Cheatsheet - Marketing Research

Not actually meant for cheating

About this document

- Some calculations are done in two equivalent ways. If so the line before them will include the comment
  ```
  # Equivalently:
  ```
- If you are not sure what a function does try `?functionname` e.g.:
  ```
  ?sum
  ```
- Required library
  - `tidyr`

Recognizing Scale Type of Data

1. Categorical
   - Nominal: Grouping variable that has **no ranking** -> blue eyes, brown eyes, ...

     blue eyes > brown eyes does **not** make sense
   - Ordinal: Grouping variable that has a **ranking** -> best grade, failing grade, ...

     best grade > failing grade makes sense
2. Continuous

   Values increase continuously (e.g decimals like 1.5, 3.14159 make sense)
   - Interval: There is no 0 point. Therefore, differences make sense but percentage differences do not -> year 2000, year 2002

     The difference between year 2000 and year 2002 is 2 years but since the starting point (from a mathematical standpoint!) is arbitrary year 0 is not the beginning of time. Thus, year 2002 is not x% greater than 2002.
   - Ratio: There is a 0 point. Percentage differences as well as absolute differences make sense -> 5 meters, 10 meters

     5 meters is 5 meters shorter than 10 meters **and** 10 meters is twice (200%) as long as 5 meters.

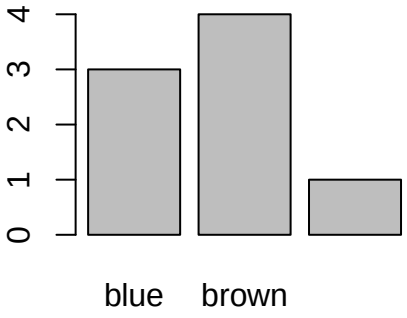
Visualization of each Scale Type

Categorical

Nominal

Bar-chart with counts of each category in no particular order. i.e.

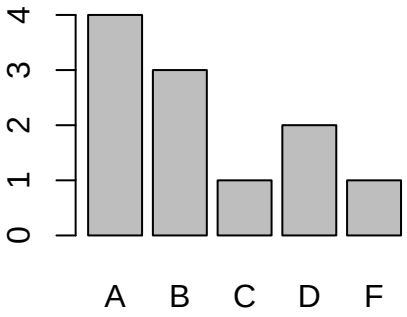
```
eyecolor <- c("brown", "green", "brown", "blue", "blue", "blue", "brown", "brown")
eyecolor_counts <- table(eyecolor)
barplot(eyecolor_counts)
```



Ordinal

Bar-chart with counts with x-axis following the ordering. i.e.

```
# American grading system (A is best, F is fail)
grades <- c("A", "A", "C", "D", "B", "A", "F", "B", "A", "D", "B")
grades <- sort(grades, decreasing = TRUE)
grades_count <- table(grades)
barplot(grades_count)
```

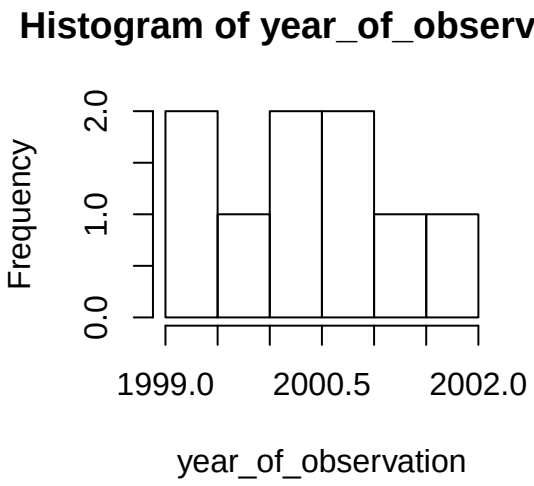


Continuous

Interval

Histogram of counts over a small range. i.e.

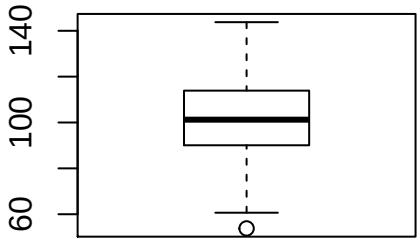
```
# e.g. .5 is second half
year_of_observation <- c(2000.5, 2000.5, 2001, 2002, 1999.5, 1999, 2000, 2001.5, 2001)
hist(year_of_observation)
```



Ratio

Boxplot, histogram or density plot of values. i.e.

```
set.seed(123)
sales <- rnorm(100, 100, 20)
boxplot(sales)
```

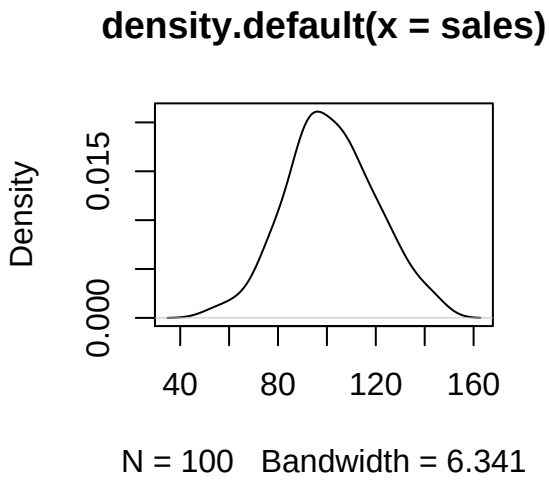



```
# There is one outlier that is more than 1.5 * interquartile-range away from the box:
any(sales<quantile(sales,0.25)-1.5*IQR(sales))
```

```
## [1] TRUE
```

```
hist(sales)
```


```
plot(density(sales))
```



Interpretation of the boxplot

The box shows the 25th percentile (i.e. the value for which 25% of the values in the data are lower and 75% are higher), the 50th percentile and the 75th percentile. By default (in R) the whiskers extend 1.5 times the interquartile range (see below) away from the box (i.e. below the 25th and above the 75th percentile).

Interpretation of density plot

see Interpretation of p-values

Measures of Location and Dispersion

Makes sense for continuous variables.

Location

- Mean

```
# Equivalently:
sum(sales)/length(sales)
```

```
## [1] 101.8081
```

```
mean(sales)
```

```
## [1] 101.8081
```

- Median

```
# Equivalently:
quantile(sales, 0.5)
```

```
##      50%
## 101.2351
```

```
median(sales)
```

```
## [1] 101.2351
```

if the distribution of the data is symmetric, like the normal distribution, mean and median are similar (theoretically the same). If you have some outliers that you would like to disregard in your research, the median might be better as it is the point where 50% of the values in the data are below and 50% above.

Dispersion

- Variance & Standard Deviation

```
var(sales)
```

```
## [1] 333.2931
```

```
# Equivalently:
sqrt(var(sales))
```

```
## [1] 18.25632
```

```
sd(sales)
```

```
## [1] 18.25632
```

- Interquartile Range

```
# Equivalently:
diff(quantile(sales, c(0.25, 0.75)))
```

```
##      75%
## 23.71347
```

```
IQR(sales)
```

```
## [1] 23.71347
```

For the same argument as above the IQR might be preferred if you'd like an estimate robust to outliers.

Why do we need Confidence Intervals?

- Read and understand slide 46 in the first slide set.
- Since we do not observe the entire population the parameter we estimate is just an approximation of the population parameter. Of course we would like to know the latter but we only observe our sample. We select a range around the sample parameter that is (theoretically, see central limit theorem) very likely to include the the true population parameter (i.e. for most of the samples we could get from the population the true parameter will lie within that range). That range is the confidence interval.

Interpretation of p-values

The p-value is the probability of observing a test-statistic (e.g. the t-statistic for the t-test or the F-statistic for the ANOVA) that is as large or larger (in absolute terms. i.e. we are interested in how far away we are from 0) as the one we get based on our sample given that the Null Hypothesis is true.

WHY? (Interpretation of area under the PDF) Think of all those density plots in the course. The area under the curve for a certain range on the x-axis represents the probability of observing a value in this range from the given distribution. For the H0 we always choose the "simpler" hypothesis (e.g. no difference; all equal; coefficient is 0). Since the H0 represents one value (e.g. 0) we can easily construct the theoretical distribution around that value given the H0 is true. Consider a t-test where the hypothesis is that the mean of sales is 100 (since I generated sales we know this to be true in this case). If we deduct our H0 from the mean of the data the result should be 0 if the H0 is true. Due to randomness in the data there is a small difference (recall: confidence intervals):

```
mean(sales) - 100 # pretty close
```

```
## [1] 1.808118
```

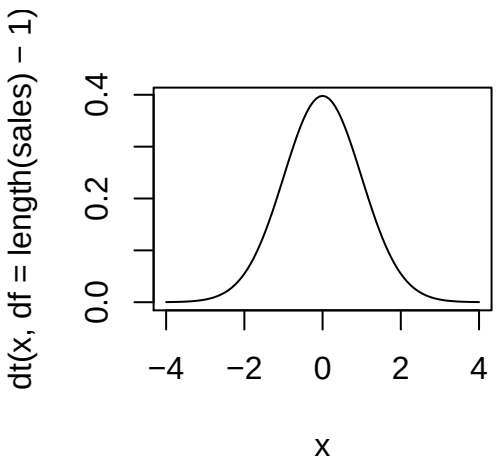
We also know that under repeated sampling the means of the samples would be normally distributed (have another look at central limit theorem if this is not clear). Since we do not know the population variance we have to use the t-distribution (thus t-test) after dividing by $\frac{s}{\sqrt{n}}$, the standard error of the mean. We observe that the mean in the data is approximately 1 standard error away from the H0 since the t-statistic is 0.99.

```
t.statistic <- (mean(sales) - 100) / (sd(sales)/ sqrt(length(sales)))
t.statistic
```

```
## [1] 0.9904068
```

We look at the distribution of t-statistics we could theoretically get (if we sample many times) from a population in which the H0 is in fact true (this is the "given the H0 is true" part). This is a distribution since we will get a slightly different mean for every sample rather than just a single value. It looks as follows:

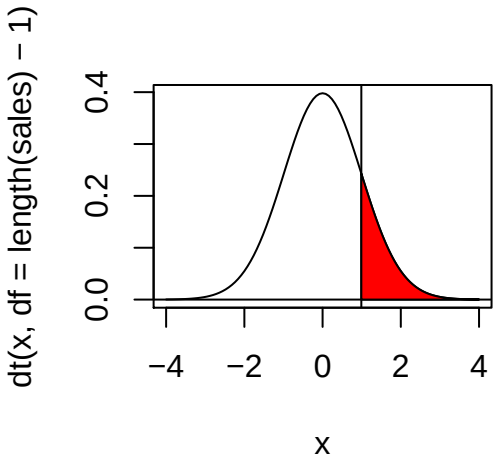
```
curve(dt(x, df = length(sales)-1), xlim = c(-4,4))
```



The interpretation of this plot is: "If the H0 is true we expect the t-statistics of most of the samples we could theoretically draw from the population to be between -2 and 2". This interpretation follows because the area under the curve for a certain interval represents the probability of observing a value in this range. The surface area between -2 and 2 is much larger than the rest.

Now let's look at our t-value (0.99) that represents one sample from this population. The question was how likely it is, given the H0 is true, to observe a value at least as far away from 0 or even farther away. Since the surface area represents probability we can use it to visualize the p-value (in red):

```
curve(dt(x, df = length(sales)-1), xlim = c(-4,4))
abline(v = t.statistic)
abline(h = 0)
polygon(c(t.statistic, seq(t.statistic, 4, 0.01), 4), c(0, dt(seq(t.statistic, 4, 0.01), df = length(sales
```



The area can be calculated using the cumulative distribution function (CDF) of the corresponding distribution (notice that we have to multiply by 2 for a two-sided test):

```
# One-sided test
p.value <- 1 - pt(t.statistic, df = length(sales)-1)
p.value
```

```
## [1] 0.1621949
```

```
t.test(sales, mu = 100, alternative = "greater")
```

```
##
##  One Sample t-test
##
## data:  sales
## t = 0.99041, df = 99, p-value = 0.1622
```

```
## alternative hypothesis: true mean is greater than 100
## 95 percent confidence interval:
##  98.77686      Inf
## sample estimates:
## mean of x
##  101.8081
```

```
# Two-sided test
p.value <- 2*(1 - pt(t.statistic, df = length(sales)-1))
p.value
```

```
## [1] 0.3243898
```

```
t.test(sales, mu = 100)
```

```
##
##  One Sample t-test
##
## data:  sales
## t = 0.99041, df = 99, p-value = 0.3244
## alternative hypothesis: true mean is not equal to 100
## 95 percent confidence interval:
##   98.18567 105.43057
## sample estimates:
## mean of x
##  101.8081
```

Error Types in Hypothesis testing

Notice that in a real life setting we do not observe the true population parameter so we never know if we made an error based on our sample.

Type I error

In the population the H0 (Null Hypothesis) is in fact true but our sample leads us to rejecting it. i.e.

H0: Mean of sales is equal to 100 (this is how I generated the sales data so we know the population in this case)

H1: Mean of sales is not equal to 100

Here we reject the H0 for the given sample if we choose a 95% confidence interval since the p-value is less than 0.05. If we choose a 99% confidence interval, we would not reject the H0 since the p-value is greater than 0.01. So a higher/wider confidence interval (equivalently lower α) leads to less Type I errors.

```
set.seed(22)
sales_sample <- sample(sales, 10)
# 100 is the population mean!
t.test(sales_sample, mu = 100)
```

```
##
##  One Sample t-test
##
## data:  sales_sample
## t = 2.523, df = 9, p-value = 0.03261
## alternative hypothesis: true mean is not equal to 100
## 95 percent confidence interval:
##  101.0867 119.9332
## sample estimates:
```

```
## mean of x
##    110.51
```

Type II error

In the population the H0 is in fact false but our sample leads us to not rejecting it.

H0: Mean is equal to 120

H1: Mean is not equal to 120

We do not reject the H0 that the population mean is equal to 120 if we choose a 95% confidence interval. If we choose a 90% confidence interval, we would reject the H0 since the p-value is less than 0.10. So a lower/shorter confidence interval (equivalently higher α) leads to less Type II errors but more Type I errors.

```
set.seed(1)
sales_sample <- sample(sales, 10)
# 100 is the population mean!
t.test(sales_sample, mu = 120)
```

```
##
##  One Sample t-test
##
## data:  sales_sample
## t = -1.8958, df = 9, p-value = 0.09049
## alternative hypothesis: true mean is not equal to 120
## 95 percent confidence interval:
##   90.87716 122.56564
## sample estimates:
## mean of x
##  106.7214
```

Formulating the Hypotheses (H0 and H1)

The H0 is always the simpler hypothesis in the sense that it is represented by a single value. In addition, we can never accept the H0. Therefore, it is usually a hypothesis we would like to reject to have an interesting finding in our research. Let's have a look at the hypotheses of the tests we have discussed:

t-test

- one sample

Use for: Continuous variable without grouping. Interest in difference to H0.

```
H0: mean is equal to specified value e.g. 100

H1: mean is not equal to specified value e.g. not 100

The H0 represents a single value, e.g. 100, whereas the H1 represents all other (possible) values
```

```
t.test(sales, mu = 100)
```

```
##
##  One Sample t-test
##
## data:  sales
## t = 0.99041, df = 99, p-value = 0.3244
```

```
## alternative hypothesis: true mean is not equal to 100
## 95 percent confidence interval:
##   98.18567 105.43057
## sample estimates:
## mean of x
##  101.8081
```

- two sample

Use for: Continuous variable observed for two groups. Interest in difference.

H0: mean of two samples are the same -> difference in mean is 0

H1: mean of two samples are not the same -> difference in mean is not 0

```
sales2 <- rnorm(100, 100, 20)
t.test(sales, sales2)
```

```
##
##  Welch Two Sample t-test
##
## data:  sales and sales2
## t = 0.06152, df = 197.86, p-value = 0.951
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -4.869043  5.182615
## sample estimates:
## mean of x mean of y
##  101.8081  101.6513
```

ANOVA

Use for: Continuous variable observed for two or more groups. Interest in difference.

H0: mean of all categories are the same

H1: at least one of the means is different

```
sales_data <- data.frame(store1 = sales, store2 = sales2, store3 = rnorm(100, 100, 20))
sales_data <- tidyr::gather(sales_data, key = store, value = sales)
head(sales_data) # first 6 lines in df
```

```
##    store     sales
## 1 store1  88.79049
## 2 store1  95.39645
## 3 store1 131.17417
## 4 store1 101.41017
## 5 store1 102.58575
## 6 store1 134.30130
```

```
tail(sales_data) # last 6 lines in df
```

```
##      store     sales
## 295 store3  92.37848
## 296 store3 108.18804
## 297 store3 133.77747
## 298 store3 131.73177
## 299 store3  93.38184
## 300 store3  54.29529
```

```r
# All have same mean:
summary(aov(sales~store, data = sales_data))
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## store         2    236   117.8   0.335  0.715
## Residuals   297 104371   351.4
```

```r
# Change one store to have a different mean:
sales_data$sales[sales_data$store == "store3"] <- sales_data$sales[sales_data$store == "store3"] - 50
summary(aov(sales~store, data = sales_data))
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## store         2 179402   89701   255.3 <2e-16 ***
## Residuals   297 104371     351
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

χ^2-test

- For equal proportions

Use for: Single categorical variable

H0: Counts of each category are equal, e.g. all eye colors occur equally frequently

H1: At least one category has a different count, e.g. at least one eye color occurs more or less frequently

```r
chisq.test(eyecolor_counts)
```

```
## Warning in chisq.test(eyecolor_counts): Chi-squared approximation may be
## incorrect
```

```
##
##  Chi-squared test for given probabilities
##
## data:  eyecolor_counts
## X-squared = 1.75, df = 2, p-value = 0.4169
```

- For independence

Use for: Two categorical variables

H0: There is no relationship between two categorical variables, e.g. eye color is independent of biological sex

H1: There is a relationship between two categorical variables, e.g. eye color depends on biological sex, i.e. is not independent

```r
set.seed(1)
female_eyecolor <- sample(eyecolor, size = length(eyecolor), replace = TRUE)
male_eyecolor <- sample(eyecolor, size = length(eyecolor), replace = TRUE)


eyecolor_data <- data.frame(f = female_eyecolor, m = male_eyecolor)
eyecolor_data <- tidyr::gather(eyecolor_data, key = 'sex', value = 'eyecolor')

chisq.test(eyecolor_data$sex, eyecolor_data$eyecolor)
```

```
## Warning in chisq.test(eyecolor_data$sex, eyecolor_data$eyecolor): Chi-
## squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
```

```
##
## data:  eyecolor_data$sex and eyecolor_data$eyecolor
## X-squared = 2.2857, df = 2, p-value = 0.3189
```

Regression

Use for: Continuous dependent variable. Interest in relationship to independent variables.

H0: Regression coefficient (β_k) is equal to 0

H1: Regression coefficient is not equal to 0

Notice that we test this hypothesis for each of the coefficients with a t-statistic similar to the t-test.

```
set.seed(1)
advertisement_spending <- rnorm(100, 50, 20)
# True intercept is 20, true beta1 is 5
sales <- 20 + 5 * advertisement_spending + rnorm(100, 0, 15)

summary(ols <- lm(sales~advertisement_spending))
```

```
##
## Call:
## lm(formula = sales ~ advertisement_spending)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.153  -9.206  -2.092   8.091  35.193
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              19.4744     4.4562    4.37 3.09e-05 ***
## advertisement_spending   4.9992     0.0808   61.88  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.44 on 98 degrees of freedom
## Multiple R-squared:  0.975,  Adjusted R-squared:  0.9748
## F-statistic:  3829 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
# R-squared and adj. R-squared calculation
(R2 <- sum((fitted(ols) - mean(sales))^2)  / sum((sales - mean(sales))^2))
```

```
## [1] 0.9750415
```

```
adjR2 <- 1 - ( (1-R2) * (length(sales) - 1) ) / (length(sales) - length(coef(ols)))
adjR2
```

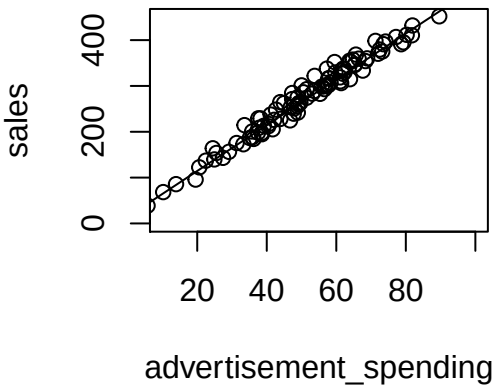
```
## [1] 0.9747868
```

Interpretation of coefficients

If we do not spend anything on advertisement we expect, on average, sales of 19.474 (this is the intercept). For each additional Euro spent on advertisements we expect an increase in sales of 4.999 in sales (this is the slope parameter β). Both coefficients are significantly different from 0.

This can be visualized as follows:

```
plot(advertisement_spending, sales, xlim = c(10, 100), ylim = c(0, 450))
lines(predict(ols, data.frame(advertisement_spending = 0:99)))
```



If we add more coefficients the interpretation of each single coefficient becomes conditional on keeping all other variables constant ("ceteris paribus").

```
price <- rgamma(100, 10, 1)
sales <- 20 + 5 * advertisement_spending - 3 * price + rnorm(100, 0, 15)
summary(mols <- lm(sales ~ advertisement_spending + price))
```

```
##
## Call:
## lm(formula = sales ~ advertisement_spending + price)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -30.858  -8.302   0.151   8.555  47.739
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)             21.55816    6.28681   3.429 0.000891 ***
## advertisement_spending   5.06330    0.07976  63.485  < 2e-16 ***
## price                   -3.42265    0.42416  -8.069 1.91e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.24 on 97 degrees of freedom
## Multiple R-squared:  0.9772, Adjusted R-squared:  0.9768
## F-statistic:  2081 on 2 and 97 DF,  p-value: < 2.2e-16
```

e.g. Keeping advertisement spending constant we expect sales to decrease by -3.423. Keeping the price constant sales will increase, on average, by 5.063. The interpretation of the intercept probably does not make sense for this regression. It is the sales if advertisement spending and price are equal to 0. However, it needs to be used for predictions

Forecasting / Calculation of Expectations

Essentially plug values for the data into the regression equation:

E.g. - First Regression: How much sales revenue would we expect if the store spends 50€ on advertisements:

```
# Equivalently:
coef(ols)['(Intercept)'] + 50 * coef(ols)['advertisement_spending']
```

```
## (Intercept)
##    269.4346
```

```
predict(ols, data.frame(advertisement_spending = 50))
```

```
##        1
```

```
## 269.4346
```

- Second regression: How much sales revenue would we expect if the store spends 30€ on advertisements and sets a price of 20€:

```
# Equivalently
coef(mols)['(Intercept)'] + 30 * coef(mols)['advertisement_spending'] + 20 * coef(mols)['price']
```

```
## (Intercept)
##    105.0042
```

```
predict(mols, data.frame(advertisement_spending = 30, price = 20))
```

```
##        1
## 105.0042
```

Correlation & Covariance

Correlation

Use for: Linear relationship between 2 continuous variables

- Coefficient always between -1 (perfect negative relationship) and 1 (perfect positive relationship)
- Coefficient of 0 indicates no relationship.
- Causal relationship needs to be established theoretically:
  - Chicken - egg problem (A causes B or B causes A?)
  - No causation at all possible e.g. two values independently grow over time
  - Third variable causes both variables for which we measure correlation -> high correlation without causality

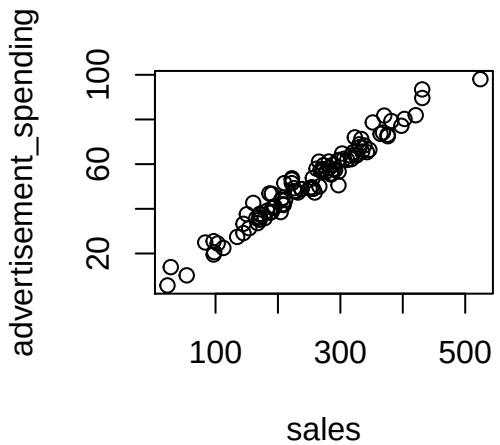
```
# Positive linear relationship
# Equivalently:
cor(sales, advertisement_spending)
```

```
## [1] 0.9807828
```

```
# standardized covariance thus between -1 and 1
cov(sales, advertisement_spending) /
  (sd(sales) * sd(advertisement_spending))
```

```
## [1] 0.9807828
```

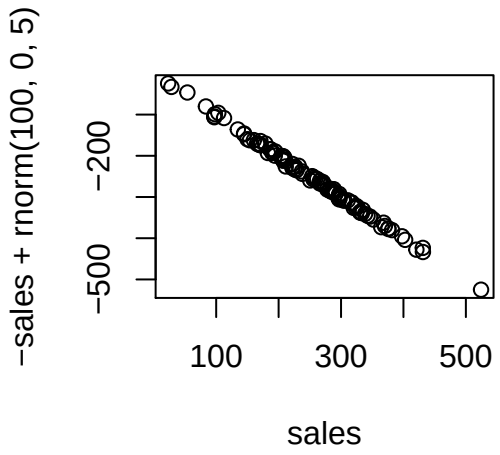
```
plot(sales, advertisement_spending)
```




```
# Negative linear Relationship
cor(sales, -sales + rnorm(100, 0, 5))
```

```
## [1] -0.9982537
```

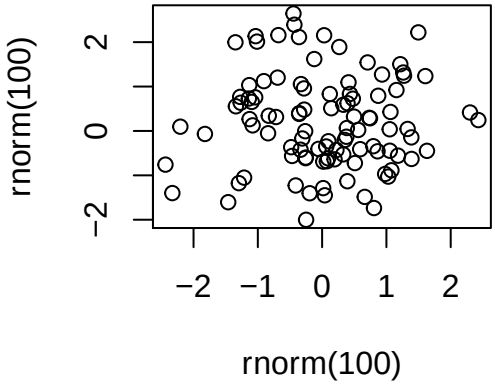
```
plot(sales, -sales + rnorm(100, 0, 5))
```




```
# No linear relationship
set.seed(10)
cor(rnorm(100), rnorm(100))
```

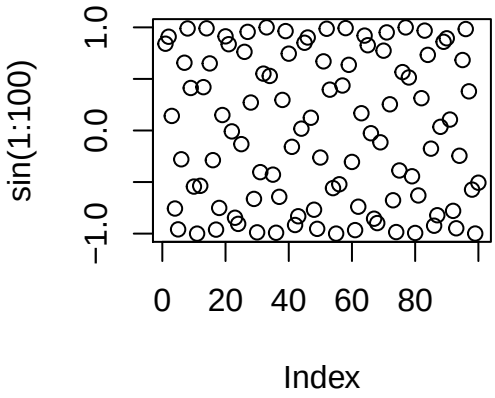
```
## [1] -0.05803451
```

```
plot(rnorm(100), rnorm(100))
```

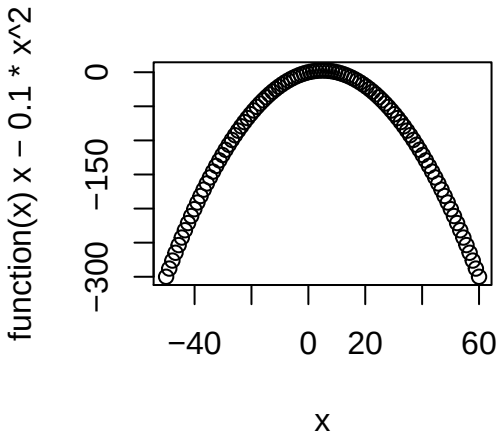


- Examples of **nonlinear** relationships

```
plot(sin(1:100))
```




```
plot(function(x) x - 0.1 * x^2, xlim = c(-50, 60), type = 'p')
```



Covariance

- Unstandardized correlation
- Same sign but different magnitude
- Value depends on variance of the data and is thus not meaningful by itself

```
# Equivalently:
sum((sales - mean(sales)) * (advertisement_spending - mean(advertisement_spending))) /
  (length(sales) - 1)
```

```
## [1] 1645.014
```

```
cov(sales, advertisement_spending)
```

```
## [1] 1645.014
```

- Notice relationship with variance

```
cov(sales, sales)
```

```
## [1] 8717.427
```

```
var(sales)
```

```
## [1] 8717.427
```