

**IntroSlides with focus on  
statistics**

# Basic steps in workflow

1. *Define environment*
2. *Import*
3. *Transform*
4. *Explore (general/outlier/distribution) (go back to 3?)*
5. *Classify scale level / distribution (based on 3/4)*
6. *Describe*
7. *Test / Model (may include step 6)*
8. *Report*

# Define environment

- Activate packages to use: *library()* / *pacman::p\_load()*
- *ggplot theme*: *theme\_set()* / *theme\_update()*
- *flextable settings*: *set\_flextable\_defaults()*
- *knitr::opts\_chunk\$set()*

```
1 if(!require(pacman)){install.packages("pacman")}
2 pacman::p_load(conflicted,tidyverse,wrappedtools, readxl,car, flextable,
3                 ggbeeswarm, ggsignif, ggridges, patchwork, easystats)
4
5 conflicts_prefer(dplyr::filter,dplyr::select)
6 theme_set(theme_light(base_size = 20))
7 gdtools::register_gfont('Roboto') # Mono'
```

```
[1] TRUE
```

```
1 set_flextable_defaults(
2   theme_fun = theme_zebra, font.size = 18, font.family = 'Roboto',
3   table.layout = 'autofit',
4   padding.bottom = .2, padding.top = .2, padding.left = 2, padding.right = 2)
5
6 knitr::opts_chunk$set(message = FALSE, warning = FALSE, comment = NA, echo = TRUE)
```

# Import

- *read\_xlsx() / read\_csv() / read\_csv2()*
- options related to separators, number formats, ranges etc.
- *rename() / rename\_with()*

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

# Glimpse at data: Find the problems?

```
1 head(rawdata, n = 15) |> flextable() |>
2 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

# Rename

```
1 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 (%)"
```

```
1 rawdata <- rawdata |>
2   rename(Region= `CODE OF SAMPLE` ) |>
3   rename_with(.fn = ~str_replace_all(.,
4                               c('AL.+UM'= 'Cup', 'C[UA]P' = 'Cup', '\\\\(\\\\w+.*\\\\)
5                               'Cup Cup'='Cup', '    =' )) |>
6   str_to_title() |> str_trim()
7 cn()
```

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

# Transform

- Change or create columns with `mutate()` / `mutate(across())`
- e.g. for log-transformation, creation of factors, text recoding

```
1 rawdata <- rawdata |>
2   mutate(
3     `Weight Of Sample After Drying` = `Weigth Of Cup + Sample After Drying` -
4       `Weight Of Empty Cup`,
5     `Dry Content (%)` = `Weight Of Sample After Drying` * 100 /
6       `Weight Of Sample Before Drying`,
7     `Moisture Content (%)` = 100 - `Dry Content (%)`)
```

Code Of Cup	Region	Weight Of Empty Cup	Weigth Of Cup + Sample	Weighth 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
	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

# Explore / group variables

## *Explore (general/outlier/distribution)*

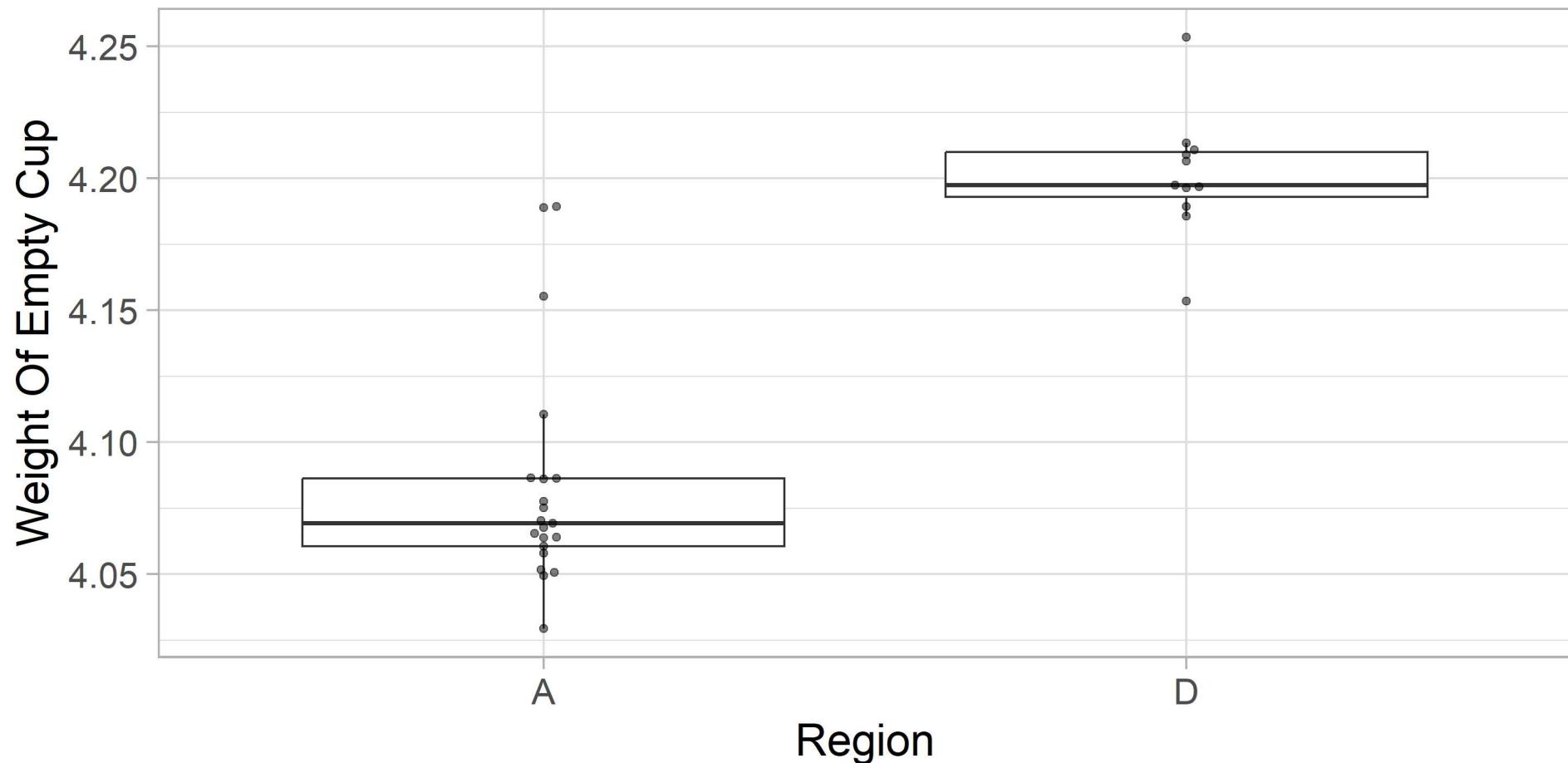
- `ggplot() + geom_boxplot() / geom_beeswarm() / geom_density()`
- `ks.test() / ksnormal() / shapiro.test()`

## *Classify scale level / distribution*

- `gaussvars / ordvars / factvars`, possibly more...
- *Store variables accordingly, e.g. FindVars()*

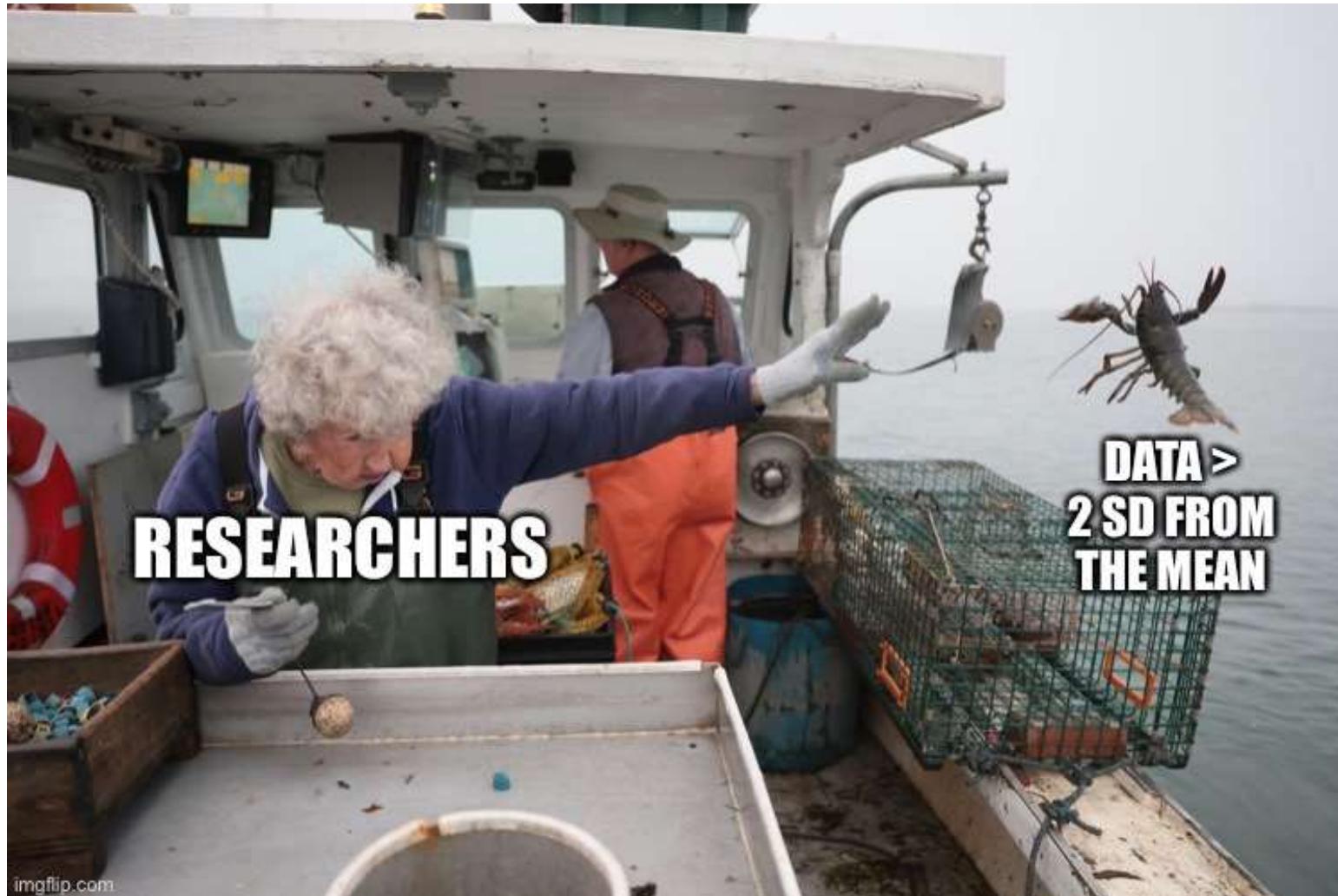
# Explore: Outlier

```
1 ggplot(data = rawdata,
2         aes(x = `Region`,
3               y = `Weight Of Empty Cup`))+  
4   geom_boxplot(outlier.alpha = 0) + #hide outliers, beeswarm will plot them  
5   geom_beeswarm(alpha=.5)
```



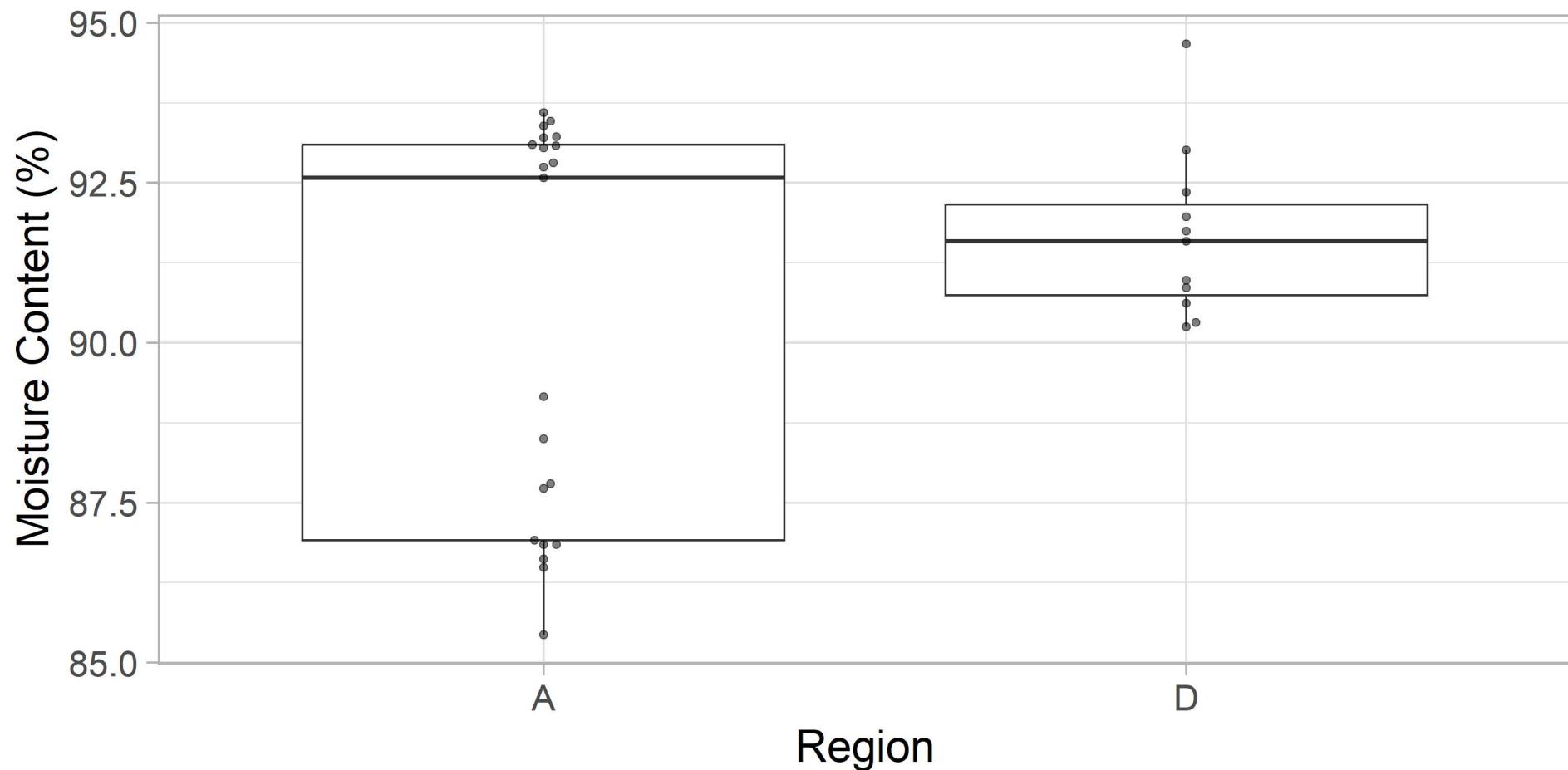
# Handle outliers?

Removal is the worst strategy possible, correct errors, think about distributions, winsorize...



# Explore: Unexpecteds

```
1 ggplot(data = rawdata,
2         aes(x = `Region`,
3               y = `Moisture Content (%)`))+  
4   geom_boxplot(outlier.alpha = 0) + #hide outliers, beeswarm will plot them  
5   geom_beeswarm(alpha=.5)
```



# *Explore: Normal distribution 1*

- Gaussian Normal distribution is required for many statistical procedures
- Common tests are graphical exploration, Shapiro-Wilk-test and Kolmogorov-Smirnov-test

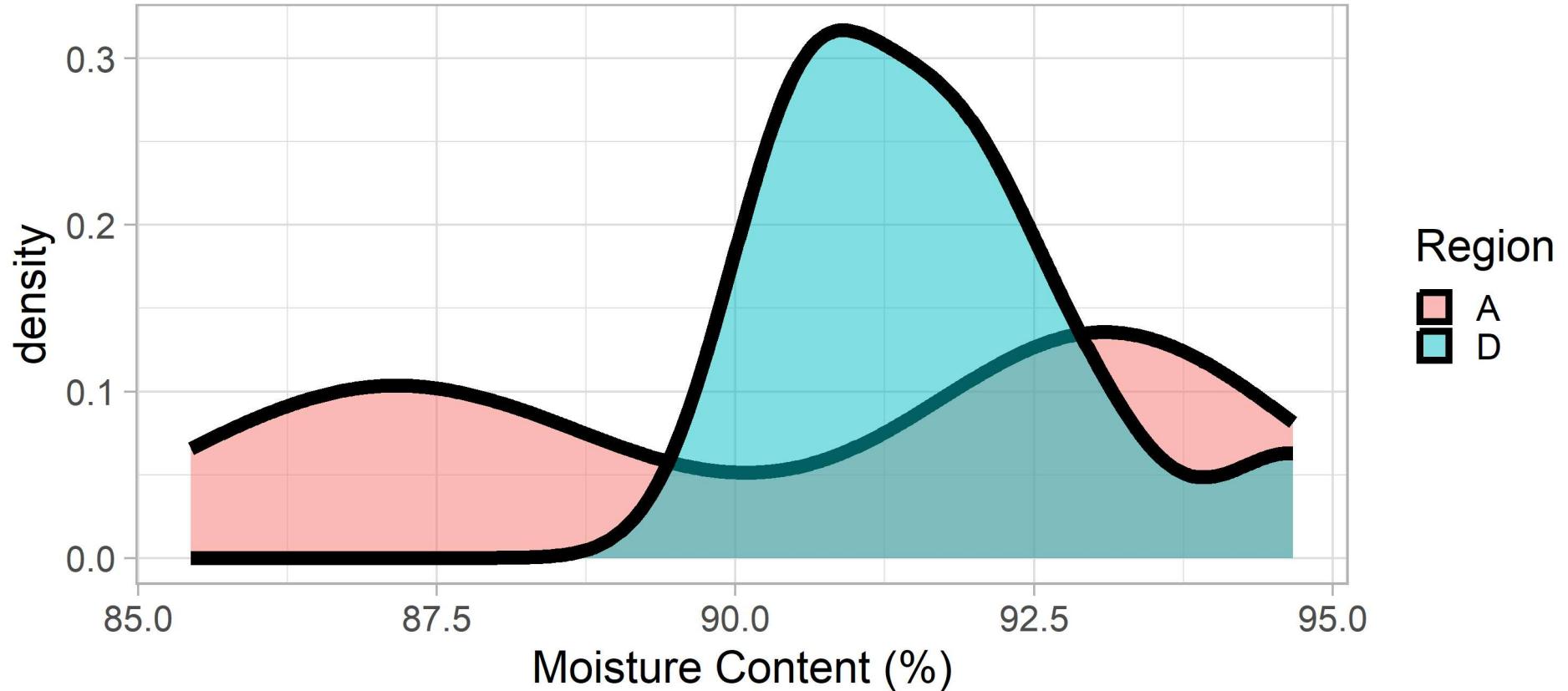
```
1 p_normal <-
2   shapiro.test(x = rawdata$`Moisture Content (%)`)
3 p_normal
```

Shapiro-Wilk normality test

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

```
1 ggplot(data = rawdata,aes(x = `Moisture Content (%)`,fill=`Region`))+  
2   geom_density(linewidth=3, alpha=.5)+  
3   ggtitle(paste('p (Shapiro)',  
4               formatP(pIn = p_normal$p.value, pretext = T)))
```

$$p (\text{Shapiro}) = 0.004$$



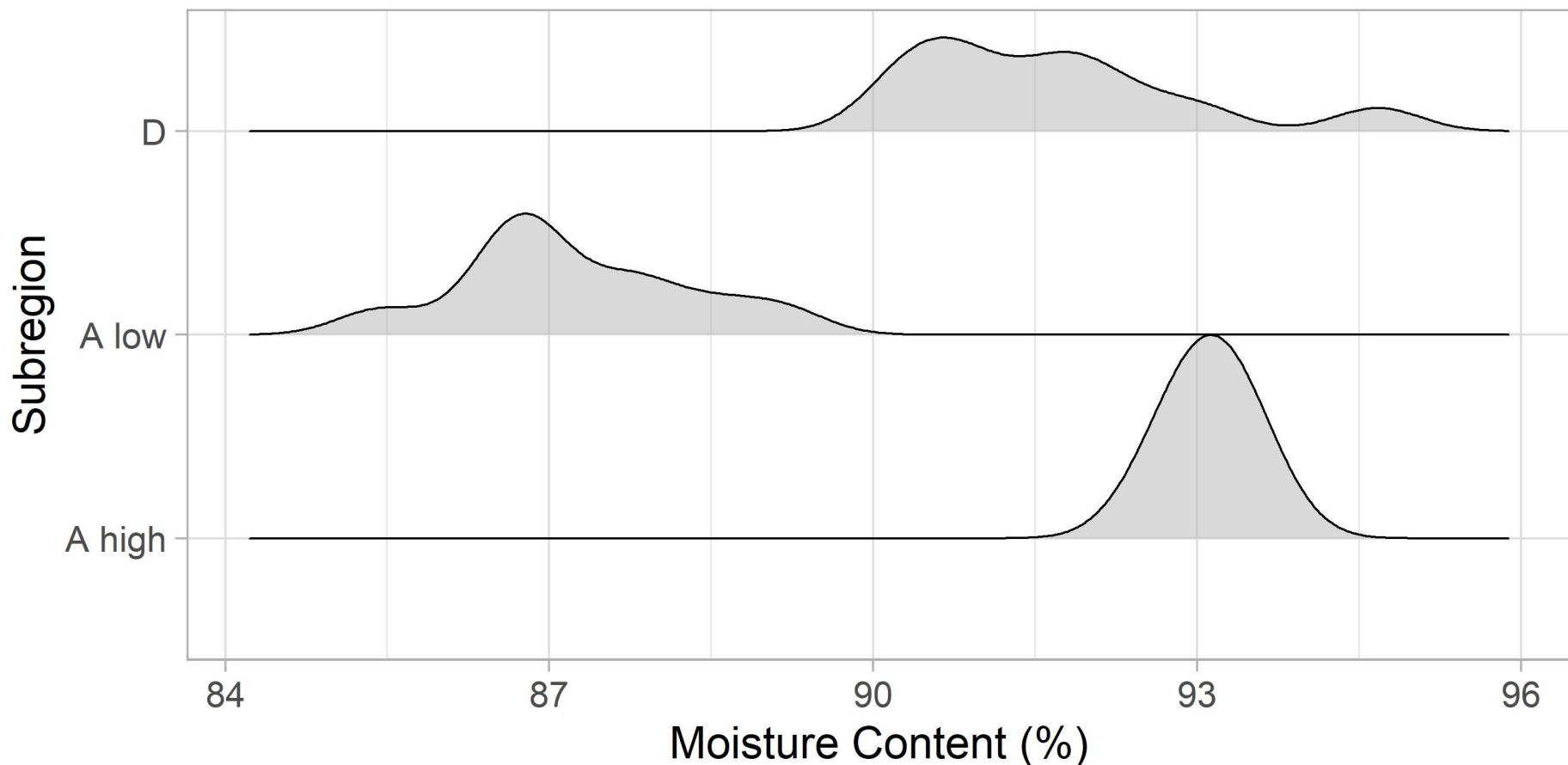
# *Explore: Normal distribution 2*

```
1 rawdata |>
2   group_by(`Region`) |>
3   summarize(across(.cols = where(is.numeric), .fns = ksnormal)) |>
4   pivot_longer(cols = -1, names_to = 'Variable', values_to = 'pKS') |>
5   pivot_wider(names_from = `Region`, values_from = pKS)

# A tibble: 7 × 3
  Variable           A       D
  <chr>            <dbl>  <dbl>
1 Weight Of Empty Cup 0.0680 0.697
2 Weigth Of Cup + Sample 0.791  0.0187
3 Weigth Of Cup + Sample After Drying 0.0556 0.969
4 Weight Of Sample Before Drying    0.742  0.0716
5 Weight Of Sample After Drying     0.0535 1.00 
6 Moisture Content (%)        0.0570 0.954
7 Dry Content (%)          0.0570 0.954
```

# Transform Subregions?

```
1 rawdata <- mutate(rawdata, Subregion = case_when(  
2   `Region`=='A' & `Moisture Content (%)` > 90 ~ 'A high',  
3   `Region`=='A' & `Moisture Content (%)` <= 90 ~ 'A low',  
4   `Region`=='D' ~ 'D') |> factor())  
5 ggplot(data = rawdata,aes(x = `Moisture Content (%)`,y=Subregion))+  
6   geom_density_ridges(alpha=.5, scale=1)
```



# *Explore: Group variables by type/distribution*

Scale level determines what statistics are appropriate

Typical scale levels are

- nominal/categorical/factorial/qualitative: just different groups (species, eye color, genotype, treatment)
- ordered categories: few groups with inherent order (quality bad<medium<good, pain between 0 and 10)
- ordinal measures: many different values, natural order, no distribution assumption (satisfaction on a scale from 0 to 100)
- measures following a Normal distribution
- possibly measures from other known distributions (beta, log-normal, poisson...), often treated as ordinal

# *Make type decision obvious/reproducible*

```
1 gaussvars <- FindVars(varnames = 'Weight Of Sample',
2                               allnames = cn(rawdata))
3 gaussvars

$index
[1] 6 7

$names
[1] "Weight Of Sample Before Drying" "Weight Of Sample After Drying"

$bticked
[1] ``Weight Of Sample Before Drying`` ``Weight Of Sample After Drying``

$count
[1] 2

1 ordvars <- FindVars(c('Cup', 'Content'), exclude = 'Code')
2 ordvars$names

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

1 factvars <- FindVars('Region', casesensitive = FALSE)
2 factvars$bticked

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

# Model

## Describe

- *mean()* / *sd()* / *meansd()*
- *median()* / *quantile()* / *median\_quart()*
- *table()* / *prop.table()* / *cat\_desc\_stats()*

## Test

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

# *Model: Describe*

**Sample size n:** per variable, if there are NAs

---

**Mean:** central tendency, the expected *typical* value

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

---

**Variance:** measure for variability/heterogeneity of data

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

---

**Standard deviation SD:** the *typical* weighted deviation from the mean

$$\sqrt{Var}$$

**Standard error of the mean SEM:** how reliable is the mean estimate, what would be the expected SD of means from repeated experiments?

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

---

**Median:** Split between lower/upper 50% of data

---

**Quartiles:** Split at 25%/50%/75% of data (more

various

general: **Quantiles**, e.g.**Percentiles**), used in boxplot

computational  
approaches

```

1 desc_gauss <- rawdata |>
2   summarize(across(.cols = gaussvars$names,
3                     .fns = meansd))
4 desc_gauss

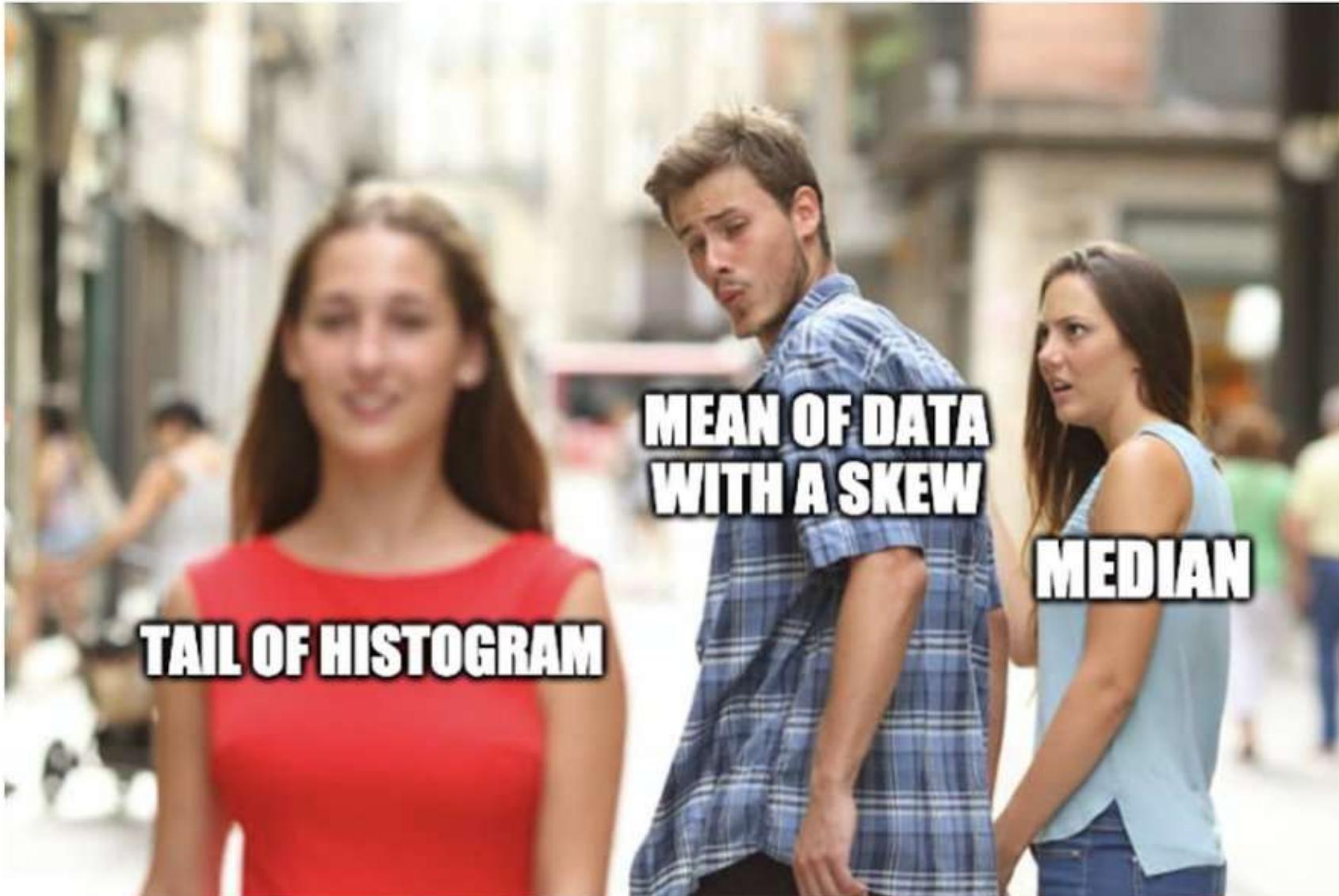
# A tibble: 1 × 2
`Weight Of Sample Before Drying` `Weight Of Sample After Drying`
<chr>                            <chr>
1 5.1 ± 0.2                         0.47 ± 0.14

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

# A tibble: 5 × 2
Measure                               `Median[1Q/3Q]` 
<chr>                                <chr>
1 Weight Of Empty Cup                  4.09 (4.06/4.19)
2 Weigth Of Cup + Sample              9.25 (9.22/9.29)
3 Weigth Of Cup + Sample After Drying 4.61 (4.45/4.72)
4 Moisture Content (%)                91.7 (88.1/93.1)
5 Dry Content (%)                     8.3 (6.9/11.9)

```

# Descriptive Stats should match distribution and data



# Model: Test

Tests require hypotheses



# Null hypothesis ?

- Working hypothesis: This is what you expect!  
E.g. treatment is lowering blood pressure more than placebo,  
transgenic animals become obese, bio reactor A is more efficient than  
B, concentration of substance is correlated with speed of reaction ...
- Null hypothesis: This is what you test!  
No difference / relation, BP under therapy = BP under placebo

# 4 possibilities:

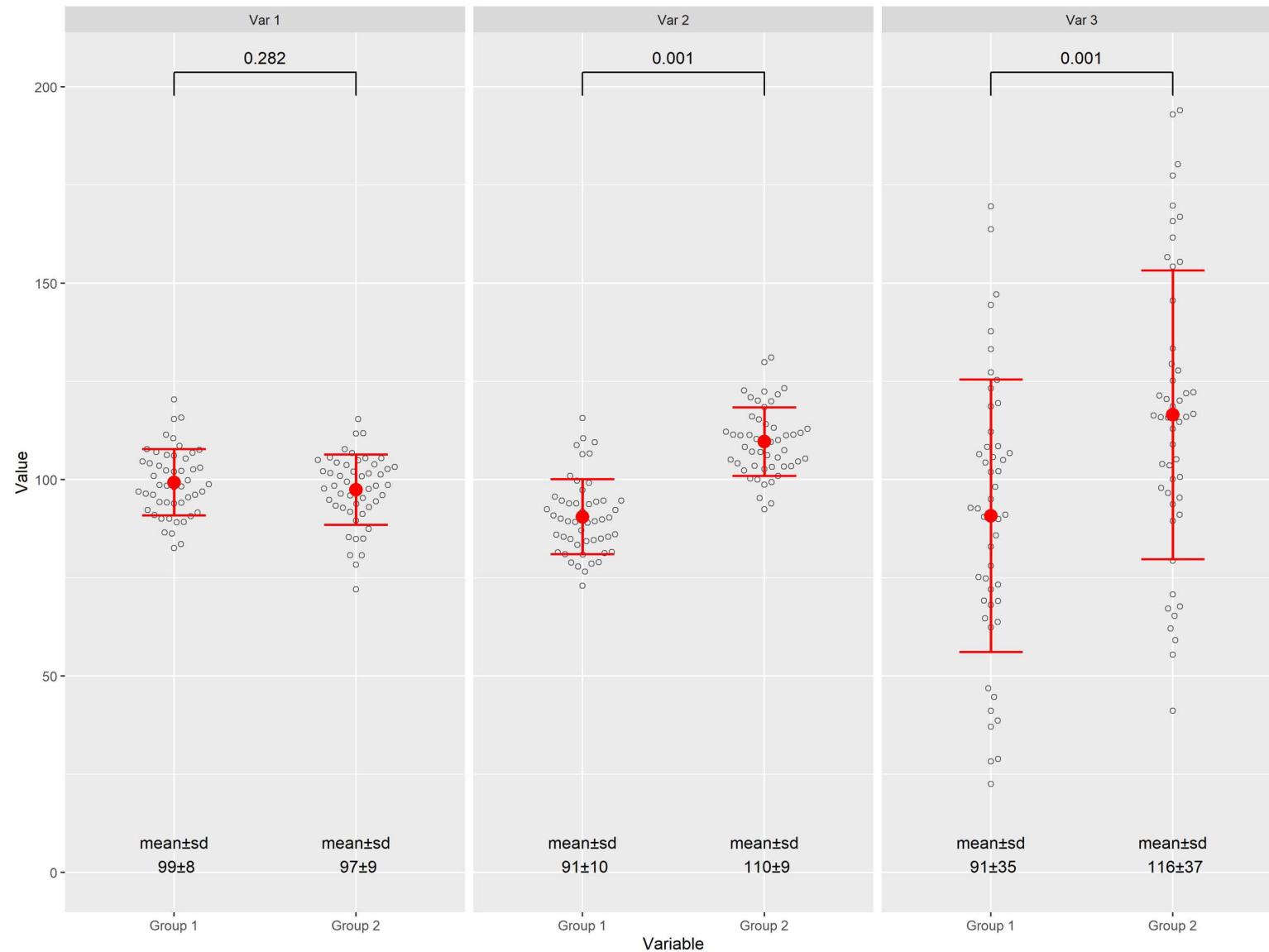
- Null hypothesis correct, test false positive (case A): alpha-error
- Null hypothesis correct, test correct negative (case B)
- Null hypothesis false, test false negative (case C): beta-error
- Null hypothesis false, test correct positive (case D)

**Significance:** NOT probability of case A, but probability of your data given the NULL hypothesis, calculated from your data, conventionally <0.05

**Power:** Probability of case D, *estimated* based on assumptions about effects and sample size, *calculation* would require knowledge of true difference, conventionally set at 0.80

# Test functions

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



# t-test

- Assumptions: Continuous data with Normal distribution
- 1 or 2 (independent or dependent) samples with/without equal variances
- how big is the mean difference relative to uncertainty?  
 $t = (\text{mean}_1 - \text{mean}_2)/\text{SEM}$
- t follows a t-distribution, allows estimation of probability of t under the NULL hypothesis

# Wilcoxon-test

- nonparametric, no distribution is assumed
- based on rank-transformed data
- insensitive to extreme values

# Test examples: *single variables*

```
1 #t-Test with test for equal variances
2 t_out <- t.test(formula=`Weight Of Sample Before Drying`~`Region`,
3                   data=rawdata,
4                   var.equal=var.test(
5                         formula=`Weight Of Sample Before Drying`~`Region`,
6                         data=rawdata)$p.value>.05)
7 t_out
```

Welch Two Sample t-test

```
data: Weight Of Sample Before Drying by Region
t = 1.6528, df = 10.363, p-value = 0.1283
alternative hypothesis: true difference in means between group A and group D is not equal
to 0
95 percent confidence interval:
-0.05025104  0.34436446
sample estimates:
mean in group A mean in group D
      5.155948        5.008891
```

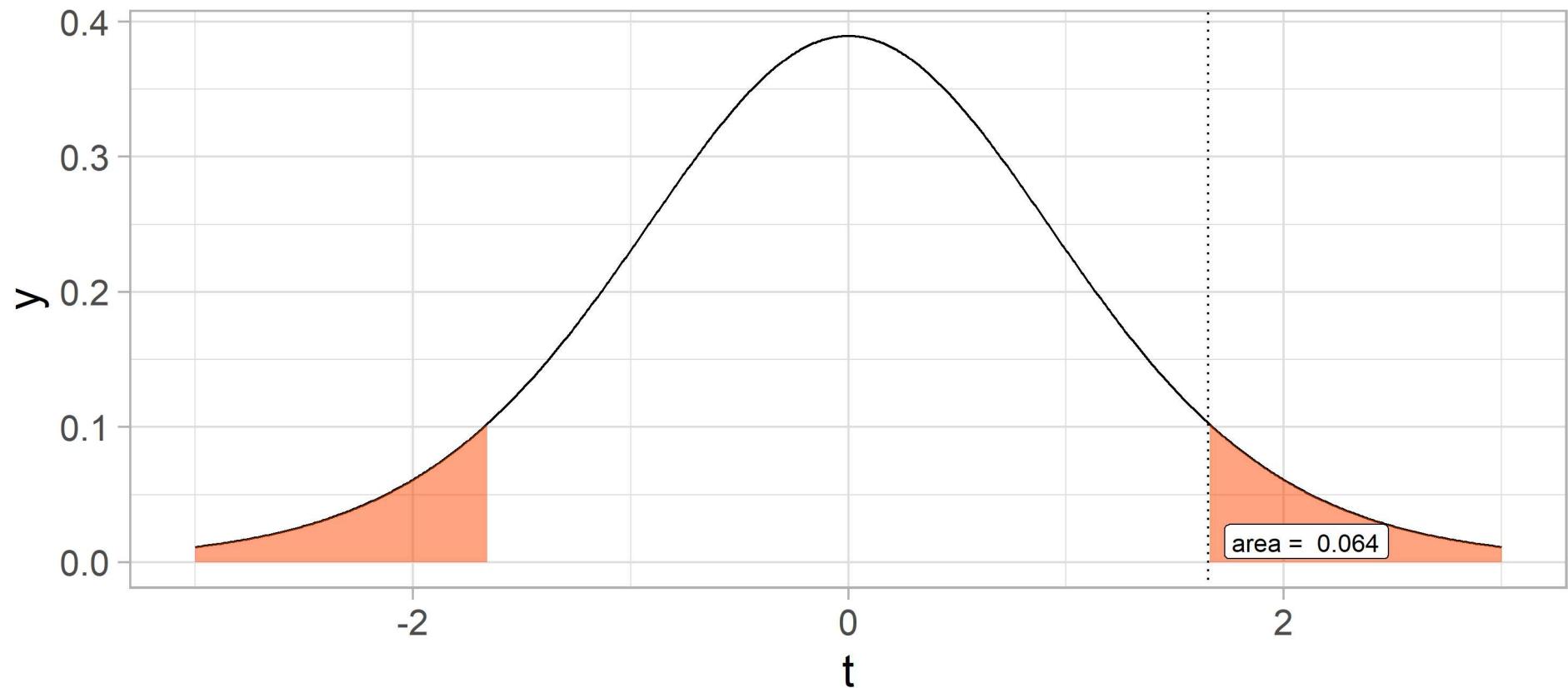
```
1 #Wilcoxon-Test
2 wilcox.test(`Moisture Content (%)`~`Region`,
3             data = rawdata)
```

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
```

# From t to p

from t-test:  $t = 1.7$ ,  $p = 0.128$



# *Model: Test 2 / multiple variables*

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

Variable	desc_all	Region A	Region D	p
Weight Of Sample Before Drying	$5.1 \pm 0.2$	$5.2 \pm 0.1$	$5.0 \pm 0.3$	0.12830
Weight Of Sample After Drying	$0.47 \pm 0.14$	$0.50 \pm 0.16$	$0.42 \pm 0.07$	0.04937

```

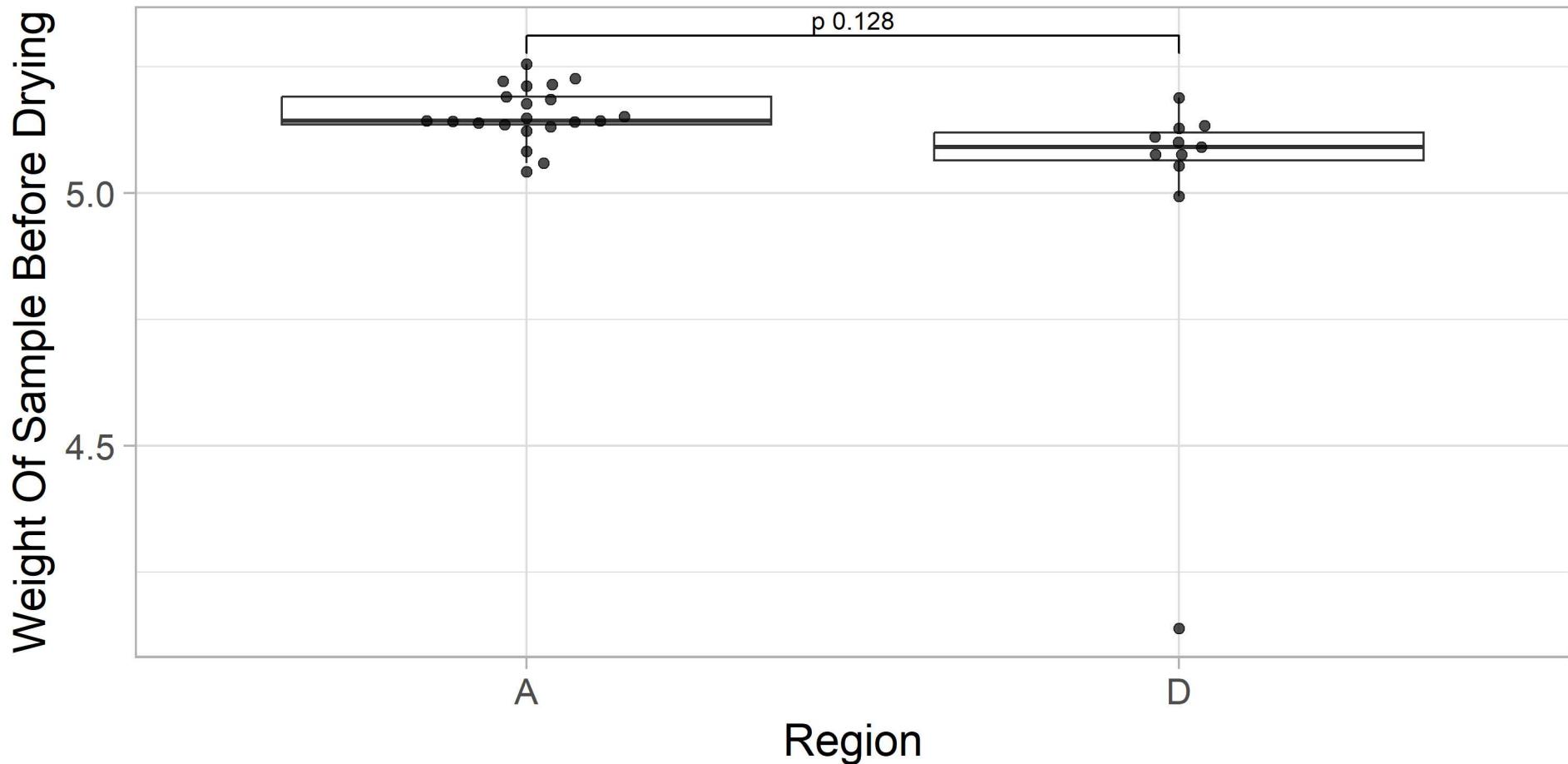
1 test_ord <- compare2numvars(data = rawdata,
2                               dep_vars = ordvars$names,
3                               indep_var = 'Region',
4                               gaussian = FALSE, round_desc = 3)
5 test_ord |> flextable() |>
6   theme_zebra(even_body = 'aquamarine', odd_body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

```

<b>Variable</b>	<b>desc_all</b>	<b>Region A</b>	<b>Region D</b>	<b>p</b>
Weight Of Empty Cup	4.09 (4.06/4.19)	4.07 (4.06/4.09)	4.20 (4.19/4.21)	0.001
Weigth Of Cup + Sample	9.25 (9.22/9.29)	9.24 (9.22/9.25)	9.29 (9.27/9.31)	0.003
Weigth Of Cup + Sample After Drying	4.61 (4.45/4.72)	4.46 (4.42/4.74)	4.62 (4.58/4.66)	0.611
Moisture Content (%)	91.7 (88.1/93.1)	92.6 (86.9/93.1)	91.6 (90.7/92.3)	0.755
Dry Content (%)	8.3 (6.9/11.9)	7.4 (6.9/13.1)	8.4 (7.7/9.3)	0.755

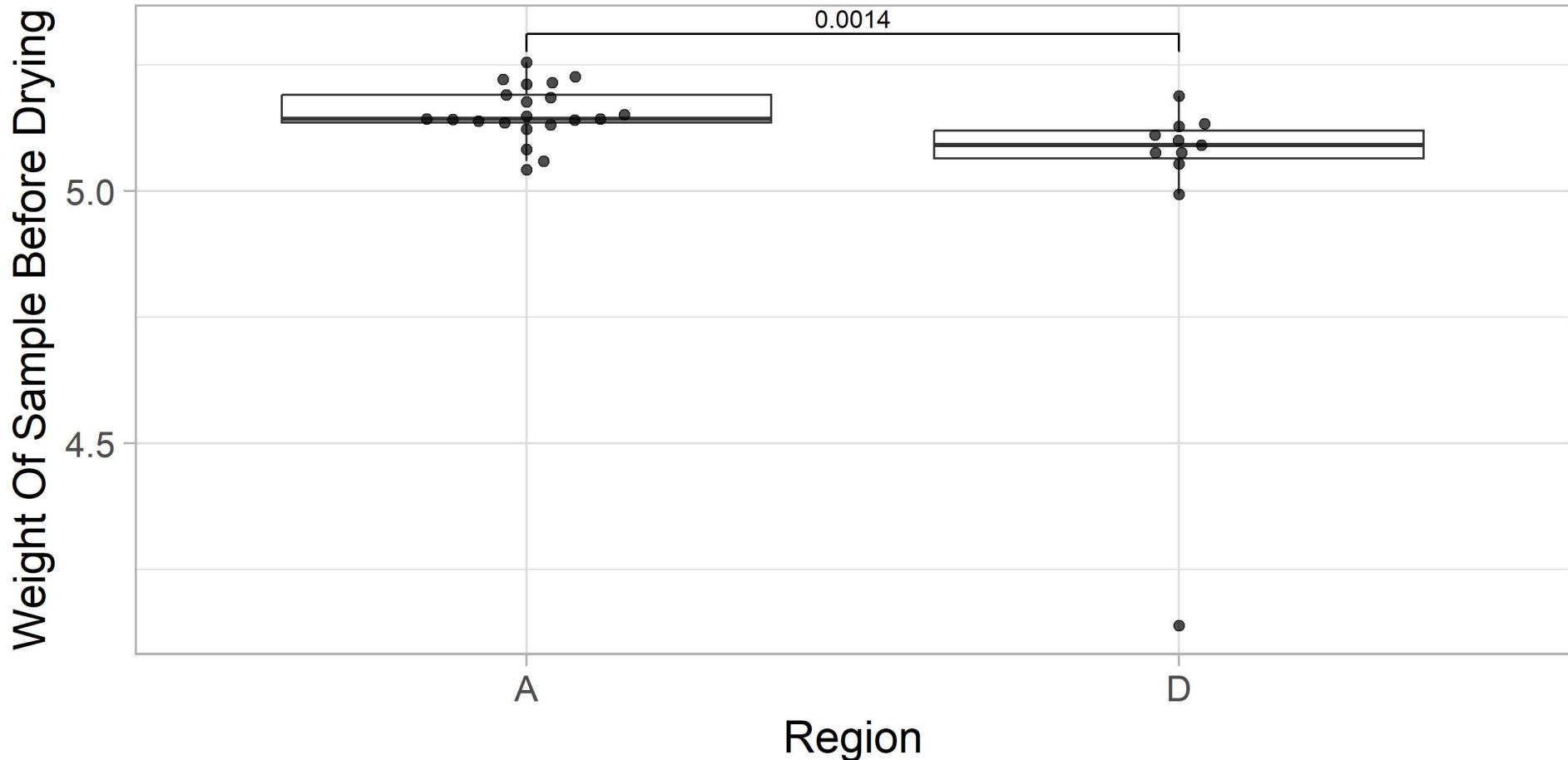
# Show results

```
1 ggplot(rawdata, aes(x = `Region`, y = `Weight Of Sample Before Drying`))+  
2   geom_boxplot(outlier.alpha = 0)+  
3   geom_beeswarm(alpha=.7, size=2,cex = 2)+  
4   geom_signif(comparisons = list(c(1,2)),  
5                 annotations = paste('p',formatP(t_out$p.value)))
```



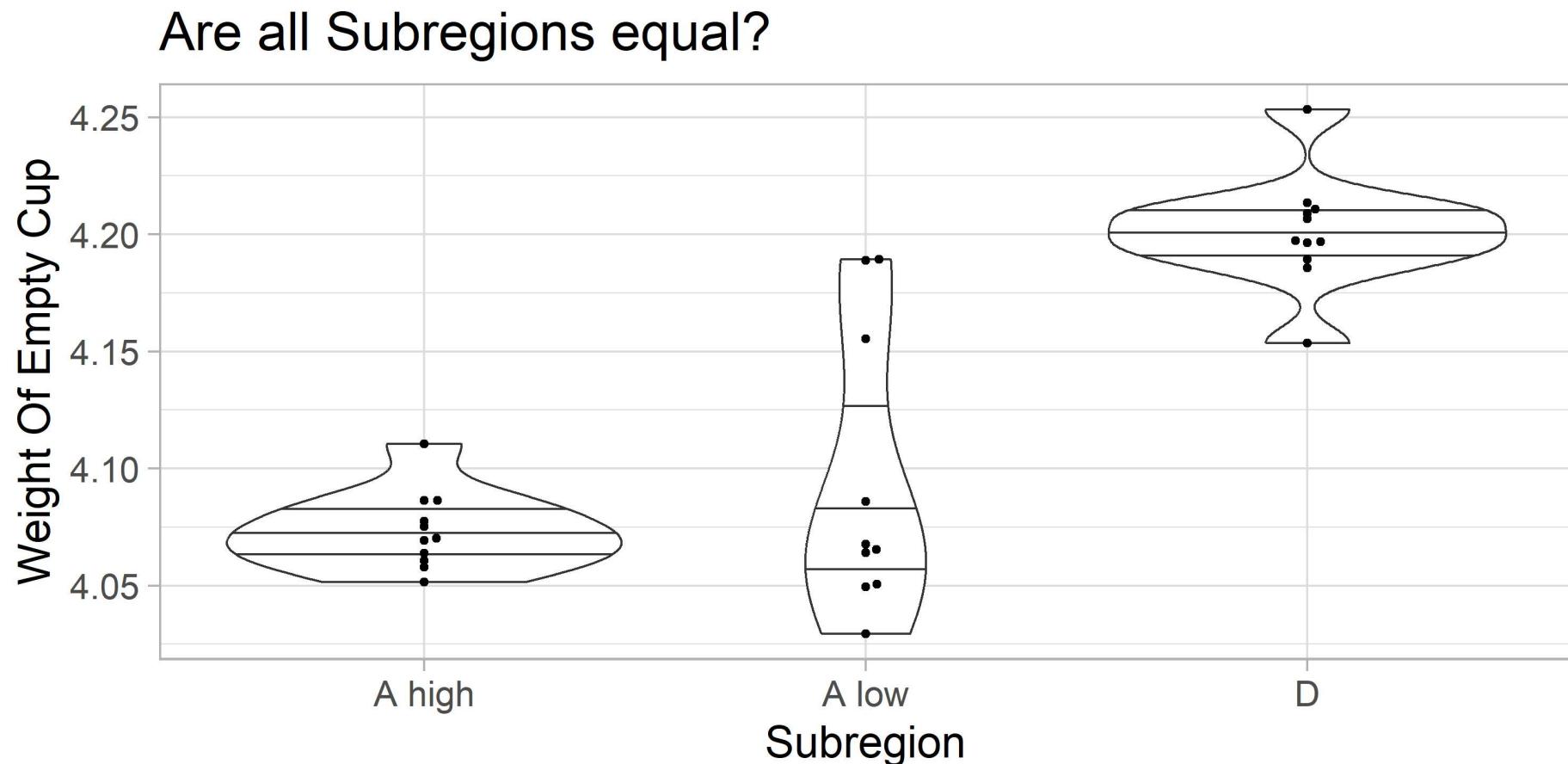
# Re-thinking test decision?

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



# *Model: linear models 1 / univariable*

```
1 plottmp <- ggplot(rawdata,aes(Subregion,`Weight Of Empty Cup`))+  
2   geom_violin(draw_quantiles = c(.25,.5,.75))+  
3   geom_beeswarm()  
4  
5   ggttitle('Are all Subregions equal?')  
6   print(plottmp)
```



# ANOVA: build model

```
1 rawdata |> group_by(Subregion) |>
2   summarize(MeanCup=mean(`Weight Of Empty Cup`) |> roundR(4)) |>
3   pivot_wider(names_from = Subregion,values_from = MeanCup) |>
4   rename_with(~paste('Mean weight empty cups\n', .x)) |> flextable() |>
5   theme_zebra(even_body = 'aquamarine',odd_body = 'antiquewhite')
```

	Mean weight empty cups	Mean weight empty cups	Mean weight empty cups
A high	A low	D	
4.074	4.095	4.201	

```
1 lm1<- lm(`Weight Of Empty Cup`~Subregion, data=rawdata)
2 lm1
```

Call:

```
lm(formula = `Weight Of Empty Cup` ~ Subregion, data = rawdata)
```

Coefficients:

(Intercept)	SubregionA	low	SubregionD
4.07358		0.02105	0.12754

# ANOVA: get p-values

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

term	df	sumsq	meansq	statistic	p.value
Subregion	2	0.10200725	0.051003623	36.19695	0.00000001310989
Residuals	29	0.04086269	0.001409058		

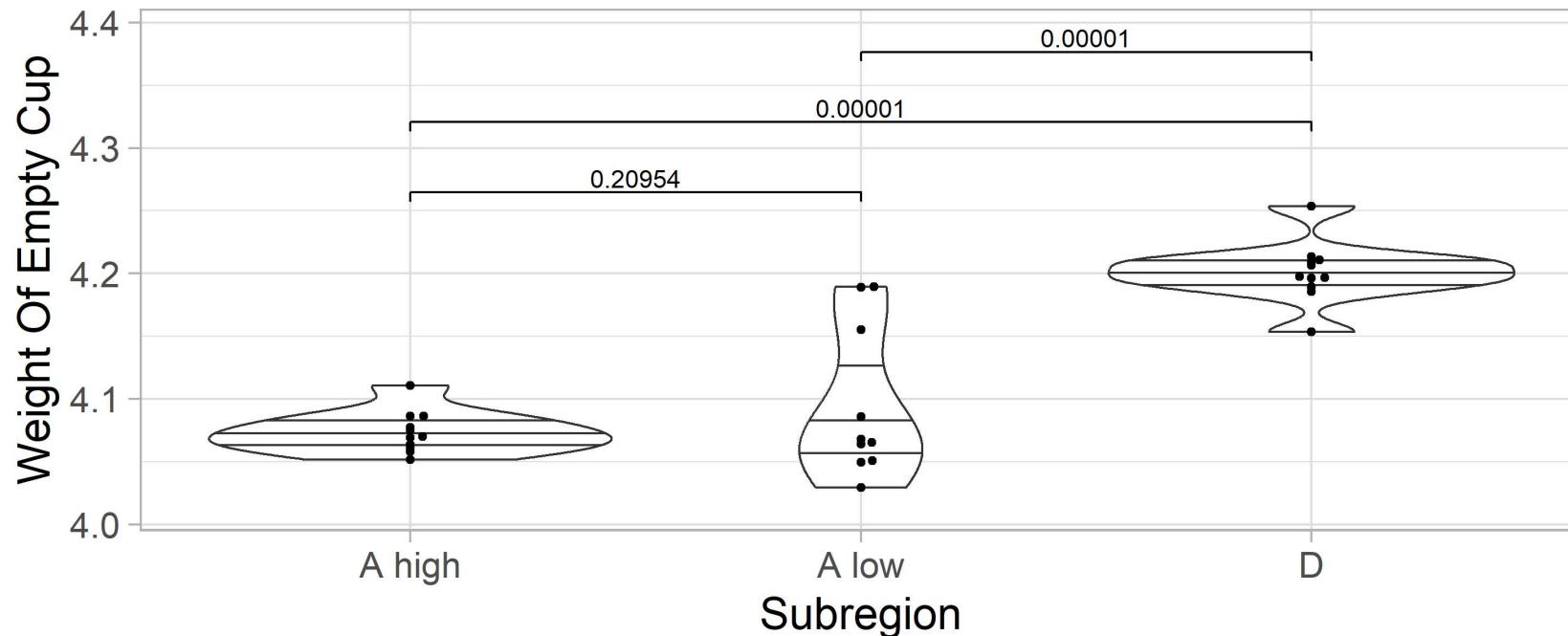
```
1 #post-hoc  
2 (posthoc_out <- pairwise.t.test(x = rawdata$`Weight Of Empty Cup`,  
3  
4  
5 p.adjust.method = 'fdr')$p.value |>  
formatP(ndigits = 5))
```

	A high	A low
A low	"0.20954"	NA
D	"0.00001"	"0.00001"

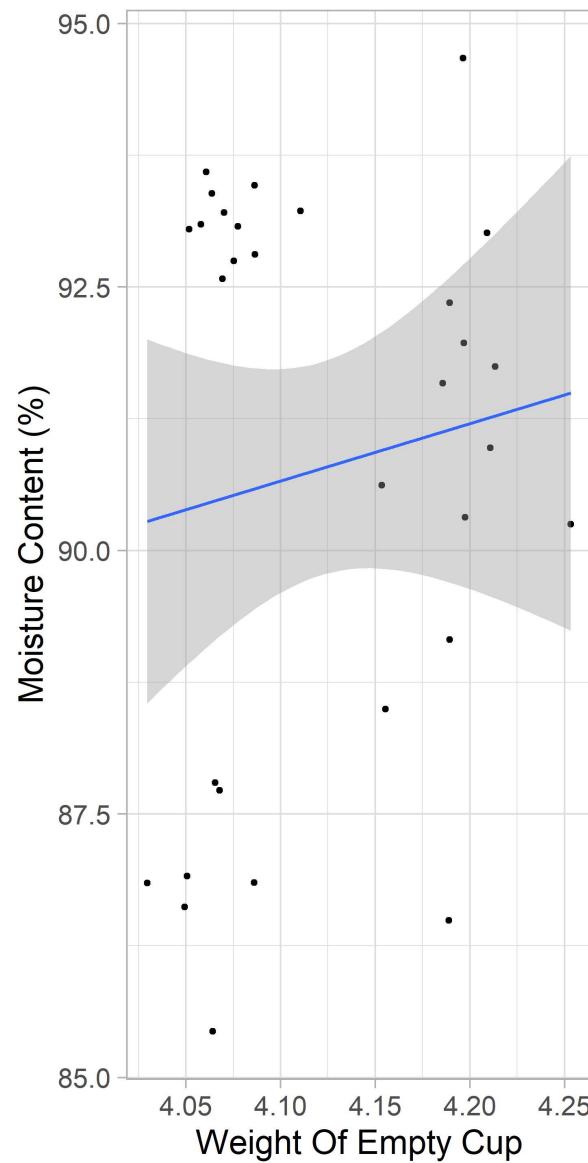
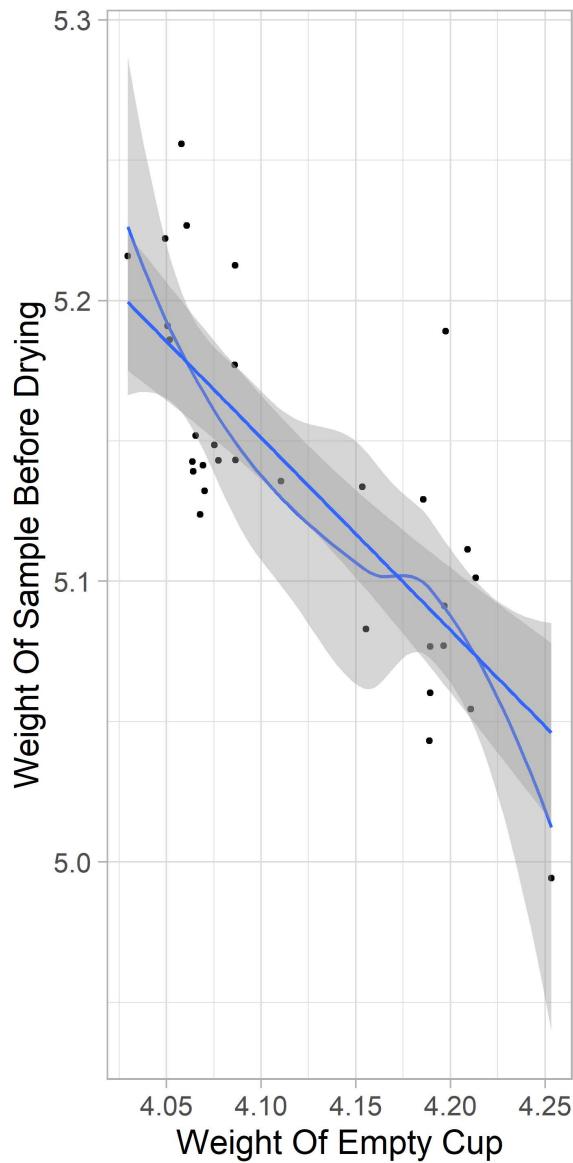
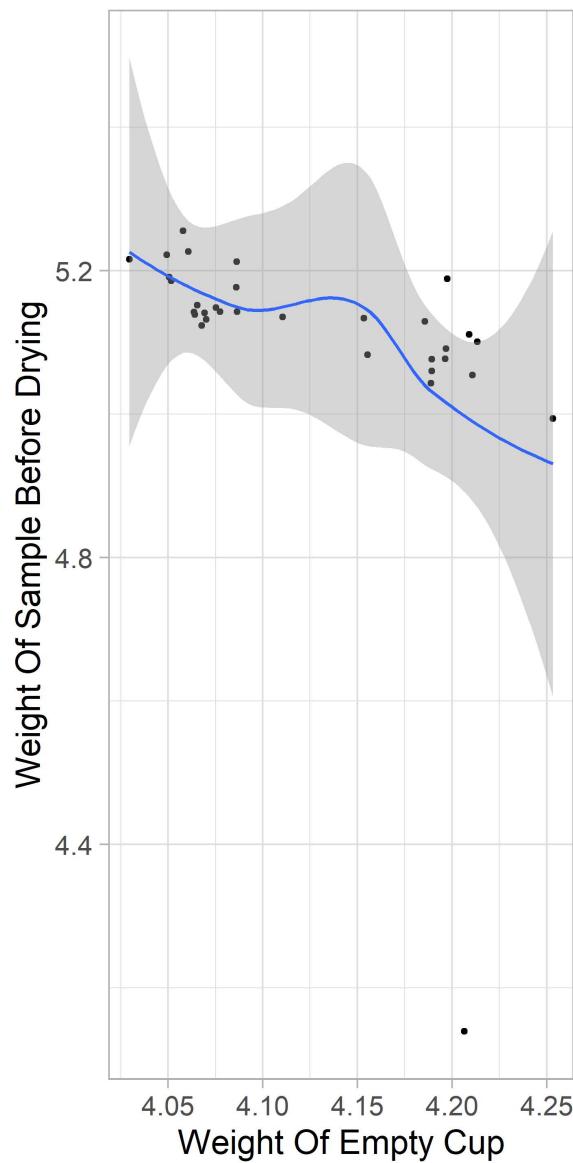
# Visualize ANOVA

```
1 ggplot(rawdata,aes(Subregion,`Weight Of Empty Cup`))+  
2   geom_violin(draw_quantiles = c(.25,.5,.75))+  
3   geom_beeswarm() +  
4   geom_signif(comparisons = list(c(1,2),c(1,3),c(2,3)),  
5                 annotations = c(posthoc_out[,1], posthoc_out[2:2]),  
6                 step_increase = .25) +  
7   scale_y_continuous(expand = expansion(mult = .1)) +  
8   ggttitle('All Subregions are equal,\nbut some Subregions are more equal than others')
```

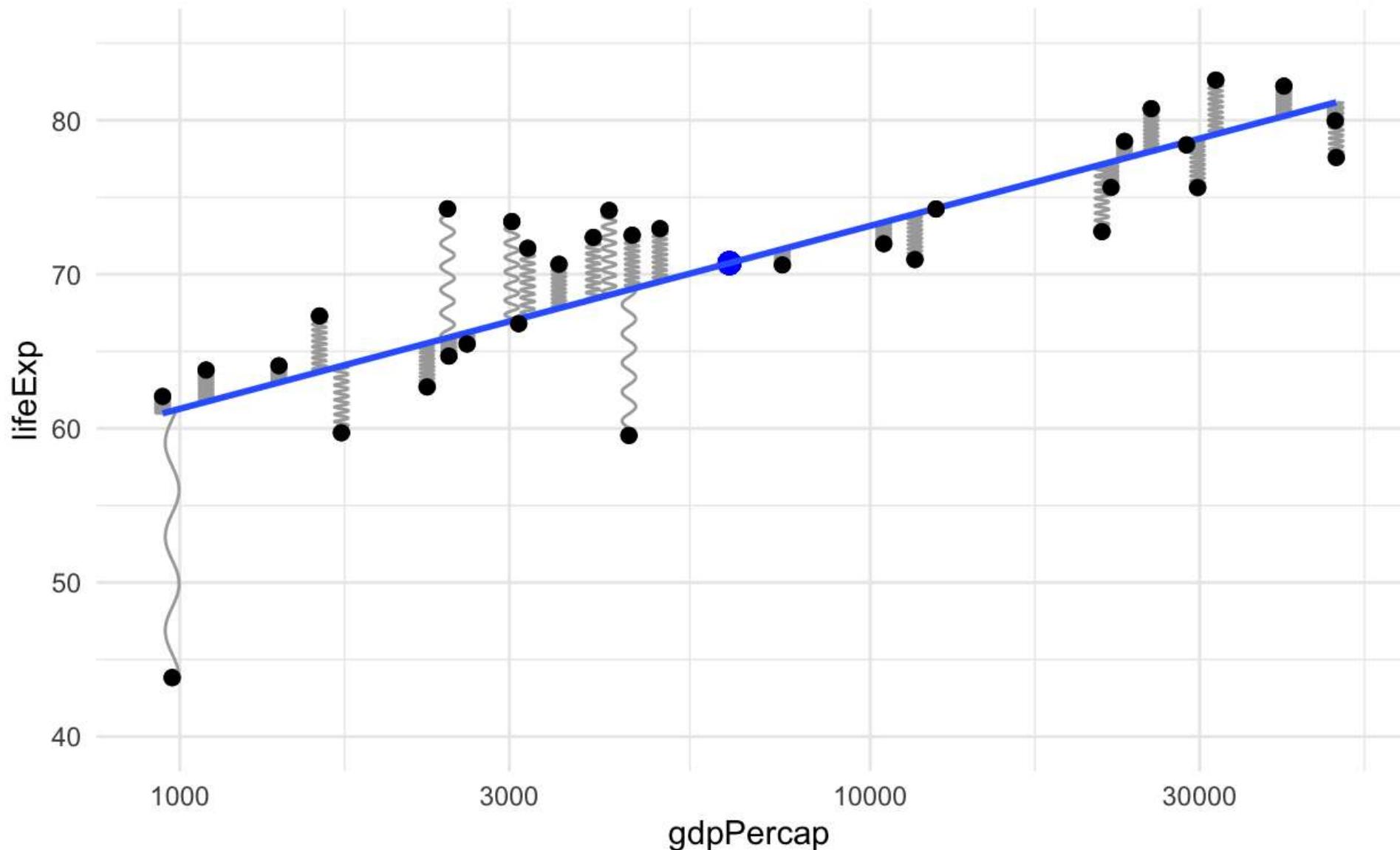
All Subregions are equal,  
but some Subregions are more equal than others



# Regression: Scatterplot



# Regression: Underlying mechanics



# Regression: Statistics

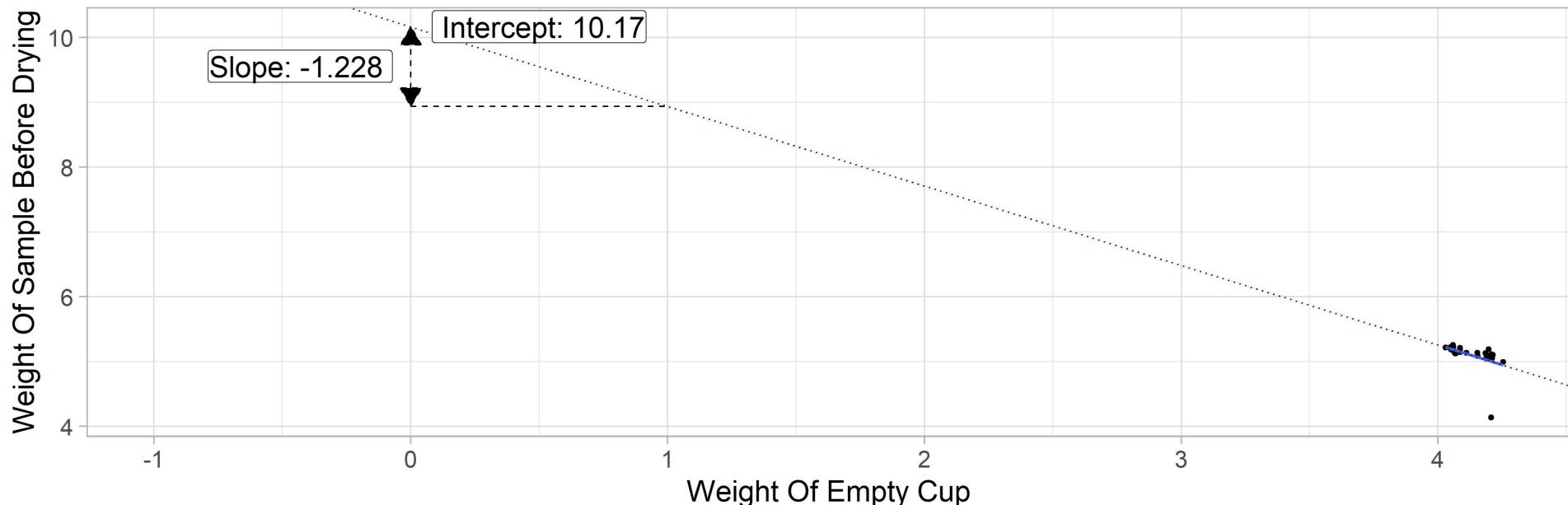
```
1 lm_out <- lm(`Weight Of Sample Before Drying` ~ `Weight Of Empty Cup`,  
2                  data=rawdata)  
3 lm_out
```

Call:

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

Coefficients:

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



# Regression: Significance

```
1 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  0.215  0.215      7.56  0.0100
2 Residuals          30  0.855  0.0285      NA     NA
```

```
1 model_parameters(lm_out)
```

Parameter	Coefficient	SE	95% CI	t(30)	p
(Intercept)	10.17	1.84	[ 6.41, 13.93]	5.52	< .001
Weight Of Empty Cup	-1.23	0.45	[-2.14, -0.32]	-2.75	0.010

# *Report*

- *RMarkdown and quarto are powerful tools to create reports and presentations*
- Export figures: `ggsave()` / `png()` / `pdf()`
- Export tables: `write_xlsx()`
- *Package `flextable` provides nice features for table formatting*

# Flextable example

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

Variable	Region A	Region D	p
Weight Of Empty Cup	4.07 (4.06/4.09)	4.20 (4.19/4.21)	0.001
Weigh Of Cup + Sample	9.24 (9.22/9.25)	9.29 (9.27/9.31)	0.003
Weigh Of Cup + Sample After Drying	4.46 (4.42/4.74)	4.62 (4.58/4.66)	0.611
Moisture Content (%)	92.6 (86.9/93.1)	91.6 (90.7/92.3)	0.755
Dry Content (%)	7.4 (6.9/13.1)	8.4 (7.7/9.3)	0.755

Significance level is set at 0.05

# Useful tools along the way

- Pick columns / rows: `select()` / `pull()` / `filter()` / `slice()`
- Change format of tibble wide <--> long (e.g. for repeated measures):  
`pivot_longer()`/`pivot_wider()`
- Regular expressions: `str_replace()` / `str_detect()` / `str...`
- Merge text elements: `paste()` / `str_glue()`
- Apply functions: `purrr::map_xxx`

