Experimental Design

Teodor Chakarov

2022-05-29

Contents

Introduction to experimantal design

```r
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(ggplot2)
library(tidyr)

# Load the ToothGrowth dataset
data("ToothGrowth")

# Perform a two-sided t-test
t.test(x = ToothGrowth$len, alternative = "two.sided", mu = 18)
```

```
##
##  One Sample t-test
##
## data:  ToothGrowth$len
## t = 0.82361, df = 59, p-value = 0.4135
## alternative hypothesis: true mean is not equal to 18
## 95 percent confidence interval:
##  16.83731 20.78936
## sample estimates:
## mean of x
##  18.81333
```

Randomization

```
# Perform a t-test
ToothGrowth_ttest <- t.test(len ~ supp, data = ToothGrowth)

# Load broom
library(broom)

# Tidy ToothGrowth_ttest
tidy(ToothGrowth_ttest)
```

```
## # A tibble: 1 x 10
##   estimate estimate1 estimate2 statistic p.value parameter conf.low conf.high
##      <dbl>     <dbl>     <dbl>     <dbl>   <dbl>     <dbl>    <dbl>     <dbl>
## 1      3.7      20.7      17.0      1.92  0.0606      55.3   -0.171      7.57
## # ... with 2 more variables: method <chr>, alternative <chr>
```

Replication

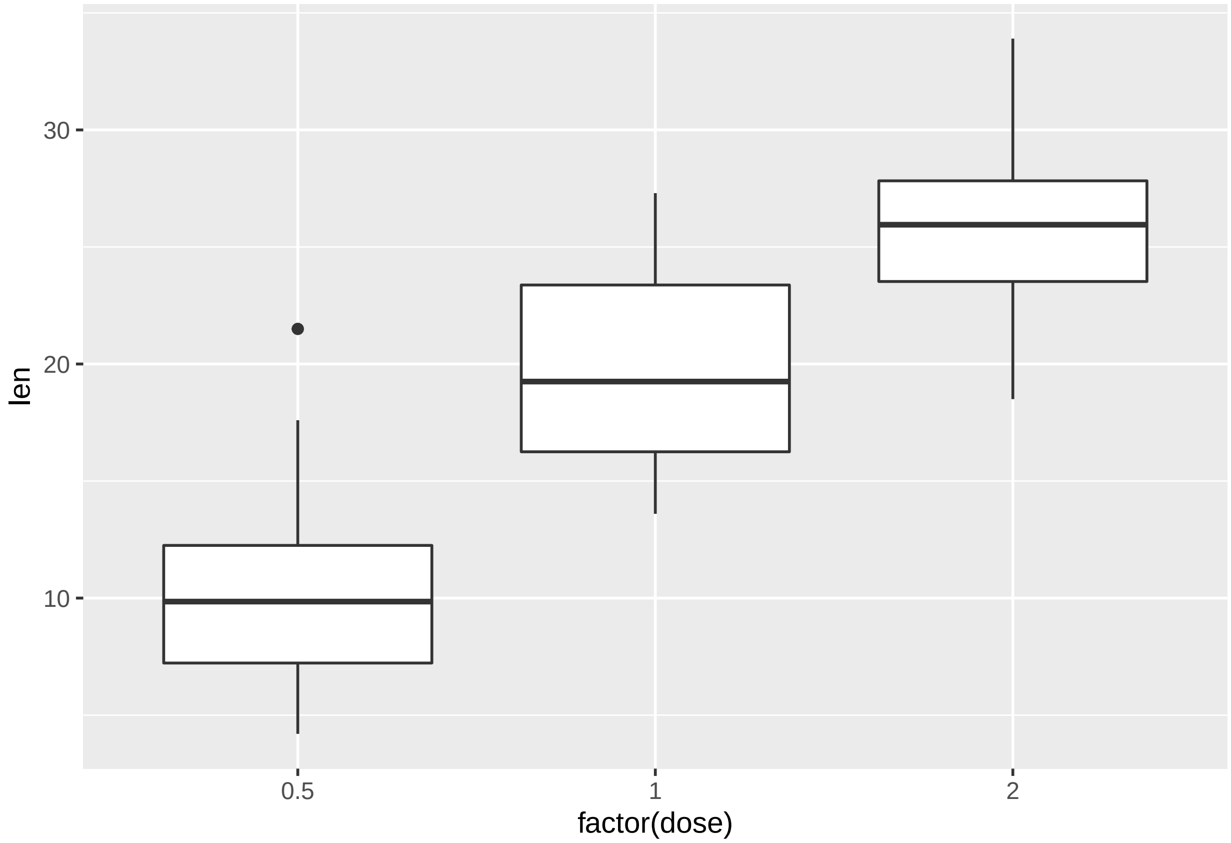
```
data(mtcars)

mtcars %>%
    count(cyl)
```

```
##   cyl  n
## 1   4 11
## 2   6  7
## 3   8 14
```

```
ToothGrowth %>%
    count(supp, dose)
```

```
##   supp dose  n
## 1   OJ  0.5 10
## 2   OJ  1.0 10
## 3   OJ  2.0 10
## 4   VC  0.5 10
## 5   VC  1.0 10
## 6   VC  2.0 10
```

```
# Create a boxplot with geom_boxplot()
ggplot(ToothGrowth, aes(x = factor(dose), y = len)) +
    geom_boxplot()
```



```
# Create ToothGrowth_aov
ToothGrowth_aov <- aov(len ~ factor(dose) + supp, data = ToothGrowth)

# Examine ToothGrowth_aov with summary()
summary(ToothGrowth_aov)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)    
## factor(dose)  2 2426.4  1213.2   82.81  < 2e-16 ***
## supp          1  205.4   205.4   14.02 0.000429 ***
## Residuals    56  820.4    14.7                     
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Hypothesis testing

```
# Less than
t.test(x = ToothGrowth$len,
       alternative = "less",
       mu = 18)
```

```
##
##  One Sample t-test
##
## data:  ToothGrowth$len
## t = 0.82361, df = 59, p-value = 0.7933
## alternative hypothesis: true mean is less than 18
## 95 percent confidence interval:
##      -Inf 20.46358
## sample estimates:
## mean of x
##  18.81333
```

```
# Greater than
t.test(x = ToothGrowth$len,
       alternative = "greater",
       mu = 18)
```

```
##
##  One Sample t-test
##
## data:  ToothGrowth$len
## t = 0.82361, df = 59, p-value = 0.2067
## alternative hypothesis: true mean is greater than 18
## 95 percent confidence interval:
##  17.16309      Inf
## sample estimates:
## mean of x
##  18.81333
```

```
library(pwr)

# Calculate power
pwr.t.test(n = 100,
           d = 0.35,
           sig.level = 0.10,
           type = "two.sample",
           alternative = "two.sided",
           power = NULL)
```

```
##
##      Two-sample t test power calculation
##
##               n = 100
##               d = 0.35
##       sig.level = 0.1
##           power = 0.7943532
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

```
# Calculate sample size
pwr.t.test(n = NULL,
           d = 0.25,
           sig.level = 0.05,
           type = "one.sample", alternative = "greater",
           power = 0.8)
```

```
##
##      One-sample t test power calculation
##
##               n = 100.2877
##               d = 0.25
##       sig.level = 0.05
##           power = 0.8
##     alternative = greater
```

Basic Experiments

Anova, factor experiments

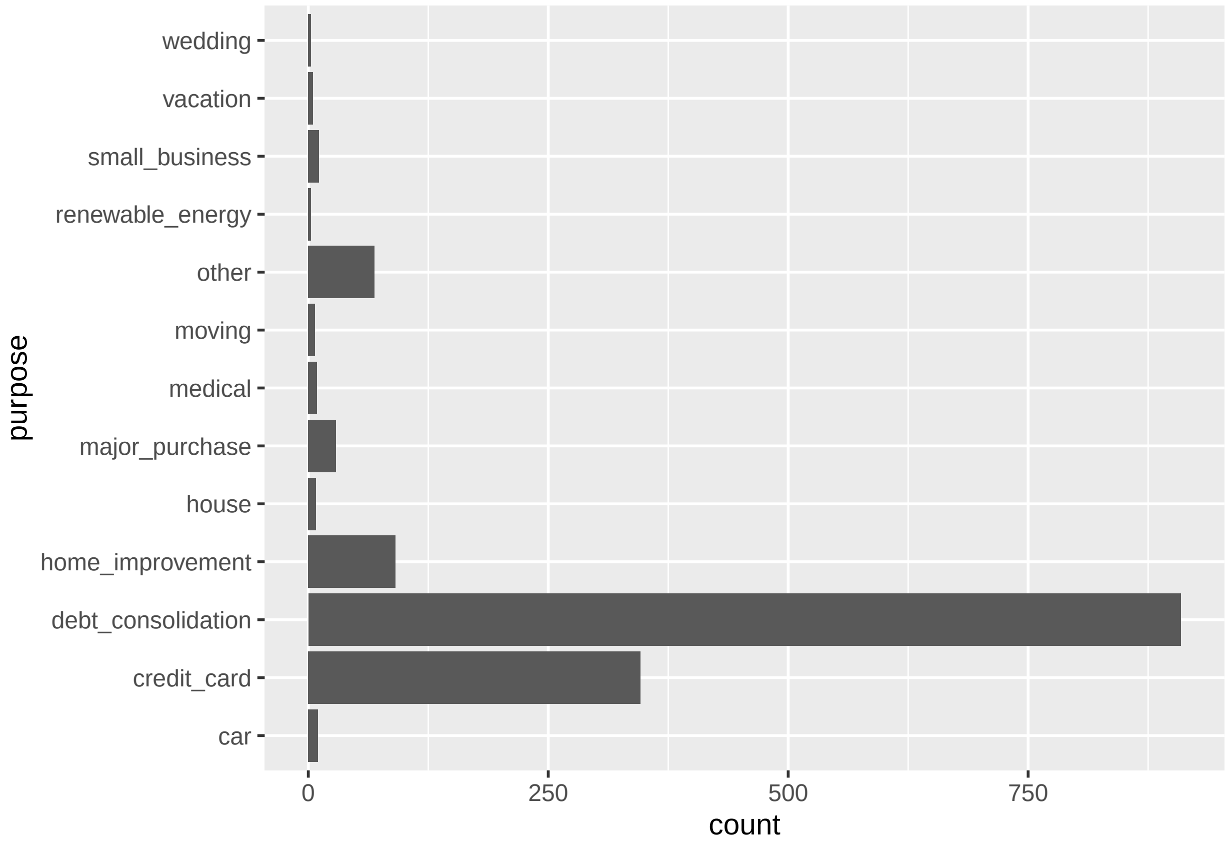
```
lendingclub <- read.csv("https://assets.datacamp.com/production/repositories/1793/datasets/e14dbe91a084

glimpse(lendingclub)
```

```
## Rows: 1,500
## Columns: 12
## $ member_id           <int> 55096114, 1555332, 1009151, 69524202, 72128084, 53~
## $ loan_amnt           <int> 11000, 10000, 13000, 5000, 18000, 14000, 8000, 500~
## $ funded_amnt         <int> 11000, 10000, 13000, 5000, 18000, 14000, 8000, 500~
## $ term                <fct> 36 months, 36 months, 60 months, 36 months, 36 mon~
## $ int_rate            <dbl> 12.69, 6.62, 10.99, 12.05, 5.32, 16.99, 13.11, 7.8~
## $ emp_length          <fct> 10+ years, 10+ years, 3 years, 10+ years, 10+ year~
## $ home_ownership      <fct> RENT, MORTGAGE, MORTGAGE, MORTGAGE, MORTGAGE, MORT~
## $ annual_inc          <dbl> 51000, 40000, 78204, 51000, 96000, 47000, 40000, 3~
## $ verification_status <fct> Not Verified, Verified, Not Verified, Not Verified~
## $ loan_status         <fct> Current, Fully Paid, Fully Paid, Current, Current,~
## $ purpose             <fct> debt_consolidation, debt_consolidation, home_impro~
## $ grade               <fct> C, A, B, C, A, D, C, A, D, B, C, B, E, C, A, C, D,~
```

```
lendingclub %>% summarize(median(loan_amnt),
                          mean(int_rate),
                          mean(annual_inc))
```

```
##   median(loan_amnt) mean(int_rate) mean(annual_inc)
## 1             13000       13.31472         75736.03
```

```
# Use ggplot2 to build a bar chart of purpose
ggplot(lendingclub, aes(purpose)) +
    geom_bar() +
    coord_flip()
```



```
# Use recode() to create the new purpose_recode variable
lendingclub$purpose_recode <- lendingclub$purpose %>% recode(
        "credit_card" = "debt_related",
        "debt_consolidation" = "debt_related",
        "medical" = "debt_related",
        "car" = "big_purchase",
        "major_purchase" = "big_purchase",
        "vacation" = "big_purchase",
        "moving" = "life_change",
        "small_business" = "life_change",
        "wedding" = "life_change",
        "house" = "home_related",
        "home_improvement" = "home_related")
```

```
# Build a linear regression model, purpose_recode_model
purpose_recode_model <- lm(funded_amnt ~ purpose_recode, data = lendingclub)

# Examine results of purpose_recode_model
summary(purpose_recode_model)
```

```
##
## Call:
## lm(formula = funded_amnt ~ purpose_recode, data = lendingclub)
##
## Residuals:
```

```
##    Min     1Q Median     3Q    Max
## -14472  -6251  -1322   4678  25761
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      9888.1     1248.9   7.917 4.69e-15 ***
## purpose_recodedebt_related       5433.5     1270.5   4.277 2.02e-05 ***
## purpose_recodehome_related       4845.0     1501.0   3.228  0.00127 **
## purpose_recodelife_change        4095.3     2197.2   1.864  0.06254 .
## purpose_recodeother              -649.3     1598.3  -0.406  0.68461
## purpose_recoderenewable_energy  -1796.4     4943.3  -0.363  0.71636
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8284 on 1494 degrees of freedom
## Multiple R-squared:  0.03473,    Adjusted R-squared:  0.0315
## F-statistic: 10.75 on 5 and 1494 DF,  p-value: 3.598e-10
```

```
# Get anova results and save as purpose_recode_anova
purpose_recode_anova <- anova(purpose_recode_model)

# Print purpose_recode_anova
purpose_recode_anova
```

```
## Analysis of Variance Table
##
## Response: funded_amnt
##                  Df     Sum Sq   Mean Sq F value    Pr(>F)
## purpose_recode    5 3.6888e+09 737756668   10.75 3.598e-10 ***
## Residuals      1494 1.0253e+11  68629950
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Examine class of purpose_recode_anova
class(purpose_recode_anova)
```

```
## [1] "anova"      "data.frame"
```

```
# Use aov() to build purpose_aov
purpose_aov <- aov(funded_amnt ~ purpose_recode, data = lendingclub)

# Conduct Tukey's HSD test to create tukey_output
tukey_output <- TukeyHSD(purpose_aov, "purpose_recode", conf.level = 0.95)

# Tidy tukey_output to make sense of the results
tidy(tukey_output)
```

```
## # A tibble: 15 x 7
##    term           contrast    null.value estimate conf.low conf.high adj.p.value
##    <chr>          <chr>            <dbl>    <dbl>    <dbl>     <dbl>       <dbl>
##  1 purpose_recode debt_relat~          0    5434.    1808.     9059.     2.91e-4
##  2 purpose_recode home_relat~          0    4845.     562.     9128.     1.61e-2
##  3 purpose_recode life_chang~          0    4095.   -2174.    10365.     4.25e-1
##  4 purpose_recode other-big_~          0    -649.   -5210.     3911.     9.99e-1
##  5 purpose_recode renewable_~          0   -1796.  -15902.    12309.     9.99e-1
##  6 purpose_recode home_relat~          0    -589.   -3056.     1879.     9.84e-1
##  7 purpose_recode life_chang~          0   -1338.   -6539.     3863.     9.78e-1
##  8 purpose_recode other-debt~          0   -6083.   -9005.    -3160.     5.32e-8
##  9 purpose_recode renewable_~          0   -7230.  -20894.     6434.     6.58e-1
## 10 purpose_recode life_chang~          0    -750.   -6429.     4929.     9.99e-1
## 11 purpose_recode other-home~          0   -5494.   -9201.    -1787.     3.58e-4
## 12 purpose_recode renewable_~          0   -6641.  -20494.     7212.     7.46e-1
## 13 purpose_recode other-life~          0   -4745.  -10636.     1147.     1.95e-1
## 14 purpose_recode renewable_~          0   -5892.  -20482.     8698.     8.59e-1
## 15 purpose_recode renewable_~          0   -1147.  -15088.    12794.     1.00e+0
```

```
# Use aov() to build purpose_emp_aov
purpose_emp_aov <- aov(funded_amnt ~ purpose_recode + emp_length, data = lendingclub)

# Print purpose_emp_aov to the console
purpose_emp_aov
```

```
## Call:
##    aov(formula = funded_amnt ~ purpose_recode + emp_length, data = lendingclub)
##
## Terms:
##                    purpose_recode   emp_length    Residuals
## Sum of Squares         3688783338   2044273211 100488872355
## Deg. of Freedom                 5           11         1483
##
## Residual standard error: 8231.679
## Estimated effects may be unbalanced
```

```
# Call summary() to see the p-values
summary(purpose_emp_aov)
```

```
##                  Df    Sum Sq   Mean Sq F value   Pr(>F)
## purpose_recode    5 3.689e+09 737756668  10.888 2.63e-10 ***
## emp_length       11 2.044e+09 185843019   2.743  0.00161 **
## Residuals      1483 1.005e+11  67760534
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Pre-modeling EDA

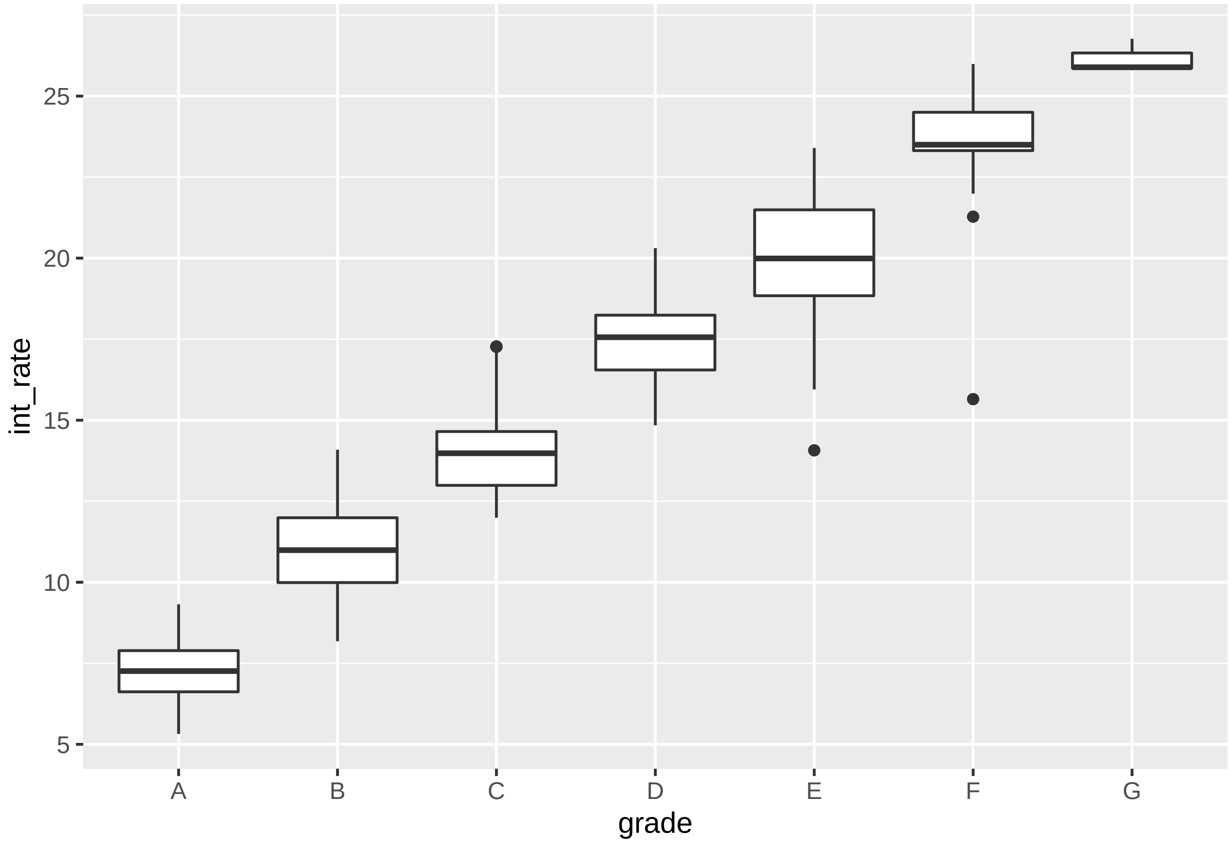
```
summary(lendingclub$int_rate)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5.32    9.99   12.99   13.31   16.29   26.77
```

```
# Examine int_rate by grade
lendingclub %>%
    group_by(grade) %>%
    summarize(mean = mean(int_rate), var = var(int_rate), median = median(int_rate))
```

```
## # A tibble: 7 x 4
##   grade  mean   var median
##   <fct> <dbl> <dbl>  <dbl>
## 1 A      7.27 0.961   7.26
## 2 B     10.9  2.08   11.0
## 3 C     14.0  1.42   14.0
## 4 D     17.4  1.62   17.6
## 5 E     20.1  2.71   20.0
## 6 F     23.6  2.87   23.5
## 7 G     26.1  0.198  25.9
```

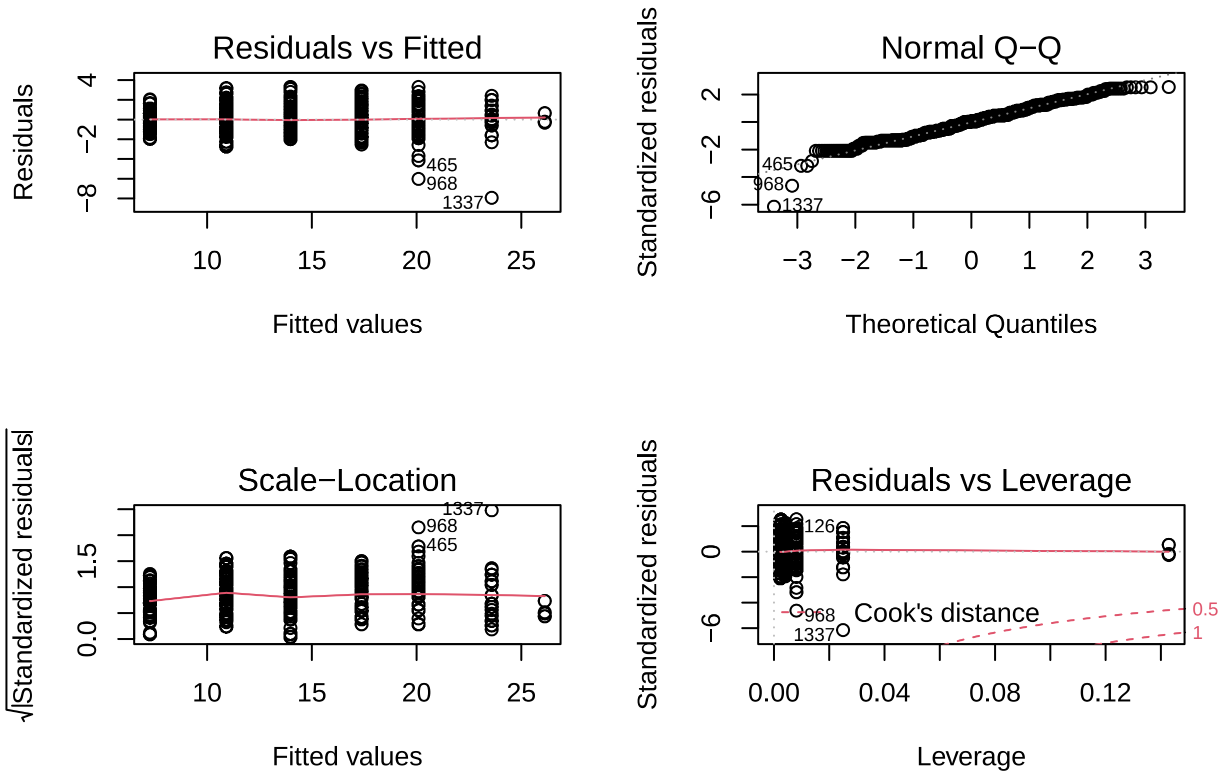
```
# Make a boxplot of int_rate by grade
ggplot(lendingclub, aes(x = grade, y = int_rate)) +
    geom_boxplot()
```



```
# Use aov() to create grade_aov plus call summary() to print results
grade_aov <- aov(int_rate ~ grade, data = lendingclub)
summary(grade_aov)
```

```
##                  Df Sum Sq Mean Sq F value Pr(>F)
## grade          6  27013    4502    2637 <2e-16 ***
## Residuals   1493   2549       2
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# For a 2x2 grid of plots:
par(mfrow = c(2, 2))

# Plot grade_aov
plot(grade_aov)
```



```
# Bartlett's test for homogeneity of variance
bartlett.test(int_rate ~ grade, data = lendingclub)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  int_rate by grade
## Bartlett's K-squared = 78.549, df = 6, p-value = 7.121e-15
```

```
# Conduct the Kruskal-Wallis rank sum test
kruskal.test(int_rate ~ grade,
             data = lendingclub)
```

```
##
```

```
##  Kruskal-Wallis rank sum test
##
## data:  int_rate by grade
## Kruskal-Wallis chi-squared = 1365.5, df = 6, p-value < 2.2e-16
```

A/B Test

```
# Load the pwr package
library(pwr)

# Use the correct function from pwr to find the sample size
pwr.t.test(n = NULL,
           d = 0.2,
           sig.level = 0.05,
           power = 0.8,
           alternative = "two.sided")
```

```
##
##      Two-sample t test power calculation
##
##               n = 393.4057
##               d = 0.2
##       sig.level = 0.05
##           power = 0.8
##     alternative = two.sided
##
## NOTE: n is number in *each* group
```

```
ggplot(lendingclub_ab, aes(x = Group, y = loan_amnt)) +
    geom_boxplot()
```

```
## Error in ggplot(lendingclub_ab, aes(x = Group, y = loan_amnt)): object 'lendingclub_ab' not found
```

```
t.test(loan_amnt ~ Group, data = lendingclub_ab)
```

```
## Error in eval(m$data, parent.frame()): object 'lendingclub_ab' not found
```

```
# Build lendingclub_multi
lendingclub_multi <- lm(loan_amnt ~ Group + grade + verification_status, data = lendingclub_ab)
```

```
## Error in is.data.frame(data): object 'lendingclub_ab' not found
```

```
# Examine lendingclub_multi results
tidy(lendingclub_multi)
```

```
## Error in tidy(lendingclub_multi): object 'lendingclub_multi' not found
```

Intro to NHANES and sampling

```
# Load haven
library(haven)

nhanes_demo <- read.csv("https://assets.datacamp.com/production/repositories/1793/datasets/2be5ca94453a
nhanes_medical <- read.csv("https://assets.datacamp.com/production/repositories/1793/datasets/d34921a925
nhanes_bodymeasures <- read.csv("https://assets.datacamp.com/production/repositories/1793/datasets/ee832
```

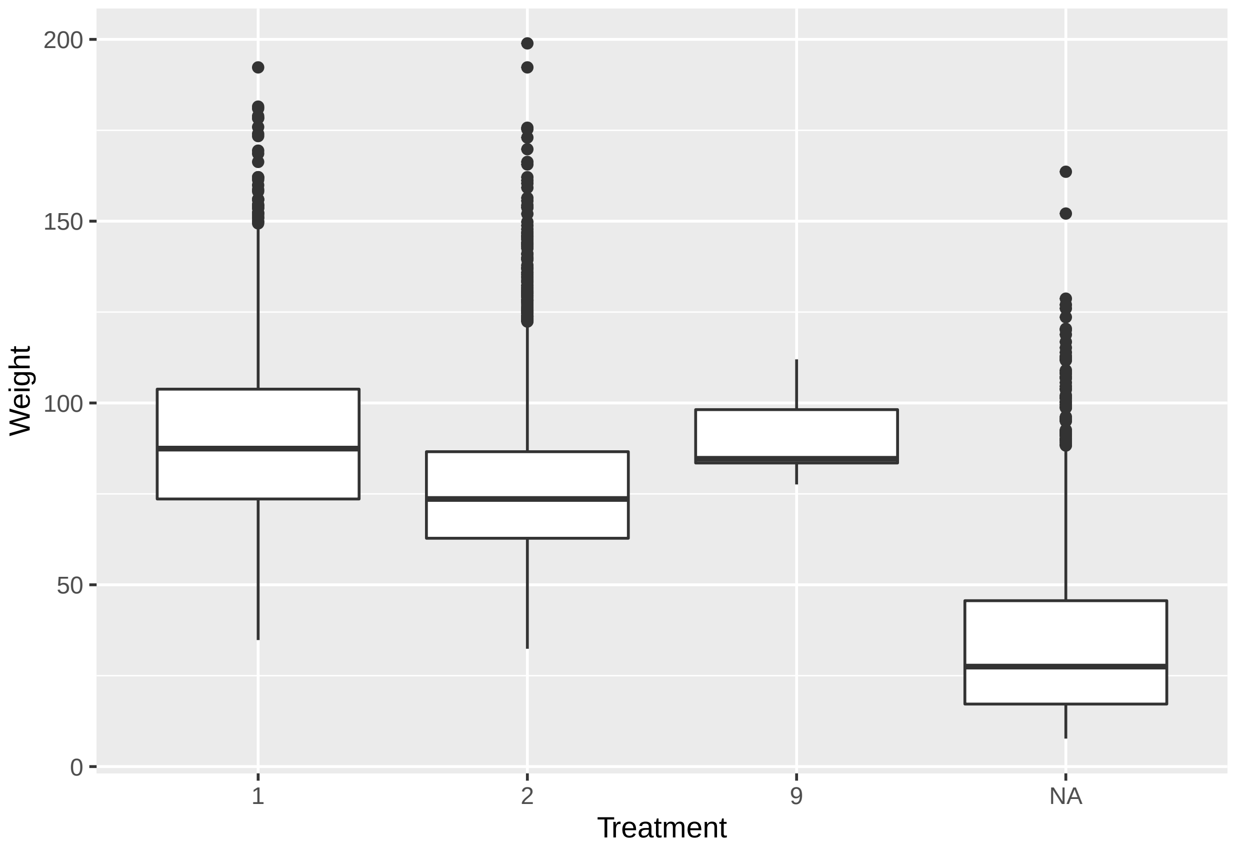
```
# Merge the 3 datasets you just created to create nhanes_combined
nhanes_combined <- list(nhanes_demo, nhanes_medical, nhanes_bodymeasures) %>%
  Reduce(function(df1, df2) inner_join(df1, df2, by = "seqn"), .)
```

```
# Fill in the dplyr code
nhanes_combined %>%
  group_by(mcq365d) %>%
  summarize(mean = mean(bmxwt, na.rm = TRUE))
```

```
## # A tibble: 4 x 2
##   mcq365d  mean
##     <int> <dbl>
## 1       1  90.7
## 2       2  76.5
## 3       9  90.8
## 4      NA  33.5
```

```
# Fill in the ggplot2 code
nhanes_combined %>%
  ggplot(aes(as.factor(mcq365d), bmxwt)) +
  geom_boxplot() +
  labs(x = "Treatment",
       y = "Weight")
```

```
## Warning: Removed 99 rows containing non-finite values (stat_boxplot).
```



```r
# Filter to keep only those 16+
nhanes_filter <- nhanes_combined %>% filter(ridageyr > 16)

# Load simputation & impute bmxwt by riagendr
library(simputation)
nhanes_final <- impute_median(nhanes_filter, bmxwt ~ riagendr)

# Recode mcq365d with recode() & examine with count()
nhanes_final$mcq365d <- recode(nhanes_final$mcq365d,
                               `1` = 1,
                               `2` = 2,
                               `9` = 2)
nhanes_final %>% count(mcq365d)
```

```
##   mcq365d    n
## 1       1 1802
## 2       2 4085
```

```r
# Use sample_n() to create nhanes_srs
nhanes_srs <- nhanes_final %>% sample_n(2500)

# Create nhanes_stratified with group_by() and sample_n()
nhanes_stratified <- nhanes_final %>% group_by(riagendr) %>% sample_n(2000)
nhanes_stratified %>%
    count(riagendr)
```

```
## # A tibble: 2 x 2
## # Groups:   riagendr [2]
##   riagendr     n
##      <int> <int>
## 1        1  2000
## 2        2  2000
```

```
# Load sampling package and create nhanes_cluster with cluster()
library(sampling)
nhanes_cluster <- cluster(nhanes_final, "indhhin2", 6, method = "srswor")
nhanes_cluster %>%
  count(indhhin2)
```

```
##   indhhin2   n
## 1        7 556
## 2        9 353
## 3       10 291
## 4       13  90
## 5       77 119
## 6       99  90
```

```
library(agricolae)

# Create designs using ls()
designs <- ls("package:agricolae", pattern = "design")
designs
```

```
##  [1] "design.ab"      "design.alpha"   "design.bib"     "design.crd"
##  [5] "design.cyclic"  "design.dau"     "design.graeco"  "design.lattice"
##  [9] "design.lsd"     "design.mat"     "design.rcbd"    "design.split"
## [13] "design.strip"   "design.youden"
```

```
str(design.rcbd)
```

```
## function (trt, r, serie = 2, seed = 0, kinds = "Super-Duper", first = TRUE,
##     continue = FALSE, randomization = TRUE)
```

```
# Build treats and rep
treats <- LETTERS[1:5]
blocks <- 4

# Build my_design_rcbd and view the sketch
my_design_rcbd <- design.rcbd(treats, r = blocks, seed = 42)
my_design_rcbd$sketch
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,] "D"  "A"  "C"  "B"  "E"
## [2,] "E"  "A"  "C"  "D"  "B"
## [3,] "D"  "B"  "E"  "A"  "C"
## [4,] "B"  "D"  "E"  "C"  "A"
```

```r
# Use aov() to create nhanes_rcbd
nhanes_rcbd <- aov(bmxwt ~ mcq365d + riagendr, data = nhanes_final)

# Check results of nhanes_rcbd with summary()
summary(nhanes_rcbd)
```

```
##                Df  Sum Sq Mean Sq F value Pr(>F)    
## mcq365d         1  229164  229164   571.2 <2e-16 ***
## riagendr        1  163069  163069   406.4 <2e-16 ***
## Residuals    5884 2360774     401                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

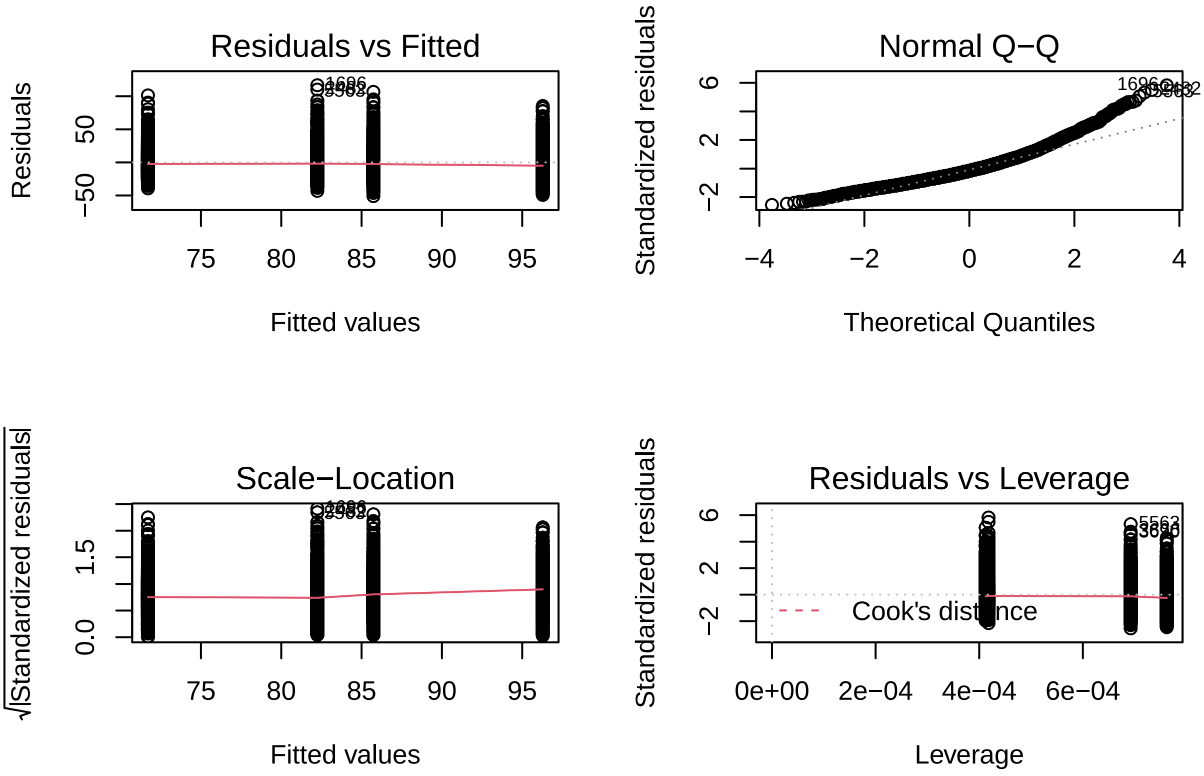
```r
# Print mean weights by mcq365d and riagendr
nhanes_final %>%
    group_by(mcq365d, riagendr) %>%
    summarize(mean_wt = mean(bmxwt, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'mcq365d'. You can override using the
## `.groups` argument.
```

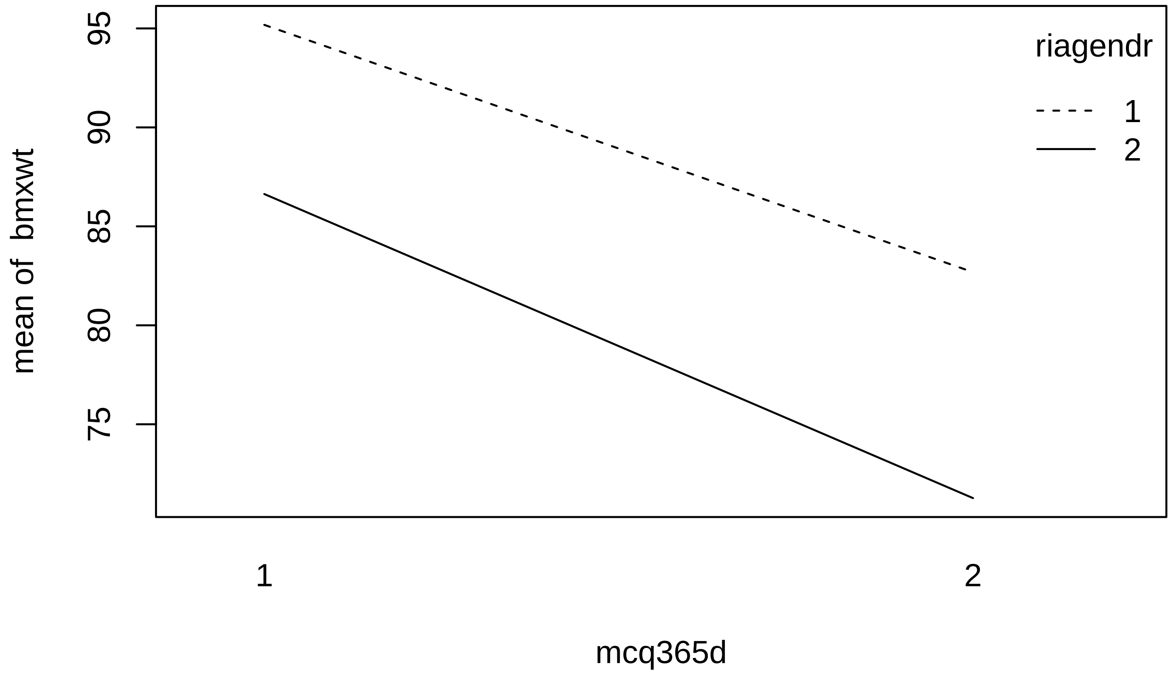
```
## # A tibble: 4 x 3
## # Groups:   mcq365d [2]
##   mcq365d riagendr mean_wt
##     <dbl>    <int>   <dbl>
## 1       1        1    95.2
## 2       1        2    86.6
## 3       2        1    82.7
## 4       2        2    71.3
```

model Validation

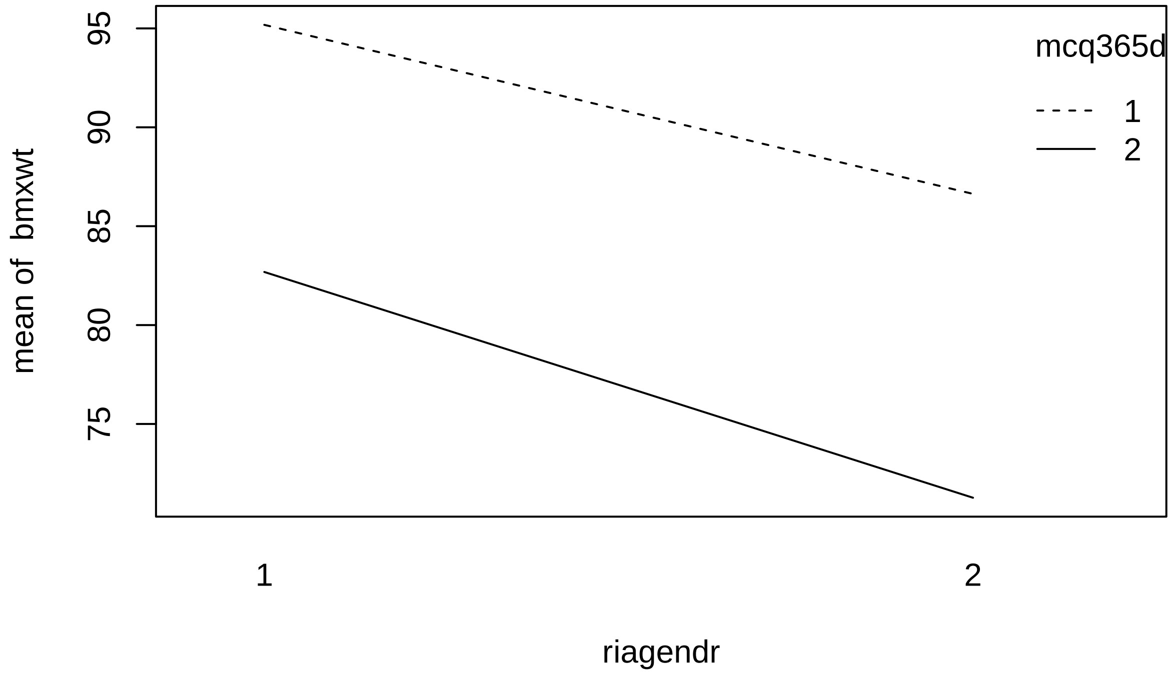
```r
# Set up the 2x2 plotting grid and plot nhanes_rcbd
par(mfrow = c(2, 2))

plot(nhanes_rcbd)
```



```
# Run the code to view the interaction plots
with(nhanes_final, interaction.plot(mcq365d, riagendr, bmxwt))
```



```
# Run the code to view the interaction plots
with(nhanes_final, interaction.plot(riagendr, mcq365d, bmxwt))
```



Balanced Incomplete Block Designs (BIBD)

```
#create my_design_bibd_1
my_design_bibd_1 <- design.bib(LETTERS[1:3], k = 4, seed = 42)
```

```
## Error in AlgDesign::optBlock(~., withinData = factor(1:v), blocksizes = rep(k, : The number of trial
```

```
#create my_design_bibd_2
my_design_bibd_2 <- design.bib(LETTERS[1:8], k = 3, seed = 42)
```

```
## Error in rep(k, b): invalid 'times' argument
```

```
# Create my_design_bibd_3
my_design_bibd_3 <- design.bib(LETTERS[1:4], k = 4, seed = 42)
```

```
##
## Parameters BIB
## ==============
## Lambda     : 2
## treatmeans : 4
## Block size : 4
## Blocks     : 2
## Replication: 2
##
## Efficiency factor 1
##
## <<< Book >>>
```

```
my_design_bibd_3$sketch
```

```
##      [,1] [,2] [,3] [,4]
## [1,] "C"  "A"  "D"  "B"
## [2,] "C"  "D"  "B"  "A"
```

```
lambda <- function(t, k, r) {
  return((r*(k-1)) / (t-1))
}

# Calculate lambda
lambda(4, 3, 3)
```

```
## [1] 2
```

```
# Build the data.frame
creatinine <- c(1.98, 1.97, 2.35, 2.09, 1.87, 1.95, 2.08, 2.01, 1.84, 2.06, 1.97, 2.22)
food <- as.factor(c("A", "C", "D", "A", "B", "C", "B", "C", "D", "A", "B", "D"))
color <- as.factor(rep(c("Black", "White", "Orange", "Spotted"), each = 3))
cat_experiment <- as.data.frame(cbind(creatinine, food, color))

# Create cat_model and examine with summary()
cat_model <- aov(creatinine ~ food + color, data = cat_experiment)
summary(cat_model)
```

```
##             Df  Sum Sq  Mean Sq F value Pr(>F)
## food         1 0.01204 0.012042   0.530  0.485
## color        1 0.00697 0.006971   0.307  0.593
## Residuals    9 0.20461 0.022735
```

```
# Calculate lambda
lambda(3, 2, 2)
```

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

```
# Create weightlift_model & examine results
weightlift_model <- aov(bmxarmc ~ weightlift_treat + ridreth1, data = nhanes_final)
```

```
## Error in eval(predvars, data, env): object 'weightlift_treat' not found
```

```
summary(weightlift_model)
```

```
## Error in summary(weightlift_model): object 'weightlift_model' not found
```

Latin Squares, Factorial experiments

```
nyc_scores <- read.csv("https://assets.datacamp.com/production/repositories/1793/datasets/6eee2fcc47c8c8

# Mean, var, and median of Math score
nyc_scores %>%
    group_by(Borough) %>%
    summarize(mean = mean(Average_Score_SAT_Math, na.rm = TRUE),
        var = var(Average_Score_SAT_Math, na.rm = TRUE),
        median = median(Average_Score_SAT_Math, na.rm = TRUE))
```

```
## # A tibble: 5 x 4
##   Borough        mean   var median
##   <fct>         <dbl> <dbl>  <dbl>
## 1 Bronx          404. 2726.   396.
## 2 Brooklyn       416. 3658.   395
## 3 Manhattan      456. 7026.   433
## 4 Queens         462. 5168.   448
## 5 Staten Island  486. 6911.   466.
```

```
nyc_scores %>%
    group_by(Teacher_Education_Level) %>%
    summarize(mean = mean(Average_Score_SAT_Math, na.rm = TRUE),
        var = var(Average_Score_SAT_Math, na.rm = TRUE),
        median = median(Average_Score_SAT_Math, na.rm = TRUE))
```

```
## Error in `group_by()`:
## ! Must group by variables found in `.data`.
## x Column `Teacher_Education_Level` is not found.
```

```
# Mean, var, and median of Math score by both
nyc_scores %>%
    group_by(Borough, Teacher_Education_Level) %>%
    summarize(mean = mean(Average_Score_SAT_Math, na.rm = TRUE),
        var = var(Average_Score_SAT_Math, na.rm = TRUE),
        median = median(Average_Score_SAT_Math, na.rm = TRUE))
```

```
## Error in `group_by()`:
## ! Must group by variables found in `.data`.
## x Column `Teacher_Education_Level` is not found.
```

Deleting Missing test scores

```
# Load naniar
library(naniar)
```

```
##
## Attaching package: 'naniar'
## The following object is masked from 'package:simputation':
##
##     impute_median
```

```
# Examine missingness with miss_var_summary()
nyc_scores %>% miss_var_summary()
```

```
## # A tibble: 22 x 3
##    variable                  n_miss pct_miss
##    <chr>                      <int>    <dbl>
##  1 Average_Score_SAT_Math        60    13.8
##  2 Average_Score_SAT_Reading     60    13.8
##  3 Average_Score_SAT_Writing     60    13.8
##  4 Percent_Tested                49    11.3
##  5 Student_Enrollment             7     1.61
##  6 Percent_White                  7     1.61
##  7 Percent_Black                  7     1.61
##  8 Percent_Hispanic               7     1.61
##  9 Percent_Asian                  7     1.61
## 10 School_ID                      0     0
## # ... with 12 more rows
```

```
# Examine missingness with md.pattern()
md.pattern(nyc_scores)
```

```
## Error in md.pattern(nyc_scores): could not find function "md.pattern"
```

```
# Impute the Math score by Borough
nyc_scores_2 <- impute_median(nyc_scores, Average_Score_SAT_Math ~ Borough)
```

```
## Error in impute_median(nyc_scores, Average_Score_SAT_Math ~ Borough): unused argument (Average_Score_
```

```
# Convert Math score to numeric
nyc_scores_2$Average_Score_SAT_Math <- as.numeric(nyc_scores_2$Average_Score_SAT_Math)
```

```
## Error in eval(expr, envir, enclos): object 'nyc_scores_2' not found
```

```
# Examine scores by Borough in both datasets, before and after imputation
nyc_scores %>%
    group_by(Borough) %>%
    summarize(median = median(Average_Score_SAT_Math, na.rm = TRUE),
              mean = mean(Average_Score_SAT_Math, na.rm = TRUE))
```

```
## # A tibble: 5 x 3
##   Borough       median  mean
##   <fct>          <dbl> <dbl>
## 1 Bronx           396.  404.
## 2 Brooklyn        395   416.
## 3 Manhattan       433   456.
## 4 Queens          448   462.
## 5 Staten Island   466.  486.
```

```r
nyc_scores_2 %>%
    group_by(Borough) %>%
    summarize(median = median(Average_Score_SAT_Math),
              mean = mean(Average_Score_SAT_Math))
```

```
## Error in group_by(., Borough): object 'nyc_scores_2' not found
```

```r
# Load agricolae
library(agricolae)

# Design a LS with 5 treatments A:E then look at the sketch
my_design_lsd <- design.lsd(trt = LETTERS[1:5], seed = 42)
my_design_lsd$sketch
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,] "E"  "D"  "A"  "C"  "B"
## [2,] "D"  "C"  "E"  "B"  "A"
## [3,] "A"  "E"  "B"  "D"  "C"
## [4,] "C"  "B"  "D"  "A"  "E"
## [5,] "B"  "A"  "C"  "E"  "D"
```

```r
# Build nyc_scores_ls_lm
nyc_scores_ls_lm <- lm(Average_Score_SAT_Math ~ Tutoring_Program + Borough + Teacher_Education_Level,
                       data = nyc_scores_ls)
```

```
## Error in is.data.frame(data): object 'nyc_scores_ls' not found
```

```r
# Tidy the results with broom
tidy(nyc_scores_ls_lm)
```

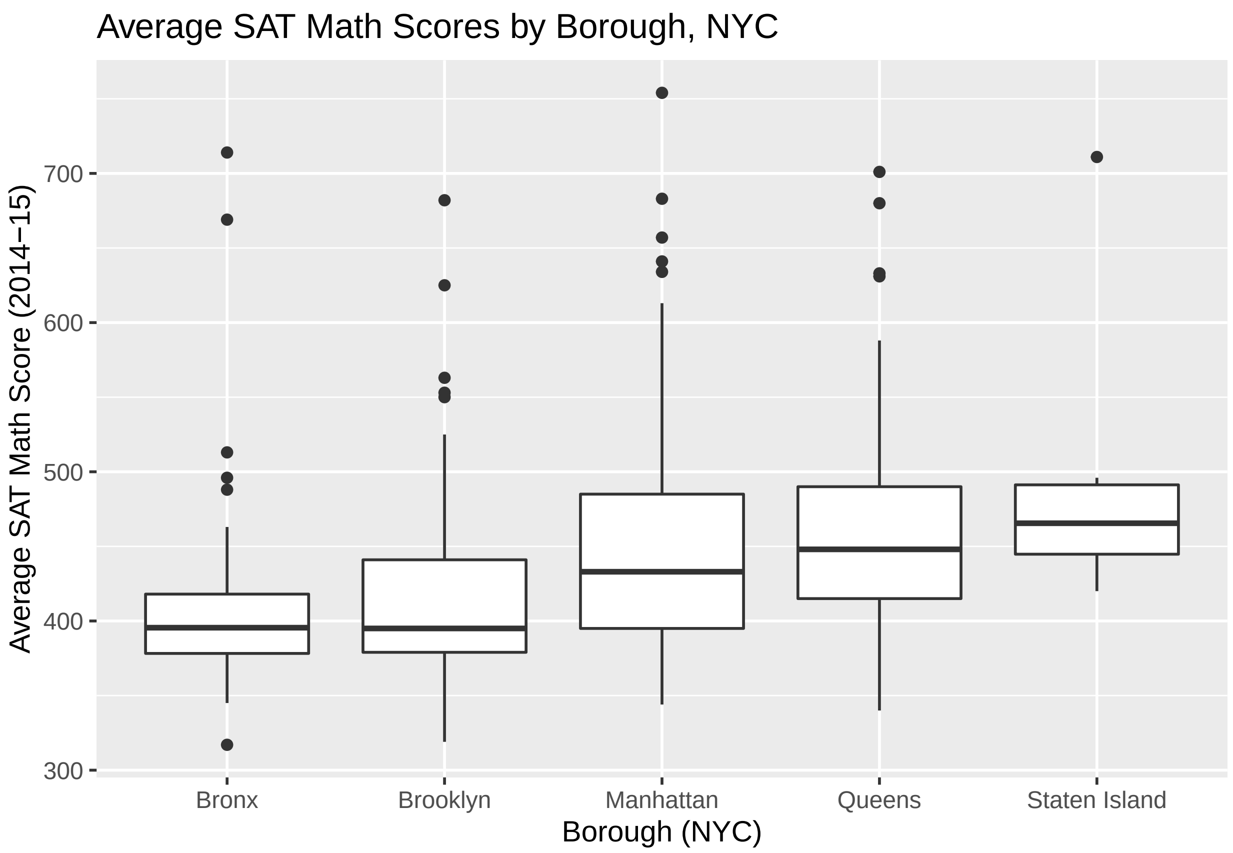
```
## Error in tidy(nyc_scores_ls_lm): object 'nyc_scores_ls_lm' not found
```

```r
# Examine the results with anova
anova(nyc_scores_ls_lm)
```

```
## Error in anova(nyc_scores_ls_lm): object 'nyc_scores_ls_lm' not found
```

```r
# Create a boxplot of Math scores by Borough, with a title and x/y axis labels
ggplot(nyc_scores, aes(Borough, Average_Score_SAT_Math)) +
  geom_boxplot() +
  labs(title = "Average SAT Math Scores by Borough, NYC",
       x = "Borough (NYC)",
       y = "Average SAT Math Score (2014-15)")
```

```
## Warning: Removed 60 rows containing non-finite values (stat_boxplot).
```



```r
# Create trt1 and trt2
trt1 <- LETTERS[1:5]
trt2 <- 1:5

# Create my_graeco_design
my_graeco_design <- design.graeco(trt1, trt2, seed = 42)

# Examine the parameters and sketch
my_graeco_design$parameters
```

```
## $design
## [1] "graeco"
##
## $trt1
## [1] "A" "B" "C" "D" "E"
##
## $trt2
## [1] 1 2 3 4 5
##
## $r
## [1] 5
##
## $serie
## [1] 2
##
```

```
## $seed
## [1] 42
##
## $kinds
## [1] "Super-Duper"
##
## [[8]]
## [1] TRUE
```

```
my_graeco_design$sketch
```

```
##      [,1]  [,2]  [,3]  [,4]  [,5]
## [1,] "D 5" "A 1" "C 3" "B 4" "E 2"
## [2,] "A 3" "C 4" "B 2" "E 5" "D 1"
## [3,] "C 2" "B 5" "E 1" "D 3" "A 4"
## [4,] "B 1" "E 3" "D 4" "A 2" "C 5"
## [5,] "E 4" "D 2" "A 5" "C 1" "B 3"
```

```
# Build nyc_scores_gls_lm
nyc_scores_gls_lm <- lm(Average_Score_SAT_Math ~ Tutoring_Program + Borough + Teacher_Education_Level +
                        data = nyc_scores_gls)
```

```
## Error in is.data.frame(data): object 'nyc_scores_gls' not found
```

```
# Tidy the results with broom
tidy(nyc_scores_gls_lm)
```

```
## Error in tidy(nyc_scores_gls_lm): object 'nyc_scores_gls_lm' not found
```

```
# Examine the results with anova
anova(nyc_scores_gls_lm)
```

```
## Error in anova(nyc_scores_gls_lm): object 'nyc_scores_gls_lm' not found
```

Factorial Experiments

```
# Load ggplot2
library(ggplot2)

# Build the boxplot for the tutoring program vs. Math SAT score
ggplot(nyc_scores,
       aes(Tutoring_Program, Average_Score_SAT_Math)) +
       geom_boxplot()
```

```
## Error in FUN(X[[i]], ...): object 'Tutoring_Program' not found
```

```
# Build the boxplot for percent black vs. Math SAT score
ggplot(nyc_scores,
       aes(Percent_Black_HL, Average_Score_SAT_Math)) +
    geom_boxplot()
```

```
## Error in FUN(X[[i]], ...): object 'Percent_Black_HL' not found
```

```
shapiro.test(nyc_scores$Average_Score_SAT_Math)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  nyc_scores$Average_Score_SAT_Math
## W = 0.84672, p-value < 2.2e-16
```