Manuscript results: models comparasion using the FluSight baseline

Victor Felix

January 14, 2026

This document has the results that we present in our manuscript draft. We replace the AUTO_AR model for the FluSight baseline.

Loading libraries.

```r
library("tidyr")
library("MMWRweek")
library("data.table")
library("caret")
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```r
library("purrr")
```

```
##
## Attaching package: 'purrr'

## The following object is masked from 'package:caret':
##
##     lift

## The following object is masked from 'package:data.table':
##
##     transpose
```

```r
library("dplyr")
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:data.table':
##
##     between, first, last

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

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

```r
library("tseries")
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```r
library("gtools")
library("forecast")
library("scoringutils")
```

```
## Note: scoringutils is currently undergoing major development changes (with an update planned for the
```

```r
library("covidHubUtils")
library("parallel")
library("future")#https://cran.r-project.org/web/packages/future/vignettes/future-4-issues.html
```

```
##
## Attaching package: 'future'

## The following object is masked from 'package:tseries':
##
##     value

## The following object is masked from 'package:caret':
##
##     cluster
```

```r
library("listenv")
```

```
##
## Attaching package: 'listenv'

## The following object is masked from 'package:purrr':
##
##     map
```

```r
library("epitools")
library("ggplot2")
library("sf")
```

```
## Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf_use_s2() is TRUE
```

```r
library("forcats")
library("ggplot2")
library("sf")
library("scales")
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard
```

```
library("ggplot2")
library("broom")
library("fields")
```

```
## Loading required package: spam

## Spam version 2.10-0 (2023-10-23) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve

## Loading required package: viridisLite

##
## Try help(fields) to get started.
```

```
library("ggpubr")
```

```
##
## Attaching package: 'ggpubr'

## The following object is masked from 'package:forecast':
##
##     gghistogram
```

```
library("patchwork")
library("ggpattern")
library("cowplot")
```

```
##
## Attaching package: 'cowplot'

## The following object is masked from 'package:patchwork':
##
##     align_plots
```

```
## The following object is masked from 'package:ggpubr':
##
##     get_legend
```

```r
library("lme4")
```

```
## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following object is masked from 'package:spam':
##
##     det

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```r
library("ez")
library("ggpubr")
```

Loading the results of each model and the shapefiles of the maps.

```r
load("models_without_logback/ES_ARIMA/ARIMA_MODELS_influenza_hospitalization_nolog.Rdata")
load("models_without_logback/ES_ADJACENT/ADJACENT_MODELS_influenza_hospitalization_nolog.Rdata")
load("models_without_logback/ES_EPIWEEK/EPIWEEK_MODELS_influenza_hospitalization_nolog.Rdata")
load("models_without_logback/ES_TEMPERATURE/TEMPERATURE_MODELS_influenza_hospitalization_nolog.Rdata")
load("models_without_logback/ES_AVERAGE/AVERAGE_MODELS_influenza_hospitalization_nolog.Rdata")

states <- read_sf("models_without_logback/shapefiles/cb_2018_us_state_500k.shp")

states <- states %>%
  rename(STATE = NAME)
```

1 Week ahead

```r
# Creating new columns with Epidemiological weeks based on target_end_week
AUTO_ARIMA_WEEK1$epiweek <- MMWRweek(AUTO_ARIMA_WEEK1$target_end_date)$MMWRweek
AUTO_ADJACENT_WEEK1$epiweek <- MMWRweek(AUTO_ADJACENT_WEEK1$target_end_date)$MMWRweek
AUTO_EPIWEEK_WEEK1$epiweek <- MMWRweek(AUTO_EPIWEEK_WEEK1$target_end_date)$MMWRweek
AUTO_TEMPERATURE_WEEK1$epiweek <- MMWRweek(AUTO_TEMPERATURE_WEEK1$target_end_date)$MMWRweek
AUTO_AVERAGE_WEEK1$epiweek <- MMWRweek(AUTO_AVERAGE_WEEK1$target_end_date)$MMWRweek

ES27_ARIMA_WEEK1$epiweek <- MMWRweek(ES27_ARIMA_WEEK1$target_end_date)$MMWRweek
ES27_ADJACENT_WEEK1$epiweek <- MMWRweek(ES27_ADJACENT_WEEK1$target_end_date)$MMWRweek
ES27_EPIWEEK_WEEK1$epiweek <- MMWRweek(ES27_EPIWEEK_WEEK1$target_end_date)$MMWRweek
ES27_TEMPERATURE_WEEK1$epiweek <- MMWRweek(ES27_TEMPERATURE_WEEK1$target_end_date)$MMWRweek
ES27_AVERAGE_WEEK1$epiweek <- MMWRweek(ES27_AVERAGE_WEEK1$target_end_date)$MMWRweek
```

```r
ES64_ARIMA_WEEK1$epiweek <- MMWRweek(ES64_ARIMA_WEEK1$target_end_date)$MMWRweek
ES64_ADJACENT_WEEK1$epiweek <- MMWRweek(ES64_ADJACENT_WEEK1$target_end_date)$MMWRweek
ES64_EPIWEEK_WEEK1$epiweek <- MMWRweek(ES64_EPIWEEK_WEEK1$target_end_date)$MMWRweek
ES64_TEMPERATURE_WEEK1$epiweek <- MMWRweek(ES64_TEMPERATURE_WEEK1$target_end_date)$MMWRweek
ES64_AVERAGE_WEEK1$epiweek <- MMWRweek(ES64_AVERAGE_WEEK1$target_end_date)$MMWRweek

# Dataframe that will be analysed for 1 Week Ahead
df_W1_NoLg <- data.frame(
  STATE = AUTO_ARIMA_WEEK1$State,
  Julian_date = AUTO_ARIMA_WEEK1$target_end_date,
  epiweek = AUTO_ARIMA_WEEK1$epiweek,
  AUTO_AR=AUTO_ARIMA_WEEK1$WIS,
  AUTO_ADJ=AUTO_ADJACENT_WEEK1$WIS,
  AUTO_EPI=AUTO_EPIWEEK_WEEK1$WIS,
  AUTO_TEMP=AUTO_TEMPERATURE_WEEK1$WIS,
  AUTO_AVG=AUTO_AVERAGE_WEEK1$WIS,

  ES27_AR=ES27_ARIMA_WEEK1$WIS,
  ES27_ADJ=ES27_ADJACENT_WEEK1$WIS,
  ES27_EPI=ES27_EPIWEEK_WEEK1$WIS,
  ES27_TEMP=ES27_TEMPERATURE_WEEK1$WIS,
  ES27_AVG=ES27_AVERAGE_WEEK1$WIS,

  ES64_AR=ES64_ARIMA_WEEK1$WIS,
  ES64_ADJ=ES64_ADJACENT_WEEK1$WIS,
  ES64_EPI=ES64_EPIWEEK_WEEK1$WIS,
  ES64_TEMP=ES64_TEMPERATURE_WEEK1$WIS,
  ES64_AVG=ES64_AVERAGE_WEEK1$WIS
)

head(df_W1_NoLg)
```

```
##     STATE Julian_date epiweek   AUTO_AR  AUTO_ADJ  AUTO_EPI AUTO_TEMP
## 1 Alabama  2022-10-29      43 102.94840 101.65743 116.57567 102.44553
## 2 Alabama  2022-11-05      44  79.69032  69.21610  74.91468  79.44376
## 3 Alabama  2022-11-12      45  62.70926  49.99889  35.82769 155.63500
## 4 Alabama  2022-11-19      46  48.13570  37.31865  16.86871  58.72064
## 5 Alabama  2022-11-26      47  62.47143  61.26857  71.46483  61.73031
## 6 Alabama  2022-12-03      48  92.04712  93.99361  98.09041 103.34642
##     AUTO_AVG    ES27_AR   ES27_ADJ   ES27_EPI  ES27_TEMP   ES27_AVG    ES64_AR
## 1 111.250523  90.719035  90.791737  92.673037  90.729616  90.986378  88.414739
## 2  47.411877   9.385328   7.062736   9.397495   9.483206   7.024391   3.165721
## 3  78.333058 153.181002 152.552002 153.031292 153.171927 157.087082 170.008450
## 4  37.777085  60.914034  72.241112  61.554400  62.060894  65.839393  68.968221
## 5  66.087165  25.397894  41.902088  24.594584  24.277733  24.098040  23.129469
## 6   6.332483  83.516853  66.325397  77.939015  88.073395   9.997281  81.674040
##     ES64_ADJ  ES64_EPI  ES64_TEMP   ES64_AVG
## 1  89.834739  93.67342  88.803267  90.337167
## 2   3.068386   3.17232   3.132254   3.094771
## 3 166.985799 169.74417 170.208228 176.077362
## 4  78.645976  67.15577  67.642403  70.855246
## 5  44.816094  23.61011  21.939414  23.700555
## 6  43.209734  74.32642  84.495765  13.384597
```

2 Weeks ahead

```r
# Creating new columns with Epidemiological weeks based on target_end_week
AUTO_ARIMA_WEEK2$epiweek <- MMWRweek(AUTO_ARIMA_WEEK2$target_end_date)$MMWRweek
AUTO_ADJACENT_WEEK2$epiweek <- MMWRweek(AUTO_ADJACENT_WEEK2$target_end_date)$MMWRweek
AUTO_EPIWEEK_WEEK2$epiweek <- MMWRweek(AUTO_EPIWEEK_WEEK2$target_end_date)$MMWRweek
AUTO_TEMPERATURE_WEEK2$epiweek <- MMWRweek(AUTO_TEMPERATURE_WEEK2$target_end_date)$MMWRweek
AUTO_AVERAGE_WEEK2$epiweek <- MMWRweek(AUTO_AVERAGE_WEEK2$target_end_date)$MMWRweek

ES27_ARIMA_WEEK2$epiweek <- MMWRweek(ES27_ARIMA_WEEK2$target_end_date)$MMWRweek
ES27_ADJACENT_WEEK2$epiweek <- MMWRweek(ES27_ADJACENT_WEEK2$target_end_date)$MMWRweek
ES27_EPIWEEK_WEEK2$epiweek <- MMWRweek(ES27_EPIWEEK_WEEK2$target_end_date)$MMWRweek
ES27_TEMPERATURE_WEEK2$epiweek <- MMWRweek(ES27_TEMPERATURE_WEEK2$target_end_date)$MMWRweek
ES27_AVERAGE_WEEK2$epiweek <- MMWRweek(ES27_AVERAGE_WEEK2$target_end_date)$MMWRweek

ES64_ARIMA_WEEK2$epiweek <- MMWRweek(ES64_ARIMA_WEEK2$target_end_date)$MMWRweek
ES64_ADJACENT_WEEK2$epiweek <- MMWRweek(ES64_ADJACENT_WEEK2$target_end_date)$MMWRweek
ES64_EPIWEEK_WEEK2$epiweek <- MMWRweek(ES64_EPIWEEK_WEEK2$target_end_date)$MMWRweek
ES64_TEMPERATURE_WEEK2$epiweek <- MMWRweek(ES64_TEMPERATURE_WEEK2$target_end_date)$MMWRweek
ES64_AVERAGE_WEEK2$epiweek <- MMWRweek(ES64_AVERAGE_WEEK2$target_end_date)$MMWRweek

# Dataframe that will be analysed for 2 Weeks Ahead
df_W2_NoLg <- data.frame(
  STATE = AUTO_ARIMA_WEEK2$State,
  Julian_date = AUTO_ARIMA_WEEK2$target_end_date,
  epiweek = AUTO_ARIMA_WEEK2$epiweek,
  AUTO_AR=AUTO_ARIMA_WEEK2$WIS,
  AUTO_ADJ=AUTO_ADJACENT_WEEK2$WIS,
  AUTO_EPI=AUTO_EPIWEEK_WEEK2$WIS,
  AUTO_TEMP=AUTO_TEMPERATURE_WEEK2$WIS,
  AUTO_AVG=AUTO_AVERAGE_WEEK2$WIS,

  ES27_AR=ES27_ARIMA_WEEK2$WIS,
  ES27_ADJ=ES27_ADJACENT_WEEK2$WIS,
  ES27_EPI=ES27_EPIWEEK_WEEK2$WIS,
  ES27_TEMP=ES27_TEMPERATURE_WEEK2$WIS,
  ES27_AVG=ES27_AVERAGE_WEEK2$WIS,

  ES64_AR=ES64_ARIMA_WEEK2$WIS,
  ES64_ADJ=ES64_ADJACENT_WEEK2$WIS,
  ES64_EPI=ES64_EPIWEEK_WEEK2$WIS,
  ES64_TEMP=ES64_TEMPERATURE_WEEK2$WIS,
  ES64_AVG=ES64_AVERAGE_WEEK2$WIS

)

head(df_W2_NoLg)
```

```
##     STATE Julian_date epiweek    AUTO_AR   AUTO_ADJ   AUTO_EPI  AUTO_TEMP
## 1 Alabama  2022-11-05      44 188.905312 183.765343 228.179513 185.137841
## 2 Alabama  2022-11-12      45  48.034143  39.218917  42.748513  48.165309
## 3 Alabama  2022-11-19      46  81.098716  69.656879  28.840078 357.027657
```

```
## 4 Alabama  2022-11-26      47  34.593933   6.491932  63.573618  67.475787
## 5 Alabama  2022-12-03      48 237.947010 270.813086 251.280702 236.723180
## 6 Alabama  2022-12-10      49   7.307581   7.344853   9.448372   7.854465
##     AUTO_AVG   ES27_AR  ES27_ADJ  ES27_EPI ES27_TEMP  ES27_AVG   ES64_AR
## 1 213.799620 156.55311 157.25555 157.82259 156.42450 155.52180 150.37049
## 2  11.773647 114.29758 117.34220 114.00493 113.59390 111.78869 155.92213
## 3 100.827045 351.92105 345.19642 353.19759 351.90586 340.90754 401.78478
## 4   8.185654  65.67085  71.09070  66.92643  67.50840  69.30220  82.35596
## 5 254.196723 144.76162 167.68187 143.08761 143.31461 143.52997 133.83584
## 6  85.378319  18.31165  28.74939  25.30183  16.05396  97.09043  15.94977
##    ES64_ADJ  ES64_EPI ES64_TEMP  ES64_AVG
## 1 154.16654 159.51410 150.98968 154.60212
## 2 151.59357 157.08640 156.57325 158.11796
## 3 394.80115 402.05978 402.16290 403.14849
## 4  88.55030  79.36624  78.91779  81.12383
## 5 175.05363 135.69637 132.76973 137.25642
## 6  64.03856  24.95665  15.18490  83.27090
```

3 Weeks ahead

```r
# Creating new columns with Epidemiological weeks based on target_end_week
AUTO_ARIMA_WEEK3$epiweek <- MMWRweek(AUTO_ARIMA_WEEK3$target_end_date)$MMWRweek
AUTO_ADJACENT_WEEK3$epiweek <- MMWRweek(AUTO_ADJACENT_WEEK3$target_end_date)$MMWRweek
AUTO_EPIWEEK_WEEK3$epiweek <- MMWRweek(AUTO_EPIWEEK_WEEK3$target_end_date)$MMWRweek
AUTO_TEMPERATURE_WEEK3$epiweek <- MMWRweek(AUTO_TEMPERATURE_WEEK3$target_end_date)$MMWRweek
AUTO_AVERAGE_WEEK3$epiweek <- MMWRweek(AUTO_AVERAGE_WEEK3$target_end_date)$MMWRweek

ES27_ARIMA_WEEK3$epiweek <- MMWRweek(ES27_ARIMA_WEEK3$target_end_date)$MMWRweek
ES27_ADJACENT_WEEK3$epiweek <- MMWRweek(ES27_ADJACENT_WEEK3$target_end_date)$MMWRweek
ES27_EPIWEEK_WEEK3$epiweek <- MMWRweek(ES27_EPIWEEK_WEEK3$target_end_date)$MMWRweek
ES27_TEMPERATURE_WEEK3$epiweek <- MMWRweek(ES27_TEMPERATURE_WEEK3$target_end_date)$MMWRweek
ES27_AVERAGE_WEEK3$epiweek <- MMWRweek(ES27_AVERAGE_WEEK3$target_end_date)$MMWRweek

ES64_ARIMA_WEEK3$epiweek <- MMWRweek(ES64_ARIMA_WEEK3$target_end_date)$MMWRweek
ES64_ADJACENT_WEEK3$epiweek <- MMWRweek(ES64_ADJACENT_WEEK3$target_end_date)$MMWRweek
ES64_EPIWEEK_WEEK3$epiweek <- MMWRweek(ES64_EPIWEEK_WEEK3$target_end_date)$MMWRweek
ES64_TEMPERATURE_WEEK3$epiweek <- MMWRweek(ES64_TEMPERATURE_WEEK3$target_end_date)$MMWRweek
ES64_AVERAGE_WEEK3$epiweek <- MMWRweek(ES64_AVERAGE_WEEK3$target_end_date)$MMWRweek


# Dataframe that will be analysed for 3 Weeks Ahead
df_W3_NoLg <- data.frame(
  STATE = AUTO_ARIMA_WEEK3$State,
  Julian_date = AUTO_ARIMA_WEEK3$target_end_date,
  epiweek = AUTO_ARIMA_WEEK3$epiweek,
  AUTO_AR=AUTO_ARIMA_WEEK3$WIS,
  AUTO_ADJ=AUTO_ADJACENT_WEEK3$WIS,
  AUTO_EPI=AUTO_EPIWEEK_WEEK3$WIS,
  AUTO_TEMP=AUTO_TEMPERATURE_WEEK3$WIS,
  AUTO_AVG=AUTO_AVERAGE_WEEK3$WIS,

  ES27_AR=ES27_ARIMA_WEEK3$WIS,
  ES27_ADJ=ES27_ADJACENT_WEEK3$WIS,
```

```
  ES27_EPI=ES27_EPIWEEK_WEEK3$WIS,
  ES27_TEMP=ES27_TEMPERATURE_WEEK3$WIS,
  ES27_AVG=ES27_AVERAGE_WEEK3$WIS,

  ES64_AR=ES64_ARIMA_WEEK3$WIS,
  ES64_ADJ=ES64_ADJACENT_WEEK3$WIS,
  ES64_EPI=ES64_EPIWEEK_WEEK3$WIS,
  ES64_TEMP=ES64_TEMPERATURE_WEEK3$WIS,
  ES64_AVG=ES64_AVERAGE_WEEK3$WIS
)

head(df_W3_NoLg)
```

```
##     STATE Julian_date epiweek   AUTO_AR  AUTO_ADJ  AUTO_EPI AUTO_TEMP
## 1 Alabama  2022-11-12      45 146.78553 132.24099 226.16803 134.66404
## 2 Alabama  2022-11-19      46  48.22303  39.47050  41.58940  48.75787
## 3 Alabama  2022-11-26      47  10.19427  10.10276  41.42233 478.20811
## 4 Alabama  2022-12-03      48  43.25931 137.56318 328.57147  10.67010
## 5 Alabama  2022-12-10      49 146.32655 199.26107 146.15502 144.52135
## 6 Alabama  2022-12-17      50  59.17719  47.79918  62.09687  58.76758
##     AUTO_AVG   ES27_AR  ES27_ADJ  ES27_EPI ES27_TEMP  ES27_AVG   ES64_AR
## 1 177.978661  83.23148  84.55082  82.15597  82.94497  72.72026  70.96738
## 2   7.153594 289.38002 291.27312 288.86154 288.37530 279.21031 376.54043
## 3  28.768875 470.10331 457.53116 473.27827 470.02991 445.94359 575.24763
## 4 113.515485  12.58747  11.86176  12.64650  12.58217  12.27429  14.35102
## 5 174.117903  30.72114  10.88272  29.74189  30.00309  10.73607  23.73183
## 6 147.677074  83.17768  99.08706  93.65631  78.85432 200.26034  80.44222
##    ES64_ADJ  ES64_EPI ES64_TEMP  ES64_AVG
## 1  77.42024  77.55526  71.70535  73.02268
## 2 372.94951 379.43593 377.65242 383.92138
## 3 565.90367 578.05058 575.84232 575.00321
## 4  16.34616  14.23757  13.84466  12.75210
## 5  17.03998  26.52609  24.55276  11.48884
## 6 162.09347  92.96110  78.79480 178.76213
```

4 Weeks ahead

```
# Creating new columns with Epidemiological weeks based on target_end_week
AUTO_ARIMA_WEEK4$epiweek <- MMWRweek(AUTO_ARIMA_WEEK4$target_end_date)$MMWRweek
AUTO_ADJACENT_WEEK4$epiweek <- MMWRweek(AUTO_ADJACENT_WEEK4$target_end_date)$MMWRweek
AUTO_EPIWEEK_WEEK4$epiweek <- MMWRweek(AUTO_EPIWEEK_WEEK4$target_end_date)$MMWRweek
AUTO_TEMPERATURE_WEEK4$epiweek <- MMWRweek(AUTO_TEMPERATURE_WEEK4$target_end_date)$MMWRweek
AUTO_AVERAGE_WEEK4$epiweek <- MMWRweek(AUTO_AVERAGE_WEEK4$target_end_date)$MMWRweek

ES27_ARIMA_WEEK4$epiweek <- MMWRweek(ES27_ARIMA_WEEK4$target_end_date)$MMWRweek
ES27_ADJACENT_WEEK4$epiweek <- MMWRweek(ES27_ADJACENT_WEEK4$target_end_date)$MMWRweek
ES27_EPIWEEK_WEEK4$epiweek <- MMWRweek(ES27_EPIWEEK_WEEK4$target_end_date)$MMWRweek
ES27_TEMPERATURE_WEEK4$epiweek <- MMWRweek(ES27_TEMPERATURE_WEEK4$target_end_date)$MMWRweek
ES27_AVERAGE_WEEK4$epiweek <- MMWRweek(ES27_AVERAGE_WEEK4$target_end_date)$MMWRweek

ES64_ARIMA_WEEK4$epiweek <- MMWRweek(ES64_ARIMA_WEEK4$target_end_date)$MMWRweek
```

```r
ES64_ADJACENT_WEEK4$epiweek <- MMWRweek(ES64_ADJACENT_WEEK4$target_end_date)$MMWRweek
ES64_EPIWEEK_WEEK4$epiweek <- MMWRweek(ES64_EPIWEEK_WEEK4$target_end_date)$MMWRweek
ES64_TEMPERATURE_WEEK4$epiweek <- MMWRweek(ES64_TEMPERATURE_WEEK4$target_end_date)$MMWRweek
ES64_AVERAGE_WEEK4$epiweek <- MMWRweek(ES64_AVERAGE_WEEK4$target_end_date)$MMWRweek

# Dataframe that will be analysed for 4 Weeks Ahead
df_W4_NoLg <- data.frame(
  STATE = AUTO_ARIMA_WEEK4$State,
  Julian_date = AUTO_ARIMA_WEEK4$target_end_date,
  epiweek = AUTO_ARIMA_WEEK4$epiweek,
  AUTO_AR=AUTO_ARIMA_WEEK4$WIS,
  AUTO_ADJ=AUTO_ADJACENT_WEEK4$WIS,
  AUTO_EPI=AUTO_EPIWEEK_WEEK4$WIS,
  AUTO_TEMP=AUTO_TEMPERATURE_WEEK4$WIS,
  AUTO_AVG=AUTO_AVERAGE_WEEK4$WIS,

  ES27_AR=ES27_ARIMA_WEEK4$WIS,
  ES27_ADJ=ES27_ADJACENT_WEEK4$WIS,
  ES27_EPI=ES27_EPIWEEK_WEEK4$WIS,
  ES27_TEMP=ES27_TEMPERATURE_WEEK4$WIS,
  ES27_AVG=ES27_AVERAGE_WEEK4$WIS,

  ES64_AR=ES64_ARIMA_WEEK4$WIS,
  ES64_ADJ=ES64_ADJACENT_WEEK4$WIS,
  ES64_EPI=ES64_EPIWEEK_WEEK4$WIS,
  ES64_TEMP=ES64_TEMPERATURE_WEEK4$WIS,
  ES64_AVG=ES64_AVERAGE_WEEK4$WIS
)

head(df_W4_NoLg)
```

```
##     STATE Julian_date epiweek   AUTO_AR  AUTO_ADJ  AUTO_EPI AUTO_TEMP  AUTO_AVG
## 1 Alabama  2022-11-19      46  77.27004  47.93548 195.75162  51.85643 104.40361
## 2 Alabama  2022-11-26      47 115.99444 106.26135 102.50278 116.05624  39.86127
## 3 Alabama  2022-12-03      48 192.02549 176.92592 265.49305 472.69279 126.51362
## 4 Alabama  2022-12-10      49  29.55947  46.55813 232.08536  91.59221  22.18855
## 5 Alabama  2022-12-17      50  81.71423  93.68834  53.08411  79.57274 115.23051
## 6 Alabama  2022-12-24      51  60.75983  37.79366  58.29653  55.23814 136.53276
##      ES27_AR  ES27_ADJ   ES27_EPI  ES27_TEMP  ES27_AVG   ES64_AR   ES64_ADJ
## 1   7.254855   7.16035   7.617808   7.312711  14.40510  11.59783   9.069458
## 2 386.488083 388.30036 385.697901 384.899766 369.92347 537.93279 538.339518
## 3 461.672199 443.53294 466.645841 461.529587 425.18023 639.83705 629.159196
## 4  84.869169  81.88044  87.162162  87.610216  84.65971 114.14164 115.632787
## 5  21.531612 113.58265  21.857114  21.762429  75.31410  29.71893  99.180700
## 6  83.471365  98.93852  94.066825  78.769468 226.31181  83.21651 181.867637
##     ES64_EPI ES64_TEMP  ES64_AVG
## 1   8.877403  11.48946  13.89780
## 2 543.121394 539.66233 557.08945
## 3 643.495810 640.74775 642.76341
## 4 110.388493 108.19323 105.29785
## 5  26.189449  27.70742  70.39027
## 6  95.721917  81.27869 202.11325
```

Filter only the flu season

```r
# Filter the dataframe for epiweek >= 40 or epiweek <= 20
filtered_df_W1_NoLg <- df_W1_NoLg %>%
  filter(epiweek >= 40 | epiweek <= 20)

# Display the filtered dataset
head(filtered_df_W1_NoLg)
```

```
##     STATE Julian_date epiweek   AUTO_AR  AUTO_ADJ  AUTO_EPI AUTO_TEMP
## 1 Alabama  2022-10-29      43 102.94840 101.65743 116.57567 102.44553
## 2 Alabama  2022-11-05      44  79.69032  69.21610  74.91468  79.44376
## 3 Alabama  2022-11-12      45  62.70926  49.99889  35.82769 155.63500
## 4 Alabama  2022-11-19      46  48.13570  37.31865  16.86871  58.72064
## 5 Alabama  2022-11-26      47  62.47143  61.26857  71.46483  61.73031
## 6 Alabama  2022-12-03      48  92.04712  93.99361  98.09041 103.34642
##     AUTO_AVG    ES27_AR   ES27_ADJ   ES27_EPI  ES27_TEMP   ES27_AVG    ES64_AR
## 1 111.250523  90.719035  90.791737  92.673037  90.729616  90.986378  88.414739
## 2  47.411877   9.385328   7.062736   9.397495   9.483206   7.024391   3.165721
## 3  78.333058 153.181002 152.552002 153.031292 153.171927 157.087082 170.008450
## 4  37.777085  60.914034  72.241112  61.554400  62.060894  65.839393  68.968221
## 5  66.087165  25.397894  41.902088  24.594584  24.277733  24.098040  23.129469
## 6   6.332483  83.516853  66.325397  77.939015  88.073395   9.997281  81.674040
##     ES64_ADJ  ES64_EPI  ES64_TEMP   ES64_AVG
## 1  89.834739  93.67342  88.803267  90.337167
## 2   3.068386   3.17232   3.132254   3.094771
## 3 166.985799 169.74417 170.208228 176.077362
## 4  78.645976  67.15577  67.642403  70.855246
## 5  44.816094  23.61011  21.939414  23.700555
## 6  43.209734  74.32642  84.495765  13.384597
```

```r
# Filter the dataframe for epiweek >= 40 or epiweek <= 20
filtered_df_W2_NoLg <- df_W2_NoLg %>%
  filter(epiweek >= 40 | epiweek <= 20)

# Display the filtered dataset
head(filtered_df_W2_NoLg)
```

```
##     STATE Julian_date epiweek    AUTO_AR   AUTO_ADJ   AUTO_EPI  AUTO_TEMP
## 1 Alabama  2022-11-05      44 188.905312 183.765343 228.179513 185.137841
## 2 Alabama  2022-11-12      45  48.034143  39.218917  42.748513  48.165309
## 3 Alabama  2022-11-19      46  81.098716  69.656879  28.840078 357.027657
## 4 Alabama  2022-11-26      47  34.593933   6.491932  63.573618  67.475787
## 5 Alabama  2022-12-03      48 237.947010 270.813086 251.280702 236.723180
## 6 Alabama  2022-12-10      49   7.307581   7.344853   9.448372   7.854465
##     AUTO_AVG   ES27_AR  ES27_ADJ  ES27_EPI ES27_TEMP  ES27_AVG   ES64_AR
## 1 213.799620 156.55311 157.25555 157.82259 156.42450 155.52180 150.37049
## 2  11.773647 114.29758 117.34220 114.00493 113.59390 111.78869 155.92213
## 3 100.827045 351.92105 345.19642 353.19759 351.90586 340.90754 401.78478
## 4   8.185654  65.67085  71.09070  66.92643  67.50840  69.30220  82.35596
## 5 254.196723 144.76162 167.68187 143.08761 143.31461 143.52997 133.83584
## 6  85.378319  18.31165  28.74939  25.30183  16.05396  97.09043  15.94977
##    ES64_ADJ  ES64_EPI ES64_TEMP  ES64_AVG
```

```
## 1 154.16654 159.51410 150.98968 154.60212
## 2 151.59357 157.08640 156.57325 158.11796
## 3 394.80115 402.05978 402.16290 403.14849
## 4  88.55030  79.36624  78.91779  81.12383
## 5 175.05363 135.69637 132.76973 137.25642
## 6  64.03856  24.95665  15.18490  83.27090
```

```
# Filter the dataframe for epiweek >= 40 or epiweek <= 20
filtered_df_W3_NoLg <- df_W3_NoLg %>%
  filter(epiweek >= 40 | epiweek <= 20)

# Display the filtered dataset
head(filtered_df_W3_NoLg)
```

```
##     STATE Julian_date epiweek   AUTO_AR  AUTO_ADJ  AUTO_EPI AUTO_TEMP
## 1 Alabama  2022-11-12      45 146.78553 132.24099 226.16803 134.66404
## 2 Alabama  2022-11-19      46  48.22303  39.47050  41.58940  48.75787
## 3 Alabama  2022-11-26      47  10.19427  10.10276  41.42233 478.20811
## 4 Alabama  2022-12-03      48  43.25931 137.56318 328.57147  10.67010
## 5 Alabama  2022-12-10      49 146.32655 199.26107 146.15502 144.52135
## 6 Alabama  2022-12-17      50  59.17719  47.79918  62.09687  58.76758
##     AUTO_AVG   ES27_AR  ES27_ADJ  ES27_EPI ES27_TEMP  ES27_AVG   ES64_AR
## 1 177.978661  83.23148  84.55082  82.15597  82.94497  72.72026  70.96738
## 2   7.153594 289.38002 291.27312 288.86154 288.37530 279.21031 376.54043
## 3  28.768875 470.10331 457.53116 473.27827 470.02991 445.94359 575.24763
## 4 113.515485  12.58747  11.86176  12.64650  12.58217  12.27429  14.35102
## 5 174.117903  30.72114  10.88272  29.74189  30.00309  10.73607  23.73183
## 6 147.677074  83.17768  99.08706  93.65631  78.85432 200.26034  80.44222
##    ES64_ADJ  ES64_EPI ES64_TEMP  ES64_AVG
## 1  77.42024  77.55526  71.70535  73.02268
## 2 372.94951 379.43593 377.65242 383.92138
## 3 565.90367 578.05058 575.84232 575.00321
## 4  16.34616  14.23757  13.84466  12.75210
## 5  17.03998  26.52609  24.55276  11.48884
## 6 162.09347  92.96110  78.79480 178.76213
```

```
# Filter the dataframe for epiweek >= 40 or epiweek <= 20
filtered_df_W4_NoLg <- df_W4_NoLg %>%
  filter(epiweek >= 40 | epiweek <= 20)

# Display the filtered dataset
head(filtered_df_W4_NoLg)
```

```
##     STATE Julian_date epiweek   AUTO_AR  AUTO_ADJ  AUTO_EPI AUTO_TEMP  AUTO_AVG
## 1 Alabama  2022-11-19      46  77.27004  47.93548 195.75162  51.85643 104.40361
## 2 Alabama  2022-11-26      47 115.99444 106.26135 102.50278 116.05624  39.86127
## 3 Alabama  2022-12-03      48 192.02549 176.92592 265.49305 472.69279 126.51362
## 4 Alabama  2022-12-10      49  29.55947  46.55813 232.08536  91.59221  22.18855
## 5 Alabama  2022-12-17      50  81.71423  93.68834  53.08411  79.57274 115.23051
## 6 Alabama  2022-12-24      51  60.75983  37.79366  58.29653  55.23814 136.53276
##      ES27_AR  ES27_ADJ   ES27_EPI  ES27_TEMP  ES27_AVG   ES64_AR   ES64_ADJ
## 1   7.254855   7.16035   7.617808   7.312711  14.40510  11.59783   9.069458
## 2 386.488083 388.30036 385.697901 384.899766 369.92347 537.93279 538.339518
```

```
## 3 461.672199 443.53294 466.645841 461.529587 425.18023 639.83705 629.159196
## 4  84.869169  81.88044  87.162162  87.610216  84.65971 114.14164 115.632787
## 5  21.531612 113.58265  21.857114  21.762429  75.31410  29.71893  99.180700
## 6  83.471365  98.93852  94.066825  78.769468 226.31181  83.21651 181.867637
##     ES64_EPI ES64_TEMP  ES64_AVG
## 1   8.877403  11.48946  13.89780
## 2 543.121394 539.66233 557.08945
## 3 643.495810 640.74775 642.76341
## 4 110.388493 108.19323 105.29785
## 5  26.189449  27.70742  70.39027
## 6  95.721917  81.27869 202.11325
```

Here we load the flusight baseline wis and replace with the AUTO_AR model for comparasion. We keep the name as AUTO_AR so we don't need to change the whole pipeline.

```r
flusight_baseline<-read.csv("flusight_wis_final.csv") # it was 3

flusight_wis<-data.frame(STATE=flusight_baseline$wis_by_STATE.STATE,
horizon=flusight_baseline$wis_by_STATE.horizon,
Julian_date=as.Date(flusight_baseline$wis_by_STATE.target_end_date),
AUTO_AR=flusight_baseline$wis_by_STATE.wis)


filtered_df_W1_NoLg <- filtered_df_W1_NoLg %>%
  select(-AUTO_AR) %>%  # remove existing AUTO_AR
  left_join(flusight_wis %>% filter(horizon == 0),
            by = c("STATE", "Julian_date"))

filtered_df_W2_NoLg <- filtered_df_W2_NoLg %>%
  select(-AUTO_AR) %>%  # remove existing AUTO_AR
  left_join(flusight_wis %>% filter(horizon == 1),
            by = c("STATE", "Julian_date"))

filtered_df_W3_NoLg <- filtered_df_W3_NoLg %>%
  select(-AUTO_AR) %>%  # remove existing AUTO_AR
  left_join(flusight_wis %>% filter(horizon == 2),
            by = c( "STATE", "Julian_date"))

filtered_df_W4_NoLg <- filtered_df_W4_NoLg %>%
  select(-AUTO_AR) %>%  # remove existing AUTO_AR
  left_join(flusight_wis %>% filter(horizon == 3),
            by = c( "STATE", "Julian_date"))
```

Computing mean weighted interval score for all target week for each model.

```r
# Define the function
calculate_mean_wis <- function(data) {
  data %>%
    group_by(STATE) %>%
    summarize(
      AUTO_AR = mean(AUTO_AR, na.rm = TRUE),
      AUTO_ADJ = mean(AUTO_ADJ, na.rm = TRUE),
      AUTO_EPI = mean(AUTO_EPI, na.rm = TRUE),
```

```r
      AUTO_TMP = mean(AUTO_TEMP, na.rm = TRUE),
      AUTO_AVG = mean(AUTO_AVG, na.rm = TRUE),

      ES27_AR = mean(ES27_AR, na.rm = TRUE),
      ES27_ADJ = mean(ES27_ADJ, na.rm = TRUE),
      ES27_EPI = mean(ES27_EPI, na.rm = TRUE),
      ES27_TMP = mean(ES27_TEMP, na.rm = TRUE),
      ES27_AVG = mean(ES27_AVG, na.rm = TRUE),

      ES64_AR = mean(ES64_AR, na.rm = TRUE),
      ES64_ADJ = mean(ES64_ADJ, na.rm = TRUE),
      ES64_EPI = mean(ES64_EPI, na.rm = TRUE),
      ES64_TMP = mean(ES64_TEMP, na.rm = TRUE),
      ES64_AVG = mean(ES64_AVG, na.rm = TRUE)
    )
}

# Now you can use the function with any dataframe
W1_NoLg <- calculate_mean_wis(filtered_df_W1_NoLg)
W2_NoLg <- calculate_mean_wis(filtered_df_W2_NoLg)
W3_NoLg <- calculate_mean_wis(filtered_df_W3_NoLg)
W4_NoLg <- calculate_mean_wis(filtered_df_W4_NoLg)

# Display the resulting dataframe
head(W1_NoLg)
```

```
## # A tibble: 6 x 16
##   STATE    AUTO_AR AUTO_ADJ AUTO_EPI AUTO_TMP AUTO_AVG ES27_AR ES27_ADJ ES27_EPI
##   <chr>      <dbl>    <dbl>    <dbl>    <dbl>    <dbl>   <dbl>    <dbl>    <dbl>
## 1 Alabama     19.2     19.4     24.2     25.4     19.5    23.7     21.7     23.8
## 2 Arizona     49.7     42.6     46.7     43.5     47.4    46.0     40.6     46.5
## 3 Arkansas    20.1     21.8     21.2     23.0     21.8    21.2     24.3     21.2
## 4 Califor~   113.      93.5    108.     112.     139.    112.     114.     114.
## 5 Colorado    18.4     17.3     17.3     18.6     17.9    17.3     17.7     16.5
## 6 Connect~    20.5     26.6     23.3     24.6     21.5    25.4     24.5     25.7
## # i 7 more variables: ES27_TMP <dbl>, ES27_AVG <dbl>, ES64_AR <dbl>,
## #   ES64_ADJ <dbl>, ES64_EPI <dbl>, ES64_TMP <dbl>, ES64_AVG <dbl>
```

```r
head(W2_NoLg)
```

```
## # A tibble: 6 x 16
##   STATE    AUTO_AR AUTO_ADJ AUTO_EPI AUTO_TMP AUTO_AVG ES27_AR ES27_ADJ ES27_EPI
##   <chr>      <dbl>    <dbl>    <dbl>    <dbl>    <dbl>   <dbl>    <dbl>    <dbl>
## 1 Alabama     29.7     30.0     36.4     41.8     32.7    42.1     38.8     42.3
## 2 Arizona     92.2    105.     105.      98.3    103.    107.      98.9    109.
## 3 Arkansas    34.1     37.7     36.6     39.0     36.5    37.0     42.0     37.2
## 4 Califor~   210.     168.     217.     245.     263.    240.     243.     242.
## 5 Colorado    31.7     27.5     29.9     31.0     27.8    28.7     29.2     27.4
## 6 Connect~    32.6     39.7     36.3     39.9     35.2    43.9     38.3     43.7
## # i 7 more variables: ES27_TMP <dbl>, ES27_AVG <dbl>, ES64_AR <dbl>,
## #   ES64_ADJ <dbl>, ES64_EPI <dbl>, ES64_TMP <dbl>, ES64_AVG <dbl>
```

```r
head(W3_NoLg)
```

```
## # A tibble: 6 x 16
##   STATE    AUTO_AR AUTO_ADJ AUTO_EPI AUTO_TMP AUTO_AVG ES27_AR ES27_ADJ ES27_EPI
##   <chr>      <dbl>    <dbl>    <dbl>    <dbl>    <dbl>   <dbl>    <dbl>    <dbl>
## 1 Alabama     34.7     35.6     44.2     46.1     39.3    53.1     47.8     53.4
## 2 Arizona    126.     152.     140.     135.     151.    151.     146.     155. 
## 3 Arkansas    47.0     51.2     47.6     53.6     48.4    51.1     57.0     51.3
## 4 Califor~   290.     256.     285.     376.     326.    388.     389.     391. 
## 5 Colorado    44.5     39.0     43.6     45.2     38.2    43.4     42.6     42.3
## 6 Connect~    46.5     53.8     51.8     57.1     52.9    64.6     52.6     64.6
## # i 7 more variables: ES27_TMP <dbl>, ES27_AVG <dbl>, ES64_AR <dbl>,
## #   ES64_ADJ <dbl>, ES64_EPI <dbl>, ES64_TMP <dbl>, ES64_AVG <dbl>
```

```r
head(W4_NoLg)
```

```
## # A tibble: 6 x 16
##   STATE    AUTO_AR AUTO_ADJ AUTO_EPI AUTO_TMP AUTO_AVG ES27_AR ES27_ADJ ES27_EPI
##   <chr>      <dbl>    <dbl>    <dbl>    <dbl>    <dbl>   <dbl>    <dbl>    <dbl>
## 1 Alabama     39.4     39.9     49.3     50.0     42.5    61.8     57.3     62.0
## 2 Arizona    157.     180.     162.     162.     177.    188.     176.     193. 
## 3 Arkansas    56.3     60.1     57.5     68.3     57.1    68.3     70.4     68.3
## 4 Califor~   348.     306.     343.     498.     354.    537.     529.     538. 
## 5 Colorado    55.7     47.0     56.1     56.1     45.1    56.6     53.7     55.8
## 6 Connect~    55.0     64.5     62.8     69.0     62.4    83.5     62.2     83.3
## # i 7 more variables: ES27_TMP <dbl>, ES27_AVG <dbl>, ES64_AR <dbl>,
## #   ES64_ADJ <dbl>, ES64_EPI <dbl>, ES64_TMP <dbl>, ES64_AVG <dbl>
```

Loading model with log-back transformations.

```r
load("models_with_logback/ES_ARIMA/ARIMA_MODELS_influenza_hospitalization.Rdata")
load("models_with_logback/ES_ADJACENT/ADJACENT_MODELS_influenza_hospitalization.Rdata")
load("models_with_logback/ES_EPIWEEK/EPIWEEK_MODELS_influenza_hospitalization.Rdata")
load("models_with_logback/ES_TEMPERATURE/TEMPERATURE_MODELS_influenza_hospitalization.Rdata")
load("models_with_logback/ES_AVERAGE/AVERAGE_MODELS_influenza_hospitalization.Rdata")
```

1 Week ahead

```r
# Creating new columns with Epidemiological weeks based on target_end_week
AUTO_ARIMA_WEEK1$epiweek <- MMWRweek(AUTO_ARIMA_WEEK1$target_end_date)$MMWRweek
AUTO_ADJACENT_WEEK1$epiweek <- MMWRweek(AUTO_ADJACENT_WEEK1$target_end_date)$MMWRweek
AUTO_EPIWEEK_WEEK1$epiweek <- MMWRweek(AUTO_EPIWEEK_WEEK1$target_end_date)$MMWRweek
AUTO_TEMPERATURE_WEEK1$epiweek <- MMWRweek(AUTO_TEMPERATURE_WEEK1$target_end_date)$MMWRweek
AUTO_AVERAGE_WEEK1$epiweek <- MMWRweek(AUTO_AVERAGE_WEEK1$target_end_date)$MMWRweek

ES27_ARIMA_WEEK1$epiweek <- MMWRweek(ES27_ARIMA_WEEK1$target_end_date)$MMWRweek
ES27_ADJACENT_WEEK1$epiweek <- MMWRweek(ES27_ADJACENT_WEEK1$target_end_date)$MMWRweek
ES27_EPIWEEK_WEEK1$epiweek <- MMWRweek(ES27_EPIWEEK_WEEK1$target_end_date)$MMWRweek
ES27_TEMPERATURE_WEEK1$epiweek <- MMWRweek(ES27_TEMPERATURE_WEEK1$target_end_date)$MMWRweek
ES27_AVERAGE_WEEK1$epiweek <- MMWRweek(ES27_AVERAGE_WEEK1$target_end_date)$MMWRweek
```

```r
ES64_ARIMA_WEEK1$epiweek <- MMWRweek(ES64_ARIMA_WEEK1$target_end_date)$MMWRweek
ES64_ADJACENT_WEEK1$epiweek <- MMWRweek(ES64_ADJACENT_WEEK1$target_end_date)$MMWRweek
ES64_EPIWEEK_WEEK1$epiweek <- MMWRweek(ES64_EPIWEEK_WEEK1$target_end_date)$MMWRweek
ES64_TEMPERATURE_WEEK1$epiweek <- MMWRweek(ES64_TEMPERATURE_WEEK1$target_end_date)$MMWRweek
ES64_AVERAGE_WEEK1$epiweek <- MMWRweek(ES64_AVERAGE_WEEK1$target_end_date)$MMWRweek

# Dataframe that will be analysed for 1 Week Ahead
df_W1 <- data.frame(
  STATE = AUTO_ARIMA_WEEK1$State,
  Julian_date = AUTO_ARIMA_WEEK1$target_end_date,
  epiweek = AUTO_ARIMA_WEEK1$epiweek,
  AUTO_AR_LB=AUTO_ARIMA_WEEK1$WIS,
  AUTO_ADJ_LB=AUTO_ADJACENT_WEEK1$WIS,
  AUTO_EPI_LB=AUTO_EPIWEEK_WEEK1$WIS,
  AUTO_TEMP_LB=AUTO_TEMPERATURE_WEEK1$WIS,
  AUTO_AVG_LB=AUTO_AVERAGE_WEEK1$WIS,

  ES27_AR_LB=ES27_ARIMA_WEEK1$WIS,
  ES27_ADJ_LB=ES27_ADJACENT_WEEK1$WIS,
  ES27_EPI_LB=ES27_EPIWEEK_WEEK1$WIS,
  ES27_TEMP_LB=ES27_TEMPERATURE_WEEK1$WIS,
  ES27_AVG_LB=ES27_AVERAGE_WEEK1$WIS,

  ES64_AR_LB=ES64_ARIMA_WEEK1$WIS,
  ES64_ADJ_LB=ES64_ADJACENT_WEEK1$WIS,
  ES64_EPI_LB=ES64_EPIWEEK_WEEK1$WIS,
  ES64_TEMP_LB=ES64_TEMPERATURE_WEEK1$WIS,
  ES64_AVG_LB=ES64_AVERAGE_WEEK1$WIS
)

head(df_W1)
```

```
##     STATE Julian_date epiweek AUTO_AR_LB AUTO_ADJ_LB AUTO_EPI_LB AUTO_TEMP_LB
## 1 Alabama  2022-10-29      43  164.97469   154.82262   167.16881    158.73461
## 2 Alabama  2022-11-05      44  152.25677   125.66999   147.56571    217.70136
## 3 Alabama  2022-11-12      45   42.76549    33.63300    42.10368     41.93627
## 4 Alabama  2022-11-19      46   27.64761    64.94758    27.33367     28.21504
## 5 Alabama  2022-11-26      47   30.84363    28.07577    29.48502     53.76514
## 6 Alabama  2022-12-03      48   70.42607    81.54591   130.42912     69.58822
##   AUTO_AVG_LB ES27_AR_LB ES27_ADJ_LB ES27_EPI_LB ES27_TEMP_LB ES27_AVG_LB
## 1   111.40345  138.67842   125.21088   160.28714    183.38737    91.96030
## 2    58.82784  100.18764    78.71110   105.33944    101.77054    46.03894
## 3    69.47533   34.89792    39.88211    34.55781     34.77695    87.57530
## 4   138.85755   60.46462   104.07048    59.09603     59.51544   170.95917
## 5    63.33534   34.33307    50.43779    33.09650     41.61223    95.06813
## 6    37.97748   65.48558    47.01762    63.64209     65.18264    41.62150
##   ES64_AR_LB ES64_ADJ_LB ES64_EPI_LB ES64_TEMP_LB ES64_AVG_LB
## 1  138.58970   133.18630   152.65640    180.91343    95.46299
## 2  104.66467    79.03814   109.79832    112.65048    46.02587
## 3   35.59296    40.77665    34.91187     35.43623    91.62425
## 4   65.23764   116.71887    64.70152     64.23071   187.23402
## 5   34.67596    39.50396    33.76894     43.53779    79.26431
## 6   65.56808    48.47490    60.83552     65.06090    40.40643
```

```
# Creating new columns with Epidemiological weeks based on target_end_week
AUTO_ARIMA_WEEK2$epiweek <- MMWRweek(AUTO_ARIMA_WEEK2$target_end_date)$MMWRweek
AUTO_ADJACENT_WEEK2$epiweek <- MMWRweek(AUTO_ADJACENT_WEEK2$target_end_date)$MMWRweek
AUTO_EPIWEEK_WEEK2$epiweek <- MMWRweek(AUTO_EPIWEEK_WEEK2$target_end_date)$MMWRweek
AUTO_TEMPERATURE_WEEK2$epiweek <- MMWRweek(AUTO_TEMPERATURE_WEEK2$target_end_date)$MMWRweek
AUTO_AVERAGE_WEEK2$epiweek <- MMWRweek(AUTO_AVERAGE_WEEK2$target_end_date)$MMWRweek

ES27_ARIMA_WEEK2$epiweek <- MMWRweek(ES27_ARIMA_WEEK2$target_end_date)$MMWRweek
ES27_ADJACENT_WEEK2$epiweek <- MMWRweek(ES27_ADJACENT_WEEK2$target_end_date)$MMWRweek
ES27_EPIWEEK_WEEK2$epiweek <- MMWRweek(ES27_EPIWEEK_WEEK2$target_end_date)$MMWRweek
ES27_TEMPERATURE_WEEK2$epiweek <- MMWRweek(ES27_TEMPERATURE_WEEK2$target_end_date)$MMWRweek
ES27_AVERAGE_WEEK2$epiweek <- MMWRweek(ES27_AVERAGE_WEEK2$target_end_date)$MMWRweek

ES64_ARIMA_WEEK2$epiweek <- MMWRweek(ES64_ARIMA_WEEK2$target_end_date)$MMWRweek
ES64_ADJACENT_WEEK2$epiweek <- MMWRweek(ES64_ADJACENT_WEEK2$target_end_date)$MMWRweek
ES64_EPIWEEK_WEEK2$epiweek <- MMWRweek(ES64_EPIWEEK_WEEK2$target_end_date)$MMWRweek
ES64_TEMPERATURE_WEEK2$epiweek <- MMWRweek(ES64_TEMPERATURE_WEEK2$target_end_date)$MMWRweek
ES64_AVERAGE_WEEK2$epiweek <- MMWRweek(ES64_AVERAGE_WEEK2$target_end_date)$MMWRweek

# Dataframe that will be analysed for 2 Week Ahead
df_W2 <- data.frame(
  STATE = AUTO_ARIMA_WEEK2$State,
  Julian_date = AUTO_ARIMA_WEEK2$target_end_date,
  epiweek = AUTO_ARIMA_WEEK2$epiweek,
  AUTO_AR_LB=AUTO_ARIMA_WEEK2$WIS,
  AUTO_ADJ_LB=AUTO_ADJACENT_WEEK2$WIS,
  AUTO_EPI_LB=AUTO_EPIWEEK_WEEK2$WIS,
  AUTO_TEMP_LB=AUTO_TEMPERATURE_WEEK2$WIS,
  AUTO_AVG_LB=AUTO_AVERAGE_WEEK2$WIS,

  ES27_AR_LB=ES27_ARIMA_WEEK2$WIS,
  ES27_ADJ_LB=ES27_ADJACENT_WEEK2$WIS,
  ES27_EPI_LB=ES27_EPIWEEK_WEEK2$WIS,
  ES27_TEMP_LB=ES27_TEMPERATURE_WEEK2$WIS,
  ES27_AVG_LB=ES27_AVERAGE_WEEK2$WIS,

  ES64_AR_LB=ES64_ARIMA_WEEK2$WIS,
  ES64_ADJ_LB=ES64_ADJACENT_WEEK2$WIS,
  ES64_EPI_LB=ES64_EPIWEEK_WEEK2$WIS,
  ES64_TEMP_LB=ES64_TEMPERATURE_WEEK2$WIS,
  ES64_AVG_LB=ES64_AVERAGE_WEEK2$WIS
)

head(df_W2)
```

```
##     STATE Julian_date epiweek AUTO_AR_LB AUTO_ADJ_LB AUTO_EPI_LB AUTO_TEMP_LB
## 1 Alabama  2022-11-05      44  277.52451   260.09905   285.18405    272.93892
## 2 Alabama  2022-11-12      45  123.76206    98.81682   119.43244    225.73550
## 3 Alabama  2022-11-19      46   27.65801    30.68134    27.41752     28.12358
## 4 Alabama  2022-11-26      47   29.02424    52.51845    28.08553     29.14046
```

```
## 5 Alabama  2022-12-03      48  168.36465    90.69261   158.96411     46.58232
## 6 Alabama  2022-12-10      49   28.16100    32.72021    80.13328     28.28261
##   AUTO_AVG_LB ES27_AR_LB ES27_ADJ_LB ES27_EPI_LB ES27_TEMP_LB ES27_AVG_LB
## 1   198.89474  231.81375   211.64222   267.41322    304.55539   162.85351
## 2    36.72868   63.48153    50.33202    67.64182     65.26766    35.19653
## 3   108.01831   59.66592    75.06981    56.00013     58.12578   149.80369
## 4    98.55272   65.08632   103.25240    62.44626     64.43724   153.72638
## 5    44.42643   56.73039    46.35015    60.59899     51.23473    42.80849
## 6    32.65251   29.66896    35.66245    30.87440     29.65345    70.94583
##   ES64_AR_LB ES64_ADJ_LB ES64_EPI_LB ES64_TEMP_LB ES64_AVG_LB
## 1  238.53237   222.99366   258.96427    299.73039   166.07500
## 2   69.05467    51.88106    72.42017     78.11781    36.04737
## 3   62.96120    76.63837    57.82458     60.28029   159.58156
## 4   56.75389    85.65467    56.00229     56.38154   135.42546
## 5   56.78263    47.00848    59.27148     50.22071    41.59863
## 6   29.36865    34.02798    30.95575     29.43398    72.17442
```

3 Weeks ahead

```
# Creating new columns with Epidemiological weeks based on target_end_week
AUTO_ARIMA_WEEK3$epiweek <- MMWRweek(AUTO_ARIMA_WEEK3$target_end_date)$MMWRweek
AUTO_ADJACENT_WEEK3$epiweek <- MMWRweek(AUTO_ADJACENT_WEEK3$target_end_date)$MMWRweek
AUTO_EPIWEEK_WEEK3$epiweek <- MMWRweek(AUTO_EPIWEEK_WEEK3$target_end_date)$MMWRweek
AUTO_TEMPERATURE_WEEK3$epiweek <- MMWRweek(AUTO_TEMPERATURE_WEEK3$target_end_date)$MMWRweek
AUTO_AVERAGE_WEEK3$epiweek <- MMWRweek(AUTO_AVERAGE_WEEK3$target_end_date)$MMWRweek

ES27_ARIMA_WEEK3$epiweek <- MMWRweek(ES27_ARIMA_WEEK3$target_end_date)$MMWRweek
ES27_ADJACENT_WEEK3$epiweek <- MMWRweek(ES27_ADJACENT_WEEK3$target_end_date)$MMWRweek
ES27_EPIWEEK_WEEK3$epiweek <- MMWRweek(ES27_EPIWEEK_WEEK3$target_end_date)$MMWRweek
ES27_TEMPERATURE_WEEK3$epiweek <- MMWRweek(ES27_TEMPERATURE_WEEK3$target_end_date)$MMWRweek
ES27_AVERAGE_WEEK3$epiweek <- MMWRweek(ES27_AVERAGE_WEEK3$target_end_date)$MMWRweek

ES64_ARIMA_WEEK3$epiweek <- MMWRweek(ES64_ARIMA_WEEK3$target_end_date)$MMWRweek
ES64_ADJACENT_WEEK3$epiweek <- MMWRweek(ES64_ADJACENT_WEEK3$target_end_date)$MMWRweek
ES64_EPIWEEK_WEEK3$epiweek <- MMWRweek(ES64_EPIWEEK_WEEK3$target_end_date)$MMWRweek
ES64_TEMPERATURE_WEEK3$epiweek <- MMWRweek(ES64_TEMPERATURE_WEEK3$target_end_date)$MMWRweek
ES64_AVERAGE_WEEK3$epiweek <- MMWRweek(ES64_AVERAGE_WEEK3$target_end_date)$MMWRweek

# Dataframe that will be analysed for 2 Week Ahead
df_W3 <- data.frame(
  STATE = AUTO_ARIMA_WEEK3$State,
  Julian_date = AUTO_ARIMA_WEEK3$target_end_date,
  epiweek = AUTO_ARIMA_WEEK3$epiweek,
  AUTO_AR_LB=AUTO_ARIMA_WEEK3$WIS,
  AUTO_ADJ_LB=AUTO_ADJACENT_WEEK3$WIS,
  AUTO_EPI_LB=AUTO_EPIWEEK_WEEK3$WIS,
  AUTO_TEMP_LB=AUTO_TEMPERATURE_WEEK3$WIS,
  AUTO_AVG_LB=AUTO_AVERAGE_WEEK3$WIS,

  ES27_AR_LB=ES27_ARIMA_WEEK3$WIS,
  ES27_ADJ_LB=ES27_ADJACENT_WEEK3$WIS,
  ES27_EPI_LB=ES27_EPIWEEK_WEEK3$WIS,
  ES27_TEMP_LB=ES27_TEMPERATURE_WEEK3$WIS,
```

```
  ES27_AVG_LB=ES27_AVERAGE_WEEK3$WIS,

  ES64_AR_LB=ES64_ARIMA_WEEK3$WIS,
  ES64_ADJ_LB=ES64_ADJACENT_WEEK3$WIS,
  ES64_EPI_LB=ES64_EPIWEEK_WEEK3$WIS,
  ES64_TEMP_LB=ES64_TEMPERATURE_WEEK3$WIS,
  ES64_AVG_LB=ES64_AVERAGE_WEEK3$WIS
)

head(df_W3)
```

```
##     STATE Julian_date epiweek AUTO_AR_LB AUTO_ADJ_LB AUTO_EPI_LB AUTO_TEMP_LB
## 1 Alabama  2022-11-12      45  255.67207   241.03987   259.01418    251.72006
## 2 Alabama  2022-11-19      46   96.15237    72.82592    93.74152    183.44917
## 3 Alabama  2022-11-26      47   39.85273    31.57650    40.18449     39.08883
## 4 Alabama  2022-12-03      48   99.81193    48.50555   137.20756     98.21383
## 5 Alabama  2022-12-10      49  110.10063    48.22772   102.41015     44.48408
## 6 Alabama  2022-12-17      50   46.25364    20.84313    52.68966     43.73901
##   AUTO_AVG_LB ES27_AR_LB ES27_ADJ_LB ES27_EPI_LB ES27_TEMP_LB ES27_AVG_LB
## 1   171.21369  196.06526   172.30067   234.10599    281.50630   123.97151
## 2    28.87514   38.74124    35.84129    39.54435     39.32222    47.37365
## 3    88.49128   65.89910    79.19608    64.75946     64.52593   147.40097
## 4    46.81402   54.26322    64.53571    55.82159     54.26457    83.05521
## 5    34.92782   39.63831    48.66583    39.05724     44.12713    74.54001
## 6    56.01685   47.13537    63.33375    49.56314     46.42059   119.68466
##   ES64_AR_LB ES64_ADJ_LB ES64_EPI_LB ES64_TEMP_LB ES64_AVG_LB
## 1  201.04627   187.07056   217.42038    276.50884   125.77841
## 2   42.95249    37.63603    42.86805     48.73175    50.21087
## 3   69.16687    80.13844    67.19970     66.66957   147.74137
## 4   54.91965    64.83746    55.47049     55.00449    84.74953
## 5   39.02389    48.27718    38.64341     44.30995    71.72136
## 6   45.95904    58.36465    48.61270     44.94409   109.02039
```

4 Weeks ahead

```
# Creating new columns with Epidemiological weeks based on target_end_week
AUTO_ARIMA_WEEK4$epiweek <- MMWRweek(AUTO_ARIMA_WEEK4$target_end_date)$MMWRweek
AUTO_ADJACENT_WEEK4$epiweek <- MMWRweek(AUTO_ADJACENT_WEEK4$target_end_date)$MMWRweek
AUTO_EPIWEEK_WEEK4$epiweek <- MMWRweek(AUTO_EPIWEEK_WEEK4$target_end_date)$MMWRweek
AUTO_TEMPERATURE_WEEK4$epiweek <- MMWRweek(AUTO_TEMPERATURE_WEEK4$target_end_date)$MMWRweek
AUTO_AVERAGE_WEEK4$epiweek <- MMWRweek(AUTO_AVERAGE_WEEK4$target_end_date)$MMWRweek

ES27_ARIMA_WEEK4$epiweek <- MMWRweek(ES27_ARIMA_WEEK4$target_end_date)$MMWRweek
ES27_ADJACENT_WEEK4$epiweek <- MMWRweek(ES27_ADJACENT_WEEK4$target_end_date)$MMWRweek
ES27_EPIWEEK_WEEK4$epiweek <- MMWRweek(ES27_EPIWEEK_WEEK4$target_end_date)$MMWRweek
ES27_TEMPERATURE_WEEK4$epiweek <- MMWRweek(ES27_TEMPERATURE_WEEK4$target_end_date)$MMWRweek
ES27_AVERAGE_WEEK4$epiweek <- MMWRweek(ES27_AVERAGE_WEEK4$target_end_date)$MMWRweek

ES64_ARIMA_WEEK4$epiweek <- MMWRweek(ES64_ARIMA_WEEK4$target_end_date)$MMWRweek
ES64_ADJACENT_WEEK4$epiweek <- MMWRweek(ES64_ADJACENT_WEEK4$target_end_date)$MMWRweek
ES64_EPIWEEK_WEEK4$epiweek <- MMWRweek(ES64_EPIWEEK_WEEK4$target_end_date)$MMWRweek
```

```r
ES64_TEMPERATURE_WEEK4$epiweek <- MMWRweek(ES64_TEMPERATURE_WEEK4$target_end_date)$MMWRweek
ES64_AVERAGE_WEEK4$epiweek <- MMWRweek(ES64_AVERAGE_WEEK4$target_end_date)$MMWRweek

# Dataframe that will be analysed for 2 Week Ahead
df_W4 <- data.frame(
  STATE = AUTO_ARIMA_WEEK4$State,
  Julian_date = AUTO_ARIMA_WEEK4$target_end_date,
  epiweek = AUTO_ARIMA_WEEK4$epiweek,
  AUTO_AR_LB=AUTO_ARIMA_WEEK4$WIS,
  AUTO_ADJ_LB=AUTO_ADJACENT_WEEK4$WIS,
  AUTO_EPI_LB=AUTO_EPIWEEK_WEEK4$WIS,
  AUTO_TEMP_LB=AUTO_TEMPERATURE_WEEK4$WIS,
  AUTO_AVG_LB=AUTO_AVERAGE_WEEK4$WIS,

  ES27_AR_LB=ES27_ARIMA_WEEK4$WIS,
  ES27_ADJ_LB=ES27_ADJACENT_WEEK4$WIS,
  ES27_EPI_LB=ES27_EPIWEEK_WEEK4$WIS,
  ES27_TEMP_LB=ES27_TEMPERATURE_WEEK4$WIS,
  ES27_AVG_LB=ES27_AVERAGE_WEEK4$WIS,

  ES64_AR_LB=ES64_ARIMA_WEEK4$WIS,
  ES64_ADJ_LB=ES64_ADJACENT_WEEK4$WIS,
  ES64_EPI_LB=ES64_EPIWEEK_WEEK4$WIS,
  ES64_TEMP_LB=ES64_TEMPERATURE_WEEK4$WIS,
  ES64_AVG_LB=ES64_AVERAGE_WEEK4$WIS
)

head(df_W4)
```

```
##     STATE Julian_date epiweek AUTO_AR_LB AUTO_ADJ_LB AUTO_EPI_LB AUTO_TEMP_LB
## 1 Alabama  2022-11-19      46  195.02213   188.18432   191.28263    184.49068
## 2 Alabama  2022-11-26      47  119.84900    93.23656   117.90577    206.13596
## 3 Alabama  2022-12-03      48  140.12527   106.19007   142.84636    145.43076
## 4 Alabama  2022-12-10      49   44.74843    39.45008    88.53987     44.36605
## 5 Alabama  2022-12-17      50   73.72079    27.81180    68.22627     94.55459
## 6 Alabama  2022-12-24      51   53.99946    21.96318    77.33404     53.39699
##   AUTO_AVG_LB ES27_AR_LB ES27_ADJ_LB ES27_EPI_LB ES27_TEMP_LB ES27_AVG_LB
## 1   108.69715  133.28250   110.64096   168.33451    214.95541    68.27622
## 2    30.53778   44.01597    40.89044    45.21651     44.79821    49.47867
## 3    49.11299   65.53483    65.59596    69.66975     65.34986    91.51636
## 4    72.19681   78.71868   113.28678    75.15470     78.07503   154.77010
## 5    55.57844   66.11022    83.59744    60.04147     76.71141   127.32916
## 6    48.81717   51.57630    65.28205    55.34849     51.35922   118.96839
##   ES64_AR_LB ES64_ADJ_LB ES64_EPI_LB ES64_TEMP_LB ES64_AVG_LB
## 1  140.62946   128.11925   153.57953    210.85208    72.38258
## 2   50.17820    43.45508    49.86112     57.17021    51.78548
## 3   68.83669    67.34827    70.94611     68.72892    96.78787
## 4   79.49211   115.03138    77.78400     78.82837   159.23288
## 5   65.00889    69.31000    61.05206     77.46774   110.23667
## 6   50.68669    61.17782    56.36945     50.28846   117.30146
```

Filter only the flu season

```r
# Filter the dataframe for epiweek >= 40 or epiweek <= 20
filtered_df_W1 <- df_W1 %>%
  filter(epiweek >= 40 | epiweek <= 20)

# Display the filtered dataset
head(filtered_df_W1)
```

```
##     STATE Julian_date epiweek AUTO_AR_LB AUTO_ADJ_LB AUTO_EPI_LB AUTO_TEMP_LB
## 1 Alabama  2022-10-29      43  164.97469   154.82262   167.16881    158.73461
## 2 Alabama  2022-11-05      44  152.25677   125.66999   147.56571    217.70136
## 3 Alabama  2022-11-12      45   42.76549    33.63300    42.10368     41.93627
## 4 Alabama  2022-11-19      46   27.64761    64.94758    27.33367     28.21504
## 5 Alabama  2022-11-26      47   30.84363    28.07577    29.48502     53.76514
## 6 Alabama  2022-12-03      48   70.42607    81.54591   130.42912     69.58822
##   AUTO_AVG_LB ES27_AR_LB ES27_ADJ_LB ES27_EPI_LB ES27_TEMP_LB ES27_AVG_LB
## 1   111.40345  138.67842   125.21088   160.28714    183.38737    91.96030
## 2    58.82784  100.18764    78.71110   105.33944    101.77054    46.03894
## 3    69.47533   34.89792    39.88211    34.55781     34.77695    87.57530
## 4   138.85755   60.46462   104.07048    59.09603     59.51544   170.95917
## 5    63.33534   34.33307    50.43779    33.09650     41.61223    95.06813
## 6    37.97748   65.48558    47.01762    63.64209     65.18264    41.62150
##   ES64_AR_LB ES64_ADJ_LB ES64_EPI_LB ES64_TEMP_LB ES64_AVG_LB
## 1  138.58970   133.18630   152.65640    180.91343    95.46299
## 2  104.66467    79.03814   109.79832    112.65048    46.02587
## 3   35.59296    40.77665    34.91187     35.43623    91.62425
## 4   65.23764   116.71887    64.70152     64.23071   187.23402
## 5   34.67596    39.50396    33.76894     43.53779    79.26431
## 6   65.56808    48.47490    60.83552     65.06090    40.40643
```

```r
# Filter the dataframe for epiweek >= 40 or epiweek <= 20
filtered_df_W2 <- df_W2 %>%
  filter(epiweek >= 40 | epiweek <= 20)

# Display the filtered dataset
head(filtered_df_W2)
```

```
##     STATE Julian_date epiweek AUTO_AR_LB AUTO_ADJ_LB AUTO_EPI_LB AUTO_TEMP_LB
## 1 Alabama  2022-11-05      44  277.52451   260.09905   285.18405    272.93892
## 2 Alabama  2022-11-12      45  123.76206    98.81682   119.43244    225.73550
## 3 Alabama  2022-11-19      46   27.65801    30.68134    27.41752     28.12358
## 4 Alabama  2022-11-26      47   29.02424    52.51845    28.08553     29.14046
## 5 Alabama  2022-12-03      48  168.36465    90.69261   158.96411     46.58232
## 6 Alabama  2022-12-10      49   28.16100    32.72021    80.13328     28.28261
##   AUTO_AVG_LB ES27_AR_LB ES27_ADJ_LB ES27_EPI_LB ES27_TEMP_LB ES27_AVG_LB
## 1   198.89474  231.81375   211.64222   267.41322    304.55539   162.85351
## 2    36.72868   63.48153    50.33202    67.64182     65.26766    35.19653
## 3   108.01831   59.66592    75.06981    56.00013     58.12578   149.80369
## 4    98.55272   65.08632   103.25240    62.44626     64.43724   153.72638
## 5    44.42643   56.73039    46.35015    60.59899     51.23473    42.80849
## 6    32.65251   29.66896    35.66245    30.87440     29.65345    70.94583
##   ES64_AR_LB ES64_ADJ_LB ES64_EPI_LB ES64_TEMP_LB ES64_AVG_LB
## 1  238.53237   222.99366   258.96427    299.73039   166.07500
## 2   69.05467    51.88106    72.42017     78.11781    36.04737
```

```
## 3   62.96120    76.63837    57.82458     60.28029    159.58156
## 4   56.75389    85.65467    56.00229     56.38154    135.42546
## 5   56.78263    47.00848    59.27148     50.22071     41.59863
## 6   29.36865    34.02798    30.95575     29.43398     72.17442
```

```r
# Filter the dataframe for epiweek >= 40 or epiweek <= 20
filtered_df_W3 <- df_W3 %>%
  filter(epiweek >= 40 | epiweek <= 20)

# Display the filtered dataset
head(filtered_df_W3)
```

```
##     STATE Julian_date epiweek AUTO_AR_LB AUTO_ADJ_LB AUTO_EPI_LB AUTO_TEMP_LB
## 1 Alabama  2022-11-12      45  255.67207   241.03987   259.01418    251.72006
## 2 Alabama  2022-11-19      46   96.15237    72.82592    93.74152    183.44917
## 3 Alabama  2022-11-26      47   39.85273    31.57650    40.18449     39.08883
## 4 Alabama  2022-12-03      48   99.81193    48.50555   137.20756     98.21383
## 5 Alabama  2022-12-10      49  110.10063    48.22772   102.41015     44.48408
## 6 Alabama  2022-12-17      50   46.25364    20.84313    52.68966     43.73901
##   AUTO_AVG_LB ES27_AR_LB ES27_ADJ_LB ES27_EPI_LB ES27_TEMP_LB ES27_AVG_LB
## 1   171.21369  196.06526   172.30067   234.10599    281.50630   123.97151
## 2    28.87514   38.74124    35.84129    39.54435     39.32222    47.37365
## 3    88.49128   65.89910    79.19608    64.75946     64.52593   147.40097
## 4    46.81402   54.26322    64.53571    55.82159     54.26457    83.05521
## 5    34.92782   39.63831    48.66583    39.05724     44.12713    74.54001
## 6    56.01685   47.13537    63.33375    49.56314     46.42059   119.68466
##   ES64_AR_LB ES64_ADJ_LB ES64_EPI_LB ES64_TEMP_LB ES64_AVG_LB
## 1  201.04627   187.07056   217.42038    276.50884   125.77841
## 2   42.95249    37.63603    42.86805     48.73175    50.21087
## 3   69.16687    80.13844    67.19970     66.66957   147.74137
## 4   54.91965    64.83746    55.47049     55.00449    84.74953
## 5   39.02389    48.27718    38.64341     44.30995    71.72136
## 6   45.95904    58.36465    48.61270     44.94409   109.02039
```

```r
# Filter the dataframe for epiweek >= 40 or epiweek <= 20
filtered_df_W4 <- df_W4 %>%
  filter(epiweek >= 40 | epiweek <= 20)

# Display the filtered dataset
head(filtered_df_W4)
```

```
##     STATE Julian_date epiweek AUTO_AR_LB AUTO_ADJ_LB AUTO_EPI_LB AUTO_TEMP_LB
## 1 Alabama  2022-11-19      46  195.02213   188.18432   191.28263    184.49068
## 2 Alabama  2022-11-26      47  119.84900    93.23656   117.90577    206.13596
## 3 Alabama  2022-12-03      48  140.12527   106.19007   142.84636    145.43076
## 4 Alabama  2022-12-10      49   44.74843    39.45008    88.53987     44.36605
## 5 Alabama  2022-12-17      50   73.72079    27.81180    68.22627     94.55459
## 6 Alabama  2022-12-24      51   53.99946    21.96318    77.33404     53.39699
##   AUTO_AVG_LB ES27_AR_LB ES27_ADJ_LB ES27_EPI_LB ES27_TEMP_LB ES27_AVG_LB
## 1   108.69715  133.28250   110.64096   168.33451    214.95541    68.27622
## 2    30.53778   44.01597    40.89044    45.21651     44.79821    49.47867
## 3    49.11299   65.53483    65.59596    69.66975     65.34986    91.51636
## 4    72.19681   78.71868   113.28678    75.15470     78.07503   154.77010
```

```
## 5    55.57844   66.11022    83.59744    60.04147     76.71141    127.32916
## 6    48.81717   51.57630    65.28205    55.34849     51.35922    118.96839
##   ES64_AR_LB ES64_ADJ_LB ES64_EPI_LB ES64_TEMP_LB ES64_AVG_LB
## 1  140.62946   128.11925   153.57953    210.85208    72.38258
## 2   50.17820    43.45508    49.86112     57.17021    51.78548
## 3   68.83669    67.34827    70.94611     68.72892    96.78787
## 4   79.49211   115.03138    77.78400     78.82837   159.23288
## 5   65.00889    69.31000    61.05206     77.46774   110.23667
## 6   50.68669    61.17782    56.36945     50.28846   117.30146
```

Calculate mean weighted interval score for influenza seasons on each model and each state.

```r
# Define the function
calculate_mean_wis <- function(data) {
  data %>%
    group_by(STATE) %>%
    summarize(
      AUTO_AR_LB = mean(AUTO_AR_LB, na.rm = TRUE),
      AUTO_ADJ_LB = mean(AUTO_ADJ_LB, na.rm = TRUE),
      AUTO_EPI_LB = mean(AUTO_EPI_LB, na.rm = TRUE),
      AUTO_TMP_LB = mean(AUTO_TEMP_LB, na.rm = TRUE),
      AUTO_AVG_LB = mean(AUTO_AVG_LB, na.rm = TRUE),

      ES27_AR_LB = mean(ES27_AR_LB, na.rm = TRUE),
      ES27_ADJ_LB = mean(ES27_ADJ_LB, na.rm = TRUE),
      ES27_EPI_LB = mean(ES27_EPI_LB, na.rm = TRUE),
      ES27_TMP_LB = mean(ES27_TEMP_LB, na.rm = TRUE),
      ES27_AVG_LB = mean(ES27_AVG_LB, na.rm = TRUE),

      ES64_AR_LB = mean(ES64_AR_LB, na.rm = TRUE),
      ES64_ADJ_LB = mean(ES64_ADJ_LB, na.rm = TRUE),
      ES64_EPI_LB = mean(ES64_EPI_LB, na.rm = TRUE),
      ES64_TMP_LB = mean(ES64_TEMP_LB, na.rm = TRUE),
      ES64_AVG_LB = mean(ES64_AVG_LB, na.rm = TRUE),
    )
}

# Now we can use the function with any dataframe
W1_Lg <- calculate_mean_wis(filtered_df_W1)
W2_Lg <- calculate_mean_wis(filtered_df_W2)
W3_Lg <- calculate_mean_wis(filtered_df_W3)
W4_Lg <- calculate_mean_wis(filtered_df_W4)

# Display the resulting dataframe
head(W1_Lg)
```

```
## # A tibble: 6 x 16
##   STATE    AUTO_AR_LB AUTO_ADJ_LB AUTO_EPI_LB AUTO_TMP_LB AUTO_AVG_LB ES27_AR_LB
##   <chr>         <dbl>       <dbl>       <dbl>       <dbl>       <dbl>      <dbl>
## 1 Alabama        21.5        18.7        22.4        24.0        19.4       20.9
## 2 Arizona        45.5        35.2        45.3        45.5        57.5       44.5
## 3 Arkansas       20.0        17.0        19.7        21.5        19.9       21.7
## 4 Califor~       81.6        81.0        94.9        94.6       111.        89.5
```

```
## 5 Colorado       16.8        17.8        17.2        18.1        18.4       17.4
## 6 Connect~       19.3        15.8        18.8        19.0        15.2       19.8
## # i 9 more variables: ES27_ADJ_LB <dbl>, ES27_EPI_LB <dbl>, ES27_TMP_LB <dbl>,
## #   ES27_AVG_LB <dbl>, ES64_AR_LB <dbl>, ES64_ADJ_LB <dbl>, ES64_EPI_LB <dbl>,
## #   ES64_TMP_LB <dbl>, ES64_AVG_LB <dbl>
```

```r
head(W2_Lg)
```

```
## # A tibble: 6 x 16
##   STATE    AUTO_AR_LB AUTO_ADJ_LB AUTO_EPI_LB AUTO_TMP_LB AUTO_AVG_LB ES27_AR_LB
##   <chr>         <dbl>       <dbl>       <dbl>       <dbl>       <dbl>      <dbl>
## 1 Alabama        31.2        25.7        30.9        30.8        26.4       28.7
## 2 Arizona        77.9        64.9        76.6        76.8        95.7       72.5
## 3 Arkansas       30.4        27.7        30.6        32.4        29.6       35.7
## 4 Califor~      160.        156.        190.        155.        194.       166.
## 5 Colorado       26.7        28.2        28.6        28.4        26.0       28.5
## 6 Connect~       31.4        25.2        29.2        30.2        24.0       31.9
## # i 9 more variables: ES27_ADJ_LB <dbl>, ES27_EPI_LB <dbl>, ES27_TMP_LB <dbl>,
## #   ES27_AVG_LB <dbl>, ES64_AR_LB <dbl>, ES64_ADJ_LB <dbl>, ES64_EPI_LB <dbl>,
## #   ES64_TMP_LB <dbl>, ES64_AVG_LB <dbl>
```

```r
head(W3_Lg)
```

```
## # A tibble: 6 x 16
##   STATE    AUTO_AR_LB AUTO_ADJ_LB AUTO_EPI_LB AUTO_TMP_LB AUTO_AVG_LB ES27_AR_LB
##   <chr>         <dbl>       <dbl>       <dbl>       <dbl>       <dbl>      <dbl>
## 1 Alabama        38.1        28.9        36.9        39.4        31.4       33.5
## 2 Arizona       106.         92.8       104.        104.        133.       100.
## 3 Arkansas       40.2        37.8        39.0        42.0        38.3       48.5
## 4 Califor~      249.        229.        285.        219.        278.       245.
## 5 Colorado       35.7        40.0        40.4        38.8        35.8       39.9
## 6 Connect~       46.1        37.7        40.8        43.9        35.3       45.9
## # i 9 more variables: ES27_ADJ_LB <dbl>, ES27_EPI_LB <dbl>, ES27_TMP_LB <dbl>,
## #   ES27_AVG_LB <dbl>, ES64_AR_LB <dbl>, ES64_ADJ_LB <dbl>, ES64_EPI_LB <dbl>,
## #   ES64_TMP_LB <dbl>, ES64_AVG_LB <dbl>
```

```r
head(W4_Lg)
```

```
## # A tibble: 6 x 16
##   STATE    AUTO_AR_LB AUTO_ADJ_LB AUTO_EPI_LB AUTO_TMP_LB AUTO_AVG_LB ES27_AR_LB
##   <chr>         <dbl>       <dbl>       <dbl>       <dbl>       <dbl>      <dbl>
## 1 Alabama        43.4        33.5        41.9        45.6        36.8       38.4
## 2 Arizona       132.        120.        130.        129.        171.       129.
## 3 Arkansas       50.3        47.8        48.9        52.0        48.8       64.6
## 4 Califor~      325.        290.        362.        270.        351.       315.
## 5 Colorado       44.1        51.5        51.5        48.2        45.5       50.7
## 6 Connect~       59.0        48.1        52.6        56.0        46.0       59.0
## # i 9 more variables: ES27_ADJ_LB <dbl>, ES27_EPI_LB <dbl>, ES27_TMP_LB <dbl>,
## #   ES27_AVG_LB <dbl>, ES64_AR_LB <dbl>, ES64_ADJ_LB <dbl>, ES64_EPI_LB <dbl>,
## #   ES64_TMP_LB <dbl>, ES64_AVG_LB <dbl>
```

COMBINING THE RESULTS WITH LOG-BACK AND NO LOG-BACK TRASNFORMATION

```r
W1<-merge(W1_Lg,W1_NoLg, by = "STATE")
W2<-merge(W2_Lg,W2_NoLg, by = "STATE")
W3<-merge(W3_Lg,W3_NoLg, by = "STATE")
W4<-merge(W4_Lg,W4_NoLg, by = "STATE")
```

Here I include a column with the best model based on the lowest mean(WIS) of each state.

```r
# BEST RESULT

cols <- colnames(W1)[-1] # get states names
W1$Best_Result <- character(nrow(W1)) # create an empty column for the best models

# Give me the model with lower WIS value
for (i in 1:nrow(W1)) {
  # Find the model with the minimum value on each row
  # based on the column name
  # save it in the best result
  W1$Best_Result[i] <- cols[which.min(W1[i, cols])]
}

# REORDER BY FREQUENCY
W1$Best_Result <- fct_infreq(W1$Best_Result)
# Create a new column to indicate if model has a log-back
# transformation or not
W1$Model_Type <- ifelse(grepl("_LB$", W1$Best_Result), "With log-back transformation", "Without log-bac

# Print the first rows
head(W1)
```

```
##         STATE AUTO_AR_LB AUTO_ADJ_LB AUTO_EPI_LB AUTO_TMP_LB AUTO_AVG_LB
## 1     Alabama   21.50088    18.71138    22.39703    23.98760    19.35844
## 2     Arizona   45.52219    35.18194    45.26514    45.54241    57.53076
## 3    Arkansas   19.96738    16.96916    19.65110    21.45397    19.86030
## 4  California   81.55021    80.96661    94.94076    94.56612   110.61367
## 5    Colorado   16.79576    17.77995    17.17619    18.14275    18.36797
## 6 Connecticut   19.25440    15.82928    18.75026    18.98928    15.19050
##   ES27_AR_LB ES27_ADJ_LB ES27_EPI_LB ES27_TMP_LB ES27_AVG_LB ES64_AR_LB
## 1   20.85880    19.25938    21.33316    21.86869    21.13189   20.99726
## 2   44.49070    36.40411    45.11440    43.39661    56.03339   42.60622
## 3   21.69890    17.41815    21.30174    22.17678    20.30253   21.23517
## 4   89.51986    82.96145    88.05622    96.47274   105.01014   89.10840
## 5   17.36470    16.52006    16.82767    18.19789    16.54575   16.61554
## 6   19.78691    17.76911    19.17224    19.24697    15.88075   19.24724
##   ES64_ADJ_LB ES64_EPI_LB ES64_TMP_LB ES64_AVG_LB   AUTO_AR AUTO_ADJ  AUTO_EPI
## 1    19.52444    21.33758    22.01204    21.30777  19.16339 19.39934  24.22010
## 2    36.35690    45.63922    43.18101    55.27515  49.73254 42.56012  46.74881
## 3    16.90517    21.00537    21.89732    18.97059  20.12105 21.82260  21.19340
## 4    88.40400    89.59663    96.06782   108.38833 113.05048 93.49131 107.92167
## 5    16.72593    16.40725    17.27189    16.59354  18.43517 17.27311  17.34732
## 6    18.63364    19.23542    18.82657    17.07286  20.54510 26.58338  23.31798
##    AUTO_TMP  AUTO_AVG   ES27_AR  ES27_ADJ  ES27_EPI  ES27_TMP  ES27_AVG
## 1  25.41574  19.49120  23.72509  21.68002  23.80977  24.33474  23.14540
## 2  43.48828  47.41803  45.98882  40.55615  46.50543  44.75483  45.97291
```

```
## 3  22.97657  21.78262  21.15178  24.31616  21.22730  21.75788  23.67186
## 4 112.41001 139.25447 112.28469 114.45865 114.18650 112.30299 133.07351
## 5  18.63101  17.91991  17.30915  17.67074  16.54699  17.92531  17.71241
## 6  24.55721  21.47748  25.41463  24.47919  25.66267  25.75117  21.18974
##      ES64_AR  ES64_ADJ  ES64_EPI  ES64_TMP  ES64_AVG Best_Result
## 1  24.60068  23.38985  24.23985  24.98227  23.98900 AUTO_ADJ_LB
## 2  47.32760  41.10185  47.88236  46.69213  47.08806 AUTO_ADJ_LB
## 3  22.81367  26.02472  22.92309  24.40946  25.80619 ES64_ADJ_LB
## 4 119.97893 124.88327 121.08555 121.35980 130.50423 AUTO_ADJ_LB
## 5  17.11431  17.59973  16.38063  17.69002  17.94483    ES64_EPI
## 6  26.77442  27.33642  26.98980  28.43992  21.76993 AUTO_AVG_LB
##                         Model_Type
## 1    With log-back transformation
## 2    With log-back transformation
## 3    With log-back transformation
## 4    With log-back transformation
## 5 Without log-back transformation
## 6    With log-back transformation
```

```r
################################################

# Extract the columns of interest
cols <- colnames(W2)[-1]  # get states names
W2$Best_Result <- character(nrow(W2)) # create an empty column for the best models
# Give me the model with lower WIS value
for (i in 1:nrow(W2)) {
  # Find the model with the minimum value on each row
  # based on the column name
  # save it in the best result
  W2$Best_Result[i] <- cols[which.min(W2[i, cols])]
}

# REORDER BY FREQUENCY
W2$Best_Result <- fct_infreq(W2$Best_Result)
# Create a new column to indicate if model has a log-back
# transformation or not
W2$Model_Type <- ifelse(grepl("_LB$", W2$Best_Result),  "With log-back transformation","Without log-bacl

# Print merged results
head(W2)
```

```
##         STATE AUTO_AR_LB AUTO_ADJ_LB AUTO_EPI_LB AUTO_TMP_LB AUTO_AVG_LB
## 1     Alabama   31.16976    25.68193    30.87672    30.83478    26.42760
## 2     Arizona   77.94043    64.87303    76.61272    76.77889    95.66276
## 3    Arkansas   30.37586    27.70001    30.62213    32.40354    29.58848
## 4  California  160.08310   156.43030   189.78418   155.17087   193.66267
## 5    Colorado   26.72887    28.24473    28.60433    28.36833    25.99055
## 6 Connecticut   31.41674    25.18517    29.20407    30.21597    23.97753
##   ES27_AR_LB ES27_ADJ_LB ES27_EPI_LB ES27_TMP_LB ES27_AVG_LB ES64_AR_LB
## 1   28.71067    26.55882    28.86954    29.18227    28.70554   28.55843
## 2   72.48261    65.31287    72.64752    69.27474    90.58515   69.83565
## 3   35.66330    28.47459    34.83340    35.30356    31.42918   34.36039
## 4  165.75755   154.26221   166.46913   161.08032   190.87587  166.77175
## 5   28.47549    26.23565    27.25269    29.09701    23.57854   26.25568
```

```
## 6   31.92099    27.19555    30.36840    30.02038    23.77804   29.72233
##   ES64_ADJ_LB ES64_EPI_LB ES64_TMP_LB ES64_AVG_LB   AUTO_AR  AUTO_ADJ  AUTO_EPI
## 1    26.72802    28.54364    28.94711    28.49263  29.70003  29.97398  36.38449
## 2    64.67340    72.26294    68.97624    87.16214  92.18751 104.67278 104.56702
## 3    27.65420    34.20213    34.79972    29.46256  34.08749  37.66000  36.64999
## 4   165.77120   174.51730   162.96468   189.74315 209.76301 168.19945 216.81694
## 5    27.03370    26.21377    26.85938    24.10903  31.66708  27.51315  29.91979
## 6    27.82723    29.27228    28.55528    24.63026  32.61786  39.70454  36.29880
##    AUTO_TMP  AUTO_AVG   ES27_AR  ES27_ADJ  ES27_EPI  ES27_TMP  ES27_AVG
## 1  41.80543  32.73492  42.08275  38.83911  42.32867  42.38859  43.75290
## 2  98.30048 103.21981 106.97999  98.93783 109.11005 106.30188 101.71638
## 3  39.01924  36.50152  36.96728  41.95746  37.21486  37.77238  40.39599
## 4 244.65152 262.81249 240.32979 243.04099 241.58930 242.91313 282.04360
## 5  31.01883  27.77390  28.68397  29.22517  27.39527  29.13544  29.00367
## 6  39.90806  35.17583  43.93562  38.25072  43.70789  46.07414  34.88137
##     ES64_AR  ES64_ADJ  ES64_EPI  ES64_TMP  ES64_AVG Best_Result
## 1  44.54860  43.92800  44.88587  44.86200  44.87401 AUTO_ADJ_LB
## 2 111.04490 100.54368 111.99654 111.43491 101.78799 ES64_ADJ_LB
## 3  39.76156  45.64826  39.72121  42.19981  44.18069 ES64_ADJ_LB
## 4 259.45167 263.40632 258.20380 260.15284 296.77384 ES27_ADJ_LB
## 5  28.35024  29.14537  27.25058  28.66675  29.54493 ES27_AVG_LB
## 6  48.33919  44.61704  48.15219  49.64837  36.21101 ES27_AVG_LB
##                     Model_Type
## 1 With log-back transformation
## 2 With log-back transformation
## 3 With log-back transformation
## 4 With log-back transformation
## 5 With log-back transformation
## 6 With log-back transformation
```

```r
######################################
# BEST RESULT
# Extract the columns of interest
cols <- colnames(W3)[-1] # get states names
W3$Best_Result <- character(nrow(W3))# create an empty column for the best models
# Give me the model with lower WIS value
for (i in 1:nrow(W3)) {
  # Find the model with the minimum value on each row
  # based on the column name
  # save it in the best result
  W3$Best_Result[i] <- cols[which.min(W3[i, cols])]
}

# REORDER BY FREQUENCY
W3$Best_Result <- fct_infreq(W3$Best_Result)
# Create a new column to indicate if model has a log-back
# transformation or not
W3$Model_Type <- ifelse(grepl("_LB$", W3$Best_Result),  "With log-back transformation","Without log-bac

# Print merged results
head(W3)
```

```
##         STATE AUTO_AR_LB AUTO_ADJ_LB AUTO_EPI_LB AUTO_TMP_LB AUTO_AVG_LB
## 1     Alabama   38.06767    28.85802    36.94506    39.35297    31.36858
```

```
## 2     Arizona  105.68808    92.84931   103.83354   103.95012   132.50762
## 3    Arkansas   40.16564    37.83843    38.98530    41.96317    38.34471
## 4  California  249.33610   228.93146   284.58703   218.78418   277.57507
## 5    Colorado   35.72667    40.00171    40.35217    38.82044    35.81360
## 6 Connecticut   46.05532    37.72490    40.79870    43.91198    35.32860
##   ES27_AR_LB ES27_ADJ_LB ES27_EPI_LB ES27_TMP_LB ES27_AVG_LB ES64_AR_LB
## 1   33.45276    30.53513    33.55950    34.18829    34.61166   33.28668
## 2  100.07908    91.84994    95.59706    96.00889   122.10859   92.95885
## 3   48.46757    38.40733    46.55857    46.64195    41.18293   45.25104
## 4  244.51737   221.19022   248.02803   229.05807   274.51717  252.39978
## 5   39.94547    37.34637    37.76580    40.46168    32.31140   35.34731
## 6   45.88978    39.08301    44.32235    42.91971    34.63863   42.37364
##   ES64_ADJ_LB ES64_EPI_LB ES64_TMP_LB ES64_AVG_LB   AUTO_AR  AUTO_ADJ  AUTO_EPI
## 1    30.79499    32.88516    34.00872    34.47802  34.65495  35.62976  44.16947
## 2    90.05179    94.39553    92.66650   117.28995 125.79800 152.44372 140.19591
## 3    36.40443    44.53144    45.03869    38.47283  46.98570  51.24779  47.60087
## 4   238.57143   267.09273   234.42772   269.05311 289.55178 255.92721 285.23593
## 5    38.77150    36.02127    36.51507    33.53203  44.47743  38.95800  43.56443
## 6    39.59984    41.61795    40.30865    34.91718  46.54781  53.78087  51.78381
##    AUTO_TMP  AUTO_AVG   ES27_AR  ES27_ADJ  ES27_EPI  ES27_TMP  ES27_AVG
## 1  46.05607  39.25860  53.05708  47.77327  53.43002  54.26636  54.87899
## 2 135.37528 151.11178 150.74098 145.74845 154.68408 150.05604 148.77660
## 3  53.62337  48.35269  51.07920  57.01549  51.29766  51.66749  54.81546
## 4 375.50795 326.25154 388.35449 388.89710 390.76645 397.99830 438.92134
## 5  45.20783  38.22422  43.44209  42.64511  42.28405  43.91010  42.01012
## 6  57.14880  52.91577  64.64686  52.62770  64.57240  67.41036  50.40027
##     ES64_AR  ES64_ADJ  ES64_EPI  ES64_TMP  ES64_AVG Best_Result
## 1  57.76643  55.91689  58.28371  58.87958  58.30163 AUTO_ADJ_LB
## 2 157.70389 148.17646 158.24416 157.63183 147.96381 ES64_ADJ_LB
## 3  53.31634  60.43716  52.70552  55.09474  59.46949 ES64_ADJ_LB
## 4 409.87994 415.11647 409.29568 416.00723 462.00180 AUTO_TMP_LB
## 5  42.75558  42.65950  41.73564  42.90381  42.79269 ES27_AVG_LB
## 6  74.42901  61.34672  74.46252  77.61053  55.22415 ES27_AVG_LB
##                  Model_Type
## 1 With log-back transformation
## 2 With log-back transformation
## 3 With log-back transformation
## 4 With log-back transformation
## 5 With log-back transformation
## 6 With log-back transformation
```

```r
###################################
# BEST RESULT
# Extract the columns of interest
cols <- colnames(W4)[-1] # get states names
W4$Best_Result <- character(nrow(W4)) # create an empty column for the best models

# Give me the model with lower WIS value
for (i in 1:nrow(W4)) {
  # Find the model with the minimum value on each row
  # based on the column name
  # save it in the best result
  W4$Best_Result[i] <- cols[which.min(W4[i, cols])]
}
```

```r
W4$Best_Result <- fct_infreq(W4$Best_Result)
# Create a new column to indicate if model has a log-back
# transformation or not
W4$Model_Type <- ifelse(grepl("_LB$", W4$Best_Result),  "With log-back transformation","Without log-bac
# Print merged results
head(W4)
```

```
##         STATE AUTO_AR_LB AUTO_ADJ_LB AUTO_EPI_LB AUTO_TMP_LB AUTO_AVG_LB
## 1     Alabama   43.42794    33.46697    41.85936    45.64649    36.77960
## 2     Arizona  131.50684   120.25171   129.57882   129.16855   170.59572
## 3    Arkansas   50.30320    47.78956    48.89116    52.00687    48.76410
## 4  California  324.71650   289.72878   361.88210   269.99206   351.12200
## 5    Colorado   44.12714    51.52363    51.46275    48.20536    45.52945
## 6 Connecticut   58.96374    48.12606    52.61345    56.02897    46.04524
##   ES27_AR_LB ES27_ADJ_LB ES27_EPI_LB ES27_TMP_LB ES27_AVG_LB ES64_AR_LB
## 1   38.39827    35.18243    38.88707    39.27315    40.92104   38.62733
## 2  129.08500   118.44168   121.14001   120.93762   155.75597  116.23067
## 3   64.58626    48.81809    61.34403    60.73860    52.45312   58.06495
## 4  315.24701   279.34082   316.04186   286.16079   344.18840  340.47004
## 5   50.73779    48.04605    47.84586    50.76351    42.03216   43.67938
## 6   58.95177    49.29686    57.15871    55.37101    44.83746   53.82269
##   ES64_ADJ_LB ES64_EPI_LB ES64_TMP_LB ES64_AVG_LB   AUTO_AR  AUTO_ADJ  AUTO_EPI
## 1    35.36200    38.61357    39.33035    41.06098  39.37741  39.93102  49.28308
## 2   115.06789   119.63801   116.76110   146.54151 156.98326 179.90852 161.64565
## 3    44.66368    56.52417    57.58076    47.41751  56.33809  60.05783  57.47855
## 4   297.92714   348.51372   293.91844   325.37360 347.65775 306.08016 342.73309
## 5    49.35154    44.92123    44.63028    43.15576  55.71557  46.98355  56.09067
## 6    49.51952    52.83570    51.37682    44.92182  54.95383  64.49127  62.79066
##    AUTO_TMP  AUTO_AVG   ES27_AR  ES27_ADJ  ES27_EPI  ES27_TMP  ES27_AVG
## 1  50.04248  42.45901  61.79601  57.30286  62.00075  63.49449  65.02397
## 2 162.11291 177.33227 187.73436 175.68197 193.36014 187.62784 181.94556
## 3  68.25641  57.05509  68.26235  70.37725  68.32286  68.97580  66.19615
## 4 498.37785 353.67732 536.84721 528.65652 537.78179 561.87951 567.14627
## 5  56.14517  45.05636  56.57592  53.72223  55.78747  56.60697  52.44099
## 6  69.02256  62.40533  83.53432  62.22517  83.30342  88.39400  61.55655
##     ES64_AR  ES64_ADJ  ES64_EPI  ES64_TMP  ES64_AVG Best_Result
## 1  69.27650  68.13969  69.50792  70.84880  71.31850 AUTO_ADJ_LB
## 2 199.30810 179.19030 200.95464 199.88634 181.99879 ES64_ADJ_LB
## 3  73.05803  75.95411  72.30468  76.37572  72.01390 ES64_ADJ_LB
## 4 580.63156 582.91973 579.32705 576.78219 633.52470 AUTO_TMP_LB
## 5  55.62434  53.71907  54.88341  55.41115  53.07662 ES27_AVG_LB
## 6 100.87372  70.62766 100.98942 104.22873  66.36596 ES27_AVG_LB
##                     Model_Type
## 1 With log-back transformation
## 2 With log-back transformation
## 3 With log-back transformation
## 4 With log-back transformation
## 5 With log-back transformation
## 6 With log-back transformation
```

Now we use a Wilcoxon test with Holm p-value adjustment for repeated comparisons to evaluate the differences in model performances.

1 WEEK AHEAD - Wilcoxon test with Holm p-value adjustment

```r
# Get all models names except baseline
model_types <- setdiff(names(W1), c("STATE", "AUTO_AR","Best_Result","Model_Type","Week_Ahead"))

# Perform Wilcoxon test for each model type against baseline
wilcox_results1 <- map_df(model_types, function(model) {
  test <- wilcox.test(W1[[model]], W1$AUTO_AR, paired = TRUE)
  data.frame(Model = model, p_value = test$p.value, W = test$statistic)
})

# p values adjustment
p_values<-wilcox_results1$p_value
p_values<-p.adjust(p_values, method = "holm")
p_values<-data.frame(p_values)

###########################################################################
p_values_wk1 <- data.frame(Model = model_types, PValue = p_values, WeekAhead=1)
p_values_wk1 <-drop_na(p_values_wk1)
p_values_wk1
```

```
##          Model     p_values WeekAhead
## 1   AUTO_AR_LB 9.873984e-01         1
## 2  AUTO_ADJ_LB 4.749010e-07         1
## 3  AUTO_EPI_LB 1.024409e-01         1
## 4  AUTO_TMP_LB 1.000000e+00         1
## 5  AUTO_AVG_LB 7.296603e-08         1
## 6   ES27_AR_LB 1.000000e+00         1
## 7  ES27_ADJ_LB 5.184158e-04         1
## 8  ES27_EPI_LB 1.000000e+00         1
## 9  ES27_TMP_LB 1.000000e+00         1
## 10 ES27_AVG_LB 1.348707e-06         1
## 11  ES64_AR_LB 1.000000e+00         1
## 12 ES64_ADJ_LB 2.130091e-03         1
## 13 ES64_EPI_LB 1.000000e+00         1
## 14 ES64_TMP_LB 1.000000e+00         1
## 15 ES64_AVG_LB 2.114293e-06         1
## 16    AUTO_ADJ 3.691573e-02         1
## 17    AUTO_EPI 3.691573e-02         1
## 18    AUTO_TMP 3.639236e-06         1
## 19    AUTO_AVG 5.875037e-01         1
## 20     ES27_AR 1.727131e-02         1
## 21    ES27_ADJ 9.775586e-04         1
## 22    ES27_EPI 9.639596e-03         1
## 23    ES27_TMP 4.383257e-04         1
## 24    ES27_AVG 1.158616e-01         1
## 25     ES64_AR 2.114293e-06         1
## 26    ES64_ADJ 5.480340e-06         1
## 27    ES64_EPI 4.749010e-07         1
## 28    ES64_TMP 4.040353e-08         1
## 29    ES64_AVG 1.253674e-03         1
```

2 WEEKS AHEAD - Wilcoxon test with Holm p-value adjustment

```r
# Get all models names except baseline
model_types <- setdiff(names(W2), c("STATE", "AUTO_AR","Best_Result","Model_Type","Week_Ahead"))

# Perform Wilcoxon test for each model type against baseline
wilcox_results2 <- map_df(model_types, function(model) {
  test <- wilcox.test(W2[[model]], W2$AUTO_AR, paired = TRUE)
  data.frame(Model = model, p_value = test$p.value, W = test$statistic)
})

# p values adjustment
p_values<-wilcox_results2$p_value
p_values<-p.adjust(p_values, method = "holm")
p_values<-data.frame(p_values)

##########################################################################
p_values_wk2 <- data.frame(Model = model_types, PValue = p_values, WeekAhead=2)
p_values_wk2 <-drop_na(p_values_wk2)
p_values_wk2
```

```
##          Model     p_values WeekAhead
## 1   AUTO_AR_LB 6.350568e-08         2
## 2  AUTO_ADJ_LB 6.330936e-12         2
## 3  AUTO_EPI_LB 4.469143e-08         2
## 4  AUTO_TMP_LB 2.483608e-05         2
## 5  AUTO_AVG_LB 3.677059e-11         2
## 6   ES27_AR_LB 1.469551e-03         2
## 7  ES27_ADJ_LB 9.851107e-10         2
## 8  ES27_EPI_LB 1.138204e-05         2
## 9  ES27_TMP_LB 8.833394e-03         2
## 10 ES27_AVG_LB 5.968559e-13         2
## 11  ES64_AR_LB 2.128287e-04         2
## 12 ES64_ADJ_LB 1.617622e-09         2
## 13 ES64_EPI_LB 3.936805e-06         2
## 14 ES64_TMP_LB 2.464577e-03         2
## 15 ES64_AVG_LB 2.060574e-13         2
## 16    AUTO_ADJ 2.729133e-01         2
## 17    AUTO_EPI 7.912913e-05         2
## 18    AUTO_TMP 3.122125e-11         2
## 19    AUTO_AVG 1.400091e-01         2
## 20     ES27_AR 2.350321e-07         2
## 21    ES27_ADJ 2.483608e-05         2
## 22    ES27_EPI 3.914529e-06         2
## 23    ES27_TMP 1.079177e-07         2
## 24    ES27_AVG 1.469551e-03         2
## 25     ES64_AR 1.245297e-10         2
## 26    ES64_ADJ 4.563432e-06         2
## 27    ES64_EPI 2.318217e-10         2
## 28    ES64_TMP 1.091394e-10         2
## 29    ES64_AVG 4.899212e-05         2
```

3 WEEKS AHEAD - Wilcoxon test with Holm p-value adjustment

```r
# Get all models names except baseline
model_types <- setdiff(names(W3), c("STATE", "AUTO_AR","Best_Result","Model_Type","Week_Ahead"))

# Perform Wilcoxon test for each model type against baseline
wilcox_results3 <- map_df(model_types, function(model) {
  test <- wilcox.test(W3[[model]], W3$AUTO_AR, paired = TRUE)
  data.frame(Model = model, p_value = test$p.value, W = test$statistic)
})

# p values adjustment
p_values<-wilcox_results3$p_value
p_values<-p.adjust(p_values, method = "holm")
p_values<-data.frame(p_values)

##########################################################################
p_values_wk3 <- data.frame(Model = model_types, PValue = p_values, WeekAhead=3)
p_values_wk3 <-drop_na(p_values_wk3)
p_values_wk3
```

```
##          Model     p_values WeekAhead
## 1   AUTO_AR_LB 9.745804e-11         3
## 2  AUTO_ADJ_LB 2.060574e-13         3
## 3  AUTO_EPI_LB 2.313527e-10         3
## 4  AUTO_TMP_LB 1.468905e-07         3
## 5  AUTO_AVG_LB 8.640200e-11         3
## 6   ES27_AR_LB 1.284934e-03         3
## 7  ES27_ADJ_LB 1.563194e-11         3
## 8  ES27_EPI_LB 3.021601e-06         3
## 9  ES27_TMP_LB 2.796021e-03         3
## 10 ES27_AVG_LB 2.060574e-13         3
## 11  ES64_AR_LB 7.926499e-05         3
## 12 ES64_ADJ_LB 5.158540e-12         3
## 13 ES64_EPI_LB 1.475448e-07         3
## 14 ES64_TMP_LB 1.299593e-05         3
## 15 ES64_AVG_LB 2.060574e-13         3
## 16    AUTO_ADJ 7.673054e-02         3
## 17    AUTO_EPI 2.448208e-04         3
## 18    AUTO_TMP 3.240075e-12         3
## 19    AUTO_AVG 7.022822e-02         3
## 20     ES27_AR 1.091962e-10         3
## 21    ES27_ADJ 3.387709e-08         3
## 22    ES27_EPI 6.115641e-10         3
## 23    ES27_TMP 3.694822e-13         3
## 24    ES27_AVG 1.873165e-06         3
## 25     ES64_AR 3.240075e-12         3
## 26    ES64_ADJ 1.012327e-08         3
## 27    ES64_EPI 6.416201e-12         3
## 28    ES64_TMP 1.776357e-12         3
## 29    ES64_AVG 2.505653e-07         3
```

4 WEEKS AHEAD - Wilcoxon test with Holm p-value adjustment

```r
# Get all models names except baseline
model_types <- setdiff(names(W4), c("STATE", "AUTO_AR","Best_Result","Model_Type","Week_Ahead"))

# Perform Wilcoxon test for each model type against baseline
wilcox_results4 <- map_df(model_types, function(model) {
  test <- wilcox.test(W4[[model]], W4$AUTO_AR, paired = TRUE)
  data.frame(Model = model, p_value = test$p.value, W = test$statistic)
})

# Perform Wilcoxon test for each model type against AUTO_AAR
p_values<-wilcox_results4$p_value
p_values<-p.adjust(p_values, method = "holm")
p_values<-data.frame(p_values)

##########################################################################
p_values_wk4 <- data.frame(Model = model_types, PValue = p_values, WeekAhead=4)
p_values_wk4 <-drop_na(p_values_wk4)
p_values_wk4
```

```
##          Model     p_values WeekAhead
## 1   AUTO_AR_LB 4.615604e-07         4
## 2  AUTO_ADJ_LB 7.027268e-12         4
## 3  AUTO_EPI_LB 2.521838e-07         4
## 4  AUTO_TMP_LB 1.087519e-05         4
## 5  AUTO_AVG_LB 8.648282e-08         4
## 6   ES27_AR_LB 4.936689e-01         4
## 7  ES27_ADJ_LB 7.236167e-11         4
## 8  ES27_EPI_LB 1.255290e-01         4
## 9  ES27_TMP_LB 4.936689e-01         4
## 10 ES27_AVG_LB 7.027268e-12         4
## 11  ES64_AR_LB 2.325804e-01         4
## 12 ES64_ADJ_LB 3.979039e-13         4
## 13 ES64_EPI_LB 2.204326e-03         4
## 14 ES64_TMP_LB 1.033283e-01         4
## 15 ES64_AVG_LB 1.044498e-11         4
## 16    AUTO_ADJ 4.887000e-01         4
## 17    AUTO_EPI 2.204326e-03         4
## 18    AUTO_TMP 1.455859e-09         4
## 19    AUTO_AVG 2.375250e-01         4
## 20     ES27_AR 8.526513e-13         4
## 21    ES27_ADJ 1.033879e-08         4
## 22    ES27_EPI 1.044498e-11         4
## 23    ES27_TMP 2.060574e-13         4
## 24    ES27_AVG 8.648282e-08         4
## 25     ES64_AR 3.979039e-13         4
## 26    ES64_ADJ 6.937825e-09         4
## 27    ES64_EPI 3.979039e-13         4
## 28    ES64_TMP 3.979039e-13         4
## 29    ES64_AVG 1.323035e-07         4
```

Let's calculate the mean(WIS) improvement relative to the AUTO ARIMA model. We will compare each model with the AUTO ARIMA model for the same states. Later we sum the results of these comparisons. Negative results indicate that there was a general improvement in the mean(WIS) for a given model type among all states.

```r
calculate_percentage_of_improvement <- function(data) {
  return(data.frame(
    AUTO_AR = (((data$AUTO_AR / data$AUTO_AR)-1) * 100),
    ES27_AR = (((data$ES27_AR/data$AUTO_AR)-1) * 100),
    ES64_AR = (((data$ES64_AR/data$AUTO_AR )-1) * 100),

    AUTO_ADJ = (((data$AUTO_ADJ/data$AUTO_AR)-1) * 100),
    ES27_ADJ = (((data$ES27_ADJ/data$AUTO_AR)-1) * 100),
    ES64_ADJ = (((data$ES64_ADJ/data$AUTO_AR)-1) * 100),

    AUTO_TMP = (((data$AUTO_TMP/data$AUTO_AR)-1) * 100),
    ES27_TMP = (((data$ES27_TMP/data$AUTO_AR)-1) * 100),
    ES64_TMP = (((data$ES64_TMP/data$AUTO_AR)-1) * 100),

    AUTO_EPI = (((data$AUTO_EPI/data$AUTO_AR)-1) * 100),
    ES27_EPI = (((data$ES27_EPI/data$AUTO_AR)-1) * 100),
    ES64_EPI = (((data$ES64_EPI/data$AUTO_AR)-1) * 100),

    AUTO_AVG = (((data$AUTO_AVG/data$AUTO_AR)-1) * 100),
    ES27_AVG = (((data$ES27_AVG/data$AUTO_AR)-1) * 100),
    ES64_AVG = (((data$ES64_AVG/data$AUTO_AR)-1) * 100),

    AUTO_AR_LB = (((data$AUTO_AR_LB/data$AUTO_AR)-1) * 100),
    ES27_AR_LB = (((data$ES27_AR_LB/data$AUTO_AR)-1) * 100),
    ES64_AR_LB = (((data$ES64_AR_LB/data$AUTO_AR)-1) * 100),

    AUTO_ADJ_LB = (((data$AUTO_ADJ_LB/data$AUTO_AR)-1) * 100),
    ES27_ADJ_LB = (((data$ES27_ADJ_LB/data$AUTO_AR)-1) * 100),
    ES64_ADJ_LB = (((data$ES64_ADJ_LB/data$AUTO_AR)-1) * 100),

    AUTO_TMP_LB = (((data$AUTO_TMP_LB/data$AUTO_AR)-1) * 100),
    ES27_TMP_LB = (((data$ES27_TMP_LB/data$AUTO_AR)-1) * 100),
    ES64_TMP_LB = (((data$ES64_TMP_LB/data$AUTO_AR)-1) * 100),

    AUTO_EPI_LB = (((data$AUTO_EPI_LB/data$AUTO_AR)-1) * 100),
    ES27_EPI_LB = (((data$ES27_EPI_LB/data$AUTO_AR)-1) * 100),
    ES64_EPI_LB = (((data$ES64_EPI_LB/data$AUTO_AR)-1) * 100),

    AUTO_AVG_LB = (((data$AUTO_AVG_LB/data$AUTO_AR)-1) * 100),
    ES27_AVG_LB = (((data$ES27_AVG_LB/data$AUTO_AR)-1) * 100),
    ES64_AVG_LB = (((data$ES64_AVG_LB/data$AUTO_AR)-1) * 100)
  ))
}

# Calculate percentage of improvement
W1_percentage_of_improvement <- calculate_percentage_of_improvement(W1)
W2_percentage_of_improvement <- calculate_percentage_of_improvement(W2)
W3_percentage_of_improvement <- calculate_percentage_of_improvement(W3)
W4_percentage_of_improvement <- calculate_percentage_of_improvement(W4)

head(W1_percentage_of_improvement)
```

```
##    AUTO_AR    ES27_AR    ES64_AR    AUTO_ADJ    ES27_ADJ    ES64_ADJ     AUTO_TMP
```

```
## 1       0 23.8042488 28.373358   1.231280  13.132491  22.054901  32.6265523
## 2       0 -7.5277087 -4.835754 -14.421996 -18.451491 -17.354216 -12.5556827
## 3       0  5.1226516 13.382093   8.456596  20.849388  29.340774  14.1916865
## 4       0 -0.6773917  6.128632 -17.301272   1.245607  10.466819  -0.5665376
## 5       0 -6.1079627 -7.164872  -6.303457  -4.146554  -4.531764   1.0623175
## 6       0 23.7016529 30.320226  29.390363  19.148529  33.055648  19.5283250
##     ES27_TMP  ES64_TMP  AUTO_EPI   ES27_EPI   ES64_EPI  AUTO_AVG  ES27_AVG
## 1  26.985609 30.364578 26.387355  24.246136  26.490438  1.710599 20.779287
## 2 -10.008971 -6.113521 -5.999553  -6.488939  -3.720260 -4.653923 -7.559694
## 3   8.134927 21.313034  5.329504   5.497963  13.925898  8.257859 17.647263
## 4  -0.661199  7.350094 -4.536747   1.004880   7.107499 23.179017 17.711583
## 5  -2.765678 -4.041968 -5.900954 -10.242229 -11.144639 -2.794940 -3.920508
## 6  25.339732 38.426795 13.496566  24.908940  31.368526  4.538223  3.137677
##    ES64_AVG  AUTO_AR_LB ES27_AR_LB ES64_AR_LB AUTO_ADJ_LB ES27_ADJ_LB
## 1 25.181406  12.1976936   8.847124   9.569653   -2.358705   0.5009126
## 2 -5.317400  -8.4659844 -10.540063 -14.329284  -29.257717 -26.8002154
## 3 28.254682  -0.7637429   7.841780   5.537103  -15.664645 -13.4331814
## 4 15.438900 -27.8638979 -20.814258 -21.178222  -28.380130 -26.6155721
## 5 -2.659781  -8.8928354  -5.806675  -9.870382   -3.554148 -10.3883160
## 6  5.961665  -6.2822928  -3.690365  -6.317145  -22.953526 -13.5116843
##   ES64_ADJ_LB AUTO_TMP_LB ES27_TMP_LB ES64_TMP_LB AUTO_EPI_LB ES27_EPI_LB
## 1    1.884095   25.174111   14.117051   14.865053   16.874040   11.322496
## 2  -26.895144   -8.425335  -12.740019  -13.173531   -8.982861   -9.285957
## 3  -15.982678    6.624487   10.216812    8.827944   -2.335606    5.867947
## 4  -21.801308  -16.350542  -14.664021  -15.022196  -16.019145  -22.108936
## 5   -9.271582   -1.586184   -1.287069   -6.310109   -6.829223   -8.719719
## 6   -9.303751   -7.572724   -6.318466   -8.364659   -8.736119   -6.682182
##   ES64_EPI_LB AUTO_AVG_LB ES27_AVG_LB ES64_AVG_LB
## 1   11.345543   1.0178393  10.2721995   11.190013
## 2   -8.230674  15.6803163  12.6694627   11.144826
## 3    4.394992  -1.2959056   0.9019587   -5.717696
## 4  -20.746355  -2.1555043  -7.1121728   -4.123957
## 5  -11.000277  -0.3645072 -10.2489500   -9.989753
## 6   -6.374680 -26.0626459 -22.7029818  -16.900602
```

```
head(W2_percentage_of_improvement)
```

```
##   AUTO_AR   ES27_AR   ES64_AR    AUTO_ADJ  ES27_ADJ  ES64_ADJ  AUTO_TMP
## 1       0 41.692623  49.99516   0.9223997 30.771307 47.905582 40.758902
## 2       0 16.046076  20.45547  13.5433407  7.322372  9.064315  6.631008
## 3       0  8.448233  16.64563  10.4804295 23.087585 33.915014 14.467943
## 4       0 14.572059  23.68800 -19.8145305 15.864565 25.573296 16.632347
## 5       0 -9.420232 -10.47409 -13.1175176 -7.711215 -7.963205 -2.047099
## 6       0 34.698056  48.19853  21.7263917 17.269241 36.787158 22.350346
##    ES27_TMP ES64_TMP  AUTO_EPI   ES27_EPI  ES64_EPI   AUTO_AVG  ES27_AVG
## 1 42.722409 51.05036 22.506601  42.520639  51.13075  10.218473 47.316036
## 2 15.310499 20.87853 13.428616  18.356651  21.48776  11.967238 10.336398
## 3 10.810112 23.79854  7.517430   9.174563  16.52726   7.081880 18.506789
## 4 15.803610 24.02227  3.362810  15.172501  23.09311  25.290200 34.458218
## 5 -7.994545 -9.47461 -5.517698 -13.489747 -13.94666 -12.294110 -8.410672
## 6 41.254352 52.21222 11.285044  33.999865  47.62522   7.842237  6.939491
##    ES64_AVG AUTO_AR_LB ES27_AR_LB  ES64_AR_LB AUTO_ADJ_LB ES27_ADJ_LB
## 1 51.090808   4.948594  -3.331150  -3.8437490   -13.52893   -10.57643
## 2 10.414070 -15.454458 -21.374807 -24.2460918   -29.62926   -29.15215
```

```
## 3 29.609703 -10.888519   4.622863   0.8006106   -18.73848   -16.46614
## 4 41.480544 -23.683824 -20.978656 -20.4951556   -25.42522   -26.45881
## 5 -6.701456 -15.594148 -10.078595 -17.0884086   -10.80729   -17.15165
## 6 11.015900  -3.682393  -2.136476  -8.8771173   -22.78716   -16.62373
##   ES64_ADJ_LB AUTO_TMP_LB ES27_TMP_LB ES64_TMP_LB AUTO_EPI_LB ES27_EPI_LB
## 1   -10.00675    3.820720   -1.743296   -2.535088    3.961915   -2.796235
## 2   -29.84582  -16.714440  -24.854527  -25.178331  -16.894694  -21.195923
## 3   -18.87285   -4.940061    3.567506    2.089427  -10.166056    2.188229
## 4   -20.97215  -26.025627  -23.208425  -22.310097   -9.524473  -20.639422
## 5   -14.63153  -10.416961   -8.115914  -15.182029   -9.671707  -13.940014
## 6   -14.68712   -7.363737   -7.963358  -12.455064  -10.466014   -6.896404
##   ES64_EPI_LB AUTO_AVG_LB ES27_AVG_LB ES64_AVG_LB
## 1  -3.8935445  -11.018257   -3.348430   -4.065324
## 2 -21.6130997    3.769761   -1.738162   -5.451253
## 3   0.3363153  -13.198415   -7.798489  -13.567808
## 4 -16.8026337   -7.675488   -9.004035   -9.544035
## 5 -17.2207550  -17.925650  -25.542423  -23.867224
## 6 -10.2568997  -26.489579  -27.101176  -24.488420
```

```r
head(W3_percentage_of_improvement)
```

```
##   AUTO_AR   ES27_AR   ES64_AR   AUTO_ADJ  ES27_ADJ  ES64_ADJ  AUTO_TMP
## 1       0 53.101018 66.690268   2.812913 37.854099 61.353269 32.898972
## 2       0 19.827805 25.362797  21.181359 15.859122 17.789204  7.613226
## 3       0  8.712229 13.473568   9.071048 21.346495 28.628841 14.127011
## 4       0 34.122639 41.556698 -11.612630 34.310034 43.365190 29.685941
## 5       0 -2.327787 -3.871277 -12.409507 -4.119652 -4.087308  1.642177
## 6       0 38.882724 59.898000  15.538989 13.061599 31.792924 22.774427
##    ES27_TMP ES64_TMP  AUTO_EPI  ES27_EPI  ES64_EPI   AUTO_AVG  ES27_AVG
## 1 56.590488 69.90235 27.455005 54.177171 68.182934  13.284251 58.358308
## 2 19.283334 25.30552 11.445262 22.962279 25.792276  20.122567 18.266269
## 3  9.964290 17.25855  1.309272  9.177193 12.173550   2.909385 16.664149
## 4 37.453236 43.67283 -1.490528 34.955635 41.354916  12.674677 51.586475
## 5 -1.275541 -3.53801 -2.052717 -4.931444 -6.164453 -14.059276 -5.547315
## 6 44.819616 66.73294 11.248649 38.722753 59.969993  13.680488  8.276350
##    ES64_AVG AUTO_AR_LB ES27_AR_LB ES64_AR_LB AUTO_ADJ_LB ES27_ADJ_LB
## 1 68.234629   9.847702  -3.469027  -3.948260   -16.72757   -11.88811
## 2 17.620166 -15.985880 -20.444617 -26.104668   -26.19174   -26.98616
## 3 26.569352 -14.515172   3.153876  -3.691889   -19.46819   -18.25739
## 4 59.557574 -13.888944 -15.553147 -12.830868   -20.93592   -23.60944
## 5 -3.787857 -19.674610 -10.189340 -20.527528   -10.06290   -16.03299
## 6 18.639636  -1.058026  -1.413652  -8.967475   -18.95451   -16.03684
##   ES64_ADJ_LB AUTO_TMP_LB ES27_TMP_LB ES64_TMP_LB AUTO_EPI_LB ES27_EPI_LB
## 1   -11.13826   13.556557  -1.3465940   -1.864751    6.608327  -3.1610185
## 2   -28.41556  -17.367426 -23.6801101  -26.337059  -17.460098 -24.0074876
## 3   -22.52019  -10.689469  -0.7316043   -4.143834  -17.027289  -0.9090456
## 4   -17.60664  -24.440398 -20.8921927  -19.037721   -1.714634 -14.3407001
## 5   -12.82881  -12.718785  -9.0287398  -17.902029   -9.274943 -15.0899546
## 6   -14.92653   -5.662632  -7.7943485  -13.403753  -12.350968  -4.7810246
##   ES64_EPI_LB AUTO_AVG_LB ES27_AVG_LB ES64_AVG_LB
## 1   -5.106896   -9.483107  -0.1249191  -0.5105342
## 2  -24.962615    5.333653  -2.9328045  -6.7632635
## 3   -5.223414  -18.390671 -12.3500631 -18.1179929
## 4   -7.756488   -4.136293  -5.1923754  -7.0794514
```

```
## 5  -19.012256  -19.479154 -27.3532646 -24.6088772
## 6  -10.590960  -24.102539 -25.5848394 -24.9864036
```

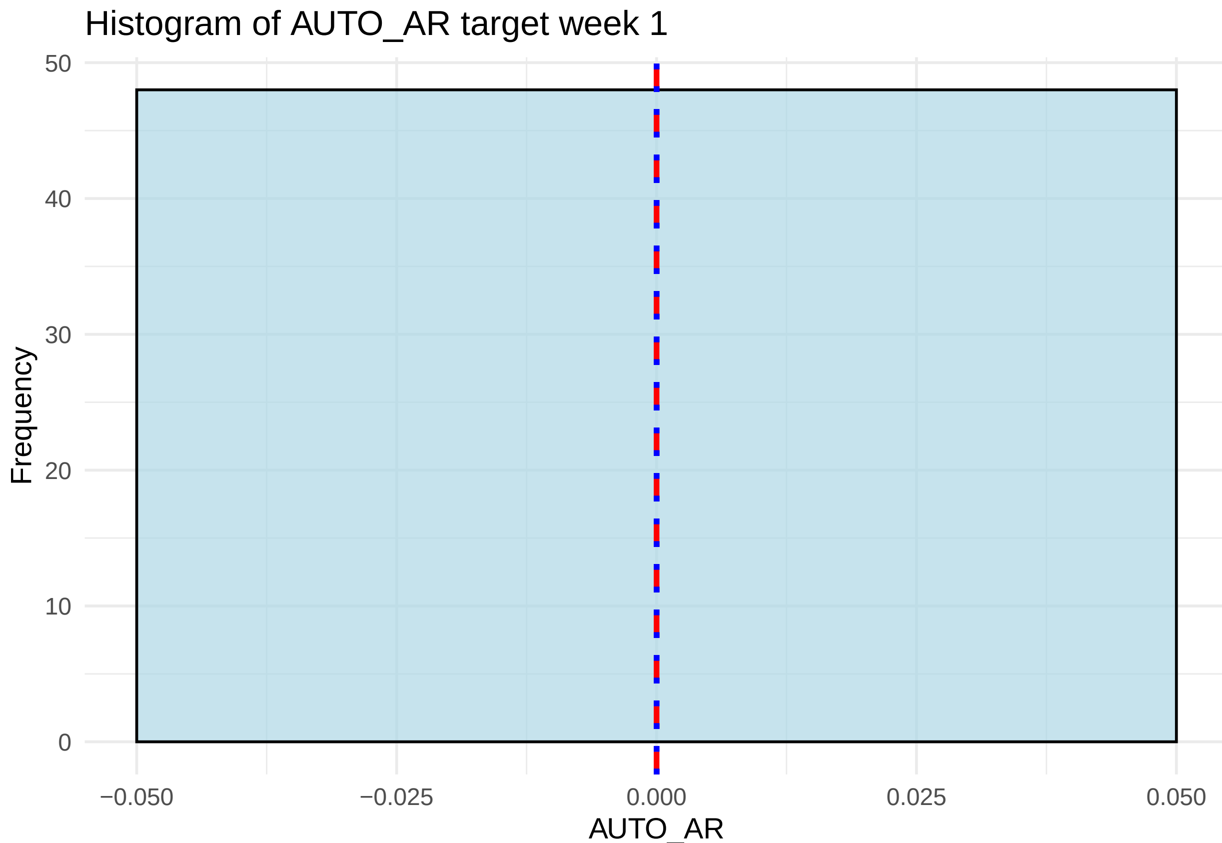
```r
head(W4_percentage_of_improvement)
```

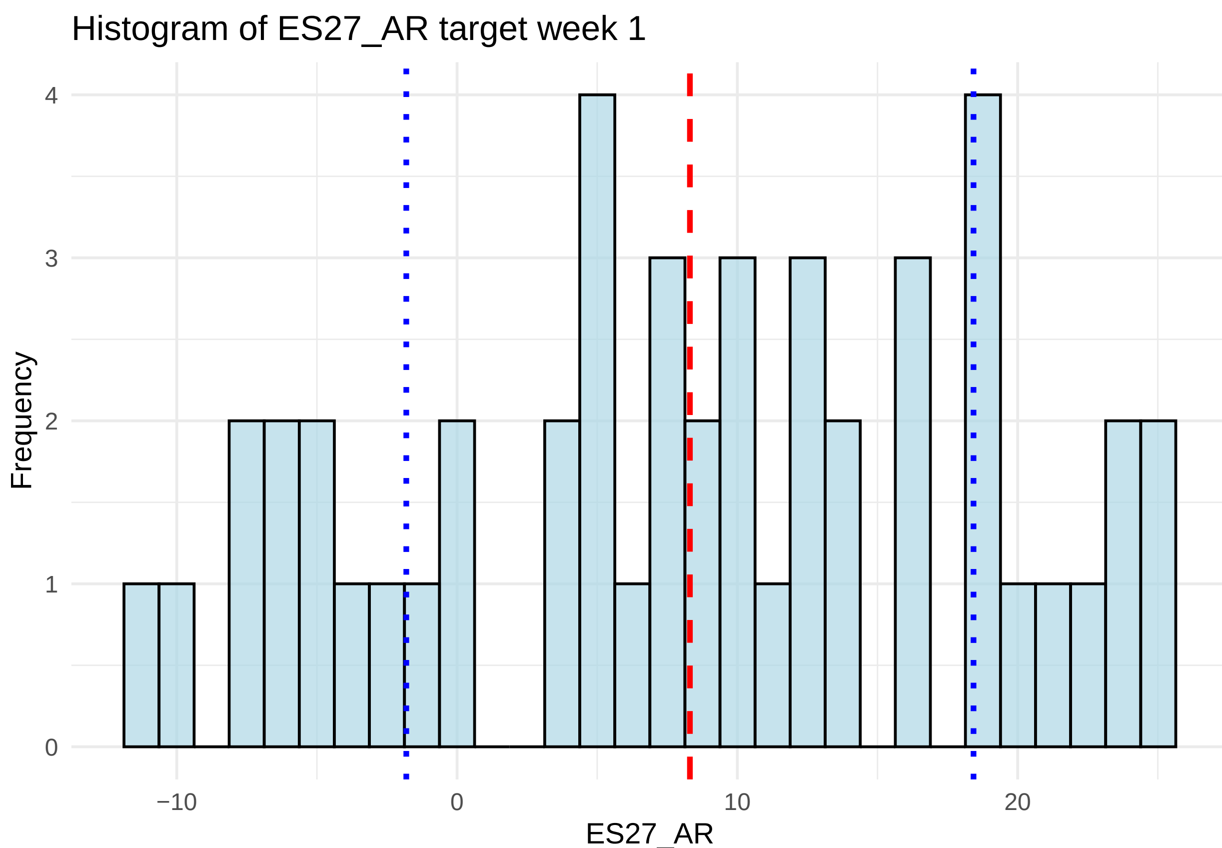
```
##   AUTO_AR   ES27_AR    ES64_AR   AUTO_ADJ  ES27_ADJ  ES64_ADJ   AUTO_TMP
## 1       0 56.932631 75.9295505   1.405898 45.522178 73.042595 27.0842247
## 2       0 19.588781 26.9613718  14.603632 11.911279 14.146125  3.2676445
## 3       0 21.165548 29.6778667   6.602531 24.919483 34.818398 21.1549971
## 4       0 54.418307 67.0124047 -11.959343 52.062344 67.670572 43.3530125
## 5       0  1.544175 -0.1637405 -15.672507 -3.577714 -3.583376  0.7710536
## 6       0 52.008179 83.5608516  17.355373 13.231720 28.521806 25.6009992
##    ES27_TMP   ES64_TMP   AUTO_EPI   ES27_EPI  ES64_EPI   AUTO_AVG  ES27_AVG
## 1 61.245982 79.9224515 25.1557215 57.4525957 76.517235   7.825799 65.130137
## 2 19.520924 27.3297205  2.9699912 23.1724591 28.010237  12.962537 15.901254
## 3 22.431916 35.5667615  2.0243189 21.2729578 28.340668   1.272685 17.498042
## 4 61.618578 65.9051749 -1.4165255 54.6871285 66.637175   1.731466 63.133507
## 5  1.599907 -0.5463776  0.6732459  0.1290369 -1.493592 -19.131470 -5.877317
## 6 60.851386 89.6659947 14.2607480 51.5880168 83.771395  13.559564 12.015030
##    ES64_AVG AUTO_AR_LB ES27_AR_LB ES64_AR_LB AUTO_ADJ_LB ES27_ADJ_LB
## 1 81.115255  10.286430  -2.486543  -1.904849  -15.009720   -10.65327
## 2 15.935157 -16.228749 -17.771485 -25.959829  -23.398383   -24.55139
## 3 27.824535 -10.711906  14.640488   3.065182  -15.173621   -13.34797
## 4 82.226544  -6.598803  -9.322600  -2.067466  -16.662644   -19.65063
## 5 -4.736474 -20.799265  -8.934283 -21.602930   -7.523827   -13.76550
## 6 20.766749   7.296873   7.275076  -2.058357  -12.424557   -10.29405
##   ES64_ADJ_LB AUTO_TMP_LB ES27_TMP_LB ES64_TMP_LB AUTO_EPI_LB ES27_EPI_LB
## 1  -10.197239   15.920504  -0.2647823  -0.1195217    6.302988   -1.245239
## 2  -26.700535  -17.718261 -22.9614563 -25.6219377  -17.456918  -22.832527
## 3  -20.722050   -7.687910   7.8108992   2.2057452  -13.218275    8.885548
## 4  -14.304472  -22.339698 -17.6889354 -15.4575331    4.091483   -9.093968
## 5  -11.422367  -13.479552  -8.8881075 -19.8962239   -7.633103  -14.124793
## 6   -9.888878    1.956431   0.7591413  -6.5091213   -4.258822    4.012234
##   ES64_EPI_LB AUTO_AVG_LB ES27_AVG_LB ES64_AVG_LB
## 1  -1.9397912  -6.5972215   3.9200855    4.275468
## 2 -23.7893168   8.6712811  -0.7817954   -6.651501
## 3   0.3302966 -13.4438177  -6.8958027  -15.834013
## 4   0.2462124   0.9964555  -0.9979213   -6.409795
## 5 -19.3740058 -18.2823634 -24.5594009  -22.542724
## 6  -3.8543766 -16.2110448 -18.4088490  -18.255348
```

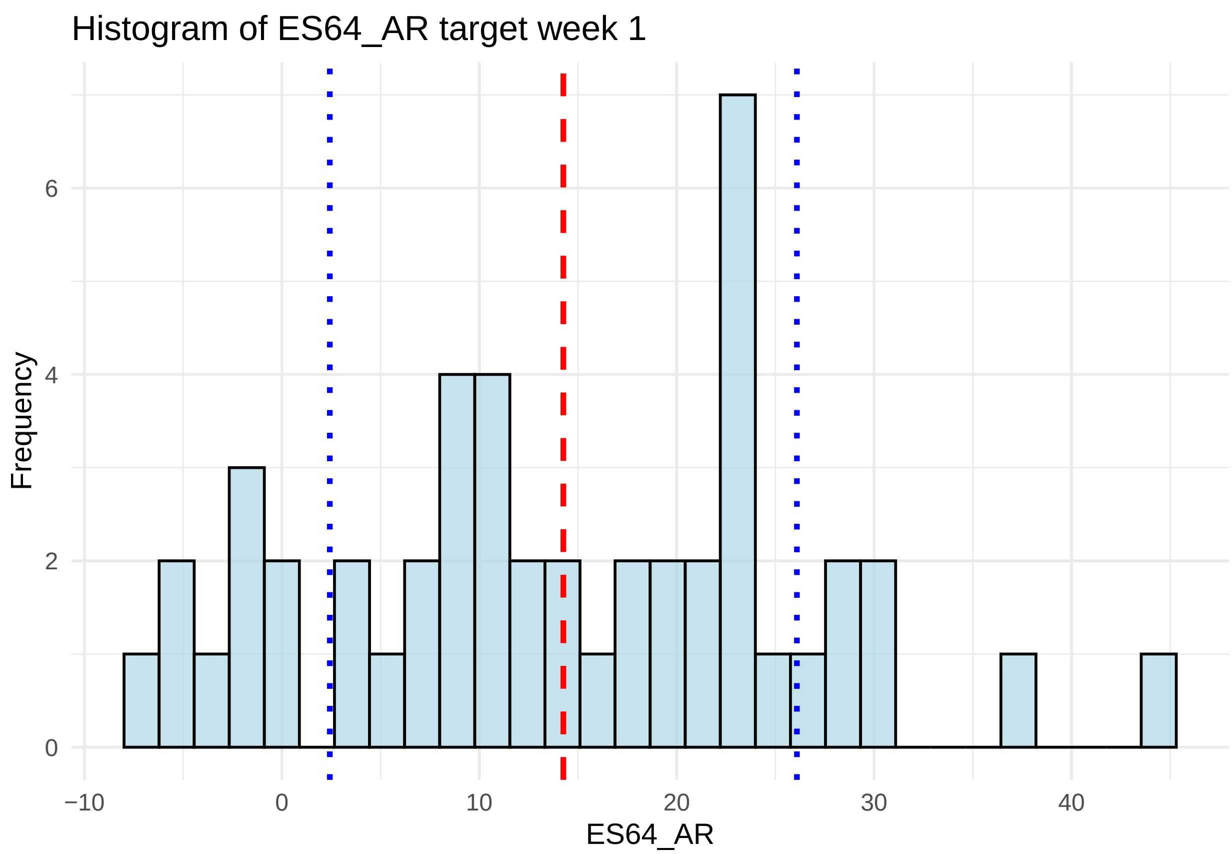
Now let's plot histograms of percentage of WIS improvement for each state including mean and standard deviation of percentage of improvement compared to auto_arima (baseline).

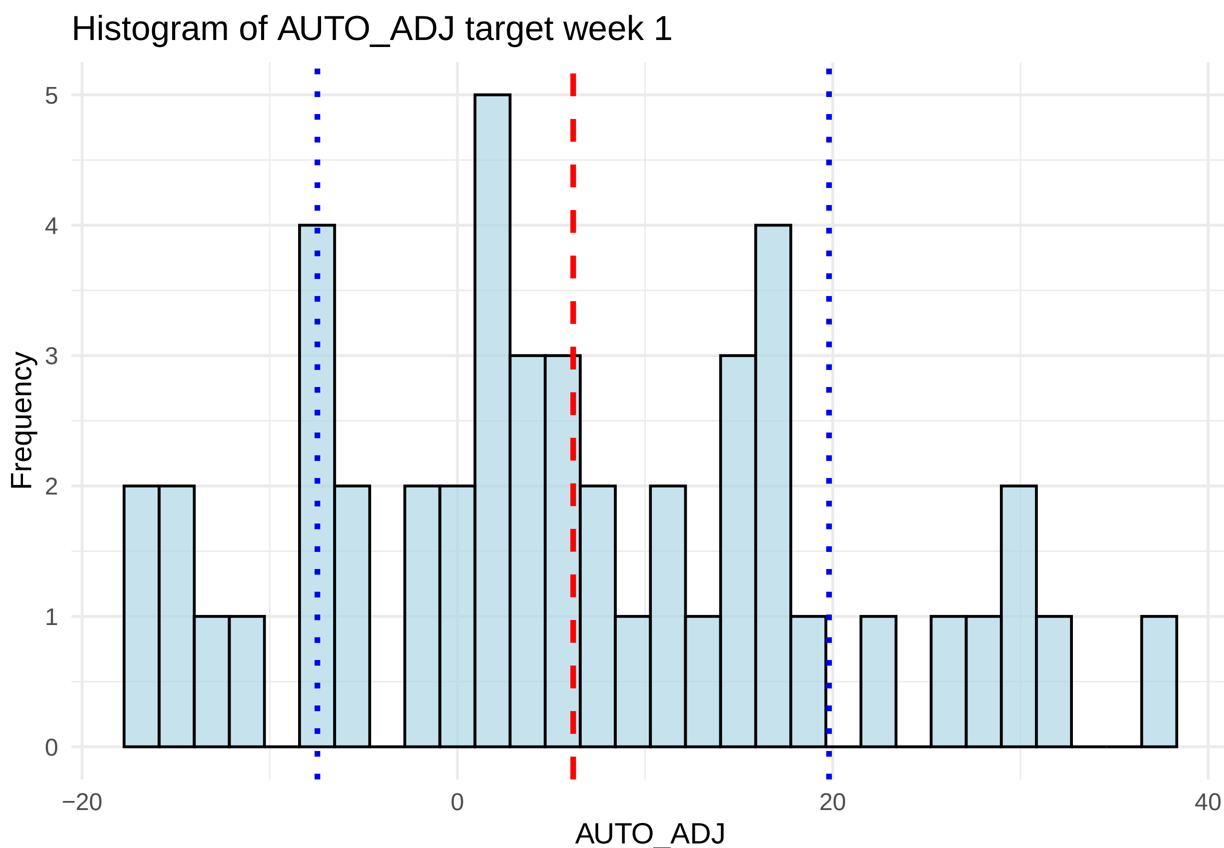
```r
# List of models names
models_names <-c(names(W1_percentage_of_improvement))

# Create an empty dataframe to store results
summary_impr <- data.frame(WeekAhead = character(), Model = character(), m = numeric(), sd = numeric(),

# List of datasets. One dataset for each target week.
datasets_list <- list("1" = W1_percentage_of_improvement,
                      "2" = W2_percentage_of_improvement,
```

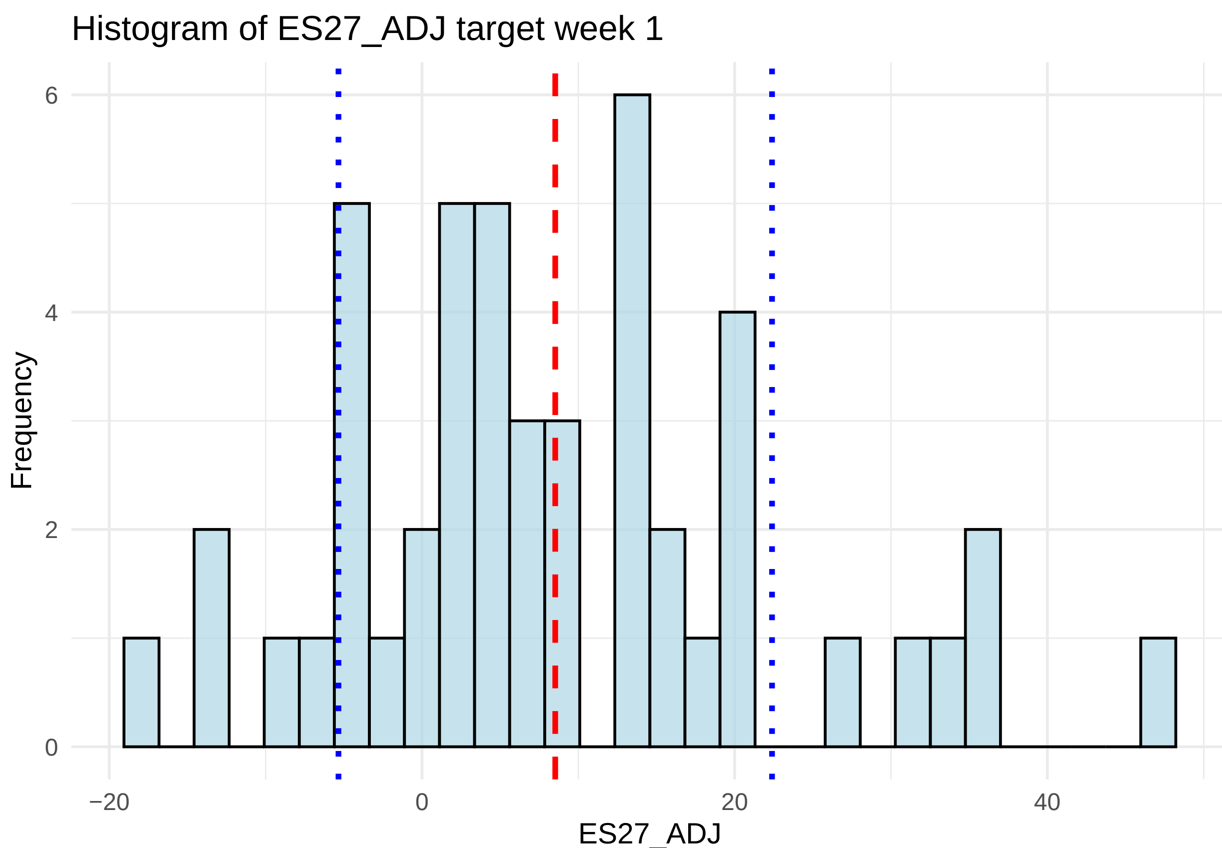
```r
                 "3" = W3_percentage_of_improvement,
                 "4" = W4_percentage_of_improvement)

# Here I have 2 for loops. One take into account the target week and the other the model.
# Loop through target weeks
for (target_week_ in names(datasets_list)) {
  dataset <- datasets_list[[target_week_]]  # Get the dataset based on target week
  # Loop through models
  for (given_model in models_names) {
    data <- dataset[[given_model]]
    mean_val <- mean(data, na.rm = TRUE)
    sd_val <- sd(data, na.rm = TRUE)

    # Append results to dataframe
    summary_impr <- rbind(summary_impr, data.frame(WeekAhead = target_week_, Model = given_model, m = me

    # Generate histogram
    p <- ggplot(dataset, aes(x = .data[[given_model]])) +
      geom_histogram(color = "black", fill = "lightblue", bins = 30, alpha = 0.7) +
      geom_vline(aes(xintercept = mean_val), color = "red", linetype = "dashed", linewidth = 1) +
      geom_vline(aes(xintercept = mean_val - sd_val), color = "blue", linetype = "dotted", linewidth = 1
      geom_vline(aes(xintercept = mean_val + sd_val), color = "blue", linetype = "dotted", linewidth = 1
      labs(title = paste("Histogram of", given_model, "target week", target_week_),
           x = given_model,
           y = "Frequency") +
      theme_minimal()

    print(p)  # Display the plot
  }
}
```

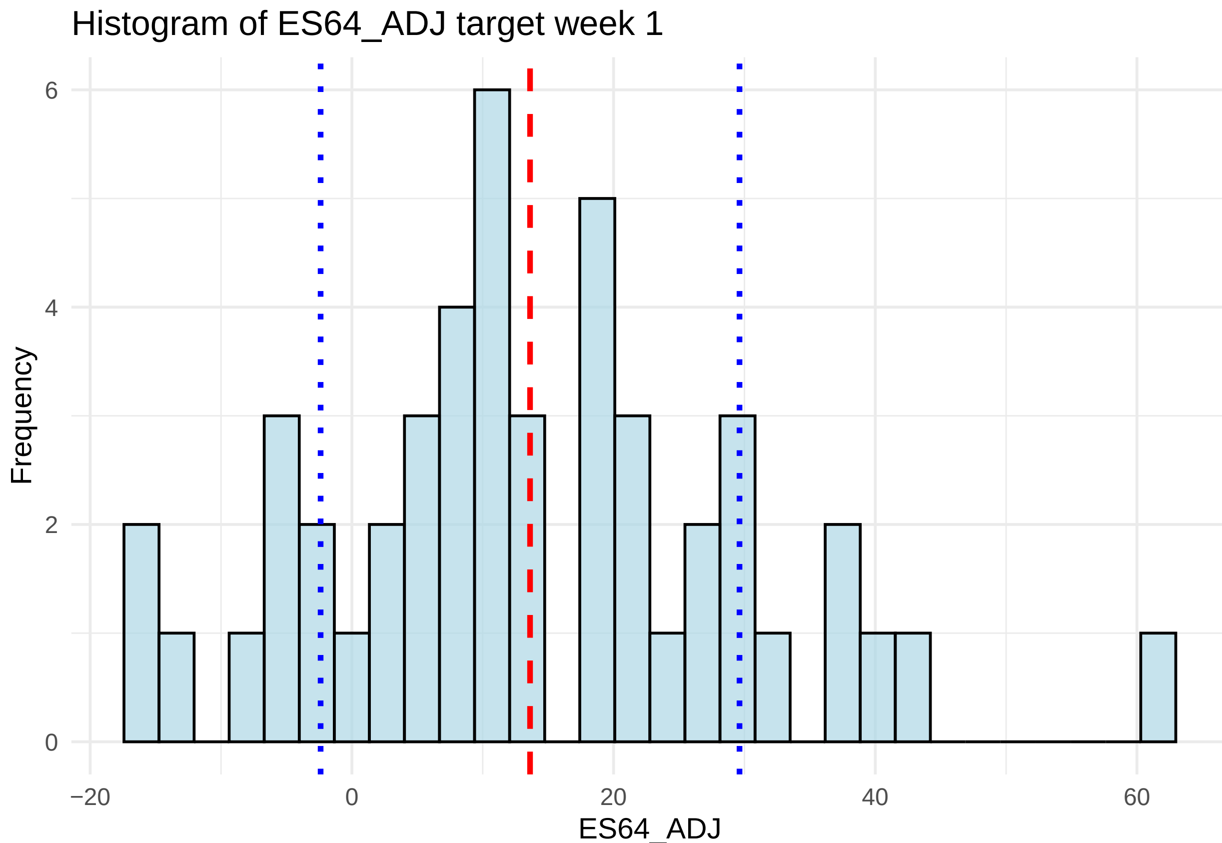

Histogram of AUTO_AR target week 1
Frequency
50
40
30
20
10
0
−0.050
−0.025
0.000
0.025
0.050
AUTO_AR

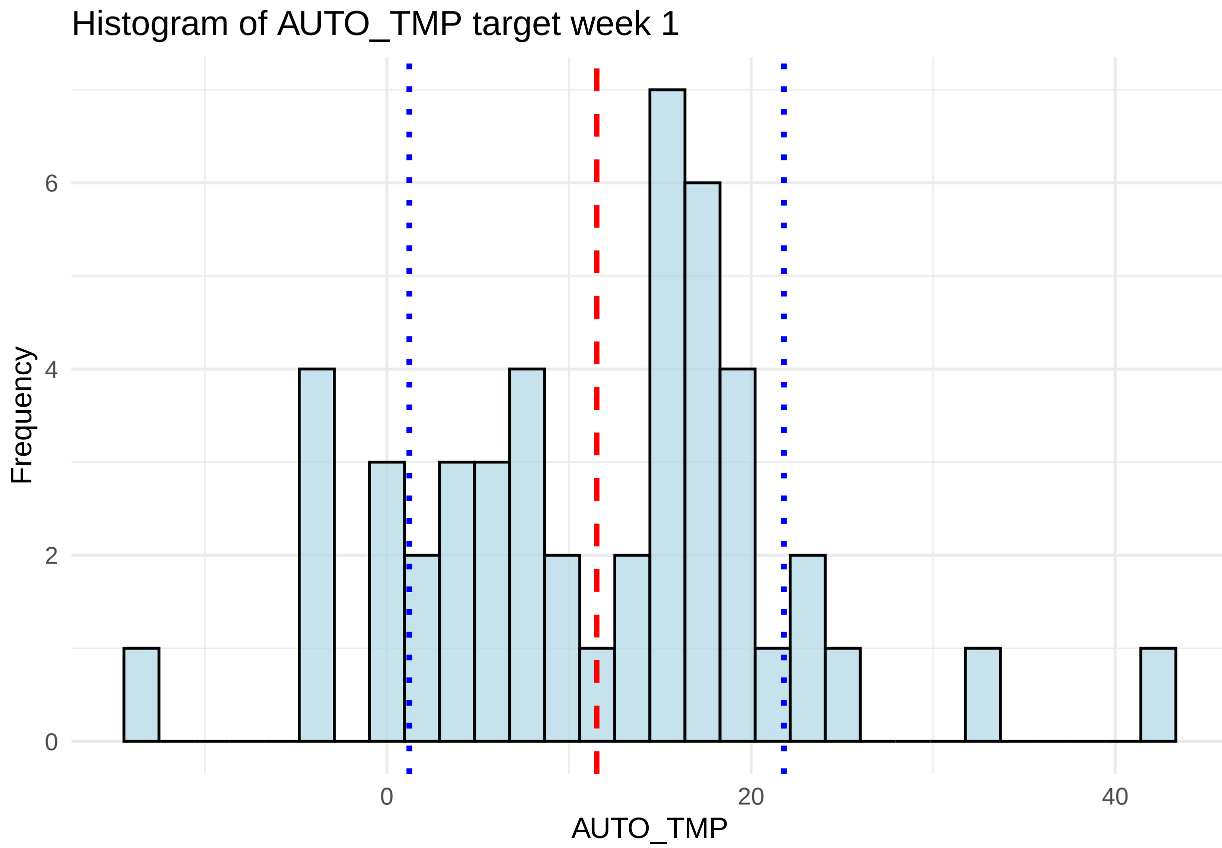

Histogram of ES27_AR target week 1
Frequency
0
1
2
3
4
−10
0
10
20
ES27_AR

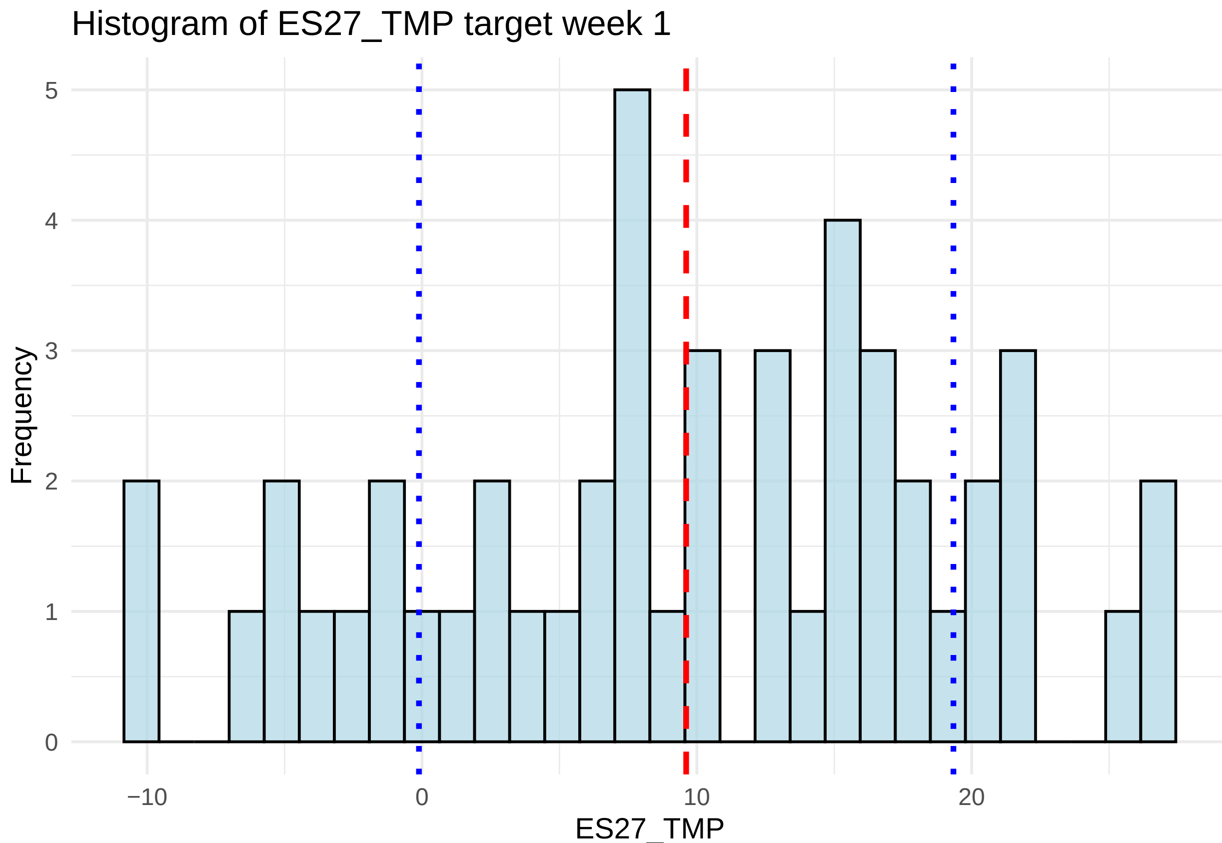

Histogram of ES64_AR target week 1
Frequency
0
2
4
6
−10
0
10
20
30
40
ES64_AR


Histogram of AUTO_ADJ target week 1
Frequency
0
1
2
3
4
5
−20
0
20
40
AUTO_ADJ

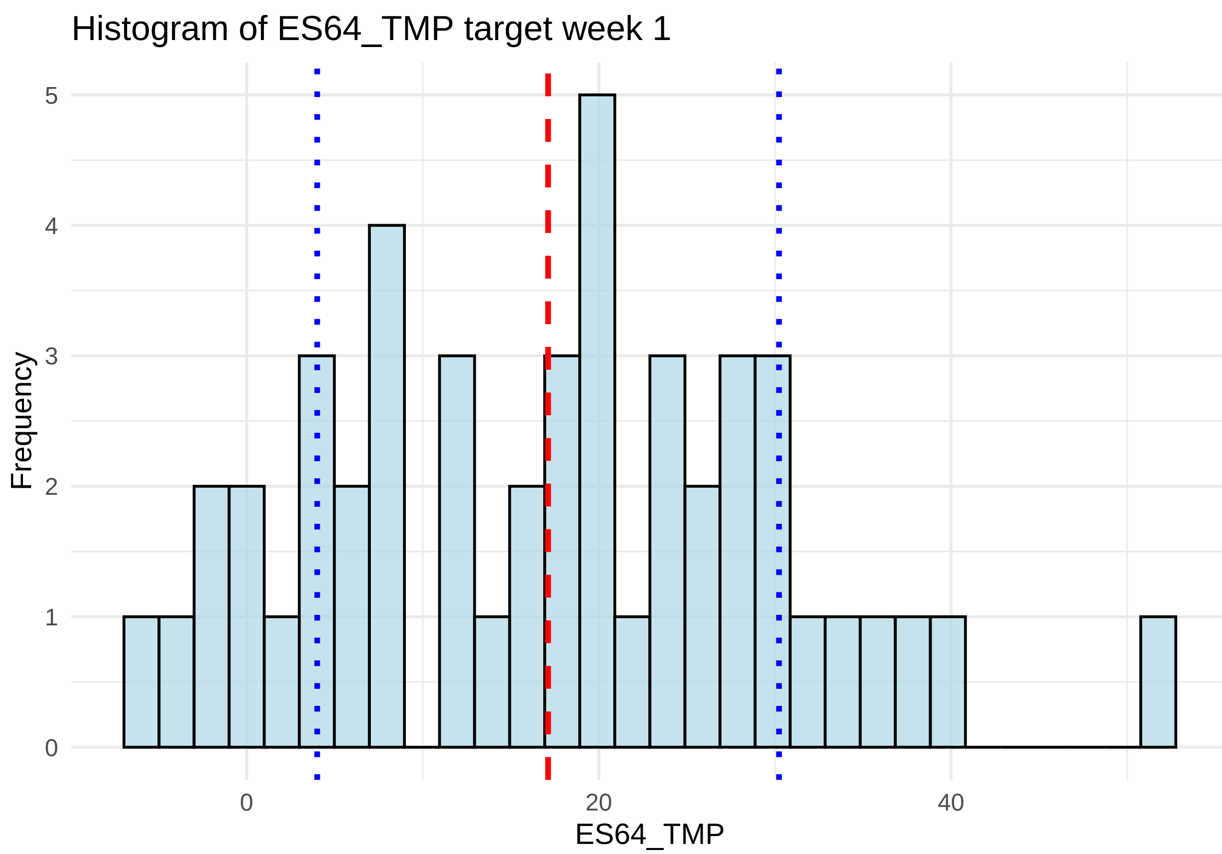

Histogram of ES27_ADJ target week 1
Frequency
ES27_ADJ
0
2
4
6
−20
0
20
40

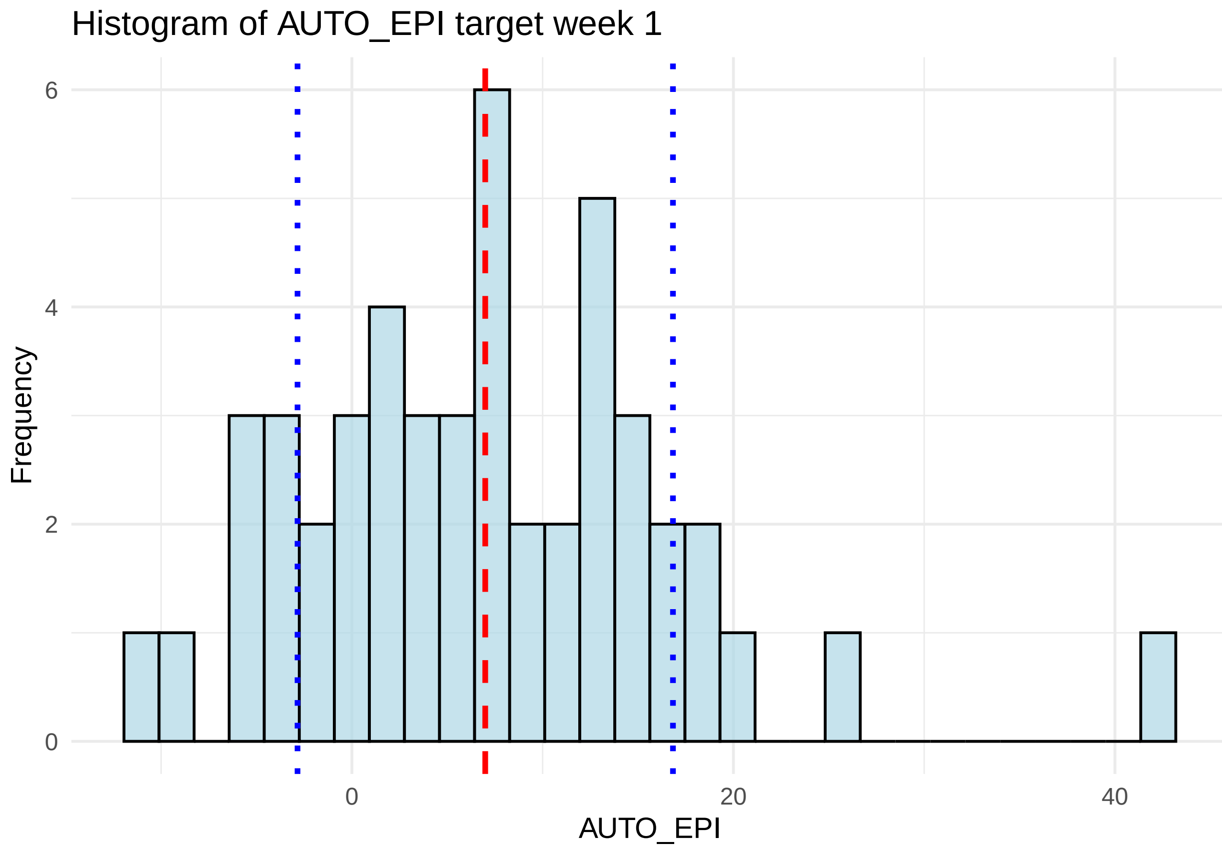

Histogram of ES64_ADJ target week 1
Frequency
0
2
4
6
−20
0
20
40
60
ES64_ADJ

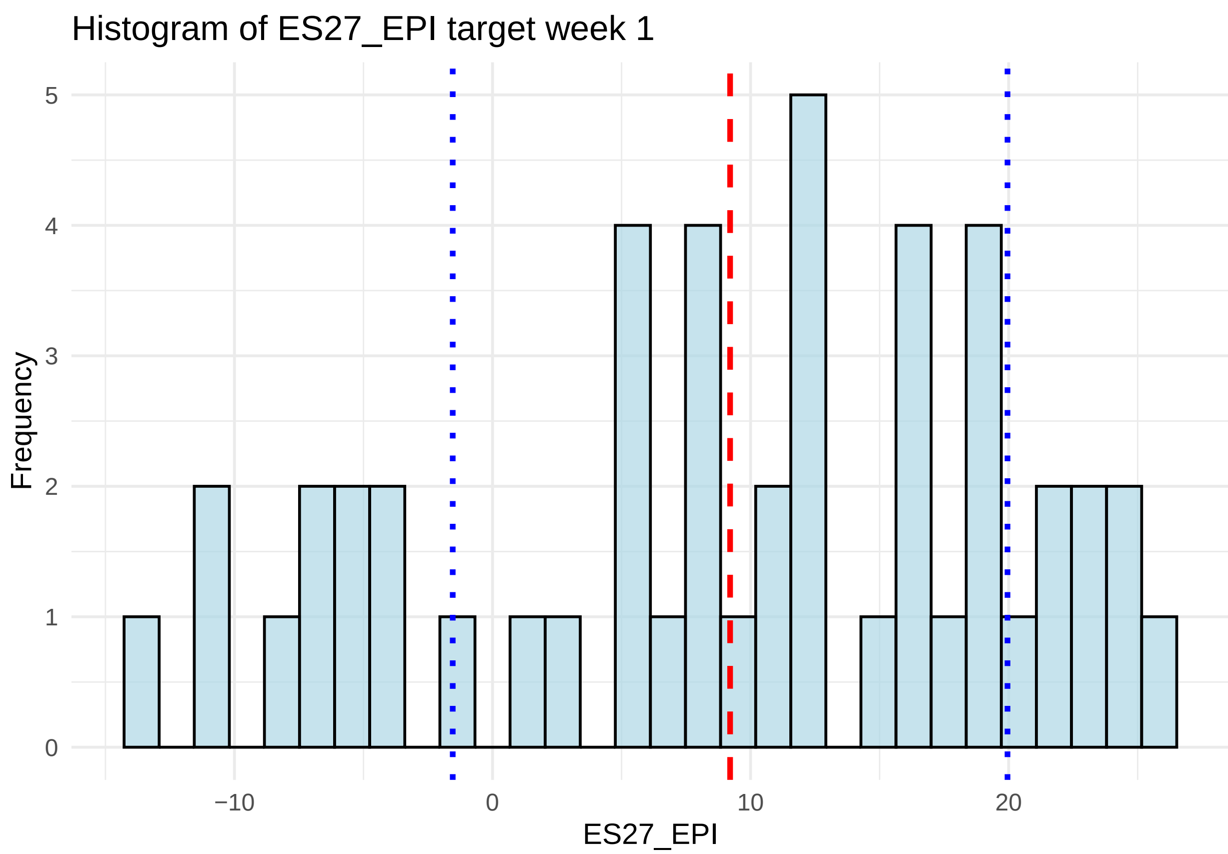

Histogram of AUTO_TMP target week 1
Frequency
6
4
2
0
0
20
40
AUTO_TMP

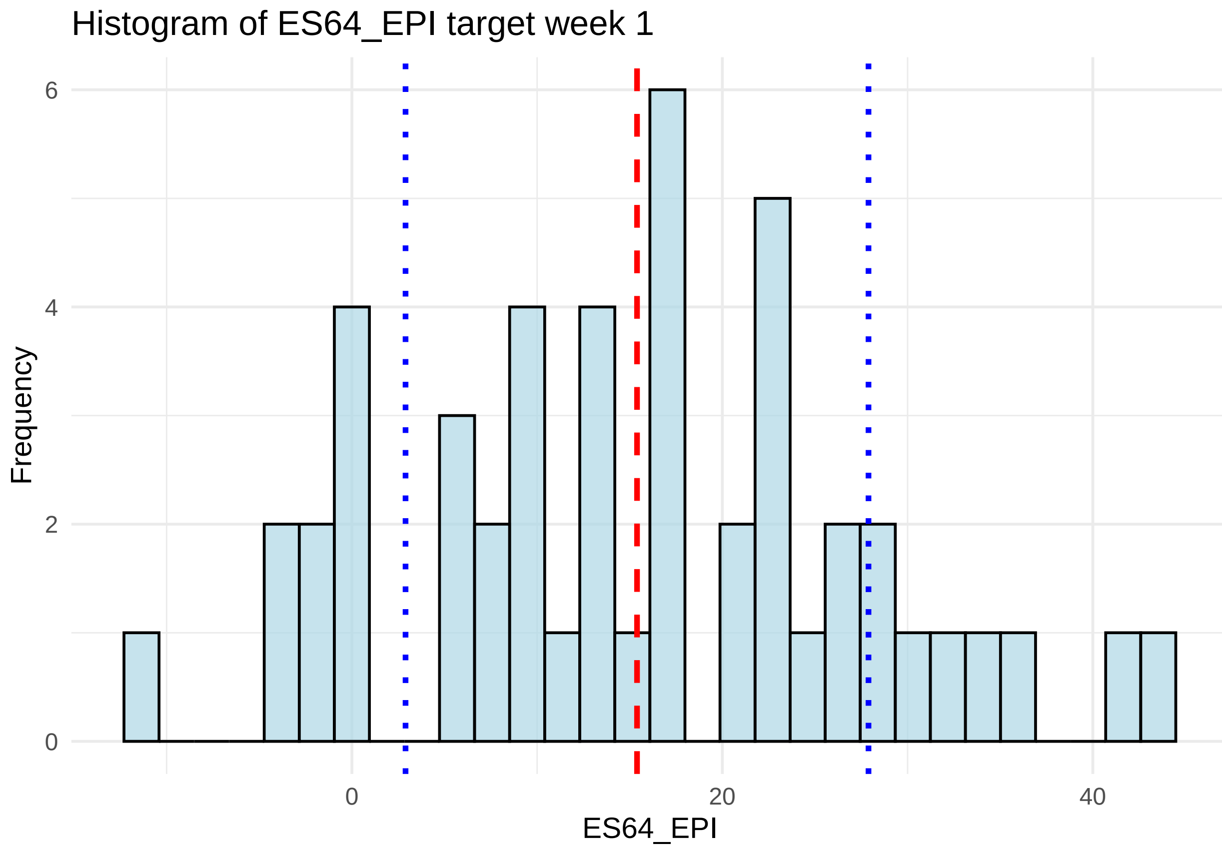


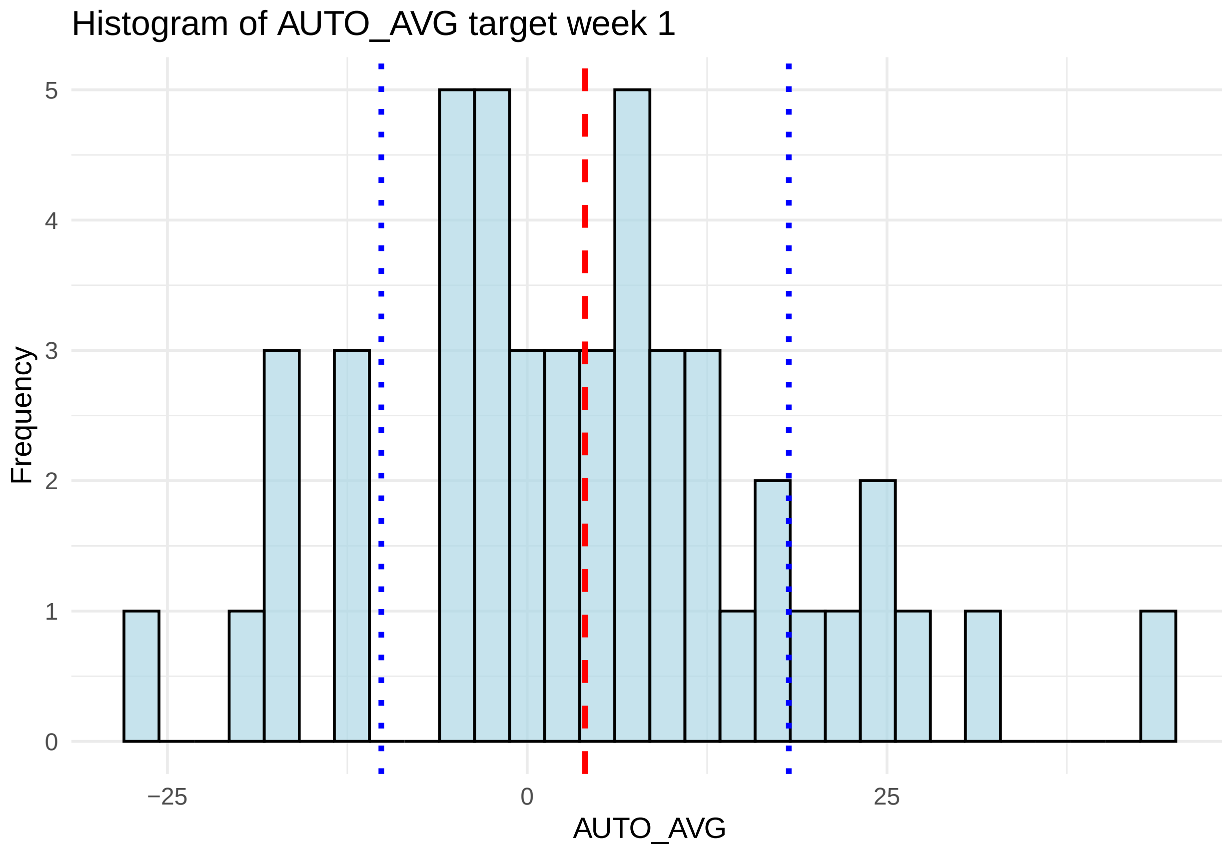
Histogram of ES27_TMP target week 1
Frequency
5
4
3
2
1
0
−10
0
10
20
ES27_TMP

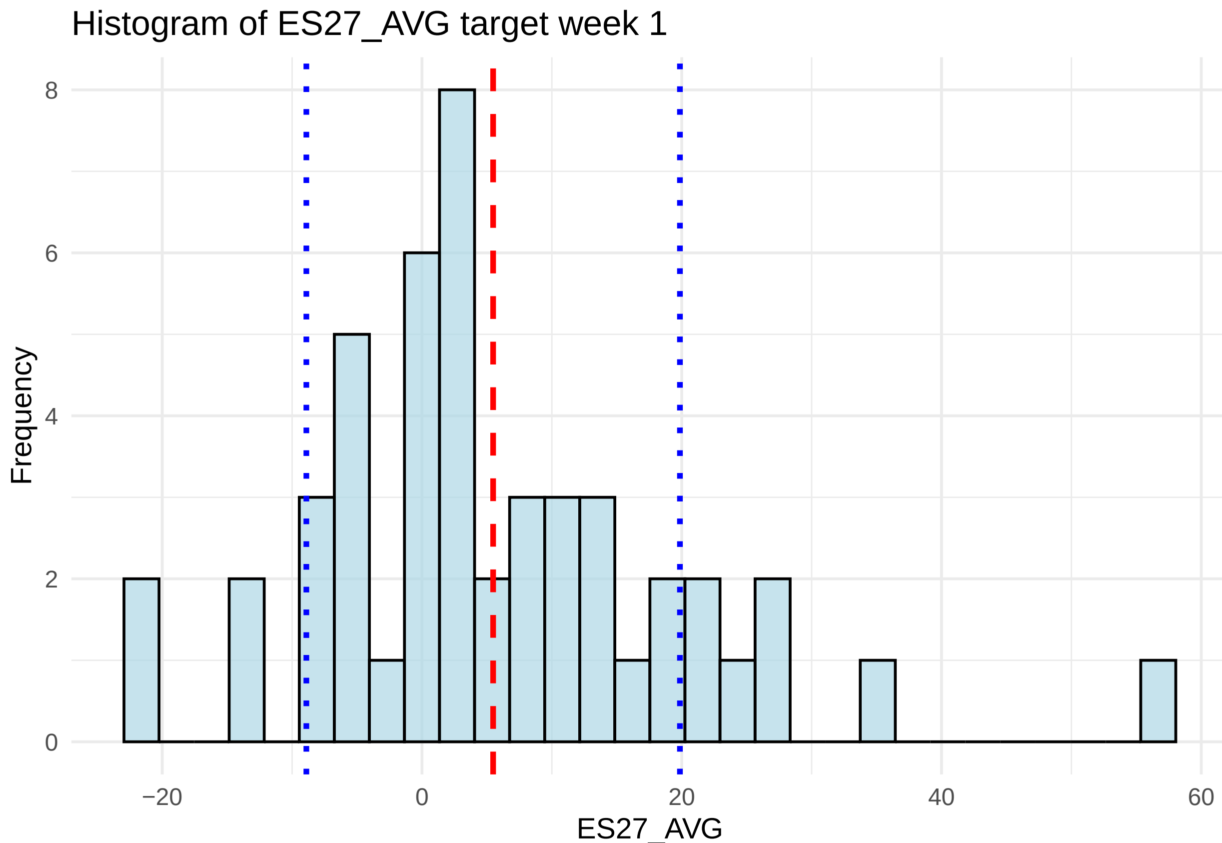

Histogram of ES64_TMP target week 1
Frequency
0
1
2
3
4
5
0
20
40
ES64_TMP

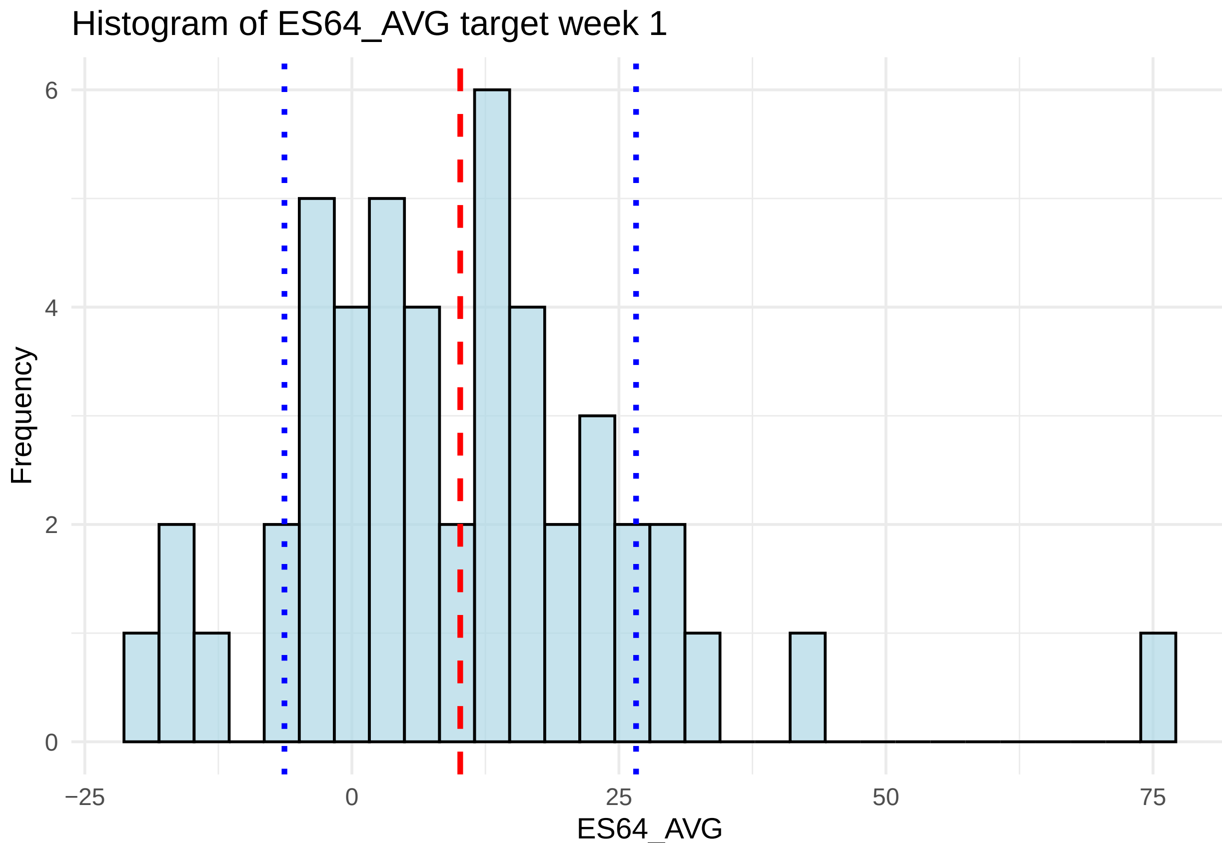

Histogram of AUTO_EPI target week 1
Frequency
0
2
4
6
0
20
40
AUTO_EPI

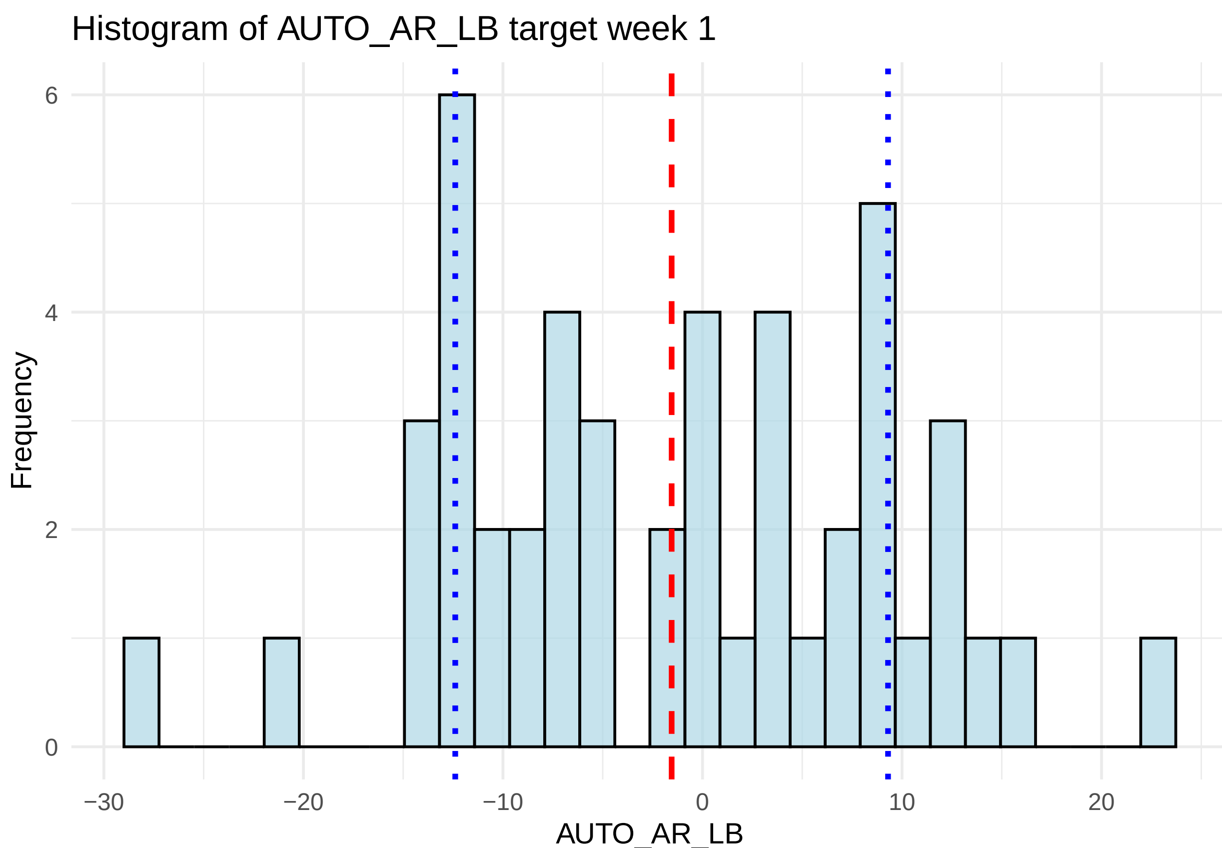

Histogram of ES27_EPI target week 1
Frequency
5
4
3
2
1
0
−10
0
10
20
ES27_EPI

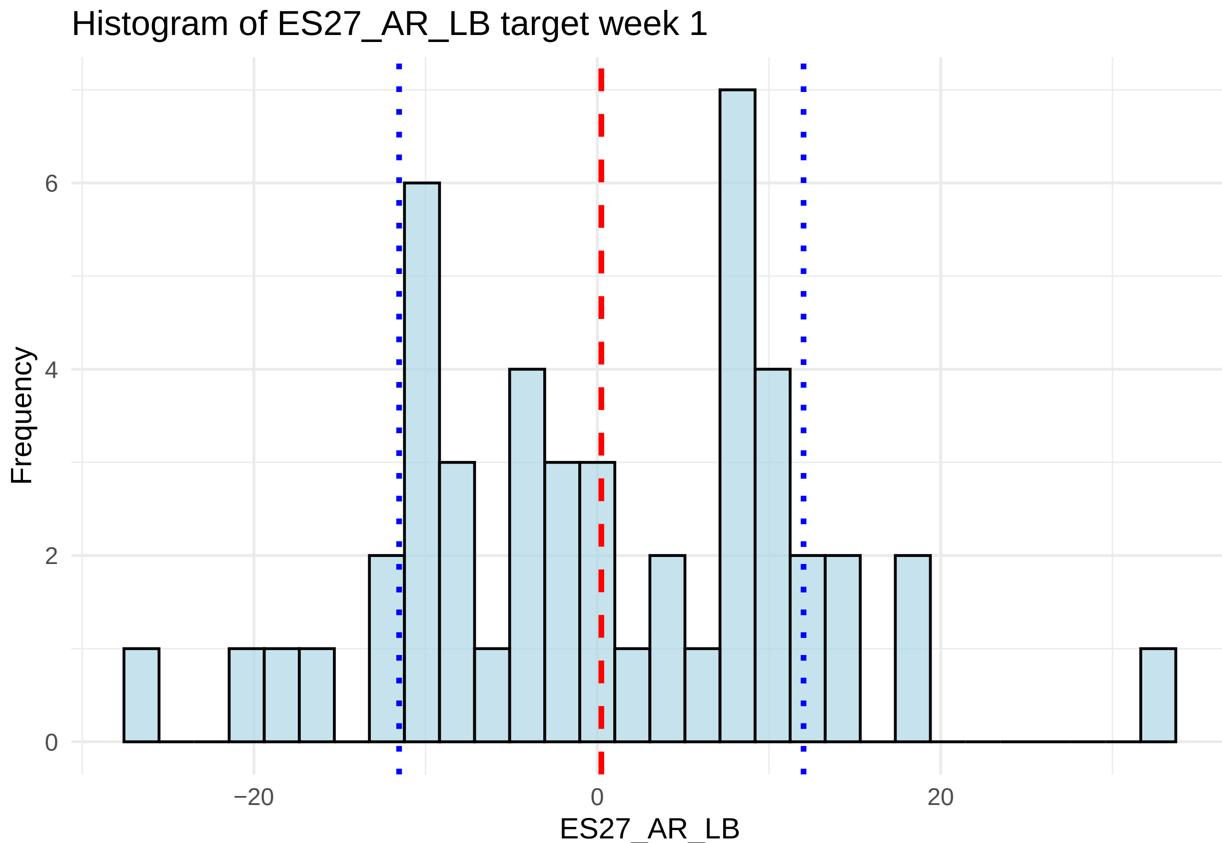

Histogram of ES64_EPI target week 1
Frequency
ES64_EPI
0
2
4
6
0
20
40

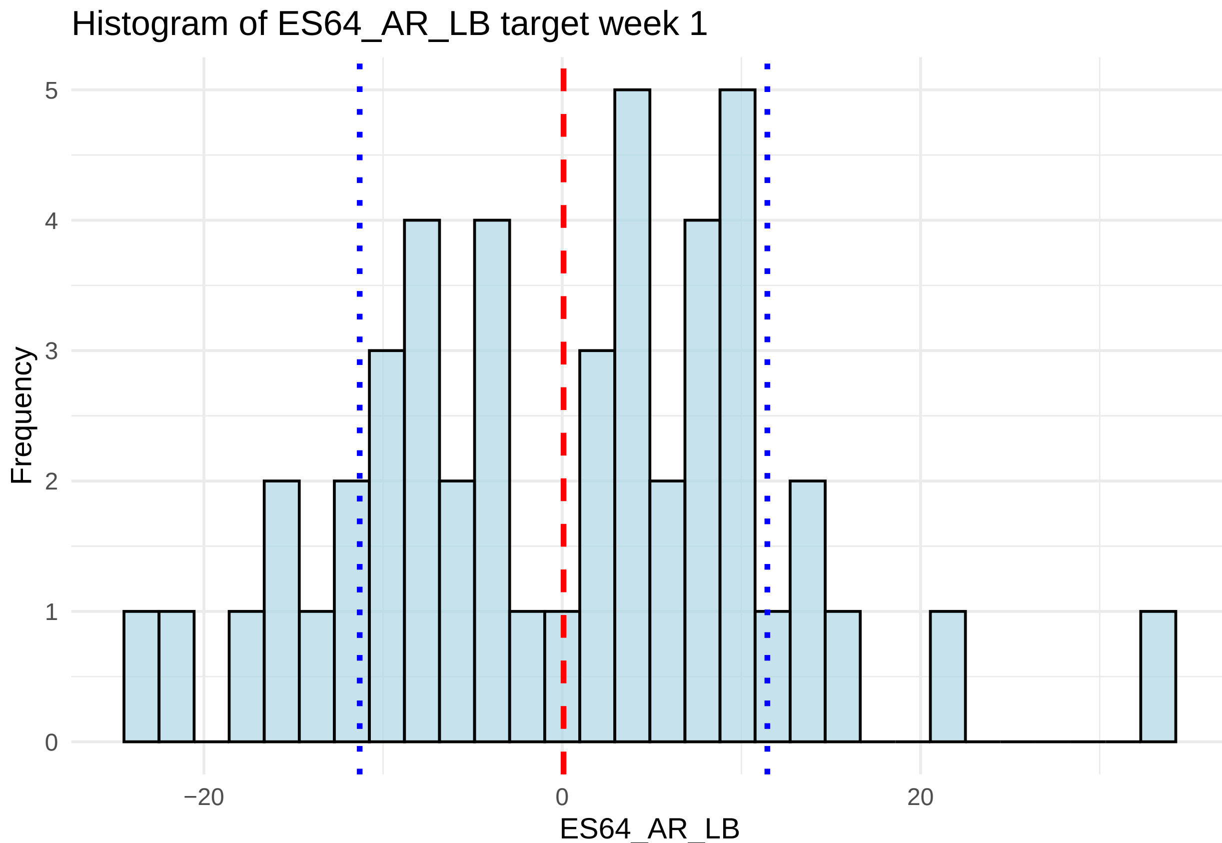

Histogram of AUTO_AVG target week 1
Frequency
5
4
3
2
1
0
−25
0
25
AUTO_AVG

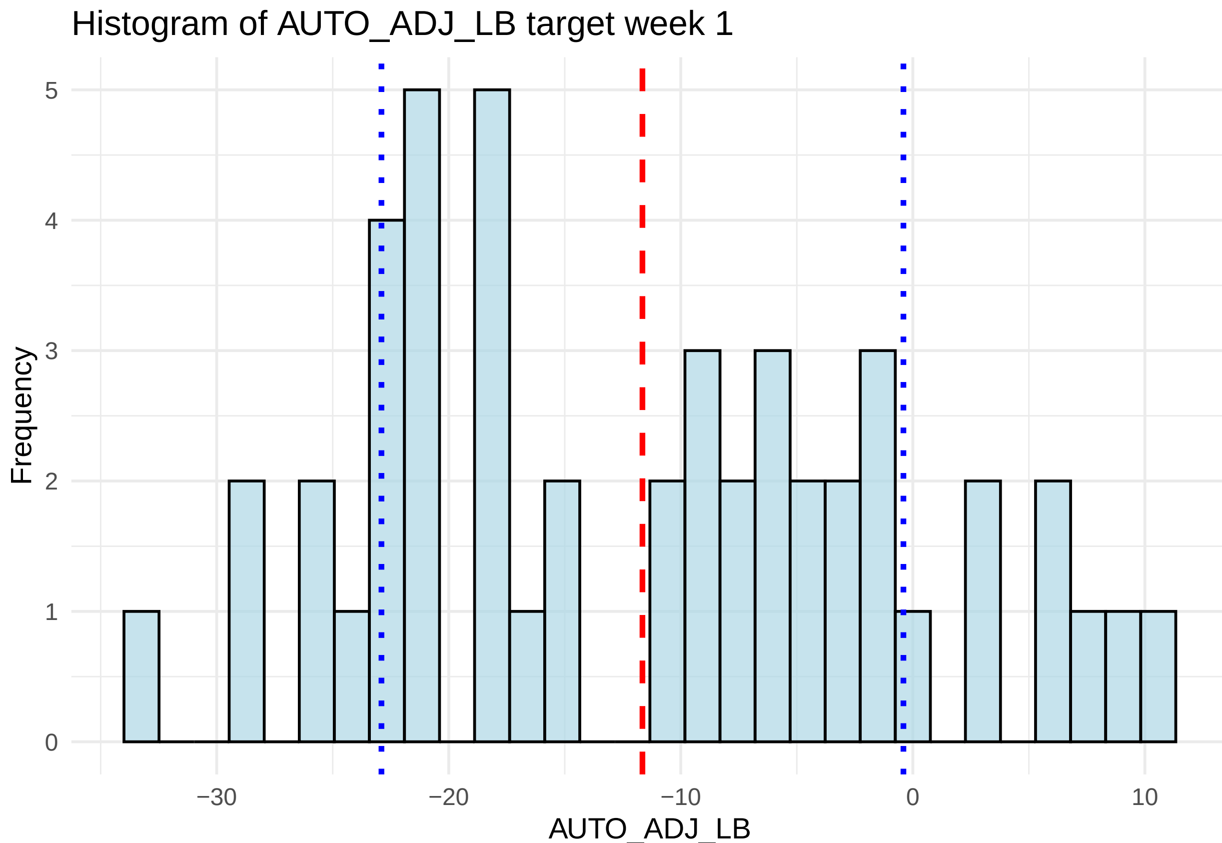

Histogram of ES27_AVG target week 1
Frequency
ES27_AVG
0
2
4
6
8
−20
0
20
40
60


Histogram of ES64_AVG target week 1
Frequency
ES64_AVG
6
4
2
0
−25
0
25
50
75

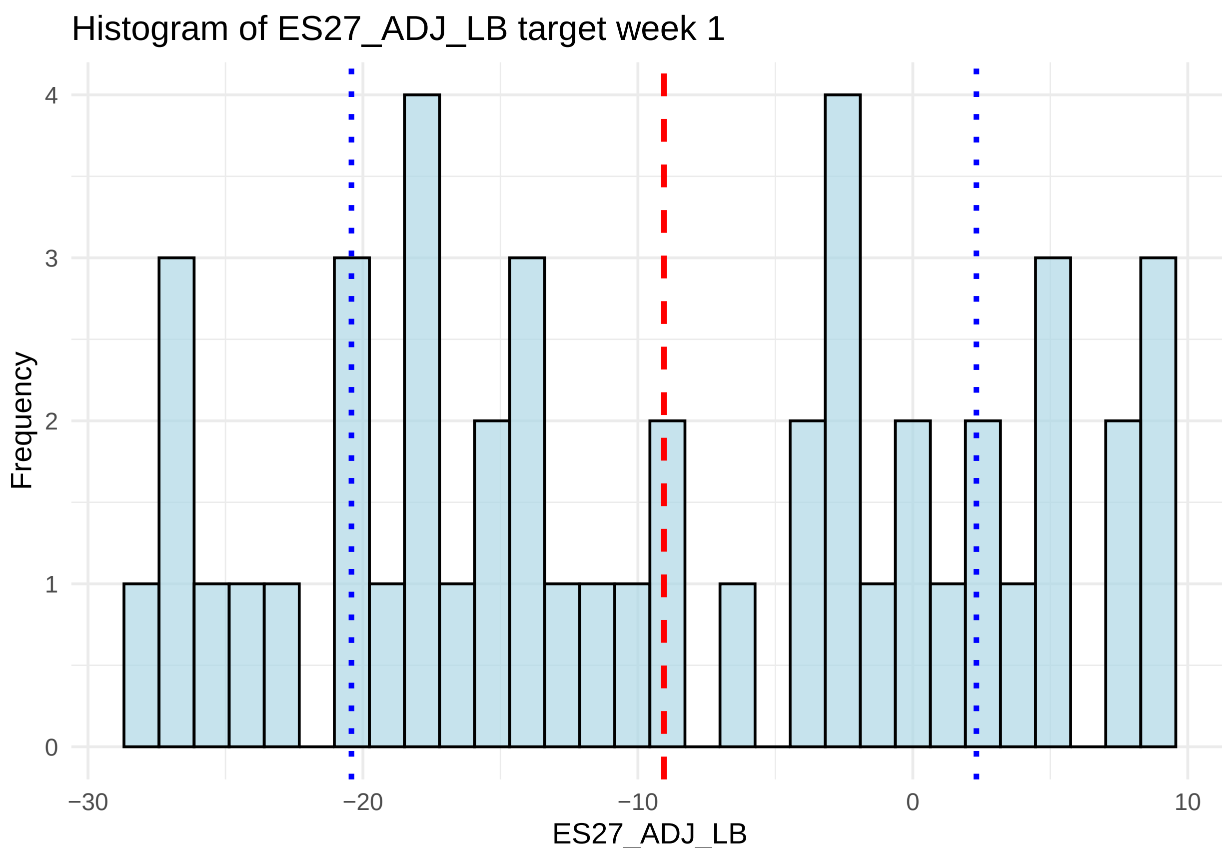

Histogram of AUTO_AR_LB target week 1
Frequency
0
2
4
6
−30
−20
−10
0
10
20
AUTO_AR_LB

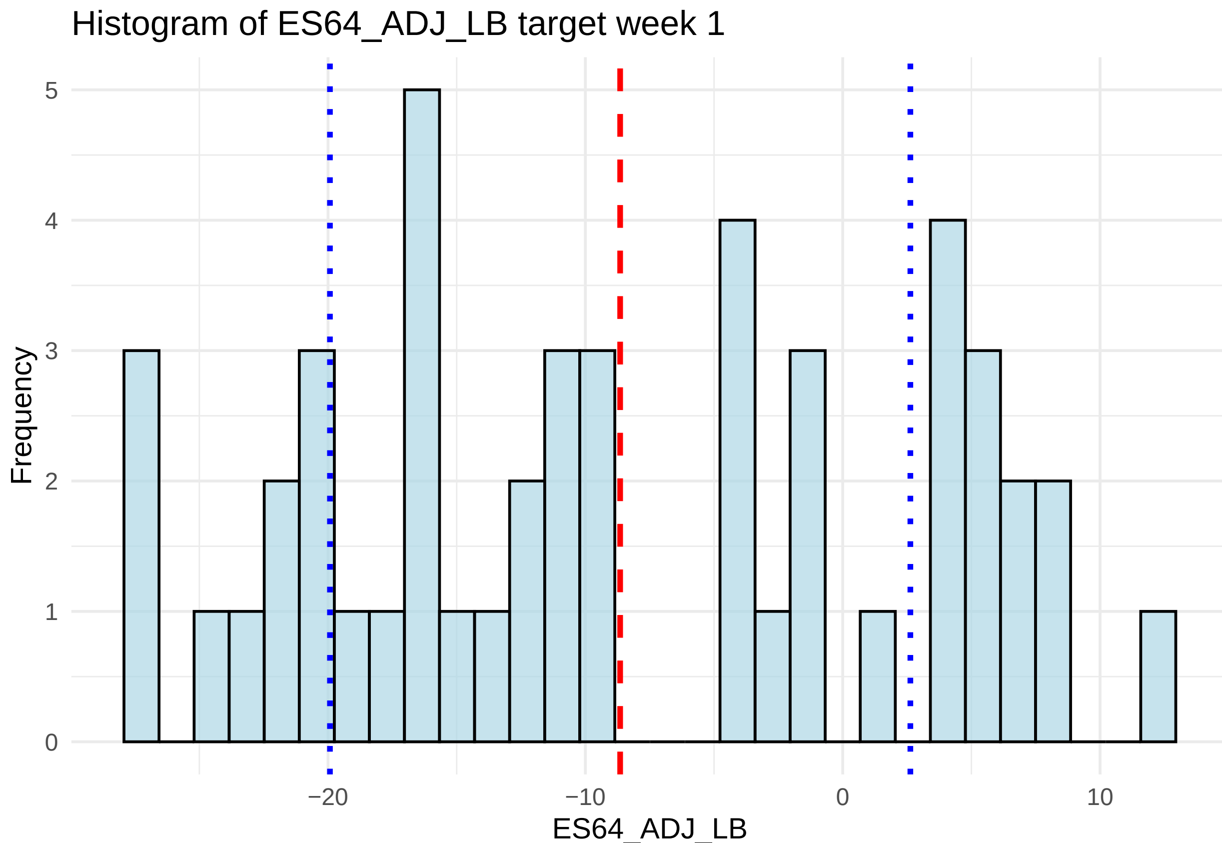

Histogram of ES27_AR_LB target week 1
Frequency
0
2
4
6
−20
0
20
ES27_AR_LB

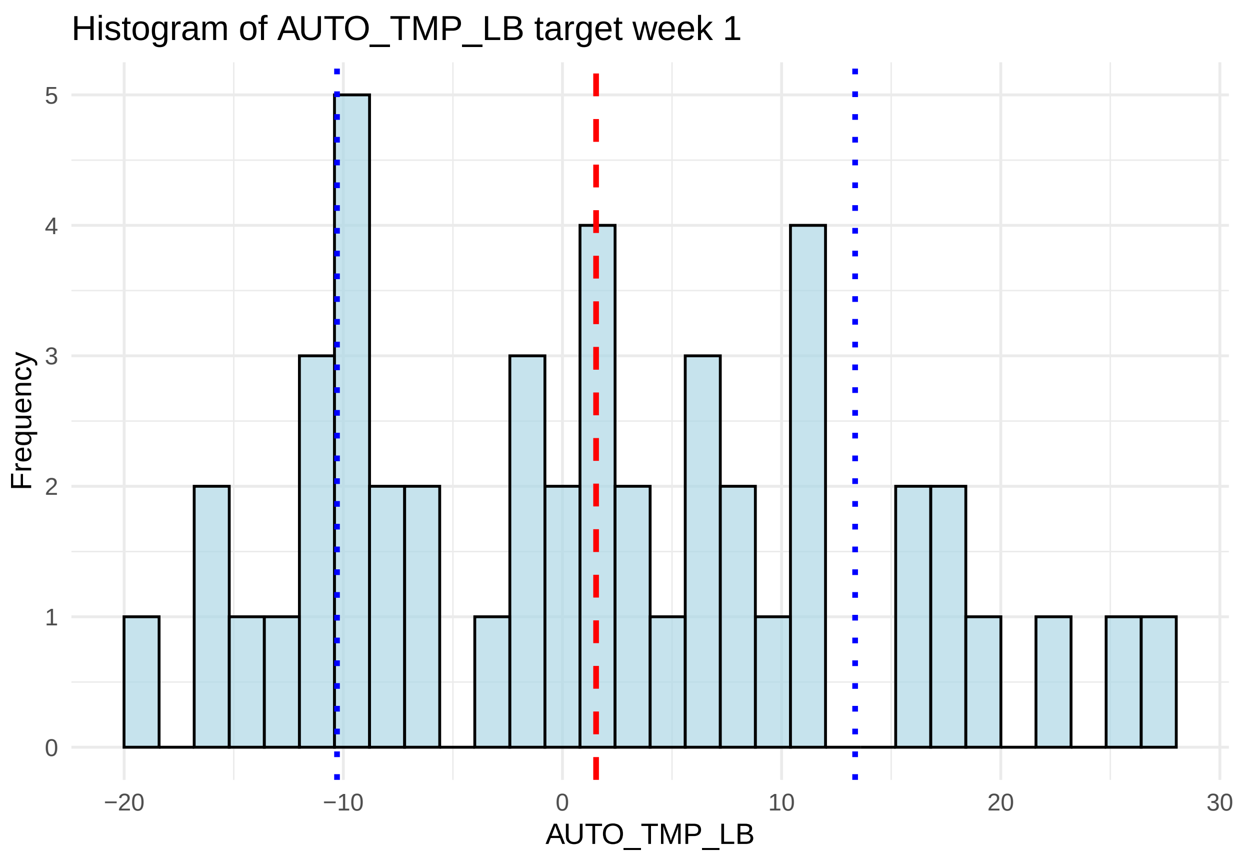

Histogram of ES64_AR_LB target week 1
Frequency
5
4
3
2
1
0
−20
0
20
ES64_AR_LB

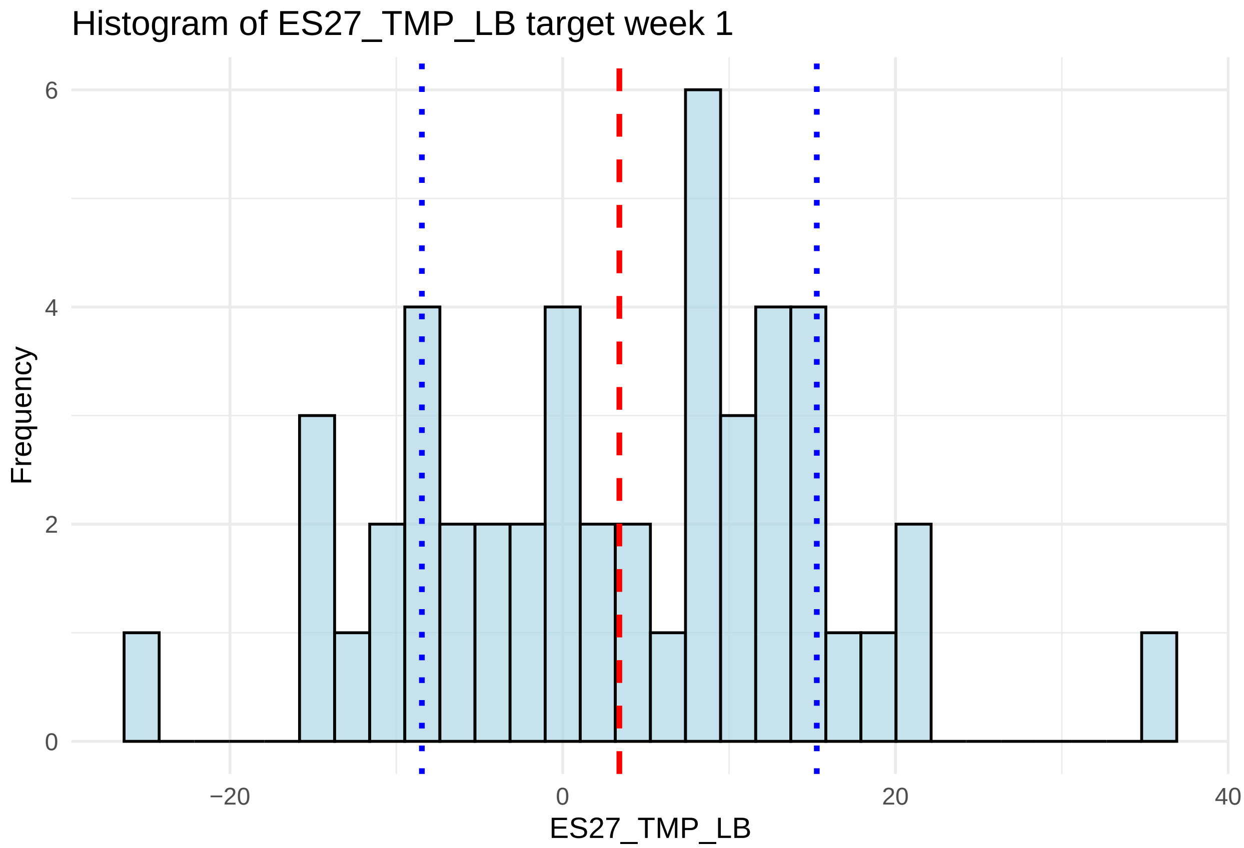


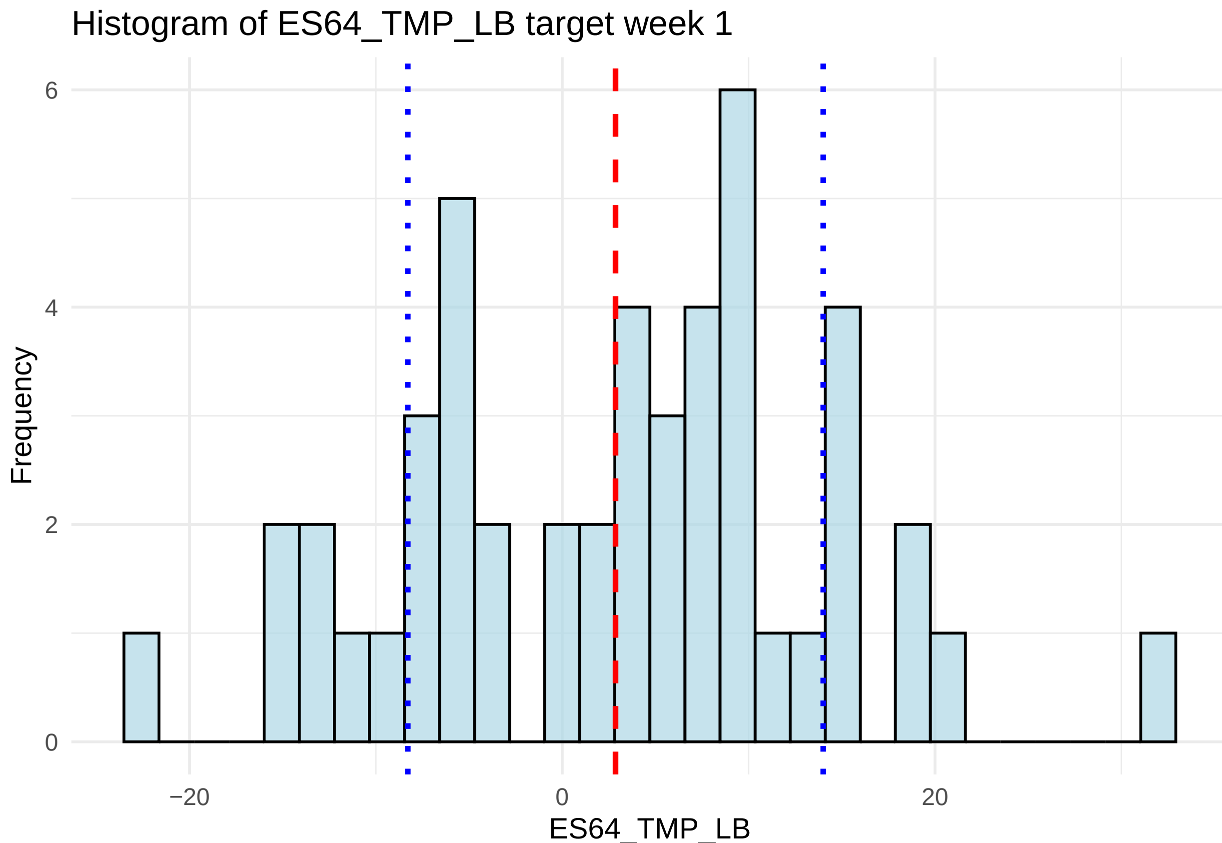
Histogram of AUTO_ADJ_LB target week 1
Frequency
0
1
2
3
4
5
−30
−20
−10
0
10
AUTO_ADJ_LB


Histogram of ES27_ADJ_LB target week 1
Frequency
0
1
2
3
4
−30
−20
−10
0
10
ES27_ADJ_LB

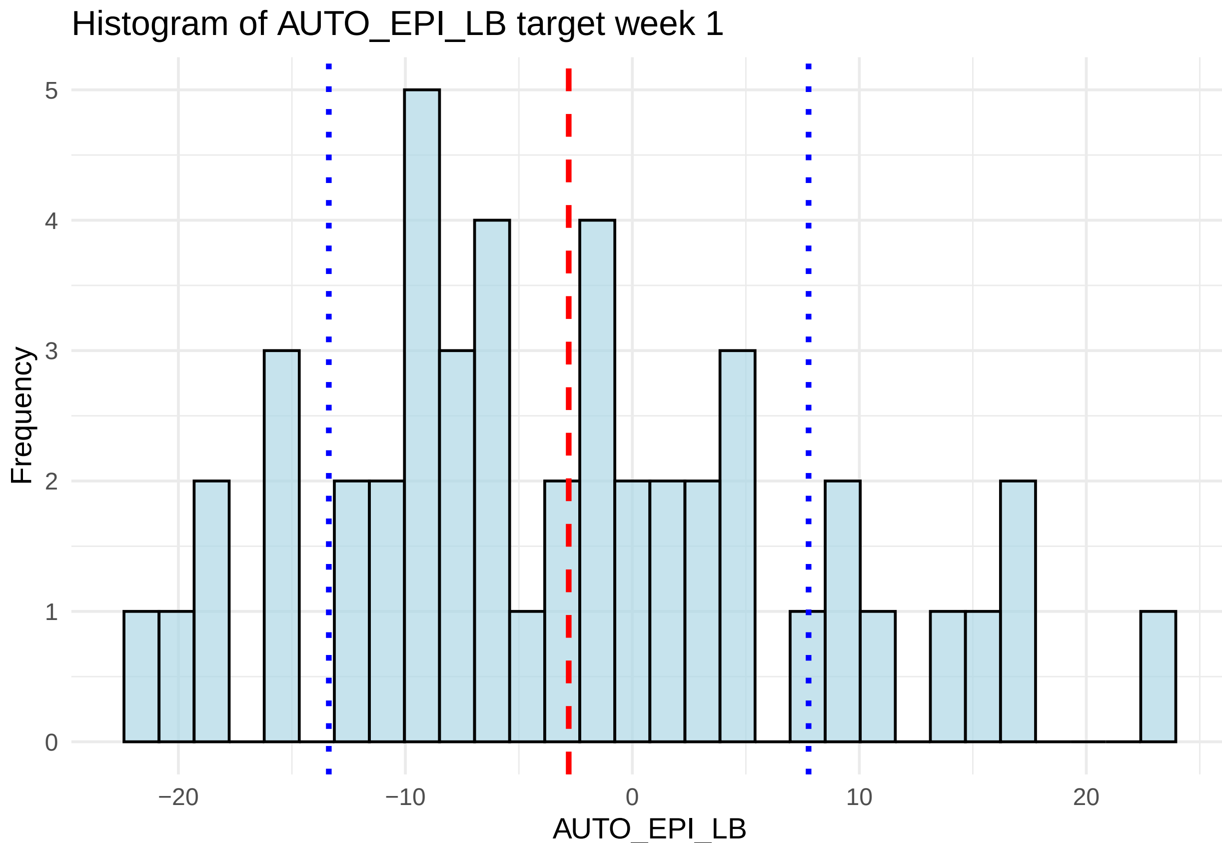

Histogram of ES64_ADJ_LB target week 1
Frequency
5
4
3
2
1
0
−20
−10
0
10
ES64_ADJ_LB


Histogram of AUTO_TMP_LB target week 1
Frequency
5
4
3
2
1
0
−20
−10
0
10
20
30
AUTO_TMP_LB

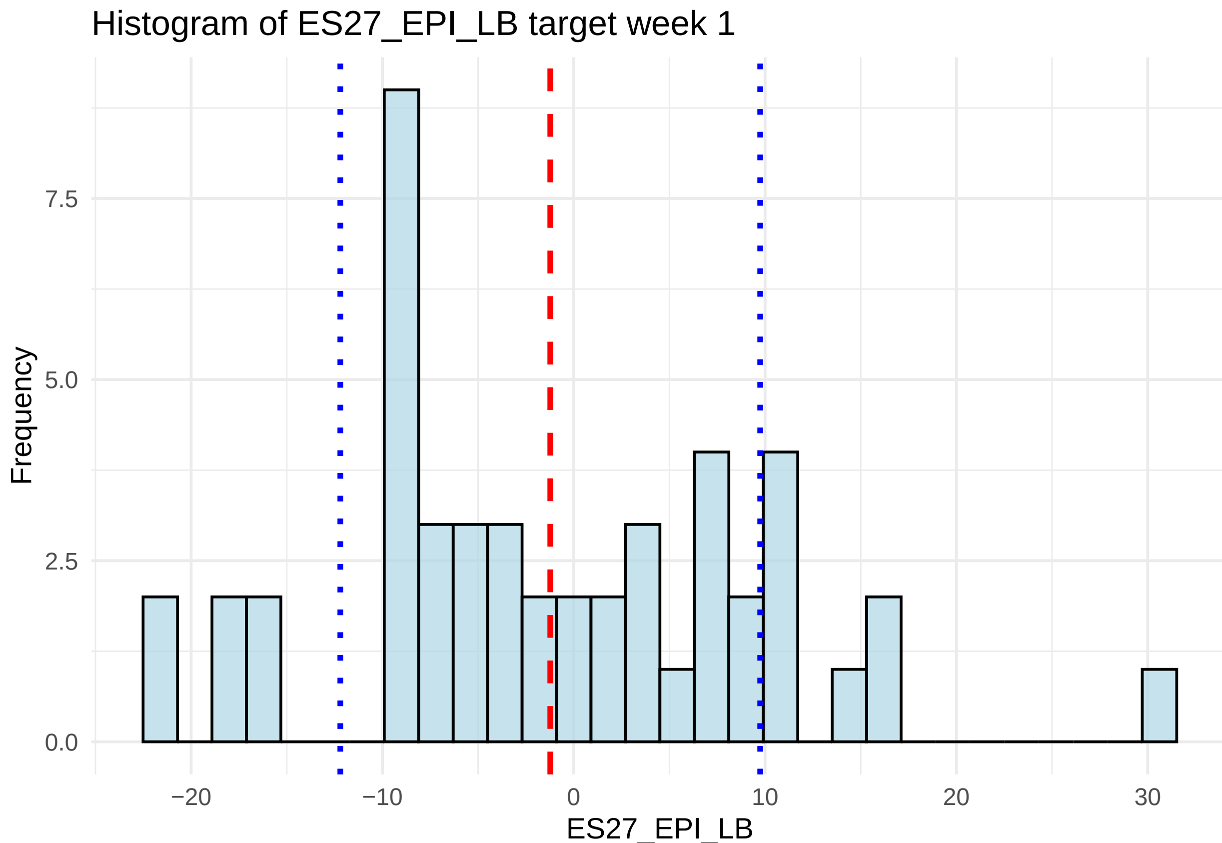

Histogram of ES27_TMP_LB target week 1
Frequency
6
4
2
0
−20
0
20
40
ES27_TMP_LB



Histogram of ES64_TMP_LB target week 1
Frequency
6
4
2
0
−20
0
20
ES64_TMP_LB

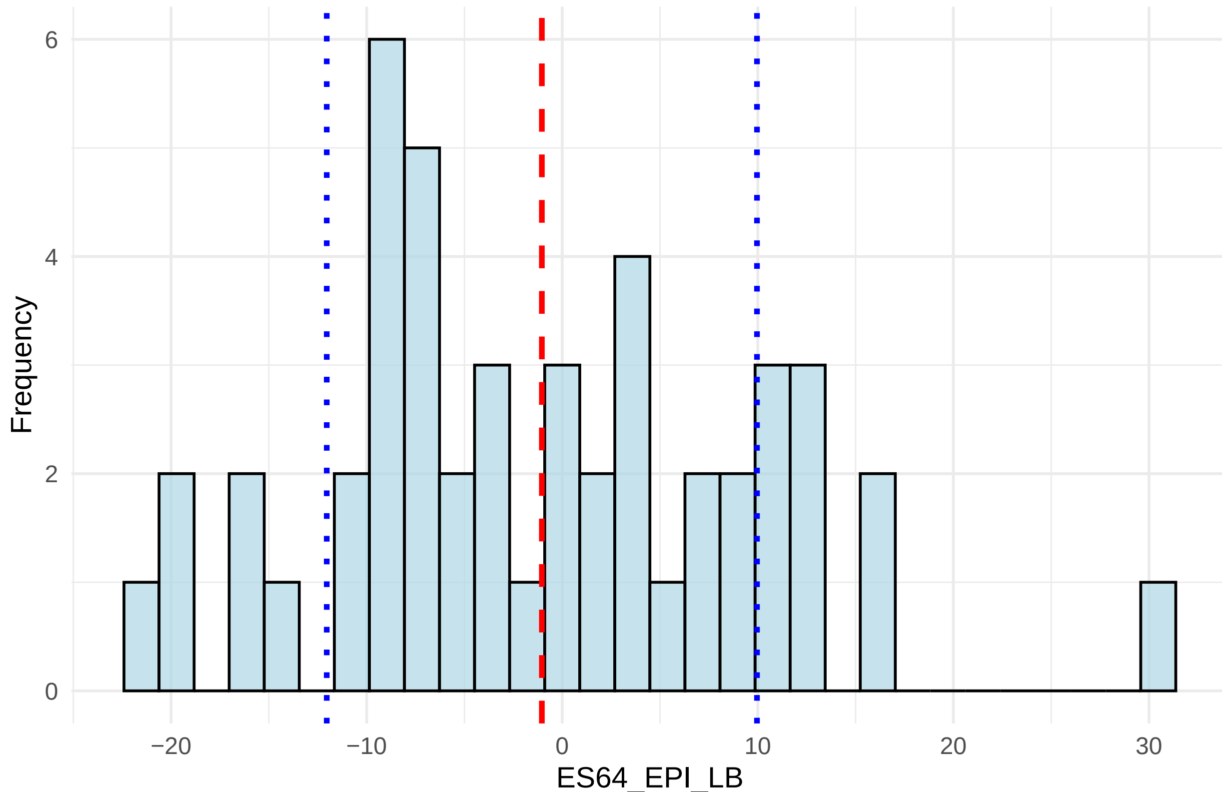


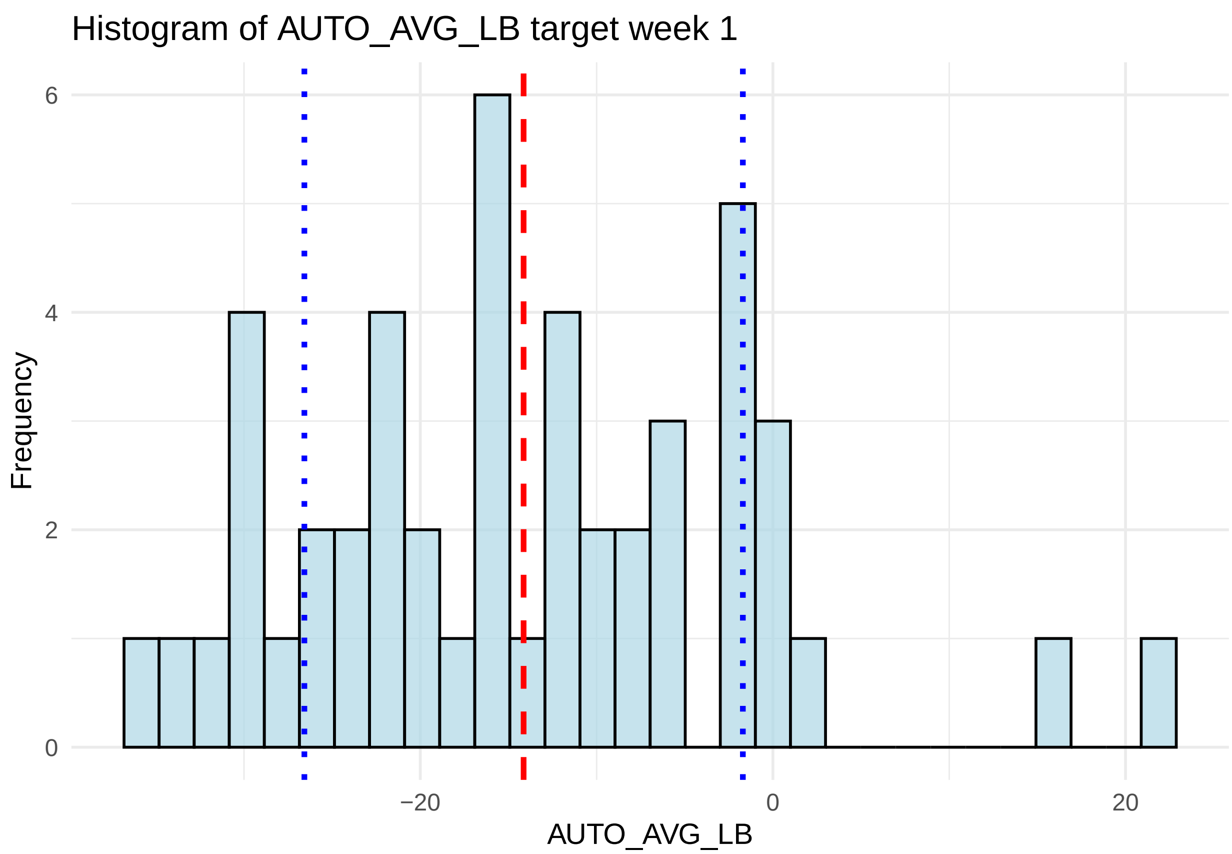
Histogram of AUTO_EPI_LB target week 1
Frequency
0
1
2
3
4
5
−20
−10
0
10
20
AUTO_EPI_LB



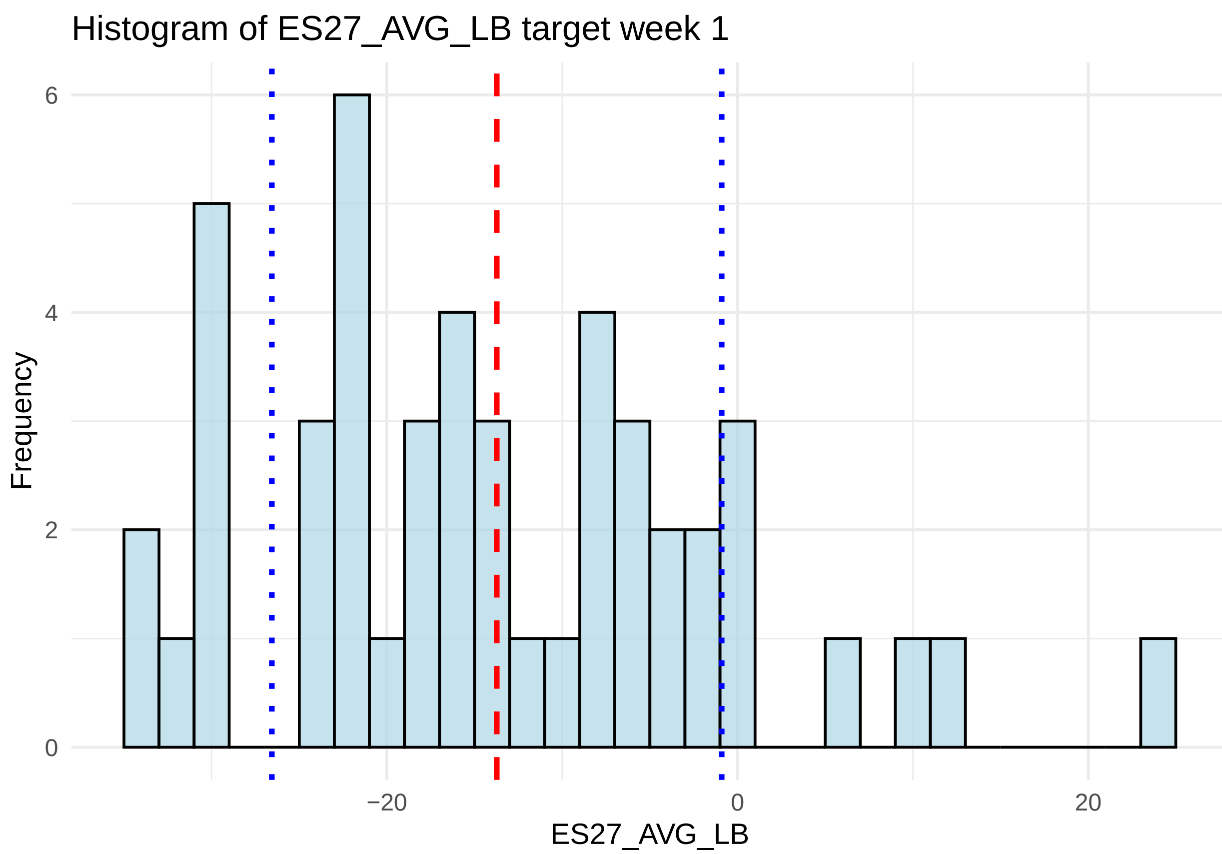
Histogram of ES27_EPI_LB target week 1
Frequency
7.5
5.0
2.5
0.0
−20
−10
0
10
20
30
ES27_EPI_LB

Histogram of ES64_EPI_LB target week 1

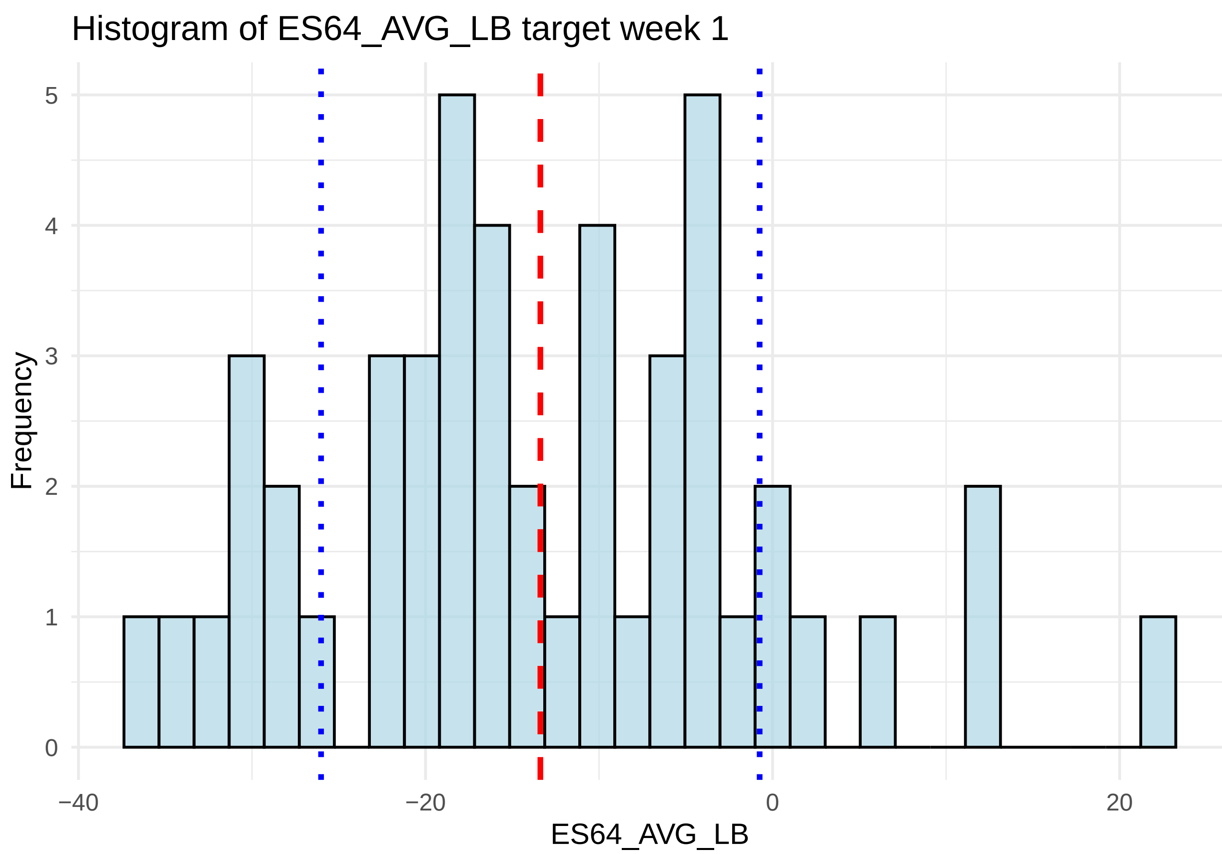


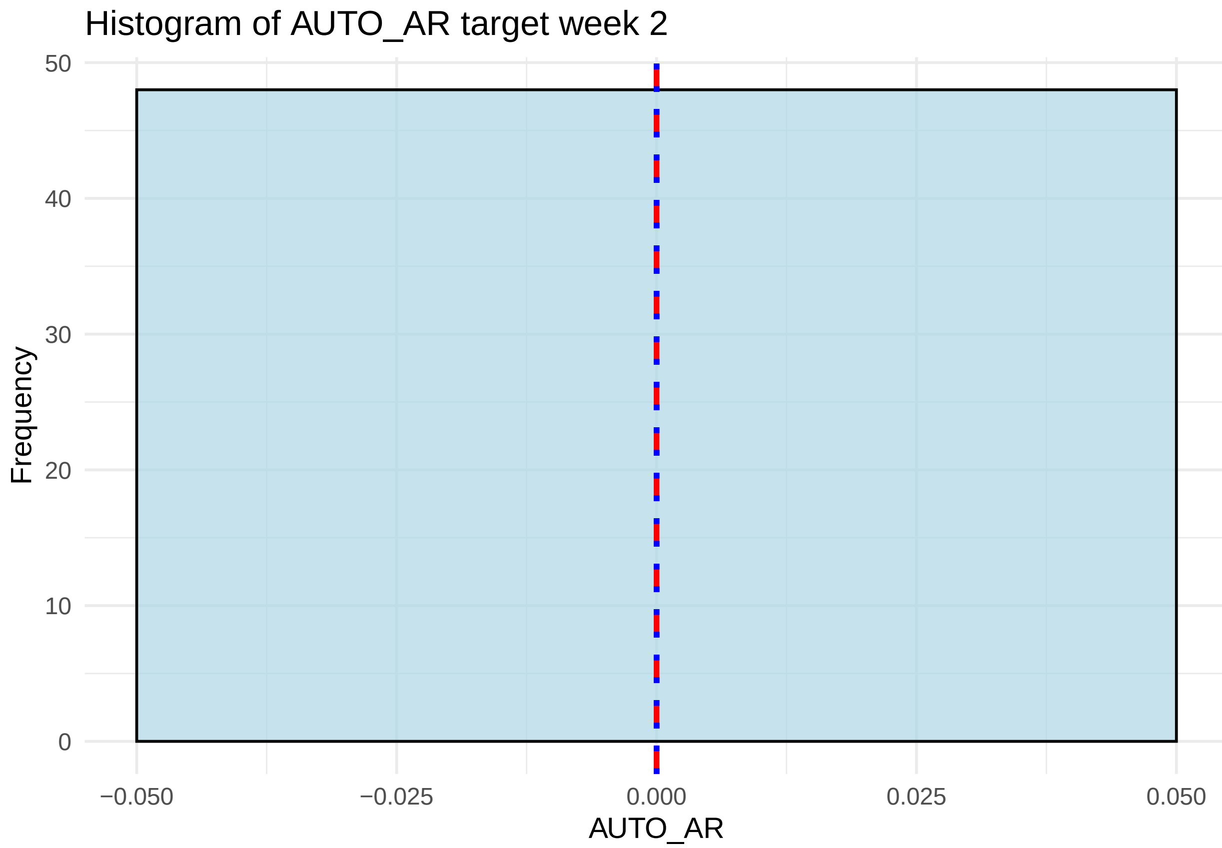


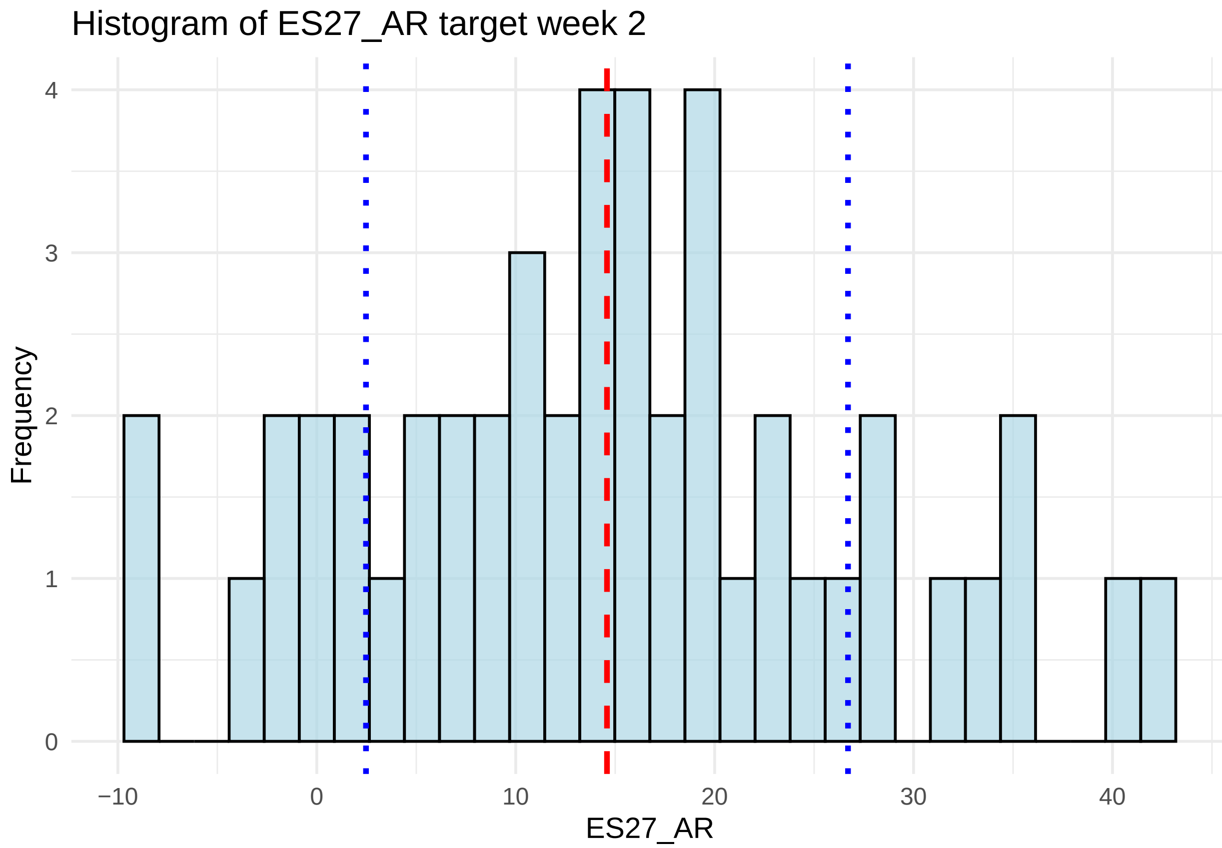
Histogram of AUTO_AVG_LB target week 1
Frequency
0
2
4
6
−20
0
20
AUTO_AVG_LB

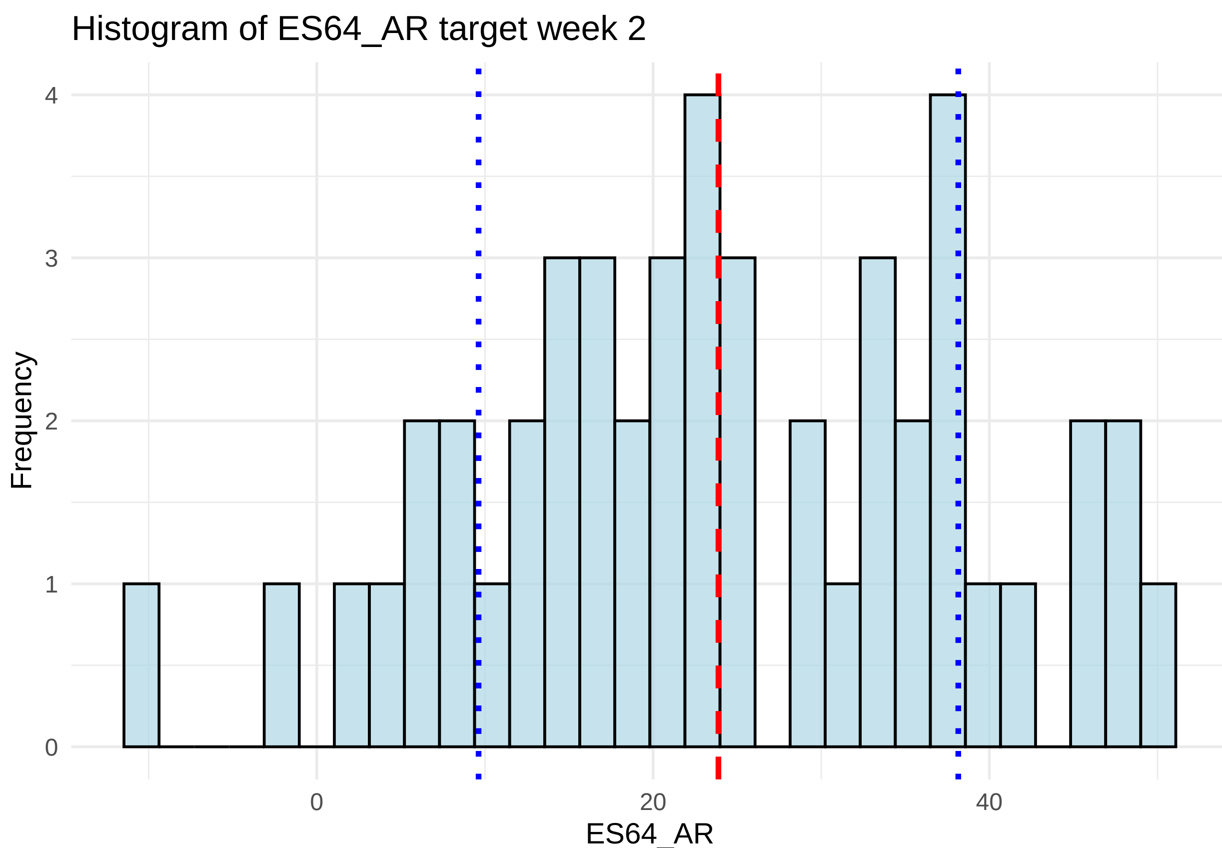


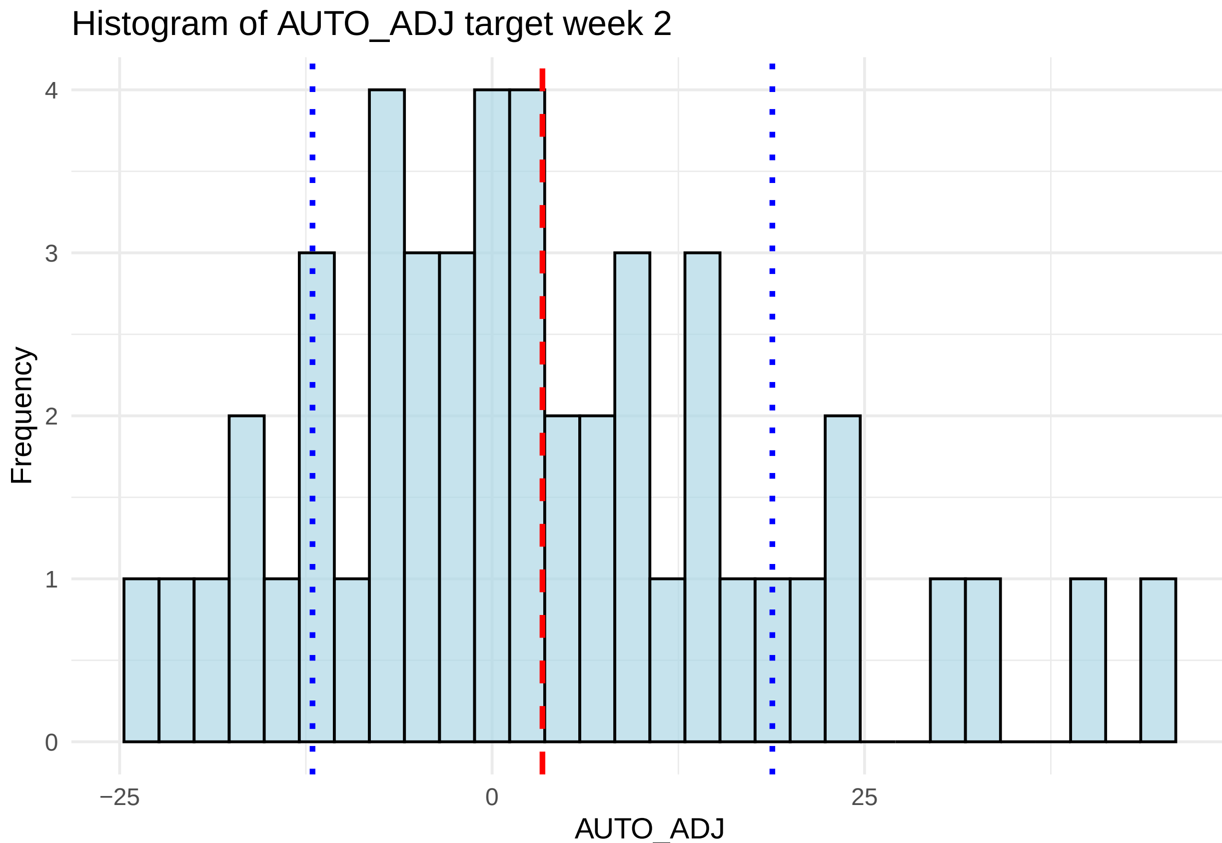
Histogram of ES27_AVG_LB target week 1
Frequency
6
4
2
0
−20
0
20
ES27_AVG_LB

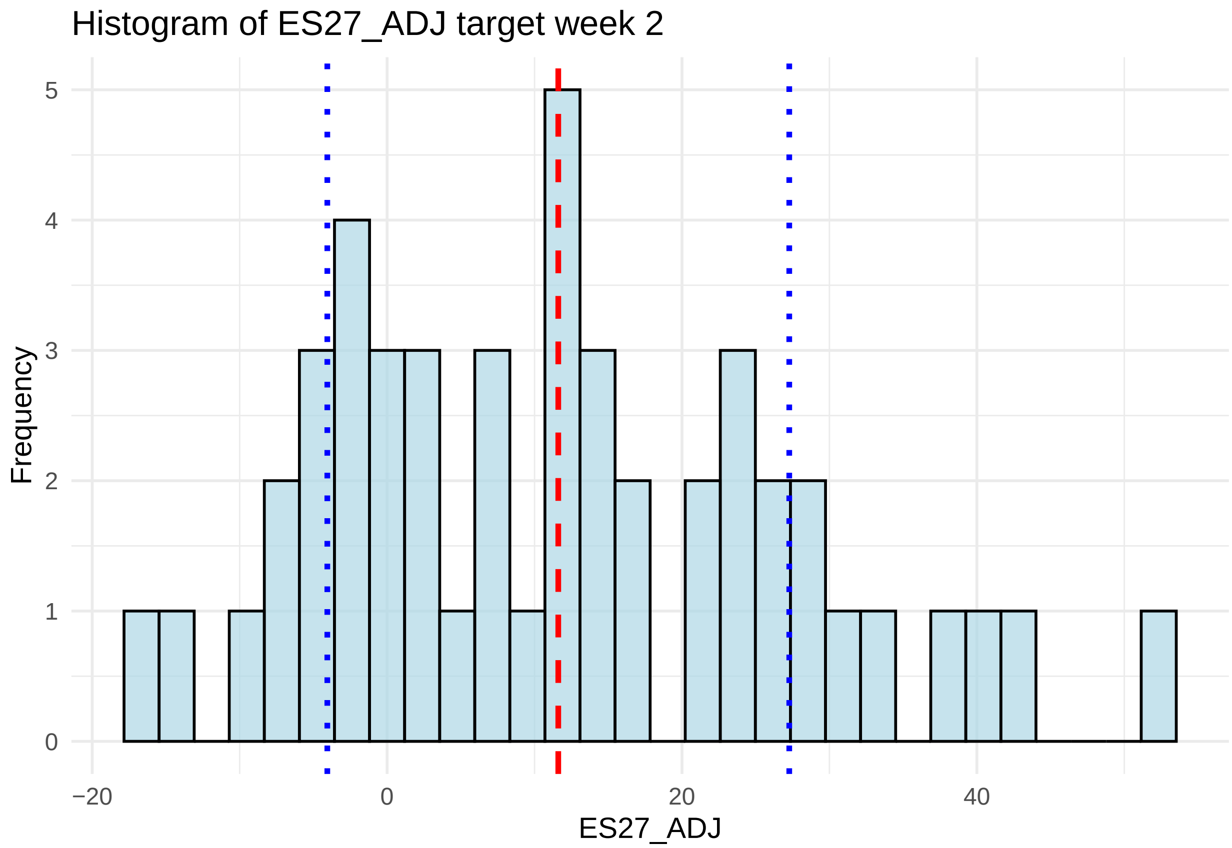

Histogram of ES64_AVG_LB target week 1
Frequency
5
4
3
2
1
0
−40
−20
0
20
ES64_AVG_LB

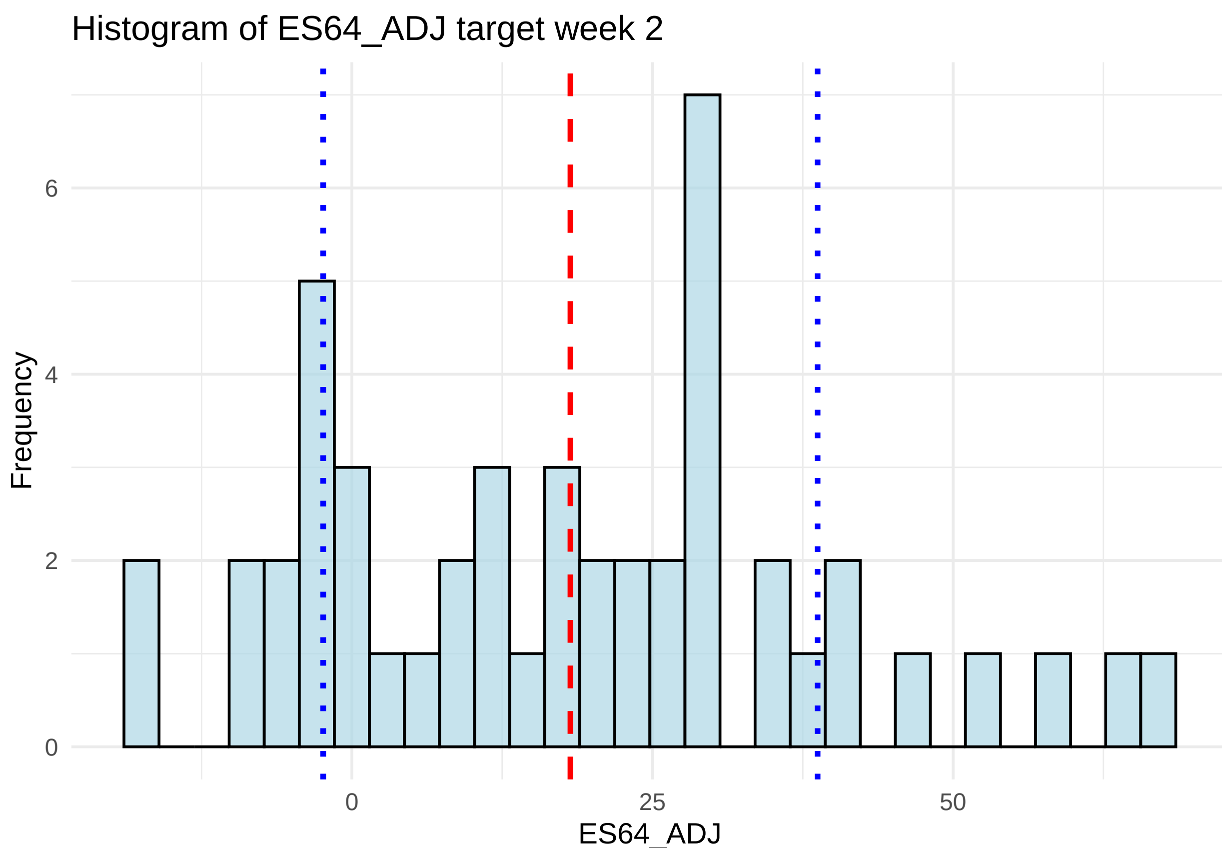

Histogram of AUTO_AR target week 2
50
40
30
20
10
0
Frequency
−0.050
−0.025
0.000
0.025
0.050
AUTO_AR

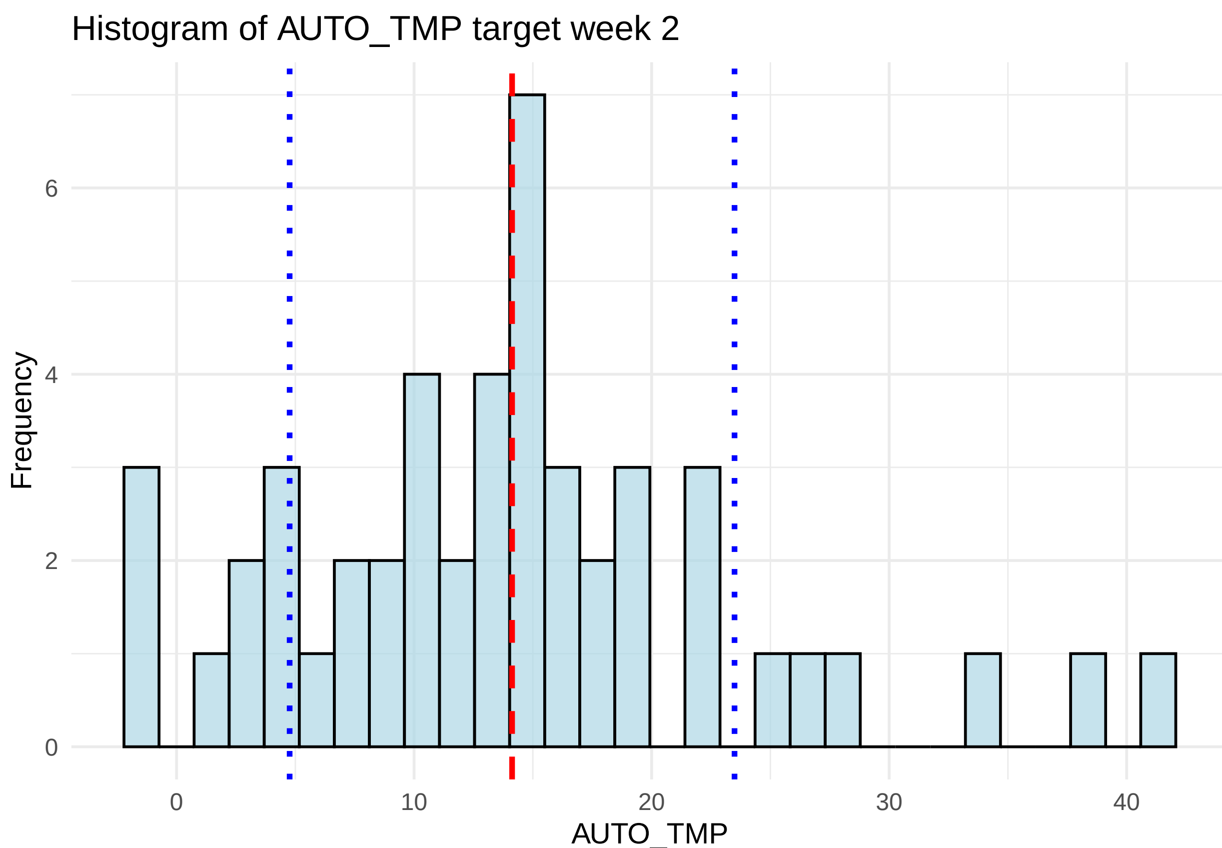

Histogram of ES27_AR target week 2
Frequency
ES27_AR
0
1
2
3
4
−10
0
10
20
30
40

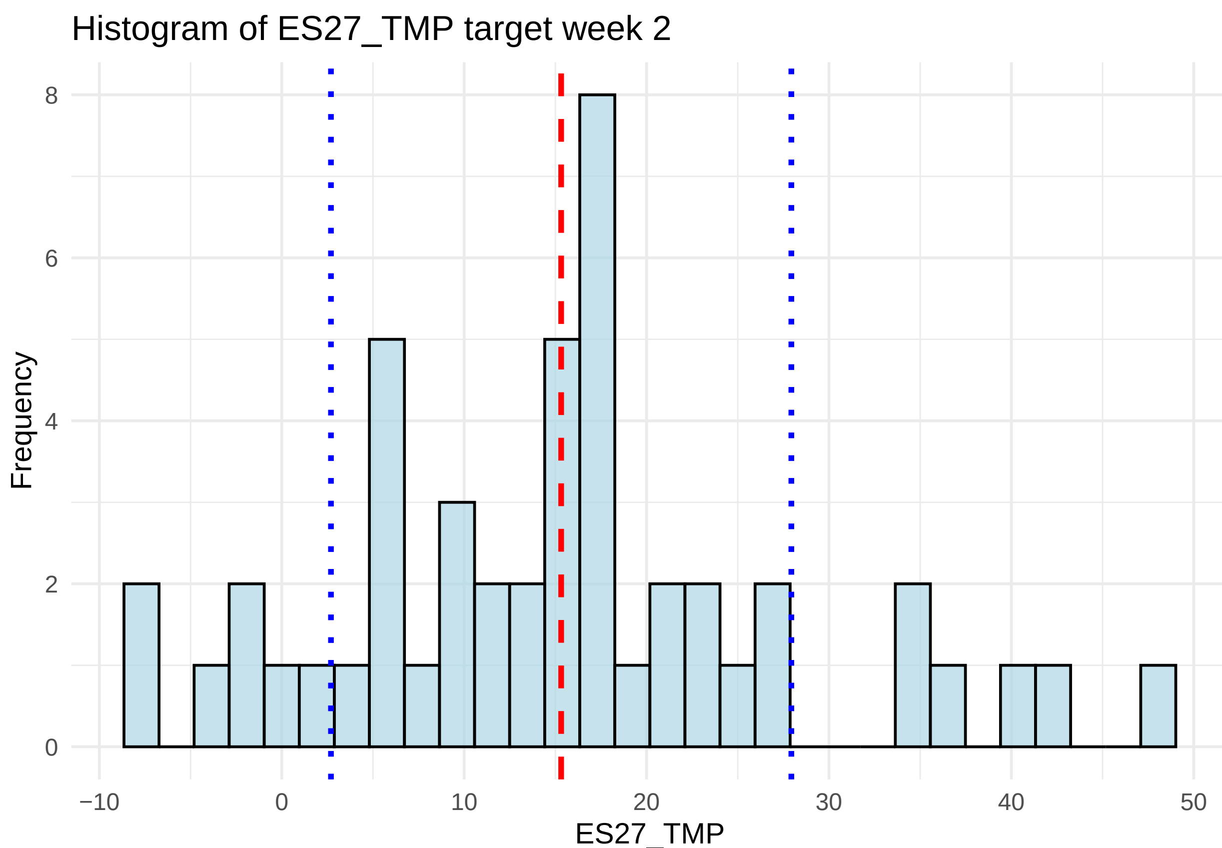

Histogram of ES64_AR target week 2
Frequency
4
3
2
1
0
0
20
40
ES64_AR

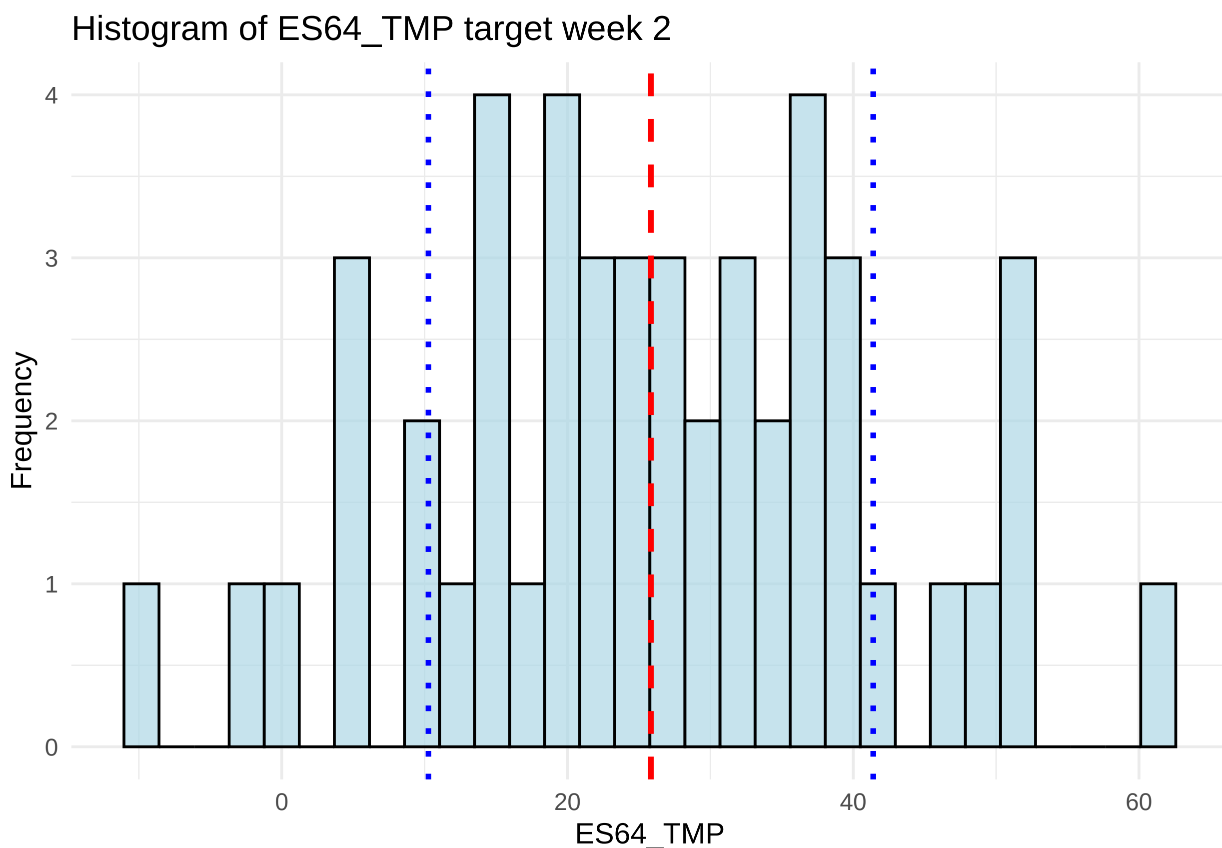

Histogram of AUTO_ADJ target week 2
Frequency
4
3
2
1
0
−25
0
25
AUTO_ADJ

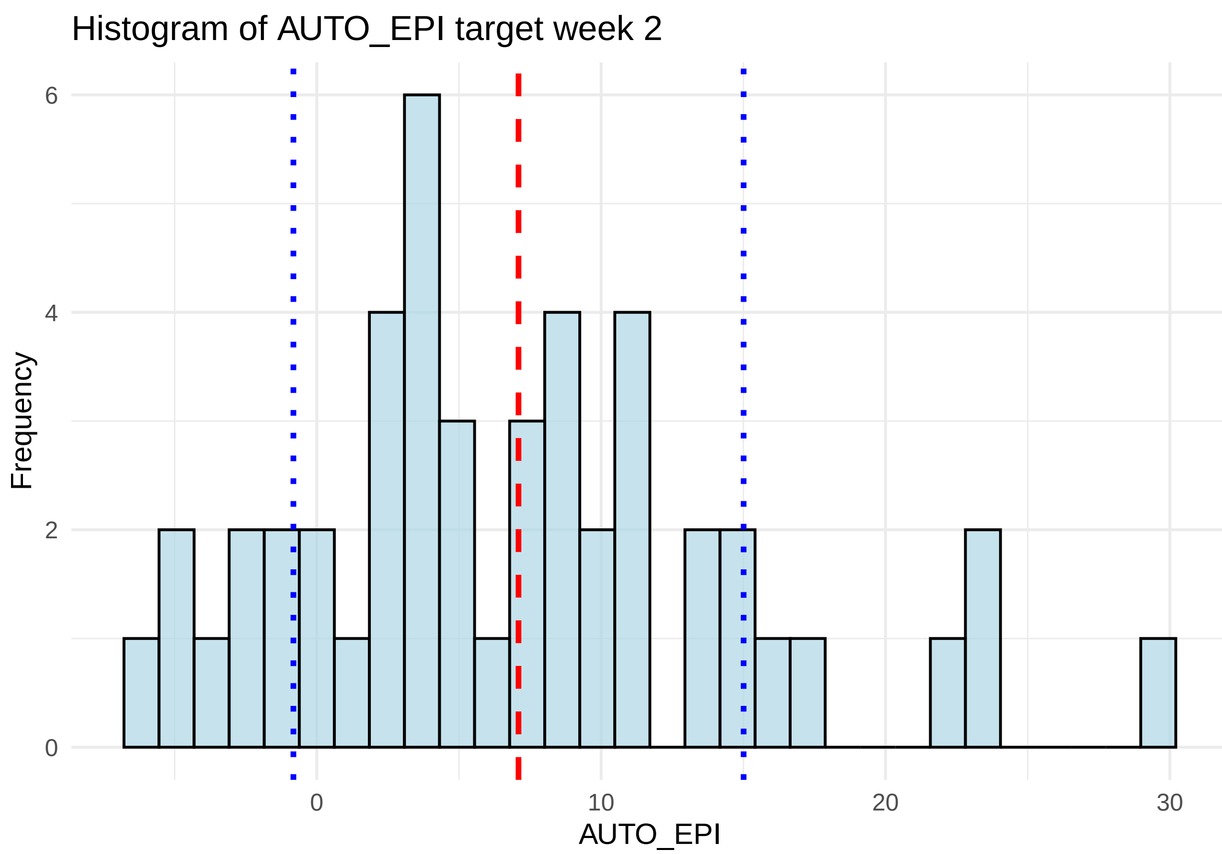

Histogram of ES27_ADJ target week 2
Frequency
5
4
3
2
1
0
−20
0
20
40
ES27_ADJ


Histogram of ES64_ADJ target week 2
Frequency
0
2
4
6
0
25
50
ES64_ADJ

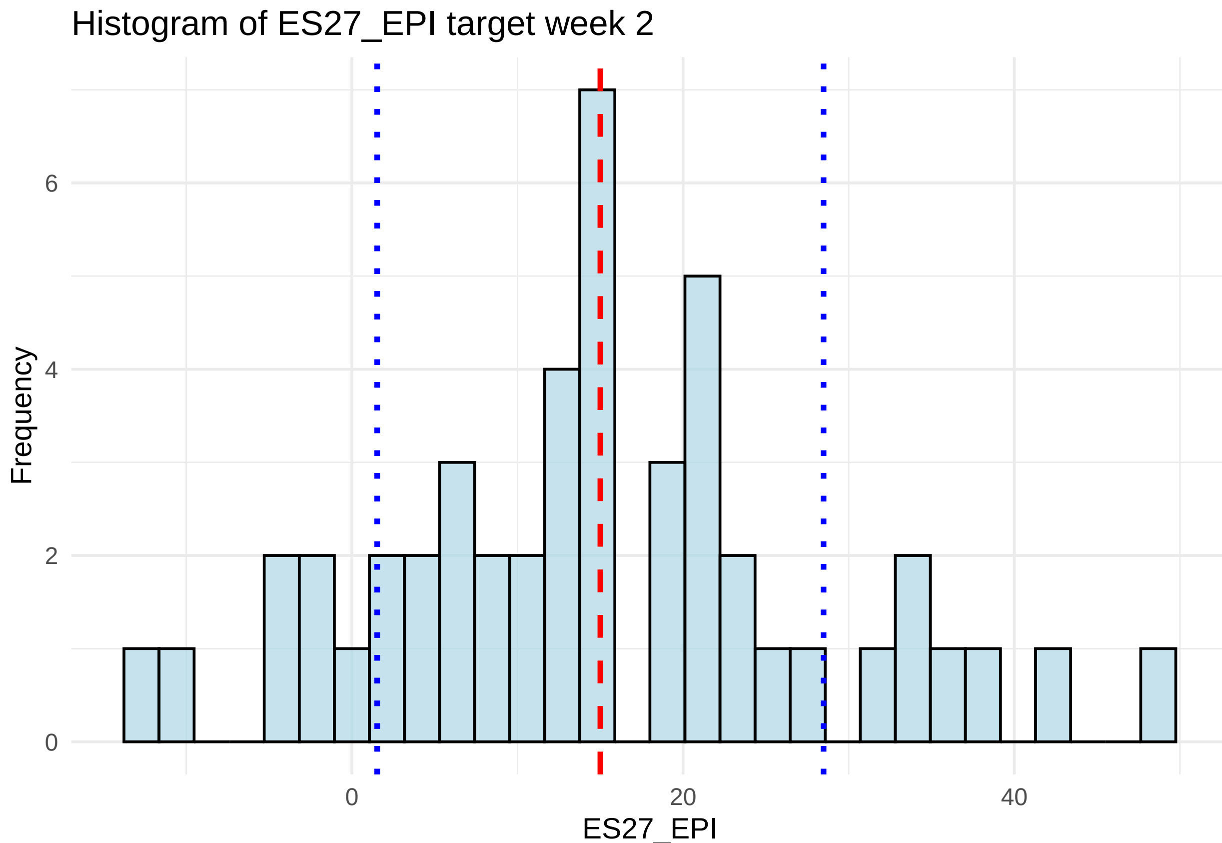

Histogram of AUTO_TMP target week 2
Frequency
0
2
4
6
0
10
20
30
40
AUTO_TMP

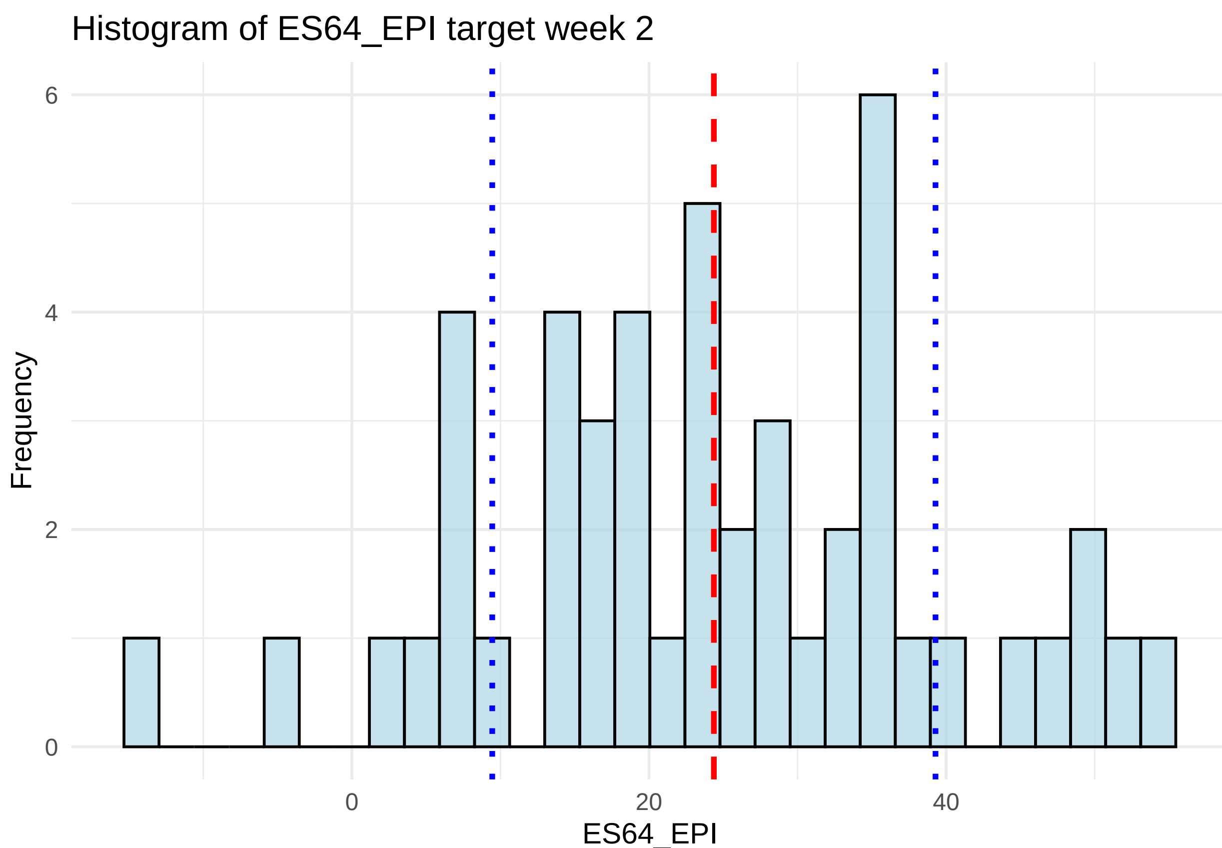

Histogram of ES27_TMP target week 2
Frequency
ES27_TMP
8
6
4
2
0
−10
0
10
20
30
40
50

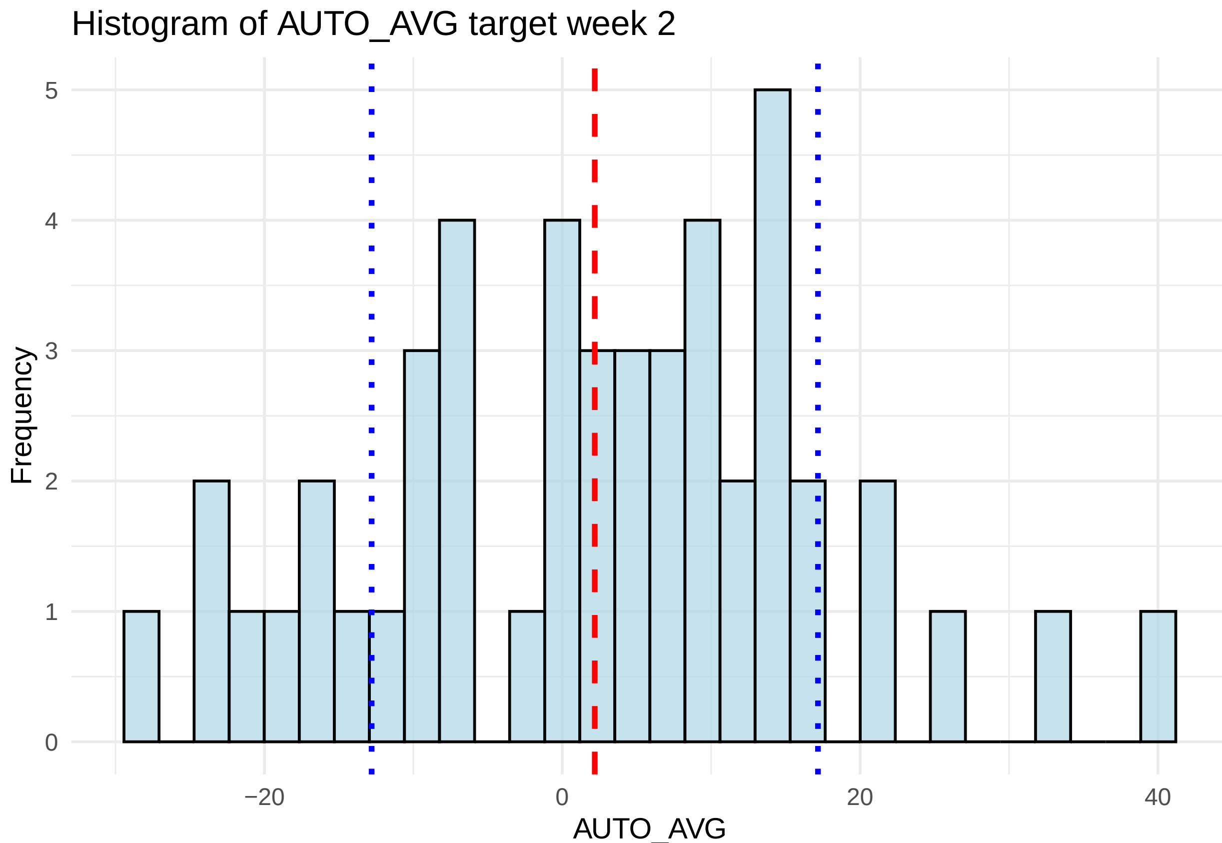

Histogram of ES64_TMP target week 2
Frequency
0
1
2
3
4
0
20
40
60
ES64_TMP

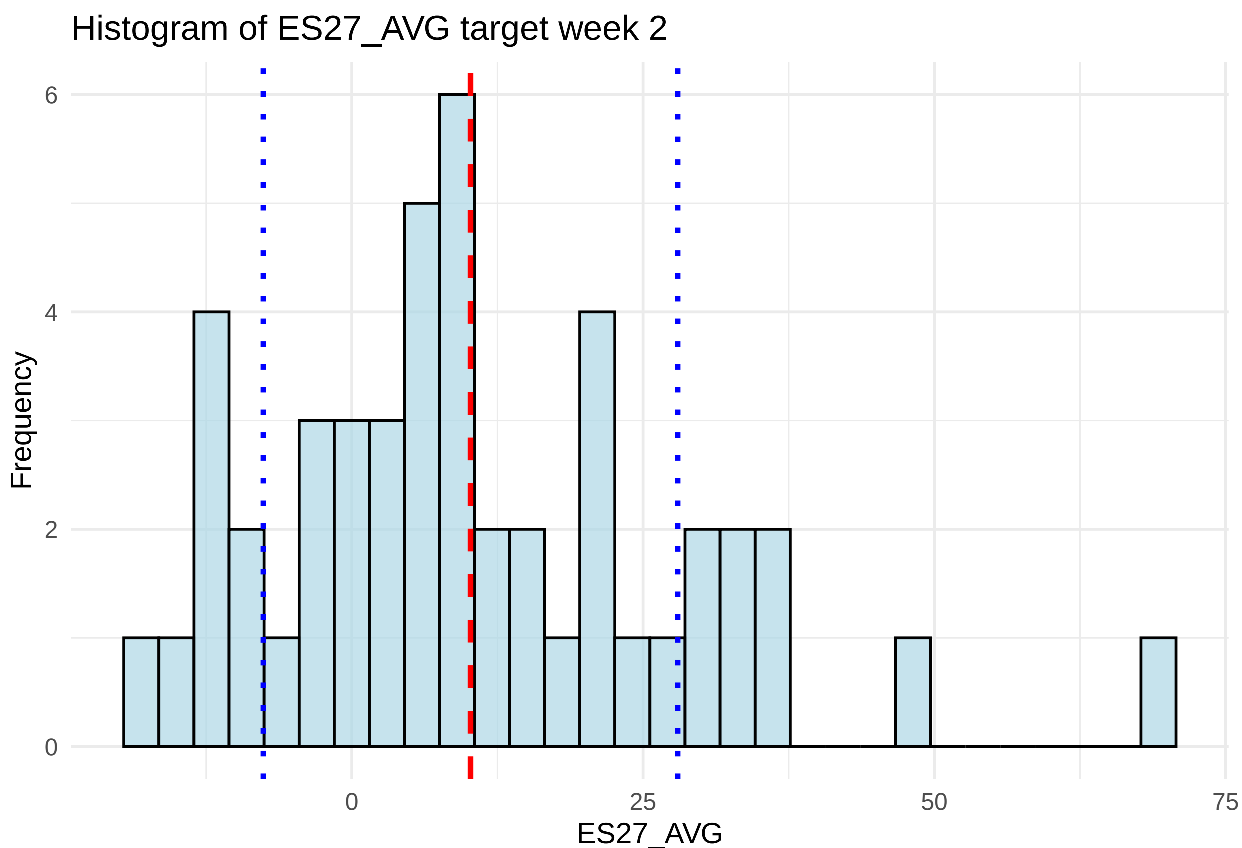


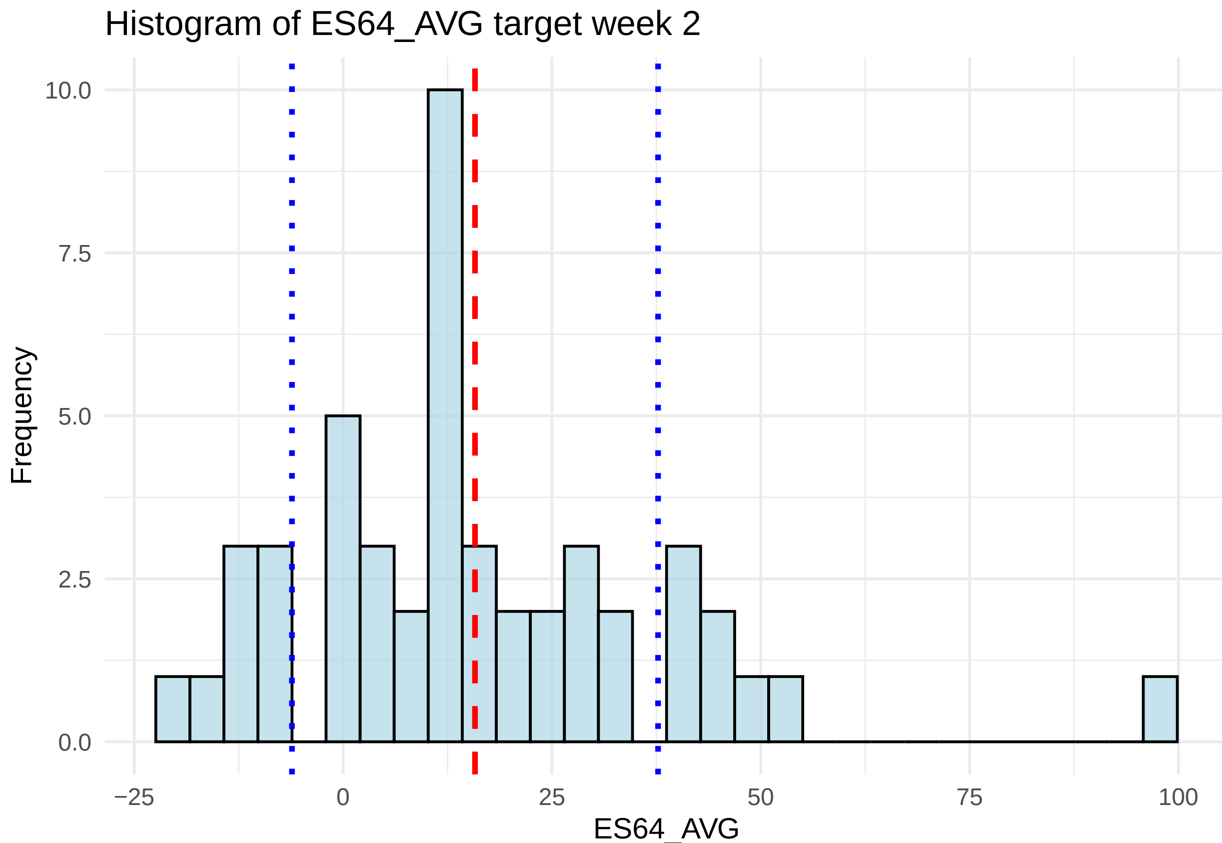
Histogram of AUTO_EPI target week 2
Frequency
0
2
4
6
0
10
20
30
AUTO_EPI

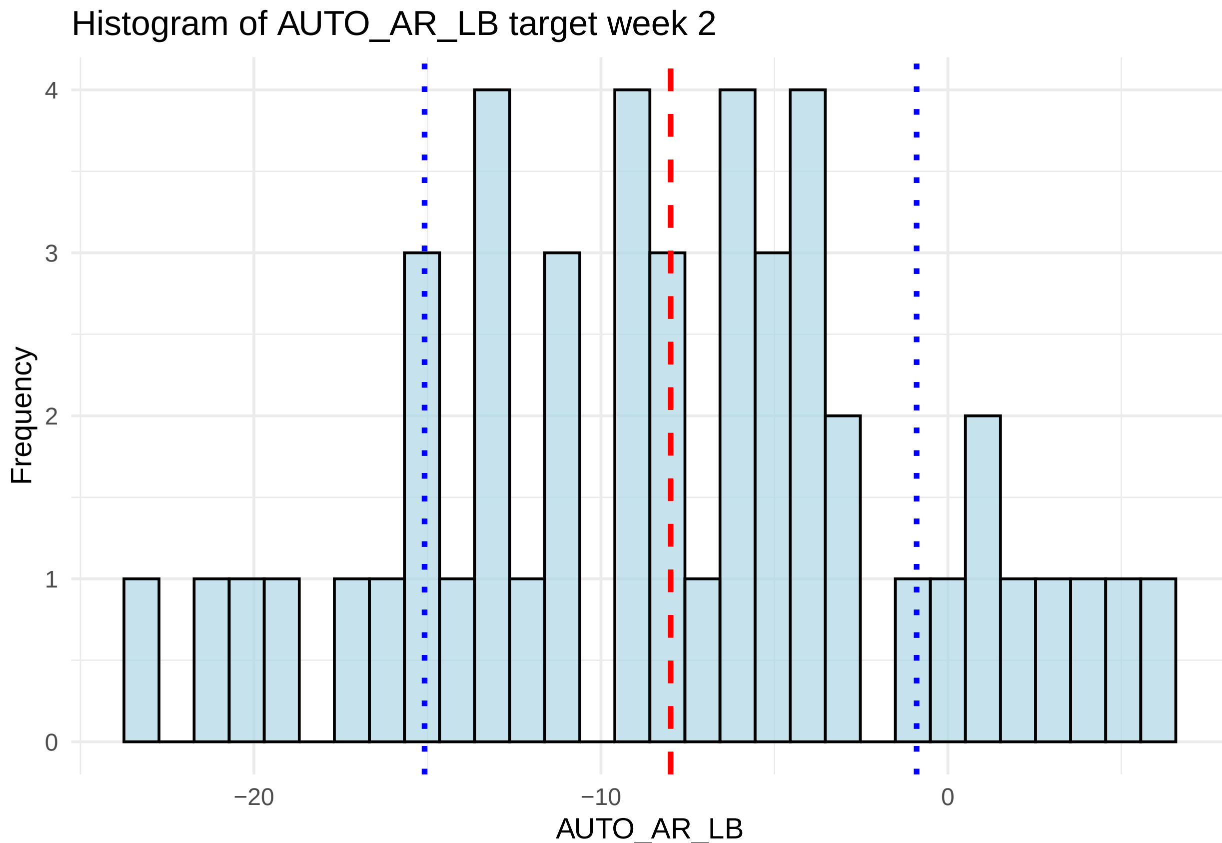

Histogram of ES27_EPI target week 2
Frequency
6
4
2
0
0
20
40
ES27_EPI

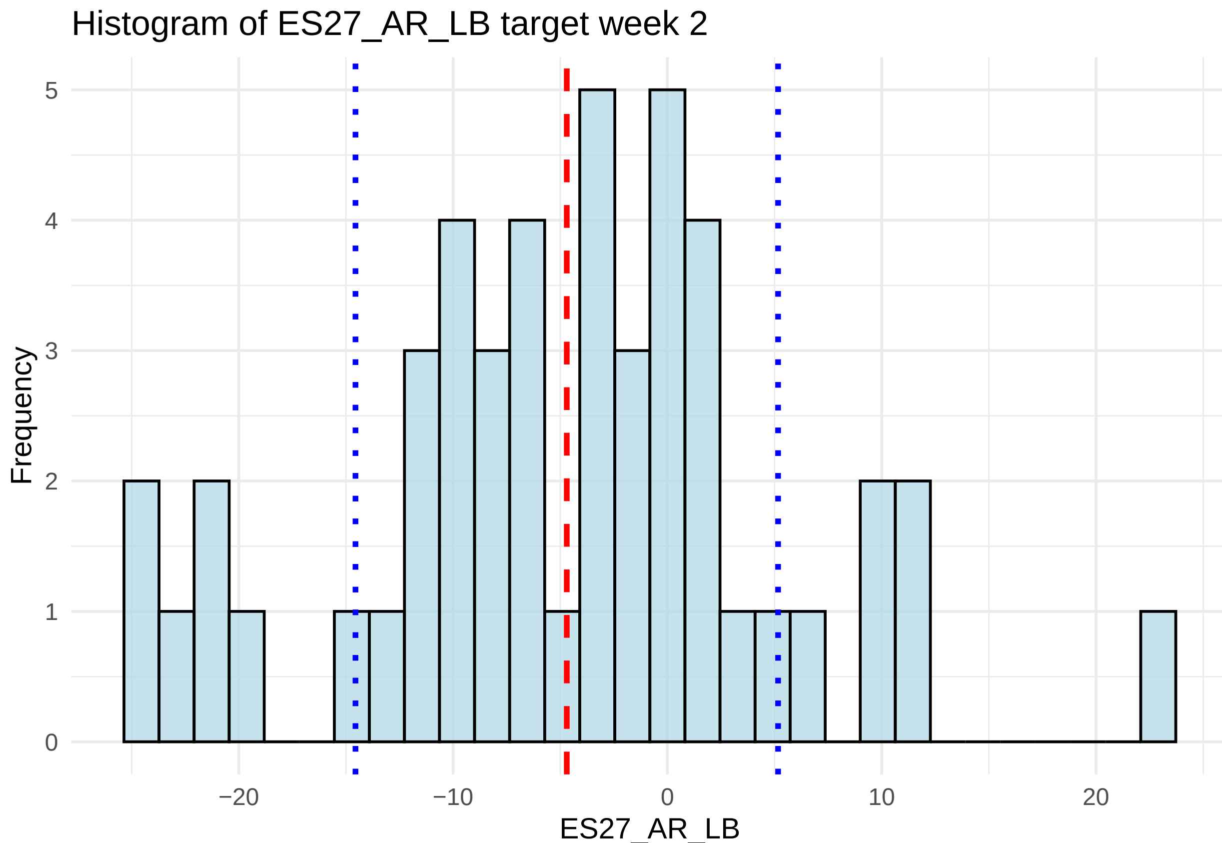

Histogram of ES64_EPI target week 2
Frequency
0
2
4
6
0
20
40
ES64_EPI

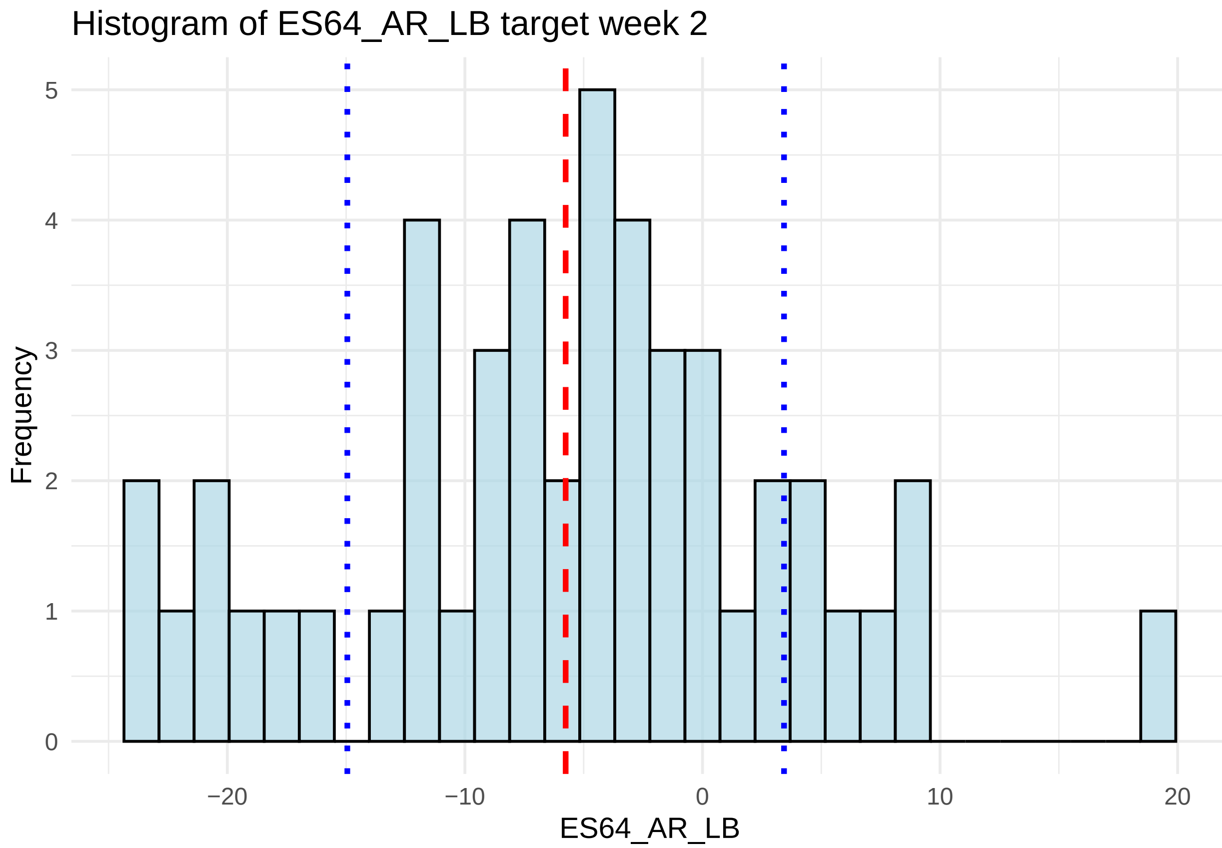

Histogram of AUTO_AVG target week 2
Frequency
5
4
3
2
1
0
−20
0
20
40
AUTO_AVG

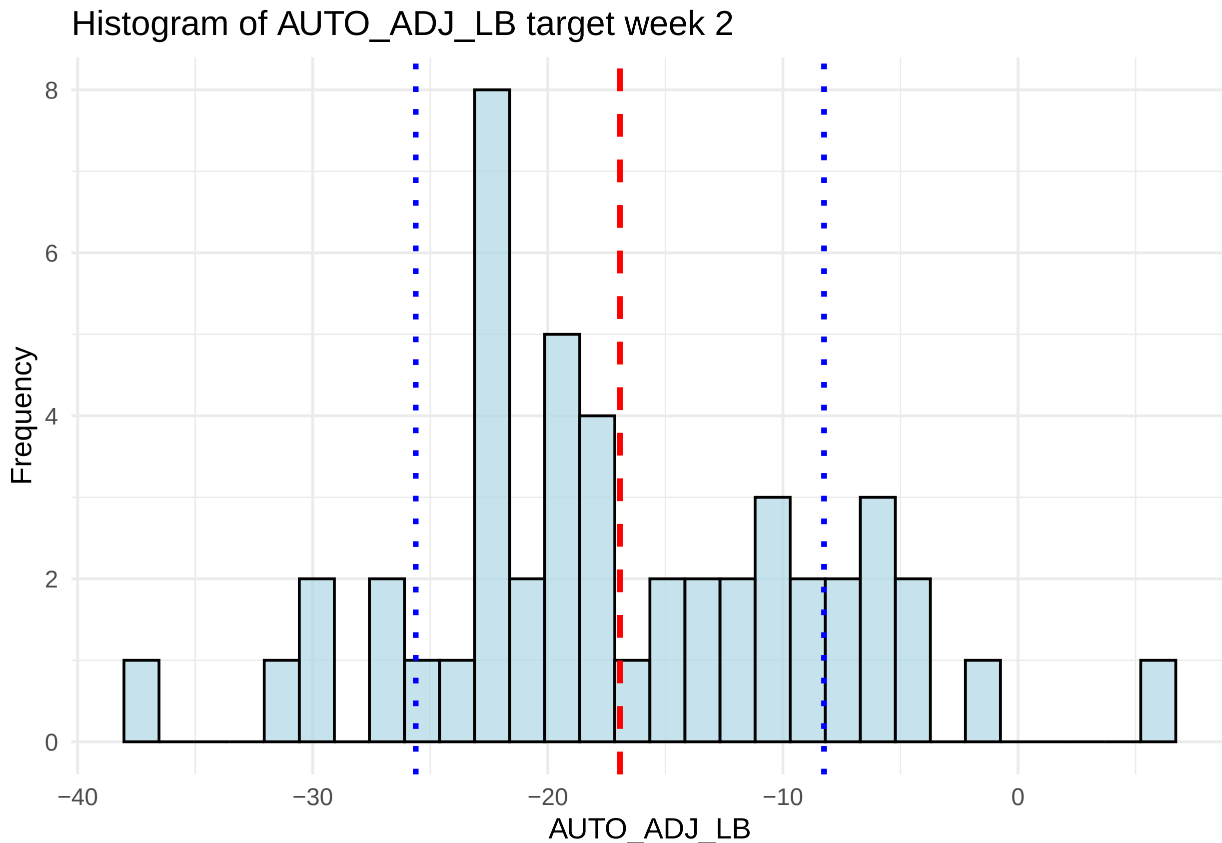

Histogram of ES27_AVG target week 2
Frequency
ES27_AVG
0
2
4
6
0
25
50
75

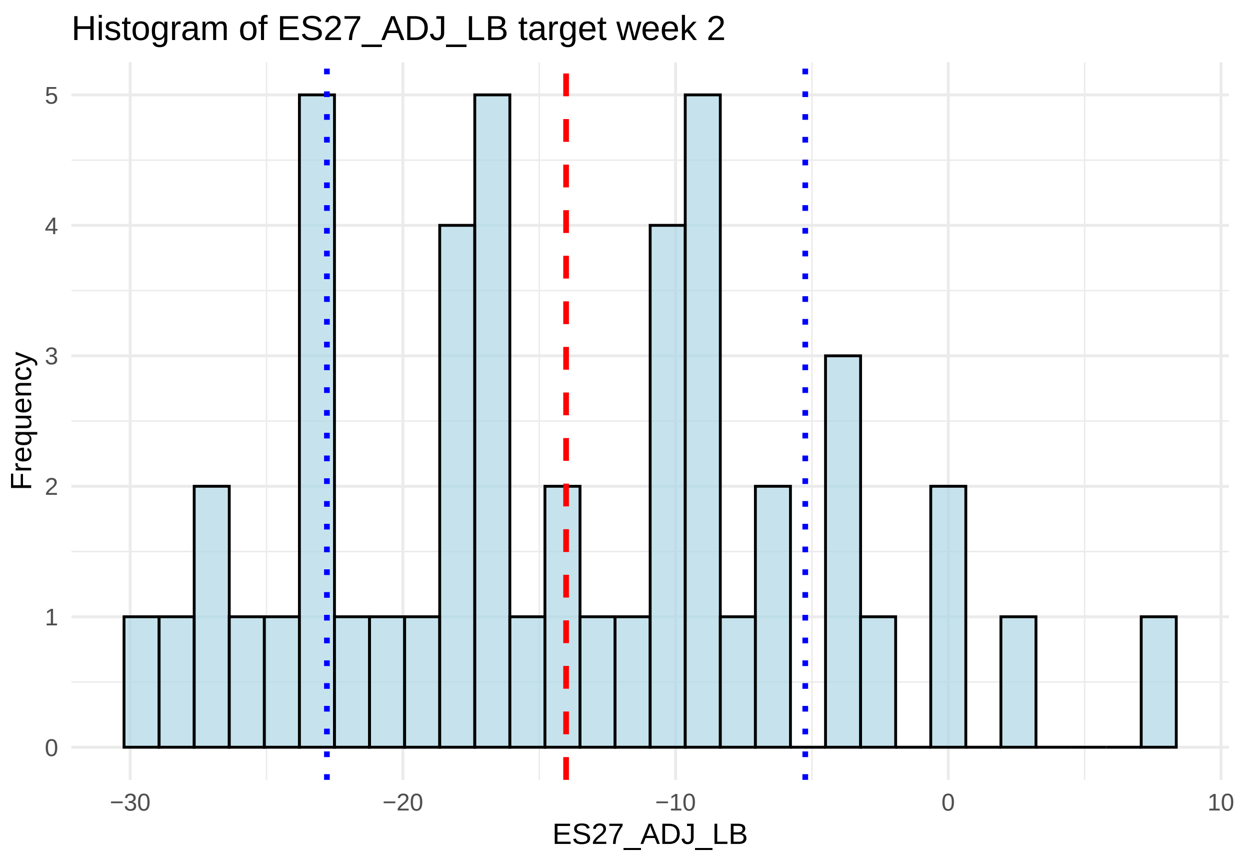

Histogram of ES64_AVG target week 2
Frequency
10.0
7.5
5.0
2.5
0.0
−25
0
25
50
75
100
ES64_AVG


Histogram of AUTO_AR_LB target week 2
Frequency
AUTO_AR_LB
4
3
2
1
0
−20
−10
0

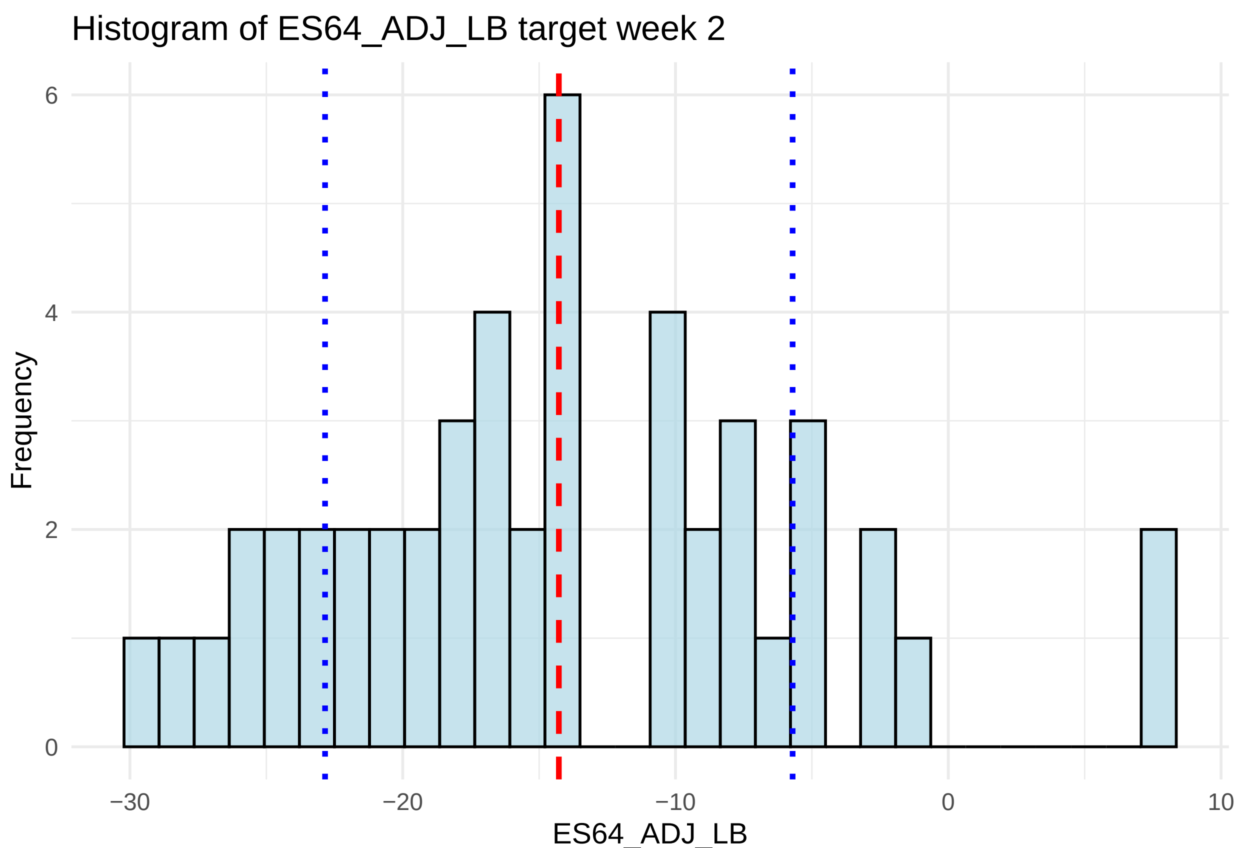

Histogram of ES27_AR_LB target week 2
Frequency
ES27_AR_LB
0
1
2
3
4
5
−20
−10
0
10
20

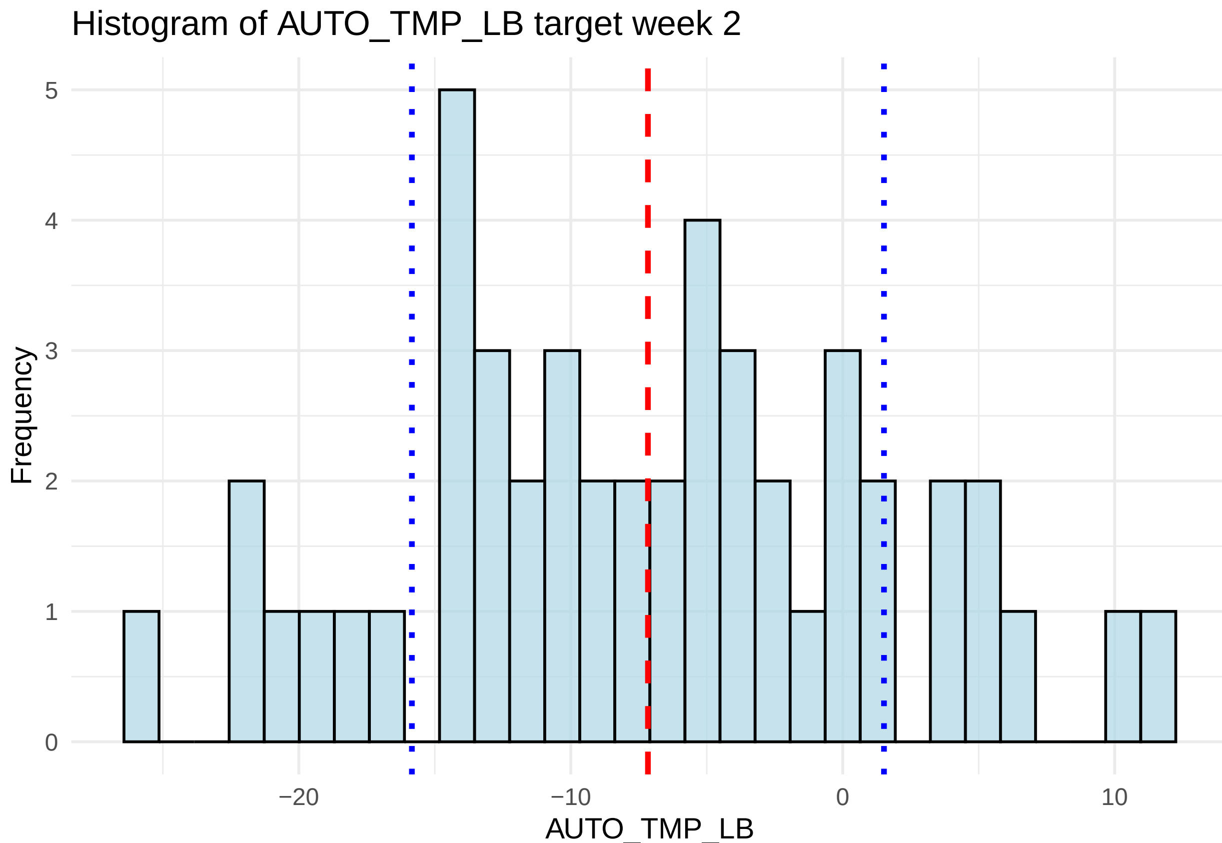

Histogram of ES64_AR_LB target week 2
Frequency
5
4
3
2
1
0
−20
−10
0
10
20
ES64_AR_LB


Histogram of AUTO_ADJ_LB target week 2
Frequency
8
6
4
2
0
−40
−30
−20
−10
0
AUTO_ADJ_LB

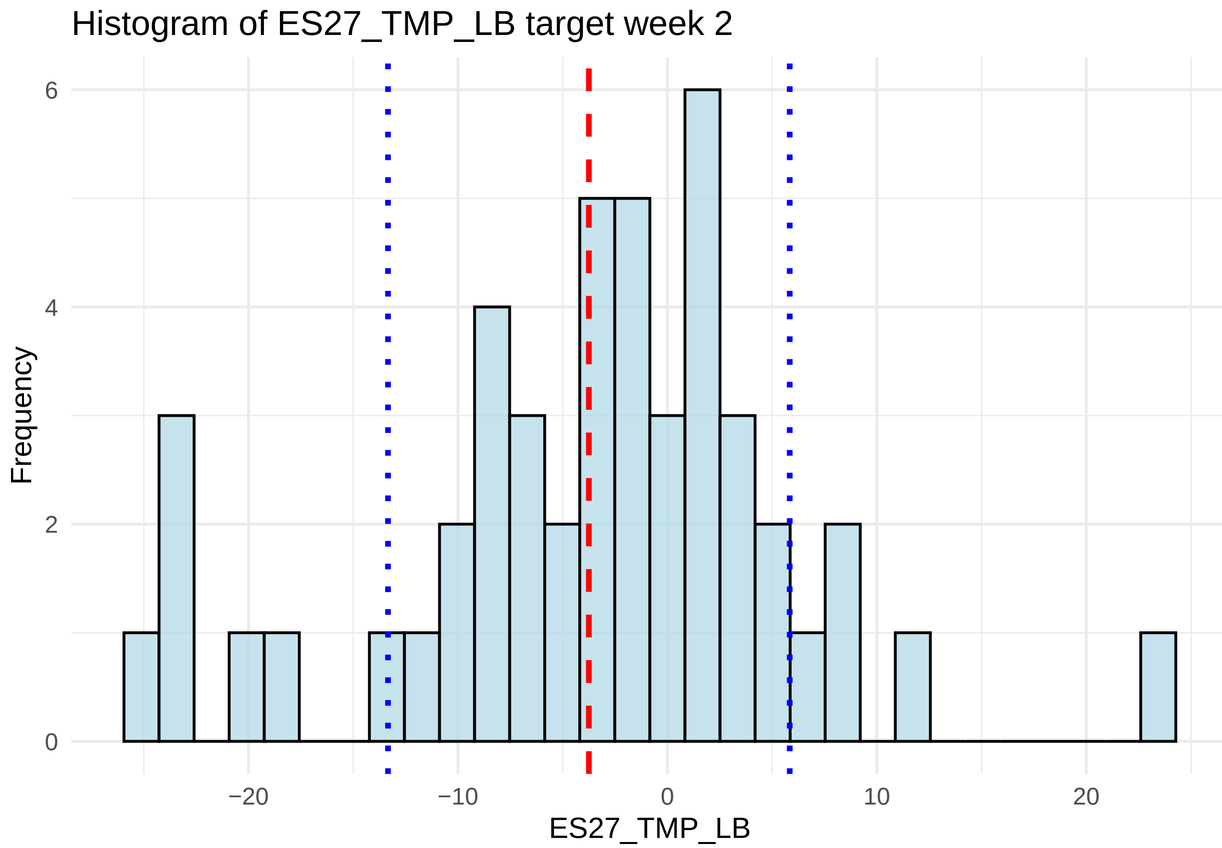


Histogram of ES27_ADJ_LB target week 2
Frequency
5
4
3
2
1
0
−30
−20
−10
0
10
ES27_ADJ_LB

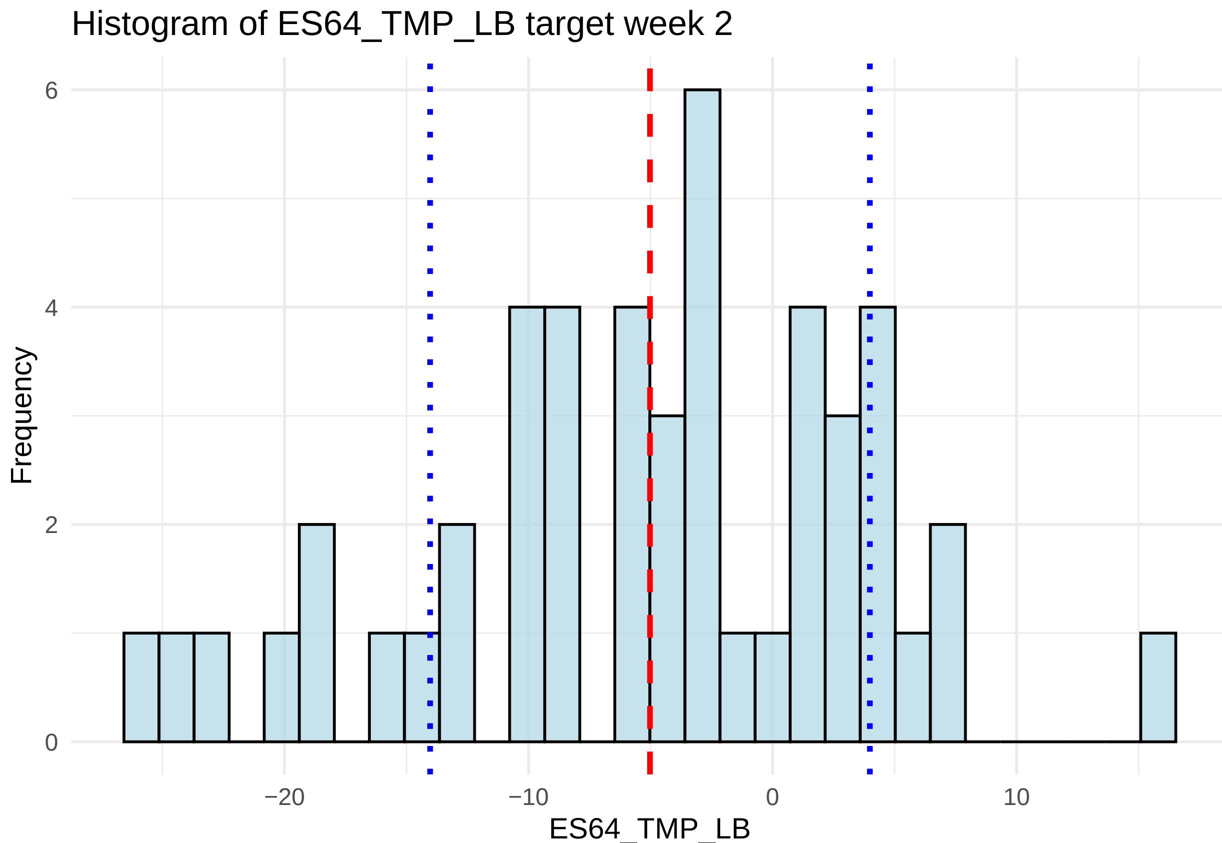

Histogram of ES64_ADJ_LB target week 2
Frequency
0
2
4
6
−30
−20
−10
0
10
ES64_ADJ_LB

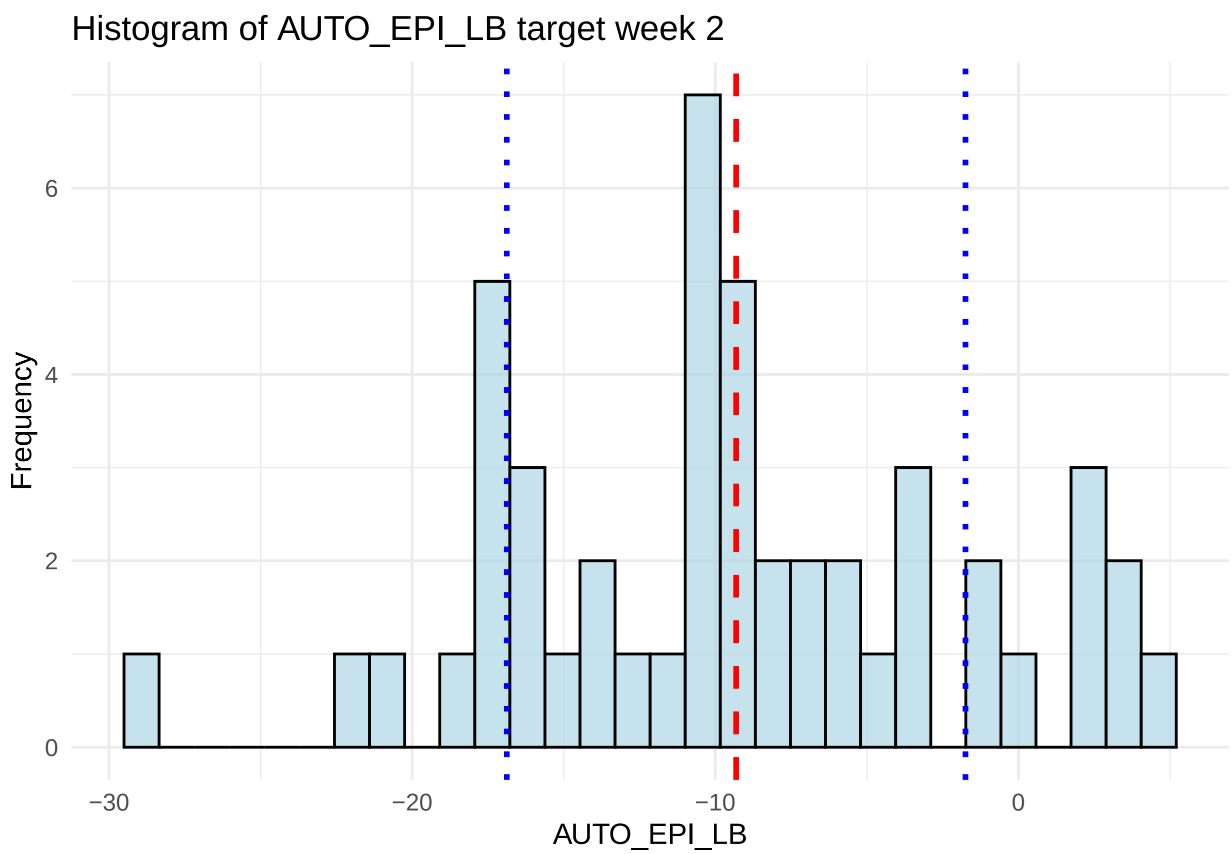


Histogram of AUTO_TMP_LB target week 2
Frequency
5
4
3
2
1
0
−20
−10
0
10
AUTO_TMP_LB

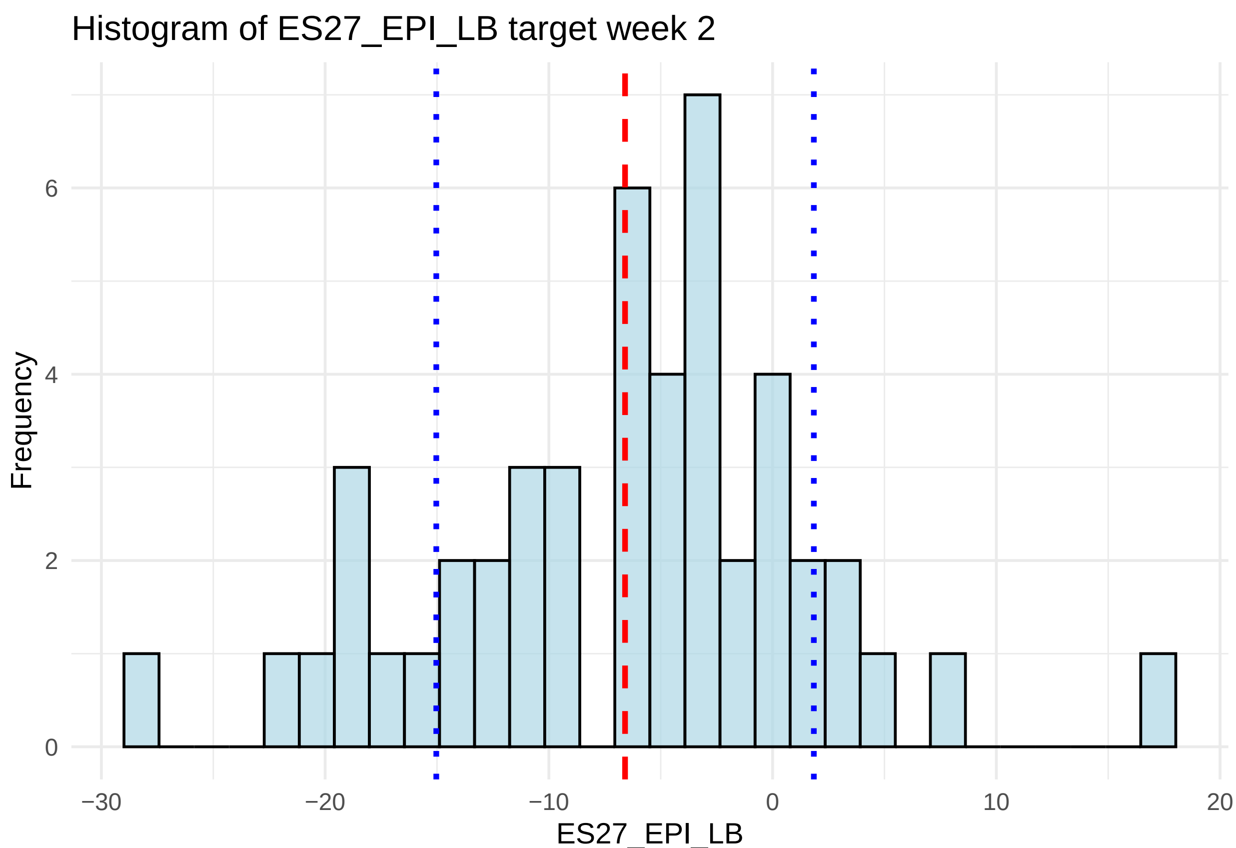


Histogram of ES27_TMP_LB target week 2
Frequency
ES27_TMP_LB
0
2
4
6
−20
−10
0
10
20

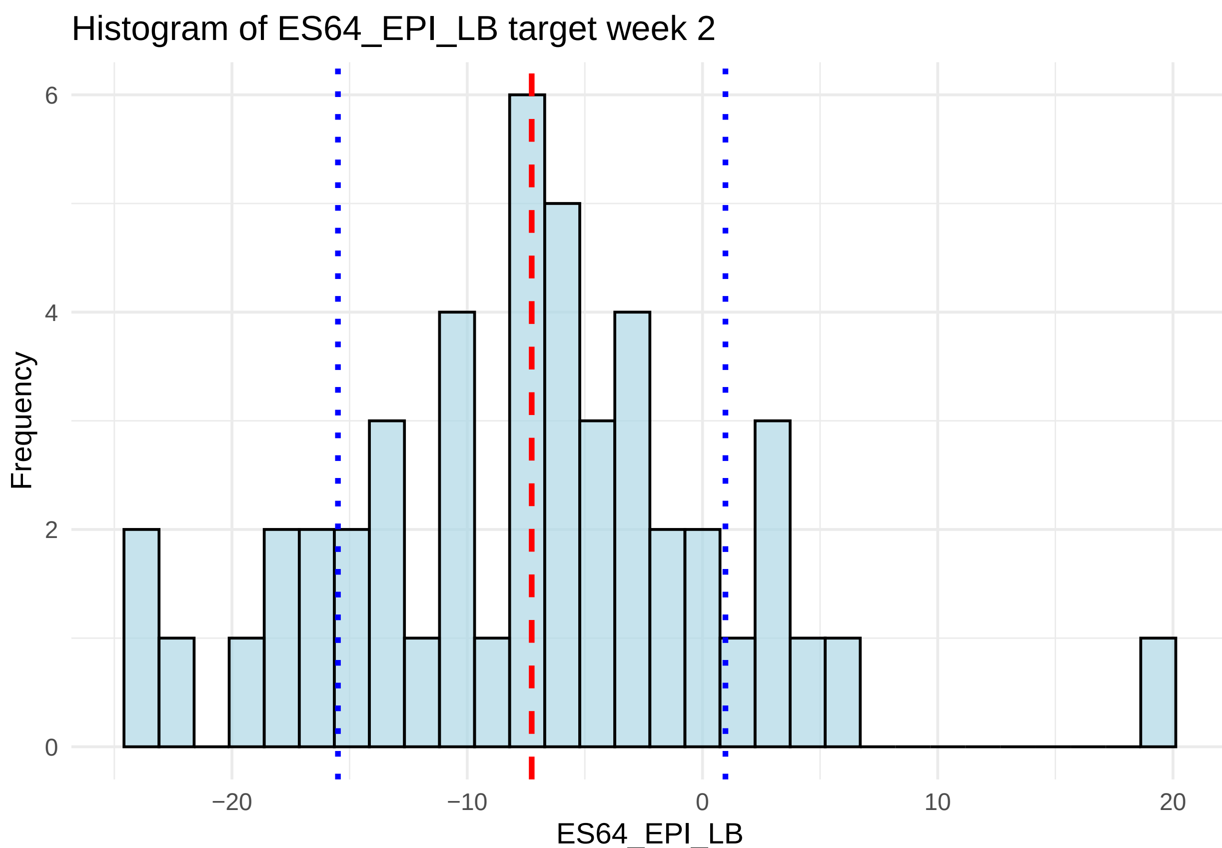

Histogram of ES64_TMP_LB target week 2
Frequency
6
4
2
0
−20
−10
0
10
ES64_TMP_LB

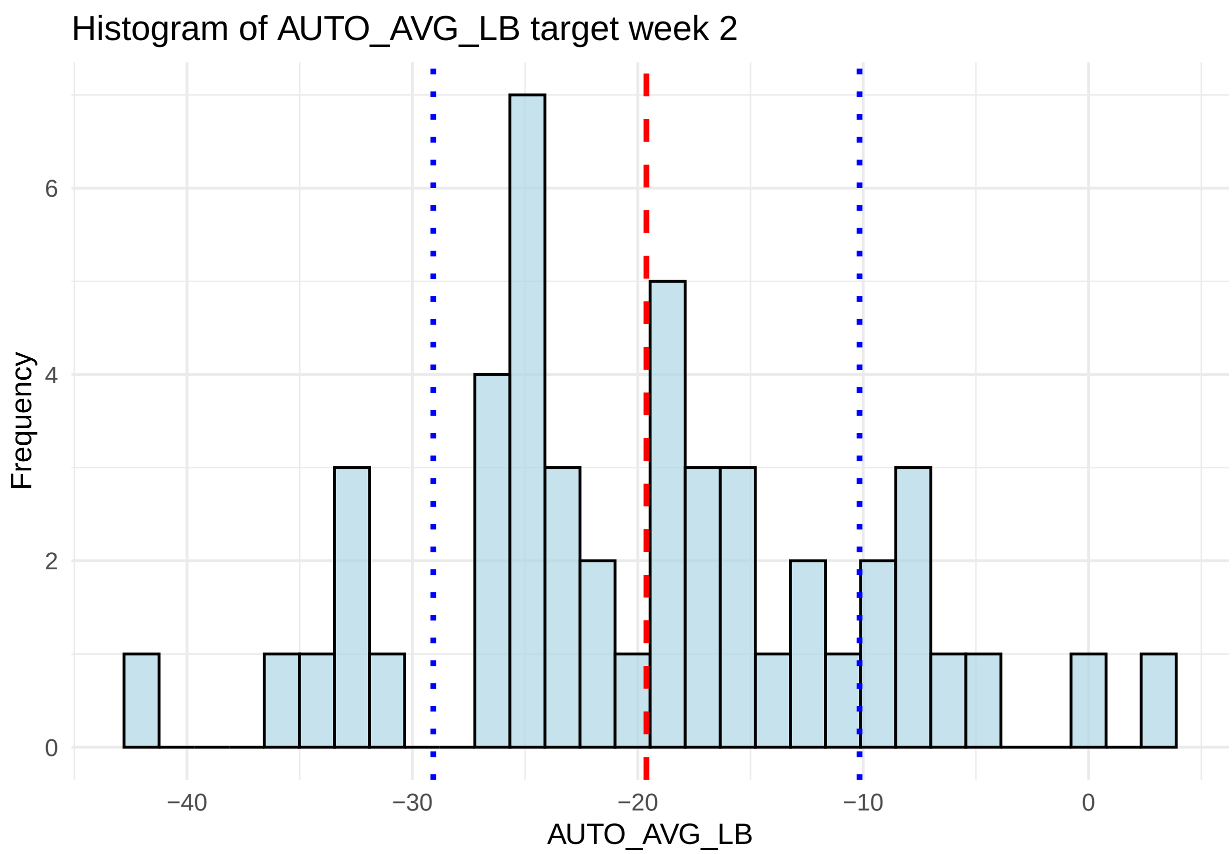


Histogram of AUTO_EPI_LB target week 2
Frequency
0
2
4
6
−30
−20
−10
0
AUTO_EPI_LB

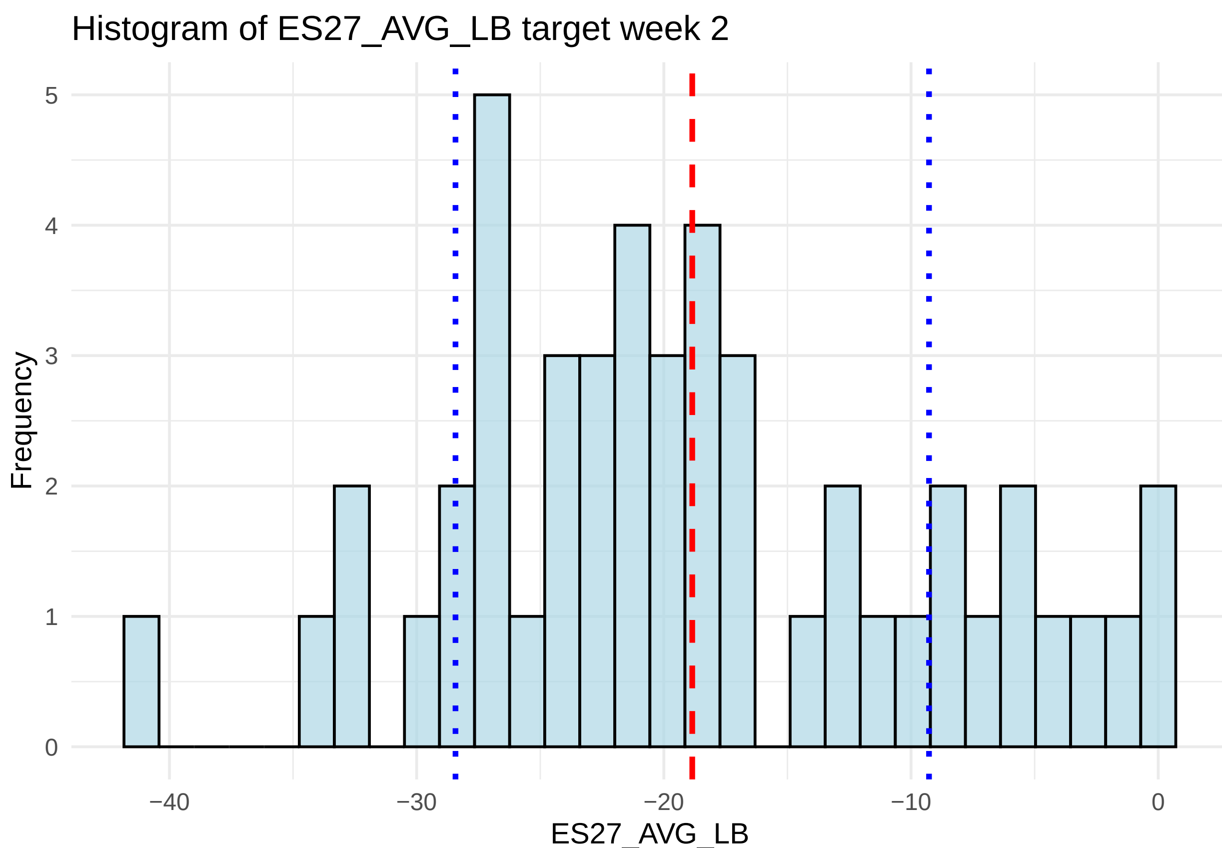


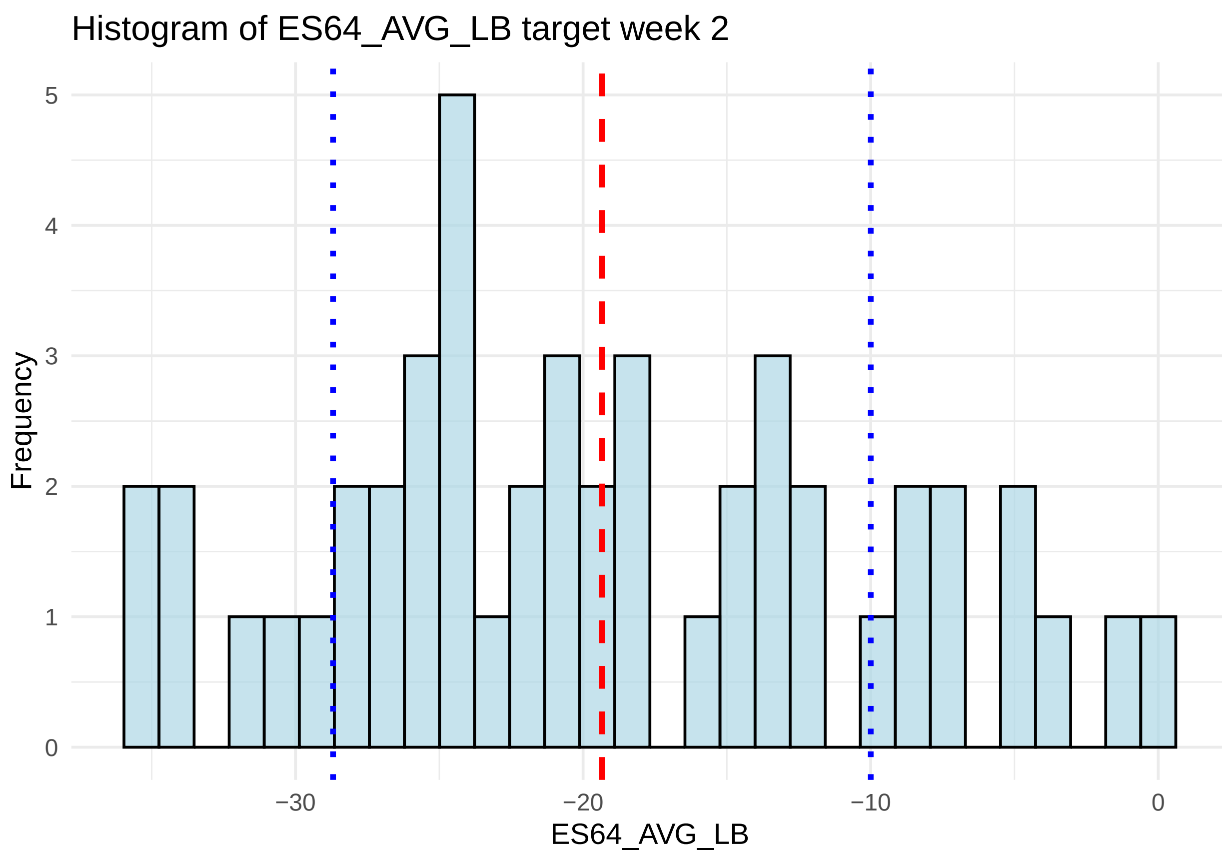
Histogram of ES27_EPI_LB target week 2
Frequency
0
2
4
6
−30
−20
−10
0
10
20
ES27_EPI_LB

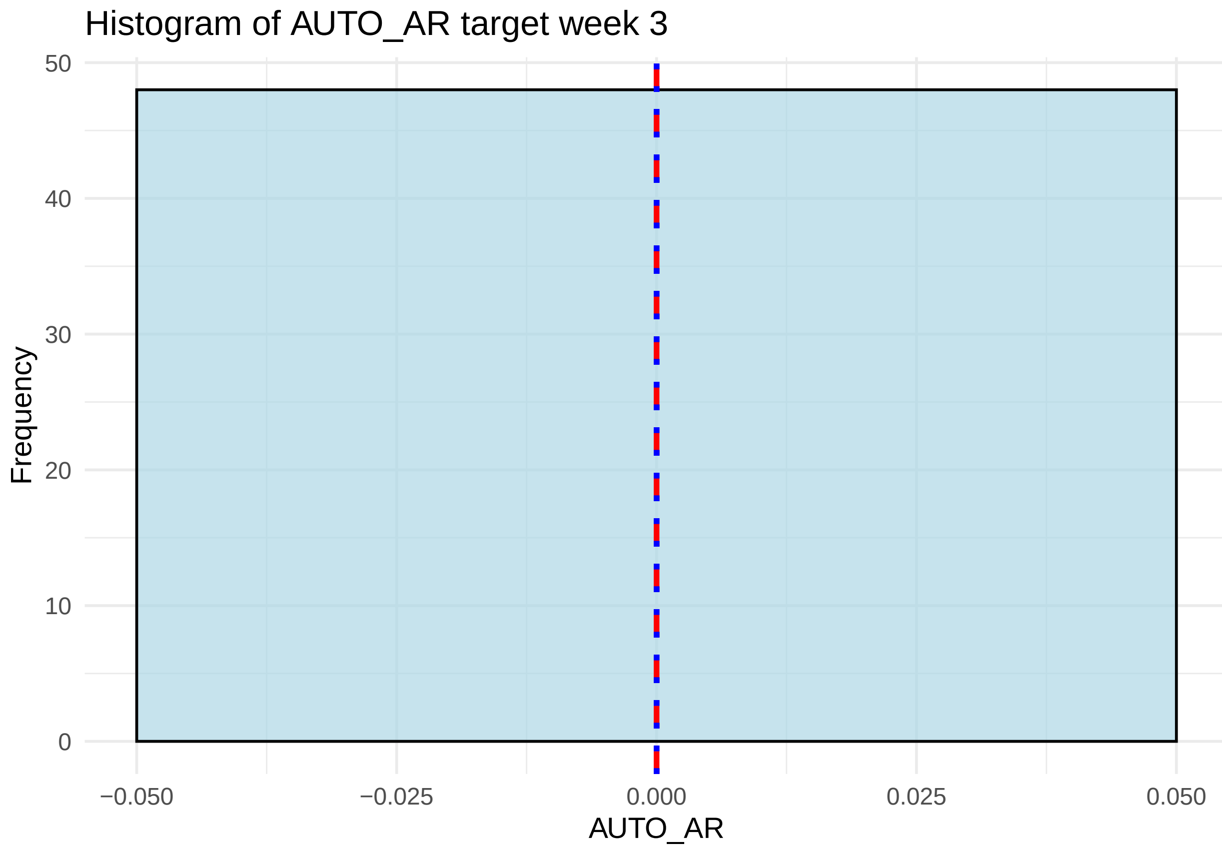

Histogram of ES64_EPI_LB target week 2
Frequency
6
4
2
0
−20
−10
0
10
20
ES64_EPI_LB



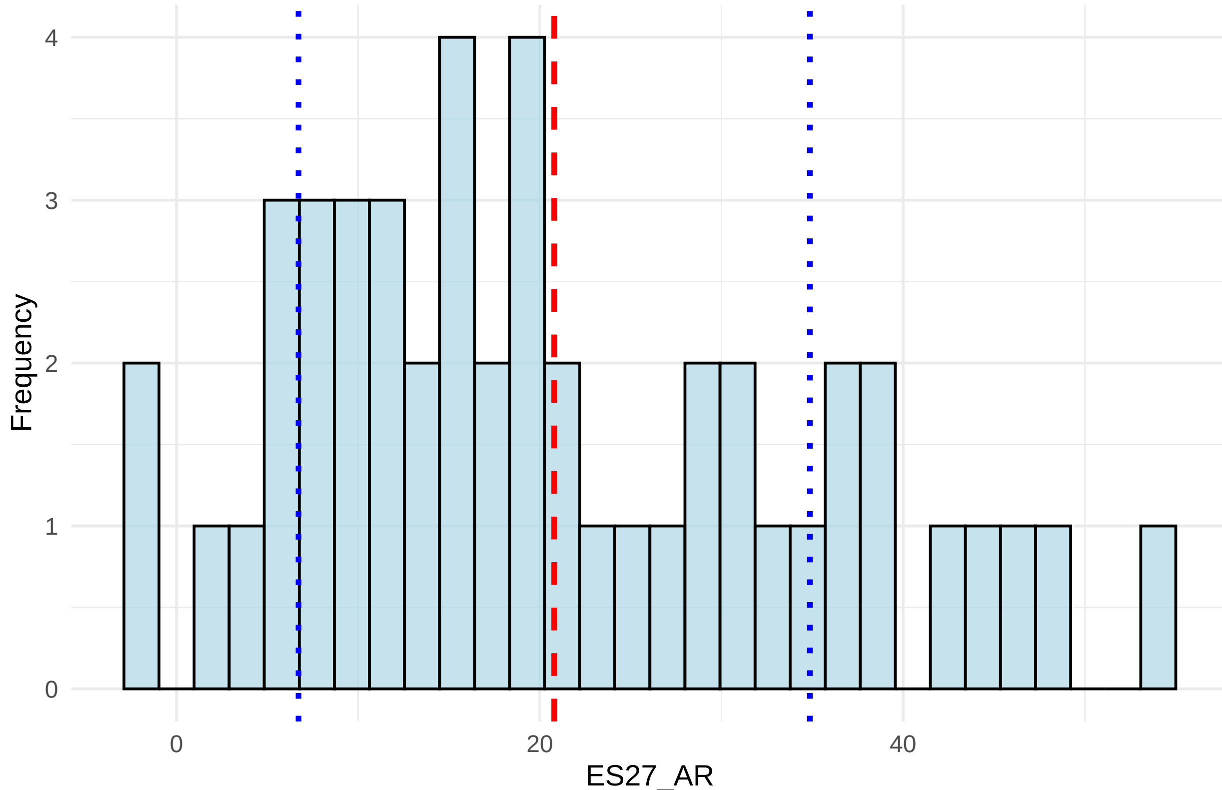
Histogram of AUTO_AVG_LB target week 2
Frequency
0
2
4
6
−40
−30
−20
−10
0
AUTO_AVG_LB

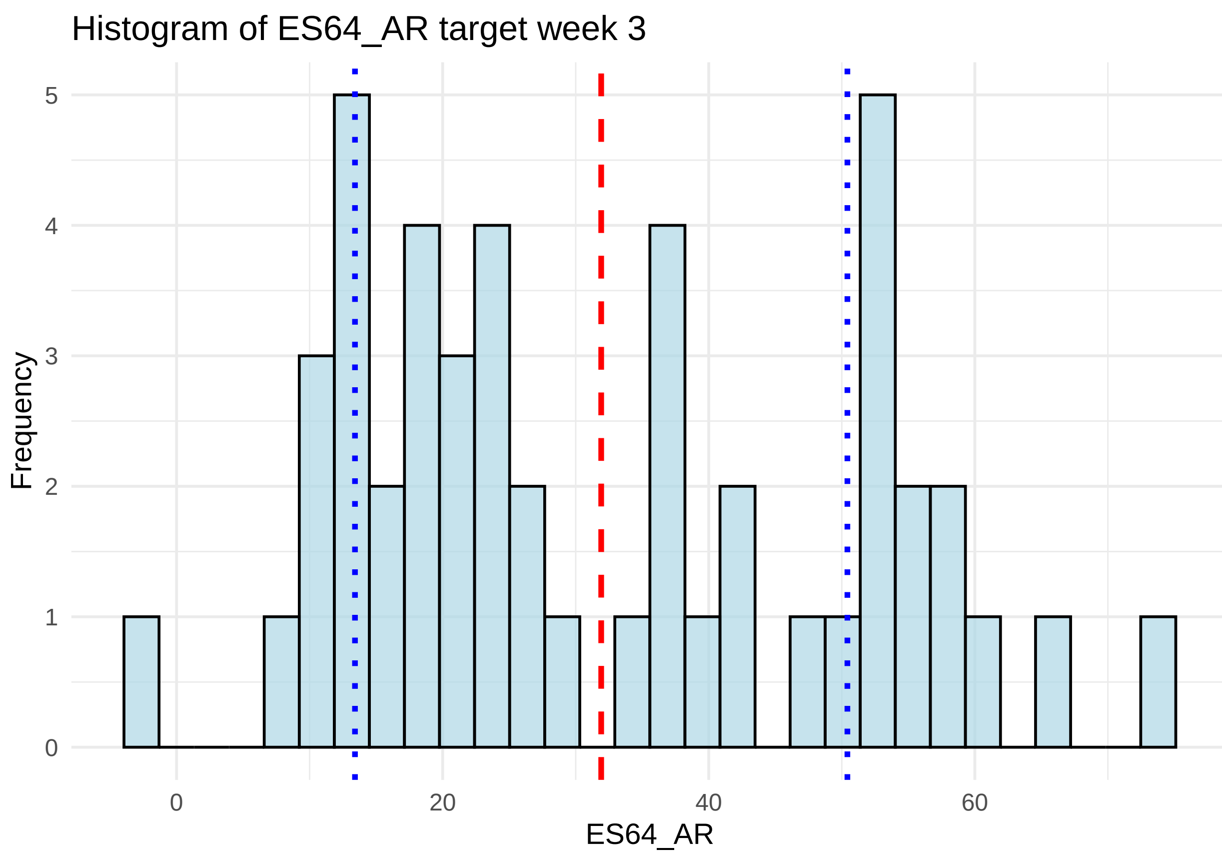

Histogram of ES27_AVG_LB target week 2
Frequency
5
4
3
2
1
0
−40
−30
−20
−10
0
ES27_AVG_LB

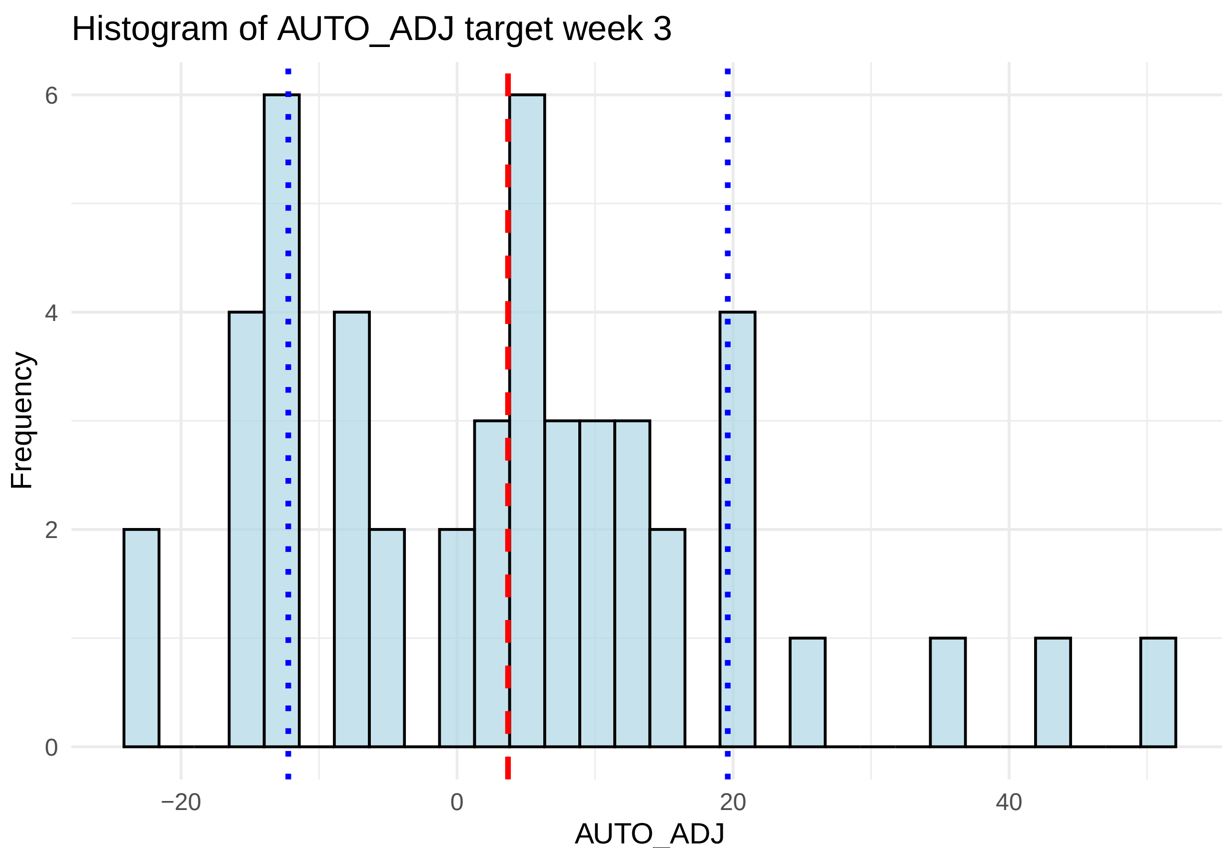

Histogram of ES64_AVG_LB target week 2
Frequency
5
4
3
2
1
0
−30
−20
−10
0
ES64_AVG_LB

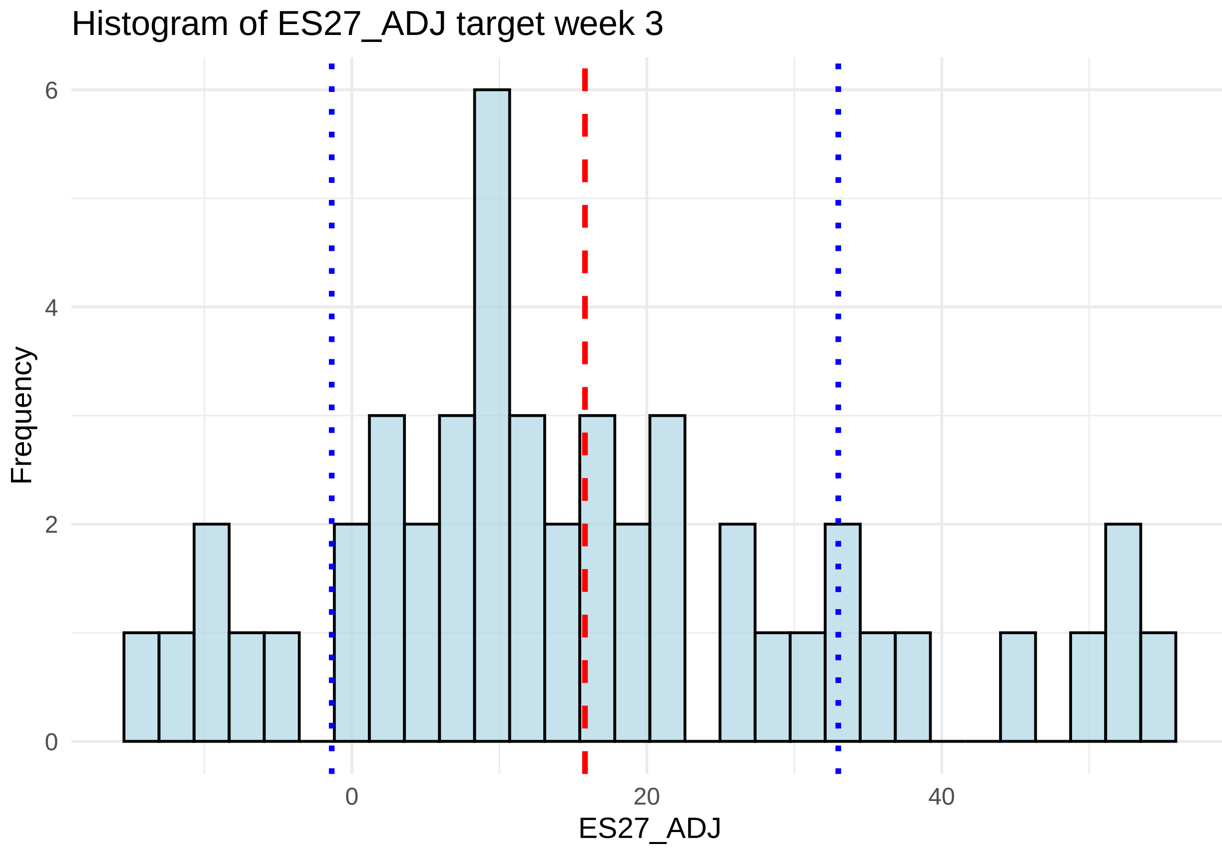

Histogram of AUTO_AR target week 3
Frequency
50
40
30
20
10
0
−0.050
−0.025
0.000
0.025
0.050
AUTO_AR

Histogram of ES27_AR target week 3

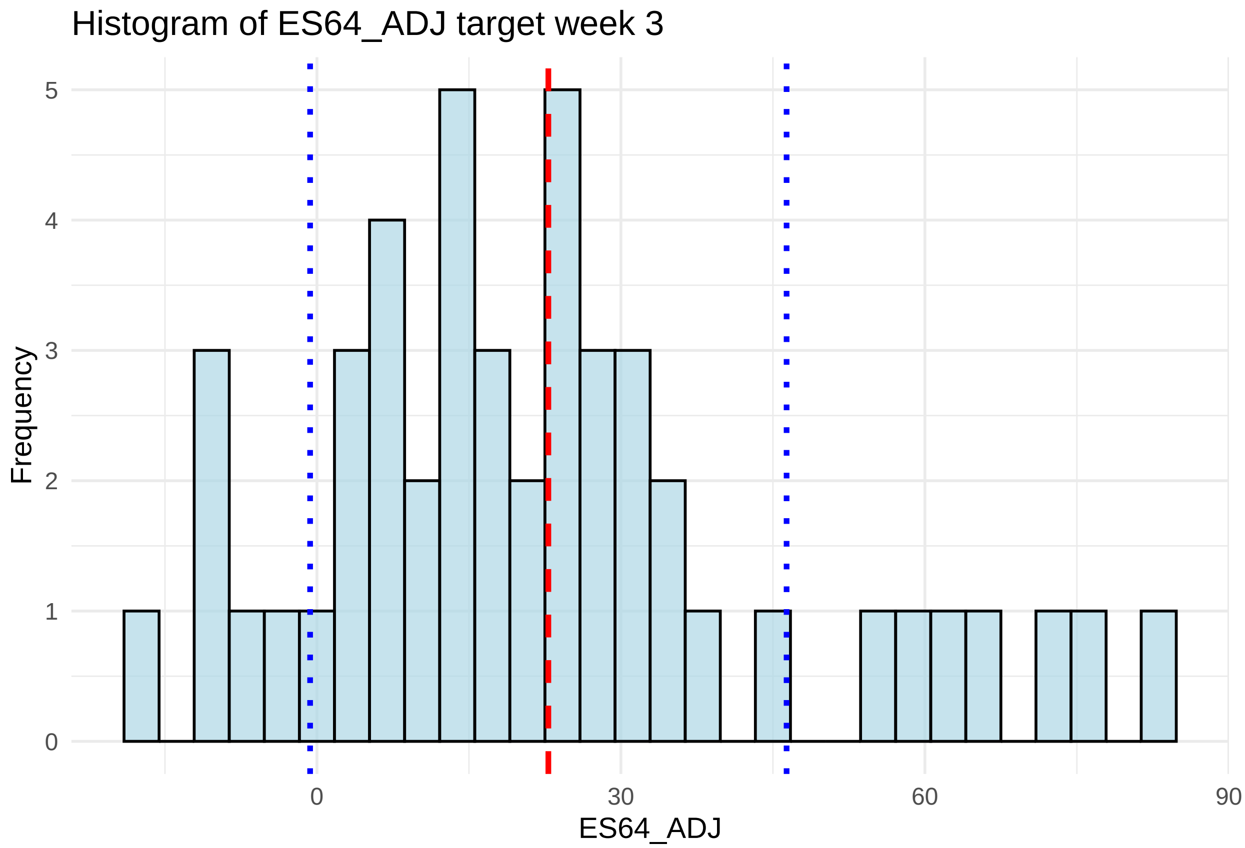


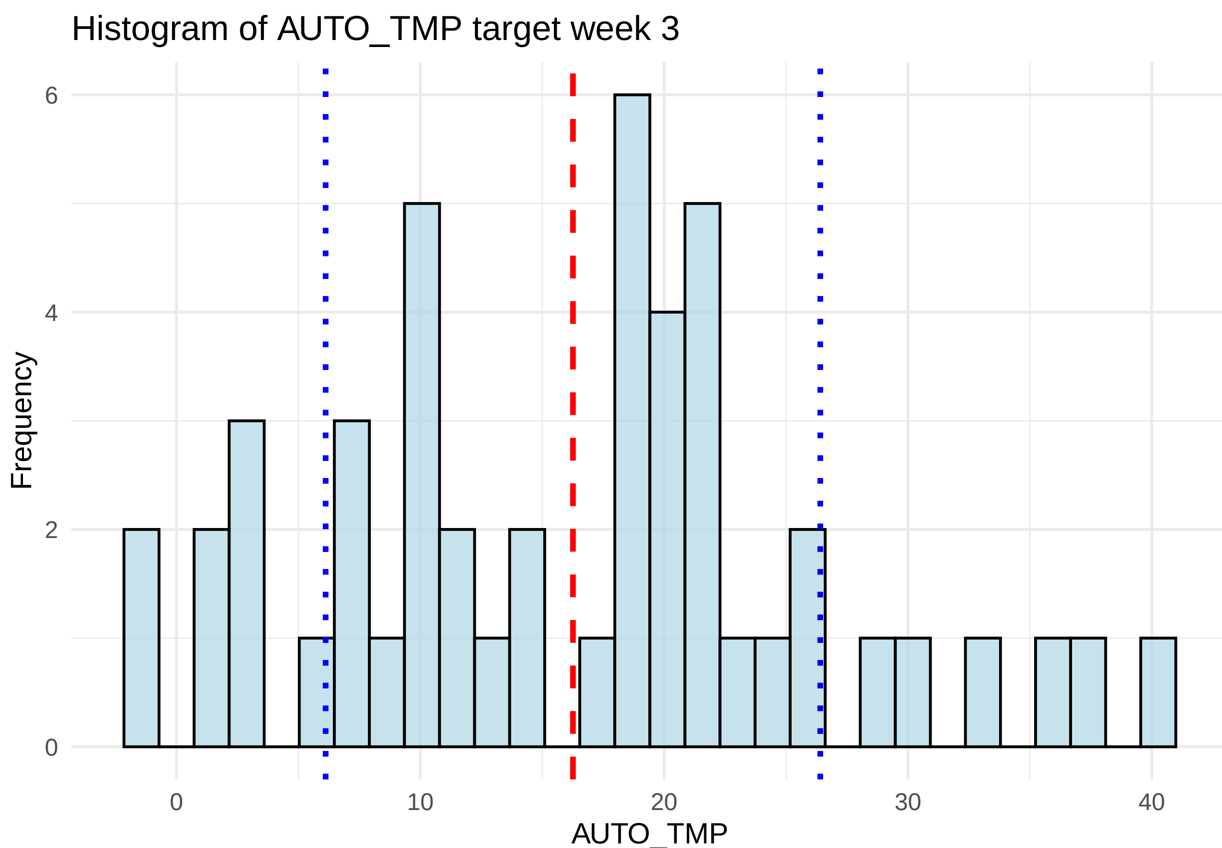

Histogram of ES64_AR target week 3
Frequency
ES64_AR
0
1
2
3
4
5
0
20
40
60


Histogram of AUTO_ADJ target week 3
Frequency
0
2
4
6
−20
0
20
40
AUTO_ADJ

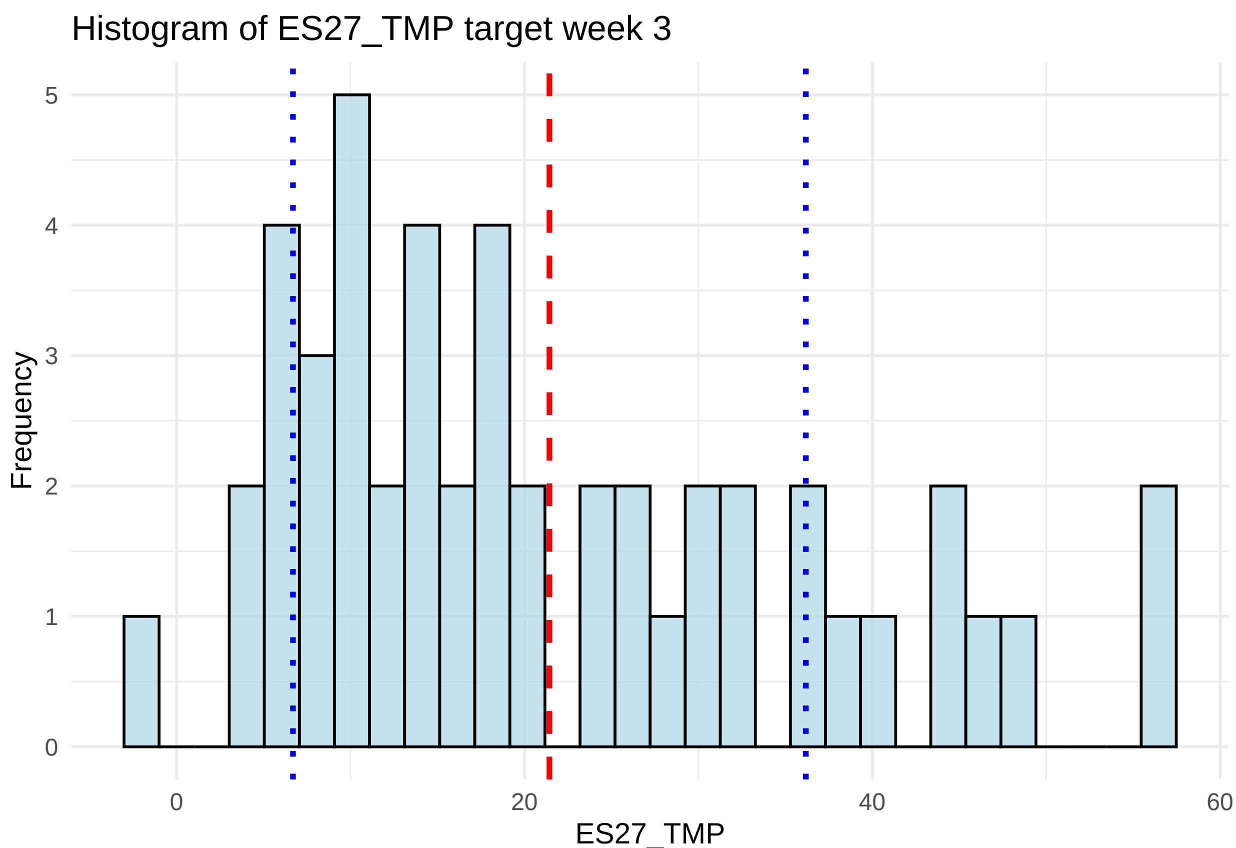


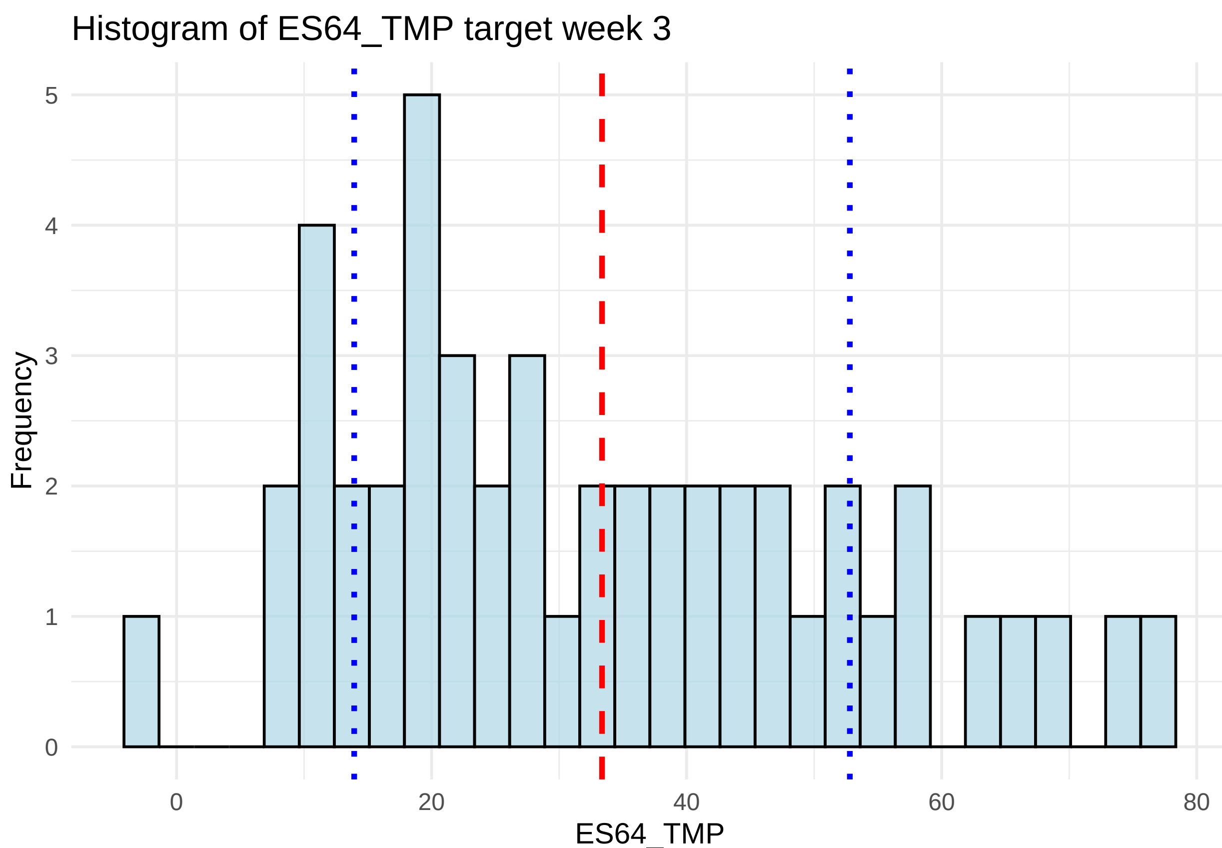
Histogram of ES27_ADJ target week 3
Frequency
ES27_ADJ
0
2
4
6
0
20
40

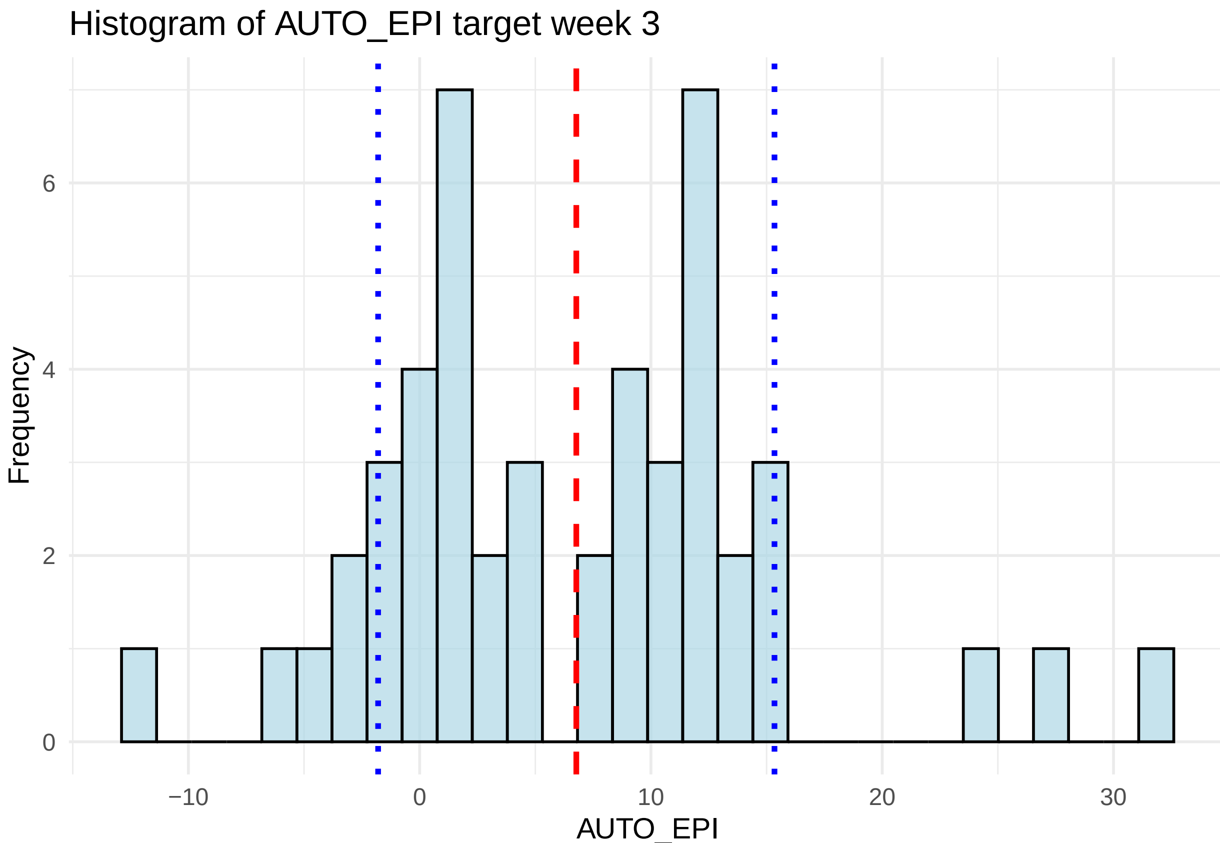

Histogram of ES64_ADJ target week 3
Frequency
5
4
3
2
1
0
0
30
60
90
ES64_ADJ



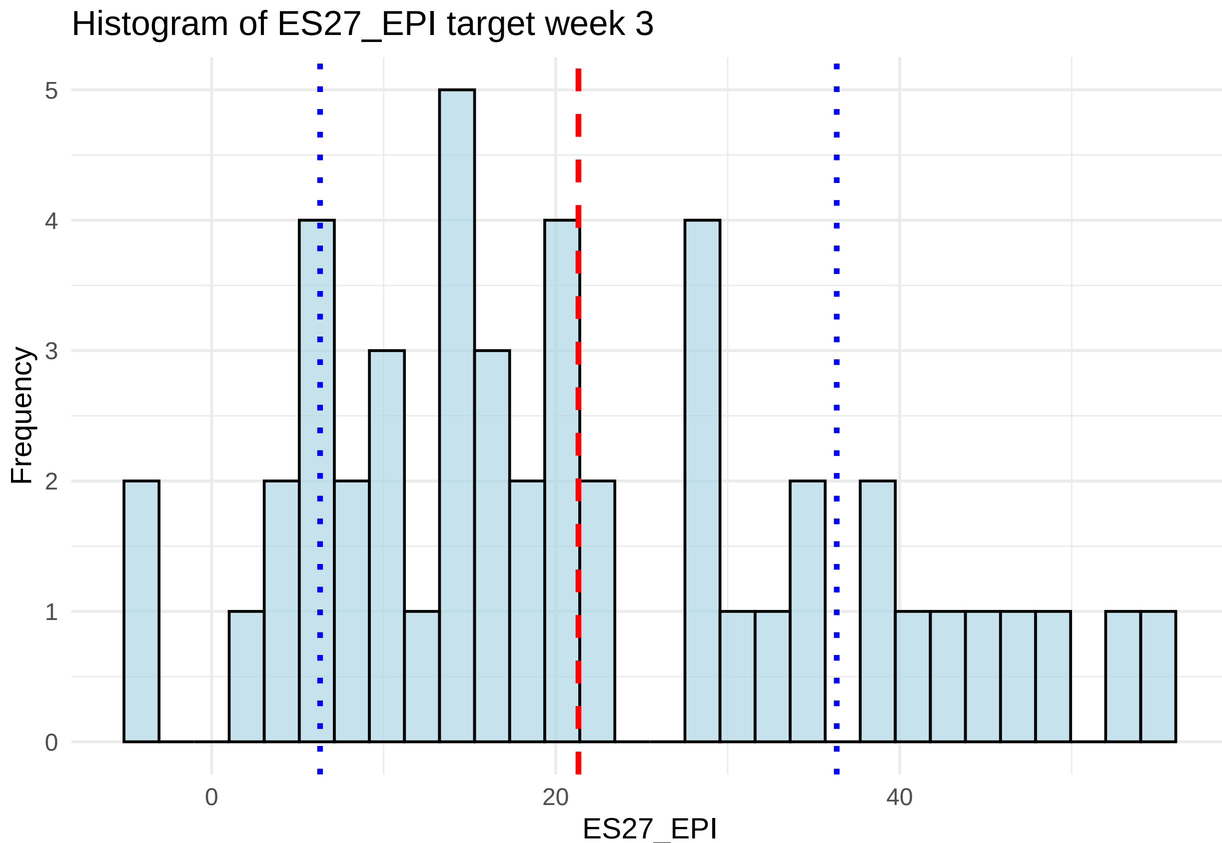
Histogram of AUTO_TMP target week 3
Frequency
0
2
4
6
0
10
20
30
40
AUTO_TMP

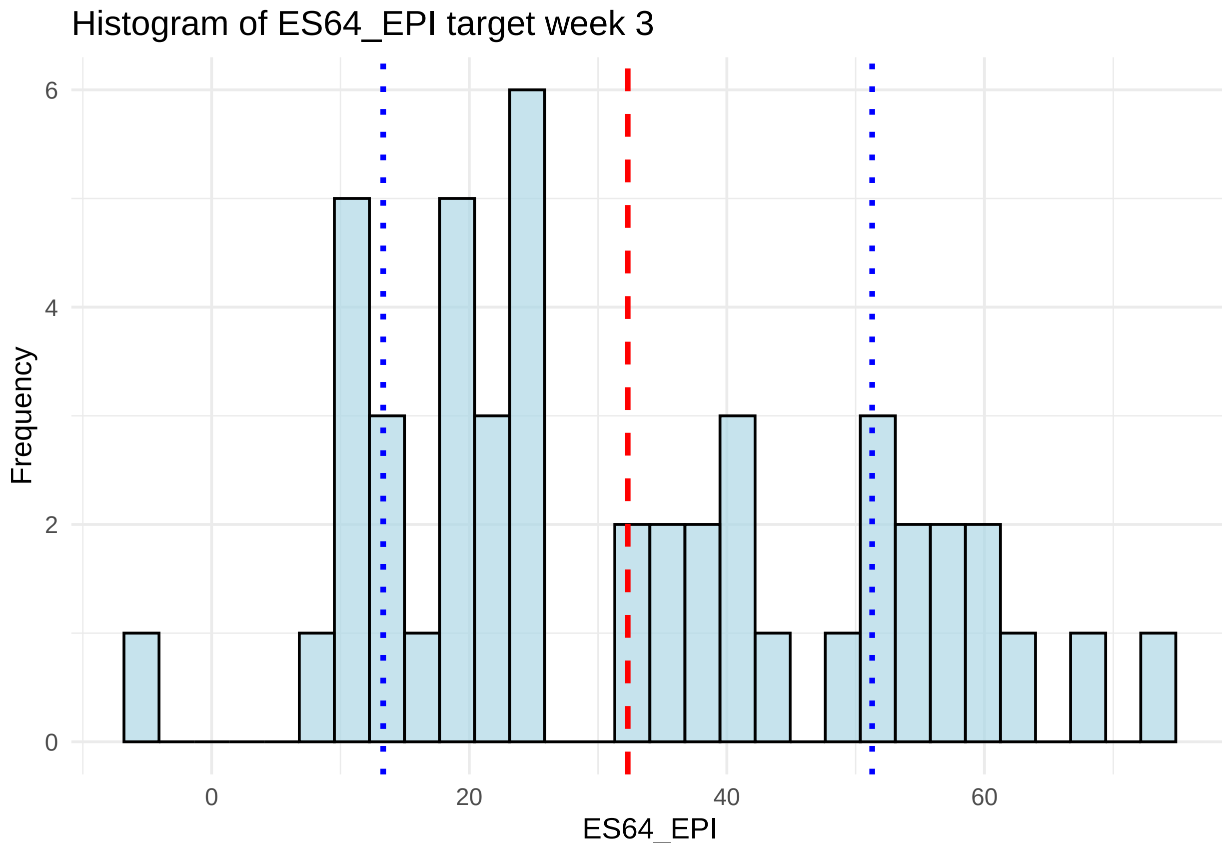

Histogram of ES27_TMP target week 3
Frequency
ES27_TMP
0
1
2
3
4
5
0
20
40
60


Histogram of ES64_TMP target week 3
Frequency
ES64_TMP
0
1
2
3
4
5
0
20
40
60
80

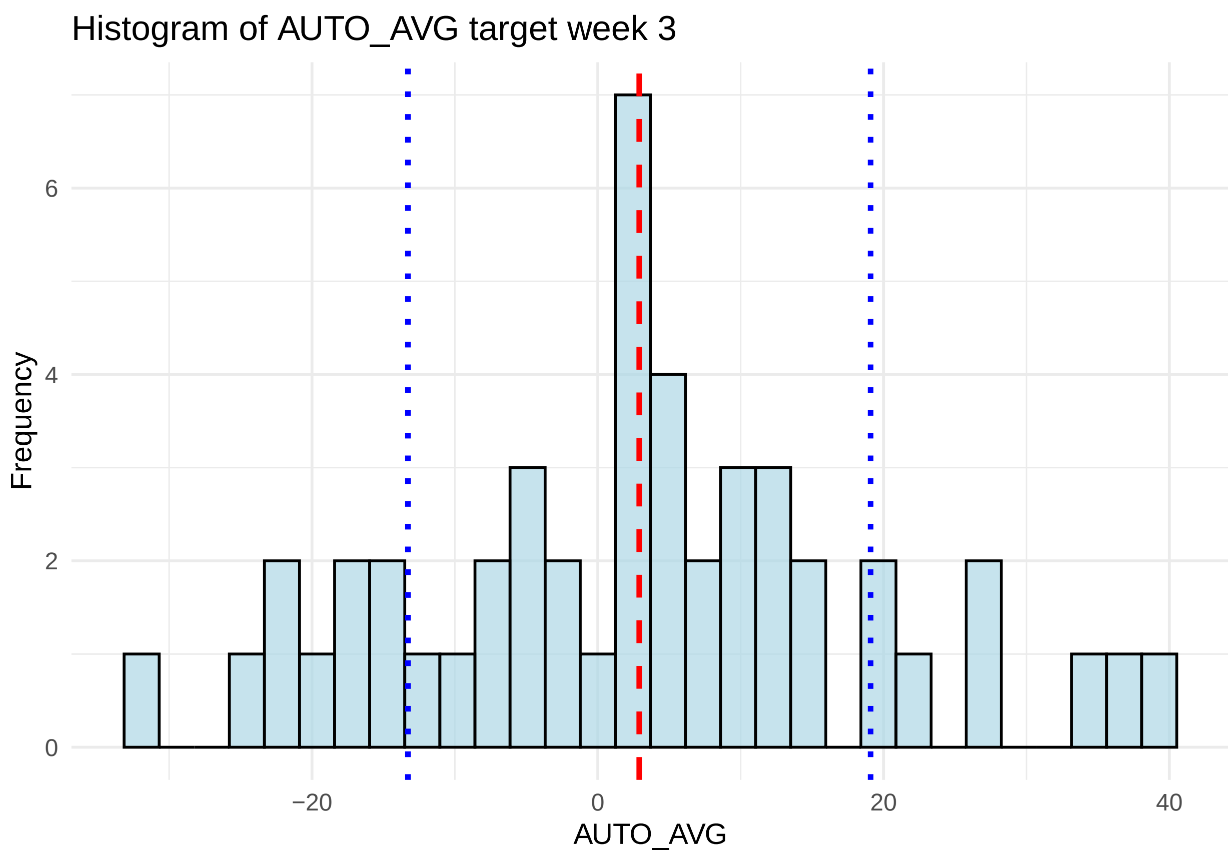


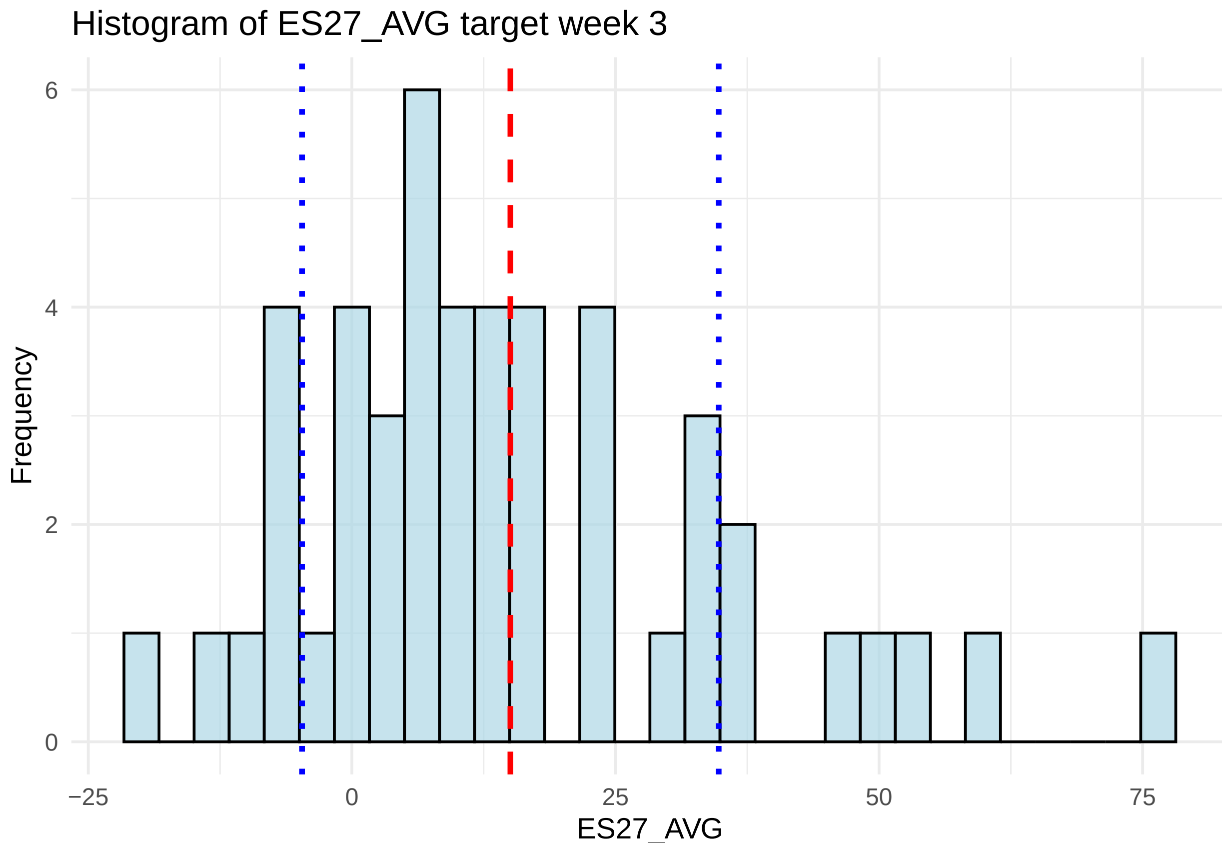
Histogram of AUTO_EPI target week 3
Frequency
AUTO_EPI
0
2
4
6
−10
0
10
20
30


Histogram of ES27_EPI target week 3
Frequency
0
1
2
3
4
5
0
20
40
ES27_EPI

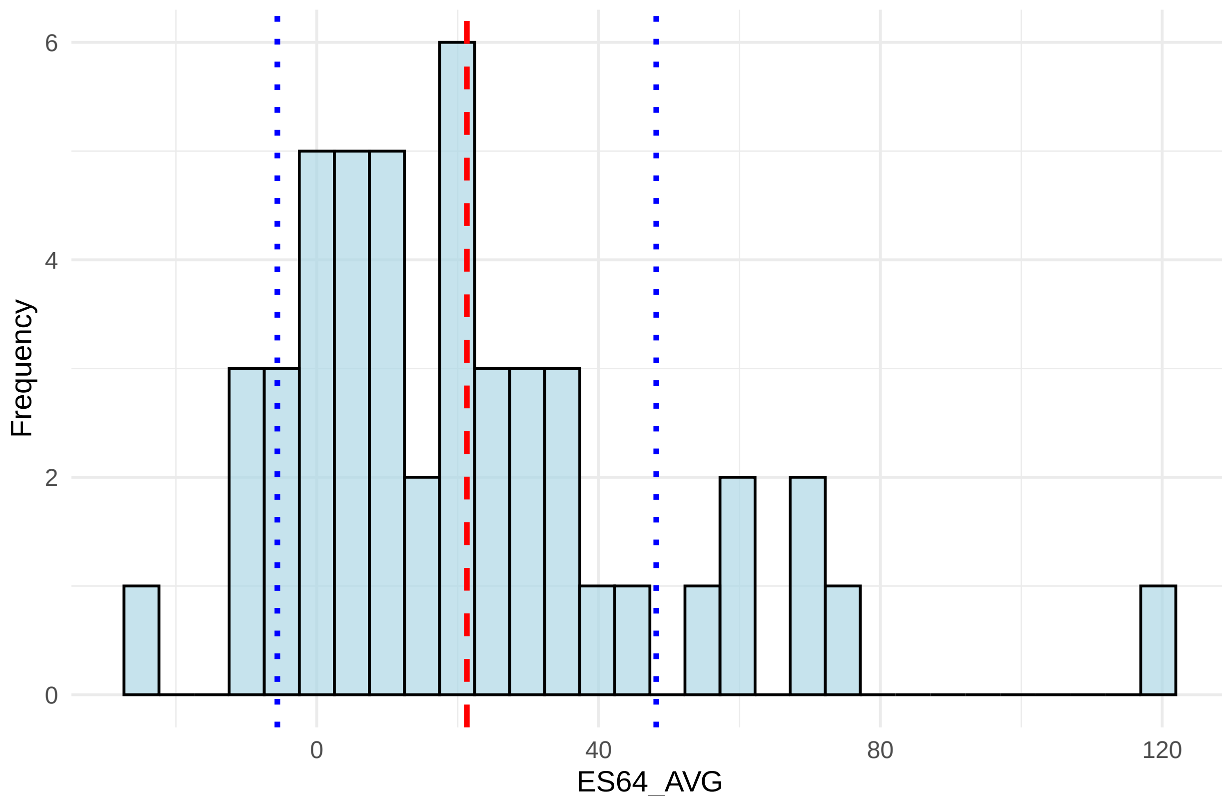


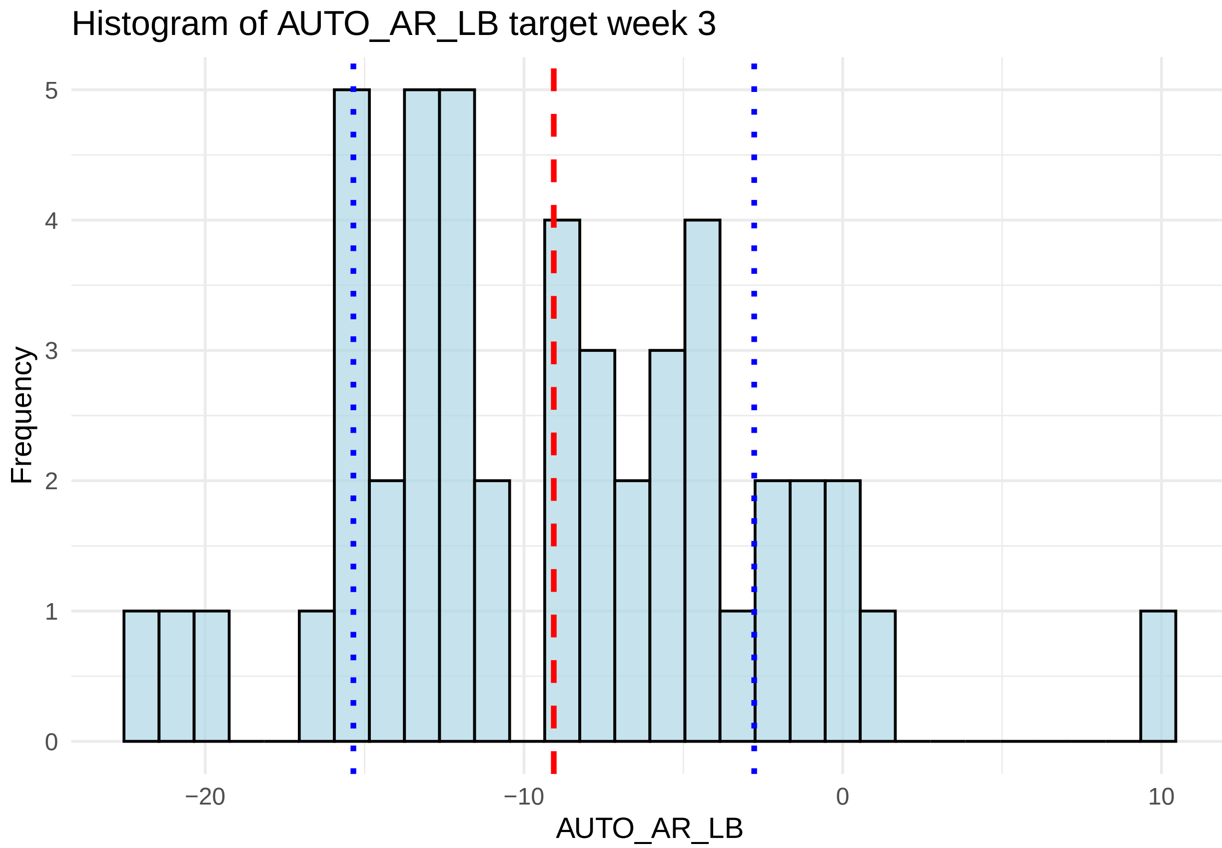
Histogram of ES64_EPI target week 3
Frequency
ES64_EPI
0
2
4
6
0
20
40
60

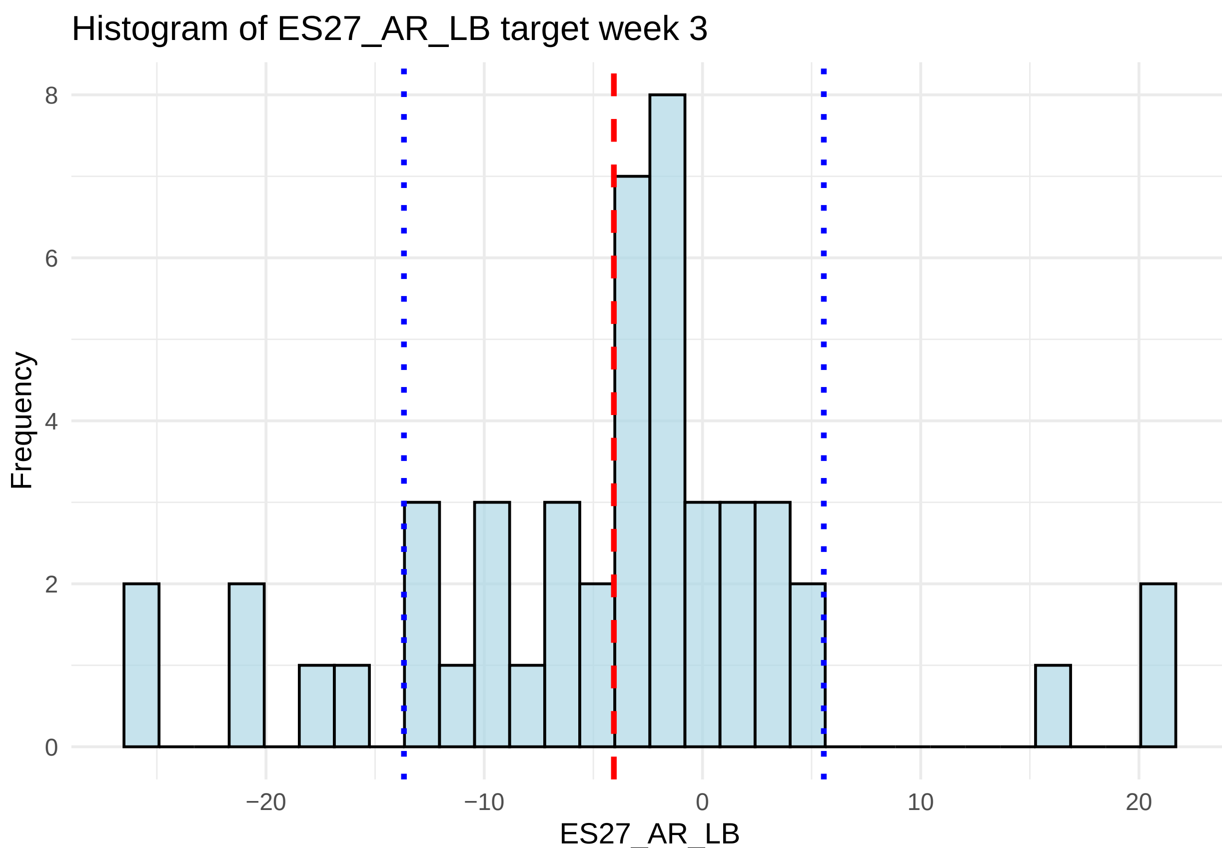

Histogram of AUTO_AVG target week 3
Frequency
0
2
4
6
−20
0
20
40
AUTO_AVG

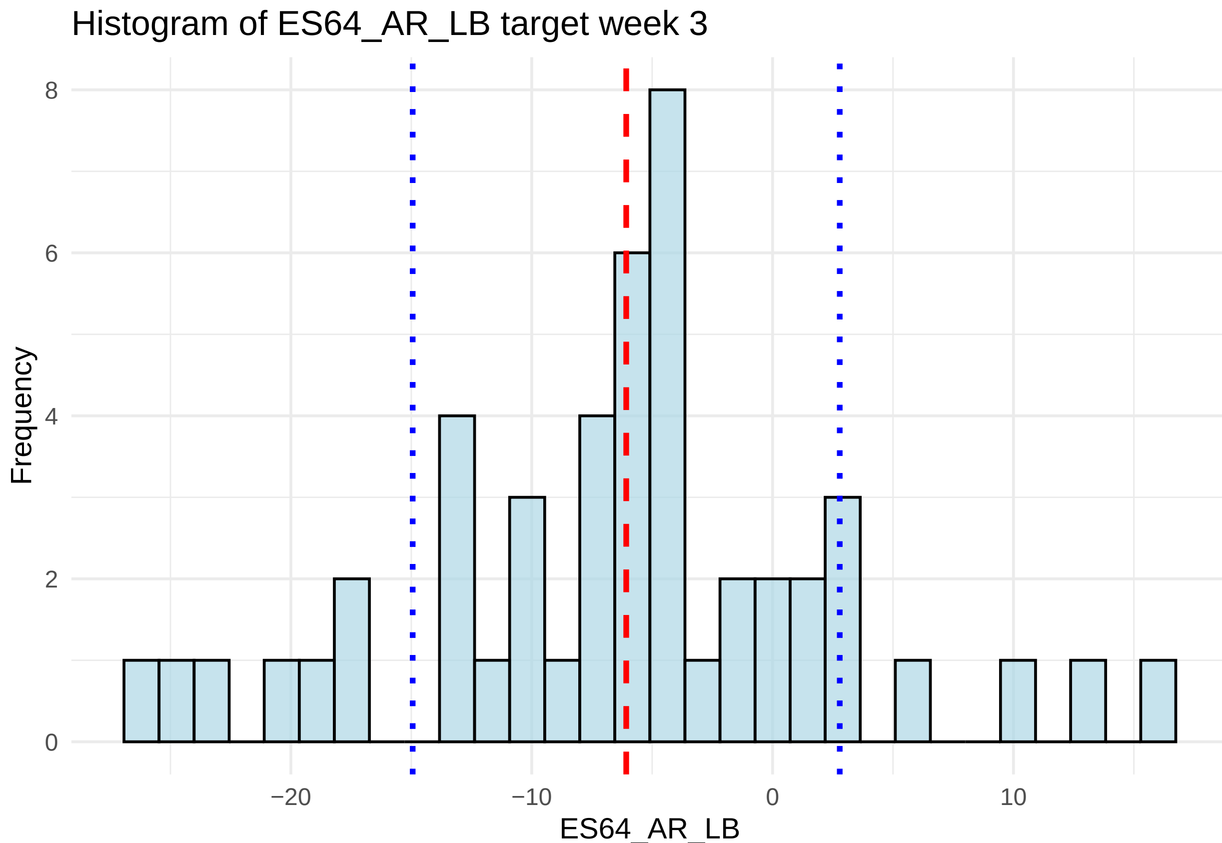

Histogram of ES27_AVG target week 3
Frequency
ES27_AVG
6
4
2
0
−25
0
25
50
75

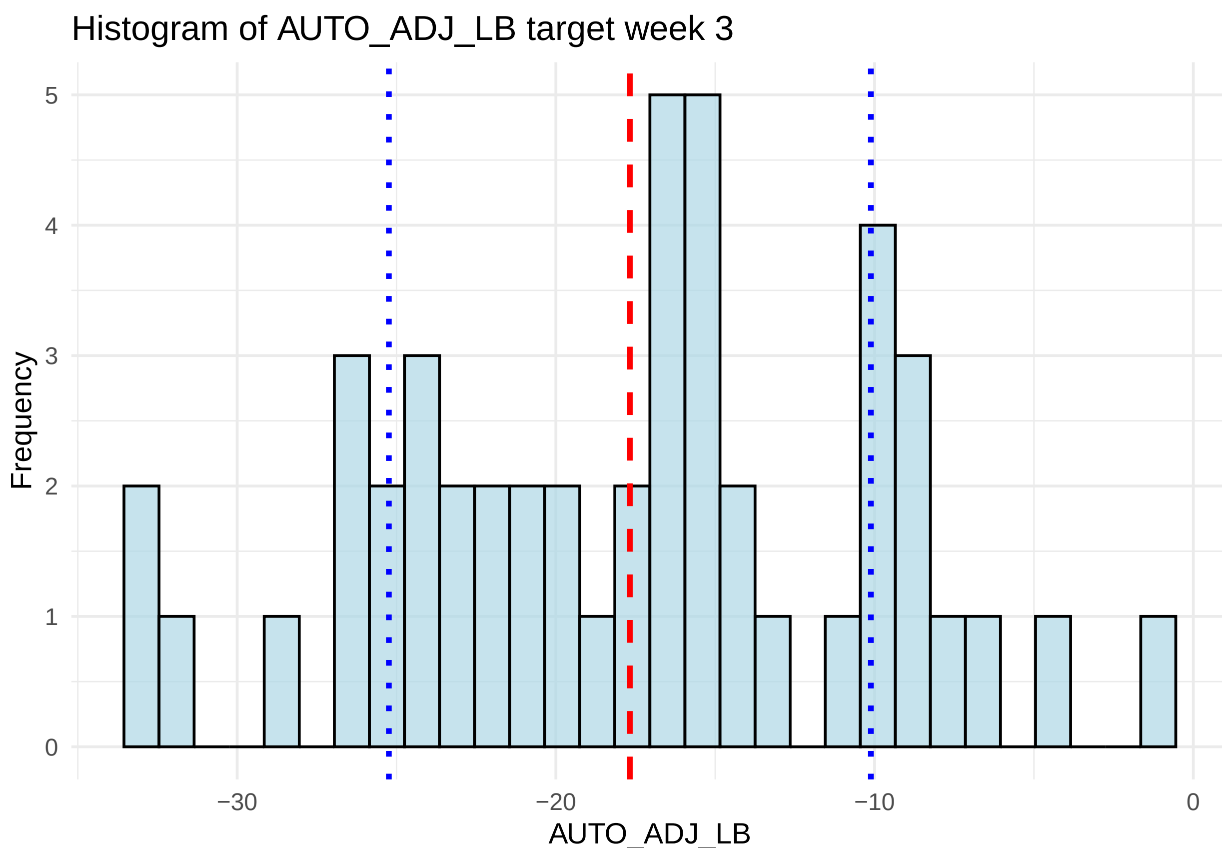
Histogram of ES64_AVG target week 3



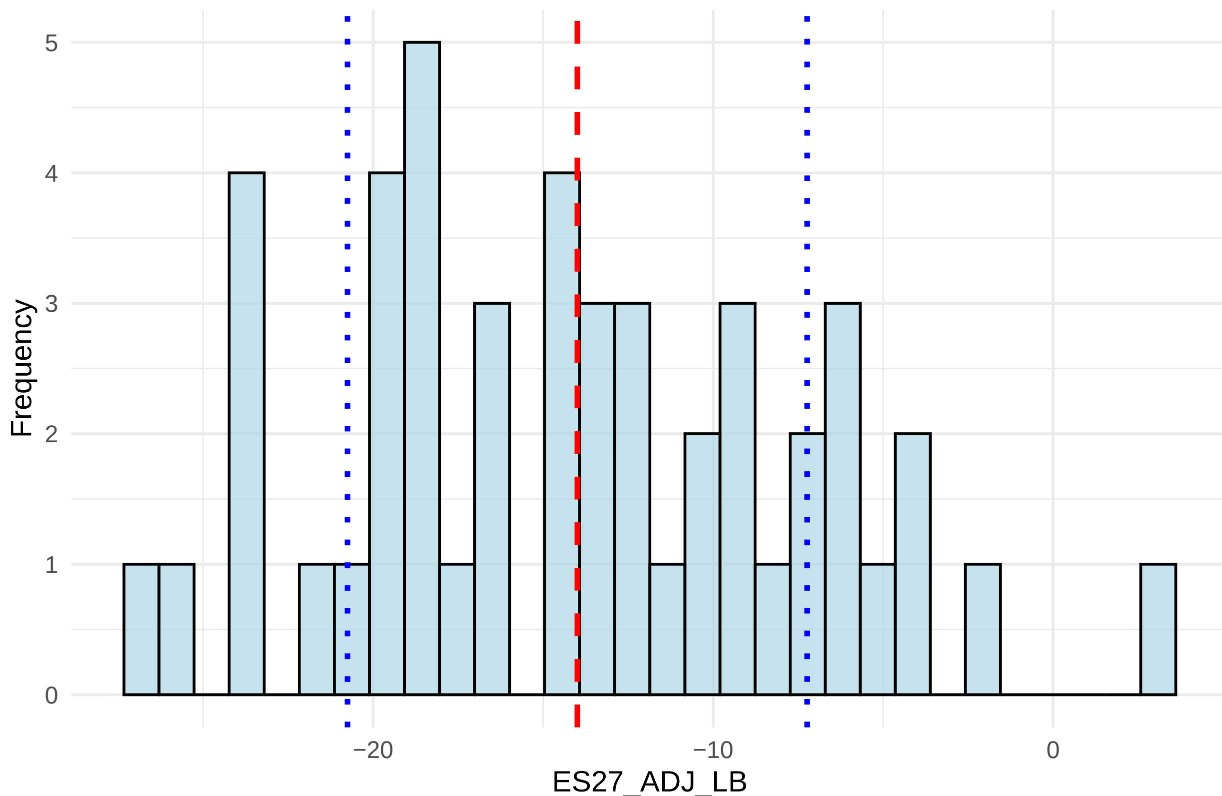

Histogram of AUTO_AR_LB target week 3
Frequency
0
1
2
3
4
5
−20
−10
0
10
AUTO_AR_LB

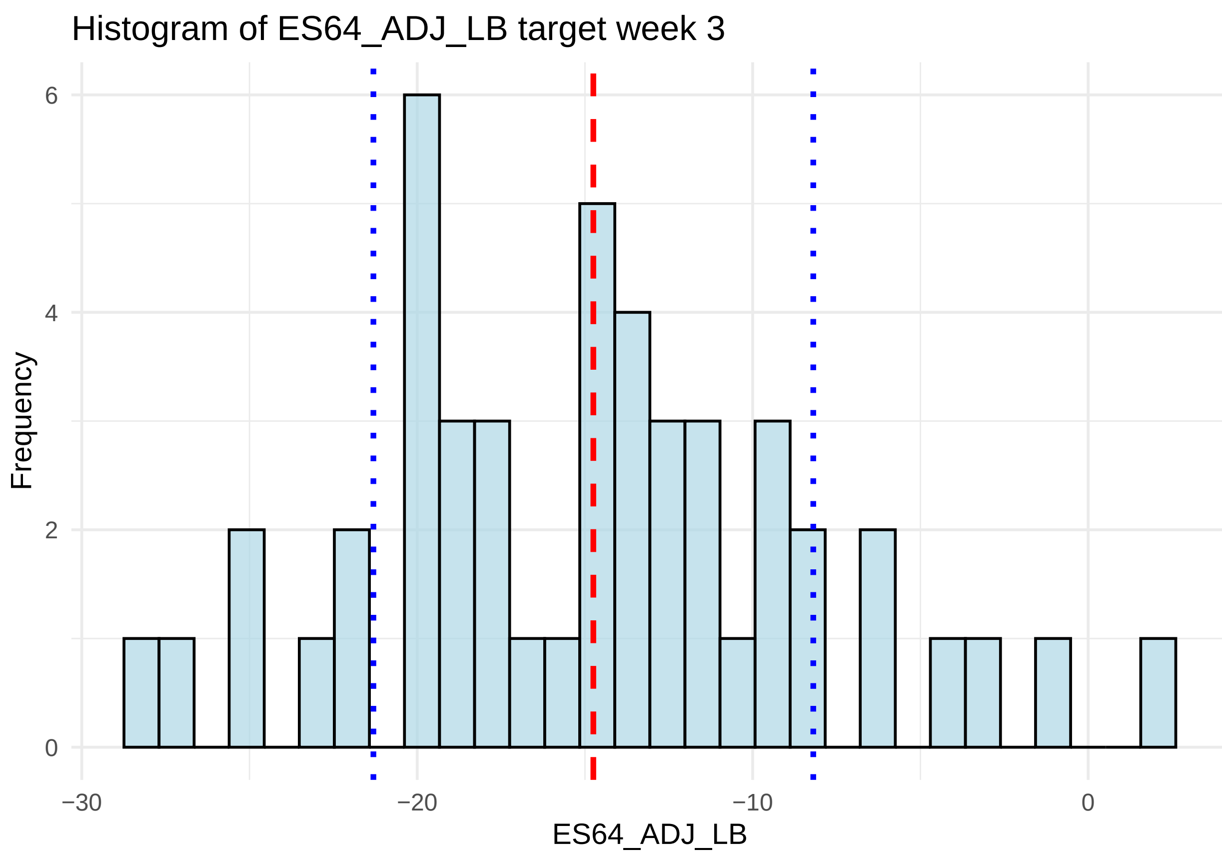

Histogram of ES27_AR_LB target week 3
Frequency
ES27_AR_LB
0
2
4
6
8
−20
−10
0
10
20

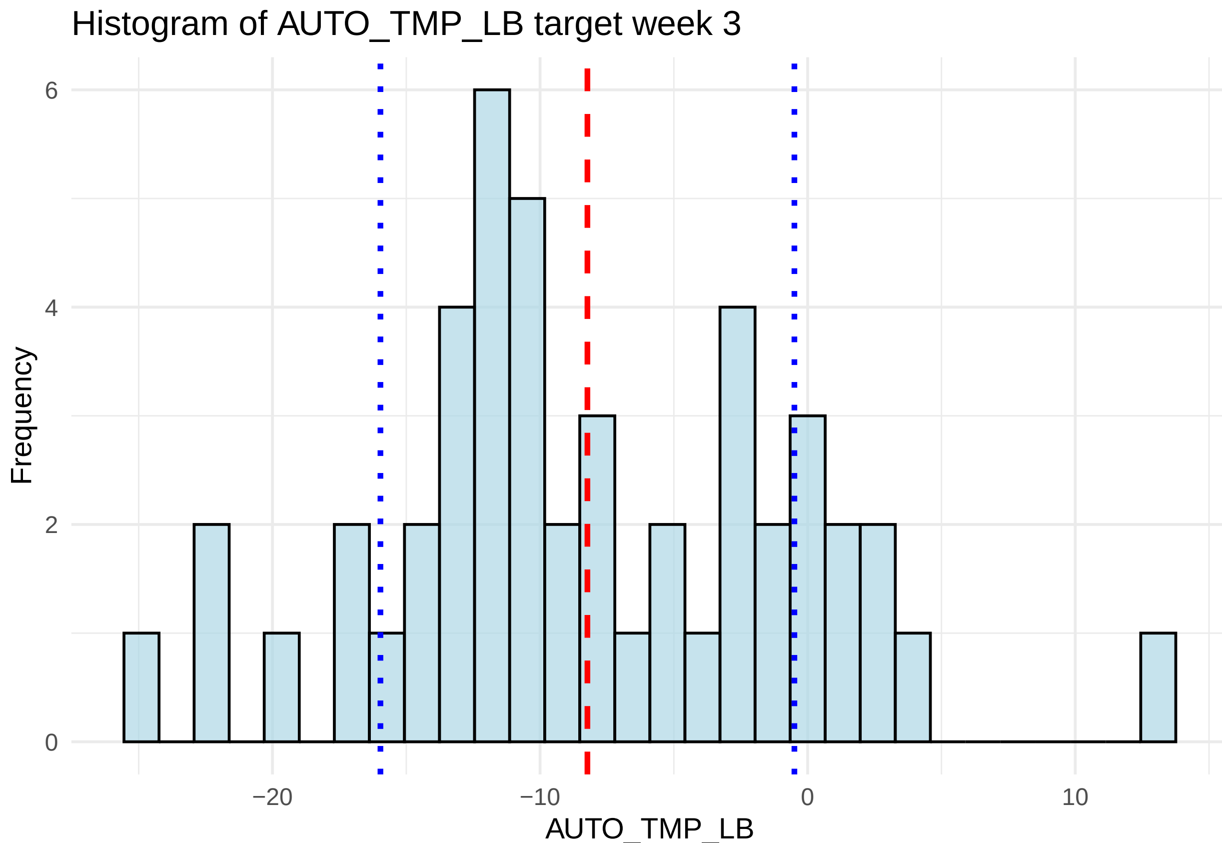

Histogram of ES64_AR_LB target week 3
Frequency
0
2
4
6
8
−20
−10
0
10
ES64_AR_LB

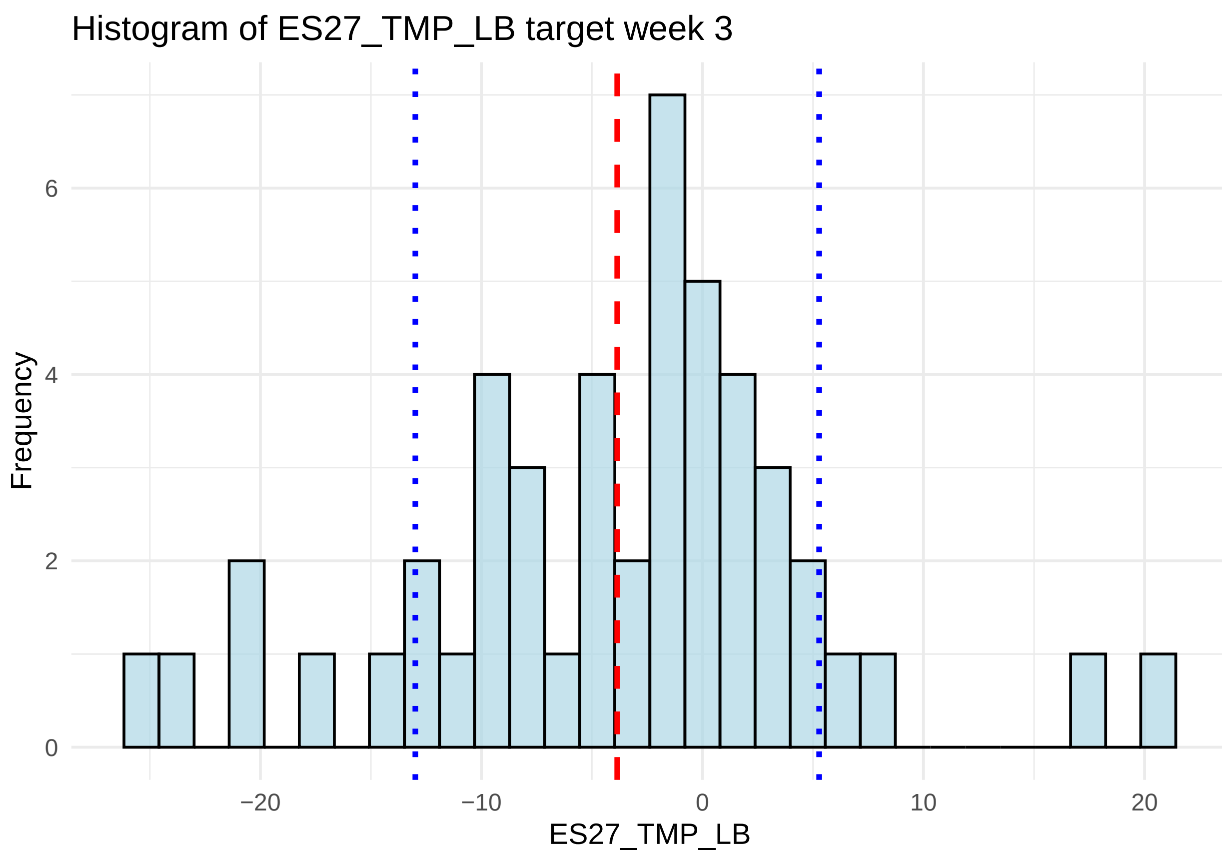

Histogram of AUTO_ADJ_LB target week 3
Frequency
0
1
2
3
4
5
−30
−20
−10
0
AUTO_ADJ_LB

Histogram of ES27_ADJ_LB target week 3

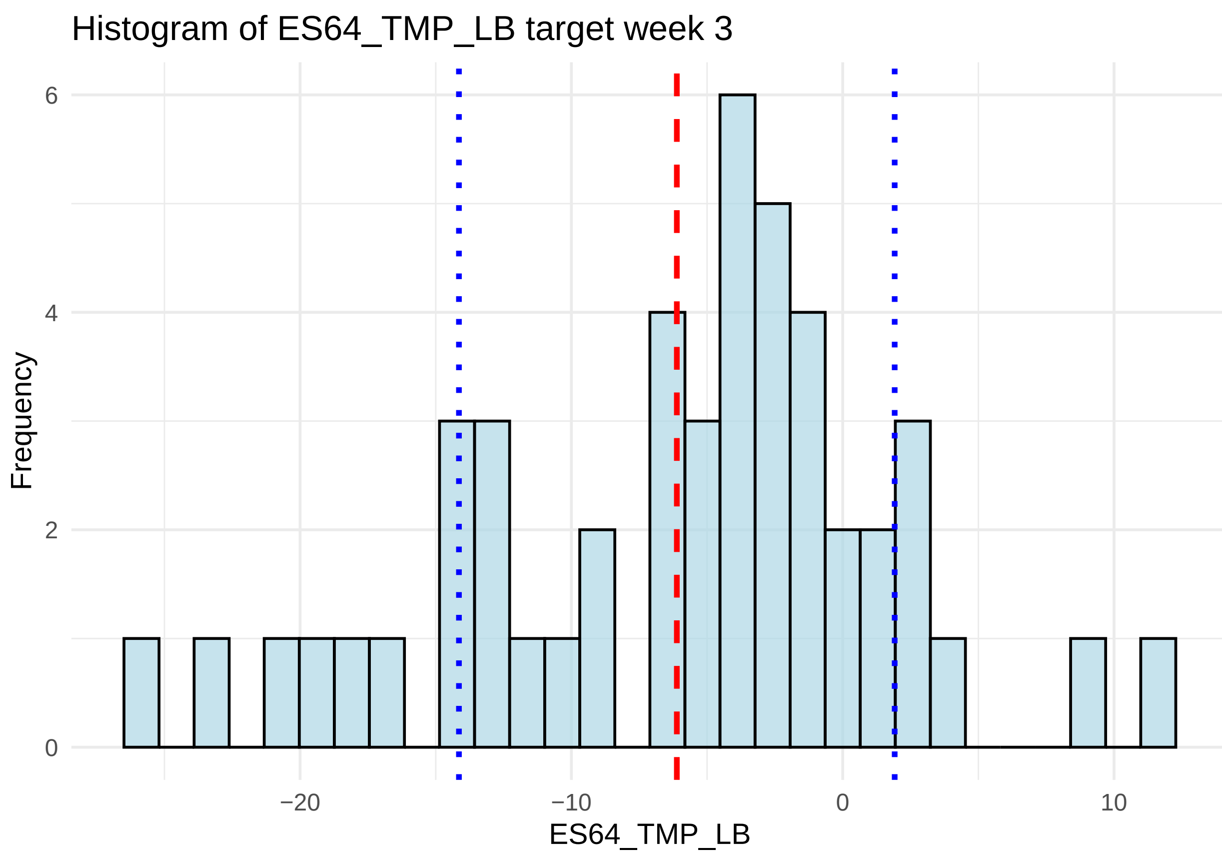



Histogram of ES64_ADJ_LB target week 3
Frequency
0
2
4
6
−30
−20
−10
0
ES64_ADJ_LB

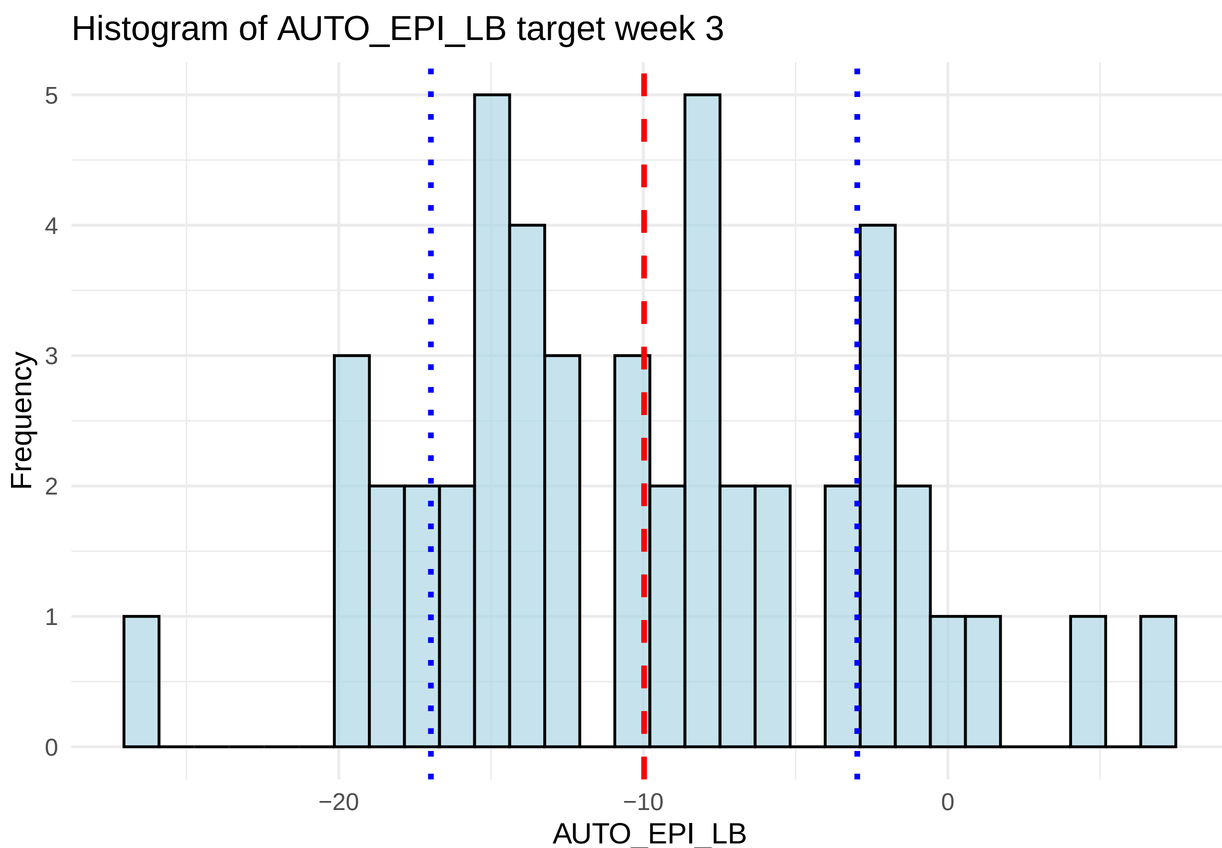

Histogram of AUTO_TMP_LB target week 3
Frequency
0
2
4
6
−20
−10
0
10
AUTO_TMP_LB

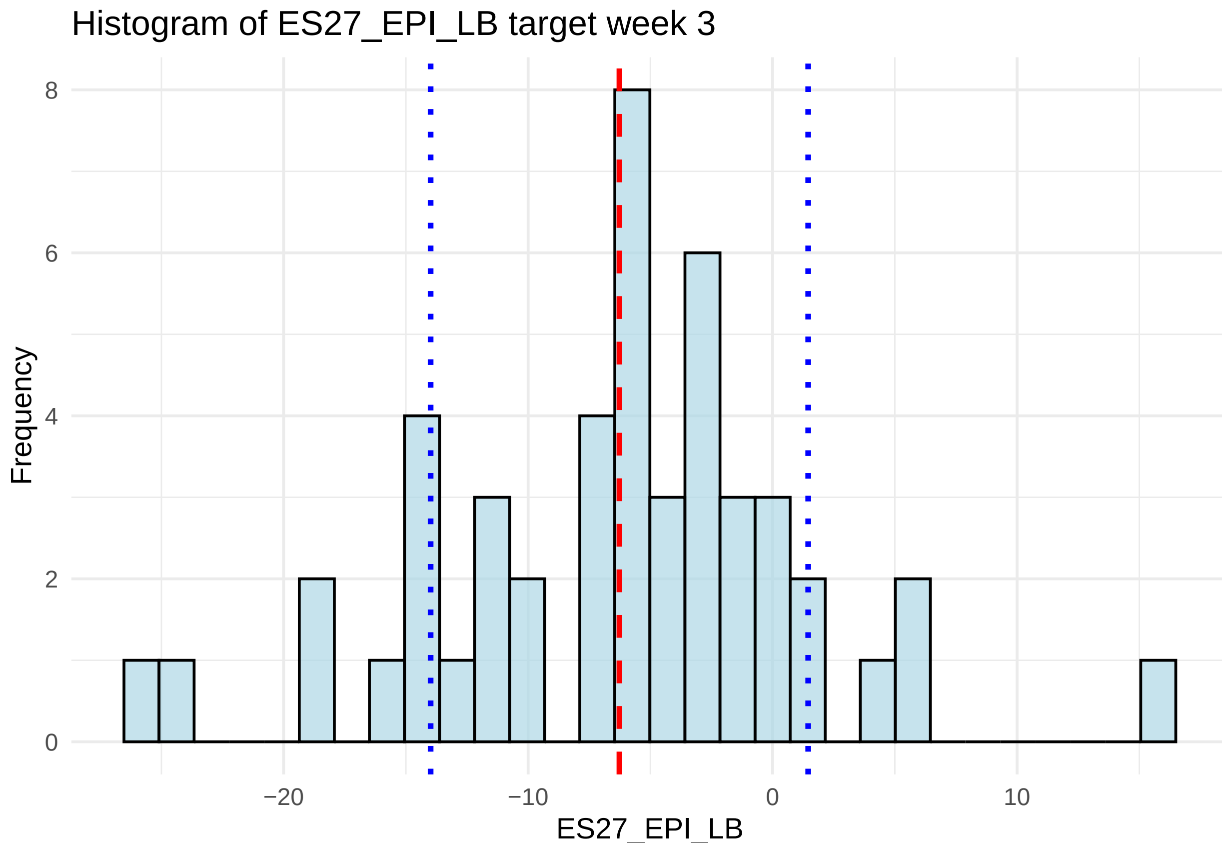


Histogram of ES27_TMP_LB target week 3
Frequency
0
2
4
6
−20
−10
0
10
20
ES27_TMP_LB

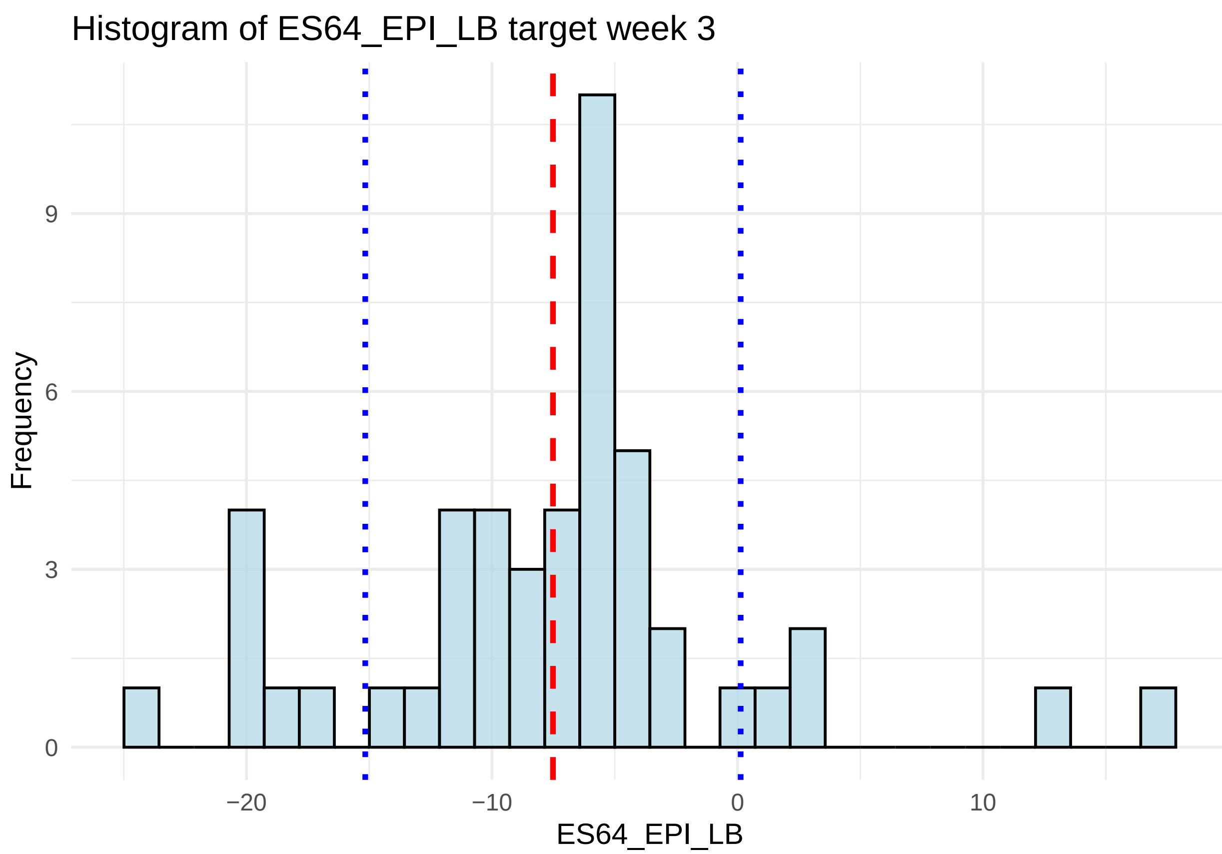


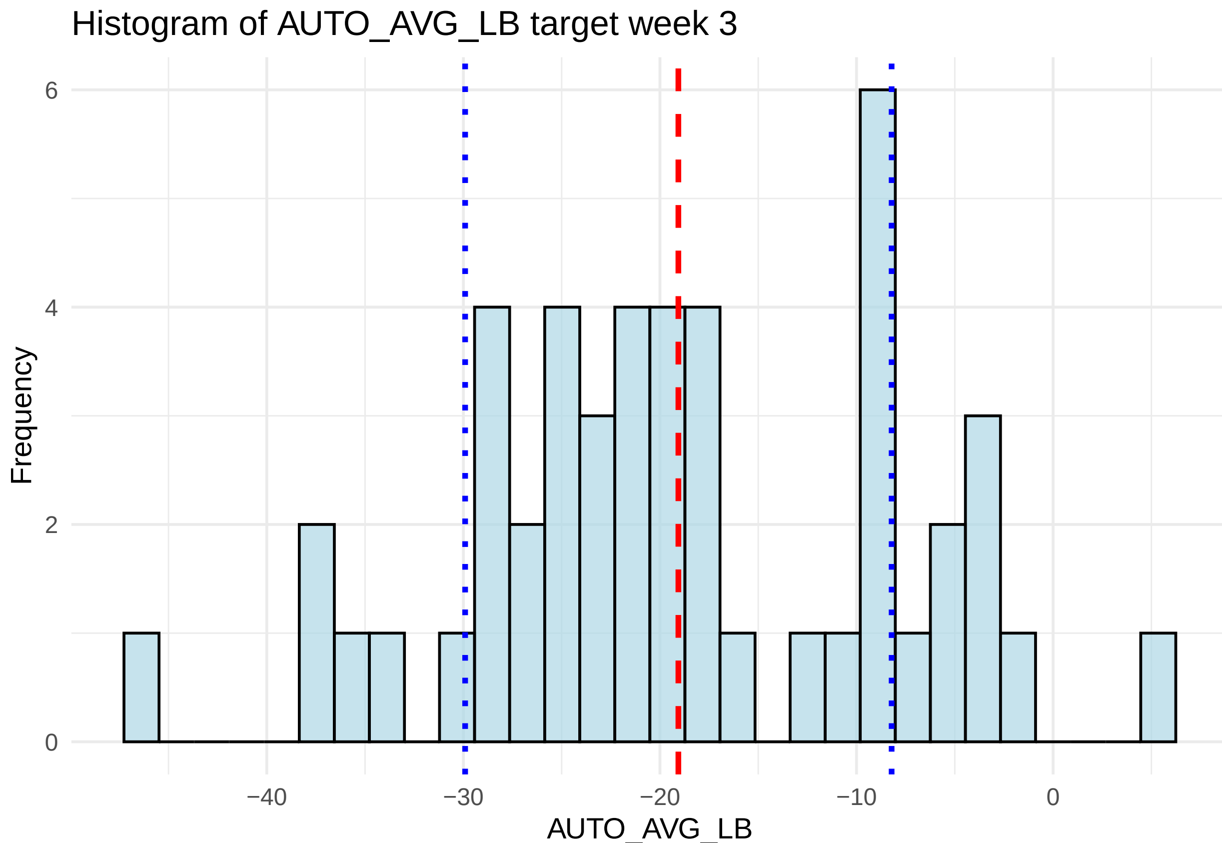
Histogram of ES64_TMP_LB target week 3
Frequency
6
4
2
0
−20
−10
0
10
ES64_TMP_LB

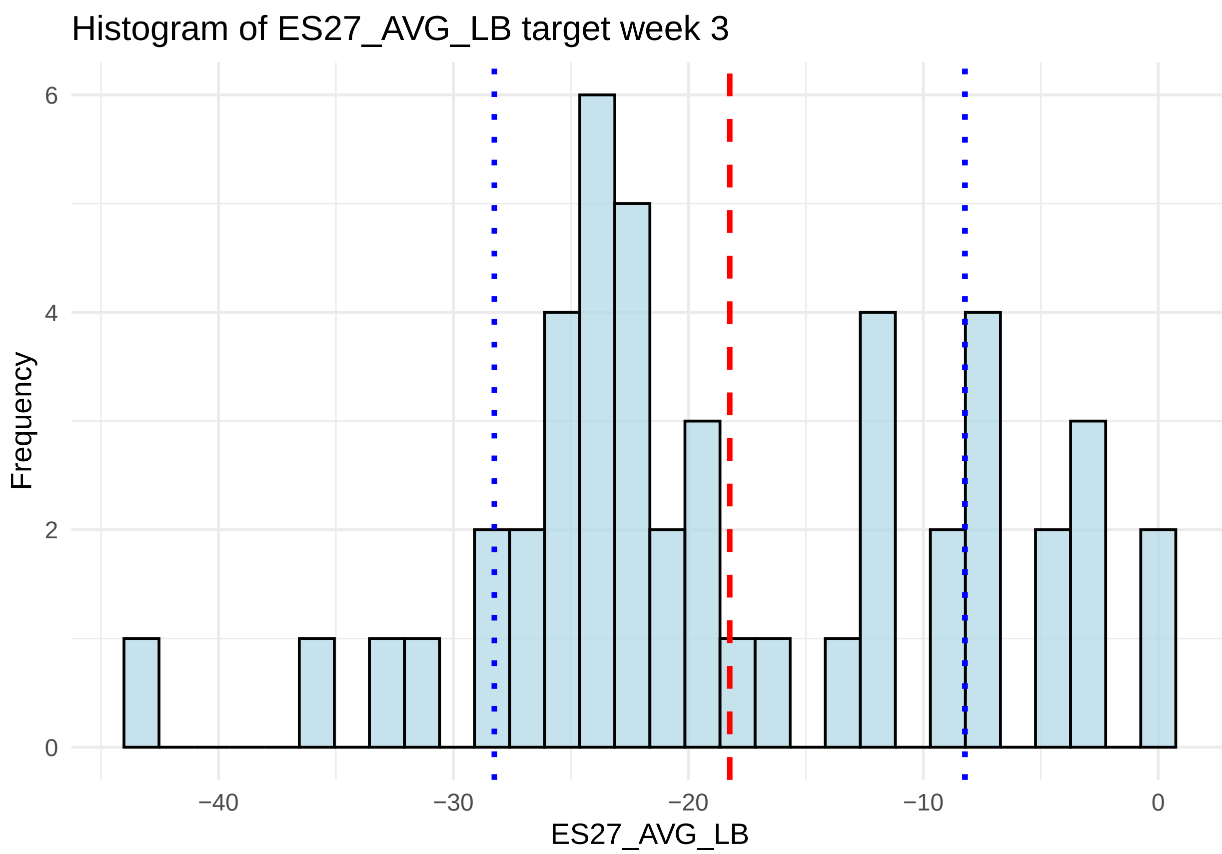

Histogram of AUTO_EPI_LB target week 3
Frequency
0
1
2
3
4
5
−20
−10
0
AUTO_EPI_LB

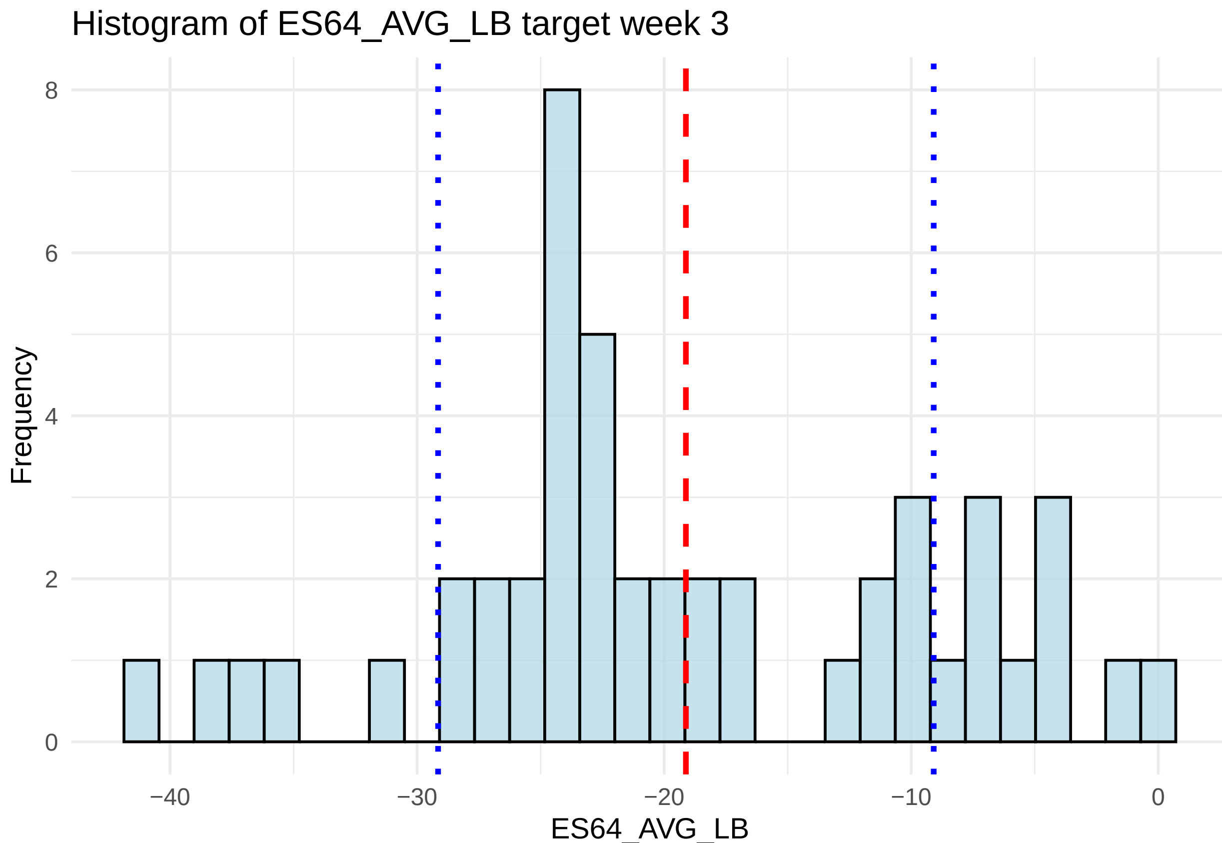


Histogram of ES27_EPI_LB target week 3
Frequency
0
2
4
6
8
−20
−10
0
10
ES27_EPI_LB

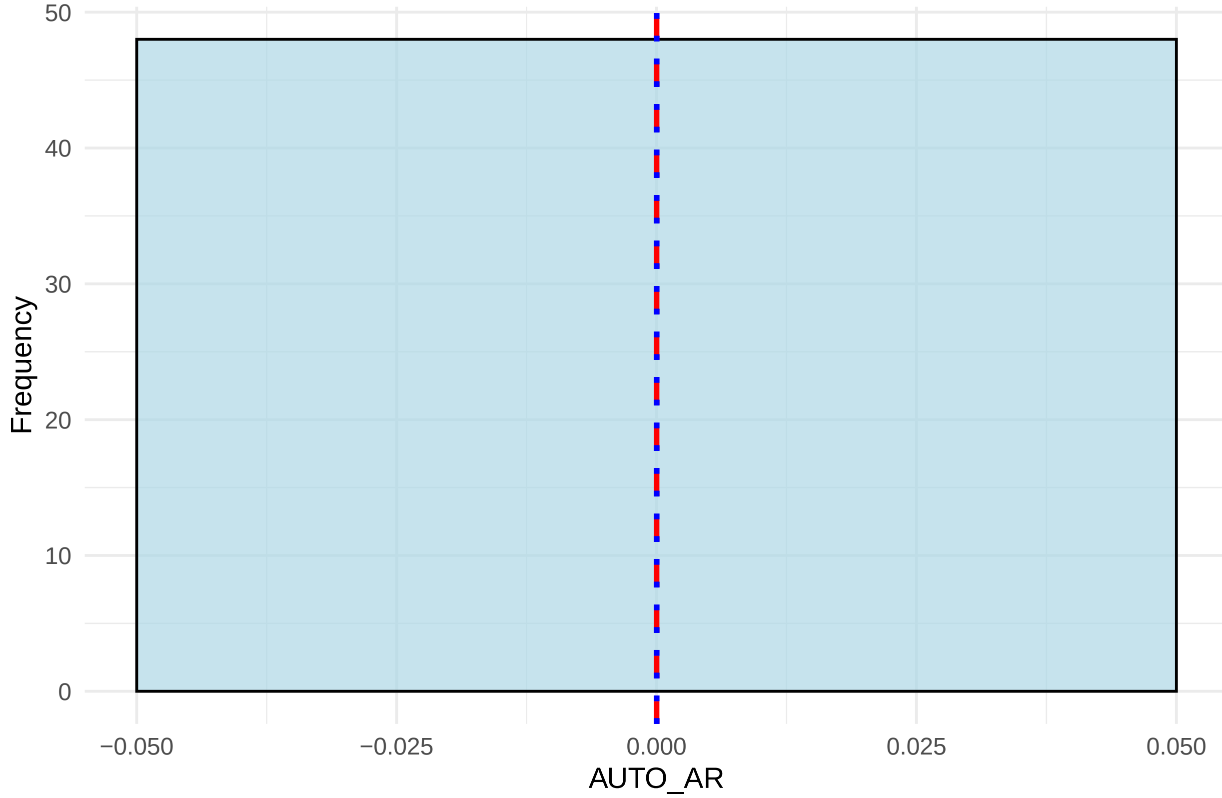

Histogram of ES64_EPI_LB target week 3
Frequency
0
3
6
9
−20
−10
0
10
ES64_EPI_LB

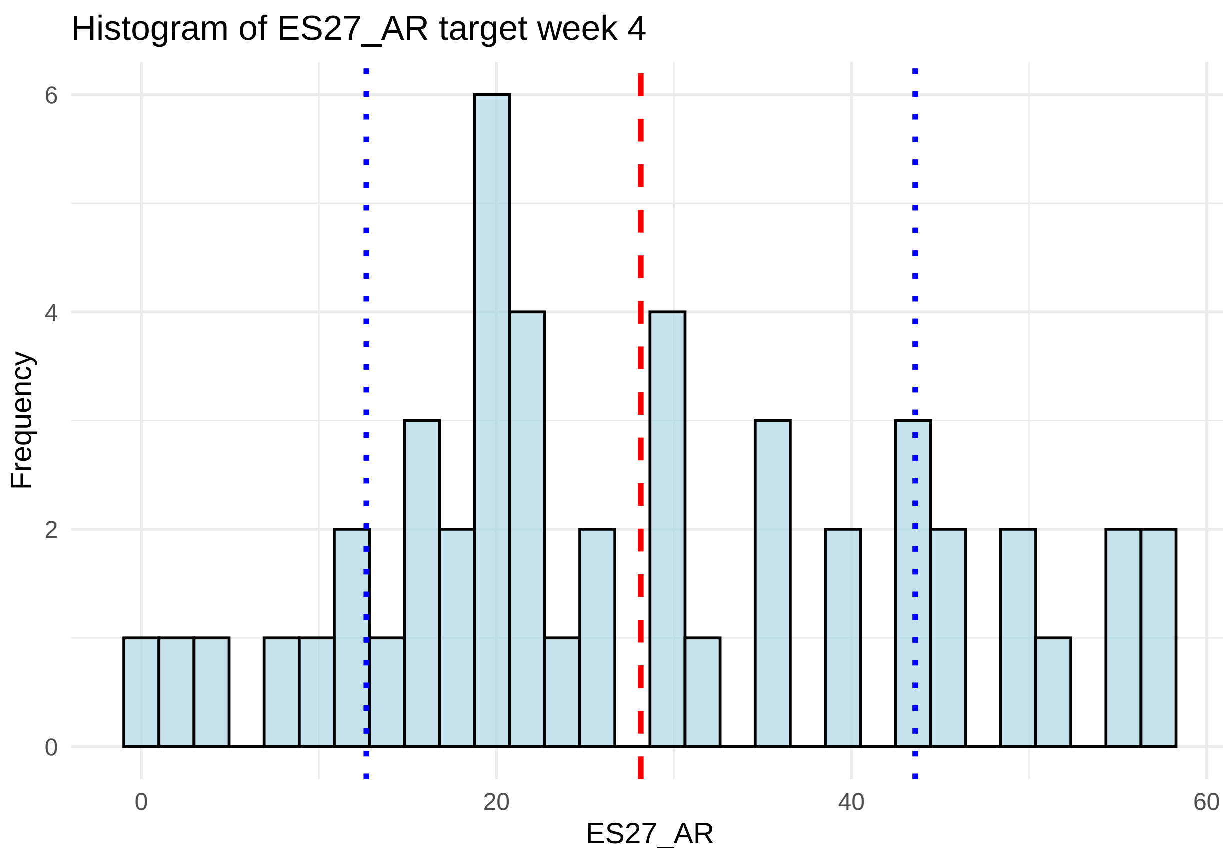

Histogram of AUTO_AVG_LB target week 3
6
4
2
0
Frequency
−40
−30
−20
−10
0
AUTO_AVG_LB

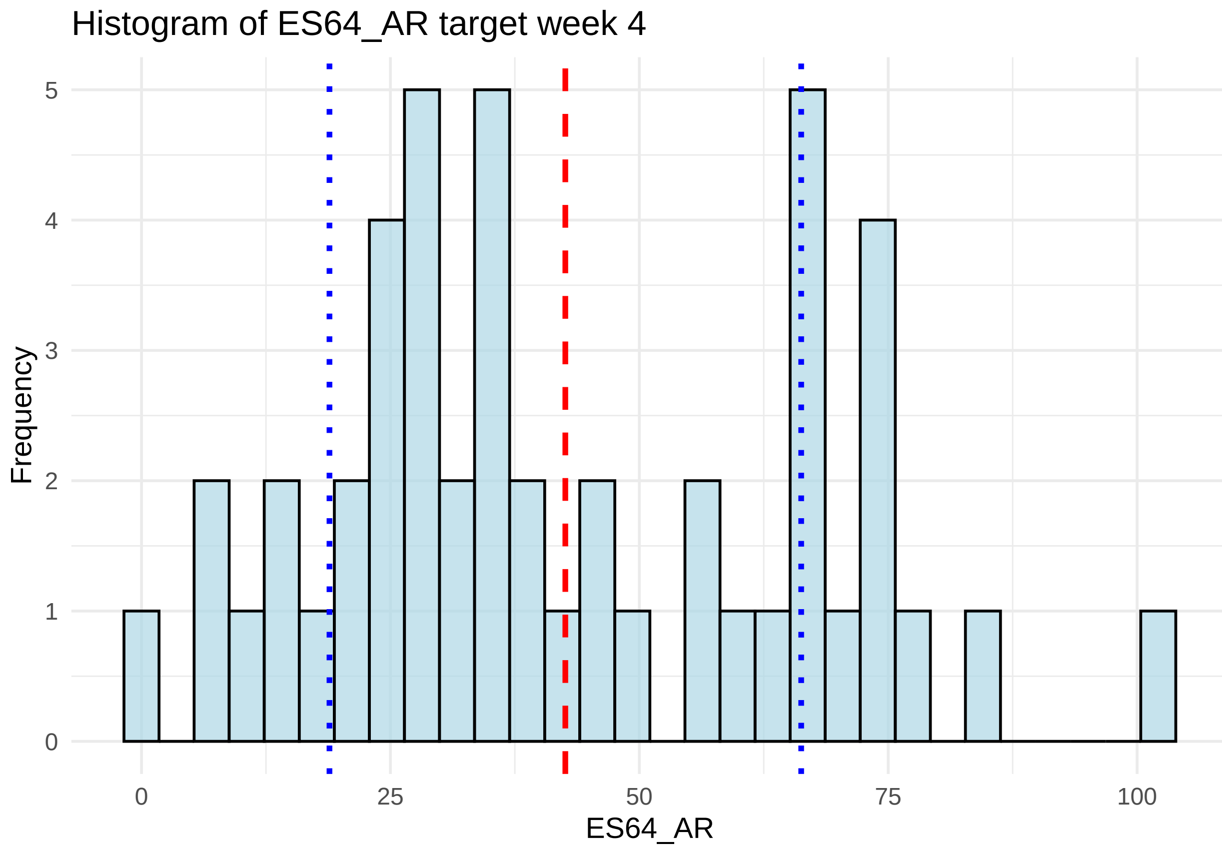

Histogram of ES27_AVG_LB target week 3
Frequency
6
4
2
0
−40
−30
−20
−10
0
ES27_AVG_LB

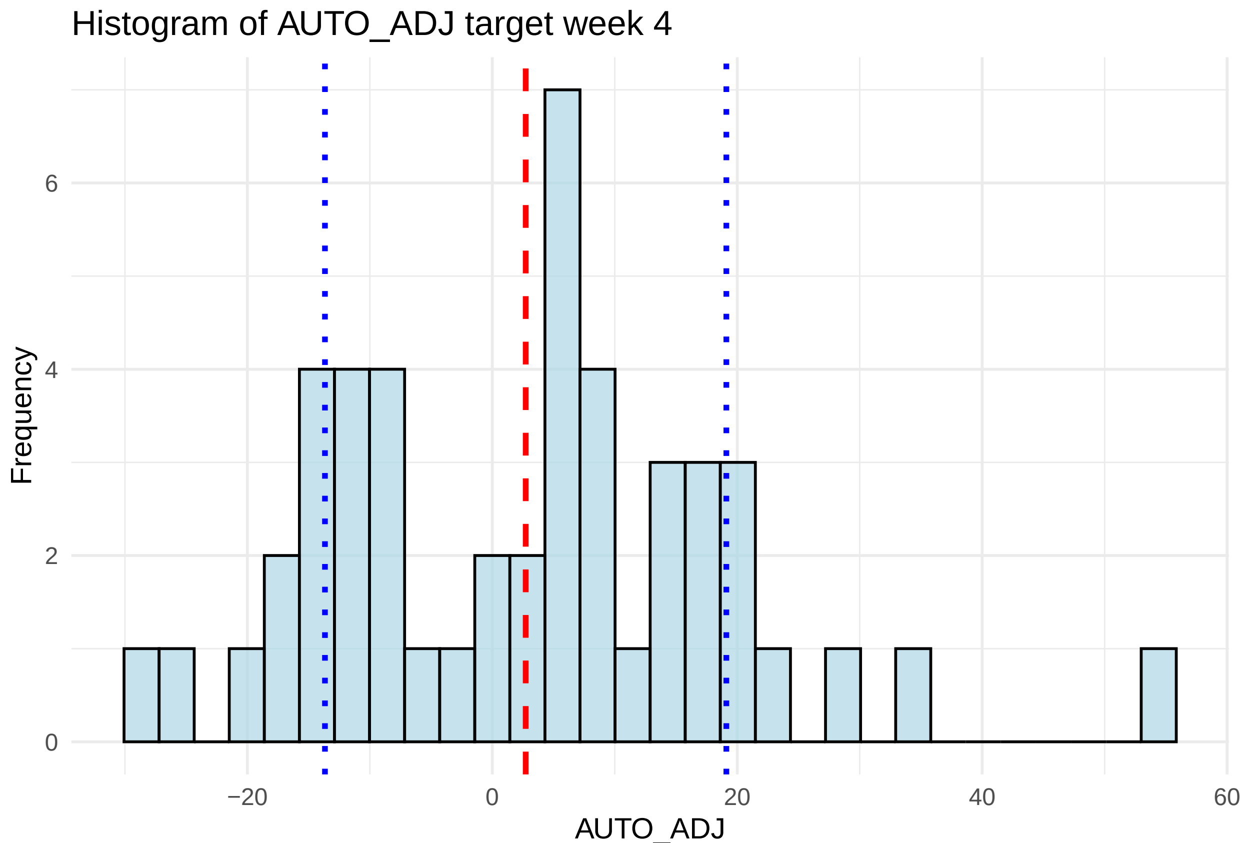

Histogram of ES64_AVG_LB target week 3
Frequency
8
6
4
2
0
−40
−30
−20
−10
0
ES64_AVG_LB

Histogram of AUTO_AR target week 4

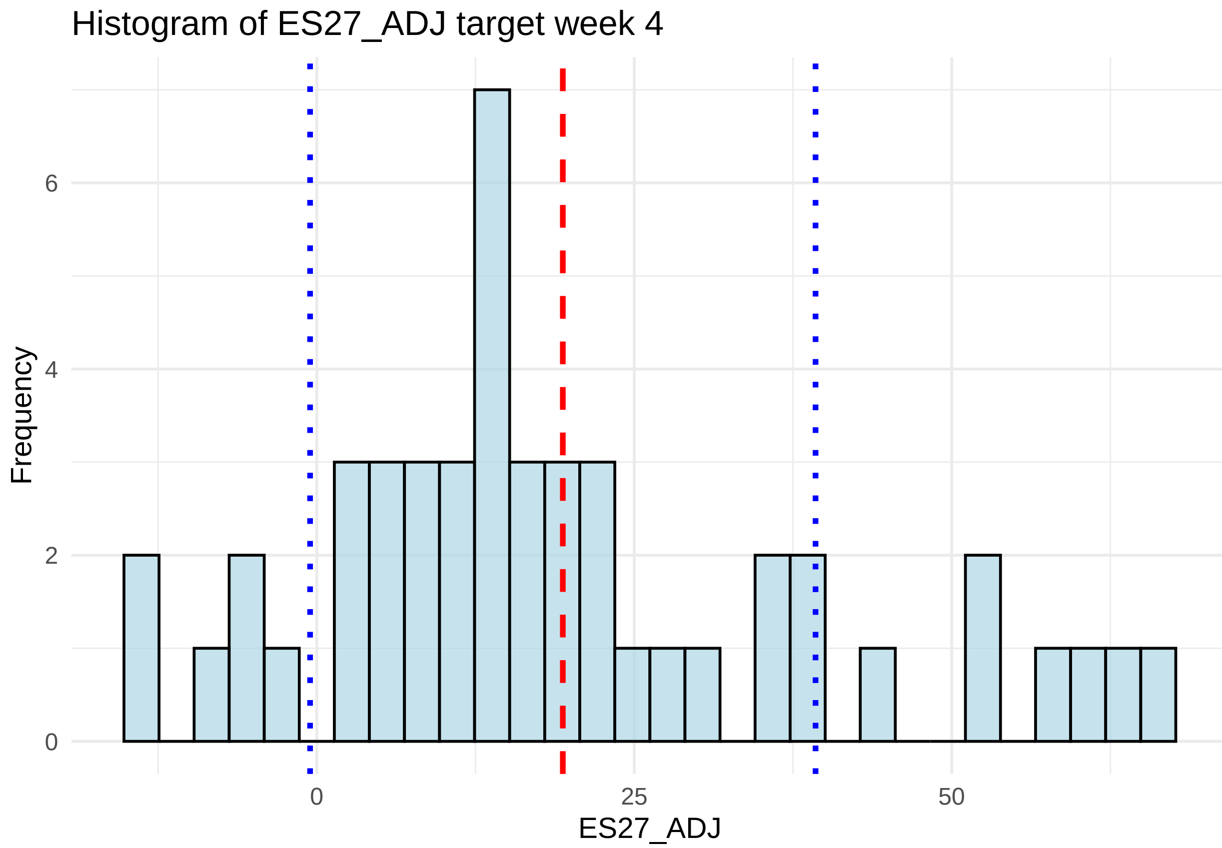


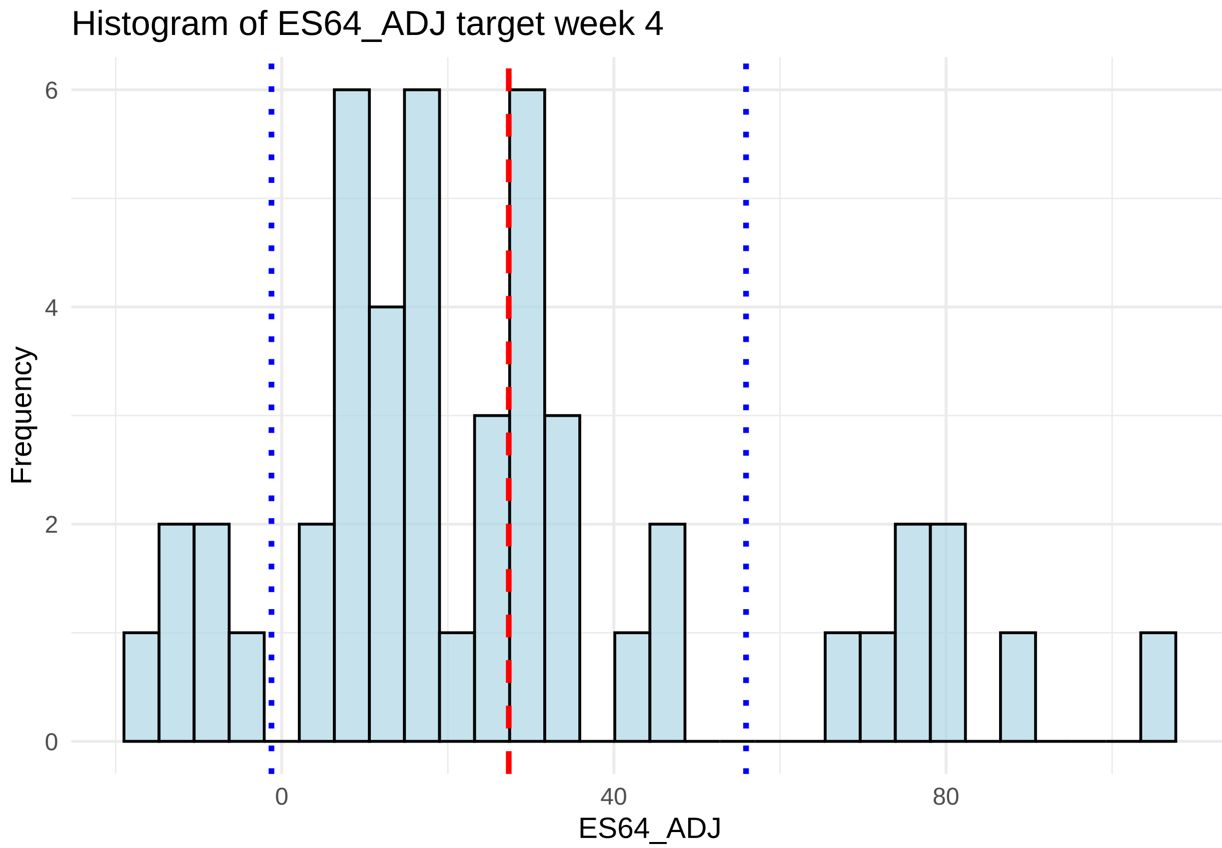

Histogram of ES27_AR target week 4
Frequency
6
4
2
0
0
20
40
60
ES27_AR

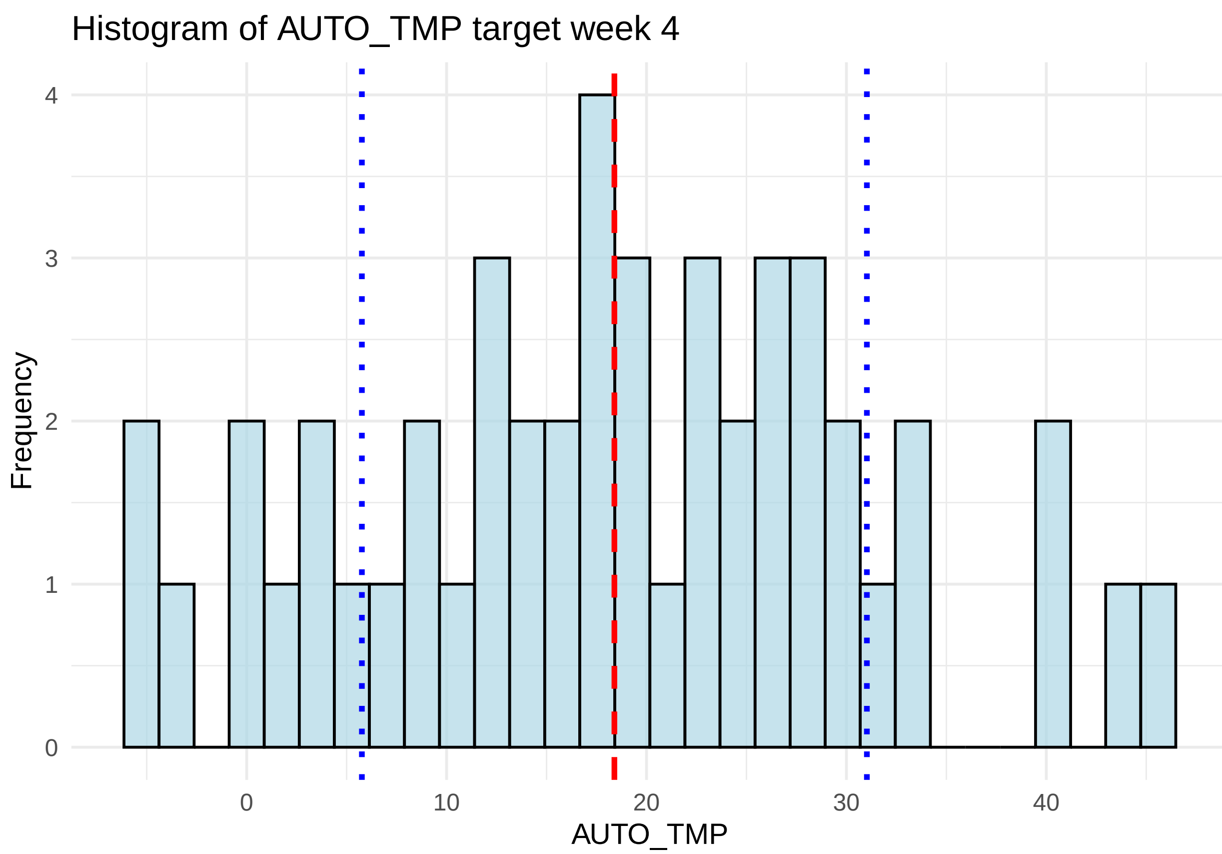

Histogram of ES64_AR target week 4
Frequency
ES64_AR
0
1
2
3
4
5
0
25
50
75
100



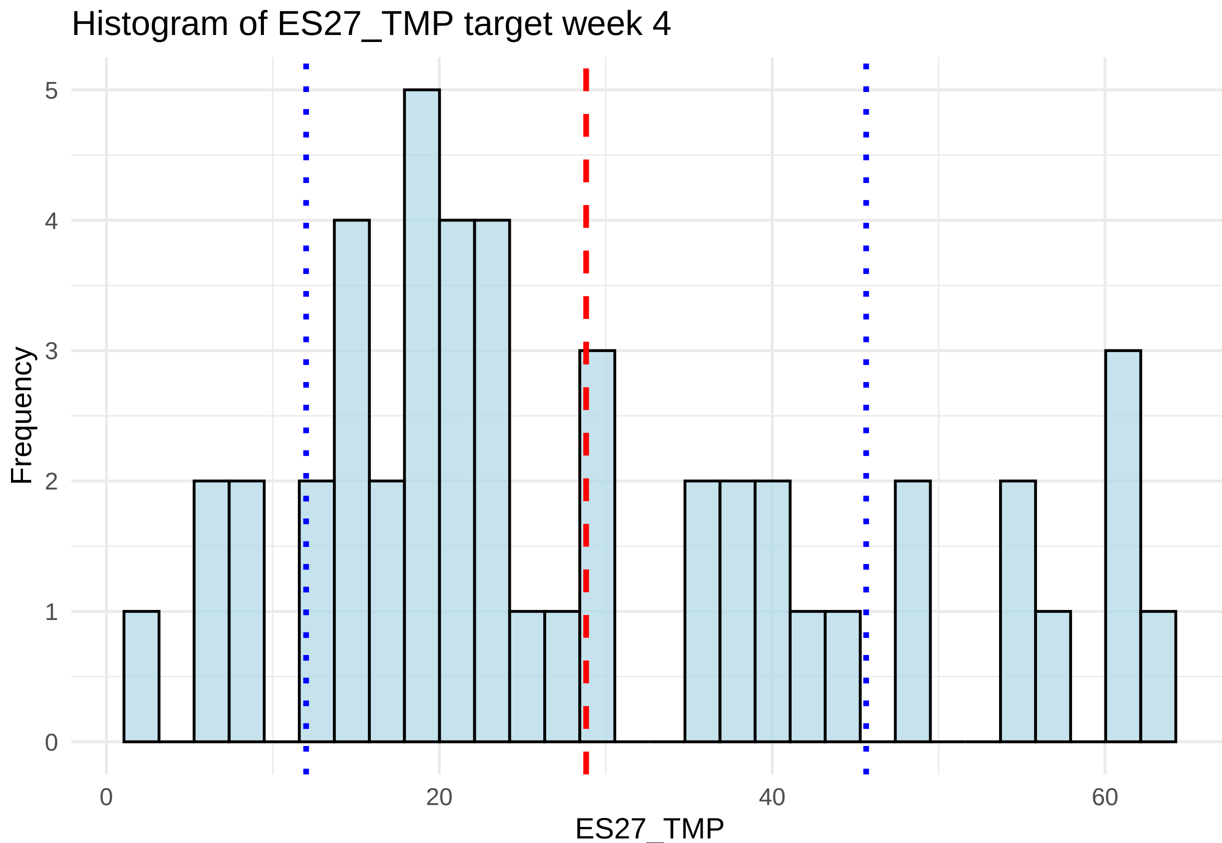
Histogram of AUTO_ADJ target week 4
Frequency
0
2
4
6
−20
0
20
40
60
AUTO_ADJ

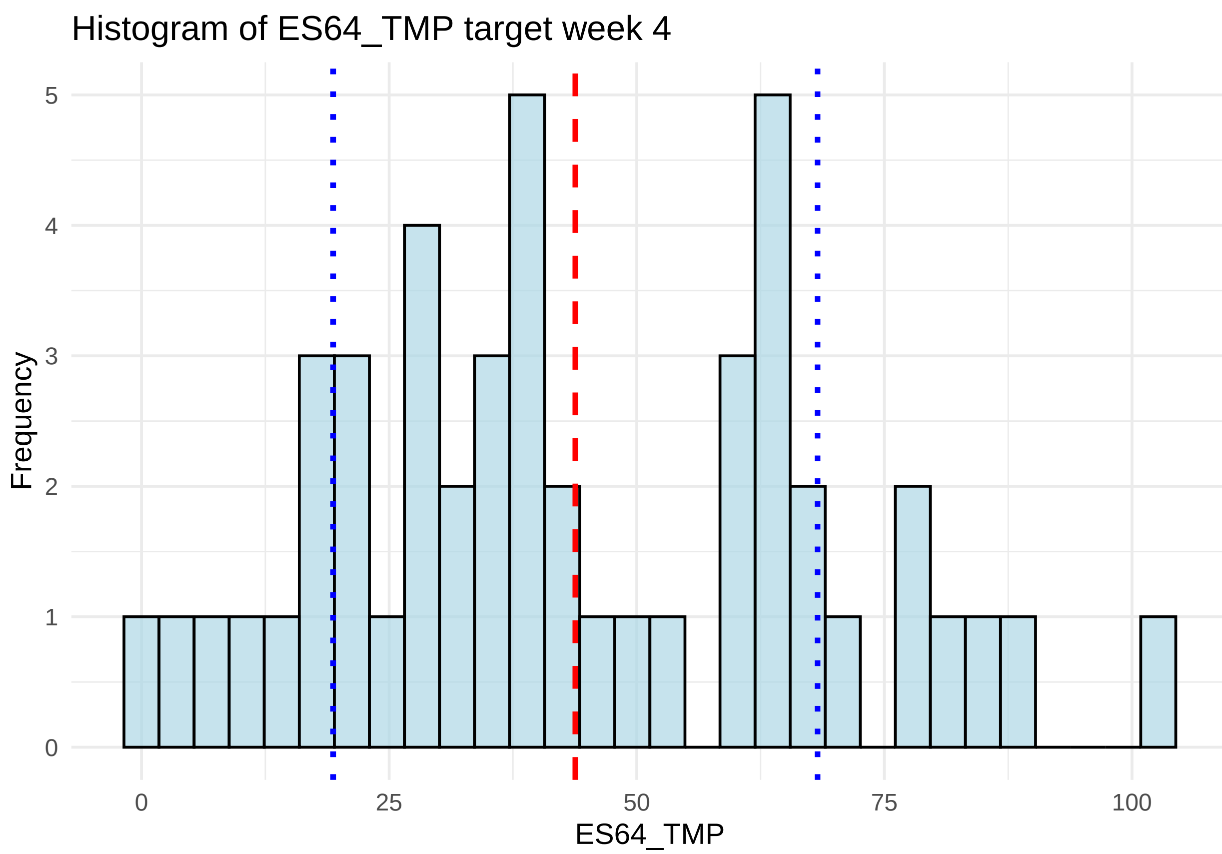

Histogram of ES27_ADJ target week 4
Frequency
0
2
4
6
0
25
50
ES27_ADJ

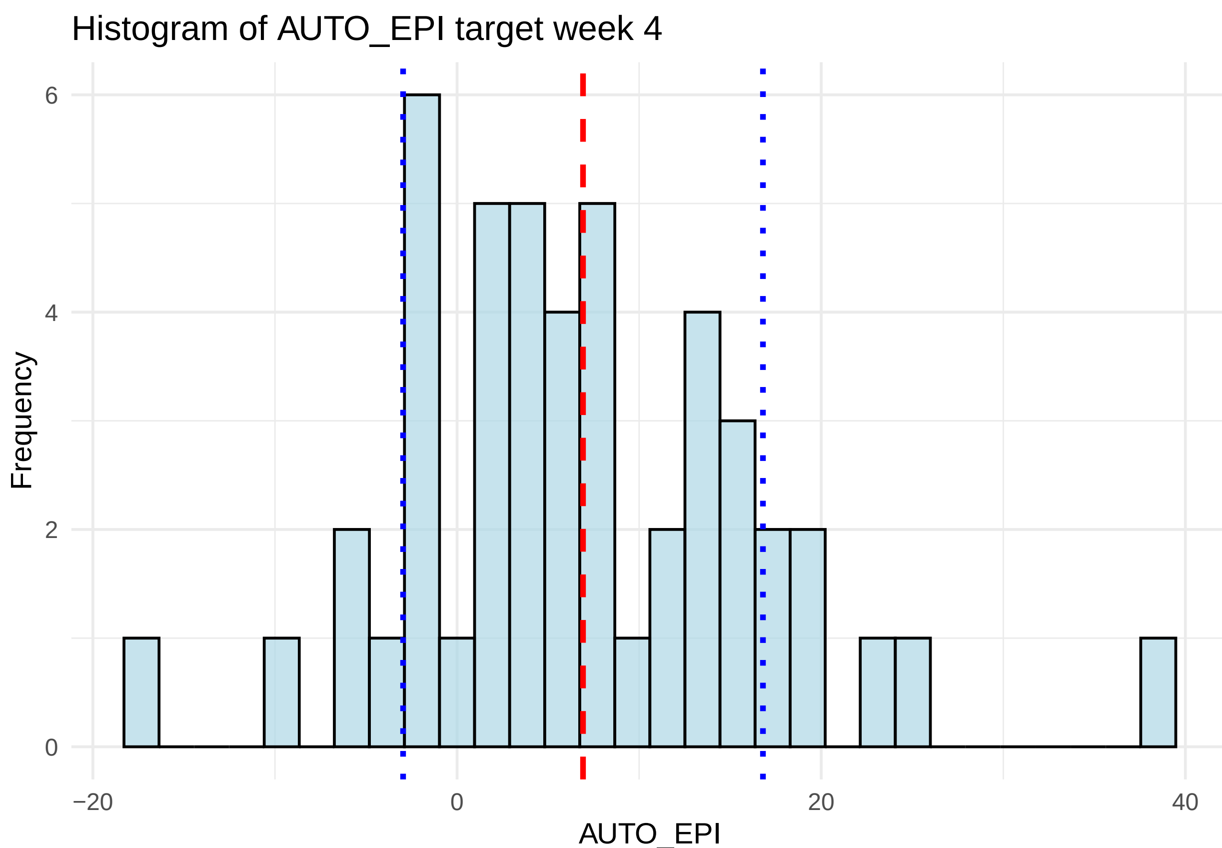

Histogram of ES64_ADJ target week 4
Frequency
ES64_ADJ
0
2
4
6
0
40
80

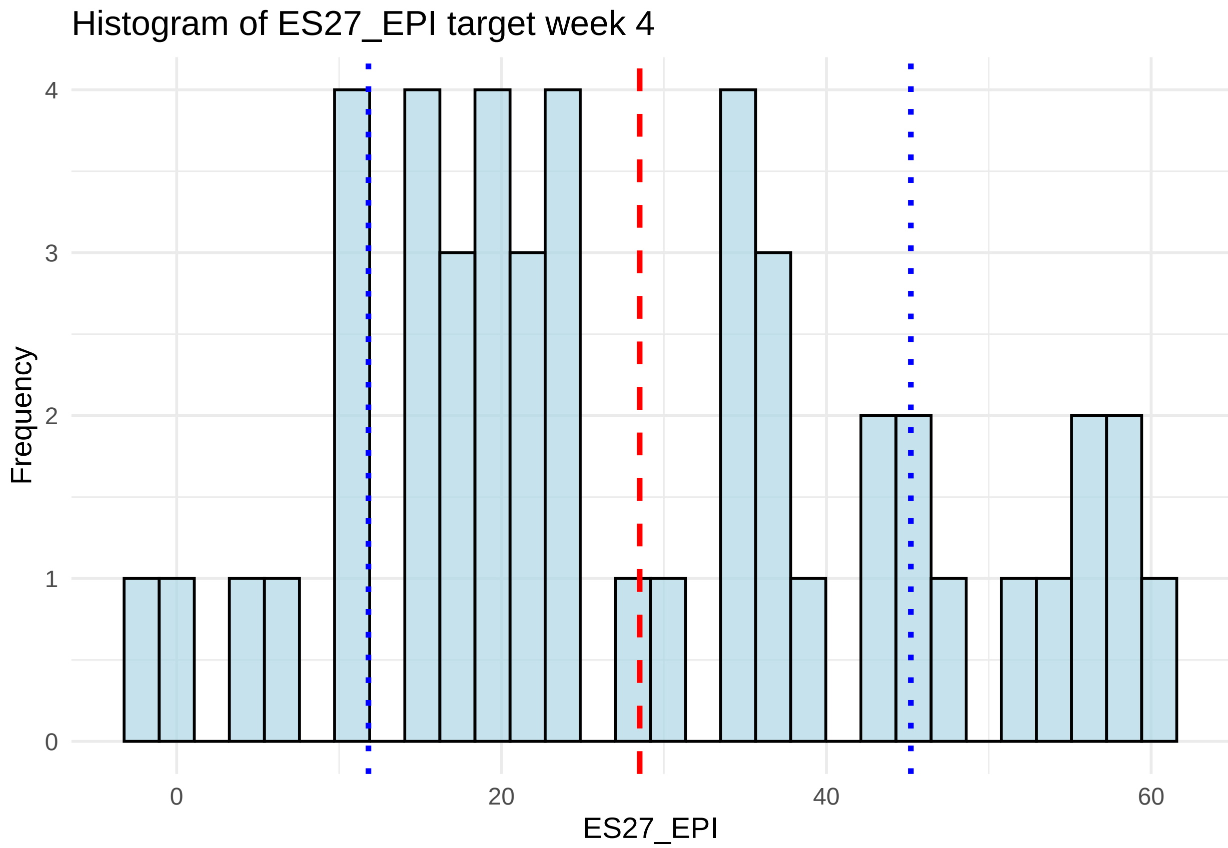


Histogram of AUTO_TMP target week 4
Frequency
0
1
2
3
4
0
10
20
30
40
AUTO_TMP

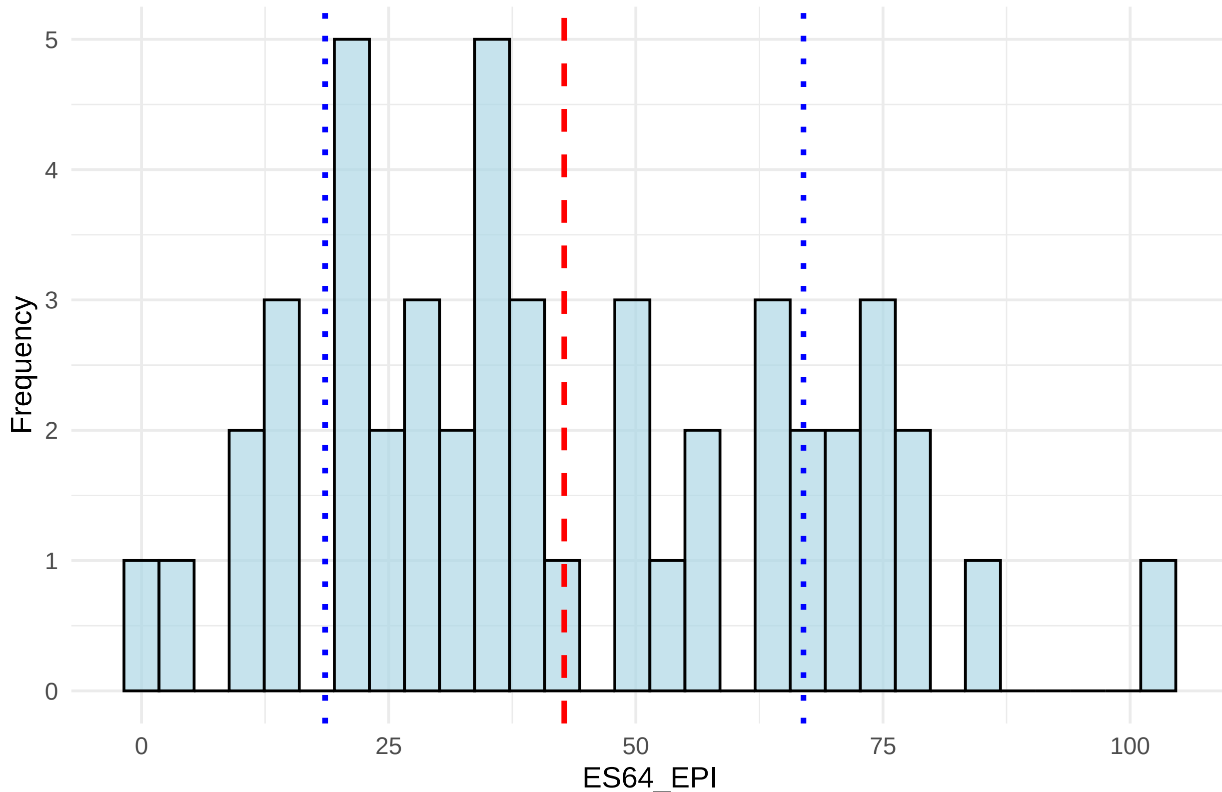

Histogram of ES27_TMP target week 4
Frequency
ES27_TMP
0
1
2
3
4
5
0
20
40
60

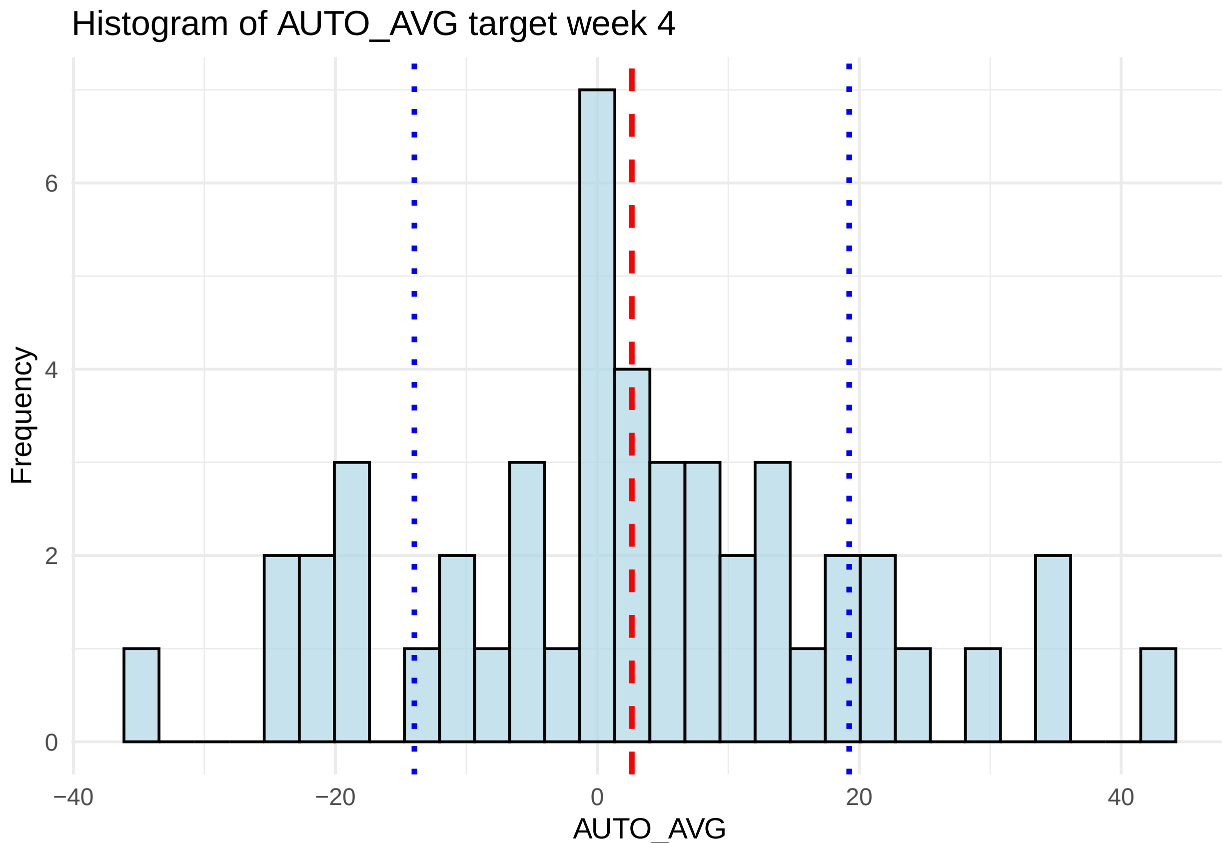

Histogram of ES64_TMP target week 4
Frequency
0
1
2
3
4
5
0
25
50
75
100
ES64_TMP

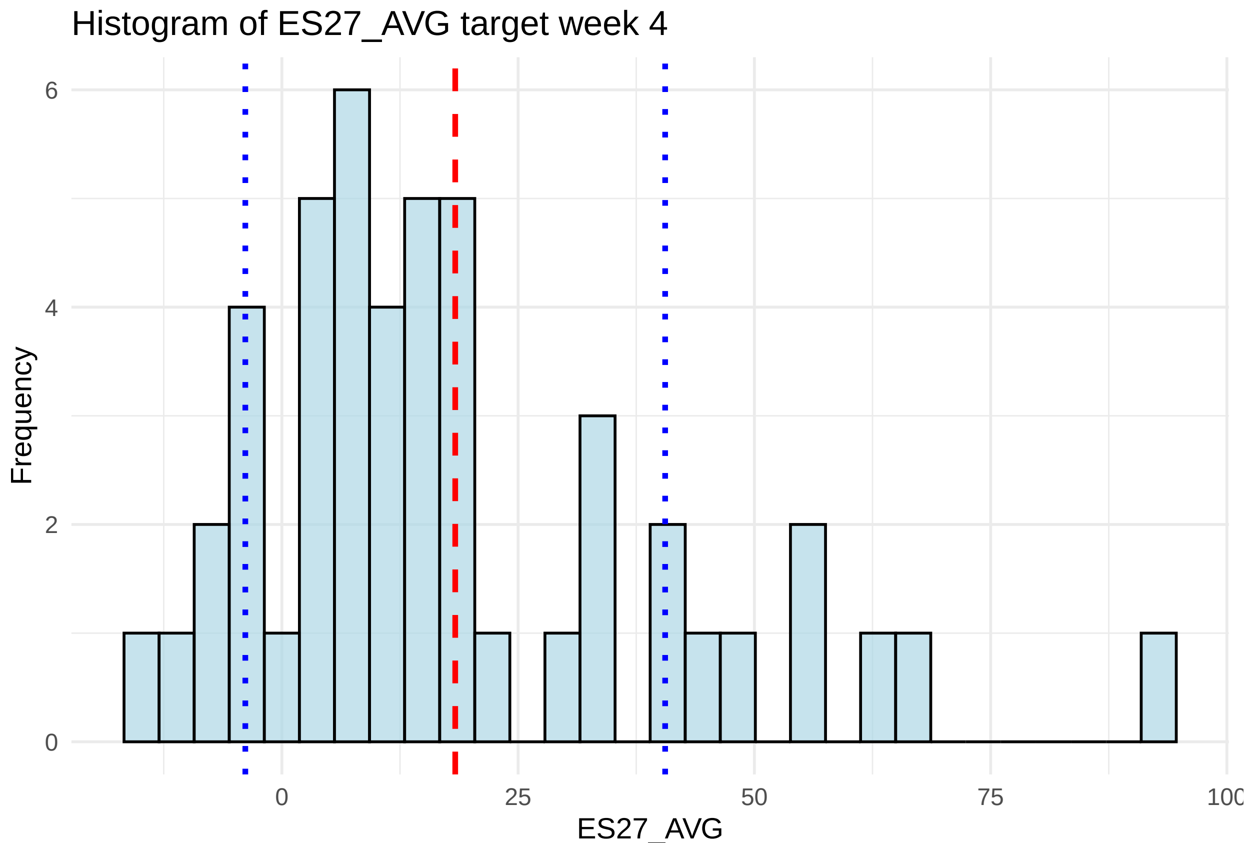

Histogram of AUTO_EPI target week 4
Frequency
AUTO_EPI
0
2
4
6
−20
0
20
40

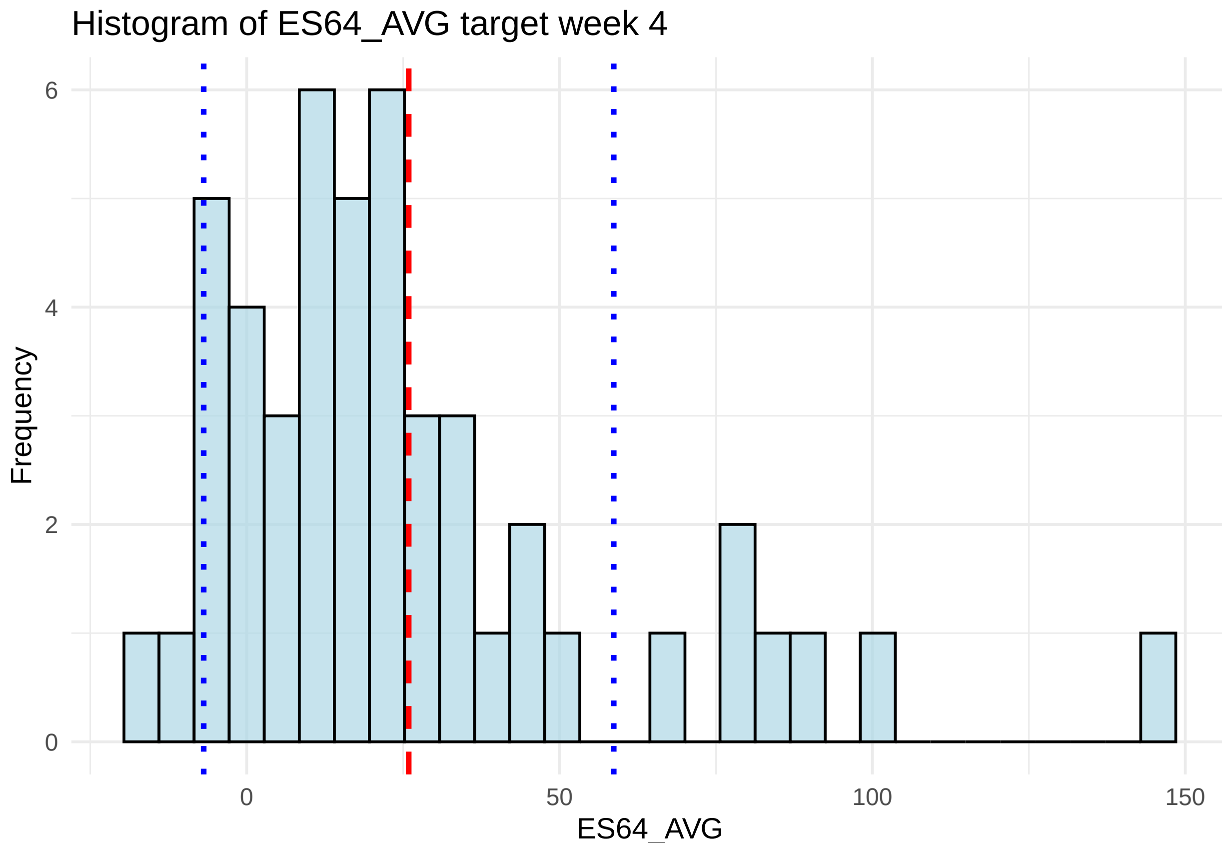

Histogram of ES27_EPI target week 4
Frequency
ES27_EPI
0
1
2
3
4
0
20
40
60

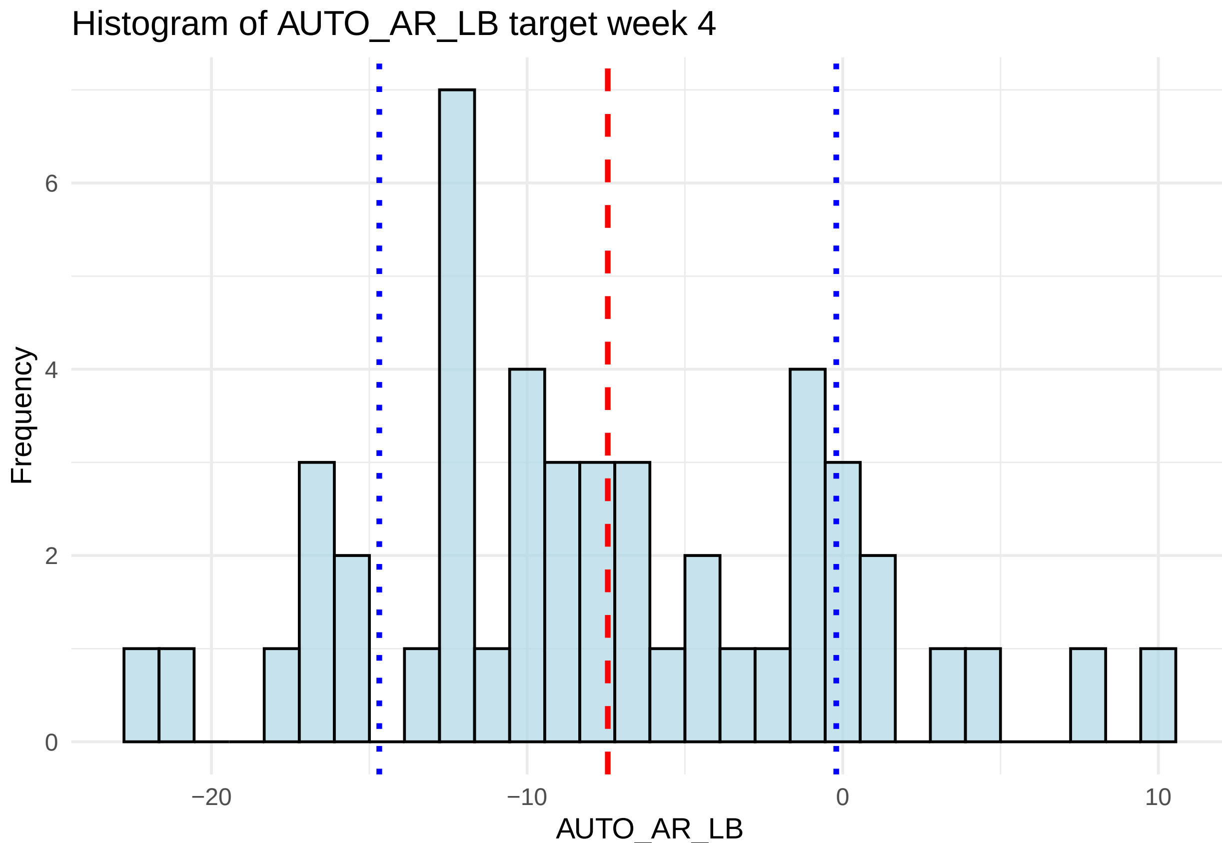
Histogram of ES64_EPI target week 4

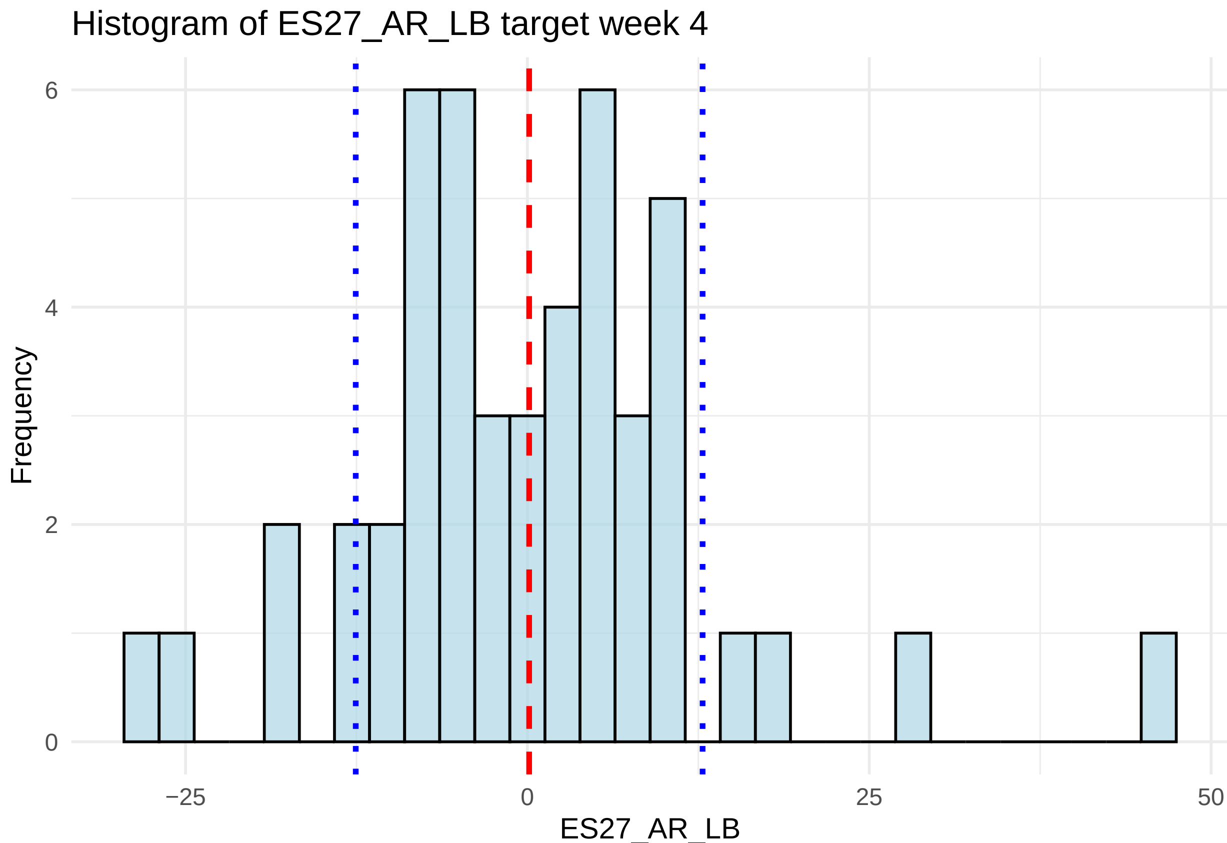



Histogram of AUTO_AVG target week 4
Frequency
0
2
4
6
−40
−20
0
20
40
AUTO_AVG

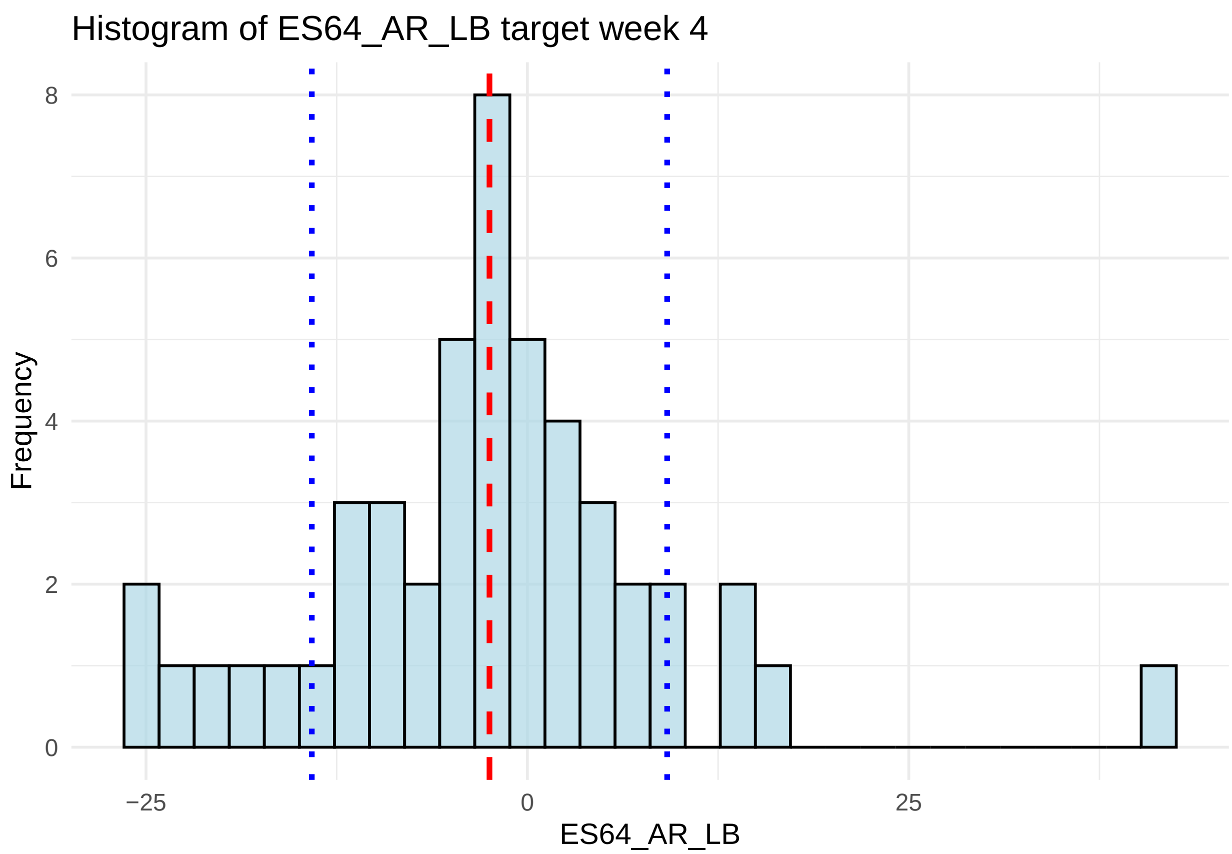

Histogram of ES27_AVG target week 4
6
4
2
0
Frequency
0
25
50
75
100
ES27_AVG

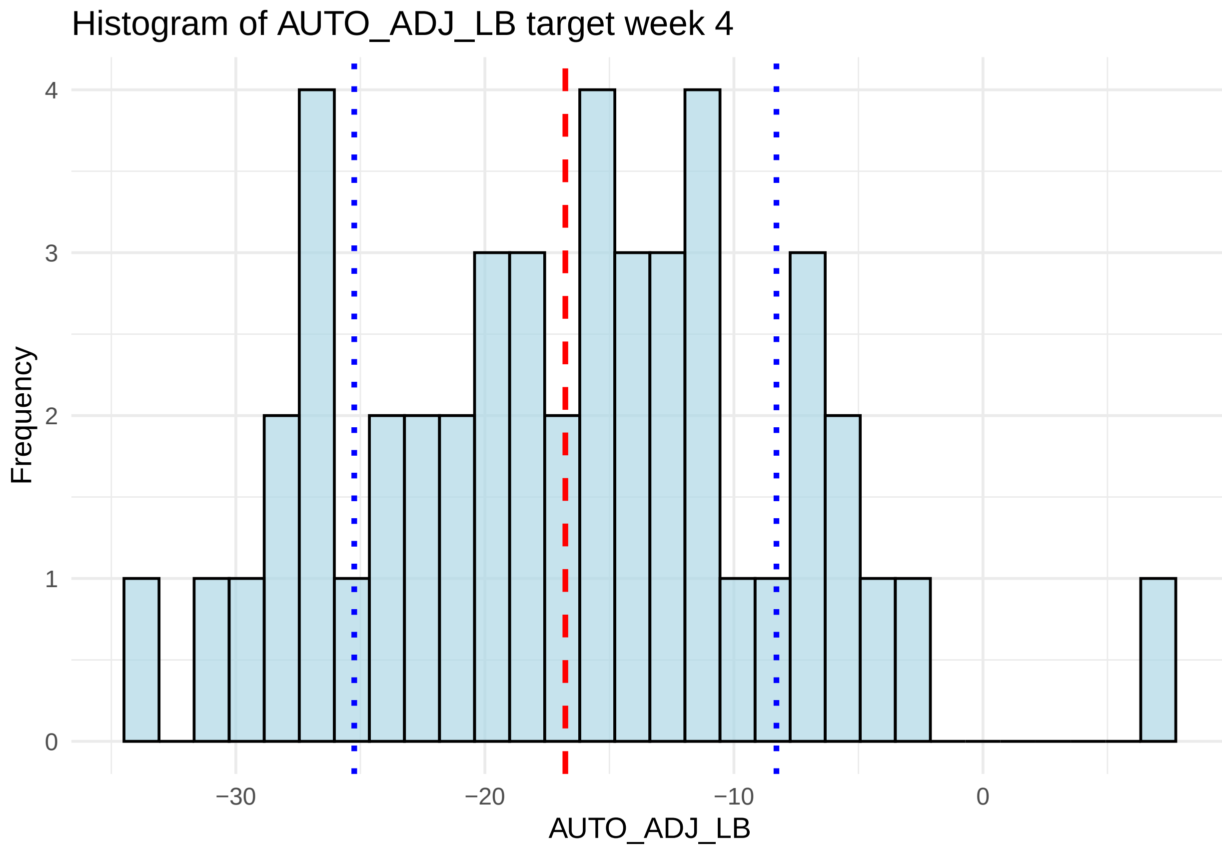

Histogram of ES64_AVG target week 4
Frequency
ES64_AVG
0
2
4
6
0
50
100
150

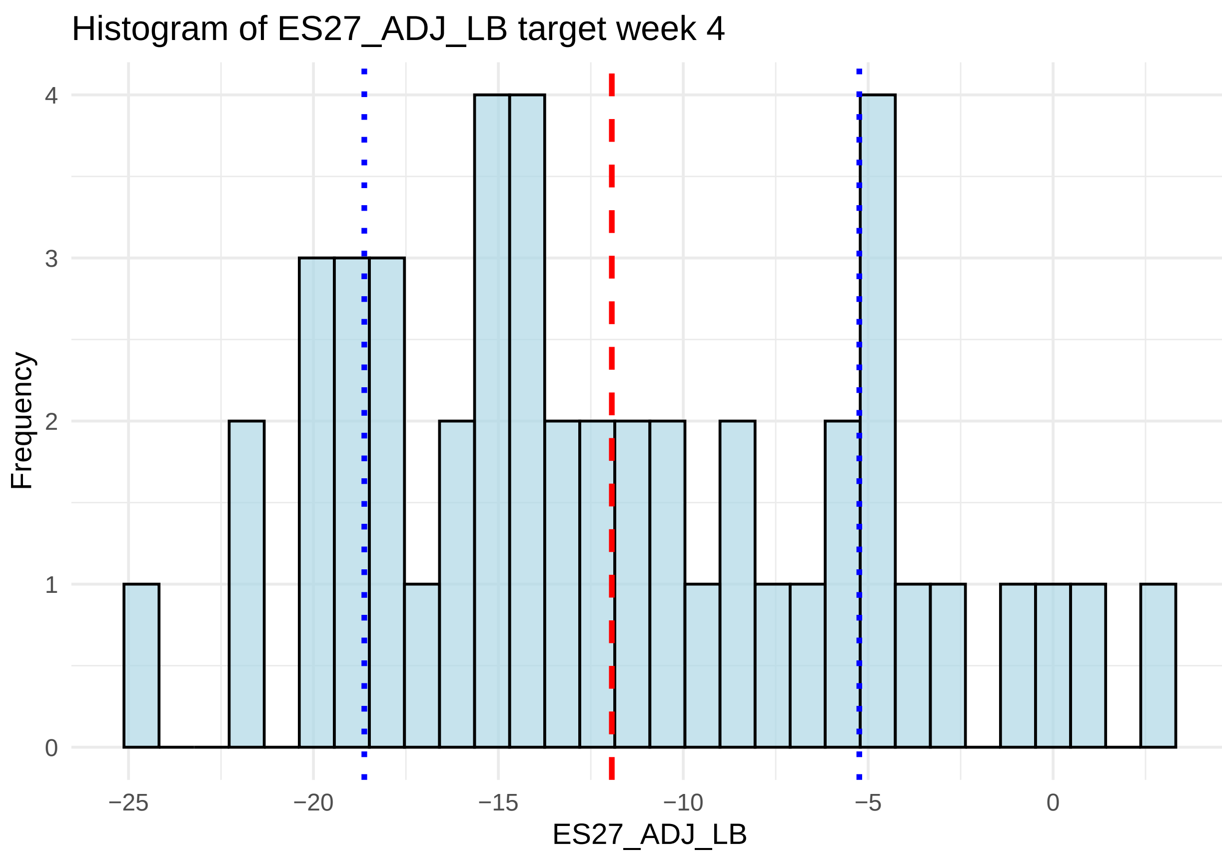

Histogram of AUTO_AR_LB target week 4
Frequency
0
2
4
6
−20
−10
0
10
AUTO_AR_LB

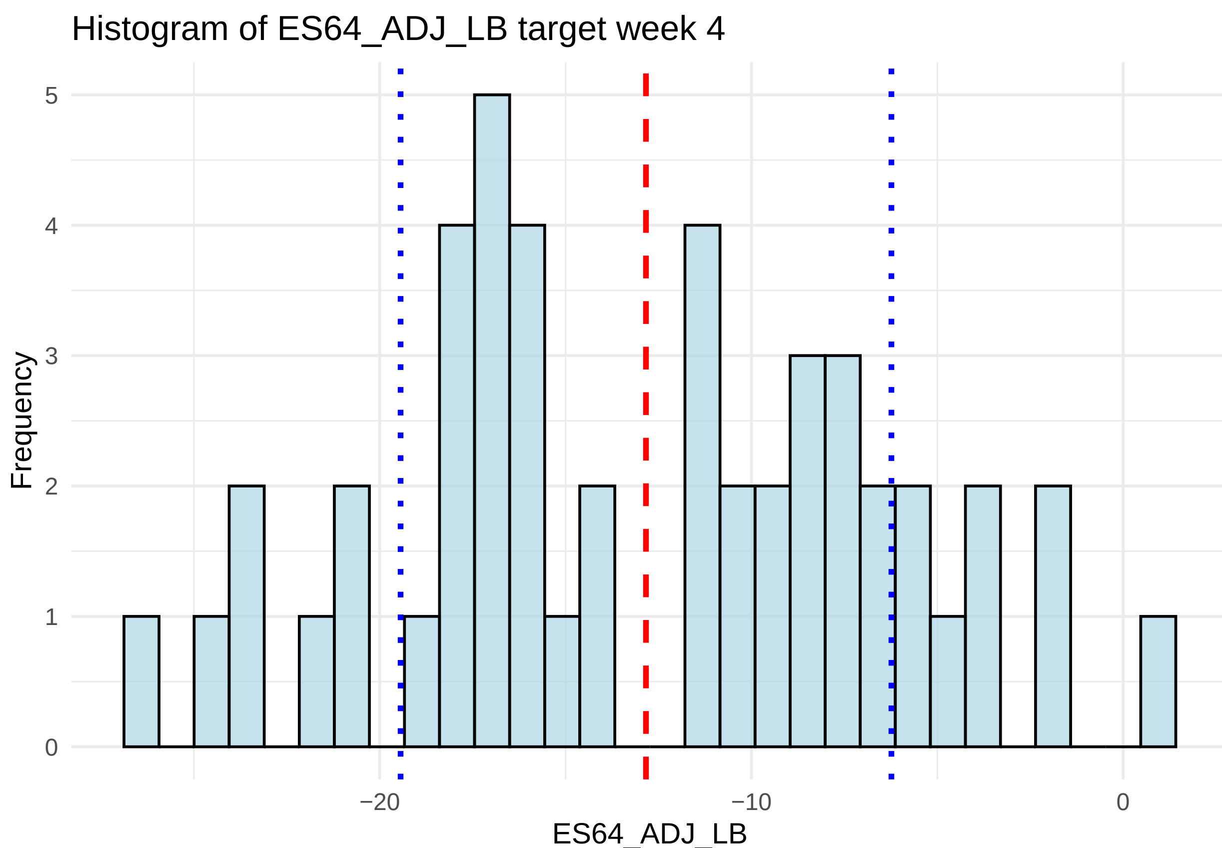


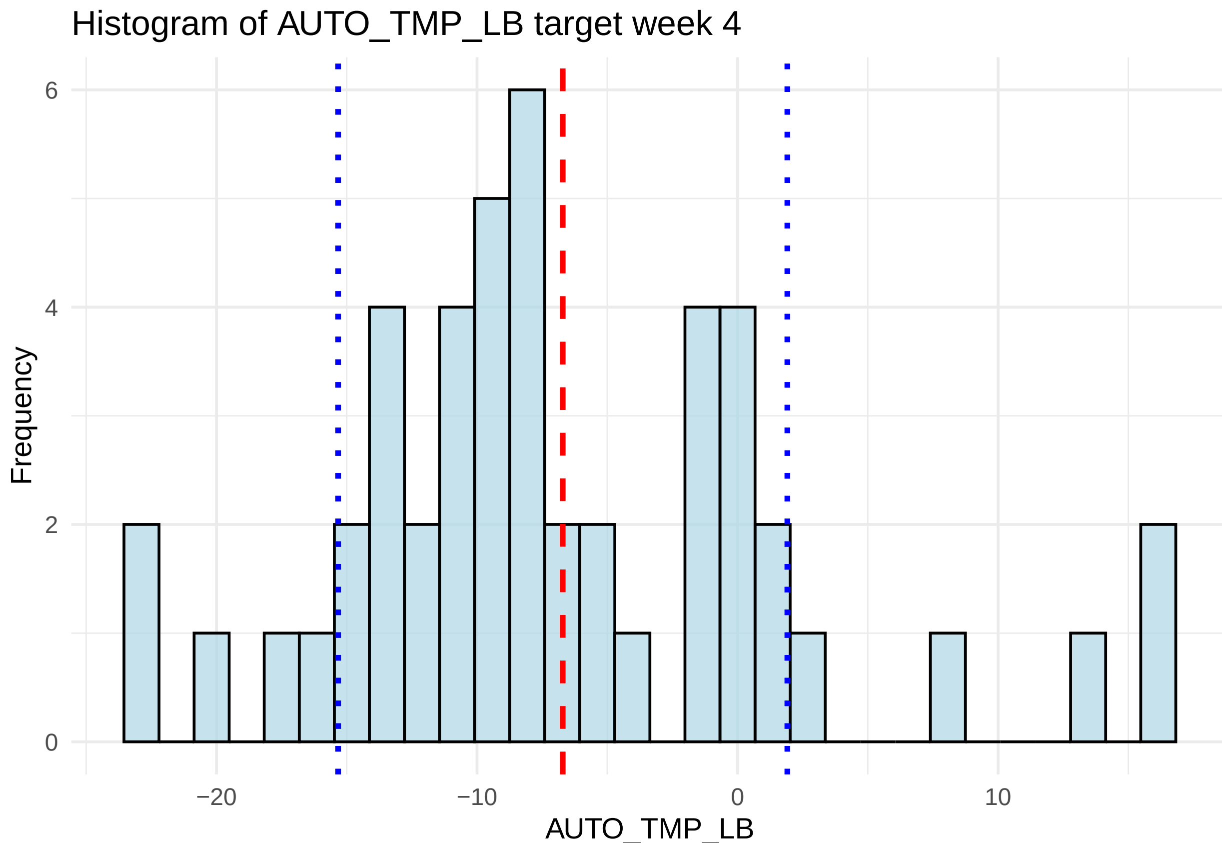
Histogram of ES27_AR_LB target week 4
Frequency
0
2
4
6
−25
0
25
50
ES27_AR_LB


Histogram of ES64_AR_LB target week 4
Frequency
0
2
4
6
8
−25
0
25
ES64_AR_LB

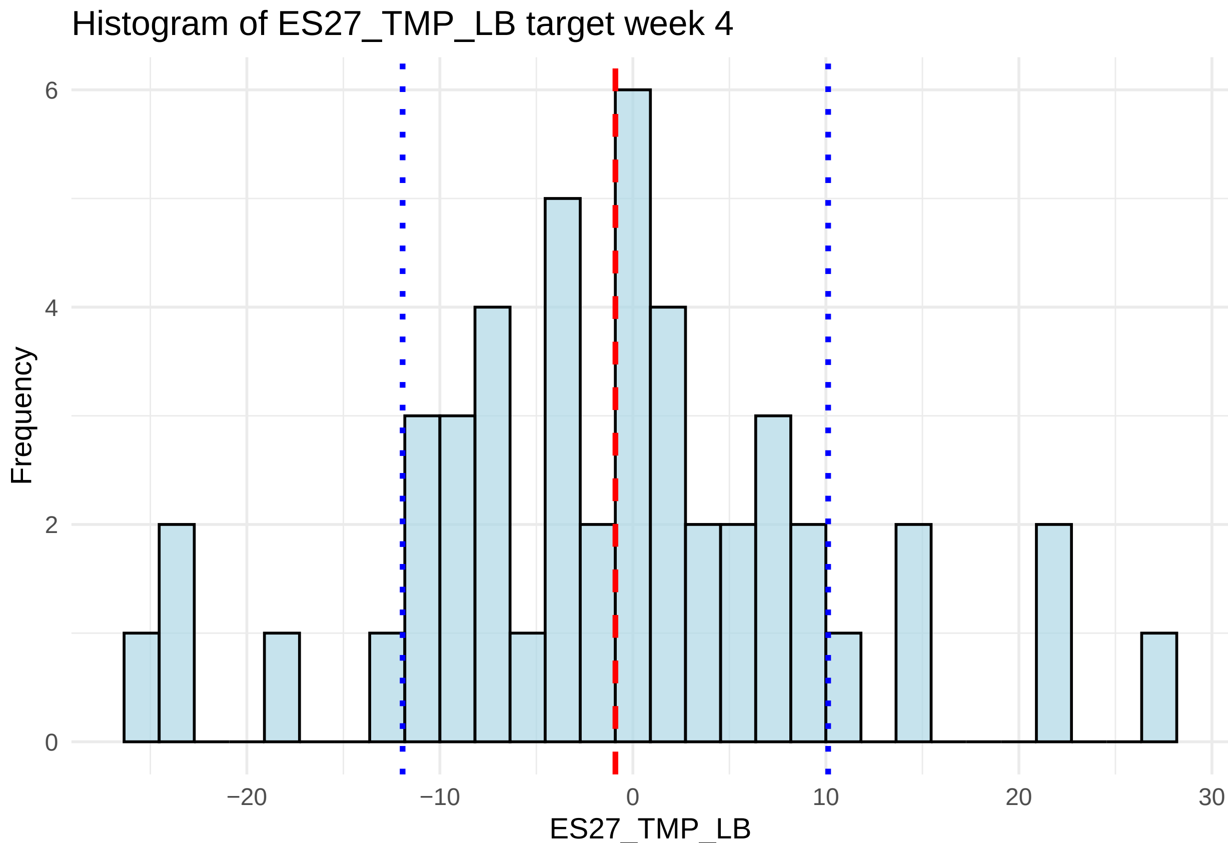

Histogram of AUTO_ADJ_LB target week 4
Frequency
4
3
2
1
0
−30
−20
−10
0
AUTO_ADJ_LB

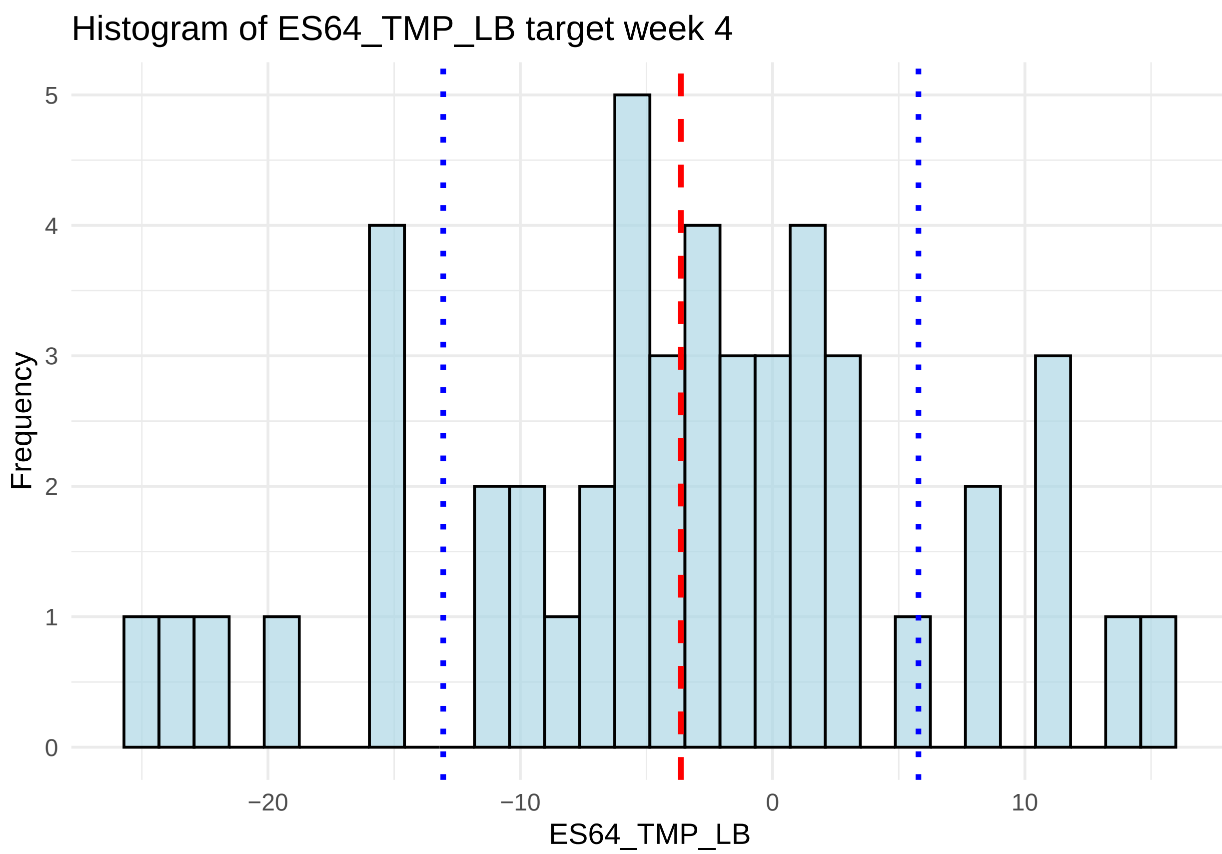

Histogram of ES27_ADJ_LB target week 4
Frequency
ES27_ADJ_LB
0
1
2
3
4
−25
−20
−15
−10
−5
0

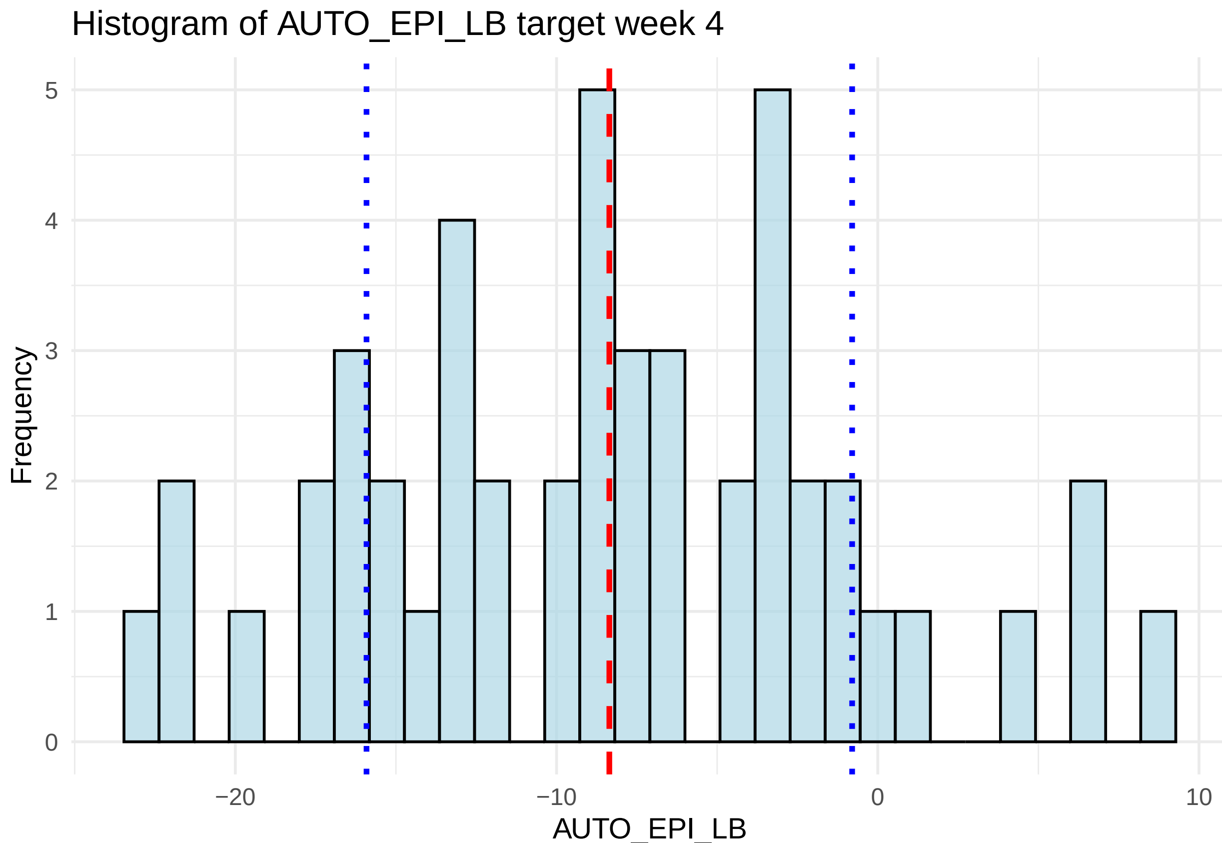

Histogram of ES64_ADJ_LB target week 4
Frequency
5
4
3
2
1
0
−20
−10
0
ES64_ADJ_LB

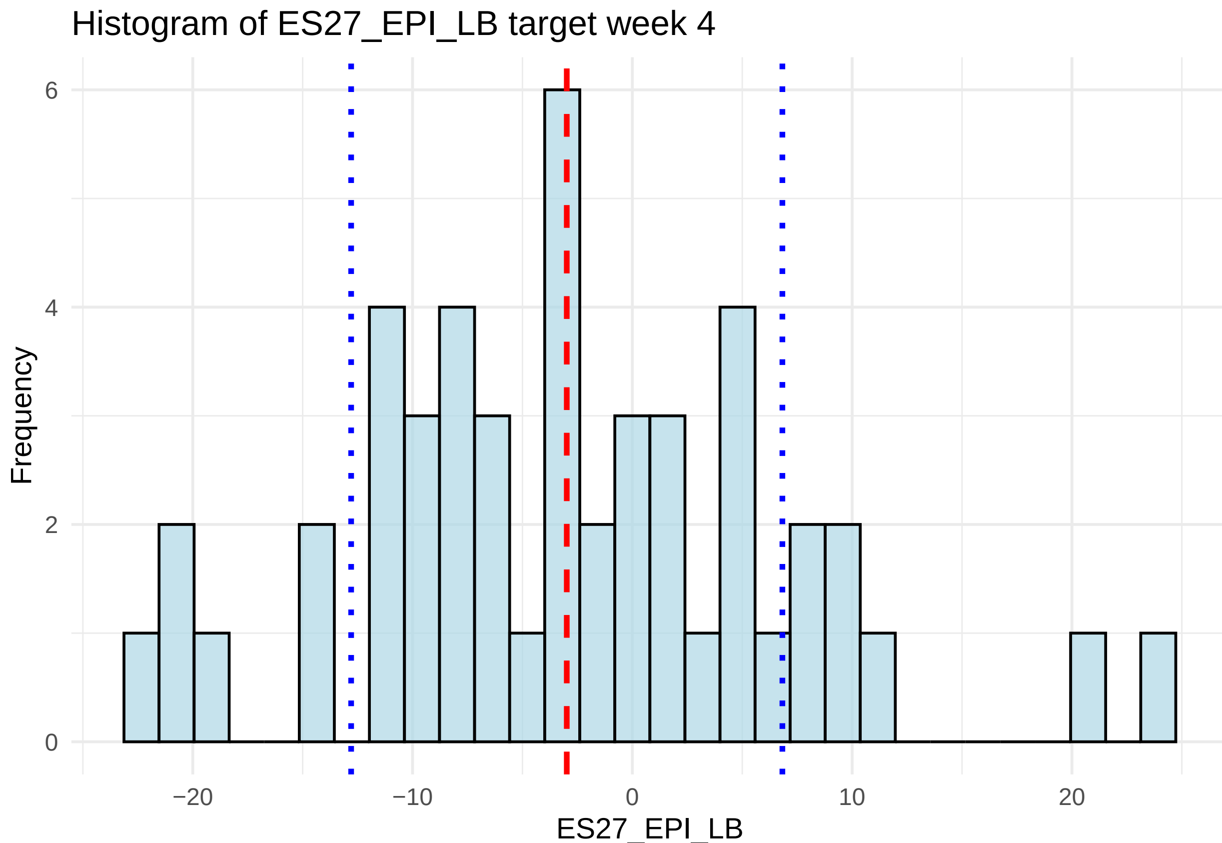

Histogram of AUTO_TMP_LB target week 4
Frequency
6
4
2
0
−20
−10
0
10
AUTO_TMP_LB

Histogram of ES27_TMP_LB target week 4

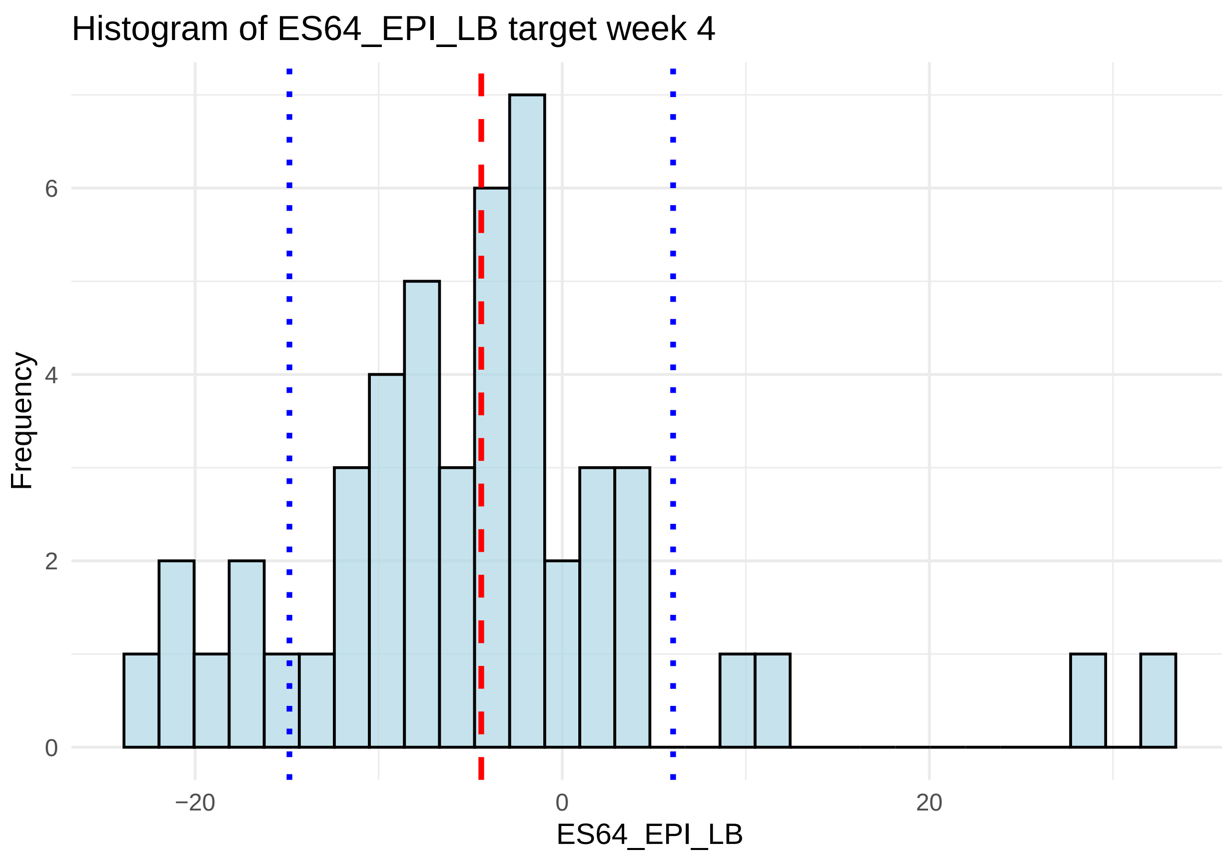


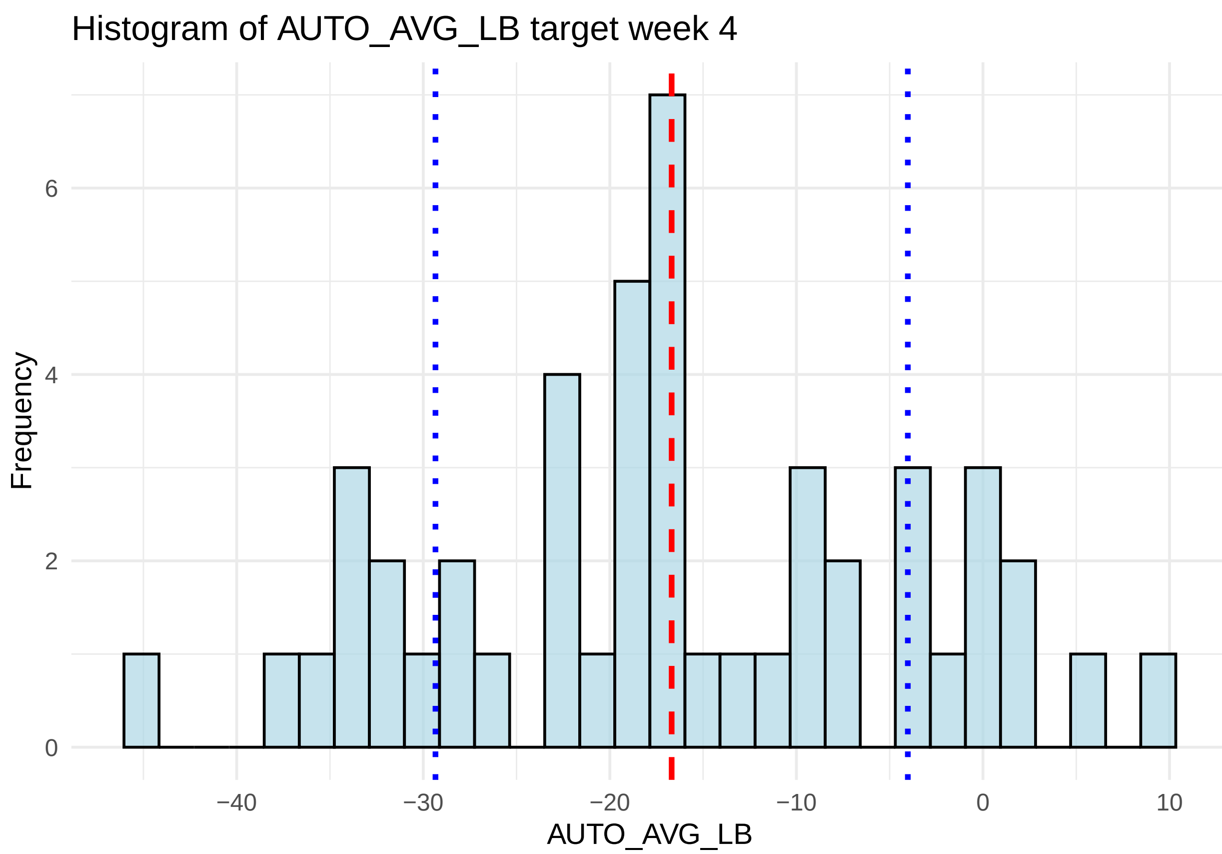

Histogram of ES64_TMP_LB target week 4
Frequency
0
1
2
3
4
5
−20
−10
0
10
ES64_TMP_LB

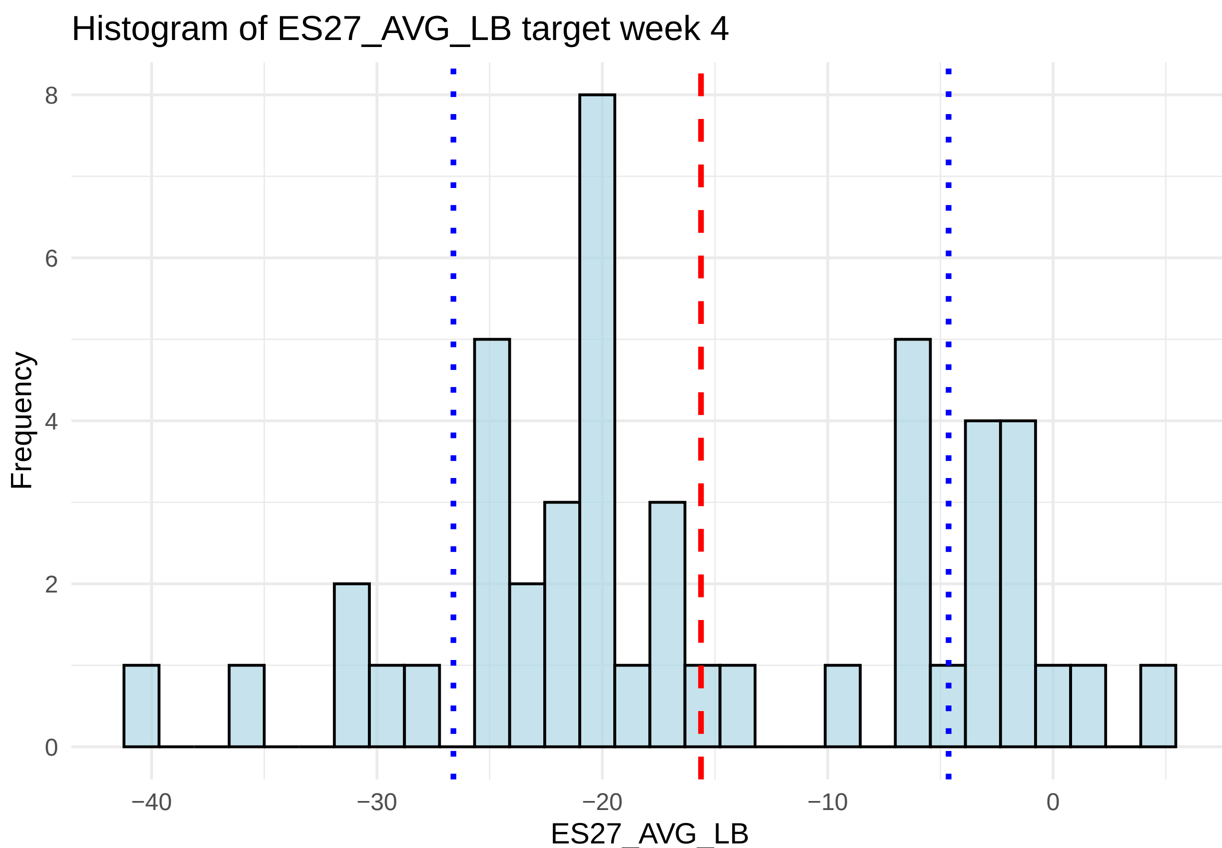

Histogram of AUTO_EPI_LB target week 4
Frequency
5
4
3
2
1
0
−20
−10
0
10
AUTO_EPI_LB

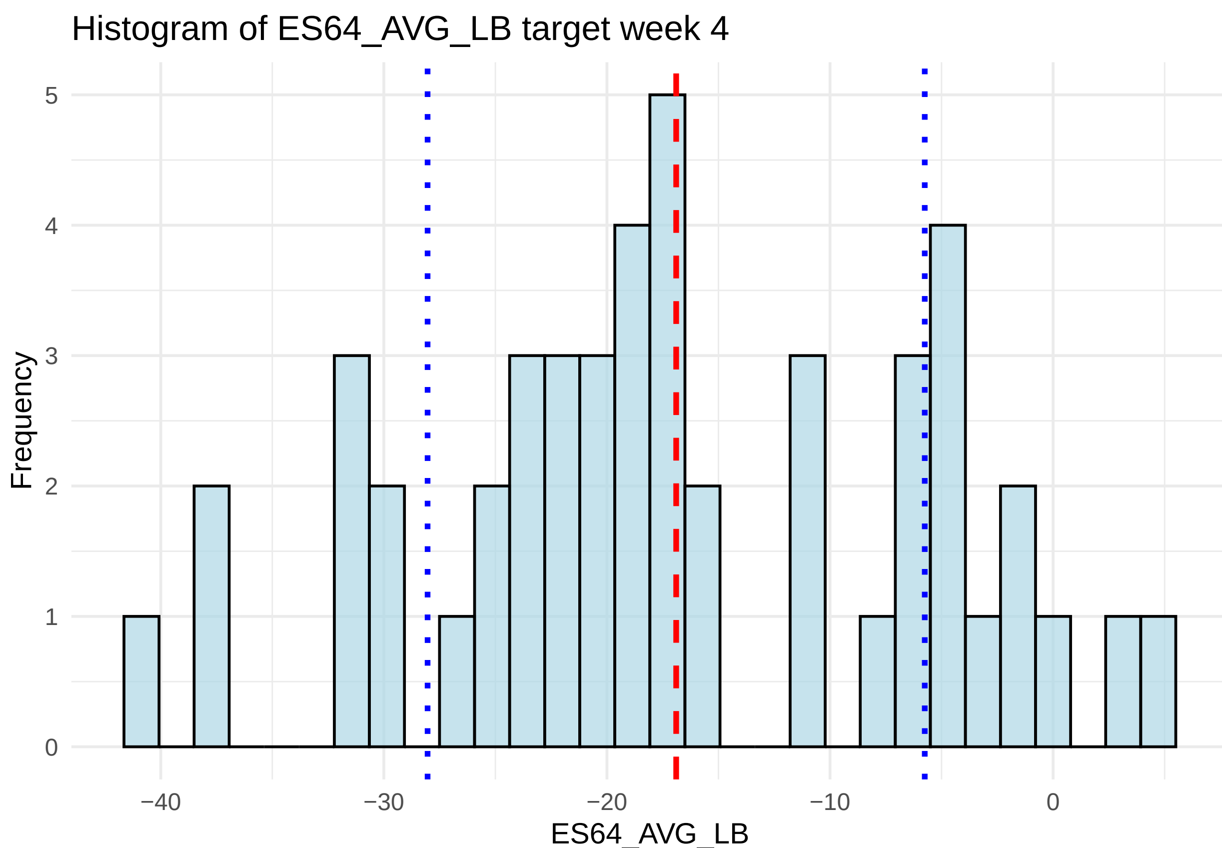


Histogram of ES27_EPI_LB target week 4
Frequency
0
2
4
6
−20
−10
0
10
20
ES27_EPI_LB


Histogram of ES64_EPI_LB target week 4
Frequency
0
2
4
6
−20
0
20
ES64_EPI_LB


Histogram of AUTO_AVG_LB target week 4
Frequency
0
2
4
6
−40
−30
−20
−10
0
10
AUTO_AVG_LB


Histogram of ES27_AVG_LB target week 4
Frequency
8
6
4
2
0
−40
−30
−20
−10
0
ES27_AVG_LB




```r
summary_impr$WeekAhead<-as.numeric(summary_impr$WeekAhead)

#summary_impr
```

Let create a combined dataset with mean differences, standard deviations and Wilcoxon Holm adjusted p-values

```r
#

all_p_values<-rbind(p_values_wk1,p_values_wk2,p_values_wk3,p_values_wk4)

p_values_and_impr <- merge(summary_impr, all_p_values, by = c("WeekAhead", "Model"))

# Create a new column "Model_Type" based on model with and without log-back transformation
p_values_and_impr$Model_Type <- ifelse(grepl("_LB$", p_values_and_impr$Model), "LB", "no_LB")

# View the updated dataframe
head(p_values_and_impr)
```

```
##   WeekAhead       Model          m       sd     p_values Model_Type
## 1         1    AUTO_ADJ   6.168405 13.63084 3.691573e-02      no_LB
## 2         1 AUTO_ADJ_LB -11.653274 11.24692 4.749010e-07         LB
## 3         1  AUTO_AR_LB  -1.545609 10.84485 9.873984e-01         LB
## 4         1    AUTO_AVG   4.021081 14.15503 5.875037e-01      no_LB
## 5         1 AUTO_AVG_LB -14.140050 12.43459 7.296603e-08         LB
## 6         1    AUTO_EPI   6.990705  9.83982 3.691573e-02      no_LB
```

```
p_values_and_impr
```

```
##     WeekAhead       Model            m        sd     p_values Model_Type
## 1           1    AUTO_ADJ   6.16840507 13.630841 3.691573e-02      no_LB
## 2           1 AUTO_ADJ_LB -11.65327420 11.246917 4.749010e-07         LB
## 3           1  AUTO_AR_LB  -1.54560882 10.844847 9.873984e-01         LB
## 4           1    AUTO_AVG   4.02108114 14.155027 5.875037e-01      no_LB
## 5           1 AUTO_AVG_LB -14.14005046 12.434586 7.296603e-08         LB
## 6           1    AUTO_EPI   6.99070537  9.839820 3.691573e-02      no_LB
## 7           1 AUTO_EPI_LB  -2.80467855 10.568915 1.024409e-01         LB
## 8           1    AUTO_TMP  11.51666930 10.288213 3.639236e-06      no_LB
## 9           1 AUTO_TMP_LB   1.53339530 11.822086 1.000000e+00         LB
## 10          1    ES27_ADJ   8.52297479 13.864926 9.775586e-04      no_LB
## 11          1 ES27_ADJ_LB  -9.05339738 11.364071 5.184158e-04         LB
## 12          1     ES27_AR   8.30731524 10.118315 1.727131e-02      no_LB
## 13          1  ES27_AR_LB   0.23548066 11.776589 1.000000e+00         LB
## 14          1    ES27_AVG   5.47687113 14.380897 1.158616e-01      no_LB
## 15          1 ES27_AVG_LB -13.73335519 12.827612 1.348707e-06         LB
## 16          1    ES27_EPI   9.20670348 10.748825 9.639596e-03      no_LB
## 17          1 ES27_EPI_LB  -1.22895586 10.972461 1.000000e+00         LB
## 18          1    ES27_TMP   9.61266206  9.723429 4.383257e-04      no_LB
## 19          1 ES27_TMP_LB   3.40463753 11.867449 1.000000e+00         LB
## 20          1    ES64_ADJ  13.61809924 16.007447 5.480340e-06      no_LB
## 21          1 ES64_ADJ_LB  -8.64842457 11.276886 2.130091e-03         LB
## 22          1     ES64_AR  14.25965923 11.830388 2.114293e-06      no_LB
## 23          1  ES64_AR_LB   0.07153891 11.376863 1.000000e+00         LB
## 24          1    ES64_AVG  10.14548680 16.458098 1.253674e-03      no_LB
## 25          1 ES64_AVG_LB -13.38295786 12.636395 2.114293e-06         LB
## 26          1    ES64_EPI  15.39425464 12.496583 4.749010e-07      no_LB
## 27          1 ES64_EPI_LB  -1.04141048 10.998777 1.000000e+00         LB
## 28          1    ES64_TMP  17.11827782 13.113365 4.040353e-08      no_LB
## 29          1 ES64_TMP_LB   2.85879891 11.145874 1.000000e+00         LB
## 30          2    AUTO_ADJ   3.37776872 15.431786 2.729133e-01      no_LB
## 31          2 AUTO_ADJ_LB -16.93492506  8.684974 6.330936e-12         LB
## 32          2  AUTO_AR_LB  -7.99329058  7.090533 6.350568e-08         LB
## 33          2    AUTO_AVG   2.18534092 14.986603 1.400091e-01      no_LB
## 34          2 AUTO_AVG_LB -19.61900115  9.456536 3.677059e-11         LB
## 35          2    AUTO_EPI   7.09299296  7.916560 7.912913e-05      no_LB
## 36          2 AUTO_EPI_LB  -9.31291581  7.565246 4.469143e-08         LB
## 37          2    AUTO_TMP  14.12543239  9.365242 3.122125e-11      no_LB
## 38          2 AUTO_TMP_LB  -7.16266835  8.680066 2.483608e-05         LB
## 39          2    ES27_ADJ  11.61043436 15.665277 2.483608e-05      no_LB
## 40          2 ES27_ADJ_LB -14.01593553  8.771028 9.851107e-10         LB
## 41          2     ES27_AR  14.58742073 12.115838 2.350321e-07      no_LB
## 42          2  ES27_AR_LB  -4.69616409  9.859563 1.469551e-03         LB
## 43          2    ES27_AVG  10.19000000 17.773932 1.469551e-03      no_LB
## 44          2 ES27_AVG_LB -18.85109043  9.576852 5.968559e-13         LB
## 45          2    ES27_EPI  15.00584943 13.476077 3.914529e-06      no_LB
## 46          2 ES27_EPI_LB  -6.59477711  8.437866 1.138204e-05         LB
## 47          2    ES27_TMP  15.31761942 12.619315 1.079177e-07      no_LB
## 48          2 ES27_TMP_LB  -3.75124123  9.589851 8.833394e-03         LB
## 49          2    ES64_ADJ  18.17360204 20.558005 4.563432e-06      no_LB
## 50          2 ES64_ADJ_LB -14.27655560  8.569898 1.617622e-09         LB
```

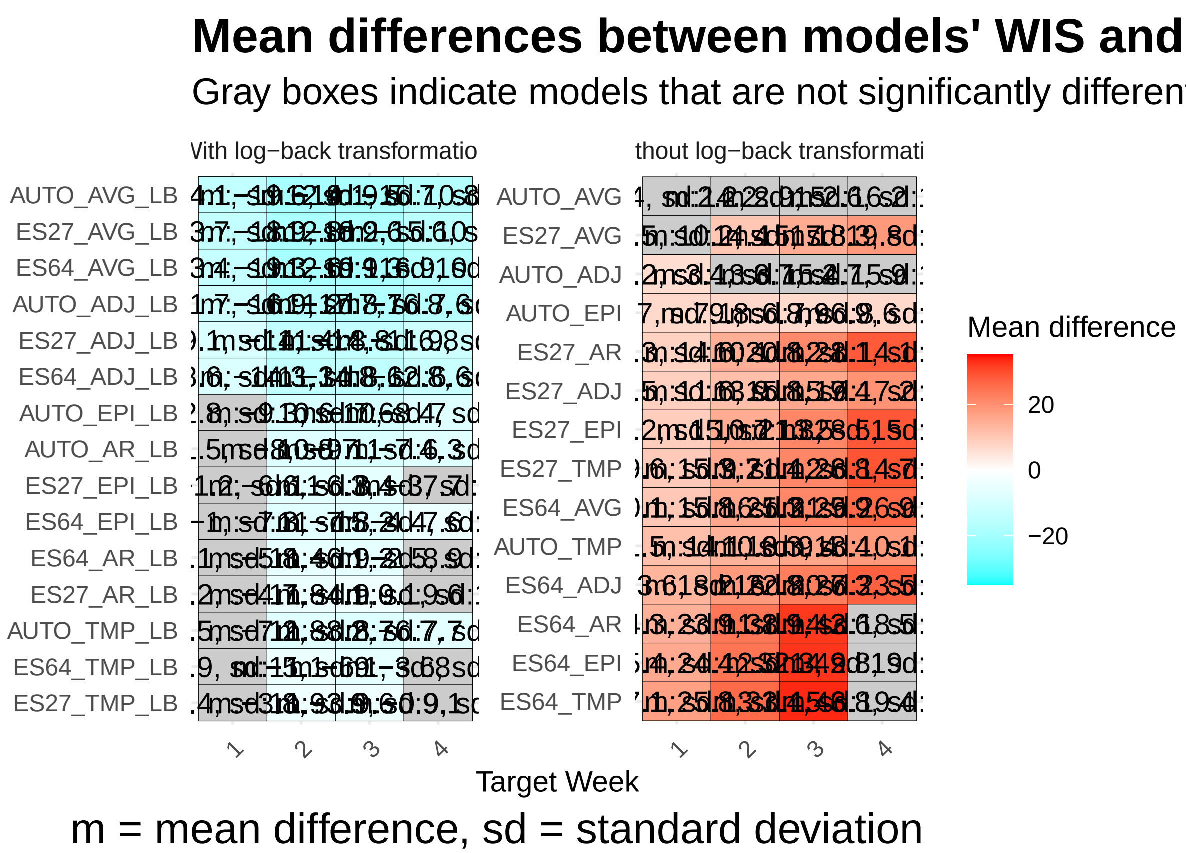
```
## 51           2     ES64_AR  23.88548718 14.264906 1.245297e-10      no_LB
## 52           2  ES64_AR_LB  -5.75846327  9.191161 2.128287e-04         LB
## 53           2    ES64_AVG  15.79183335 21.912398 4.899212e-05      no_LB
## 54           2 ES64_AVG_LB -19.34615942  9.348206 2.060574e-13         LB
## 55           2    ES64_EPI  24.36744609 14.919075 2.318217e-10      no_LB
## 56           2 ES64_EPI_LB  -7.25940037  8.235776 3.936805e-06         LB
## 57           2    ES64_TMP  25.83528103 15.565727 1.091394e-10      no_LB
## 58           2 ES64_TMP_LB  -5.02444384  9.010510 2.464577e-03         LB
## 59           3    AUTO_ADJ   3.69250265 15.921225 7.673054e-02      no_LB
## 60           3 AUTO_ADJ_LB -17.67912231  7.561750 2.060574e-13         LB
## 61           3  AUTO_AR_LB  -9.06435408  6.287092 9.745804e-11         LB
## 62           3    AUTO_AVG   2.90307762 16.188219 7.022822e-02      no_LB
## 63           3 AUTO_AVG_LB -19.06256167 10.847887 8.640200e-11         LB
## 64           3    AUTO_EPI   6.77267372  8.569299 2.448208e-04      no_LB
## 65           3 AUTO_EPI_LB  -9.97599714  6.999556 2.313527e-10         LB
## 66           3    AUTO_TMP  16.25788365 10.144435 3.240075e-12      no_LB
## 67           3 AUTO_TMP_LB  -8.23078658  7.735098 1.468905e-07         LB
## 68           3    ES27_ADJ  15.80846249 17.174119 3.387709e-08      no_LB
## 69           3 ES27_ADJ_LB -13.99253797  6.759889 1.563194e-11         LB
## 70           3     ES27_AR  20.79022126 14.076291 1.091962e-10      no_LB
## 71           3  ES27_AR_LB  -4.06018283  9.618860 1.284934e-03         LB
## 72           3    ES27_AVG  15.03334313 19.759282 1.873165e-06      no_LB
## 73           3 ES27_AVG_LB -18.24205023 10.015062 2.060574e-13         LB
## 74           3    ES27_EPI  21.32223783 15.017365 6.115641e-10      no_LB
## 75           3 ES27_EPI_LB  -6.26640084  7.722007 3.021601e-06         LB
## 76           3    ES27_TMP  21.43251307 14.744022 3.694822e-13      no_LB
## 77           3 ES27_TMP_LB  -3.85532996  9.134082 2.796021e-03         LB
## 78           3    ES64_ADJ  22.83776127 23.512217 1.012327e-08      no_LB
## 79           3 ES64_ADJ_LB -14.75052969  6.556387 5.158540e-12         LB
## 80           3     ES64_AR  31.91571042 18.505248 3.240075e-12      no_LB
## 81           3  ES64_AR_LB  -6.07746889  8.864474 7.926499e-05         LB
## 82           3    ES64_AVG  21.29460549 26.890525 2.505653e-07      no_LB
## 83           3 ES64_AVG_LB -19.12053981 10.030686 2.060574e-13         LB
## 84           3    ES64_EPI  32.30266686 18.980568 6.416201e-12      no_LB
## 85           3 ES64_EPI_LB  -7.51753987  7.645105 1.475448e-07         LB
## 86           3    ES64_TMP  33.36408070 19.433721 1.776357e-12      no_LB
## 87           3 ES64_TMP_LB  -6.11490155  8.030259 1.299593e-05         LB
## 88           4    AUTO_ADJ   2.72204591 16.388097 4.887000e-01      no_LB
## 89           4 AUTO_ADJ_LB -16.76946445  8.476175 7.027268e-12         LB
## 90           4  AUTO_AR_LB  -7.44397879  7.239350 4.615604e-07         LB
## 91           4    AUTO_AVG   2.63474133 16.599766 2.375250e-01      no_LB
## 92           4 AUTO_AVG_LB -16.68674838 12.658730 8.648282e-08         LB
## 93           4    AUTO_EPI   6.91840289  9.881631 2.204326e-03      no_LB
## 94           4 AUTO_EPI_LB  -8.35682186  7.558895 2.521838e-07         LB
## 95           4    AUTO_TMP  18.39330146 12.631438 1.455859e-09      no_LB
## 96           4 AUTO_TMP_LB  -6.70939645  8.620634 1.087519e-05         LB
## 97           4    ES27_ADJ  19.37319259 19.902542 1.033879e-08      no_LB
## 98           4 ES27_ADJ_LB -11.93229676  6.692316 7.236167e-11         LB
## 99           4     ES27_AR  28.12812650 15.456594 8.526513e-13      no_LB
## 100          4  ES27_AR_LB   0.12937403 12.682893 4.936689e-01         LB
## 101          4    ES27_AVG  18.34519402 22.212702 8.648282e-08      no_LB
## 102          4 ES27_AVG_LB -15.62216605 10.986420 7.027268e-12         LB
## 103          4    ES27_EPI  28.50318924 16.697998 1.044498e-11      no_LB
## 104          4 ES27_EPI_LB  -2.98488091  9.809813 1.255290e-01         LB
```

```
## 105         4    ES27_TMP  28.82059668 16.824034 2.060574e-13      no_LB
## 106         4 ES27_TMP_LB  -0.90600001 11.023080 4.936689e-01         LB
## 107         4    ES64_ADJ  27.33604896 28.576564 6.937825e-09      no_LB
## 108         4 ES64_ADJ_LB -12.83751285  6.601585 3.979039e-13         LB
## 109         4     ES64_AR  42.56949358 23.692103 3.979039e-13      no_LB
## 110         4  ES64_AR_LB  -2.48523863 11.647566 2.325804e-01         LB
## 111         4    ES64_AVG  25.88810897 32.764547 1.323035e-07      no_LB
## 112         4 ES64_AVG_LB -16.89580941 11.142586 1.044498e-11         LB
## 113         4    ES64_EPI  42.76233842 24.191151 3.979039e-13      no_LB
## 114         4 ES64_EPI_LB  -4.41017256 10.450558 2.204326e-03         LB
## 115         4    ES64_TMP  43.79860336 24.451177 3.979039e-13      no_LB
## 116         4 ES64_TMP_LB  -3.63694643  9.416580 1.033283e-01         LB
```

Here we are plotting the mean differences and standard deviation

```
# Get model order based based on lower values on WeekAhead == 1
model_order <- p_values_and_impr %>%
  filter(WeekAhead == 1) %>%
  arrange(desc(m)) %>%
  pull(Model)

# Prepare data with significance level, ordering, and model name
p_df <- p_values_and_impr %>%
  mutate(Significance = ifelse(p_values < 0.05 , "Significant", "Not Significant"),
         Model_Type = ifelse(grepl("_LB$", Model), "With log-back transformation", "Without log-back tra
         Model = factor(Model, levels = model_order))

# Heatmap plot
plot_m_heatmap <- ggplot(p_df, aes(x = factor(WeekAhead), y = Model, fill = m)) +
  geom_tile(aes(fill = ifelse(Significance == "Significant", m, NA)), color = "black") +  # Fill only s
  geom_tile(data = p_df %>% filter(Significance == "Not Significant"),
            aes(x = factor(WeekAhead), y = Model), fill = "gray80", color = "black") +  # Gray for non-
  geom_text(aes(label = paste0("m:", round(m, 1),", sd:", round(sd, 1))),
            color = "black", size = 4) +  # Add text labels
  scale_fill_gradient2(
  low = "cyan",
  mid = "white",
  high = "red",
  midpoint = 0,     # <-- Sets 0 as the midpoint
  na.value = "gray80",
  name = "Mean difference",
  limits = c(-35, 35)
)+
  labs(title = "Mean differences between models' WIS and the FluSight baseline across 48 states",
       subtitle = "Gray boxes indicate models that are not significantly different from the baseline (p
         caption = "m = mean difference, sd = standard deviation",
       x = "Target Week",
       y = "") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
  plot.caption = element_text(size = 16),
    plot.title = element_text(size = 18, face = "bold"),
    plot.subtitle = element_text(size = 14),) +
  facet_wrap(~ Model_Type, scales = "free_y")
```

```r
# Print plot
print(plot_m_heatmap)
```




```r
ggsave("Fig12.jpg", plot_m_heatmap, width = 14, height = 7 )
```

Now let's organize the data to plot some maps with the best models.

```r
# MAP 1 week ahead
W1_map <- data.frame(
  STATE = W1$STATE,
  percentage_improvement = W1_percentage_of_improvement$AUTO_AVG_LB,
  AUTO_AVG_LB=W1$AUTO_AVG_LB
)

# Include geometry
W1_map <- states %>%
  left_join(W1_map, by = c("STATE")) %>%
  drop_na()

# MAP 2 weeks a ahead
W2_map <- data.frame(
  STATE = W2$STATE,
```

```
  percentage_improvement = W2_percentage_of_improvement$AUTO_AVG_LB,
  AUTO_AVG_LB=W2$AUTO_AVG_LB
)

# Include geometry
W2_map <- states %>%
  left_join(W2_map, by = c("STATE"))%>%
  drop_na()

# MAP 3 weeks a ahead
W3_map <- data.frame(
  STATE = W3$STATE,
  percentage_improvement = W3_percentage_of_improvement$AUTO_AVG_LB,
  AUTO_AVG_LB=W3$AUTO_AVG_LB
)

# Include geometry
W3_map <- states %>%
  left_join(W3_map, by = c("STATE"))%>%
  drop_na()

# MAP 4 weeks ahead
W4_map <- data.frame(
  STATE = W4$STATE,
  percentage_improvement = W4_percentage_of_improvement$AUTO_AVG_LB,
  AUTO_AVG_LB=W4$AUTO_AVG_LB
)

# Include geometry
W4_map <- states %>%
  left_join(W4_map, by = c("STATE"))%>%
  drop_na()
```

Let's plot some maps with the with the mean difference (%) between the AUTO_AVG_LB and the AUTO ARIMA models for the same state.

1 week ahead percentage of improvement

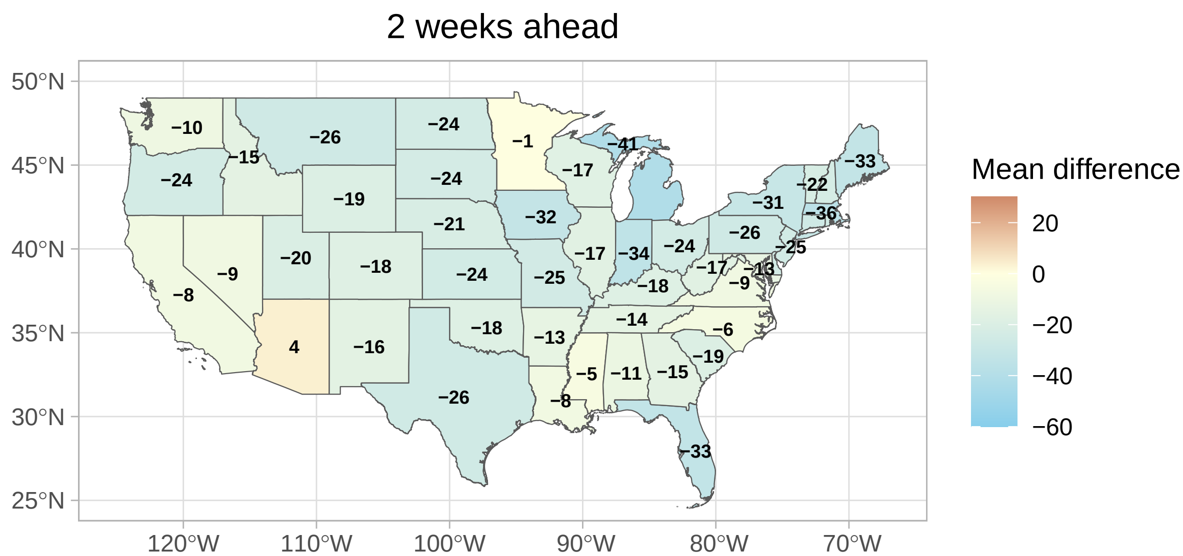
```
ES_1WEEK <- ggplot(W1_map) +
  geom_sf(aes(fill = percentage_improvement)) +
  scale_fill_gradient2(low ="skyblue" , mid = "lightyellow", high = "darkred", midpoint = 0, limits = c
  ggtitle("1 weeks ahead") +
  theme_light() +
  theme(legend.position = "right",
    plot.title = element_text(hjust = 0.5)) +
  geom_sf_text(data = W1_map, aes(label = round(percentage_improvement,0)),
               size = 2.6,
               color = "black",
               check_overlap = TRUE,fontface = "bold") +
  labs(
    fill = "Mean difference",
      x = "",
      y = ""
  )
```

```
x_limits <- c(-125, -67)  # Set the desired longitude range
y_limits <- c(25, 50)    # Set the desired latitude range

ES_1WEEK<-ES_1WEEK + coord_sf(xlim = x_limits, ylim = y_limits)
```

2 weeks ahead

```
ES_2WEEK <- ggplot(W2_map) +
  geom_sf(aes(fill = percentage_improvement)) +
  scale_fill_gradient2(low ="skyblue" , mid = "lightyellow", high = "darkred", midpoint = 0, limits = c
  ggtitle("2 weeks ahead") +
  theme_light() +
  theme(legend.position = "right",
    plot.title = element_text(hjust = 0.5)) +
  geom_sf_text(data = W2_map, aes(label = round(percentage_improvement,0)),
               size = 2.6,
               color = "black",
               check_overlap = TRUE,fontface = "bold") +
  labs(
    fill = "Mean difference",
       x = "",
       y = ""
  )  # Adding a subtitle


x_limits <- c(-125, -67)  # Set the desired longitude range
y_limits <- c(25, 50)    # Set the desired latitude range

ES_2WEEK + coord_sf(xlim = x_limits, ylim = y_limits)
```

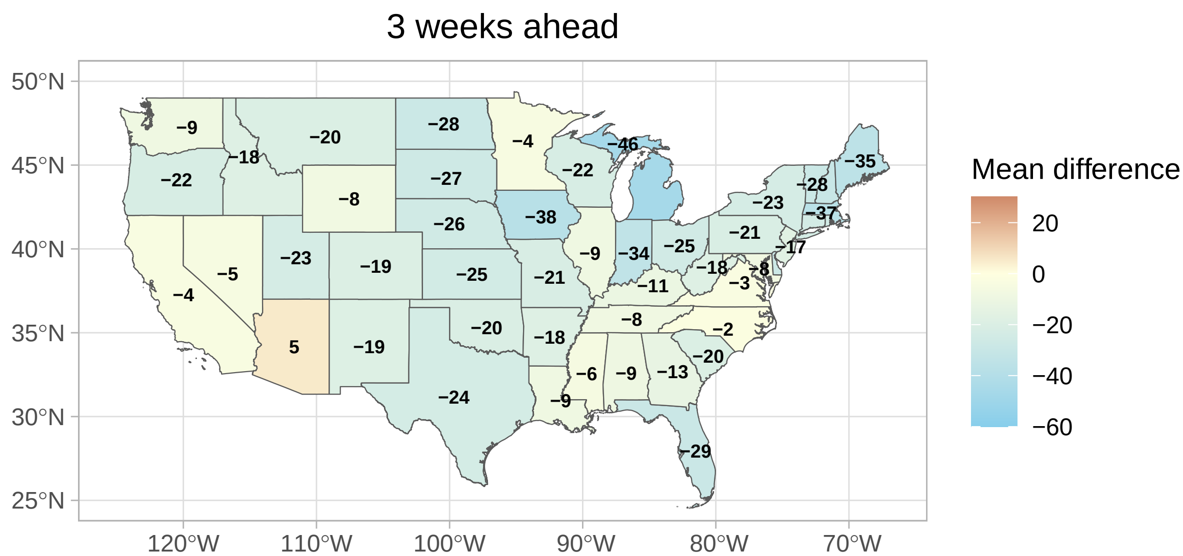
```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not
## give correct results for longitude/latitude data
```



3 weeks ahead

```
ES_3WEEK <- ggplot(W3_map) +
  geom_sf(aes(fill = percentage_improvement)) +  # Fill based on the best model
  scale_fill_gradient2(low ="skyblue" , mid = "lightyellow", high = "darkred", midpoint = 0, limits = c
  ggtitle("3 weeks ahead") +
  theme_light() +
  theme(legend.position = "right",
    plot.title = element_text(hjust = 0.5)) +
  geom_sf_text(data = W3_map, aes(label = round(percentage_improvement,0)),
               size = 2.6,
               color = "black",
               check_overlap = TRUE,fontface = "bold") +  # Display percentage improvement in each stat
  labs(
    fill =  "Mean difference",  # Label for the legend
       x = "",
       y = ""
  )  # Adding a subtitle

x_limits <- c(-125, -67)  # Set the desired longitude range
y_limits <- c(25, 50)     # Set the desired latitude range

ES_3WEEK + coord_sf(xlim = x_limits, ylim = y_limits)
```

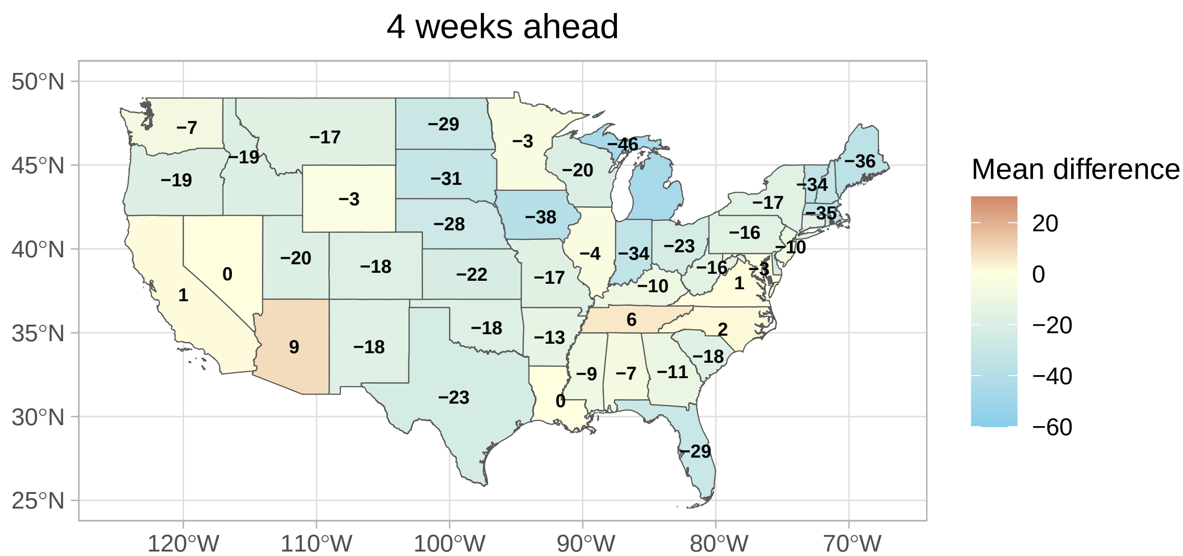
```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not
## give correct results for longitude/latitude data
```



4 weeks ahead

```
ES_4WEEK <- ggplot(W4_map) +
  geom_sf(aes(fill = percentage_improvement)) +  # Fill based on the best model
  scale_fill_gradient2(low ="skyblue" , mid = "lightyellow", high = "darkred", midpoint = 0, limits = c
  ggtitle("4 weeks ahead") +
  theme_light() +
  theme(legend.position = "right",
    plot.title = element_text(hjust = 0.5)) +
  geom_sf_text(data = W4_map, aes(label = round(percentage_improvement,0)),
               size = 2.6,
               color = "black",
               check_overlap = TRUE,fontface = "bold") +  # Display percentage improvement in each stat
  labs(
    fill = "Mean difference",  # Label for the legend
       x = "",
       y = ""
  )  # Adding a subtitle

x_limits <- c(-125, -67)  # Set the desired longitude range
y_limits <- c(25, 50)     # Set the desired latitude range

ES_4WEEK + coord_sf(xlim = x_limits, ylim = y_limits)
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not
## give correct results for longitude/latitude data
```



Combining plot on a 2x2 grid.

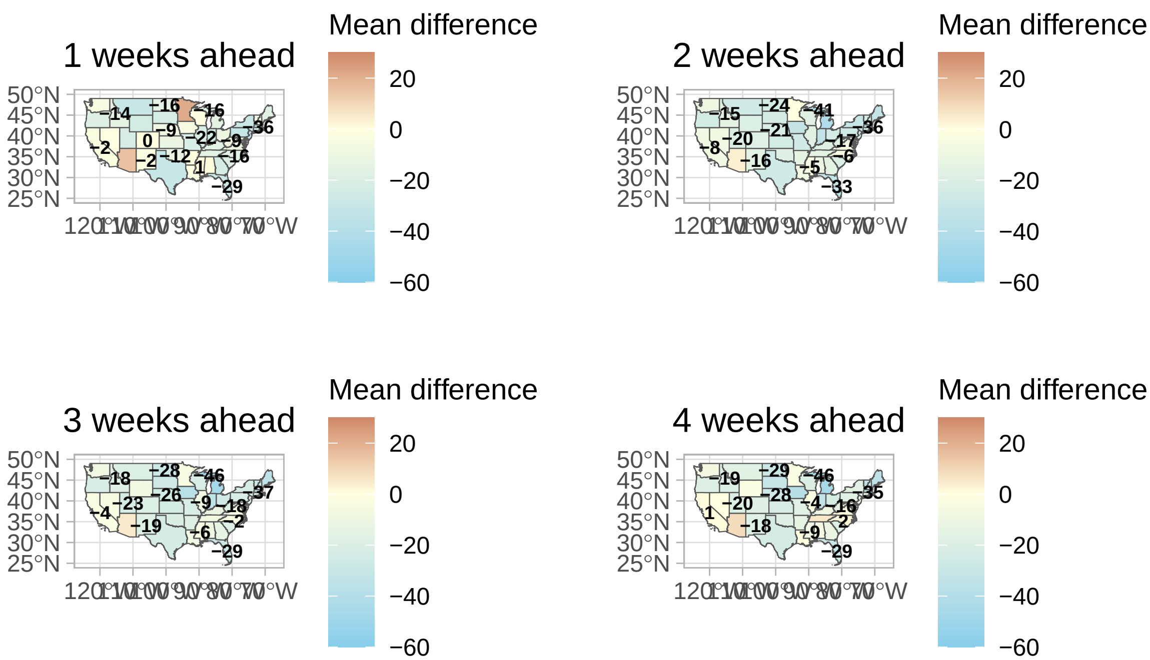
```
# Ensure each plot has coord_sf applied
ES_1WEEK <- ES_1WEEK + coord_sf(xlim = x_limits, ylim = y_limits)
```

```
## Coordinate system already present. Adding new coordinate system, which will
## replace the existing one.
```

```
ES_2WEEK <- ES_2WEEK + coord_sf(xlim = x_limits, ylim = y_limits)
ES_3WEEK <- ES_3WEEK + coord_sf(xlim = x_limits, ylim = y_limits)
ES_4WEEK <- ES_4WEEK + coord_sf(xlim = x_limits, ylim = y_limits)

# Combine in a 2x2 grid
combined_plot <- (ES_1WEEK | ES_2WEEK) /
                 (ES_3WEEK | ES_4WEEK) +
  plot_annotation(
    title = "",
    subtitle = "",
    theme = theme(plot.title = element_text(size = 16, face = "bold"),
                  plot.subtitle = element_text(size = 14))
  )

# Print
print(combined_plot)
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not
```

```
## give correct results for longitude/latitude data

## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not
## give correct results for longitude/latitude data

## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not
## give correct results for longitude/latitude data

## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not
## give correct results for longitude/latitude data
```




```r
# Save to file
ggsave("Fig13.jpg", combined_plot, width =12, height =7)
```

```
## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not
## give correct results for longitude/latitude data

## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not
## give correct results for longitude/latitude data

## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not
## give correct results for longitude/latitude data

## Warning in st_point_on_surface.sfc(sf::st_zm(x)): st_point_on_surface may not
## give correct results for longitude/latitude data
```

Loading datasets for regression models analysis

```
pop_data <- read.csv("models_with_logback/regression_features/population_data.csv") # resident populati
sovi_data <- read.csv("models_with_logback/regression_features/sovi_2010.2014.csv") # SOVI index
bric_data<-read.csv("models_with_logback/regression_features/bric2015.csv") # BRIC index
humidity_data<-read.csv("models_with_logback/regression_features/humidity_climatology_1990_2020.csv") #
temperature_data<-read.csv("models_with_logback/regression_features/temperature_climatology_1990_2020.c
```

REGRESSION MODELS

Now we will run the regression models for evaluating if the best model percentage of improvement in each state is related to given independent variables.

```
# List of data frames
WIS_dataframes <- list(W1 = W1_map, W2 = W2_map, W3 = W3_map, W4=W4_map)
regression_models<-data.frame()
```

Percentage of improvement regression analysis

AUTO_AVG_LB WIS x RESIDENT POPULATION 2020

```
for(i in c(1,2,3,4)){

  # Combining data for the same states
  WIS_pop_data <- inner_join(WIS_dataframes[[i]], pop_data, by = "STATE")
  # Fitting the regression model
  model <- lm((WIS_pop_data$percentage_improvement) ~ (WIS_pop_data$Resident_population_2020))
  # View the model summary
  model_summary <- summary(model)
  # Getting the R and p values for the plot
  r_squared <- round(model_summary$r.squared, 3)
  p_value <- signif(model_summary$coefficients[2, 4], 3)
  # Saving the results in a dataframe
  regression_models2 <- data.frame(
    independent_variable = "Resident Population (2020)",
    Week_Ahead = i,
    r_squared = r_squared,
    p_value = p_value
  )
  # Append to the main results dataframe
  regression_models <- rbind(regression_models, regression_models2)
}

regression_models
```

```
##          independent_variable Week_Ahead r_squared p_value
## 1 Resident Population (2020)          1     0.022   0.313
## 2 Resident Population (2020)          2     0.000   0.973
## 3 Resident Population (2020)          3     0.021   0.326
## 4 Resident Population (2020)          4     0.032   0.220
```

AUTO_AVG_LB WIS x POPULATION DENSITY

```r
for(i in c(1,2,3,4)){

  WIS_pop_data <- inner_join(WIS_dataframes[[i]], pop_data, by = "STATE")
  # Fit the regression model
  model <- lm((WIS_pop_data$percentage_improvement) ~ (WIS_pop_data$Population_density_2020))
  # View the model summary
  model_summary <- summary(model)
  # Extract R-squared and p-value
  r_squared <- round(model_summary$r.squared, 3)
  p_value <- signif(model_summary$coefficients[2, 4], 3)
  # saving the results in a dataframe
  regression_models2<-data.frame("independent_variable"="Population Density (2020)","Week_Ahead"=i, "r_s
  # Append to the main results dataframe
  regression_models<-rbind(regression_models,regression_models2)
}
```

Here we will weight the Social Vulnerability Index (SoVI) and Baseline Resilience Indicators for Communities (BRIC) county indexes which for each states based on the population size in each county.

```r
############################################################
# SOVI BY STATE
# weighted by population size in each county
sovi_by_state <- sovi_data %>%
  filter(!is.nan(sovi)) %>%  # Exclude rows where 'sovi' is NaN
  group_by(state.name) %>%
  summarize(weighted_mean = weighted.mean(sovi, w = population.2020, na.rm = TRUE))

colnames(sovi_by_state)[1] <- "STATE"

############################################################
# BRIC BY STATE
# weighted by population size in each county

bric_by_state <- bric_data %>%
  group_by(state.name) %>%
  summarize(across(16:22, ~ weighted.mean(.x, w = population.2020, na.rm = TRUE), .names = "weighted_mea
colnames(bric_by_state)[1] <- "STATE"
```

AUTO_AVG_LB WIS x SOVI

```r
for(i in c(1,2,3,4)){

  WIS_sovi<-NULL
  WIS_sovi <- inner_join(WIS_dataframes[[i]], sovi_by_state, by = "STATE")
  # Fit the regression model
  model <- lm((WIS_sovi$percentage_improvement) ~ (WIS_sovi$weighted_mean))
  # View the model summary
  model_summary <- summary(model)
  # Extract R-squared and p-value
  r_squared <- round(model_summary$r.squared, 3)
  p_value <- signif(model_summary$coefficients[2, 4], 3)
  # saving the results in a dataframe
  regression_models2<-data.frame("independent_variable"="Social Vulnerability Index (SoVI)","Week_Ahead
```

```
  # appending the results
  regression_models<-rbind(regression_models,regression_models2)
}
```

AUTO_AVG_LB WIS x BRIC SOCIAL

```
for(i in 1:4){
  WIS_bric <- inner_join(WIS_dataframes[[i]], bric_by_state, by = "STATE")
  # Fit the regression model
  model <- lm((WIS_bric$percentage_improvement) ~ (WIS_bric$weighted_mean_z_bric.social))
  # View the model summary
  model_summary <- summary(model)
  # Extract R-squared and p-value
  r_squared <- round(model_summary$r.squared, 3)
  p_value <- signif(model_summary$coefficients[2, 4], 3)
  # saving the results in a dataframe
  regression_models2<-data.frame("independent_variable"="BRIC Social (2015)","Week_Ahead"=i, "r_squared
  # appending the results
  regression_models<-rbind(regression_models,regression_models2)
}
```

AUTO_AVG_LB WIS x BRIC ECONOMIC

```
for(i in 1:4){
  WIS_bric <- inner_join(WIS_dataframes[[i]], bric_by_state, by = "STATE")
  # Fit the regression model
  model <- lm((WIS_bric$percentage_improvement) ~ (WIS_bric$weighted_mean_z_bric.economic))
  # View the model summary
  model_summary <- summary(model)
  # Extract R-squared and p-value
  r_squared <- round(model_summary$r.squared, 3)
  p_value <- signif(model_summary$coefficients[2, 4], 3)
  # saving the results in a dataframe
  regression_models2<-data.frame("independent_variable"="BRIC Economic (2015)","Week_Ahead"=i, "r_square
  regression_models<-rbind(regression_models,regression_models2)
}
```

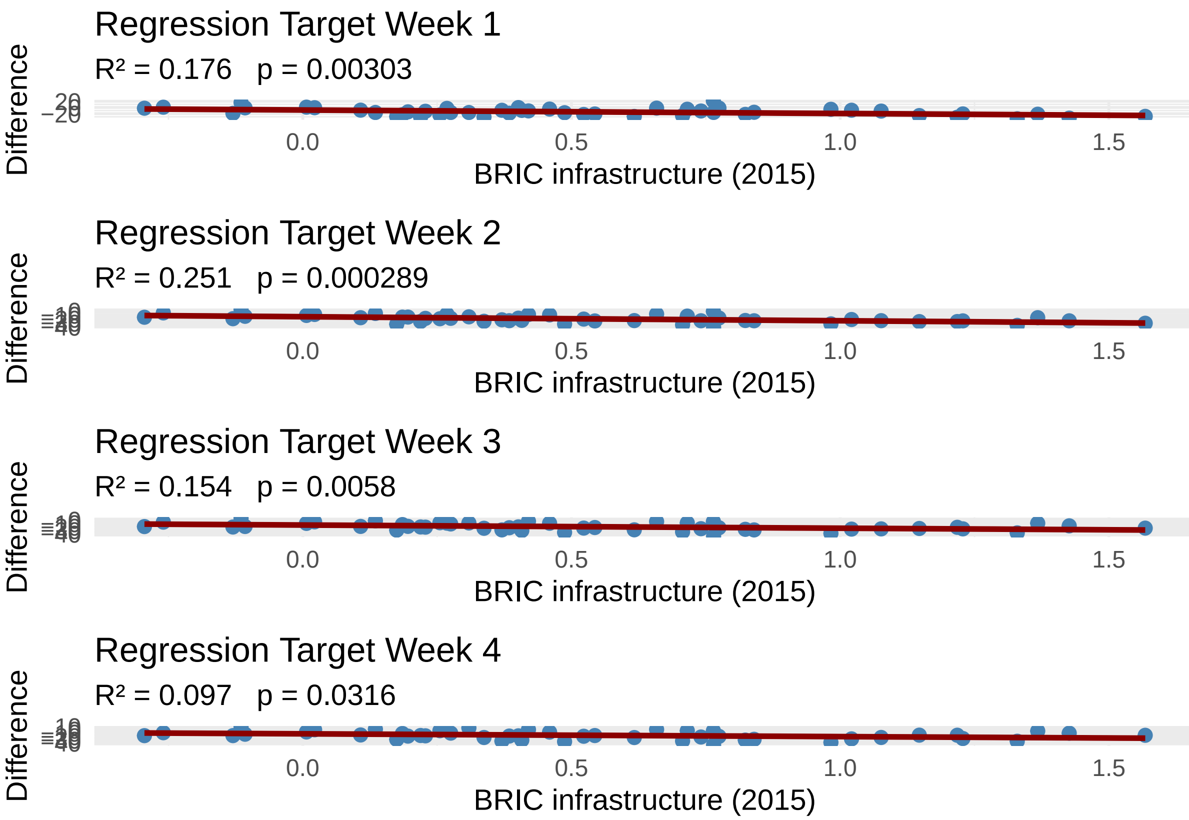
AUTO_AVG_LB WIS x BRIC Infrastructure

```
for (i in 1:4){
  WIS_bric <- inner_join(WIS_dataframes[[i]], bric_by_state, by = "STATE")
  # Fit the regression model
  model <- lm((WIS_bric$percentage_improvement) ~ (WIS_bric$weighted_mean_z_bric.infrastructure))
  # View the model summary
  model_summary <- summary(model)
  # Extract R-squared and p-value
  r_squared <- round(model_summary$r.squared, 3)
  p_value <- signif(model_summary$coefficients[2, 4], 3)
  # saving the results in a dataframe
  regression_models2<-data.frame("independent_variable"="BRIC Infrastructure (2015)","Week_Ahead"=i, "r_
  # appending results
  regression_models<-rbind(regression_models,regression_models2)
}
```

Plotting best regression model

```r
# Prepare a list of plots
bric_plots <- list()

for (i in 1:4) {
  # Join the WIS and BRIC data
  WIS_bric <- inner_join(
    WIS_dataframes[[i]],
    bric_by_state,
    by = "STATE"
  )

  # Fit the linear model
  model <- lm(
    percentage_improvement ~ weighted_mean_z_bric.infrastructure,
    data = WIS_bric
  )
  ms <- summary(model)

  # Extract R-squared and p-value
  r2 <- round(ms$r.squared, 3)
  p  <- signif(ms$coefficients[2, 4], 3)

  # Build the ggplot
  p <- ggplot(WIS_bric,
              aes(x = weighted_mean_z_bric.infrastructure,
                  y = percentage_improvement)) +
    geom_point(color = "steelblue", size = 2) +
    geom_smooth(method = "lm", se = FALSE, color = "darkred") +
    labs(
      title    = paste("Regression","Target Week", i),
      subtitle = paste0("R² = ", r2, "   p = ", p),
      x        = "BRIC infrastructure (2015)",
      y        = "Difference"
    ) +
    theme_minimal()

  # Store it
  bric_plots[[i]] <- p
}

# Combine the plots vertically
combined_regressions <- wrap_plots(bric_plots, ncol = 1)

# Display
combined_regressions
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



AUTO_AVG_LB WIS x BRIC institutional

```
for (i in 1:4){
  WIS_bric <- inner_join(WIS_dataframes[[i]], bric_by_state, by = "STATE")
  # Fit the regression model
  model <- lm((WIS_bric$percentage_improvement) ~ (WIS_bric$weighted_mean_z_bric.institutional))
  # View the model summary
  model_summary <- summary(model)
  # Extract R-squared and p-value
  r_squared <- round(model_summary$r.squared, 3)
  p_value <- signif(model_summary$coefficients[2, 4], 3)
  # saving the results in a dataframe
  regression_models2<-data.frame("independent_variable"="BRIC Institutional (2015)","Week_Ahead"=i, "r_
  # appending results
  regression_models<-rbind(regression_models,regression_models2)
}
```

AUTO_AVG_LB WIS x BRIC community

```
for(i in 1:4){
  WIS_bric <- inner_join(WIS_dataframes[[1]], bric_by_state, by = "STATE")
  # Fit the regression model
  model <- lm((WIS_bric$percentage_improvement) ~ (WIS_bric$weighted_mean_z_bric.community))
  # View the model summary
  model_summary <- summary(model)
  # Extract R-squared and p-value
```

```
  r_squared <- round(model_summary$r.squared, 3)
  p_value <- signif(model_summary$coefficients[2, 4], 3)
  # saving the results in a dataframe
  regression_models2<-data.frame("independent_variable"="BRIC Community (2015)","Week_Ahead"=i, "r_squa
  regression_models<-rbind(regression_models,regression_models2)
}
```

AUTO_AVG_LB WIS x BRIC environment

```
for(i in 1:4){
  WIS_bric <- inner_join(WIS_dataframes[[i]], bric_by_state, by = "STATE")
  # Fit the regression model
  model <- lm((WIS_bric$percentage_improvement) ~ (WIS_bric$weighted_mean_z_bric.environment))
  # View the model summary
  model_summary <- summary(model)
  # Extract R-squared and p-value
  r_squared <- round(model_summary$r.squared, 3)
  p_value <- signif(model_summary$coefficients[2, 4], 3)
  # saving the results in a dataframe
  regression_models2<-data.frame("independent_variable"="BRIC Environment (2015)","Week_Ahead"=i, "r_sq
  regression_models<-rbind(regression_models,regression_models2)
}
```

AUTO_AVG_LB WIS x BRIC total

```
for(i in 1:4){
  # BRIC INDEX
  WIS_bric <- inner_join(WIS_dataframes[[i]], bric_by_state, by = "STATE")
  # Fit the regression model
  model <- lm((WIS_bric$percentage_improvement) ~ (WIS_bric$weighted_mean_z_bric.total))
  # View the model summary
  model_summary <- summary(model)
  # Extract R-squared and p-value
  r_squared <- round(model_summary$r.squared, 3)
  p_value <- signif(model_summary$coefficients[2, 4], 3)
  # saving the results in a dataframe
  regression_models2<-data.frame("independent_variable"="BRIC total (2015)","Week_Ahead"=i, "r_squared"
  regression_models<-rbind(regression_models,regression_models2)
}
```

TEMPERATUE - ERA5

Now we will look at regression models that uses mean temperature and specific humidity.

AUTO_AVG_LB WIS x Temperature ERA5 Data

```
for(i in 1:4){
  # TEMPERATURE DATA from ERA5
  WIS_temp <- inner_join(WIS_dataframes[[i]], temperature_data, by = "STATE")
  # Fit the regression model
```

```r
  model <- lm((WIS_temp$percentage_improvement) ~ (WIS_temp$mean))
  # View the model summary
  model_summary <- summary(model)
  # Extract R-squared and p-value
  r_squared <- round(model_summary$r.squared, 3)
  p_value <- signif(model_summary$coefficients[2, 4], 3)
  # saving the results in a dataframe
  regression_models2<-data.frame("independent_variable"="Mean Temperature (1990-2020)","Week_Ahead"=i,
  # appending results
  regression_models<-rbind(regression_models,regression_models2)
}
```

AUTO_AVG_LB WIS x Specific Humidity ERA5 Data

```r
# regression results
print(regression_models)
```

```
##                 independent_variable Week_Ahead r_squared  p_value
## 1         Resident Population (2020)          1     0.022 0.313000
## 2         Resident Population (2020)          2     0.000 0.973000
## 3         Resident Population (2020)          3     0.021 0.326000
## 4         Resident Population (2020)          4     0.032 0.220000
## 5          Population Density (2020)          1     0.200 0.001420
## 6          Population Density (2020)          2     0.074 0.062200
## 7          Population Density (2020)          3     0.017 0.382000
## 8          Population Density (2020)          4     0.002 0.752000
## 9  Social Vulnerability Index (SoVI)          1     0.000 0.918000
## 10 Social Vulnerability Index (SoVI)          2     0.003 0.696000
## 11 Social Vulnerability Index (SoVI)          3     0.000 0.999000
## 12 Social Vulnerability Index (SoVI)          4     0.000 0.969000
## 13                BRIC Social (2015)          1     0.014 0.426000
## 14                BRIC Social (2015)          2     0.101 0.027700
## 15                BRIC Social (2015)          3     0.182 0.002490
## 16                BRIC Social (2015)          4     0.181 0.002590
## 17              BRIC Economic (2015)          1     0.001 0.829000
## 18              BRIC Economic (2015)          2     0.030 0.239000
## 19              BRIC Economic (2015)          3     0.084 0.046000
## 20              BRIC Economic (2015)          4     0.094 0.033700
## 21        BRIC Infrastructure (2015)          1     0.176 0.003030
## 22        BRIC Infrastructure (2015)          2     0.251 0.000289
## 23        BRIC Infrastructure (2015)          3     0.154 0.005800
## 24        BRIC Infrastructure (2015)          4     0.097 0.031600
## 25         BRIC Institutional (2015)          1     0.007 0.571000
## 26         BRIC Institutional (2015)          2     0.000 0.914000
## 27         BRIC Institutional (2015)          3     0.017 0.382000
## 28         BRIC Institutional (2015)          4     0.014 0.423000
## 29             BRIC Community (2015)          1     0.000 0.888000
## 30             BRIC Community (2015)          2     0.000 0.888000
## 31             BRIC Community (2015)          3     0.000 0.888000
## 32             BRIC Community (2015)          4     0.000 0.888000
## 33           BRIC Environment (2015)          1     0.009 0.530000
## 34           BRIC Environment (2015)          2     0.061 0.089300
## 35           BRIC Environment (2015)          3     0.135 0.010200
```

```
## 36           BRIC Environment (2015)          4     0.147 0.007120
## 37                 BRIC total (2015)          1     0.022 0.314000
## 38                 BRIC total (2015)          2     0.163 0.004450
## 39                 BRIC total (2015)          3     0.268 0.000164
## 40                 BRIC total (2015)          4     0.249 0.000305
## 41      Mean Temperature (1990-2020)          1     0.042 0.161000
## 42      Mean Temperature (1990-2020)          2     0.218 0.000826
## 43      Mean Temperature (1990-2020)          3     0.249 0.000308
## 44      Mean Temperature (1990-2020)          4     0.230 0.000569
```

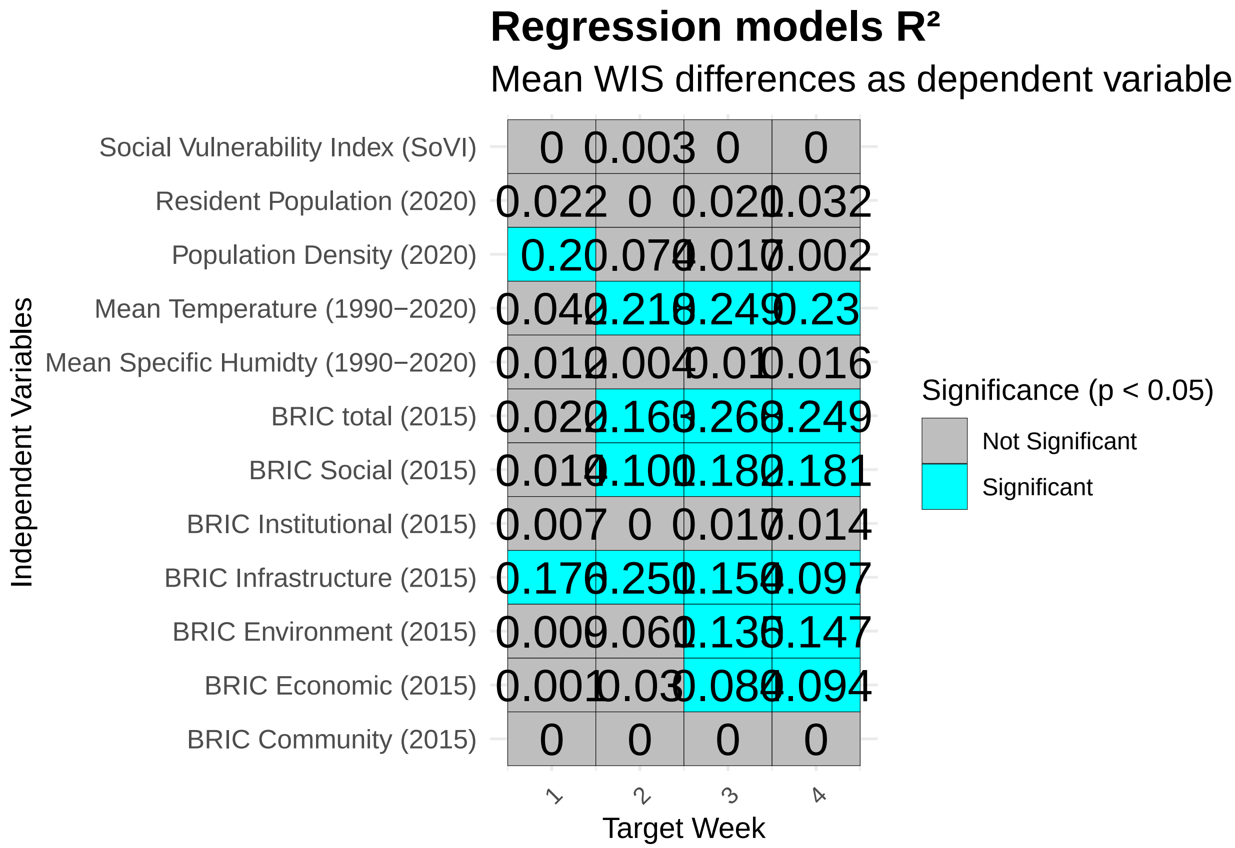
```r
for(i in 1:4){
  # HUMIDITY DATA from ERA5
  WIS_humidity <- inner_join(WIS_dataframes[[i]], humidity_data, by = "STATE")
  # Fit the regression model
  model <- lm((WIS_humidity$percentage_improvement) ~ (WIS_humidity$mean))
  # View the model summary
  model_summary <- summary(model)
  # Extract R-squared and p-value
  r_squared <- round(model_summary$r.squared, 3)
  p_value <- signif(model_summary$coefficients[2, 4], 3)
  # saving the results in a dataframe
  regression_models2<-data.frame("independent_variable"="Mean Specific Humidty (1990-2020)","Week_Ahead
  # appending results
  regression_models<-rbind(regression_models,regression_models2)
}

# Plot the table
regression_models
```

```
##                 independent_variable Week_Ahead r_squared  p_value
## 1         Resident Population (2020)          1     0.022 0.313000
## 2         Resident Population (2020)          2     0.000 0.973000
## 3         Resident Population (2020)          3     0.021 0.326000
## 4         Resident Population (2020)          4     0.032 0.220000
## 5          Population Density (2020)          1     0.200 0.001420
## 6          Population Density (2020)          2     0.074 0.062200
## 7          Population Density (2020)          3     0.017 0.382000
## 8          Population Density (2020)          4     0.002 0.752000
## 9  Social Vulnerability Index (SoVI)          1     0.000 0.918000
## 10 Social Vulnerability Index (SoVI)          2     0.003 0.696000
## 11 Social Vulnerability Index (SoVI)          3     0.000 0.999000
## 12 Social Vulnerability Index (SoVI)          4     0.000 0.969000
## 13                BRIC Social (2015)          1     0.014 0.426000
## 14                BRIC Social (2015)          2     0.101 0.027700
## 15                BRIC Social (2015)          3     0.182 0.002490
## 16                BRIC Social (2015)          4     0.181 0.002590
## 17              BRIC Economic (2015)          1     0.001 0.829000
## 18              BRIC Economic (2015)          2     0.030 0.239000
## 19              BRIC Economic (2015)          3     0.084 0.046000
## 20              BRIC Economic (2015)          4     0.094 0.033700
## 21        BRIC Infrastructure (2015)          1     0.176 0.003030
## 22        BRIC Infrastructure (2015)          2     0.251 0.000289
## 23        BRIC Infrastructure (2015)          3     0.154 0.005800
## 24        BRIC Infrastructure (2015)          4     0.097 0.031600
```

```
## 25         BRIC Institutional (2015)         1     0.007 0.571000
## 26         BRIC Institutional (2015)         2     0.000 0.914000
## 27         BRIC Institutional (2015)         3     0.017 0.382000
## 28         BRIC Institutional (2015)         4     0.014 0.423000
## 29             BRIC Community (2015)         1     0.000 0.888000
## 30             BRIC Community (2015)         2     0.000 0.888000
## 31             BRIC Community (2015)         3     0.000 0.888000
## 32             BRIC Community (2015)         4     0.000 0.888000
## 33           BRIC Environment (2015)         1     0.009 0.530000
## 34           BRIC Environment (2015)         2     0.061 0.089300
## 35           BRIC Environment (2015)         3     0.135 0.010200
## 36           BRIC Environment (2015)         4     0.147 0.007120
## 37                 BRIC total (2015)         1     0.022 0.314000
## 38                 BRIC total (2015)         2     0.163 0.004450
## 39                 BRIC total (2015)         3     0.268 0.000164
## 40                 BRIC total (2015)         4     0.249 0.000305
## 41      Mean Temperature (1990-2020)         1     0.042 0.161000
## 42      Mean Temperature (1990-2020)         2     0.218 0.000826
## 43      Mean Temperature (1990-2020)         3     0.249 0.000308
## 44      Mean Temperature (1990-2020)         4     0.230 0.000569
## 45 Mean Specific Humidty (1990-2020)         1     0.012 0.465000
## 46 Mean Specific Humidty (1990-2020)         2     0.004 0.688000
## 47 Mean Specific Humidty (1990-2020)         3     0.010 0.497000
## 48 Mean Specific Humidty (1990-2020)         4     0.016 0.391000
```

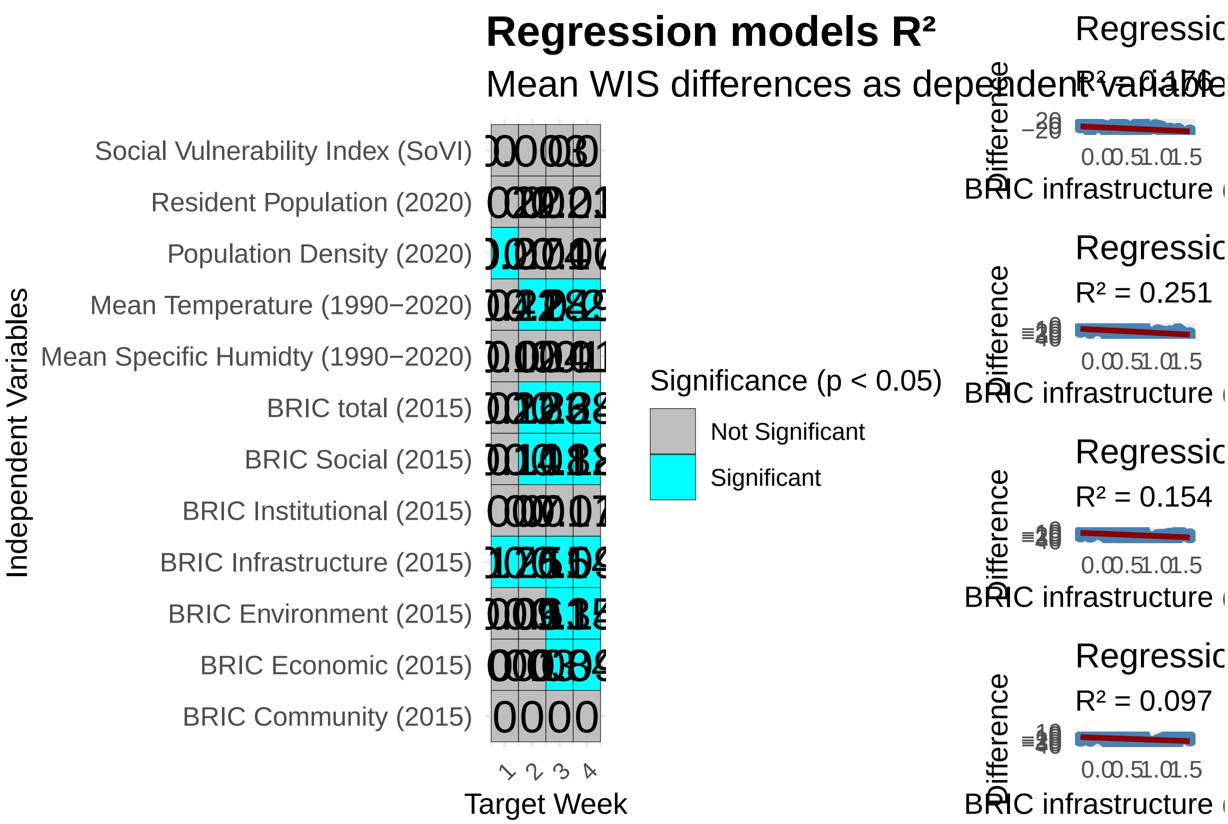
SUMMARY OF RESULTS

```r
# Define a significance threshold (p < 0.05)
regression_models <- regression_models %>%
  mutate(significance = ifelse(p_value < 0.05, "Significant", "Not Significant"))

# Plot
reg2<-ggplot(regression_models, aes(x = Week_Ahead, y = independent_variable, fill = significance)) +
  geom_tile(color = "black") +  # Add black borders to squares
  geom_text(aes(label = round(r_squared, 3)), color = "black", size = 6) +  # Text inside boxes
  scale_fill_manual(values = c("Significant" = "cyan", "Not Significant" = "gray")) +
  labs(title = "Regression models R²",
       subtitle = "Mean WIS differences as dependent variable",
       x = "Target Week",
       y = "Independent Variables",
       fill = "Significance (p < 0.05)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1),
        axis.text.y = element_text(size = 10),
        plot.title = element_text(size = 16, face = "bold"),
                  plot.subtitle = element_text(size = 14))

reg2
```




```r
reg2_scatter <- reg2 | combined_regressions
reg2_scatter
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```




```r
ggsave(reg2_scatter, height = 6, width = 12, filename="Fig14.jpg")
```

```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```

```r
#ggsave(reg2, filename="Fig9.jpg",height = 6, width = 8)
```

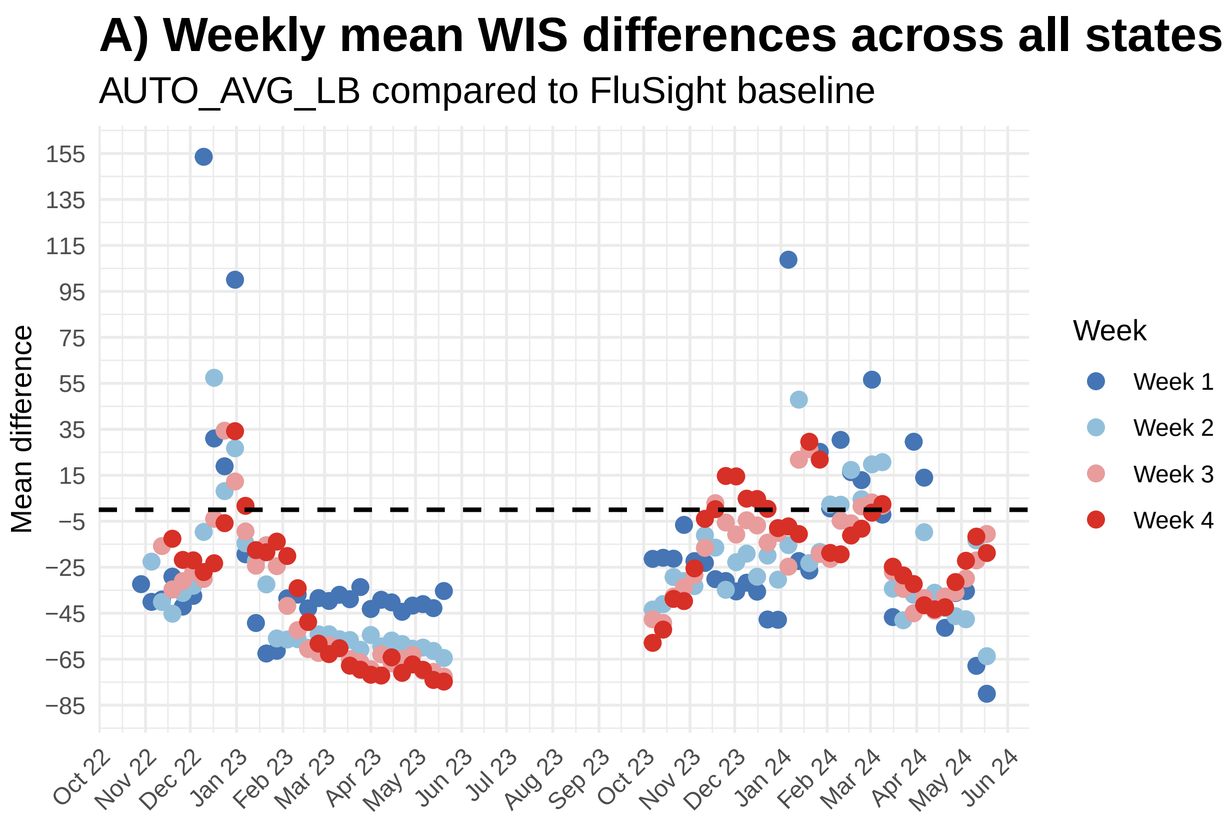
Here we will be computing and plotting mean WIS improvement by epidemiological weeks comparing the best model and the baseline model.

```r
# Create a list of data frames for the four weeks
weeks <- 1:4
results_by_epiweek_list <- list()

for (week in weeks) {
  # Merge the data frames for the current week
  combined_dataframes_ <- merge(get(paste0("filtered_df_W", week, "_NoLg")),
                        get(paste0("filtered_df_W", week)))

  # Sum values by Julian date across all states
  mean_improvement_ <- combined_dataframes_ %>%
    group_by(Julian_date) %>%
```

```
    summarize(
      AUTO_AR = mean(AUTO_AR, na.rm = TRUE),  # Mean across all states
      AUTO_AVG_LB = mean(AUTO_AVG_LB, na.rm = TRUE) # Mean across all states
    ) %>%
    # Calculate the difference ratio in percentage
    mutate(Difference =  ((AUTO_AVG_LB / AUTO_AR)-1)*100,
           Week = paste0("Week ", week))

  # Store the mean improvement by julian date
  results_by_epiweek_list[[week]] <- mean_improvement_
}

# Combine all weeks into a single data frame
results_by_epiweek_all <- bind_rows(results_by_epiweek_list)

# Plot the results
model_improv<-ggplot(results_by_epiweek_all, aes(x = Julian_date, y = (Difference), color = Week)) +
  geom_point(size = 2.5) +
  geom_hline(yintercept = 0, color = "black", linetype = "dashed", size = 0.8) +
  theme_minimal() +
  labs(
    title = "A) Weekly mean WIS differences across all states",
    subtitle = "AUTO_AVG_LB compared to FluSight baseline",
    x = "",
    y = "Mean difference"
  ) +
  scale_y_continuous(
    limits = c(-85, 155),
    breaks = seq(-85, 155, by = 20)
  ) +
  scale_x_date(
    date_breaks = "1 month",
    date_labels = "%b %y"
  ) +
scale_color_manual(
  values = c(
    "Week 1" = "#4575B4",  # Blue
    "Week 2" = "#91BFDB",  # Light blue
    "Week 3" = "#E89C9C",  # Light red
    "Week 4" = "#D73027"   # Red
  )
) +
  theme(
    plot.title = element_text(size = 18, face = "bold"),
    plot.subtitle = element_text(size = 14),
    axis.text.x = element_text(angle = 45, hjust = 1)
  )
```

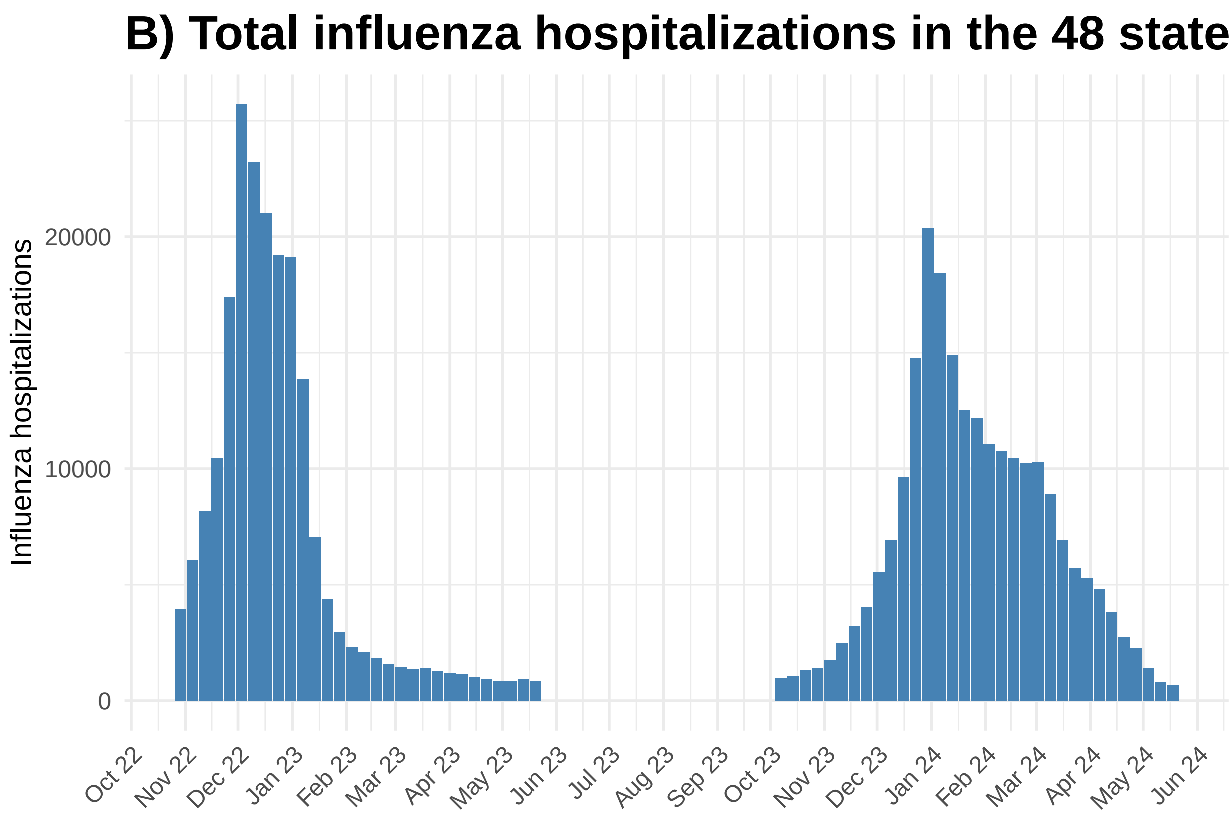
```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
model_improv
```



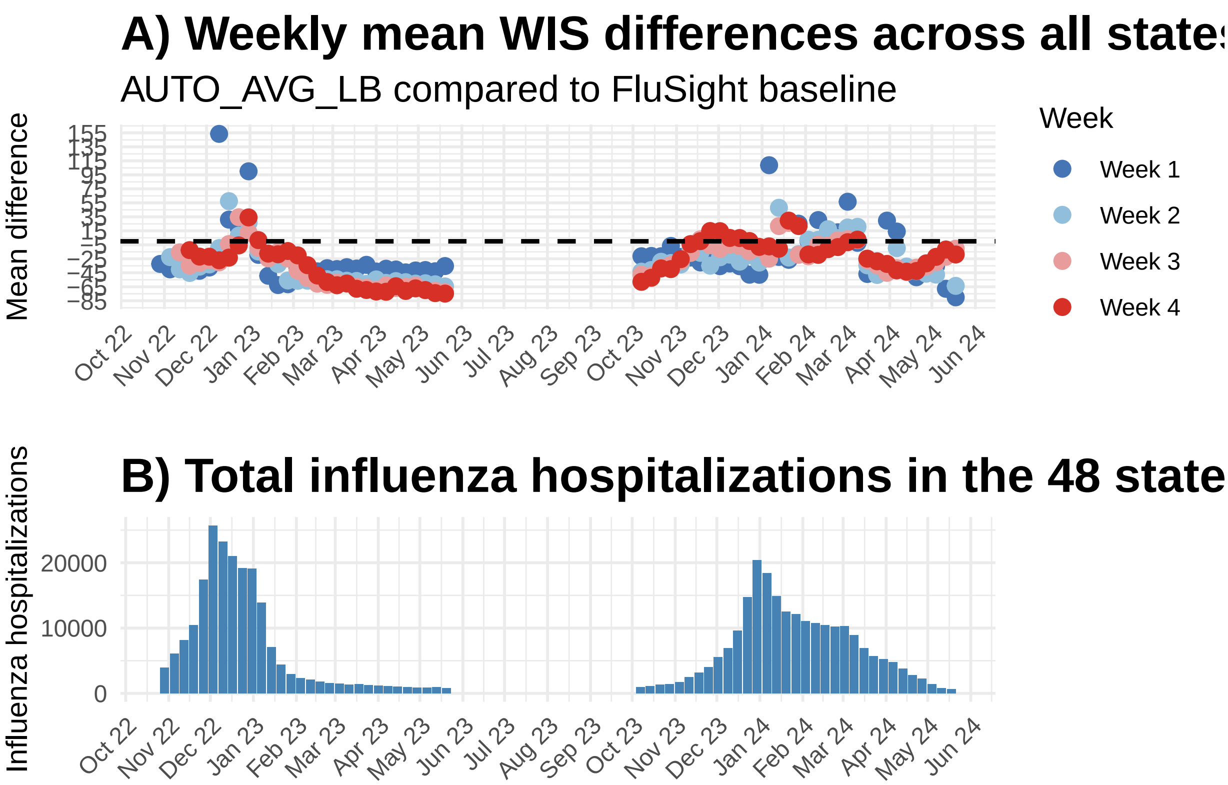
Total hospitalizations

```
# get total hospitalizations for the 48 states
total_hospitalizations_by_week <- AUTO_ADJACENT_WEEK1 %>%
  filter(epiweek >= 40 | epiweek <= 20) %>%
  group_by(target_end_date) %>%
  summarise(mean_cases = sum(cases, na.rm = TRUE))

# plotting results
hospital_plot <-ggplot(total_hospitalizations_by_week, aes(x = target_end_date, y = mean_cases)) +
  geom_bar(stat = "identity",fill = "steelblue") +
  labs(title = "B) Total influenza hospitalizations in the 48 states",
       x = "",
       y = "Influenza hospitalizations") +
  scale_x_date(
    date_breaks = "1 month",
    date_labels = "%b %y"
  ) +
  theme_minimal()+
  theme(plot.title = element_text(size = 18, face = "bold"),
    plot.subtitle = element_text(size = 14),
    axis.text.x = element_text(angle = 45, hjust = 1))

hospital_plot
```



Combining model improvement and total hospitalizations

```
# Combining results in a single plot
combined_plot <-  model_improv /hospital_plot
combined_plot
```




```r
# Save the combined plot
ggsave("Fig15.jpg", combined_plot, width = 7, height = 7, dpi = 600)
```