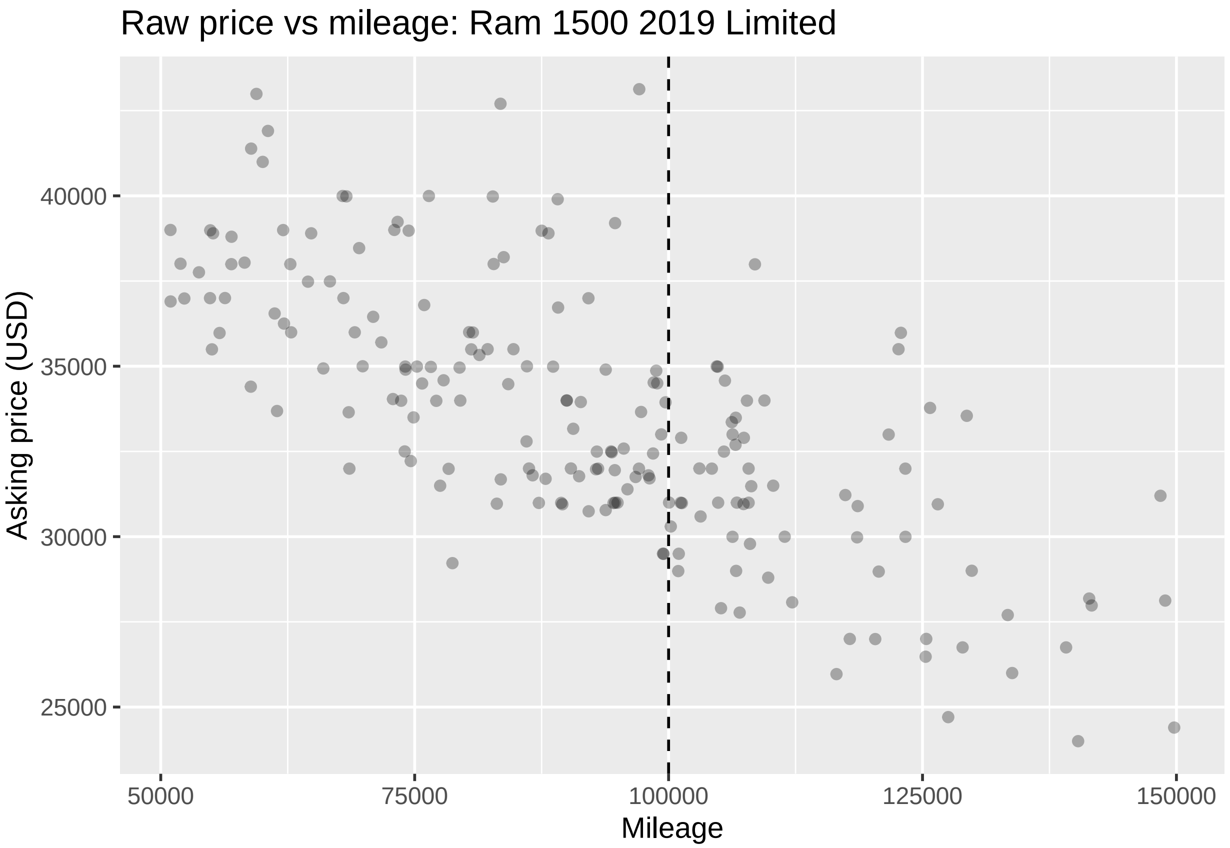
```r
                              cutoff_miles = 100000,
                              max_window   = 20000,
                              db_path      = db_path,
                              table_nm     = table_nm) {
cat("\n====================================\n")
cat("   RDD PIPELINE START\n")
cat("====================================\n\n")

cat("---- 0. PARAMETERS ----\n")
cat("make:  ", ifelse(is.null(make0),  "ALL", make0),  "\n")
cat("model: ", ifelse(is.null(model0), "ALL", model0), "\n")
cat("year:  ", ifelse(is.null(year0),  "ALL", year0),  "\n")
cat("trim:  ", ifelse(is.null(trim0),  "ALL", trim0),  "\n")
cat("cutoff (miles): ", cutoff_miles, "\n")
cat("max window (+/- miles): ", max_window, "\n\n")

## 1. CONNECT & LOAD DATA ----
cat("==== 1. CONNECTING TO DB & LOADING DATA ====\n")

# Conditional filtering depending on which arguments are NULL
if (!is.null(make0))  car_tbl <- car_tbl %>% filter(make  == make0)
if (!is.null(model0)) car_tbl <- car_tbl %>% filter(model == model0)
if (!is.null(year0))  car_tbl <- car_tbl %>% filter(year  == year0)
if (!is.null(trim0))  car_tbl <- car_tbl %>% filter(trim  == trim0)

car_df <- car_tbl %>%
  select(make, model, year, trim, price, mileage) %>%
  collect()

cat("Number of raw rows pulled: ", nrow(car_df), "\n\n")

if (nrow(car_df) == 0) {
  cat(">>> No observations for this selection. Exiting.\n")
  return(invisible(NULL))
}

## 2. CLEANING & RUNNING VARIABLE ----
cat("==== 2. CLEANING DATA & DEFINING RUNNING VARIABLE ====\n")

car_df <- car_df %>%
  mutate(
    price   = as.numeric(price),
    mileage = as.numeric(mileage)
  ) %>%
  filter(!is.na(price), !is.na(mileage))

# Optional outlier removal
car_df <- car_df %>%
  filter(
    price   > 1000,   price   < 150000,
    mileage > 10000,  mileage < 300000
  )
```

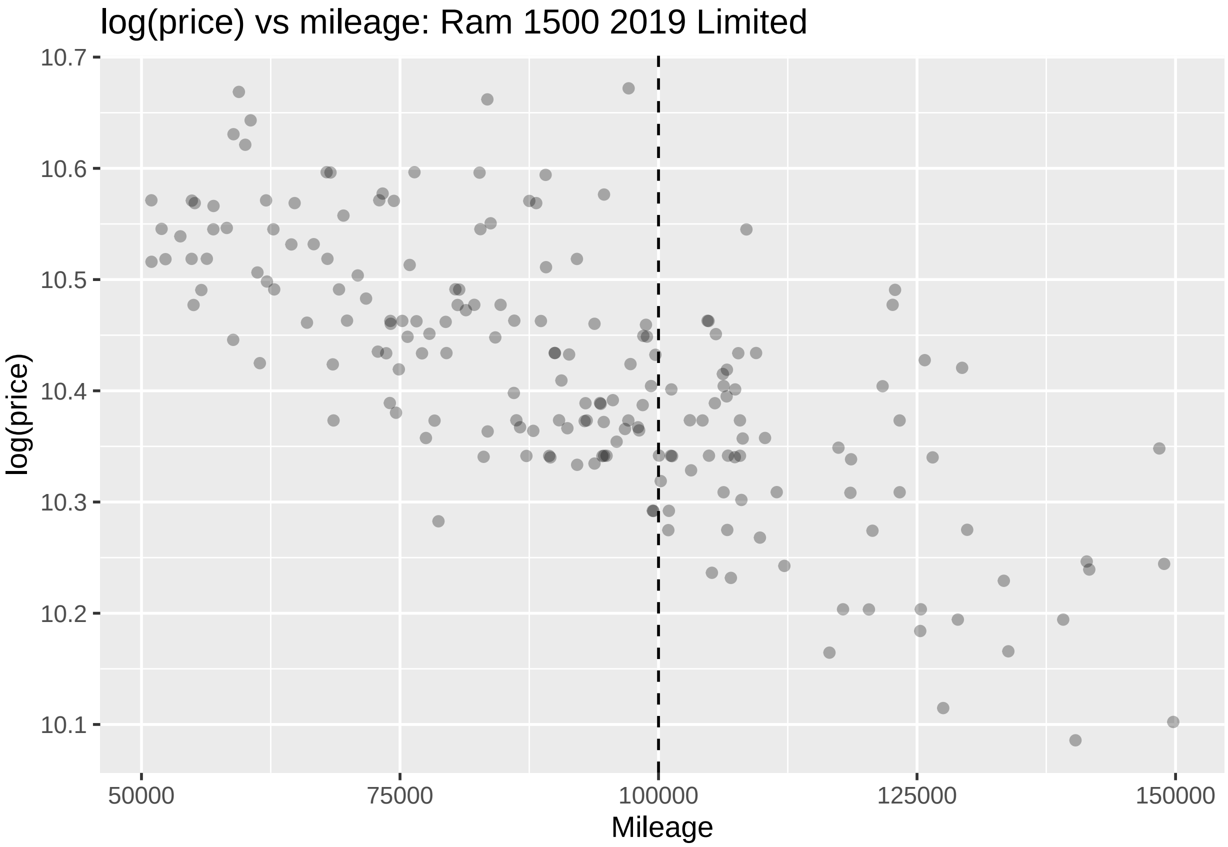
```
car_df <- car_df %>%
  mutate(
    mileage_centered = mileage - cutoff_miles,
    log_price        = log(price),
    above_cutoff     = if_else(mileage >= cutoff_miles, 1L, 0L)
  ) %>%
  filter(abs(mileage_centered) <= max_window)

car_df <- car_df %>%
  filter(
    !is.na(make),
    !is.na(model),
    !is.na(year),
    !is.na(trim)
  )

cat("Rows after cleaning & windowing: ", nrow(car_df), "\n")
cat("Summary of mileage_centered:\n")
print(summary(car_df$mileage_centered))
cat("Counts above vs below cutoff:\n")
print(table(car_df$above_cutoff))
cat("\n")

if (nrow(car_df) < 30) {
  cat(">>> WARNING: very few observations after filtering; RDD may be underpowered.\n\n")
}

## 3. BASELINE COVARIATES / POOLED LOGIC ----
cat("==== 3. BUILDING BASELINE COVARIATES (POOLED LOGIC) ====\n")

fe_terms <- c()
# Only include a variable in the FE/covariates if we did NOT fix it in the filter
if (is.null(make0))  fe_terms <- c(fe_terms, "make")
if (is.null(model0)) fe_terms <- c(fe_terms, "model")
if (is.null(year0))  fe_terms <- c(fe_terms, "factor(year)")
if (is.null(trim0))  fe_terms <- c(fe_terms, "trim")

if (length(fe_terms) > 0) {
  fe_formula <- as.formula(paste("~", paste(fe_terms, collapse = " + ")))
  cat("Using baseline covariates with formula:\n  ", deparse(fe_formula), "\n")

  Z <- model.matrix(fe_formula, data = car_df)

  # drop intercept column
  if (ncol(Z) > 1) {
    Z <- Z[, -1, drop = FALSE]
    use_covs <- TRUE
    cat("Number of baseline covariate columns: ", ncol(Z), "\n\n")
  } else {
    # only intercept; effectively no covariates
    Z <- NULL
    use_covs <- FALSE
    cat("Only intercept in baseline covariates; treating as no covariate adjustment.\n\n")
  }
} else {
  fe_formula <- NULL
  Z <- NULL
  use_covs <- FALSE
  cat("All of make/model/year/trim fixed. No covariate adjustment used.\n\n")
}

# helper wrapper to call rdrobust with or without covs
rd_with_covs <- function(y, x, c, p, kernel, bwselect = "mserd", h = NULL, all = NULL) {
  common_args <- list(y = y, x = x, c = c, p = p, kernel = kernel, bwselect = bwselect)
  if (!is.null(h))   common_args$h   <- h
  if (!is.null(all)) common_args$all <- all

  if (use_covs) {
    common_args$covs <- Z
  }
  do.call(rdrobust, common_args)
}

## 4. DESCRIPTIVE PLOTS ----
cat("==== 4. DESCRIPTIVE PLOTS ====\n")

label_vec <- c(
  if (is.null(make0))  "All makes"  else make0,
  if (is.null(model0)) "All models" else model0,
  if (is.null(year0))  "All years"  else as.character(year0),
  if (is.null(trim0))  "All trims"  else trim0
)
car_label <- paste(label_vec, collapse = " ")

p_raw_price <- ggplot(car_df, aes(x = mileage, y = price)) +
  geom_point(alpha = 0.3) +
  geom_vline(xintercept = cutoff_miles, linetype = "dashed") +
  labs(
    title = paste("Raw price vs mileage:", car_label),
    x = "Mileage",
    y = "Asking price (USD)"
  )

p_raw_log <- ggplot(car_df, aes(x = mileage, y = log_price)) +
  geom_point(alpha = 0.3) +
  geom_vline(xintercept = cutoff_miles, linetype = "dashed") +
  labs(
    title = paste("log(price) vs mileage:", car_label),
    x = "Mileage",
    y = "log(price)"
  )

p_hist_mileage <- ggplot(car_df, aes(x = mileage_centered)) +
  geom_histogram(bins = 60) +
  geom_vline(xintercept = 0, linetype = "dashed") +
  labs(
    title = "Distribution of mileage around cutoff",
    x = paste0("Mileage - ", cutoff_miles),
    y = "Count"
  )

print(p_raw_price)
print(p_raw_log)
print(p_hist_mileage)
cat("Descriptive plots printed.\n\n")

## 5. MAIN RDD ESTIMATES (log price & price) ----
cat("==== 5. MAIN SHARP RDD ESTIMATES ====\n")
cat("5.1 RDD on log(price)\n")

rd_log_main <- rd_with_covs(
  y = car_df$log_price,
  x = car_df$mileage,
  c = cutoff_miles,
  p = 1,
  kernel = "triangular",
  bwselect = "mserd",
  all = "i"
)
print(summary(rd_log_main))
cat("\n")

cat("5.2 RDD on price levels\n")
rd_price_main <- rd_with_covs(
  y = car_df$price,
  x = car_df$mileage,
  c = cutoff_miles,
  p = 1,
  kernel = "triangular",
  bwselect = "mserd",
  all = "i"
)
print(summary(rd_price_main))
cat("\n")

## 6. RDD PLOTS (rdplot) ----
cat("==== 6. RDD PLOTS (rdplot) ====\n")

rdplot_log <- rdplot(
  y = car_df$log_price,
  x = car_df$mileage,
  c = cutoff_miles,
  title   = paste("RDD plot: log(price) vs mileage around", cutoff_miles),
  x.label = "Mileage",
  y.label = "log(price)"
)

rdplot_price <- rdplot(
  y = car_df$price,
```

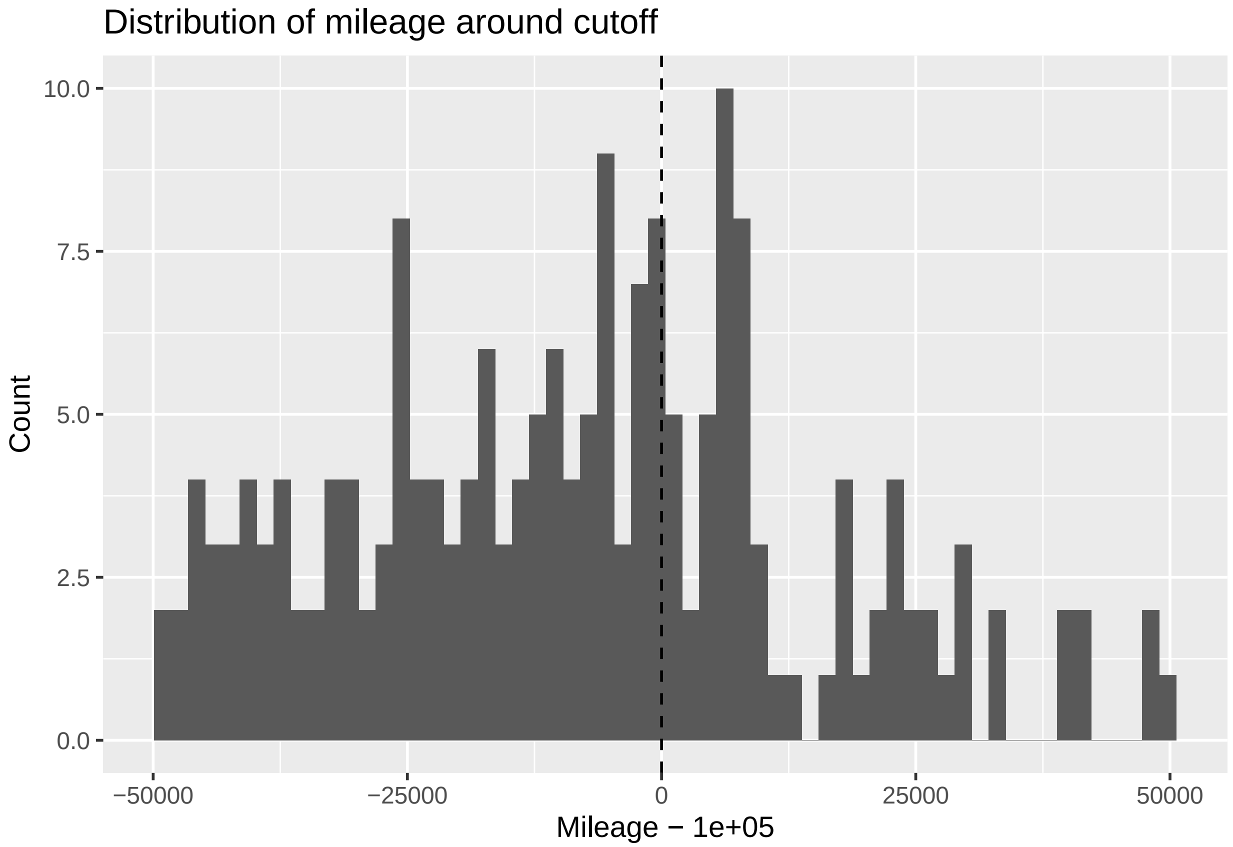
```r
    x = car_df$mileage,
    c = cutoff_miles,
    title   = paste("RDD plot: price vs mileage around", cutoff_miles),
    x.label = "Mileage",
    y.label = "Price"
  )

  cat("rdplot objects created (printed in plotting window).\n\n")

  cat("====================================\n")
  cat("   RDD PIPELINE COMPLETE\n")
  cat("====================================\n\n")

  # Return key objects for later use
  invisible(list(
    data              = car_df,
    covariate_formula = fe_formula,
    covariates_matrix = Z,
    rd_log_main       = rd_log_main,
    rd_price_main     = rd_price_main,
    plots = list(
      raw_price     = p_raw_price,
      raw_log       = p_raw_log,
      hist_mileage  = p_hist_mileage
    ),
    rdplots = list(
      log_price  = rdplot_log,
      price      = rdplot_price
    )
  ))
}
result_ram_2019_limited <- run_rdd_for_vehicle(
  make0   = "Ram",
  model0  = "1500",
  year0   = 2019L,
  trim0   = "Limited",
  cutoff_miles = 100000,
  max_window   = 50000
)
```

```
##
## ====================================
##    RDD PIPELINE START
## ====================================
##
## ---- 0. PARAMETERS ----
## make:   Ram
## model:  1500
## year:   2019
## trim:   Limited
## cutoff (miles):  1e+05
## max window (+/- miles):  50000
##
## ==== 1. CONNECTING TO DB & LOADING DATA ====
## Number of raw rows pulled:  218
##
## ==== 2. CLEANING DATA & DEFINING RUNNING VARIABLE ====
## Rows after cleaning & windowing:  189
## Summary of mileage_centered:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -49043  -25385   -7870   -8196    6300   49781
## Counts above vs below cutoff:
##
##   0   1
## 123  66
##
## ==== 3. BUILDING BASELINE COVARIATES (POOLED LOGIC) ====
## All of make/model/year/trim fixed. No covariate adjustment used.
##
## ==== 4. DESCRIPTIVE PLOTS ====
```

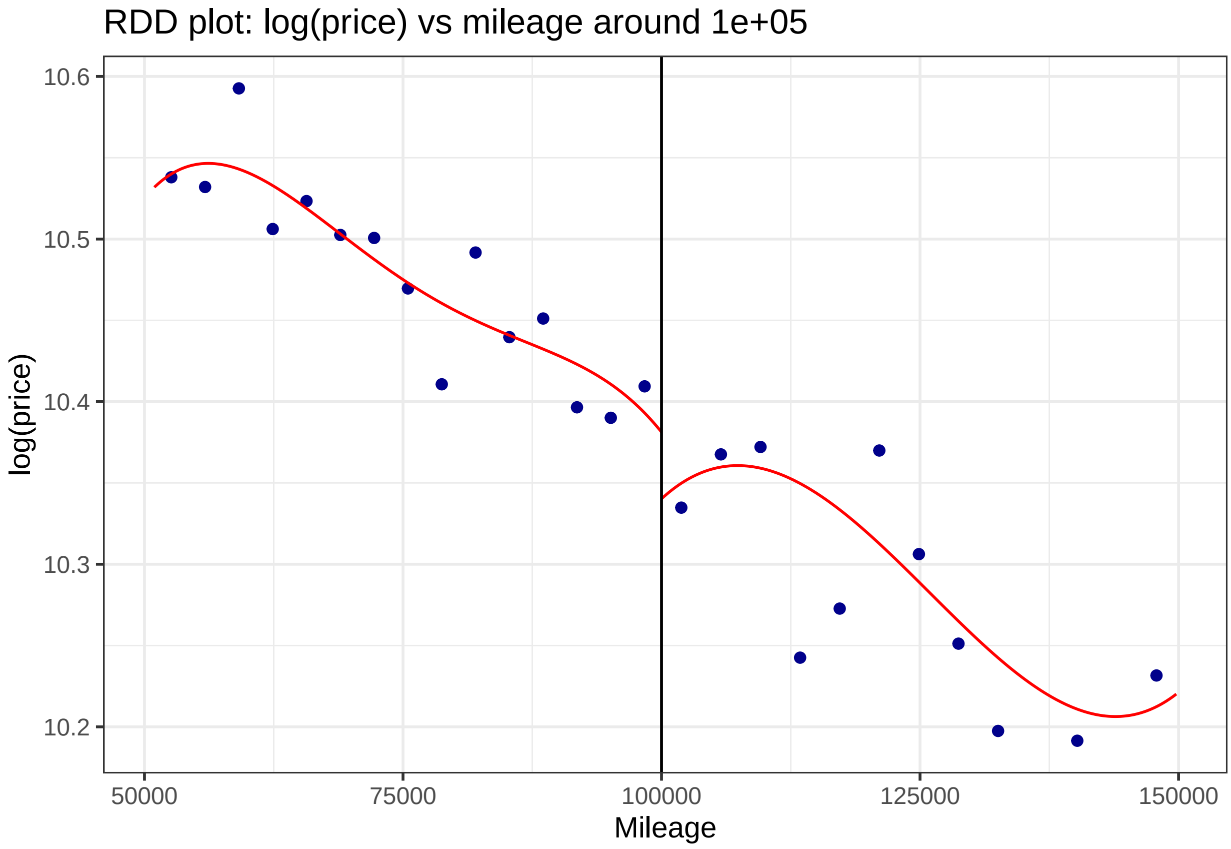


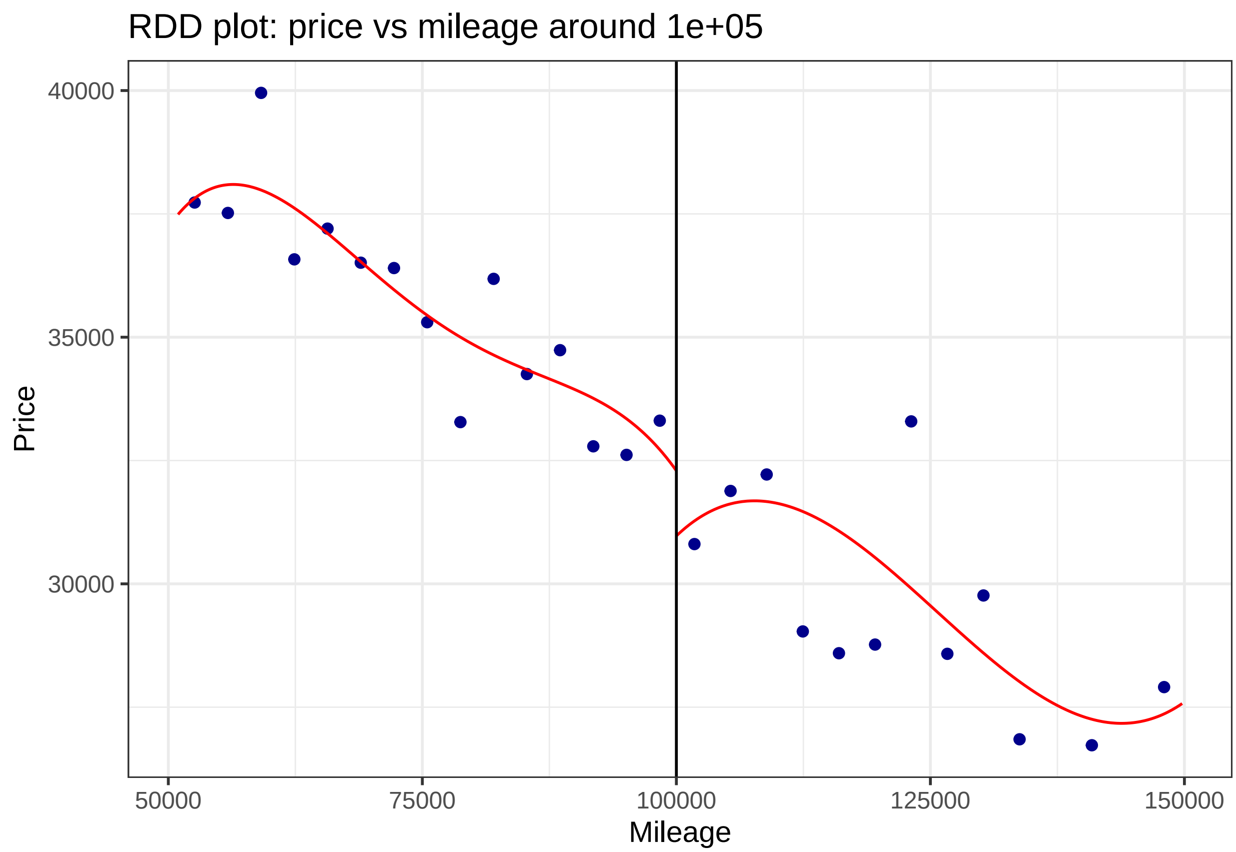
Raw price vs mileage: Ram 1500 2019 Limited


log(price) vs mileage: Ram 1500 2019 Limited
log(price)
10.7
10.6
10.5
10.4
10.3
10.2
10.1
Mileage
50000
75000
100000
125000
150000



```
## Descriptive plots printed.
##
## ==== 5. MAIN SHARP RDD ESTIMATES ====
## 5.1 RDD on log(price)
## Sharp RD estimates using local polynomial regression.
##
## Number of Obs.                 189
## BW type                      mserd
## Kernel                  Triangular
## VCE method                      NN
##
## Number of Obs.                 123           66
## Eff. Number of Obs.             49           37
## Order est. (p)                   1            1
## Order bias  (q)                  2            2
## BW est. (h)              14668.493    14668.493
## BW bias (b)              23776.708    23776.708
## rho (h/b)                    0.617        0.617
## Unique Obs.                    123           66
##
## =====================================================================================
##         Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
## =====================================================================================
##   Conventional    -0.053     0.027    -1.975     0.048    [-0.106 , -0.000]
## Bias-Corrected    -0.062     0.027    -2.304     0.021    [-0.115 , -0.009]
##         Robust    -0.062     0.030    -2.049     0.040    [-0.122 , -0.003]
## =======================================================================================
## NULL
##
## 5.2 RDD on price levels
## Sharp RD estimates using local polynomial regression.
##
## Number of Obs.                  189
## BW type                       mserd
## Kernel                   Triangular
## VCE method                       NN
##
## Number of Obs.                  123           66
## Eff. Number of Obs.              47           37
## Order est. (p)                    1            1
## Order bias  (q)                   2            2
## BW est. (h)               13928.536    13928.536
## BW bias (b)               23218.950    23218.950
## rho (h/b)                     0.600        0.600
## Unique Obs.                     123           66
##
## =======================================================================================
##         Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
## =======================================================================================
##   Conventional -1829.612   909.624    -2.011     0.044 [-3612.442 , -46.782]
## Bias-Corrected -2180.071   909.624    -2.397     0.017 [-3962.901 , -397.241]
##         Robust -2180.071  1012.618    -2.153     0.031 [-4164.766 , -195.376]
## =======================================================================================
## NULL
##
## ==== 6. RDD PLOTS (rdplot) ====
```


RDD plot: log(price) vs mileage around 1e+05
log(price)
10.6
10.5
10.4
10.3
10.2
50000
75000
100000
125000
150000
Mileage

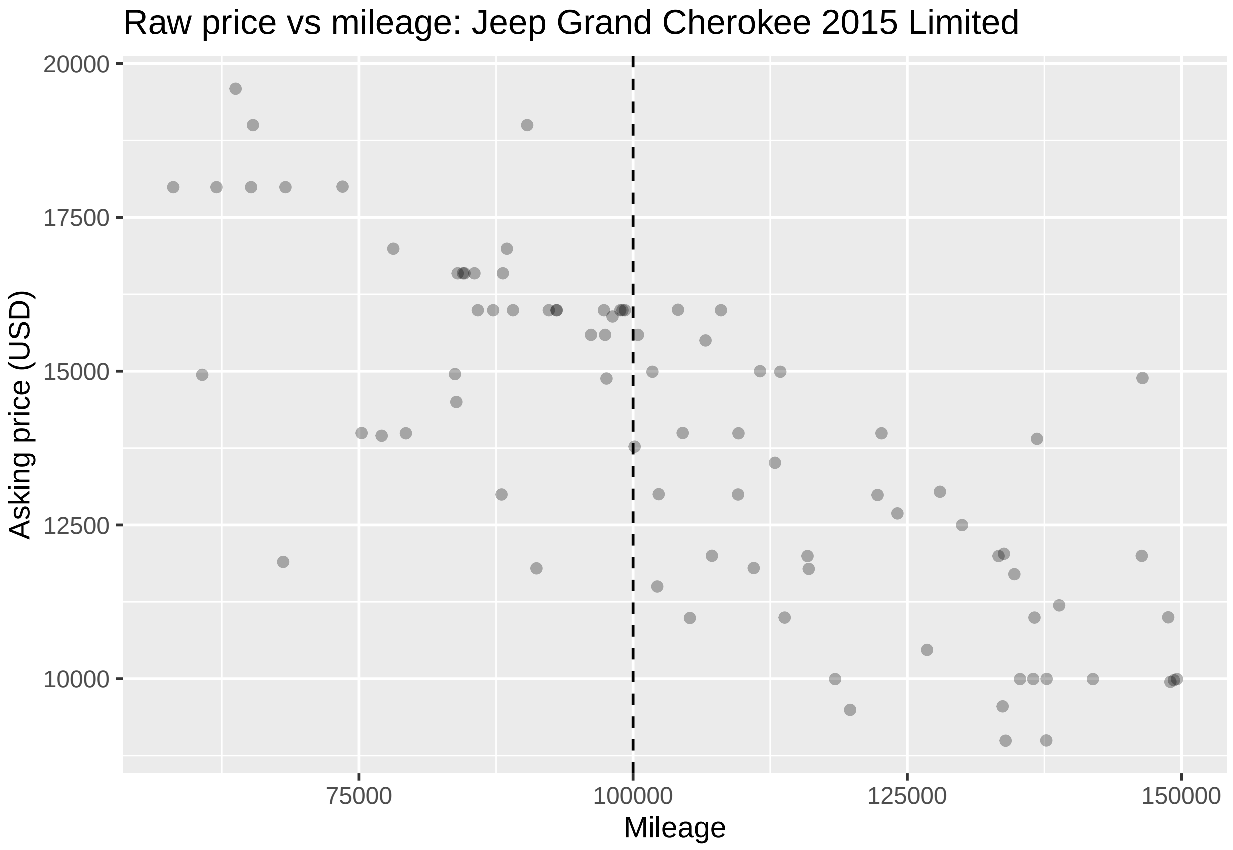


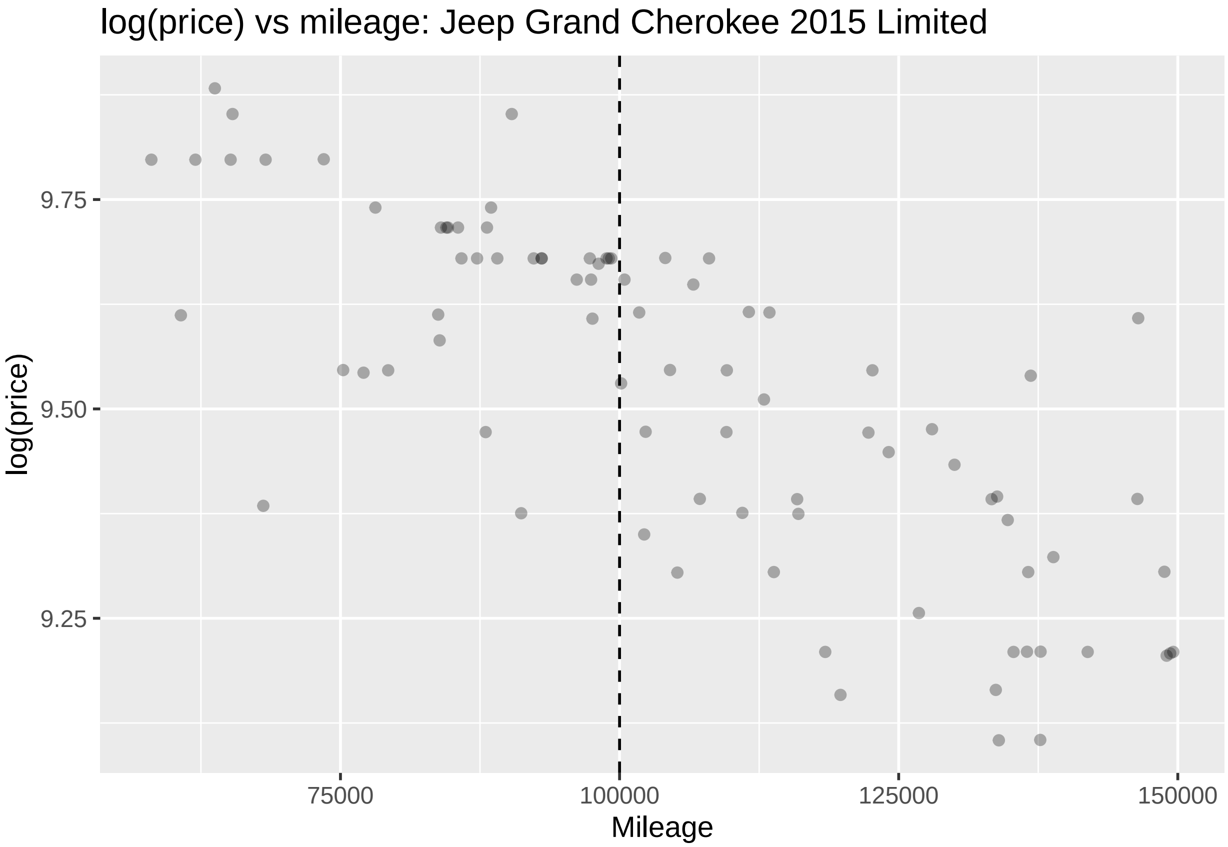
```
## rdplot objects created (printed in plotting window).
##
## =====================================
##    RDD PIPELINE COMPLETE
## =====================================
```

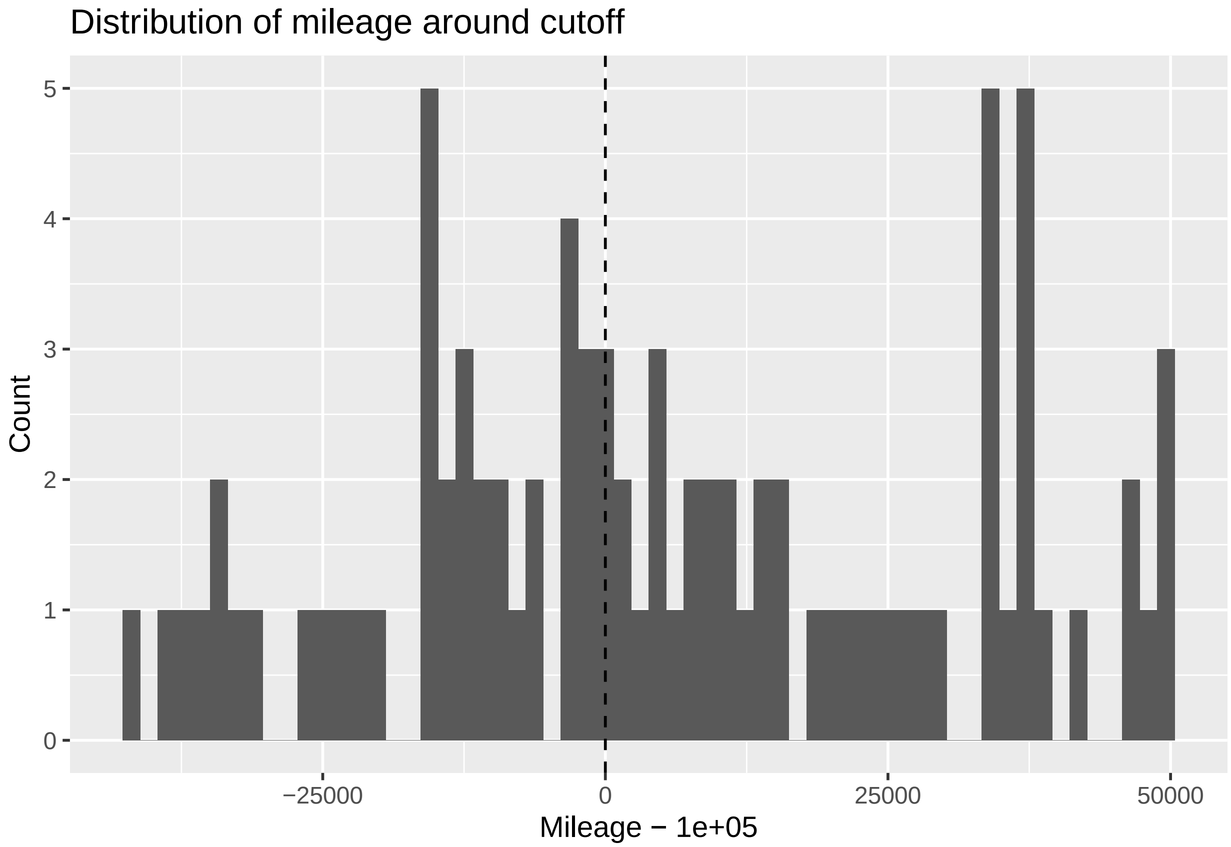
```
result_jeep_limited <- run_rdd_for_vehicle(
  make0   = "Jeep",
  model0 = "Grand Cherokee",
  year0   = 2015L,
  trim0   = "Limited",
  cutoff_miles = 100000,
  max_window    = 50000
)
```

```
##
## =====================================
##    RDD PIPELINE START
## =====================================
##
## ---- 0. PARAMETERS ----
## make:    Jeep
## model:   Grand Cherokee
## year:    2015
## trim:    Limited
## cutoff (miles):  1e+05
## max window (+/- miles):  50000
```

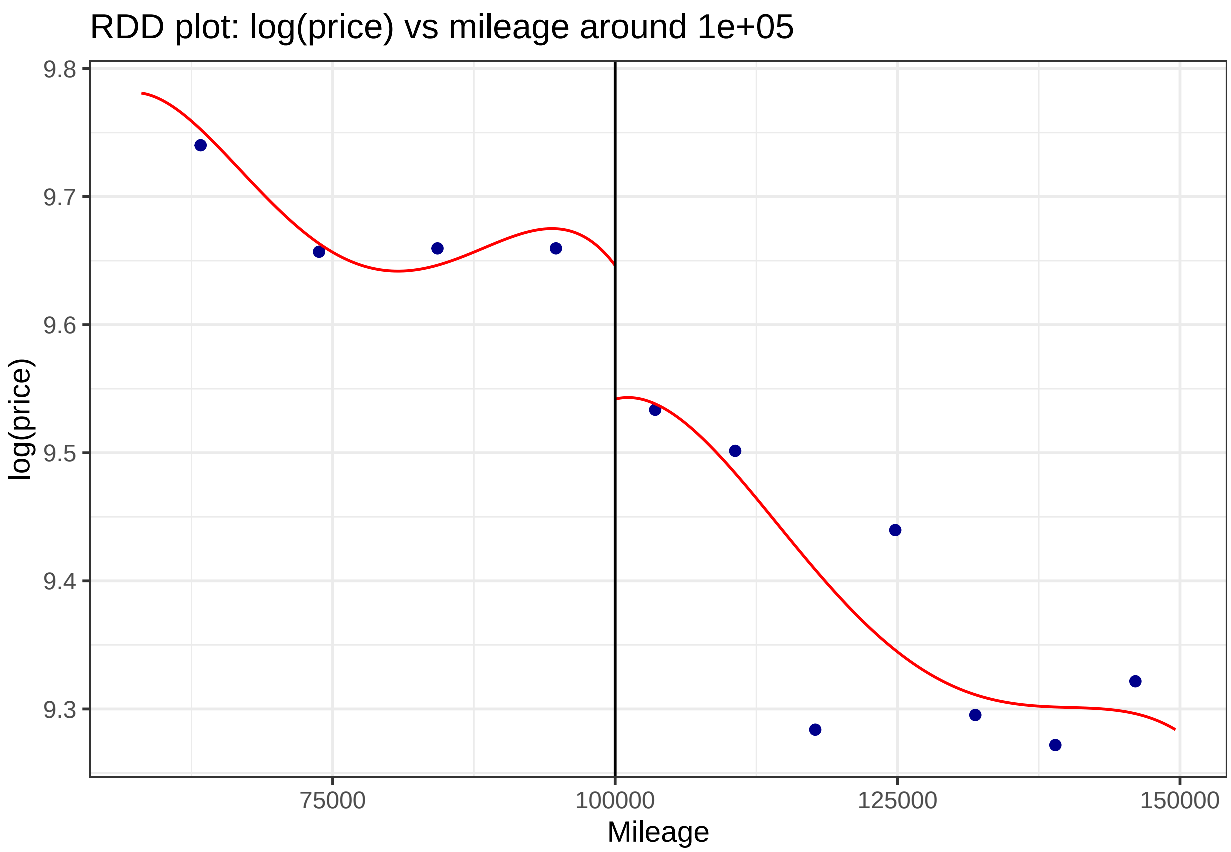
```
##
## ==== 1. CONNECTING TO DB & LOADING DATA ====
## Number of raw rows pulled:  99
##
## ==== 2. CLEANING DATA & DEFINING RUNNING VARIABLE ====
## Rows after cleaning & windowing:  85
## Summary of mileage_centered:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -41936  -11995    2336    5651   26811   49597
## Counts above vs below cutoff:
##
##  0  1
## 38 47
##
## ==== 3. BUILDING BASELINE COVARIATES (POOLED LOGIC) ====
## All of make/model/year/trim fixed. No covariate adjustment used.
##
## ==== 4. DESCRIPTIVE PLOTS ====
```

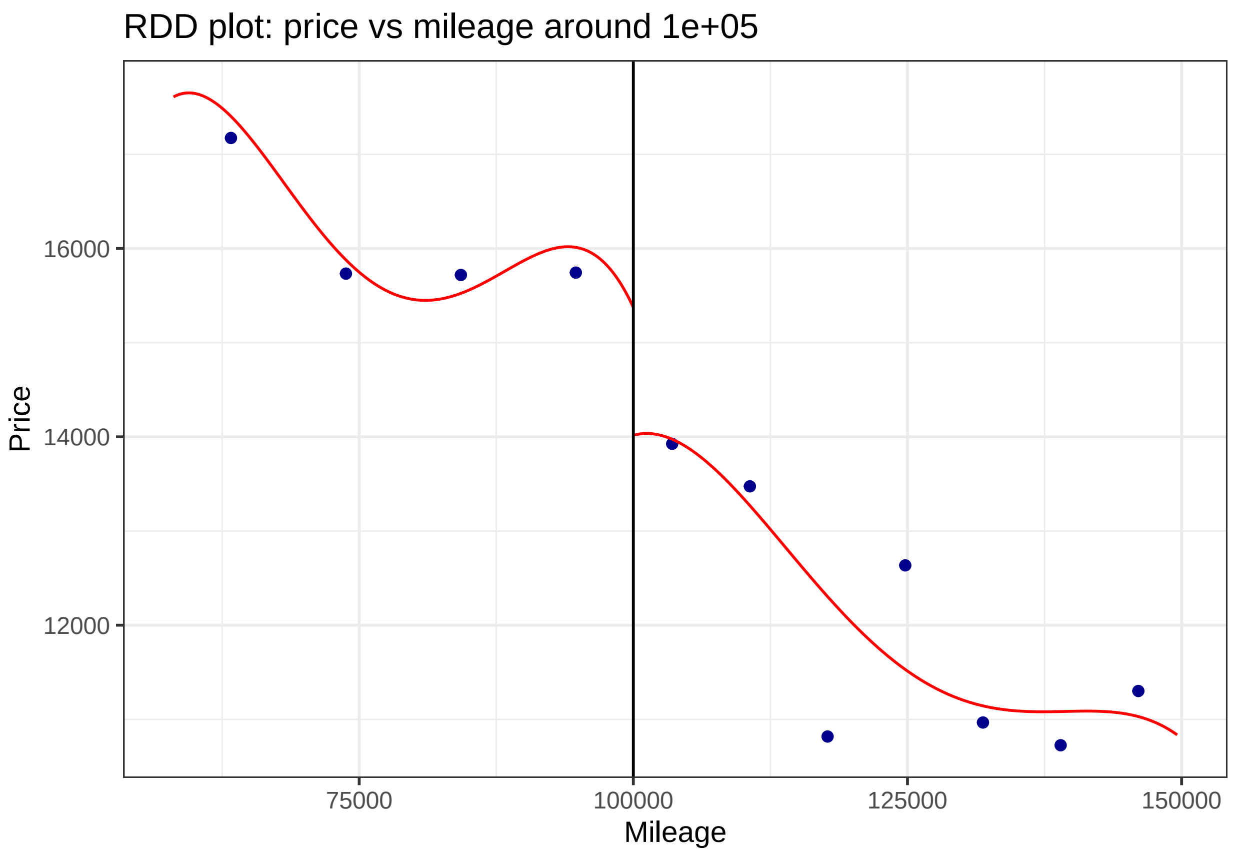



log(price) vs mileage: Jeep Grand Cherokee 2015 Limited
log(price)
9.75
9.50
9.25
Mileage
75000
100000
125000
150000



```
## Descriptive plots printed.
##
## ==== 5. MAIN SHARP RDD ESTIMATES ====
## 5.1 RDD on log(price)
## Sharp RD estimates using local polynomial regression.
##
## Number of Obs.                   85
## BW type                       mserd
## Kernel                   Triangular
## VCE method                       NN
##
## Number of Obs.                   38           47
## Eff. Number of Obs.              18           18
## Order est. (p)                    1            1
## Order bias  (q)                   2            2
## BW est. (h)               13971.099    13971.099
## BW bias (b)               22528.059    22528.059
## rho (h/b)                     0.620        0.620
## Unique Obs.                      38           47
##
## =======================================================================================
##         Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
## =======================================================================================
##   Conventional    -0.124     0.057    -2.168     0.030    [-0.235 , -0.012]
## Bias-Corrected    -0.144     0.057    -2.519     0.012    [-0.255 , -0.032]
##         Robust    -0.144     0.067    -2.138     0.033    [-0.275 , -0.012]
## =======================================================================================
## NULL
##
## 5.2 RDD on price levels
## Sharp RD estimates using local polynomial regression.
##
## Number of Obs.                   85
## BW type                       mserd
## Kernel                   Triangular
## VCE method                       NN
##
## Number of Obs.                   38           47
## Eff. Number of Obs.              20           18
## Order est. (p)                    1            1
## Order bias  (q)                   2            2
## BW est. (h)               15300.539    15300.539
## BW bias (b)               23609.224    23609.224
## rho (h/b)                     0.648        0.648
## Unique Obs.                      38           47
##
## =======================================================================================
##         Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
## =======================================================================================
##   Conventional -1695.197   768.395    -2.206     0.027 [-3201.223 , -189.172]
## Bias-Corrected -1861.273   768.395    -2.422     0.015 [-3367.298 , -355.247]
##         Robust -1861.273   900.246    -2.068     0.039 [-3625.722 , -96.823]
## =======================================================================================
## NULL
##
## ==== 6. RDD PLOTS (rdplot) ====
```


RDD plot: log(price) vs mileage around 1e+05
log(price)
9.8
9.7
9.6
9.5
9.4
9.3
75000
100000
125000
150000
Mileage

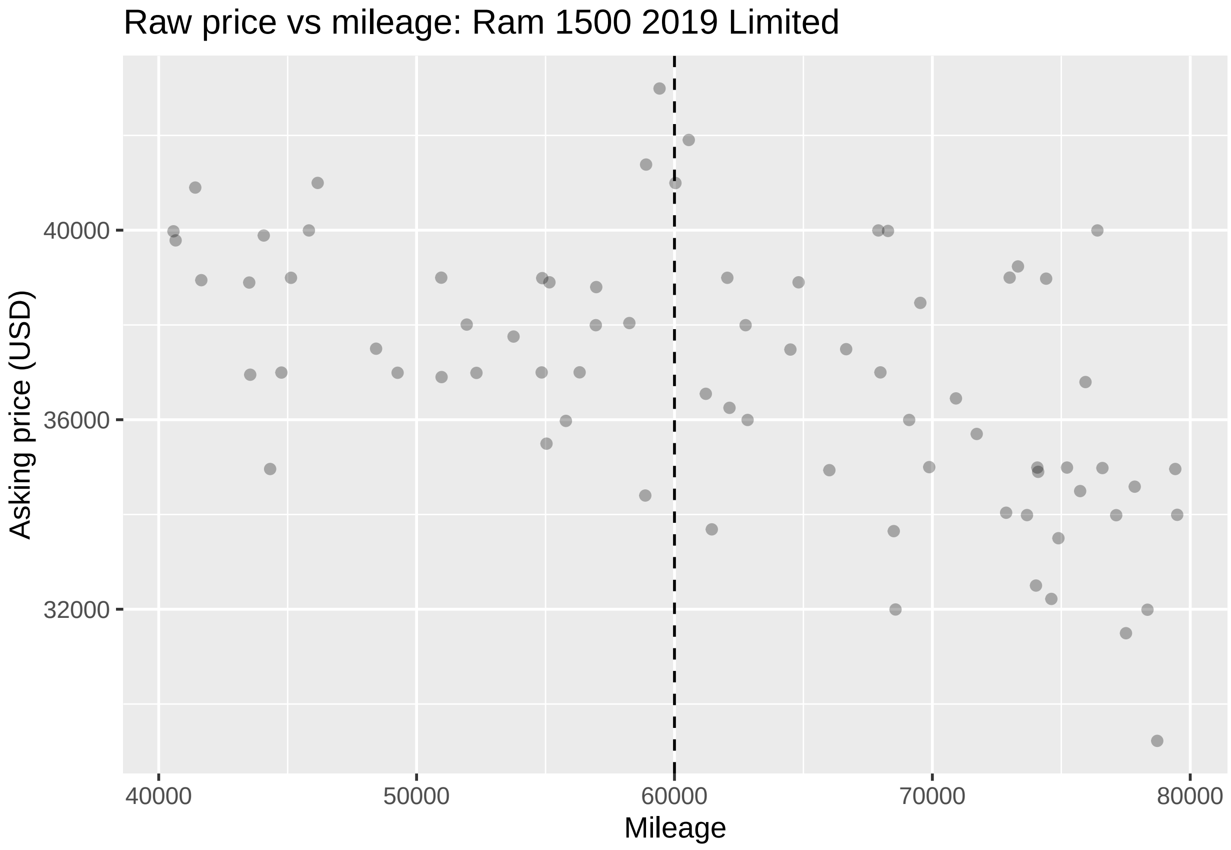


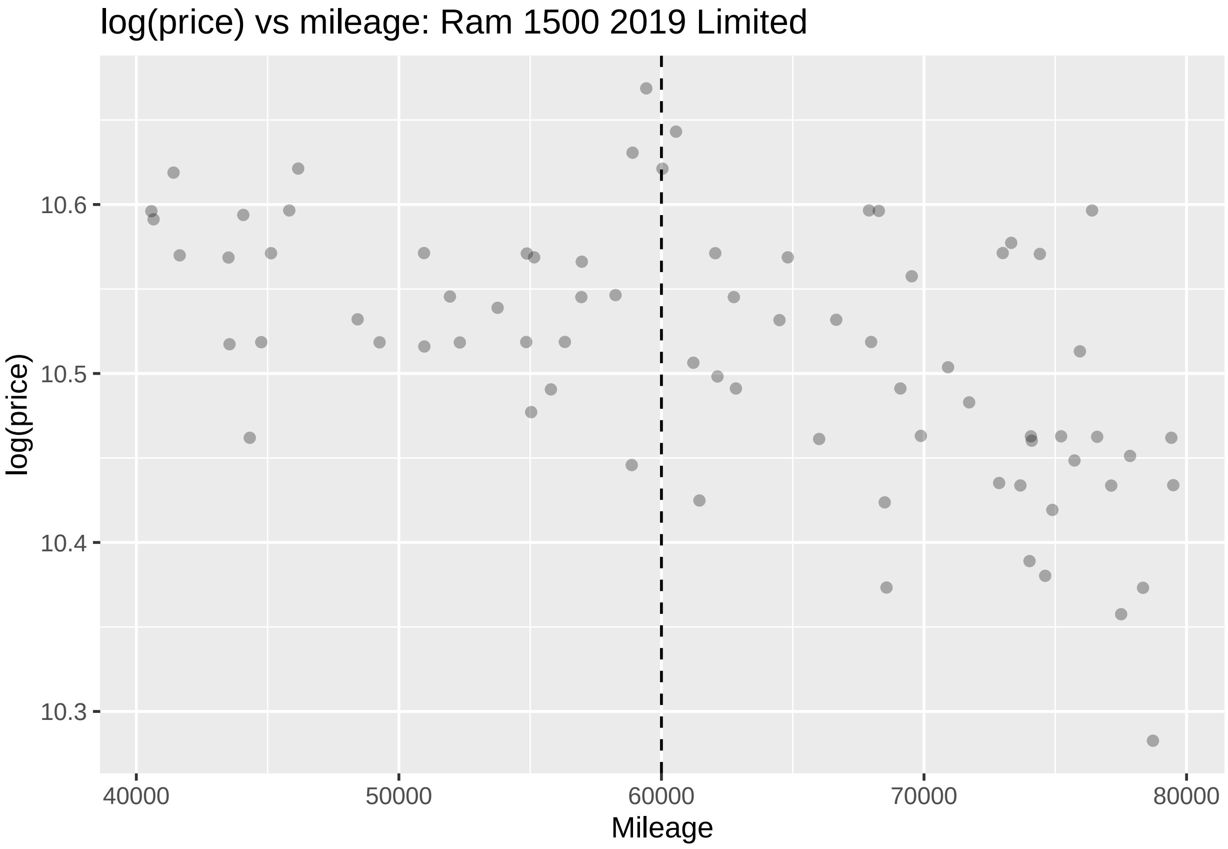
```
## rdplot objects created (printed in plotting window).
##
## =====================================
##    RDD PIPELINE COMPLETE
## =====================================
```

```
result_ram_2019_limited_60k <- run_rdd_for_vehicle(
  make0   = "Ram",
  model0 = "1500",
  year0   = 2019L,
  trim0   = "Limited",
  cutoff_miles = 60000,
  max_window    = 20000
)
```

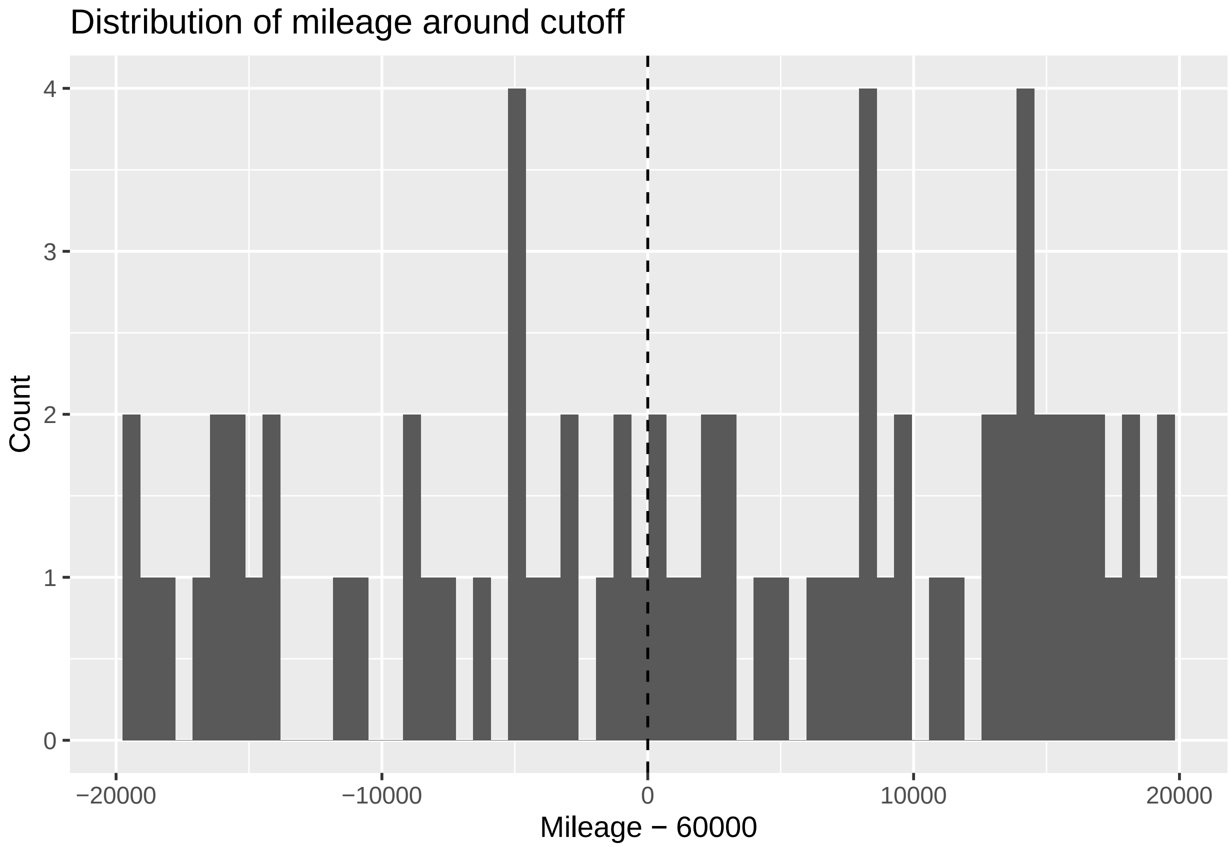
```
##
## =====================================
##    RDD PIPELINE START
## =====================================
##
## ---- 0. PARAMETERS ----
## make:    Ram
## model:   1500
## year:    2019
## trim:    Limited
## cutoff (miles):  60000
## max window (+/- miles):  20000
```

```
##
## ==== 1. CONNECTING TO DB & LOADING DATA ====
## Number of raw rows pulled:  218
##
## ==== 2. CLEANING DATA & DEFINING RUNNING VARIABLE ====
## Rows after cleaning & windowing:  75
## Summary of mileage_centered:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -19427   -5695    2758    2338   13495   19494
## Counts above vs below cutoff:
##
##  0  1
## 31 44
##
## ==== 3. BUILDING BASELINE COVARIATES (POOLED LOGIC) ====
## All of make/model/year/trim fixed. No covariate adjustment used.
##
## ==== 4. DESCRIPTIVE PLOTS ====
```

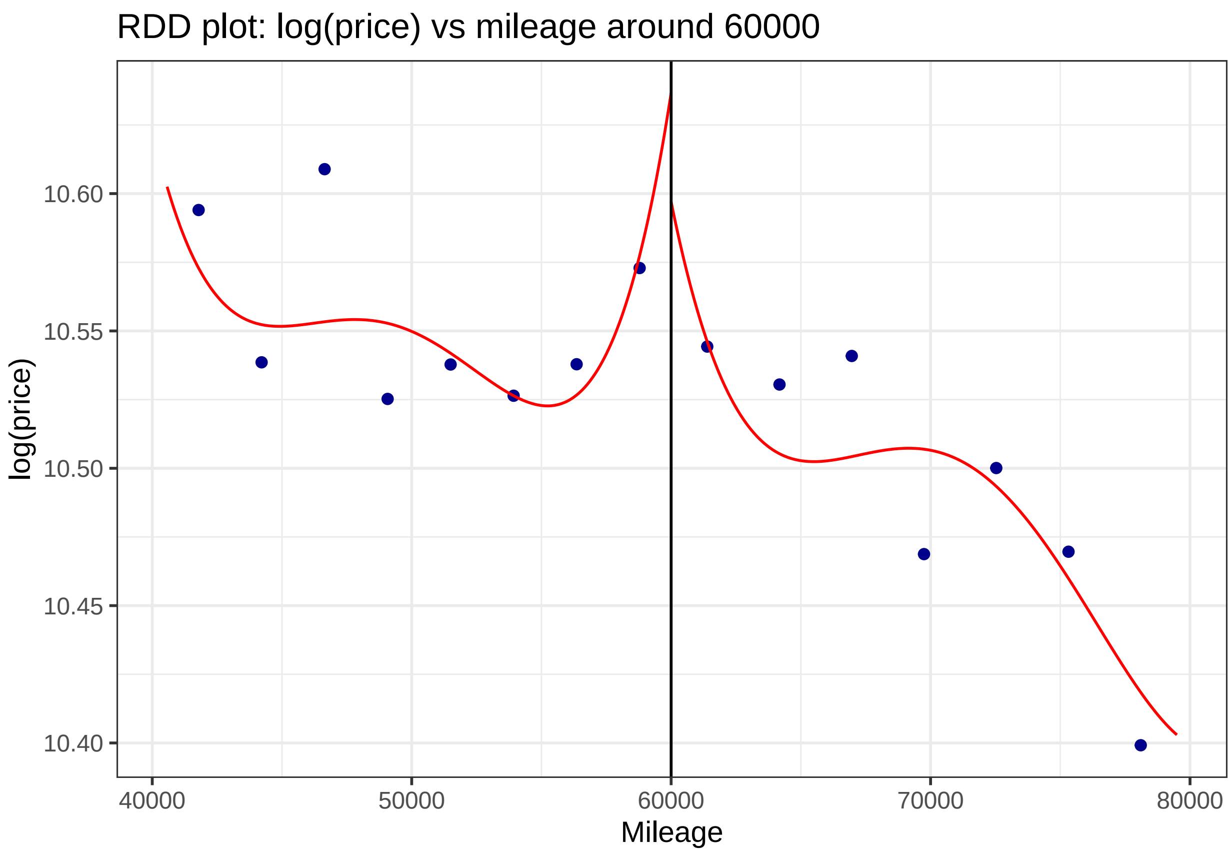


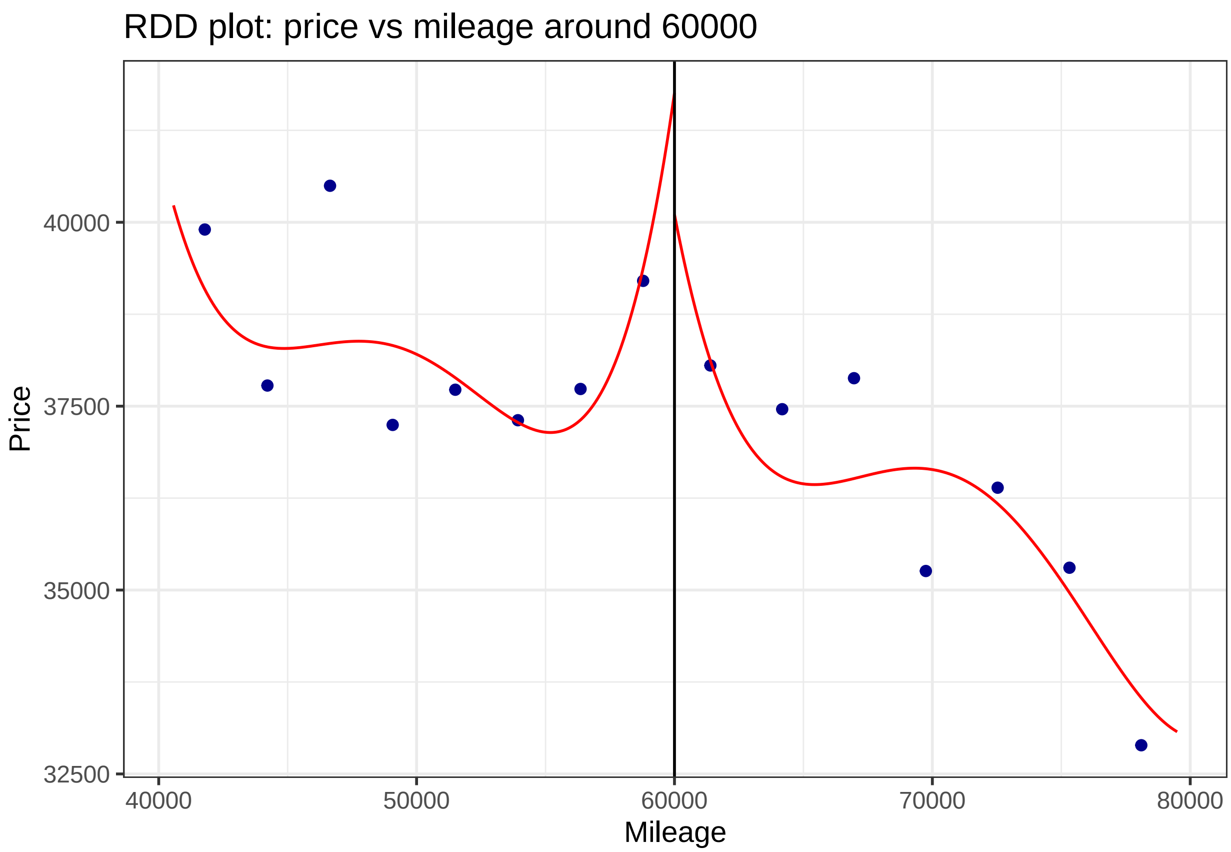


log(price) vs mileage: Ram 1500 2019 Limited
log(price)
10.6
10.5
10.4
10.3
Mileage
40000
50000
60000
70000
80000


Distribution of mileage around cutoff

```
## Descriptive plots printed.
##
## ==== 5. MAIN SHARP RDD ESTIMATES ====
## 5.1 RDD on log(price)
## Sharp RD estimates using local polynomial regression.
##
## Number of Obs.                   75
## BW type                       mserd
## Kernel                   Triangular
## VCE method                       NN
##
## Number of Obs.                   31           44
## Eff. Number of Obs.              13           12
## Order est. (p)                    1            1
## Order bias  (q)                   2            2
## BW est. (h)                7167.751     7167.751
## BW bias (b)                9508.894     9508.894
## rho (h/b)                     0.754        0.754
## Unique Obs.                      31           44
##
## ========================================================================================
##         Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
## ========================================================================================
##   Conventional    -0.021     0.081    -0.259     0.796    [-0.179 , 0.137]
## Bias-Corrected    -0.026     0.081    -0.326     0.744    [-0.185 , 0.132]
##         Robust    -0.026     0.102    -0.259     0.796    [-0.226 , 0.173]
## =======================================================================================
## NULL
##
## 5.2 RDD on price levels
## Sharp RD estimates using local polynomial regression.
##
## Number of Obs.                   75
## BW type                       mserd
## Kernel                   Triangular
## VCE method                       NN
##
## Number of Obs.                   31             44
## Eff. Number of Obs.              13             12
## Order est. (p)                    1              1
## Order bias  (q)                   2              2
## BW est. (h)                6847.105       6847.105
## BW bias (b)                9163.717       9163.717
## rho (h/b)                     0.747          0.747
## Unique Obs.                      31             44
##
## =======================================================================================
##         Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
## =======================================================================================
##   Conventional  -820.626  3157.905    -0.260     0.795 [-7010.005 , 5368.754]
## Bias-Corrected -1101.478  3157.905    -0.349     0.727 [-7290.857 , 5087.902]
##         Robust -1101.478  4016.593    -0.274     0.784 [-8973.855 , 6770.899]
## =======================================================================================
## NULL
##
## ==== 6. RDD PLOTS (rdplot) ====
```


RDD plot: log(price) vs mileage around 60000

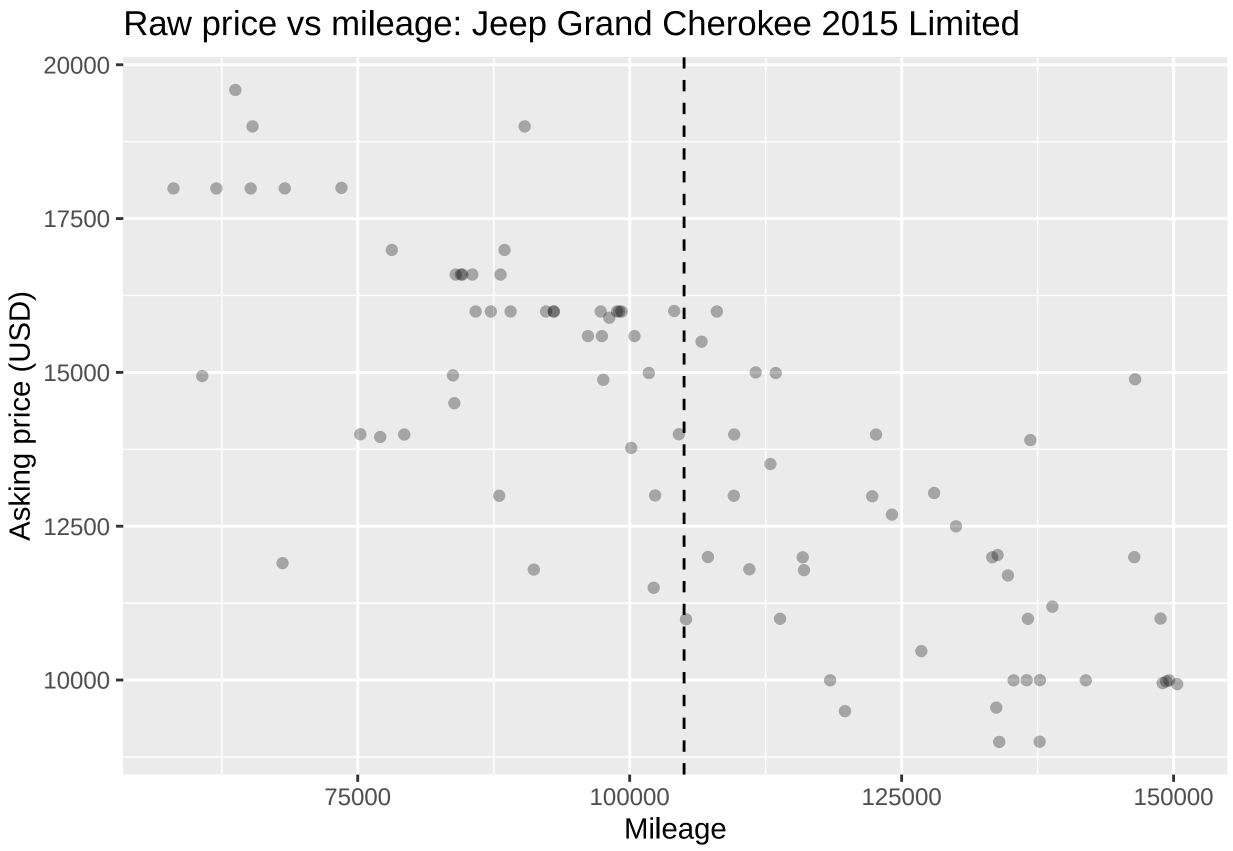


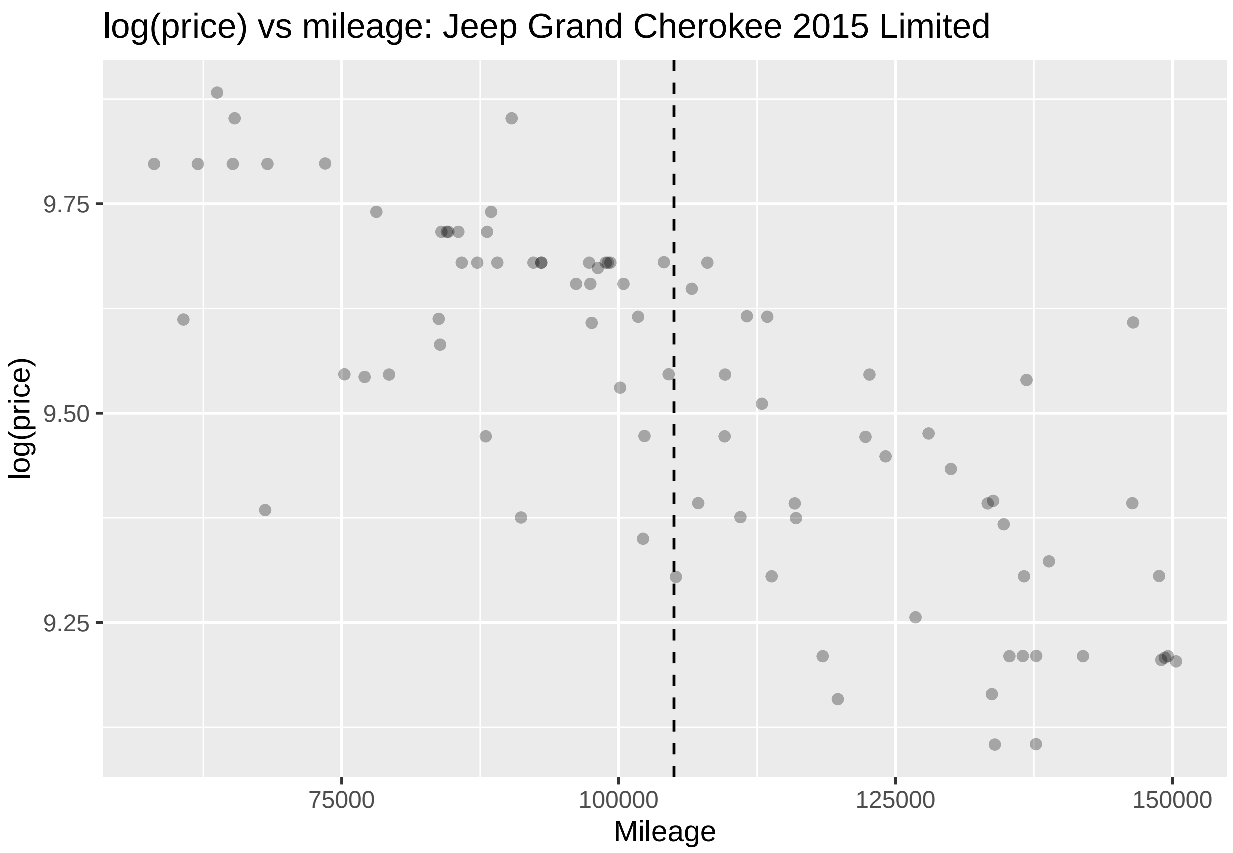
```
## rdplot objects created (printed in plotting window).
##
## ====================================
##    RDD PIPELINE COMPLETE
## ====================================
```

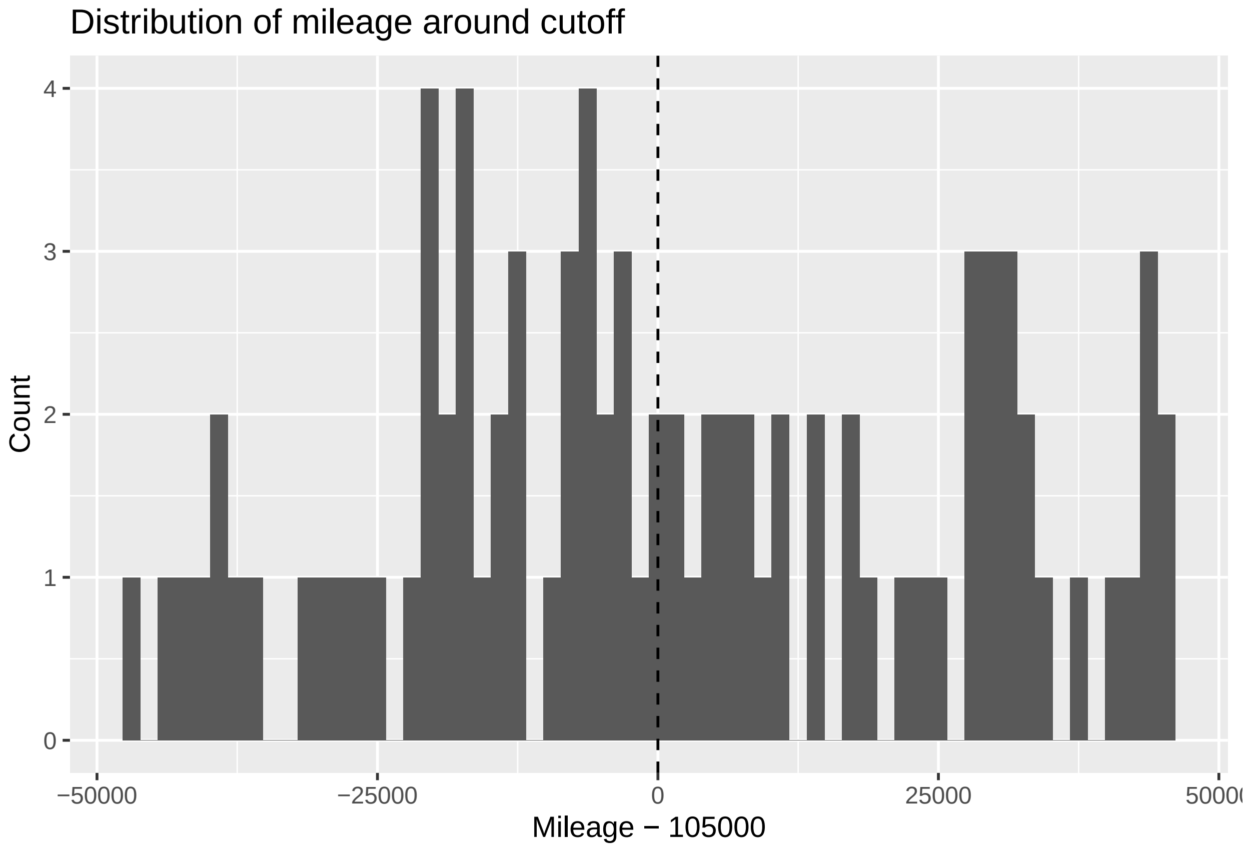
```r
result_jeep_limited_105k <- run_rdd_for_vehicle(
  make0   = "Jeep",
  model0 = "Grand Cherokee",
  year0   = 2015L,
  trim0   = "Limited",
  cutoff_miles = 105000,
  max_window    = 50000
)
```

```
##
## ====================================
##    RDD PIPELINE START
## ====================================
##
## ---- 0. PARAMETERS ----
## make:    Jeep
## model:   Grand Cherokee
## year:    2015
## trim:    Limited
## cutoff (miles):  105000
## max window (+/- miles):  50000
```

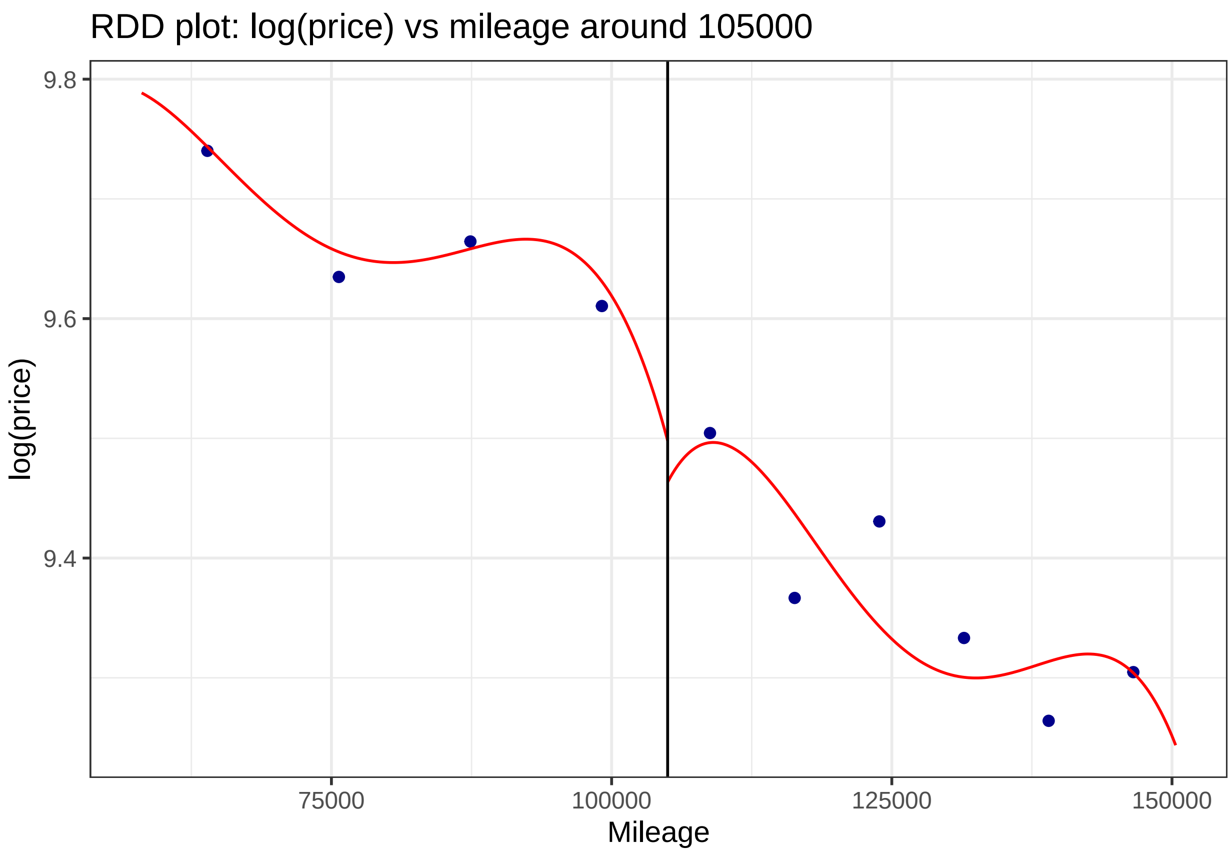
```
##
## ==== 1. CONNECTING TO DB & LOADING DATA ====
## Number of raw rows pulled:  99
##
## ==== 2. CLEANING DATA & DEFINING RUNNING VARIABLE ====
## Rows after cleaning & windowing:  86
## Summary of mileage_centered:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -46936  -16965   -1786    1170   22693   45330
## Counts above vs below cutoff:
##
##  0  1
## 45 41
##
## ==== 3. BUILDING BASELINE COVARIATES (POOLED LOGIC) ====
## All of make/model/year/trim fixed. No covariate adjustment used.
##
## ==== 4. DESCRIPTIVE PLOTS ====
```

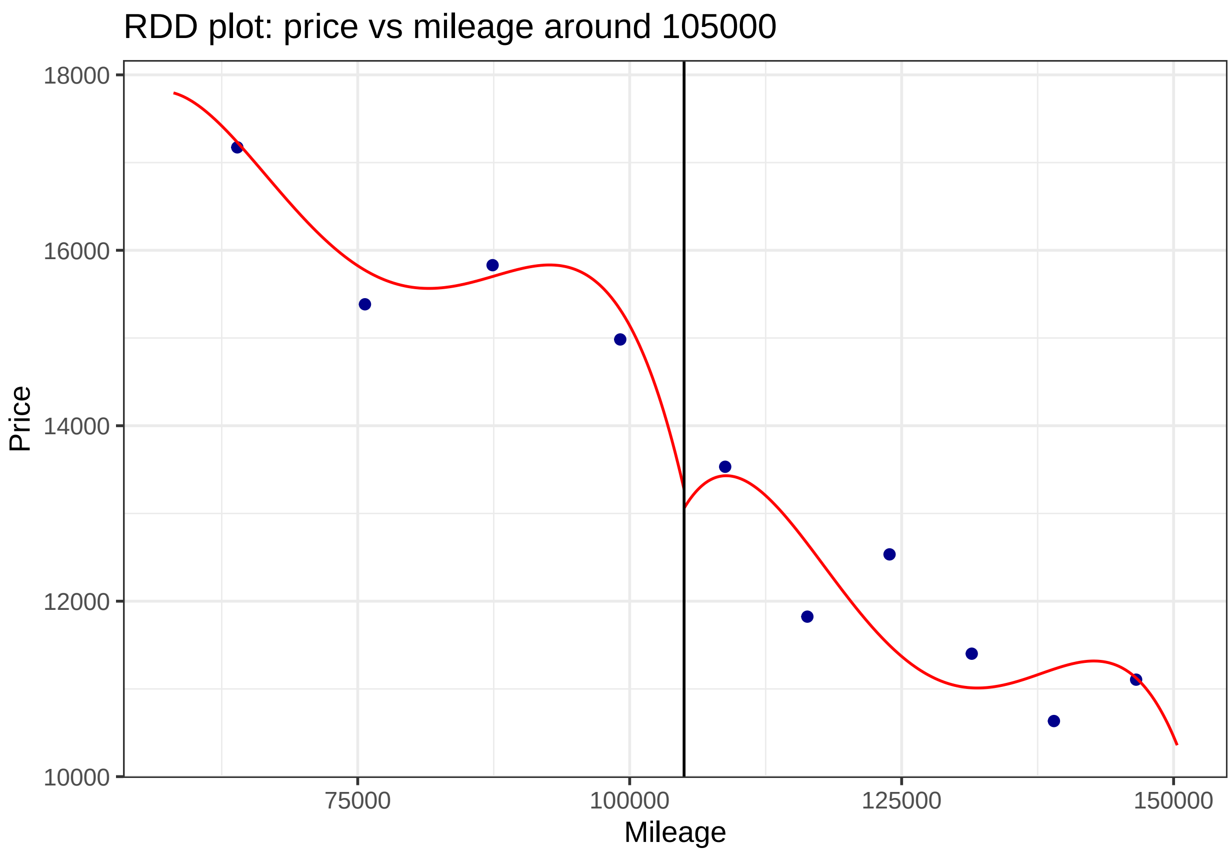



log(price) vs mileage: Jeep Grand Cherokee 2015 Limited
log(price)
9.75
9.50
9.25
Mileage
75000
100000
125000
150000


Distribution of mileage around cutoff

```
## Descriptive plots printed.
##
## ==== 5. MAIN SHARP RDD ESTIMATES ====
## 5.1 RDD on log(price)
## Sharp RD estimates using local polynomial regression.
##
## Number of Obs.                   86
## BW type                       mserd
## Kernel                   Triangular
## VCE method                       NN
##
## Number of Obs.                   45           41
## Eff. Number of Obs.              25           17
## Order est. (p)                    1            1
## Order bias  (q)                   2            2
## BW est. (h)               18097.532    18097.532
## BW bias (b)               29268.315    29268.315
## rho (h/b)                     0.618        0.618
## Unique Obs.                      45           41
##
## =======================================================================================
##         Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
## =======================================================================================
##   Conventional    -0.027     0.108    -0.251     0.802    [-0.240 , 0.185]
## Bias-Corrected    -0.004     0.108    -0.036     0.971    [-0.216 , 0.208]
##         Robust    -0.004     0.132    -0.030     0.976    [-0.263 , 0.255]
## =======================================================================================
## NULL
##
## 5.2 RDD on price levels
## Sharp RD estimates using local polynomial regression.
##
## Number of Obs.                 86
## BW type                     mserd
## Kernel                 Triangular
## VCE method                     NN
##
## Number of Obs.                 45           41
## Eff. Number of Obs.            24           17
## Order est. (p)                  1            1
## Order bias  (q)                 2            2
## BW est. (h)              17685.552    17685.552
## BW bias (b)              29006.388    29006.388
## rho (h/b)                    0.610        0.610
## Unique Obs.                    45           41
##
## =======================================================================================
##         Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
## =======================================================================================
##   Conventional  -289.569  1491.053    -0.194     0.846 [-3211.979 , 2632.841]
## Bias-Corrected    38.177  1491.053     0.026     0.980 [-2884.232 , 2960.587]
##         Robust    38.177  1804.510     0.021     0.983 [-3498.597 , 3574.952]
## =======================================================================================
## NULL
##
## ==== 6. RDD PLOTS (rdplot) ====
```


RDD plot: log(price) vs mileage around 105000
log(price)
9.8
9.6
9.4
Mileage
75000
100000
125000
150000

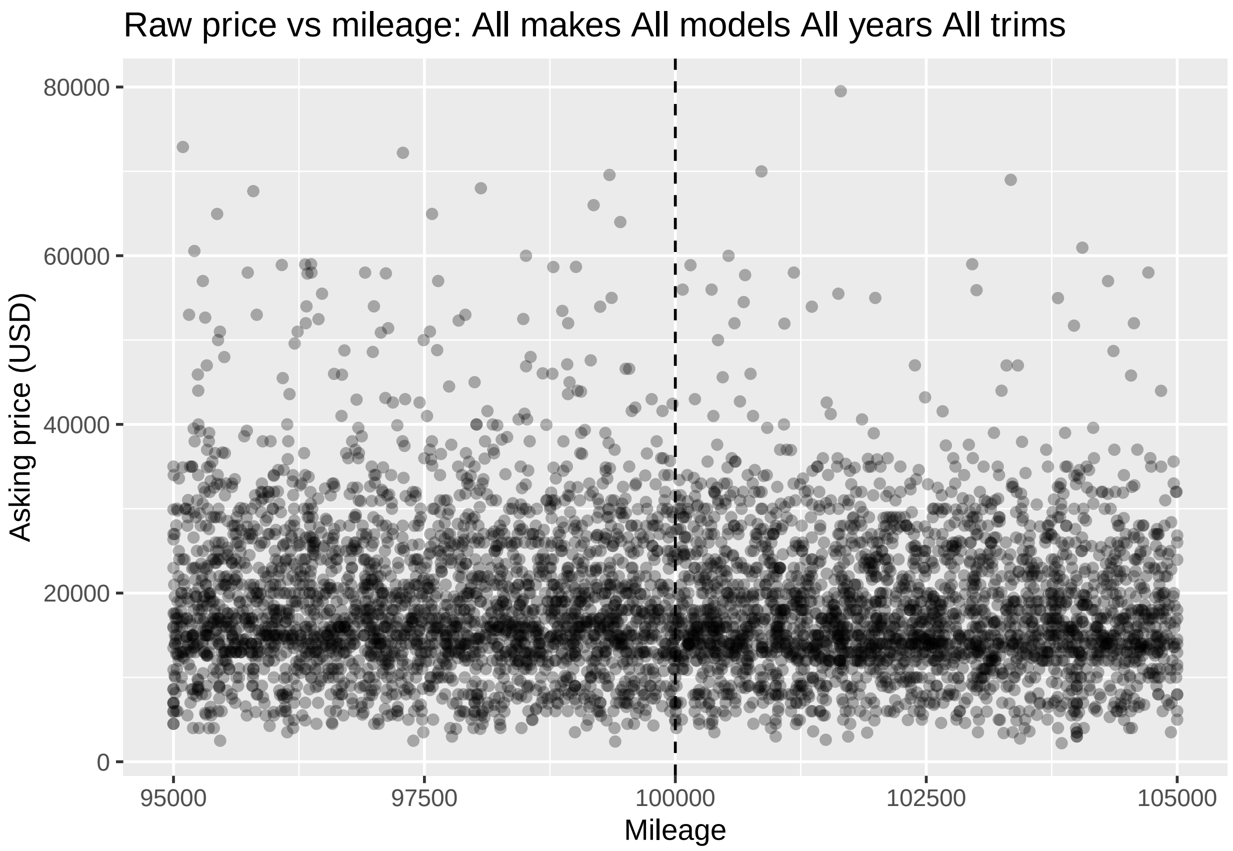


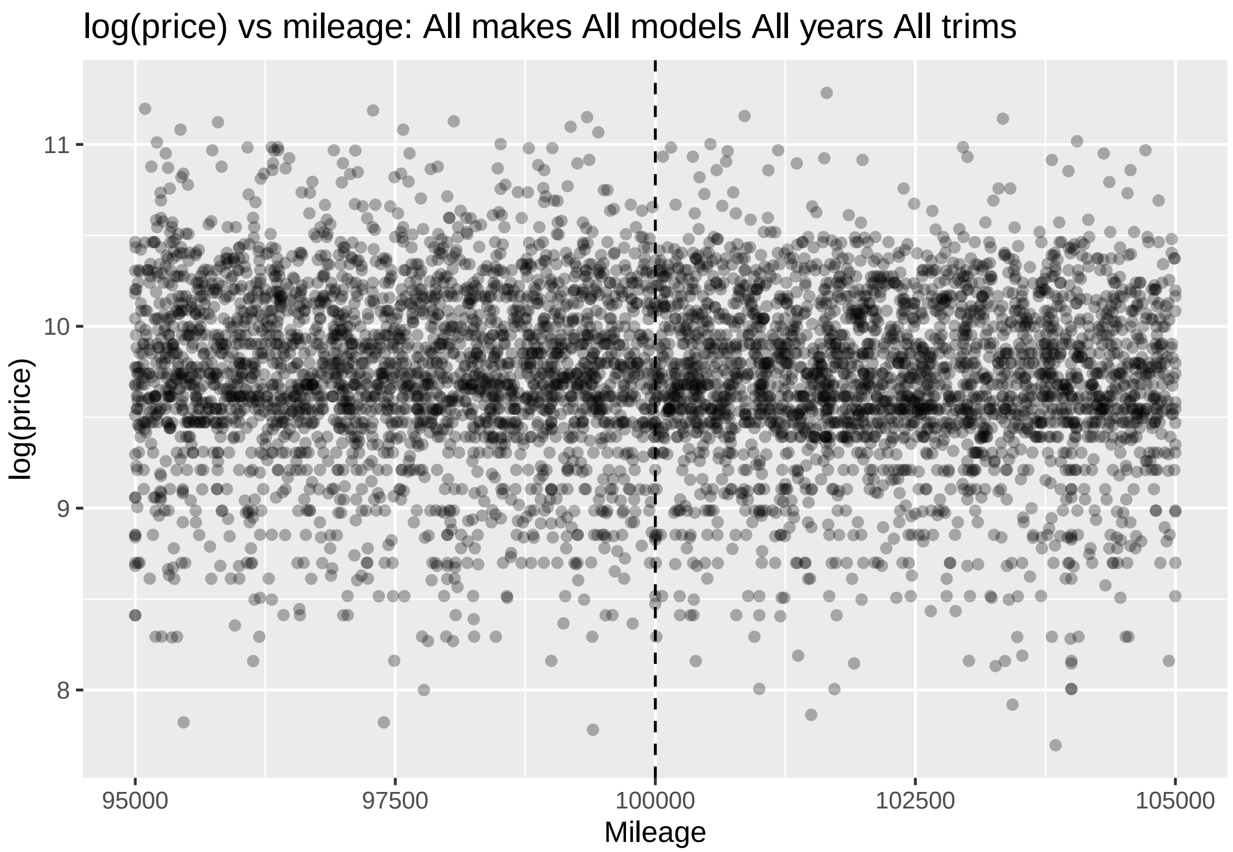
```
## rdplot objects created (printed in plotting window).
##
## ====================================
##    RDD PIPELINE COMPLETE
## ====================================
```

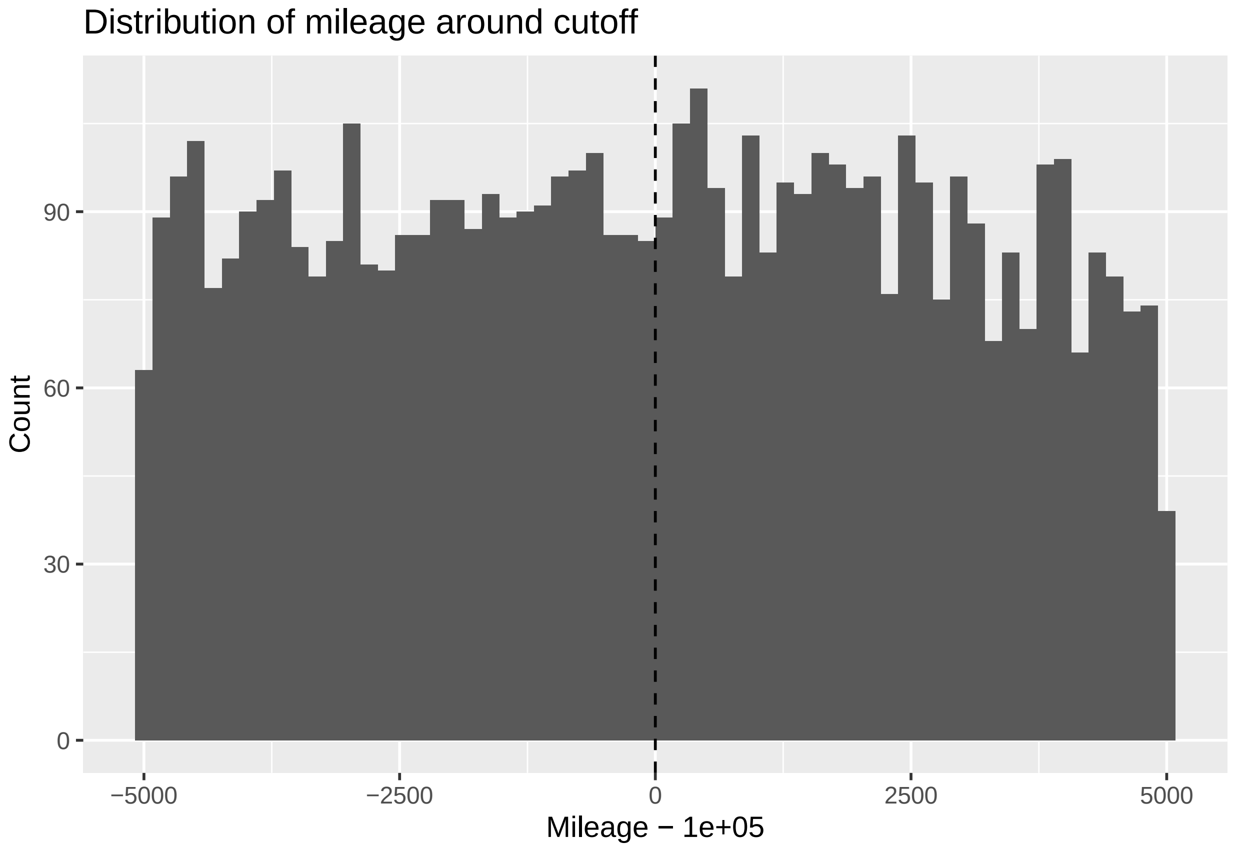
```
result_pooled_all <- run_rdd_for_vehicle(
  make0   = NULL,
  model0 = NULL,
  year0   = NULL,
  trim0   = NULL,
  cutoff_miles = 100000,
  max_window    = 5000
)
```

```
##
## ====================================
##    RDD PIPELINE START
## ====================================
##
## ---- 0. PARAMETERS ----
## make:   ALL
## model:  ALL
## year:   ALL
## trim:   ALL
## cutoff (miles):  1e+05
## max window (+/- miles):  5000
```

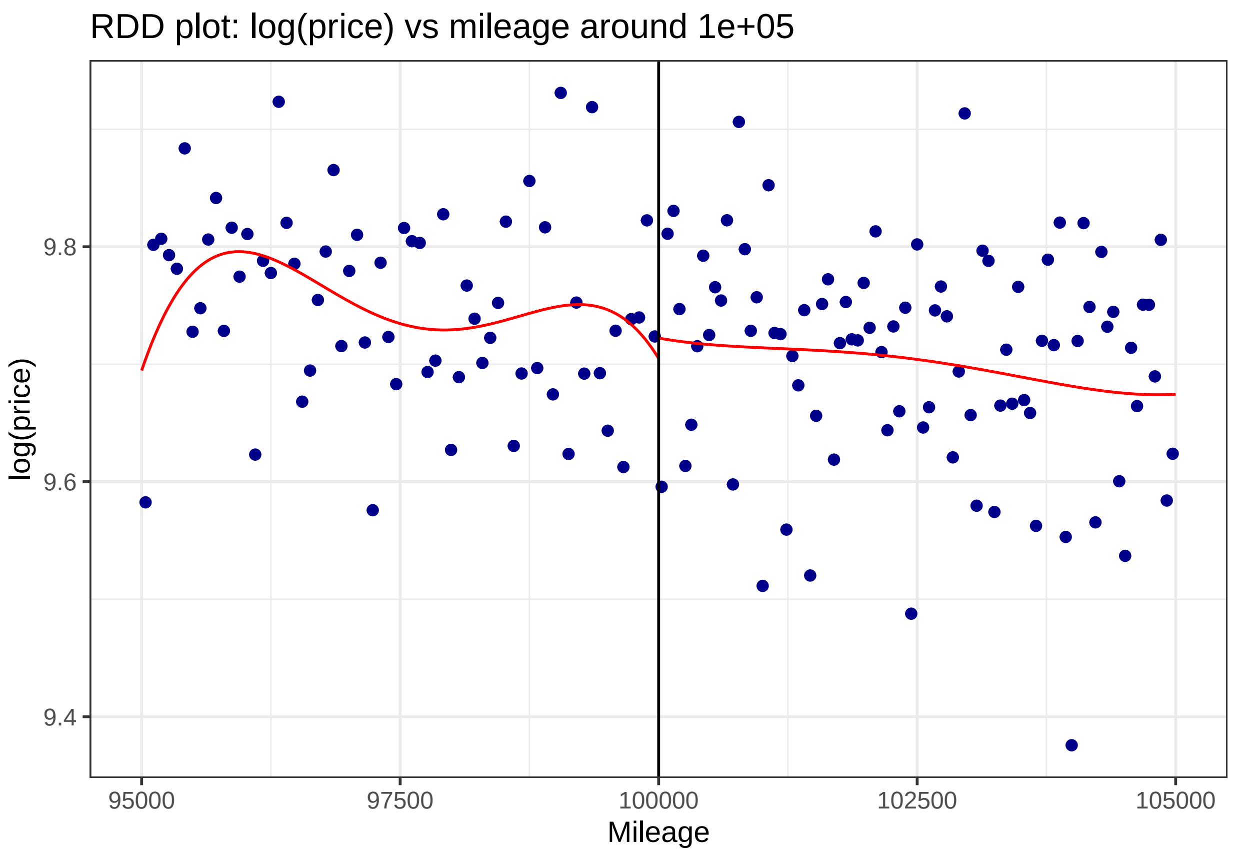
```
##
## ==== 1. CONNECTING TO DB & LOADING DATA ====
## Number of raw rows pulled:  167760
##
## ==== 2. CLEANING DATA & DEFINING RUNNING VARIABLE ====
## Rows after cleaning & windowing:  5263
## Summary of mileage_centered:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -5000.00 -2522.50   -24.00   -84.34  2317.00  5000.00
## Counts above vs below cutoff:
##
##    0    1
## 2648 2615
##
## ==== 3. BUILDING BASELINE COVARIATES (POOLED LOGIC) ====
## Using baseline covariates with formula:
##    ~make + model + factor(year) + trim
## Number of baseline covariate columns:  1214
##
## ==== 4. DESCRIPTIVE PLOTS ====
```



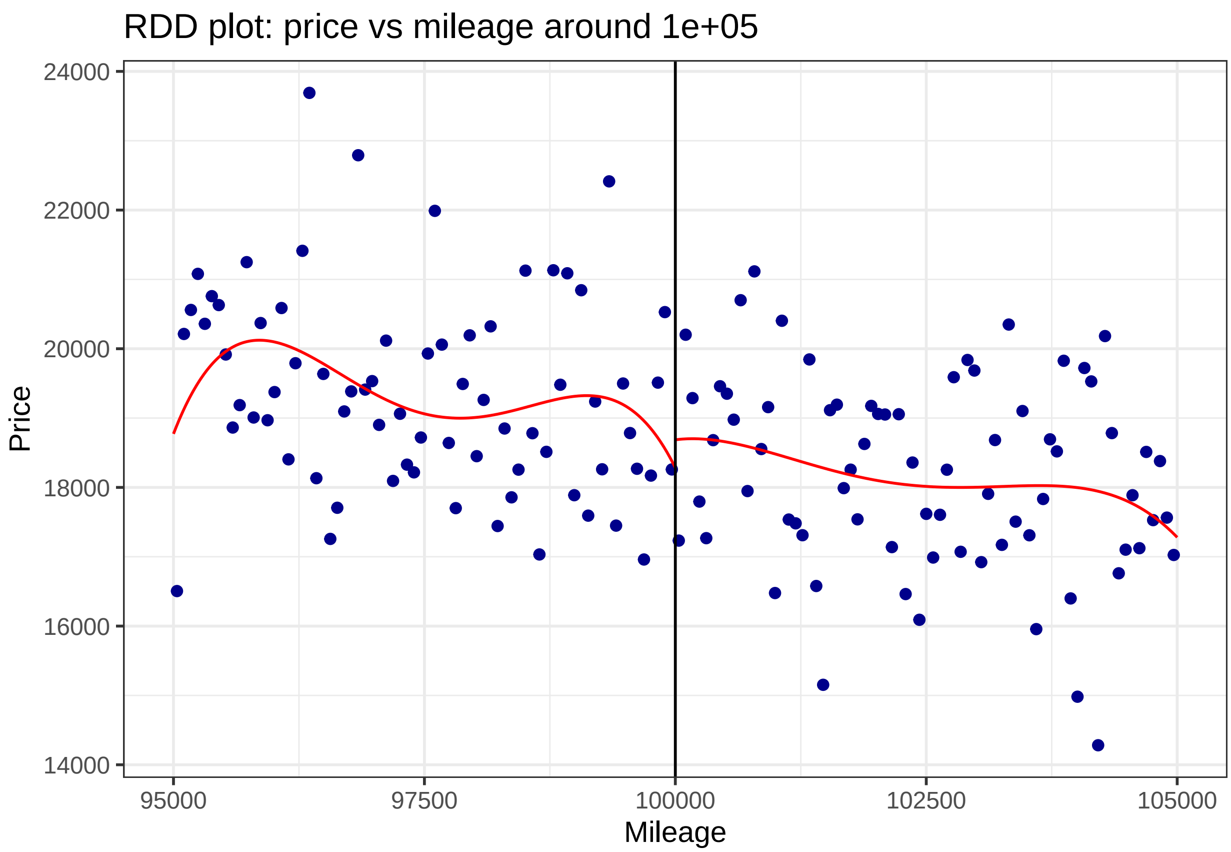

log(price) vs mileage: All makes All models All years All trims
log(price)
11
10
9
8
Mileage
95000
97500
100000
102500
105000



```
## Descriptive plots printed.
##
## ==== 5. MAIN SHARP RDD ESTIMATES ====
## 5.1 RDD on log(price)

## Warning in (function (y, x, c = NULL, fuzzy = NULL, deriv = NULL, p = NULL, :
## Multicollinearity issue detected in covs. Redundant covariates dropped.

## Warning in (function (y, x, c = NULL, fuzzy = NULL, deriv = NULL, p = NULL, :
## Mass points detected in the running variable.

## Covariate-adjusted Sharp RD estimates using local polynomial regression.
##
## Number of Obs.                 5263
## BW type                       mserd
## Kernel                   Triangular
## VCE method                       NN
##
## Number of Obs.                 2648         2615
## Eff. Number of Obs.             128          169
## Order est. (p)                    1            1
## Order bias  (q)                   2            2
## BW est. (h)                 284.865      284.865
## BW bias (b)                 433.915      433.915
## rho (h/b)                     0.656        0.656
## Unique Obs.                    2029         1980
##
## =============================================================================
##         Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
## =====================================================================================
##   Conventional    -0.160     0.029    -5.566     0.000    [-0.217 , -0.104]
## Bias-Corrected    -0.169     0.029    -5.874     0.000    [-0.226 , -0.113]
##         Robust    -0.169     0.047    -3.631     0.000    [-0.261 , -0.078]
## =====================================================================================
## NULL
##
## 5.2 RDD on price levels

## Warning in (function (y, x, c = NULL, fuzzy = NULL, deriv = NULL, p = NULL, :
## Multicollinearity issue detected in covs. Redundant covariates dropped.

## Warning in (function (y, x, c = NULL, fuzzy = NULL, deriv = NULL, p = NULL, :
## Mass points detected in the running variable.

## Covariate-adjusted Sharp RD estimates using local polynomial regression.
##
## Number of Obs.                 5263
## BW type                       mserd
## Kernel                   Triangular
## VCE method                       NN
##
## Number of Obs.                 2648         2615
## Eff. Number of Obs.             379          444
## Order est. (p)                    1            1
## Order bias  (q)                   2            2
## BW est. (h)                 742.203      742.203
## BW bias (b)                1170.142     1170.142
## rho (h/b)                     0.634        0.634
## Unique Obs.                    2029         1980
##
## =====================================================================================
##         Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
## =====================================================================================
##   Conventional   -41.378   321.663    -0.129     0.898  [-671.826 , 589.071]
## Bias-Corrected  -146.240   321.663    -0.455     0.649  [-776.688 , 484.208]
##         Robust  -146.240   390.234    -0.375     0.708  [-911.085 , 618.605]
## =====================================================================================
## NULL
##
## ==== 6. RDD PLOTS (rdplot) ====
## [1] "Mass points detected in the running variable."
```



```
## [1] "Mass points detected in the running variable."
```

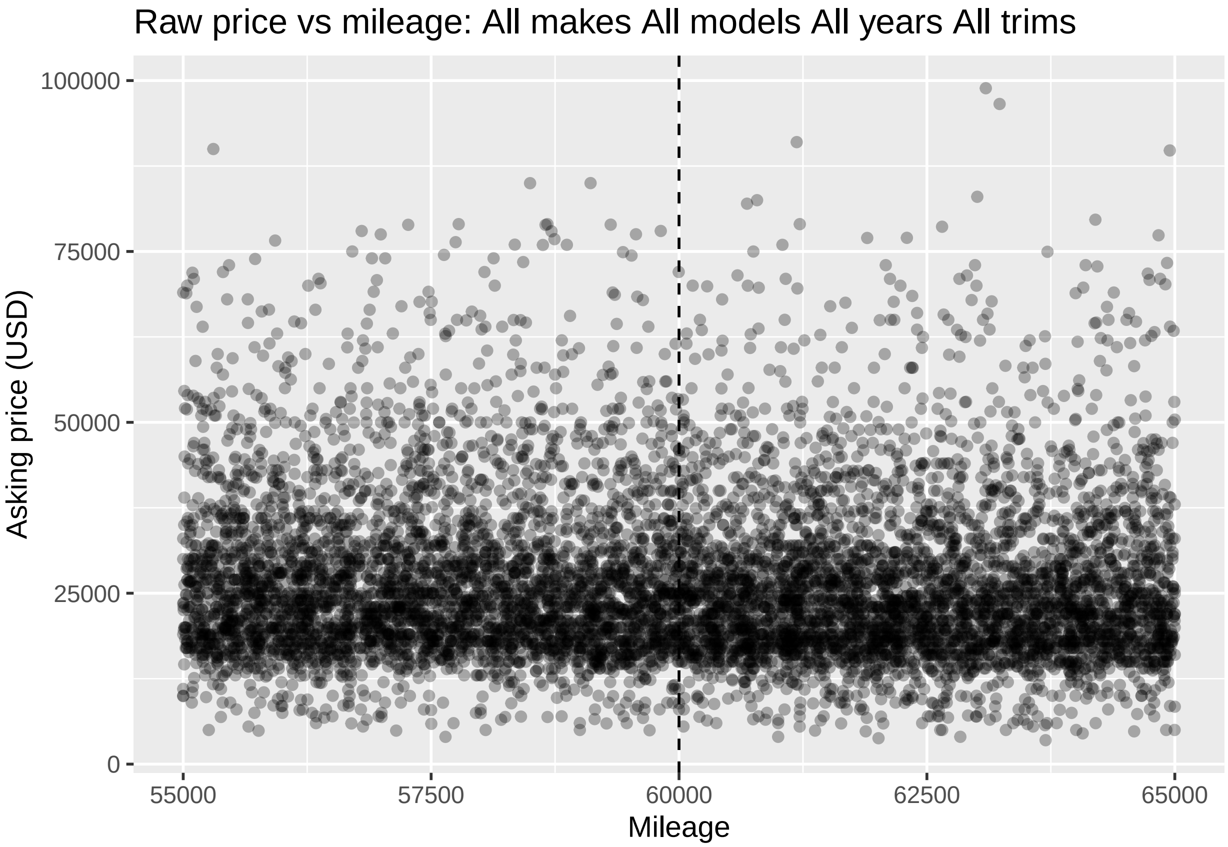


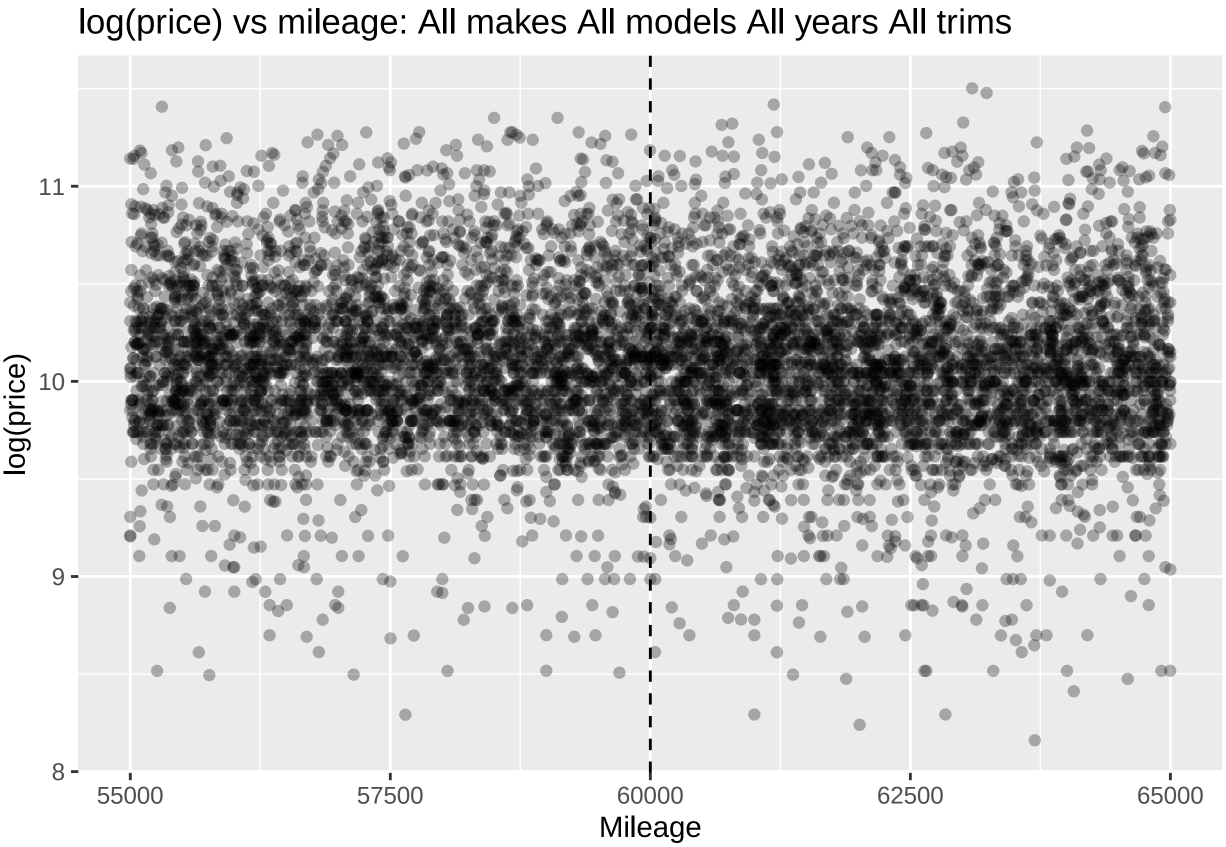
```
## rdplot objects created (printed in plotting window).
##
## ====================================
##    RDD PIPELINE COMPLETE
## ====================================
```

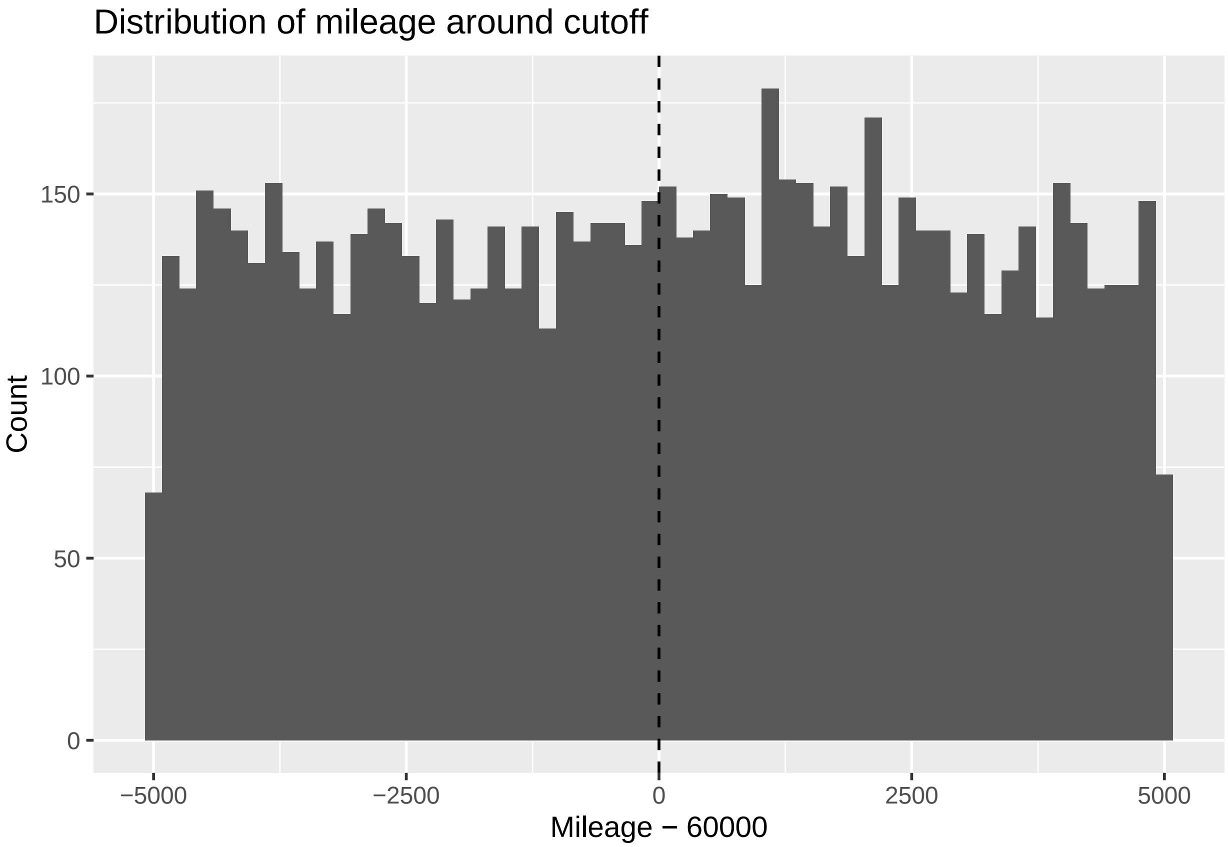
```
result_pooled_all_60k <- run_rdd_for_vehicle(
  make0   = NULL,
  model0 = NULL,
  year0   = NULL,
  trim0   = NULL,
  cutoff_miles = 60000,
  max_window    = 5000
)
```

```
##
## ====================================
##    RDD PIPELINE START
## ====================================
##
## ---- 0. PARAMETERS ----
## make:    ALL
## model:   ALL
## year:    ALL
## trim:    ALL
## cutoff (miles):  60000
## max window (+/- miles):  5000
```

```
##
## ==== 1. CONNECTING TO DB & LOADING DATA ====
## Number of raw rows pulled:  167760
##
## ==== 2. CLEANING DATA & DEFINING RUNNING VARIABLE ====
## Rows after cleaning & windowing:  8141
## Summary of mileage_centered:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -5000.00 -2487.00    78.00    15.67  2429.00  5000.00
## Counts above vs below cutoff:
##
##    0    1
## 3988 4153
##
## ==== 3. BUILDING BASELINE COVARIATES (POOLED LOGIC) ====
## Using baseline covariates with formula:
##    ~make + model + factor(year) + trim
## Number of baseline covariate columns:  1613
##
## ==== 4. DESCRIPTIVE PLOTS ====
```


Raw price vs mileage: All makes All models All years All trims


log(price) vs mileage: All makes All models All years All trims
log(price)
11
10
9
8
55000
57500
60000
62500
65000
Mileage

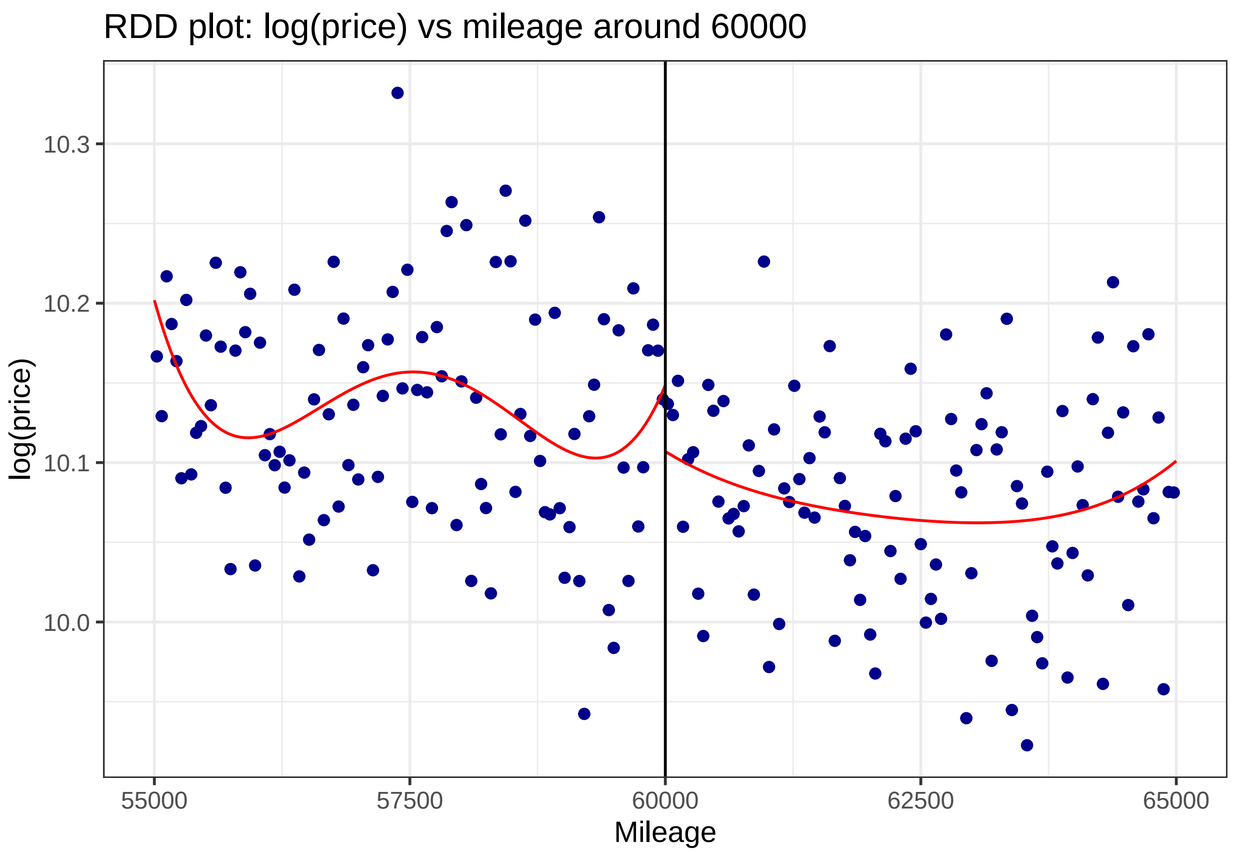


```
## Descriptive plots printed.
##
## ==== 5. MAIN SHARP RDD ESTIMATES ====
## 5.1 RDD on log(price)

## Warning in (function (y, x, c = NULL, fuzzy = NULL, deriv = NULL, p = NULL, :
## Multicollinearity issue detected in covs. Redundant covariates dropped.

## Warning in (function (y, x, c = NULL, fuzzy = NULL, deriv = NULL, p = NULL, :
## Mass points detected in the running variable.

## Covariate-adjusted Sharp RD estimates using local polynomial regression.
##
## Number of Obs.                 8141
## BW type                       mserd
## Kernel                   Triangular
## VCE method                       NN
##
## Number of Obs.                 3988         4153
## Eff. Number of Obs.             322          338
## Order est. (p)                    1            1
## Order bias  (q)                   2            2
## BW est. (h)                 390.255      390.255
## BW bias (b)                 592.781      592.781
## rho (h/b)                     0.658        0.658
## Unique Obs.                    2721         2773
##
## =============================================================================
##         Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
## =======================================================================================
##   Conventional     0.012     0.017     0.733     0.464    [-0.020 , 0.045]
## Bias-Corrected     0.048     0.017     2.878     0.004     [0.015 , 0.080]
##         Robust     0.048     0.033     1.450     0.147    [-0.017 , 0.113]
## =======================================================================================
## NULL
##
## 5.2 RDD on price levels
```

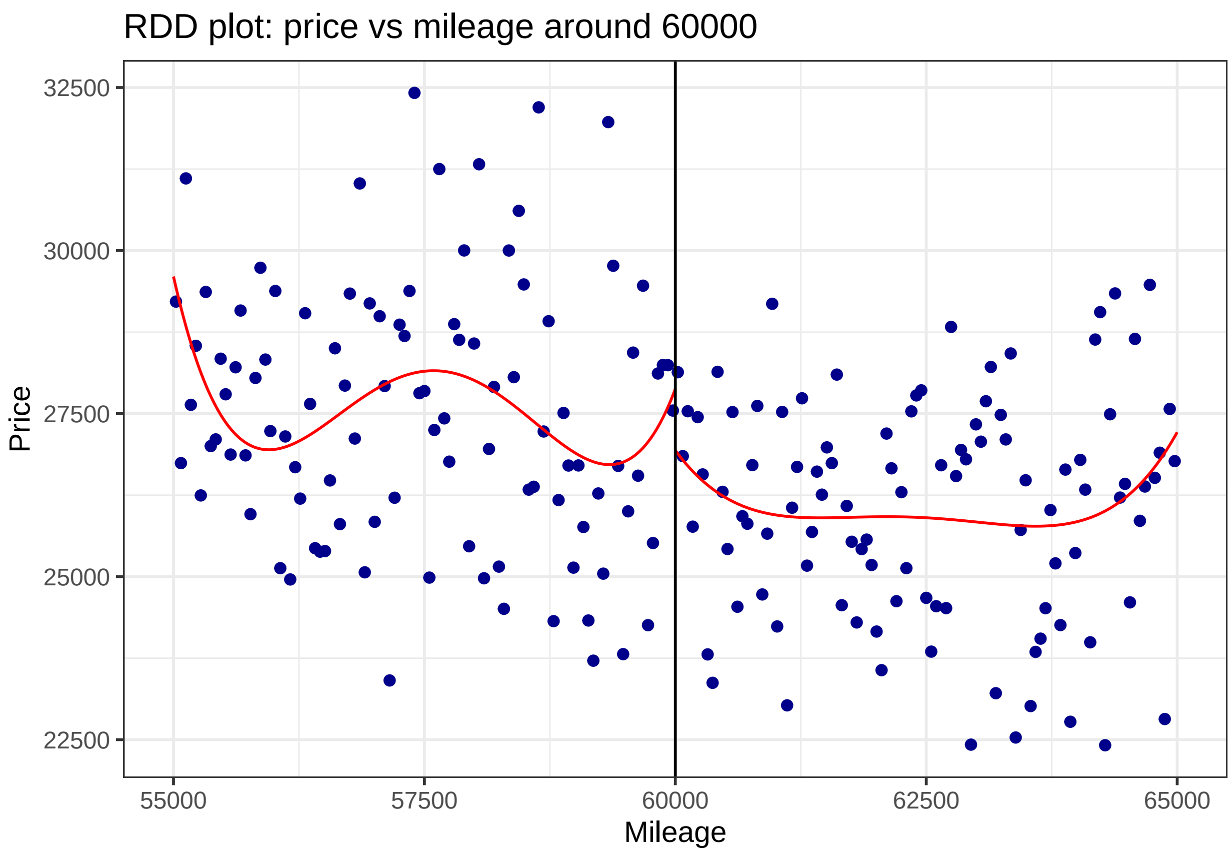
```
## Warning in (function (y, x, c = NULL, fuzzy = NULL, deriv = NULL, p = NULL, :
## Multicollinearity issue detected in covs. Redundant covariates dropped.
```

```
## Warning in (function (y, x, c = NULL, fuzzy = NULL, deriv = NULL, p = NULL, :
## Mass points detected in the running variable.
```

```
## Covariate-adjusted Sharp RD estimates using local polynomial regression.
##
## Number of Obs.                 8141
## BW type                       mserd
## Kernel                   Triangular
## VCE method                       NN
##
## Number of Obs.                 3988         4153
## Eff. Number of Obs.             671          710
## Order est. (p)                    1            1
## Order bias  (q)                   2            2
## BW est. (h)                 811.583      811.583
## BW bias (b)                1276.997     1276.997
## rho (h/b)                     0.636        0.636
## Unique Obs.                    2721         2773
##
## =======================================================================================
##         Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
## =======================================================================================
##   Conventional   196.476   316.158     0.621     0.534  [-423.182 , 816.135]
## Bias-Corrected   209.917   316.158     0.664     0.507  [-409.741 , 829.576]
##         Robust   209.917   378.947     0.554     0.580  [-532.804 , 952.639]
## =======================================================================================
## NULL
##
## ==== 6. RDD PLOTS (rdplot) ====
## [1] "Mass points detected in the running variable."
```



```
## [1] "Mass points detected in the running variable."
```

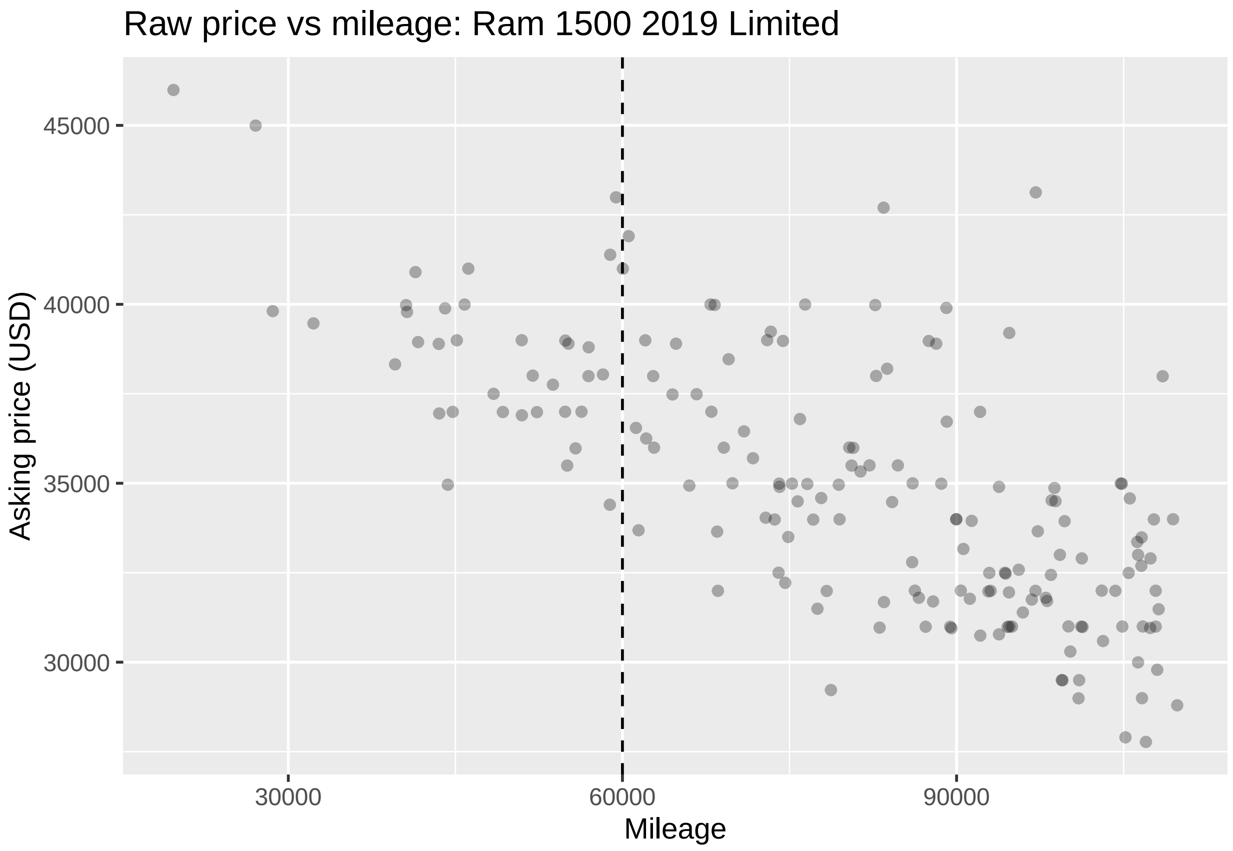


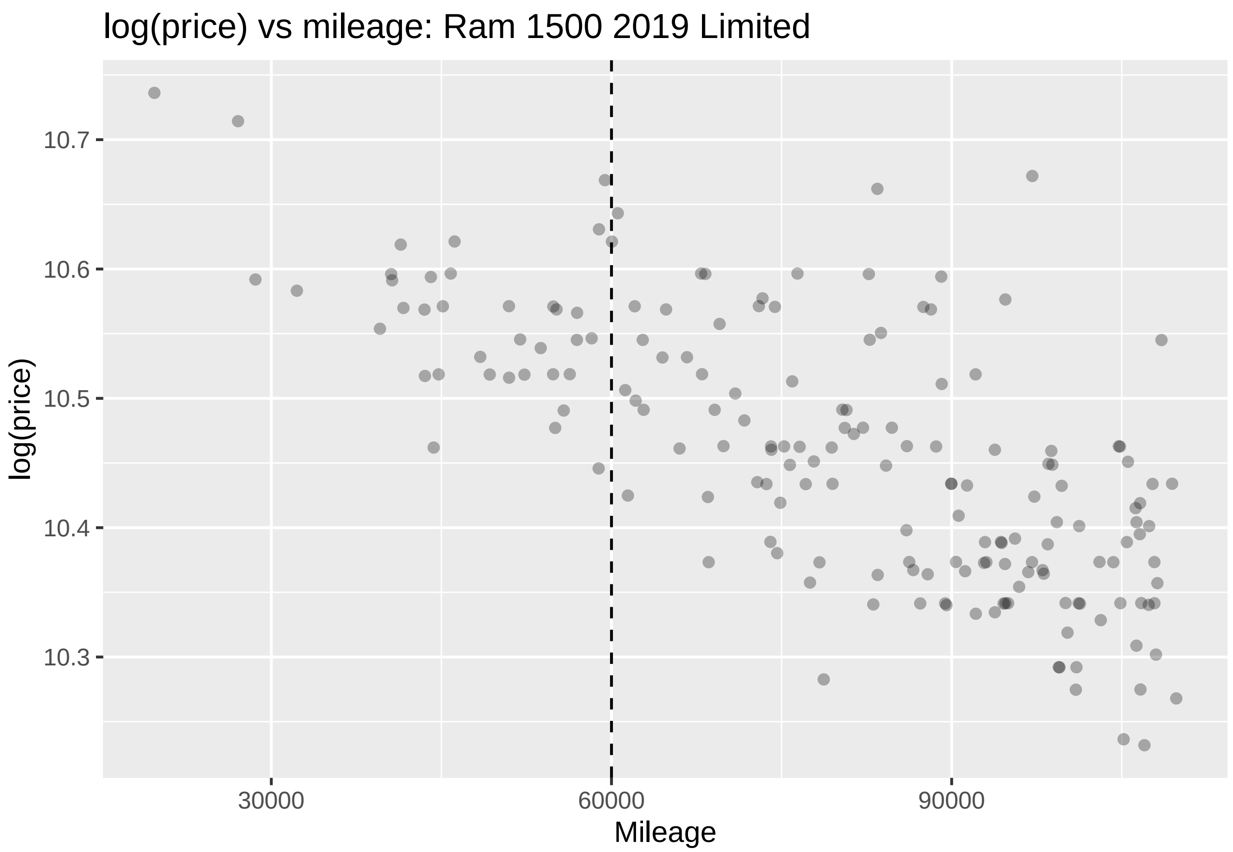
```
## rdplot objects created (printed in plotting window).
##
## ====================================
##    RDD PIPELINE COMPLETE
## ====================================
```

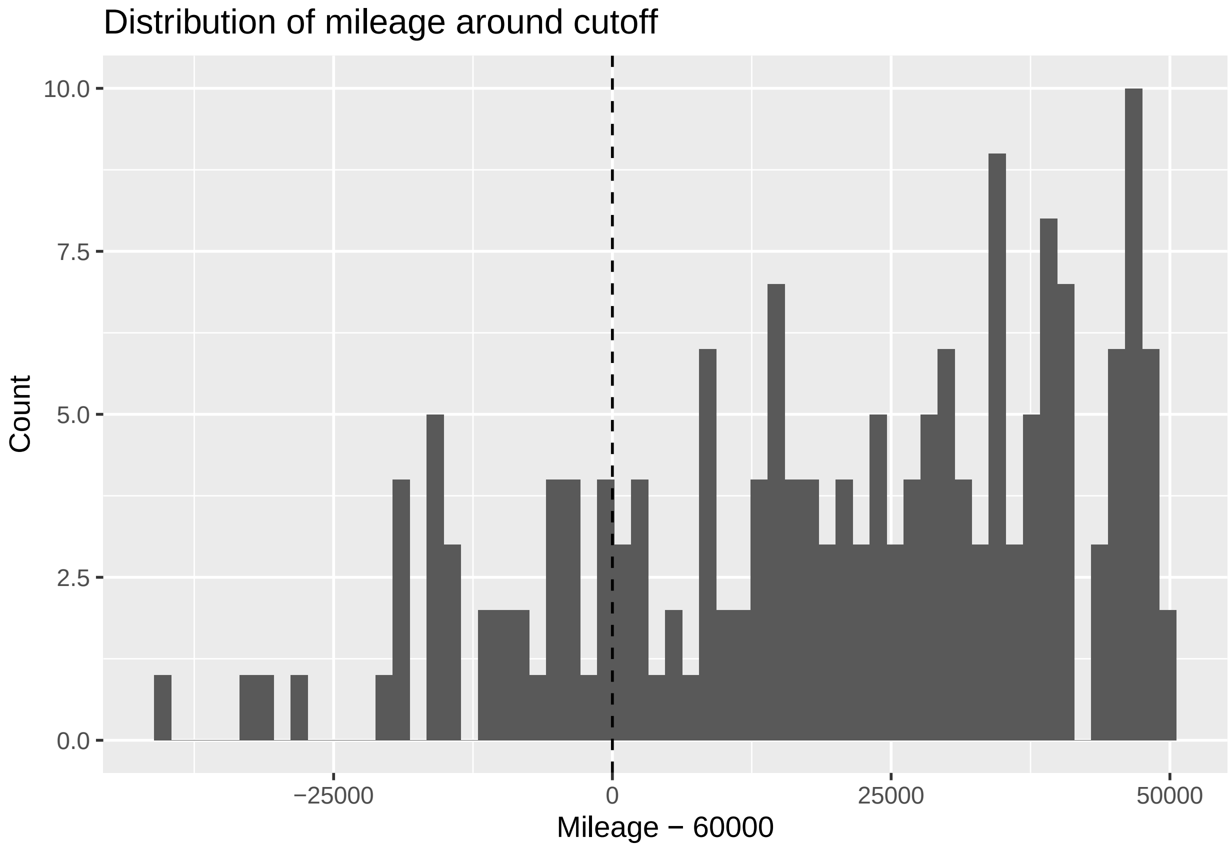
```
result_ram_2019_limited_60k <- run_rdd_for_vehicle(
  make0   = "Ram",
  model0 = "1500",
  year0   = 2019L,
  trim0   = "Limited",
  cutoff_miles = 60000,
  max_window    = 50000
)
```

```
##
## ====================================
##    RDD PIPELINE START
## ====================================
##
## ---- 0. PARAMETERS ----
## make:    Ram
## model:   1500
## year:    2019
## trim:    Limited
## cutoff (miles):  60000
## max window (+/- miles):  50000
```

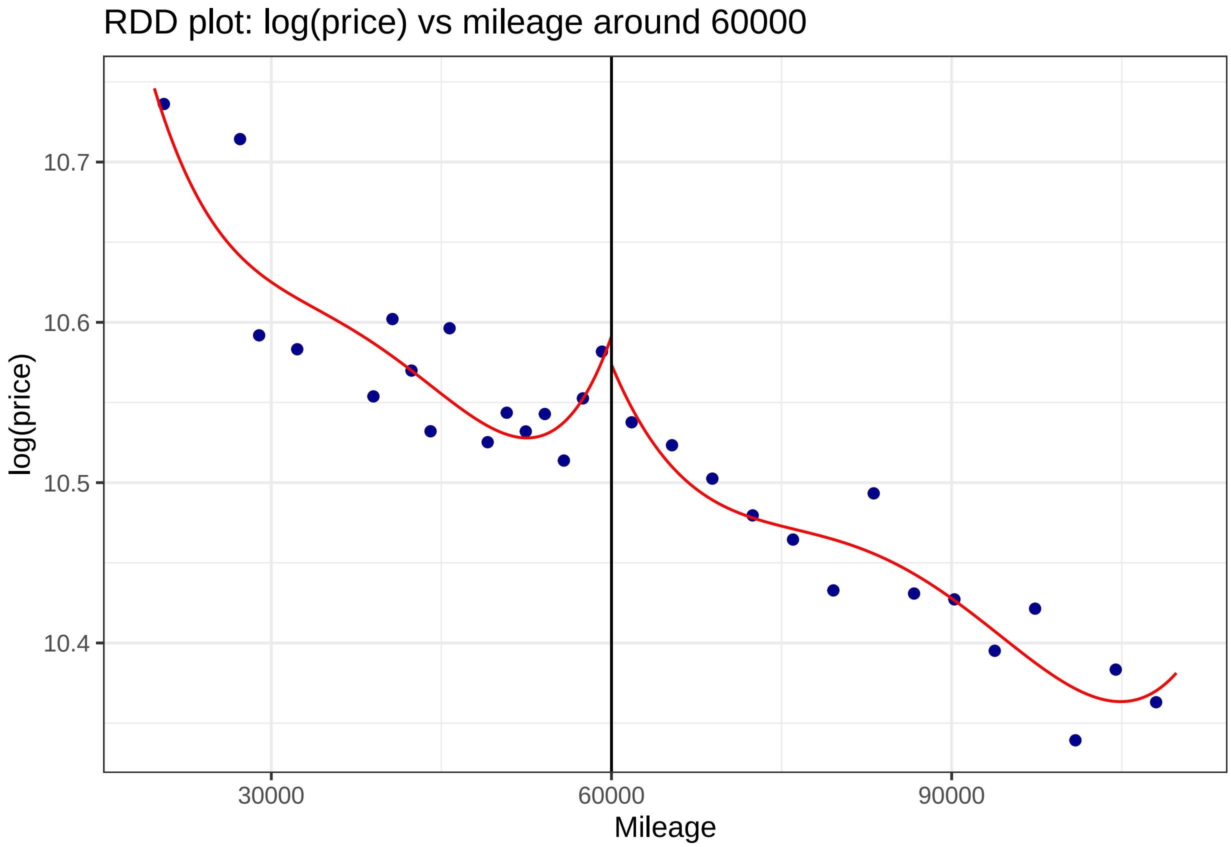
```
##
## ==== 1. CONNECTING TO DB & LOADING DATA ====
## Number of raw rows pulled:  218
##
## ==== 2. CLEANING DATA & DEFINING RUNNING VARIABLE ====
## Rows after cleaning & windowing:  176
## Summary of mileage_centered:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -40311    4081   23270   19797   37474   49808
## Counts above vs below cutoff:
##
##   0   1
##  36 140
##
## ==== 3. BUILDING BASELINE COVARIATES (POOLED LOGIC) ====
## All of make/model/year/trim fixed. No covariate adjustment used.
##
## ==== 4. DESCRIPTIVE PLOTS ====
```



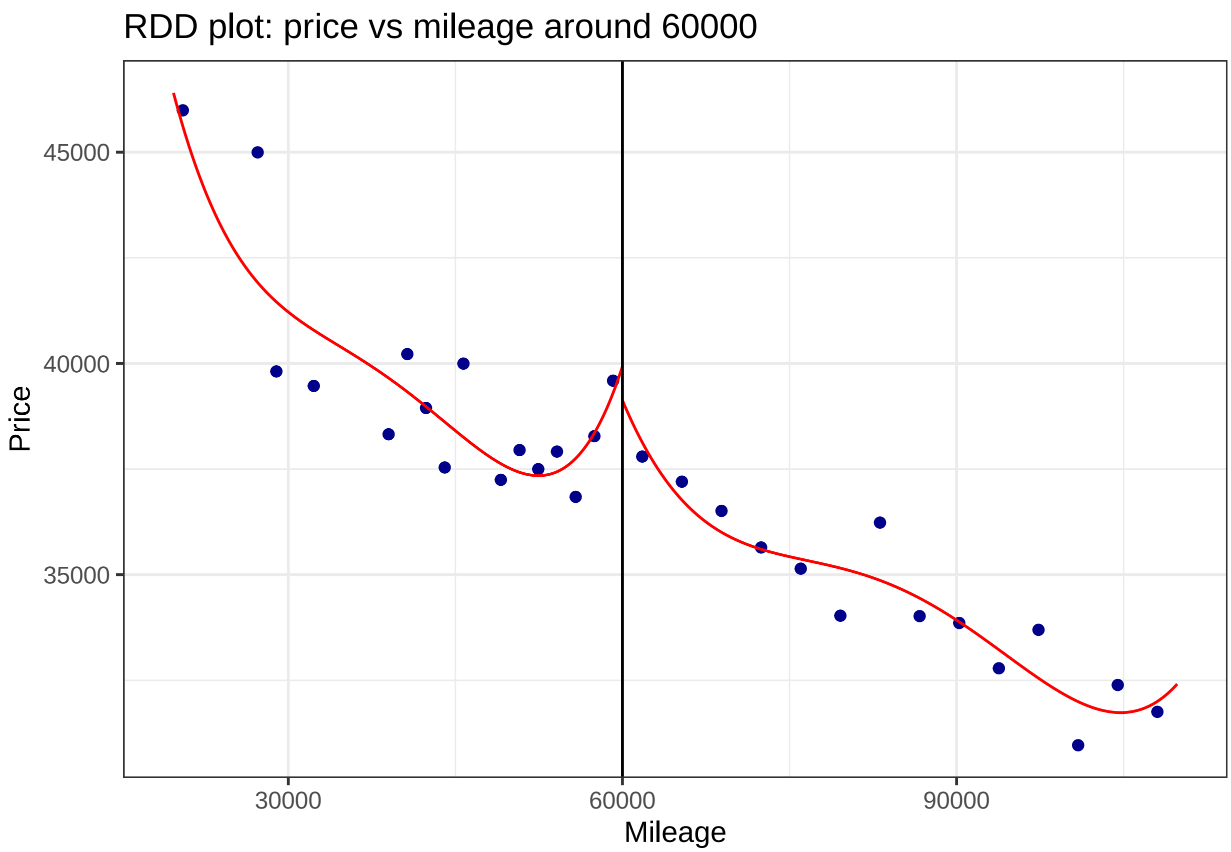

log(price) vs mileage: Ram 1500 2019 Limited
log(price)
10.7
10.6
10.5
10.4
10.3
Mileage
30000
60000
90000



```
## Descriptive plots printed.
##
## ==== 5. MAIN SHARP RDD ESTIMATES ====
## 5.1 RDD on log(price)
## Sharp RD estimates using local polynomial regression.
##
## Number of Obs.                  176
## BW type                       mserd
## Kernel                   Triangular
## VCE method                       NN
##
## Number of Obs.                   36          140
## Eff. Number of Obs.              15           17
## Order est. (p)                    1            1
## Order bias  (q)                   2            2
## BW est. (h)                8689.677     8689.677
## BW bias (b)               14132.863    14132.863
## rho (h/b)                     0.615        0.615
## Unique Obs.                      36          140
##
## ========================================================================================
##         Method     Coef. Std. Err.         z     P>|z|      [ 95% C.I. ]
## ========================================================================================
##   Conventional    -0.026     0.072    -0.359     0.720    [-0.167 , 0.115]
## Bias-Corrected    -0.032     0.072    -0.451     0.652    [-0.173 , 0.109]
##         Robust    -0.032     0.087    -0.374     0.708    [-0.202 , 0.137]
## ========================================================================================
## NULL
##
## 5.2 RDD on price levels
## Sharp RD estimates using local polynomial regression.
##
## Number of Obs.                  176
## BW type                       mserd
## Kernel                   Triangular
## VCE method                       NN
##
## Number of Obs.                   36          140
## Eff. Number of Obs.              15           17
## Order est. (p)                    1            1
## Order bias  (q)                   2            2
## BW est. (h)                8611.228     8611.228
## BW bias (b)               14083.294    14083.294
## rho (h/b)                     0.611        0.611
## Unique Obs.                      36          140
##
## ========================================================================================
##         Method     Coef. Std. Err.         z     P>|z|       [ 95% C.I. ]
## ========================================================================================
##   Conventional -1052.669  2778.388    -0.379     0.705 [-6498.209 , 4392.872]
## Bias-Corrected -1320.023  2778.388    -0.475     0.635 [-6765.564 , 4125.517]
##         Robust -1320.023  3341.269    -0.395     0.693 [-7868.790 , 5228.743]
## ========================================================================================
## NULL
##
## ==== 6. RDD PLOTS (rdplot) ====
```



RDD plot: log(price) vs mileage around 60000



```
## rdplot objects created (printed in plotting window).
##
## ====================================
##    RDD PIPELINE COMPLETE
## ====================================
```