GLM

Mingli Xu

2025-04-29

```
library(readr)
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
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

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

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

```
library(tidyr)

# Load & prep data
proj <- read_csv("project_data(2).csv") %>%
  mutate(observation_date = as.Date(observation_date, "%Y/%m/%d"))
```

```
## New names:
## * `` -> `...13`
## Rows: 672 Columns: 15
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): observation_date
## dbl (13): UMCSENT_interp, hourly_earning, BBKMGDP, CPI, CPILFESL, discourage...
## lgl  (1): ...13
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```r
nasdaq <- read_csv("nasdaqmonth.csv") %>%
  mutate(month = as.Date(month))
```

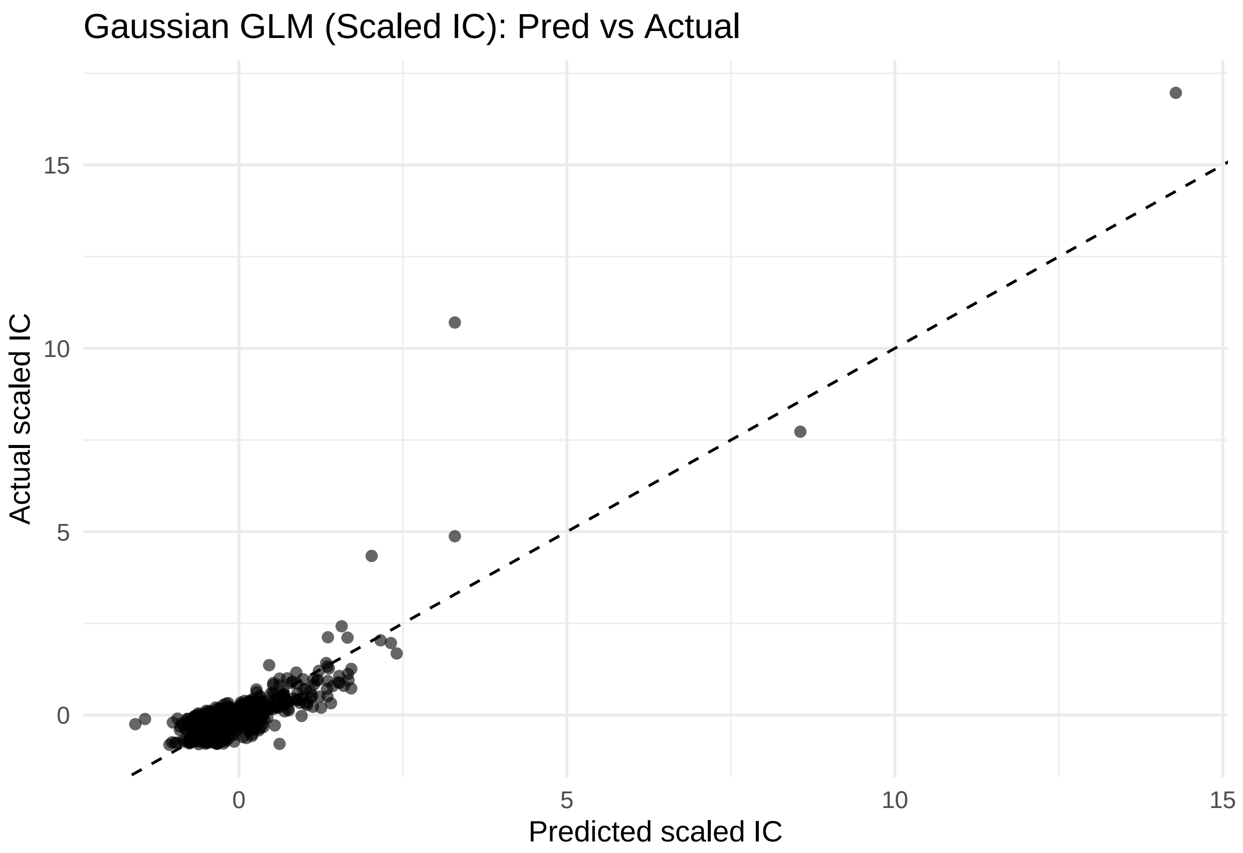
```
## Rows: 651 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (1): monthly_average
## date (1): month
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```r
df <- left_join(proj, nasdaq, by = c("observation_date"="month")) %>%
  filter(!is.na(monthly_average))

# Drop rows with any missing modeling vars
df_sub <- df %>%
  drop_na(IC, CC,
          UMCSENT_interp, hourly_earning, BBKMGDP, CPI,
          FEDFUNDS, num_losers, unemployment_level, monthly_average)

# Scale IC & CC
df_sub <- df_sub %>%
  mutate(
    scaled_IC = scale(IC)[,1],
    scaled_CC = scale(CC)[,1]
  )

# Shift for Gamma (must be > 0)
min_IC <- min(df_sub$scaled_IC)
min_CC <- min(df_sub$scaled_CC)
df_sub <- df_sub %>%
  mutate(
    shifted_IC = scaled_IC - min_IC + 0.01,
    shifted_CC = scaled_CC - min_CC + 0.01
  )

# Fit Gaussian GLMs
glm_IC_g <- glm(scaled_IC ~ UMCSENT_interp + hourly_earning +
                  BBKMGDP + CPI + FEDFUNDS + num_losers +
                  unemployment_level + monthly_average,
                data = df_sub, family = gaussian())
glm_CC_g <- glm(scaled_CC ~ UMCSENT_interp + hourly_earning +
                  BBKMGDP + CPI + FEDFUNDS + num_losers +
                  unemployment_level + monthly_average,
                data = df_sub, family = gaussian())

# Fit Gamma-log GLMs on shifted responses
glm_IC_gl <- glm(shifted_IC ~ UMCSENT_interp + hourly_earning +
                   BBKMGDP + CPI + FEDFUNDS + num_losers +
                   unemployment_level + monthly_average,
                 data = df_sub, family = Gamma(link="log"))
```

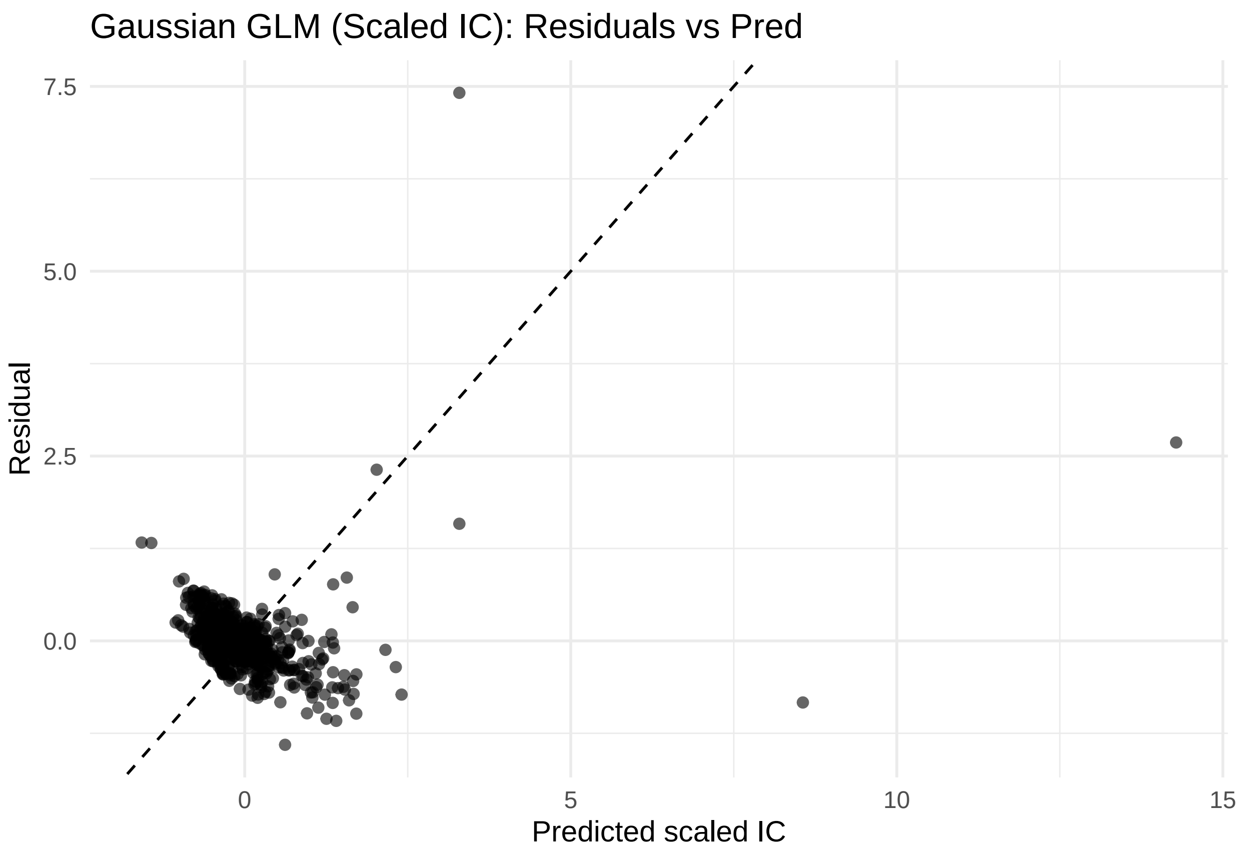
```
## Warning: glm.fit: algorithm did not converge
```

```
glm_CC_gl <- glm(shifted_CC ~ UMCSENT_interp + hourly_earning +
                   BBKMGDP + CPI + FEDFUNDS + num_losers +
                   unemployment_level + monthly_average,
                 data = df_sub, family = Gamma(link="log"))

# Add preds & residuals
df_sub <- df_sub %>%
  mutate(
    pred_IC_g    = predict(glm_IC_g),
    res_IC_g     = scaled_IC - pred_IC_g,
    pred_CC_g    = predict(glm_CC_g),
    res_CC_g     = scaled_CC - pred_CC_g,
    pred_IC_gl   = predict(glm_IC_gl, type="response"),
    res_IC_gl    = shifted_IC - pred_IC_gl,
    pred_CC_gl   = predict(glm_CC_gl, type="response"),
    res_CC_gl    = shifted_CC - pred_CC_gl
  )

# Plotting function
plot_pair <- function(data, x, y, title, xlab, ylab){
  ggplot(data, aes_string(x=x, y=y)) +
    geom_point(alpha=0.6) +
    geom_abline(slope=1, intercept=0, linetype="dashed") +
    labs(title=title, x=xlab, y=ylab) +
    theme_minimal()
}

# Gaussian: IC
print(plot_pair(df_sub, "pred_IC_g", "scaled_IC",
                "Gaussian GLM (Scaled IC): Pred vs Actual",
                "Predicted scaled IC", "Actual scaled IC"))
```

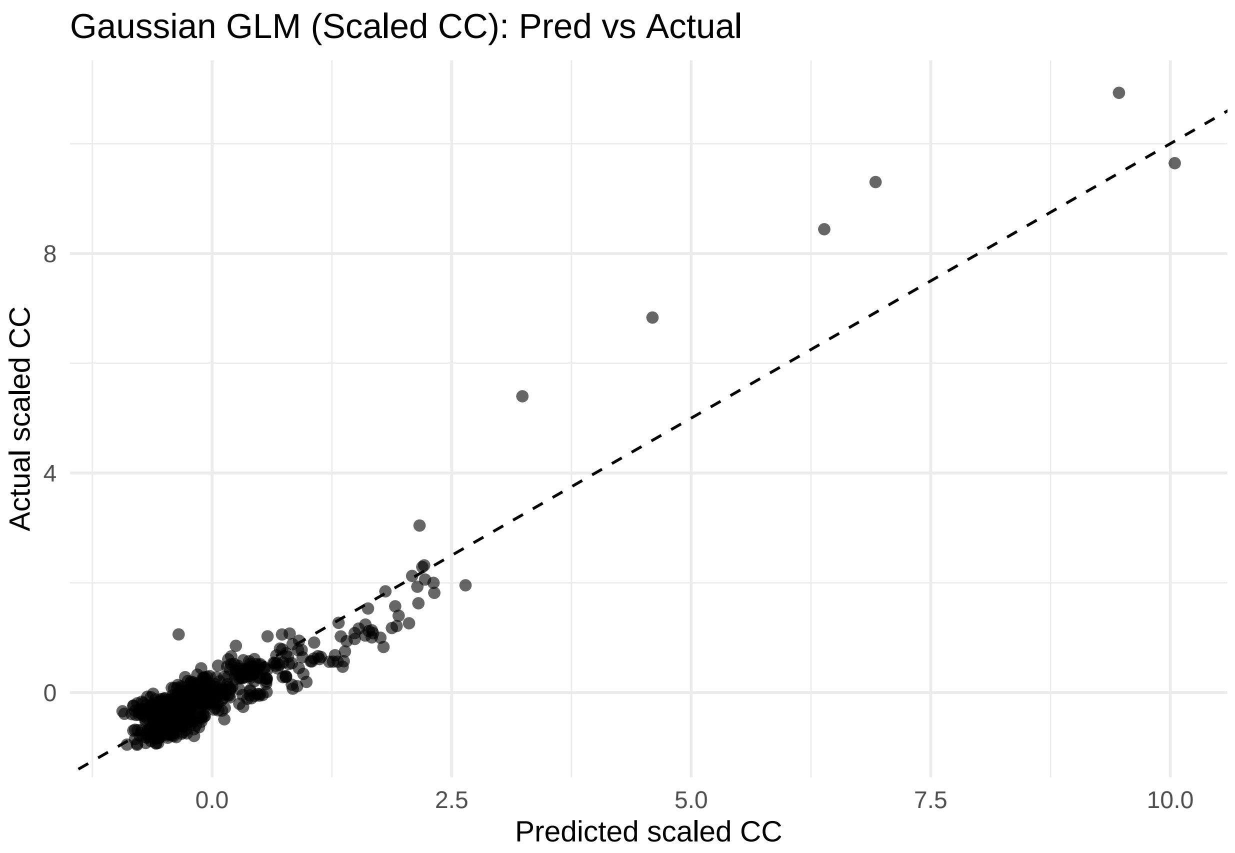
```
## Warning: ‘aes_string()‘ was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with ‘aes()‘.
## i See also ‘vignette("ggplot2-in-packages")‘ for more information.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



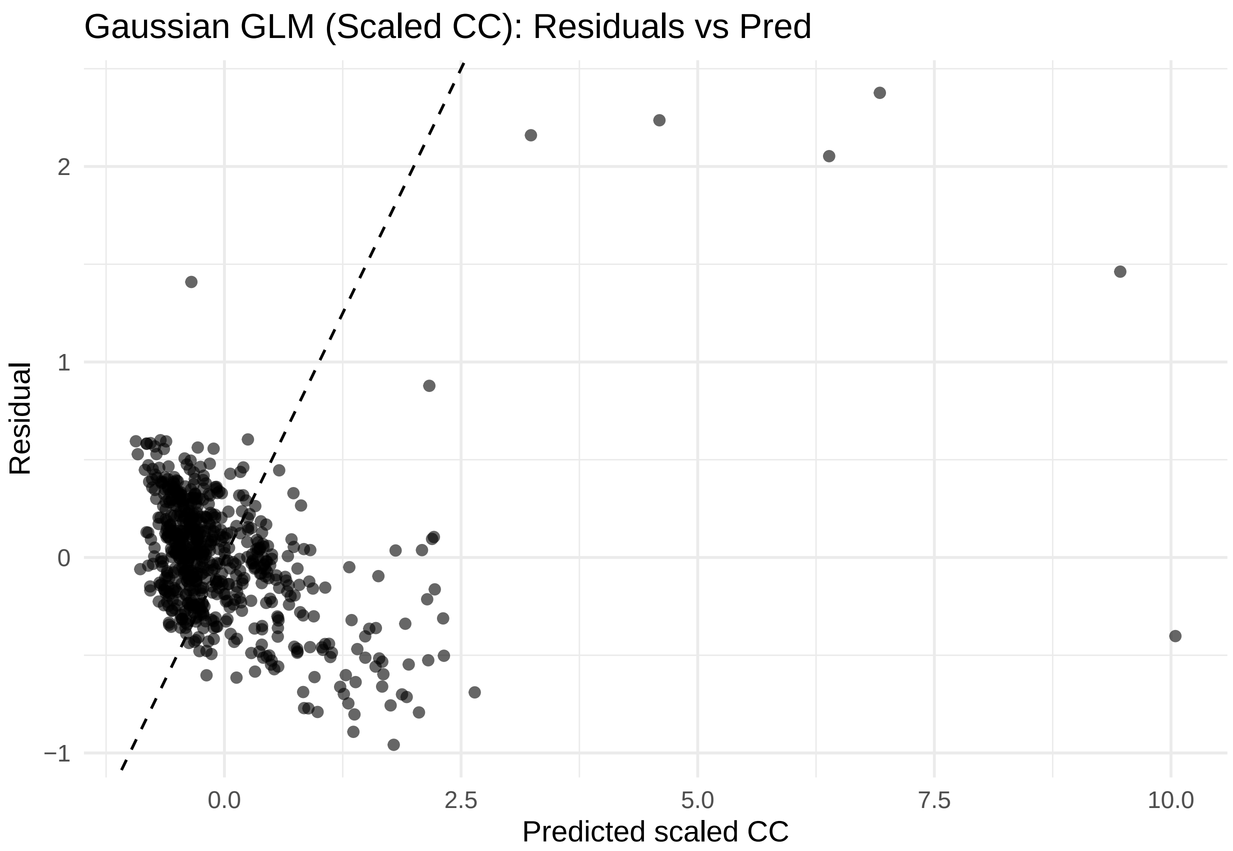
```r
print(plot_pair(df_sub, "pred_IC_g", "res_IC_g",
                "Gaussian GLM (Scaled IC): Residuals vs Pred",
                "Predicted scaled IC", "Residual"))
```



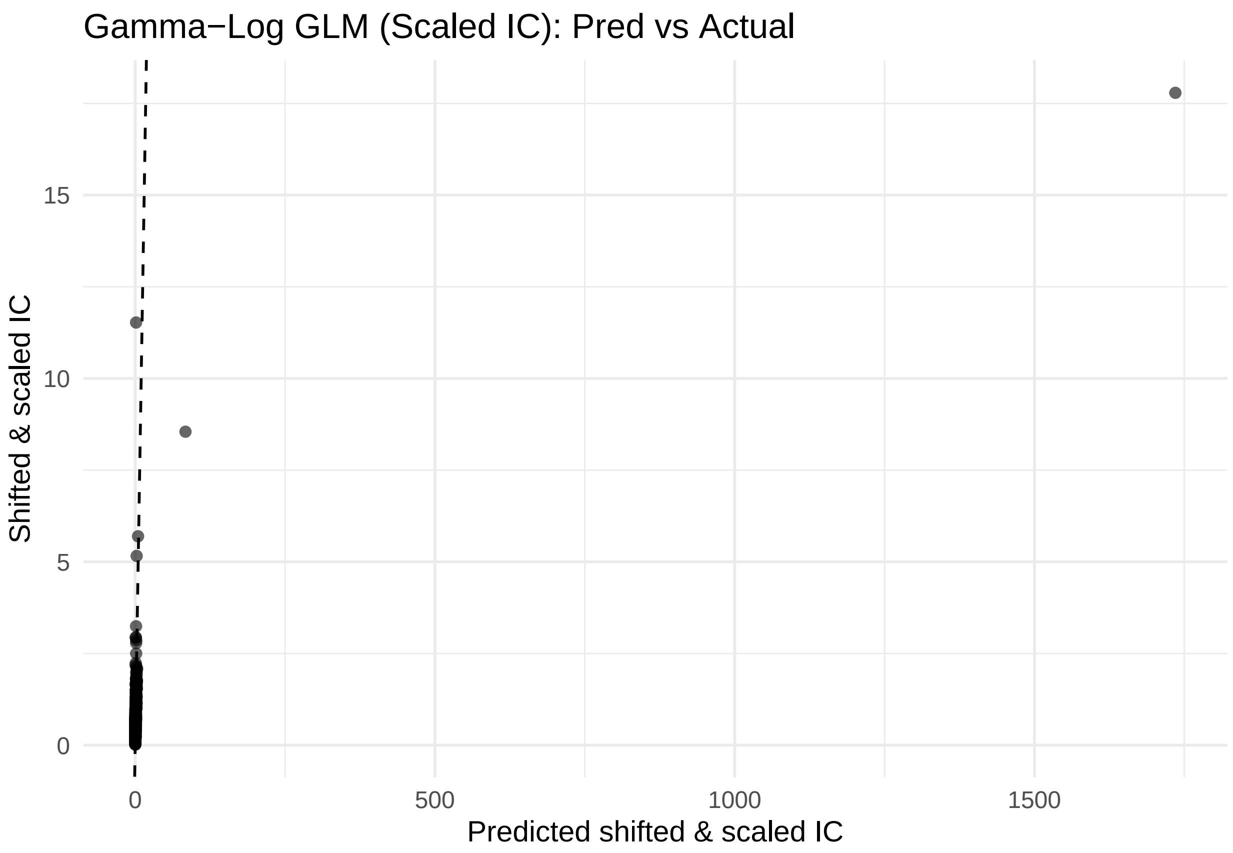
```r
# Gaussian: CC
print(plot_pair(df_sub, "pred_CC_g", "scaled_CC",
                "Gaussian GLM (Scaled CC): Pred vs Actual",
                "Predicted scaled CC", "Actual scaled CC"))
```


Gaussian GLM (Scaled CC): Pred vs Actual

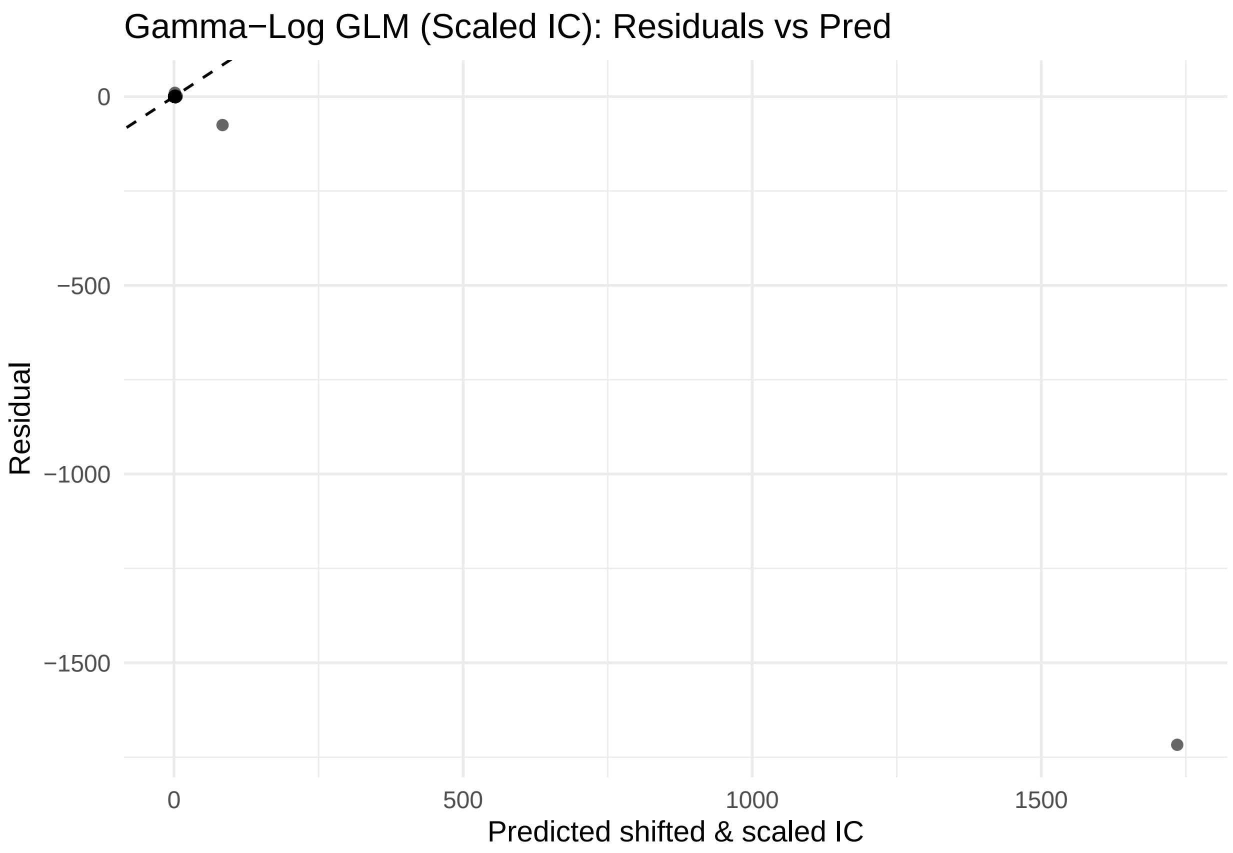
```
print(plot_pair(df_sub, "pred_CC_g", "res_CC_g",
                "Gaussian GLM (Scaled CC): Residuals vs Pred",
                "Predicted scaled CC", "Residual"))
```



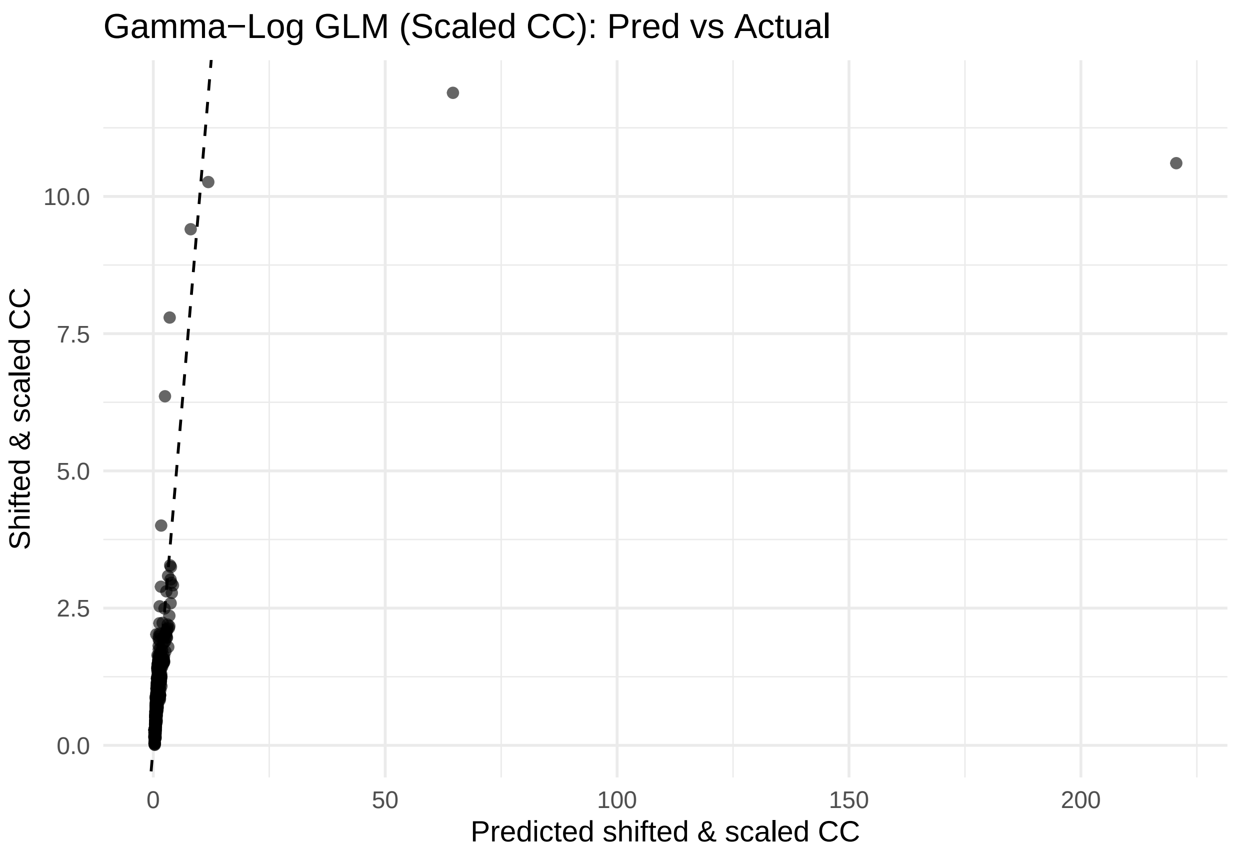
```
# Gamma-Log: IC
print(plot_pair(df_sub, "pred_IC_gl", "shifted_IC",
                "Gamma-Log GLM (Scaled IC): Pred vs Actual",
                "Predicted shifted & scaled IC", "Shifted & scaled IC"))
```



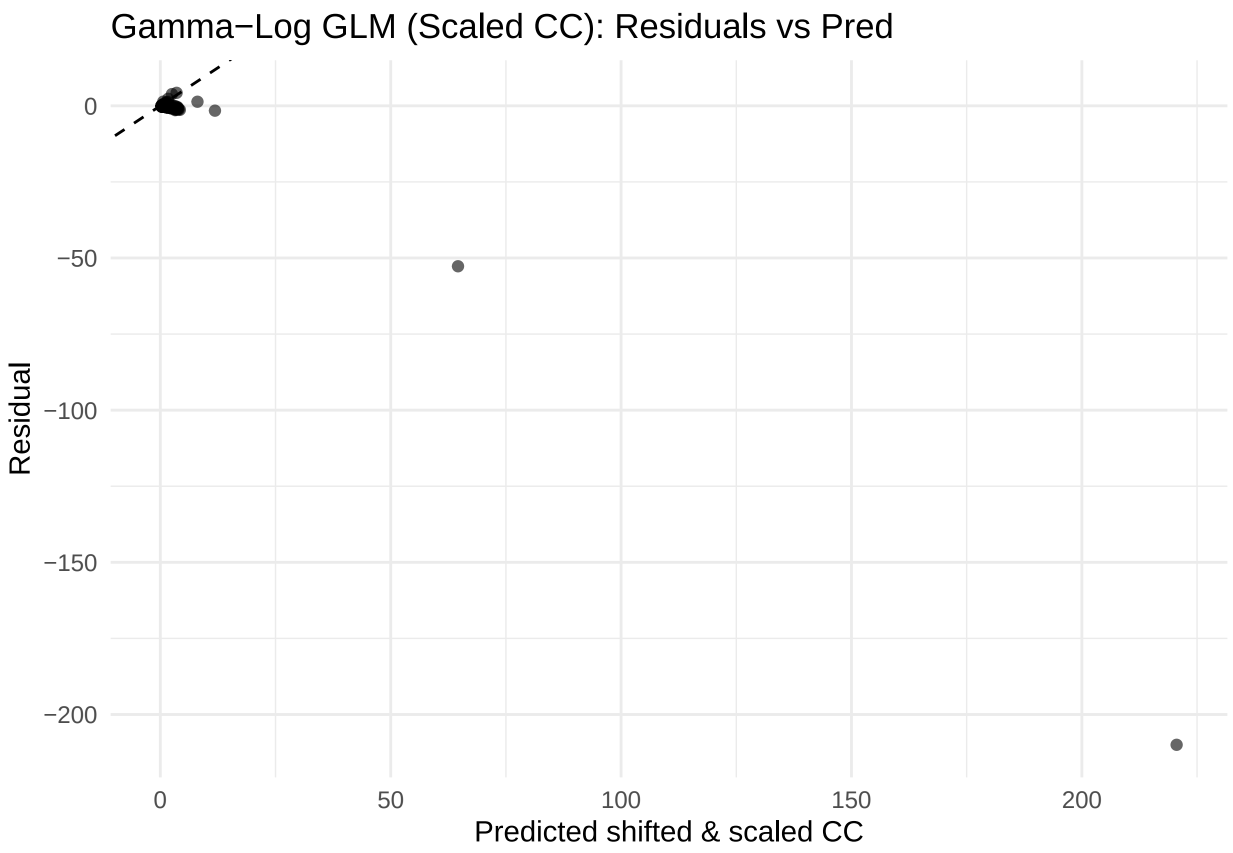
```r
print(plot_pair(df_sub, "pred_IC_gl", "res_IC_gl",
                "Gamma-Log GLM (Scaled IC): Residuals vs Pred",
                "Predicted shifted & scaled IC", "Residual"))
```



```r
# Gamma-Log: CC
print(plot_pair(df_sub, "pred_CC_gl", "shifted_CC",
                "Gamma-Log GLM (Scaled CC): Pred vs Actual",
                "Predicted shifted & scaled CC", "Shifted & scaled CC"))
```



```
print(plot_pair(df_sub, "pred_CC_gl", "res_CC_gl",
                "Gamma-Log GLM (Scaled CC): Residuals vs Pred",
                "Predicted shifted & scaled CC", "Residual"))
```



```r
library(dplyr)
library(ggplot2)

# Un-shift the Gamma-log predictions back to scaled_IC / scaled_CC scale
df_sub <- df_sub %>%
  mutate(
    pred_IC_gl_scaled = pred_IC_gl + (min_IC - 0.01),
    pred_CC_gl_scaled = pred_CC_gl + (min_CC - 0.01)
  )

# Define MSE function
mse <- function(actual, predicted) mean((actual - predicted)^2)

# Compute MSE for each model
mse_IC_gauss <- mse(df_sub$scaled_IC, df_sub$pred_IC_g)
mse_IC_gamma <- mse(df_sub$scaled_IC, df_sub$pred_IC_gl_scaled)
mse_CC_gauss <- mse(df_sub$scaled_CC, df_sub$pred_CC_g)
mse_CC_gamma <- mse(df_sub$scaled_CC, df_sub$pred_CC_gl_scaled)

# Assemble into a table
library(tibble)
mse_scaled_df <- tibble(
  Model          = c("IC (Gaussian)", "IC (Gamma-log)",
                     "CC (Gaussian)", "CC (Gamma-log)"),
  MSE            = c(mse_IC_gauss, mse_IC_gamma,
                     mse_CC_gauss, mse_CC_gamma)
)
print(mse_scaled_df)
```

```
## # A tibble: 4 x 2
##   Model               MSE
##   <chr>             <dbl>
## 1 IC (Gaussian)     0.225
## 2 IC (Gamma-log) 4639.
## 3 CC (Gaussian)     0.120
## 4 CC (Gamma-log)   73.7
```