Data Science in Medicine: Lab 2

Hypothesis testing in R

```
# Author: Areti Manataki
# Last updated: 6th October 2020
# Description: This file is used as part of Lab 2 in the Data Science in Medici
ne course.
# Additional files needed: i) parenthood.Rdata, ii) chapek9.Rdata, iii) harpo.R
data
# iv) parenthood2.Rdata, v) DataScienceClass.csv, vi) DataScienceClass2.csv

# Instructions for students:
# To run a command, place your cursor on any part of it and click Ctrl+Enter (o
r Commd+Enter)
# To write a comment, include "#" at the beginning of the corresponding line.
```

Part 1: Hypothesis testing

Correlation between numerical variables

```
# import data (Rdata format)
load("parenthood.Rdata")

# explore the data
str(parenthood)
```

```
## 'data.frame':    100 obs. of  4 variables:
##  $ dan.sleep : num  7.59 7.91 5.14 7.71 6.68 5.99 8.19 7.19 7.4 6.58 ...
##  $ baby.sleep: num  10.18 11.66 7.92 9.61 9.75 ...
##  $ dan.grump : num  56 60 82 55 67 72 53 60 60 71 ...
##  $ day       : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
head(parenthood)
```

	dan.sleep	baby.sleep	dan.grump	day
	<dbl>	<dbl>	<dbl>	<int>
1	7.59	10.18	56	1
2	7.91	11.66	60	2
3	5.14	7.92	82	3
4	7.71	9.61	55	4

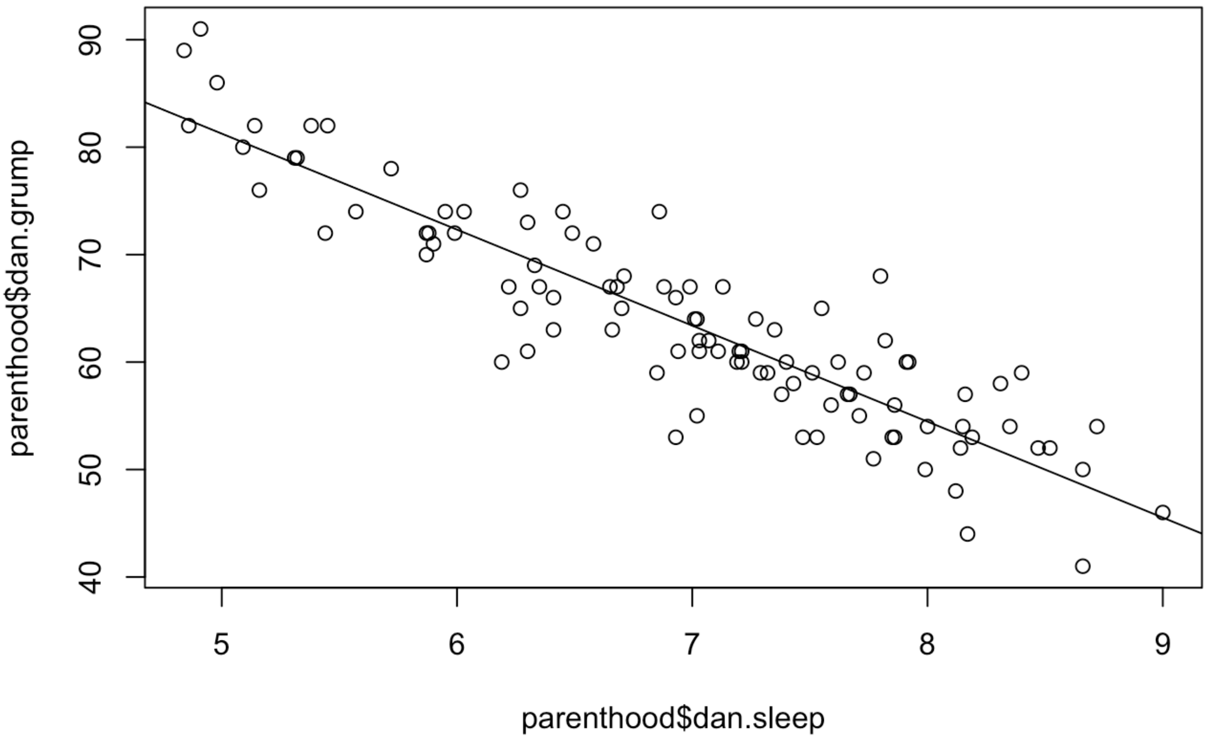
	dan.sleep	baby.sleep	dan.grump	day
	<dbl>	<dbl>	<dbl>	<int>
5	6.68	9.75	67	5
6	5.99	5.04	72	6

6 rows

```
summary(parenthood)
```

```
##    dan.sleep       baby.sleep       dan.grump          day        
##  Min.   :4.840   Min.   : 3.250   Min.   :41.00   Min.   :  1.00  
##  1st Qu.:6.293   1st Qu.: 6.425   1st Qu.:57.00   1st Qu.: 25.75  
##  Median :7.030   Median : 7.950   Median :62.00   Median : 50.50  
##  Mean   :6.965   Mean   : 8.049   Mean   :63.71   Mean   : 50.50  
##  3rd Qu.:7.740   3rd Qu.: 9.635   3rd Qu.:71.00   3rd Qu.: 75.25  
##  Max.   :9.000   Max.   :12.070   Max.   :91.00   Max.   :100.00
```

```
# visualise in scatterplot (labeling should be better!)
plot(parenthood$dan.sleep, parenthood$dan.grump)
abline(lm(parenthood$dan.grump ~ parenthood$dan.sleep))
```



```
# get correlation coefficient for two variables
cor(parenthood$dan.sleep, parenthood$dan.grump)
```

```
## [1] -0.903384
```

```
# carry out correlation testing
cor.test(parenthood$dan.sleep, parenthood$dan.grump)
```

```
##
##  Pearson's product-moment correlation
##
## data:  parenthood$dan.sleep and parenthood$dan.grump
## t = -20.854, df = 98, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.9340614 -0.8594714
## sample estimates:
##       cor
## -0.903384
```

Association between categorical variables

```
# import data (Rdata format)
load("chapek9.Rdata")

# explore the data (and check whether categorical)
str(chapek9)
```

```
## 'data.frame':    180 obs. of  2 variables:
##  $ species: Factor w/ 2 levels "robot","human": 1 2 2 2 1 2 2 1 2 1 ...
##  $ choice : Factor w/ 3 levels "puppy","flower",..: 2 3 3 3 3 2 3 3 1 2 ...
```

```
summary(chapek9)
```

```
##   species      choice
##  robot:87   puppy : 28
##  human:93   flower: 43
##             data  :109
```

```
head(chapek9)
```

	species <fctr>	choice <fctr>
1	robot	flower
2	human	data
3	human	data

	species <fctr>	**choice** <fctr>
4	human	data
5	robot	data
6	human	flower

6 rows

```
# get observed frequencies
chapekFrequencies <- table(chapek9$choice, chapek9$species)
chapekFrequencies
```

```
##
##          robot human
##   puppy     13    15
##   flower    30    13
##   data      44    65
```

```
# get expected frequencies
chisq.test(chapekFrequencies)$expected
```

```
##
##             robot    human
##   puppy  13.53333 14.46667
##   flower 20.78333 22.21667
##   data   52.68333 56.31667
```

```
# carry out chi-square testing
chisq.test(chapekFrequencies)
```

```
##
##  Pearson's Chi-squared test
##
## data:  chapekFrequencies
## X-squared = 10.722, df = 2, p-value = 0.004697
```

Comparing the means of two independently drawn samples

```
# import data (Rdata format)
load("harpo.Rdata")

# explore the data
str(harpo)
```

```
## 'data.frame':    33 obs. of  2 variables:
##  $ grade: num  65 72 66 74 73 71 66 76 69 79 ...
##  $ tutor: Factor w/ 2 levels "Anastasia","Bernadette": 1 2 2 1 1 2 2 2 2 2
...
```

```
head(harpo)
```

	grade	tutor
	<dbl>	<fctr>
1	65	Anastasia
2	72	Bernadette
3	66	Bernadette
4	74	Anastasia
5	73	Anastasia
6	71	Bernadette

6 rows

```
summary(harpo)
```

```
##      grade              tutor   
##  Min.   :55.00   Anastasia :15  
##  1st Qu.:67.00   Bernadette:18  
##  Median :72.00                  
##  Mean   :71.55                  
##  3rd Qu.:76.00                  
##  Max.   :90.00                  
```

```
# carry out independent samples t-testing
t.test(formula = grade ~ tutor, data = harpo, var.equal = TRUE)
```

```
## 
##  Two Sample t-test
## 
## data:  grade by tutor
## t = 2.1154, df = 31, p-value = 0.04253
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   0.1965873 10.7589683
## sample estimates:
##  mean in group Anastasia mean in group Bernadette 
##                 74.53333                 69.05556 
```

Part 2: Missing values

```
# import data (Rdata format)
load("parenthood2.Rdata")

# explore the data (missing values detected!)
str(parenthood2)
```

```
## 'data.frame':    100 obs. of  4 variables:
##  $ dan.sleep : num  7.59 7.91 5.14 7.71 6.68 5.99 8.19 7.19 7.4 6.58 ...
##  $ baby.sleep: num  NA 11.66 7.92 9.61 9.75 ...
##  $ dan.grump : num  56 60 82 55 NA 72 53 60 NA 71 ...
##  $ day       : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
head(parenthood2)
```

	dan.sleep <dbl>	**baby.sleep** <dbl>	**dan.grump** <dbl>	**day** <int>
1	7.59	NA	56	1
2	7.91	11.66	60	2
3	5.14	7.92	82	3
4	7.71	9.61	55	4
5	6.68	9.75	NA	5
6	5.99	5.04	72	6

6 rows

```
summary(parenthood2)
```

```
##    dan.sleep       baby.sleep       dan.grump          day        
##  Min.   :4.840   Min.   : 3.250   Min.   :41.00   Min.   :  1.00  
##  1st Qu.:6.285   1st Qu.: 6.460   1st Qu.:56.00   1st Qu.: 25.75  
##  Median :7.030   Median : 8.200   Median :61.00   Median : 50.50  
##  Mean   :6.977   Mean   : 8.114   Mean   :63.15   Mean   : 50.50  
##  3rd Qu.:7.785   3rd Qu.: 9.610   3rd Qu.:70.25   3rd Qu.: 75.25  
##  Max.   :9.000   Max.   :12.070   Max.   :89.00   Max.   :100.00  
##  NA's   :9       NA's   :11       NA's   :8                       
```

```
# get the mean
mean(parenthood2$baby.sleep)
```

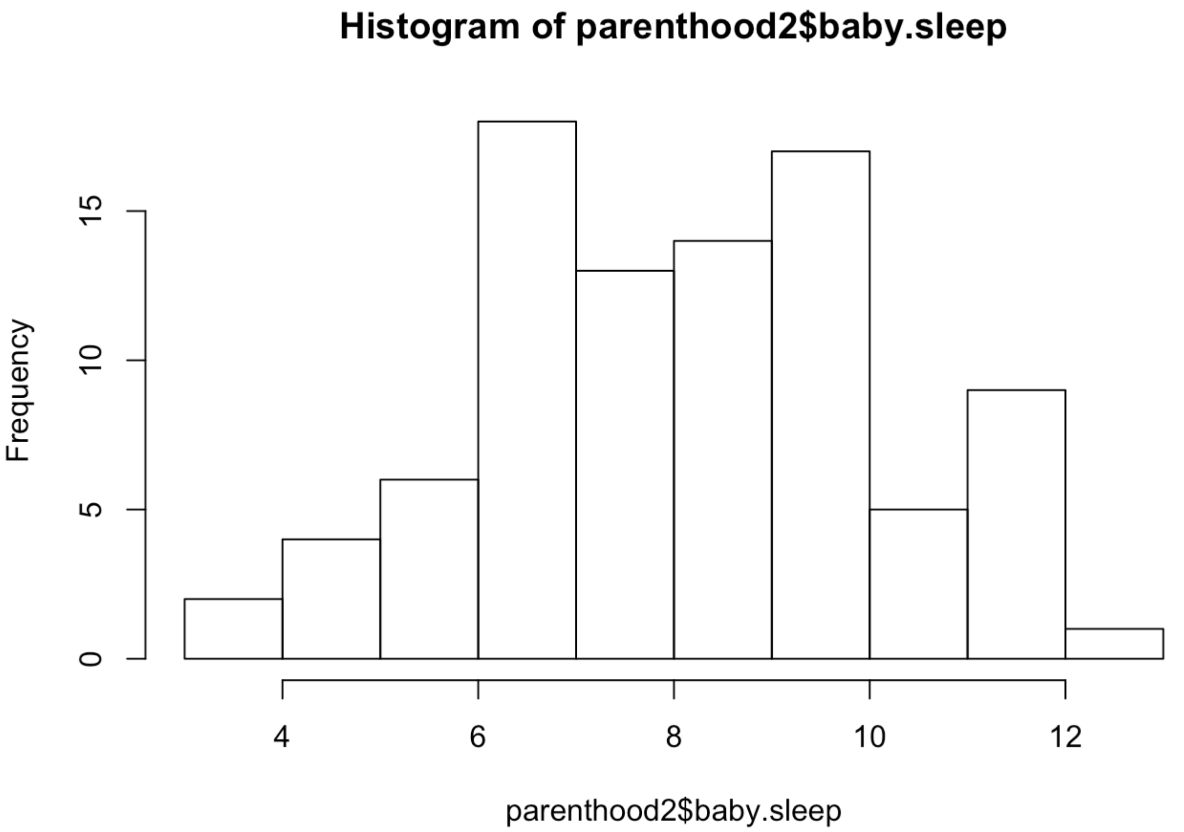
```
## [1] NA
```

```
mean(parenthood2$baby.sleep, na.rm=TRUE)
```

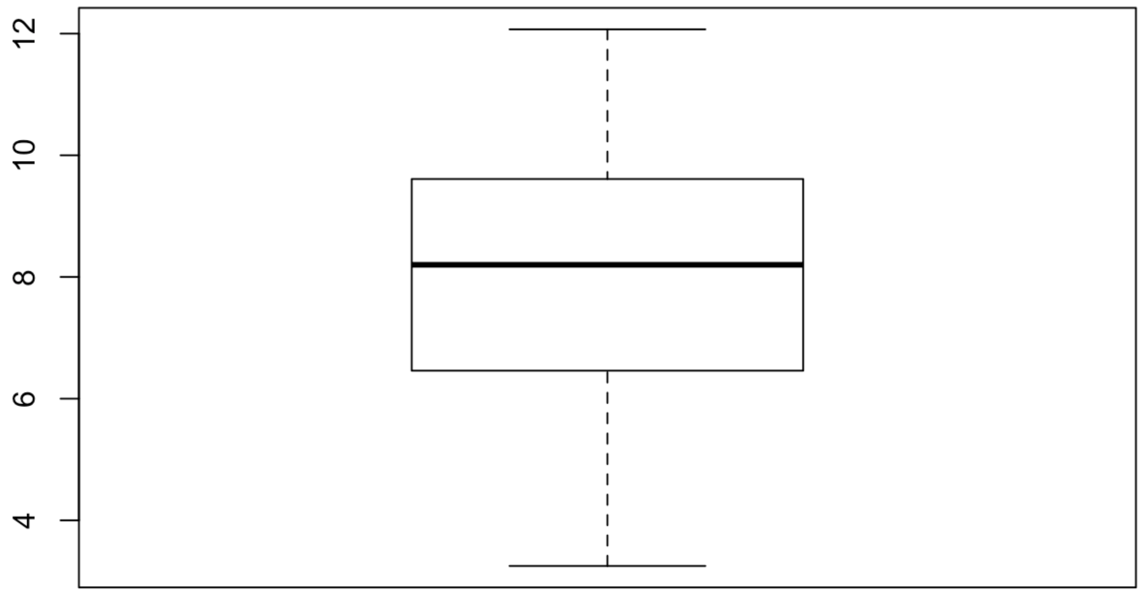
```
## [1] 8.114494
```

Note that: mean, median, sd and other descriptive statistics functions return NA, unless we set na.rm=TRUE, while summary detects missing values and ignores them when calculating descriptive statistics

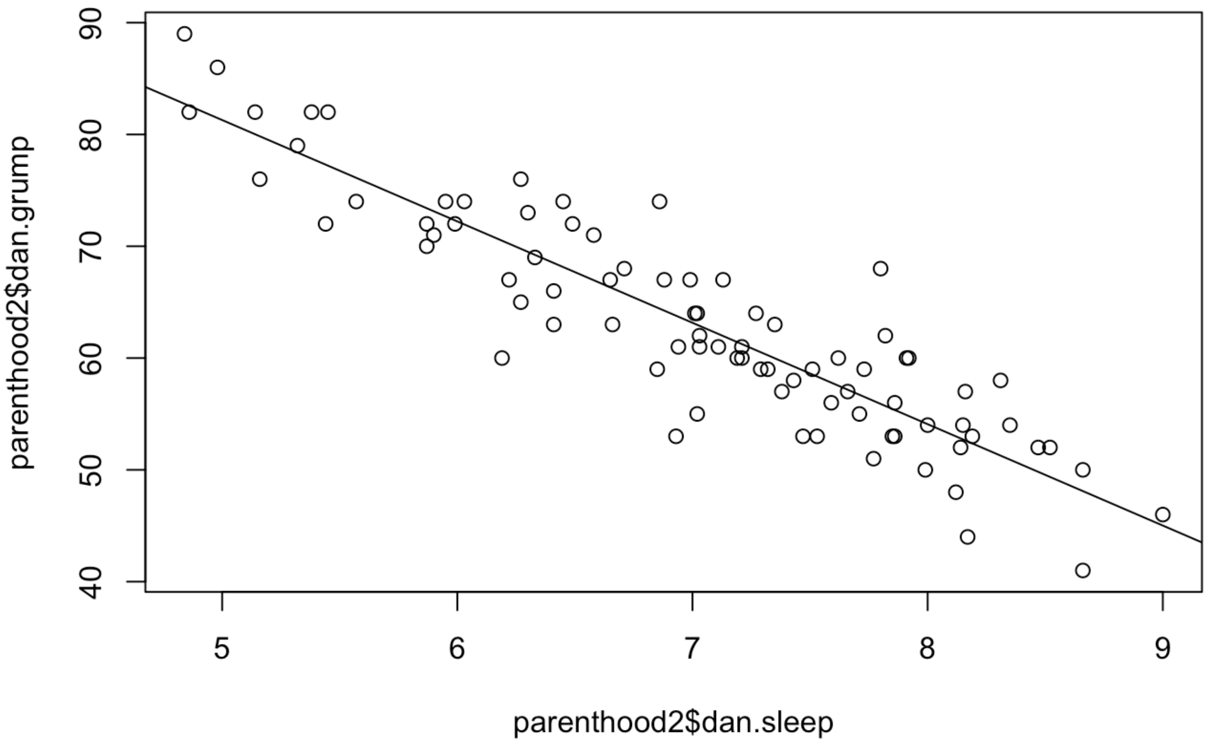
```
# visualise columns that contain missing values
hist(parenthood2$baby.sleep)
```



```
boxplot(parenthood2$baby.sleep)
```



```
plot(parenthood2$dan.sleep, parenthood2$dan.grump)
abline(lm(parenthood2$dan.grump ~ parenthood2$dan.sleep))
```



```
# get the correlation coefficient for two variables that contain missing values
cor(parenthood2$dan.sleep, parenthood2$dan.grump)
```

```
## [1] NA
```

```
cor(parenthood2$dan.sleep, parenthood2$dan.grump, use="complete.obs")
```

```
## [1] -0.9034424
```

Part 3: More data manipulation

```
# import data (csv format)
ds_class2 <- read.csv("DataScienceClass2.csv", header = TRUE, sep = ",")

# get a feel for the data
str(ds_class2)
```

```
## 'data.frame':    30 obs. of  7 variables:
##  $ Grades        : int  69 70 86 42 54 79 69 35 43 58 ...
##  $ Degree        : Factor w/ 4 levels "Informatics",..: 4 1 2 3 1 3 3 2 2 1
...
##  $ Hours.of.sleep: num  8 6.4 8.3 6.2 6 7.4 9 6.1 6.3 6.7 ...
##  $ Gender        : Factor w/ 2 levels "Female","Male": 1 2 1 2 2 1 1 2 2 1
...
##  $ Nationality   : Factor w/ 3 levels "EU","International",..: 3 2 2 3 2 2 1
1 1 3 ...
##  $ StudyYear     : int  2 4 4 4 4 1 4 2 3 1 ...
##  $ DoB           : Factor w/ 30 levels "1990-12-02","1993-08-12",..: 30 3 14
29 21 9 10 13 23 16 ...
```

```
head(ds_class2)
```

	Gra...	Degree	Hours.of.sleep	Gen...	Nationality	StudyYear	DoB
	<int>	<fctr>	<dbl>	<fctr>	<fctr>	<int>	<fctr>
1	69	Psychology	8.0	Female	UK	2	2003-03-29
2	70	Informatics	6.4	Male	International	4	1995-02-12
3	86	Mathematics	8.3	Female	International	4	1998-07-22
4	42	Medicine	6.2	Male	UK	4	2002-08-29
5	54	Informatics	6.0	Male	International	4	2000-06-15
6	79	Medicine	7.4	Female	International	1	1997-04-19

6 rows

Data coercion: converting data from one type to another

```
# coercing to numeric
ds_class2$StudyYear<- as.numeric(ds_class2$StudyYear)
class(ds_class2$StudyYear)
```

```
## [1] "numeric"
```

```
# coercing to character
ds_class2$StudyYear<- as.character(ds_class2$StudyYear)

# coercing to int
ds_class2$StudyYear<- as.integer(ds_class2$StudyYear)

# coercing to factor
ds_class2$StudyYear<- as.factor(ds_class2$StudyYear)

# coercing to date
ds_class2$DoB <- as.Date(ds_class2$DoB, "%Y-%m-%d")
```

Subsetting data

```
# Subset data using the subset() function
df1 <- subset(ds_class2, Degree == "Medicine")
print(df1)
```

```
##    Grades   Degree Hours.of.sleep Gender   Nationality StudyYear
## 4      42 Medicine            6.2   Male            UK         4
## 6      79 Medicine            7.4 Female International         1
## 7      69 Medicine            9.0 Female            EU         4
## 13     68 Medicine            6.9 Female International         4
## 14     40 Medicine            6.5 Female            EU         3
## 22     69 Medicine            7.0 Female            UK         3
## 23     63 Medicine            7.1   Male            EU         2
## 24     75 Medicine            8.1   Male            EU         1
## 26     58 Medicine            6.1 Female            EU         4
##           DoB
## 4  2002-08-29
## 6  1997-04-19
## 7  1997-09-11
## 13 1997-04-12
## 14 2001-06-02
## 22 1996-07-17
## 23 1999-03-14
## 24 2001-05-22
## 26 1995-04-17
```

```
# Use subset() on a numeric/integer column
df2 <- subset(ds_class2, Grades >= 70)
print(df2)
```

```
##    Grades      Degree Hours.of.sleep Gender   Nationality StudyYear
## 2      70 Informatics            6.4   Male International         4
## 3      86 Mathematics            8.3 Female International         4
## 6      79    Medicine            7.4 Female International         1
## 11     95 Informatics            8.6 Female            UK         4
## 16     86 Informatics            7.2   Male International         5
## 17     84  Psychology            9.2   Male            UK         2
## 18     75 Mathematics            7.3   Male            UK         4
## 24     75    Medicine            8.1   Male            EU         1
## 25     71 Mathematics            8.2 Female            EU         1
##           DoB
## 2  1995-02-12
## 3  1998-07-22
## 6  1997-04-19
## 11 1998-07-29
## 16 1996-08-29
## 17 1998-07-05
## 18 1999-02-25
## 24 2001-05-22
## 25 2002-03-02
```

```
# Use subset() and specify which column(s) to keep
df3 <- subset(ds_class2, Grades >= 70, select = Hours.of.sleep)
print(df3)
```

```
##    Hours.of.sleep
## 2             6.4
## 3             8.3
## 6             7.4
## 11            8.6
## 16            7.2
## 17            9.2
## 18            7.3
## 24            8.1
## 25            8.2
```

```
# Selecting (or keeping) variables
vars_to_keep <- c("Degree", "Hours.of.sleep", "Nationality")
df4 <- ds_class2[vars_to_keep]
str(df4)
```

```
## 'data.frame':    30 obs. of  3 variables:
##  $ Degree        : Factor w/ 4 levels "Informatics",..: 4 1 2 3 1 3 3 2 2 1
...
##  $ Hours.of.sleep: num  8 6.4 8.3 6.2 6 7.4 9 6.1 6.3 6.7 ...
##  $ Nationality   : Factor w/ 3 levels "EU","International",..: 3 2 2 3 2 2 1
1 1 3 ...
```

```
# Excluding (or dropping) variables
df5 <- ds_class2[, -1]
str(df5)
```

```
## 'data.frame':    30 obs. of  6 variables:
##  $ Degree        : Factor w/ 4 levels "Informatics",..: 4 1 2 3 1 3 3 2 2 1
...
##  $ Hours.of.sleep: num  8 6.4 8.3 6.2 6 7.4 9 6.1 6.3 6.7 ...
##  $ Gender        : Factor w/ 2 levels "Female","Male": 1 2 1 2 2 1 1 2 2 1
...
##  $ Nationality   : Factor w/ 3 levels "EU","International",..: 3 2 2 3 2 2 1
1 1 3 ...
##  $ StudyYear     : Factor w/ 5 levels "1","2","3","4",..: 2 4 4 4 4 1 4 2 3
1 ...
##  $ DoB           : Date, format: "2003-03-29" "1995-02-12" ...
```

Renaming columns

```
# Get the column names
names(ds_class2)
```

```
## [1] "Grades"          "Degree"          "Hours.of.sleep" "Gender"
## [5] "Nationality"     "StudyYear"       "DoB"
```

```
# Change the name of the 7th column
names(ds_class2)[7] <- "DateOfBirth"

# Change the name of the column "Hours.of.sleep"
names(ds_class2)[names(ds_class2) == "Hours.of.sleep"] <- "HoursOfSleep"
```

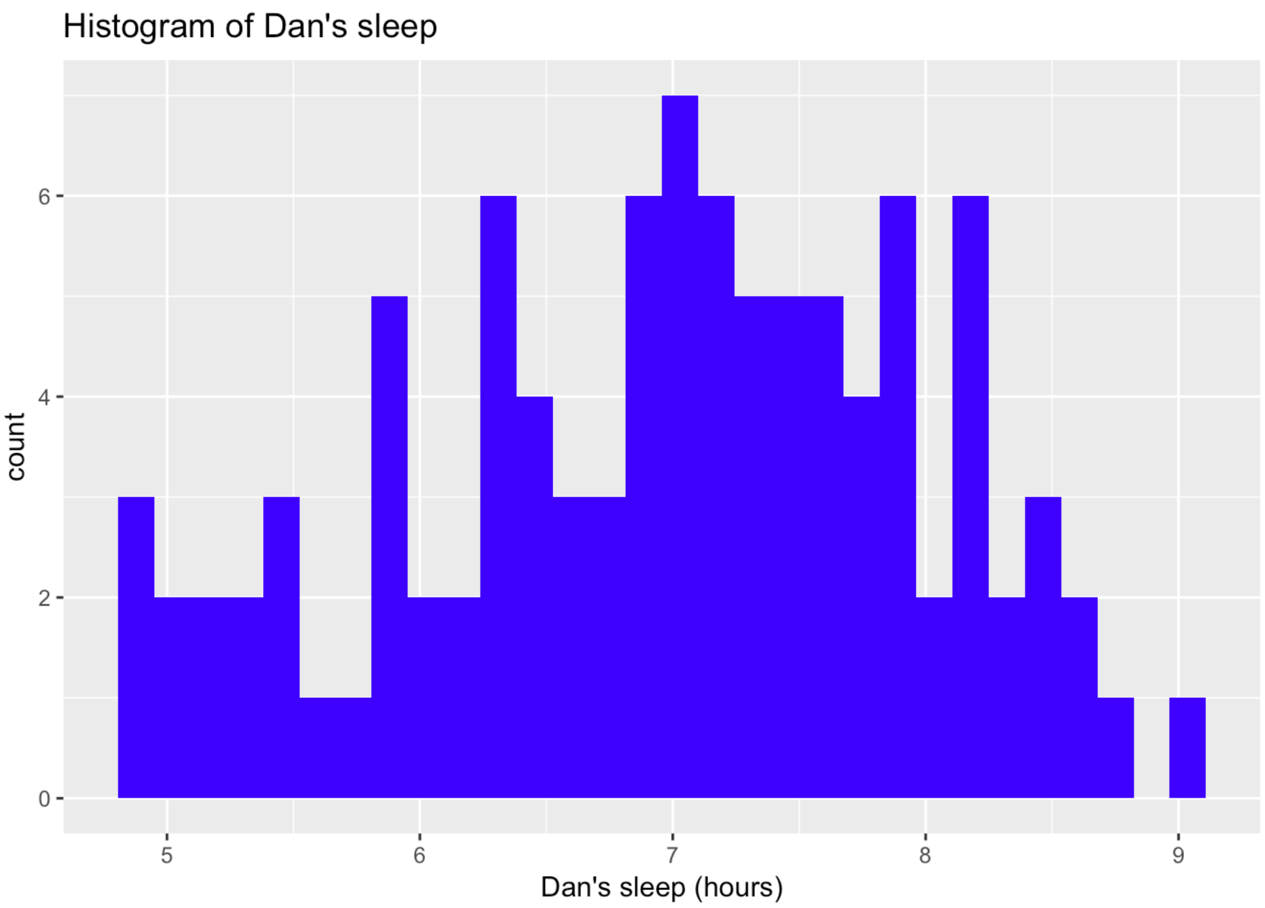
Part 4: Using packages

```
#If the package is not already installed, you will need to install it
#install.packages("ggplot2")

#loading a package (a package needs to be loaded every time you need to use it)
library("ggplot2")

# get a histogram with ggplot2
ggplot(parenthood,
       aes(x = dan.sleep)) +
  geom_histogram(fill = "blue") + # add histogram geom in blue
  labs(title ="Histogram of Dan's sleep", x = "Dan's sleep (hours)") # add labe
ls
```

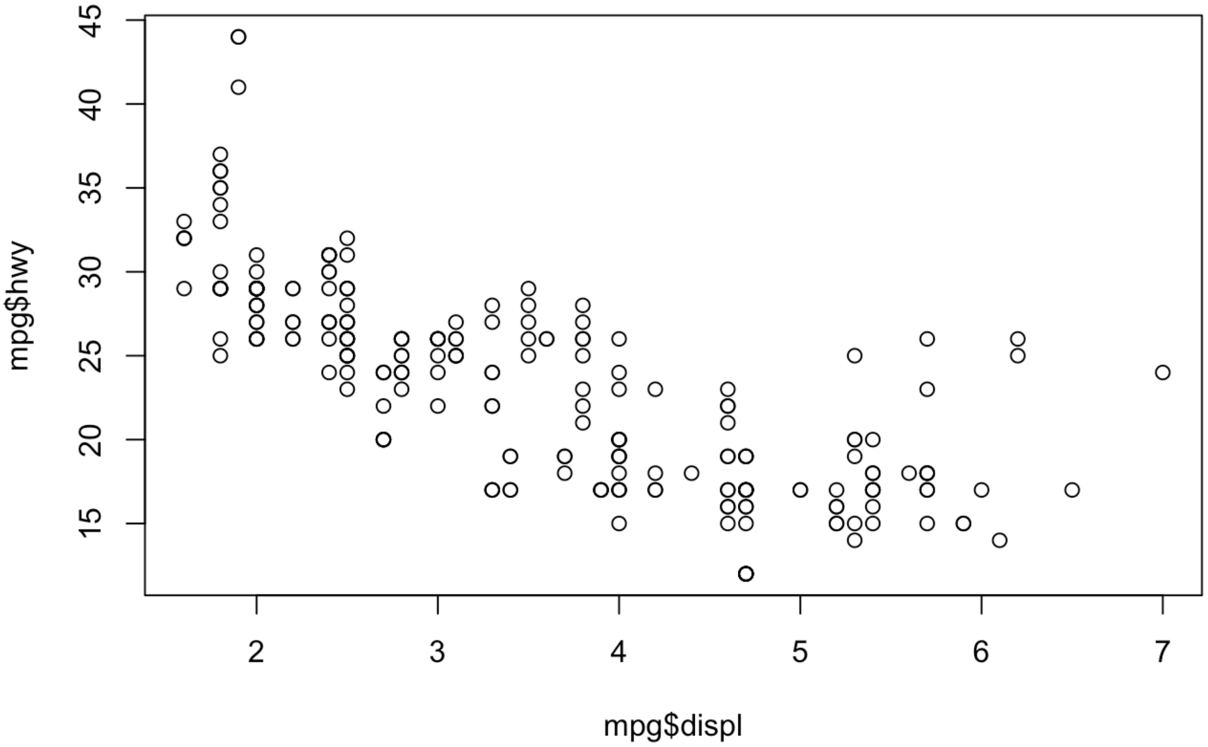
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



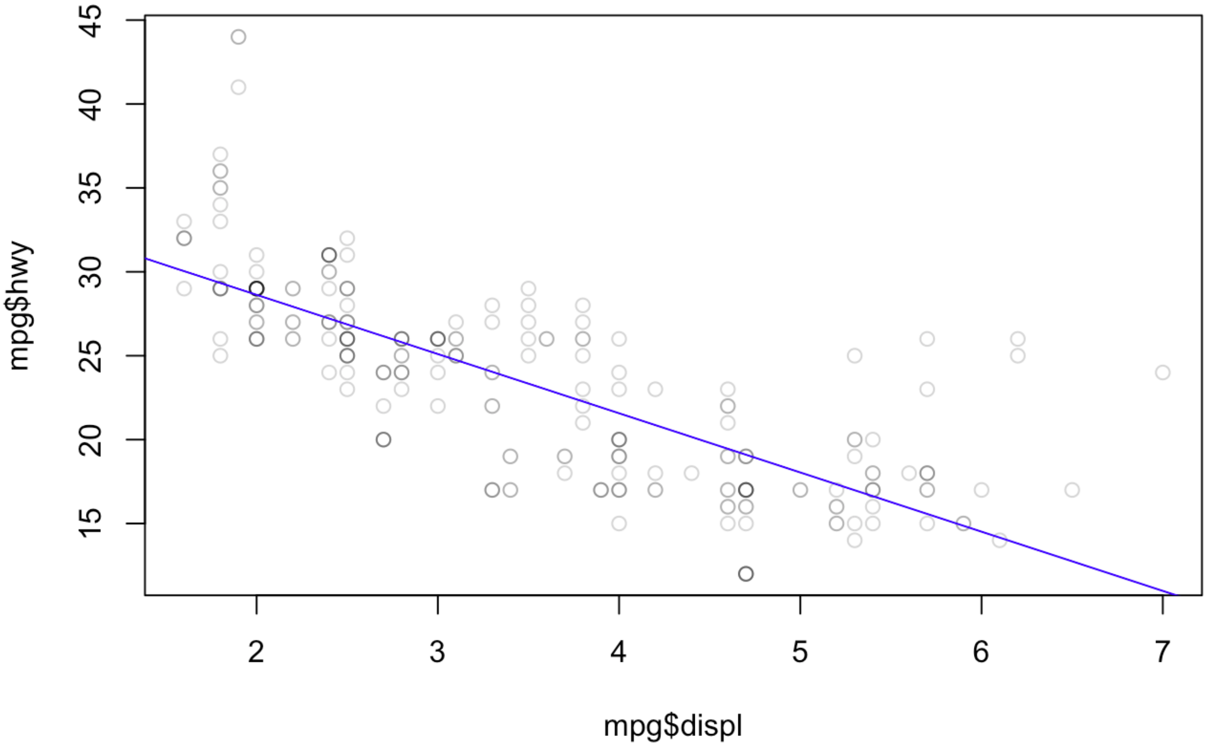
```
# Dealing with overplotting (example with mpg dataset from ggplot2)
str(mpg)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    234 obs. of  11 variables:
##  $ manufacturer: chr  "audi" "audi" "audi" "audi" ...
##  $ model       : chr  "a4" "a4" "a4" "a4" ...
##  $ displ       : num  1.8 1.8 2 2 2.8 2.8 3.1 1.8 1.8 2 ...
##  $ year        : int  1999 1999 2008 2008 1999 1999 2008 1999 1999 2008 ...
##  $ cyl         : int  4 4 4 4 6 6 6 4 4 4 ...
##  $ trans       : chr  "auto(l5)" "manual(m5)" "manual(m6)" "auto(av)" ...
##  $ drv         : chr  "f" "f" "f" "f" ...
##  $ cty         : int  18 21 20 21 16 18 18 18 16 20 ...
##  $ hwy         : int  29 29 31 30 26 26 27 26 25 28 ...
##  $ fl          : chr  "p" "p" "p" "p" ...
##  $ class       : chr  "compact" "compact" "compact" "compact" ...
```

```
plot(mpg$displ, mpg$hwy)
```



```
plot(mpg$displ, mpg$hwy,
     col="#00000033")
abline(lm(mpg$hwy ~ mpg$displ), col = "blue")
```



Part 5: Further practice

Use the dataset provided in Tutorial 2 to practise with what you’ve learnt in this lab