Einführung mit Fokus auf Statistik

Etappen im Analyseablauf

1. *Definition der Arbeitsumgebung*
2. *Import der Daten*
3. *Transformationen der Daten*
4. *Exploration (allgemein/Extremwerte/Verteilung) (zurück auf 3?)*
5. *Klassifikation von Skalenniveaus / Verteilungen (basierend auf 3/4)*
6. *Beschreibende Statistik*
7. *Test / Modellierung (kann Schritt 6 beinhalten)*
8. *Reporterstellung*

Definition der Arbeitsumgebung

- *Aktivierung benötigter Pakete: library() / pacman::p_load()*
- *ggplot Themen: theme_set() / theme_update()*
- *flextable Einstellungen: set_flextable_defaults()*
- *knitr::opts_chunk$set()*

```
if(!requireNamespace("pacman")){install.packages("pacman")}
pacman::p_load(conflicted,tidyverse,wrappedtools, readxl,car, flextable,
               ggbeeswarm, ggsignif, ggridges, patchwork,ggrepel, easystats)

conflicts_prefer(dplyr::filter,dplyr::select)
theme_set(theme_light(base_size = 20))
gdtools::register_gfont('Roboto')# Mono')
```

```
[1] TRUE
```

```
set_flextable_defaults(
  theme_fun = theme_zebra, font.size = 18, font.family = 'Roboto',
  table.layout = 'autofit',
  padding.bottom = .2, padding.top = .2, padding.left = 2, padding.right = 2)

knitr::opts_chunk$set(message = FALSE, warning = FALSE, comment = NA, echo = T
```


Import

- *read_xlsx() / read_csv() / read_csv2()*
- Optionen beziehen sich auf Trennzeichen, Zahlenformate, Bereiche etc.
- *rename() / rename_with()*

```
rawdata <- read_excel('Data/DOC-20230130-WA0000_.xlsx',
                      sheet = 1,col_names = TRUE)
```

Erster Blick auf die Daten: Problemsuche

```
head(rawdata,n = 15) |> flextable()|>
  theme_zebra(even_body = 'aquamarine',odd_body = 'antiquewhite')
```

CODE OF CUP	CODE OF SAMPLE	WEIGHT OF EMPTY ALUMINUM(wt)	WEIGTH OF ALMINIUM CUP + SAMPLE (Wt + s)	WEIGTH OF ALUMINIUM CAP + SAMPLE AFTER DRYING (Wt-AL +s+d)	weight of sample before drying (Wts)	weight of sample after drying (Wts+d)	MOISTURE CONTENT (%)
69	D	4.1974	9.3865	4.7000	5.1891	4.6865	90.31431
	D	4.1964	9.2734	4.4670	5.0770	4.8064	94.67008
A	D	4.2108	9.2653	4.6670	5.0545	4.5983	90.97438
114	D	4.2134	9.3146	4.6345	5.1012	4.6801	91.74508
M1	D	4.1856	9.3147	4.6171	5.1291	4.6976	91.58722
a/17	D	4.2090	9.3204	4.5661	5.1114	4.7543	93.01366
8	D	4.1894	9.2661	4.5778	5.0767	4.6883	92.34936
33	D	4.1968	9.2880	4.6057	5.0912	4.6823	91.96849
M	D	4.1535	9.2872	4.6350	5.1337	4.6522	90.62080
E/18/1	D	4.2534	9.2476	4.7403	4.9942	4.5073	90.25069
24/A2	D	4.2066	8.3463	4.5849	4.1397	3.7614	90.86166
13	A	4.1554	9.2384	4.7402	5.0830	4.4982	88.49498
Xp	A	4.1893	9.2495	4.7381	5.0602	4.5114	89.15458
2p/029	A	4.0654	9.2173	4.6940	5.1519	4.5233	87.79868
15	A	4.0641	9.2032	4.8124	5.1391	4.3908	85.43908

Umbenennen von Variablen

```
colnames(rawdata)
```

```
[1] "CODE OF CUP"
[2] "CODE OF SAMPLE"
[3] "WEIGHT OF EMPTY ALUMINUM(wt)"
[4] "WEIGTH OF ALMINIUM CUP + SAMPLE (Wt + s)"
[5] "WEIGTH OF ALUMINIUM CAP + SAMPLE  AFTER DRYING (Wt-AL +s+d)"
[6] "weight of sample before drying (Wts)"
[7] "weight of sample after drying (Wts+d)"
[8] "MOISTURE CONTENT (%)"
```

```
rawdata <- rawdata |>
  rename(Region=`CODE OF SAMPLE`) |>
  rename_with(.fn = ~str_replace_all(
    .,
    c("GTH"="GHT", 'AL.+UM'= 'Cup',
      'C[UA]P' = 'Cup','\\(\\w+.*\\)'='',
      'Cup Cup'='Cup','  '=' ')) |>
      str_to_title() |> str_trim())
cn()
```

```
[1] "Code Of Cup"                          "Region"
[3] "Weight Of Empty Cup"                  "Weight Of Cup + Sample"
[5] "Weight Of Cup + Sample After Drying" "Weight Of Sample Before Drying"
[7] "Weight Of Sample After Drying"        "Moisture Content (%)"
```

Transformationen

- *Ändern oder Erzeugen von Spalten mit mutate() / mutate(across())*
- *z.B. für log-Transformation, Erzeugen von Faktoren, Text -Umkodierung*

```
rawdata <- rawdata |>
  mutate(
    `Code Of Cup`=case_when(
      is.na(`Code Of Cup`)~paste("sample",row_number()),
      .default = `Code Of Cup`),
    `Weight Of Sample After Drying`=`Weight Of Cup + Sample After Drying`-
      `Weight Of Empty Cup`,
    `Dry Content (%)`=`Weight Of Sample After Drying`*100/
      `Weight Of Sample Before Drying`,
    `Moisture Content (%)`=100-`Dry Content (%)`)
```

Code Of Cup	Region	Weight Of Empty Cup	Weight Of Cup + Sample	Weight Of Cup + Sample After Drying	Weight Of Sample Before Drying	Weight Of Sample After Drying	Moisture Content (%)	Dry Content (%)
69	D	4.1974	9.3865	4.7000	5.1891	0.5026	90.31431	9.685687
sample 2	D	4.1964	9.2734	4.4670	5.0770	0.2706	94.67008	5.329919
A	D	4.2108	9.2653	4.6670	5.0545	0.4562	90.97438	9.025621
114	D	4.2134	9.3146	4.6345	5.1012	0.4211	91.74508	8.254920
M1	D	4.1856	9.3147	4.6171	5.1291	0.4315	91.58722	8.412782

Exploration / Variablengruppierung

Exploration (allgemein/Extremwerte/Verteilung)

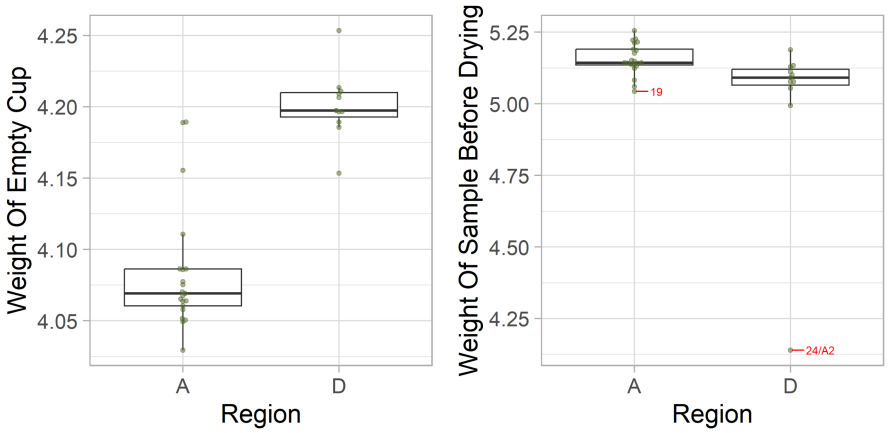
- *ggplot()+geom_boxplot() / geom_beeswarm() / geom_density()*
- *ks.test() / ksnormal() / shapiro.test()*

Klassifikation nach Skalenniveau / Verteilung

- *gaussvars / ordvars / factvars, possibly more…*
- *Speichern von Variablengruppen, z.B. ColSeeker()*

Exploration: Extremwerte

```
p1 <- ggplot(data = rawdata, aes(x = `Region`, y = `Weight Of Empty Cup`))+
  geom_boxplot(outlier.alpha = 0) + #hide outliers, beeswarm will plot them
  geom_beeswarm(alpha=.5, color="darkolivegreen")
p2 <- ggplot(data = rawdata, aes(x = `Region`, y = `Weight Of Sample Before Dr
  geom_boxplot(outlier.alpha = 0) +
  geom_beeswarm(alpha=.5, color="darkolivegreen")+
  geom_text_repel(data=. %>% group_by(Region) %>%
            filter(`Weight Of Sample Before Drying` %in%
                     boxplot.stats(`Weight Of Sample Before Drying`, coef=1.5
          aes(label=`Code Of Cup` , y=`Weight Of Sample Before Drying`),
          nudge_x=0.1, colour="red", size=3, hjust=0)
p1|p2
```

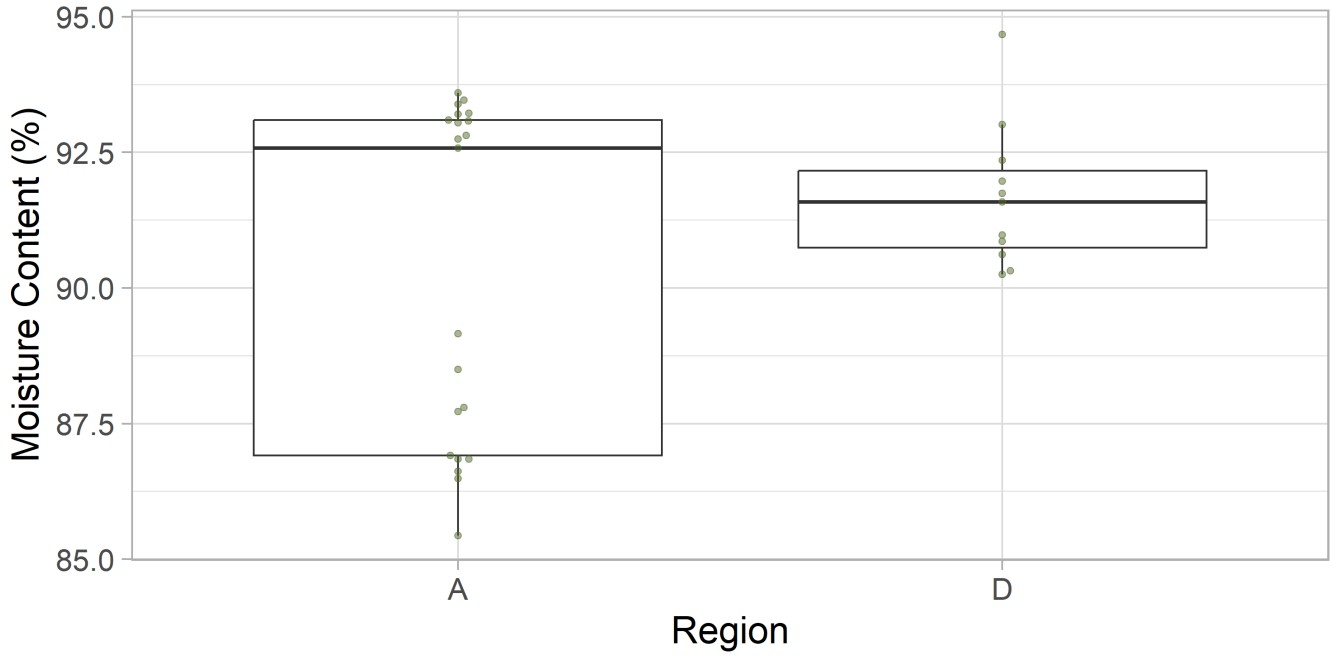


Behandlung Extremwerte/Ausreißer?

Entfernen ist die schlechteste Option, Korrektur von Eingabefehlern, Ändern der Verteilung oder Winsorizierung…

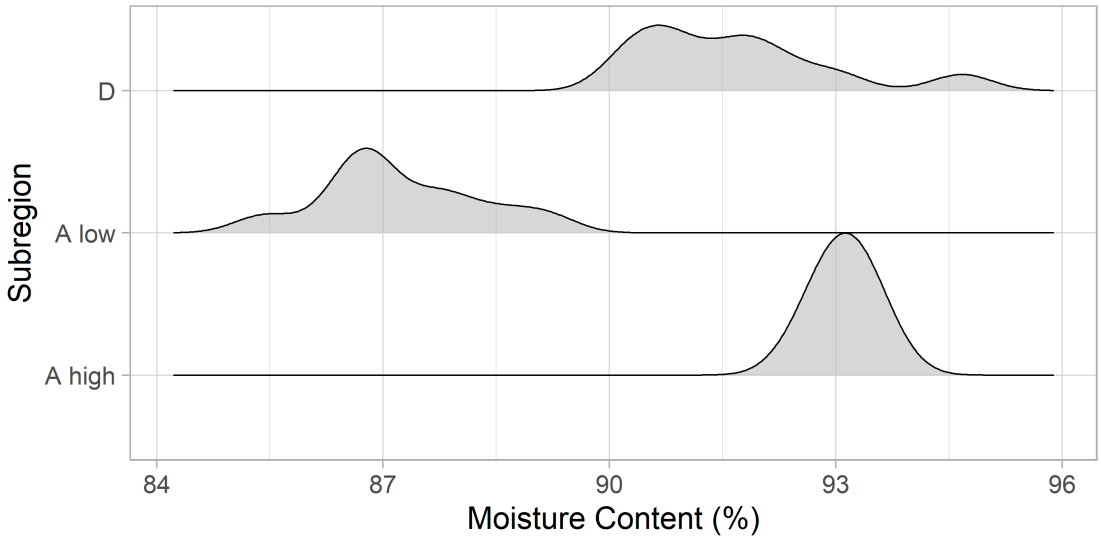
Exploration: Unerwartetes

```r
ggplot(data = rawdata,
       aes(x = `Region`,
           y = `Moisture Content (%)`))+
  geom_boxplot(outlier.alpha = 0) +
  geom_beeswarm(alpha=.5, color="darkolivegreen")
```



Transformation in Subregionen?

```r
rawdata <- mutate(rawdata, Subregion = case_when(
  `Region`=='D' ~ 'D',
  `Region`=='A' & `Moisture Content (%)` > 90 ~ 'A high',
  `Region`=='A' & `Moisture Content (%)` <= 90 ~ 'A low') |>
    factor())
# Test for Region A is redundant here, but more verbose.

ggplot(data = rawdata,aes(x = `Moisture Content (%)`,y=Subregion))+
  geom_density_ridges(alpha=.5, scale=1)
```



Exploration: Normalverteilung 1

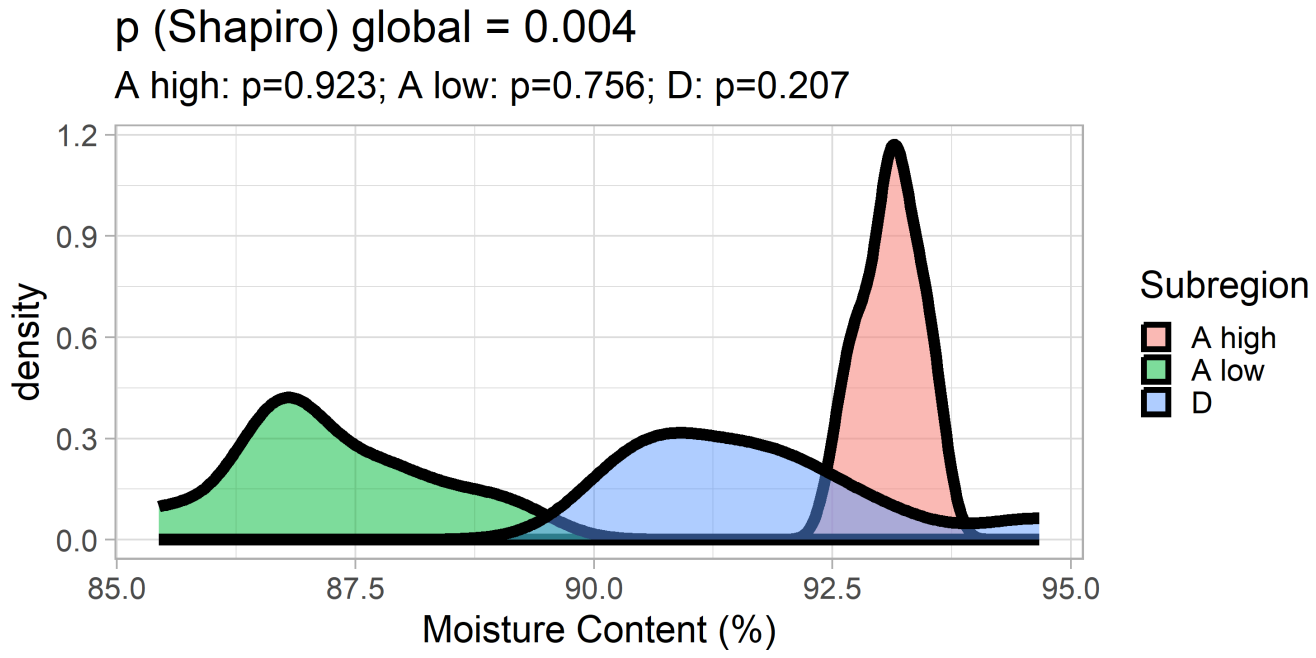
- Gausssche Glockenkurve / Normalverteilung ist Voraussetzung vieler statistischen Verfahren
- Übliche Tests sind graphische Exploration, Shapiro-Wilk-Test Und Kolmogorov-Smirnov-Test

```
p_normal <-
  shapiro.test(x = rawdata$`Moisture Content (%)`)
p_normal
```

```
    Shapiro-Wilk normality test

data:  rawdata$`Moisture Content (%)`
W = 0.89133, p-value = 0.003752
```



```
ggplot(data = rawdata,aes(x = `Moisture Content (%)`,fill=`Subregion`))+
  geom_density(linewidth=3, alpha=.5)+
  labs(title = paste('p (Shapiro) global',
                     formatP(pIn = p_normal$p.value, pretext = T)),
       subtitle = rawdata |> group_by(Subregion) |>
  summarize(pNormal=shapiro.test(`Moisture Content (%)`)$p.value |> formatP())
  unite('p(Normal)',sep = ': p=') |> pull(1) |> paste(collapse = '; '))
```



Exploration: Normalverteilung 2

```r
rawdata |>
  group_by(`Subregion`) |>
  summarize(across(.cols = where(is.numeric),
                   .fns = ~ksnormal(.x) |> #computing p-value
                     formatP(mark = T))) |> #formatting p-value
  pivot_longer(cols = -1, names_to = 'Variable',
               values_to = 'pKS') |> #intermediate, all p-values in 1 column
  pivot_wider(names_from = `Subregion`,
              values_from = pKS) #spreading across subregions
```

```
# A tibble: 7 × 4
  Variable                          `A high`    `A low`     D
  <chr>                             <chr>       <chr>       <chr>
1 Weight Of Empty Cup               0.989 n.s.  0.444 n.s.  0.697 n.s.
2 Weight Of Cup + Sample            0.785 n.s.  0.980 n.s.  0.019 *
3 Weight Of Cup + Sample After Drying 0.900 n.s. 0.710 n.s. 0.969 n.s.
4 Weight Of Sample Before Drying    0.196 n.s.  0.999 n.s.  0.072 +
5 Weight Of Sample After Drying     0.976 n.s.  0.555 n.s.  1.000 n.s.
6 Moisture Content (%)              0.975 n.s.  0.733 n.s.  0.954 n.s.
7 Dry Content (%)                   0.975 n.s.  0.733 n.s.  0.954 n.s.
```


Exploration: Gruppierung der Variablen nach Typ/Verteilung

Skalenniveau bestimmt angemessene Statistiken

Typische Skalenniveaus sind

Type Entscheidung dokumentieren / reproduzierbar

```
gaussvars <- ColSeeker(data=rawdata,namepattern = c('Weight','Content'))
gaussvars
```

```
$index
[1] 3 4 5 6 7 8 9

$names
[1] "Weight Of Empty Cup"                 "Weight Of Cup + Sample"
[3] "Weight Of Cup + Sample After Drying" "Weight Of Sample Before Drying"
[5] "Weight Of Sample After Drying"       "Moisture Content (%)"
[7] "Dry Content (%)"

$bticked
[1] "`Weight Of Empty Cup`"
[2] "`Weight Of Cup + Sample`"
[3] "`Weight Of Cup + Sample After Drying`"
[4] "`Weight Of Sample Before Drying`"
[5] "`Weight Of Sample After Drying`"
[6] "`Moisture Content (%)`"
```

```
ordvars <- ColSeeker(namepattern='Weight.+Sample', exclude = 'After')
ordvars$names
```

```
[1] "Weight Of Cup + Sample"         "Weight Of Sample Before Drying"
```

```
factvars <- ColSeeker(namepattern='region',casesensitive = FALSE)
factvars$bticked
```

```
[1] "`Region`"    "`Subregion`"
```


Modellierung

Beschreibende Statistik

- *mean() / sd() / meansd()*
- *median() / quantile() / median_quart()*
- *table() / prop.table() / cat_desc_stats()*

Tests

- *t.test() / lm()+[Aa]nova() / compare2numvars()*
- *wilcox.test()*
- *fisher.test() / glm(family=binomial)*

Modellierung: Deskriptiv

Stichprobengröße n: pro Variable, wenn fehlende Werte auftreten

---

Mittelwert: zentrale Tendenz, erwarteter *typischer* Wert

$$\frac{\sum x}{n}$$

Varianz: Kennwert für Variabiliät/Heterogenität der Daten

$$\frac{\sum (x - mean)^2}{n - 1}$$

Standardabweichung SD: *typische* gewichtete Abweichung vom Mittelwert

$$\sqrt{Var}$$

Standardfehler des Mittelwerts: wie zuverlässig ist die Mittelwerts*schätzung*, was wäre die zu erwartende SD der Mittelwerte aus wiederholten Experimenten?

$$\frac{SD}{\sqrt{n}}$$

Median: Trennung der unteren/oberen 50% der Daten

Quartile: Trennung bei 25%/50%/75% der Daten (allgemein: **Quantile**, z.B.**Perzentile**), Grundlage des Boxplot

verschiedene Berechnungsmethoden

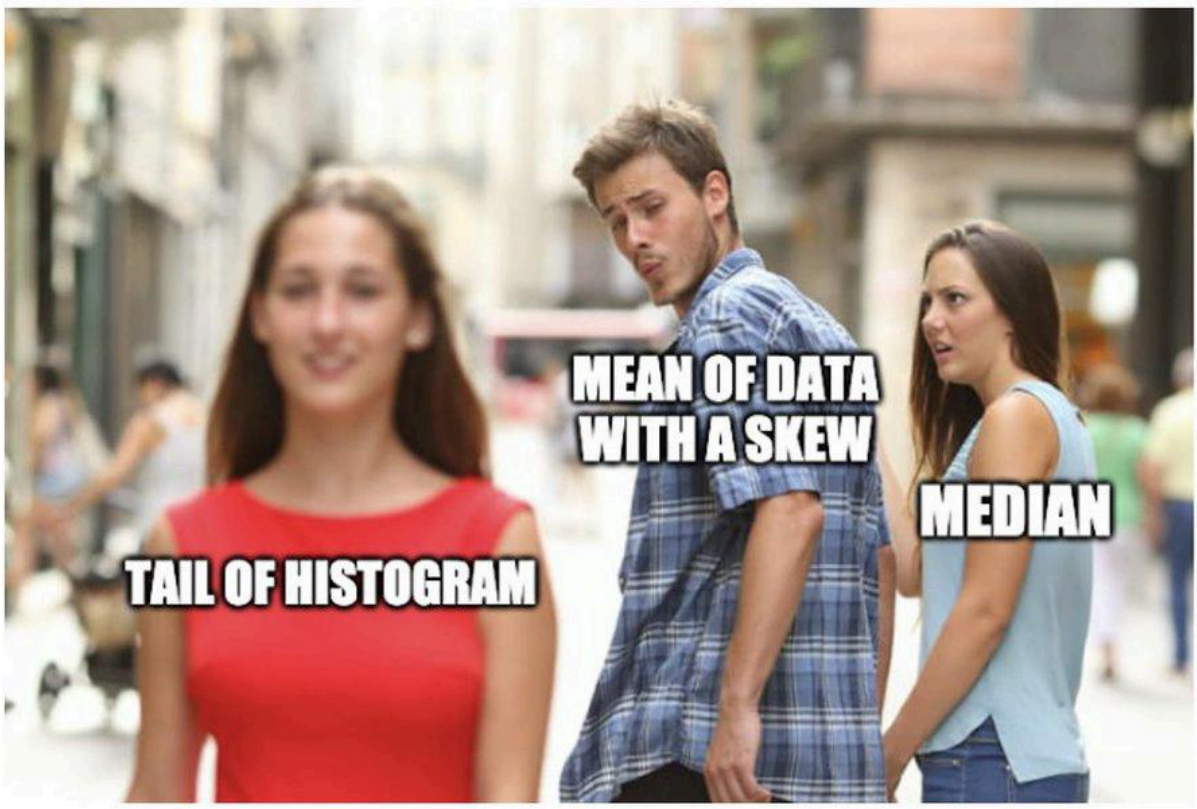
```r
desc_gauss <- rawdata |>
  summarize(across(.cols = gaussvars$names,
                   .fns = meansd))
desc_gauss
```

```
# A tibble: 1 × 7
  `Weight Of Empty Cup` `Weight Of Cup + Sample` Weight Of Cup + Sample After …¹
  <chr>                 <chr>                    <chr>
1 4.1 ± 0.1             9.2 ± 0.2                4.6 ± 0.1
# ℹ abbreviated name: ¹`Weight Of Cup + Sample After Drying`
# ℹ 4 more variables: `Weight Of Sample Before Drying` <chr>,
#   `Weight Of Sample After Drying` <chr>, `Moisture Content (%)` <chr>,
#   `Dry Content (%)` <chr>
```

```r
desc_ord <- rawdata |>
  summarize(across(ordvars$names,.fns=~median_quart(.x,roundDig = 3))) |>
  pivot_longer(everything(),
               names_to = 'Measure', values_to = 'Median[1Q/3Q]')
desc_ord
```

```
# A tibble: 2 × 2
  Measure                       `Median[1Q/3Q]`
  <chr>                         <chr>
1 Weight Of Cup + Sample        9.25 (9.22/9.29)
2 Weight Of Sample Before Drying 5.14 (5.09/5.18)
```


Deskriptive Statistik sollte zu Verteilung und Daten passen



Modellierung: Tests

Tests benötigen Hypothesen

Nullhypothese ?

- Arbeitshypothese: Üblicherweise ein erwarteter Effekt!
  z.B. Behandlung senkt den Blutdruck stärker als ein Placebo, transgene Tiere werden adipös, Bioreaktor A ist effizienter als B, Konzentration einer Substanz ist korreliert mit der Reaktionsgeschwindigkeit …
- Nullhypothese: Dies wird getestet!
  Kein Unterschied / Zusammenhang, Blutdruck unter Therapie = BD unter Placebo

4 Möglichkeiten:

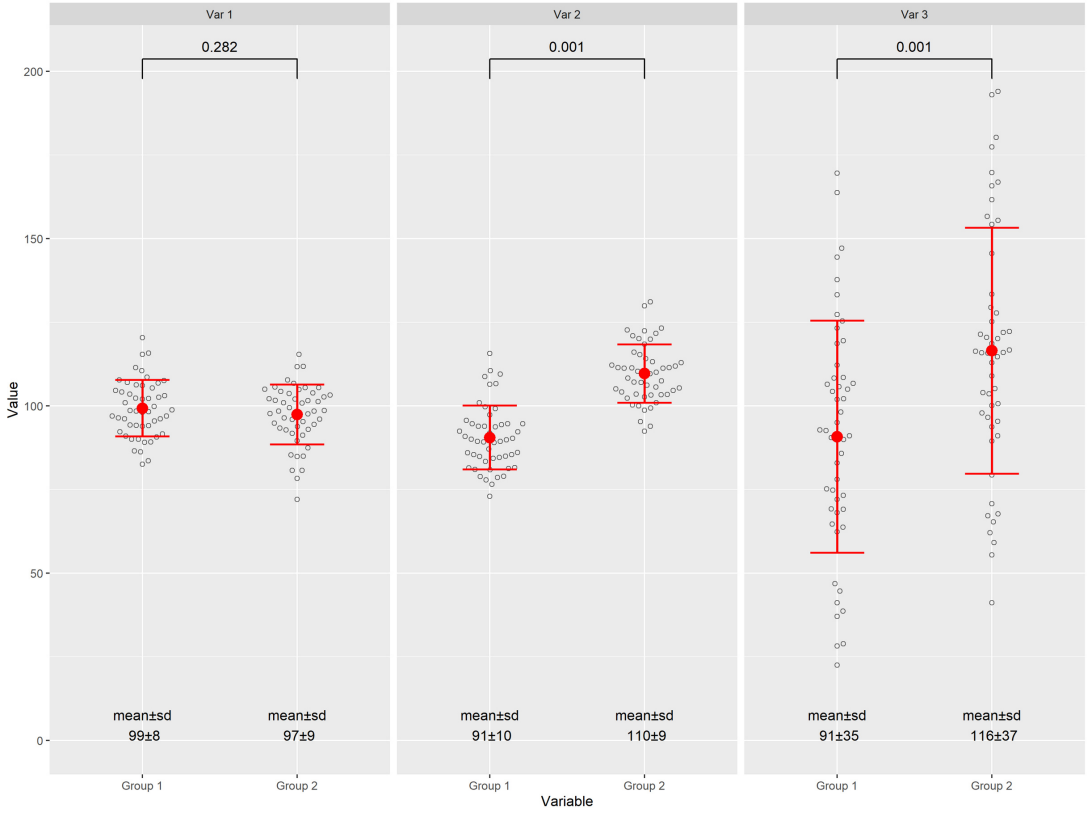
- Nullhypothese korrekt, Test falsch positiv (Fall A): alpha-Fehler
- Nullhypothese korrekt, Test korrekt negativ (Fall B)
- Nullhypothese falsch, Test falsch negativ (Fall C): beta-Fehler
- Nullhypothese falsch, Test korrekt positiv (Fall D)

Signifikanz: NICHT Wahrscheinlichkeit von Fall A, sondern Wahrscheinlichkeit der Daten/beobachteten Effekte unter Annahme der NULLhypothese, berechnet aus den Daten, üblicherweise<0.05

Power: Wahrscheinlichkeit von Fall D, falls Nullhyothese falsch ist; *geschätzt* aus Annahmen zu Effektstärke, Variabilitäten und Fallzahl, *Berechnung* würde Wissen um wahre Unterschiede voraussetzen, üblicherweise = 0.80; daraus leitet sich beta-Fehler-Wahrscheinlichkeit von 0.20 ab!

Testfunktionen

t-Test / Wilcoxon-Test (aka Mann-Whitney U-test)



t-Test

- Voraussetzung: Kontinuierliche Daten mit Normalverteilung
- 1 or 2 (unabhängige or abhängige) Stichproben mit/ohne gleiche Varianzen
- wie groß ist der Mittelwertsunterschied relativ zur Unsicherheit der Mittelwerte?
  $t = (mean_1 - mean_2)/SEM$
- t folgt einer t-Verteilung, das erlaubt die Schätzung der Wahrscheinlichkeit von t unter der NULLhypothese

Wilcoxon-test

- nichtparametrisch, keine Vereilungsannahme
- basiert auf rang-transformierten Daten
- unempfindlich gegen Extremwerte

Test Beispiele: *einzelne Variablen*

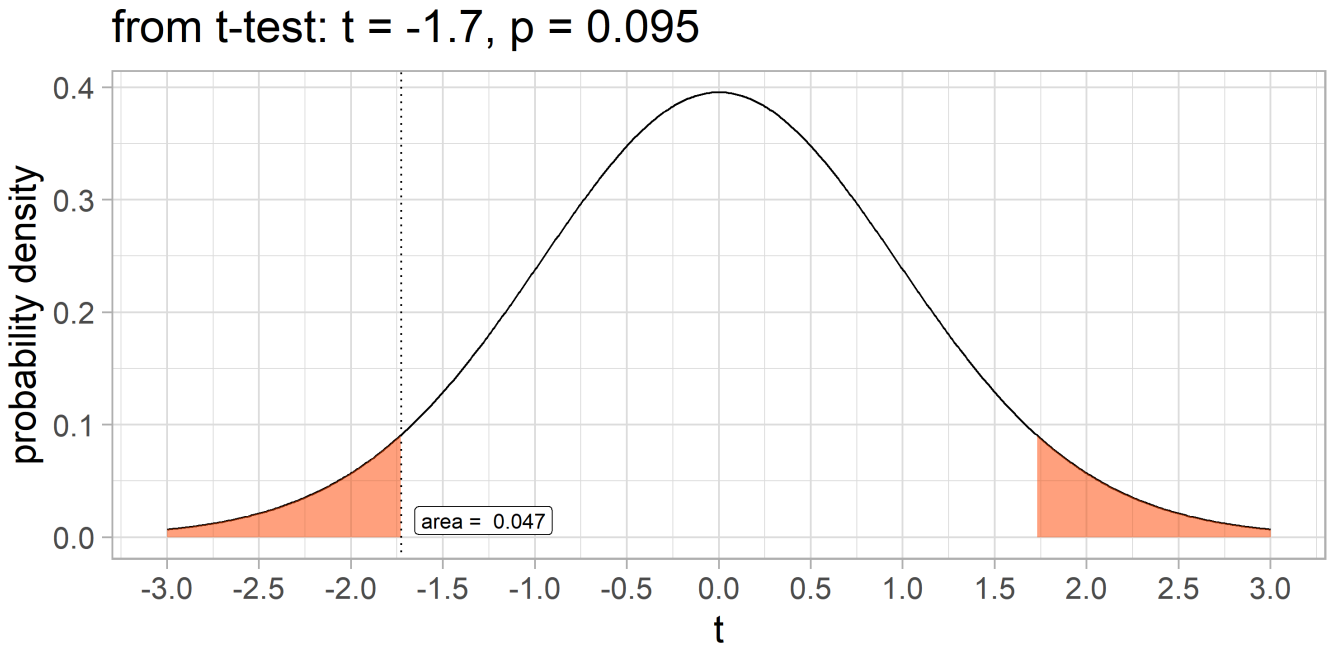
```r
#t-Test with test for equal variances
t_out <- t.test(formula=`Moisture Content (%)`~`Region`, data=rawdata,
                var.equal=var.test(
                  formula=`Moisture Content (%)`~`Region`,
                  data=rawdata)$p.value>.05)
t_out
```

```
    Welch Two Sample t-test

data:  Moisture Content (%) by Region
t = -1.7274, df = 29.239, p-value = 0.09465
alternative hypothesis: true difference in means between group A and group D is not equal
to 0
95 percent confidence interval:
 -2.9624835  0.2490486
sample estimates:
mean in group A mean in group D
       90.31199        91.66870
```

```r
#Wilcoxon-Test
wilcox.test(`Moisture Content (%)`~`Region`,
            data = rawdata)
```

```
    Wilcoxon rank sum exact test

data:  Moisture Content (%) by Region
W = 107, p-value = 0.7547
alternative hypothesis: true location shift is not equal to 0
```

Von t zu p



Modellierung: Test 2 / mehrere Variablen

```
test_gauss <- compare2numvars(data = rawdata,
                              dep_vars = gaussvars$names,
                              indep_var = 'Region',
                              gaussian = TRUE,
                              round_p = 5)
test_gauss |> flextable()|>
  theme_zebra(even_body = 'aquamarine',odd_body = 'antiquewhite')
```

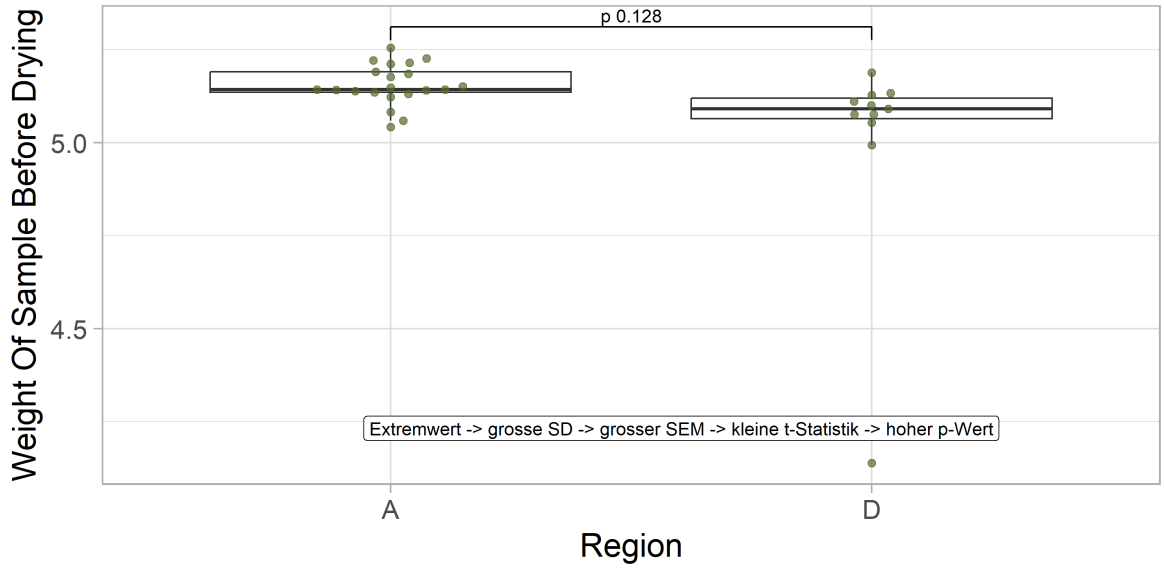
Variable	desc_all	Region A	Region D	p
Weight Of Empty Cup	4.1 ± 0.1	4.1 ± 0.0	4.2 ± 0.0	0.00001
Weight Of Cup + Sample	9.2 ± 0.2	9.2 ± 0.0	9.2 ± 0.3	0.74227
Weight Of Cup + Sample After Drying	4.6 ± 0.1	4.6 ± 0.2	4.6 ± 0.1	0.42213
Weight Of Sample Before Drying	5.1 ± 0.2	5.2 ± 0.1	5.0 ± 0.3	0.12830
Weight Of Sample After Drying	0.47 ± 0.14	0.50 ± 0.16	0.42 ± 0.07	0.04937
Moisture Content (%)	91 ± 3	90 ± 3	92 ± 1	0.09465
Dry Content (%)	9.2 ± 2.7	9.7 ± 3.1	8.3 ± 1.3	0.09465

```r
test_ord <- compare2numvars(data = rawdata,
                            dep_vars = ordvars$names,
                            indep_var = 'Region',
                            gaussian = FALSE,round_desc = 3)
test_ord |> flextable() |>
  theme_zebra(even_body = 'aquamarine',odd_body = 'antiquewhite')
```

Variable	desc_all	Region A	Region D	p
Weight Of Cup + Sample	9.25 (9.22/9.29)	9.24 (9.22/9.25)	9.29 (9.27/9.31)	0.003
Weight Of Sample Before Drying	5.14 (5.09/5.18)	5.14 (5.13/5.20)	5.09 (5.06/5.13)	0.001

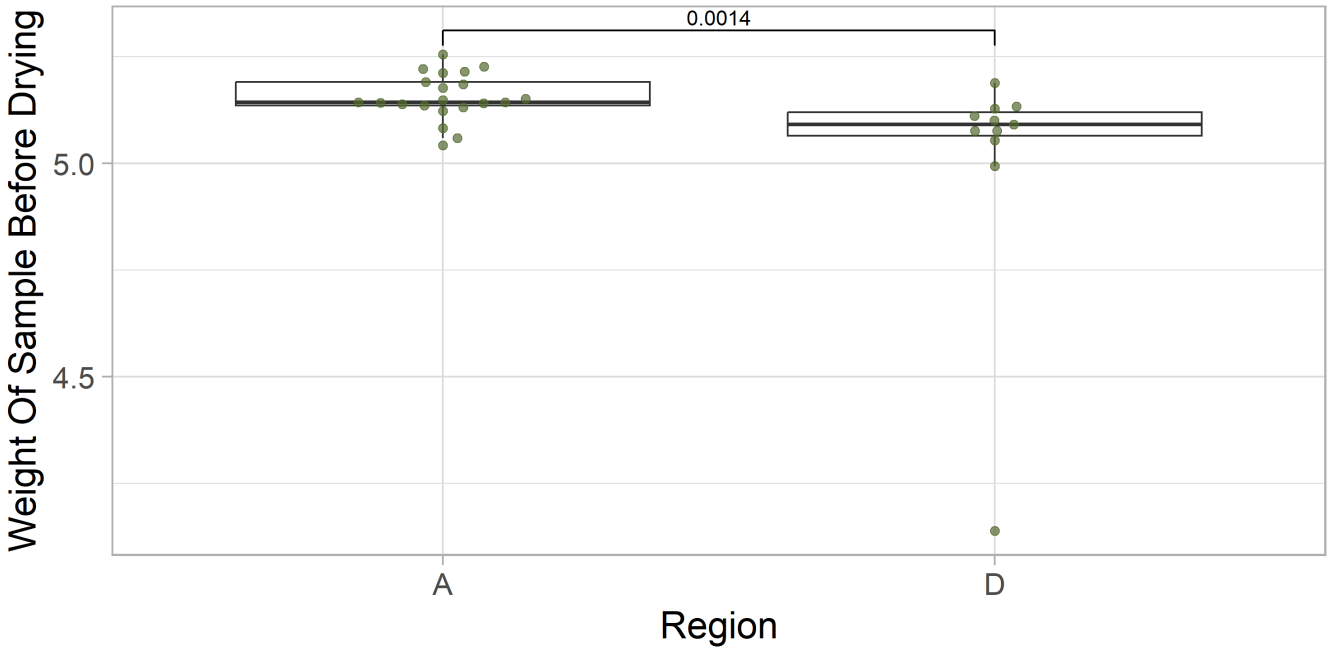
Ergebnisdarstellung

```r
ggplot(rawdata, aes(x = `Region`,y = `Weight Of Sample Before Drying`))+
  geom_boxplot(outlier.alpha = 0)+
  geom_beeswarm(alpha=.7, size=2,cex = 2, color="darkolivegreen")+
  annotate(geom = 'label',x=2,y=4.2,
           label='Extremwert -> grosse SD -> grosser SEM -> kleine t-Statistik
           hjust=0.8,vjust=0)+
  geom_signif(comparisons = list(c(1,2)),
              annotations = paste('p',formatP(t_out$p.value)))
```



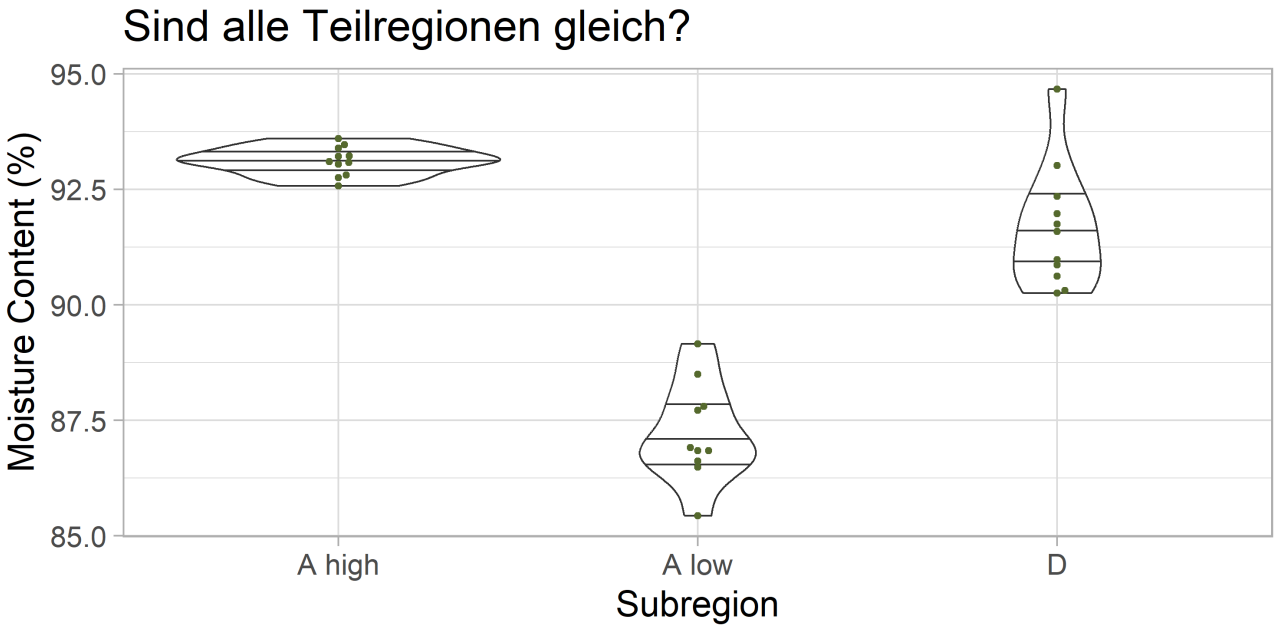
Umentscheidung bei Testauswahl?

```
ggplot(rawdata, aes(x = `Region`,y = `Weight Of Sample Before Drying`))+
  geom_boxplot(outlier.alpha = 0)+
  geom_beeswarm(alpha=.7, size=2,cex = 2, color="darkolivegreen")+
  geom_signif(comparisons = list(c(1,2)),test = wilcox.test)
```



Modellierung: lineare Modelle 1 / univariable

```
plottmp <- ggplot(rawdata,aes(Subregion,`Moisture Content (%)`))+
  geom_violin(draw_quantiles = c(.25,.5,.75))+
  geom_beeswarm(color="darkolivegreen")+

  ggtitle('Sind alle Teilregionen gleich?')
print(plottmp)
```



ANOVA: Modellbildung

```
rawdata |> group_by(Subregion) |>
  summarize(MeanMoisture=mean(`Moisture Content (%)`) |> roundR(4)) |>
  pivot_wider(names_from = Subregion,values_from = MeanMoisture) |>
  rename_with(~paste('Mean moisture %\n',.x)) |> flextable()|>
  theme_zebra(even_body = 'aquamarine',odd_body = 'antiquewhite')
```

Mean moisture % A high	Mean moisture % A low	Mean moisture % D
93.11	87.23	91.67

```
lm1<- lm(`Moisture Content (%)`~Subregion, data=rawdata)
lm1
```

```
Call:
lm(formula = `Moisture Content (%)` ~ Subregion, data = rawdata)

Coefficients:
   (Intercept)  SubregionA low       SubregionD
        93.112          -5.879           -1.443
```


ANOVA: p-Werte

```
anova(lm1) |> broom::tidy() |> flextable()|>
  theme_zebra(even_body = 'aquamarine',odd_body = 'antiquewhite')
```

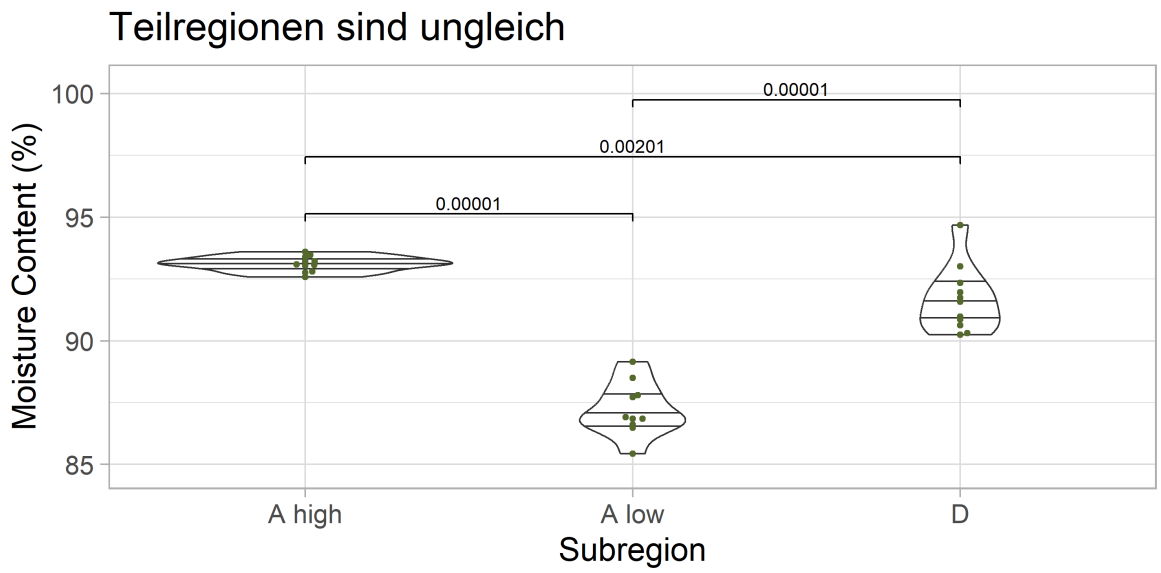
term	df	sumsq	meansq	statistic	p.value
Subregion	2	194.34239	97.1711969	97.76477	0.0000000000001292127
Residuals	29	28.82393	0.9939285		

```
#post-hoc
(posthoc_out <- pairwise.t.test(x = rawdata$`Moisture Content (%)`,
                                g = rawdata$Subregion,
                                p.adjust.method = 'fdr')$p.value |>
    formatP(ndigits = 5))
```

```
       A high    A low
A low "0.00001" "    NA"
D     "0.00201" "0.00001"
```

Visualisierung ANOVA

```
ggplot(rawdata,aes(Subregion,`Moisture Content (%)`))+
  geom_violin(draw_quantiles = c(.25,.5,.75))+
  geom_beeswarm(color="darkolivegreen")+
  geom_signif(comparisons = list(c(1,2),c(1,3),c(2,3)),
              annotations = c(posthoc_out[,1], posthoc_out[2,2]),
              step_increase = .25)+
  scale_y_continuous(expand = expansion(mult = .1))+
  ggtitle('Teilregionen sind ungleich')
```



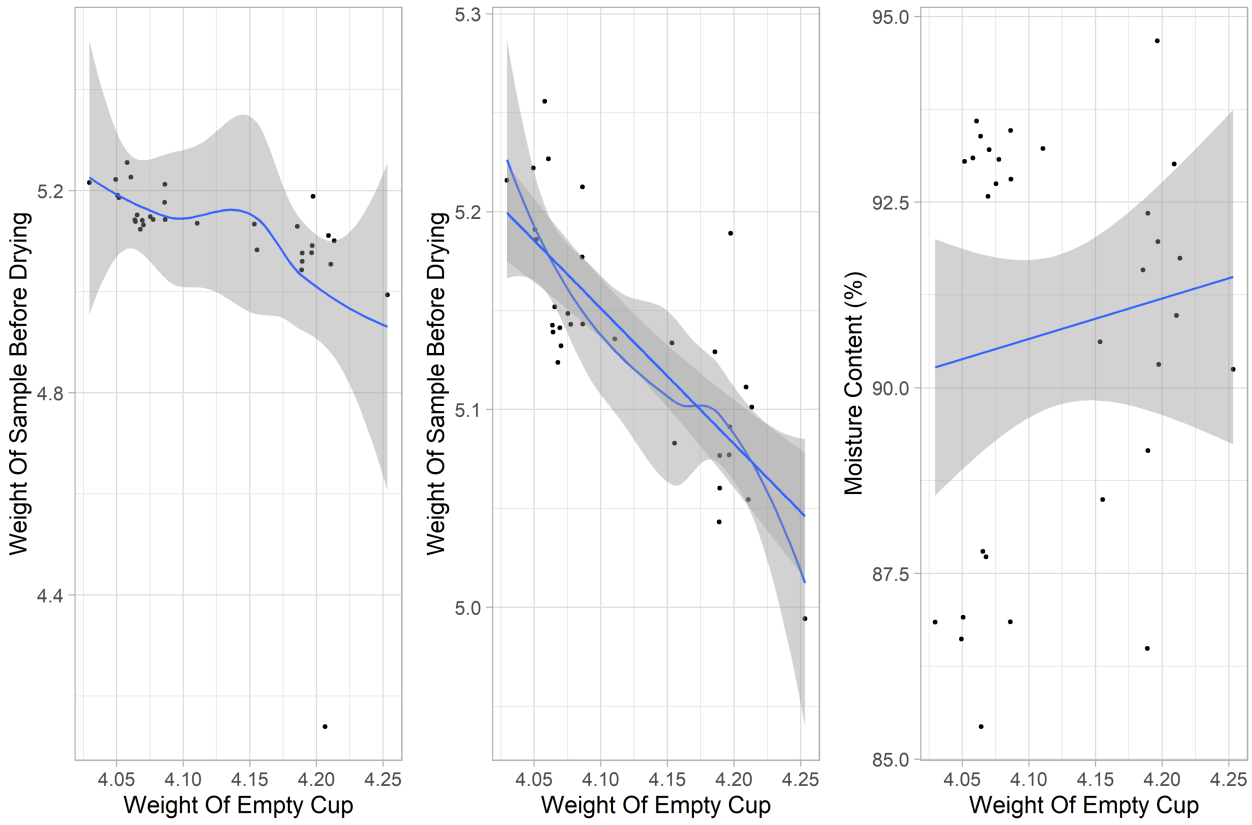
Analyse von mehr als 1 Zielgröße

```
test_out <- compare_n_numvars(.data=rawdata,
                              dep_vars=gaussvars$names,
                              indep_var='Subregion',
                              gaussian=TRUE)
test_out$results |>
  select(Variable,contains("fn"),multivar_p) |>
  rename_with(~str_remove(.x," fn")) |>
  flextable()|>
  theme_zebra(even_body = 'aquamarine',odd_body = 'antiquewhite') |>
  add_footer_lines(
    values='b bedeutet Unterschied zu Gruppe 2, c Unterschied zu Gruppe 3')
```

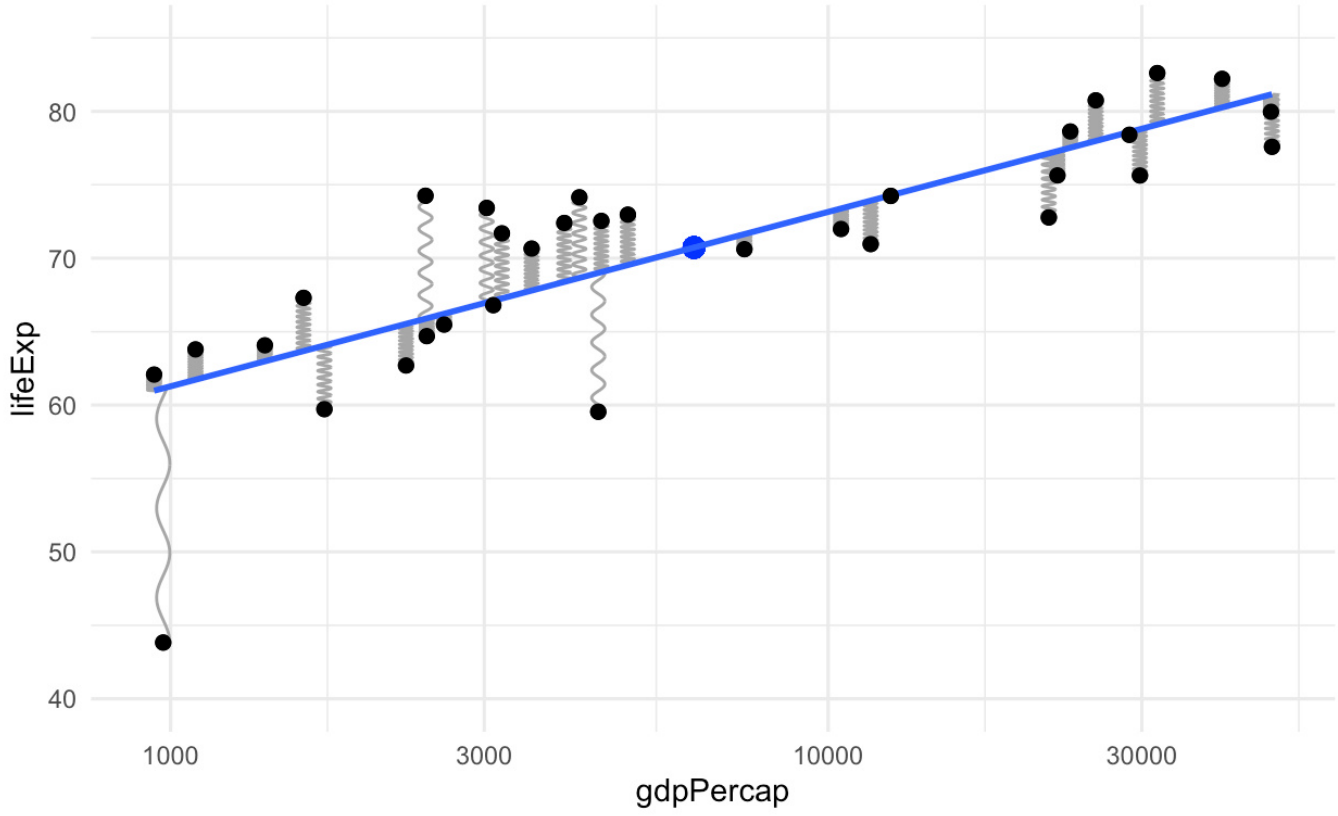
Variable	Subregion A high	Subregion A low	Subregion D	multivar_p
Weight Of Empty Cup	4.1 ± 0.0 c	4.1 ± 0.1 c	4.2 ± 0.0	0.001
Weight Of Cup + Sample	9.2 ± 0.0	9.2 ± 0.0	9.2 ± 0.3	0.894
Weight Of Cup + Sample After Drying	4.4 ± 0.0 bc	4.8 ± 0.1 c	4.6 ± 0.1	0.001
Weight Of Sample Before Drying	5.2 ± 0.0	5.1 ± 0.1	5.0 ± 0.3	0.095
Weight Of Sample After Drying	0.36 ± 0.02 bc	0.66 ± 0.06 c	0.42 ± 0.07	0.001
Moisture Content (%)	93 ± 0 bc	87 ± 1 c	92 ± 1	0.001
Dry Content (%)	7 ± 0 bc	13 ± 1 c	8 ± 1	0.001

b bedeutet Unterschied zu Gruppe 2, c Unterschied zu Gruppe 3

Regression: Scatterplot



Regression: Grundlegender *Mechanismus*



Regression: Statistik

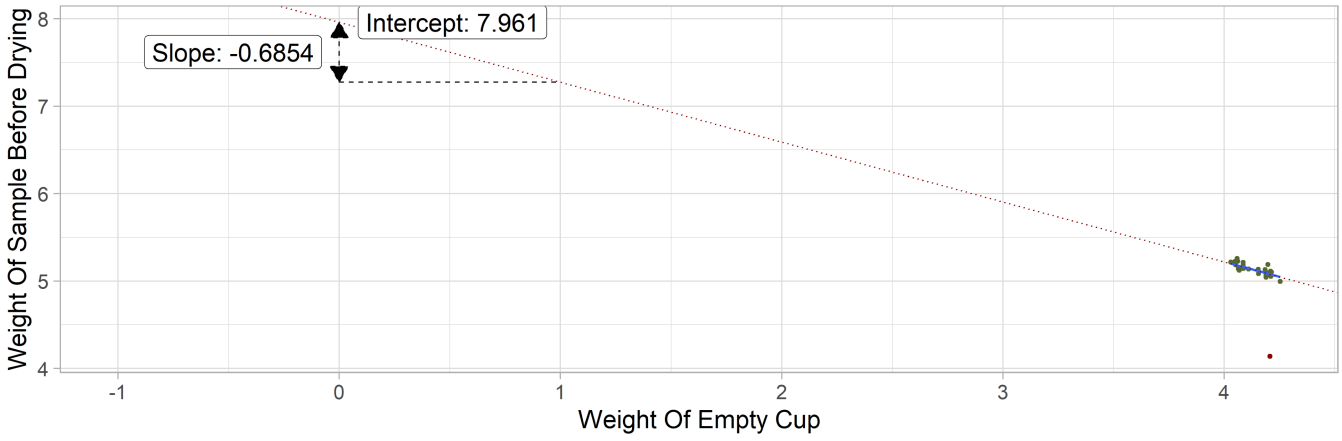
```
lm_out0 <- lm(`Weight Of Sample Before Drying` ~ `Weight Of Empty Cup`,
              data=rawdata)
lm_out0
```

```
Call:
lm(formula = `Weight Of Sample Before Drying` ~ `Weight Of Empty Cup`,
    data = rawdata)

Coefficients:
          (Intercept)  `Weight Of Empty Cup`
               10.169                 -1.228
```

```
# filtering outlier
lm_out <- lm(`Weight Of Sample Before Drying` ~ `Weight Of Empty Cup`,
              data=rawdata |> filter(`Weight Of Sample Before Drying`>4.5))
lm_out
```

```
Call:
lm(formula = `Weight Of Sample Before Drying` ~ `Weight Of Empty Cup`,
    data = filter(rawdata, `Weight Of Sample Before Drying` >
        4.5))

Coefficients:
          (Intercept)  `Weight Of Empty Cup`
               7.9612                 -0.6854
```


Regression: Visualisierung



Regression: Signifikanz

```
anova(lm_out) |> broom::tidy()
```

```
# A tibble: 2 × 6
  term                   df  sumsq  meansq statistic      p.value
  <chr>                <int>  <dbl>   <dbl>     <dbl>        <dbl>
1 `Weight Of Empty Cup`     1 0.0638 0.0638       42.1  0.000000425
2 Residuals                29 0.0440 0.00152      NA   NA
```

```
model_parameters(lm_out)
```

```
Parameter           | Coefficient |   SE |         95% CI | t(29) |      p
-------------------------------------------------------------------------
(Intercept)         |        7.96 | 0.44 | [ 7.07,  8.85] | 18.28 | < .001
Weight Of Empty Cup |       -0.69 | 0.11 | [-0.90, -0.47] | -6.49 | < .001
```


Berichtserstellung

- *RMarkdown und quarto sind mächtige Werkzeuge für Berichte und Präsentationen*
- Export von Abbildungen: ggsave() / png() / pdf()
- Export von Tabellen: write_xlsx()
- *Paket flextable bietet viele Möglichkeiten zur Tabellenformatierung*

Flextable Beispiel

```
test_ord |> select(-desc_all) |> rename_with(~str_remove(.,'Code Of ')) |>
  flextable() |>
  theme_zebra(even_body = 'aquamarine',odd_body = 'antiquewhite')|>
  italic(~p<=0.05,j = 1) |> bg(~p<=0.05,j = 4,bg = 'yellow') |>
  set_caption('Treatment effects, measures following a normal distribution') |
  add_footer_lines('Significance level is set at 0.05') |>
  fontsize(size = 12,part = 'footer')
```

Variable	Region A	Region D	p
Weight Of Cup + Sample	9.24 (9.22/9.25)	9.29 (9.27/9.31)	0.003
Weight Of Sample Before Drying	5.14 (5.13/5.20)	5.09 (5.06/5.13)	0.001

Significance level is set at 0.05

Nützliche Werkzeuge

- Auswahl Spalten / Zeilen: select() / pull() / filter() / slice()
- Umformatierung von Tabellen breit <–> lang (z.B. für wiederholte Messungen):
  pivot_longer()/pivot_wider()
- Reguläre Ausdrücke: str_replace() / str_detect() / str_…
- Zusammenfügen von Textelementen: paste() / str_glue()
- Anwendung von Funktionen: purrr::map_xxx