

GERT JANSSENSWILLEN, BENOÎT DEPAIRE

EXPLORATIEVE EN DESCRIPTIEVE DATA ANALYSE

Contents

<i>Voorwoord</i>	5
<i>Hoe de lecture notes te gebruiken</i>	5
<i>Hoe de tutorials te gebruiken</i>	6
<i>Over de auteurs</i>	6
<i>Disclaimer</i>	6
1 [Lecture notes] Introductiecollege	7
1.1 Data science?	7
1.2 Data & Data types	18
1.3 Referenties	22
2 [Lecture notes] Data visualisatie	23
2.1 Perceptual ranking	24
2.2 Gestaltprincipes	24
2.3 Data Visualization Pitfalls	25
2.4 Univariate visualisaties (1 variabele)	25
2.5 Bivariate visualisatie (2 variabelen)	35
2.6 Multivariate visualisaties (meer dan 2 variabelen)	49
2.7 Visualisaties voor communicatie	55
2.8 How charts lie	56
2.9 Referenties	57

3	<i>[Tutorial] Data visualisatie</i>	59
3.1	<i>Voor je begint</i>	59
3.2	<i>Introductie</i>	59
3.3	<i>Verschillende geometries</i>	61
3.4	<i>Layout van onze grafieken verbeteren</i>	76
3.5	<i>Geavanceerde plots</i>	85
3.6	<i>Background material</i>	92

Voorwoord

Dit boek bevat de lecture notes en tutorials voor het opleidingsonderdeel “Exploratieve en Descriptieve Data Analyse” (1ste Ba Handelsingenieur/Handelsingenieur in de Beleidsinformatica) aan de Universiteit Hasselt. De lectures notes dienen ter begeleiding van de hoorcolleges, terwijl de tutorials telkens een vervolg zijn hierop ter voorbereiding van de werkzittingen.

Hoe de lecture notes te gebruiken

Het idee van de lecture notes is om een begeleidende tekst aan te reiken ter ondersteuning van de slide-decks die gebruikt worden tijdens de hoorcolleges. Deze tekst is “bullet-point”-gewijs opgebouwd en helpt het verhaal dat tijdens het hoorcollege wordt verteld terug op te roepen. Daarnaast zal er per hoofdstuk ook een *referentielijst* aangereikt worden met werken die de diverse topics in detail uitleggen.

- Neem de lecture notes mee naar het hoorcollege (digitaal of geprint), en gebruik deze om belangrijke aspecten tijdens het hoorcollege te markeren en korte nota's toe te voegen. Ga zeker niet de volledige uitleg van het hoorcollege noteren. Dit is vaak niet mogelijk en indien je er toch in slaagt zal je tijdens het hoorcollege niet in staat zijn geweest om een eerste keer te reflecteren over de leerstof.
- Bestudeer na de les de lecture notes samen met de notities. Controleer of je alles begrijpt en waar nodig noteer je aanvullingen. Probeer een overzicht te verkrijgen van de diverse concepten die je tijdens het hoorcollege bestudeerd hebt en tracht na te gaan hoe je deze inzichten kunt gebruiken voor exploratieve en descriptieve data analyse.
- (optioneel) Lees de bronnen in de referentielijst. Indien er elementen niet duidelijk zijn in je eigen notities of de lecture notes, dan ga je best gericht op zoek naar de antwoorden op je vragen in de referentiewerken.

Hoe de tutorials te gebruiken

De tutorials zijn een logisch gevolg op de leerstof in het hoorcollege en bereiden je voor op de oefening in de werkzittingen. In de tutorials worden de concepten uit het hoorcollege geïllustreerd in R code. Het is niet enkel de bedoeling de tutorials te lezen, maar ook zelf de voorbeelden uit te proberen in Rstudio. Je vindt de nodige datasets hiervoor telkens terug op Blackboard. Zonder de tutorials grondig te bekijken heeft het geen zin om naar de werkzittingen te komen.

Over de auteurs

dr. Gert Janssenswillen is academisch medewerker aan de faculteit Bedrijfseconomische Wetenschappen (BINF Business Informatics) van de Universiteit Hasselt. Na het verwerven van zijn diploma Handel ingenieur in de Beleidsinformatica in 2014, behaalde hij in 2019 een PhD in de Bedrijfseconomie aan de Universiteit Hasselt. Tijdens zijn doctoraat ontwikkelde hij de open-source R packages-suite bu-paR, welke wereldwijd gebruikt wordt door bedrijven en organisaties voor de analyse van bedrijfsprocessen. Hij spreekt regelmatig op business process management - conferenties, zoals BPM, ICPM, en SIMPDA, alsook R conferenties, zoals useR. Sinds 2019 is hij lid van het organisatie comité van de Europese R User Meeting (eRum).

Prof. dr. Benoît Depaire is hoofddocent Beleidsinformatica aan de Universiteit Hasselt en lid van de onderzoeksgroep Beleidsinformatica. Zijn onderzoeksinteresse situeert zich rond de topics data mining, data-gedreven procesanalyse en statistiek met een focus op de extractie van bedrijfskundige inzichten uit data. Als voorzitter van het onderwijsmanagementteam voor de opleiding Beleidsinformatica, alsook op basis van zijn jarenlange ervaring als docent, heeft hij een onderwijsexpertise uitgebouwd rond diverse topics zoals projectmanagement, business process management, data analyse en de rol van IT in de moderne bedrijfswereld. Daarnaast houdt hij zich ook bezig met dienstverlening naar de bedrijfswereld toe door middel van gastlezingen, adviesverstrekking en toegepaste onderzoeksprojecten.

Disclaimer

Niets uit deze uitgave mag worden verveelvoudigd, opgeslagen in een geautomatiseerd gegevensbestand en/of openbaar gemaakt in enige vorm of op enige wijze, hetzij elektronisch, mechanisch, door fotokopieën, opnemen of op enige andere manier zonder voorafgaande schriftelijke toestemming van de uitgever.

1

[Lecture notes] Introductiecollege

1.1 Data science?

1.1.1 Verschillende soorten van data analyse

- Er zijn verschillende manieren om data analyse taken te classificeren.
- De classificatie die we hier hanteren is gebaseerd op het doel van de data analyse.

Descriptieve data analyse

- Deze analyse focust zich op het beschrijven van de data.
- Deze analyse gaat over het samenvatten van de grote hoeveelheid data in enkele statistische cijfers en grafieken.
- Deze analyse wordt gebruikt als je een grote hoeveelheid data krijgt en je snel inzicht wilt krijgen in de data.
- Voorbeelden:
 - Je hebt een dataset met alle studieresultaten van de studenten van 1ste bachelor HI/BI en je wilt weten wat de gemiddelde score is per vak.
 - Je hebt de verkoopscijfers van het afgelopen jaar en je wil weten welke drie producten het beste verkochten (zowel in aantal als in omzet).
- Descriptieve data analyse zegt alleen iets over de realiteit die door de data is beschreven. Je kan **geen** conclusies trekken die verder reiken dan de geobserveerde data.
- Je kan een descriptieve data analyse vergelijken met het werk van een detective die als taak heeft een beschrijving te maken van de misdaadscene.

Exploratieve data analyse

- Exploratieve analyse focust op het verkennen van de data en het zoeken naar interessante patronen en afwijkingen van deze patronen.
- Net als bij descriptieve data analyse zal exploratieve analyse de beschikbare data beschrijven en zeggen de resultaten **niets** over ongeobserveerde feiten.
- In tegenstelling tot bij descriptieve data analyse, gaat exploratieve data analyse verder dan het louter beschrijven van de data en tracht men interessante patronen te ontdekken in de data.
- Voorbeelden:
 - Zijn er specifieke kenmerken van studenten die sterk gerelateerd zijn aan hun studieresultaten.
 - Zijn er opmerkelijke verschillen tussen vakken wat betreft de punten die behaald worden. Zo ja, wat zijn dan deze verschillen.
 - Zijn er producten in ons gamma die gevoelig zijn voor seizoenseffecten?
- Je kan een exploratieve data analyse vergelijken met het werk van een detective die als taak heeft verbanden te ontdekken tussen verschillende bewijsstukken om zo inzicht te verschaffen wat er gebeurd is tijdens de misdaad.

Confirmatorische data analyse

- Confirmatorische analyse focust op het bevestigen of weerleggen van vermoedens die men heeft met behulp van de beschikbare data.
- In tegenstelling tot descriptieve en exploratieve data analyse zal men bij confirmatorische data analyse wel conclusies trekken die verder gaan dan de geobserveerde data.
- Omdat confirmatorische data analyses ook uitspraken doen over ongeobserveerde data, is er altijd een mate van onzekerheid over de correctheid van de resultaten.
- Voorbeelden:
 - Halen studenten met 8u Wiskunde achtergrond betere resultaten dan studenten met 6u Wiskunde achtergrond? In welke mate zijn we zeker dat dit voor alle studenten geldt en niet enkel voor de studenten waarover we data hebben?
 - Verkoopt product X beter bij mannen dan bij vrouwen? In welke mate zijn we zeker dat dit verschil niet een toevalligheid in de data is?

- Je kan een confirmatorische data analyse vergelijken met het werk van een rechter die op basis van het aangeboden bewijsmateriaal moet beslissen of er genoeg bewijs is om iemand te veroordelen van de misdaad.

Predictieve data analyse

- Het doel van predictieve analyse is om op basis van de beschikbare data voorspellingen te doen over de toekomst of over nieuwe of alternatieve situaties.
- Net als bij confirmatorische data analyse zal predictieve data analyse uitspraken doen die ook van toepassing zijn voor ongeobsioneerde feiten/situaties.
- Bijgevolg is er net als bij confirmatorische data analyse dus een zekere onzekerheid over de conclusies die men trekt.
- Voorbeelden:
 - Zal een studente die met meer dan 80% haar diploma van het middelbaar onderwijs behaalt slagen in eerste zit voor het vak *Exploratieve en Descriptive Data Analyse*?
 - Zullen de verkoopcijfers van product Y het komende jaar verder stijgen en met hoeveel procent?
- Je kan een predictieve data analyse vergelijken met het werk van een detective die op basis van het bewijsmateriaal op een misdaadscene moet voorspellen waar en wanneer de dader opnieuw zal toeslaan.

1.1.2 Rol van data in de bedrijfswereld

- Er zijn verschillende redenen waarom bedrijven data bijhouden. Deze kunnen we onderverdelen in volgende categorieën: Geschiedenis bijhouden, beslissing nemen en voorspellingen maken.

Geschiedenis bijhouden

- Je registreert feiten zodat je achteraf met zekerheid kunt weten wat de realiteit in het verleden was.
- Dit is belangrijk als je wilt evalueren of een bedrijf goed beheerd wordt. Hiervoor heb je inzicht in het verleden nodig.
- De gegevens die worden bijgehouden in een boekhouding en jaarrekeningen zijn hier een typisch voorbeeld van.

Dagelijkse werking

- Omdat een bedrijf zijn dagelijkse werking kan uitvoeren, is het essentieel een up to date zicht te hebben van de werkelijkheid. Als

een klant belt met een klacht over een levering, dan moet je als onderneming kunnen achterhalen wat de klant precies besteld heeft, of dit reeds geleverd is, of de klant al betaald heeft, enzovoort.

Zonder deze informatie kan een onderneming haar dagelijkse werking niet garanderen.

- Om de dagelijkse werking te verzekeren, hebben bedrijven altijd al data bijgehouden. Denk maar aan informatie over aankoop- en verkooporders, de financiële gegevens in de boekhouding, de afschriften van een bank, de productieplanning, enzovoort.

Beslissingen nemen

- Een bedrijf neemt dagelijks talrijke beslissingen op verschillende niveaus
 - Operationeel.
 - * Vb: Moet ik een nieuwe bestelling plaatsen voor grondstof X of hebben we nog genoeg voorraad?
 - * Dit zijn typisch zeer frequente beslissingen die nodig zijn om de dagelijkse werking te garanderen.
 - * Deze beslissingen worden genomen door mensen op de werkplaats of door het (lager) management.
 - Tactisch/Management.
 - * Vb: Sluit ik best een exclusief contract af met 1 leverancier voor grondstof X voor een vaste periode en tegen een vaste verkoopprijs of koop ik wanneer nodig tegen de marktprijs?
 - * Deze beslissingen worden minder frequent genomen dan operationele beslissingen en zijn typisch nodig om de werking van de onderneming op middellange termijn te optimaliseren.
 - * Deze beslissingen worden genomen door het management van een onderneming en hebben een aanzienlijke impact.
 - Strategisch.
 - * Vb: Zullen we grondstof X aankopen op de markt of beslissen we deze grondstof zelf te produceren?
 - * Deze beslissingen hebben een zeer grote impact op de onderneming en worden niet frequent genomen. Ze vergen typisch ook lange voorbereidingstijd en bepalen de richting en toekomst van de onderneming op lange termijn.
 - * Deze beslissingen worden genomen door het topmanagement van een onderneming.
- Data kan bedrijven helpen bij het nemen van beslissingen.
 - Dit betekent echter niet dat beslissingen enkel en alleen op data gebaseerd zijn.

- Vaak wordt data gecombineerd met ervaring en expertise om een beslissing te nemen.
- Bij het nemen van beslissingen op basis van data, kunnen we zowel patronen in historische data gebruiken alsook voorspellen op basis van data.

Producten

- Data als inherent onderdeel van een product
 - Social media
 - Netflix
 - Spotify
 - Google
 - Uber
 - Deliveroo
 - ...
- Data heeft niet langer puur ondersteunende rol

Voorbeeld: Netflix

- Netflix Prize (2006)
 - Wereldwijde open competitie voor de constructie van een nieuw algoritme dat moest voorspellen hoe goed een klant een film zou beoordelen op basis van zijn of haar filmvoorkeuren.
 - Winnaar was het team dat als eerste een verbetering van 10% kon realiseren ten opzichte van het algoritme van Netflix zelf.
 - Eerste prijs was 1 miljoen USD.
 - Hiervoor stelde Netflix een dataset ter beschikking met 100 miljoen filmbeoordelingen van 500 000 klanten met betrekking tot 18 000 films.
- Het kunnen voorspellen hoe hun klanten gaan reageren op specifieke films/series laat Netflix toe hun aanbod aan films en series te optimaliseren om het huidige klantenbestand te behouden en nieuwe klanten aan te trekken.
- De hoeveelheid data die door Netflix wordt verzameld is enorm.
 - In 2016 had Netflix 93.8 miljoen leden.
 - Netflix weet wanneer je pauzeert.
 - Netflix weet op welke dagen en welke uren je kijkt.
 - Netflix weet wat je kijkt.
 - Netflix weet van waar je kijkt.
 - Netflix weet op welk soort toestellen je kijkt.
 - Netflix weet wanneer je definitief stopt met het bekijken van een serie.

- Netflix weet hoe snel je verschillende afleveringen van een serie achter elkaar kijkt.
- Netflix weet welke titels je zoekt.
- Netflix komt op deze manier zeer veel te weten over het kijkgedrag van zijn klanten en kan op basis van deze inzichten betere beslissingen nemen. Bijvoorbeeld:
 - Netflix ontdekt uit haar data dat 40% van haar klanten een serie zijn begonnen te kijken die door het oorspronkelijke productiehuis is stopgezet.
 - Stel dat Netflix uit de data ook ontdekt dat 85% van deze klanten de serie volledig uitkijken zonder dat het tempo waar tegen men afleveringen kijkt significant afneemt.
 - Op basis van deze inzichten kan Netflix eventueel beslissen om de rechten van de serie te kopen (die goedkoop zullen zijn aangezien de serie was stopgezet) en zelf een nieuw seizoen voor de serie te maken.
- House of Cards
 - Netflix deed het beste bod voor de serie House of Cards waardoor het won van kanalen zoals HBO.
 - Ze kochten initieel 2 seizoenen van de serie waar een prijskaartje aan vast hing van meer dan 100 miljoen dollar.
 - Deze beslissing was voor een groot stuk gebaseerd op data:
 - * Netflix leerde uit haar data dat haar klanten geïnteresseerd waren in producties van regisseur David Fincher.
 - * Netflix leerde uit haar data dat haar klanten geïnteresseerd waren in de oorspronkelijke Britse versie van House of Cards.
 - * Netflix leerde uit haar data dat haar klanten geïnteresseerd waren in producties met Kevin Spacey.
 - Maar ook na de beslissing om deze serie te maken, bleef Netflix haar data gebruiken om slimme beslissingen te nemen.
 - * Er werden verschillende trailers gemaakt en afhankelijk van je voorkeuren kreeg je een trailer op maat te zien.
 - * Klanten die vooral graag Kevin Spacey zagen, kregen een trailer waar vooral Kevin Spacey in voorkwam.
 - * Klanten die vooral geïnteresseerd waren in films van David Fincher, kregen een trailer te zien die de typische "look&feel" had van David Fincher.
 - * Klanten die ook de Britse versie hadden gezien, kregen een trailer te zien die vooral op het verhaal focuste.

1.1.3 Data revoluties

- Data over de maatschappij
 - Het verzamelen van data is iets dat teruggaat tot in de oudheid.
 - Denk hierbij aan de volkstellingen die reeds plaatsvonden ten tijden van de Romeinen.
 - Een volkstelling gaat alle inwoners van een bevolking registreren, samen met diverse kenmerken zoals burgerlijke status, leeftijd, geslacht, enzovoort.
 - Volkstellingen waren en zijn nog steeds belangrijk voor een overheid om de impact van haar openbaar beleid te kunnen inschatten.
- Scientific Management
 - Frederick Taylor
 - Eind 19de eeuw
 - Benaderde het organiseren van werk op een wetenschappelijke manier.
 - Ging data verzamelen om vervolgens te analyseren hoe men werk efficiënter kon organiseren.
 - Een van de eerste vormen van dataverzameling en -analyse om bedrijfswaarde (productiviteit) te creëren.
 - Beperkt in hoeveelheid data omdat registratie en analyse nog manueel gebeurde.
- Het ontstaan van het digitale tijdperk
 - Met de uitvinding van de computer tijdens en na de tweede wereldoorlog, is de mensheid het digitale tijdperk ingegaan.
 - De computer zorgt ervoor dat we data in een digitale vorm (als een reeks van één en nullen) opslaan. Dit biedt het voordeel dat exacte kopieën van de data gemaakt kunnen worden met één muisklik.
- Digitalisatie van de werkvloer
 - Computers op de werkvloer dateert terug tot midden vorige eeuw, maar de grote doorbraak komt er met de opkomst van de personal computer
 - * 1977: Apple Home Computer II
 - * 1981: IBM Personal Computer
 - * Eind jaren 80, begin jaren 90 was de PC wijdverspreid op de werkvloer.
 - * Dit liet toe meer data te registreren, maar deze was nog moeilijk te delen met andere computers.
 - Opkomst Internet/WWW in de bedrijfswereld

- * 1990: De technologie voor WWW werd publiek gedeeld door Tim Berners-Lee.
- * Dankzij WWW en internettechnologie werd het steeds een-voudiger om digitaal werk te delen.
- Opkomst van e-commerce
 - * 1995: Begin van dot-com bubble/hype.
 - * Opkomst van digitale ondernemingen (vb. Amazon, Netflix, Google, ...).
 - * Digitale handel maakt het eenvoudiger om gegevens hierover te registreren.
- Digitalisatie van mensen
 - Opkomst Web 2.0 (begin 2000)
 - * Inhoud van het web wordt nu gecreëerd door de bezoekers/gebruikers/klanten.
 - * Websites worden dynamisch (passen zich aan de context en bezoeker aan).
 - Opkomst sociale media
 - * Gebruikers gaan spontaan hun leven digitaliseren.
 - * Hiervoor worden diverse media gebruikt (foto, video, tekst, ...).
 - * Facebook, Twitter, Instagram, Persoonlijke blogs,
 - * Nog nooit heeft zo'n groot deel van de wereldbevolking informatie gecreëerd en gedeeld met de rest van de wereld.
- Digitalisatie van dingen
 - Opkomst goedkope sensoren
 - Steeds meer "dingen" (machines, auto's, huishoudtoestellen, huizen, steden, ...) worden 'intelligent'.
 - Internet of Things (IoT): Al deze intelligente dingen worden via het Internet met elkaar verbonden.
 - De hoeveelheid data die hiermee gegenereerd zal worden is ongezien.
 - Volgens IDC studie waren in 2013 reeds 7% van de "verbibbare dingen" geconnecteerd aan het Internet of Things.
 - In dezelfde studie voorspellen ze dat dit zal stijgen tot 15% in 2020.
 - In 2013 werd 2% van alle data in het digitaal universum geproduceerd door het IoT.
 - Verwacht wordt dat dit zal stijgen tot 10% in 2020.

1.1.4 Data explosion

- De hoeveelheid data die de laatste decennia gegenereerd en opgeslagen wordt is enorm toegenomen.

- Deze groei is exponentieel (de groei gaat steeds sneller). Meer specifiek verdubbelt de hoeveelheid data in het digitaal universum iedere 2 jaar.
- In 2018 bestond het digitaal universum uit 33 Zetabytes data
 - 1 Zetabyte = 1024 Exabytes
 - 1 Exabyte = 1024 Petabytes
 - 1 Petabyte = 1024 Terabytes
 - 1 Terabyte = 1024 Gigabytes
- Volgens studies zal het digitaal universum in 2035 uit 2142 Zetabytes bestaan.
 - Dit is een toename van ca. 28% per jaar, i.e. een verdubbeling elke 3 jaar.

1.1.5 Waarover verzamelen bedrijven data

- Het ultieme doel van een onderneming is gegevens te verzamelen die hen toelaten om het gedrag van hun omgeving beter te begrijpen, alsook de werking van hun eigen onderneming.
- Onder omgeving verstaan we:
 - Klanten
 - Concurrenten
 - Leveranciers
 - Alternatieve markten
 - Overheden
- Onder werking van eigen onderneming vertaan we o.a.:
 - Werknemers
 - Processen
 - Producten
 - Diensten

1.1.6 Van data tot 'actionable insights'

- Data
 - Data verwijst typisch naar de gegevens die geregistreerd en opgeslagen worden.
 - Data beschrijft een heel klein aspect van een realiteit (bijvoorbeeld op welk exact tijdstip ben ik aflevering 2 van "House of Cards" beginnen te kijken).
 - Data op zich heeft echter heel weinig waarde.
- Informatie

- Als we echter data gaan analyseren, dan kunnen we dit transformeren tot informatie.
- Informatie beschrijft een realiteit en gaat typisch op zoek naar patronen in de data en afwijkingen op deze patronen.
- Bijvoorbeeld: Ik kijk typisch House of Cards gedurende de week om 20u00 's avonds, maar stop meestal met kijken om 20u30, waardoor ik in de week zelden een aflevering in 1 keer uitkijk.
- Informatie is beschrijvend en zegt ons WAT de realiteit is.
- Actionable Insights
 - Actionable Insights is informatie die ons niet enkel zegt WAT de realiteit is, maar ons ook het inzicht verschafft HOE we moeten handelen.
 - Niet alle informatie is actionable.
 - Op basis van actionable insights en in combinatie met onze eigen ervaringen en kennis die we reeds bezitten, komen we soms tot inzichten die beschrijven HOE we moeten handelen.

1.1.7 Data Scientists

- Nieuwe jobomschrijving.
- Verantwoordelijk om data te transformeren naar 'actionable insights' en hier iets mee te doen om bedrijfswaarde te creëren.
- Omschreven als meest 'sexy job' van de 21ste eeuw door HBR
 - Opvolgers van de Wall Street 'Quants' uit de jaren 80 en 90.
- Vaardigheden
 - Bedrijfskunde
 - * Productontwikkeling
 - * Management
 - Machine Learning / Big Data
 - * Ongestructureerde data
 - * Gestruktureerde data
 - * Machine Learning
 - * Big Data
 - Wiskunde en Operationeel Onderzoek
 - * Optimalisatie
 - * Wiskunde
 - * Simulatie
 - Programmeren
 - Statistiek
 - * Visualisatie

- * Tijdreeksanalyse
- * Wetenschappelijk onderzoek
- * Data Manipulatie
- 4 profielen van data scientists
 - Data Businessperson
 - * Focust voornamelijk hoe data omzet kan genereren.
 - * Vaak in een leidinggevende rol.
 - * Werken zelf ook met data en beschikken over de nodige technische vaardigheden.
 - Data Creatives
 - * Zijn in staat een volledige data analyse zelfstandig uit te voeren.
 - * Hebben een hele brede bagage aan technische vaardigheden.
 - * Beschikken in zekere mate over bedrijfskundige vaardigheden.
 - * Gaan vaak innovatief om met data.
 - Data Developer
 - * Is voornamelijk gefocust op de technische uitdagingen met betrekking tot het beheer van data.
 - * Sterke programmeervaardigheden. Zijn in staat productiecode te schrijven.
 - * Zijn sterk in het gebruik van machine learning technieken.
 - Data Researcher
 - * Vaak mensen met een wetenschappelijke achtergrond (doctoraat).
 - * Sterk in statistische vaardigheden en wetenschappelijk onderzoek.

1.1.8 De kunst van data analyse

- Data analyse is een kunst. Net als bij iedere kunst, kunnen we hierbij drie componenten onderscheiden: kennis en vaardigheden, ervaring en creativiteit.
- Kennis en vaardigheden
 - Als data analyst moet je de juiste hulpmiddelen kunnen identificeren voor het voorgelegde probleem.
 - Deze diverse hulpmiddelen moet je zo goed mogelijk beheersen.
 - Bij (exploratieve) data analyse gaat het hierbij zowel over analyses als over datavaardigheden.
 - Dit aspect kun je leren en laat je reeds toe om correcte analyses uit te voeren.
- Ervaring

- Hoe meer data je analyseert, hoe beter je er in wordt.
- Ook laat ervaring toe om sneller vaste patronen in je werk te herkennen en efficiënter te worden in wat je doet.
- Ervaring is ook essentieel om complexere uitdagingen beheersbaar te maken.
- Dit deel kunnen we je niet 'leren', maar heb je wel volledig in de hand.
- Creativiteit
 - Een kunstenaar die over kennis, vaardigheden en ervaring beschikt, maar creativiteit ontbreekt, kan perfecte replica's maken van een kustwerk, maar kan zelf geen nieuwe kunst creëren.
 - Creativiteit is in staat zijn op een nieuwe en onverwachte manier naar data te kijken en deze te visualiseren.
 - Het is niet zeker dat dit aspect aan te leren is. Maar dit hoeft niet te verhinderen dat je een goede data scientist wordt, zolang je maar voldoende aandacht besteedt aan de andere twee componenten.

1.1.9 De kracht van descriptieve en exploratieve data analyse

<https://www.youtube.com/watch?v=RUwS1uAdUcI>

1.2 Data & Data types

- Data is het resultaat van een meting van een attribuut van een specifiek object met een specifiek meetinstrument.
 - Het object verwijst naar wat je gaat meten.
 - * vb.: Student "Karel Jespers".
 - Een object hoort meestal tot een verzameling van objecten. Deze verzameling wordt ook wel de populatie genoemd.
 - * vb.: Populatie "Studenten 1ste Ba HI/BI".
 - Een specifiek object uit de populatie wordt ook wel element genoemd.
 - * vb.: "Karel Jespers" is een element uit de populatie "Student 1ste Ba HI/BI".
 - Je meet altijd een specifiek aspect van het object. Omdat de meetwaarde van dit aspect kan variëren tussen verschillende objecten (elementen) in je verzameling (populatie), worden zulke aspecten ook variabelen genoemd.
 - * vb.: Lengte is een specifiek aspect (variabele) van de student "Karel Jespers" (element).

- De meting gebeurt met behulp van een meetinstrument. Het is belangrijk te beseffen dat een meetinstrument altijd een zekere nauwkeurigheid heeft (tot hoeveel cijfers na de komma exact kan je meten?) en mogelijk ook onderhevig kan zijn aan willekeurige en/of systematische meetfouten.
 - * vb.: Student "Karel Jespers" wordt gemeten met een meetlat bevestigd tegen de muur. De meetlat heeft een nauwkeurigheid van 1cm, dus we kunnen zijn lengte niet uitdrukken in milimeters. Verder is de meetlat 2cm te laag opgehangen. Bijgevolg is er een systematische meetfout van 2cm. Tenslotte wordt de meting geregistreerd door een arts die vluchtig kijkt waar de student uitkomt op de meetlat. Het is dus niet onmogelijk dat de werkelijke lengte (willekeurig) afwijkt van de geregistreerde lengte.
 - * Tenzij anders vermeld wordt, gaan we in dit hoofdstuk uit van meetinstrumenten met oneindige nauwkeurigheid en zonder meetfouten.
- De uitkomst van een meting voor een specifiek element wordt de waarde genoemd.
 - * vb.: 1m80 is de waarde van de variabele "lengte" voor element "student Karel Jespers"

1.2.1 Dataset

- Een dataset is een verzameling van data waarbij
 - Iedere rij één element uit de populatie voorstelt.
 - Iedere kolom een variabele is die gemeten wordt.
 - De verschillende rijen verschillende elementen uit dezelfde populatie voorstellen.
 - De waarde in een cel de meting is van de betreffende variabele voor het betreffend element.

luchthaven	maatschappij	datum	vertrek_vertraging
EWR	United Air Lines Inc.	2013-01-01 05:15:00	2
LGA	United Air Lines Inc.	2013-01-01 05:29:00	4
JFK	American Airlines Inc.	2013-01-01 05:40:00	2
LGA	Delta Air Lines Inc.	2013-01-01 06:00:00	-6
EWR	United Air Lines Inc.	2013-01-01 05:58:00	-4
EWR	JetBlue Airways	2013-01-01 06:00:00	-5
LGA	ExpressJet Airlines Inc.	2013-01-01 06:00:00	-3
JFK	JetBlue Airways	2013-01-01 06:00:00	-3
LGA	American Airlines Inc.	2013-01-01 06:00:00	-2
JFK	JetBlue Airways	2013-01-01 06:00:00	-2

Table 1.1: Uitgaande vluchten NYC

1.2.2 Klassieke datatypologie

- Klassieke onderverdeling van data

- Nominaal, Ordinaal, Interval en Ratio
- Gebaseerd op de publicatie “On the Theory of Scales of Measurement” (1946)
 - * Beschrijft een hiërarchie van ‘datatypes’
 - Alles wat ordinaal is, is ook nominaal, maar niet omgekeerd.
 - Alles wat interval is, is ook ordinaal, maar niet omgekeerd.
 - Alles wat ratio is, is ook interval, maar niet omgekeerd.
 - * Identificeert geschikte statistische testen voor ieder type.
- Ieder datatype voldoet aan één of meerdere van de volgende eigenschappen:
 - Identiteit: Iedere waarde heeft een unieke betekenis.
 - Grootorde: Er is een natuurlijke volgorde tussen de waarden.
 - Gelijke intervals: Eenheidsverschillen zijn overal even groot.
Dus het verschil tussen 1 en 2 is even groot als het verschil tussen 19 en 20.
 - Absoluut nulpunt: De waarde 0 betekent dat er ook feitelijk niets aanwezig is van de variabele en is niet een arbitrair gekozen nulpunt.

Nominaal

- Voorbeelden:
 - Geslacht: Man, Vrouw.
 - Ondernemingsvorm: vzw, bvba, nv.
- Voldoet enkel aan de eigenschap ‘identiteit’.
- Dit betekent dat we enkel concluderen of twee waardes gelijk zijn of niet. Er bestaat geen natuurlijke volgorde tussen de verschillende waardes.

Ordinaal

- Voorbeeld:
 - Opleidingsniveau: Lager onderwijs, Middelbaar onderwijs, Hoger onderwijs.
 - Klanttevredenheid: Ontevreden, Matig tevreden, Tevreden, Zeer tevreden.
- Voldoet aan de eigenschappen ‘identiteit’ en ‘grootorde’.
- Dit betekent dat we niet alleen kunnen concluderen of twee waardes gelijk zijn of niet. Het is ook mogelijk te bepalen welke waarde ‘groter’ is.
- We kunnen echter niet zeggen hoeveel groter één waarde is dan de andere.

Interval

- Voorbeeld:
 - Temperatuur (Celsius).
- Voldoet aan de eigenschappen ‘identiteit’, ‘grootorde’ en ‘gelijke intervals’.
- We kunnen nu twee waardes vergelijken, bepalen welke groter is alsook de verschillen tussen waardes met elkaar vergelijken.
 - We kunnen dus stellen dat het verschil tussen 8 en 9 graden Celsius daadwerkelijk minder groot is dan het verschil tussen 12 en 20 graden Celsius.

Ratio

- Voorbeeld:
 - Gewicht
- Voldoet aan alle 4 de eigenschappen.
- We kunnen verschillende gewichten met elkaar vergelijken, we kunnen bepalen wat zwaarder is en we kunnen gewichtsverschillen onderling vergelijken. Hierbij komt nu ook nog dat we kunnen zeggen hoeveel keer iets zwaarder is dan iets anders.
- Dit is een gevolg van het feit dat de waarde o nu feitelijk betekent dat iets geen gewicht heeft.

1.2.3 De klassieke datatypologie is misleidend

- Voorbeeld:
 - Op een feestje wordt bij het binnengaan oplopende nummers toegewezen aan iedere gast, beginnend bij 1.
 - Tijdens het feestje wordt er een tombola georganiseerd en wie nummer 126 heeft, heeft gewonnen.
 - 1 gast vergelijkt dit nummer met haar kaartje en ziet dat ze gewonnen heeft. Zij beschouwde de waarde op haar ticket dus als een nominale variabele want het enige wat ze vergelijkt is of de waarde op haar ticket verschillend is van de winnende waarde.
 - Een andere gast kijkt naar zijn kaartje en ziet dat hij nummer 56 heeft. Hij concludeert dat hij te vroeg is binnengekomen en beschouwt de waarde op zijn kaartje dus als ordinaal.
 - Nog een andere gast heeft een kaartje met nummer 70 en beschikt over bijkomende data omtrent het ritme waarmee gasten zijn binnengekomen. Deze gast kan dus schatten hoeveel

later hij had moeten binnenkomen om te winnen en interpreteert zijn nummer dus als een interval variabele.

- Dit voorbeeld illustreert dat het datatype niet een vaststaand kenmerk is van de data, maar afhankelijk is van de vraag die je tracht te beantwoorden en de extra informatie waarover je beschikt.

1.2.4 Alternatieve datatypologie

- Alternatieve taxonomie van data
 - Graden: vb. academische graad: "op voldoende wijze", "onderscheiding", "grote onderscheiding", ... (geordende labels)
 - Rangordes: vb. plaats in voetbalklassement: 1, 2, 3, ..., 16 (gehele getallen die beginnen bij 1)
 - Fracties: vb. percentage opgenomen verlof: van 0% tot 100% (ligt tussen 0 en 1, als percentage uit te drukken).
 - Aantallen: vb aantal kinderen: 0, 1, 2, ... (niet-negatieve gehele waarden).
 - Hoeveelheden: vb. inkomen (niet-negatieve reële waarden).
 - Saldo: vb. winst (negatieve en positieve reële waarden).
- Voor deze cursus volstaat het meestal een onderscheid te maken tussen nominale en ordinale variabelen (samen "categorische" variabelen) en continue variabelen
 - Categorisch:
 - * Nominaal
 - * Ordinaal.
 - Continu: (Interval + Ratio)

1.3 Referenties

1. Data Scientist, the Sexiest Job of the 21st Century
2. Netflix Prize
3. How Netflix Uses Analytics
4. The Digital Universe of Opportunities - website
5. The Digital Universe of Opportunities - videoclip
6. How the Computer Changed the Office Forever
7. History of Computers in the Workplace
8. Web 1.0, 2.0, 3.0
9. From Data to Understanding
10. Analyzing the Analyzers
11. Scales of Measurement
12. Nominal, Ordinal, Interval, and Ratio Typologies are Misleading

2

[Lecture notes] Data visualisatie

- Vaak de eerste stap om zicht te krijgen op de data.
- Relatief eenvoudig om patronen te zien, maar minder geschikt om exacte waarden te zien.
- We moeten hierbij onderscheid maken tussen exploratieve visualisaties en informatieve visualisaties om een boodschap over te brengen.
 - Exploratieve visualisaties dienen om snel inzicht te krijgen in patronen in de data. Men besteedt hierbij veel minder aandacht aan de opmaak van de visualisatie. Vaak is deze visualisatie tijdelijk en niet bedoeld voor communicatie naar derden.
 - Communicatieve visualisaties dienen om een boodschap over te brengen aan derden. Hier dient men heel veel aandacht te besteden aan de opmaak zodat de boodschap duidelijk en helder gecommuniceerd wordt.
- We kunnen bij exploratieve visualisaties een onderscheid maken tussen univariate, bivariate en multivariate visualisaties.

De grafieken in dit hoofdstuk zijn gebaseerd op volgende dataset omtrent vluchten vertrekende uit New York.

```
## Rows: 329,174
## Columns: 7
## $ luchthaven      <fct> EWR, LGA, JFK, LGA, EWR, EWR, LGA, JFK, LGA, JFK, ~
## $ maatschappij    <chr> "United Air Lines Inc.", "United Air Lines Inc.", ~
## $ datum           <dttm> 2013-01-01 05:15:00, 2013-01-01 05:29:00, 2013-01-
## $ vertrek_vertraging <dbl> 2, 4, 2, -6, -4, -5, -3, -3, -2, -2, -2, -2, -2, -~ 
## $ aankomst_vertraging <dbl> 11, 20, 33, -25, 12, 19, -14, -8, 8, -2, -3, 7, -1~
## $ afstand          <dbl> 1400, 1416, 1089, 762, 719, 1065, 229, 944, 733, 1~
## $ vliegtijd         <dbl> 227, 227, 160, 116, 150, 158, 53, 140, 138, 149, 1~
```

2.1 *Perceptual ranking*

Niet alle elementen waarmee een visualisatie wordt opgebouwd zijn even makkelijk accuraat te begrijpen.

Geordend van meest accuraat naar minst accuraat

- Positie op gemeenschappelijke schaal
- Positie op (identieke) niet-gemeenschappelijke schalen
- Lengte
- Richting, helling & hoek
- Oppervlakte
- Volume
- Schaduw en saturatie
- Kleur

In realiteit complexer dan bovenstaande ranking

- ook link met type variabele
 - e.g. kleur werkt beter met nominale variabele dan met continue variabele

Perceptie is niet altijd het meest belangrijke - data visualisatie als kunstvorm - sommige elementen van een grafiek zijn belangrijker (liefst hoog op percentie ranking) dan andere (mogen lager op ranking)

Perceptie vs visuals - geen 1 op 1 relatie: keuze van uitwerking visual bepaald waar op de perceptuele ranking je je bevindt - niet alles in een visualisatie heeft een gelijkwaardige perceptuele ranking

2.2 *Gestaltprincipes*

Principes die helpen bepalen hoe we visuele elementen waarnemen

- Proximity: elementen dichter bij elkaar behoren tot dezelfde groep
- Similarity: we groeperen elementen op kleur, vorm, richting, etc.
- Enclosure: begrensde objecten worden ervaren als groep
- Closure: neiging om hiaten te negeren en mentaal aan te vullen
- Continuity: elementen op één lijn, of elementen die elkaar voorzetten, worden als groep gezien
- Connection: verbonden elementen worden als groep gezien

Preattentative processing = gebruik van deze principes op de nadruk op bepaalde elementen te leggen: contracts creëren.

2.3 Data Visualization Pitfalls

Data Visualisatie kan bepaalde cognitieve biasen versterken

- Framing: het vertellen van een bepaald verhaal kan worden verstrekt door een data visualisatie, waardoor vergeten wordt de context van een andere hoek te benaderen.
- Availability: makkelijker toegankelijke informatie wordt beschouwd als belangrijker/relevanter. Door visualisaties wordt dit versterkt
- Overconfidence: meer vertrouwen in grafisch informatie
- Anchoring: wanneer verwachtingen overmatig worden gebaseerd op bepaalde eikpunten. In visualisatie kan dit worden gedaan met een bepaalde configuratie van assen of annotaties.
- Confirmation bias: visualisaties kunnen bevestigend werken in het licht van eerdere informatie. Kritisch blik en andere perspectieven blijven nodig.

2.4 Univariate visualisaties (1 variabele)

- Als we slechts 1 variabele bestuderen, dan zijn we voornamelijk geïnteresseerd in de spreiding van de data. Dit wordt de verdeling van de data genoemd.
- Welke vragen kunnen we beantwoorden met dit soort visualisaties?
 - Wat is de meest voorkomende waarde van de data? Dit wordt ook de modus genoemd.
 - Bezit de data 1 modus, i.e. 1 waarde die duidelijk dominant is, of meerdere modi?
 - * Indien er slechts 1 afgetekende modus is, dan wordt de verdeling unimodaal genoemd.
 - * Indien er meerdere modi zijn (dominante waarden), dan wordt de verdeling multimodaal genoemd.
 - * Een multimodale verdeling kan er op wijzen dat de objecten in je data niet allemaal van hetzelfde type zijn en dat je in feiten twee populaties in je data aanwezig hebt.
 - Is de data geconcentreerd rond de modus of eerder breed verspreid. Met andere woorden, wat is de spreiding? Dit geeft inzicht in de variabiliteit van de data.
 - Is de data gelijkmatig verdeeld aan weerszijden van de modus of zien we duidelijk meer data aan één zijde van de verdeling? Indien er meer data aan één zijde van de verdeling ligt (ten opzichte van de modus) dan zegt men dat de verdeling asymetrisch verdeeld is.

- Zijn er waarden die opmerkelijk ver van de modus verwijderd zijn en geïsoleerd zijn van andere observaties? Dit worden extreme waarden of outliers genoemd. Deze verdienen meestal extra aandacht.

2.4.1 Categorische variabele

Staafdiagram

- Op de X-as staan de verschillende waarden van de categorische variabele. (Fig. 2.1)
- Bij iedere waarde tekenen we een verticale balk die aangeeft hoe vaak die waarde in de dataset voorkomt.

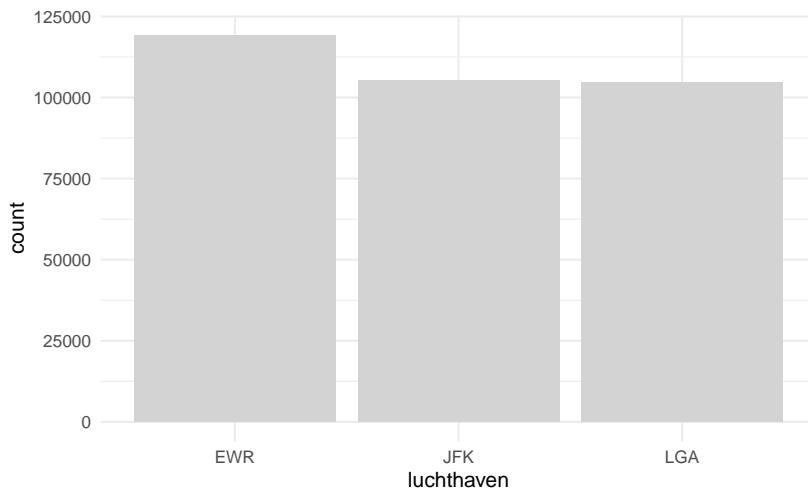


Figure 2.1: Staafdiagram luchthavens

- Minder geschikt indien er veel waarden zijn. Dan wordt de X-as snel onleesbaar.(Fig. 2.2)
- Je kan natuurlijk de labels roteren. Maar dit kan nog steeds onhandig zijn om te lezen. (Fig. 2.3).
- In geval van een **nominale** variabele zijn er twee mogelijkheden om de waarden te rangschikken:
 - Alfabetisch. (standaard) Dit is handig om snel waarden terug te vinden.
 - Volgens frequentie. Dit is handig om snel te zien welke waarden vaak/weinig voorkomen en geeft ook een beter beeld van de verdeling van de waarden. (Fig. 2.4)

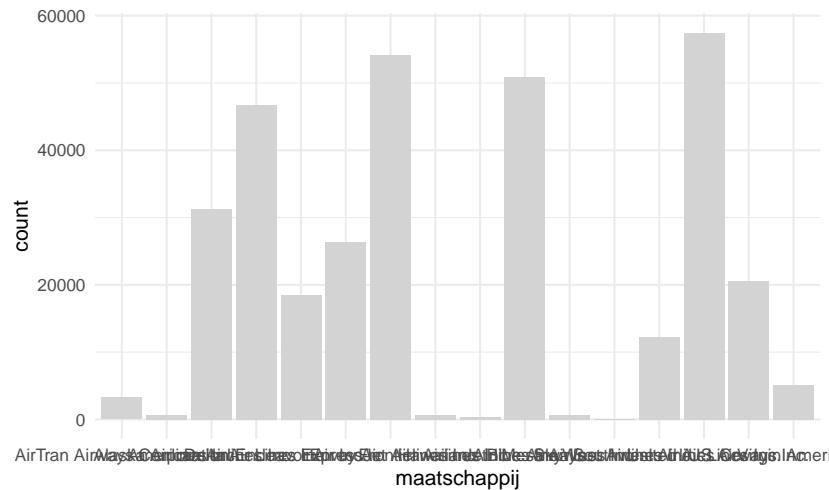


Figure 2.2: Staafdiagram maatschappijen

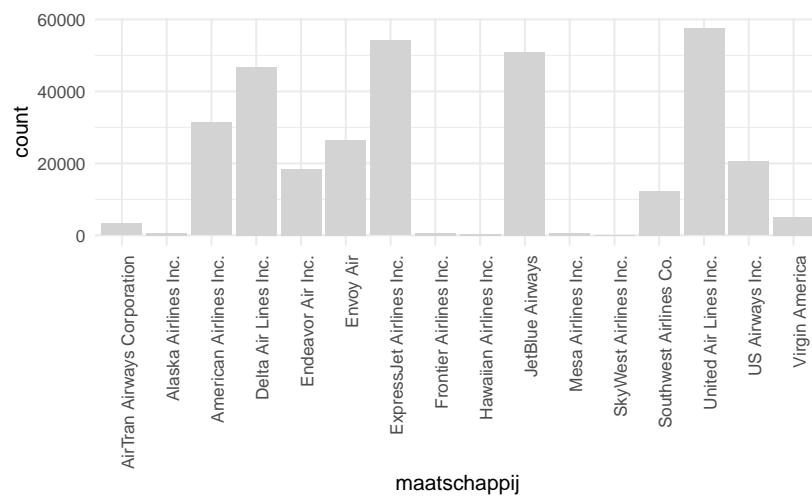


Figure 2.3: Staafdiagram met geroteerde labels

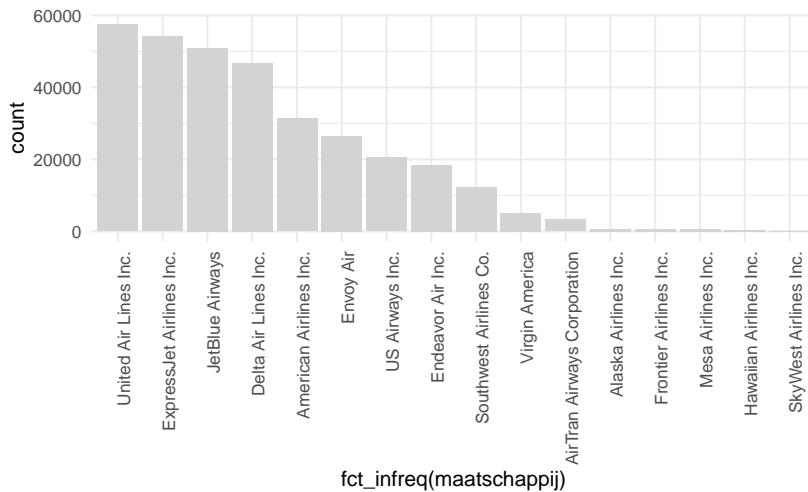


Figure 2.4: Staafdiagram gesorteerd op frequentie

- In het geval van een **ordinale** variabele houd je best de intrinsieke volgorde van de waarden aan.
- Je kan ook een horizontaal staafdiagram maken. (Fig. 2.5)
 - Zelfde principe, maar dan met horizontale balken.
 - Is handiger om de verschillende waarden te lezen, vooral indien dit er veel zijn.

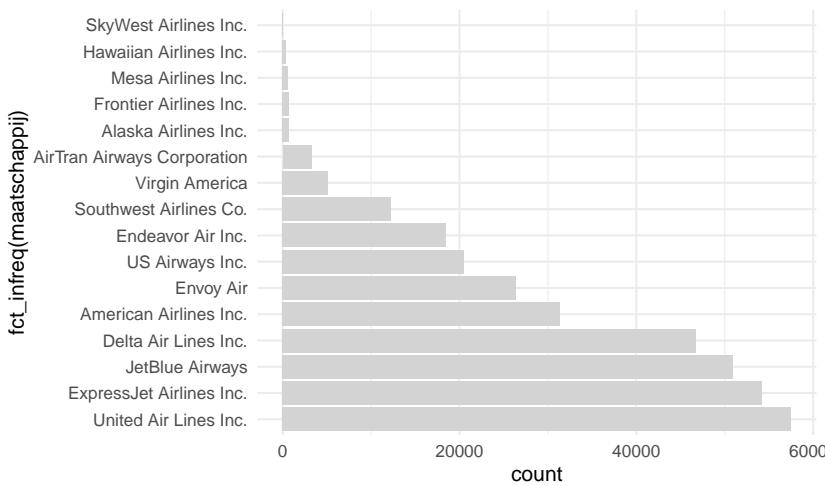


Figure 2.5: Verticaal staafdiagram gesorteerd op frequentie

Dotplot

- In plaats van balken te gebruiken om de frequentie van een waarde aan te geven, kan je dit ook met punten doen. (Fig. 2.6)

- Een dotplot laat duidelijker zien waar de sprongen in de verdeling zit. Daarom is de dotplot vooral relevant als je de waarden ordent volgens frequentie.

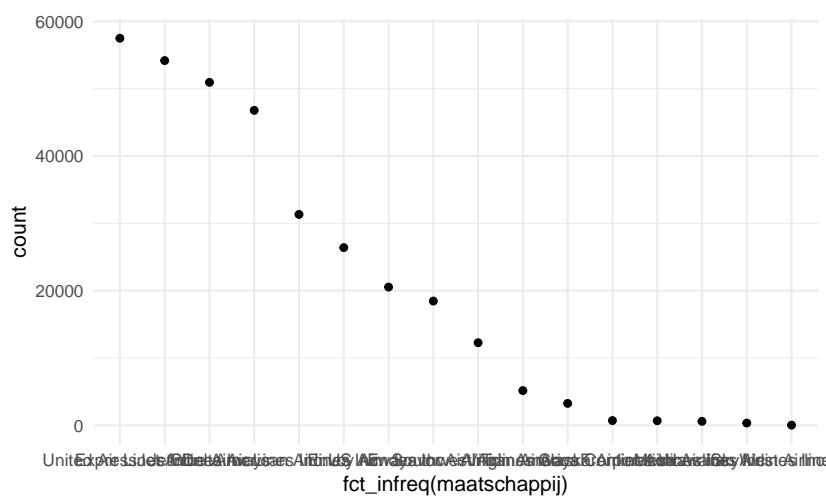


Figure 2.6: Dotplot maatschappij

- Net als de barplot kan je zowel een verticale als horizontale dotplot maken. (Fig. 2.7)

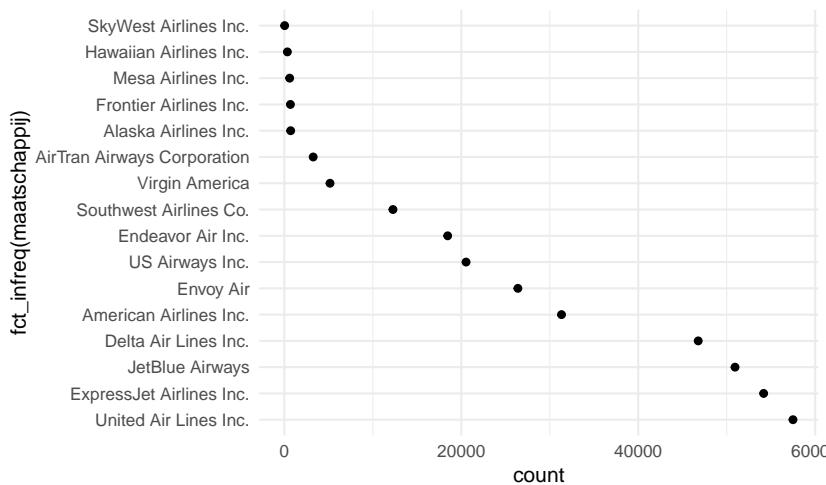


Figure 2.7: Verticale dotplot

'Stacked' staafdiagram

- We maken nu slechts 1 kolom. Iedere waarde is een andere kleur en neemt een deel van de balk in beslag. De volledige balk stelt 100% van de data voor. (Fig. 2.8)

- Kan nuttig zijn om data cumulatief te bestuderen.
- Hiermee kunnen we vragen beantwoorden zoals: "Welke waarden moeten we nemen om met zo weinig mogelijk waarden x% van de objecten te hebben?"

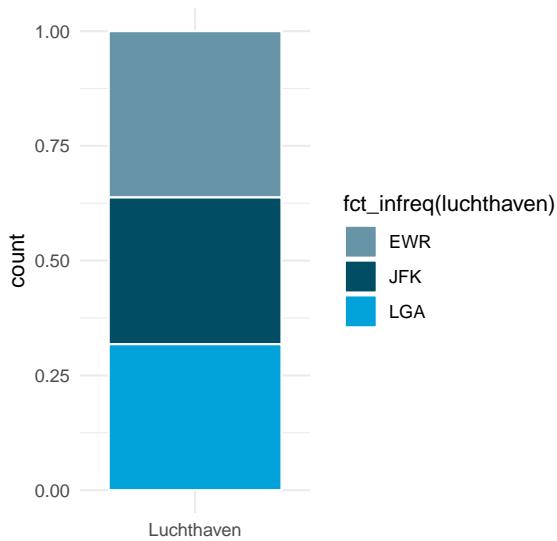


Figure 2.8: Stacked barplot

- We kunnen ook horizontale versies maken. (Fig. 2.9)

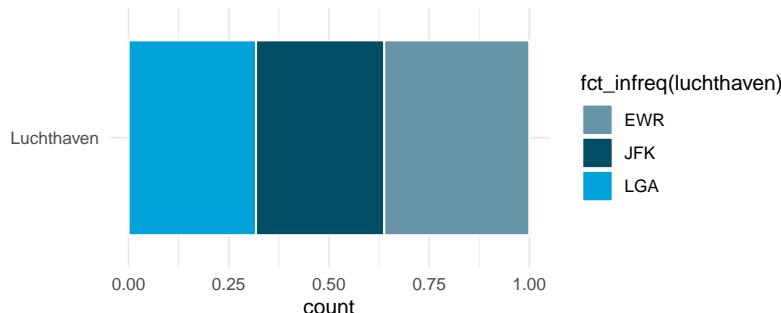


Figure 2.9: Horizontale stacked barplot

- Univariate stacked barcharts kunnen soms wat *raar* overkomen.
Vaak komt een gewone barchart beter over.

Andere soorten

- treemap: indelen van rechthoekige oppervlakte volgens categorische variabelen

- pie chart

- Moeilijk te interpreteren.
- Verschillen tussen waarden zijn enkel duidelijk bij grote verschillen, terwijl barplots en dotplots deze ook bij kleine verschillen kunnen tonen.
- Voor cumulatieve analyses van de data zijn barplots beter omdat het hier eenvoudiger is om af te leiden waar x% zicht bevindt.

2.4.2 Continue variabele

Histogram

- Analoog met barplot, alleen gaan we hier eerst onze “categorieën” definiëren. (Fig. 2.10)
- Dit wordt ‘binning’ genoemd en wordt bepaald door een binbreedte te kiezen.
- Je kan de binbreedte rechtstreeks kiezen of bepalen door vast te leggen hoeveel categorieën/bins je wenst.

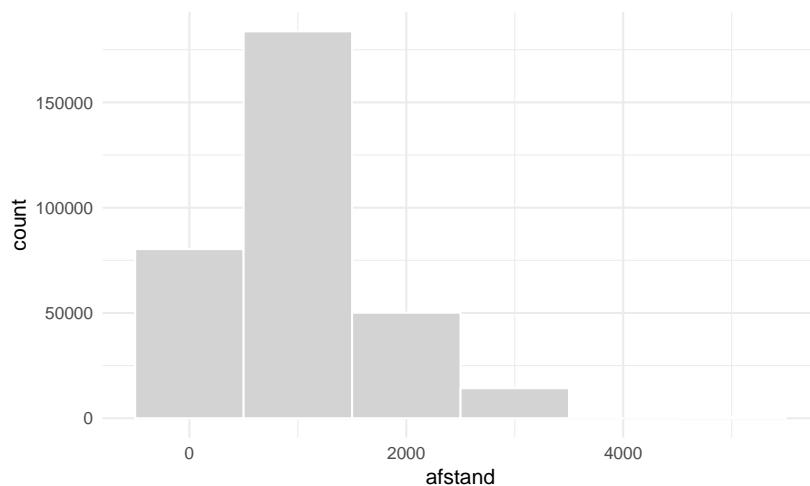


Figure 2.10: Histogram with binwidth 1000

- Voor de visualisatie, worden alle waarden gegroepeerd per ‘bin’.
- De binbreedte kan een enorme impact hebben op het uitzicht van de verdeling. (Fig. 2.11 - 2.12)
 - Hoe breder de bins, hoe minder modi je kan detecteren.
 - Hoe smaller de bins, hoe meer modi je gaat zien, hoewel dit niet altijd even betekenisvol is.

- Hoe smaller de bins, hoe minder data er in iedere bin gaat zitten en dan kunnen patronen wel in jouw dataset bestaan maar louter ten gevolge van toeval.

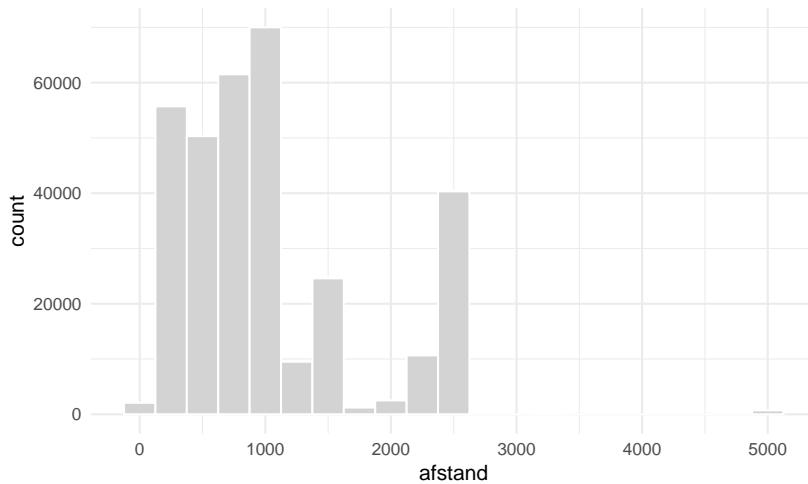


Figure 2.11: Histogram with binwidth 250

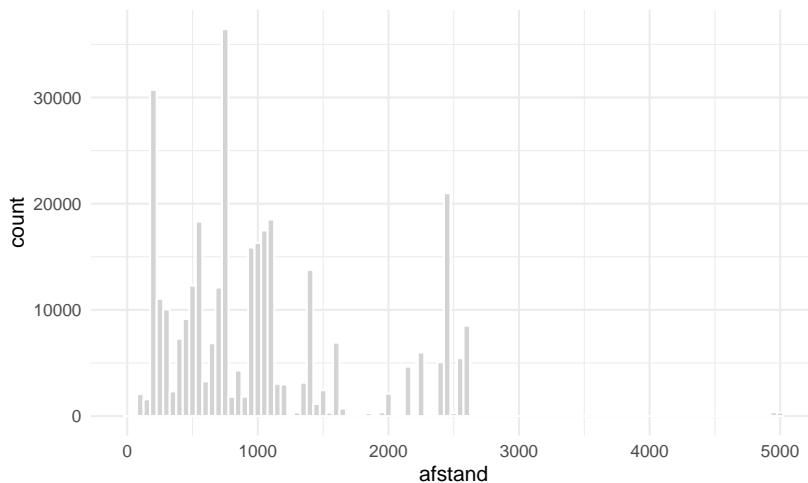


Figure 2.12: Histogram with binwidth 50

Density

- Variant van histogram.
- In plaats van staven wordt er een curve getekend. (Fig. 2.13)
- De oppervlakte onder de curve is steeds gelijk aan 1
- Hoe hoger de curve, hoe meer observaties ter hoogte van deze waarde (hoe hoger de densiteit)

- De waarde van de y-as heeft geen directe betekenis.

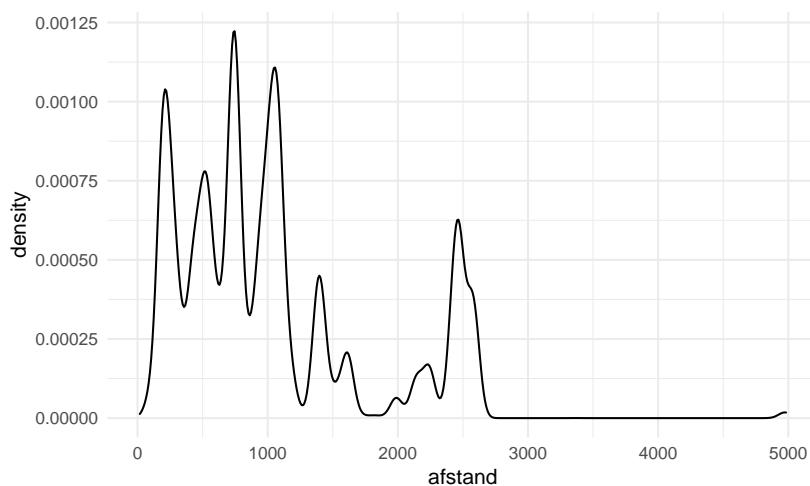


Figure 2.13: Density plot

Boxplot

- De lijn in het midden duidt de mediaan aan. Dit betekent dat 50% van je data onder deze lijn ligt, terwijl 50% er boven ligt. (Fig. 2.14)
- De box in het midden duidt de middelste 50% van je data aan. Dit wordt ook de interkwartiel-box genoemd. Dit betekent dat 25% van je data onder deze box zit en nog eens 25% boven deze box ligt. Hoe groter de box, des te meer de data gespreid is.
- Indien de box aan één zijde van de mediaanlijn groter is dan aan de andere zijde, dan wijst dit er op dat de data meer gespreid is aan die kant.
- De "whiskers" geven de laatste datapunten aan die als "normaal" beschouwd worden. Datapunten buiten deze grenzen beschouwt een boxplot als outliers of extreme waarden.
- De grens waar data van normaal naar extreem overgaat wordt door de boxplot bepaald door anderhalf keer de grootte van de interkwartiel-box op te tellen (en af te trekken) van de bovenste (onderste) grens van de interkwartiel-box. Punten die hier buiten liggen zijn outliers en worden als aparte punten aangeduid. De uitersten van de whiskers duiden de laatste datapunten aan binnen deze grenzen.
- Het is niet abnormaal dat er outliers in je data aanwezig zijn.
- Bij normaal verdeelde data zal je gemiddeld 7 outliers per 1000 datapunten mogen verwachten.

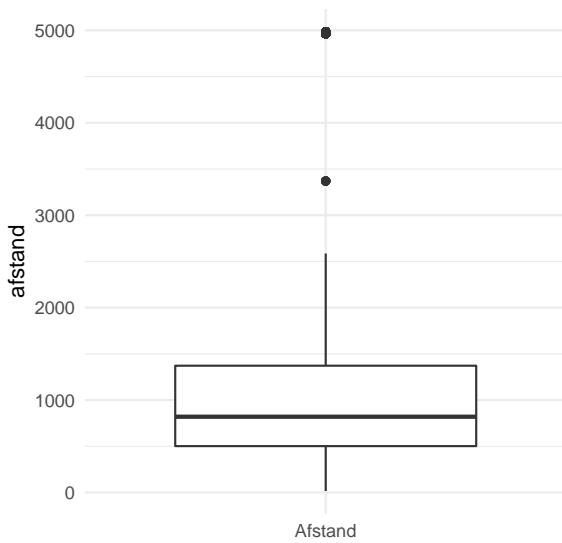


Figure 2.14: Verticale boxplot vertrekvertraging

- Een normale verdeling is een bepaalde manier waarop data waarden verdeeld kunnen zijn die in de realiteit vaak voorkomt.
- Indien je echter veel meer outliers ziet op je boxplot visualisatie, dan is de kans reëel dat er meer aan de hand is:
 - Er zijn bijvoorbeeld systematische meetfouten
 - De objecten in je data zijn in feite op bepaalde aspecten significant verschillend waardoor je ze apart zou moeten bestuderen.
- Je kan een boxplot ook roteren. (Fig 2.15)
- Boxplots komen beter tot hun recht bij bivariate analyses dan bij univariate analyses.

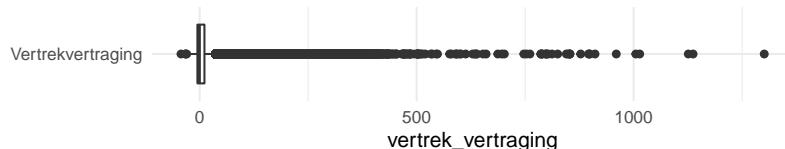


Figure 2.15: Horizontale boxplot vertrekvertraging

Violin plot

- Een violin plot kan je beschouwen als een combinatie van een histogram en een boxplot. (Fig. @ref(fig:2_10a))
- Net als bij een boxplot wordt op verticale wijze getoond hoe de data verspreid is.
- Opnieuw kan je ervoor kiezen de grafiek te roteren. (Fig. @ref(fig:2_10b))

- Net als bij een histogram kan je goed zien waar het volume (de massa) van de data zich bevindt.
- Net als bij een histogram kan je detecteren hoeveel modi de data bezit.
- In tegenstelling tot de boxplot, kan je bij een violinplot wel niet duidelijk zien waar bijvoorbeeld het ‘midden’ van je data is.

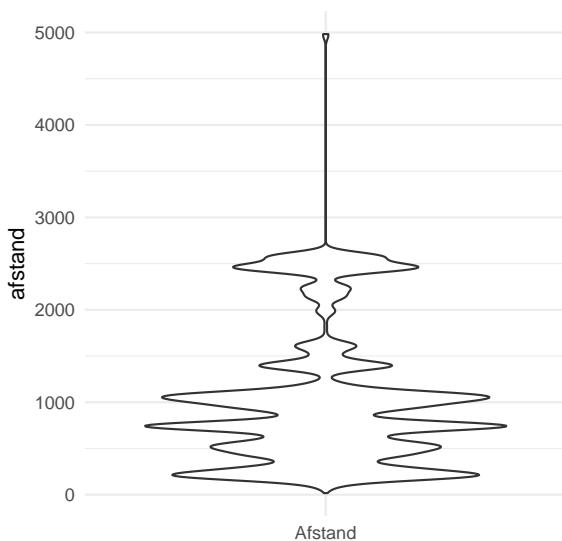


Figure 2.16: Verticale violin plot afstand

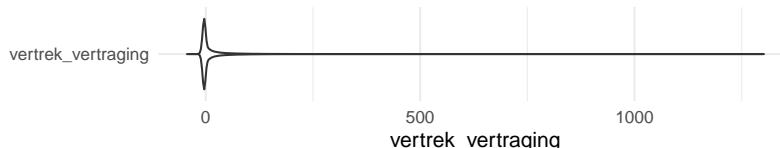


Figure 2.17: Horizontale violin plot vertrekvertraging

Jitter plot

- puntenwolk waarbij willekeurige “noise” (ruis) wordt toegevoegd.
- de ruis zorgt ervoor dat datapunten niet overlappen, en dat het duidelijk is waar de massa zich bevindt.
- Fig. 2.18 toont een vergelijking van violin, boxplot, point en jitter plot.

2.5 Bivariate visualisatie (2 variabelen)

- Wanneer we de relatie tussen 2 variabelen bekijken is het een-voudig te denken in *oorzaak* en *gevolg termen*.¹

¹ Zie opmerking i.v.m. correlatie versus causaliteit, 2.8.

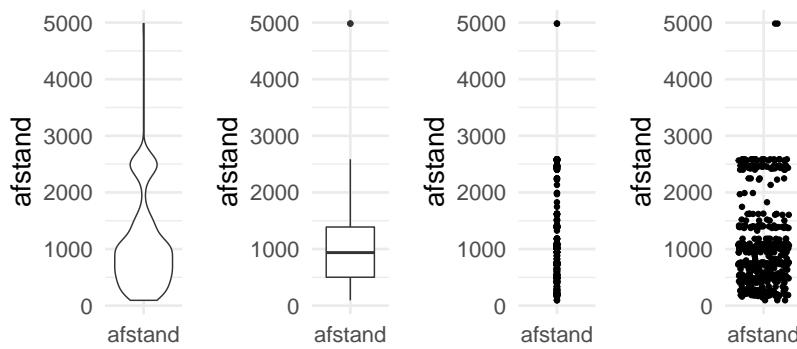


Figure 2.18: Violin, boxplot, point en jitter

- De variabele die we het label “oorzaak” geven, zullen we voorstaan “onafhankelijke variabele” noemen.
- De variabele die we het label “gevolg” geven, zullen we voorstaan “afhankelijke variabele” noemen.
- Waar we eigenlijk in geïnteresseerd zijn bij een visualisatie van 2 variabelen is de impact van de onafhankelijke variabele op de afhankelijke variabele weer te geven.
- Alle vragen die we kunnen stellen bij de visualisatie van één variabele, kunnen we nog steeds stellen, met telkens de bijkomende vraag of het waargenomen patroon verandert als de onafhankelijke variabele van waarde verandert.

2.5.1 Situatie 1: De onafhankelijke variabele is categorisch

Indien de afhankelijke variabele een continue variabele is kan je:

- meerdere boxplots op 1 grafiek visualiseren, met telkens 1 boxplot per waarde van de onafhankelijke variabele. (Fig. 2.19)
- meerdere violinplots op 1 grafiek tonen, met telkens 1 violinplot per waarde van de onafhankelijke variabele. (Fig. 2.20)
- meerdere histogrammen op 1 grafiek tonen
 - Hiervoor gebruiken we facetten: we tekenen voor elke waarde van de onafhankelijke variabele een apart assenstelsel. (Fig. 2.21)
- meerdere density plots
- Hiervoor kunnen we facetten gebruiken, ofwel de density plots over elkaar tekenen en onderscheiden met kleur. (Fig. 2.22-2.23)

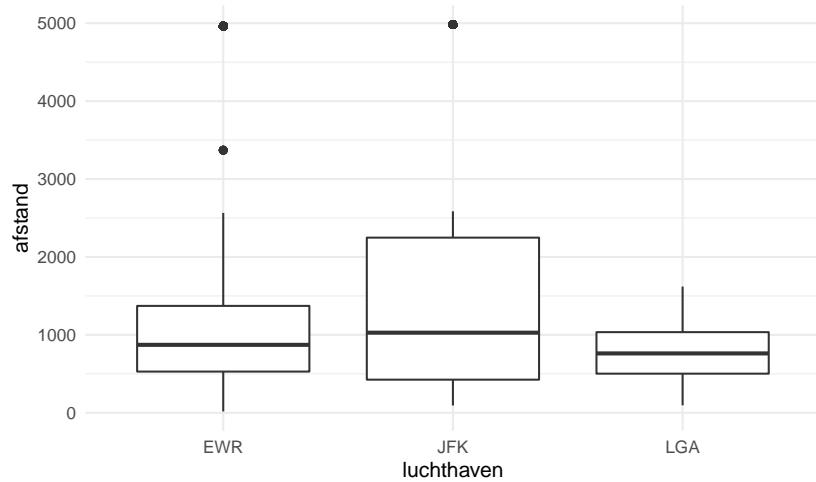


Figure 2.19: Bivariate boxplot

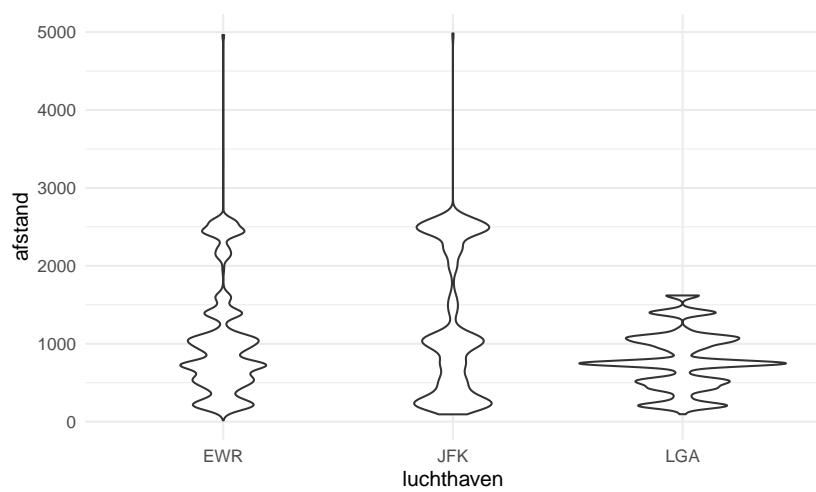


Figure 2.20: Bivariate violin plot

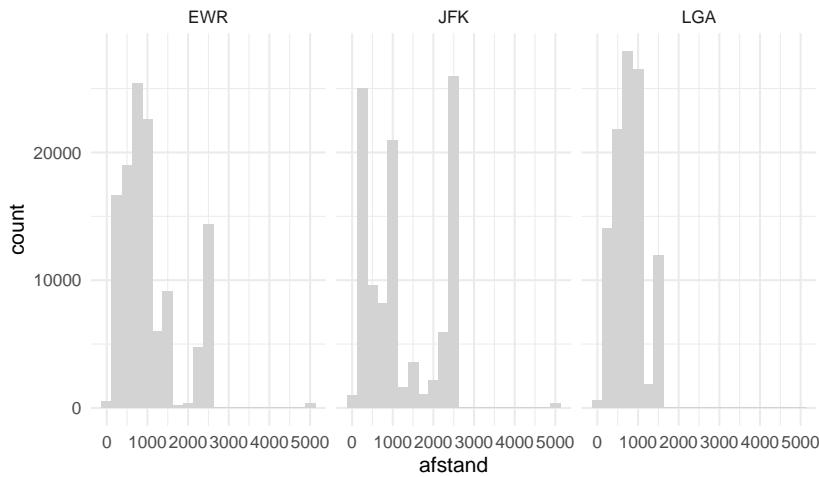


Figure 2.21: Bivariate histogram plot

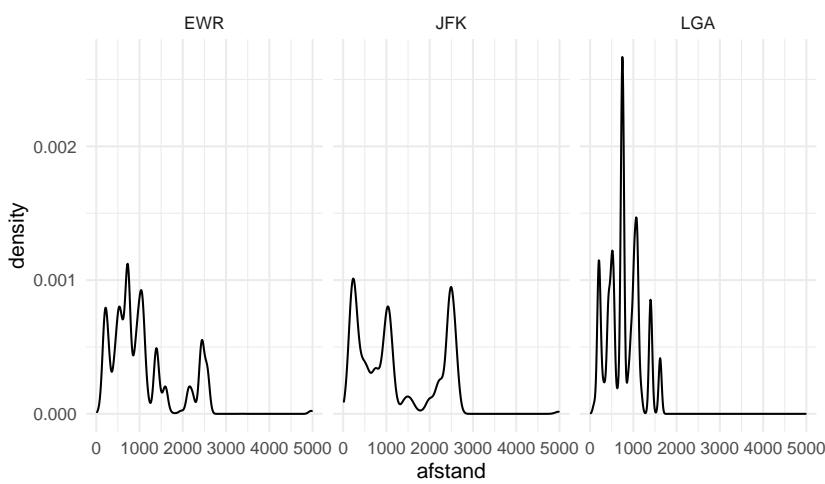


Figure 2.22: Bivariate density plot - apart

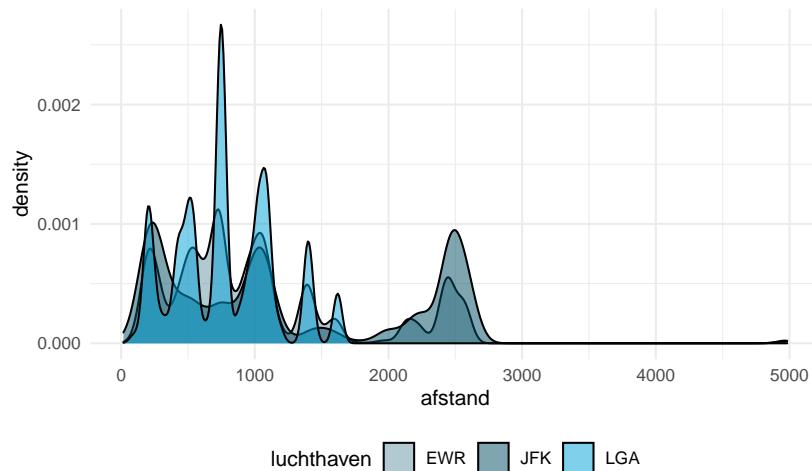
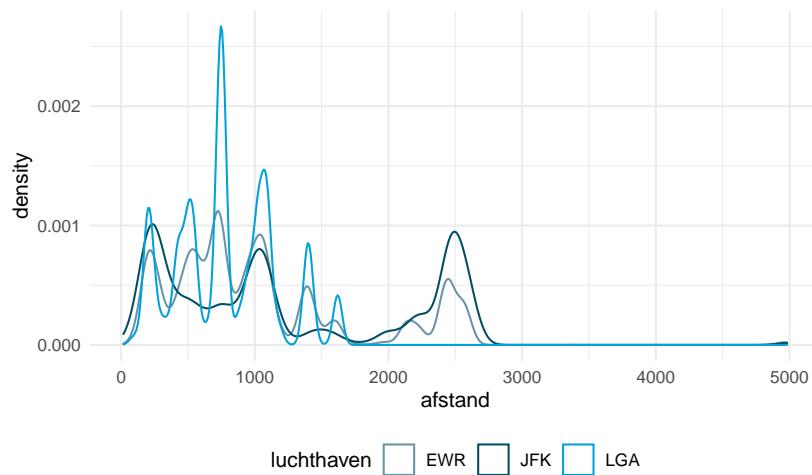


Figure 2.23: Bivariate density plot - overlappend



Indien de afhankelijke variabele een **categorische variabele** is:

- Kan je meerdere barplots op 1 grafiek visualiseren, met telkens de bars gegroepeerd per waarde van de onafhankelijke variabele.

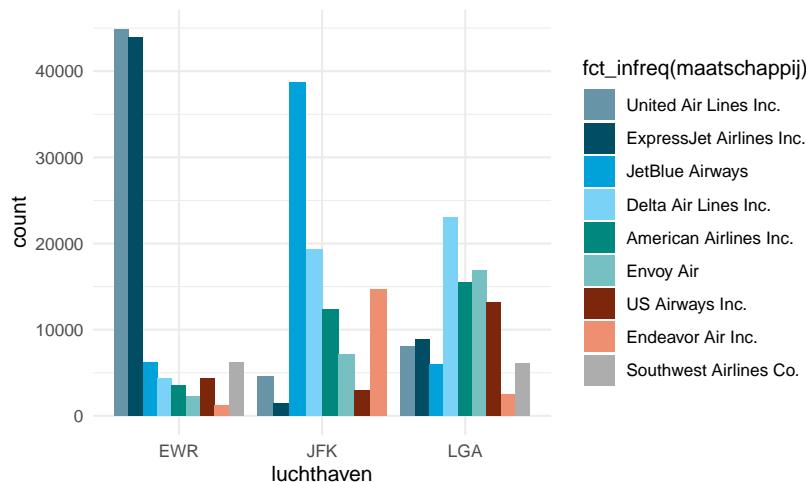


Figure 2.24: Bivariate barplot

- Kan je meerdere stacked barplots op 1 grafiek plaatsen, met telkens een volledige stack per waarde van de onafhankelijke variabele.

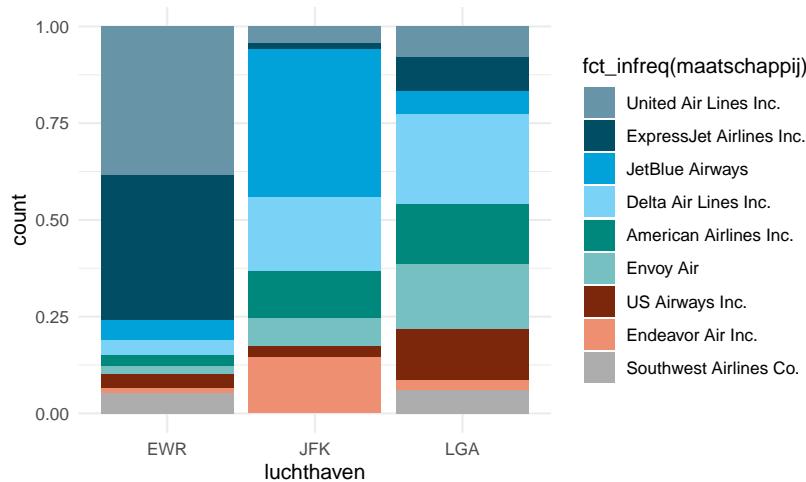
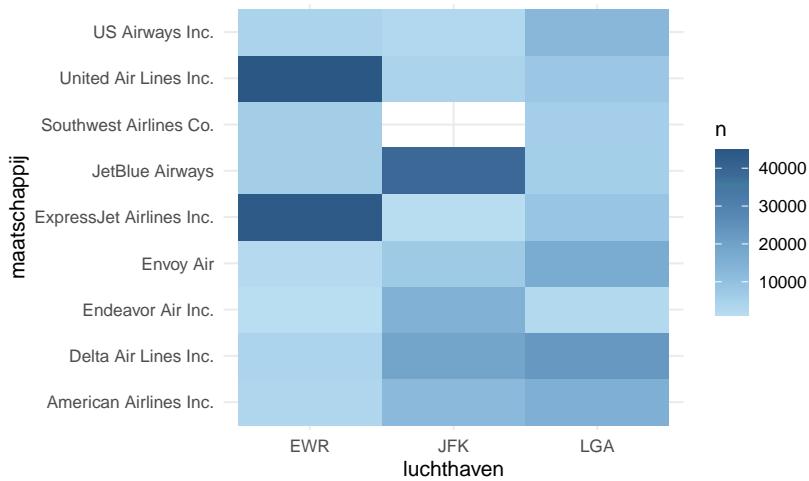


Figure 2.25: Bivariate stacked barplot

- Kan je een heatmap (of tile plot) gebruiken. Hierbij plaats je 2 categorische variabelen op de x-as en y-as, respectievelijk.

- Voor elke combinatie van waarden is er een tegel die je kan inkleuren volgens de frequentie van de combinatie.



- Je kan bijkomende ook de exacte waarde in elke tegel plotten.



Let op wanneer beide variabelen categorisch zijn, is het nog steeds van belang welke je beschouwd als afhankelijke en welke als onafhankelijke. Technisch kan je ze omdraaien, maar de betekenis van je visualisatie is niet dezelfde!

Andere mogelijkheden:

- treemap (Fig. 2.27)
- mosaic plot (Fig. 2.28)

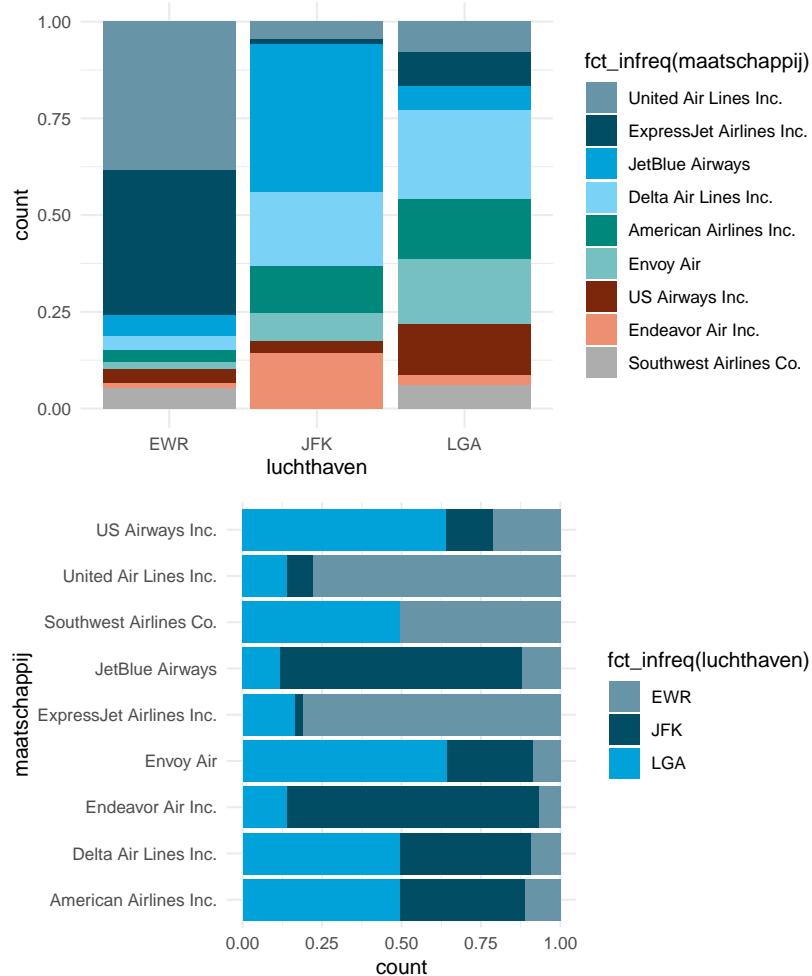


Figure 2.26: Twee verschillende stacked barcharts van luchthaven en maatschappij.



Figure 2.27: Treemap luchthaven en maatschappij.

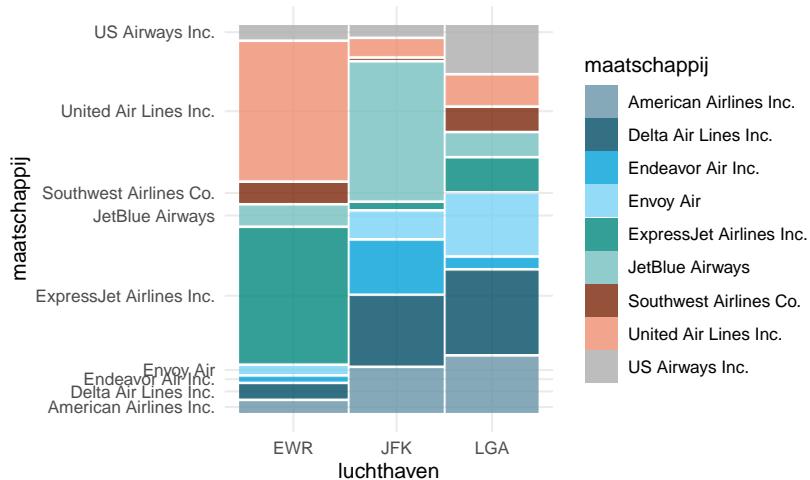


Figure 2.28: Mosaic plot luchthaven en maatschappij.

2.5.2 Situatie 2: De onafhankelijke variabele is continue

In dit geval kan je geen aparte plot per mogelijke waarde van de onafhankelijke variabele maken omdat er mogelijk oneindig veel waarden zijn.

Indien de afhankelijke variabele continu is, dan kan je een scatterplot maken.

- Iedere observatie is een punt in je grafiek, waarbij de x-waarde op de grafiek overeenkomt met de waarde van de onafhankelijke variabele en de y-waarde op de grafiek overeenkomt met de waarde van de afhankelijke variabele.

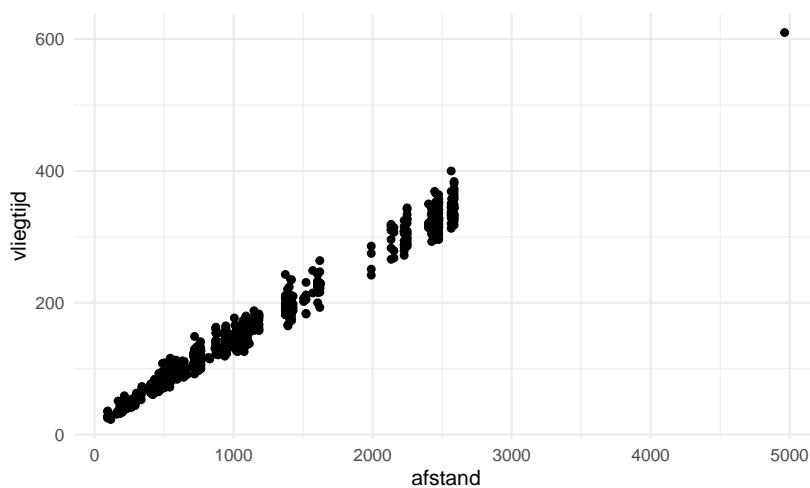


Figure 2.29: Scatterplot

- Om patronen beter te herkennen kan je een “trend-lijn” toevoegen.

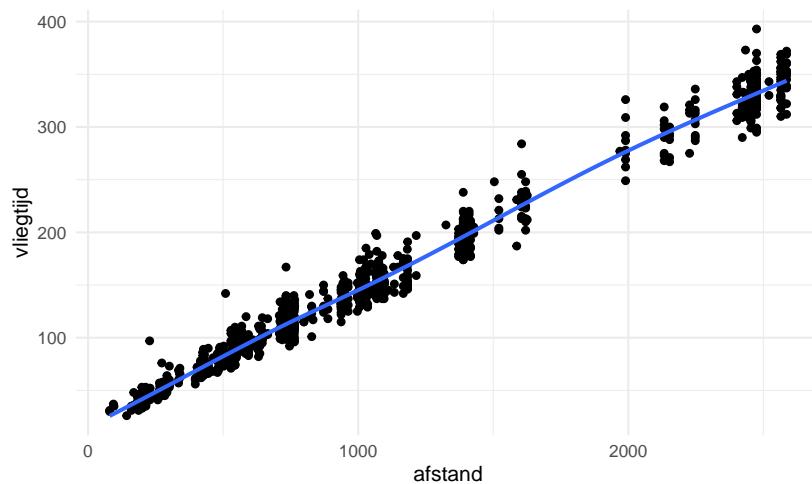


Figure 2.30: Scatterplot met trendlijn

- Bij scatterplots is er gevaar voor overplotting
- Mogelijke oplossingen
 - 2D histogram: verdeel veld op in vierkante bins en tel per bin hoeveel data punten er zijn
 - Hexplot: analoog, maar gebruik zeshoekige bins ipv vierkanten. Voordeel: punten binnen elke zeshoek liggen dichter bij het middelpunt van de bin.

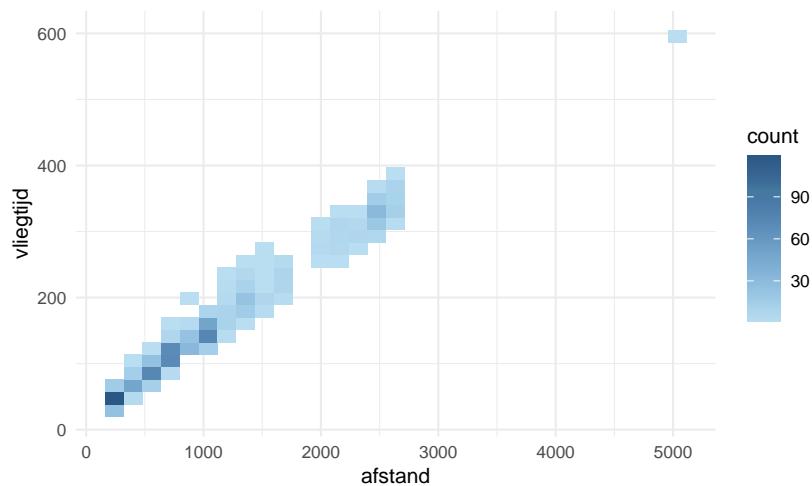


Figure 2.31: Scatterplot met trendlijn

Figure 2.32: Hexplot met trendlijn



Indien de afhankelijke variabele categorisch is, dan kan je niet rechtstreeks een betekenisvolle plot maken omdat er waarschijnlijk te weinig datapunten zijn voor iedere mogelijke waarde van de onafhankelijke variabele.

- Wat je dan best kan doen, is de onafhankelijke continue variabele categorisch maken door deze in te delen in bins/intervallen. En dan ben je terug in de situatie waarbij de onafhankelijke variabele categorisch is. We komen hierop terug in het hoofdstuk over Data Voorbereiding.

2.5.3 Situatie 3: De onafhankelijke variabele is tijd

- Tijd kunnen we zien als continue variabele
 - Bijgevolg zelfde grafieken mogelijk als wanneer onafhankelijke variabele continue is
 - * Tijd + continue afhankelijk -> scatterplot, 2D histograms, hex bins
 - * Tijd + categorisch afhankelijk -> probleem: tijd categoriseren (zie verder).
- Wanneer we één enkele variabele voorstellen doorheen de tijd is er per tijdseenheid maar 1 data punt. Hieronder wordt de gemiddelde vertrekvertraging per dag getoond.

In dat geval is het beter om in plaats van punten een lijngrafiek te gebruiken.

Indien je een beperkt aantal punten hebt (hieronder bijvoorbeeld één maand van de vluchtgegevens) kan je ervoor kiezen om zowel

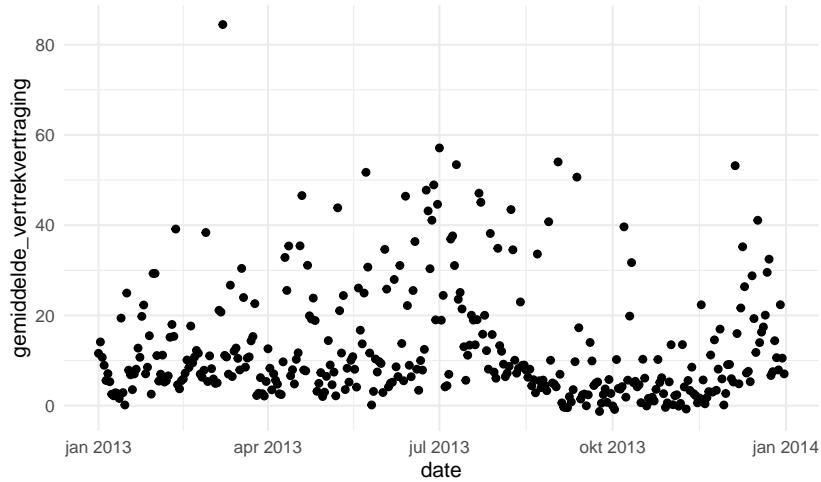


Figure 2.33: Puntenwolk met tijd op x-as

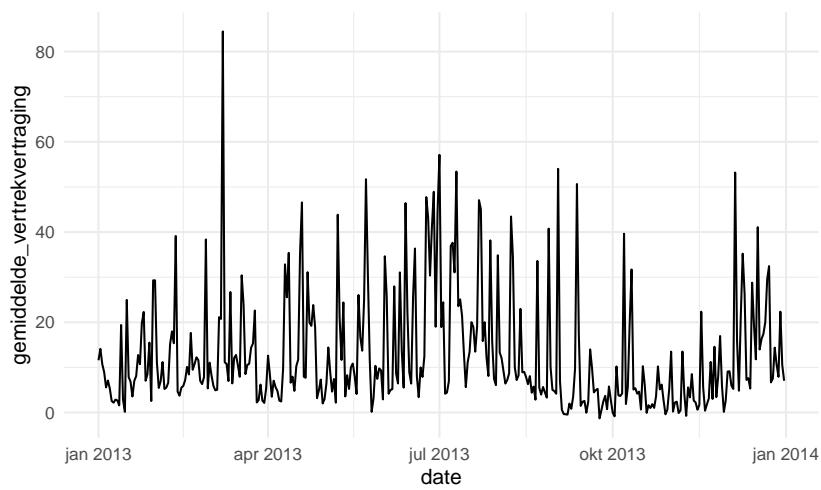


Figure 2.34: Lijngrafiek

punten als lijnen te tonen. Op die manier is het makkelijker individuele data punten af te lezen.

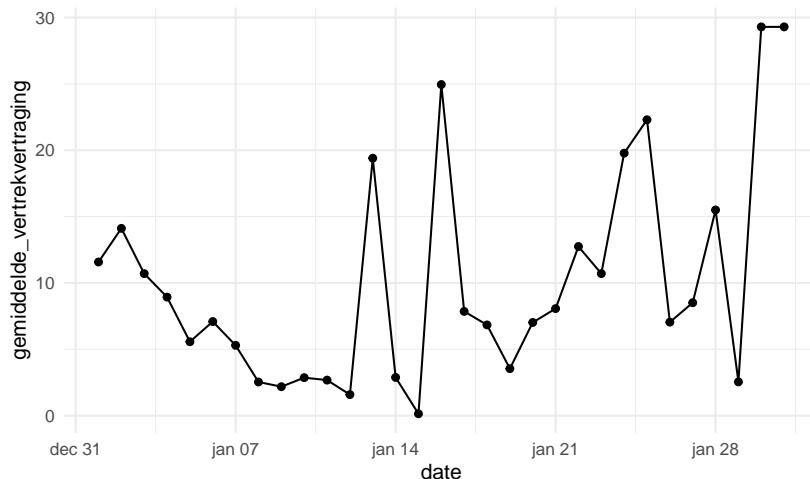


Figure 2.35: Lijn grafiek met punten

Indien we veel datapunten hebben, wat hier het geval is, kan een lijngrafiek zeer chaotisch worden. We kunnen daarom ervoor kiezen om onze tijd in te delen in categoriën. Bijvoorbeeld, in plaats van de dagelijkse gemiddelde vertrekvertraging, kunnen we de gemiddelde vertrekvertraging per maand berekenen en tonen.

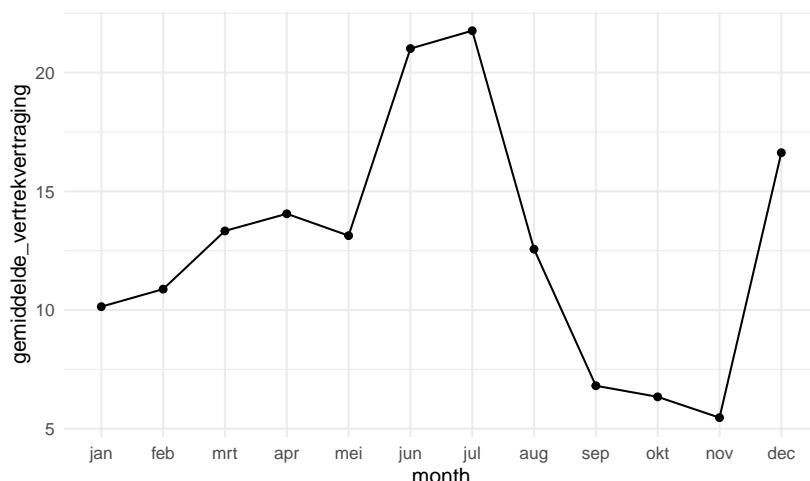


Figure 2.36: Lijngrafiek van gemiddelde vertrekvertraging per maand.

- Op dit moment verliezen we daardoor wel veel informatie. Maar we kunnen dit nu ook beschouwen als een visualisatie van een categorische variabele (maand) t.o.v. een continue. Waardoor we de technieken voor dit type bivariate visualizaties kunnen toepassen.

Bijvoorbeeld boxplots. We zien nu zowel de algemene trend als outliers. In februari was er bijvoorbeeld een dag waar de gemiddelde vertraging ver boven de normale trend lag.

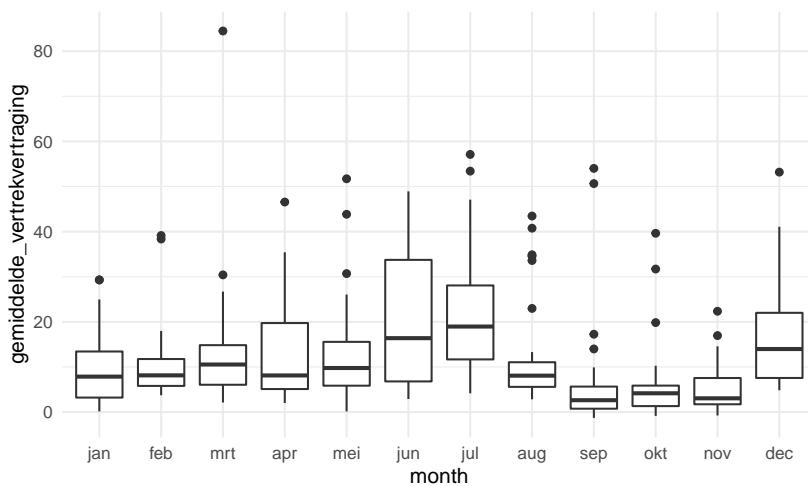


Figure 2.37: Boxplots van gemiddelde dagelijkske vertrekvertraging voor elke maand.

- Wanneer we de tijd gecategoriseerd hebben kunnen we ook categorische variabelen weergeven als afhankelijke. Bijvoorbeeld, zijn er verschillen in het aantal vluchten per maatschappij doorheen de tijd. We kunnen hier dezelfde types grafieken als voor bivariate cat+cat visualisaties gebruiken, bijvoorbeeld stacked barcharts.

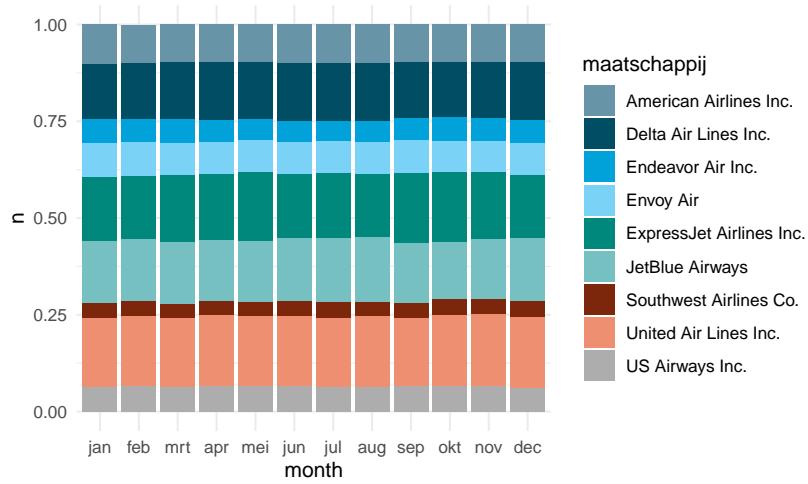


Figure 2.38: Verdeling van aantal vluchten over maatschappijen per maand.

- We kunnen categorizeren op maand, jaar, etc. Maar ook op tijdspecifiekere kenmerken, zoals bijvoorbeeld de dag van de week

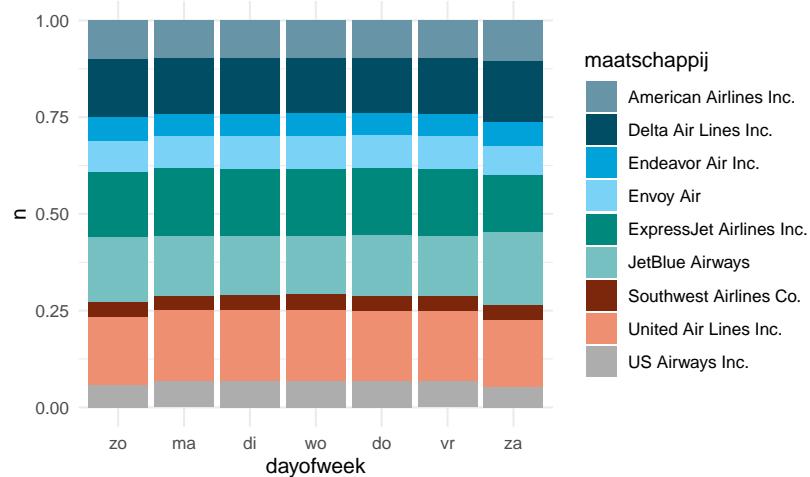


Figure 2.39: Verdeling van aantal vluchten over maatschappijen per dag van de week.

- Of het uur van de dag

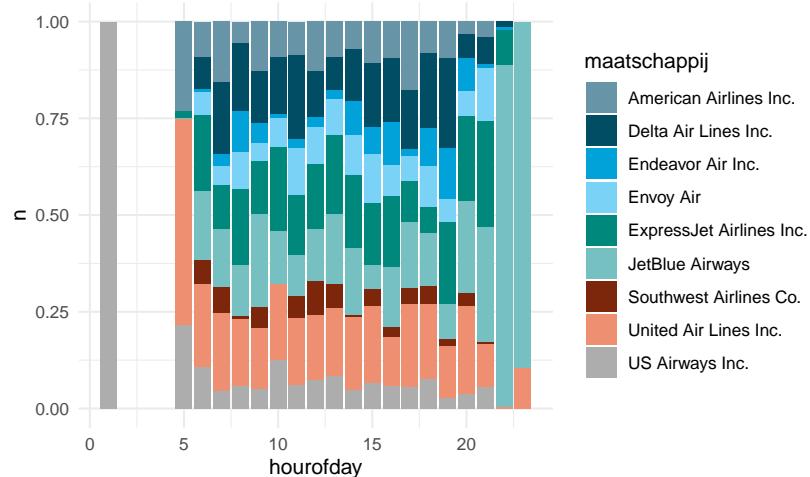


Figure 2.40: Verdeling van aantal vluchten over maatschappijen per vertrekuur.

2.6 Multivariate visualisaties (meer dan 2 variabelen)

- Datavisualisatie van patronen tussen meer dan 2 variabelen worden snel te complex om te interpreteren.
- Het basisprincipe is wel eenvoudig.
 - Je hebt typisch 1 afhankelijke variabele (Y) en een aantal onafhankelijke variabelen (A, B, ...).
 - Je visualizeert eerst Y en A (bivariaat)

- Je voegt dan de volgende variabelen (B, c, \dots) stap voor stap toe aan de grafiek.
 - * Door de bivariate grafiek te herhalen in verschillende facetten (een voor elke waarde van B).
 - * Door verschillende kleuren te gebruiken voor elke waarde van B
- Bij multivariate visualisaties zijn er afhankelijk van de data types oneindig veel mogelijke grafieken die je kan maken.
 - Het is vaak afhankelijk van de data welke grafiek het “best past”
 - Enkel wanneer de onafhankelijk variabele continu is zijn de keuzes beperkt en ben je vaak genoodzaakt om deze om te zetten naar categoriëen.

2.6.1 Voorbeeld: In welke mate hangt de vertrek vertraging af van de luchthaven en de afstand?

Stap 1. Vertraging vs. afstand

- Beide continue: scatterplot

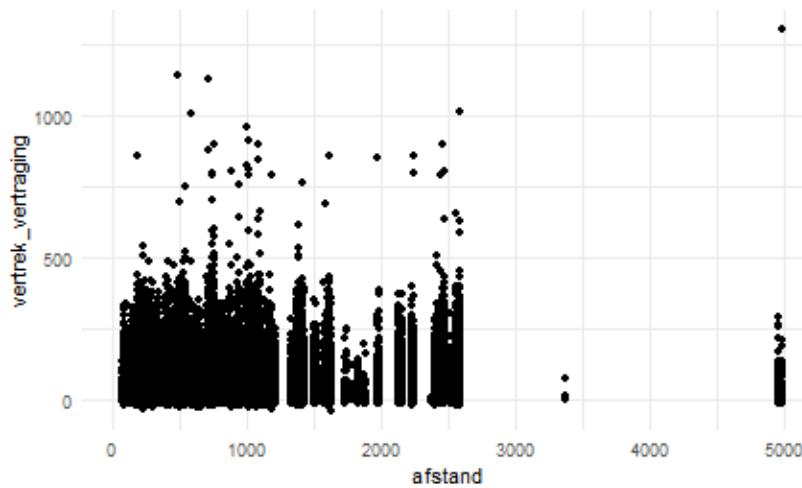


Figure 2.41: Vertrekvertraging vs afstand

Stap 2. Voeg invloed van luchthaven toe.

- Optie 1: gebruik kleur om de verschillende luchthavens te differenteren. Een trendlijn kan hier helpen.
- Geen geweldig resultaat in dit geval.

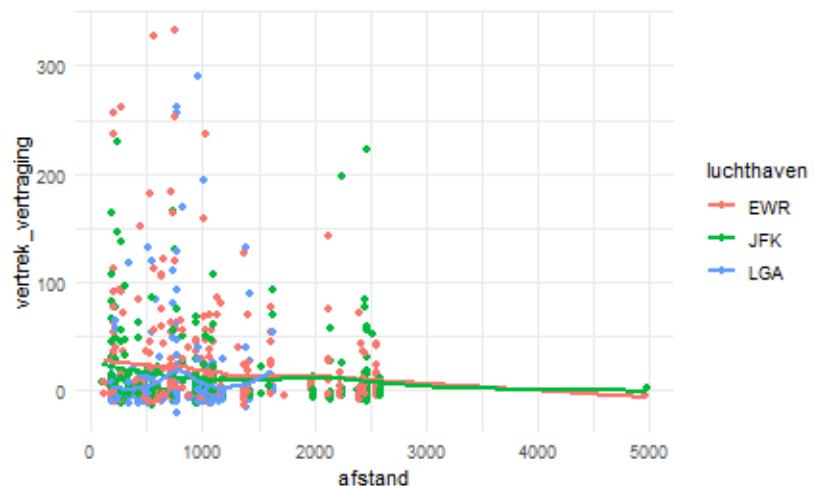


Figure 2.42: Vertrekvertraging vs afstand en luchthaven

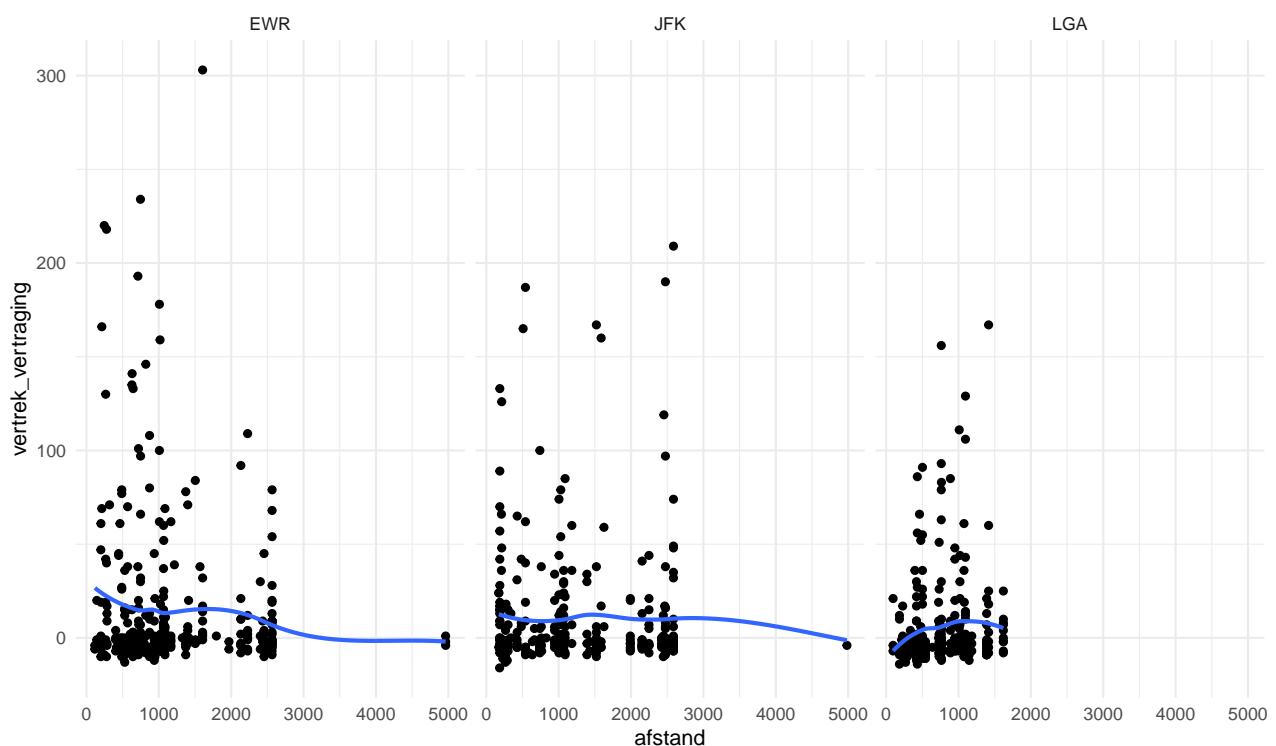


Figure 2.43: Vertrekvertraging vs afstand en luchthaven

- Optie 2: Gebruik facetten voor de verschillende luchthavens.
- Optie 3: Facets, maar gebruik hex bins

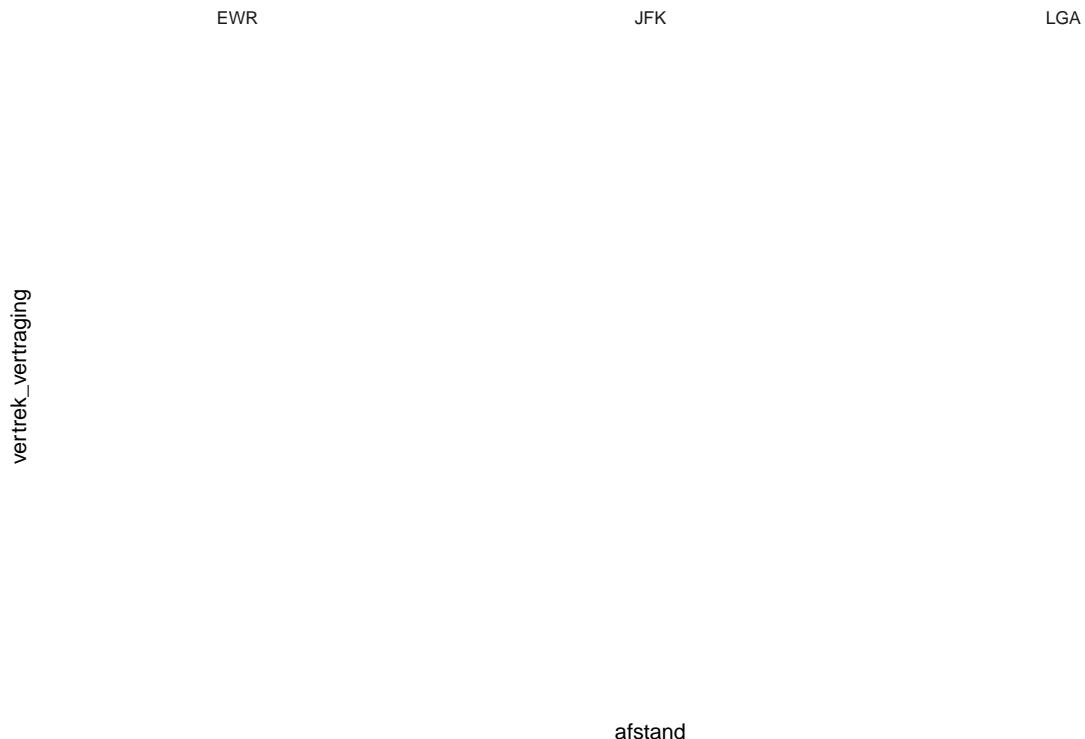


Figure 2.44: Vertrekvertraging vs afstand en luchthaven, hexbins

2.6.2 Voorbeeld: multivariaat tijd

Situatie 1: Variabelen hebben dezelfde eenheid.

Voorbeeld: vertrekvertraging en aankomstvertraging. Je kan lijngrafieken tekenen met meerdere lijnen op hetzelfde assenstelsel.

- Of je kan er voor kiezen elke lijn in een afzonderlijk paneel te tonen

Situatie 2: Variabelen hebben niet dezelfde eenheid

Voorbeeld: de gemiddelde levensverwachting en gdp per capita doorheen de tijd. In dit geval ben je genoodzaakt 2 panelen te gebruiken.

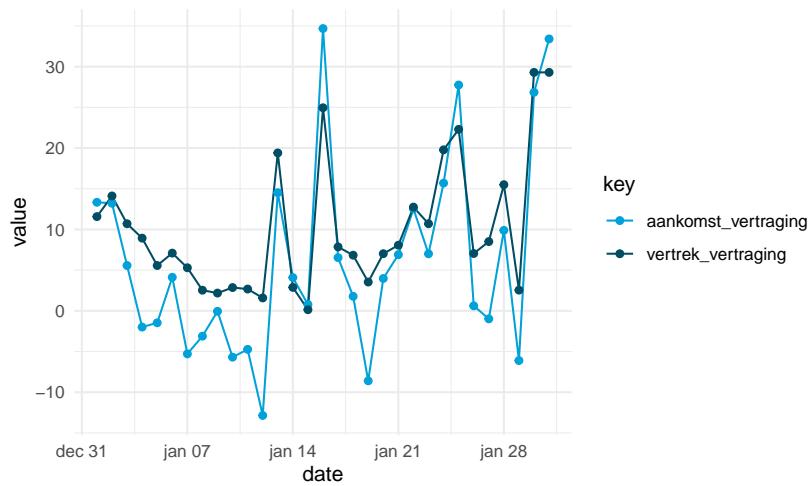


Figure 2.45: Evolutie van 2 variabelen over tijd in één grafiek (zelfde meeteenheid)

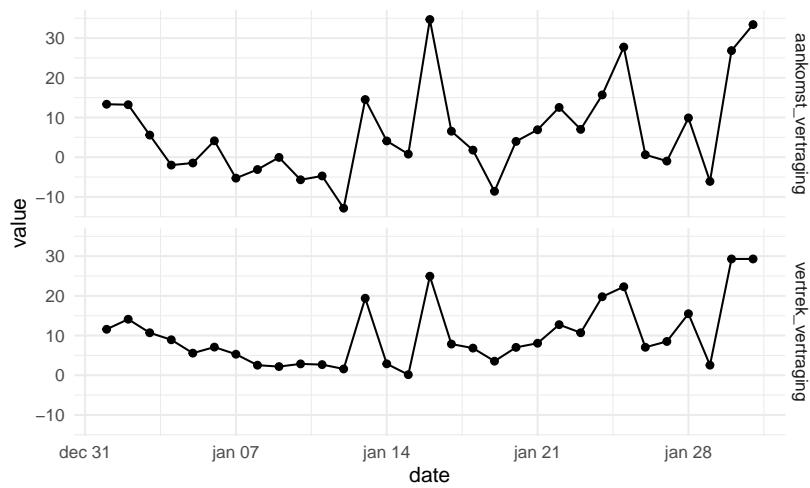


Figure 2.46: Evolutie van 2 variabelen over tijd in afzondelijke panels.

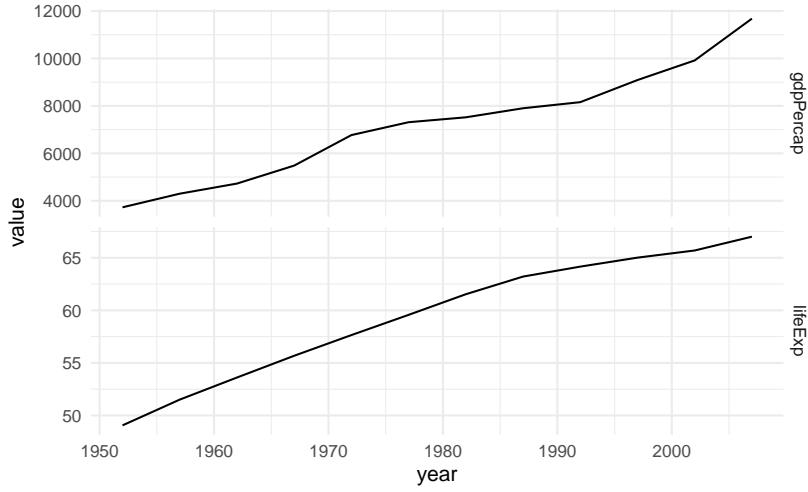


Figure 2.47: Evolutie van 2 variabelen met andere eenheden in afzonderlijke panels.

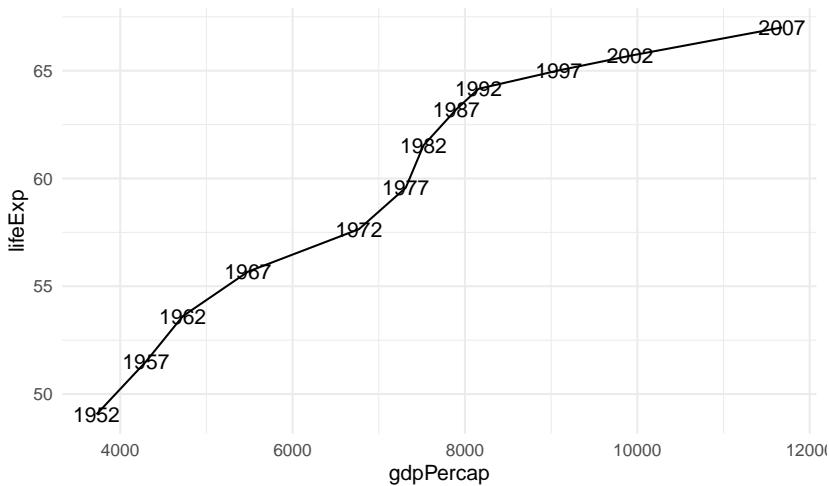


Figure 2.48: Evolutie van 2 variabelen (levensverwachting en inkomen per capita) aan de hand van connected scatterplot.

Optie 2: Maak een connected scatterplot. Toon een punt voor elke meting, waarbij x en y elk een variabele voorstellen. Verbindt dat elk punt in chronologische volgorde.

Variant, per continent:

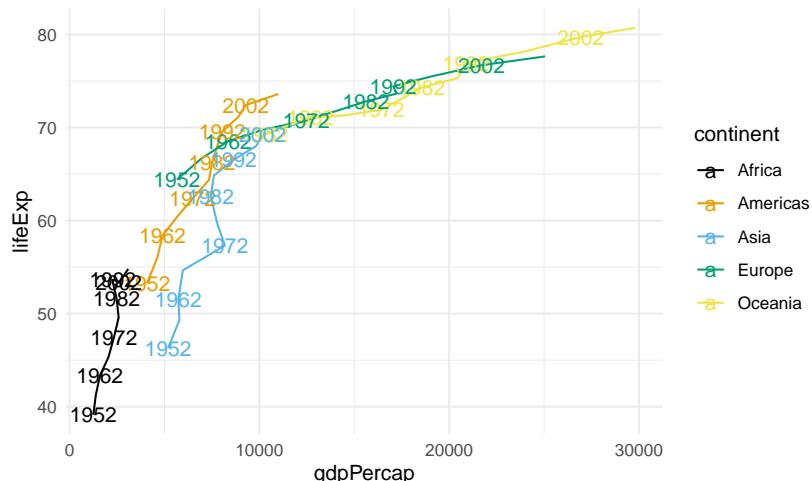


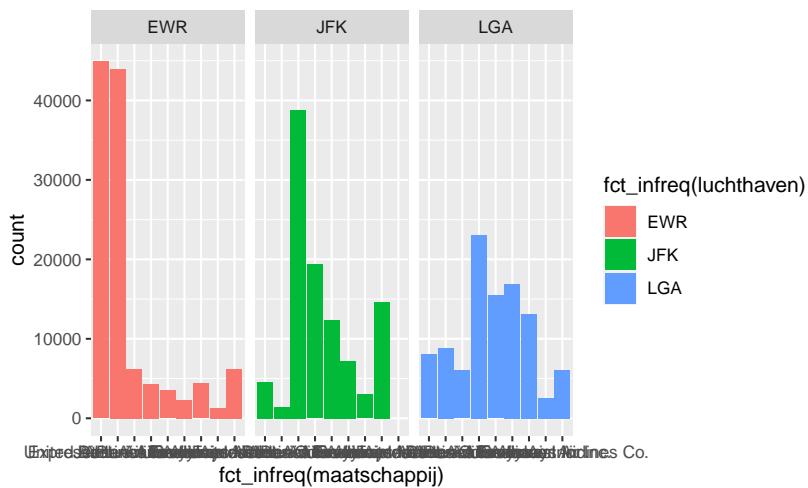
Figure 2.49: Evolutie van 2 variabelen aan de hand van connected scatterplot - verschillende groepen.

2.7 Visualisaties voor communicatie

Wanneer uiteindelijk beslist om een visualisatie te gebruiken om te communiceren, zorg ervoor dat

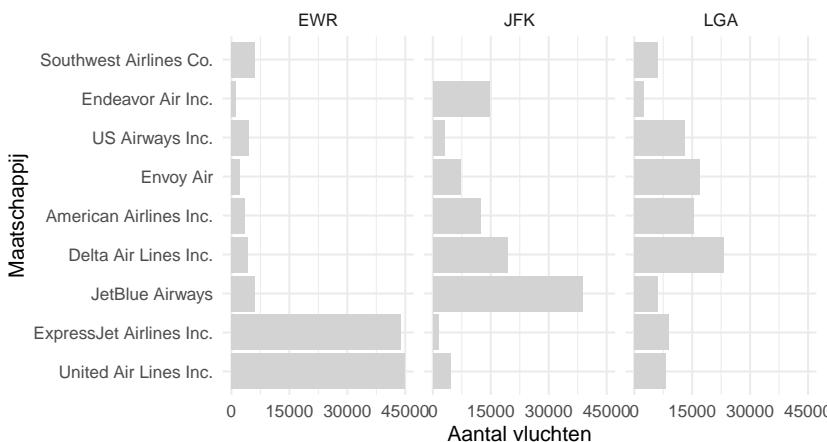
- de grafiek leesbaar is
- je kleur enkel gebruikt waar nodig.
- je correcte as-labels gebruikt
- je geen thema gebruikt dat te druk/overheersend is
- je een gepaste titel voorziet.

2.7.1 Voorbeeld: voor ~ goed voor exploratie



2.7.2 Voorbeeld: na ~ goed voor communicatie

Meest actieve maatschappijen per luchthaven.



Merk op: ver van alle grafieken getoond in dit hoofdstuk zijn goed voor communicatie zonder aanpassingen.

2.8 How charts lie

2.8.1 Causaliteit vs correlatie

- Van zodra er twee (of meer) variabelen zijn, gaan we op zoek naar patronen in relaties tussen de variabelen.
 - Het is belangrijk en essentieel te beseffen dat mensen een automatische reflex hebben om te denken in termen van oorzaak-gevolg als we kijken naar relaties tussen twee variabelen.

- Het is echter niet omdat er een duidelijke relatie bestaat tussen twee variabelen (correlatie), dat hier sprake is van een oorzaak-gevolg verband (causaliteit).
- Bijvoorbeeld: Indien in de zomer de verkoop van paraplu's sterk stijgt, dan zal de graanopbrengst in het najaar dalen. Dit betekent niet dat de verkoop van paraplu's een impact heeft op de graanopbrengst. Wat hier waarschijnlijk gebeurt, is dat door hevige regenvallen in de zomermaanden, de verkoop van paraplu's is toegenomen en de graanoogst tegenvalt.
 - * Soms is het intuïtief zeer onwaarschijnlijk dat de waargenomen correlatie causaliteit impliceert. Kijk hiervoor maar eens naar de voorbeelden op [http://www.tylervigen.com/
spurious-correlations](http://www.tylervigen.com/spurious-correlations)
 - * Wanneer het echter plausibel is dat de waargenomen correlatie causaliteit voorstelt, is het belangrijk dat we tegen onze natuurlijke reflex in gaan en niet in termen van oorzaak-gevolg denken.
 - * Het aantonen van causaliteit is nooit mogelijk met descriptieve en exploratieve data analyse!

2.9 Referenties

- Information is Beautiful
- Fundamentals of Data Visualization
- R Graph Gallery
- Data to viz
- Spurious correlations
- Misleading election map

3

[Tutorial] Data visualisatie

3.1 Voor je begint

Voordat je met deze zelfstudie begint, moet je eerst het pakket `ggplot2` installeren, als je dat nog niet gedaan hebt. Je kunt dit doen met de volgende regel code:

```
install.packages("ggplot2")
```

In ieder geval, moet je het pakket in je sessie laden.

```
library(ggplot2)
```

Je hebt ook twee datasets nodig, `movies` en `diamonds`. Beide worden als .RDS bestand bij deze tutorial geleverd.

```
movies <- readRDS("movies.RDS")
diamonds <- readRDS("diamonds.RDS")
```

3.2 Introductie

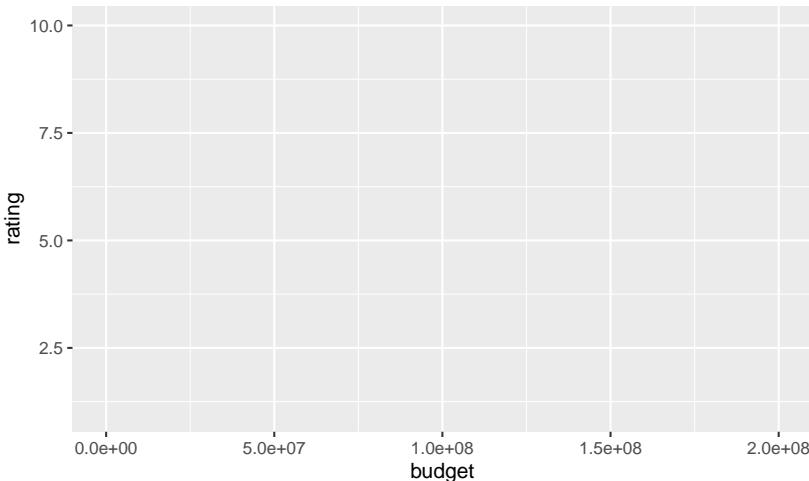
Het maken van een plot met `ggplot2` begint met de `ggplot()` functie. De `ggplot` functie heeft twee belangrijke argumenten

- **data:** dit definieert de dataset die voor de plot moet worden gebruikt. Dit moet een `data.frame` zijn.
- **mapping:** de mapping zal bepalen hoe de variabelen worden *mapped* op de esthetica¹ van de plot, zoals verderop zal worden uitgelegd. Deze mapping moet altijd worden gemaakt met de `aes()` functie .

We willen bijvoorbeeld een scatterplot maken van de films, waarbij we de x-as gebruiken voor hun budget en de y-as voor hun waardering. We roepen `ggplot` dan als volgt aan:

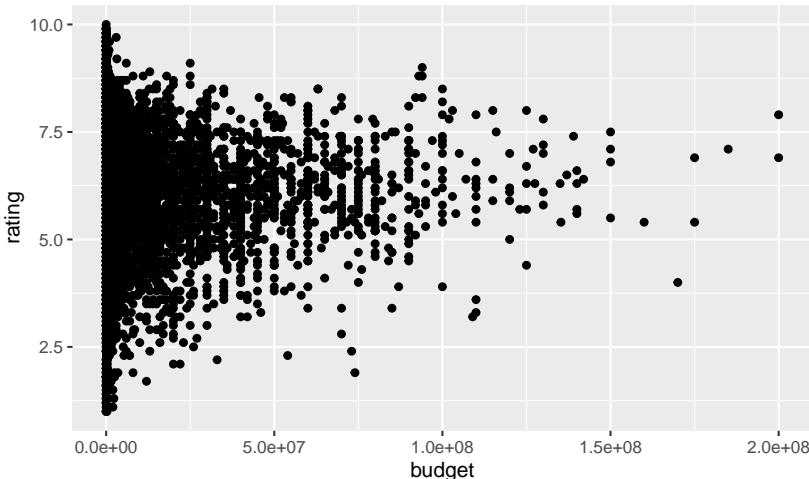
```
ggplot(data = movies, mapping = aes(x = budget, y = rating))
```

¹ De *aesthetics* van een `ggplot`-grafiek zijn de visuals die we in de grafiek zien: positie, kleur, vorm, grootte, linetype, enz.



Zoals je ziet, creëert deze regel code een plot met de assen zoals gedefinieerd. Er worden echter geen gegevens gevisualiseerd. De reden hiervoor is dat ggplot nog niet weet hoe we het willen visualiseren. We moeten wat genoemd wordt *een geometrische layer* toevoegen. Om een scatterplot te maken, die uit *punten* bestaat, voegen we `geom_point` aan de plot toe.

```
ggplot(data = movies, mapping = aes(x = budget, y = rating)) +
  geom_point()
```

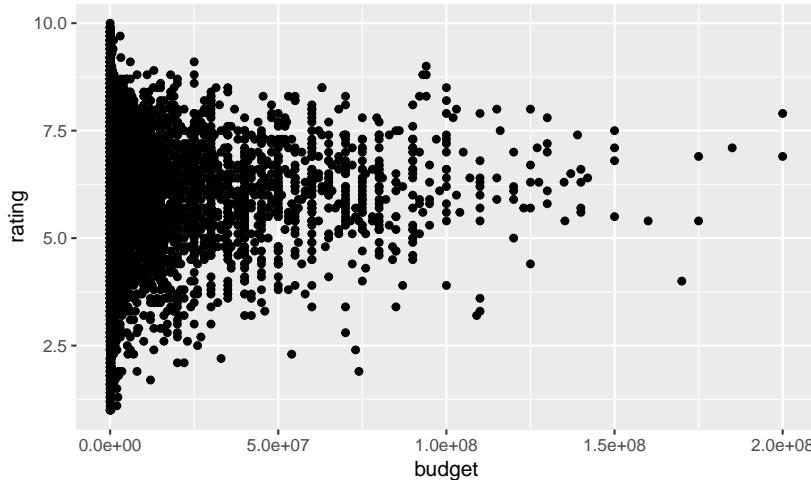


Dit lijkt er al meer op. Merk op dat de geometrische laag aan de plot is toegevoegd door gebruik te maken van het `+` symbool. Op deze manier kunnen meerdere lagen aan dezelfde plot worden toegevoegd, evenals titels, labels, en configuraties van de lay-out, zoals we verderop zullen zien.

Merk op dat we de mapping van de *aesthetics* ook in de geometrische laag zelf kunnen plaatsen. Dit maakt onze code voorlopig iets leesbaarder, omdat we de geometrische layer en de mapping ervan op

dezelfde regel plaatsen.

```
ggplot(data = movies) +
  geom_point(mapping = aes(x = budget, y = rating))
```



We hebben nu onze allereerste plot gemaakt! In de volgende secties zullen we leren hoe we verschillende geom-layers en *aesthetics* kunnen gebruiken en hoe we de lay-out van onze grafieken kunnen verbeteren.

3.3 Verschillende geometries

Naast `geom_point` bestaan er nog veel meer verschillende geometrieën om gegevens in ggplot te plotten. Je kunt ze bekijken door ‘`geom_`’ in het console in te typen en door de auto-complete-lijst te navigeren. Elk van de geom-layers komt met zijn eigen specifieke set van *aesthetics* die in kaart gebracht kan (en soms moet) worden. In deze tutorial zullen we ons vooral richten op de volgende geometrische lagen:

- `geom_point`
- `geom_histogram`
- `geom_boxplot`
- `geom_violin`
- `geom_bar`
- `geom_col`

3.3.1 *geom_point*

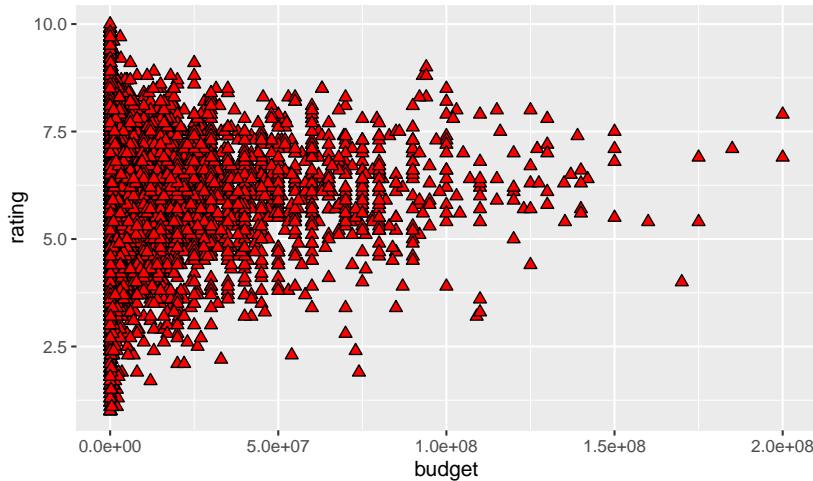
We hebben `geom_point` al gebruikt om onze eerste grafiek te maken, waarbij we twee variabelen in kaart brachten op de esthetica *x* en

y. Er zijn echter nog enkele andere aesthetics's die met deze laag kunnen worden ingesteld. Laten we ze eens in meer detail bekijken.

- **x:** dit bepaalt de positie van de punten langs de x-as
- **y:** hiermee bepaal je de positie van de punten langs de y-as
- **color:** hiermee bepaal je de kleur van de punten
- **shape:** hiermee bepaal je het type punten dat uitgezet moet worden²
- **fill:** hiermee bepaal je hoe de punten gevuld worden (voor vormen 21-25)
- **size:** hiermee bepaal je de grootte van de punten
- **stroke:** hiermee bepaal je de breedte van de rand
- **alpha:** hiermee bepaal je de mate van doorzichtigheid

De onderstaande grafiek bijvoorbeeld toont zwarte driehoeken, gevuld met rood, met een grootte van 2. Merk op dat de positie van de driehoeken precies dezelfde is als de positie van de punten in de vorige grafiek.

```
ggplot(data = movies) +
  geom_point(
    mapping = aes(x = budget, y = rating),
    shape = 24,
    fill = "red",
    color = "black",
    size = 2
  )
```



0	1	2	3	4
□	○	△	+	×
5	6	7	8	9
◇	▽	▣	*	◊
10	11	12	13	14
⊕	⊗	■	⊗	□
15	16	17	18	19
■	●	▲	◆	●
20	21	22	23	24
●	●	■	◆	▲
25				▼

Naast een normaal punt, kunnen veel verschillende vormen worden uitgezet in ggplot. Deze figuur toont de belangrijkste. De vormen kunnen worden gebruikt door de aesthetic 'shape' in te stellen op het bijbehorende nummer.

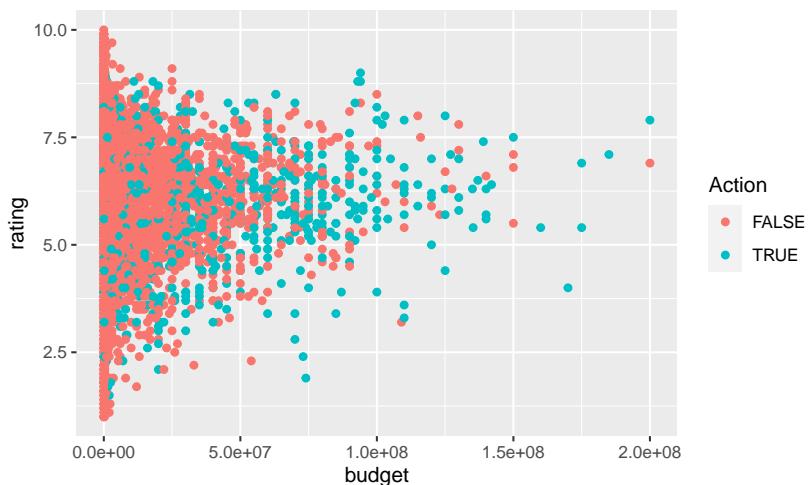
Maar wacht, er is hier iets belangrijks aan de hand! Terwijl de x- en y-aesthetics binnen de aes-mapping werden gedefinieerd, werden de andere aesthetic erbuiten gedefinieerd. Waarom is dat?

In feite kunnen aesthetics op twee verschillende manieren worden ingesteld:

1. Ze kunnen worden *mapped* naar een variabele in de dataset.
2. Ze kunnen worden ingesteld op één vaste waarde.

In ons voorbeeld worden x en y gekoppeld aan twee variabelen in de gegevens, nl. budget en rating, terwijl de andere aesthetics, vorm, fill, color en size, worden ingesteld op vaste waarden. Hoewel sommige aesthetics typisch altijd gemapt worden, zoals x- en y- posities, kunnen sommige andere zowel een vaste waarde als een gemapte variabele zijn. Bijvoorbeeld, wat gebeurt er als we de kleur van punten toewijzen aan een variabele, zeg de variabele Action.

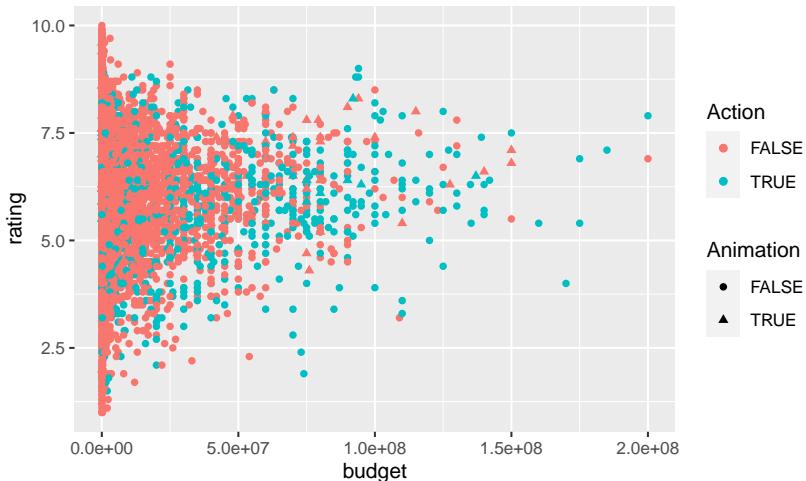
```
ggplot(data = movies) +
  geom_point(mapping = aes(
    x = budget,
    y = rating,
    color = Action
  ))
```



We zien dat de punten nu gekleurd zijn met betrekking tot de waarde in de variabele Action. Action films krijgen een groene kleur, terwijl andere films een rode kleur krijgen, wat we kunnen zien in de legende die verscheen.

Ook de andere vaste aesthetics kunnen worden gebruikt in een mapping. Het volgende voorbeeld gebruikt de variabele Animatie voor de shape.

```
ggplot(data = movies) +
  geom_point(mapping = aes(
    x = budget,
    y = rating,
    color = Action,
    shape = Animation
  ))
```



Geweldig! We begrijpen nu volledig de `geom_point` laag, en de werking van de `aesthetics`-mapping. Nu is het tijd om enkele andere geometrische layers te bekijken. We beginnen met histogrammen.

3.3.2 `geom_histogram`

De `geom_histogram` layer kan worden gebruikt om een histogram uit te zetten. Zoals je al zou moeten weten, geeft een histogram de verdeling van **een** continue variabele weer. Bijgevolg moet er alleen een x-aesthetic worden ingesteld, en geen y-aesthetic. De volledige lijst van aesthetic's is als volgt:

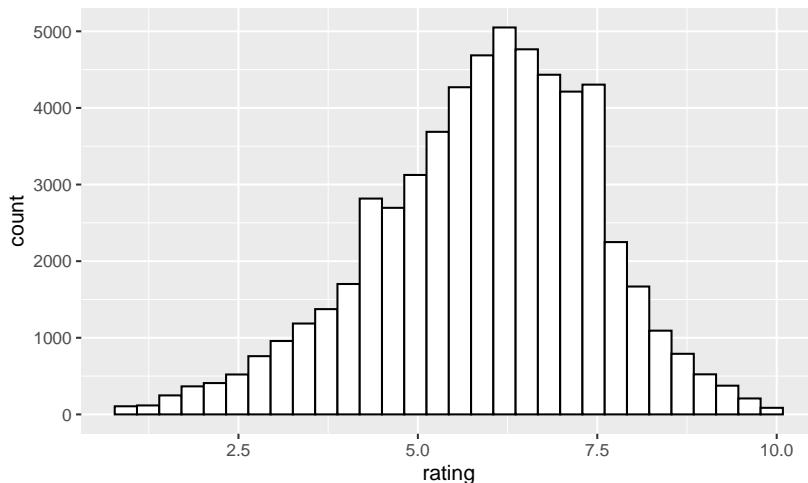
- **x**: dit bepaalt de variabele die gebruikt moet worden
- **color**: hiermee bepaal je de kleur van de randen
- **fill**: hiermee bepaal je met welke kleur het histogram gevuld wordt
- **size**: hiermee bepaal je de grootte van de rand
- **linetype**: hiermee bepaal je het type van de rand³
- **alpha**: hiermee bepaal je de mate van doorzichtigheid
- **weight**: hiermee bepaal je hoe de waarnemingen gewogen moeten worden. Standaard wordt elke waarneming als één gewogen.

Gebruik makend van onze kennis over het gebruik van aesthetics van voorheen, is het nu heel eenvoudig om een histogram te maken. Laten we een histogram maken voor de beoordeling van films. We geven het een zwarte rand met een witte vulling. Klaar om te proberen?

```
ggplot(movies) +
  geom_histogram(aes(rating), color = "black", fill = "white")
```

0. 'blank'	---
1. 'solid'	---
2. 'dashed'	---
3. 'dotted'	---
4. 'dotdash'	---
5. 'longdash'	---
3. 6. 'twodash'	---

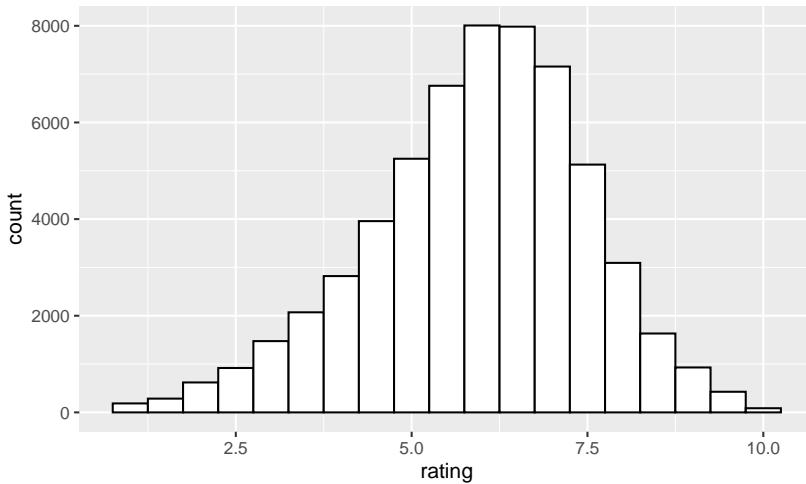
Naast een normale lijn, kunnen verschillende andere linetypes geplot worden in ggplot. De types kunnen worden gebruikt door de aesthetic 'linetype' in te stellen op het bijbehorende nummer, of op de naam van het type. Net al shape voor `geom_point`.



Merk op dat verschillende dingen werden weggelaten in deze twee lijnen van codes. In het bijzonder de argumentnaam *data* in *ggplot*, *mapping* in *geom_histogram* en *x* in *aes*. Aangezien we weten dat dit de eerste argumenten van deze functies zijn, kunnen we ze veilig weggelaten, zolang we de juiste volgorde van argumenten aanhouden. We kunnen echter *color* en *fill* niet weggelaten, omdat dit niet het tweede en derde argument van *geom_histogram* zijn. Speel in geval van twijfel op veilig en schrijf de juiste argumentnamen.

Dat is echter niet het enige dat hier opvallend is. Inderdaad, er verschijnt een waarschuwing: `stat_bin() using bins = 30. Pick better value with binwidth.` Deze waarschuwing herinnert ons aan het feit dat een standaard waarde voor het aantal bins is gekozen door *geom_histogram*, die waarschijnlijk niet geschikt is voor onze grafiek. We kunnen de binbreedte veranderen door deze als argument toe te voegen aan de aanroep *geom_histogram*.

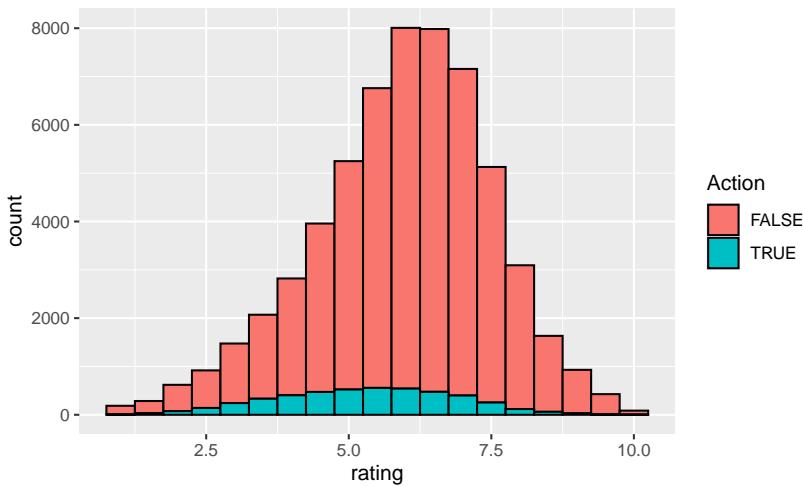
```
ggplot(movies) +
  geom_histogram(aes(rating), color = "black", fill = "white", binwidth = 0.5)
```



Vaak heeft de binbreedte een belangrijke invloed op hoe het verkregen histogram eruit ziet. Zorgvuldig configureren van dit argument door te experimenteren met verschillende waarden is daarom belangrijk.

In de laatste plot hebben we een vaste kleur gebruikt voor het vullen van de balken van het histogram. Maar zoals we onderussen al weten, kunnen we die ook toewijzen aan een variabele in de gegevens. Laten we de variabele *Action* nog een keer gebruiken.

```
ggplot(movies) +
  geom_histogram(aes(rating, fill = Action), color = "black", binwidth = 0.5)
```



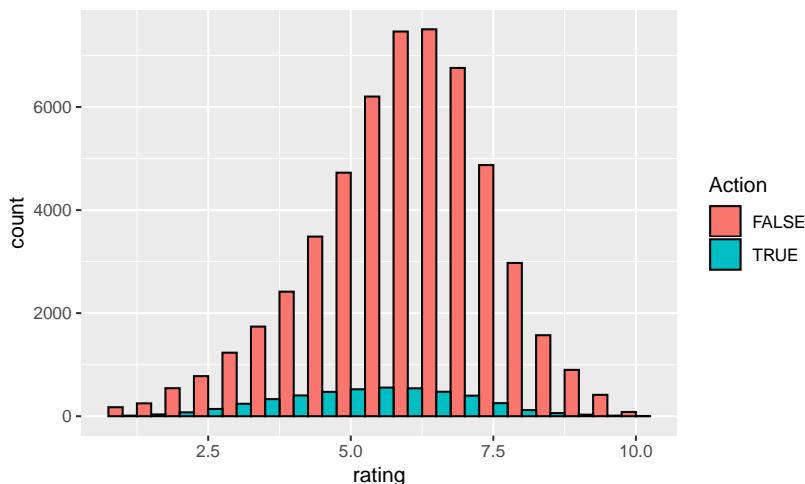
Merk op hoe we het fill-argument binnen de `aes` functie hebben geplaatst. Nu is elke staaf gevuld met twee kleuren: een deel voor actiefilms, en de rest voor andere films. Hoewel het niet erg duidelijk is in deze grafiek, lijkt het erop dat het centrum van het histogram voor actiefilms iets meer naar links ligt.

Merk op hoe de notatie verandert bij de overgang naar de `aes`-

mapping: variabelennamen worden altijd zonder aanhalingstekens gebruikt, terwijl vaste aesthetics (kleuren, vormen, linetypes) met aanhalingstekens worden gebruikt (behalve voor getallen). Het is belangrijk om dit niet door elkaar te halen! Nooit aanhalingstekens rond namen van variabelen!

Standaard zijn de histogrammen voor de verschillende *fills stacked*, d.w.z. boven elkaar geplaatst. We kunnen echter het *position* argument van `geom_histogram` gebruiken om de staven naast elkaar te plaatsen, of *dodged*.

```
ggplot(movies) +
  geom_histogram(aes(rating, fill = Action),
    color = "black", binwidth = 0.5, position = "dodge")
)
```



Met *position = "dodge"* worden de balken voor actiefilms en niet-actiefilms naast elkaar geplaatst, in plaats van boven elkaar. We kunnen teruggaan naar de oorspronkelijke grafiek door *position = "stack"* te gebruiken, of door dit argument weg te laten. Later zullen we zien hoe we beter met dergelijke zaken kunnen omgaan door gebruik te maken van rasters van verschillende plots, of zogenaamde *facets*.

Alles goed tot nu toe? Laten we eens kijken naar een andere manier om de verdeling van continue variabelen te visualiseren, namelijk de boxplot.

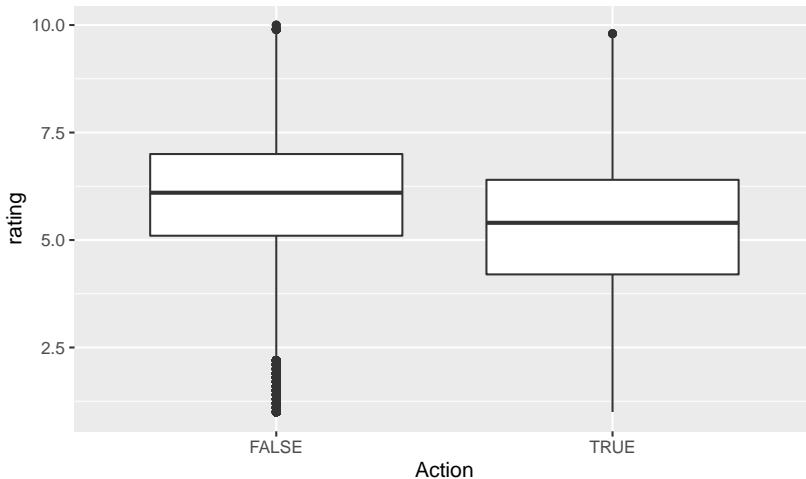
3.3.3 `geom_boxplot`

Een boxplot plaatst de waarden van de variabele op de y-as. Dus, als we een boxplot willen maken voor ratings, moeten we `aes(y = ratings)` gebruiken. Er is hier echter iets tricky aan de hand... De aesthetics voor `geom_boxplot` is de volgende

- **x:** dit definieert de variabele die voor de x-as wordt gebruikt
- **y:** dit definieert de variabele voor de y-as
- **color:** hiermee bepaal je de kleur van de randen
- **fill:** hiermee bepaal je hoe de boxplot wordt opgevuld
- **size:** hiermee bepaal je de grootte van de rand
- **linetype:** hiermee bepaal je het type van de rand
- **alpha:** Hiermee bepaal je de mate van doorzichtigheid

Dus, de boxplot heeft zowel een x-variabele als een y-variabele nodig? Dat lijkt op het eerste gezicht vreemd. De reden hierachter is dat in de filosofie van ggplot, altijd *iets* moet geplot worden op zowel de x- als de y-as. Hoewel alleen een x-variabele wordt gegeven aan een histogram, zal het frequenties berekenen om op de y-as te plotten. Bij boxplots gebeurt dat echter niet. Bijgevolg moet de x-as worden gebruikt om verschillende categorieën in kaart te brengen waarvan de verdeling vervolgens kan worden vergeleken. Zo kunnen we bijvoorbeeld de waardering voor actiefilms vergelijken met die voor andere films.

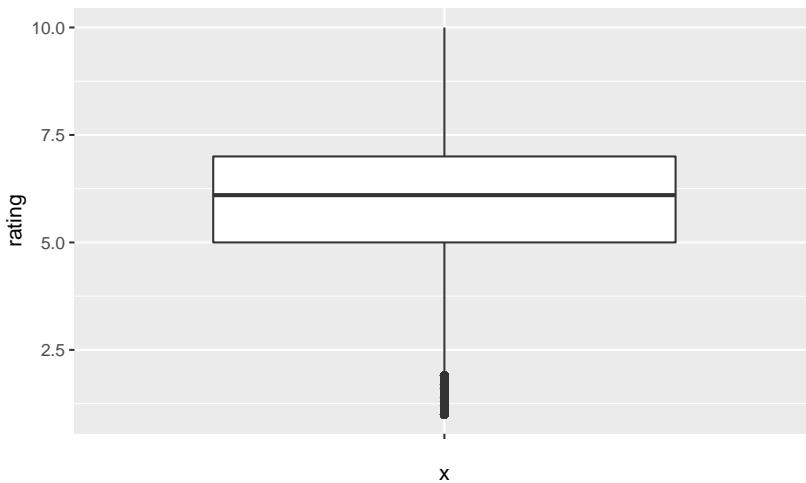
```
ggplot(movies) +
  geom_boxplot(aes(Action, rating))
```



Hier zien we dat, zoals we al vermoedden, actiefilms een lagere waardering hebben in vergelijking met andere films. We kunnen verder de kleur en de vulling van de boxplot veranderen zoals voorheen, alsook het linetype, de grootte van de rand, of de transparantie.

Maar wat als we gewoon een boxplot willen tekenen van de totale waardering, zonder een variabele te moeten specificeren voor de x-as? Een kleine workaround is hier nodig. Een mogelijkheid is om een *empty string* te gebruiken voor de x-as toewijzing.

```
ggplot(movies) +
  geom_boxplot(aes("", rating))
```



Merk op dat dit het label “x” creëert voor de x-as, waar we normaal de naam van de variabele zouden vinden die erop is weergegeven. Later zullen we zien hoe we dit label kunnen weglaten om onze grafiek een beetje mooier te maken.

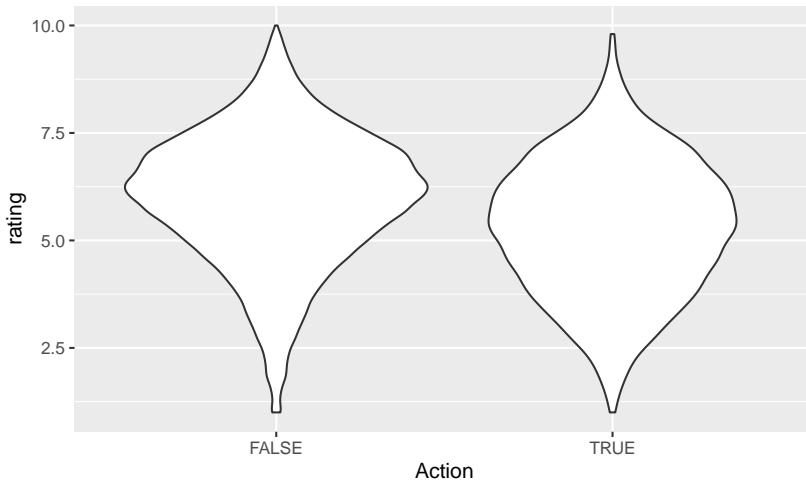
3.3.4 geom_violin

De violin-plot is vergelijkbaar met de boxplot, maar geeft in meer detail weer waar de massa van de waarden zich bevindt. De aesthetics is dezelfde als bij een boxplot.

- **x:** dit definieert de variabele die voor de x-as wordt gebruikt
- **y:** dit definieert de variabele voor de y-as
- **color:** hiermee bepaal je de kleur van de randen
- **fill:** hiermee bepaal je hoe de plot wordt opgevuld
- **size:** hiermee bepaal je de grootte van de rand
- **linetype:** hiermee bepaal je het type van de rand
- **alpha:** Hiermee bepaal je de mate van doorzichtigheid

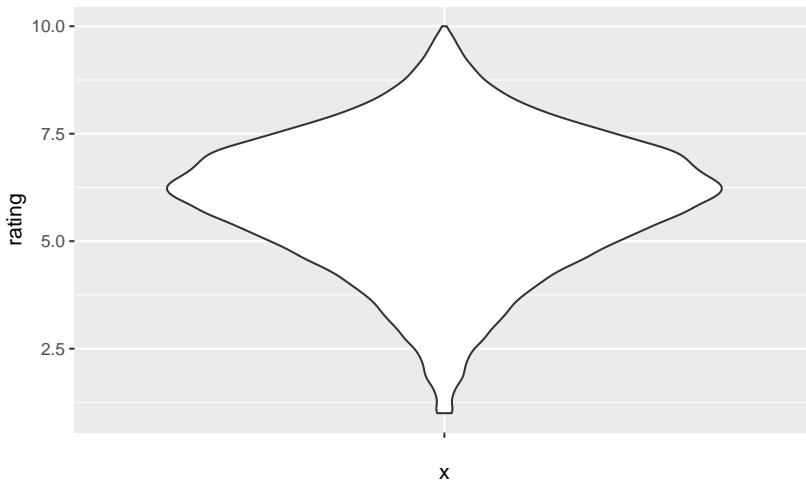
Laten we dezelfde grafieken maken, nu met de violin-plot.

```
ggplot(movies) +
  geom_violin(aes(Action, rating))
```



Het is nu waarschijnlijk wel duidelijk waar dit type grafiek zijn naam aan te danken heeft. Zoals je kunt zien, houden violin-plots het midden tussen boxplots en histogrammen. Omdat hun breedte genormaliseerd is, kunnen ze beter gebruikt worden voor vergelijkingen. Ook hier kunnen we dezelfde workaround gebruiken als we de algemene verdeling willen plotten.

```
ggplot(movies) +
  geom_violin(aes("", rating))
```



Tot nu toe hebben we drie verschillende manieren gezien om de verdeling van continue variabelen te analyseren. Nu gaan we kijken naar barplots, die kunnen worden gebruikt om categorische verdelingen weer te geven.

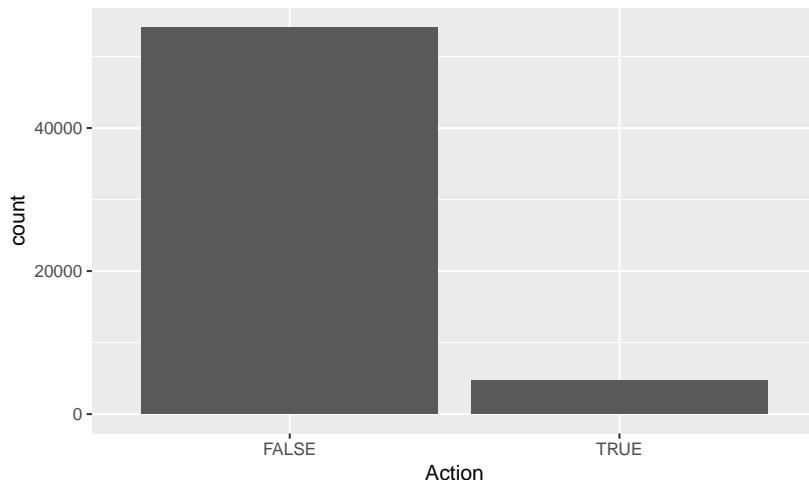
3.3.5 *geom_bar*

Net als een histogram, heeft een barplot alleen een x variabele nodig. Het verschil is dat deze variabele categorisch moet zijn, terwijl ze voor histogrammen continu moet zijn. De volledige lijst van aesthetics is de volgende:

- **x**: dit bepaalt de variabele die voor de x-as gebruikt wordt
- **color**: hiermee bepaal je de kleur van de randen
- **fill**: hiermee bepaal je hoe de balken gevuld worden
- **size**: hiermee bepaal je de grootte van de rand
- **linetype**: hiermee bepaal je het type van de rand
- **alpha**: hiermee bepaal je de mate van transparantie
- **weight**: hiermee bepaal je hoe de waarnemingen gewogen moeten worden. Standaard wordt elke waarneming als één gewogen.

We kunnen een eenvoudig staafdiagram maken dat laat zien hoeveel actiefilms er zijn, en hoeveel andere films, en wel als volgt.

```
ggplot(movies) +
  geom_bar(aes(Action))
```



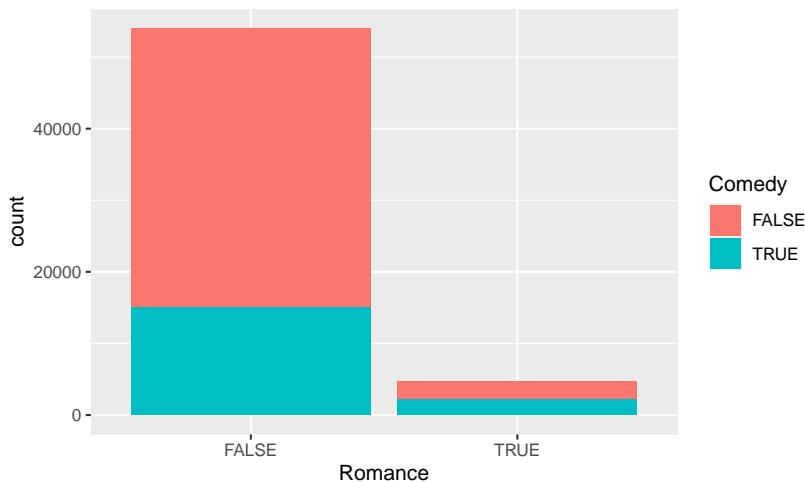
Verder kunnen we hier nog kleuren aan toevoegen, volgens het aantal Animatiefilms. We zien onmiddellijk dat er bijna geen actiefilms zijn die ook animatiefilms zijn.

```
ggplot(movies) +
  geom_bar(aes(Action, fill = Animation))
```



We kunnen hetzelfde doen voor Romantische en Komedie films.

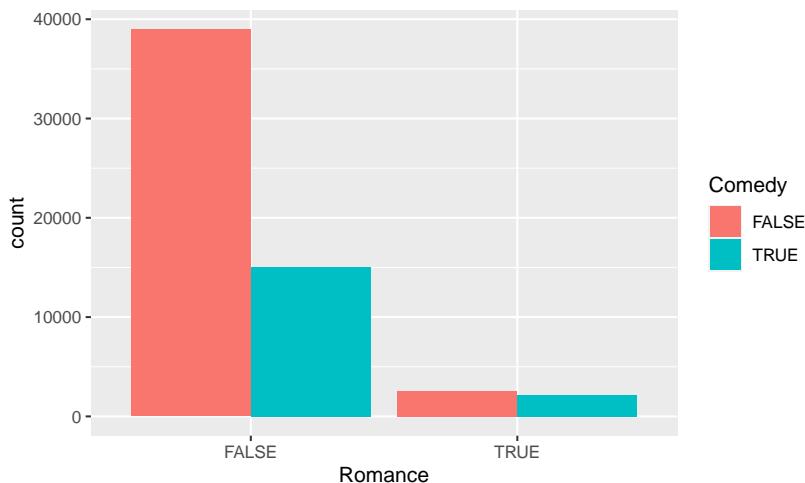
```
ggplot(movies) +
  geom_bar(aes(Romance, fill = Comedy))
```



Daarentegen is hier te zien dat ongeveer de helft van de romantische films ook komedies zijn, wat meer is in vergelijking met niet-romantische films.

Vergeet niet dat we in het geval van histogrammen de position konden veranderen in "dodge", waardoor de balken naast elkaar kwamen te staan. Hetzelfde kan hier worden gedaan.

```
ggplot(movies) +
  geom_bar(aes(Romance, fill = Comedy), position = "dodge")
```



Een derde mogelijkheid die voor de position beschikbaar is, is de staven te verlengen zodat zij dezelfde hoogte hebben. Het resultaat is dat we de verdeling van de waarden als een deel van een geheel zullen waarnemen. In plaats van de absolute frequentie zullen de labels op de y-as nu de procentpunten weergeven.

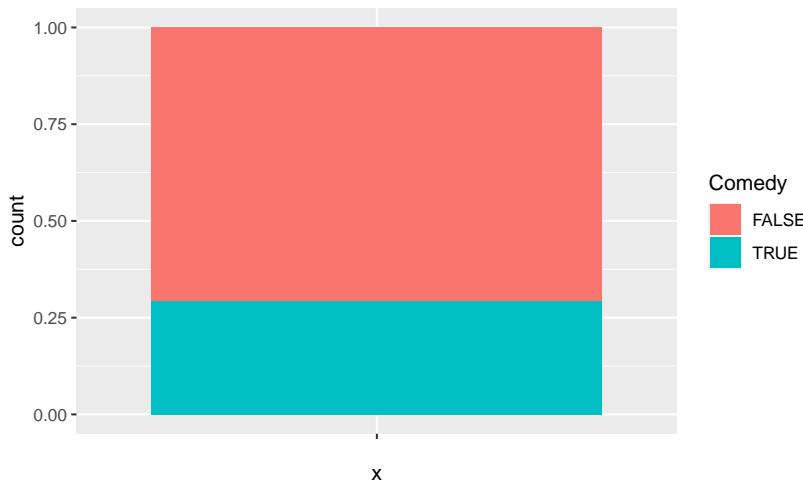
```
ggplot(movies) +
  geom_bar(aes(Romance, fill = Comedy), position = "fill")
```



Tenslotte, als we dit willen doen om de verdeling van één variabele te tonen, kunnen we dezelfde workaround gebruiken als voorheen en de x-aesthetic op "" zetten. De plot hieronder zal het deel van alle films tonen die komedies zijn.⁴

```
ggplot(movies) +
  geom_bar(aes("", fill = Comedy), position = "fill")
```

⁴ Merk op dat het in dergelijke gevallen volkomen logisch is de plot minder breed te maken, of hem 90 graden om te draaien en minder hoog te maken. We komen hier later op terug.



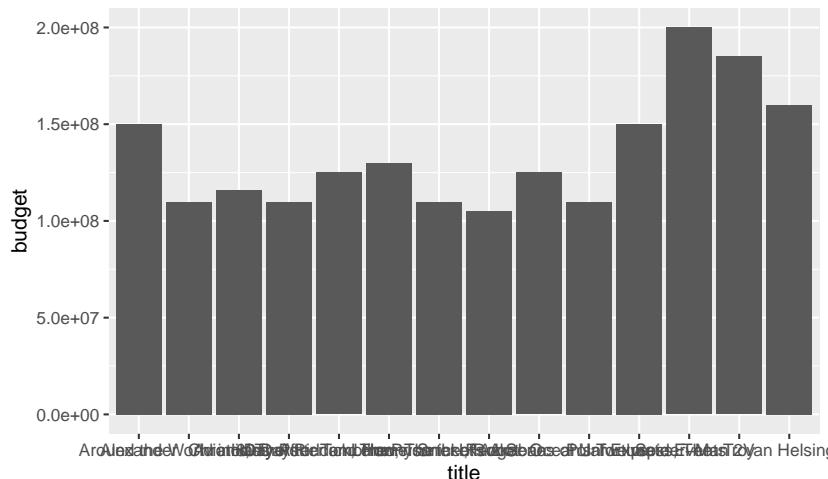
3.3.6 *geom_col*

Wanneer we *geom_bar* gebruiken, wordt de hoogte van de balken berekend aan de hand van de frequentie van de categorische variabele. Soms willen we echter een staafdiagram plotten met waarden die al in de data zitten, of waarden die we zelf hebben berekend. Bijvoorbeeld, wat als we een staafdiagram willen met het budget van een reeks films? In zo'n geval kunnen we *geom_col* gebruiken. "col" geeft aan dat we een kolom in de gegevens willen gebruiken om de hoogte van de balken in te stellen. De aesthetics is hetzelfde voor *geom_bar*, alleen moeten we nu een variabele specificeren voor de y-as uiteraard.

- **x:** dit bepaalt de variabele die voor de x-as gebruikt wordt
- **x:** dit bepaalt de variabele die voor de y-as gebruikt wordt
- **color:** hiermee bepaal je de kleur van de randen
- **fill:** hiermee bepaal je hoe de balken gevuld worden
- **size:** hiermee bepaal je de grootte van de rand
- **linetype:** hiermee bepaal je het type van de rand
- **alpha:** hiermee bepaal je de mate van transparantie
- **weight:** hiermee bepaal je hoe de waarnemingen gewogen moeten worden. Standaard wordt elke waarneming als één gewogen.

Laten we een staafdiagram maken van het budget van alle films uit 2004 waarvan het budget hoger was dan 100 miljoen.

```
filter(movies, year == 2004, budget > 100000000) %>%
  ggplot() +
  geom_col(aes(title, budget))
```



Zie je iets vreemds in de code? Maak je geen zorgen als je de eerste regel niet begrijpt. Al wat je moet weten is dat we films uit 2004 hebben gefilterd met een budget hoger dan 100 miljoen. Het vreemde uitzienende `%>%` symbol zal ervoor zorgen dat deze gegevens doorgegeven worden aan ggplot. We zullen hier in een andere sessie op terugkomen.⁵

We hebben nu filmtitels uitgezet op de x-as en budget op de y-as. Geweldig! Of toch niet? De waarden op de x-as zijn wat onoverzichtelijk en onleesbaar. Het is nu tijd om aandacht te besteden aan de layout van onze plots!

⁵ Als je dit zelf wilt proberen, zorg er dan voor dat het pakket dplyr is geïnstalleerd en geladen voordat je de filter gebruikt.

3.3.7 Other geometries

Tot dusver hebben we de belangrijkste geom-layers gebruikt om eenvoudige visualisaties te maken: scatterplots, histogrammen, boxplots, violinplots en barplots. We hebben echter slechts het topje van de ijsberg besproken, want er bestaan nog veel meer types, sommige eenvoudig en sommige meer geavanceerd. Een overzicht van alle geoms en hun toepassingen kan gevonden worden in de [ggplot Cheat Sheet](#), waarvan hier een uittreksel wordt getoond. Wees niet bang om iets uit te proberen!

Geoms - Use a geom to represent data points, use the geom's aesthetic properties to represent variables. Each function returns a layer.			
Graphical Primitives		Two Variables	
a <- ggplot(seals, aes(x = long, y = lat)) b <- ggplot(economics, aes(date, unemploy))		Continuous X, Continuous Y h <- ggplot(mpg, aes(cty, hwy)) A e + geom_label(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE) x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, just B e + geom_jitter(height = 2, width = 2) x, y, alpha, color, fill, shape, size C e + geom_point() x, y, alpha, color, fill, shape, size, stroke D e + geom_quantile() x, y, alpha, color, group, linetype, size, weight E e + geom_rug(sides = "bt") x, y, alpha, color, linetype, size F e + geom_smooth(method = lm) x, y, alpha, color, fill, group, linetype, size, weight G e + geom_text(aes(label = cty), nudge_x = 1, nudge_y = 1, check_overlap = TRUE) x, y, label, alpha, angle, color, family, fontface, hjust, lineheight, size, just	Continuous Bivariate Distribution h <- ggplot(diamonds, aes(carat, price)) H h + geom_bin2d(binwidth = c(0.25, 500)) x, y, alpha, color, fill, linetype, size, weight I h + geom_hex() x, y, alpha, color, fill, size
a + geom_curve(aes(yend = lat + delta_lat, xend = long + delta_long, curvature = 2)) x, y, alpha, angle, color, curvature, size b + geom_path(lineend = "butt", linejoin = "round", linemitre = 1) x, y, alpha, color, group, linetype, size b + geom_polygon(aes(group = group)) x, y, alpha, color, fill, group, linetype, size a + geom_rect(aes(min = long, ymin = lat, xmax = long + delta_long, ymax = lat + delta_lat)) xmax, xmin, ymax, ymin, alpha, color, fill, linetype, size b + geom_ribbon(aes(min = unemploy - 900, ymax = unemploy + 900)) x, y, alpha, color, fill, group, linetype, size a + geom_segment(aes(yend = lat + delta_lat, xend = long + delta_long)) x, y, end, alpha, color, linetype, size a + geom_spoke(aes(yend = lat + delta_lat, xend = long + delta_long)) x, y, angle, radius, alpha, color, linetype, size	 	Discrete X, Continuous Y f <- ggplot(mpg, aes(class, hwy)) A f + geom_bar(stat = "identity") x, y, alpha, color, fill, linetype, size, weight B f + geom_boxplot() x, y, lower, middle, upper, ymax, ymin, alpha, color, fill, group, linetype, shape, size, weight C f + geom_dotplot(binaxis = "y", stackdir = "center") x, y, alpha, color, fill, group D f + geom_violin(scale = "area") x, y, alpha, color, fill, group, linetype, size, weight	Continuous Function i <- ggplot(economics, aes(date, unemploy)) I i + geom_area() x, y, alpha, color, fill, linetype, size J i + geom_line() x, y, alpha, color, group, linetype, size K i + geom_step(direction = "hv") x, y, alpha, color, group, linetype, size
c <- ggplot(mpg, aes(hwy)) c + geom_area(stat = "bin") x, y, alpha, color, fill, linetype, size a + geom_area(aes(y..density..), stat = "bin") x, y, alpha, color, fill, group, linetype, size, weight c + geom_density(kernel = "gaussian") x, y, alpha, color, fill c + geom_dotplot() x, y, alpha, color, fill c + geom_freqpoly() x, y, alpha, color, group, linetype, size a + geom_freqpoly(aes(y..density..)) c + geom_histogram(binwidth = 5) x, y, alpha, color, fill, linetype, size, weight a + geom_histogram(aes(y..density..)) d + geom_bar() x, alpha, color, fill, linetype, size, weight	 	Discrete X, Discrete Y g <- ggplot(diamonds, aes(cut, color)) A g + geom_count() x, y, alpha, color, fill, shape, size, stroke	Visualizing error df <- data.frame(grp = c("A", "B"), fit = 4:5, se = 1:2) j <- ggplot(df, aes(grp, fit, ymin = fit - se, ymax = fit + se)) J j + geom_crossbar(fatten = 2) x, y, max, ymin, alpha, color, fill, group, linetype, size K j + geom_errorbar() x, ymax, ymin, alpha, color, group, linetype, size, width (also geom_errorbarh()) L j + geom_linearrange() x, ymin, ymax, alpha, color, group, linetype, size M j + geom_pointrange() x, y, min, max, alpha, color, fill, group, linetype, shape, size
seals\$z <- with(seals, sqrt(delta_long^2 + delta_lat^2)) l <- ggplot(seals, aes(long, lat))		Maps data <- data.frame(murder, state = tolower(rownames(USAarrests))) map <- map_data("state") k <- ggplot(data, aes(fill = murder)) k + geom_map(aes(map_id = state), map = map) + expand_limits(x = map\$long, y = map\$lat) map_id, alpha, color, fill, linetype, size	Three Variables l + geom_raster(aes(fill = z), hjust = 0.5, vjust = 0.5, interpolate = FALSE) x, y, alpha, fill l + geom_tile(aes(fill = z)) x, y, alpha, color, fill, linetype, size, width

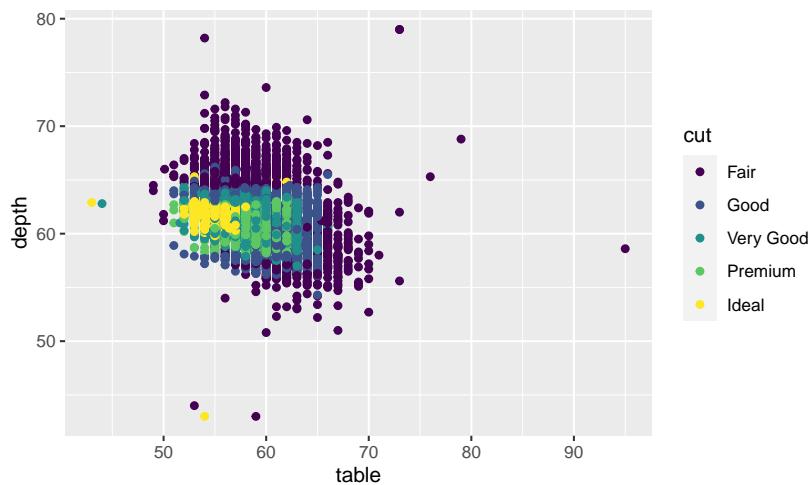
3.4 Layout van onze grafieken verbeteren

Tot nu toe hebben we vooral gekeken naar verschillende soorten plots en hoe we die op onze gegevens kunnen plotten. In deze sectie zullen we ons concentreren op de presentatie van de plot, bv. titels, kleuren, assen, enz. De in dit deel geïntroduceerde concepten kunnen voor elk type plot worden toegepast, ongeacht welk geometrisch object wordt gebruikt.

In dit deel zal de dataset "diamanten" worden gebruikt. De onderstaande plot zal als uitgangspunt worden gebruikt.⁶

```
ggplot(diamonds) +
  geom_point(aes(table, depth, color = cut))
```

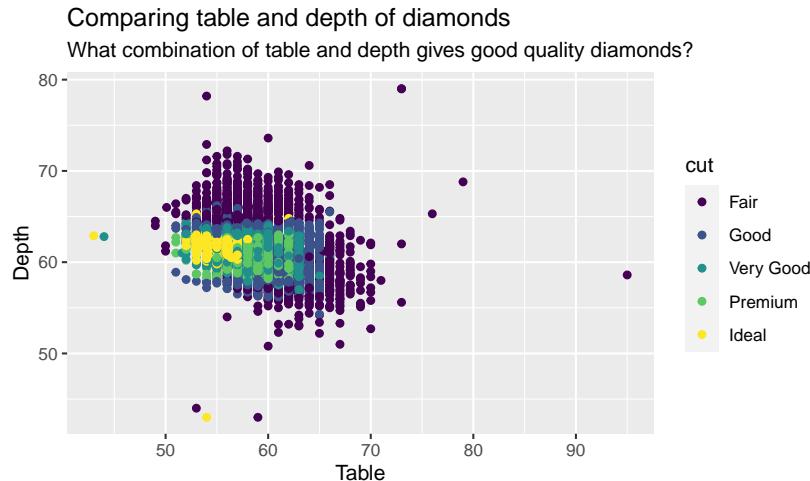
⁶ De tabel van een diamant verwijst naar het vlakke facet van de diamant dat kan worden gezien wanneer de steen naar boven wordt gekeerd. De diepte van een diamant is de hoogte (in millimeters) gemeten van de culet tot de tafel.



3.4.1 Titels

Een van de belangrijkste dingen om aan onze plot toe te voegen zijn titels. Titels worden gebruikt om betekenis te geven aan zowel de assen als de plot zelf. De meest eenvoudige manier om titels toe te voegen is door gebruik te maken van de functie `labs()`. In deze functie kunnen o.a. volgende argumenten worden ingesteld: de titel, de ondertitel, x voor het x label en y voor het y label. De `labs` functie kan gewoon aan de plot worden toegevoegd als een extra laag.

```
data("diamonds")
ggplot(diamonds) +
  geom_point(aes(table, depth, color = cut)) +
  labs(
    title = "Comparing table and depth of diamonds",
    subtitle = "What combination of table and depth gives good quality diamonds?",
    x = "Table",
    y = "Depth"
  )
```



Je zult zien dat onze grafiek er al veel beter uitziet als er titels aan toegevoegd zijn! Er is echter nog veel meer te verbeteren.

3.4.2 Theme

Het *theme* van een plot bepaalt het algemene uiterlijk: de rasterlijnen, de achtergrond, de grootte van de tekst, titels en legende, de positie van de legende, enz. Het thema kan handmatig worden gedefinieerd door een `theme()` laag toe te voegen aan de plot en door de benodigde argumenten in te stellen. (Je kunt kijken op `?theme` om te zien welke argumenten beschikbaar zijn). Dit is echter een omslachtige aanpak. Gelukkig zijn er enkele voorgedefinieerde thema's voorzien in ggplot:

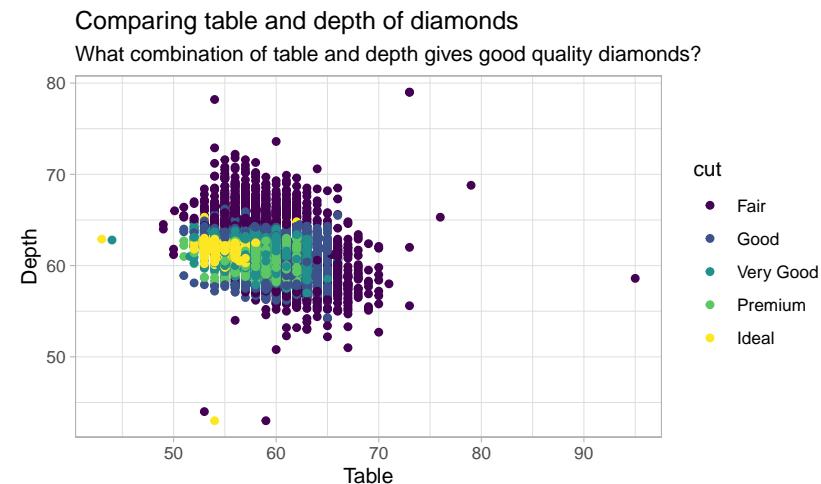
- `theme_gray`: het standaardthema (tot nu toe gebruikt)
- `theme_bw`: een thema voor zwart-wit plots
- `theme_dark`: een donker thema voor contrast
- `theme_classic`: een minimaal thema
- `theme_light`: een ander minimaal thema
- `theme_linedraw`: nog een minimaal thema
- `theme_minimal`: nog een minimaal thema
- `theme_void`: een leeg thema

Voel je vrij om met sommige van deze thema's te experimenteren. Bij voorkeur kunt u een aantal van de minimale thema's gebruiken. Hier, gebruikten we het `theme_light` thema.⁷

```
ggplot(diamonds) +
  geom_point(aes(table, depth, color = cut)) +
  labs(
    title = "Comparing table and depth of diamonds",
    subtitle = "What combination of table and depth gives good quality diamonds?",
```

⁷ Voor de meeste lagen is het niet belangrijk in welke volgorde ze aan een plot worden toegevoegd. Echter, als je handmatig wijzigingen aanbrengt met `theme`, zorg er dan voor dat je ze na een voorgedefinieerd thema plaatst, anders zullen je wijzigingen worden overschreven.

```
x = "Table",
y = "Depth"
) +
theme_light()
```



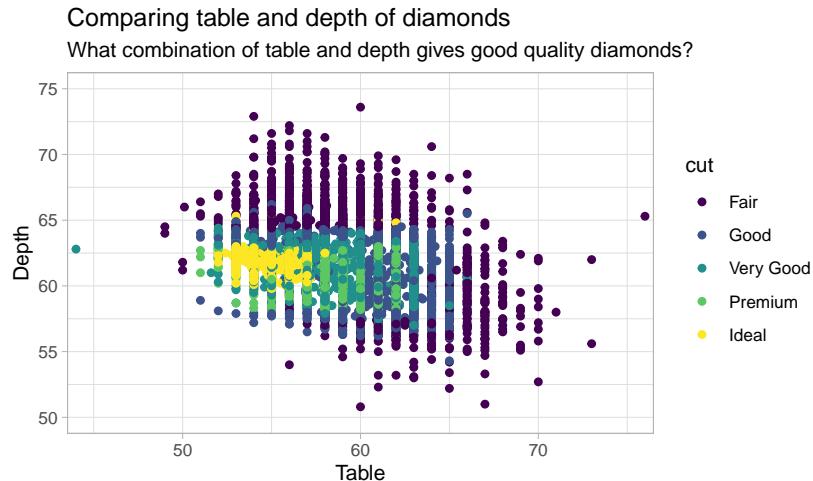
Wanneer je nog niet tevreden bent één van deze thema's, kunt je het pakket `ggthemes` installeren om nog meer thema's te verkrijgen, zoals het thema van *The Economist*, fivethirtyeight.com, of *Google Docs*.

3.4.3 Het coördinatenstelsel

Het uiterlijk van de grafiek wordt niet alleen bepaald door de titels en de grafieken. Ook de assen in het assenstelsel verdienen de nodige aandacht. Een van de dingen die moeten worden bepaald zijn de grenzen van het coördinatenstelsel. Dit kan worden gedaan met de functie `coord_cartesian` en zijn argumenten `xlim` en `ylim`. Beide argumenten verwachten een numerieke vector van lengte twee. Laten we eens kijken hoe dit werkt in ons voorbeeld.⁸

```
ggplot(diamonds) +
  geom_point(aes(table, depth, color = cut)) +
  labs(
    title = "Comparing table and depth of diamonds",
    subtitle = "What combination of table and depth gives good quality diamonds?",
    x = "Table",
    y = "Depth"
  ) +
  theme_light() +
  coord_cartesian(xlim = c(45, 75), ylim = c(50, 75))
```

⁸Een cartesisch coördinatenstelsel is een coördinatenstelsel dat elk punt op unieke wijze in een vlak specificeert door een paar numerieke coördinaten, die de getekende afstanden tot het punt zijn van twee vaste loodrecht op elkaar staande gerichte lijnen, gemeten in dezelfde lengte-eenheid. Het is vernoemd naar wetenschapper René Descartes.



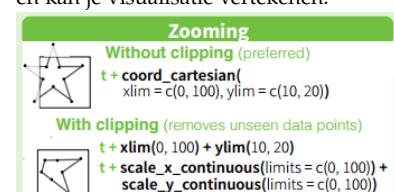
We hebben nu de x-as beperkt tot het interval van 45 tot 75, terwijl we de y-as hebben beperkt tot het interval 50 tot 75. Er zijn enkele alternatieven voor het cartesisch coördinatenstelsel die minder vaak worden gebruikt:

- coord_equal: een coördinatenstelsel waarbij de x-as en de y-as gelijk geschaald zijn (d.w.z. verhouding = 1)
- coord_fixed: een assenstelsel met een vaste verhouding (maar niet noodzakelijk 1)
- coord_polar: een coördinatensysteem voor polaire plots, of cirkeldiagrammen
- coord_map: een assenstelsel voor het plotten van geografische data.

Naast het instellen van de grenzen van het coördinatensysteem, kunnen we ook de breaks op de x-as en de y-as instellen. Dit kan respectievelijk met de functies scale_x_continuous en scale_y_continuous. Beide functies hebben een *breaks* argument. Dit argument kan worden gegeven als een vector van waarden die als labels op de as moeten worden geplot.⁹ We kunnen de functie seq gebruiken om deze vector te maken: d.w.z. seq(0, 10, 5) zal een vector teruggeven die begint bij 0 en oploopt tot tien met intervallen van 5.¹⁰

```
ggplot(diamonds) +
  geom_point(aes(table, depth, color = cut)) +
  labs(
    title = "Comparing table and depth of diamonds",
    subtitle = "What combination of table and depth gives good quality diamonds?",
    x = "Table",
    y = "Depth"
  ) +
```

⁹ Merk op dat de scale_..continuous functies ook een argument limits hebben om de grenzen van de assen in te stellen, dat gebruikt kan worden in plaats van coord_cartesian. Er is echter een belangrijk verschil. Coord_cartesian zal inzoomen op de grenzen zonder andere datapunten weg te gooien. Het instellen van de grenzen binnen de scale-functies echter, zal datapunten weggooien en kan je visualisatie vertekenen.

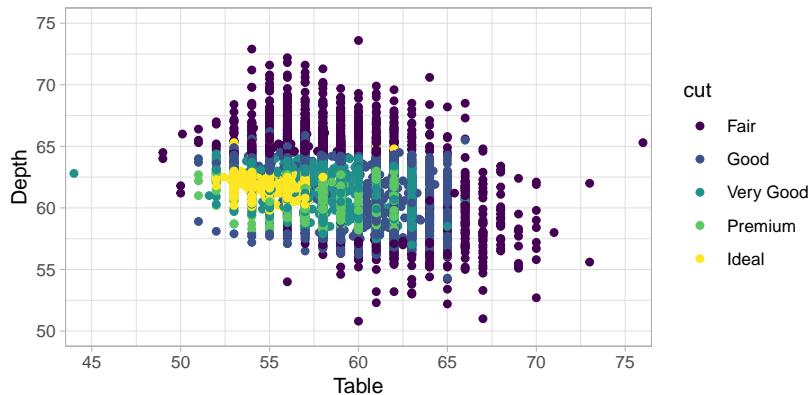


¹⁰ De titels van de assen die we gedefinieerd hebben met de labs functie kunnen ook ingesteld worden in de scale-functies met het argument name. Naarmate je meer vertrouwd raakt met het gebruik van ggplot2, zul je vaak merken dat er meerdere manieren zijn om hetzelfde doel te bereiken.

```
theme_light() +
coord_cartesian(xlim = c(45, 75), ylim = c(50, 75)) +
scale_x_continuous(breaks = seq(45, 75, 5)) +
scale_y_continuous(breaks = seq(50, 75, 5))
```

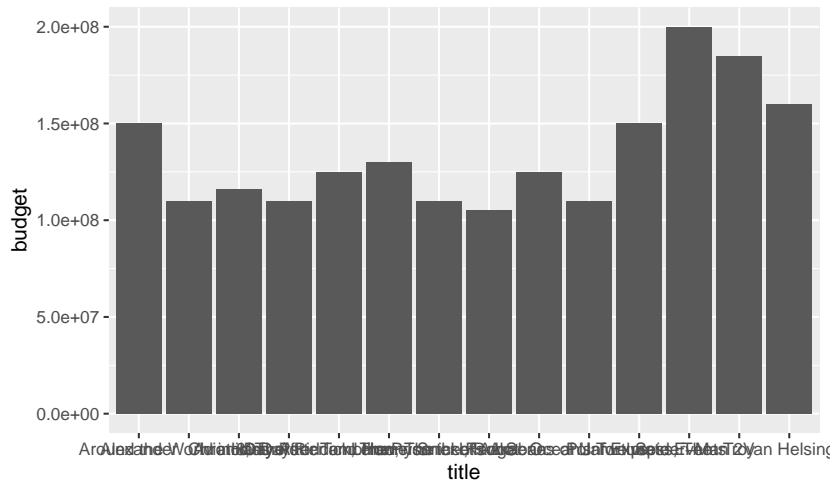
Comparing table and depth of diamonds

What combination of table
and depth gives good quality diamonds?



Een andere handige functie is de `coord_flip` functie, die we zullen illustreren met de volgende grafiek die we eerder zagen.

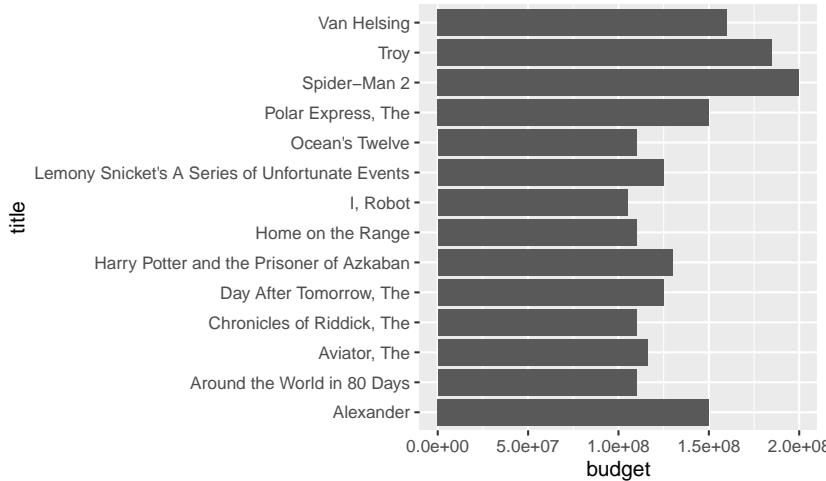
```
filter(movies, year == 2004, budget > 100000000) %>%
ggplot() +
geom_col(aes(title, budget))
```



Zoals je zich wellicht herinnert, overlapten de filmtitels op de x-as elkaar en waren daardoor onleesbaar. Een manier om dit te verhelpen is door de hele grafiek *om te draaien*, zodat de labels van de x-as op de y-as komen te staan, en horizontaal kunnen worden gelezen.

```
filter(movies, year == 2004, budget > 100000000) %>%
ggplot() +
```

```
geom_col(aes(title, budget)) +
coord_flip()
```



Een andere optie is om de oorspronkelijke oriëntatie te behouden, maar de oriëntatie van de labels op de x-as te veranderen. Je kunt ze bijvoorbeeld 45 of 90 graden draaien. Dit kan worden gedaan met de `theme` functie. Klaar om te experimenteren? Daag jezelf uit!

Naarmate onze code meer en meer regels bevat, wordt onze plot mooier en mooier! Goed gedaan! Het laatste op onze lijst zijn kleuren.

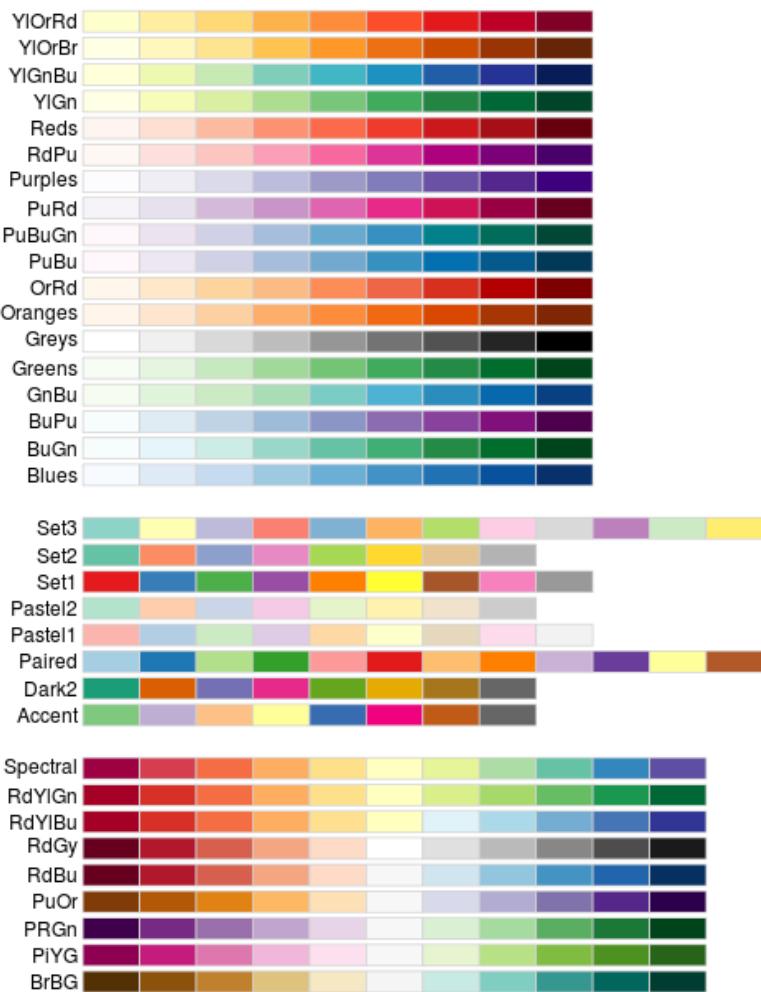
3.4.4 Color scales

Vaak gebruiken we kleur of fill om categorische gegevens te visualiseren, zoals de kwaliteit in onze grafiek over diamanten. Standaard zal ggplot een regenboog-thema gebruiken. Er zijn echter veel meer paletten beschikbaar. We kunnen deze toevoegen door `scale_color_brewer` of `scale_fill_brewer` te gebruiken, afhankelijk van of het om een kleur of fill-kleur gaat. Beide layers hebben een palette argument, waarvan je hieronder de mogelijke waarden kunt vinden.

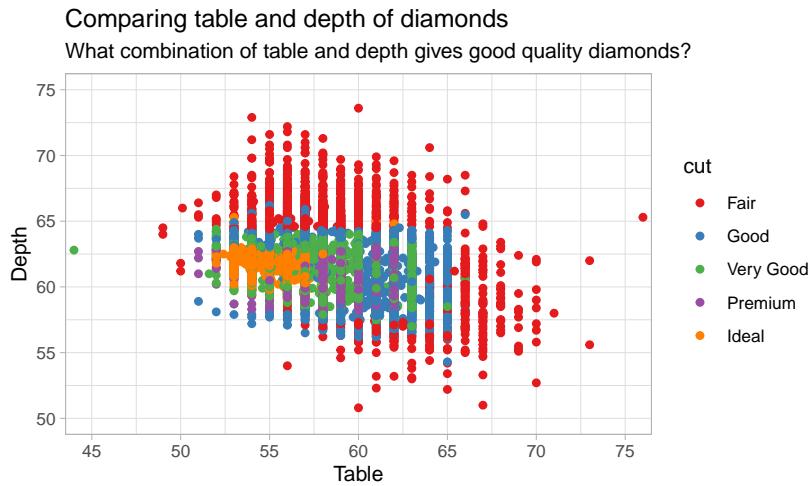
Bijvoorbeeld, laten we het Set1 palet gebruiken.

```
ggplot(diamonds) +
  geom_point(aes(table, depth, col = cut)) +
  labs(
    title = "Comparing table and depth of diamonds",
    subtitle = "What combination of table and depth gives good quality diamonds?",
    x = "Table",
    y = "Depth"
  ) +
  theme_light() +
```

Figure 3.1: R color palettes

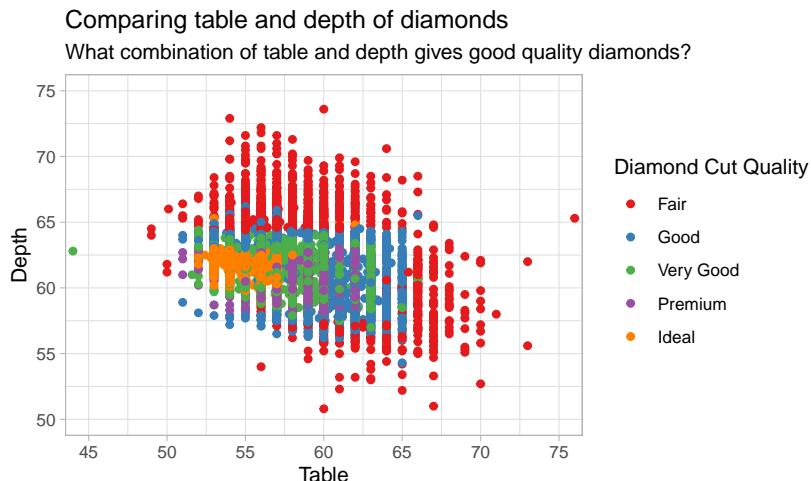


```
coord_cartesian(xlim = c(45, 75), ylim = c(50, 75)) +
scale_x_continuous(breaks = seq(45, 75, 5)) +
scale_y_continuous(breaks = seq(50, 75, 5)) +
scale_color_brewer(palette = "Set1")
```



De scale_._brewer functies hebben ook het argument name, waarmee we de naam van de legende kunnen instellen, en het argument guide, waarmee de legende wordt verwijderd als deze op FALSE is gezet.

```
ggplot(diamonds) +
  geom_point(aes(table, depth, col = cut)) +
  labs(
    title = "Comparing table and depth of diamonds",
    subtitle = "What combination of table and depth gives good quality diamonds?",
    x = "Table",
    y = "Depth"
  ) +
  theme_light() +
  coord_cartesian(xlim = c(45, 75), ylim = c(50, 75)) +
  scale_x_continuous(breaks = seq(45, 75, 5)) +
  scale_y_continuous(breaks = seq(50, 75, 5)) +
  scale_color_brewer(palette = "Set1", name = "Diamond Cut Quality")
```



Naast de standaard kleurenpaletten die beschikbaar zijn, zijn er nog veel meer te vinden in de pakketten `ggthemes` en `ggsci`. Ze kunnen worden gebruikt door `scale_color` of `scale_fill` + naam van het palet toe te voegen. Voel je vrij om er nog meer te ontdekken!

3.5 Geavanceerde plots

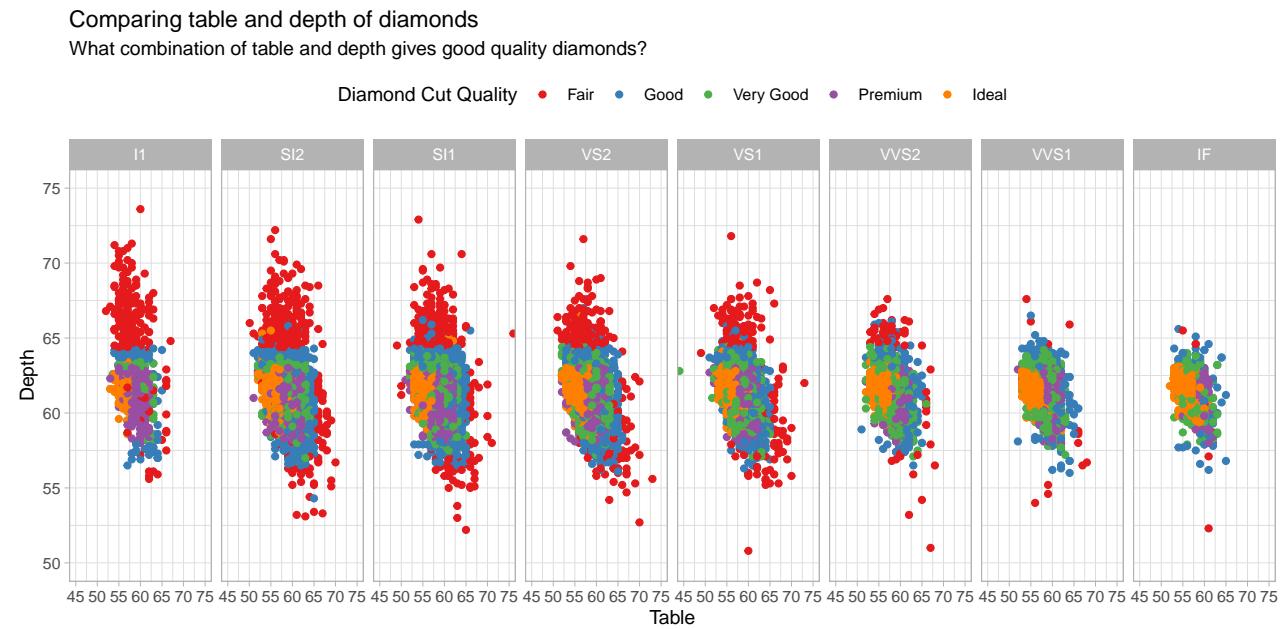
Vaak wil je plots vergelijken voor verschillende categorieën van een variabele. In ons voorbeeld hebben we gekeken voor welke combinaties van table en depth, de cut-quality van de diamant goed was. Nu willen we weten of er een verschil is tussen de 8 verschillende clarities in de gegevens. Eén manier zou zijn om 8 verschillende plots te maken, één voor elk van de niveaus. Dit zou echter omslachtig zijn om te doen. Gelukkig kunnen we de functie `facet_grid` gebruiken om verschillende plots binnen een plot te maken.

3.5.1 Gebruikmaken van Facets

Deze functie verwacht een formule in de vorm van $A \sim B$ waarbij A en B twee categorische variabelen zijn. Voor elke combinatie van waarden van A en B wordt een andere plot geconstrueerd, en deze worden gerangschikt in een rooster waarbij de waarden van A elk een rij vormen en de waarden van B elk een kolom vormen. Vergelijkingen van meer dan 2 variabelen zijn mogelijk met een formule van de vorm $A + B \sim C$. Een vergelijking langs één variabele is mogelijk door een punt te gebruiken in plaats van een variabelenaam, d.w.z. $\cdot \sim A$ of $A \sim \cdot$.

In de volgende plot gebruiken we facetten om onze plot opnieuw te tekenen voor elk van de helderheidsniveaus (clarity). Bovendien is de legende bovenaan geplaatst om meer ruimte te creëren.

```
ggplot(diamonds) +
  geom_point(aes(table, depth, col = cut)) +
  labs(
    title = "Comparing table and depth of diamonds",
    subtitle = "What combination of table and depth gives good quality diamonds?",
    x = "Table",
    y = "Depth"
  ) +
  theme_light() +
  coord_cartesian(xlim = c(45, 75), ylim = c(50, 75)) +
  scale_x_continuous(breaks = seq(45, 75, 5)) +
  scale_y_continuous(breaks = seq(50, 75, 5)) +
  scale_color_brewer(palette = "Set1", name = "Diamond Cut Quality") +
  facet_grid(. ~ clarity) +
  theme(legend.position = "top")
```



Een alternatief, meestal geschikt voor vergelijkingen langs één variabele, is facet_wrap. In plaats van één rij te maken (zoals facet_grid doet), zal het de plots ordenen in een raster met een opgegeven aantal kolommen of rijen. Hieronder hebben we ze in 3 kolommen gerangschikt.

```
ggplot(diamonds) +
  geom_point(aes(table, depth, col = cut)) +
  labs(
    title = "Comparing table and depth of diamonds",
```

```

subtitle = "What combination of table and depth gives good quality diamonds?",
x = "Table",
y = "Depth"
) +
theme_light() +
coord_cartesian(xlim = c(45, 75), ylim = c(50, 75)) +
scale_x_continuous(breaks = seq(45, 75, 5)) +
scale_y_continuous(breaks = seq(50, 75, 5)) +
scale_color_brewer(palette = "Set1", name = "Diamond Cut Quality") +
facet_wrap(~clarity, ncol = 3) +
theme(legend.position = "top")

```

Dit is veel efficienter dan 8 grafieken naast elkaar!

3.5.2 Meerdere layers combineren

Tot nu toe hebben we slechts één geometrische layer tegelijk gebruikt. Het is echter perfect mogelijk om verschillende lagen te combineren. Zo kunnen we bijvoorbeeld het geom_text label gebruiken om data labels toe te voegen aan een staafdiagram. Geom_text is een geometrische laag die we inderdaad kunnen gebruiken om tekst in een grafiek te zetten. We hebben geom_text nog niet eerder gezien, maar de werking ervan zal eenvoudig zijn, gebaseerd op alles wat we al weten. We bouwen verder op een vorige grafiek, die we een iets fraaier uiterlijk hebben gegeven.

```

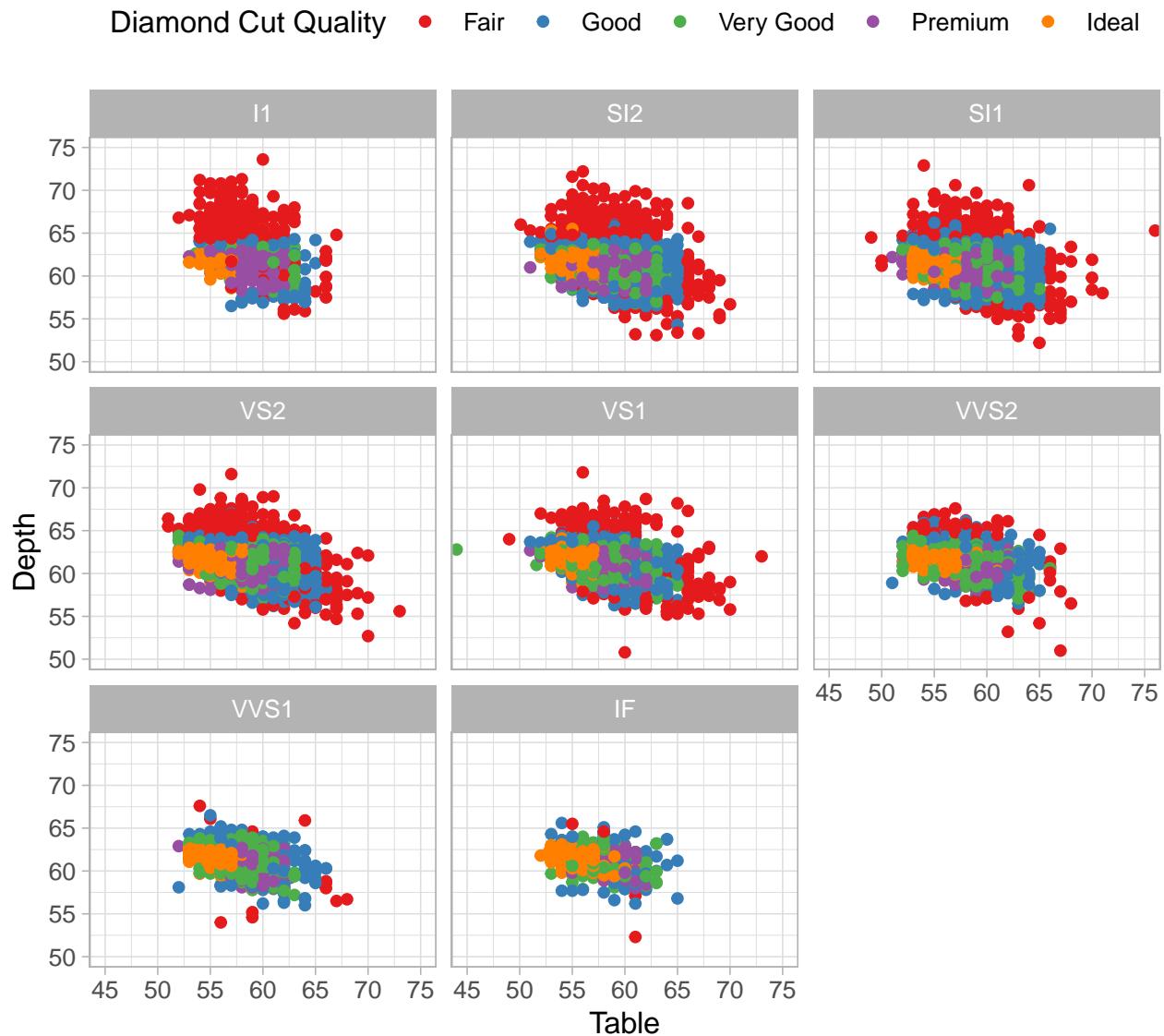
filter(movies, year == 2004, budget > 100000000) %>%
  ggplot() +
  geom_col(aes(reorder(title, budget), budget / 1000000), fill = "grey") +
  coord_flip() +
  labs(
    title = "Movies budgets",
    subtitle = "What was the budget of the movies from 2004 with a budget higher than 100 million?",
    x = "Title",
    y = "Budget (in million dollars)"
  ) +
  theme_light()

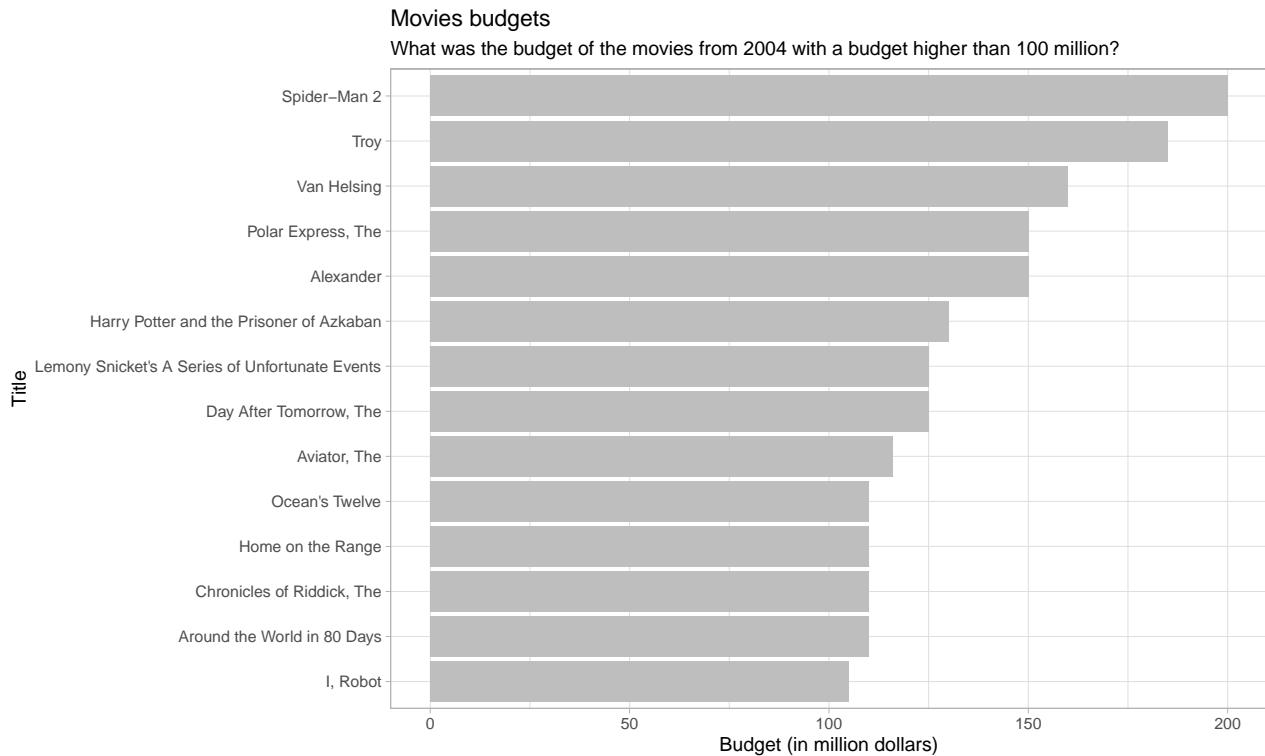
```

Merk op dat we de y-waarde veranderd hebben in budget/1000000, zodat de waarden in miljoenen zijn. Verder is het belangrijk op te merken dat, aangezien we coord_flip hebben gebruikt, onze labels in labs ook zijn verwisseld. Dus verschijnt het x-label op de y-as en het y-label op de x-as. Dit is precies wat we zouden willen, omdat het verwijderen van coord_flip in de toekomst de labels niet in de war zal sturen.

Comparing table and depth of diamonds

What combination of table and depth gives good quality diamonds?

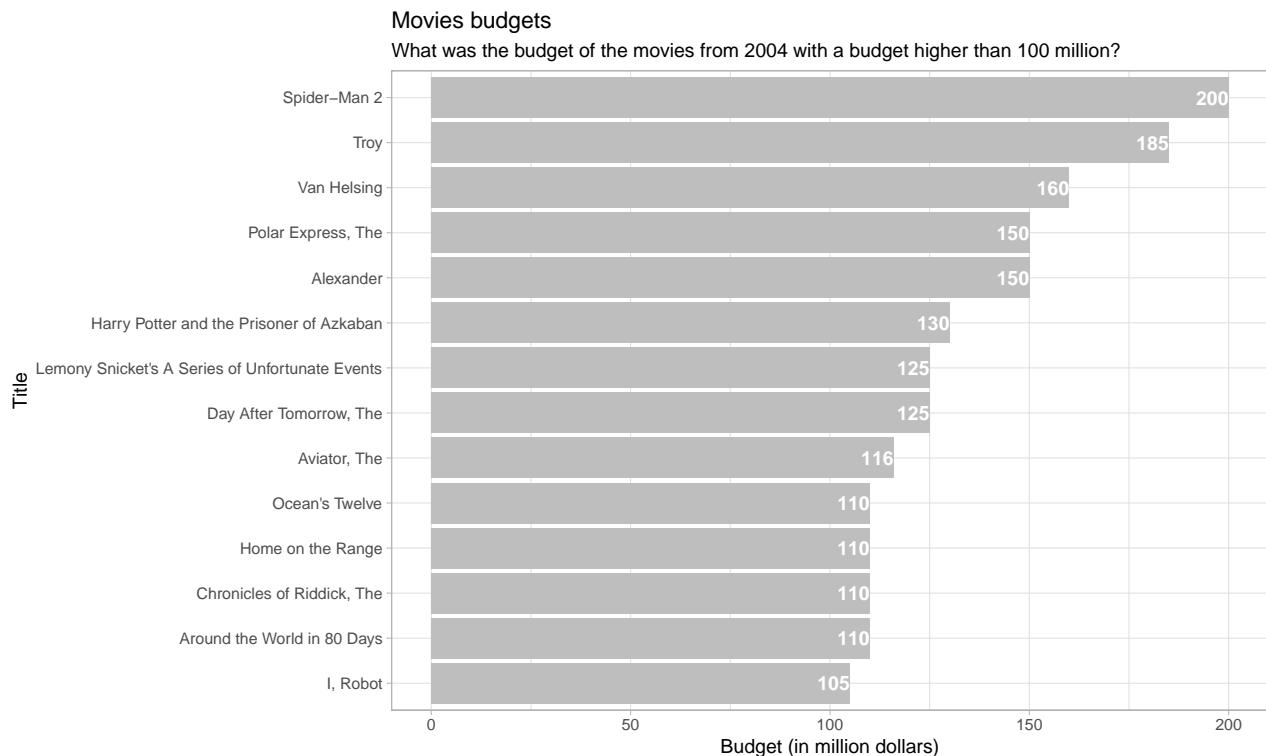




We willen nu het exacte aantal miljoenen boven op de staven zetten. Om dit te doen, voegen we geom_text toe, en geven het dezelfde mapping voor x en y als de geom_col laag. Verder voegen we een mapping toe voor het label, d.w.z. de tekst die moet worden weergegeven. We geven de tekst een vet lettertype en een witte kleur. Tenslotte zorgt de instelling hjust op 1 ervoor dat de tekstlabels horizontaal rechts worden uitgelijnd. Dit zorgt ervoor dat de tekst volledig binnen de balken blijft, en er niet uit valt.

```
filter(movies, year == 2004, budget > 100000000) %>%
  ggplot() +
  geom_col(aes(reorder(title, budget), budget / 1000000), fill = "grey") +
  geom_text(aes(reorder(title, budget), budget / 1000000,
    label = budget / 1000000
  ), color = "white", fontface = "bold", hjust = 1) +
  coord_flip() +
  labs(
    title = "Movies budgets",
    subtitle = "What was the budget of the movies from 2004 with a budget higher than 100 million?",
    x = "Title",
    y = "Budget (in million dollars)"
  ) +
```

```
theme_light()
```



De toevoeging van de tweede geom layer lijkt een beetje omslachtig, omdat we de mapping voor x en y moesten herhalen, wat nog erger werd omdat er de reorder functie en de deling door een miljoen bij betrokken waren. Er is toch wel een betere manier om dit te doen? We noemen het aesthetics-overerving

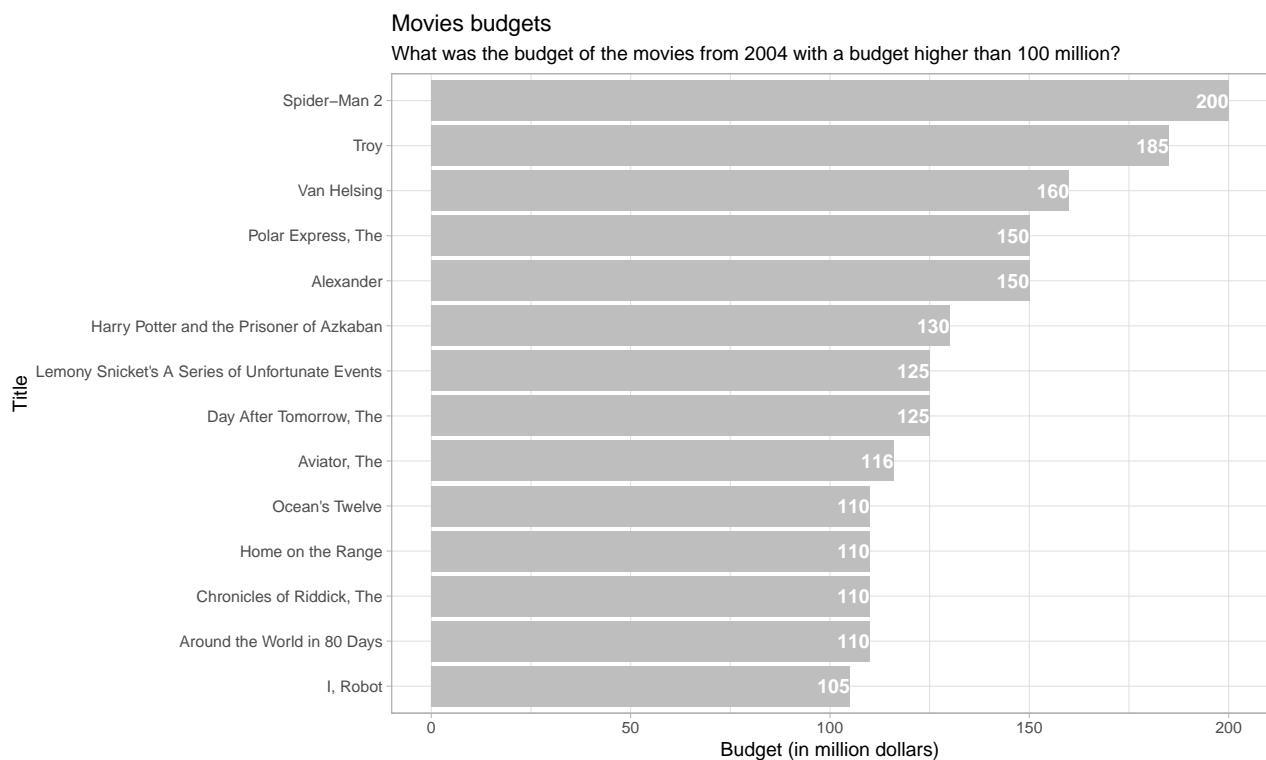
3.5.3 Aes-overerving

Aes-erfelijkheid, of overerving van de aes-mapping, betekent dat elk van de geom layers die wordt toegevoegd aan een ggplot functie de mapping *eft* die is gespecificeerd in die ggplot functie-aanroep. Zoals je je misschien nog herinnert van het begin, kan een aes-mapping in zowel geom layers worden geplaatst als in ggplot zelf. We leren nu dat er een klein maar belangrijk verschil is.

Wanneer verschillende layers (een deel van) een mapping gemeen hebben, is het het beste om dit deel te verplaatsen naar de ggplot aanroep. Op die manier hoef je dit niet te herhalen in de layers die het gebruiken. En, als een van de lagen deze mapping niet gebruikt, kunt je deze eenvoudig overschrijven door een nieuwe mapping op te geven in die laag. Laten we eens kijken naar een voorbeeld.

De vorige plot die we maakten kan eenvoudiger als volgt gemaakt worden:

```
filter(movies, year == 2004, budget > 100000000) %>%
  ggplot(aes(reorder(title, budget), budget / 1000000)) +
  geom_col(fill = "grey") +
  geom_text(aes(label = budget / 1000000), color = "white", fontface = "bold", hjust = 1) +
  coord_flip() +
  labs(
    title = "Movies budgets",
    subtitle = "What was the budget of the movies from 2004 with a budget higher than 100 million?",
    x = "Title",
    y = "Budget (in million dollars)"
  ) +
  theme_light()
```



Door de mapping voor x en y naar ggplot te verplaatsen, is er geen mapping nodig in geom_col, en enkel een mapping voor label in geom_text. Beide geom layers erven het andere deel van de mapping van de ggplot functie-aanroep. Echt, dit is veel efficiënter!

3.6 *Background material*

Je beheerst nu al heel wat van het plotten met ggplot2. Je hebt beiden geleerd hoe je verschillende geometrische lagen kunt gebruiken om gegevens weer te geven, hoe je ze kunt combineren, hoe je facetten kunt gebruiken, hoe je je code efficiënter kunt maken met aes-overerving, en last but not least, hoe je je plot een mooi uiterlijk kunt geven. Gefeliciteerd!

Als je graag nog meer wilt weten, kan je het volgende achtergrondmateriaal bekijken:

- [The ggplot Cheat Sheet](#), which provides an overview of all basic functionality in the ggplot2 package.
- [The ggplot documentation](#)
- [An even more comprehensive ggplot2 tutorial](#)