# Demonstration #2

Exploratory Data Analytics: outliers

Jeffrey Yarus

08 juillet, 2025

## Contents



## 1 Demo Summary

In this demo we present:

1. **An introduction to data analytics**
2. **How to graphically identify outliers in your data**
   - Box plots, Violin plots, Histograms, and scatter plots
3. **How to take the log of a variable**
   - To show a linear relationship with a second (symmetrically distributed) variable in a scatter plot
4. **How to improve tables using additional "kable_styling" arguments**
   - This will improve the look of your output when "knitting"

## 2 Terminology from this Demo

1. **C.I.**
   - Continuous Integration: A software development strategy for teams to increases development speed and quality
2. **<-**
   - Assignment operator; identifies the symbolic name you assigning
3. **Box Plot (box and whisker plot)**
   - It displays the five-number summary of a set of data; the minimum, first quartile, median, third quartile, and maximum
4. **Violin plot**

- Violin plots are a method of plotting numeric data and can be considered a combination box plot and kernel density plot

5. **Chunk**
   - Section of R code
6. **Histogram**
   - Histogram: a type of bar chart displaying the variation in continuous data by grouping numbers into ranges
7. **Scatter Plot / Cross Plot**
   - A type of plot using Cartesian coordinates to display values for typically two variables for a set of data
8. **Interpretive Languages** Coding languages that is generally "interpreted", without compiling a program into machine instructions

# 3 Initializing all of our packages.

```
library(knitr)
library(magrittr)
library(tidyverse)
library(here)
library(ggplot2)
library(GGally)
library(ggpubr)
library(cowplot)
library(ggforce)
library(plotrix)
library(kableExtra)

library(gstlearn)

getwd()
```

```
## [1] "/home/fors/Projets/gstlearn/gstlearn_studies/R/west_texas/gstlearn/scripts"
```

Following the same steps in the previous script to import the West Texas oil field data set as a tibble. We show the

**CHANGE PATHWAY TO MAKE IT UNIVERSAL, FIGURE OUT WHAT IS GOING ON WITH HEAD()!!!**

# 4 Demo Summary

In this demo we present:

1. **An introduction to data analytics**
2. **How to graphically identify outliers in your data**
   - Box plots, Violin plots, Histograms, and scatter plots
3. **How to take the log of a variable**
   - To show a linear relationship with a second (symmetrically distributed) variable in a scatter plot
4. **How to improve tables using additional "kable_styling" arguments**
   - This will improve the look of your output when "knitting"

# 5 Terminology from this Demo

1. **C.I.**

- Continuous Integration: A software development strategy for teams to increases development speed and quality

2. <-
   - Assignment operator; identifies the symbolic name you assigning
3. **Box Plot (box and whisker plot)**
   - It displays the five-number summary of a set of data; the minimum, first quartile, median, third quartile, and maximum
4. **Violin plot**
   - Violin plots are a method of plotting numeric data and can be considered a combination box plot and kernel density plot
5. **Chunk**
   - Section of R code
6. **Histogram**
   - Histogram: a type of bar chart displaying the variation in continuous data by grouping numbers into ranges
7. **Scatter Plot / Cross Plot**
   - A type of plot using Cartesian coordinates to display values for typically two variables for a set of data
8. **Interpretive Languages** Coding languages that is generally "interpreted", without compiling a program into machine instructions

Following the same steps in the previous script to import the West Texas oil field data set as a tibble. We show the

**CHANGE PATHWAY TO MAKE IT UNIVERSAL, FIGURE OUT WHAT IS GOING ON WITH HEAD()!!!**

```
# This chunk reads in the data and peeks at its contents.

file.name <- "../../data/WT-2D-all-outlier.csv"

df <- read_csv(file.name)

df %<>%
  mutate(across(matches("N_|L_"), factor))
```

```
head(df, 20) %>%
  kable(caption = "The Data Frame") %>%
  kable_styling(
    full_width = FALSE,
    latex_options = c("scale_down", "hold_position"),
    bootstrap_options = c("striped", "hover"),
    font_size = 6,
    position = "center"
  ) %>%
  kable_classic(full_width = FALSE)
```

```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

Table 1: The Data Frame

| F_Top | N_Well | C_X_ft | C_Y_ft | L_3_FACIES | P_Delta_t | P_KH_md | P_PHI | P_PHI_pct | P_Thickness | P_Top_ft |
|---|---|---|---|---|---|---|---|---|---|---|
| S_Siltstone | 5001 | 11564 | 5691 | 2 | 60.8 | 3.51 | 0.094 | 9.4 | 28 | 4275 |
| S_Siltstone | 5002 | 10679 | 13706 | 2 | 59.1 | 5.81 | 0.093 | 9.3 | 44 | 4232 |
| S_Siltstone | 5003 | 6311 | 36307 | 1 | 55.9 | 0.77 | 0.059 | 5.9 | 53 | 4317 |
| S_Siltstone | 5004 | 2754 | 11437 | 1 | 52.5 | 0.55 | 0.044 | 4.4 | 41 | 4178 |
| S_Siltstone | 5005 | 9386 | 24799 | 3 | 63.3 | 6.29 | 0.099 | 9.9 | 51 | 4262 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| S_Siltstone | 5006 | 10761 | 24755 | 3 | 63.1 | 2.90 | 0.099 | 9.9 | 41 | 4278 |
| S_Siltstone | 5007 | 10769 | 23463 | 3 | 65.9 | 12.32 | 0.113 | 11.3 | 42 | 4264 |
| S_Siltstone | 5008 | 9145 | 23413 | 1 | 57.9 | 0.79 | 0.074 | 7.4 | 51 | 4252 |
| S_Siltstone | 5009 | 8201 | 23557 | 3 | 64.4 | 8.19 | 0.104 | 10.4 | 51 | 4258 |
| S_Siltstone | 5010 | 6727 | 23472 | 1 | 55.8 | 1.45 | 0.071 | 7.1 | 48 | 4200 |
| S_Siltstone | 5011 | 3834 | 23446 | 1 | 53.9 | 1.60 | 0.062 | 6.2 | 67 | 4120 |
| S_Siltstone | 5012 | 2780 | 23494 | 1 | 54.8 | 1.57 | 0.063 | 6.3 | 61 | 4104 |
| S_Siltstone | 5013 | 3994 | 22098 | 1 | 53.5 | 1.51 | 0.060 | 6.0 | 66 | 4112 |
| S_Siltstone | 5014 | 5482 | 22214 | 1 | 53.9 | 1.79 | 0.065 | 6.5 | 60 | 4153 |
| S_Siltstone | 5015 | 6692 | 22138 | 2 | 60.8 | 8.21 | 0.091 | 9.1 | 57 | 4191 |
| S_Siltstone | 5016 | 9324 | 22199 | 3 | 63.5 | 4.48 | 0.100 | 10.0 | 45 | 4230 |
| S_Siltstone | 5017 | 10817 | 22305 | 2 | 61.7 | 2.73 | 0.093 | 9.3 | 46 | 4253 |
| S_Siltstone | 5018 | 10761 | 20774 | 1 | 57.4 | 2.68 | 0.073 | 7.3 | 36 | 4264 |
| S_Siltstone | 5019 | 9350 | 20761 | 3 | 62.5 | 3.42 | 0.096 | 9.6 | 42 | 4225 |
| S_Siltstone | 5020 | 8237 | 20748 | 1 | 57.2 | 3.82 | 0.079 | 7.9 | 44 | 4207 |

## 5.1 Inputs:

We are assigning variables in R to character values containing certain column names of the data set.

```
# We are assigning column names to variables that we just initialized.
xlon <- "C_X_ft"
ylat <- "C_Y_ft"
out_analysis <- "Raw"
property <- "P_PHI_pct"
```

## 5.2 Data Analytics:

This next chunk focuses on understanding the "shape" of the data distribution for a given variable. To do this we will look at a boxplot, a violin plot, and a frequency distribution, or "histogram." The R package that we will generally use is called "**ggplot2**." We will use the ggplot2 package to build these various graphics. Throughout this course you will get very accustomed to using ggplot2. It reads the data frame that we read in along with the property (variable name) you assign to "**property**" from the previous chunk.

### 5.2.1 Boxplot and Violin Plot:

A "boxplot" or "box and whiskers" plot is a graphical representation of 5 summary statistical metrics; minimum value, first quartile (25% of the data appears below this value), median (second quartile, 50% below), third quartile (75% below), and maximum. The whiskers of a boxplot are the lines that extend from the first quartile to the minimum and the third quartile to the maximum value. The boxplot was originally introduced by John Tukey in 1969. They are particularly useful in comparing distributions across groups.

A "violin" plot is a way to visualize a smoothed "shape" of some data, as a kind of continuous replacement for the discrete histogram. It has the same summary statistics as a histogram or box plot but additionally shows the smoothed probability density of the data continuously across the entire distribution. Simply stated, the violin plot is like a smoothed version of the histogram shape. Instead of frequency on its y-axis, it measures density (scaled) so that the area under the curve is equal to 1. Also, the density curve is plotted on on both sides like mirror images and takes on the appearance of a "violin."

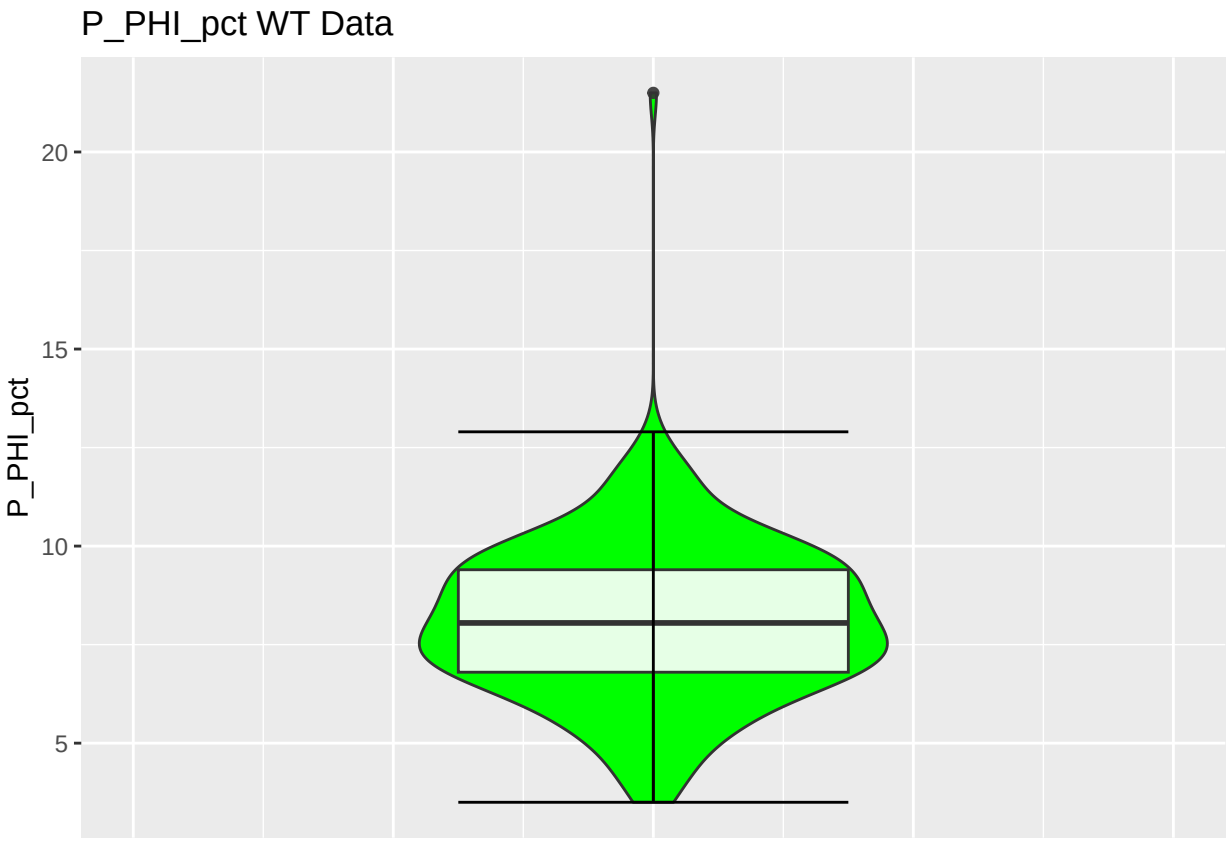
In the plot below, we overlay the boxplot with the violin plot. If you comment (hashtag at the beginning of the line) out line 104 and execute the chunk, you will see only the boxplot.

Does the graphic help you see an outlier in this data?

```
# We are using a standard pipe in this operation. Simply put, we are pushing
#the df variable into the ggplot() function. There is no overwriting (dual
# direction) property within #this type of pipe.
df %>%

# Here, I am trying to reference a column via an object called 'property',
```

```r
# which is a character string. But, I   am inside a tidyverse function (aes)
# so I cannot reference it directly. This is part of the rules in a tidy
# evaluation. !!sym is used to first turn the character string (property) into
# a symbol, and the double bang unquotes it.
# Recall that 'Property' is an object that is type character, type vector.
# The sym function turns it into something called a symbol, a different type
# of object.  The double "bang" is saying to use this symbol as code.
  ggplot(aes(x = 1, !!sym(property))) +
  geom_violin(fill = "green") +
  geom_boxplot(alpha = .9) +  #if you set the alpha value at 0, you will not
# see the outlier
  stat_boxplot(geom = "errorbar") +
  # The theme can control the title, text, and tics.
  theme(
    # Removes the tics.
    axis.ticks.x = element_blank(),
    # Removes the text.
    axis.text.x = element_blank(),
    # Removes the axis numbers.
    axis.title.x = element_blank()) +
  # %s is a placeholder (flag).
  ggtitle(label = sprintf("%s WT Data", property)) +
  xlim(0, 2)
```



### 5.2.2 Histogram:

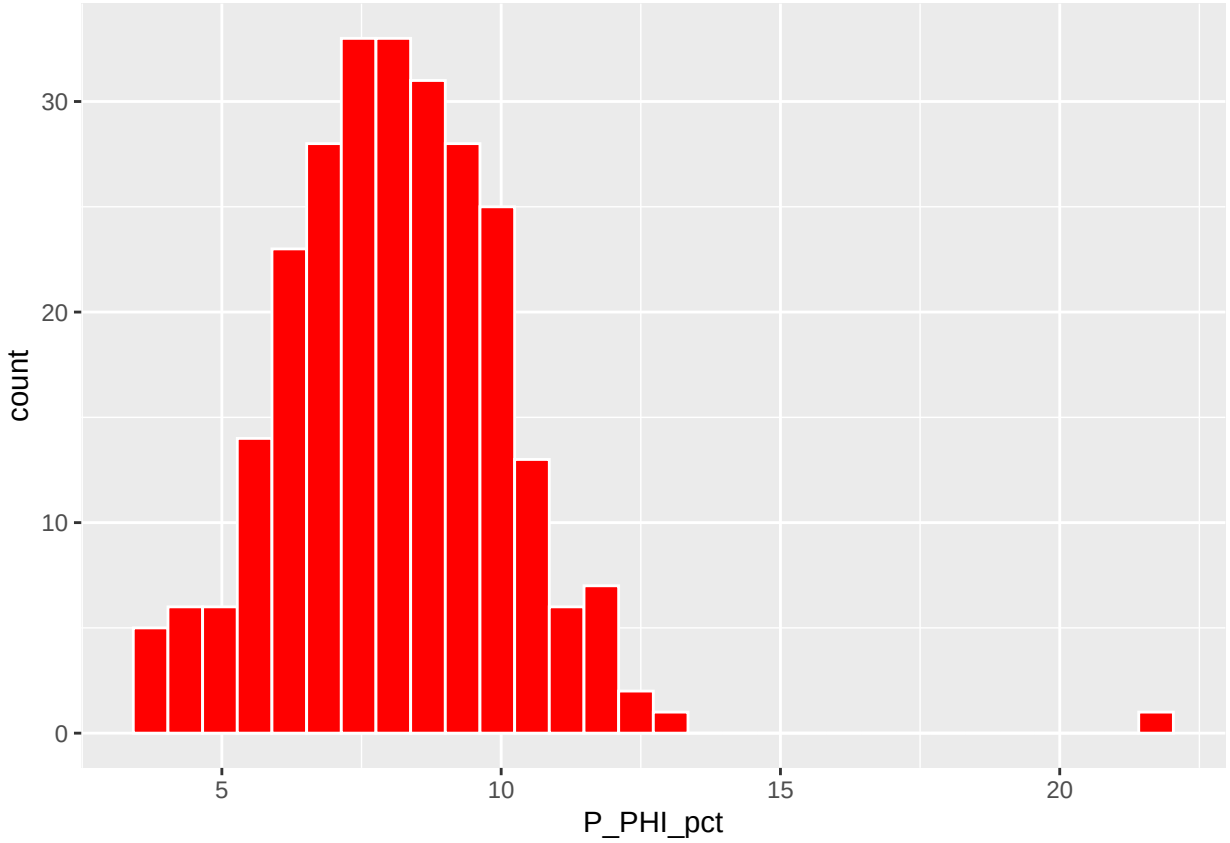
This next chunk builds a histogram, again using the ggplot library previously loaded. A histogram, or frequency distribution, is a graphical representation of the data distribution of a single variable. It record how often values fall within specified intervals or classes. It is distinguished from a bar plot in that the x-axis is continuous, i.e. the end of one interval or class is the beginning of of the next. In a bar chart, classes are categorical and can be arranged in any order.

```
# This is the same general process as shown in the previous chunk, we are just
# applying it to a histogram.
df %>%

  # filter(outlier == FALSE) %>%  #Uncomment to filter outlier

    ggplot(aes(!!sym(property))) +
  geom_histogram(fill = "red", color = "white")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



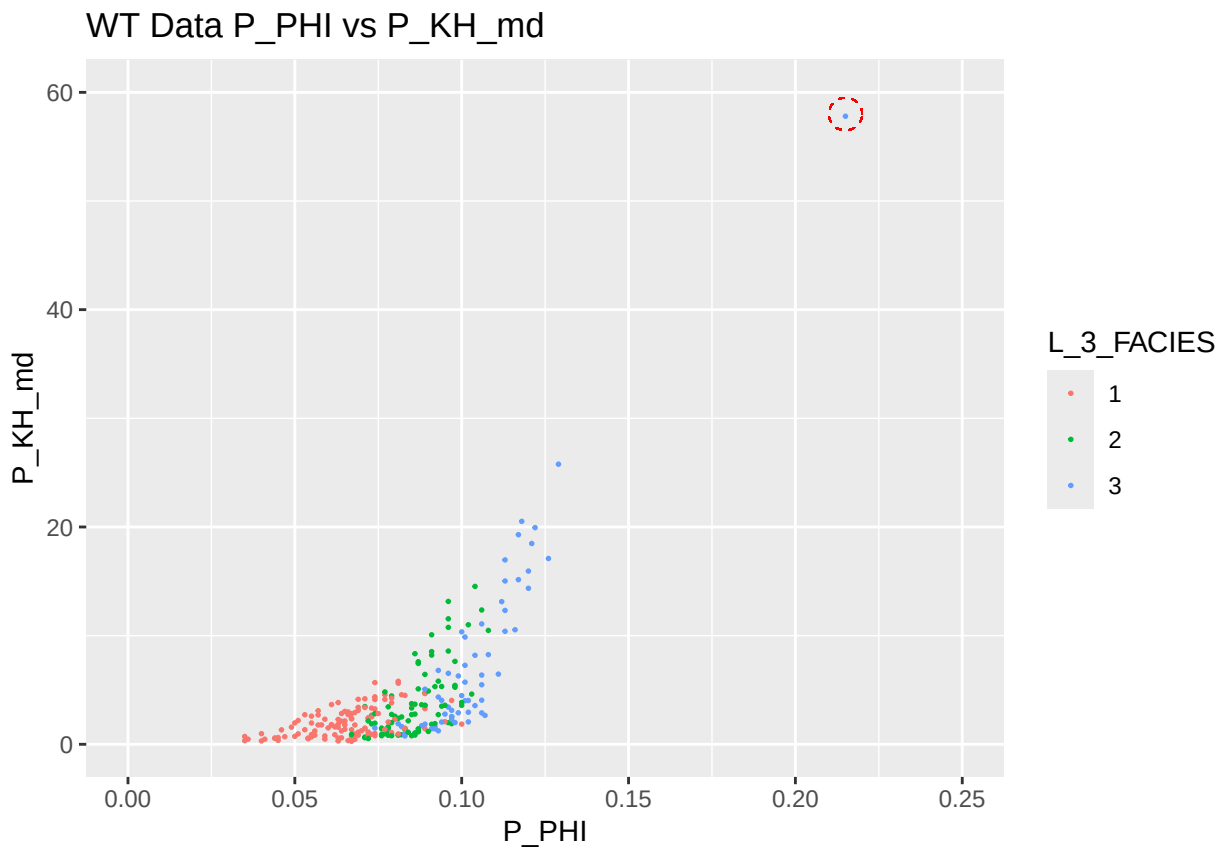

### 5.2.3 Scatter Plot:

Inputs: in this case, the variables are identified as property_1 and property_2. This allows a more generic way of manipulating the various properties in the data frame. By changing the variable name (seen in green as a string variable) you can operate on different inputs.

```
# Assigning the column names to the property variables.
property_1 <- "P_PHI"
property_2 <- "P_KH_md"
```

Here we produce a scatter plot of the initial two properties.# xlim(0, 25) +

```
# Using a pipe and the ggplot2 package to create a graph where we plot property
# 2 vs. property 1 and cassify points based on their facies.
df %>%
  ggplot(
    aes(x = !!sym(property_1),
        y = !!sym(property_2))) +
  # We color each point based on its value in the L_3_Facies column.
  geom_point(
    aes(color = L_3_FACIES),
    size = .3) +
  geom_ellipse(
    size = .3,
    linetype = 2,
    color = "red",
    aes(
      x0 = 0.215,
      y0 = 58,
      a = .005,
      b = 1.5,
      angle = 0
    ),
    color = "black"
  ) +
  ggtitle(
    label = sprintf("WT Data %s vs %s", property_1, property_2)) +
  # Limit for the x-axis bounds.
  xlim(0, .25) +
  # Limit for the y-axis bounds.
  ylim(0, 60)
```

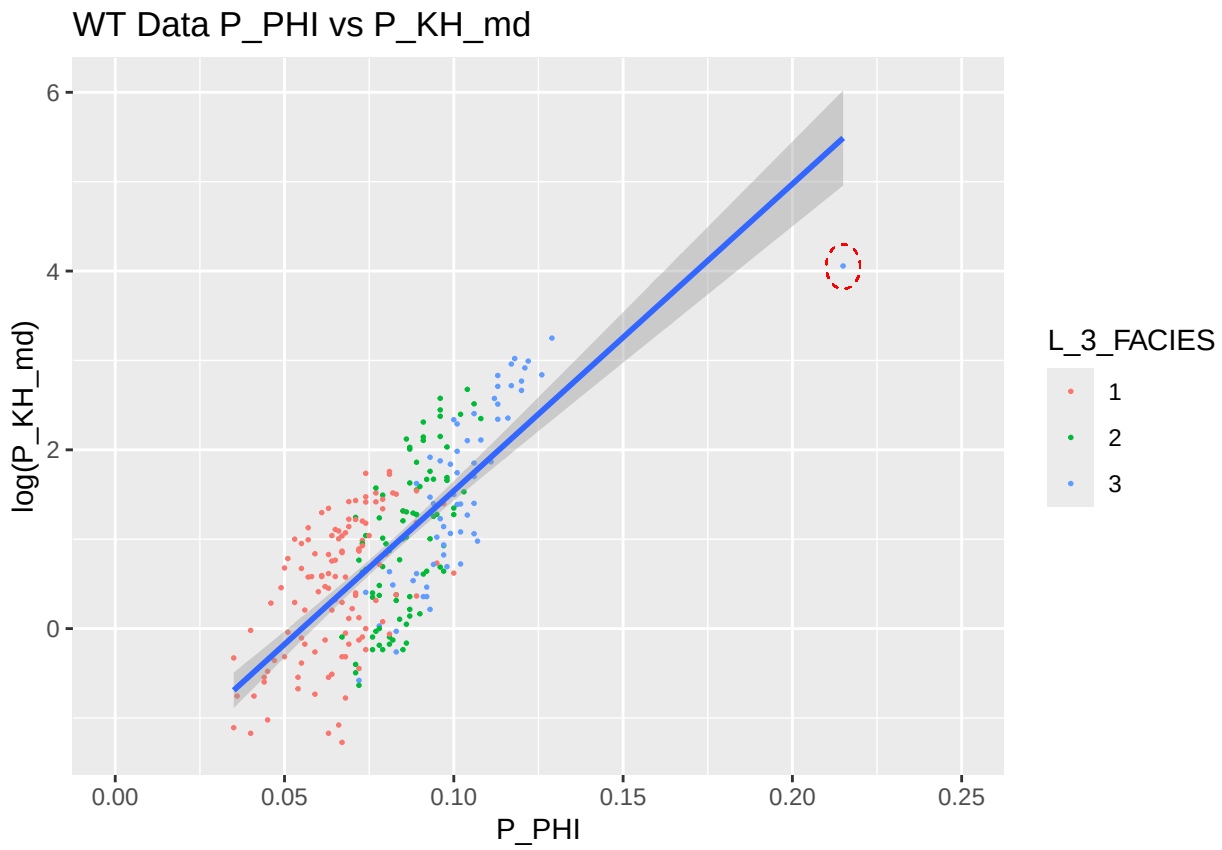



Can you see the outlier?

```
# This is the same plot as above except we are taking the log of property 2 and
# fitting a linear model to our data.
df %>%
ggplot(aes(x = !!sym(property_1), y = log(!!sym(property_2)))) +
  geom_point(
    aes(color = L_3_FACIES),
    size = .3) +
  geom_ellipse(
    size = .3,
    linetype = 2,
    color = "red",
    aes(
      x0 = 0.215,
      y0 = 4.05,
      a = .005,
      b = 0.250,
      angle = 0
    ),
    color = "black"
  ) +
  ggtitle(label = sprintf("WT Data %s vs %s", property_1, property_2)) +
  xlim(0, .25) +
  # Fit and plot a linear model.
  geom_smooth(method = lm)
```

```
## Warning: Duplicated aesthetics after name standardisation: colour
```

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




Can you see the outlier?

```
# Using a pipe to show a summary of the data frame. You can also simply type
# "summary(df)" instead.  Adding kable_styling helps to produce nice tables
# Create and format a summary table that fits in PDF output
df %>%
  summary() %>%
  kable(caption = "Summary Statistics", row.names = FALSE) %>%
  # Apply uniform column width and clean borders
  column_spec(1, width = "1.0cm", border_left = TRUE) %>%
  column_spec(2:(ncol(summary(df)) - 1), width = "1.0cm") %>%
  column_spec(ncol(summary(df)), width = "1.0cm", border_right = TRUE) %>%
  # Style the table
  kable_styling(
    full_width = FALSE,
    latex_options = c("scale_down", "hold_position"),
    bootstrap_options = c("striped", "hover"),
    font_size = 6.5,
    position = "center"
  ) %>%
  kable_classic(full_width = FALSE)
```

```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

Table 2: Summary Statistics

| F_Top | N_Well | C_X_ft | C_Y_ft | L_3_FACIES | S_Delta_t | P_KH_md | P_PHI | P_PHI_pct | P_Thickness | P_Top_ft |
|---|---|---|---|---|---|---|---|---|---|---|
| Length:262 | 5001 : 1 | Min. : 1198 | Min. : 670 | 1:115 | Min. :48.30 | Min. : 0.280 | Min. :0.03500 | Min. : 3.500 | Min. :21.00 | Min. :4086 |
| Class :character | 5002 : 1 | 1st Qu.: 5310 | 1st Qu.: 8747 | 2: 85 | 1st Qu.:55.20 | 1st Qu.: 1.240 | 1st Qu.:0.06800 | 1st Qu.: 6.800 | 1st Qu.:38.00 | 1st Qu.:4197 |
| Mode :character | 5003 : 1 | Median : 8299 | Median :19536 | 3: 62 | Median :58.75 | Median : 2.415 | Median :0.08050 | Median : 8.050 | Median :48.50 | Median :4254 |
| NA | 5004 : 1 | Mean : 8613 | Mean :19174 | NA | Mean :58.52 | Mean : 3.984 | Mean :0.08112 | Mean : 8.112 | Mean :47.92 | Mean :4269 |
| NA | 5005 : 1 | 3rd Qu.:11858 | 3rd Qu.:28828 | NA | 3rd Qu.:61.77 | 3rd Qu.: 4.365 | 3rd Qu.:0.09400 | 3rd Qu.: 9.400 | 3rd Qu.:58.75 | 3rd Qu.:4310 |
| NA | 5006 : 1 | Max. :18799 | Max. :37746 | NA | Max. :69.90 | Max. :57.800 | Max. :0.21500 | Max. :21.500 | Max. :81.00 | Max. :4677 |
| NA | (Other):256 | NA | NA | NA | NA | NA | NA | NA | NA | NA |

“‘