

Palmer Penguins Part 1: 3-Minute Video Production Guide

Building a concise “EDA and Simple Regression” video using open-source tools

TABLE OF CONTENTS

1. Available Analysis Images
 2. Video Structure
 3. Complete Script
 4. Slide Design Specs
 5. Production Workflow
 - Method A: Audio-Only Narration
 - Method B: Picture-in-Picture with OBS
 - Method C: AI Avatar for Intro/Outro
 6. Step-by-Step Instructions
 - Creating Slides with LibreOffice Impress
 - Recording Audio with Audacity
 - Recording with OBS (Picture-in-Picture)
 - Editing with Kdenlive
 - Adding Polish Effects
 - Generating Captions with Whisper
 - Creating Thumbnail
 - YouTube Upload
 7. File Organization
 8. Final Checklist
-

AVAILABLE ANALYSIS IMAGES

The following PNG files from the R analysis are in `posts/palmer_penguins_part1/`:

Filename	Description	Slide
penguin-hero-part1.png	Hero/title image	1, 8
eda-overview.png	Combined species + relationship	2
correlation-matrix.png	Correlation heatmap	3
simple-regression-model.png	Regression with confidence band	4, 5
model-diagnostics.png	Residual diagnostic plot	6
species-comparison.png	Body mass boxplot by species	7

Color Palette (High Contrast):

Adelie: #FF6B6B (coral)
Chinstrap: #9B59B6 (purple)
Gentoo: #2E86AB (deep ocean blue)
Text: #2C3E50 (dark gray)
Background: #FFFFFF (white)
Accent: #27AE60 (green)

VIDEO STRUCTURE

Duration: 3 minutes **Slides:** 8 **Word count:** ~350 words

Section	Duration	Content
Hook & Data Overview	30 sec	Intro + 333 penguins, 3 species
Visual Exploration	45 sec	Scatter plot + correlation $r = 0.87$
Simple Regression	1 min	Model equation, $R^2 = 0.762$
Limitations & Next	45 sec	Species clustering, Part 2 preview

COMPLETE SCRIPT

[SLIDE 1: Title]

Duration: 10 seconds **IMAGE:** penguin-hero-part1.png

“Palmer Penguins Part 1: Exploratory Data Analysis and Simple Regression. Can we predict a penguin’s body mass just from its flipper length? Let’s find out.”

[SLIDE 2: The Data]

Duration: 20 seconds **IMAGE:** eda-overview.png

“We have 333 penguins from three species: Adelie, Chinstrap, and Gentoo. The scatter plot on the right already hints at something interesting—longer flippers seem to go with heavier bodies. Gentoo penguins, in blue, are clearly the largest. But how strong is this relationship?”

[SLIDE 3: Correlation]

Duration: 20 seconds **IMAGE:** correlation-matrix.png

“The correlation between flipper length and body mass is 0.87—that’s strong. But correlation only tells us variables are related. To make predictions, we need regression. Let’s fit a line through these points and see how well it works.”

[SLIDE 4: The Model]

Duration: 25 seconds **IMAGE:** simple-regression-model.png

“Here’s our simple linear regression. The equation: Body Mass equals negative 5,781 plus 49.7 times Flipper Length. That slope means every additional millimeter of flipper adds about 50 grams of body mass. The R-squared is 0.762—flipper length alone explains 76 percent of the variation in body mass. That’s surprisingly good for one predictor.”

[SLIDE 5: Making Predictions]

Duration: 20 seconds **IMAGE:** simple-regression-model.png (with callout overlay)

“What does this mean in practice? A penguin with 200-millimeter flippers? We predict about 4,100 grams. The gray band shows our uncertainty. Field researchers could use this to estimate body mass quickly—just measure the flipper.”

[SLIDE 6: The Problem]

Duration: 25 seconds IMAGE: model-diagnostics.png

“But there’s a problem. Look at the residuals—the prediction errors—colored by species. See how they cluster? Gentoo residuals sit together, separate from Adelie and Chinstrap. Our model treats all penguins the same, but species clearly matters. We’re missing something important.”

[SLIDE 7: Key Takeaways]

Duration: 15 seconds IMAGE: species-comparison.png

“What did we learn? Flipper length predicts body mass well—76 percent of variance. But species differences create patterns our simple model misses. The residuals tell us we can do better.”

[SLIDE 8: Next Up]

Duration: 15 seconds IMAGE: penguin-hero-part1.png

“In Part 2, we add species to the model. Spoiler: R-squared jumps from 0.76 to over 0.86. That’s the power of thinking carefully about your data. See you next time.”

Total: ~350 words 2:50 at natural pace

8-SLIDE DECK SUMMARY

Slide	Image	Key Content
1. Title	penguin-hero-part1.png	“Can flipper length predict mass?”
2. The Data	eda-overview.png	333 penguins, 3 species
3. Correlation	correlation-matrix.png	$r = 0.87$, strong relationship
4. The Model	simple-regression-model.png	Equation + $R^2 = 0.762$
5. Predictions	simple-regression-model.png	200mm \rightarrow 4,100g example
6. The Problem	model-diagnostics.png	Species clustering in residuals
7. Takeaways	species-comparison.png	76% explained, species matters
8. Next Up	penguin-hero-part1.png	Part 2 preview: $R^2 \rightarrow 0.86$

SLIDE DESIGN SPECS

Dimensions: 1920 \times 1080 (16:9)

Typography:

- Headings: Inter or Source Sans Pro, 56pt, bold

- Body: 32pt regular
- Minimal text—let images dominate

Layout:

- Image fills 2/3 of slide
- One key message per slide
- 40%+ whitespace

PRODUCTION WORKFLOW

Required Software

Tool	Version	Purpose	Install (macOS)
LibreOffice Impress	24.x	Create slides	<code>brew install --cask libreoffice</code>
Audacity	3.7+	Record audio narration	<code>brew install --cask audacity</code>
OBS Studio	32.x	Record video with PiP	<code>brew install --cask obs</code>
Kdenlive	25.x	Edit video	<code>brew install --cask kdenlive</code>
Whisper	latest	Auto-captions	<code>pip install openai-whisper</code>
ImageMagick	7.x	Create thumbnail	<code>brew install imagemagick</code>

Documentation Links: - Audacity Manual - OBS Knowledge Base - Kdenlive Manual

Method A: Audio-Only Narration (Slides + Voiceover)

Best for: Quick production, no webcam needed

Timeline:

Video: [slide-1] [slide-2] [slide-3] [slide-4] [slide-5] [slide-6] [slide-7] [slide-8]

Audio: [narr1] [narr2] [narr3] [narr4] [narr5] [narr6] [narr7] [narr8]

Workflow:

1. Create slides in LibreOffice Impress
2. Export slides as PNG
3. Record audio in Audacity (one file per slide)
4. Combine in Kdenlive
5. Export MP4

Method B: Picture-in-Picture (Narrator + Slides with OBS)

Best for: More engaging, personal connection with viewers

Layout:

```

+-----+
|               |
|    SLIDE CONTENT    |
|               |
|               |
|               +-----+
|               | WEBCAM |
|               | (you)  |
|               +-----+
|               |

```

+-----+

Workflow:

1. Create slides in LibreOffice Impress
2. Set up OBS with slide window + webcam overlay
3. Record entire presentation in one take (or per slide)
4. Minor edits in Kdenlive if needed
5. Export MP4

Method C: AI Avatar for Intro/Outro (Hybrid Approach)

Best for: Professional appearance without webcam, combines personal touch with clean voiceover-only slides

Concept: Use AI-generated talking avatar for intro (Slide 1) and outro (Slide 8), with pure voiceover for statistical content (Slides 2-7). Background music plays only during avatar segments.

Timeline Structure:

+-----+			+-----+		
INTRO	SLIDES 2-7 (VOICEOVER)			OUTRO	
Avatar	No music - pure narration focus			Avatar	
+Music				+Music	
10 sec	~2:30			15 sec	
+-----+			+-----+		

Why This Works:

- **Cognitive load theory:** Music competes with learning during statistical explanations (StatQuest, 3Blue1Brown use no music)
- **Personal connection:** Avatar intro/outro adds human element
- **Professional polish:** Music bookends signal “beginning” and “end”

AI Avatar Platforms:

Platform	Pricing	Best For
HeyGen	Free tier + paid	High quality, many avatar styles
Synthesia	Subscription	Enterprise, multilingual
Vidnoz	Free tier	Quick tests, budget-friendly
D-ID	Free tier + paid	Creative Studio, custom avatars

Workflow:

1. **Create avatar clips:**
 - Upload intro script (Slide 1 narration, ~25 words)
 - Upload outro script (Slide 8 narration, ~35 words)
 - Generate avatar videos (MP4)
 - Download both clips
2. **Record slides 2-7 narration:**
 - Use Audacity for voiceover only
 - No music during statistical content
3. **Source background music:**
 - Pixabay Music, FreePD, or YouTube Audio Library
 - Choose: calm, minimal, instrumental
 - ~15-30 seconds needed for intro/outro only
4. **Assemble in Kdenlive:**

- V1: Avatar intro → Slide images 2-7 → Avatar outro
- A1: Avatar audio (built-in) → Voiceover → Avatar audio
- A2: Music (intro only, -15dB) → silence → Music (outro, -15dB)

Music Guidelines (Intro/Outro Only):

- **Volume:** -15 dB to -20 dB (well below narration)
- **Duration:** Match avatar clip length (10-15 sec each)
- **Style:** Upbeat but not distracting, no lyrics
- **Fade:** 1-second fade out at end of intro, 1-second fade in at outro
- **Sources:**
 - Pixabay Music (free, no attribution)
 - FreePD (public domain)
 - YouTube Audio Library

STEP-BY-STEP INSTRUCTIONS

1. CREATING SLIDES WITH LIBREOFFICE IMPRESS

Automated Method (Python Script) A Python script `create_slides.py` is available to generate slides automatically:

```
cd /path/to/palmer_penguins_part1
python3 create_slides.py
```

This creates `video_production/slides/Palmer_Penguins_Part1_3min.odp` with all 8 slides.

Manual Method

1. Open LibreOffice Impress

```
open /Applications/LibreOffice.app
```

2. Create new presentation

- File → New → Presentation
- Select blank template

3. Set slide size

- Slide → Slide Size → 16:9

4. For each slide:

- Insert → Image → select the appropriate PNG
- Add text box for title (56pt, bold)
- Add text box for key message (36pt)
- Position image to fill ~2/3 of slide

5. Save the presentation

- File → Save As → `Palmer_Penguins_Part1_3min.odp`

Export Slides as PNG Via Command Line:

```
# Export to PDF first
/Applications/LibreOffice.app/Contents/MacOS/soffice \
--headless --convert-to pdf \
```

```
--outdir ./png_export Palmer_Penguins_Part1_3min.odp

# Convert PDF pages to PNG (1920x1080)
cd png_export
pdftoppm -png -r 300 \
  -scale-to-x 1920 -scale-to-y 1080 \
  Palmer_Penguins_Part1_3min.pdf slide
```

This creates slide-1.png through slide-8.png.

2. RECORDING AUDIO WITH AUDACITY

Initial Setup

1. Open Audacity

```
open -a Audacity
```

2. Set recording to Mono (via Audio Setup Toolbar - preferred method)

- Click **Audio Setup** button in the toolbar
- Select **Recording Channels** → **1 (Mono)**
- Or: Audacity → Settings (Cmd+,) → **Audio Settings** → Channels → 1

3. Select microphone

- Click **Audio Setup** button → **Recording Device**
- Select your microphone from the list

4. Test levels

- Click microphone icon to enable monitoring
- Speak normally—meter should peak between **-12 dB** and **-6 dB**

Recording Each Slide For each slide (1-8):

1. Press **R** to start recording
2. Read the script for that slide
3. Press **Space** to stop
4. **File** → **Export Audio** (Cmd+Shift+E)
5. Save as **slide1_narration.wav** (then slide2, slide3, etc.)
6. **Select All** (Cmd+A) → **Delete** to clear the track
7. Repeat for next slide

Post-Processing (Each File)

1. Normalize:

- Select All (Cmd+A)
- Effect → Volume and Compression → Normalize → -1.0 dB → Apply

2. Noise Reduction:

- Select 1-2 seconds of silence
- Effect → Noise Removal → Noise Reduction → Get Noise Profile
- Select All (Cmd+A)
- Effect → Noise Removal → Noise Reduction → 12 dB → Apply

3. Trim silence at start/end

4. Export as WAV

3. RECORDING WITH OBS (PICTURE-IN-PICTURE)

OBS is better than QuickTime when you need to record webcam + slides simultaneously.

Install OBS

```
brew install --cask obs
```

Initial Setup

1. Open OBS

```
open -a OBS
```

2. First-time setup wizard

- Select “Optimize for recording”
- Choose 1920x1080, 30fps
- Complete wizard

Configure Video Settings

1. OBS → Settings → Video

- Base Resolution: 1920x1080
- Output Resolution: 1920x1080
- FPS: 30

2. OBS → Settings → Output

- Output Mode: Advanced (for more control)
- Recording Format: **MKV** (recoverable if OBS crashes; remux to MP4 later)
- Encoder: NVENC (NVIDIA), AMF (AMD), or x264 (CPU)
- Recording Path: `video_production/`
Tip: After recording, use File → Remux Recordings to convert MKV to MP4.

Create Scene with PiP Layout

1. Create new scene

- Click + under Scenes
- Name it “Slides with PiP”

2. Add slide source (background)

- Click + under Sources
- Select **Window Capture**
- Name it “Slides”
- Choose your LibreOffice Impress window (or image viewer)
- Click OK
- Resize to fill the canvas

3. Add webcam source (overlay)

- Click + under Sources
- Select **Video Capture Device**
- Name it “Webcam”
- Select your camera
- Click OK

4. Position webcam as PiP

- Click on the webcam source in the preview
- Drag corners to resize to ~25% of frame

- Drag to bottom-right corner
 - Right-click → Transform → Edit Transform for precise positioning:
 - Size: 480 x 270 (or similar)
 - Position: 1400, 780 (bottom-right with margin)
5. **Optional: Add border to webcam**
- Right-click webcam source → Filters
 - Add **Rounded Rectangle** or use a colored border

Recording Workflow **Option A: Record All at Once**

1. Open your slides in LibreOffice Impress (Presentation mode: F5)
2. In OBS, verify the scene shows slides + your webcam
3. Click **Start Recording**
4. Present all 8 slides, advancing manually
5. Click **Stop Recording**
6. File saved to your recording path

Option B: Record Per Slide

1. Display slide 1
2. Click **Start Recording**
3. Read slide 1 script
4. Click **Stop Recording** → saves `slide1_with_pip.mp4`
5. Advance to slide 2, repeat

OBS Keyboard Shortcuts Set these in OBS → Settings → Hotkeys:

- Start Recording: `Cmd+Shift+R`
- Stop Recording: `Cmd+Shift+S`

This lets you control recording without switching windows.

Tips for Good PiP Recording

- **Lighting:** Face a window or use a ring light
 - **Eye contact:** Look at the camera (near your screen top), not the screen
 - **Background:** Clean, uncluttered background
 - **Audio:** Use external mic if possible (USB mic > laptop mic)
 - **Framing:** Head and shoulders visible, centered in webcam frame
-

4. EDITING WITH KDENLIVE

Create New Project

1. **Open Kdenlive**

```
open -a Kdenlive
```
2. **Project Settings dialog appears**
 - Project folder: Browse to `video_production/`
 - Video profile: Select **HD 1080p 30 fps**
 - Click **OK**
3. **Save project**
 - File → Save (`Cmd+S`)
 - Name: `Palmer_Penguins_Part1.kdenlive`

Import Files

1. **Right-click in Project Bin** (top-left panel)
2. Select **Add Clip or Folder**
3. Navigate to `video_production/slides/png_export/`
4. Select all 8 slide PNGs → Open
5. Repeat for audio files from `video_production/audio/`

You should have 16 items in Project Bin (8 images + 8 audio).

Build Timeline (Audio-Only Method) For each slide:

1. Drag `slide-1.png` onto **V1** track (video)
2. Drag `slide1_narration.wav` onto **A1** track (audio)
3. Extend the image duration to match audio:
 - Click on image clip
 - Drag right edge until it aligns with audio end
4. Place next slide/audio immediately after
5. Repeat for all 8 slides

V1: [slide-1] [slide-2] [slide-3] [slide-4] [slide-5] [slide-6] [slide-7] [slide-8]

A1: [audio-1] [audio-2] [audio-3] [audio-4] [audio-5] [audio-6] [audio-7] [audio-8]

Build Timeline (OBS PiP Method) If you recorded one continuous video:

1. Drag your OBS recording onto V1
2. Done! (May need minor trimming at start/end)

If you recorded per-slide:

1. Drag each clip onto V1 in sequence
2. Trim any gaps or mistakes

Add Transitions

1. Drag `slide-2` slightly left to overlap `slide-1` by ~0.3 seconds
2. Kdenlive auto-creates a dissolve transition
3. Repeat for remaining slides

Recommended transition duration: 0.3-0.5 seconds (subtle, not distracting)

Export Video

1. **Project → Render** (or Render button in toolbar)
2. Configure:
 - Output file: `video_production/pp.mp4`
 - Preset: **MP4 - H264/AAC**
 - Resolution: 1920x1080
3. Click **Render to File**
4. Wait for completion

5. ADDING POLISH EFFECTS

These techniques add professional quality without overwhelming the educational content.

Dissolve Transitions (0.3-0.5 seconds) Already covered in Kdenlive editing above. Key points:

- **Duration:** 0.3 seconds for quick cuts, 0.5 seconds for emphasis
- **Where:** Between all slides except intro/outro (if using avatar)
- **Why:** Smoother than hard cuts, easier on the eyes

Ken Burns Effect (Subtle Zoom/Pan on Images) Adds visual interest to static analysis images.

In Kdenlive:

1. Select an image clip on the timeline (e.g., `slide-4.png`)
2. Go to **Effects** tab → search “Transform”
3. Add **Transform** effect to clip
4. Set keyframes:
 - **Start:** Scale 100%, Position centered
 - **End:** Scale 105%, Position slightly shifted
5. Duration: Match slide duration (slow, subtle movement)

Recommended settings:

- Scale change: 100% → 105% (subtle zoom in)
- Position drift: 0-20 pixels over slide duration
- Apply to: Main analysis images (slides 2-6)
- Skip for: Title slide, conclusion slide

Alternative - Crop/Pan effect:

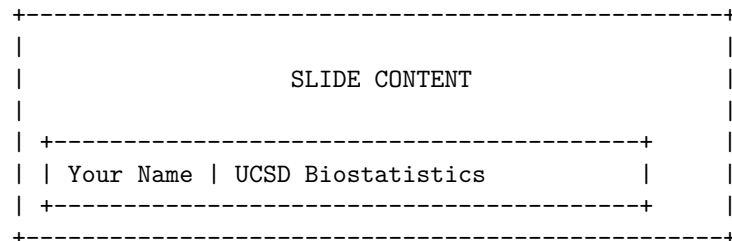
1. Effects → Transform → Position and Zoom
2. Keyframe start: full frame
3. Keyframe end: slight zoom to regression line or key area

Lower Third (Name Overlay on Slide 1) Professional touch showing presenter name.

In Kdenlive:

1. **Create text clip:**
 - Project → Add Title Clip
 - Background: Semi-transparent black (`#000000`, 60% opacity)
 - Text: “Your Name | UCSD Biostatistics” (or your affiliation)
 - Font: Inter or Source Sans Pro, 28pt, white
 - Position: Bottom left, 80px from edges
2. **Add to timeline:**
 - Place on V2 track above slide-1
 - Duration: 5-8 seconds (not full slide duration)
3. **Animate (optional):**
 - Add Slide effect: animate in from left over 0.3 seconds
 - Fade out at end

Lower Third Layout:



Fade In/Out (Video Start/End) Fade In (start of video):

1. Select first clip on timeline
2. Right-click → Add Effect → Fade → Fade In
3. Duration: 1 second

Fade Out (end of video):

1. Select last clip on timeline
2. Right-click → Add Effect → Fade → Fade Out
3. Duration: 1 second

Audio fades:

1. Select audio track
2. Drag the small triangles at start/end of audio clip
3. Create 0.5-second audio fades to prevent pops

Background Music (Intro/Outro Only) Per cognitive load research, music during statistical explanations hurts learning retention. Use music only for intro and outro.

Setup in Kdenlive:

1. Import music file to Project Bin
2. Drag to **A2** track (separate from narration on A1)
3. Position:
 - Under slide 1 (intro): 10 seconds
 - Under slide 8 (outro): 15 seconds
4. **Cut** the music clip to leave silence for slides 2-7

Volume adjustment:

1. Select music clip
2. Right-click → Adjust Audio → Volume
3. Set to **-15 dB to -18 dB** (much quieter than narration at -6 dB)

Fade music:

1. End of intro music: Add 1-second fade out
2. Start of outro music: Add 1-second fade in

Audio Timeline:

A1: [narr1] [narr2] [narr3] [narr4] [narr5] [narr6] [narr7] [narr8]	
A2: [music]	[music]
↑ fade out	fade in ↑
(end of intro)	(start of outro)

Recommended music sources:

- Pixabay Music - Free, no attribution
- FreePD - Public domain
- YouTube Audio Library

Music selection criteria:

- Instrumental only (no lyrics)
 - Calm, upbeat, not distracting
 - Match energy: professional but welcoming
 - Duration: At least 30 seconds (you'll use ~25 total)
-

6. GENERATING CAPTIONS WITH WHISPER

Generate Captions

```
cd video_production
whisper pp.mp4 --model base --output_format vtt --output_dir .
```

This creates pp.vtt.

Review and Correct Whisper may misrecognize technical terms. Common corrections needed:

Wrong	Correct
flipper links	flipper length
a delngen strap	Adelie, Chinstrap
Gen 2	Gentoo
R square	R-squared
scatter plant	scatter plot

Edit pp.vtt in any text editor to fix errors.

7. CREATING THUMBNAIL

Using ImageMagick (Command Line)

```
cd video_production

convert slides/png_export/slide-1.png -resize 1280x720! \
  -fill 'rgba(0,0,0,0.4)' -draw 'rectangle 0,500 1280,720' \
  -font Helvetica-Bold -pointsize 72 -fill white \
  -gravity south -annotate +0+100 'Part 1' \
  -font Helvetica -pointsize 36 -fill white \
  -gravity south -annotate +0+50 'Simple Regression in 3 min' \
  -fill '#27AE60' -gravity northeast -pointsize 48 \
  -annotate +40+40 'R2 = 0.76' \
  thumbnail.png
```

This creates a 1280x720 thumbnail with:

- Slide 1 as background
 - Dark overlay at bottom
 - “Part 1” in large white text
 - Subtitle below
 - Green R² badge in top-right
-

8. YOUTUBE UPLOAD

Upload via Browser

1. Go to youtube.com/upload
2. Drag pp.mp4 onto the page
3. Fill in details:

Title:

Palmer Penguins Part 1: Simple Regression in 3 Minutes

Description:

Can flipper length predict body mass? Let's find out with 333 penguins!

Timestamps:

0:00 - The Question
0:10 - Meet the Data (333 penguins, 3 species)
0:30 - Correlation: $r = 0.87$
0:50 - Simple Regression Model
1:15 - $R^2 = 0.762$ (76% variance explained)

Key Results:

- Flipper length explains 76% of body mass variance
- Every 1mm of flipper = 50g of body mass
- But species matters-residuals cluster by species!

Part 2: Adding species to the model → R^2 jumps to 0.86!

#DataScience #Statistics #Regression #Penguins #RStats

Tags/Keywords:

palmer penguins, simple regression, R programming, data science, statistics tutorial, linear regression, EDA, exploratory data analysis

4. **Upload thumbnail:** Click “Upload thumbnail” → select `thumbnail.png`
5. **Add captions:**
 - After upload completes, go to YouTube Studio
 - Select your video → Subtitles
 - Add → Upload file → select `pp.vtt`
6. Set visibility and publish

Find Your Video After uploading:

1. Go to `studio.youtube.com`
2. Click **Content** in left sidebar
3. Your video appears in the list
4. Click three dots () → **Get shareable link**

FILE ORGANIZATION

```
palmer_penguins_part1/
  index.qmd                # Blog post (generates images)
  Palmer_Penguins_Video_3min_Guide.md # This guide
  create_slides.py         # Python script to generate ODP

# Analysis images (generated by index.qmd)
penguin-hero-part1.png
eda-overview.png
correlation-matrix.png
simple-regression-model.png
model-diagnostics.png
```

```

species-comparison.png

video_production/
  slides/
    Palmer_Penguins_Part1_3min.odp  # LibreOffice presentation
  png_export/
    slide-1.png
    slide-2.png
    ... (8 PNG files)

  audio/
    slide1_narration.wav
    slide2_narration.wav
    ... (8 WAV files)

Palmer_Penguins_Part1.kdenlive      # Kdenlive project
pp.mp4                             # Final video
pp.vtt                              # Captions
thumbnail.png                       # YouTube thumbnail

```

FINAL CHECKLIST

Before Recording

- ☐ Script printed or on second monitor
- ☐ Quiet room, no background noise
- ☐ Mic 6-8 inches from mouth (audio) or good USB mic (OBS)
- ☐ Water nearby
- ☐ Test recording levels (-12 to -6 dB)
- ☐ For OBS: Good lighting, clean background, camera at eye level
- ☐ For AI Avatar: Intro/outro scripts ready (~25 and ~35 words)

Before Export

- ☐ All 8 slides sync with narration
- ☐ Transitions smooth (dissolve 0.3-0.5 sec, no jarring cuts)
- ☐ Audio levels consistent: narration -6 to -3 dB
- ☐ Background music -15 to -18 dB (intro/outro only)
- ☐ No background noise or pops
- ☐ PiP webcam properly positioned (if using)

Polish Effects Applied

- ☐ Dissolve transitions between slides (0.3-0.5 sec)
- ☐ Ken Burns effect on analysis images (subtle 5% zoom)
- ☐ Lower third name overlay on slide 1 (5-8 sec)
- ☐ Fade in at video start (1 sec)
- ☐ Fade out at video end (1 sec)
- ☐ Music fades: out after intro, in before outro

Before Upload

- ☐ Video plays correctly (watch it through!)
- ☐ Captions reviewed (technical terms correct)

- ☐ Thumbnail created (1280×720)
- ☐ Description includes timestamps
- ☐ All links tested

After Upload

- ☐ Add captions (.vtt file) in YouTube Studio
- ☐ Upload custom thumbnail
- ☐ Add to playlist (Palmer Penguins Series)
- ☐ Share link

QUICK REFERENCE COMMANDS

```
# Regenerate analysis images (update colors in index.qmd first)
cd /path/to/qblog
quarto render posts/palmer_penguins_part1/index.qmd

# Generate slides
cd posts/palmer_penguins_part1
python3 create_slides.py

# Export slides to PNG
cd video_production/slides
/Applications/LibreOffice.app/Contents/MacOS/soffice \
  --headless --convert-to pdf \
  --outdir png_export Palmer_Penguins_Part1_3min.odp
cd png_export
pdftoppm -png -r 300 \
  -scale-to-x 1920 -scale-to-y 1080 \
  Palmer_Penguins_Part1_3min.pdf slide

# Generate captions
cd video_production
whisper pp.mp4 --model base --output_format vtt --output_dir .

# Create thumbnail
convert slides/png_export/slide-1.png -resize 1280x720! \
  -fill 'rgba(0,0,0,0.4)' -draw 'rectangle 0,500 1280,720' \
  -font Helvetica-Bold -pointsize 72 -fill white \
  -gravity south -annotate +0+100 'Part 1' \
  -font Helvetica -pointsize 36 -fill white \
  -gravity south -annotate +0+50 'Simple Regression in 3 min' \
  -fill '#27AE60' -gravity northeast -pointsize 48 \
  -annotate +40+40 'R2 = 0.76' \
  thumbnail.png
```

CLI-BASED WORKFLOW (ALTERNATIVE TO GUI)

Most of the video production can be done via command line using `ffmpeg` and `sox`. This is faster for batch processing and reproducible builds.

Install CLI Tools

```
# macOS
brew install ffmpeg sox imagemagick poppler

# Linux (Debian/Ubuntu)
sudo apt install ffmpeg sox imagemagick poppler-utils
```

Audio Processing with Sox

```
cd video_production/audio

# Normalize audio to -1dB peak
sox slide1_raw.wav slide1_norm.wav gain -n -1

# Noise reduction (two-step process)
# Step 1: Get noise profile from 1 sec of silence
sox slide1_raw.wav -n trim 0 1 noiseprof noise.prof
# Step 2: Apply noise reduction
sox slide1_raw.wav slide1_clean.wav noisered noise.prof 0.21

# Add fade in/out (0.5 sec each)
sox slide1_clean.wav slide1_final.wav fade t 0.5 0 0.5

# Batch process all slides
for i in {1..8}; do
    sox slide${i}_raw.wav -n trim 0 1 noiseprof noise.prof
    sox slide${i}_raw.wav slide${i}_clean.wav noisered noise.prof 0.21
    sox slide${i}_clean.wav slide${i}_final.wav \
        gain -n -1 fade t 0.3 0 0.3
done
```

Video Assembly with FFmpeg

```
cd video_production

# Create file list for concatenation
cat > slides.txt << 'EOF'
file 'slides/png_export/slide-1.png'
duration 10
file 'slides/png_export/slide-2.png'
duration 20
file 'slides/png_export/slide-3.png'
duration 20
file 'slides/png_export/slide-4.png'
duration 25
file 'slides/png_export/slide-5.png'
duration 20
file 'slides/png_export/slide-6.png'
duration 25
file 'slides/png_export/slide-7.png'
duration 15
file 'slides/png_export/slide-8.png'
duration 15
EOF
```

```
# Create video from images (no audio yet)
ffmpeg -f concat -i slides.txt -vf "scale=1920:1080,format=yuv420p" \
-c:v libx264 -r 30 slides_only.mp4
```

Concatenate Audio Files

```
cd video_production/audio

# Create list of audio files
cat > audio_list.txt << 'EOF'
file 'slide1_final.wav'
file 'slide2_final.wav'
file 'slide3_final.wav'
file 'slide4_final.wav'
file 'slide5_final.wav'
file 'slide6_final.wav'
file 'slide7_final.wav'
file 'slide8_final.wav'
EOF

# Concatenate all audio
ffmpeg -f concat -safe 0 -i audio_list.txt -c:a pcm_s16le narration.wav
```

Combine Video + Audio

```
cd video_production

# Merge video and audio
ffmpeg -i slides_only.mp4 -i audio/narration.wav \
-c:v copy -c:a aac -b:a 192k \
-map 0:v:0 -map 1:a:0 \
pp_no_effects.mp4
```

Add Polish Effects with FFmpeg

```
cd video_production

# Add fade in (1 sec) and fade out (1 sec) to video
# Assumes video is 170 seconds (2:50)
ffmpeg -i pp_no_effects.mp4 \
-vf "fade=t=in:st=0:d=1,fade=t=out:st=169:d=1" \
-af "afade=t=in:st=0:d=0.5,afade=t=out:st=169.5:d=0.5" \
-c:v libx264 -c:a aac \
pp_with_fades.mp4
```

Add Ken Burns Effect (Zoom) to Single Image

```
# Subtle zoom in over 20 seconds (100% to 105%)
ffmpeg -loop 1 -i slide-4.png -t 20 \
-vf "scale=8000:-1,zoompan=z='min(zoom+0.0002,1.05)':x='iw/2-(iw/zoom/2)':y='ih/2-(ih/zoom/2)':d=600:" \
-c:v libx264 -pix_fmt yuv420p \
slide4_kenburns.mp4
```

Add Lower Third Overlay

```
# Add text overlay for first 8 seconds
ffmpeg -i pp_with_fades.mp4 \
  -vf "drawtext=text='Your Name | UCSD Biostatistics':fontfile=/System/Library/Fonts/Helvetica.ttc:font:
  -c:v libx264 -c:a copy \
  pp_with_lower_third.mp4
```

Add Background Music (Intro/Outro Only)

```
cd video_production
```

```
# Trim music for intro (10 sec with fade out)
ffmpeg -i background_music.mp3 -t 10 \
  -af "afade=t=out:st=9:d=1,volume=-15dB" \
  music_intro.wav
```

```
# Trim music for outro (15 sec with fade in)
ffmpeg -ss 0 -i background_music.mp3 -t 15 \
  -af "afade=t=in:st=0:d=1,volume=-15dB" \
  music_outro.wav
```

```
# Create silent padding for middle section (assumes 145 sec middle)
ffmpeg -f lavfi -i anullsrc=r=48000:cl=stereo -t 145 silence.wav
```

```
# Concatenate: intro silence + silence + outro silence
ffmpeg -i music_intro.wav -i silence.wav -i music_outro.wav \
  -filter_complex "[0:a][1:a][2:a]concat=n=3:v=0:a=1[out]" \
  -map "[out]" music_track.wav
```

```
# Mix narration (A1) with music (A2)
ffmpeg -i audio/narration.wav -i music_track.wav \
  -filter_complex "[0:a][1:a]amix=inputs=2:duration=longest[out]" \
  -map "[out]" final_audio.wav
```

```
# Combine final audio with video
ffmpeg -i slides_only.mp4 -i final_audio.wav \
  -c:v libx264 -c:a aac -b:a 192k \
  pp_final.mp4
```

Complete CLI Script

Save as build_video.sh:

```
#!/bin/bash
set -e

PROJECT_DIR="video_production"
cd "$PROJECT_DIR"

echo "=== Step 1: Process audio files ==="
cd audio
for i in {1..8}; do
  if [ -f "slide${i}_raw.wav" ]; then
    sox slide${i}_raw.wav -n trim 0 1 noiseprof noise.prof 2>/dev/null || true
    sox slide${i}_raw.wav slide${i}_clean.wav noisered noise.prof 0.21
```

```

    sox slide${i}_clean.wav slide${i}_final.wav gain -n -1 fade t 0.3 0 0.3
    echo "    Processed slide${i}"
fi
done
cd ..

echo "=== Step 2: Concatenate audio ==="
ffmpeg -y -f concat -safe 0 -i audio/audio_list.txt \
    -c:a pcm_s16le audio/narration.wav

echo "=== Step 3: Create video from slides ==="
ffmpeg -y -f concat -i slides.txt \
    -vf "scale=1920:1080,format=yuv420p" \
    -c:v libx264 -r 30 slides_only.mp4

echo "=== Step 4: Combine video + audio ==="
ffmpeg -y -i slides_only.mp4 -i audio/narration.wav \
    -c:v libx264 -c:a aac -b:a 192k \
    -map 0:v:0 -map 1:a:0 \
    pp_draft.mp4

echo "=== Step 5: Add fades ==="
DURATION=$(ffprobe -v error -show_entries format=duration \
    -of default=noprint_wrappers=1:nokey=1 pp_draft.mp4 | cut -d. -f1)
FADE_OUT=$((DURATION - 1))

ffmpeg -y -i pp_draft.mp4 \
    -vf "fade=t=in:st=0:d=1,fade=t=out:st=${FADE_OUT}:d=1" \
    -af "afade=t=in:st=0:d=0.5,afade=t=out:st=${FADE_OUT}.5:d=0.5" \
    -c:v libx264 -c:a aac \
    pp.mp4

echo "=== Step 6: Generate captions ==="
whisper pp.mp4 --model base --output_format vtt --output_dir .

echo "=== Done! Output: pp.mp4 ==="

Make executable and run:

chmod +x build_video.sh
./build_video.sh

```

CLI vs GUI Comparison

Task	CLI Command	When to Use CLI
Audio normalize	<code>sox gain -n -1</code>	Batch processing
Noise reduction	<code>sox noisered</code>	Consistent settings
Video assembly	<code>ffmpeg -f concat</code>	Reproducible builds
Transitions	<code>ffmpeg xfade</code>	Simple dissolves
Ken Burns	<code>ffmpeg zoompan</code>	Batch apply to images
Lower third	<code>ffmpeg drawtext</code>	Same text every video
Fades	<code>ffmpeg fade/afade</code>	Consistent timing
Music mixing	<code>ffmpeg amix</code>	Precise volume control

Use GUI (Kdenlive) when: - Fine-tuning individual clip timing - Complex multi-track editing - Visual preview needed - One-off adjustments

Use CLI (ffmpeg/sox) when: - Batch processing multiple videos - Reproducible, scripted workflow - CI/CD pipeline integration - Consistent settings across series

Total production time:

- Audio-only: 4-5 hours
- PiP webcam: 5-6 hours
- AI Avatar hybrid: 5-6 hours (includes avatar generation time)