Lab

Julian Philip S. Quijano

2025-12-07

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.4.3
```

```
## Warning: package 'ggplot2' was built under R version 4.4.3
```

```
## Warning: package 'tibble' was built under R version 4.4.3
```

```
## Warning: package 'tidyr' was built under R version 4.4.3
```

```
## Warning: package 'readr' was built under R version 4.4.3
```

```
## Warning: package 'purrr' was built under R version 4.4.3
```

```
## Warning: package 'dplyr' was built under R version 4.4.3
```

```
## Warning: package 'forcats' was built under R version 4.4.3
```

```
## Warning: package 'lubridate' was built under R version 4.4.3
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.1     v stringr   1.5.1
## v ggplot2   4.0.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(car)
```

```
## Warning: package 'car' was built under R version 4.4.3
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 4.4.3
```

```
##
## Attaching package: 'car'
##
## The following object is masked from 'package:dplyr':
##
##     recode
##
## The following object is masked from 'package:purrr':
##
##     some
```

```
library(rstatix)
```

```
## Warning: package 'rstatix' was built under R version 4.4.3
```

```
##
## Attaching package: 'rstatix'
##
## The following object is masked from 'package:stats':
##
##     filter
```

```
library(ggpubr)
```

```
## Warning: package 'ggpubr' was built under R version 4.4.3
```

```
df = read.csv("Alzheimers Mice Data.csv")
df$AD_Status <- as.factor(df$AD_Status)
df$Treatment <- as.factor(df$Treatment)
df
```

```
##    AD_Status Treatment Training Memory
## 1          1         1       12     10
## 2          1         1       15     12
## 3          1         1       13     13
## 4          1         1       12     10
## 5          1         1       14     13
## 6          1         2       15     13
## 7          1         2       17     13
## 8          1         2       16     14
## 9          1         2       17     15
## 10         1         2       14     11
## 11         1         3       13     12
## 12         1         3       14     11
## 13         1         3       18     15
## 14         1         3       15     10
## 15         1         3       16     14
## 16         1         4       14     12
## 17         1         4       13     11
## 18         1         4       12     10
## 19         1         4       14     13
## 20         1         4       15     10
```

```
## 21          2          1        17       9
## 22          2          1        16       8
## 23          2          1        17      10
## 24          2          1        14       8
## 25          2          1        13       8
## 26          2          2        14       7
## 27          2          2        18      10
## 28          2          2        16       5
## 29          2          2        17       9
## 30          2          2        14       7
## 31          2          3        13       8
## 32          2          3        14       7
## 33          2          3        18       9
## 34          2          3        15       8
## 35          2          3        16       9
## 36          2          4        14       7
## 37          2          4        13       9
## 38          2          4        12       5
## 39          2          4        14       8
## 40          2          4        15       4
```

Assumption #1: You have one dependent variable that is measured at the continuous level.

Remark: The dependent variable is the Maze Errors (for Training Day and Memory Day) which can be measured continuously

Assumption #2: You have two independent variable that consists of three or more categorical, independent groups.

Remark: AD_Status and Treatment

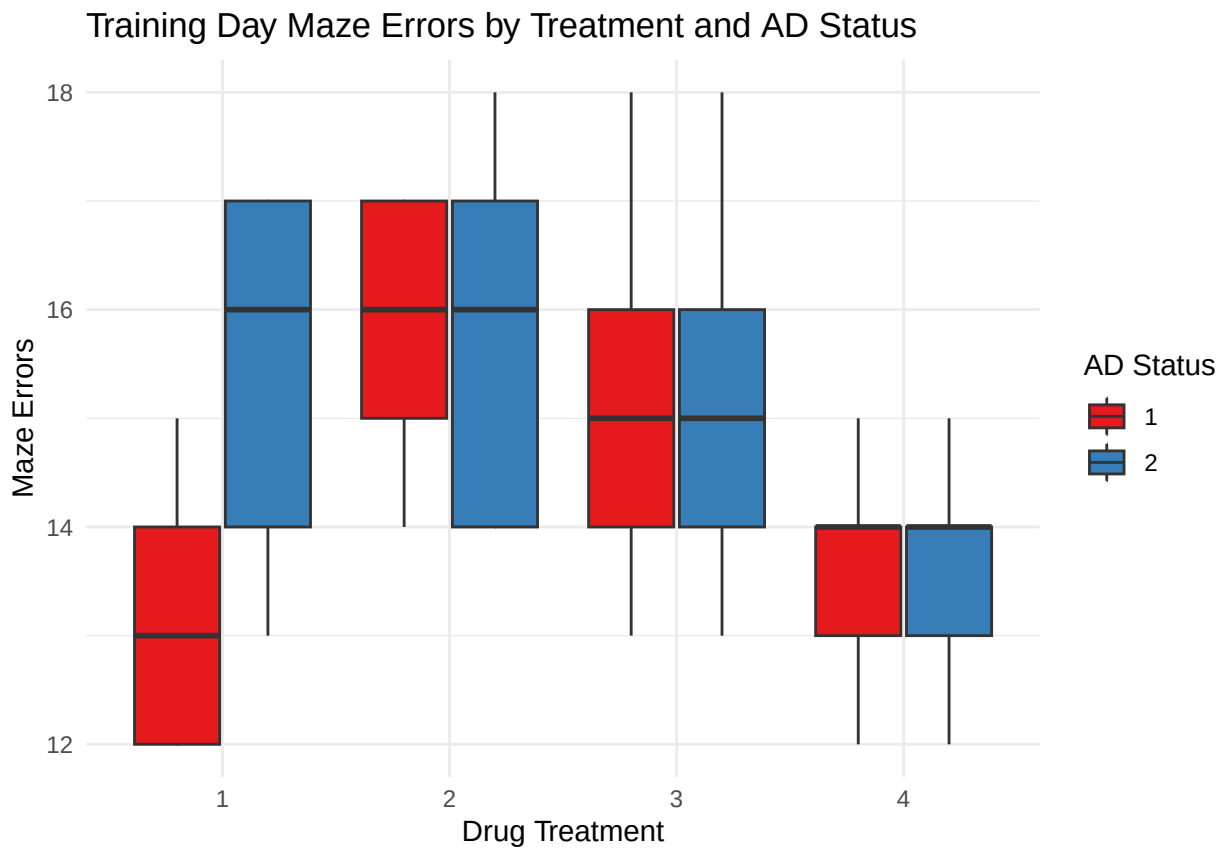
Assumption #3: You should have independence of observations, which means that there is no relationship between the observations in each group of the independent variable or among the groups themselves.

Remark: Mouse performance is observed independently.

```
#Training Day
df %>%
  group_by(AD_Status, Treatment) %>%
  identify_outliers(Training)
```

```
## [1] AD_Status  Treatment  Training   Memory     is.outlier is.extreme
## <0 rows> (or 0-length row.names)
```

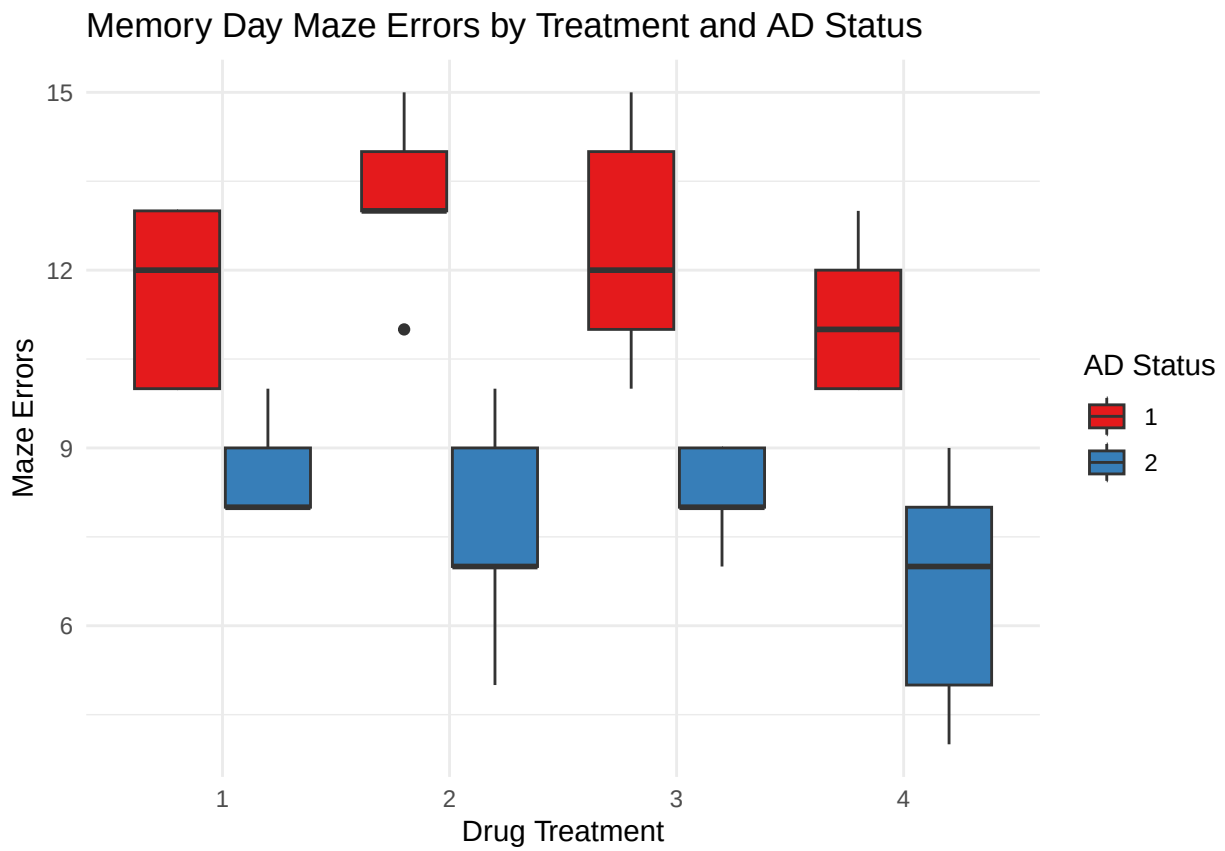
```
ggplot(df, aes(x=Treatment, y=Training, fill=AD_Status)) +
  geom_boxplot(position=position_dodge(width=0.8)) +
  labs(title="Training Day Maze Errors by Treatment and AD Status",
       x="Drug Treatment",
       y="Maze Errors") +
  scale_fill_brewer(palette="Set1", name="AD Status") +
  theme_minimal()
```




```
#Memory Day
df %>%
  group_by(AD_Status, Treatment) %>%
  identify_outliers(Memory)
```

```
## # A tibble: 1 x 6
##   AD_Status Treatment Training Memory is.outlier is.extreme
##   <fct>     <fct>        <int>  <int> <lgl>      <lgl>
## 1 1         2               14     11 TRUE       FALSE
```

```
ggplot(df, aes(x=Treatment, y=Memory, fill=AD_Status)) +
  geom_boxplot(position=position_dodge(width=0.8)) +
  labs(title="Memory Day Maze Errors by Treatment and AD Status",
       x="Drug Treatment",
       y="Maze Errors") +
  scale_fill_brewer(palette="Set1", name="AD Status") +
  theme_minimal()
```



Assumption #4: There should be no significant outliers in the three or more groups of your independent variable in terms of the dependent variable.

Remark: The results show that for Training Day, there were no significant outliers, and for Memory Day there was one outlier, but not significant.

```
# Training Day
df %>%
  group_by(AD_Status, Treatment) %>%
  shapiro_test(Training)
```

```
## # A tibble: 8 x 5
##   AD_Status Treatment variable statistic     p
##   <fct>     <fct>     <chr>        <dbl> <dbl>
## 1 1         1         Training     0.902 0.421
## 2 1         2         Training     0.902 0.421
## 3 1         3         Training     0.979 0.928
## 4 1         4         Training     0.961 0.814
## 5 2         1         Training     0.867 0.254
## 6 2         2         Training     0.894 0.377
## 7 2         3         Training     0.979 0.928
## 8 2         4         Training     0.961 0.814
```

```
# Memory Day
df %>%
  group_by(AD_Status, Treatment) %>%
  shapiro_test(Memory)
```

```
## # A tibble: 8 x 5
##   AD_Status Treatment variable statistic      p
##   <fct>     <fct>     <chr>        <dbl>  <dbl>
## 1 1         1         Memory       0.803 0.0857
## 2 1         2         Memory       0.956 0.777
## 3 1         3         Memory       0.952 0.754
## 4 1         4         Memory       0.902 0.421
## 5 2         1         Memory       0.771 0.0460
## 6 2         2         Memory       0.953 0.758
## 7 2         3         Memory       0.881 0.314
## 8 2         4         Memory       0.952 0.754
```

Assumption #5: Your dependent variable should be approximately normally distributed for each group of the independent variable.

Remark: Given the P-values for all types are $p > 0.05$, then the scores are approximately normally distributed for all groups.

```
leveneTest(Training ~ AD_Status*Treatment, data=df)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  7  0.4346 0.8731
##       32
```

```
leveneTest(Memory ~ AD_Status*Treatment, data=df)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  7  0.8275 0.5722
##       32
```

Assumption #6. You have homogeneity of variances (i.e., the variance of the dependent variable is equal in each group of your independent variable).

Remark: Levene's Test for Training Day and Memory Day shows homogeneity with p values > 0.05

```
# Training Day
anova_training = anova_test(
  data = df,
  dv = Training,
  between = c(AD_Status, Treatment),
  effect.size = "pes"
)
get_anova_table(anova_training)
```

```
## ANOVA Table (type II tests)
##
##                Effect DFn DFd     F     p p<.05   pes
## 1           AD_Status   1  32 1.216 0.278       0.037
## 2           Treatment   3  32 3.789 0.020     * 0.262
## 3 AD_Status:Treatment   3  32 1.216 0.320       0.102
```

```r
# Memory Day
anova_memory = anova_test(
  data = df,
  dv = Memory,
  between = c(AD_Status, Treatment),
  effect.size = "pes"
)
cat('\n\n')
```

```r
get_anova_table(anova_memory)
```

```
## ANOVA Table (type II tests)
##
##                Effect DFn DFd      F        p p<.05   pes
## 1           AD_Status   1  32 75.313 6.45e-10     * 0.702
## 2           Treatment   3  32  1.920 1.46e-01       0.153
## 3 AD_Status:Treatment   3  32  1.151 3.44e-01       0.097
```

```r
# Training Day
df %>%
  group_by(AD_Status) %>%
  pairwise_t_test(
    Training ~ Treatment,
    p.adjust.method = "holm"
  )
```

```
## # A tibble: 12 x 10
##    AD_Status .y.   group1 group2    n1    n2      p p.signif  p.adj p.adj.signif
##  * <fct>     <chr> <chr>  <chr>  <int> <int>  <dbl> <chr>     <dbl> <chr>
##  1 1         Trai~ 1      2          5     5 0.0119 *        0.0714 ns
##  2 1         Trai~ 1      3          5     5 0.0444 *        0.177  ns
##  3 1         Trai~ 2      3          5     5 0.522  ns       1      ns
##  4 1         Trai~ 1      4          5     5 0.668  ns       1      ns
##  5 1         Trai~ 2      4          5     5 0.0289 *        0.144  ns
##  6 1         Trai~ 3      4          5     5 0.1    ns       0.3    ns
##  7 2         Trai~ 1      2          5     5 0.714  ns       1      ns
##  8 2         Trai~ 1      3          5     5 0.854  ns       1      ns
##  9 2         Trai~ 2      3          5     5 0.584  ns       1      ns
## 10 2         Trai~ 1      4          5     5 0.113  ns       0.563  ns
## 11 2         Trai~ 2      4          5     5 0.057  ns       0.342  ns
## 12 2         Trai~ 3      4          5     5 0.155  ns       0.621  ns
```

```r
# Memory Day
df %>%
  group_by(AD_Status) %>%
  pairwise_t_test(
    Memory ~ Treatment,
    p.adjust.method = "holm"
  )
```

```
## # A tibble: 12 x 10
##    AD_Status .y.    group1 group2    n1    n2      p p.signif p.adj p.adj.signif
```

```
##  * <fct>     <chr>  <chr>  <chr>  <int> <int>  <dbl> <chr>    <dbl> <chr>
##  1 1         Memory 1      2          5     5 0.138  ns       0.69  ns
##  2 1         Memory 1      3          5     5 0.446  ns       1     ns
##  3 1         Memory 2      3          5     5 0.446  ns       1     ns
##  4 1         Memory 1      4          5     5 0.701  ns       1     ns
##  5 1         Memory 2      4          5     5 0.0687 ns       0.412 ns
##  6 1         Memory 3      4          5     5 0.259  ns       1     ns
##  7 2         Memory 1      2          5     5 0.323  ns       1     ns
##  8 2         Memory 1      3          5     5 0.689  ns       1     ns
##  9 2         Memory 2      3          5     5 0.549  ns       1     ns
## 10 2         Memory 1      4          5     5 0.0581 ns       0.348 ns
## 11 2         Memory 2      4          5     5 0.323  ns       1     ns
## 12 2         Memory 3      4          5     5 0.122  ns       0.61  ns
```

Compare the training day errors based on drug treatments and AD status.

There was no significant main effect of AD status on Training Day errors, $F(1, 32) = 1.216$, $p = .278$, $eta^2 = .037$. This indicates that, on average, transgenic and wild-type mice did not differ significantly in errors during training.

There was a significant main effect of Drug Treatment, $F(3, 32) = 3.789$, $p = .020$, $eta^2 = .262$, indicating that the number of errors differed depending on which drug the mice received.

The interaction between AD status and Drug Treatment was not significant, $F(3, 32) = 1.216$, $p = .320$, $eta^2 = .102$, suggesting that the effect of the drug was similar for transgenic and wild-type mice.

Some drug pairs showed significant differences within AD groups, e.g., Drug 1 vs Drug 2 in AD group 1, $p = .0119$.

These differences suggest that certain drugs improved performance during training for some groups, but AD status did not significantly modify the effect.

Finally, Training performance was mostly influenced by the drug administered, not by whether the mice were transgenic or wild-type. No interaction effect means that drug effects were consistent across AD

Compare the memory day errors based on drug treatments and AD status.

There was a significant main effect of AD status, $F(1, 32) = 75.313$, $p < .001$, $eta^2 = .702$. Transgenic mice made more errors than wild-type mice, showing poorer memory performance.

There was no significant main effect of Drug Treatment, $F(3, 32) = 1.920$, $p = .146$, $eta^2 = .153$, suggesting that overall, the type of drug did not significantly affect memory errors.

The interaction between AD status and Drug Treatment was not significant, $F(3, 32) = 1.151$, $p = .344$, $eta^2 = .097$, indicating that the effect of drug treatment was similar across AD statuses.

No significant pairwise differences between treatments were observed after Holm correction, consistent with the non-significant main effect of Drug Treatment.

Finally, Memory performance was primarily influenced by AD status, with transgenic mice performing worse. Drug treatment did not significantly improve memory, and the effect of drugs did not differ between transgenic and wild-type mice.

Github Link: https://github.com/SylTana/APM1111-QUIJANO-JULIAN_PHILIP-Statistical-Theory-/tree/main/SA2