MLDS 401: Homework 3

Arielle Weinstein, Jack Bailey, Junbo (Jacob) Lian, Veronica Lin

2025-10-01

```
# Install once in Console if missing:
# install.packages(c("ggplot2","dplyr","broom","sandwich","lmtest","knitr","ISLR2"))
need <- c("ggplot2","dplyr","broom","sandwich","lmtest","knitr","ISLR2")
miss <- setdiff(need, rownames(installed.packages()))
if(length(miss)) install.packages(miss, quiet = TRUE)
invisible(lapply(intersect(need, rownames(installed.packages())),
                 library, character.only = TRUE))

options(contrasts = c("contr.treatment", "contr.poly"))
```

```
# data loading
data_dir  <- "E:/mlds/MLDS 401 Machine Learning/course_files_export/Homework/HW3"
auto_path <- if (file.exists(file.path(data_dir, "auto.txt"))) file.path(data_dir, "auto.txt")
part_path <- if (file.exists(file.path(data_dir, "part.csv"))) file.path(data_dir, "part.csv")

# Load files
auto <- read.table(auto_path, header = TRUE, na.strings = c("NA","?",""))
part <- read.csv(part_path)

# Cleaning
names(auto) <- trimws(names(auto))
auto$origin <- factor(auto$origin, levels = 1:3, labels = c("US","Europe","Japan"))
auto$lmpg   <- log(auto$mpg)
auto$lw     <- log(auto$weight)

# Sanity checks
stopifnot(all(c("mpg","weight","year","origin") %in% names(auto)))
stopifnot(all(c("y","x","tx","wc") %in% names(part)))

# Variables for Q6
d <- transform(part,
  tx  = as.integer(tx),
  ly  = log(y + 1),
  lx  = log(x + 1),
  lwc = log(wc + 1)
)
```

1 1. (JWHT 3.9) auto Data

```
# 1(a): Origin breakdown
tab_origin  <- table(auto$origin, useNA = "ifany")
prop_origin <- prop.table(tab_origin)
knitr::kable(
  as.data.frame.matrix(rbind(Freq = tab_origin, Prop = round(prop_origin,3))),
  caption = "Origin: frequency and proportion"
)
```

Table 1: Origin: frequency and proportion

	US	Europe	Japan
Freq	248.000	70.000	79.000
Prop	0.625	0.176	0.199

```
# 1(b): OLS with basic diagnostics (complete cases to avoid NA plot issues)
auto_cc <- auto[complete.cases(auto[c("mpg","origin","weight","year")]), ]

m1 <- lm(mpg ~ origin + weight + year, data = auto_cc)
tidy_m1   <- broom::tidy(m1, conf.int = TRUE)
glance_m1 <- broom::glance(m1)
knitr::kable(tidy_m1, digits = 4, caption = "Model 1: OLS for mpg")
```

Table 2: Model 1: OLS for mpg

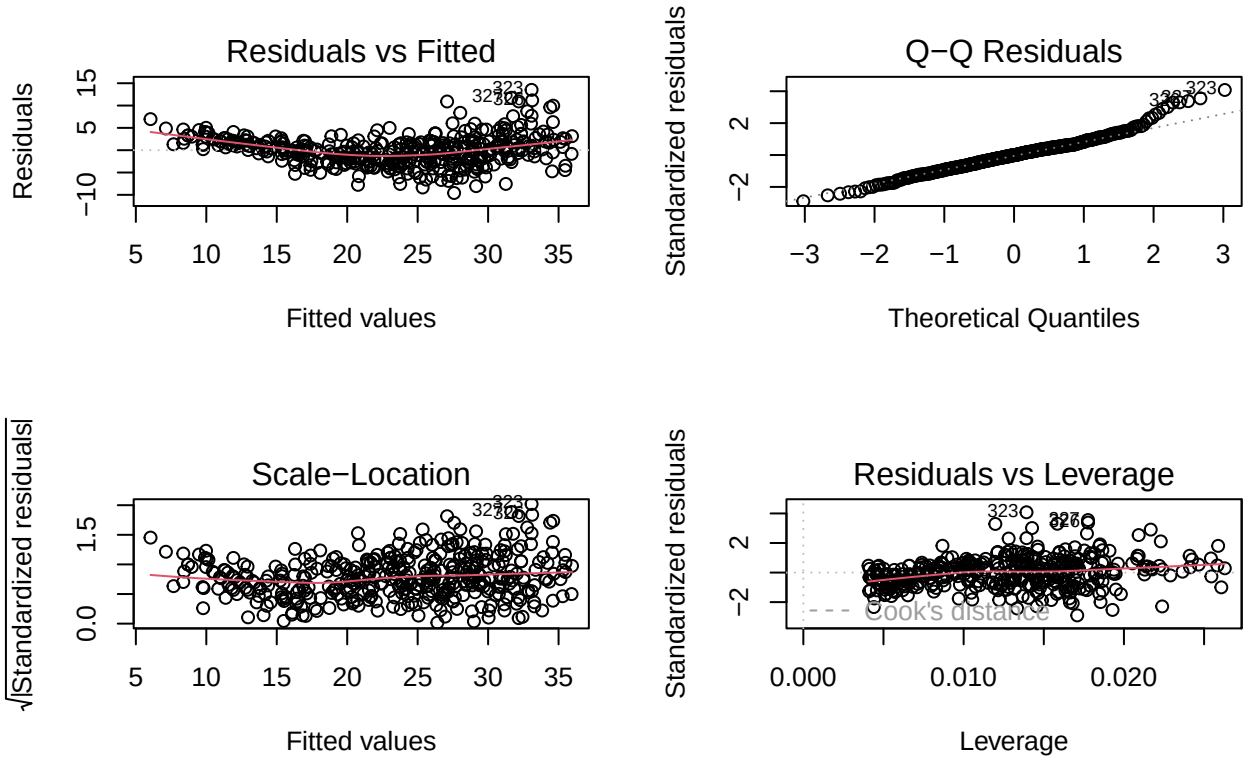
term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	-18.5853	3.9684	-4.6833	0	-26.3873	-10.7833
originEurope	2.0965	0.5110	4.1026	0	1.0918	3.1012
originJapan	2.2030	0.5157	4.2714	0	1.1890	3.2169
weight	-0.0059	0.0003	-22.9087	0	-0.0064	-0.0054
year	0.7740	0.0481	16.0748	0	0.6793	0.8686

```
knitr::kable(glance_m1, digits = 4, caption = "Model 1: fit statistics")
```

Table 3: Model 1: fit statistics

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
0.8201	0.8183	3.3362	446.7528	0	4	-1039.116	2090.232	2114.136	4362.945	392	397

```r
par(mfrow = c(2,2)); plot(m1); par(mfrow = c(1,1))
```




```r
# 1(c): Log-linear with quadratic year
auto_cc2 <- auto[complete.cases(auto[c("lmpg","origin","lw","year")]), ]

m2 <- lm(lmpg ~ origin + lw + year + I(year^2), data = auto_cc2)
tidy_m2   <- broom::tidy(m2, conf.int = TRUE)
glance_m2 <- broom::glance(m2)
knitr::kable(tidy_m2, digits = 4, caption = "Model 2: OLS for log(mpg)")
```

Table 4: Model 2: OLS for log(mpg)

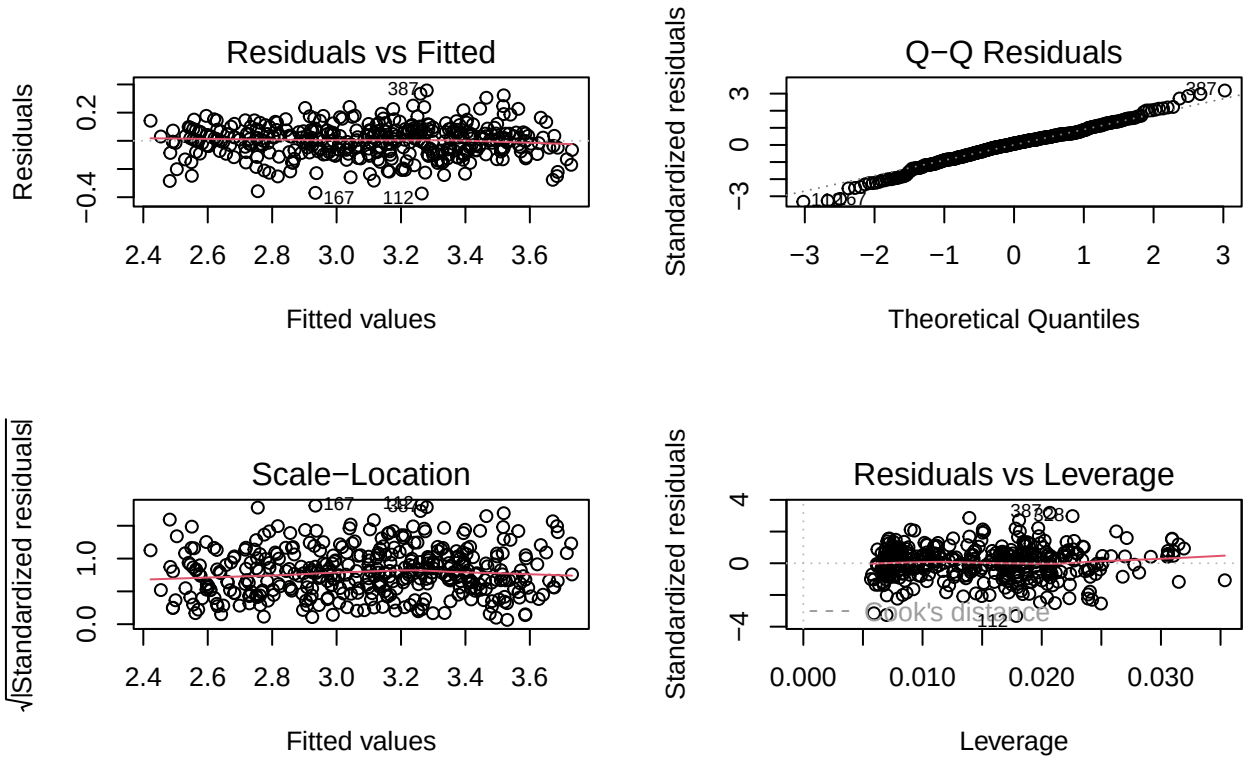
term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	18.4693	2.6834	6.8828	0.0000	13.1936	23.7450
originEurope	0.0668	0.0176	3.7908	0.0002	0.0322	0.1015
originJapan	0.0320	0.0179	1.7823	0.0755	-0.0033	0.0672
lw	-0.8750	0.0270	-32.3618	0.0000	-0.9282	-0.8219
year	-0.2560	0.0712	-3.5946	0.0004	-0.3960	-0.1160
I(year^2)	0.0019	0.0005	4.0648	0.0001	0.0010	0.0028

```
knitr::kable(glance_m2, digits = 4, caption = "Model 2: fit statistics")
```

Table 5: Model 2: fit statistics

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
0.8898	0.8884	0.1136	631.7314	0	5	303.2463	-592.4926	-564.6051	5.0449	391	397

```
par(mfrow = c(2,2)); plot(m2); par(mfrow = c(1,1))
```

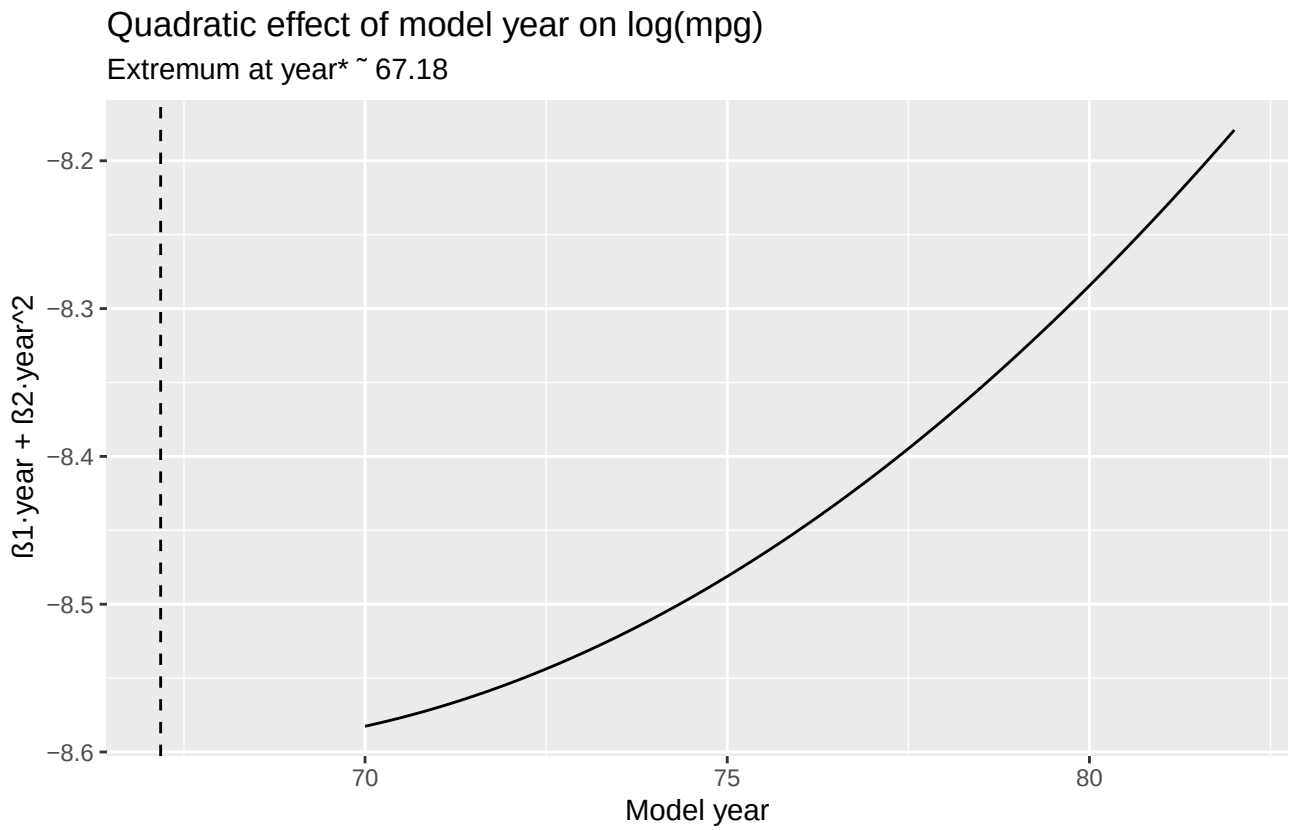



```
# 1(d): Extremum of the quadratic effect and visualization
co <- coef(m2); b1 <- unname(co[["year"]]); b2 <- unname(co[["I(year^2)"]])
year_star <- -b1/(2*b2)

grid <- data.frame(year = seq(min(auto_cc2$year, na.rm=TRUE),
                              max(auto_cc2$year, na.rm=TRUE), by = 0.1))
grid$effect <- b1*grid$year + b2*grid$year^2

ggplot(grid, aes(year, effect)) +
```

```
  geom_line() +
  geom_vline(xintercept = year_star, linetype = 2) +
  labs(title = "Quadratic effect of model year on log(mpg)",
       subtitle = paste0("Extremum at year*   ", round(year_star, 2)),
       x = "Model year", y = " 1·year +  2·year^2")
```




```
year_star
```

```
## [1] 67.17812
```

```
## The coefficient on 'weight' is negative in both levels and logs, indicating higher weight is
```

2 2. (ACT 3.2) Hat Matrix for SLR

```
# Q2: Hat matrix for simple linear regression (with intercept)

# Use 'year' with complete cases
stopifnot("year" %in% names(auto))
```

```r
x <- auto$year
x <- x[!is.na(x)]
n <- length(x); stopifnot(n >= 2)

# Design matrix: intercept + x
X <- cbind(1, x)

# Definition: H = X (X'X)^{-1} X'
XtX  <- t(X) %*% X
H    <- X %*% solve(XtX) %*% t(X)

# Closed form: h_ij = 1/n + (x_i - xbar)(x_j - xbar)/Sxx
xbar <- mean(x)
Sxx  <- sum((x - xbar)^2); stopifnot(Sxx > 0)
H2   <- outer(x, x, function(xi, xj) 1/n + ((xi - xbar)*(xj - xbar))/Sxx)

# Diagnostics
checks <- c(
  `max|H - H_closed|` = max(abs(H - H2)),
  symmetry            = max(abs(H - t(H))),
  idempotent          = max(abs(H %*% H - H)),
  `trace(H)=p=2`      = abs(sum(diag(H)) - 2),
  `row sums = 1`      = max(abs(rowSums(H) - 1)),
  `col sums = 1`      = max(abs(colSums(H) - 1))
)
knitr::kable(as.data.frame(t(checks)), digits = 6,
             caption = "Hat matrix properties (  0 except exact traces/sums).")
```

Table 6: Hat matrix properties (0 except exact traces/sums).

max|H - H_closed|	symmetry	idempotent	trace(H)=p=2	row sums = 1	col sums = 1
0	0	0	0	0	0

```r
# Leverages and closed-form agreement
lev      <- diag(H)
lev_theo <- 1/n + (x - xbar)^2 / Sxx
lev_err  <- max(abs(lev - lev_theo))

knitr::kable(
  data.frame(
    `max|h_ii - closed form|` = lev_err,
    `mean(h_ii)`              = mean(lev),
    `sum(h_ii)=trace(H)`      = sum(lev)
  ),
```

```
  digits = 6,
  caption = "Leverage diagnostics."
)
```

Table 7: Leverage diagnostics.

max.h_ii...closed.form.	mean.h_ii.	sum.h_ii..trace.H.
0	0.005038	2

```
# Six-number summary (avoid summary()/dimnames issues)
lev_stats <- c(
  Min  = min(lev),
  Q1   = as.numeric(quantile(lev, 0.25)),
  Med  = median(lev),
  Mean = mean(lev),
  Q3   = as.numeric(quantile(lev, 0.75)),
  Max  = max(lev)
)
knitr::kable(as.data.frame(t(lev_stats)), digits = 6,
             caption = "Leverage distribution (h_ii).")
```

Table 8: Leverage distribution (h_ii).

Min	Q1	Med	Mean	Q3	Max
0.002519	0.003257	0.004194	0.005038	0.007146	0.009207

3 3. (ACT 3.6) GLS Mean and Covariance (Illustration)

```
# Q3: GLS mean & covariance - illustration with diagonal Sigma (WLS=GLS)

# Preconditions
stopifnot(all(c("mpg","weight") %in% names(auto)))

# Complete cases
dat <- subset(auto, !is.na(mpg) & !is.na(weight))
stopifnot(nrow(dat) >= 3)

# Design and response
yg <- dat$mpg
Xg <- cbind(1, dat$weight)

# Heteroskedastic pattern: var proportional to weight^2 (illustrative)
```

```r
wvar_raw <- (dat$weight / mean(dat$weight))^2
wvar <- pmax(wvar_raw, .Machine$double.eps)

# Σ = diag(wvar), Σ^{-1} = diag(1/wvar)
Winv <- diag(1 / wvar)

# GLS estimator: beta = (X' Σ^{-1} X)^{-1} X' Σ^{-1} y
XtWinvX <- t(Xg) %*% Winv %*% Xg
XtWinvy <- t(Xg) %*% Winv %*% yg
beta_gls <- solve(XtWinvX, XtWinvy)
rownames(beta_gls) <- c("(Intercept)","weight")

# Covariance: Var(beta_hat) = sigma^2 * (X' Σ^{-1} X)^{-1}
e_gls   <- yg - Xg %*% beta_gls
rss_gls <- as.numeric(t(e_gls) %*% Winv %*% e_gls)
p       <- ncol(Xg); n <- nrow(Xg)
sigma2_hat <- rss_gls / (n - p)
cov_gls <- sigma2_hat * solve(XtWinvX)

# Present coefficients & SEs
res_gls <- data.frame(
  term     = rownames(beta_gls),
  estimate = as.numeric(beta_gls),
  SE       = sqrt(diag(cov_gls))
)
knitr::kable(res_gls, digits = 6,
             caption = "GLS estimates and approximate SEs (diagonal Σ).")
```

Table 9: GLS estimates and approximate SEs (diagonal Σ).

term	estimate	SE
(Intercept)	48.333105	0.905422
weight	-0.008382	0.000342

```r
# Sanity check: WLS equals GLS when Σ is diagonal
wls <- lm(yg ~ Xg[,2], weights = 1/wvar)
res_wls <- data.frame(term = names(coef(wls)), estimate = as.numeric(coef(wls)))
knitr::kable(res_wls, digits = 6,
             caption = "WLS coefficients (should match GLS under diagonal Σ).")
```

Table 10: WLS coefficients (should match GLS under diagonal Σ).

term	estimate
(Intercept)	48.333105
Xg[, 2]	-0.008382

```
# Show (X' Σ^{-1} X)^{-1} (information matrix inverse up to sigma^2)
inv_info <- solve(XtWinvX)
knitr::kable(round(inv_info, 6),
             caption = "(X' Σ^{-1} X)^{-1} - information matrix inverse (up to  ^2).")
```

Table 11: (X’ Σ^{-1} X)^{-1} — information matrix inverse (up to ^2).

0.028888	-1.1e-05
-0.000011	0.0e+00

4 4. WLS Intuition with n = 2

```
sig1 <- 4; sig2 <- 1
w1 <- (1/sig1^2) / (1/sig1^2 + 1/sig2^2); w2 <- 1 - w1
knitr::kable(t(c(w1 = w1, w2 = w2)), digits = 6,
             caption = "Optimal unbiased weights")
```

Table 12: Optimal unbiased weights

w1	w2
0.058824	0.941176

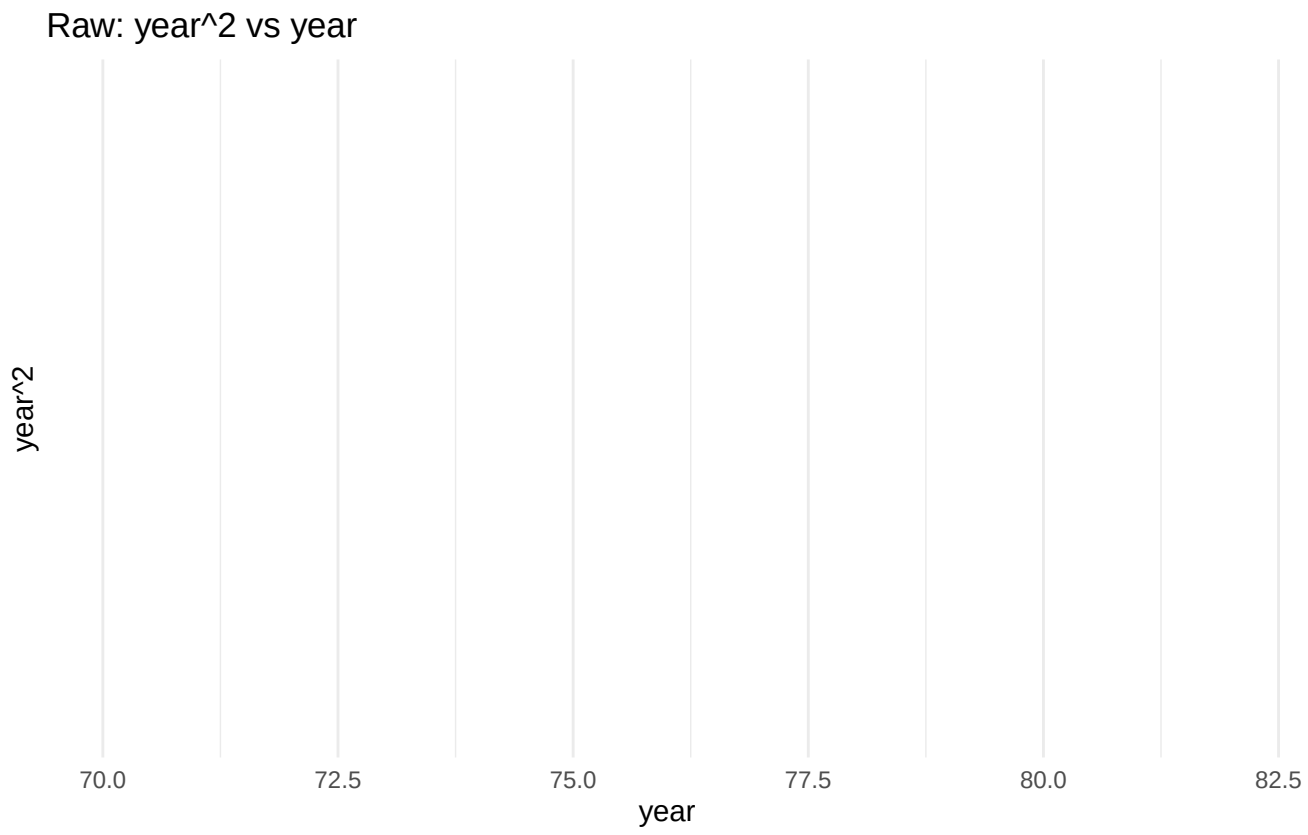
5 5. Quadratic Regression: Uncentered vs Centered Equivalence

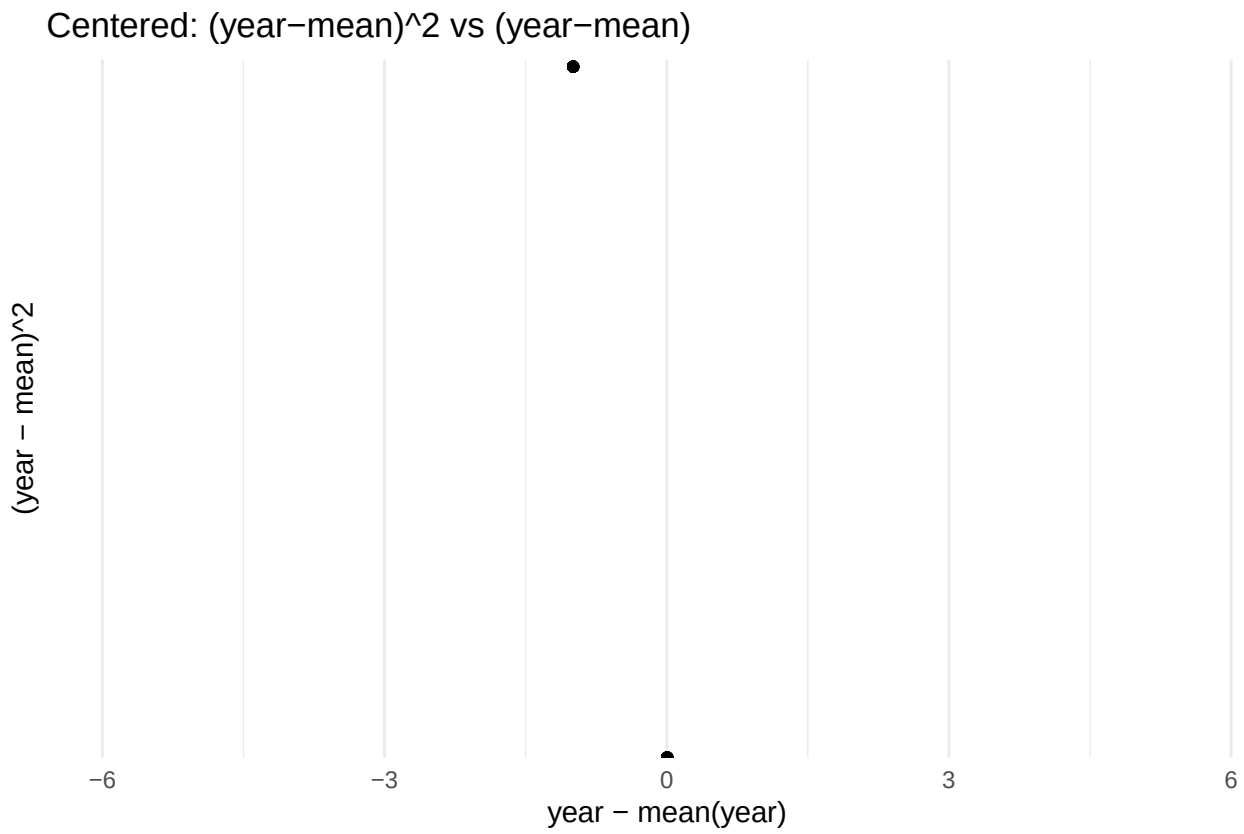
```
# Uncentered
fit_unc  <- lm(mpg ~ year + I(year^2), data = auto)
coef_unc <- coef(fit_unc)

# Correlation before centering
cor_raw <- cor(auto$year, auto$year^2)
```

```r
# Centering
xbar <- mean(auto$year, na.rm = TRUE)
auto$yc <- auto$year - xbar
cor_cen <- cor(auto$yc, auto$yc^2)

# Visual
g1 <- ggplot(auto, aes(year, I(year^2))) + geom_point(alpha=.6) + theme_minimal() +
  labs(title="Raw: year^2 vs year", x="year", y="year^2")
g2 <- ggplot(auto, aes(yc, I(yc^2))) + geom_point(alpha=.6) + theme_minimal() +
  labs(title="Centered: (year-mean)^2 vs (year-mean)", x="year - mean(year)", y="(year - mean)
g1; g2
```






```r
# Centered fit and mapping back
fit_cen  <- lm(mpg ~ yc + I(yc^2), data = auto)
coef_cen <- coef(fit_cen)

g0 <- coef_cen["(Intercept)"]; g1c <- coef_cen["yc"]; g2c <- coef_cen["I(yc^2)"]
beta0_hat <- g0 - g1c*xbar + g2c*xbar^2
beta1_hat <- g1c - 2*g2c*xbar
beta2_hat <- g2c

eq_table <- rbind(
  uncentered            = coef_unc,
  recovered_from_center = c("(Intercept)"=beta0_hat, "year"=beta1_hat, "I(year^2)"=beta2_hat)
)
knitr::kable(eq_table, digits = 6, caption = "Uncentered vs recovered-from-centered coefficient
```

Table 13: Uncentered vs recovered-from-centered coefficients

	(Intercept)	year	I(year^2)
uncentered	577.2523	-15.8409	0.112301
recovered_from_center	577.2523	-15.8409	0.112301

```r
knitr::kable(t(c(cor_before = cor_raw, cor_after = cor_cen, mean_year = xbar)),
             digits = 6, caption = "Centering reduces correlation between x and x^2")
```

Table 14: Centering reduces correlation between x and x^2

cor_before	cor_after	mean_year
0.999759	0.014414	75.99496

6 6. Social-Media Contest and Post-Period Spending (part.csv)

```r
# Model 1
m6_1 <- lm(ly ~ lx + tx, data = d)
coeftest_m6_1 <- lmtest::coeftest(m6_1, vcov = sandwich::vcovHC(m6_1, type = "HC1"))
knitr::kable(broom::tidy(m6_1, conf.int = TRUE), digits = 4, caption = "Model 1: OLS")
```

Table 15: Model 1: OLS

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	-0.3170	0.0415	-7.6314	0	-0.3985	-0.2356
lx	0.8032	0.0120	66.6570	0	0.7796	0.8268
tx	0.2444	0.0284	8.5907	0	0.1886	0.3001

```r
knitr::kable(broom::tidy(coeftest_m6_1), digits = 4, caption = "Model 1: Robust (HC1) t-tests")
```

Table 16: Model 1: Robust (HC1) t-tests

term	estimate	std.error	statistic	p.value
(Intercept)	-0.3170	0.0346	-9.1629	0
lx	0.8032	0.0116	69.3199	0
tx	0.2444	0.0284	8.6013	0

```r
# Model 2
m6_2 <- lm(ly ~ lx + tx + lwc, data = d)
coeftest_m6_2 <- lmtest::coeftest(m6_2, vcov = sandwich::vcovHC(m6_2, type = "HC1"))
knitr::kable(broom::tidy(m6_2, conf.int = TRUE), digits = 4, caption = "Model 2: OLS")
```

Table 17: Model 2: OLS

term	estimate	std.error	statistic	p.value	conf.low	conf.high
(Intercept)	-0.3082	0.0417	-7.3984	0.0000	-0.3899	-0.2266
lx	0.8003	0.0121	66.1682	0.0000	0.7766	0.8240
tx	0.0504	0.0766	0.6580	0.5105	-0.0997	0.2005
lwc	0.0738	0.0271	2.7287	0.0064	0.0208	0.1269

```
knitr::kable(broom::tidy(coeftest_m6_2), digits = 4, caption = "Model 2: Robust t-tests")
```

Table 18: Model 2: Robust t-tests

term	estimate	std.error	statistic	p.value
(Intercept)	-0.3082	0.0347	-8.8771	0.0000
lx	0.8003	0.0116	68.7402	0.0000
tx	0.0504	0.0787	0.6402	0.5220
lwc	0.0738	0.0281	2.6247	0.0087

```
# 6(c)
tx_row <- broom::tidy(coeftest_m6_1) %>% dplyr::filter(term == "tx")
knitr::kable(tx_row, digits = 4, caption = "tx effect (Model 1, robust)")
```

Table 19: tx effect (Model 1, robust)

term	estimate	std.error	statistic	p.value
tx	0.2444	0.0284	8.6013	0

```
# 6(d)
b_tx <- coef(m6_1)["tx"]
times <- unname(exp(b_tx))
knitr::kable(t(data.frame(`multiplier on (y+1)` = times)), digits = 4)
```

multiplier.on..y.1.	1.2768

```
# (e)-(h)
cat("Adding log(wc+1) markedly attenuates the raw `tx` effect, consistent with omitted-variable
```

```
## Adding log(wc+1) markedly attenuates the raw `tx` effect, consistent with omitted-variable
```

7 Appendix: Session Info

```
sessionInfo()
```

```
## R version 4.5.0 (2025-04-11 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 11 x64 (build 26100)
##
## Matrix products: default
##   LAPACK version 3.12.1
##
## locale:
## [1] LC_COLLATE=Chinese (Simplified)_China.utf8
## [2] LC_CTYPE=Chinese (Simplified)_China.utf8
## [3] LC_MONETARY=Chinese (Simplified)_China.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=Chinese (Simplified)_China.utf8
##
## time zone: America/Chicago
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] ISLR2_1.3-2    knitr_1.50     lmtest_0.9-40  zoo_1.8-14     sandwich_3.1-1
## [6] broom_1.0.10   dplyr_1.1.4    ggplot2_4.0.0
##
## loaded via a namespace (and not attached):
##  [1] gtable_0.3.6       compiler_4.5.0     tidyselect_1.2.1   tidyr_1.3.1
##  [5] scales_1.4.0       yaml_2.3.10        fastmap_1.2.0      lattice_0.22-6
##  [9] R6_2.6.1           labeling_0.4.3     generics_0.1.4     backports_1.5.0
## [13] tibble_3.3.0       pillar_1.11.0      RColorBrewer_1.1-3 rlang_1.1.6
## [17] xfun_0.52          S7_0.2.0           cli_3.6.5          withr_3.0.2
## [21] magrittr_2.0.3     digest_0.6.37      grid_4.5.0         rstudioapi_0.17.1
## [25] lifecycle_1.0.4    vctrs_0.6.5        evaluate_1.0.5     glue_1.8.0
## [29] farver_2.1.2       rmarkdown_2.30     purrr_1.1.0        tools_4.5.0
## [33] pkgconfig_2.0.3    htmltools_0.5.8.1
```