Data Viz Challenges Cheat Sheet

Valentin Kodderitzsch 3895157

2024-05-22

Introduction

The most relevant lectures are: L2, L3, L7 and L8. But I will include all chalenges.

L2: Ggplot basics, group by color/shape, manually adjust scales, legends

L3: Proportions, bar chart, group/stacked bar chart

L7: Trends, time series, scatter plot with trend line

L8: Geospatial data

```
setwd("/Users/valentinkodderitzsch/Coding/r-for-stats/semester_2/data_viz/cheat_sheet")

library(ggplot2)
library(dplyr)
library(tidyr) # Long vs wide transformation
library(see) # scale_fill_okabeito color scheme
```

Overview

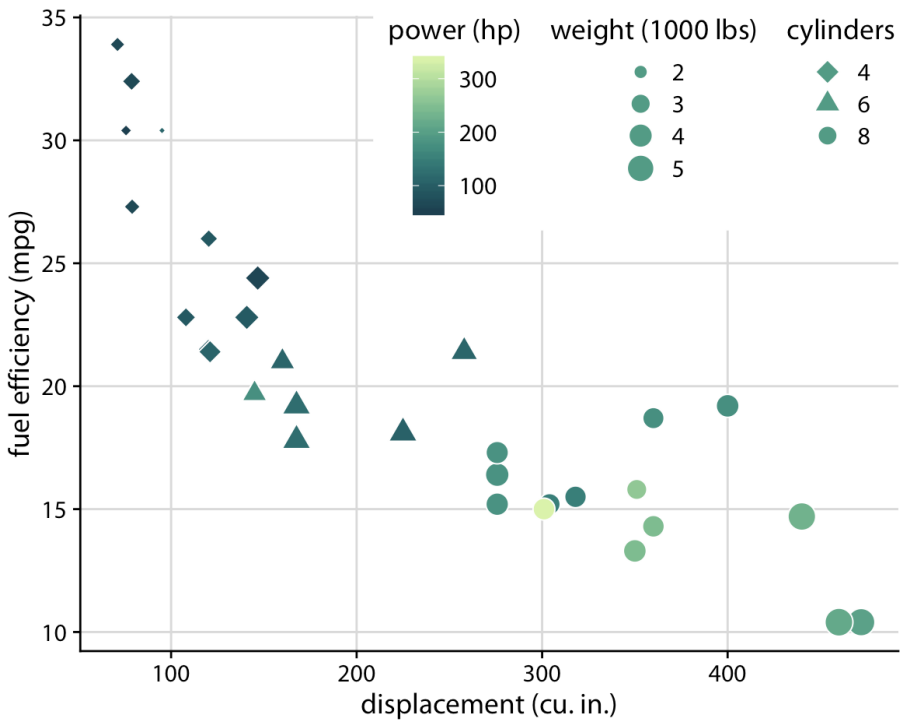


Figure 2.5: Fuel efficiency versus displacement, for 32 cars (1973–74 models). This figure uses five separate scales to represent data: (i) the *x* axis (displacement); (ii) the *y* axis (fuel efficiency); (iii) the color of the data points (power); (iv) the size of the data points (weight); and (v) the shape of the data points (number of cylinders). Four of the five variables displayed (displacement, fuel efficiency, power, and weight) are numerical continuous. The remaining one (number of cylinders) can be considered to be either numerical discrete or qualitative ordered. Data source: *Motor Trend*, 1974.

Figure 1: L2 Scatter, redundant coding

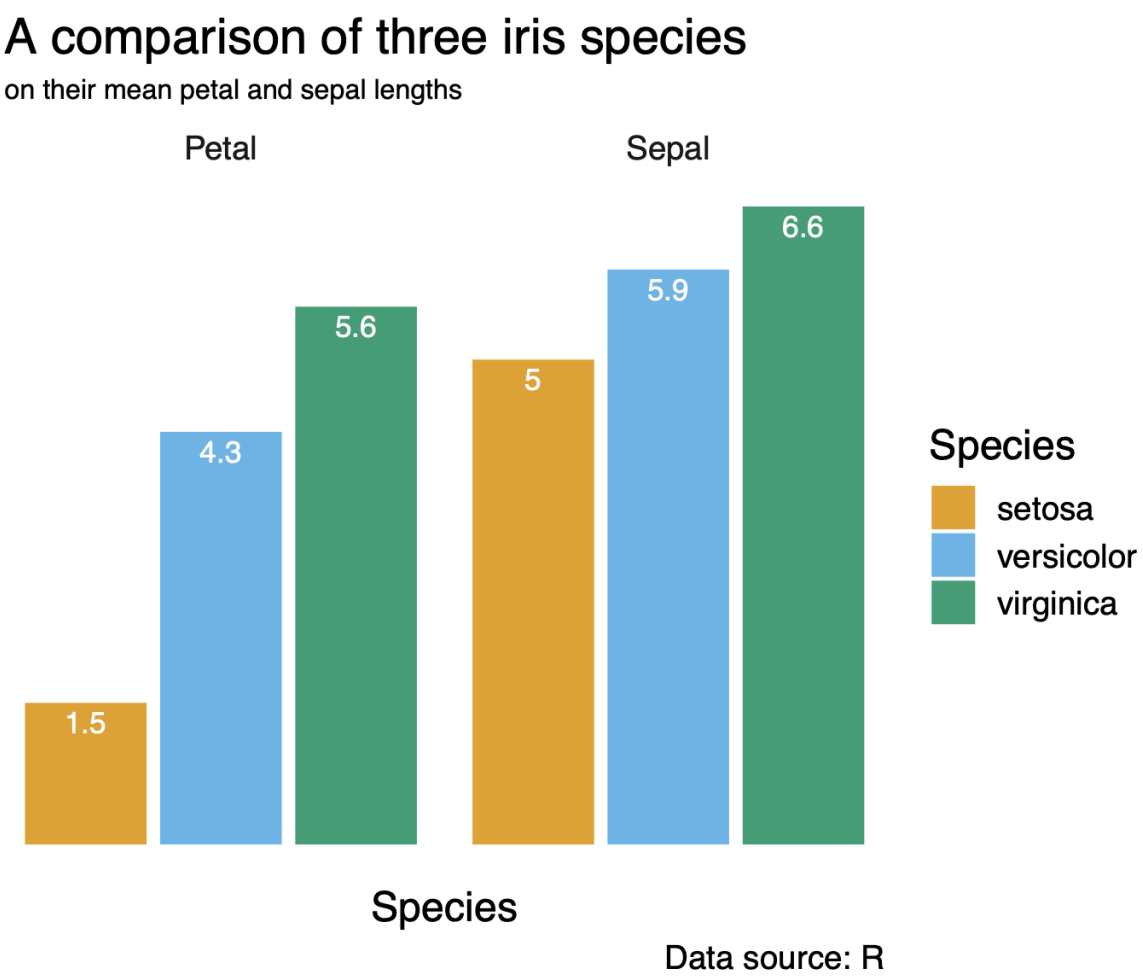


Figure 2: L3 Grouped bar chart plus facet grid

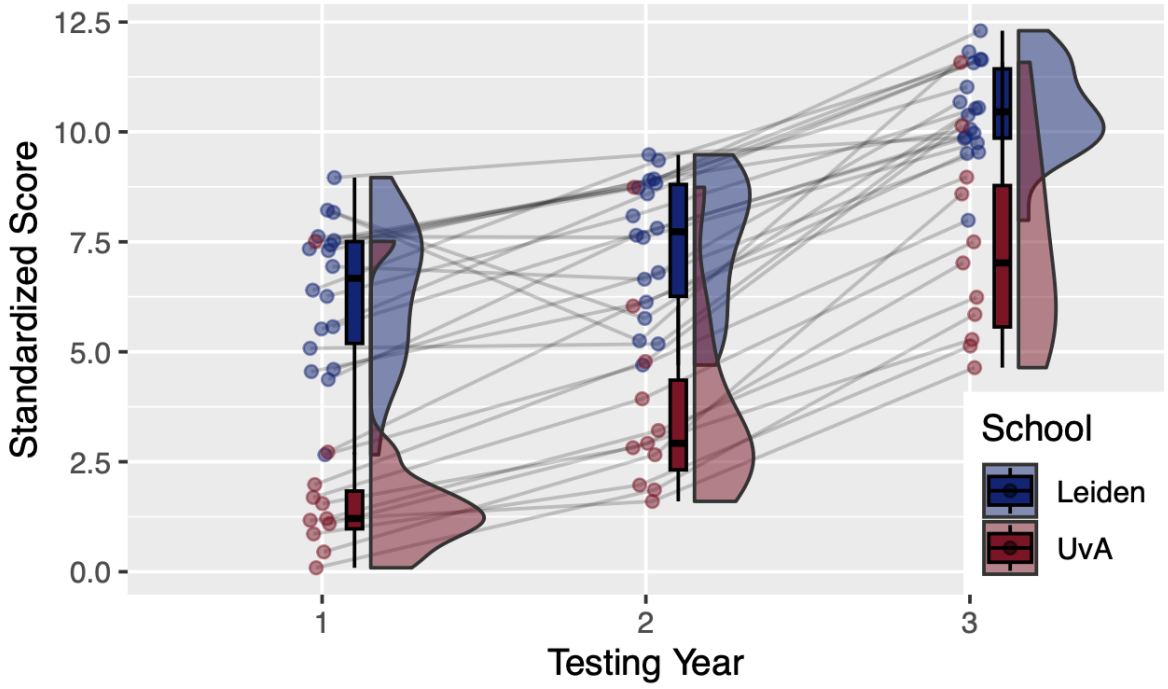


Figure 3: L4 Grouped distributions (Rain plot)

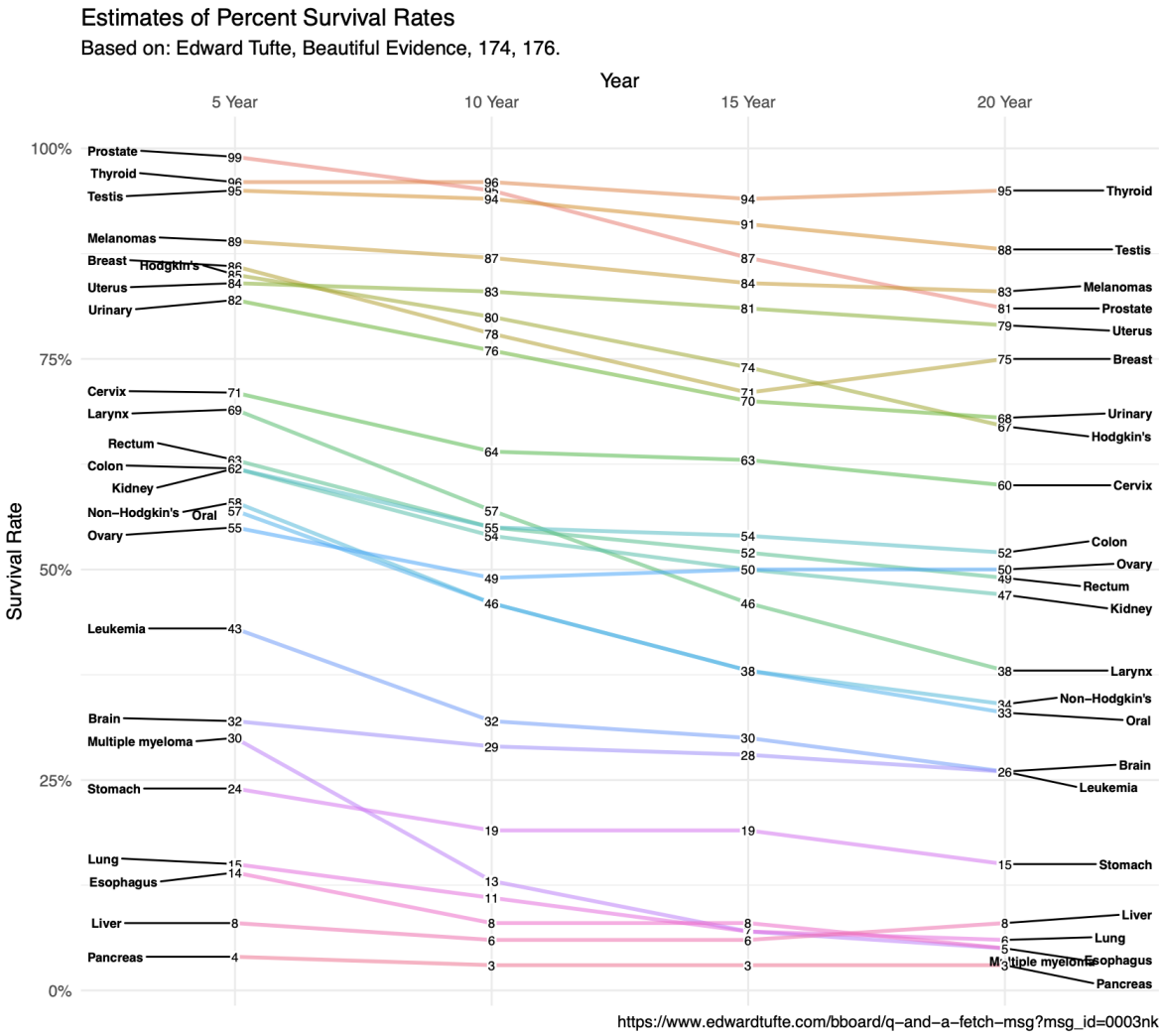


Figure 4: L6 plot 1: Slope graph (grouped trajectory plot)

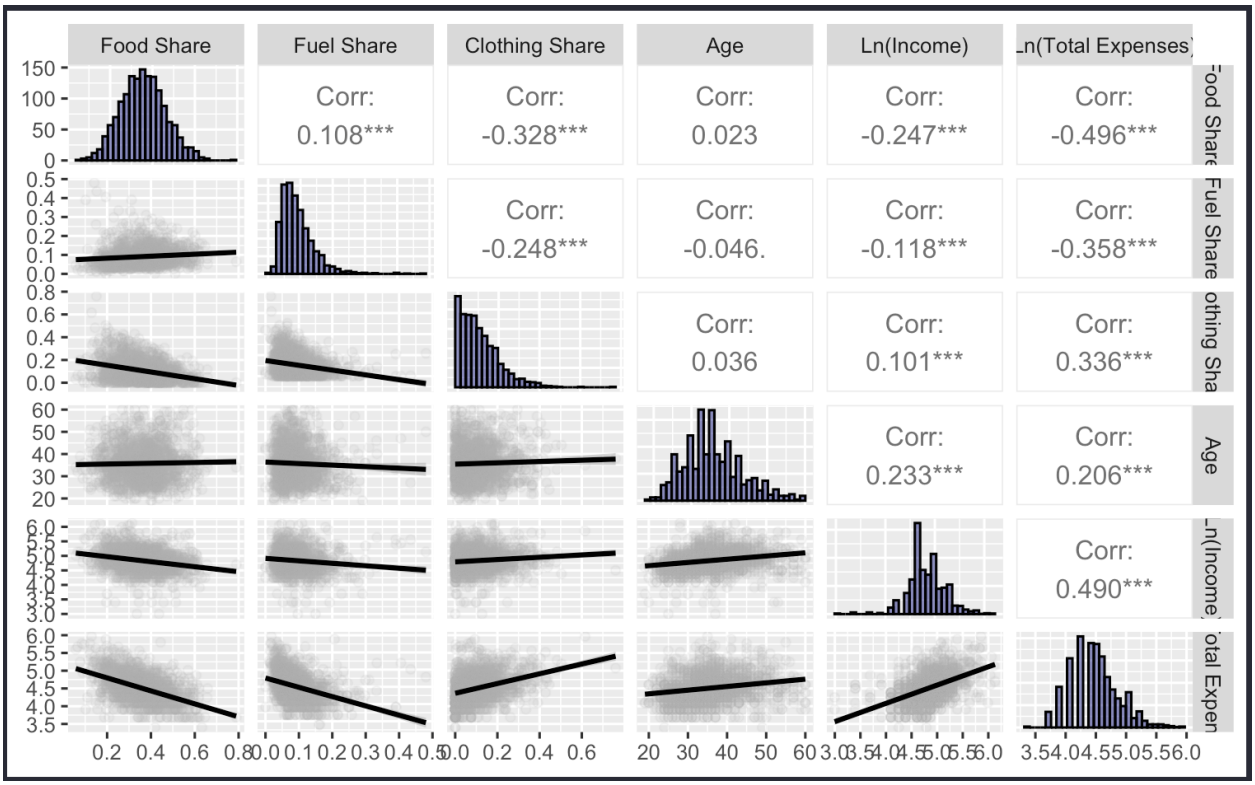


Figure 5: L6 plot 2: GGpairs with custom functions

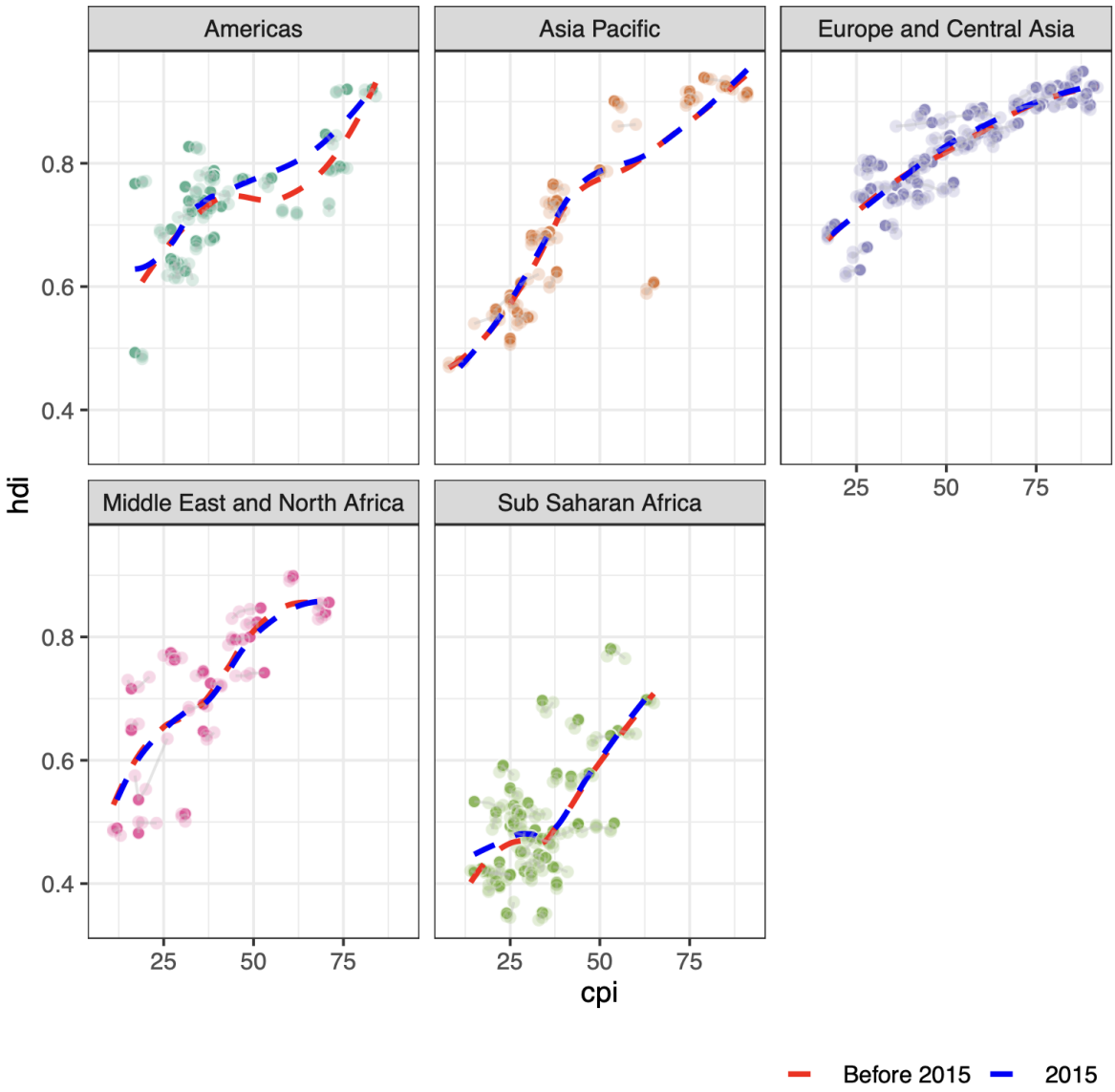


Figure 6: L7 plot 1: Scatter plot with trend line, facet wrap

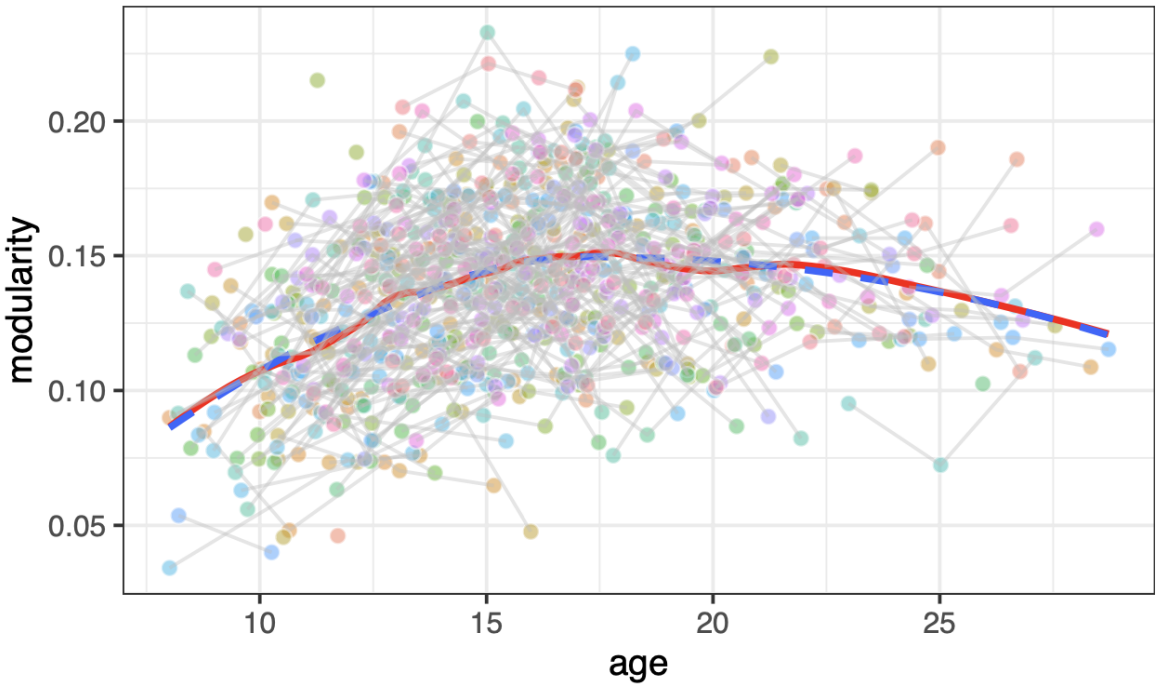
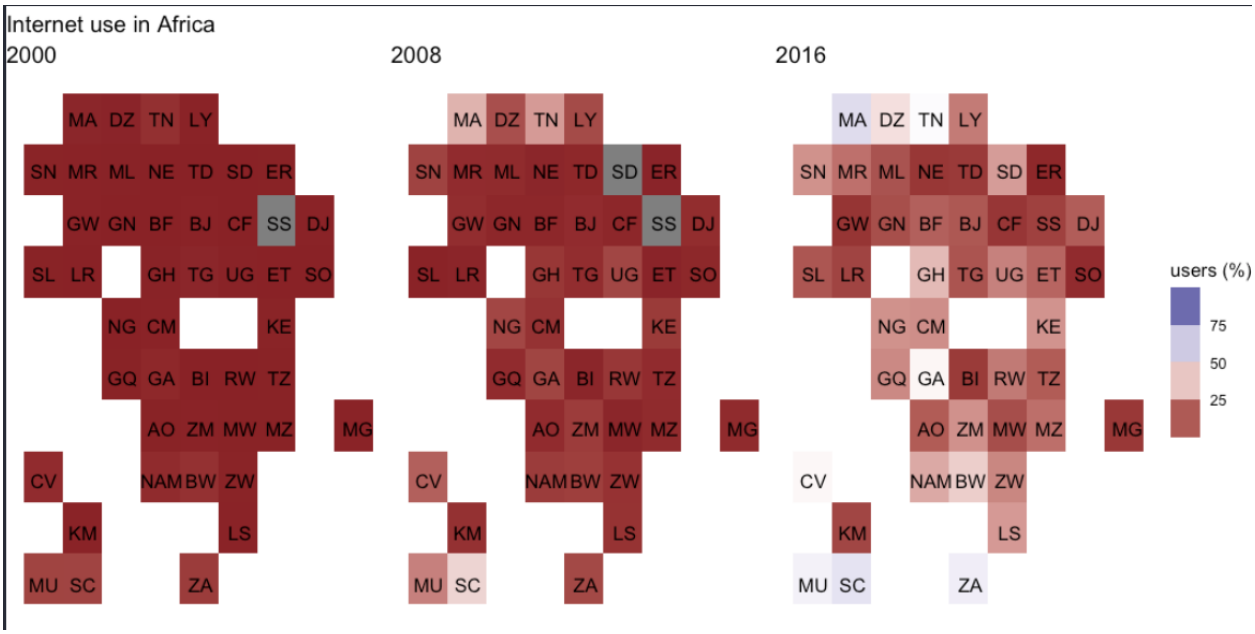


Figure 7: L7 plot 2: Scatter plot with trend line



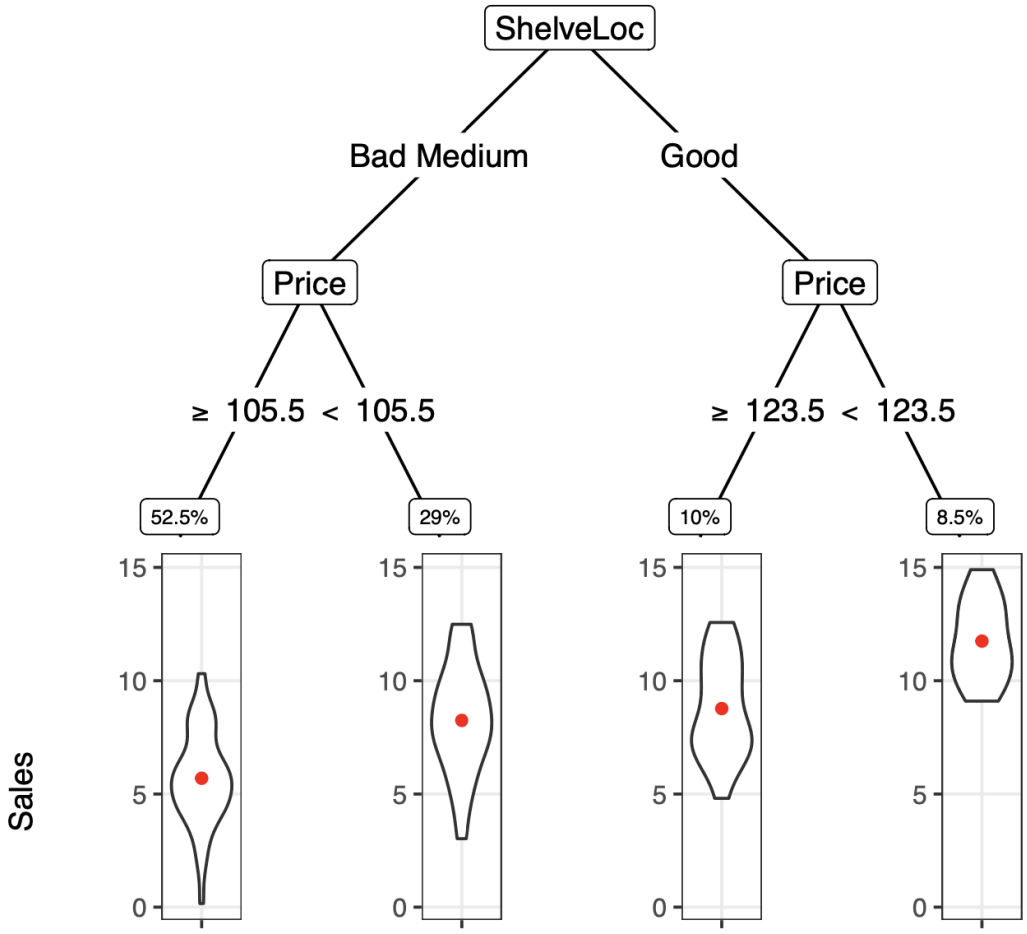


Figure 1: Regression tree.

L9 was a work group so on visualization here.

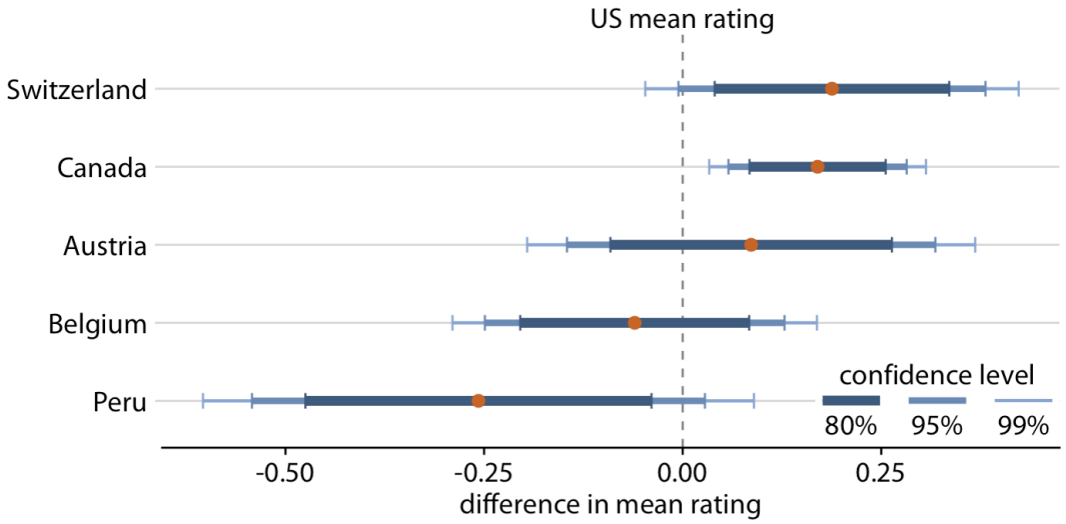


Figure 16.8: Mean chocolate flavor ratings for manufacturers from five different countries, relative to the mean rating of U.S. chocolate bars. Canadian chocolate bars are significantly higher rated that U.S. bars. For the other four countries there is no significant difference in mean rating to the U.S. at the 95% confidence level. Confidence levels have been adjusted for multiple comparisons using Dunnett's method. Data source: Brady Brelinski, Manhattan Chocolate Society

Figure 8: L11 Dot plot for uncertainty

L1

Theory only so no challenge

L2 Scatter, redundant coding

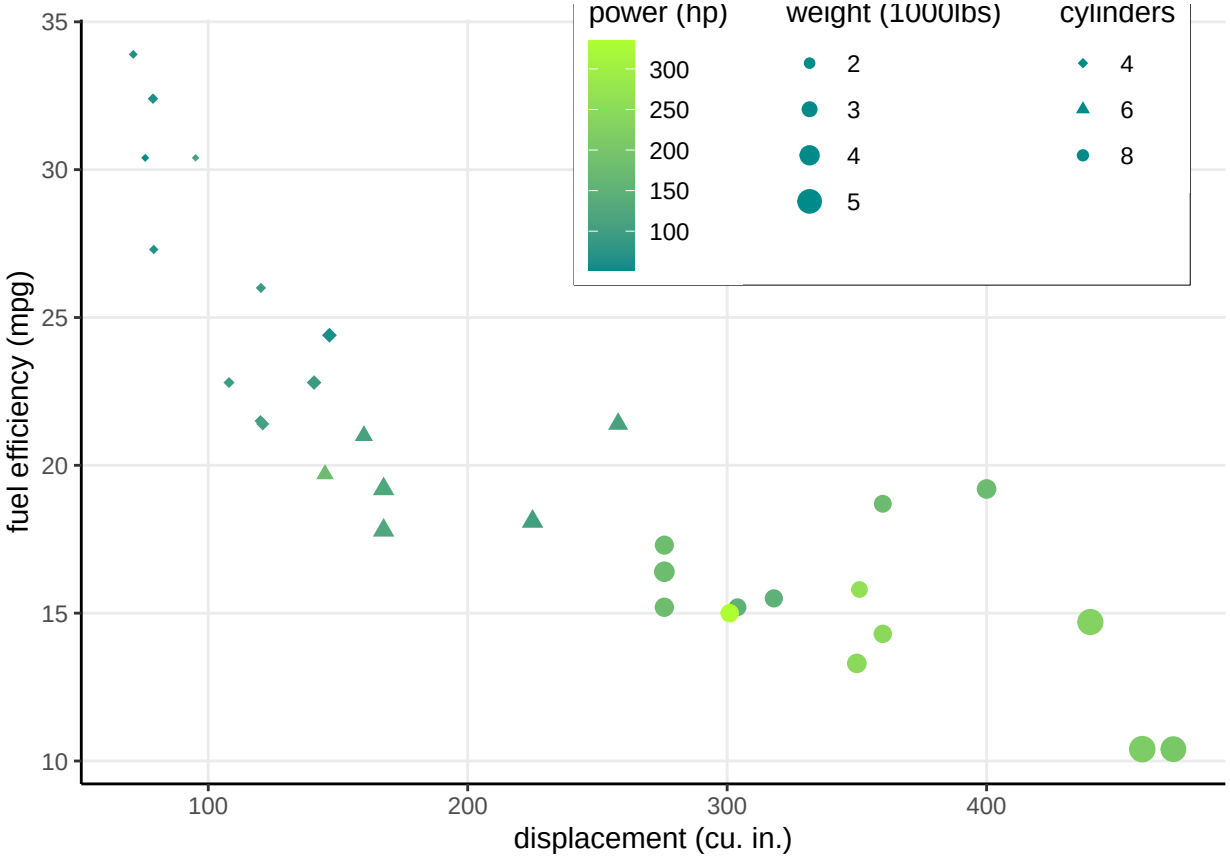
Challenge: Visualize plot 2.5 from the book (chapter 2.2)

Redundant coding: Manually adjust red. cod. scales.

Legend: Manually specify the position and the background and override aesthetics.

```
# ![L2 challange](./pics_for_challenges/L2_scatter_plot.png)
ggplot(data = mtcars,
       mapping = aes(x = disp,
                     y = mpg,
                     shape = as.factor(cyl),
                     color = hp,
                     size = wt)) +
  geom_point() + # no lines around the shapes
  labs(
    x = "displacement (cu. in.)",
    y = "fuel efficiency (mpg)",
    shape = "cylinders",
    color = "power (hp)",
    size = "weight (1000lbs)"
  ) +
  scale_shape_manual(values = c(18,17,19)) +
  scale_radius(range = c(1,4)) +
 scale_color_gradient(low = "darkcyan", high = "greenyellow") + # colors are approximate
theme_bw() +
  theme(legend.position = c(0.7,0.85),
        legend.box = "horizontal",
        legend.box.background = element_rect(fill = "white", #white background behind legend
                                             linewidth = 0),
        panel.border = element_blank(),
        panel.grid.minor = element_blank(),
        axis.line = element_line()) +
 guides(shape = guide_legend(order = 3, override.aes = list(color = "darkcyan")),
        color = guide_colorbar(order = 1),
        size = guide_legend(order = 2, override.aes = list(color = "darkcyan")))
```

```
## Warning: A numeric ‘legend.position‘ argument in ‘theme()‘ was deprecated in ggplot2
## 3.5.0.
## i Please use the ‘legend.position.inside‘ argument of ‘theme()‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```


power (hp)
300
250
200
150
100
weight (1000lbs)
2
3
4
5
cylinders
4
6
8
fuel efficiency (mpg)
35
30
25
20
15
10
100
200
300
400
displacement (cu. in.)

L3 Grouped bar chart plus facet grid

```
# Iris data to long format:
long_iris <- iris %>%
  gather(key= 'part', value = 'value', Sepal.Length, Sepal.Width, Petal.Length, Petal.Width) %>%
  separate(part,c('part','measure'), sep = '\\.')


head(iris)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

```
head(long_iris)
```

```
##   Species  part measure value
## 1  setosa Sepal  Length   5.1
## 2  setosa Sepal  Length   4.9
## 3  setosa Sepal  Length   4.7
## 4  setosa Sepal  Length   4.6
## 5  setosa Sepal  Length   5.0
## 6  setosa Sepal  Length   5.4
```

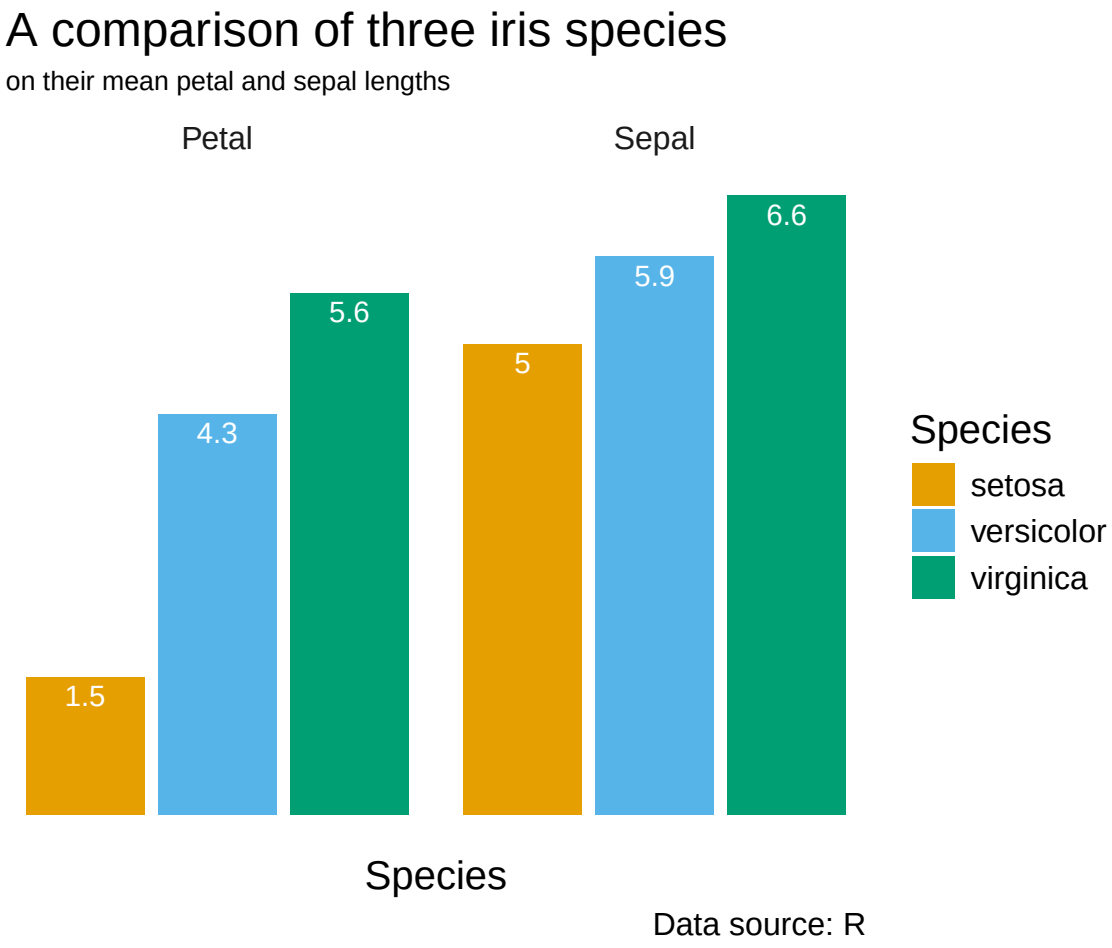
Bar chart: Group by species, Rest is not that important (Subset length, Stat summary uses mean)

Facet grid: Creates 2 grouped bar charts (petal vs sepal)

The rest is just aesthetics

```
ggplot(data = long_iris[which(long_iris$measure == "Length"),],
       mapping = aes(x = Species,
                     y = value,
                     fill = Species)) +
    geom_bar(stat = "summary",
             fun = "mean") +

    # split plot between parts (petal vs. sepal)
    facet_grid(cols = vars(part)) +

    # add values in bars:
    geom_text(mapping = aes(label = round(after_stat(y), digits = 1)),
              stat = "summary",
              fun = "mean",
              nudge_y = -0.2,
              color = "white",
              size = 4) +
```

```
  # change colors:
  scale_fill_okabeito() +

  # ad labels
  labs(x = "Species",
       y = NULL,
       title = "A comparison of three iris species",
       subtitle = "on their mean petal and sepal lengths",
       caption = "Data source: R") +

  # adjust theme elements:
  theme_minimal() +
  theme(axis.text.x=element_blank(), #remove x axis labels
        axis.ticks.x=element_blank(), # remove x axis ticks
        axis.text.y=element_blank(), #remove x axis labels
        axis.ticks.y=element_blank(), # remove y axis ticks
        text = element_text(size = 15), # enlarge text
        plot.subtitle = element_text(size = 10),
        panel.grid.major = element_blank(), # make subtitle smaller
        panel.grid.minor = element_blank())   #remove y axis labels
```



L4 Grouped distributions (Rain plot)

```
# Read data
eoy <- read.csv("data_for_challenges/end-of-year.csv", header=TRUE)
```

Rain plot: Cov is the covariate color, so the dots are colored by school

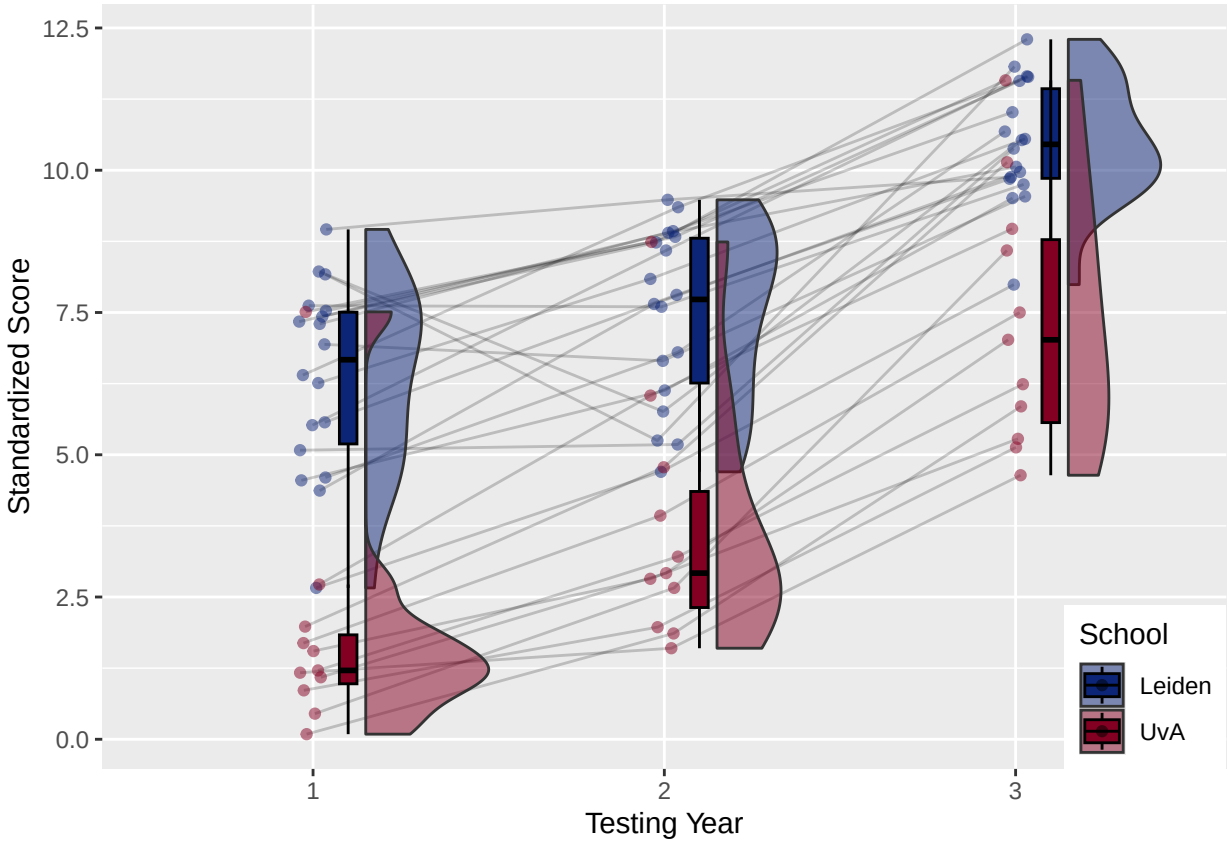
Rain plot: Id.long.var specifies that the dots are **connected**

```
# Need extra library for rain plot
library(ggrain)
```

```
## Registered S3 methods overwritten by 'ggpp':
##   method                  from
##   heightDetails.titleGrob ggplot2
##   widthDetails.titleGrob  ggplot2
```

```
ggplot(data=eoy, aes(x=factor(year), y=score, fill=school)) +

  # Manually set the box plot color
  # Set the covariate to school = points are colored by school
  # Id.long.var is id = CONNECTS the dots
  geom_rain(boxplot.args=list(color="black", outlier.shape=NA),
            cov="school",
            alpha=0.5,
            id.long.var="id") +

  # Scale color and fill
  scale_fill_manual(values=c("#0c2577","#840022")) +
  scale_color_manual(values=c("#0c2577","#840022")) +

  # Adjust legend position
  theme(legend.justification=c(1, 0), legend.position=c(1, 0)) +
  guides(color="none") +
  labs(x="Testing Year", y="Standardized Score", fill="School")
```

```
## Warning: Duplicated aesthetics after name standardisation: alpha
```



L5 (work group)

So no plots

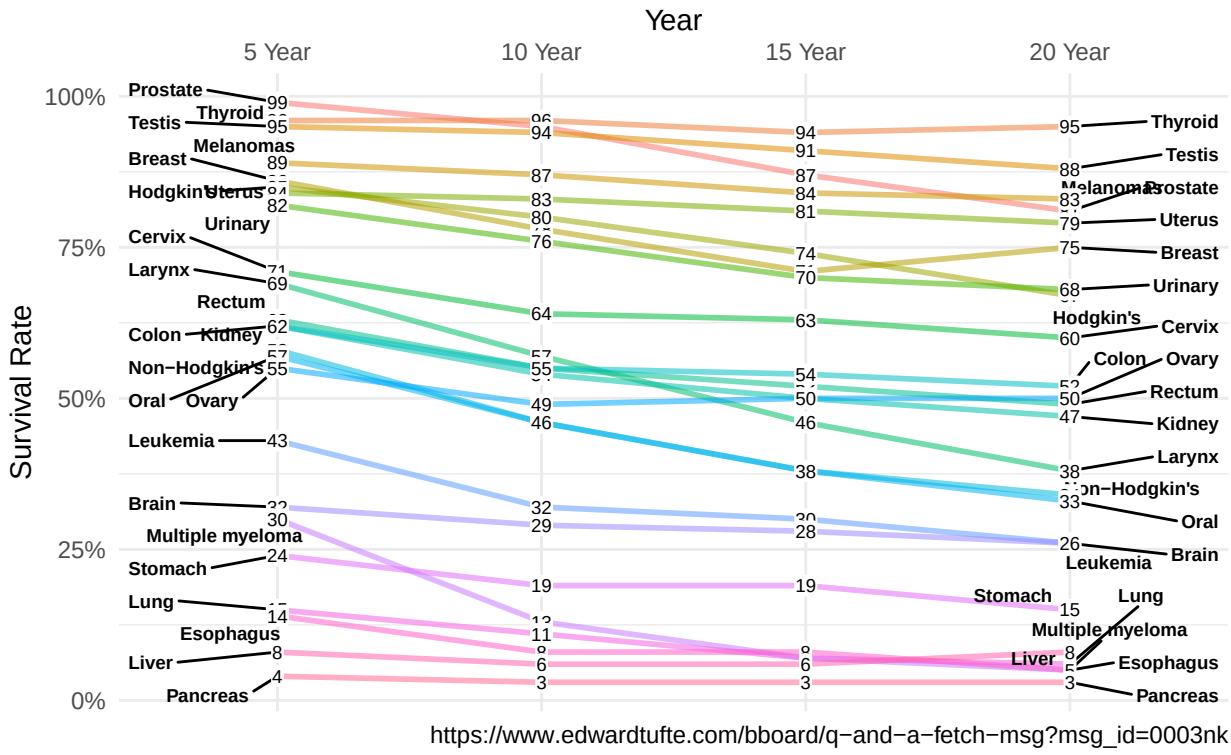
L6 Association

Slope graph (grouped trajectory plot)

```
# Load data
load("data_for_challenges/cancer.rda")
load("data_for_challenges/household.rda")
```

```
# Load additional library for lables called geom_text_repel
library(ggrepel)

ggplot(data = cancer, aes(x = Year, y = Survival, group = Type)) +

  # Lines grouped by type
  geom_line(aes(color = Type, alpha = 1), linewidth = 1) +

  # Text is based on the "Type" variables, subsetted on year == 5
```

```r
geom_text_repel(data = subset(cancer, Year == "5 Year"),
          aes(label = Type),
          size = 2.5,
          nudge_x = -0.5,
          fontface = "bold") +

# Text is based on the "Type" variables, subsetted on year == 20
geom_text_repel(data = subset(cancer, Year == "20 Year"),
          aes(label = Type),
          size = 2.5,
          nudge_x = 0.5,
          fontface = "bold") +

# Display numbers (Survival variable) at each time point ON THE LINE
geom_label(aes(label = Survival),
           size = 2.5,
           label.padding = unit(0.05, "lines"),
           label.size = 0.0) +
# Titles
labs(
  title = "Estimates of Percent Survival Rates",
  subtitle = "Based on: Edward Tufte, Beautiful Evidence, 174, 176.",
  caption = "https://www.edwardtufte.com/bboard/q-and-a-fetch-msg?msg_id=0003nk"
  ) +

# Make the axis pretty, e.g 100 percent
scale_x_discrete(position = "top") +
scale_y_continuous(name = "Survival Rate",
                   labels = scales::percent_format(scale = 1)) +

# Remove background grid and remove legend
theme_minimal()+
theme(legend.position = "none")
```



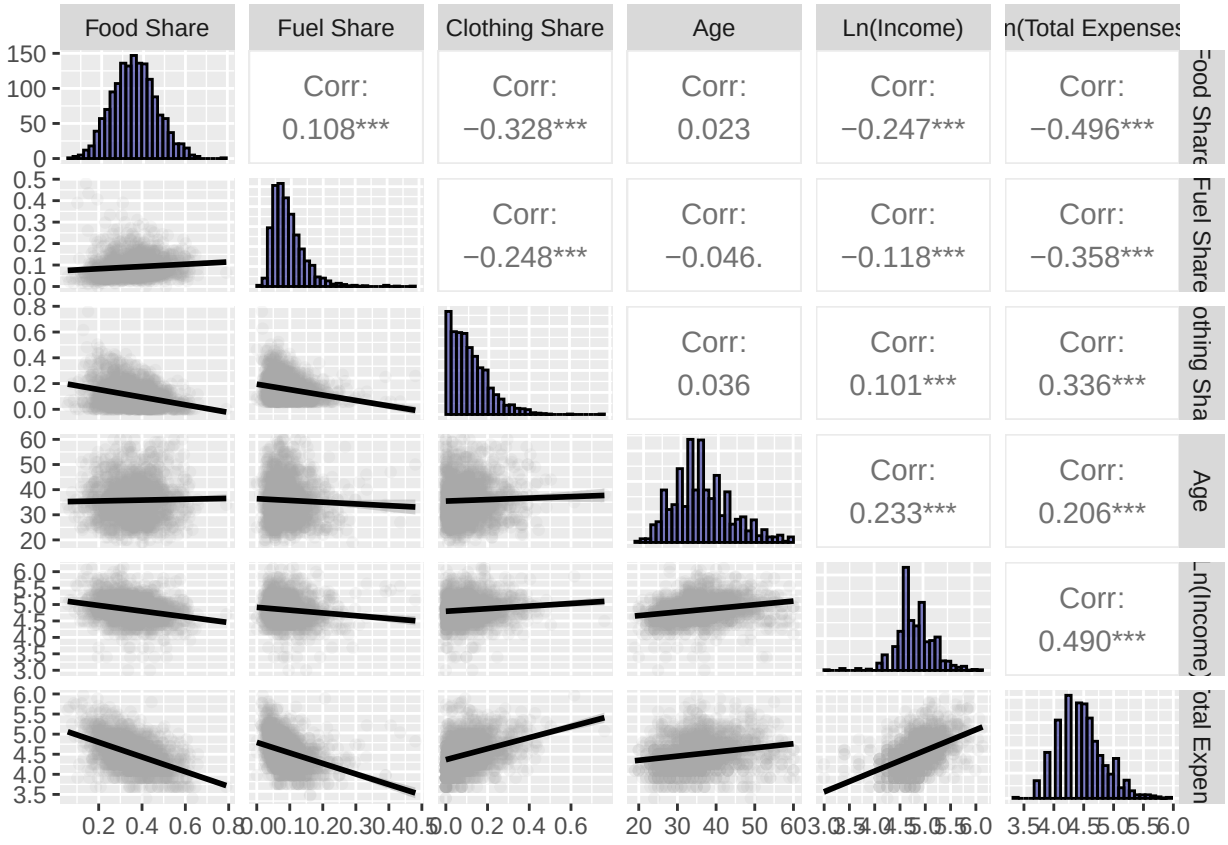
GGpairs with custom functions

```
# Needed for ggpairs
library(GGally)
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
# IMPORTANT: Arguments need to include "mapping" and "..."
my_scatter <- function(data, mapping, ...) {
  ggplot(data = data, mapping = mapping) +

    # Scatter plot
    geom_point(color='darkgrey', alpha=.1) +

    # Add regression line
    geom_smooth(method='lm', se=TRUE, formula='y ~ x', color='black') +

    # Make axis pretty
    scale_y_continuous(breaks=scales::pretty_breaks()) +
    scale_x_continuous(breaks=scales::pretty_breaks())
}
```

```
# IMPORTANT: Arguments need to include "mapping" and "..."
my_hist <- function(data, mapping, ...) {
  ggplot(data = data, mapping = mapping) +

    # Histogram with set number of bins
    geom_histogram(bins = 30,
                   color = 'black',
                   alpha=0.5,
                   fill = "darkblue") +

    # Make axis pretty
    scale_y_continuous(breaks=scales::pretty_breaks()) +
    scale_x_continuous(breaks=scales::pretty_breaks())
}

ggpairs(household,
        lower = list(continuous = my_scatter),
        diag = list(continuous = my_hist),
        columns = c(1:3, 7:9),
        columnLabels = c("Food Share", "Fuel Share", "Clothing Share",
                         "Age", "Ln(Income)", "Ln(Total Expenses)"))
```

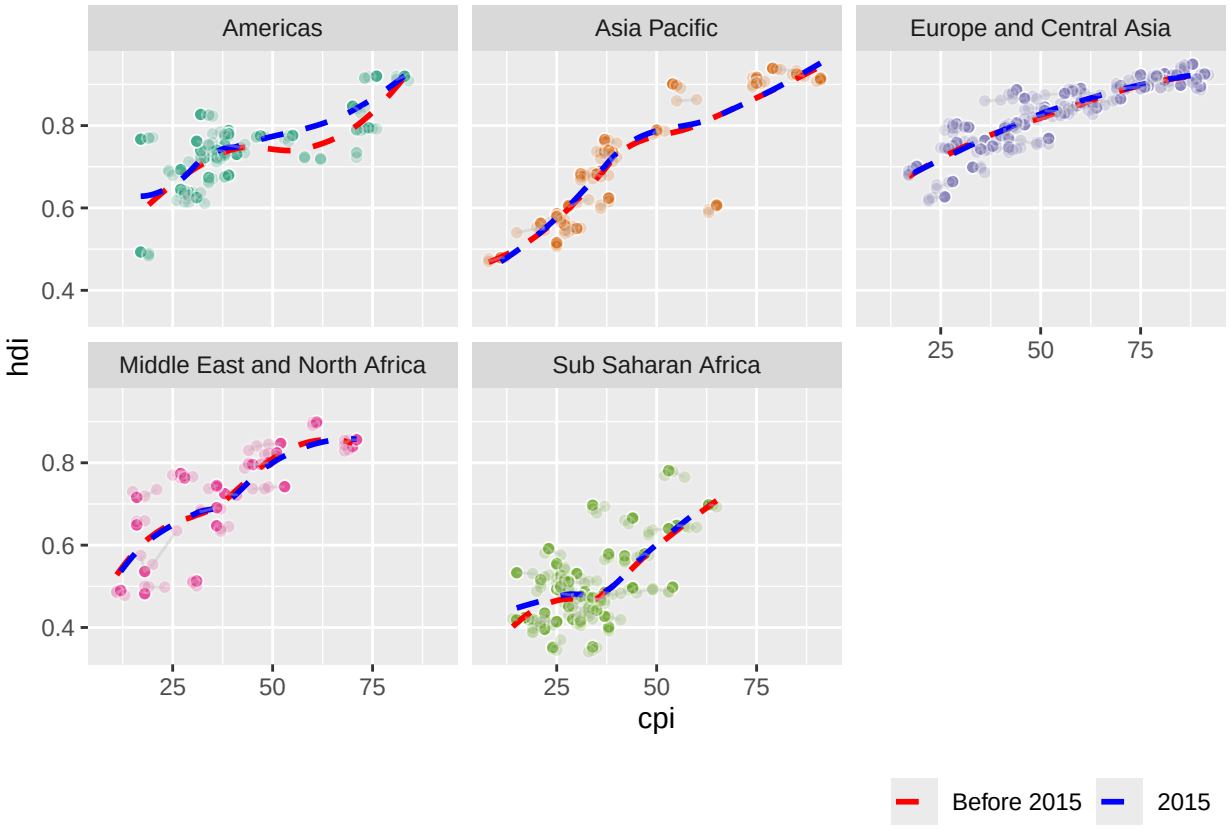


L7 Trends

Facet wrap

```
# colours from colourbewer
library(RColorBrewer)
region_cols <- brewer.pal(n = 5, name = "Dark2")


# Load the data
load("data_for_challenges/corruption.rda")

# Color is binary: Indicator for the year 2015
ggplot(corruption, aes(x = cpi, y = hdi,
                       colour = year == 2015)) +

  # Scatter plot grouped by "region"
  geom_point(aes(fill = region, alpha = year == 2015),
             colour = "white", pch = 21, size = 2)+

  # Add spline
  geom_smooth(method = "loess",
              span = 0.8,
              se = FALSE,
              linetype = 2) +


  geom_line(mapping = aes(group = country) ,
            colour = "grey", alpha = 0.4, show.legend = FALSE)+
  facet_wrap(~region) +

  # Remove the region legend
  scale_fill_manual(values = region_cols, guide = FALSE)  +

  # Update the alpha from the original (0, 1) range
  scale_alpha_discrete(range = c(0.2, 0.8), guide = FALSE) +

  # Spline color and legend position
  scale_colour_manual(values = c("red", "blue"),
                      name = NULL, labels = c("Before 2015", "2015") )  +
  theme(legend.position = "bottom",  legend.justification = "right")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

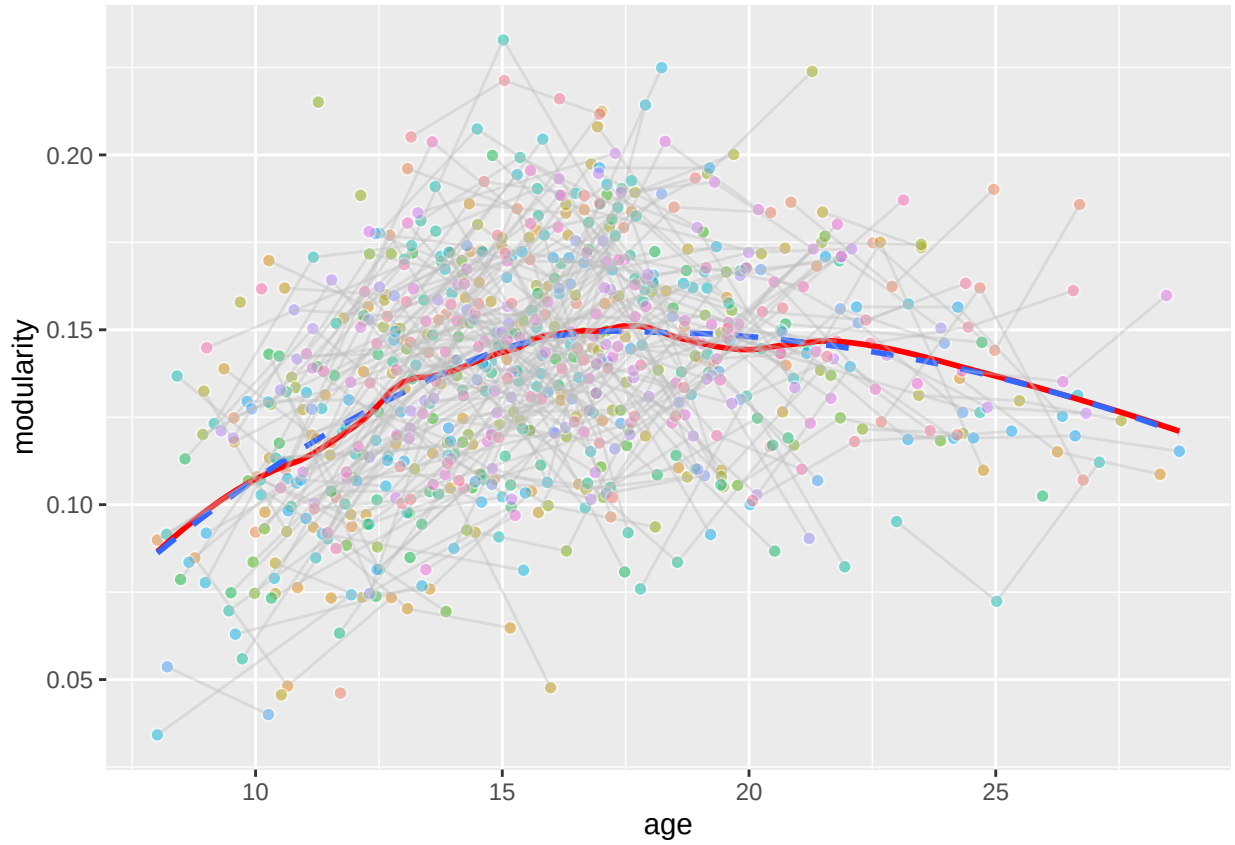


Multiple splines

```
load("data_for_challenges/learning.rda")

# For some reason you cannot specify fill in the main aes
ggplot(data = learn,
       mapping = aes(x=age, y=modularity ) ) +

  # Scatter plot: Fill on ID
  geom_point(mapping = aes(fill = factor(id)),
             colour = "white", pch = 21,
             size = 2, alpha=.5, show.legend = FALSE)+

  # First spline (solid red line)
  geom_smooth(method = "loess",
              span=0.3,
              se = FALSE,
              linetype = 1,
              color="red",
              alpha=.01) +

  # Second spline (dashed blue line)
  geom_smooth(method = "loess",
              span = 0.8,
```

```
              se = FALSE,
              linetype = 2) +

  # 3 data points per subject: Connect them via a line
  geom_line(mapping = aes(group = factor(id)),
            color="grey",
            alpha = 0.4, show.legend = FALSE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



L8 Geo

This map requires lots of data wrangling beforehand... Also, the geo facet library is required.

```
library(geofacet)

load("data_for_challenges/internet.rda")

# Only select those countries from the "internet" df that also have coordinates
## Coordinates are geofacet
africa_countries_grid1 <- geofacet::africa_countries_grid1
africa_internet<- subset(internet, country %in% africa_countries_grid1$name)
```

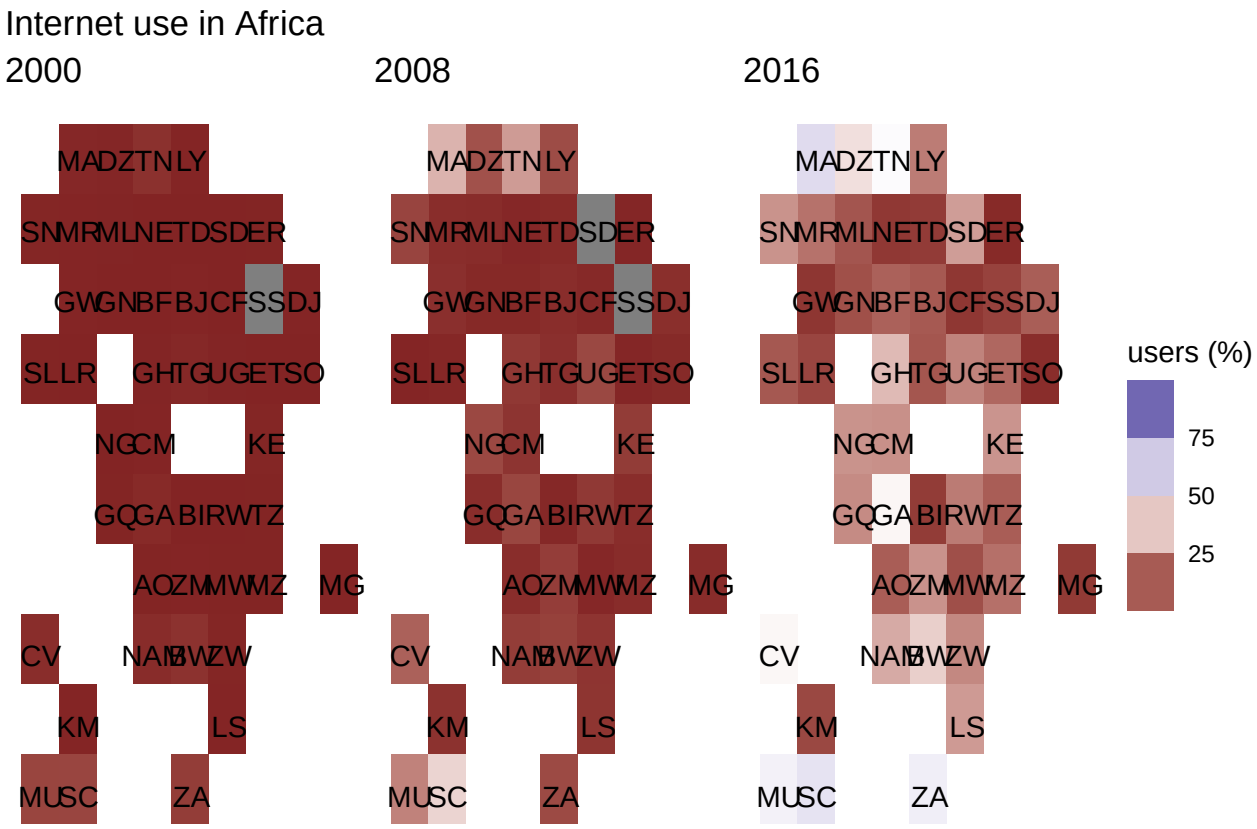
```r
# Merge to include columns with grid points (row + col)
africa_internet_heat <- left_join(africa_internet,
                                  africa_countries_grid1, by = c("country" = "name"))
```

Now we can actually plot. We create 3 different plots and the combine them.

IMPORTANT: The 10 by 9 grid captures all African countries.

```r
# Set theme void for nicer maps
theme_set(theme_void())

# All geo-spatial informaiton is stored in the "row/col" coordinates
africa_internet2000 <-
  ggplot(data = africa_internet_heat %>% filter(year == 2000),
       mapping = aes(x = col, y = -row, # because the grid starts top left
                     group = country, fill = users)) +
  geom_bin2d(binwidth = 0.99) + # so they fill the squares
  geom_text(mapping = aes(label = code),
            nudge_x = 0.5, nudge_y = -0.5) +
  scale_fill_gradient2(guide = "colorsteps",
                       name = "users (%)",
                       midpoint = 50,
                       limits = c(0,100)) +
  labs(title = "2000")

africa_internet2008 <-
  ggplot(data = africa_internet_heat %>% filter(year == 2008),
       mapping = aes(x = col, y = -row, # because the grid starts top left
                     group = country, fill = users)) +
  geom_bin2d(binwidth = 0.99) + # so they fill the squares
  geom_text(mapping = aes(label = code),
            nudge_x = 0.5, nudge_y = -0.5) +
  scale_fill_gradient2(guide = "colorsteps",
                       name = "users (%)",
                       midpoint = 50,
                       limits = c(0,100)) +
  labs(title = "2008")

africa_internet2016 <-
  ggplot(data = africa_internet_heat %>% filter(year == 2016),
       mapping = aes(x = col, y = -row, # because the grid starts top left
                     group = country, fill = users)) +
  geom_bin2d(binwidth = 0.99) + # so they fill the squares
  geom_text(mapping = aes(label = code),
            nudge_x = 0.5, nudge_y = -0.5) +
  scale_fill_gradient2(guide = "colorsteps",
                       midpoint = 50,
                       name = "users (%)",
                       limits = c(0,100)) +
  labs(title = "2016")

# This would just create 3 plots
# africa_internet2000
```

```
# africa_internet2008
# africa_internet2016

# Instead we put them on a single plot using the "plot_layout" function
library(patchwork)
africa_internet2000 + africa_internet2008 + africa_internet2016 +
  plot_layout(guides = "collect") +
  plot_annotation(title = "Internet use in Africa")
```




```
# Rese the theme
theme_set(theme_grey())
```

L9 Work group

L9 was a work group. Ignore L10 and L11 as they cover less important topics.

L10 Tree

First fit the model.

```
library(ISLR)
data("Carseats")
set.seed(9731)

library(rpart)
carseats.train <- Carseats[sample(1:nrow(Carseats),200),]
fit.tree.reg <- rpart(Sales ~., data=carseats.train, method="anova", cp=0.05)
```

Now plot it.

```
library(ggparty)
```

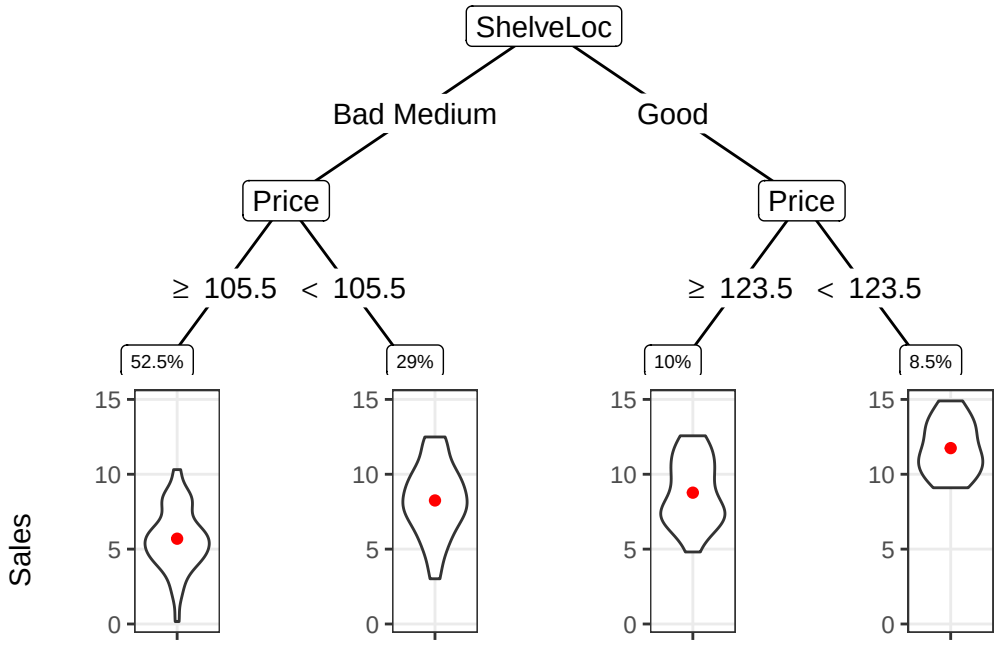
```
## Loading required package: partykit
```

```
## Loading required package: grid
```

```
## Loading required package: libcoin
```

```
## Loading required package: mvtnorm
```

```
# Some transformatio
party.tree<- as.party(fit.tree.reg)

ggparty(party.tree)+
  geom_edge()+
  # variable labels
  geom_node_label(aes(label = splitvar),
                  ids = "inner") +
  # split labels
  geom_edge_label()+

  # terminal nodes
  geom_node_label(aes(label = paste0(nodesize/200 *100, "%")),
                  ids = "terminal",
                  nudge_y = .02,
                  size = 2.5)+
  geom_node_plot(gglist =
                 list(aes(x = "", y=Sales),
                      geom_violin(),
                      stat_summary(fun = "mean",  geom = "point", colour = "red"),
                      theme_bw(),
                      theme(axis.title.x=element_blank(),
                            axis.text.x=element_blank(),
                            panel.grid.minor = element_blank()
                            )
                 ),
                shared_axis_labels = TRUE,
                scales = "free_x",
                width = .6, height = 0.8)
```



L11 Point estimate with CI

```
load("data_for_challenges/cacao.rda")

# data cleaning:
## select countries
cacao_fig16.8 <- filter(cacao,
                        cacao$location %in%
                          c("Austria", "Belgium", "Canada", "Peru",
                            "Switzerland"))

## calculate difference with US
USA_mean <- mean(cacao$rating[cacao$location == "U.S.A."], na.rm = TRUE)
cacao_fig16.8$difference <- cacao_fig16.8$rating - USA_mean

# Plot:

ggplot(cacao_fig16.8, aes(x=reorder(x=location, X=difference, FUN=mean),
                                        y=difference)) +
  # 99% error bar
  geom_errorbar(mapping=aes(color="99%"), stat="summary", fun.data="mean_se",
                  size=1, fun.args = list(mult = qnorm((1-.99)/2)),
                linewidth = 0.75, width = 0.2) +
```

```
 # 95% error bar
 geom_errorbar(mapping=aes(color="95%"), stat="summary", fun.data="mean_se",
                 size=1, fun.args = list(mult = qnorm((1-.95)/2)),
               linewidth = 1.5, width = 0.2) +

 # 80% error bar
 geom_errorbar(mapping=aes(color="80%"), stat="summary", fun.data="mean_se",
                 size=1, fun.args = list(mult = qnorm((1-.8)/2)),
               linewidth = 2.25, width = 0.2) +

 # Mean point estimate
 geom_point(stat="summary", fun="mean", size=3, colour = "#D55E00") +

 ylim(c(NA, 0.5))+
 xlab(NULL) +
 scale_y_continuous(name="difference in mean rating", breaks=c(-.5, -.25, 0, .25))+
 coord_flip()+

 geom_hline(yintercept = 0, linetype="dashed", color = "darkgrey")+
 annotate("text", x=5.5, y=0, label="US mean rating")+

 # Override the colors by linking the color as "factor" to a true color value
  scale_color_manual(
   name = "confidence level",
   values = c(
    `80%` = "darkblue",
    `95%` = "#0072B2",
    `99%` = "lightblue"),
   guide = guide_legend(
     direction = "horizontal",
     title.position = "top",
     label.position = "bottom",
     override.aes = list(linewidth = c(`80%` = 2.25,
                                       `95%` = 1.5,
                                       `99%` = 0.75)),
     keywidth = 2)
 ) +

theme_minimal()+
 theme(legend.position = c(0.9, 0.12), # update theme
       panel.grid.minor.x = element_blank(),
       panel.grid.major.x = element_blank(),
       axis.line.x = element_line(),
       axis.ticks.x = element_line(color = "black"),
       axis.ticks.length.x = unit(3, "pt"),
         legend.box.background = element_rect(color = "white")

       )
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
```

```
## generated.

## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```