Reproducing Markdown

2024-05-28

Setup

Loading Libraries

```
library(foreign)
library(quantreg)

## Loading required package: SparseM

##
## Attaching package: 'SparseM'

## The following object is masked from 'package:base':
##
##     backsolve

library(ggplot2)
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

Setting Working Directory and Loading Data

```
setwd("~/Desktop/Misc Projects/Happiness")
# set up working directory, you will need to change into your own.
dat <- read.dta("2012_hse_and_shes_combined.dta") # load the data
subd <- subset(dat, eqvinc > 0 & wemwbs > 0 & bmival > 0)
```

Median Regression

```
# Count observations above and below eqvinc = 50,000
subd$income_group <- ifelse(subd$eqvinc > 50000, "Above £50k", "Below £50k")
table(subd$income_group)
```

```
##
## Above £50k Below £50k
##       1332       5862

# Changing the Reference Category
subd$income_group <- as.factor(subd$income_group)
subd$income_group <- relevel(subd$income_group, ref = "Below £50k")

# Global Model with Interactions
model_basic <- rq(formula = wemwbs ~ log(eqvinc) * income_group, tau = 0.5
,data = subd)
model_basic

## Call:
## rq(formula = wemwbs ~ log(eqvinc) * income_group, tau = 0.5,
##     data = subd)
##
## Coefficients:
##                          (Intercept)                          log(eqvinc)
##                            26.531388                            2.546013
##              income_groupAbove £50k log(eqvinc):income_groupAbove £50k
##                            11.740475                           -1.130463
##
## Degrees of freedom: 7194 total; 7190 residual

summary(model_basic)

##
## Call: rq(formula = wemwbs ~ log(eqvinc) * income_group, tau = 0.5,
##     data = subd)
##
## tau: [1] 0.5
##
## Coefficients:
##                                    Value    Std. Error t value  Pr(>|t|)
## (Intercept)                        26.53139  2.46409   10.76722  0.00000
## log(eqvinc)                         2.54601  0.24437   10.41880  0.00000
## income_groupAbove £50k             11.74048  6.58100    1.78400  0.07447
## log(eqvinc):income_groupAbove £50k -1.13046  0.58955   -1.91751  0.05521

# difference in medians of happiness at household incomes of £15,000 and
£250,000

wellbeing_high <- 26.53139 + 11.74048 +( 2.54601 -1.13046 )*log(250000)
wellbeing_low <- 26.53139+2.54601*log(15000)

wellbeing_high - wellbeing_low

## [1] 4.85272
```

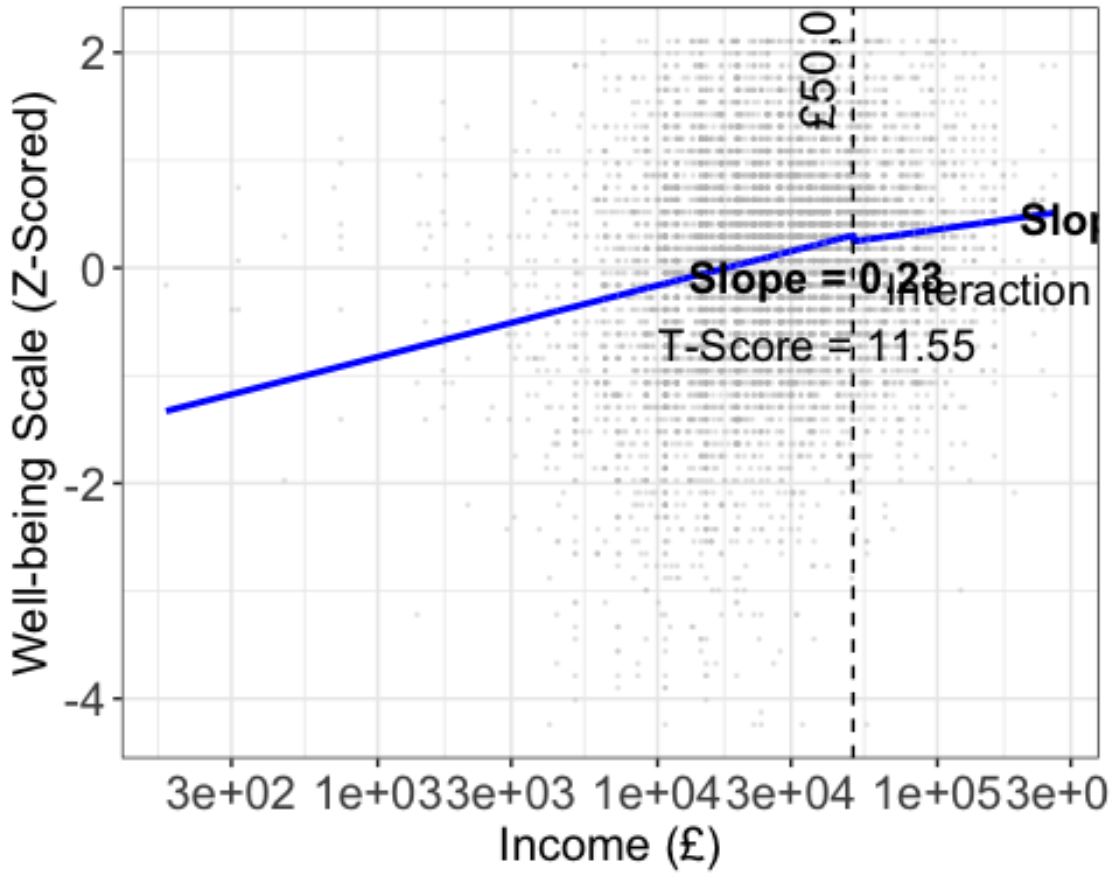
Getting Standardised Coefficients

```r
# Standardising the Data

data_standardised <- subd %>%
  mutate(
    wemwbs_scaled = scale(wemwbs),
    log_eqvinc_scaled = scale(log(eqvinc))
  )

model_standardised <- rq(formula = wemwbs_scaled ~ log_eqvinc_scaled *
income_group, tau = 0.5, data = data_standardised)
summary(model_standardised)

##
## Call: rq(formula = wemwbs_scaled ~ log_eqvinc_scaled * income_group,
##     tau = 0.5, data = data_standardised)
##
## tau: [1] 0.5
##
## Coefficients:
##                                          Value    Std. Error t value
Pr(>|t|)
## (Intercept)                               0.10562  0.01497    7.05527
0.00000
## log_eqvinc_scaled                         0.22813  0.02190   10.41880
0.00000
## income_groupAbove £50k                    0.03125  0.07983    0.39147
0.69546
## log_eqvinc_scaled:income_groupAbove £50k -0.10129  0.05283   -1.91751
0.05521

data_standardised$predicted <- predict(model_standardised, data =
data_standardised)
```

Getting Coefficients and Plotting

```r
# Dataframe for metrics
slope_rich <- 0.22813 -0.10129
intercept_rich <- 0.10562 + 0.03125

# Dataframe for metrics
metrics_low_standardized <- data.frame(
  slope = 0.22813,
  intercept = 0.10562,
  slope_t_value = 11.54772
)

# Dataframe for standardized coefficients - High Income Group
metrics_high_standardized <- data.frame(
  slope = slope_rich,
  intercept = intercept_rich,
```

```r
  interaction_t_value = -1.91751
)

subd <- subd %>%
  mutate(predicted = predict(model_basic, newdata = subd))

adjust <- (-0.5)

# Create the plot
plot1 <- ggplot(data_standardised, aes(x = eqvinc, y = wemwbs_scaled)) +
  geom_point(size = 0.5, shape = 16, color = "grey", alpha = 0.4) +
  geom_line(aes(y = predicted), size = 1, color = "blue") +
  geom_vline(xintercept = 50000, linetype = "dashed", color = "black") +
  annotate("text", x = 50000, y = max(data_standardised$wemwbs_scaled), label
= "£50,000", color = "black", angle = 90, vjust = -0.5, size = 5.5) +
  annotate("text", x = 10000, y = metrics_low_standardized$intercept +
metrics_low_standardized$slope * log(1) + adjust,
           label = paste0('atop(bold("Slope = ',
round(metrics_low_standardized$slope, 2),
                          '"), "\nT-Score = ',
round(metrics_low_standardized$slope_t_value, 2), '")'),
           color = "black", size = 5, hjust = 0, parse = TRUE) +
  annotate("text", x = 65000, y = metrics_high_standardized$intercept +
metrics_high_standardized$slope * log(1),
           label = paste0('atop(bold("Slope = ',
round(metrics_high_standardized$slope, 2),
                          '"), "\nInteraction T-Score = ',
round(metrics_high_standardized$interaction_t_value, 2), '")'),
           color = "black", size = 5, hjust = 0, parse = TRUE) +
  theme_bw() +
  labs(x = "Income (£)", y = "Well-being Scale (Z-Scored)") +
  scale_x_continuous(trans = "log10", breaks = scales::log_breaks(n = 8)) +
  theme(axis.title.x = element_text(size = 15),
        axis.title.y = element_text(size = 15),
        axis.text = element_text(size = 15),
        legend.position = "none")  # Remove legend

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

# Display the plot
print(plot1)
```



Percentile Regression

First, with the same percentiles as KK (2022).

```
# Quantile Plots at Different Percentiles
taus<-c(0.15, 0.3, 0.5, 0.7, 0.85)

models  <- rq(formula = wemwbs ~ log(eqvinc) * income_group,tau = taus,
                 data = subd)

# t-values
slope_t_values <- lapply(summary(models), function(x) x$coefficients[, "t
value"])
slope_t_values_df <- do.call(rbind, slope_t_values)

# t-values for slope
# Create the logical indicator for "Above £50k"
subd$below_50k <- as.integer(subd$income_group != "Above £50k")
table(subd$below_50k)
```

```
##
##    0    1
## 1332 5862
```

```
models_b <- rq(wemwbs ~ log(eqvinc) * below_50k, tau = taus, data = subd)
slope_above_t_values <- lapply(summary(models_b), function(x)
x$coefficients[, "t value"])
slope_above_t_values_df <- do.call(rbind, slope_above_t_values)
```

This is the easiest way I found of getting the t-values for the slope above £50,000.

```
# We want the coefficients on log(eqvinc) and the interaction
metrics_low <- data.frame(
  intercept = coef(models)[1, ],
  slope_coef = coef(models)[2, ],
  slope_t_values = slope_t_values_df[,2],
  income_group = "Below £50k")

metrics_high <- data.frame(
  intercept = coef(models)[1, ]+coef(models)[3, ],
  slope_coef = coef(models)[2, ]+coef(models)[4, ],
  slope_above_t_values = slope_above_t_values_df[,2],
  income_group = "Above £50k")

metrics_low
```

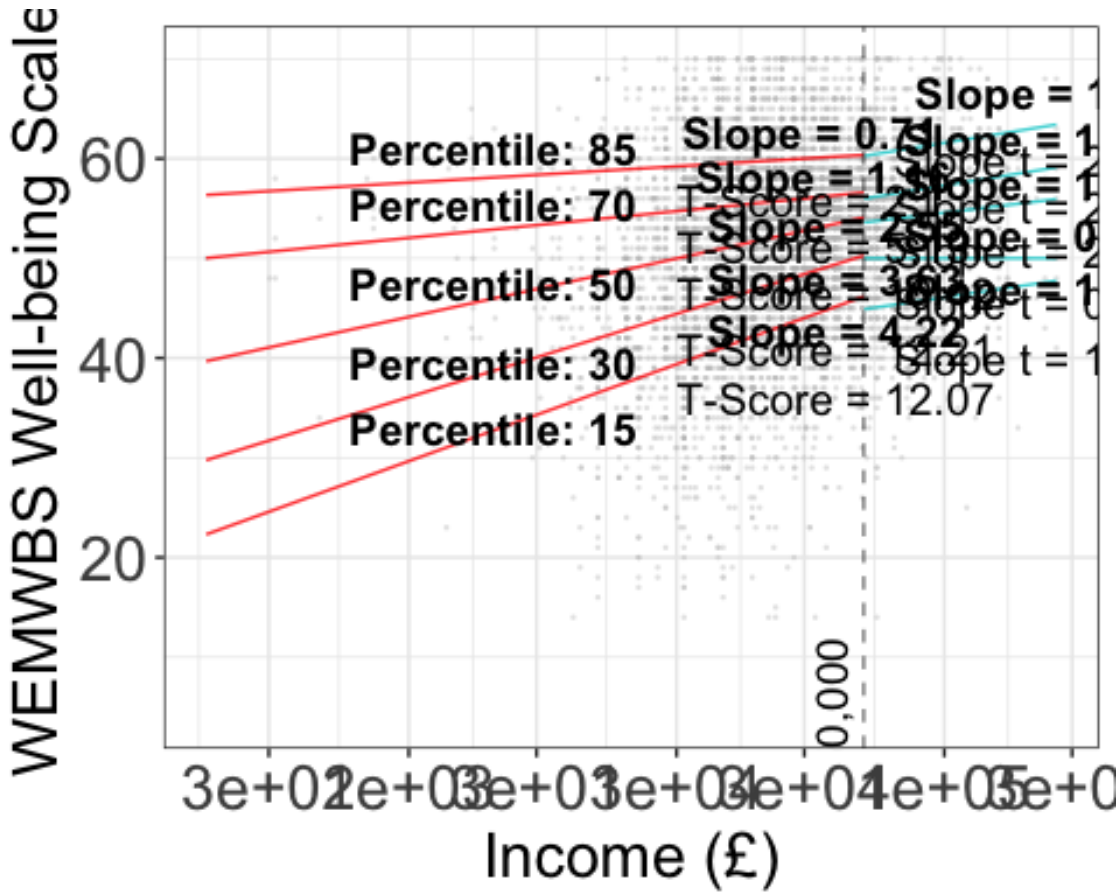
```
##            intercept slope_coef slope_t_values income_group
## tau= 0.15  0.5151016  4.2195350      12.065510   Below £50k
## tau= 0.30 11.0259389  3.6268616      12.212213   Below £50k
## tau= 0.50 26.5313875  2.5460127      10.418799   Below £50k
## tau= 0.70 44.0141718  1.1630388       5.126952   Below £50k
## tau= 0.85 52.6938699  0.7089466       2.101577   Below £50k
```

Plotting

```
# Generate plot
offset_high <- c(-2, -1.5, 0, 2, 2.5)

p1 <- ggplot(NULL, aes(x = eqvinc, y = wemwbs, colour = income_group)) +

  # Plot points for income groups
  geom_point(data = subd, size = 0.5, shape = 16, color = "grey", alpha =
0.4) +

  # Add quantile regression lines for "Below £60K" income group
  geom_quantile(data = subset(subd, income_group == "Below £50k"),
                color = "red", alpha = 0.8, quantiles = taus) +

  # Add quantile regression lines for "Above £60K" income group
  geom_quantile(data = subset(subd, income_group == "Above £50k"),
                color = "darkturquoise", alpha = 0.8, quantiles = taus) +
```

```r
  # Label for Income Threshold
    annotate("text", x = 50000, y = max(data_standardised$wemwbs_scaled)+2,
label = "£50,000", color = "black", angle = 90, vjust = -0.5, size = 5) +

  # Annotations for Below
  annotate("text", x = 10000, y = metrics_low$intercept +
metrics_low$slope_coef * log(10000),
           label = paste0('atop(bold("Slope = ',
round(metrics_low$slope_coef, 2),
                          '"), "\nT-Score = ',
round(metrics_low$slope_t_values, 2), '")'),
           color = "black", size = 5, hjust = 0, parse = TRUE) +

  # Add labels for "Above £50K" coefficients
  annotate("text", x = 65000, y = metrics_high$intercept +
metrics_high$slope_coef * log(55000) + offset_high,
           label = paste0('atop(bold("Slope = ',
round(metrics_high$slope_coef, 2),
                          '"), "\nSlope t = ',
round(metrics_high$slope_above_t_values, 2), '")'),
           color = "black", size = 5, hjust = 0, parse = TRUE) +
  # Labels for Percentiles
  annotate("text", x = 600, y = metrics_low$intercept +
metrics_low$slope_coef*log(1000),
           label = paste0('atop(bold("Percentile: ', taus*100, '"))'),
           color = "black", size = 5, hjust = 0, parse = TRUE) +

  # Add a dashed vertical line at £50,000
  geom_vline(xintercept = 50000, linetype = "dashed", color = "black", alpha
= 0.5) +

  # Apply themes and labels
  theme_bw() +
  labs(x = "Income (£)", y = "WEMWBS Well-being Scale") +
  scale_x_continuous(trans = "log10", breaks = scales::log_breaks(n = 8)) +
  theme(axis.title.x = element_text(size = 20),
        axis.title.y = element_text(size = 20),
        axis.text = element_text(size = 20),
  )

p1
```

```
## Smoothing formula not specified. Using: y ~ x
## Smoothing formula not specified. Using: y ~ x
```



Getting a table

```
# Table
# For Slopes Below £50k, and Interaction Effects
taus0<-c(0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.5, 0.7, 0.85)

abs1  <- rq(formula = wemwbs ~ log(eqvinc) * income_group,tau = taus0,
              data = subd)
summary(abs1)

##
## Call: rq(formula = wemwbs ~ log(eqvinc) * income_group, tau = taus0,
##     data = subd)
##
## tau: [1] 0.05
##
## Coefficients:
##                                  Value     Std. Error t value   Pr(>|t|)
## (Intercept)                      -26.43929   5.69480   -4.64271
0.00000
## log(eqvinc)                        6.30151   0.57019   11.05153
0.00000
## income_groupAbove £50k            49.64921  19.83266    2.50341
```

```
0.01232
## log(eqvinc):income_groupAbove £50k  -4.73344   1.78227   -2.65585
0.00793
##
## Call: rq(formula = wemwbs ~ log(eqvinc) * income_group, tau = taus0,
##     data = subd)
##
## tau: [1] 0.1
##
## Coefficients:
##                                    Value     Std. Error t value   Pr(>|t|)
## (Intercept)                        -11.32407   4.54067   -2.49392
0.01266
## log(eqvinc)                          5.17361   0.44702   11.57362
0.00000
## income_groupAbove £50k              54.32407  16.62295    3.26802
0.00109
## log(eqvinc):income_groupAbove £50k  -5.17361   1.49545   -3.45958
0.00054
##
## Call: rq(formula = wemwbs ~ log(eqvinc) * income_group, tau = taus0,
##     data = subd)
##
## tau: [1] 0.15
##
## Coefficients:
##                                    Value    Std. Error t value  Pr(>|t|)
## (Intercept)                         0.51510  3.56310    0.14457  0.88506
## log(eqvinc)                         4.21954  0.34972   12.06551  0.00000
## income_groupAbove £50k             25.66676 14.18109    1.80993  0.07035
## log(eqvinc):income_groupAbove £50k -2.49045  1.27022   -1.96064  0.04996
##
## Call: rq(formula = wemwbs ~ log(eqvinc) * income_group, tau = taus0,
##     data = subd)
##
## tau: [1] 0.2
##
## Coefficients:
##                                    Value    Std. Error t value  Pr(>|t|)
## (Intercept)                         6.39203  2.50050    2.55630  0.01060
## log(eqvinc)                         3.79639  0.25145   15.09791  0.00000
## income_groupAbove £50k             32.06117  6.60881    4.85128  0.00000
## log(eqvinc):income_groupAbove £50k -3.02516  0.58359   -5.18370  0.00000
##
## Call: rq(formula = wemwbs ~ log(eqvinc) * income_group, tau = taus0,
##     data = subd)
##
## tau: [1] 0.25
##
## Coefficients:
```

```
##                                     Value    Std. Error t value  Pr(>|t|)
## (Intercept)                          6.37860  2.65033    2.40672  0.01612
## log(eqvinc)                          3.94341  0.26645   14.79972  0.00000
## income_groupAbove £50k              29.76992 11.57279    2.57241  0.01012
## log(eqvinc):income_groupAbove £50k -2.83327  1.03606   -2.73466  0.00626
##
## Call: rq(formula = wemwbs ~ log(eqvinc) * income_group, tau = taus0,
##     data = subd)
##
## tau: [1] 0.3
##
## Coefficients:
##                                     Value    Std. Error t value  Pr(>|t|)
## (Intercept)                         11.02594  2.98159    3.69801  0.00022
## log(eqvinc)                          3.62686  0.29699   12.21221  0.00000
## income_groupAbove £50k              38.97406 12.73308    3.06085  0.00222
## log(eqvinc):income_groupAbove £50k -3.62686  1.14678   -3.16264  0.00157
##
## Call: rq(formula = wemwbs ~ log(eqvinc) * income_group, tau = taus0,
##     data = subd)
##
## tau: [1] 0.35
##
## Coefficients:
##                                     Value    Std. Error t value  Pr(>|t|)
## (Intercept)                         14.41792  2.76531    5.21385  0.00000
## log(eqvinc)                          3.41770  0.27509   12.42390  0.00000
## income_groupAbove £50k              23.38379 13.15653    1.77735  0.07555
## log(eqvinc):income_groupAbove £50k -2.21623  1.18663   -1.86766  0.06185
##
## Call: rq(formula = wemwbs ~ log(eqvinc) * income_group, tau = taus0,
##     data = subd)
##
## tau: [1] 0.5
##
## Coefficients:
##                                     Value    Std. Error t value  Pr(>|t|)
## (Intercept)                         26.53139  2.46409   10.76722  0.00000
## log(eqvinc)                          2.54601  0.24437   10.41880  0.00000
## income_groupAbove £50k              11.74048  6.58100    1.78400  0.07447
## log(eqvinc):income_groupAbove £50k -1.13046  0.58955   -1.91751  0.05521
##
## Call: rq(formula = wemwbs ~ log(eqvinc) * income_group, tau = taus0,
##     data = subd)
##
## tau: [1] 0.7
##
## Coefficients:
##                                     Value    Std. Error t value  Pr(>|t|)
## (Intercept)                         44.01417  2.30377   19.10528  0.00000
```

```
## log(eqvinc)                            1.16304  0.22685    5.12695  0.00000
## income_groupAbove £50k               -8.46292 10.61310   -0.79740  0.42524
## log(eqvinc):income_groupAbove £50k  0.72587  0.96054    0.75569  0.44986
##
## Call: rq(formula = wemwbs ~ log(eqvinc) * income_group, tau = taus0,
##     data = subd)
##
## tau: [1] 0.85
##
## Coefficients:
##                                    Value     Std. Error t value   Pr(>|t|)
## (Intercept)                         52.69387   3.36888   15.64137
0.00000
## log(eqvinc)                          0.70895   0.33734    2.10158
0.03563
## income_groupAbove £50k             -13.04785   9.01512   -1.44733
0.14785
## log(eqvinc):income_groupAbove £50k   1.19578   0.79741    1.49958
0.13377
```

Same method as above, but for all the different taus.

```
# For Slopes for Above 60k
# Create the logical indicator for "Above £60k"
subd$below_50k <- as.integer(subd$income_group != "Above £50k")

abs2 <- rq(wemwbs ~ log(eqvinc) * below_50k, tau = taus0, data = subd)
summary(abs2)

##
## Call: rq(formula = wemwbs ~ log(eqvinc) * below_50k, tau = taus0, data =
subd)
##
## tau: [1] 0.05
##
## Coefficients:
##                       Value     Std. Error t value   Pr(>|t|)
## (Intercept)           23.20992  18.99746    1.22174   0.22185
## log(eqvinc)            1.56807   1.68860    0.92862   0.35312
## below_50k            -49.64921  19.83266   -2.50341   0.01232
## log(eqvinc):below_50k  4.73344   1.78227    2.65585   0.00793
##
## Call: rq(formula = wemwbs ~ log(eqvinc) * below_50k, tau = taus0, data =
subd)
##
## tau: [1] 0.1
##
## Coefficients:
##                       Value     Std. Error t value   Pr(>|t|)
## (Intercept)           43.00000  15.99077    2.68905   0.00718
```

```
## log(eqvinc)             0.00000   1.42707    0.00000   1.00000
## below_50k             -54.32407  16.62295   -3.26802   0.00109
## log(eqvinc):below_50k   5.17361   1.49545    3.45958   0.00054
##
## Call: rq(formula = wemwbs ~ log(eqvinc) * below_50k, tau = taus0, data =
subd)
##
## tau: [1] 0.15
##
## Coefficients:
##                        Value     Std. Error t value   Pr(>|t|)
## (Intercept)            26.18186  13.72616    1.90744   0.05650
## log(eqvinc)             1.72909   1.22113    1.41597   0.15683
## below_50k             -25.66676  14.18109   -1.80993   0.07035
## log(eqvinc):below_50k   2.49045   1.27022    1.96064   0.04996
##
## Call: rq(formula = wemwbs ~ log(eqvinc) * below_50k, tau = taus0, data =
subd)
##
## tau: [1] 0.2
##
## Coefficients:
##                        Value     Std. Error t value   Pr(>|t|)
## (Intercept)            38.45320   6.11751    6.28576   0.00000
## log(eqvinc)             0.77122   0.52664    1.46442   0.14312
## below_50k             -32.06117   6.60881   -4.85128   0.00000
## log(eqvinc):below_50k   3.02516   0.58359    5.18370   0.00000
##
## Call: rq(formula = wemwbs ~ log(eqvinc) * below_50k, tau = taus0, data =
subd)
##
## tau: [1] 0.25
##
## Coefficients:
##                        Value     Std. Error t value   Pr(>|t|)
## (Intercept)            36.14852  11.26523    3.20886   0.00134
## log(eqvinc)             1.11014   1.00121    1.10880   0.26755
## below_50k             -29.76992  11.57279   -2.57241   0.01012
## log(eqvinc):below_50k   2.83327   1.03606    2.73466   0.00626
##
## Call: rq(formula = wemwbs ~ log(eqvinc) * below_50k, tau = taus0, data =
subd)
##
## tau: [1] 0.3
##
## Coefficients:
##                        Value     Std. Error t value   Pr(>|t|)
## (Intercept)            50.00000  12.37907    4.03907   0.00005
## log(eqvinc)             0.00000   1.10766    0.00000   1.00000
## below_50k             -38.97406  12.73308   -3.06085   0.00222
```

```
## log(eqvinc):below_50k   3.62686   1.14678    3.16264   0.00157
##
## Call: rq(formula = wemwbs ~ log(eqvinc) * below_50k, tau = taus0, data =
subd)
##
## tau: [1] 0.35
##
## Coefficients:
##                       Value     Std. Error t value   Pr(>|t|)
## (Intercept)            37.80171  12.86263    2.93888   0.00330
## log(eqvinc)             1.20147   1.15431    1.04086   0.29798
## below_50k             -23.38379  13.15653   -1.77735   0.07555
## log(eqvinc):below_50k   2.21623   1.18663    1.86766   0.06185
##
## Call: rq(formula = wemwbs ~ log(eqvinc) * below_50k, tau = taus0, data =
subd)
##
## tau: [1] 0.5
##
## Coefficients:
##                       Value     Std. Error t value   Pr(>|t|)
## (Intercept)            38.27186   6.10228    6.27173   0.00000
## log(eqvinc)             1.41555   0.53652    2.63840   0.00835
## below_50k             -11.74048   6.58100   -1.78400   0.07447
## log(eqvinc):below_50k   1.13046   0.58955    1.91751   0.05521
##
## Call: rq(formula = wemwbs ~ log(eqvinc) * below_50k, tau = taus0, data =
subd)
##
## tau: [1] 0.7
##
## Coefficients:
##                       Value    Std. Error t value  Pr(>|t|)
## (Intercept)           35.55126 10.36004    3.43157  0.00060
## log(eqvinc)            1.88891  0.93337    2.02375  0.04303
## below_50k              8.46292 10.61310    0.79740  0.42524
## log(eqvinc):below_50k -0.72587  0.96054   -0.75569  0.44986
##
## Call: rq(formula = wemwbs ~ log(eqvinc) * below_50k, tau = taus0, data =
subd)
##
## tau: [1] 0.85
##
## Coefficients:
##                       Value    Std. Error t value  Pr(>|t|)
## (Intercept)           39.64602  8.36200    4.74121  0.00000
## log(eqvinc)            1.90472  0.72254    2.63615  0.00840
## below_50k             13.04785  9.01512    1.44733  0.14785
## log(eqvinc):below_50k -1.19578  0.79741   -1.49958  0.13377
```