L05 – Olympic Runs

10. Mai 2021

Contents


```
library(tidyverse)
```

1 Top Oliympic Running Medalists – dplyr Warm-Up

Wir laden ein `.csv`-Datei mit Daten von Teilnehmer:innen der olympischen Spiele.

```
# We are only interested in following disciplines
std_runs <- c("100 metres", "200 metres", "400 metres", "800 metres",
              "1,500 metres", "5,000 metres", "10,000 metres", "Marathon")

read_csv("athlete_events.csv") %>%
  filter(Sport == "Athletics") %>%
  select(Name, Sex, Age, Height, Weight, Year, Event, Medal) %>%
  mutate(
    Event = str_remove(Event, "Athletics Women's "),
    Event = str_remove(Event, "Athletics Men's ")) %>%
  filter(Event %in% std_runs) ->
  athletes
```

```
athletes
## # A tibble: 15,919 x 8
##    Name                          Sex     Age Height Weight  Year Event      Medal
##    <chr>                         <chr> <dbl>  <dbl>  <dbl> <dbl> <chr>      <chr>
##  1 "Cornelia \"Cor\" Aalten (-~ F        18    168     NA  1932 100 metres <NA>
##  2 "Jamale (Djamel-) Aarrass (~ M        30    187     76  2012 1,500 met~ <NA>
##  3 "Antonio Abadia Beci"        M        26    170     65  2016 5,000 met~ <NA>
##  4 "Jos Manuel Abascal Gmez"    M        22    182     67  1980 1,500 met~ <NA>
##  5 "Jos Manuel Abascal Gmez"    M        26    182     67  1984 1,500 met~ Bron~
##  6 "Georgia Abatzidou"          F        35    155     43  2004 Marathon   <NA>
##  7 "Carlos Rodolfo Abaunza Bal~ M        18    168     60  2004 100 metres <NA>
##  8 "Gana Abba Kimet"            M        26     NA     NA  1972 100 metres <NA>
##  9 "Abubakar Abbas Abbas"       M        20    175     66  2016 400 metres <NA>
## 10 "Maher Abbas"                M        22    178     78  1988 400 metres <NA>
## # ... with 15,909 more rows
```

Das Tibble `athletes` kann mit `View(athletes)` oder einem Klick auf das Objekt im Environment-Tab von

RStudio genauer betrachtet werden.

Aufgabe: Erstelle eine Tabelle von Athletinnen:innen sortiert nach Medaillen-Rang, dh absteigend nach Anzahl der Goldmedaillen, bei Gleichstand werden zuerst Silber- dann Bronzemedaillen in Bertacht gezogen. Unten stehen Zeile 6 bis 10 der erwarteten Ausgabe.

Beginne mit folgenden Zeilen.

```
athletes %>%
  pivot_wider(names_from = Medal, values_from = Medal) # %>% TODO
```

Nutze dann `dplyr`-Verben wie `mutate()`, `group_by()`, `summarize()`, `arrange()`, `transmute()`, ...

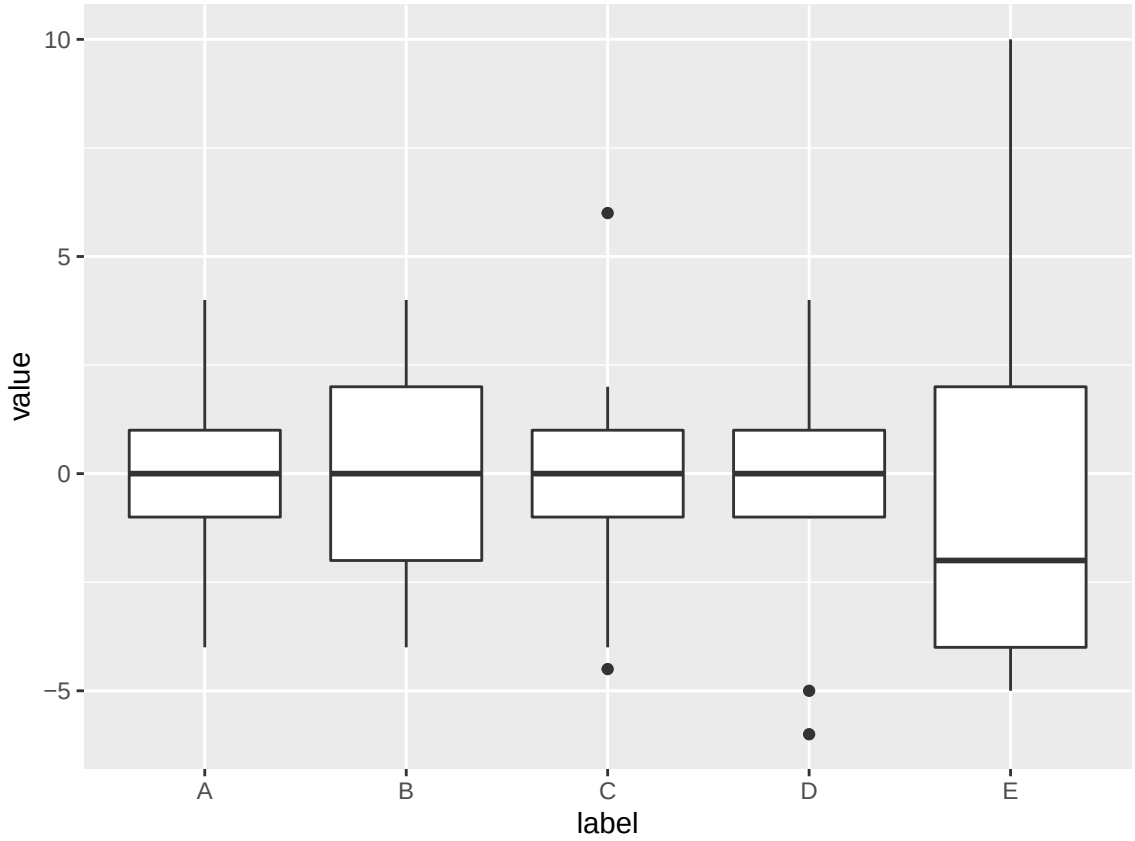
```
print(medal_ranking[6:10,])
## # A tibble: 5 x 4
##   Name                                  Gold Silver Bronze
##   <chr>                                <int>  <int>  <int>
## 1 "Frederick Carlton \"Carl\" Lewis"       3      1      0
## 2 "James Davies \"Jim\" Lightbody"         3      1      0
## 3 "Kenenisa Bekele Beyecha"                3      1      0
## 4 "Tirunesh Dibaba Keneni"                 3      0      3
## 5 "Charles Archibald \"Archie\" Hahn"      3      0      0
```

2 Age Distribution among Women's Marathon Participants – Box-Plot

Ein *Box-Plot* (auch *Box-Whiskers-Plot*) fasst die Verteilung einer reellen Variable visuell zusammen.

Durch die Addition von `geom_boxplot()` fügen wir einem ggplot-Objekt ein Box-Plot hinzu.

```
v <- list(
  c(-4,-1,0,1,4), #A
  c(-4,-2,0,2,4), #B
  c(-4.5,-4,-1,-1,0,1,1,2,6), #C
  c(  -6,-5,-1,-1,0,1,1,3,4), #D
  -6 + c(1,2,4,8,16)) #E
tb <- tibble(
  label = rep(LETTERS[1:length(v)], sapply(v, length)),
  value = unlist(v))
ggplot(tb) + geom_boxplot(aes(x = label, y = value))
```

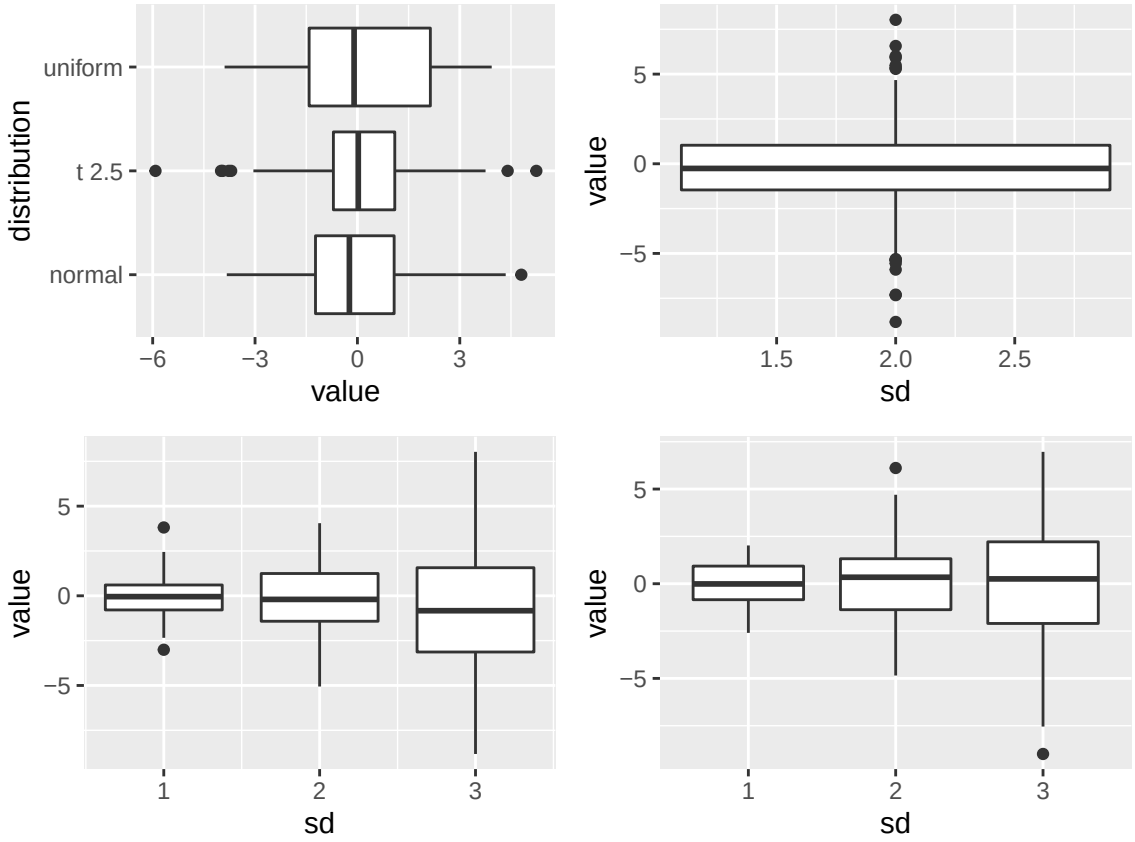


Die dicke waagrechte Linie gibt den Median der entsprechenden Variable an. Die Box umschließt jeweils die Hälfte der Werte oberhalb und unterhalb des Medians. D.h. die Grenzen der Box sind das erste und dritte *Quartil*. Die von der Box ausgehenden Linien heißen *Whiskers* (Schnurrhaare). Sie erstrecken sich wenn möglich jeweils zu den extremalen Datenpunkten, sind aber höchstens 1,5-mal so lange wie der *Interquartilabstand* (also die Gesamthöhe der Box).

`geom_box()` benötigt die *aesthetics* `x` und `y`. Ist eine der beiden diskret (character-Vektor oder factor) und die andere kontinuierlich (double-Vektor), wird automatisch bzgl der diskreten Variable gruppiert. Liegen beide Variablen in kontinuierlichem Format vor, wobei aber bzgl einer der beiden Variablen gruppiert werden soll, so können wir diese als *aesthetic* `group` angeben.

```
set.seed(1)
n <- 100

tb1 <- tibble(
  distribution = rep(c("uniform", "normal", "t 2.5"), each=n), # discrete
  value = c(runif(n, min=-4, max=4), rnorm(n, sd=2), rt(n, df=2.5)) # continuous
)
ggplot(tb1) + geom_boxplot(aes(y = distribution, x = value)) -> p1

tb2 <- tibble(
  sd = rep(1:3, each=n), # continuous
  value = c(rnorm(n, sd=1), rnorm(n, sd=2), rnorm(n, sd=3)) # continuous
)
ggplot(tb2) + geom_boxplot(aes(x = sd, y = value)) -> p2

ggplot(tb2) + geom_boxplot(aes(x = sd, y = value, group = sd)) -> p3
```

```r
tb3 <- tibble(
  sd = factor(rep(1:3, each=n)), # discrete
  value = c(rnorm(n, sd=1), rnorm(n, sd=2), rnorm(n, sd=3)) # continuous
)
ggplot(tb3) + geom_boxplot(aes(x = sd, y = value)) -> p4

gridExtra::grid.arrange(p1, p2, p3, p4, nrow=2)
## Warning: Continuous x aesthetic -- did you forget aes(group=...)?
```



Viele weitere Optionen von `geom_boxplot()` sind in der Hilfe (`?geom_boxplot`) beschrieben.

Aufgabe:

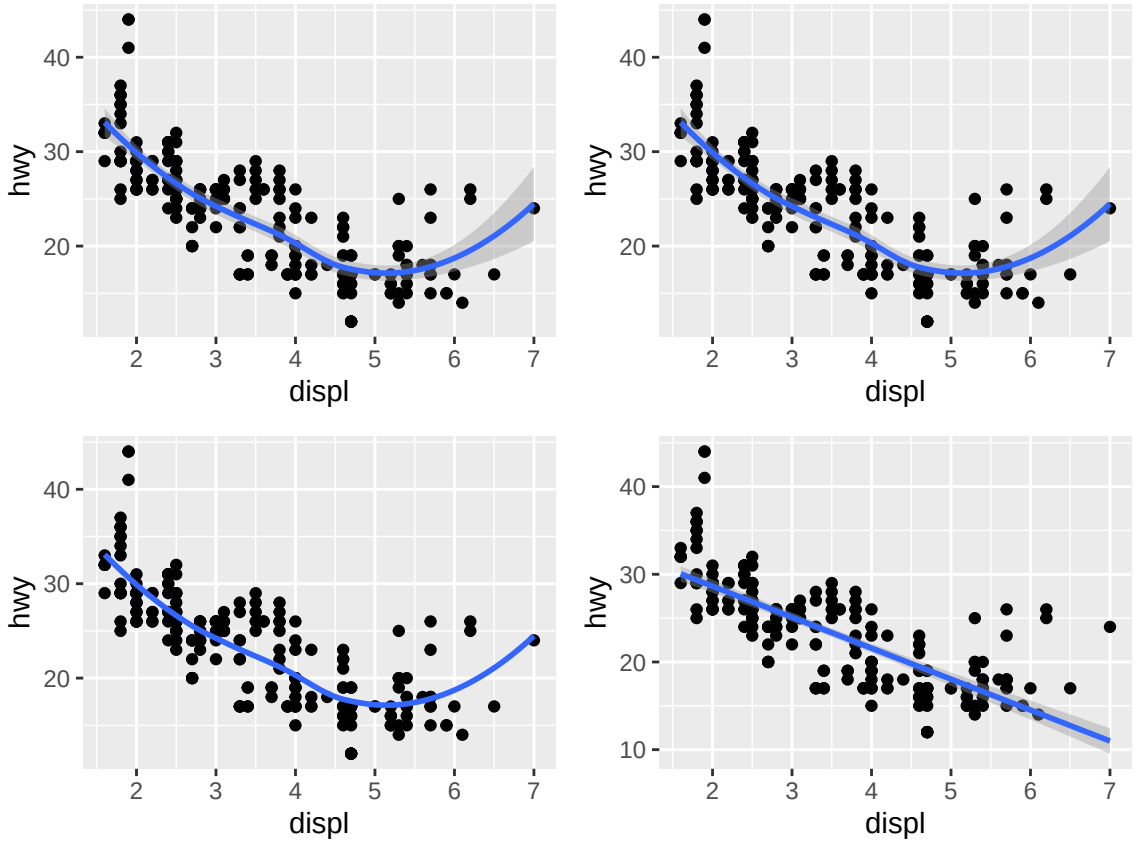
1. Erstelle eine Box-Plot aus dem Tibble `d`, welcher für die alle Olympischen Spiele nach 1980 jeweils die Altersverteilung der Marathonläuferinnen visualisiert.
2. Füge dem Plot mit `geom_point()` für jedes Jahr die drei Medaillengewinnerinnen hinzu. Hierbei kann `drop_na()` hilfreich sein. Die Punkte sollen dabei der Medaille entsprechend via `medal_color` eingefärbt werden.

```r
medal_color <- c(Bronze = "#6A3805", Silver = "#B4B4B4", Gold = "#AF9500")

athletes %>%
  filter(Event == "Marathon", Sex == "F", Year > 1980) ->
  d

# TODO ggplot()
```

3 Change in Height of Male Runners – Regression

Mit `geom_smooth()` fügen wir einem Plot eine Regressionskurve hinzu. Dabei können verschiedene Methoden verwendet werden. Die Standrad-Methode `"loess"` passt eine glatte Kurve an die Daten an. Die dabei benutzte Methode der lokalen Polynomregression werden wir in einem Vorlesungskapitel in Phase II besprechen. Standardmäßig wird ein punktweises 95%-Konfidenzinterval angezeigt.

```
plt <- ggplot(mpg, aes(displ, hwy)) + geom_point()
gridExtra::grid.arrange(
  plt + geom_smooth(),
  # equivalent:
  plt + geom_smooth(method = "loess"), # cf ?loess
  # method = 'loess', but without displaying a confidence interval
  plt + geom_smooth(se = FALSE),
  # linear model
  plt + geom_smooth(method = "lm"))
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```



Aufgabe:

1. Nutze dplyr-Verben, um aus `athletes` für alle Spiele nach 1900 und alle Disziplinen jeweils den Durchschnitt der Körpergröße (`Height`) der männlichen Teilnehmer (deren Größe angegeben ist) zu berechnen und in einer Spalte `MeanHeight` von `d` zu speichern.

```r
athletes %>%
  mutate(Event = factor(Event, levels=std_runs)) %>% # use factor for ordering
  filter(Sex == "M", Year > 1900) ->
  # TODO calculate average height
  d
```

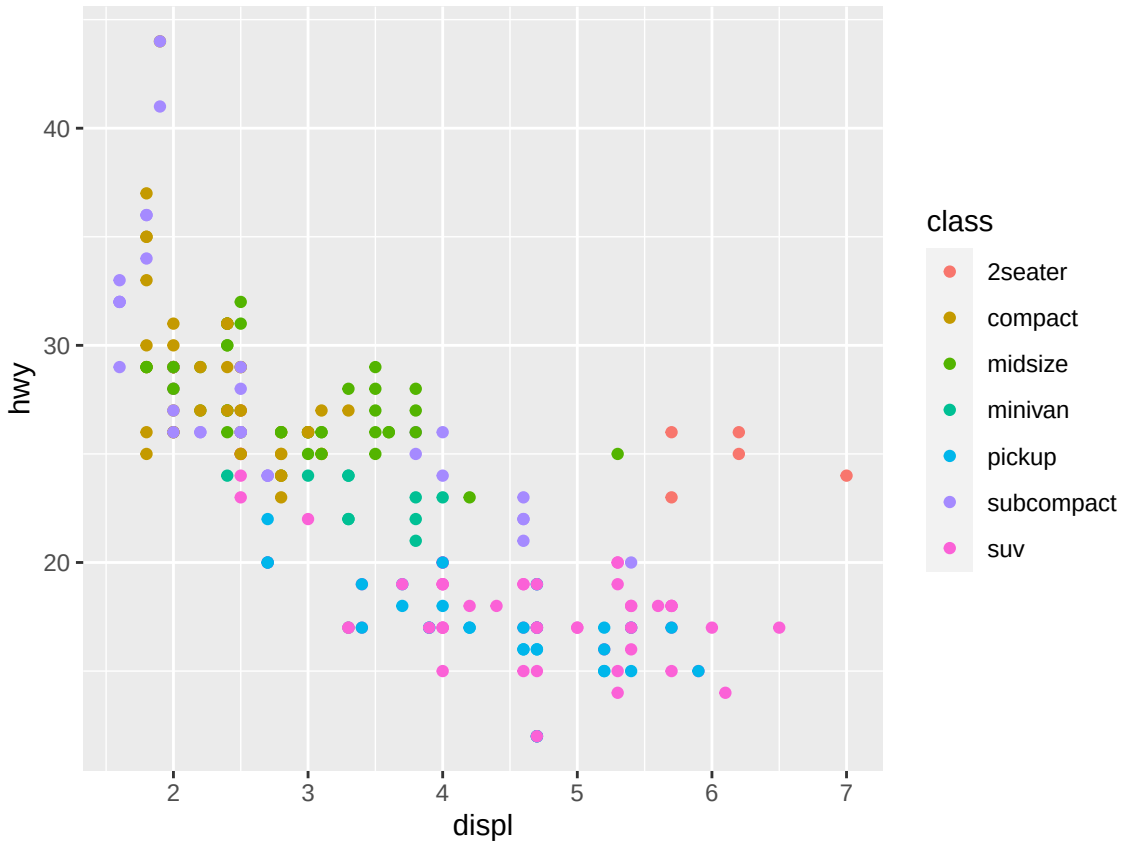
2. Füge dem Plot unten mittels `geom_smooth()` für jede Disziplin die lineare Regressionsgerade hinzu, um einen visuellen Hinweis auf einen möglichen Trend der Größenentwicklung zu erhalten. Dabei sollen keine Konfidenzintervalle angezeigt werden.

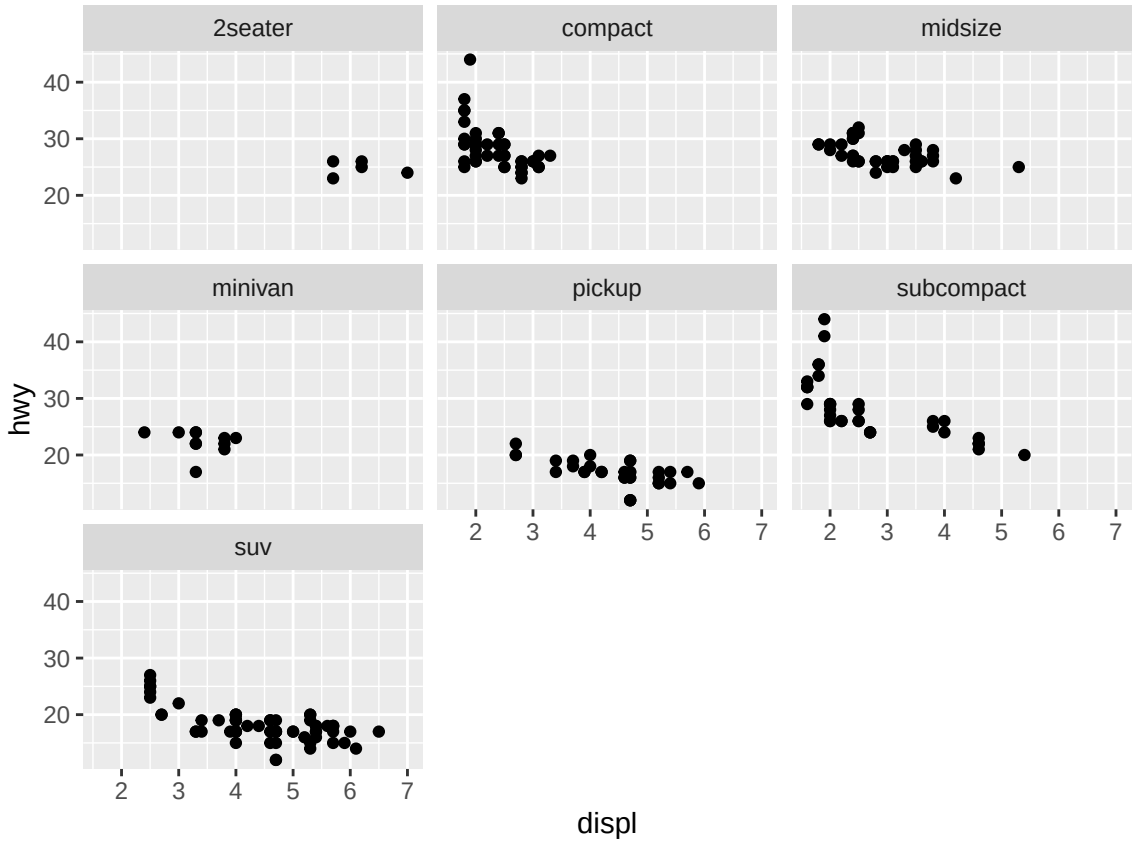
```r
ggplot(d, aes(x = Year, y = MeanHeight, color = Event)) +
  geom_point() +
  ggtitle("Men's runs - mean across participants")
# TODO add geom_smooth
```

4 Bonus: Medalist Times – Facet Wrap

Mit dem sogenannten *Facetting* erzeugen wir mehrere Plots in einer Grafik, die verschiedene Teilmengen der Daten anzeigen. Wir addieren dazu `facet_wrap()` zu einem ggplot-Objekt. Die Gruppierungsvariable wird `facet_wrap()` in der Funktion `vars()` übergeben – ähnlich der aesthetics, die in `aes()` übergeben werden.

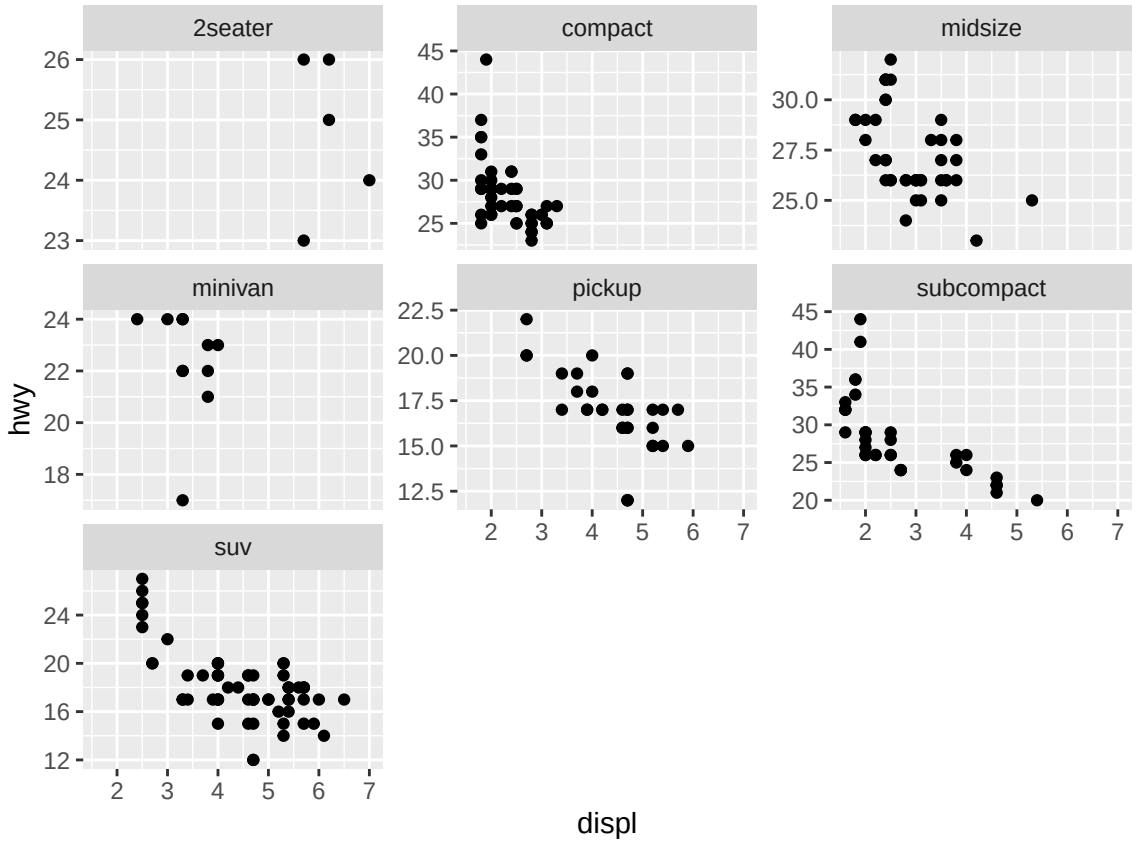
```r
# two ways of distingishing classes of cars in the plot
# 1.: different color in same plot
ggplot(mpg, aes(displ, hwy, color=class)) +
  geom_point()
# 2.: individual plots
ggplot(mpg, aes(displ, hwy)) +
  geom_point() +
  facet_wrap(vars(class))
```


hwy
40
30
20
2
3
4
5
6
7
displ
class
2seater
compact
midsize
minivan
pickup
subcompact
suv



Beachte, dass alle Plots die gleichen Skalen an den Achsen haben. Ist dies nicht gewünscht, setze das Argument `scales` entsprechend.

```
ggplot(mpg, aes(displ, hwy)) +
  geom_point() +
  facet_wrap(vars(class), scales = "free_y") # "free_x", "free_y", "free", default is "fixed"
```



Wir laden nun einen anderen Datensatz. Dieser enthält die Zielzeiten der Medaillengewinner:innen bei den Lauf-Disziplinen der olympischen Spiele.

```
# Times are given as strings with inconsistent format.
# Need custom function for conversion in seconds
str2sec <- function(s) {
  s %>%
    str_split(":|h|-") %>%
    sapply(function(x) {
      v <- as.double(x)
      v3 <- c(0,0,0)
      v3[(4-length(v)):3] <- v
      v3[1] * 3600 + v3[2] * 60 + v3[3]
    })
}

# We are only interested in following disciplines
std_runs <- c("100M", "200M", "400M", "800M", "1500M", "5000M", "10000M", "Marathon")

read_csv("results.csv") %>%
  mutate(
    Event = str_remove(Event, " Men"),
    Event = str_remove(Event, " Women")) %>%
  filter(Event %in% std_runs) %>%
  mutate(Result = str2sec(Result)) %>% # Result now is time in seconds
  drop_na() ->
  runs
```

```
medal_color <- c(B = "#6A3805", S = "#B4B4B4", G = "#AF9500")
```

```
runs
## # A tibble: 903 x 8
##    Gender Event  Location   Year Medal Name                  Nationality Result
##    <chr>  <chr>  <chr>     <dbl> <chr> <chr>                 <chr>        <dbl>
##  1 M      10000M Rio        2016 G     Mohamed FARAH         USA          1505.
##  2 M      10000M Rio        2016 S     Paul Kipngetich TANUI KEN          1626.
##  3 M      10000M Rio        2016 B     Tamirat TOLA          ETH          1626.
##  4 M      10000M Beijing    2008 G     Kenenisa BEKELE       ETH          1621.
##  5 M      10000M Beijing    2008 S     Sileshi SIHINE        ETH          1623.
##  6 M      10000M Beijing    2008 B     Micah KOGO            KEN          1624.
##  7 M      10000M Sydney     2000 G     Haile GEBRSELASSIE    ETH          1638.
##  8 M      10000M Sydney     2000 S     Paul TERGAT           KEN          1638.
##  9 M      10000M Sydney     2000 B     Assefa MEZGEBU        ETH          1640.
## 10 M      10000M Barcelona  1992 G     Khalid SKAH           MAR          1667.
## # ... with 893 more rows
```

Aufgabe:

Erstelle mittels `facet_warp()` eine Grafik, die für jede Disziplin einen Plot enthält. In diesem Plot sollen die Jahreszahlen der jeweiligen olympischen Spiele gegen die Zielzeiten der Medaillengewinner:innen mittels `geom_point()` aufgetragen werden. Zeichne Frauen und Männer mit unterschiedlichen Symbolen (aesthetic `shape`) und jeden Punkt mit der der Medaille entsprechenden Farbe.