

# R Notebook

Load the necessary libraries to be used in the code

```
library(tidyr)
library(tidyverse)
library(dplyr)
library(ggplot2)
library(readxl)           # Voor het inlezen van data uit Excel
library(lubridate)        # Voor het interpreteren van datums, tijden en periodes
library(RSQLite)          # voor de ondersteuning van SQL in R
library(leaflet)
```

Haal data op uit Excel en plaats die in een SQLite database

```
database <- DBI::dbConnect(RSQLite::SQLite(), ":memory:")
copy_to(database, name="buurt",      df = read_excel('~R/Datasets/Van der Binckes Sharing/
Buurten.xlsx'))
copy_to(database, name="CAO",        df = read_excel('~R/Datasets/Van der Binckes Sharing/
CAO.xlsx'))
copy_to(database, name="fiets",      df = read_excel('~R/Datasets/Van der Binckes Sharing/
Fietsen.xlsx'))
copy_to(database, name="fietstype", df = read_excel('~R/Datasets/Van der Binckes Sharing/
FietsTypes.xlsx'))
copy_to(database, name="klant",      df = read_excel('~R/Datasets/Van der Binckes Sharing/
Klanten.xlsx'))
copy_to(database, name="locatie",    df = read_excel('~R/Datasets/Van der Binckes Sharing/
Locaties.xlsx'))
copy_to(database, name="medewerker", df = read_excel('~R/Datasets/Van der Binckes Sharing/
Medewerkers.xlsx'))
copy_to(database, name="onderhoud", df = read_excel('~R/Datasets/Van der Binckes Sharing/
OnderhoudsData.xlsx'))
copy_to(database, name="rooster",    df = read_excel('~R/Datasets/Van der Binckes Sharing/
Rooster.xlsx'))
copy_to(database, name="schade",     df = read_excel('~R/Datasets/Van der Binckes Sharing/
Schades.xlsx'))
copy_to(database, name="sharing",    df = read_excel('~R/Datasets/Van der Binckes Sharing/
SharingData.xlsx'))
copy_to(database, name="stadsdeel",  df = read_excel('~R/Datasets/Van der Binckes Sharing/
Stadsdelen.xlsx'))
copy_to(database, name="weer",       df = read_excel('~R/Datasets/Van der Binckes Sharing/
Weerdata.xlsx'))
copy_to(database, name="wijk",       df = read_excel('~R/Datasets/Van der Binckes Sharing/
Wijken.xlsx'))
```

#Maak een overzicht van de sharing

## Aantal fietsen per type en uitvoering

Met dit SQL commando krijg je het aantal fietsen per type en uitvoering in een tabel

```
select FietsType.ID, FietsType.OMSCHRIJVING, fietstype.UITVOERING, count(*) AS Aantal
from sharing
inner join Fiets
  on Fiets.ID = sharing.fietsid
inner join Fietstype
  on Fietstype.ID = fiets.FIETSTYPEID
GROUP BY Fietstype.id, Fietstype.omschrijving, fietstype.UITVOERING
ORDER BY 1
```

Verzin nu zelf een manier om dat met ggplot te presenteren

ID	OMSCHRIJVING	UITVOERING	Aantal
<dbl>	<chr>	<chr>	<int>
3	CargoBike Classic	Korte bak	2993
4	CargoBike Classic	Classic	2780
9	CargoBike Cruiser	Korte bak	5650
10	CargoBike Cruiser	Cruiser	4006
20	Bimas Kids	Groen	2064
21	Bimas Kids	Economy	2605

6 rows

## Aantal klanten per land

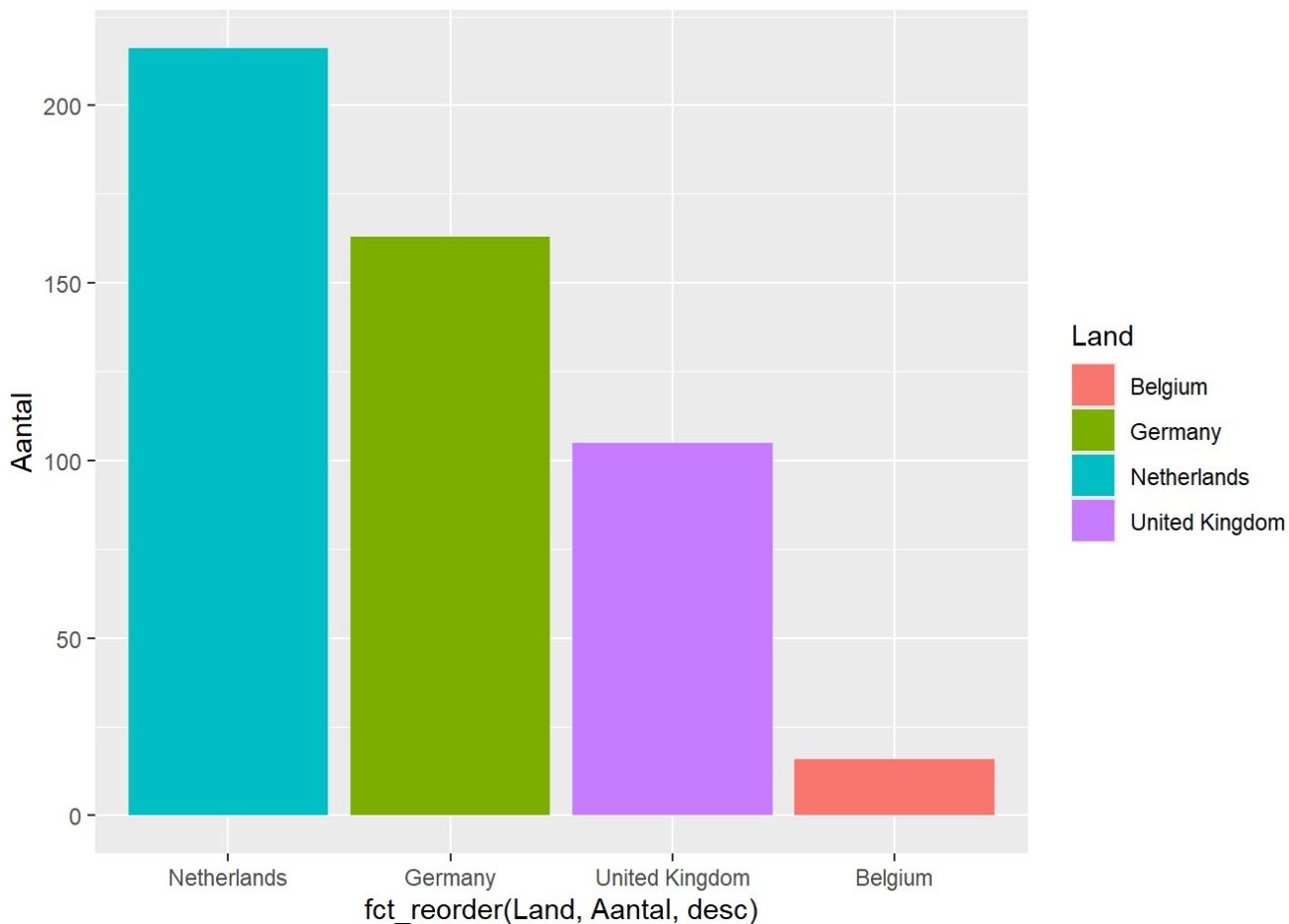
Maak met SQL een tabel die er zo uitziet

```
frameKlantenPerLand
```

Land	Aantal
<chr>	<int>
Belgium	16
Germany	163
Netherlands	216
United Kingdom	105

4 rows

Maak aan de hand deze tabel onderstaande kolomdiagram



## Kilometers per Leverancier

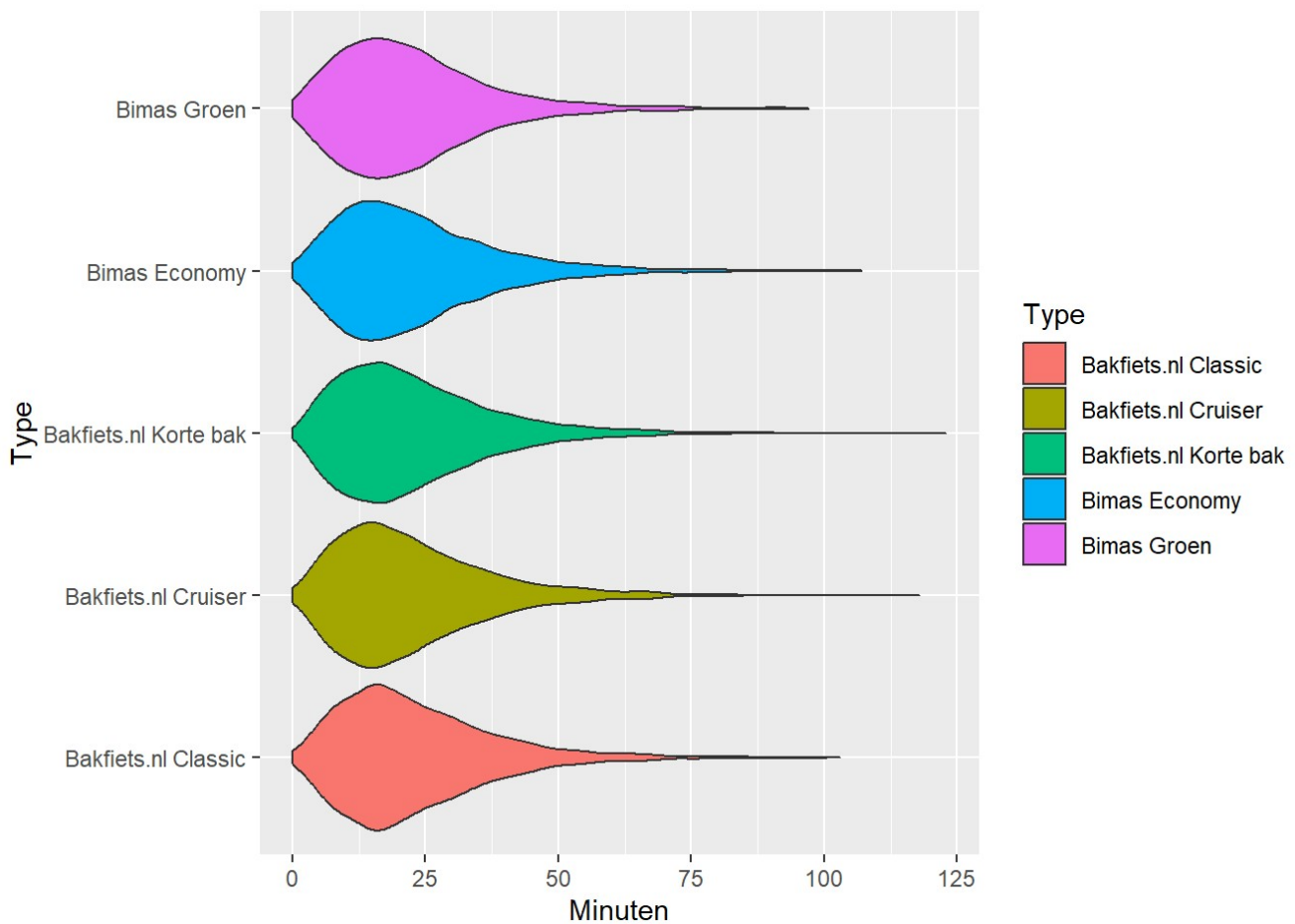
Gebruik SQL om een tabel te maken die er zo uitziet

```
frameKilometersPerLeverancier
```

Type <chr>	Afstand <dbl>	Minuten <dbl>
Bakfiets.nl Classic	3.5	9
Bakfiets.nl Classic	5.4	29
Bakfiets.nl Classic	4.1	16
Bakfiets.nl Classic	0.5	1
Bakfiets.nl Classic	6.1	32
Bakfiets.nl Classic	5.2	13
Bakfiets.nl Classic	9.3	22
Bakfiets.nl Classic	2.1	7
Bakfiets.nl Classic	3.0	9
Bakfiets.nl Classic	5.0	16
1-10 of 10,000 rows		
Previous 1 2 3 4 5 6 ... 1000 Next		

## Gebruik vervolgens ggplot om een violin chart te maken.

Het voordeel van een violin chart is dat je goed kunt zien hoe de verdeling van de kilometers is. Beter dan alleen een gemiddelde en zelfs beter dan in een box-plot.



## Totale kosten (onderhoud en schade) per fietstype

Maak met SQL een tabel die er zo uitziet

```
frameKostenPerFietsType
```

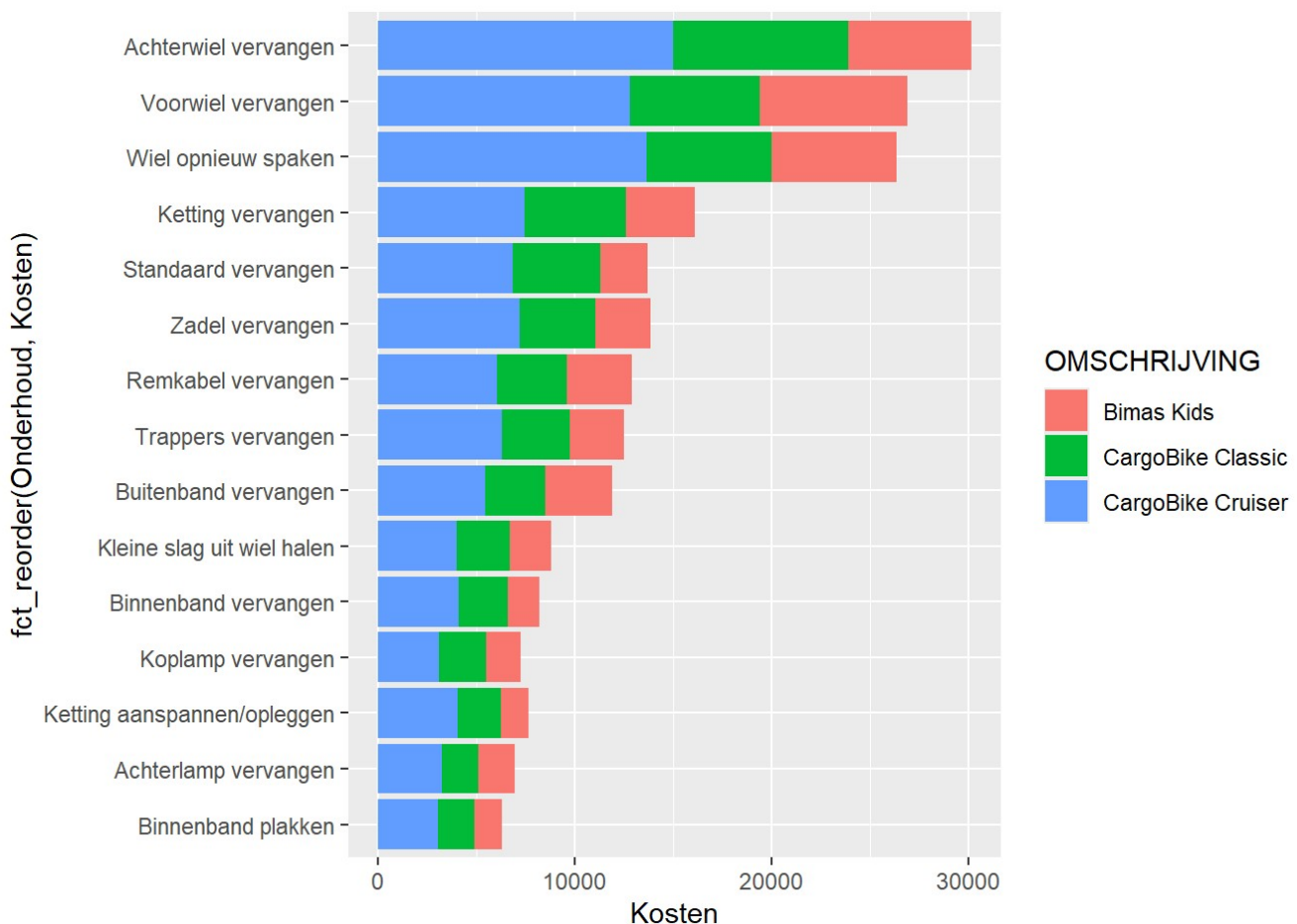
LEVERAN...<chr>	OMSCHRIJVING<chr>	UITVOER...<chr>	Type<chr>
Bakfiets.nl	CargoBike Classic	Classic	Bakfiets.nl: CargoBike Classic (Classic)
Bakfiets.nl	CargoBike Classic	Classic	Bakfiets.nl: CargoBike Classic (Classic)
Bakfiets.nl	CargoBike Classic	Classic	Bakfiets.nl: CargoBike Classic (Classic)
Bakfiets.nl	CargoBike Classic	Classic	Bakfiets.nl: CargoBike Classic (Classic)
Bakfiets.nl	CargoBike Classic	Classic	Bakfiets.nl: CargoBike Classic (Classic)
Bakfiets.nl	CargoBike Classic	Classic	Bakfiets.nl: CargoBike Classic (Classic)
Bakfiets.nl	CargoBike Classic	Classic	Bakfiets.nl: CargoBike Classic (Classic)
Bakfiets.nl	CargoBike Classic	Classic	Bakfiets.nl: CargoBike Classic (Classic)

LEVERAN...	OMSCHRIJVING	UITVOER...	Type
<chr>	<chr>	<chr>	<chr>
Bakfiets.nl	CargoBike Classic	Classic	Bakfiets.nl: CargoBike Classic (Classic)
Bakfiets.nl	CargoBike Classic	Classic	Bakfiets.nl: CargoBike Classic (Classic)

1-10 of 90 rows | 1-4 of 6 columns

Previous123456...9Next

Maak aan de hand van de tabel een staafdiagram dat er zo uitziet.



Een heatmap op basis van het aantal kilometers van/naar een stadsdeel

Maak met SQL een tabel die er zo uitziet

```
frameKmsBeginEindeStadsdeel
```

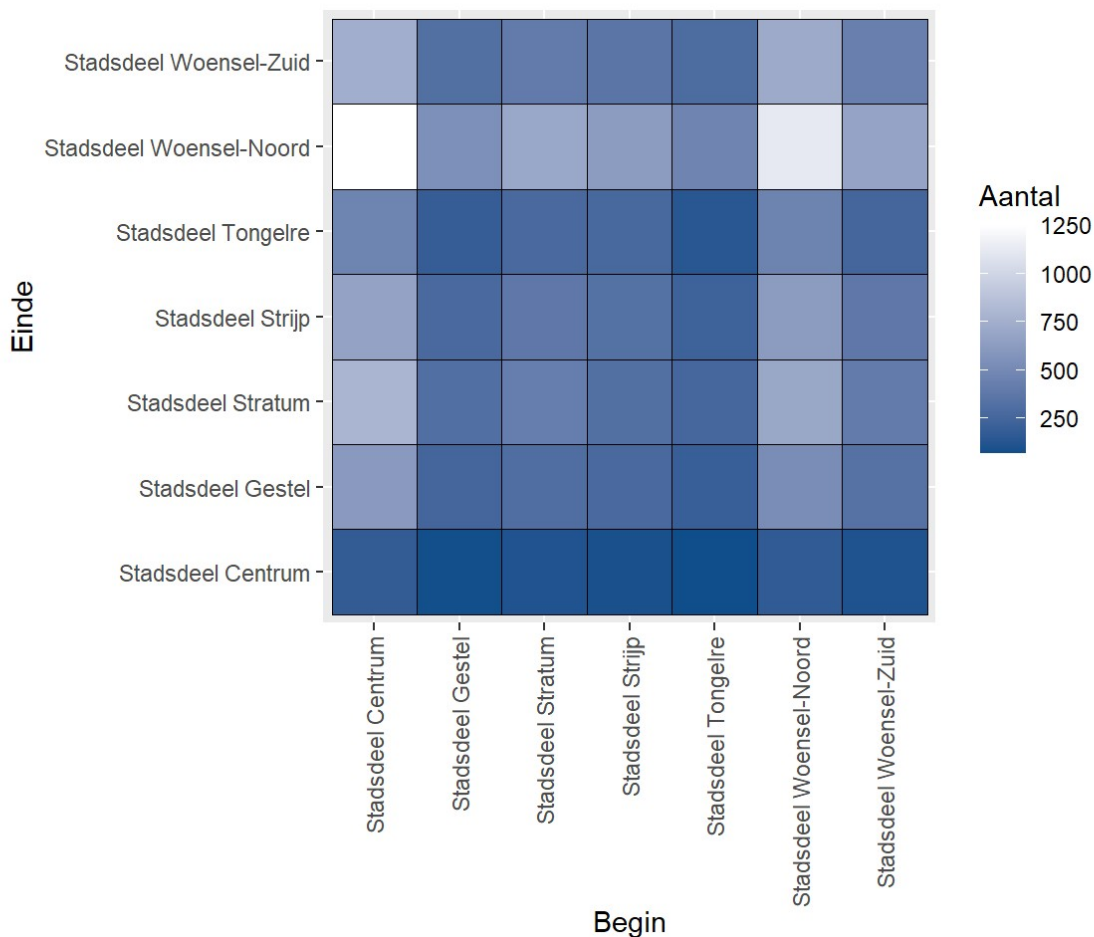
Begin	Einde	Afstand	Aantal
<chr>	<chr>	<dbl>	<int>
Stadsdeel Centrum	Stadsdeel Centrum	165.7	174
Stadsdeel Centrum	Stadsdeel Gestel	2684.5	607

Begin <chr>	Einde <chr>	Afstand <dbl>	Aantal <int>
Stadsdeel Centrum	Stadsdeel Stratum	2752.2	791
Stadsdeel Centrum	Stadsdeel Strijp	3833.9	666
Stadsdeel Centrum	Stadsdeel Tongelre	2075.2	469
Stadsdeel Centrum	Stadsdeel Woensel-Noord	7545.8	1256
Stadsdeel Centrum	Stadsdeel Woensel-Zuid	2095.6	747
Stadsdeel Gestel	Stadsdeel Centrum	285.4	73
Stadsdeel Gestel	Stadsdeel Gestel	585.9	249
Stadsdeel Gestel	Stadsdeel Stratum	1865.9	314

1-10 of 49 rows

Previous 1 2 3 4 5 Next

Maak op basis daarvan een heatmap die er zo uitziet



## Gebruik van datums

### Maak van de datum een tekst

%Y = jaar %m = maand %W = week (let op met weeknummers! Die zijn niet altijd aansluitend bij de Europese kalender) %D = dag %H = uur %M = minuut %S = seconde %s = unix datum formaat %j = dag in

het jaar %w = dag van de week (0=zo 1=ma 2=di 3=wo 4=do 5=vr 6=za)

```
select strftime('%Y-%W', Datum) as week
, count(*)
from weer
inner join sharing
on strftime('%Y-%M-%D', sharing.dtVertrek) = strftime('%Y-%M-%D')
group by week
```

0 records

**weekcount(\*)**

## Voorbeeld hoe je datums en tijden vertaalt van naar het ene naar het andere formaat

dtVertrek uit sharing vertalen naar tekst Datum als tekst: DateTime(dtVertrek, 'unixepoch') Datum uit dtVertrek: DATE(DateTime(dtVertrek, 'unixepoch')) Tijd uit dtVertrek: TIME(DateTime(dtVertrek, 'unixepoch'))

```
select id
, DateTime(dtVertrek, 'unixepoch') as 'SharingDate'
, strftime("%Y %j", DateTime(dtVertrek, 'unixepoch'))
, DATE(DateTime(dtVertrek, 'unixepoch')) as 'SharingDate'
, TIME(DateTime(dtVertrek, 'unixepoch')) as 'SharingTime'
, KlantID
from sharing
order by 2, 1
```

Displaying records 1 - 10

ID	SharingDate	strftime("%Y %j", DateTime(dtVertrek, 'unixepoch'))	SharingDate	SharingTime	KlantID
2	2023-07-11 15:05:00	2023 192	2023-07-11	15:05:00	474
94	2023-07-11 17:58:00	2023 192	2023-07-11	17:58:00	1279
180	2023-07-11 20:10:00	2023 192	2023-07-11	20:10:00	93
243	2023-07-12 09:34:00	2023 193	2023-07-12	09:34:00	576
333	2023-07-12 12:32:00	2023 193	2023-07-12	12:32:00	1422
7	2023-07-12 17:10:00	2023 193	2023-07-12	17:10:00	890
57	2023-07-13 04:16:00	2023 194	2023-07-13	04:16:00	299
486	2023-07-13 08:50:00	2023 194	2023-07-13	08:50:00	745
107	2023-07-13 13:50:00	2023 194	2023-07-13	13:50:00	681

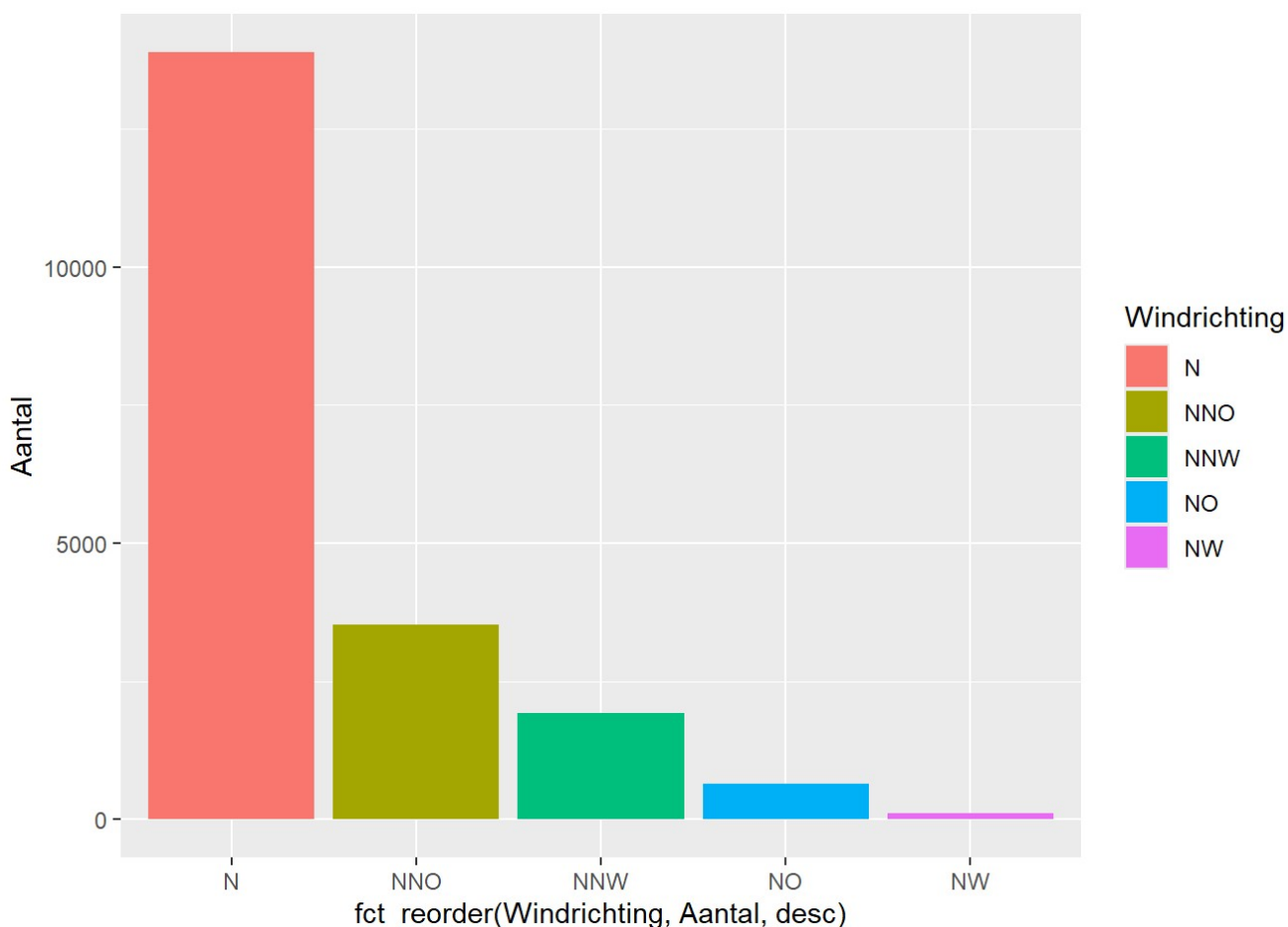
ID	SharingDate	strftime("%Y %j", DateTime(dtVertrek, 'unixepoch'))	SharingDate	SharingTime	KlantID
111	2023-07-13 17:36:00	2023 194	2023-07-13	17:36:00	1867

## Koppel de data van de tabel weer op basis van datum

```
select weer.Windrichting as 'Windrichting'
, count(*) as 'Aantal'
from sharing
inner join weer
on weer.Datum = Date(DateTime(sharing.dtVertrek, 'unixepoch'))
group by weer.Windrichting
order by 1 asc
```

## Gebruik dat in een overzicht

```
ggplot (frameWindrichting, aes(x = fct_reorder(Windrichting, Aantal, desc), y=Aantal, fill
= Windrichting)) +
  geom_col()
```



## Aantal van ritten per buurt



```
select locatie.BUURTCODE as 'BuurtID'
, buurt.BUURTNAAM as 'Buurt'
, avg(locatie.LON) as 'longitude'
, avg(locatie.LAT) as 'latitude'
, count(*) as 'aantal'
from sharing
inner join locatie
on locatie.ID = sharing.LocatieBeginID
inner join buurt
on buurt.BUURTCODE = locatie.BUURTCODE
where (TIME(DateTime(sharing.dtVertrek, 'unixepoch')) between '22:00' and '23:59')
OR (TIME(DateTime(sharing.dtVertrek, 'unixepoch')) between '00:00' and '06:00')
group by locatie.BUURTCODE, buurt.BUURTNAAM
order by 1, 2, 3
```

## Definieer categorieën met een begin- en eindaantal

```
numBreaks <- c(0,10,20,30,40,50,60,70,80,90,50000)
numColors <- c("0","1","2","3","4","5","6","7","8","9")
frameLocaties <- frameLocaties %>%
  mutate(colorGroup = cut(aantal, breaks = numBreaks, right = FALSE, labels = numColors))
frameLocaties
```

BuurtID <chr>	Buurt <chr>	longitude <dbl>	latitude <dbl>	aantal <int>	colorGroup <fct>						
111	Binnenstad	5.480238	51.43809	79	7						
112	Bergen	5.473688	51.43625	59	5						
113	Witte Dame	5.471711	51.43958	18	1						
114	Fellenoord	5.475127	51.44365	24	2						
115	TU-terrein	5.491104	51.44791	15	1						
211	Irisbuurt	5.492342	51.43417	53	5						
212	Rochusbuurt	5.488278	51.43221	37	3						
213	Elzent-Noord	5.482810	51.43091	43	4						
214	Tuindorp	5.494440	51.42909	69	6						
215	Joriskwartier	5.490566	51.42775	34	3						
1-10 of 115 rows		Previous	1	2	3	4	5	6	...	12	Next

## Geef aan welke kleur aan de categorie gekoppeld wordt

```
pal <- colorFactor(
  palette = c("purple4", "royalblue", "dodgerblue3", "deepskyblue1", "turquoise2", "paleg
reen3", "darkgoldenrod1", "chocolate1", "red", "brown4"),
  domain = c("0", "1", "2", "3", "4", "5", "6", "7", "8", "9"))
```

Laat per buurt zien op de kaart van Eindhoven hoeveel ritten er geweest zijn. Gebruik daarbij de kleurendefinitie.

```
leaflet() %>%  
  addTiles() %>%  
    addCircleMarkers(lng = frameLocaties$longitude, lat = frameLocaties$latitude  
      , color = pal(frameLocaties$colorGroup)  
      , label= paste(frameLocaties$Buurt, ":", format(frameLocaties$aantal, big.mark =  
        ".", decimal.mark = ","))  
      , radius = 8 + frameLocaties$aantal/10)
```

