Graphical visualisation with ggplot2

- **Geom functions are used to display the data in a multitude of ways**

  - **geom_point()**: scatterplot
  - **geom_smooth()**: smoothed conditional means
  - **geom_histogram()**: histogram
  - **geom_density()**: smoothed variant of the histogram
  - **geom_bar() / geom_col()**: bars with bases on the x-axis
  - **geom_line()**: lines
  - **geom_boxplot()**: boxes-and-whiskers

- **Geoms differ in the aesthetics they require:**
  - **geom_point()** requires both x and y values
  - **geom_histogram()** only requires x vales

- On the following slides we will give examples for each of the new ones

Graphical visualisation with ggplot2

- **geom_histogram()**: histogram
- It is used to plot the distribution of a single, continuous variable

```
>ggplot(data = plantData, aes(x = Sepal.Length)) + geom_histogram()

# important parameter: binwidth: change the width of bins
>ggplot(data = plantData, aes(x = Sepal.Length)) + geom_histogram(binwidth = 2)

# aes(fill = …): color the bars according to their groups
> bplot= ggplot(data = plantData, aes(x = Sepal.Length)) + geom_histogram(binwidth = 2, aes(fill = Species))

# scale_fill_manual(): change the color manully,
>bplot + scale_fill_manual(values=c("#999999", "#E69F00", "#56B4E9"))
```

Introduction to Machine Learning

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Graphical visualisation with ggplot2

- **geom_density()**: smoothed variant of the histogram
- It is used to plot the distribution of a single, continuous variable

```
>ggplot(data = plantData, aes(x = Sepal.Length) ) + geom_density()

# aes(color = …):  used to create multiple density curves
>ggplot(data = plantData, aes(x = Sepal.Length )) +geom_density(aes(color=Species))

#aes(fill = …): is not  usefull due to overlapping regions
>ggplot(data = plantData, aes(x = Sepal.Length )) + geom_density(aes(fill=Species))

# scale_color_manual(): change the color manully of lines and points,
>ggplot(data = plantData, aes(x = Sepal.Length )) +geom_density(aes(color=Species))
 + scale_color_manual(values=c("#999999", "#E69F00", "#56B4E9"))
```

Graphical visualisation with ggplot2

- **geom_bar()**: bars with bases on the x-axis
- It is used to plot the distribution of a single, discrete/categorical variable

  >ggplot(data = plantData, aes(x = Species)) + geom_bar()

  # aes(fill = …): color the bars according to their groups
  # color the bars using this parameter
  >ggplot(data = plantData, aes(x = Species)) + geom_bar(aes(fill=Species))

  # create a pie chart from a bar chart with a single bar using **coord_polar()**
  >ggplot(data = plantData, aes(x ="", fill= Species)) + geom_bar()+ coord_polar(theta = "y")
  # x is assigned nothing so that we get a single bar divided into the different species
  # theta indicates the variable to map the angle to (x or y)
- If you want the heights of bars to represent values in the data, use **geom_col()** instead.

  >ggplot(data = plantData, aes(x = Species, y = Petal.Length)) + geom_col()

Exercises with R: Graphical visualisation with ggplot2

Exercise:

- create random subsets with 75 sample size from plantData without replacement
- draw the graphics bar plot, pie chart and histogram again.

```
randomNumbers= sample(1:nrow(plantData), 75, replace=F)
newData= plantData[randomNumbers, ]
ggplot(data = newData, aes(x = Species)) + geom_bar(aes(fill=Species))
ggplot(data = newData, aes(x ="", fill= Species)) + geom_bar()+ coord_polar(theta = "y")
ggplot(data = newData, aes(x = Sepal.Length)) + geom_histogram(binwidth = 1, aes(fill=Species) )
```

Introduction to Machine Learning

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Exercises with R: Graphical visualisation with ggplot2

Exercise: Write a **function** that should create random subsets with different sample size from plantData with and without replacement. Further, draw the graphic (scatterplot, histogram or pie chart) selected by the user.

```
randomSubsets=function(dataSet, sampleSize, replacement, graphic){
    randomNumbers= sample(1:nrow(dataSet), sampleSize, replace=replacement)
    newData= dataSet[randomNumbers, ]

   if(graphic=="scatterplot"){
        ggplot(data = newData, aes(x = Sepal.Width,y = Sepal.Length)) + geom_point(aes(colour = Species))
   }
   else if (graphic=="histogram"){
        ggplot(data = newData, aes(x = Sepal.Length)) + geom_histogram(binwidth = 1, aes(fill=Species) )
   }
   else if (graphic=="piechart"){
      ggplot(data = newData, aes(x ="", fill= Species)) + geom_bar()+ coord_polar(theta = "y")
   }
}
randomSubsets(plantData, 75,FALSE, "piechart")
randomSubsets(plantData, 100, TRUE, "histogram")
```

Exercises with R: Graphical visualisation with ggplot2

- **geom_line()**: lines
- It is used to plot the distribution of two variables representing points
- We don't have an index variable in the dataset so we simulate one using the function ***seq_along(vector)***, which creates the integer sequence 1, 2, …, vector.length

```
>seq_along(c(4,7,12)) #[1] 1 2 3

>ggplot(data = plantData, aes(x = seq_along(Sepal.Length),y = Sepal.Length)) +
        geom_line()  # A single line is created connecting all points

 # aes(group = …):  group the points
>ggplot(data = plantData, aes(x = seq_along(Sepal.Length),y = Sepal.Length)) +
        geom_line(aes(group = Species))

#aes(color = …): colors each line differently
#aes(linetype = …): changes the type of the line for each group
>ggplot(data = plantData, aes(x = seq_along(Sepal.Length), y=Sepal.Length)) +
  geom_line(aes(group=Species, color=Species, linetype=Species))
```

Exercises with R: Graphical visualisation with ggplot2

- **geom_boxplot()**: boxes-and-whiskers
- It is used to plot the distribution of a variable grouped by another one and is useful for showing outliers and differences between the groups

```
>ggplot(data = plantData, aes(x = Species, y = Sepal.Length)) + geom_boxplot()

# aes(fill = …): and aes(color = …) : color the boxes based on their group
>ggplot(data = plantData, aes(x = Species, y = Sepal.Length)) + geom_boxplot(aes(fill = Species))

# outlier.colour, shape, and size: highlight the outlier
>boxPlot= ggplot(data = plantData, aes(x = Species, y=Sepal.Length))
>prm1=geom_boxplot(aes(fill=Species), outlier.colour="red", outlier.shape=8,  outlier.size=4)
>boxPlot+prm1

#rotate boxplot
>boxPlot+prm1 + coord_flip()

#Change the legend position
boxPlot+prm1 + theme(legend.position="top")  # position-> top, bottom, none
```

Exercises with R: Graphical visualisation with ggplot2

- **Exercise:** boxplots for the combination of two variables
  - Create a vector which contains the variables "**small**" and "**medium**" for plant sizes
  - Using **sample** function and **plant size vector**, create a **new vector with 150 random variables**
  - Insert this vector into the **plantData** data frame as a new column **size**
  - Using **interaction** function, combine the variables size and species in plantData and save it in a new column
  - Create a boxplot with respect to the combined variables and Sepal.Length of the plants

Exercises with R: Graphical visualisation with ggplot2

- **Exercise:** boxplots for the combination of two variables
  - Create a vector which contains the size of plants as “**small**” and “**medium**”

    ```
    >size=c("small", "medium")
    ```

  - Using **sample** function and **size of plants**, create a **new vector with 150** elements

    ```
    >rndSample=sample(size, 150, replace = T)
    ```

  - Insert this vector into the **plantData** data frame as a new column **size**

    ```
    >plantData$size=rndSample
    ```

  - Using **interaction** function, combine the variables size and Species in plantData and save it in a new column

    ```
    >plantData$comb= interaction(plantData$Species, plantData$size)
    ```

  - Create a boxplot with respect to the combined variables and Sepal.Length of the plants

    ```
    >ggplot(data = plantData, aes(x = comb, y=Sepal.Length)) +   geom_boxplot(aes(fill=Species))
    ```

Exercises with R: Graphical visualisation with ggplot2

- Additional functions for changing labels and axes
  - **lims()**: This function allows us to the restrict the range of the axes to certain values

    ```
    >ggplot(data = plantData, aes(x = seq_along(Sepal.Length),y = Sepal.Length, shape = Species)) +
     geom_point() + lims(x = c(50,120), y = c(5,7))
    ```

  - **labs()**: Give labels to the axes and also allows to add title, subtitle and caption

    ```
    >ggplot(data = plantData, aes(x = seq_along(Sepal.Length),y = Sepal.Length, shape = Species)) +
     geom_point() + labs(x = "X", y = "Y", title = "Title", subtitle = "Sub", caption = "Caption“)
    ```

  - Remove legends added by for example **aes(fill = …)** with **guides()**

    ```
    >ggplot(data = plantData, aes(x = Species, y = Sepal.Length)) +
      geom_boxplot(aes(fill = Species)) + guides(fill = “none”)
    ```

Introduction to Machine Learning

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Exercises with R: Graphical visualisation with ggplot2

- **Faceting** allows to split the plot into multiple small plots

```
#Facet_grid(): split the plot into multiple small plots
# for the row-splitting  -> put the variable before ~,
>ggplot(data = plantData, aes(x = comb, y=Sepal.Length)) +   geom_boxplot(aes(fill=Species))
+facet_grid(Species~.)

# for the column-splitting -> put the variable after ~
>ggplot(data=plantData, aes(x=size, y=Sepal.Length)) +   geom_boxplot(aes(fill=Species))
+facet_grid(~Species)
```

- ***Save the* Plots**

```
>p = ggplot(data = plantData, aes(x = Species, y = Sepal.Length)) + geom_boxplot() #assign it to a variable
>ggsave("myplot.jpg", plot = p)   # ggsave(): save a plot to a file
```

Introduction to Machine Learning

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Exercises with R: Graphical visualisation with ggplot2

Exercise: Write a **function** that should create random subsets with different sample size from plantData with replacement. Further,

- It should normalize the **Petal.Width and Petal.Length** values with min-max normalization

$$x_i^{norm} = \frac{x_i - min(x)}{max(x) - min(x)}$$

- It should draw multiple graphics on a page
  - You will need "**gridExtra**" package and **grid.arrange** function
  - **Graphic 1:** a scatter plot & use **red**, **black** and **orange** colors for the points according to species & **shape** the points of each species (the function is **shape** used with **aes**)
  - **Graphic 2:** a histogram for Petal.Length & use **red**, **black** and **orange** colors for bars according to species
  - **Graphic 3:** a density plot for Petal.Length with **red**, **black** and **orange** colors
  - **Graphic 4:** a pie chart for plant sizes with **red**, and **orange** colors
  - **Graphic 5 -6:** box plots with **red**, **black** and **orange** colors according to
    - plant sizes and Petal.Width as well as Petal.Length
    - split them in columns according to Species and highlight the outliers
- Save the graphics in a png file, its name is given by the user