Machine Learning Foundations
Introduction to Statistics

Quantifying our
Confidence about Results and
Making Predictions of the Future

Jon Krohn, Ph.D.

jonkrohn.com/talks

github.com/jonkrohn/ML-foundations

Machine Learning Foundations
Introduction to Statistics

Slides: jonkrohn.com/talks

Code: github.com/jonkrohn/ML-foundations

Stay in Touch:

jonkrohn.com to sign up for email newsletter

linkedin.com/in/jonkrohn

jonkrohn.com/youtube

twitter.com/JonKrohnLearns

The Pomodoro Technique

Rounds of:

- 25 minutes of work
- with 5 minute breaks

Questions best handled at breaks, so save questions until then.

When people ask questions that have already been answered, do me a favor and let them know, politely providing response if appropriate.

Except during breaks, I recommend attending to this lecture only as topics are not discrete: Later material builds on earlier material.

POLL

What is your level of familiarity with Statistics?

- Little to no exposure
- Some understanding of the theory
- Deep understanding of the theory
- Deep understanding of the theory and experience applying statistical models with code

POLL

What is your level of familiarity with Machine Learning?

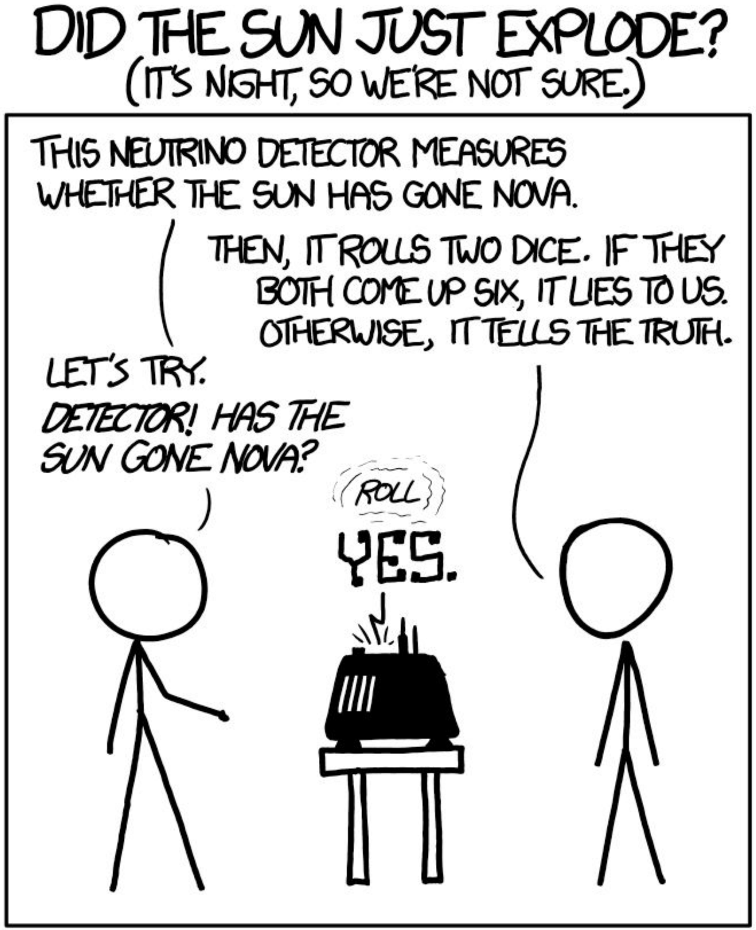
- Little to no exposure, or exposure to theory only
- Experience applying machine learning with code
- Experience applying machine learning with code and some understanding of the underlying theory
- Experience applying machine learning with code and strong understanding of the underlying theory

Introduction to Statistics

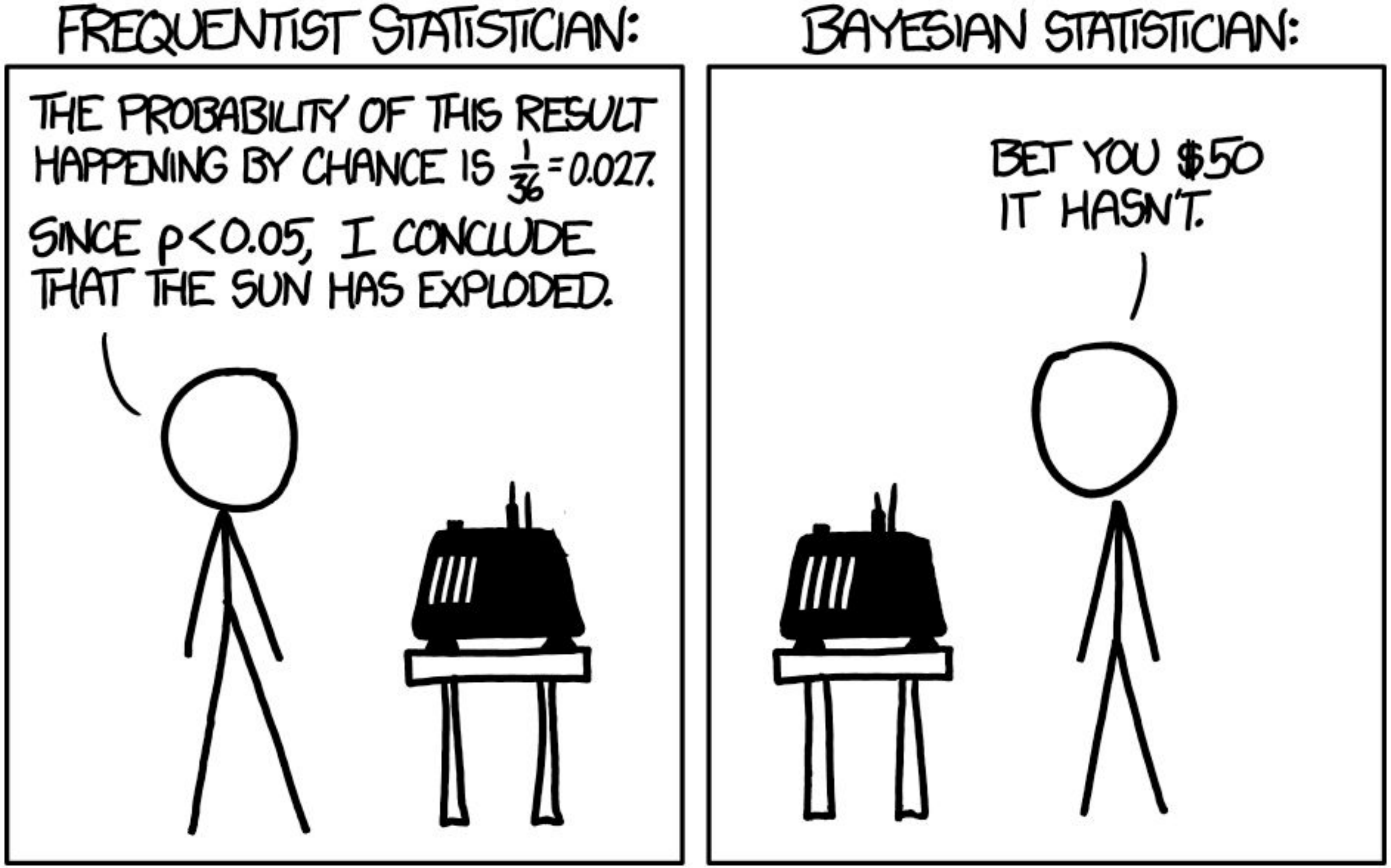
1. Frequentist Statistics
2. Regression
3. Bayesian Statistics

Segment 1: Frequentist Statistics

- Frequentist vs Bayesian Statistics
- Review of Relevant Probability Theory
- z-scores and Outliers
- p-values
- Comparing Means with t-tests
- Confidence Intervals
- ANOVA: Analysis of Variance
- Pearson Correlation Coefficient
- R-Squared Coefficient of Determination
- Correlation vs Causation
- Correcting for Multiple Comparisons



imgs.xkcd.com/comics/frequentists_vs_bayesians.png



imgs.xkcd.com/comics/frequentists_vs_bayesians.png

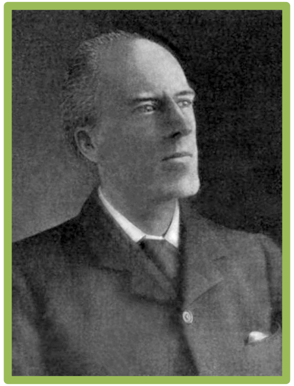
Bayesian Statistics

- Can incorporate prior knowledge
- “There’s 80% chance it’ll rain today.”
- **Thomas Bayes**
  - 1763: “Bayes’ theorem”
- **Pierre-Simon Laplace**
  - Late 18th / early 19th c.
- Drawbacks:
  - Beliefs are icky to some
  - Generally computationally expensive

Image in public domain

Frequentist Statistics

- "Objective" probabilities
- "On 100 days exactly like today, it would rain on 80 of them."
- Arbitrary "significance threshold"
- 1837: Siméon Denis Poisson
- 19th c.: expanded by, e.g., Mill, Venn, Boole
- 20th c.
  - (Sir) **R.A. Fisher**
  - (declined Sir) **Karl Pearson**
- Generally computationally inexpensive



Images in public domain

Applications of Stats to ML

- Examine data distributions (incl. outputs, prospective inputs)
    - Deepen understanding of your data
    - Identify irregularities
    - Reshape inputs toward standard normal
- Examine relationships between data
    - Guides modeling approach
- Compare model performances
- Ensure model isn't biased against particular demographic groups
- Bayesian stats has today become a type of ML used where:
    - Sample sizes tend to be not very large
    - Typically have evidence for priors (initial parameter values)

ML Foundations Series

Intro to Statistics **builds upon** and is **foundational for**:

1. Intro to Linear Algebra
2. Linear Algebra II: Matrix Operations
3. Calculus I: Limits & Derivatives
4. Calculus II: Partial Derivatives & Integrals
5. **Probability & Information Theory**
6. **Intro to Statistics**
7. Algorithms & Data Structures
8. Optimization

Probability Theory Review

- Measures of Central Tendency
- Measures of Dispersion
- Gaussian Distributions
- The Central Limit Theorem

Hands-on code demo: `6-statistics.ipynb`

Segment 2: Regression

- Features: Independent vs Dependent Variables
- Linear Regression to Predict Continuous Values
- Fitting a Line to Points on a Cartesian Plane
- Ordinary Least Squares
- Logistic Regression to Predict Categories
- (Deep) ML vs Frequentist Statistics

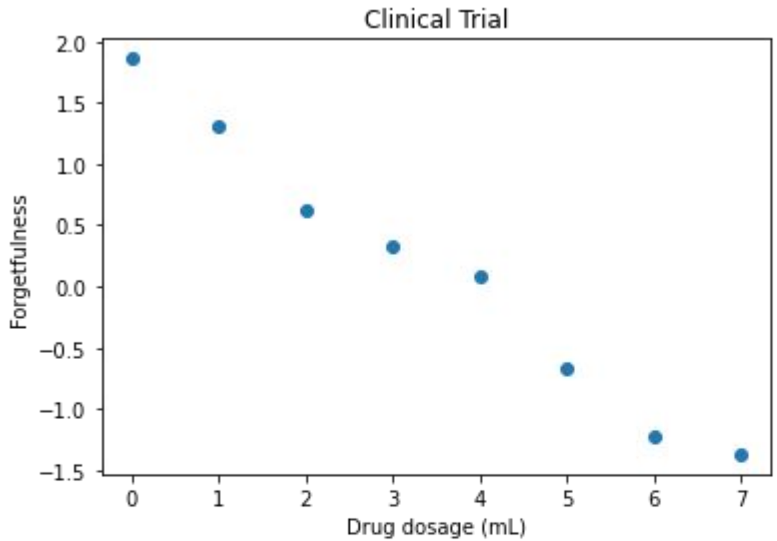
Independent vs Dependent Variables

Outcome:

- Dependent variable
- Typically denoted with *y*
- Cartesian vertical axis

Feature:

- *In*dependent variable
- May *predict* the outcome
- Typically denoted with *x*
- Cartesian horizontal axis
- Ideally can be explicitly adjusted, not only measured



Linear Regression for Continuous Values

house price

distance to school

there could be m features (many!)

$$y = a + b x_1 + c x_2 + \ldots + m x_m$$

"y-intercept"

number of bedrooms

$$y = a + bx_1 + cx_2 + \ldots + mx_m$$

$$\left[\begin{array}{c|ccccc}
y_1 & a & + bx_{1,1} & + cx_{1,2} & + \ldots & + mx_{1,m} \\
y_2 & a & + bx_{2,1} & + cx_{2,2} & + \ldots & + mx_{2,m} \\
\vdots & \vdots & \vdots & \vdots & & \vdots \\
y_n & a & + bx_{n,1} & + cx_{n,2} & + \ldots & + mx_{n,m}
\end{array}\right]$$

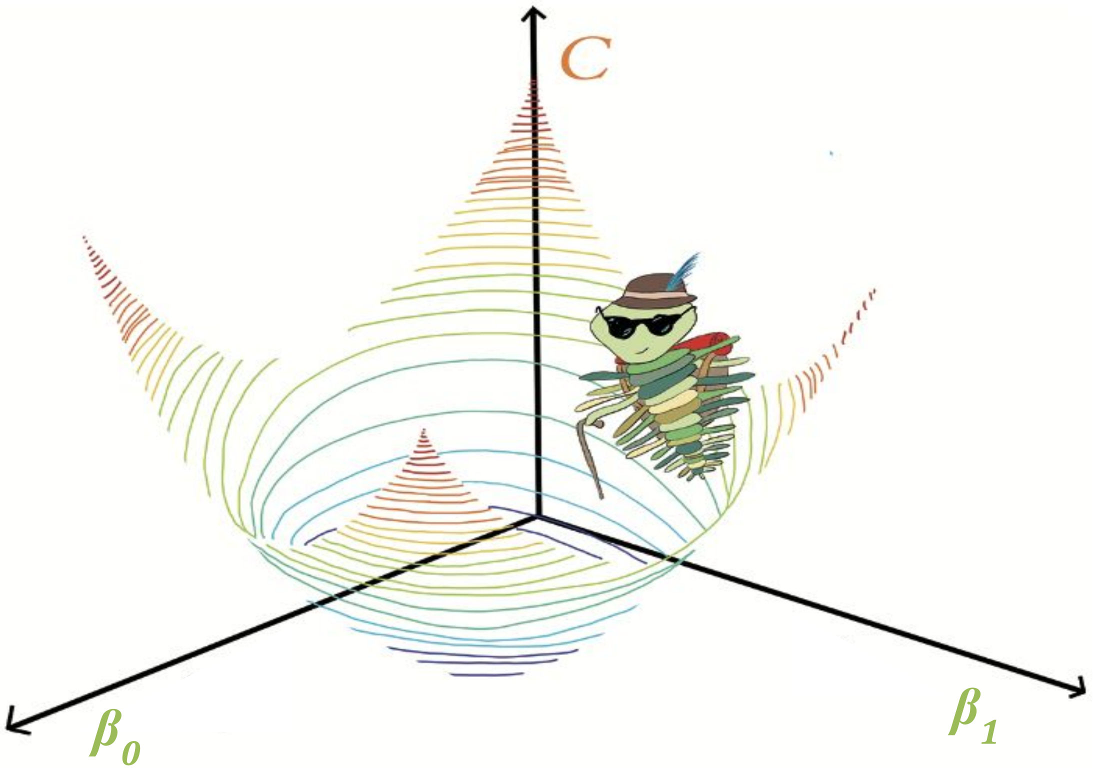
For any house i in the dataset,
y_i = price and $x_{i,1}$ to $x_{i,m}$ are its features.

We solve for parameters a, b, c to m

$$n \text{ cases tall} \left\{ \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \right. = \underbrace{\begin{bmatrix} 1 & x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ 1 & x_{2,1} & x_{2,2} & \cdots & x_{2,m} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,m} \end{bmatrix}}_{m \text{ features wide}} \begin{bmatrix} a \\ b \\ c \\ \vdots \\ m \end{bmatrix}$$

Hands-on code demo: `6-statistics.ipynb`

∇C: the Gradient of Cost (= Error)



Hands-on code demo

Matrix Inversion

In the equation $y = Xw$:

- We know the outcomes y
- We know the features X
- Vector w contains the unknowns, in this case β_0 and β_1

Assuming X^{-1} exists, matrix inversion can solve for X:

$$Xw = y$$

$$X^{-1}Xw = X^{-1}y$$

$$I_n w = X^{-1}y$$

$$w = X^{-1}y$$

Matrix Inversion

$$\begin{cases} 4b + 2c = 4 \\ -5b - 3c = -7 \end{cases}$$

$$X = \begin{bmatrix} X_{1,1} & X_{1,2} \\ X_{2,1} & X_{2,2} \end{bmatrix} = \begin{bmatrix} 4 & 2 \\ -5 & -3 \end{bmatrix} \qquad y = \begin{bmatrix} 4 \\ -7 \end{bmatrix}$$

$$w = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} b \\ c \end{bmatrix} = X^{-1} y$$

Hands-on code demo

(Deep) ML vs Frequentist Statistics

Primarily, I use (deep) ML to train my production algorithms.

However, **I regularly use frequentist statistics** to:

- Better understand training data
- Clean training data
    - Investigate/remove outliers
    - Transform toward standard normal with Box-Cox
- Make decisions with a quantitative degree of confidence w.r.t.:
    - Model hyperparameters
    - Model outputs
        - Where misclassifications occur
        - Whether there are unwanted biases
- Occasionally, train models with relatively few data and features

(Deep) ML vs Frequentist Statistics

In general, **ML** becomes necessary:

- When we have thousands of data points or more
    - SGD overcomes RAM / numerical computation constraints

In particular, **deep learning** enables us to:

- Handles many features (esp. large files: images, video, audio)
- Handle many outputs; exotic architectures / training strategies
- Automatically identify hierarchical, highly-abstract patterns
- Automatically fit interaction terms
- Automatically fit non-linear relationships

However: As we move from frequentist stats to ML, and particularly to deep learning, it can come at the cost of explainability / understanding.

Segment 3: Bayesian Statistics

- When to use Bayesian Statistics
- Prior Probabilities
- Bayes' Theorem
- PyMC3 Notebook
- Resources for Further Study

Bayesian Statistics

Older theory than Frequentist statistics:

- However, today is ML approach that scales to large datasets

Relative to Frequentist stats:

- No arbitrary (e.g., $\alpha = .05$) threshold, which can become pointless with many instances of data

Relative to other ML approaches:

- Typically smaller feature set, where we have *prior* information for some or all of the features

Prior Probabilities

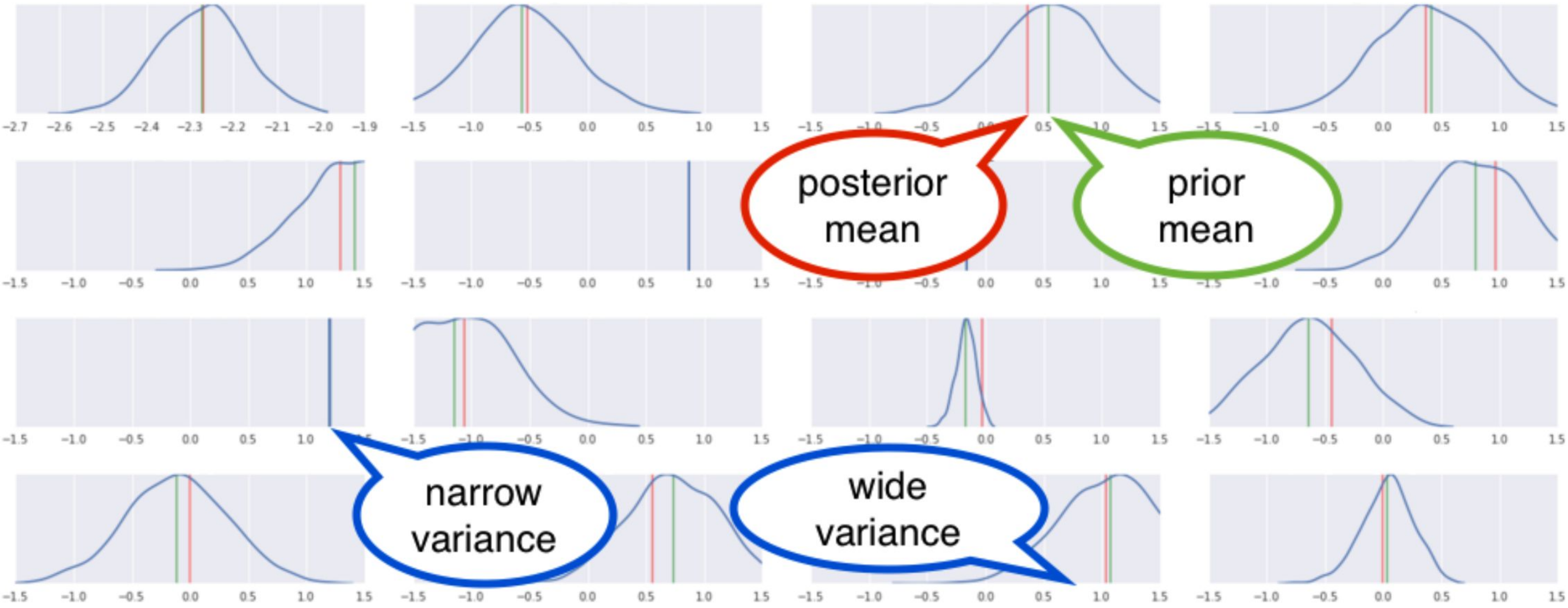
Can be obtained from:

- Observations, e.g., experiments
- Existing literature / knowledge
- Tangentially-related model results
- Reasoning
- Hunches

Can be allowed to move quite a bit so no need to stress.

Can be relatively fixed if you desire.

Can be relatively uninformative (e.g., sampled from uniform dist.)



jonkrohn.com/s/JSM_2016_official_proceedings.pdf

Hands-on code demo

Next Class: *Algos & Data Structures*

- Critical computer science for efficient ML / data science
- Big O Notation
- Most widely-used data structures, incl.:
  - Lists
  - Dictionaries
  - Tree- and graph-based structures
- Most important algorithms, incl. for:
  - Searching
  - Sorting
  - Hashing
  - Traversing graphs

Resources for Further Study

Next steps in the *ML Foundations* series:

- *Optimization*
  - SGD for regression through to deep learning
  - Avoiding overfitting

Books:

- Larry Wasserman's *All of Statistics* (free from Springer)
  - Concisely covers probability, Frequentist, and Bayesian stats
- E.T. Jaynes' *Probability Theory*

POLL *with Multiple Answers Possible*

What other topics interest you most?

- Linear Algebra
- Calculus
- Probability Theory
- More Frequentist Stats
- More Bayesian Stats
- Computer Science (e.g., algorithms, data structures)
- Machine Learning Basics
- Advanced Machine Learning, incl. Deep Learning
- Something Else

Stay in Touch

jonkrohn.com to sign up for email newsletter

linkedin.com/in/jonkrohn

jonkrohn.com/youtube

twitter.com/JonKrohnLearns

A4N
the
ARTIFICIAL
NEURAL
NETWORK
NEWS
NETWORK

GQR

PLACEHOLDER
FOR:

5-Minute Timer

PLACEHOLDER
FOR:

10-Minute Timer

PLACEHOLDER
FOR:

15-Minute Timer