Tutorial 2
Introduction to R and ggplot
Statistical Methods in Research
COSC 6323
Spring 2018

Ioannis Pavlidis
Dinesh Majeti
George Panagopoulos

Computational Physiology Lab

ipavlidis@uh.edu
dmajeti@uh.edu
gpanagopoulos@uh.edu

January 26, 2018

Overview

1. Factors
2. Apply functions
3. Data set
4. ggplot2
5. Exercises on Histogram
6. Exercises on Boxplot

Factors

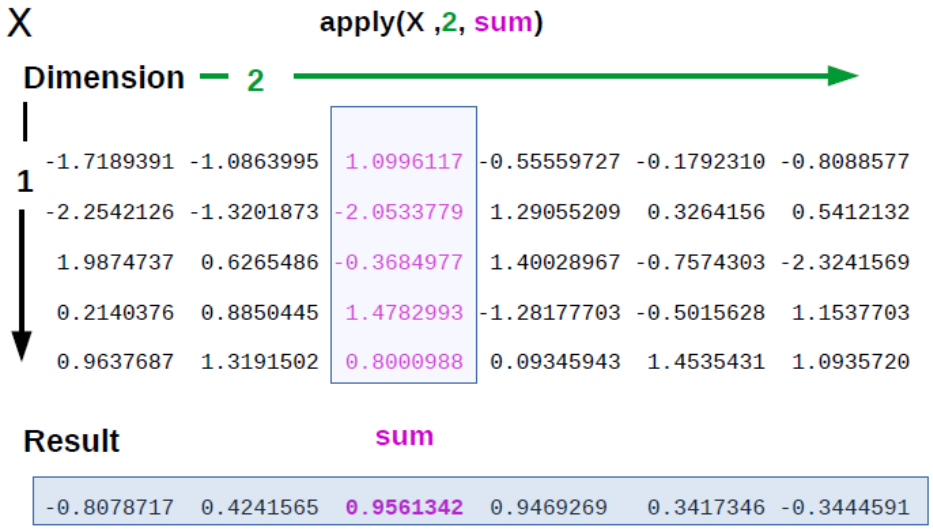
- **Categorical variables in R**

```
> sampleData <- sample(0:1, 20, replace = TRUE)
> is.factor(sampleData)
> is.numeric(sampleData)
> myFactor <- factor(sampleData,
labels = c("low", "high"))
> is.factor(sampleData)
```

Split Apply Combine functions - Apply

- apply functions in slices of matrices, arrays, lists and dataframes
- avoid explicit use of loop constructs

```
X=matrix(rnorm(30), nrow=5, ncol=6)
apply(X,2,sum)
```



Split Apply Combine functions - lapply

- lapply
- apply a function to each element of a list and get a list back

```
# create a list with 2 elements
l = list(a = 1:10, b = 11:20)
# the mean of the values in each element
lapply(l, mean)
$a
[1] 5.5

$b
[1] 15.5
```

Split Apply Combine functions - tapply

- tapply
- apply a function to subsets of a vector and the subsets are defined by some other vector, usually a factor

```
> x = 1:20
> letters = c("a","b","c","d","e")
> y = factor(rep(letters, each = 4))
> y
 [1] a a a a b b b b c c c c d d d d e e e e
Levels: a b c d e
> x
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
> tapply(x, y, sum)
 a  b  c  d  e
10 26 42 58 74
```

Data set

- 'mtcars' dataset - Motor Trend Car Road Tests
- Built-in data frame in R
- The data was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973-74 models).

```
>data(mtcars)
> dim(mtcars)
nrow(mtcars), ncol(mtcars)
> str(mtcars)
```

Data set

- 32 observations with 11 variables.

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1

Figure 1: mtcars dataset

```
> help mtcars
```

Exploring the dataset

- Ways to access

```
> mtcars[1, 4]
> mtcars["Mazda RX4", "hp"]
> head(mtcars)
> tail(mtcars, 5)
x[i,j] element at row i, column j
x[i,] row i
x[,j] column j
x[,c(1,3)] columns 1 and 3
mtcars["AMC Javelin",]row named "name"
mtcars[,"mpg"] column named "mpg"
```

Exploring the dataset

- Ways to access

```
> mtcars$mpg
> names(mtcars)
> mpg  # not currently visible
> attach(mtcars)
> mpg
> table(cyl)
> detach()
```

Histogram

```
hist(mtcars$hp, xlab="hp",
main="Histogram of horse power of cars")
```

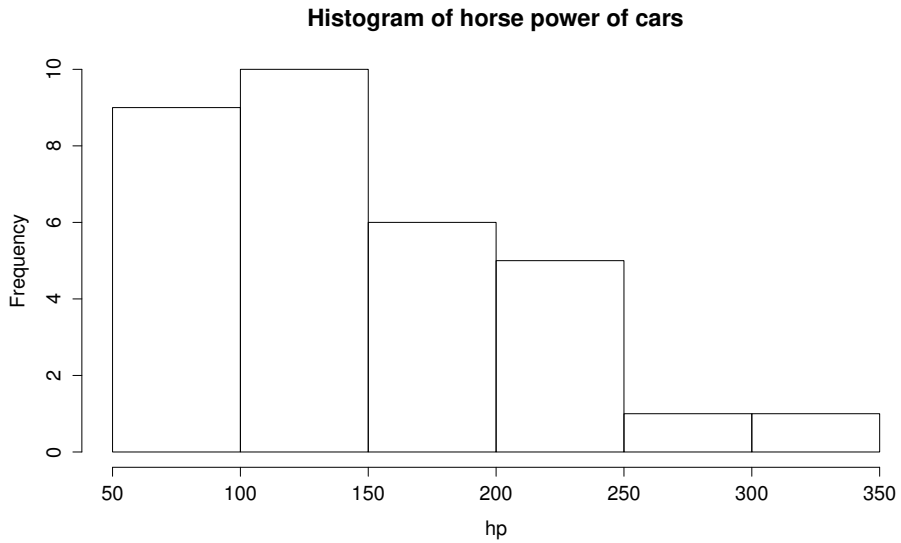


Figure 2: Histogram in R

Histogram with custom bins

```
hist(mtcars$hp, xlab="hp",
breaks=seq(0, 350, by=10),
main="Histogram of horse power of cars")
```

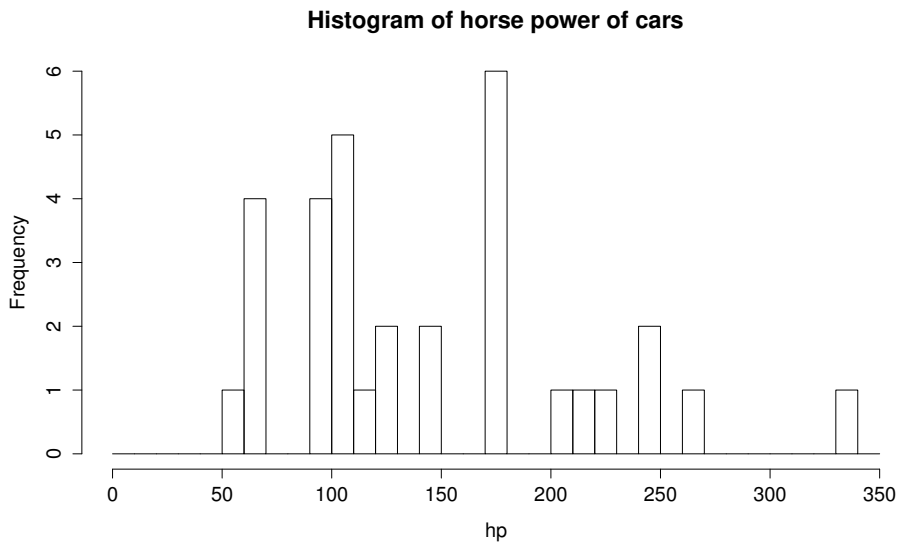


Figure 3: Histogram with custom bin width

Histogram with density curve

```
hist(mtcars$hp, xlab="hp",
main="Histogram of horse power of cars", freq=FALSE)
lines(density(mtcars$hp))
```

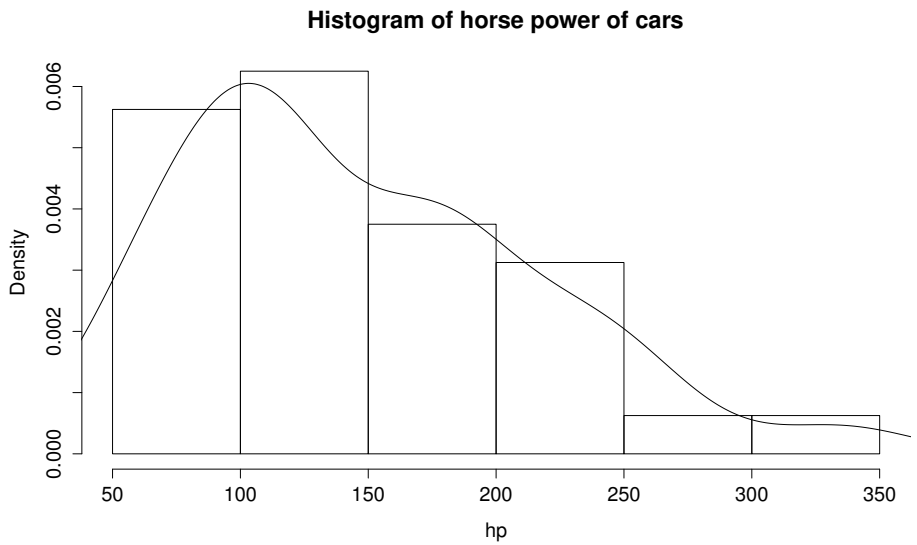


Figure 4: Histogram with density curve

ggplot2

- On his website (http://had.co.nz/ggplot2/) package author Hadley Wickham describes ggplot2 as -
- "plotting system for R, based on the grammar of graphics... It takes care of many of the fiddly details that make plotting a hassle (like drawing legends) as well as providing a powerful model of graphics that makes it easy to produce complex multi-layered graphics."
- independently specify plot building blocks and combine them to create just about any kind of graphical display you want
- modular
- flexible

Grammar of graphics

- Leland Wilkinson (2005)
- **Data** are the variables mapped to aesthetic features of the graph.
- **Geoms** are the objects/shapes you see on the graph.
- **Stats** are stastical transformations that summarize data, such as the mean or confidence intervals.
- **Scales** define which aesthetic values are mapped to data values. Legends and axes display these mappings.
- **Coordiante systems** define the plane on which data are mapped on the graphic.
- **Faceting** splits the data into subsets to create multiple variations of the same graph (paneling).

ggplot2

- 2 major functions -
- `qplot()` - for quick plots
- `ggplot()` - for fine, granular control of everything

```
install.packages("ggplot2")
library(ggplot2)
```

qplot - Histogram

```
qplot(data=mtcars, x=hp)
```

```
ggplot(data=mtcars, aes(hp)) + geom_histogram()
```

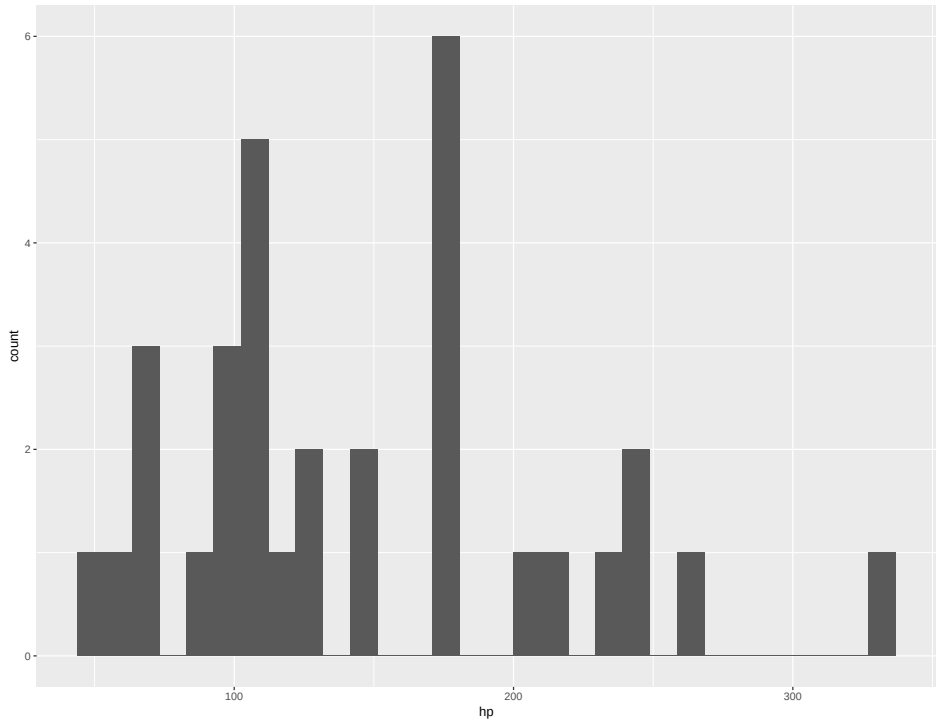


Figure 5: Histogram Plot

Exercise on Histogram

- Obtain the histogram of mpg (miles per gallon) with the following bin widths
  - 2
  - 3
  - 4
  - 9
- All the plots should be in a single figure
- Ensure that the scale on all the plots is the same

Hints

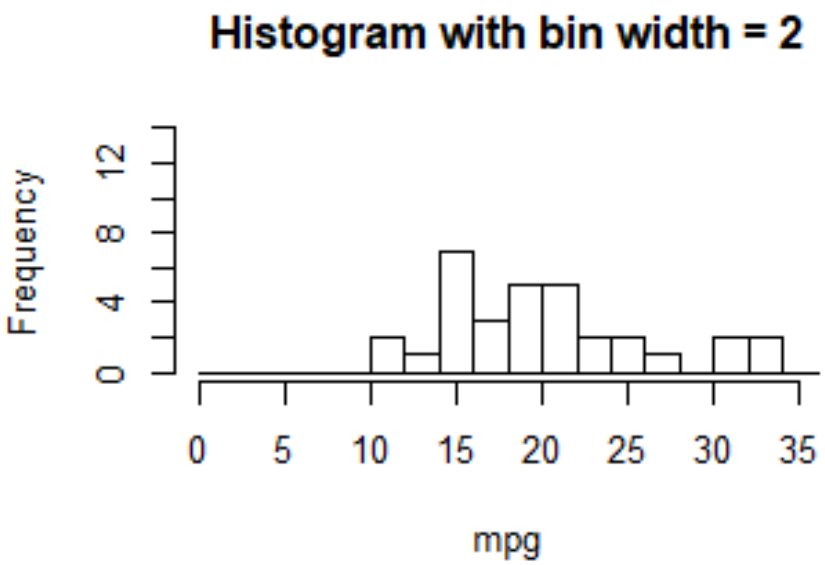
- Customizing the bins of the histogram

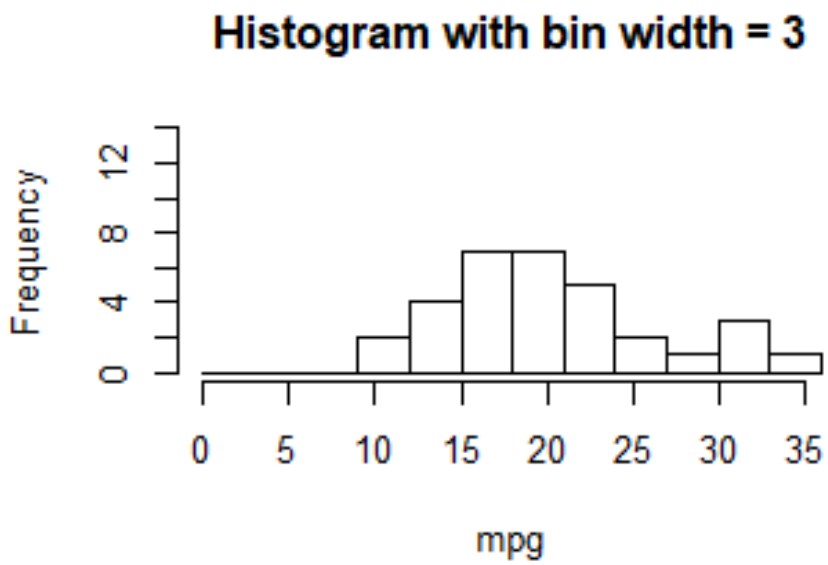
  `breaks`
- Having multiple plots together

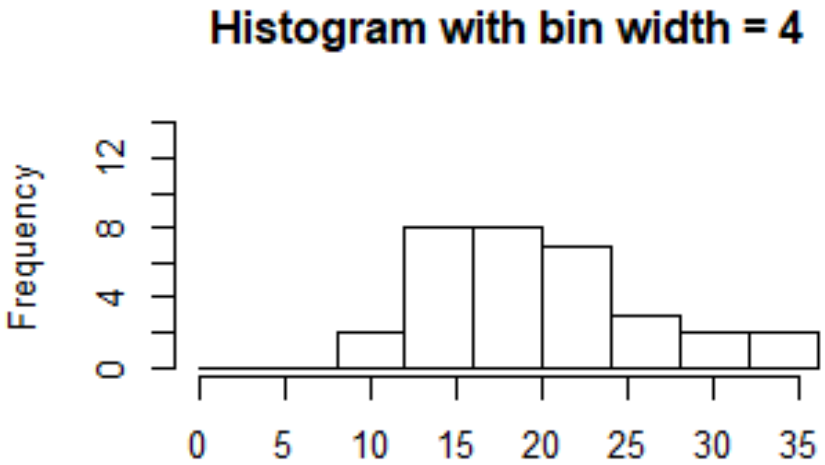
  `par()`, `mfrow`
- Set yscale of histogram

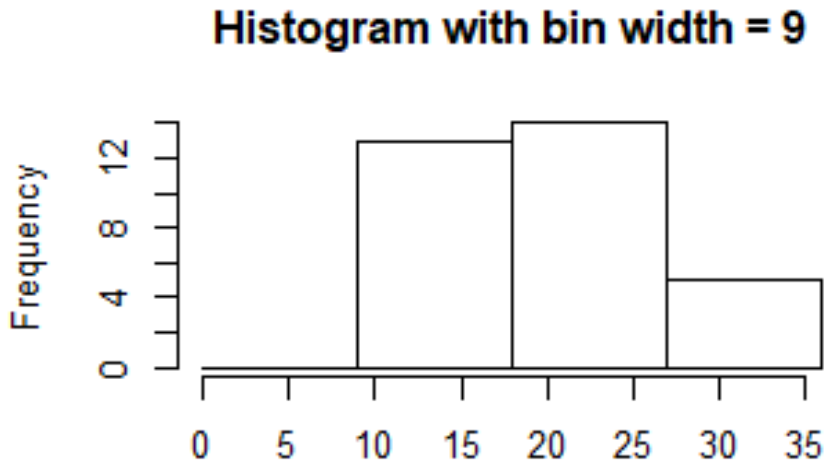
  `ylim`

Solution


Histogram with bin width = 2
Frequency
0
4
8
12
0
5
10
15
20
25
30
35
mpg


Histogram with bin width = 3
Frequency
0
4
8
12
0
5
10
15
20
25
30
35
mpg


Histogram with bin width = 4
Frequency
0
4
8
12
0
5
10
15
20
25
30
35


Histogram with bin width = 9
Frequency
0
4
8
12
0
5
10
15
20
25
30
35

Solution

```
bins2 = seq(0, 36, by=2)
bins3 = seq(0, 36, by=3)
bins4 = seq(0, 36, by=4)
bins9 = seq(0, 36, by=9)
par( mfrow=c(2,2) )
yrange = c(0,14)
hist(mpg, breaks=bins2, main="Histogram with bin width = 2",
ylim=yrange)
hist(mpg, breaks=bins3, main="Histogram with bin width = 3",
ylim=yrange)
hist(mpg, breaks=bins4, main="Histogram with bin width = 4",
ylim=yrange)
hist(mpg, breaks=bins9, main="Histogram with bin width = 9",
ylim=yrange)
```

Exercise on Boxplot

- Create a box plot of mpg for cars with 5 gears
- Also plot the mean in the boxplot with the appropriate labels and titles
- To plot the mean, use

  `points()`

Solution

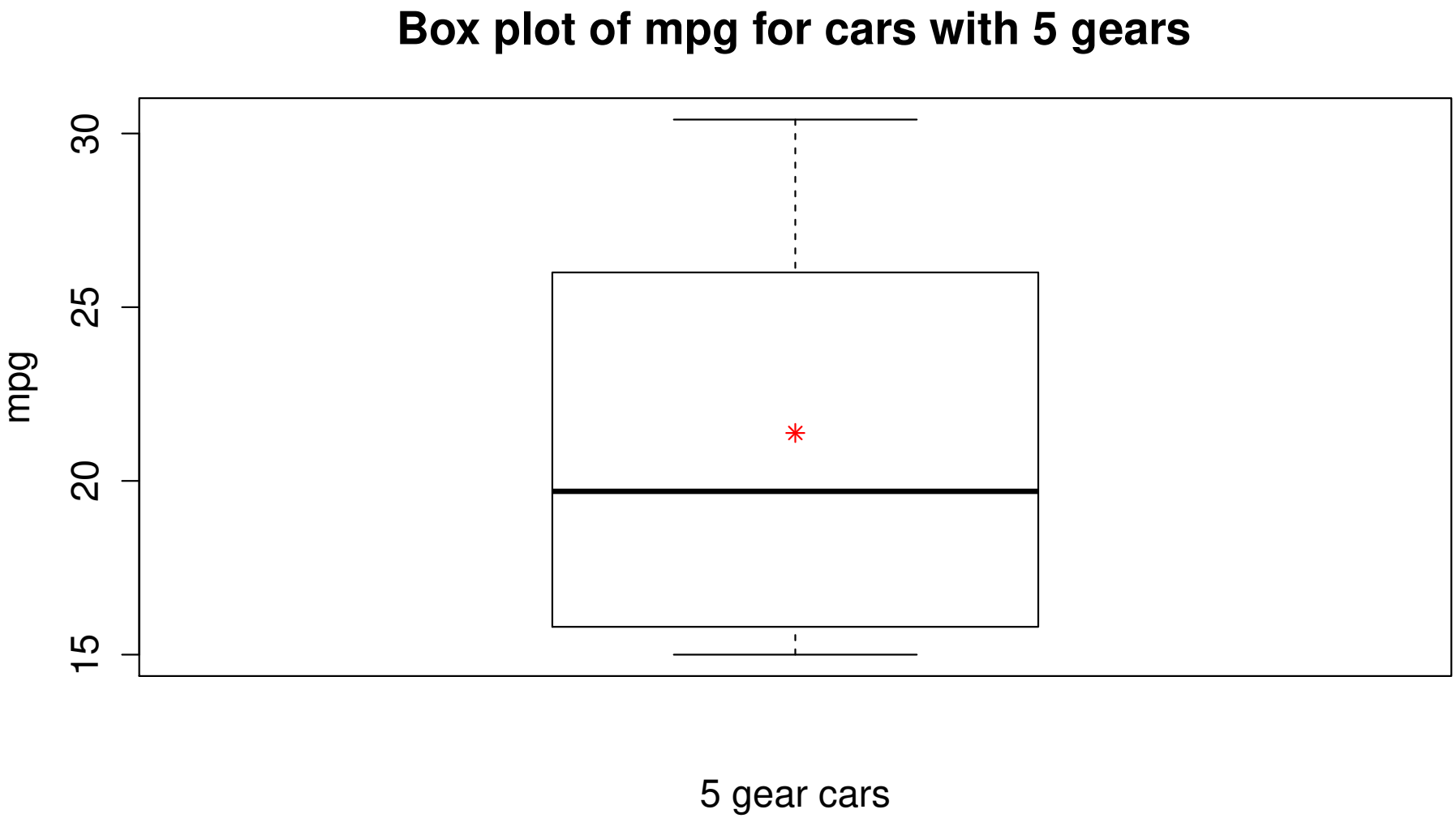


Figure 7: Boxplot

Solution

```
boxplot(mpg[gear==5], xlab="5 gear cars", ylab="mpg")
mean_value = mean(mtcars$mpg[mtcars$gear==5])
points(mean_value, col="red", pch=8)
title("Box plot of mpg for cars with 5 gears")
```

Exercise on Boxplot

- Create a factor of the three car types: fast, medium, and slow, based on the number of gears
- Slow cars have 3 gears, medium cars have 4 gears and fast cars have 5 gears
- Create a box plot of the mpg (miles per gallon) for all the 3 types of cars in a single figure
- The plot should include the mean values.

Solution

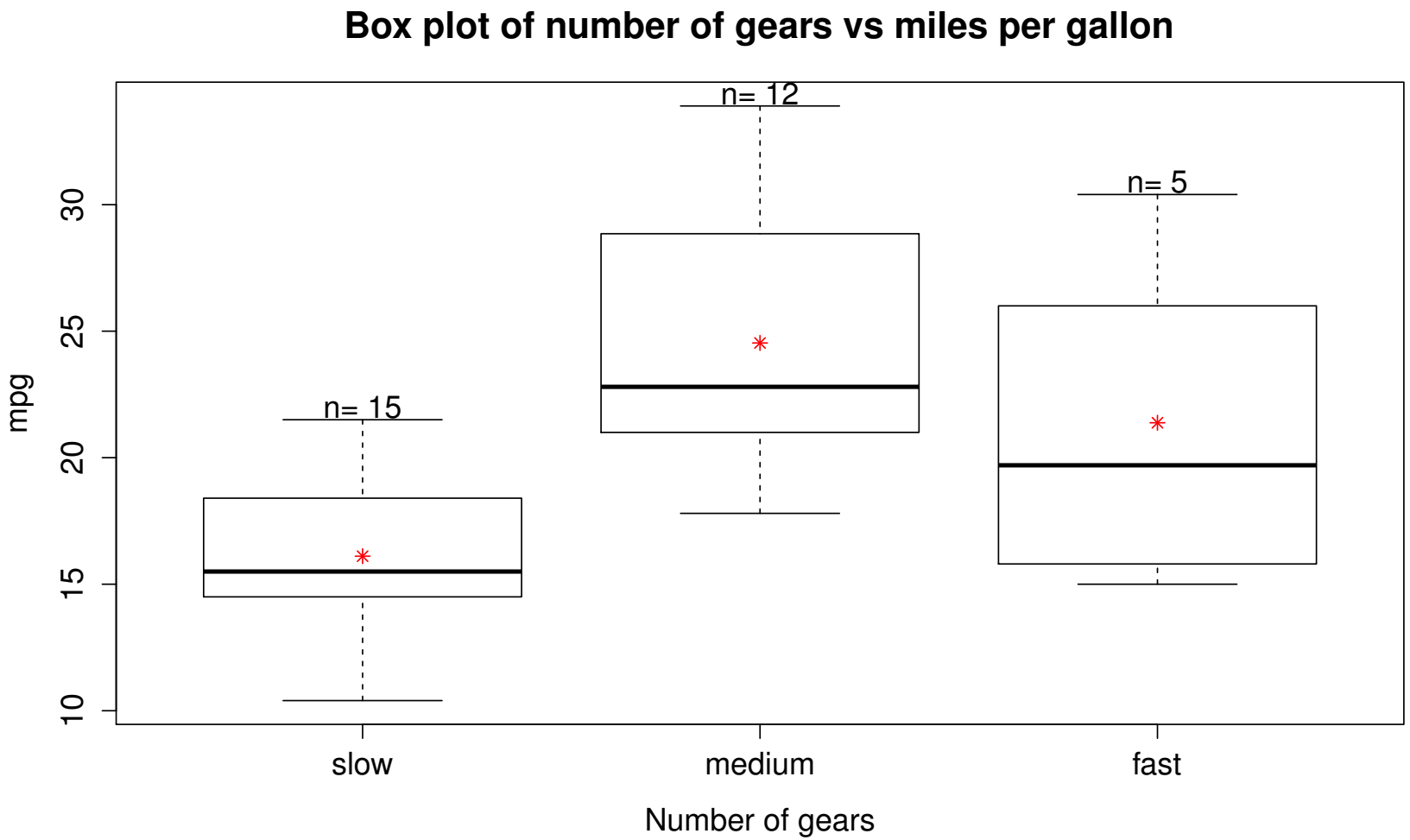


Figure 8: Boxplot

Solution

```
gearFactor = factor(mtcars$gear,
labels=c("slow","medium","fast"))
boxplot(mtcars$mpg ~ gearFactor, xlab="Number of gears",
ylab="mpg")
title("Box plot of number of gears vs miles per gallon")
mean_values = tapply(mpg, gearFactor, mean)
points(mean_values, col="red", pch=8)
```

Solution

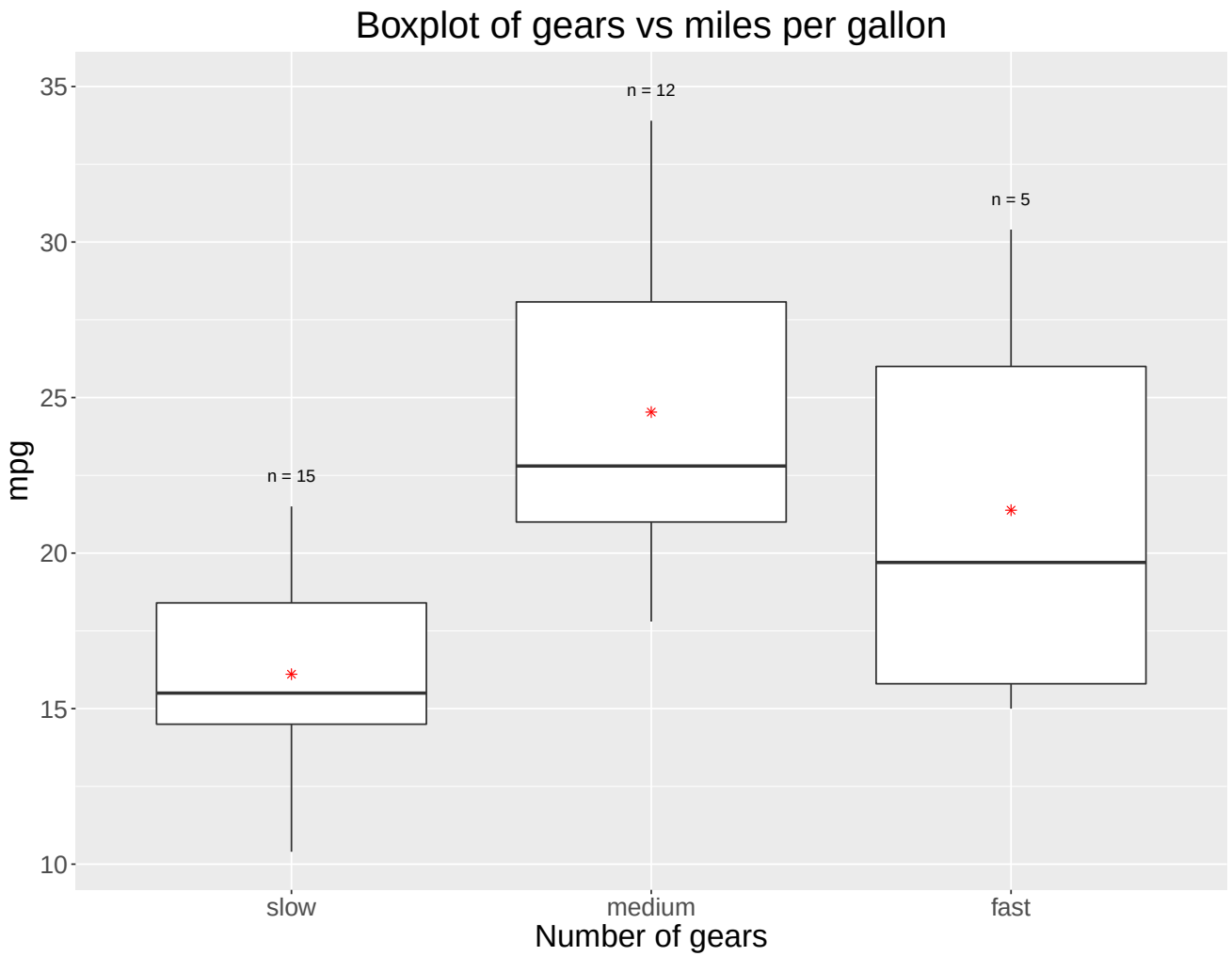


Figure 9: Boxplot - ggplot

Solution

```
give.n <- function(x){
  return(data.frame(y = max(x)+1,
                    label = paste0("n = ",length(x))))
}

ggplot(mtcars, aes(x = gearFactor, y = mpg)) +
  geom_boxplot() +
  scale_x_discrete(name="Number of gears") +
  ggtitle("Boxplot of gears vs miles per gallon") +
  theme(plot.title = element_text(hjust = 0.5)) +
  stat_summary(fun.y="mean", geom="point", size=2, pch=8,
               color="red") +
  stat_summary(fun.data = give.n, geom = "text") +
  theme(text = element_text(size=20))
```

More Exercises with Boxplots

- Create a box plot of the hp (horse power) for all the 3 types of cars in a single figure
- Create a factor of the three car types: 4cyl, 6cyl and 8cyl, based on the number of cylinders
- For the three types of cars, create a single figure with boxplots for
  - hp (horse power)
  - mpg (miles per gallon)
  - gear
  - wt (weight)

For more on R

- R reference manuals - http://cran.r-project.org/manuals.html
- R reference card - http://cran.r-project.org/doc/contrib/Short-refcard.pdf
- Introduction to Apply - https://www.datacamp.com/community/tutorials/r-tutorial-apply-family
- Apply functions - http://stackoverflow.com/questions/3505701/r-grouping-functions-sapply-vs-lapply-vs-apply-vs-tapply-
- ggplot2 - http://www.ats.ucla.edu/stat/r/seminars/ggplot2_intro/ggplot2_intro.htm
- ggplot2 vs base graphics - http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html