

Supporting Information: Open science with style: A reproducible reproducible research report

Thomas P. Urbach
Kutas Lab
Cognitive Science Department
University of California, San Diego

Contents

Summary	1
System setup	2
Installing conda environments	2
Installing L ^A T _E X	2
Linux Installation via network	2
OSX Installation	3
Windows	3
Source: author_analysis.ipynb	4
Source: research_report.tex	29
Source: author_si.tex	44
Source: fig1.tex	50
Source: fig3.tex	51
Source: Makefile	55
Source: research_report.bib	57

Summary

Additional information can go here and be formatted to APA 6th guidelines or something else. The supporting information and main document can use the same research_report.bib file so the references match. With `\usepackage[xr,user,titleref]{zref}`, you can cross-reference back and forth between documents ...the main report and this SI. For instance here is a reference to Figure 2 in the main document. The reference is via the label, i.e., `\zlabel{ms:multipanel}` so if the figure is moved to a different page or its number changes

because of additions or deletions, this reference by number will update automatically. The following sections show the source files that generated the plots, figures, manuscript and supporting information pdfs.

APA 7th docs <http://ctan.math.washington.edu/tex-archive/macros/latex/contrib/apa7/apa7.pdf> APA 6th docs <http://ctan.math.utah.edu/ctan/tex-archive/macros/latex/contrib/biblatex-contrib/biblatex-apa6/biblatex-apa6.pdf>

System setup

Installing conda environments

If you already use conda environments in a recent linux operating system, you can install a minimal conda environment to run the notebooks like so and follow the prompt (or omit `-y` to the end of the command to install the packages without prompting).

```
conda create -n apa67_report pandas pyarrow matplotlib jupyter firefox -y
$ activate apa67_report
$ jupyter notebooks
```

If you are not yet set up to use conda environments, you can follow the instructions to download and install a minimal conda installer, miniconda3 ([link](#)). This provides just enough infrastructure to create a conda environment and install packages as shown in the example above. If you want to create conda environments and install packages faster, then install the 'mamba' conda package ([link](#)).

If you are not yet set up to use conda environments and don't want to be then you are on your own. You can run `pipeline_1.ipynb` if you have numpy, pandas, matplotlib and jupyter. You need the `spudtr` package to run `pipeline_2.ipynb`. Older versions are available via pip install, but there is no assurance it is compatible with the versions of packages you already have installed.

Installing L^AT_EX

Linux Installation via network. You do not need to be root or admin to install TeX Live over the networks and best practices are to install your copy in your directory. That way you control the version and packages you use. First read through the quick installation instructions [here](https://www.tug.org/texlive/acquire-netinstall.html). Then, (summarizing from <https://www.tug.org/texlive/acquire-netinstall.html>):

1. Download [install-tl-unx.tar.gz](https://www.tug.org/texlive/acquire-netinstall.html) to some scratch/working directory, unpack the archive, change to the new directory it made, i.e., `install-tl-YEARMONTHDAY` for whatever version, and run the installer.

```
$ tar -xf install-tl-unx.tar.gz
$ cd install-tl-20200814
$ perl install-tl
```

Follow the prompts, make sure you are happy with and have write permissions in the default installation directory, and press "i" to install.

2. Update your `/.bashrc` file with the path to the new TeX Live installation.

```
PATH=/home/turbach/texlive/2020/bin/x86_64-linux:$PATH
INFOPATH=/home/turbach/2020/texmf-dist/doc/info:$INFOPATH
MANPATH=/home/turbach/2020/texmf-dist/doc/man:$MANPATH
```

That's it, you have a complete functioning installation of L^AT_EX with the latest packages, TeX Live 2020 as of this writing.

The installation probably has everything you need including the `apa6` and `apa7` styles used for this report.

If there is a new package or update you want and you want to manage the TeX packages with the TeX Live GUI you also need to install `perl/tk`. There is a conda package for this, you can install into any compatible conda env.

```
$ conda activate some_general_purpose_env
$ conda install perl-tk -c BioBuilds -y
```

OSX Installation. See instructions for MacTeX here: <https://www.tug.org/mactex/>

Windows. See Quick Install instructions [here](#) and Windows installer instructions [here](#).

Source: `author_analysis.ipynb`

The pdf of the notebook is generated by `jupyter convert ... -to pdf`. The LaTeX package `pdfpages` is used to slurp it into the SI pdf.

apa_analysis

September 5, 2021

1 Reproducible results for L^AT_EX manuscripts

- arbitrary narrative text and results
- pandas L^AT_EX table generation
- custom APA-style table generation
- APA-style graphics styled with matplotlib style sheets

WARNING: Running this code the first time downloads an 87MB EEG data file to your disk from Zenodo.

The package dependencies are python, numpy, pandas, pyarrow, matplotlib, jupyter

2 The reproducible data analysis

Set up Python packages for data analysis and visualization

Guard the conda environment and EEG file MD5 checksum

```
[1]: import os
import re
import copy
import hashlib
import warnings
from pathlib import Path
import pprint as pp
import platform
import numpy as np
import pandas as pd

# matplotlib and packages for plot tuning
import matplotlib as mpl
from matplotlib import pyplot as plt
from matplotlib import cycler
from matplotlib import cm

# guard conda environment
conda_env = os.environ["CONDA_DEFAULT_ENV"] if "CONDA_DEFAULT_ENV" in os.
↳environ.keys() else None
```

```

if conda_env and not conda_env == "apa67_report_090421":
    msg = (
        f"unknown conda env {conda_env}, to reproduce the report on linux_
        ↪create run these:\n\n"
        f"    conda create -n apa67_report_090421 --files environment.txt\n"
        f"    conda activate \n\n"
    )
    warnings.warn(msg)

# fetch the EEG recording from Zenodo if it isn't found locally
ARCHIVE = r"https://zenodo.org/record/4099632/files/"

DATA_F = "sub000p3.ms1500.epochs.feather"
if not Path(DATA_F).exists():
    print(f"downloading {DATA_F} from Zenodo ... please wait")
    pd.read_feather(ARCHIVE + DATA_F).to_feather(DATA_F)
    print("ok")

# guard the data file MD5 ... note the pd.read_feather file md5 is NOT == to_
    ↪zenodo md5.
with open(DATA_F, 'rb') as _f:
    checksum = hashlib.md5(_f.read()).hexdigest()
    if not checksum == "faedff42de40ff1972baecf61f804aea":
        raise ValueError(f"bad md5 checksum {DATA_F}")

print(f"{DATA_F} ok")

for pkg in [np, pd, mpl]:
    print(pkg.__name__, pkg.__version__)

```

```

sub000p3.ms1500.epochs.feather ok
numpy 1.21.2
pandas 1.3.2
matplotlib 3.4.3

```

3 Experiment parameters

3.1 Electrode and fiducial landmark locations

```

[2]: # -----
# scalp electrodes, EOG, mastoids, ground
import io
sph26_txt = io.StringIO("""
channel  phi   theta  ch_type
MiPf    90.0   90.0    eeg

```

```

LLPf  90.0  126.0  eeg
LLFr  90.0  162.0  eeg
LLTe  90.0  198.0  eeg
LLOc  90.0  234.0  eeg
MiOc  90.0  270.0  eeg
RLOc  90.0  306.0  eeg
RLTe  90.0  342.0  eeg
RLFr  90.0   18.0  eeg
RLPf  90.0   54.0  eeg
LMPf  59.0  108.0  eeg
LDFr  59.0  144.0  eeg
LDCe  59.0  180.0  eeg
LDPa  59.0  216.0  eeg
LMOc  59.0  252.0  eeg
RMOc  59.0  288.0  eeg
RDPa  59.0  324.0  eeg
RDCe  59.0    0.0  eeg
RDFr  59.0   36.0  eeg
RMPf  59.0   72.0  eeg
LMFr  26.0  126.0  eeg
LMCe  26.0  198.0  eeg
MiPa  26.0  270.0  eeg
RMCe  26.0  342.0  eeg
RMFr  26.0   54.0  eeg
MiCe   0.0    0.0  eeg
A1    130.0  205.0  ref
A2    130.0  335.0  ref
lle   140.0  120.0  eog
rle   140.0   60.0  eog
lhz   108.0  130.0  eog
rhz   108.0   50.0  eog
nasion 108.0   90.0  fid
lpa    108.0  180.0  fid
rpa    108.0    0.0  fid
gnd     72.0   90.0  gnd
""")

# parse lcoations into a data frame
SPH_LOCS = pd.read_csv(sph26_txt, sep="\s+")
SPH_LOCS.insert(3, "r", np.sin(SPH_LOCS["phi"]))
SPH_LOCS

def sph2cart(row):
    """convert spherical coordinates to 2-D cartesian"""
    row = row.copy()
    label, phi, theta, r, ch_type = [*row]

```

```

deg2rad = 2.0 * np.pi / 360.0
phi *= deg2rad
theta *= deg2rad

x = np.cos(theta) * np.sin(phi)
y = np.sin(theta) * np.sin(phi)
z = np.cos(phi)

# lambert projection
lambert_x = x * np.sqrt(1 / (1 + z))
lambert_y = y * np.sqrt(1 / (1 + z))

row['x'], row['y'], row['z'] = x, y, z
row['x_lambert'], row['y_lambert'] = lambert_x, lambert_y

return row

SPH_CART_LOCS = SPH_LOCS.apply(lambda row: sph2cart(row), axis=1)

```

3.2 Data columns and indexes

```

[3]: INDEXES = ["epoch_id", "time_ms"]
EEG_MIDLINE = ["MiPf", "MiCe", "MiPa", "MiOc"]
EXPT_VARS = ["bin", "tone", "stimulus", "accuracy"]

EEG_COLUMNS = SPH_LOCS.query("ch_type == 'eeg'")["channel"].tolist()
COI = INDEXES + EXPT_VARS + EEG_COLUMNS # EEG_MIDLINE

```

3.3 Groom the recordings for analysis

```

[4]: data = pd.read_feather("sub000p3.ms1500.epochs.feather")
data.rename(columns={"match_time": "time_ms"}, inplace=True)
data["epoch_id"] = data["epoch_id"].astype(int)
data.rename(columns={"stim": "stimulus"}, inplace=True)

# data QC screening
display(len(data.epoch_id.unique()))
good_epoch_ids = data.query("time_ms==0 and log_flags==0").epoch_id
data = data.query("epoch_id in @good_epoch_ids")
print(data.columns)

good_epochs = []
absmax = 125
for epoch_id, epoch in data.groupby("epoch_id"):
    vals = epoch[EEG_COLUMNS].to_numpy().flatten()
    if vals.max() - vals.min() <= absmax:

```



```

# center EEG on mean amplitude 200 - 0 ms prestimulus
epoch[EEG_COLUMNS] = (
    epoch[EEG_COLUMNS]
    - epoch.query("time_ms >= -200 and time_ms < 0")[EEG_COLUMNS].mean()
)
good_epochs.append(epoch)

p3_eeg = pd.concat(good_epochs, axis=0)

# save
p3_eeg[COI].reset_index(drop=True).to_feather("p3_eeg.fthr")

```

600

```

Index(['epoch_id', 'data_group', 'dblock_path', 'dblock_tick_idx',
      'dblock_ticks', 'crw_ticks', 'raw_evcodes', 'log_evcodes', 'log_ccodes',
      'log_flags', 'epoch_match_tick_delta', 'epoch_ticks', 'dblock_srate',
      'match_group', 'idx', 'dlim', 'anchor_str', 'match_str', 'anchor_code',
      'match_code', 'anchor_tick', 'match_tick', 'anchor_tick_delta',
      'is_anchor', 'regex', 'ccode', 'instrument', 'bin', 'tone', 'stimulus',
      'accuracy', 'acc_type', 'time_ms', 'anchor_time', 'anchor_time_delta',
      'diti_t_0', 'diti_hop', 'diti_len', 'pygarv', 'lle', 'lhz', 'MiPf',
      'LLPf', 'RLPf', 'LMPf', 'RMPf', 'LDf', 'RDf', 'LLf', 'RLf', 'LMf',
      'RMf', 'LMCe', 'RMf', 'MiCe', 'MiPa', 'LDCe', 'RDCe', 'LDPa', 'RDPa',
      'LMOc', 'RMOc', 'LLTe', 'RLTe', 'LLOc', 'RLOc', 'MiOc', 'A2', 'HEOG',
      'rle', 'rhz'],
      dtype='object')

```

3.4 Load the groomed EEG data

```

[5]: p3_df = pd.read_feather("p3_eeg.fthr")
p3_events = p3_df.query("time_ms == 0 and stimulus != 'cal'")[INDEXES +
    ↳ EXPT_VARS]

display(len(p3_df.epoch_id.unique()))
display(p3_events.shape)

```

447

(239, 6)

3.5 Tabulate stimulus event counts by experimental condition

```
[6]: event_table = pd.crosstab(p3_events.stimulus, p3_events.tone, margins=True)

# event_table.columns = [col for col in event_table.columns]
event_table.reset_index(inplace=True)

# event_table["stimulus"] = event_table["stimulus"].str.capitalize()
# event_table.columns = event_table.columns.str.capitalize()

event_table.set_index("stimulus", inplace=True)
display(event_table)
```

tone	hi	lo	All
stimulus			
standard	107	94	201
target	14	24	38
All	121	118	239

4 Example: Linking data and arbitrary text

```
[7]: # data variables from the table for clarity
n_trials = event_table["All"]["All"]
n_standards = event_table.loc["standard"]["All"]
n_targets = event_table.loc["target"]["All"]

# a bit of data validation
assert n_standards + n_targets == event_table["All"]["All"]

# compute the proportion ... a derived value
p_targets = n_targets / (n_standards + n_targets)
n_trials, n_standards, n_targets, p_targets
```

```
[7]: (239, 201, 38, 0.1589958158995816)
```

```
[8]: # embed data into formatted LaTeX via the variables

arbitrary_text = f"""
% These two paragraphs are generated when the analysis is run

The essential feature of reproducible report generation is linking
data from the analysis with the text of the report. Style conventions
like APA 6th, 7th and others are
strict and varied which means the only general solution is a mechanism
for linking the analysis data and results to arbitrary text formatted
arbitrarily. This is an old problem, solved long ago by string formatting
functions, e.g., \mintinline{c}{sprintf()} in C, which reappears in
```

various forms in scripting languages like R, MATLAB, and Python where the f-string function (Python 3.6+) streamlines mixing text and variables.

To illustrate, the same Jupyter notebook that runs the analysis also generates a text file containing the entire contents of the preceding paragraph and this one, including the following sentence that describes the number of trials in each experimental condition.

```
%%
```

```
%% In the next sentence, the Python f-string formatter embeds variables  
%% computed during the analysis directly into the generated text which  
%% typeset to APA 6th style specifications.
```

```
%%
```

```
After screening artifacts, the proportion of target trials in the data  
analyzed was {p_targets:0.3f} ({{\it N}} = {n_trials} trials, {n_standards}  
standards, {n_targets} targets).
```

```
%%
```

```
This narrative description formats the quantitative results in APA 6th style  
while the values are filled in by the same variables used to compute them. This  
technique can be used to generate reproducible descriptions of an  
entire results sections or portions thereof.
```

```
"""
```

```
# show (optional)
```

```
print(arbitrary_text)
```

```
# write the text to a file for import into the manuscript
```

```
with open("generated/arbitrary_text.tex", "w") as fh:
```

```
    fh.write(arbitrary_text)
```

```
% These two paragraphs are generated when the analysis is run
```

The essential feature of reproducible report generation is linking data from the analysis with the text of the report. Style conventions like APA 6th, 7th and others are strict and varied which means the only general solution is a mechanism for linking the analysis data and results to arbitrary text formatted arbitrarily. This is an old problem, solved long ago by string formatting functions, e.g., `\mintinline{c}{sprintf()}` in C, which reappears in various forms in scripting languages like R, MATLAB, and Python where the f-string function (Python 3.6+) streamlines mixing text and variables.

To illustrate, the same Jupyter notebook that runs the analysis also generates a text file containing the entire contents of the preceding paragraph and this one, including the following sentence that describes the number of trials in each experimental condition.

```
%%
```

```

%% In the next sentence, the Python f-string formatter embeds variables
%% computed during the analysis directly into the generated text which
%% typeset to APA 6th style specifications.
%%
After screening artifacts, the proportion of target trials in the data
analyzed was 0.159 ({\it N} = 239 trials, 201
standards, 38 targets).
%%
This narrative description formats the quantitative results in APA 6th style
while the values are filled in by the same variables used to compute them. This
technique can be used to generate reproducible descriptions of an
entire results sections or portions thereof.

```

5 Example: Table 1

An easy LaTeX table with `pandas.DataFrame.to_latex()`

The output is not quite APA 6th style.

```

[9]: # show
      print(event_table.to_latex())

      # save
      event_table.to_latex('generated/p3_table1.tex')

```

```

\begin{tabular}{lrrrr}
\toprule
tone & hi & lo & All & \\
stimulus & & & & \\
\midrule
standard & 107 & 94 & 201 & \\
target & 14 & 24 & 38 & \\
All & 121 & 118 & 239 & \\
\bottomrule
\end{tabular}

```

6 Example: Table 2

An APA 6th style LaTeX table built with Python

Build the header, data rows and columns, footer strings, then write the LaTeX file.

```

[10]: def df_to_tex(df):
        """format df values as a LaTeX string of rows x columns table data"""

```

```

    df_str = df.applymap(lambda x: f"{x}".capitalize()) # convert the data to
    ↪APA style text
    tex_cols = df_str.apply(lambda row: " & ".join(row), axis=1) # join the
    ↪columns with &
    tex_rows_cols = (r" \\ " + "\n").join(tex_cols) # join the rows with \\
    return tex_rows_cols

# 1. build the table header by hand thanks to APA style
table1_header = f"""
\\begin{{tabular}}{{l1l1l}}
\\toprule
& \\multicolumn{{2}}{{c}}{{Tone}} & \\
\\cmidrule{{2-3}}
& {" & ".join([s.capitalize() for s in event_table.columns])} \\
\\midrule
"""

# 2. build the table rows and columns
table1_rows = df_to_tex(event_table.reset_index())

# 3. build table footer
table1_footer = "\\ \\ \\n\\bottomrule \\n\\end{{tabular}}"

# assemble the text
table1_tex = table1_header + table1_rows + table1_footer

# show
print(table1_tex)

# save for the manuscript
with open("generated/p3_table2.tex", "w") as fh:
    fh.write(table1_tex)

```

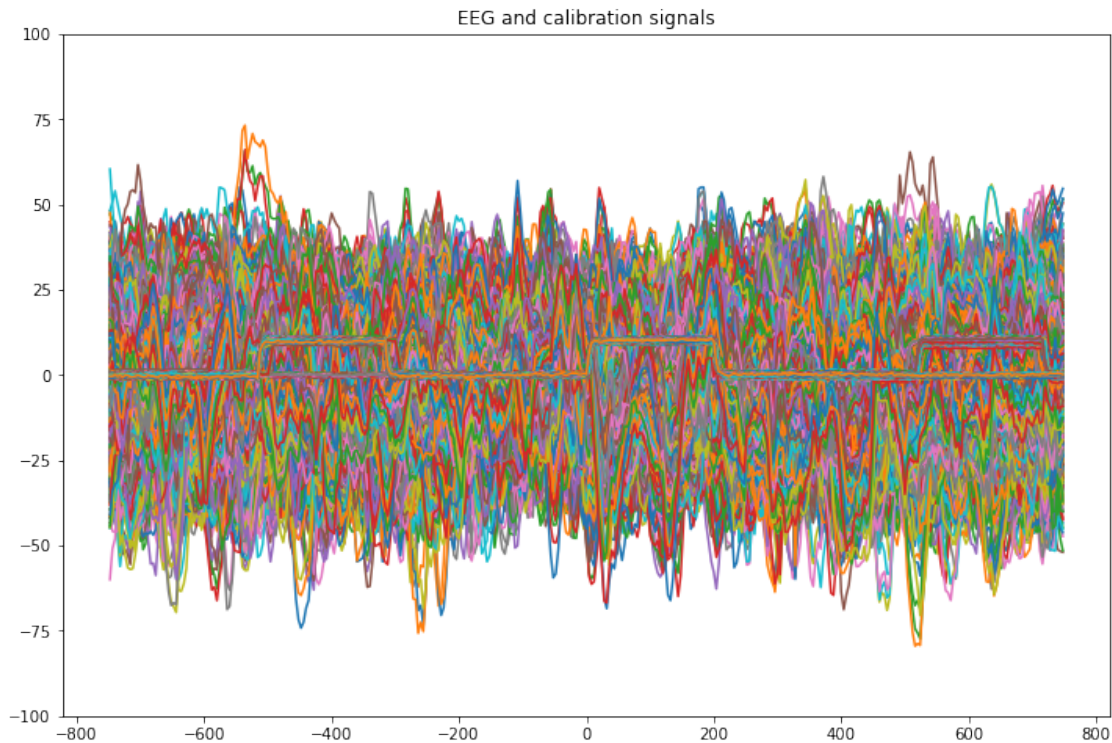
```

\\begin{tabular}{l1l1l}
\\toprule
& \\multicolumn{2}{c}{Tone} & \\
\\cmidrule{2-3}
& Hi & Lo & All \\
\\midrule
Standard & 107 & 94 & 201 \\
Target & 14 & 24 & 38 \\
All & 121 & 118 & 239\\
\\bottomrule
\\end{tabular}

```

6.1 EEG data preview

```
[11]: f_eeg, ax = plt.subplots(figsize=(12, 8))
ax.set_title("EEG and calibration signals")
ax.set_ylim(-100, 100)
times = p3_df.time_ms.unique()
for epoch_id, epoch in p3_df.groupby("epoch_id"):
    ax.plot(times, epoch[EEG_COLUMNS])
```



6.2 Compute time-domain average ERPs

```
[12]: p3_erp = p3_df.groupby(["stimulus", "time_ms"]).mean()[EEG_COLUMNS]
p3_std = p3_df.groupby(["stimulus", "time_ms"]).std()[EEG_COLUMNS]
p3_n = p3_df.groupby(["stimulus", "time_ms"]).count()[EEG_COLUMNS] # n's
    ↳ differs by condition after data QC

for df in [p3_erp, p3_std, p3_n]:
    df.columns.name = "channel"
```

6.3 Example Figure: P300 midline ERP plots with Psychological Science matlab style sheets

<https://www.psychologicalscience.org/publications/aps-figure-format-style-guidelines>

2020-08-11

(emphasis in bold added here)

Details:

Please note that yellow may not show up well, especially in line graphs.

In **all labels including the key(the first letter of each important word and of any word of at least 4 letters should be capitalized.**

Exception: Units of measure indicated in parentheses don't have the first letter capitalized, e.g., "Response Time (ms)."

Minus signs **NOT HYPHENS** should be used to indicate negative numbers or subtraction (a minus sign can be inserted by holding down the key on a computer keyboard while pressing 0, 1, 5, 0 on the number pad, in sequence).

Do not insert a box around a key or a figure.)

A graph should have two axes (ordinate and abscissa) only. Do not include extraneous axes. In mathematical expressions, there should be a single letter space before and after each operator: =, ×, +, ?, <, >, etc.

Exception: Do not insert spaces in subscripts or superscripts.

The **ordinate axis must be labeled to indicate the nature of the quantities referred to**. For example, if a graph shows response times (ordinate) in various conditions (abscissa), the ordinate must be labeled "Response Time," in addition to showing the numerical values.

Numerical values on the ordinate axis should be oriented horizontally. If a figure includes error bars, they must be explained in the caption. In the case of a bar graph, be sure that error bars are easily visible (e.g., a black error bar will be invisible in a data bar with a black or dark-gray fill).

Font style and size:

Labels and numbers in figures should be in **Helvetica Neue 57 Condensed roman font**. (If you do not have this font installed on your device, please use regular **Helvetica** or Arial font.)

Do not use boldface font unless it's intended to highlight something. In that case, the caption should explain what the boldface indicates.

Symbols referring to variables should be in Helvetica Neue 57 Condensed italic font. (If you do not have this font installed on your device, please use regular Helvetica or Arial font.) Otherwise, do not use italics.

Greek letters (e.g., regression coefficients) should not be in italics.

All **ordinate and abscissa** quantities, or any sublabel along the ordinate or abscissa, should be in **9-point** font.

All **main ordinate and abscissa labels** should be in **10-point** font.

The **title** header (at the top of a figure), if there is one, should be in **12-point** font.

Keys should be in **9-point** font.

This includes the height of boxes illustrating fills in a bar graph and symbols used to differentiate lines in a line graph.

Whenever possible, the **key should be placed toward the top of a graph** (i.e., toward the top inside the graph or above the graph, as space allows).

Symbols (e.g., squares, diamonds) plotted in a graph should be no smaller than the corresponding symbols in the key.

Panel labels (a, b, c, etc.) should be in 18-point font, lowercase, positioned to the upper left of the corresponding panels. They should not be followed by periods or surrounded by parentheses.

All other text in graphs (e.g., a label for a graphed line or symbol) should be in **9-point** font.

```
[13]: # seaborn bright
colors = ['#003FFF', '#03ED3A', '#E8000B', '#8A2BE2'] # , '#FFC400', '#00D7FF']

n_colors = len(colors)

psych_sci_fig = {
    # set matplotlib style paramaters to Psych Science specs
    "font.sans-serif": ["Arial", "Helvetica", "DejaVu Sans"],
    "font.size": 18, # default size for panel label
    "axes.labelsize": 10, # X, Y axis labels
    "axes.titlesize": 12, # axes title
    "xtick.labelsize": 9,
    "ytick.labelsize": 9,
    "legend.fontsize": 9,
    "legend.frameon": False,
    "lines.linewidth": 2,
    "lines.markersize": 8,

    # set other aesthetics to taste
    "lines.color": "lightgray",
    "lines.solid_capstyle": "round",
    "lines.dash_capstyle": "round",
    "lines.dashdot_pattern": [6.4, 1.6, 1.0, 1.6],
    "lines.dashed_pattern": [4.0, 5.0],
    "lines.dotted_pattern": [0.01, 2.5],

    "axes.spines.top": False,
    "axes.spines.right": False,
    "axes.spines.bottom": False,
    "axes.spines.left": False,
    "axes.prop_cycle": (
        cycler(lw=["1", "2", "3", "3.5"])
        + cycler(ls=["-", "-", "-", "--"])
    )
}
```



```

}

# this cycles colors from our colorbrewer palette
cco = (cycler(color=colors))

# this "cycles" all black lines
cbw = cycler(color=["k"] * len(colors))

# Figures work in color or black-and-white
panels = {
    "a": {"subtitle": "color", "lines": cco},
    "b": {"subtitle": "black-and-white", "lines": cbw}
}

n_chan = len(EEG_MIDLINE)

for fig_n, (panel, design) in enumerate(panels.items()):
    with plt.style.context(psych_sci_fig):

        # update panel style with line colors
        plt.rcParams["axes.prop_cycle"] = (
            plt.rcParams["axes.prop_cycle"]
            + design["lines"]
        )

        # new figure
        f_ep, axs = plt.subplots(n_chan, 1, figsize=(6, 2 * n_chan),
            ↪sharex=True, sharey=True)

        for axi, chan in enumerate(EEG_MIDLINE):

            ax = axs[axi]

            # zero-lines
            ax.axvline(0, alpha=0.4)
            ax.axhline(0, alpha=0.4)
            ax.text(0.05, 0.9, s=chan, transform=ax.transAxes, fontsize=9)

            # ERP waveforms, line styles from the style sheet
            for stim, erp in p3_erp.query("stimulus != 'cal'").
            ↪groupby(["stimulus"]):
                erp = erp.reset_index()
                time = erp.time_ms.unique()
                ax.plot(time, erp[chan], label=stim)

```

```

# panel label and title
if axi == 0:
    ax.text(-0.1, 1.1, s=f"{panel}", transform=ax.transAxes)
    ax.set_title(f"Auditory Oddball P300 ERP_
→({design['subtitle']})")
    ax.legend(loc="upper right", ncol=2)

ax.set(xlim=(-250, 650))
ax.set(ylim=(-16, 16))

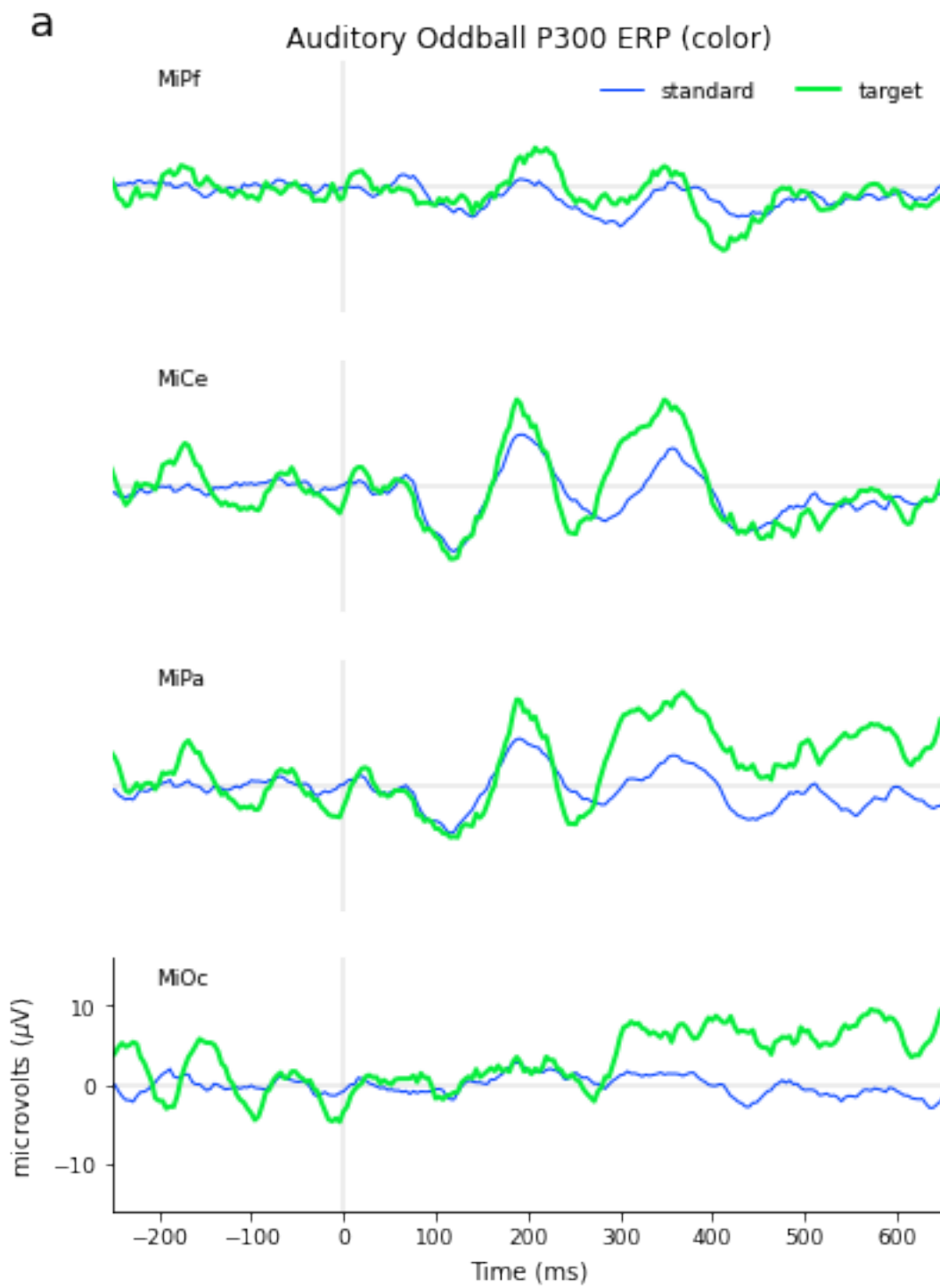
# style the axes
if axi == n_chan - 1:

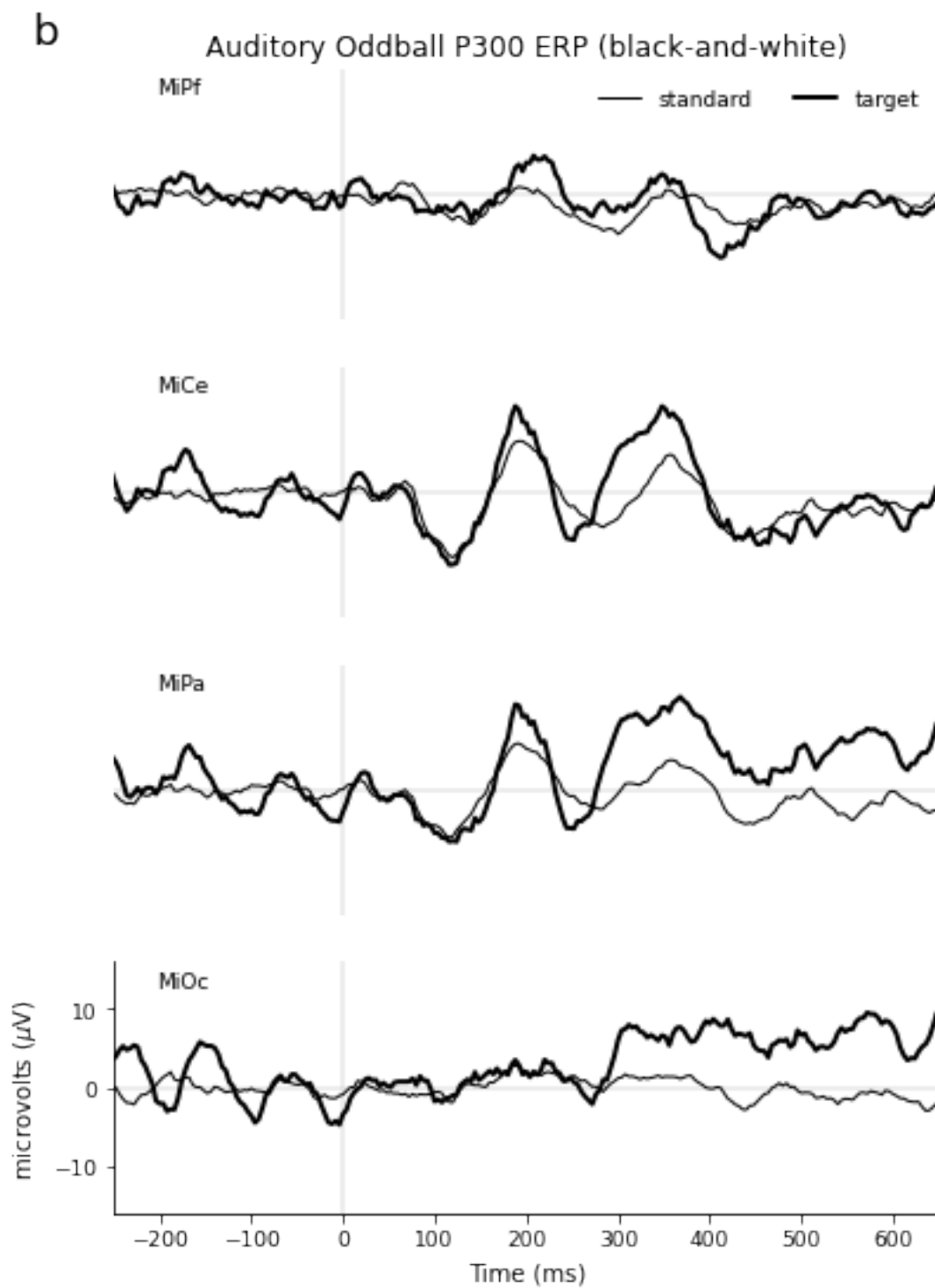
    ax.set_xlabel("Time (ms)")
    ax.spines["left"].set_visible(True)
    ax.spines["bottom"].set_visible(True)

    ax.set_ylabel(r"microvolts ( $\mu\mathrm{V}$ )")
else:
    ax.tick_params(bottom=False, labelbottom=False)
    ax.tick_params(left=False, labelleft=False)

f_ep.tight_layout()
f_ep.savefig(f"generated/p3_midline_plot{fig_n+1}.pdf")

```





7 Plot ERP scalp distribution and decorations

- box highlight an interval with `ax.axvspan(from, to, ...)`

- add uncertainty intervals around $y \pm u$ with `ax.fill_between(x, y1=y + u, y2=y-u, ...)`
- highlight a cond1 vs. cond2 effect in an interval with `ax.fill_between(x, y1=cond1, y2=cond2, where, ...)`

```
[14]: # more styling for bare axes ...
```

```
head_trace_style = {
    "xtick.bottom": False,
    "xtick.labelbottom": False,
    "ytick.left": False,
    "ytick.labelleft": False,
    "axes.prop_cycle": cco,
    "font.size": 9,
}
```

```
# semi-topographic locations
```

```
MPL_32_HEAD = {
    'w': .15,
    'h': .1,
    'chanlocs': {
        'cal': (0.0625, 0.2),
        'lle': (0.25, 0.85),
        'rle': (0.625, 0.85),
        'lhz': (0.0625, 0.85),
        'rhz': (0.8125, 0.85),
        'MiPf': (0.4375, 0.725),
        'MiCe': (0.4375, 0.425),
        'MiPa': (0.4375, 0.275),
        'MiOc': (0.4375, 0.125),
        'LLPf': (0.1875, 0.725),
        'RLPf': (0.6875, 0.725),
        'LMPf': (0.3125, 0.65),
        'RMPf': (0.5625, 0.65),
        'LLFr': (0.0625, 0.5),
        'RLFr': (0.8125, 0.5),
        'LMFr': (0.3125, 0.5),
        'RMFr': (0.5625, 0.5),
        'LDFr': (0.1875, 0.575),
        'RDFr': (0.6875, 0.575),
        'LDCe': (0.1875, 0.425),
        'RDCe': (0.6875, 0.425),
        'LLTe': (0.0625, 0.35),
        'RLTe': (0.8125, 0.35),
        'LMCe': (0.3125, 0.35),
        'RMCe': (0.5625, 0.35),
        'LMOc': (0.3125, 0.2),
    }
}
```

```

        'RMOc': (0.5625, 0.2),
        'LDPa': (0.1875, 0.275),
        'RDPa': (0.6875, 0.275),
        'LLOc': (0.1875, 0.125),
        'RLOc': (0.6875, 0.125),
        'A2': (0.8125, 0.2)
    }
}

MPL_MIDLINE = {
    'w': .75,
    'h': .2,
    'chanlocs': {
        'MiPf': (0.1, 0.7),
        'MiCe': (0.1, 0.5),
        'MiPa': (0.1, 0.3),
        'MiOc': (0.1, 0.1),
        'cal': (0.1, 0.1),
    }
}

```

7.1 Define the decorations

```

[15]: # timeline, ticks, and labels
tmin, tmax = -200, 600
timeline_ticks = [-200, 0, 200, 400, 600]
timeline_ticklabels = [-200, 0, 200, 400, "600 ms"]

# cal bar in x, y data units
cal_bar_time = 0 # ms
cal_bar_min = 0 # uV
cal_bar_max = 5 # uV
cal_tick_width = 25 # ms

# cal bar line aesthetics
cal_bar_kws = {"color": "black", "lw": 1}

# cal bar label kwargs
cal_bar_label = {
    "x": cal_bar_time + cal_tick_width,
    "y": cal_bar_max / 2.0,
    "s": f"{cal_bar_max}" + r"$\mu\mathrm{V}$",
    "ha": "left",
    "va": "center",
}

```

```

# channel label kwargs, label text is per channel
chan_label = {
    "x": cal_bar_time,
    "y": cal_bar_max ,
    "ha": "center",
    "va": "bottom",
}

# -----
# montage and figure proportions

# chans = EEG_MIDLINE + ["cal"]
# chan_layout = MPL_MIDLINE

chans = EEG_COLUMNS + ["cal"]
chan_layout = MPL_32_HEAD
figsize = (16, 14)

# conditions to plot, add "cal" for fun
plot_stim = ["standard", "target"]

```

```

[16]: # plot it
with plt.style.context([psych_sci_fig, head_trace_style]):

    fig, axs = plt.subplots(len(chans), figsize=figsize, sharey=True,
        ↳sharex=True)

    # proportions
    chan_width = chan_layout["w"] # .2
    chan_height = chan_layout["h"] # .1

    for axi, chan in enumerate(chans):

        # axis
        ax = axs[axi]
        ax.patch.set_alpha(0.0) # see through
        ax.set_xlim(tmin, tmax)

        # lower left corner for this channel
        x0, y0 = chan_layout["chanlocs"][chan]

        # locate this channel
        bbox = mpl.transforms.Bbox([x0, y0], [x0 + chan_width, y0 +
        ↳chan_height]))
        ax.set_position(bbox)

```

```

# ERP waveforms, line styles from the style sheet
for stim, erp in p3_erp.query("stimulus in @plot_stim").
↳groupby(["stimulus"]):

    # all axes get timeline, vertical cal bar
    ax.axhline(0, color='lightgray')
    ax.plot(
        [0, 0],
        [cal_bar_min, cal_bar_max],
        **cal_bar_kws
    )

    # -----
    # special handling for cal and timeline
    if chan == "cal":
        ax.spines["bottom"].set_position(("data", 0))
        ax.set_xticks(timeline_ticks)
        ax.set_xticklabels(timeline_ticklabels)
        ax.tick_params(bottom=True, labelbottom=True)
        ax.plot(
            [cal_bar_time, cal_tick_width],
            [cal_bar_max, cal_bar_max],
            **cal_bar_kws
        )
        ax.text(**cal_bar_label)
        continue

    # -----
    # ERP label and traces
    ax.text(s=chan, **chan_label)
    erp = erp.reset_index()
    time = erp.time_ms.unique()
    ax.plot(time, erp[chan], label=stim)

    # Example: highlight P300 effect
    if stim == 'target':
        # pick one condition, fill to the other
        y2 = p3_erp.query("stimulus=='standard'")[chan]
        when = (time >= 250) & (time < 400) # highlight interval
        ax.fill_between(
            time,
            y1=erp[chan],
            y2=y2,
            where=when,
            color="magenta",
            alpha=.3
        )

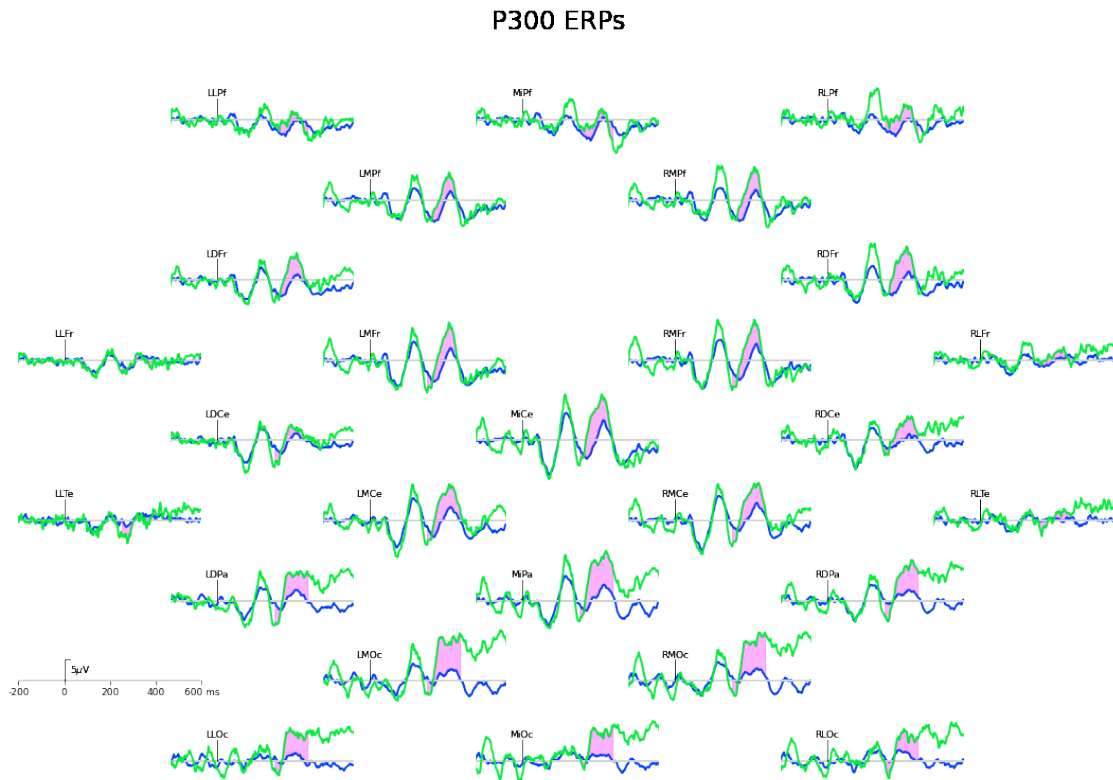
```



```

# set the title on the way out, ax doesn't matter, position is in fig_
→ coords.
ax.text(x=.45, y=.85, s="P300 ERPs", size=24, transform=fig.transFigure)

```



8 Compute mean P300 ERP

```

[17]: # compute mean amplitude 250 - 450 ms, standards, targets, and_
→ difference=target - standard
p300_amp = (
    p3_erp.query(
        "stimulus != 'cal' and time_ms >= 250 and time_ms < 450"
    ).groupby("stimulus")[EEG_COLUMNS]
    .mean()#
    .T
    .unstack()
    .to_frame()
)
p300_amp.columns = ["amplitude"]

```

```
# compute the P300 effect: target - standard
p300_amp_diff = p300_amp.unstack(0).apply(lambda row : row[1] - row[0], axis=1).
↳to_frame()
p300_amp_diff.columns = ["amplitude"]
p300_amp_diff["stimulus"] = "difference"
p300_amp_diff = p300_amp_diff.reset_index().set_index(["stimulus", "channel"])

p300_amp = pd.concat([p300_amp, p300_amp_diff])
p300_amp
```

```
[17]:
```

		amplitude
stimulus	channel	
standard	MiPf	-2.322935
	LLPf	-1.924552
	LLFr	-0.239627
	LLTe	0.564894
	LLOc	0.213780
...	...	
difference	LMCe	3.214108
	MiPa	5.914676
	RMCe	2.656957
	RMFr	1.931009
	MiCe	3.280381

[78 rows x 1 columns]

8.1 Merge P300 mean amplitude with electrode locations

```
[18]: p300_amp_locs = (
    p300_amp.reset_index("stimulus")
    .merge(
        SPH_CART_LOCS[["channel", "x_lambert", "y_lambert"]],
        on="channel"
    )
)
p300_amp_locs
```

```
[18]:
```

	channel	stimulus	amplitude	x_lambert	y_lambert
0	MiPf	standard	-2.322935	6.123234e-17	1.000000
1	MiPf	target	-2.309737	6.123234e-17	1.000000
2	MiPf	difference	0.013199	6.123234e-17	1.000000
3	LLPf	standard	-1.924552	-5.877853e-01	0.809017
4	LLPf	target	-1.445431	-5.877853e-01	0.809017
..
73	RMFr	target	0.246611	1.869914e-01	0.257372
74	RMFr	difference	1.931009	1.869914e-01	0.257372
75	MiCe	standard	-1.254791	0.000000e+00	0.000000

76	MiCe	target	2.025590	0.000000e+00	0.000000
77	MiCe	difference	3.280381	0.000000e+00	0.000000

[78 rows x 5 columns]

```
[19]: head_plot_style = {
    "axes.xmargin": 0.1,
    "axes.ymargin": 0.1,
    "axes.spines.left": False,
    "axes.spines.bottom": False,
    "xtick.color": "none",
    "ytick.color": "none",
    "lines.markersize": 20
}

# set up the color mapping
lower, upper = -11, 11
n_shades = 10 # for each color

n_colors = (2 * n_shades) + 2
bounds = np.linspace(lower, upper, n_colors + 1)
bwr_norm = mpl.colors.BoundaryNorm(bounds, n_colors)

# get blue-white-red divergent colormap
bwr_cmap = mpl.cm.get_cmap('bwr', n_colors)

with plt.style.context([psych_sci_fig, head_plot_style]):
    fig, axs = plt.subplots(1, 3, figsize=(14, 4),)

    stimulus = ["standard", "target", "difference"]
    for axi, stim in enumerate(stimulus):
        data = p300_amp_locs.query("stimulus == @stim")
        ax = axs[axi]
        if stim == "difference":
            ax.set_title("P300 Effect (Target - Standard)")
        else:
            ax.set_title(f"{stim.capitalize()} P300")
        p = ax.scatter(
            data["x_lambert"],
            data["y_lambert"],
            c=data["amplitude"],
```

```

        marker="o",
        cmap = bwr_cmap,
        norm=bwr_norm,
        lw=.5,
        edgecolor='k'
    )
    ax.set_aspect(0.9)

axins = axs[-1].inset_axes([1.2, 0, .075, 1])
cb = fig.colorbar(
    p,
    cax=axins,
    ticks=bounds,
)
cb.ax.tick_params(axis="y", color='k')
cb.ax.set_yticklabels(bounds, color='k')
#cb.ax.yaxis.set_major_formatter(mpl.ticker.StrMethodFormatter("{x:5.1f}"))
cb.ax.yaxis.set_major_formatter(mpl.ticker.StrMethodFormatter("{x:5.1f}"))
cb.ax.text(
    x=0.5,
    y=1.05,
    s=r"$\mu\mathrm{V}$",
    fontsize=9,
    transform=cb.ax.transAxes,
    ha="center"
)
fig.savefig("generated/p3_head_plot3.pdf", format="pdf",
→bbox_inches="tight")

```



Source: research_report.tex

This is the LaTeX for the main report.

```

1  % for PsychSci APA6 TeXLive 2020 use this with biber/biblatex + styel=apa6
2  % figure note is not supported, put it in the caption
3  \documentclass[helv,10pt,man,floatsintext]{apa6} %% man <-> jou <-> doc
4  \usepackage{csquotes}
5  \usepackage[backend=biber,style=apa6]{biblatex}
6  \addbibresource{apa_ms.bib}
7
8  % if you like line numbers ...
9  \usepackage{lineno}
10 %\linenumbers
11
12 \usepackage[american]{babel}
13 % \usepackage[utf8x]{inputenc}
14 \usepackage[utf8]{inputenc}
15 \usepackage{amsmath}
16 \usepackage{graphicx}
17 \usepackage{multirow}
18 \usepackage{multicol}
19 \usepackage{xcolor}
20
21
22 % for tracking changes
23 \usepackage[draft]{changes}
24 \definecolor{skyblue2}{rgb}{.203, .395, .640}
25 \definecolor{orange2}{rgb}{.957, .473, .000}
26 \definecolor{plum2}{rgb}{.457, .313, .480}
27 \definechangesauthor[name=TPU, color=skyblue2]{TPU}
28 \definechangesauthor[name=ABC, color=orange2]{ABC}
29 \definechangesauthor[name=XYZ, color=plum2]{XYZ}
30
31
32 % to include one or more pages of multipage pdfs
33 \usepackage{pdfpages}
34
35 % for cross-references back to the main doc
36 % use \zref{} and \zlabel{} instead of latex native \ref{} and \label{}
37 \usepackage[xr, user, titleref]{zref}
38 \zexternaldocument{apa_si} % other .tex file to cross reference
39
40 % to help control location of figures and tables
41 % \usepackage{float}
42
43 % highlight computer source code
44 \definecolor{bgc}{rgb}{.96,.96,.96}
45 \usepackage{minted}
46 \setminted[latex]{
47   xleftmargin=0.5in,
48   xrightmargin=0.5in,
49   style=bw,
50   frame=none, % lines,
51   bgcolor=bgc,

```

```

52     fontsize=\footnotesize,
53     linenos
54 }
55
56 % for clickable URL links in pdfs
57 \usepackage{hyperref}
58 \hypersetup{
59     colorlinks=true,
60     citecolor=blue,
61     linkcolor=blue,
62     filecolor=blue,
63     urlcolor=blue,
64 }
65
66 % use this to prevent LaTeX errors when urls break across pages
67 %% \hypersetup{draft}
68
69 \title{Open science with style: A reproducible reproducible research report}
70
71 \shorttitle{Open science with style}
72
73 \author{Thomas P. Urbach}
74 \lefthead{Urbach}
75
76 \affiliation{
77     Cognitive Science Department \\\
78     University of California, San Diego \\\
79     \today
80 }
81
82 \abstract{When the culmination of research is a research report, the
83     culmination of reproducible research must be a reproducible
84     report. To accomplish this, three problems must be solved: 1) the
85     results of the reproducible data analysis must be incorporated into
86     the narrative text, tables, and figures of the document; 2) the
87     document must comply with the byzantine typographical requirements
88     of professional publication style guides and their idiosyncratic
89     modifications by various publishers; 3) the different parts and
90     pieces of the report (manuscript, supplementary information,
91     figures, tables, captions) must be reproducible digital objects in
92     whatever specific document and image file format is required by the
93     online platforms for submission to the journal and production by the
94     publisher. This report describes and demonstrates a flexible and
95     generalizable approach that combines freely available open source
96     data analysis and document preparation software tools to solve these
97     three problems. The report itself is reproducibly generated by the
98     approach it describes and demonstrates for psychologists with
99     real-world examples: the manuscript is formatted in American
100     Psychological Association style and the digital objects are
101     generated as required for the online submission and production
102     platforms used by {\it Proceedings of the National Academy of
103     Sciences}. The source code is publicly available and may be cloned
104     from the GitHub repository or downloaded from the Open Science
105     Foundation archive and freely modified or adapted for non-commercial
106     purposes under the Creative Commons

```

```
107 Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA
108 4.0) license. This reproducible report, together with the source
109 code that reproduced it, comprise a complete self-contained
110 tutorial, demonstration, and template for general use. }
111
112 \begin{document}
113 \maketitle
114
115 \section{Introduction}
116
117 For any research project, after all the work of experimental design,
118 implementation, and data acquisition are in place, and the data
119 analysis is complete, there still remains the task of preparing and
120 publishing the peer-reviewed research report with a clear and accurate
121 presentation of the results through the text, tables, and
122 figures. However, the ``research report'' is an abstraction; in
123 practice it takes various forms on its trajectory from the authors'
124 desks to dissemination as a journal article in print and online in
125 digital form(s). For the authors, there all the usual chores of
126 document preparation: Writing the narrative text with qualitative and
127 quantitative analysis results, creating high-resolution graphics for
128 figures, preparing tables of data and results, adding and deleting
129 citations and bibliographic references, embedding links to URLs, and
130 aligning cross-references to elements within or across documents,
131 e.g., to the separate online supplementary information. During
132 preparation and revision the report is in flux and must be editable
133 with changes to the text tracked across versions. For pre-print
134 archives and (re-)submission to peer-reviewed journals the text and
135 graphics are composited into a usually un-editable but easily
136 transmissible and viewable digital snapshot, e.g., typically Portable
137 Document Format (PDF). Finally, for journal and book publishers, the
138 process is unwound and the report must be comprised of separate
139 editable text and ``camera ready'' high-resolution graphics suitable
140 for production in digital form for online viewing and print
141 form. Throughout these transformations for publication, the report
142 must also satisfy specific style requirements and for psychologists
143 this often means a variation of the 6th Edition of
144 the Publication Manual of the American Psychological
145 Association~(\cite{APAStyle6th}). Or maybe the 7th Edition. In short,
146 as a research report evolves from inception to DOI, it must sometimes
147 change and other times freeze in various highly specific forms and
148 digital file formats as it passes through different hands with
149 different requirements.
150
151 When the goal of reproducible research is fully embraced, the
152 ``research report'' must also be reproducible throughout these stages
153 of preparation, revision, submission, and production. This requires
154 solving three problems: 1) the results of the reproducible data
155 analysis must be incorporated into the narrative text, tables, and
156 figures of the document; 2) the document must comply with the
157 byzantine typographical requirements of professional publication style
158 guides and their idiosyncratic modifications by various publishers; 3)
159 the different parts and pieces of the report (manuscript,
160 supplementary information, figures, tables, captions) must be
161 reproducible digital objects in whatever specific document and image
```

file format is required by the vagaries of an online journal submission platform and then subsequently by a different online production platform. Solutions to each of these problems individually abound, the challenge is to combine them reproducibly. For instance, reproducible data analyses are becoming commonplace though the use of scientific computing platforms and open source scripting languages like Python and R encapsulated in virtual environments (conda, virtualenv) and containers (Docker, singularity). However the technology for solving the data analysis problem is decoupled from the strict typesetting requirements of different publication styles. On the other hand, mature document preparation software like Microsoft Word and `\LaTeX{}` provide the fine-grained control of formatting necessary to comply with idiosyncratic style guidelines. However, typing or copy-pasting the results decouples the report from the analysis. The results of the analysis may be reproducible when the analysis is revised by co-authors or reviewers, but the results do not propagate to all the digital objects that comprise the parts and pieces of the report for (re-)submission and production.

This self-reproducing tutorial describes and demonstrates one approach to solving all three problems at once using mature freely available open-source computer software, a working knowledge of `\LaTeX{}`~(`\cite{latexproject}`), and no more knowledge of computer programming than is already required to implement the reproducible data analysis it reports. The tutorial includes a sample reproducible data analysis pipeline with open-access data but focuses mainly on the reproducible report per se, i.e., solutions to the second and third problem needed to bridge the gap between the end of the reproducible data analysis and the DOI of the peer-reviewed publication in an academic journal. In addition to programmatically combining the data analysis results with the narrative text, tables, and figures of the report, the complete `\LaTeX{}` source code listings in the Supplementary Materials provide working examples of some features generally useful for manuscript preparation: tracking changes across revisions, preparing camera ready graphics, automating cross-references within and between documents, formatting and masking the citations and bibliography, generating Portable Document Files, compositing documents and pieces of documents in text and PDF file formats, and preparing an author's manuscript for distribution while a published article is embargoed. The Supplementary Information provides instructions for installing the open source software required to reproduce the data analysis and this report. The complete source code for the data analysis and report generation is publicly available and may be downloaded from the Open Science Foundation archive or cloned from the GitHub repository under a Creative Commons CC BY 4.0 license~(`\cite{ccby4.0}`) and used as a template and freely modified for other purposes with appropriate attribution.

`\section{Method}`

`\begin{figure}[ht]`

`\caption{Generating a reproducible APA 6th style research report: 1)`

Executing the reproducible data analysis code generates the complete results which appear as-is in the Supplementary

Information. Selected results to be reported in the manuscript are


```

217     exported to separate files as minimally styled narrative text and
218     tables, and PDF graphics. 2. The graphics exported by the analysis
219     are converted to camera ready APA-style figure graphic PDFs for the
220     manuscript. 3. The Supporting information \LaTeX{} file is typeset
221     as a document PDF which includes the complete analysis source,
222     results, graphics, and document source. 4. The \LaTeX{} manuscript
223     is typeset as a document PDF which includes the results text
224     generated by the data analysis, the camera ready PDF figures, and
225     bibliography.} \zlabel{ms:report_generation}
226 \includegraphics[width=.95\textwidth]{images/report_generation.png}
227
228 \end{figure}
229
230
231 This approach to generating reproducible research reports requires
232 the four main components, outlined schematically in
233 Figure~\zref{ms:report_generation}. While the approach is flexible and
234 generalizable, the specific examples are selected for researchers in
235 Psychology and demonstrate how to satisfy all the requirements (except
236 word count) for submitting and publishing a research report in the
237 journal, {\it Psychological Science}. Accordingly the manuscript is
238 structured with a Cover Page, Abstract, Introduction, Method, Results,
239 and Discussion~\parencite{APSStructStyle} and formatted according to
240 the APA 6th edition style~\parencite{APAStyle6th}. The approach here
241 is readily adapted to the APA 7\textsuperscript{th} Edition with a
242 change of the document
243 class~({\href{https://www.overleaf.com/project/5f3053af0af0dc00016f191b}{apa7}})
244 and minor modifications to the text described in Supporting
245 Information. The approach generalizes to other publication styles for
246 which \LaTeX{} style files have been defined. A conveniently inventory
247 is collected here:
248 \href{https://www.overleaf.com/latex/templates/tagged/academic-journal}{Overleaf.com
249 Templates\textmdash Academic Journal}. Many styles are community
250 contributions, for instance,
251 \href{https://www.overleaf.com/latex/templates/tagged/arxiv}{arXiv,
252 bioRxiv}.
253 %
254 A number of journals and publishers provide official styles, such as
255 %
256 \href{https://www.overleaf.com/latex/templates/tagged/npg}{Nature},
257 \href{https://www.overleaf.com/latex/templates/tagged/pnas}{Proceedings of the National Academy of Sciences},
258 \href{https://www.overleaf.com/latex/templates/tagged/elife-official}{eLife}
259 %
260 and publishers
261 %
262 \href{https://www.overleaf.com/latex/templates/tagged/cup-official}{Cambridge University Press},
263 \href{https://www.overleaf.com/latex/templates/oup-general-template/fqkhyshcbpwv}{Oxford University Press},
264 \href{https://www.overleaf.com/latex/templates/tagged/springer}{Springer}
265 %
266 including
267 %
268 \href{
269 https://www.overleaf.com/latex/templates/a-demonstration-of-the-latex2e-class-file-for-sage-publications/jcd
270 }{
271 SAGE

```

```

272 },
273 %
274 the publisher of {\em Psychological Science}.
275 %
276 The Supporting Information for this report provides installation
277 instructions for the necessary software and complete source code
278 listings for the analyses, documents, and figures which are freely
279 available under the CC-BY-4.0 license and may serve as templates for
280 a range of research projects in the psychological sciences.
281
282
283 \subsection{Data analysis pipeline: \mintinline{bash}{apa_analysis.ipynb}}
284
285 For demonstration purposes, a toy reproducible data analysis pipeline
286 is implemented in a Jupyter notebook running a Python
287 kernel~(\cite{kluEtAl2016}). The pipeline (down)loads and
288 transforms a sample EEG dataset~(\cite{Urbach2020z}), computes summary
289 measures, and generates figures and text output. The particulars are
290 incidental, the data may as well be response times and the analysis
291 could be implemented in R, MATLAB, or any language that can format
292 numerical values as strings, write string variables to a text file,
293 and export as PDF, EPS, PNG, JPEG (or a format programmatically
294 convertible to one of these). This PDF is used for vector graphics and
295 PNG for raster graphics in this report since these have proved
296 reliable and both support transparency; EPS and JPEG also work if
297 these are required by the publisher.
298
299
300 \subsection{Preparing camera ready figures with \LaTeX{} and Ti{\it k}Z}
301
302 Ideally, graphic images generated by an analysis pipeline will be in
303 final ``camera ready'' form but this is not always practical or
304 possible. A figure may require annotations, e.g., math notation, not
305 supported by the figure generator and a multipanel figure may need to
306 combine images from different sources. To demonstrate how this may be
307 done programmatically for reproducibility, three of the ``rough'' plot
308 graphics generated by the analysis pipeline are reconfigured, annotated
309 and converted into two camera-ready APA-style manuscript figures
310 (Figure~\zref{ms:multipanel} and Figure~\zref{ms:tikzfig}) using \LaTeX{} and
311 the Ti{\it k}Z graphic library without additional software or manual
312 editing.
313
314
315 \subsection{Manuscript: \mintinline{bash}{apa_ms.tex}}
316
317 LaTeX{} is a form of markup language where
318 the document text is intermingled with short typesetting
319 instructions. For instance, {\it this phrase is typeset in italics},
320 and the instruction looks like this:
321 %
322 \mintinline{latex}{\it this phrase is typeset in italics}}.
323 %
324 Mathematical symbols and more complex equations are very
325 well-supported and set in the same way, e.g., partial eta squared
326 ( $\eta_p^2$ ) is set like so: \mintinline{latex}{ $\eta_p^2$ }. Other

```

```

327 instructions are more general. For instance, the manuscript document
328 begins with this,
329 \mintinline{latex}{\documentclass[man, helv, 10pt, draftall, floatsintext]{apa6}},
330 that says to typeset the document as a manuscript, in Helvetica 10
331 point font with a draft watermark on all pages, formatted to the APA
332 6th Edition style except that tables and figures should be placed near
333 where they appear in the text ('floatsintext') rather than collected
334 at the end. This style, including the deviation from the APA 6th table
335 and figure position, corresponds to the submission guidelines for
336 Psychological Science~\parencite{PsychSciSubmissions2020}. Like all
337 \LaTeX{} files, the main manuscript file is a plain text document and
338 thus virus-free, portable, viewable, and editable with any text
339 editor, although one that supports LaTeX syntax highlighting
340 on-the-fly syntax error checking is strongly recommended.
341
342 \subsubsection{Supplementary Information: \mintinline{bash}{apa_si.tex}}
343
344 Supplementary Information is as much a part of the report as the
345 manuscript and must be likewise reproducible. For demonstration here,
346 the Supplementary Information is comprised of a separate \LaTeX{}
347 file. It provides instructions for downloading this report from
348 public repositories and installing the software to reproduce it. It
349 also includes source code listings of the Makefile used to reproduce
350 portions or all of the analysis, source code and output of the entire
351 executed analysis Jupyter notebook and listings of all the \LaTeX{}
352 files used to generate the report, figures, and supporting
353 information, which includes the self-reflexive listing of the
354 Supporting Information listing itself.
355
356
357 \subsection{Reproducing the report: \mintinline{bash}{Makefile}}
358
359 The \mintinline{bash}{make} program is a widely used command line
360 utility for managing the execution of a interdependent computer code
361 in complex programming projects, where changes in one file may might
362 impact some but not all other files. Reproducible data analysis and
363 report generation is similar in that, e.g., generating the
364 camera-ready figure PDFs depends on the rough plots generated by the
365 analysis which in turn depends on executing the analysis. The make
366 utility provides a useful mechanism for expressing the
367 interdependencies and compartmentalizing the project as work
368 progresses, e.g., \mintinline{bash}{make analysis} or
369 \mintinline{bash}{make fig2} or \mintinline{bash}{make ms} while
370 \mintinline{bash}{make all} ensures that all the components execute in
371 the correct order to completely reproduce the analysis and generate
372 all the files and documents for the figures, manuscript, supporting
373 information. Here is a summary of the make file components for
374 generating this report, execution times are for a high performance
375 workstation.
376
377 \begin{description}
378
379 \item [\mintinline{bash}{make analysis} (45 s)] Reproduce the data analysis by
380 executing all the computer code in the analysis notebook start to
381 finish. This has four side effects:

```

```

382
383 \begin{enumerate}
384   \item The data analysis computations are executed and the results captured
385     as standard output and plots in the Jupyter notebook cells.
386   \item Results to be included in the manuscript as narrative text and
387     tables are embedded in text strings, minimally formatted to APA
388     style with \LaTeX{}, and exported as separate text files (.tex).
389   \item Plots to be included in the manuscript figures are exported as
390     PDF graphics.
391   \item After execution is complete, a snapshot of the complete
392     notebook\textemdash text, computer code, and results captured in
393     the output cells\textemdash is exported to a PDF file. The PDF is
394     included in its entirety in the Supplementary Information.
395 \end{enumerate}
396
397 \item [\mintinline{bash}{make fig1} (1 s)] Run
398   \mintinline{latex}{pdflatex fig1.tex} to convert two rough plot
399   graphics as generated by the analysis pipeline into the camera-ready
400   Figure~\zref{ms:multipanel} graphic shown in the manuscript.
401
402 \item [\mintinline{bash}{make fig2} (1 s)] Run \mintinline{latex}{pdflatex fig2.tex}
403   to convert the rough plot graphic generated by the analysis
404   pipeline into the camera-ready Figure~\zref{ms:tikzfig} graphic shown in the
405   manuscript.
406
407 \item [\mintinline{bash}{make figs} (47 s)] Execute the analysis to generate the rough PDF graphic
408   output files then make fig1 and fig2 as above.
409
410 \item [\mintinline{bash}{make ms} (9 s)] Run \mintinline{bash}{pdflatex apa_ms.tex}
411   to generate the manuscript PDF.
412
413 \item [\mintinline{bash}{make si} (4 s)] Run \mintinline{bash}{pdflatex apa_si.tex}
414   to generate the Supporting Information PDF.
415
416 \item [\mintinline{bash}{make all}] Run make figs to execute the
417   analysis and generated camera ready figures then make ms and si
418   enough times to update and the cross-references between the
419   manuscript and supplementary information.
420
421 \end{description}
422
423
424 \label{sec:results}
425
426 \section{Results}
427
428 The results are this report and the Supplementary Information. Both
429 are reproducibly reproduced using freely available open source
430 software, a working knowledge of \LaTeX{} and no more computer
431 programming than the Python used for the data analysis. A few points
432 merit further discussion.
433
434 \section{Discussion}
435
436 \subsection{Linking data and arbitrary text}

```

```

437
438 % This is the complete latex for this entire section
439 \input{generated/arbitrary_text.tex}
440
441
442 The listing below shows the minimally styled \LaTeX{} text generated
443 by the analysis pipeline. For illustration, it includes comments
444 (\%), narrative text with the data values filled in programmatically,
445 and \mintinline{latex}{\it N }, which italicizes the capital N
446 according to APA 6th style:
447
448 % this shows it as a source listing
449 \inputminted{latex}{generated/arbitrary_text.tex}
450
451
452 \subsection{Tables}
453
454 The ability to link data with arbitrary text is nowhere more valuable
455 than in preparing reproducible data tables styled to editorial
456 standards. The primary challenges are the intricate requirements for
457 laying out headings and notes as illustrated by the following excerpts,
458 drawn from the 40 pages of APA Publication Manual 7th edition table
459 guidelines:
460
461 \begin{quote}
462 {\bf headings} Tables may include a variety of headings depending on
463 the nature and arrangement of the data. All tables should include
464 column headings, including a stub heading (heading for the leftmost
465 column). Some tables also include column spanners, decked heads, and
466 table spanners (see Section 7.12)
467
468 \ldots
469
470 {\bf notes:} These types of notes (general, specific, and probability)
471 appear below the table as needed to describe contents of the table
472 that cannot be understood from the table title or body alone \ldots
473 \end{quote}
474
475 \noindent
476 It is straightforward to reproducibly link table text to the analysis
477 data they tabulate. It is less straightforward, but still tractable to
478 do while also generating the three types of notes, four types of
479 headings and column spanners, and a border at the top and bottom of
480 the table, beneath column headings (including decked heads), and above
481 column spanners.' (p. 205)
482
483 The tabular exhibit labeled Table~\zref{ms:table1} illustrates a
484 not-quite conforming tabular array of data. When the analysis runs,
485 the table is reproducibly generated as a \LaTeX{} .tex file with one
486 line of code \mintinline{python}{pandas.DataFrame.to_latex()}.
487 \footnote{
488   For analyses scripted in R, the \mintinline{R}{xtable} library
489   similarly generates \LaTeX{} format table from dataframes
490   \url{https://cran.r-project.org/web/packages/xtable/index.html}.
491 }

```

```

492 The .tex file is imported into the manuscript the same way as the arbitrary
493 text file above.
494
495 \begin{table}[ht]
496   \centering
497   \caption{A non-APA Style data table and note generated
498     as \LaTeX{} by calling \mintinline{python}{pandas.DataFrame.to_latex()}.} \zlabel{ms:table1}
499   \begin{threeparttable}
500     \input{generated/p3_table1.tex}
501     \begin{tablenotes}[flushleft]
502       Note: Python variables are conventionally lower case.
503     \end{tablenotes}
504   \end{threeparttable}
505 \end{table}
506
507 \noindent
508 This approach is simple and easy and well-suited for data tables
509 presented in supporting information where styling requirements are
510 typically less strict. When easily generated tables will not do, the
511 fall back is arbitrary text generation. A few lines of Python code
512 and common string formatting methods suffice to generate the \LaTeX{}
513 required to format the table header, footer, notes and row data to APA
514 style. The following listing shows the programmatically generated
515 \LaTeX{}, the result is shown as Table~\zref{ms:table2}. The Python
516 source code to is Jupyter notebook in the-Supporting
517 Information.
518
519 \inputminted{latex}{generated/p3_table2.tex}
520
521
522 \begin{table}[ht]
523   \centering
524   \caption{An APA style data table and note generated as \LaTeX{} with
525     a few lines of pure Python.}
526   \zlabel{ms:table2}
527
528   \centering
529   \begin{threeparttable}
530     \input{generated/p3_table2.tex}
531     \begin{tablenotes}[para, flushleft]
532       Note: APA Style capitalization.
533     \end{tablenotes}
534   \end{threeparttable}
535 \end{table}
536
537
538
539 \subsection{Figures}
540
541 Graphics figures in PNG, PDF, and JPEG can be included in a \LaTeX{}
542 document with the \mintinline{latex}{command}. Of these PDF seems to
543 be the most reliable for vector graphics (plots, line drawings,
544 charts, plots) and PNG for raster graphics. Including figures is
545 straightforward, creating figures for a data analysis reproducibly is
546 another matter. In some case it may be possible to generate

```

camera-ready graphics from the data analysis pipeline itself. Although this takes some effort to fine tune at the outset when Reviewer 2 insists on some mid-stream revision that requires re-running the analysis, the change propagates all the way through to the final figures included in the report. However this is not always possible. One recourse is to use an interactive vector graphics manipulation programs like Inkscape to import the graphic and edit to style but, like manually typing results into a data table, the results may change but the representation of the results does not.

Since hand editing figures amounts to using a mouse to select a sequence of drawing commands, it can be done programmatically with the right vector graphics manipulation tools. In the LaTeX ecosystem, a particularly powerful package for this is `\href{https://en.wikipedia.org/wiki/PGF/TikZ}{Ti{\it k}Z}` and the learning curve is correspondingly steep. However, for simple tasks like laying out and annotating the figures, it is reasonably straightforward. The tikz figure is a canvas with coordinates. Graphics can be placed and aligned, and drawing elements like lines, arrows, and shading added. Figure~\zref{ms:multipanel} and Figure~\zref{ms:tikzfig} are worked examples of this approach and show how to convert graphics generated by the data analysis into ``camera ready'' figures to APAstyle specifications saved as separate PDF files for upload to the publisher. Figure~\zref{ms:multipanel} is a simple example that lays out two graphics side by side and Figure~\zref{ms:tikzfig} illustrates a more elaborate example that selects portions of a single graphic, rearranges and resizes them and adds additional graphic and text annotations. The LaTeX and Ti{\it k}Z code for both figures is listed in the Supplemental Information.

```

\begin{figure}[ht]
  \caption{
    A complete multi-panel color figure generated
    reproducibly from the data to Psychological Science figure
    specifications. The figure is generated using the matplotlib package in
    Jupyter Notebook running a Python kernel. The code illustrates
    some useful Python idioms and matplotlib functionality including
    style sheets, the style context manager, how to lay out panels,
    add labels including with mathematical symbols, and export the figure as
    as a PDF graphic.
  }
  \zlabel{ms:multipanel}
  \centering
  \includegraphics[width=.95\textwidth]{apa_fig1.pdf}
\end{figure}

\begin{figure}[ht]
  \caption{Reproducible figure layout and annotation. Panel a shows
    the pdf as generated by the analysis script and a stock montage
    image. Panel b shows the ``camera ready'' figure output generated
    by post-processing the generated graphic with LaTeX and the

```



```

602     Ti{\it k}Z drawing library as part of the documentation generation
603     pipeline. The data are the same as in Figure~\zref{ms:multipanel}
604     }\zlabel{ms:tikzfig} \includegraphics[width=\textwidth]{apa_fig2.pdf}
605 \end{figure}
606
607
608 %% % Figure 2
609 %% \begin{figure*}[ht]
610 %% \centering
611 %% \includegraphics[width=0.9\textwidth]{fig2.pdf}
612
613 %% \caption{
614 %%   Simple resizing and clipping can be done in LaTeX{} by tuning the
615 %%   options for includegraphics. This is the same .pdf plot as
616 %%   in Figure~\ref{fig_1} resized with to 90\% of the width of the text.
617 %% }\zlabel{lp_filt}
618 %% \end{figure*}
619
620
621
622
623
624 \subsection{Citations, masked citations, and references}
625
626 In LaTeX{} citations in the text are indicated by typing commands
627 like \mintinline{latex}{\cite{}} with the author, name, year,
628 parenthesis information for APA style are determined when the document
629 is typeset. Typing the citation commands amounts to
630 ``cite-while-you-write''. LaTeX automatically generates a bibliography
631 in the APA style from the corresponding .bib file (bibliography
632 database) according to the citations that appear in the text. There
633 lots of options for citation format, see the
634 \mintinline{latex}{biblatex} and \mintinline{latex}{apa6} docs for
635 reference. For instance, the \mintinline{latex}{\parencite} command
636 generates a formatted citation in parentheses
637 \parencite[Lampert1986]. The cite command generates one without
638 parentheses, as in~\cite[Lampert1986]. When manuscript submission
639 requires citation masking for blind review, the masked variants of the
640 citation commands, e.g., \mintinline{latex}{\maskparencite} can be
641 used: \maskparencite[Lampert1986]. The masked citations are indicated
642 in bold when the manuscript is typeset normally and replaced with {\it
643   (1 citation removed for masked review)} when typeset with the mask
644   option.
645
646 The .bib file is a text file with bibliography entries that have the
647 usual author, title, data, publisher, fields, and a great many others,
648 in a specific format. There are several options for where to get the
649 .bib file. Scientific literature search engines, publisher websites
650 routinely export citations in .bib format which can be copy-pasted
651 instead of tediously typed. If a reference manager is already being
652 used, it may also be able to export its references to .bib format. And
653 there are a number of reference managers that are designed from the
654 ground up to use .bib. As of this writing, the open-source JabRef
655 seems to have emerged as pick of the litter, being fully featured
656 enough to support general use and working across platforms. BibDesk

```


712 resubmission of a document following revision requires both ``clean''
 713 version with the changes made and a draft version marked up to
 714 indicate where the revisions were made. For cases where there are two
 715 versions of a .tex document and the changes are not explicitly marked
 716 up inline, the command line utility program
 717 `\mintinline{bash}{latexdiff}` can be used to automatically generate a
 718 single pdf with the differences between the versions indicated as in
 719 changes. Both of these features are best suited to marking revisions
 720 and changes in the text of relative similar documents and are not
 721 well-suited to track massive restructuring or revisions to figures and
 722 tables. Here is the list of changes explicitly marked up in the
 723 previous paragraph.
 724
 725 `\listofchanges`
 726
 727
 728 `\subsection{Compositing documents: files and file formats}`
 729
 730 Various files and formats are required go submit and publish a
 731 research report. These may include a main editable manuscript
 732 (document), supporting information (document, data), figures (vector
 733 and raster image graphics files), tables, and bibliographic
 734 info. Journals and publishers have divergent interests (readability
 735 for evaluation in review vs. production for print and digital
 736 formats) and (thus) different requirements for document
 737 preparation. This is further complicated by open-access policies that
 738 require authors to deposit a final pre-publication manuscript if the
 739 publisher won't (but most do, eventually). For submission to
 740 Psychological Science for instance, the file formats are `\LaTeX` (.tex)
 741 for editable text and Portable Document Format (.pdf) for graphics, a
 742 vector format that is scalable without loss of resolution. To submit
 743 the report to the journal for review the .tex and .pdf graphic files
 744 composited into a single .pdf file and all files
 745 uploaded~`\cite{PsychSciSubmissions2020, PsychSciFigs2013}`. Whereas the
 746 journal submission portal requires the a single composited document
 747 with text and graphics all in one, the publisher's portal requires the
 748 separate editable text and graphics files, i.e., the .tex and graphics
 749 .pdfs.
 750
 751 Working with `\LaTeX` simplifies some aspects of this by allowing
 752 files in different digital formats to be included in documents in
 753 various ways. As illustrated by linking results and arbitrary text for
 754 narrative descriptions and tables, separate files of `\LaTeX` can be
 755 inserted directly into the document as if typed in place. This allows
 756 the tables to be reproducibly prepared as separate files (as required
 757 by some publishers) and also incorporated in exactly the same form in
 758 the body of the manuscript (as also required by these publishers). The
 759 same holds for the camera ready graphics for Figure~`\zref{ms:multipanel}` and
 760 `\zref{ms:tikzfig}` which are also separate files included as-is in the
 761 manuscript. Additionally the `\mintinline{latex}{\includepdf}` package,
 762 allows all or selected pages of a multi-page PDF documents to be
 763 included in a `\LaTeX` as demonstrated in by the Supplementary
 764 Information that includes the entire PDF of the fully executed data
 765 analysis Jupyter Notebook. Finally, the `\mintinline{latex}{\minted}`
 766 package used extensively throughout this document will import the

```
767 contents of separate files into the \LaTeX{} document and also
768 highlight the code according to the syntax of the specific language,
769 e.g., Python, R, \LaTeX{} which is of great value in documenting
770 scripted reproducible research pipelines. The Supplemental Information
771 demonstrates this by importing and highlighting all the \LaTeX{} files
772 used in the production and reproduction of this tutorial report.
773
774 \subsection{Author manuscripts}
775
776 Whereas journals may require submission as a double spaced manuscript,
777 the published articles typeset single space in two columns with
778 figures and tables where they belong are generally easier to read.
779 Switching the \mintinline{latex}{documentclass} option from man
780 (manuscript) to jou (journal) typesets the document in a
781 more-nearly-journal-like format (Figure~\zref{ms:apa67_jou}), which
782 may be useful for distributing working drafts or post-publication
783 author manuscripts during a publisher's embargo period.
784
785 \begin{figure}
786 \caption{Example of typesetting this document with the jou option}
787 \zlabel{ms:apa67_jou}
788 \centering
789 \includegraphics[width=.65\textwidth]{images/apa67_jou.png}
790 \end{figure}
791
792
793 \section{Conclusion}
794
795 There are many ways to prepare a research report but far fewer to do
796 so reproducibly while at the same time satisfying the requirements of
797 publication styles and online journal submission and production
798 platforms. This report illustrates one approach that does so and
799 dovetails with best practices in open science data analysis. Once
800 a reproducible analysis is in place, the additional cost of the
801 reproducible report is acquiring a working knowledge of \LaTeX{} and
802 if necessary Ti{\it}Z.
803
804 \newpage
805 \printbibliography
806
807 \end{document}
```

Source: author_si.tex

This is the LaTeX for this Supporting Information, i.e., it is typesetting itself.

```

1  \documentclass[helv,letter,doc,natbib,11pt]{apa6} %% man <-> jou <-> doc
2  \usepackage[american]{babel}
3  \usepackage[utf8x]{inputenc}
4
5  \renewcommand{\familydefault}{\sfdefault}
6
7  \usepackage{amsmath}
8  \usepackage{graphicx}
9  \usepackage[colorinlistoftodos]{todonotes}
10 \usepackage{xcolor}
11
12
13 % use this for URLs
14 \usepackage{hyperref}
15 \hypersetup{
16     colorlinks=true,
17     citecolor=blue,
18     linkcolor=blue,
19     filecolor=blue,
20     urlcolor=blue,
21 }
22
23 % for cross references back to the main doc
24 % use \zref{} and \zlabel{} instead of latex native \ref{} and \label{}
25 \usepackage[xr, user, titleref]{zref}
26 \externaldocument{apa_ms} % other .tex file to cross reference
27
28 % use this to include text files verbatim (not shown)
29 \usepackage{verbatim}
30
31 %use this package for highlighted source, e.g., research_report.tex
32 \usepackage{minted}
33 \setminted[latex]{
34     frame=lines,
35     bgcolor=bgc,
36     fontsize=\footnotesize,
37     linenos
38 }
39
40 % use this to include multipage pdf docs, e.g., conveted jupyter notebook, other docs
41 \usepackage{pdfpages}
42
43
44 \title{Supporting Information: Open science with style: A reproducible reproducible research report}
45 \shorttitle{Supporting Information: Reproducible reports with LaTeX{}}
46 \author{Thomas P. Urbach}
47 \affiliation{Kutas Lab \\\ Cognitive Science Department \\\ University of California, San Diego}
48
49 \begin{document}
50 \maketitle
51

```

```

52  \tableofcontents
53
54  \section{Summary}
55
56  Additional information can go here and be formatted to APA 6th
57  guidelines or something else. The supporting information and main
58  document can use the same research\_report.bib file so the references
59  match. With {\tt \textbackslash usepackage[xr,user,titleref]\{zref\}},
60  you can cross-reference back and forth between documents \ldots the
61  main report and this SI. For instance here is a reference to
62  Figure~\zref{ms:multipanel} in the main document. The reference is via the
63  label, i.e., {\tt \textbackslash zlabel\{ms:multipanel\}} so if the figure
64  is moved to a different page or its number changes because of
65  additions or deletions, this reference by number will update
66  automatically. The following sections show the source files that
67  generated the plots, figures, manuscript and supporting information pdfs.
68
69
70  APA 7th docs \url{http://ctan.math.washington.edu/tex-archive/macros/latex/contrib/apa7/apa7.pdf}
71  APA 6th docs \url{http://ctan.math.utah.edu/ctan/tex-archive/macros/latex/contrib/biblatex-contrib/biblatex-apa}
72
73
74
75
76  % For APA 7th TeXLive 2020 use change the figure captions from
77  %
78  %   \caption{First sentence. Rest of the caption.}
79  %
80  % to
81  %
82  %   \caption{First sentence.} \figurenote{Rest of caption.}
83  %
84  % and this preamble
85  %
86  % \documentclass[man,biblatex,10pt]{apa7}
87  % \usepackage{csquotes}
88  % \DeclareLanguageMapping{american}{american-apa}
89  % \usepackage[backend=biber,style=apa]{biblatex}
90
91
92
93  \section{System setup}
94
95  \subsection{Installing conda environments}
96
97  If you already use conda environments in a recent linux operating
98  system, you can install a minimal conda environment to run the
99  notebooks like so and follow the prompt (or omit -y to the end of the
100  command to install the packages without prompting).
101
102  \begin{minted}{bash}
103  conda create -n apa67_report pandas pyarrow matplotlib jupyter firefox -y
104  $ activate apa67_report
105  $ jupyter notebooks
106  \end{minted}

```

```

107
108 If you are not yet set up to use conda environments, you can follow
109 the instructions to download and install a minimal conda installer,
110 miniconda3
111 (\href{https://docs.conda.io/en/latest/miniconda.html}). This provides
112 just enough infrastructure to create a conda environemnt and install
113 packages as shown in the example above. If you want to create conda
114 environments and install packages faster, then install the `mamba`
115 conda package (\href{https://mamba.readthedocs.io/en/latest/}).
116
117 If you are not yet set up to use conda environments and don't want to
118 be then you are on your own. You can run pipeline\_1.ipynb if you have
119 numpy, pandas, matplotlib and jupyter. You need the spudtr package to
120 run pipeline\_2.ipynb. Older versions are available via pip install,
121 but there is no assurance it is compatible with the versions of
122 packages you already have installed.
123
124 \subsection{Installing \LaTeX}
125
126 \subsubsection{Linux Installation via network}
127
128 You do not need to be root or admin to install TeX Live over the
129 networks and best practices are to install your copy in your
130 directory. That way you control the version and packages you
131 use. First read through the quick installation instructions
132 \href{https://www.tug.org/texlive/quickinstall.html}{here}. Then,
133 (summarizing from
134 \url{https://www.tug.org/texlive/acquire-netinstall.html}):
135
136 \begin{enumerate}
137
138 \item Download \url{install-tl-unx.tar.gz} to some scratch/working
139 directory, unpack the archive, change to the new directory it
140 made, i.e., \mbox{install\textendash tl\textendash YEARMONTHDAY} for
141 whatever version, and run the installer.
142
143 \begin{minted}{bash}
144 $ tar -xf install-tl-unx.tar.gz
145 $ cd install-tl-20200814
146 $ perl install-tl
147 \end{minted}
148
149 Follow the prompts, make sure you are happy with and have write
150 permissions in the default installation directory, and press ``i''
151 to install.
152
153 \item Update your ~/.bashrc file with the path to the new TeX Live
154 installation.
155
156 \begin{minted}{bash}
157 PATH=/home/turbach/texlive/2020/bin/x86_64-linux:$PATH
158 INFOPATH=/home/turbach/2020/texmf-dist/doc/info:$INFOPATH
159 MANPATH=/home/turbach/2020/texmf-dist/doc/man:$MANPATH
160 \end{minted}
161

```

```

162 \end{enumerate}
163
164 That's it, you have a complete functioning installation of \LaTeX{}
165 with the latest packages, TeX Live 2020 as of this writing.
166
167 The installation probably has everything you need including the apa6
168 and apa7 styles used for this report.
169
170 If there is a new package or update you want and you want to manage
171 the TeX packages with the TeX Live GUI you also need to install
172 perl/tk. There is a conda package for this, you can install into any
173 compatible conda env.
174
175 \begin{minted}{bash}
176 $ conda activate some_general_purpose_env
177 $ conda install perl-tk -c BioBuilds -y
178 \end{minted}
179
180
181 \subsubsection{OSX Installation}
182
183 See instructions for MacTeX here: \url{https://www.tug.org/mactex/}
184
185 \subsubsection{Windows}
186
187
188 See Quick Install instructions
189 \href{https://www.tug.org/texlive/quickinstall.html}{here}
190
191 and Windows installer instructions
192 \href{https://www.tug.org/texlive/acquire-netinstall.html}{here}.
193
194
195 % The next two sections show the (converted-to-pdf)
196 % jupyter notebook for generating the figures, lateLaTeX{} .tex file for the main report and the jupyter
197 % notebook that generates the pdf plots for the filter figures.
198
199 % -----
200 % Jupyter notebook source
201 \newpage
202 \normalsize
203 \section{Source: author\_analysis.ipynb}\zlabel{si:analysis_nb}
204
205 The pdf of the notebook is generated by {\tt jupyter convert ... --to pdf}. The
206 LaTeX{} package {\tt pdfpages} is used to slurp it into the SI pdf.
207
208 \includepdf[pages={1-}]{apa_analysis}
209
210
211 % -----
212 % research report LaTeX
213 \newpage
214 \section{Source: {\tt research\_report.tex}}\zlabel{apa_ms_tex}
215 This is the LaTeX{} for the main report.
216

```

```

217 \definecolor{bgc}{rgb}{1.0,.96,1.0}
218 \inputminted{latex}{apa_ms.tex}
219
220
221 % -----
222 % supporting information LaTeX
223 \newpage
224 \section{Source: {\tt author\_si.tex}}\zlabel{apa_si_tex}
225
226 This is the LaTeX{} for this Supporting Information, i.e., it is
227 typesetting itself.
228
229 \inputminted{latex}{apa_si.tex}
230
231
232 % -----
233 % Figure 1 LaTeX
234 \newpage
235 \section{Source: {\tt fig1.tex}}\zlabel{si:fig1_src}
236 This is basic LaTeX{} template for a free-standing .tex file that pdflatex can turn
237 into a .pdf graphic for import or upload. It is just the graphic, no caption or numbering.
238
239 \inputminted{latex}{apa_fig1.tex}
240
241
242 % -----
243 % Figure 2 LaTeX
244 \newpage
245 \section{Source: {\tt fig3.tex}}\zlabel{si:fig3_src}
246
247 This is the LaTeX{} for the multipanel TikZ figure with fancy layout
248 and annotation stuff. Again, just for the pdf graphic, no caption.
249
250 \inputminted{latex}{apa_fig2.tex}
251
252 % -----
253 % Makefile
254 \newpage
255 \section{Source: \mintinline{makefile}{Makefile}}\zlabel{si:makefile_src}
256 This is the Makefile used to build/rebuild the ms, si, figs individually
257 and all the documents in one fell-swoop.
258
259 \inputminted{makefile}{Makefile}
260
261
262 % -----
263 % bib
264 \newpage
265 \section{Source: {\tt research\_report.bib}}\zlabel{si:bib_src}
266 This is the .bib for citations and references, shared by the ms and this SI.
267
268 \inputminted{bibtex}{apa_ms.bib}
269
270 % Supporting Information References (if any)
271 \bibliography{research_report}

```


272

273

```
\end{document}
```

Source: fig1.tex

This is basic LaTeX template for a free-standing .tex file that pdflatex can turn into a .pdf graphic for import or upload. It is just the graphic, no caption or numbering.

```
1  %% use this to make a free-standing pdf graphc instead of a paginated latex doc
2
3  % bare bones 2-panel figure, no annotations
4  \documentclass[border=0in]{standalone}
5  \usepackage{graphicx}
6  \begin{document}
7  \includegraphics[width=.45\textwidth]{generated/p3_midline_plot1.pdf}
8  \includegraphics[width=.45\textwidth]{generated/p3_midline_plot2.pdf}
9  \end{document}
```

Source: fig3.tex

This is the LaTeX for the multipanel TikZ figure with fancy layout and annotation stuff. Again, just for the pdf graphic, no caption.

```

1  %% use this to make a free-standing pdf graphic instead of a paginated latex doc
2  \documentclass[border=0in]{standalone}
3
4  % dejavu san serif matches matplotlib default
5  \usepackage{dejavu}
6  \renewcommand*\familydefault{\sfdefault} % set base font to sans serif
7  \usepackage[T1]{fontenc}
8  \usepackage{amsmath} % math symbols
9  %% \usepackage{pbox}
10 \usepackage{tikz}
11
12 \usetikzlibrary{arrows,shapes,backgrounds,shadows,fit,positioning,scopes, calc}
13
14 %% whitesmoke background
15 \definecolor{whitesmoke}{rgb}{.9607843137, .9607843137, .9607843137}
16
17 %% style general layout
18 \tikzstyle{background rectangle} = [fill=whitesmoke]
19 \tikzstyle{background rectangle} = [fill=white]
20 \tikzstyle{every node} = [outer sep=0pt, inner sep=3pt]
21
22 %% define the plot label spec: #1=tag, #2=location, #3=text
23 \def\plabel[#1]#2#3{
24   \node [left, scale=1.0] (#1) at (#2.north west) {#3};
25 }
26
27 \begin{document}
28 \begin{tikzpicture}[
29   >=stealth, %% shape of the annotation arrows
30   show background rectangle,
31   %% inner frame sep=2mm % sep = bleed or 0 for tight background
32 ]
33
34   % -----
35   % Panel a figure as generated
36   \coordinate (axy) at (0, 0);
37   \plabel[label-a]{axy}{a};
38   \node [
39     anchor=north west,
40     rectangle,
41     fill=whitesmoke
42   ] (p3-head-pdf) at (label-a.north east){
43     \includegraphics[height=1in]{generated/p3_head_plot3.pdf}
44   };
45
46   \node[
47     xshift=.75in,
48     rectangle,
49     fill=whitesmoke
50   ] (montage) at (p3-head-pdf.east){

```

```

51     \includegraphics[height=1in]{images/TopHead.pdf}
52 };
53
54
55
56 % -----
57 % Panel b TikZ layout and annotations
58
59 % crop top and bottom of generated pdf
60 \newcommand{\tbtrim}{0.4in}
61 \newcommand{\mathfontscale}{2}
62
63 \coordinate [yshift=-0.5in] (bxy) at (p3-head-pdf.south west);
64 \plabel[label-b]{bxy}{b};
65
66 % P300 effect in a shadow box
67
68 % frame + drop shadow
69 \node (b-effect-box) [
70     anchor=north west,
71     draw=black!40,
72     fill=white,
73     rounded corners=4pt,
74     drop shadow,
75     minimum height=1.5in,
76     minimum width=1.9in
77 ]
78 at (label-b.south east) {};
79
80 % electrode scatter + colorbar
81 \node (p3-effect) at (b-effect-box) {
82     \includegraphics[
83         clip,
84         trim={8.1in, \tbtrim, 1.35in, \tbtrim},
85         height=1.125in
86     ]{generated/p3_head_plot3.pdf}
87     \includegraphics[
88         clip,
89         trim={11.75in, 0, 0, 0},
90         height=1.125in
91     ]{generated/p3_head_plot3.pdf}
92 };
93
94
95 % montage head
96 \node[
97     xshift=.2in,
98     yshift=-.2in,
99     opacity=.25
100 ] (montage) at (b-effect-box.north west){
101     \includegraphics[height=.25in]{images/TopHead.pdf}
102 };
103
104 % Title
105 \node [

```

```

106     anchor=south,
107     scale=.66
108 ] (effect-label) at (b-effect-box.north) {
109     P300 ERP effect (Target  $\$-\$$  Standard)
110 };
111
112 % annotation text
113 \node [
114     anchor=north west,
115     xshift=0.025in,
116     yshift=0.05in,
117     scale=.5
118 ] (post-pointer) at (p3-effect.south) {
119     posterior maximum
120 };
121
122 % annotation arrow
123 \coordinate [xshift=-0.03in, yshift=-0.425in] (RDPa) at (p3-effect);
124 \draw [->] (post-pointer.north west) -- (RDPa);
125
126
127 % equals (=)
128 \node [scale=\mathfontscale, anchor=west] (text-equals) at (b-effect-box.east){ $\$=\$$ };
129
130 % P300 target
131 \node [anchor=west] (b-target) at (text-equals.east){
132     \includegraphics[
133         clip,
134         trim={4.25in, \tbtrim, 5.25in, \tbtrim},
135         height=1in
136     ]{generated/p3_head_plot3.pdf}
137
138 };
139 \node [anchor=south, scale=.66] (target-label) at (b-target.north) {
140     Target
141 };
142
143
144
145 % minus (-)
146 \node [scale=\mathfontscale, anchor=west] (text-minus) at (b-target.east){ $\$-\$$ };
147
148 % P300 standard
149 \node [anchor=west] (b-standard) at (text-minus.east){
150     \includegraphics[
151         clip,
152         trim={0.5in, \tbtrim, 9.0in, \tbtrim},
153         height=1in
154     ]{generated/p3_head_plot3.pdf}
155 };
156 \node [anchor=south, scale=.66] (standard-label) at (b-standard.north) {
157     Standard
158 };
159
160 \end{tikzpicture}

```

161 `\end{document}`

Source: `Makefile`

This is the Makefile used to build/rebuild the ms, si, figs individually and all the documents in one fell-swoop.

```
# TODO: for reproducibility check we are running in the right conda environment

# where to find the files
HOME_DIR = /home/turbach/TPU_Projects/demos/latex/apa_6th_example

# jupyter notebook figure generator ... slurp the actual research data
# and generate the pdf plots that will appear in the ms and si Figures
JUPYTER_CONVERT = jupyter nbconvert --ExecutePreprocessor.timeout=None --execute

export_env:
    conda list --explicit > environment.txt

# the minted syntax highlighting package insists on -shell-escape
ms:
    pdflatex -shell-escape apa_ms
    biber apa_ms
    pdflatex -shell-escape apa_ms
    pdflatex -shell-escape apa_ms

si:
    pdflatex -shell-escape apa_si
    biber apa_ms
    pdflatex -shell-escape apa_si
    pdflatex -shell-escape apa_si

bib:
    pdflatex -shell-escape apa_ms
    pdflatex -shell-escape apa_si
    biber apa_ms

# for long-running jobs use --ExecutePreprocessor.timeout=None
analysis: export_env
    jupyter nbconvert --execute --to pdf ./apa_analysis.ipynb

fig1:
    pdflatex apa_fig1.tex

fig2:
    pdflatex apa_fig2.tex
```

```
figs: analysis fig1 fig2
```

```
# remove intermediate latex files aux, log and stash backup files
```

```
# move
```

```
clean_aux:
```

```
    latexmk -c
```

```
# multiple passes to get the zref cross-document cross references right
```

```
all: figs bib si ms si ms si
```

```
# build everything then wipe the intermediate stuff
```

```
for_upload: all clean_aux
```


Source: research_report.bib

This is the .bib for citations and references, shared by the ms and this SI.

```
@book{APAStyle6th,
  author =      {{American Psychological Association}},
  title =      {Publication Manual of the American Psychological
                Association},
  edition =      {6th},
  publisher =    {American Psychological Association},
  pages =      272,
  year =      2010,
  type =      {Book}
}

@misc{APSStructStyle,
  title =      {Manuscript Structure, Style, and Content Guidelines},
  publisher =    {Association for Psychological Science},
  url =      {https://www.psychologicalscience.org/publications/ms-structure-guideline},
  urldate =     {2020-08-11},
  type =      {Web Page}
}

@ARTICLE{Lamport1986,
  author =      {L[eslie] A. Lamport},
  title =      {The Gnats and Gnus Document Preparation System},
  journal =      {G-Animal's Journal},
  year =      1986,
  volume =      41,
  number =      7,
  pages =      "73+",
  month =      jul,
}

@misc{PsychSciFigs2013,
  title =      {{APS} Figure Format and Style Guidelines},
  publisher =    {Association for Psychological Science},
  month =      10,
  url =      {https://www.psychologicalscience.org/publications/aps-figure-format-style},
  urldate =     {2020-08-11},
  year =      2013,
  type =      {Web Page}
}
```

```

@article{PsychSciSubmissions2020,
  author =      {},
  title =       {Psychological Science 2020 Submission Guidelines},
  publisher =    {Association for Psychological Science},
  urldate =      {2020-08-11},
  url-modified = {2020-07-13},
  url =
                {https://www.psychologicalscience.org/publications/psychological_science/}

  year =        2020,
  type =         {Web Page}
}

@misc{Urbach2020z,
  author =       {Urbach, T.~P.},
  title =        {eeg-workshops/mkpy\_data\_examples/data [data set]},
  DOI =          {10.5281/zenodo.4099632},
  year =         2020,
  month =        11,
}

@misc{ccby4.0,
  title =        {Creative Commons
                  Attribution-NonCommercial-ShareAlike 4.0
                  International (CC BY-NC-SA 4.0) [software license]},
  url =          {https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode},
}

@incollection{kluEtAl2016,
  title =        {Jupyter Notebooks-a publishing format for
                  reproducible computational workflows.},
  author =       {Kluyver, T. and Ragan-Kelley, B. and P{\`e}rez,
                  F. and Granger, B.~E. and Bussonnier, M. and
                  Frederic, J. and Kelley, K. and Hamrick, J.~B. and
                  Grout, J. and Corlay, S. and others},
  booktitle =    {Positioning and Power in Academic Publishing:
                  Players, Agents and Agendas},
  editor =       {Loizides, F. and Schmidt, B.},
  volume =       2016,
  year =         2016,
  doi =
                {https://doi.org/10.3233/10.3233/978-1-61499-649-1-87}
}

@misc{latexproject,
  title =        {\LaTeX{} --- A document preparation system
}

```

```
author = [software],  
url = {{\LaTeX{} developers}},  
}
```