MTH2006 - Coursework 6

1 Mosquitoes and the effectiveness of insecticides

We are analysing a study that was conducted to explore the effectiveness of different types and doses of insecticide on mosquitoes. The data from this is seen as `'mosquito'` in the code (seen in Appendix A) and this data set tracks the outcome (survival time of the mosquitoes in tens of hours) and includes the variables `'insecticide'` (denoting the type of insecticide), `'dose'` (denoting the size of the dose administered) and `'block'` (denoting the block). We consider a low dose of insecticide A the control group and henceforth make all comparisons relative to this.

The experiment has a sample size of $n = 48$, and utilises 3 levels of dosage ('low', 'medium', 'high') with 4 types of insecticide ('A', 'B', 'C', 'D'). The formula for replicates is given by

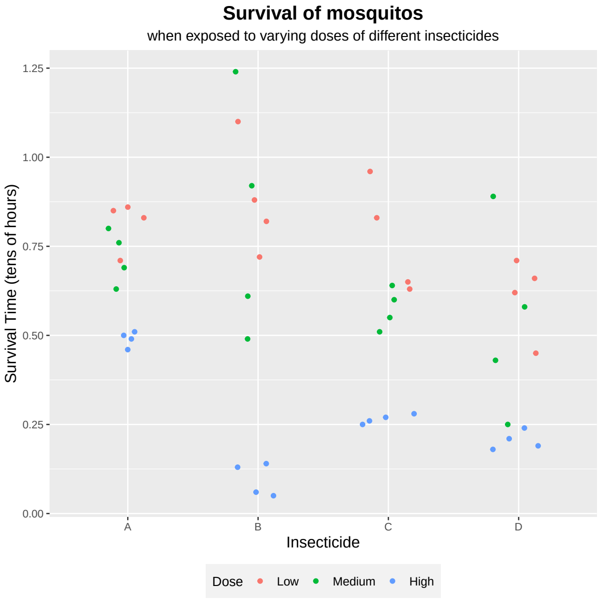
$$\mathrm{N}^0 \text{ of replicates} = \frac{48}{3 \cdot 4} = 4,$$

hence, we have 4 replicates in this experiment.

1.1 The Data

We must first do some housekeeping to make the data easier to use. First, we change the variable, `'dose'`, such that is becomes a factor with levels 'low', 'medium' and 'high'. Secondly, we then rearrange the data set such that it is now ordered by dose level. It is now examine possible to examine the data in a more insightful fashion. We plot the data, seen in Figure 1a in order to establish if there is any discernible relationship between survival time, dosage and insecticide type.

We are wary to derive too much meaning from this small sample size, however, even with $n = 48$, we can see that there is some semblance of a relationship formed between dosage, type of insecticide and survival time. Clearly, perhaps as it to be expected, we see that a high dosage is the most effective in shortening survival time.



(a) The survival time of the mosquitoes is heavily dependent on the dosage and insecticide that is administered

We can also see, in this sample, that insecticide A performs the worse at higher dosage than any other insecticide, Insecticide B performs the best, while C and D are broadly similar. Also of note is the significant spread of data at the low and medium doses of insecticides B, C, D, and to a lesser extent, A.

From this, we can hypothesize that our eventual linear model should show that a high dose of insecticide B should be the most effective, though we cannot rely on this hypothesis to be accurate.

1.2 Analysis of Variance and Linear models

We will now fit analysis of variance models to our data in order to establish the best linear model to use. In the case of this experiment we use treatment contrast compared to a low dose of insecticide A as this is an obvious and simple reference category to use. We model the the survival time of mosquitoes as a function of `dose`, `interaction` and an interaction between the two. We first use For analysis of variance models, where we continue to add variables to the model until any additions are not significant. The p-values given by R's inbuilt *aov()* function are seen in Table 2. We also examine the adjusted R-squared from the *lm()* function. This value essentially measures how well the data is modelled while penalising over-complicated models, these can also be seen on Table 2.

Model - `aov(hours10` ∼)	Variable	p-value	Significant?
insecticide	-	2.6243e-01	No
dose	-	9.7177e-10	Yes
dose: *insecticide*	-	4.9604e-09	Yes
dose + insecticide	*dose*	1.5474e-10	Yes
	insecticide	1.5980e-02	Yes
dose + insecticide + dose : insecticide	*dose*	3.3289e-11	Yes
	insecticide	6.3681e-03	Yes
	dose:insecticide	2.6083e-02	Yes

Table 2: Analysis of variance conducted on various combinations of variables, with p-value and whether it is significant at the 5%

Model - `lm(hours10` ∼)	Adj. R-Squared
dose	0.0235
insecticide	0.6510
dose : insecticide	0.7210
dose + insecticide	0.5850
dose + insecticide + dose : insecticide	0.7210

Table 3: Adjusted R-Squared values for various permutations of the linear model

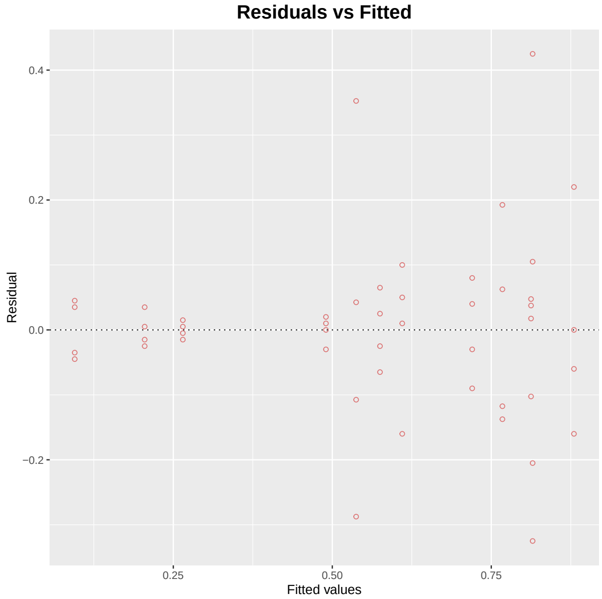
Clearly, Table 2 shows that we have a significant model with both variables and the interaction included. We also see that this permutation of the model has a significantly higher adjusted R-squared, in Table 3, than any other competing model (aside from the interaction alone). Hence, we will model the survival time of mosquitoes using both the dosage and insecticide type as well as an interaction between the two i.e. our model is

`hours10` ∼ `dose + insecticide + dose:insecticide`.

1.3 Final Model

We can further establish the effectiveness of this model by examining how the residuals look plotted against fitted values, seen in Figure 4a. We note there is some banding around the zero line particularly at the smaller fitted values. This could suggest some unaccounted for variables, which in this case either have been insufficiently, or perhaps not at all, been accounted for in the design of the experiment. We also observe the few outliers particularly in the very top and bottom of the far right hand side of the graph.

From this established model we can now examine the coefficients to establish the most effective dosage and insecticide, these coefficients can be seen in Table 5. From Table 5, we can say that a high dosage of Insecticide B is the most effective at shortening the lifespan of mosquitoes as it provides the lowest such value in this model, coming in at 0.85 hours, or 51 minutes.



(a) Residuals plotted against the fitted values for the linear model `lm(hours10 ∼ dose + insecticide + dose:insecticide)`

	Insecticide			
Dose	A	B	C	D
Low	0.8125	+ 0.0675	-0.0450	-0.2025
Medium	- 0.0925	- 0.0925 + 0.0675 + 0.0275	- 0.0925 -0.0450 - 0.1000	- 0.0925 - 0.2025 + 0.0200
High	- 0.3225	- 0.3225 + 0.0675 - 0.4625	- 0.3225 - 0.0450 - 0.1800	-0.3225 - 0.2025 - 0.0825

Table 5: Expected survival times of mosquitoes in tens of hours by insecticide and dosage, as predicted by our model - values are relative to low dose of insecticide A

2 Rainfall in Seavington

2.1 Parametric Model

2.1.1 CDF and Inverse CDF

We are examining rainfall data from Seavington, Somerset in 2020. Firstly, we must modify the data set to exclude all zero values. After doing this, we model these rainfall amounts by assuming they are independent, identically distributed variables with cumulative distribution function

$$F_Y(y;\alpha,\beta) = \begin{cases} 1 - \exp\left[-(\beta y)^{\alpha}\right] & \text{for } y > 0, \\ 0 & \text{otherwise,} \end{cases}$$

where $\alpha, \beta > 0$. We fit the CDF with maximum likelihood estimates, $\hat{alpha} = 0.775$ and $\hat{\beta} = 0.272$. The CDF function, as defined in `R`, is shown below, and will take our MLE estimates and values for y its input.

```
cdf <- function(params, y) { # params = c(alpha, beta)
  if (y > 0) {1 - exp( - (params[2] * y)^params[1] ) }
  else { 0 } # if less than or equal to 0 assign value of 0
}
```

We can now find the inverse CDF. For this case, we relabel $F_Y(y)$ as y and y as x and aim to rearrange for x in terms of y. For the purposes of simplicity, we only consider the case $y > 0$, as it is trivial that the inverse CDF will be the same as the CDF, in that it will give 0 for $y < 0$. Hence, we have

$$\begin{aligned} x &= 1 - \exp\left[-(\beta y)^\alpha\right], \\ -\log(1-x) &= (\beta y)^\alpha, \\ y &= \frac{(-\log(1-x))^{1/\alpha}}{\beta}. \end{aligned}$$

Therefore, by relabelling and including the case $y < 0$, we obtain

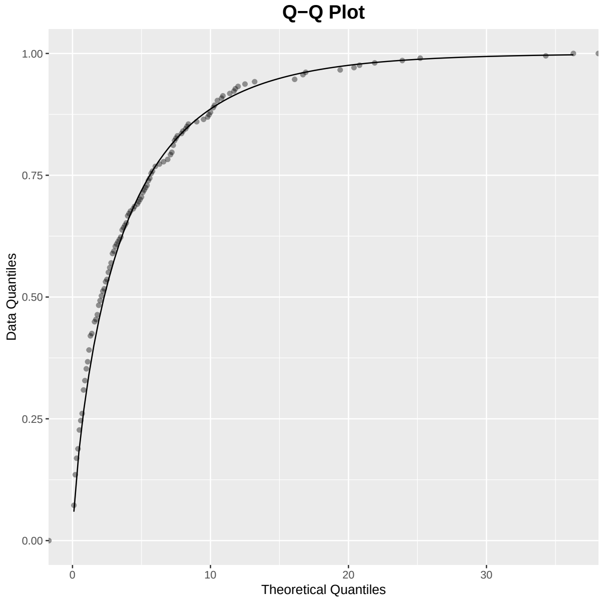
$$F_Y^{-1}(y) = \begin{cases} \frac{(-\log(1-y))^{1/\alpha}}{\beta} & \text{for } y > 0, \\ 0 & \text{otherwise.} \end{cases}$$

Below we can see the inverse CDF defined as a function in R.

```
inv_cdf <- function(params, y){ params = c(alpha, beta)
  if (y > 0) {
    ( ( - log( 1 - y ) ) ^(1/params[1]) / params[2]) }
  else { 0 } # if less than or equal to 0 assign value of 0
}
```

2.1.2 Q-Q Plot

Figure 6a on the right shows a QQ-plot of our CDF and data. We observe a almost remarkable level of tracking between our data and samples drawn from our CDF. There is no significant skewness at either tail. We note there this is some difference on the left-hand side but this is relatively small small. With such a high level of similarity between the quantiles we can draw the conclusion that the MLEs, $\hat{a\alpha}$ and $\hat{\beta}$, give a good model for the non-zero rainfall amounts.



(a) Comparing quantiles from our sample distribution to quantiles from the data

2.1.3 Chi-squared test

We can use Pearson's chi-squared test to assess whether $F_Y(y; \hat{\alpha}, \hat{\beta})$ is a good model for the non-zero rainfall amounts. We use bins of size 5 for values between 0 and 40, and a bin of $[40, \infty]$.Our hypotheses are

H_0 : The sample is in the distribution $F_Y(y; \hat{\alpha}, \hat{\beta})$,

H_1 : The sample is not in the distribution $F_Y(y; \hat{\alpha}, \hat{\beta})$.

In order to calculate the test statistic we must first calculate (1) the number of values observed in each bin in the sample and, (2) the number of values we could expect to observe in each bin, based on our model.

We find the expected values by calculating the difference in the value of CDF between the boundaries of the bins, and multiplying each of these values by the sample size, n. The observed and expected values can be seen in Table 7.

Bins	$(0, 5)$	$[5, 10)$	$[10, 15)$	$[15, 20)$	$[20, 25)$	$[25, 30)$	$[30, 35)$	$[35, 40)$	$[40, \infty)$
Expected	149	35	13	6	3	1	1	0	0
Observed	146	36	13	5	4	1	1	1	0

Table 7: Number of values observed each in bin in the sample versus the expected number based on the distribution F_Y

Our test statistic is then defined by

$$\sum_{i=1}^{9} \frac{(0_i - E_i)^2}{E_i},$$

where O_i and E_i are the observed and expected values, respectively, for the ith bin, if the bins are sequentially labelled 1 to 9. Hence, our test statistic is 3.1810. Our degrees of freedom is given by

$$\textit{number of bins} - \textit{ number of parameters in the model } - 1.$$

Hence, we have the degree of freedom to be $9 - 2 - 1 = 6$.

Now, inputting the test statistic and degrees of freedom into the `pchisq()` function in `R` , we obtain a p-value of 0.7858. Testing at the 5% level we find that there is not significant evidence to reject H_0. Hence, we can say the sample is from the distribution $F_Y(y; \hat{\alpha}, \hat{\beta})$. Therefore, we have a good model for the sample, which is also the conclusion we reached when examining our Q-Q plot. Hence, we have a fairly strong argument that our model is a good fit for the sample.

2.2 Non-parametric model

We will consider a kernel density estimate, $\hat{f}_h(y)$, for a sample of data $y_1, \ldots, y_n$. We will create a kernel density estimate on square-root-transformed rainfall amounts, but will produce a PDF estimate on the original scale of the amounts. We use the following estimate

$$\hat{f}_h^{\text{sqrt}}(y) = \frac{1}{nh} \sum_{i=1}^{n} \frac{w(|\sqrt{y} - \sqrt{y_i}|/h)}{2\sqrt{y}},$$

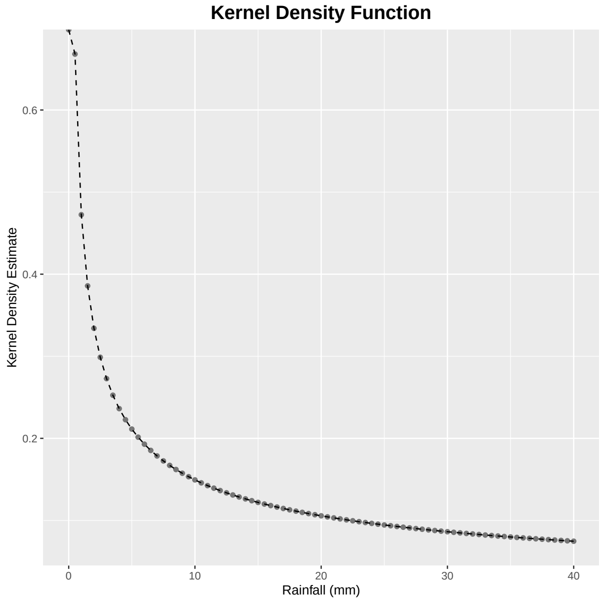
using the square-root-transformed sample of the data to produce our PDF, where h is our bandwidth, y_i is our data and $w()$ is our kernel function. We create this as a function in `R` which can be seen below.

```
# Takes input of y-value, data points, the specified distribution
# and the bandwidth
kde_sqr <- function(y, y_i, distribution, h) {
  x = 0 # Initialising  x
  n = length(y_i) # specifying n
  for (i in 0:n) { # looping and summing for all points of data

    x <- x + ( density( ( sqrt(y) - sqrt(y_i[i]) ) / h, kernel = distribution,
                        bw = h) ) / (2 * sqrt(y))
    # density function calculated for sqrt data and y ,for distribution before
    # being appropriately divided
  }
  fy_sqr = x / ( n * h ) # dividing data by nh and returning
  return(fy_sqr)
}
```

We now take our function and use it to create a graph of our KDE estimate of the PDF. We use a bandwidth of 0.3 with a normal kernel function. We note how the function looks a similar shape to a graph of e^{-x}, and hence it bares similarity to the PDF that would be generated by our CDF function, $F_Y y; \alpha, \beta$.

We evaluate $\hat{f}_h^{\text{sqrt}}(y)$ at a series of points for a normal kernel function and $h = 0.3$. This can be seen in Table 9.



(a) Kernel density of estimate of our probability density function - for $h = 0.3$ with a normal kernel distribution

$y =$	0.5	1	2	5	10
$\hat{f}_h^{\text{sqrt}}(y) =$	0.6681159	0.4724293	0.3340579	0.2112768	0.1493953

Table 9: `kde_sqr(y)` evaluated at various values of y

A Mosquitoes code

```r
# ---- MTH2006 - Coursework 2 - Q1 ----
rm(list = ls())
set.seed(110322)

mosquito <- read.csv('mosquito2021.csv')
alpha = 0.05 # setting significance level

# ---- 1(a) - Relabeling ----
# Modifying data set to treat dose levels as factors and
# reordering
mosquito <- mutate(mosquito, dose = factor(dose, levels =
                                          c('low', 'medium', 'high'))) %>%
            arrange(match(dose, c('low', 'medium', 'high' )))
head(mosquito)

# ---- 1(b) - Plotting data ----
mosq.plot <- ggplot(mosquito, aes(x = insecticide, y = hours10)) +
  geom_jitter(width = 0.2, height = 0,  aes(colour = dose))

# Formatting and saving plot
format <- theme(plot.title = element_text(hjust = 0.5 ,face = "bold", size = 16),
                plot.subtitle =  element_text(hjust = 0.5, size = 12),
                axis.title = element_text(size = 13),
                legend.position = 'bottom',
                legend.background = element_rect(fill = 'grey95'),
                legend.text = element_text(size = 10)) +
          scale_colour_discrete(labels = c('Low', 'Medium', 'High'))
mosq.plot <- mosq.plot +
                  labs(x = 'Insecticide', y = 'Survival Time (tens of hours)',
                       colour = 'Dose', title = 'Survival of mosquitos',
                       subtitle = 'when exposed to varying
                       doses of different insecticides') +
                  format
pdf('mosq_plot.pdf')
  mosq.plot
dev.off()

# ---- 1(c) - Replicates and Sample sizes ----
n <- nrow(mosquito) # sample size
n.dose <- length(unique(mosquito$dose)) # number of doses
n.insect <- length(unique(mosquito$insecticide)) # number of insecticides
repl <- n / (n.dose * n.insect) # number of replicates


# ---- 1(d) -  Fit ANOVA and linear models ----
# Fitting ANOVA models for all possible models
aov.both <- summary(aov(hours10 ~ dose + insecticide, mosquito))
aov.dose <- summary(aov(hours10 ~ dose, mosquito))
aov.insect <-summary(aov(hours10 ~ insecticide, mosquito))
aov.int <- summary(aov(hours10 ~ dose:insecticide, mosquito))
aov.all <- summary(aov(hours10 ~ dose + insecticide + dose:insecticide, mosquito))

# Fitting possible linear models
lm.both <- summary(lm(hours10 ~ dose + insecticide, mosquito))
lm.dose <- summary(lm(hours10 ~ dose, mosquito))
lm.insect <- summary(lm(hours10 ~ insecticide, mosquito))
lm.int <- summary(lm(hours10 ~ dose:insecticide, mosquito))
lm.all <- summary(lm(hours10 ~ dose + insecticide + dose:insecticide, mosquito))

# Isolating p-values
p.vals <- c( aov.dose[[1]][1,5], aov.insect[[1]][1,5], aov.int[[1]][1,5],
             aov.both[[1]][1,5], aov.both[[1]][2,5],
             aov.all[[1]][1,5], aov.all[[1]][2,5], aov.all[[1]][3,5])

# Doing significance test on all models
sig = NULL
for (i in 0:length(p.vals)){
  sig[i] = ( p.vals[i] < alpha )
}
# p-values table
p.tab <- tibble('Model' = c('Dose', 'Insecticide', 'Interaction',
                            'Both-Dose', 'Both-Insect',
                            'All - Dose', 'All - Insecticide',
                            'All - Interaction'),
                'P-Values' = p.vals, 'Significant?' = sig)
print(p.tab)

# Isolating adjusted R vals
adjR.vals <- c( lm.dose$adj.r.squared, lm.insect$adj.r.squared,
                lm.both$adj.r.squared,
                lm.int$adj.r.squared, lm.all$adj.r.squared)

# Adjusted R-squared table
adjR.tab <- tibble('Model' = c('Both', 'Dose', 'Insecticide',
                               'Interaction', 'All'),
                   'Adj R Values' =  adjR.vals)
print(adjR.tab)


# Fitting best model
model <- lm(hours10 ~ dose + insecticide + dose:insecticide, mosquito)

# ---- 1(e) - Residual Plot ----
model.resid <- tibble('Fitted' = fitted(model), 'Residual' = resid(model))

# Plotting, formatting and saving residual plot
resid.plot <- ggplot(model.resid) +
                geom_point(aes(x =  Fitted, y = Residual),
                          alpha = 0.5, fill = 'white',
                          colour = 'red3', shape = 1) +
                geom_hline(aes(yintercept = 0), alpha = 0.8, lty = 'dotted') +
                labs(x = 'Fitted values',
                     title = 'Residuals vs Fitted') +
               theme(plot.title = element_text(hjust = 0.5 ,
                                               face = "bold", size = 16))
pdf('resid_plot.pdf')
  resid.plot
dev.off()

# ----  1(f) - Best insecticide ----
coef <- coef(summary(model))
```

B Rainfall code

```
# ---- MTH2006 - Coursework 2 - Q2 ----
rm(list = ls())
set.seed(110322)
setwd("~/OneDrive - University of Exeter/RStudio/Coursework 2")

# ---- 2(a) - Models ----
# Modifying data for values greater than 0
rainfall <- read.csv('seavington2020.csv')
rainfall <- filter(rainfall, prcp > 0)
prcp <- rainfall$prcp


params = c(0.775, 0.272) # setting MLE parameters

# Calling functions of cdf and inverse cdf
cdf <- function(params, y) {
  if (y > 0) {1 - exp( - (params[2] * y)^params[1] ) }
  else { 0 }
}
inv_cdf <- function(params, y){
  if (y > 0) {
    ( ( - log( 1 - y ) ) ^(1/params[1]) / params[2]) }
  else { 0 }
}

fy <- cdf(params, prcp)
inv.fy <- inv_cdf(params, fy)

# Plotting, formatting, saving Q-Q plot
qqplot <- ggplot(rainfall, aes(x = prcp)) +
                stat_ecdf(geom = 'point', alpha = 0.4) +
                stat_function(fun = cdf , args = list(params = params))
qqplot <- qqplot + labs(x = 'Theoretical Quantiles', y = 'Data Quantiles',
                        title = 'Q-Q Plot') +
                   theme(plot.title =
                           element_text(hjust = 0.5 , face = "bold", size = 16))
pdf('qq_plot.pdf')
  qqplot
dev.off()

# ---- 2(b)  - Hypothesis test for chi-squared ----
# H_0 - Sample is in the distribution
# H_1 - Sample is not in the distribution

bins <- seq(0,40,5) # setting desired bins
bins[length(bins)+1] <- Inf # modifying right hand boundary to be infinity
obs <- table(cut(prcp, bins)) # finding number of values in each bin from data

F_Y = NULL # calculating cdf for each bin boundary
for (i in 1:length(bins)){
  F_Y[i] <- cdf(params, bins[i])}

# Calculating expected numbers for each bin
expec <- nrow(rainfall) * diff(F_Y)

# Comparison of expected vs observed values
comp <- tibble('Expected' = round(expec), 'Obeserved' = obs)

# Conducting hypothesis test
test.stat <- sum((obs - expec)^2 / expec) # test statistic
d.f <- length(expec) - length(params) - 1 # degrees of freedom (J - p - 1),
pval <- 1 - pchisq(test.stat, d.f) # p-value calculation

# Not significant at the 5% level - no significant evidence to reject H0

# ---- 2(c) - Kernel Density Estimate ----
# Takes input of y-value, data points, the specified distribution
# and the bandwidth
```

```r
kde_sqr <- function(y, y_i, distribution, h) {
  x = 0 # Initialising  x
  n = length(y_i) # specifying n
  for (i in 0:n) { # looping and summing for all points of data

    x <- x + ( density( ( sqrt(y) - sqrt(y_i[i]) ) / h, kernel = distribution,
                        bw = h) ) / (2 * sqrt(y))
  }
  fy_sqr = x / ( n * h ) # dividing data by nh and returning
  return(fy_sqr)
}
# KDE for points for a plot
points <- seq(0,40,0.5)
kde.points <- kde_sqr(points, prcp, 'gaussian', 0.3)
kde.tab <- tibble('Points' = points, 'Est' = kde.points)

# Plotting points
kde.plot <- ggplot(kde.tab, aes(x = points, y = kde.points)) +
                geom_point(alpha = 0.3) +
                geom_line(lty = 2) +
                labs(x = 'Rainfall (mm)', y = 'Kernel Density Estimate')

# Evaluating set values
vals2 <- c(0.5, 1, 2, 5, 10)
kde.vals2 <- kde_sqr(vals2, prcp, 'gaussian', 0.3)
kde.tab2 <- tibble('y' = vals2, 'KDE' = kde.vals2)
```