* **کیفیت‌سنجی اولیه روی همه‌ی CSVها: حذف نمونه‌های نامعتبر با پرچم‌های اعتبار (اولویت با BPOGV، درصورت نبود FPOGV)، و حذف اطراف پلک‌زدن‌ها (± چند ثانیه). ✅**
* **کار روی نرخ نمونه‌برداری ثابت (۱۵۰Hz) و ساخت خلاصه‌ی کیفیت هر فایل/شرکت‌کننده/سؤال. ✅**
* **جداسازی منطقی دنباله به دو فاز:**
  1. **«خواندن سؤال» تا وقتی برای اولین‌بار N نمونهٔ پیاپی در ناحیه‌ی جواب‌ها دیده شود (N≈۵ یا ۱۰).**
  2. **«بررسی گزینه‌ها» از همان لحظه به بعد (ممکن است گاهی نگاه برگردد به سؤال).**
* **هم‌طول کردن دنباله‌ها برای یادگیری عمیق: برای هر فاز طول زمانی ثابت انتخاب شود (بر اساس سریع‌ترین شرکت‌کننده یا یک حد مینیممِ مشترک مثل ~۲۰s) تا ورودی‌ها هم‌اندازه شوند.**
* **ویژگی‌ها: به‌جای مختصات خام، از تفاضلِ سطرهای پیاپی استفاده کنیم:  
  ΔFPOGX, ΔFPOGY, ΔBPOGX, ΔBPOGY, ΔLPD, ΔRPD (+ درصورت نیاز خودِ مقادیر خام هم می‌توانند بیایند تا شبکه خودش انتخاب کند).  
  به‌علاوه یک پرچم blink (۰/۱) از BKID/BKDUR.**
* **تعریف AOI‌ها و کدگذاری یک‌-از-چند (one-hot) در هر سطر:  
  Question, Wrong answers, Correct answer, Right-Top (Timer)، Right-Bottom (Submit). (برای سؤال‌هایی که «جواب درست ندارد»، فقط «answers» به‌صورت «wrong» در نظر گرفته می‌شود.)**
* **سه «شرط آزمایشی» جدا تحلیل شوند:**
  1. **با جواب درست (+/− تایمر)، 2) بدون جواب درست + تایمر، 3) بدون جواب درست − تایمر.**
* **برچسب Effort/No-Effort: با آستانه‌های زمانی (یک آستانه‌ی پایین برای حذف پاسخ‌های خیلی سریع/تصادفی و یک آستانه‌ی بالا برای «تلاش بالا»). در سؤال‌های «بدون جواب درست» برچسب صرفاً بر مبنای زمان است. (فرمول نهایی را خواهند فرستاد، فعلاً پارامترپذیر می‌گذاریم.)**
* **شناسایی outlier‌ها و تست‌های نامعتبر (خیلی سریع/خیلی کند/کیفیت پایین) و حذف/گزارش آن‌ها.**

**نقشه‌ی مرحله‌به‌مرحله (وضعیت اکنون)**

**مرحله ۱ — QC & Summary (انجام شد)**

* **اسکریپت 01\_stage1\_qc\_prep.py:**
  + **فیلتر اعتبار + حذف اطراف blink (±۲s) + چک SR=150.**
  + **محاسبه‌ی kept\_ratio, duration\_valid\_s, n\_blinks, blinks\_per\_min.**
  + **خروجی: stage1/stage1\_summary.csv (مرتب‌شده بر اساس participant و question). ✅**

**مرحله ۲ — برچسب‌گذاری AOI و جداسازی دو فاز (اقدام بعدی)**

* **نیاز داریم مرزبندی AOIها به مختصات نرمال‌شده‌ی صفحه (چون FPOG\* بین ۰..۱ است):**
  1. **y\_split\_question\_answers (مثلاً ۰.۵)،**
  2. **right\_x\_min (مرز شروع ستون راست)،**
  3. **right\_y\_split برای تفکیک Timer/Submit،**
  4. **نقشه‌ی مستطیل‌های چهار گزینه (A/B/C/D) — ثابت برای همه‌ی سؤال‌ها.**
* **از answers.json می‌فهمیم کدام گزینه «درست» است تا ستون‌های one-hot «Correct/Wrong» را بسازیم.**
* **یافتن نقطه‌ی شروع فاز ۲: N نمونهٔ پیاپی در AOI جواب‌ها (N پیش‌فرض=۵؛ پارامتر).**

**مرحله ۳ — مهندسی ویژگی‌ها و هم‌طول‌سازی**

* **تولید ویژگی‌ها به‌صورت دنباله‌ی زمانی: Δها + blink + بردار one-hot AOI.**
* **تعیین طول ثابت برای هر فاز:**
  + **از بین همه‌ی شرکت‌کننده‌ها، زمانِ کمینه‌ی قابل‌قبول را برای فاز سؤال (Tq) و برای فاز جواب (Ta) می‌گیریم (بهتر است با صدک ۱۰% به‌جای دقیقاً مینیمم تا از outlierها در امان باشیم).**
  + **همه‌ی دنباله‌ها را به این طول‌ها برش/Pad می‌کنیم.**
* **خروجی: فایل‌های میانجی مثلاً stage2/features\_parquet/… به‌همراه متادیتا.**

**مرحله ۴ — برچسب Effort / No-Effort**

* **پیاده‌سازی برچسب‌گذاری پارامترپذیر بر اساس زمان پاسخ و (وقتی موجود باشد) درستی پاسخ:**
  + **اگر زمان < t\_low ⇒ رد/حذف (random/بی‌اعتبار).**
  + **اگر زمان ≥ t\_high ⇒ Effort.**
  + **بازه‌ی میانی را یا «No-Effort» بگیریم یا «نامشخص» (طبق فرمولی که استاد می‌فرستد).**
* **در حالت «بدون جواب درست» فقط زمان ملاک است.**
* **خروجی: labels.csv با ستون‌های participant, question, condition, label.**

**مرحله ۵ — پالایش نهایی و آوتلایر**

* **حذف شرکت‌کننده/سؤال‌هایی که: kept\_ratio پایین، یا زمان‌های بسیار غیرعادی دارند.**
* **گزارش تجمیعی: چند فایل حذف شد، به چه دلیل.**

**مرحله ۶ — آماده‌سازی برای مدل**

* **ساخت تنسورهای train/val/test با شکل یکنواخت:  
  [batch, time, features] برای هر فاز (می‌توان دو فاز را جدا مدل کرد یا کانکات).**
* **ذخیره در stage3/dataset/… + اسکریپت بارگذار PyTorch.**

**مرحله ۷ — اولین خط‌پایه‌ها**

* **یک مدل ساده (LogReg/MLP) + یک مدل دنباله‌ای (LSTM/GRU) فقط برای sanity-check.**
* **گزارش دقت/ROC و خطاهای رایج.**

**مرحله ۱ (QC & Prep)**

1. **پاکسازی سطرها (row-level):**
   * **فقط فریم‌هایی که یکی از پرچم‌های اعتبار نگاه ۱ است را نگه دار: FPOGV یا BPOGV یا LPOGV یا RPOGV.**
   * **فریم‌های پلک (BKID>0) و چند فریم اطرافش را حذف کن.**
   * **مختصات نگاه خارج از بازهٔ [0,1] را حذف/Null کن.**
   * **اگر LPUPILD/RPUPILD داری، مردمک را میلی‌متری بساز (×1000). وگرنه از LPD/RPD (پیکسل).**
2. **پاکسازی آزمون‌ها/سؤال‌ها (trial-level):**
   * **برای هر *سؤال×شرکت‌کننده* درصد فریم‌های معتبر را حساب کن؛ اگر مثلاً <60–70٪ بود آن تریل را کنار بگذار (مصادیق «خیلی صفر/نامعتبر» که تو جلسه گفتن).**
   * **زمان هر تریل را داری؛ اگر نه، از اندیس نمونه با 150Hz بسازیم.**
3. **آوتلایرِ زمانی و برچسب موقت «effort» (طبق همان منطقی که در جلسه توضیح دادند، تا فرمول نهایی برسد):**
   * **برای هر سؤال (جداگانه برای «با تایمر»/«بی‌تایمر») Q1/Q3 و IQR بگیر.**
   * **خیلی سریع‌ها: زمان < Q1 - 1.5\*IQR ⇒ «نامعتبر» (حذف؛ رفتار رندومی/بی‌خواندن).**
   * **خیلی کندها: زمان > Q3 + 1.5\*IQR ⇒ effort=1 (بدون توجه به درست/غلط).**
   * **بازهٔ میانی: اگر پاسخ غلط ⇒ effort=1، اگر درست ⇒ effort=0.**
   * **بلوک «بدون پاسخ درست» را بعداً جدا تحلیل می‌کنیم؛ فعلاً بعد از حذفِ خیلی سریع‌ها، همه را effort=1 درنظر بگیر (طبق صحبت جلسه دوم برای مجموعه‌ی “رندوم”ها).**
4. **فیچرهای پایه که لازم داریم (برای مراحل بعد):**
   * **اختلاف‌های پیاپی حرکت چشم: ΔFPOGX, ΔFPOGY (و در صورت نیاز برای BPOG).**
   * **مردمک (mm) یا پیکسل.**
   * **ستون «valid\_row» و در ادامه ستون «label\_effort» موقت.**

**خروجی‌های مرحله ۱ که تحویل می‌دی**

* **gaze\_clean\_stage1.csv (سطرهای معتبر + فیچرها).**
* **یک جدول خلاصهٔ تریل‌ها: تعداد حذف به‌خاطر پلک/نامعتبر، درصد اعتبار هر تریل، و برچسب موقت effort بر اساس زمان و درستی.**
* **آمار Q1/Q3/IQR برای هر سؤال (جدا برای با/بی‌تایمر).**

**وقتی فرمول رسمی اساتید برای Effort رسید، فقط بخش ۳ را با آستانه‌های جدید بازاجرا می‌کنیم؛ کل pipeline پارامتریک است.**

**نقشهٔ راه عملی (خلاصه و مرحله‌به‌مرحله بر اساس دو جلسه)**

**مرحلهٔ ۲ — QC و برچسب‌گذاری اولیه (فقط سؤالاتِ «دارای پاسخ صحیح»)**

1. **اینونتوری فایل‌ها: برای هر Participant، وجود answers.json و ۱۵ تا Q\*.csv را چک و لیست کن. (گزارش تعداد کم/زیاد یا مفقودی‌ها)**
2. **همگام‌سازی پاسخ‌ها و متادیتا: از question\_exams/Participant\_X.json تشخیص بده هر گزینه در کدام ربعِ پاسخ‌ها افتاده (اندیس ۰..۳ ⇐ بالاچپ/بالاراست/پایین‌چپ/پایین‌راست).**
3. **فیلتر اعتبار: داخل هر Qk.csv فقط ردیف‌هایی با FPOGV==1 (و در صورت نیاز L/R) نگه دار.**
4. **زمان‌ها و آوتلایر (به‌ازای هر سؤال، بین همهٔ شرکت‌کننده‌ها):**
   * **کوارتیل‌ها را از time\_spent حساب کن؛ پایین‌تر از حد پایین ⇒ «نامعتبر/حذف».**
   * **بالاتر از حد بالا ⇒ «Effort=1» خودکار.**
5. **Effort برای بقیه (زمان‌های میانی):**
   * **اگر پاسخ غلط است ⇒ Effort=1، اگر درست ⇒ Effort=0.**
6. **خروجی این مرحله: یک جدول «trial-level» مثل:  
   participant, section, qnum, qid, has\_correct, chosen\_option, is\_timeout, time\_spent, outlier\_low, outlier\_high, label\_effort.**

**تا اینجا اصلاً وارد «بدون پاسخ صحیح» نمی‌شویم؛ دقیقاً همان پیشنهادی که در جلسه بود.**

**مرحلهٔ ۳ — برچسب «Random» و ورود تدریجیِ «بدون پاسخ صحیح»**

1. **برای ۵ سؤالِ «بدون پاسخ صحیح»: بعد از حذف خیلی‌سریع‌ها، همهٔ باقی‌ها را Random=1 در نظر بگیر (طبق نظر اساتید).**
2. **برای سؤالاتِ دارای پاسخ صحیح: فعلاً Random=0 (بعداً قواعد پالایش می‌گذاریم).**
3. **خروجی: جدول قبلی + ستون label\_random.**

**مرحلهٔ ۴ — نگاشت AOI و تفکیک فازها (فاز ۱/۲)**

1. **با aoi\_schema\_base\_v1.json به هر ردیف نگاه یک area\_id بده (۰=خارج AOI).**
2. **تشخیص شروع فاز ۲: وقتی نگاه برای N=10 نمونهٔ پیاپی وارد «ناحیهٔ پاسخ‌ها» شد.**
3. **برای ردیف‌های فاز ۲، ستون کمکی is\_correct\_area را پر کن (در سؤالات دارای پاسخ صحیح).**
4. **خروجی: برای هر Qk.csv یک نسخهٔ «aoi\_mapped» (یا ستون‌های افزوده) + یک خلاصهٔ فازها.**

**مرحلهٔ ۵ — استخراج ویژگی‌ها برای مدل**

* **روی ردیف‌های معتبر: تفاضل‌های متوالی ΔFPOGX, ΔFPOGY, ΔBPOGX, ΔBPOGY, ΔLPD, ΔRPD + پرچم Blink.**
* **نمونه‌برداری/برابر‌سازی طول توالی‌ها: ابتدا ساده شروع می‌کنیم (پنجرهٔ ثابت با طول مشخص برای هر فاز).**
* **دیتاستِ مدل را جداگانه برای «بدون زمان» و «با زمان» می‌سازیم (همان‌طور که جلسه گفت).**

**مرحله 1**

**خلاصهٔ اجرایی**

* یک اسکریپت پایتون به نام 01\_stage1\_qc\_prep.py ساختیم که همهٔ CSVها رو از پوشهٔ outputs به‌صورت بازگشتی می‌خوانه.
* از روی مسیر فایل‌ها، شمارهٔ شرکت‌کننده (participant\_X) و شمارهٔ سؤال (QY) رو استخراج کردیم.
* کیفیت داده‌ها رو per file ارزیابی کردیم، آمار کلیدی حساب کردیم، و یک جدول خلاصه ساختیم.
* خروجی نهایی: فایل stage1\_summary.csv داخل پوشهٔ stage1 که **بر اساس شمارهٔ شرکت‌کننده و شمارهٔ سؤال مرتب شده**.

اجرای نمونه:

nginx

CopyEdit

python 01\_stage1\_qc\_prep.py ^

--in\_dir "...\outputs" ^

--out\_dir "...\stage1" ^

--sr 150 --blink\_pad 2 --keep\_ratio\_ok 0.6

**دقیقاً چه محاسباتی انجام شد؟**

1. **تشخیص رکوردهای معتبر (valid samples)**
   * اولویت با BPOGV==1 بود؛ اگر نبود از FPOGV==1؛ و اگر هیچ‌کدام نبود، از (LPOGV==1) OR (RPOGV==1) استفاده شد.
   * این ماسکِ اعتبار، «نمونه‌های قابل استفاده» رو مشخص کرد.
2. **حذف حاشیهٔ اطراف پلک‌زدن (blink padding)**
   * با --blink\_pad 2 به ازای هر پلک، ۲ ثانیه قبل و بعد حذف شد (بر حسب تعداد نمونه با --sr 150).
   * این کار jitter ناشی از پلک رو از دادهٔ معتبر حذف می‌کنه.
3. **شمارش پلک‌ها**
   * طبق مستندات Gazepoint، BKID در فریم‌هایی که پلک دیده می‌شه > 0 است.
   * قبلاً ممکن بود هر فریمِ داخلِ یک blink شمرده بشه؛ ما اصلاح کردیم و فقط **آغازِ هر blink** رو می‌شماریم (ترنزیشن ۰ → >۰).
4. **شاخص‌های خروجی برای هر فایل**
   * rows\_total: تعداد کل سطرها.
   * rows\_kept: تعداد سطرهای معتبر بعد از اعمال ماسک و پَدینگ.
   * kept\_ratio = rows\_kept / rows\_total.
   * duration\_total\_s = rows\_total / sr و duration\_valid\_s = rows\_kept / sr.
   * n\_blinks: تعداد blink (شروع‌ها).
   * blinks\_per\_min = n\_blinks / (duration\_total\_s/60).
   * sample\_rate\_hz: نرخ نمونه‌برداری (پیش‌فرض ۱۵۰).
   * valid\_by\_tracker: پرچم پذیرش کیفیت بر اساس آستانهٔ --keep\_ratio\_ok (پیش‌فرض 0.6).
5. **مرتب‌سازی و ذخیره**
   * سطرهای خلاصه بر اساس شمارهٔ واقعی شرکت‌کننده و سؤال **صعودی** مرتب شدند (participant\_1 → Q1..Q15، بعد participant\_2 → …).
   * همه در stage1\_summary.csv ذخیره شد (UTF-8-SIG برای نمایش درست فارسی در Excel).

**چه چیزهایی قابل تنظیم است؟**

* --sr: اگر دستگاه/تنظیمات دیگری داشته باشی.
* --blink\_pad: شدت حذف اطراف blink (به ثانیه).
* --keep\_ratio\_ok: آستانه‌ی قبولی کیفیت (برای ستون valid\_by\_tracker).

**خروجی مرحله ۱ به چه درد مرحله ۲ می‌خورد؟**

* الان دقیقاً می‌دونی کدوم فایل‌ها کیفیت کافی دارند (valid\_by\_tracker = TRUE) و کدوم‌ها نه.
* در مرحله ۲ می‌تونیم:
  + فقط فایل‌های «قبول‌شده» رو وارد تحلیل کنیم،
  + به answers.json هر شرکت‌کننده join بزنیم،
  + پنجره‌های زمانی/متریک‌های نگاه (مثلاً میانگین/واریانس POG، dwell، تعداد fixationها…) را بسازیم،
  + و وقتی فرمول effort/not-effort از استاد رسید، بر همان اساس برچسب‌گذاری کنیم.

مرحله 2:

**Stage 2 دقیقاً چه می‌کند؟**

۱) ورودی‌ها را می‌گیرد (مسیرها و پارامترها) →  
۲) هندسه‌ی رابط را مستقیماً از **config.ini** به مختصات نرمال 0..1 تبدیل می‌کند →  
۳) برای هر شرکت-کننده و هر سؤال، نگاه را روی **AOI**‌های دقیق مپ می‌کند →  
۴) فازها را تفکیک می‌کند (**Q** تا قبل از ورود پایدار به پاسخ‌ها، بعدش **A**) →  
۵) صحیح/غلط بودن نمونه‌های نگاه به گزینه‌ها را محاسبه می‌کند →  
۶) خروجی‌های per-question (اختیاری) و خلاصه‌ی نهایی **stage2\_summary.csv** را می‌سازد.

**ورودی‌ها و اجرا**

پارامترهای کلیدی:

* --in\_dir پوشه‌ی **outputs** (داخلش participant\_\* و Q#.csv)
* --out\_dir پوشه‌ی خروجی stage2
* --config\_ini همان config.ini رابط
* --questions\_json بانک سؤالات (id گزینه‌ی صحیح با -C ختم می‌شود)
* --exams\_dir پوشه‌ی question\_exams (ترتیب شخصی هر شرکت‌کننده)
* --sr نرخ نمونه‌برداری (Hz، پیش‌فرض 150)
* --consec\_answer\_samples آستانه‌ی شروع فاز Answer (مثلاً 5 نمونه‌ی پیاپی)
* --timer\_split مرز عمودی AOI تایمر در پنل راست (پیش‌فرض 0.5 = نیمه‌ی بالا)
* --save\_per\_question اگر بدهید، CSV تمیز هر سؤال هم ذخیره می‌شود.

**استخراج هندسه از config.ini**

از بخش DIMENSIONS مستقیماً نسبت‌ها خوانده می‌شود:

* پهنای پنل‌ها:  
  left\_panel\_ratio = lp ، right\_panel\_ratio = 1 - lp
* تقسیمات عمودی پنل چپ:  
  بالایی «سؤال» (question\_section\_ratio = qsr) و پایینی «گزینه‌ها» (options\_section\_ratio = osr)
* پدینگ‌ها و چیدمان گزینه‌ها:  
  options\_frame\_padx\_ratio (ofpx) پدینگ افقی،  
  options\_frame\_pady\_ratio (ofpy) پدینگ عمودی،  
  option\_button\_columns (cols) تعداد ستون‌ها (معمولاً ۲)،  
  option\_button\_spacing\_ratio (spx) فاصله افقی بین ستون‌ها،  
  option\_button\_margin\_ratio (spy) فاصله عمودی بین ردیف‌ها.
* دکمه Submit دقیقِ سمت راست (چون با place گذاشته‌اید):  
  submit\_padx\_ratio (relx) و submit\_pady\_ratio (rely) مرکز دکمه داخل پنل راست،  
  submit\_width\_ratio نسبت به **عرض پنل راست**،  
  submit\_height\_ratio نسبت به **ارتفاع کل صفحه**.

همه‌ی این‌ها به مستطیل‌های نرمال‌شده 0..1 تبدیل می‌شوند، بنابراین اگر رزولوشن نمایش عوض شود، محاسبات دقیق می‌ماند.

**تعریف AOI‌ها**

برای هر نمونه‌ی نگاه (x,y در بازه‌ی 0..1):

* **question**: پنل چپ، نیمه‌ی بالایی (y < qsr)
* **answers (outer rect)**: پنل چپ، نیمه‌ی پایینی + پدینگ‌های ofpx/ofpy
* **answers/گزینه‌ها (مستطیل‌های داخلی)**:  
  داخل outer rect یک شبکه‌ی cols ستونه و rows = ceil(n\_options/cols) ساخته می‌شود.  
  فاصله‌های بین ستون‌ها spx \* lp و بین ردیف‌ها spy \* osr است.  
  ترتیب پر شدن مستطیل‌ها: ردیف به ردیف، از چپ به راست (کاملاً مطابق چیدمان UI).
* **right\_timer**: پنل راست و y < timer\_split (تقریب پایدار برای تایمر)
* **right\_submit**: مستطیل دقیق بر اساس place (مرکز و عرض/ارتفاع از config)
* **right\_bg**: بقیه‌ی پنل راست که نه تایمر است و نه Submit
* **background**: هر جا که جزو موارد بالا نباشد

اولویت برچسب AOI در هر نمونه:  
right\_submit > right\_timer > answers > question > right\_bg > background

ستون خروجی area\_id دقیقاً یکی از این‌هاست.

**نگاشت نمونه‌ها به گزینه‌ی خاص**

برای هر سؤال، از فایل Participant\_X.json لیست **id**‌های گزینه‌ها (به همان ترتیب نمایش) می‌آید.  
برای هر نمونه‌ی داخل answers:

* اگر در محدوده‌ی مستطیل گزینه‌ی i باشد، مقدار ستون opt\_id\_at\_sample می‌شود **id** همان گزینه (مثلاً 4-1-C).
* اگر نمونه روی مرز دو مستطیل بیفتد، اولین مستطیلی که در پیمایش (چپ→راست، بالا→پایین) بخورد انتخاب می‌شود.
* اگر خارج از answers باشد، مقدار خالی می‌ماند.

**محاسبه‌ی صحیح/غلط در سطح نمونه**

* از **questions.json** درست/غلط بودن را می‌گیریم: هر گزینه‌ای که id‌اش با -C تمام شود، **گزینه‌ی صحیح** است.
* اگر در Participant\_\*.json فیلد has\_correct\_answer = false بود (سؤال بدون گزینه‌ی صحیح)، هر برخورد با گزینه‌ها به عنوان «غلط» شمرده می‌شود.
* ستون‌ها:
  + aoi\_correct = 1 اگر opt\_id\_at\_sample == correct\_option\_id
  + aoi\_wrong = 1 اگر داخل گزینه‌ای باشد ولی ≠ گزینه‌ی صحیح (یا سؤال بدون پاسخ صحیح)

**تفکیک فازها (Q/A)**

هدف: زمان «خواندن سؤال» تا زمان «ورود پایدار به پاسخ‌ها».

* از اولین **شروع پنجره‌ای** که consec\_answer\_samples نمونه‌ی **پیاپی** در AOI=answers (و **valid**) دیده شود، فاز **A** آغاز می‌شود.
* قبلش **Q** است.
* اگر هرگز پایدار وارد answers نشود، کل سؤال **Q** می‌ماند.

خروجی‌ها:

* phase (هر نمونه: Q یا A)
* phase\_onset\_s (زمان آغاز A بر حسب ثانیه؛ اگر نیفتد خالی)

**اعتبار نمونه و زمان‌بندی**

* مختصات نگاه از ستون‌های دستگاه با کشف خودکار نام ستون‌ها می‌آید (FPOGX/FPOGY و چند نام جایگزین).
* sample\_valid زمانی True است که **x و y هر دو بین 0 و 1** باشند.
* t\_s = index / sr زمان نسبی هر نمونه است.
* زمان‌ها در خلاصه:
  + duration\_total\_s = تعداد نمونه‌های **valid** / sr
  + duration\_Q\_s = مجموع زمان نمونه‌های Q و valid
  + duration\_A\_s = total - Q

**خروجی‌های per-question (اختیاری)**

اگر --save\_per\_question بدهید، برای هر Q# یک CSV تمیز داخل پوشه‌ی participant ساخته می‌شود. ستون‌های مهم:

* t\_s, x, y, sample\_valid
* in\_left, in\_right, in\_question, in\_answers\_rect, in\_right\_timer, in\_right\_bg, in\_submit
* area\_id (برچسب AOI نهایی)
* opt\_id\_at\_sample (id گزینه‌ی زیر نگاه، اگر داخل answers باشد)
* aoi\_correct, aoi\_wrong (۰/۱ در سطح نمونه)
* phase (Q/A)
* مشخصات: participant, section, question\_id, question\_number, has\_correct\_answer

**خلاصه‌ی نهایی: stage2\_summary.csv**

برای هر شرکت‌کننده و هر سؤال یک ردیف:

* شناسنامه: participant, section, question\_number, question\_id
* کیفیت داده: rows\_total, rows\_valid
* زمان‌ها: duration\_total\_s, duration\_Q\_s, duration\_A\_s, phase\_onset\_s
* شمارش AOI‌ها: hits\_timer, hits\_submit, hits\_answers, hits\_question
* صحت نمونه‌ها: correct\_samples, wrong\_samples
* کلید پاسخ: has\_correct\_answer, correct\_option\_id

فایل نهایی قبل از ذخیره **بر اساس (participant, question\_number)** مرتب می‌شود تا ترتیب Q1..Qn برای هر نفر حفظ شود.

**نکات تنظیم و تفسیر**

* اگر حس کردید شروع فاز A دیر/زود تشخیص داده می‌شود، --consec\_answer\_samples را تغییر دهید (مثلاً 3 یا 7).
* اگر AOI تایمر زیادی/کم می‌افتد، --timer\_split را کم و زیاد کنید (مثلاً 0.45 یا 0.55).
* چیدمان گزینه‌ها به‌صورت کامل از config می‌آید؛ هر تغییری در فاصله‌ها/پدینگ‌ها/ابعاد دکمه Submit را در **config.ini** بزنید تا محاسبات AOI دقیق بماند.
* اگر فایلی باز باشد (Excel)، ویندوز اجازه‌ی overwrite نمی‌دهد؛ ببندید یا --out\_dir دیگری بدهید.

مرحله 3

**مرحله ۳: تجمیع نتایج و محاسبه‌ی شاخص‌ها**

**1) ورودی‌ها**

* **stage2\_summary.csv** (خروجی مرحله ۲): برای هر سؤالِ هر شرکت‌کننده، آمار AOI و فازها (Q/A) را دارد؛ از جمله:
  + participant, section, question\_number, question\_id
  + rows\_total, rows\_valid, duration\_total\_s, duration\_Q\_s, duration\_A\_s, phase\_onset\_s
  + شمارش AOI: hits\_timer, hits\_submit, hits\_answers, hits\_question
  + کیفیت نمونه‌ها: correct\_samples, wrong\_samples
  + کلیدهای پاسخ: has\_correct\_answer, correct\_option\_id
* **outputs/participant\_\*/answers.json** (در صورت وجود): انتخاب شرکت‌کننده و زمان صرف‌شده روی هر سؤال:
  + section, question\_number, question\_id, chosen\_option, time\_spent
* **questions.json** (صرفاً به‌عنوان پشتیبان برای درستی کلید درست؛ هرچند stage2\_summary.csv خودش correct\_option\_id را می‌آورد).
* آرگومان‌ها: مسیرها، و مسیر خروجی stage3.

**2) آماده‌سازی داده‌ها**

* خواندن stage2\_summary.csv به DataFrame اصلی m.
* استخراج یک ستون کمکی برای مرتب‌سازی عددی:
  + participant\_num = عدد انتهای participant (مثلاً از participant\_23 → 23).
* اگر فایل‌های answers.json وجود داشته باشند، برای همه‌ی participant\_\*ها خوانده می‌شوند و به یک جدول «پاسخ‌ها» تبدیل می‌شوند:
  + ستون‌ها: participant, section, question\_number, question\_id, chosen\_option, time\_spent.
* **ادغام پاسخ‌ها با stage2**:
  + کلید امن ادغام: participant + question\_number (چون مرحله ۲ دقیقاً بر اساس ترتیب Q1..Qn ساخته شده)؛ در صورت نیاز question\_id هم بررسی/همسان‌سازی می‌شود.
  + نتیجه: به m ستون‌های chosen\_option و time\_spent اضافه می‌شود.
* محاسبه‌ی **درستی پاسخ** در هر ردیف:
  + اگر has\_correct\_answer == True و chosen\_option با correct\_option\_id برابر باشد → is\_correct = 1؛
  + اگر has\_correct\_answer == True و پاسخ نادرست/خالی باشد → is\_correct = 0؛
  + اگر has\_correct\_answer == False → مقدار درستی به‌صورت تهی یا محاسبه نمی‌شود.

نکته: اگر answers.json در پوشه‌ی یک شرکت‌کننده نباشد یا کلیدها جور نشوند، ستون chosen\_option خالی می‌ماند و طبیعتاً is\_correct برای آن ردیف‌ها صفر می‌شود. (این حالت را از روی CSV خروجی به‌راحتی می‌توان دید.)

**3) خروجی ۱: stage3\_per\_question.csv (سطح «هر سؤال برای هر شرکت‌کننده»)**

این فایل در واقع همان stage2\_summary است به‌علاوه‌ی ستون‌های پاسخ:

* ستون‌های کلیدی:  
  participant, section, question\_number, question\_id, chosen\_option, is\_correct, time\_spent, …
* آمار AOI و زمان‌ها همان است که مرحله ۲ تولید کرده:  
  duration\_Q\_s, duration\_A\_s, phase\_onset\_s, hits\_\*
* **مرتب‌سازی** پیشنهادی برای خوانایی پایدار:
  + ابتدا بر اساس participant\_num، سپس section، سپس question\_number.  
    (با این کار در Excel هم ترتیب همیشه درست دیده می‌شود.)

**4) خروجی ۲: stage3\_by\_participant.csv (تجمیع «برای هر شرکت‌کننده»)**

برای هر شرکت‌کننده، از روی ردیف‌هایش در per\_question شاخص‌های زیر محاسبه می‌شود:

* n\_questions = تعداد سؤالاتی که واقعاً برای او در stage2 خلاصه شده‌اند.
* میانگین‌ها (بدون احتساب NaN):
  + mean\_phase\_onset\_s
  + mean\_duration\_Q\_s
  + mean\_duration\_A\_s
* جمع AOIها:
  + sum\_hits\_timer, sum\_hits\_submit, sum\_hits\_answers, sum\_hits\_question
* دقت (Accuracy):
  + accuracy\_overall = میانگین is\_correct روی ردیف‌هایی که has\_correct\_answer == True
  + accuracy\_part1 = میانگین is\_correct فقط در section == Part1
  + accuracy\_part2 = میانگین is\_correct فقط در section == Part2
* ستون کمکی: participant\_num (برای سورت عددی)
* **مرتب‌سازی خروجی**: بر اساس participant\_num.

اگر answers.json نداشته باشید یا chosen\_option خالی باشد، این دقت‌ها صفر می‌شوند (همان چیزی که در خروجی شما دیده می‌شد). با افزودن/درست‌کردن پاسخ‌ها، همین کدها دقت را درست محاسبه می‌کنند.

**5) خروجی ۳: stage3\_by\_question.csv (تجمیع «برای هر شماره سؤال»)**

برای هر question\_id، بین همه‌ی شرکت‌کنندگان محاسبه می‌شود:

* n\_participants یا تعداد ردیف‌هایی که آن سؤال را دارند (باید برابر تعداد شرکت‌کنندگان باشد).
* میانگین‌های زمانی مشابه: mean\_phase\_onset\_s, mean\_duration\_Q\_s, mean\_duration\_A\_s
* جمع AOIها: sum\_hits\_timer, sum\_hits\_submit, sum\_hits\_answers, sum\_hits\_question
* accuracy\_overall (میانگین is\_correct برای آن سؤال بین شرکت‌کنندگان، روی ردیف‌های دارای جواب درست)
* **مرتب‌سازی خروجی**: بر اساس question\_id.

**6) کنترل کیفیت و لاگ‌های اجرای اسکریپت**

اسکریپت در پایان چند پیام چک چاپ می‌کند تا سریعاً بفهمید داده‌ها معقول‌اند:

* [per\_question] rows=..., participants=..., questions(min..max)=(1, 15)  
  (باید 15 سؤال و 30 شرکت‌کننده را تأیید کند؛ تعداد ردیف‌ها معمولاً 30×15=450+1 هدر)
* [by\_participant] rows=30 (unique participants)
* [by\_question] rows=15 (unique questions)

اگر هرکدام از این اعداد کمتر بود، یعنی داده‌های ورودی ناقص است (فایل/پوشه‌یی جا مانده، یا mismatch در join).

**7) نکات اجرایی/نمایشی**

* **مرتب‌سازی پایدار**: با داشتن participant\_num و سورت قبل از ذخیره، ترتیب در Excel هم همیشه درست است. اگر ترتیب را دستی در Excel عوض کردید، کافی است روی همین ستون‌ها مرتب کنید.
* **هشدارهای pandas**: برای groupby.apply، در کد از group\_keys=False استفاده شده تا رفتارهای آینده‌ی pandas با خروجی‌ها سازگار باشد و اخطارها به حداقل برسد.
* **کارایی**: این مرحله صرفاً جمع و جور کردن CSV است (فایل‌های خام را نمی‌خواند)، بنابراین سریع و سبک اجرا می‌شود.