Exercies

Natali Tckvitishvili

2024-07-28

Applied Statistics in R

Natali Tckvitishvili

Load libraries

```
#install.packages("extraDistr")
#install.packages('tinytex')
#tinytex::install_tinytex()

library(tidyverse)
library(ggplot2)
library(stats)
library(ggpubr)
library(extraDistr)
library(tinytex)
```

Exercise 1

a. load data & add new variable - good

```
wine <- read.csv("winequality-white.csv", sep = ";")
wine <- mutate(wine, good = ifelse(quality > 5, 1, 0))

head(wine)
```

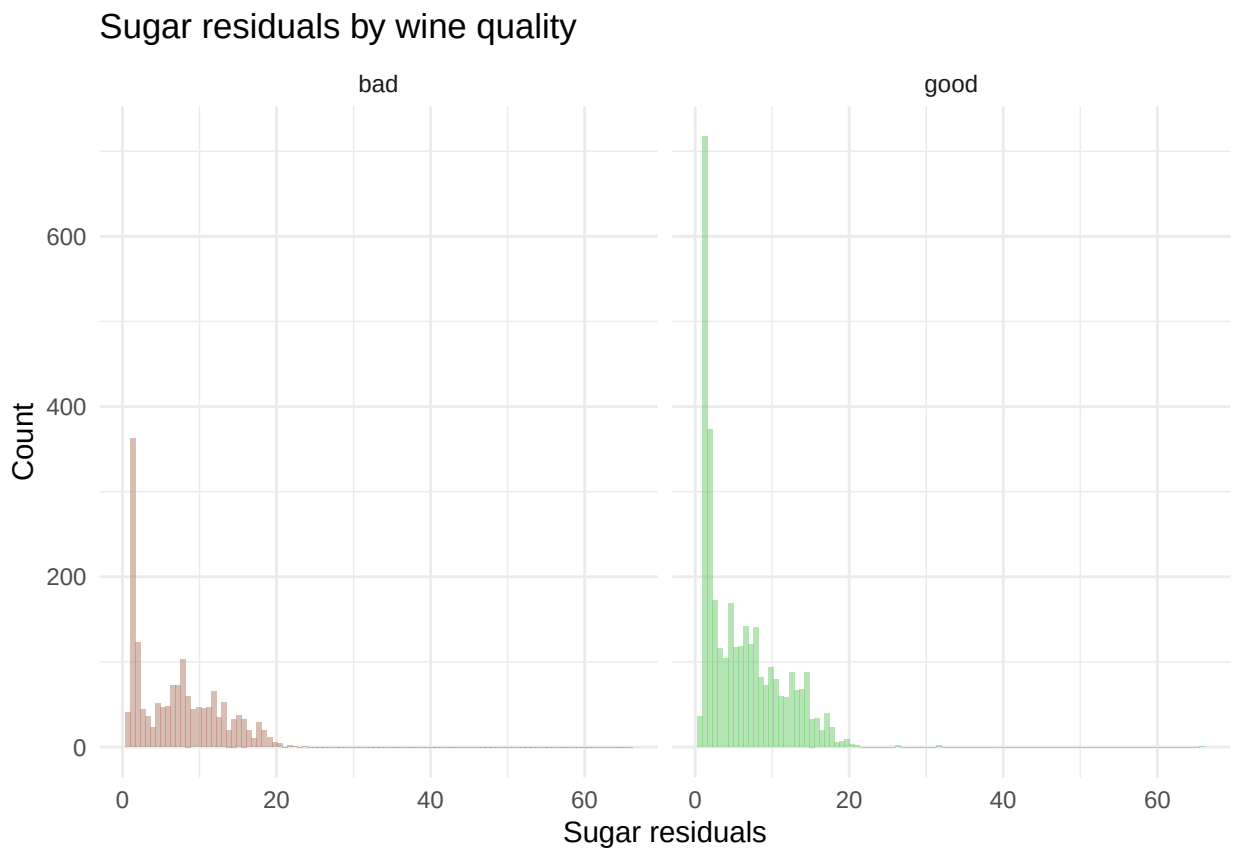
b. residual.sugar analysis

First I'd specify the analysed variable to add more flexibility to the further analysis, when we'll need to make the same calculation for another variable.

```
analysed_variable <- wine$residual.sugar
wine$analysed_variable <- analysed_variable
```

- histograms for good and bad quality wines

```r
good_labels <- c("0" = "bad", "1" = "good")

ggplot(wine, aes(x = analysed_variable, fill = as.factor(good))) +
  geom_histogram(position = "identity", alpha = .5, bins = 100) +
  scale_fill_manual(values=c("#b37d69", "#6dcc6b")) +
  facet_wrap(vars(good), labeller=labeller(good = good_labels)) +
  theme_minimal() +
  theme(legend.position = "none") +
  labs (
    x = "Sugar residuals",
    y = "Count",
    title = "Sugar residuals by wine quality"
  )
```



According to the graphs, both bad and good quality wines have sugar residuals near zero, however, for good wines this number is higher than for bad ones. Moreover, sugar residuals of good quality wines have a smoother decrease in frequency, most of them have less sugar. Therefore, we can assume that sugar residuals may have negative correlation with wine quality.

- summary statistics

```r
summary <- wine %>%
  group_by(good) %>%
  summarise(
    n = n(),
```

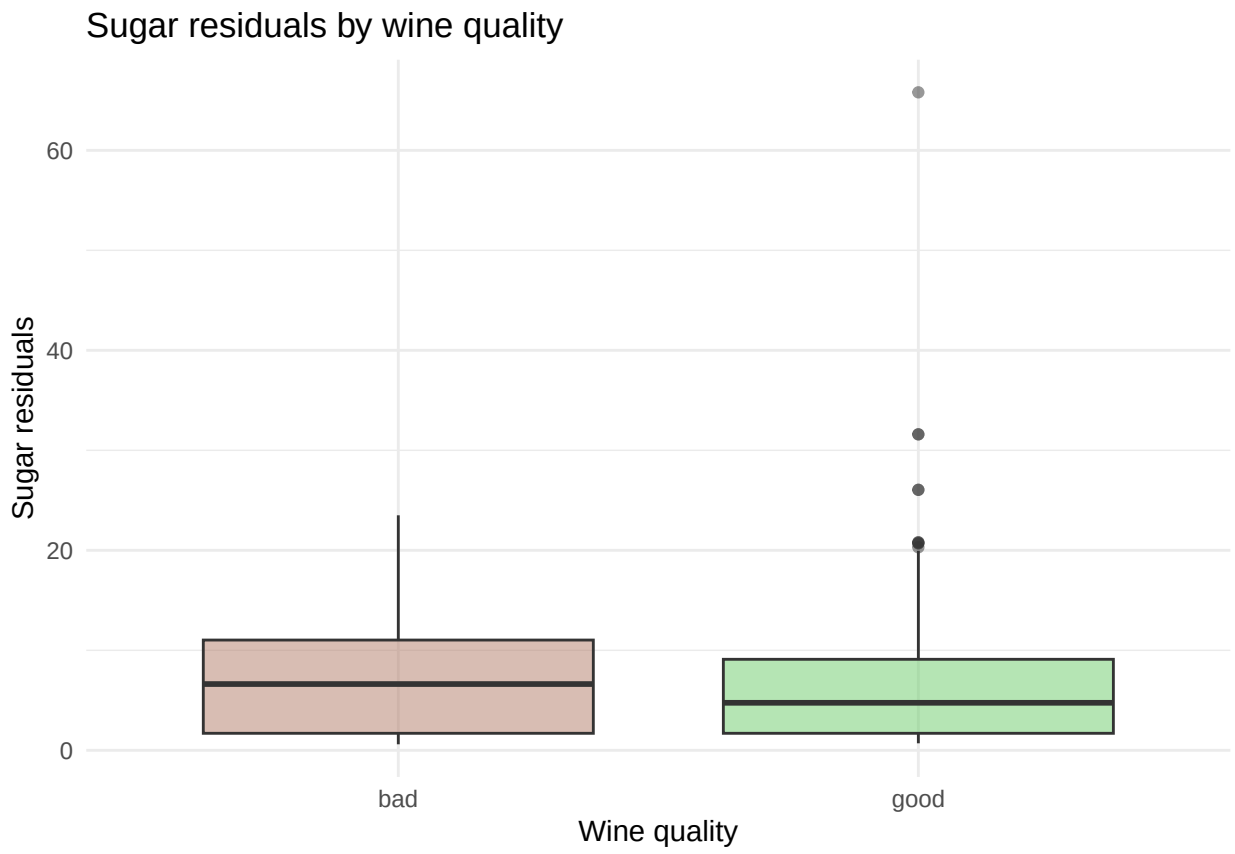
```
    mean = mean(analysed_variable),
    median = median(analysed_variable),
    sd = sd(analysed_variable),
    iqr = IQR(analysed_variable),
    max = max(analysed_variable),
    min = min(analysed_variable)
  )
data.frame(summary)
```

```
##   good    n     mean median       sd   iqr  max min
## 1    0 1640 7.054451  6.625 5.283594 9.325 23.5 0.6
## 2    1 3258 6.057658  4.750 4.929353 7.400 65.8 0.7
```

For the "bad" quality wines both mean and median of the sugar residuals are higher than for the "good" wines, which probably (I say probably here as we haven't checked the significance of this difference yet) means that bad wines on average contain more sugar than good ones. They also have a higher variance and range between the values which may mean that the variety of the bad wines is bigger than of the good ones. What is interesting, there is an observation of a good wine with 65.8 sugar residuals which is a huge number compared to the mean and median. This might be an outlier, we'll see if that's true drawing a boxplot. Additionally, for good wines the difference between the mean and median is quite big, so we can assume that there are more outliers that impact the mean.

- boxplots

```
ggplot(wine, aes(x = as.factor(good), y = analysed_variable, fill = as.factor(good))) +
    geom_boxplot(alpha = .5) +
    scale_fill_manual(values=c("#b37d69", "#6dcc6b")) +
    scale_x_discrete(labels = good_labels) +
    theme_minimal() +
    theme(legend.position = "none") +
    labs (
      x = "Wine quality",
      y = "Sugar residuals",
      title = "Sugar residuals by wine quality"
    )
```

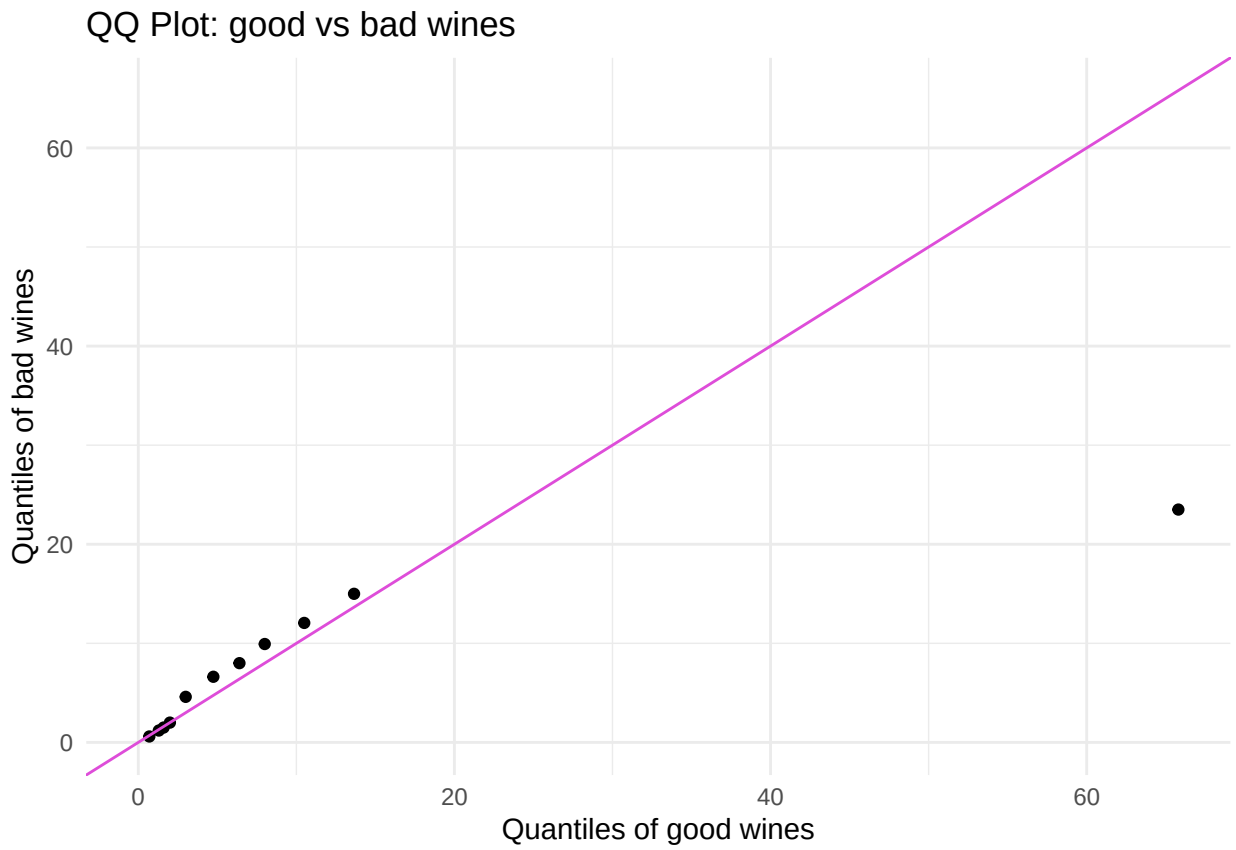


As stated above, the good wine with 65.8 sugar residuals must be an outlier, accoring to the boxplots. There are two more outliers, and all of them impact the mean. Assuming that better wines on average have less sugar, these observations might be a quality estimation error / human factor or there are sugary wines which are considered good in the modern somelier society.

- QQ plot to compare samples

```
good_wine <- wine %>%
  filter(good == 1)

bad_wine <- wine %>%
  filter(good == 0)

quantiles <- seq(0, 1, 0.1)
good_quantiles <- quantile(good_wine$analysed_variable, quantiles)
bad_quantiles <- quantile(bad_wine$analysed_variable, quantiles)

qq_wine <- data.frame(good_quantiles, bad_quantiles)

ggplot(qq_wine, aes(x = good_quantiles, y = bad_quantiles)) +
    geom_point() +
    geom_abline(slope = 1, intercept = 0, color = "#de4dd9") +
    xlim(0, max(good_quantiles, bad_quantiles)) +
    ylim(0, max(good_quantiles, bad_quantiles)) +
    theme_minimal() +
    labs(title = "QQ Plot: good vs bad wines",
```

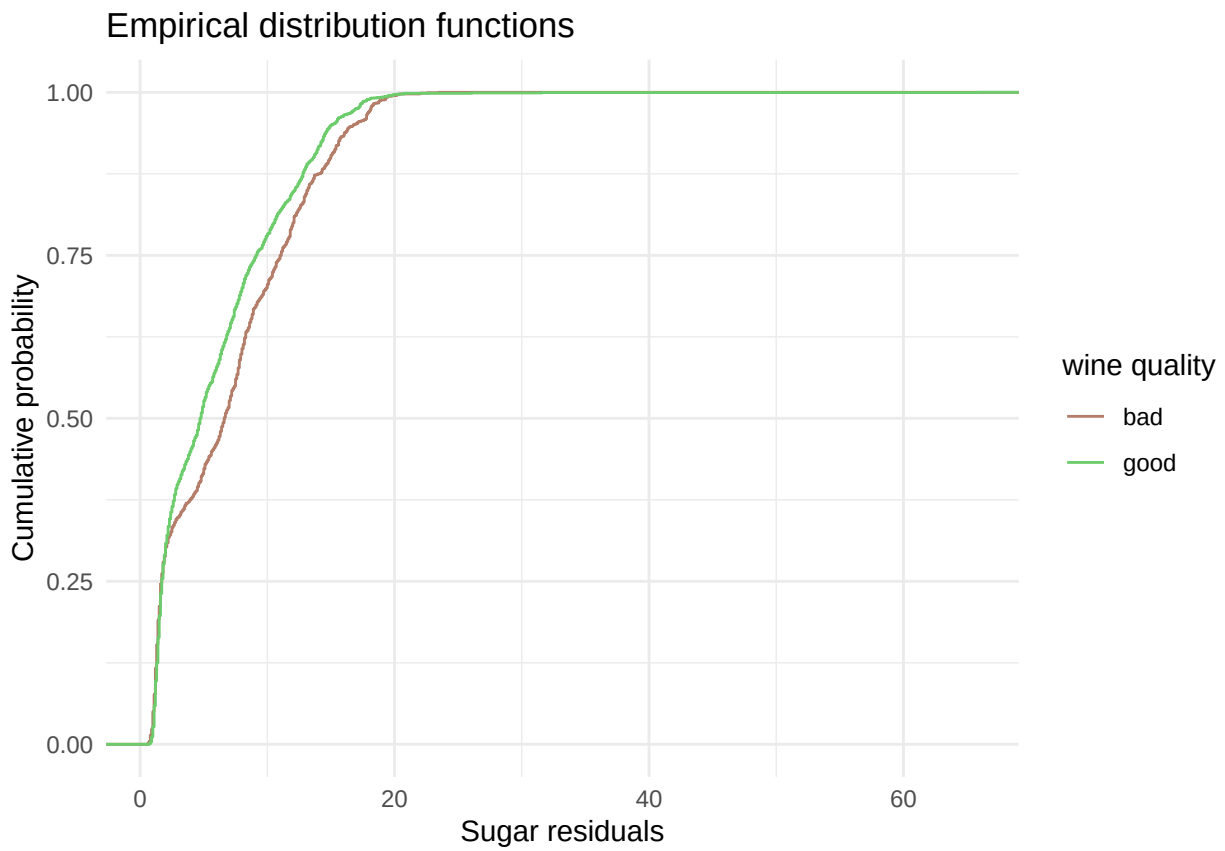
```
        x = "Quantiles of good wines",
        y = "Quantiles of bad wines")
```



The distribution of both samples seems to be similar and right-skewed (this can also be seen on the histograms above). We also see here the outlier.

- Empirical distribution functions

```
ggplot(wine, aes(x = analysed_variable, color = as.factor(good))) +
  stat_ecdf(geom = "step") +
  scale_color_manual(values=c("#b37d69", "#6dcc6b"), labels=good_labels, name="wine quality") +
  theme_minimal() +
  labs (
    x = "Sugar residuals",
    y = "Cumulative probability",
    title = "Empirical distribution functions"
  )
```



The distribution of sugar residuals in good wines is more concentrated around lower values than bad wines. Moreover, values are spread out (graphs are smooth).

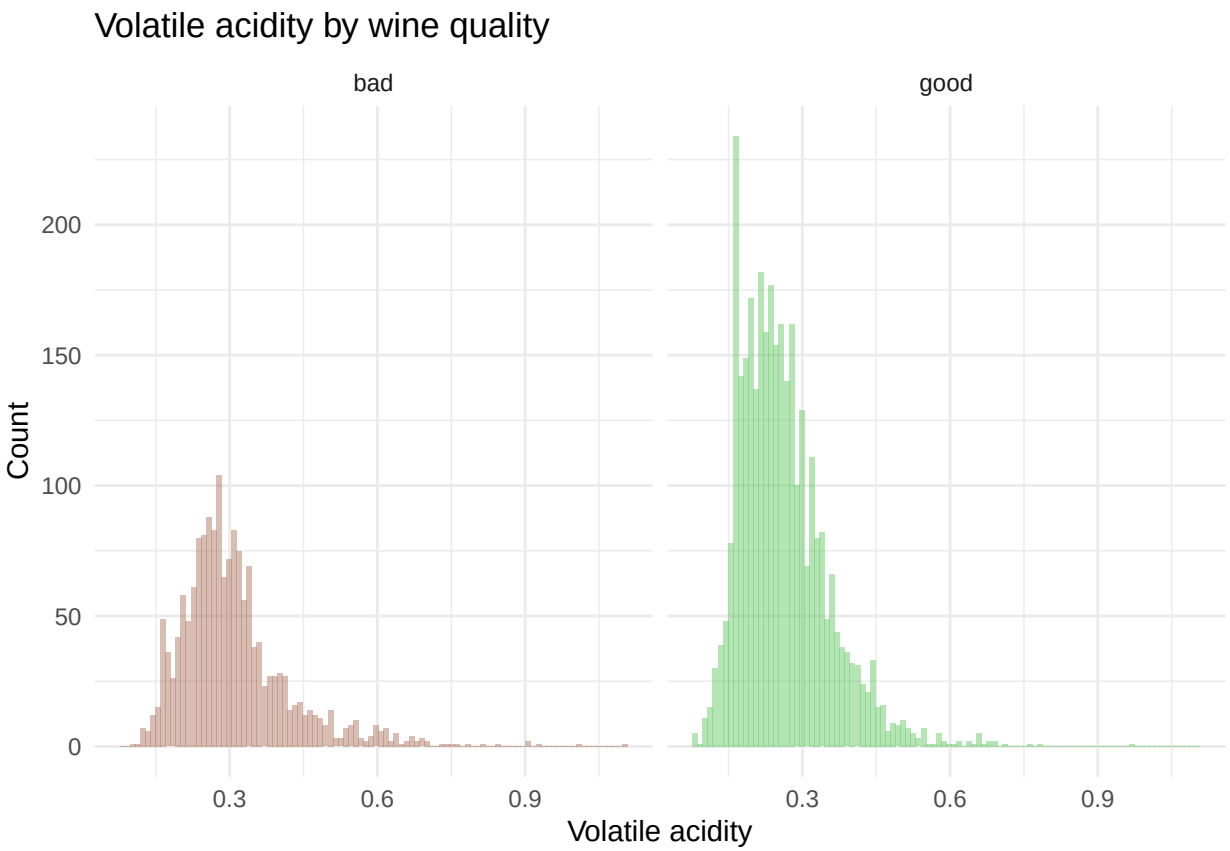
All of those graphs and summary statistics show the same: good wines in general have lower sugar residuals than bad ones.

c. volatile.acidity

```
analysed_variable <- wine$volatile.acidity
wine$analysed_variable <- analysed_variable
```

- histograms for good and bad quality wines

```
ggplot(wine, aes(x = analysed_variable, fill = as.factor(good))) +
  geom_histogram(position = "identity", alpha = .5, bins = 100) +
  scale_fill_manual(values=c("#b37d69", "#6dcc6b")) +
  facet_wrap(vars(good), labeller=labeller(good = good_labels)) +
  theme_minimal() +
  theme(legend.position = "none") +
  labs (
    x = "Volatile acidity",
    y = "Count",
    title = "Volatile acidity by wine quality"
  )
```



It can be said that means of volatile acidity for both good and bad wines do not seem do be significantly different, however, for good wines it may be a little less than for the bad ones. Both distributions are a little right-skewed as well.

- summary statistics

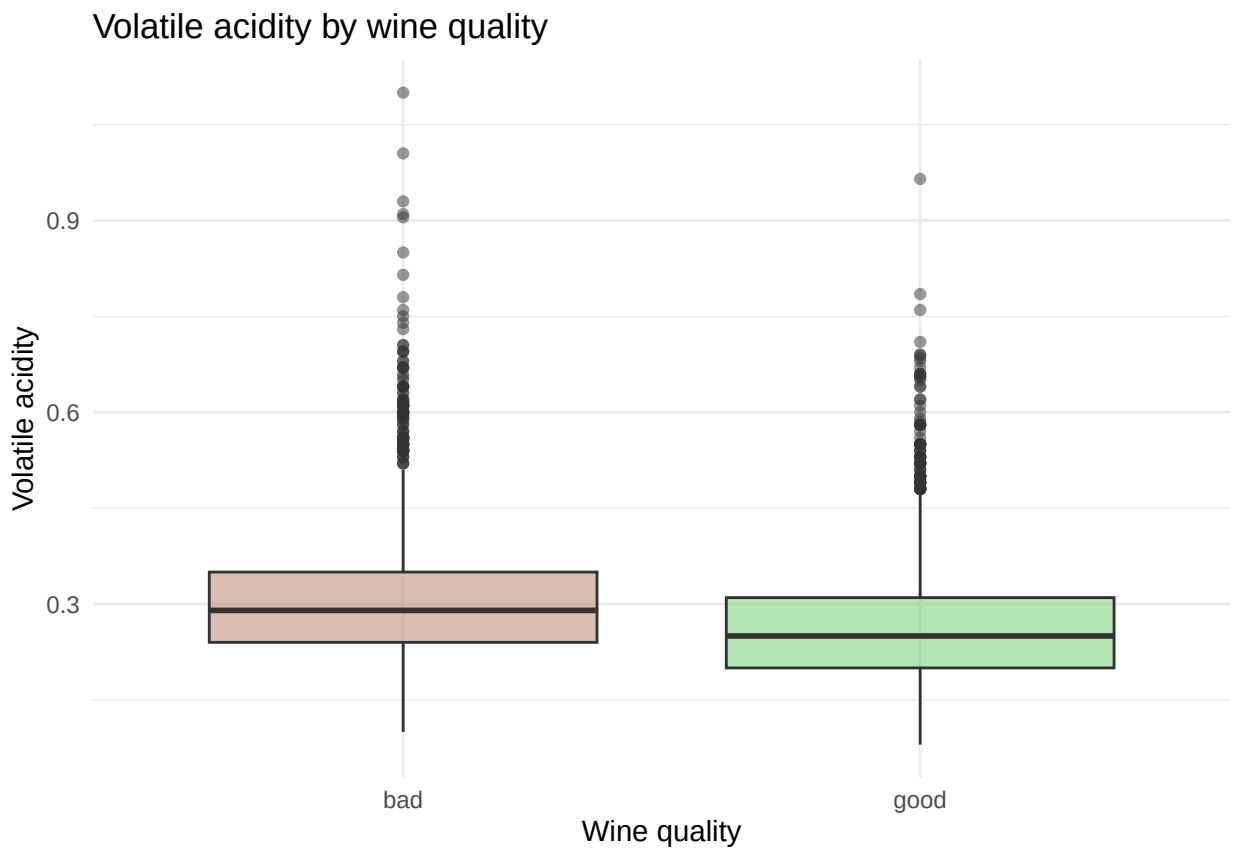
```
summary <- wine %>%
  group_by(good) %>%
  summarise(
    n = n(),
    mean = mean(analysed_variable),
    median = median(analysed_variable),
    sd = sd(analysed_variable),
    iqr = IQR(analysed_variable),
    max = max(analysed_variable),
    min = min(analysed_variable)
  )
data.frame(summary)
```

```
##   good    n      mean median        sd  iqr   max  min
## 1    0 1640 0.3102652   0.29 0.1125479 0.11 1.100 0.10
## 2    1 3258 0.2621209   0.25 0.0901360 0.11 0.965 0.08
```

Similarly to sugar residual, for the "bad" quality wines both mean and median of the volatile acidity are higher than for the "good" wines, although difference is not that huge.

- boxplots

```
ggplot(wine, aes(x = as.factor(good), y = analysed_variable, fill = as.factor(good))) +
    geom_boxplot(alpha = .5) +
    scale_fill_manual(values=c("#b37d69", "#6dcc6b")) +
    scale_x_discrete(labels = good_labels) +
    theme_minimal() +
    theme(legend.position = "none") +
    labs (
      x = "Wine quality",
      y = "Volatile acidity",
      title = "Volatile acidity by wine quality"
    )
```

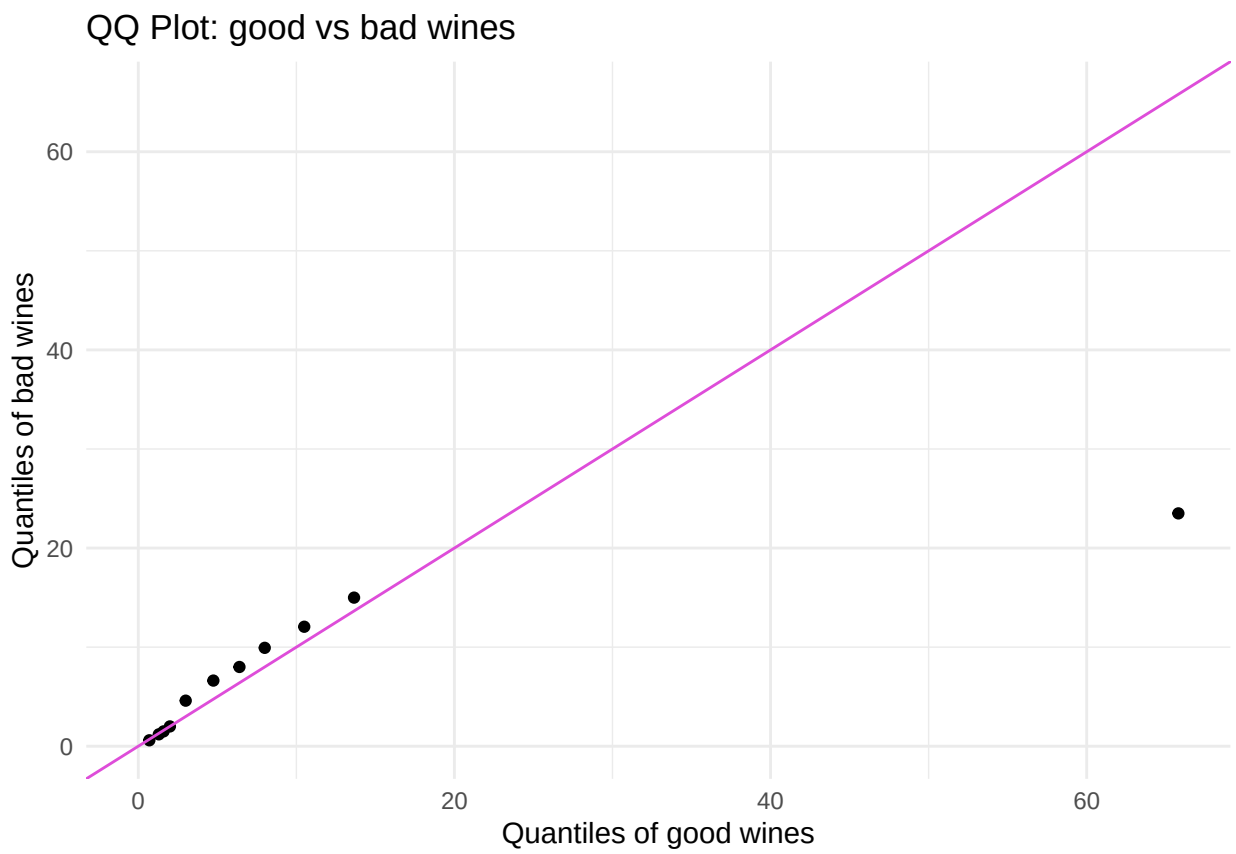


Looks like we have much more outliers here than in sugar residuals. In general, bad wines seem to have more acids (boxplot is located higher).

- QQ plot to compare samples

```
good_quantiles <- quantile(good_wine$analysed_variable, quantiles)
bad_quantiles <- quantile(bad_wine$analysed_variable, quantiles)

qq_wine <- data.frame(good_quantiles, bad_quantiles)

ggplot(qq_wine, aes(x = good_quantiles, y = bad_quantiles)) +
```

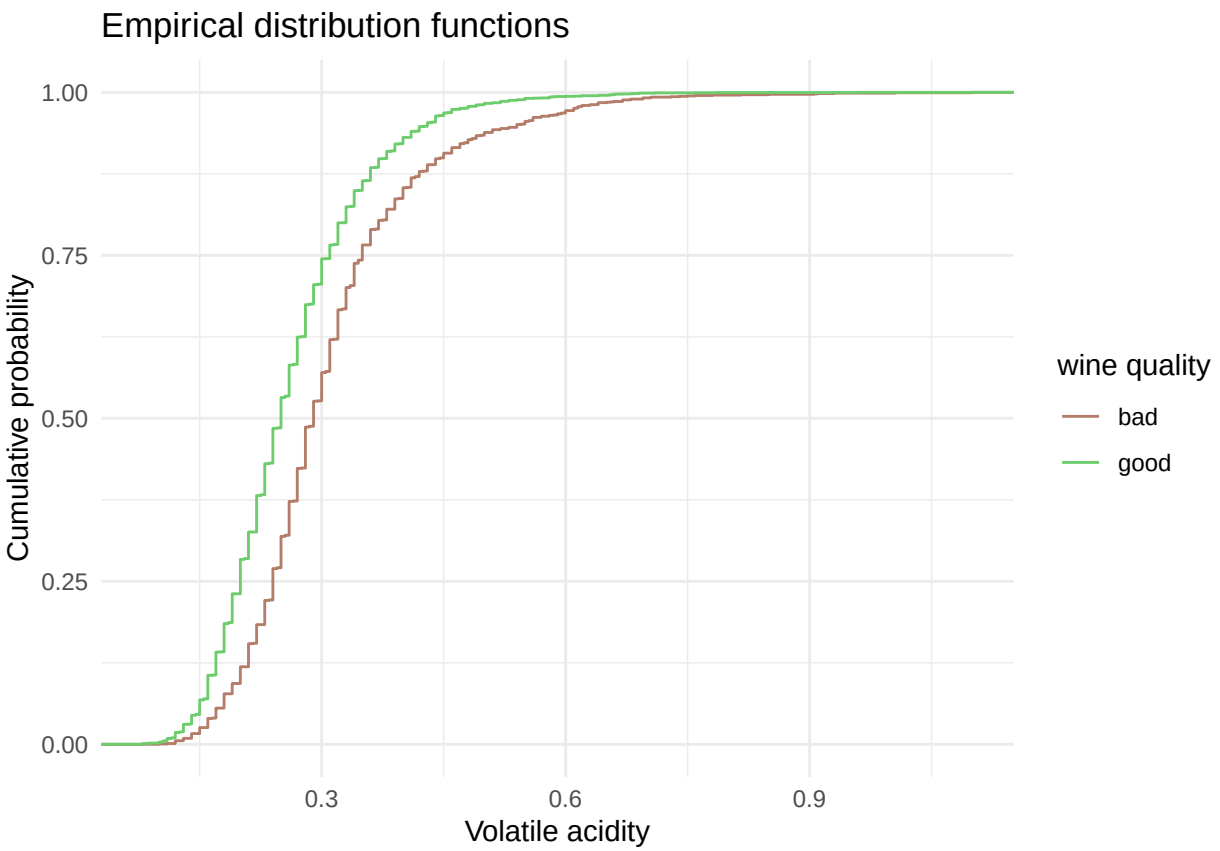
```
    geom_point() +
    geom_abline(slope = 1, intercept = 0, color = "#de4dd9") +
    xlim(0, max(good_quantiles, bad_quantiles)) +
    ylim(0, max(good_quantiles, bad_quantiles)) +
    theme_minimal() +
    labs(title = "QQ Plot: good vs bad wines",
         x = "Quantiles of good wines",
         y = "Quantiles of bad wines")
```



The distribution of both samples seems to be similar but with a difference in scal (variance) as dots do not fall on the y = x line, but still form the straight line.

- Empirical distribution functions

```
ggplot(wine, aes(x = analysed_variable, color = as.factor(good))) +
  stat_ecdf(geom = "step") +
  scale_color_manual(values=c("#b37d69", "#6dcc6b"), labels=good_labels, name="wine quality") +
  theme_minimal() +
  labs (
    x = "Volatile acidity",
    y = "Cumulative probability",
    title = "Empirical distribution functions"
  )
```



Good wines have generally lower values that bad ones; steepness demonstrates that a large number of observations is concentrated within a small range of values. Generally, all graphs incicate that good wines tend to have lower volatile acidity than bad wines.

Exercise 2

```
analysed_variable <- wine$pH
wine$analysed_variable <- analysed_variable
```

a. histogram

mean and standard deviation for plotting normal density

```
mean_pH <- mean(wine$analysed_variable)
sd_pH <- sd(wine$analysed_variable)

mean_pH
```

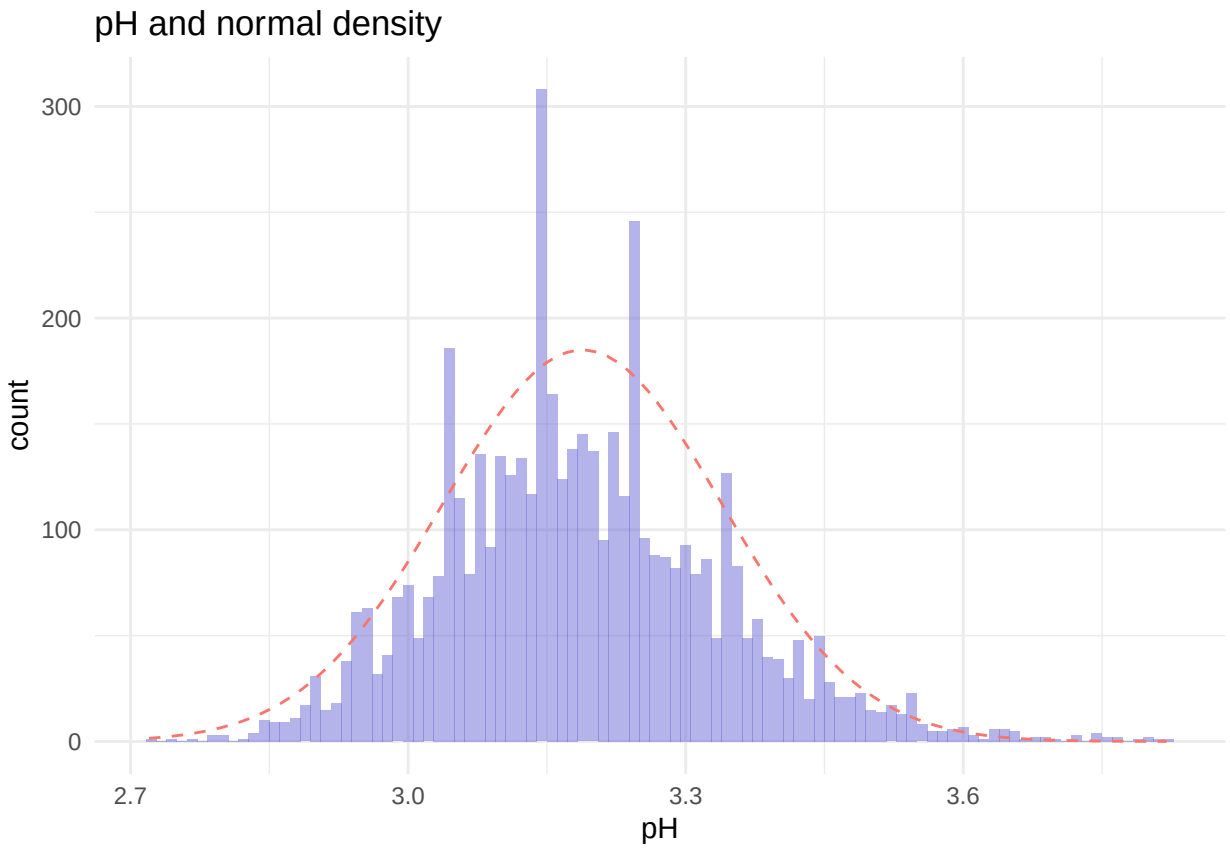
```
## [1] 3.188267
```

```
sd_pH
```

```
## [1] 0.1510006
```

- all wines

```
ggplot(wine, aes(x = analysed_variable)) +
  geom_histogram(position = "identity", alpha = .5, bins = 100, fill="#6a6ad9") +
  stat_function(fun = function(x) # scale normal density to be seen
    dnorm(x, mean = mean_pH, sd = sd_pH) * 70, aes(color = "#d27786"), linetype = "dashed") +
  theme_minimal() +
  theme(legend.position = "none") +
  labs (
    x = "pH",
    y = "count",
    title = "pH and normal density"
  )
```



- good and bad wines

```
good_wine <- wine %>%
  filter(good == 1)

bad_wine <- wine %>%
  filter(good == 0)

mean_pH_good <- mean(good_wine$analysed_variable)
sd_pH_good <- sd(good_wine$analysed_variable)
```

```r
mean_pH_bad <- mean(bad_wine$analysed_variable)
sd_pH_bad <- sd(bad_wine$analysed_variable)

mean_pH_good
```

```
## [1] 3.197231
```

```r
sd_pH_good
```

```
## [1] 0.1535172
```

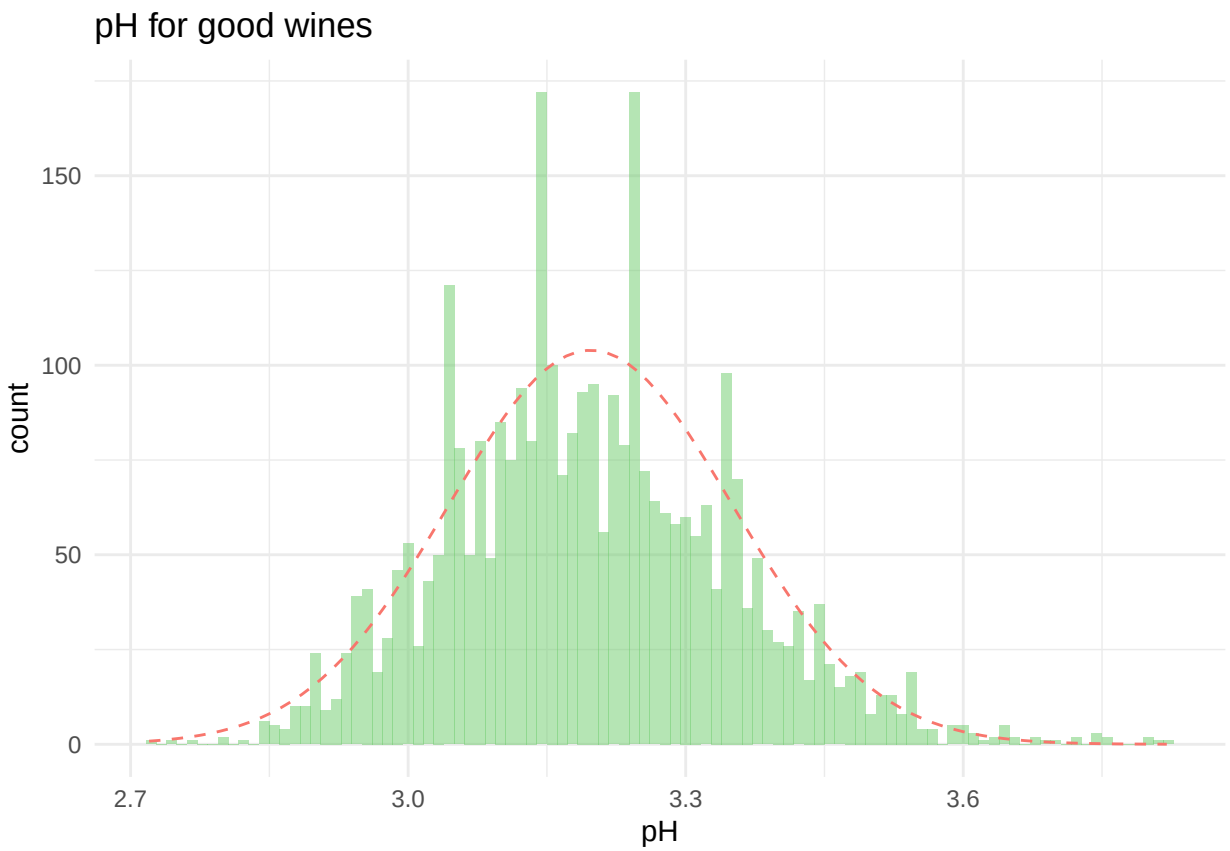
```r
mean_pH_bad
```

```
## [1] 3.170457
```

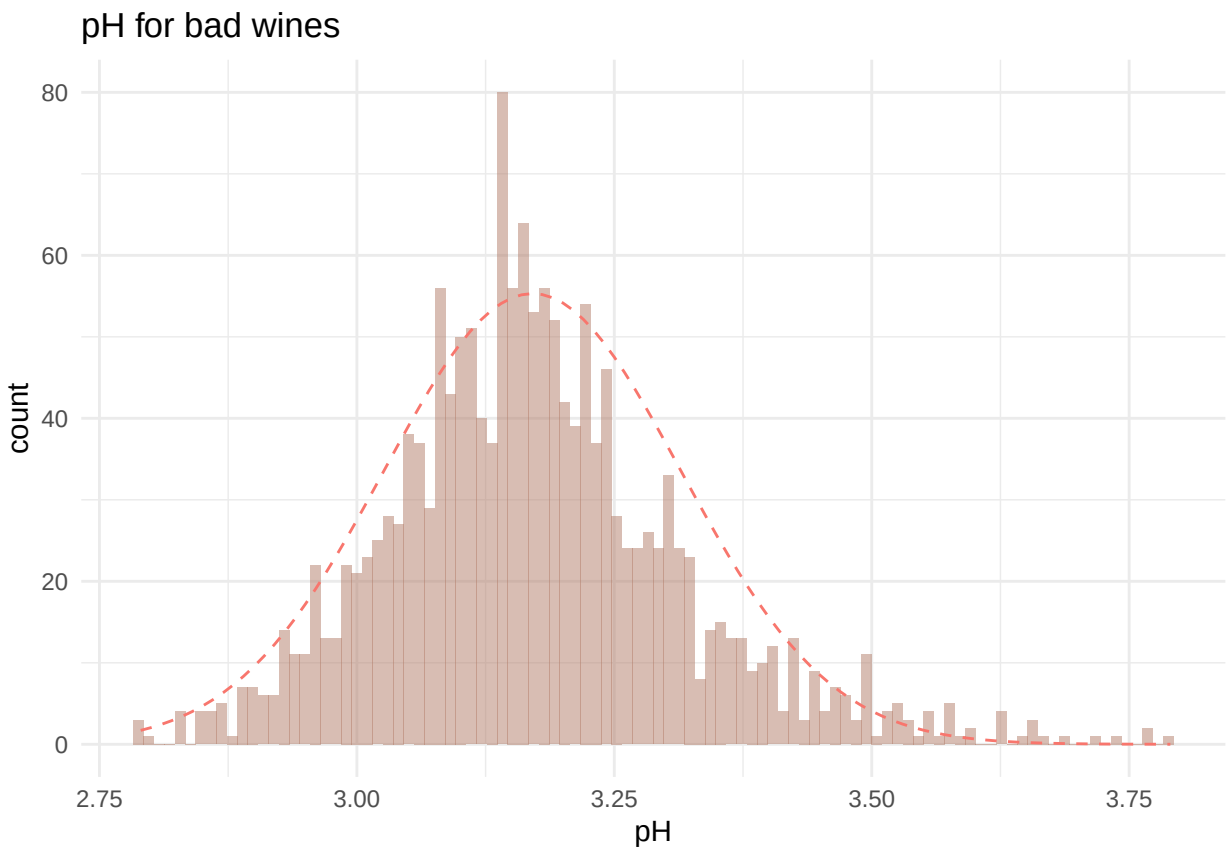
```r
sd_pH_bad
```

```
## [1] 0.1442744
```

```r
ggplot(good_wine, aes(x = analysed_variable)) +
  geom_histogram(position = "identity", alpha = .5, bins = 100, fill="#6dcc6b") +
  stat_function(fun = function(x)
    dnorm(x, mean = mean_pH_good, sd = sd_pH_good) * 40, aes(color = "#d27786"), linetype = "dashed") +
  theme_minimal() +
  theme(legend.position = "none") +
  labs (
    x = "pH",
    y = "count",
    title = "pH for good wines"
  )
```




```
ggplot(bad_wine, aes(x = analysed_variable)) +
  geom_histogram(position = "identity", alpha = .5, bins = 100, fill="#b37d69") +
  stat_function(fun = function(x)
    dnorm(x, mean = mean_pH_bad, sd = sd_pH_bad) * 20, aes(color = "#d27786"), linetype = "dashed") +
  theme_minimal() +
  theme(legend.position = "none") +
  labs (
    x = "pH",
    y = "count",
    title = "pH for bad wines"
  )
```



It can be said that pH follows the normal distribution for bad wines, but for good wines and wines in total the distribution seems to be bimodal with two peaks.

b. QQ plots

- all wines

```
qq_all <- ggplot(wine, aes(sample = pH)) +
  stat_qq() +
  stat_qq_line() +
  theme_minimal() +
  labs (
    x = "theoretical quantiles",
    y = "empirical quantiles"
  )
```

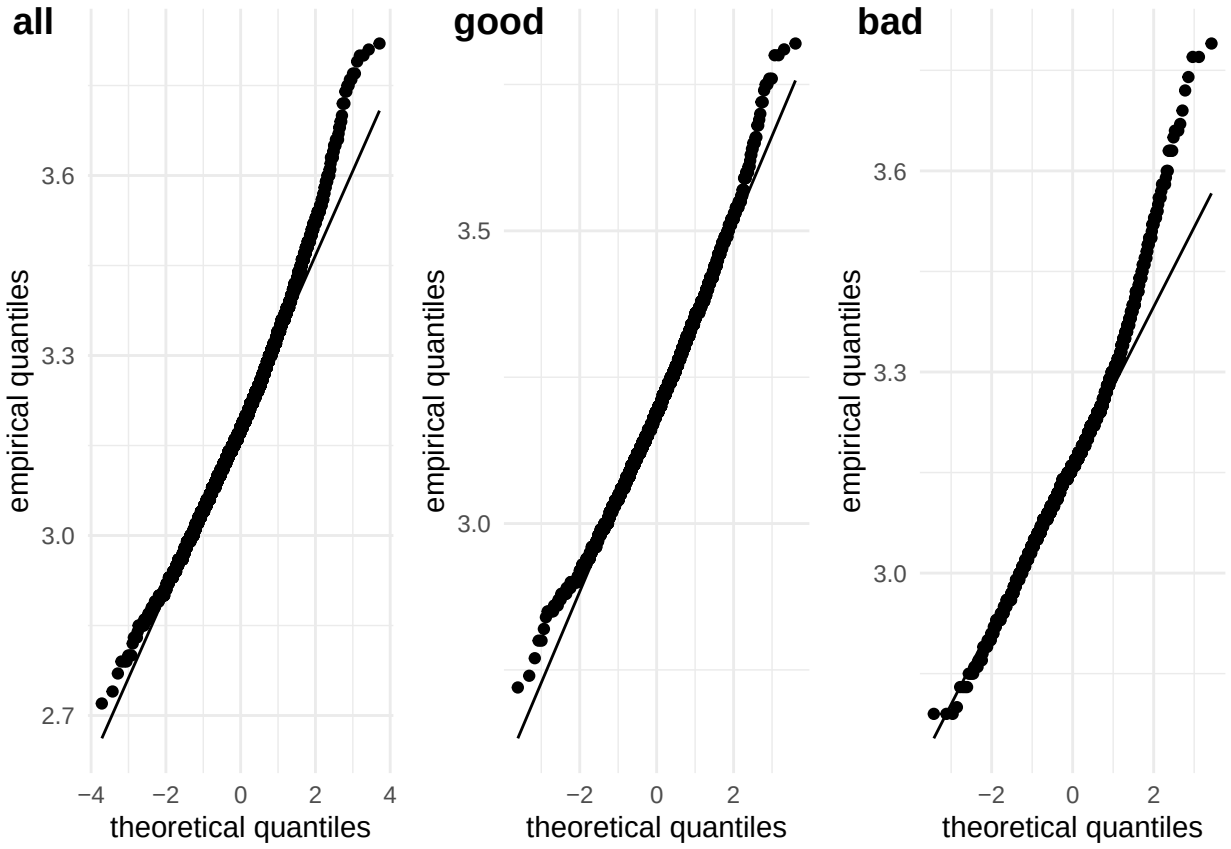
- good wines

```
qq_good <- ggplot(good_wine, aes(sample = pH)) +
  stat_qq() +
  stat_qq_line() +
  theme_minimal() +
  labs (
    x = "theoretical quantiles",
    y = "empirical quantiles"
  )
```

- bad wines

```
qq_bad <- ggplot(bad_wine, aes(sample = pH)) +
  stat_qq() +
  stat_qq_line() +
  theme_minimal() +
  labs (
    x = "theoretical quantiles",
    y = "empirical quantiles"
  )
```

```
ggarrange(qq_all, qq_good, qq_bad, labels = c("all", "good", "bad"), ncol = 3, nrow = 1)
```



b. PP plots

- all wines

```
pp_all <- ggplot(wine, aes(sample = pH)) +
  stat_qq(distribution = qnorm, dparams = list(mean = mean_pH, sd = sd_pH)) +
  stat_qq_line(distribution = qnorm, dparams = list(mean = mean_pH, sd = sd_pH)) +
  theme_minimal() +
  labs (
    x = "theoretical probabilities",
    y = "empirical probabilities"
  )
```

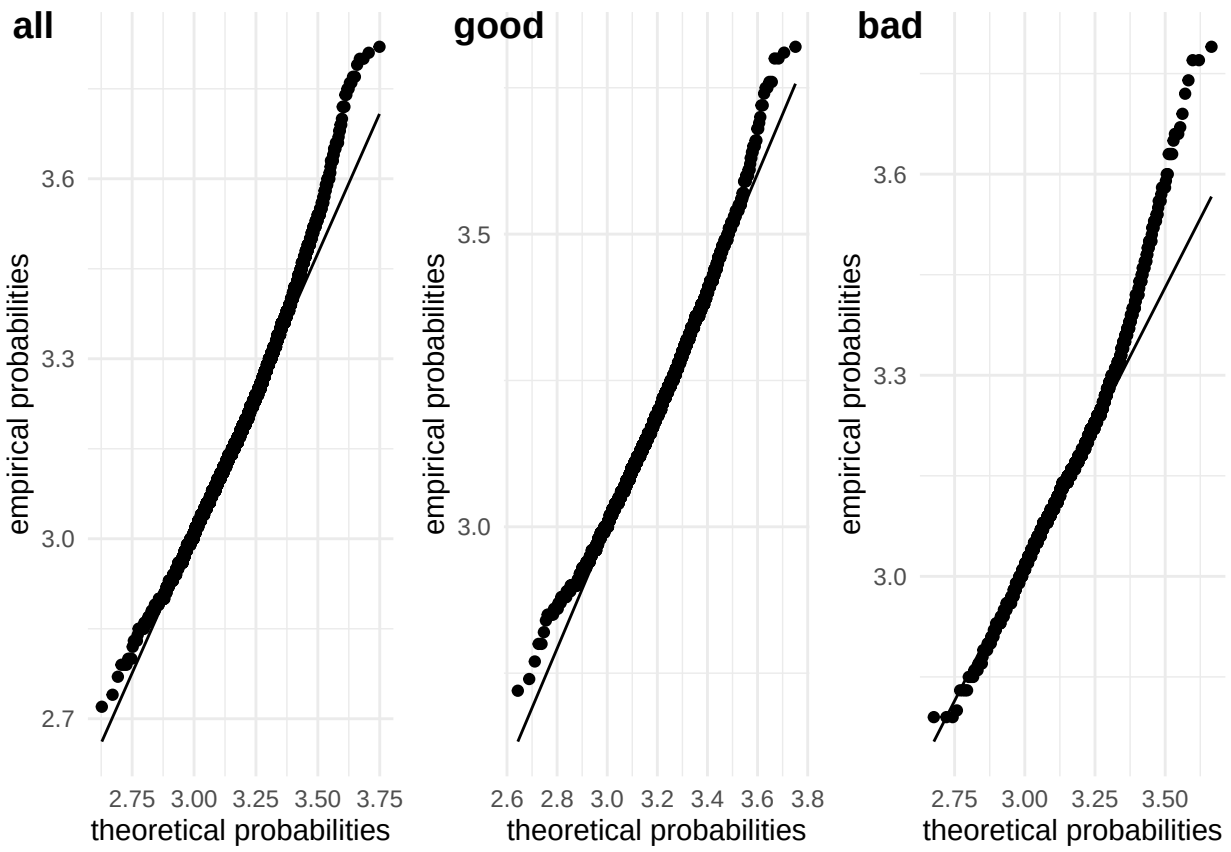
- good wines

```
pp_good <- ggplot(good_wine, aes(sample = pH)) +
  stat_qq(distribution = qnorm, dparams = list(mean = mean_pH_good, sd = sd_pH_good)) +
  stat_qq_line(distribution = qnorm, dparams = list(mean = mean_pH_good, sd = sd_pH_good)) +
  theme_minimal() +
  labs (
    x = "theoretical probabilities",
    y = "empirical probabilities"
  )
```

- bad wines

```
pp_bad <- ggplot(bad_wine, aes(sample = pH)) +
  stat_qq(distribution = qnorm, dparams = list(mean = mean_pH_bad, sd = sd_pH_bad)) +
  stat_qq_line(distribution = qnorm, dparams = list(mean = mean_pH_bad, sd = sd_pH_bad)) +
  theme_minimal() +
  labs (
    x = "theoretical probabilities",
    y = "empirical probabilities"
  )
```

```
ggarrange(pp_all, pp_good, pp_bad, labels = c("all", "good", "bad"), ncol = 3, nrow = 1)
```

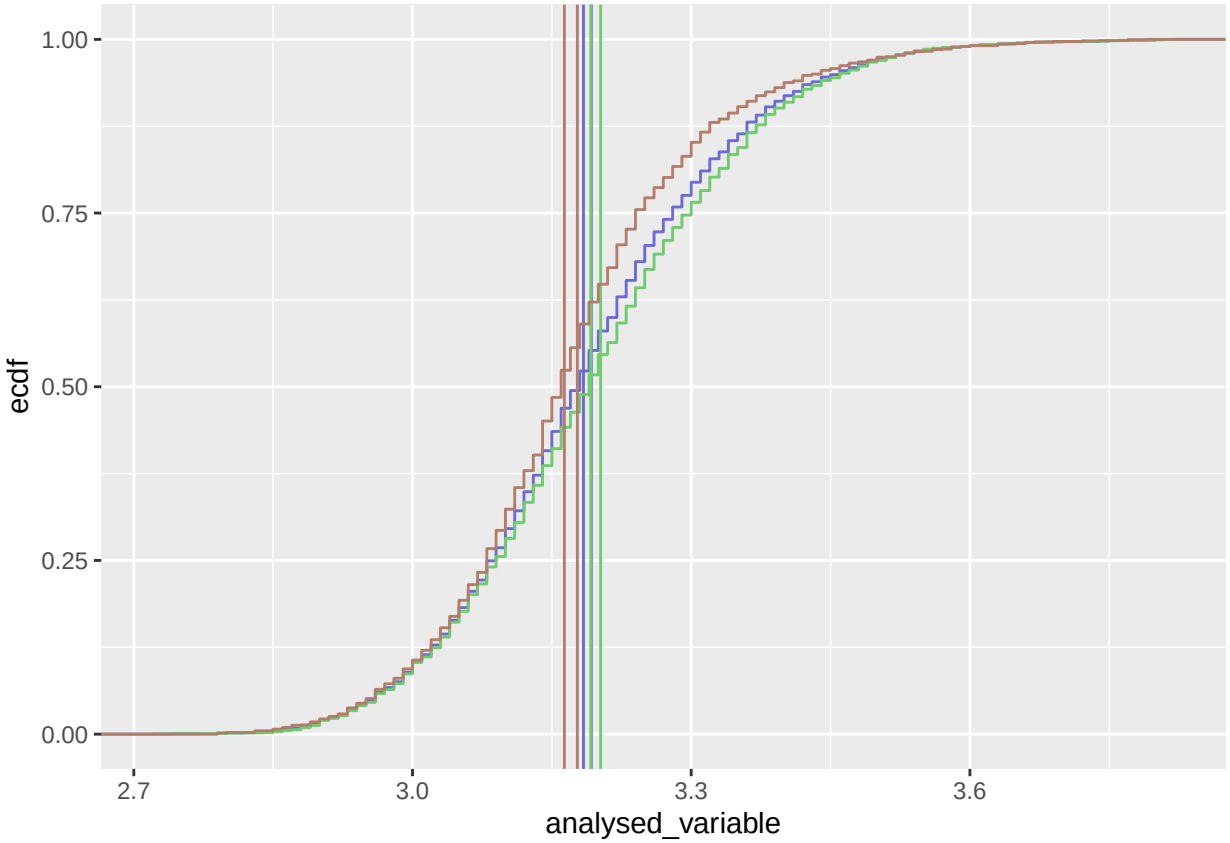


Samples probably do not follow normal distribution, as tails of QQ and PP plots do not lay on the line.

c. empirical distribution function + confidence intervals

```
# calculate confidence bands

bands <- function(data, analysed_variable, alpha) {
  n <- length(data$analysed_variable)
  mean <- mean(data$analysed_variable)
  sd <- sd(data$analysed_variable)
  z_alpha <- qnorm(1 - alpha / 2)
  lower <- mean - z_alpha * sd / sqrt(n)
  upper <- mean + z_alpha * sd / sqrt(n)
  data.frame(lower, upper)
}

alpha <- 0.05
all_bands <- bands(wine, analysed_variable, alpha)
good_bands <- bands(good_wine, analysed_variable, alpha)
bad_bands <- bands(bad_wine, analysed_variable, alpha)
```

```
ggplot() +
  stat_ecdf(data = wine, aes(x = analysed_variable), color = "#6a6ad9") +
  geom_vline(xintercept = all_bands$lower, color = "#6a6ad9") +
  geom_vline(xintercept = all_bands$upper, color = "#6a6ad9") +
  stat_ecdf(data = good_wine, aes(x = analysed_variable), color = "#6dcc6b") +
  geom_vline(xintercept = good_bands$lower, color = "#6dcc6b") +
  geom_vline(xintercept = good_bands$upper, color = "#6dcc6b") +
  stat_ecdf(data = bad_wine, aes(x = analysed_variable), color = "#b37d69") +
  geom_vline(xintercept = bad_bands$lower, color = "#b37d69") +
  geom_vline(xintercept = bad_bands$upper, color = "#b37d69")
```



d. EDF + uniform confidence bands

e. EDFs for good and bad wines

Exercise 3

a. MLE for mu

b. quantile

- for 20 observations

```
set.seed(999)

n <- 20
mu <- 1
sigma <- 1
sample_20 <- rlaplace(n, mu, sigma)

real_median_20 <- median(sample_20)

MLE_20 <- c()
diff_20 <- c()

for(type in 1:9) {
```

```
  MLE_20[type] <- quantile(sample_20, 0.5, type = type)
  diff_20[type] <- MLE_20[type] - real_median_20
}

diff_20
```

```
## [1] -0.01284831  0.00000000 -0.01284831 -0.01284831  0.00000000  0.00000000
## [7]  0.00000000  0.00000000  0.00000000
```

So, types 2, 5, 6, 7, 8, 9 of function quantile predict better than 1, 3 and 4 types Here, the choice of quantile type significantly affects the result because this sample may not be representative due to small number of observations

- for 1000 observations

```
set.seed(999)

n <- 1000
mu <- 1
sigma <- 1
sample_1000 <- rlaplace(n, mu, sigma)

real_median_1000 <- median(sample_1000)

MLE_1000 <- c()
diff_1000 <- c()

for(type in 1:9) {
  MLE_1000[type] <- quantile(sample_1000, 0.5, type = type)
  diff_1000[type] <- MLE_1000[type] - real_median_1000
}

diff_1000
```

```
## [1] -0.0001569595  0.0000000000 -0.0001569595 -0.0001569595  0.0000000000
## [6]  0.0000000000  0.0000000000  0.0000000000  0.0000000000
```

For 1000 observations the best predictors are 2, 5, 6-9 types of fn quantile, similar as for 20 observations. It is noticable that the difference for other types of quantile fn is much smaller than for 20 observations, which means that increasing the number of observations estimator gets more confident and shows better results Looks like we just proved the law of large numbers :)

c. MLE function

```
mle_laplace <- function(data) {
  log_lik_laplace <- function(mu, data) {
    return(sum(abs(data - mu))) # from calculations in a
  }
  result <- optimise(
    log_lik_laplace,
    interval = c(min(data), max(data)),
```

```
    data = data)
  return(result$minimum)
}

mle_laplace(sample_20)
```

```
## [1] 1.066128
```

```
mean(MLE_20)
```

```
## [1] 1.063047
```

```
mle_laplace(sample_1000)
```

```
## [1] 1.028546
```

```
mean(MLE_1000)
```

```
## [1] 1.028591
```

Function 'optimise' uses a combination of golden section search and successive parabolic interpolation to find the minimum or maximum in the selected interval. Golden section search uses golden ratio to narrow the range of values that potentially can be the extremums, while successive parabolic interpolation fits a quadratic function through three points, then the vertex is used for new fitting and so on.

The Newton-Raphson method is not suitable for Laplace function, as it requires continuously differentable function, but Laplace one is not smooth at the point $/mu = X$