Workout & Heart Rate Analysis

Paola Calle

Contents

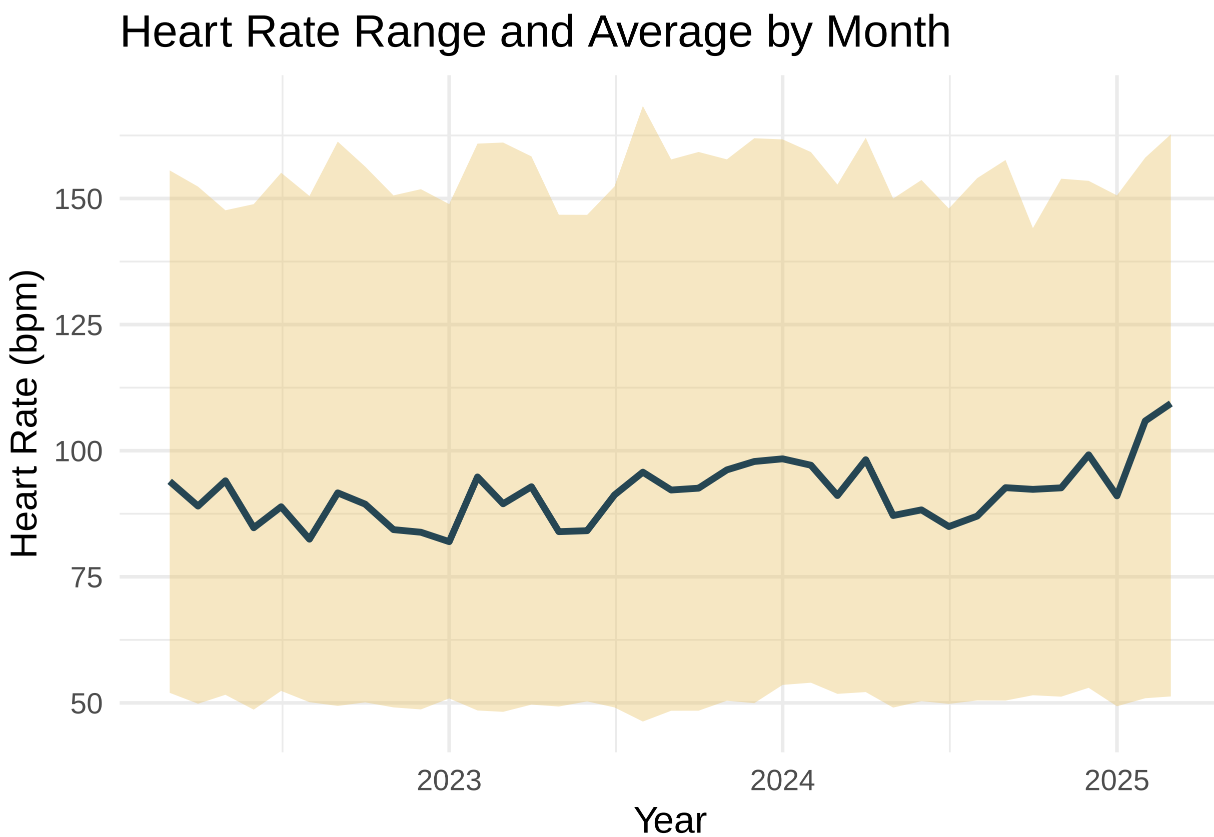
The Data

This analysis uses personal health data collected from Apple Health via Apple Watch, spanning from 2022 to the present. The dataset includes daily and monthly summaries of various activity and biometric metrics automatically recorded by Apple's health tracking ecosystem.

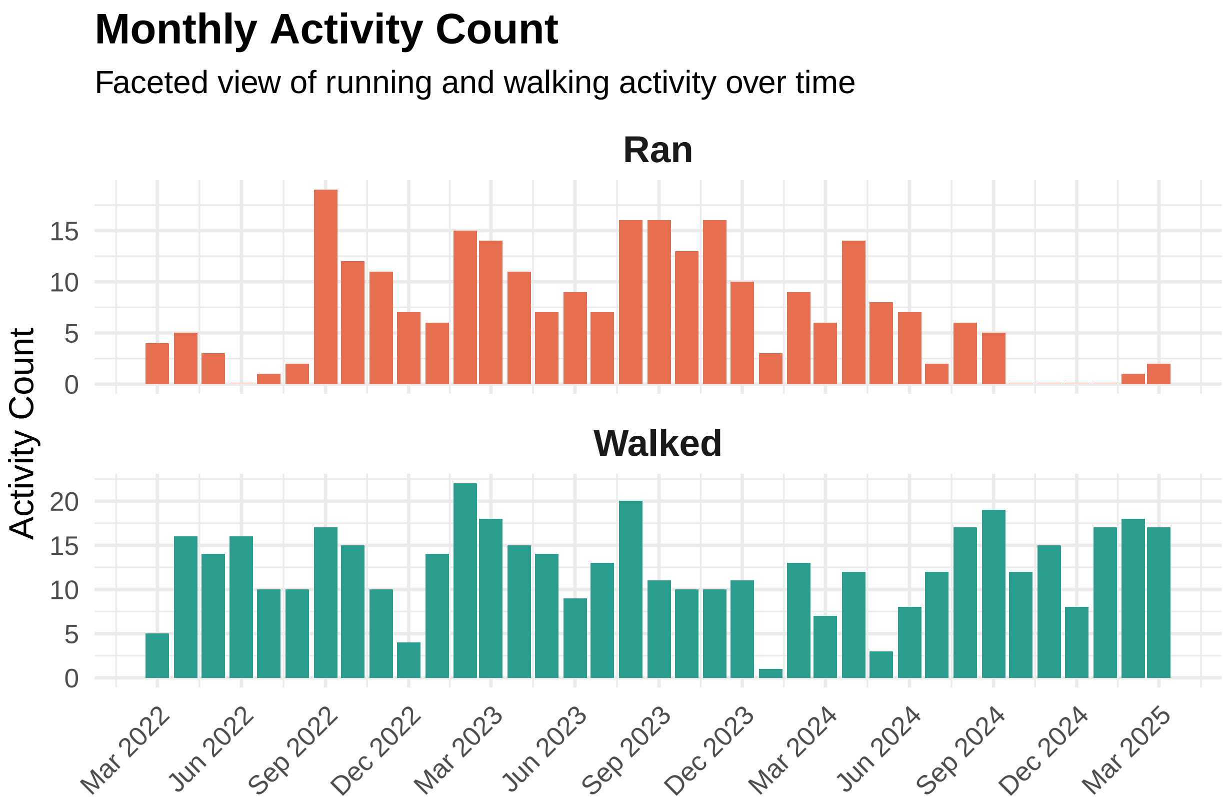
Intial Graphs

Heart Rate Across Time

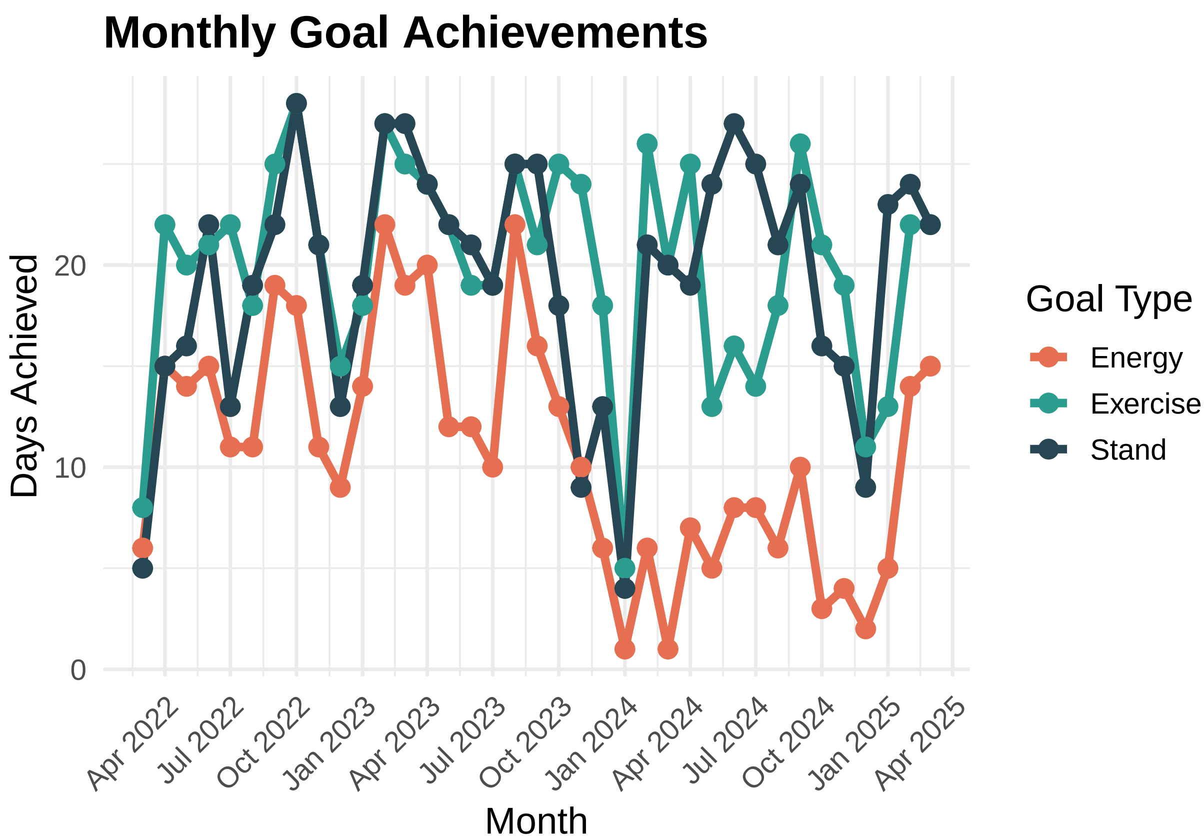
The plot shows monthly heart rate trends from 2022 to early 2025. Average heart rate stays mostly stable around 80–100 bpm, with a noticeable spike in early 2025. The shaded area shows a consistent range between low and high heart rates each month.


Heart Rate Range and Average by Month
Heart Rate (bpm)
150
125
100
75
50
2023
2024
2025
Year

Times Ran and Walked Across Time



Number of days each goal was achieved per month



Analaysis 1: Did I work out that day?

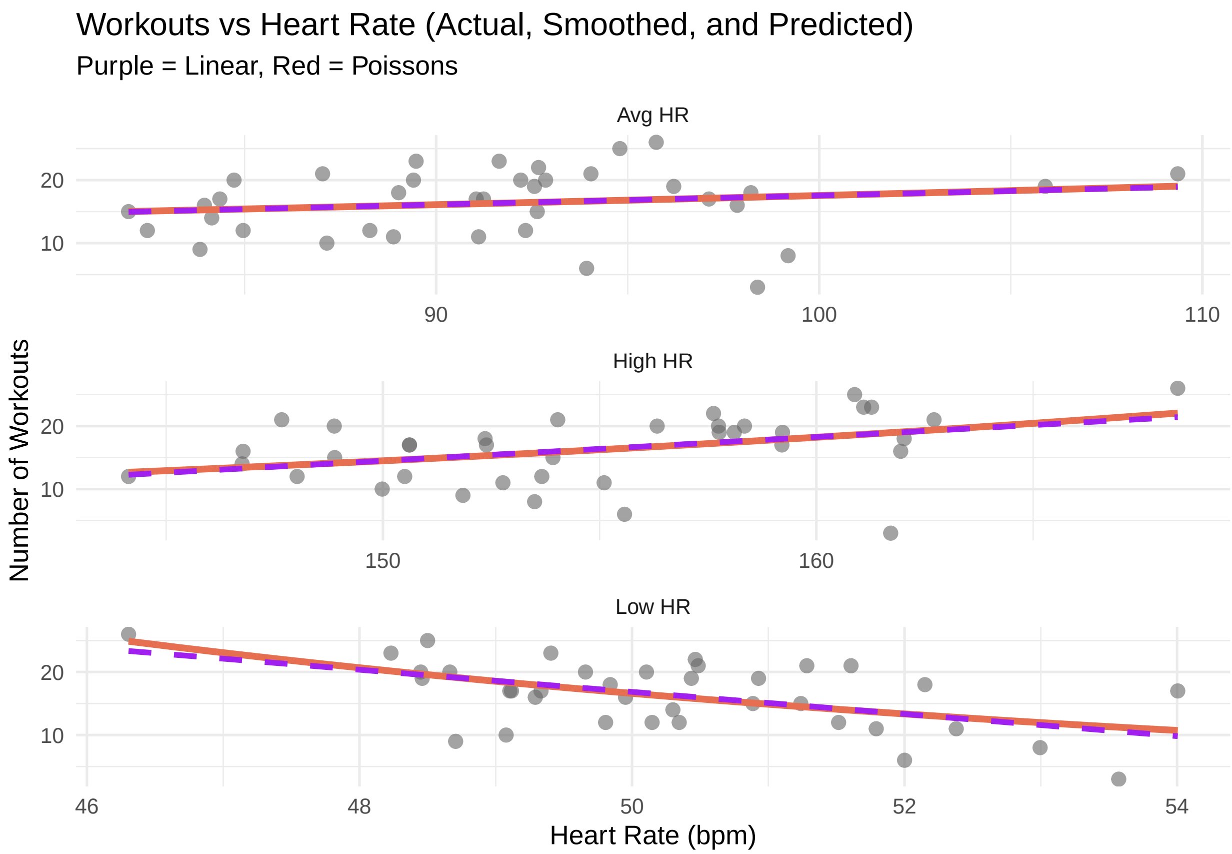
Poisson (Predicting Count of Workout Days Across Months) Vs. Linear Model

The Poisson (red) and linear (purple) lines are nearly identical across all three heart rate types. Because the Poisson line closely follows the linear regression line, we can conclude:

- The relationship between heart rate and workout count is well captured using a Poisson model, which is statistically more appropriate for counts.
- There's no major non-linearity or overdispersion visible that would make the Poisson model clearly inferior or inappropriate.

```
low_model <- glm(num_worked_out ~ low_heart_rate, family = poisson(link = "log"),
                 data = parsed_monthly_summary)
high_model <- glm(num_worked_out ~ high_heart_rate, family = poisson(link = "log"),
                  data = parsed_monthly_summary)
avg_model <- glm(num_worked_out ~ avg_heart_rate, family = poisson(link = "log"),
                 data = parsed_monthly_summary)
```

```r
low_lm_model <- lm(num_worked_out ~ low_heart_rate, data = parsed_monthly_summary)
high_lm_model <- lm(num_worked_out ~ high_heart_rate, data = parsed_monthly_summary)
avg_lm_model <- lm(num_worked_out ~ avg_heart_rate, data = parsed_monthly_summary)
```



Tests

Table 1: Model Summary Table

model	beta	SE	z_score	z_squared	p_value	deviance_explained
Low HR	-0.109	0.026	4.213	17.751	0.00003	17.951
High HR	0.023	0.007	3.127	9.780	0.00176	9.742
Avg HR	0.009	0.007	1.320	1.743	0.18673	1.722

Interpretation of Results So...

Low Heart Rate (Resting HR)

- Strongest predictor (highest z, lowest p-value, most deviance explained)
- Negative beta = Lower resting HR is associated with more workouts

- A 1 bpm decrease in resting HR increases expected workouts by ~10% (e^-0.109 0.897 = ~10% reduction in workouts per 1 bpm increase)

High HR adds value but to a lesser degree.

- Also significant (p ~ .0018)
- Positive beta = Higher peak HR is associated with more workouts
- Each 1 bpm increase in high HR predicts a ~2.3% increase in workout days.

Average Heart Rate

- Not significant (p = 0.187)
- Possibly too noisy or generic a measure to reflect true activity behavior

Logistic Binary (Predicting Whether I Worked Out on a Given Day)

```
low_model <- glm(worked_out ~ low_heart_rate, family = binomial(), data = parsed_clean_by_day)
high_model <- glm(worked_out ~ high_heart_rate, family = binomial(), data = parsed_clean_by_day
avg_model <- glm(worked_out ~ day_avg_heart_rate, family = binomial(), data = parsed_clean_by_d
```

model	beta	SE	z_score	p_value	odds_ratio
Low HR	0.011	0.015	0.730	0.46538	1.011
High HR	0.054	0.004	14.512	0.00000	1.056
Avg HR	0.090	0.006	14.108	0.00000	1.094

Average Heart Rate

- Has the strongest effect on predicting workout likelihood.
- Every 1 bpm increase in avg HR is associated with a 9.4% increase in the odds of having worked out that day.
- Highly statistically significant ($p < 0.0001$), indicating a robust relationship.

High Heart Rate

- Also a significant predictor.
- Every 1 bpm increase in high HR increases the odds of working out by 5.6%.
- Likely reflects workout intensity – a higher max HR is a strong indicator that I engaged in physical effort.

Low Heart Rate

- Not statistically significant (p = 0.465).
- Small beta (0.011) and an odds ratio close to 1 suggest no reliable link between low HR (resting) and working out on that specific day.
- May be more useful for modeling long-term fitness, rather than day-level behavior.

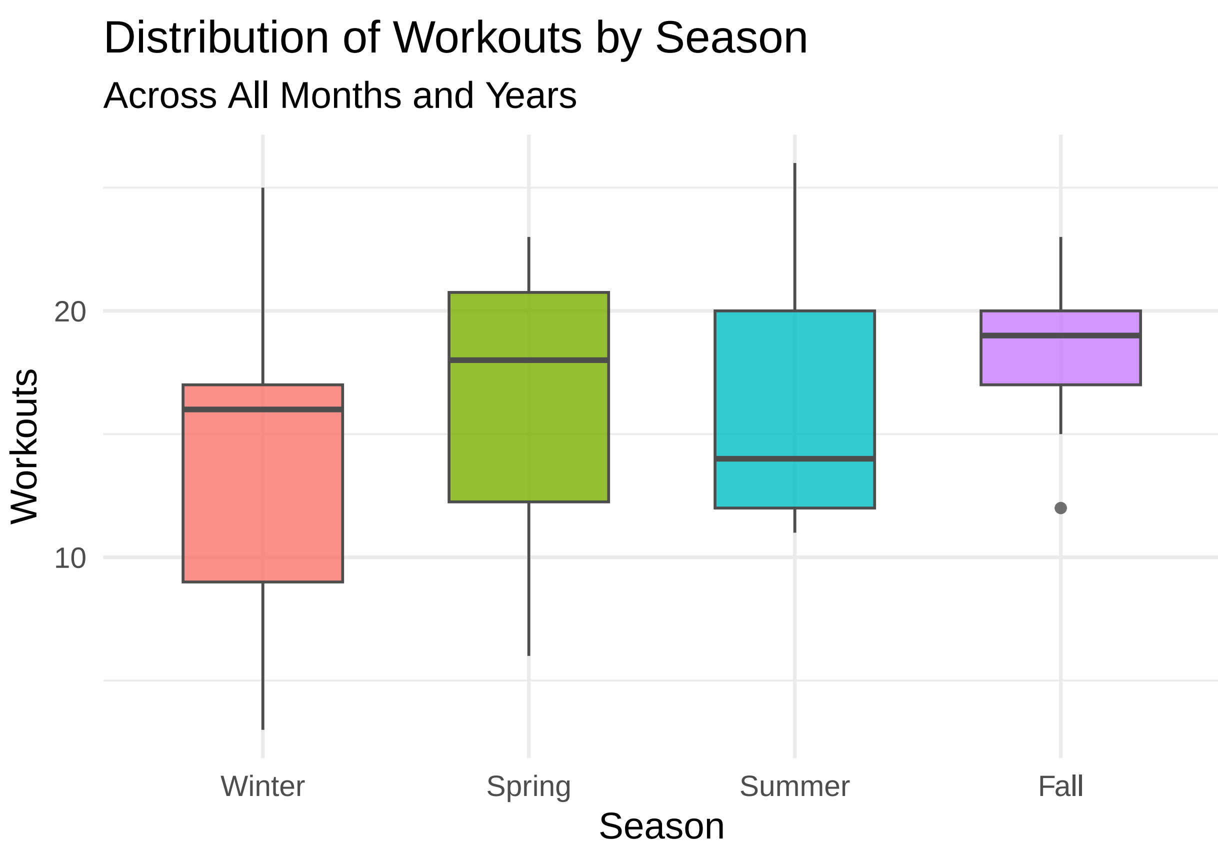
Analaysis 2: Seasons

Count of Workouts per Month

This boxplot shows how my monthly workout counts vary across seasons, summarizing data over multiple years.

Table 3: Summary of Workout Behavior by Season

Season	Activity Level	Notes
Spring	High & consistent	Strong, steady routine
Summer	Variable	Some very active, some low months
Fall	High & consistent	Likely a return to routine
Winter	Lower	Less frequent workouts, higher drop-off



Poisson Regression: Can It Predict My Monthly Workout Count... Based on Season?

```
poisson_season_model <- glm(
  num_worked_out ~ season,
```

```r
  data = parsed_monthly_summary,
  family = poisson(link = "log")
)

# summary(poisson_season_model)
```

Table 4: Coefficients Table: Poisson Regression of Workouts by Season

Term	Estimate	p-value	Interpretation
Intercept	2.66259	< 0.001	Baseline: Winter. Exp(2.66) ~ 14.3 workouts/month in winter.
Spring	0.13469	0.252	Not statistically significant. Spring may slightly increase workouts, but we're not confident.
Summer	0.11692	0.334	Also not significant. Summer doesn't strongly differ from winter in workout counts.
Fall	0.25818	0.028	Statistically significant! Fall months have higher workout counts compared to winter (about 29% more, exp(0.258) ~ 1.29).

Model Fit

- Null deviance: 70.58
- Residual deviance: 65.65
- Chi-squared test p-value: 0.1766

This means that while fall stands out, season overall is not a strong predictor of workout count across all months and years.

Logistic Regression — Daily Probability of Working Out

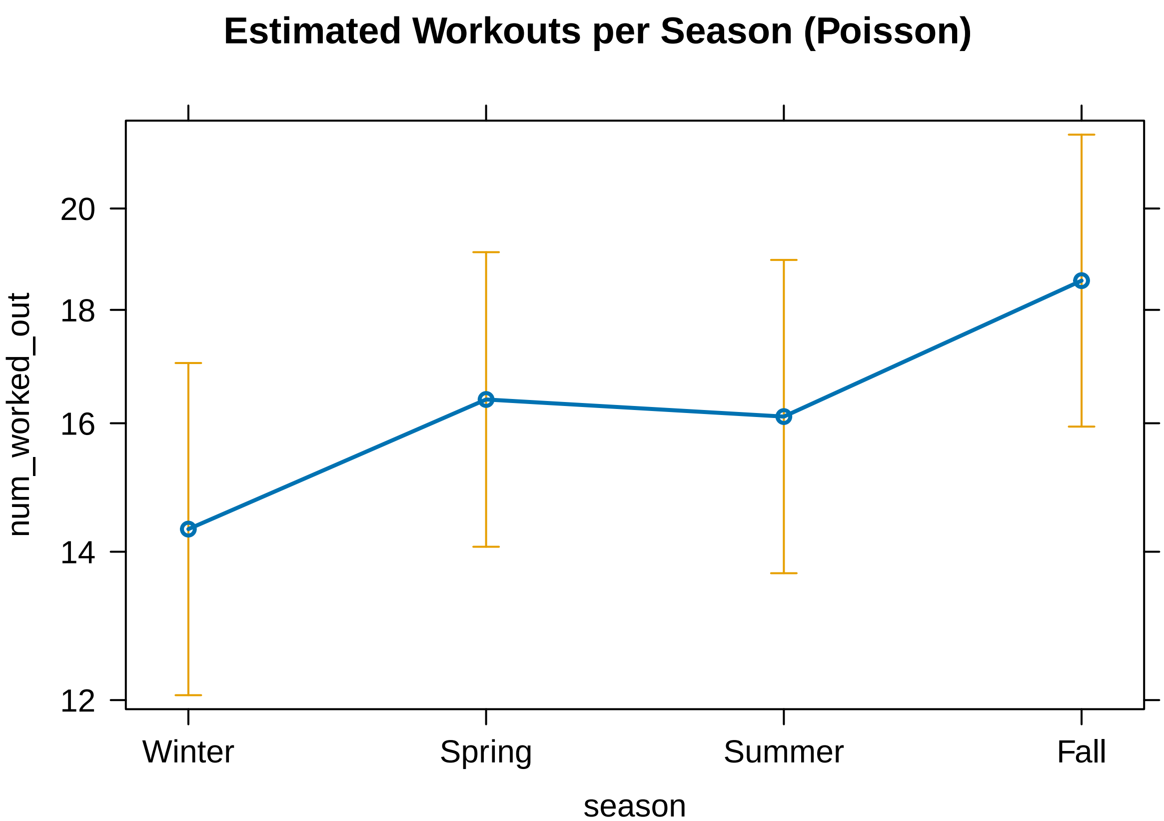
```r
# Binary outcome: worked out or not
logit_season_model <- glm(
  worked_out ~ season,
  data = parsed_clean_by_day,
  family = binomial()
)
```

Table 5: Logistic Regression Coefficients: Predicting Daily Workout from Season

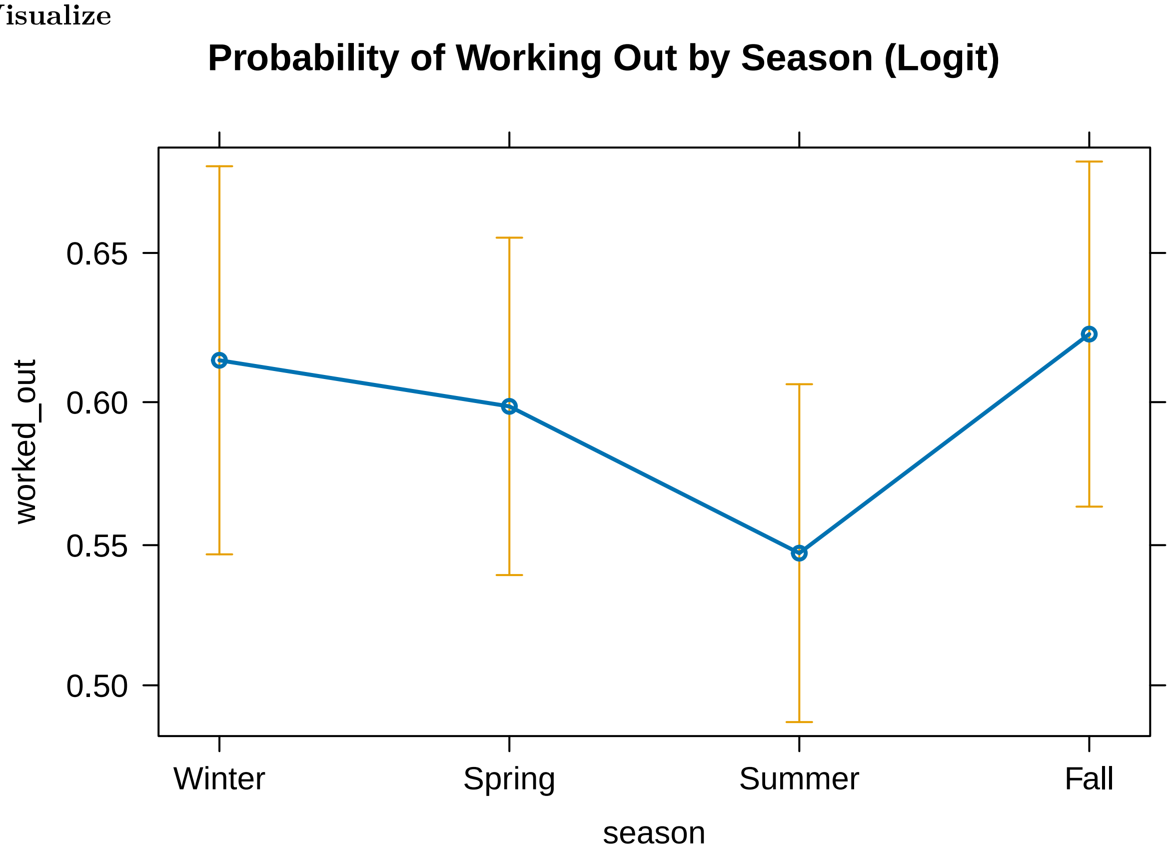
Term	Estimate	p-value	Interpretation
Intercept (Winter)	0.46536	0.001	Winter is the baseline. Converts to ~61% chance of working out (exp(0.465)/(1 + exp(0.465)) 0.614).
Spring	-0.06598	0.725	Not significant. Slightly lower odds of working out vs. winter.
Summer	-0.27612	0.142	Also not significant. Suggests decreased odds, but we can't confidently say so.
Fall	0.03751	0.843	Nearly no effect, and not statistically significant.

Chi-squared test (ANOVA): The p = 0.2938 indicate that season as a whole does not significantly improve model fit.

```
# anova(logit_season_model)
```


Estimated Workouts per Season (Poisson)
num_worked_out
20
18
16
14
12
Winter
Spring
Summer
Fall
season

Visualize


Probability of Working Out by Season (Logit)
worked_out
0.65
0.60
0.55
0.50
Winter
Spring
Summer
Fall
season

Consistency Score - Goals Meet

Trifecta Days as Count Outcome (Poisson Regression) – Monthly

Conclusion: I am slightly more likely to hit my trifecta goals in summer (with a potential ~31% increase), but the effect of season overall isn't statistically strong. Winter remains my base with ~8 trifecta days/month.

```
glm_trifecta <- glm(num_trifecta_days ~ season, data = monthly_consistency, family = poisson()
#summary(glm_trifecta)
```

Table 6: Poisson Regression: Predicting Trifecta Days by Season

Term	Estimate	p-value	Interpretation
Intercept (Winter)	2.0655	< 0.001	Baseline (Winter): On average, ~7.88 trifecta days/month in winter. –> exp(2.0655) 7.88
Spring	0.2471	0.111	Not statistically significant. Spring may increase trifecta days by ~28% (exp(0.247)), but we can't say confidently.
Summer	0.2699	0.087	Marginally significant (p ~ 0.087). Could imply ~31% more trifecta days than winter (exp(0.27) ~ 1.31).
Fall	0.2482	0.117	Similar to spring — slight positive trend (~28% increase), but not statistically strong.

Model Odds of "Trifecta Day" (logistic version) – Daily

Season has no significant influence on the likelihood of hitting a trifecta day. I am about 1 in 3 likely to hit all 3 goals on any given day in winter – and that probability stays pretty stable across seasons.

```
parsed_clean_by_day <- parsed_clean_by_day |>
  mutate(trifecta_day = activeEnergyGoalAchieved & appleExerciseTimeGoalAchieved & appleStandHo

glm_trifecta_day <- glm(trifecta_day ~ factor(season), data = parsed_clean_by_day, family = bin
```

Table 7: Logistic Regression: Predicting Trifecta Days from Season

Term	Estimate	p-value	Interpretation
Intercept (Winter)	-0.672	< 0.001	Winter is the baseline. Converts to ~33.8% chance of a trifecta day: exp(-0.672) / (1 + exp(-0.672)) 0.338.
Spring	0.134	0.487	Not significant. Small (non-reliable) increase in odds vs. winter.
Summer	0.057	0.770	No meaningful difference from winter.

Term	Estimate	p-value	Interpretation
Fall	0.007	0.973	Almost identical to winter — essentially no effect.

Ran and Season

This model estimates the likelihood of going for a run on a given day using season as the predictor. Fall is my most reliably active season for running, with a significant increase in the likelihood of going for a run compared to winter. Spring and summer show no significant change, but summer might actually suppress my running tendencies a bit.

```
ran_season <- glm(ran ~ factor(season), data = parsed_clean_by_day, family = binomial())
```

Table 8: Logistic Regression: Predicting Running Behavior by Season

Term	Estimate	p-value	Interpretation
Intercept (Winter)	-1.1371	< 0.001	Winter is the baseline. This converts to a 24.2% chance of running: exp(-1.1371)/(1+exp(-1.1371)) 0.242
Spring	0.1428	0.498	Not significant. Slight increase in odds vs. winter, but not reliable.
Summer	-0.3215	0.153	Not significant, but suggests lower odds of running in summer vs. winter.
Fall	0.4884	0.018	Statistically significant. Fall has 63% higher odds of running compared to winter. (exp(0.4884) 1.63)

Model Fit Summary

- Null deviance: 1171.0
- Residual deviance: 1153.9
- AIC: 1161.9 Season explains some variance in running behavior — particularly due to Fall's significance

Brute Force Athlete Heart Rate ROC

Exhaustive AUC Comparison of Predictor Combinations for is_athlete Classification

`is_athlete` Variable

- is_athlete = TRUE –> My lowest recorded heart rate for the day was between 40 and 60 bpm, which is a common physiological range for trained athletes.

- is_athlete = FALSE –> My lowest heart rate fell outside that range (either below 40 or above 60 bpm), so they likely don't exhibit resting HR levels consistent with trained athletes.

This script performs automated model selection by:

- Testing all combinations of predictors (1 to 6 variables at a time).
- Fitting a logistic regression model to predict whether a day belongs to an "athlete" heart rate profile.
- Evaluating each model's predictive performance using AUC (Area Under the ROC Curve).
- Returning a sorted table of models ranked by their AUC.

Why AUC?

- AUC reflects how well the model distinguishes between classes (is_athlete = TRUE/FALSE).
- AUC of 1.0 = perfect model, 0.5 = random guessing.
- Higher AUC = better classification performance.

Conclusion

The most predictive combination of whether I exhibit "athlete-like heart rate patterns" includes stand hours, walking, running, and seasonal context with it AUCs reaching 0.803, which indicates very strong predictive power for a binary classification model in health behavior.

```r
parsed_clean_by_day <- parsed_clean_by_day |>
  mutate(is_athlete = low_heart_rate >= 40 & low_heart_rate <= 60)

# define predictors
predictors <- c("factor(season)", "EnergyBurned", "ExerciseTime", "standHours", "ran", "walk")

# all combinations of predictors (excluding empty set)
predictor_combos <- unlist(lapply(1:length(predictors), function(n) {
  combn(predictors, n, simplify = FALSE)
}), recursive = FALSE)

# evaluate each model
combo_results <- map_dfr(predictor_combos, function(vars) {
  formula_str <- paste("factor(is_athlete) ~", paste(vars, collapse = " + "))
  model <- glm(as.formula(formula_str), data = parsed_clean_by_day, family = binomial())
  pred <- predict(model, type = "response")

  # Clean NAs
  valid <- complete.cases(pred, parsed_clean_by_day$is_athlete)

  if (length(unique(parsed_clean_by_day$is_athlete[valid])) < 2) {
    return(NULL) # skip if only one class is present
  }

  roc_obj <- roc(parsed_clean_by_day$is_athlete[valid], pred[valid])
  tibble(
    predictors = paste(vars, collapse = " + "),
```

```r
    auc = as.numeric(auc(roc_obj))
  )
})

# sort by best AUC
combo_results <- combo_results |>
  arrange(desc(auc))

combo_results |>
  head(5) |>
  kable()
```

predictors	auc
factor(season) + ExerciseTime + standHours + ran + walk	0.8032106
factor(season) + EnergyBurned + ExerciseTime + standHours + ran + walk	0.8030793
factor(season) + EnergyBurned + standHours + ran + walk	0.8030356
factor(season) + standHours + ran + walk	0.8013297
factor(season) + ExerciseTime + standHours + walk	0.8010454