A Fancy metaphor title: Explain what to explore

Novembver 2025

Socio-Economic Background

Williamsburg, Virginia, known for its historical significance and educational institutions, presents a paradox within its socio-economic landscape. Despite a reputation as a peaceful, tourist-driven city, Williamsburg struggles with a significant housing crisis that reflects broader socio-economic challenges. A low poverty rate is offset by a high housing burden, where a substantial portion of residents—especially those working in essential local industries—face steep costs to maintain stable housing. Tourism and education drive the city's economy, yet many workers cannot afford to live within city limits, as high property values and rents cater more to short-term tourist needs than to long-term residents (Trivette 2021).

Exploratory Data Analysis Executive Summary

- **Project Objective**:

- **Data Source**:

- **Data Reliability**:

- **THEME FOUND123**:

- **Limitations of the Analysis**:

Research Questions

-
-

Data Source and Assumptions

Data Cleaning Process

Notable Findings

Finding 1. Finding Theme

State the finding in a few lines (See Figure 1 and Table 1).

Finding 2. (See Figure 2 and Table 2).

Finding 3. (See Figure 3 and Table 3).

Limitations

Visualizations

- Working repo could be fount at:

  https://github.com/zzeng05/ZENG1-LIU2-727FINAL-scaVSeply.git

```r
# Fetch SCA Data
cs_url <- "https://data.sca.isr.umich.edu/data-archive/mine.php#"


#function to fetch any sca table from data site
get_sca_table <- function(table_num,

                             from_year = 2008,

                             to_year   = 2025,

                             freq      = "monthly") {


  body_list <- list(

    table = as.character(table_num),

    format = "html",

    from    = as.character(from_year),

    to      = as.character(to_year),

    freq   = freq

  )


  res <- POST(cs_url, body = body_list, encode = "form")

  stop_for_status(res)


  page <- read_html(res)
```

```
  page %>%

    html_node("div.output table") %>%

    html_table(fill = TRUE)

}
```

```
#fetch & clean table1: Consumer Sentiment Index

CS <- get_sca_table(1) %>%

  # same logic as your cs_tables[[2]] cleaning

  rename(

    month_chr = X1,

    year_chr  = X2,

    cs_chr    = X3

  ) %>%

  slice(-1) %>%  # drop the header row inside the table

  mutate(

    month = as.integer(month_chr),

    year  = as.integer(year_chr),

    cs    = as.numeric(cs_chr),

    date  = ymd(sprintf("%04d-%02d-01", year, month))

  ) %>%

  arrange(date) %>%

  select(date, cs, year, month)


head(CS)
```

```
# A tibble: 6 x 4

  date          cs  year month
```

```
    <date>      <dbl> <int> <int>

1 2008-01-01  78.4  2008     1

2 2008-02-01  70.8  2008     2

3 2008-03-01  69.5  2008     3

4 2008-04-01  62.6  2008     4

5 2008-05-01  59.8  2008     5

6 2008-06-01  56.4  2008     6
```

```
#fetch & clean table26: Expected change in business condition in a year

ECB <- get_sca_table(26) %>%

  { setNames(., as.character(unlist(.[1, ]))) } %>%

  slice(-1) %>%

  mutate(

    Month      = as.integer(Month),

    Year       = as.integer(Year),

    `Better`= as.numeric(`Better`),

    Same       = as.numeric(Same),

    `Worse` = as.numeric(`Worse`),

    `DK; NA`    = as.numeric(`DK; NA`),

    Relative   = as.numeric(Relative),

    date       = as.Date(sprintf("%04d-%02d-01", Year, Month))

  ) %>%

  relocate(date)


head(ECB)
```

```
# A tibble: 6 x 8
```

	date	Month	Year	Better	Same	Worse	`DK; NA`	Relative
	<date>	<int>	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	2008-01-01	1	2008	23	50	26	1	97
2	2008-02-01	2	2008	21	49	28	2	93
3	2008-03-01	3	2008	19	46	32	3	87
4	2008-04-01	4	2008	19	44	35	2	84
5	2008-05-01	5	2008	16	41	41	2	75
6	2008-06-01	6	2008	15	42	41	2	74

```r
#fetch & clean table30: Expected Change in Unemployment During the Next Year
UNEPLY1 <- get_sca_table(30) %>%
  { setNames(., as.character(unlist(.[1, ]))) } %>%
  slice(-1) %>%
  mutate(
    Month = as.integer(Month),
    Year  = as.integer(Year)
  ) %>%
  mutate(
    across(
      -c(Month, Year),
      ~ suppressWarnings(as.numeric(.))
    ),
    date = as.Date(sprintf("%04d-%02d-01", Year, Month))
  ) %>%
  relocate(date)


head(UNEPLY1)
```

```
# A tibble: 6 x 8
  date       Month  Year  Less  Same  More `DK; NA` Relative
  <date>     <int> <int> <dbl> <dbl> <dbl>    <dbl>    <dbl>
1 2008-01-01     1  2008     6    46    47        1       59
2 2008-02-01     2  2008     9    41    50        0       59
3 2008-03-01     3  2008     7    38    55        0       52
4 2008-04-01     4  2008     5    36    59        0       46
5 2008-05-01     5  2008     3    41    56        0       47
6 2008-06-01     6  2008     5    31    64        0       41
```

```
UNEPLY_supp <- UNEPLY1 %>%
  transmute(
    date,
    less_unemp  = Less,
    same_unemp  = Same,
    more_unemp  = More,
    dk_unemp    = `DK; NA`,
    rel_unemp   = Relative,
    net_unemp_expect = less_unemp - more_unemp
  )


head(UNEPLY_supp)
```

```
# A tibble: 6 x 7
  date       less_unemp same_unemp more_unemp dk_unemp rel_unemp
  <date>          <dbl>      <dbl>      <dbl>    <dbl>     <dbl>
1 2008-01-01          6         46         47        1        59
```

2 2008-02-01	9	41	50	0	59
3 2008-03-01	7	38	55	0	52
4 2008-04-01	5	36	59	0	46
5 2008-05-01	3	41	56	0	47
6 2008-06-01	5	31	64	0	41

```
# i 1 more variable: net_unemp_expect <dbl>
```
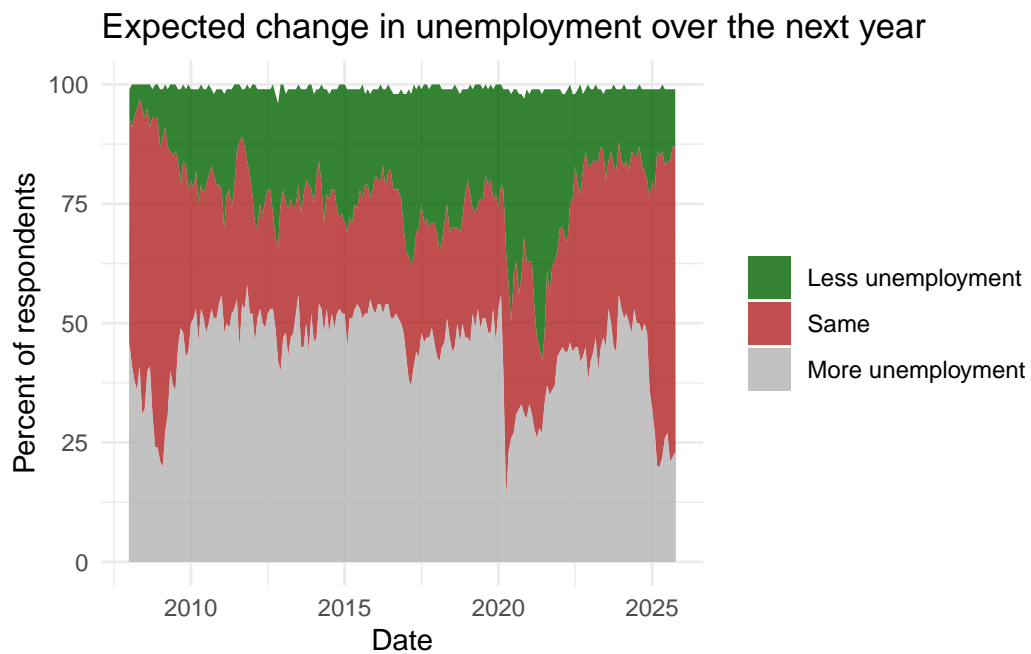
```r
UNEPLY_long <- UNEPLY_supp %>%
  select(date, less_unemp, same_unemp, more_unemp) %>%
  pivot_longer(
    cols = -date,
    names_to  = "expectation",
    values_to = "share"
  )


ggplot(UNEPLY_long, aes(x = date, y = share, fill = expectation)) +
  geom_area(alpha = 0.8) +
  scale_fill_manual(
    values = c(
      less_unemp = "darkgreen",
      same_unemp = "grey70",
      more_unemp = "firebrick"
    ),
    labels = c("Less unemployment", "Same", "More unemployment")
  ) +
  labs(
    title = "Expected change in unemployment over the next year",
```

```
    x       = "Date",

    y       = "Percent of respondents",

    fill  = NULL

  ) +

  theme_minimal()
```

Expected change in unemployment over the next year


```
# Fetch BLS Data


bls_key <- "554489bce3f14059aaa2dbb976d62372" #need to change back to path before submission


# BLS series IDs:

# - LNS14000000: Unemployment rate (CPS, seasonally adjusted)

# - CES0000000001: All employees, total nonfarm (CES, thousands, SA)

series_ids <- c("LNS14000000", "CES0000000001")
```

```r
# Request monthly data from 2008 to latest
res <- POST(
  url   = "https://api.bls.gov/publicAPI/v2/timeseries/data/",
  body  = list(
  seriesid        = series_ids,
  startyear       = "2008",
  endyear         = "2025",
  registrationkey = bls_key),
  encode = "json"
)


stop_for_status(res)


bls_list <- fromJSON(content(res, as = "text", encoding = "UTF-8"), simplifyDataFrame = TRUE)


# Transform BLS data to long format
series_tbl <- bls_list$Results$series


bls_long <- map_dfr(seq_len(nrow(series_tbl)), function(i) {
  series_id <- series_tbl$seriesID[i]
  dat       <- series_tbl$data[[i]]   # this is a data.frame for that series


  as_tibble(dat) %>%
    transmute(
      series_id = series_id,
      year      = as.integer(year),
```

```
      period,
      value    = as.numeric(value)
    ) %>%
    # keep monthly observations (M01-M12)
    filter(str_starts(period, "M")) %>%
    mutate(
      month = as.integer(str_remove(period, "M")),
      date  = ymd(sprintf("%04d-%02d-01", year, month))
    )
})


head(bls_long)
```

```
# A tibble: 6 x 6
  series_id    year period value month date
  <chr>       <int> <chr>  <dbl> <int> <date>
1 LNS14000000  2025 M09      4.4     9 2025-09-01
2 LNS14000000  2025 M08      4.3     8 2025-08-01
3 LNS14000000  2025 M07      4.2     7 2025-07-01
4 LNS14000000  2025 M06      4.1     6 2025-06-01
5 LNS14000000  2025 M05      4.2     5 2025-05-01
6 LNS14000000  2025 M04      4.2     4 2025-04-01
```

```
# CPS unemployment rate
UNEPLOY_R <- bls_long %>%
  filter(series_id == "LNS14000000") %>%
```

```
  arrange(date) %>%

  transmute(

    date,

    unrate = value

  )


# CES total nonfarm employment & monthly job change

Job_V <- bls_long %>%

  filter(series_id == "CES0000000001") %>%

  arrange(date) %>%

  transmute(

    date,

    nonfarm_emp = value,

    job_change  = nonfarm_emp - lag(nonfarm_emp)

  )


head(UNEPLOY_R)
```

```
# A tibble: 6 x 2

  date        unrate

  <date>       <dbl>

1 2008-01-01    5

2 2008-02-01    4.9

3 2008-03-01    5.1

4 2008-04-01    5

5 2008-05-01    5.4
```

```
6 2008-06-01    5.6
```

head(Job_V)

```
head(Job_V)
```
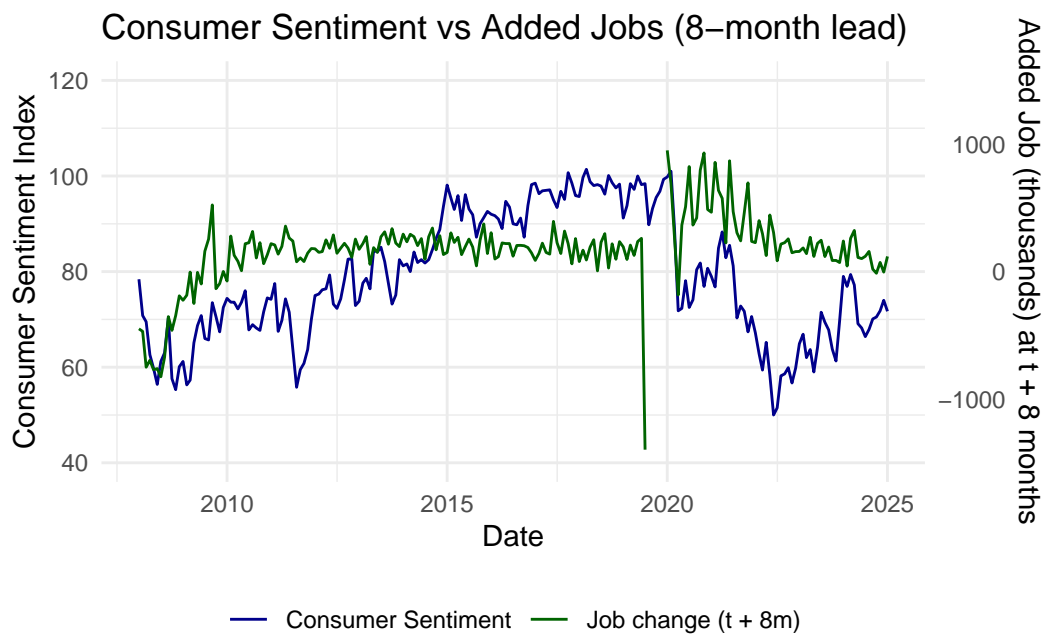
```
# A tibble: 6 x 3
  date       nonfarm_emp job_change
  <date>           <dbl>      <dbl>
1 2008-01-01      138391         NA
2 2008-02-01      138327        -64
3 2008-03-01      138257        -70
4 2008-04-01      138038       -219
5 2008-05-01      137851       -187
6 2008-06-01      137698       -153
```
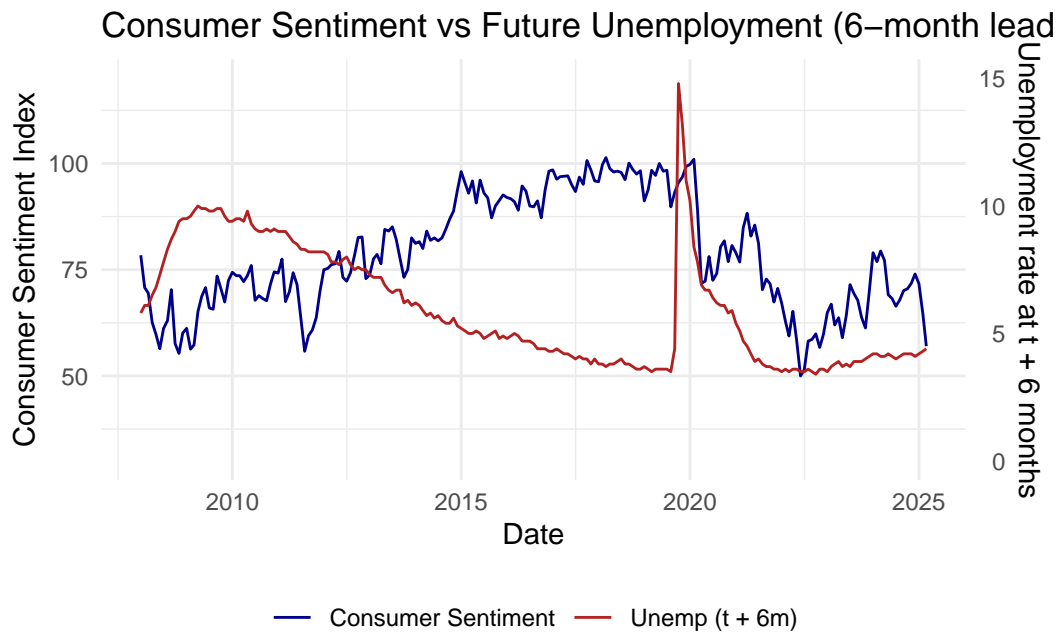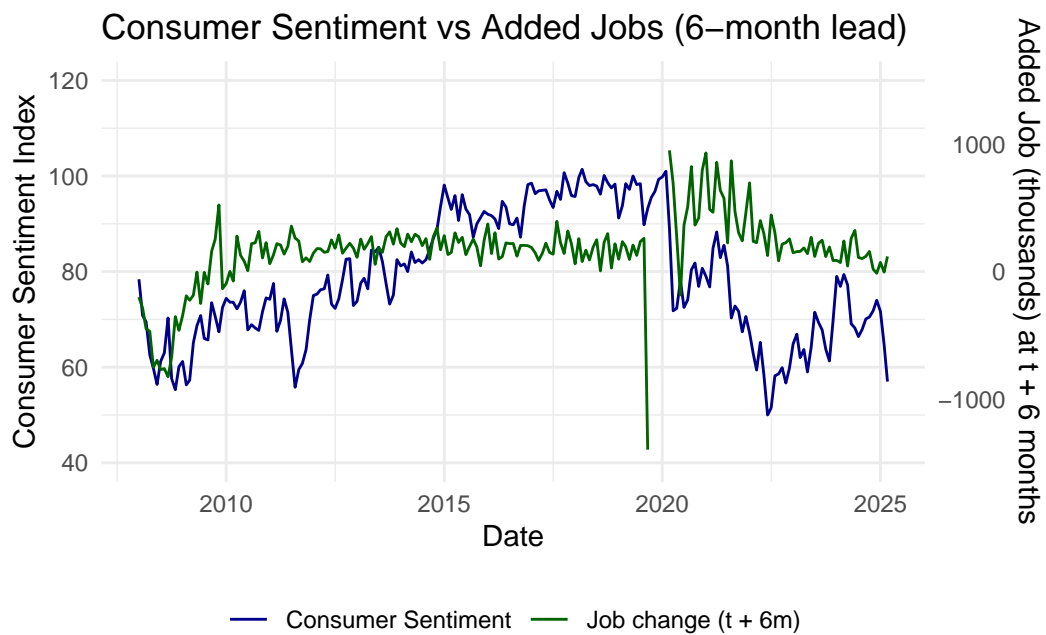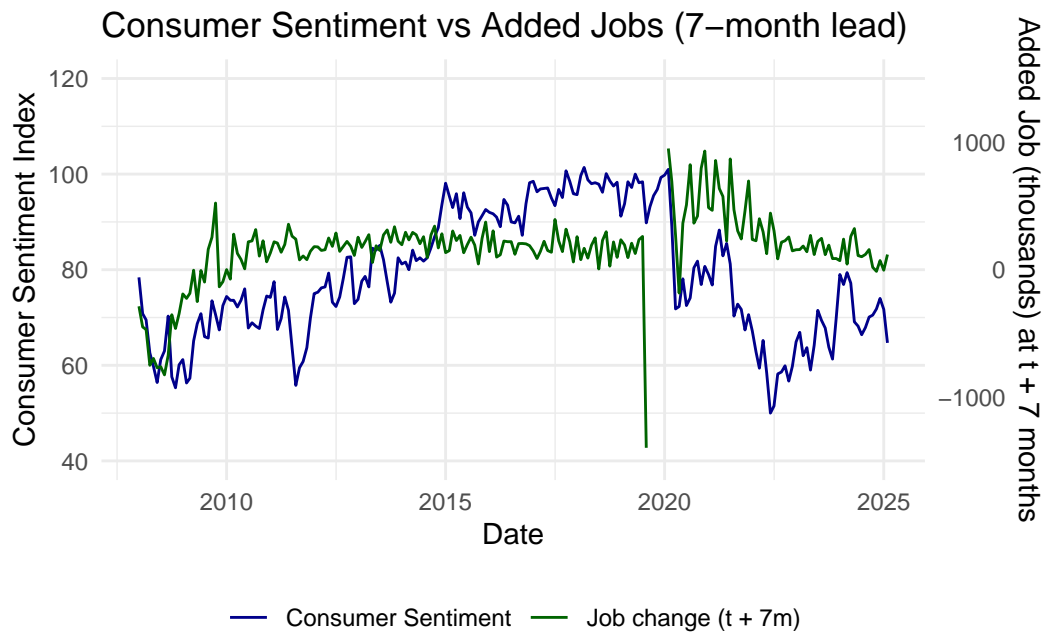
Figure 1. Lagged Time-Series of Consumer Sentiment and Unemployment Rate/Job Change

```
# A tibble: 6 x 5
  date          cs unrate nonfarm_emp job_change
  <date>     <dbl>  <dbl>       <dbl>      <dbl>
1 2008-01-01  78.4      5      138391         NA
2 2008-02-01  70.8    4.9      138327        -64
3 2008-03-01  69.5    5.1      138257        -70
4 2008-04-01  62.6      5      138038       -219
5 2008-05-01  59.8    5.4      137851       -187
6 2008-06-01  56.4    5.6      137698       -153
```
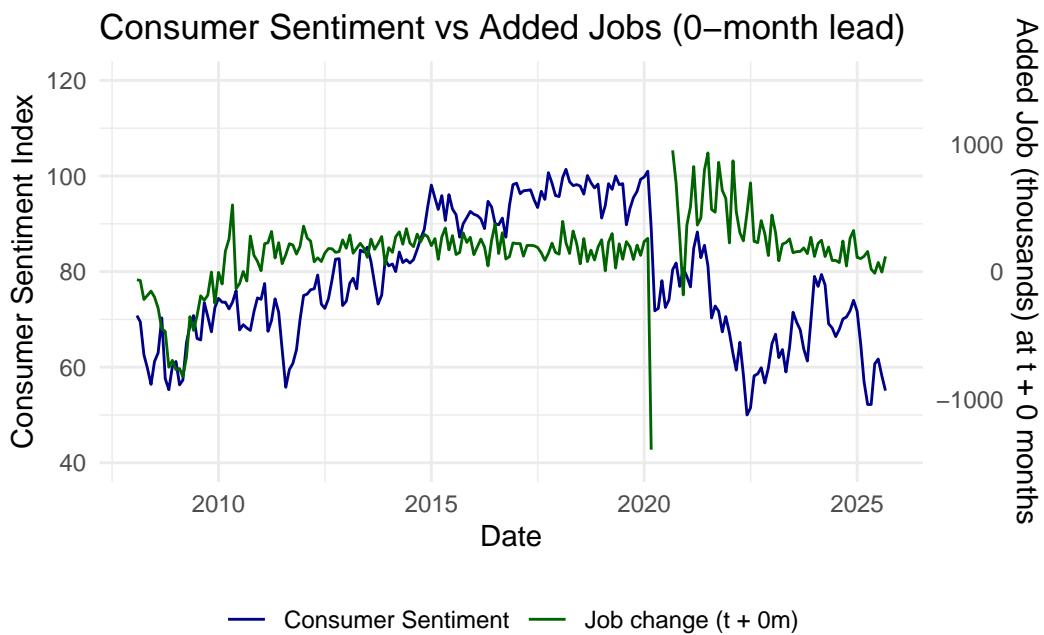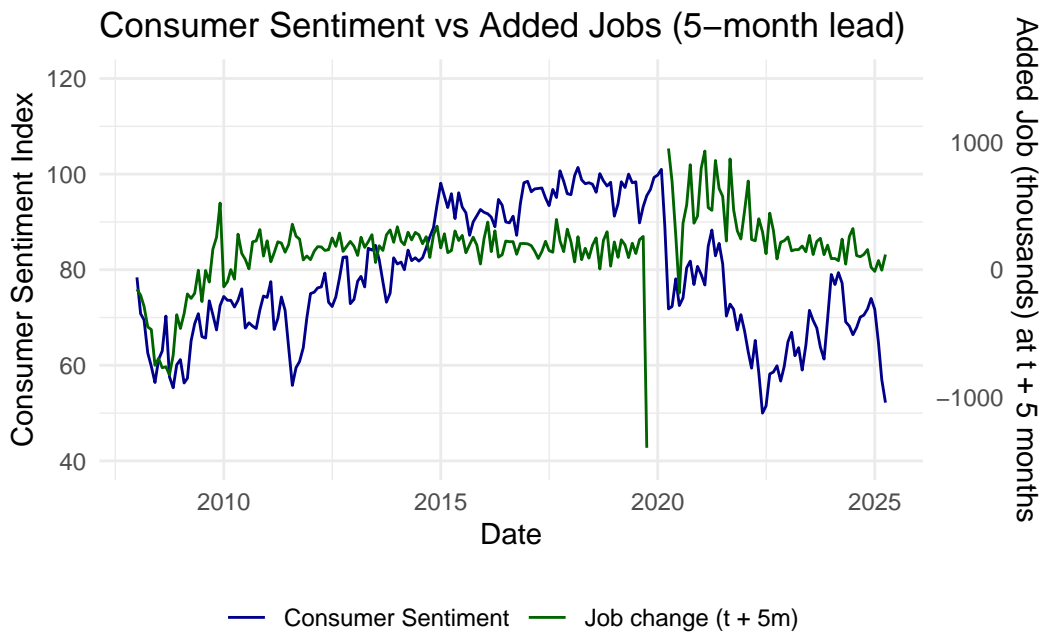
Consumer Sentiment vs Future Unemployment (6−month lead

Consumer Sentiment vs Added Jobs (8−month lead)

Consumer Sentiment vs Added Jobs (7−month lead)

Consumer Sentiment vs Added Jobs (6−month lead)

Consumer Sentiment vs Added Jobs (5–month lead)

— Consumer Sentiment — Job change (t + 5m)

Consumer Sentiment vs Added Jobs (0–month lead)

— Consumer Sentiment — Job change (t + 0m)

```
## Main dual-axis plot: CS vs future job creation

## with unemployment extremes overlaid (no separate y-axis)


lag_months  <- 3                    # choose 3, 6, or 12
```

```r
cs_limits   <- c(40, 120)        # left axis (CS)
jobs_limits <- c(-1000, 1000)    # right axis (job change, thousands)
unemp_limits <- c(2, 12)         # used only for scaling unemployment vertically


df <- macro %>%
  arrange(date) %>%
  mutate(
    jobchg_lead = lead(job_change, lag_months),
    unrate_lead = lead(unrate,     lag_months)
  ) %>%
  filter(!is.na(cs), !is.na(jobchg_lead), !is.na(unrate_lead))


# unpack limits
cs_min   <- cs_limits[1];  cs_max <- cs_limits[2]
jb_min   <- jobs_limits[1]; jb_max <- jobs_limits[2]
ur_min   <- unemp_limits[1]; ur_max <- unemp_limits[2]


cs_range <- cs_max - cs_min
jb_range <- jb_max - jb_min
ur_range <- ur_max - ur_min


# scale job change & unemployment into CS vertical space for plotting
df <- df %>%
  mutate(
    jobchg_scaled = cs_min + (jobchg_lead - jb_min) * (cs_range / jb_range),
    unrate_scaled = cs_min + (unrate_lead - ur_min) * (cs_range / ur_range)
```

```r
  )


# pick a few extreme unemployment values (e.g., 3 highest & 3 lowest)

top3 <- df %>% slice_max(order_by = unrate_lead, n = 3)

bottom3 <- df %>% slice_min(order_by = unrate_lead, n = 3)


unemp_extremes <- bind_rows(top3, bottom3) %>%

  arrange(date) %>%

  distinct(date, .keep_all = TRUE)


# legend labels

cs_label    <- "Consumer Sentiment"

job_label   <- paste0("Job change (t + ", lag_months, "m)")

unemp_label <- "Unemployment (no axis)"


ggplot(df, aes(x = date)) +

  # main series: Consumer Sentiment (left axis)

  geom_line(aes(y = cs, color = cs_label), linewidth = 0.7) +


  # main series: job creation (right axis, scaled onto CS scale)

  geom_line(aes(y = jobchg_scaled, color = job_label),

            linewidth = 0.7) +


  # overlay unemployment (same lag), no axis - dashed grey line

  geom_line(aes(y = unrate_scaled, color = unemp_label),

            linewidth = 0.4, alpha = 0.5, linetype = "dashed") +
```

```r
# label only extreme high/low unemployment months
geom_point(
  data = unemp_extremes,
  aes(y = unrate_scaled),
  color = "black", size = 1.8
) +
geom_text(
  data = unemp_extremes,
  aes(y = unrate_scaled,
      label = sprintf("%.1f%%", unrate_lead)),
  vjust = -0.4,
  size  = 3,
  color = "black"
) +


# dual y-axes: left = CS, right = job change
scale_y_continuous(
  name   = "Consumer Sentiment Index",
  limits = cs_limits,
  sec.axis = sec_axis(
    trans = ~ (.- cs_min) * (jb_range / cs_range) + jb_min,
    name  = paste0("Monthly change in nonfarm employment (thousands), t + ",
                   lag_months, " months")
  )
) +
```

```r
  scale_color_manual(
    values = setNames(
      c("steelblue", "darkgreen", "grey50"),
      c(cs_label, job_label, unemp_label)
    )
  ) +
  labs(
    title = paste0("Consumer Sentiment vs Future Job Creation (", lag_months, "-month lead)"),
    subtitle = "Unemployment rate at the same horizon is shown as a dashed line; extremes are
    x = "Date",
    color = NULL
  ) +
  theme_minimal() +
  theme(
    legend.position = "bottom"
  )
```

Warning: Removed 2 rows containing missing values or values outside the scale range (`geom_point()`).

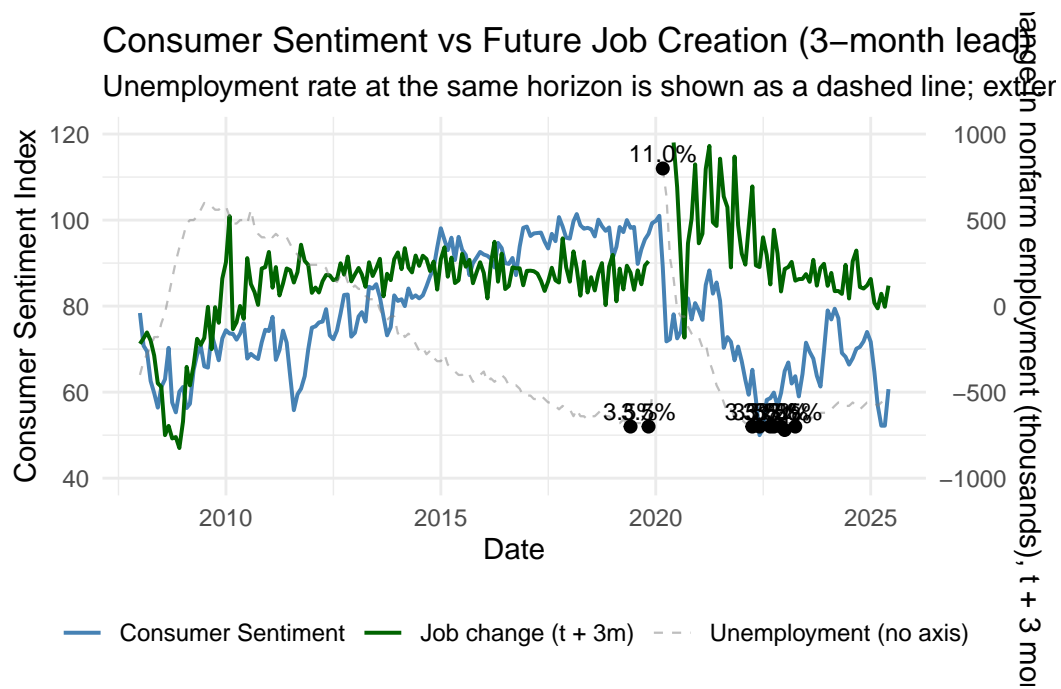Warning: Removed 2 rows containing missing values or values outside the scale range (`geom_text()`).

Consumer Sentiment vs Future Job Creation (3–month lead)

Unemployment rate at the same horizon is shown as a dashed line; exter

Consumer Sentiment Index

larger nonfarm employment (thousands), t + 3 mo

11.0%

3.5% 3.5% 3.5% 3.5% 3.5% 3.6%

Date

— Consumer Sentiment   — Job change (t + 3m)   – – Unemployment (no axis)

Figure 2. Scatter: net expectations vs subsequent unemployment change

Net unemployment expectations vs. 3–month change in actual

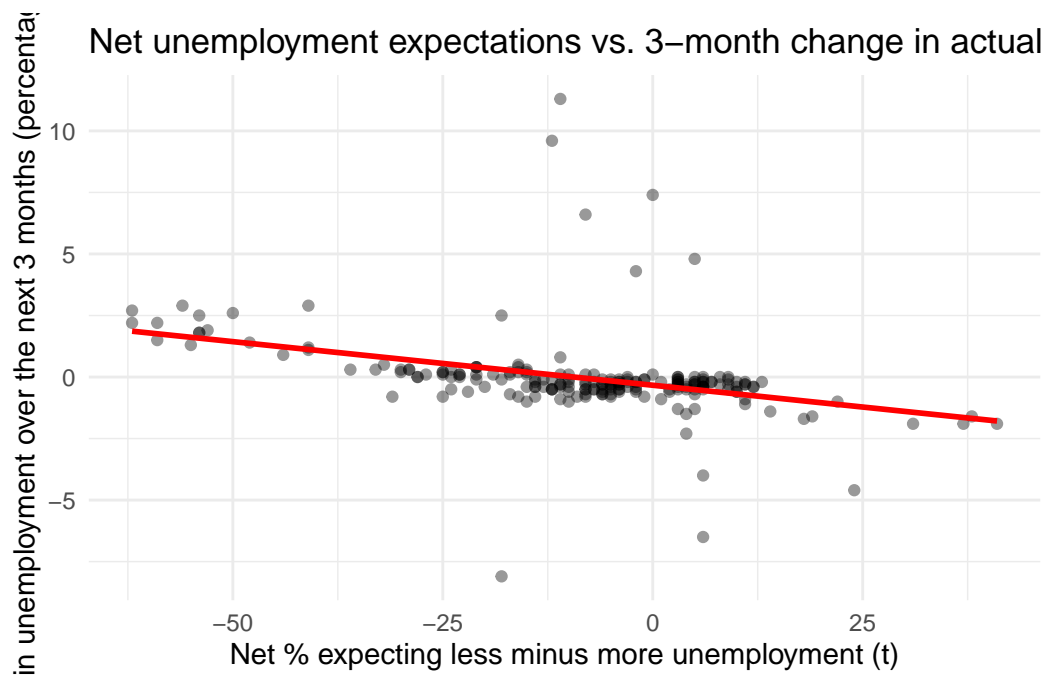**Figure 3. Time series: net expectations vs actual unemployment (dual axis)**

```r
plot_expect_vs_unemp_dual <- function(data_macro2,

                                      cs_limits   = c(-60, 60),   # for net expectations

                                      unemp_limits= c(2, 12)) {


  df <- data_macro2 %>%

    filter(!is.na(net_unemp_expect), !is.na(unrate)) %>%

    arrange(date)



  ex_min <- cs_limits[1]
```

```r
ex_max <- cs_limits[2]

ur_min <- unemp_limits[1]

ur_max <- unemp_limits[2]


ex_rng <- ex_max - ex_min

ur_rng <- ur_max - ur_min


df <- df %>%
  mutate(
    unrate_scaled = ex_min + (unrate - ur_min) * (ex_rng / ur_rng)
  )


ggplot(df, aes(x = date)) +
  geom_line(aes(y = net_unemp_expect, color = "Net expectations (Less-More)")) +
  geom_line(aes(y = unrate_scaled,    color = "Unemployment")) +
  scale_y_continuous(
    name   = "Net (Less-More) unemployment expectations",
    limits = cs_limits,
    sec.axis = sec_axis(
      trans = ~ (.- ex_min) * (ur_rng / ex_rng) + ur_min,
      name  = "Unemployment"
    )
  ) +
  scale_color_manual(
    values = setNames(
      c("darkgreen", "firebrick"),
```

```
        c("Net expectations (Less-More)", "Unemployment")
    )
  ) +

  labs(

    title = "Net Unemployment Expectations vs Actual Unemployment",

    x     = "Date",

    color = NULL

  ) +

  theme_minimal() +

  theme(legend.position = "bottom")
}


plot_expect_vs_unemp_dual(macro2)
```


Net Unemployment Expectations vs Actual Unemployment

Indications

Strengthening Tenant Protections.

Conclusion & Outlook

References

https://data.sca.isr.umich.edu/data-archive/mine.php https://www.bls.gov/cps

https://www.bls.gov/ces