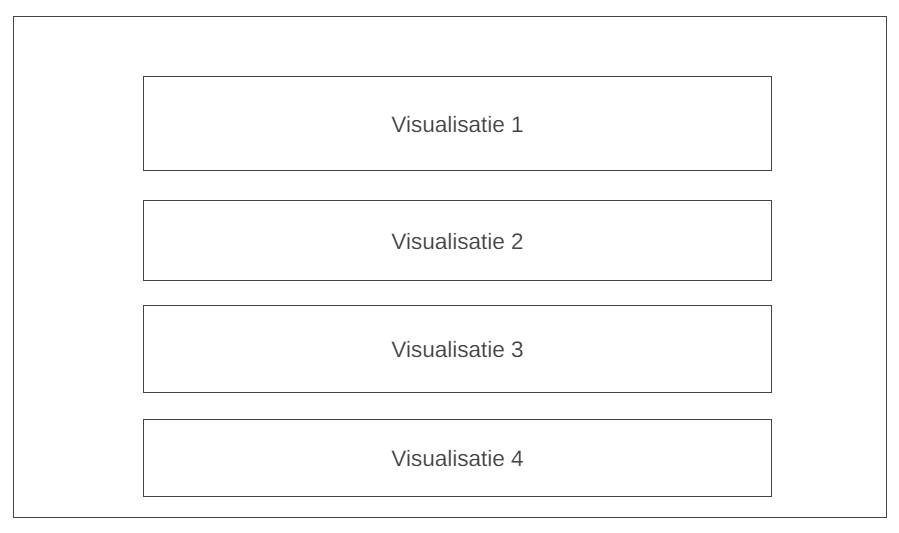
* Implementeren van de 3D visualisatie (maandelijkse verkopen per pizzanaam en maat).
* Uitvoeren van tests volgens testplan.
* Documentatie afwerken, inclusief het Scrum Master logboek en het testplan.

# 4. Wireframes

De wireframes dienen als eerste ontwerp van de lay-out en structuur van het dashboard. Ze geven een visueel overzicht van de verschillende secties, waar gebruikers de visualisaties kunnen bekijken. De wireframes zijn ontworpen met de belangrijkste klantvereisten en de doelstellingen van het dashboard in gedachten.

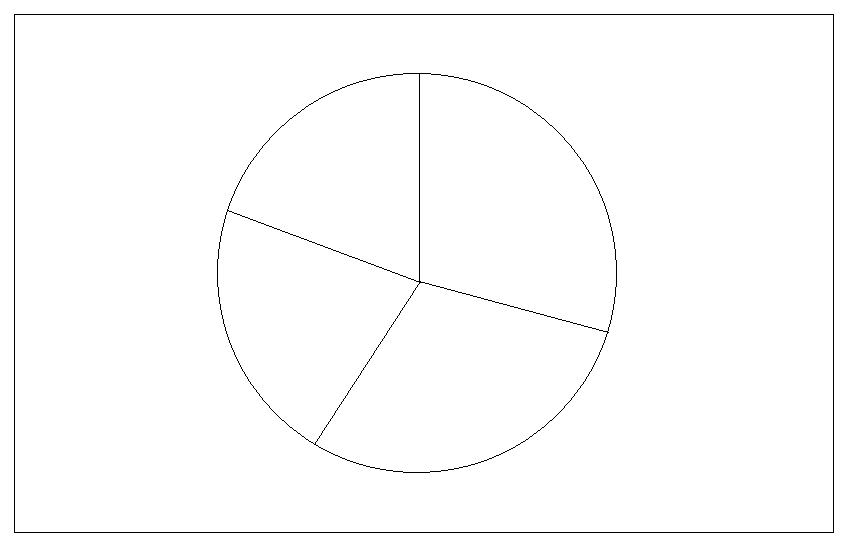
#### Wireframe 1: Hoofdpagina van het dashboard

* **Doel**: dit is de startpagina van het dashboard, die toegang biedt tot alle visualisaties.
* **Elementen**:
  + **Navigatiebalk**: bevat links naar de verschillende visualisaties.



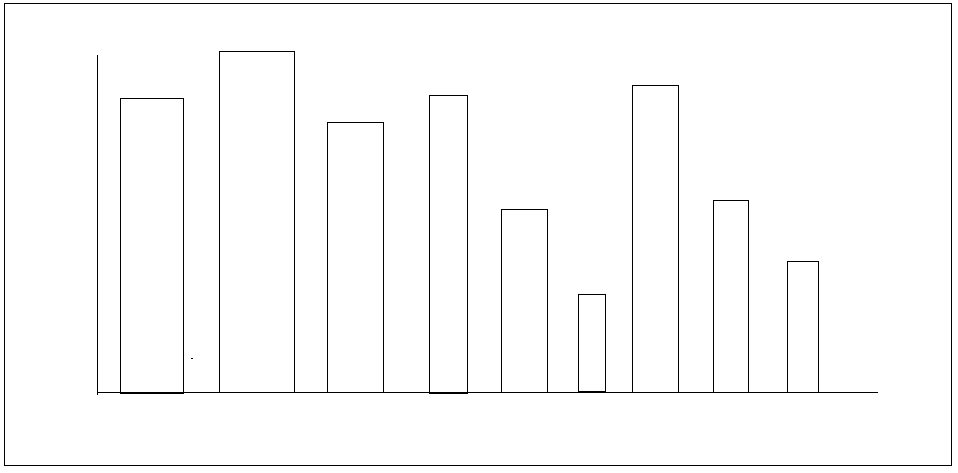
#### Wireframe 2: 1D Visualisatie – Percentage verkochte pizza’s per categorie

* **Doel**: laat een eenvoudige, overzichtelijke 1D visualisatie zien van het percentage verkochte pizza's per categorie.
* **Elementen**:
  + **Grafiek**: een cirkeldiagram dat het relatieve aandeel van elke categorie toont.
  + **Weergave**: pizzacategorieën weergave in procenten met kleurcodering.



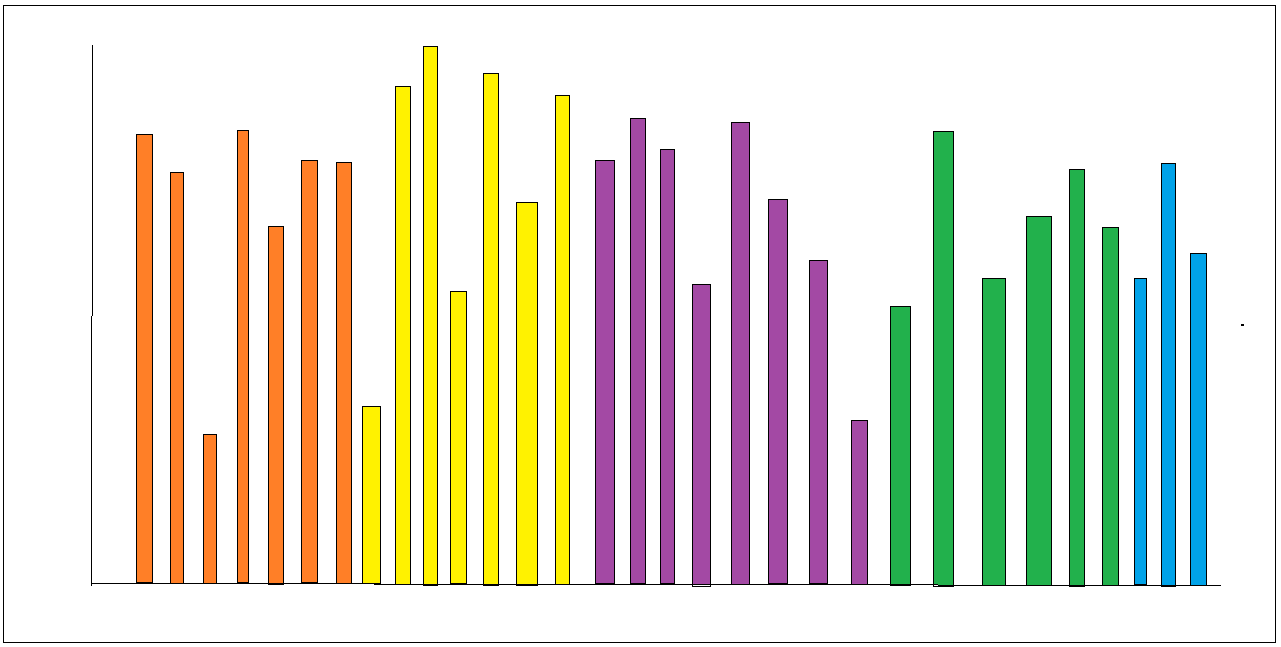
#### Wireframe 3: 2D Visualisatie – Meest verkochte pizza’s inclusief maten in 2015

* **Doel**: visualiseer welke pizza’s en maten het meest zijn verkocht in het jaar 2015.
* **Elementen**:
  + **Grafiek**: een staafdiagram dat pizza’s en verkoopaantallen variërend met bijbehorende verkoopcijfers weergeeft.
  + **Weergave:** X-as geeft pizza’s met maten weer. Y-as geeft verkoopaantallen weer.



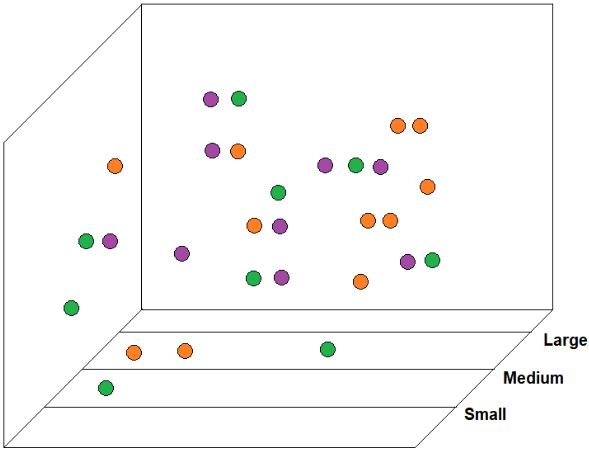
#### Wireframe 4: 2D Visualisatie per categorie – verkochte pizza’s

* **Doel**: geeft per categorie de verkochte aantallen pizza's weer, met een onderverdeling per type pizza binnen elke categorie.
* **Elementen**:
  + **Grafiek**: een staafdiagram dat de verkoopcijfers per pizza categorie weergeeft.
  + **Weergave**: kleurcodes voor snelle identificatie van categorieën. Y-as geeft verkoopaantallen weer. X-as geeft pizza’s per categorie weer.



#### Wireframe 5: 3D Visualisatie – Maandelijkse verkoop per pizza en maat

* **Doel**: laat het aantal verkochte pizza's zien per maand, per pizza en per maat voor een uitgebreide analyse.
* **Elementen**:
  + **Grafiek**: een Scatterplot-grafiek die per maand de verkoopdata toont voor elk type pizza in verschillende maten.
  + **Interactie**: mogelijkheid om de grafiek te roteren of in te zoomen voor specifieke details.
  + **Legenda**: kleurcoderingen om de verschillende pizza’s te onderscheiden.
  + **Weergave:** Y-as geeft verkoopaantallen weer. X-as geeft maanden weer. Z-as geeft pizzamaat weer.



**5. Python visualisaties**

**1D Visualisatie: percentage verkochte pizza's per categorie.**  
Een pie-chart is simpel en in een oogopslag te begrijpen. Hierdoor kan de gebruiker makkelijk zien hoeveel verkocht wordt per pizza categorie.

**2D Visualisatie (2015): meest verkochte pizzanaam en maat.**  
Door een barchart te gebruiken kan de gebruiker direct zien welke pizza het meest verkocht wordt in verhouding tot de andere verkochte pizza’s. Hier is in overleg met de opdrachtgever specifiek gekozen om de top 10 weer te geven, het resultaat is namelijk 1 pizza. Alleen is 1 resultaat niet te vergelijken, vandaar de top 10 om een verhouding tot de andere meest verkochte pizza’s weer te geven.

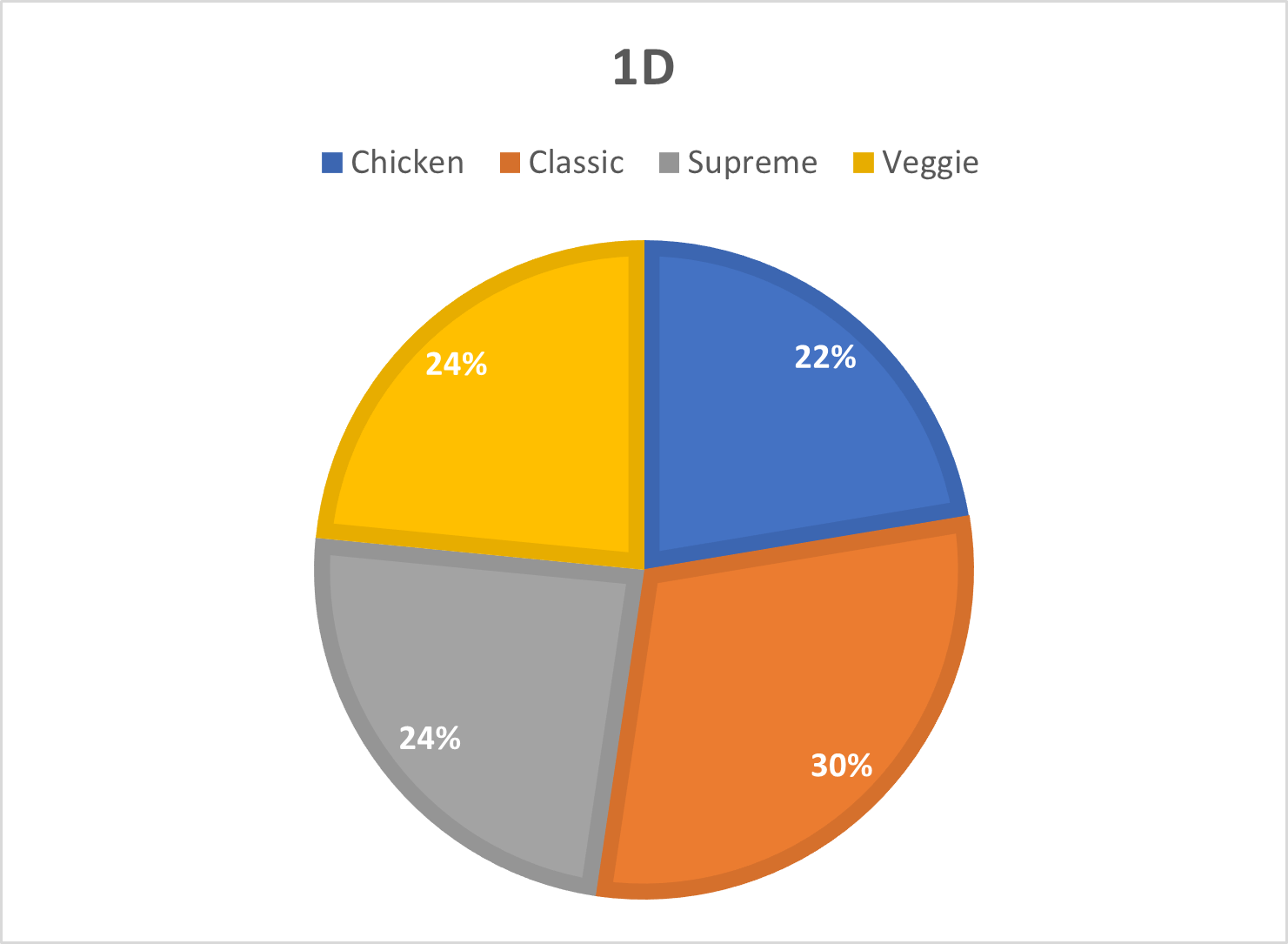
**2D Visualisatie per categorie: verkochte pizzanamen per categorie.**  
Een multi-gekleurde barchart geeft de mogelijk om categorieën te onderscheiden. Ook heeft de gebruiker overzicht van de verkoop per type pizza.

**3D Visualisatie: maandelijkse verkopen per pizzanaam en maat.**  
Voor deze visualisatie hebben we gekozen voor een 3D scatter plot (spreidingsdiagram). Hierdoor kunnen we aan de gebruiker alle gevraagde data in een visualisatie weergeven zonder dat er visuele overlap is tussen de verschillende maten of maanden.

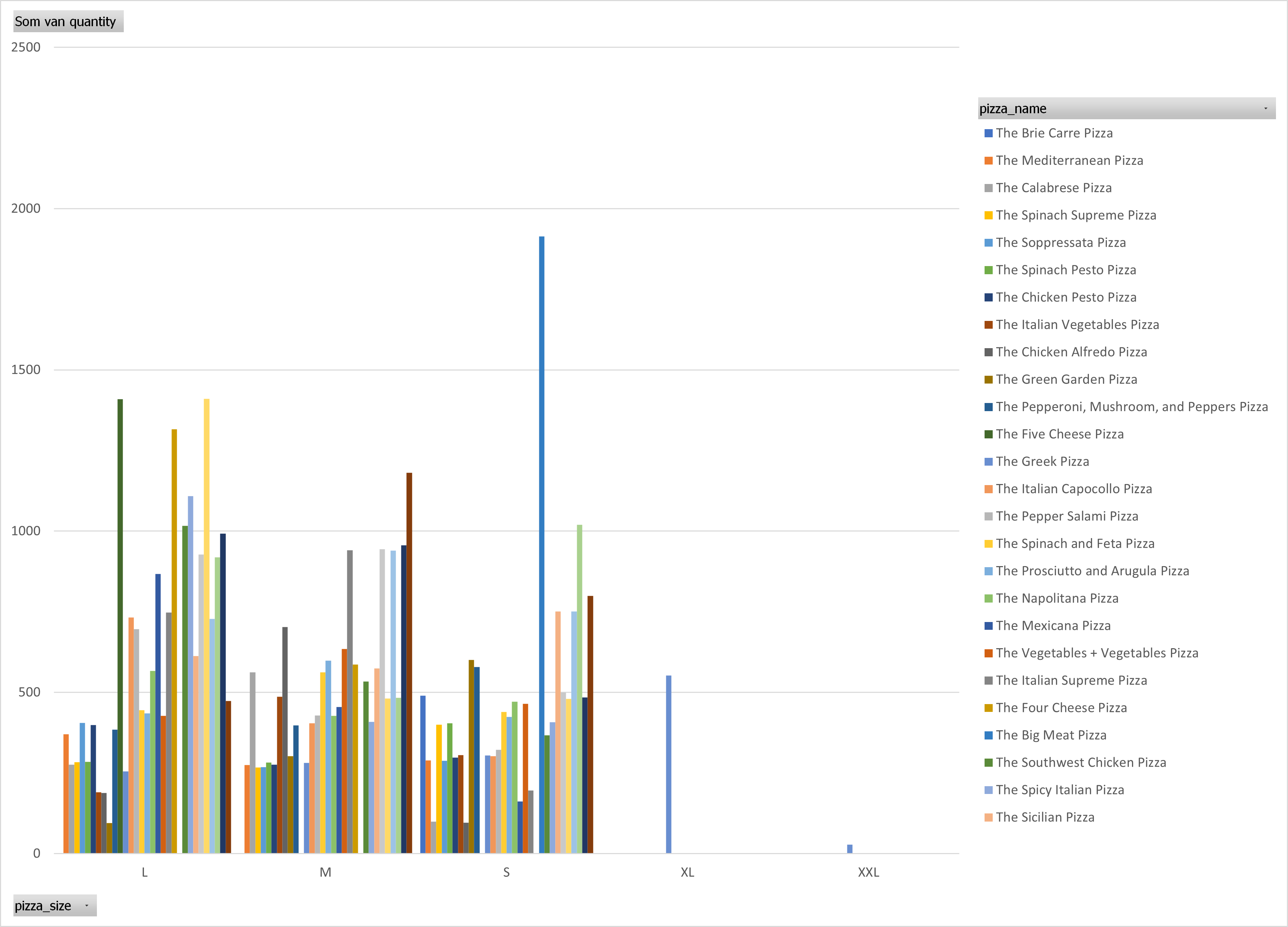
**6. Excel visualisaties**

Hieronder zijn de visualisaties die met behulp van Excel zijn gemaakt op basis van dezelfde data. Het verschil met de Python visualisaties is dat de Excel visualisaties veel gedetailleerder zijn, dat heeft als nadeel dat ongevraagde data ook wordt weergegven. De reden daarvoor is dat in Excel minder makkelijk kan worden gefilterd in vergelijking met Python. Hierdoor zijn de Python visualisaties een stuk bruikbaarder dan de visualisaties die met Excel zijn gegenereerd.

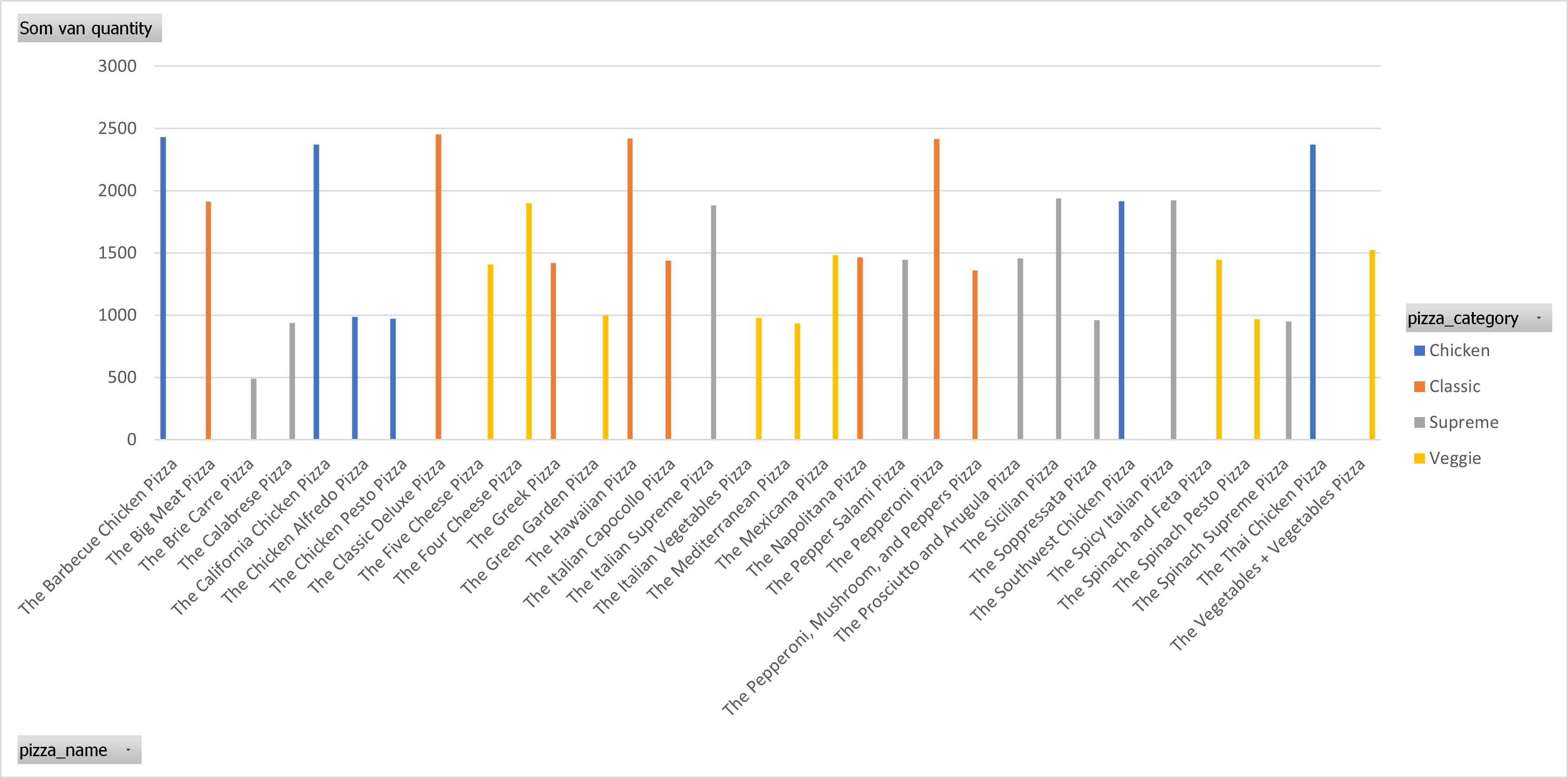
**1D Visualisatie: percentage verkochte pizza's per categorie.**



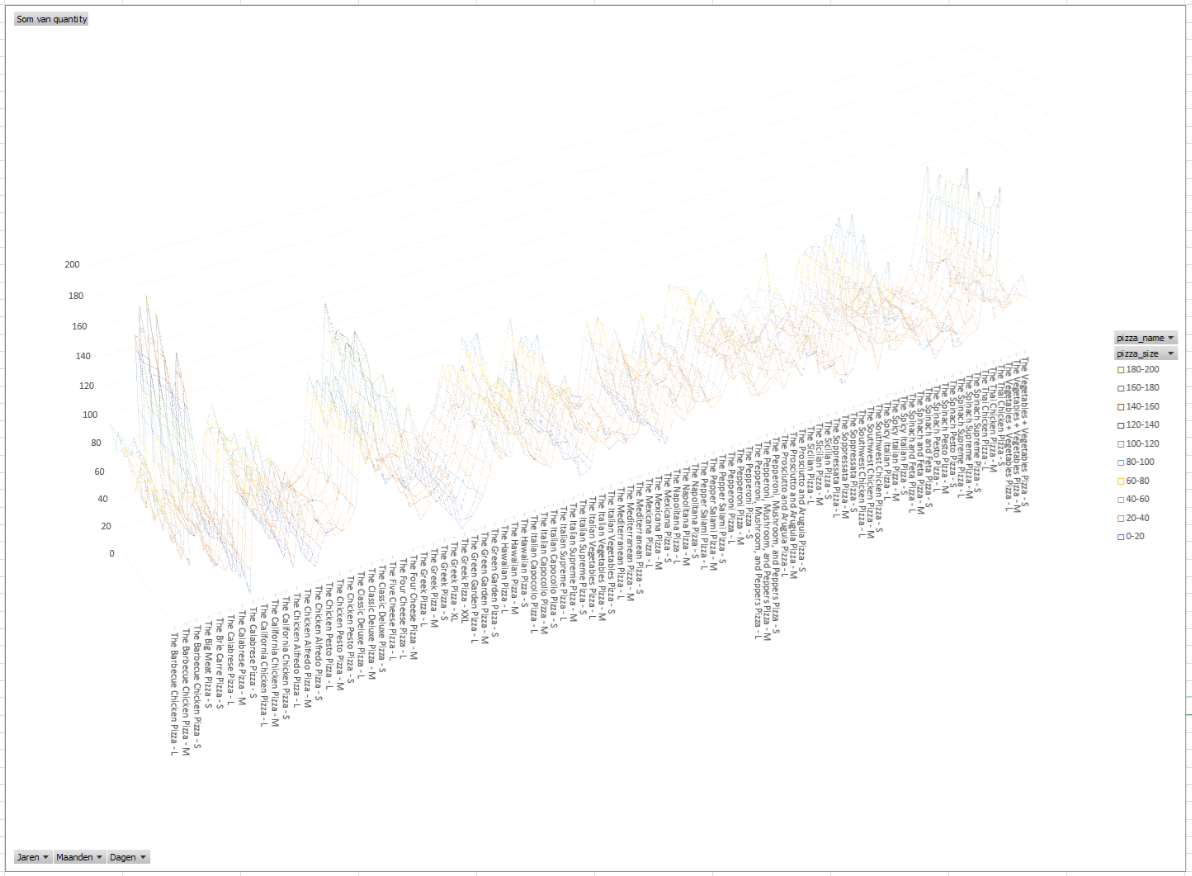
**2D Visualisatie (2015): meest verkochte pizzanaam en maat.**



**2D Visualisatie per categorie: verkochte pizzanamen per categorie.**



**3D Visualisatie per maand per pizzanaam en per maat het totaal aantal verkochte pizza’s**

****

**7. Code en hulpmiddelen**

Python code + libraries: zie de Python bestanden en/of GitHub: <https://github.com/KyleLukassen/Casus-Blok-1.git> Voor het uitvoeren van de code zijn verschillende Python libraries nodig. Deze moeten handmatig worden gedownload en geïnstalleerd.   
Excel-bestand: zie het ingeleverde Excel-bestand voor de gebruikte data en bovenstaande visualisaties.  
GitHub: samenwerken en delen van de Python bestanden.

Trello: project borden > <https://trello.com/b/cO1Y2Ps1/project-board>

**8. Logboek Scrum Master**

In het logboek van de scrum master is de documentatie van activiteiten, beslissingen en voortgang van elke sprint terug te vinden. Het logboek is als apart document (Excel bestand) bijgevoegd.

# 9. Testplan en testrealisatie

Testplan: het testplan omvat de teststrategie en scenario’s die elk aspect van het dashboard controleren.

## Functionaliteitstesten

* Visualisaties: controleer of de visualisaties de juiste gegevens tonen zoals beschreven in de user stories.

*Resultaat: Visualitaties leveren de gegevens zoals beschreven in de user stories.*

* Interactiviteit: 3D scatter plot is te roteren.

*Resultaat: Door met de rechtermuisknop te klikken en vast te houden is de scatter plot grafiek 360 graden te roteren.*

## Performance Testen

* Laadtijden: elke visualisatie moet binnen de aangegeven tijd laden. Afhankelijk van tijd die Python nodig heeft om de database aan te maken of te laden.

*Resultaat: Indien een database niet beschikbaar is, wordt deze aangemaakt. De gemiddelde tijd voor deze handeling is ongeveer 6 seconden. Iedere visualisatie laadt de data en geeft deze in maximaal 5 seconden weer.*

* Responsiviteit: dashboard moet soepel werken op diverse schermformaten.

*Resultaat: Het dashboard werkt zoals verwacht op 4 verschillende laptops met beeldschermen van verschillende afmetingen. Het dashboard ziet op alle 4 de apparaten exact hetzelfde uit qua design en formaat.*

## Data Validatie

* Excel validatie: test de Python visualisaties door Excel grafieken te genereren en waarden te vergelijken. Zie hiervoor het Excel-bestand en bovenstaande Excel visualisaties.

*Resultaat: De grafieken die in Excel gegenereerd worden komen overeen met de grafieken zoals Python deze genereert. Het is echter niet mogelijk om binnen Excel een 3D-scatterplot te genereren. Deze visualisatie valt daarom niet te controleren met Excel. Verder was het ook niet mogelijk om de data op dezelfde manier te filteren. Om deze reden zien de grafieken er niet 100% identiek uit, deze zijn echter wel correct en geven dezelfde resultaten als de grafieken binnen het dashboard.*

* Consistency check: data moet consistent en actueel zijn.

*Resultaat: Bij iedere activatie van een visualisatie wordt dezelfde grafiek getoond, visuele output is consistent. Aanpassingen in het Excel bestand worden weergegeven in de grafieken.*