Tidyverse course: ggplot Code

Table of Contents

[Introduction 2](#_Toc5036299)

[Scatterplot 3](#_Toc5036300)

[Basic graph 3](#_Toc5036301)

[with colours 4](#_Toc5036302)

[this also works 5](#_Toc5036303)

[with log: log axis 6](#_Toc5036304)

[with log: log values 6](#_Toc5036305)

[with bigger points and axes and graph titles 7](#_Toc5036306)

[change the general theme 8](#_Toc5036307)

[change colour scheme and ticks on x-axis and y-axis and improve legend 9](#_Toc5036308)

[change colours manually 9](#_Toc5036309)

[Exercise 1 10](#_Toc5036310)

[Stripchart 11](#_Toc5036311)

[jitter, bigger points and colours 12](#_Toc5036312)

[control of the jitter 13](#_Toc5036313)

[add a line for the mean and a title for the y-axis 13](#_Toc5036314)

[pretty changes 14](#_Toc5036315)

[change order of levels on x axis, center the title and different colour scheme 15](#_Toc5036316)

[Reading a file and pre-processing it. 17](#_Toc5036317)

[Histogram 18](#_Toc5036318)

[Histogram day 1 18](#_Toc5036319)

[Exercise 2: 22](#_Toc5036320)

[Boxplot 26](#_Toc5036321)

[Violinplot (beanplot) 27](#_Toc5036322)

[Barchart 30](#_Toc5036323)

[Linegraph 33](#_Toc5036324)

[Exercise 3: 35](#_Toc5036325)

[Exercise 4: 37](#_Toc5036326)

[Exercise 5: 39](#_Toc5036327)

[Stacked bar: categorical data 42](#_Toc5036328)

[Exercise 6 46](#_Toc5036329)

[To save a graph 48](#_Toc5036330)

# Introduction

This document contains all of the code shown in the slides for the ggplot section of the tidyverse course. A separate document has the answers to the exercises used in the course.

library("tidyverse")

## Warning: package 'tidyverse' was built under R version 3.5.3

## -- Attaching packages --------------------------------------------------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.0 v purrr 0.3.2   
## v tibble 2.0.1 v dplyr 0.8.0.1  
## v tidyr 0.8.3 v stringr 1.4.0   
## v readr 1.3.1 v forcats 0.4.0

## Warning: package 'ggplot2' was built under R version 3.5.3

## Warning: package 'tibble' was built under R version 3.5.3

## Warning: package 'tidyr' was built under R version 3.5.3

## Warning: package 'readr' was built under R version 3.5.3

## Warning: package 'purrr' was built under R version 3.5.3

## Warning: package 'dplyr' was built under R version 3.5.3

## Warning: package 'forcats' was built under R version 3.5.3

## -- Conflicts ------------------------------------------------------------------------------------ tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library("ggplot2")  
  
msleep

## # A tibble: 83 x 11  
## name genus vore order conservation sleep\_total sleep\_rem sleep\_cycle  
## <chr> <chr> <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 Chee~ Acin~ carni Carn~ lc 12.1 NA NA   
## 2 Owl ~ Aotus omni Prim~ <NA> 17 1.8 NA   
## 3 Moun~ Aplo~ herbi Rode~ nt 14.4 2.4 NA   
## 4 Grea~ Blar~ omni Sori~ lc 14.9 2.3 0.133  
## 5 Cow Bos herbi Arti~ domesticated 4 0.7 0.667  
## 6 Thre~ Brad~ herbi Pilo~ <NA> 14.4 2.2 0.767  
## 7 Nort~ Call~ carni Carn~ vu 8.7 1.4 0.383  
## 8 Vesp~ Calo~ <NA> Rode~ <NA> 7 NA NA   
## 9 Dog Canis carni Carn~ domesticated 10.1 2.9 0.333  
## 10 Roe ~ Capr~ herbi Arti~ lc 3 NA NA   
## # ... with 73 more rows, and 3 more variables: awake <dbl>, brainwt <dbl>,  
## # bodywt <dbl>

class(msleep)

## [1] "tbl\_df" "tbl" "data.frame"

Remove NA values from vore

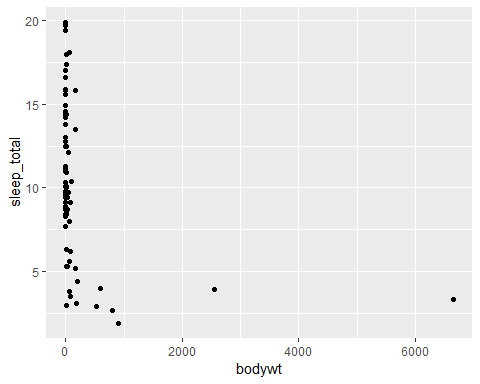
msleep %>% filter(!is.na(vore)) -> msleep.clean  
msleep.clean

## # A tibble: 76 x 11  
## name genus vore order conservation sleep\_total sleep\_rem sleep\_cycle  
## <chr> <chr> <chr> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 Chee~ Acin~ carni Carn~ lc 12.1 NA NA   
## 2 Owl ~ Aotus omni Prim~ <NA> 17 1.8 NA   
## 3 Moun~ Aplo~ herbi Rode~ nt 14.4 2.4 NA   
## 4 Grea~ Blar~ omni Sori~ lc 14.9 2.3 0.133  
## 5 Cow Bos herbi Arti~ domesticated 4 0.7 0.667  
## 6 Thre~ Brad~ herbi Pilo~ <NA> 14.4 2.2 0.767  
## 7 Nort~ Call~ carni Carn~ vu 8.7 1.4 0.383  
## 8 Dog Canis carni Carn~ domesticated 10.1 2.9 0.333  
## 9 Roe ~ Capr~ herbi Arti~ lc 3 NA NA   
## 10 Goat Capri herbi Arti~ lc 5.3 0.6 NA   
## # ... with 66 more rows, and 3 more variables: awake <dbl>, brainwt <dbl>,  
## # bodywt <dbl>

# Scatterplot

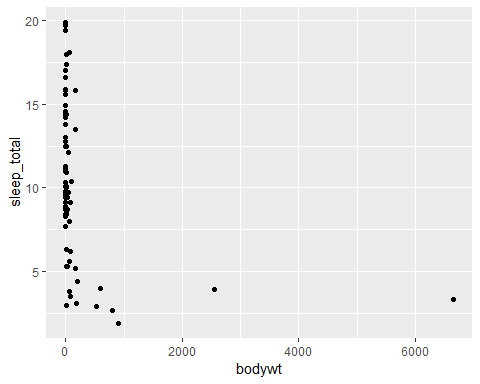
## Basic graph

ggplot(  
 msleep.clean,   
 aes(x=bodywt, y=sleep\_total)  
)+geom\_point() -> scatterplot  
  
scatterplot



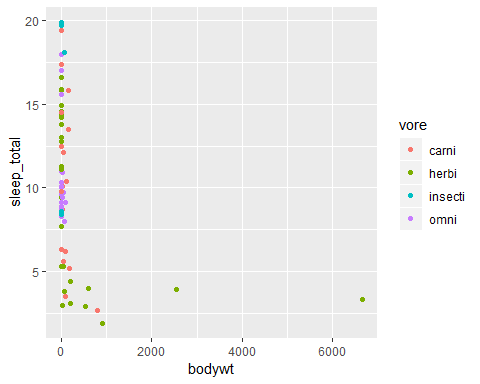
If we don’t want to save the filtered version we can pipe straight from the filter into ggplot without saving.

msleep %>%   
 filter(!is.na(vore)) %>%  
 ggplot(  
 aes(x=bodywt, y=sleep\_total)  
 )+geom\_point()



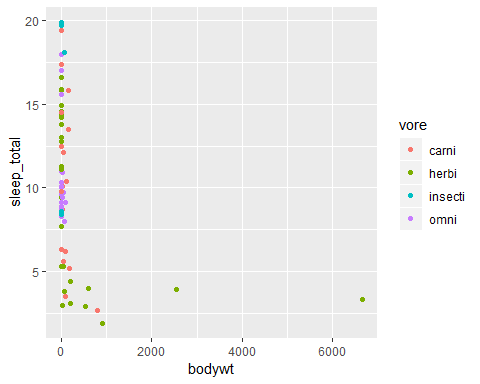
### with colours

ggplot(  
 msleep.clean,   
 aes(x=bodywt, y=sleep\_total, colour=vore)  
)+geom\_point()



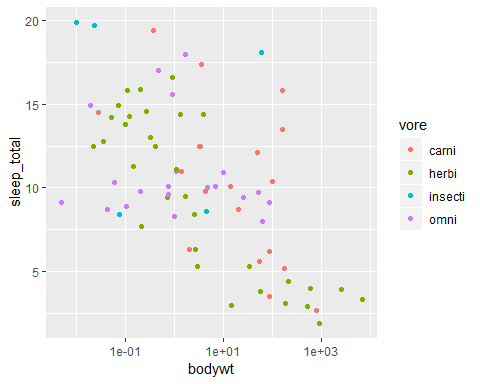
### this also works

ggplot(  
 msleep.clean,   
 aes(x=bodywt, y=sleep\_total)  
)+geom\_point(aes(colour=vore))



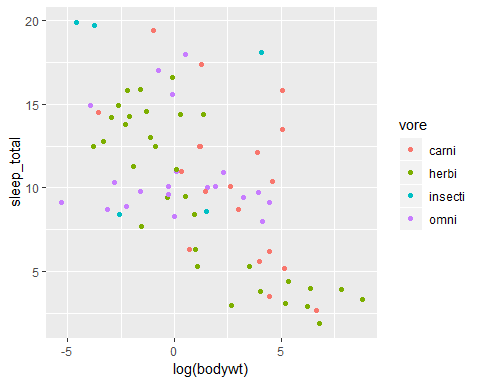
### with log: log axis

ggplot(  
 msleep.clean,   
 aes(x=bodywt, y=sleep\_total, colour=vore)  
)+geom\_point() -> scatterplot  
  
scatterplot+scale\_x\_log10()



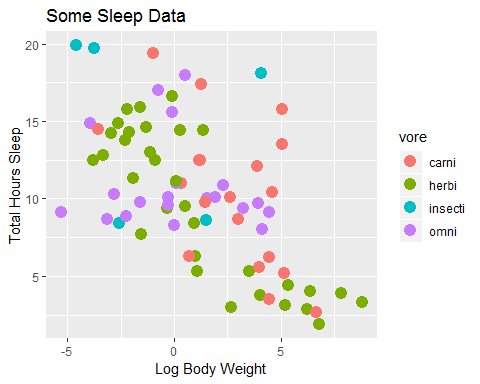
### with log: log values

ggplot(  
 msleep.clean,   
 aes(x=log(bodywt), y=sleep\_total,colour=vore)  
)+geom\_point()



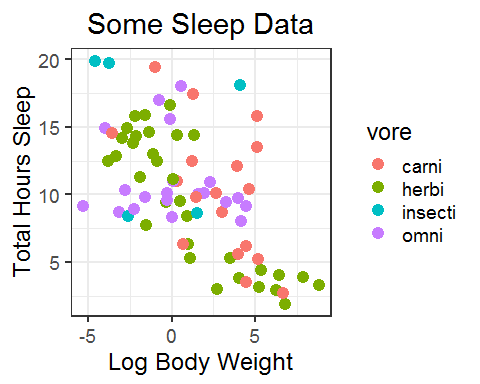
### with bigger points and axes and graph titles

ggplot(  
 msleep.clean,   
 aes(x=log(bodywt), y=sleep\_total,colour=vore)  
) +  
 geom\_point(size=4) +  
 xlab("Log Body Weight") +   
 ylab("Total Hours Sleep") +   
 ggtitle("Some Sleep Data") -> scatterplot  
  
scatterplot



### change the general theme

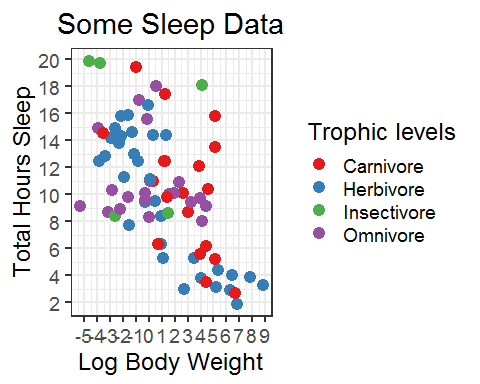
theme\_set(theme\_bw(base\_size=18))  
  
scatterplot+theme(plot.title = element\_text(hjust = 0.5)) -> scatterplot  
scatterplot



### change colour scheme and ticks on x-axis and y-axis and improve legend

This adds to the previous plot rather than recreating it.

scatterplot +  
 scale\_colour\_brewer(  
 palette="Set1",   
 name="Trophic levels",   
 labels=c("Carnivore", "Herbivore", "Insectivore", "Omnivore")  
 ) +  
 scale\_x\_continuous(breaks=-5:10) +  
 scale\_y\_continuous(breaks=seq(0,20, 2)) -> scatterplot  
scatterplot

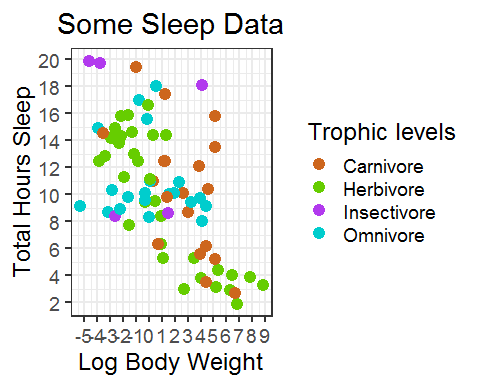


### change colours manually

scatterplot +  
 scale\_color\_manual(  
 values=c("chocolate3", "chartreuse3", "darkorchid2","cyan3"),  
 name="Trophic levels",   
 labels=c("Carnivore", "Herbivore", "Insectivore", "Omnivore")  
 ) -> scatterplot

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

scatterplot



Tadah! So beautiful :-)

## Exercise 1

The file up\_down\_expression.txt contains an expression comparison dataset with an extra column which classifies the rows into one of 3 groups (up, down or unchanging).

Load: “up\_down\_expression.txt” Check the structure of the file Plot a scatterplot geom\_point() with: up in red, down in blue unchanging in grey Main title: “Expression data” Colour legend: “Down, Unchanging and Up” Axis labels: “Condition 1” and “Condition 2”

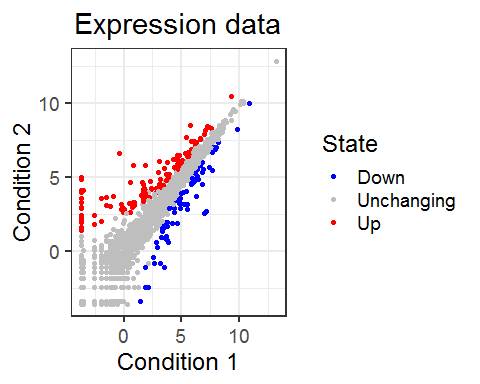
expression <- read\_tsv("up\_down\_expression.txt")

## Parsed with column specification:  
## cols(  
## Gene = col\_character(),  
## Condition1 = col\_double(),  
## Condition2 = col\_double(),  
## State = col\_character()  
## )

expression

## # A tibble: 5,196 x 4  
## Gene Condition1 Condition2 State   
## <chr> <dbl> <dbl> <chr>   
## 1 A4GNT -3.68 -3.44 unchanging  
## 2 AAAS 4.55 4.39 unchanging  
## 3 AASDH 3.72 3.48 unchanging  
## 4 AATF 5.08 5.02 unchanging  
## 5 AATK 0.471 0.560 unchanging  
## 6 AB015752.4 -3.68 -3.59 unchanging  
## 7 ABCA7 3.45 3.83 unchanging  
## 8 ABCA9-AS1 -3.68 -3.59 unchanging  
## 9 ABCC11 -3.53 -1.86 unchanging  
## 10 ABCC3 0.931 3.26 up   
## # ... with 5,186 more rows

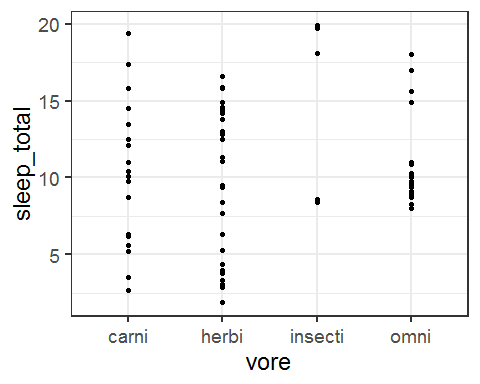
expression.scatter<-ggplot(expression, aes(Condition1, Condition2, colour=State))+  
 geom\_point()+  
 scale\_colour\_manual(values=c("blue", "grey", "red"),  
 name="State",   
 labels=c("Down", "Unchanging", "Up"))+  
 xlab("Condition 1") +   
 ylab("Condition 2") +   
 ggtitle("Expression data")+  
 theme(plot.title = element\_text(hjust = 0.5))  
   
expression.scatter



Now, let’s try another type a graph: a stripchart. It is similar to a scatterplot but the x variable is qualitative or categorical in nature.

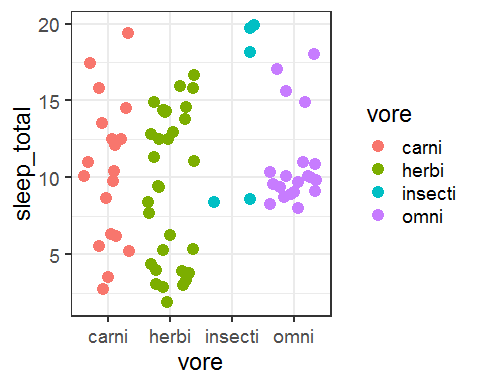
# Stripchart

ggplot(  
 msleep.clean,   
 aes(vore, sleep\_total)  
 )+geom\_point()



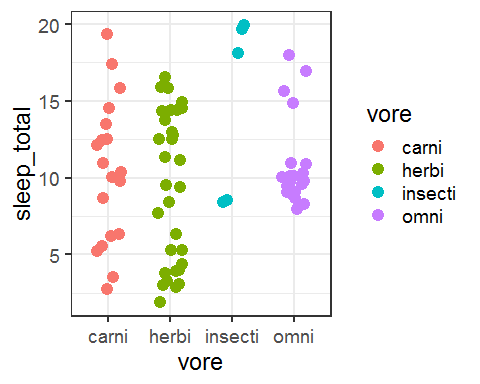
### jitter, bigger points and colours

ggplot(  
 msleep.clean,  
 aes(vore,sleep\_total, colour=vore)  
) + geom\_point(size=4,position="jitter")



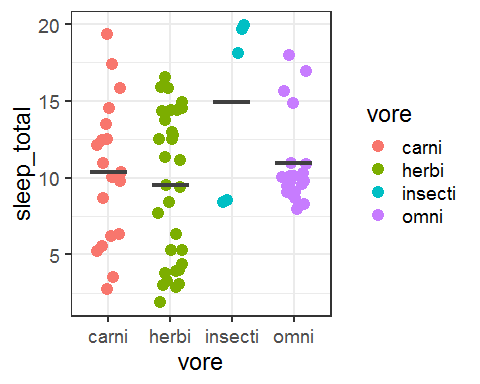
### control of the jitter

ggplot(  
 msleep.clean,   
 aes(vore, sleep\_total, colour=vore)  
) +  
 geom\_jitter(  
 width = .2,  
 size=4  
 ) -> stripchart  
  
stripchart



### add a line for the mean and a title for the y-axis

stripchart +  
 stat\_summary(  
 fun.y="mean",  
 geom='errorbar',   
 aes(ymin=..y.., ymax=..y..),   
 width=0.6,   
 size=1.5,  
 colour="grey25"  
 ) -> stripchart  
  
stripchart



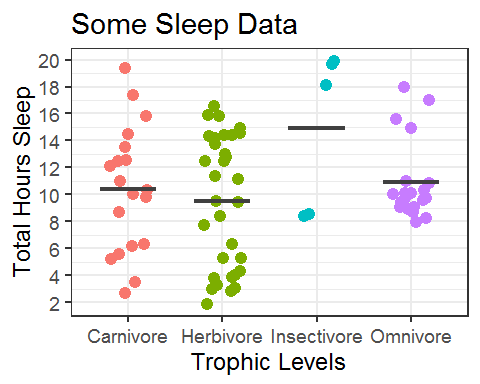
Little piece of code to calculate mean and SEm the tidyverse way.

msleep.clean %>%  
 group\_by(vore) %>%  
 summarise(sleep=mean(sleep\_total), sem=sd(sleep\_total)/sqrt(n()))

## # A tibble: 4 x 3  
## vore sleep sem  
## <chr> <dbl> <dbl>  
## 1 carni 10.4 1.07   
## 2 herbi 9.51 0.862  
## 3 insecti 14.9 2.65   
## 4 omni 10.9 0.659

### pretty changes

stripchart +  
 ylab("Total Hours Sleep") +  
 xlab("Trophic Levels") +  
 ggtitle("Some Sleep Data") +  
 scale\_y\_continuous(breaks=seq(0, 20, 2)) +  
 scale\_x\_discrete(labels=c("Carnivore", "Herbivore", "Insectivore", "Omnivore")) +  
 theme(legend.position = "none") -> stripchart  
  
stripchart

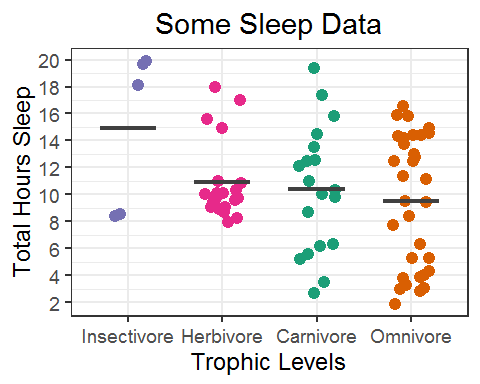


### change order of levels on x axis, center the title and different colour scheme

stripchart +  
 scale\_colour\_brewer(palette="Dark2")+  
 scale\_x\_discrete(  
 limit=c("insecti","omni","carni", "herbi"),  
 labels=c("Insectivore", "Herbivore", "Carnivore", "Omnivore"))+  
 theme(plot.title = element\_text(hjust = 0.5)  
 ) -> stripchart

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

stripchart

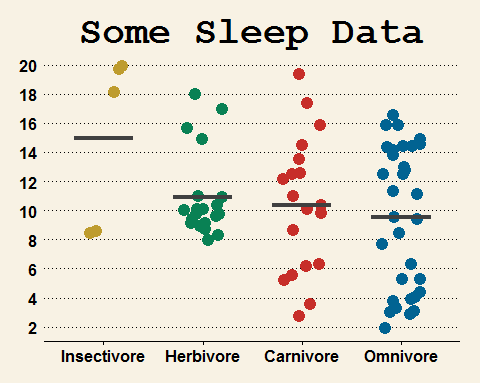


library("ggthemes")

## Warning: package 'ggthemes' was built under R version 3.5.3

stripchart+  
 theme\_wsj()+  
 scale\_colour\_wsj("colors6")+  
 theme(legend.position = "none")+  
 theme(plot.title = element\_text(hjust = 0.5))

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



Now, let’s try some other data. DownloadFestival datacontains the hygiene scores (0-5) of 810 concert goers over three days of music festival.

### Reading a file and pre-processing it.

read\_csv("DownloadFestival.csv") -> festival.data

## Parsed with column specification:  
## cols(  
## ticknumb = col\_double(),  
## gender = col\_character(),  
## day1 = col\_double(),  
## day2 = col\_double(),  
## day3 = col\_double()  
## )

festival.data

## # A tibble: 810 x 5  
## ticknumb gender day1 day2 day3  
## <dbl> <chr> <dbl> <dbl> <dbl>  
## 1 2111 Male 2.64 1.35 1.61   
## 2 2229 Female 0.97 1.41 0.290  
## 3 2338 Male 0.84 NA NA   
## 4 2384 Female 3.03 NA NA   
## 5 2401 Female 0.88 0.08 NA   
## 6 2405 Male 0.85 NA NA   
## 7 2467 Female 1.56 NA NA   
## 8 2478 Female 3.02 NA NA   
## 9 2490 Male 2.29 NA NA   
## 10 2504 Female 1.11 0.44 0.55   
## # ... with 800 more rows

max(festival.data$day1)

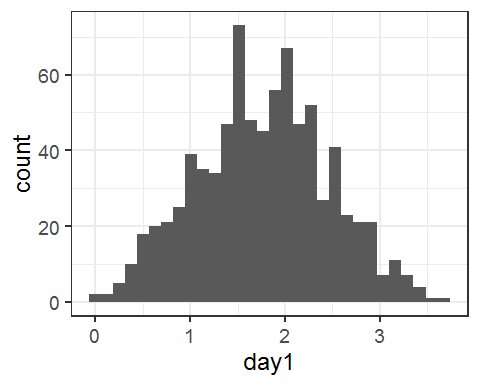
## [1] 3.69

# Histogram

### Histogram day 1

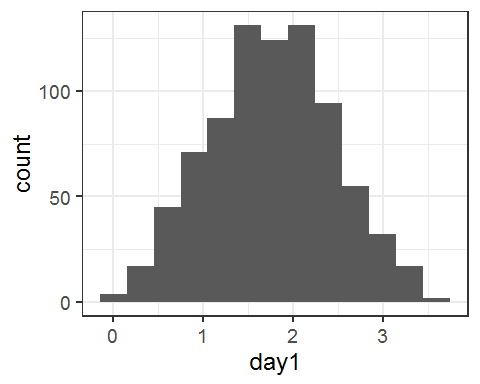
ggplot(  
 festival.data,   
 aes(day1)  
)+geom\_histogram()

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



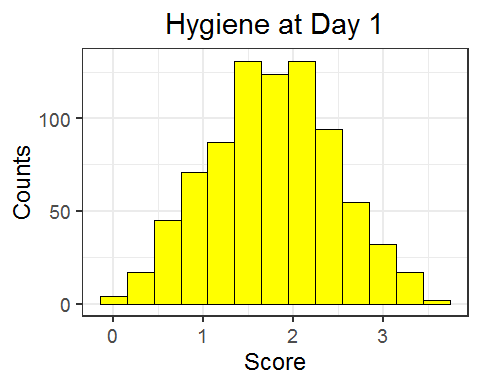
Let’s make it prettier.

ggplot(  
 festival.data,   
 aes(day1)  
)+geom\_histogram(binwidth=0.3)



Now change colour and add titles to axes.

ggplot(  
 festival.data,   
 aes(day1)  
)+geom\_histogram(binwidth=0.3, color="black", fill="yellow")+  
 labs(x="Score", y="Counts")+  
 theme(plot.title = element\_text(hjust = 0.5))+  
 ggtitle("Hygiene at Day 1") -> Day1Histogram  
Day1Histogram

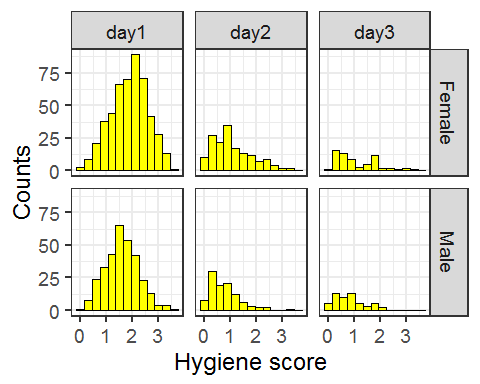


Now we want to plot all 3 days per gender. So we need to reshape the file. We are also going to remove the NAs.

festival.data %>%  
 gather(day,score,-ticknumb,-gender) -> festival.data.stack  
  
festival.data.stack %>% filter(!is.na(score)) -> festival.data.stack  
festival.data.stack

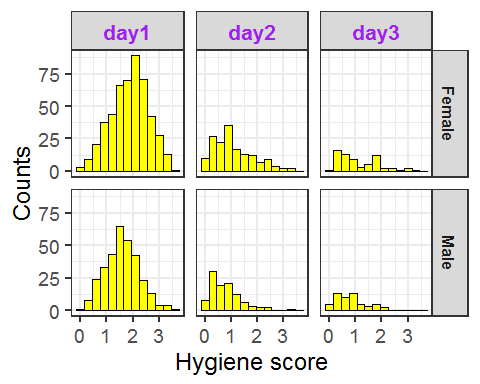
## # A tibble: 1,197 x 4  
## ticknumb gender day score  
## <dbl> <chr> <chr> <dbl>  
## 1 2111 Male day1 2.64  
## 2 2229 Female day1 0.97  
## 3 2338 Male day1 0.84  
## 4 2384 Female day1 3.03  
## 5 2401 Female day1 0.88  
## 6 2405 Male day1 0.85  
## 7 2467 Female day1 1.56  
## 8 2478 Female day1 3.02  
## 9 2490 Male day1 2.29  
## 10 2504 Female day1 1.11  
## # ... with 1,187 more rows

ggplot(festival.data.stack,aes(score))+  
 geom\_histogram(binwidth=0.3, color="black", fill="yellow")+  
 labs(x="Hygiene score", y="Counts")+  
 facet\_grid(gender~day) -> histogram.3days  
histogram.3days



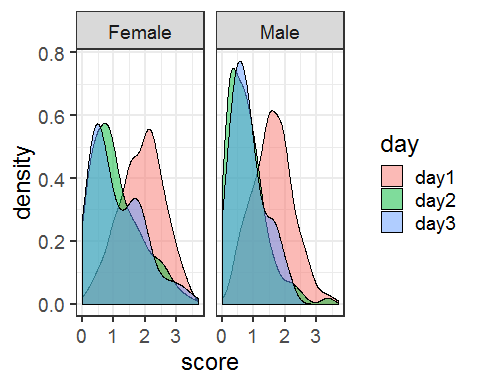
It is possible to modify the labels of the facets. here are some examples below.

histogram.3days<-ggplot(festival.data.stack,aes(score))+  
 geom\_histogram(binwidth=0.3, color="black", fill="yellow")+  
 labs(x="Hygiene score", y="Counts")+  
 facet\_grid(gender~day)+  
 theme(strip.text.x = element\_text(size = 16, colour = "purple", face="bold"),  
 strip.text.y = element\_text(size=12, face="bold"))  
histogram.3days



Density plots as below.

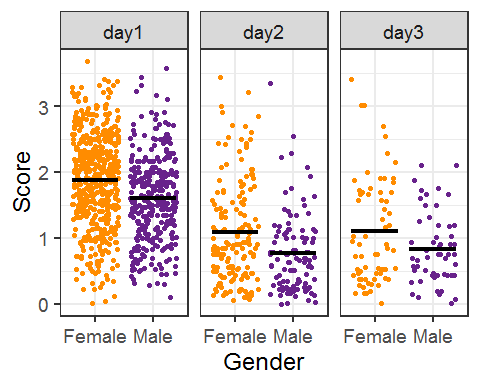
density.3days<-ggplot(festival.data.stack, aes(score))+  
 geom\_density(aes(group=day, fill=day), alpha=0.5)+  
 facet\_grid(~gender)  
density.3days



## Exercise 2:

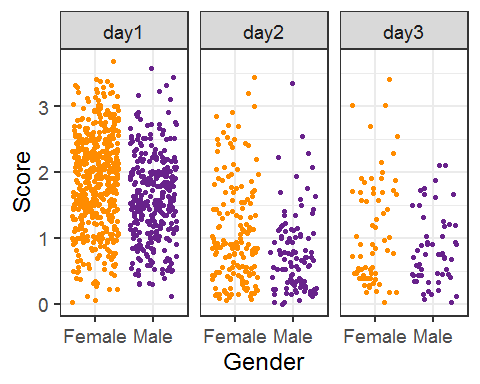
Plot a stripchart representing all 3 days and each gender

stripchart <-ggplot(festival.data.stack, aes(gender, score, colour=gender))+   
 facet\_grid(~day)+  
 geom\_point(position="jitter")+  
 scale\_colour\_manual(values=c("darkorange", "darkorchid4"))+   
 stat\_summary(geom='errorbar',fun.y=mean, aes(ymin=..y.., ymax=..y..),   
 colour="black", width=0.8, size=1.5)+  
 labs(x="Gender", y="Score")+  
 theme(legend.position = "none")  
stripchart



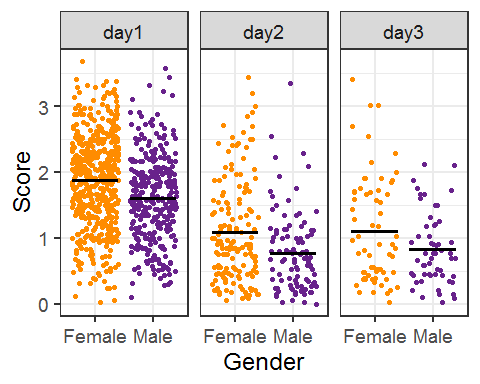
From a stripchart, we can add a line for the mean or any other descriptive geom as a stat summary.

stripchart<-ggplot(festival.data.stack, aes(gender, score,colour=gender))+facet\_grid(~day)+  
 geom\_point(position="jitter")+  
 scale\_colour\_manual(values=c("darkorange", "darkorchid4"))+  
 labs(x="Gender", y="Score")+  
 theme(legend.position = "none")  
   
stripchart



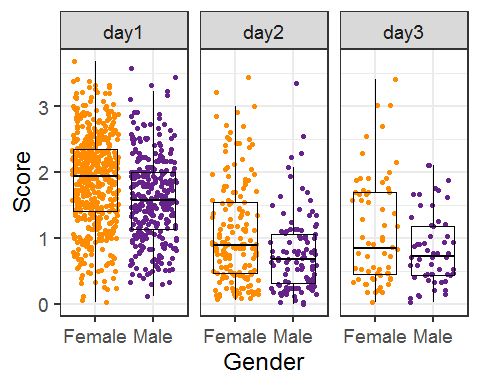
We saw how to add a mean:

stripchart+  
 stat\_summary(fun.y="mean",geom="errorbar", aes(ymin=..y.., ymax=..y..), width=0.8, colour="black", size = 1.3)



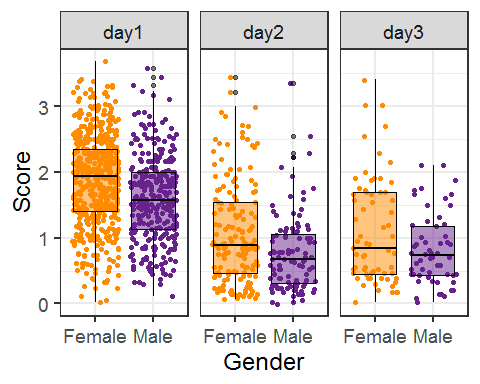
Now let’s add a boxplot.

stripchart+  
 geom\_boxplot(alpha=0, colour="black")



We can make it prettier.

stripchart+  
 geom\_boxplot(aes(gender, score, fill=gender), alpha=0.5, colour="black")+  
 scale\_fill\_manual(values=c("darkorange", "darkorchid4"))



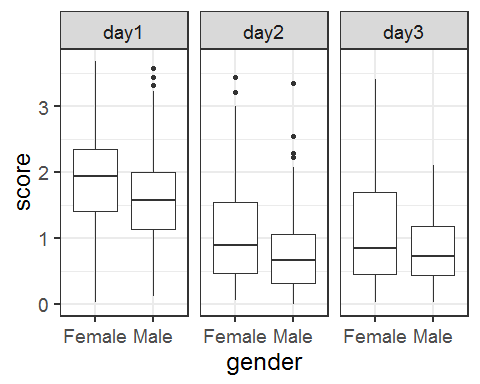
Speaking of making graphs prettier, we can improve on the boxplot.

# Boxplot

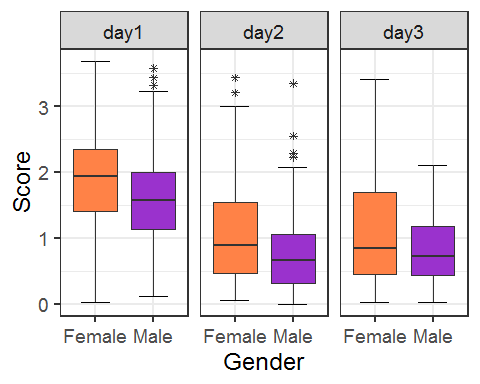
We can change order on the x-axis if we want:

boxplot+scale\_x\_discrete(limits=c(“Male”,“Female”))

boxplot<-ggplot(festival.data.stack, aes(gender,score))+  
 geom\_boxplot()+  
 facet\_grid(~day)  
  
boxplot

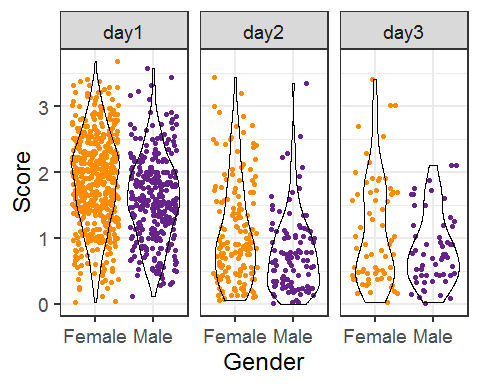


boxplot <-ggplot(festival.data.stack, aes(gender,score, fill=gender))+  
 facet\_grid(~day)+  
 stat\_boxplot(geom="errorbar", width=0.5)+   
 geom\_boxplot(outlier.shape=8)+  
 theme(legend.position = "none")+  
 scale\_fill\_manual(values=c("sienna1","darkorchid3 "))+  
 labs(x="Gender", y="Score")  
boxplot



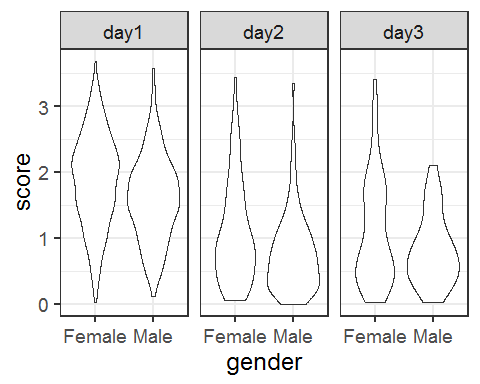
# Violinplot (beanplot)

stripchart+  
 geom\_violin(alpha=0, colour="black")



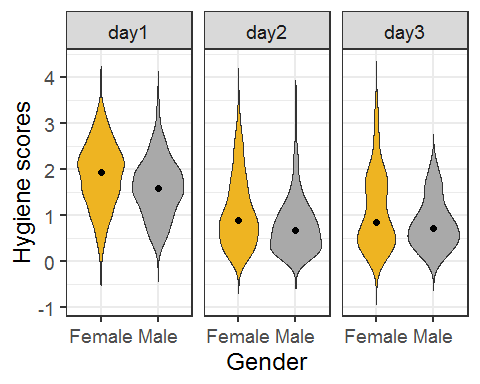
Basic command

violinplot<-ggplot(festival.data.stack, aes(gender,score))+geom\_violin()+facet\_grid(~day)  
violinplot

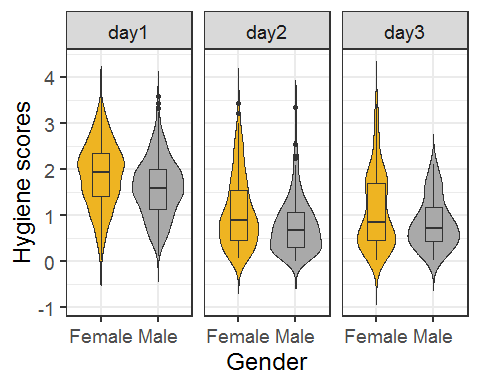


Prettier:

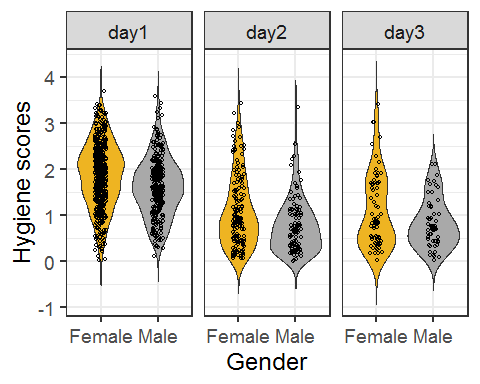
violinplot<-ggplot(festival.data.stack, aes(gender,score,fill=gender))+  
 facet\_grid(~day)+  
 geom\_violin(trim = FALSE)+  
 scale\_fill\_manual(values=c("goldenrod2","darkgrey"))+  
 theme(legend.position="none")+  
 stat\_summary(fun.y=median, geom="point", size=2, color="black")+  
 labs(x="Gender", y="Hygiene scores")  
violinplot



violinplot+geom\_boxplot(width=0.3)



violinplot+geom\_jitter(width=0.1,size=1, shape=1)



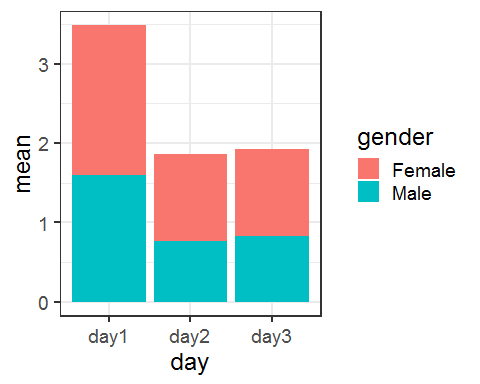
# Barchart

First we want to calculate the means and sem and store the values in a file.

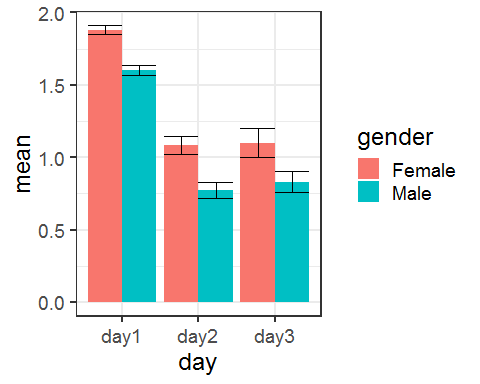
festival.data.stack %>%  
 group\_by(gender,day) %>%  
 summarise(mean=mean(score), sem=sd(score)/sqrt(n())) -> score.sem  
  
score.sem

## # A tibble: 6 x 4  
## # Groups: gender [2]  
## gender day mean sem  
## <chr> <chr> <dbl> <dbl>  
## 1 Female day1 1.88 0.0316  
## 2 Female day2 1.08 0.0608  
## 3 Female day3 1.10 0.0990  
## 4 Male day1 1.60 0.0362  
## 5 Male day2 0.773 0.0585  
## 6 Male day3 0.829 0.0721

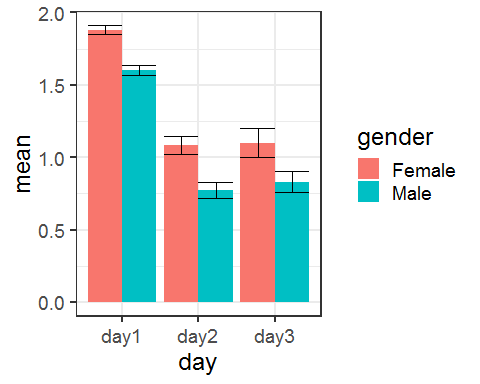
barchart<-ggplot(score.sem, aes(day,mean, fill=gender))+  
 geom\_bar(stat="identity")  
barchart



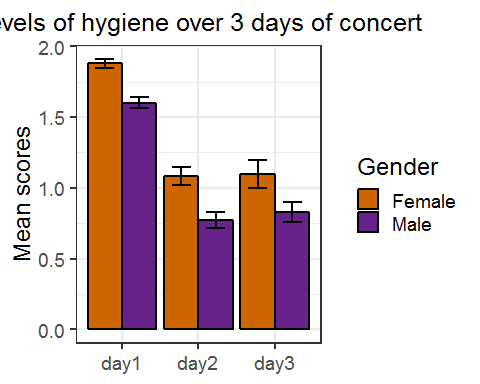
barchart<-ggplot(score.sem, aes(day,mean, fill=gender))+  
 geom\_bar(stat="identity", position="dodge")+  
 geom\_errorbar(aes(ymin=mean-sem, ymax=mean+sem), position="dodge")  
barchart



barchart<-ggplot(score.sem, aes(day,mean, fill=gender))+  
 geom\_bar(position="dodge", stat="identity")+  
 geom\_errorbar(aes(ymin=mean-sem, ymax=mean+sem), position="dodge")  
barchart



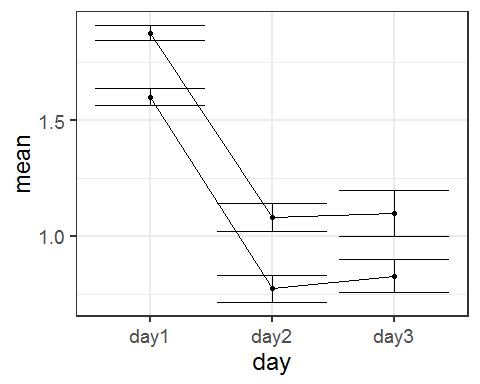
barchart<-ggplot(score.sem, aes(day,mean, fill=gender))+  
 geom\_bar(position="dodge", colour="black",stat="identity",size=1)+  
 geom\_errorbar(aes(ymin=mean-sem, ymax=mean+sem), width=.5, position=position\_dodge(0.9), size=1)+  
 ylab("Mean scores")+   
 ggtitle("Levels of hygiene over 3 days of concert")+  
 theme(plot.title = element\_text(hjust = 0.5))+  
 theme(plot.title = element\_text(size = 19))+  
 theme(axis.title.x=element\_blank())+  
 scale\_fill\_manual(values=c("darkorange3", "darkorchid4"), name="Gender")  
barchart



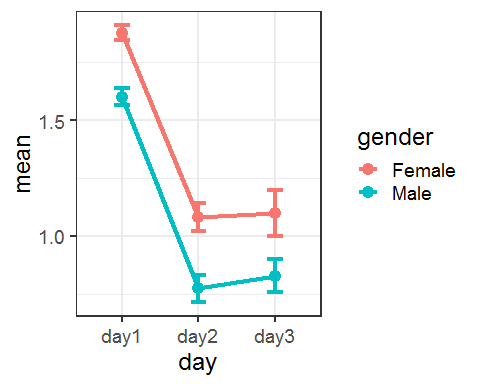
So beautiful!

# Linegraph

linegraph<-ggplot(score.sem, aes(day, mean, group=gender))+  
 geom\_line()+  
 geom\_point()+  
 geom\_errorbar(aes(ymin=mean-sem, ymax=mean+sem))  
  
linegraph

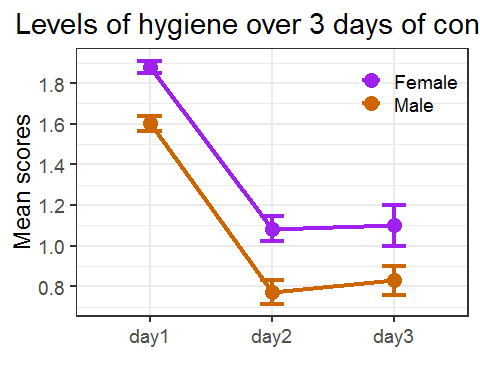


linegraph<-ggplot(score.sem, aes(day,mean, colour=gender, group=gender))+  
 geom\_line(size=1.5)+  
 geom\_point(size=4)+  
 geom\_errorbar(aes(ymin=mean-sem, ymax=mean+sem), width=.2, size=1.5)  
linegraph



Change the position of the legend.

linegraph<-ggplot(score.sem, aes(day,mean, colour=gender, group=gender))+  
 geom\_line(size=1.5)+  
 geom\_point(size=5)+  
 geom\_errorbar(aes(ymin=mean-sem, ymax=mean+sem), width=.2, size=1.5)+  
 labs(x="", y="Mean scores")+  
 scale\_y\_continuous(breaks=seq(0, 2, 0.2))+  
 ggtitle("Levels of hygiene over 3 days of concert")+  
 theme(plot.title = element\_text(hjust = 0.5))+  
 scale\_colour\_manual(values=c("purple","darkorange3"), name="")+  
 theme(legend.position = c(0.85, 0.9))+  
 theme(legend.text=element\_text(size=14))+  
 theme(legend.background = element\_rect(fill = "transparent"))  
  
linegraph



## Exercise 3:

The file contains positional count data for 3 different datasets (a WT and two mutants).

Plot a graph showing all 3 datasets on the same plot

Load: “chromosome\_position\_data.txt” Check the structure of the file Restructure the file from wide to long format gather() Rename the column: “Genotype” and “Value” Plot a basic line graph

chromosome<-read\_tsv("chromosome\_position\_data.txt")

## Parsed with column specification:  
## cols(  
## Position = col\_double(),  
## Mut1 = col\_double(),  
## Mut2 = col\_double(),  
## WT = col\_double()  
## )

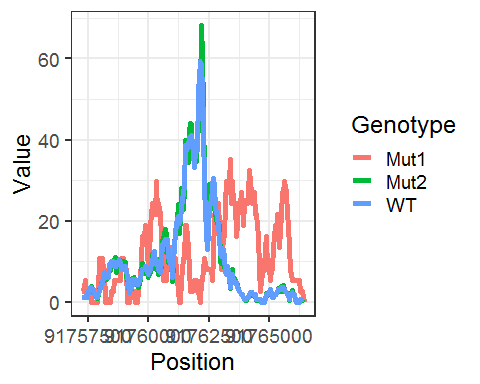
chromosome

## # A tibble: 184 x 4  
## Position Mut1 Mut2 WT  
## <dbl> <dbl> <dbl> <dbl>  
## 1 91757273 2.71 1.34 1.25  
## 2 91757323 2.71 1.3 1.25  
## 3 91757373 5.41 1.14 1.25  
## 4 91757423 2.71 1.58 1.88  
## 5 91757473 2.71 1.19 1.25  
## 6 91757523 2.71 2.82 2.5   
## 7 91757573 2.71 3.15 3.75  
## 8 91757623 0 4.05 3.75  
## 9 91757673 0 2.94 3.12  
## 10 91757723 0 2.69 3.12  
## # ... with 174 more rows

chromosome %>%  
 gather(Genotype, Value,-Position) -> chromosome.long  
chromosome.long

## # A tibble: 552 x 3  
## Position Genotype Value  
## <dbl> <chr> <dbl>  
## 1 91757273 Mut1 2.71  
## 2 91757323 Mut1 2.71  
## 3 91757373 Mut1 5.41  
## 4 91757423 Mut1 2.71  
## 5 91757473 Mut1 2.71  
## 6 91757523 Mut1 2.71  
## 7 91757573 Mut1 2.71  
## 8 91757623 Mut1 0   
## 9 91757673 Mut1 0   
## 10 91757723 Mut1 0   
## # ... with 542 more rows

chromosome.linegraph<-ggplot(chromosome.long, aes(x=Position, y=Value, group=Genotype, colour=Genotype))+  
geom\_line(size=2)  
chromosome.linegraph



## Exercise 4:

Plot a graph showing the relationship between age and weight for a typical baby over the first 9 months.

Load: weight\_chart.txt Check the structure of the file Plot a basic line graph Plot a prettier version: Change the size and the colour of the points Change the thickness and the colour of the line Change the y-axis: scale from 2 to 10 kgs Change the x-axis: scale from 0 t 10 months Change the labels on both the axis Add a title to the graph

weight<-read\_tsv("weight\_chart.txt")

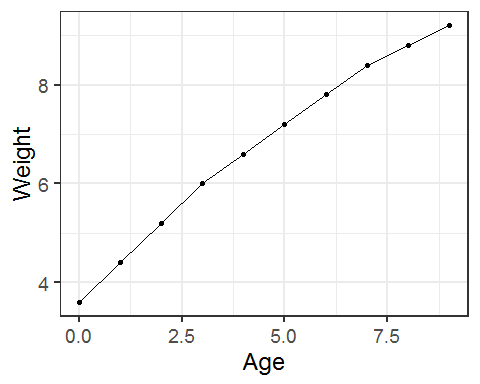
## Parsed with column specification:  
## cols(  
## Age = col\_double(),  
## Weight = col\_double()  
## )

weight

## # A tibble: 10 x 2  
## Age Weight  
## <dbl> <dbl>  
## 1 0 3.6  
## 2 1 4.4  
## 3 2 5.2  
## 4 3 6   
## 5 4 6.6  
## 6 5 7.2  
## 7 6 7.8  
## 8 7 8.4  
## 9 8 8.8  
## 10 9 9.2

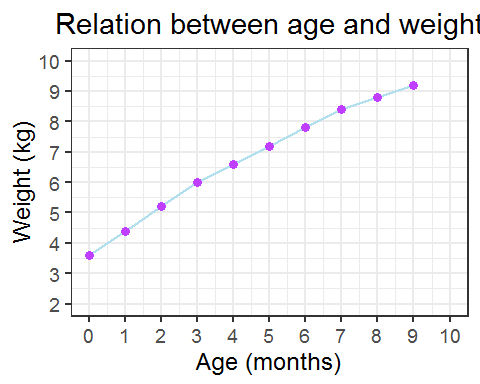
Basic graph:

weight.linegraph<-ggplot(weight, aes(Age, Weight))+  
 geom\_line()+  
 geom\_point()  
weight.linegraph



Pretty graph:

weight.linegraph<-ggplot(weight, aes(Age, Weight))+  
 geom\_line(size=1, colour="lightblue2")+  
 geom\_point(shape=16, size=3, colour="darkorchid1")+  
 scale\_y\_continuous(breaks=2:10, limits = c(2, 10))+  
 scale\_x\_continuous(breaks=0:10, limits = c(0, 10))+  
 labs(x="Age (months)", y="Weight (kg)")+  
 ggtitle("Relation between age and weight")+  
 theme(plot.title = element\_text(hjust = 0.5))  
weight.linegraph



## Exercise 5:

The file brain\_bodyweight.txt contains data for the log10 brain and bodyweight for a range of species, along with an SEM measure for each point.

Plot these data on a scatterplot with error bars showing the mean +/- SEM and the names of the datasets under each point.

Load: brain\_bodyweight.txt Check the structure of the file Plot a basic graph. You will need: geom\_barh() for the horizontal error bars geom\_text() for the labels

Plot a prettier version.

brain.bodyweight<- read\_tsv("brain\_bodyweight.txt")

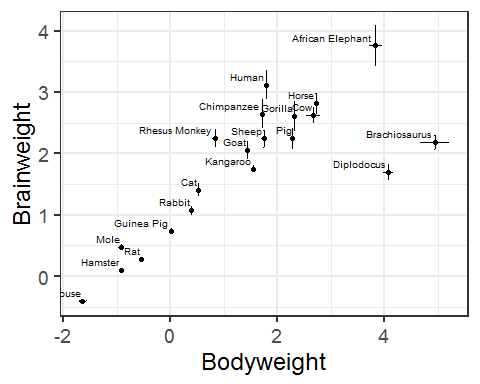
## Parsed with column specification:  
## cols(  
## Species = col\_character(),  
## Bodyweight = col\_double(),  
## Brainweight = col\_double(),  
## Bodyweight.SEM = col\_double(),  
## Brainweight.SEM = col\_double()  
## )

brain.bodyweight

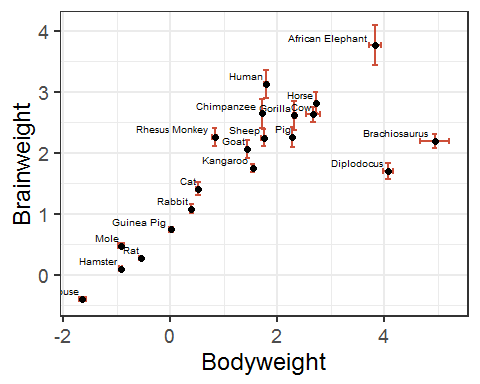
## # A tibble: 20 x 5  
## Species Bodyweight Brainweight Bodyweight.SEM Brainweight.SEM  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 Cow 2.67 2.63 0.125 0.127   
## 2 Goat 1.44 2.06 0.0193 0.142   
## 3 Guinea Pig 0.0170 0.740 0.00163 0.0359   
## 4 Diplodocus 4.07 1.70 0.101 0.134   
## 5 Horse 2.72 2.82 0.00367 0.169   
## 6 Cat 0.519 1.41 0.0292 0.102   
## 7 Gorilla 2.32 2.61 0.0268 0.243   
## 8 Human 1.79 3.12 0.0361 0.231   
## 9 African Elephant 3.82 3.76 0.112 0.329   
## 10 Rhesus Monkey 0.833 2.25 0.0465 0.146   
## 11 Kangaroo 1.54 1.75 0.0175 0.0587   
## 12 Hamster -0.921 0.1 0.0274 0.00981  
## 13 Mouse -1.64 -0.398 0.0686 0.0234   
## 14 Rabbit 0.398 1.08 0.0115 0.0669   
## 15 Sheep 1.74 2.24 0.0410 0.136   
## 16 Chimpanzee 1.72 2.64 0.0164 0.238   
## 17 Brachiosaurus 4.94 2.19 0.269 0.112   
## 18 Rat -0.553 0.279 0.0337 0.00188  
## 19 Mole -0.914 0.477 0.0617 0.0477   
## 20 Pig 2.28 2.26 0.0133 0.160

Basic

brain.bodyweight.graph<-ggplot(brain.bodyweight, aes(x=Bodyweight, y=Brainweight))+  
 geom\_point()+  
 geom\_errorbar(aes(ymin=Brainweight-Brainweight.SEM, ymax=Brainweight+Brainweight.SEM))+  
 geom\_errorbarh(aes(xmin=Bodyweight-Bodyweight.SEM, xmax=Bodyweight+Bodyweight.SEM))+  
 geom\_text(aes(label=Species), hjust = 1.05, vjust = -0.6, size=2.7)  
brain.bodyweight.graph



brain.bodyweight.graph<-ggplot(brain.bodyweight, aes(x=Bodyweight, y=Brainweight))+  
 geom\_point()+  
 geom\_errorbar(aes(ymin=Brainweight-Brainweight.SEM, ymax=Brainweight+Brainweight.SEM), width=.1, size=1, colour="tomato3")+  
 geom\_errorbarh(aes(xmin=Bodyweight-Bodyweight.SEM, xmax=Bodyweight+Bodyweight.SEM), height=.1, size=1, colour="tomato3")+  
 geom\_point(size=2)+  
 geom\_text(aes(label=Species), hjust = 1.1, vjust = -0.6, size=2.7)  
brain.bodyweight.graph

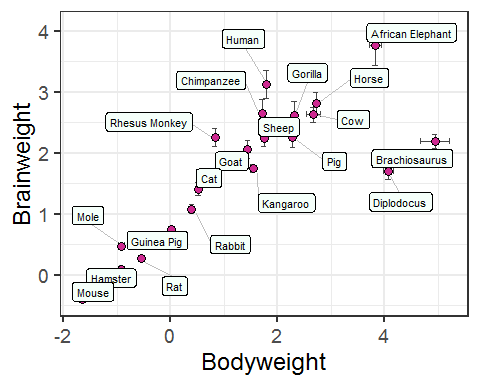


Prettier with ggrepel

library("ggrepel")

## Warning: package 'ggrepel' was built under R version 3.5.3

ggplot(brain.bodyweight, aes(x=Bodyweight, y=Brainweight))+  
 geom\_errorbar(aes(ymin=Brainweight-Brainweight.SEM, ymax=Brainweight+Brainweight.SEM),   
width=.1, size=0.5, colour="grey28")+  
 geom\_errorbarh(aes(xmin=Bodyweight-Bodyweight.SEM, xmax=Bodyweight+Bodyweight.SEM),   
height=.1, size=0.5, colour="grey28")+  
 geom\_point(shape=21, size=3, colour="black", fill="maroon3")+  
 geom\_label\_repel(aes(label = Species), box.padding=0.6, point.padding =0.5,   
fill="mintcream", segment.colour="grey", size=3) -> brain.bodyweight.graph  
brain.bodyweight.graph



# Stacked bar: categorical data

Changing<-read\_csv("Changing.csv")

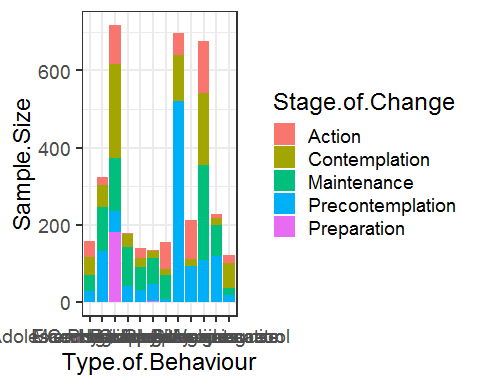
## Parsed with column specification:  
## cols(  
## Type.of.Behaviour = col\_character(),  
## Sample.Size = col\_double(),  
## Stage.of.Change = col\_character()  
## )

Changing

## # A tibble: 60 x 3  
## Type.of.Behaviour Sample.Size Stage.of.Change   
## <chr> <dbl> <chr>   
## 1 Smoking cessation 108 Precontemplation  
## 2 Smoking cessation 187 Contemplation   
## 3 Smoking cessation 0 Preparation   
## 4 Smoking cessation 134 Action   
## 5 Smoking cessation 247 Maintenance   
## 6 Quitting cocaine 8 Precontemplation  
## 7 Quitting cocaine 15 Contemplation   
## 8 Quitting cocaine 0 Preparation   
## 9 Quitting cocaine 71 Action   
## 10 Quitting cocaine 62 Maintenance   
## # ... with 50 more rows

Basic

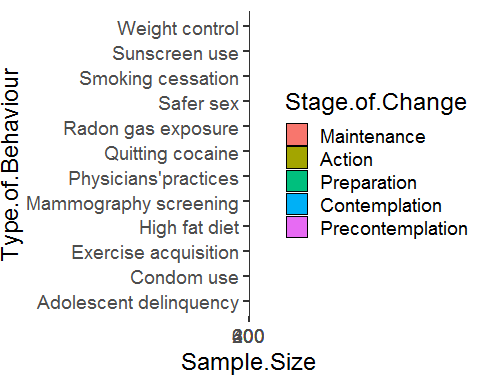
stackedBar<-ggplot(Changing, aes(Type.of.Behaviour, Sample.Size, fill=Stage.of.Change))+  
geom\_bar(stat="identity")  
stackedBar



Changing the order of the levels: factor(variable name, levels = c(“”, “” .))

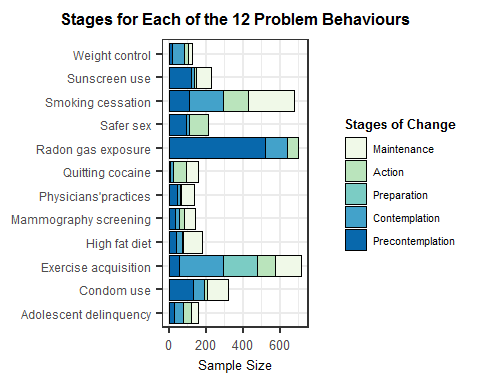
Rotate the graph to read the x-axis labels: coord\_flip()

Changing$Stage.of.Change <- factor(Changing$Stage.of.Change, levels = c("Maintenance","Action","Preparation","Contemplation","Precontemplation"))  
  
stackedBar<-ggplot(Changing, aes(Type.of.Behaviour, Sample.Size, fill = Stage.of.Change))+  
 geom\_bar(stat="identity", colour="black")+  
 coord\_flip()  
stackedBar



Prettier:

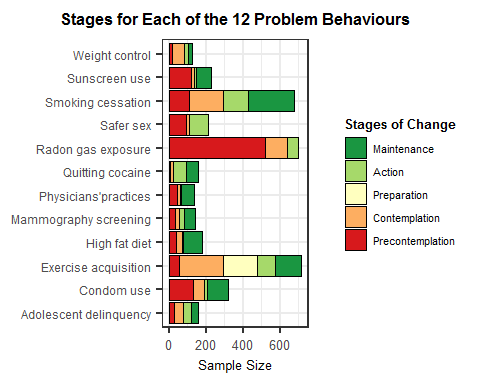
stackedBar<-stackedBar+  
 labs(title="Stages for Each of the 12 Problem Behaviours", y="Sample Size", fill="Stages of Change")+  
 theme(plot.title = element\_text(hjust = 0.5, size=12, face="bold"))+  
 theme(axis.title.y=element\_blank())+  
 scale\_fill\_brewer(palette = 4)+  
 theme(axis.text.x = element\_text(size=10), axis.text.y = element\_text(size=9))+  
 theme(legend.text=element\_text(size=8), legend.title=element\_text(size=10, face="bold"))+  
 theme(axis.title.x = element\_text(size=10))  
stackedBar



With a divergent palette

stackedBar+scale\_fill\_brewer(palette="RdYlGn", direction=-1)

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



## Exercise 6

Let’s plot the same data but as percentages

Plot the Changing data as percentages.

Change the format of the file into contingency xtabs() Calculate the percentages prop.table() Change the format into a dataframe as.data.frame() Check your file head() Plot the data as before using a suitable divergent palette

contingency.table100<-prop.table(xtabs(Sample.Size~Type.of.Behaviour+Stage.of.Change, Changing),1)\*100  
contingency.table100

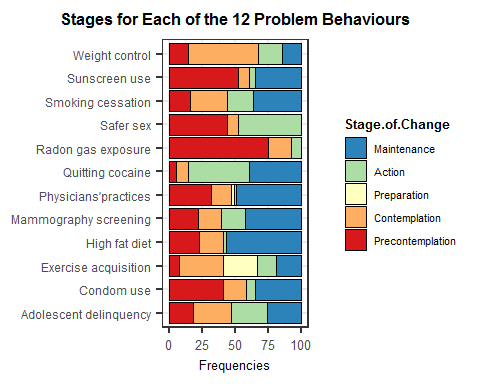
## Stage.of.Change  
## Type.of.Behaviour Maintenance Action Preparation Contemplation  
## Adolescent delinquency 25.786164 27.044025 0.000000 28.930818  
## Condom use 35.294118 6.191950 0.000000 17.956656  
## Exercise acquisition 19.386332 14.086471 25.383543 33.751743  
## High fat diet 56.666667 2.777778 0.000000 17.777778  
## Mammography screening 42.553191 18.439716 0.000000 17.021277  
## Physicians'practices 49.629630 1.481481 2.222222 14.814815  
## Quitting cocaine 39.743590 45.512821 0.000000 9.615385  
## Radon gas exposure 0.000000 8.166189 0.000000 17.335244  
## Safer sex 0.000000 47.887324 0.000000 7.981221  
## Smoking cessation 36.538462 19.822485 0.000000 27.662722  
## Sunscreen use 35.242291 4.405286 0.000000 7.929515  
## Weight control 14.634146 17.886179 0.000000 52.845528  
## Stage.of.Change  
## Type.of.Behaviour Precontemplation  
## Adolescent delinquency 18.238994  
## Condom use 40.557276  
## Exercise acquisition 7.391911  
## High fat diet 22.777778  
## Mammography screening 21.985816  
## Physicians'practices 31.851852  
## Quitting cocaine 5.128205  
## Radon gas exposure 74.498567  
## Safer sex 44.131455  
## Smoking cessation 15.976331  
## Sunscreen use 52.422907  
## Weight control 14.634146

Changing.percent<-as.data.frame(contingency.table100)  
Changing.percent

## Type.of.Behaviour Stage.of.Change Freq  
## 1 Adolescent delinquency Maintenance 25.786164  
## 2 Condom use Maintenance 35.294118  
## 3 Exercise acquisition Maintenance 19.386332  
## 4 High fat diet Maintenance 56.666667  
## 5 Mammography screening Maintenance 42.553191  
## 6 Physicians'practices Maintenance 49.629630  
## 7 Quitting cocaine Maintenance 39.743590  
## 8 Radon gas exposure Maintenance 0.000000  
## 9 Safer sex Maintenance 0.000000  
## 10 Smoking cessation Maintenance 36.538462  
## 11 Sunscreen use Maintenance 35.242291  
## 12 Weight control Maintenance 14.634146  
## 13 Adolescent delinquency Action 27.044025  
## 14 Condom use Action 6.191950  
## 15 Exercise acquisition Action 14.086471  
## 16 High fat diet Action 2.777778  
## 17 Mammography screening Action 18.439716  
## 18 Physicians'practices Action 1.481481  
## 19 Quitting cocaine Action 45.512821  
## 20 Radon gas exposure Action 8.166189  
## 21 Safer sex Action 47.887324  
## 22 Smoking cessation Action 19.822485  
## 23 Sunscreen use Action 4.405286  
## 24 Weight control Action 17.886179  
## 25 Adolescent delinquency Preparation 0.000000  
## 26 Condom use Preparation 0.000000  
## 27 Exercise acquisition Preparation 25.383543  
## 28 High fat diet Preparation 0.000000  
## 29 Mammography screening Preparation 0.000000  
## 30 Physicians'practices Preparation 2.222222  
## 31 Quitting cocaine Preparation 0.000000  
## 32 Radon gas exposure Preparation 0.000000  
## 33 Safer sex Preparation 0.000000  
## 34 Smoking cessation Preparation 0.000000  
## 35 Sunscreen use Preparation 0.000000  
## 36 Weight control Preparation 0.000000  
## 37 Adolescent delinquency Contemplation 28.930818  
## 38 Condom use Contemplation 17.956656  
## 39 Exercise acquisition Contemplation 33.751743  
## 40 High fat diet Contemplation 17.777778  
## 41 Mammography screening Contemplation 17.021277  
## 42 Physicians'practices Contemplation 14.814815  
## 43 Quitting cocaine Contemplation 9.615385  
## 44 Radon gas exposure Contemplation 17.335244  
## 45 Safer sex Contemplation 7.981221  
## 46 Smoking cessation Contemplation 27.662722  
## 47 Sunscreen use Contemplation 7.929515  
## 48 Weight control Contemplation 52.845528  
## 49 Adolescent delinquency Precontemplation 18.238994  
## 50 Condom use Precontemplation 40.557276  
## 51 Exercise acquisition Precontemplation 7.391911  
## 52 High fat diet Precontemplation 22.777778  
## 53 Mammography screening Precontemplation 21.985816  
## 54 Physicians'practices Precontemplation 31.851852  
## 55 Quitting cocaine Precontemplation 5.128205  
## 56 Radon gas exposure Precontemplation 74.498567  
## 57 Safer sex Precontemplation 44.131455  
## 58 Smoking cessation Precontemplation 15.976331  
## 59 Sunscreen use Precontemplation 52.422907  
## 60 Weight control Precontemplation 14.634146

Plot the data as percentages.

stackedBar.percent<-ggplot(Changing.percent,aes(Type.of.Behaviour, Freq, fill = Stage.of.Change))+  
 geom\_bar(stat="identity",colour="black")+  
 coord\_flip()+  
 scale\_fill\_brewer(palette = "Spectral", direction=-1)+  
 labs(title="Stages for Each of the 12 Problem Behaviours", y="Frequencies")+  
 theme(axis.title.y=element\_blank())+  
 theme(plot.title = element\_text(hjust = 0.5, size=12, face="bold"))+  
 theme(axis.text.x = element\_text(size=10), axis.text.y = element\_text(size=9))+  
 theme(legend.text=element\_text(size=8), legend.title=element\_text(size=10, face="bold"))+  
 theme(axis.title.x = element\_text(size=10))  
stackedBar.percent



### To save a graph

Linegraph.saved<-ggsave(Linegraph, file=“Line.png”)