

ggplot2 grafiskā sistēma

Didzis Elferts

2016-11-16

Saturs

1	Pamatojums	5
2	Ievads	7
2.1	Dati	7
2.2	Attēlu saglabāšana	8
3	Formas	9
3.1	geom_point()	9
3.2	geom_bar()	12
3.3	geom_col()	14
3.4	geom_line()	16
3.5	geom_path()	16
3.6	geom_boxplot()	19
3.7	geom_count()	19
3.8	geom_histogram()	21
3.9	geom_jitter()	24
4	Skalas	27
4.1	scale_x_continuous() un scale_y_continuous()	27
5	Koordinātu sistēmas	35
6	Attēlu sadalīšana	37
7	Attēla noformēšana	39
7.1	Definētās attēla tēmas	39

Nodaļa 1

Pamatojums

Programmā R ir iespējams veidot attēlus izmantojot dažādas attēlu veidošanas sistēmas, no kurām viena ir ggplot2 (Wickham (2009)) . Šīs sistēmas pamatā ir attēlu veidošanas gramatika.



Šī grāmata ir licenzēta atbilstoši Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License nosacījumiem.

Nodaļa 2

Ievads

Veidojot attēlus ggplot2 sistēmā, var izmantot divas funkcijas: `qplot()` vai `ggplot()`. Pirmā funkcija ir paredzēta ātrai attēla veidošanai, bet tai ir mazākas iespējas tikt modificētai, tāpēc šīs grāmatas ietvaros tā netiks izmantota. Šajā grāmatā visi piemēri balstīsies uz funkciju `ggplot()`. Šai funkcijai kā pirmais arguments ir jānorāda datu tabula/rāmis (var arī nenorādīt, bet tad tā jānorāda kā arguments `geom_...()` vai `stat_...()` funkcijās). Šis objekts būtu jābūt tādā, ko programma R uztver kā data frame. Nākamie argumenti ir x un y vērtības, kā arī citi mainīgie, ja no tiem ir jābūt atkarīgai krāsai, formai, utt. Visi mainīgie tiek norādīti funkcijā `aes()`. Ir jāatceras, ka `aes()` jānorāda tikai mainīgā (kolonnas) nosaukums, neliekot to pēdīnās, kā arī neizmantojot pieraktu `dati$mainigais`. Pieraksts ar `$` zīmi var radīt dīvainu (nepareizu rezultātu). Funkcijā `aes()` nav obligāti rakstīt `x=...` un `y=...` - trūkstot šiem argumentiem, pirmais mainīgais tiks uztverts kā x, bet otrais kā y.

2.1 Dati

ggplot2 sistēmas iespēju apskatīšanai izmantoti R iekļautie datu objekti CO2 un mpg (iekļauts paketē ggplot2). CO2 ir eksperimenta rezultāti par sala tolerenci sugai *Echinochloa crus-galli*. Datu objektā ir piecas kolonnas: (1) Plant - augs identifikators; (2) Type - augs izcelsmes vieta; (3) Treatment - eksperimenta apstākļi (divas kategorijas); (4) conc - vides CO2 koncentrācija; (5) uptake - uzņemtā CO2 apjoms.

```
data(CO2)
head(CO2)
```

```
##   Plant   Type Treatment conc uptake
## 1   Qn1 Quebec nonchilled   95   16.0
## 2   Qn1 Quebec nonchilled  175   30.4
## 3   Qn1 Quebec nonchilled  250   34.8
## 4   Qn1 Quebec nonchilled  350   37.2
## 5   Qn1 Quebec nonchilled  500   35.3
## 6   Qn1 Quebec nonchilled  675   39.2
```

Objektā mpg ir informācija par degvielas patēriņu, kas parādīta 11 kolonnās: (1) manufacturer - ražotājs; (2) model - automašīnas modelis; (3) displ - dzinēja tilpums; (4) year - ražošanas gads; (5) cyl - cilindru skaits; (6) trans - transmisijas tips; (7) drv - velkošie riteņi; (8) cty - jūdžu skaits/gallons pilsētā; (9) hwy - ūdžu skaits/gallons uz šosejas; (10) fl - degvielas veids; (11) class - automašīnas veids.

```
library(ggplot2)
data(mpg)
head(mpg)
```

```
## # A tibble: 6 × 11
##   manufacturer model displ  year   cyl    trans  drv   cty   hwy   fl
##         <chr> <chr> <dbl> <int> <int>    <chr> <chr> <int> <int> <chr>
## 1      audi   a4    1.8  1999     4  auto(l5)   f    18    29    p
## 2      audi   a4    1.8  1999     4 manual(m5)   f    21    29    p
## 3      audi   a4    2.0  2008     4 manual(m6)   f    20    31    p
## 4      audi   a4    2.0  2008     4  auto(av)   f    21    30    p
## 5      audi   a4    2.8  1999     6  auto(l5)   f    16    26    p
## 6      audi   a4    2.8  1999     6 manual(m5)   f    18    26    p
## # ... with 1 more variables: class <chr>
```

2.2 Attēlu saglabāšana

ggplot2 sistēmā izveidoto attēlu saglabāšanu var veikt ar funkciju `ggsave()`, kuru izpilda pēc attēlā izveidošanas un kurā kā pamatarguments ir jānorāda vēlamais attēlā nosaukums ar nepieciešamo paplašinājumu (png, eps, ps, tex, pdf, jpeg, tiff, bmp, svg, wmf (tikai uz windows)). Papildus var norādīt attēla izmēru (`width=` un `height=`). Pēc noklusējuma izmērs ir collās, bet var mainīt uz `cm` vai `mm` ar argumentu `units=`. Arguments `dpi=` rastra tipa attēliem maina izšķirtspēju.

```
library(ggplot2)
data(CO2)
ggplot(CO2,aes(conc,uptake))+geom_point()
ggsave("Attels_1.png",width = 10,height = 6, units="cm")
```


Nodaļa 3

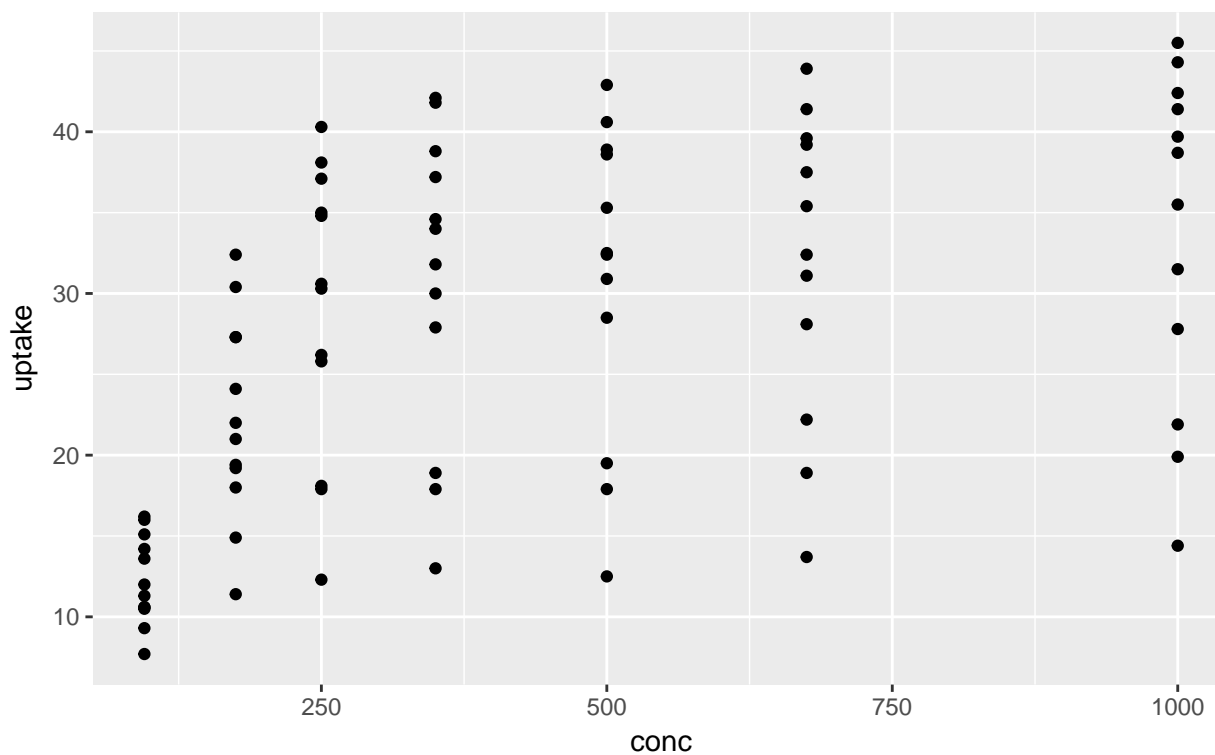
Formas

ggplot2 sistēmā ir iespējams vienus un tos pašus datus attēlot dažādos veidos, izvēloties atbilstošo datu attēlošanas formu jeb `geom_...()`. Vairumā gadījumu ir jānorāda x un y vērtības, bet atsevišķos gadījumos ir nepieciešami arī papildus mainīgie, vai arī nepieciešamas tikai x vērtības (piemēram, histogrammai).

3.1 `geom_point()`

Ar `geom_point()` ir iespējams veidot izkliedes attēlus (scatterplot) (3.1 attēls).

```
library(ggplot2)
ggplot(CO2, aes(conc, uptake)) + geom_point()
```



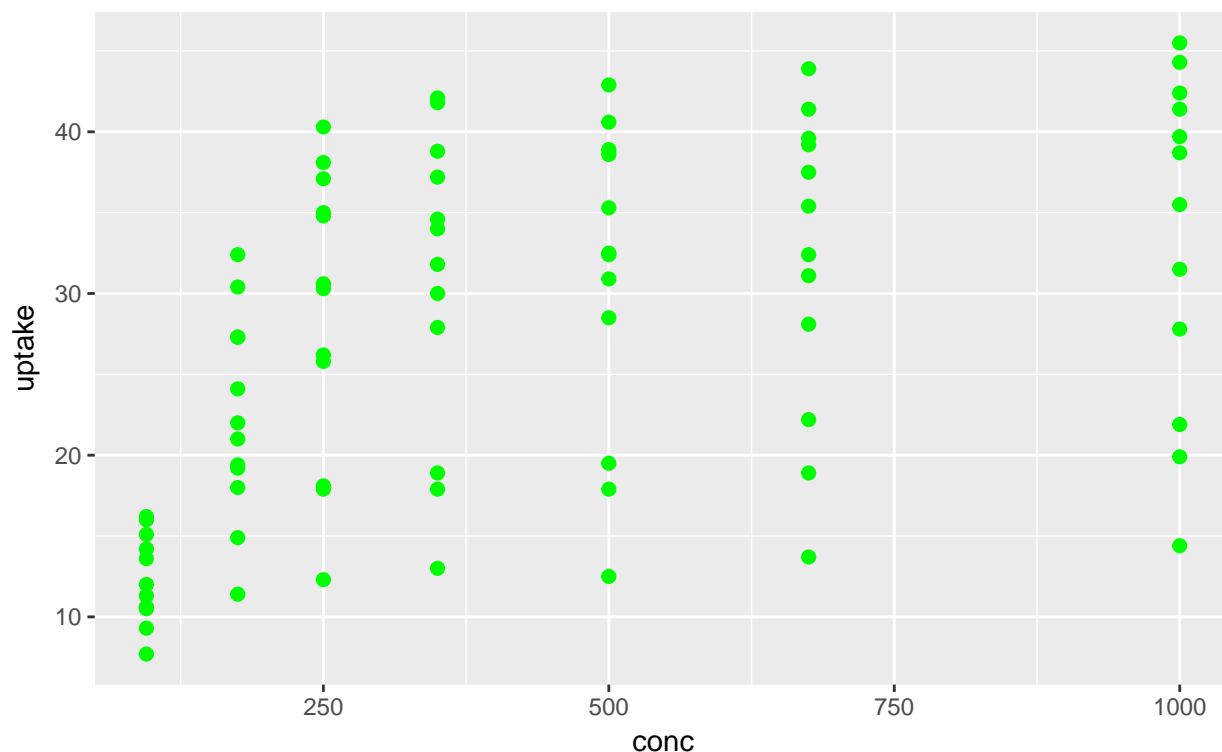
Att. 3.1: Izkliedes attēla piemērs

Punktiem ir iespējams mainīt krāsu (`color=`), formu (`shape=`), lielumu (`size=`) un caurspīdīgumu (`alpha=`). Mainot šos parametrus ir jānolemj pēc kādiem principiem tas notiks - parametrs būs vienāds visiem punktiem, vai arī tas mainīsies atkarībā no kāda cita mainīgā datus.

Ja parametram ir jābūt vienādam visiem punktiem, tad tas ir jānorāda ārpus funkcijas `aes()` pašā `geom_...` vai `ggplot()` funkcijā. Toties, ja parametram ir jāmainās atkarībā no mainīgā, tad tas obligāti jāliek funkcijā `aes()`.

Šajā piemērā punktu krāsa un lielums ir mainīts visiem punktiem uzreiz (3.2 attēls). Krāsu var norādīt kā tās angļu nosaukumu (tos var apskatīt ar funkciju `colors()`) vai arī izmantojot heksadecimālo kodu.

```
ggplot(CO2,aes(conc,uptake))+geom_point(color="green",size=2)
```



Att. 3.2: Izklīdes attēls, kurā krāsa un lielums visiem punktiem vienāds

Ja arguments `color=` atrodas `aes()` iekavās un tas ir atkarīgs no kāda mainīgā, tad punktu krāsa mainīsies atbilstoši vērtībām, kā arī parādīsies atbilstošā lēģenda. Krāsu maiņa un lēģendas veids ir atkarīgs no tā, kāda veida mainīgais ir izmantots. Ja krāsa ir atkarīga no kategorijas mainīgā, tad krāsas mainīsies diskrēti (3.3 attēls).

```
ggplot(CO2,aes(conc,uptake,color=Type))+geom_point()
```

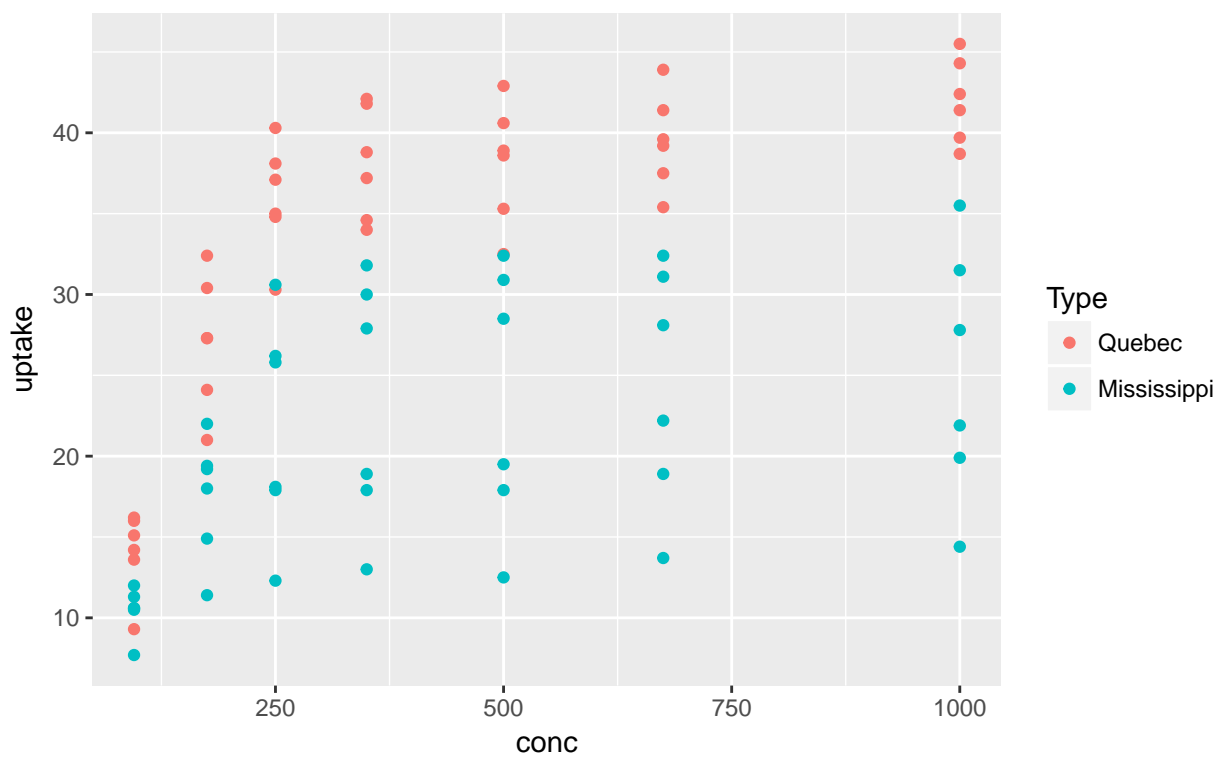
Toties norādāt kā mainīgo tādu, kas ir skaitlisks, krāsa mainīsies kā gradients (3.4 attēls).

```
ggplot(CO2,aes(conc,uptake,color=uptake))+geom_point()
```

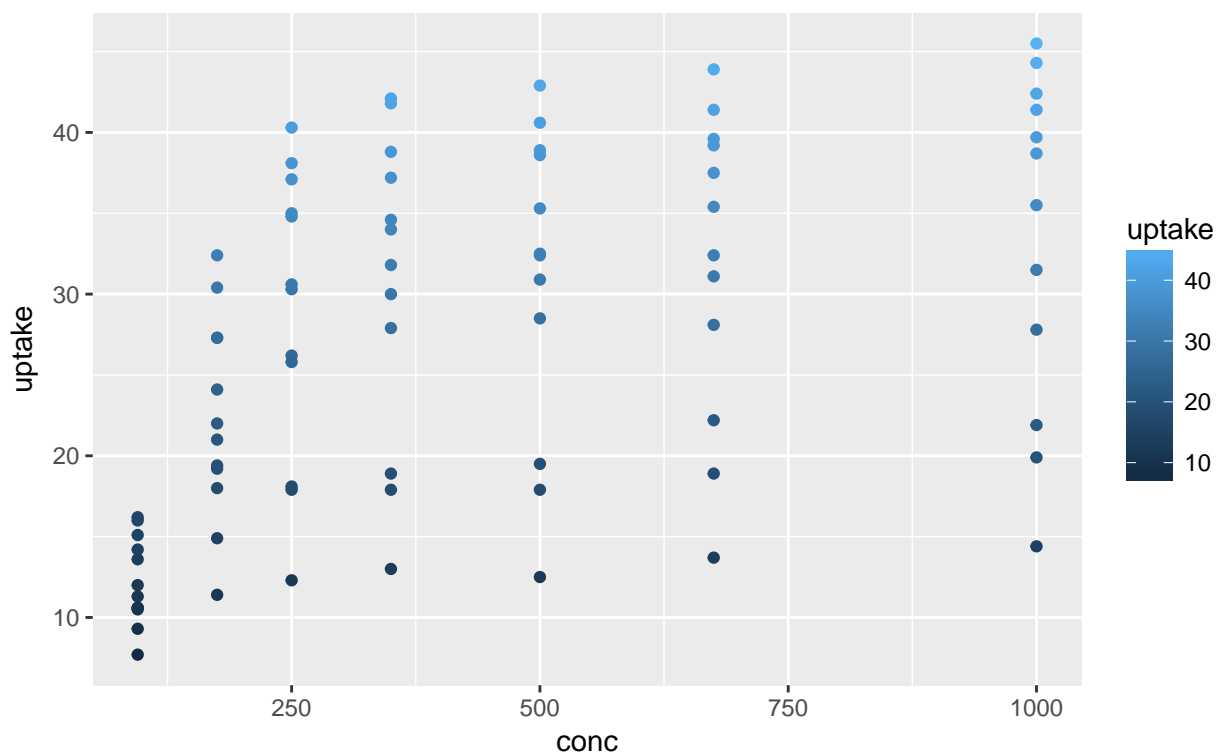
Punktu forma var mainīties tikai atkarībā no kategorijas mainīgā (3.5 attēls).

```
ggplot(CO2,aes(conc,uptake,shape=Type))+geom_point()
```

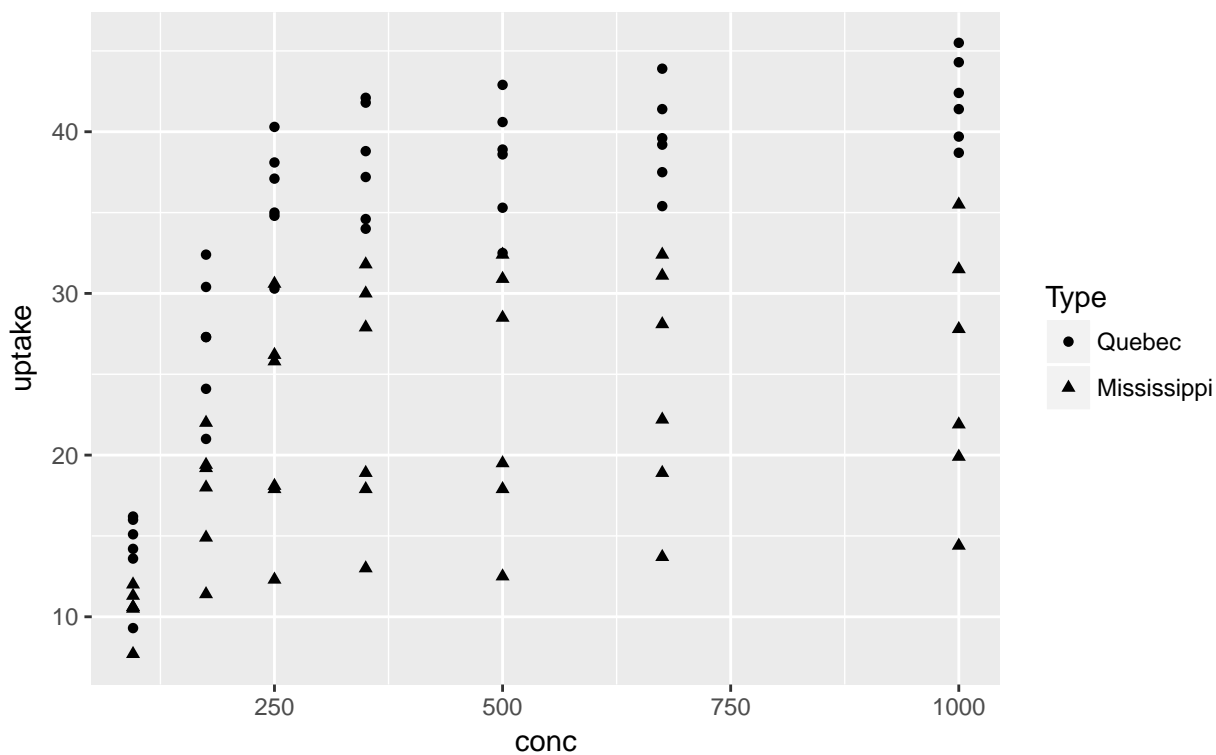
Ir iespējams panākt arī, ka, piemēram, punktu forma mainās atkarībā no viena mainīgā, bet krāsa atkarībā no cita mainīgā. Šajā gadījumā parādīsies arī divas lēģendas (3.6 attēls).



Att. 3.3: Izklides attēls, kurā krāsa ir atkarīga no kategorijas mainīgā



Att. 3.4: Izklides attēls, kurā krāsa ir atkarīga no skaitliska mainīgā



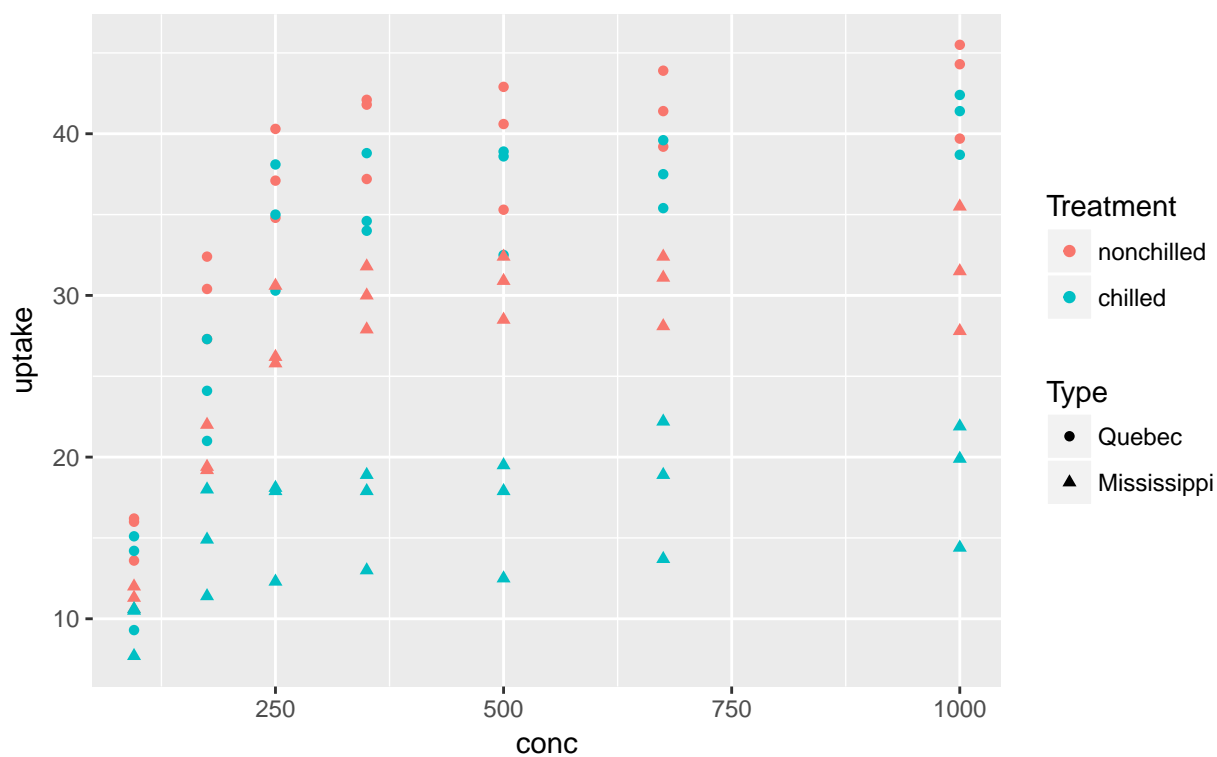
Att. 3.5: Izkliedes attēls, kurā forma ir atkarīga no kategorijas mainīgā

```
ggplot(CO2, aes(conc, uptake, shape=Type, color=Treatment)) + geom_point()
```

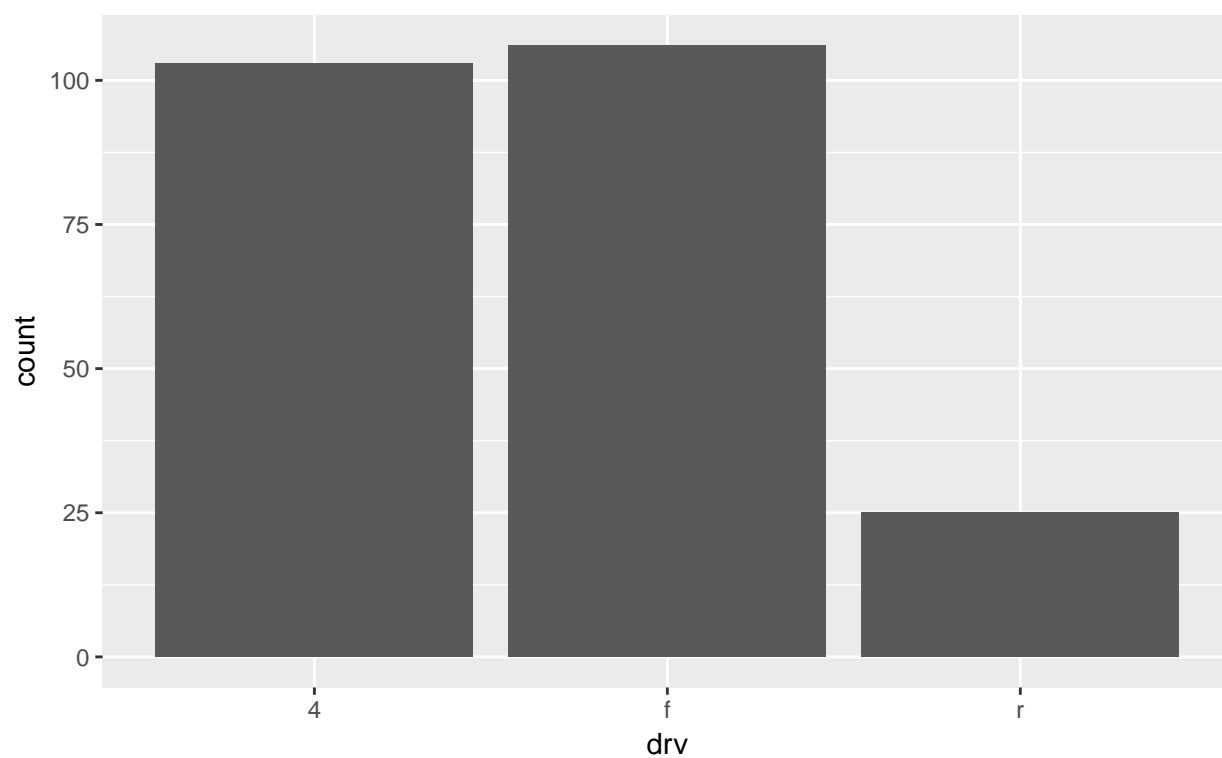
3.2 geom_bar()

Stabiņu attēlus veido ar funkciju `geom_bar()`. Šai funkcijai `aes()` ir jānorāda tikai x vērtības (diskrētas), jo novērojumu skaits katrā klasē tiek saskaitīts automātiski (`geom_bar()` balstās un `stat_count()`) (?? attēls).

```
ggplot(mpg, aes(drv)) + geom_bar()
```

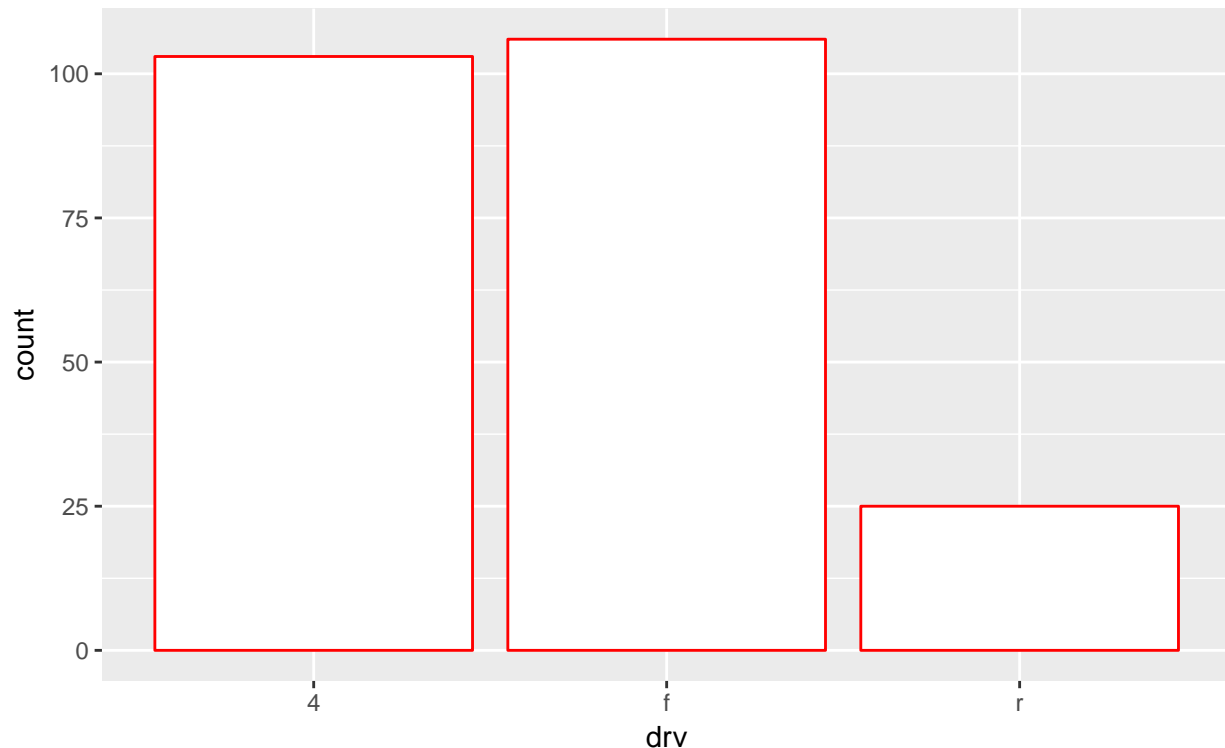


Att. 3.6: Izklīdes attēls, kurā forma un krāsa ir atkarīga no dažādiem kategorijas mainīgajiem



Stabiņu attēlā katram stabiņam ir iespējams mainīt krāsu (`color=`) un aizpildījumu (`fill=`). Arguments `color=` nosaka līnijas krāsu apkārt katram no stabiņiem, bet `fill=` nosaka paša stabiņa krāsu (aizpildījumu) (3.7 attēls).

```
ggplot(mpg,aes(drv))+geom_bar(fill="white",color="red")
```



Att. 3.7: Stabiņu attēls, kurā stabiņu krāsa un aizpildījums visiem vienāds

Padarot aizpildījumu atkarīgu no kāda kategorijas mainīgā, izveidojas stabiņu attēls, kur pie katras x mainīgā kategorijas, stabiņš ir sadalīts pa daļām balstoties uz jauno mainīgo (3.8 attēls).

```
ggplot(mpg,aes(drv,fill=factor(cyl)))+geom_bar()
```

Pieliekot argumentu `position="dodge"`, var panākt, ka pie katras x kategorijas stabiņi ir viens otram blakus, nevis viens virs otra (3.9 attēls).

```
ggplot(mpg,aes(drv,fill=factor(cyl)))+geom_bar(position="dodge")
```

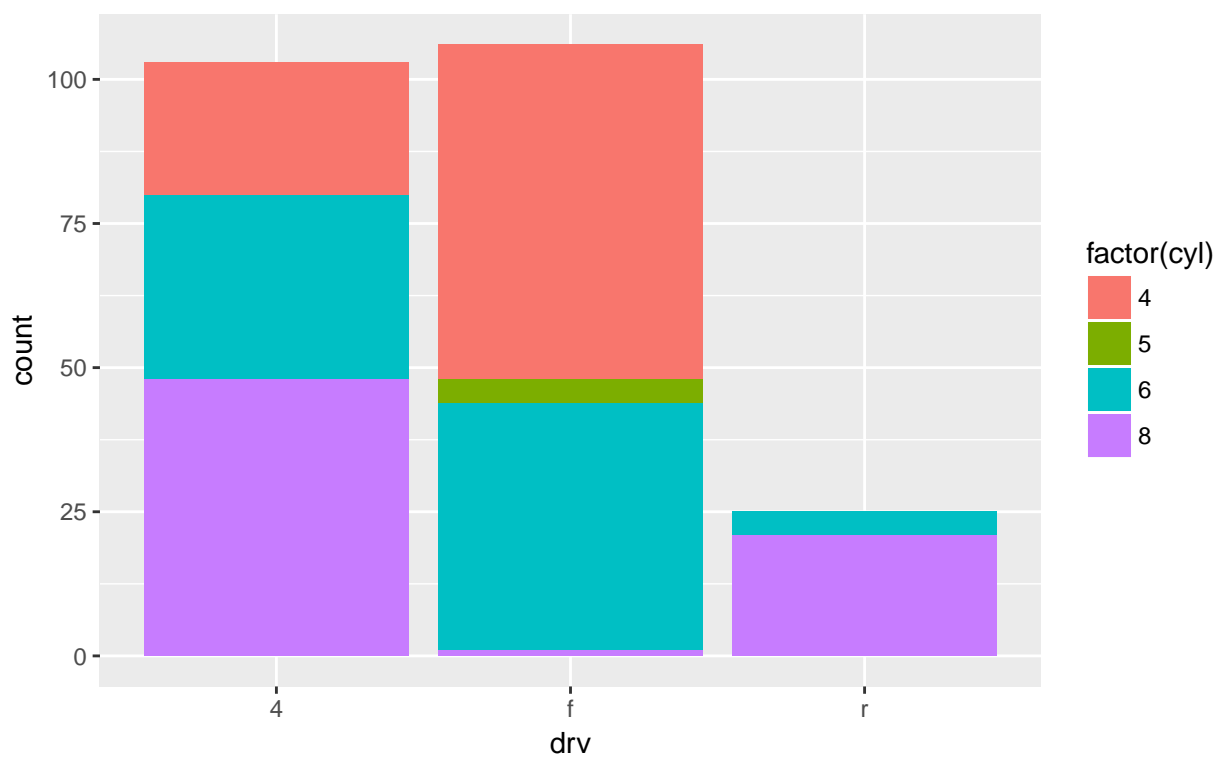
3.3 geom_col()

Gadījumos, kad dati ir jau apkopoti un ir nepieciešams izveidot stabiņu attēlu, tad labāk izmantot `geom_col()`, kam jānorāda gan x vērtības, gan arī atbilstošās y vērtības (skaiti) (3.10 attēls).

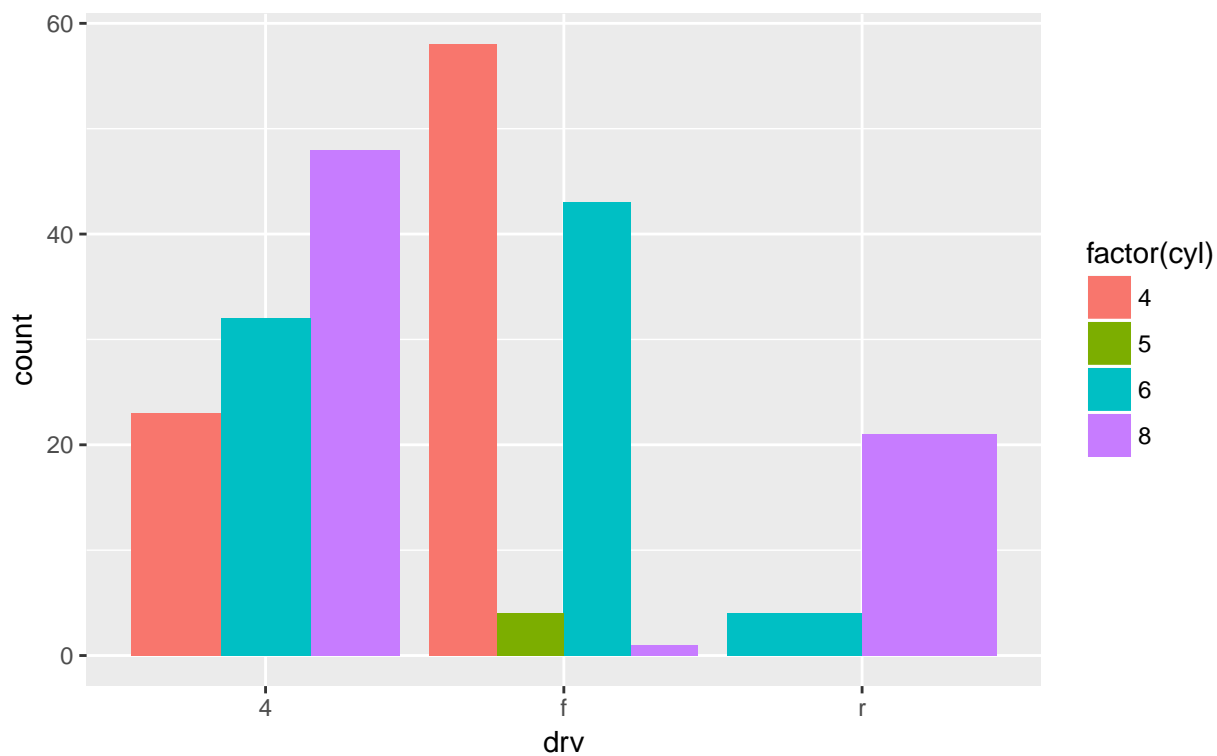
```
dati <- data.frame(Dzimums=c("S","V"),Skaitis=c(23,45))
dati
```

```
##   Dzimums Skaitis
## 1      S      23
## 2      V      45
```

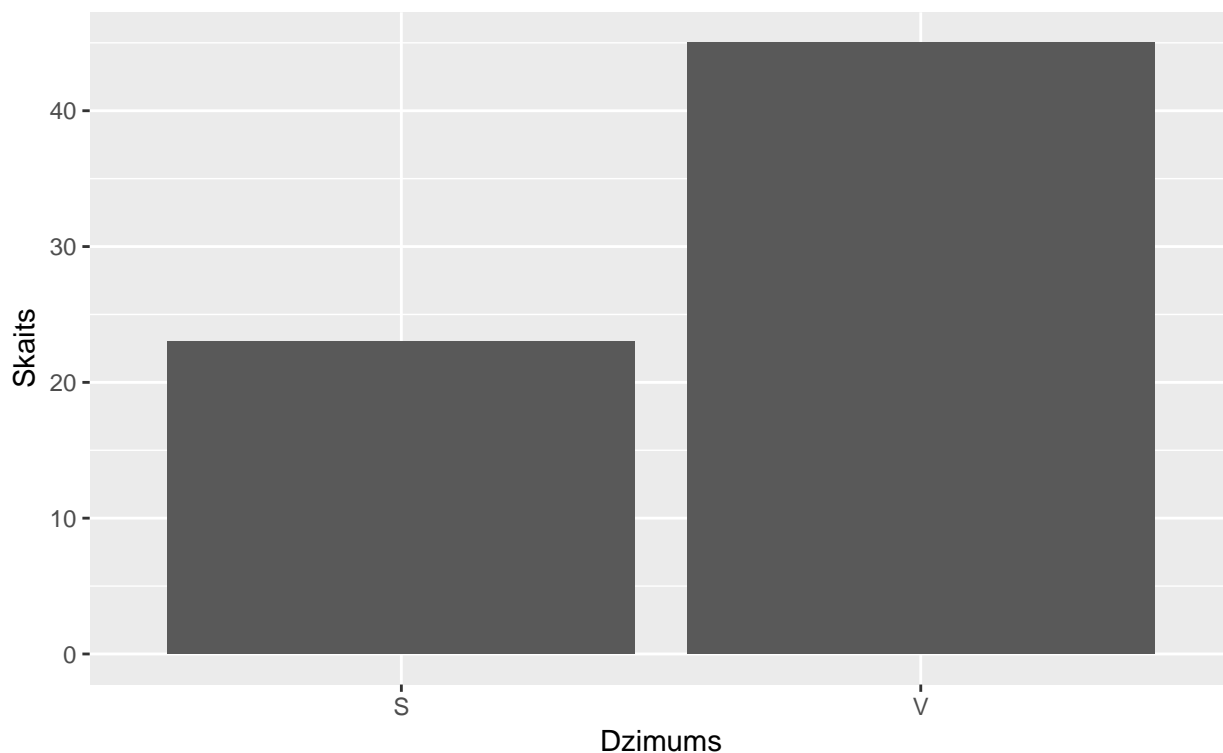
```
ggplot(dati,aes(Dzimums,Skaitis))+geom_col()
```



Att. 3.8: Stabiņu attēls, kurā stabiņu aizpildījums atkarīgs no mainīgā



Att. 3.9: Stabiņu attēls, kurā stabiņu aizpildījums atkarīgs no mainīgā



Att. 3.10: Stabiņu attēls, kurā skaiti jau doti tabulā

3.4 geom_line()

Datu punktu savienošanai ar līniju var izmantot `geom_line()`, kas savieno punktus no mazākās x vērtības līdz lielākajai x vērtībai (3.11 attēls).

```
ggplot(mpg, aes(cty, hwy)) + geom_line()
```

Līnijām ir iespējams mainīt tās platumu (`size=`), krāsu (`color=`) un līnijas veidu (`linetype=`) (3.12 attēls).

```
ggplot(mpg, aes(cty, hwy)) + geom_line(color="red", size=1.5, linetype=2)
```

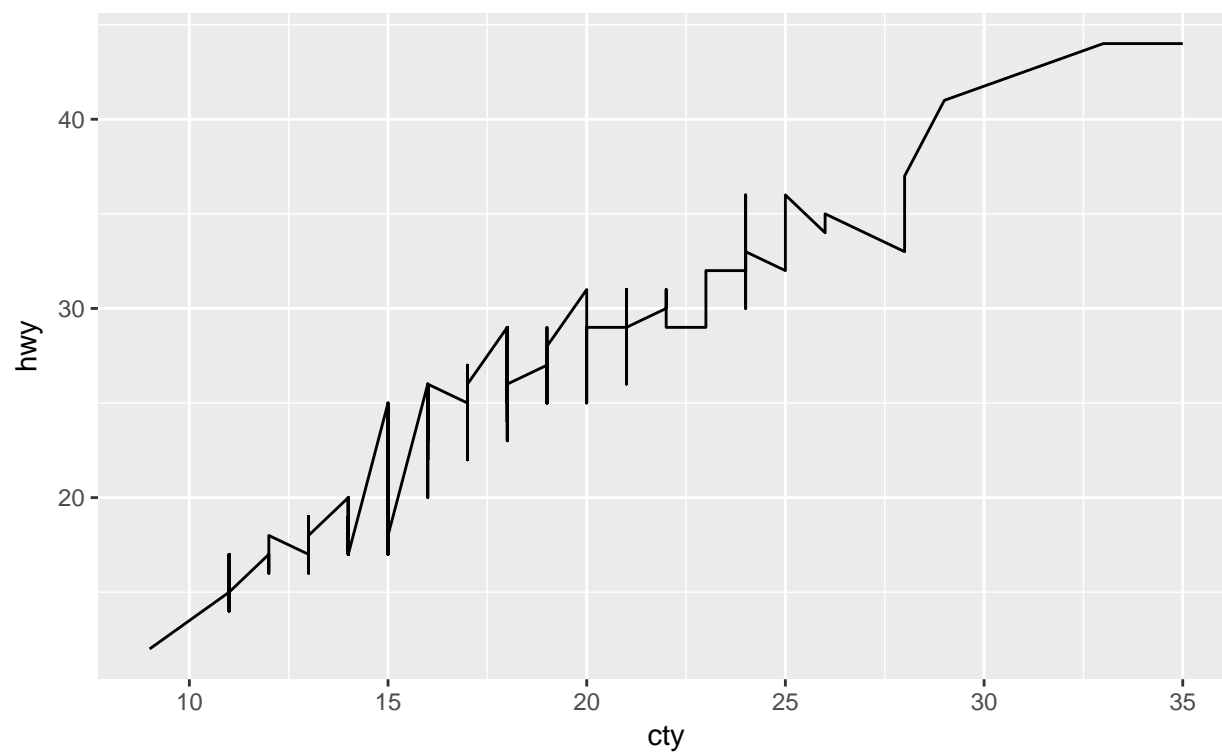
Ja kāds no līnijas parametriem ir atkarīgs no diskrēta trešā mainīgā, tad parādīsies tik daudz līnijas, cik mainīgajam ir līmeņi (3.13 attēls).

```
ggplot(mpg, aes(cty, hwy, color=drv)) + geom_line()
```

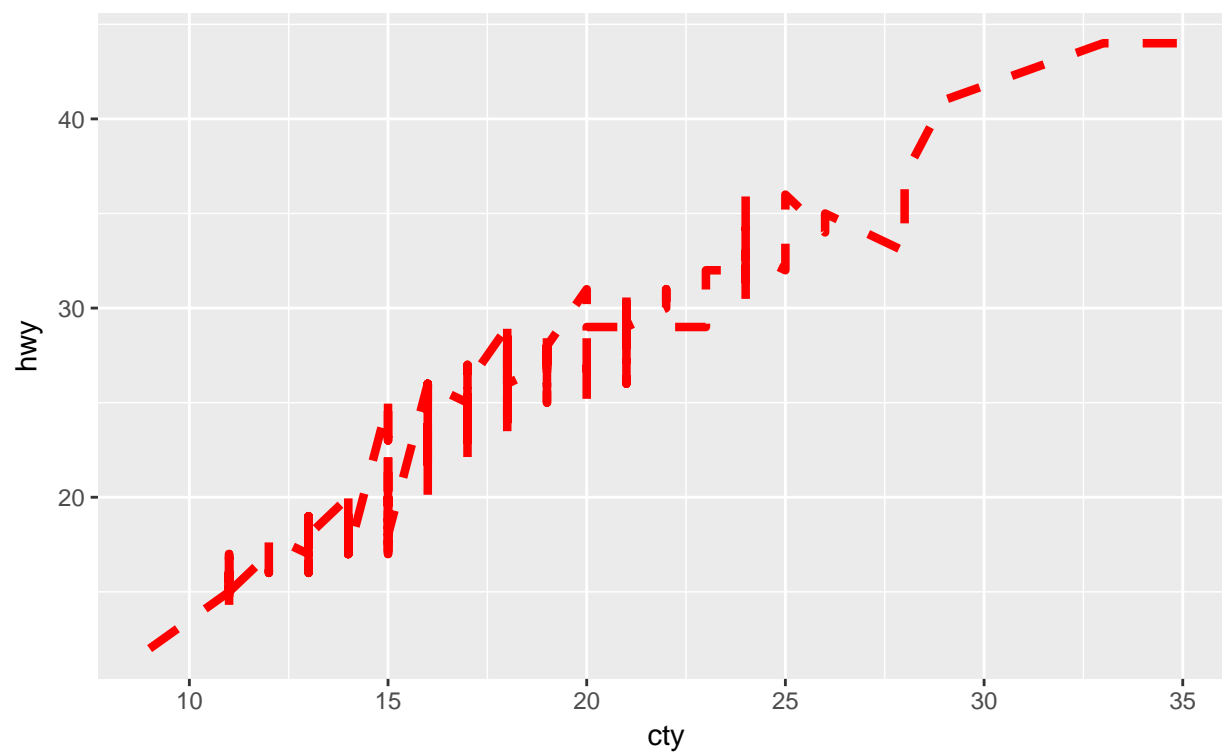
3.5 geom_path()

Līniju diagrammai līdzīgs ir arī `geom_path()`, bet šajā gadījumā punkti tiek savienoti tādā secībā, kādā tie parādās datu tabulā (3.14 attēls). `geom_path()` ir īpaši noderīgs gadījumos, ja jāsavieno x un y koordinātes pārvietošanās ceļam.

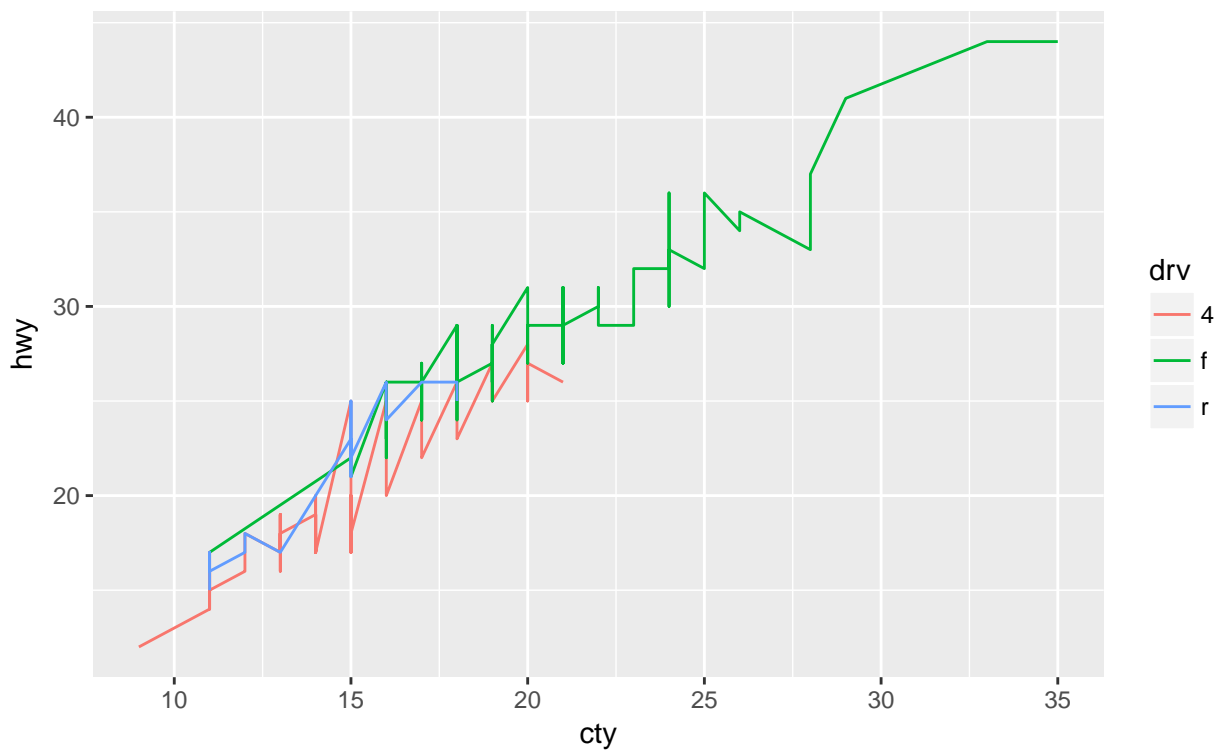
```
ggplot(mpg, aes(cty, hwy)) + geom_path()
```

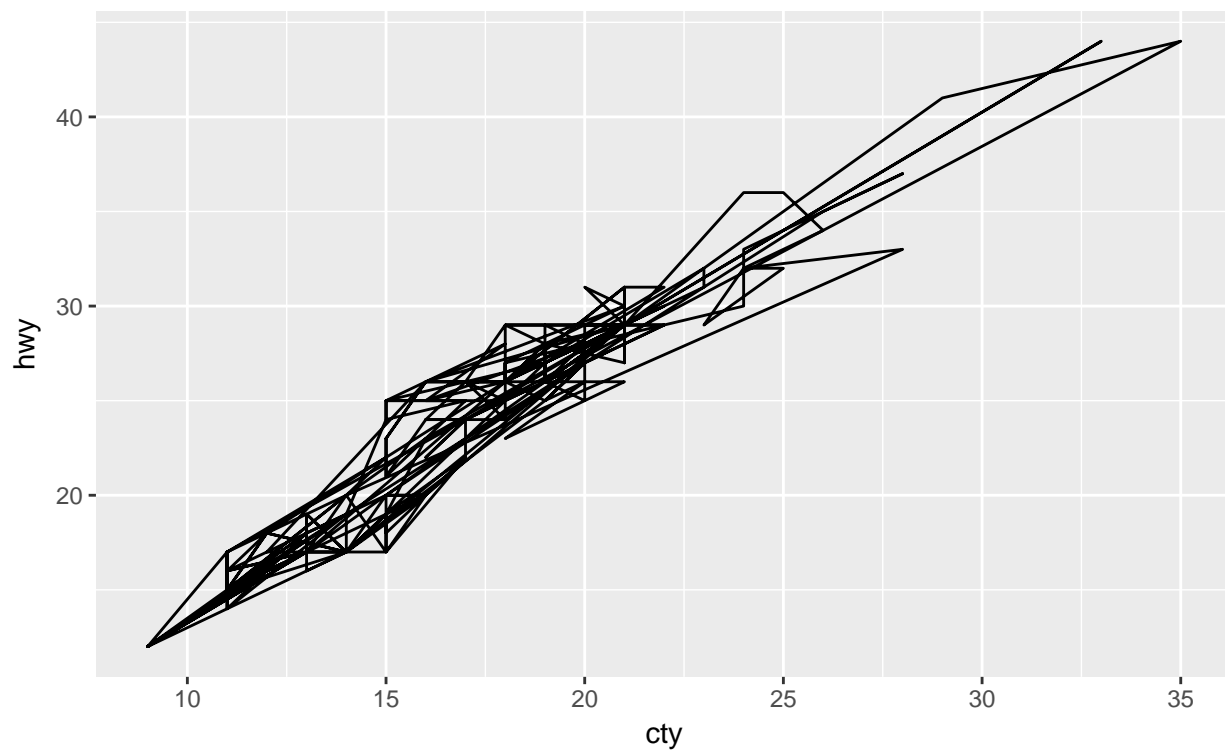
Att. 3.11: Līniju diagrammas piemērs



Att. 3.12: Līnija ar izmainītiem parametriem



Att. 3.13: Līnija, kuras krāsa atkarīga no mainīgā

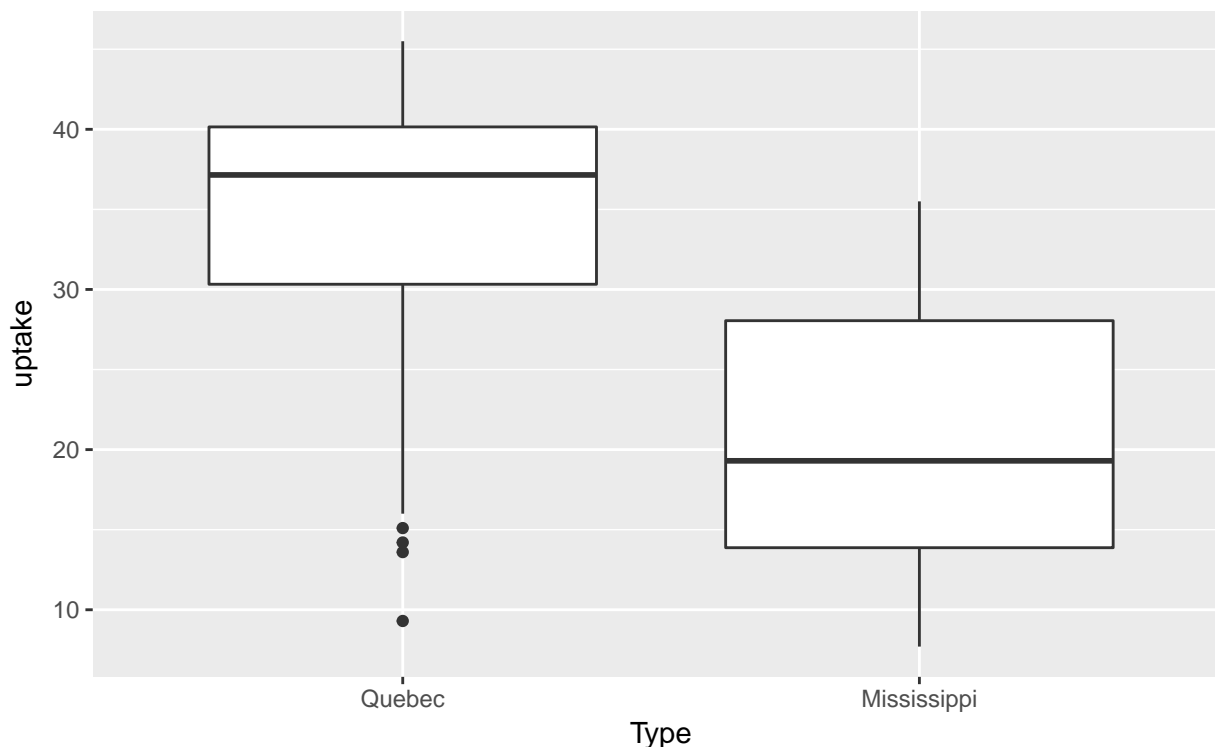


Att. 3.14: Punkti, kas savienoti ar līniju to izvietojuma secībā

3.6 geom_boxplot()

Vērtībamplitūdas diagrammas veidošanai izmanto `geom_boxplot()`. Šim attēla veida x vērtībām ir jābūt kvalitatīviem datiem, vai arī skaitliskiem datiem, kas pārvērsti par faktoru. y vērtībām obligāti ir jābūt skaitliskām (3.15 attēls).

```
ggplot(CO2,aes(Type,uptake))+geom_boxplot()
```



Att. 3.15: Vērtībamplitūdas diagrammas piemērs

Līdzīgi kā stabiņu attēlam vērtībamplitūdas diagrammā var mainīt līniju un punktu krāsu (`color=`) vai arī “kastītes” aizpildījumu (`fill=`) (3.16 attēls).

```
ggplot(CO2,aes(Type,uptake))+geom_boxplot(color="green",fill="red")
```

Izlēcēju (neraksturīgo vērtību) punktu krāsu, formu un izmēru var mainīt arī atsevišķi, izmantojot argumentus `outlier.color=`, `outlier.shape=` un `outlier.size=` (3.17 attēls).

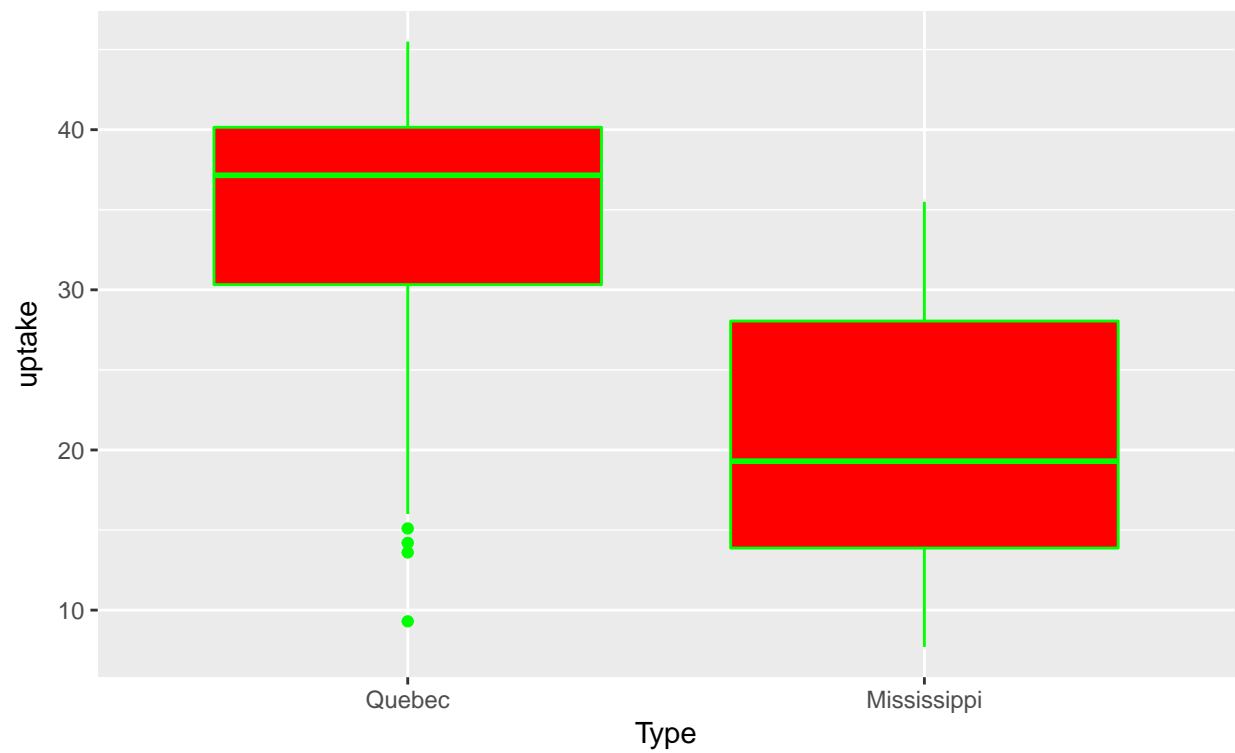
```
ggplot(CO2,aes(Type,uptake))+geom_boxplot(outlier.color = "red",outlier.shape = 13,outlier.size = 3)
```

Ja arguments `fill=` atrodas funkcijas `aes()` iekavās un tas ir atkarīgs no kāda mainīgā, tad pie katras atbilstošās x vērtības, vērtībamplitūdas diagramma tiek sadalīta tik daļās, cik līmeņi ir papildus mainīgajam, kā arī parādās atbilstošā lēģenda ar izmantotajām aizpildījuma krāsām (3.18 attēls).

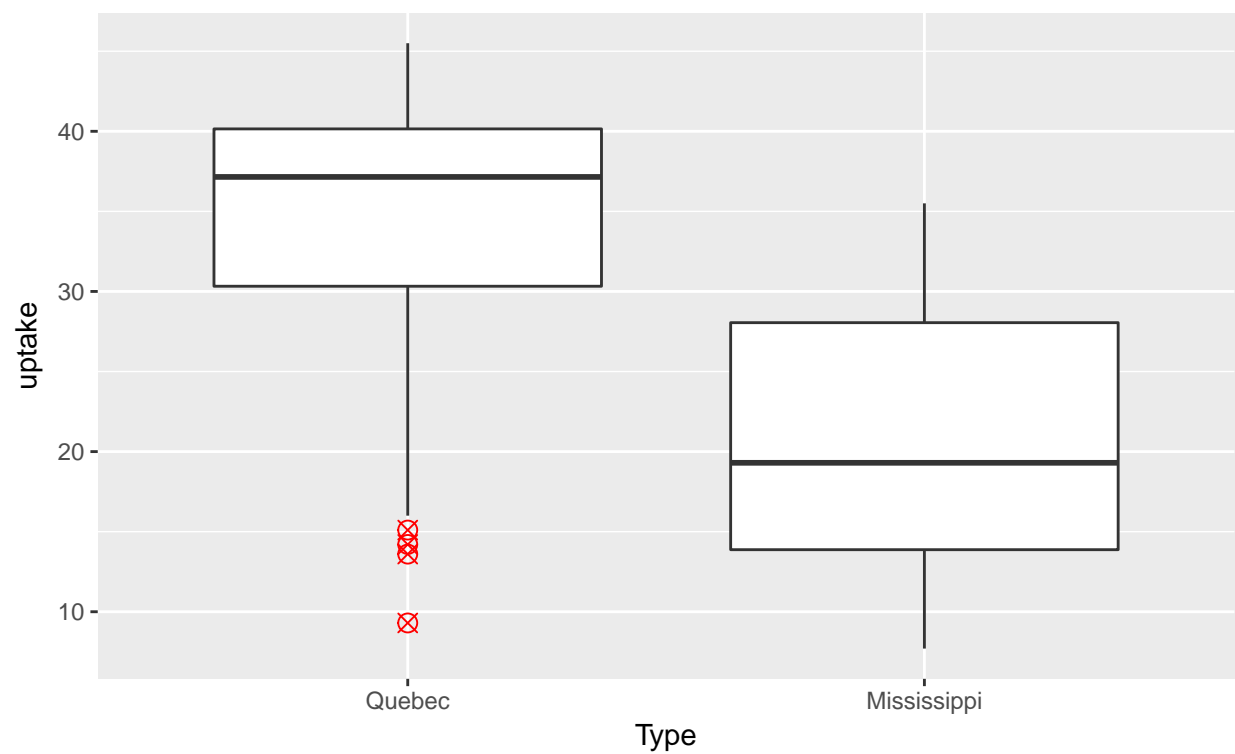
```
ggplot(CO2,aes(Type,uptake,fill=Treatment))+geom_boxplot()
```

3.7 geom_count()

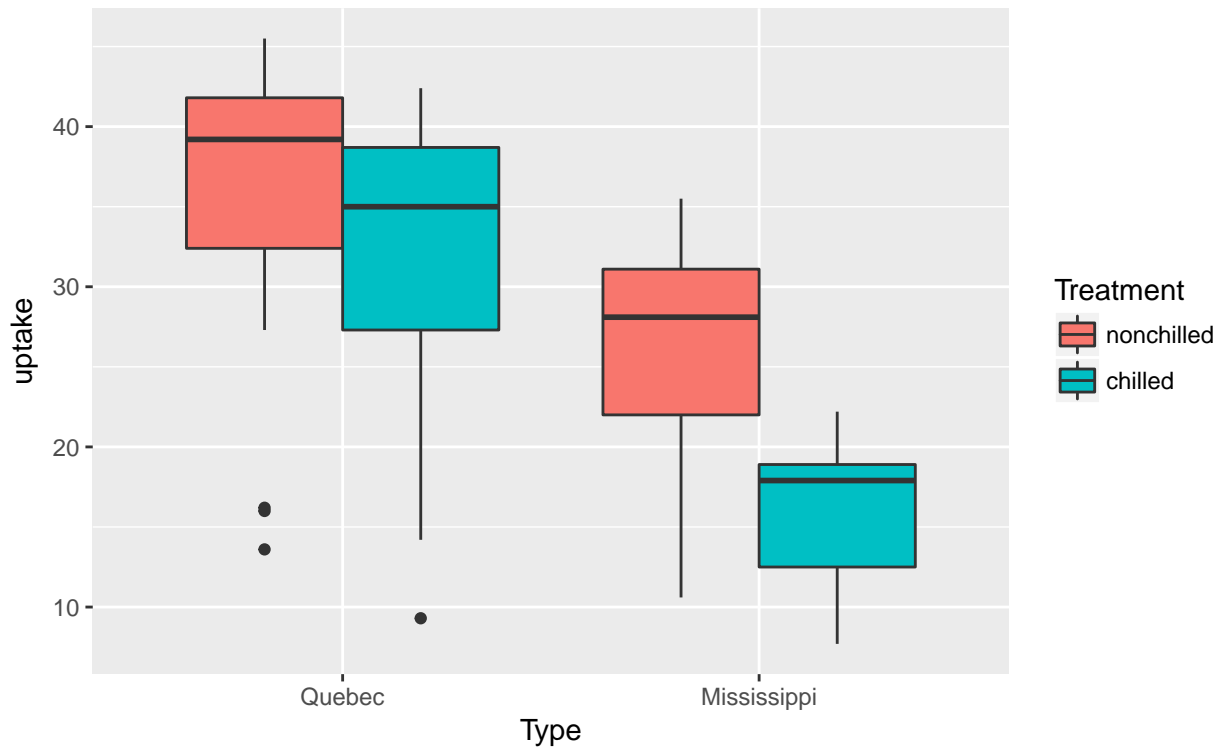
Gadījumos, kad nepieciešams attēlot izkliedes diagrammu, bet ir vērojama punktu pārklāšanās (pie vienādām x un y vērtībām ir vairāki novērojumi), var izmantot `geom_count()`, kas parāda cik daudz novērojumu ir



Att. 3.16: Vērtībamplitūdas diagramma ar izmainītu līniju un kastītes krāsu



Att. 3.17: Vērtībamplitūdas diagramma ar izmainītu izlēcēju krāsu, formu un izmēru



Att. 3.18: Vērtībampitūdas diagramma, kurā katram faktora līmenim diagramma sadalīta daļās

konkrētajām x un y vērtībām (3.19 attēls).

```
ggplot(mpg, aes(cty, hwy)) + geom_count()
```

Ja `aes()` funkcijā norāda argumentu `size=..prop..`, tad punktu lielums ir parāda proporciju nevis skaitu (3.20 attēls).

```
ggplot(mpg, aes(cty, hwy)) + geom_count(aes(size=..prop..))
```

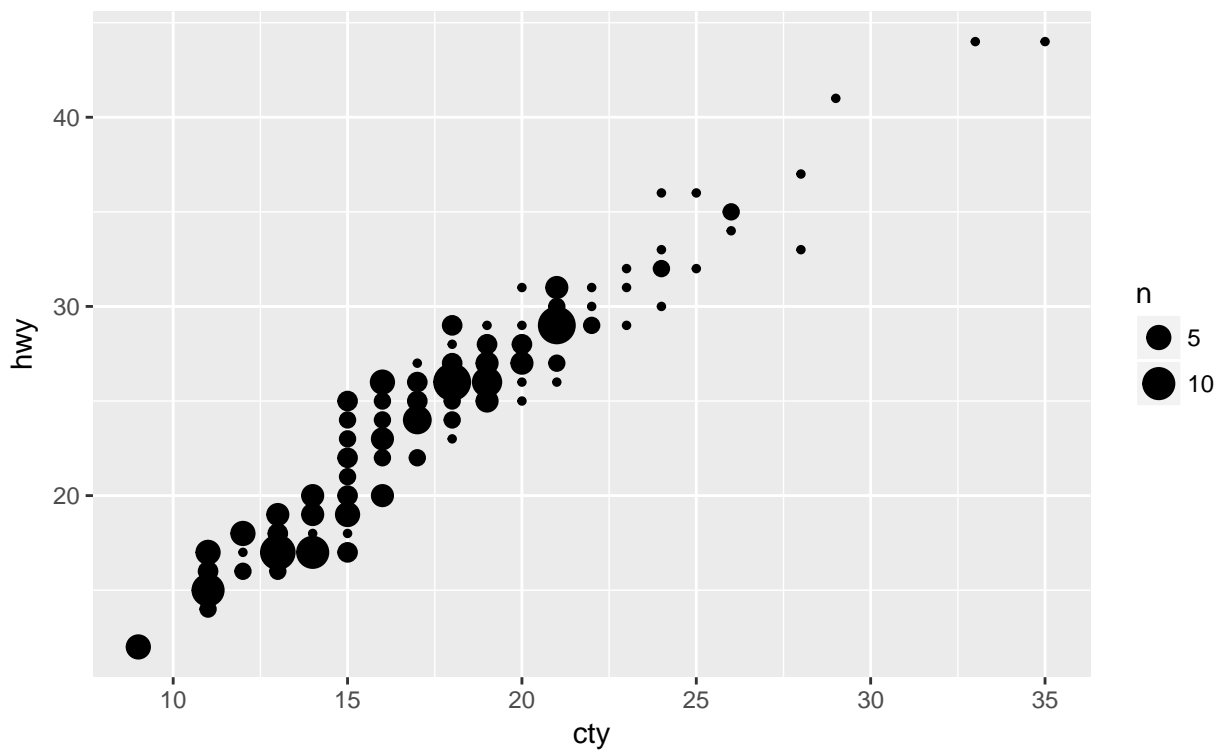
3.8 geom_histogram()

Histogrammas veidošanai izmanto `geom_histogram()`, kam ir nepieciešamas tikai x vērtības. Pēc noklusējuma dati tiek dalīti trīs klasēs (3.21 attēls).

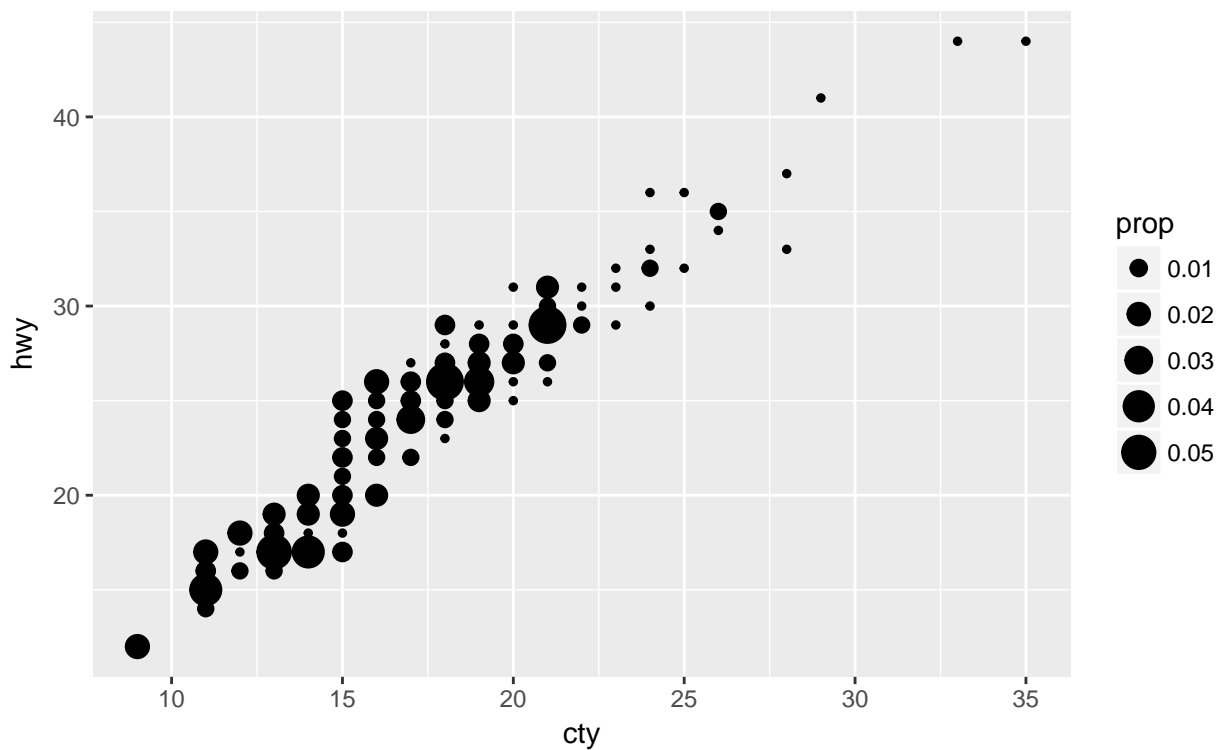
```
ggplot(CO2, aes(uptake)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

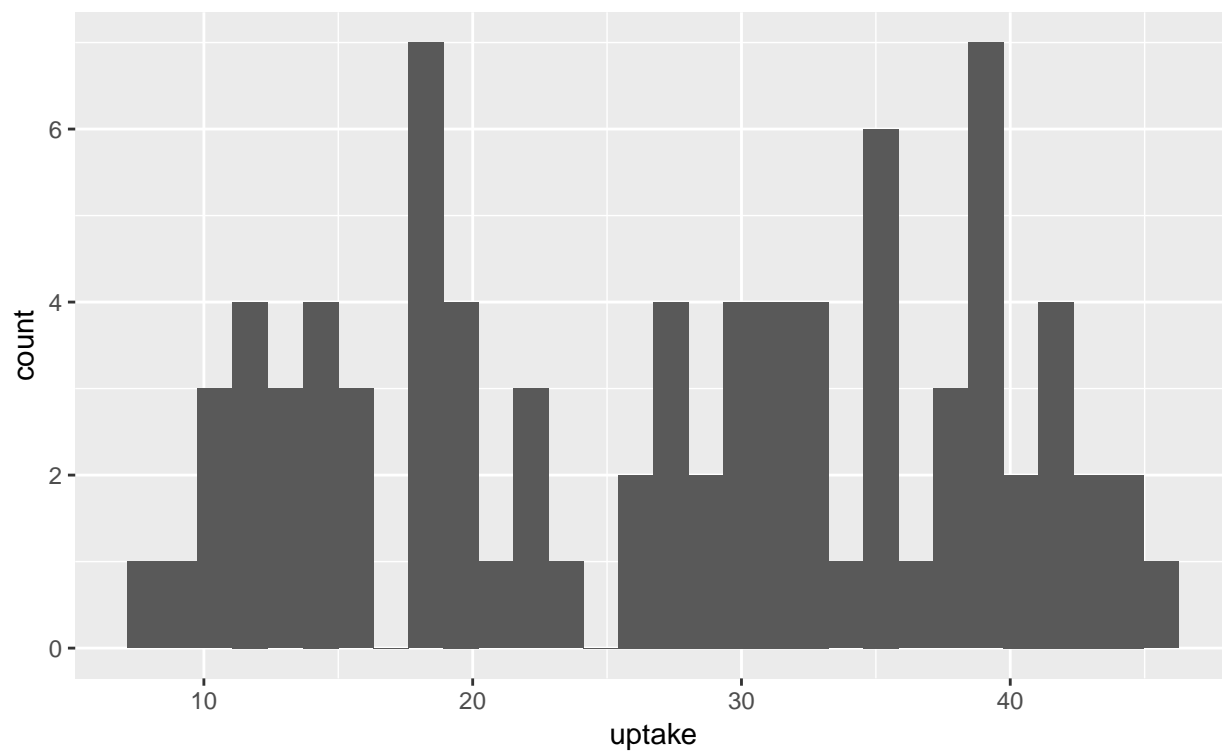
Ar argumentu `binwidth=` ir iespējams mainīt dalījuma klases lielumu, tādēji mainot klašu skaitu un histogrammas izskatu (?? attēls). Var arī norādīt vēlamo klašu skaitu ar argumentu `bins=`.



Att. 3.19: Izkļiedes diagramma, kur punktu lielums atkarīgs no novērojumu skaita

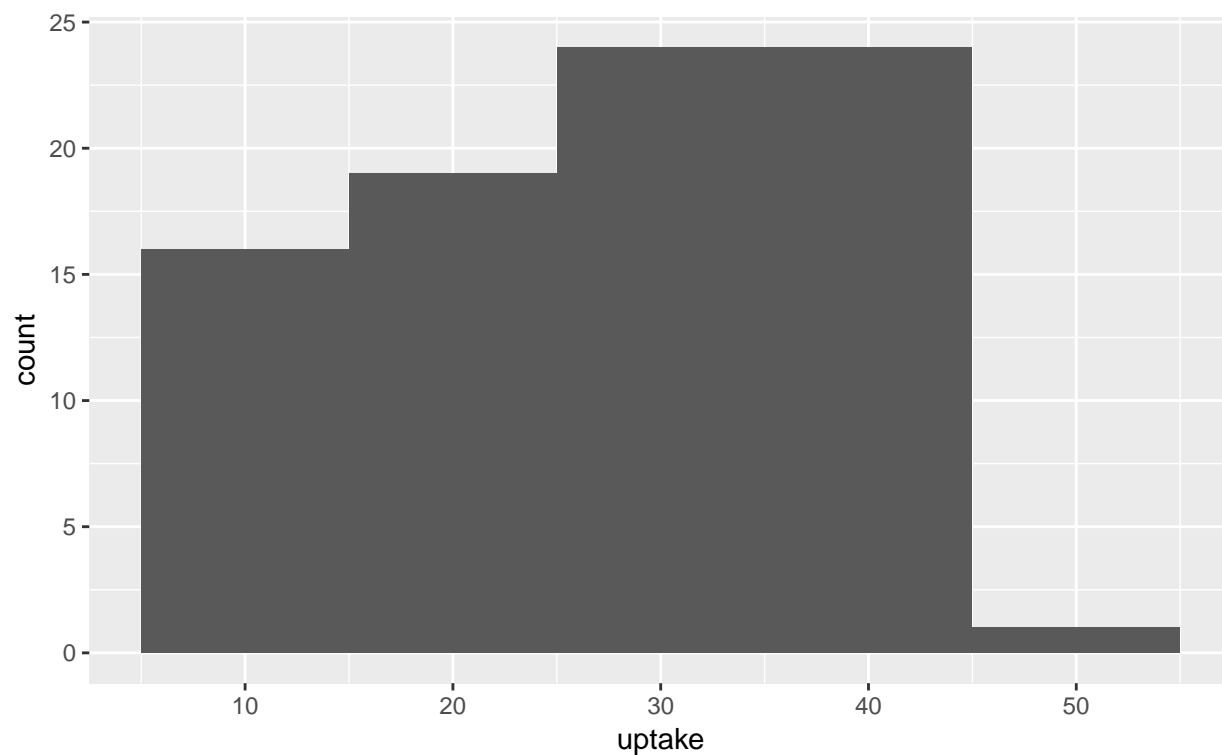


Att. 3.20: Izkļiedes diagramma, kur punktu lielums atbilst novērojumu proporcijai



Att. 3.21: Histogrammas piemērs

```
ggplot(CO2,aes(uptake)) + geom_histogram(binwidth = 10)
```

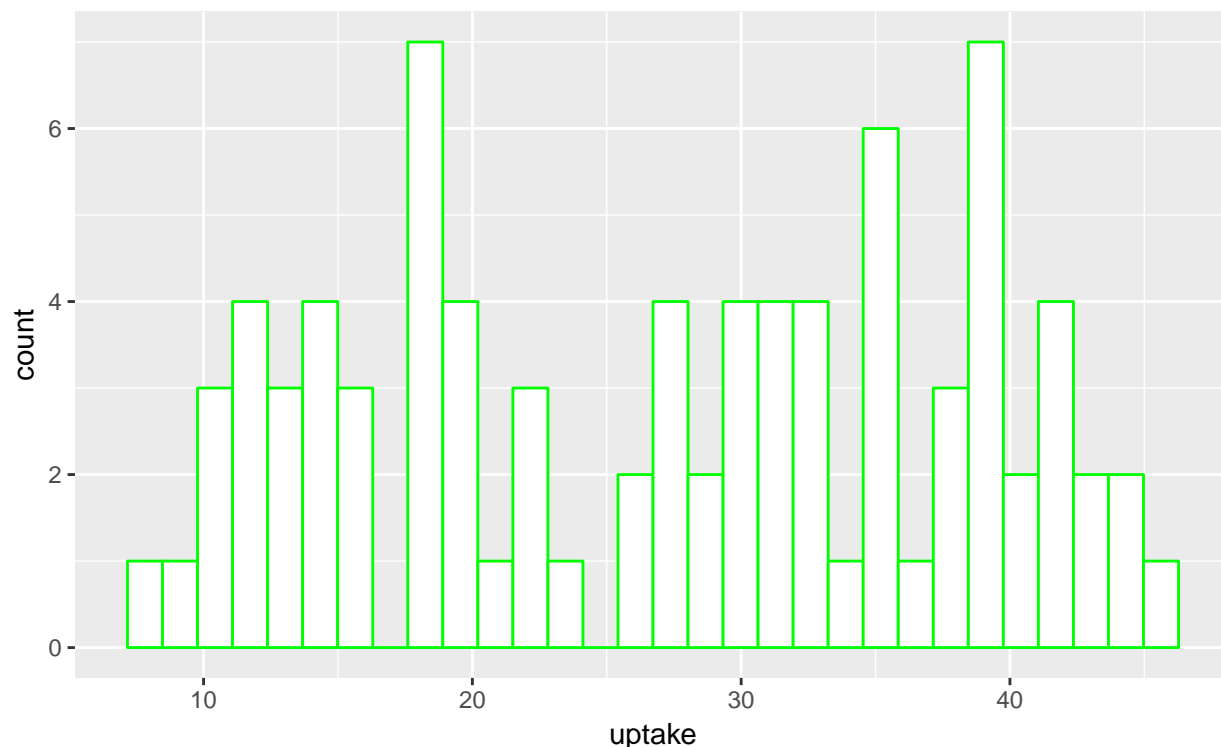


Tā kā histogrammā parādās stabiņi, tad tiem ir iespējams mainīt gan līnijas krāsu (`color=`), gan arī

aizpildījumu (`fill=`) (?? attēls).

```
ggplot(CO2,aes(uptake)) + geom_histogram(color="green",fill="white")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Norādot, ka aizpildījums ir atkarīgs no mainīgā, izveidosies histogramma, kurā katrs stabiņš sadalīts daļās atbilstoši novērojumu skaitam katrā no līmeņiem (3.22 attēls).

```
ggplot(CO2,aes(uptake,fill=Type)) + geom_histogram(binwidth = 5)
```

3.9 geom_jitter()

Gadījumos, kad nepieciešams izveidot izkliedes (punktu) diagrammu, bet vērojamo vērtību pārklāšanās (daudz identisku vērtību), var izmantot `geom_jitter()`, kur punktiem tiek veikta neliela nobīde x vai y (vai abu) ass virzienā, lai novērstu pārklāšanos. Šādu attēlošanas veidu sevišķi ērti izmantot, ja x vērtības ir kategorijas mainīgais, jo tad izkliede notiek tikai x ass virzienā, bet y ass virzienā redzamas reālās vērtības (3.23 attēls).

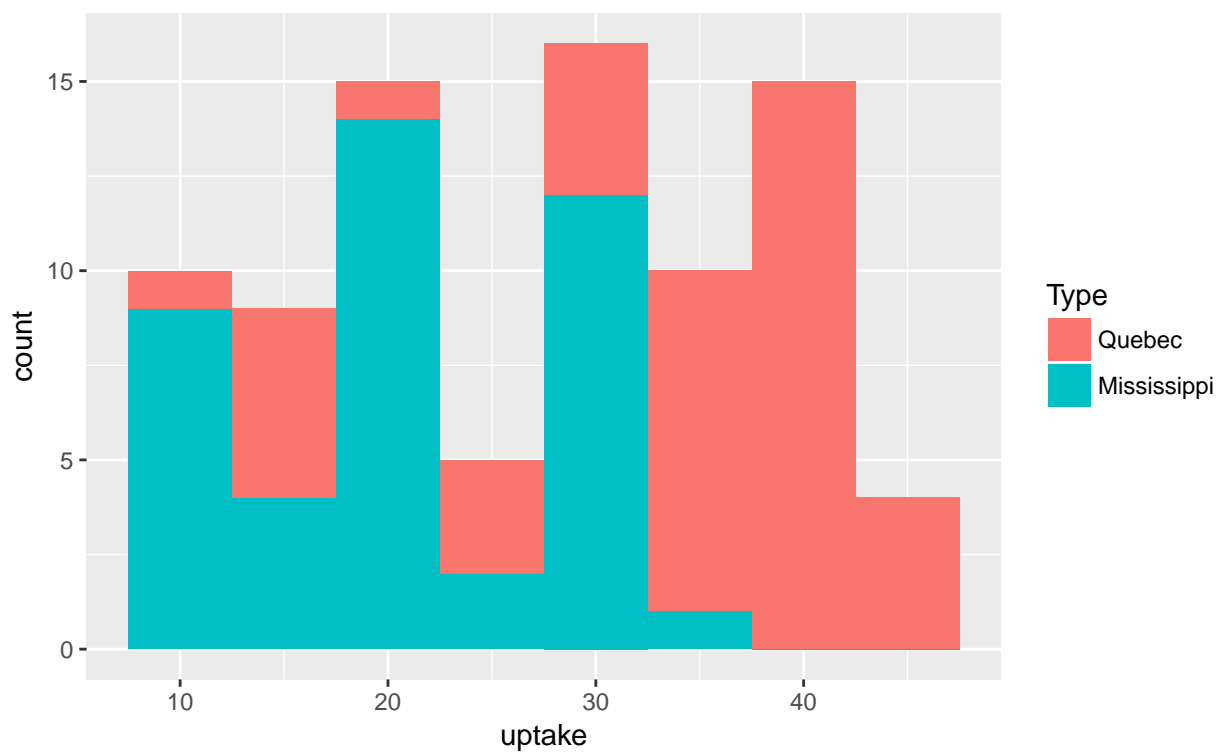
```
ggplot(CO2,aes(Type,uptake))+geom_jitter()
```

`geom_jitter()` ir labi izmantot kombinācijā ar `geom_boxplot()`, jo tādējādi gan parādās reālās vērtības, gan arī vērtību apkopojums (3.24 attēls).

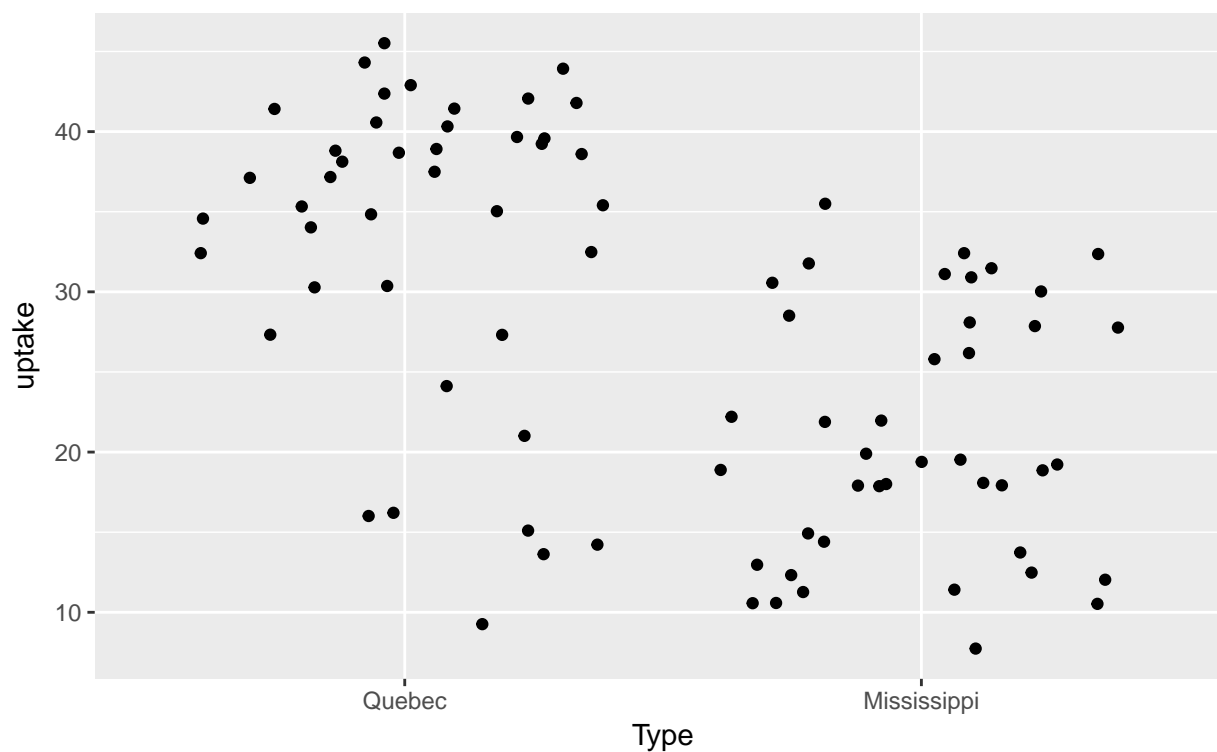
```
ggplot(CO2,aes(Type,uptake))+geom_boxplot()+geom_jitter()
```

`geom_jitter()` un `geom_boxplot()` var kombinēt arī gadījumos, kad vērtībamplitūdas diagramma ir sadalīta atbilstoši trešai mainīgā līmeņiem, bet šajā gadījumā papildus ir jānorāda arguments `position=position_jitterdodge()`, lai punktu izvietojums atbilstu reālajam vērtību sadalījumam pa līmeņiem (3.25 attēls).

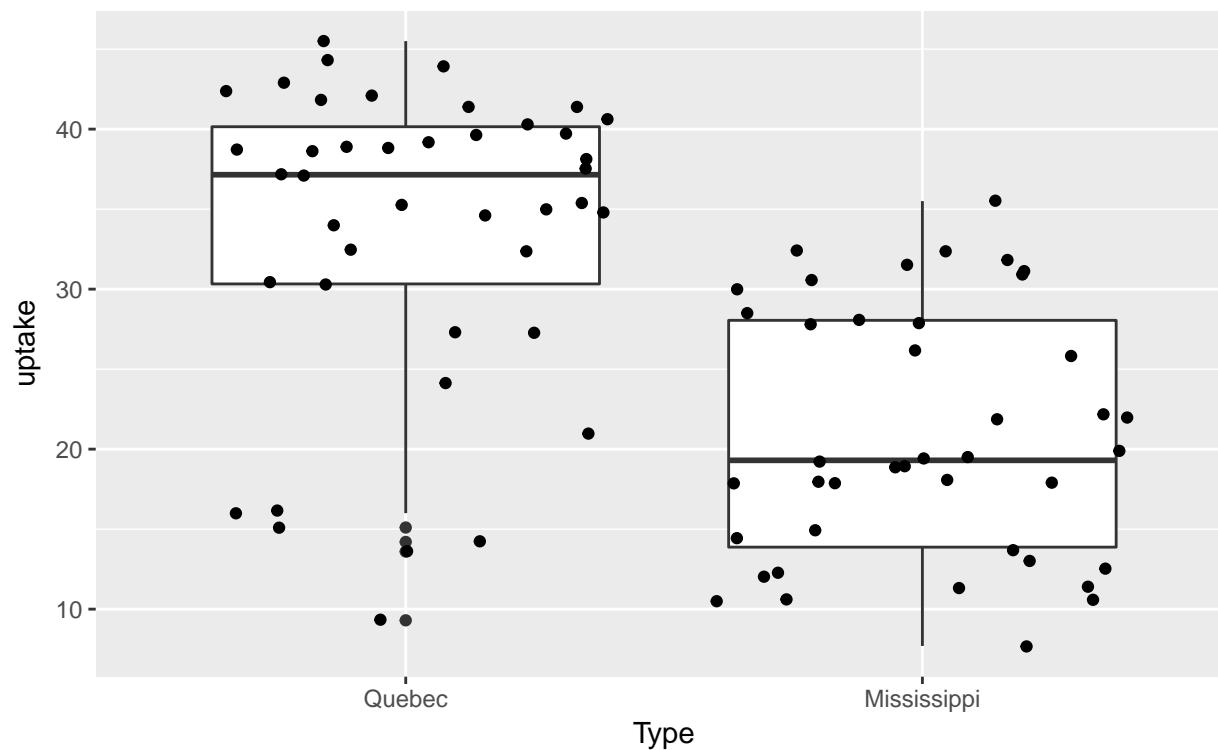
```
ggplot(CO2,aes(Type,uptake,fill=Treatment))+geom_boxplot()+
  geom_jitter(position=position_jitterdodge())
```

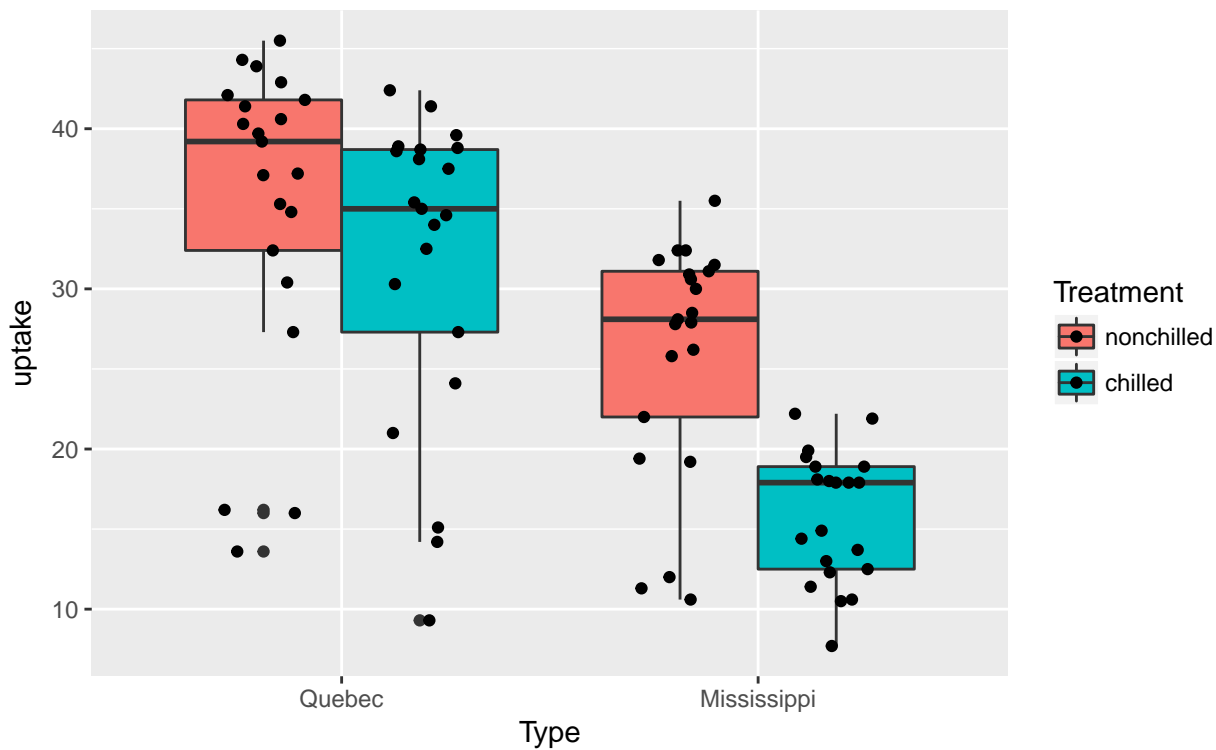
Att. 3.22: Histogrammas, kur aizpildījums atkarīgs no mainīgā



Att. 3.23: Izkļiedes diagramma, kurā x virzienā nejauši mainīts punktu izvietojums



Att. 3.24: Izkļiedes diagrammas un vērtībamplitūdas diagrammas kombinācija



Att. 3.25: Izkļiedes diagrammas un vērtībamplitūdas diagrammas kombinācija gadījumā, kad iesaistīts trešais mainīgais dalījuma līmeņiem

Nodaļa 4

Skalas

Uz x un y ass esošo vērtību, kā arī punktu, līniju, stabiņu krāsu, formu, izmēru utt. vērtību mainīšanai ir jāizmanto speciālas skalu maiņas funkcijas, kuru nosaukumi sastāv no trīs vārdiem. Visām funkcijām pirmais vārds ir `scale`, otrais vārds parāda, kāda veida skala tā ir - x, y vai attiecīgi krāsu (`color`), aizpildījuma (`fill`), līniju veida (`linetype`), punktu formas (`shape`), izmēra (`size`) vai caurspīdīguma (`alpha`). Funkcijas nosaukumā trešais vārds norāda kāda veida vērtības ir izmantotas skalas izveidē - nepārtrauktas (`continuous`) vai diskrētas (`discrete`), kā arī ir citi papildus veidi, piemēram, `manual` (vērtības nosaka manuāli), `gradient` (attiecas uz krāsām un aizpildījumiem).

4.1 `scale_x_continuous()` un `scale_y_continuous()`

x un y ass vērtību maiņai, ja tās skaitliskas (nepārtrauktas), izmanto attiecīgi funkcijas `scale_x_continuous()` un `scale_y_continuous()`. Izmantojot šīs funkcijas var mainīt asu parakstus (arguments `name=`), pozīcijas, kurās parādās skaitļi (`breaks=`) (4.1 attēls).

```
ggplot(CO2,aes(conc,uptake))+geom_point()+
  scale_x_continuous("Koncentrācija",breaks=c(200,400,500))+
  scale_y_continuous("Uzņemtais apjoms")

## Warning in grid.Call(L_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for character 0x7f

## Warning in grid.Call(L_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on 'Koncentrācija' in 'mbcsToSbcs': dot substituted for <c4>

## Warning in grid.Call(L_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on 'Koncentrācija' in 'mbcsToSbcs': dot substituted for <81>

## Warning in grid.Call(L_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on 'Uzņemtais apjoms' in 'mbcsToSbcs': dot substituted for <c5>

## Warning in grid.Call(L_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on 'Uzņemtais apjoms' in 'mbcsToSbcs': dot substituted for <86>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Uzņemtais apjoms' in 'mbcsToSbcs': dot substituted
## for <c5>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Uzņemtais apjoms' in 'mbcsToSbcs': dot substituted
## for <86>
```

```
## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
## conversion failure on 'Uzņemtais apjoms' in 'mbcsToSbcs': dot substituted  
## for <c5>  
  
## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
## conversion failure on 'Uzņemtais apjoms' in 'mbcsToSbcs': dot substituted  
## for <86>  
  
## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
## conversion failure on 'Koncentrācija' in 'mbcsToSbcs': dot substituted for  
## <c4>  
  
## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
## conversion failure on 'Koncentrācija' in 'mbcsToSbcs': dot substituted for  
## <81>  
  
## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
## conversion failure on 'Koncentrācija' in 'mbcsToSbcs': dot substituted for  
## <c4>  
  
## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
## conversion failure on 'Koncentrācija' in 'mbcsToSbcs': dot substituted for  
## <81>  
  
## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
## conversion failure on 'Uzņemtais apjoms' in 'mbcsToSbcs': dot substituted  
## for <c5>  
  
## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
## conversion failure on 'Uzņemtais apjoms' in 'mbcsToSbcs': dot substituted  
## for <86>  
  
## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
## conversion failure on 'Uzņemtais apjoms' in 'mbcsToSbcs': dot substituted  
## for <c5>  
  
## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
## conversion failure on 'Uzņemtais apjoms' in 'mbcsToSbcs': dot substituted  
## for <86>  
  
## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
## conversion failure on 'Koncentrācija' in 'mbcsToSbcs': dot substituted for  
## <c4>  
  
## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
## conversion failure on 'Koncentrācija' in 'mbcsToSbcs': dot substituted for  
## <81>  
  
## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :  
## conversion failure on 'Koncentrācija' in 'mbcsToSbcs': dot substituted for  
## <c4>  
  
## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
```

```

## conversion failure on 'Koncentrācija' in 'mbcsToSbcs': dot substituted for
## <81>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Koncentrācija' in 'mbcsToSbcs': dot substituted for
## <c4>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Koncentrācija' in 'mbcsToSbcs': dot substituted for
## <81>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Koncentrācija' in 'mbcsToSbcs': dot substituted for
## <c4>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Koncentrācija' in 'mbcsToSbcs': dot substituted for
## <81>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Koncentrācija' in 'mbcsToSbcs': dot substituted for
## <c4>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Koncentrācija' in 'mbcsToSbcs': dot substituted for
## <81>

## Warning in grid.Call.graphics(L_text, as.graphicsAnnot(x$label), x
## $x, x$y, : conversion failure on 'Koncentrācija' in 'mbcsToSbcs': dot
## substituted for <c4>

## Warning in grid.Call.graphics(L_text, as.graphicsAnnot(x$label), x
## $x, x$y, : conversion failure on 'Koncentrācija' in 'mbcsToSbcs': dot
## substituted for <81>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Uzņemtais apjoms' in 'mbcsToSbcs': dot substituted
## for <c5>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Uzņemtais apjoms' in 'mbcsToSbcs': dot substituted
## for <86>

## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Uzņemtais apjoms' in 'mbcsToSbcs': dot substituted
## for <c5>

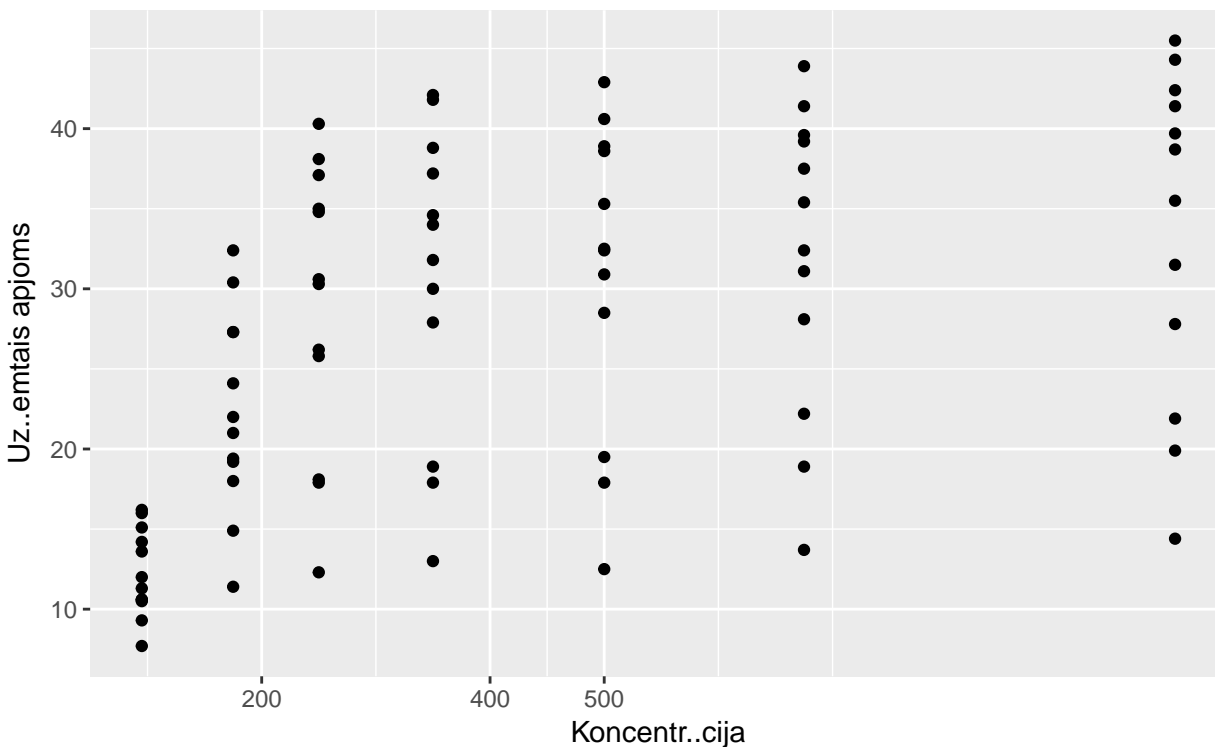
## Warning in grid.Call(L_textBounds, as.graphicsAnnot(x$label), x$x, x$y, :
## conversion failure on 'Uzņemtais apjoms' in 'mbcsToSbcs': dot substituted
## for <86>

## Warning in grid.Call.graphics(L_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on 'Uzņemtais apjoms' in 'mbcsToSbcs': dot
## substituted for <c5>

## Warning in grid.Call.graphics(L_text, as.graphicsAnnot(x$label), x$x, x
## $y, : conversion failure on 'Uzņemtais apjoms' in 'mbcsToSbcs': dot
## substituted for <86>

```

Ar argumentu `limits=` ir iespējams mainīt katras ass garumu, bet jāņem vērā, ka gadījumā, ja jaunais garums būs mazāks nekā vērtību amplitūda, tad vērtības ārpus ass garumu tiks izslēgtas no attēla (to



Att. 4.1: Nepārtraukto asu piemērs

parāda arī brīdinājums par izslēgtām vērtībām), ietekmējot attēlojumu (4.2 attēls). Tas īpaši attiecas uz stabīgu attēliem, vai attēliem ar trenda līniju.

```
ggplot(CO2,aes(conc,uptake))+geom_point()+
  scale_x_continuous(limits=c(200,600))+
  scale_y_continuous(limits=c(0,50))
```

```
## Warning: Removed 48 rows containing missing values (geom_point).
```

y un x asi ir iespējams arī pārvietot attiecīgi uz labo pusi vai uz augšu, norādot argumentu `position=` (4.3 attēls).

```
ggplot(CO2,aes(conc,uptake))+geom_point()+
  scale_y_continuous(position = "right")+
  scale_x_continuous(position = "top")
```

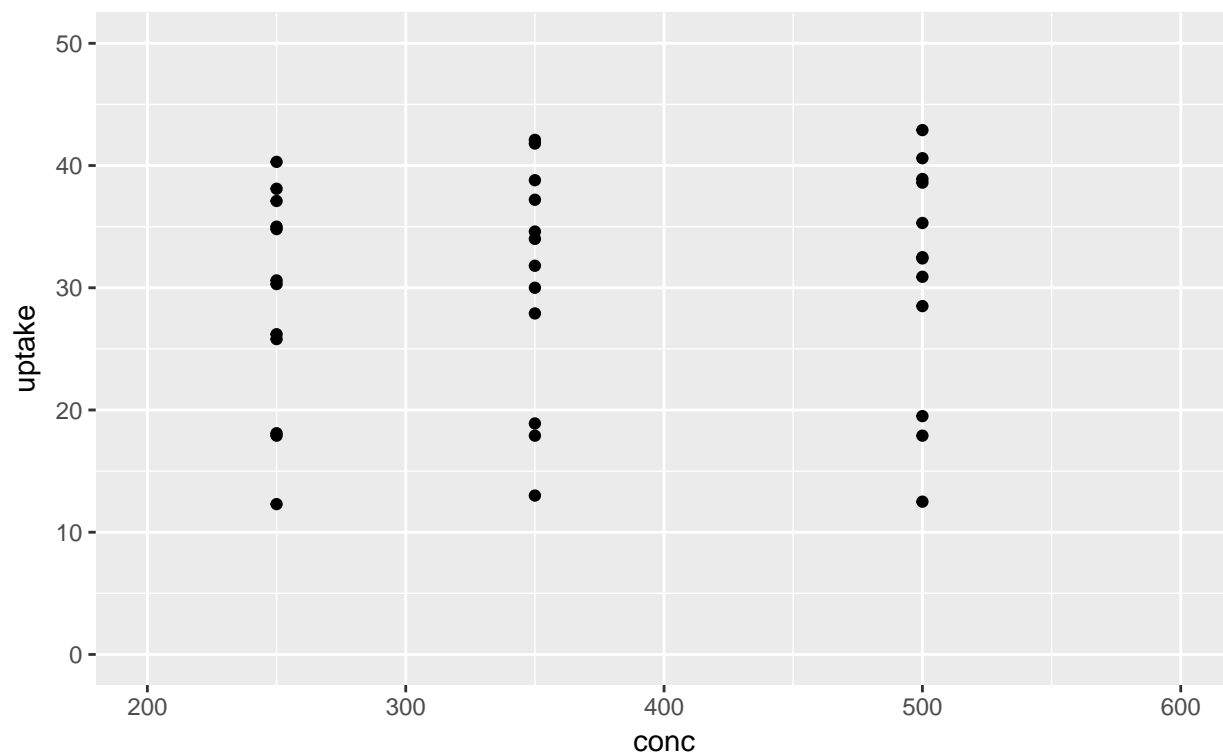
Ar argumentu `sec.axis=` gan x, gan y asij ir iespējams izveidot otro asi, bet tikai ar nosacījumu, ka otrā ass ir tieša pamatass transformāciju (4.4 attēls). Tas nozīmē, ka nevar izveidot otru asi, kas parāda pavisam citas vērtības.

```
ggplot(CO2,aes(conc,uptake))+geom_point()+
  scale_y_continuous(sec.axis = sec_axis(~./100,name="Otrā y ass"))
```

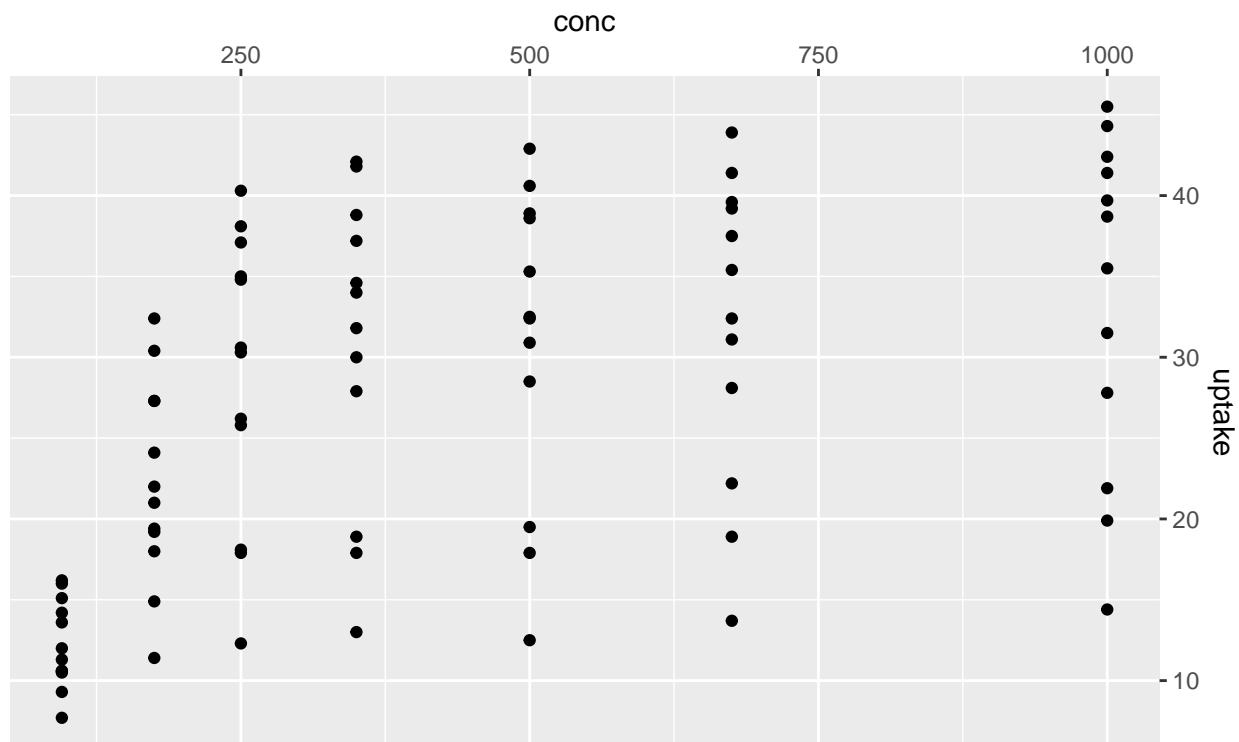
```
## Warning in grid.Call(L_stringMetric, as.graphicsAnnot(x$label)): font
## metrics unknown for character 0x7f
```

```
## Warning in grid.Call(L_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on 'Otrā y ass' in 'mbcsToSbcs': dot substituted for <c4>
```

```
## Warning in grid.Call(L_stringMetric, as.graphicsAnnot(x$label)): conversion
## failure on 'Otrā y ass' in 'mbcsToSbcs': dot substituted for <81>
```

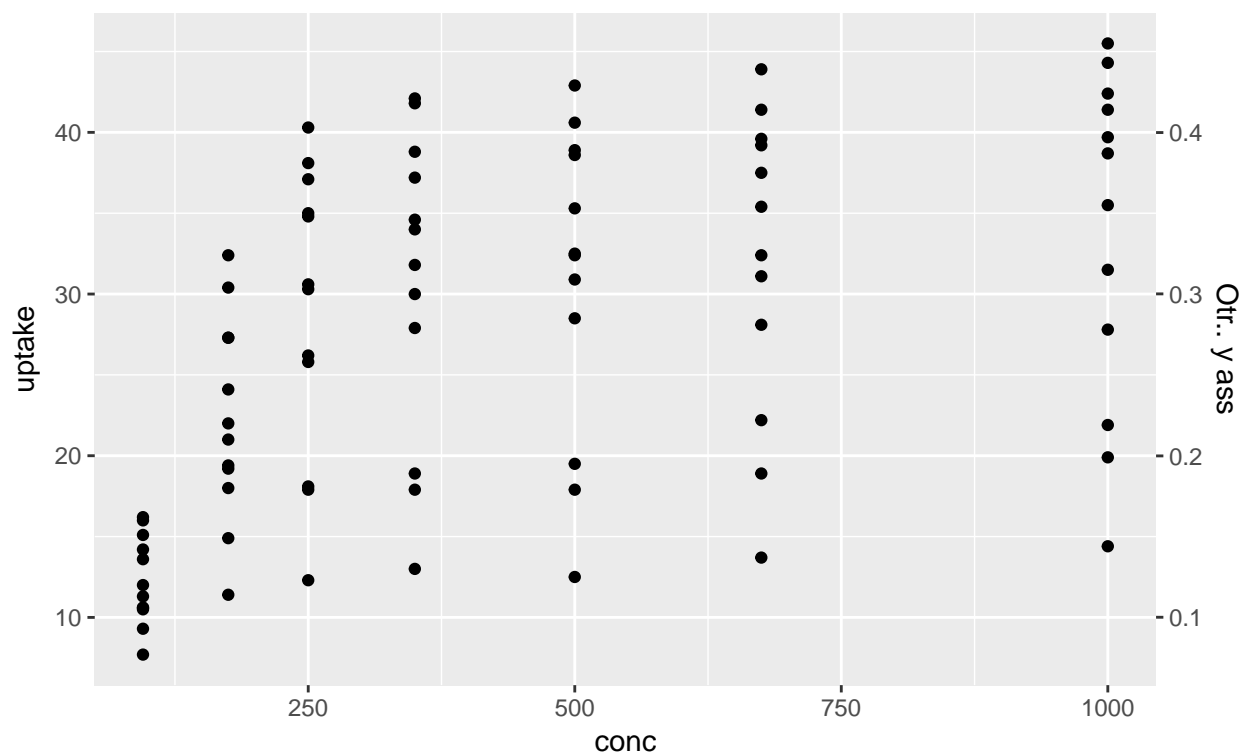


Att. 4.2: Izmainītas nepārtrauktās ass piemērs



Att. 4.3: Pārvietotas x un y asis


```
## $y, : conversion failure on 'Otrā y ass' in 'mbcsToSbcs': dot substituted  
## for <81>
```



Att. 4.4: Attēls ar otru y asi, kas ir pirmās transformācija

Nodaļa 5

Koordinātu sistēmas

Nodaļa 6

Attēlu sadalīšana

Nodaļa 7

Attēla noformēšana

ggplot2 sistēmā izveidoto attēlu izskata mainīšanai var izmantot iepriekš sagatavotas attēla noformēšanas tēmas, vai arī var mainīt katru elementu atsevišķi. Mainot noformējumu, mainās tikai tās attēla daļas, kas nav saistītas attēlojamiem datiem.

7.1 Definētās attēla tēmas

Paketē ggplot2 ir definētas astoņas gatavas tēmas attēla izskata maiņai. Attēlā noformējums mainās, pieskaitot klāt atbilstošo tēmas funkciju. Katrā tēmā papildus ir iespējams mainīt pamatteksta izmēru (`base_size=`) un pamatfontu (`base_family=`). Gatavos tēmu noformējumus protams var papildināt arī ar citām izmaiņām konkrētiem elementiem. Attēlos 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8 un 7.9 parādīts kā izskatās sākotnējais attēls un kā tas mainās, izmantojot kādu no gatavajām tēmām.

```
ggplot(CO2,aes(conc,uptake,color=Type))+geom_point()
```

```
ggplot(CO2,aes(conc,uptake,color=Type))+geom_point() + theme_bw()
```

```
ggplot(CO2,aes(conc,uptake,color=Type))+geom_point() + theme_classic()
```

```
ggplot(CO2,aes(conc,uptake,color=Type))+geom_point() + theme_dark()
```

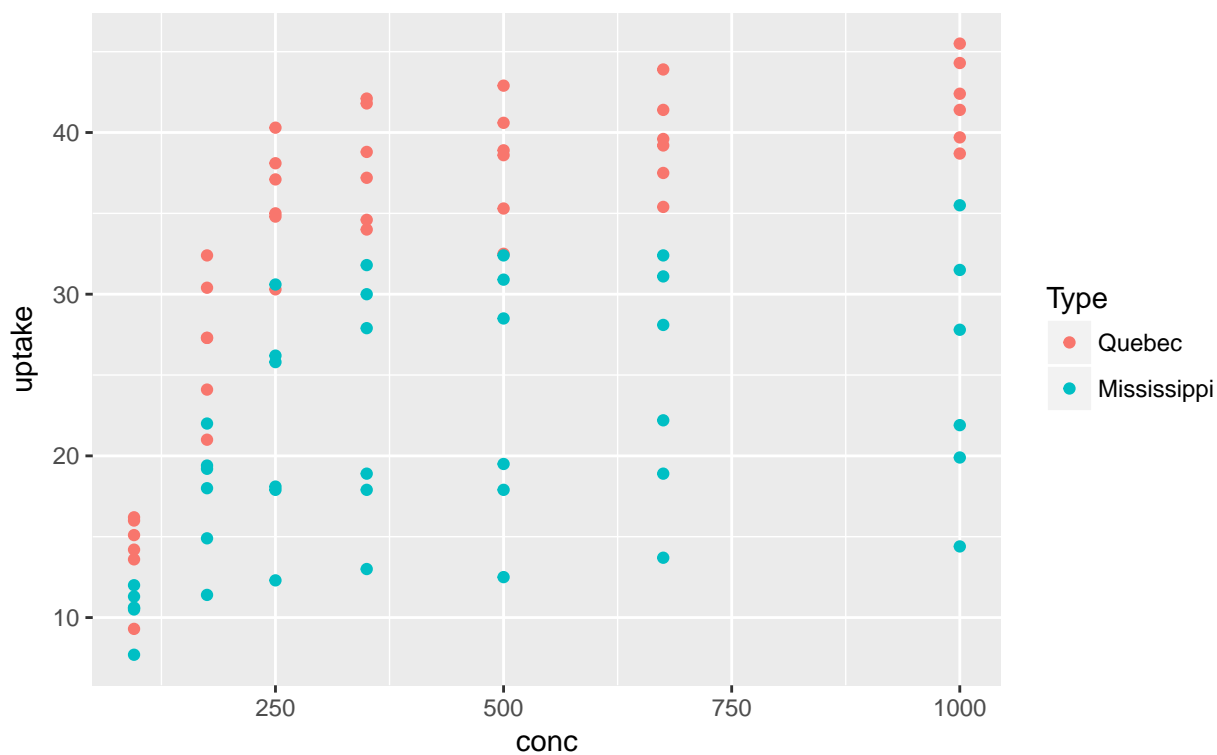
```
ggplot(CO2,aes(conc,uptake,color=Type))+geom_point() + theme_grey()
```

```
ggplot(CO2,aes(conc,uptake,color=Type))+geom_point() + theme_light()
```

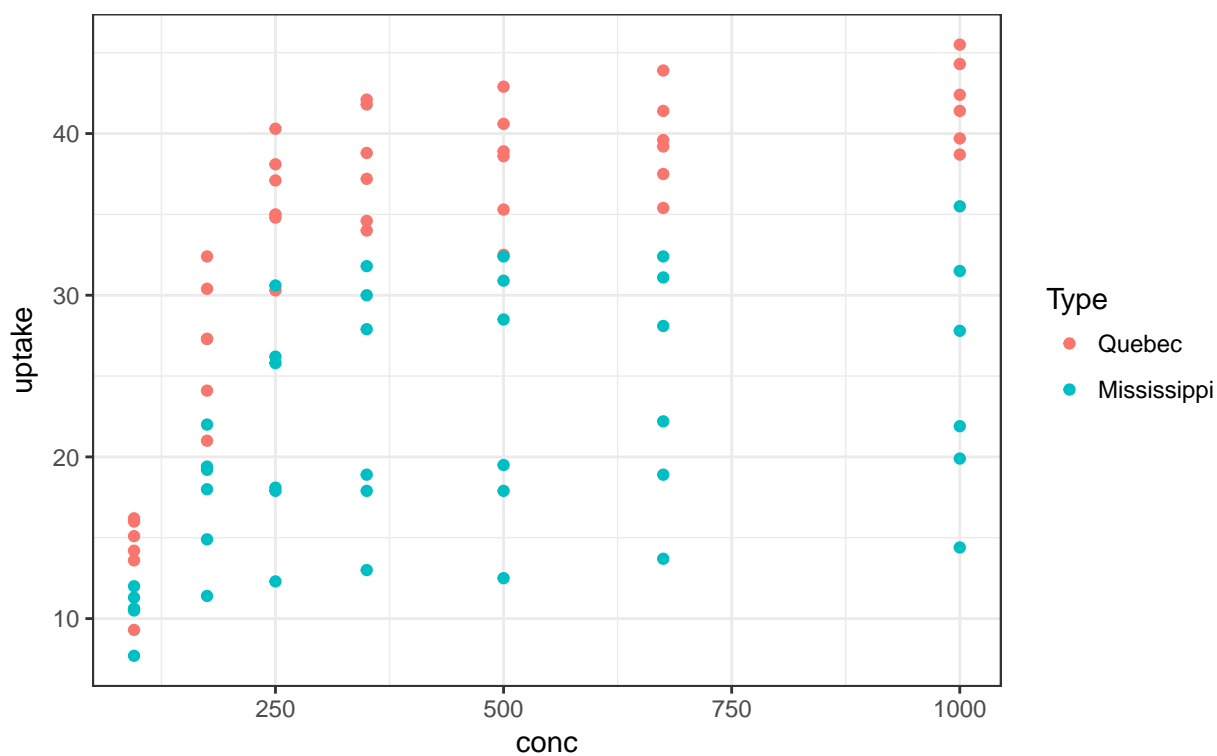
```
ggplot(CO2,aes(conc,uptake,color=Type))+geom_point() + theme_linedraw()
```

```
ggplot(CO2,aes(conc,uptake,color=Type))+geom_point() + theme_minimal()
```

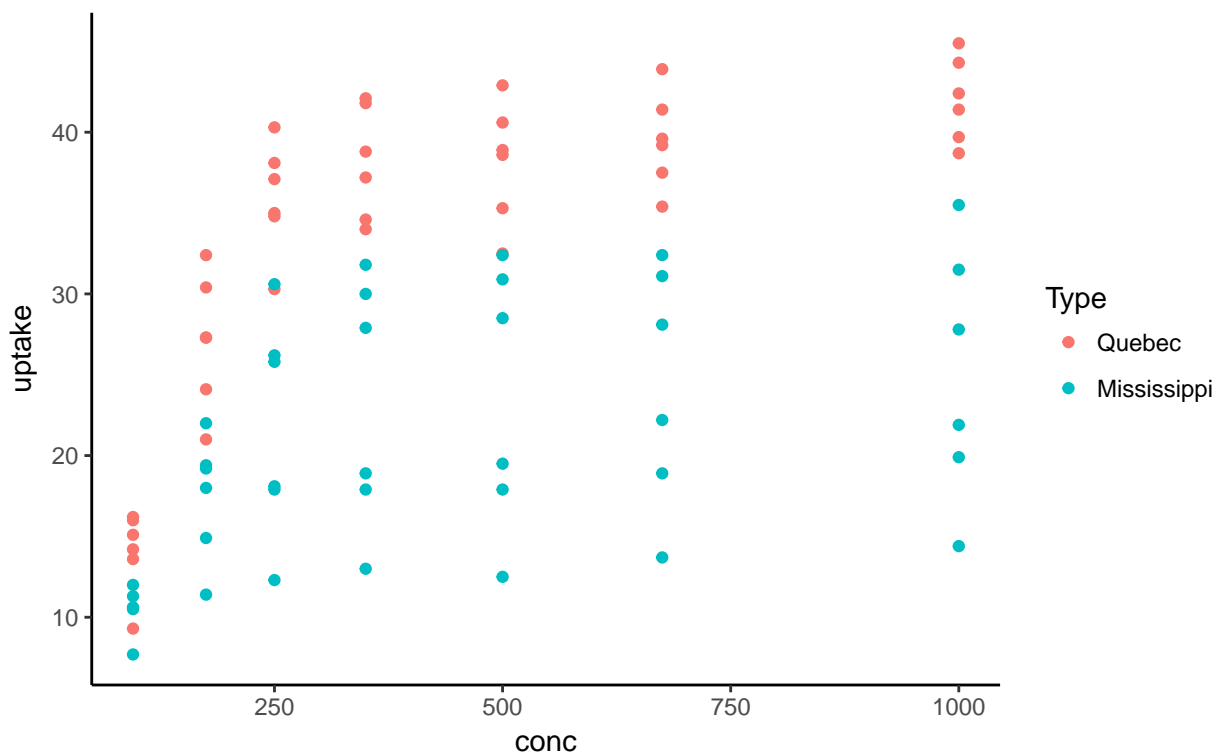
```
ggplot(CO2,aes(conc,uptake,color=Type))+geom_point() + theme_void()
```



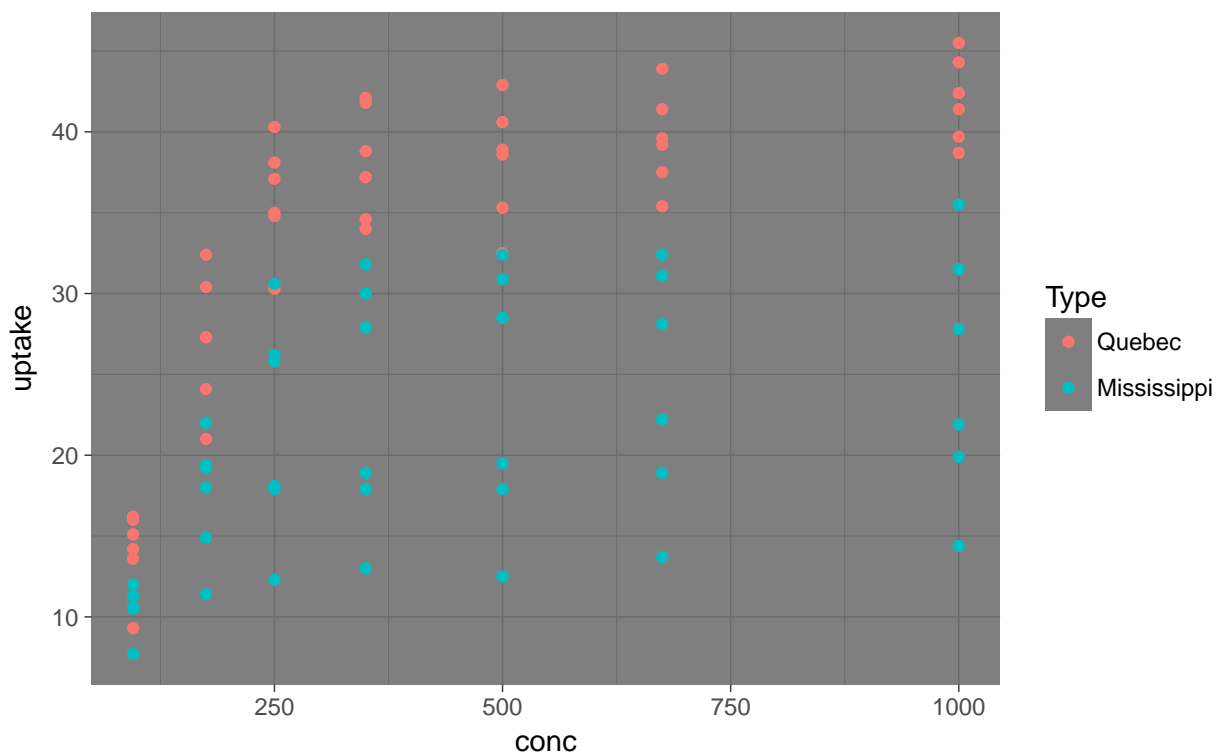
Att. 7.1: Attēls bez papildus noformējuma



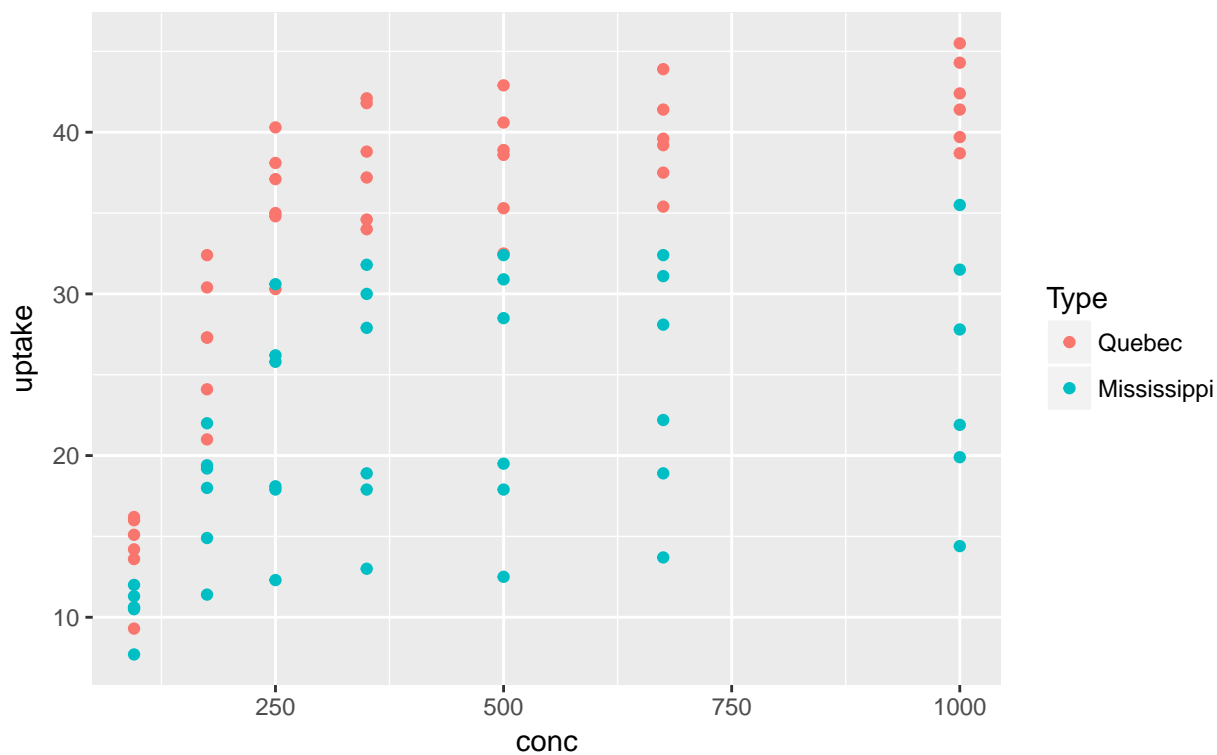
Att. 7.2: Attēls ar theme_bw()



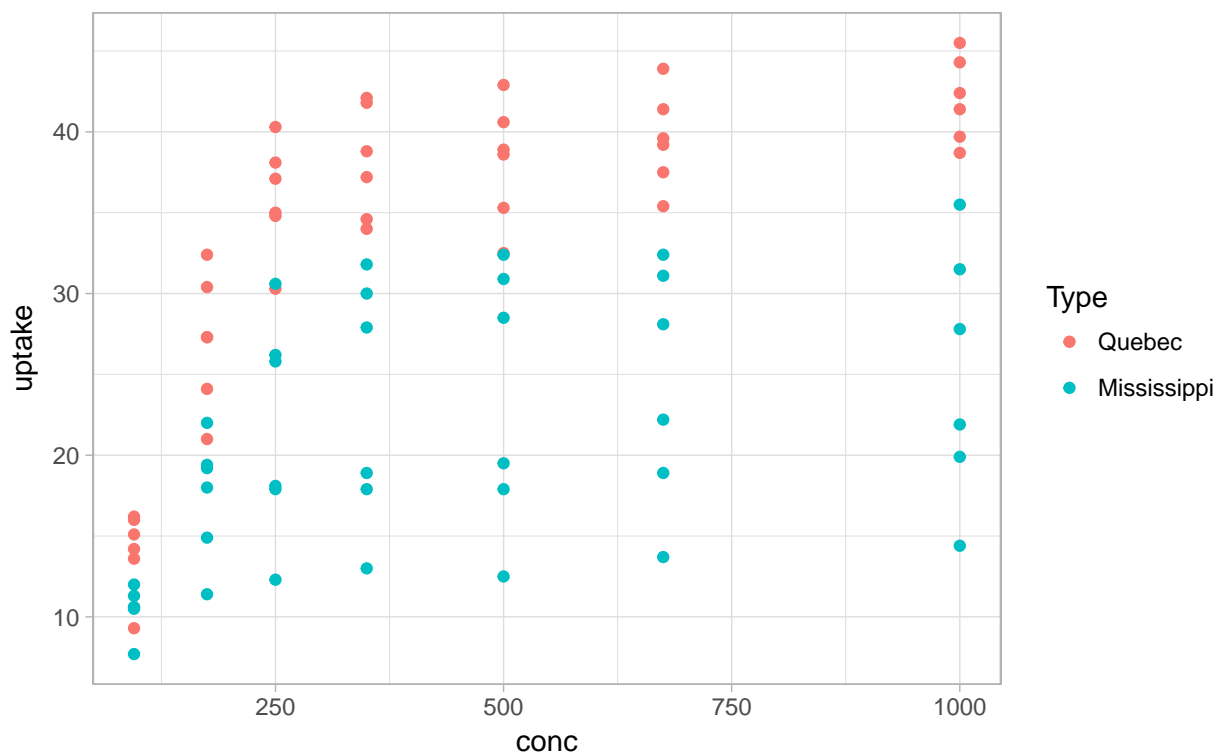
Att. 7.3: Attēls ar theme_classic()



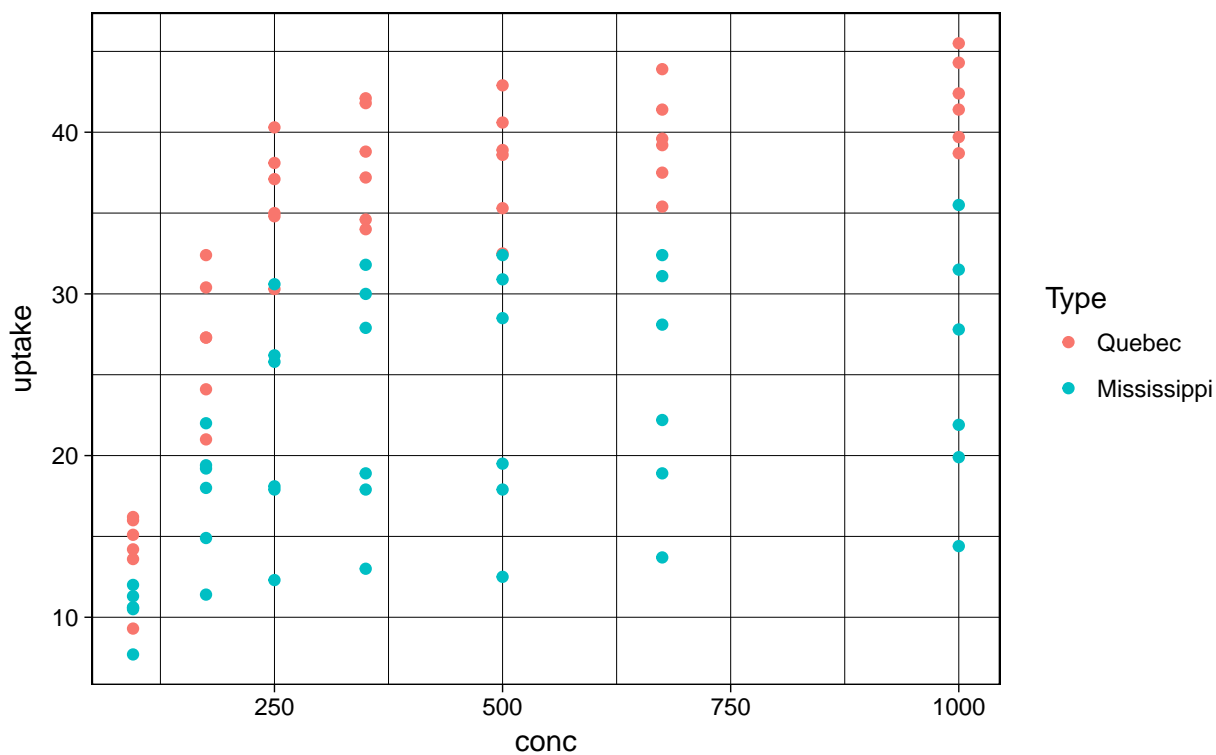
Att. 7.4: Attēls ar theme_dark()



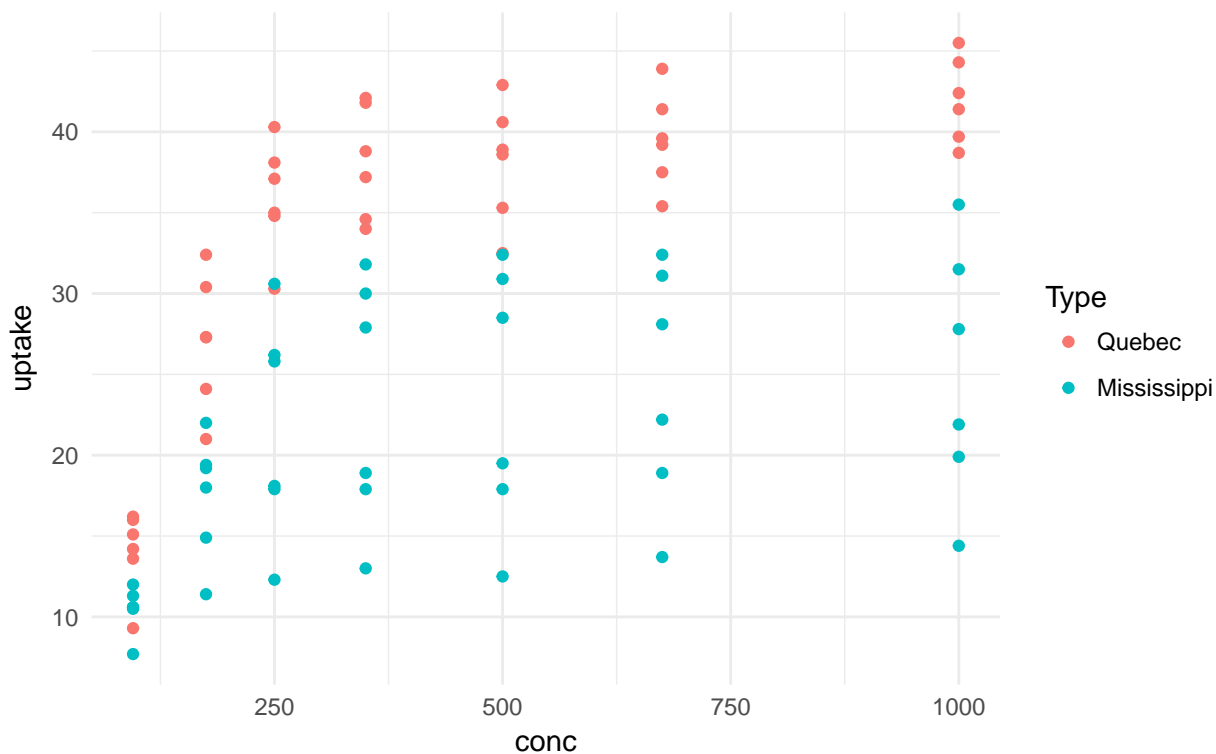
Att. 7.5: Attēls ar theme_grey()



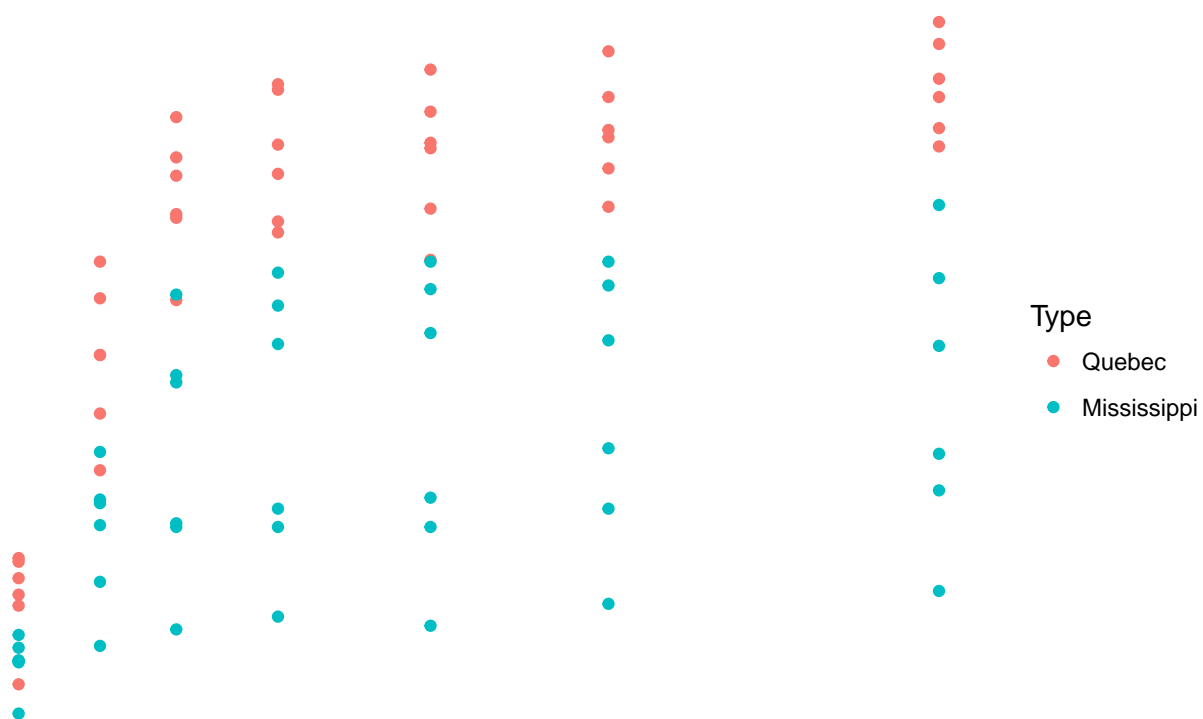
Att. 7.6: Attēls ar theme_light()



Att. 7.7: Attēls ar theme_linedraw()



Att. 7.8: Attēls ar theme_minimal()



Att. 7.9: Attēls ar theme_void()

Literatūra

Wickham, H. (2009). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.