(Elementary) Regression Methods & Computational Statistics (405.952)

Part II: Regression

Assoz.Prof. Dr. Wolfgang Trutschnig
Arbeitsgruppe Stochastik/Statistik
Fachbereich Mathematik
Universität Salzburg
www.trutschnig.net

Salzburg, October-November 2018

Quick Reminder: Pearson correlation coefficient ρ

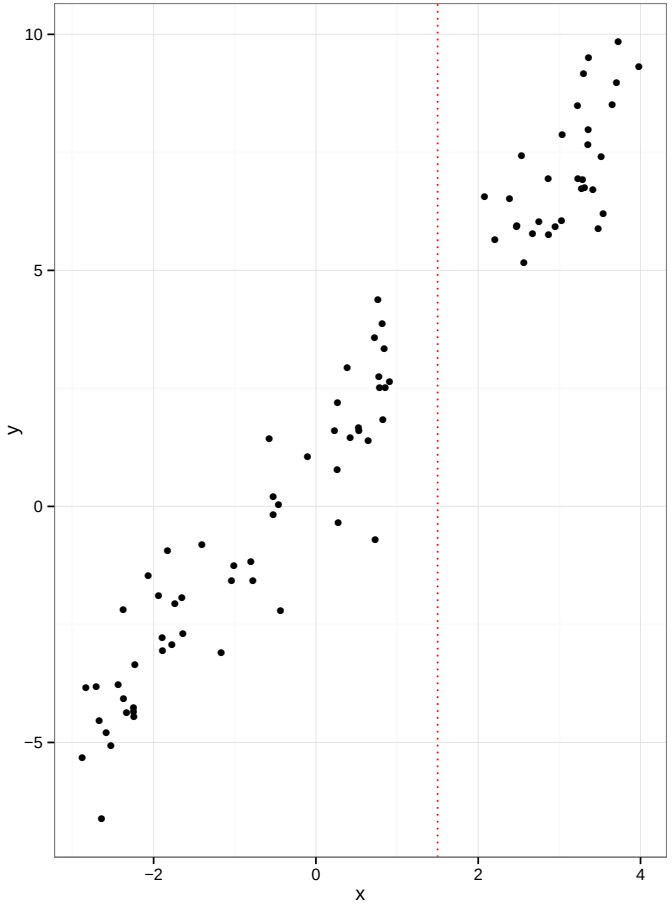


Figure: What is the correlation coefficient of the drawn sample?

- The graphic depicts a sample $(x_1, y_1), \ldots, (x_n, y_n)$.
- Give a rough estimate of the correlation coefficient ρ of the sample
- How can ρ be calculated?
- Let s_x (resp. s_y) denote the standard deviation of the x-coordinates (y-coordinates) of the sample, i.e.

$$s_x = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \overline{x}_n)^2}$$

$$s_y = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \overline{y}_n)^2}$$

Quick Reminder: Pearson correlation coefficient ρ

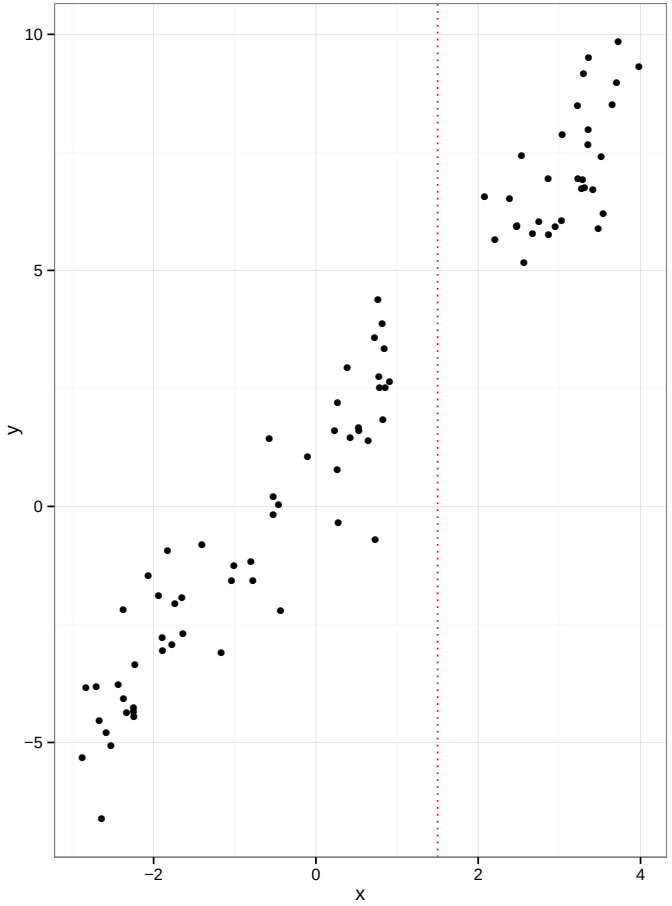


Figure: What is the correlation coefficient of the drawn sample?

- Let s_{xy} denote the (empirical) covariance of the sample, i.e.

$$s_{xy} = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \overline{x}_n)(y_i - \overline{y}_n)$$

- The (Pearson) correlation coefficient ρ_{xy} is defined as

$$\rho_{xy} = \frac{s_{xy}}{s_x s_y}$$

if $s_x, s_y > 0$.

- In our case we get $\rho_{xy} = 0.97464$
- How can this value be interpreted?

Properties of ρ

- Whenever ρ_{xy} exists (i.e. whenever $s_x, s_y > 0$) we have $-1 \leq \rho_{xy} \leq 1$
- We have $\rho_{xy} = \rho_{yx}$. As a consequence we will simply write ρ in the sequel
- $\rho = 1$ if and only if $(x_1, y_1), \ldots, (x_n, y_n)$ lie on a straight line with positive slope
- $\rho = -1$ if and only if $(x_1, y_1), \ldots, (x_n, y_n)$ lie on a straight line with negative slope
- In case of $\rho = 0$ we call the sample $(x_1, y_1), \ldots, (x_n, y_n)$ uncorrelated
- $\rho = 0$ is not a measure of dependence - it only measures *linear dependence*
- $\rho = 0$ means that there is no linear dependence
- If instead of $(x_1, y_1), \ldots, (x_n, y_n)$ we consider $(2x_1, 3y_1), \ldots, (2x_n, 3y_n)$, what happens to ρ?
- If instead of $(x_1, y_1), \ldots, (x_n, y_n)$ we consider $(-2x_1, -3y_1), \ldots, (-2x_n, -3y_n)$, what happens to ρ?

Quick Reminder: Pearson correlation coefficient ρ

- If instead of $(x_1, y_1), \ldots, (x_n, y_n)$ we consider $(-2x_1, -3y_1), \ldots, (-2x_n, -3y_n)$, what happens to ρ?

```
file <- url("http://www.trutschnig.net/geo_reg1.RData")
load(file)
A<-geo_reg1
head(geo_reg1)

cor(A$x,A$y)
cor(2*A$x,3*A$y)
cor(-2*A$x,-3*A$y)
cor(-2*A$x,3*A$y)
```

- ρ does not change under *linear* transformations with the same sign
- ρ changes, however, under non-linear transformations:
- If instead of $(x_1, y_1), \ldots, (x_n, y_n)$ we consider $(x_1^3, y_1^3), \ldots, (x_n^3, y_n^3)$ then we get $\rho = 0.9$

Spearman rank correlation ρ_S

- Assume we want to have a measure quantifying if there is a monotonic relationship between the x- and the y-coordinates of a sample $(x_1, y_1), \ldots, (x_n, y_n)$
- 'Monotonic relationship' (or concordance) in the sense that if the x-coordinates increase then also the y-coordinates (grow or fall together).
- There is no need for the relationship to be linear
- One natural idea is to work with ranks - best explained by some simple examples:

```
x1 <- c(3, 1, 4, 15, 13)
r1 <- rank(x1)
x1
#[1] 3  1  4 15 13
r1
#[1] 2 1 3 5 4
```

Spearman rank correlation ρ_S

```
x1 <- c(3, 1, 3, 15, 13)
r1 <- rank(x1)
x1
#[1] 3  1  3 15 13
r1
#[1] 2.5 1.0 2.5 5.0 4.0
```

- The values are sorted - the rank $rk(x_i)$ of observation x_i is the position after the ranking
- In case of ties averages of the ranks will be calculated (other choices are optional in the function)
- From $(x_1, y_1), \ldots, (x_n, y_n)$ we get the sample ranks $(rk_x(x_1), rk_y(y_1)), \ldots, (rk_x(x_n), rk_y(y_n))$
- $rk_x(x_i)$ is the rank of observation x_i among $x_1, \ldots, x_n$
- $rk_y(y_i)$ is the rank of observation y_i among $y_1, \ldots, y_n$
- The Spearman rank correlation is defined as the Pearson correlation of these ranks

Example

- Considering the following sample of size $n = 5$

x	y	rk.x	rk.y
3.05	10.21	2.00	2.00
1.38	2.19	1.00	1.00
4.32	19.31	3.00	3.00
15.51	241.08	5.00	5.00
7.08	50.81	4.00	4.00

- What can be seen?
- For ρ_S we get $\rho_s = 1$

```
cor(rank(E$x),rank(E$y))
cor(E$x,E$y,method="spearman")
```

Properties of ρ_S:

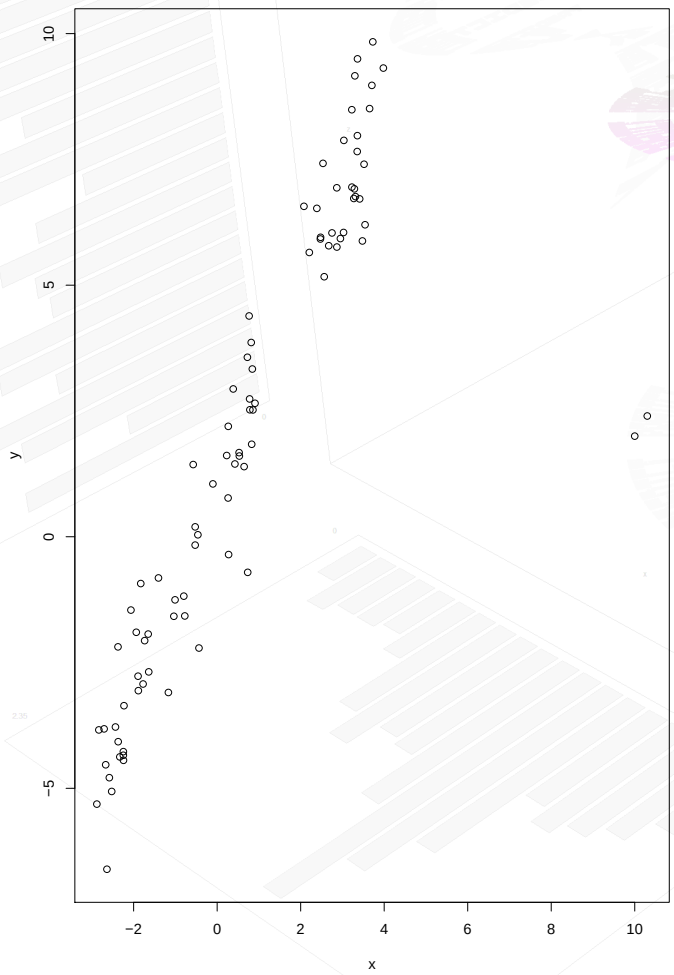
- Whenever ρ_S exists we have $-1 \leq \rho_{xy} \leq 1$
- ρ_S is symmetric too
- $\rho_S = 1$ if and only if for each pair $(x_i, y_i), (x_j, y_j)$ we have $x_i \leq x_j$ if and only if $y_i \leq y_j$
- $\rho_S = -1$ if and only if for each pair $(x_i, y_i), (x_j, y_j)$ we have $x_i \leq x_j$ if and only if $y_i \geq y_j$
- $\rho_S = 0$ is not a measure of dependence - it only measures *monotonic dependence*
- $\rho_S = 0$ means that there is no monotonic relationship dependence
- If instead of $(x_1, y_1), \ldots, (x_n, y_n)$ we consider $(2x_1, 3y_1), \ldots, (2x_n, 3y_n)$, what happens to ρ_S?
- If instead of $(x_1, y_1), \ldots, (x_n, y_n)$ we consider $(-2x_1, -3y_1), \ldots, (-2x_n, -3y_n)$, what happens to ρ_S?
- If instead of $(x_1, y_1), \ldots, (x_n, y_n)$ we consider $(x_1^3, y_1^3), \ldots, (x_n^3, y_n^3)$, what happens to ρ_S?

Spearman rank correlation ρ_S

```
file <- url("http://www.trutschnig.net/geo_reg1.RData")
load(file)
A<-geo_reg1
head(geo_reg1)

cor(A$x,A$y,method = "spearman")
cor(2*A$x,3*A$y,method = "spearman")
cor(-2*A$x,-3*A$y,method = "spearman")
cor(A$x^3,A$y^3,method = "spearman")
```

- For all four cases we get $\rho_S = 0.9633945$
- Easy to verify: ρ_S is invariant under monotonic transformations (both increasing or both decreasing)
- Let's add two outliers to A and see how ρ and ρ_S change

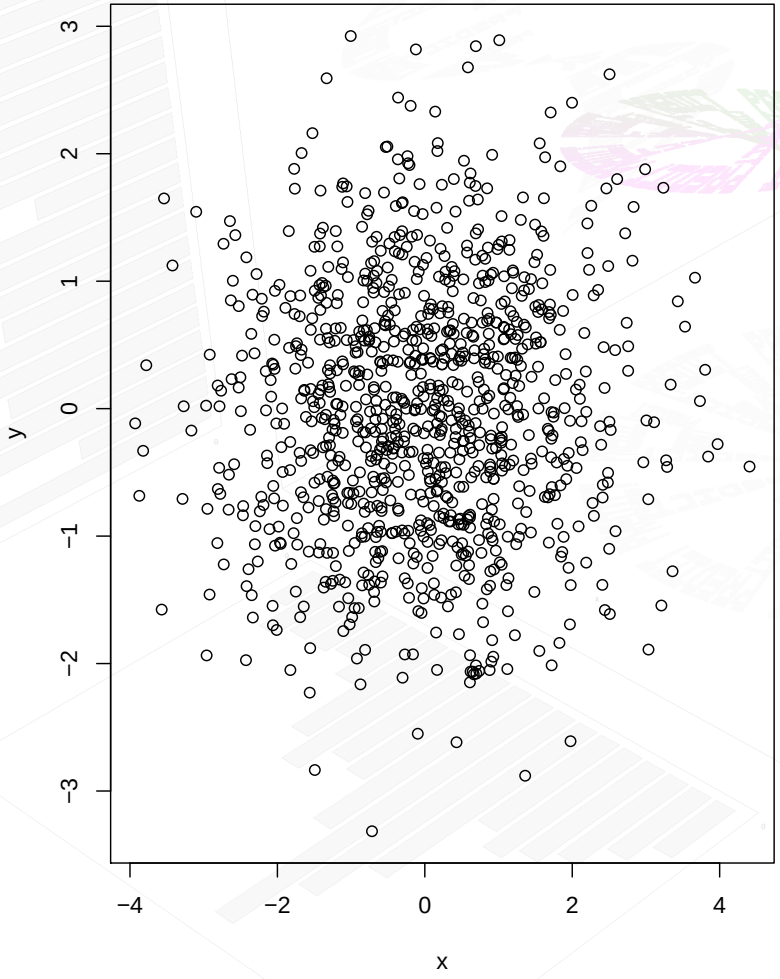
Spearman rank correlation ρ_S

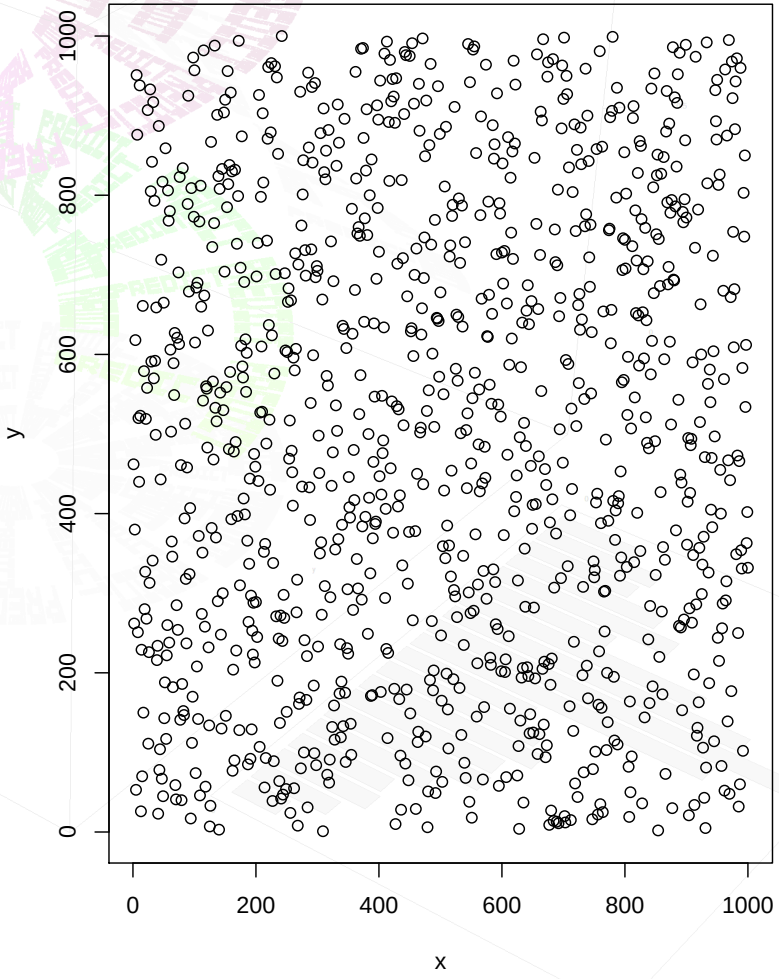



```
Dazu<-data.frame(x=c(10,10.3),y=c
    (2,2.4))
A1<-rbind(A,Dazu)
plot(A1)
cor(A1$x,A1$y)
cor(A1$x,A1$y,method = "spearman")
```

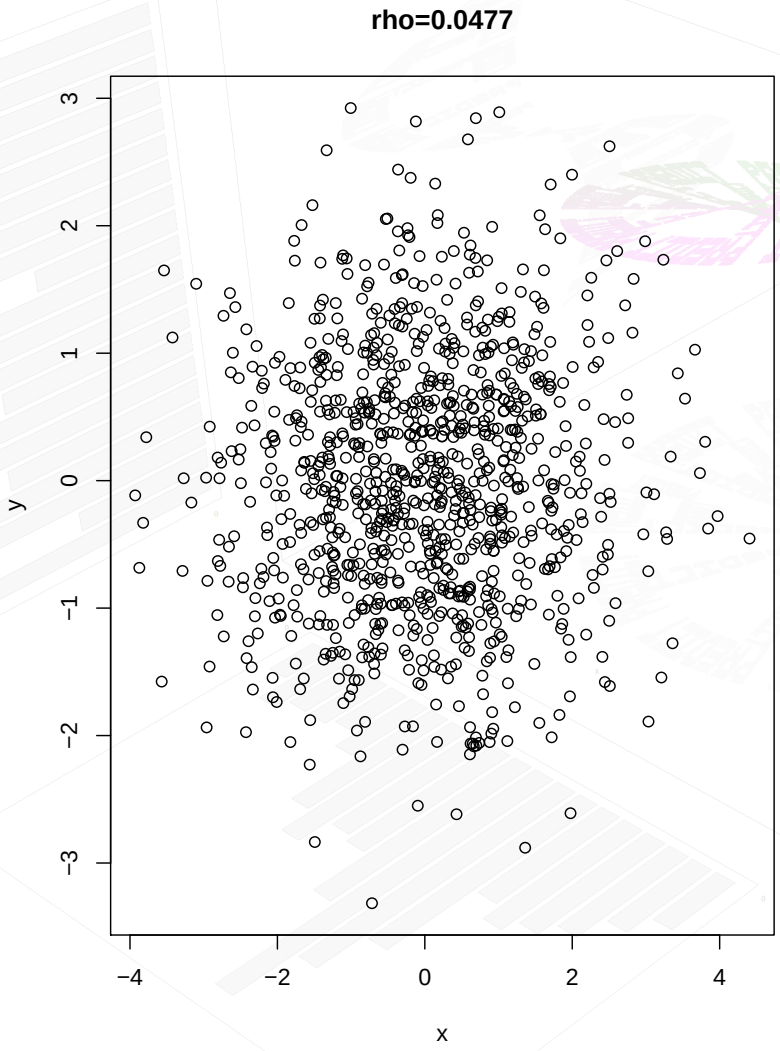
- Which is more influenced by the two new points?
- We get $\rho = 0.8187617$ (before $\rho = 0.97464$)
- Moreover $\rho_S = 0.9349794$ (before $\rho_S = 0.9633945$)
- ρ is less robust against outliers than ρ_S
- Rank-based quantities are generally robust

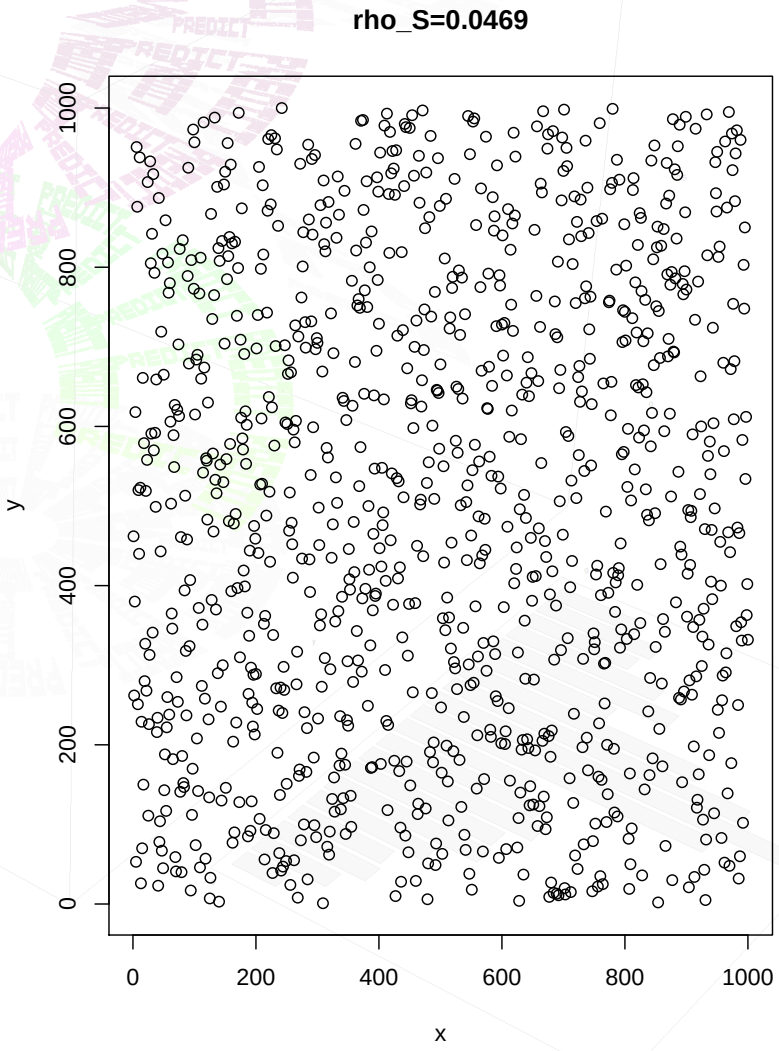
Some examples and exercises



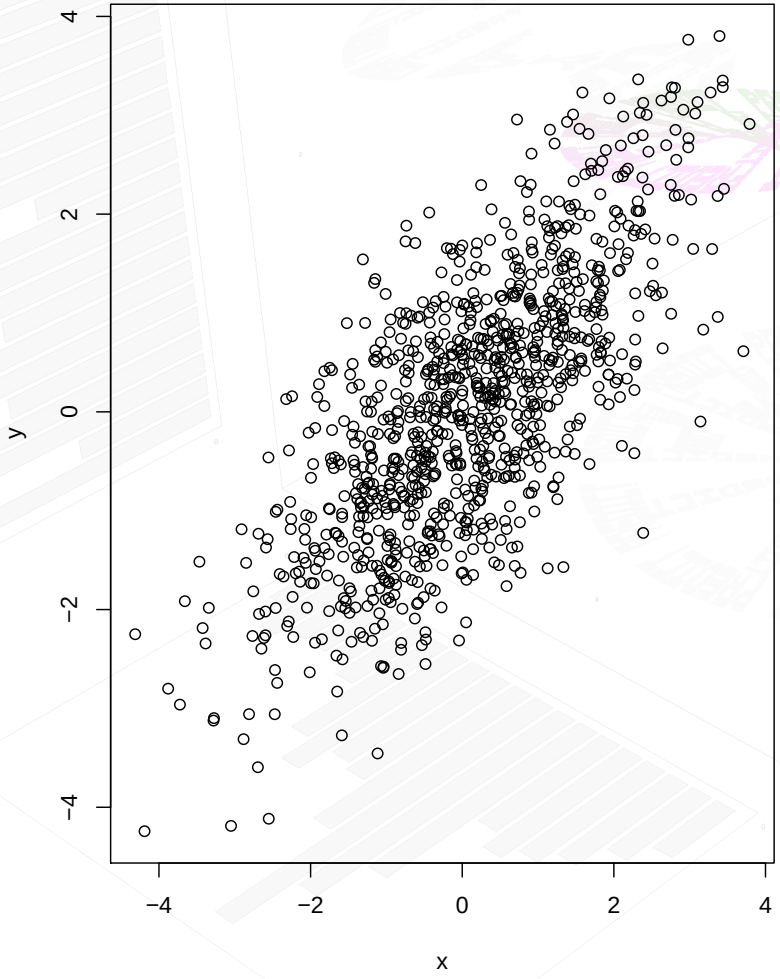


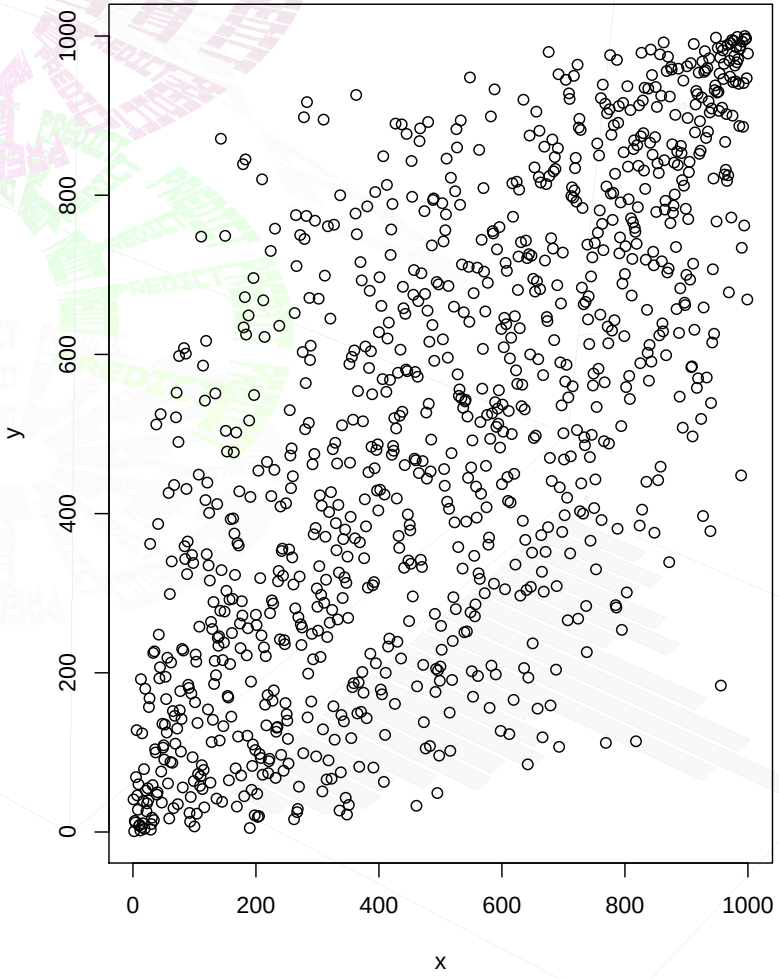
Some examples and exercises



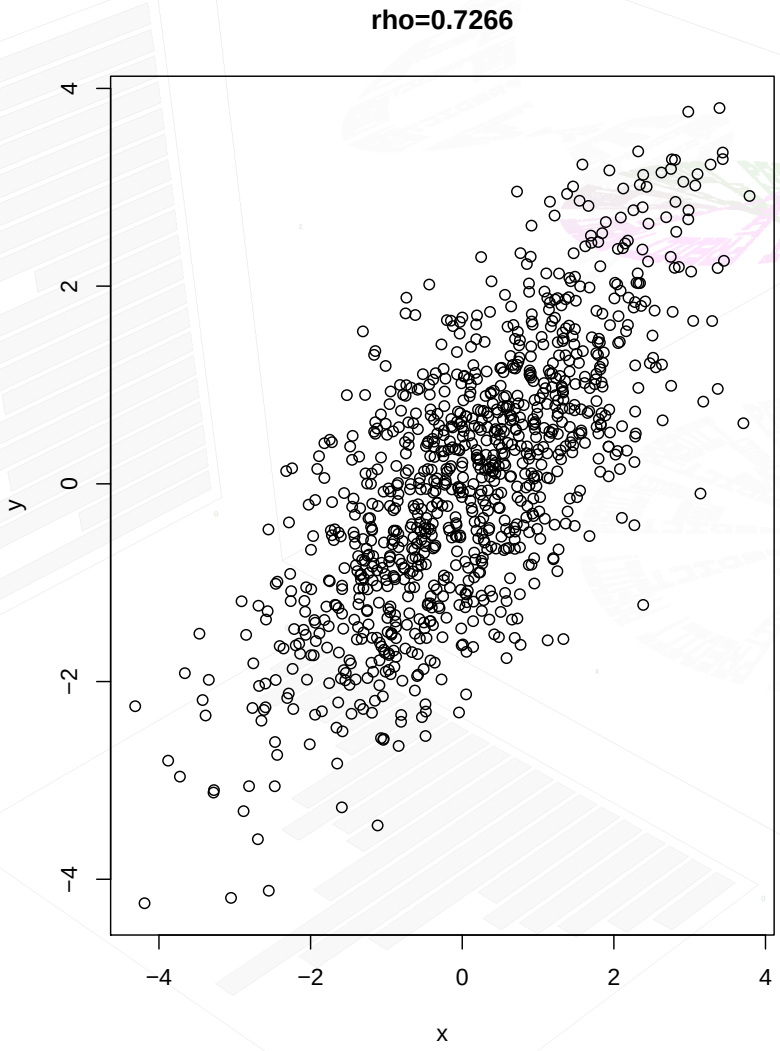


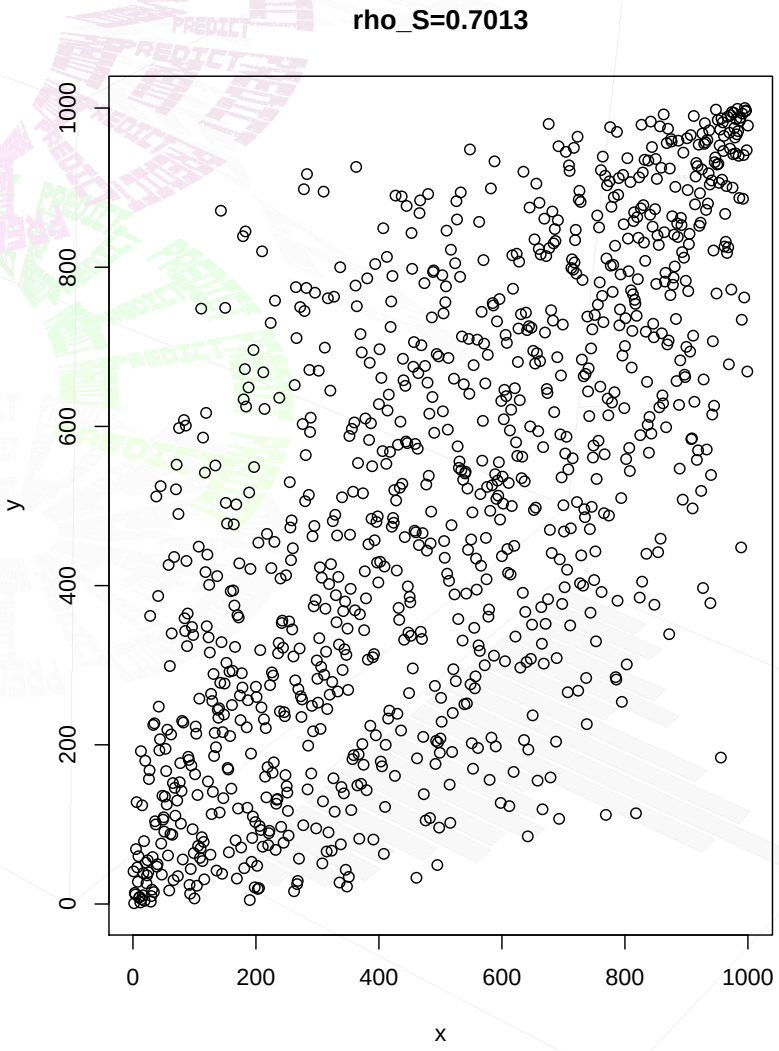
Some examples and exercises



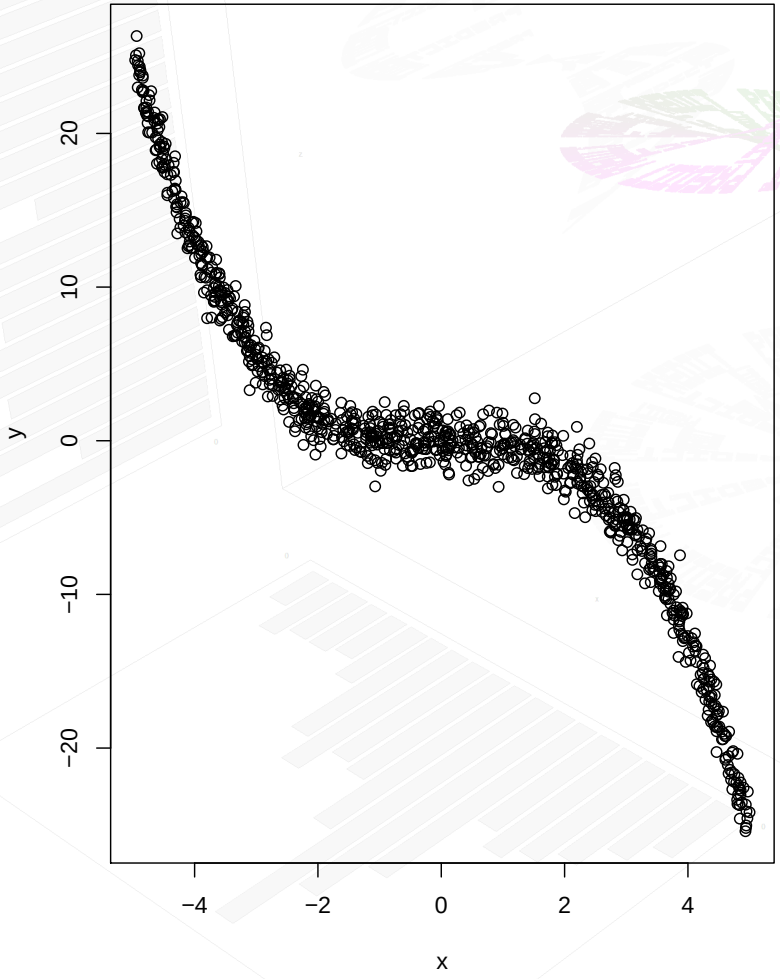


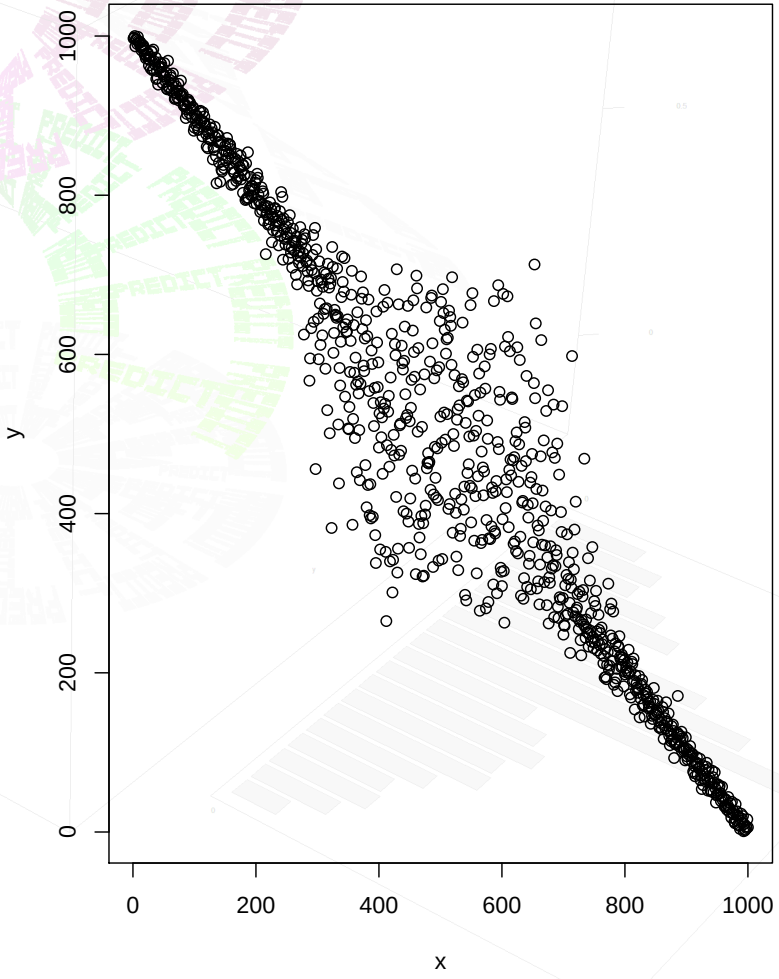
Some examples and exercises



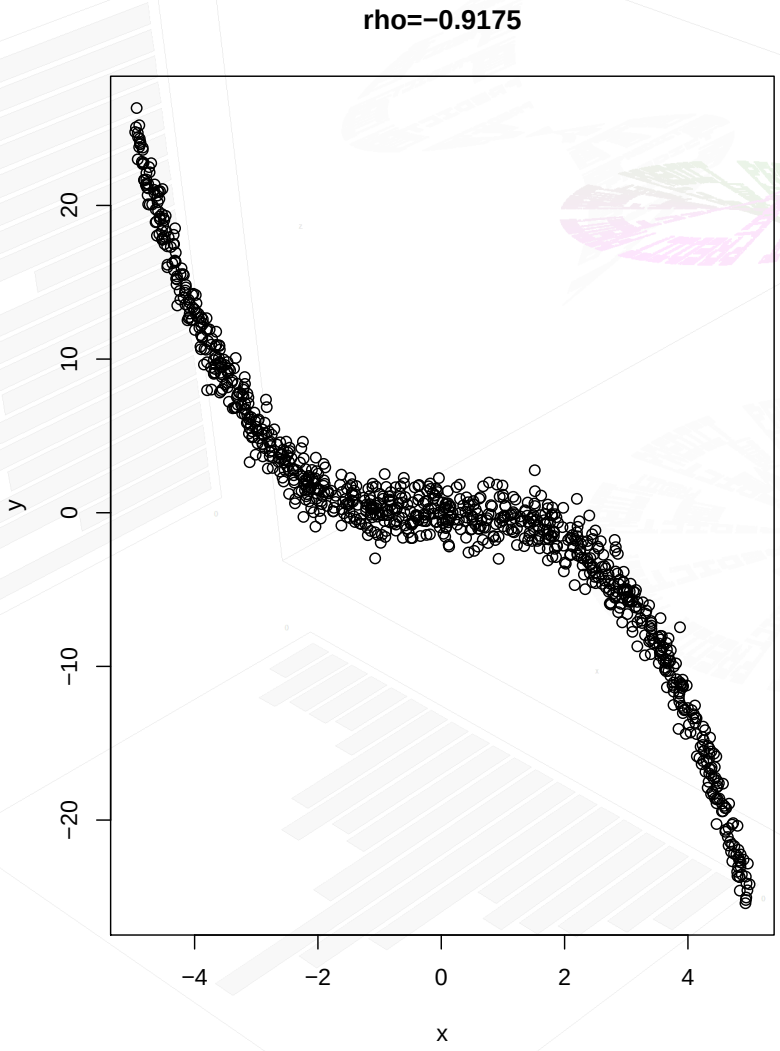


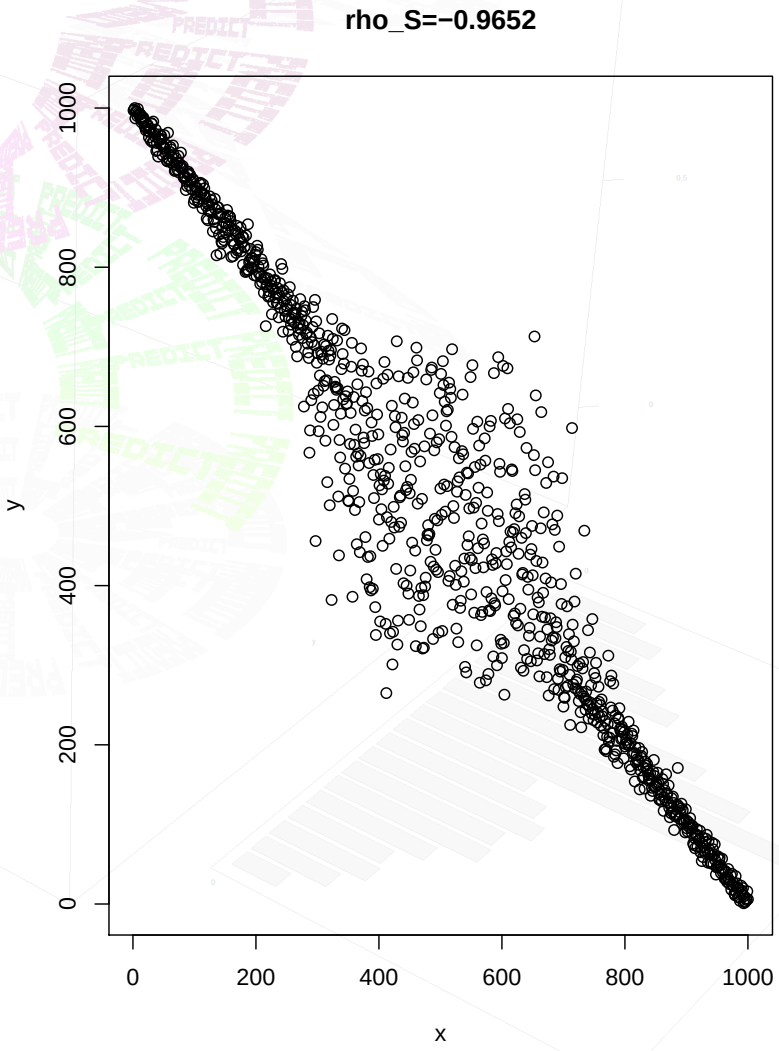
Some examples and exercises



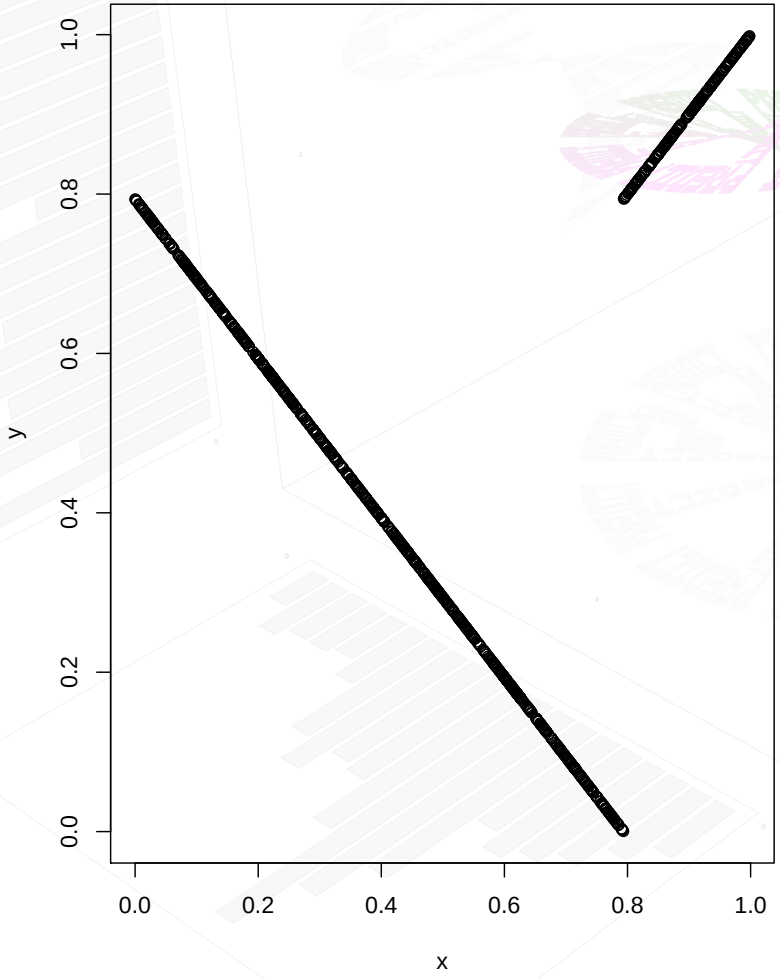


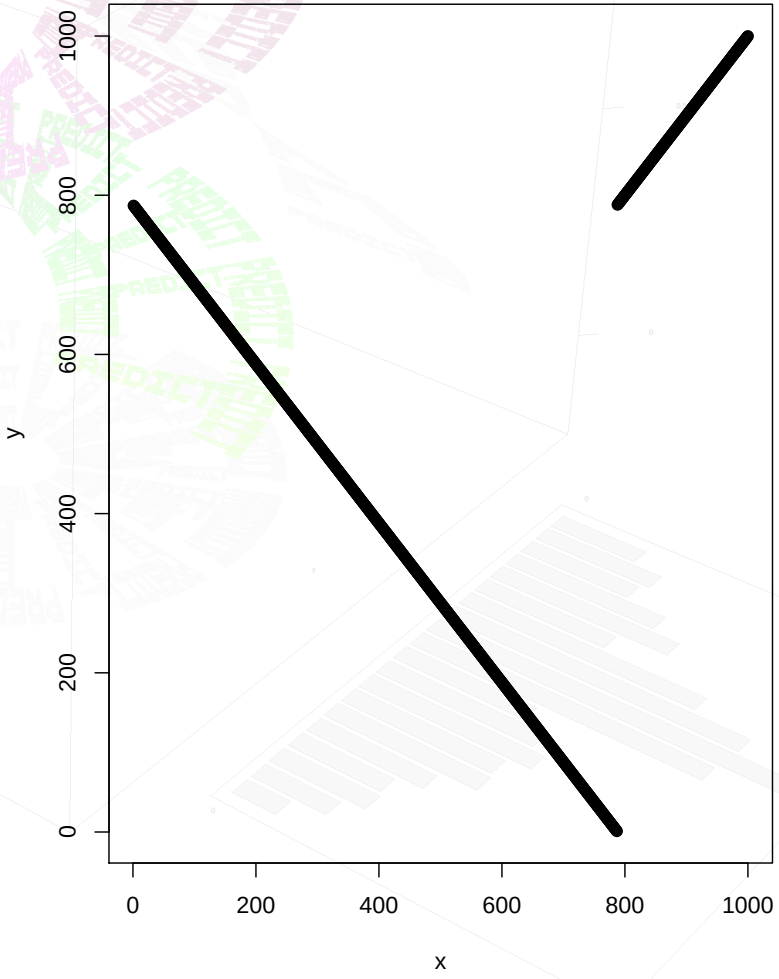
Some examples and exercises



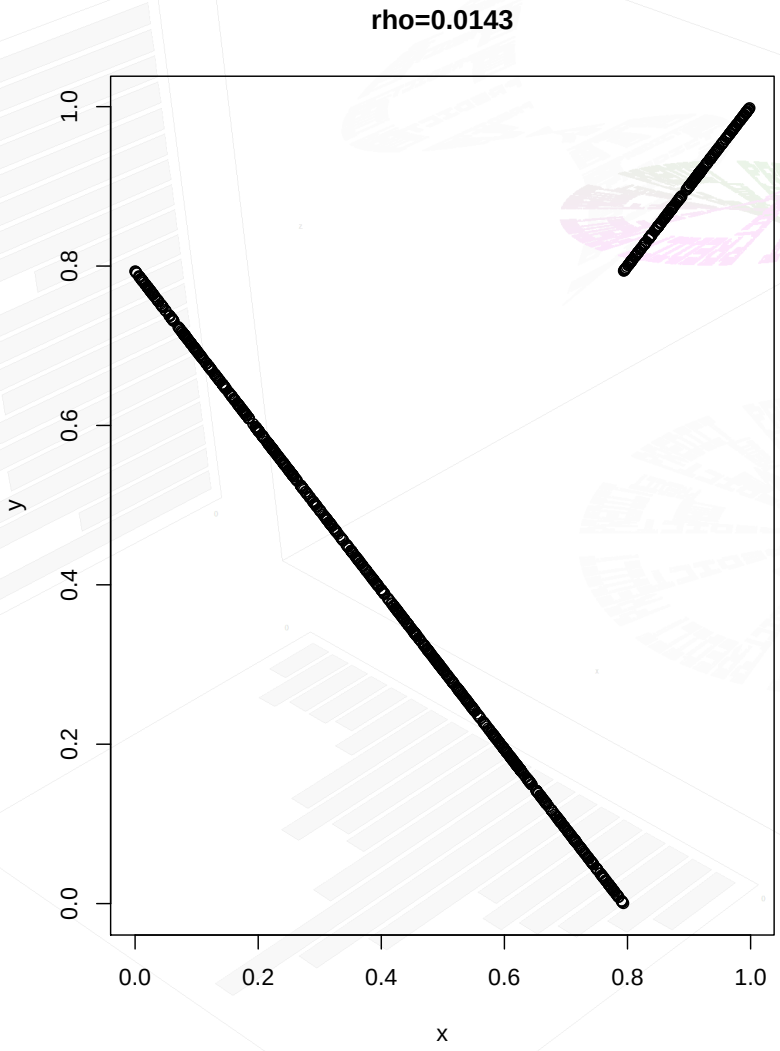


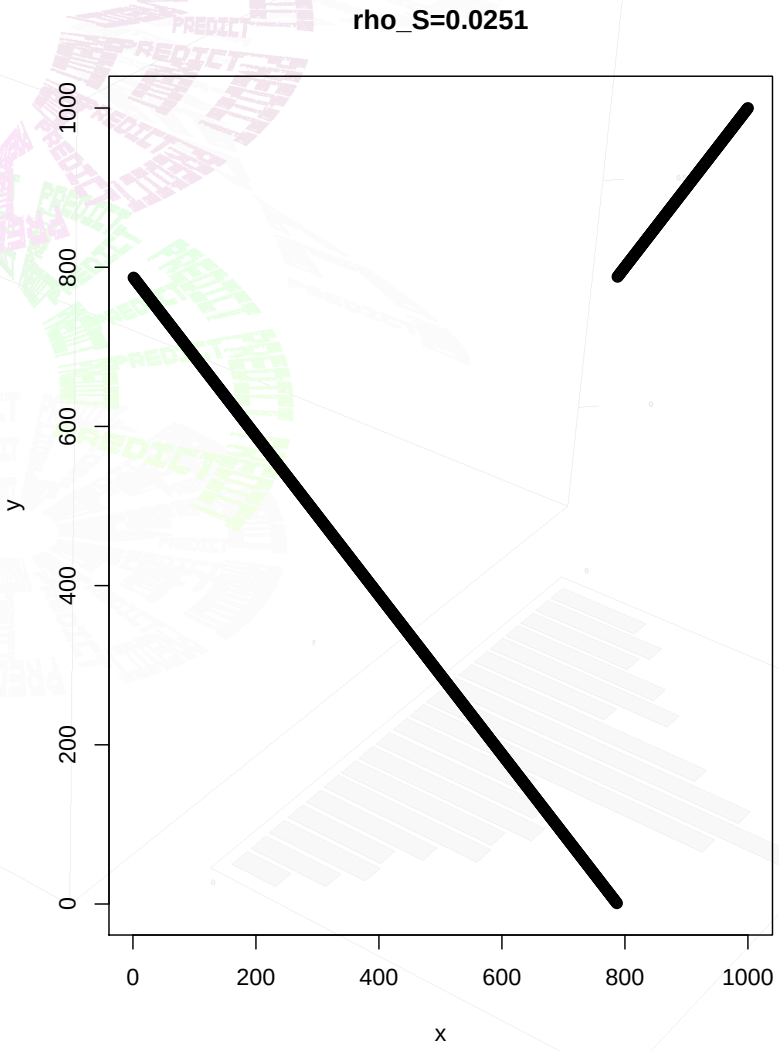
Some examples and exercises



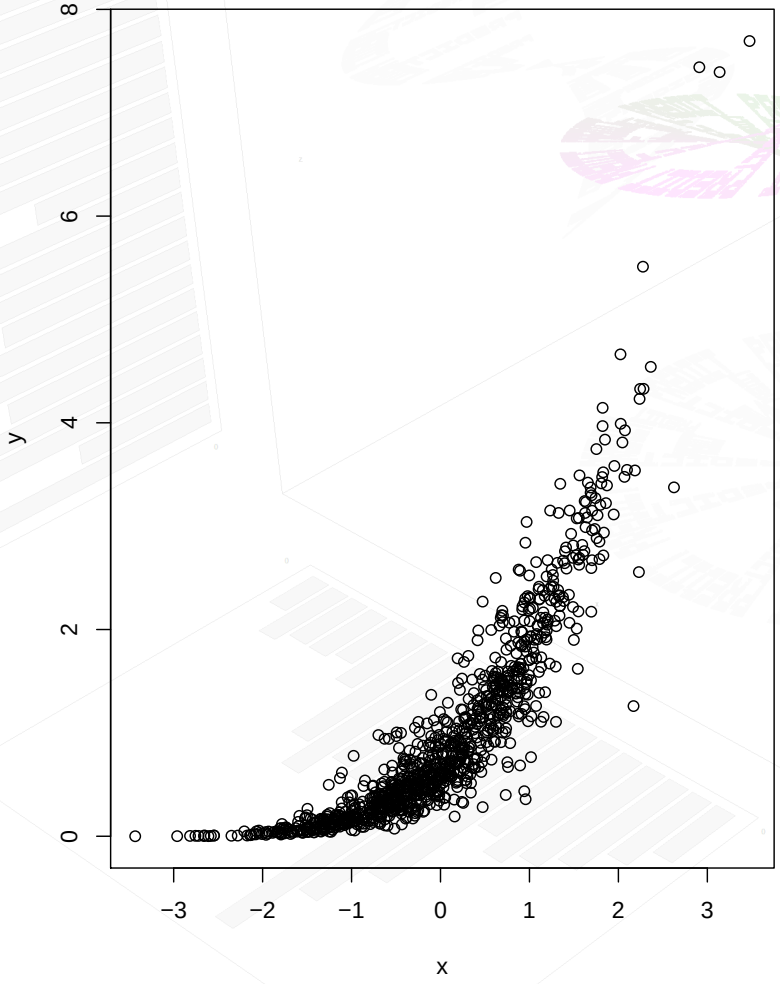


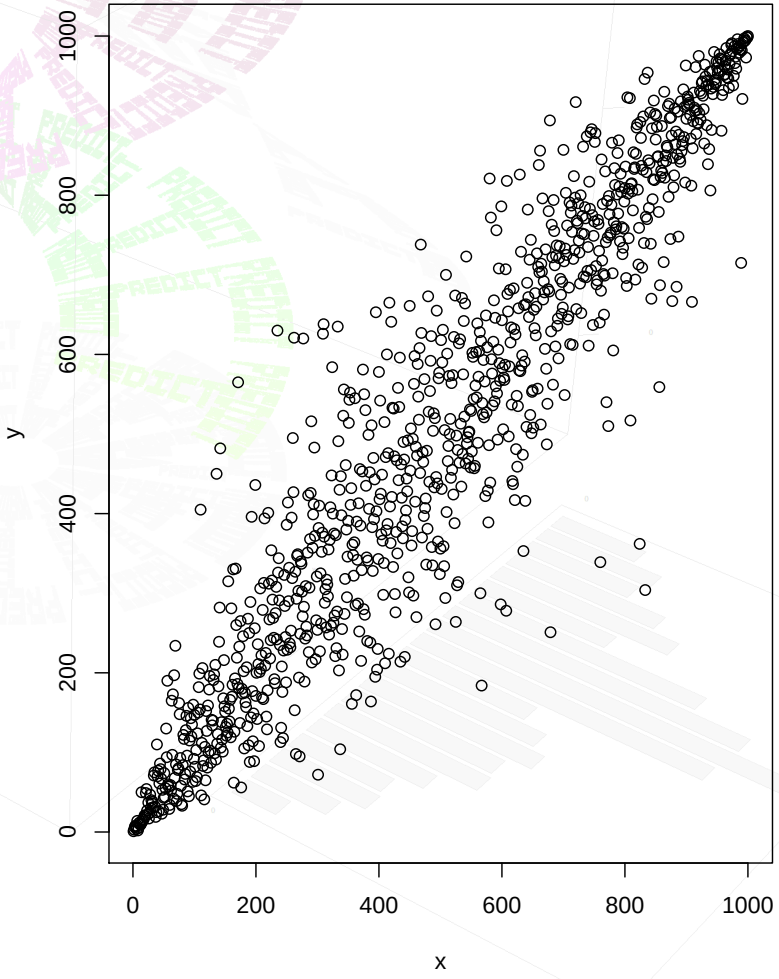
Some examples and exercises



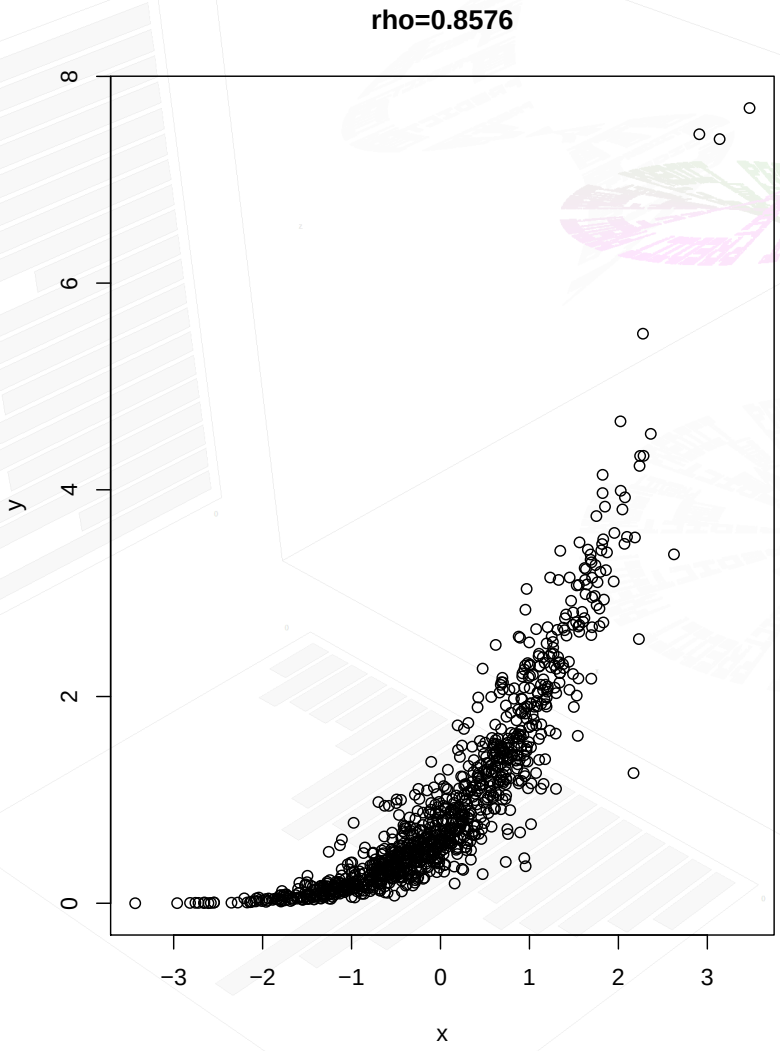


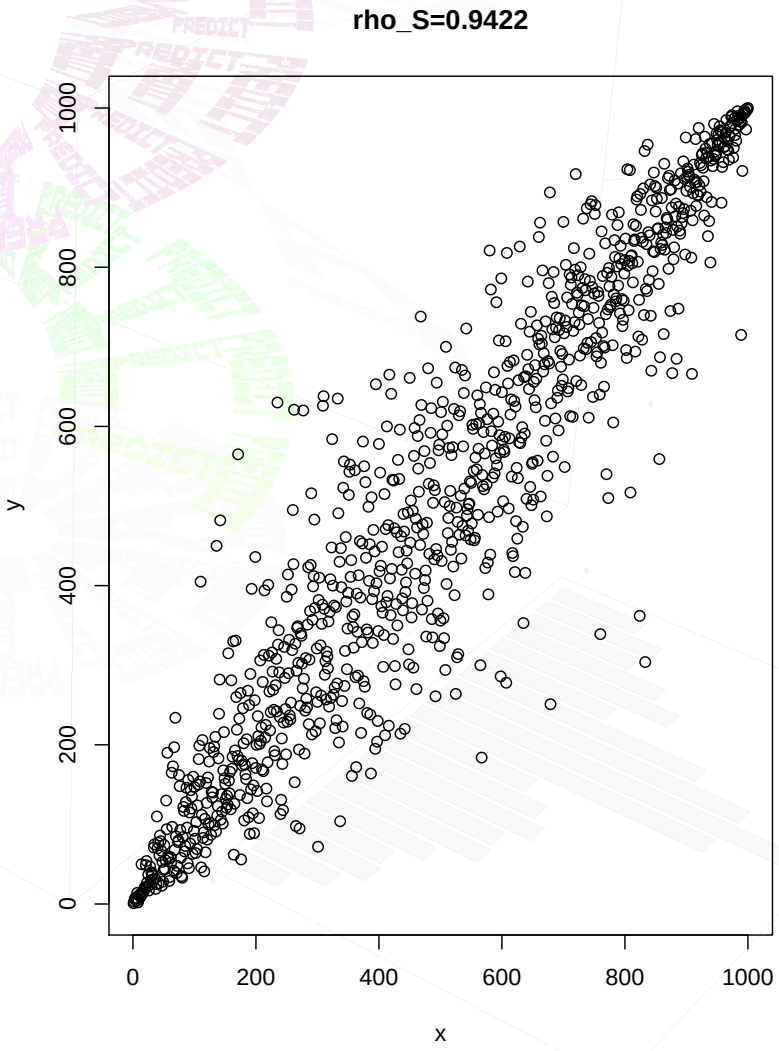
Some examples and exercises





Some examples and exercises





Exercise 05:

- Extend parts of 'R-Codes_Regression01.R' at http://www.trutschnig.net/courses to a knitR report featuring the following:
  - Consider 'Reg_ex01.RData' and produce (i) a scatterplot of the sample as well as (ii) a scatterplot of the ranks
  - Hint: Use 'par(mfrow=c(1,2))' to get two plots in one window
  - Write a sentence below the graphic containing the values of ρ and ρ_S (via Sexpr)
  - Repeat the previous steps with the other four datasets 'Reg_ex02.RData',...,'Reg_ex05.RData'. Start a new section for each dataset
- @Advanced knitR users: Use one loop to create all graphics and one loop (within a chunk) to write all five sections

Known:

- We know that there is a relationship between quantities X and Y of the following form:

$$Y = r(X) + \varepsilon \tag{1}$$

- r is an **unknown** function and ε is a random error fulfilling $\mathbb{E}(\varepsilon) = 0$
- Usually we also assume that ε is not influenced by X (might be a too restrictive condition in various situations)
- We call X the **predictor** and Y the **response**

Wanted:

- Based on observations $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ from (1) we want to determine/estimate the function r (why?)
- If we have a good estimator $\hat{r}$ of r then we can predict Y for arbitrary values of X by considering $\hat{r}(X)$

Example (Offer optimization in supermarkets)

- A supermarket chain wants to optimize their offers
- If the price is only reduced by 5% then the sales numbers will only go up a bit
- If the price is reduced by 50% then the sales numbers will go up a lot but the company might earn less because the margin is too small
- Objective: Determine the optimal price reduction in the sense that the supermarket's profit is maximal
- X...price reduction (absolute or percentage) of a certain product
- Y...net earnings (based on this product)
- $Y = r(X) + \varepsilon$
- What do you think: Is the model solely based on price reduction as predictor good?
- Which other predictors would you choose?

Plan for the next weeks:

- Study different types of functions r and develop a toolbox of estimators
- Learn that a graphical analysis of the data is inevitable
- Use simulations to study the quality of the introduced estimators
- We will start with standard parametric models (linear, polynomial, logistic) and then discuss nonparametric alternatives (kernel regression, loess, etc.)
- Learn how to do (parametric and nonparametric) regression and forecasting in R
- We start with the simplest situation - univariate linear regression

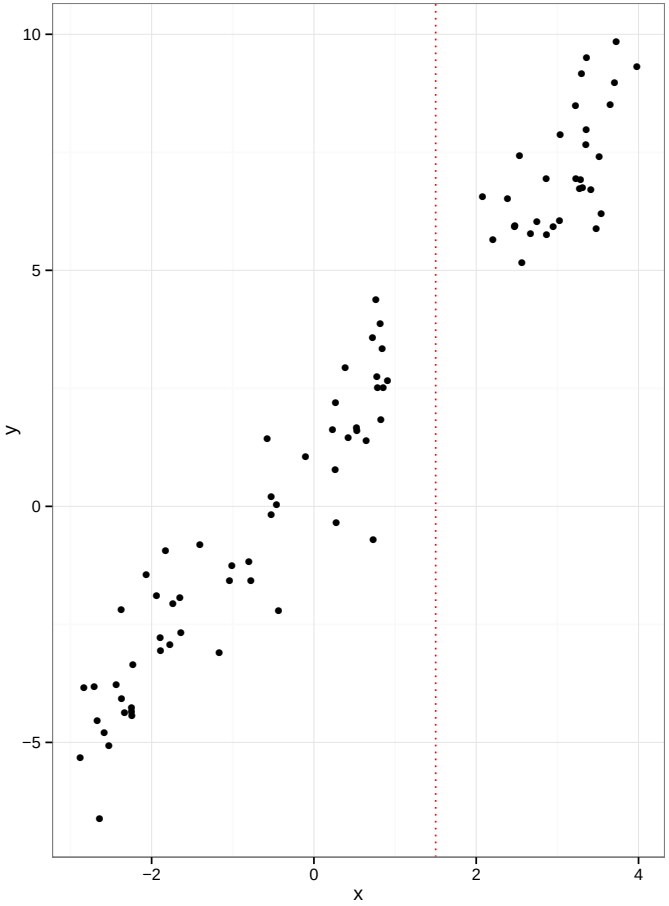


Figure: Prediction at the point $x = 1.5$?

- The graphic depicts measurements $(x_1, y_1), \ldots, (x_n, y_n)$
- It is known that the data comes from the following linear model

$$Y = \underbrace{aX + b}_{r(X)} + \varepsilon$$

- In other words: $y_i = ax_i + b + \varepsilon_i$ for $i \in \{1, \ldots, n\}$
- ε_i...samples of the random error ε fulfilling $\mathbb{E}(\varepsilon) = 0$ that do not influence each other and are not influenced by x_i
- Wanted: Forecast the y-value at the point $x = 1.5$

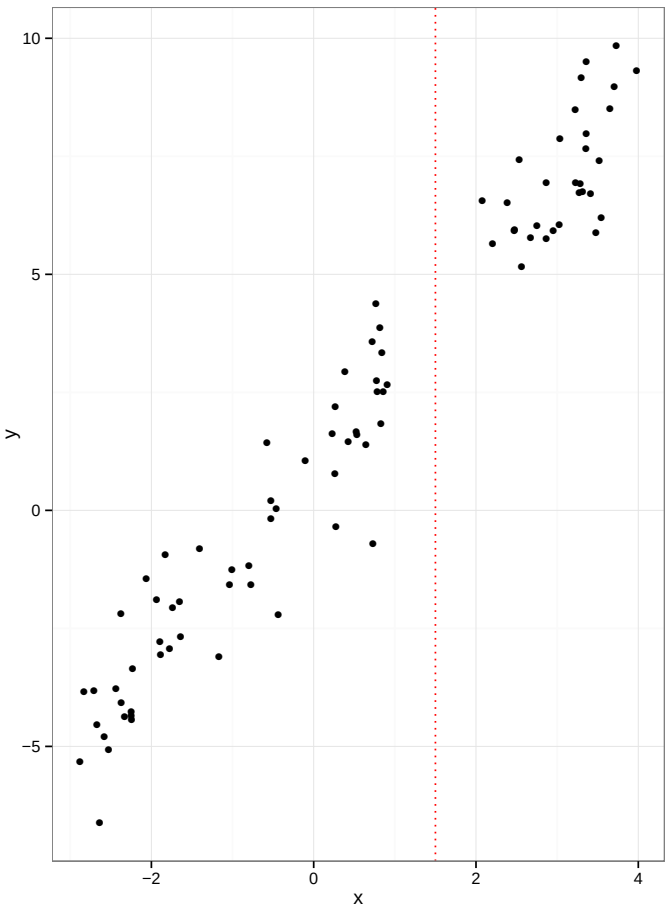
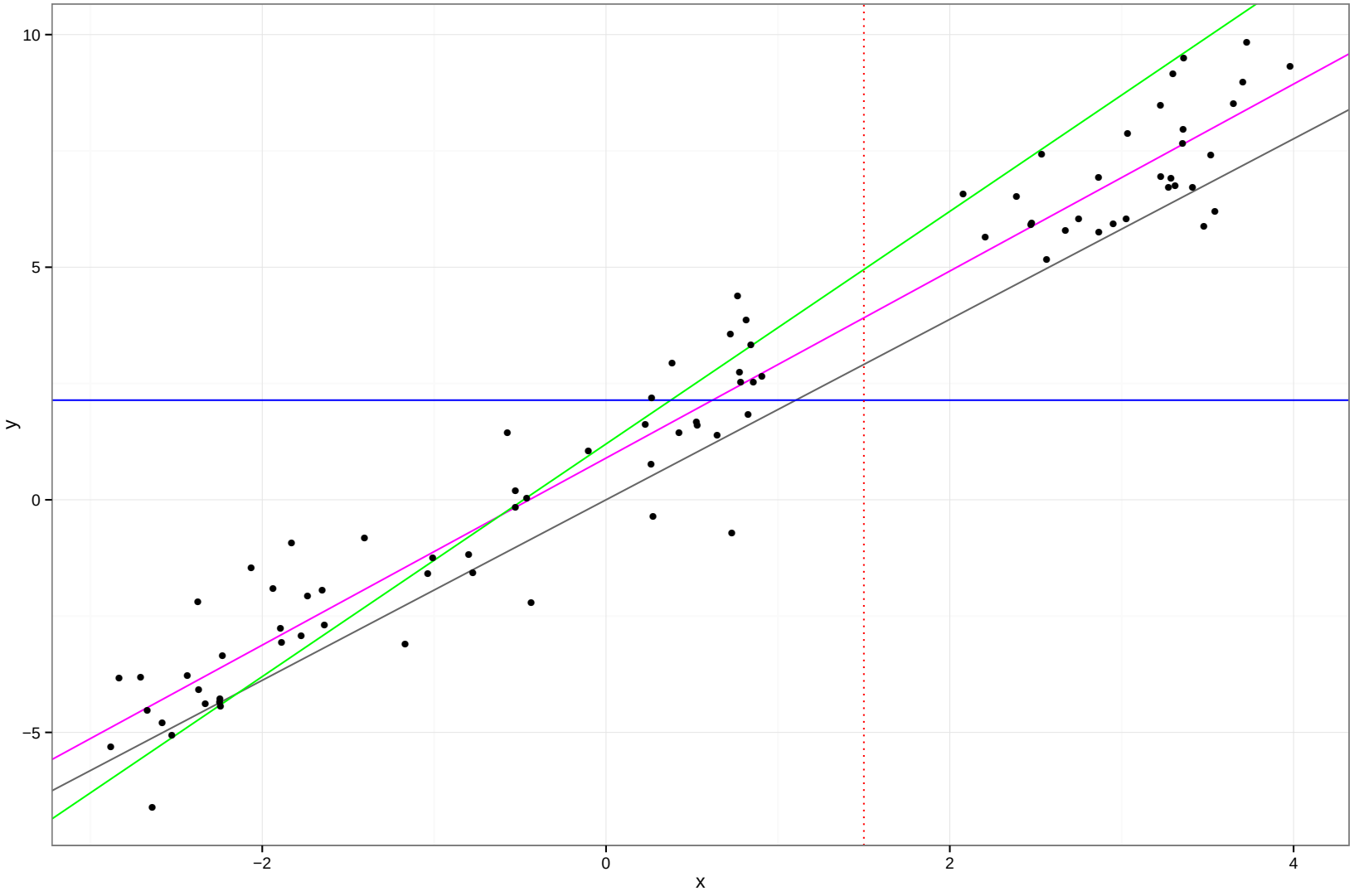


Figure: Prediction at the point $x = 1.5$?

- How would you predict the value at the point $x = 1.5$?
- Problem: We do not know the parameters a and b
- Choose a and b in such a way that the straight line $y = ax + b$ fits the data in the best possible way
- Denote the optimal values by $\hat{a}$ and $\hat{b}$
- Given $\hat{a}$ and $\hat{b}$, predict $\hat{y} = \hat{a}\,1.5 + \hat{b}$.
- Which of the following straight lines fits best?

Univariate setting



- Choose those values for $\tilde{a}$ and $\tilde{b}$ that minimize the prediction errors at the points in the sample
- Choosing $\tilde{a}$ and $\tilde{b}$ as parameters we would forecast $\tilde{a}x_i + \tilde{b}$ for x_i
- The error r_i we make is $r_i = y_i - (\tilde{a}x_i + \tilde{b}) = y_i - \tilde{a}x_i - \tilde{b}$ Plot r_i
- The sum of all squared errors is given by

$$F(\tilde{a}, \tilde{b}) := \sum_{i=1}^{n} \left(y_i - \tilde{a}x_i - \tilde{b}\right)^2 \tag{2}$$

- Choose $\tilde{a}$ and $\tilde{b}$ in such a way that $F(\tilde{a}, \tilde{b})$ is minimal.
- Analytic calculation yields the following optimal values

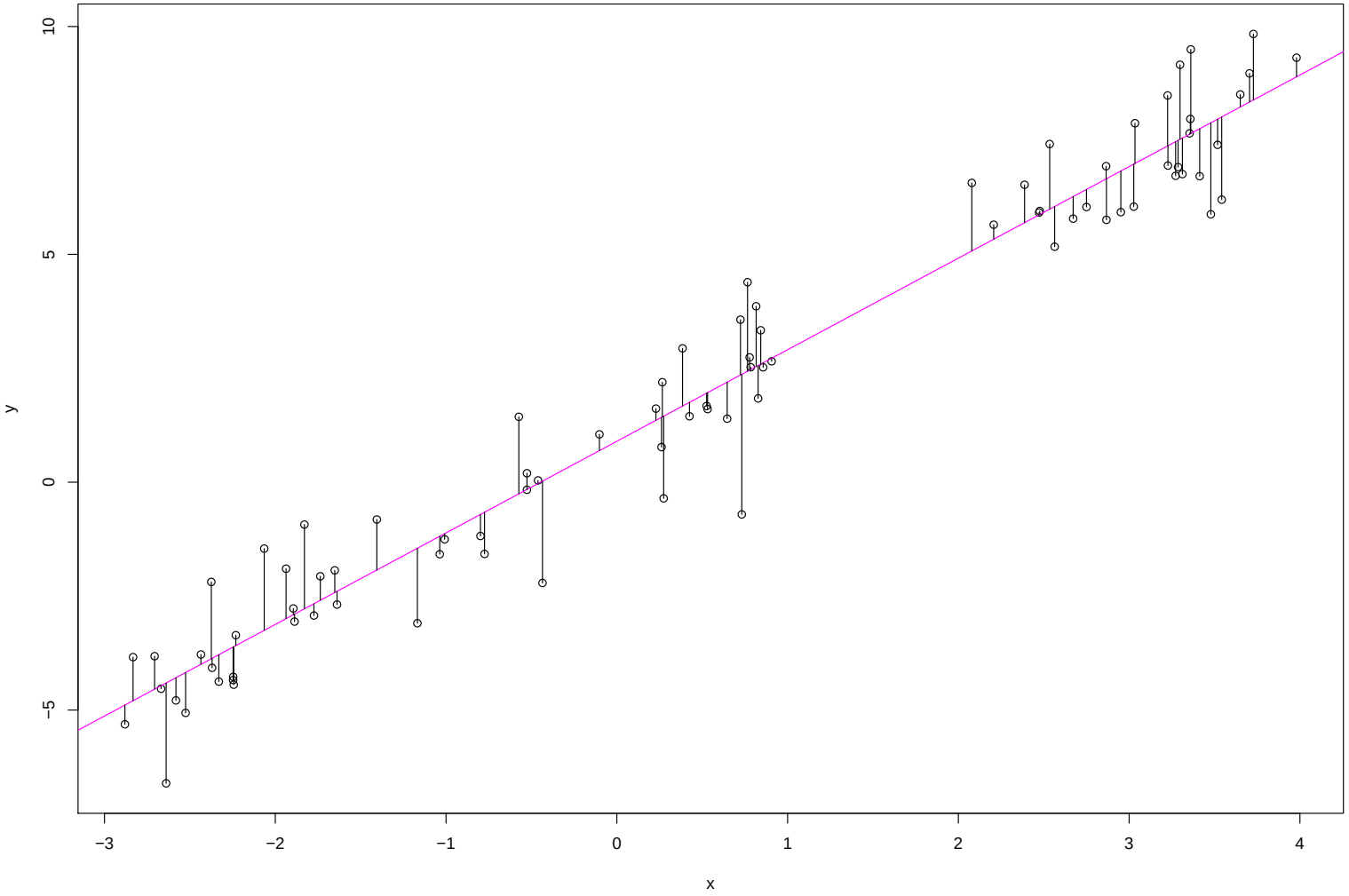
$$\hat{a} = \frac{\sum_{i=1}^{n}(x_i - \overline{x}_n)(y_i - \overline{y}_n)}{\sum_{i=1}^{n}(x_i - \overline{x}_n)^2} = \frac{s_{xy}}{s_x^2} \tag{3}$$

$$\hat{b} = \overline{y}_n - \hat{a}\,\overline{x}_n \tag{4}$$

- For our given sample we get $\hat{a} = 2.010$ and $\hat{b} = 0.897$.
- The forecast at the point $x = 1.5$ therefore is $y = 2.01 \cdot 1.5 + 0.897 = 3.912$

Univariate setting

Back



- Before fitting linear models in R some additional observations:
- The estimate slope $\hat{a} = \frac{s_{xy}}{s_x^2}$ looks a bit like the Pearson correlation $\rho = \frac{s_{xy}}{s_x s_y}$
- Using both expressions we get

$$\hat{a} = \rho \frac{s_y}{s_x}$$

- Increasing x by one standard deviation s_x increases y by ρ standard deviations s_y, in fact

$$\hat{r}(x + s_x) = \hat{a}(x + s_x) + \hat{b} = \underbrace{\hat{a}x + \hat{b}}_{y} + \hat{a}s_x = y + \rho \frac{s_y}{s_x} s_x = y + \rho s_y$$

- How do we quantify if our optimal model offers a good explanation of the model?

- A natural idea is the *coefficient of determination* R^2
- Easy to show (@mathematicians: prove it!):

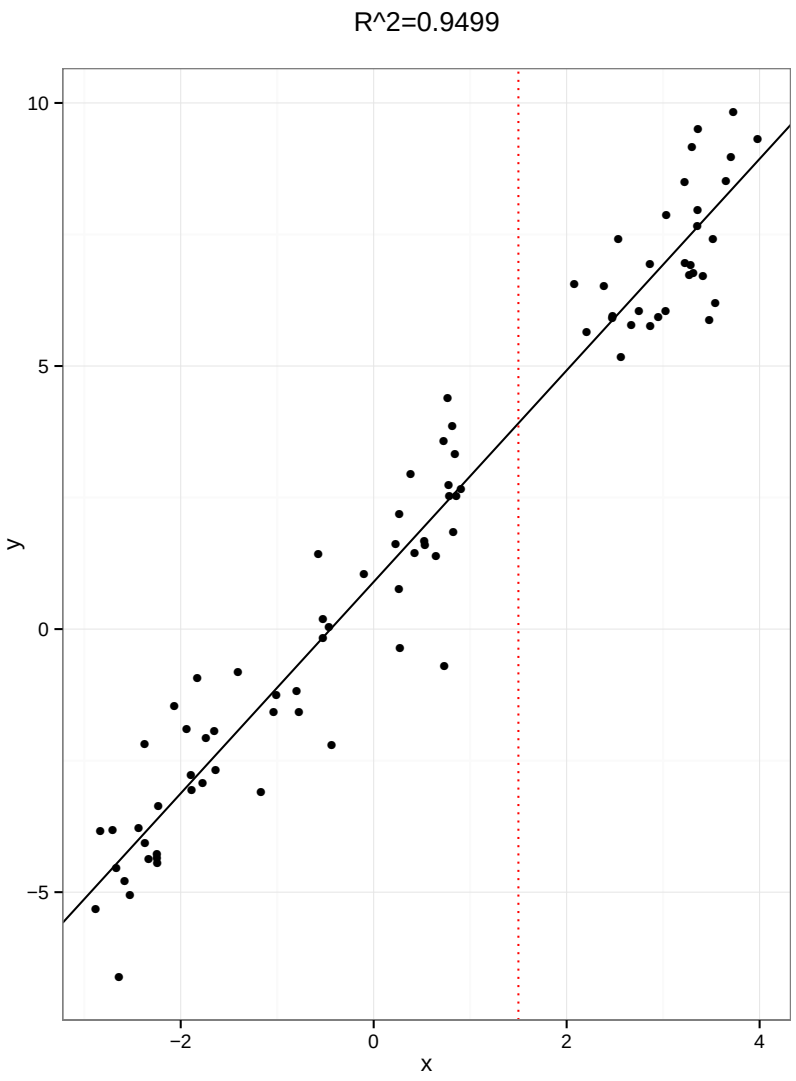
$$\sum_{i=1}^{n}(y_i - \overline{y}_n)^2 = \sum_{i=1}^{n}\underbrace{(y_i - \hat{y}_i)^2}_{r_i^2} + \sum_{i=1}^{n}(\hat{y}_i - \overline{\hat{y}}_n)^2$$

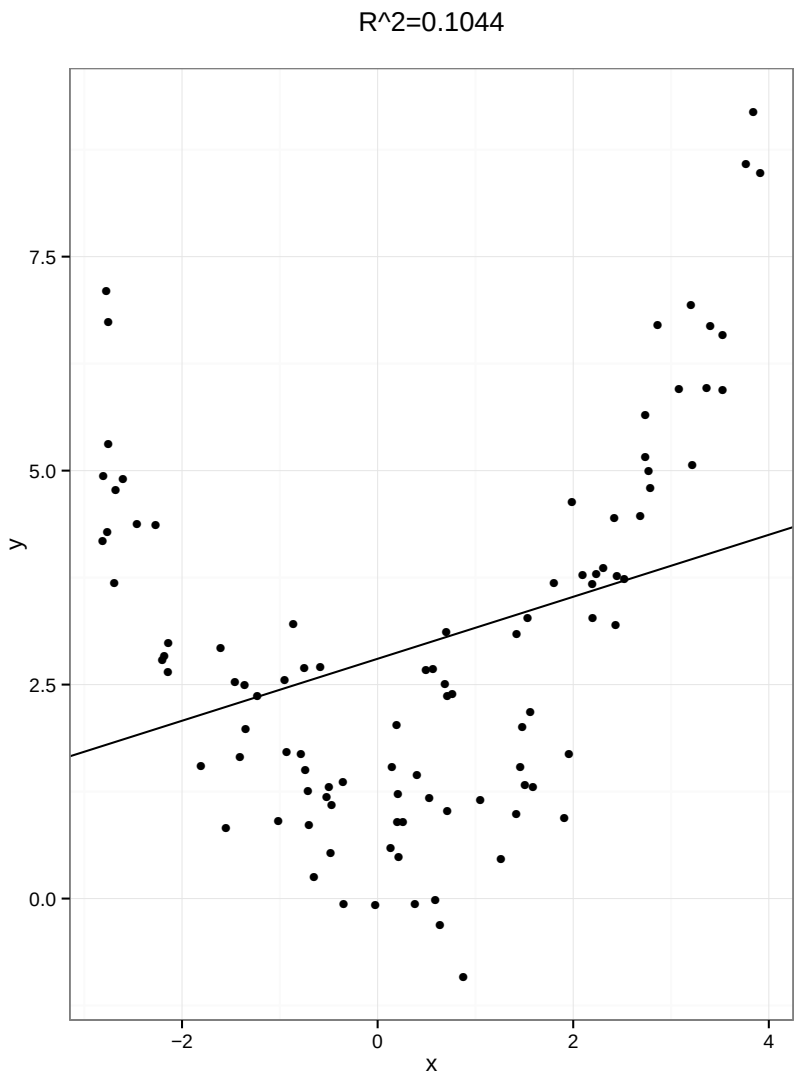
- Variance of $y_1, \ldots, y_n$ equals the variance of the residuals plus the variance of the forecasts $\hat{y}_1, \ldots, \hat{y}_n$
- Calculate

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y}_n)^2} = \frac{\sum_{i=1}^{n}(\hat{y}_i - \overline{\hat{y}}_n)^2}{\sum_{i=1}^{n}(y_i - \overline{y}_n)^2} \tag{5}$$

- R^2 is the proportion of y-variance explained by the model

Univariate setting





Properties of R^2:

- We have $0 \leq R^2 \leq 1$
- The higher R^2 the higher the percentage of variance explained by the model
- If R^2 is close to 1 then the model explains the data very well
- If R^2 is close to 0 the model does not help much to explain the data
- There should be a strong interrelation between R^2 and the correlation ρ of the original sample $(x_1, y_1), \ldots, (x_n, y_n)$...
- Calculations in R will make this clear

```r
file <- url("http://www.trutschnig.net/geo_reg1.RData")
load(file)
head(geo_reg1)
A<-geo_reg1

model<-lm(data=A,y~x)      #use whatever name you want instead of model
summary(model)
```

- yields

```

Call:
  lm(formula = y ~ x, data = A)

Residuals:
   Min      1Q  Median      3Q     Max
-3.07477 -0.63681 -0.03544  0.70030  1.95308
```

- and

```
Coefficients:
  Estimate   Std. Error t value   Pr(>|t|)
(Intercept) 0.89704    0.11406   7.865    1.13e-11 ***
  x         2.00965    0.05035   39.917   < 2e-16 ***
  ---
  Signif. codes:  0   ***   0.001   **   0.01   *   0.05   .
          0.1        1

Residual standard error: 1.03 on 84 degrees of freedom
Multiple R-squared:  0.9499, Adjusted R-squared:  0.9493
F-statistic:  1593 on 1 and 84 DF,  p-value: < 2.2e-16
```

- Calculate the prediction for $x = 1.5$

```
ND<-data.frame(x=c(1.5))
p<-predict(model,new=ND)
p
3.91152
```

Exercise 06:

- Go through part 01 of R-Code_Regression02.R and figure out what the commands do
- Write a knitR report featuring the following (minimal requirements):
- A scatterplot of the data including the regression line
- A table containing the first 6 rows of the data
- The estimated parameters $\hat{a}$ and $\hat{b}$
- A boxplots of the residuals $r_1, \ldots, r_n$
- Mean and median of the residuals in a sentence using Sexpr
- Calculate ρ and ρ_S of the data and include the values in a small table
- Forecast $r(x)$ for $x \in \{0, 0.1, 0.2, \ldots, 0.9, 1\}$ and include the resulting forecasts in a table

Exercise 07 (Extension of Exercise 04):

- Extend R-Code_RTR.R to a knitR report summarizing the dataset RTR_SBG.RData
- The following items have to be contained in the report (minimal requirements)
  - A scatterplot of the GPS-coordinates of the measurements
  - Time, coordinates, district and mobile of the fastest measurement
  - Boxplots illustrating the download-speed development per operator over time
  - Two tables containing the first 6 rows of the data (as on the last two pages)
  - A table containing the average download speed per district (use 'summaryBy')
  - A table listing the number of measurements per district and operator (use 'summaryBy' and choose 'length' as FUN)
- Include a scatterplot of 'dls' and 'uls' and add the linear regression
- Repeat the previous step for the ranks

Exercise 08 for mathematicians:

Verify the following assertions

- Eq. (5) concerning R^2
- $\overline{r}_n = \frac{1}{n}\sum_{i=1}^{n} r_i = 0$ with $r_i = y_i - \hat{y}_i$
- $R^2 = \rho^2$

Summary @univariate linear regression

- $(x_1, y_1), \ldots, (x_n, y_n)$ are observations from the model $Y = aX + b + \varepsilon$
- Thereby ε was a random error fulfilling $\mathbb{E}(\varepsilon) = 0$; set $\sigma^2 = \mathbb{V}(\varepsilon)$
- In other words: $y_i = ax_i + b + \varepsilon_i$ for every $i \in \{1, \ldots, n\}$
- Using least squares we got the following estimators $\hat{a}$ of a and $\hat{b}$ of b

$$\hat{a} = \frac{\sum_{i=1}^{n}(x_i - \overline{x}_n)(y_i - \overline{y}_n)}{\sum_{i=1}^{n}(x_i - \overline{x}_n)^2} = \frac{s_{xy}}{s_x^2} \tag{6}$$

$$\hat{b} = \overline{y}_n - \hat{a}\,\overline{x}_n \tag{7}$$

- We hope to get $\hat{a} \approx a$ and $\hat{b} \approx b$, i.e. we hope that the estimates are close to the true values
- Will this always be the case?
- When can we expect to get good estimates?

Plan for today:

- We analyze the quality of the estimates $\hat{a}, \hat{b}$ by running simulations (no deeper knowledge in probability theory and statistics needed, simulations should be part of EVERY statistics lecture!)
- @mathematicians in semester ≥ 7: Also try to prove the results analytically!

How to run the simulations

- Fix a and b
- Fix the sample size n
- Consider (or generate) some values $x_1, \ldots, x_n$
- Generate random errors $\varepsilon_1, \ldots, \varepsilon_n$
- Set $y_i = ax_i + b + \varepsilon_i$
- Consider the sample $(x_1, y_1), \ldots, (x_n, y_n)$ and calculate $\hat{a}$ and $\hat{b}$
- Check how close $\hat{a}$ and $\hat{b}$ are to a and b
- Repeat the above steps several times

@Performance

```
#one simulation
a<-2;b<-1
n<-100
x<-runif(n,-3,4)
error<-rnorm(n,0,1)    #error from normal distribution N(0,1)
y<-a*x+b+error
A<-data.frame(x=x,y=y)
plot(A)
model<-lm(data=A,y~x)
abline(model)
summary(model)
```

- yields

```

Call:
lm(formula = y ~ x, data = A)

Residuals:
     Min       1Q   Median       3Q      Max
-2.20647 -0.66814 -0.09888  0.77627  1.95348
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.76146    0.10149   7.503 2.87e-11 ***
x            2.09902    0.04686  44.790  < 2e-16 ***
---
Signif. codes:  0  ***  0.001  **  0.01  *  0.05  .  0.1    1

Residual standard error: 0.9631 on 98 degrees of freedom
Multiple R-squared:  0.9534, Adjusted R-squared:  0.9529
F-statistic:  2006 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
sum(model$residuals^2)/(n-2)
```

- yields

```

[1] 0.9275251
```

```r
#several runs
R<-1000
E<-data.frame(a=rep(0,R),b=rep(0,R))

a<-2;b<-1
n<-100
for(i in 1:R){
  x<-runif(n,-3,4)
  error<-rnorm(n,0,1)
  y<-a*x+b+error
  A<-data.frame(x=x,y=y)
  model<-lm(data=A,y~x)
  E[i,]<-as.numeric(coefficients(model))[2:1]
}
```

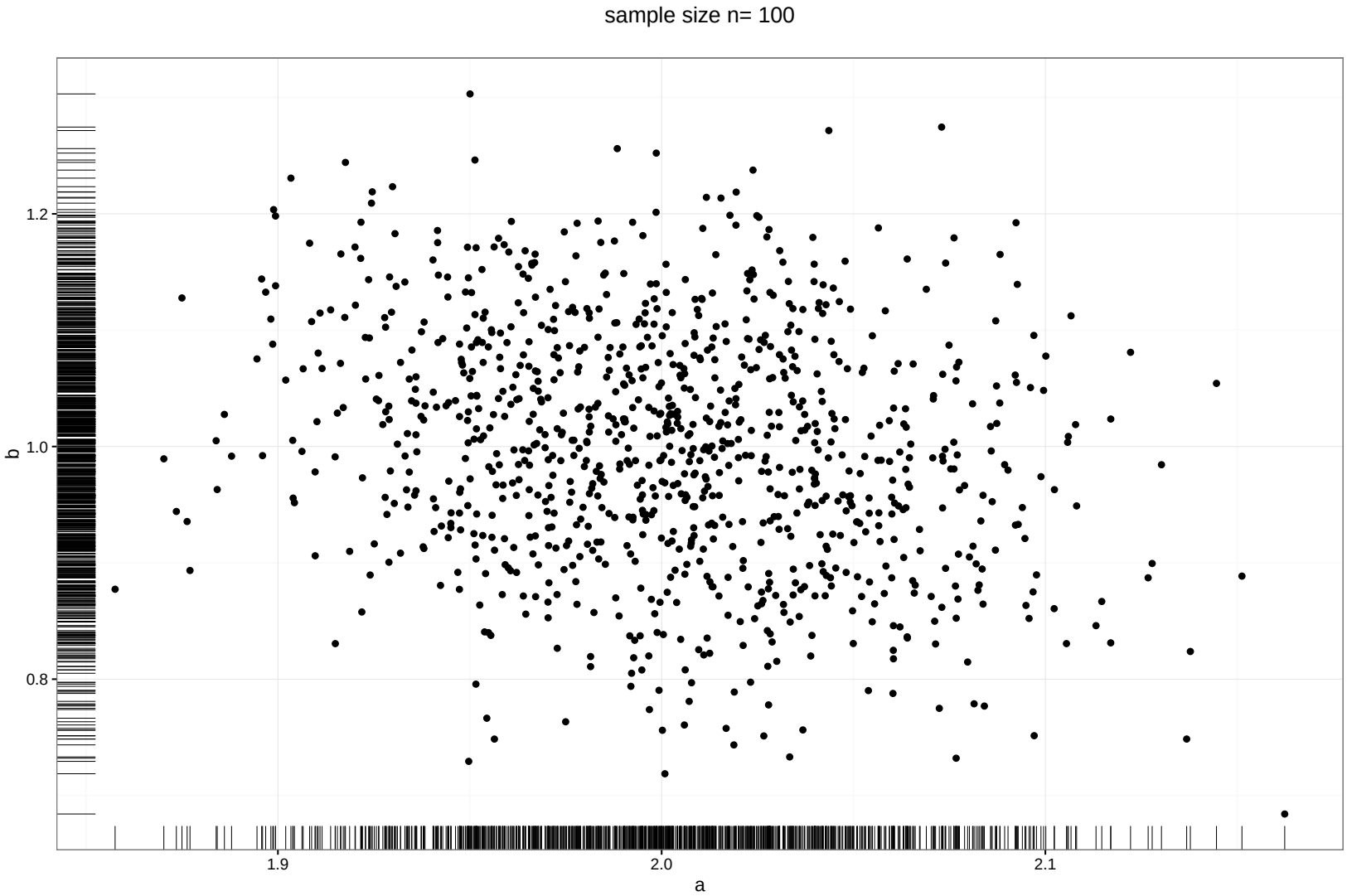
- yields

```
          a         b
1 1.994841 0.8079434
2 1.987354 1.0237531
3 1.951075 0.9251133
4 1.999110 1.0721703
5 1.996653 0.8200005
6 1.968700 1.0383586
```

```
        a                b
Min.   :1.858   Min.   :0.6841
1st Qu.:1.966   1st Qu.:0.9280
Median :2.001   Median :0.9994
Mean   :2.002   Mean   :0.9989
3rd Qu.:2.035   3rd Qu.:1.0722
Max.   :2.162   Max.   :1.3031
```

- What does the table tell us?
- A graphical overview also helps to interpret the results

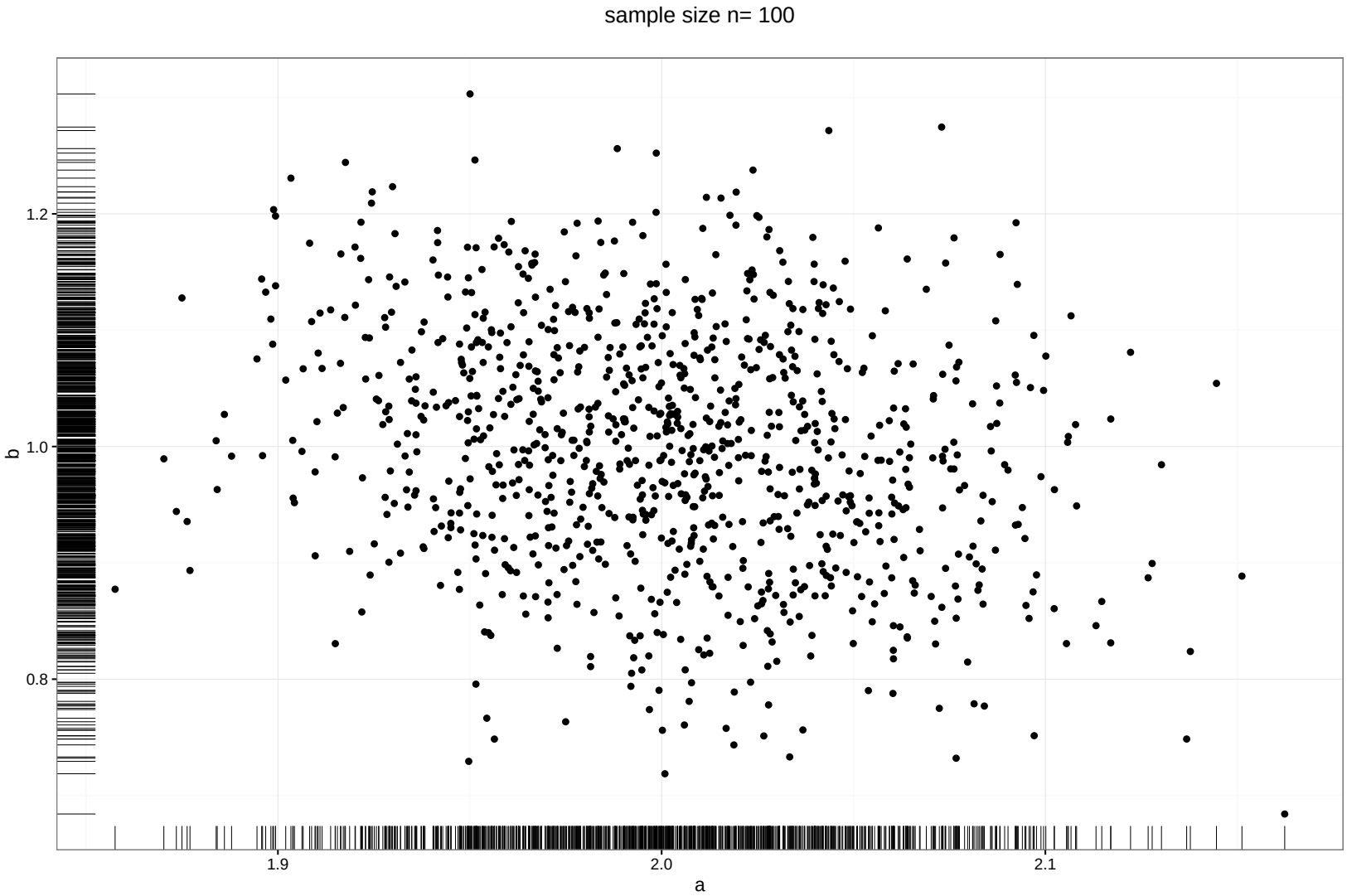
@Performance



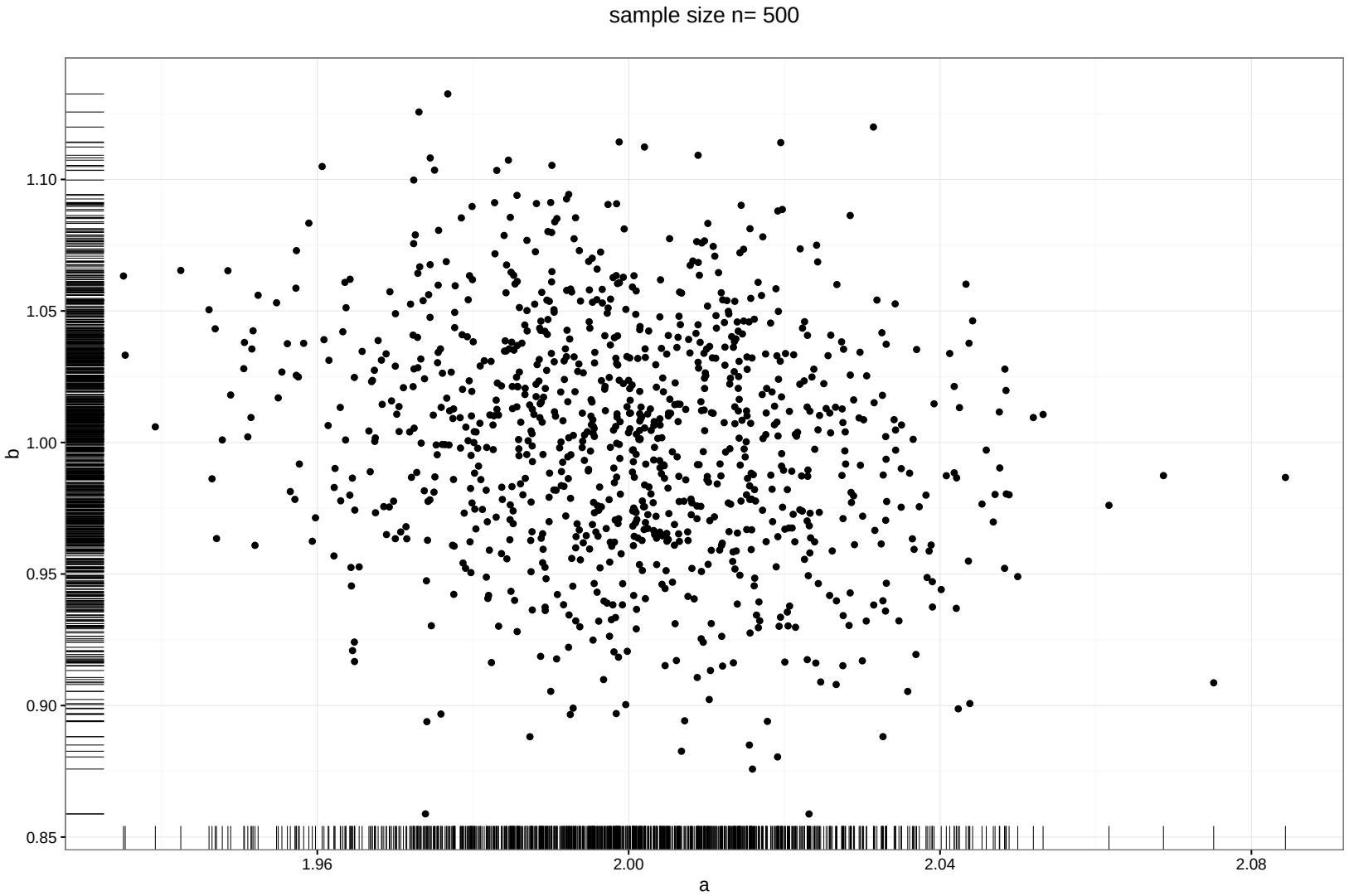
Natural related questions:

- What happens if the sample size n is increased?
- **The more info the better the estimates should (on average) be!**
- What other parameter in the simulation could have an influence on the quality of the estimates?
- Answer: The variance σ^2 of ε is important
- **The higher the variance the poorer the estimates**
- Repeat the simulation (several runs) for higher and lower sample size and vary the variance of the error

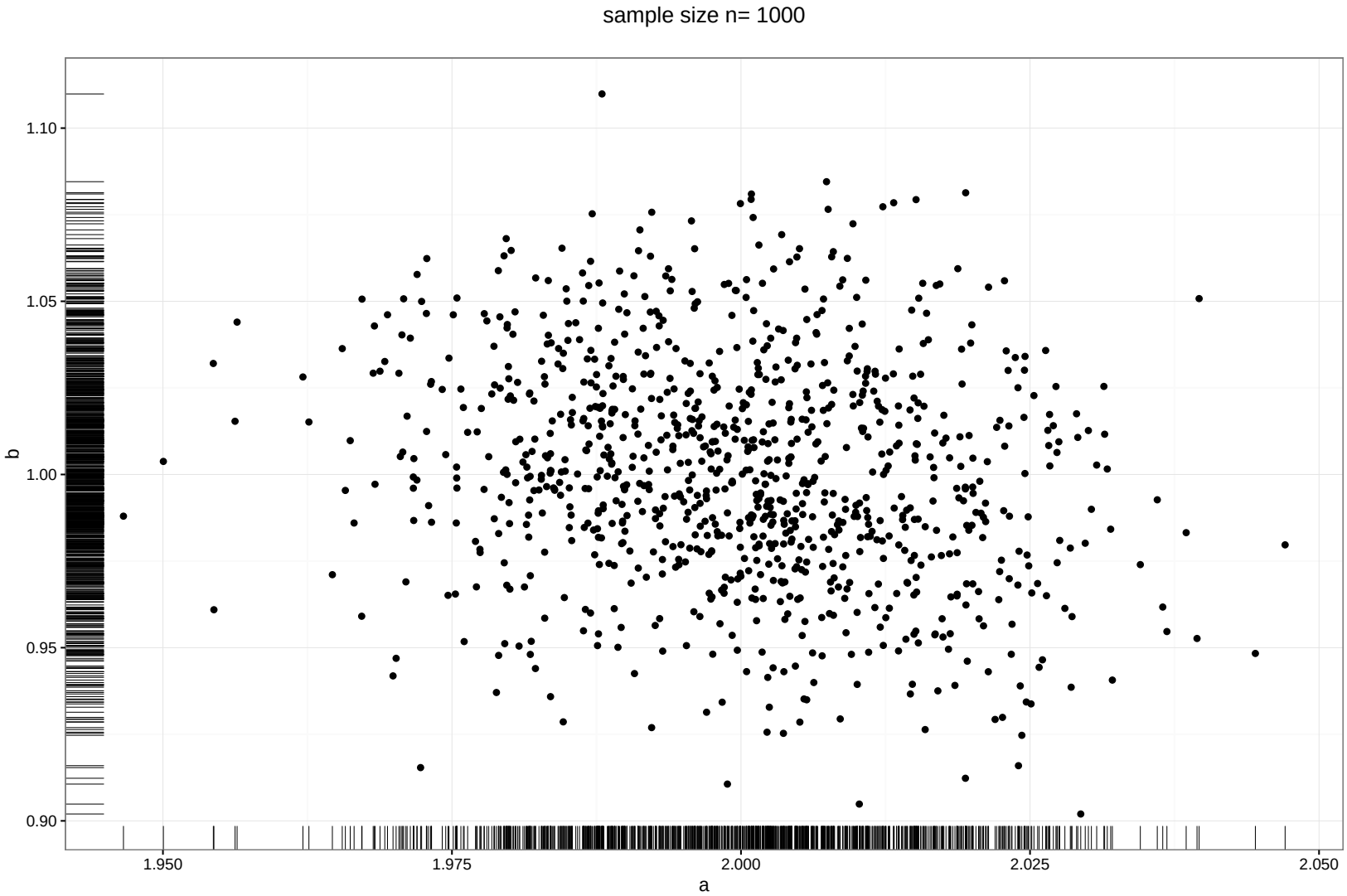
Influence of the parameters at stake



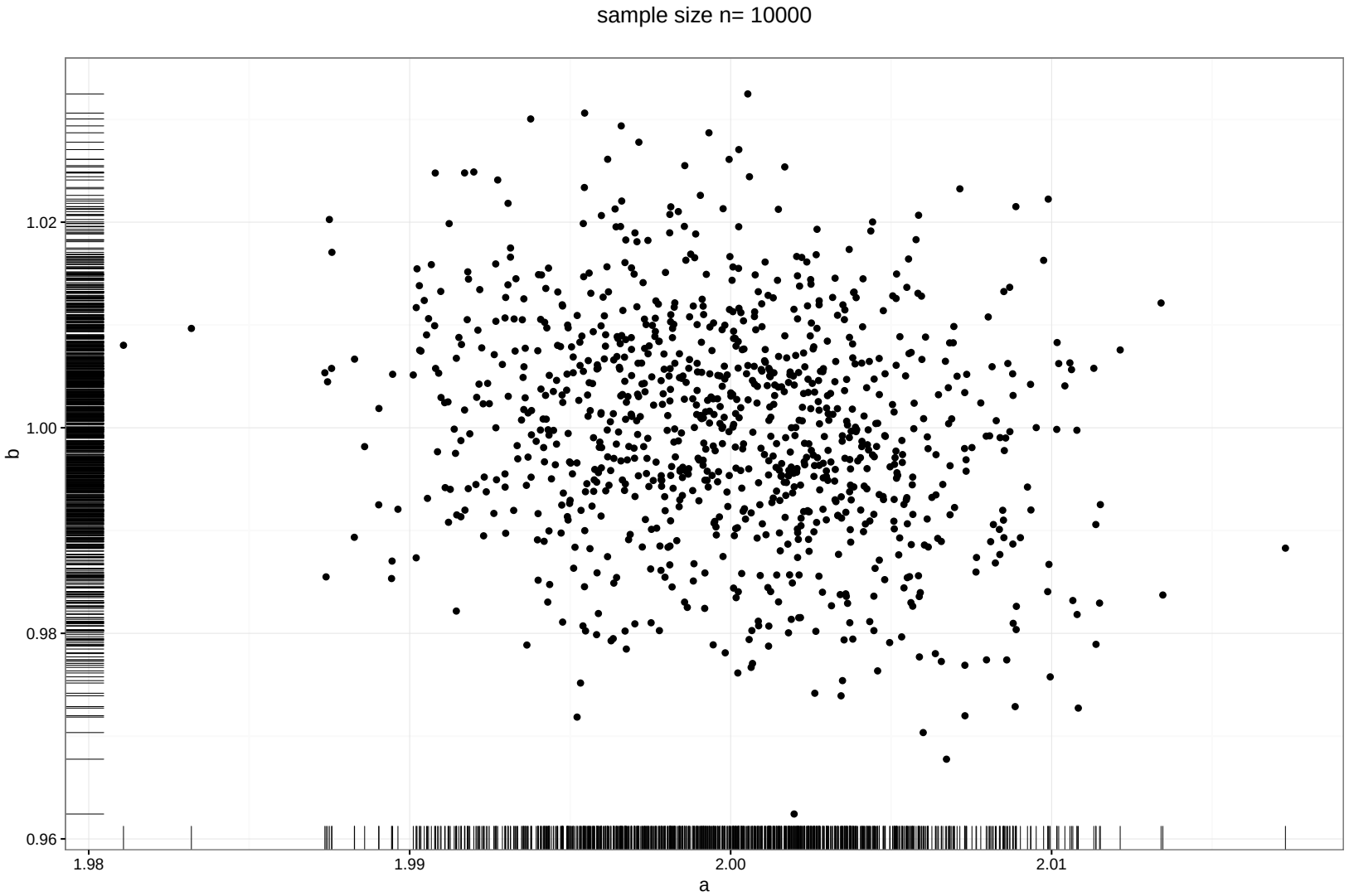
Influence of the parameters at stake



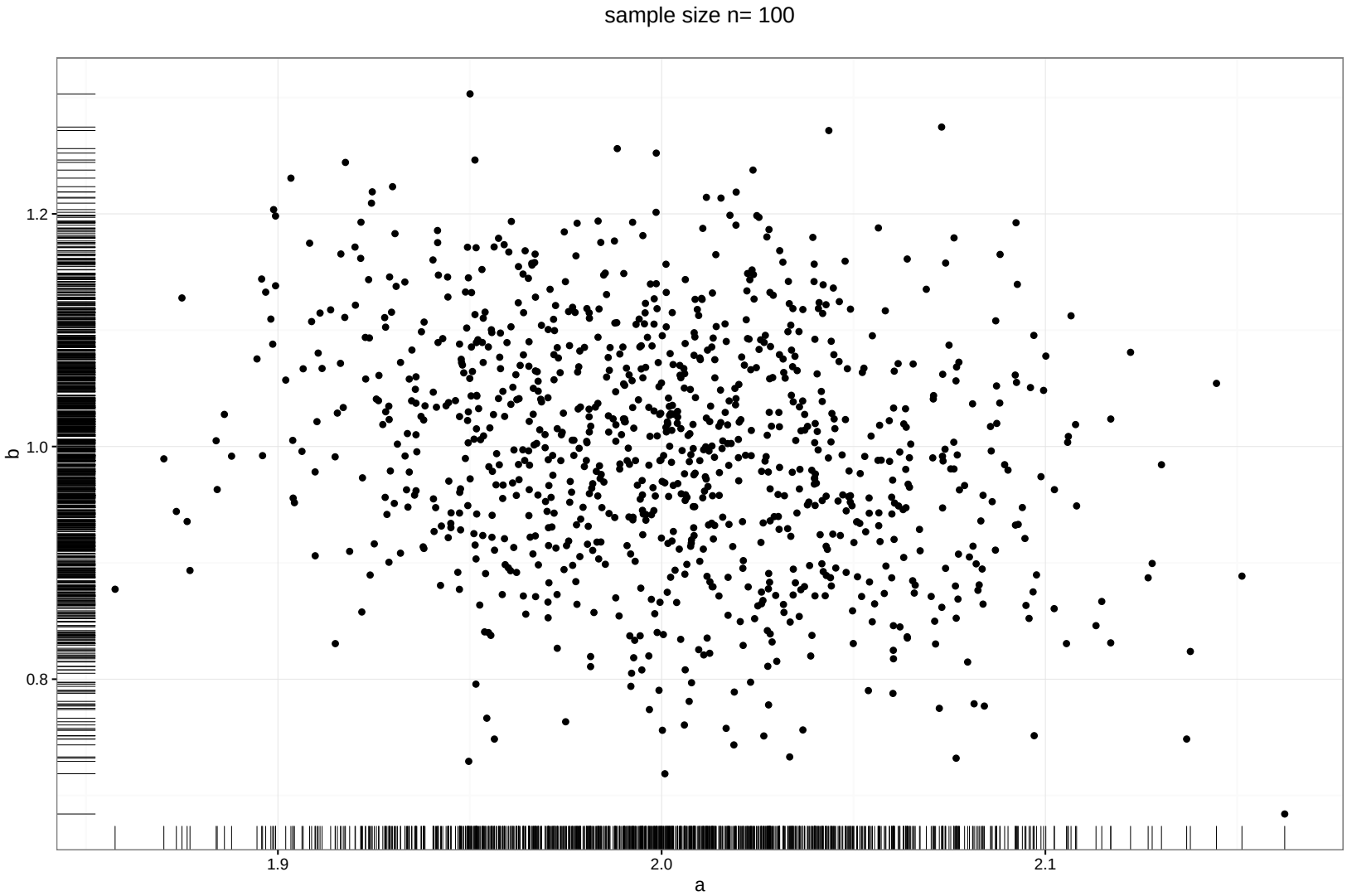
Influence of the parameters at stake



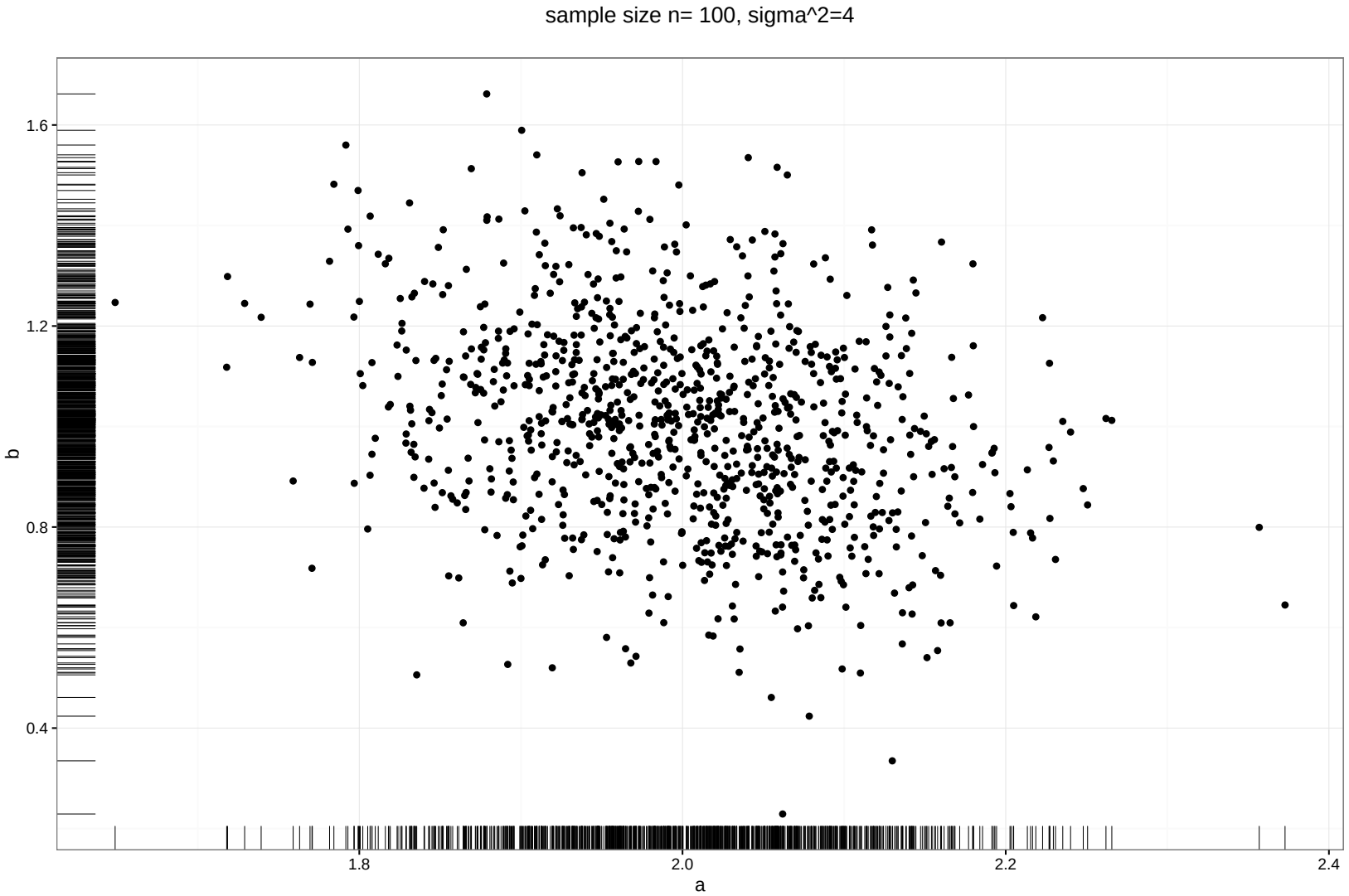
Influence of the parameters at stake



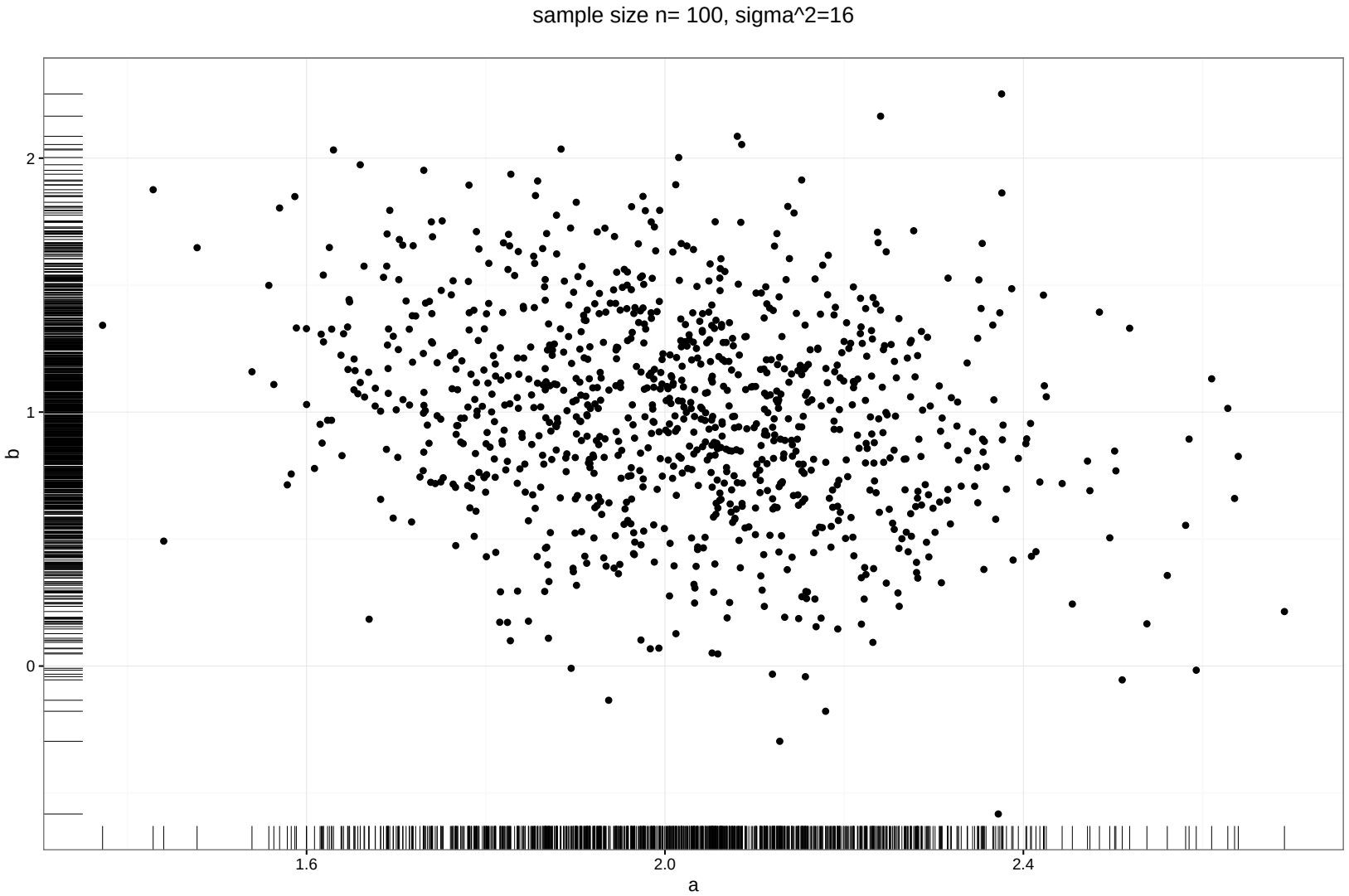
Influence of the parameters at stake



Influence of the parameters at stake



Influence of the parameters at stake



Exercise 09:
Use parts of R-Code_Regression03.R to write a knitR simulation study featuring the following (minimal requirements):

Section 1:

- Simulate a sample of size $n = 100$ from the model $Y = 0.5X - 1 + \varepsilon$ whereby $\varepsilon \sim \mathcal{N}(0, 0.5)$
- Include a scatterplot of the data including the regression line
- The estimated parameters $\hat{a}$ and $\hat{b}$
- A boxplots of the residuals $r_1, \ldots, r_n$
- Mean and median of the residuals in a sentence using Sexpr
- Calculate ρ and ρ_S of the data and include the values in a small table
- Forecast $r(x)$ for $x \in \{0, 0.1, 0.2, \ldots, 0.9, 1\}$ and include the resulting forecasts in a table

Section 2:

- Simulate a sample of size $n = 100$ from the model $Y = 0.5X - 1 + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, 0.5)$
- Save the estimated parameters $\hat{a}$ and $\hat{b}$ in a data.frame A
- Repeat the previous two steps $R = 1000$ times
- Include a boxplots of the estimates $\hat{a}_1, \ldots, \hat{a}_R$ and a boxplot of the estimates $\hat{b}_1, \ldots, \hat{b}_R$
- Calculate the biggest, the smallest and the median value of $\hat{a}_1, \ldots, \hat{a}_R$ and report the values in a table
- Calculate the biggest, the smallest and the median value of $\hat{b}_1, \ldots, \hat{b}_R$ and report the values in a table
- Repeat the previous steps for bigger sample size and/or for bigger variance of the errors

Section 3+4:

- In the literature one frequently sees that errors should have normal distribution
- Consider $\mathcal{U}(-1,1)$-distributed errors using the command error=runif(n,-1,1) and repeat the tasks in Section 1 and Section 2 for this situation
- Do we also get good results in this setting?
- Summarize your observations in some sentences

Section 5 (only for mathematicians in semester ≥ 7):

- Analyze standard properties of the estimators $\hat{a}, \hat{b}$ (under the assumption that the values $x_1, \ldots, x_n$ are fixed) - are they (strongly) consistent, are they unbiased?
- Which distribution does $\hat{a}$ have if we know that $\varepsilon_1, \ldots, \varepsilon_n$ is a sample from $\varepsilon \sim \mathcal{N}(0, \sigma^2)$?
- Which distribution does $\hat{b}$ have if we know that $\varepsilon_1, \ldots, \varepsilon_n$ is a sample from $\varepsilon \sim \mathcal{N}(0, \sigma^2)$?
- What can be said in the general case only knowing $\mathbb{E}(\varepsilon) = 0$ and $\mathbb{V}(\varepsilon) < \infty$

- Natural generalization of the univariate linear model to several predictors
- Consider the case of two predictors X_1 and X_2 (analogously for more than two) and one response variable Y.
- In this case the model is of the form: $Y = a_1 X_1 + a_2 X_2 + \varepsilon$
- In other words: we assume that the data $(x_{1,1}, x_{2,1}, y_1), (x_{1,2}, x_{2,2}, y_2), \ldots (x_{1,n}, x_{2,n}, y_n)$ fulfills $y_i = a_1 x_{1,i} + a_2 x_{2,i} + b + \varepsilon_i$
- As before: ε_i... independent random errors with fixed variance and $\mathbb{E}(\varepsilon_i) = 0$
- We proceed as in the univariate case and want to minimize

$$F(\tilde{a}_1, \tilde{a}_2, \tilde{b}) := \sum_{i=1}^{n} \left(y_i - \tilde{a}_1 x_{1,i} - \tilde{a}_2 x_{2,i} - \tilde{b}\right)^2 \tag{8}$$

- Choose $\tilde{a}_1, \tilde{a}_2, \tilde{b}$ in such a way that $F(\tilde{a}_1, \tilde{a}_2, \tilde{b})$ is minimal
- We fit the two-dimensional model to the data geo_reg_d2.RData

```
#two dim:
file <- url("http://www.trutschnig.net/geo_reg_d2.RData")
load(file)
A<-geo_reg_d2
head(A)

model<-lm(data=A,y~x1+x2)
model
summary(model)
```

- yields

```
Call:
  lm(formula = y ~ x1 + x2, data = A)

Residuals:
   Min      1Q  Median      3Q     Max
-3.5631 -0.6376  0.0564  0.9176  2.1860
```

- and

```
Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.04404    0.11306    0.39    0.698
x1           2.93824    0.04889   60.10   <2e-16 ***
x2           1.96832    0.06229   31.60   <2e-16 ***
  ---
  Signif. codes:  0   ***   0.001   **   0.01   *   0.05   .
          0.1        1

Residual standard error: 1.096 on 97 degrees of freedom
Multiple R-squared:  0.9791, Adjusted R-squared:  0.9787
F-statistic:  2273 on 2 and 97 DF,  p-value: < 2.2e-16
```

- Calculate the prediction for $x1 = 1.5$ and for $x = 0$

```
ND<-data.frame(x1=c(1.5),x2=c(0))
p<-predict(model,new=ND)
p
4.4514
```

Exercise 11: Analogously to the case of univariate linear regression do a simulation study to find out whether the estimates $\tilde{a}_1, \tilde{a}_2, \tilde{b}$ are close to the true values a_1, a_2, b. Proceed as follows and use http://www.trutschnig.net/R-Codes_Regression03.R as draft:

- Fix a_1, a_2 and b and fix the sample size n
- Consider (or generate) some values $x_{1,1}, \ldots, x_{1,n}$ and $x_{2,1}, \ldots, x_{2,n}$
- Generate random errors $\varepsilon_1, \ldots, \varepsilon_n$
- Set $y_i = b + a_1 x_{1,i} + a_2 x_{2,i} + \varepsilon_i$ for every $i \in \{1, \ldots, n\}$
- Consider the sample $(x_{1,1}, x_{2,1}, y_1), \ldots, (x_{1,n}, x_{2,n}, y_n)$ and calculate $\hat{a}_1, \hat{a}_2$ and $\hat{b}$
- Check how close $\hat{a}_1$, $\hat{a}_2$ and $\hat{b}$ are to a_1,a_2 and b
- Repeat the above steps at least $R = 1000$ times

Summarize the most important observations (influence of sample size, variance of the errors, etc.) in a knitR report.

- Syntax analogous to the univariate setting
- R-output analogous to the univariate setting
- Increasing the sample size yields better estimates (of the coefficients b, a_1, a_2)
- Increasing the variance of the errors implies (on average) worse estimates
- Anything else that might be relevant? Can we always fit a multivariate linear model without further ado?

Example (Two-dimensional linear regression)

Consider the following R-Code (also contained in http://www.trutschnig.net/R-Codes_Regression05.R)

```
n <- 100
x1 <- runif(n=n,-10,10)
x2 <- 2*x1+runif(n,-0.1,0.1)
y <- 3*x1+x2+runif(n,-1,1)
A <- data.frame(x1=x1,x2=x2,y=y)
```

Example (Two-dimensional linear regression, cont.)

```
model <- lm(data=A, y~x1+x2)
model
```

- Yields

```
Call:
  lm(formula = y ~ x1 + x2, data = A)

Coefficients:
  (Intercept)           x1           x2
      0.09016      4.74766      0.12710
```

- Rerunning the code yields

```
Call:
lm(formula = y ~ x1 + x2, data = A)

Coefficients:
  (Intercept)           x1           x2
      0.02081      6.08233     -0.53973
```

Example (Two-dimensional linear regression, cont.)

- Increasing the sample size n improves the results but the estimates still vary a lot
- What is the reason for this (unexpected) bad behavior?

- The reason is that the explanatory variables x_1 and x_2 have very high correlation

```
cor(A$x1,A$x2)
[1] 0.9999889
```

- This problem/phenomenon is usually referred to as **multicollinearity**
- Before fitting a multivariate linear model check the correlations of the explanatory variables!
- Only include variables with low correlation in the model!

- Practical problem: Many explanatory variables - which ones are relevant?
- **Overfitting**: Too complicated model; model that contains irrelevant variables.
- Possible way out: Choose model with lowest BIC-value (i.e. the lowest value of the Bayesian Information Criterion).
- In R one can (among others) use the *leaps* package.
- There exist many other strategies, e.g. stepwise regression.

Example (Leaps package against overfitting)

- We generate data with one explained and four explanatory variables.
- The explained variable y does not depend of all four variables.
- Nevertheless we fit a four-dimensional model.

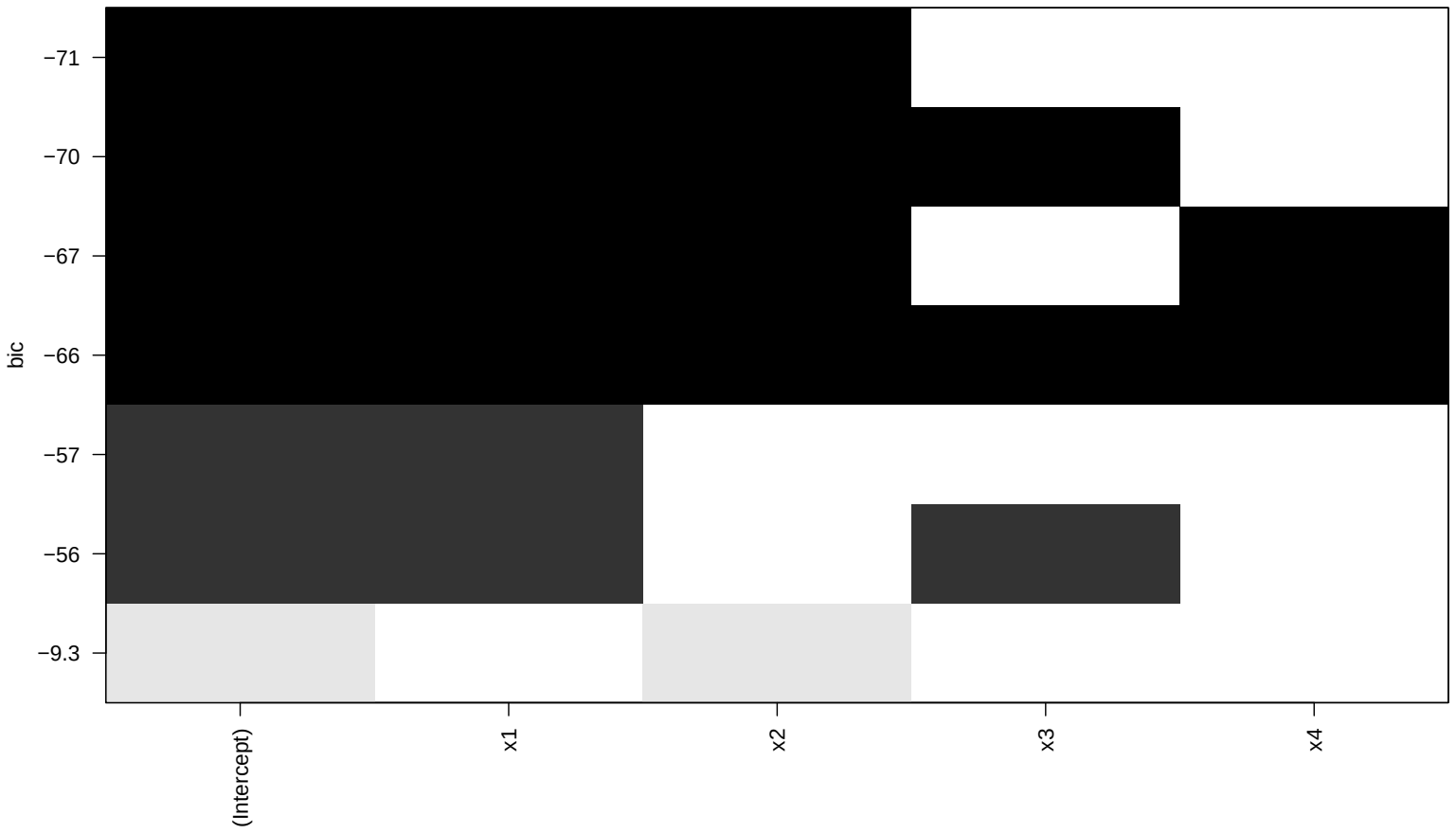
Things to consider

```
n<-101
x1<-runif(n,0,2)
x2<-runif(n,0,2)
x3<-runif(n,0,2)
x4<-runif(n,0,2)

error<-rnorm(n,0,1)
y<-2*x1+x2+error
A<-data.frame(x1=x1,x2=x2,x3=x3,x4=x4,y=y)
model<-lm(data=A,y~x1+x2+x3+x4)
summary(model)

Call:
  lm(formula = y ~ x1 + x2 + x3 + x4, data = A)

Residuals:
   Min      1Q  Median      3Q     Max
-2.47598 -0.59972 -0.01829  0.73862  2.76560

Coefficients:
  Estimate Std. Error t value Pr(>|t|)
  (Intercept)  0.72576    0.31912   2.274   0.0252 *
  x1           1.67958    0.17298   9.710 6.25e-16 ***
  x2           0.73938    0.16818   4.396 2.84e-05 ***
  x3          -0.31001    0.17304  -1.792   0.0764 .
  x4          -0.01027    0.17761  -0.058   0.9540

Residual standard error: 0.9939 on 96 degrees of freedom
Multiple R-squared:  0.5843, Adjusted R-squared:  0.567
F-statistic: 33.74 on 4 and 96 DF,  p-value: < 2.2e-16
```

```
library(leaps)
ov<-regsubsets(y~x1+x2+x3+x4,data=A,nbest=2)
summary(ov)
plot(ov)

Subset selection object
Call: regsubsets.formula(y ~ x1 + x2 + x3 + x4, data = A, nbest = 2)
4 Variables  (and intercept)
Forced in Forced out
x1      FALSE      FALSE
x2      FALSE      FALSE
x3      FALSE      FALSE
x4      FALSE      FALSE
```

- plot(ov) yields the following graphic

Things to consider



- We know that there is a relationship between quantities X and Y of the following form:

$$Y = r(X) + \varepsilon \tag{9}$$

- r is an unknown function and ε is a random error fulfilling $\mathbb{E}(\varepsilon) = 0$.
- Based on observations $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ from (9) we want to determine/estimate the function r.
- If we have a good estimator $\hat{r}$ of r then we can predict Y for arbitrary values of X by considering $\hat{r}(X)$.
- All regression functions considered so far were *parametric*, i.e. r was fully determined by (a fixed number of) parameters.
- Example: $r(x) = ax + b$ (univariate linear gression)
- Example: $r(x) = a_1 x + a_2 x^2 + b$ (univariate quadratic regression)
- Example: $r(x_1, x_2) = a_1 x_1 + a_2 x_2 + b$ (two-dimensional linear regression)

Kernel regression - the basics

- **Problem: In practise not even the a parametric description of r is known**.
- What to do? $\rightarrow$ Use nonparametric techniques: kernel regression, local weighted linear/polynomial regression, etc.
- **(Nadaraya-Watson) Kernregression:** Given data $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$ from the regression model.
- Set

$$\hat{r}_n(x) := \frac{\sum_{i=1}^n K\left(\frac{x-x_i}{h}\right) y_i}{\sum_{i=1}^n K\left(\frac{x-x_i}{h}\right)}, \qquad \text{where} \tag{10}$$

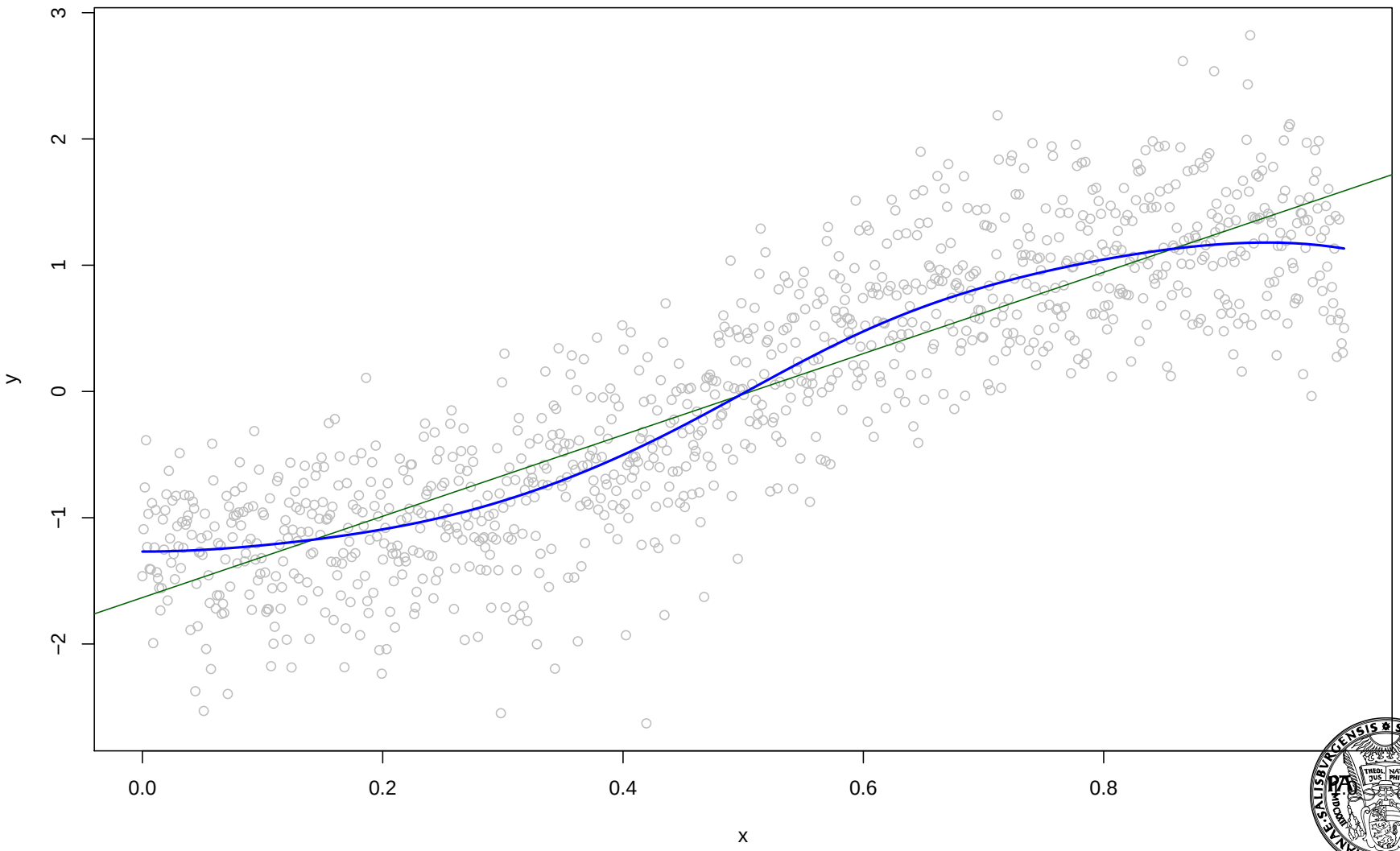
- K...**Kernel**: Probability density, e.g. density of $\mathcal{N}(0,1)$ or density of $\mathcal{U}(-1,1)$.
- $h > 0$...**Bandwidth** (Smoothing parameter)
- x...Point at which the estimator is evaluated/for which the forecast is calculated.
- NB: $\hat{r}_n(x)$ is a weighted mean of the values $y_1, \ldots, y_n$ - the bigger $|x - x_i|$ the less weight has y_i for the calculation of $\hat{r}_n(x)$.

Example (Kernel Regression)

- We load the dataset reg_data.RData and fit a linear regression.
- Additionally, we calculate the kernel regression (estimator).

```
dir <- url("http://www.trutschnig.net/reg_data.RData")
load(dir)
A<-reg_data
head(A)
plot(A, col="gray")
abline(lm(data=A, y~x), col="darkgreen")

library(sm)
nreg<-sm.regression(A$x,A$y, eval.points=c(0.6), display="none")
                    #kernel regression at x=0.6
nreg$estimate
points(0.6, nreg$estimate, col="blue", cex=1)

nreg<-sm.regression(A$x,A$y, eval.points=seq(0,1,length=101),
      display="none")
lines(nreg$eval.points, nreg$estimate, type="l", col="blue", lwd=2)
```

Kernel regression - the basics



Example (Kernel regression, cont.)

- The function sm.regression selects the bandwidth h automatically.
- Nevertheless, h can also be chosen manually.
- Illustration of the effect of the bandwidth h to the kernel regression (estimator) $\rightarrow$ shiny app.
- Conclusion: The smaller h the faster the weights drop with the distance.
- Too big h yields too much smoothing.
- Too small h yields a very shaky regression estimator.
- Calculate R^2 for the calculated regression.

```
nreg<-sm.regression(A$x,A$y,eval.points=A$x,display="none")
res<-A$y-nreg$estimate
R2<-1-var(res)/var(A$y)
R2
```

- The last example was purely descriptive.
- We want to evaluate the quality of kernel regression estimates vis simulations.
- We consider $\hat{r}_n$ a good estimator if it is close to the regression function r uniformly.
- We proceed as in the parametric setting:
  - choose a regression function $r(x)$
  - generate samples $(x_1, y_1), \ldots, (x_n, y_n)$ with $y_i = r(x_i) + \varepsilon_i$
  - calculate the kernel regression $\hat{r}_n$
  - check how close $\hat{r}_n$ and r are

Example (Quality check I)

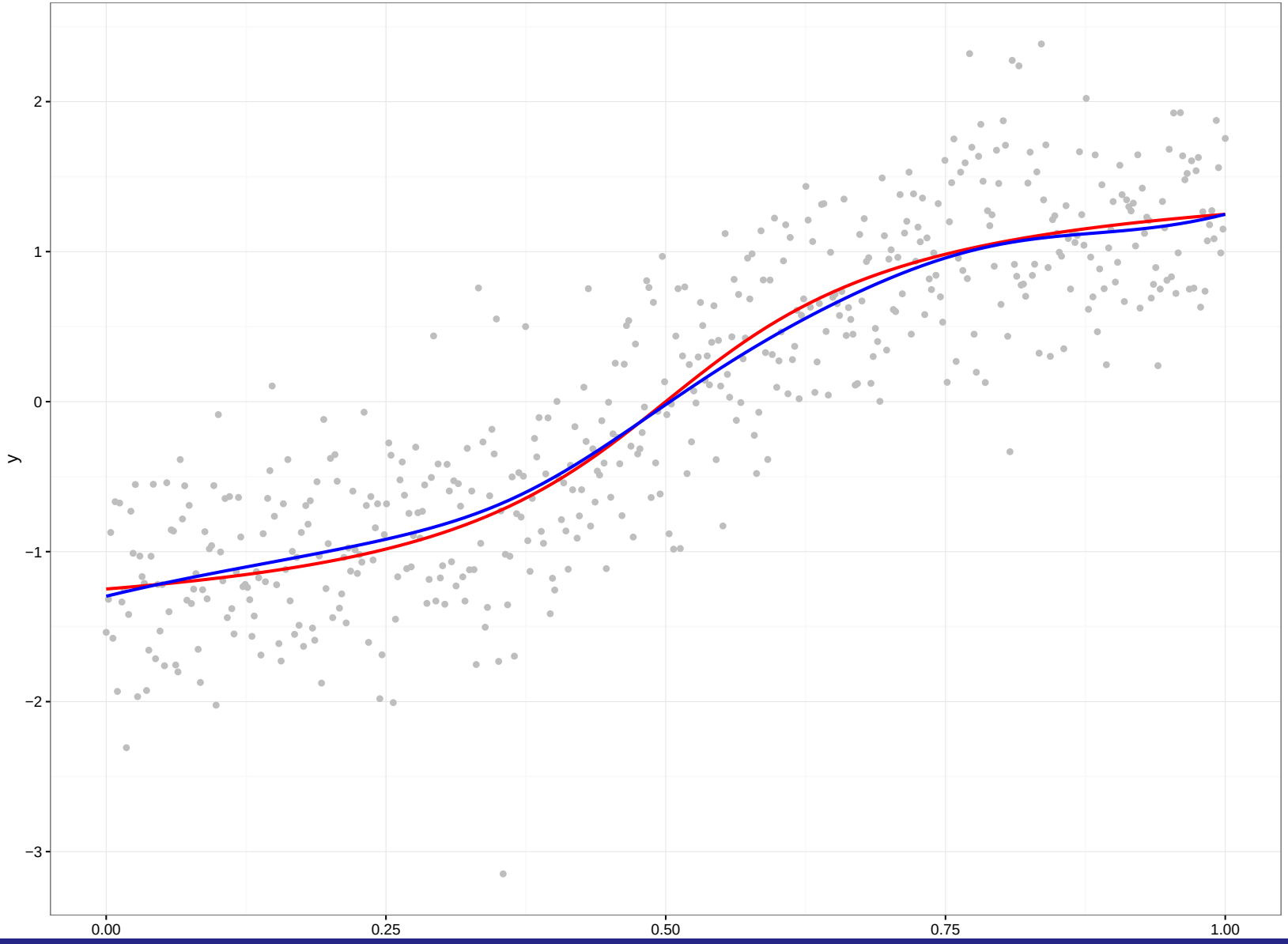
- Model $Y = \arctan(6x - 3) + \varepsilon$
- In other words: the regression function r is given by $r(x) = \arctan(6x - 3)$.
- Use the following R-Code to generate the data and calculate the kernel regression (also contained in R-Codes_Regression05.R)

Example (Quality check I)

```
n <- 500
x <- seq(0,1,length=n)
error <- rnorm(n,0,0.5)
y <- atan(6*x-3)+error
A <- data.frame(x=x,y=y)
plot(A,col="gray")

lines(x,atan(6*x-3),type="l",col="red",lwd=2)
nreg <- sm.regression(A$x,A$y,eval.points=seq(0,1,length=101),
      display="none")
lines(nreg$eval.points,nreg$estimate,type="l",col="blue",lwd=2)
```

- yields

Kernel regression - the basics

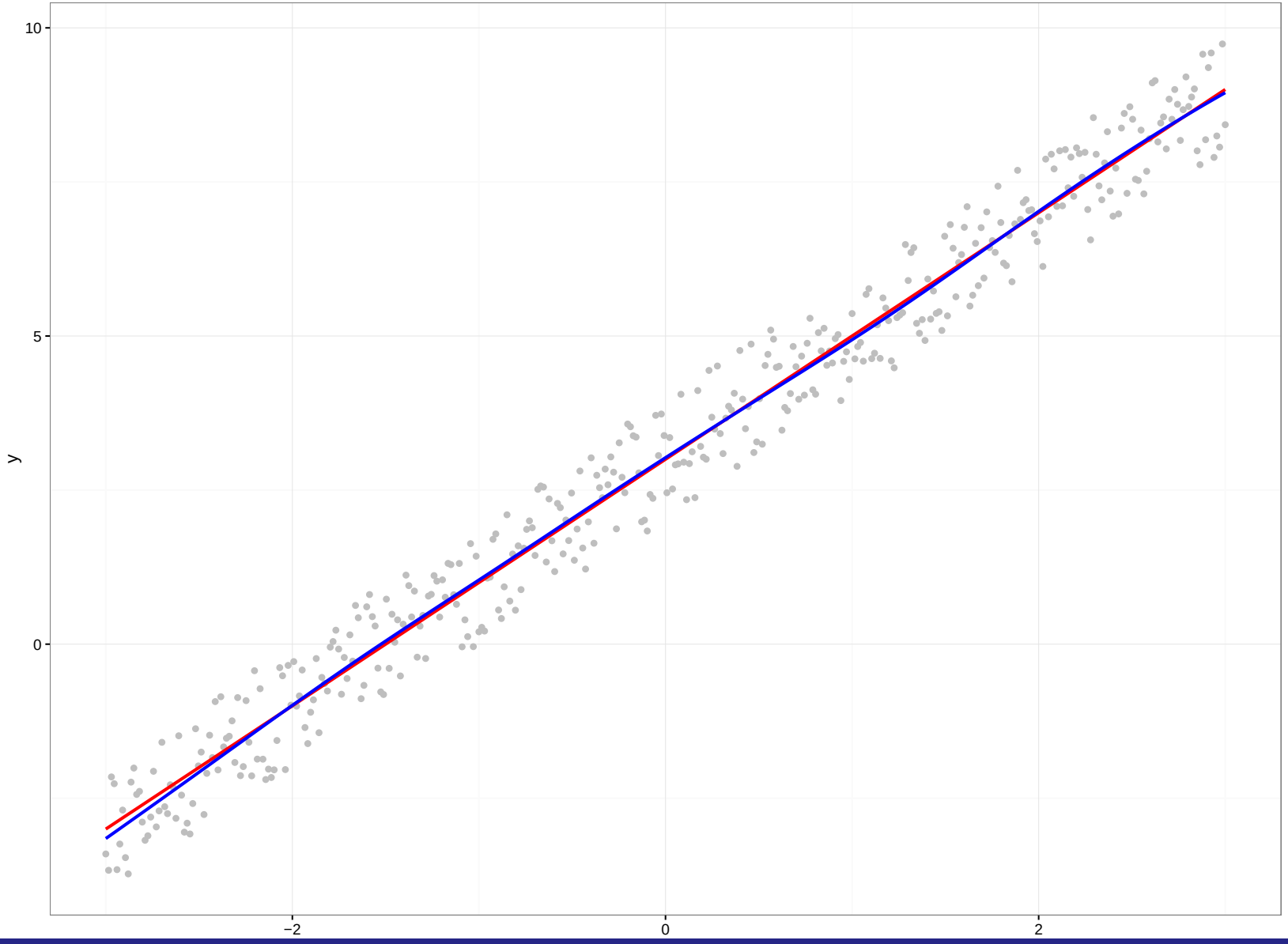


Example (Quality check II)

- Model $Y = 2X + 3 + \varepsilon$
- In other words: the regression function r is given by $r(x) = 2x + 3$.
- Use the following R-Code to generate the data and calculate the kernel regression (also contained in R-Codes_Regression05.R)

```
a <- 2; b <- 3
n <- 400
x <- seq(-3,3,length=n)
y <- a*x+b+runif(n,-1,1)
A <- data.frame(x=x,y=y)
plot(A,col="gray")
abline(3,2,col="red")

nreg <- sm.regression(A$x,A$y,eval.points=A$x,display="none")
lines(nreg$eval.points,nreg$estimate,type="l",col="blue",lwd=2)

R2 <- 1-var(A$y-nreg$estimate)/var(A$y)
R2
```

Kernel regression - the basics

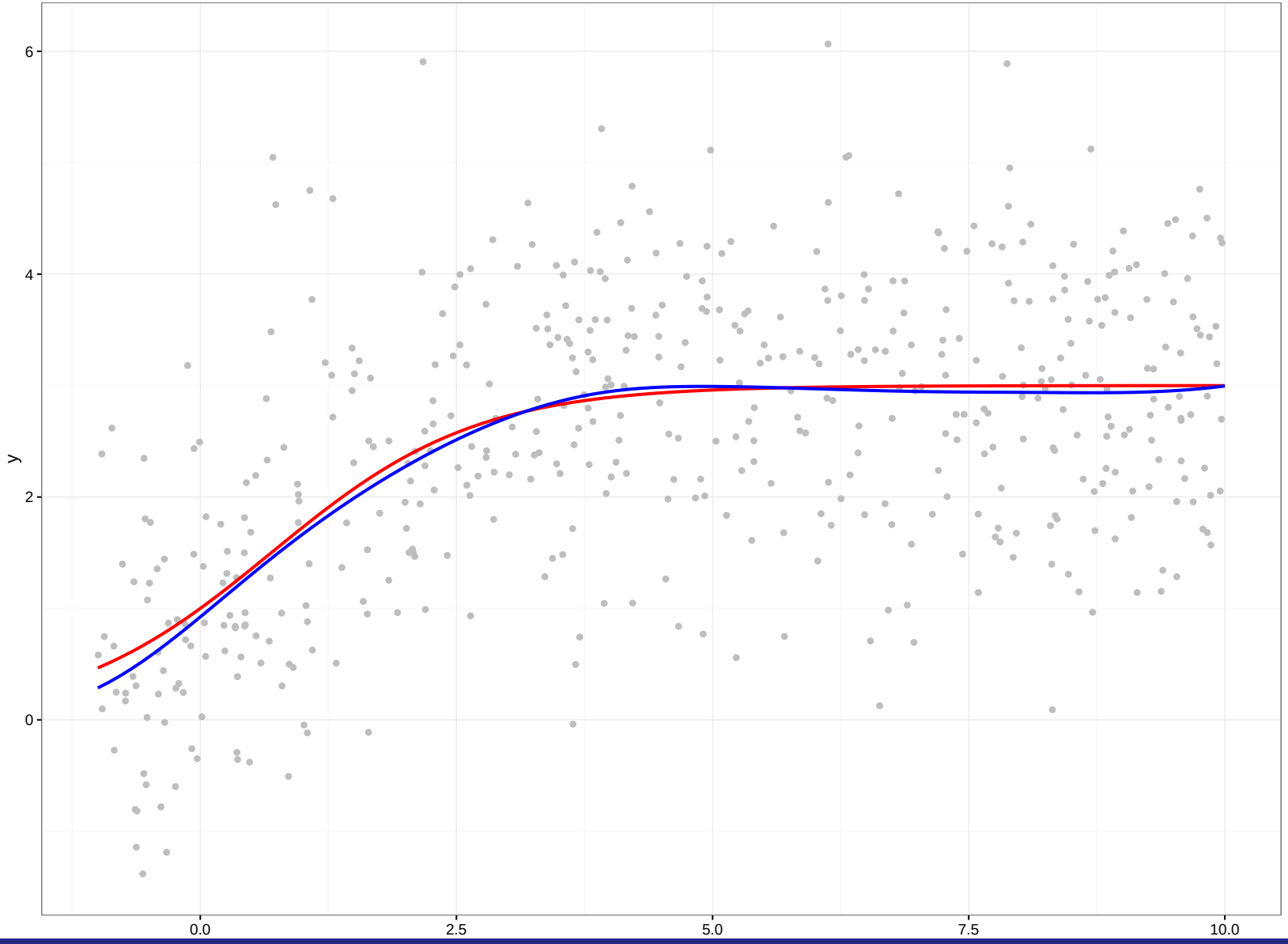


Example (Quality check III)

- Model $Y = \frac{3}{1+2e^{-X}} + \varepsilon$
- In other words: the regression function r is given by $r(x) = \frac{3}{1+2e^{-x}}$.
- Use the following R-Code to generate the data and calculate the kernel regression (also contained in R-Codes_Regression05.R)

```
n <- 500
a <- 3; b<-2
x <- runif(n,-1,10)
error <- rnorm(n,0,1)
r <- function(x){y<-a/(1+b*exp(-x))}
y <- r(x) + error
A <- data.frame(x=x,y=y)
plot(A,col="gray")
xg <- seq(-1,10,length=101)
lines(xg,f(xg),col="red")

nreg<-sm.regression(A$x,A$y,eval.points=xg,display="none")
lines(nreg$eval.points,nreg$estimate,type="l",col="blue",lwd=2)
```

Kernel regression - the basics



Summary:

- Kernel regression seems to yield good results for sufficiently big large sample size n.
- Convergence to the true regression function (convergence of $\hat{r}_n$ to $r(x)$) can also be proved mathematically (strongly consistent estimator in L^1)
- Kernel regression also works in dimensions two and three, in higher dimensions things get more complicated (sample size, computing time).
- It often makes sense to play with the bandwidth h and observe what happens (i.e. not only to use the implemented bandwidth h).
- Concluding example in R-shiny.

Exercise 12: Substitute the linear regression in the quality check II example (lines 67-83 in R-Codes_Regression05.R) by a quadratic regression (with parameters of your choice) and check if the kernel regression also detects the quadratic regression function. Summarize the most important observations in a knitR report featuring the following elements:

- The true regression function $r = b + a_1 x + a_2 x^2$ (with your concrete choice of the parameters b, a_1, a_2).
- A graphic depicting the sample, the true regression function and the kernel regression for sample size $n = 200$.
- A graphic depicting the sample, the true regression function and the kernel regression for sample size $n = 2000$.
- Calculate $\hat{r}_n(x)$ for $x \in \{-1, -0.9, \ldots, 0.9, 1\}$ and include the forecasts in a table.
- Report what happens if the variance of the errors is increased.

Exercise 13: The data set SBP.RData contains the following data for 8.000 patients: age, BMI (body mass index), SBP (systolic blood pressure). Using 104-114 lines in R-Codes_Regression05.R three age groups of patients are built.

To do list:

- Using kernel regression estimate the regression function r with $SBP = r(BMI)$ individually for each of the three age groups.
- For each of the three age groups predict the (average) SBP-value for a patient having BMI 25 and a patient having BMI 30.
- For each of the three groups calculate the percentage increase from BMI 25 to BMI 30.
- Summarize the most important observations as well as a quick summary of the data (how many patients in each group, range of BMI, etc.) in a knitR report.