

# Presentation

Quadrant count ratio (QCR):

```
qcr <- function(X, Y) {  
  r_Q <- 0  
  med_x <- median(X)  
  med_y <- median(Y)  
  for (i in 1:length(X)) {  
    r_Q <- r_Q + sign(X[[i]] - med_x) * sign(Y[[i]] - med_y)  
  }  
  r_Q <- r_Q/length(X)  
}
```

Least absolute deviation (LAD):

```
lad <- function(X, Y) {  
  med_x <- median(X)  
  med_y <- median(Y)  
  r_Q <- qcr(X, Y)  
  k_Q_20 <- 1.491  
  q_x <- IQR(x)  
  q_y <- IQR(Y)  
  beta_1 <- r_Q * q_y/q_x  
  beta_0 <- med_y - beta_1 * med_x  
  beta <- list(beta_1 = beta_1, beta_0 = beta_0)  
}
```

```
x <- seq(-1.8, 2, by = 0.2)  
e <- rnorm(length(x))  
y <- 2 + 2 * x + e  
dta <- list(x = x, y = y)  
lin_mod <- lm(y ~ x, data = dta)  
lad_coeffs <- lad(x, y)  
p <- ggplot(data = data.frame(dta), aes(x = x, y = y, colour = Legend)) + geom_point(aes(colour = "Sample")) +  
  geom_segment(aes(x = -1.8, xend = 2, y = -1.6, yend = 6, colour = "Model"), size = 1.05) +  
  stat_smooth(se = F, method = "lm", aes(colour = "LS")) + xlim(-1.8, 2) + geom_segment(aes(x = -1.8,  
  xend = 2, y = lad_coeffs$beta_1 * (-1.8) + lad_coeffs$beta_0, yend = lad_coeffs$beta_1 *  
  2 + lad_coeffs$beta_0, colour = "LAD", size = 1.05) + scale_colour_manual(values = c("blue",  
  "red", "orangered4", "black"), guide = guide_legend(override.aes = list(linetype = c(rep("solid",  
  3), "blank"), shape = c(rep(NA, 3), 16))))  
ggsave("resources\\usual_sample_regression.pdf", p, device = "pdf")
```

## Saving 6.5 x 4.5 in image

## `geom\_smooth()` using formula 'y ~ x'

```
us_LS_coeffs <- paste("\\beta_0 \\approx ", toString(round(lin_mod$coefficients[1],  
  digits = 2)), "\\;\\;\\beta_1 \\approx ", toString(round(lin_mod$coefficients[2],  
  digits = 2)), sep = "")  
file_handler <- file("resources\\us_LS_coeffs.tex")
```

```

writeLines(us_LS_coeffs, file_handler)
close(file_handler)
us_LAD_coeffs <- paste("\\beta_{OR} \\approx ", toString(round(lad_coeffs$beta_0,
  digits = 2)), "\\;\\;\\beta_{1R} \\approx ", toString(round(lad_coeffs$beta_1,
  digits = 2)), sep = "")
file_handler <- file("resources\\us_LAD_coeffs.tex")
writeLines(us_LAD_coeffs, file_handler)
close(file_handler)

x <- seq(-1.8, 2, by = 0.2)
e <- rnorm(length(x))
y <- 2 + 2 * x + e
y[1] = y[1] + 10
y[20] = y[20] - 10
dta <- list(x = x, y = y)
lin_mod <- lm(y ~ x, data = dta)
lad_coeffs <- lad(x, y)
p <- ggplot(data = data.frame(dta), aes(x = x, y = y, colour = Legend)) + geom_point(aes(colour = "Sample",
  size = 10)) + geom_segment(aes(x = -1.8, xend = 2, y = -1.6, yend = 6, colour = "Model"), size = 1.05) +
  stat_smooth(se = F, method = "lm", aes(colour = "LS")) + xlim(-1.8, 2) + geom_segment(aes(x = -1.8,
  xend = 2, y = lad_coeffs$beta_1 * (-1.8) + lad_coeffs$beta_0, yend = lad_coeffs$beta_1 *
  2 + lad_coeffs$beta_0, colour = "LAD"), size = 1.05) + scale_colour_manual(values = c("blue",
  "red", "orangered4", "black"), guide = guide_legend(override.aes = list(linetype = c(rep("solid",
  3), "blank"), shape = c(rep(NA, 3), 16))))
ggsave("resources\\perturbated_sample_regression.pdf", p, device = "pdf")

## Saving 6.5 x 4.5 in image
## `geom_smooth()` using formula 'y ~ x'

pert_LS_coeffs <- paste("\\beta_0 \\approx ", toString(round(lin_mod$coefficients[1],
  digits = 2)), "\\;\\;\\beta_1 \\approx ", toString(round(lin_mod$coefficients[2],
  digits = 2)), sep = "")
file_handler <- file("resources\\pert_LS_coeffs.tex")
writeLines(pert_LS_coeffs, file_handler)
close(file_handler)
pert_LAD_coeffs <- paste("\\beta_{OR} \\approx ", toString(round(lad_coeffs$beta_0,
  digits = 2)), "\\;\\;\\beta_{1R} \\approx ", toString(round(lad_coeffs$beta_1,
  digits = 2)), sep = "")
file_handler <- file("resources\\pert_LAD_coeffs.tex")
writeLines(pert_LAD_coeffs, file_handler)
close(file_handler)

```