

GÖRAN KIRCHNER

NOTES ON R

1

Packages

2

Data

```
head(movies)
```

Title	Year	Rating	Runtime	Critic.Score	Box.Office	Awards	International
The Whole Nine Yards	2000	R	98	45	57.3	FALSE	FALSE
Cirque du Soleil: Journey of Man	2000	G	39	45	13.4	TRUE	FALSE
Gladiator	2000	R	155	76	187.3	TRUE	TRUE
Dinosaur	2000	PG	82	65	135.6	TRUE	FALSE
Big Momma's House	2000	PG-13	99	30	0.5	TRUE	TRUE
Gone in Sixty Seconds	2000	PG-13	118	24	101	TRUE	FALSE

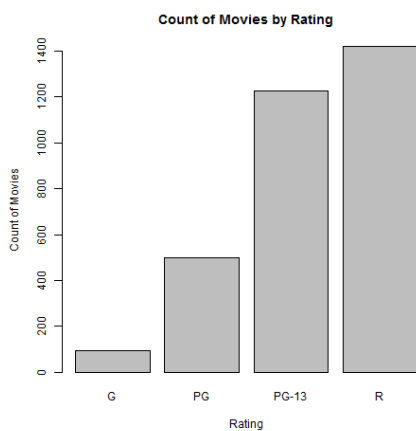
3

Simple Visualization

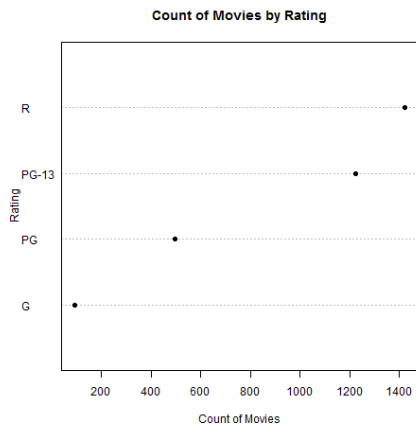
3.1 One Categorical Variable

3.1.1 base

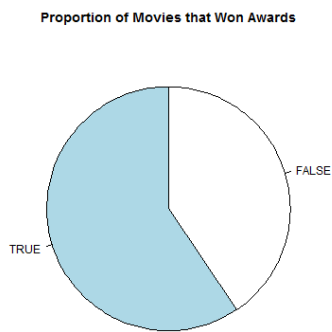
```
plot(  
  x = movies$Rating,  
  main = "Count of Movies by Rating",  
  xlab = "Rating",  
  ylab = "Count of Movies")
```



```
dotchart(  
  x = table(movies$Rating),  
  pch = 16,  
  main = "Count of Movies by Rating",  
  xlab = "Count of Movies",  
  ylab = "Rating")
```



```
pie(
  x = table(movies$Awards),
  clockwise = TRUE,
  main = "Proportion of Movies that Won Awards")
```

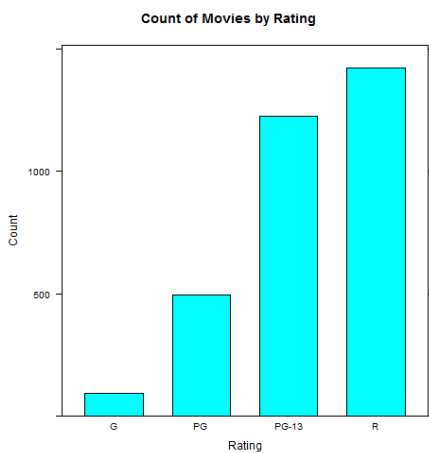


3.1.2 *lattice*

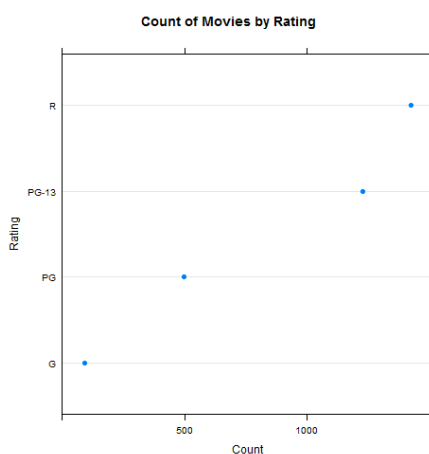
```
library(lattice)
# Create frequency table of ratings
movies <- read.csv("data/movies.csv")
table <- table(movies$Rating)
ratings <- as.data.frame(table)
names(ratings)[1] <- "Rating"
names(ratings)[2] <- "Count"
print(ratings)
```

Rating	Count
G	93
PG	497
PG-13	1225
R	1423

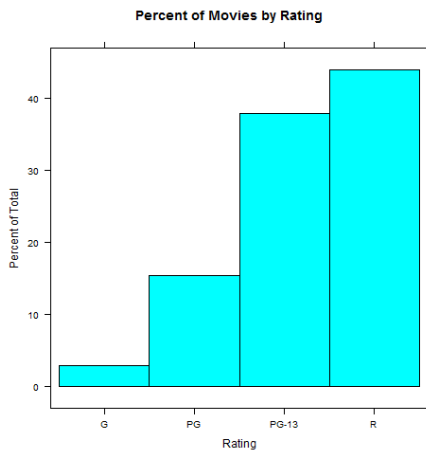

```
library(lattice)
# Create frequency table of ratings
movies <- read.csv("data/movies.csv")
table <- table(movies$Rating)
ratings <- as.data.frame(table)
names(ratings)[1] <- "Rating"
names(ratings)[2] <- "Count"
barchart(
  x = Count ~ Rating,
  data = ratings,
  main = "Count of Movies by Rating",
  xlab = "Rating")
```



```
library(lattice)
# Create frequency table of ratings
movies <- read.csv("data/movies.csv")
table <- table(movies$Rating)
ratings <- as.data.frame(table)
names(ratings)[1] <- "Rating"
names(ratings)[2] <- "Count"
dotplot(
  x = Rating ~ Count,
  data = ratings,
  main = "Count of Movies by Rating",
  ylab = "Rating")
```

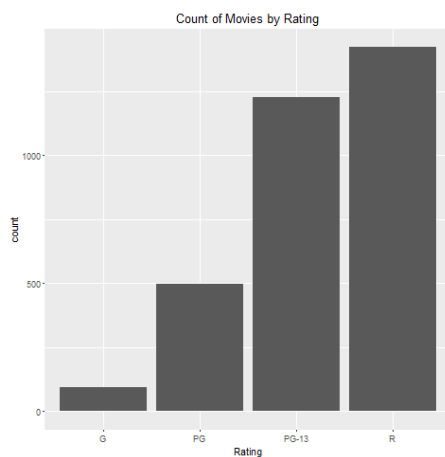


```
library(lattice)
# Create frequency table of ratings
movies <- read.csv("data/movies.csv")
table <- table(movies$Rating)
ratings <- as.data.frame(table)
names(ratings)[1] <- "Rating"
names(ratings)[2] <- "Count"
histogram(
  x = ~Rating,
  data = movies,
  main = "Percent of Movies by Rating")
```

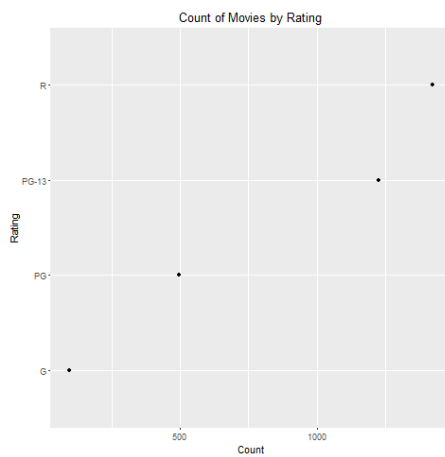


3.1.3 *ggplot2*

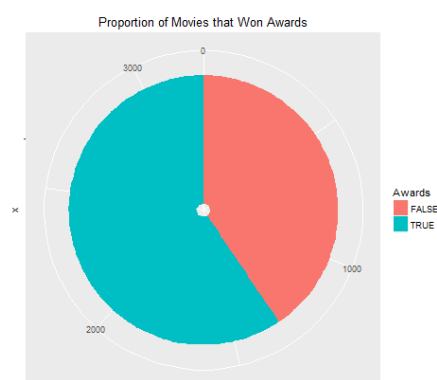
```
library(ggplot2)
movies <- read.csv("data/movies.csv")
ggplot(
  data = movies,
  aes(x = Rating)) +
  geom_bar() +
  ggtitle("Count of Movies by Rating")
```



```
library(ggplot2)
library(lattice)
# Create frequency table of ratings
movies <- read.csv("data/movies.csv")
table <- table(movies$Rating)
ratings <- as.data.frame(table)
names(ratings)[1] <- "Rating"
names(ratings)[2] <- "Count"
ggplot(
  data = ratings,
  aes(x = Rating, y = Count)) +
  geom_point() +
  coord_flip() +
  ggtitle("Count of Movies by Rating")
```



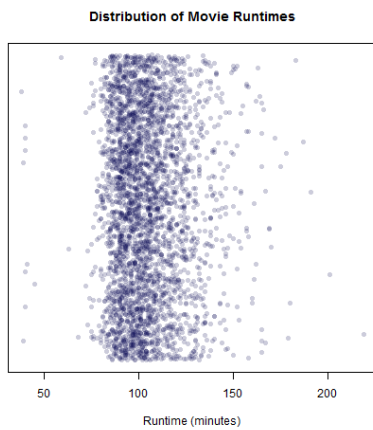
```
library(ggplot2)
movies <- read.csv("data/movies.csv")
ggplot(
  data = movies,
  aes(x = "", fill = Awards)) +
  geom_bar() +
  coord_polar(theta = "y") +
  ggtitle("Proportion of Movies that Won Awards") +
  ylab("")
```



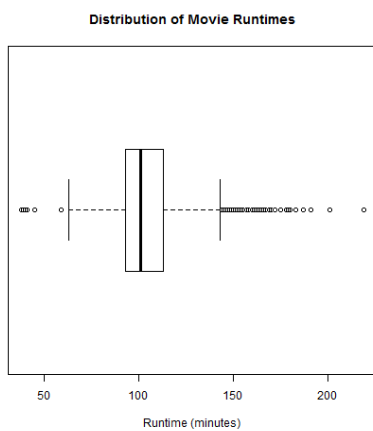
3.2 One Numeric Variable

3.2.1 base

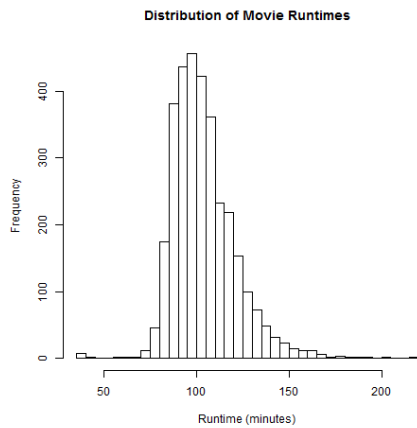
```
movies <- read.csv("data/movies.csv")
plot(
  x = movies$Runtime,
  y = jitter(rep(0, nrow(movies))),
  main = "Distribution of Movie Runtimes",
  xlab = "Runtime (minutes)",
  ylab = "",
  yaxt = "n",
  pch = 16,
  col = rgb(0, 0, 0.3, 0.2))
```



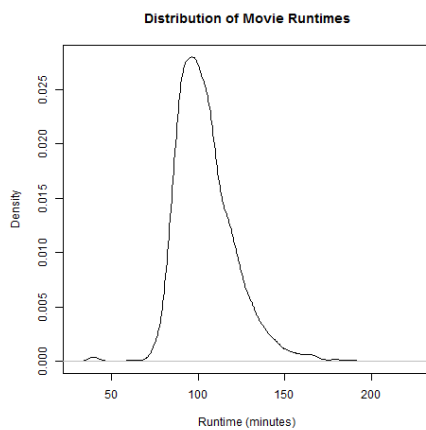
```
movies <- read.csv("data/movies.csv")
boxplot(
  x = movies$Runtime,
  horizontal = TRUE,
  main = "Distribution of Movie Runtimes",
  xlab = "Runtime (minutes)")
```



```
movies <- read.csv("data/movies.csv")
hist(
  x = movies$Runtime,
  breaks = 30,
  main = "Distribution of Movie Runtimes",
  xlab = "Runtime (minutes)")
```

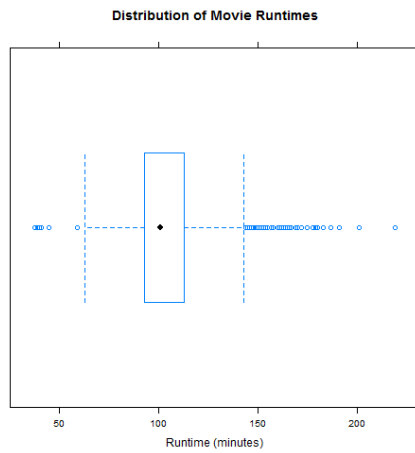


```
movies <- read.csv("data/movies.csv")
plot(
  x = density(movies$Runtime),
  main = "Distribution of Movie Runtimes",
  xlab = "Runtime (minutes)")
```

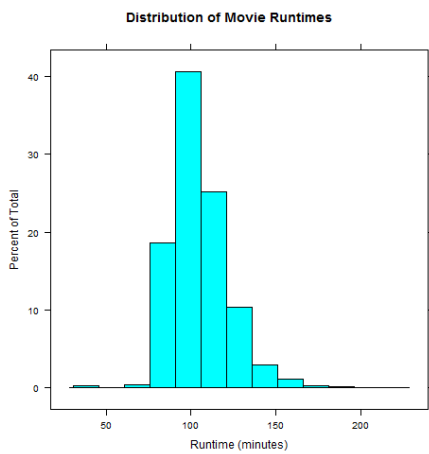


3.2.2 *lattice*

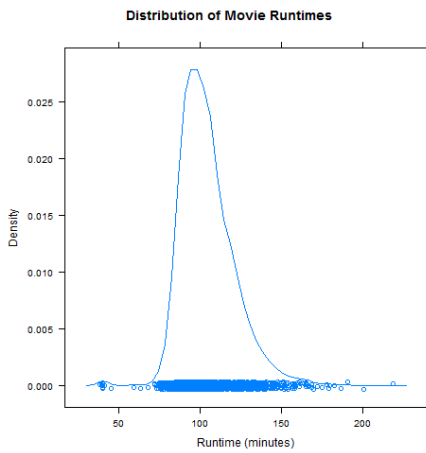
```
movies <- read.csv("data/movies.csv")
library(lattice)
bwplot(
  x = ~Runtime,
  data = movies,
  main = "Distribution of Movie Runtimes",
  xlab = "Runtime (minutes)")
```



```
movies <- read.csv("data/movies.csv")
library(lattice)
histogram(
  x = ~Runtime,
  data = movies,
  main = "Distribution of Movie Runtimes",
  xlab = "Runtime (minutes)")
```

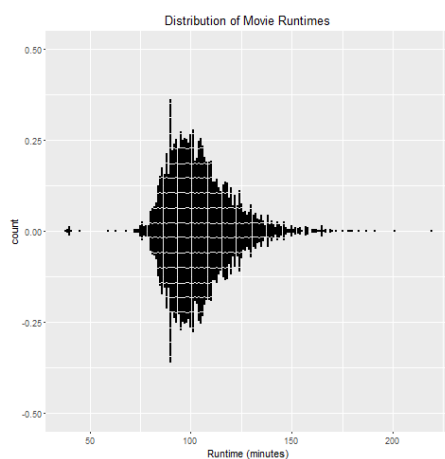


```
movies <- read.csv("data/movies.csv")
library(lattice)
densityplot(
  x = ~Runtime,
  data = movies,
  main = "Distribution of Movie Runtimes",
  xlab = "Runtime (minutes)")
```

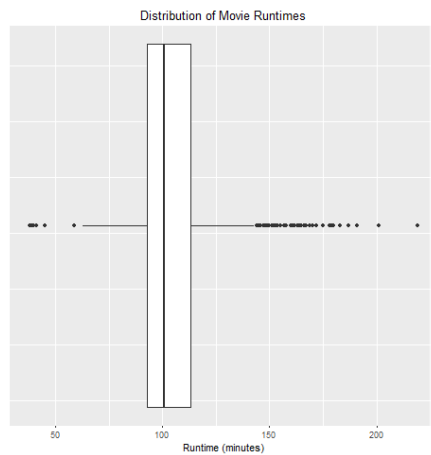


3.2.3 *ggplot2*

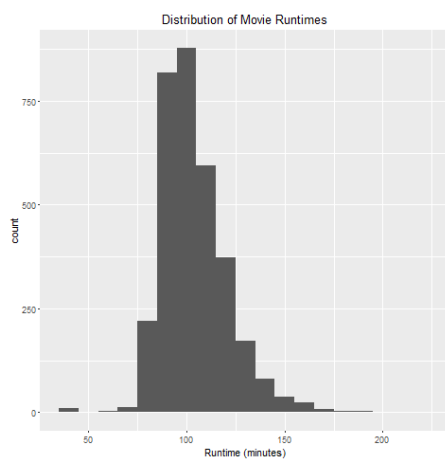
```
library(ggplot2)
movies <- read.csv("data/movies.csv")
ggplot(
  data = movies,
  aes(x = Runtime, stat = "count")) +
  geom_dotplot(
    binwidth = 1,
    stackdir = "center") +
  ggtitle("Distribution of Movie Runtimes") +
  xlab("Runtime (minutes)")
```



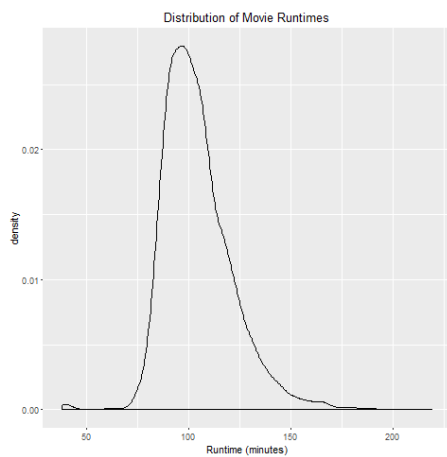
```
library(ggplot2)
movies <- read.csv("data/movies.csv")
ggplot(
  data = movies,
  aes(x = Runtime, y = Runtime)) +
  geom_boxplot() +
  coord_flip() +
  ggtitle("Distribution of Movie Runtimes") +
  xlab("") +
  ylab("Runtime (minutes)") +
  theme(
    axis.text.y = element_blank(),
    axis.ticks.y = element_blank())
```



```
library(ggplot2)
movies <- read.csv("data/movies.csv")
ggplot(
  data = movies,
  aes(x = Runtime)) +
  geom_histogram(binwidth = 10) +
  ggtitle("Distribution of Movie Runtimes") +
  xlab("Runtime (minutes)")
```



```
library(ggplot2)
movies <- read.csv("data/movies.csv")
ggplot(
  data = movies,
  aes(x = Runtime)) +
  geom_density() +
  ggtitle("Distribution of Movie Runtimes") +
  xlab("Runtime (minutes)")
```

3.3 *Two Categorical Variables*

3.4 *Two Numeric Variables*

3.5 *Both a Categorical and a Numeric Variable*

Intermediate Visualization

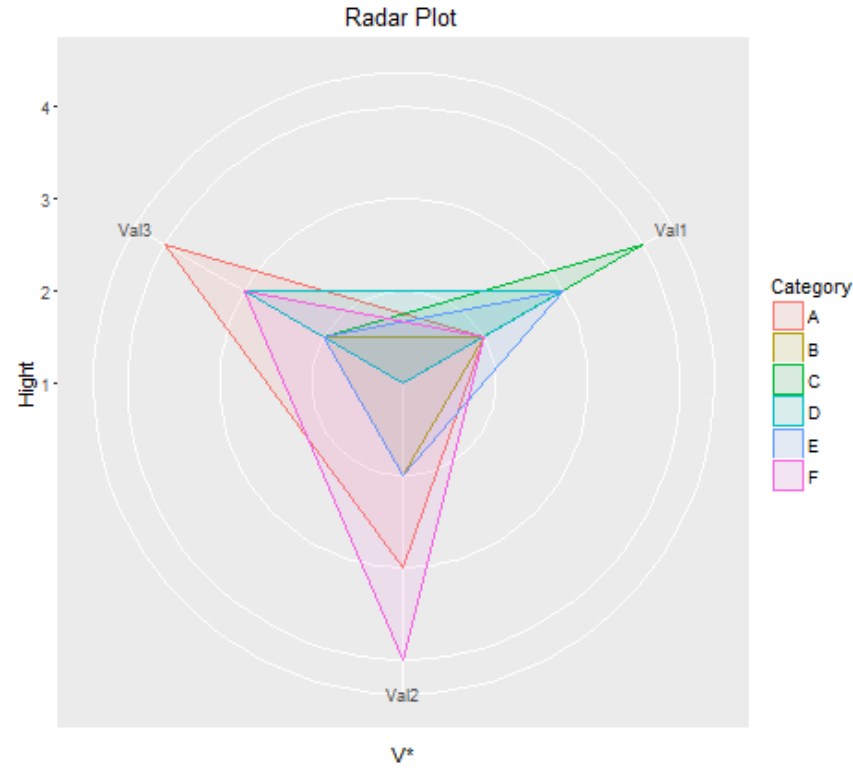
4.1 Radar Plot

Category	Val1	Val2	Val3
A	2	3	4
B	2	2	2
C	4	1	2
D	3	1	3
E	3	2	2
F	2	4	3

Table 4.1: Some Values

```
coord_radar <- function (theta = "x", start = 0, direction = 1)
{
  theta <- match.arg(theta, c("x", "y"))
  r <- if (theta == "x")
    "y"
  else "x"
  ggproto("CordRadar", CoordPolar, theta = theta, r = r, start = start,
    direction = sign(direction),
    is_linear = function(coord) TRUE)
}

ggplot(xs, aes(x=V, y=H, color=Category, group=Category, fill=Category)) +
geom_polygon(alpha=.1) +
coord_radar() +
ggtitle("Radar Plot") +
xlab("V*") + ylab("Hight")
```



5

Advanced Visualization

6

Quellen

6.1 *General*

- <http://www.cookbook-r.com/>
- <http://www.datendesign-r.de/beispiele/>
- <https://www.rstudio.com/resources/cheatsheets/>

6.2 *Special*

- <http://stackoverflow.com/questions/22064611/how-to-draw-rotated-axes-in-r>