Villin-Cre vs Pck1ff TG Tolerance Test

Lehninger

2025-01-29

Here is some old, published data from my graduate school, Am J Physiol Gastrointest Liver Physiol. 2018 Apr 6 315(2) G249–G258.

The original analysis was in Excel and GraphPad Prism, circa 2014.

The experiment tested if the knockout mouse had impaired triglyceride absorption.

```
#remember to remove the "," from the cells!
#read the CSV and load a tibble
#tg_tolerance <- read_csv("G:/My Drive/CODING - - -/DataSets/AP41_wide.csv
")
tg_tolerance <- read_csv("C:/Users/pottsau/Downloads/AP41_wide.csv")
```

```
## Rows: 52 Columns: 5
## ── Column specification
────────────────────────────────────────────────────────
## Delimiter: ","
## chr (1): genotype
## dbl (4): animal_num, 0hr_tg, 2hr_tg, 4hr_tg
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this
message.
```

```
head(tg_tolerance)
```

```
## # A tibble: 6 × 5
##   animal_num `0hr_tg` `2hr_tg` `4hr_tg` genotype
##        <dbl>    <dbl>    <dbl>    <dbl> <chr>
## 1          1       73     1199     2124 Pck1ff
## 2          2       63      742     6577 Pck1ff
## 3          3       NA       NA       NA Pck1ff
## 4          4       45      955     3436 Pck1ff
## 5          5       33     1077     5876 Pck1ff
## 6          6       39     1062     3710 Pck1ff
```

```
# make the wide table into a narrow table

tg_tolerance_wide <- tg_tolerance %>%
  pivot_longer(cols = c(`0hr_tg`, `2hr_tg`, `4hr_tg`),  # Specify the columns
to gather into 1 column
               names_to = "time",  # New column name for the old column names
```

```
              values_to = "TG_conc",
              values_drop_na = TRUE)  # New column name for the values, drop
NA values

head(tg_tolerance_wide)
```

```
## # A tibble: 6 × 4
##   animal_num genotype time   TG_conc
##        <dbl> <chr>    <chr>    <dbl>
## 1          1 Pck1ff   0hr_tg      73
## 2          1 Pck1ff   2hr_tg    1199
## 3          1 Pck1ff   4hr_tg    2124
## 4          2 Pck1ff   0hr_tg      63
## 5          2 Pck1ff   2hr_tg     742
## 6          2 Pck1ff   4hr_tg    6577
```

```
#remember to deal with NA and NAN values! na.rm = TRVE

summary_stats <- tg_tolerance_wide %>%
  group_by(genotype, time) %>%
  summarize(
    mean_value = mean(TG_conc, na.rm = TRUE),
    median_value = median(TG_conc, na.rm = TRUE),
    sd_value = sd(TG_conc, na.rm = TRUE),
    min_value = min(TG_conc, na.rm = TRUE),
    max_value = max(TG_conc, na.rm = TRUE)
  )
```

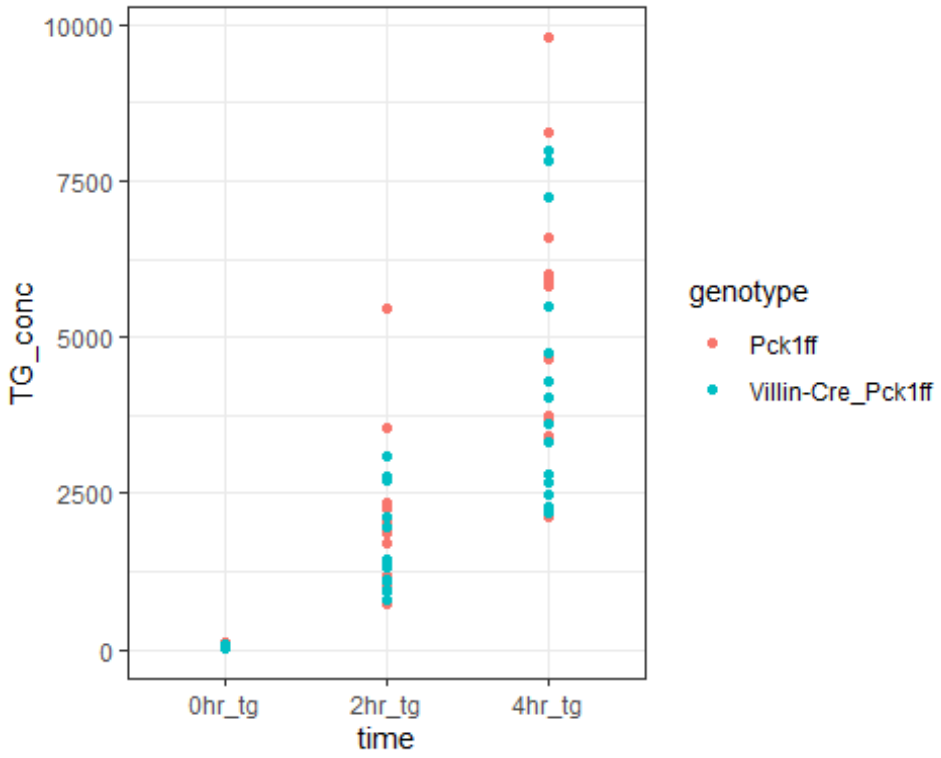
```
## `summarise()` has grouped output by 'genotype'. You can override using the
## `.groups` argument.
```

```
print(summary_stats)
```

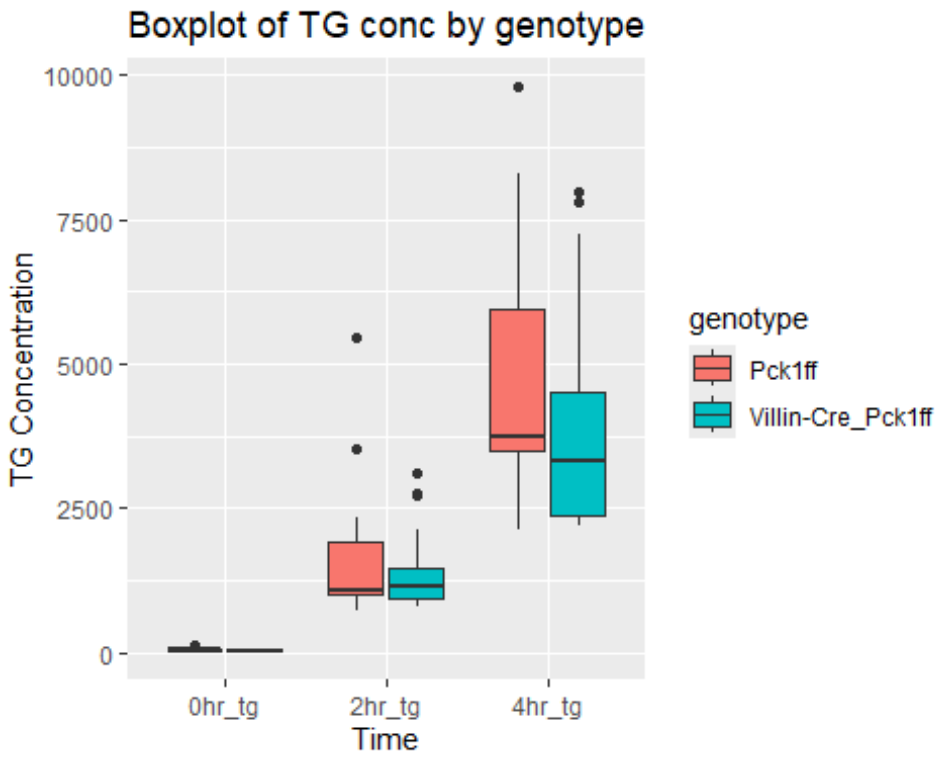
```
## # A tibble: 6 × 7
## # Groups:   genotype [2]
##   genotype          time   mean_value median_value sd_value min_value
max_value
##   <chr>             <chr>       <dbl>        <dbl>    <dbl>     <dbl>
<dbl>
## 1 Pck1ff            0hr_tg       61.5         51.5     26.9        33
125
## 2 Pck1ff            2hr_tg     1606.        1077     1091.        742
5449
## 3 Pck1ff            4hr_tg     4803.        3727     1935.       2124
9799
## 4 Villin-Cre_Pck1ff 0hr_tg       47.2         40       15.8        29
82
## 5 Villin-Cre_Pck1ff 2hr_tg     1434.        1138      661.        803
3111
## 6 Villin-Cre_Pck1ff 4hr_tg     3923.        3314     1940.       2185
7975
```

```r
ggplot(data = tg_tolerance_wide)  + #set data as the tg_tolerance_wide tibble
  geom_point(mapping = aes(x = time, y = TG_conc, color = genotype)) + # map x to time and y to TG_conc and set colorization by genotype
  geom_smooth(mapping = aes(x = time, y = TG_conc, linetype = genotype), se = TRUE) +  # map x to time and y to TG_conc and set linetype to genotype and KEEP error shading w/ se
  theme_bw()
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```




```r
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'

ggplot(tg_tolerance_wide, aes(x = time, y = TG_conc, fill = genotype)) +
  geom_boxplot() +
  labs(x = "Time", y = "TG Concentration") +
  ggtitle("Boxplot of TG conc by genotype")
```




```
# NA values removed during pivot_longer() above

# Perform two-way ANOVA

tg_ANOVA <- aov(TG_conc ~ time * genotype, data = tg_tolerance_wide)

summary(tg_ANOVA)
```

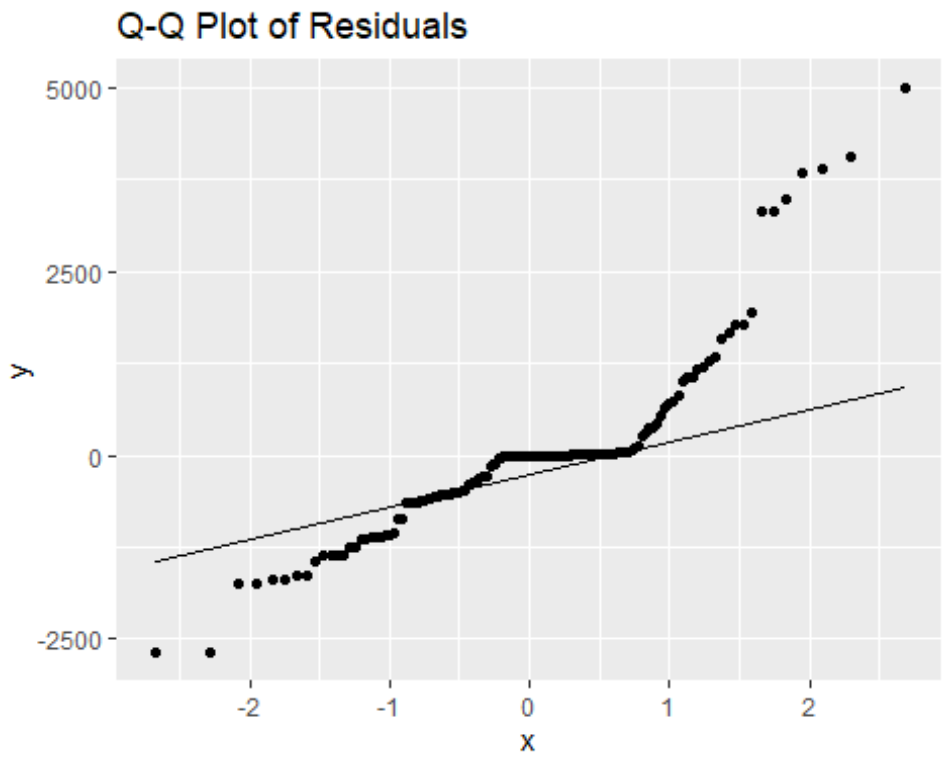
```
##                 Df    Sum Sq   Mean Sq F value Pr(>F)
## time             2 429986265 214993133 141.419 <2e-16 ***
## genotype         1   4269644   4269644   2.809 0.0962 .
## time:genotype    2   4785259   2392629   1.574 0.2112
## Residuals      129 196112629   1520253
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#NORMALITY 1/3:  assign residuals
#Shapiro-Wilk test: This formal test assesses the normality of the residuals.

# Extract residuals into a new column of the df
tg_tolerance_wide$residuals <- residuals(tg_ANOVA)

head(tg_tolerance_wide)
```

```
## # A tibble: 6 × 5
##   animal_num genotype time   TG_conc residuals
##        <dbl> <chr>    <chr>    <dbl>     <dbl>
```

```
## 1          1 Pck1ff   0hr_tg      73     11.5
## 2          1 Pck1ff   2hr_tg    1199   -407.
## 3          1 Pck1ff   4hr_tg    2124  -2679.
## 4          2 Pck1ff   0hr_tg      63      1.45
## 5          2 Pck1ff   2hr_tg     742   -864.
## 6          2 Pck1ff   4hr_tg    6577   1774.
```

```r
#NORMALITY 2/3: Visualize Residuals
# Visual inspection with QQ-plot
ggplot(tg_tolerance_wide, aes(sample = residuals)) +
  stat_qq() +
  stat_qq_line() +
  ggtitle("Q-Q Plot of Residuals")
```




```r
#NORMALITY 3/3: Shappiro WIlk test
# Shapiro-Wilk test for normality
shapiro.test(tg_tolerance_wide$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  tg_tolerance_wide$residuals
## W = 0.84248, p-value = 1.047e-10
```

```r
# High p-value (e.g., p > 0.05): Data is likely normally distributed.
# Low p-value (e.g., p ≤ 0.05): Data is likely NOT normally distributed.
#RESULT: NOT normally distributed, try log transformation
```

```r
# Log transform the 'TG_conc' column
tg_tolerance_wide$TG_conc_log <- log(tg_tolerance_wide$TG_conc)

# Print the transformed data
print(tg_tolerance_wide)
```

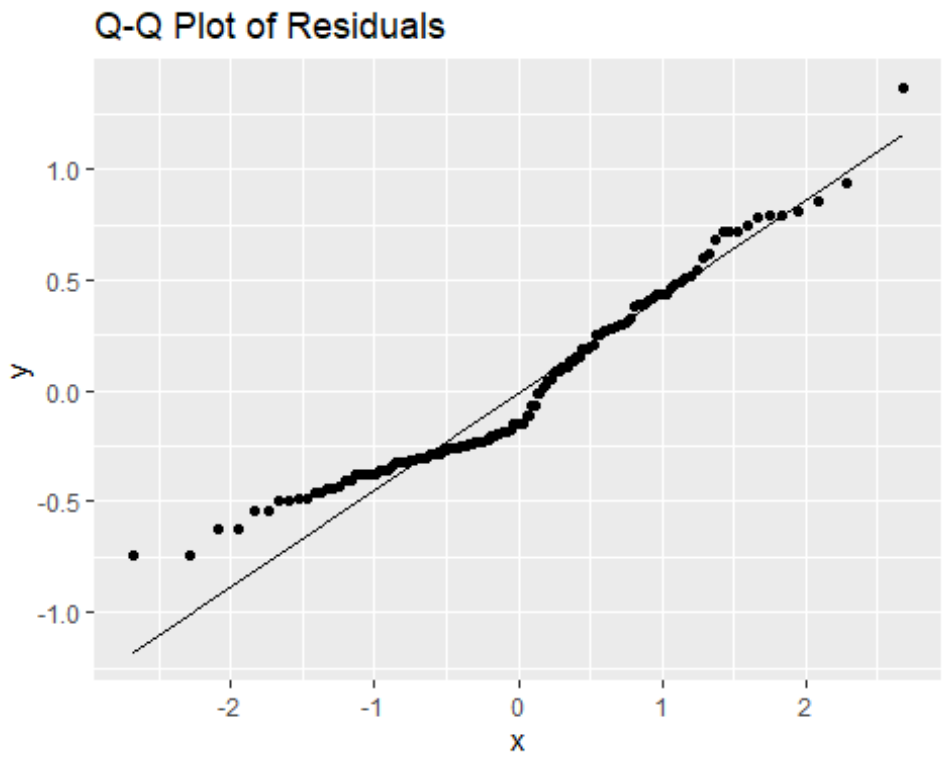
```
## # A tibble: 135 × 6
##    animal_num genotype time   TG_conc residuals TG_conc_log
##         <dbl> <chr>    <chr>    <dbl>     <dbl>       <dbl>
##  1          1 Pck1ff   0hr_tg      73     11.5         4.29
##  2          1 Pck1ff   2hr_tg    1199   -407.          7.09
##  3          1 Pck1ff   4hr_tg    2124  -2679.          7.66
##  4          2 Pck1ff   0hr_tg      63      1.45        4.14
##  5          2 Pck1ff   2hr_tg     742   -864.          6.61
##  6          2 Pck1ff   4hr_tg    6577   1774.          8.79
##  7          4 Pck1ff   0hr_tg      45    -16.5         3.81
##  8          4 Pck1ff   2hr_tg     955   -651.          6.86
##  9          4 Pck1ff   4hr_tg    3436  -1367.          8.14
## 10          5 Pck1ff   0hr_tg      33    -28.5         3.50
## # ℹ 125 more rows
```

```r
# Assuming your data frame is named 'your_data'
# and the columns are named 'dependent_variable', 'factor1', 'factor2'
# NA values removed during pivot_longer() above

# Perform two-way ANOVA
#model <- aov(dependent_variable ~ factor1 * factor2, data = your_data)

# Summarize the ANOVA results
#summary(model)

tg_ANOVA_LOGtf <- aov(TG_conc_log ~ time * genotype, data =
tg_tolerance_wide)

summary(tg_ANOVA_LOGtf)
```

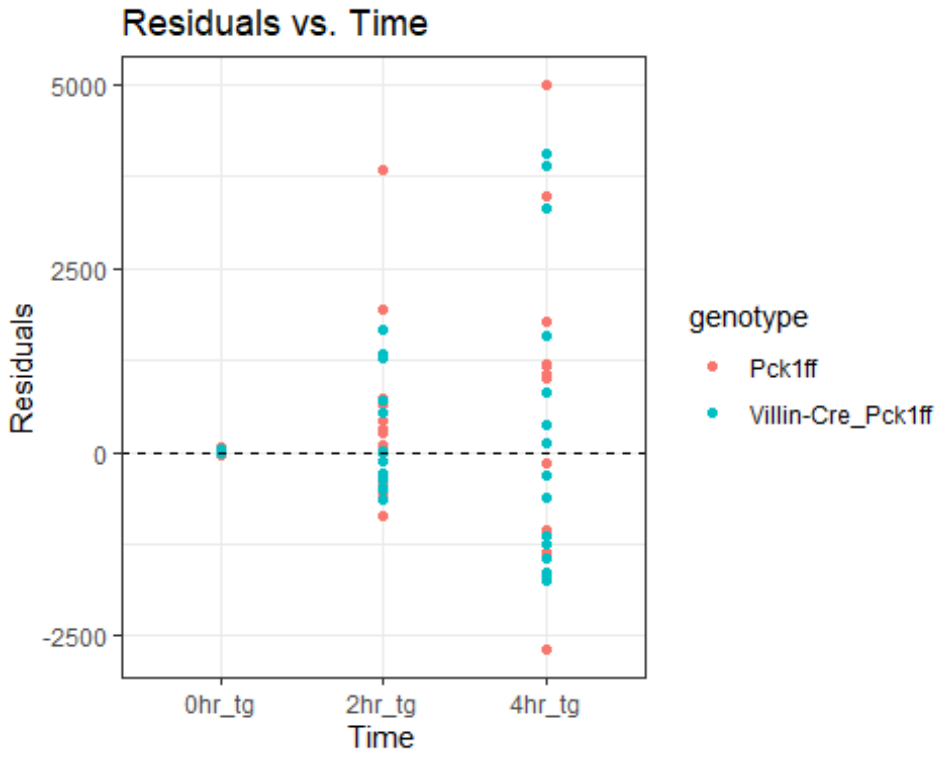
```
##                Df Sum Sq Mean Sq  F value Pr(>F)
## time            2  466.1  233.07 1350.230 <2e-16 ***
## genotype        1    1.0    1.00    5.784 0.0176 *
## time:genotype   2    0.2    0.12    0.719 0.4890
## Residuals     129   22.3    0.17
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# Shapiro-Wilk test for normality
tg_tolerance_wide$residuals_LOG <- residuals(tg_ANOVA_LOGtf)

shapiro.test(tg_tolerance_wide$residuals_LOG)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  tg_tolerance_wide$residuals_LOG
## W = 0.94578, p-value = 3.911e-05

# High p-value (e.g., p > 0.05): Data is likely normally distributed.
# Low p-value (e.g., p ≤ 0.05): Data is likely NOT normally distributed.

# Check for Normality
# Visual inspection with QQ-plot
ggplot(tg_tolerance_wide, aes(sample = residuals_LOG)) +
  stat_qq() +
  stat_qq_line() +
  ggtitle("Q-Q Plot of Residuals")
```




```
#Homoscedasticity 1/1:

ggplot(tg_tolerance_wide, aes(x = time, y = residuals, color = genotype)) +
  geom_point() +
  geom_hline(yintercept = 0, linetype = "dashed") +
  geom_smooth(method = "loess", se = FALSE, color = "blue") +
  labs(title = "Residuals vs. Time", x = "Time", y = "Residuals") +
  theme_bw()

## `geom_smooth()` using formula = 'y ~ x'
```




```
leveneTest(residuals(tg_ANOVA_LOGtf) ~ interaction(time, genotype), data =
tg_tolerance_wide) # interaction term for two-way

## Levene's Test for Homogeneity of Variance (center = median)
##        Df F value Pr(>F)
## group   5  0.2804 0.9232
##       129

# p<0.05; unequal variance
# p>0.05; acceptable variance
```