Cytotoxicity on Osteosarcoma Cells and Osteoblasts

Julius Kasiske

2022-09-06

1. Motivation and Procedure

In medicine, occasional needs for artificially grown bone structures may arise, yielding a need to research best possible materials to use, such that optimal solutions can be fielded across care units.

This research compares seven materials growing on osteosarcoma cells and osteoblasts. The measurements and lab research was conducted by my colleague, a medical student, who measured multiple extinction values in each group across eight different scaffolds and then recorded means and standard deviations across these scaffolds and measurements. After she generated the necessary data, I had three key tasks:

1. Data Wrangling, where I needed to tidy and manipulate data sets in order to properly plot the results and test them against null hypotheses.
2. Data Visualization, which encompassed choosing appropriate plots and building a custom ggplot2-theme.
3. Hypothesis Testing, including validating various assumptions that different testing methods make and choosing the most appropriate one based on these assumptions holding in each particular case.

As in this research, I was a mere statistical extension of the medical research and am not qualified to speak to medical implications, I will focus on elaborating on select portions of my work in data wrangling, plotting and testing instead of on the medical implications.

The research was split into two lab experiments.

2. Lab Experiment One

In the first lab experiment, measures were recorded for different concentrations of the respective extract.

2.1 Data Wrangling

After loading an Excel-sheet into R, the table initially looked as follows:

```
##           material      mean         sd concentration
## 1: PEKK 4h Plasma 0.9213026 0.10394363          1.00
## 2: PEKK 4h Plasma 1.0195472 0.20766995          0.75
## 3: PEKK 4h Plasma 0.9992650 0.05359917          0.50
## 4: PEKK 4h Plasma 1.1155162 0.04678242          0.25
## 5:        So_PPSU 0.9672094 0.04762213          1.00
## 6:        So_PPSU 0.9371325 0.06345083          0.75
## 7:        So_PPSU 0.9096280 0.04641385          0.50
```

It shows for each material and for each concentration, the mean and standard deviation across the eight scaffolds. In this experiment, the data wrangling work was limited in R, as I did a lot of it in Excel before loading the table into R. However, for illustrative purposes, I chose to represent the `concentration` column as percentages instead of decimals. For that, I coded the following conversion method and applied it. The following is an excerpt from the code:

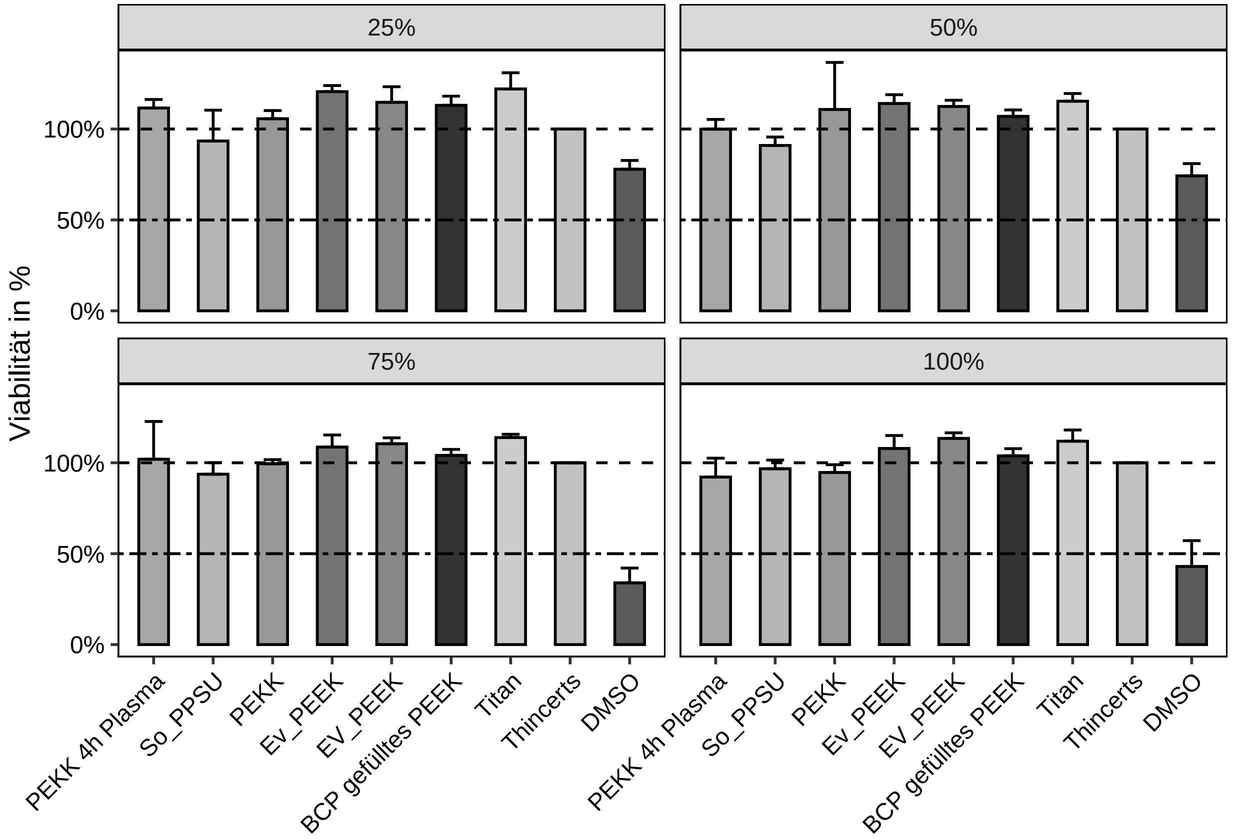
```r
## 1.3 change how concentration column is displayed
### 1.3.1 code conversion method
conversion1 <- function(decimal){
  output <- rep(NA, length(decimal))
  for(i in 1:length(decimal)){
    if(decimal[i] == 1.00) output[i] <- "100%"
    if(decimal[i] == 0.75) output[i] <- "75%"
    if(decimal[i] == 0.50) output[i] <- "50%"
    if(decimal[i] == 0.25) output[i] <- "25%"
  }
  return(output)
}
### 1.3.2 apply method to new vector and add vector to data
concentration_neu <- conversion1(data1[, concentration])
data1[, concentration := concentration_neu]
```

2.2 Data Visualization

Now, I wanted to display the means of the viablity values alongside their respective standard deviations for each material and concentration. For that, I first added a column for the `geom_errorbar` function to know the standard deviation. I then used the following ggplot2 code:

```r
# 2 PLOTTING VIABILITY
## 2.1 add ymax for errorbars in data
data1[, ymax := mean + sd]
## 2.2 plot the result
p1 <- ggplot(data1) +
  ### 2.2.1 geometric objects and faceting
  geom_col(aes(reorder(material, number), mean, fill = material),
           color = "black", width = 0.5) +
  geom_errorbar(aes(x = reorder(material, number), ymin = mean, ymax = ymax, width = 0.3)) +
  facet_wrap(~ factor(concentration,levels = c("25%", "50%", "75%", "100%"))) +
  ### 2.2.2 axis modifications and hlines
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.text = element_text(colour = "black")) +
  labs(x = NULL, y = "Viabilität in %") +
  scale_y_continuous(labels = scales::percent) +
  geom_hline(yintercept = 1, linetype = "dashed") +
  geom_hline(yintercept = 0.5, linetype = "twodash") +
  ### 2.2.3 general theme
  theme_l +
  theme(legend.position = "none") +
  scale_fill_grey()
```

The above code then rendered the following plot:

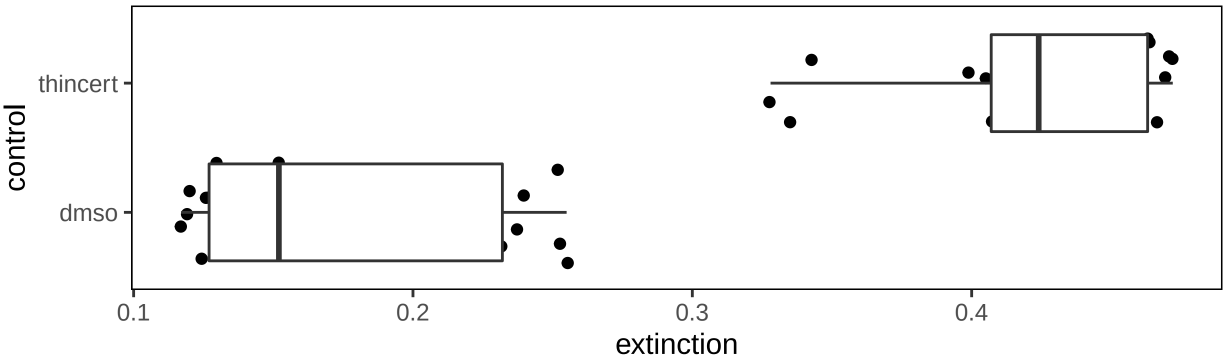


2.3 Statistical Testing

Note that we have seven materials of interest, but nine are displayed in the facet plot. The two on the right are for controls. `Thincert` is a positive control with perfect viability of 100% whereas `DMSO`, standing for dimethyl sulfoxide, is a toxin that serves as a negative control. I was tasked with testing the significance of the difference between the means for DMSO and Thincerts for 100% concentration. Thus I would perform a two-sample or A/B test for the group mean difference and thus implicitly test the association between the variable signaling DMSO/Thincert class membership and the variable containing all the means. My first task was again to load and tidy the data, with the final table for assumption validation and later testing looking as follows:

```
##      control extinction
## 1:      dmso      0.217
## 2:      dmso      0.220
## 3:      dmso      0.223
## 4: thincert      0.463
## 5: thincert      0.466
## 6: thincert      0.469
```

In two-sample testing, primal assumptions to test are bivariate distributions of groups. In the general case of the two sample t test, strict gaussianity must hold. Additionally, the t test assumes equal variance in both groups. Whats more, should mean difference be driven by only a few outliers, it might make sense to use a wilcoxon rank-sum test. Thus, I first checked gaussianity and outlier existence to confirm applicability of said parametric methods and to possibly exlude the wilcoxon rank-sum test.



The above plot indicates a lack of extreme outliers and while not confirming normality, they indicate a distribution that is close enough to normal for me to apply parametric methods in this specific case. In addition, I now had to test the equal variance assumption in order to decide between the welsh test and the t test.

```
##     control standard_dev
## 1:     dmso   0.05509853
## 2: thincert   0.04300266
```

Due to the similarity of standard deviations across groups, a t test is appropriate. I will however perform both in order to confirm my choice. Testing in R yields the following p values:

$p_w = 3.5267 \cdot 10^{-18}$ for the welsh test

$p_t = 9.7316 \cdot 10^{-19}$ for the t test

Both tests indicate significance at any reasonable significance level.

3. Lab Experiment Two

In the second experiment, the extinction values were recorded for multiple days after application of the materials. Measures were taken two, five and seven days after.

3.1 Data Wrangling

Again, I initially loaded an Excel-file into R, which looked as follows at first:

```
##              material       sd_0    mean_0       sd_2    mean_2       sd_5
## 1:     PEKK 4h Plasma 0.09151509 1.5217083 0.08611313 0.7134583 0.12589025
## 2:            So_PPSU 0.09285529 1.1945833 0.22398478 0.4474167 0.05386826
## 3:               PEKK 0.15783324 1.5308750 0.07518657 0.7654167 0.11163705
## 4:            EV_PEEK 0.19297182 0.8862083 0.04578207 0.1981250 0.01769396
## 5: BCP gefülltes PEEK 0.08421978 1.1196875 0.11882071 0.3135000 0.07740190
## 6:              Titan 0.23006969 0.9575000 0.12904772 0.3326667 0.09896966
##        mean_5       sd_7     mean_7
## 1: 0.22891667 0.16097863 0.16608333
## 2: 0.08804167 0.03449555 0.11045833
## 3: 0.13192500 0.24330289 0.18125000
## 4: 0.03791667 0.01093669 0.01616667
## 5: 0.11687500 0.09314467 0.11329167
## 6: 0.15125000 0.03435009 0.15529167
```

The data set shows means and standard deviations for each material for all days. For visualizing the data, I needed to have the table in a long format w.r.t. the `mean` and `sd` columns and have access to the respective day, thus creating a new variable for days. I did so by first melting the table and then separating the resulting columns. After that first step and running the following code:

```
## 1.2 data wrangling
data_ob1.1 <- melt(data = data_ob1, id.vars = c("material"),
                   variable.name = "value", value.name = "extinktion") %>%
  separate(., col = value, into = c("measure", "day"))
```

the table looked like this:

```
##               material measure day extinktion
## 1:     PEKK 4h Plasma      sd   0 0.09151509
## 2:            So_PPSU      sd   0 0.09285529
## 3:               PEKK      sd   0 0.15783324
## 4:            EV_PEEK    mean   7 0.01616667
## 5: BCP gefülltes PEEK    mean   7 0.11329167
## 6:              Titan    mean   7 0.15529167
```

I now wanted to have individual variables for the means and the standard deviations. I thus recasted the table w.r.t. the `measure` column. I then added a column for the errorbars in the plot and one that indicates the order in which I want to display the materials later on. I ran the following code:

```
data_ob1.2 <- dcast(data_ob1.1, ... ~ measure, value.var = "extinktion")
data_ob1.2[, c("ymin", "ymax") := .(mean-sd, mean+sd)]
data_ob1.2[, number := c(rep(6, 4), rep(5, 4), rep(4, 4), rep(3, 4), rep(1, 4),
                         rep(2, 4), rep(7, 4))]
```

That finally yielded the inteded table:

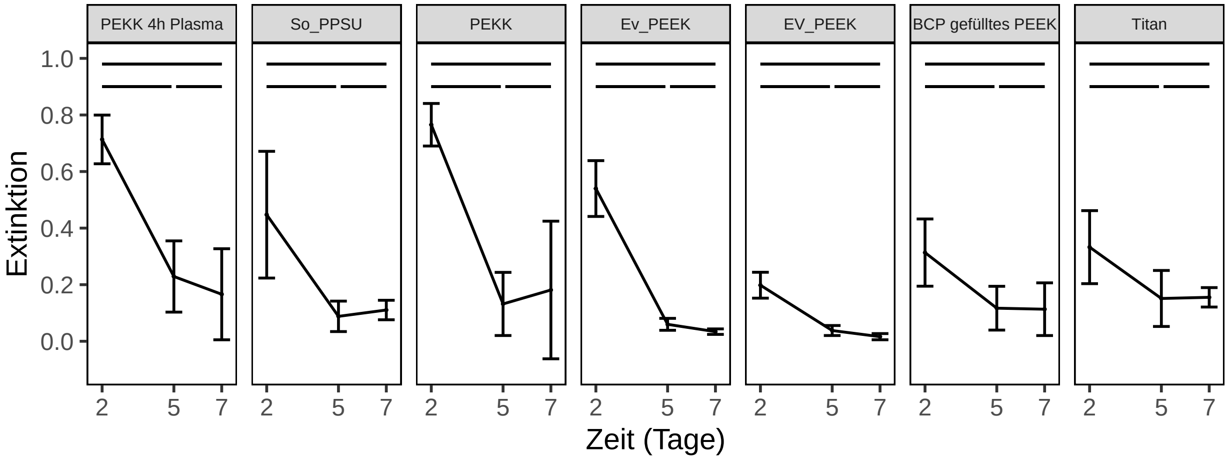
```
##               material day      mean         sd       ymin      ymax number
## 1: BCP gefülltes PEEK   0 1.1196875 0.08421978 1.03546772 1.2039073      6
## 2: BCP gefülltes PEEK   2 0.3135000 0.11882071 0.19467929 0.4323207      6
## 3: BCP gefülltes PEEK   5 0.1168750 0.07740190 0.03947310 0.1942769      6
## 4:              Titan   2 0.3326667 0.12904772 0.20361894 0.4617144      7
## 5:              Titan   5 0.1512500 0.09896966 0.05228034 0.2502197      7
## 6:              Titan   7 0.1552917 0.03435009 0.12094157 0.1896418      7
```

3.2 Data Visualization

For the resulting plot, I wanted the increases/decreases across the days to be visible for each material, while showing the respective standard deviations. I thus decided on a facetted line plot with errorbars, here is an excerpt from the code that created the plot:

```
# 2. PLOTTING
p2 <- ggplot(data_ob1.2[day != "0"], color = "black") +
  ## 2.1 geometric objects
  geom_point(aes(as.numeric(day), mean),
             shape = 1, size = 0.1, fill = "black") +
  geom_line(aes(as.numeric(day), mean)) +
  geom_errorbar(aes(x = as.numeric(day), ymin = ymin, ymax = ymax), width = 0.7) +
  facet_wrap(~factor(material, levels = data_ob1.2[order(number), unique(material)]),
             nrow = 1) +
  ## 2.2 axis modifications
  scale_x_continuous(breaks = c(2, 5, 7)) +
  scale_y_continuous(breaks = c(0, 0.2, 0.4, 0.6, 0.8, 1), limits = c(-0.1,1)) +
  labs(x = "Zeit (Tage)", y = "Extinktion") +
  ## 2.3 theme and crossbars for significance
  theme_l +
  geom_segment(aes(y = 0.9, yend = 0.9, x = 2, xend = 4.9), color = "black") +
  geom_segment(aes(y = 0.9, yend = 0.9, x = 5.1, xend = 7), color = "black") +
  geom_segment(aes(y = 0.98, yend = 0.98, x = 2, xend = 7), color = "black")
```

This was the final result:

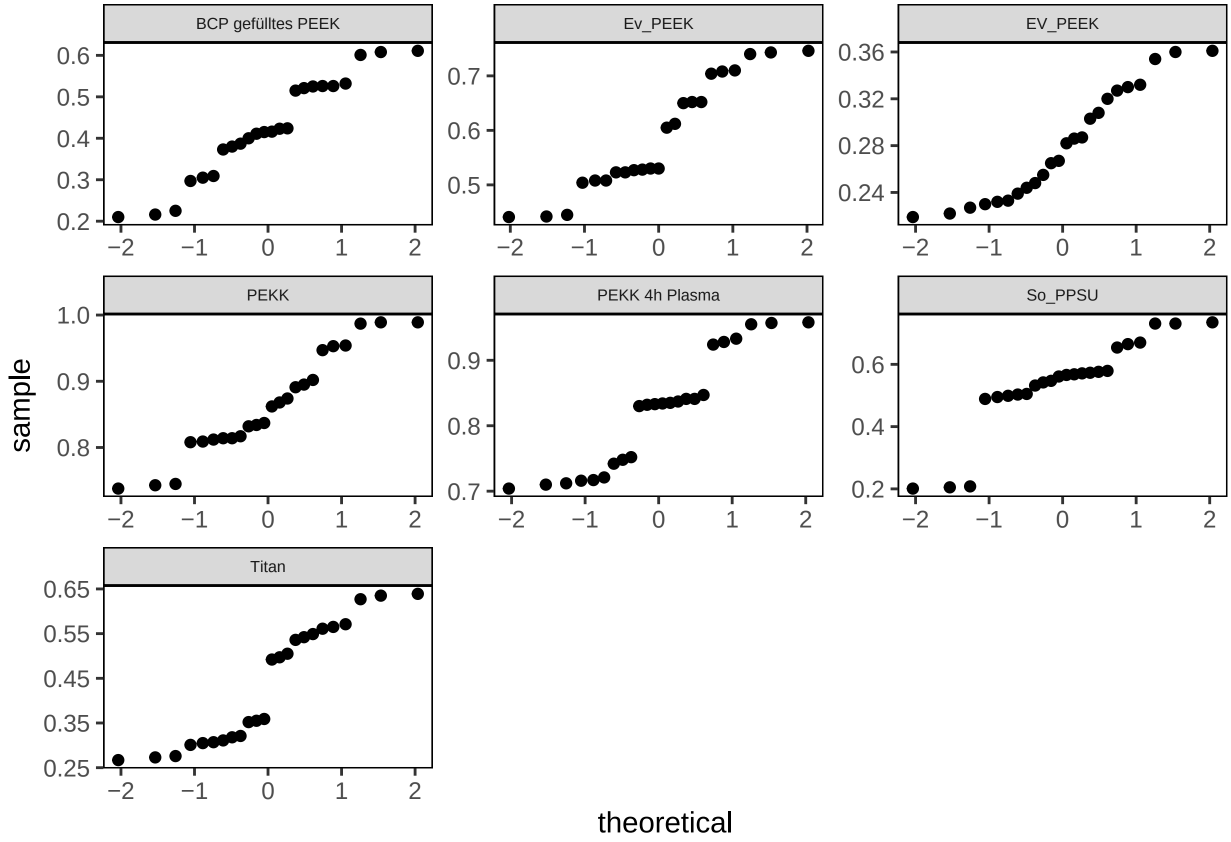


Note that I added vertical lines for each across-day difference. These are there s.t. I can add asterisks for significant differences later after I tested each difference.

3.3 Statistical Testing

As mentioned above, I want to test the significance of group mean differences across days for all materials. I want to test three differences for each material, from day two to day five, day five to day seven and day two to day seven, for seven materials. I thus performed $3 \cdot 7 = 21$ tests, I will however only showcase a subset of those for the sake of simplicity.

My first step was again to choose the correct testing method by evaluating if the assumptions that each method makes hold or not. As I mentioned, for the parametric tests that I know of to be applicable, both groups must be normally distributed, an assumption I tested via qq-plots.



The above plot shows the qq-plots for day two. For no material, a normal distribution seems to exist, as its tails seem to be heavier as would be hypothesized. The same holds for the other days. Consequentially, parametric methods would not be useful. Thus I had to use non-parametric (monte carlo) permutation testing, effectively creating a distribution under the null by permuting.

For permutation testing, I initially had to code a method that computes the group mean difference after each permutation. I coded the method to work on data.tables:

```
mean_diff <- function(dt, ear, lat){
  diff <- dt[, .(mean = mean(extinktion, na.rm = T)), by = day][day == ear, mean] -
    dt[, .(mean = mean(extinktion, na.rm = T)), by = day][day == lat, mean]
  return(diff)
}
```

I then needed to code a method that would execute the permutations, using the `mean_diff` function. The function needs input paramters for material, the two days for the difference computation and one default parameter `m` for the number of permutations, here 1000.

```
permutation_test <- function(mat, ear, lat, m = 1000){
  mean_res <- rep(NA, m)
  data <- data_ob2.2[day %in% c(ear, lat) & material == mat]
  # original mean difference
  diff_org <- mean_diff(data, ear, lat)
  # resample and compute mean difference
  for(i in 1:m){
    mean_res[i] <- mean_diff(data[, day := sample(day)], ear, lat)
  }
  # p value computation
  p_value <- (length(mean_res[abs(mean_res) > abs(diff_org)]) + 1) / (m + 1)
  # plot result
  plot <- ggplot(NULL, aes(mean_res)) +
    geom_histogram(fill = "grey", color = "black") +
    geom_vline(xintercept = diff_org, linetype = "dashed") +
    annotate(geom = "text", x = diff_org, y = 25,
             label = round(diff_org, 3), size = 2) +
    ggtitle(paste0("p-Wert: ",
                   round(p_value, 7), "\nfor ",
                   ear," on ", lat, " for ", mat)) +
    theme_l +
    theme(plot.title = element_text(size = 8),
          axis.title = element_blank())

  print(paste0(mat, ": ", p_value))
  return(plot)
}
```

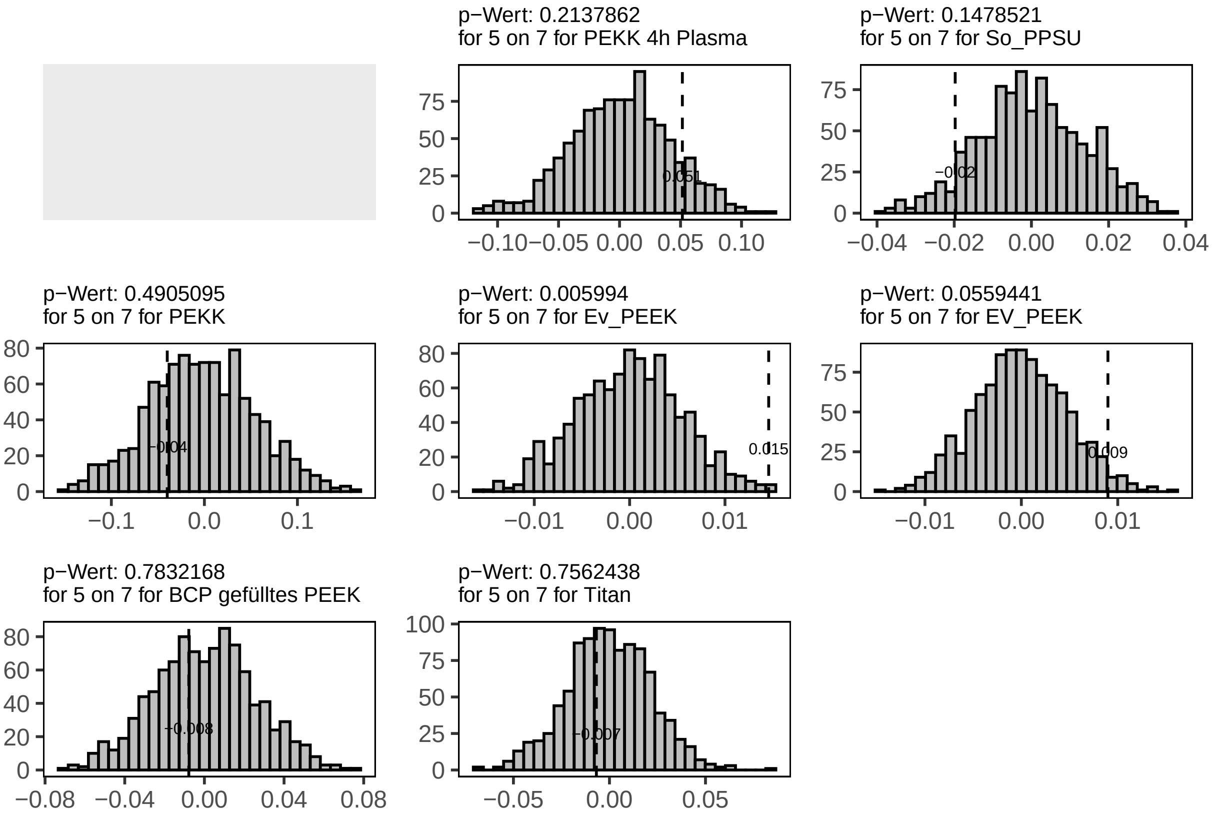
Note that this method first computes the respective T_{obs}, then resamples the `day` column in the subsetted data set to randomly permute the results and then computes that same mean difference after each resampling. In that way, the method creates a distribution that is later plotted alongside the T_{obs}. Then, p values are computed for two-sided testing. The p values are computes as follows:

$p = \frac{n+1}{m+1}$, where $n = p(|\Delta| > |T_{obs}|)$

Also note that this method only tests one material at a time, giving rise to a need to incorporate that method into one that loops over all materials at once. My code excerpt is down below:

```
### 3.4.4 write "meta" testing method
meta_permute <- function(ear, lat){
  mats <- as.vector(data_ob2.2[, unique(material)])
  pl <- ggplot()
  for(i in 1:length(mats)){
    output <- permutation_test(mats[i], ear, lat)
    pl <- pl + output
  }
  return(pl)
}
```

As the `permutation_test` method returned the plots, `meta_permute` now aggregates the individual plots and returns all necessary information on all materials for one inter-day difference specified. Below is the output for day five to day seven.



p−Wert: 0.2137862
for 5 on 7 for PEKK 4h Plasma

p−Wert: 0.1478521
for 5 on 7 for So_PPSU

p−Wert: 0.4905095
for 5 on 7 for PEKK

p−Wert: 0.005994
for 5 on 7 for Ev_PEEK

p−Wert: 0.0559441
for 5 on 7 for EV_PEEK

p−Wert: 0.7832168
for 5 on 7 for BCP gefülltes PEEK

p−Wert: 0.7562438
for 5 on 7 for Titan