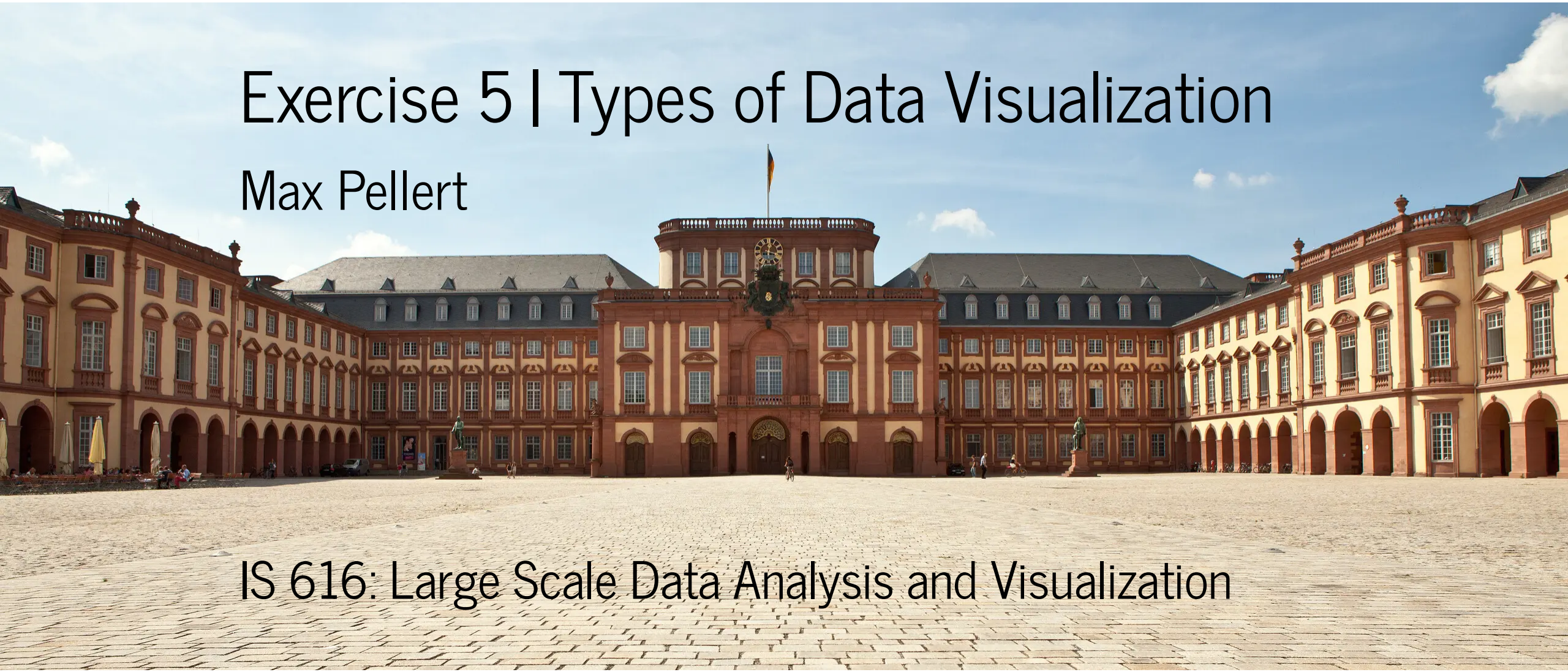
Exercise 5 | Types of Data Visualization

Max Pellert



IS 616: Large Scale Data Analysis and Visualization

Exam Examples (not final!)

1 General Questions

6 points

Which of the following statements are true? *1 point per question*

1 Single Choice Questions

10 points

Given are the following 10 statements. For each statement, indicate whether it is *True* or *False* and justify your answer in one sentence.

A scatter plot is a technique to visualize one-dimensional data. –> False, scatter plots are used for bi-variate data, i.e. they are two-dimensional.

The oldest examples of visualizations in human history are maps. –> True, maps go back at least to the Ptolemaic Kingdom in today's Egypt in 200BC.

Exam Examples (not final!)

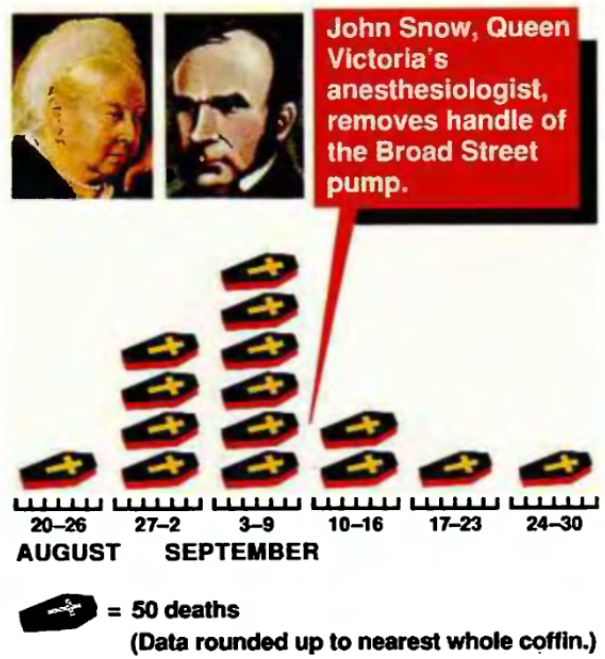
Task 4. Data visualization — 4 points

The code snippet provided below creates a plot. Your task is to identify which symbol (if any) will be present at the given coordinates. If no symbol is found at the specified coordinates, write "None".

Such code snippets will be provided both for Python and for R

It could also be the other way round, that I give you some lines of code (again both for Python and R) that you have to complete to arrive at a certain specified outcome

Exam Examples (not final!)



Name three points to improve this visualization by referring to principles that we covered in the course (1-2 sentences max per point)

Exam Examples (not final!)

Person	Income
1	20.000 US$
2	150.000 US$
3	40.000 US$
4	55.000 US$
…	…

Describe the format of the data (numeric, ordinal, …), what type of visualization you want to use for that data and why (be concise, max 4-6 sentences!) and provide a simple sketch

Exam Examples (not final!)

religion	<$10k	$10–20k	$20–30k	$30–40k	$40–50k	$50–75k
Agnostic	27	34	60	81	76	137
Atheist	12	27	37	52	35	70
Buddhist	27	21	30	34	33	58
Catholic	418	617	732	670	638	1116
Don't know/refused	15	14	15	11	10	35
Evangelical Prot	575	869	1064	982	881	1486
Hindu	1	9	7	9	11	34
Historically Black Prot	228	244	236	238	197	223
Jehovah's Witness	20	27	24	24	21	30
Jewish	19	19	25	25	30	95

Table 4: The first ten rows of data on income and religion from the Pew Forum. Three columns, $75–100k, $100–150k and >150k, have been omitted.

Is that data tidy and ready to use for the proposed visualization? If not, give a brief visual sketch how you would re-arrange it for the task, which operations you need for that and justify your reasoning (be concise, max. 4-6 sentences!)

Replicating Anscombes Quartett

In R: https://rpubs.com/debosruti007/anscombeQuartet

In Python:
https://matplotlib.org/stable/gallery/specialty_plots/anscombe.html

R Libraries

```r
# Start by running this code chunk to make sure you have all the librarie

list.of.packages <- c("datasets","ggplot2","fBasics","grid","gridExtra")

new.packages <- list.of.packages[!(list.of.packages %in% installed.packa

if(length(new.packages)) install.packages(new.packages,repos="https://cl

# to install from github
# if(!("" %in% installed.packages()[,"Package"])){
#  remotes::install_github("")
# }

# if your into problems, it can be helpful to update all packages, by ru
# install.packages(list.of.packages)

# or a specific one
```

R libraries

```
for(each in list.of.packages){
  library(each, character.only = T)
}

# or
# sapply(list.of.packages, library, character.only = TRUE)
```

Data

```
anscombe <- datasets::anscombe

anscombe
```

```
##    x1 x2 x3 x4    y1   y2    y3    y4
## 1  10 10 10  8  8.04 9.14  7.46  6.58
## 2   8  8  8  8  6.95 8.14  6.77  5.76
## 3  13 13 13  8  7.58 8.74 12.74  7.71
## 4   9  9  9  8  8.81 8.77  7.11  8.84
## 5  11 11 11  8  8.33 9.26  7.81  8.47
## 6  14 14 14  8  9.96 8.10  8.84  7.04
## 7   6  6  6  8  7.24 6.13  6.08  5.25
## 8   4  4  4 19  4.26 3.10  5.39 12.50
## 9  12 12 12  8 10.84 9.13  8.15  5.56
## 10  7  7  7  8  4.82 7.26  6.42  7.91
## 11  5  5  5  8  5.68 4.74  5.73  6.89
```

Same Statistics

```
fBasics::basicStats(anscombe)
```

```
##                     x1        x2        x3        x4        y1
## nobs         11.000000 11.000000 11.000000 11.000000 11.000000 11.0000
## NAs           0.000000  0.000000  0.000000  0.000000  0.000000  0.0000
## Minimum       4.000000  4.000000  4.000000  8.000000  4.260000  3.1000
## Maximum      14.000000 14.000000 14.000000 19.000000 10.840000  9.2600
## 1. Quartile   6.500000  6.500000  6.500000  8.000000  6.315000  6.6950
## 3. Quartile  11.500000 11.500000 11.500000  8.000000  8.570000  8.9500
## Mean          9.000000  9.000000  9.000000  9.000000  7.500909  7.5009
## Median        9.000000  9.000000  9.000000  8.000000  7.580000  8.1400
## Sum          99.000000 99.000000 99.000000 99.000000 82.510000 82.5100
## SE Mean       1.000000  1.000000  1.000000  1.000000  0.612541  0.6125
## LCL Mean      6.771861  6.771861  6.771861  6.771861  6.136083  6.1360
## UCL Mean     11.228139 11.228139 11.228139 11.228139  8.865735  8.8657
## Variance     11.000000 11.000000 11.000000 11.000000  4.127269  4.1276
## Stdev         3.316625  3.316625  3.316625  3.316625  2.031568  2.0316
## Skewness      0.000000  0.000000  0.000000  2.466911 -0.048374 -0.9786
## Kurtosis     -1.528926 -1.528926 -1.528926  4.520661 -1.199123 -0.5143
```

Statistics more in detail

```
# Mean
sapply(1:8, function(x) mean(anscombe[ , x]))
```

```
## [1] 9.000000 9.000000 9.000000 9.000000 7.500909 7.500909 7.500000 7.
```

```
# Variance
sapply(1:8, function(x) var(anscombe[ , x]))
```

```
## [1] 11.000000 11.000000 11.000000 11.000000  4.127269  4.127629  4.12
## [8]  4.123249
```

Statistics more in detail

```
# Correlation
round(sapply(1:4, function(x) cor(anscombe[ , x], anscombe[ , x+4])),2)
```

```
## [1] 0.82 0.82 0.82 0.82
```

```
sapply(1:4, function(x) cor(anscombe[ , x], anscombe[ , x+4]))
```

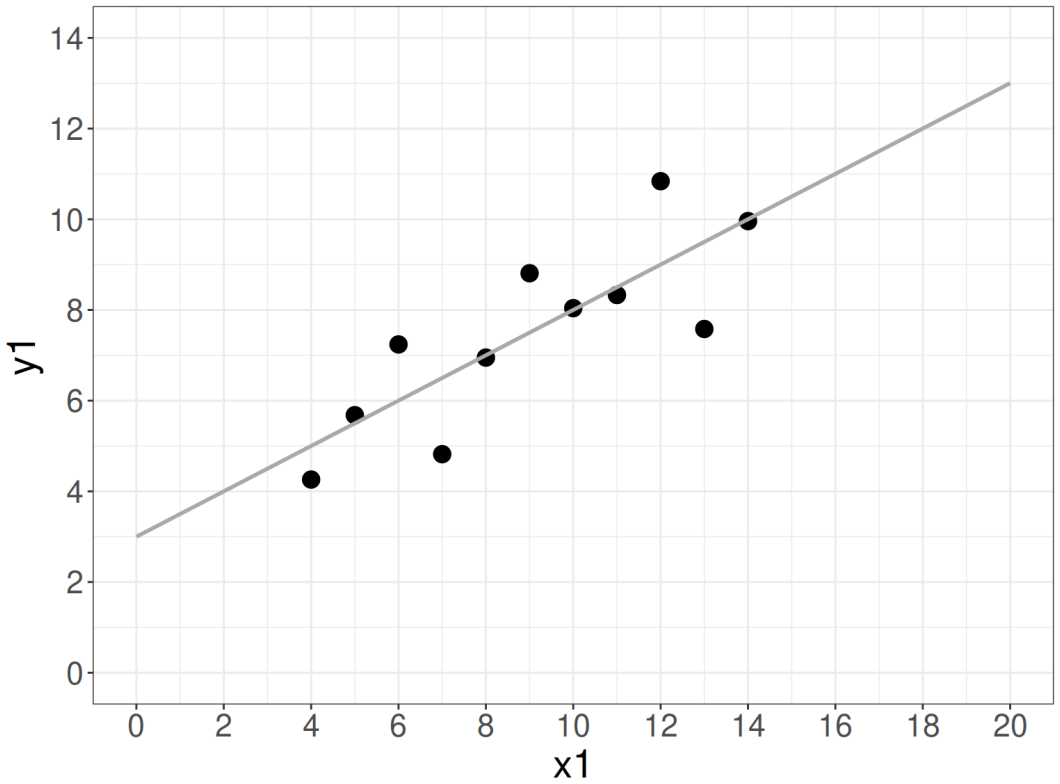
```
## [1] 0.8164205 0.8162365 0.8162867 0.8165214
```

Plotting function

```
create_plot <- function(dataset_x,dataset_y,size_points=4,size_text=21){
  ggplot(anscombe,
         aes({{ dataset_x }},{{ dataset_y }})) +
  geom_point(
    size = size_points) +
  geom_smooth(method="lm", se=F, fullrange = TRUE,
              color="darkgrey") +
  scale_x_continuous(
    breaks = seq(0,20,2)) +
  scale_y_continuous(
    breaks = seq(0,14,2)) +
  expand_limits(x = c(0,20), y = c(0,14)) +
  labs(x = deparse(substitute(dataset_x)),
       y = deparse(substitute(dataset_y))) +
  theme_bw() +
  theme(text=element_text(size=size_text))
}
```

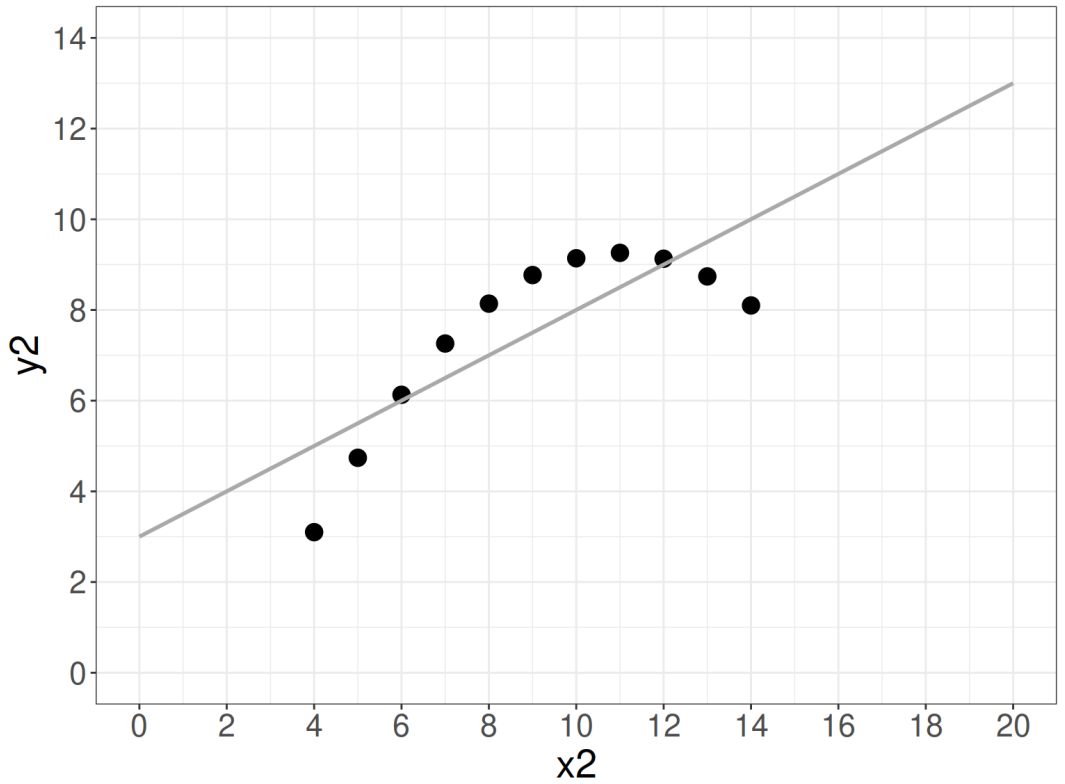
First scatter plot

```
create_plot(x1, y1)
```



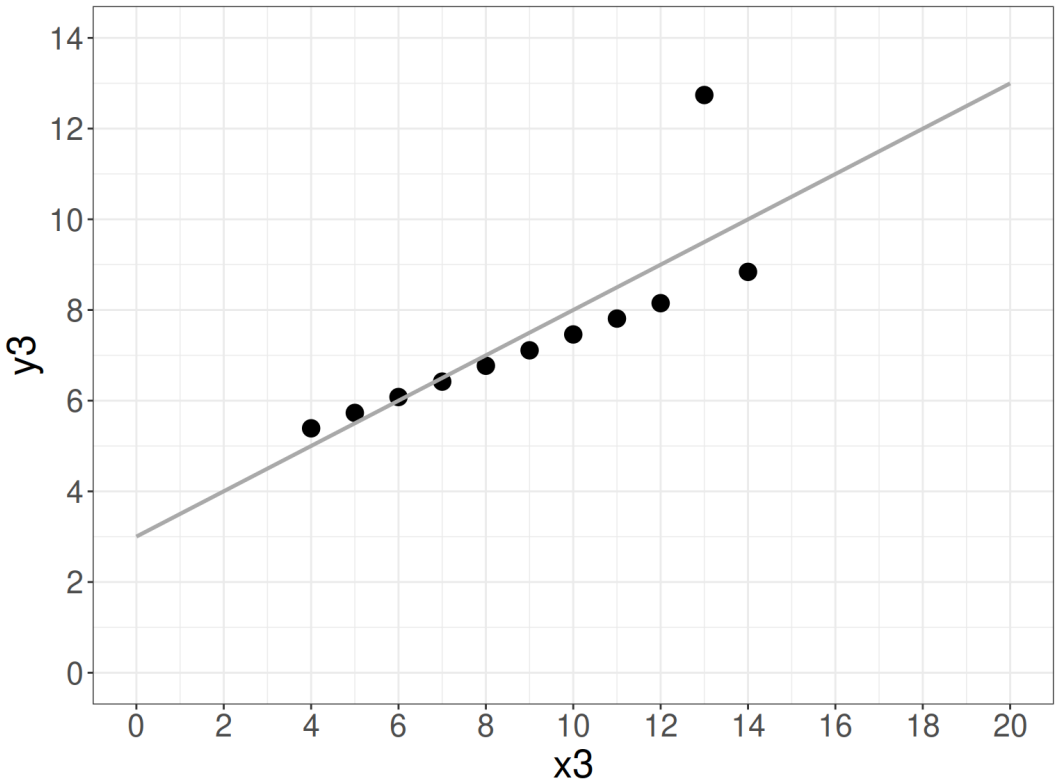
Second scatter plot

```
create_plot(x2, y2)
```



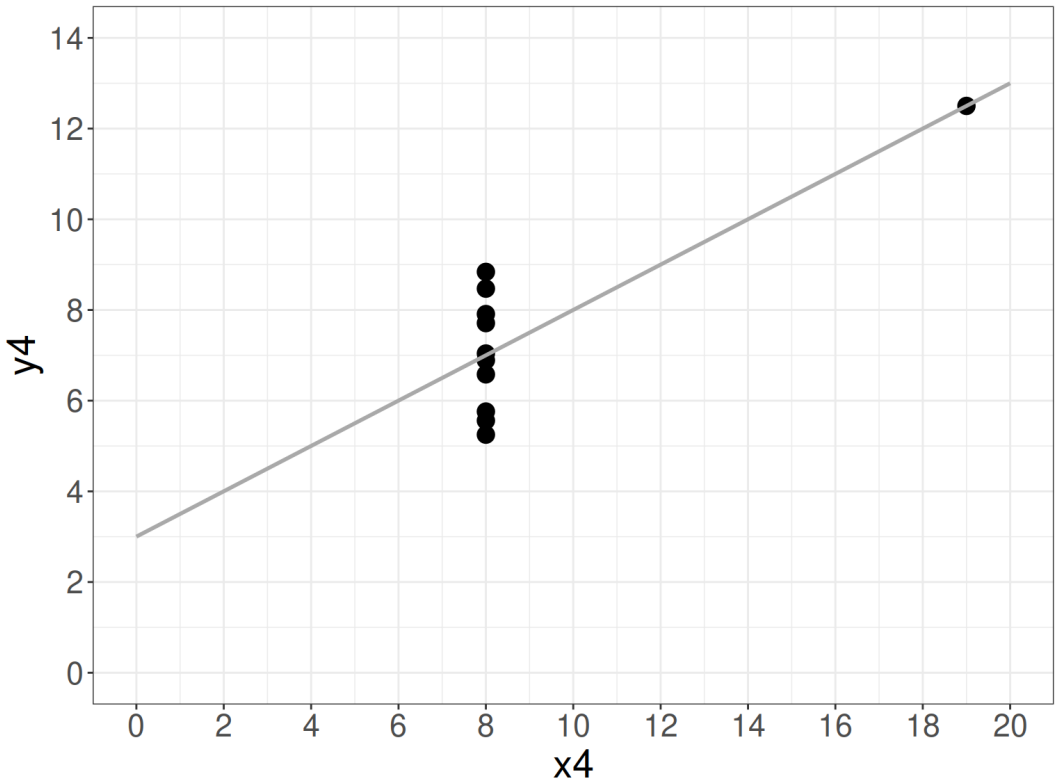
Third scatter plot

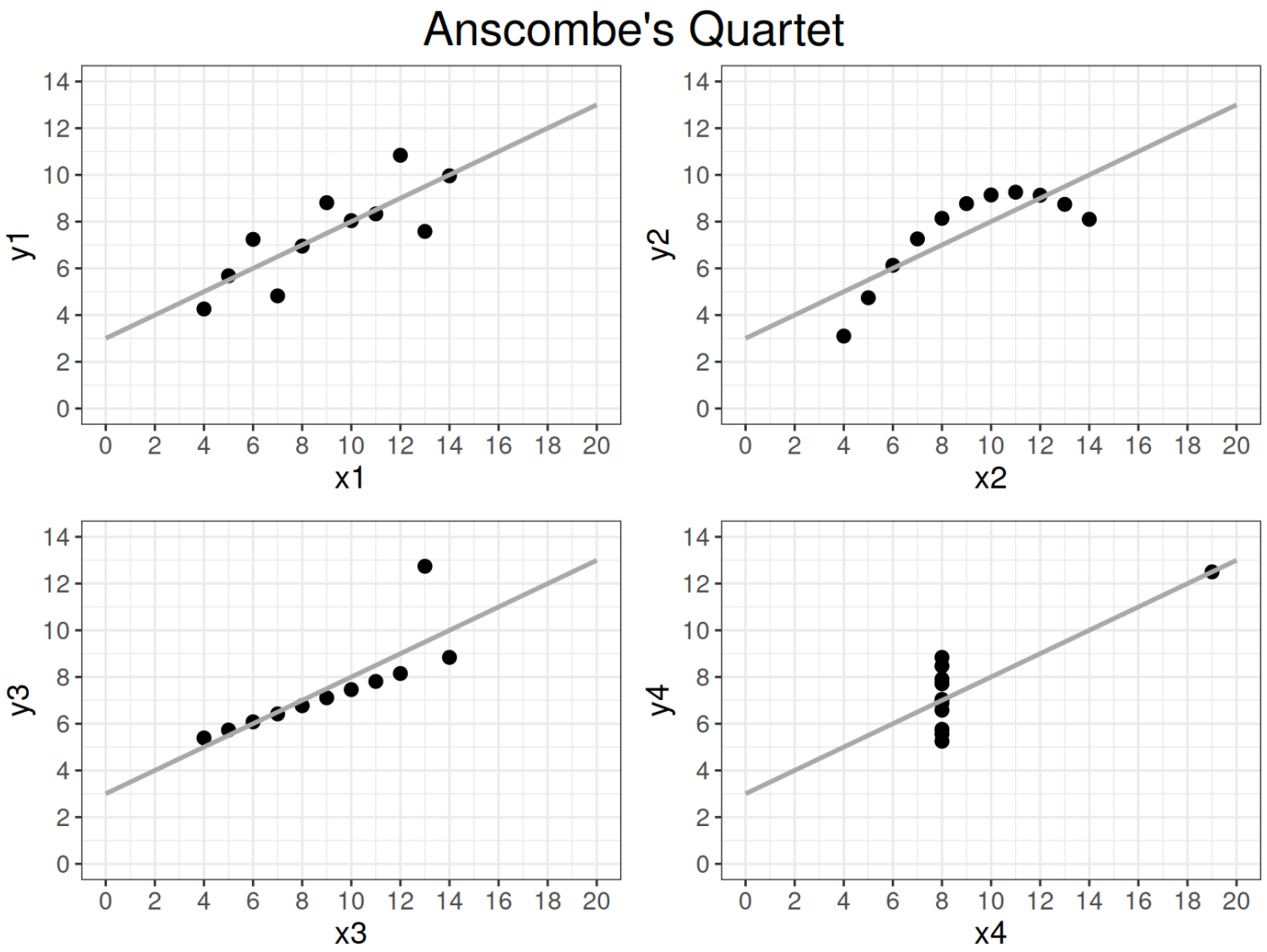
```
create_plot(x3, y3)
```



Fourth scatter plot

```
create_plot(x4, y4)
```




Anscombe's Quartet
y1
x1
y2
x2
y3
x3
y4
x4

Combining all four plots

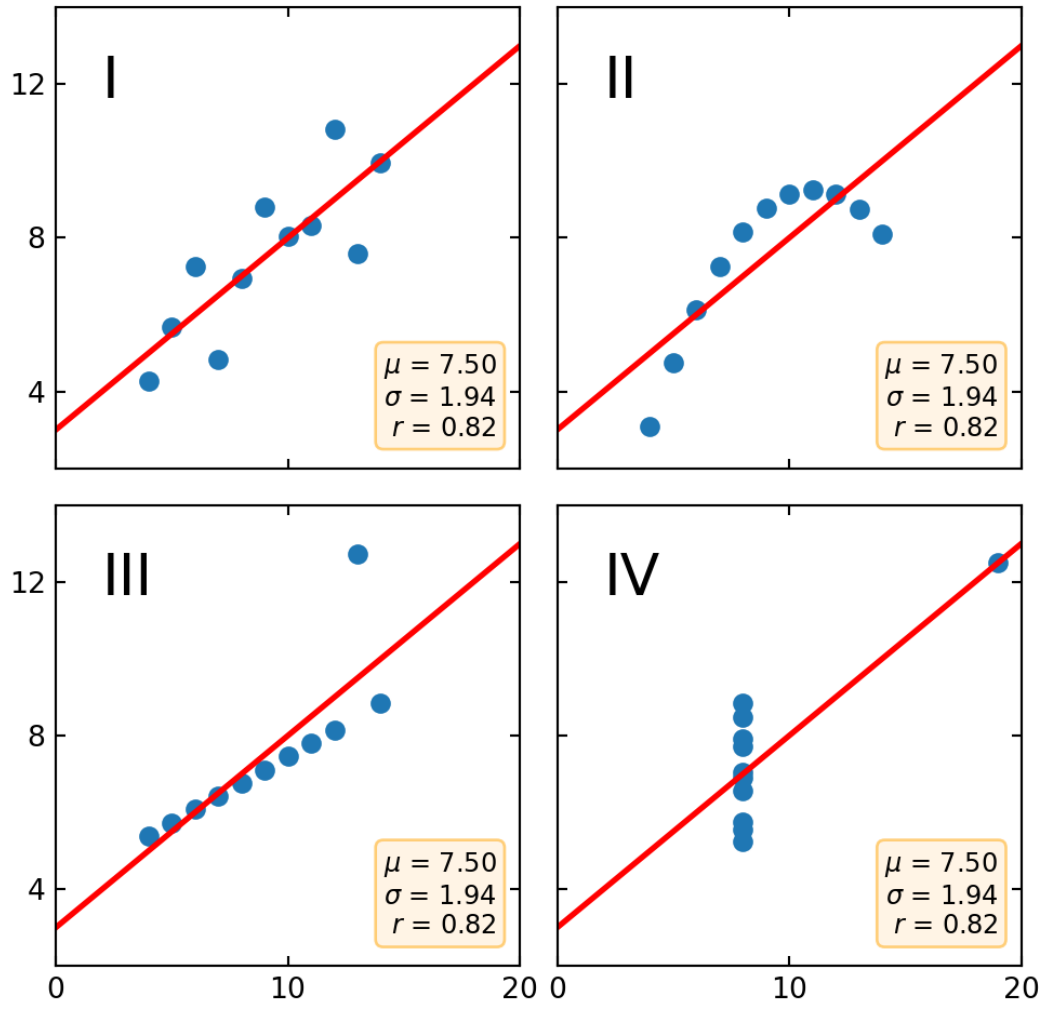
```
size_points <- 2.5

size_text <- 14

grid.arrange(grobs = list(
  create_plot(x1,y1,size_points,size_text),
  create_plot(x2,y2,size_points,size_text),
  create_plot(x3,y3,size_points,size_text),
  create_plot(x4,y4,size_points,size_text)),
            ncol = 2,
            top = textGrob("Anscombe's Quartet",
                           gp=gpar(fontsize=21, font=8)))
```

Python libraries and data

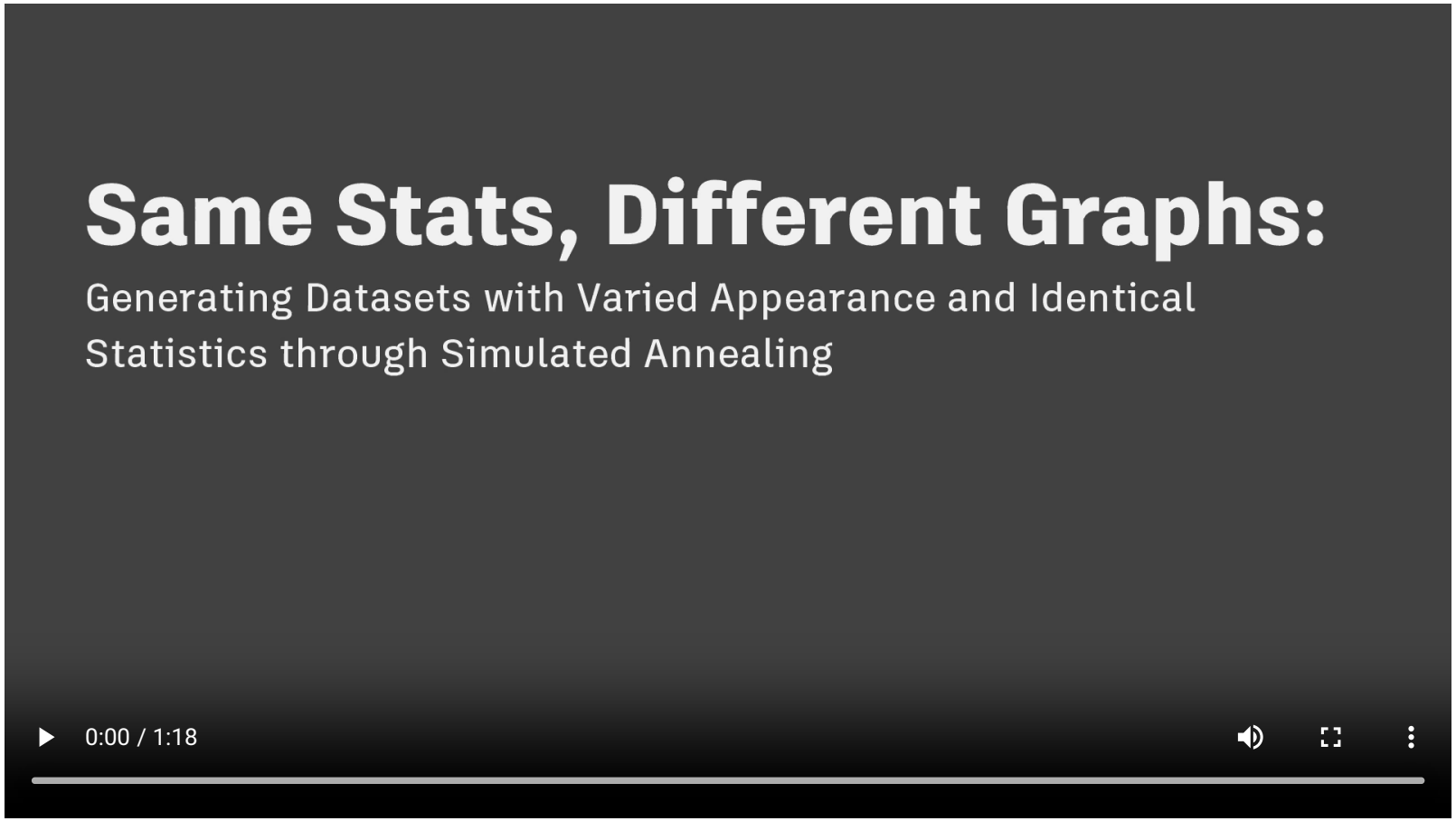
```
import matplotlib.pyplot as plt
import numpy as np

x = [10, 8, 13, 9, 11, 14, 6, 4, 12, 7, 5]
y1 = [8.04, 6.95, 7.58, 8.81, 8.33, 9.96, 7.24, 4.26, 10.84, 4.82, 5.68]
y2 = [9.14, 8.14, 8.74, 8.77, 9.26, 8.10, 6.13, 3.10, 9.13, 7.26, 4.74]
y3 = [7.46, 6.77, 12.74, 7.11, 7.81, 8.84, 6.08, 5.39, 8.15, 6.42, 5.73]
x4 = [8, 8, 8, 8, 8, 8, 8, 19, 8, 8, 8]
y4 = [6.58, 5.76, 7.71, 8.84, 8.47, 7.04, 5.25, 12.50, 5.56, 7.91, 6.89]

datasets = {
    'I': (x, y1),
    'II': (x, y2),
    'III': (x, y3),
    'IV': (x4, y4)
}
```

matplotlib

```
fig, axs = plt.subplots(2, 2, sharex=True, sharey=True, figsize=(6, 6),
                        gridspec_kw={'wspace': 0.08, 'hspace': 0.08})
axs[0, 0].set(xlim=(0, 20), ylim=(2, 14))
axs[0, 0].set(xticks=(0, 10, 20), yticks=(4, 8, 12))

for ax, (label, (x, y)) in zip(axs.flat, datasets.items()):
    ax.text(0.1, 0.9, label, fontsize=20, transform=ax.transAxes, va='to
    ax.tick_params(direction='in', top=True, right=True)
    ax.plot(x, y, 'o')

    # linear regression
    p1, p0 = np.polyfit(x, y, deg=1)  # slope, intercept
    ax.axline(xy1=(0, p0), slope=p1, color='r', lw=2)

    # add text box for the statistics
    stats = (f'$\\mu$ = {np.mean(y):.2f}\n'
             f'$\\sigma$ = {np.std(y):.2f}\n'
```


I
II
III
IV
$\mu = 7.50$
$\sigma = 1.94$
$r = 0.82$
12
8
4
0
10
20

Going beyond the quartett



Until next week

Read Chapter 1 of

Tufte, E. R. (2001). The visual display of quantitative information (2nd ed.). Graphics Press.

that is provided to you

Acknowledgements

https://www.youtube.com/watch?v=DbJyPELmhJc